

Ανώτατο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Θεσσαλονίκης



HEAL-Link v3.0

Φοιτητής  
Ασμανίδης Υπάτιος

Εισηγητής  
Δέρβος Δημήτρης

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η εργασία αυτή εκπονήθηκε κατά το Ακαδημαϊκό έτος 2005-2006 στα πλαίσια της πτυχιακής εργασίας του τμήματος πληροφορικής του Α.Τ.Ε.Ι Θεσσαλονίκης.

Θα ήθελα να ευχαριστήσω την κα. Κλωντίνη Δέρβου, τον κ. Βλάση Χατζησταύρου και τον κ. Πισπιρίγγα Λεωνίδα για τις οδηγίες, παρατηρήσεις, επισημάνσεις κατά την κατασκευή της νέας έκδοσης του HEAL-Link.

Ιδιαίτερα θα ήθελα να ευχαριστήσω τον κ. Δημήτρη Δέρβο για την υπομονή του, καθώς και την πολύτιμη βοήθεια και καθοδήγηση σε όλα τα τμήματα της κατασκευής την νέας έκδοσης.

Με εκτίμηση,  
Υπάτιος

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>ΚΕΦΑΛΑΙΟ 1<sup>ο</sup> : ΠΕΡΙΓΡΑΦΗ, ΑΝΤΙΚΕΙΜΕΝΟ, ΑΠΑΙΤΗΣΕΙΣ</b>	<b>4</b>
<b>1.1 ΠΕΡΙΓΡΑΦΗ</b>	<b>4</b>
<b>1.2 ΑΝΤΙΚΕΙΜΕΝΟ</b>	<b>4</b>
1.2.1 Στόχος	4
1.2.2 Το υπάρχον σύστημα	5
<b>1.3 ΑΠΑΙΤΗΣΕΙΣ</b>	<b>5</b>
1.3.1 Εφαρμογή	5
1.3.2 Sessions	5
1.3.3 Υπηρεσίες Κεντρικής Πύλης	6
1.3.4 Εξατομίκευση Πύλης	8
<b>ΚΕΦΑΛΑΙΟ 2<sup>ο</sup> : ΤΕΧΝΟΛΟΓΙΑ</b>	<b>12</b>
<b>2.1 ΕΙΣΑΓΩΓΗ</b>	<b>12</b>
<b>2.2 Application Server</b>	<b>13</b>
<b>2.3 Application servers vs Web Servers</b>	<b>15</b>
<b>2.4 Jakarta Tomcat</b>	<b>17</b>
<b>2.5 Εγκατάσταση Tomcat σε περιβάλλον windows</b>	<b>18</b>
<b>2.6 Εγκατάσταση Tomcat σε περιβάλλον Linux</b>	<b>23</b>
<b>2.7 Γλώσσα Σεναρίου JSP ( Java Server Pages )</b>	<b>25</b>
2.7.1 Γενικά	25
2.7.2 Σύνταξη JSP	26
2.7.3 JSP και SQL	29
<b>2.8 Σύστημα Διαχείρισης Βάσεων Δεδομένων</b>	<b>32</b>
<b>ΚΕΦΑΛΑΙΟ 3ο : Η ΕΦΑΡΜΟΓΗ</b>	<b>33</b>
<b>3.1 Εισαγωγή στα Sessions</b>	<b>33</b>
3.1.1 Αναζήτηση – Δημιουργία συνόδου	35
3.1.2 Ανάκτηση πληροφορίας από σύνοδο	35
3.1.3 Αποθήκευση δεδομένων στην σύνοδο	37
<b>3.2 Διαχείριση Sessions στο HealLink</b>	<b>37</b>
<b>3.3 Βάση Δεδομένων του HealLink</b>	<b>48</b>
<b>3.4 Παράδειγμα από το HEAL-Link v3.0</b>	<b>52</b>
3.4.1 loginuser.jsp	52

3.4.2 dbcon.java	60
<b>ΚΕΦΑΛΑΙΟ 4<sup>ο</sup> : ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ</b>	<b>64</b>
4.1 Κεντρική Πύλη	64
4.1.1 Περιβάλλον	64
4.1.2 Πλοήγηση	65
4.1.3 Υπηρεσίες Αναζήτησης	72
4.1.4 Νέα του HEAL-Link	73
4.1.5 MyHEAL-Link	74
4.2 Εξατομίκευση Πύλης	76
4.2.1 Είσοδος	76
4.2.2 Επεξεργασία Προφίλ	77
4.2.3 Ενημέρωση για νέα περιοδικά	78
4.2.4 Αλλαγή προσωπικών στοιχείων	79
4.2.5 Λεπτομέρειες Συνδέσεων	80
<b>ΚΕΦΑΛΑΙΟ 5<sup>ο</sup> : ΠΡΟΤΑΣΕΙΣ</b>	<b>81</b>
5.1 Τεχνολογία	81
5.2 Εφαρμογή και υπηρεσίες	82
5.2.1 Αλλαγή συνθηματικού	82
5.2.2 Forum	82
5.2.3 Περισσότερες επιλογές διαμόρφωσης προφίλ	83
<b>Βιβλιογραφία</b>	<b>84</b>

## **ΚΕΦΑΛΑΙΟ 1<sup>Ο</sup> : ΠΕΡΙΓΡΑΦΗ, ΑΝΤΙΚΕΙΜΕΝΟ, ΑΠΑΙΤΗΣΕΙΣ**

### **1.1 ΠΕΡΙΓΡΑΦΗ**

Ο κόμβος HEAL-Link (<http://www.heal-link.gr>) του δικτύου των Ελληνικών Ακαδημαϊκών Βιβλιοθηκών αποτελεί το κύριο σημείο πρόσβασης της ακαδημαϊκής κοινότητας της Ελλάδας και της Κύπρου, σε ένα σύνολο ηλεκτρονικών εκδόσεων, μιας πληθώρας διεθνών επιστημονικών τεχνικών περιοδικών.

### **1.2 ΑΝΤΙΚΕΙΜΕΝΟ**

#### **1.2.1 Στόχος**

Η εξέλιξη της διεπαφής του **HEAL-Link** μέσω διαδικτύου, ώστε αυτή να αποκτήσει υφή πύλης (portal), δηλαδή, ενός κεντρικού σημείου εκκίνησης και πρόσβασης σε διάφορες υπηρεσίες και πληροφορίες. Αυτού του είδους η εξέλιξη προαπαιτεί την σύνδεση του περιβάλλοντος με βάση δεδομένων ώστε να καταστεί δυνατή η μετατροπή των στατικών του σελίδων σε δυναμικού τύπου. Η σύνδεση με βάση δεδομένων σημαίνει και επιπλέον δυνατότητα αυτόματης καταγραφής δεδομένων που αφορούν τη χρήση του περιβάλλοντος, δεδομένα τα οποία μπορούν να υποστούν στατιστική επεξεργασία στη συνέχεια να χρησιμοποιούνται για την εξαγωγή σημαντικών συμπερασμάτων όσον αφορά τη χρήση του κόμβου.

Ο στόχος της νέας έκδοσης συγκεντρώνεται τόσο στην καλύτερη απόδοση της Πύλης σε σενάρια υπερφόρτωσης, όσο και στην καλύτερη διαχείριση των συνόδων των χρηστών. Η δημιουργία ενός συστήματος που να προβλέπει και να καταγράφει όλα τα πιθανά σενάρια συμπεριφοράς των χρηστών, ώστε να οδηγεί στην εξαγωγή στατιστικών στοιχείων που βρίσκονται πιο κοντά στα γεγονότα της πύλης. Η λειτουργία αυτού του συστήματος θα πρέπει να είναι σταθερή και αποδοτική, ώστε να προσφέρει τις υπηρεσίες του με τον καλύτερο δυνατό τρόπο και να εγγυάται την ακεραιότητα των συναλλαγών.

### **1.2.2 Το υπάρχον σύστημα**

Το υπάρχον σύστημα παρέχει πλοήγηση στους τίτλους με πολλούς τρόπους καθώς και μια μηχανή αναζήτησης στους εκδοτικούς οίκους των περιοδικών. Το όλο σύστημα στηρίζεται στην τεχνολογία php που πραγματοποιεί συνδέσεις με την βάση δεδομένων του HEAL-Link που χρησιμοποιεί το RDBMS DB2 v8.1 της IBM. Φιλοξενείται σε έναν διακομιστή διαδικτύου που χρησιμοποιεί το λογισμικό Apache. Η διαχείριση των συνόδων γίνεται μόνο σε επίπεδο χρηστών οι οποίοι έχουν πραγματοποιήσει εισαγωγή στο σύστημα και δεν ελέγχει περιπτώσεις μη ομαλής απομάκρυνσης του χρήστη από το υπο περιβάλλον του.

### **1.3. ΑΠΑΙΤΗΣΕΙΣ**

Έπειτα από συζητήσεις με το αρμόδιο προσωπικό της Βιβλιοθήκης, των τμημάτων Φυσικής και Πληροφορικής του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης, αποφασίστηκαν οι παρακάτω απαιτήσεις, για την πύλη HEAL-Link v3.0

#### **1.3.1 Εφαρμογή**

Η εφαρμογή θα φιλοξενηθεί στον διακομιστή εφαρμογών Jakarta-Tomcat με στόχο την βελτίωση της ταχύτητας και απόδοσης του συστήματος, καθώς και την σταθερότητα της εφαρμογής, αφαιρώντας ένα μεγάλο κομμάτι φόρτου εργασίας από τον διακομιστή εφαρμογών. Η μεταφορά αυτή συνεπάγεται και την αλλαγή της γλώσσας σεναρίου από php σε jsp ( java server pages).

#### **1.3.2 Sessions**

Η διαχείριση των συνόδων (sessions) θα πρέπει να προβλέπει και να καταγράφει όλα τα πιθανά σενάρια συμπεριφοράς των χρηστών, όπως απομάκρυνση χωρίς αποσύνδεση από το υπο περιβάλλον του, επισκέψεις σε περιοδικά σαν επισκέπτης και μετέπειτα εισαγωγή στο σύστημα κ.α.

Η υλοποίηση της διαχείρισης θα πραγματοποιηθεί μέσω των πακέτων που προσφέρουν οι τεχνολογίες JSP και Servlet.

### 1.3.3 Υπηρεσίες Κεντρικής Πύλης

Οι υπηρεσίες που προσφέρει η κεντρική πύλη της έκδοσης 3.0 περιλαμβάνουν και υποστηρίζουν πλήρως τις υπηρεσίες την έκδοσης 2.0. Έτσι υπάρχουν πολλαπλοί τρόποι αναζήτησης ηλεκτρονικών περιοδικών, στήλη με τα νέα του HEAL-Link, πρόσβαση στην υπηρεσία εξατομίκευσης MyHEAL-Link, οδηγίες χρήσης της πύλης.

#### Αναζήτηση περιοδικών με 4 εναλλακτικούς τρόπους

##### 1. Από έναν Αλφαβητικό Κατάλογο

Ο αλφαβητικός κατάλογος θα περιέχει **τίτλους των περιοδικών** οπότε ο χρήστης θα αναζητά περιοδικά με βάση το πρώτο γράμμα του τίτλου τους.

##### 2. Από την ιεραρχία των Θεματικών Κατηγοριών

Ο χρήστης θα επιλέγει **Κατηγορία -> Υποκατηγορία -> Θεματικό Όρο** και θα του εμφανίζονται οι σχετικοί τίτλοι περιοδικών, οι οποίοι θα είναι ταυτόχρονα και links στους αντίστοιχους δικτυακούς τόπους των εκδοτών, όπου θα γίνεται διαθέσιμος ο **Πίνακας Περιεχομένων** τους.

##### 3. HEAL-Link Search

Θα ενσωματωθεί η προϋπάρχου μηχανή μετα-αναζήτησης, μέσω της οποίας ο χρήστης μπορεί να αναζητήσει ένα ή περισσότερα **άρθρα περιοδικών** χρησιμοποιώντας μία ή δύο λέξεις-κλειδιά και ένα από τα λογικά συνεκτικά OR και AND. Ο χρήστης έχει τη δυνατότητα να περιορίζει τον στόχο της αναζήτησης, καθορίζοντας τους εκδότες για τα περιοδικά των οποίων ενδιαφέρεται. Αυτό το τελευταίο επιτυγχάνεται επιλέγοντας το κυκλάκι 'Ναι' (για αποκλεισμό: το κυκλάκι 'Όχι') στα δεξιά του ονόματος του αντίστοιχου εκδότη.

#### 4. Γρήγορη Αναζήτηση

Αποτελεί νέο τρόπο αναζήτησης ηλεκτρονικών περιοδικών στην πύλη. Η αναζήτηση πραγματοποιείται με ερώτημα της μηχανής στην βάση που διατηρεί η δικτυακή πύλη. Ο χρήστης εισάγει τις λέξεις – κλειδιά που επιθυμεί και η μηχανή πραγματοποιεί ερώτημα στη βάση, αναζητώντας τις λέξεις κλειδιά. Τα αποτελέσματα της αναζήτησης είναι κατηγοριοποιημένα, γεγονός που δίνει τη δυνατότητα στον χρήστη να πραγματοποιήσει πλοήγηση στα περιεχόμενα που τον ενδιαφέρουν.

#### **Νέα του HEAL-Link**

Θα εμφανίζονται αρχικά 3 τίτλοι-σύνδεσμοι Νέων οι οποίοι θα οδηγούν στην εμφάνιση ολόκληρων των αντίστοιχων κειμένων.

#### **Πρόσβαση στο MyHEAL-Link**

Θα υπάρχει ένας χώρος για τις απαραίτητες διαδικασίες για την πρόσβαση του χρήστη στο υπό-περιβάλλον εξατομίκευσης του HEAL-Link. Θα ζητείται το e-mail και ο κωδικός του χρήστη προκειμένου να έχει πρόσβαση στο MyHEAL-Link. Το MyHEAL-Link θα είναι e-mail controlled, δηλαδή, θα ελέγχεται αν ανήκει το e-mail σε κάποιο από τα εγγεγραμμένα στο HEAL-Link ιδρύματα και αν όχι δεν θα γίνεται αποδεκτό. Επίσης αν ο χρήστης δεν χρησιμοποιήσει το λογαριασμό του για 6 μήνες, αυτός θα κλείνει και θα πρέπει να δημιουργήσει καινούριο. Το τελευταίο θα γίνεται για να μην υπάρχουν χρήστες που έχουν πάψει να χρησιμοποιούν το λογαριασμό τους στο HEAL-Link.

#### Εγγραφή Νέου Μέλους

Σε ένα ηλεκτρονικό έντυπο αίτησης (e-form), θα ζητούνται τα στοιχεία του χρήστη, τα οποία θα πρέπει να συμπληρωθούν όλα. Τα στοιχεία αυτά θα είναι το ονοματεπώνυμό του, το ίδρυμα προέλευσης, η ιδιότητά του χρήστη στο συγκεκριμένο ίδρυμα, και απαραίτητως το e-mail του. Μόλις δώσει τα στοιχεία του



θα λαμβάνει με e-mail το password για να κάνει login στο σύστημα. Εφόσον κάνει login στο MyHEAL-Link του θα μπορεί να διαμορφώνει το προφίλ του.

### Ανάκτηση Κωδικού Πρόσβασης

Για την περίπτωση που θα ξεχάσει ή θα χάσει ο χρήστης τον κωδικό του, θα ζητείται το e-mail του όπου και θα του αποστέλλεται ο κωδικός πρόσβασής του.

### **Οδηγίες**

Πρόκειται για έναν συνοπτικό οδηγό λειτουργίας της πύλης που είναι διαθέσιμος στον χρήστη σε ηλεκτρονική μορφή.

#### **1.3.4 Εξατομίκευση της Πύλης**

Όταν θα μπει ο χρήστης για πρώτη φορά στο περιβάλλον MyHEAL-Link δεν θα υπάρχει τίποτα προεπιλεγμένο. Για να αρχίσει να διαμορφώνει το προφίλ του θα πρέπει να επιλέξει *Επεξεργασία Προφίλ* από κάποιο μενού. Ο όρος **Προφίλ** στην συγκεκριμένη περίπτωση, σημαίνει, την επιλογή των περιοδικών που θα εμφανίζονται στο χρήστη όταν θα μπει στο λογαριασμό του, και την επιλογή των θεματικών όρων που τον ενδιαφέρουν ώστε να ενημερώνεται για αφίξεις νέων περιοδικών σχετικών με τα ενδιαφέροντα του. Η επιλογή περιοδικών θα γίνεται μέσω του αλφαβητικού καταλόγου και της ιεραρχίας των θεματικών κατηγοριών καθώς και από την μηχανή γρήγορης αναζήτησης της κεντρικής πύλης που ενσωματώνεται και στο υπό περιβάλλον του χρήστη, και η επιλογή θεματικών όρων από την ιεραρχία των θεματικών κατηγοριών και μέσω της επιλογής «Αυτόματη Ειδοποίηση για Νέα Περιοδικά»

## **Επεξεργασία Προφίλ**

Οι 3 τρόποι με τους οποίους θα γίνεται η δημιουργία και ενημέρωση του προφίλ του χρήστη περιγράφονται ως εξής :

### Αλφαβητικός Κατάλογος

Ο χρήστης επιλέγει περιοδικά με βάση τον τίτλο τους από έναν αλφαβητικό κατάλογο. Εάν επιθυμεί να ενημερώνεται για νέους τίτλους του ενδιαφέροντος του, θα έχει τη δυνατότητα απευθείας πρόσβασης στους Θεματικούς Όρους και επιλογής εξ' αυτών.

### Μέσω της ιεραρχίας των Θεματικών Κατηγοριών

Ο χρήστης επιλέγει Κατηγορία -> Υποκατηγορία -> Θεματικούς Όρους, όπου και του παρουσιάζονται οι θεματικοί όροι τους οποίους τσεκάρει εάν επιθυμεί να ειδοποιείται για νέους σχετικούς τίτλους και οι τίτλοι και επιλέγει αυτούς που τον ενδιαφέρουν. Στην ενημέρωση προφίλ θα εμφανίζονται σκιασμένες οι διαδρομές που είχε ακολουθήσει ο χρήστης για να επιλέξει τα περιοδικά του ενδιαφέροντος του.

### Αυτόματη Ειδοποίηση για νέα περιοδικά

Η επιλογή αυτή να είναι παρόμοια με αυτή της ιεραρχίας, με τη διαφορά ότι ο χρήστης θα φτάνει μόνο μέχρι το επίπεδο των θεματικών όρων, τους οποίους και θα επιλέγει για την ενημέρωση του για αφίξεις νέων περιοδικών σχετικών με αυτούς.

### Γρήγορη μηχανή αναζήτησης

Τα αποτελέσματα της γρήγορης αναζήτησης παρουσιάζονται κατηγοριοποιημένα στον χρήστη. Οι διάφορες κατηγορίες των αποτελεσμάτων είναι

- Κατηγορίες
- Υποκατηγορίες
- Θεματικοί Όροι
- Ηλεκτρονικά Περιοδικά

Ο χρήστης μπορεί να επιλέξει ηλεκτρονικά περιοδικά από τα αποτελέσματα της γρήγορης αναζήτησης και να τα προσθέσει στο προφίλ του ή να πλοηγηθεί στις διάφορες κατηγορίες, υποκατηγορίες και θεματικούς όρους και να επιλέξει από εκεί τα ηλεκτρονικά περιοδικά που επιθυμεί να υπάρχουν στο προφίλ του.

### **Αλλαγή Προσωπικών Στοιχείων**

Ο χρήστης θα πρέπει να έχει τη δυνατότητα αλλαγής των προσωπικών του στοιχείων, για την περίπτωση αλλαγής της κατάστασής του ( από Φοιτητής σε Καθηγητής ή από Καθηγητής μιας βαθμίδας σε κάποιας ανώτερης).

### **Αυτόματη Ειδοποίηση για Νέα Περιοδικά**

Όταν θα υπάρχουν νέα περιοδικά σχετικά με τους θεματικούς όρους που έχει επιλέξει για να ενημερώνεται, μπαίνοντας ο χρήστης στο λογαριασμό του, θα εμφανίζεται ένα pop-up παράθυρο, όπου θα εμφανίζονται με link όποια νέα περιοδικά μπήκαν στους θεματικούς όρους που τον ενδιαφέρουν. Από αυτό το pop-up παράθυρο θα μπορεί να επισκεφτεί τα εν λόγω νέα περιοδικά ένα-ένα. Ακολουθώντας το link τους θα ανοίγει νέο παράθυρο, από όπου θα μπορεί να τα ελέγξει., και εφόσον κάποιο ή κάποια τον ενδιαφέρουν, θα μπορεί να επανέλθει στο pop-up παράθυρο (το οποίο θα παραμένει ανοιχτό) , και να τσεκάρει ότι τον ενδιαφέρει.

Εδώ πρέπει να πούμε ότι, αριστερά των links των περιοδικών, θα υπάρχουν δυο στήλες τσεκαρίσματος. «ΝΑΙ» και «ΟΧΙ». Θα μπορεί να τσεκάρει μία μόνο από τις δυο ή καμιά :

- Όσα επιλέξει να τον ενδιαφέρουν θα μπουν στο προφίλ του
- Όσα επιλέξει να μην τον ενδιαφέρουν δεν θα εμφανιστούν ξανά σε pop-up παράθυρο, και
- Όσα δεν επιλέξει σε καμιά στήλη, σημαίνει ότι δεν έχει αποφασίσει ακόμα (ή δεν τα έχει ελέγξει), οπότε και θα επανεμφανίζονται σε pop-up παράθυρο, στο επόμενο login. Αυτές οι επανεμφανίσεις θα γίνονται μέχρις ότου το σύστημα πάψει (αυτόματα) να θεωρεί τα εν λόγω περιοδικά ότι είναι 'νέα' (2 μήνες

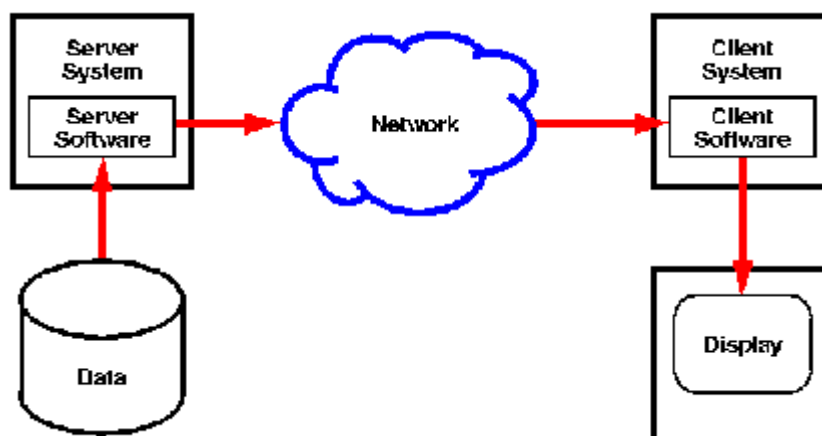
αφότου καταχωρηθεί η άφιξή τους στο σύστημα – το διάστημα αυτό θα το καθορίζει ο διαχειριστής της εφαρμογής).

## ΚΕΦΑΛΑΙΟ 2<sup>ο</sup> : ΤΕΧΝΟΛΟΓΙΑ

### 2.1 ΕΙΣΑΓΩΓΗ

Η μεγάλη ανάπτυξη της τεχνολογίας στις μέρες μας, η εξάπλωση του internet και οι μεγάλες ταχύτητες που προσφέρονται σε πολύ χαμηλές τιμές, έχουν σαν συνέπεια την αύξηση των απαιτήσεων των υπολογιστικών συστημάτων που παίζουν τον ρόλο των διακομιστών. Το ηλεκτρονικό εμπόριο έχει μπει πλέον για τα καλά στη ζωή μας και οι καθημερινές ηλεκτρονικές συναλλαγές είναι πλέον γεγονός. Εκτός από το ηλεκτρονικό εμπόριο, το internet αποτελεί μια πηγή ανεξάντλητων πόρων και δεδομένων για οποιοδήποτε θέμα, γεγονός που το καθιστά πολύ ελκυστικό σε πολλούς χρήστες.

Τα λεγόμενα portals ανάλογα με το περιεχόμενό τους αποτελούν πηγή και οδηγός για πληροφορίες σε διάφορα θέματα που φιλοξενούν. Σαν αποτέλεσμα έχουμε την αυξημένη απαίτηση εξυπηρέτησης από τους πελάτες (clients) που ζητούν κάποια πληροφορία από τους διακομιστές (servers)



Εικόνα 2.1 Client/Server Τεχνολογία

Σαν αποτέλεσμα έχουμε την υπερφόρτωση των διακομιστών που οδηγεί πολλές φορές σε αδυναμία να εξυπηρετήσει τους πελάτες. Για αυτό χρειάζεται εκτός από μεγάλη υπολογιστική ισχύ, καλύτερη διαχείριση από πλευράς λογισμικού ώστε

να μπορεί ο διακομιστής να εξυπηρετήσει τους πελάτες με την μεγαλύτερη δυνατή απόδοση. Και σε αυτό ακριβώς το σημείο θα κάνουμε εισαγωγή στους Application Servers ή διακομιστές εφαρμογών και στην διαφορά τους με τους web servers.

## 2.2 Application Server

Οι διακομιστές εφαρμογών είναι εφοδιασμένοι με τα κατάλληλα εργαλεία, εφαρμογές και τεχνολογίες ώστε να προσφέρουν την δυνατότητα στις επιχειρήσεις να αναπτύσσουν και να διαχειρίζονται έναν μεγάλο αριθμό από εφαρμογές. Ο τομέας των διακομιστών εφαρμογών έχει χαρακτηριστεί σαν από τους πιο σπουδαίους αναπτυσσόμενους τομείς στην τεχνολογία του διαδικτύου.

Από την πλευρά του σχεδιαστή, η κεντρική αλλαγή που φέρνουν οι διακομιστές εφαρμογών είναι ο διαχωρισμός της επιχειρηματικής λογικής από την λογική παρουσίασης και την λογική των βάσεων δεδομένων. Στην ουσία οι διακομιστές διαδικτύων βοηθούν στην δημιουργία εφαρμογών τριών επιπέδων όπου η βάση δεδομένων είναι λογικά διαχωρισμένη από την επιχειρηματική λογική (πολλές φορές είναι και φυσικά διαχωρισμένη από την επιχειρηματική λογική).

Η ανάπτυξη ενός διαχειριστή συνεπάγεται την εμπλοκή με πολύπλοκα ζητήματα όπως ο έλεγχος ταυτοχρονισμού, η εξασφάλιση όλων των πιθανών προϊόντων των βάσεων δεδομένων, την διαχείριση των συνδέσεων του δικτύου, database pooling, υποστήριξη κληρονομικότητας, υποστήριξη clustering, προστασία αποτυχίας, επεκτασιμότητα της εφαρμογής και της επίδοσής της.

Οι διακομιστές εφαρμογών μπορούν να απλοποιήσουν τα πιο πάνω ζητήματα για την διαδικασία της ανάπτυξης της εφαρμογής, φροντίζοντας για τα περισσότερα, αν όχι για όλα, επιτρέποντας στους σχεδιαστές της εφαρμογής να συγκεντρωθούν στον σκοπό για τον οποίο δημιουργήθηκε η ανάγκη της εφαρμογής, την επιχειρηματική ανάγκη.

Σε αυτό το σημείο θα αναφερθούμε στα χαρακτηριστικά ενός διακομιστή εφαρμογών.

- Υποστήριξη τόσο για επιχειρηματική λογική όσο και για λογική εφαρμογών. Τυπικά αυτό υλοποιείται με τεχνολογικά συστατικά όπως COM, CORBA και Java – RMI, εφαρμοσμένα σε ένα πλαίσιο εργασίας Microsoft Windows DNA ή Enterprise JavaBeans.
- Πολυνηματικές προσβάσεις και Database connection pooling. Αυτά είναι δυο πολύ μεγάλα σχεδιαστικά ζητήματα που έχουν να αντιμετωπίσουν οι σχεδιαστές των αναπτυσσόμενων εφαρμογών. Ο έλεγχος ταυτοχρονισμού δεν αποτελεί ζήτημα για τους σχεδιαστές.
- Δημιουργία αντικειμένων COM και CORBA. Αυτό είναι πιθανότατα ένα χαρακτηριστικό που θα πρέπει να εκμεταλλευτεί, εκτός κι αν η επιχείρηση έχει αποφασίσει στην ανάπτυξη με ένα συγκεκριμένο μοντέλο.
- Clustering Support, ισορροπία υπερφόρτωσης και χαρακτηριστικά failover.
- Ακεραιότητα συναλλαγών
- Συνδεσιμότητα σε παλαιότερα συστήματα συναλλαγών και βάσεως δεδομένων
- Ασφαλής συναλλαγές και υποστήριξη για SSL/HTTPS

Αν και τα παραπάνω αποτελούν ένα πολύ καλό μικρό σύνολο των χαρακτηριστικών, υπάρχουν κι άλλα που μπορεί να ενδιαφέρουν όπως

- Δυνατότητα ανάπτυξης στο διαδίκτυο και εξυπηρέτηση των χρηστών μέσω του internet, intranet και Extranet
- Παρακολούθηση διαδικασίας συναλλαγών. Αυτή η δυνατότητα διατίθεται από μεγάλες εταιρίες όπως BEA Systems και IBM
- Χαρακτηριστικά για την βελτίωση της διαδικασίας επεξεργασίας και συναλλαγών
- Υποστήριξη για την συνεργασία με άλλους διακομιστές εφαρμογών
- Ενσωμάτωση με εργαλεία ανάπτυξης και με χαρακτηριστικά προσανατολισμένα στην ταχύτερη ανάπτυξη.

### 2.3 Application servers vs Web Servers

Το κύριο μέλημα ενός web server είναι να σερβίρει web pages στους φυλλομετρητές (browsers). Οι περισσότεροι web servers μπορούν επίσης να δεχτούν και να επεξεργαστούν δεδομένα από τους χρήστες, να μορφοποιήσουν δεδομένα, να πραγματοποιήσουν αυθεντικοποίηση και την παροχή ασφαλείας. Επίσης μπορούν να εκτελέσουν scripts όπως perl JavaScript και VBScript.

Από την άλλη μεριά οι διακομιστές εφαρμογών προετοιμάζουν το υλικό για τον web server, δηλαδή περισυλλέγει πληροφορίες και δεδομένα από μια βάση δεδομένων, πραγματοποιεί ελέγχους ασφαλείας, διαχειρίζεται τις συνόδους των χρηστών κ.α.

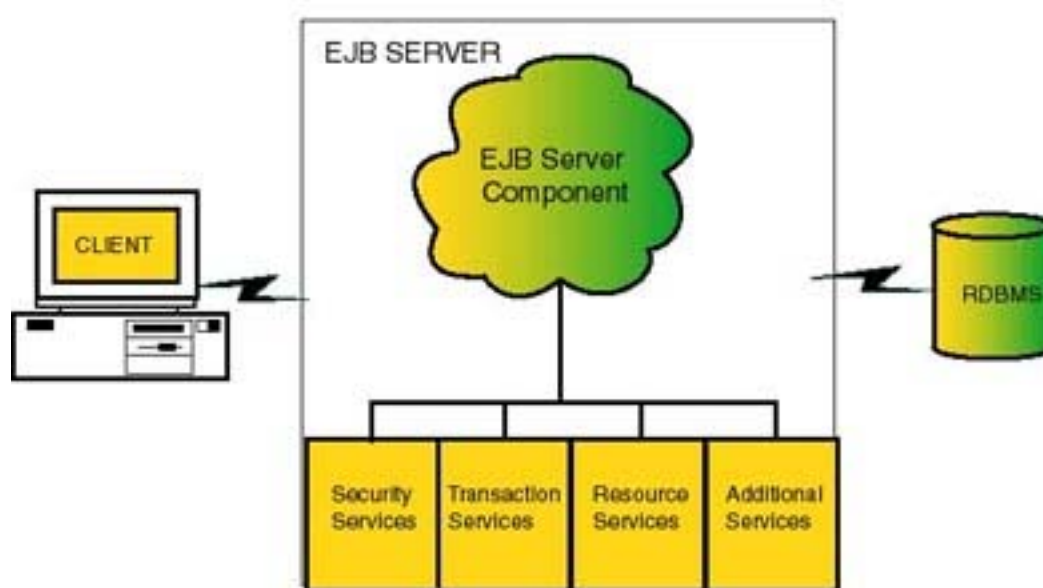
Σαν αποτέλεσμα έχουμε την μείωση του φόρτου εργασίας στους web servers λόγω του ότι ένα μεγάλο κομμάτι επεξεργασίας πραγματοποιείται από τον application server. Πολλοί χρήστες σημαίνει πολλές αιτήσεις εξυπηρέτησης, πολλούς ελέγχους ασφαλείας, πολλές συναλλαγές και επομένως ο web server υπερφορτώνεται τόσο πολύ που μπορεί να τον οδηγήσει σε DoS ( Denial Of Service) κατάσταση. Βέβαια μπορεί μοιραστεί το βάρος και σε άλλους web servers αλλά προτείνεται να χρησιμοποιηθεί ένας application server γι' αυτό το σκοπό. Οι web servers δεν σχεδιάστηκαν για να διαχειρίζονται μεγάλες ποσότητες δεδομένων και οι σχεδιαστές των διακομιστών αυτών κατά καιρούς χρειάστηκε να χρησιμοποιήσουν διάφορες τεχνικές και επιπρόσθετα προϊόντα ώστε να λύσουν τέτοιου είδους προβλήματα με διάφορα επίπεδα επιτυχίας. Συμπερασματικά αν μια εφαρμογή διαχειρίζεται μεγάλο όγκο δεδομένων ή τα δεδομένα είναι πολύπλοκα κατά κάποιο τρόπο, τότε ο διακομιστής εφαρμογών δεν είναι μόνο πιο αποτελεσματικός αλλά παρέχει και καλύτερο έλεγχο.

Οι web servers για κάθε αίτηση εξυπηρέτησης ανοίγουν και μία νέα διεργασία με αποτέλεσμα την υπερφόρτωση του συστήματος και την κατανάλωση πολλών πόρων. Από την άλλη μεριά οι διακομιστές εφαρμογών διαχειρίζονται τις



αιτήσεις εξυπηρέτησης ανοίγοντας νήματα που σημαίνει καλύτερη διαχείριση των πόρων του συστήματος.

Οι διακομιστές εφαρμογών είναι σχεδιασμένοι για την διαχείριση δεδομένων, επικοινωνία με βάσεις δεδομένων, αξιοπιστία δεδομένων, μετάφραση xml. Παρέχουν μια συλλογή από εργαλεία όπως Pooled Data Connections, data caches, session persistence και προστασία αποτυχίας διακομιστή.



Εικόνα 2.2 Application Server

Οι διακομιστές εφαρμογών είναι από τα πιο πολύπλοκα λογισμικά που έχουν σχεδιαστεί ποτέ. Έχουν πραγματοποιηθεί σπουδαία πράγματα με τους web servers ανάμεσα τους και πολύπλοκες εφαρμογές, όμως είχαν σαν κόστος τη μεγάλη διάρκεια και προσπάθεια και ήταν εφαρμογές της μιας χρήσης. Οι Διακομιστές Εφαρμογών από την άλλη πετυχαίνουν την πραγματοποίηση αυτών των εφαρμογών σε κανονική βάση χωρίς τεράστιες προσπάθειες.

Όπως φαίνεται μέχρι τώρα οι διακομιστές εφαρμογών είναι ίσως η καλύτερη λύση για μεγάλες και πολύπλοκες εφαρμογές που πρόκειται να καταναλώνουν μια μεγάλη ποσότητα υπολογιστικών πόρων. Αλλά θα πρέπει να πάρουμε στα σοβαρά ότι πρόκειται για ένα πολύπλοκο κομμάτι λογισμικού που συχνά απαιτεί ιδιαίτερες γνώσεις προκειμένου να λειτουργήσει σωστά.

## 2.4 Jakarta Tomcat

Για τις ανάγκες της εφαρμογής χρησιμοποιούμε σαν διακομιστή εφαρμογών τον Jakarta Tomcat. Ο Tomcat είναι ένας διακομιστής εφαρμογών ανοιχτού κώδικα που διανέμεται κάτω από την άδεια της Apache Software License. Ο Tomcat είναι ένας Servlet/JSP περιεκτής και περιέχει και επιπλέον εργαλεία και χαρακτηριστικά που τον καθιστούν χρήσιμο για την ανάπτυξη δικτυακών εφαρμογών. Είναι ένας διακομιστής εφαρμογών που είναι χτισμένος σε Java και επιτρέπει την εύκολη ανάπτυξη και διαχείριση κατανεμημένων Java εφαρμογών στο internet, intranet ή extranet.

Ο Tomcat φαίνεται να αποτελεί φαινόμενο ευρέως διαδεδομένο. Αυτό οφείλεται αναμφίβολα στην ανεξαρτησία πλατφόρμας που προσφέρει η Java και στην ευκολία εκμάθησης της γλώσσας. Κατά πολλούς η χρησιμοποίηση ενός Java διακομιστή για την ανάπτυξη εφαρμογών είναι πολύ πιο εύκολη και αρκετά πιο γραμμική από την ανάπτυξη χρησιμοποιώντας την κύρια ανταγωνιστική πρόταση της Microsoft (BackOffice).

Εκτός από τα δύο κυριότερα πλεονεκτήματα, ανεξαρτησία πλατφόρμας και ευκολία εκμάθησης, υπάρχουν και άλλα χαρακτηριστικά όπως

- Υποστήριξη CORBA και COM μέσω των JavaBeans προδιαγραφών
- Υποστήριξη Servlet – συνεπάγεται ότι η επιχειρηματική λογική δεν χρειάζεται να επικεντρώνεται στον διακομιστή. Μπορεί να υπάρχει ένας αριθμός από servlets – εφαρμογών που παράγονται από τον διακομιστή και χειρίζονται δεδομένα και διαδικασίες σε διαφορετικά μηχανήματα. Η βάση δεδομένων μπορεί να υπάρχει σε διαφορετικό μηχάνημα επίσης.

Ο Tomcat προσφέρει ένα πλήθος από χαρακτηριστικά και εργαλεία όπως

- Βελτιστοποίηση απόδοσης και ελαχιστοποίηση του συλλέκτη σκουπιδιών.
- Εργαλείο ανάπτυξης εφαρμογών επιτρέποντας την επικύρωση και την μεταγλώττιση των web εφαρμογών πριν την εγκατάστασή του στην παραγωγή

- Πλήρης παρακολούθηση διακομιστή χρησιμοποιώντας JMX και τον διαχειριστή διαδικτυακών εφαρμογών
- Αξιοπιστία
- Βελτιωμένη διαχείριση των Taglibs
- Ενσωμάτωση πλατφόρμας με Windows ή Unix – Linux
- Ενσωματωμένη υποστήριξη διαχείρισης ασφάλειας

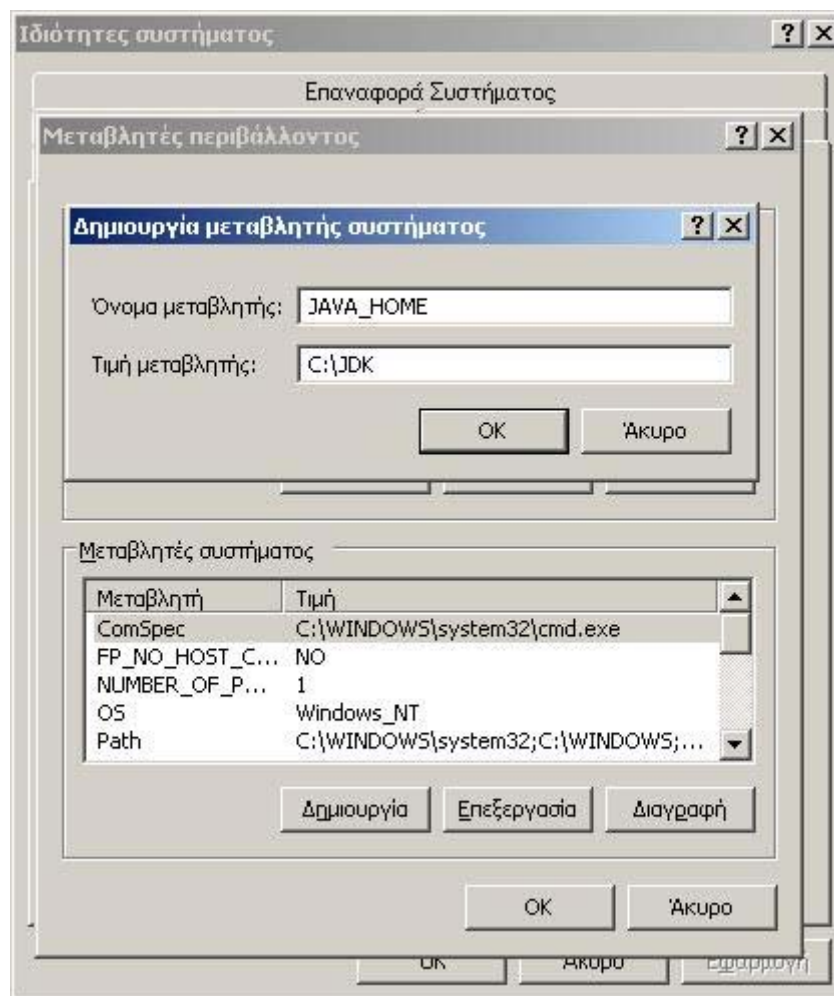
## 2.5 Εγκατάσταση Tomcat σε περιβάλλον windows

Η εγκατάσταση του tomcat σε περιβάλλον windows δεν διαφέρει καθόλου από τις άλλες εγκαταστάσεις προγραμμάτων που έχουμε συνηθίσει να πραγματοποιούμε. Υπάρχει το setup wizard που με απλά βήματα καθοδηγεί τον απλό χρήστη να εγκαταστήσει τον διακομιστή εφαρμογών. Απαραίτητη προϋπόθεση για την πραγματοποίηση της εγκατάστασης είναι η ύπαρξη του jdk στο συγκεκριμένο υπολογιστή και η ύπαρξη μεταβλητών συστήματος που να δείχνουν το γεγονός αυτό.

Το jdk ( Java Development Kit ) μπορεί να βρεθεί από το επίσημο δικτυακό τόπο της Sun Microsystems για την java που βρίσκεται στην ηλεκτρονική διεύθυνση <http://www.java.sun.com> Από εκεί μπορεί να μεταμορφωθεί δωρεάν η τελευταία έκδοση του πακέτου. Πρόκειται για ένα εκτελέσιμο αρχείο που αφού το μεταφορτώσουμε το τρέχουμε για να πραγματοποιήσουμε την εγκατάσταση. Σημαντικό, σε αυτό το σημείο, είναι να γνωρίζουμε την τοποθεσία στην οποία θα γίνει εγκατάσταση το πακέτο για να μπορέσουμε μετά να δηλώσουμε μια μεταβλητή συστήματος **JAVA\_HOME** που να περιέχει την συγκεκριμένη διαδρομή. Για να ορίσουμε την μεταβλητή συστήματος, ακολουθούμε την εξής διαδικασία (Πλατφόρμα Windows 2000/XP/2003 family).

Από τον πίνακα ελέγχου επιλέγουμε **Σύστημα (System)** και από το παράθυρο **Για προχωρημένους(Advanced)** επιλέγουμε **Μεταβλητές περιβάλλοντος**. Στο παράθυρο μεταβλητές συστήματος, επιλέγουμε Δημιουργία και πληκτρολογούμε στο όνομα μεταβλητής την τιμή **JAVA\_HOME** και στην τιμή μεταβλητής την διαδρομή

στο πακέτο. Αν για παράδειγμα το έχουμε εγκαταστήσει στο C:\JDK, τότε γράφουμε στα πεδία τις τιμές όπως φαίνεται στην εικόνα 2.3.



Εικόνα 2.3 Εισαγωγή Μεταβλητής

Με τον ίδιο τρόπο δηλώνουμε τις μεταβλητές συστήματος TOMCAT\_HOME και CATALINA\_HOME που θα μας χρειαστούν για τον διακομιστή εφαρμογών και η τιμή τους θα πρέπει να δείχνει στην διαδρομή που θα εγκατασταθεί.

Αφού ολοκληρώσουμε την εγκατάσταση του JDK συνεχίζουμε με την εγκατάσταση του Tomcat που μπορούμε να τον βρούμε και να τον μεταφορτώσουμε από την ηλεκτρονική διεύθυνση <http://tomcat.apache.org>. Αφού μεταφορτωθεί το τρέχουμε με διπλό κλικ στο αρχείο που βρίσκεται πλέον στον υπολογιστή μας και αν εγκαταστήσαμε σωστά το JDK τότε το πρώτο μήνυμα που θα πάρουμε από την

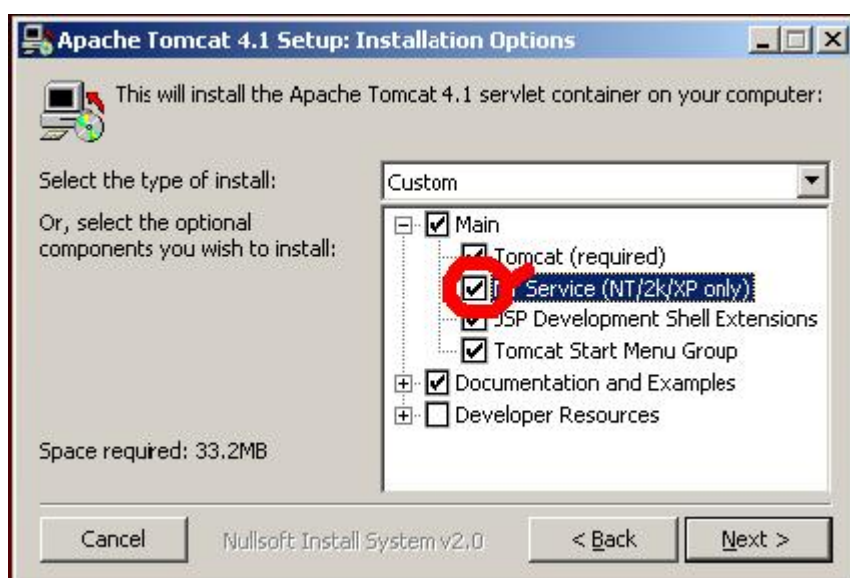
εγκατάσταση της εφαρμογής, θα έχει να κάνει με την επιβεβαίωση της έκδοσης JDK που πρόκειται να χρησιμοποιήσει.



Εικόνα 2.4 Εγκατάσταση Tomcat

Τα βήματα είναι απλά, θα χρειαστεί να επιλέξουμε τον κατάλογο που θα γίνει η εγκατάσταση ο διακομιστής που θα πρέπει να είναι ο ίδιος με την τιμή που είχαμε δηλώσει για την μεταβλητή συστήματος TOMCAT\_HOME. Επίσης θα μας ζητήσει να δηλώσουμε ποια συστατικά θέλουμε να εγκατασταθούν στο σύστημα μας. Εδώ καλό θα ήταν να επιλέξουμε το κουτάκι NT Service (NT/2k/XP only) που θα μας εγκαταστήσει τον διακομιστή σαν υπηρεσία, πράγμα που θα μας διευκολύνει στην διαχείριση της εκκίνησης και του τερματισμού του διακομιστή.

Αφού τελειώσει η εγκατάσταση και αν έχουμε επιλέξει το κουτάκι με το NT Service τότε ο διακομιστής θα ξεκινήσει να λειτουργεί.



Εικόνα 2.5 NT Service

Για να επιβεβαιώσουμε την επιτυχία της εγκατάστασης ανοίγουμε τον αγαπημένο μας browser και πληκτρολογούμε την διεύθυνση <http://localhost:8080>. Προεπιλεγμένα ο διακομιστής εφαρμογών χρησιμοποιεί την θύρα 8080 για να παρέχει τις υπηρεσίες του. Αυτό βέβαια μπορούμε να το αλλάξουμε όπως θα δούμε παρακάτω. Η σελίδα που θα εμφανισθεί στον browser θα πρέπει να είναι παρόμοια με την εικόνα 2.6, ανάλογα βέβαια με την έκδοση του διακομιστή που χρησιμοποιούμε.

Τα αρχεία παραμετροποίησης του Tomcat βρίσκονται στον κατάλογο **TOMCAT\_HOME\conf**. Δηλαδή αν τον έχουμε εγκαταστήσει στον κατάλογο **c:\tomcat** τότε θα βρίσκονται στον κατάλογο **c:\tomcat\conf**. Γενικά η δομή του καταλόγου του Tomcat έχει ως εξής

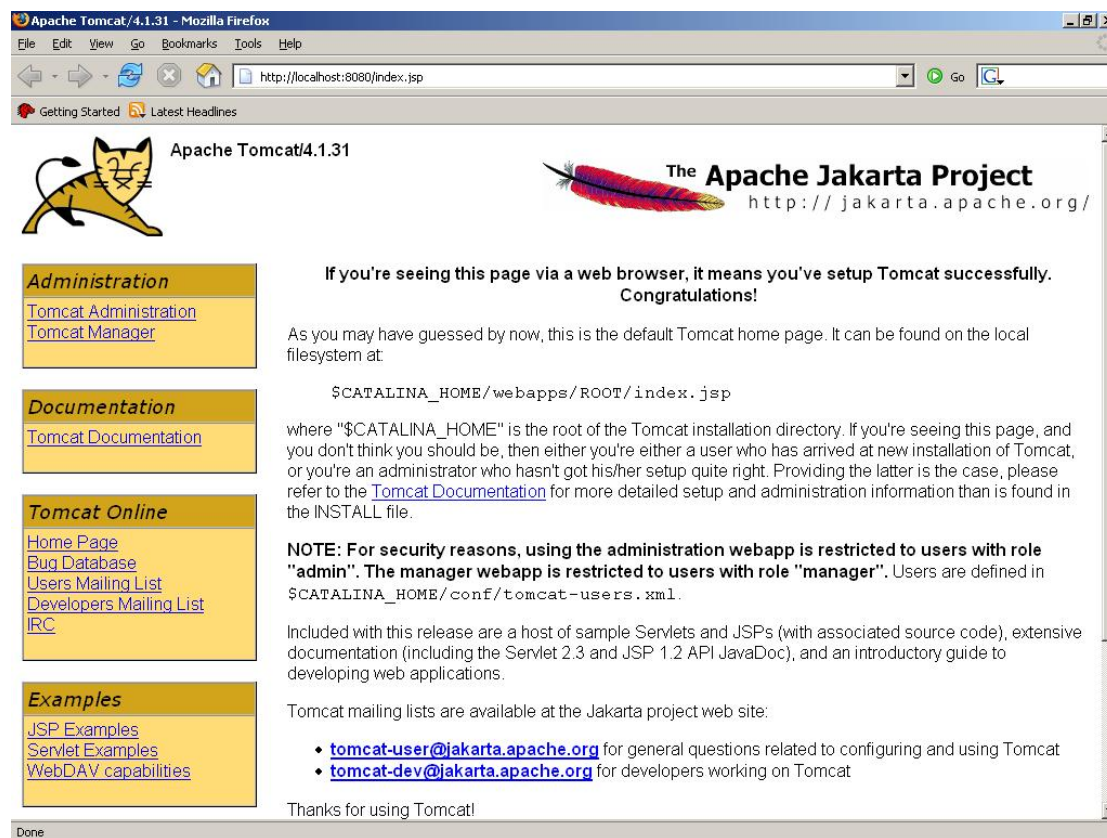
- **\bin** – Εδώ βρίσκονται τα σενάρια κελύφους για την εκκίνηση και τερματισμό του διακομιστή σε μορφή \*.bat και επίσης κάποια βοηθητικά αρχεία
- **\conf** – Αρχεία παραμετροποίησης για τον διακομιστή με σημαντικότερο το **server.xml** που είναι το κεντρικό αρχείο για την λειτουργία του διακομιστή. Υπάρχουν επίσης αρχεία για επιβολή πολιτικής ασφαλείας όπως tomcat-users και catalina.policy
- **\logs** – Τα αρχεία ιστορικού και καταγραφής σφαλμάτων βρίσκονται προεπιλεγμένα εδώ. Σε περίπτωση σφάλματος του διακομιστή ή μη σωστής λειτουργίας του, εδώ πιθανότατα θα βρείτε τον λόγο.
- **\webapps** – Εδώ φιλοξενούνται και αναπτύσσονται οι εφαρμογές μας.
- **\temp** – Εδώ φιλοξενούνται τα προσωρινά αρχεία που παράγονται από την Java. Ο κατάλογος αυτός πρέπει να υπάρχει για να λειτουργήσει σωστά ο διακομιστής
- **\common** – Διάφορες βιβλιοθήκες και κλάσεις

Για να εγκαταστήσουμε την εφαρμογή μας ή να ξεκινήσουμε να την αναπτύσσουμε έχουμε δύο επιλογές. Η πρώτη είναι να εγκαταστήσουμε την εφαρμογή μας χειροκίνητα, από το αρχείο server.xml όπου πρέπει να δηλώσουμε την διαδρομή της εφαρμογής μας στον υπολογιστή καθώς και την διαδρομή της ηλεκτρονικής διεύθυνσης. Μία γενικότερη σύνταξη αυτής της δήλωσης φαίνεται

παρακάτω, υποθέτοντας ότι η εφαρμογή μας βρίσκεται στον κατάλογο c:\myApp και θέλουμε να έχει την ηλεκτρονική <http://localhost:8080/myApp>.

```
<Context path="/myApp" docBase="c:\myApp" debug="1" reloadable="true"/>
```

Η πιο πάνω πρόταση πρέπει να μπει σε συγκεκριμένο σημείο μέσα στο αρχείο server.xml και συγκεκριμένα πριν από της προτάσεις </Host> </Engine> </Service>. Σε περίπτωση που θέλουμε να δηλώσουμε virtual host τότε η πρόταση θα πρέπει να βρίσκεται πριν από τις προτάσεις </Engine></Service> και έχει διαφορετική σύνταξη.



Εικόνα 2.6 Αρχική σελίδα του Tomcat

Η δομή του καταλόγου της εφαρμογής μας είναι η παρακάτω.

- \ - Εδώ βρίσκονται όλα τα αρχεία jsp, html και ότι άλλο χρειαζόμαστε, όπως φωτογραφίες και άλλα. Επίσης μπορούμε να δημιουργήσουμε καταλόγους και να τους χρησιμοποιήσουμε ανάλογα με τις ανάγκες μας.
- \WEB-INF – Εδώ βρίσκεται συνήθως το αρχείο web.xml που περιέχει την περιγραφή της εφαρμογής και τα διάφορα συστατικά που χρησιμοποιεί.
- \WEB-INF\classes – Εδώ βρίσκονται οι java κλάσεις που μπορούμε να χρησιμοποιήσουμε στην εφαρμογή μας μέσω των JavaBeans

## 2.6 Εγκατάσταση Tomcat σε περιβάλλον Linux

Για την δοκιμή της εγκατάστασης του διακομιστή χρησιμοποιήθηκε η διανομή Slackware Linux 10.1 με πυρήνα 2.6.13 και με JDK 1.4.2\_08.

Από την ηλεκτρονική διεύθυνση <http://tomcat.apache.org> μεταφορτώνουμε την έκδοση του διακομιστή για Linux και συγκεκριμένα το binary αρχείο με την κατάληξη tar.gz. Αφού το έχουμε στον υπολογιστή μας το αποσυμπιέζουμε με την εντολή

```
tar xvzf jakarta-tomcat-xxxx.tar.gz
```

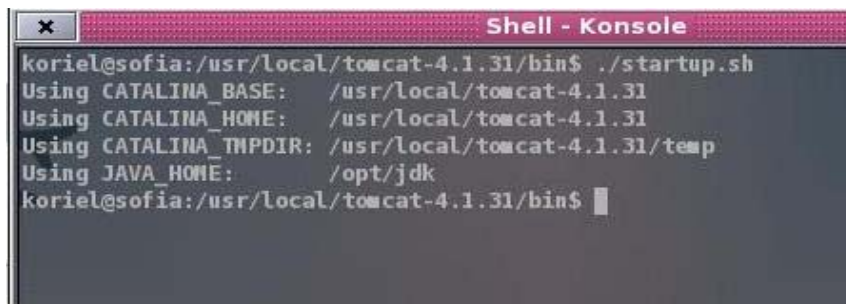
όπου xxxx είναι η έκδοση που έχουμε μεταφορτώσει. Έπειτα δηλώνουμε αν δεν υπάρχει μεταβλητή για το που βρίσκεται το JDK. Αυτή τη δήλωση καλό είναι να την βάλουμε σε ένα αρχείο εκκίνησης ώστε να δηλώνεται αυτόματα με την εκκίνηση του συστήματος, π.χ. rc.local, bash\_profile. Για να δηλώσουμε την μεταβλητή πληκτρολογούμε

```
export JAVA_HOME=/usr/local/jdk
```

όπου /usr/local/jdk είναι η διαδρομή που έχουμε εγκατεστημένο το JDK. Τώρα πηγαίνουμε στον κατάλογο που έχει δημιουργηθεί από την αποσυμπίεση του



διακομιστή και στον υποκατάλογο /bin και εκτελούμε την εντολή ./startup.sh. Αυτό θα ξεκινήσει τον διακομιστή και μπορούμε να το δούμε ανοίγοντας έναν browser και πληκτρολογώντας την διεύθυνση <http://localhost:8080>. Αυτό ήταν. Για το τερματισμό του διακομιστή από τον ίδιο κατάλογο πληκτρολογούμε την εντολή ./shutdown.sh.



```
koriel@sofia:/usr/local/tomcat-4.1.31/bin$ ./startup.sh
Using CATALINA_BASE:   /usr/local/tomcat-4.1.31
Using CATALINA_HOME:   /usr/local/tomcat-4.1.31
Using CATALINA_TMPDIR: /usr/local/tomcat-4.1.31/temp
Using JAVA_HOME:       /opt/jdk
koriel@sofia:/usr/local/tomcat-4.1.31/bin$
```

Εικόνα 2.7 Εκκίνηση Tomcat σε περιβάλλον Linux

Γενικότερα η εγκατάσταση του Tomcat σε περιβάλλον Linux είναι αρκετά εύκολη και η συμπεριφορά του αρκετά πιο σταθερή και αξιόπιστη από την ανάλογη σε περιβάλλον Linux. Η τεχνογνωσία όμως γύρω από το λειτουργικό σύστημα ανοιχτού κώδικα είναι περιορισμένη και η συντήρηση ενός διακομιστή βασισμένο σε Linux είναι αρκετά πιο δύσκολη από την αντίστοιχη ενός Window διακομιστή. Για την ανάγκη του HealLink ο διακομιστής εφαρμογών είναι εγκατεστημένος σε περιβάλλον Windows.

## 2.7 Γλώσσα Σεναρίου JSP ( Java Server Pages )

### 2.7.1 Γενικά

Η JSP είναι μια γλώσσα σεναρίου server-side τεχνολογίας. Αυτό σημαίνει ότι η εκτέλεση του κώδικα της και η εξαγωγή των αποτελεσμάτων πραγματοποιείται στην πλευρά του διακομιστή και όχι στην πλευρά του χρήστη. Είναι πολύ πιο ισχυρή γλώσσα από τις client-side γλώσσες κι αυτό γιατί δεν βασίζονται στην δυναμική ενός άγνωστου συστήματος – αφού οι client-side γλώσσες μεταγλωττίζονται από τον πελάτη – αλλά στην δυναμική του διακομιστή. Ο προγραμματιστής μπορεί να δοκιμάσει τον κώδικά του στον διακομιστή και εφόσον λειτουργεί, μπορεί να είναι σίγουρος ότι θα λειτουργεί σωστά και σε όλους τους πελάτες.

Στην ανάπτυξη των προδιαγραφών της JSP η Sun Microsystems συνεργάστηκε με πρωτοποριακούς web-servers, application servers και εργαλεία ανάπτυξης, με στόχο να δημιουργήσει ένα λογισμικό που να είναι ανεξάρτητο πλατφόρμας. Αυτό που κατάφερε είναι η δημιουργία ενός εργαλείου που επιταχύνει την ανάπτυξη των εφαρμογών με τρόπους όπως:

- Διαχωρισμός του κώδικα από την παρουσίαση  
Με την χρησιμοποίηση της τεχνολογίας JSP οι σχεδιαστές χρησιμοποιούν ετικέτες HTML και XML για να σχεδιάσουν και να μορφοποιήσουν την σελίδα παρουσίασης. Χρησιμοποιούν JSP ετικέτες ή scriptlets για να παράγουν τις δυναμικές σελίδες. Η λογική που παράγει αυτές τις σελίδες είναι πακεταρισμένη σε ετικέτες και συστατικά JavaBeans, τα οποία εκτελούνται στην πλευρά του διακομιστή. Ο διακομιστής μεταγλωττίζει τις JSP ετικέτες και τα scriptlets παράγοντας το τελικό αποτέλεσμα στον browser. Αυτό βοηθάει τους σχεδιαστές στην διαφύλαξη του κώδικα και εξασφαλίζει την φορητότητα σε οποιοδήποτε browser.
- Δημιουργία επαναχρησιμοποιήσιμων συστατικών  
Οι περισσότερες JSP σελίδες μπορούν να χρησιμοποιηθούν και σε άλλες εφαρμογές.

- Απλοποίηση ανάπτυξης με ετικέτες  
Οι περισσότεροι σχεδιαστές διαδικτυακών εφαρμογών δεν είναι γνώριμοι με τις γλώσσες σεναρίου. Η JSP τεχνολογία ενσωματώνει πολλές από τις μεθόδους που χρειάζονται για την δημιουργία δυναμικού περιεχομένου, σε ετικέτες εύκολες στην χρήση. Οι ετικέτες αυτές μπορούν να έχουν πρόσβαση σε JavaBeans συστατικά, να ορίσουν ή να αποκτήσουν τιμές από μεταβλητές και να εκτελέσουν διάφορες λειτουργίες που είναι πιο δύσκολες και πιο χρονοβόρες για να προγραμματιστούν.

Από τα πιο μεγάλα πλεονεκτήματα της τεχνολογίας είναι

- Επειδή η γλώσσα σεναρίου JSP είναι βασισμένη στην γλώσσα προγραμματισμού Java και επειδή όλες οι σελίδες JSP μεταγλωττίζονται σε Java Servlets, έχουν όλα τα πλεονεκτήματα της Java τεχνολογίας όπως καλή διαχείριση της μνήμης και ασφάλεια και πρόσβαση σε όλη την οικογένεια
- Ως κομμάτι της Java τεχνολογίας, η JSP ενσωματώνει την φιλοσοφία ‘Γράψε μια φορά, Χρησιμοποίησε το οπουδήποτε’.

### 2.7.2 Σύνταξη JSP

Η σύνταξη των σελίδων JSP είναι παρόμοια με την HTML ή XML σύνταξη, με επιπλέον συστατικά και στοιχεία που επεξεργάζονται από την μηχανή JSP. Καλύτερα όμως να σας παρουσιάσουμε την τεχνολογία με ένα παράδειγμα. Ο παρακάτω κώδικας είναι μια απλή σελίδα JSP που μας εμφανίζει ένα μήνυμα.

```
<HTML>  
<body>  
<% ="This is a JSP page" %>  
</body>  
</HTML>
```

Όπως φαίνεται ο κώδικας JSP (scriptlet) περικλείεται από τους τελεστές `<%` και `%>`. Ότι τοποθετείτε ανάμεσα στους τελεστές, μεταγλωττίζεται και το αποτέλεσμα περνάει στον πελάτη και εμφανίζεται στον browser.

Ο τρόπος με τον οποίο μπορούμε να συμπεριλάβουμε τα πακέτα της Java στις JSP σελίδες δηλώνεται στην αρχή της σελίδας μας και ακολουθεί την πιο κάτω ντιρεκτίβα:

```
<%@ page import="package.name" %>
```

Για την χρησιμοποίηση ενός Java Bean πρέπει να συμπεριλάβουμε την παρακάτω ετικέτα στην αρχή της σελίδας μας.

```
<jsp:useBean id="beanid" class="classtouse">  
</jsp:useBean>
```

Για να έχουμε πρόσβαση στις μεθόδους του JavaBean που δημιουργήσαμε χρησιμοποιούμε το όνομα της ταυτότητας που του δώσαμε στην δήλωση του. Για παράδειγμα αν του δώσαμε `id="myBean"` και υπάρχει μια μέθοδος `getMessage()` που μας επιστρέφει μια τιμή, τότε η πρόσβαση στην μέθοδο αυτή πετυχένεται με την δήλωση

```
<% myBean.getMessage() %>
```

Το JavaBean θα πρέπει να βρίσκεται στον κατάλογο `classes` της εφαρμογής μας. Ας το δούμε καλύτερα με ένα παράδειγμα που χρησιμοποιεί το JavaBean `myBean` που υλοποιεί την πιο πάνω μέθοδο.

Κλάση `myBean.java`

```
package myPackage;  
public class myBean{  
    static String myMessage="This is a JavaBean";
```

```
public String getMessage() {  
    return myMessage;  
}  
}
```

Σελίδα myBean.jsp

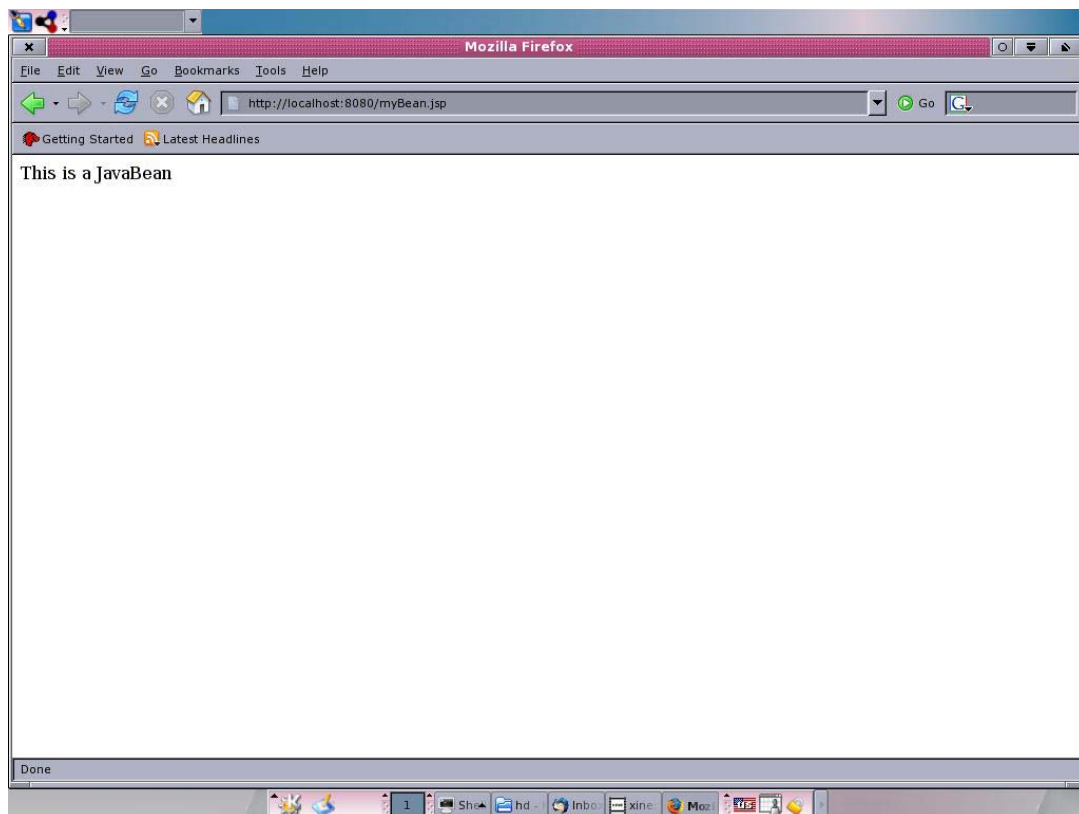
```
<jsp:useBean id="myBean" class="myPackage.myBean">  
</jsp:useBean>
```

```
<%
```

```
    out.println(myBean.getMessage());
```

```
%>
```

Το αποτέλεσμα φαίνεται στην παρακάτω εικόνα.



Εικόνα 2.8 JavaBean

Η αίσθηση που δίνει ο προγραμματισμός σε γλώσσα σεναρίου JSP μοιάζει πολύ με τον προγραμματισμό της γλώσσας Java, την καθιστά πολύ εύκολη στην εκμάθηση και βοηθάει τους σχεδιαστές να αναπτύσσουν τις εφαρμογές τους γρήγορα και εύκολα, δίνοντας τους το πλεονέκτημα να ασχοληθούν πιο πολύ με την εφαρμογή παρά με την γλώσσα υλοποίησής της. Ένα άλλο πλεονέκτημα είναι η ευκολία με την οποία κάποιος μπορεί να διαβάσει και να διορθώσει μια εφαρμογή που είναι γραμμένη σε γλώσσα JSP, ειδικά αν η δομή του κώδικα είναι προσεγμένη.

### 2.7.3 JSP και SQL

Ενδιαφέρον εδώ για τις ανάγκες του HEAL-Link παρουσιάζει η αλληλεπίδραση της εφαρμογής με την βάση δεδομένων. Η συνδεσιμότητα των JSP σελίδων με την βάση δεδομένων για ανάκτηση και επεξεργασία δεδομένων πραγματοποιείται μέσω του JDBC οδηγού. Ο πελάτης κάνει μια αίτηση για να ανακτήσει στοιχεία που βρίσκονται σε μια βάση δεδομένων. Η μηχανή JSP επικοινωνεί με την βάση με τη βοήθεια του JDBC οδηγού και αφού πραγματοποιήσει τις απαραίτητες εργασίες παρουσιάζει το αποτέλεσμα στον πελάτη όπως φαίνεται και στην παρακάτω εικόνα.



Εικόνα 2.8 Επικοινωνία JSP με Βάση Δεδομένων

Αρχικά δημιουργούμε ένα αντικείμενο σύνδεσης για την βάση δεδομένων το οποίο χρησιμοποιούμε για όλες μας τις εργασίες. Η δημιουργία του αντικειμένου πραγματοποιείται από την πρόταση:

```
Class.forName("drivername").newInstance();
```

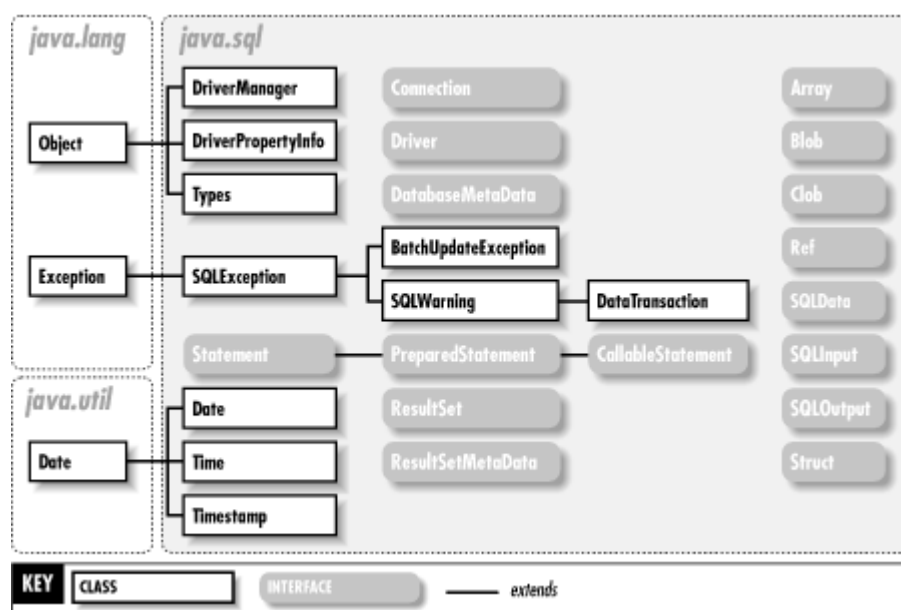
Στην θέση του drivername βάζουμε τον ανάλογο οδηγό ανάλογα με την σύνδεση που θέλουμε να πραγματοποιήσουμε όπως π.χ. μέσω odbc ή μέσω δικτύου. Έπειτα

δημιουργούμε την σύνδεση χρησιμοποιώντας την ανάλογα με την περίπτωση πρόταση.

### Connection

```
con=DriverManager.getConnection("databasepath","username","password");
```

Όταν πραγματοποιηθεί η σύνδεση μπορούμε να δημιουργήσουμε ερωτήματα προς την βάση χρησιμοποιώντας τις μεθόδους του πακέτου `java.sql.*` της Java. Φαίνεται στην παρακάτω εικόνα το δέντρο του πακέτου `sql` της Java.



Εικόνα 2.9 java.sql Πακέτο

Οι JSP επωφελούνται από την `java` στο γεγονός ότι μπορεί να πραγματοποιήσουν σύνδεση με οποιαδήποτε βάση υποστηρίζει η `java`. Υπάρχουν αρκετοί οδηγοί τους οποίους μπορούμε να συμπεριλάβουμε στην εφαρμογή μας ώστε να εκμεταλευτούμε την σύνδεσή μας με μια βάση δεδομένων. Ενδεικτικά αναφέρονται μερικές από τις βάσεις που υποστηρίζονται:

- IBM DB2
- MySQL
- PostgreSQL
- Microsoft ACCESS

- SQL Server

Για να ενσωματώσουμε οδηγίες sql στις JSP σελίδες μας μπορούμε να το πραγματοποιήσουμε είτε με scriptlets, είτε με JavaBeans. Για να χρησιμοποιήσουμε scriptlets θα πρέπει να συμπεριλάβουμε το πακέτο java.sql στην αρχή της σελίδας μας και να γράψουμε τον κώδικά μας μέσα στις ετικέτες <% %>. Παρουσιάζουμε ένα μικρό παράδειγμα πιο κάτω.

```
<% @ page import="java.sql.*"%>
```

```
<%  
try {  
Class.forName("COM.ibm.db2.jdbc.net.DB2Driver").newInstance();  
Connection  
con=DriverManager.getConnection("jdbc:odbc:xxxxxx","xxxxx","xxxxx");  
Statement stmt=con.createStatement();  
String sql="select someField from someTable";  
ResultSet rs=stmt.executeQuery(sql);  
while(rs.next()) {  
    out.println(rs.getString("someField"));  
}  
rs.close();  
stmt.close();  
con.close();  
} catch(SQLException e) { out.println(e.toString()); }  
%>
```

Στο παραπάνω παράδειγμα συνδεθήκαμε σε μια βάση δεδομένων με όνομα xxxxxx με το όνομα χρήστη και τον κωδικό μας. Κάναμε ερώτημα στη βάση ζητώντας το πεδίο someField από τον πίνακα someTable και μετά εκτυπώσαμε τα αποτελέσματα στον browser. Όλη η διαδικασία βρίσκεται σε ένα μπλοκ try ... catch για να χειριστούμε τυχόν σφάλμα. Εδώ σε περίπτωση σφάλματος το εκτυπώνουμε στον browser.



## 2.8 Σύστημα Διαχείρισης Βάσεων Δεδομένων

Το σύστημα διαχείρισης βάσεων δεδομένων είναι λογισμικό ειδικά σχεδιασμένο ώστε να διευκολύνει τη χρήση και συντήρηση μεγάλου όγκου πληροφορίας. Η ανάγκη και η χρησιμοποίηση ενός τέτοιου συστήματος έχει γίνει απαραίτητη στις περισσότερες εφαρμογές, καθώς παρέχει αξιοπιστία στην διαχείριση της πληροφορίας σε σχέση με τους παραδοσιακούς τρόπους (Διαχείριση πληροφορίας σε αρχεία με χρήση γλώσσας προγραμματισμού τρίτου επιπέδου).

Το σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων είναι ένα σύστημα που ακολουθεί το σχεσιακό μοντέλο. Στο σχεσιακό μοντέλο τα δεδομένα εκφράζονται με μαθηματικές σχέσεις, π.χ. ο υπάλληλος της εταιρίας ΤΑΔΕ, το σύνολο των φοιτητών του ΤΕΙ ΤΑΔΕ κλπ. Στο σχεσιακό σύστημα, η βάση δεδομένων είναι σχεδιασμένη και οργανωμένη ώστε τα δεδομένα να διαχειρίζονται με βάση την σχέση που έχουν μεταξύ τους. Το σύστημα αυτό έχει δημιουργηθεί από μία ομάδα, με οργανωτή τον Dr. Edmund F. Codd και χρηματοδοτήθηκε από την IBM στις αρχές του 1970. Μερικές βάσεις που υποστηρίζουν το σχεσιακό μοντέλο είναι οι

- Oracle
- SQL Server
- DB2
- Sybase

Για τις ανάγκες της εφαρμογής HealLink χρησιμοποιείται το σχεσιακό σύστημα βάσεων δεδομένων της IBM, η DB2 v8.1.

## ΚΕΦΑΛΑΙΟ 3ο : Η ΕΦΑΡΜΟΓΗ

### 3.1 Εισαγωγή στα Sessions

Το http είναι μη συνδεδεμένο πρωτόκολλο που σημαίνει ότι κάθε φορά που ο browser κάνει αίτηση για μια ιστοσελίδα στον διακομιστή, ο τελευταίος δεν κρατάει καμιά πληροφορία για τον πελάτη. Αυτό δημιουργεί αρκετά μειονεκτήματα σε μια διαδικτυακή εφαρμογή όπως το online-shop, όταν ο διακομιστής δεν μπορεί να θυμηθεί προηγούμενες συναλλαγές. Έτσι αν κάνετε τις αγορές σας στο online shop και προσθέσετε ένα προϊόν στο καλάθι σας, ο διακομιστής δεν είναι σε θέση να θυμηθεί το περιεχόμενο του καλαθιού στις προηγούμενες προσθήκες, γεγονός που δημιουργεί προβλήματα στους σχεδιαστές τέτοιων εφαρμογών.

Υπάρχουν τρεις λύσεις στο παραπάνω πρόβλημα

- Cookies. Μπορούν να χρησιμοποιηθούν http cookies για να αποθηκεύσουν πληροφορίες για μία σύνοδο χρήστη και να μπορούν να την ανακτήσουν κάθε στιγμή. Τα cookies είναι στην ουσία αντικείμενα τα οποία είναι συνδεδεμένα με τον πελάτη που έχει ανοίξει μια σύνδεση με τον διακομιστή. Μπορεί να κρατήσει οποιαδήποτε πληροφορία θέλουμε για τον πελάτη ώστε να είμαστε σε θέση να τις χρησιμοποιήσουμε αργότερα. Αυτή η λύση αποτελεί και την πλέον διαδεδομένη και αποδεκτή από τους περισσότερους σχεδιαστές εφαρμογών αλλά υπάρχουν μερικές λεπτομέρειες που θα πρέπει να παραμετροποιηθούν και να ληφθούν σοβαρά υπόψη όπως
  - Ο χρόνος λήξης των cookies.
  - Το είδος και το μέγεθος της πληροφορίας που αποθηκεύονται στα cookies όπως για παράδειγμα αριθμοί πιστωτικών καρτών, κωδικοί πρόσβασης, ογκώδεις πληροφορίες που σε συνδυασμό με την κίνηση του διακομιστή μπορεί να οδηγήσει στην κατανάλωση μεγάλου όγκου δεδομένων.
- URL Rewriting. Μπορούν να ενσωματωθούν δεδομένα στο τέλος κάθε url που

προσδιορίζουν την σύνοδο και ο διακομιστής μπορεί να χρησιμοποιήσει αυτά τα δεδομένα για να αναγνωρίσει την διακεκριμένη σύνοδο και να είναι σε θέση να δώσει πρόσβαση στα δεδομένα που είναι αποθηκευμένα γι' αυτή τη σύνοδο. Αυτή είναι επίσης μια πολύ καλή λύση αλλά έχει κάποια σημεία που πρέπει να προσεχτούν όπως η προσθήκη των δεδομένων που χρειάζονται για την αναγνώριση των συνόδων στο τέλος κάθε url, γεγονός που μπορεί να δημιουργήσει αρκετά προγραμματιστικά προβλήματα στους σχεδιαστές καθώς χρειάζεται ιδιαίτερη προσοχή.

- Κρυφά Πεδία. Οι φόρμες HTML έχουν κάποια πεδία τα οποία δεν είναι ορατά στον χρήστη και ακολουθούν την σύνταξη “<input type=hidden name=’session’ value=’someValue’>” τα οποία μπορούν να χρησιμοποιηθούν για την αποθήκευση και διαχείριση της συνόδου. Η μέθοδος αυτή έχει το μεγάλο μειονέκτημα ότι θα πρέπει όλες οι ιστοσελίδες να παράγονται δυναμικά.

Τα servlets παρέχουν μια εξαιρετική τεχνική λύση στο παραπάνω πρόβλημα που ακούει στο όνομα HttpSession API. Πρόκειται για μία υψηλού επιπέδου διασύνδεση που είναι χτισμένη πάνω από τα cookies και το URL Rewriting. Στην πραγματικότητα, οι διακομιστές χρησιμοποιούν cookies αν υποστηρίζονται από τον browser και αν όχι γυρίζουν αυτόματα στο URL Rewriting. Ο σχεδιαστής που χρησιμοποιεί servlets δεν χρειάζεται να ανησυχεί για το αν υποστηρίζονται cookies από τον browser, ούτε να διαχειρίζεται cookies και url rewriting, ενώ συγχρόνως του δίνεται αρκετός χώρος για να αποθηκεύσει τις συνόδους και τα δεδομένα τους, γεγονός που καθιστά την τεχνολογία servlets πιο ελκυστική για τη δημιουργία εφαρμογών.

Η χρησιμοποίηση των συνόδων έχει να κάνει με την αναζήτηση της συνόδου που αναλογεί στην συγκεκριμένη σύνδεση, η δημιουργία νέας συνόδου, η ανάκτηση πληροφορίας που βρίσκεται στην διακεκριμένη σύνοδο, η αποθήκευση δεδομένων σε αυτήν και τέλος ο τερματισμός της συνόδου.

### 3.1.1 Αναζήτηση – Δημιουργία συνόδου.

Αυτή η λειτουργία πραγματοποιείται με την κλήση της μεθόδου `getSession` της κλάσης `HttpServletRequest`. Αν επιστρέψει `NULL` τότε υπάρχει η δυνατότητα δημιουργίας συνόδου, αν και αυτό μπορεί να πραγματοποιηθεί απευθείας αν καλέσουμε την μέθοδο `getSession(true)`. Έτσι το πρώτο μας βήμα είναι να δηλώσουμε την πρόταση:

```
HttpSession session=request.getSession(true);
```

### 3.1.2 Ανάκτηση πληροφορίας από σύνοδο

Τα `HttpSessions` βρίσκονται στον διακομιστή αποθηκευμένα και συσχετίζονται αυτόματα με τον πελάτη με έναν διαφανή μηχανισμό παρόμοιο με τα `cookies` και το `url rewriting`. Τα αντικείμενα αυτά έχουν μια δομή δεδομένων που επιτρέπουν την αποθήκευση οποιαδήποτε μορφής και μεγέθους πληροφορία. Στην έκδοση 2.1 υπάρχει η μέθοδος `getValue("key")` που επιτρέπει την ανάκτηση της πληροφορίας που είναι αποθηκευμένη στην μεταβλητή `"key"`. Η επιστρεφόμενη τιμή είναι αντικείμενο και θα πρέπει να γίνει η ανάλογη μετατροπή για να ανακτηθεί ο κατάλληλος τύπος της πληροφορίας που είναι αποθηκευμένη στον διακομιστή. Αυτό βοηθάει στην καλύτερη διαχείριση της πληροφορίας, καθώς και της δομής της, απελευθερώνοντας τον σχεδιαστή από δημιουργία ανάλογης δομής δεδομένων. Αν δεν υπάρχει αντικείμενο με αυτό το όνομα τότε επιστρέφεται η τιμή `NULL`. Στην έκδοση 2.2 η μέθοδος αυτή έχει εγκαταλειφθεί και έχει αντικατασταθεί με την μέθοδο `getAttribute()` για δυο λόγους.

- Το όνομα `getAttribute` ταιριάζει περισσότερο με την μέθοδο `setAttribute()` καθώς η ανάλογη της `getValue` είναι η `putValue` και όχι η `setValue`
- Η `setAttribute` ενσωματώνει έναν ακροαστή `HttpSessionBindingListener` που παρακολουθεί τις τιμές, ενώ η `putValue` όχι.

Ένα μικρό παράδειγμα της δομής του κώδικα για την ανάκτηση δεδομένων από μια σύνοδο φαίνεται παρακάτω:

```
HttpSession session = request.getSession(true);
```

```
ShoppingCart previousItems =  
(ShoppingCart)session.getAttribute("previousItems").toString();  
if (previousItems != null) {  
    doSomethingWith(previousItems);  
} else {  
    previousItems = new ShoppingCart(...);  
    doSomethingElseWith(previousItems);  
}
```

Αν υπάρχουν πολλές μεταβλητές με δεδομένα στη σύνοδο, ένας τρόπος για να πάρετε τα ονόματά τους είναι η κλήση της μεθόδου `getAttributeNames` που επιστρέφει έναν πίνακα συμβολοσειρών με τα ονόματα των πεδίων αυτών.

Παρόλο που η αποθηκευμένη πληροφορία είναι αυτή που ενδιαφέρει κυρίως τους σχεδιαστές εφαρμογών, υπάρχουν και άλλες πληροφορίες που αποθηκεύονται σε μια σύνοδο και μπορεί να εξυπηρετήσουν μερικές φορές τους σχεδιαστές. Οι πληροφορίες αυτές με τις αντίστοιχες μεθόδους είναι

2. `getId` – Αυτή η μέθοδος επιστρέφει τον μοναδικό αναγνωριστικό που παράγεται αυτόματα κατά τη δημιουργία της συνόδου.
3. `isNew` – Επιστρέφει `true` αν ο browser βλέπει για πρώτη φορά τη σύνοδο, συνήθως επειδή μόλις δημιουργήθηκε. Επιστρέφει `false` αν η σύνοδος προϋπάρχει.
4. `getLastAccessedTime` – Αυτή η μέθοδος επιστρέφει την χρονική στιγμή, μετρημένη σε `milliseconds`, της τελευταίας επαφής που είχε ο πελάτης με την σύνοδο.
5. `getMaxInactiveInterval` – Επιστρέφει τον χρονικό διάστημα, σε δευτερόλεπτα, που μένει ενεργή μία σύνοδος χωρίς να επικοινωνήσει ο πελάτης. Αν η τιμή είναι αρνητική σημαίνει ότι η σύνοδος δεν λήγει ποτέ.

### 3.1.3 Αποθήκευση δεδομένων στην σύνοδο

Η αποθήκευση δεδομένων γίνεται με την μέθοδο `setAttribute` και ένα ζευγάρι που παραμέτρων που περιλαμβάνει το όνομα του πεδίου και την τιμή που αυτό θα περιέχει. Για παράδειγμα αν θέλουμε να αποθηκεύσουμε σε μια σύνοδο το όνομα και την ηλικία ενός πελάτη μπορούμε να χρησιμοποιήσουμε την μέθοδο ως εξής:

```
session.setAttribute("Name","someName");
```

```
session.setAttribute("Age","someAge");
```

## 3.2 Διαχείριση Sessions στο HealLink

Τα sessions στην πύλη HealLink είναι ενεργά καθ' όλη την διάρκεια της επίσκεψης του χρήστη και λειτουργούν διαφορετικά σε περίπτωση που ο χρήστης είναι επισκέπτης και στην περίπτωση που ο χρήστης είναι μέλος.

Στην αρχική επίσκεψη του χρήστη στην πύλη, δημιουργείται μία σύνοδος η οποία έχει σαν μέγιστη διάρκεια, χωρίς να έχει επικοινωνία με τον χρήστη, 30 λεπτά. Κατά την δημιουργία της συνόδου επηρεάζεται τόσο η δομή των δεδομένων της συνόδου, όσο και η βάση δεδομένων του HealLink.

Για τη δομή των δεδομένων της συνόδου έχουμε τις εξής καταχωρήσεις

```
session.setMaxInactiveInterval(1800);
```

```
session.setAttribute("name","guest");  
  
session.setAttribute("status","guest");  
  
session.setAttribute("email","guest");  
  
session.setAttribute("logintime",time);  
  
session.setAttribute("remoteIp",ip);  
  
session.setAttribute("u_id","-1");
```

Παραπάνω ορίζουμε τη μέγιστη διάρκεια ζωής της συνόδου, όταν παραμένει ανενεργή, σε 1800 δευτερόλεπτα, δηλαδή σε 30 λεπτά. Καταχωρούμε την τιμή “guest” στις μεταβλητές name, status και email. Η τιμή της μεταβλητης time υπολογίζεται τη στιγμή που δημιουργείτε η νέα σύνοδος από τις παρακάτω δύο γραμμές κώδικα

```
java.util.Date date=new java.util.Date();
```

```
String time=date.toLocaleString();
```

και αποθηκεύεται στην μεταβλητή logintime. Η διεύθυνση IP του πελάτη αποθηκεύεται στην τιμή remoteIp, αφού την ανακτήσουμε από την μέθοδο

```
request.getRemoteAddr();
```

τέλος αποθηκεύουμε στην μεταβλητή u\_id την τιμή -1.

Όλες οι παραπάνω μεταβλητές που αποθηκεύονται, θα χρειαστούν για ενημέρωση των πινάκων σε περιπτώσεις χρήσης της δικτυακής πύλης από τον επισκέπτη.

Όπως είπαμε, εκτός από την δομή των δεδομένων της συνόδου, ενημερώνονται και πίνακες από τη βάση δεδομένων του HealLink.

Ο πίνακας Sessions κρατάει τα αναγνωριστικά των συνόδων που είναι ενεργές. Αποτελείται από 4 πεδία. Το πεδίο session που είναι και το κύριο κλειδί, είναι τύπου varchar (200) και αποθηκεύει το αναγνωριστικό κωδικό της συνόδου. Το πεδίο u\_id είναι τύπου integer και κρατάει το κωδικό του μέλους που είναι συνδεδεμένος με τη σύνοδο. Στην περίπτωση που δεν είναι μέλος αλλά επισκέπτης, τότε η τιμή που αποθηκεύετε είναι η -1. Το username, τύπου varchar(80) κρατάει το όνομα χρήστη του μέλους που είναι συνδεδεμένος με την σύνοδο. Εδώ σε περίπτωση που δεν είναι μέλος τότε αποθηκεύει την τιμή guest. Τέλος το πεδίο time, τύπου varchar(200), κρατάει την χρονική στιγμή που δημιουργήθηκε η σύνοδος. Οι τιμές u\_id, username και time αντιστοιχούν στις αντίστοιχες τιμές των μεταβλητών u\_id, email και logintime της συνόδου όπως βλέπουμε στην παρακάτω εντολή SQL.

```
insert into db2admin.sessions(session,userid,username,time) values
(""+session.getId()+"
,"+Integer.parseInt(session.getAttribute("u_id").toString())+",
+session.getAttribute("email").toString()+"','"+session.getAttribute("logintime").toStr
ing()+"");
```



Ο πίνακας user\_stats στην έκδοση του HealLink v2.0 κρατούσε στατιστικά στοιχεία για τα μέλη του HealLink. Στην νέα έκδοση ο πίνακας αυτός ενημερώνεται και στην περίπτωση που ο χρήστης είναι επισκέπτης. Ο πίνακας αυτός αποτελείται από τα πεδία email, logintime, ip, logouttime, session. Στην περίπτωση που ο χρήστης είναι επισκέπτης και δημιουργείται η νέα σύννοδος, τότε οι τιμές που βρίσκονται στις μεταβλητές email, logintime, remoteIp αποθηκεύονται στον πίνακα user\_stats σύμφωνα με την παρακάτω πρόταση

```
String sql="insert into db2admin.user_stats(email,logintime,ip,session) values ('"+session.getAttribute("email").toString()+"', '"+session.getAttribute("logintime").toString()+"', '"+session.getAttribute("remoteIp").toString()+"', '"+session.getId()+"");
```

Συμπερασματικά με το που γίνει επίσκεψη στη πύλη HealLink, δημιουργείται μία νέα σύννοδος με χαρακτηριστικά επισκέπτη και έπειτα ακολουθούν τα παρακάτω σενάρια.

### **3.2.1 Ο επισκέπτης είναι μέλος και πραγματοποιεί είσοδο στο υπό περιβάλλον του, χωρίς να πραγματοποιήσει οποιαδήποτε δραστηριότητα σαν επισκέπτης**

Ο επισκέπτης με το που εισέρχεται στη δικτυακή πύλη, επιλέγει να συνδεθεί στο υπό περιβάλλον του HealLink κατευθείαν, χωρίς να επισκεφθεί κανένα περιοδικό κατά τη διάρκεια που η σύννοδος του, τον έχει καταγεγραμμένο σαν επισκέπτη. Στην περίπτωση αυτή, με την πραγματοποίηση της εισόδου του στο υπό περιβάλλον, αλλάζουν τόσο χαρακτηριστικά της συνόδου, όσο και δεδομένα στη βάση δεδομένων.

Για τη σύννοδο έχουμε τις αλλαγές που περιγράφονται παρακάτω και σύμφωνα με τον ακόλουθο κώδικα.

```
session.setAttribute("name",name);
```

```
session.setAttribute("u_id",u_id);
```

```
session.setAttribute("email",email);
```

όπου οι τιμές u\_id, name και email αντιστοιχούν στις ανάλογες τιμές του μέλους που πραγματοποίησε τη είσοδο στο σύστημα. Αυτές οι τιμές ανακτώνται από τη βάση δεδομένων και συγκεκριμένα από τον πίνακα user, μετά την πραγματοποίηση της αυθεντικοποίησης του χρήστη με το σύστημα.

Οι πίνακες που ενημερώνονται κατά την είσοδο του χρήστη είναι οι sessions και user\_stats. Στον πίνακα sessions γίνεται ενημέρωση της εγγραφής με το ανάλογο αναγνωριστικό συνόδου, όπου αλλάζουν οι τιμές των πεδίων u\_id και username στις τιμές που αντιστοιχούν στον χρήστη που πραγματοποίησε την είσοδο όπως φαίνεται παρακάτω

```
String          sql="update          db2admin.sessions          set
userid="+session.getAttribute("u_id").toString()+",
username="" +session.getAttribute("email").toString()+"          where
session="" +currentSession+""";
```

Η τιμή που περιέχει η currentSession έχει καταχωρηθεί πιο πριν από την πρόταση

```
currentSession=session.getId();
```

Στον πίνακα user\_stats ενημερώνονται τα πεδία email, session σύμφωνα με την παρακάτω πρόταση

```
sql="update db2admin.user_stats set  
email='"+session.getAttribute("email").toString()+"', session='ok' where  
session='"+currentSession+"'";
```

Το πεδίο session παίρνει την τιμή “ok” για να μην υπάρχει περίπτωση ύπαρξης συνόδου με το ίδιο αναγνωριστικό, γεγονός που θα οδηγούσε σε λανθασμένη ενημέρωση των αντίστοιχων πεδίων.

Σε αυτό το σενάριο υπάρχουν δυο εκδοχές αποσύνδεσης του χρήστη από το σύστημα και ξεχωριστά η κάθε μία περίπτωση ενεργοποιεί τον ανάλογο μηχανισμό διαχείρισης της συνόδου.

Στην περίπτωση που ο χρήστης αποσυνδεθεί από το σύστημα χρησιμοποιώντας την επιλογή έξοδος, όπως φαίνεται στην παρακάτω εικόνα 3.1, τότε ακολουθεί η διαδικασία όπου

HELLENIC ACADEMIC LIBRARIES LINK  
HEAL LINK  
ΔΙΚΤΥΟ ΕΛΛΗΝΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ

Αρχική Προσωπικά Στοιχεία **Εξόδος**

**Καλώς ήρθατε**  
Υπάτιος  
Τετάρτη 9 Νοεμβρίου  
Συνολικές Συνδέσεις: 111  
Λεπτομέρειες

**Αναζήτηση**  
● Γρήγορη Αναζήτηση  
  
Αναζήτηση  
● Σύνθετη Αναζήτηση

**Προφίλ**  
● Αλφαβητικός Κατάλογος  
● Θεματικές Κατηγορίες  
● Αυτόματη Ειδοποίηση

**Κατασκευή και Διαχείριση της πύλης:**  
Φλώρου Πόπη  
Πιστηρίγγας Λεωνίδας  
Χατζησταύρου Βλάσης

ΕΛΛΑΣ 2006  
ΠΑΙΔΕΙΑ ΜΠΡΟΣΤΑ  
2<sup>ο</sup> Επιχειρησιακό Πρόγραμμα  
Εκπαίδευσης και Αρχικής  
Επαγγελματικής Κατάρτισης

Εικόνα 3.1 Έξοδος από το σύστημα

ενημερώνονται οι πίνακες user\_stats και sessions και τερματίζεται η σύννοδος. Από τον πίνακα sessions διαγράφεται η εγγραφή της συνόδου με τα διακεκριμένο αναγνωριστικό σύμφωνα με την πρόταση

```
String sql="delete from db2admin.sessions where session='"+session.getId()+" and
userid='"+session.getAttribute("u_id");"
```

Στον πίνακα user\_stats γίνεται ενημέρωση της αντίστοιχης εγγραφής, όπου καταγράφεται η χρονική στιγμή εξόδου του χρήστη

```
sql="update db2admin.user_stats set logouttime='"+logouttime+"' where email='"+  
session.getAttribute("email")+"' and  
logintime='"+session.getAttribute("logintime").toString()+"' and  
ip='"+request.getRemoteAddr()+"'";
```

Τέλος γίνεται η διαγραφή της συνόδου πραγματοποιώντας κλήση της μεθόδου `invalidate()` και προώθηση του χρήστη στην αρχική σελίδα του HealLink.

Στην περίπτωση που ο χρήστης δεν αποσυνδεθεί με την επιλογή “Έξοδος” αλλά αποχωρήσει απλά από τον σταθμό εργασίας του, κλείνοντας απλά τον browser του ή αφήνοντας τον ανοιχτό ή ακόμη ανοίγοντας μια νέα σελίδα έξω από τον δικτυακό χώρο του HealLink, τότε η σύνοδος του παραμένει ενεργή για 1800 δευτερόλεπτα, με το πέρας των οποίων η σύνοδος λήγει. Τη στιγμή που πραγματοποιείται η λήξη της συνόδου, ενεργοποιείται ο ακροαστής `MyListener`. Ο ακροαστής `MyListener` είναι υλοποίηση της κλάσης `HttpSessionListener` κληρονομώντας της μεθόδους `sessionCreated` και `sessionDestroyed`. Η μέθοδος `sessionDestroyed` καλείται με το που λήγει μια σύνοδος και μπορούμε να την εκμεταλευτούμε για να διαχειριστούμε την ομαλή έξοδο του χρήστη από το σύστημα. Αυτό γίνεται, γιατί στην περίπτωση που περιγράψαμε πιο πάνω, θα είχαμε δυο λανθασμένες πληροφορίες στους πίνακες `user_stats` και `sessions`. Στον πίνακα `user_stats` δεν θα είχαμε τιμή στο πεδίο `logouttime` και επομένως δεν θα γνωρίζαμε τη χρονική στιγμή που ο χρήστης αποσυνδέθηκε από την πύλη. Στον πίνακα `sessions` θα είχαμε εγγραφή με αναγνωριστικό συνόδου το οποίο έχει λήξει και επομένως η εγγραφή αυτή δεν θα έπρεπε να υπάρχει.

Στην μέθοδο `sessionDestroyed` η πρώτη γραμμή έχει να κάνει με την αποθήκευση του αναγνωριστικού της συνόδου στην μεταβλητή `ses` που είναι τύπου `String`. Έπειτα γίνεται κλήση σε τρεις μεθόδους που εμπλουτίζουν τον ακροαστή `MyListener`. Η μέθοδος `getVars(ses)` δημιουργεί ερωτήματα στη βάση με σκοπό να

αποθηκεύσει στις μεταβλητές `timeIn`, `email`, `ip` τις αντίστοιχες τιμές από τα πεδία των πινάκων `sessions`, `user_stats`. Συγκεκριμένα τα ερωτήματα που γίνονται στη βάση φαίνονται στις παρακάτω δυο προτάσεις.

```
String sql="select username, time from db2admin.sessions where session='"+ses+"'";
```

```
String sql="select ip from db2admin.user_stats where email='"+email+"' and logintime='"+ timeIn +"'";
```

Η μέθοδος `updateStats` κάνει ενημέρωση του πίνακα `user_stats` στην τιμή του πεδίου `logouttime`. Για να κάνει ενημέρωση του πίνακα `user_stats` πρώτα παίρνει μαζεύει τις πληροφορίες που χρειάζεται με ένα ερώτημα στη βάση και με κλήση της μεθόδου `getLastAccessedTime()`. Το ερώτημα που πραγματοποιεί στη βάση έχει να κάνει με την ανάκτηση του `username` και της ώρας εισόδου του χρήστη.

```
String sql="select username,time from db2admin.sessions where session='"+ses+"'";
```

Η κλήση της μεθόδου `getLastAccessedTime()` γίνεται πιο πριν και αποθηκεύει στην μεταβλητή, τύπου `long`, `minutes` την τελευταία φορά που ο πελάτης είχε πρόσβαση στη σύνοδο.

```
long minutes = se.getSession().getLastAccessedTime();
```

Από την τιμή της μεταβλητής `minutes` δημιουργούμε την χρονική στιγμή εξόδου του χρήστη από την πύλη.

HEAL-Link v3.0

```
java.util.Date date=null;
```

```
date=new Date(minutes);
```

```
String timeOut=date.toLocaleString();
```

Και κάνουμε την ενημέρωση του πίνακα user\_stats

```
String sql="update db2admin.user_stats set logouttime='"+timeOut+"' where  
email='"+email+"' and logintime='"+timeIn+"' and ip='"+ip+"'";
```

Τέλος η μέθοδος DeleteSession() διαγράφει από τον πίνακα sessions τη σύνοδο που ετοιμάζεται να λήξει.

Σε κάθε περίπτωση που προκύψει κάποιο σφάλμα από τις μεθόδους του ακροαστή MyListener τότε το σφάλμα αυτό καταγράφεται με την βοήθεια της κλάσης FileWrite, σε ένα αρχείο κειμένου, προκειμένου να χρησιμοποιηθεί για αποσφαλμάτωση.

### **3.2.2 Ο επισκέπτης είναι μέλος και πραγματοποιεί είσοδο στο υπό περιβάλλον του, ενώ πριν έχει επισκεφθεί περιοδικά**

Στην περίπτωση που ο επισκέπτης είναι μέλος, αλλά πριν την εισαγωγή του στο σύστημα έχει επισκεφθεί περιοδικά, τότε οι επισκέψεις αυτές έχουν καταγραφεί

στον πίνακα `journal_stats` με την τιμή “guest” στο πεδίο `email`. Για να διορθώσουμε τέτοιου είδους εγγραφές, με την εισαγωγή του χρήστη στο υπο περιβάλλον του ακολουθούν οι επόμενες ενέργειες.

Ενημερώνονται οι πίνακες `user_stats`, `journal_stats` και `sessions`. Στον πίνακα `user_stats` ενημερώνεται η εγγραφή της συνόδου, αλλάζοντας την τιμή του πεδίου `email` στο όνομα χρήστη του μέλους και την τιμή `session` στην τιμή “ok”, ώστε να μην υπάρξει σύγχυση σε επόμενες συνόδους.

```
sql="update db2admin.user_stats set email='"+session.getAttribute("email").toString()+"', session='ok' where session='"+currentSession+"'";
```

Στον πίνακα `journal_stats` αλλάζουν όλες οι εγγραφές που έχουν το αναγνωριστικό της τρέχουσας συνόδου, ενημερώνοντας την τιμή του πεδίου `email`, στο όνομα χρήστη του μέλους και την τιμή της συνόδου στην τιμή “ok”.

```
sql="update db2admin.journal_stats set email='"+session.getAttribute("email").toString()+"', session='ok' where session='"+currentSession+"'";
```

Τέλος στον πίνακα `sessions` ενημερώνονται οι τιμές των πεδίων `userid` και `username` στις τιμές του χρήστη που πραγματοποίησε την είσοδό του στην πύλη.

```
String sql="update db2admin.sessions set userid = " + session.getAttribute("u_id").toString()+"', username='"+session.getAttribute("email").toString()+"' where session = '"+currentSession+"'";
```



Τέλος η συμπεριφορά του συστήματος στην χρήστη είναι όμοια με αυτή που περιγράψαμε στην ενότητα 3.2.1.

### **3.2.3 Ο επισκέπτης χρησιμοποιεί την πύλη και τις υπηρεσίες που προσφέρει και αποχωρεί**

Σε αυτή την περίπτωση η πύλη παρακολουθεί και καταγράφει τις κινήσεις του επισκέπτη στους πίνακες `journal_stats`, `user_stats` και `sessions` χρησιμοποιώντας το αναγνωριστικό της συνόδου, το όνομα “`guest`” και την `ip` διεύθυνση του επισκέπτη για να τον αναγνωρίζει. Μετά την αποχώρησή του, η σύνοδος λήγει μετά από 30 λεπτά, όπου καλείται ο ακροαστής `MyListener` για να ολοκληρώσει τη σύνοδό του όπως περιγράψαμε στην ενότητα 3.2.1.

## **3.3 Βάση Δεδομένων του HealLink**

Η δομή της Βάσης Δεδομένων του HealLink v3.0 είναι η ίδια με την δομή της έκδοσης 2.0. Οι μοναδικές αλλαγές βρίσκονται στους πίνακες `user_stats`, `sessions`, `journal_stats`. Ακολουθεί μια σύντομη περιγραφή των πινάκων της εφαρμογής.

- **Πίνακας Publisher** – Περιέχει τους εκδοτικούς οίκους με τους οποίους συνεργάζεται η πύλη.
- **Πίνακας Category** – Περιέχει τις κατηγορίες των περιοδικών.
- **Πίνακας Subcategory** – Περιέχει τις υποκατηγορίες που έχει κάθε κατηγορία των περιοδικών.

- **Πίνακας Subject** – Περιέχει τα θέματα που έχει κάθε υποκατηγορία.
- **Πίνακας Subject\_Subcat** – Περιέχει εγγραφές που δηλώνουν ποια θέματα ανήκουν σε ποιες κατηγορίες.
- **Πίνακας Journal** – Περιέχει τα περιοδικά που υπάρχουν στην πύλη και μπορούν να τα επισκεφθούν οι χρήστες.
- **Πίνακας J\_Subcat** – Περιέχει εγγραφές για το ποια περιοδικά ανήκουν σε ποιες υποκατηγορίες.
- **Πίνακας J\_Subject** – Περιέχει εγγραφές για το ποια περιοδικά ανήκουν σε ποιο θέμα.
- **Πίνακας Status** – Περιέχει την ιδιότητα που έχουν τα μέλη της πύλης, π.χ. Φοιτητές, Καθηγητές κ.α.
- **Πίνακας Institution** – Περιέχει εγγραφές με τα ιδρύματα που είναι εξουσιοδοτημένα να έχουν πρόσβαση στις υπηρεσίες του HealLink.
- **Πίνακας Counter** – Περιέχει την τιμή του μετρητή επισκέψεων στην κεντρική πύλη.
- **Πίνακας User** – Περιέχει τα στοιχεία των χρηστών που έχουν εγγραφεί στην πύλη.
- **Πίνακας User\_Profile** – Περιέχει εγγραφές του προφίλ των χρηστών. Ποια περιοδικά έχουν στο προφίλ τους.
- **Πίνακας U\_J\_NEW** – Περιέχει τις εγγραφές για αυτόματη ειδοποίηση νέων περιοδικών.
- **Πίνακας U\_Subj\_New** – Περιέχει τις εγγραφές για αυτόματη ειδοποίηση νέων θεμάτων.
- **Πίνακας Sessions** – Περιέχει τις συνόδους που είναι ενεργές στην κάθε χρονική στιγμή. Ο πίνακας Sessions στην έκδοση 2.0 είχε την παρακάτω

δομή.

```
CREATE TABLE SESSIONS(Session varchar(200) NOT NULL,USERID  
INTEGER NOT NULL WITH DEFAULT 0,USERNAME VARCHAR(80)  
NOT NULL, TIME VARCHAR(200) not null, primary key (session, userid) )
```

Για να λειτουργήσει η έκδοση 3.0 αφαιρέθηκε το userid από τα κύρια κλειδιά ώστε να μπορούμε να έχουμε πολλές αναφορές στον χρήστη "Guest" με userid=-1.

- **Πίνακας Journal\_Stats** – Περιέχει τα στατιστικά των επισκέψεων των περιοδικών. Στην έκδοση 2.0 η δομή του πίνακα ήταν η παρακάτω

```
CREATE TABLE JOURNAL_STATS("EMAIL" VARCHAR(80) NOT  
NULL, "IP" VARCHAR(25) NOT NULL,"TIME" VARCHAR (50) NOT  
NULL, "TYPE" VARCHAR(10) NOT NULL,"JOURNAL" VARCHAR  
(255) NOT NULL, PRIMARY KEY(IP,TIME,TYPE,JOURNAL)) DATA  
CAPTURE NONE
```

Για τις ανάγκες της έκδοσης 3.0 είχαμε την προσθήκη του πεδίου session που περιέχει το αναγνωριστικό της συνόδου. Αυτή η αλλαγή πραγματοποιήθηκε για να μπορούμε να παρακολουθούμε και να καταγράφουμε τον χρήστη σε περίπτωση που επισκέφθηκε περιοδικά σαν επισκέπτης και μετά έκανε εισαγωγή στο σύστημα.

- **Πίνακας Profile\_Stats** – Περιέχει στατιστικά στοιχεία για την είσοδο του χρήστη στην πύλη. Καταγράφει την διεύθυνση IP και την χρονική στιγμή για κάθε μέλος.
- **Πίνακας User\_Stats** – Περιέχει στατιστικά για την συμπεριφορά των χρηστών στην πύλη. Η δομή του πίνακα στην έκδοση 2.0 είχε την παρακάτω δομή.

```
CREATE TABLE USER_STATS("EMAIL" VARCHAR(80) NOT NULL,"LOGINTIME" VARCHAR(50) NOT NULL, "IP" VARCHAR(25) NOT NULL, "LOGOUTTIME" VARCHAR (50), PRIMARY KEY (EMAIL,LOGINTIME,IP)) DATA CAPTURE NONE
```

Για την ανάγκη της έκδοσης 3.0 χρειάστηκε να βγάλουμε το πεδίο email από τα κύρια κλειδιά και να προσθέσουμε το πεδίο session που καταγράφει το αναγνωριστικό της συνόδου. Αυτή η καταχώρηση πραγματοποιήθηκε για να υπάρχει η δυνατότητα για την παρακολούθηση και καταγραφή της συμπεριφοράς των μελών.

- **Πίνακας Visits\_Stats** – Πίνακας που περιέχει εγγραφές σχετικά με τις επισκέψεις στην κεντρική πύλη.
- **Πίνακας New** – Πίνακας που περιέχει τα νέα της πύλης του HealLink.
- **Πίνακας Switches** – Πίνακας που περιέχει διακόπτες για τις διάφορες υπηρεσίες του HealLink, όπως διαθεσιμότητα πύλης, καταγραφή στατιστικών.
- **Πίνακας Admin** – Περιέχει το όνομα χρήστη και το συνθηματικό του διαχειριστή της πύλης που έχει πρόσβαση στο σελίδα διαχείρισης του HealLink.

### 3.4 Παράδειγμα από το HEAL-Link v3.0

#### 3.4.1 loginuser.jsp

Σε αυτή τη σελίδα JSP πραγματοποιείται η διαδικασία εισαγωγής του χρήστη στο υπο περιβάλλον του. Καλείται όταν ο χρήστης εισάγει το όνομα χρήστη και συνθηματικό και επιλέξει είσοδος. Στην αρχή της σελίδας έχουμε τη δήλωση

```
<%@page session="true"%>
```

που δηλώνουν την ενεργοποίηση των συνόδων για την εφαρμογή μας. Στις γραμμές 2-4 έχουμε την συμπερίληψη πακέτων της java που χρειάζονται για να χρησιμοποιήσουμε ορισμένες μεθόδους της.

```
<%@page import="java.sql.*"%>
```

για την χρησιμοποίηση μεθόδων για σύνδεση στη βάση δεδομένων, δημιουργία και εκτέλεση ερωτημάτων και αποθήκευση τους σε αντικείμενα για επεξεργασία.

```
<%@page import="java.util.*"%>
```

για την χρησιμοποίηση μεθόδων όπως Date

Στις γραμμές 5 έως 17 έχουμε την συμπερίληψη δύο JavaBean αντικειμένων, των UserInfo και dbcon.

```
<jsp:useBean id="loginParams" class="HealLink.UserInfo">  
  <jsp:setProperty name="loginParams"  
    property="username"  
    param="HealUserName"/>  
  <jsp:setProperty name="loginParams"  
    property="password"  
    param="HealPass"/>  
</jsp:useBean>
```

Το JavaBean UserInfo του πακέτου HealLink περιέχει το όνομα χρήστη και το συνθηματικό που πληκτρολόγησε ο χρήστης, στην φόρμα της αρχικής σελίδας για να πραγματοποιήσει την εισαγωγή του στο σύστημα. Του δίνουμε το id loginParams μέσω του οποίου θα μπορούμε να έχουμε πρόσβαση στις μεταβλητές username και password.

```
<jsp:useBean id="db" class="HealLink.dbcon">
</jsp:useBean>
```

Αυτό το JavaBean δημιουργεί ένα αντικείμενο της κλάσης dbcon του πακέτου HealLink που χρησιμοποιείται για να πραγματοποιήσει σύνδεση στη βάση δεδομένων, να εκτελέσει ερωτήματα και ενημερώσεις και τέλος να τερματίσει την σύνδεση. Του δίνουμε το id dbcon μέσω του οποίου θα έχουμε πρόσβαση στις μεθόδους connectDB(), closeDB(), executeQuery(), executeUpdate().

```
<jsp:include page="/Sessions.jsp" />
```

Εδώ έχουμε την συμπερίληψη της σελίδας Sessions.jsp η οποία ελέγχει για το αν υπάρχει ενεργή σύννοδος, ενώ αν δεν υπάρχει την δημιουργεί.

```
<%
java.util.Date date=new java.util.Date();
String time=date.toLocaleString();
%>
```

Δημιουργία ενός αντικειμένου date και ενός αντικειμένου String που περιέχει την συμβολοσειρά της ημερομηνίας date.

```
int flag=0;
boolean success = false;
String name="";
String u_id="";
String email="";
```

Δημιουργία βοηθητικών μεταβλητών για την διαδικασία ελέγχου εξουσιοδότησης των χρηστών. Η μεταβλητή. Θα δούμε παρακάτω πως χρησιμοποιούνται.

```
if (!loginParams.getUsername().equals("")) {
```

Εδώ ελέγχουμε αρχικά αν ο χρήστης έχει συμπληρώσει το πεδίο όνομα χρήστη, ώστε να συνεχίσουμε την διαδικασία ελέγχου. Εάν ο χρήστης δεν έχει συμπληρώσει το πεδίο αυτό τότε του εμφανίζεται ένα μήνυμα λάθους.

```
db.connectDB());
```

Σύνδεση με τη βάση δεδομένων.

```
String sql="select EMAIL,PASSWORD,FNAME, u_id FROM db2admin.USER  
WHERE lower(email)='"+loginParams. getUserLower()+"''";
```

```
ResultSet rs=db.executeQuery(sql);
```

Δημιουργία και εκτέλεση ερωτήματος στη βάση όπου ζητάμε εάν υπάρχει μέλος με το όνομα χρήστη που δηλώθηκε στην φόρμα εισόδου και αποθήκευση του αποτελέσματος στο αντικείμενο rs τύπου ResultSet.

```
if (rs.next()) {  
    if (rs.getString("password").trim().equals(loginParams.getPassword().trim())) {  
        success = true;  
    }  
    name=rs.getString("fname").trim();  
    u_id=rs.getString("u_id").trim();  
    email=rs.getString("email").trim();  
} else  
out.println("<META HTTP-EQUIV=\"Refresh\" CONTENT=\"0\" URL=  
MyHealError.jsp ?username = bad\">");
```

Εδώ σε περίπτωση που υπάρχει αποτέλεσμα στο αντικείμενο rs – που το ελέγχουμε με την συνθήκη if(rs.next()) – πραγματοποιούμε ένα δεύτερο έλεγχο για το συνθηματικό του χρήστη, με την συνθήκη “if (rs.getString("password").trim().equals(loginParams.getPassword().trim()))”. Εάν το συνθηματικό είναι ίδιο με το συνθηματικό που υπάρχει στη βάση δεδομένων γι’ αυτόν τον χρήστη, τότε η τιμή της μεταβλητής success γίνεται true, και αποθηκεύουμε στις μεταβλητές name, u\_id και email τις τιμές fname, u\_id και email που ανακτούμε από τη βάση δεδομένων. Εάν δεν υπάρχει μέλος με το όνομα χρήστη που δηλώθηκε στη φόρμα εισόδου τότε ο χρήστης μεταφέρεται στη σελίδα MyHealError.jsp όπου και του εμφανίζεται το ανάλογο μήνυμα λάθους.

Στην περίπτωση που το όνομα χρήστη είναι σωστό αλλά υπάρχει λάθος στο συνθηματικό τότε ο χρήστης μεταφέρεται στην σελίδα MyHealError.jsp με το ανάλογο μήνυμα λάθους.

```
if (!success) {
    out.println("<META HTTP-EQUIV=\"Refresh\" CONTENT=\"0; URL=
MyHealError.jsp? username=true\">");
}
```

Στην περίπτωση που η διαδικασία ελέγχου είναι επιτυχής, τότε δημιουργούμε ένα ερώτημα στη βάση για να επιβεβαιώσουμε ότι δεν υπάρχει, στον πίνακα sessions, εγγραφή με το αναγνωριστικό της τρέχουσας συνόδου και το userid του χρήστη που θέλει να πραγματοποιήσει την είσοδό του στο σύστημα.

```
sql="select session,userid from db2admin.sessions where
session='"+session.getId()+"'";
rs=db.executeQuery(sql);
```

Εάν υπάρχει εγγραφή στον πίνακα sessions τότε εξετάζουμε τις περιπτώσεις όπου το userid είναι ίσο με -1, που ανήκει στον χρήστη επισκέπτη και αναθέτουμε την τιμή 0 στην μεταβλητή flag, Αν το userid ανήκει στον χρήστη, οπότε έχουμε το σενάριο όπου πραγματοποίησε είσοδο, ενώ ήταν είδη στο σύστημα, αναθέτουμε την τιμή 1 στην μεταβλητή flag και τέλος την περίπτωση που δεν υπάρχει τέτοια εγγραφή,



γεγονός απίθανο, αναθέτουμε την τιμή 2 στην μεταβλητή flag. Τέλος αποδεσμεύουμε το αντικείμενο rs και κλείνουμε την σύνδεση μας με την βάση δεδομένων.

```
if (rs.next()) {
    String thisSession=rs.getString("session");
    if (rs.getString("userid").equals(session.getAttribute("u_id").toString()) && (
u_id.equals ((session.getAttribute("u_id").toString()))&&(!u_id.equals("-1"))) ) {
        flag=1;
    } else if (session.getAttribute("u_id").toString().equals("-1")) {
        flag=0;
    }
    else flag=2;
}
rs.close();
db.closeDB();
}
```

Στη συνέχεια έχουμε τα σενάρια για τις διάφορες τιμές της μεταβλητής flag. Στην περίπτωση που η flag ισούται με 1 τότε μεταφέρουμε τον χρήστη στο υπο περιβάλλον του, χωρίς να πραγματοποιήσουμε καμία ενημέρωση στη βάση, γιατί ο χρήστης έχει είδη πραγματοποιήσει την είσοδό του στο σύστημα.

```
if (flag==1) out.println("<META HTTP-EQUIV=\"Refresh\" CONTENT=\"0; URL
= logged.jsp ?id ="+session.getId()+">");
```

στην περίπτωση που η τιμή είναι ίση με 2, γεγονός απίθανο ( ο χρήστης προσπάθησε να μπει στο υπο περιβάλλον του, γράφοντας το αντίστοιχο url στο πεδίο διευθύνσεων του browser), μεταφέρει τον χρήστη σε μια σελίδα με ένα μήνυμα λάθους.

```
else if (flag==2) out.println("<META HTTP-EQUIV=\"Refresh\" CONTENT=\"0;
URL = MyHealErrorLogged.jsp?username=true\">");
```

Στην περίπτωση που η τιμή είναι ίση με 0, τότε σημαίνει ότι ο χρήστης πραγματοποιεί κανονική εισαγωγή στο σύστημα. Συνεχίζουμε με τον έλεγχο του συνθηματικού και την ανάθεση τιμών στη σύννοδο του χρήστη.

```

else if (flag==0) {
    if (success) {

        String currentSession=session.getId();
        String ip=request.getRemoteAddr();
        session.setMaxInactiveInterval(1800);
        session.setAttribute("name",name);
        session.setAttribute("u_id",u_id);
        session.setAttribute("email",email);
        session.setAttribute("remoteIp",ip);
    }
}

```

Στη συνέχεια πραγματοποιούμε σύνδεση με τη βάση και δημιουργία ερωτήματος σχετικά με τους διακόπτες στατιστικών της πύλης. και ανάλογα πραγματοποιούμε ενημέρωση στον πίνακα user\_stats.

```

String flag1="0";
ResultSet res=db.executeQuery("select stats_switch from db2admin.switches");
if (res.next()) {
    flag1=res.getString("stats_switch");
}
res.close();
if (flag1.equals("1")) {
    String sql=null;
    if (flag==0) {
        sql="update db2admin.user_stats set
email='"+session.getAttribute("email").toString()+"', session='ok' where
session='"+currentSession+"'";
    }
    else {
        session.setAttribute("logintime",time);
    }
}

```

```

        sql="insert into db2admin.user_stats(email,logintime,ip,logouttime,session)
values(""+session.getAttribute("email").toString()+""+session.getAttribute("logintime").toString()+""+request.getRemoteAddr()+"";'','ok');
    }
    int rs1=db.executeUpdate(sql);

}

```

Η ενημέρωση του πίνακα user\_stats έχει να κάνει με την χρονική στιγμή και από ποια ip διεύθυνση πραγματοποιήσε εισαγωγή στο σύστημα ο χρήστης. Σε περίπτωση που στον πίνακα sessions υπάρχει αναγνωριστικό συνόδου με αυτό που έχει ο χρήστης που προσπαθεί να πραγματοποιήσει εισαγωγή, τότε η τιμή της μεταβλητής flag είναι 0 και πραγματοποιούμε update στον πίνακα user\_stats αλλάζοντας το email στο email του χρήστη. Σε άλλη περίπτωση πραγματοποιούμε νέα εισαγωγή στον πίνακα user\_stats με τα στοιχεία του χρήστη.

Στη συνέχεια πραγματοποιούμε ενημερώσεις στους πίνακες sessions και journal\_stats για να διορθώσουμε τυχόν εγγραφές που καταχωρήθηκαν όσο ο χρήστης ήταν επισκέπτης.

```

String          sql="update          db2admin.sessions          set
userid="" +session.getAttribute("u_id").toString()+",
username="" +session.getAttribute("email").toString()+""          where
session="" +currentSession+""";

```

```

int rs1=db.executeUpdate(sql);

```

```

        sql="update          db2admin.journal_stats          set
email="" +session.getAttribute("email").toString()+",          session='ok'          where
session="" +currentSession+""";
        rs1=db.executeUpdate(sql);

```

Τέλος υπάρχει ένα ερώτημα στη βάση, για την αναζήτηση νέων περιοδικών για τον χρήστη-μέλος με βάση τις επιλογές που έχει κάνει από την υπηρεσία αυτόματη ενημέρωση της Πύλης. Εάν υπάρχουν τότε εισάγουμε ένα νέο χαρακτηριστικό στη σύνοδο με το όνομα newj και την τιμή “true”, ενώ αν δεν υπάρχει του δίνουμε την τιμή “false”.

```
db.connectDB();
sql="select j.j_id from db2admin.journal j, db2admin.j_subject js,
db2admin.u_subj_new us where j.j_id=js.j_id and us.s_id=js.s_id and
j.date_entered>(cast(''+mydate+'' as date) - 60 DAYS) and us.u_id="+u_id+" and
J.j_id not in (Select j_id from db2admin.u_j_new where u_id="+u_id+)";
ResultSet rsc=db.executeQuery(sql);
if(rsc.next()) {
    session.setAttribute("newj","true");
} else {
    session.setAttribute("newj","false");
}
db.closeDB();
```

Τέλος, ο χρήστης μεταφέρεται στο υπο περιβάλλον του και είναι έτοιμος να χρησιμοποιήσει τις υπηρεσίες του HealLink.

```
out.println("<META HTTP-EQUIV=\"Refresh\" CONTENT=\"0; URL=
logged.jsp?id = " +session.getId()+">");
```

### 3.4.2 dbcon.java

Η κλάση dbcon χρησιμοποιείται από τις σελίδες του HealLink για τις συναλλαγές με την βάση δεδομένων της πύλης, όπως σύνδεση, αποσύνδεση, πραγματοποίηση ερωτημάτων, ενημερώσεων, διαγραφών. Αποτελείται από τέσσερις μεθόδους που θα αναλύσουμε παρακάτω. Όπως φαίνεται στην πρώτη γραμμή, η κλάση dbcon ανήκει στο πακέτο HealLink, ενώ στις γραμμές 2 και 3 έχουμε την εισαγωγή των πακέτων io.\* και sql.\* της java.

```
package HealLink;  
import java.sql.*;  
import java.io.*;
```

αμέσως παρακάτω ξεκινάει το σώμα της κλάσης dbcon και η δήλωση τεσσάρων αντικειμένων για την κλάση.

```
public class dbcon{  
  
private Connection con;
```

Αντικείμενο που περιέχει την σύνδεση με τη βάση δεδομένων και μας επιτρέπει την εκτέλεση των συναλλαγών.

```
private ResultSet rs;
```

Αντικείμενο στο οποίο αποθηκεύονται τα αποτελέσματα των ερωτημάτων.

```
private Statement stmt;
```

Αντικείμενο που χρησιμοποιείται για να προετοιμάσει τα ερωτήματα που πρόκειται να διεκπεραιώσουμε με τη βάση.

```
FileWrite log=new FileWrite();
```

Στιγμιότυπο της κλάσης FileWrite που γράψαμε για την ανάγκη της εφαρμογής, προκειμένου να καταγράφει τα παραγόμενα σφάλματα σε κάποιο αρχείο λαθών, το οποίο χρησιμοποιούμε για απόσφαλμάτωση.

Η πρώτη μέθοδος της κλάσης είναι η connectDB() τύπου void και πραγματοποιεί τη σύνδεση με τη βάση δεδομένων. Όλη η μέθοδος βρίσκεται σε ένα μπλοκ try ... catch για να χειριστούμε τυχόν εξαιρέσεις.

```
public void connectDB() {
    try {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

this.con=DriverManager.getConnection("jdbc:odbc:xxxxx","xxxxx","xxxxx");
        } catch(Exception e) { log.setFile("c:\\error.log"); log.write(e.toString());
log.close(); }
}
```

Για να πραγματοποιήσουμε την σύνδεση θα πρέπει πρώτα να δηλώσουμε τον οδηγό που θα χρησιμοποιήσουμε όπως φαίνεται στην γραμμή 3. Μετά πραγματοποιούμε τη σύνδεση με τη βάση και την αναθέτουμε στο αντικείμενο con που έχουμε δηλώσει στην αρχή της κλάσης. Τέλος σε περίπτωση εξαίρεση ή σφάλματος, καταγράφουμε το σφάλμα με τη βοήθεια του στιγμιότυπου log της κλάσης FileWrite. Με την μέθοδο setFile("c:\\error.log") δηλώνουμε το όνομα του αρχείου στο οποίο θα καταγράψουμε το σφάλμα, στο παράδειγμα μας το αρχείο είναι το error.log που βρίσκεται στον κατάλογο c:\ . Η μέθοδος write(e.toString()) καταγράφει το σφάλμα που περιέχει το αντικείμενο e. Και τέλος κλείνουμε το αρχείο για να υποθηκεύσουμε το περιεχόμενό του.

Η επόμενη μέθοδος κλείνει την σύνδεση με τη βάση που περιέχει το αντικείμενο con.

```
public void closeDB() {
    try {
        this.con.close();
```

```
        }catch(Exception e) { log.setFile("c:\\error.log); log.write(e.toString());  
log.close(); }  
}
```

Πάλι στην περίπτωση σφάλματος, το καταγράφουμε σε ένα αρχείο λαθών, αφού το σώμα της μεθόδου βρίσκεται σε ένα μπλοκ try ... catch.

Η μέθοδος executeQuery πραγματοποιεί ερωτήματα στη βάση και επιστρέφει τα αποτελέσματα στο αντικείμενο τύπου ResultSet. Και σε αυτή τη μέθοδο έχουμε μπλοκ try ... catch για την καταγραφή σφαλμάτων. Αρχικά χρησιμοποιούμε το αντικείμενο stmt για να δημιουργήσουμε και να προετοιμάσουμε το ερώτημα για τη βάση και στη συνέχεια εκτελούμε το ερώτημα και επιστρέφουμε το αντικείμενο rs, που περιέχει τα αποτελέσματα του ερωτήματος.

```
public ResultSet executeQuery(String sql) {  
    ResultSet buffSet=null;  
    try{  
        this.stmt=this.con.createStatement();  
        this.rs=this.stmt.executeQuery(sql);  
        return this.rs;  
    }catch(Exception e) { log.setFile("c:\\error.log); log.write(e.toString());  
log.close(); return buffSet; }  
}
```

Στην περίπτωση σφάλματος, θα πρέπει να επιστρέψουμε ένα αντικείμενο ResultSet. Γι' αυτό το λόγο στην αρχή της μεθόδου δηλώνουμε ένα αντικείμενο τύπου ResultSet με την τιμή null. Το ερώτημα δημιουργείται με την δήλωση

```
this.stmt=this.con.createStatement();
```

και εκτελείται με τη δήλωση

```
this.rs=this.stmt.executeQuery(sql);
```

Τέλος έχουμε την μέθοδο `updateQuery()` με την οποία πραγματοποιούμε ενημερώσεις και διαγραφές στη βάση. Η μέθοδος αυτή επιστρέφει την τιμή 1 σε περίπτωση επιτυχίας και την τιμή -1 σε περίπτωση αποτυχίας, ενώ συγχρόνως καταγράφει το σφάλμα στο αρχείο σφαλμάτων, όπως και οι πιο πάνω μέθοδοι. Όπως βλέπουμε η μέθοδος `executeUpdate` είναι πανομοιότυπη με την μέθοδο `executeQuery`. Η διαφορά τους βρίσκεται στον τύπο επιστρεφόμενου αντικειμένου της εκτέλεσης του sql ερωτήματος και στην μέθοδο εκτέλεσης.

```
public int executeUpdate(String sql) {  
    try{  
        this.stmt=this.con.createStatement();  
        this.stmt.executeUpdate(sql);  
        return 1;  
    }catch(Exception e) { log.setFile("c:\\error.log); log.write(e.toString());  
        log.close(); return -1; }  
}
```

Η δήλωση που εκτελεί το ερώτημα είναι η

```
this.stmt.executeUpdate(sql);
```



## ΚΕΦΑΛΑΙΟ 4<sup>ο</sup> : ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ

### 4.1 Κεντρική Πύλη

#### 4.1.1 Περιβάλλον

Στην παρακάτω εικόνα βλέπουμε την νέα αρχική οθόνη της πύλης. Η διασύνδεση αυτή σχεδιάστηκε από τον κ. Λεωνίδα Πισπιρίγγα, ο οποίος ήταν ο σχεδιαστής της διασύνδεσης της έκδοσης 2.0.

HEAL LIBRARIES LINK  
ΔΙΚΤΥΟ ΕΛΛΗΝΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ

Αρχική Πληροφορίες Για Βιβλιοθηκονόμους SELL

**Αναζήτηση**

- Γρήγορη Αναζήτηση
- Αναζήτηση
- Σύνθετη Αναζήτηση

**Υπηρεσίες**

- Αλφαβητικός Κατάλογος
- Θεματικές Κατηγορίες
- Βιβλιογραφικές Βάσεις
- Ηλεκτρονικά Βιβλία
- Εκδότες

**My HEAL-Link**

E-mail:

Κωδικός:

Είσοδος

- Νέος χρήστης
- Υπενθύμιση κωδικού
- Οδηγίες

**Ηλεκτρονικές Πηγές**

Μέσω αυτής της διαδικτυακής πύλης τα μέλη της Κοινότητας έχουν πρόσβαση σε πλήρες κείμενο ηλεκτρονικών περιοδικών και βιβλίων και σε βιβλιογραφικές βάσεις δεδομένων. Ο έλεγχος πρόσβασης γίνεται με αναγνώριση IP διευθύνσεων ασφαλείας από τους εκδότες χωρίς να μπλέκεται σε αυτήν τη διαδικασία η δημιουργία λογαριασμού **My-HEAL-Link**, που είναι μια υπηρεσία εξατομίκευσης.

Η νομοθεσία περί πνευματικής ιδιοκτησίας εφαρμόζεται για τις ηλεκτρονικές εκδόσεις με τον ίδιο τρόπο που εφαρμόζεται για τις έντυπες. Χρησιμοποιώντας την υπηρεσία ηλεκτρονικών περιοδικών πρέπει να γνωρίζετε ότι:

- αποδέχεται τις διατάξεις της ισχύουσας νομοθεσίας περί Πνευματικής Ιδιοκτησίας και Συγγενικών Δικαιωμάτων (Ν. 2121/93, Ν. 3049/2002 άρθρο 14 και Ν. 3057/2002 άρθρο 91) και ότι απαγορεύεται η συστηματική αποθήκευση ή εκτύπωση όλου του περιεχομένου των τευχών των περιοδικών και των βιβλίων
- δηλώνετε υπεύθυνα ότι τα τεκμήρια θα χρησιμοποιηθούν αποκλειστικά για προσωπική σας μελέτη ή έρευνα

**Νέα**

- Οι εκδότες SpringerLink και Wiley διαθέτουν τα backfiles των περιοδικών τους με διαφορετικό τρόπο από ότι οι άλλοι εκδότες. Συγκεκριμένα διαθέτουν δύο ξεχωριστές ιστοσελίδες για τα contents του κάθε περιοδικού, μία για τα current contents και μία για τα backfiles.

Επομένως:

- για τα backfiles του SpringerLink θα πρέπει να εντοπίσετε στον αλφαβητικό κατάλογο του εκδότη τις δύο εγγραφές του περιοδικού που σας ενδιαφέρει (μία με τον τίτλο του περιοδικού και μία με τον τίτλο-σημείωση Historical Archive) και από εκεί να κάνετε click στη δεύτερη εγγραφή.
- για τα backfiles του Wiley, αφού ανοίξετε, από τον αλφαβητικό κατάλογο της HEAL-Link, την ιστοσελίδα του περιοδικού που σας ενδιαφέρει θα πρέπει να κάνετε click στο σύνδεσμο που υπάρχει κάτω από τον τίτλο του περιοδικού με την σημείωση "See Also".

• PROQUEST 5000	02-10-2005
• Ενθάρρυνση για νέες συμφωνίες που έκανε η HEAL-Link	01-10-2005
• Υποψηφιότητα για το πρώτο SLA Europa Information Professional Award (SLA EIP 2005)	29-06-2005

Λίστα όλων των Νέων

Παλιό σελίδα HEAL-Link

**Κατασκευή και Διαχείριση της πύλης:**  
Φώλου Πόπη  
Πισπιρίγγας Λεωνίδας  
Χατζησταύρου Βλάδης

ΕΛΛΑΣ 2006  
ΠΑΙΔΕΙΑ ΜΠΡΟΣΤΑ  
2<sup>η</sup> Επιστημονικά Βελτιωμένη  
Εκδοτέρας και Αρχικής  
Επικοινωνιακής Κατάρτισης

Εικόνα 4.1 Αρχική Οθόνη του HEAL-Link v3.0

Αριστερά της αρχικής οθόνης, υπάρχουν επιλογές όπως σύνδεση με την υπηρεσία εξατομίκευσης της πύλης, δημιουργία νέου χρήστη, υπενθύμιση κωδικού

πρόσβασης, μηχανή αναζήτησης, διάφοροι τρόποι πλοήγησης στα ηλεκτρονικά περιοδικά του HEAL-Link.

#### **4.1.2 Πλοήγηση**

Στα αριστερά της κεντρικής οθόνης και στον δεύτερο πίνακα, δίνεται η δυνατότητα στον χρήστη να χρησιμοποιήσει 2 από τους 4 εναλλακτικούς τρόπους πλοήγησης στην πύλη.


Ο πρώτος τρόπος είναι η πλοήγηση με βάση τον θεματικό κατάλογο της πύλης, όπου ο χρήστης επιλέγει την κατηγορία που τον ενδιαφέρει. Τα επίπεδα για την εμφάνιση των περιοδικών από αυτή την μέθοδο πλοήγησης είναι τρία. Αρχικά με την επιλογή της κατηγορίας, εμφανίζονται οι υποκατηγορίες και επιλέγοντας μία υποκατηγορία εμφανίζονται οι αντίστοιχοι θεματικοί όροι γι' αυτή την υποκατηγορία. Τέλος επιλέγοντας κάποιον από τους θεματικούς όρους, εμφανίζονται τα ηλεκτρονικά περιοδικά που ανήκουν στον εν λόγω θεματικό όρο. Οι εικόνες 4.2 – 4.5 δείχνουν κάθε στάδιο αυτής της διαδικασίας.

Σε κάθε στάδιο της πλοήγησης, ο χρήστης θα έχει άμεση πρόσβαση στις υπόλοιπες υπηρεσίες της πύλης, ώστε να μπορεί ανά πάσα στιγμή να τις χρησιμοποιήσει, καθώς και να επιστρέψει στην αρχική σελίδα της πύλης.

Στο τελικό επίπεδο αυτής της διαδικασίας, ο χρήστης επιλέγοντας το περιοδικό που τον ενδιαφέρει, μεταφέρεται στον αντίστοιχο δικτυακό τόπο προκειμένου να επισκεφθεί την επιλογή του.

65

HELLENIC ACADEMIC  
**HEAL** LIBRARIES LINK  
ΔΙΚΤΥΟ ΕΛΛΗΝΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ

Αρχική
Πληροφορίες
Για Βιβλιοθηκονόμους
SELL

Αναζήτηση

- Γρήγορη Αναζήτηση

Αναζήτηση

- Σύνθετη Αναζήτηση

Υπηρεσίες

- Αλφαριθμητικός Κατάλογος
- Θεματικές Κατηγορίες
- Βιβλιογραφικές Βάσεις
- Ηλεκτρονικά Βιβλία
- Εκδότες

My HEAL-Link

E-mail:

Κωδικός:

Είσοδος

- Νέος χρήστης
- Υπενθύμιση κωδικού
- Οδηγίες

### Θεματικές Κατηγορίες

- Agriculture*
- Auxiliary sciences of history*
- Bibliography. Library science. Information resources*
- Education*
- Fine arts*
- General*
- Geography. Maps. Anthropology. Recreation*
- History*
- History: America (Western Hemisphere)*
- Language and literature*
- Law*
- Medicine*
- Military science*
- Music and books on music*
- Naval science*
- Philosophy. Psychology. Religion*
- Political science*
- Science*
- Social sciences*
- Technology*



**Κατασκευή και Διαχείριση της πύλης:**  
Φλώρου Πόπη  
Παπαγιώργας Αλεξάνδρος  
Χατζησταύρου Βλάσης





Adobe Acrobat - ...
Κεφάλαιο 4o.do...
Περιγραφή.doc ...
Δίκτυο Ελλήνι...
Mailbox.gr - Moz...
[Web2SMS] - M...
subject.JPG - Z...
Wind

Εικόνα 4.2 Περιεχόμενο κατηγοριών της Πύλης

The screenshot shows the HEAL-Link website interface. At the top, the URL is [http://www.heal-link.gr/journals/subcat.jsp?cat\\_id=4&cname=Education](http://www.heal-link.gr/journals/subcat.jsp?cat_id=4&cname=Education). The page header features the HEAL-Link logo and navigation links: Αρχική, Πληροφορίες, Για Βιβλιοθηκονόμους, and SELL. The main content area is divided into several sections:

- Αναζήτηση (Search):** Includes a search box with a "Αναζήτηση" button and links for "Γρήγορη Αναζήτηση" and "Σύνθετη Αναζήτηση".
- Υπηρεσίες (Services):** Lists services such as "Αλφαβητικός Κατάλογος", "Θεματικές Κατηγορίες", "Βιβλιογραφικές Βάσεις", "Ηλεκτρονικά Βιβλία", and "Εκδόσεις".
- My HEAL-Link:** Contains a login form with fields for "E-mail:" and "Κωδικός:", a "Είσοδος" button, and links for "Νέος χρήστης", "Υπενθάρμηση κωδικού", and "Οδηγίες".
- Θεματικές Κατηγορίες (Thematic Categories):** The main focus, listing subcategories under "Education":
  - Education (General)
  - History of education
  - Special aspects of education
  - Student fraternities and societies, United States
  - Textbooks
  - Theory and practice of education

At the bottom, there is a footer with contact information for "Κατασκευή και Διαχείριση της πύλης:" (Flouri P. Pappa, Pitsirigias Leonidas, Chatzistavrou Vlassis) and logos for the Ministry of Education and the 2008 Athens Olympics.

Εικόνα 4.3 Περιεχόμενο των θεματικών υποκατηγοριών της κατηγορίας Education

The screenshot shows the HEAL-Link website interface. At the top, there is a navigation bar with links for 'Αρχική', 'Πληροφορίες', 'Για Βιβλιοθηκονόμους', and 'SELL'. The main content area is divided into three columns. The left column contains three sections: 'Αναζήτηση' (Search) with a search box and 'Αναζήτηση' button, 'Υπηρεσίες' (Services) with a list of services, and 'My HEAL-Link' with an email and password field and an 'Είσοδος' (Login) button. The middle column is titled 'Θεματικές Κατηγορίες' (Thematic Categories) and lists various categories under the heading 'Θεματικοί όροι της υποκατηγορίας History of education'. The right column contains logos for the Ministry of Education and Religious Affairs and the European Union.

**HEAL-Link LIBRARIES LINK**  
ΔΙΚΤΥΟ ΕΛΛΗΝΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ

**Αναζήτηση**

- Γρήγορη Αναζήτηση
- Σύνθετη Αναζήτηση

**Υπηρεσίες**

- Αλφαβητικός Κατάλογος
- Θεματικές Κατηγορίες
- Βιβλιογραφικές Βάσεις
- Ηλεκτρονικά Βιβλία
- Εκδότες

**My HEAL-Link**

E-mail:

Κωδικός:

**Είσοδος**

- Νέος χρήστης
- Υπενθέρμηση κωδικού
- Οδηγίες

**Θεματικές Κατηγορίες**

Θεματικοί όροι της υποκατηγορίας **History of education**

- College students
- Counseling in higher education -- United States
- Education -- Brazil
- Education -- China
- Education -- Europe
- Education -- History
- Education -- Mexico
- Education -- Social aspects -- China
- Education -- United States
- Education, Higher
- Education, Higher -- Europe
- Education, Higher -- United States
- Education, Secondary
- Education, Secondary -- United States
- Educational change
- High schools
- Learning
- Teaching
- Universities and colleges -- United States

**Κατασκευή και Διαχείριση της πύλης:**  
Φλώρου Πόπη  
Πιστηρίγγος Λεωνίδας  
Χατζησταύρου Βλίσσης

**ΕΛΛΑΣ 2008**  
**ΠΑΙΔΕΙΑ ΜΠΡΟΣΤΑ**  
2<sup>ο</sup> Επιχειρησιακό Πρόγραμμα  
Εκπαίδευσης και Αρχικής  
Επαγγελματικής Κατάρτισης

Adobe Acrobat - ... | Κεφάλαιο 4o.do... | Περιγραφή.doc ... | Δίκτυο Ελληνι... | Mailbox.gr - Moz... | [Web25MS] - M... | subj.JPG - Ζωγρ... | Windo

Εικόνα 4.4 Θεματικοί όροι της υποκατηγορίας History of Education

The screenshot shows the HEAL-Link website interface. At the top, there is a navigation bar with links for 'Αρχική', 'Πληροφορίες', 'Για Βιβλιοθηκονόμους', and 'SELL'. The main content area is titled 'Θεματικές Κατηγορίες' (Thematic Categories) and is under the subcategory 'History of education'. It lists two periodicals under the 'High schools' category:

- High School Magazine (Name changed to Principal Leadership (High School Ed.))** by Wilson, published from 1998 to 2000.
- Principal Leadership (High School Ed.) (formerly High School Magazine)** by Wilson, published from 2000 onwards.

On the left side, there are three utility boxes: 'Αναζήτηση' (Search) with a search bar and 'Αναζήτηση' button; 'Υπηρεσίες' (Services) with links to 'Αλφαβητικός Κατάλογος', 'Θεματικές Κατηγορίες', 'Βιβλιογραφικές Βάσεις', 'Ηλεκτρονικά Βιβλία', and 'Εκδότες'; and 'My HEAL-Link' with an email registration form and 'Είσοδος' button.

At the bottom, there is a footer with the text 'Κατασκευή και Διαχείριση της πύλης: Φλόρου Πόπη, Πιστηρίγγας Λεωνίδας, Χατζησταύρου Βλίσσης' and logos for 'ΕΛΛΑΣ 2006' and 'ΠΑΙΔΕΙΑ ΜΠΡΟΣΤΑ'.

Εικόνα 4.5 Περιοδικά του θεματικού όρου *High Schools*

Ο δεύτερος τρόπος πλοήγησης διατίθεται από την υπηρεσία Αλφαβητικού Καταλόγου, από όπου ο χρήστης επιλέγει ανάμεσα στα γράμματα του αλφαβήτου και του εμφανίζεται μία λίστα με τα αντίστοιχα περιοδικά. Επιλέγοντας ένα από αυτά τα περιοδικά, μεταφέρεται στον δικτυακό τόπο του περιοδικού, όπου και μπορεί να διαβάσει ή να μεταφορτώσει το ηλεκτρονικό περιοδικό του. Οι εικόνες 4.6 και 4.7 μας παρουσιάζουν στιγμιότυπα αυτής της υπηρεσίας.

The screenshot shows a web browser window with the URL <http://www.heal-link.gr/journals/alphasearch.jsp>. The page header includes the HEAL-Link logo (HELLENIC ACADEMIC LIBRARIES LINK) and the text ΔΙΚΤΥΟ ΕΛΛΗΝΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ. A navigation bar contains links for Αρχική, Πληροφορίες, Για Βιβλιοθηκονόμους, and SELL. The main content area is titled "Αλφαβητικός Κατάλογος" and displays the alphabet "0-9 ABCDEFGHIJKLMNOPQRSTUVWXYZ". On the left, there are three sidebar sections: "Αναζήτηση" with search options for "Γρήγορη Αναζήτηση" and "Σύνθετη Αναζήτηση"; "Υπηρεσίες" listing "Αλφαβητικός Κατάλογος", "Θεματικές Κατηγορίες", "Βιβλιογραφικές Βάσεις", "Ηλεκτρονικά Βιβλία", and "Εκδότες"; and "My HEAL-Link" with an email and password login form and options for "Νέος χρήστης", "Υπενθάρμηση κωδικού", and "Οδηγίες". At the bottom, there is a footer with contact information for the website's construction and management, and logos for the Ministry of Education and Religious Affairs and the National Library of Greece.

Εικόνα 4.6 Αλφαβητικός κατάλογος

The screenshot shows the HEAL-Link website interface. At the top, there is a navigation bar with links for 'Αρχική', 'Πληροφορίες', 'Για Βιβλιοθηκονόμους', and 'SELL'. The main content area is titled 'Θεματικές Κατηγορίες' (Thematic Categories) and lists two categories under the subcategory 'History of education':

- High School Magazine (Name changed to Principal Leadership (High School Ed.))** by Wilson, covering the years 1998 - 2000.
- Principal Leadership (High School Ed.) (formerly High School Magazine)** by Wilson, covering the years 2000 - .

On the left side, there are three main sections: 'Αναζήτηση' (Search) with a search box and 'Αναζήτηση' button; 'Υπηρεσίες' (Services) listing 'Αλφαβητικός Κατάλογος', 'Θεματικές Κατηγορίες', 'Βιβλιογραφικές Βάσεις', 'Ηλεκτρονικά Βιβλία', and 'Εκδότες'; and 'My HEAL-Link' with an email registration form and 'Είσοδος' button.

At the bottom, there is a footer with the text 'Κατασκευή και Διαχείριση της πύλης: Φλώρου Πόπη, Παπαρίγγας Λεωνίδας, Χατζησταύρου Βλάσης' and logos for 'ΕΛΛΑΣ 2008' and 'ΠΑΙΔΕΙΑ ΜΠΡΟΣΤΑ'.

Εικόνα 4.7 Αποτελέσματα περιοδικών από την χρησιμοποίηση του Αλφαβητικού καταλόγου



### 4.1.3 Υπηρεσίες Αναζήτησης

Στην αρχική οθόνη, υπάρχει η δυνατότητα αναζήτησης περιοδικών στην πύλη χρησιμοποιώντας μία από τις δύο μηχανές αναζήτησης που προσφέρει το HEAL-Link. Η πρόσβαση στην γρήγορη αναζήτηση είναι άμεση καθώς υπάρχει η ανάλογη φόρμα στον πρώτο πίνακα, αριστερά της οθόνης. Στη φόρμα αυτή, ο χρήστης συμπληρώνει τις λέξεις κλειδιά που επιθυμεί και επιλέγοντας αναζήτηση ξεκινάει η διαδικασία εύρεσης κατηγοριών, υποκατηγοριών, θεματικών ορών και τίτλων ηλεκτρονικών περιοδικών που αντιστοιχούν στις λέξεις κλειδιά που έδωσε ο χρήστης. Η αναζήτηση γίνεται με βάση την επιλογή LIKE που προσφέρει η sql και το ερώτημα απευθύνεται στους ανάλογους πίνακες, βάση των οποίων κατηγοριοποιούμε τα αποτελέσματα.

Εδώ ο χρήστης μπορεί είτε να επιλέξει άμεσα τον τίτλο περιοδικού που τον ενδιαφέρει και εφόσον βέβαια ικανοποιεί τις προσδοκίες του ή να συνεχίσει με την πλοήγηση του στις κατηγορίες, υποκατηγορίες ή θεματικούς όρους που εμφάνισε η μηχανή αναζήτησης.

Κάτω από την μηχανή γρήγορης αναζήτησης, υπάρχει ένας σύνδεσμος που οδηγεί στην μηχανή αναζήτησης που ενσωματώνεται σε όλες τις εκδόσεις τους HEAL-Link. Ο τίτλος του συνδέσμου είναι “Σύνθετη αναζήτηση”, ενώ η φόρμα της μηχανής αυτής φαίνεται στην εικόνα 4.8.

Ο χρήστης μπορεί να αναζητήσει ένα ή περισσότερα άρθρα περιοδικών χρησιμοποιώντας μία ή δύο λέξεις-κλειδιά και ένα από τα λογικά συνεκτικά OR και AND. Επίσης, μπορεί να περιορίσει τον στόχο της αναζήτησης, καθορίζοντας τους εκδότες για τα περιοδικά των οποίων ενδιαφέρεται. Αυτό το τελευταίο επιτυγχάνεται επιλέγοντας το κυκλάκι 'Ναι' (για αποκλεισμό: το κυκλάκι 'Όχι') στα δεξιά του ονόματος του αντίστοιχου εκδότη

HELENIC ACADEMIC LIBRARIES LINK  
ΔΙΚΤΥΟ ΕΛΛΗΝΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ

Αρχική Πληροφορίες Για Βιβλιοθηκονόμους SELL

### Αναζήτηση

• Γρήγορη Αναζήτηση  
• Σύνθετη Αναζήτηση

**Υπηρεσίες**

- Αλφαβητικός Κατάλογος
- Θεματικές Κατηγορίες
- Βιβλιογραφικές Βάσεις
- Ηλεκτρονικά Βιβλία
- Εκδότες

**My HEAL-Link**

E-mail:   
Κωδικός:   
**Είσοδος**

- Νέος χρήστης
- Υπενθμίση κωδικού
- Οδηγίες

### Σύνθετη Αναζήτηση

Αναζήτηση άρθρων με λέξεις-κλειδιά στα ηλεκτρονικά περιοδικά των εκδοτών

**Προσοχή:** Πρέπει να θέσετε μία λέξη σε κάθε πεδίο ειδώλλως τα αποτελέσματά σας δεν θα είναι τα αναμενόμενα. Ίσως χρειαστεί να περιμένετε ώστε να απαντήσουν οι Μηχανές που χρησιμοποιεί το η Σύνθετη Αναζήτηση. Είναι πιθανό να χρειαστείτε 3 ή 4 λεπτά.

AND

Elsevier's ScienceDirect:	<input checked="" type="radio"/> ΝΑΙ <input type="radio"/> ΟΧΙ
Springer's Link:	<input checked="" type="radio"/> ΝΑΙ <input type="radio"/> ΟΧΙ
Wilson Web:	<input checked="" type="radio"/> ΝΑΙ <input type="radio"/> ΟΧΙ
Emerald:	<input checked="" type="radio"/> ΝΑΙ <input type="radio"/> ΟΧΙ
Kluwer:	<input checked="" type="radio"/> ΝΑΙ <input type="radio"/> ΟΧΙ
Institute of Physics:	<input checked="" type="radio"/> ΝΑΙ <input type="radio"/> ΟΧΙ
American Chemical Society:	<input checked="" type="radio"/> ΝΑΙ <input type="radio"/> ΟΧΙ
Oxford University Press:	<input checked="" type="radio"/> ΝΑΙ <input type="radio"/> ΟΧΙ
Wiley InterScience:	<input checked="" type="radio"/> ΝΑΙ <input type="radio"/> ΟΧΙ
ACM (Association of Computing Machinery):	<input checked="" type="radio"/> ΝΑΙ <input type="radio"/> ΟΧΙ
Blackwell Publishing:	<input checked="" type="radio"/> ΝΑΙ <input type="radio"/> ΟΧΙ
Taylor & Francis:	<input checked="" type="radio"/> ΝΑΙ <input type="radio"/> ΟΧΙ
Lippincott, Williams & Wilkins:	<input checked="" type="radio"/> ΝΑΙ <input type="radio"/> ΟΧΙ

**Αναζήτηση**

Κατασκευή και Διαχείριση της πύλης:  
Φλώρα Πόπη  
Πισπιρίγγας Λεωνίδας  
Χατζησταύρου Βλάσης

ΕΛΛΑΣ 2008  
ΠΑΙΔΕΙΑ ΠΡΩΤΑ  
2<sup>ο</sup> Πανεπιστημιακό Πεδίο  
Εκπαίδευσης και Κρατικής  
Επιχειρησιακής Κατάνυξης

τση ... Adobe Acrobat ... Κεφάλαιο 4ο.do... Περιγραφή.doc ... Δίκτυο Ελληνι... [Web2SMS] - M... alpha.JPG - Ζωγ... Windows Messe... Apache Tomcat/...

Εικόνα 4.8 Σύνθετη αναζήτηση της Πύλης

Η σύνθετη μηχανή αναζήτησης, που στην ουσία είναι μια μηχανή μετά-αναζήτησης, αναζητεί τις λέξεις κλειδιά που εισήγαγε ο χρήστης στους αντίστοιχους εκδοτικούς οίκους που έχει επιλέξει, εμφανίζοντας τα αντίστοιχα αποτελέσματα στην οθόνη.

#### 4.1.4 Νέα του HEAL-Link

Στην αρχική οθόνη της πύλης και κάτω υπάρχουν οι τρεις πιο πρόσφατοι τίτλοι νέων της πύλης. Ο χρήστης επιλέγοντας έναν από αυτούς μπορεί να διαβάσει ολόκληρο το περιεχόμενο του τίτλου. Επίσης κάτω από τα νέα υπάρχει ένας σύνδεσμος με τίτλο “ Λίστα όλων των νέων “ που εμφανίζει τη λίστα όλων των τίτλων νέων της πύλης και από την οποία ο χρήστης έχει την δυνατότητα να επισκεφθεί οποιονδήποτε τίτλο επιθυμεί.

#### 4.1.5 MyHEAL-Link

Στον τρίτο πίνακα αριστερά της οθόνης, ο χρήστης μπορεί να χρησιμοποιήσει την υπηρεσία MyHEAL-Link της πύλης, που είναι ένα υπο-περιβάλλον εξατομίκευσης. Για να έχει πρόσβαση σε αυτή την υπηρεσία θα πρέπει πρώτα να δημιουργήσει ένα λογαριασμό. Η διαδικασία αυτή ενεργοποιείται με την επιλογή του συνδέσμου “ Νέος Χρήστης “ που βρίσκεται στον ίδιο πίνακα.

Για να ολοκληρωθεί η εγγραφή θα πρέπει ο χρήστης να συμπληρώσει τα στοιχεία του στην φόρμα που του εμφανίζεται και φαίνεται στην εικόνα 4.9.

The screenshot shows the HEAL-Link website interface. At the top, there is a navigation bar with the following links: Αρχική, Πληροφορίες, Για Βιβλιοθηκονόμους, and SELL. The main content area is divided into three columns. The left column contains three sections: 'Αναζήτηση' (Search) with options for 'Γρήγορη Αναζήτηση' and 'Σύνθετη Αναζήτηση'; 'Υπηρεσίες' (Services) with a list of links including 'Αλφαβητικός Κατάλογος', 'Θεματικές Κατηγορίες', 'Βιβλιογραφικές Βάσεις', 'Ηλεκτρονικά Βιβλία', and 'Εκδότες'; and 'My HEAL-Link' with an 'Είσοδος' (Login) button and links for 'Νέος χρήστης', 'Υπενθύμιση κωδικού', and 'Οδηγίες'. The middle column is titled 'Νέος Χρήστης' (New User) and contains a registration form with the following fields: 'Όνομα:', 'Επώνυμο:', 'Ίδρυμα:' (with a dropdown menu set to 'ΤΕΙ ΘΕΣΣΑΛΟΝΙΚΗΣ'), 'Ιδιότητα:' (with a dropdown menu set to 'Φοιτητής'), and 'E-mail:'. A 'Συνέχεια' (Continue) button is located below the form. The right column is empty. At the bottom of the page, there are logos for HEAL, the Ministry of Education and Religious Affairs, and the National Library of Greece.

Εικόνα 4.9 Φόρμα δημιουργίας νέου λογαριασμού

Για να γίνει δεκτή η εγγραφή του χρήστη θα πρέπει να ανήκει σε ένα από τα ιδρύματα που υποστηρίζει η πύλη. Αυτό συνεπάγεται την κατοχή, από την πλευρά

του χρήστη, μιας διεύθυνσης ηλεκτρονικού ταχυδρομείου που να είναι αποδεκτή από την πύλη. Η διαδικασία ολοκληρώνεται με την αποστολή του συνθηματικού στην ηλεκτρονική διεύθυνση που έχει δηλώσει ο χρήστης. Η ηλεκτρονική διεύθυνση του χρήστη αποτελεί και το όνομα χρήστη για το MyHEAL-Link.

Σε περίπτωση που ο χρήστης ξεχάσει ή χάσει το συνθηματικό του, υπάρχει υπηρεσία υπενθύμισης. Ο χρήστης επιλέγει τον σύνδεσμο υπενθύμιση κωδικού που υπάρχει στον ίδιο πίνακα και του εμφανίζεται μία φόρμα που του ζητάει να συμπληρώσει το email με το οποίο πραγματοποίησε την εγγραφή του στην πύλη. Σε αυτό το email θα του αποσταλεί το συνθηματικό του.

The screenshot shows the HEAL-Link website interface. At the top, there is a navigation bar with the following links: Αρχική, Πληροφορίες, Για Βιβλιοθηκονόμους, and SELL. The main content area is divided into three columns. The left column contains three sections: 'Αναζήτηση' (Search) with links for 'Γρήγορη Αναζήτηση' and 'Σύνθετη Αναζήτηση', 'Υπηρεσίες' (Services) with links for 'Αλφαριθμητικός Κατάλογος', 'Θεματικές Κατηγορίες', 'Βιβλιογραφικές Βάσεις', 'Ηλεκτρονικά Βιβλία', and 'Εκδότες', and 'My HEAL-Link' with an 'Είσοδος' (Login) button and links for 'Νέος χρήστης', 'Υπενθύμιση κωδικού', and 'Οδηγίες'. The middle column is titled 'Υπενθύμιση Κωδικού' (Password Recovery) and contains a form with a text input field and an 'Αποστολή' (Send) button. The right column is empty. At the bottom, there are logos for the Ministry of Education and Religious Affairs, the National Library of Greece, and the National Archives and Library Organization of Greece.

Εικόνα 4.10 Φόρμα για υπενθύμιση κωδικού πρόσβασης

## 4.2 Εξατομίκευση Πύλης

### 4.2.1 Είσοδος

Η είσοδος στο υπο-περιβάλλον εξατομίκευσης του HEAL-Link (MyHEAL-Link) πραγματοποιείται από την αρχική σελίδα της πύλης και από τον τρίτο πίνακα που βρίσκεται στα αριστερά της οθόνης. Ο χρήστης καλείται να εισάγει το όνομα χρήστη, που είναι η ηλεκτρονική διεύθυνση που χρησιμοποίησε για την δημιουργία του λογαριασμού του, και το συνθηματικό που του έχει αποσταλεί από την πύλη. Εάν είναι η πρώτη φορά που επισκέπτεται την πύλη ο χρήστης, τότε το προφίλ του είναι άδειο. Ο χρήστης έχει τη δυνατότητα να διαμορφώσει αυτό το προφίλ ανάλογα με τα ενδιαφέροντα του. Η οθόνη που βλέπει ο χρήστης για πρώτη φορά φαίνεται στην εικόνα 4.11

The screenshot shows the MyHEAL-Link user interface. At the top, there is a navigation bar with the following tabs: Αρχική, Προσωπικά Στοιχεία, and Έξοδος. The main content area is divided into several sections:

- Καλώς ήρθατε**: A welcome message for the user 'Υπάτιος', dated 'Τρίτη 22 Νοεμβρίου', with 'Συνολικές Συνδέσεις: 0' and a link for 'Λεπτομέρειες'. A message states 'Τα Περιοδικά μου δεν βρέθηκαν.'
- Αναζήτηση**: A search section with a search box and a 'Αναζήτηση' button. It includes links for 'Γρήγορη Αναζήτηση' and 'Σύνθετη Αναζήτηση'.
- Προφίλ**: A profile section with links for 'Αλφαβητικός Κατάλογος', 'Θεματικές Κατηγορίες', and 'Αυτόματη Ειδοποίηση'.

At the bottom of the page, there is a footer with the following information:

- Κατασκευή και Διαχείριση της πύλης:** Φλώρου Πόπη, Πισπρίγγας Λεωνίδας, Χατζησταύρου Βλάσης.
- ΕΛΛΑΣ 2006** logo.
- ΠΑΙΔΕΙΑ ΜΠΡΟΣΤΑ** logo, with the text: 2<sup>ο</sup> Επικεντρωτικό Πρόγραμμα Εκπαίδευσης και Αρχικής Επαγγελματικής Κατάρτισης.

4.11 Αρχική οθόνη του MyHEAL-Link

#### 4.2.2 Επεξεργασία προφίλ

Όπως φαίνεται στην αρχική οθόνη του MyHEAL-Link υπάρχουν τρεις διαφορετικοί τρόποι για να διαμορφωθεί το προφίλ του χρήστη. Υπάρχει η διαμόρφωση με βάση τον αλφαβητικό κατάλογο, με βάση τις θεματικές κατηγορίες και η αυτόματη ειδοποίηση, καθώς και η ενημέρωση από την μηχανή γρήγορης αναζήτησης.

Στην διαμόρφωση με βάση τον αλφαβητικό κατάλογο ο χρήστης επιλέγει το γράμμα που τον ενδιαφέρει για να αναζητήσει το ηλεκτρονικό περιοδικό που θέλει να εισάγει στο προφίλ του. Από τη λίστα με τα περιοδικά που εμφανίζονται για το αντίστοιχο γράμμα, ο χρήστης μπορεί να επιλέξει τα περιοδικά που τον ενδιαφέρουν και να τα προσθέσει στο προφίλ του.

Στην διαμόρφωση με βάση τον θεματικό κατάλογο, ο χρήστης μπορεί να πλοηγηθεί ανάμεσα στις κατηγορίες, υποκατηγορίες και θεματικούς όρους, με στόχο να εντοπίσει το περιοδικό που θέλει να εισάγει στο προφίλ του.

Με τη βοήθεια της μηχανής αναζήτησης ο χρήστης εισάγει τις λέξεις κλειδιά που τον ενδιαφέρουν και επιλέγει την αναζήτηση. Τα αποτελέσματα παρουσιάζονται κατηγοριοποιημένα και ο χρήστης μπορεί είτε να πλοηγηθεί στις κατηγορίες, υποκατηγορίες ή θεματικούς όρους που εμφανίστηκαν, είτε να προσθέσει στο προφίλ του κάποιο ή κάποια από τα περιοδικά που εμφανίστηκαν όπως φαίνεται στην εικόνα 4.12.

Αν ο χρήστης έχει ενημερώσει το προφίλ του, τότε η αρχική οθόνη που βλέπει θα περιλαμβάνει και τα αντίστοιχα περιοδικά που τον ενδιαφέρουν.

The screenshot shows the HEAL-Link user interface. At the top, there is a navigation bar with 'Αρχική', 'Προσωπικά Στοιχεία', and 'Έξοδος'. The main content area is titled 'Γρήγορη Αναζήτηση' and 'Αποτελέσματα'. On the left, there are three sidebar panels: 'Καλώς ήρθατε' (Welcome) showing the user's name 'Υπάτιος' and login date 'Τρίτη 22 Νοεμβρίου', 'Αναζήτηση' (Search) with a search box and 'Αναζήτηση' button, and 'Προφίλ' (Profile) with links to 'Αλφαβητικός Κατάλογος', 'Θεματικές Κατηγορίες', and 'Αυτόματη Ειδοποίηση'. The main content area lists categories: 'Κατηγορίες' (Agriculture), 'Υποκατηγορίες' (Agriculture (General), Animal culture, Aquaculture, Fisheries, Angling, Plant culture), 'Θεματικοί όροι' (Agriculture, Agriculture -- Experimentation, Culture), and 'Περιοδικά' (Experimental Agriculture, Victorian Literature and Culture). A 'Καταχώρηση' button is at the bottom.

Εικόνα 4.12 Γρήγορη αναζήτηση στο προφίλ του χρήστη

#### 4.2.3 Ενημέρωση για νέα περιοδικά

Η υπηρεσία αυτή δίνει την δυνατότητα στον χρήστη να ενημερώνεται για αφίσξεις νέων περιοδικών με βάση τον θεματικό όρο που έχει επιλέξει ο χρήστης. Τα περιοδικά θεωρούνται ως νέα για 2 μήνες μετά την εισαγωγή τους στη βάση του HEAL-Link.

Όταν ο χρήστης πραγματοποιεί εισαγωγή στο σύστημα, τότε αν υπάρχουν νέα περιοδικά για τον θεματικό όρο που έχει επιλέξει να ενημερώνεται, τότε του εμφανίζεται ένα παράθυρο με τους τίτλους αυτών των περιοδικών και την δυνατότητα επιλογής για να τα εισάγει ή όχι στο προφίλ του. Η επιλογής είναι της μορφής ΝΑΙ ή ΟΧΙ οπότε ο χρήστης επιλέγει το ανάλογο κυκλάκι δίπλα στον τίτλο

του περιοδικού. Εάν ο χρήστης κλείσει το παράθυρο χωρίς να επιλέξει τότε το παράθυρο θα εμφανίζεται κάθε φορά που ο χρήστης συνδέεται με το υπο-περιβάλλον του, ωστόσο τα περιοδικά θεωρηθούν παλαιά, δηλαδή μετά από δύο μήνες.

#### 4.2.4 Αλλαγή προσωπικών στοιχείων

Επιλέγοντας αυτή την υπηρεσία από τον σύνδεσμο που βρίσκεται στην κορυφή της οθόνης με όνομα “ Προσωπικά Στοιχεία “ ο χρήστης έχει την δυνατότητα να αλλάξει τα στοιχεία του όπως όνομα, επώνυμο, ιδιότητα, ίδρυμα όπως φαίνεται στην παρακάτω εικόνα.

The screenshot shows the HEAL-Link website interface. At the top, there is a navigation bar with the following menu items: Αρχική, Προσωπικά Στοιχεία (selected), Έξοδος. The main content area is titled 'Προσωπικά Στοιχεία' and contains a form with the following fields:

- Όνομα:
- Επώνυμο:
- E-mail:
- Ιδιότητα:
- Ίδρυμα:

There is a 'Συνέχεια' button at the bottom right of the form. On the left side, there are three main sections: 'Καλώς ήρθατε' (Welcome) with the user's name and login date, 'Αναζήτηση' (Search) with options for 'Γρήγορη Αναζήτηση' and 'Σύνθετη Αναζήτηση', and 'Προφίλ' (Profile) with links to 'Αλφabetικός Κατάλογος', 'Θεματικές Κατηγορίες', and 'Αυτόματη Ειδοποίηση'.

At the bottom of the page, there are logos for HEAL-Link, the Ministry of Education and Religious Affairs, and the National Center for Academic Libraries (ΕΛΛΑΔΑ 2006).

Εικόνα 4.13 Αλλαγή προσωπικών στοιχείων



#### **4.2.5 Λεπτομέρειες συνδέσεων**

Στο πρώτο πίνακα αριστερά της οθόνης φαίνονται οι συνολικές συνδέσεις που πραγματοποίησε ο χρήστης στο υπό-περιβάλλον του. Ακριβώς από κάτω υπάρχει ένας σύνδεσμος με όνομα “ Λεπτομέρειες “ που εμφανίζει λεπτομέρειες των συνδέσεων του χρήστη. Αυτά που εμφανίζει είναι η διεύθυνση ip από την οποία πραγματοποίησε τη σύνδεση ο χρήστης και την χρονική στιγμή που έγινε η σύνδεση.

## ΚΕΦΑΛΑΙΟ 5<sup>ο</sup> : ΠΡΟΤΑΣΕΙΣ

### 5.1 Τεχνολογία

Στο παρόν σύστημα, ο διακομιστής εφαρμογών και η βάση δεδομένων φιλοξενούνται σε διαφορετικά μηχανήματα που και τα δύο χρησιμοποιούν σαν λειτουργικό σύστημα την οικογένεια λειτουργικών συστημάτων Microsoft Windows. Η σταθερότητα του λειτουργικού συστήματος δεν είναι και η καλύτερη δυνατή σύμφωνα με πολλές έρευνες και συμπεράσματα στην πάροδο του χρόνου. Επίσης η απόδοση του λειτουργικού συστήματος για τέτοιου είδους υπηρεσίες δεν βρίσκεται σε πολύ υψηλά επίπεδα.

Οι μεγαλύτερες εταιρίες χρησιμοποιούν λειτουργικά συστήματα όπως Linux και Solaris για να πετύχουν σταθερότητα και καλύτερη απόδοση σε τέτοιου είδους υπηρεσίες. Οι περισσότεροι διακομιστές χρησιμοποιούν ένα από τα δύο λειτουργικά με τα Windows να έχουν το μικρότερο κομμάτι συμμετοχής.

Τόσο η βάση δεδομένων DB2 της IBM, όσο και ο διακομιστής εφαρμογών Jakarta-tomcat μπορούν να φιλοξενηθούν σε λειτουργικό Linux και θα είχαν καλύτερη συμπεριφορά σε στιγμές υπερφόρτωσης της εφαρμογής. Το μειονέκτημα εντοπίζεται στην δυσκολία εγκατάστασης της βάσης δεδομένων IBM DB2 που προσφέρεται με τη μορφή rpm πακέτων. Την υποστήριξη αυτών των πακέτων δεν την υποστηρίζουν όλες οι διανομές Linux όπως Slackware, ενώ εργαλεία που μετατρέπουν τέτοια πακέτα στα αντίστοιχα της διανομής π.χ. rpm2tgz, δεν εγγυώνται πάντα τα επιθυμητά αποτελέσματα. Έτσι οι διανομές που συστήνουμε για την φιλοξενία αυτού του RDBMS είναι οι RedHat, SuSE και Mandrake. Από την άλλη ο Jakarta-tomcat δεν παρουσιάζει καμία δυσκολία εγκατάστασης σε καμία διανομή Linux.

Ένα ακόμα μειονέκτημα της μεταφοράς της εφαρμογής σε περιβάλλον Linux είναι η απαίτηση ιδιαίτερης τεχνογνωσίας γύρω από το λειτουργικό. Η μη φιλική δομή του λειτουργικού – αν και οι τελευταίες διανομές εστιάζουν την προσοχή τους

στην φιλικότητα ως προς τον χρήστη – απαιτεί ιδιαίτερη γνώση και διαχείριση του συστήματος προκειμένου να παραμείνει υγιές.

## **5.2 Εφαρμογή και υπηρεσίες**

### **5.2.1 Αλλαγή Συνθηματικού**

Ανάμεσα στις προτάσεις για βελτίωση και εμπλουτισμό της εφαρμογής και των υπηρεσιών της βρίσκεται και δυνατότητα αλλαγής συνθηματικού. Ο χρήστης – μέλος της πύλης να έχει την δυνατότητα να αλλάξει το συνθηματικό που χρησιμοποιεί για την είσοδό του στο υπό – περιβάλλον του. Το συνθηματικό που παράγεται από έναν αλγόριθμο κατά την διαδικασία εγγραφής, αποτελείται από γράμματα και αριθμούς και είναι δύσκολος στην απομνημόνευση. Η δυνατότητα αλλαγής σε κάποιο συνθηματικό που θα μπορεί να θυμάται πιο εύκολα ο χρήστης θα μπορούσε να φανεί χρήσιμη υπηρεσία.

### **5.2.2 Forum**

Η δημιουργία ενός forum όπου θα μπορούσαν να υποβάλλονται απορίες και να ανοίγονται συζητήσεις γύρω από τα ηλεκτρονικά περιοδικά που προσφέρει η πύλη θα μπορούσε να φανεί χρήσιμη. Για παράδειγμα θα μπορούσε κάποιος να χρησιμοποιήσει αυτή την υπηρεσία προκειμένου να υποβάλει ερώτημα σχετικά με την αναζήτηση ενός θέματος που δεν μπόρεσε να εντοπίσει από τους τρόπους αναζήτησης που προσφέρει η πύλη. Να ανοίξει συζήτηση γύρω από ένα θέμα κάποιου περιοδικού όπου θα μπορούσαν να εκφέρουν την γνώμη τους, η οποία μπορεί να βοηθούσε κάποιους να καταλάβουν καλύτερα διάφορες έννοιες ή άρθρα.

Το μειονέκτημα αυτής της υπηρεσίας εστιάζεται στην δέσμευση προσωπικού για την παρακολούθηση του Forum, προκειμένου να εξασφαλίσει τη σωστή λειτουργία του.

### **5.2.3 Περισσότερες επιλογές διαμόρφωσης προφίλ**

Στην παρούσα εφαρμογή, όσα ηλεκτρονικά περιοδικά εισάγει ο χρήστης στο προφίλ του, αυτά εμφανίζονται στην αρχική οθόνη. Σε περίπτωση που ο χρήστης έχει επιλέξει πολλά περιοδικά στο προφίλ του, τότε υπάρχει δυσκολία εύρεσης του περιοδικού που θέλει να επισκεφθεί. Εδώ θα μπορούσε να υπάρχει μια υπηρεσία που θα έδινε την δυνατότητα στον χρήστη να κατηγοριοποιήσει τα περιοδικά του, δημιουργώντας φακέλους.

Ο χρήστης θα μπορεί να δημιουργεί φακέλους με το όνομα που αυτός επιθυμεί, μέσα στο προφίλ του. Στους φακέλους αυτούς θα του δίνεται η δυνατότητα να προσθέτει περιοδικά. Αυτό θα τον βοηθούσε στο να δημιουργήσει ένα είδος ευρετηρίου, ώστε να βρίσκει τα περιοδικά που τον ενδιαφέρουν πιο γρήγορα.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- ✓ **Συστήματα Διαχείρισης Βάσεων Δεδομένων ( 2<sup>η</sup> Έκδοση )** by Raghu Ramakrishnan – Johannes Gehrke  
**ISBN : 960-8050-85-5**
- ✓ **JSP The Complete Reference** by Phillip Hanna  
**ISBN : 007-2127-68-6**
- ✓ **Apache Tomcat Bible** by Narayan  
**ISBN : 076-4526-06-5**

Επίσης χρησιμοποιήθηκαν άρθρα από διάφορους δικτυακούς τόπους όπως :

- ✓ <http://www.java.sun.com>
- ✓ <http://tomcat.apache.org>
- ✓ <http://www.jspin.com>
- ✓ <http://www.oreilly.com/catalog/javabeans/>