



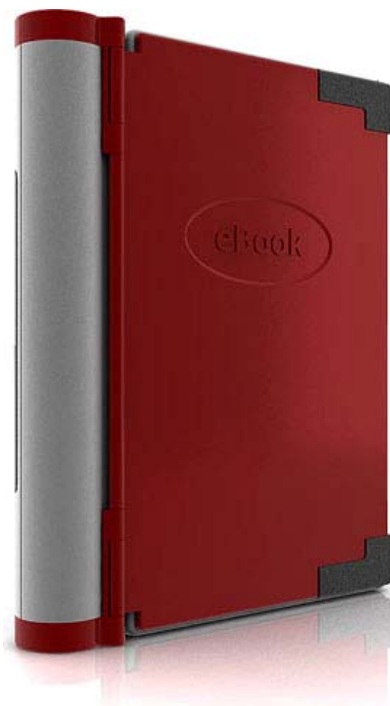
---

**ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ  
ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΘΕΣΣΑΛΟΝΙΚΗΣ**

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

---

**ΣΥΣΤΗΜΑ ΑΝΑΓΝΩΣΗΣ - ΔΙΑΧΕΙΡΙΣΗΣ  
ΗΛΕΚΤΡΟΝΙΚΩΝ ΒΙΒΛΙΩΝ**



**Πτυχιακή εργασία**

Ανδριώτης Απόστολος

Επιβλέπων Καθηγητής : Σαμαράς Απόστολος

**ΘΕΣΣΑΛΟΝΙΚΗ 2007**

"We should not see print and electronic literature as in competition, but rather in conversation. The more voices that join in, the richer the dialogue is likely to be."

*N. Katherine Hayles*

## ΠΕΡΙΕΧΟΜΕΝΑ

|   |    |
|---|----|
| ΠΡΟΛΟΓΟΣ .....  | 7  |
| ΚΕΦΑΛΑΙΟ 1 .....  | 8  |
| ΕΙΣΑΓΩΓΗ .....  | 8  |
| 1.1 Περιγραφή του προβλήματος.....  | 8  |
| 1.2 Ανάγκη επέκτασης .....  | 8  |
| 1.3 Προοπτική του προϊόντος .....   | 9  |
| 1.4 Περιγραφή των εργαλείων που χρησιμοποιήθηκαν .....                            | 9  |
| 1.4.1 Γλώσσα προγραμματισμού Java .....   | 10 |
| 1.4.2 Βάση δεδομένων PostgreSQL.....  | 12 |
| 1.4.3 Αρχιτεκτονική MVC (Model – View – Controller).....                          | 13 |
| 1.4.3.1 Τρόπος λειτουργίας.....   | 13 |
| 1.4.3.2 Πλεονεκτήματα .....   | 14 |
| ΚΕΦΑΛΑΙΟ 2 .....  | 16 |
| ΣΥΣΤΗΜΑΤΑ ΑΝΑΓΝΩΣΗΣ – ΔΙΑΧΕΙΡΙΣΗΣ ΗΛΕΚΤΡΟΝΙΚΩΝ ΒΙΒΛΙΩΝ .....                      | 16 |
| 2.1 Αναδρομή στην ιστορία των ηλεκτρονικών βιβλίων .....                          | 16 |
| 2.2 Πλεονεκτήματα των ηλεκτρονικών βιβλίων .....                                  | 17 |
| 2.3 Μειονεκτήματα των ηλεκτρονικών βιβλίων.....                                   | 18 |
| 2.4 Λογισμικά ανάγνωσης ηλεκτρονικών βιβλίων .....                                | 18 |
| 2.4.1 Περιγραφή της προηγούμενης έκδοσης του συστήματος (Take A Note) .....       | 19 |
| 2.4.2 Microsoft Reader .....  | 20 |
| 2.4.3 Adobe Reader .....  | 21 |
| 2.5 Προτεινόμενες επεκτάσεις του συστήματος .....                                 | 23 |
| ΚΕΦΑΛΑΙΟ 3 .....  | 25 |
| ΑΝΑΛΥΣΗ ΛΕΙΤΟΥΡΓΙΩΝ ΣΥΣΤΗΜΑΤΟΣ ΑΝΑΓΝΩΣΗΣ – ΔΙΑΧΕΙΡΙΣΗΣ ΗΛΕΚΤΡΟΝΙΚΩΝ ΒΙΒΛΙΩΝ ..... | 25 |

|   |    |
|---|----|
| 3.1 Προδιαγραφές συμβατών βιβλίων με την εφαρμογή.....                          | 25 |
| 3.2 Σύγκριση λειτουργιών τρέχουσας έκδοσης με την προηγούμενη.....              | 26 |
| 3.2.1 Λειτουργία υπογράμμισης (Highlight) .....                                 | 26 |
| 3.2.1.1 Περιγραφή.....  | 26 |
| 3.2.1.2 Διαφορές με την πρώτη έκδοση .....                                      | 26 |
| 3.2.1.3 Υλοποίηση .....   | 27 |
| 3.2.2 Λειτουργία σελιδοδεικτών .....  | 27 |
| 3.2.3 Λειτουργία προσωπικών σημειώσεων .....                                    | 28 |
| 3.2.3.1 Περιγραφή.....  | 28 |
| 3.2.3.2 Υλοποίηση .....   | 28 |
| 3.2.3.3 Τρόπος αποθήκευσης των σημειώσεων .....                                 | 29 |
| 3.2.3.4 Επιπλέον χαρακτηριστικά .....   | 30 |
| 3.2.3.5 Εκτύπωση σημειώσεων.....  | 31 |
| 3.3 Νέες λειτουργίες .....  | 31 |
| 3.3.1 Λειτουργία ανοίγματος βιβλίου.....  | 31 |
| 3.3.1.1 Περιγραφή.....  | 31 |
| 3.3.1.2 Υλοποίηση .....   | 32 |
| 3.3.1.3 Προβλήματα που αντιμετωπίστηκαν κατά την υλοποίηση.....                 | 32 |
| 3.3.2 Λειτουργία εγκατάστασης βιβλίου .....                                     | 33 |
| 3.3.2.1 Περιγραφή.....  | 33 |
| 3.3.2.2 Υλοποίηση .....   | 33 |
| 3.3.3 Λειτουργία αναζήτησης .....   | 34 |
| 3.3.3.1 Περιγραφή.....  | 34 |
| 3.3.3.2 Υλοποίηση .....   | 34 |
| 3.3.3.3 Δημιουργία δέντρου και συνδέσμων.....                                   | 36 |
| 3.3.3.4 Προβλήματα που αντιμετωπίστηκαν κατά την υλοποίηση της αναζήτησης ..... | 36 |
| 3.3.4 Λειτουργία αλλαγής της εμφάνισης της εφαρμογής.....                       | 37 |

|  |    |
|--|----|
| 3.3.4.1 Περιγραφή.....   | 37 |
| 3.3.4.2 Υλοποίηση .....  | 38 |
| 3.3.5 Λειτουργία εκτύπωσης κειμένου .....                      | 39 |
| 3.3.5.1 Περιγραφή.....   | 39 |
| 3.3.5.2 Υλοποίηση .....  | 39 |
| 3.3.5.3 Υλοποίηση της μεθόδου print() .....                    | 40 |
| 3.3.5.4 Παραδοχές και εξαρτήσεις.....                          | 41 |
| 3.3.6 Λειτουργία αλλαγής κωδικού .....                         | 41 |
| 3.3.6.1 Περιγραφή.....   | 41 |
| 3.3.6.2 Υλοποίηση .....  | 41 |
| 3.3.7 Λειτουργία σύνδεσης ως διαχειριστής .....                | 42 |
| 3.3.7.1 Περιγραφή.....   | 42 |
| 3.3.7.2 Υλοποίηση .....  | 42 |
| 3.4 Λειτουργίες διαθέσιμες μόνο στους διαχειριστές.....        | 42 |
| 3.4.1 Περιγραφή.....   | 42 |
| 3.4.2 Δημιουργία νέου χρήστη.....                              | 43 |
| 3.4.3 Διαγραφή χρήστη .....                                    | 43 |
| 3.4.4 Δημιουργία νέου βιβλίου .....                            | 43 |
| 3.4.4.1 Παραδοχές και εξαρτήσεις.....                          | 45 |
| 3.5 Επιπλέον χαρακτηριστικά .....                              | 46 |
| 3.5.1 Κρυπτογράφηση των κωδικών των χρηστών (Encryption) ..... | 46 |
| 3.5.1.1 Περιγραφή.....   | 46 |
| 3.5.1.2 Υλοποίηση .....  | 46 |
| 3.5.1.3 Εναλλακτικές λύσεις.....                               | 47 |
| 3.5.1.4 Δημιουργία αρχικών χρηστών .....                       | 48 |
| 3.5.2 Διασφάλιση ακεραιότητας χρηστών.....                     | 48 |
| 3.5.2.1 Περιγραφή.....   | 48 |

|  |    |
|--|----|
| 3.5.2.2 Υλοποίηση .....                                    | 49 |
| 3.5.3 Χρήση ανακυκλώσιμων συνδέσεων (Connection Pool)..... | 49 |
| 3.5.3.1 Περιγραφή.....                                     | 49 |
| 3.5.3.2 Τρόπος λειτουργίας.....                            | 50 |
| 3.5.3.3 Υλοποίηση .....                                    | 50 |
| 3.6 Βάση δεδομένων.....                                    | 53 |
| ΚΕΦΑΛΑΙΟ 4 .....   | 56 |
| ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....               | 56 |
| 4.1 Συμπεράσματα.....                                      | 56 |
| 4.2 Μελλοντικές επεκτάσεις .....                           | 56 |
| ΒΙΒΛΙΟΓΡΑΦΙΑ .....   | 58 |
| ΠΑΡΑΡΤΗΜΑ 1.....   | 61 |
| Απαιτήσεις συστήματος .....                                | 62 |
| Εγχειρίδιο τοπικής εγκατάστασης .....                      | 62 |
| Εγχειρίδιο Χρήσης.....                                     | 69 |

## ΠΡΟΛΟΓΟΣ

Κατά το ακαδημαϊκό έτος 2006 – 2007, οι φοιτητές Γκίσης Τραϊάνος και Ζαφειρόπουλος Ιωάννης παρέδωσαν την πτυχιακή εργασία με τίτλο «Ένα σύστημα προσαρμογής ηλεκτρονικού βιβλίου στις ανάγκες των χρηστών» υπό την επίβλεψη του κ. Σταμάτη Δημοσθένη. Η εργασία αφορούσε στην ανάπτυξη συστήματος λογισμικού ανάγνωσης και επεξεργασίας ηλεκτρονικών βιβλίων, τα οποία ο χρήστης μπορεί να χρησιμοποιήσει παρόμοια με την έντυπη μορφή τους.

Ο χρήστης του συστήματος, το οποίο και ονομάστηκε “Take A Note 1.0”, μπορεί να εκτελεί διάφορες λειτουργίες όπως για παράδειγμα να υπογραμμίζει σημεία του κειμένου τα οποία αυτός θεωρεί σημαντικά, να κρατάει σημειώσεις σε κάθε σελίδα, και να βάζει σελιδοδείκτες για πιο εύκολη πρόσβαση στις σελίδες του βιβλίου. Κάθε φορά που συνδέεται στο σύστημα, έχει πρόσβαση μονάχα στις δικές του ρυθμίσεις, καθιστώντας έτσι το βιβλίο μοναδικό και προσαρμοζόμενο στις ανάγκες του. Μέσω του συστήματος αυτού, ένα βιβλίο ουσιαστικά διανέμεται σε πολλά αντίτυπα, τα οποία ο κάθε χρήστης είναι ελεύθερος να τροποποιήσει κατά την βούληση του.

Με την παρούσα πτυχιακή εργασία θα γίνουν τα πρώτα βήματα επέκτασης και βελτίωσης αυτού του συστήματος.

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω τον κ. Σαμαρά Απόστολο για την πολύτιμη βοήθεια του καθ’ όλο το διάστημα της υλοποίησης, καθώς επίσης και για τις ιδέες και διορθώσεις που πρότεινε.

Θεσσαλονίκη, Νοέμβριος 2007  
Ανδριώτης Απόστολος

# ΚΕΦΑΛΑΙΟ 1

## ΕΙΣΑΓΩΓΗ

### 1.1 Περιγραφή του προβλήματος

Το σύστημα προσαρμοζόμενου ηλεκτρονικού βιβλίου “Take A Note 1.0”, ήταν μια πρώτη ικανοποιητική προσέγγιση στο θέμα της ανάγνωσης - διαχείρισης ηλεκτρονικών βιβλίων. Οι λειτουργίες του κάλυπταν επαρκώς τις βασικές ανάγκες του χρήστη, καθιστώντας την διαδικασία ανάγνωσης ηλεκτρονικού βιβλίου εύκολη, γρήγορη και κατ’ επέκταση αποδοτική. Πρέπει επίσης σε αυτό το σημείο να διευκρινιστεί ότι το σύστημα αυτό αποτελούσε υποσύστημα ενός μεγαλύτερου πληροφοριακού συστήματος με όνομα **tei-Book**, το οποίο και είχε προταθεί τους κκ. Δ. Σταμάτη, Α. Τσαδύρα, Κ. Χατζάρα στο άρθρο τους “*Developing Personalized E-Books: A Multi-Layered Approach*”.

Στόχος της εργασίας αυτής είναι η επέκταση του αρχικού συστήματος ώστε να καλύπτει περισσότερες από τις ανάγκες/απαιτήσεις του χρήστη, αλλά και η βελτίωση των υπάρχουσών λειτουργιών. Απώτερος σκοπός είναι η κατασκευή ενός ολοκληρωμένου και φιλικού προς τον χρήστη συστήματος που θα ανταποκρίνεται πλήρως στις ανάγκες του. Το έναυσμα για να επιχειρηθεί κάτι τέτοιο δόθηκε από την ίδια την αρχική ομάδα ανάπτυξης, οι οποίοι στο έγγραφο τους, προσέφεραν ένα εκτενή κατάλογο με προτεινόμενες μελλοντικές βελτιώσεις/προσθήκες.

### 1.2 Ανάγκη επέκτασης

Η τεχνολογίες πληροφορικής εξελίσσονται με ραγδαίους και συχνά αμείλικτους ρυθμούς στις μέρες μας. Ο χρήστης βομβαρδίζεται καθημερινά από σωρεία νέων εφαρμογών που προσφέρουν νέες και πιο προηγμένες λειτουργίες. Ως αποτέλεσμα αυτής της τακτικής είναι αφενός η αύξηση των απαιτήσεων των χρηστών, αφετέρου η δημιουργία και διάθεση λογισμικού τέτοιας πολυπλοκότητας που αποθαρρύνει τον μέσο χρήστη που δεν είναι εξοικειωμένος με εξειδικευμένες ορολογίες και διαδικασίες.



Η επέκταση του αρχικού συστήματος Take A Note 1.0, έγινε με γνώμονα την πληρέστερη κάλυψη των αναγκών του χρήστη και παράλληλα την δημιουργία ενός user interface μειωμένης πολυπλοκότητας. Έτσι ενώ εμπλουτίστηκαν οι ήδη υπάρχουσες λειτουργίες και προστέθηκαν στο σύστημα νέες, δεν υπάρχει επιβάρυνση στην χρηστικότητα και τον χειρισμό του συστήματος.

### 1.3 Προοπτική του προϊόντος

Ο τρόπος υλοποίησης του συστήματος το καθιστά χρήσιμο σχεδόν σε κάθε τομέα της μέσης και ανώτερης εκπαίδευσης. Προσφέρει λύση για την διάθεση βιβλίων σε πολλαπλά αντίτυπα με ελάχιστο έως και μηδενικό κόστος, παρέχοντας ταυτόχρονα εργαλεία και λειτουργίες που βοηθάνε τους σπουδαστές και εμπλουτίζουν την εκπαιδευτική διαδικασία. Σε κανένα βαθμό δεν υποστηρίζουμε ότι το παρόν σύστημα είναι ικανό να αντικαταστήσει πλήρως τα έντυπα βιβλία. Ωστόσο, μπορεί να χρησιμοποιηθεί επικουρικά συμβάλλοντας στην μετάβαση της εκπαιδευτικής διεργασίας στην ψηφιακή εποχή.

### 1.4 Περιγραφή των εργαλείων που χρησιμοποιήθηκαν

Η προηγούμενη ομάδα ανάπτυξης του συστήματος είχε χρησιμοποιήσει για την υλοποίηση του κώδικα την γλώσσα προγραμματισμού Java και την PostgreSQL για την υλοποίηση της βάσης δεδομένων, για ποικίλους λόγους. Για την βελτίωση της εφαρμογής, αποφασίστηκε να χρησιμοποιηθούν τα ίδια εργαλεία για λόγους συμβατότητας, εξοικονόμησης χρόνου (για παράδειγμα η περίπτωση αντικατάστασης της Java με κάποια άλλη γλώσσα προγραμματισμού συνεπάγεται και την εκ νέου προσαρμογή όλου του μέχρι τώρα συστήματος στα δεδομένα της νέας γλώσσας) αλλά κυρίως για τους λόγους που παρατίθενται παρακάτω. Ακόμη αποφασίστηκε και η διατήρηση της αρχιτεκτονικής ανάπτυξης που προσφέρει το μοντέλο MVC (Model View Control). Συνοψίζοντας, χρησιμοποιήθηκαν τα παρακάτω εργαλεία :

- 1) Η γλώσσα προγραμματισμού Java και το σύστημα ODBC-JDBC για να ενωθεί η εφαρμογή με την βάση δεδομένων.
- 2) Η βάση δεδομένων PostgreSQL
- 3) Αρχιτεκτονική Model/View/Controller (MVC)
- 4) Για την συγγραφή και τροποποίηση του κώδικα το εργαλείο Netbeans IDE (Integrated Development Environment).

- 5) Το πρόγραμμα InstallAnywhere 8 της Macrovision για την δημιουργία των εκτελέσιμων αρχείων
- 6) Για την δημιουργία του UML διαγράμματος το πρόγραμμα ArgoUML 0.18.1
- 7) Για την δημιουργία του διαγράμματος των συσχετίσεων της βάσης δεδομένων (ER) η δοκιμαστική έκδοση του προγράμματος MicroOLAP Database Designer for PostgreSQL

Ακολουθεί μια σύντομη περιγραφή ορισμένων από τα παραπάνω εργαλεία.

### 1.4.1 Γλώσσα προγραμματισμού Java

Ένας από τους λόγους που προτιμήθηκε η Java έναντι των άλλων γλωσσών προγραμματισμού είναι τα πλεονεκτήματα που έχει έναντι άλλων γλωσσών προγραμματισμού τα οποία και παρουσιάζουμε στην συνέχεια.

#### Αντικειμενοστραφής γλώσσα

Οι αντικειμενοστραφείς γλώσσες προγραμματισμού παρέχουν πολλά πλεονεκτήματα στην ανάπτυξη λογισμικού, που έχουν σχέση με την επαναχρησιμοποίηση των αντικείμενων, την μεγαλύτερη ελαστικότητα της γλώσσας και την ανάπτυξη αρθρωτών προγραμμάτων. Για χρόνια, η αντικειμενοστραφής τεχνολογία θεωρούνταν ελιτιστική και απαιτούσε μεγάλες επενδύσεις σε εκπαίδευση και υποδομή. Η Java βοήθησε την αντικειμενοστραφή τεχνολογία να μπει στην καθημερινότητα, παρέχοντας απλή και καθαρή δομή η οποία επιτρέπει στον προγραμματιστή να διαβάζει και να γράφει κώδικα πολύ εύκολα.

#### Ευκολία χρήσης

Καμία γλώσσα προγραμματισμού δεν είναι εύκολη, αλλά η Java συγκρινόμενη με την αρκετά δημοφιλή γλώσσα C++, θεωρείτε ως πολύ πιο απλή και εύκολη γλώσσα αντικειμενοστραφούς προγραμματισμού. Μερικώς σχεδιασμένη με βάση τη C++, έχει αντικαταστήσει την πολυπλοκότητα της πολλαπλής κληρονομικότητας που συναντάτε στην C++ με μια απλούστερη δομή που ονομάζεται Interface, έχοντας επίσης εξαλείψει την χρήση των pointers.

Ο κύριος λόγος όμως που η Java είναι πολύ πιο εύκολη από την C++, είναι η χρήση της αυτόματης κατανομής της μνήμης, όπως επίσης και ο αυτόματος garbage collector. Ιδιαίτερα απαιτητικές εργασίες που γίνονται μόνες τους στην Java, στη C++ απαιτείται από τον ίδιο τον προγραμματιστή να τις εκτελέσει.

### Ανεξαρτησία πλατφόρμας

Ένας από τους πιο ισχυρούς λόγους για κάποιον να μεταβεί στην Java είναι και το γεγονός ότι είναι ανεξάρτητη πλατφόρμας. Είναι συμβατή με τις περισσότερες πλατφόρμες υλικού και λογισμικού, συμπεριλαμβανομένων των Windows, των Macintosh και αρκετών διανομών του Unix. Με το να μεταφερθεί ένα ήδη υπάρχον λογισμικό σε γλώσσα Java, καθίσταται αυτομάτως και συμβατό με όλες αυτές τις πλατφόρμες λογισμικού, αναιρώντας οποιαδήποτε εξάρτηση από υλικό ή λειτουργικό σύστημα.

### Μεταφράζεται σε bytecode

Για να εκτελεστεί ένα πρόγραμμα που είναι γραμμένο σε Java, πρέπει πρώτα να μεταφραστεί (Interpreted) σε κώδικα Java Virtual Machine, ο οποίος και ονομάζεται bytecode. Ο bytecode κώδικας είναι ανεξάρτητος της μηχανής και μπορεί να τρέξει σε οποιοδήποτε μηχάνημα έχει τον μεταφραστή (interpreter) της Java. Κανονικά, ένας compiler θα μετέφραζε μια υψηλού επιπέδου γλώσσα προγραμματισμού σε κώδικα μηχανής. Και αυτός ο κώδικας θα μπορούσε μονάχα να εκτελεστεί από το μηχάνημα που τον παρήγαγε. Εάν το πρόγραμμα πρέπει να εκτελεστεί και σε άλλο μηχάνημα, πρέπει και η διαδικασία της μετάφρασης (compilation) να εκτελεστεί ξανά. Για παράδειγμα εάν κάνουμε compile πρόγραμμα γραμμένο σε C++ σε Windows λειτουργικό, ο εκτελέσιμος κώδικας που θα παραχθεί θα μπορεί να τρέξει μόνο στα Windows.

### Ασφάλεια

Η Java είναι μια από τις πρώτες γλώσσες προγραμματισμού που θεωρούν την ασφάλεια μέρος του σχεδιασμού τους. Ολόκληρη η Java, μαζί με όλα τα στοιχεία από τα οποία αποτελείται, αναπτύχθηκε έχοντας κατά νου την ασφάλεια. Ο compiler, ο interpreter και όλοι οι browsers που είναι συμβατοί με Java, περιέχουν πολλά επίπεδα μέτρων ασφαλείας, που είναι σχεδιασμένα ώστε να μειώνουν το ρίσκο οποιουδήποτε συμβιβασμού σχετικά με την ασφάλεια, την απώλεια δεδομένων και ακεραιότητας των χρηστών του προγράμματος. Δεδομένου των τεράστιων προβλημάτων που σχετίζονται με την εκτέλεση αμφιβόλου αξιοπιστίας κώδικα με φαινομενικά ασφαλή τρόπο, τα μέτρα ασφαλείας της Java είναι πολύ μπροστά, ακόμα και από εκείνα που αναπτύχθηκαν για να ασφαλίσουν στρατιωτικά συστήματα.

### Αξιοπιστία

Η Java προσφέρει πολλά επίπεδα ελέγχων αξιοπιστίας, με απαρχή την ίδια την γλώσσα. Πολλά από τα χαρακτηριστικά της C και της C++ που είναι επιζήμια για την αξιοπιστία του προγράμματος, όπως για παράδειγμα είναι οι pointers και η αυτόματη μετατροπή μεταξύ διαφορετικών τύπων, αποφεύγονται στην Java. Ο compiler της Java παρέχει πολλά επίπεδα επιπρόσθετων ελέγχων για την αναγνώριση μη ταιριάσματος μεταξύ τύπων

και άλλων ασυνεπειών. Το σύστημα εκτέλεσης της Java (Java Runtime System) επαναλαμβάνει πολλούς από τους ελέγχους που γίνονται από τον compiler και εκτελεί και κάποιους επιπλέον για να πιστοποιήσει ότι τα bytcodes σχηματίζουν ένα έγκυρο πρόγραμμα.

## 1.4.2 Βάση δεδομένων PostgreSQL

Η βάση δεδομένων PostgreSQL επιλέχθηκε αρχικά για την ελεύθερη διάθεση της τόσο για προσωπική όσο και για επαγγελματική χρήση. Επιπρόσθετα, είναι σύστημα ανοικτού κώδικα (open source), το οποίο πρακτικά σημαίνει ότι ο καθένας μπορεί να προμηθευτεί τον πηγαίο κώδικα και να το μορφοποιήσει.

### Χαρακτηριστικά

Η PostgreSQL έχει διαθέσιμα τα περισσότερα χαρακτηριστικά από αυτά που έχουν και τα μεγάλα εμπορικά συστήματα διαχείρισης βάσεων δεδομένων, όπως είναι η διεξαγωγή εργασιών (transactions), οι υπο-επιλογές (subselects), τα εναύσματα (triggers), οι όψεις (views) και η ακεραιότητα ξένων κλειδιών (foreign key referential integrity). Έχει επίσης και κάποια χαρακτηριστικά που τα εμπορικά πακέτα δεν έχουν, όπως για παράδειγμα η ελευθερία δημιουργίας νέων τύπων δεδομένων από τους χρήστες και η κληρονομικότητα.

### Αξιοπιστία

Η ομάδα ανάπτυξης της PostgreSQL αντιλαμβάνεται το γεγονός ότι ένα DBMS πρέπει να είναι αξιόπιστο, διαφορετικά είναι άχρηστο. Στοχεύουν στο να βγάζουν καλά ελεγμένο, σταθερό κώδικα με όσο το δυνατόν λιγότερα λάθη. Κάθε έκδοση έχει περάσει τουλάχιστον από ένα μήνα ελέγχου, το οποίο επιβεβαιώνετε και από το μέχρι τώρα ιστορικό σταθερών και αξιόπιστων εκδόσεων.

### Υποστήριξη

Η PostgreSQL έχει από πίσω της μια πολύ ενεργή κοινότητα από επαγγελματίες προγραμματιστές και απλούς χρήστες οι οποίοι είναι πρόθυμοι να βοηθήσουν στο να βρεθεί λύση σε ό,τι πρόβλημα και αν παρουσιαστεί με μηδενικό κόστος.

Συγκριτικά με άλλες βάσεις δεδομένων όπως η MySQL και η Interbase :

- ✓ Είναι πιο γρήγορη και πιο αποδοτική
- ✓ Υποστηρίζει κληρονομικότητα

- ✓ Είναι πιο ανθεκτική σε περιπτώσεις κρσαρίσματος και πτώσης του ρεύματος
- ✓ Υποστηρίζει την δημιουργία νέων τύπων δεδομένων από τους χρήστες
- ✓ Υποστηρίζει άπειρο αριθμό σειρών, άπειρο μέγεθος βάσης και οι πίνακες μπορούν να είναι μεγέθους μέχρι και 16TB
- ✓ Υποστηρίζει υποερωτήματα (γνώρισμα το οποίο το έχει και η Interbase αλλά όχι η MySQL)
- ✓ Έχει διαθέσιμο ένα από τα καλύτερα κείμενα επεξήγησης
- ✓ Και το σημαντικότερο δεν ανήκει σε κάποια εταιρεία, αλλά αναπτύσσεται από κοινότητα, διασφαλίζοντας έτσι το μέλλον της.

Βάσεις οι οποίες αναπτύσσονται από εταιρείες εξαρτώνται άμεσα από την τύχη της εκάστοτε εταιρείας, το οποίο σημαίνει ότι μπορεί κάποια στιγμή στο μέλλον το DBMS να μην υποστηρίζεται πλέον.

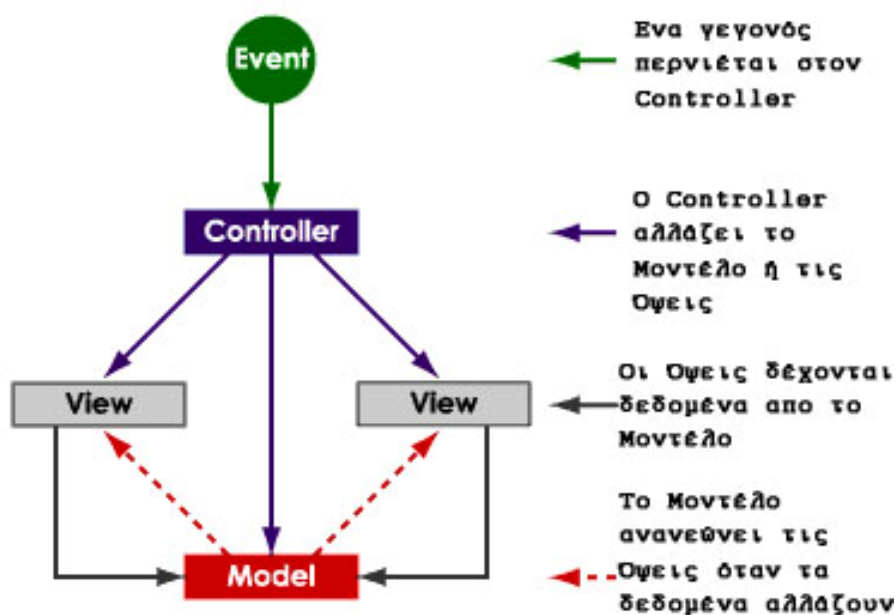
### **1.4.3 Αρχιτεκτονική MVC (Model – View – Controller)**

Το MVC είναι ένα κλασσικό σχεδιαστικό πρότυπο που χρησιμοποιείται συχνά από εφαρμογές που χρειάζονται την δυνατότητα να διατηρούν πολλαπλές διαφορετικές παρουσιάσεις/όψεις των ίδιων δεδομένων. Το MVC πρότυπο βασίζεται στον σαφή διαχωρισμό των αντικειμένων σε τρεις κατηγορίες – τα μοντέλα για να διατηρούν τα δεδομένα (model), τις όψεις (view) για την παρουσίαση όλων ή μέρους των δεδομένων και τους controllers για να διαχειρίζονται τις ενέργειες που επηρεάζουν τα μοντέλα ή τις όψεις.

Εξαιτίας αυτού του διαχωρισμού, πολλές όψεις και πολλοί controllers μπορούν να αλληλεπιδρούν με το ίδιο μοντέλο.

#### **1.4.3.1 Τρόπος λειτουργίας**

Ο τρόπος λειτουργίας του μοντέλου MVC μπορεί να αναπαρασταθεί με το παρακάτω γράφημα :



Εικόνα 1.1 : Τρόπος λειτουργίας του προτύπου MVC

Ένα γεγονός (event) προκαλεί τον Controller να αλλάξει το μοντέλο, την όψη ή και τα δύο. Όταν ένας Controller αλλάζει τα δεδομένα ή τις ιδιότητες κάποιου μοντέλου, ανανεώνονται και όλες οι όψεις που εξαρτώνται από αυτό. Με τον ίδιο τρόπο όταν ένας Controller αλλάζει μια όψη, για παράδειγμα με το να αποκαλύπτει περιοχές που μέχρι πρότινος δεν ήταν ορατές, η όψη αντλεί δεδομένα από το μοντέλο για να ανανεωθεί.

### 1.4.3.2 Πλεονεκτήματα

Μερικά από τα πλεονεκτήματα της αρχιτεκτονικής MVC είναι τα παρακάτω :

- 1) **Σαφήνεια σχεδιασμού** : Παρατηρώντας κανείς απλά τις μεθόδους του μοντέλου, μπορεί εύκολα να καταλάβει τον τρόπο με τον οποίο μπορεί να ελέγξει τη συμπεριφορά του. Το γεγονός αυτό καθιστά όλο το πρόγραμμα ευκολότερο στην υλοποίηση και συντήρηση του.
- 2) **Αποδοτικός σχεδιασμός** : Ο τρόπος με τον οποίο σχεδιάζεται το πρόγραμμα επιτρέπει τόσο στον χρήστη όσο και στον προγραμματιστή να αντικαταστήσει τα συστατικά της εφαρμογής κατά βούληση. Οι αλλαγές σε κάποιο μέρος του προγράμματος δεν συνδέονται με άλλα μέρη, διευκολύνοντας την αποσφαλμάτωση.
- 3) **Πολλαπλές όψεις** : Η εφαρμογή μπορεί να εμφανίσει την κατάσταση ενός μοντέλου με πολλούς διαφορετικούς τρόπους. Όλες οι όψεις χρησιμοποιούν τα ίδια ακριβώς δεδομένα αλλά τα παρουσιάζουν με τελείως διαφορετικό τρόπο.
- 4) **Ευκολία ανάπτυξης** : Οι Controllers και οι όψεις μπορούν να αναπτύσσονται καθ' όλη την διάρκεια ανάπτυξης του μοντέλου. Όλες οι

παλιές όψεις και οι παλιοί controllers μπορούν να χρησιμοποιούνται όσο η κοινή διεπιφάνεια διατηρείται.

## ΚΕΦΑΛΑΙΟ 2

### ΣΥΣΤΗΜΑΤΑ ΑΝΑΓΝΩΣΗΣ – ΔΙΑΧΕΙΡΙΣΗΣ ΗΛΕΚΤΡΟΝΙΚΩΝ ΒΙΒΛΙΩΝ

#### 2.1 Αναδρομή στην ιστορία των ηλεκτρονικών βιβλίων

Η θεωρία των φορητών ηλεκτρονικών βιβλίων έχει αρχίσει να εξελίσσεται από πολύ παλιά, από τα τέλη ακόμα της δεκαετίας του '60 και τις αρχές του '70, πολύ πριν οι ηλεκτρονικοί υπολογιστές γίνουν διαθέσιμοι στο ευρύ κοινό. Το 1968, ο Alan Kay, μεταπτυχιακός τότε φοιτητής στο πανεπιστήμιο της Utah, παρουσίασε ένα ηλεκτρονικό βιβλίο με το όνομα Dynabook. Ο Kay οραματίστηκε ένα φορητό υπολογιστή με δυνατότητες αλληλεπίδρασης και με χαρακτηριστικά μια επίπεδη οθόνη και ασύρματη επικοινωνία. Η ιδέα του Dynabook αναπτύχθηκε περαιτέρω κατά την διάρκεια εργασίας του Kay πάνω στους προσωπικούς υπολογιστές και το γραφικό περιβάλλον στο ερευνητικό κέντρο Xerox Palo Alto.

Ο Andries Van Dam, αναγνωρίζεται από πολλούς ως αυτός που επινόησε τον όρο «ηλεκτρονικό βιβλίο (electronic book)», πριν από 40 χρόνια. Το '67 και το '68, ο Van Dam ηγήθηκε μιας ομάδας η οποία και ανέπτυξε το πρώτο hypertext σύστημα. «Το Hyper Text Editing System έτρεχε σε ένα τερματικό IBM/360 με 128 K μνήμης και χρηματοδοτήθηκε από την IBM, η οποία και αργότερα το πούλησε στο Houston Manned Spacecraft Center, όπου και χρησιμοποιήθηκε για να παραχθεί η τεκμηρίωση για το διαστημικό πρόγραμμα Apollo». Κατά την διάρκεια του 1970, χρησιμοποιήθηκε ένα σύστημα ανάκτησης και επεξεργασίας αρχείων (File Retrieval and Editing System - FRESS) από φοιτητές και μέλη πανεπιστημίου του Brown. «Το FRESS είχε δυναμική ιεραρχία (όπως κεφάλαια/ενότητες) , διπλής κατεύθυνσης συνδέσμους παραπομπών και συνδέσμους με λέξεις κλειδιά».

Το 1971, ο Michael Hart ξεκίνησε το σχέδιο Guttenberg (<http://www.gutenberg.org/>), όταν του δόθηκε επιχορήγηση του ύψους των 100 εκατομμυρίων δολαρίων από το εργαστήριο έρευνας υλικού (Materials Research Lab) του πανεπιστημίου του Illinois. Ο Hart πληκτρολόγησε την «Διακήρυξη της Ανεξαρτησίας» και διέθεσε το κείμενο μέσω πολλών δικτύων. Από το '71, η ομάδα του προγράμματος Guttenberg είναι υπεύθυνη για την διάθεση κοντά 2,100 τίτλων.



Σύμφωνα με το άρθρο «*The Seybold Report on Internet Publishing*», οι εκδοτικοί οίκοι ψηφιοποιούν βιβλία για περισσότερο από μία δεκαετία. Κατά την διάρκεια των αρχών του '90, η Sony προσπάθησε να υλοποιήσει διάφορες εκδόσεις ενός φορητού ηλεκτρονικού βιβλίου, δίχως επιτυχία. Την ίδια στιγμή, εγκυκλοπαίδειες με τη μορφή CD-ROM, οι οποίες συνδύαζαν στοιχεία αλληλεπίδρασης και multimedia, έγιναν αρκετά δημοφιλής στο καταναλωτικό κοινό. Το Adobe Acrobat (το οποίο και χρησιμοποιούσε αρχεία που έγιναν γνωστά ως PDF – Portable Document Format), παρουσιάστηκε ως εμπορικό λογισμικό για την έκδοση και παρουσίαση ηλεκτρονικών βιβλίων. Ωστόσο, οι συσκευές χειρός για την ανάγνωση ηλεκτρονικών βιβλίων, είναι ένα σχετικά πρόσφατο φαινόμενο, το οποίο έκανε την εμφάνιση του κατά το δεύτερο μισό του 1998.

## 2.2 Πλεονεκτήματα των ηλεκτρονικών βιβλίων

Υπάρχουν αρκετά πλεονεκτήματα με την χρήση συστημάτων ηλεκτρονικών βιβλίων. Τα πιο σημαντικά από αυτά είναι:

- ✓ Γρήγορη και εύκολη αναζήτηση λέξεων – φράσεων εντός του κειμένου
- ✓ Απαιτείται ελάχιστος αποθηκευτικός χώρος σε σχέση με τα συμβατικά βιβλία. Για παράδειγμα, 500 τίτλοι βιβλίων μπορούν να χωρέσουν σε ένα απλό CD. Με την εξέλιξη της τεχνολογίας ο χώρος συνεχώς συρρικνώνεται παρασύροντας και το κόστος αποθήκευσης (blue ray disc, hd-dvd).
- ✓ Τα ηλεκτρονικά βιβλία μπορούν να διατίθενται για απεριόριστο χρονικό διάστημα και δεν φθείρονται. Έτσι οι αναγνώστες μπορούν να διατηρούν αναλλοίωτα βιβλία που χρησιμοποιούν αλλά και να αποκτήσουν βιβλία τα οποία πλέον δεν είναι διαθέσιμα σε έντυπη μορφή.
- ✓ Αναγνώστες που ίσως έχουν δυσκολίες στην ανάγνωση εντύπων, μπορούν να επωφεληθούν από το γεγονός ότι το μέγεθος του κειμένου είναι προσαρμόσιμο. Ακόμη, λογισμικό text-to-speech μπορεί να χρησιμοποιηθεί, μετατρέποντας το γραπτό κείμενο σε ήχο.
- ✓ Εξαλείφεται το πρόβλημα της σκίασης του κειμένου που βρίσκεται στο εσωτερικό περιθώριο του βιβλίου.
- ✓ Μηδενικό ή ελάχιστο κόστος αντιγραφής ενός ηλεκτρονικού βιβλίου. Αντίγραφα μπορούν να δημιουργηθούν άμεσα και σε οποιαδήποτε επιθυμητή ποσότητα.

- ✓ Ηλεκτρονικά βιβλία μπορούν να εκδοθούν από ανεξάρτητους εκδοτικούς οίκους ή ακόμα και από ιδιώτες, το οποίο συνεπάγεται μεγαλύτερη εκδοτική και συγγραφική ελευθερία.
- ✓ Με την χρήση του διαδικτύου, η ευκολία διάθεσης των ηλεκτρονικών βιβλίων αποτελεί μεγάλο πλεονέκτημα.
- ✓ Τα λάθη στο κείμενο μπορούν εύκολα και γρήγορα να διορθωθούν.
- ✓ Δεν καταναλώνονται πόροι από το φυσικό περιβάλλον για την αντιγραφή τους.

### **2.3 Μειονεκτήματα των ηλεκτρονικών βιβλίων**

Όπως σε κάθε νέα τεχνολογία υπάρχουν εν γένει μειονεκτήματα από την εφαρμογή της. Τα πιο σημαντικά από αυτά είναι:

- ✓ Είναι απαραίτητη η ύπαρξη κάποιας ηλεκτρονικής συσκευής προκειμένου να τα διαβάσει κάποιος.
- ✓ Πολλές μορφές ηλεκτρονικών βιβλίων χρειάζονται ειδικό λογισμικό, το οποίο μπορεί και να μην διατίθεται ελεύθερα.
- ✓ Με τεχνικές διαχείρισης ηλεκτρονικών δικαιωμάτων μπορεί να τίθενται περιορισμοί σχετικά με το τι μπορεί κάποιος να κάνει με αυτό. Για παράδειγμα μπορεί να μην είναι δυνατό η ιδιοκτησία ενός βιβλίου να μεταφερθεί σε κάποιον άλλον, κάτι το οποίο και συναντάτε πολύ συχνά με τα έντυπα βιβλία.
- ✓ Η ανάλυση μιας οθόνης είναι πολύ χαμηλότερης ποιότητας συγκριτικά με το χαρτί.
- ✓ Το περιεχόμενο του ηλεκτρονικού βιβλίου μπορεί να αλλαχθεί δίχως την άδεια του συγγραφέα ή του εκδότη.
- ✓ Η ανάγνωση ενός ηλεκτρονικού βιβλίου μπορεί να γίνει πολύ κουραστική μετά από λίγη ώρα και ο αναγνώστης να χάσει τη συγκέντρωση του.

### **2.4 Λογισμικά ανάγνωσης ηλεκτρονικών βιβλίων**

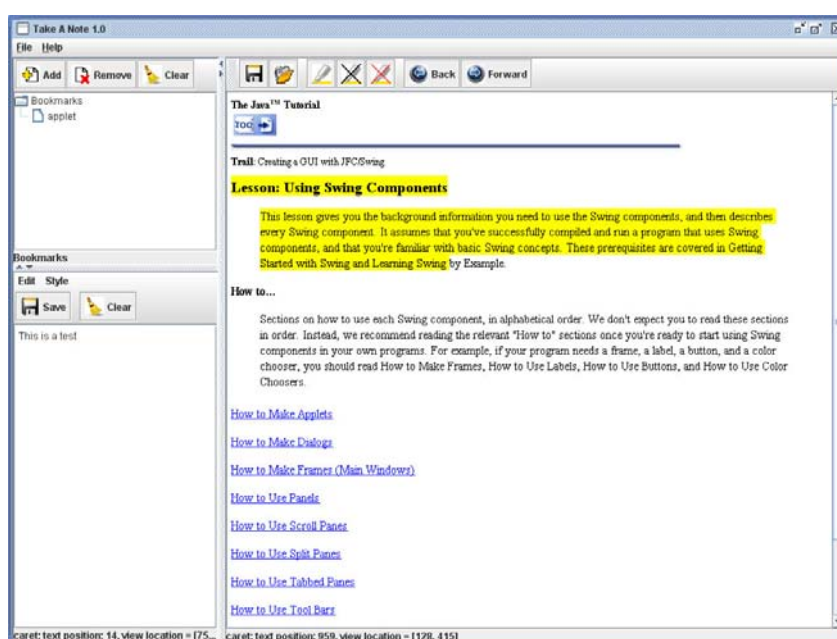
Προκειμένου να καταλήξουμε σε μία πιο εμπειριστατωμένη λίστα με νέες λειτουργίες και βελτιώσεις για το υπάρχον σύστημα, αφενός διερωτηθήκαμε οι ίδιοι τι απαιτήσεις θα είχαμε σαν χρήστες ενός τέτοιου προγράμματος και αφετέρου δοκιμάσαμε ορισμένα από τα ήδη υπάρχοντα

λογισμικά που ειδικεύονται σε αυτόν τον τομέα. Θα ήταν φυσικά παράληψη να μην αναφέρουμε ότι σημαντικό ρόλο έπαιξαν και οι προτάσεις που διατυπώθηκαν στο έγγραφο της εργασίας της οποίας συνέχεια αποτελεί η παρούσα, από τους Ζαφειρόπουλο και Γκίση.

## 2.4.1 Περιγραφή της προηγούμενης έκδοσης του συστήματος (Take A Note)

### Βασικές λειτουργίες

- ❖ **Highlight** : Η λειτουργία Highlight δίνει την δυνατότητα στον αναγνώστη-χρήστη να επέμβει πάνω στο e-book και να υπογραμμίσει τα σημεία του κειμένου που αυτός θεωρεί ότι χρειάζονται ιδιαίτερη προσοχή.
- ❖ **Bookmarks** : Τα bookmarks προσομοιώνουν την λειτουργία των συνδετήρων και προσφέρουν στο χρήστη δυνατότητα εύκολης και γρήγορης πρόσβασης σε όλες εκείνες τις σελίδες του βιβλίου που αντιστοιχούν σε bookmarks.
- ❖ **Notes** : Η λειτουργία αυτή δίνει την δυνατότητα στον χρήστη-αναγνώστη να επέμβει πάνω στο βιβλίο και να καταγράψει κάποια προσωπικά του σχόλια για μεγαλύτερη κατανόηση και επεξήγηση.
- ❖ **New User Register** : Εφόσον κάποιος έχει συνδεθεί στο σύστημα με δικαιώματα διαχειριστή (administrator), μπορεί να δημιουργήσει νέους χρήστες, ορίζοντας για αυτούς τα username και password (όνομα χρήστη και κωδικός πρόσβασης) με τα οποία θα μπορούν να συνδεθούν.



Εικόνα 2.1 : Η προηγούμενη έκδοση της εφαρμογής Take A Note

## 2.4.2 Microsoft Reader

Λογισμικό ανάγνωσης και διαχείρισης ηλεκτρονικών βιβλίων, αναπτυγμένο από την Microsoft, σε έκδοση για Windows και Windows CE (για Pocket PC συσκευές). Κάνει χρήση αρχείων με επέκταση .lit. Αυτή τη στιγμή (31/1/07) βρίσκεται στην έκδοση 2.1.1 για Windows

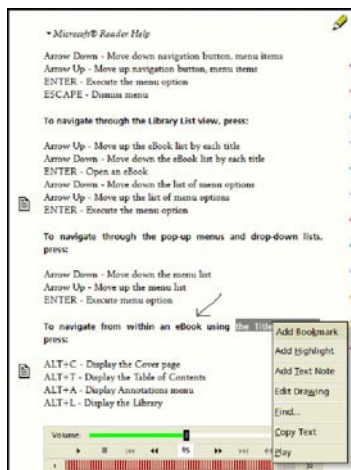
### Βασικές Λειτουργίες

- ❖ Add Highlight : Υπογράμμιση της επιλεγμένης λέξης ή και πρότασης με προεπιλεγμένο (default) χρώμα το κίτρινο. Εύκολα προσβάσιμη επιλογή από το αναδυόμενο (popup) menu, το οποίο εμφανίζεται είτε με απλό click πάνω στην επιθυμητή λέξη είτε με απλό μαρκάρισμα της πρότασης.
- ❖ Add Bookmark : Μαρκάρισμα της τρέχουσας σελίδας και εμφάνιση συμβόλου σελιδοδείκτη στην δεξιά μεριά της σελίδας με χρώμα αυτόματα επιλεγμένο από το σύστημα. Μειονέκτημα αποτελεί το γεγονός ότι ο χρήστης δεν μπορεί να επιλέξει κάποιο όνομα για τον σελιδοδείκτη για διευκόλυνση της πρόσβασης στην επιθυμητή σελίδα.
- ❖ Add Text Note : Μέσω αυτής της λειτουργίας ο χρήστης μπορεί να εισάγει τα προσωπικά του σχόλια στην σελίδα. Γίνεται συσχέτιση σχολίου και κειμένου με εισαγωγή χαρακτηριστικού συμβόλου δίπλα από την γραμμή που επιλέγει ο χρήστης.
- ❖ Add Drawing : Αντίστοιχο εργαλείο με το Pencil του MS Paint. Ουσιαστικά πρόκειται για ένα εικονικό μολύβι με το οποίο ο χρήστης, με χρήση του mouse , μπορεί να υπογραμμίσει επί του κειμένου, να σχεδιάσει κτλ.

### Επιπλέον λειτουργίες

Οι παρακάτω, ενδεικτικές, λειτουργίες δεν υπάρχουν στην standard έκδοση του προγράμματος. Πρόκειται για πακέτα, διαθέσιμα μέσω του site της Microsoft, τα οποία ο χρήστης πρέπει να κατεβάσει και να εγκαταστήσει.

- ❖ Text-to-Speech : Πρόκειται για λειτουργία με την οποία ο χρήστης έχει ουσιαστικά την δυνατότητα να ακούσει το κείμενο που βρίσκεται στη οθόνη του.
- ❖ Dictionaries : Λειτουργία η οποία προσθέτει λεξικολογικά σημεία αναφοράς στην ηλεκτρονική βιβλιοθήκη του χρήστη. Διαθέσιμη σε αρκετές γλώσσες.



Εικόνα 2.2 : Το πρόγραμμα Microsoft Reader

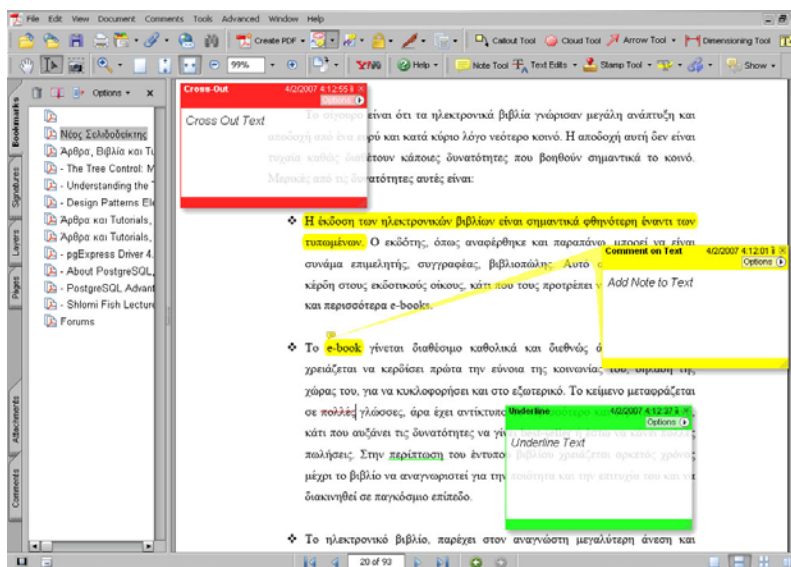
### 2.4.3 Adobe Reader

Πρόκειται για ίσως το δημοφιλέστερο αυτή τη στιγμή λογισμικό ανάγνωσης ηλεκτρονικών βιβλίων. Δέχεται αρχεία τύπου PDF (Portable Document Format) , format αναπτυγμένο από την ίδια την Adobe, και παρέχει πολλά εργαλεία που κάνουν την εμπειρία της ανάγνωσης ηλεκτρονικών βιβλίων πιο ελκυστική και εύκολη. Η σχετικά πρόσφατη έκδοση 6.0 του προγράμματος αντικατέστησε το μέχρι πρότινος επικρατέστερο λογισμικό στο χώρο Acrobat eBook Reader 2.2

#### Βασικές λειτουργίες

- ❖ Add Note to Text : Η Adobe έχει ενσωματώσει στο πρόγραμμα της μία από τις καλύτερες λειτουργίες εισαγωγής και διαχείρισης σχολίων. Ο χρήστης εισάγει στο ανάλογο παραθυράκι τα σχόλια που επιθυμεί και αυτομάτως εμφανίζεται πάνω στην σελίδα ένα εικονίδιο που αντιπροσωπεύει τα σχόλια. Με drag-and-drop μπορεί μεταφερθεί οπουδήποτε πάνω στην σελίδα, δίνοντας έτσι τη δυνατότητα πλήρους συσχέτισης κειμένου και σχολίου.
- ❖ Underline Text : Παρόμοιο εργαλείο με αυτό της εισαγωγής σχολίου, με τη διαφορά ότι όχι μόνο εισάγουμε σχόλιο αλλά και υπογραμμίζουμε την επιλεγμένη λέξη/πρόταση με κίτρινο χρώμα.
- ❖ Cross Out Text : Παραλλαγή του προηγούμενου. Δεν γίνεται υπογράμμιση αλλά strike through του κειμένου με γραμμή κόκκινου χρώματος.

- ❖ Add Bookmark : Κάνοντας δεξί click και επιλέγοντας το αντίστοιχο εργαλείο μπορούμε να προσθέσουμε την τρέχουσα σελίδα στον κατάλογο με τα Bookmarks (σελιδοδείκτες) για γρήγορη και εύκολη προσπέλαση της. Επίσης, μπορούμε με εύκολο τρόπο να αλλάξουμε το όνομα του σελιδοδείκτη καθώς επίσης και να τον μετακινήσουμε στο κατάλογο που εμφανίζεται αριστερά της σελίδας.
- ❖ Search : Μπορούμε να αναζητήσουμε μία λέξη ή φράση μέσα στο βιβλίο μας. Το πρόγραμμα μας επιστρέφει μία λίστα με τις προτάσεις όπου βρέθηκε η λέξη και με ένα απλό click μεταφερόμαστε στην αντίστοιχη σελίδα.
- ❖ Zoom In/Out : Ο χρήστης έχει την δυνατότητα να κάνει zoom in/zoom out επί του κειμένου. Χρήσιμο εργαλείο όταν σημεία του κειμένου δεν είναι αρκετά ευκρινή.
- ❖ Create Link : Δυνατότητα μετατροπής λέξης ή πρότασης σε HyperLink (Υπέρ-συνδέσμου) . Με click πάνω στην λέξη ο χρήστης μπορεί να μεταφερθεί σε κάποιο δικτυακό τόπο, σχετικού περιεχομένου.



Εικόνα 2.3 : Το πρόγραμμα Adobe Reader

## 2.5 Προτεινόμενες επεκτάσεις του συστήματος

Έχοντας αναλύσει τα εμπορικά πακέτα λογισμικού για ανάγνωση ηλεκτρονικών βιβλίων, την προηγούμενη έκδοση της εφαρμογής και λαμβάνοντας σοβαρά υπ' όψη τις παρατηρήσεις που έγιναν από την προηγούμενη ομάδα ανάπτυξης, καταστήσαμε μια λίστα με τις τροποποιήσεις και τις προσθήκες που θα επιχειρήσουμε :

- **Τροποποίηση της βάσης δεδομένων**  
Κρίθηκε σκόπιμη η αλλαγή της δομής της βάσης δεδομένων που χρησιμοποιεί η εφαρμογή. Η σχεδίαση της νέας βάσης δεδομένων θα περιλαμβάνει λιγότερους πίνακες με πολλές εγγραφές σε αντίθεση με την υπάρχουσα βάση (σε κάθε χρήστη αντιστοιχούν τρεις πίνακες, ένας για κάθε λειτουργία που παρέχει το πρόγραμμα).
- **Προσπέλαση των bookmarks και των άλλων εργαλείων από μενού**  
Για ευκολία του χρήστη αλλά και από άποψη «συμβατότητας» του layout της εφαρμογής μας με αυτό των περισσότερων συστημάτων, θεωρήθηκε χρήσιμη η ύπαρξη menu που περιέχουν, κατηγοριοποιημένες, τις διαθέσιμες λειτουργίες.
- **Δυνατότητα επιλογής του ηλεκτρονικού βιβλίου**  
Μια σημαντική προσθήκη στο παρόν σύστημα είναι η ανάγνωση και επεξεργασία πολλαπλών βιβλίων. Στην παρούσα έκδοση ο χρήστης περιορίζεται από την ύπαρξη ενός και μόνο βιβλίου ενώ αλλαγή βιβλίου μπορεί να γίνει μόνο προγραμματιστικά, από κάποιον που έχει γνώσεις αλλά και πρόσβαση στον πηγαίο κώδικα.
- **Επιλογή του χρώματος της υπογράμμισης**  
Η λειτουργία αυτή δίνει την δυνατότητα στους χρήστες να επιλέξουν διαφορετικά χρώματα υπογραμμιστών για διαφορετικής σημασίας χωρίων του βιβλίου τους.
- **Αλλαγή στο layout της εφαρμογής**
- **Δημιουργία ενός .exe πακέτου**  
Κατά την πρώτη επαφή μας με το υπάρχον σύστημα, παρατηρήσαμε την σχετικά αυξημένη δυσκολία της εγκατάστασης και παραμετροποίησης του. Κρίθηκε λοιπόν χρήσιμη η ύπαρξη ενός πακέτου εγκατάστασης, αυτοματοποιώντας έτσι διαδικασίες οι οποίες μπορεί και να προκαλέσουν την δυσφορία του χρήστη.

- **Αναζήτηση λέξεων ή φράσεων**

Ένα από τα βασικά πλεονεκτήματα του ηλεκτρονικού βιβλίου έναντι των συμβατικών είναι ο εύκολος εντοπισμός λέξεων ή φράσεων μέσα στο κείμενο. Η λειτουργία αυτή απουσιάζει από την παρούσα έκδοση του προγράμματος και θα προστεθεί στον νέο σύστημα.

- **Ασφάλεια χρήσης και ακεραιότητας δεδομένων των χρηστών**

Δεδομένου ότι η εφαρμογή προορίζεται και για διαδικτυακή χρήση, κρίθηκε σκόπιμο να δοθεί βαρύτητα στην ασφάλεια και προστασία των κρίσιμων δεδομένων του χρήστη από κακόβουλες επιθέσεις.

- **Εκτύπωση κειμένου**

- **Αποθήκευση της μορφοποίησης των σημειώσεων**

Προκειμένου η λειτουργία των σημειώσεων (notes) επί του κειμένου να προσομοιώνει καλύτερα την πραγματικότητα, θεωρήθηκε σκόπιμο να γίνει προσπάθεια αποθήκευσης της μορφοποίησης του κειμένου στη βάση για μετέπειτα ανάκτηση. Στην παρούσα έκδοση ο χρήστης μπορεί να μορφοποιήσει το κείμενο που πληκτρολογεί, αλλά δεν υπάρχει δυνατότητα αποθήκευσης της.

- **Ταχύτερη επικοινωνία με την βάση με χρήση του Connection Pool**

Το σύστημα κάνει εκτεταμένη χρήση της βάσης δεδομένων με πολλές προσπελάσεις σε αυτήν με συνέπεια τους υψηλούς χρόνους απόκρισης της εφαρμογής. Λύση στο συγκεκριμένο πρόβλημα μπορεί να δοθεί με την χρήση ενός "Connection Pool" (ανακυκλώσιμες συνδέσεις).



## ΚΕΦΑΛΑΙΟ 3

### ΑΝΑΛΥΣΗ ΛΕΙΤΟΥΡΓΙΩΝ ΣΥΣΤΗΜΑΤΟΣ ΑΝΑΓΝΩΣΗΣ – ΔΙΑΧΕΙΡΙΣΗΣ ΗΛΕΚΤΡΟΝΙΚΩΝ ΒΙΒΛΙΩΝ

#### 3.1 Προδιαγραφές συμβατών βιβλίων με την εφαρμογή

Τα χαρακτηριστικά που πρέπει να έχει ένα βιβλίο προκειμένου να είναι συμβατό με την εφαρμογή έχουν παραμείνει τα ίδια με αυτά που ίσχυαν και κατά την πρώτη έκδοση του προγράμματος. Θεωρήσαμε όμως αναγκαίο να αναφερθούν και στο παρών έγγραφο για διευκρινιστικούς λόγους, όπως περιγράφονται από τους Ζαφειρόπουλο και Γκίση στο έγγραφο «Ένα σύστημα προσαρμογής ηλεκτρονικού βιβλίου στις ανάγκες των χρηστών». Αυτά είναι :

1. Το βιβλίο πρέπει να είναι σχηματισμένο από html αρχεία.
2. Η οργάνωση των αρχείων πρέπει να είναι σε βάθος ενός επιπέδου. Για παράδειγμα στο βιβλίο “Java” όλα τα αρχεία που το αποτελούν θα πρέπει να βρίσκονται στον φάκελο “Java” και όχι σε κάποιο υποφάκελο.
3. Η κάθε σελίδα του βιβλίου αντιστοιχεί σε ένα αρχείο με μοναδικό όνομα. Χαρακτηριστικό ιδιαίτερα σημαντικό αφού όλη η εφαρμογή έχει δομηθεί πάνω σε αυτήν την παραδοχή.
4. Τα html αρχεία είναι συνδεδεμένα μεταξύ τους με την βοήθεια ενός αλγορίθμου πλοήγησης, ο οποίος κατ’ ελάχιστο έχει πλήκτρα – προηγούμενη σελίδα – επόμενη σελίδα – TOC (Table Of Contents - Περιεχόμενα)

## 3.2 Σύγκριση λειτουργιών τρέχουσας έκδοσης με την προηγούμενη

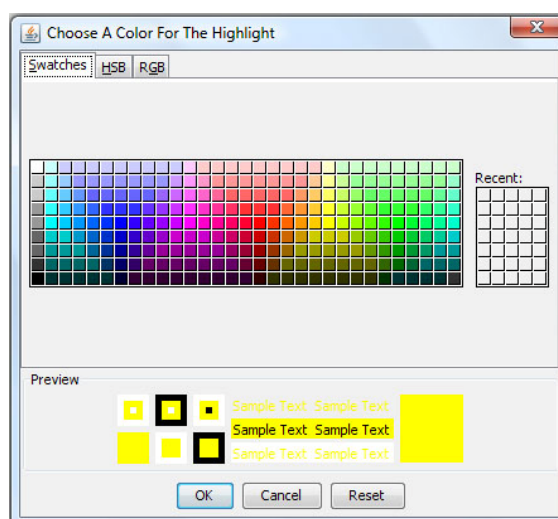
### 3.2.1 Λειτουργία υπογράμμισης (Highlight)

#### 3.2.1.1 Περιγραφή

Η λειτουργία της υπογράμμισης είναι η μόνη λειτουργία, που δίνει στον απλό χρήστη την δυνατότητα επέμβασης επί του κειμένου του ηλεκτρονικού βιβλίου. Ο χρήστης μπορεί σύμφωνα με την προσωπική του κρίση να επιλέξει τα χωρία εκείνα τα οποία χρήζουν ιδιαίτερης προσοχής και να τα μαρκάρει.

#### 3.2.1.2 Διαφορές με την πρώτη έκδοση

Στην αρχική έκδοση του συστήματος η υπογράμμιση λέξεων/φράσεων γινόταν με προεπιλεγμένο (default) χρώμα το κίτρινο. Στην παρούσα έκδοση, ο χρήστης καλείται να επιλέξει το χρώμα της αρεσκείας του. Η χρησιμότητα αυτής της προσθήκης έγκειται στο γεγονός ότι για διαφορετικής σημασίας χωρία μπορεί να επιλεγθεί και διαφορετικό χρώμα. Έτσι, αφενός διευκολύνεται η διαδικασία ανάγνωσης του βιβλίου και αφετέρου γίνεται καλύτερη προσομοίωση της πραγματικότητας, καθώς πολλοί είναι οι αναγνώστες έντυπων βιβλίων που επιλέγουν ποικίλα χρώματα για τις υπογραμμίσεις τους.



Εικόνα 3.1: Η παλέτα χρωμάτων από την οποία γίνεται η επιλογή του χρώματος υπογράμμισης

### 3.2.1.3 Υλοποίηση

Ο τρόπος υλοποίησης της λειτουργίας έχει μείνει κατά το μεγαλύτερο ποσοστό ο ίδιος με την αρχική έκδοση του συστήματος. Η κύρια διαφοροποίηση έγκειται στο γεγονός ότι ο χρήστης αφού επιλέξει το προς υπογράμμιση χωρίο, πρέπει να διαλέξει και το χρώμα που επιθυμεί από μία ευρεία γκάμα επιλογών η οποία του παρέχεται με χρήση του ColorChooser που προσφέρει η Java. Επιπλέον, στην βάση έχει προστεθεί και ένα νέο πεδίο τύπου integer με όνομα color, στο οποίο εισάγεται για κάθε υπογράμμιση το ακέραιο αντίστοιχο του χρώματος που επέλεξε ο χρήστης, ούτως ώστε όταν ξαναφορτωθεί η σελίδα, η συγκεκριμένη υπογράμμιση να έχει το σωστό χρώμα.

Στον κώδικα διαφορά υπάρχει κατά την δημιουργία της υπογράμμισης. Στην παλιά έκδοση η προσθήκη ενός highlight γινόταν με την χρήση των δεικτών αρχής και τέλους της υπογράμμισης και του κίτρινου χρώματος :

```
ref = hilite.addHighlight(tempbeg, tempend, h);
```

όπου h είναι ένα αντικείμενο τύπου MyHighlightPainter με προκαθορισμένο το κίτρινο χρώμα σαν παράμετρο. Στην νέα έκδοση, η προσθήκη γίνεται με τον ίδιο τρόπο με τη διαφορά ότι αυτή τη φορά το προκαθορισμένο χρώμα έχει αντικατασταθεί με μια μεταβλητή τύπου Color(), της οποίας η τιμή τροφοδοτείται μέσω ενός αντικειμένου τύπου JColorChooser. Για να γίνει πιο κατανοητό ακολουθούν επιλεκτικά κάποια κομμάτια από τον κώδικα :

```
JColorChooser cch = new JColorChooser();
color = cch.showDialog(ViewClass.frame,"Choose A Color For The
                                Highlight",Color.yellow);
ref = hilite.addHighlight(tempbeg, tempend, new MyHighlightPainter(color));
```

Μόλις ο χρήστης επιλέξει κάποιο χρώμα, η μεταβλητή color παίρνει την αντίστοιχη τιμή του χρώματος που προτίμησε ο χρήστης.

### 3.2.2 Λειτουργία σελιδοδεικτών

Η λειτουργία των σελιδοδεικτών δεν έχει καμία διαφοροποίηση με την προηγούμενη έκδοση. Για περισσότερες πληροφορίες ανατρέξτε στο έγγραφο της προηγούμενης πτυχιακής *«Ένα σύστημα προσαρμογής ηλεκτρονικού βιβλίου στις ανάγκες των χρηστών»*, γραμμένο από τους Γκίση Τραϊανό και Ζαφειρόπουλο Ιωάννη υπό της επίβλεψη του κ. Σταμάτη Δημοσθένη.

### 3.2.3 Λειτουργία προσωπικών σημειώσεων

#### 3.2.3.1 Περιγραφή

Δεν είναι λίγες οι φορές που κατά την διάρκεια ανάγνωσης κάποιου έντυπου βιβλίου, έχουμε κρατήσει σημειώσεις πάνω σε δυσνόητα σημεία του κειμένου, ούτως ώστε κατά την επόμενη ανάγνωση να βοηθηθούμε στην κατανόηση των σημείων αυτών. Αυτό είναι και το χαρακτηριστικό, το οποίο προσπαθούμε να προσομοιώσουμε με το να παρέχεται στον χρήστη ένας χώρος, όπου θα μπορεί να κρατήσει σημειώσεις. Η λειτουργία αυτή υποστηριζόταν και στην πρώτη έκδοση της εφαρμογής επιτυχώς. Ο χρήστης πληκτρολογούσε το κείμενο της αρεσκείας του, αποθηκευόταν στη βάση και κάθε φορά που ο συγκεκριμένος χρήστης φόρτωνε την σελίδα, εμφανιζόταν και το κείμενο που είχε εισάγει. Επιπλέον, είχε και την δυνατότητα μορφοποίησης του κειμένου, με το να μπορεί να αλλάξει την γραμματοσειρά, το μέγεθος του κειμένου και το χρώμα του. Το μόνο αρνητικό ήταν πως ό,τι αλλαγές και αν έκανε στο κείμενο, αυτές δεν αποθηκευόντουσαν εξαιτίας μιας ιδιομορφίας της βάσης που έχει χρησιμοποιηθεί. Στην παρούσα έκδοση του συστήματος έγινε προσπάθεια να ξεπεραστεί το συγκεκριμένο πρόβλημα, δίχως να χρειαστεί να αλλαχθεί το DBMS, με επιτυχία. Πλέον, ό,τι μορφοποίηση και αν γίνει στο κείμενο, αυτή θα αποθηκευτεί στην βάση και οι σημειώσεις θα ανακτηθούν αυτούσιες.

#### 3.2.3.2 Υλοποίηση

Για την αποθήκευση του κειμένου μαζί την μορφοποίηση επιλέχθηκε η λύση του serialization ενός αντικειμένου. Object serialization (σειριοποίηση ενός αντικειμένου), είναι η διαδικασία κατά την οποία η κατάσταση ενός αντικειμένου σώζεται ως μια ακολουθία από bytes, καθώς επίσης και η διαδικασία της ανακατασκευής αυτών των byte σε αντικείμενο κάποια στιγμή στο μέλλον. Η διαδικασία αυτή υποστηρίζεται πλήρως από το Java Serialization API (Application Programming Interface), το οποίο παρέχει τις απαραίτητες μεθόδους και κλάσεις για τον σκοπό αυτό. Η μεθοδολογία που ακολουθείται είναι η εξής :

Ό,τι βλέπουμε στο πεδίο εισαγωγής κειμένου (αντικείμενο JTextPane), στην Java κρατάτε σε μία μεταβλητή τύπου StyledDocument. Το περιεχόμενο της μεταβλητής αυτής είναι και αυτό που πρέπει να μετατραπεί σε σειρά από bytes και να αποθηκευτεί στην βάση. Αυτό επιτυγχάνεται με το να την μετατρέψουμε σε εξερχόμενη ροή δεδομένων (object output stream) με την

βοήθεια μιας μεταβλητής τύπου `ByteArrayOutputStream`. Το κομμάτι κώδικα με το οποίο γίνεται η συγκεκριμένη διαδικασία είναι το εξής :

```
ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
ObjectOutputStream out = new ObjectOutputStream(outputStream);
out.writeObject(doca);
```

όπου `doca` είναι η μεταβλητή τύπου `StyledDocument`.

Το μόνο που απομένει μετά από αυτό είναι να αποθηκευτεί αυτή η ροή στην βάση δεδομένων. Ο τρόπος με τον οποίο γίνεται αυτό θα περιγραφεί λίγο παρακάτω.

Για την ανάκτηση τώρα των bytes γίνεται η ακριβώς αντίθετη διαδικασία. Μέσα από την μέθοδο `LoadNotes()` της κλάσης `Notes`, ανακτώνται από την βάση τα bytes, και μετατρέπονται σε εισερχόμενη ροή δεδομένων (`object input stream`), με την βοήθεια μεταβλητής τύπου `ByteArrayInputStream` :

```
ByteArrayInputStream inputStream = new ByteArrayInputStream(matrix);
ObjectInputStream objectInput = new ObjectInputStream(inputStream);
doca = (StyledDocument)objectInput.readObject();
```

όπου `matrix` είναι ο πίνακας με τα bytes που μας επέστρεψε το αντίστοιχο query που έγινε στην βάση.

Μετά από αυτό, δεν έχουμε να κάνουμε τίποτα άλλο από το να ορίσουμε ως `StyledDocument` αυτό που ανακτήσαμε :

```
textPane.setStyleedDocument(doca);
```

### 3.2.3.3 Τρόπος αποθήκευσης των σημειώσεων

Το πιο δύσκολο μέρος αυτής της διαδικασίας ήταν να αποθηκευτούν επιτυχώς τα bytes στην βάση μας. Αρχικά έπρεπε να αποφασιστεί ποιος τύπος BLOB (Binary Large Object) δεδομένων θα χρησιμοποιούνταν για το αντίστοιχο πεδίο της βάσης. Οι επιλογές που προσφέρει η PostgreSQL είναι δυο. Είτε ως `bytea` είτε ως `oid` (Large Object).

| Characteristic     | BYTEA                            | OID  |
|--------------------|----------------------------------|--|
| Max. allowed space | 1 GB                             | 2 GB   |
| Data access        | As a whole                       | Stream-style                                       |
| Storage            | In defined table                 | In pg_largeobject system table                     |
| Data manipulation  | Using SQL and escaping sequences | Only within transaction block by special functions |
| Loading            | Preload                          | On demand  |

Βλέποντας το παραπάνω σχήμα, μπορούμε να συμπεράνουμε ότι ο oid τύπος είναι ανώτερος από τον bytea. Το κύριο μειονέκτημα όμως των oid, είναι ότι όλα τα δεδομένα αποθηκεύονται σε ένα πίνακα του συστήματος, τον pg\_largeobject, και στο πεδίο του δικού μας πίνακα – εν προκειμένω ο πίνακας notes – αποθηκεύεται μονάχα μία αναφορά προς τον pg\_largeobject. Αυτό σημαίνει ότι δεν έχουμε πλήρη έλεγχο πάνω στα δεδομένα μας. Και να διαγράψουμε την αναφορά από τον πίνακα notes, τα πραγματικά δεδομένα θα παραμείνουν στον πίνακα του συστήματος. Επί προσθέτως, μέχρι στιγμής, δεν έχει υλοποιηθεί, από την πλευρά της Postgres, τρόπος με τον οποίο να μπορεί να διαγραφεί ή να μεταβληθεί κάποια εγγραφή του εν λόγω πίνακα. Με αυτό ως δεδομένο και έχοντας στο μυαλό μας το γεγονός ότι κάθε φορά κατά την ανάγνωση ενός βιβλίου, για κάθε σελίδα και για κάθε χρήστη αποθηκεύονται και οι αντίστοιχες σημειώσεις, ερχόμαστε στο συμπέρασμα πως μετά από λίγο καιρό χρήσης του προγράμματος από ένα αριθμό χρηστών, ο όγκος των δεδομένων που κρατάει ο pg\_largeobject θα είναι τεράστιος, με ποιο πιθανό αποτέλεσμα η απόδοση του προγράμματος να πέσει. Για αυτόν τον λόγο, επιλέχθηκε τελικά ο τύπος bytea.

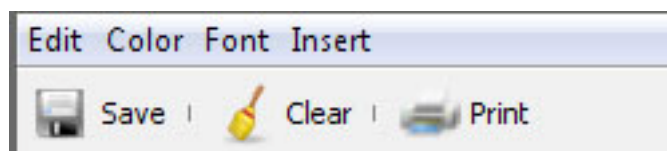
Από εκεί και πέρα τα πράγματα ήταν απλά. Η εισαγωγή των δεδομένων γίνεται κανονικά στον πίνακα μας, με μόνο άξιο μνείας σημείο το γεγονός ότι η εισαγωγή έπρεπε απαραίτητως να γίνει με χρήση Prepared Statements

```
update = con.prepareStatement("UPDATE NOTES SET USERNOTES = ? WHERE
USERNAME= ? AND FILE= ? AND BOOK= ?");
update.setBytes(1, outputStream.toByteArray());
.
.
.
update.executeUpdate();
```

### 3.2.3.4 Επιπλέον χαρακτηριστικά

Για καλύτερη εκμετάλλευση του νέου χαρακτηριστικού, αυτού της διατήρησης της μορφοποίησης των σημειώσεων, έγιναν και κάποιες αλλαγές και στο μενού του panel των notes. Στην ανανεωμένη έκδοση του συστήματος, το παλιό μενού με όνομα Style διασπάστηκε και εμπλουτίστηκε,

για να δημιουργηθούν δυο καινούρια. Το πρώτο φέρει το όνομα Color, και μέσω αυτού μπορεί κάποιος να επιλέξει το χρώμα του κειμένου. Αυτό μπορεί να γίνει είτε διαλέγοντας ένα από τα έξη προκαθορισμένα χρώματα που εμφανίζονται είτε επιλέγοντας όποιο άλλο επιθυμεί μέσω του ColorChooser panel που προσφέρει η Java, κάνοντας κλικ στην επιλογή "Choose a color". Το δεύτερο μενού ονομάζεται Font, και μέσω αυτού μπορούμε να προσπελάσουμε χαρακτηριστικά, όπως είναι η γραμματοσειρά, το μέγεθος των γραμμάτων και το αν το κείμενο θα είναι Bold, Underline ή Italic. Τέλος, ένα άλλο γνώρισμα των ανανεωμένων σημειώσεων, είναι και αυτό της δυνατότητας εισαγωγής εικόνας. Μέσω του τέταρτου μενού που υπάρχει, με όνομα Insert, και της επιλογής Image File, μπορεί κανείς να περιηγηθεί στα αρχεία του υπολογιστή του, να επιλέξει κάποια εικόνα και να την εισάγει στην περιοχή των σημειώσεων του.



Εικόνα 3.2 : Το νέο μενού των σημειώσεων. Στα δεξιά διακρίνεται το μενού Insert μέσω του οποίου ο χρήστης μπορεί να εισάγει μια εικόνα

### 3.2.3.5 Εκτύπωση σημειώσεων

Επιλέγοντας το αντίστοιχο κουμπί εκτύπωσης, ο χρήστης έχει την δυνατότητα να εκτυπώσει τις σημειώσεις του. Κάνοντας τις απαραίτητες ενέργειες, το κείμενο θα σταλεί στον επιλεγμένο εκτυπωτή και θα εκτυπωθεί αυτούσιο, διατηρώντας την όποια μορφοποίηση καθώς επίσης και τις εικόνες που έχουν εισαχθεί. Η υλοποίηση είναι παρόμοια με αυτήν της λειτουργίας εκτύπωσης του κειμένου (βλ. ενότητα Λειτουργία εκτύπωσης κειμένου).

## 3.3 Νέες λειτουργίες

### 3.3.1 Λειτουργία ανοίγματος βιβλίου

#### 3.3.1.1 Περιγραφή

Από την στιγμή που υποστηρίζεται η ύπαρξη πολλαπλών βιβλίων, είναι απαραίτητη και η ύπαρξη λειτουργίας μέσω της οποίας κάποιος θα μπορεί να ανοίγει τα υπάρχοντα βιβλία.

### 3.3.1.2 Υλοποίηση

Το άνοιγμα ενός βιβλίου γίνεται μέσω μια λίστας. Εκτελείται ένα απλό ερώτημα στην βάση δεδομένων και επιστρέφονται δυναμικά, τα ονόματα και οι σχετικές διευθύνσεις όλων των βιβλίων που έχουν καταχωρηθεί. Στη λίστα την οποία και βλέπει ο χρήστης, εμφανίζονται οι ονομασίες των βιβλίων, όπως αυτές έχουν καταχωρηθεί στο πεδίο bookname του πίνακα books. Επιλέγοντας κάποιο βιβλίο, φορτώνεται η αντίστοιχη σελίδα μέσω της σχετικής διεύθυνσης (πεδίο url), καθώς επίσης και οι σελιδοδείκτες, οι υπογραμμίσεις και οι σημειώσεις του χρήστη που έχει κάνει στο συγκεκριμένο βιβλίο. Η συσχέτιση μεταξύ του ονόματος και της διεύθυνσης μέσα στον κώδικα γίνεται με την βοήθεια δυο διανυσμάτων (Vectors).

```
result = stm.executeQuery("SELECT bookname,url FROM books;");

while(result.next()){
    books.addElement(result.getString(1));
    book_url.addElement(result.getString(2));
}
```

Έχοντας και τα δυο διανύσματα την ίδια αρχική χωρητικότητα και το ίδιο βήμα αύξησης χωρητικότητας, κάθε στιγμή έχουν την σωστή αντιστοιχία μεταξύ ονόματος και διεύθυνσης. Έτσι για παράδειγμα, όταν το διάνυσμα books έχει στην δεύτερη του θέση το τρίτο κατά σειρά βιβλίο, το διάνυσμα book\_url θα έχει την σχετική διεύθυνση αυτού.

### 3.3.1.3 Προβλήματα που αντιμετωπίστηκαν κατά την υλοποίηση

1. Όταν άνοιγε ένας χρήστης κάποιο βιβλίο, παρόλο που φορτωνόντουσαν οι σωστές σημειώσεις, παρέμεναν θεατές και αυτές του προηγούμενου βιβλίου με αποτέλεσμα να επικαλύπτονται. Το ίδιο γινόταν και με τις υπογραμμίσεις και τους σελιδοδείκτες. Το πρόβλημα λύθηκε με το να αποθηκεύονται και μετά να αφαιρούνται από την όψη αυτόματα οι παλιές σημειώσεις, υπογραμμίσεις και σελιδοδείκτες πρώτου ανοιχθεί το νέο βιβλίο.
2. Παρόμοιο πρόβλημα παρουσιαζόταν και σε περίπτωση που πριν ανοιχτεί το βιβλίο ο χρήστης είχε εκτελέσει κάποια αναζήτηση. Αφού άνοιγε το βιβλίο τα αποτελέσματα παρέμεναν ορατά, κάτι το οποίο και εμπόδιζε τη σωστή λειτουργία του προγράμματος. Λύθηκε με τον καθαρισμό των αποτελεσμάτων πριν την φόρτωση της νέας σελίδας.



### 3.3.2 Λειτουργία εγκατάστασης βιβλίου

#### 3.3.2.1 Περιγραφή

Το πρόγραμμα δίνει την δυνατότητα στους διαχειριστές να εισάγουν νέα βιβλία (βλ. ενότητα Δημιουργία νέου βιβλίου). Σε περίπτωση που ο χρήστης δουλεύει τοπικά το πρόγραμμα, παρόλο που κάποιο βιβλίο μπορεί να εμφανίζεται στη λίστα με τα διαθέσιμα βιβλία (βλ. προηγούμενη ενότητα) ενδεχομένως να μην το έχει διαθέσιμο στον τοπικό του δίσκο. Για να λυθεί αυτή η ασυμφωνία, αποφασίσαμε να συμπεριλάβουμε και μια λειτουργία με την οποία ο απλός χρήστης θα μπορεί εύκολα να εγκαθιστά τα νέα βιβλία που δημιουργούν οι διαχειριστές, στον τοπικό του υπολογιστή. Απαραίτητη προϋπόθεση για να δουλέψει σωστά η συγκεκριμένη λειτουργία είναι ο διαχειριστής που δημιούργησε το βιβλίο, να διαθέσει το zip αρχείο που περιέχει το βιβλίο, στους απλούς χρήστες, είτε μέσω διαδικτύου είτε με οποιονδήποτε άλλο τρόπο.

#### 3.3.2.2 Υλοποίηση

Αυτό που ουσιαστικά κάνει η εν λόγω λειτουργία είναι να αποσυμπιέζει το αρχείο που περιέχει το βιβλίο στον σωστό φάκελο, ώστε ο χρήστης να μπορεί να το ανοίξει με το πρόγραμμά μας.

Αφού κάποιος διαχειριστής έχει δημιουργήσει το νέο βιβλίο, στον πίνακα books της βάσης δεδομένων θα υπάρχει και η σωστή αντίστοιχη καταχώρηση, που περιλαμβάνει το όνομα του βιβλίου και την σχετική διεύθυνση της πρώτης του σελίδας. Ο χρήστης, μέσω της λειτουργίας εγκατάστασης, καλείται να επιλέξει το όνομα του βιβλίου που θέλει να εγκαταστήσει μέσω ενός drop down menu, το οποίο γεμίζει δυναμικά με ένα ερώτημα στην βάση το οποίο επιστρέφει τα καταχωρημένα ονόματα των βιβλίων. Αφού τα στοιχεία του προς εγκατάσταση βιβλίου επιβεβαιωθούν, το πρόγραμμα δημιουργεί τον απαραίτητο φάκελο στον οποίο θα αντιγραφούν τα αρχεία και του δίνει για όνομα το όνομα του βιβλίου παραλείποντας τα κενά

```
String path = bookName.getSelectedItem().toString().replaceAll(" ", "") + "/";
File folder = new File(FOLDER + path);
folder.mkdir();
```

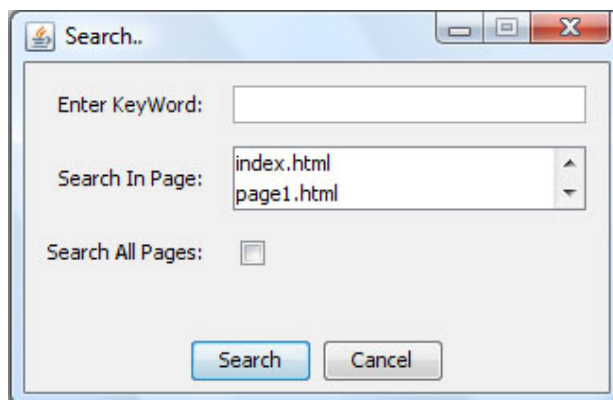
Να επισημανθεί εδώ ότι η διεύθυνση μέσα στην οποία πρέπει να αντιγραφεί το βιβλίο για να λειτουργήσει είναι η /build/classes, που βρίσκεται κάτω από τον κεντρικό φάκελο εγκατάστασης του προγράμματος. Αφού δημιουργηθεί

και ο φάκελος, τα πάντα είναι έτοιμα για την αποσυμπίεση του αρχείου. Καλώντας μέσα στον κώδικα την μέθοδο `unzipFile()` (βρίσκεται στην κλάση `InstallBook`), διαβάζετε το zip αρχείο, δημιουργούνται οι όποιοι υποφάκελοι υπάρχουν και τέλος αποσυμπιέζεται το αρχείο στην διεύθυνση που προαναφέρθηκε.

### 3.3.3 Λειτουργία αναζήτησης

#### 3.3.3.1 Περιγραφή

Ένα από τα μεγαλύτερα πλεονεκτήματα της ηλεκτρονικής μορφής βιβλίων έναντι των έντυπων είναι η δυνατότητα που δίνουν στο χρήστη για πολύ εύκολο εντοπισμό μιας λέξης ή φράσης εντός του κειμένου και κατ' επέκταση η εύκολη εύρεση της ενότητας που τους ενδιαφέρει κάθε φορά. Ο χρήστης, με το πάτημα ενός κουμπιού, μπορεί γρήγορα να εντοπίσει την λέξη/φράση που επιθυμεί, όσο μεγάλο και αν είναι το βιβλίο που διαβάσει. Αυτή ακριβώς την λειτουργία προσπαθήσαμε να υλοποιήσουμε και στο δικό μας σύστημα ανάγνωσης ηλεκτρονικών βιβλίων, εκμεταλλευόμενοι έτσι την τεχνολογία με στόχο την διευκόλυνση του αναγνώστη .



Εικόνα 3.3 : Το παράθυρο μέσα από το οποίο ο χρήστης μπορεί να εκτελέσει την αναζήτηση του

#### 3.3.3.2 Υλοποίηση

Η λειτουργία της αναζήτησης, υλοποιείται στο πρόγραμμα μας, ουσιαστικά με δυο μεθόδους. Και οι δυο ονομάζονται `search` (και βρίσκονται μέσα στην κλάση `Search`). Η μία δέχεται ως παραμέτρους ένα πίνακα από `Strings`, ο οποίος περιέχει τα ονόματα των σελίδων όπου πρόκειται να γίνει η αναζήτηση, και μια μεταβλητή επίσης τύπου `String`, η οποία περιέχει την προς αναζήτηση λέξη/φράση. Η δεύτερη μέθοδος, καλείται με την χρήση δυο απλών `String`. Το πρώτο θα περιέχει το όνομα της σελίδας και το δεύτερο την

λέξη/φράση για την οποία ο χρήστης θέλει να ψάξει. Όπως ίσως έγινε ήδη κατανοητό, η πρώτη χρησιμοποιείται για να γίνει αναζήτηση σε ολόκληρο το βιβλίο που είναι ανοικτό και η δεύτερη για να γίνει αναζήτηση σε μία μόνο σελίδα, την οποία και επιλέγει ο χρήστης.

Αναλυτικότερα η αρχική λειτουργία και των δυο μεθόδων είναι να διαβάσουν το html αρχείο/αρχεία στο οποίο πρόκειται να εκτελεστεί η αναζήτηση, αποθηκεύοντας σε μια μεταβλητή τύπου String (με όνομα στον κώδικα docText) το κείμενο του αρχείου, δίχως όμως να πάρουν κανένα από τα όποια tags της html. Το μόνο που διαφοροποιεί μέχρι τώρα τις δύο μεθόδους, είναι η ύπαρξη ενός for(...) loop στη μέθοδο αναζήτησης σε ολόκληρο το βιβλίο, το οποίο χρησιμεύει στο να διαβαστούν όλες οι σελίδες. Αφού διαβαστεί με επιτυχία το κείμενο και αποθηκευτεί στην μεταβλητή, μπορούμε να ξεκινήσουμε την αναζήτηση. Με την βοήθεια ενός for(...) loop το οποίο διαβάζει τα περιεχόμενα της docText μεταβλητής, ελέγχουμε εάν η λέξη που εισήγαγε ο χρήστης εμφανίζεται στο κείμενο, και αν ναι, σε ποια θέση του String βρίσκεται – επιτυγχάνεται απλά με τη χρήση της μεθόδου indexOf().

```
if(docText.indexOf(word) != -1)
```

Εάν αληθεύει ο παραπάνω έλεγχος, δημιουργείται ένα νέο String που αποτελεί υποδιαίρεση του αρχικού. Το νέο String ξεκινάει από την θέση (index) του πρώτου γράμματος της λέξης και τελειώνει στο index όπου βρίσκεται ο εικοστός επόμενος χαρακτήρας. Με λίγα λόγια, καταλήγουμε να έχουμε ένα String, που περιέχει την λέξη και είκοσι επιπλέον χαρακτήρες μέσα από το κείμενο. Όλο αυτό το προσθέτουμε σε ένα διάνυσμα (Vector), το οποίο στο τέλος θα περιέχει όλα τα αποτελέσματα.

```
vect.add(j,docText.substring(docText.indexOf(word),docText.indexOf(word)+20));
```

Για να συνεχιστεί η αναζήτηση ομαλά, πρέπει το μέρος του κειμένου που έχει ψάξει μέχρι τώρα ο αλγόριθμος μας, να αφαιρεθεί από το αρχικό. Γι' αυτό δημιουργούμε άλλη μια υποδιαίρεση του κειμένου, το οποίο αυτή τη φορά ξεκινάει αμέσως μετά από την θέση που βρήκαμε την λέξη.

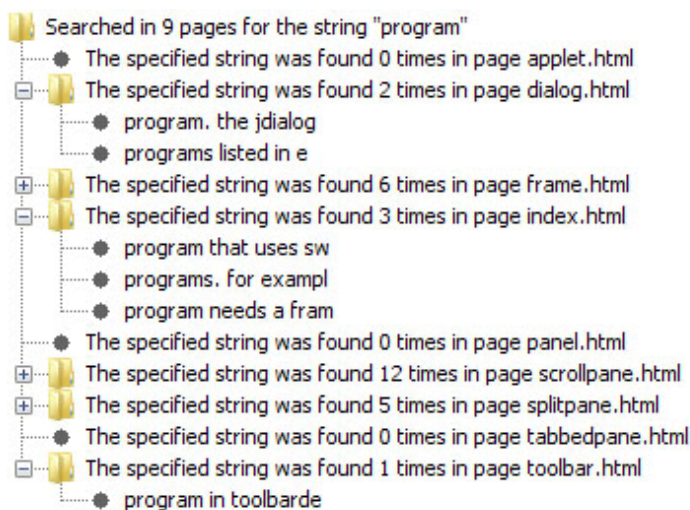
```
docText = docText.substring(docText.indexOf(word)+ word.length());
```

Όλη η παραπάνω διαδικασία επαναλαμβάνεται μέχρι να διαβαστεί όλο το κείμενο. Τελευταία ενέργεια που γίνεται, είναι να δημιουργηθεί το δέντρο με τα αποτελέσματα, καθώς επίσης και οι σύνδεσμοι (links), τους οποίους όταν τους επιλέγει ο χρήστης, θα μεταφέρεται στο σωστό σημείο, της σωστής σελίδας.

### 3.3.3.3 Δημιουργία δέντρου και συνδέσμων

Το δέντρο και οι σύνδεσμοι, δημιουργούνται με την βοήθεια δύο κλάσεων, των SearchInfo και SearchTreeController. Η κλάση SearchTreeController, η οποία και υλοποιεί τον TreeSelectionListener, είναι υπεύθυνη για την δημιουργία του link με την σωστή κάθε φορά σελίδα και για την υπογράμμιση της λέξης που βρέθηκε. Κάνοντας κλικ ο χρήστης σε ένα κόμβο του δέντρου, δημιουργείται από την κλάση ένα αντικείμενο τύπου SearchInfo, του οποίου τα γνωρίσματα είναι η λέξη που έχει βρεθεί, το όνομα της σελίδας στην οποία βρέθηκε η συγκεκριμένη λέξη και ο δείκτης θέσης της λέξης μέσα στο κείμενο. Αυτομάτως, φορτώνεται η αντίστοιχη σελίδα και υπογραμμίζεται με γκρι χρώμα η λέξη και οι επόμενοι είκοσι χαρακτήρες μέσα στο κείμενο. Το δέντρο με όλα τα αποτελέσματα το οποίο εμφανίζεται, δημιουργείται με την βοήθεια ενός DefaultMutableTreeNode παίρνοντας ως όρισμα ένα αντικείμενο τύπου SearchInfo.

Τέλος, πρέπει να σημειωθεί ότι η αναζήτηση δεν είναι case sensitive.



Εικόνα 3.4 : Το δέντρο με τα αποτελέσματα μιας αναζήτησης. Κάνοντας κλικ σε ένα κόμβο μεταφερόμαστε στην αντίστοιχη σελίδα

### 3.3.3.4 Προβλήματα που αντιμετωπίστηκαν κατά την υλοποίηση της αναζήτησης

1) Εάν ο χρήστης αναζητήσει μια λέξη η οποία βρίσκεται στο τέλος του κειμένου, η προσπάθεια διαίρεσης του κειμένου

```
docText = docText.substring(docText.indexOf(word)+ word.length());
```

δεν είναι δυνατή, και το πρόγραμμα πετούσε εξαίρεση (exception), εμποδίζοντας την περαιτέρω σωστή λειτουργία του προγράμματος. Αντιμετωπίστηκε πιάνοντας αυτήν την εξαίρεση (μπλοκ try - catch) και ορίζοντας την διακοπή του loop με την εντολή break.

2) Παρόμοιο πρόβλημα για τον ίδιο λόγο, αντιμετωπίστηκε και κατά την προσπάθεια εισαγωγής της λέξης που βρέθηκε μαζί με τους είκοσι επόμενους χαρακτήρες στο διάνυσμα. Στην περίπτωση που η λέξη βρίσκεται στο τέλος του κειμένου η γραμμή κώδικα

```
vect.add(j,docText.substring(docText.indexOf(word),docText.indexOf(word)+20));
```

κατέληγε σε Exception. Αντιμετωπίστηκε με παρεμφερή τρόπο, πιάνοντας την εξαίρεση και παραλείποντας αυτή τη φορά την προσπάθεια εισαγωγής των επόμενων χαρακτήρων

```
try{
    vect.add(j,docText.substring(docText.indexOf(word),docText.indexOf(word)+20));
}catch(StringIndexOutOfBoundsException out){
    vect.add(j,docText.substring(docText.indexOf(word)));
}
```

3) Εάν ο χρήστης επιχειρούσε αναζήτηση δίχως να έχει πρώτα καθαρίσει τα αποτελέσματα κάποιας προηγούμενης, δημιουργούνταν προβλήματα στην εμφάνιση των αποτελεσμάτων. Για την ακρίβεια τα νέα αποτελέσματα δεν εμφανιζόντουσαν καθόλου και προστίθονταν νέα scrollPanels. Για να αντιμετωπιστεί, ορίστηκε να γίνεται καθαρισμός των αποτελεσμάτων με το που πατιέται το κουμπί της αναζήτησης.

### 3.3.4 Λειτουργία αλλαγής της εμφάνισης της εφαρμογής

#### 3.3.4.1 Περιγραφή

Ελέγχοντας το πρόγραμμα σε διαφορετικούς υπολογιστές, και χρησιμοποιώντας μία συγκεκριμένη εμφάνιση(look and feel) για την εφαρμογή μας (αυτή που αντιστοιχεί στην εμφάνιση του συστήματος - SystemLookAndFeel), παρατηρήθηκε ότι αναλόγως το θέμα του λειτουργικού του κάθε υπολογιστή, η εμφάνιση του προγράμματος μας μεταβαλλόταν σε μεγάλο βαθμό. Το γεγονός αυτό, καθιστούσε την εφαρμογή σε ορισμένα συστήματα καθόλου ελκυστική προς τον χρήστη. Για τον λόγο αυτό, και για να μην προκληθεί η δυσφορία του απλού χρήστη, αποφασίσαμε να συμπεριλάβουμε λειτουργία, μέσω της οποίας ο καθένας, θα μπορεί επιλέξει

μέσα από ένα εύρος εναλλακτικών επιλογών, την εμφάνιση που ταιριάζει στα δικά του κριτήρια.

### 3.3.4.2 Υλοποίηση

Η επιλογή γίνεται μεταξύ πέντε προκαθορισμένων τιμών που αφορούν την εμφάνιση της εφαρμογής και μεταξύ δύο τιμών που αφορούν το θέμα μιας συγκεκριμένης εμφάνισης (της Cross Platform Look and Feel). Οι τιμές της εμφάνισης στον κώδικα έχουν αποθηκευτεί σε ένα δισδιάστατο πίνακα 5x2. Η πρώτη στήλη αντιστοιχεί στο όνομα της εμφάνισης, το οποίο και θα βλέπει ο χρήστης μέσω του drop down menu, και η δεύτερη στην ονομασία του πακέτου της Java, το οποίο και περιλαμβάνει την συγκεκριμένη εμφάνιση. Επιλέγοντας μια τιμή από το drop down menu, η τιμή πακέτου που αντιστοιχεί στο επιλεγμένο όνομα αποθηκεύεται στην μεταβλητή lookAndFeel ώστε να φορτωθεί και η ανάλογη εμφάνιση. Σε περίπτωση που επιλεγθεί η Cross Platform Look And Feel εμφάνιση, ενεργοποιείται και ένα δεύτερο drop down menu, απ' όπου ο χρήστης μπορεί να διαλέξει μεταξύ δύο τιμών θέματος, των Default Theme και Ocean Theme. Κάνοντας κλικ πάνω στο κουμπί Ok, αφενός γίνεται η αλλαγή της εμφάνισης και αφετέρου αποθηκεύεται στον πίνακα των users η επιλογή του χρήστη, ούτως ώστε την επόμενη φορά που θα εκτελέσει το πρόγραμμα, να φορτωθεί η εμφάνιση της επιλογής του.

| Ονομασία                     | Πακέτο   |
|------------------------------|--|
| GTK Look And Feel            | com.sun.java.swing.plaf.gtk.GTKLookAndFeel         |
| Motif Look And Feel          | com.sun.java.swing.plaf.motif.MotifLookAndFeel     |
| Windows Look And Feel        | com.sun.java.swing.plaf.windows.WindowsLookAndFeel |
| Cross Platform Look And Feel | javax.swing.plaf.metal.MetalLookAndFeel            |
| System Look And Feel         | UIManager.getSystemLookAndFeelClassName()          |

Το κομμάτι του κώδικα που είναι υπεύθυνο για την αλλαγή της εμφάνισης και του θέματος είναι το παρακάτω :

```

UIManager.setLookAndFeel(lookAndFeel);
if(lookAndFeel.equals("javax.swing.plaf.metal.MetalLookAndFeel")){
    if (theme.equals("Default Theme"))
        MetalLookAndFeel.setCurrentTheme(new DefaultMetalTheme());
    else if (theme.equals("Ocean Theme"))
        MetalLookAndFeel.setCurrentTheme(new OceanTheme());
UIManager.setLookAndFeel(new MetalLookAndFeel());

SwingUtilities.updateComponentTreeUI(ViewClass.frame);
ViewClass.frame.pack();

```

### 3.3.5 Λειτουργία εκτύπωσης κειμένου

#### 3.3.5.1 Περιγραφή

Κάποια στιγμή, κάποιος από τους χρήστες μπορεί να χρειαστεί ένα από τα βιβλία που διαβάζει ή απλά ένα μέρος αυτών και σε έντυπη μορφή, είτε γιατί μπορεί να μην έχει εύκολη πρόσβαση σε ηλεκτρονικό υπολογιστή είτε για οποιονδήποτε άλλο λόγο. Γι' αυτό, για την διευκόλυνση του, αποφασίστηκε να του παρέχεται η δυνατότητα εκτύπωσης του κειμένου, μαζί με τις όποιες υπογραμμίσεις έχει κάνει.

#### 3.3.5.2 Υλοποίηση

Το πρώτο που πρέπει να γίνει για την εκτύπωση, είναι να δημιουργηθεί ένα αντικείμενο τύπου PrinterJob. Η PrinterJob είναι η κύρια κλάση που ελέγχει όλες τις εργασίες που σχετίζονται με την εκτύπωση. Μέσω αυτής μπορεί να ετοιμαστεί μια νέα διεργασία εκτύπωσης, να εμφανιστεί στον χρήστη το αντίστοιχο, προαιρετικό, παράθυρο και φυσικά να εκτυπώσει. Στον κώδικα μας, το αντικείμενο αυτό, δημιουργείται στην κλάση Controller, μόλις ο χρήστης επιλέξει Print μέσα από το μενού File.

```
PrinterJob pj = PrinterJob.getPrinterJob();
```

Δεύτερο βήμα είναι να οριστεί το τι θα εκτυπωθεί και με ποιον τρόπο θα εκτυπωθεί.

```
pj.setPrintable(new AppPrinter());
```

Η AppPrinter κλάση, υλοποιεί το interface Printable, και περιέχει μία μόνο μέθοδο, την print(), η οποία και διαχειρίζεται τις προς εκτύπωση σελίδες. Την μέθοδο θα την εξηγήσουμε παρακάτω. Επόμενο βήμα είναι η εμφάνιση του παραθύρου, μέσω του οποίου ο χρήστης μπορεί να ορίσει διάφορες παραμέτρους, με σημαντικότερη την επιλογή του εκτυπωτή.

```
pj.printDialog()
```

Το εν λόγω παράθυρο θα φαίνεται μέχρι ο χρήστης επιβεβαιώσει ή ακυρώσει την εκτύπωση. Η κλήση της μεθόδου θα επιστρέψει true εάν επιβεβαιωθεί η εκτύπωση ή false αν ακυρωθεί. Σε περίπτωση που επιστραφεί true, θα σταλεί το κείμενο για εκτύπωση.

```

if (pj.printDialog()) {
    try {
        pj.print();
    } catch (PrinterException exc) {
        JOptionPane.showMessageDialog(this, "An error occurred while
        printing. Please check your printer", "Error Message",
        JOptionPane.ERROR_MESSAGE);
    }
}

```

Η εξαίρεση θα προκύψει αν προκληθεί οποιοδήποτε λάθος κατά την διαδικασία της εκτύπωσης, και θα εμφανιστεί το αντίστοιχο μήνυμα λάθους στον χρήστη.

### 3.3.5.3 Υλοποίηση της μεθόδου print()

Όπως έχει ήδη αναφερθεί, η μέθοδος print(Graphics g, PageFormat pf, int page) της κλάσης AppPrinter είναι υπεύθυνη για την διαχείριση των προς εκτύπωση σελίδων. Με λίγα λόγια το πώς και τι θα εκτυπωθεί. Στην εφαρμογή μας, αυτό που κάθε φορά εκτυπώνεται είναι η σελίδα του βιβλίου που εκείνη την στιγμή είναι ανοικτή. Μέσω του προγράμματος όμως ο χρήστης διαβάζει html αρχεία, και κάθε σελίδα του βιβλίου δεν σημαίνει απαραίτητα ότι αντιστοιχεί και σε μια πραγματική A4 σελίδα. Γι' αυτό το πρώτο που πρέπει να υπολογιστεί, είναι το πλήθος των πραγματικών σελίδων που θα χρειαστούν. Αυτό γίνεται με το να διαιρεθεί της ύψος της σελίδας του προγράμματος μας με το ύψος της σελίδας στην οποία πρόκειται να εκτυπωθεί (έστω για παράδειγμα το ύψος μίας A4 κόλλας).

```

double height = pf.getImageableHeight();
double width = pf.getImageableWidth();
int totalNumPages = (int)Math.ceil(ViewClass.editorPane.getHeight() / height);

```

όπου pf είναι το format της σελίδας, που έχει οριστεί από τον χρήστη μέσω του παραθύρου εκτύπωσης. Χρησιμοποιώντας την μεταβλητή pf για την εύρεση του ύψους με το οποίο θα υπολογιστεί ο αριθμός των σελίδων, είμαστε πάντα σίγουροι ότι ο αριθμός τους θα είναι ανάλογος με το είδος σελίδας που επέλεξε ο χρήστης. Έτσι, αν για παράδειγμα επιλέξει να εκτυπώσει σε A3 σελίδες, ο συνολικός αριθμός που θα χρειαστούν θα είναι μικρότερος από ότι εάν επέλεγε σελίδες A4 κ.ο.κ. Να διευκρινιστεί επίσης, ότι με την χρήση της μεθόδου Math.ceil(double a) παίρνουμε την μικρότερη δεκαδική τιμή που δεν είναι μικρότερη από το όρισμα a (στην περίπτωση μας το αποτέλεσμα της διαίρεσης), και ισούται με έναν ακέραιο. Για ευνόητους λόγους την τιμή αυτή την μετατρέπουμε σε ακέραιο (casting), ώστε να την χρησιμοποιήσουμε παρακάτω. Έχοντας τον απαιτούμενο αριθμό σελίδων,



μπορούμε πλέον να καθορίσουμε τις υπόλοιπες παραμέτρους της σελίδας και το τι θα εκτυπωθεί.

```
Graphics2D g2 = (Graphics2D)g;
g2.translate(0f,-(page*height));
g2.setClip(0, (int)(page*height),(int) width, (int) height);
g2.scale(0.6,0.6);

ViewClass.editorPane.printAll(g2);
```

Με τη χρήση της μεθόδου `printAll(g2)`, ορίζουμε στο πρόγραμμα να εκτυπωθεί οτιδήποτε είναι ορατό εντός του πλαίσιο του `editorPane`, ο οποίος στην περίπτωση μας περιέχει την σελίδα του βιβλίου συμπεριλαμβανομένων και των υπογραμμίσεων.

### 3.3.5.4 Παραδοχές και εξαρτήσεις

- 1) Όταν ορίζουμε το εύρος των προς εκτύπωση σελίδων (π.χ. από 1 έως 3), θα εκτυπωθεί από την τρέχουσα σελίδα του βιβλίου το κομμάτι εκείνο που αντιστοιχεί σε τρεις σελίδες. Δεν θα εκτυπωθούν τρεις διαφορετικές σελίδες του βιβλίου (π.χ. οι σελίδες `index`, `applet` και `dialog`).
- 2) Υποστήριξη για εκτύπωση πολλών αντιτύπων δεν υπάρχει στην παρούσα έκδοση.

## 3.3.6 Λειτουργία αλλαγής κωδικού

### 3.3.6.1 Περιγραφή

Κάθε χρήστης, έχει την δυνατότητα αλλαγής του προσωπικού του κωδικού πρόσβασης. Κάποιος μπορεί να θέλει να αλλάξει τον κωδικό του είτε για να επιλέξει κάποιον τον οποίο μπορεί πιο εύκολα να απομνημονεύσει είτε για λόγους ασφαλείας (βλ. και ενότητα Διασφάλιση ακεραιότητας χρηστών).

### 3.3.6.2 Υλοποίηση

Η διαδικασία που ακολουθείται είναι η τυπική για παρόμοιες περιπτώσεις. Ο χρήστης καλείται να εισάγει τον παλιό του κωδικό, και τον καινούριο εις διπλούν για επιβεβαίωση. Στον κώδικα εκτελούνται συνολικά τέσσερις έλεγχοι. Ο πρώτος αφορά το αν κάποιο από τα πεδία είναι άδειο. Ο

δεύτερος και ο τρίτος υπάρχουν για να επιβεβαιωθεί ότι τα πεδία New Password και Confirm New Password έχουν το ίδιο ακριβώς περιεχόμενο και ο τέταρτος για να επιβεβαιωθεί ότι ο κωδικός που εισήγαγε στο πεδίο Old Password, όντως αντιστοιχεί στα τρέχον κωδικό πρόσβασης του συνδεδεμένου χρήστη. Αφού περαστούν επιτυχώς και οι 4 έλεγχοι, ανανεώνεται το πεδίο password1 του πίνακα users για τον εν λόγω χρήστη και το αντίστοιχο πεδίο του πίνακα administrators σε περίπτωση που ο χρήστης είναι καταχωρημένος ως διαχειριστής.

### **3.3.7 Λειτουργία σύνδεσης ως διαχειριστής**

#### **3.3.7.1 Περιγραφή**

Σε περίπτωση που κάποιος έχει εισέλθει στο σύστημα σαν απλός χρήστης και στην πορεία χρειαστεί να συνδεθεί ως διαχειριστής μπορεί να το κάνει χωρίς να χρειαστεί να κλείσει και να ξαναοίξει το πρόγραμμα μέσω της επιλογής Login as administrator.

#### **3.3.7.2 Υλοποίηση**

Το πρόγραμμα, αφού επιβεβαιώσει ότι τα δοθέντα στοιχεία αντιστοιχούν σε χρήστη με δικαιώματα διαχείρισης, σώζει τις πιθανές αλλαγές οι οποίες έγιναν κατά την διάρκεια της χρήσης της εφαρμογής με τον προηγούμενο λογαριασμό και αφαιρεί από την όψη τις σημειώσεις, τις υπογραμμίσεις και τους σελιδοδείκτες. Κατόπιν, αφού πρώτα ανακτήσει από την βάση και φορτώσει την εμφάνιση (βλ. ενότητα Λειτουργία αλλαγής της εμφάνισης της εφαρμογής) της επιλογής του διαχειριστή, φορτώνει το προκαθορισμένο βιβλίο της εφαρμογής (The Java Tutorial) καθώς επίσης και όλες τις σημειώσεις, τις υπογραμμίσεις και τους σελιδοδείκτες του νέου χρήστη. Εάν καθ' όλη την διάρκεια προκύψει κάποιο σφάλμα, ο χρήστης θα ειδοποιηθεί με το ανάλογο μήνυμα.

## **3.4 Λειτουργίες διαθέσιμες μόνο στους διαχειριστές**

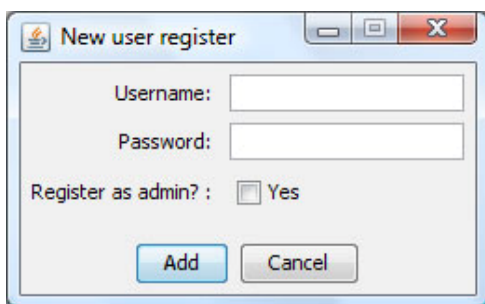
### **3.4.1 Περιγραφή**

Προκειμένου να διευκολυνθούν οι χρήστες με δικαιώματα administrator, έχει δημιουργηθεί ένα ξεχωριστό μενού, το οποίο είναι διαθέσιμο σε αυτούς και μόνο. Φέρει το όνομα Admin Tools, και εμφανίζεται εφόσον ο συνδεδεμένος χρήστης είναι καταχωρημένος στον πίνακα administrators, γεγονός το οποίο και τον πιστοποιεί ως διαχειριστή του

συστήματος. Μέσω του εν λόγω μενού είναι προσπελάσιμες τρεις λειτουργίες. Οι New User Register, Delete User και New Book.

### 3.4.2 Δημιουργία νέου χρήστη

Μέσω αυτής της επιλογής, ο διαχειριστής έχει την δυνατότητα να δημιουργήσει νέους χρήστες ορίζοντας τους τα username και password που θα χρησιμοποιούν για την σύνδεση τους στο πρόγραμμα. Επί προσθέτως ο διαχειριστής έχει τη δυνατότητα να παραχωρήσει στον νέο χρήστη δικαιώματα διαχείρισης, επιλέγοντας την αντίστοιχη επιλογή με όνομα Register as admin. Αυτό σημαίνει πως το username και password που έχει εισάγει θα καταχωρηθούν εκτός από τον πίνακα users της βάσης, και στον πίνακα administrators.



Εικόνα 3.5 : Το παράθυρο μέσα από το οποίο μπορεί να γίνει εγγραφή νέου χρήστη

### 3.4.3 Διαγραφή χρήστη

Κατά την ίδια λογική, ο διαχειριστής μπορεί να διαγράψει από το σύστημα ένα χρήστη. Εκτελώντας ένα απλό ερώτημα στην βάση, εμφανίζονται όλοι οι καταχωρημένοι χρήστες του συστήματος, εξαιρουμένου του χρήστη που είναι επί του παρόντος συνδεδεμένος. Επιλέγοντας έναν από τους χρήστες και επιβεβαιώνοντας, για ασφάλεια, την διαγραφή, ο χρήστης σβήνεται από τους αντίστοιχους πίνακες της βάσης.

### 3.4.4 Δημιουργία νέου βιβλίου

Προκειμένου να μην υπάρχει ανάγκη, ο διαχειριστής να έχει οποιαδήποτε επαφή απευθείας με την βάση δεδομένων, υλοποιήθηκε και μια λειτουργία, μέσω της οποίας θα μπορεί να καταχωρεί στο σύστημα νέα βιβλία.

Το μόνο που χρειάζεται, είναι να έχει το βιβλίο σε μορφή αρχείου .zip. Το πρόγραμμα αναλαμβάνει να αποσυμπιέσει το αρχείο, να αντιγράψει τα περιεχόμενα του κάτω από τον φάκελο classes που βρίσκεται με την σειρά του κάτω από τον φάκελο build, μέσα σε φάκελο με το όνομα που έχει εισάγει ο διαχειριστής και να το καταχωρήσει στην βάση δεδομένων

Οι μέθοδοι που είναι κατά κύριο λόγο υπεύθυνες για αυτήν την λειτουργία είναι η doBrowse() και η unzipFile() της κλάσης NewBookFrame. Αφού επιλέξει ο χρήστης το συμπιεσμένο αρχείο που περιέχει το βιβλίο μέσω της επιλογής Browse, αναλαμβάνει η doBrowse() μέθοδος να διαβάσει τα περιεχόμενα του αρχείου, και να εμφανίσει στο drop down μενού τα ονόματα των html αρχείων που περιέχει.

```

fis = new FileInputStream(packetURL.getText());
zin = new ZipInputStream(new BufferedInputStream(fis));

while((entry = zin.getNextEntry()) != null) {
    files.add(entry.getName());
}

indexName.addItem(files.get(i));

```

Αυτή η διαδικασία έχει ως στόχο να διευκολύνει τον διαχειριστή στο να επιλέξει το αρχείο εκείνο το οποίο θα χρησιμοποιηθεί ως η πρώτη σελίδα του βιβλίου. Εάν αντί του drop down μενού, υπήρχε πεδίο εισαγωγής χαρακτήρων, τότε θα υπήρχε μεγάλη πιθανότητα ορθογραφικού λάθους το οποίο θα μας οδηγούσε σε λανθασμένη καταχώρηση στην βάση και κατ'επέκταση στη μη λειτουργία του νέου βιβλίου. Αφού γίνουν όλες οι απαραίτητες διαδικασίες από τον χρήστη και πατηθεί το κουμπί Ok, αναλαμβάνει η μέθοδος unzipFile(). Μέσω αυτής γίνονται όλες οι απαραίτητες ενέργειες για την σωστή αποσυμπιέση του αρχείου. Αφού διαβαστεί το αρχείο, δημιουργούνται οι υποφάκελοι, αν υπάρχουν και αντιγράφονται όλα τα αρχεία στο σωστό σημείο. Όλο το βιβλίο μπαίνει κάτω από φάκελο με όνομα το όνομα του βιβλίου που εισήγαγε ο χρήστης δίχως τα κενά.

Διάβασμα του αρχείου :

```

fis = new FileInputStream(file);
zin = new ZipInputStream(new BufferedInputStream(fis));

```

Δημιουργία των υποφακέλων, εάν υπάρχουν :

```

if(entry.getName().contains("/")){
    File dir = new File(FOLDER + path +
        entry.getName().substring(0,entry.getName().lastIndexOf("/")+1));
    dir.mkdir();
}

```

Αποσυμπιέση των αρχείων :

```

fos = new FileOutputStream(FOLDER + path + entry.getName());
dest = new BufferedOutputStream(fos, BUFFER);

while ((count = zin.read(data, 0, BUFFER)) != -1)
    dest.write(data, 0, count);

```

Τέλος, έχοντας πλέον όλα τα απαραίτητα στοιχεία διαθέσιμα – το όνομα του βιβλίου και την σχετική διεύθυνση της πρώτης σελίδας - καταχωρείται και στον πίνακα books της βάσης μας, εγγραφή που αφορά το καινούριο βιβλίο.

Καθ' όλη την διάρκεια της λειτουργίας δημιουργίας νέου βιβλίου, γίνονται συνεχώς έλεγχοι, ώστε ο χρήστης να ειδοποιηθεί με το κατάλληλο μήνυμα σε περίπτωση που προκύψει κάποιο λάθος.

#### 3.4.4.1 Παραδοχές και εξαρτήσεις

- 1) Το συμπιεσμένο αρχείο πρέπει να έχει συγκεκριμένη μορφή. Όλα τα html αρχεία, ή αλλιώς οι σελίδες του βιβλίου, πρέπει να βρίσκονται στο πρώτο επίπεδο. Δεν πρέπει δηλαδή να περιέχονται μέσα σε άλλο φάκελο. Από εκεί και πέρα το zip μπορεί να έχει όσα επίπεδα φακέλων και υποφακέλων μπορεί να χρειάζονται (μέσα σε φάκελο συνήθως, υπάρχουν οι όποιες εικόνες χρησιμοποιεί το βιβλίο). Εικόνα με παράδειγμα της δομής ενός τέτοιου αρχείου μπορείτε να βρείτε στο Παράρτημα, στην ενότητα Εγχειρίδιο Χρήσης, στην παράγραφο που αναφέρεται στην εγκατάσταση ενός βιβλίου.
- 2) Το όνομα του βιβλίου μπορεί να είναι μέχρι και 60 χαρακτήρες σε μήκος.
- 3) Το όνομα του βιβλίου δεν πρέπει να είναι γραμμένο με ελληνικούς χαρακτήρες.
- 4) Κάθε βιβλίο πρέπει να έχει μοναδικό όνομα στην βάση.
- 5) Το συμπιεσμένο αρχείο πρέπει να είναι τύπου .zip

## 3.5 Επιπλέον χαρακτηριστικά

### 3.5.1 Κρυπτογράφηση των κωδικών των χρηστών (Encryption)

#### 3.5.1.1 Περιγραφή

Στο τελικό στάδιο του το σύστημα μπορεί να διανέμεται ως διαδικτυακή εφαρμογή. Ο χρήστης έχει την επιλογή να το εκτελεί είτε τοπικά είτε μέσω διαδικτύου, αλλά η βάση δεδομένων θα βρίσκεται σε κάποιον διαδικτυακό τόπο, ώστε όλοι οι χρήστες να έχουν πρόσβαση στα δεδομένα από οπουδήποτε. Αυτό το χαρακτηριστικό όμως εγείρει το σημαντικό ζήτημα της ασφάλειας των δεδομένων. Ένα πρώτο βήμα για την διαφύλαξη των δεδομένων έγινε με το να κρυπτογραφηθούν οι κωδικοί πρόσβασης που έχουν οι χρήστες. Για τον σκοπό αυτό, επιλέχθηκε το πρότυπο AES (Advanced Encryption Standard) , το οποίο το 2000 αντικατέστησε το απαραιτούμενο DES (Data Encryption Standard).

Το AES είναι ένα ομοσπονδιακό πρωτόκολλο για συμμετρική κρυπτογράφηση. Υποστηρίζει συνδυασμούς κλειδιών και μεγέθους μπλοκ των 128, 192 και 256 bit. Ο αλγόριθμος που επιλέχθηκε για αυτό το πρωτόκολλο ονομάζεται Rijndael και επινοήθηκε από δύο Βέλγους κρυπτογράφους. Από την έκδοση Java 2 SDK, Standard Edition (J2SE) v 1.4.0 και μετά, το πακέτο Java Cryptography Extension (JCE) ενσωματώθηκε στα SDK και JRE, κάτι το οποίο σημαίνει ότι δεν χρειάζεται να γίνει εγκατάσταση κάποιου επιπλέον πακέτου για να γίνει χρήση των αλγορίθμων κρυπτογράφησης.

#### 3.5.1.2 Υλοποίηση

Για την κρυπτογράφηση στο πρόγραμμα μας χρησιμοποιείται η κλάση με όνομα AES. Αυτή περιέχει όλες τις απαραίτητες μεθόδους για κρυπτογράφηση/αποκρυπτογράφηση καθώς και την μέθοδο για την δημιουργία ενός τυχαίου κλειδιού. Το κλειδί αυτό, αποθηκεύεται σε ένα αρχείο με όνομα secret.key και είναι η βάση για όλη την διαδικασία της κρυπτογράφησης/αποκρυπτογράφησης. Από την στιγμή που θα δημιουργηθούν χρήστες και κρυπτογραφηθούν οι κωδικοί πρόσβασης τους σύμφωνα με ένα συγκεκριμένο κλειδί, αυτό θα πρέπει να μείνει αναλλοίωτο. Λόγω του ότι παράγεται τυχαία εάν χαθεί ή αλλαχθεί, η ανάκτηση των κωδικών θα είναι αδύνατη.

Αφού έχουμε στην κατοχή μας ένα μοναδικό κλειδί, μπορούμε πλέον να κρυπτογραφήσουμε ένα κωδικό. Η διαδικασία αυτή εκτελείται είτε κατά την δημιουργία του λογαριασμού του ή φυσικά σε περίπτωση που ο χρήστης επιλέξει να αλλάξει τον κωδικό του. Καλείται η μέθοδος `encrypt()`, και ο κωδικός αποθηκεύεται πλέον στην βάση ως μια σειρά από κωδικοποιημένα bytes (bytea τύπος δεδομένων για την PostgreSQL ) και όχι σαν απλή συμβολοσειρά (String) που χρησιμοποιούταν μέχρι πρότινος. Με τον τρόπο αυτό είναι αδύνατον για κάποιον να ανακτήσει τους κωδικούς ακόμα και εάν αποκτήσει με κάποιον τρόπο πρόσβαση στην βάση.

Η διαδικασία αποκρυπτογράφησης είναι η ακριβώς αντίθετη με αυτήν της κρυπτογράφησης. Κάθε φορά που κάποιος χρήστης εισέρχεται στο πρόγραμμα, ανακτάται από την βάση ο κωδικός του και καλείται η μέθοδος `decrypt()` για να τον αποκρυπτογραφήσει.

### 3.5.1.3 Εναλλακτικές λύσεις

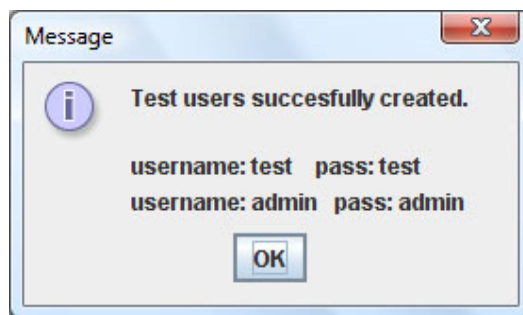
Η χρήση του προτύπου AES δεν αποτελεί και την μόνη λύση με την οποία η κρυπτογράφηση των κωδικών των χρηστών θα μπορούσε να υλοποιηθεί. Εναλλακτικά θα μπορούσε να χρησιμοποιηθεί το πρότυπο DES (Data Encryption Standard), το οποίο είναι επίσης πρότυπο συμμετρικής κρυπτογράφησης. Το DES χρησιμοποιεί για την κρυπτογράφηση κλειδί των 56 bits (εν αντιθέσει με το AES που μπορεί να χρησιμοποιήσει 128, 192 και 256 bit ), και έχει σταματήσει πλέον να χρησιμοποιείται ευρέως (όπως έχει ήδη αναφερθεί το AES είναι αυτό που το αντικατέστησε), για τον απλό λόγο ότι μπορεί κάποιος εύκολα να το σπάσει με brute force. Η επιλογή μας οπότε του AES έναντι του DES είναι εύλογη.

Τα AES και DES είναι πρότυπα συμμετρικής κρυπτογράφησης. Αυτό πρακτικά σημαίνει ότι το ίδιο κλειδί χρησιμοποιείται και για την κρυπτογράφηση αλλά και για την αποκρυπτογράφηση των δεδομένων. Το κύριο μειονέκτημα της συμμετρικής κρυπτογράφησης είναι η διάθεση του κλειδιού. Το ίδιο κλειδί πρέπει να το έχουν και οι δυο μεριές – αυτός που κρυπτογραφεί το μήνυμα και αυτός που το αποκρυπτογραφεί. Οπότε απαραίτητη προϋπόθεση είναι η διάθεση του κλειδιού με κάποιον ασφαλή τρόπο (συνήθως μέσω ασφαλών καναλιών). Τη λύση σε αυτό το πρόβλημα προσφέρει η κρυπτογράφηση με δημόσιο κλειδί (public key cryptography). Με αυτούς τους αλγορίθμους, δημιουργείται ένα ζευγάρι κλειδιών - ένα δημόσιο κλειδί (public key) και ένα ισοδύναμο μυστικό κλειδί (private key). Για την κρυπτογράφηση χρησιμοποιείται το public κλειδί το οποίο και μπορεί να διατεθεί ελεύθερα χωρίς κανένα κίνδυνο, εν αντιθέσει με το private κλειδί, το οποίο και πρέπει να φυλάσσεται καλά. Η χρήση κάποιου προτύπου

κρυπτογράφησης με δημόσιο κλειδί μπορεί να προσφέρει αυξημένη ασφάλεια αλλά είναι πολύ πιο αργή συγκριτικά με πρότυπα όπως είναι τα AES και DES. Για να μην επιβαρύνουμε λοιπόν την εφαρμογή μας με παραπάνω υπολογιστικό κόστος, αποφασίσαμε ως μια πρώτη προσέγγιση μιας ασφαλέστερης λειτουργίας να χρησιμοποιήσουμε κάποιον από τους διαθέσιμους συμμετρικούς αλγορίθμους.

#### 3.5.1.4 Δημιουργία αρχικών χρηστών

Ένα λογικό και αναμενόμενο πρόβλημα που προέκυψε, ήταν το πώς κατά την πρώτη εκτέλεση του προγράμματος θα δημιουργηθούν οι πρώτοι χρήστες. Μέχρι τώρα, ο administrator της βάσης, μπορούσε να εισάγει κατευθείαν στα αντίστοιχα πεδία του πίνακα users το ονόματα και τους κωδικούς νέων χρηστών. Από την στιγμή όμως που χρησιμοποιείται κρυπτογράφηση, είναι αδύνατο για τον οποιονδήποτε να κάνει κάτι τέτοιο. Για τον λόγο αυτό δημιουργήθηκε η κλάση με όνομα CreateTestUsers, η οποία κατά την εκτέλεση της, το μόνο που κάνει είναι να εισάγει στη βάση δύο χρήστες. Ένα απλό χρήστη με όνομα πρόσβασης test και κωδικό test, και ένα χρήστη με δικαιώματα administrator με όνομα και κωδικό πρόσβασης admin. Με τους δύο λογαριασμούς αυτούς, μπορεί κάποιος να συνδεθεί στο πρόγραμμα και να αρχίσει να δημιουργεί νέους λογαριασμούς κατά βούληση.



Εικόνα 3.6 : Το μήνυμα επιτυχούς δημιουργίας των αρχικών χρηστών

### 3.5.2 Διασφάλιση ακεραιότητας χρηστών

#### 3.5.2.1 Περιγραφή

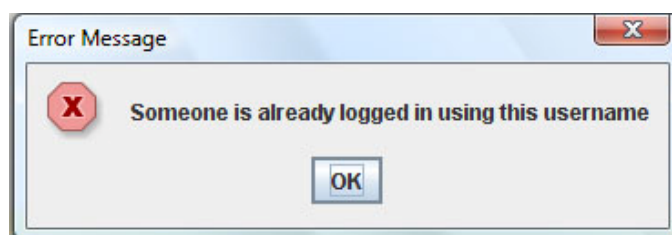
Ένα άλλο θέμα με το οποίο ασχοληθήκαμε είναι αυτό της ακεραιότητας των χρηστών. Λήφθηκε ως δεδομένο ότι τα στοιχεία του κάθε λογαριασμού είναι μοναδικά και τα έχει στην κατοχή του μόνο ο χρήστης στον οποίο αναφέρονται. Με δεδομένο αυτό, ερχόμαστε εύκολα στο συμπέρασμα ότι με ένα συγκεκριμένο λογαριασμό, στο σύστημα κάθε στιγμή μπορεί να είναι κάποιος συνδεδεμένος μια μόνο φορά. Για λόγους ασφάλειας λοιπόν αλλά και



για λόγους ακεραιότητας των δεδομένων, αποφασίστηκε να υλοποιηθεί κατά την είσοδο ενός χρήστη έλεγχος, σχετικά με το αν είναι συνδεδεμένος εκείνη την στιγμή κάποιος άλλος με τα ίδια στοιχεία.

### 3.5.2.2 Υλοποίηση

Η υλοποίηση αυτού του ελέγχου είναι αρκετά απλή. Στον πίνακα users προστέθηκε ένα ακόμα πεδίο με όνομα logged, τύπου Boolean και αρχική προκαθορισμένη τιμή false. Κάθε φορά που ένας χρήστης πάει να συνδεθεί, γίνεται έλεγχος της τιμής του συγκεκριμένου πεδίου και εφόσον προκύψει ότι η τιμή του είναι false προβαίνει στις απαραίτητες λειτουργίες για την εκκίνηση του προγράμματος. Κατά την είσοδο του χρήστη μεταξύ όλων των άλλων διαδικασιών, ορίζεται και η τιμή του πεδίου logged να δείχνει true. Αντίστοιχα κατά την έξοδο από το πρόγραμμα, η τιμή του επανέρχεται σε false. Έτσι, εάν κάποιος πάει να συνδεθεί με λογαριασμό που χρησιμοποιείται ήδη θα δεχθεί ένα μήνυμα λάθους και θα αποτύχει η σύνδεση.



Εικόνα 3.7 : Το μήνυμα λάθους που επιστρέφεται

### 3.5.3 Χρήση ανακυκλώσιμων συνδέσεων (Connection Pool)

#### 3.5.3.1 Περιγραφή

Ένα από τα μειονεκτήματα της εφαρμογής ήταν η αυξημένη καθυστέρηση, η οποία οφειλόταν στις πολλές προσπελάσεις που πραγματοποιούσε στην βάση δεδομένων. Δεδομένου ότι μόνο για την αλλαγή μίας σελίδας πραγματοποιούνταν τουλάχιστον τρεις διαφορετικές συνδέσεις (μεταξύ εφαρμογής-βάσης δεδομένων) - μία για να φορτωθούν οι σελιδοδείκτες (bookmarks), μία για τις σημειώσεις και μία για τις υπογραμμίσεις (highlights) του εκάστοτε χρήστη - κρίθηκε απαραίτητη η εύρεση λύσης, η οποία θα βελτίωνε την απόδοση της εφαρμογής.

Μέχρι πρότινος η μεθοδολογία είχε ως εξής. Δημιουργούσαμε μία σύνδεση με την βάση, εκτελούσαμε κάποιες εργασίες (π.χ. ανάκτηση των προσωπικών σημειώσεων του χρήστη) και τέλος κλείναμε την σύνδεση. Για

απλά προγράμματα που δεν απαιτούν συχνή επικοινωνία, η διαδικασία αυτή δουλεύει χωρίς ιδιαίτερα προβλήματα. Δυστυχώς όμως, σε μεγάλες εφαρμογές τα αποτελέσματα της διαδικασίας αυτής δεν είναι καθόλου ικανοποιητικά. Αυτό οφείλεται στο γεγονός ότι η δημιουργία μίας σύνδεσης είναι πολύ δαπανηρή διεργασία για τις περισσότερες μηχανές βάσεως δεδομένων. Δεδομένου αυτού και συνυπολογίζοντας το γεγονός ότι κατά την λειτουργία του προγράμματος δημιουργούνταν και καταστρέφονταν πάρα πολλές συνδέσεις, γίνεται αντιληπτό το μέγεθος του προβλήματος. Πλέον το σχήμα «άνοιγμα σύνδεσης, επικοινωνία με την βάση, κλείσιμο σύνδεσης» όχι μόνο δεν προσφέρει ικανοποιητικά αποτελέσματα, αλλά και συνεισφέρει σε μεγάλο βαθμό στην μη ομαλή λειτουργία του προγράμματος.

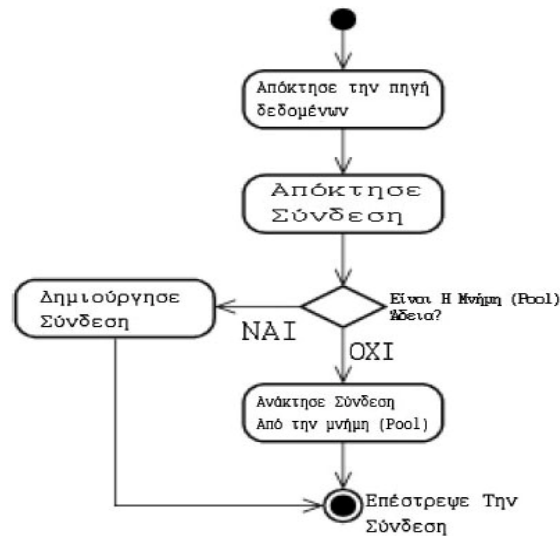
### 3.5.3.2 Τρόπος λειτουργίας

Η λύση δόθηκε με την χρήση ανακυκλώσιμων συνδέσεων(Connection Pooling). Ανακύκλωση συνδέσεων είναι ο μηχανισμός, μέσω του οποίου οι ανοικτές συνδέσεις αποθηκεύονται προσωρινά σε μια λανθάνουσα μνήμη (cache) για χρήση και επαναχρησιμοποίηση τους από διαφορετικά μέρη της εφαρμογής. Έτσι όταν η εφαρμογή καλεί την μέθοδο δημιουργίας σύνδεσης, το πρώτο που κάνει είναι να ελέγξει την προσωρινή μνήμη. Εάν υπάρχει κάποια σύνδεση αποθηκευμένη, αντί να δημιουργήσει μια νέα (αυξάνοντας έτσι το υπολογιστικό κόστος) επαναχρησιμοποιεί την ήδη υπάρχουσα. Αφού εκτελέσει τις όποιες εργασίες έχει, ενώ φαινομενικά τερματίζει την σύνδεση, την επιστρέφει στην μνήμη, κρατώντας ανοικτή την φυσική διασύνδεση. Έτσι, υπάρχει μια συνεχής ανακύκλωση συνδέσεων, μειώνοντας αισθητά το κόστος.

### 3.5.3.3 Υλοποίηση

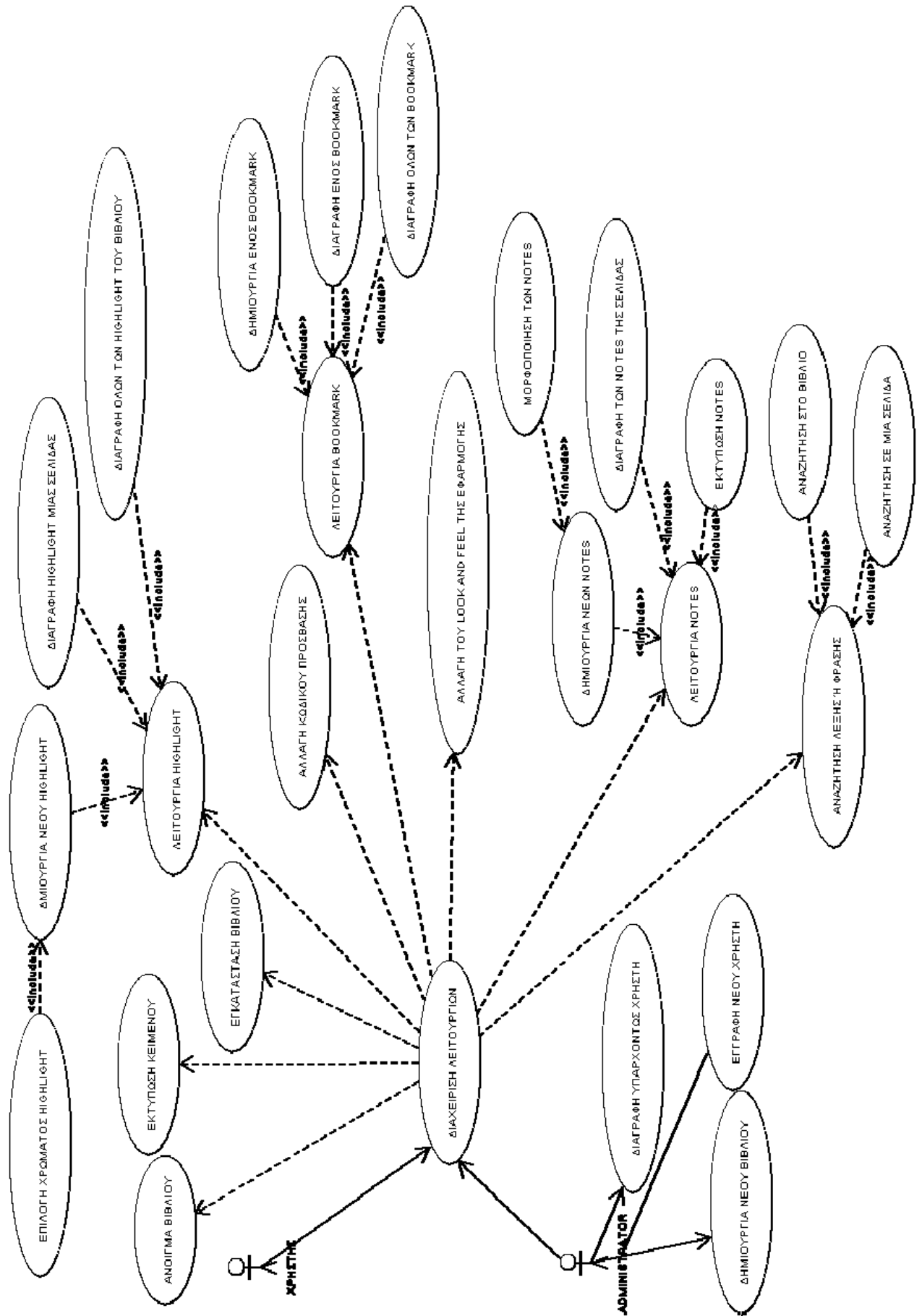
Πρώτο βήμα ήταν να βρεθεί κάποια υλοποίηση ενός περιβάλλοντος Connection Pool, το οποίο θα συνεργαζόταν σωστά με την εφαρμογή μας, μιας και η Java δεν περιέχει υλοποιημένο κάποιο έτοιμο πακέτο για τον σκοπό αυτό. Ύστερα από δοκιμές διαφόρων API (Application Programming Interface) από αρκετούς διαφορετικούς κατασκευαστές, καταλήξαμε στην χρήση του προαιρετικού πακέτου που προσφέρει η PostgreSQL (JDBC 2.0 Optional Package), ως αυτό το οποίο καλύπτει καλύτερα τις ανάγκες μας.

Από προγραμματιστική άποψη, δεν υπάρχει καμία διαφορά μεταξύ των συνηθισμένων συνδέσεων και των ανακυκλώσιμων (pooled).



Εικόνα 3.8 : Διάγραμμα που δείχνει τον τρόπο με τον οποίο λειτουργούν οι ανακυκλώσιμες συνδέσεις

Όπως φαίνεται και στο σχήμα, η εφαρμογή επικοινωνεί μόνο με την υλοποίηση μιας πηγής δεδομένων (DataSource), η οποία κρατάει όλες τις πληροφορίες που χρειάζεται η σύνδεση. Εσωτερικά, η πηγή δεδομένων είναι αυτή που επικοινωνεί με την κρυφή μνήμη η οποία περιέχει όλες τις ανακυκλώσιμες συνδέσεις, καθιστώντας έτσι όλη την διαδικασία διάφανη στον προγραμματιστή. Έτσι, ενώ η κλήση της μεθόδου δημιουργίας σύνδεσης (getConnection() ) φαίνεται να λειτουργεί ακριβώς με τον ίδιο τρόπο με τις απλές συνδέσεις, στην πραγματικότητα αντί να δημιουργηθεί νέα σύνδεση ανακτάται μια ήδη υπάρχουσα από την μνήμη.



Εικόνα 3.9 : Το use case διάγραμμα του συστήματος

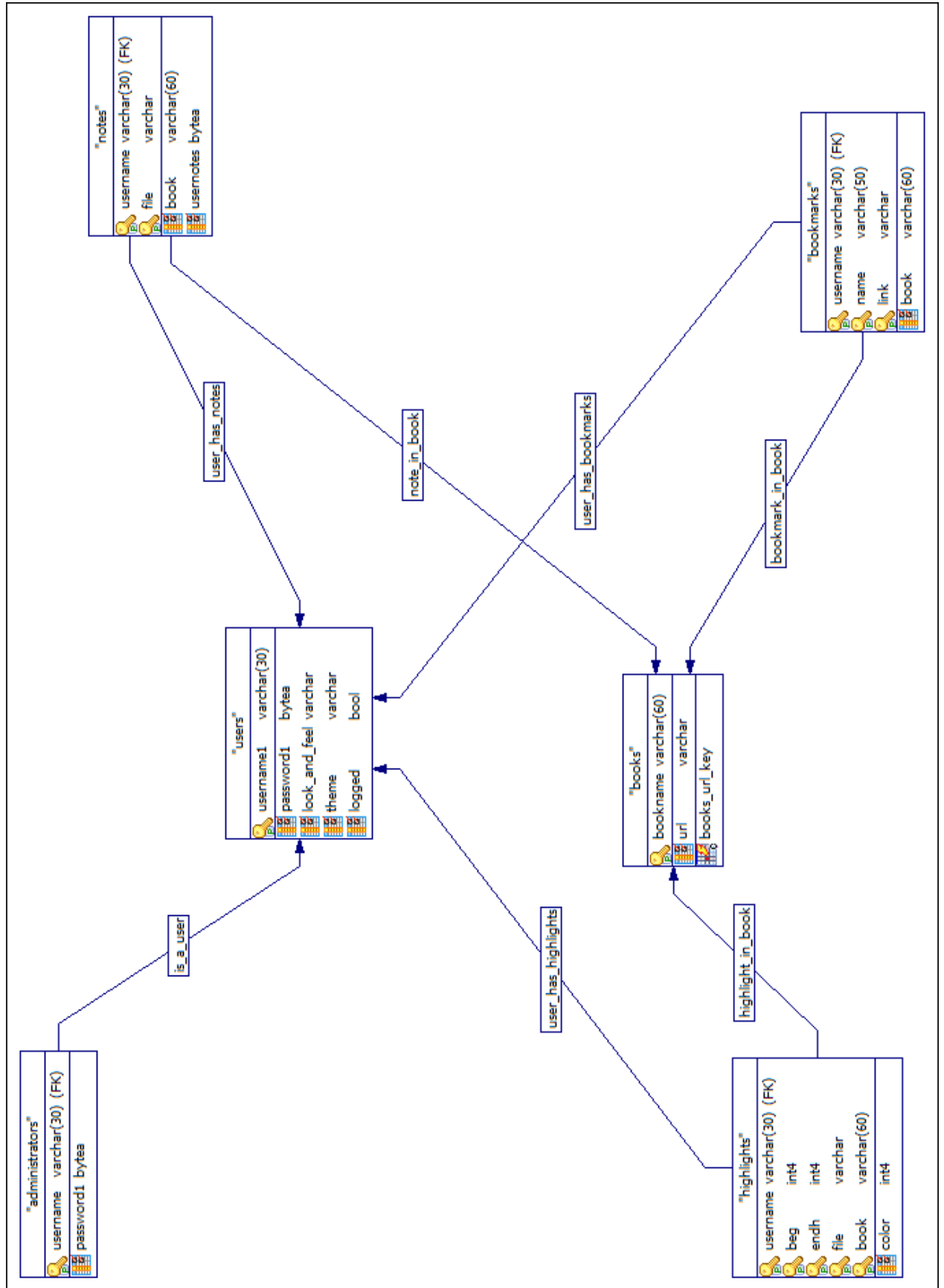
### 3.6 Βάση δεδομένων

Προκειμένου να υποστηριχθούν οι νέες λειτουργίες, αλλά και για να δημιουργηθεί ένα ορθότερο (κατά προσωπική εκτίμηση) σχεσιακό μοντέλο, η βάση δεδομένων επανασχεδιάστηκε. Πρωταρχικός στόχος, πέραν της υποστήριξης των νέων λειτουργιών, ήταν η δημιουργία μιας βάσης με ένα μικρό και σταθερό αριθμό πινάκων εν αντιθέσει με το προηγούμενο μοντέλο, κατά το οποίο ο αριθμός των πινάκων ήταν άμεσα εξαρτημένος από το πλήθος των λογαριασμών χρηστών που υπήρχαν (κάθε χρήστης είχε αποκλειστικούς τρεις πίνακες οι οποίοι δημιουργούνταν κατά την πρώτη του είσοδο στο σύστημα – ένα πίνακα για τους σελιδοδείκτες, ένα για τις υπογραμμίσεις και ένα τις σημειώσεις του).

Η βάση πλέον αποτελείται από 6 πίνακες :

- Πίνακας *Users* : Περιέχει τα στοιχεία που χαρακτηρίζουν κάθε χρήστη
  - ❖ `username1`: κρατάει το όνομα του κάθε χρήστη
  - ❖ `password1`: καταχωρείται ο προσωπικός κωδικός κάθε χρήστη
  - ❖ `look_and_feel`: περιέχει εμφάνιση του προγράμματος που έχει επιλεγθεί. Η προκαθορισμένη τιμή του πεδίου είναι η `com.sun.java.swing.plaf.windows.WindowsLookAndFeel`, που αντιστοιχεί σε εμφάνιση παρόμοια με αυτή των `Windows`.
  - ❖ `theme`: περιέχει το επιθυμητό `theme` για συγκεκριμένο `look and feel`. Εφόσον το επιλεγμένο `look and feel` δεν προσφέρει παραπάνω από ένα `theme` η τιμή του πεδίου είναι `null`.
  - ❖ `Logged`: παίρνει τιμές `true` ή `false` αναλόγως με το αν ο χρήστης είναι συνδεδεμένος στο σύστημα. Αρχική προκαθορισμένη τιμή είναι η `false`.
- Πίνακας *administrators*: Περιέχει τα στοιχεία των χρηστών οι οποίοι έχουν δικαιώματα διαχείρισης
  - ❖ `username`: το όνομα του κάθε χρήστη
  - ❖ `password1`: ο προσωπικός κωδικός κάθε χρήστη
- Πίνακας *books*: Περιέχει τα στοιχεία όλων των διαθέσιμων βιβλίων

- ❖ bookname: το όνομα του βιβλίου
  - ❖ url: η σχετική διεύθυνση της πρώτης σελίδας του βιβλίου
- Πίνακας bookmarks: Περιέχει τα στοιχεία όλων των σελιδοδεικτών
- ❖ username: το όνομα του χρήστη που δημιούργησε τον εκάστοτε σελιδοδείκτη
  - ❖ name: το όνομα που έχει δοθεί στον σελιδοδείκτη
  - ❖ link: το αρχείο/σελίδα στο οποίο δείχνει ο σελιδοδείκτης
  - ❖ book: το όνομα του βιβλίου που περιέχει τη συγκεκριμένη σελίδα
- Πίνακας *highlights*: Περιέχει τα στοιχεία που αφορούν τις υπογραμμίσεις
- ❖ username: το όνομα του χρήστη που δημιούργησε την υπογράμμιση
  - ❖ beg: η θέση μέσα στο κείμενο από την οποία ξεκινάει η υπογράμμιση
  - ❖ endh: η θέση του κειμένου όπου τελειώνει η υπογράμμιση
  - ❖ file: το αρχείο/σελίδα στο οποίο δημιουργήθηκε η υπογράμμιση
  - ❖ book: το όνομα του βιβλίου που περιέχει τη συγκεκριμένη σελίδα
  - ❖ color: το χρώμα που επιλέχθηκε
- Πίνακας *notes*: Περιέχει τις πληροφορίες σχετικά με τις σημειώσεις
- ❖ username: το όνομα του χρήστη που δημιούργησε την σημείωση
  - ❖ file: το αρχείο/σελίδα στο οποίο δημιουργήθηκε η σημείωση
  - ❖ book: το όνομα του βιβλίου που περιέχει τη συγκεκριμένη σημείωση στη συγκεκριμένη σελίδα
  - ❖ usernotes: το κείμενο της σημείωσης που έχει πληκτρολογήσει ο χρήστης



Εικόνα 3.10 : Το διάγραμμα των συσχετίσεων της βάσης δεδομένων

## ΚΕΦΑΛΑΙΟ 4

### ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

#### 4.1 Συμπεράσματα

Η ανάπτυξη μίας πλατφόρμας ανάγνωσης ηλεκτρονικού βιβλίου είναι μια αρκετά πολύπλοκη διαδικασία. Ο χρήστης ενός τέτοιου συστήματος έχει πολλές απαιτήσεις και η σύγκριση των ηλεκτρονικών βιβλίων με τα έντυπα είναι κάθε στιγμή αναπόφευκτη. Προκειμένου να ικανοποιηθεί ο χρήστης, πρέπει αφενός να του προσφερθούν δυνατότητες που έχει και κατά την ανάγνωση ενός απλού βιβλίου και αφετέρου λειτουργίες που εκμεταλλεύονται την προσφερόμενη τεχνολογία.

Βλέποντας το σύστημα στην τελική του μορφή, κρίνουμε ότι ο στόχος για την δημιουργία ενός ολοκληρωμένου συστήματος ανάγνωσης ηλεκτρονικών βιβλίων επιτεύχθηκε. Πλέον, έχουμε στα χέρια μας ένα σύστημα το οποίο μπορεί να χρησιμοποιηθεί αποδοτικά για την διάθεση και για την ανάγνωση επιπλέον βιβλίων σε εκπαιδευτικά ιδρύματα. Με την βελτίωση λειτουργιών όπως είναι η υπογράμμιση και την προσθήκη νέων όπως για παράδειγμα η αναζήτηση ή η υποστήριξη πολλαπλών βιβλίων, η διαδικασία ανάγνωσης καθίσταται εύκολη και αποδοτική.

Και από την οπτική του διαχειριστή, το σύστημα μπορεί να θεωρηθεί πλήρες. Πλέον δεν υπάρχει ανάγκη για απευθείας πρόσβαση στην βάση δεδομένων, αφού έχουν στην διάθεση τους όλα τα απαραίτητα εργαλεία για την ορθή λειτουργία του συστήματος. Μέσα από το ίδιο το πρόγραμμα, μπορούν να δημιουργηθούν ή να διαγραφούν χρήστες και να εγκατασταθούν νέα βιβλία.

Τέλος, με την βελτίωση του τρόπου λειτουργίας του συστήματος και με ενσωμάτωση χαρακτηριστικών όπως είναι η χρήση ανακυκλώσιμων συνδέσεων και η κρυπτογράφηση των κωδικών πρόσβασης, έχουμε καταλήξει σε ένα αξιόπιστο, αποτελεσματικό και ασφαλές σύστημα.

#### 4.2 Μελλοντικές επεκτάσεις

Το γεγονός ότι το σύστημα που αναπτύχθηκε αποτελεί συνέχεια κάποιου, δεν σημαίνει ότι δεν υπάρχουν περιθώρια περαιτέρω ανάπτυξης και



βελτίωσης. Όπως κάθε άλλο σύστημα, έτσι και αυτό πρέπει να δεχτεί ακόμα αρκετές βελτιώσεις και προσθήκες μέχρι να καλύψει πλήρως τις ανάγκες και απαιτήσεις των χρηστών. Για το λόγο αυτό θεωρήσαμε απαραίτητο να κατασκευάσουμε μία λίστα με ενδεικτικές λειτουργίες ή βελτιώσεις που μπορούν να προστεθούν στο σύστημα:

- 1) Ένα από τα πλεονεκτήματα που έχει το ηλεκτρονικό βιβλίο έναντι του έντυπου, είναι ότι μπορεί να εκμεταλλευτεί την προσφερόμενη τεχνολογία. Θα ήταν λοιπόν καλή ιδέα λοιπόν, να γίνει προσπάθεια ενσωμάτωσης πολυμέσων (όπως είναι για παράδειγμα ο ήχος και το βίντεο) στην εφαρμογή μας.
- 2) Ένα πολύ χρήσιμο χαρακτηριστικό θα ήταν να γίνει με κάποιο τρόπο συσχέτιση των σημειώσεων με τμήματα του κειμένου, ώστε ο χρήστης να μπορεί να κρατά σε κάθε σελίδα αρκετές σημειώσεις οι οποίες θα αναφέρονται και σε διαφορετικό χωρίο του κειμένου.
- 3) Θα μπορούσε επίσης να γίνει μετατροπή της εφαρμογής σε πλήρως διαδικτυακή, ώστε να μην είναι απαραίτητο ο χρήστης να την εγκαταστήσει στον υπολογιστή του. Αυτό θα μπορούσε να γίνει είτε μετατρέποντας την σε applet είτε Java Web Start πρόγραμμα.
- 4) Χρήσιμο επίσης θα ήταν κάθε βιβλίο που ανοίγει ο χρήστης να ανοίγει και σε διαφορετικό tab, ώστε να μπορούν να υπάρχουν πολλά βιβλία ανοικτά ταυτόχρονα.
- 5) Καλό θα ήταν οι διαχειριστές να μπορούν να διαγράφουν βιβλία.
- 6) Όσον αφορά τους διαχειριστές, σίγουρα θα ήταν χρήσιμο να έχουν και δυνατότητα διαγραφής πολλών χρηστών μαζί.
- 7) Η αναζήτηση θα μπορούσε να εκτελείται και σε ένα σύνολο από βιβλία
- 8) Θα πρέπει να βελτιωθεί και η λειτουργία της εκτύπωσης ώστε να υποστηρίζετε η εκτύπωση πολλαπλών αντιγράφων
- 9) Καλό θα κατά την δημιουργία ή εγκατάσταση ενός βιβλίου να υποστηρίζονται περισσότεροι τύποι συμπιεσμένων αρχείων όπως για παράδειγμα τα .rar αρχεία ή για το Linux τα .tar και .tar.gz.
- 10) Θα μπορούσε επίσης να δοκιμαστεί και κάποιο πρότυπο κρυπτογράφησης με δημόσιο κλειδί (όπως είναι για παράδειγμα ο αλγόριθμος RSA) και να μελετηθούν οι επιπτώσεις του στο υπολογιστικό κόστος της εφαρμογής.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

### Βιβλία

- Γκίσης Τραϊανός, Ζαφειρόπουλος Ιωάννης 2006, Ένα σύστημα προσαρμογής ηλεκτρονικού βιβλίου στις ανάγκες των χρηστών, ΑΤΕΙ Θεσσαλονίκης, Θεσσαλονίκη
- Cay S. Horstmann, and Gary Cornell Core 2004, Core Java™ 2 Volume I - Fundamentals, Seventh Edition, Prentice Hall PTR
- Cay S. Horstmann, and Gary Cornell Core 2004, Core Java™ 2 Volume II - Advanced Features, Seventh Edition, Prentice Hall PTR
- Marty Hall, and Larry Brown 2001, Core Web Programming (2<sup>nd</sup> Edition), Sun Microsystems INC California
- Reese George 2000, Database Programming with JDBC and Java 2nd Edition, O'Reilly & Associates, Inc. , USA
- Patrick Niemeyer & Joshua Peck 1997, Exploring Java, O'Reilly & Associates, Inc.
- Elliotte Rusty Harold 2006, Java™ I/O, 2nd Edition, O'Reilly, USA
- Clark S. Lindsey, Johnny S. Tolliver, and Thomas Lindblad 2005, JavaTech: An Introduction to Scientific and Technical Computing with Java, Cambridge University Press
- Herbert Schildt 2002, Java™ 2: The complete reference, Fifth Edition, The McGraw-Hill Companies, USA
- Lance Andersen 2006, JDBC™ 4.0 Specification, Sun Microsystems INC California
- Simon Bennett, Steve McRobb, and Ray Farmer 1999, Object – Oriented Systems Analysis and Design using UML, McGraw-Hill Publishing Company, Cambridge
- The PostgreSQL Global Development Group 2006, PostgreSQL 8.2.0 Documentation

### Άρθρα

- Vita Voom Software Team 2006, About PostgreSQL, <http://www.vitavoom.com/postgresql.html>
- BYTEA vs OID (Large Objects), <http://www.microolap.com/products/connectivity/postgresdac/help/TipsAndTricks/ByteaVsOid.htm>
- Qusay H. Mahmoud 2002, Compressing and Decompressing Data using Java,

<http://java.sun.com/developer/technicalArticles/Programming/compression/>

- Mandar Chitnis, Pravin Tiwari, & Lakshmi Ananthamurthy, Creating Use Case Diagrams,  
[http://www.developer.com/design/article.php/10925\\_2109801\\_2](http://www.developer.com/design/article.php/10925_2109801_2)
- Wikipedia 2007, e-book, <http://en.wikipedia.org/wiki/E-book>
- Charlotte Johnson 2002, eBook Advantages over pBooks,  
[http://faculty.mckendree.edu/william\\_harroff/ebe/ebook\\_advantages\\_over\\_pbooks.htm](http://faculty.mckendree.edu/william_harroff/ebe/ebook_advantages_over_pbooks.htm)
- Charlotte Johnson 2002, eBook Disadvantages Compared to pBooks,  
[http://faculty.mckendree.edu/william\\_harroff/ebe/ebook\\_disadvantage.htm](http://faculty.mckendree.edu/william_harroff/ebe/ebook_disadvantage.htm)
- Stephanie Ardito 2000, Electronic Books: To “E” or not to “E”; that is the question, <http://infotoday.com/searcher/apr00/ardito.htm>
- PostgreSQL Global Development Team 2006, PostgreSQL Advantages, <http://www.postgresql.org/about/advantages>
- Pradnya Choudhari, Web Angel Volunteer 2001, Java Advantages & Disadvantages, [http://arizonacommunity.com/articles/java\\_32001.shtml](http://arizonacommunity.com/articles/java_32001.shtml)
- *Robert Eckstein* 2007, Java SE Application Design With MVC,  
<http://java.sun.com/developer/technicalArticles/javase/mvc/index.html>
- Sun Microsystem Inc. 2003, Java™ 2 SDK Standard Edition Documentation, <http://java.sun.com/j2se/1.4.2/docs/>
- 2008, JDBC-Tutorial, <http://www.jdbc-tutorial.com/index.htm>
- Sun Microsystem Inc. 1998, Location-Independent Access to Resources,  
<http://java.sun.com/j2se/1.5.0/docs/guide/lang/resources.html#overview>
- Karan Dass 2006, Model-View-Controller (MVC) Architecture,  
<http://www.indiawebdevelopers.com/technology/java/mvcarchitecture.asp>
- eNode 2002, Model-View-Controller Pattern,  
<http://www.enode.com/x/markup/tutorial/mvc.html>
- The PostgreSQL Global Development Group 2002, PostgreSQL 7.3.20 Documentation,  
<http://www.postgresql.org/docs/7.3/interactive/index.html>

- MageLang Institute 1999, Short Course ,  
<http://java.sun.com/developer/onlineTraining/GUI/Swing1/shortcourse.html>
- Sun Microsystem Inc. 2007, The Java Tutorials,  
<http://java.sun.com/docs/books/tutorial/>
- Maydene Fisher 1999, The JDBC Tutorial: Chapter 3 - Advanced Tutorial, <http://java.sun.com/developer/Books/JDBCTutorial/index.html>
- Rags Srinivas 2003, Using AES with Java Technology,  
[http://java.sun.com/developer/technicalArticles/Security/AES/AES\\_v1.html](http://java.sun.com/developer/technicalArticles/Security/AES/AES_v1.html)
- Dickson College 2006, Writing a Bibliography (Harvard System),  
<http://dicksonc.act.edu.au/Library/bibliog.html>

#### Forums

- Dev Shed Forums, <http://forums.devshed.com/>
- Java Hellenic User Group, <http://www.jhug.gr/phpnuke/PHP-Nuke-6.5/html/index.php>
- Netbeans.org Forums, <http://www.nabble.com/Netbeans.org-f2602.html>
- Sun Java Technology Forums, <http://forum.java.sun.com/index.jspa>

*Σημείωση : Τελευταία προσπέλαση στις παραπάνω σελίδες έγινε στις 18/11/2007*

## ΠΑΡΑΡΤΗΜΑ 1

## Απαιτήσεις συστήματος

Ελάχιστες απαιτήσεις hardware

- ❖ Υπολογιστής Pentium 166MHz
- ❖ Ελεύθερος σκληρός δίσκος : 79 MB για την εγκατάσταση της Java(JRE ), 137 MB για την εγκατάσταση της PostgreSQL και 40 MB για την εγκατάσταση του προγράμματος (Take A Note 2.0). Συνολικά 256 MB
- ❖ 64 MB RAM

Ελάχιστες απαιτήσεις software

- ❖ Java Runtime Environment (JRE) 6  
(δωρεάν διανομή στο <http://java.sun.com/javase/downloads/index.jsp> )
- ❖ PostgreSQL 8.2  
(δωρεάν διανομή στο <http://www.postgresql.org/download/>)
- ❖ Λειτουργικό Σύστημα : Windows (NT, 2000, XP, Vista, 2003 Server Edition), Linux (συμβατό με τις περισσότερες διανομές), Sun Solaris

## Εγχειρίδιο τοπικής εγκατάστασης

Η τοπική εγκατάσταση χωρίζεται σε 4 βήματα :

- 1) Εγκατάσταση της Java
- 2) Εγκατάσταση της PostgreSQL – Δημιουργία εξυπηρετητή – Δημιουργία Βάσης – Εισαγωγή στοιχείων
- 3) Δημιουργία της γέφυρας JDBC
- 4) Εγκατάσταση της εφαρμογής (Take A Note 2.0)

Αν η εγκατάσταση γίνεται με σκοπό την διαδικτυακή λειτουργία του προγράμματος, η διαδικασία εγκατάστασης της PostgreSQL μπορεί να παραληφθεί.

### Διαδικασία εγκατάστασης :

1) Μπορείτε είτε να κατεβάσετε το πακέτο JRE 6 από εδώ ή να προμηθευτείτε την έκδοση του πακέτου εγκατάστασης της εφαρμογής Take A Note 2 που αντιστοιχεί στο λειτουργικό περιβάλλον που χρησιμοποιείτε, που περιλαμβάνει και το JRE πακέτο. Ακολουθήστε τα βήματα εγκατάστασης.

2) Εγκατάσταση PostgreSQL 8.1

Κατεβάστε από το διαδίκτυο, από την σελίδα <http://www.postgresql.org/download/> το αρχείο εγκατάστασης της PostgreSQL (προτείνεται η έκδοση 8.1). αφού αποσυμπιέσουμε το αρχείο postgresql-8.1.4-1.zip ακολουθούμε τα παρακάτω βήματα :

- ❖ Εκτελούμε το αρχείο postgresql-8.1.msi.
- ❖ Αφού εμφανιστεί ο οδηγός εγκατάστασης, επιλέγουμε την επιθυμητή γλώσσα εγκατάστασης και πατάμε Start .
- ❖ Στις δύο επόμενες οθόνες πατάμε απλά Next.
- ❖ Στην επόμενη οθόνη πατάμε το βελάκι δίπλα από κει που λέει PostgreSQL και επιλέγουμε Entire feature will be installed on local hard drive. Αλλάζουμε αν θέλουμε τον προκαθορισμένο κατάλογο εγκατάστασης και πατάμε Next.
- ❖ Στην επόμενη οθόνη, αφήνουμε ως έχει την επιλογή Install as a service, αν θέλουμε κάθε φορά που ανοίγει ο υπολογιστής να ξεκινάει και η PostgreSQL. Αν το από-επιλέξουμε, όποτε θέλουμε να ξεκινήσει η Postgres θα πρέπει να το κάνουμε χειρονακτικά, μέσω της επιλογής Start Service που θα εμφανιστεί μόλις ολοκληρωθεί η εγκατάσταση. Το Account name το ορίζουμε να είναι postgres, ενώ τον κωδικό μπορούμε να τον ορίσουμε όπως θέλουμε. Εισάγουμε τον επιθυμητό κωδικό στα πεδία Account Password και Verify Password και πατάμε Next.
- ❖ Στην επόμενη οθόνη ορίζουμε ως Locale το Greek , Greece , το encoding το θέτουμε στη τιμή ISO\_8859\_7, στο Superuser name βάζουμε pgadmin και τέλος στα πεδία Password και Password(again) βάζουμε την τιμή 1041983. Εάν εισαχθεί διαφορετική τιμή στο password, τότε η είσοδος μας στο πρόγραμμα δεν θα είναι δυνατή. Στις επόμενες οθόνες πατάμε Next.
- ❖ Μόλις ολοκληρωθεί η εγκατάσταση πατάμε Finish.

Η PostgreSQL είναι πλέον εγκατεστημένη στον υπολογιστή μας. Τώρα μπορούμε να προχωρήσουμε στην δημιουργία του εξυπηρετητή και της βάσης που θα χρησιμοποιεί το πρόγραμμά μας.

Ξεκινάμε το pgAdmin III (μπορείτε να το βρείτε στο All Programs του Start Menu κάτω από τον φάκελο PostgreSQL 8.1) και επιλέγουμε το πρώτο εργαλείο από τα αριστερά (Add a connection to a server). Ορίζουμε όλα τα πεδία στο παράθυρο σύμφωνα με την παρακάτω εικόνα :

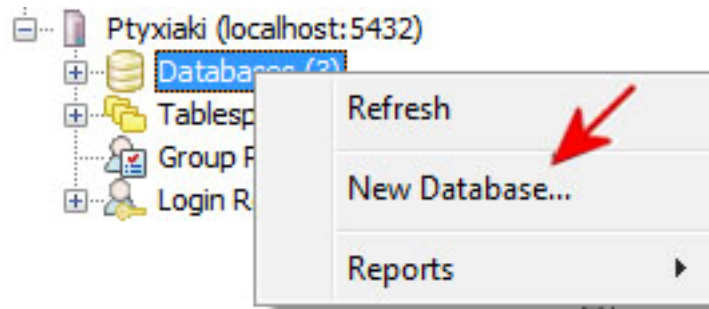


Εικόνα Π1 : Δημιουργία σύνδεσης σε εξυπηρετητή

Η εισαγωγή του ονόματος Ptyxiaki στο πεδίο Description είναι πολύ σημαντική γιατί το πρόγραμμα ψάχνει εξ ορισμού να συνδεθεί σε ένα εξυπηρετητή με αυτό το όνομα. Στο πεδίο με όνομα password βάζουμε τον κωδικό 1041983. Πατάμε OK. Μετά από αυτό είμαστε έτοιμοι να δημιουργήσουμε την βάση δεδομένων για το πρόγραμμά μας.

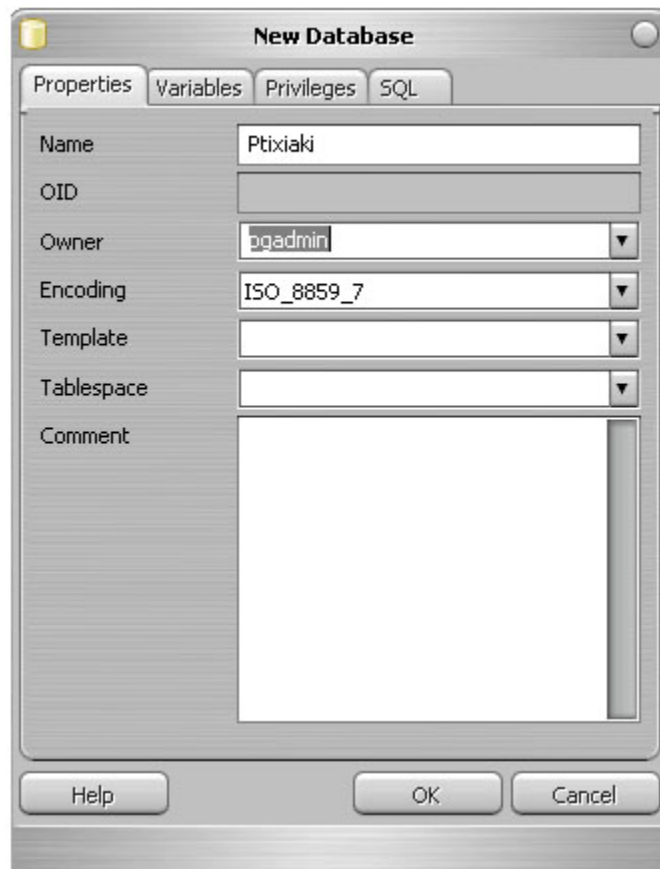
Έχοντας συνδεθεί ως pgadmin, κάνουμε δεξί κλικ στο αντικείμενο Databases που βρίσκεται στη δεξιά μεριά της οθόνης μας και επιλέγουμε New Database...





Εικόνα Π2 : Επιλογή για δημιουργία νέας βάσης

Στο πεδίο Name βάζουμε Ptxiaki και στο πεδίο Owner επιλέγουμε την τιμή pgadmin. Πατάμε το κουμπί OK για να δημιουργηθεί η βάση.



Εικόνα Π3 : Δημιουργία νέας βάσης

Τώρα, είναι ώρα να δημιουργήσουμε τους πίνακες μας. Επεκτείνουμε την ιεραρχία Ptxiaki -> Databases -> Ptxiaki -> Schemas -> public και επιλέγουμε το Tables. Από το μενού των εργαλείων που βρίσκεται στο πάνω μέρος της οθόνης μας, επιλέγουμε το εργαλείο με όνομα Execute arbitrary SQL queries και ένα νέο παράθυρο θα ανοίξει. Μέσα στο νέο παράθυρο, αντιγράψτε τα παρακάτω Create Tables :

```

CREATE TABLE users
(
username1 varchar(10) NOT NULL,
password1 bytea,
look_and_feel varchar DEFAULT
'com.sun.java.swing.plaf.windows.WindowsLookAndFeel':character varying,
theme varchar,
logged boolean DEFAULT false,
CONSTRAINT users_pkey PRIMARY KEY (username1)
);

```

```

CREATE TABLE administrators
(
username varchar (10) NOT NULL,
password1 bytea,
CONSTRAINT administrators_pkey PRIMARY KEY (username),
CONSTRAINT is_a_user FOREIGN KEY (username)
REFERENCES users (username1) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE CASCADE
);

```

```

CREATE TABLE books
(
bookname varchar (60) NOT NULL,
url varchar NOT NULL,
CONSTRAINT books_pkey PRIMARY KEY (bookname),
CONSTRAINT books_url_key UNIQUE (url)
);

```

```

CREATE TABLE bookmarks
(
username varchar (10) NOT NULL,
name varchar (50) NOT NULL,
link varchar NOT NULL,
book varchar ,
CONSTRAINT bookmarks_pkey PRIMARY KEY (username, name, link),
CONSTRAINT bookmark_in_book FOREIGN KEY (book)
REFERENCES books (url) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE CASCADE,
CONSTRAINT user_has_bookmarks FOREIGN KEY (username)
REFERENCES users (username1) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE CASCADE
);

```

```

CREATE TABLE highlights
(
username varchar (10) NOT NULL,
beg integer NOT NULL,
endh integer NOT NULL,

```

```

file varchar NOT NULL,
book varchar NOT NULL,
color integer,
CONSTRAINT highlights_pkey PRIMARY KEY (username, beg, endh, file, book),
CONSTRAINT highlight_in_book FOREIGN KEY (book)
REFERENCES books (url) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE CASCADE,
CONSTRAINT user_has_highlights FOREIGN KEY (username)
REFERENCES users (username1) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE CASCADE
);

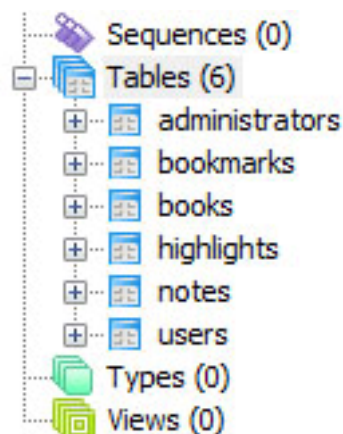
```

```

CREATE TABLE notes
(
username varchar (10) NOT NULL,
file varchar NOT NULL,
book varchar ,
usenotes bytea,
CONSTRAINT notes_pkey PRIMARY KEY (username, file),
CONSTRAINT note_in_book FOREIGN KEY (book)
REFERENCES books (url) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE CASCADE,
CONSTRAINT user_has_notes FOREIGN KEY (username)
REFERENCES users (username1) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE CASCADE
);

```

Αφού αντιγράψουμε τα Create Tables, πατάμε το κουμπί Execute Query που βρίσκεται στην γραμμή εργαλείων, για να εκτελεστούν οι εντολές. Κάνοντας Refresh στο σχήμα public θα πρέπει να έχετε μια εικόνα παρόμοια με την παρακάτω :



Εικόνα Π4 : Οι πίνακες που δημιουργήθηκαν

Σειρά έχει η εισαγωγή στοιχείων. Με τον ίδιο τρόπο που δημιουργήσαμε τους πίνακες θα εισάγουμε και τα απαραίτητα για τρέξει η εφαρμογή στοιχεία, στον

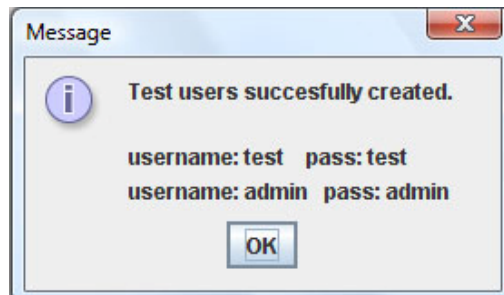
πίνακα books. Κάνουμε λοιπόν αντιγραφή/επικόλληση το παρακάτω στο παράθυρο εισαγωγής ερωτημάτων και το εκτελούμε :

```
INSERT INTO books VALUES ('The Java Tutorial','TheJavaTutorial/index.html');
```

Τώρα το μόνο που μένει είναι να δημιουργήσουμε τους δυο αρχικούς χρήστες του προγράμματος. Στο cd με τα έγγραφα της πτυχιακής, θα βρείτε και ένα φάκελο με όνομα CreateTestUsers. Αυτός ο φάκελος περιέχει συνολικά τρία αρχεία. Δυο αρχεία με επέκταση .class και ένα αρχείο με επέκταση .key. Αυτά είναι και τα μόνα απαραίτητα αρχεία για την δημιουργία των αρχικών χρηστών. Ανοίξτε τον Command Prompt (στα windows θα τον βρείτε στο Start -> All programs -> Accessories ), μεταβείτε στον εν λόγω φάκελο και εκτελέστε την εντολή :

```
java CreateTestUsers
```

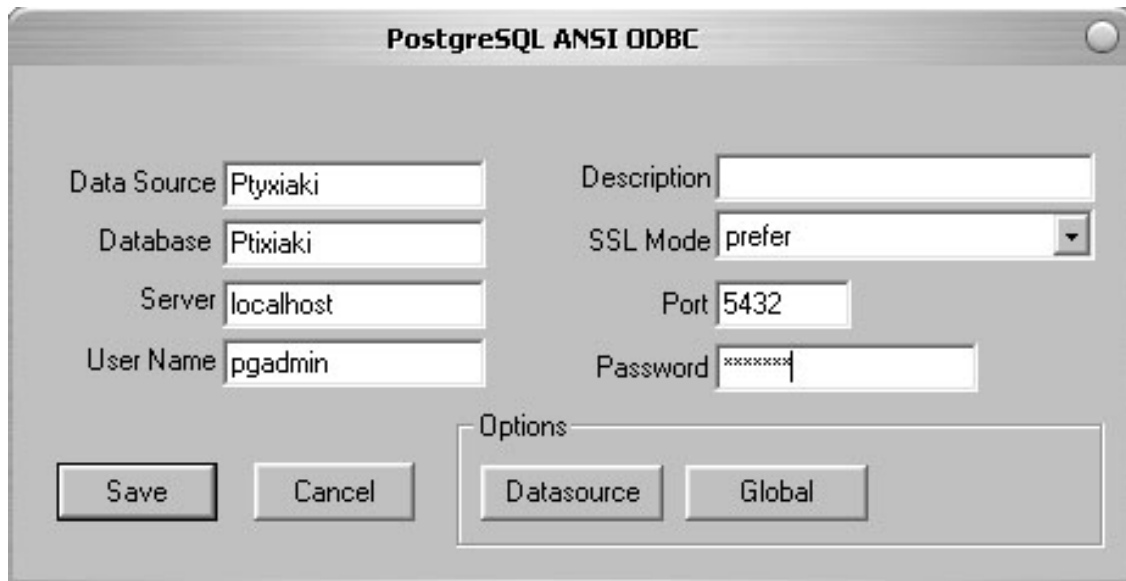
Αν η εισαγωγή έγινε επιτυχώς θα εμφανιστεί και το ανάλογο μήνυμα. Με αυτή την διαδικασία δημιουργήθηκαν δύο χρήστες για το σύστημα μας. Ένα χρήστης με δικαιώματα διαχειριστή, ο οποίος για username και password έχει admin, και ένας απλός χρήστης με username και password test.



Εικόνα Π5 : Το μήνυμα επιτυχούς δημιουργίας των νέων χρηστών

3) Επόμενο βήμα είναι η δημιουργία της JDBC γέφυρας, η οποία θα συνδέει το σύστημα με την βάση.

Σε περιβάλλον windows από τον πίνακα ελέγχου (Control Panel), επιλέγουμε Εργαλεία Διαχείρισης (Administrative Tools) και στη συνέχεια Πηγές Δεδομένων (Data Sources (ODBC)). Στην καρτέλα User DSN πατάμε προσθήκη (Add), επιλέγουμε PostgreSQL ANSI και πατάμε τέλος (Finish). Στο επόμενο παράθυρο που θα εμφανιστεί εισάγουμε τα στοιχεία όπως την παρακάτω εικόνα και στο πεδίο password εισάγουμε 1041983

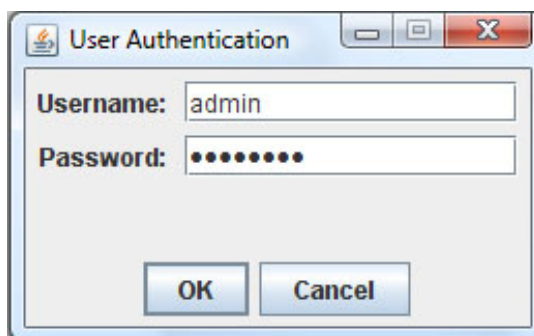


Εικόνα Π6 : Δημιουργία της γέφυρας

- 4) Τελευταίο βήμα είναι η εγκατάσταση του προγράμματος Take A Note 2.0. Κατεβάζουμε και εκτελούμε το αρχείο που ανταποκρίνεται στο λειτουργικό μας σύστημα. Επιλέγουμε την γλώσσα της προτίμησής μας και πατάμε Ok. Στην επόμενη οθόνη απλά πατάμε Next, ενώ στην επόμενη μπορούμε αν θέλουμε να αλλάξουμε τον προκαθορισμένο φάκελο εγκατάστασης. Πατάμε και πάλι Next. Από εδώ μπορούμε να ορίσουμε το που θα δημιουργηθούν εικονίδια. Επιλέγουμε που προτιμάμε να δημιουργηθούν οι συντομεύσεις και πατάμε Next και μετά Install. Το πρόγραμμα έχει πλέον εγκατασταθεί και είναι έτοιμο για χρήση.


## Εγχειρίδιο Χρήσης

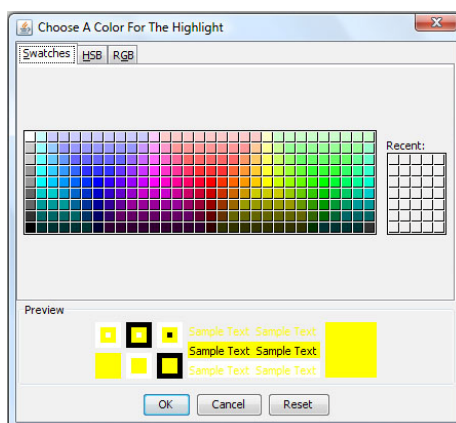
Προκειμένου να συνδεθείτε επιτυχώς στο πρόγραμμα θα πρέπει να έχετε στην κατοχή σας ένα προσωπικό username και password, το οποίο θα προμηθευτείτε από τον διαχειριστή του συστήματος. Εάν η εγκατάσταση είναι τοπική (έχετε δηλαδή στήσει την βάση δεδομένων στον προσωπικό σας υπολογιστή) , μπορείτε να συνδεθείτε χρησιμοποιώντας ένα από τους κωδικούς που δημιουργήθηκαν κατά την εγκατάσταση του προγράμματος (βλ. ενότητα [Εγχειρίδιο τοπικής εγκατάστασης](#)).



Αφού εισαχθούν τα στοιχεία επιτυχώς είσαστε έτοιμοι να χρησιμοποιήσετε το πρόγραμμα.

### Υπογραμμίσεις (Highlights)

Για να υπογραμμίσετε κάποιο σημείο του κειμένου, αφού επιλέξετε με το ποντίκι το επιθυμητό χωρίο, πατήστε το κουμπί , είτε μέσω το αναδυόμενου μενού κάνοντας δεξί κλικ κάπου μέσα στο κείμενο, είτε από την μπάρα εργαλείων που βρίσκετε στο πάνω μέρος της οθόνης είτε μέσω το μενού Tools. Και οι τρεις τρόποι έχουν το ίδιο αποτέλεσμα. Αφού πιάσετε το κουμπί, θα εμφανιστεί μία παλέτα χρωμάτων, από όπου μπορείτε να διαλέξετε το χρώμα με το οποίο το κείμενο θα υπογραμμιστεί.





Εικόνα Π7 : Η παλέτα με τα διαθέσιμα χρώματα

Την επιλογή του χρώματος μπορείτε να την κάνετε με τρεις διαφορετικούς τρόπους, επιλέγοντας κάποιο από τα αντίστοιχα tabs. Διαλέξτε τον τρόπο που σας βολεύει καλύτερα, επιλέξτε το χρώμα της αρεσκείας σας και πατήστε OK. Η υπογράμμιση σας δημιουργήθηκε.

### Διαγραφή υπογραμμίσεων


Σας δίνεται η δυνατότητα να διαγράψετε κάποιες από τις υπογραμμίσεις σας. Μπορείτε να διαγράψετε τις υπογραμμίσεις είτε της

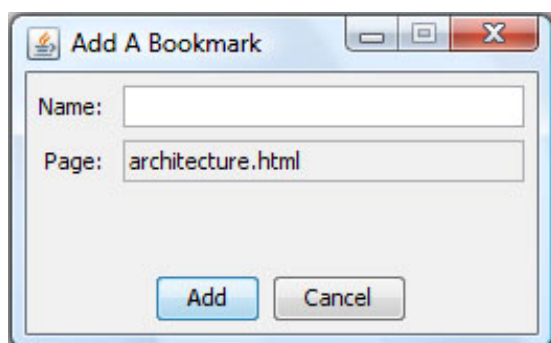
τρέχουσας σελίδας είτε όλου του βιβλίου. Πατώντας το κουμπί  διαγράφονται τα highlights τις σελίδας που έχετε ανοικτή, ενώ πατώντας το

πλήκτρο  και επιβεβαιώνοντας την απόφαση σας, θα διαγράψετε όλες τις υπογραμμίσεις που έχουν δημιουργηθεί στο βιβλίο που διαβάζετε. Από την στιγμή που διαγραφούν, δεν υπάρχει τρόπος ανάκτησής τους.

### Σελιδοδείκτες (Bookmarks)

Στο κάθε βιβλίο, μπορείτε να μαρκάρετε κάποιες (ή και όλες) από τις σελίδες, δημιουργώντας με αυτόν τον τρόπο σελιδοδείκτες, ώστε η πρόσβαση σας σε αυτές να είναι ταχύτερη και ευκολότερη. Το κουμπί δημιουργίας νέου

σελιδοδείκτη είναι το , και βρίσκετε σε τέσσερα σημεία. Στο αναδυόμενο μενού που εμφανίζεται όταν κάνετε δεξί κλικ οπουδήποτε μέσα στο κείμενο, στην μπάρα εργαλείων η οποία βρίσκετε στο πάνω μέρος της εφαρμογής, μέσα στο μενού με ονομασία Tools και τέλος στο πάνω αριστερό μέρος της οθόνης όπου και βρίσκετε το panel αφιερωμένο στους σελιδοδείκτες. Όποιον τρόπο και αν επιλέξετε, θα εμφανιστεί ένα νέο παράθυρο όπου μπορείτε να εισάγετε το όνομα που επιθυμείτε ο σελιδοδείκτης σας να έχει.



Εικόνα Π8 : Παράθυρο δημιουργίας νέου Bookmark

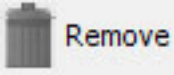
Αφού πληκτρολογήσετε το όνομα, πατήστε Add και ο σελιδοδείκτης θα δημιουργηθεί . Σε περίπτωση που ήδη έχετε βάλει σελιδοδείκτη στην συγκεκριμένη σελίδα το πρόγραμμα θα σας ενημερώσει με το ανάλογο μήνυμα λάθους. Εάν η δημιουργία ήταν επιτυχής, στο πάνω αριστερό μέρος της οθόνης θα εμφανιστεί ένα δένδρο με παρόμοια μορφή με το παρακάτω :

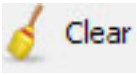


Εικόνα Π9 : Το δένδρο με τα Bookmarks

## Διαγραφή bookmark

Σε περίπτωση που επιθυμείτε να διαγράψετε κάποιον ή όλους του σελιδοδείκτες σας, κάντε μία από τις ακόλουθες ενέργειες :

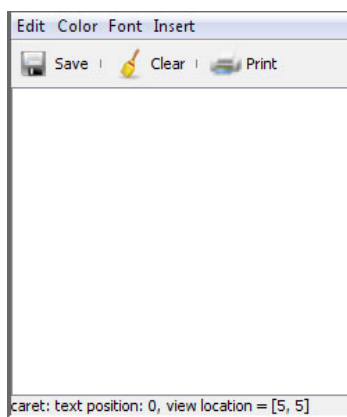
Εάν επιθυμείτε τη διαγραφή ενός και μόνο σελιδοδείκτη, αφού τον επιλέξετε, πατήστε το κουμπί  που βρίσκετε στο πάνω αριστερό μέρος της οθόνης σας.

Εάν επιθυμείτε την διαγραφή όλων των bookmark που έχετε δημιουργήσει πατήστε το κουμπί . Αφού επιβεβαιώσετε την επιλογή σας, οι σελιδοδείκτες του βιβλίου θα διαγραφούν.

Μετά την διαγραφή κάποιου σελιδοδείκτη δεν υπάρχει τρόπος επαναφοράς του.

## Σημειώσεις

Στο μεσαίο αριστερό panel της οθόνης σας βρίσκετε ο χώρος μέσα στον οποίο μπορείτε να εισάγετε τις προσωπικές σας σημειώσεις.



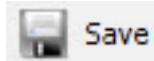
Εικόνα Π10 : Το panel των σημειώσεων

Το μόνο που έχετε να κάνετε είναι να πληκτρολογήσετε το κείμενο που θέλετε. Οι σημειώσεις σας θα αποθηκευτούν αυτόματα κατά την μετάβαση σας σε κάποια άλλη σελίδα. Μπορείτε επίσης να μορφοποιήσετε το κείμενο, αλλάζοντας για παράδειγμα το χρώμα το γραμμάτων ή την γραμματοσειρά μέσω των αντίστοιχων επιλογών που σας παρέχονται μέσα στα αντίστοιχα μενού. Ακόμη, εάν επιθυμείτε, μπορείτε να εισάγεται και κάποια εικόνα της αρεσκείας σας μέσω του μενού Insert. Να αναφερθεί εδώ πως τα αποδεκτά format των εικόνων για εισαγωγή στις σημειώσεις είναι τα εξής : .jpeg, .jpg,



.gif, .tiff, .tif και .png. Ό,τι μορφοποίηση και αν κάνετε στις σημειώσεις, αυτές θα αποθηκευτούν και θα ανακτηθούν αυτούσιες.

Στην τρέχουσα έκδοση του προγράμματος, υπάρχουν και τρία επιπλέον κουμπιά που σχετίζονται με τις σημειώσεις. Με το κουμπί



Save

αποθηκεύονται στην βάση οι σημειώσεις σας, με το κουμπί



Clear

διαγράφονται όλες οι σημειώσεις και τέλος με το κουμπί

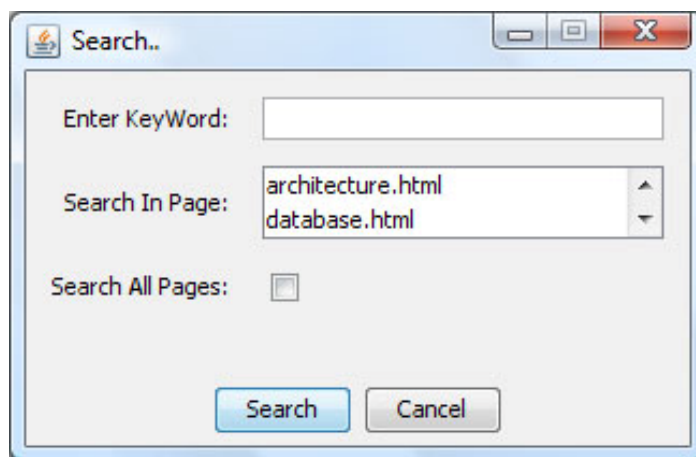


Print

μπορείτε να εκτυπώσετε τις σημειώσεις της τρέχουσας σελίδας. Η εκτύπωση θα έχει ακριβώς την ίδια μορφή με αυτήν που βλέπετε μέσα από το πρόγραμμα. Αυτό σημαίνει ότι μαζί με το μορφοποιημένο κείμενο θα εκτυπωθούν και οι όποιες εικόνες έχετε εισάγει.

### Αναζήτηση (Search)

Η λειτουργία της αναζήτησης είναι προσπελάσιμη είτε μέσω του μενού Tools είτε από το κουμπί με τίτλο New Search στο κάτω μέρος της οθόνης (συντόμευση Alt+S). Όταν ξεκινήσετε μια νέα αναζήτηση το ακόλουθο παράθυρο θα εμφανιστεί :



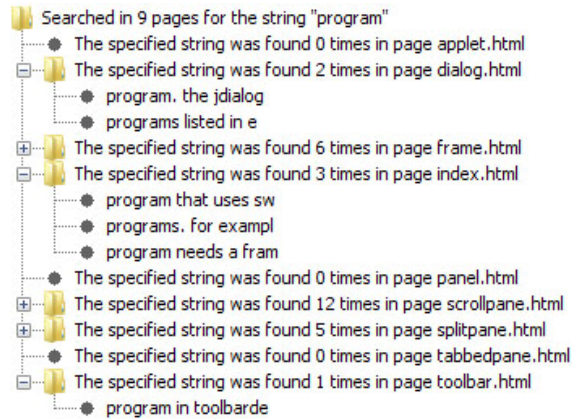
Εικόνα Π11 : Το παράθυρο από το οποίο μπορείτε να εκτελέσετε μια αναζήτηση

Από εδώ μπορείτε να ψάξετε για κάποια λέξη ή φράση, ή σε μία μόνο σελίδα του βιβλίου ή σε ολόκληρο το βιβλίο.

Εισάγετε την λέξη για την οποία θέλετε να ψάξετε στο πεδίο που φέρει το όνομα Enter KeyWord. Δεν έχει σημασία εάν την γράψετε με κεφαλαία ή με μικρά γράμματα αφού η αναζήτηση δεν είναι case sensitive. Σε περίπτωση που ψάχνετε με ελληνικούς χαρακτήρες θα πρέπει να συμπεριλάβετε και τους τόνους στις λέξεις σας, διαφορετικά η αναζήτηση δεν θα σας επιστρέψει τα επιθυμητά αποτελέσματα. Επόμενο βήμα είναι να επιλέξετε το πού θέλετε να ψάξετε. Στο πεδίο Search In Page εμφανίζονται όλες οι σελίδες του βιβλίου που διαβάζετε. Επιλέξτε την επιθυμητή και πατήστε το κουμπί Search. Εάν η αναζήτηση θέλετε να γίνει σε ολόκληρο το βιβλίο επιλέξτε το κουτάκι με όνομα

Search All Pages. Σε περίπτωση που έχετε επιλέξει κάποια μεμονωμένη σελίδα και τσεκάρετε και το κουτί Search All Pages η αναζήτηση θα εκτελεστεί σε όλο το βιβλίο.

Τα αποτελέσματα της αναζήτησης θα εμφανιστούν με μορφή δέντρου στο κάτω μέρος της οθόνης σας ταξινομημένα ανά σελίδα και θα έχουν μορφή παρόμοια με την παρακάτω εικόνα :

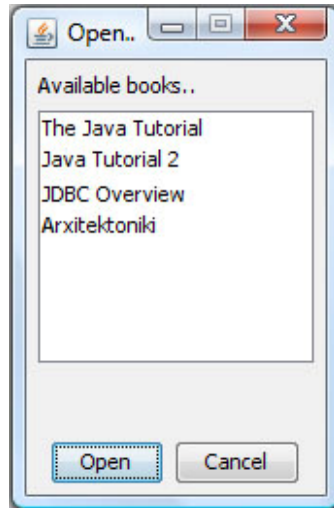


Εικόνα Π12 : Τα αποτελέσματα μιας αναζήτησης

Επιλέγοντας κάποιο από τα αποτελέσματα (μαρκαρισμένα στο δέντρο με βούλα) θα μεταφερθείτε στην αντίστοιχη σελίδα και στο σημείο που βρέθηκε η λέξη. Η λέξη θα είναι μαρκαρισμένη με γκρι χρώμα μαζί με τους επόμενους 20 χαρακτήρες από αυτήν. Για να καθαρίσετε την περιοχή των αποτελεσμάτων πατήστε το πλήκτρο με όνομα Clear Search στην κάτω μεριά του προγράμματος. Αν επιχειρήσετε νέα αναζήτηση χωρίς να έχετε καθαρίσει τα αποτελέσματα, αυτά θα καθαριστούν αυτόματα.

### Άνοιγμα βιβλίου

Μέσω του προγράμματος έχετε πρόσβαση σε πολλά ξεχωριστά βιβλία. Αν επιθυμείτε να ανοίξετε κάποια από τα υπάρχοντα από το μενού File επιλέξτε Open. Μία λίστα με τα διαθέσιμα βιβλία θα εμφανιστεί.



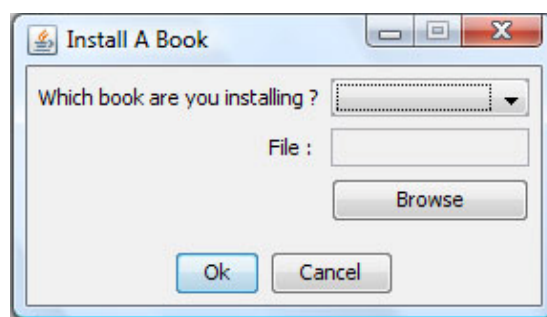
Εικόνα Π13 : Η λίστα με τα διαθέσιμα βιβλία

Επιλέξτε ένα από αυτά και πατήστε Open. Το βιβλίο θα ανοίξει και το πρόγραμμα θα φορτώσει αυτόματα τις σημειώσεις, τις υπογραμμίσεις και τους σελιδοδείκτες που έχετε κάνει για αυτό το βιβλίο.

Σε περίπτωση που δουλεύετε το πρόγραμμα δικτυακά, κάποιο από τα εμφανιζόμενα βιβλία ενδέχεται να μην είναι διαθέσιμο. Θα πρέπει να το κατεβάσετε και να το εγκαταστήσετε ο ίδιος. Οι οδηγίες εγκατάστασης ενός βιβλίου ακολουθούν.

### Εγκατάσταση βιβλίου

Αφού προμηθευτείτε το αρχείο zip από τον διαχειριστή του προγράμματος, που περιέχει το βιβλίο που θέλετε να εγκαταστήσετε επιλέξτε την λειτουργία Install a book που βρίσκεται μέσα στο μενού Tools. Ένα νέο παράθυρο θα εμφανιστεί :



Εικόνα Π14 : Το παράθυρο εγκατάστασης νέου βιβλίου

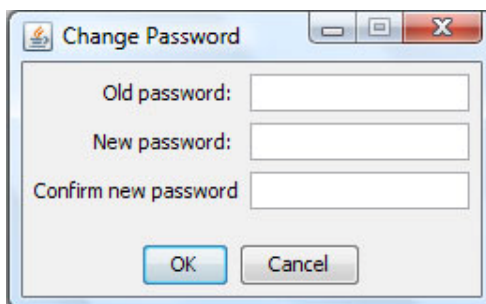
Από το drop down menu επιλέξτε το όνομα του βιβλίου που πρόκειται να εγκαταστήσετε. Μέσω του κουμπιού Browse ψάξτε στα αρχεία του υπολογιστή σας, βρείτε τα αρχείο που προμηθευτήκατε από τον διαχειριστή και ανοίξτε το. Πατήστε Ok. Θα σας ζητηθεί να επιβεβαιώσετε τα στοιχεία που εισήγατε. Αν είστε σίγουροι ότι είναι τα σωστά πιάστε και πάλι Ok. Το νέο βιβλίο εγκαταστάθηκε και είναι έτοιμο για χρήση.

### Σύνδεση σαν διαχειριστής

Αν εκτός από το όνομα και κωδικό απλού χρήστη έχετε κωδικό και όνομα που αντιστοιχούν σε διαχειριστή του συστήματος, μπορείτε μέσα από το πρόγραμμα να συνδεθείτε με αυτά τα στοιχεία δίχως να χρειαστεί να το κλείσετε και να το ξανανοίξετε. Από το μενού File επιλέξτε Login As Administrator. Εισάγεται τα στοιχεία που αντιστοιχούν σε διαχειριστή και πατήστε Ok. Προσέξτε ότι ένα νέο μενού προστέθηκε με όνομα Admin Tools.

### Αλλαγή κωδικού

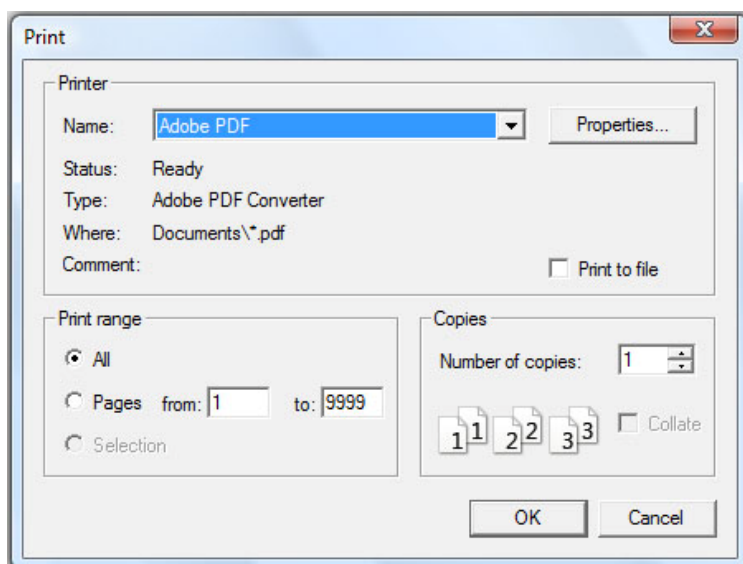
Αν επιθυμείτε να αλλάξετε τον προσωπικό σας κωδικό, μέσα από το μενού File επιλέξτε Change your password. Στο παράθυρο που θα εμφανιστεί, στο πρώτο πεδίο εισάγεται τον τρέχων κωδικό σας και στα δυο επόμενα τον επιθυμητό καινούριο κωδικό. Αν τα πεδία New password και Confirm new password συμφωνούν, ο κωδικός σας θα αλλάξει επιτυχώς.



Εικόνα Π15 : Από εδώ μπορείτε να αλλάξετε τον προσωπικό σας κωδικό

### Εκτύπωση κειμένου

Για να εκτυπώσετε το κείμενο της τρέχουσας σελίδας από το μενού File επιλέξτε Print. Στο παράθυρο που θα εμφανιστεί, διαλέξτε τον εκτυπωτή με τον οποίο θέλετε να εκτυπώσετε και πατήστε Ok. Μπορείτε ακόμη εάν θέλετε να αλλάξετε και τις άλλες παραμέτρους της εκτύπωσης όπως είναι ο προσανατολισμός ή το μέγεθος της σελίδας.




Εικόνα Π16 : Το παράθυρο της εκτύπωσης

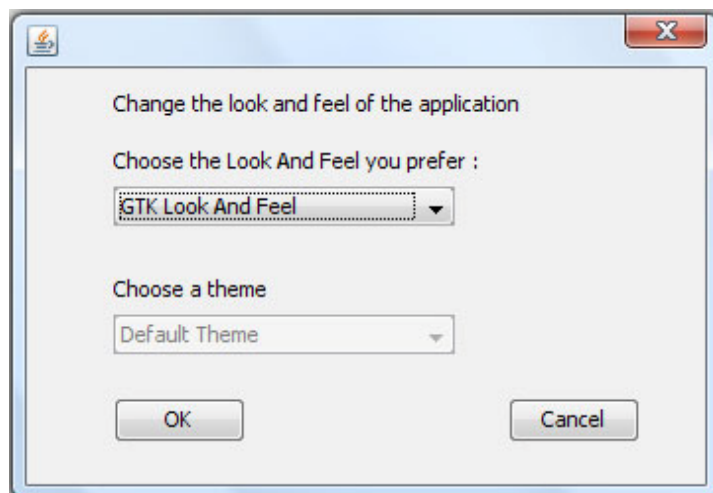
Κάθε φορά, μπορείτε να εκτυπώσετε μόνο την τρέχουσα σελίδα. Επίσης, στην παρούσα έκδοση του προγράμματος, δεν είναι διαθέσιμη η επιλογή εκτύπωσης πολλαπλών αντιγράφων.

### Αλλαγή της εμφάνισης του προγράμματος

Έχετε την δυνατότητα να επιλέξετε την εμφάνιση που θέλετε να έχει το πρόγραμμα σας από μία λίστα με συνολικά πέντε διαφορετικές εμφανίσεις. Για να δείτε τις διαθέσιμες εμφανίσεις και να διαλέξετε μία, μέσα από το μενού Tools επιλέξτε Change look and feel. Από το πρώτο drop down menu διαλέξτε την εμφάνιση που προτιμάτε. Εάν διαλέξετε την εμφάνιση με όνομα Cross Platform Look And Feel, από τα δεύτερο drop down menu το οποίο πλέον θα είναι ενεργοποιημένο, μπορείτε να διαλέξετε και το θέμα που προτιμάτε για την συγκεκριμένη εμφάνιση. Η εμφάνιση και το θέμα της επιλογής σας θα αποθηκευτούν και θα φορτώνεται κάθε φορά που εκκινείτε το πρόγραμμα.

**ΣΗΜΕΙΩΣΗ.** Κάποιες από τις εμφανίσεις ενδέχεται να μην είναι άμεσα διαθέσιμες. Αν η εμφάνιση που επιλέξατε δεν ενεργοποιήθηκε μόλις πατήσατε Ok επανεκκινήστε το πρόγραμμα.

**ΣΗΜΕΙΩΣΗ 2.** Μόλις αλλάξετε την εμφάνιση οι υπογραμμίσεις σας δεν θα φαίνονται. Πατήστε απλά το κουμπί ανανέωσης  που βρίσκεται στην μπάρα εργαλείων στο πάνω μέρος της οθόνης και αυτές θα επανέλθουν κανονικά.



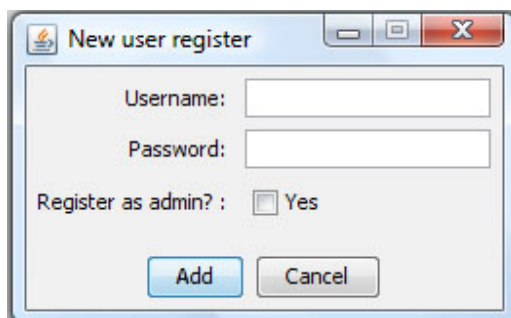
Εικόνα Π17 : Το μενού για αλλαγή της εμφάνισης και του θέματος

### Λειτουργίες διαθέσιμες μόνο στους διαχειριστές του προγράμματος

Αν συνδεθείτε με δικαιώματα διαχειριστή θα έχετε διαθέσιμες επιπλέον τις παρακάτω λειτουργίες οι οποίες είναι όλες προσπελάσιμες από το μενού Admin Tools :

#### Δημιουργία νέου χρήστη

Για να δημιουργήσετε ένα νέο λογαριασμό επιλέξτε New User Register και ένα νέο παράθυρο θα εμφανιστεί.



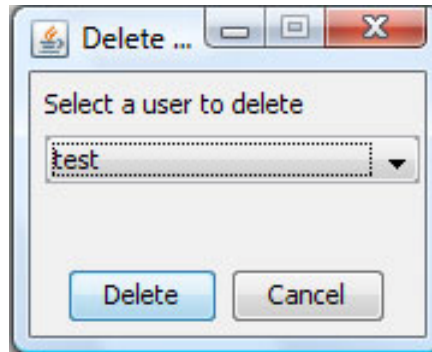
Εικόνα Π18 : Το παράθυρο για εγγραφή νέου χρήστη

Στον πρώτο πεδίο εισάγετε το όνομα που θέλετε ο νέος χρήστης να έχει και στο δεύτερο τον κωδικό του. Σε περίπτωση που θέλετε ο νέος χρήστης να έχει δικαιώματα διαχειριστή επιλέξτε το κουτί με όνομα Register as admin? . Πατήστε Add. Ο νέος χρήστης μόλις δημιουργήθηκε.

#### Διαγραφή χρήστη

Για να διαγράψετε κάποιον χρήστη του προγράμματος επιλέξτε την λειτουργία Delete User. Από το drop down menu βρείτε το όνομα του προς

διαγραφή χρήστη, επιλέξτε το και πατήστε Delete. Επιβεβαιώστε την διαγραφή.

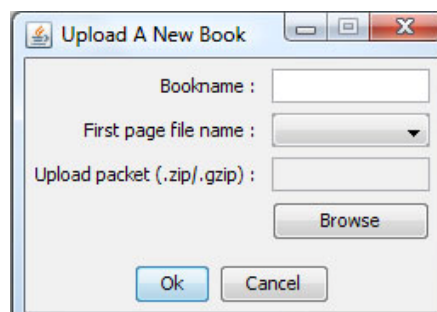


Εικόνα Π19 : Το παράθυρο διαγραφής χρήστη

### Δημιουργία νέου βιβλίου

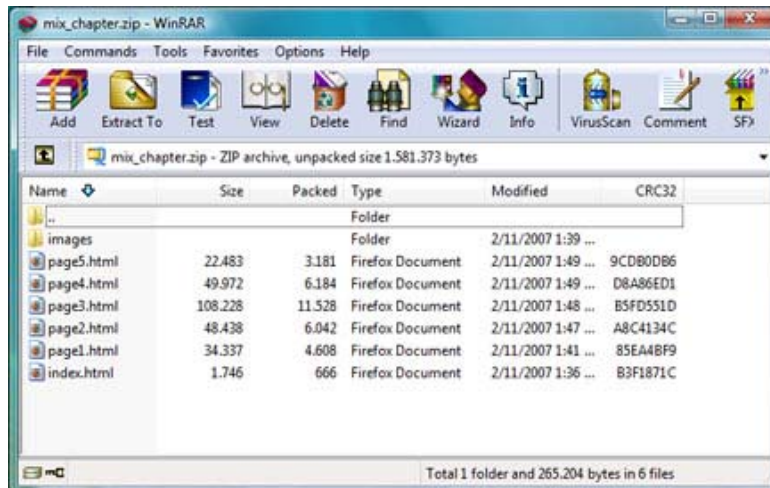
Προκειμένου να δημιουργήσετε ένα νέο βιβλίο ακολουθήστε τα εξής βήματα :

- 1) Από το μενού Admin Tools επιλέξτε New Book
- 2) Στο πρώτο πεδίο εισάγεται το όνομα που το καινούριο βιβλίο θέλετε να έχει
- 3) Πατήστε Browse ώστε να εντοπίσετε το zip αρχείο που περιέχει το βιβλίο. Αφού το βρείτε πατήστε Ok
- 4) Στο drop down menu θα εμφανιστούν όλα τα html αρχεία που περιέχει το zip. Επιλέξτε εκείνο που θέλετε να ορίσετε ως πρώτη σελίδα του βιβλίου
- 5) Πατήστε Ok και το νέο βιβλίο θα δημιουργηθεί



Εικόνα Π20 : Το παράθυρο για την δημιουργία νέου βιβλίου

**ΠΡΟΣΟΧΗ** Το αρχείο zip πρέπει να έχει συγκεκριμένη μορφή. Το βιβλίο δεν θα πρέπει να περιέχεται μέσα σε κάποιον φάκελο. Ένα παράδειγμα της μορφής που πρέπει να έχει το zip είναι το παρακάτω :



Εικόνα Π21 : Παράδειγμα της δομής που πρέπει να έχει το zip αρχείο

Όπως φαίνεται και στην εικόνα όλες οι σελίδες του βιβλίου (τα html αρχεία) θα πρέπει να βρίσκονται στο πρώτο επίπεδο και όχι μέσα σε κάποιο φάκελο. Η ύπαρξη υποφακέλων στους οποίους μπορεί να περιέχονται για παράδειγμα οι εικόνες, είναι επιτρεπτή.





ΑΛΕΞΑΝΔΡΕΙΟ  
ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΘΕΣΣΑΛΟΝΙΚΗΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Πτυχιακή Εργασία

# ΣΥΣΤΗΜΑ ΑΝΑΓΝΩΣΗΣ - ΔΙΑΧΕΙΡΙΣΗΣ ΗΛΕΚΤΡΟΝΙΚΩΝ ΒΙΒΛΙΩΝ



Ανδριώτης Απόστολος  
Επιβλέπων Καθηγητής : Σαμαράς Απόστολος



# Εισαγωγή

Προηγούμενο σύστημα :

- Ικανοποιητική προσέγγιση στο θέμα ανάγνωσης – διαχείρισης ηλεκτρονικών βιβλίων
- Επαρκής κάλυψη των βασικών αναγκών του χρήστη



# Σκοπός της παρούσας εργασίας

Επέκταση και βελτίωση του συστήματος

- Προσθήκη νέων λειτουργιών
- Βελτίωση των ήδη υπαρχόντων

Στόχος μας η κάλυψη περισσότερων απαιτήσεων και αναγκών του χρήστη ώστε να δημιουργηθεί ένα ολοκληρωμένο και φιλικό προς τον χρήστη σύστημα

# Μεθοδολογία εύρεσης των νέων λειτουργιών

- Προσωπική εκτίμηση κατόπιν δοκιμής του συστήματος
- Προτάσεις που διατύπωσε η προηγούμενη ομάδα ανάπτυξης
- Επισκόπηση λογισμικών ανάγνωσης e-book
  - ❖ Microsoft Reader
  - ❖ Adobe Reader



# Take A Note 1.0

Λειτουργίες που προσέφερε :

- Highlight – Υπογράμμιση χωρίων του κειμένου με κίτρινο χρώμα
- Bookmarks – Προσομοίωση της λειτουργίας των συνδετήρων.
- Notes – Δυνατότητα καταγραφής προσωπικών σχολίων
- New User Register – Δυνατότητα δημιουργίας νέων χρηστών από τον διαχειριστή

# Microsoft Reader

## Βασικές λειτουργίες :

- ❖ Add Highlight – Υπογράμμιση λέξης/πρότασης με κίτρινο χρώμα
- ❖ Add Bookmark – Μαρκάρισμα της τρέχουσας σελίδας.
- ❖ Add Text Note – Εισαγωγή προσωπικών σχολίων στην επιλεγμένη γραμμή.
- ❖ Add Drawing - Αντίστοιχο εργαλείο με το Pencil του MS Paint

## Επιπλέον λειτουργίες :

- ❖ Text-to-Speech – Δυνατότητα να ακούσει ο χρήστης το κείμενο που βρίσκεται στην οθόνη του
- ❖ Dictionaries – Προσθήκη λεξικών σε διάφορες γλώσσες

# Adobe Reader

## Βασικές λειτουργίες :

- Add note to text – Εισαγωγή σχολίων. Πλήρης συσχέτιση κειμένου και σημειώσεων
- Underline text – Υπογράμμιση της λέξη με κίτρινο χρώμα και εισαγωγή σχολίων εάν το επιθυμεί ο χρήστης
- Cross out text – Παρόμοιο εργαλείο με το προηγούμενο με την διαφορά ότι η επιλεγμένη λέξη γίνεται strike through με κόκκινο χρώμα.
- Add bookmark – Προσθήκη σελιδοδείκτη στην σελίδα
- Search – Αναζήτηση λέξης ή φράσης
- Zoom in/out - Δυνατότητα για Zoom in/out επί του κειμένου
- Create link – Μετατροπή λέξης/πρότασης σε υπέρ-σύνδεσμο

# Σύγκριση συστημάτων

- Microsoft Reader :
  - Καλύπτει τις βασικές απαιτήσεις του χρήστη με εξαίρεση τα επιπλέον πακέτα
  - Μειονέκτημα αποτελεί ο τρόπος υλοποίησης των σελιδοδεικτών
- Adobe Reader :
  - Από τις καλύτερες υλοποιήσεις εισαγωγής σχολίων
  - Έχει χαρακτηριστικά τα οποία δεν κρίνονται απαραίτητα (για παράδειγμα το cross-out text)
- Κύρια μειονεκτήματα του προς ανάπτυξη συστήματος :
  - Δεν υποστηρίζει πολλαπλά βιβλία
  - Δεν έχει λειτουργία αναζήτησης



# Βελτίωση λειτουργιών

- Λειτουργία υπογράμμισης :
  - Δυνατότητα επιλογής χρώματος
- Λειτουργία προσωπικών σημειώσεων :
  - Δυνατότητα αποθήκευσης της μορφοποίησης
  - Δυνατότητα εισαγωγής εικόνας
  - Εκτύπωση σημειώσεων



# Νέες λειτουργίες

Δύο κατηγορίες :

- Λειτουργίες διαθέσιμες σε όλους τους χρήστες του συστήματος
- Λειτουργίες διαθέσιμες μόνο στους διαχειριστές



# Λειτουργίες διαθέσιμες στους διαχειριστές

- Πρόσβαση των λειτουργιών από μενού διαθέσιμο μόνο στους διαχειριστές
- Στόχος είναι ο διαχειριστής να μην χρειάζεται να παρεμβαίνει απευθείας στην βάση δεδομένων

## Λειτουργίες :

- Δημιουργία νέου χρήστη
- Διαγραφή υπάρχοντος χρήστη
- Δημιουργία νέου βιβλίου

# Λειτουργίες διαθέσιμες σε όλους τους χρήστες

- Άνοιγμα βιβλίου
- Αναζήτηση
- Εκτύπωση κειμένου
- Δυνατότητα αλλαγής της εμφάνισης της εφαρμογής
- Αλλαγή κωδικού
- Λειτουργία σύνδεσης ως διαχειριστής
- Εγκατάσταση βιβλίου

## Επιπλέον χαρακτηριστικά

- Κρυπτογράφηση των κωδικών των χρηστών με χρήση του προτύπου AES(Advanced Encryption Standard)
- Χρήση ανακυκλώσιμων συνδέσεων (Connection Pool)
- Διασφάλιση ακεραιότητας χρηστών. Με τον ίδιο λογαριασμό δεν μπορούν να είναι συνδεδεμένοι δύο χρήστες ταυτόχρονα

# Τρόπος υλοποίησης

- Κάθε χρήστης έχει προσωπικό λογαριασμό
- Οι αλλαγές που εκτελεί ο κάθε χρήστης αποθηκεύονται στον αντίστοιχο πίνακα της βάσης (οι υπογραμμίσεις στον πίνακα Highlights, οι σημειώσεις στον πίνακα Notes κλπ)
- Τα διαθέσιμα προς τους χρήστες βιβλία είναι καταχωρημένα στον πίνακα Books
- Αν ο χρήστης εκτός του πίνακα Users είναι καταχωρημένος και στον πίνακα Administrators έχει δικαιώματα διαχείρισης



# Δημιουργία εκτελέσιμου αρχείου

- Σε περίπτωση διαδικτυακής λειτουργίας του προγράμματος το μόνο που πρέπει να κάνει ο χρήστης είναι να διαλέξει και να εκτελέσει το install αρχείο που αντιστοιχεί στο λειτουργικό του

# Συμπεράσματα

- Το σύστημα μπορεί να καλύψει αρκετές από τις ανάγκες ενός χρήστη λογισμικού ανάγνωσης ηλεκτρονικών βιβλίων
- Μπορεί να χρησιμοποιηθεί αποτελεσματικά από κάποιο εκπαιδευτικό ίδρυμα, για την διάθεση επιπλέον συγγραμμάτων



# Ερωτήσεις

