

**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**  
**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ**  
**ΥΠΟΛΟΓΙΣΤΩΝ**  
**ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ**  
**ΥΠΟΛΟΓΙΣΤΩΝ**  
**ΖΩΓΡΑΦΟΥ 157 73, ΑΘΗΝΑ**



ΕΒΓΔ-ΔΙΠΛ-2001-05

01 Οκτωβρίου 2001

**ΗΛΕΚΤΡΟΝΙΚΗ ΑΙΤΗΣΗ ΕΓΓΡΑΦΗΣ ΦΟΙΤΗΤΗ**

**ΜΥΛΩΝΑΣ ΦΟΙΒΟΣ-ΑΠΟΣΤΟΛΟΣ**

**ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Τίμος Σελλής**



**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**  
**ΕΡΓΑΣΤΗΡΙΟ ΣΥΣΤΗΜΑΤΩΝ ΒΑΣΕΩΝ ΓΝΩΣΕΩΝ**  
**ΚΑΙ ΔΕΔΟΜΕΝΩΝ**

## Πρόλογος

Σκοπός του παρόντος κειμένου είναι η πλήρης περιγραφή και παρουσίαση της διπλωματικής μου εργασίας. Αντικείμενό της αποτέλεσε η εκ του μηδενός σχεδίαση, υλοποίηση και κατασκευή ενός υπολογιστικού συστήματος, μέσω του οποίου οι φοιτητές του Τμήματος Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου, θα έχουν τη δυνατότητα να εγγράφονται για την παρακολούθηση μαθημάτων στα ακαδημαϊκά εξάμηνα με χρήση του Internet.

Η διπλωματική αυτή εργασία εκπονήθηκε με την επίβλεψη του Εργαστηρίου Βάσεων Γνώσεων και Δεδομένων (dblab) του καθηγητή κ. Τ. Σελλή, ο οποίος αποτέλεσε και τον επιβλέποντα καθηγητή της. Στην ανάπτυξη του συστήματος μεγάλη βοήθεια και συμβουλές παρείχε ο υποψήφιος διδάκτορας Σ. Σκιαδόπουλος. Επίσης, θεωρείται και ήταν πολύτιμη η βοήθεια διαφόρων μελών του dblab. Θέλω ξεχωριστά να αναφέρω τον Γ. Αδαμόπουλο για τις πολύτιμες συμβουλές του και την υπομονή του μαζί μου.

Τέλος, οφείλω να εκφράσω τις ιδιαίτερες ευχαριστίες μου στην Α. Πολυκράτη, από το τμήμα μηχανογράφησης του Ε.Μ.Π., χωρίς τη βοήθεια της οποίας οποιαδήποτε απόπειρα υλοποίησης του συγκεκριμένου έργου θα ήταν αδύνατη.

Μυλωνάς Φοίβος – Απόστολος

Οκτώβριος 2001

## Περίληψη στα ελληνικά

Η παρούσα διπλωματική εργασία έχει ως κύριους στόχους της την ανάπτυξη ενός συστήματος ηλεκτρονικών εγγραφών των φοιτητών στα εξάμηνα σπουδών του Τμήματος Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου με βάση το Πρόγραμμα Σπουδών του, την ανάπτυξη κατάλληλων interfaces ελέγχου και παρεμβάσεων στις παραπάνω εγγραφές από το προσωπικό της γραμματείας και εν γένει την μεταφορά της σημερινής γραφειοκρατικής διαδικασίας με απλό, κατανοητό και εύχρηστο τρόπο στο Internet γενικά και το world wide web συγκεκριμένα.

Αν θελήσουμε να συνοψίσουμε τα βασικά στοιχεία αυτής της διπλωματικής εργασίας, αυτά επικεντρώνονται σε τρεις βασικούς άξονες. Ο πρώτος περιστρέφεται γύρω από τις λειτουργικές παραμέτρους και την αισθητική παρουσίαση της εφαρμογής, και οριοθετεί την ευχρηστία, την ευκολία πρόσβασης και την «ελαφρότητα» της, μέσω του web. Ο δεύτερος αφορά την αξιοπιστία, την ασφάλεια και την διασύνδεση με τα υπάρχοντα συστήματα με τα οποία καλείται να συνεργαστεί, και περιστρέφεται γύρω από τη χρήση βάσης δεδομένων για τα δεδομένα και την οργάνωση των χρηστών. Τέλος, ο τρίτος άξονας έχει να κάνει με την κατηγοριοποίηση των χρηστών του συστήματος σε ομάδες, καθεμία από τις οποίες έχει διαφορετική θεώρηση του περιεχομένου της εφαρμογής, διαφορετικές δυνατότητες πρόσβασης στα επιμέρους υποσυστήματά της και εν γένει διαφορετικές παρεχόμενες υπηρεσίες.

Συμπερασματικά, λοιπόν, πρόκειται για ένα πλήρες σύστημα ηλεκτρονικών εγγραφών, το οποίο αναμένεται να δώσει άμεσα λύσεις στα υπάρχοντα προβλήματα της διαδικασίας εγγραφής και το οποίο είναι ιδιαίτερα εύπλαστο και σε θέση να ακολουθεί κάθε φορά τις εκάστοτε παρουσιαζόμενες ανάγκες.

**Λέξεις Κλειδιά:** βάσεις δεδομένων, πληροφοριακό σύστημα εγγραφής σε μαθήματα, Internet, www.

## **Περίληψη στα αγγλικά / English summary**

The main goal of the current diploma thesis is the development of a fully integrated system, which will allow students of the Electrical & Computer Engineering Department of the National Technical University of Athens, to register online for their semester courses. Suitable control and access interfaces were developed for the above system, in order to be used from the registrar office, which is responsible for the whole registration process. Generally speaking, this thesis is responsible for the correct adaptation of the current procedure to the Internet and the world wide web, in a clear, precise and user-friendly way.

In order to summarize the basic directions of this thesis, it is suitable to talk about three main issues. The first one focused on the operational variables and the graphical layout of the project, which is designed for optimal user-friendliness, usability and scalability. The second one has to do with the reliability of the system, its security issues and its connectivity with current subsystems. Because of its nature, collaboration with several integrated operational systems is needed and therefore performed by its implementation. Finally, the third issue deals with the users, their needs, their access levels and their information feeds, and categorizes them in several groups according to the above.

In conclusion, the system developed in this diploma thesis is a fully organized, scalable system for online registrations and can be used from students, faculty and personnel of the Department. It provides to the point solutions of the current registration procedure problems and is adaptive to all future developments and expansions of the Course Program.

**Keywords:** databases, courses registration system, Internet, www

## Πίνακας περιεχομένων

# 1

## ΕΙΣΑΓΩΓΗ.....11

- 1.1 Αντικείμενο της διπλωματικής ..... 11
- 1.2 Οργάνωση του τόμου..... 12

# 2

## ΘΕΩΡΗΤΙΚΑ ΣΤΟΙΧΕΙΑ .....15

- 2.1 Internet, world wide web και σημερινή εποχή..... 15
  - 2.1.1 Πρωτόκολλα και υπηρεσίες ..... 16
  - 2.1.2 world wide web: Η κύρια δικτυακή υπηρεσία σήμερα ..... 18
- 2.2 Hypertext Transfer Protocol (http): ..... 19
  - 2.2.1 Ορισμός:..... 19
  - 2.2.2 Γενικά:..... 19
- 2.3 Secure Socket Layer (SSL) - Secure-Hypertext Transfer Protocol (s-http):..... 23
  - 2.3.1 Ορισμός:..... 23
  - 2.3.2 http vs. s-http:..... 23
- 2.4 Web caching: ..... 24
  - 2.4.1 Ορισμός:..... 24
  - 2.4.2 Γενικά:..... 24
- 2.5 Κατανομή φορτίου (load balancing):..... 26
  - 2.5.1 Εισαγωγή:..... 26
  - 2.5.2 Κατηγοριοποίηση συστημάτων..... 26
  - 2.5.3 Σύστημα εγγραφών: ..... 32
- 2.6 HyperText Markup Language (html):..... 32
  - 2.6.1 Ορισμός:..... 32
  - 2.6.2 Γενικά:..... 33
- 2.7 Javascript: ..... 34
  - 2.7.1 Ορισμός:..... 34
  - 2.7.2 Γενικά:..... 34
  - 2.7.3 Client-Side JavaScript: ..... 35
  - 2.7.4 Server-Side JavaScript: ..... 37
  - 2.7.5 Σύστημα εγγραφών: ..... 39

2.8	Extensible Markup Language (xml):	40
2.8.1	Ορισμός:	40
2.8.2	Γενικά:	40
2.9	Databases:	41
2.9.1	Εισαγωγή – Ορισμοί	41
2.9.2	Ιστορική αναδρομή	42
2.9.3	DBMS - Γενικά	43
2.9.4	Διαδικασία Ανάπτυξης Βάσεων Δεδομένων	46
2.9.5	Σχεσιακό μοντέλο	47
2.9.6	Σύνοψη	49
2.10	Structured Query Language (SQL):	49
2.10.1	Ορισμός:	49
2.10.2	Γενικά:	49
2.11	Oracle:	52
2.11.1	Γενικά:	52
2.11.2	Oracle8i:	53
2.12	Java:	54
2.12.1	Ορισμός	54
2.12.2	Εισαγωγή	55
2.12.3	Εργαλεία	56
2.12.4	Περιγραφή	57
2.12.5	Java - Javascript	59
2.13	Java Database Connectivity (JDBC)	59
2.14	Java Server Pages (JSP):	62
2.14.1	Ορισμός	62
2.14.2	Γενικά	62
2.14.3	Πρακτικά	64
2.14.4	Εκτέλεση JSP	69
2.14.5	Πλεονεκτήματα	70
2.14.6	http-session tracking:	72
2.15	JavaBeans:	74
2.15.1	Ορισμός-Γενικά	74
2.15.2	Πλεονεκτήματα	75
2.16	Business Components for Java (BC4J)	75
2.16.1	Ορισμός:	75
2.16.2	Είδη BC4J	76
2.17	Oracle Jdeveloper:	82

# 3

## ΠΕΡΙΓΡΑΦΗ ΠΡΟΒΛΗΜΑΤΟΣ .....83

3.1	Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών	83
3.1.1	Ιστορικά στοιχεία:	83
3.1.2	Τομείς:	84

3.2	Εκπαιδευτική διαδικασία και Πρόγραμμα Σπουδών .....	85
3.2.1	Στόχοι του Εκπαιδευτικού Προγράμματος Σπουδών .....	85
3.2.2	Μαθήματα .....	85
3.2.3	Διάρκεια των Σπουδών.....	86
3.2.4	Διάρθρωση του προγράμματος Σπουδών.....	86
3.2.5	Ροές .....	86
3.2.6	Απαιτήσεις για τη λήψη διπλώματος.....	87
3.2.7	Αλγόριθμος επιλογής μαθημάτων ροών.....	87
3.2.8	Διπλωματική Εργασία .....	88
3.2.9	Βαθμολογία – βαθμός διπλώματος.....	88
3.2.10	Αριθμητικά στοιχεία από το πρόγραμμα σπουδών.....	88
3.3	Ορισμός αναγκών .....	90
3.4	Ορισμός πρακτικού προβλήματος .....	91
3.5	Η μέχρι σήμερα Λύση.....	91
3.6	Λύση πρακτικού προβλήματος .....	93
3.7	Ομάδες Χρηστών ενός τέτοιου Συστήματος.....	94
3.8	Τελικό Ζητούμενο.....	95

# 4

## ΑΝΑΛΥΣΗ ΤΗΣ ΛΥΣΗΣ .....

4.1	Χρήση του Δικτύου .....	97
4.2	Χρήση Βάσης Δεδομένων.....	98
4.3	Χρήση τεχνολογίας JSP .....	98
4.4	Περιγραφή του Προγράμματος Σπουδών .....	99
4.5	Αναζήτηση - Εγγραφή Μαθημάτων ακαδημαϊκού εξαμήνου .....	99
4.6	Τμήμα εγγραφής μαθημάτων.....	100
4.7	Τμήμα επιλογής κατεύθυνσης / ροών .....	101
4.8	Υποσύστημα καθηγητών .....	102
4.9	Υποσύστημα γραμματείας.....	102
4.10	Υποσύστημα administrator .....	103

# 5

## ΣΧΕΔΙΑΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΗΕ.....105

5.1	Σχεδίαση της Βάσης Δεδομένων .....	105
5.1.1	Οντότητες.....	106
5.1.2	Κρυπτογράφηση passwords: .....	110
5.2	Σχεδίαση του προγράμματος ΗΕ .....	111
5.2.1	Εισαγωγή – επεξήγηση διαγραμμάτων .....	111
5.2.2	Σχεδίαση πολιτικής πρόσβασης: .....	118

# 6

## ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΗΕ.....121

6.1	Επιλογή Πλατφόρμας και Εργαλείων .....	121
6.2	Υλοποίηση βάσης δεδομένων:.....	122
6.3	Υλοποίηση BC4J: .....	124
6.4	Υλοποίηση JSP files: .....	125
6.5	Υλοποίηση <u>static</u> html: .....	126
6.6	Υλοποίηση dynamic html <u>από Javascript</u> : .....	127
6.7	Υλοποίηση Javascript: .....	129
6.8	Υλοποίηση Java: .....	130
6.9	Υλοποίηση επικοινωνίας με βάση φοιτητολογίου.....	138
6.10	Υλοποίηση hardware-OS: .....	139
6.11	Υλοποίηση BIND: .....	139
6.12	Υλοποίηση firewall:.....	140
6.13	Υλοποίηση SSL: .....	141



# 7

## ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ - MANUAL .....143

7.1	Εισαγωγή .....	143
7.2	Είσοδος στο Σύστημα – κεντρική Σελίδα.....	144
7.3	Υποσύστημα προπτυχιακών φοιτητών .....	145
7.3.1	Μαθήματα κορμού: .....	150
7.3.2	Μαθήματα ροών:.....	152
7.3.3	Υποχρεωτικά μαθήματα:.....	153
7.3.4	Μαθήματα επιλογής:.....	154
7.3.5	Links εξαμήνων:.....	155
7.3.6	Διαδικασία εγγραφής: .....	158
7.4	Υποσύστημα γραμματείας: .....	178
7.4.1	Εγγραφές μαθημάτων - Εποπτεία: .....	180
7.4.2	Εγγραφές μαθημάτων - Εισαγωγή νέας εγγραφής: .....	186
7.4.3	Εγγραφές μαθημάτων – Υποσύστημα Αναζήτησης:.....	186
7.4.4	Εγγραφές μαθημάτων – Έλεγχος & Εποπτεία: .....	193
7.4.5	Εγγραφές μαθημάτων – Εγγραφές προς αποστολή: .....	200
7.4.6	Εγγραφές κατεύθυνσης/ροών - Εποπτεία:.....	203
7.4.7	Εγγραφές κατεύθυνσης/ροών - Εισαγωγή νέας εγγραφής: .....	214
7.4.8	Εγγραφές κατεύθυνσης/ροών – Υποσύστημα Αναζήτησης: .....	215
7.5	Υποσύστημα administrator:.....	218
7.6	Έξοδος από το Σύστημα – logout page.....	227

# 8

## ΑΠΑΙΤΗΣΕΙΣ ΣΥΣΤΗΜΑΤΟΣ .....229

8.1	Απαιτήσεις client-side: .....	229
8.2	Απαιτήσεις server-side:.....	230

# 9

## ΕΠΙΛΟΓΟΣ .....231

9.1	Μελλοντικές Επεκτάσεις .....	231
9.2	Σύνοψη και Συμπεράσματα .....	232

# 10

## **ΒΙΒΛΙΟΓΡΑΦΙΑ.....235**

10.1	Βιβλία: .....	235
10.2	Links: .....	236
10.3	RFCs: .....	237
10.4	Περιοδικά: .....	237
10.5	pdf: .....	238

# 1

## *Εισαγωγή*

### *1.1 Αντικείμενο της διπλωματικής*

Το αντικείμενο της παρούσας διπλωματικής εργασίας σχετίζεται άμεσα με τις ανάγκες του Τμήματος Η.Μ.&Μ.Υ. για τεχνολογική εξέλιξη και αναβάθμιση όσον αφορά τις γραφειοκρατικές διαδικασίες και την διάδραση με τους φοιτητές του. Επικεντρώνεται, δε, στην δημιουργία ενός αξιόπιστου, εύχρηστου, λειτουργικού και φιλικού προς το χρήστη συστήματος, μέσω του οποίου θα είναι δυνατή η πλήρης αυτοματοποίηση της διαδικασίας εγγραφής ενός προπτυχιακού ή μεταπτυχιακού φοιτητή σε κάποιο από τα εξάμηνα φοίτησης, καθώς και ο διεξοδικός έλεγχος και η διαχείριση τέτοιου είδους πολλαπλών εγγραφών από το κατάλληλο προσωπικό.

Στα πλαίσια παραγωγής ενός ολοκληρωμένου μηχανογραφικού συστήματος και της απεμπλοκής της μέχρι σήμερα διαδικασίας εγγραφής από τα γκισέ της γραμματείας, βελτιστοποιήθηκαν οι παρεχόμενες λειτουργίες και υπηρεσίες του προς τους χρήστες, με γνώμονα κυρίως την τάχιστη εξυπηρέτησή τους, την επιτυχή ολοκλήρωση μιας εγγραφής και την καταχώρησή της στο κεντρικό σύστημα του φοιτητολογίου. Κατέστη, επίσης, δυνατή, μέσω της ορθής και προσεκτικής σχεδίασης του και της υλοποίησης με συγκεκριμένα εργαλεία και γλώσσες προγραμματισμού, η απροβλημάτιστη επεκτασιμότητα του σε ένα πολυ-σύστημα διάδρασης του Τμήματος, και των υπηρεσιών που αυτό παρέχει, με τους φοιτητές του.

## 1.2 Οργάνωση του τόμου

Η δόμηση του τόμου αυτού (**Τόμος Ι**) αναλύεται σε κεφάλαια και ενότητες κεφαλαίων. Συγκεκριμένα έχουμε:

Στο πρώτο κεφάλαιο (**Κεφάλαιο 1**) του τόμου ασχολούμαστε με την εισαγωγή στο αντικείμενο και τις ανάγκες της διπλωματικής εργασίας. Στην πρώτη ενότητα παρουσιάζεται το αντικείμενο της, οι ανάγκες του προβλήματος προς επίλυση και ο απώτερος στόχος του όλου εγχειρήματος.

Ακολουθεί (**Κεφάλαιο 2**) το θεωρητικό υπόβαθρο της εργασίας, παραθέτοντας μια αναλυτική αναφορά στις χρησιμοποιούμενες τεχνολογίες και τεχνικές κατά τη διάρκεια ανάπτυξής της. Ξεκινάμε στη πρώτη ενότητα του κεφαλαίου με μια αναφορά στο μέσο (Internet) και την επίδρασή του στη σημερινή καθημερινή ζωή και στη συνέχεια σταδιακά εισάγουμε τον αναγνώστη στις βασικές έννοιες και τεχνολογίες που χρησιμοποιήθηκαν. Η ενότητα 2 είναι αφιερωμένη στο πρωτόκολλο http, ενώ η ενότητα 3 προχωράει ένα βήμα παραπέρα στην secure έκδοσή του(s-http). Στην τέταρτη και πέμπτη ενότητα ασχολούμαστε με τεχνολογίες και τεχνικές web-caching και load balancing αντίστοιχα, στοιχείων απαραίτητων για την ορθή λειτουργία ενός τέτοιου μεγέθους συστήματος. Η έκτη, έβδομη και όγδοη ενότητα είναι αφιερωμένες στην παρουσίαση των κατά κόρων χρησιμοποιούμενων γλωσσών web προγραμματισμού, html, Javascript και xml, ενώ στις ενότητες 9 και 10 γίνεται μια εκτενής ανάλυση των σχεσιακών βάσεων δεδομένων και της γλώσσας SQL, με την επόμενη ακριβώς ενότητα 11 να έρχεται συνοδευτικά αναλύοντας τα βασικά ιδιαίτερα χαρακτηριστικά του χρησιμοποιούμενου εμπορικού πακέτου RDBMS της Oracle8i. Οι ενότητες 12 έως 15 σχετίζονται άμεσα με τη γλώσσα προγραμματισμού Java και τις στηριζόμενες σε αυτή χρησιμοποιούμενες από το σύστημα τεχνολογίες, ενώ το κεφάλαιο κλείνει με δύο ακόμα ενότητες αφιερωμένες στη θεμελιώδη λογική των Business Components for Java (BC4J) και του χρησιμοποιούμενου εργαλείου Oracle JDeveloper. Να σημειωθεί στο σημείο αυτό ότι η παραπάνω ομαδοποίηση έγινε με γνώμονα την καλύτερη κατανόηση των βασικών θεωρητικών δομών πάνω στις οποίες στηρίχθηκε η εκπόνηση αυτής της διπλωματικής εργασίας.

Στη συνέχεια (**Κεφάλαιο 3**) αναπτύσσεται διεξοδικά η σημερινή μορφή του Τμήματος Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών και η δομή του Προγράμματος Σπουδών του. Ιδιαίτερη έμφαση δίδεται, όπως είναι αναμενόμενο, στην επιλογή μαθημάτων και κατευθύνσεων/ροών, με πλήρη ανάλυση των αλγορίθμων επιλογής και των διαφορετικών δυνατοτήτων των φοιτητών. Συγκεκριμένα, στην πρώτη ενότητα παρατίθενται κάποια ιστορικά στοιχεία για το Τμήμα, ενώ στην αμέσως επόμενη επικεντρωνόμαστε στην εκπαιδευτική διαδικασία και το Πρόγραμμα Σπουδών. Κατόπιν, στηριζόμενοι στα παραπάνω, στις ενότητες 3 και 4 αναφερόμαστε στον ορισμό των αναγκών και του προβλήματος που στόχο έχει να λύσει η παρούσα διπλωματική εργασία. Στις επόμενες ενότητες 5 και 6 παρατίθεται η μέχρι σήμερα αντιμετώπισή του προβλήματος και η οριστική μέσω του σχεδιαζόμενου συστήματος λύση του. Ακολουθεί μια κατηγοριοποίηση στους εν δυνάμει χρήστες ενός τέτοιου συστήματος προκειμένου να διερευνηθεί και με αυτόν τρόπο τυχούσα ή μη χρησιμότητά του και τελικά στην τελευταία ενότητα του

κεφαλαίου έχουμε το τελικό ζητούμενο, το οποίο δεν είναι άλλο από την κατασκευή ενός τέτοιου συστήματος.

Κατόπιν (**Κεφάλαιο 4**), πραγματοποιείται συστηματική θεώρηση και ανάλυση της λύσης που θα ακολουθηθεί. Στην πρώτη ενότητα του κεφαλαίου εστιάζεται η λειτουργικότητα και χρησιμότητα της επιλογής του `www` ως `container` του όλου συστήματος, ενώ παρατίθενται και κάποια άλλα παραδείγματα αντίστοιχων συστημάτων ιδρυμάτων του εξωτερικού. Η δεύτερη ενότητα ασχολείται με την αναγκαιότητα της χρήσης RDBMS ως βάση πάνω στην οποία θα στηριχθεί ολόκληρο το σύστημα, παραθέτοντας ταυτόχρονα μια σειρά πλεονεκτημάτων μιας τέτοιας χρήσης. Στην τρίτη ενότητα γίνεται κατανοητή η χρησιμότητα της τεχνολογίας JSP για την παρουσίαση δυναμικά παραγόμενων σελίδων στο `web`, ενώ η επόμενη ενότητα αποτελεί μια μεταβατική εισαγωγή στα επιμέρους λειτουργικά χαρακτηριστικά της λύσης που περιγράφεται. Μετά την συνοπτική περιγραφή των βασικών χαρακτηριστικών του Προγράμματος Σπουδών στην επόμενη ενότητα, έρχεται η διεξοδική περιγραφή των δυνατοτήτων που παρέχονται στους επιμέρους χρήστες του συστήματος, ξεκινώντας από τους αναμενόμενους κύριους χρήστες του, τους προπτυχιακούς φοιτητές του Τμήματος. Κατόπιν, στις δύο επόμενες ενότητες αναλύονται σε όλα τα βήματά τους η διαδικασία εγγραφής φοιτητή σε μαθήματα οποιουδήποτε εξαμήνου και η διαδικασία εγγραφής/επιλογής κατεύθυνσης και ροών στο 6<sup>ο</sup> ονομαστικό εξάμηνο φοίτησης. Τέλος, οι τελευταίες τρεις ενότητες ασχολούνται με τα υποσυστήματα καθηγητών, προσωπικού γραμματείας και `administrator`, με ιδιαίτερη εκτενή αναφορά σε πρώτη φάση στις κυρίως διαχειριστικές δυνατότητες του προσωπικού της γραμματείας.

Στο πέμπτο κεφάλαιο (**Κεφάλαιο 5**) παρατίθεται αναλυτικά η σχεδίαση της βάσης δεδομένων του συστήματος στην πρώτη ενότητα, ενώ ακολουθεί η ενότητα της περιληπτικής σχεδίασης του συστήματος, κυρίως σε επίπεδο φυσικού σχεδιασμού του.

Το επόμενο κεφάλαιο (**Κεφάλαιο 6**) ασχολείται με την καθαυτό υλοποίηση του Συστήματος Ηλεκτρονικών Εγγραφών, ξεκινώντας από την περιγραφή της πλατφόρμας και των εργαλείων που χρησιμοποιήθηκαν κατά την ανάπτυξή του. Στην επόμενη ενότητα γίνεται αναφορά στα επιμέρους εργαλεία του RDBMS που συνέβαλαν στην ανάπτυξη της βάσης του, ενώ αμέσως μετά παρατίθενται πληροφορίες σχετικές με την αρχιτεκτονική BC4J και παρουσιάζονται κατάλληλα screenshots από Entity και View Objects του υλοποιηθέντος συστήματος προς καλύτερη κατανόηση των αναγραφόμενων. Ακολουθεί παρουσίαση ποσοτικών κυρίως χαρακτηριστικών από τη συγγραφή δυναμικών σελίδων με τη χρήση JSP files, με παράθεση όμως ενδεικτικών παραδειγμάτων από την ίδια την εφαρμογή. Όμοια διαδικασία ακολουθείται και για τα static html αρχεία στην επόμενη ενότητα, ενώ οι επόμενες δύο ενότητες ασχολούνται με στοιχεία και screenshots από τη συγγραφή τόσο δυναμικών σελίδων από Javascript, όσο και συναρτήσεων ελέγχου των στοιχείων που εισάγουν οι χρήστες οπουδήποτε στο σύστημα με Javascript. Στη συνέχεια ακολουθεί μια ενότητα παρουσίας στοιχείων για το τμήμα της εφαρμογής το οποίο έχει στηριχθεί στην γλώσσα Java, καθώς και μια μικρή ενότητα όπου γίνεται αναφορά στην τεχνοτροπία διασύνδεσης της βάσης με τη βάση του φοιτητολογίου του Ε.Μ.Π. και μεταφοράς των data από και προς σε αυτήν. Οι λειτουργικές παράμετροι ανάπτυξης του όλου συστήματος έπονται ενώ στην επόμενη ενότητα πραγματοποιείται εκτενής αναφορά στην λειτουργία του BIND για το υλοποιημένο

σύστημα. Τέλος, ακολουθεί συνοπτική περιγραφή του firewall που χρησιμοποιείται για την προστασία του συστήματος, ενώ πραγματοποιείται και μια μικρής έκτασης παρουσίαση του s-http implementation του συστήματος.

Στο έβδομο κεφάλαιο (**Κεφάλαιο 7**) του τόμου αυτού αναπτύσσεται το εγχειρίδιο χρήσης της αναπτυχθείσας εφαρμογής. Αρχικά έχουμε την ενότητα της εισαγωγής, όπου και ομαδοποιείται το περιεχόμενο της, ενώ ακολουθεί η ενότητα της εισαγωγής στο σύστημα, όπου παρουσιάζεται η διαδικασία login. Στην τρίτη ενότητα παρουσιάζεται πλήρως η πλοήγηση στο υποσύστημα των φοιτητών, στην τέταρτη αυτό της γραμματείας και στην πέμπτη το υποσύστημα του administrator του συστήματος. Το κεφάλαιο κλείνει με την ενότητα του τελικού logout από το σύστημα.

Το όγδοο κεφάλαιο (**Κεφάλαιο 8**) του παρόντος τόμου ασχολείται με τις απαιτήσεις του συστήματος από τους χρήστες του, οι οποίες και συνοψίζονται στην πρώτη ενότητά του.

Ο επίλογος της διπλωματικής εργασίας αναπτύσσεται στο ένατο κεφάλαιο (**Κεφάλαιο 9**) σε έκταση δύο ενότητων. Η πρώτη αναφέρεται στα συμπεράσματα που είμαστε σε θέση να εξάγουμε από το όλο εγχείρημα, ενώ στη δεύτερη γίνεται λόγος για κάποιες μελλοντικές επεκτάσεις, οι οποίες μπορούν να πραγματοποιηθούν στο Σύστημα Ηλεκτρονικών Εγγραφών, με σκοπό τον εμπλουτισμό των προσφερομένων υπηρεσιών και λειτουργιών προς τους χρήστες του.

Τέλος, στο δέκατο και τελευταίο κεφάλαιο του τόμου (**Κεφάλαιο 10**) αναφέρεται η πλήρης βιβλιογραφία που χρησιμοποιήθηκε για τη συγγραφή του παρόντος κειμένου, αλλά και για την ανάπτυξη της ίδιας της εφαρμογής, κατηγοριοποιημένη σε ενότητες ανάλογα το μέσο που χρησιμοποιήθηκε. Συγκεκριμένα, η πρώτη ενότητα παρουσιάζει τα βιβλία που χρησιμοποιήθηκαν ή γίνονται αναφορές μέσα στο κείμενο, η δεύτερη ενότητα τα http links από resources στο web, η τρίτη τα RFCs, η τέταρτη τα διάφορα περιοδικά που χρησιμοποιήθηκαν ως πηγές και η πέμπτη τα κείμενα σε μορφή Portable Data Format (pdf) .

# 2

## ***Θεωρητικά στοιχεία***

### ***2.1 Internet, world wide web και σημερινή εποχή***

Το διαδίκτυο (Internet) έχει μπει πλέον για τα καλά στην καθημερινή ζωή μιας μεγάλης πλειοψηφίας ανθρώπων της σημερινής κοινωνίας, και ιδιαίτερα εκείνων που σχετίζονται άμεσα με την ακαδημαϊκή κοινότητα και έχει γίνει αναπόσπαστο κομμάτι της. Πρόκειται για μια νέα πραγματικότητα, για έναν νέο τρόπο εργασίας, εκπαίδευσης και άντλησης πληροφοριών, πέρα βέβαια από τις όποιες άλλες δυνατότητες μπορεί αυτό να παρέχει στον άνθρωπο, όσον αφορά τη διασκέδαση και την ψυχαγωγία του.

Η περιήγηση στο Internet ανοίγει τους κρουνούς της πληροφορίας και της πληροφόρησης, με αποτέλεσμα τα πάντα να βρίσκονται σε απόσταση ενός click, αλλά και καταργεί τις γεωγραφικές αποστάσεις μεταξύ των ανθρώπων κάνοντας την επικοινωνία πιο εύκολη από ποτέ: αναφορές σε σημαντικά γεγονότα, τα τελευταία νέα, τα καιρικά φαινόμενα, οι εξελίξεις στην τεχνολογία ακόμα και τα αποτελέσματα αθλητικών αγώνων, αλλά και οι φίλοι και συγγενείς σε μια άλλη ήπειρο είναι πλέον στο δωμάτιό μας. Επιτυγχάνεται με άλλα λόγια γρήγορη και άμεση επικοινωνία ανθρώπων παγκοσμίως και σε όλα τα επίπεδα (ψυχαγωγικό, ενημερωτικό, επαγγελματικό, εμπορικό, κ.τ.λ.)...

Πώς όμως τα καταφέρνει όλα αυτά το Internet σήμερα;

Το Internet είναι το μεγαλύτερο δίκτυο υπολογιστών. Αριθμεί εκατοντάδες χιλιάδες υπολογιστές συνδεδεμένους μεταξύ τους και καινούριοι να συνδέονται σε αυτό με ολοένα αυξανόμενο ρυθμό. Ξεκίνησε κυρίως για εκπαιδευτικούς σκοπούς, αλλά

μετεξελίχθηκε σε ένα παγκόσμιο δίκτυο ποικίλων δραστηριοτήτων. Αποτελείται στην ουσία από πολλούς υπολογιστές συνδεδεμένους μεταξύ τους σε υποδίκτυα τοπικού χαρακτήρα, τα λεγόμενα Local Area Networks (LAN), τα οποία συνδέονται περαιτέρω υλοποιώντας μεγαλύτερα Wide Area Networks (WAN), τα οποία και συνεχίζουν την αλυσίδα κατά κάποιο τρόπο δημιουργώντας τελικά έναν πλήρως αποκεντρωμένο γράφο, με εκατομμύρια κόμβους-υπολογιστές. Χαρακτηριστικό του αποτελεί η εκ φύσεως παντελής έλλειψη κεντρικής διοίκησης, γεγονός που του προσδίδει τεράστια δύναμη ως μέσο.

Οι προοπτικές ανάπτυξης του ανοίγονται τεράστιες. Ήδη νέες μορφές των βασικών του πρωτοκόλλων κάνουν την εμφάνισή τους, με ριζοσπαστικές αλλαγές στην σχεδίαση και υλοποίησή τους (βλ. IPv6, [L2]), γεγονός που του δίνει νέες διαστάσεις. Το Internet2 [L4] βρίσκεται ήδη προ των πυλών με ταχύτητες πολλαπλάσιες του υπάρχοντος δικτύου, ενώ μια πληθώρα τεχνολογιών και εφαρμογών στηρίζεται, χρησιμοποιεί και αναπτύσσει το ίδιο το Internet.

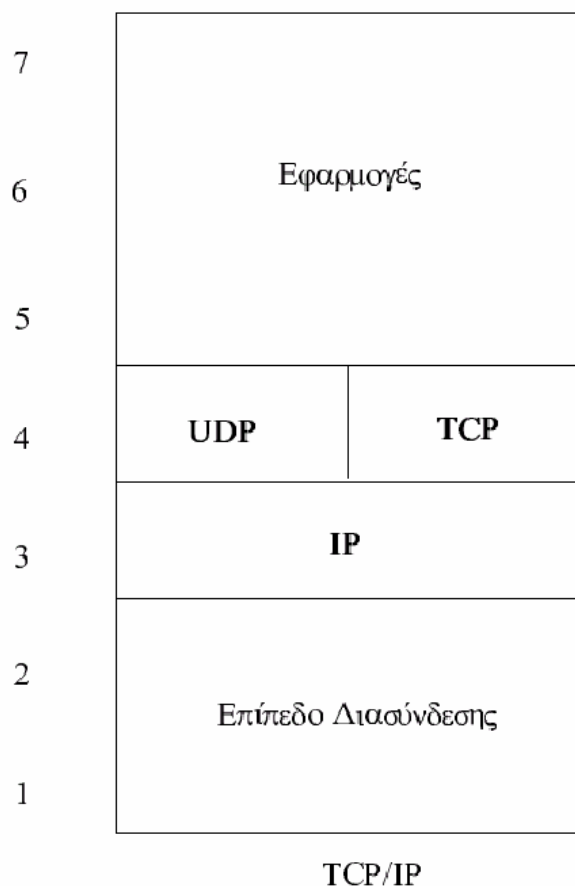
### ***2.1.1 Πρωτόκολλα και υπηρεσίες***

Προγραμματιστικά κοιτώντας, το Internet έχει χτιστεί πάνω σε μια σουίτα πρωτοκόλλων που είναι γνωστά με την ονομασία Transmission Control Protocol/Internet Protocol - TCP/IP (ή User Datagram Protocol/Internet Protocol - UDP/IP, ανάλογα το είδος της μεταφοράς των πακέτων) και αποτελούν de facto standards για τη μετάδοση δεδομένων σε αυτό. Η ανάπτυξή τους έγινε αρχικά από το Υπουργείο Εθνικής Άμυνας των Ηνωμένων Πολιτειών της Αμερικής (Η.Π.Α.) στη δεκαετία του 1970 και η πρώτη τους ευρεία χρησιμοποίηση πραγματοποιήθηκε στο δίκτυο υπολογιστών ARPANET, το πρώτο πειραματικό WAN με μεταγωγή πακέτου. Το NSFNET αποτέλεσε τη διάδοχη κατάσταση στις Η.Π.Α., για να φτάσουμε τελικά στη σημερινή μορφή του Internet γύρω στο 1993. Η ανάπτυξη του τελευταίου ήταν ταχύτατη και μάλλον άναρχη, με μία κεντρική αρχή διαχείρισης, τη λίγο έως πολύ γνωστή Internet Network Information Center – InterNIC.

Η διασύνδεση των τοπικών LANs γίνεται με μεταγωγείς πακέτου (ηλεκτρονικούς ή πλέον και οπτικούς), routers και gateways, που πραγματοποιούν δρομολόγηση βασισμένη στα πρωτόκολλα IP, η δε αξιοπιστία της μετάδοσης των πακέτων εξασφαλίζεται από τους ακραίους χρήστες μέσω των πρωτοκόλλων TCP ή UDP.

Η στοίβα των επιπέδων στην οικογένεια των TCP/IP προτύπων έχει ως εξής και βασίζεται στην αρχιτεκτονική των στρωμάτων, μεθοδολογία απαραίτητη στην υλοποίηση πρωτοκόλλων επικοινωνίας:





Εικόνα 2.1.1.α

Οι εφαρμογές που προσφέρονται στους τελικούς χρήστες είναι πολλές και ποικίλες. Ενδεικτικά αναφέρουμε:

- **www**: Ίσως η πιο διαδεδομένη από τις εφαρμογές και σίγουρα η υπεύθυνη για την μαζική απήχηση του Internet στους ανθρώπους. Πολύ απλή στη χρήση ακόμα και για τον αρχάριο χρήστη, εύκολα προσβάσιμη και με τεράστιες δυνατότητες επικοινωνίας, αποτελεί και τη βασική εφαρμογή που θα χρησιμοποιήσει το σύστημα εγγραφών.
- **e-mail**: Πολύ διαδεδομένη και αυτή η εφαρμογή, καθώς δίνει τη δυνατότητα εύκολης ανταλλαγής μηνυμάτων σε παγκόσμιο επίπεδο μέσα σε λίγα δευτερόλεπτα.
- **ftp**: Εφαρμογή μεταφοράς αρχείων υπολογιστών.
- **snmp**: Εφαρμογή διαχείρισης κόμβων ενός δικτύου
- **telnet**: Εφαρμογή πρόσβασης σε απομακρυσμένους υπολογιστές, κ.ο.κ.

Για περισσότερες πληροφορίες ο αναγνώστης καλείται να ανατρέξει στη σχετική βιβλιογραφία [MXKΣ94]

### **2.1.2 world wide web: Η κύρια δικτυακή υπηρεσία σήμερα**

Ας δούμε, όμως, λίγο αναλυτικότερα το world wide web (www), μιας και η χρήση του αποτελεί σημαντικότερο κομμάτι του συστήματος εγγραφών, και μοναδικό τρόπο πρόσβασης σε αυτό για τους χρήστες του.

Αν και γεννημένο σχετικά πρόσφατα - το 1989 στο CERN της Ελβετίας από τον Tim Berners-Lee [L3], με περαιτέρω εκλέπτυνσή του από τον ίδιο και τον Robert Cailliau το 1990 [11] - αυτό δε στάθηκε εμπόδιο στην ραγδαία εξάπλωσή του που συμπαρέσυρε και όλο το Internet σε τρομακτική ανάπτυξη μαζί του. Δίνει την δυνατότητα εύκολης και προσιτής αναπαράστασης της πληροφορίας, ακόμα και στον νέο χρήστη. Την δυνατότητα αυτή την οφείλει κυρίως στο γραφικό περιβάλλον που χρησιμοποιεί, στην δυνατότητα χρήσης υπηρεσιών πολυμέσων (Εικόνα – γραφικά – ήχος – video) και στην ανοιχτή σε νέους τύπους δεδομένων αρχιτεκτονική του. Είναι δομημένο σε σελίδες (web pages) και ανοικτό ουσιαστικά σε όλους, όσους με στοιχειώδεις γνώσεις της γλώσσας συγγραφής web pages **Hypertext Markup Language (html)** μπορούν να φτιάξουν μία σελίδα και να την κρεμάσουν σε κάποιον κόμβο-server του δικτύου. Η ουσία της ζωής του www είναι η δυνατότητα αλληλοσύνδεσης των σελίδων μέσω **hyperlinks** (ή απλά: **links**), δημιουργώντας τελικά έναν ιστό (web) από αυτές<sup>1</sup>. Τέλος, το πρωτόκολλο επικοινωνίας των κόμβων-υπολογιστών που έχουν κρεμασμένες τις σελίδες και λειτουργούν 24ώρες/24ώρο ως servers, με οποιονδήποτε υπολογιστή-client στο δίκτυο θελήσει να δει μία κρεμασμένη σελίδα, είναι συνήθως το Hypertext Transfer Protocol (http), ενώ σε περιπτώσεις που υπάρχει ανάγκη για extra ασφάλεια στη συναλλαγή χρησιμοποιείται το Secure - Hypertext Transfer Protocol – (s-http).

Οι clients χρησιμοποιούν web-browsers για την απεικόνιση των σελίδων και γενικά την πλοήγηση στο www, ενώ οι servers τρέχουν ειδικούς δαίμονες, τους web servers.

Στις επόμενες ενότητες του κεφαλαίου γίνεται μια εκτενής αναφορά στα θεωρητικά στοιχεία των τεχνολογιών που χρησιμοποιήθηκαν κατά την υλοποίηση του συστήματος εγγραφών, ενώ περαιτέρω πρακτική εφαρμογή και παράμετροι υλοποίησης τους συζητούνται στο κεφάλαιο 6 του Τόμου Ι. Υπάρχει σαφής κατηγοριοποίηση στις ενότητες αυτές και ο εξοικειωμένος αναγνώστης καλείται να προσπεράσει όποιες από αυτές θεωρεί ότι κατέχει πλήρως.

---

<sup>1</sup> Το www δηλαδή χρησιμοποιεί ένα σύστημα οργάνωσης και κωδικοποίησης πληροφοριών βασισμένο σε hypertexts, διευκολύνοντας έτσι την πλοήγηση των χρηστών στις διάφορες περιοχές δεδομένων. Το κυριότερο χαρακτηριστικό και πλεονέκτημα των hypertexts είναι η δυνατότητα δημιουργίας hyperlinks μεταξύ κειμένων που βρίσκονται ακόμα και σε διαφορετικούς υπολογιστές συνδεδεμένους σε κάποιο δίκτυο (εν προκειμένω στο Internet). Τα αρχεία αυτά των hypertexts είναι γνωστά σαν web pages.

## 2.2 Hypertext Transfer Protocol (http):

### 2.2.1 Ορισμός:

Το πρωτόκολλο http είναι ουσιαστικά ένα σύνολο κανόνων για την ανταλλαγή αρχείων (κειμένου, γραφικών, εικόνων, ήχου, video και άλλων) στο www. Είναι ένα πρωτόκολλο στρώματος εφαρμογής στη στοίβα TCP/IP, παντελώς ανεξάρτητο από τα υποκείμενα πρωτόκολλα σε αυτό, καθώς και από το δίκτυο στο οποίο υλοποιείται. Χρησιμοποιείται από το www από τη στιγμή που πρωτοεμφανίστηκε το 1990, με την πρώτη του έκδοση HTTP/0.9. Ενδιαμέσως υπήρξε η έκδοση HTTP/1.0 που είναι ευρέως διαδεδομένη, ενώ σήμερα χρησιμοποιούμε πλέον στις εφαρμογές μας την τελευταία έκδοση HTTP/1.1, που περιγράφεται με λεπτομέρεια στο **RFC-2616** [RFC1].

### 2.2.2 Γενικά:

Στη γενική του θεώρηση το πρωτόκολλο http είναι ανταλλαγής μηνυμάτων αίτησης/απάντησης (request/response protocol), όπου ο πελάτης (client) στέλνει ένα request στον εξυπηρετητή (server) και ο τελευταίος απαντάει με ένα response στον client. Κάθε server διαθέτει έναν http daemon, ο οποίος αναμένει πάντοτε requests από clients και φροντίζει για την εξυπηρέτησή τους. Πρακτικά, ένας client αποτελείται από κάποιον web browser, ο οποίος και στέλνει τα requests. Η διαδικασία που ακολουθείται κάθε φορά που ένας χρήστης ενός browser «ανοίγει» μια web σελίδα ή κάνει click πάνω σε ένα hyperlink είναι συνοπτικά η ακόλουθη:

Ο browser «φτιάχνει» το κατάλληλο http-request και το αποστέλλει στην IP-διεύθυνση που καθορίζεται από το URL. Στον server που βρίσκεται πίσω από αυτήν τη διεύθυνση, ο http daemon που τρέχει λαμβάνει το request και αφού ολοκληρώσει την επεξεργασία του, επιστρέφει το ζητούμενο αρχείο, διαμορφώνοντας το server http-response του.

Οι διευθύνσεις που γράφουμε στον browser μας κατά την πλοήγησή μας στο www, αποτελούν ουσιαστικά το συντακτικό ενός http\_URL:

```
http_URL = "http:" "://" host [ ":" port ] [ abs_path [ "?" query ] ]  
, όπου:
```

**host:** όνομα κάποιου server στο δίκτυο.

**port:** default για το http είναι το 80, αλλά μπορούμε να ορίσουμε εμείς κάποιο διαφορετικό. Αν δεν ορίζεται εννοείται το 80.

**abs\_path:** το «απόλυτο μονοπάτι» στο directory structure του server, όπου βρίσκεται το αρχείο μας.

Η κύρια διαφορά του HTTP/1.0 με το HTTP/1.1 έχει να κάνει με τον τρόπο ανταλλαγής της πληροφορίας μεταξύ client-server. Στο HTTP/1.0 κάθε ζεύγος request/response ολοκληρώνεται σε μία TCP σύνδεση μεταξύ client-server. Αντίθετα, στο HTTP/1.1 η default συμπεριφορά είναι να υπάρχουν περισσότερα του ενός ζεύγη ανταλλαγής πληροφορίας request/response μεταξύ client και server, πριν ολοκληρωθεί η current TCP σύνδεση μεταξύ τους. Αυτό ονομάζεται persistent connection και ελάφρυνε σημαντικά το φορτίο στους HTTP servers.

Τα πλεονεκτήματα του persistent connection είναι αρκετά:

- Θεωρώντας λιγότερες TCP συνδέσεις, εξοικονομούνται CPU time και μνήμη στους routers και στους hosts οποιουδήποτε είδους (clients, servers, proxies, gateways, tunnels, or caches)
- Μπορεί να γίνει pipelining<sup>2</sup> των http-requests/responses σε μία σύνδεση.
- Μειώνοντας τον αριθμό των πακέτων στο δίκτυο αποφεύγεται η περίπτωση συμφόρησης.
- Μειώνεται η καθυστέρηση (latency) σε συνεχόμενα requests, καθώς δεν υπάρχουν συνεχόμενα TCP handshakes σε αυτά.

Επίσης, σημαντικές διαφορές που βελτιώνουν κατά πολύ την απόδοση του πρωτοκόλλου είναι το γεγονός ότι εφόσον ένας browser δηλώσει ότι μπορεί να αποσυμπιέσει html και non-image data αρχεία, ο server τα συμπιέζει για τη μεταφορά τους στο δίκτυο, με αποτέλεσμα την μείωση του φορτίου που περνάει από αυτό. Επίσης, το HTTP/1.1 δίνει τη δυνατότητα για virtual hosting, ήτοι την αντιστοίχιση πολλών domain names στην ίδια IP address, δυνατότητα που θα χρησιμοποιήσουμε και στο υπό παρουσίαση σύστημα εγγραφών.

- **http-request:**

Το request του client περιλαμβάνει στην πρώτη γραμμή του την request method, το Uniform Resource Identifier (URI) και την έκδοση του πρωτοκόλλου που θα χρησιμοποιηθεί. Κατόπιν μετά από μια κενή γραμμή ακολουθεί το κύριο μέρος του μηνύματος, που περιέχει όλες τις προς μεταφορά πληροφορίες του client:

```
Request      = Request-Line
               * ( (general-header
                   | request-header
                   | entity-header ) CRLF )
               CRLF
               [ message-body ]
```

```
Request-Line  = Method SP Request-URI SP HTTP-Version CRLF
```

---

<sup>2</sup> Η τεχνική του pipelining επιτρέπει σε έναν client να κάνει πολλαπλά requests χωρίς να περιμένει για καθένα response.

Το Method token καθορίζει την μέθοδο που θα εφαρμοστεί. Δυνατές μέθοδοι είναι:

```
Method      = "OPTIONS"  
              | "GET"  
              | "HEAD"  
              | "POST"  
              | "PUT"  
              | "DELETE"  
              | "TRACE"  
              | "CONNECT"  
              | extension-method
```

**extension-method = token**

Ο return code ενός http-response ενημερώνει πάντοτε τον client αν κάποια από τις παραπάνω μεθόδους είναι ή όχι επιτρεπτή για το ζητούμενο resource στον server, μιας και αυτό το στοιχείο μπορεί να αλλάξει δυναμικά. Επίσης, μόνο οι μέθοδοι GET και HEAD είναι υποχρεωτικό να υποστηρίζονται από γενικού σκοπού servers, όλες οι υπόλοιπες είναι optional.

Ένα παράδειγμα Request Line από το σύστημά μας θα είναι:

```
GET /~she/index.html HTTP/1.1
```

- **http-response:**

Αφού λάβει και επεξεργαστεί ένα request message, ο server απαντάει με ένα http-response message. Το response message περιέχει στη πρώτη γραμμή του ένα status line, όπου αναφέρεται η έκδοση του πρωτοκόλλου και ένας κώδικας επιτυχίας ή αποτυχίας (3-ψήφιο αριθμητικό και κείμενο). Ακολουθεί μετά από μια κενή γραμμή το κύριο μέρος του μηνύματος, όπου περιέχονται οι όποιες πληροφορίες προς μετάδοση του server. Συγκεκριμένα:

```
Response      = Status-Line  
                *(( general-header  
                    | response-header  
                    | entity-header ) CRLF)  
                CRLF  
                [ message-body ]
```

**Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF**

Δυνατές τιμές για το 3-ψήφιο Status-code είναι:

- ο **1xx**: Informational - Request received, continuing process
- ο **2xx**: Success - The action was successfully received understood, and accepted

- ο **3xx**: Redirection - Further action must be taken in order to complete the request
- ο **4xx**: Client Error - The request contains bad syntax or cannot be fulfilled
- ο **5xx**: Server Error - The server failed to fulfill an apparently valid request

Κοινά χρησιμοποιούμενα παραδείγματα ζευγών Status-Code και Reason-Phrase είναι:

200 OK

The request has succeeded.

301 Moved

The requested resource has been assigned a new permanent URI and any future references to this resource SHOULD use one of the returned URIs. Clients with link editing capabilities ought to automatically.

401 Unauthorized

The request requires user authentication.

403 Forbidden

The server understood the request, but is refusing to fulfil it.

404 Not Found

The server has not found anything matching the Request-URI.

500 Internal Server Error

The server encountered an unexpected condition, which prevented it from fulfilling the request.

Οι παραπάνω κώδικες είναι επεκτάσιμοι και οι εφαρμογές δε χρειάζεται να γνωρίζουν όλους τους κώδικες για να λειτουργήσουν. Είναι απαραίτητο όμως να γνωρίζουν την τάξη του Status-Code, έτσι ώστε να μπορούν να τον κατατάξουν.

Ένα παράδειγμα Response-Line από το σύστημά μας θα είναι:

HTTP/1.1 200 OK

## 2.3 *Secure Socket Layer (SSL) - Secure-Hypertext Transfer Protocol (s-http):*

### 2.3.1 *Ορισμός:*

Το πρωτόκολλο (Secure-Hypertext Transfer Protocol) **s-http** είναι ουσιαστικά μια επέκταση του πρωτοκόλλου http που επιτρέπει την ασφαλή μετακίνηση και ανταλλαγή αρχείων στο www. Κάθε s-http αρχείο είναι encrypted και/ή διαθέτει ένα digital certificate. Αποτελεί δε μία εναλλακτική λύση στο τομέα αυτό, απέναντι στο Secure Socket Layer (SSL) πρωτόκολλο, με την κύρια διαφορά τους να εντοπίζεται στο ίδιο το στρώμα που λειτουργούν. Συγκεκριμένα το SSL λειτουργεί στο στρώμα μεταφοράς, με τα οφέλη που αυτό συνεπάγεται (π.χ. ανεξαρτησία από εφαρμογές), ενώ το s-http στο στρώμα εφαρμογών. Η φιλοσοφία κρυπτογράφησης είναι επίσης διαφορετική, με το SSL να κρυπτογραφεί όλο το επικοινωνιακό κανάλι, ενώ το s-http κάθε μήνυμα ανεξάρτητα, επιτρέποντας στο χρήστη την παραγωγή ψηφιακών υπογραφών σε οποιοδήποτε είδος μηνυμάτων κατά την επικοινωνία.

Συνοψίζοντας, το s-http είναι ένα ασφαλές επικοινωνιακό πρωτόκολλο, το οποίο λειτουργεί με ανταλλαγή μηνυμάτων, σχεδιασμένο να συνεργάζεται με το απλό http πρωτόκολλο και να το χρησιμοποιούν οι ίδιες εφαρμογές. Λεπτομερής περιγραφή του γίνεται στο **RFC-2660** [RFC3], ενώ τελευταία έκδοση είναι το S-HTTP/1.4. Το SSL είναι ένα ευρέως διαδεδομένο ασφαλές πρωτόκολλο για τη διαχείριση της ασφάλειας κατά τη μετάδοση ενός μηνύματος στο Internet, χρησιμοποιεί την public-and-private key κρυπτογράφηση της RSA με ψηφιακό certificate, ενώ πρόσφατα εξελίχθηκε στο Transport Layer Security (TLS).

### 2.3.2 *http vs. s-http:*

Οι κύριες διαφορές στα μηνύματα των δύο συναφών πρωτοκόλλων (**http** και **s-http**) εντοπίζονται στα εξής σημεία:

#### *Request:*

Το request του client περιλαμβάνει την μέθοδο secure στο Request-Line του, ενώ το URI πρέπει να είναι το \*. Τελικά η μορφή θα είναι με ένα παράδειγμα η εξής:

Secure                    \*                    Secure-HTTP/1.4

#### *Response:*

Το response του server πρέπει να έχει ως περιγραφή πρωτοκόλλου το Secure-HTTP/1.4. Π.χ.:

Secure-HTTP/1.4      200      OK

Γενικά, επίσης, οι όποιοι headers του απλού http πρέπει να βρίσκονται μέσα στο κύριο message body του s-http μηνύματος για λόγους ασφαλείας. Μόνη εξαίρεση στα παραπάνω αποτελούν συγκεκριμένοι headers, οι οποίοι πρέπει να είναι ορατοί σε agents που δεν έχουν πρόσβαση στα ενθυλακωμένα δεδομένα, οπότε και τοποθετούνται στα headers του s-http. Οτιδήποτε εκτός από τα “Content-Type” και “Content-Privacy-Domain” στο header του μηνύματος είναι προαιρετικά, ενώ το message body πρέπει να διαχωρίζεται από το header με δύο επιτυχή CRLFs.

## **2.4 Web caching:**

Αναμφισβήτητα, το βασικό πρόβλημα του www και του Internet γενικά, σήμερα, έχει να κάνει με το διαθέσιμο bandwidth και την καθυστέρηση στην information retrieval από το web. Ο χρόνος που μεσολαβεί από τη στιγμή που θα ζητηθεί από τον client μία web σελίδα, μέχρι τη στιγμή που η σελίδα αυτή θα φτάσει στον υπολογιστή του, ορίζεται ως web document latency. Και αναντίρρηση ένας από τους πιο ακριβούς πόρους του δικτύου είναι το bandwidth. Αν στα παραπάνω προσθέσει κανείς και την τάση που υπάρχει στους ανθρώπους για burst φαινόμενα επισκεψιμότητας, με ορισμένα web sites να δέχονται συνεχώς requests που ζητάνε συνεχώς τα ίδια documents, τότε εύκολα καταλαβαίνει το μέγεθος του προβλήματος.

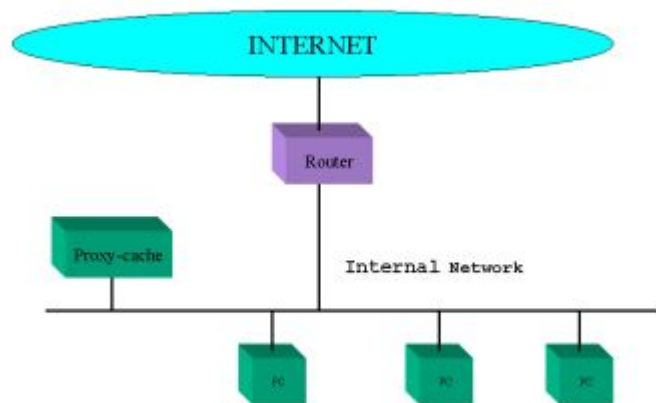
### **2.4.1 Ορισμός:**

Η βασική ιδέα πίσω από το web caching είναι να αποθηκεύονται τα ζητούμενα από έναν χρήστη αρχεία πληροφορίας σε κάποια κοινή τοποθεσία, και έτσι να αποφεύγεται η επαναλαμβανόμενη και ακριβή διαδικασία retrieving αυτών από τους ακόλουθους χρήστες που θα ζητήσουν τις ίδιες πληροφορίες. Το τελευταίο επιτυγχάνεται παρέχοντας στους υπόλοιπους χρήστες την ζητούμενη πληροφορία από την τοπική κοινή τοποθεσία και όχι από το original web site. Με τον τρόπο αυτό η εξυπηρέτηση των χρηστών γίνεται πλέον σε πολύ μικρότερο χρόνο, αφού πραγματοποιείται τοπικά. Με μετριοπαθείς μετρήσεις ένα μεσαίου μεγέθους σύστημα μπορεί να εξυπηρετήσει από την cache μέχρι και 60% των client requests.

### **2.4.2 Γενικά:**

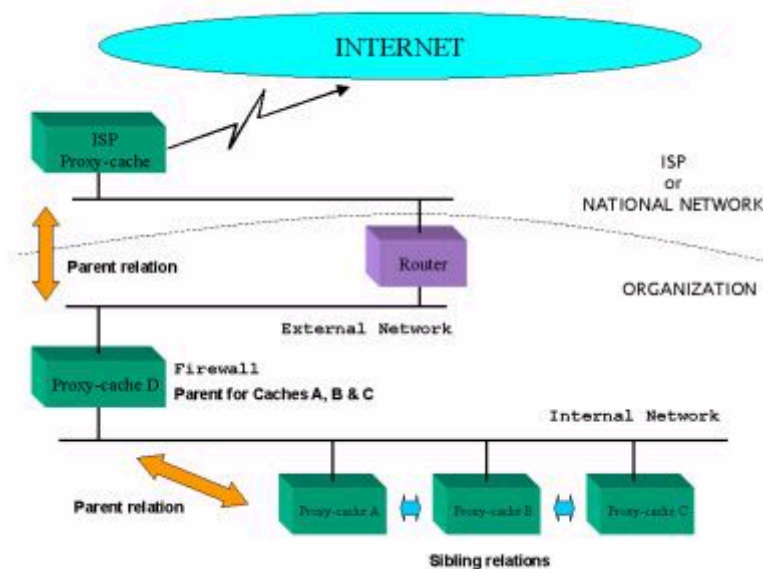
Διακρίνουμε δύο είδη web caching [L5], ανάλογα του ποιος πραγματοποιεί το caching: **browser caching** και **proxy-server caching**. Προφανώς η πρώτη επιλογή είναι client specific και εξαρτάται από το κατά πόσον ο χρήστης έχει ενεργοποιήσει την επιλογή στον browser του, η δε δεύτερη απαιτεί τον σχεδιασμό και υλοποίηση του server-side συστήματος με κατάλληλο εξοπλισμό, ενώ μπορεί να ενισχύσει και την ασφάλεια του όλου συστήματος. Τυπική σχεδίαση συστήματος με proxy-server μπορεί να θεωρηθεί η ακόλουθη:





Εικόνα 2.4.2.α

Σε περαιτέρω επίπεδα μπορεί ένα σύστημα να διαθέτει πολλαπλά επίπεδα web cache (multi-level web caching) και το πρωτόκολλο για την επικοινωνία αυτών των web caches μεταξύ τους είναι το Internet Cache Protocol (ICP), το οποίο και περιγράφεται στα **RFC-2186** [RFC4] και **RFC-2187** [RFC5]:



Εικόνα 2.4.2.β

Πρέπει να σημειώσουμε εδώ ότι είναι πολύ σημαντική σχεδιαστική απόφαση η χρήση ή μη proxy-server cache στο σύστημα και πρέπει να ληφθούν σοβαροί παράγοντες υπόψη. Για παράδειγμα, θα πρέπει να γνωρίζουμε εκ των προτέρων το είδος των web σελίδων που θα εξυπηρετεί ο server, καθότι μόνο στατικές html σελίδες θα πρέπει να τοποθετούνται στην cache και όχι δυναμικά παραγόμενες σελίδες, καθώς το περιεχόμενο των τελευταίων αλλάζει κάθε φορά που πραγματοποιείται προσπέλαση σε αυτές. Τυπικό παράδειγμα non-cacheable web σελίδων αποτελούν database queries, καθώς και securely transmitted σελίδες (π.χ.: SSL documents). Αντίθετα καθαρά static html σελίδες είναι θεμιτό να τοποθετούνται σε cache.

Για τις ανάγκες του συστήματος εγγραφών, θα βασιστούμε προς το παρόν στο ήδη υπάρχον σύστημα web cache του δικτύου δεδομένων του Εθνικού Μετσόβιου Πολυτεχνείου, το οποίο και χρησιμοποιεί την βέλτιστη λύση αυτή τη στιγμή στον τομέα αυτό, το Squid [L6]. Σε δεύτερο επίπεδο μπορεί να γίνει η βελτίωσή του με χρήση εξειδικευμένου proxy server για το στατικό κομμάτι του web site.

## ***2.5 Κατανομή φορτίου (load balancing):***

### ***2.5.1 Εισαγωγή:***

Οι βασικές αρχές που διέπουν έναν κατανεμημένο web server σήμερα έχουν αφενός μεν να αντιμετωπίσουν προβλήματα σχετιζόμενα με διαδικασίες προς - αλλά και από - τους χρήστες του συστήματος, αφετέρου δε προβλήματα που έχουν να κάνουν με το ίδιο το υλικό και το περιεχόμενο του web server. Έτσι, απαραίτητο στοιχείο θεωρείται η διαφάνεια προς τους χρήστες, οι οποίοι δεν θα πρέπει να καταλαβαίνουν και να «νοιώθουν», είτε με τη μορφή καθυστέρησης στην πρόσβαση, είτε με τη μορφή γρήγορων μεν αλλά εμφανών δε redirects που κουράζουν, την ύπαρξη στο παρασκήνιο συστήματος δυναμικής κατανομής φορτίου στους web servers. Επίσης, ισορροπημένη κατανομή των αιτήσεων από τους χρήστες και του φόρτου χρησιμοποίησης των στοιχείων του συστήματος πρέπει να θεωρείται δεδομένη, διαφορετικά φαινόμενα bottleneck σε κάποιο κόμβο δύνανται να εμφανιστούν, ενώ σε τρίτη φάση το ίδιο το content που παρουσιάζεται στους χρήστες μπορεί να βρίσκεται ήδη διάσπαρτο σε πολλά σημεία του συστήματος και να απαιτείται διαφορετική προσέγγιση και παρουσίασή του στους χρήστες, αν οι τελευταίοι ζητάνε διαφορετικά μεταξύ τους πράγματα από άποψη content.

### ***2.5.2 Κατηγοριοποίηση συστημάτων***

Τα διάφορα υπάρχοντα συστήματα διακρίνονται σε δύο μεγάλες ομάδες [L7], ανάλογα με το που πραγματοποιείται η κατανομή στους διάφορους κόμβους του συστήματος, στον client ή στον server, σε μία client-server προσέγγιση όπως είναι το web. Έτσι έχουμε τα client-based συστήματα και τα server-based συστήματα:

### 2.5.2.1 *client-based συστήματα:*

- **web-clients**, όπου η επιλογή και η κατανομή των requests στους αντίστοιχους servers πραγματοποιείται από τους ίδιους τους web-browsers των χρηστών, κατά τη διάρκεια εκπόνησης του request, χωρίς ουσιαστικά οι τελευταίοι να παρατηρούν οποιαδήποτε τέτοια διαδικασία.
- **client-side proxies**, όπου έχουμε τον web-browser να πραγματοποιεί λειτουργίες proxy κόμβου, cache-άροντας κάποιες web pages ανάλογα με κάποια κριτήρια, ο καθορισμός των οποίων απαιτεί την επέμβαση του χρήστη εκ των προτέρων και απαιτεί γι' αυτό τη συνεργασία του.

Και στις δύο περιπτώσεις ο client (browser ή proxy) επιλέγει τυχαία κάποιο node του cluster. Έχουν και οι δύο παραπάνω προσεγγίσεις το πλεονέκτημα της χαμηλής πολυπλοκότητας από την πλευρά του web-server. Από την άλλη πλευρά υπάρχουν προφανώς τα προβλήματα της έλλειψης μεταφερσιμότητας από την πλευρά του client, καθώς και της συνολικά μηδενικής επεκτασιμότητας του συστήματος.

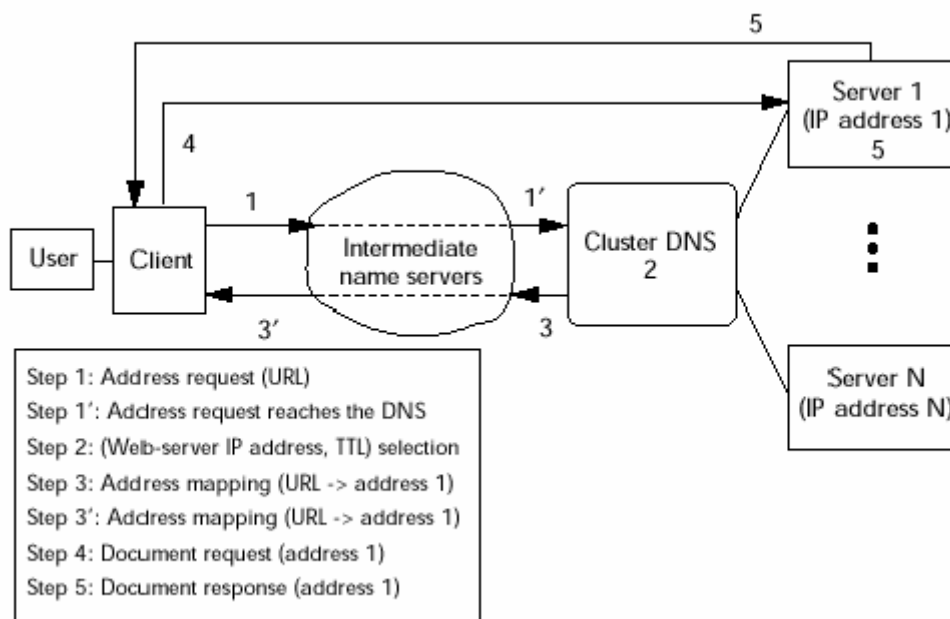
Δε θα ασχοληθούμε περαιτέρω με τα client based, καθώς η υλοποίησή τους δεν παρουσιάζει κάποιο ιδιαίτερο ενδιαφέρον ή δυσκολία, ενώ πάσχουν και από πλευράς user-transparency.

### 2.5.2.2 *server-based συστήματα:*

- DNS-based συστήματα
- dispatcher based συστήματα
- server-based συστήματα.

Μεμονωμένα για το κάθε ένα από τα τρία παραπάνω συστήματα μπορούμε να συνοψίσουμε τα εξής:

### DNS-based συστήματα:



Εικόνα 2.5.2.2.α

Όπως παρατηρούμε και στο παραπάνω σχήμα, η διαδικασία αποτελείται από 5 βασικά βήματα. Αρχικά γίνεται το request σε μορφή URL από τον client, αυτό περνάει από διάφορους name-servers και τελικά φτάνει στον cluster-DNS-server του συστήματος, ο οποίος με βάση κατάλληλο αλγόριθμο επιλογής server πραγματοποιεί το address mapping του URL στην τελική IP διεύθυνση του server που θα εξυπηρετήσει την αίτηση, στέλνοντας πίσω στον client την πληροφορία. Κατόπιν ο client ξεκινάει νέο request απευθείας στον server, την IP του οποίου πλέον γνωρίζει. Σε επίπεδο URL λοιπόν επιτυγχάνεται η θεμιτή διαφάνεια και η όλη διαδικασία «ασχολείται» μόνο με IP διευθύνσεις.

Οι αλγόριθμοι επιλογής διακρίνονται σε:

- ✓ constant TTL (Time To Live) και
- ✓ adaptive TTL.

Οι μεν πρώτοι, όπως «προδίδει» και το όνομά τους διατηρούν πάντα σταθερή τιμή TTL για τα εκάστοτε πακέτα και διακρίνονται όσον αφορά τη διαδικασία επιλογής στους εξής τέσσερις αλγόριθμους:

1. Round-Robin DNS
2. Server-state based
3. Client-state based
4. Συνδυασμός server και client αλγορίθμων,

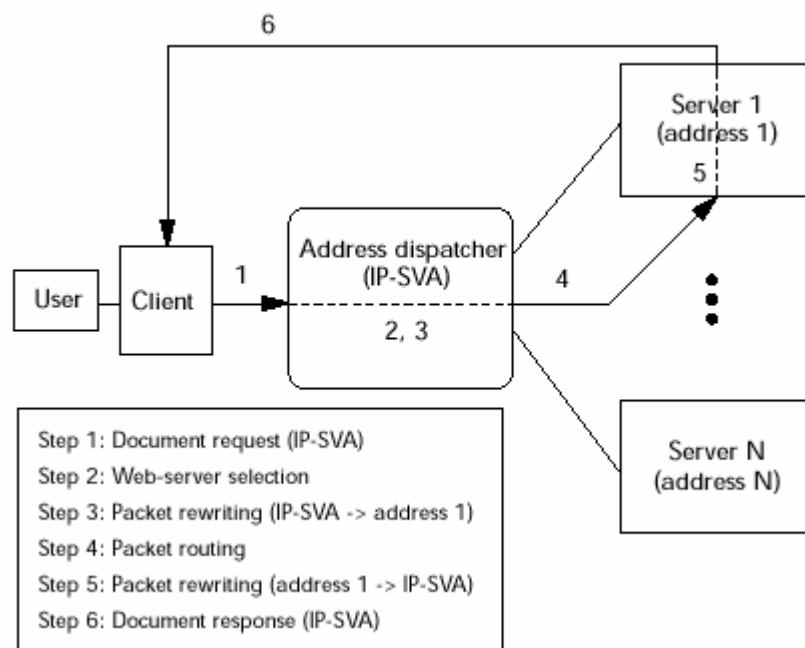
ενώ στους adaptive TTL η διαδικασία περιλαμβάνει δύο βήματα, την επιλογή του web server node με κάποιον από τους παραπάνω αλγόριθμους και την απόφαση για την TTL τιμή για τα πακέτα.

Συνοψίζοντας, στα DNS-based συστήματα δεν υπάρχει πλήρης έλεγχος των requests, λόγω των ιδίων των χαρακτηριστικών του DNS (caching, συν ότι μηδενικές ή πολύ μικρές τιμές για το TTL ενδεχομένως να απορριφθούν). Το πρόβλημα αυτό αντιμετωπίζεται εν μέρει χρησιμοποιώντας αλγόριθμους απόδοσης δυναμικού TTL. Η μέθοδος παρέχει κάποιες δυνατότητες επέκτασης και μπορεί να χρησιμοποιηθεί τόσο σε LAN όσο και σε WAN περιβάλλοντα.

#### Dispatcher-based:

Στα dispatcher-based συστήματα υπάρχει διαφάνεια στο επίπεδο IP, ενώ ο διαχωρισμός πραγματοποιείται με βάση το μηχανισμό δρομολόγησης σε:

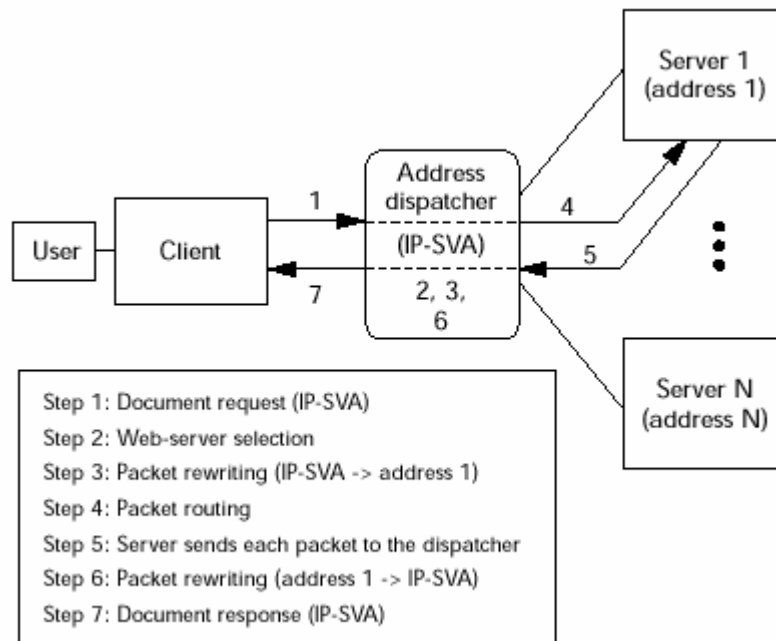
- packet single-rewriting



Εικόνα 2.5.2.2.β

Η διαδικασία αποτελείται από 6 βήματα και ξεκινάει εμφανώς πάλι με το request από τον client στον κόμβο που λειτουργεί ως dispatcher διευθύνσεων και κατέχει τη μία γνωστή IP-SVA διεύθυνση. Αυτός με τη σειρά του πραγματοποιεί την αντιστοίχιση στον κατάλληλο web server node, αφού προηγουμένως επανεγγράψει τη διεύθυνση του πακέτου και τοποθετήσει σε αυτή, τη διεύθυνση του server node (με όποια μορφή αυτή παρέχεται). Κατόπιν το πακέτο φτάνει στον server, εκεί γίνεται ξανά η διαδικασία επανεγγραφής της διεύθυνσής του στην αρχική well known IP-SVA και τελικά φτάνει η απάντηση στον client.

- *packet double-rewriting*



Εικόνα 2.5.2.2.γ

Η διαδικασία τώρα αποτελείται από 7 βήματα και ξεκινάει κλασικά με το request από τον client στον κόμβο που λειτουργεί ως dispatcher διευθύνσεων και κατέχει τη μία γνωστή IP-SVA διεύθυνση. Εκεί γίνεται πάλι η επιλογή του web server node, καθώς και η επανεγγραφή της διεύθυνσης του πακέτου με τη διεύθυνση του server. Κατόπιν δρομολογείται το πακέτο στον server, εκείνος στέλνει την απάντησή του πίσω στον dispatcher, εκεί πραγματοποιείται εκ νέου η διαδικασία επανεγγραφής (αυτή τη φορά από τη διεύθυνση του server και τη δική της μορφή στην IP-SVA διεύθυνση) και τελικά επιστρέφεται η ζητούμενη απάντηση στον client.

- *packet forwarding*

Όσον αφορά τώρα το packet forwarding έχουν επικρατήσει δύο τεχνικές:

- Network Dispatcher, όπου εφαρμόζεται κυρίως σε LANs και αποτελεί εμπορικό προϊόν, και το
- ONE-IP, το οποίο μπορεί περαιτέρω να διαχωριστεί σε routing-based και broadcast-based.

- *http-redirection*

Τέλος, στην περίπτωση της http-redirection τεχνικής, μπορούμε να έχουμε είτε server-state-based dispatching, είτε location-based dispatching, πάντοτε μέσω της χρήσης του http πρωτοκόλλου και των δυνατοτήτων που αυτό παρέχει με meta tags και directives.

Αξιολογώντας συνολικά τις dispatcher-based αρχιτεκτονικές θα μπορούσαμε να υποστηρίξουμε ότι υπάρχει πλήρης έλεγχος των client requests μέσω της χρήσης του dispatcher. Ωστόσο, η κεντρική αυτή διαχείριση ενδεχομένως να οδηγήσει σε bottleneck και single point of failure για όλο το σύστημα, με δυσάρεστες συνέπειες γι' αυτό. Παρόλα αυτά η τεχνική είναι εφαρμόσιμη τόσο σε LANs όσο και σε WANs (με κύρια εφαρμογή της σήμερα σε LANs), ενώ θα πρέπει να ληφθεί υπόψη και κάποιο overhead που υπάρχει λόγω της επεξεργασίας των πακέτων και του rewriting των διευθύνσεών τους, το οποίο βέβαια ποικίλει από υλοποίηση σε υλοποίηση.

#### Server-based συστήματα:

Στα server-based συστήματα έχουμε 2 level dispatching:

Ta client requests ανατίθενται απ' ευθείας από το DNS στους web servers, ενώ ένας server μπορεί να επαναθέσει ένα request σε οποιοδήποτε άλλο server! Ο διαχωρισμός πραγματοποιείται ανάλογα με το μηχανισμό του redirection σε:

- **http-redirection**, όπου στο πρώτο level γίνεται καθαρό Round-Robin DNS και στο δεύτερο level γίνεται packet rewriting με χρησιμοποίηση τυχόν πληροφορίας για τη κατάσταση του server ή hashes. Με τη παραπάνω διαδικασία προφανώς υπεισέρχεται κάποιο overhead, τόσο από το packet rewriting όσο και από το sharing της πληροφορίας των servers μεταξύ τους.
- **packet-redirection**, όπου στο πρώτο level γίνεται κι εδώ καθαρό Round-Robin DNS, ενώ στο δεύτερο γίνεται http-redirection με απόλυτη εξάρτηση από τη κατάσταση του server. Υπεισέρχεται και σε αυτή τη περίπτωση κάποιο overhead λόγω του information sharing και (κυρίως) λόγω του διπλασιασμού του αριθμού των συνδέσεων.

Συνοψίζοντας, η server-based αρχιτεκτονική βασίζεται σε κατανεμημένο αλγόριθμο επιλογής server και έχει αναμφισβήτητα μεγάλη επεκτασιμότητα, καθώς τα παραπάνω μπορούν να υλοποιηθούν τόσο σε LANs όσο και σε WANs. Παρέχεται, όπως και στην περίπτωση του dispatcher, πλήρης έλεγχος των client requests, ενώ μειονέκτημά της θεωρείται η ύπαρξη μεγάλου latency, είτε λόγω των πολλαπλών συνδέσεων, είτε λόγω του packet rewriting.

### 2.5.3 Σύστημα εγγραφών:

Η αναγκαιότητα ενός συστήματος δυναμικής κατανομής φορτίου στο σύστημα εγγραφών είναι κάτι παραπάνω από επιβεβλημένη. Η χρησιμότητα του web site είναι αναμφισβήτητη και η επισκεψιμότητά του αναμένεται να κυμανθεί σε υψηλά για το είδος του επίπεδα, της τάξεως των 1000 επισκέψεων ημερησίως. Κοινή διαπίστωση αποτελεί επίσης το γεγονός ότι η ποιότητα υπηρεσίας ενός web-server προς τους τελικούς χρήστες του συστήματος καθορίζεται από δύο παράγοντες, την ταχύτητα της μετάδοσης των δεδομένων στο δίκτυο (network-transfer speed) και το response-time του server, όπου μεν ο πρώτος δίνεται από το διαθέσιμο bandwidth, ενώ ο δεύτερος από τα διαθέσιμα resources και τη δυνατότητα δυναμικής κατανομής φορτίου.

Από τα παραπάνω συστήματα dynamic load balancing που εξετάσαμε, επιλέχθηκε τελικά για το σύστημα εγγραφών η λύση του DNS-based συστήματος με adaptive TTL αλγόριθμο και Round-Robin DNS, για λόγους απόδοσης, μιας και απαιτείται πλήρης διαφάνεια σε επίπεδο URL για τους χρήστες, οι οποίοι καλούνται να θυμούνται μόνο μία διεύθυνση:

<http://www.dblab.ntua.gr/~she>

, καθώς και δεν υπάρχει απαίτηση κάποιας χρονοβόρας προσπάθειας υλοποίησης ή/και αλλαγές στο υπάρχον setup του δικτύου δεδομένων του Πολυτεχνείου.

## 2.6 *HyperText Markup Language (html):*

### 2.6.1 Ορισμός:

Η Hypertext Markup Language (HTML), παρόλο που η ονομασία της παραπέμπει σε κάποια γλώσσα προγραμματισμού, δεν συγκαταλέγεται σε αυτές με την αυστηρή έννοια του όρου. Θα λέγαμε ότι είναι μάλλον ένα σύνολο από σύμβολα και κωδικούς (tags), που εισάγονται σε ένα αρχείο, το οποίο πρόκειται να απεικονιστεί σε έναν browser στο www. Τα tags αυτά καθοδηγούν τον browser στο να απεικονίσει στη σωστή θέση και με τον σωστό τρόπο τις λέξεις, τις εικόνες και ό,τι άλλο μπορεί να περιέχει μία html σελίδα. Τα περισσότερα από τα tags έρχονται σε ζεύγη, που υποδηλώνουν το πότε αρχίζει και το πότε τελειώνει κάποιο effect απεικόνισης στον browser.

Η html αποτελεί επίσημη σύσταση του World Wide Web Consortium (W3C) [L8] και σήμερα πλέον δεν νοείται browser που να μην την υποστηρίζει πλήρως, στην τελευταία έκδοσή της HTML 4.01 [L9]. Ωστόσο, μερικοί από τους πλέον δημοφιλείς browsers, όπως ο Microsoft Internet Explorer 5.5 και ο Netscape Navigator 6.0 δεν περιορίζονται στα standards της και υλοποιούν ορισμένα extra non-standard tags και extensions, με αποτέλεσμα είτε τον περιορισμό των επισκεπτών του site σε χρήστες με συγκεκριμένες εκδόσεις browsers, είτε τη λανθασμένη απεικόνιση στην οθόνη της ίδιας σελίδας ενός site από browser σε browser.



### 2.6.2 Γενικά:

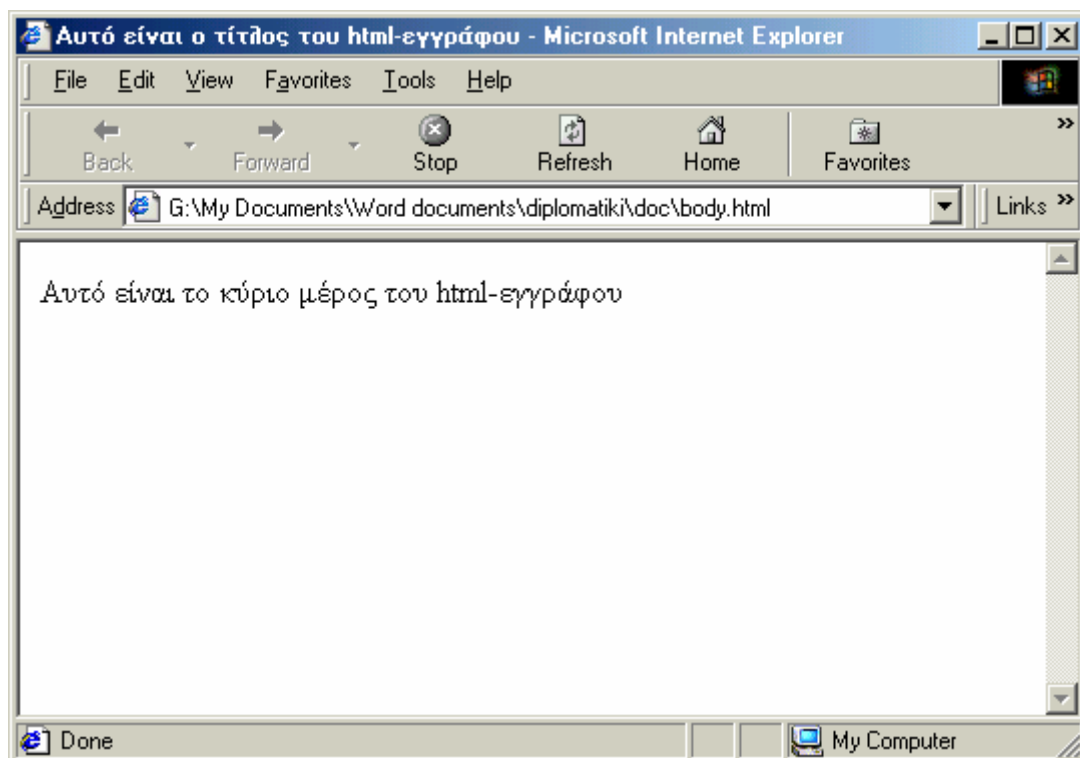
Πρέπει να τονίσουμε ότι η html, λόγω και της ίδιας της μορφής της, είναι εντελώς ανεξάρτητη από τις διάφορες πλατφόρμες υπολογιστών που κυκλοφορούν στην αγορά. Το μόνο που χρειάζεται είναι ένας browser για την απεικόνιση.

Για την κατασκευή ενός html-document πρέπει να ακολουθηθούν κάποιοι συντακτικοί κανόνες και σε αυτό το σημείο μόνο η html μοιάζει με τις κλασικές γλώσσες προγραμματισμού. Η γενική μορφή σύνταξης ενός απλού εγγράφου html είναι η ακόλουθη:

```
<html>
<head>
<title>Αυτό είναι ο τίτλος του html-εγγράφου</title>
</head>

<body>
Αυτό είναι το κύριο μέρος του html-εγγράφου
</body>
</html>
```

Το παραπάνω κομμάτι κώδικα στον Internet Explorer 5.5 απεικονίζεται ως εξής:



Εικόνα 2.6.2.α

Τέλος, στα τελευταία χαρακτηριστικά της έκδοσης html 4.0 είναι ανάμεσα στα άλλα:

- cascading style sheets, τα οποία παρέχουν τη δυνατότητα για έλεγχο του περιεχομένου των web σελίδων σε πολλαπλά επίπεδα
- δυνατότητες για πιο «πλούσιες» φόρμες
- υποστήριξη frames (η οποία ήταν ήδη υλοποιημένη σε πολλούς browsers)
- δυνατότητα για multilanguage κατανομή web σελίδων

## **2.7 Javascript:**

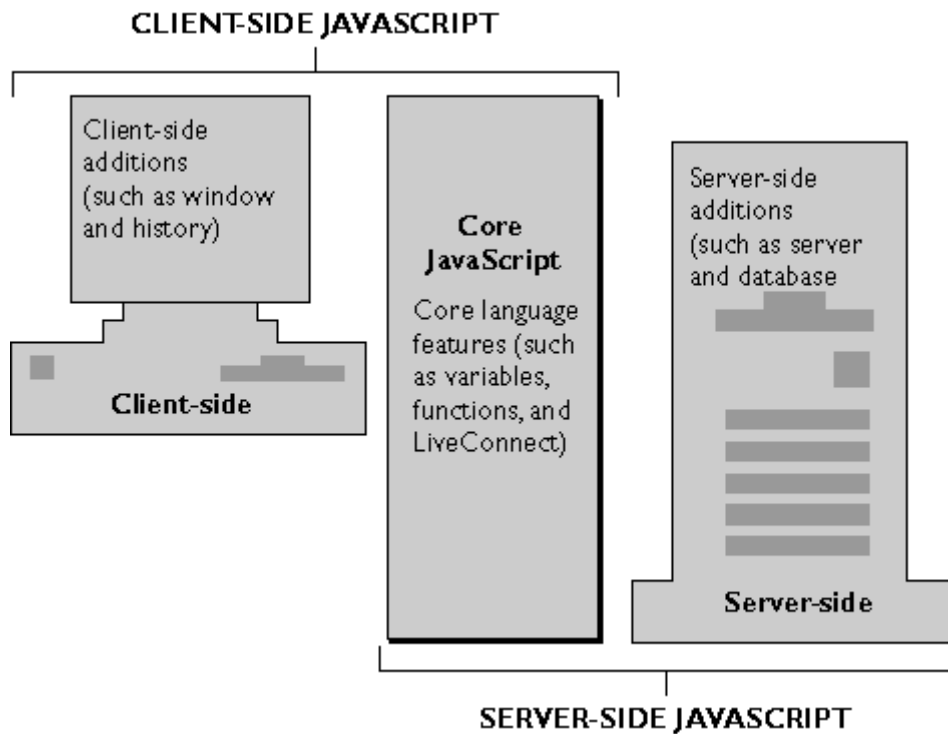
### **2.7.1 Ορισμός:**

Η **Javascript** είναι ουσιαστικά μία object-based scripting language, γεννημένη από την Netscape [L11] και όπως και η html παντελώς ανεξάρτητη από πλατφόρμες υπολογιστών. Χρησιμοποιείται ευρέως σε client εφαρμογές, αλλά περιλαμβάνει πλέον και server-side extensions. Η τελευταία έκδοσή της είναι η Javascript 1.3 [L10]

### **2.7.2 Γενικά:**

Με τη Javascript μπορεί κανείς να υλοποιήσει δυναμικές html σελίδες, οι οποίες επεξεργάζονται δεδομένα τα οποία εισάγει ο χρήστης, να χειρίζεται αρχεία, καθώς και να προσπελαύνει μέχρι και βάσεις δεδομένων. Τόσο η server-side όσο και η client-side Javascript μοιράζονται την ίδια core language, η οποία αποτελεί το standard ECMA-262 στο οποίο έχουν γίνει κάποιες προσθήκες.

Η core language περιέχει ένα σύνολο από core objects, όπως: Array, Boolean, Date, Function, Math, Number, Object, String. Επίσης καθορίζει τις εκφράσεις, τις δηλώσεις, τις εντολές, τους τελεστές, τις reserved words και τη γραμματική της γλώσσας. Οι βασικές προεκτάσεις της χωρίζονται στην client-side Javascript και στην server-side Javascript. Μία γενική δομή της γλώσσας φαίνεται στο ακόλουθο σχήμα:

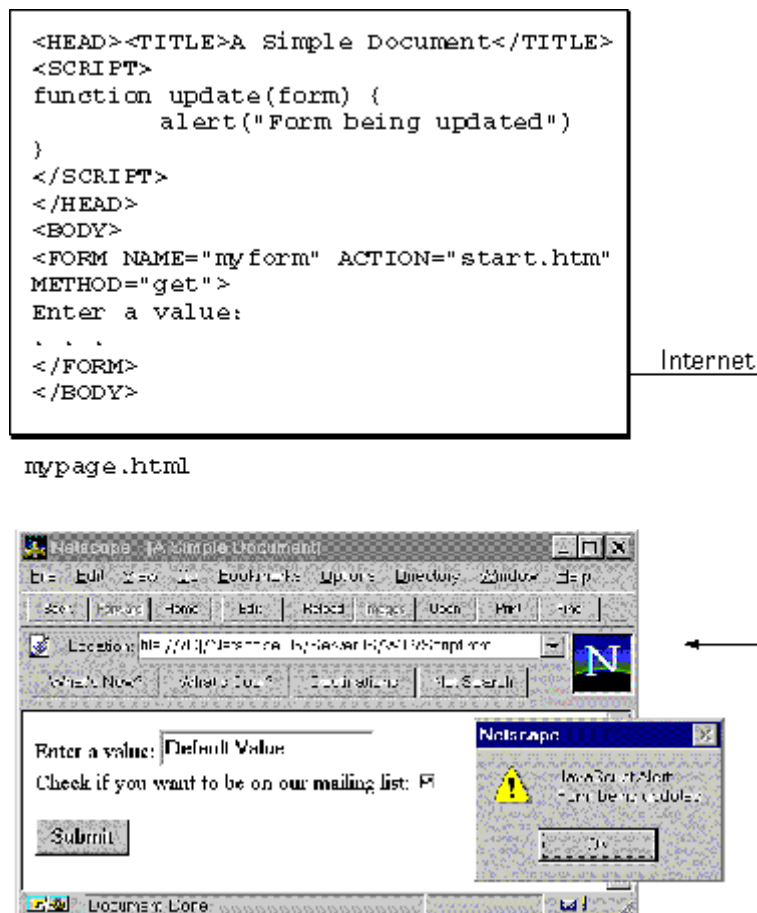


Εικόνα 2.7.2.α

Προφανώς οι προσθήκες στην core language της client-side Javascript διαφέρουν από εκείνες της server-side Javascript στο ότι η πρώτη ασχολείται αποκλειστικά με την εκτέλεση κώδικα Javascript στον client browser, ενώ η δεύτερη εκτελεί κώδικα Javascript στον εκάστοτε server. συνδυασμός και των δύο περιπτώσεων είναι δυνατόν να υπάρξει.

### 2.7.3 Client-Side JavaScript:

Στη σημερινή εποχή όλοι οι browsers υποστηρίζουν δηλώσεις client-side Javascript (για ιστορικούς λόγους να αναφέρουμε ότι όλοι οι browsers της Netscape από την έκδοση 2.0 και πάνω και της Microsoft από την έκδοση 3.0 και πάνω, υποστήριζαν Javascript), οι οποίες είναι ενσωματωμένες σε μία html σελίδα. Μόλις ο browser ζητήσει μια τέτοια html σελίδα, ο server στέλνει όλο το περιεχόμενο της σε αυτόν. Ο browser στον client τότε διαβάζει τη σελίδα από πάνω προς τα κάτω, απεικονίζει την html και εκτελεί τον κώδικα Javascript όπως προχωράει «κατεβαίνοντας» την σελίδα. Η όλη διαδικασία παρουσιάζεται στο παρακάτω σχήμα:



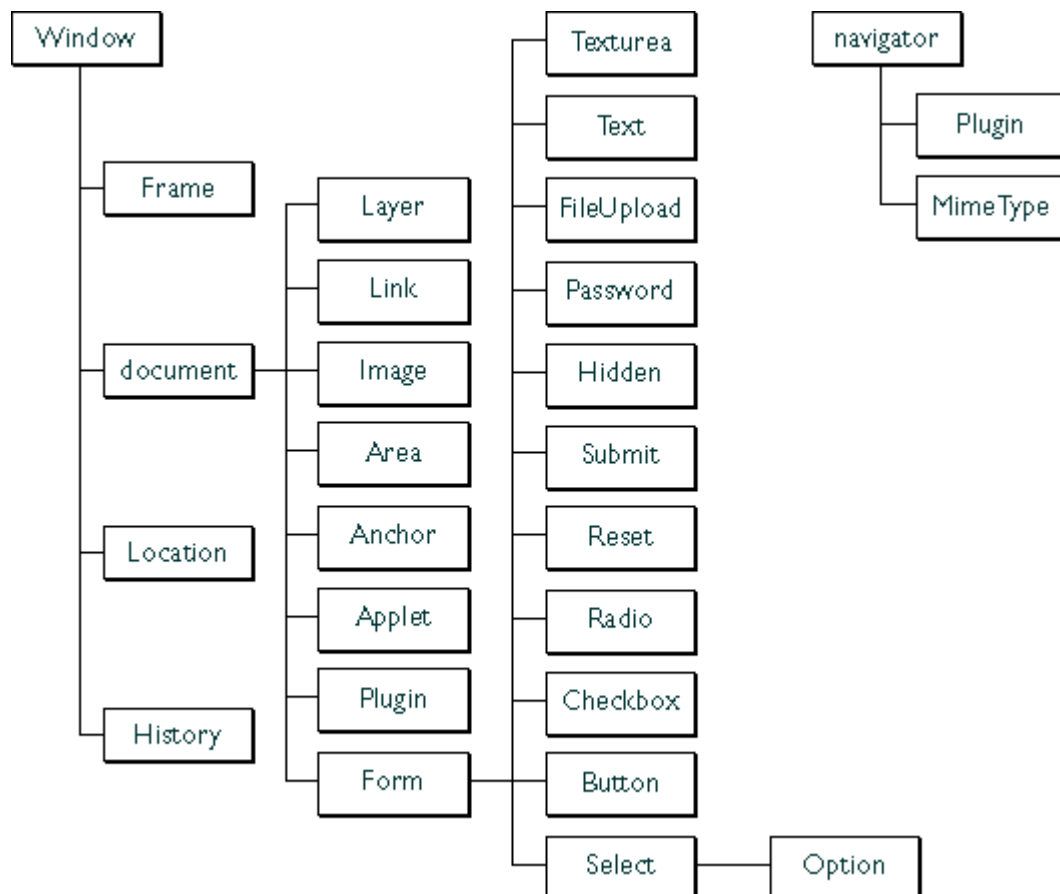
Εικόνα 2.7.3.α

Ο client-side JavaScript κώδικας μπορεί να αλληλεπιδράσει με events του χρήστη, όπως για παράδειγμα mouse clicks ή εισαγωγές στοιχείων σε φόρμες. Όλα τα παραπάνω πρέπει να τονίσουμε ότι πραγματοποιούνται αποκλειστικά στη μεριά του client, χωρίς να μεταφέρεται τίποτα στο δίκτυο, μετά την αρχική μεταφορά πληροφορίας ως απάντηση στο αρχικό request του, εξοικονομώντας έτσι πολύτιμους πόρους δικτύου και επιτυγχάνοντας άμεση αλληλεπίδραση με τον χρήστη.

Η client-side Javascript περιέχει κάποια επιπρόσθετα των core αντικείμενα. Αυτά είναι:

Anchor, Applet, Area, Button, Checkbox, document, event, FileUpload, Form, Frame, Hidden, History, Image, Layer, Link, Location, MimeType, navigator, Option, Password, Plugin, Radio, Reset, screen, Select, Submit, Text, Textarea, Window.

Συνδυασμός των παραπάνω αντικειμένων επιτρέπεται (π.χ.: document.form.checkbox.value), θεωρώντας μία ιεραρχία αντικειμένων ως εξής:



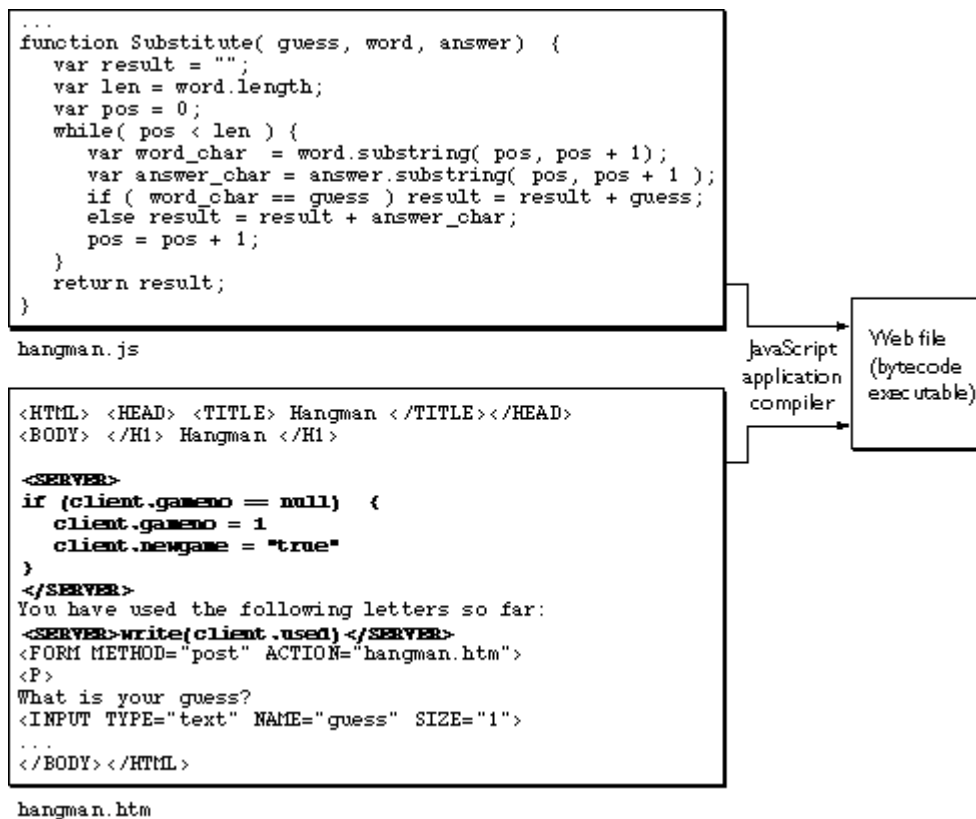
Εικόνα 2.7.3.β

#### 2.7.4 Server-Side JavaScript:

Η ίδια διαδικασία ενσωμάτωσης του Javascript κώδικα σε html σελίδες ακολουθείται και στην περίπτωση της server-side Javascript. Οι δυνατότητες που παρέχονται εδώ είναι σαφώς περισσότερες, αφού με χρήση κατάλληλων δηλώσεων μπορεί να γίνει προσπέλαση βάσεων δεδομένων, χειρισμός αρχείων στο file system του server και επικοινωνία με άλλες πολύπλοκες εφαρμογές μέσω Java ή/και LiveWire.

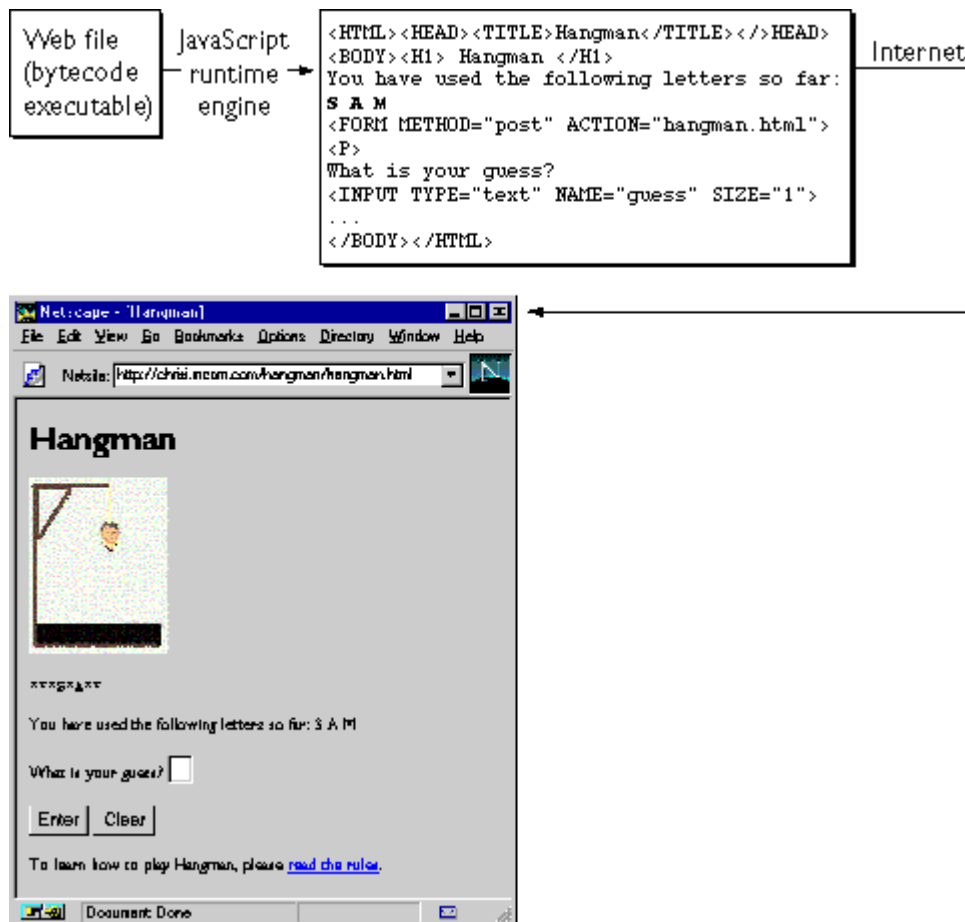
Μία σοβαρή αντίθεση σε σχέση με τη client-side Javascript έχει να κάνει με το γεγονός ότι ο server-side κώδικας γίνεται compile σε bytecode εκτελέσιμα αρχεία και θα πρέπει να υπάρχει υποστήριξη στον web server με κατάλληλη JavaScript runtime engine προκειμένου να μπορέσουν να εκτελεστούν αυτά. Γι' αυτό το λόγο η διαδικασία παραγωγής τέτοιου κώδικα αναλύεται σε δύο στάδια.

Στο πρώτο στάδιο παράγεται ο html κώδικας με ενσωματωμένο οποιοδήποτε είδος κώδικα Javascript και γίνεται compile σε ένα εκτελέσιμο αρχείο:



Εικόνα 2.7.4.α

Στη συνέχεια σε ένα δεύτερο στάδιο, καθώς ο browser ζητάει μια σελίδα, η runtime engine του server χρησιμοποιεί το εκτελέσιμο που παράχθηκε στο πρώτο στάδιο για να επιστρέψει στον client δυναμικά τη σωστή σελίδα. Η server-side Javascript προφανώς ξεχωρίζει από τα <server> </server> tags της στην html σελίδα που είναι ενσωματωμένη και αυτά μόνο εκτελούνται από τον server. Η διαδικασία έχει ως εξής:



Εικόνα 2.7.4.β

Τέλος, και η server-side Javascript προσθέτει τα δικά της αντικείμενα όπως αναφέρθηκε. Αυτά είναι τα ακόλουθα:

blob, client, Connection, Cursor, database, DbPool, File, Lock, project, request, Resultset, SendMail, server, Stproc

### 2.7.5 Σύστημα εγγραφών:

Στο συγκεκριμένο σύστημα εγγραφών περιοριστήκαμε στη χρήση client-side Javascript, κυρίως για λόγους απλότητας των web σελίδων. Επίσης, σημαντικό μερίδιο στην απόφαση αυτή είχε και η επιλογή της αποφυγής ανάμιξης διαφορετικών παρεμφερών τεχνολογιών, γεγονός που θα συνέβαινε, καθώς όλο το site είναι βασισμένο σε Java Server Pages, με αντίστοιχο configuration του (Java enabled) web server. Η χρήση της Javascript έγινε ως επί το πλείστον για validation του user input στις φόρμες εισαγωγής στοιχείων, για αισθητική αναβάθμιση/καθοδήγηση (βλ. π.χ. παράθυρο ημερολογίου, Εικόνα 7.3.6.1.στ ) και για υλοποίηση κανόνων του Οδηγού Σπουδών κατά την επιλογή μαθημάτων. Αναλυτικότερα θα γίνει περιγραφή σε επόμενα κεφάλαια.

## **2.8 Extensible Markup Language (xml):**

### **2.8.1 Ορισμός:**

Η Extensible Markup Language (XML) αποτελεί το παγκόσμιο format για δομημένα κείμενα και πληροφορία στο web. Με τον όρο «δομημένα κείμενα και πληροφορία» εννοούμε π.χ. spreadsheets, address books, οικονομικές δοσοληψίες, τεχνικά σχέδια, κ.α.. Η XML αποτελεί ένα σύνολο κανόνων, μετατροπών και guidelines για τη σχεδίαση text format γι' αυτά τα δεδομένα με τέτοιο τρόπο, έτσι ώστε να παραχθούν τελικά αρχεία που θα είναι εύκολα παραγόμενα, επεκτάσιμα και ευκολοδιάβαστα, μονοσήμαντα, δεν θα εξαρτώνται από πλατφόρμες υπολογιστών και θα είναι εύκολα παραμετροποιήσιμα σε επίπεδο localization/internationalization. Τελευταία έκδοση specifications για την XML είναι η XML 1.0 [L12], ενώ αποτελεί επίσημη σύσταση του World Wide Web Consortium (W3C) από το Φεβρουάριο του 1998 [L13], η οποία μαζί με τα RFC-1766 [RFC 7] και ISO/IEC10646, ISO-639, ISO-3166 προσφέρει πλήρη περιγραφή και κατανόησή της.

### **2.8.2 Γενικά:**

Και η XML, λοιπόν, όπως η html, χρησιμοποιεί tags και attributes. Όμως, ενώ η html καθορίζει επακριβώς τι σημαίνει και τι επίδραση έχει στον browser κάθε tag ή attribute, η XML χρησιμοποιεί τα tags μόνο για να διαχωρίσει ουσιαστικά την πληροφορία σε τμήματα και αφήνει ολοκληρωτικά την επεξεργασία της στην εφαρμογή που θα τη διαβάσει.. Για παράδειγμα, ενώ το <p> σε html ορίζει ρητά μία παράγραφο στο κείμενο, στην XML ένα <p> μπορεί να σημαίνει οτιδήποτε, ανάλογα με το περιεχόμενο του XML αρχείου.

Τα αρχεία XML είναι ουσιαστικά απλά αρχεία κειμένου, όπως άλλωστε και τα αρχεία της html, για να είναι εύκολα προσβάσιμα σε διορθώσεις. Όμως εδώ οι συντακτικοί κανόνες είναι πολύ πιο αυστηροί σε σχέση με την html, όπου και να ξέχναγε κάποιος το κλείσιμο ενός tag ή τα “ σε ένα attribute δεν γινόταν αντιληπτό από τον χρήστη. Στην XML ένα τέτοιο λάθος θα μετέτρεπε το αρχείο σε άχρηστο, καθώς η εφαρμογή που είναι υπεύθυνη για το διάβασμα του θα αποτύγχανε να το διαβάσει και θα προκαλούσε την άμεση παραγωγή error alert.

Στα πλαίσια του συστήματος εγγραφών έγινε χρήση XML κατά το configuration setup του tomcat και κατά την δημιουργία των BC4J με το εργαλείο Oracle JDeveloper. Παράδειγμα ενός τέτοιου XML αρχείου μπορεί να αναζητηθεί στο συνοδευτικό printout του κώδικα του συστήματος.



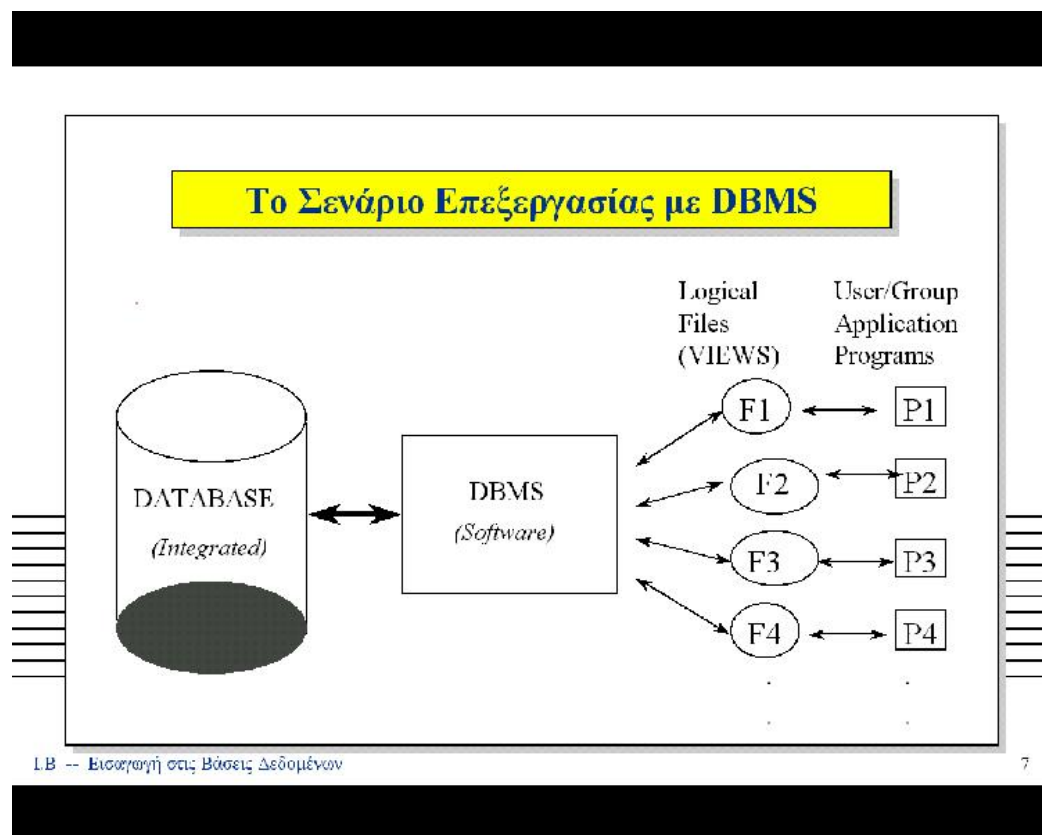
## 2.9 Databases:

### 2.9.1 Εισαγωγή – Ορισμοί

Με τον όρο «*database - βάση δεδομένων*» εννοούμε γενικά μια ολοκληρωμένη συλλογή από σχετιζόμενα δεδομένα, όπου τα δεδομένα είναι γνωστά γεγονότα που μπορούν να καταγραφούν και που έχουν κάποια υπονοούμενη σημασία. Έχει δε τις παρακάτω υπονοούμενες ιδιότητες:

- Αναπαριστά μια άποψη του πραγματικού κόσμου που ονομάζεται μικρόκοσμος.
- Είναι μια λογικά συνεκτική συλλογή δεδομένων που έχει κάποια εγγενή σημασία.
- Έχει κάποιο συγκεκριμένο σκοπό, απευθύνεται σε συγκεκριμένους χρήστες και προορίζεται για συγκεκριμένες εφαρμογές.

Μια βάση δεδομένων μπορεί να έχει οποιοδήποτε μέγεθος και πολυπλοκότητα και συντηρείται κατά κανόνα από κάποιο **Database Management System (DBMS)**. Το τελευταίο ορίζεται ως μια αυτοτελής συλλογή προγραμμάτων που επιτρέπουν στους χρήστες να δημιουργήσουν, να επεξεργαστούν και να συντηρήσουν μια βάση δεδομένων με ιδιαίτερα αποδοτικό τρόπο. Ο συνδυασμός της βάσης δεδομένων με το DBMS αποτελούν ένα Database System (DBS). Το σενάριο επεξεργασίας με ένα DBMS έχει σε γενικές γραμμές ως εξής:



Εικόνα 2.9.1.α

Ο *ορισμός* μιας βάσης δεδομένων περιλαμβάνει την προδιαγραφή των τύπων, των δομών και των περιορισμών των δεδομένων που θα αποθηκευτούν σε αυτή. Η *κατασκευή* της είναι η διαδικασία αποθήκευσης των δεδομένων σε ένα μέσο αποθήκευσης όπως π.χ. ο σκληρός δίσκος ενός υπολογιστή. Τέλος ο *χειρισμός* μιας βάσης δεδομένων περιλαμβάνει λειτουργίες όπως υποβολή queries για ανάκτηση συγκεκριμένων δεδομένων, ενημέρωση της βάσης έτσι ώστε να βρίσκεται σε συμφωνία με τον μικρόκοσμο που αναπαριστά, καθώς και παραγωγή αναφορών από τα δεδομένα.

### **2.9.2 Ιστορική αναδρομή**

Ήταν το 1970 όταν ο Ted Codd ανέπτυξε το θεωρητικό ορισμό του *σχεσιακού μοντέλου* δεδομένων, το οποίο αποτέλεσε το θεμέλιο λίθο της τεχνολογίας βάσεων δεδομένων όπως τη γνωρίζουμε σήμερα. Το μοντέλο αυτό χαρακτηρίζεται από τη δυνατότητα μαζικής επεξεργασίας των δεδομένων και από την ανεξαρτησία των εφαρμογών από τη φυσική υλοποίηση.

Όμως οι βάσεις δεδομένων δεν ήταν ιδιαίτερα διαδεδομένες στη δεκαετία του 1970 και ιδιαίτερα στους τελικούς χρήστες. Στα μέσα της δεκαετίας κυριαρχούσαν, αν μπορεί να ειπωθεί κάτι τέτοιο, δύο σχεσιακά συστήματα, η Ingres και το System R. Σήμερα απόγονοι των παραπάνω συστημάτων θεωρούνται η Ingres, η Sybase και ο Microsoft SQL Server για την Ingres, και η Oracle και IBM DB2 για το System R.

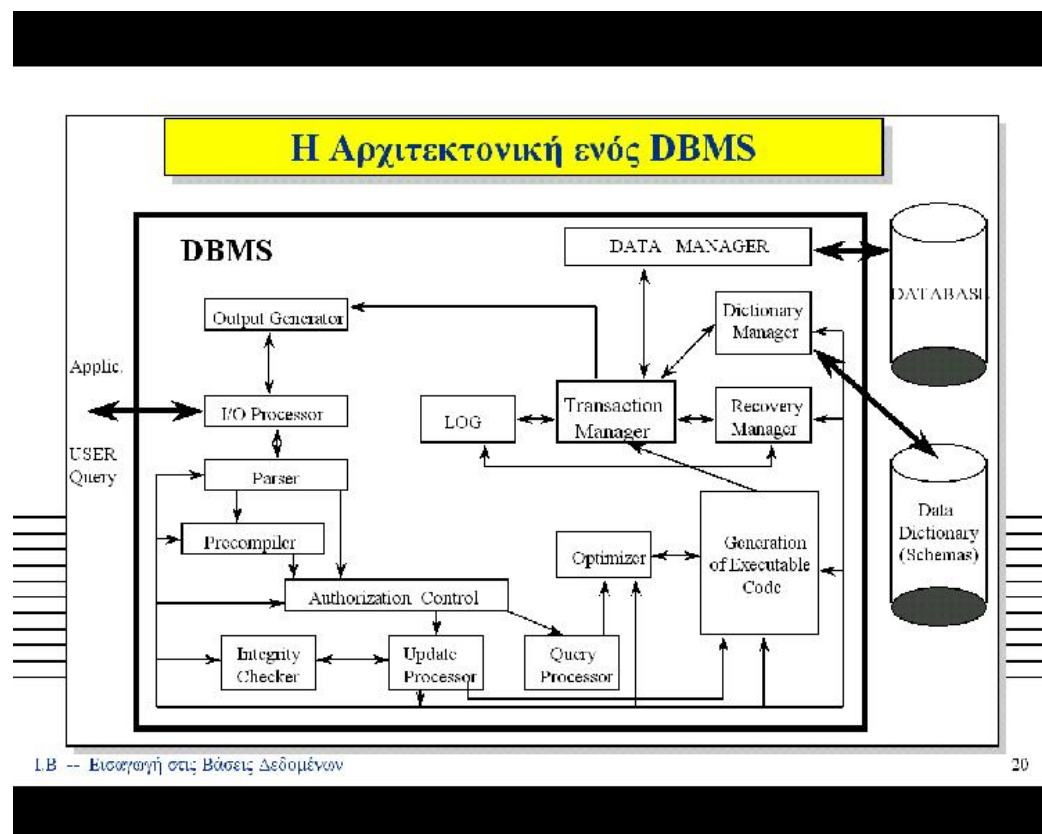
Στις αρχές της δεκαετίας του 1980 ξεκίνησε η ουσιαστική ανάπτυξη των σχεσιακών συστημάτων, με την Oracle να κυριαρχεί σταδιακά στο χώρο. Την αλματώδη ανάπτυξη τη δεκαετία του 1980 βοήθησαν σημαντικά και άλλοι παράγοντες, όπως η ανάπτυξη υψηλού επιπέδου ισχυρότατων γλωσσών προγραμματισμού και επεξεργασίας, η πρόοδος στη θεωρία των βάσεων δεδομένων και η ανάπτυξη εν τέλει της ίδιας της Structured Query Language (SQL). Ήταν αναμφισβήτητα η εποχή των Relational Database Systems. Τέλος, προς το τέλος της δεκαετίας έκαναν την εμφάνισή τους και τα πρώτα αντικειμενοστρεφή και ευφυή συστήματα.

Η προηγούμενη δεκαετία του 1990 ήταν η εποχή των σύνθετων οντοτήτων στις βάσεις δεδομένων, όπως τα πολυμέσα, software components και engineering αντικείμενα τα οποία διαμόρφωσαν μια νέα κατηγορία συστημάτων βάσεων δεδομένων, τα Object-Relational Database Systems. Περαιτέρω, γινόταν όλο και περισσότερο λόγος για έννοιες όπως κατανεμημένες βάσεις δεδομένων, παράλληλοποίηση, συστήματα βάσεων δεδομένων σε προσωπικούς υπολογιστές, συστήματα βάσεων δεδομένων στο web.

Τελικά σήμερα η βάση δεδομένων αποτελεί αναπόσπαστο κομμάτι οποιουδήποτε medium ή large scale συστήματος, τόσο σε επίπεδο προσωπικών υπολογιστών, όσο και στο Internet. Με χρήση ενός προγράμματος βάσης δεδομένων απλοποιείται και διευκολύνεται πληθώρα εργασιών, από την πιο απλή, όπως η οργάνωση μιας αποθήκης, έως την πιο σύνθετη, όπως η συλλογή δεδομένων από ένα διαστημόπλοιο. Στο Internet δε, με τον όγκο της πληροφορίας να αυξάνεται συνεχώς, η χρήση database καθίσταται επιτακτική για οποιοδήποτε αρκετά μεγάλο web site. Για περισσότερες πληροφορίες ο αναγνώστης καλείται να ανατρέξει στη σχετική βιβλιογραφία [Βασ99].

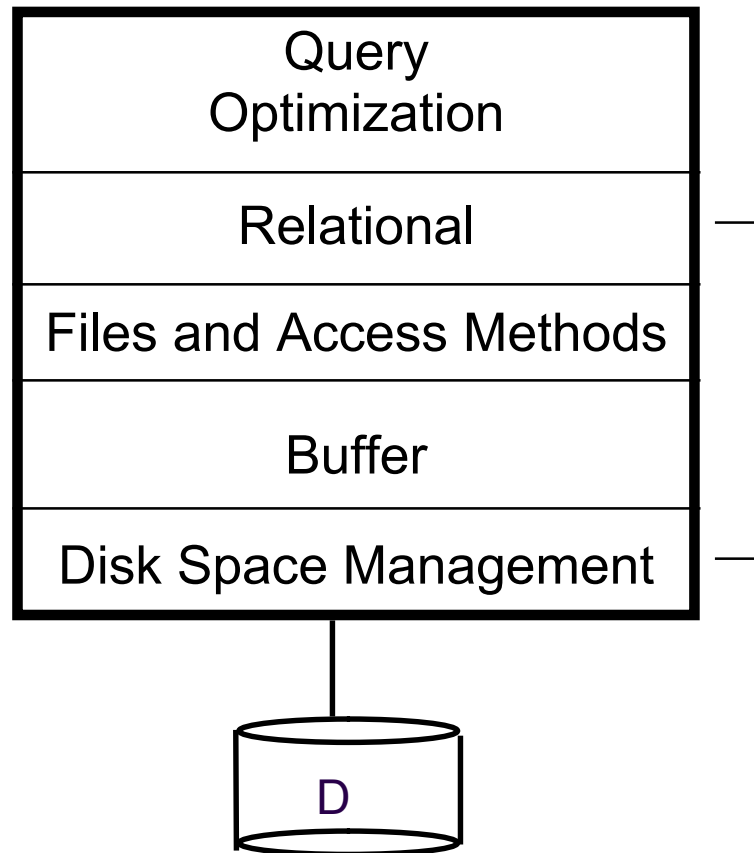
### 2.9.3 DBMS - Γενικά

Η αρχιτεκτονική ενός DBMS παρουσιάζεται αναλυτικά στο παρακάτω σχήμα και σε πρώτη όψη τρομάζει τον ανάδοχο αναγνώστη. Με μια πιο ψύχραιμη ματιά παρατηρεί κανείς την ύπαρξη διακριτών οντοτήτων, κάθε μια από τις οποίες έχει συγκεκριμένη αποστολή και στόχο στο όλο σύστημα και η οποία αποτελεί αναπόσπαστο τμήμα του.

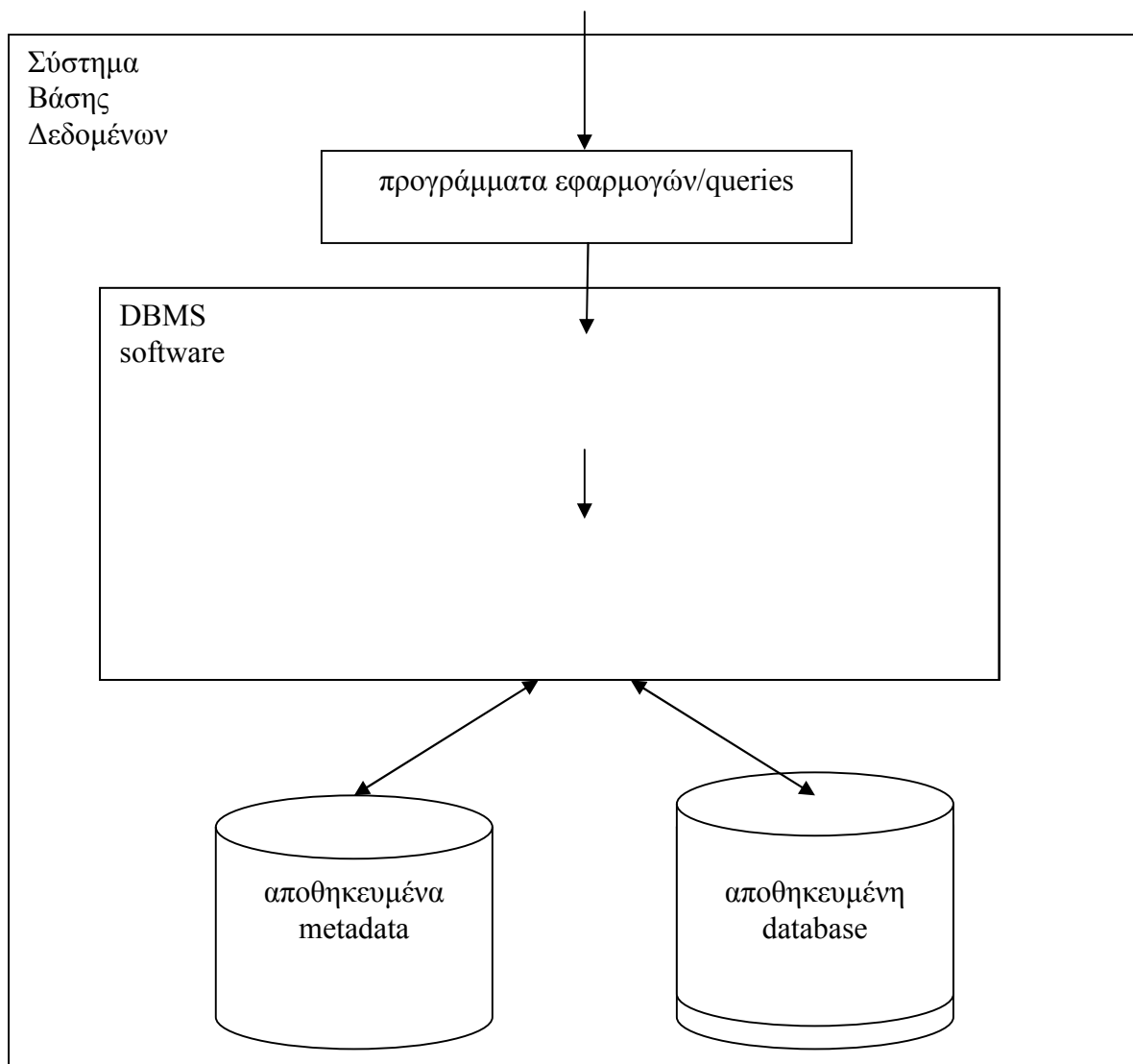


Εικόνα 2.9.3.α

Αν απλουστεύσουμε λίγο τα παραπάνω, θα καταλήγαμε ίσως στα εξής σχήματα:



Εικόνα 2.9.3.β



Εικόνα 2.9.3.γ

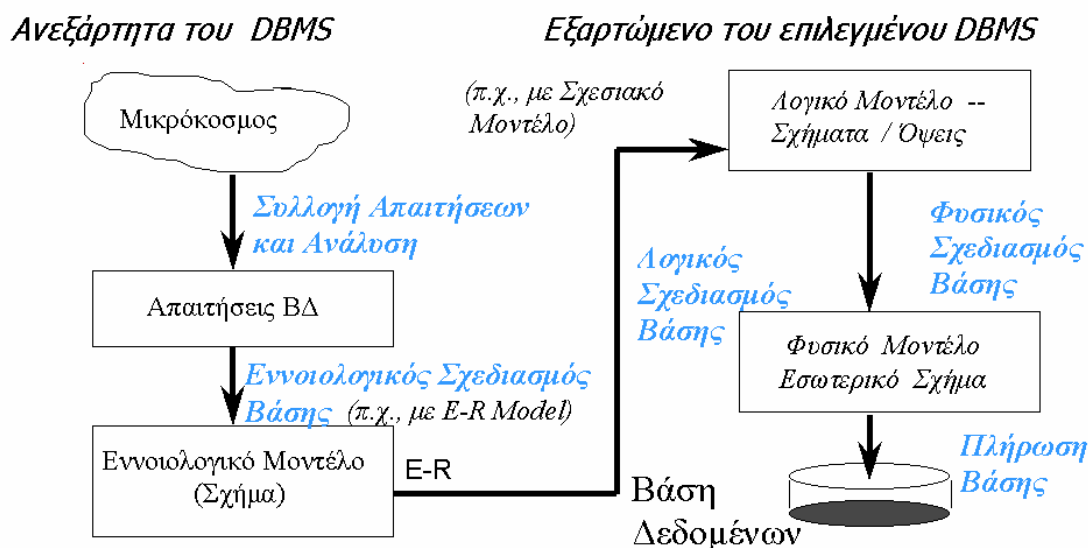
Και λέμε ίσως, γιατί είναι πιθανές διάφορες αρχιτεκτονικές DBMS χωρίς κάποια να είναι δεσμευτική. Ο βασικός λόγος της ευρείας χρήσης και διάδοσης των DBMS είναι η οργάνωση των πληροφοριών μέσα από εξειδικευμένους μηχανισμούς αποθήκευσης των δεδομένων, καθώς και οι user-friendly γλώσσες και περιβάλλοντα αναζήτησης τους. Μεταξύ των πλεονεκτημάτων, που παρέχει η διαχείριση μιας βάσης δεδομένων μέσα από ένα DBMS, είναι η ανάκαμψη από καταστροφές του συστήματος, η συνδρομικότητα, η γρήγορη ανάπτυξη εφαρμογών, η ακεραιότητα, η ασφάλεια, κ.α. Στην ανεξαρτησία των προγραμμάτων από τις φυσικές δομές βοηθούν πεδία αφαίρεσης, ενώ η επιτυχής διαχείριση αυτή καθεαυτή ενός τέτοιου συστήματος αποτελεί ιδιαίτερα δύσκολη εργασία που πραγματοποιούν υπεύθυνοι διαχειριστές (Database Administrators - DBAs).

#### 2.9.4 Διαδικασία Ανάπτυξης Βάσεων Δεδομένων

Η διαδικασία της ανάπτυξης μιας βάσης δεδομένων είναι ουσιαστικά η διαδικασία μετατροπής της γνώσης του πραγματικού μικρόκοσμου σε μια βάση δεδομένων και για καλύτερη κατανόηση της διακρίνεται σε συνεχείς αυτοτελείς διαδικασίες:

- ο Συλλογή απαιτήσεων και ανάλυση
- ο Εννοιολογικός σχεδιασμός της βάσης
- ο Λογικός σχεδιασμός της βάσης
- ο Φυσικός σχεδιασμός της βάσης
- ο Φόρτωση της βάσης με δεδομένα

### Πλήρης Διαδικασία Ανάπτυξης ΒΔ



Εικόνα 2.9.4.α

Η *πρώτη φάση* ανάπτυξης της βάσης δεδομένων είναι απαραίτητη για την «εξόρυξη» των απαιτήσεων από περιγραφές του πραγματικού μικρόκοσμου. Πρόκειται για μία δύσκολη εννοιολογικά διαδικασία η οποία απαιτεί εμπειρία και γνώση, καθώς συχνά στηρίζεται σε ασαφείς και ημιτελείς περιγραφές.

Στη συνέχεια ακολουθεί ο *σχεδιασμός* της βάσης, το αποτέλεσμα του οποίου θα είναι και το σχήμα της. Ο σχεδιασμός πραγματοποιείται ακολουθώντας ένα συγκεκριμένο μοντέλο, ενώ υπάρχουν τρεις διαφορετικοί αυτοτελείς σχεδιασμοί που δημιουργούν τελικά τρία διαφορετικά εννοιολογικά σχήματα της βάσης: Ο εννοιολογικός σχεδιασμός, ο λογικός σχεδιασμός και ο φυσικός σχεδιασμός. Αναλυτικότερα:

- Ο εννοιολογικός σχεδιασμός αποτελεί ουσιαστικά μία προσπάθεια για ξεκαθάρισμα των εννοιών και ως κύριο στόχο έχει την παραγωγή μίας

αφαιρετικής, αλλά και ταυτόχρονα πλήρους περιγραφής του τμήματος του μικρόκοσμου που θα αναπαρασταθεί στη βάση. Στη φάση αυτή πραγματοποιείται ο προσδιορισμός των σημασιολογικών αντικειμένων, των σχέσεων μεταξύ τους καθώς και των κανόνων που ισχύουν εν γένει. Το εννοιολογικό μοντέλο που κυριάρχησε, από τη δεκαετία του 1980 κιόλας, είναι το μοντέλο οντοτήτων-συσχετίσεων (**Entity-Relationship model, E-R**). Στους βασικούς λόγους της ευρείας διάδοσης του συγκαταλέγονται η απλότητά του, η σαφήνεια του και ο γραφικός συμβολισμός που ακολουθεί.

- Ο λογικός σχεδιασμός ασχολείται με την διαδικασία μετατροπής ενός εννοιολογικού μοντέλου σε τυπικά σχήματα εκφρασμένα στο υποστηριζόμενο από το DBMS μοντέλο δεδομένων (π.χ., **Σχεσιακό Μοντέλο**).
- Τέλος, ο φυσικός σχεδιασμός έχει ως κύριο στόχο την **απόδοση** του όλου συστήματος και αποτελεί τη διαδικασία καθορισμού των αρχείων που υλοποιούν τον λογικό σχεδιασμό και των οργανώσεων αυτών, καθώς και των δομών ευρετηρίων που αποτελούν το εσωτερικό σχήμα.

Η τελική φάση είναι η διαδικασία φόρτωσης της βάσης με τα δεδομένα, συνήθως μέσω βοηθητικών προγραμμάτων του DBMS για μαζική επεξεργασία.

Για τις παραπάνω φάσεις υπάρχουν διάφορα προγραμματιστικά εργαλεία που αυτοματοποιούν τις διαδικασίες σε μεγάλο βαθμό, όπως π.χ. ο Oracle Designer / Developer.

### 2.9.5 Σχεσιακό μοντέλο

Στη σημερινή εποχή το πλέον δημοφιλές μοντέλο είναι το σχεσιακό που προτάθηκε όπως αναφέρθηκε από τον Codd το 1970. Σχεδόν όλα τα μεγάλα DBMS είναι καθαρά σχεσιακά. Η βασική αρχή μιας σχεσιακής βάσης δεδομένων είναι ότι είναι ένα σύνολο από σχέσεις (relations-tables), που είναι και το μοναδικό είδος δομών. Κάθε στήλη στον πίνακα έχει επικεφαλίδα που ονομάζεται γνώρισμα (attribute-field) και πεδία τιμών, ενώ κάθε γραμμή ονομάζεται πλειάδα (tuple) και παριστάνει τα χαρακτηριστικά μιας οντότητας στο μοντέλο. Ένα στιγμιότυπο σχέσης (relation instance) είναι προφανώς ένα σύνολο πλειάδων.

Τα βασικά χαρακτηριστικά των σχέσεων συνοψίζονται στα εξής:

- Η διάταξη των γνωρισμάτων σε μία σχέση έχει σημασία, σε αντίθεση με την διάταξη των πλειάδων.
- Κάθε πλειάδα αποθηκεύεται **μία** φορά σε μία σχέση.
- Μια τιμή μπορεί να παρουσιάζεται πολλές φορές σε μία στήλη και είναι ατομική.
- Υπάρχει μια ειδική τιμή το κενό (null) και χρησιμοποιείται για να παραστήσει στη βάση μια τιμή που είναι μη-εφαρμόσιμη (non-applicable) ή άγνωστη (unknown).

Υπάρχουν διάφορα είδη περιορισμών, όπως:

- ❖ Δομικοί περιορισμοί: κλειδιού, ακεραιότητας οντότητας και αναφορικής ακεραιότητας. Ανάλογα το εμπορικό πακέτο που χρησιμοποιείται υπάρχει και αντίστοιχη υλοποίησή τους στη βάση.
- ❖ Περιορισμοί πεδίου τιμών.
- ❖ Περιορισμοί στηλών, οι οποίοι αναφέρονται στις τιμές των γνωρισμάτων.
- ❖ User-defined περιορισμοί, όπου μια σειρά μηχανισμών χρησιμοποιούνται για την υποστήριξη τέτοιων κανόνων σε ένα σχεσιακό σύστημα, όπως stored procedures, triggers, methods (για object-oriented συστήματα).

Παράδειγμα από το σχεσιακό μοντέλο του συστήματος εγγραφών παρατίθεται στο παρακάτω σχήμα:

Contents of "SCOTT"."VIEWS" - Microsoft Internet Explorer

Database: PHIVOSDB  
Date: 7 Σεπτέμβριος 2001 3:07:32 μμ

KAT	AA	Y	L	H	D	T	S	Z	E	TITLOS	HMER	KODIKOS	CURRDATE	AM	EKS_EGGR	EPONYMO	ONOMA	ONOM_PATROS	ODOS	NUM	PER	TK	THL
01	1	O	O	O								6456											
01	1	O	O	O																			
01	1	O	O	O																			
01	1	O																					
01		O	O	M																			
01	1	O	O	O																			
01	1	O	O	O																			
01	1	O	O	M							28-Mai - 2001 12:00:00 AM												
01	1	O	O	M					M		24-Mai - 2001 12:00:00 AM												
01	1	O		M		O		M			31-Mai - 2001 12:00:00 AM												
01	1	O	O	M	M						21-Mai - 2001 12:00:00 AM						ΦΟΙΒΟΣ						
01	1	O	O	O	O	M	M	O	M		24-Mai - 2001 12:00:00 AM												
01	1	O	O	O	O			M			31-Mai - 2001 12:00:00 AM												
											22-Mai -												

Εικόνα 2.9.5.α



### 2.9.6 Σύνοψη

Συμπερασματικά, αν θελήσουμε να αναφέρουμε τα χαρακτηριστικά εκείνα που έχουν μετατρέψει πλέον τις βάσεις δεδομένων και τα DBMS σε αναπόσπαστο τμήμα των υπολογιστικών εφαρμογών και όχι μόνο, θα καταλήγαμε στα εξής σημεία:

- ☆ Σε σχέση με τις παραδοσιακές εφαρμογές επεξεργασίας αρχείων οι βάσεις δεδομένων έχουν να αντιπαρατάξουν την αφαίρεση δεδομένων, την ιδιαίτερα σημαντική ανεξαρτησία προγραμμάτων-δεδομένων και προγραμμάτων-πράξεων, την υποστήριξη όψεων πολλαπλών χρηστών και το διαμοιρασμό των δεδομένων μεταξύ πολλαπλών δοσοληψιών.
- ☆ Στον τομέα της διαχείρισης, του σχεδιασμού και της χρήσης μιας βάσης δεδομένων, τα DBMS παρέχουν πολύπλοκους μηχανισμούς ελέγχου των δεδομένων, περιορισμούς μη εξουσιοδοτημένης προσπέλασης, δυνατότητες για εξαγωγή συμπερασμάτων μέσω επαγωγικών κανόνων, παροχή πολλαπλών interfaces, αναπαράσταση πολύπλοκων συσχετίσεων μεταξύ των δεδομένων, επιβολή περιορισμών ορθότητας και τέλος την πολύ σημαντική παροχή αντιγράφων και δυνατότητας recovery.
- ☆ Περαιτέρω πλεονεκτήματα θεωρούνται οι δυνατότητες για την επιβολή προτύπων, η ελάττωση του χρόνου ανάπτυξης των εφαρμογών και η διαθεσιμότητα ενημερωμένων πληροφοριών σε όλους τους χρήστες.

## 2.10 Structured Query Language (SQL):

### 2.10.1 Ορισμός:

Η Structured Query Language (SQL) αποτελεί τη standard διαδραστική γλώσσα στις σχεσιακές βάσεις δεδομένων. Είναι επίσημη σύσταση τόσο του American National Standards Institute (ANSI), όσο και του International Organization for Standardization (ISO) [L14] και πρωτοέκανε την εμφάνιση της ως Sequel 75 στο System-R της IBM τη δεκαετία του 1970. Τελευταίο κοινά αποδεκτό και επίσημο standard της γλώσσας είναι η SQL/92, ενώ και η SQL-3 με τις object-oriented προεκτάσεις της βρίσκεται πολύ κοντά σε τελική αναγνώριση ως πρότυπο. Παρόλα αυτά, πολλά εμπορικά DBMS υποστηρίζουν και κάποιες επιπλέον προεκτάσεις της SQL, όπως την PL/SQL η Oracle, η οποία επιτρέπει την ανάμιξη εντολών SQL με procedural constructs.

### 2.10.2 Γενικά:

Η SQL είναι ειδική στον ορισμό και τη διαχείριση δομημένων δεδομένων μέσω ενός υψηλού επιπέδου interface, που επιτρέπει την επερώτηση και την ανανέωσή τους. Επίσης παρέχει τη δυνατότητα διαχείρισης του σχήματος της βάσης και των τυχόν διαφορετικών όψεων για την υποστήριξη πολλαπλών χρηστών, μέσα από ένα πλούσιο μοντέλο meta-πληροφορίας. Διαχειρίζεται πολύ καλά την ταυτόχρονη πρόσβαση χρηστών με διαφορετικά δικαιώματα προσπέλασης και διαχείρισης των δεδομένων, ενώ παρέχει και περιορισμούς ακεραιότητας. Στηρίζεται σε ένα καλά θεμελιωμένο

μαθηματικό υπόβαθρο, το οποίο βασίζεται στον κατηγορικό λογισμό πρώτης τάξης (Relational Calculus).

Ουσιαστικά η SQL χωρίζεται σε δύο βασικά μέρη, το DDL και το DML:

- Data Definition Language (DDL): Αποτελεί τη γλώσσα ορισμού της SQL και χρησιμοποιείται για create, drop και alter τις περιγραφές των σχέσεων στη βάση δεδομένων.
- Data Manipulation Language (DML): Παρέχονται δυνατότητες βασισμένες στη σχεσιακή άλγεβρα, καθώς και δυνατότητες εισαγωγής και διαγραφής στη βάση.

Υπάρχουν τέσσερις βασικές εντολές:

- ▶ **select**
- ▶ **insert**
- ▶ **update**
- ▶ **delete,**

το δε αποτέλεσμα μιας εντολής ή πράξης σε σχέσεις είναι πάντοτε μία καινούρια σχέση. Χρήσιμες λειτουργίες της SQL είναι τα nested queries, η διάταξη των αποτελεσμάτων με βάση την επιθυμία του χρήστη, οι λειτουργίες συνόλων (set operations), τα aggregate functions (min, max, avg, count, sum) και τα joined relations. Σημαντική διαφορά από την παραπάνω παρουσίαση περί βάσεων αποτελεί το γεγονός ότι στην SQL οι πίνακες δεν είναι σχέσεις, με άλλα λόγια επιτρέπουν την ύπαρξη διπλές πλειάδες.

Παράδειγμα χρήσης της SQL στη βάση του συστήματος εγγραφών είναι π.χ.:

```
SELECT StM.K_M,  
       StM.ROWID AS ROW_ID1,  
       StM.TITLOS,  
       StM.VR_UPAR,  
       StM.AK,  
       Ypcourse.OMADA,  
       Ypcourse.ROWID AS ROW_ID2,  
       Pltab.PLTEXT  
FROM ST_M StM, YPCOURSE Ypcourse, OMADAEPI Omadaepi, PLTAB Pltab  
where Pltab.PLCODE = Ypcourse.OMADA and StM.AK = 2001 and StM.K_TM=3 and StM.AK =  
Ypcourse.AK and Ypcourse.KAT='05' and Ypcourse.EJ='09' and StM.K_M = Ypcourse.K_M and  
Ypcourse.OMADA IN  
( '8001','8002','8003','8004','8005','8006','8007','8008','8009','8010','8011','8012','  
8113','2015','8111','8112')  
union  
SELECT StM.K_M,  
       StM.ROWID AS ROW_ID1,  
       StM.TITLOS,  
       StM.VR_UPAR,  
       StM.AK,  
       Omadaepi.OMADAEPI,  
       Omadaepi.ROWID AS ROW_ID3,  
       Pltab.PLTEXT  
FROM ST_M StM, YPCOURSE Ypcourse, OMADAEPI Omadaepi, PLTAB Pltab  
where Omadaepi.OMADAEPI = Pltab.PLCODE and StM.AK = 2001 and StM.K_TM=3 and StM.AK =  
Omadaepi.AK and Omadaepi.KAT='05' and Omadaepi.EJ='09' and StM.K_M = Omadaepi.K_M and  
Omadaepi.OMADAEPI IN  
( '8001','8002','8003','8004','8005','8006','8007','8008','8009','8010','8011','8012','  
8113','2015','8111','8112')  
order by 6,3
```

Συχνά, εντολές DML εμφυτεύονται σε προγράμματα μιας άλλης γλώσσας προγραμματισμού (host language). Άλλοτε πάλι αντί να αλλάζει ο compiler, όπως παραπάνω, προστίθεται μια βιβλιοθήκη με κλήσεις στη βάση δεδομένων API (Application Programming Interface). Το **ODBC** της Microsoft έχει γίνει το C/C++ standard στα Windows, το **JDBC** της SUN είναι το ανάλογο για τη Java. Η διαδικασία που πραγματοποιείται ανεξάρτητα του DBMS που χρησιμοποιείται είναι ένας οδηγός να παγιδεύει τις κλήσεις και να τις μεταφράζει σε κώδικα για το DBMS, ενώ η βάση μπορεί να βρίσκεται στο δίκτυο.

Παράδειγμα τέτοιου API από το σύστημα εγγραφών δίνεται στο παρακάτω απόσπασμα κώδικα από την Java κλάση για το user validation του υποσυστήματος της γραμματείας:

```
public void userdatabase(String username, String password) {

    System.out.println("for debugging reasons only! No 1\n");
    String url = "jdbc:oracle:thin:@127.0.0.1:1521:phivosdb";
    Connection con;
    Statement stmt;
    column1=null;
    column2=null;

    String selectString;

    System.out.println(username);
    System.out.println(password);

    selectString = "SELECT * FROM TESTV WHERE USERID='"
+username+ "' AND PASSWORD='"+ password+ "' " ;

    System.out.println(selectString);

    System.out.println("for debugging reasons only! No 2\n");

    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        System.out.println("\nFor debugging reasons only! No 3\n");
    }

    catch(java.lang.ClassNotFoundException e) {
        System.err.print("ClassNotFoundException: ");
        System.err.println(e.getMessage());
    }

    try {
        System.out.println("\nFor debugging reasons only! No
4\n");

        con = DriverManager.getConnection(url, "scott", "tiger");
        stmt = con.createStatement();

        System.out.println("\nFor debugging reasons only! No
5\n");

        ResultSet rs = stmt.executeQuery(selectString);
        while(rs.next()) {
```

```

        column1 = rs.getString(1);
        column2 = rs.getString(2);
        System.out.println(column1 + "    "+column2);
        /* for debugging reasons only! */

        count++;
    }

    count--;
    stmt.close();
    con.close();

    System.out.println("\nFor debugging reasons only!    No
6\n");
}

catch(SQLException ex) {

    System.err.println("SQLException: " + ex.getMessage());
}

}

```

## 2.11 Oracle:

### 2.11.1 Γενικά:

Στην παγκόσμια αγορά βάσεων δεδομένων η κυρίαρχη εταιρία σήμερα είναι η Oracle, κατέχοντας το 33% και αποτελώντας την πρώτη επιλογή των ανθρώπων στο χώρο των βάσεων δεδομένων όταν ακούν τον όρο relational database. Σημαντικοί σταθμοί στην ιστορία της εταιρίας [L15] αποτέλεσαν:

- 1977: Ίδρυση της εταιρίας από τον Lawrence J. Ellison.
- 1979: Προσφέρει το πρώτο εμπορικό σχεσιακό DBMS.
- 1995: Προσφέρει το πρώτο 64-bit RDBMS και παρέχει δυνατότητα σύνδεσης στο web μέσω ενός kit.
- 1997: Παρέχει την πρώτη web database.
- 1999: Παρέχει το πρώτο RDBMS για το λειτουργικό σύστημα Linux.

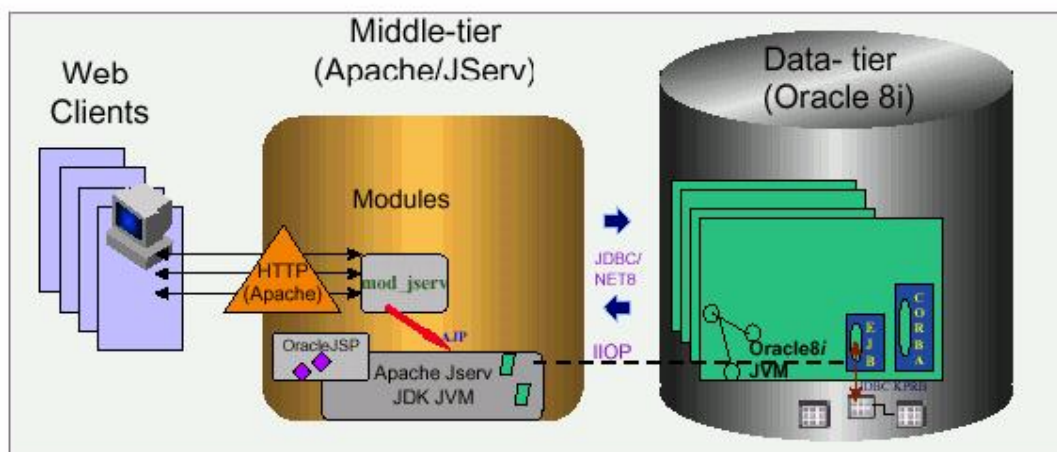
Αποτελεί την πρώτη εταιρία λογισμικού παγκοσμίως που ανέπτυξε και εξέλιξε 100% Internet-enabled enterprise software σε όλη τη γκάμα των προϊόντων της (database, server, enterprise business applications, και application development and decision support tools). Διαθέτει πάνω από 43000 υπαλλήλους σε παγκόσμιο επίπεδο, πραγματοποιεί πωλήσεις σε 150 χώρες και παρέχει δωρεάν όλα τα προϊόντα της στο Internet προς download.

### 2.11.2 Oracle8i:

Ουσιαστικά η Oracle8i ξεφεύγει από τον αυστηρό θεωρητικό ορισμό μιας βάσης δεδομένων, προσφέροντας αρκετές επιπλέον δυνατότητες στον προγραμματιστή. Κατά βάση αποτελεί μία αντικειμενο-σχεσιακή βάση δεδομένων (**object-relational database**), διαθέτει τη δική της Java Virtual Machine (**8iJVM**) με πλήρη υποστήριξη J2EE (**servlets, JSPs, Enterprise JavaBeans**) και επιτρέπει στα αποθηκευμένα προγράμματα (stored programs) και σκανδάλες (triggers) να είναι γραμμένα σε Java. Η session-based αρχιτεκτονική της Oracle8iJVM επιτρέπει σε κάθε session χρήστη να έχει τη δική του JVM με τις δικές του Java μεταβλητές και collectors «σκουπιδιών», αυξάνοντας έτσι την επεκτασιμότητα του συστήματος σε εκατοντάδες χρήστες και ελαττώνοντας σημαντικά τα φορτία μνήμης και σύνδεσης.

Η τελευταία έκδοσή της μάλιστα, **Oracle8i release 3 (8.1.7)**, περιλαμβάνει επιπλέον την Oracle Servlet Engine (**OSE**) δίνοντας την δυνατότητα στην Oracle να επεξεργάζεται εκτός από τις κλασικές IIOP και Net8 συνδέσεις απευθείας και http requests, ενώ συμμορφώνεται πλήρως και με τις προδιαγραφές Servlet 2.2. Επίσης υλοποιεί πλήρως το **JDBC** standard για Java Database Connectivity, παρέχει extra δυνατότητες για γλώσσες πέρα από τα Αγγλικά με το National Language Support (**NLS**) και διαθέτει πληθώρα **PL/SQL** packages προς χρήση π.χ. με το JDBC.

Είναι δυνατόν να υλοποιηθούν διάφοροι συνδυασμοί για ανάπτυξη web applications βασισμένες στον Oracle8i Server. Όπως θα αναλυθεί διεξοδικά παρακάτω, στη συγκεκριμένη περίπτωση επιλέξαμε τη λύση ενός ενδιάμεσου Apache web server με Tomcat 3.2 για κατάλληλη υποστήριξη JSPs και σε γενικές γραμμές το setup του συστήματος έχει ως εξής:



Εικόνα 2.1.12.α

Επιγραμματικά, τα πλεονεκτήματα και μειονεκτήματα μιας τέτοιας αρχιτεκτονικής συνοψίζονται στα εξής:

Πλεονεκτήματα:

- τα στατικά αρχεία του web application επεξεργάζονται γρηγορότερα περιοριζόμενα στον Apache web server.
- Το module **mod\_jserv** του Apache μπορεί να μοιράσει requests σε πολλαπλές JVM για καλύτερη κατανομή φορτίου σε πρώτη φάση.
- Υπάρχει σαφής διαχωρισμός της βάσης δεδομένων από το υπόλοιπο σύστημα, παρόλο που η συνεργασία όλων των μερών είναι άρρηκτη μέσω κυρίως Java tools.

#### Μειονεκτήματα:

- Η δυνατότητα για πολλαπλές JVM στο μεσαίο επίπεδο πρέπει να γίνει σχολαστικά configured «με το χέρι».
- Αρχεία JSP που περιλαμβάνουν πολλαπλές SQL-queries εκτελούνται κατά κανόνα κάπως αργά.
- Ο Apache στο μεσαίο επίπεδο περιορίζει τις δυνατότητες για υποστήριξη Servlet 2.2 του OSE σε Servlet 2.0.

## **2.12 Java:**

### **2.12.1 Ορισμός**

Η Java [L16] αποτελεί μία ιδιαίτερα διαδεδομένη αντικειμενοστρεφή γλώσσα προγραμματισμού ειδικά σχεδιασμένη για χρήση στο κατανεμημένο περιβάλλον του Internet. Πέρα από αυτό θα μπορούσε να ισχυριστεί κανείς ότι διαμορφώνει μια ολόκληρη πλατφόρμα ανάπτυξης λογισμικού, βασισμένη στη δύναμη των δικτύων γενικότερα και με moto την αρχή: «Write Once, Run Anywhere™», η οποία συνοψίζει την ιδέα ότι το ίδιο software θα πρέπει να τρέχει σε διαφορετικά είδη υπολογιστών, λειτουργικών συστημάτων, palmtops, κινητών τηλεφώνων, τηλεοράσεων, gadgets κ.λ.π μηχανημάτων και συσκευών.

Δημιουργήθηκε από την εταιρία Sun Microsystems το 1995 και σχεδόν αμέσως κυριάρχησε στον κόσμο των δικτυωμένων υπολογιστών. Έχει το “look and feel” της γλώσσας C++, με τη διαφορά ότι είναι πιο εύκολη στην εκμάθηση, χωρίς αυτό να σημαίνει ότι μαθαίνεται και σε ελάχιστο χρόνο. Η Java μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών, τόσο σε επίπεδο ενός και μόνο υπολογιστή, όσο και σε επίπεδο πολλαπλών servers και clients σε ένα δίκτυο. Μπορεί επίσης να χρησιμοποιηθεί για την υλοποίηση μικρών modules εφαρμογών ή applets σε μία web σελίδα. Το τελευταίο χαρακτηριστικό επιτρέπει στον χρήστη να αλληλεπιδράσει με την web σελίδα σε επίπεδο client. Όλοι οι μεγάλοι browsers διαθέτουν πλέον τη δική τους JVM για τέτοιου είδους requests, ενώ όλοι οι μεγάλοι κατασκευαστές λειτουργικών συστημάτων διαθέτουν Java compiler στο λειτουργικό που παρέχουν.

### 2.12.2 Εισαγωγή

Αναμφισβήτητα η Java αποτελεί μια μικρή επανάσταση στο χώρο των γλωσσών προγραμματισμού. Ήρθε να καλύψει τις ανάγκες ανάπτυξης λογισμικού την εποχή της ραγδαίας εξάπλωσης του Internet και του www. Από τα βασικά χαρακτηριστικά της είναι:

- ☆ **Αντικειμενοστρεφής:** Πρόκειται για την πλέον object-oriented γλώσσα προγραμματισμού, με ομοιότητες εντολών με τη C++, αλλά και πολλά διαφορετικά νέα στοιχεία. Ένα αντικείμενο<sup>3</sup> είναι σε θέση να εκμεταλλευτεί το γεγονός ότι ανήκει σε μία κλάση και να κληρονομήσει χαρακτηριστικά και ιδιότητες, με τη μορφή κώδικα πάντα, κοινά για ολόκληρη την κλάση. Τη διέπουν όλες οι βασικές αρχές του αντικειμενοστρεφούς προγραμματισμού, όπως:
  - **ιεραρχία τύπων/κλάσεων, κληρονομικότητα:** Η κληρονομικότητα βασίζεται, όπως αναφέρθηκε, στην ιδέα του να μπορούμε να δώσουμε εξειδικευμένα χαρακτηριστικά σε κάποια αντικείμενα, χωρίς τα τελευταία να χάνουν τις γενικότερες ιδιότητες που θα πρέπει να τα διακρίνουν. Αν θεωρήσουμε, λοιπόν, ένα δένδρο κλάσεων, κάθε κλάση κληρονομεί από την πατρική της όλες τις instance variables και τις μεθόδους της. Σε αυτές, η θυγατρική υποκλάση είναι δυνατόν να προσθέσει τις δικές της, εξειδικεύοντας έτσι την πατρική υπερκλάση.
  - **πολλαπλή κληρονομικότητα:** Η περίπτωση όπου μία υποκλάση κληρονομεί λειτουργίες και χαρακτηριστικά από περισσότερες της μίας υπερκλάσεις.
  - **πολυμορφισμός:** Πολυμορφισμός είναι το γεγονός ότι διάφορα αντικείμενα ή/και κλάσεις ολόκληρες αντιδρούν διαφορετικά στο ίδιο μήνυμα.
  - **υπέρβαση:** Είναι η περίπτωση όπου κάποιο μήνυμα υλοποιείται με διαφορετική μέθοδο στην κορυφή της ιεραρχίας από ότι σε κάποιο χαμηλότερο επίπεδο.
  - **υπερφόρτωση:** Δύο μηνύματα σε διαφορετικές κλάσεις συμφωνούν στο interface τους.

#### ☆ Δημιουργία ανεξάρτητων εφαρμογών και applets

- ☆ **Interpreted γλώσσα:** Τα προγράμματα που παράγονται είναι portable στο δίκτυο. Κι αυτό γιατί ο μεταγλωττιστής της Java δεν παράγει απ' ευθείας εκτελέσιμο κώδικα από τον πηγαίο, αλλά μια μορφή ενδιάμεσου ψευδοκώδικα (bytecode), ο οποίος έχει πολύ μικρό μέγεθος (της τάξεως κάποιων Kbytes) και για να τρέξει απαιτεί την ύπαρξη μιας JVM, ή αλλιώς ενός interpreter, που

---

<sup>3</sup> Ένα αντικείμενο είναι μια οντότητα στη μνήμη η οποία περιέχει δεδομένα και μεθόδους μέσω των οποίων μπορούμε να αλλάξουμε τα δεδομένα ή να επικοινωνήσουμε με το αντικείμενο. Ο κώδικας που ορίζει ένα αντικείμενο λέγεται κλάση του αντικείμενου αυτού. Η κλάση χρησιμοποιείται για την κατασκευή πανομοιότυπων αντικειμένων-αντιγράφων, ενώ τα τελευταία λέγονται στιγμιότυπα (instances) της κλάσης.

θα αναλάβει την μετατροπή του ενδιάμεσου ψευδοκώδικα σε εκτελέσιμο. Αυτό ακριβώς είναι και το ιδιαίτερο χαρακτηριστικό που επιτρέπει στα προγράμματα αυτά να τρέχουν σε οποιοδήποτε μηχάνημα διαθέτει μία JVM, ανεξάρτητα του αν αυτό είναι υπολογιστής ή τοστιέρα! Καταργείται λοιπόν έτσι η έννοια του platform-specific κώδικα.

- ☆ **Distributed:** Είναι δυνατόν διαφορετικά τμήματα ενός προγράμματος να συλλέγουν από διαφορετικά sites στο δίκτυο, δίνοντας έτσι μια κατανεμημένη διάσταση στον προγραμματισμό.
- ☆ **Robust:** Ο κώδικας της Java είναι «κλειστός» με τα αντικείμενα να μην μπορούν να θεωρήσουν references σε εξωτερικά σε αυτά ή άλλα αντικείμενα δεδομένα. Αυτό εξασφαλίζει ότι μια εντολή δεν μπορεί να περιέχει την διεύθυνση δεδομένων μιας άλλης εφαρμογής ή/και του ίδιου του λειτουργικού συστήματος, πράγμα που θα εξανάγκαζε την εφαρμογή ή το λειτουργικό να τερματιστεί απότομα (crash). Η JVM πραγματοποιεί έναν αριθμό ελέγχων σε κάθε αντικείμενο προκειμένου να εξασφαλιστεί η ακεραιότητα των προγραμμάτων.
- ☆ **Ασφάλεια:** Είναι σχεδιασμένη έτσι ώστε να ελαχιστοποιείται η πιθανότητα προσβολής του συστήματος του χρήστη από κάποιο κακόβουλο applet.
- ☆ **multithreaded:** Υποστηρίζει εγγενώς τη χρήση πολλαπλών threads, ακόμα και σε συστήματα με έναν επεξεργαστή. Προκειμένου να το πετύχει αυτό, η Java υλοποιεί το δικό της χρονοδρομολογητή (scheduler), ο οποίος αναλαμβάνει το multithreading με πλήρη διαφάνεια προς το χρήστη.
- ☆ **multimedia εφαρμογές:** Τέλος, η Java είναι η νούμερο ένα γλώσσα στην υλοποίηση multimedia εφαρμογών, τόσο σε desktop επίπεδο, όσο και σε επίπεδο δικτύου, λόγω κυρίως των ευκολιών που παρέχει σε αυτό τον τομέα στους προγραμματιστές, με τις ιδιαίτερα πλούσιες βιβλιοθήκες και την ευελιξία της.

### 2.12.3 Εργαλεία

Τα κυριότερα εργαλεία-προγράμματα του Java Development Kit (JDK), έκδοση 1.3 είναι:

- **javac:** ο Java compiler.
- **java:** ο Java interpreter.
- **javaw:** like java, αλλά για το λειτουργικό σύστημα Windows μόνο.
- **jdb:** ο Java debugger
- **javap:** ο Java disassembler
- **javadoc:** πρόγραμμα για αυτόματη κατασκευή documentation για εφαρμογές.



## 2.12.4 Περιγραφή

### ❖ κλάση

Ένα τυπικό παράδειγμα ορισμού κλάσης από το σύστημά μας είναι:

```
public class myRowSetBrowser extends DataWebBeanImpl {}
```

Γενικά, η δήλωση των πεδίων/μεθόδων στην κλάση μπορεί να ακολουθεί όποια σειρά θέλει ο προγραμματιστής, ενώ μπορεί να γίνει και χρήση τους πριν ακόμα δηλωθούν.

### ❖ *access specifiers*

Υπάρχουν τέσσερις διαφορετικές δηλώσεις access specifiers:

- **public**, που σημαίνει ότι το συγκεκριμένο πεδίο ή μέθοδος μπορεί να χρησιμοποιηθεί από οποιονδήποτε, ακόμα και από ξένο αντικείμενο.
- **protected**, που σημαίνει ότι το πεδίο/μέθοδος που το έχει είναι ορατό μόνο στις μεθόδους της κλάσης αυτής και των υποκλάσεων της.
- **private**, που σημαίνει ότι το πεδίο/μέθοδος είναι ορατό μόνο στις μεθόδους αυτής της κλάσης.
- **τίποτα**, που σημαίνει ότι το πεδίο/μέθοδος είναι public, αλλά μόνο για το συγκεκριμένο package.

### ❖ *modifiers*

Είναι δεσμευμένες λέξεις της Java που προσδίδουν συγκεκριμένες επιπλέον ιδιότητες ή/και χαρακτηρίζουν τα πεδία/μεθόδους που τα έχουν. Οι τρεις πιο σημαντικοί είναι:

- **final**: ορίζει ότι το πεδίο είναι σταθερά και όχι μεταβλητή, για τις μεθόδους ορίζει ότι απαγορεύεται το override, ενώ για τις κλάσεις ορίζει ότι δεν επιτρέπεται subclassing.
- **static**: το πεδίο/μέθοδος που το έχει είναι μοναδικό για όλο το στιγμιοτύπο της κλάσης.
- **synchronized**: για μεθόδους σημαίνει ότι το αντικείμενο αυτό είναι threadsafe.



```

try {
    Class.forName("oracle.jdbc.driver.OracleDriver");
}

catch (java.lang.ClassNotFoundException e) {
    System.err.print("ClassNotFoundException: ");
    System.err.println(e.getMessage());
}

```

### 2.12.5 Java - Javascript

Μία κοινή, τέλος, παρανόηση που γίνεται στο χώρο των γλωσσών προγραμματισμού είναι η σύγχυση της Java της Sun Microsystems με τη Javascript της Netscape. Εκτός όμως από το πρώτο συνθετικό, οι δύο γλώσσες παρουσιάζουν σημαντικότερες διαφορές. Η Javascript επεξεργάζεται σε υψηλότερο επίπεδο, είναι τρομακτικά ευκολότερη στην εκμάθηση και φυσικά ουδεμία σχέση έχει με αντικειμενοστραφή προγραμματισμό, αποτελώντας μία ιδιαίτερα εύχρηστη scripting γλώσσα.

Ακόμα, *μόνο* η Java μπορεί να χρησιμοποιηθεί για την ανάπτυξη standalone εφαρμογών, καθώς και mini-εφαρμογών, των applets. Η Javascript είναι ιδιαίτερα διαδεδομένη για τη διάδραση με τους χρήστες ενός web site και στην συντριπτική της πλειοψηφία βρίσκεται ενσωματωμένη μέσα σε σελίδες html.

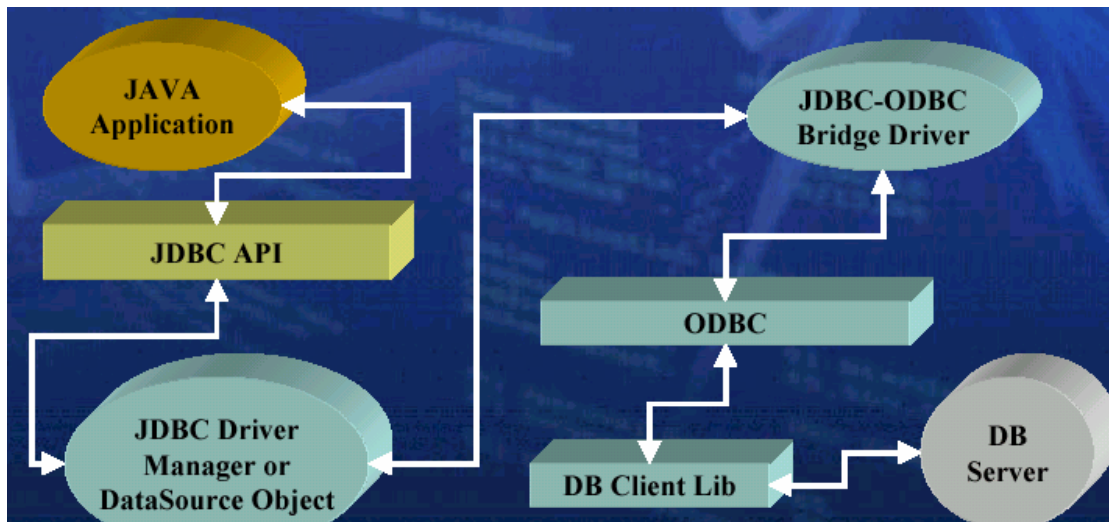
## 2.13 Java Database Connectivity (JDBC)

Το **Java Database Connectivity (JDBC)** είναι μία προδιαγραφή Application Program Interface (**API**) για τη διασύνδεση προγραμμάτων γραμμένα σε Java με τα δεδομένα σε ορισμένες δημοφιλείς σχεσιακές βάσεις δεδομένων, όπως οι Oracle, MS SQL Server, DB2, Informix, Sybase, Interbase, MySQL. Ουσιαστικά προσφέρει τρεις δυνατότητες:

- 1) Διασύνδεση με βάσεις δεδομένων
- 2) Αποστολή SQL εντολών
- 3) Επεξεργασία των αποτελεσμάτων

Καθορίστηκε αρχικά από την Sun Microsystems επιτρέποντας σε ανεξάρτητους κατασκευαστές να επεκτείνουν την υλοποίηση προσθέτοντας τα δικά τους χαρακτηριστικά και drivers. Βασίζεται στο X/Open SQL Call Level Interface και σε συμφωνία με το SQL/92 Entry Level standard.

Είναι παρόμοιο με το Open Database Connectivity (**ODBC**) και με τη χρήση κατάλληλου προγράμματος «γέφυρας» είναι δυνατή η χρήση του JDBC για την προσπέλαση βάσεων δεδομένων μέσω του ODBC interface, με την προϋπόθεση ότι σε κάθε client πρέπει να υπάρχει ODBC binary code.



Εικόνα 2.13.α

Με το JDBC είναι δυνατή η προσπέλαση προγραμματιστικά βάσεων δεδομένων σε δίκτυο και πάνω από το Internet. Παράδειγμα τέτοιας χρήσης αποτελεί και το ακόλουθο κομμάτι κώδικα από ένα αρχείο του συστήματος εγγραφών:

```
String url = "jdbc:oracle:thin:@127.0.0.1:1521:phivosdb";
Connection con;
Statement stmt;

.
.
.

con = DriverManager.getConnection(url, "scott", "tiger");
stmt = con.createStatement();
```

Το JDBC καθορίζει ένα σύνολο από αντικειμενοστραφείς κλάσεις που μπορεί να χρησιμοποιήσει ο προγραμματιστής προκειμένου να κατασκευάσει εντολές SQL για σύνδεση στη βάση δεδομένων. Ένα επιπλέον σύνολο από κλάσεις περιγράφει το JDBC driver API. Υποστηρίζονται οι πιο κοινοί τύποι δεδομένων της SQL, με ανάλογη αντιστοίχσή τους σε τύπους δεδομένων της Java. Ακολουθεί κομμάτι κώδικα προς κατανόηση του API:

```
Connection con = // connect
    DriverManager.getConnection(url, "login", "pass"); Statement stmt
= con.createStatement(); // set up stmt
String query = "SELECT COF_NAME, PRICE FROM COFFEES";
ResultSet rs = stmt.executeQuery(query);
try { // handle exceptions
    // loop through result tuples
    while (rs.next()) {
        String s = rs.getString("COF_NAME");
        Float n = rs.getFloat("PRICE");
        System.out.println(s + "    " + n);
    }
}
```

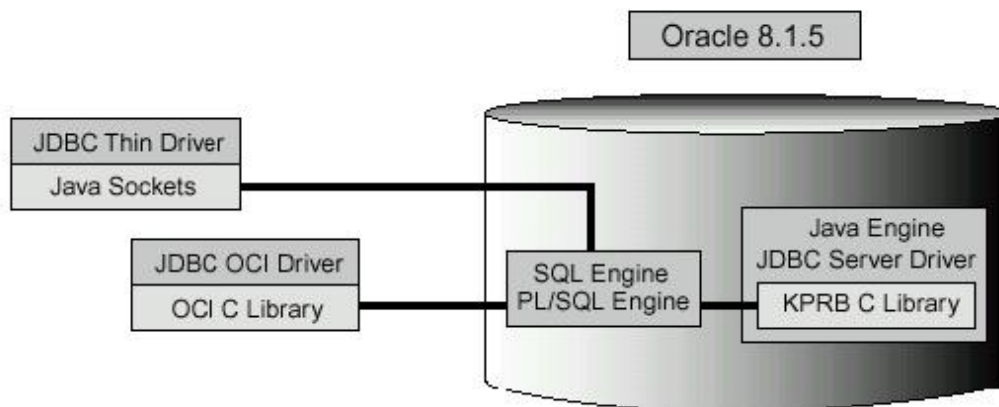
```

} catch(SQLException ex) {
    System.out.println(ex.getMessage ()
        + ex.getSQLState () + ex.getErrorCode ());
}

```

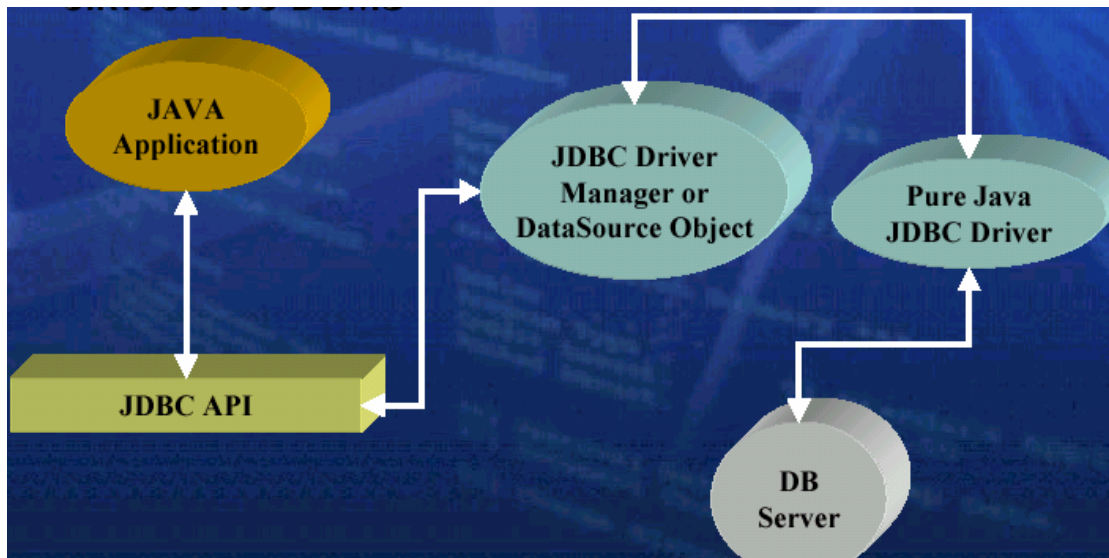
Ο τρόπος πρόσβασης στη βάση είναι ανεξάρτητος υλοποίησης και απαιτεί την ύπαρξη κατάλληλου JDBC-driver. Ανάλογα τη βάση που χρησιμοποιείται υπάρχουν και οι κατάλληλοι drivers. Συγκεκριμένα για την Oracle8i που χρησιμοποιούμε στο σύστημά μας, υπάρχουν οι εξής drivers:

- JDBC Thin Client-Side Driver
- JDBC OCI Client-Side Driver
- JDBC Server Driver



*Εικόνα 2.13.β*

Επιλέχθηκε ο πρώτος από αυτούς για λόγους απλότητας στο configuration, στο setup και στη χρήση. Είναι γραμμένος 100% σε Java και συμφωνεί με το JDBC 1.22 standard, ενώ δεν απαιτείται καμία εγκατάσταση Oracle-specific λογισμικού στον client προκειμένου να δουλέψει. Είναι τύπου 4 και η λογική που ακολουθεί φαίνεται στο παρακάτω σχήμα:



Εικόνα 2.13.γ

## 2.14 Java Server Pages (JSP):

### 2.14.1 Ορισμός

Ο όρος **Java Server Page (JSP)** αναφέρεται σε μια τεχνολογία στα πλαίσια της πλατφόρμας J2EE, για την παραγωγή εφαρμογών με δυναμικό περιεχόμενο και για τον έλεγχο του περιεχομένου ή/και της εμφάνισης web σελίδων μέσω της χρήσης **servlets**. Τα servlets είναι μικρά προγράμματα που καθορίζονται στην web σελίδα και τρέχουν στον web server, προκειμένου να αλλάξουν το περιεχόμενό της προτού αυτό σταλεί στον client-χρήστη. Αναπτύχθηκαν από τη **Sun Microsystems** (με την πρώτη προδιαγραφή JSP 1.0 να βγαίνει δημόσια τον Ιούνιο του 1999), την ίδια εταιρία που ανέπτυξε και την γλώσσα προγραμματισμού Java, η οποία και παρείχε ελεύθερα τις προδιαγραφές τους (τελευταία έκδοση: **JSP version 1.2** [L26],[L27],[L28]) και παρουσιάζουν αρκετές ομοιότητες με την τεχνολογία **Active Server Pages (ASP)** της Microsoft, παρέχοντας όμως στους προγραμματιστές όλα τα πλεονεκτήματα της γλώσσας Java. Κι αυτό γιατί κάθε JSP καλεί στην ουσία ένα πρόγραμμα Java που εκτελείται στον web server πριν την αποστολή στον χρήστη, σε αντίθεση με τα ASPs που χρησιμοποιούν scripts και έναν script interpreter (π.χ.: VBScript ή Jscript) αντίστοιχα.

### 2.14.2 Γενικά

Η τεχνολογία των JSPs [L17]-[L20] επιτρέπει την ανάπτυξη και εύκολη συντήρηση, πλούσιων σε πληροφορία, δυναμικών web σελίδων, οι οποίες είναι ανεξάρτητες πλατφόρμας υλοποίησης (όπως άλλωστε ολόκληρη η οικογένεια της Java). Βασικό χαρακτηριστικό της αποτελεί ο διαχωρισμός της διαπροσωπείας χρήστη (user interface) από τη διαδικασία παραγωγής περιεχομένου, επιτρέποντας έτσι στον

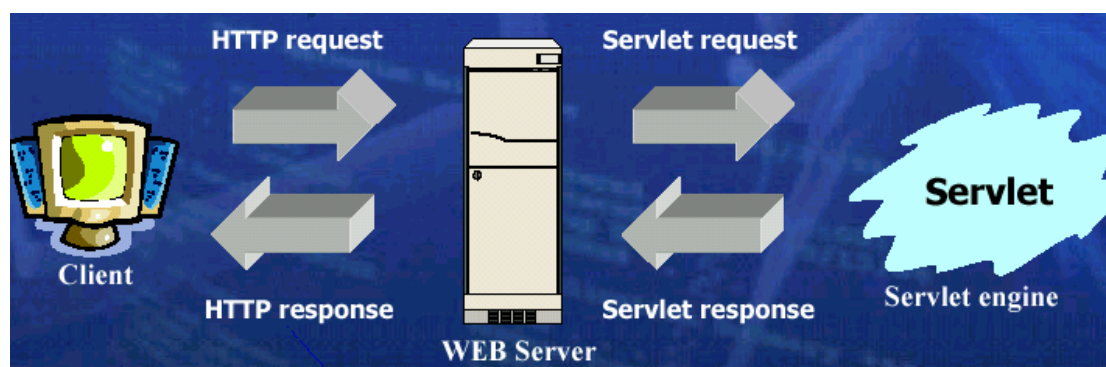
σχεδιαστή να αλλάζει ολόκληρη τη σελίδα, χωρίς να επηρεάζεται το δυναμικό περιεχόμενό της.

Η τεχνολογία JSP χρησιμοποιεί tags, όπως η XML και scriptlets γραμμένα σε Java προκειμένου να ενθυλακώσει τη λογική που παράγει το περιεχόμενο της web σελίδας. Επιπρόσθετα, η λογική της εφαρμογής μπορεί να βρίσκεται σε server-based resources, όπως τα JavaBeans, τις οποίες και προσπελαύνει η σελίδα μέσω των παραπάνω tags και scriptlets. Με το διαχωρισμό της λογικής της σελίδας από το σχεδιασμό και την παρουσίαση και υποστηρίζοντας επαναχρησιμοποιήσιμο component-based σχεδιασμό, τα JSPs αποτελούν μία τάχιστα και σχετικά εύκολη διαδικασία web-based εφαρμογών.

- *jsp-servlets:*

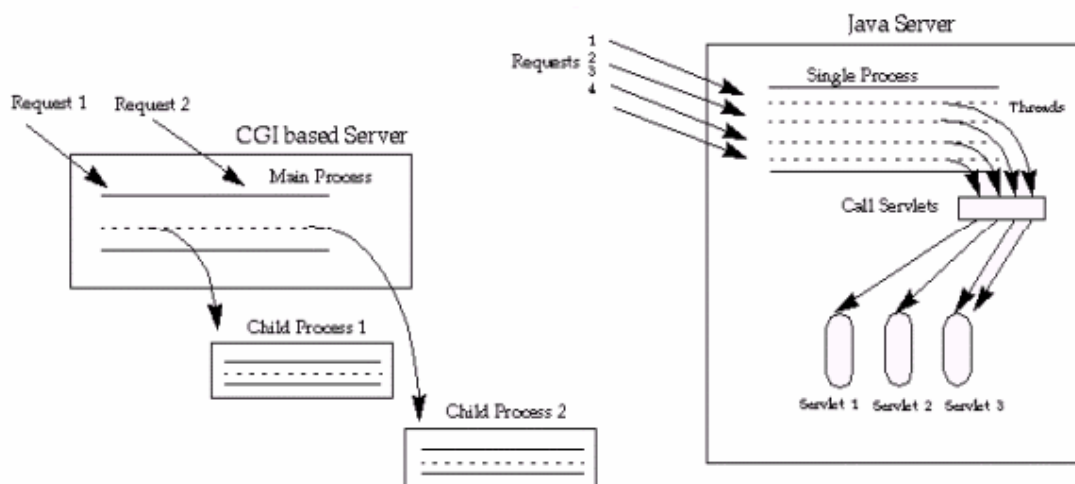
Η τεχνολογία JSP αποτελεί ουσιαστικά προέκταση της τεχνολογίας **Java Servlets**. Τα τελευταία είναι server-side components ανεξάρτητα πλατφόρμας και 100% γραμμένα σε Java, τα οποία επεκτείνουν τις δυνατότητες των Java enabled web servers με ελάχιστο overhead, κόστος συντήρησης και υποστήριξη. Φορτώνονται on demand σε εκείνο το μέρος του συστήματος που τα χρειάζεται και σε συνδυασμό με τα JSPs παρέχουν την καλύτερη ίσως εναλλακτική για δυναμικό προγραμματισμό σελίδων.

Ένα servlet είναι στην ουσία ένα πρόγραμμα Java που τρέχει σε έναν web server, δέχεται ένα http-request από τον browser, παράγει δυναμικό περιεχόμενο και παρέχει ένα http-response πίσω στον client. Παρέχει ασφαλή, βασισμένη στο δίκτυο και στο robust περιβάλλον της Java, πρόσβαση στα δεδομένα που παράγει δυναμικά και είναι αρκετά ευέλικτο.



Εικόνα 2.14.2.α

Προκάτοχός του θεωρείται το **Common Gateway Interface (CGI)**, το οποίο χρησιμοποιείται ακόμα και σήμερα για την δυναμική παραγωγή σελίδων. Η κύρια διαφορά τους είναι ότι όταν χρησιμοποιείται το CGI, κάθε νέο request «γεννά» μια καινούρια διεργασία, ενώ στην περίπτωση των servlets, κάθε request διαχειρίζεται από ένα νέο thread, με αποτέλεσμα τα servlets να ξεπερνάνε τα προβλήματα scalability που αντιμετωπίζουν τα CGI-based συστήματα. Ακόμη, σημαντικός προβληματισμός υπάρχει στη χρήση των cgi-scripts σε σχέση με παράγοντες ασφαλείας του συστήματος στο οποίο τρέχουν, θέματα τα οποία ξεπερνάνε με επιτυχία τα servlets.



Εικόνα 2.14.2.β

Παρόλα αυτά, υπάρχει σημαντική εννοιολογική διαφορά ανάμεσα στις JSPs και τα servlets. Συγκεκριμένα, ενώ στα servlets η έμφαση δίνεται στον κώδικα Java, στις JSPs η έμφαση δίνεται στην html ή XML. Στα πρώτα, μπορεί κανείς να συμπεριλάβει κώδικα html μέσα σε μια Java class στο output, ενώ στα JSPs ουσιαστικά ενσωματώνονται τμήματα κώδικα Java σε ένα html ή XML αρχείο. Θεωρητικά, λοιπόν, οι δύο αυτές προσεγγίσεις φαίνονται ισοδύναμες, αφού ο JSP translator μετατρέπει το JSP-style συντακτικό σε κώδικα Java και Java servlet-style συντακτικό, όμως σε επίπεδο ανάπτυξης web applications η επιλογή των JSPs είναι σαφώς πιο βολική, εύκολη και σαφέστερη.

### 2.14.3 Πρακτικά

Στα JSPs παρέχεται η δυνατότητα ανάμιξης στατικού με δυναμικό περιεχόμενο στις web σελίδες, ενώ ουσιαστικά τα JSPs μπορούν να θεωρηθούν ως προγράμματα Java ενσωματωμένα σε html κώδικα. Το JSP μοντέλο προγραμματισμού επιτρέπει την εκτέλεση, μέσω Java scriptlets, δηλώσεων και εκφράσεων, οι οποίες βρίσκονται αναμεμιγμένες με τον στατικό html κώδικα στην ίδια σελίδα. Για παράδειγμα, στην επόμενη JSP, Java scriptlets παίρνουν το όνομα του χρήστη από ένα απλό http request object (στην προκειμένη περίπτωση μια απλή φόρμα εισαγωγής) και την σημερινή ημερομηνία, και τα τυπώνουν ακριβώς από κάτω:

```
<%@ page language = "java" errorPage="errorpage.jsp" import =
"java.util.*, oracle.jbo.*, javax.naming.*, oracle.jdeveloper.html.*,
oracle.jbo.html.databeans.*" contentType="text/html; charset=ISO-8859-
7" %>
<html>
<head>
<title> Μία πλή JSP σελίδα </title>
</head>
<body>
```



```

<% String user=request.getParameter("user"); %>

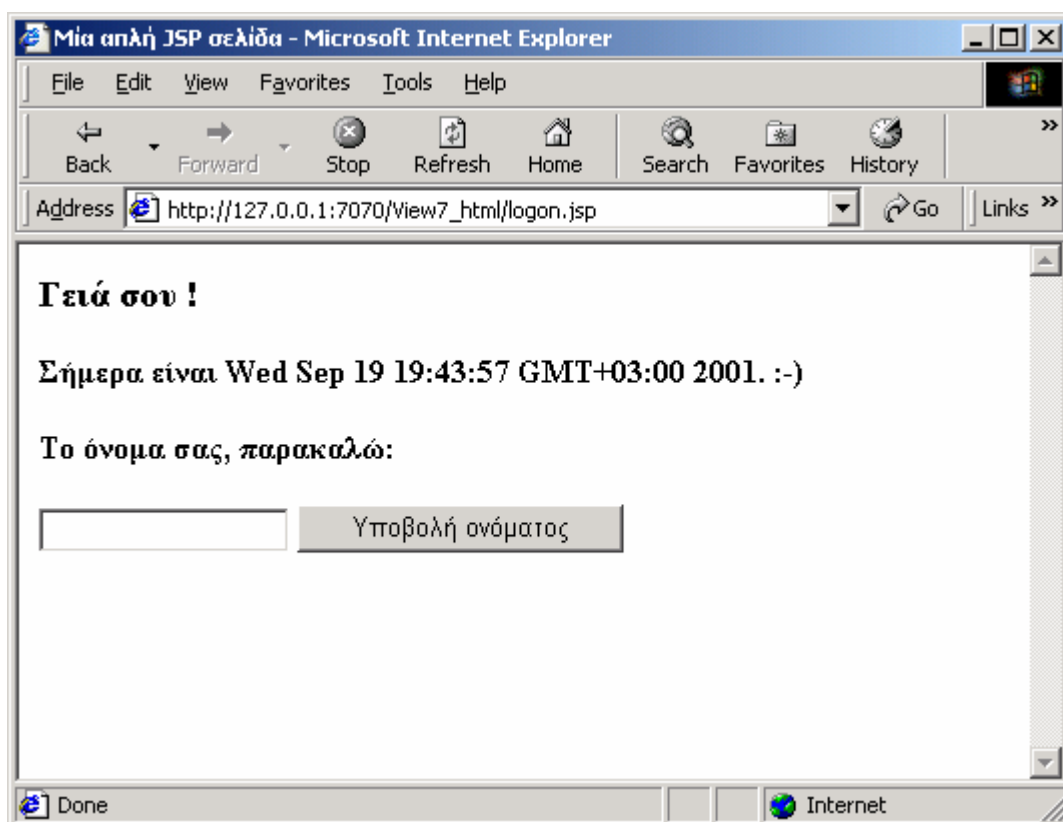
<h3> Γειά σου <%= (user==null) ? "" : user %>!</h3>

<p>
<b> Σήμερα είναι <%= new java.util.Date() %>. :-)</b>
</p>

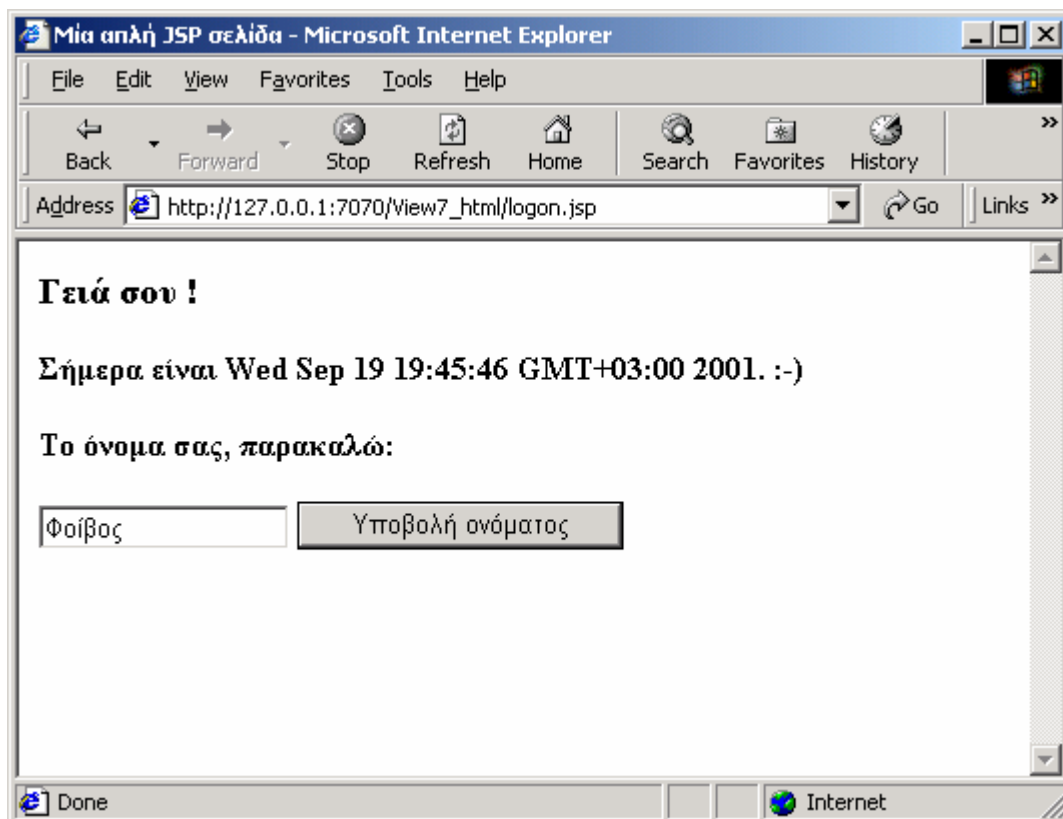
<b> Το όνομα σας, παρακαλώ:</b>

<form method="get">
<input type="text" name="user" size="15">
<input type="submit" value="Υποβολή ονόματος">
</form>
</body>
</html>

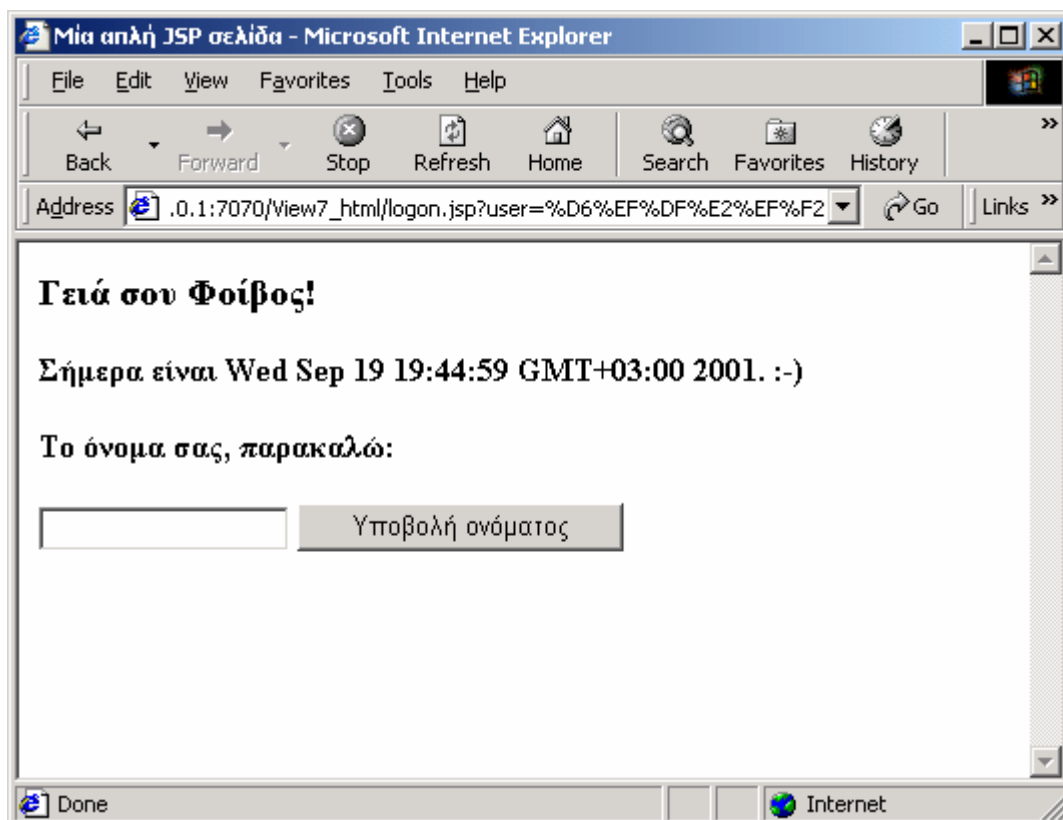
```



Εικόνα 2.14.3.α



Εικόνα 2.14.3.β



Εικόνα 2.14.3.γ

Τα Java scriptlets είναι τμήματα καθαρού κώδικα Java, ο οποίος τρέχει ως μέρος του servlet που παράγεται από τη μετάφραση του JSP. Παράδειγμα τέτοιας χρήσης είναι η περίπτωση όπου μία JSP πραγματοποιεί ένα query στη βάση δεδομένων και επιστρέφει τα αποτελέσματα σε έναν html πίνακα, μαζί με άλλο στατικό html περιεχόμενο:

```
<% while (acount>=0) { %>

<tr>

    <td><br></td>
    <td align="center"><jsp:getProperty    name="myntuacarsbean"
property="low_season_rate" /> <br></td>
    <td><br></td>
    <td align="center"><jsp:getProperty    name="myntuacarsbean"
property="high_season_rate" /> <br></td>
</tr>

<% myntuacarsbean.decrcount(); acount--; %>
<% } %>
```

Βέβαια, σε μία τυπική JSP σελίδα το μεγαλύτερο μέρος του κώδικα και της business logic<sup>4</sup> βρίσκεται ενθυλακωμένο σε κάποιο επαναχρησιμοποιήσιμο JavaBean, προκειμένου για καλύτερη απόδοση του όλου συστήματος. Αυτές ακριβώς οι δυνατότητες χρήσης επαναχρησιμοποιήσιμων JavaBeans components, καθώς και η υποστήριξη XML και XSL, κάνουν την τεχνολογία JSP ιδιαίτερα δυναμική και προσιτή στη χρήση. Στο παρακάτω γενικό παράδειγμα μιας σελίδας JSP χρησιμοποιούμε μία κλήση στο default JavaBean “**DBBean**” του OracleJSP για να προσπελάσουμε τη βάση δεδομένων του συστήματος και αν κάνουμε ένα απλό query, τεχνική που αποτελεί τυπικό σενάριο για μία εφαρμογή βάσης δεδομένων, ανεξάρτητα του χρησιμοποιούμενου JavaBean:

```
<%@ page import="java.sql.*" %>
<!-------
* This is a basic JavaServer Page that uses a DB Access Bean and queries
* dept and emp tables in schema scott and outputs the result in an html
* table.
* connStr can be set by the user. A sample connect string is
* jdbc:oracle:thin:@hostname:portnumber:ORCL817
*-----!>
<jsp:useBean id="dbbean" class="oracle.jsp.dbutil.DBBean" scope="session">
<jsp:setProperty name="dbbean" property="User" value="scott"/>
<jsp:setProperty name="dbbean" property="Password" value="tiger"/>
<jsp:setProperty name="dbbean" property="URL"
value="<%= (String)session.getValue(\"connStr\") %>" />
</jsp:useBean>
```

<sup>4</sup> Υπάρχει ένας βασικός διαχωρισμός των μερών ενός web application:

Data logic <--> Business logic <--> Presentation logic

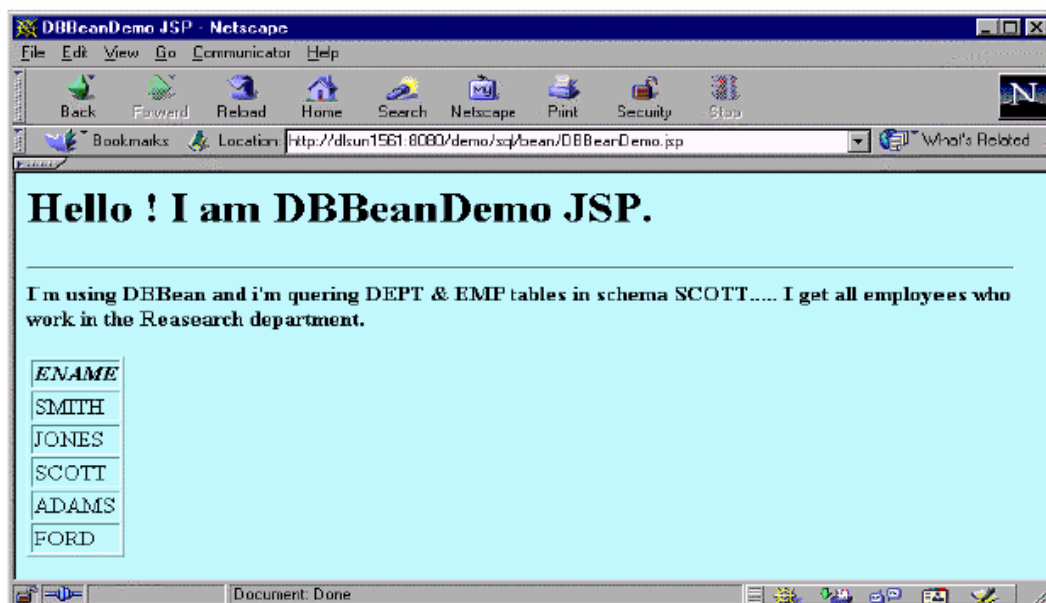
Με τον όρο **data logic** αναφερόμαστε στο καθαρό περιεχόμενο του web site. Ο τρόπος με τον οποίο θα γίνει ο διαχωρισμός του σε σελίδες, πώς θα γίνεται το navigation από τη μια σελίδα στην άλλη, τι θα παρουσιάζουμε σε κάθε σελίδα κ.λ.π. ονομάζεται **business logic**. Τέλος, **presentation logic** είναι η αισθητική παρουσίαση και look της σελίδας. Προφανώς, όσο πιο ξεκάθαρα μεταξύ τους είναι τα παραπάνω, τόσο πιο σαφές και ορθό προγραμματιστικά είναι η εφαρμογή μας.

```

<HTML>
<HEAD>
<TITLE>
DBBeanDemo JSP
</TITLE>
</HEAD>
<BODY BGCOLOR=EOFFFO>
<H1> Hello
<%= (request.getRemoteUser() != null? ", " + request.getRemoteUser() : "")
%>
! I am DBBeanDemo JSP.
</H1>
<HR>
<B> I'm using DBBean and querying DEPT & EMP tables in schema SCOTT.....
I get all employees who work in the Research department.
</B>
<P>
<%
try {
String sql_string = " select ENAME from EMP,DEPT " +
" where DEPT.DNAME = 'RESEARCH' " +
" and DEPT.DEPTNO = EMP.DEPTNO";
// Make the Connection
dbbean.connect();
// Execute the SQL and get a HTML table
out.println(dbbean.getResultAsHTMLTable(sql_string));
// Close the Bean to close the connection
dbbean.close();
} catch (SQLException e) {
out.println("<P>" + "There was an error doing the query:");
out.println ("<PRE>" + e + "</PRE> \n <P>");
}
%>
</BODY>
</HTML>

```

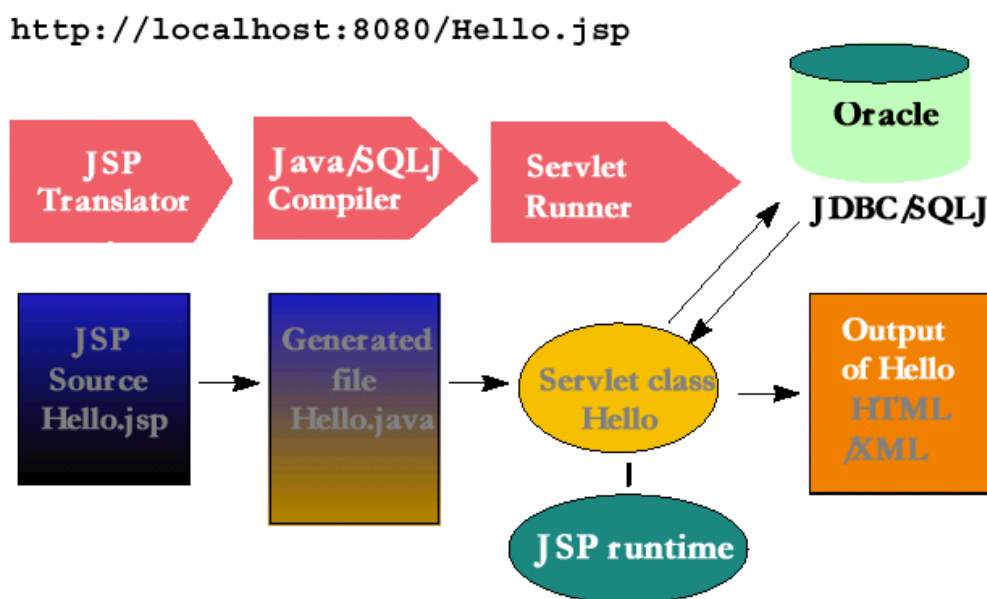
Το αποτέλεσμα της παραπάνω σελίδας θα είναι:



Εικόνα 2.14.3.δ

#### 2.14.4 Εκτέλεση JSP

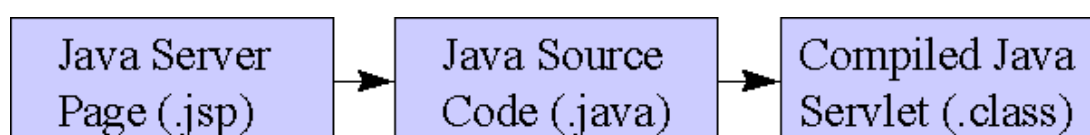
Το παρακάτω σχήμα δείχνει με παραστατικό τρόπο τη διαδικασία εκτέλεσης μιας JSP. Αρχικά ο χρήστης επιλέγει ένα URL στον browser του που περιέχει μια JSP σελίδα με όνομα Hello.jsp, για λόγους απλότητας στον τοπικό υπολογιστή. Το αρχείο JSP μεταφράζεται από τον JSP translator σε .java αρχείο, το οποίο με τη σειρά του μεταγλωττίζεται σε ένα .class αρχείο ενδιάμεσου bytecode. Στη συνέχεια το αρχείο αυτό εκτελείται κατά το χρόνο εκτέλεσης (JSP runtime) όπως ένα servlet. Το τελευταίο επικοινωνεί με την βάση δεδομένων Oracle8i του συστήματος μέσω JDBC drivers εκτελώντας καθαρή SQL σε αυτή. Το αποτέλεσμα στέλνεται πίσω στον browser του χρήστη σε html ή/και xml format.



Εικόνα 2.14.4.α

Αναλυτικά, ακολουθούνται τα εξής στάδια κατά την εκτέλεση ενός JSP:

- **Φάση μετάφρασης:** Η JSP engine πραγματοποιεί όλα τα βήματα της μετάφρασης με πλήρη διαφάνεια προς το χρήστη. Αν η JSP δεν είναι invoked ή έχει αλλάξει, η JSP engine τη μεταφράζει σε καθαρό πηγαίο κώδικα Java, ο οποίος περιέχει ένα Java servlet. Το τελευταίο υλοποιεί το javax.servlet.Servlet interface και περιλαμβάνει τη λογική που περιέχεται στο .jsp αρχείο. Αυτό με τη σειρά του μεταγλωττίζεται από την JSP engine σε ψευδοκώδικα Java παράγοντας ένα .class αρχείο. Αν όμως η JSP έχει ήδη γίνει translated από κάποιο προηγούμενο request και δεν έχει αλλάξει στο ενδιάμεσο διάστημα, τότε η παραπάνω φάση μετάφρασης παραλείπεται εντελώς.



Εικόνα 2.14.4.β

- **Φάση προώθησης του request:** Η φάση αυτή είναι κοινή τόσο για τις JSPs όσο και για τα servlets. Αν έχει ήδη φορτωθεί το servlet, τότε χρησιμοποιείται το ίδιο servlet αντικείμενο για την απάντηση στον client, αλλιώς το φορτώνεται το νέο .class αρχείο και δημιουργείται ένα νέο servlet object.

#### 2.14.5 Πλεονεκτήματα

Συμπερασματικά, τα κυριότερα πλεονεκτήματα των JSPs μπορούν να συγκεντρωθούν στα εξής:

- **“Write Once, Run Anywhere”<sup>TM</sup>:** Είναι ανεξάρτητα πλατφόρμας σε επίπεδο δυναμικών web σελίδων, web servers και server components, γεγονός ιδιαίτερα σημαντικό που τους δίνει σημαντικό προβάδισμα απέναντι στον ανταγωνισμό (βλ. **ASP**, Netscape Server Application Programming Interface-**NSAPI**, Internet Server Application Programming Interface-**ISAPI**). Με άλλα λόγια, μπορεί κανείς να γράψει JSP σελίδες σε οποιαδήποτε πλατφόρμα υπολογιστών, να τις τρέξει σε οποιοδήποτε Java enabled web server και να τις προσπελάσει από οποιοδήποτε web browser.
- **Υψηλής ποιότητας εργαλεία υποστήριξης:** Ακριβώς η παραπάνω ιδιότητα των JSPs δίνει τη δυνατότητα στον προγραμματιστή να επιλέξει από μια μεγάλη γκάμα εργαλείων το καταλληλότερο για τις ανάγκες του project που ετοιμάζει.
- **Διαχωρισμός των ρόλων:** μια μερίδα προγραμματιστών γράφουν reusable components που αλληλεπιδρούν με server-side αντικείμενα, ενώ μία άλλη ασχολείται με την συνένωση στατικού και δυναμικού περιεχομένου και δεδομένων. Με άλλα λόγια απομονώνεται η business/application logic του project σε standard, επαναχρησιμοποιούμενα JavaBeans components από την presentation logic, που μπορεί να υλοποιηθεί με στατική html, tags και Java scriptlets.
- **Επαναχρησιμοποίηση components και tag libraries:** Μια πληθώρα reusable components είναι διαθέσιμη μέσω των JSP: JavaBeans components, Enterprise JavaBeans components και tag libraries προσφέρουν πλήρη ευελιξία προγραμματισμού και σχεδιασμού, καθώς και εξοικονόμηση χρόνου κατά τη διαδικασία παραγωγής μιας εφαρμογής.
- **Διαχωρισμός δυναμικού-στατικού περιεχομένου:** Απλοποιεί και αποσαφηνίζει την παραγωγή περιεχομένου.
- **Compiled once:** Οι JSP μεταγλωττίζονται την πρώτη φορά που θα γίνουν invoked και μετέπειτα τρέχουν ως μεταγλωττισμένος κώδικας, σε αντίθεση με άλλες παρεμφερείς τεχνολογίες, όπως τα ASP και οι Perl-based τεχνολογίες.
- **Καλή κληρονομιά:** Εν γένει κληρονομούν όλα τα προτερήματα της οικογένειας της Java, παρέχουν πρόσβαση σε standard Java APIs και

αποτελούν ανοικτό standard. Είναι σχετικά εύκολα στη χρήση και διαθέτουν απλό και βολικό συντακτικό. Ενδεικτικά παρατίθεται ο παρακάτω συγκριτικός πίνακας, για τις πλατφόρμες ανάπτυξής τους και κάποια θεωρητικά στοιχεία:

	<b>Native Platforms Supported</b>	<b>Interpreted or Compiled?</b>	<b>Component Model</b>	<b>Implementation Language</b>
Java Server Page (JSP)	Any Platform with a JVM & JSP Engine	Compiled	JavaBeans, Enterprise JavaBeans	Java
Active Server Pages	Microsoft Windows/IIS	Interpreted	COM	VB/C++
Netscape Server Application Programming Interface (NSAPI)	Netscape Servers	Compiled	None	C/C++
Internet Server Application Programming Interface (ISAPI)	Microsoft Windows/IIS	Compiled	COM	C/C++
Common Gateway Interface (CGI)/Perl	Various	Interpreted	None	Perl

Ακολουθεί, τέλος, μια λίστα με τα πλέον χρησιμοποιούμενα jsp tags και τις επεξηγήσεις τους:

JSP tags	επεξήγηση
<code>&lt;!-- comment [ &lt;%=expression %&gt; ] --&gt;</code>	σχόλιο που αποστέλλεται ως viewable page source στον πελάτη
<code>&lt;%-- comment --%&gt;</code>	σχόλιο που δεν αποστέλλεται στον πελάτη
<code>&lt;%! declaration %&gt;</code>	δήλωση μεταβλητών ή μεθόδων
<code>&lt;%= expression %&gt;</code>	έκφραση
<code>&lt;% code fragment %&gt;</code>	απόσπασμα
<code>&lt;%@ include file="relative URL" %&gt;</code>	include ενός αρχείου στον πηγαίο κώδικα του JSP
<code>&lt;% @ page [ ... ] %&gt;</code>	λίστα attributes, που εφαρμόζονται σε ολόκληρη την JSP.
<code>&lt;%@ taglib uri="URIToTagLibrary"</code>	ορισμός ενός tag library και prefix

<code>prefix="tagPrefix" %&gt;</code>	για τα custom tags στην JSP.
<code>&lt;jsp:forward page="{relativeURL / &lt;%= expression %&gt; }" /&gt;</code>	πραγματοποιεί forward ενός client request προς επεξεργασία σε ένα άλλο αρχείο html, jsp ή servlet.
<code>&lt;jsp:getProperty name="beanInstanceName" property="propertyName" /&gt;</code>	παίρνει την τιμή ενός Bean property.
<code>&lt;jsp:setProperty name="beanInstanceName" /&gt;</code>	ορίζει την τιμή ενός Bean property.
<code>&lt;jsp:include page=".." flush="true" /&gt;</code>	συμπεριλαμβάνει δεδομένα σε ένα JSP από ένα άλλο αρχείο, χωρίς να τα κάνει parse.
<code>&lt;jsp:useBean id=".." ... /&gt; .. .. &lt;/jsp:useBean&gt;</code>	εντοπίζει ή instantiates ένα Bean με συγκεκριμένο όνομα και εμβέλεια.

#### 2.14.6 http-session tracking:

Τα JSPs κάνουν ευρεία χρήση του Java interface `javax.servlet.http.HttpSession` [L25], το οποίο παρέχει έναν αποδοτικό τρόπο αναγνώρισης ενός χρήστη κατά τη διάρκεια πολλαπλών page requests σε ένα web site και τη δυνατότητα αποθήκευσης πληροφοριών σχετικών με τον συγκεκριμένο χρήστη. Συγκεκριμένα ο container του servlet χρησιμοποιεί το interface αυτό προκειμένου να φτιάξει ένα session μεταξύ ενός http client και ενός http server. Το session αυτό διατηρείται για μια συγκεκριμένη χρονική περίοδο για παραπάνω της μίας σύνδεσης ή page request από το χρήστη και είναι προφανώς μοναδικό ανά χρήστη, καθώς υπάρχει 1 προς 1 αντιστοιχία. Ο server είναι σε θέση να διατηρεί το session αυτό με διάφορους τρόπους, οι πιο διαδεδομένοι από τους οποίους είναι κυρίως τα cookies και το url-rewriting.

Το παραπάνω interface επιτρέπει σε ένα servlet ενίοτε να προσπελάσει και ενίοτε να χειραγωγήσει πληροφορίες σχετικές με το session, όπως το μοναδικό ανά session *session identifier*, την ώρα δημιουργίας του session *session creation time* και την ώρα τελευταίας προσπέλασής του από τον client *last accessed time*, καθώς και να προσδέσει αντικείμενα (objects) σε αυτό, επιτρέποντας έτσι σε πληροφορίες χρήστη να περνάνε αμετάβλητες ανάμεσα στις διάφορες page requests ή connections. Όταν μια εφαρμογή αποθηκεύσει ένα object στο session, ή το απομακρύνει από αυτό, τότε το session ελέγχει αν το object αυτό υλοποιεί (implements) το `HttpSessionBindingListener` και αν το κάνει, τότε ειδοποιεί το αντικείμενο ότι προσδέθηκε ή αφαιρέθηκε από το session.

Ένα servlet, βέβαια, θα πρέπει να μπορεί να αντιμετωπίσει περιπτώσεις «δύσκολων» clients, όπου οι τελευταίοι έχουν απενεργοποιήσει τα cookies και για τις οποίες έχουν προβλεφθεί κατάλληλες υλοποιήσεις και μέθοδοι, όπως η χρήση url-rewriting, όπου το session id γράφεται ως μέρος του ίδιου του url query string parameter και των



isNew() και getSession μεθόδων. Οι τελευταίες διαφέρουν από JSP container σε JSP container, αλλά η γενική ιδέα τους παραμένει η ίδια.

Επίσης, θα πρέπει κανείς να αναλογιστεί ότι σε κάθε χρήστη αντιστοιχεί τουλάχιστον ένα session με κάποια objects, οπότε ανάλογα το φορτίο χρηστών που αναμένεται, θα πρέπει να γίνει και λογική χρήση τους. Για παράδειγμα, αν αποθηκεύονται 512k ανά χρήστη στο session object και αναμένονται 1000 ταυτόχρονοι (ή και «χρονικά γείτονες») online χρήστες, θα απαιτείται σε επίπεδο hardware από το σύστημα τουλάχιστον 512 MB RAM αποκλειστικά για τη διαχείριση των μεταβλητών του session. Συνεπώς, κατά τη χρησιμοποίηση του interface αυτού απαιτείται προσεκτικός σχεδιασμός και συνεργασία με όλες τις συνιστώσες του συστήματος.

Συνοψίζοντας, ένα session object αποτελεί έναν optimal τρόπο διατήρησης του state του εκάστοτε χρήστη για ένα υπό φυσιολογικές συνθήκες stateless http web site, πράγμα ιδιαίτερα σημαντικό για την εφαρμογή που αναπτύχθηκε.

Τέλος, να αναφέρουμε ότι το καθάρισμα όλων των παραμέτρων (π.χ. identification) του εκάστοτε χρήστη, οι οποίες έχουν περάσει ως αντικείμενα στο session του, πραγματοποιείται σε μία logout page (βλ. και § 7.6), στην οποία και θα μεταβεί εν προκειμένω απαραίτητως και αυτόματα ο χρήστης κατά την έξοδό του. Μέρος του κώδικα της τελευταίας έχει ως εξής, όπου παρατηρούμε και τις αναφορές στο session:

```
<TITLE>Logout page</TITLE>

</head>

<body>

<%
    session= request.getSession(true);
    System.out.println("1 "+session.getId());

// *****

    System.out.println("session creation time");
    session.getCreationTime();
    System.out.println(session.getCreationTime());

    System.out.println("session last accessed time");
    session.getLastAccessedTime();
    System.out.println(session.getLastAccessedTime());

    System.out.println("session isNew?");
    session.isNew();
    System.out.println(session.isNew());

// *****

    session.removeAttribute("que");
    session.removeAttribute("quekat");
    session.removeAttribute("epileksam");
    session.removeAttribute("epiluser");
    session.removeAttribute("test");
    session.removeAttribute("queafter");
    session.removeAttribute("am");
```

```

    session.removeAttribute("kodikos");

    System.out.println("session is invalidated now!");
    System.out.println("-----");
    System.out.println("-----");
    System.out.println("Teleytaio simeio enos session");
    System.out.println(session.getMaxInactiveInterval());
    System.out.println("-----");
    System.out.println("-----");
    session.invalidate();
%>

<br><br><br>

```

Παρατηρούμε ότι πραγματοποιείται manual invalidation του http-session αντί της αναμονής του εκάστοτε server (π.χ. Tomcat) για timeout, παράμετρος που καθορίζεται και μέσα από την εφαρμογή με χρήση της μεθόδου `setMaxInactiveInterval(int interval)`.

## 2.15 JavaBeans:

### 2.15.1 Ορισμός-Γενικά

Τα JavaBeans είναι ένα αντικειμενοστραφές προγραμματιστικό interface που αναπτύχθηκε από τη Sun Microsystems και το οποίο επιτρέπει την κατασκευή επαναχρησιμοποιούμενων εφαρμογών ή/και «blocks που κατασκευάζουν κώδικα» (program building blocks), που ονομάζονται components. Αυτά μπορούν να γίνουν deploy στο δίκτυο ή/και σε οποιαδήποτε πλατφόρμα λειτουργικού συστήματος. μια πολύ διαδεδομένη χρήση τους αποτελεί η μετατροπή απλών web σελίδων σε σελίδες με interactive χαρακτηριστικά, μέσω και της χρήσης JSPs. Τελευταία έκδοση προδιαγραφών είναι η **JavaBeans 1.01** [L22].

Προκειμένου ο προγραμματιστής να δημιουργήσει αυτά τα components, γράφει καθαρό κώδικα Java και συμπεριλαμβάνει σε αυτόν δηλώσεις και εντολές JavaBeans, οι οποίες και περιγράφουν *ιδιότητες* του εκάστοτε component, όπως χαρακτηριστικά user interface, και *events* που σκανδαλίζουν ένα bean έτσι ώστε να επικοινωνήσει με άλλα beans στο ίδιο container ή αλλού στο δίκτυο.

Η προδιαγραφή από τη Sun Microsystems των JavaBeans [L22] καθορίζει μια σειρά από interfaces και λεπτομέρειες packaging που υλοποιούνται από ένα σύνολο από Java classes προκειμένου να γίνει ένα JavaBean. Ουσιαστικά, δηλαδή, ένα JavaBean είναι ένα «πακεταρισμένο σύνολο από Java classes» (packaged set of Java classes) και μπορεί να το χρησιμοποιήσει κανείς όπως οποιαδήποτε άλλη Java class στην εφαρμογή που κατασκευάζει. Είναι σε θέση να:

- instantiate το JavaBean μέσα στο κώδικά του
- ορίζει τις ιδιότητες του JavaBean (set the JavaBeans properties)
- καλεί μεθόδους
- register event handlers για events

### 2.15.2 Πλεονεκτήματα

Τα πλεονεκτήματα των JavaBeans είναι αρκετά και κάνουν την ανάπτυξη εφαρμογών γρηγορότερη και πιο αποδοτική. Συνοψίζοντας:

- ❖ **standard component model:** Αποτελούν ένα **ανοικτό** standard για components, με αποτέλεσμα ο οποιοσδήποτε να μπορεί να εισάγει και ενσωματώσει την τεχνολογία αυτή στις εφαρμογές του.
- ❖ **platform independence:** Όπως κάθε προϊόν που ανήκει στην οικογένεια της Java, έτσι και τα JavaBeans είναι παντελώς ανεξάρτητα από πλατφόρμες υπολογιστών. Συνεπώς το σύνολο των JavaBeans που κατασκευάζει ο προγραμματιστής μπορεί να γίνει deployed σε διάφορα περιβάλλοντα, χωρίς να υπάρξει οποιαδήποτε αλλαγή στον κώδικά τους και χωρίς να χρειάζεται να ανησυχήσει κανείς για προβλήματα συμβατότητας.
- ❖ **components «τρίτων» κατασκευαστών:** Στον τομέα αυτό υπάρχουν ήδη ένας μεγάλος αριθμός reusable JavaBeans από «τρίτους» κατασκευαστές, τα οποία μπορούν εύκολα να μετατραπούν και να καλύψουν συγκεκριμένες ανάγκες λογισμικού.

## 2.16 Business Components for Java (BC4J)

### 2.16.1 Ορισμός:

Ο όρος Oracle **Business Components for Java (BC4J)** αποτελεί το προγραμματιστικό framework, υλοποιημένο σε Java και XML, του εργαλείου **Oracle JDeveloper** για την ανάπτυξη scalable, multi-tier εφαρμογών βάσεων δεδομένων από επαναχρησιμοποιούμενα business components. Είναι ένα 100%-Java framework με επιπρόσθετες δυνατότητες XML, το οποίο και επιτρέπει την παραγωγική ανάπτυξη scalable multi-tier εφαρμογών με πλήρη υποστήριξη Java.

Τα BC4J [L23] παρέχουν στον προγραμματιστή ένα σύνολο «έξυπνων» blocks κατασκευής λογισμικού, τα οποία και συνεργάζονται μεταξύ τους για να διευθύνουν κοινά facilities, όπως:

- ☆ Να συγγράφουν και να τεστάρουν παραγωγικά την business logic σε components, τα οποία και αλληλεπιδρούν απ' ευθείας με την βάση δεδομένων.
- ☆ Να επαναχρησιμοποιούν ευέλικτα την business logic μέσα από πολλαπλά SQL-based views από δεδομένα.
- ☆ Να προσπελαίνουν ικανοποιητικά και να ενημερώνουν/ανανεώνουν τα views από servlets, JSPs, και Thin-Java Swing clients.
- ☆ Να παραμετροποιούν εύκολα την functionality της εφαρμογής σε στρώματα, χωρίς να απαιτείται μεταβολή της ήδη ανεπτυγμένης εφαρμογής.

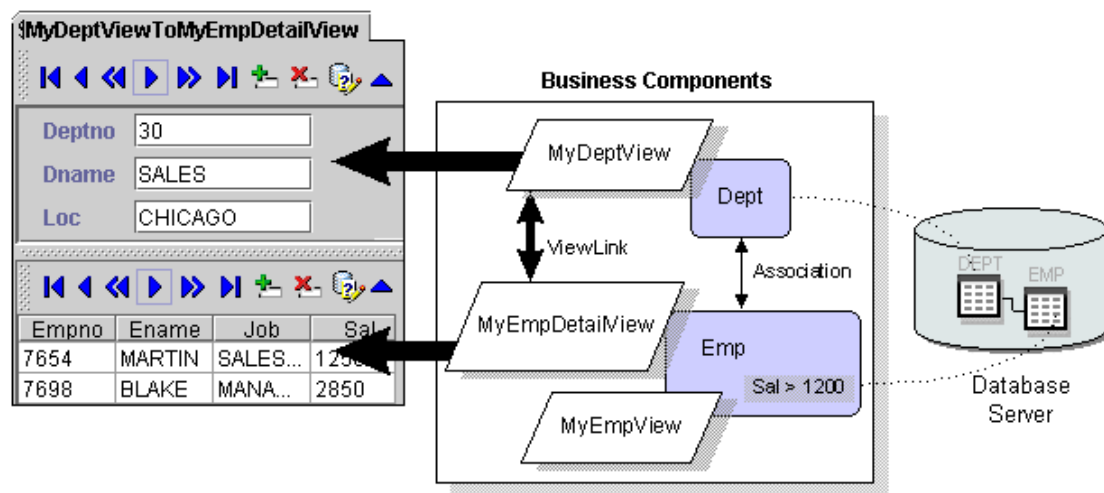
Με όλα τα παραπάνω γίνεται σαφές ότι τα BC4J ωθούν τον προγραμματιστή να συγκεντρωθεί στην υλοποίηση των λύσεων, με αποτέλεσμα να ελαττώνεται σημαντικά το ρίσκο, ο χρόνος και το κόστος παραγωγής μιας εφαρμογής,

### 2.16.2 Είδη BC4J

Υπάρχουν διάφορα είδη business components. Στο σύστημα εγγραφών χρησιμοποιήσαμε τα ακόλουθα (χρησιμοποιώ τους αγγλικούς όρους αμετάφραστους, καθώς αποδίδουν καλύτερα, κατά την άποψή μου, την λογική που ακολουθείται):

Business Components	Περιγραφή
Entity Object	Περιέχει την business logic για έναν πίνακα, ένα view ή ένα synonym της βάσης δεδομένων.
View Object	Χρησιμοποιεί SQL queries για να καθορίσει φιλτραρισμένα υποσύνολα attributes από τα παραπάνω entity objects.
Association	Καθορίζει την πιθανή σχέση μεταξύ δύο entity objects.
View Link	Καθορίζει την πιθανή σχέση μεταξύ δύο view objects.
Application Module	Παρέχει ένα λογικό container για τα view objects, τα view links, και τις δοσοληψίες που πραγματοποιούνται.

Κάθε BC4J αντιπροσωπεύεται από ένα αρχείο XML και ένα ή περισσότερα αρχεία Java. Στο αρχείο XML αποθηκεύονται σημαντικά metadata, ενώ στα αρχεία Java αποθηκεύεται ο κώδικας του εκάστοτε αντικειμένου, ο οποίος και υλοποιεί συγκεκριμένη συμπεριφορά, η οποία εξαρτάται από την ίδια την εφαρμογή. Κάθε αντικείμενο δε, είναι οργανωμένο σε ένα «πακέτο» (package) ακολουθώντας το γνωστό directory-based συντακτικό της γλώσσας Java. Στο παρακάτω σχήμα φαίνονται παραστατικά όλα τα παραπάνω:



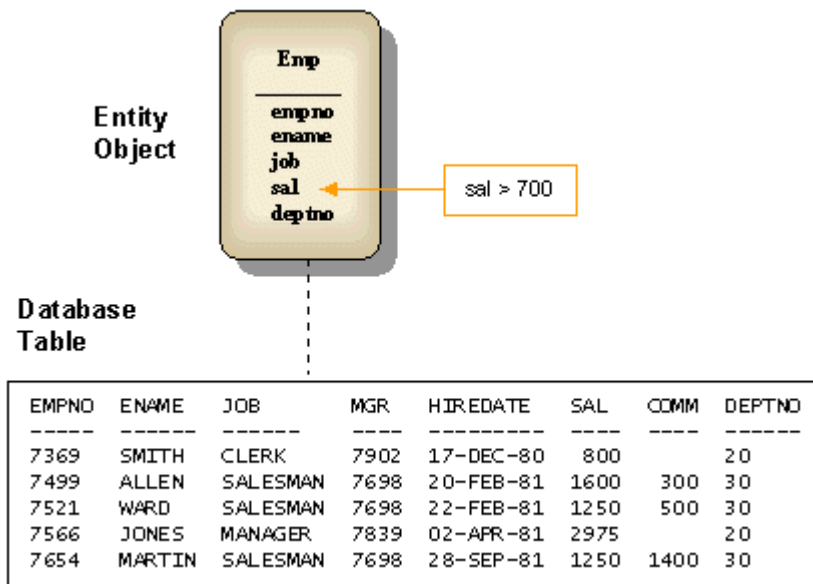
Εικόνα 2.16.2.α

ο **Entity Objects:**

Ένα entity object παρέχει μια αναπαράσταση σε μορφή αντικειμένου των δεδομένων, τα οποία τοποθετεί στην cache από μια γραμμή ενός πίνακα, όψης ή συνωνύμου στη βάση δεδομένων. Κάθε πίνακας της βάσης δεδομένων αναπαρίσταται σαν ένα ξεχωριστό entity object, οι δε στήλες του αντιπροσωπεύονται από τα attributes του entity object. Προφανώς μπορεί και είναι ακριβώς θεμιτή η επαναχρησιμοποίηση της business logic που ενθυλακώνεται με τον παραπάνω τρόπο στα entity objects.

Επίσης, είναι δυνατός ο ορισμός σχέσης μεταξύ δύο entity objects, μέσω της χρήσης ενός association. Ένα association ορίζει αυτή τη σχέση σε επίπεδο entity object πηγής και προορισμού και ταιριάζει attributes μεταξύ αυτών. Τυπικά αυτά τα attributes καθορίζονται από τους περιορισμούς κλειδιών που διέπουν τους ίδιους τους πίνακες της βάσης που αντιπροσωπεύουν, αλλά μπορεί κανείς να ορίσει και association χρησιμοποιώντας οποιοδήποτε entity object attribute.

Τα entity objects δεν προσπελούνται απ' ευθείας από τον client, αλλά οι τελευταίοι χρησιμοποιούν για την αλλαγή των δεδομένων και την προσπέλασή τους γενικά, τα view objects. Οι όποιες αλλαγές αντικατοπτρίζονται στη βάση δεδομένων όταν γίνεται commit η δοσοληψία, παραμένοντας μέχρι τότε στην cache. Προφανώς, entity και view objects επικοινωνούν άμεσα μεταξύ τους μέσω events και κλήσεων Java. Στο παρακάτω σχήμα φαίνεται ένα παράδειγμα entity object με το όνομα Emp από τον default πίνακα EMP της Oracle8i, με την προσθήκη μάλιστα και ενός κανόνα validation πάνω στο attribute sal:



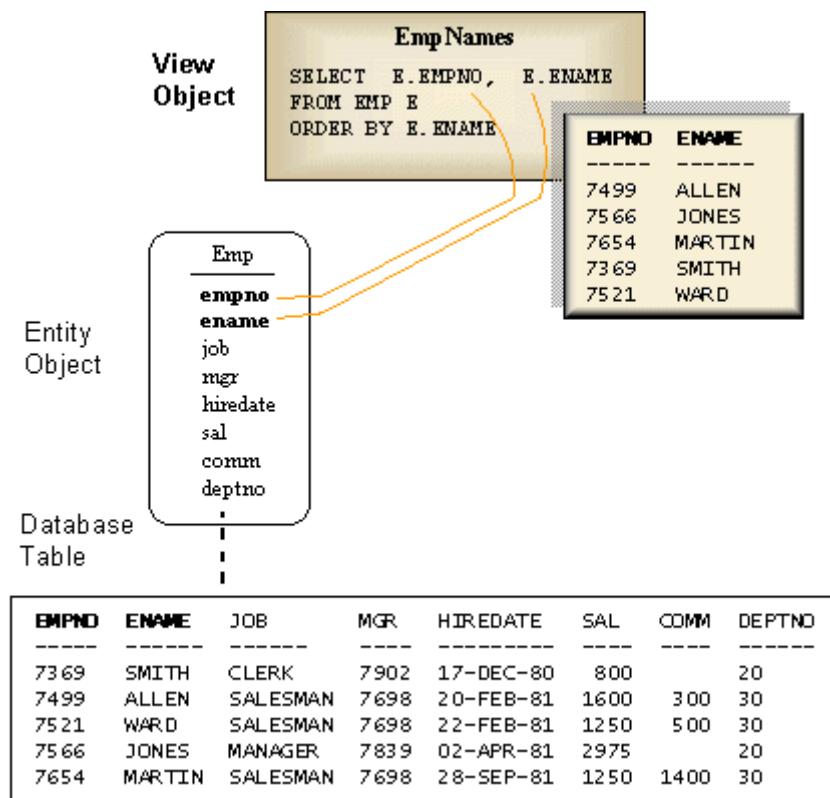
Εικόνα 2.16.2.β

#### ○ *View Objects:*

Ένα view object είναι σύμφωνα με τα παραπάνω ένα component που ενθυλακώνει καθαρό κώδικα SQL καθώς και metadata, τα οποία και αντιστοιχούν τις στήλες στη δήλωση select σε attributes ενός ή και περισσότερων entity objects. Πραγματοποιεί όλη τη «βρώμικη» δουλειά, ενώνοντας, φιλτράροντας και σορτάροντας τα δεδομένα, μετατρέποντας τα έτσι σε μία μορφή κατάλληλη για παρουσίαση. Ο client δεν χρειάζεται να γνωρίζει πλέον τίποτα για τα υποκείμενα entity objects, και μπορεί απερίσπαστος να πλοηγηθεί ή να ενημερώσει τα σύνολα γραμμών που του παρέχονται από τα view objects.

Σε αυτή την περίπτωση είναι δυνατός ο ορισμός πολλαπλών view objects πάνω σε ένα entity object, με άλλα κάθε φορά χαρακτηριστικά, ενώ ένα view object μπορεί να επιλέγει δεδομένα από πολλά διαφορετικά entity objects. Η πληροφορία τοποθετείται στην cache στο επίπεδο των entity objects και όλες οι αναφορές view objects κατά τη διάρκεια της ίδιας δοσοληψίας την μοιράζονται. Συνακόλουθα, οποιαδήποτε αλλαγή πραγματοποιηθεί μέσω ενός view object, γίνεται αμέσως προσβάσιμη από τα άλλα view objects στην ίδια δοσοληψία.

Η διασύνδεση view objects σε επίπεδο master-detail είναι δυνατή μέσω των view links σε ένα application module, με την υποσημείωση ότι τα detail view objects συγχρονίζονται αυτόματα με τα master view objects τους. Στο παρακάτω σχήμα φαίνεται ένα παράδειγμα view object με το όνομα EmpNames, το οποίο και λειτουργεί πάνω στο entity object του προηγούμενου παραδείγματος, Emp, παρέχοντας τελικά μια όψη του πίνακα EMP από τη βάση δεδομένων.



Εικόνα 2.16.2.γ

Τέλος, η κύρια διαφορά associations και view links είναι ότι τα πρώτα είναι διπλής κατεύθυνσης, ενώ τα δεύτερα μονής.

- *Application modules:*

Ένα application module τώρα, αποτελεί όπως το ορίσαμε ένα λογικό container για αντικείμενα σχετιζόμενα με ένα συγκεκριμένο task. Παρέχουν ένα απλό runtime διασύνδεσης δεδομένων, υλοποιώντας μία σύνδεση ανά application model. Αυτό το επιτυγχάνουν προγραμματιστικά μέσω της ύπαρξης της κλάσης AppRegistry, την οποία και χρησιμοποιούν οι clients, όποιοι και αν είναι αυτοί, προκειμένου να προσπελάσουν, να διευθύνουν και να μοιραστούν εν γένει ένα pool από application modules.

Γενικά, ένα application pool είναι ιδιαίτερα σημαντικό στην όλη διαδικασία, καθώς;

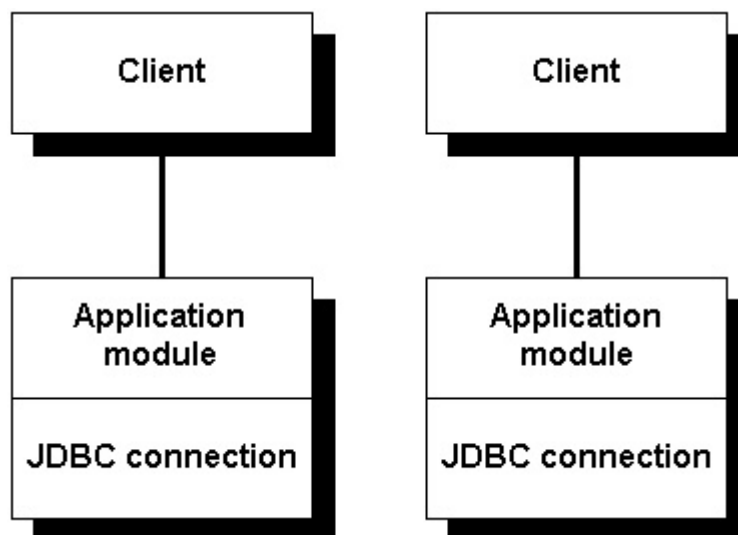
- ☆ καθορίζει ένα session στη βάση δεδομένων και ένα boundary δοσοληγιών
- ☆ ελέγχει την ταυτόχρονη πρόσβαση στα δεδομένα της βάσης
- ☆ ομαδοποιεί τα σχετιζόμενα view objects σε ένα μοντέλο δεδομένων, το οποίο οι clients μπορούν να προσπελάσουν σε μία μόνο περιήγηση του δικτύου
- ☆ αποθηκεύει τον application-specific κώδικα Java σε ένα αντικείμενο στο μεσαίο επίπεδο.

- ☆ δημιουργεί view objects μόνο προς ανάγνωση
- ☆ προσπελαύνει βοηθητικές μεθόδους για δοκιμές και debugging.

- ***Application module pooling:***

Το pooling του application module αποτελεί ένα απαραίτητο feature μίας scalable εφαρμογής καθώς επιτρέπει σε ένα μικρό σχετικά αριθμό από application module instances να εξυπηρετούν ένα συγκριτικά μεγαλύτερο αριθμό από requests. Επιτρέπει δε στους clients την επαναχρησιμοποίηση των παραπάνω στιγμιότυπων, αντί για δημιουργία νέων για κάθε ξεχωριστό client request.

Σε περίπτωση που δεν ακολουθηθεί η σχεδίαση του application module pooling, θα έχουμε την παρακάτω διαδικασία, όπου για κάθε client request δημιουργείται ένα καινούριο στιγμιότυπο, καθώς και μια νέα JDBC connection:

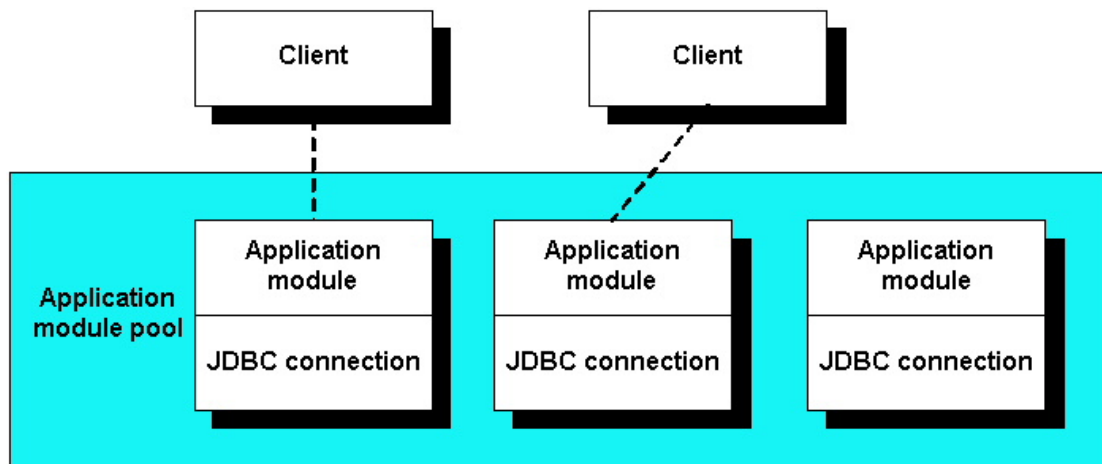


Εικόνα 2.16.2.δ

Προφανώς η επίδοση και η scalability του παραπάνω συστήματος υποφέρουν από τις ολοένα αυξανόμενες JDBC συνδέσεις και από την ύπαρξη πολλαπλών στιγμιότυπων του ίδιου application module στην μνήμη του.

Προχωρώντας ένα βήμα παραπάνω, υπάρχει η δυνατότητα για application module pooling χωρίς την επιπλέον ύπαρξη JDBC connection pooling. Στην περίπτωση αυτή ακολουθείται η παρακάτω διαδικασία, όπου κάθε application module διαθέτει τη δική του σύνδεση με τη βάση, την οποία και διατηρεί για όλη τη διάρκεια της ζωής του. Οι clients συνδέονται στα application modules, με αποτέλεσμα κάθενας να έχει τη δική του middle-tier connection.

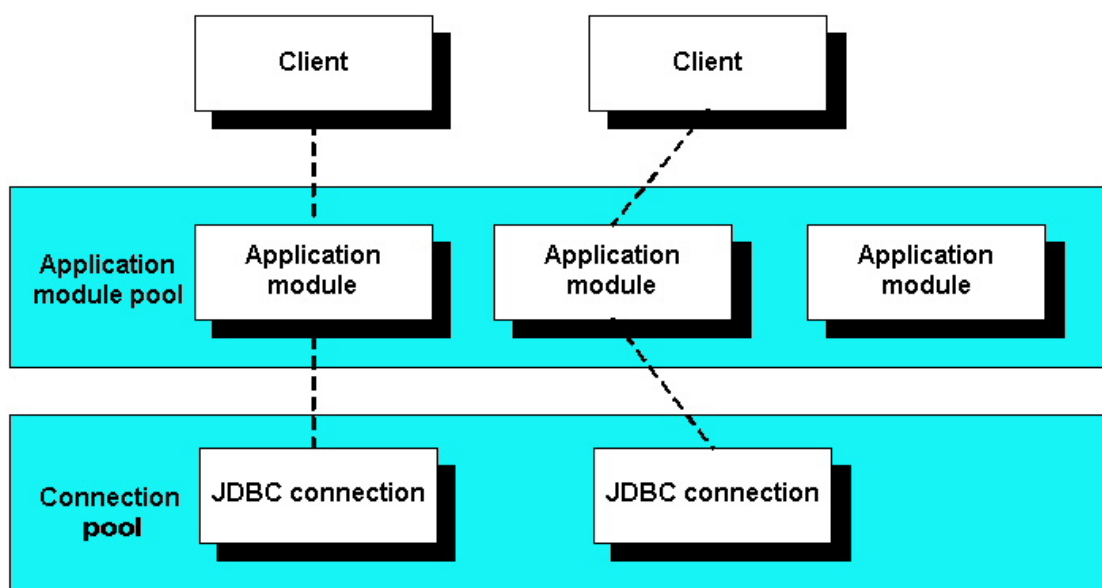




Εικόνα 2.16.2.ε

Όταν υπάρξει ζήτηση για ένα application module instance από τον client, επιλέγεται και κρατείται ένα από το pool των στιγμιότυπων, ενώ όταν ο client δεν το χρειάζεται πλέον, απελευθερώνεται πάλι πίσω σε αυτό (π.χ. με το expire ενός http session). Αν δεν υπάρχουν διαθέσιμα στιγμιότυπα στη δεξαμενή, τότε δημιουργείται ένα καινούριο, ενώ πρέπει να σημειώσουμε ότι υπάρχει ένας pool manager για κάθε στιγμιότυπο JVM. Δεν υφίσταται άνω όριο στη δημιουργία νέων στιγμιότυπων, με αποτέλεσμα να μην υπάρχει κανένας περιορισμός από το business components framework στον αριθμό των εν δυνάμει εξυπηρετούμενων clients. Στην πράξη βέβαια άλλες παράμετροι του συστήματος επιβάλλουν περιορισμούς σε αυτόν τον αριθμό.

Τέλος, είναι δυνατή και μια περαιτέρω επέκταση των παραπάνω, με χρήση pooling τόσο σε επίπεδο application modules, όσο και σε επίπεδο JDBC connections. Η λογική που ακολουθείται είναι η ίδια και στην περίπτωση αυτή. Υπάρχει ένα connection pool για κάθε μοναδικό συνδυασμό username, password και database, με αποτέλεσμα να θεωρείται ιδανικό για τις περιπτώσεις που τα παραπάνω παραμένουν σταθερά σε μία εφαρμογή.



Εικόνα 2.16.2.στ

Στην υλοποίηση του συγκεκριμένου συστήματος επιλέχθηκε η λογική της διατήρησης του state τόσο του application module όσο και της database ανάμεσα στα http requests σε επίπεδο application module instance. Για το λόγο αυτό, πραγματοποιείται «καθάρισμα» των στιγμιότυπων ανάμεσα σε http requests προερχόμενα από διαφορετικούς clients.

## **2.17 Oracle Jdeveloper:**

Πρόκειται για ένα περιβάλλον ανάπτυξης εφαρμογών λογισμικού, πλήρως συμβατού με την J2EE. Περιέχει έναν πολύ καλό Java debugger για το debugging των παραγόμενων εφαρμογών, προσφέρει μια πληθώρα editors και προγραμματιστικών visual εργαλείων για την ανάπτυξη εφαρμογών και components, όπως τα applets, τα JavaBeans, οι JavaServer Pages (**JSP**), τα **servlets**, τα αντικείμενα **CORBA** και τα Enterprise JavaBeans (**EJB**). Ιδιαίτερώς δε, προσφέρει την τεχνολογία των Business Components for Java (**BC4J**), για την δημιουργία scalable, υψηλής απόδοσης εφαρμογών Internet.

Μία version του **Borland Jbuilder** αποτελεί το Integrated Development Environment (IDE) του Jdeveloper [24], παρέχοντας ένα στέρεο και αποδοτικό user interface για τους προγραμματιστές. Αυτό όμως που κάνει το εργαλείο αυτό ιδιαίτερα ξεχωριστό είναι το framework που παρέχει για τα BC4J.

Στα πλεονεκτήματά του ακόμα είμαστε σε θέση να θεωρήσουμε την υποστήριξη και άλλων βάσεων δεδομένων πέραν της «μητρικής» Oracle, καθώς είναι πλήρως συμβατό με το SQL-92 JDBC standard,. Επίσης, σημαντική είναι και η ευελιξία που παρέχεται στον προγραμματιστή να κάνει deploy τις ίδιες εφαρμογές σε Java Server Pages, EJBs και CORBA components χρησιμοποιώντας τον ίδιο κώδικα γι' αυτές. Με τον τρόπο αυτό ο βασικός κώδικας μιας εφαρμογής καθίσταται περισσότερο επαναχρησιμοποιήσιμος από ποτέ και για διάφορα είδη πλατφόρμων, από web browsers μέχρι και ασύρματες συσκευές χειρός.

Τέλος, στις προδιαγραφές του προϊόντος αναγράφονται ως δυνατές πλατφόρμες εγκατάστασής του οι: Windows NT και Windows 2000 με δυνατότητες deployment σε οποιοδήποτε σύστημα υποστηρίζει Java. Οι minima απαιτήσεις του σε hardware είναι: Pentium 200 MHz, 96 MB RAM και 280 MB χώρος στο σκληρό δίσκο, με το σύστημα στο οποίο δούλεψε για τη κατασκευή components του συστήματος εγγραφών να διαθέτει: Pentium III 800 MHz, 512 MB RAM και 40 GB χώρο στο σκληρό, αντίστοιχα.

# 3

## **ΠΕΡΙΓΡΑΦΗ ΠΡΟΒΛΗΜΑΤΟΣ**

### ***3.1 Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών***

Το Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών (**Η.Μ.& Μ.Υ.**) του Εθνικού Μετσόβιου Πολυτεχνείου (**Ε.Μ.Π.**) με την σημερινή του ονομασία αριθμεί 10 χρόνια ζωής. Η ιστορία όμως του τμήματος είναι πολύ πιο παλιά και χρονολογείται από το 1917 ως ενιαίο τότε Τμήμα Μηχανολόγων-Ηλεκτρολόγων. Θεωρείται, δε, ότι αποτελεί την κορυφή στον τομέα της τεχνολογικής έρευνας και ανάπτυξης, αλλά και στην παροχή άρτιας και πλήρους εκπαίδευσης στους προπτυχιακούς και μεταπτυχιακούς φοιτητές του.

#### ***3.1.1 Ιστορικά στοιχεία:***

Ο διαχωρισμός των γνωστικών αντικειμένων της ενιαίας Σχολής Μηχανολόγων-Ηλεκτρολόγων γινόταν όλο και πιο αναγκαίος, ήδη από τη δεκαετία του 1960, εξαιτίας της τεράστιας τεχνολογικής προόδου και στις δύο περιοχές. Τελικά έλαβε χώρα το 1975, με ταυτόχρονη δημιουργία δύο κύκλων σπουδών, του Ηλεκτρονικού και του Ενεργειακού Μηχανικού.

Το 1982 η Σχολή Ηλεκτρολόγων Μηχανικών μετονομάστηκε σε Τμήμα Ηλεκτρολόγων Μηχανικών, ενώ το 1984 δημιουργήθηκε ακόμα ένας κύκλος σπουδών, αυτός του Μηχανικού Υπολογιστών και Πληροφορικής. Στη σημερινή του μορφή μετατράπηκε το Τμήμα το 1991, όταν και μετονομάστηκε σε Τμήμα

Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, το δε υπάρχον πρόγραμμα σπουδών θεσμοθετήθηκε το 1993 προσφέροντας τέσσερις κατευθύνσεις σπουδών, ως εξής:

1. Ηλεκτρονικής και Συστημάτων
2. Πληροφορικής
3. Επικοινωνιών
4. Ενέργειας

Τέλος, από το 1982 όλα τα Τμήματα των Ανώτατων Εκπαιδευτικών Ιδρυμάτων της χώρας (συνεπώς και το Εθνικό Μετσόβιο Πολυτεχνείο) υποχρεούνται να είναι οργανωμένα σε Τομείς με συγγενέστερα αντικείμενα, έτσι ώστε να γίνεται καλύτερος συντονισμός της εκπαιδευτικής διαδικασίας και έρευνας. Έτσι, λοιπόν, ιδρύθηκαν αρχικά τρεις Τομείς, στους οποίους και μοιράστηκαν το προσωπικό, καθώς και οι εκπαιδευτικές και ερευνητικές δραστηριότητες του Τμήματος. Από το ακαδημαϊκό έτος 2000-2001 δε, έγινε αναδιοργάνωση του Τμήματος σε επτά Τομείς.

### **3.1.2 Τομείς:**

- ✓ Τομέα Ηλεκτρομαγνητικών Εφαρμογών Ηλεκτροοπτικής και Ηλεκτρονικών Υλικών
- ✓ Τομέα Συστημάτων Μετάδοσης Πληροφορίας και Τεχνολογίας Υλικών
- ✓ Τομέα Σημάτων Ελέγχου και Ρομποτικής
- ✓ Τομέα Τεχνολογίας Πληροφορικής και Υπολογιστών
- ✓ Τομέα Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής
- ✓ Τομέα Ηλεκτρικής Ισχύος
- ✓ Τομέα Ηλεκτρικών Βιομηχανικών Διατάξεων και Συστημάτων Αποφάσεων

## 3.2 Εκπαιδευτική διαδικασία και Πρόγραμμα Σπουδών

### 3.2.1 Στόχοι του Εκπαιδευτικού Προγράμματος Σπουδών

Το πρόγραμμα σπουδών του τμήματος καθορίζεται από την Επιτροπή Προπτυχιακών Σπουδών του Τμήματος και στοχεύει κυρίως στην παροχή στους φοιτητές όλων των απαραίτητων και απαιτούμενων γνώσεων για την επαγγελματική αντιμετώπιση των σύγχρονων επαγγελματικών προβλημάτων, αλλά και μια επιτυχή μεταπτυχιακή εξειδίκευση υψηλού επιπέδου. Η άμεση και ουσιαστική αναπροσαρμογή του που πραγματοποιείται κατά καιρούς αντανakλά την διαρκή προσπάθεια του Τμήματος για εκσυγχρονισμό και ανανέωση, και προς αυτό το σκοπό συντέλεσαν η εισαγωγή νέων μαθημάτων και η προσθήκη τεσσάρων κατευθύνσεων σπουδών το 1995 και η κατάργηση και αναπροσαρμογή μαθημάτων το 2000.

### 3.2.2 Μαθήματα

Τα μαθήματα χωρίζονται σε δυο κατηγορίες:

1. Μαθήματα κορμού που διδάσκονται στη διάρκεια των πρώτων πέντε (5) εξαμήνων και είναι **κοινά** για όλους τους φοιτητές.
2. Μαθήματα των Ροών τα οποία επιλέγονται από τους φοιτητές ανάλογα με την κατεύθυνση σπουδών που θα επιλέξουν.

Επίσης, όλα τα μαθήματα διακρίνονται σε υποχρεωτικά, κατ' επιλογήν υποχρεωτικά και προαιρετικά, ανεξάρτητα της παραπάνω κατηγοριοποίησής τους.

1. Υποχρεωτικά είναι τα θεμελιώδη μαθήματα που θεωρούνται απαραίτητα για να μπορέσει να αποκτήσει ο φοιτητής το αναγκαίο γνωστικό υπόβαθρο του Ηλεκτρολόγου Μηχανικού και Μηχανικού Υπολογιστών και να προετοιμαστεί κατάλληλα για την εξειδίκευσή του σε μια από τις εμβασθύνσεις που του προσφέρει το Τμήμα.
2. Κατ' επιλογήν υποχρεωτικά είναι εκείνα τα μαθήματα που υλοποιούν την εξειδίκευση της προτίμησης του φοιτητή και του δίνουν τα εφόδια για παραπέρα σπουδές ή επιστημονική δραστηριότητα.
3. Προαιρετικά τέλος, είναι μαθήματα τα οποία μπορεί να επιλέξει προς παρακολούθηση ο φοιτητής ελεύθερα, κατά την κρίση και επιθυμία του, για να διευρύνει το πεδίο των γνώσεών του. Για να θεωρηθεί όμως ότι ο φοιτητής διδάχθηκε το μάθημα, πρέπει να έχει εξεταστεί με επιτυχία σ' αυτό.
4. Ξένες γλώσσες που διδάσκονται σήμερα στο Ε.Μ.Π. είναι τέσσερις:
  - Αγγλικά
  - Γαλλικά
  - Γερμανικά
  - Ιταλικά

Ο κύκλος σπουδών στις ξένες γλώσσες διαρκεί τέσσερα εξάμηνα και αντιστοιχεί με ένα υποχρεωτικό μάθημα του προγράμματος σπουδών. Ο βαθμός του δε συνυπολογίζεται στο βαθμό διπλώματος (Η ρύθμισή αυτή οφείλουμε να πούμε ότι ισχύει για τους φοιτητές του Τμήματος που εισήχθησαν σε αυτό από το ακαδημαϊκό έτος 1997-98 και μετά).

### **3.2.3 Διάρκεια των Σπουδών**

Σύμφωνα με το σύστημα σπουδών του Ε.Μ.Π., η φοίτηση στο Πολυτεχνείο διαρκεί πέντε (5) χρόνια. Κάθε χρόνος χωρίζεται σε δύο περιόδους ή εξάμηνα: το χειμερινό (Σεπτέμβριος - Ιανουάριος) και το εαρινό (Φεβρουάριος - Ιούνιος). Από τα δέκα εξάμηνα σπουδών, τα εννέα πρώτα είναι αφιερωμένα στην παρακολούθηση μαθημάτων, ενώ το δέκατο στην εκπόνηση της διπλωματικής εργασίας.

### **3.2.4 Διάρθρωση του προγράμματος Σπουδών**

Οι σπουδαστές που εισέρχονται στο Τμήμα από το ακαδημαϊκό έτος 1993-94 και μετά ακολουθούν το σύστημα ροών μαθημάτων, το οποίο επιτρέπει ευρύ φάσμα επιλογής ειδিকেύσεων. Δεδομένου ότι το Τμήμα χορηγεί ενιαίο τίτλο σπουδών, όλοι οι σπουδαστές υποχρεούνται να αποκτήσουν ένα ελάχιστο πυρήνα γνώσεων σε όλες τις μείζονες γνωστικές περιοχές του κλάδου, κατά τη διάρκεια των πέντε (5) πρώτων κοινών εξαμήνων. Στη συνέχεια, για τα εξάμηνα 6ο μέχρι και το 9ο, το πρόγραμμα σπουδών είναι δομημένο σε ροές ειδίκευσης.

### **3.2.5 Ροές**

Οι ροές ειδίκευσης του προπτυχιακού προγράμματος σπουδών είναι οι ακόλουθες:

1. Ροή Υ: Υπολογιστικά Συστήματα
2. Ροή Λ: Λογισμικό Η/Υ
3. Ροή Η: Ηλεκτρονική –Κυκλώματα – Υλικά
4. Ροή Δ: Επικοινωνίες και Δίκτυα Υπολογιστών
5. Ροή Τ: Κύματα και Τηλεπικοινωνίες
6. Ροή Σ: Σήματα, Έλεγχος και Ρομποτική
7. Ροή Ζ: Ηλεκτρομηχανική Μετατροπή Ενέργειας, Υψηλές Τάσεις και Εγκαταστάσεις
8. Ροή Ε: Συστήματα Ηλεκτρικής Ενέργειας
9. Ροή Ο: Διοίκηση και Απόφαση
10. Ροή Ι: Βιοϊατρική
11. Ροή Φ: Φυσική
12. Ροή Μ: Μαθηματικά

### 3.2.6 Απαιτήσεις για τη λήψη διπλώματος

Το πρόγραμμα σπουδών επιτρέπει την απόκτηση ενιαίου Διπλώματος Ηλεκτρολόγου Μηχανικού και Μηχανικού Υπολογιστών με την επιτυχή περάτωση εν γένει 59 μαθημάτων, με εξαίρεση έναν αριθμό φοιτητών οι οποίοι συνέπεσαν με τις αλλαγές σε αυτό το τελευταίο ακαδημαϊκό έτος και κατά περίπτωση είναι δυνατόν να λάβουν δίπλωμα με επιτυχή περάτωση 60 μαθημάτων. Από τα παραπάνω μαθήματα τα 34 (ή 35) είναι μαθήματα κορμού και τα 25 κατευθύνσεων και ροών. Πρέπει δε να υπακούουν στους ακόλουθους κανόνες:

- ☆ πρέπει να είναι οργανωμένα σαν συνδυασμός είτε
  1. τριών (3) πλήρων – κυρίων ροών και τεσσάρων (4) ελευθέρων μαθημάτων ή
  2. δύο (2) πλήρων – κυρίων ροών, δύο (2) μισών – συμπληρωματικών ροών και τριών (3) ελευθέρων μαθημάτων.
- ☆ τουλάχιστον 18 από τα μαθήματα αυτά πρέπει να ανήκουν στις ροές Υ, Λ, Η, Δ, Τ, Σ, Ζ και Ε.
- ☆ από τα ανθρωπιστικά μαθήματα μπορεί να επιλεγεί το πολύ ένα (1).

Η πλήρης ροή αποτελείται από επτά (7) μαθήματα, η μισή ροή από τέσσερα (4) μαθήματα, ενώ τα υποχρεωτικά μαθήματα και των πλήρων και των μισών ροών ορίζονται στους αντίστοιχους πίνακες του προγράμματος σπουδών.

### 3.2.7 Αλγόριθμος επιλογής μαθημάτων ροών

Ο αλγόριθμος επιλογής μαθημάτων ροών, έχοντας πάντοτε τις παραπάνω γενικές απαιτήσεις υπόψη, συνοψίζεται στα παρακάτω και είναι αυτός ακριβώς που υλοποιήθηκε στο σύστημα online εγγραφών:

1. Επιλέγεται μία από τις τέσσερις δυνατές κατευθύνσεις σπουδών.
  - i. Κατεύθυνση Ηλεκτρονικής και Συστημάτων
  - ii. Κατεύθυνση Πληροφορικής
  - iii. Κατεύθυνση Επικοινωνιών
  - iv. Κατεύθυνση Ενέργειας
2. Για τη συγκεκριμένη κατεύθυνση επιλέγεται ένας από τους επιτρεπόμενους συνδυασμούς ροών του Πίνακα Ι του Οδηγού Σπουδών (σελ. 53) και που έχει υλοποιηθεί με κατάλληλο φιλικό προς το χρήστη τρόπο στο σύστημα εγγραφών.
3. Διευρύνεται το πλάτος σπουδών του φοιτητή, έτσι ώστε τελικά να έχει επιλέξει είτε
  - i. 2 πλήρεις – κύριες ροές και 2 μισές – συμπληρωματικές ροές, είτε
  - ii. 3 πλήρεις – κύριες ροές

Υπάρχει το δικαίωμα **μία μόνο φορά** στη διάρκεια των σπουδών του ο φοιτητής να αλλάξει την επιλογή κατευθύνσεων και/ή ροών που πραγματοποιεί στο 6<sup>ο</sup> εξάμηνο. Επιτρέπεται η εγγραφή το πολύ σε 13 μαθήματα ανά εξάμηνο από τα οποία όχι περισσότερα από 8 να ανήκουν στο τρέχον ονομαστικό εξάμηνο εγγραφής και άνω.

### **3.2.8 Διπλωματική Εργασία**

Η Διπλωματική Εργασία είναι μια εκτεταμένη αναλυτική εργασία που εκπονείται από τους τελειόφοιτους φοιτητές στο τέλος των σπουδών τους, με σκοπό την ολοκλήρωση των γνώσεών τους και την παρουσίαση των ικανοτήτων τους. Κάθε φοιτητής είναι υποχρεωμένος να επιλέξει μία ροή στο γενικότερο γνωστικό αντικείμενο της οποίας θα εκπονήσει τη διπλωματική εργασία του. Τυπικά η εκπόνηση της διπλωματικής εργασίας ξεκινάει στο τέλος του 9<sup>ου</sup> εξαμήνου σπουδών, υπό την προϋπόθεση ότι ο φοιτητής δεν οφείλει περισσότερα των επτά (7) μαθημάτων του κανονικού προγράμματος σπουδών.

### **3.2.9 Βαθμολογία – βαθμός διπλώματος**

Η βαθμολογία σε όλα τα μαθήματα εκφράζεται με την κλίμακα: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 και 10, χωρίς τη χρήση κλασματικού μέρους και με βάση επιτυχίας το βαθμό 5, ενώ για τη διπλωματική εργασία, η βάση επιτυχίας είναι το 5,5, οπότε και επιτρέπεται η χρήση μισού βαθμού.

Οι φοιτητές βαθμολογούνται μόνο στα μαθήματα που έχουν δηλώσει στις αιτήσεις εγγραφής τους, ενώ η γραμματεία οφείλει να ανακοινώνει εγκαίρως τις καταστάσεις με τους εγγεγραμμένους φοιτητές ανά μάθημα. Οι καθηγητές, δε, δίνουν βαθμούς μόνο σε όσους φοιτητές αναφέρονται στα δελτία βαθμολογίας, τα οποία και συντάσσονται ανάλογα με τις καταχωρημένες εγγραφές. Τέλος, μαθήματα που δεν έχουν δηλωθεί στην αίτηση εγγραφής δεν αναφέρονται στην αναλυτική βαθμολογία του φοιτητή. Από τα παραπάνω καθίσταται άμεσα κατανοητή η σπουδαιότητα και η αναγκαιότητα ενός πλήρους μηχανογραφημένου συστήματος online εγγραφών, ελέγχων και αξιολόγησής των εγγραφών αυτών, προκειμένου να είναι εφικτή η εύρυθμη και ομαλή λειτουργία του Τμήματος.

### **3.2.10 Αριθμητικά στοιχεία από το πρόγραμμα σπουδών**

Ακολουθούν ορισμένα χρήσιμα στατιστικά στοιχεία από το πρόγραμμα σπουδών, τα οποία αναφέρονται και χρησιμοποιούνται στο σύστημα online εγγραφών, όπως αυτά αναφέρονται στον σημερινό Οδηγό Σπουδών του Τμήματος:

- Ο συνολικός αριθμός μαθημάτων που απαιτούνται για την απονομή Διπλώματος H.M. & M.Y. είναι 59. Από αυτά τα 34 είναι μαθήματα κορμού και τα 25 είναι μαθήματα κατευθύνσεων και ροών.
- Η μέση διάρκεια σπουδών ανέρχεται σε 241,8 ώρες και κατανέμεται ως εξής: στα μαθήματα κορμού αντιστοιχούν κατά μέσο όρο 148 ώρες (61,2%), ενώ στις κατευθύνσεις αναλογούν κατά μέσο όρο 93,8 ώρες (38,8%).



- Τα γενικά μαθήματα κορμού είναι 13 ή 14, ανάλογα με το εάν ο φοιτητής επιλέξει στο 4<sup>ο</sup> εξάμηνο γενικό ή ηλεκτρολογικό μάθημα, οπότε οι αντίστοιχες ώρες διδασκαλίας είναι 60 ή 64 (24,8% ή 26,5% αντίστοιχα).
- Τα ηλεκτρολογικά μαθήματα κορμού είναι 21 ή 20 και οι αντίστοιχες ώρες διδασκαλίας 88 ή 84 (36,4% ή 34,7%).
- Τα μαθήματα ειδίκευσης είναι 25, από τα οποία τα 12 είναι υποχρεωτικά των κατευθύνσεων, ενώ τα 13 είναι επιλογές των ροών. Για τα υποχρεωτικά μαθήματα των κατευθύνσεων απαιτούνται κατά μέσο όρο 49,2 ώρες (20,3%), ενώ άλλες κατά μέσο όρο 44,6 ώρες (18,5%) αντιστοιχούν στα μαθήματα επιλογής των ροών και στα ελεύθερα μαθήματα .
- Η μέση διάρκεια μαθήματος είναι 4,10 ώρες. Αναλυτικότερα, αυτή αναλύεται σε: γενικά μαθήματα κορμού 4,59 ώρες, ηλεκτρολογικά μαθήματα κορμού 4,20 ώρες, υποχρεωτικά μαθήματα κατευθύνσεων 4,10 ώρες και μαθήματα επιλογής ροών 3,43 ώρες.
- Ο μέσος όρος διδασκαλίας εβδομαδιαίως είναι 26,9 ώρες και αναλύεται σε εξάμηνα κορμού 29,6 ώρες και εξάμηνα κατευθύνσεων 23,5 ώρες.
- Τα μαθήματα πληροφορικής του κορμού κυμαίνονται από 5 έως 6 (23 έως 27 ώρες διδασκαλίας, ήτοι 9,5% έως 11,2%). Τα επιπλέον μαθήματα πληροφορικής που μπορεί να παρακολουθήσει κάποιος που ακολουθεί τη σχετική κατεύθυνση ανέρχονται το πολύ σε 18 και αντιστοιχούν σε 31 ώρες υποχρεωτικών μαθημάτων και 31,5 ώρες επιλογών (συνολικά 25,8 %).
- Οι καθηγητές του Τμήματος διδάσκουν συνολικά 150 μαθήματα μέσης διάρκειας 3,66 ωρών. Καθηγητές άλλων Τμημάτων του Ε.Μ.Π. διδάσκουν 41 μαθήματα μέσης διάρκειας 4,10 ωρών.
- Για τα ηλεκτρολογικά μαθήματα ροών και κατευθύνσεων ο φοιτητής παρακολουθεί ανάλογα με τις επιλογές του τις αναγραφόμενες στον ακόλουθο πίνακα ώρες:

Ροή	Ώρες μαθημάτων όταν πλήρης			Ώρες μαθημάτων όταν μισή		
	Υποχρεωτικά	Επιλογές	Σύνολο	Υποχρεωτικά	Επιλογές	Σύνολο
Υ	15	12,4	27,4	8	6,3	14,3
Λ	16	12,8	58,8	8	6,7	14,7
Η	14	14	58	8	6,8	14,8
Δ	15	12,8	27,8	8	6,5	14,5
Τ	20	12,4	32,4	11	6,7	17,7
Σ	19	13,2	32,2	11	6,8	17,8
Ζ	21	14,8	35,8	10	8,0	18,0
Ε	16	12,4	28,4	8	7,1	15,1
Ο	12	6,8	18,8	6	6,6	12,6
Ι	16	5,6	21,6	8	7,3	15,3

Επομένως, η μέση διάρκεια σπουδών ανά κατεύθυνση κυμαίνεται από:

Κατεύθυνση Ηλεκτρονικής και Συστημάτων:	90-102 ώρες
Κατεύθυνση Πληροφορικής:	94-101 ώρες
Κατεύθυνση Επικοινωνιών:	94-105 ώρες
Κατεύθυνση Ενέργειας:	98-110 ώρες

### **3.3 Ορισμός αναγκών**

Η μέχρι τώρα διαδικασία εγγραφής σε ένα ακαδημαϊκό εξάμηνο, η οποία περιγράφεται αναλυτικά παραπάνω, πραγματοποιούνταν σε καθορισμένο χρονικό διάστημα στα γκισέ του χώρου της γραμματείας του Τμήματος. Αμέτρητα προβλήματα και ταλαιπωρία ανέμεναν όσους εκ των πραγμάτων ήθελαν να σπουδάσουν στο Τμήμα και ήταν αναγκασμένοι να εγγραφούν σε κάποιο ακαδημαϊκό εξάμηνο. Ουρές πολυάριθμων ατόμων, εκνευρισμοί, ευτράπελα, πανικός γιατί δεν προλάβαιναν τις προθεσμίες, λάθη στις αιτήσεις εγγραφής, είναι λίγα μόνο από όσα συνέβαιναν με εξαιρετική συνέπεια κατά τις δύο περιόδους εγγραφών.

Ήταν, λοιπόν, επιτακτική η ανάγκη για μεταφορά της διαδικασίας εγγραφής των φοιτητών, από τα γκισέ, το ωράριο και τα γραφεία της γραμματείας, στον απέραντο ηλεκτρονικό «χώρο» του Internet και στην απεριόριστη πρόσβαση 24ώρες/24ώρο και για τις 14 ημέρες που διαρκούν επίσημα οι εγγραφές. Κι αυτό, γιατί πλέον η περιήγηση στο Internet για τον καθηγητή, τον φοιτητή και τον υπάλληλο της γραμματείας είναι εν έτη 2001 κάτι το εύκολο και γρήγορα προσβάσιμο, χάρη και στις νέες εγκαταστάσεις που παρέχει το Εθνικό Μετσόβιο Πολυτεχνείο στα μέλη της κοινότητας του εδώ και κάποιο καιρό. Βέβαια, εκ των ων ουκ άνευ θεωρείται η εισαγωγή προς ελεύθερη χρήση από τους ενδιαφερόμενους υλικοτεχνικού εξοπλισμού, μέσω του οποίου θα μπορεί ανά πάσα στιγμή να προσπελαστεί το εν λόγω σύστημα.

Λαμβάνοντας, δε, υπόψην όλα όσα αναφέρθηκαν στην ενότητα 3.2.9, σχετικά με την ιδιαίτερη σημασία μιας εγγραφής ενός φοιτητή στη φοιτητική του πορεία και στις συναλλαγές του με τη διοικητική λειτουργία του Τμήματος, είμαστε εύκολα σε θέση να δούμε τη σπουδαιότητα και την αναγκαιότητα ενός πλήρους τέτοιου online συστήματος. Η παραπάνω διαπίστωση αποτέλεσε και το έναυσμα για την υλοποίηση της ιδέας αυτού του συστήματος online εγγραφών. Καθίσταται απόλυτα σαφές πως οι διάφοροι ενδιαφερόμενοι έχουν ανάγκη ενός εύκολα προσβάσιμου «εργαλείου» το οποίο θα βοηθά τους υπαλλήλους της γραμματείας στη διαδικασία ελέγχου, πιστοποίησης και καταχώρησης της εγγραφής, θα παρέχει κάποια στατιστικά στοιχεία εγγραφών στους καθηγητές, **αλλά κυρίως** θα παρέχει την δυνατότητα στους φοιτητές να εγγράφονται γρήγορα και προπάντων εύκολα στα μαθήματα και την κατεύθυνση της επιλογής τους.

### 3.4 Ορισμός πρακτικού προβλήματος

Απαιτείται λοιπόν η δημιουργία ενός αξιόπιστου **24h/24h online web site**, όπου θα μπορεί οποιοδήποτε μέλος της ακαδημαϊκής κοινότητας του Τμήματος να προσπελάσει χρήσιμες, σε σχέση με τις εγγραφές στα εξάμηνα, αλλά και όχι μόνο γι' αυτές, πληροφορίες. Ειδικότερα, πρέπει να αντιμετωπιστεί το μείζον πρακτικό πρόβλημα της εγγραφής των φοιτητών σε ένα ακαδημαϊκό εξάμηνο, τόσο κατά τις χειμερινές, όσο και κατά τις θερινές δεκαπενθήμερες περιόδους εγγραφών.

Μέσω εύκολα κατανοητού interface με το χρήστη, απλού, κατανοητού και σαφούς κειμένου, και με τη χρήση εικόνων και πινάκων θα πρέπει κάθε κατηγορία χρηστών (βλ. και κεφάλαιο 3.7) να μπορεί να προσπελάσει εύκολα και γρήγορα την πληροφορία. Θα πρέπει επίσης κάθε χρήστης του συστήματος να είναι σε θέση να τη χρησιμοποιήσει μόνος του εύκολα και απροβλημάτιστα, διαφορετικά το όλο εγχείρημα θα επέφερε το αντίθετο από το θεμιτό αποτέλεσμα.

Απαραίτητη θεωρείται επίσης, η διεξοδική επίλυση όλων των προβλημάτων υλοποίησης του εν λόγω εγχειρήματος, σε οποιοδήποτε επίπεδο κι αν ορίζονται αυτά, από το γραφειοκρατικό μέχρι το καθαρά τεχνικό κομμάτι της.

### 3.5 Η μέχρι σήμερα Λύση

Μέχρι σήμερα, τα «εργαλεία» που χρησιμοποιούνταν ευρέως από τους φοιτητές κυρίως, αλλά και από το διδακτικό προσωπικό και το προσωπικό της γραμματείας κατά την περίοδο των εγγραφών ήταν το φυλλάδιο της αίτησης εγγραφής και ο έντυπος **Οδηγός Σπουδών** του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών.

Ο Οδηγός Σπουδών παρέχει ικανοποιητική, όπως είδαμε παραπάνω, περιγραφή του Προγράμματος Σπουδών, του αλγόριθμου επιλογής κατεύθυνσης, ροών και δυνατών μαθημάτων σε αυτές, καθώς και όλων των μαθημάτων που προσφέρονται ανάλογα με τα εξάμηνα (για τα μαθήματα κορμού) ή τις ροές (για τα μαθήματα κατεύθυνσης).

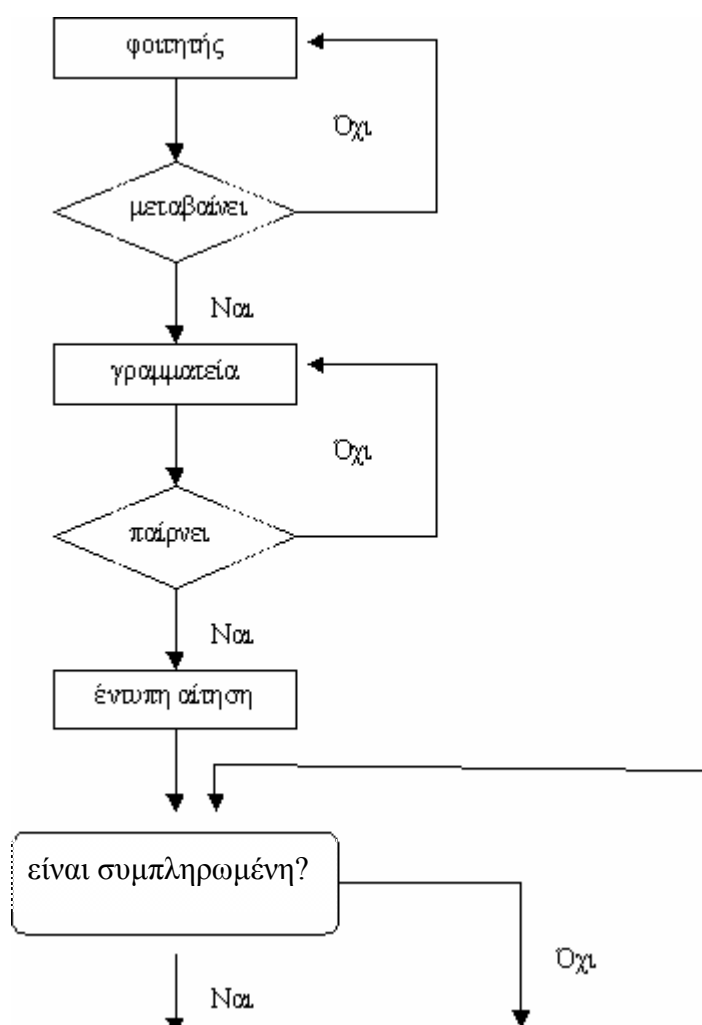
Το έγγραφο της αίτησης εγγραφής, όμως, είναι μάλλον δυσανάγνωστο, με συγκεχυμένες πληροφορίες, συγκεντρωμένες όλες για όλους. Ακόμα και αν κάποιος φοιτητής δεν βρισκόταν σε ονομαστικό εξάμηνο εγγραφής που να απαιτεί εγγραφή/επιλογή μαθημάτων ροών/κατευθύνσεων, ήταν υποχρεωμένος να «περιδιαβεί» ανάμεσα σε όλες τις πληροφορίες της αίτησης που αναφέρονται σε αυτά. Πέρα από αυτό, απαιτούσε εισαγωγή, με τον προσωπικό γραφικό χαρακτήρα του καθενός, προσωπικών στοιχείων του φοιτητή, με ανάλογη πιθανότητα κακογραμμένων και δυσανάγνωστων από τη γραμματεία στοιχείων, αλλά και λάθους κατά την ίδια την εισαγωγή. Επίσης «με το χέρι» αναγραφόταν από το φοιτητή η ημερομηνία κατάθεσης της αίτησης, με αποτέλεσμα να παρουσιάζονται και σε αυτό το σημείο λάθη, και παραβλέψεις, καθώς η εισαγωγή αυτού του στοιχείου ήταν αποκομμένη από το υπόλοιπο έγγραφο στο τέλος του. Τέλος, η επιβεβαίωση του ατόμου που υπέβαλε την αίτηση γινόταν μόνο με την υπογραφή του, χωροταξικά στο τέλος και αυτή του εγγράφου, με αποτέλεσμα και αυτή να ξεχνιέται εύκολα και να

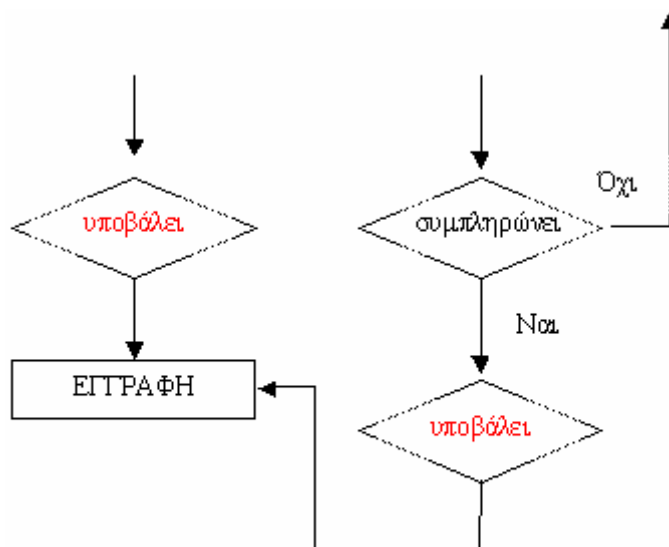
καλείται συνήθως ο φοιτητής εκ νέου στη γραμματεία για διορθώσεις την «εβδομάδα διορθώσεων».

Είναι φανερό, λοιπόν, πως η παραπάνω υλοποίηση παρουσιάζει πολλά μειονεκτήματα σε σχέση με την ηλεκτρονική, αυτοματοποιημένη υλοποίησή της και κάτι παραπάνω από προφανές πως οι παραπάνω μέθοδοι δεν καλύπτουν αποτελεσματικά εκείνες τις ανάγκες των χρηστών για εγγραφή σε ακαδημαϊκό εξάμηνο σπουδών του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών.

Περαιτέρω, οι παραπάνω χειρόγραφες αιτήσεις μεταφέρονταν στο υπάρχον υπολογιστικό σύστημα της Γραμματείας από το προσωπικό της, το οποίο συμπλήρωνε τις φόρμες του συστήματος αυτού, ακολουθώντας την εξής τεχνική: από τη μία πλευρά του γραφείου διάβαζαν τις έντυπες αιτήσεις των φοιτητών και από την άλλη εισήγαγαν με τη χρήση του ποντικιού τα σωστά πεδία στο σύστημα, με ό,τι μπορεί αυτό να συνεπάγεται για λάθη, παραλείψεις ή καθυστερήσεις.

Συνοψίζοντας, το flowchart μιας τυπικής διαδικασίας εγγραφής με το υπάρχον σύστημα θα μπορούσε να παρασταθεί χονδρικά ως εξής:





### 3.6 Λύση πρακτικού προβλήματος

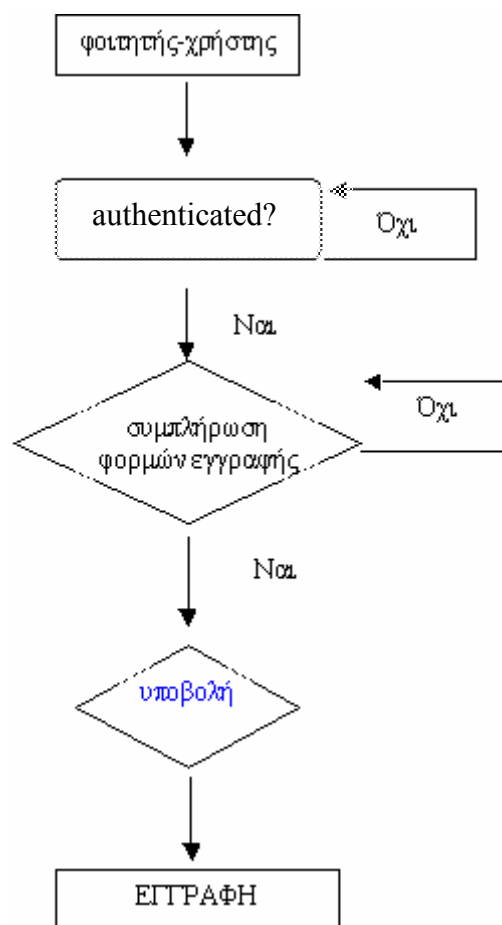
Λαμβάνοντας υπόψη όλα τα παραπάνω προέκυψε ο σχεδιασμός ενός ολοκληρωμένου συστήματος μηχανογράφησης των εγγραφών, με το κύριο μέρος του να αποτελείται από ένα web site στο world wide web (www). Ακολουθήθηκε μια συγκεκριμένη διαδικασία για την ανάπτυξή του, η οποία για να υλοποιηθεί σωστά, απαιτήσε την ανάπτυξη ενός πλήθους συστημάτων και υποσυστημάτων και την επίλυση των διαφόρων υποπροβλημάτων που προέκυψαν στην πορεία και εκ των οποίων άλλα είχαν προβλεφθεί και άλλα ήταν αδύνατο να εκτιμηθούν a priori.

Βασικός άξονας του συστήματος είναι η κατασκευή απλών και δυναμικών στη συντριπτική πλειοψηφία τους web pages, με μεγάλη έμφαση στο περιεχόμενό τους, μιας και εκεί θα δοθεί το μεγαλύτερο βάρος, στην κατανόηση, δηλαδή, από πλευράς χρηστών του τρόπου εγγραφής στα μαθήματα/κατευθύνσεις, αλλά και της λειτουργίας ολόκληρου του web site, προς άντληση της κατάλληλης πληροφορίας εύκολα και γρήγορα.

Εύκολα προσβάσιμα και κατανοητά περιεχόμενα ανά ομάδες χρηστών είναι διαθέσιμα, με διαφορετικά επίπεδα λειτουργικότητας και δικαιωμάτων σε καθεμία. Η επεκτασιμότητα του όλου έργου είναι προφανής, ενώ προφανώς τροποποίηση ή αλλαγή του περιεχομένου του web site από μη εξουσιοδοτημένο άτομο δεν είναι δυνατή και θεμιτή. Ο εξουσιοδοτημένος χρήστης μπορεί είτε να επανέλθει όσες φορές θέλει σε κάποιο τμήμα του, όπου δεν απαιτείται εισαγωγή πληροφορίας από αυτόν, αλλά μόνο άντληση (π.χ. τα τμήματα που αναφέρονται στην κατηγοριοποίηση των μαθημάτων του Τμήματος), όπως στην περίπτωση των φοιτητών-χρηστών, είτε να έχει πλήρη πρόσβαση σε ολόκληρο το σύστημα, όπως στην περίπτωση του προσωπικού της γραμματείας που ασκεί εποπτικό και ουσιαστικό ρόλο στη διαδικασία των εγγραφών.

Επίσης, να σημειώσουμε ότι το όλο σύστημα απευθύνεται σε χρήστες οι οποίοι γνωρίζουν την Ελληνική γλώσσα, καθότι είναι γραμμένο εξ ολοκλήρου σε αυτή, και διαθέτουν στοιχειώδεις γνώσεις χρήσης προσωπικών υπολογιστών, όπως άλλωστε θεωρείται ότι ισχύει σήμερα για το προσωπικό της γραμματείας και τους φοιτητές του Τμήματος.

Τέλος, αν θέλαμε να αντιπαραθέσουμε το flowchart της σημερινής διαδικασίας εγγραφών της προηγούμενης ενότητας, με κάποιο αντίστοιχο για τη διαδικασία στο ηλεκτρονικό σύστημα εγγραφών, θα καταλήγαμε για το τελευταίο στο εξής:



Η απλοποίηση της διαδικασίας είναι εμφανής από την πρώτη ματιά, αναλογιζόμενοι και την ηλεκτρονική μορφή της.

### 3.7 Ομάδες Χρηστών ενός τέτοιου Συστήματος

Όπως προαναφέραμε, ένα τέτοιο σύστημα θα παρείχε χρήσιμες υπηρεσίες σε τρία διαφορετικά είδη χρηστών, ανάλογα πάντα με τις ανάγκες τους. Ειδικότερα:

- ο **Φοιτητές**: Οι φοιτητές είναι οι κύριοι ενδιαφερόμενοι, αφού αυτοί καλούνται να αντεπεξέλθουν στη σημερινή διαδικασία εγγραφών και αυτοί θα χρησιμοποιήσουν κατά κόρον το εν λόγω νέο σύστημα. Προφανώς ευχρηστία,

απλότητα και ταχύτητα του συστήματος αποτελούν εκ των ων ουκ άνευ για αυτού, καθώς και ευκολία πρόσβασης.

- ο **Καθηγητές-Διδακτικό προσωπικό:** Οι διδάσκοντες θα ήθελαν από την πλευρά τους να μπορούν να αντλήσουν στατιστικά στοιχεία για τον αριθμό των φοιτητών που εγγράφονται κάθε εξάμηνο στα μαθήματά τους, ή/και γενικότερα στατιστικά στοιχεία για κάποιο γνωστικό αντικείμενο που τους ενδιαφέρει. Και όλα αυτά θα επιθυμούσαν να γίνονται απρόσκοπτα και γρήγορα.
- ο **Γραμματεία:** Οι υπάλληλοι της γραμματείας θα μπορούσαν μέσω του συστήματος αυτού να απαλλαγούν από ένα μεγάλο βάρος και φόρτο εργασίας κατά τη διάρκεια των περιόδων εγγραφών και να μειώσουν σημαντικά τις καθυστερήσεις στην εργασία τους στον τομέα αυτό.

Τέλος, εκ των πραγμάτων, απαιτείται και η ύπαρξη ενός τουλάχιστον **τεχνικού διαχειριστή** του όλου συστήματος, ο οποίος μέσω του ίδιου κεντρικού interface θα είναι σε θέση να προσπελάσει, να συντηρήσει και να διαχειριστεί εν γένει οποιαδήποτε παράμετρό του.

### **3.8 Τελικό Ζητούμενο**

Το τελικό ζητούμενο στο συγκεκριμένο πρόβλημα είναι η κατασκευή ενός πλήρους online συστήματος ηλεκτρονικών εγγραφών (Σ.Η.Ε.), το οποίο με όσο πιο σαφή και εύκολο τρόπο γίνεται να βοηθά τους φοιτητές - καθηγητές - λοιπούς ενδιαφερομένους να αντλήσουν την κατάλληλη πληροφορία και να αλληλεπιδράσουν με αυτό. Απαιτείται, δε, συνεργασία του συστήματος αυτού με τα υπάρχοντα (υπο-)συστήματα που χρησιμοποιούνται για το σκοπό αυτό, όπως το φοιτητολόγιο του Ε.Μ.Π. και η υπάρχουσα μηχανογράφηση της Γραμματείας του Τμήματος Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών.





# 4

## *ΑΝΑΛΥΣΗ της ΛΥΣΗΣ*

### *4.1 Χρήση του Δικτύου*

Όπως αναφέρθηκε προηγουμένως, το σύστημα το οποίο θα χρησιμοποιηθεί για την εγγραφή των φοιτητών και την άντληση στατιστικών στοιχείων θα πρέπει πρωτίστως να είναι εύκολα προσβάσιμο. Το www καλύπτει την ανάγκη αυτή. Το σύστημα είναι προσβάσιμο μέσω δικτύου στην ηλεκτρονική διεύθυνση:

<http://www.dblab.ece.ntua.gr/~she>

Με δεδομένο ότι στο τμήμα H.M. & M.Y., αλλά και γενικότερα στο χώρο του Πολυτεχνείου, η χρήση του Internet είναι κοινός τόπος, αλλά υπάρχουν και τα απαραίτητα στοιχεία (ταχύτατο οπτικό και ηλεκτρονικό δίκτυο δεδομένων, σύνδεση με το www, δυνατοί servers, www administrators, βάσεις δεδομένων, DBAs) για την σωστή ανάπτυξη ενός τέτοιου προγράμματος, η επιλογή της πλατφόρμας του δικτύου ήταν ίσως η πιο αποτελεσματική και κρίνεται ιδιαίτερα αποτελεσματική. Εξάλλου, η όλο και αυξανόμενη χρήση και λειτουργικότητα του www σε όλο τον κόσμο και στη χώρα μας, καθιστά το σύστημα πολύ εύκολα προσβάσιμο για τον καθένα από οποιοδήποτε μέρος.

Τέλος, σε πολλά προηγμένα εκπαιδευτικά ιδρύματα του εξωτερικού υπάρχει ήδη υλοποιημένο ένα τέτοιο σύστημα στο web, και ήταν εν γένει κάτι το οποίο έλειπε από το ελληνικό Πολυτεχνείο. Παραδείγματα τέτοιων συστημάτων αποτελούν:

Michigan Institute of Technology, MIT:

<http://student.mit.edu/>

University of California, Riverside:

<http://www.students.ucr.edu/paws/tour/enrollment.html>

University of New Hampshire:

[https://webcat.unh.edu/sis\\_plsql/twgkwbis.P\\_WWWLogin](https://webcat.unh.edu/sis_plsql/twgkwbis.P_WWWLogin)

## **4.2 Χρήση Βάσης Δεδομένων**

Οι πληροφορίες που υπάρχουν στο σύστημα εγγραφών και μπορούν φυσικά να προσπελαστούν από τους κάθε λογής authenticated χρήστες του, είναι όλες εκείνες που αναφέρονται στον έντυπο Οδηγό Σπουδών και έχουν να κάνουν με τη διαδικασία εγγραφών στο Τμήμα, αλλά και άλλες πρόσθετες, όπως πληροφορίες για το διδακτικό προσωπικό. Επίσης, καθώς η διαδικασία εγγραφής αυτή καθ' αυτή απαιτεί την αλληλεπίδραση με τους χρήστες και την εισαγωγή στοιχείων από αυτούς, τα οποία είναι απαραίτητο να φυλάσσονται σε ασφαλή χώρο και να προωθούνται προς επεξεργασία στη βάση δεδομένων του φοιτητολογίου, οδηγούμαστε εκ των πραγμάτων σε χρήση αποδοτικών τρόπων αποθήκευσης πληροφορίας.

Όπως, λοιπόν, γίνεται αντιληπτό, η χρήση μιας βάσης δεδομένων για την ασφαλή αποθήκευση όλης αυτής της πληροφορίας είναι απαραίτητη. Η χρήση όμως της βάσης δεδομένων παρέχει συνάμα και όλα τα πλεονεκτήματα μιας τέτοιας επιλογής όπως:

- Άρτια οργάνωση της πληροφορίας που συνεπάγεται και εύκολη προσπέλαση, προσθήκη, ενημέρωση, αφαίρεση ή τροποποίησή της.
- Ασφάλεια δεδομένων και δοσοληψιών.
- Ταχύτητα ανάκτησης πληροφορίας κάτι που συνεπάγεται και αποδοτικότητα του όλου συστήματος.
- Διαχωρισμός δεδομένων και προγράμματος, γεγονός το οποίο έχει σαν συνέπεια να μην επηρεάζεται η λογική του συστήματος από τη λογική των δεδομένων, να είναι ανεξάρτητο και συνακόλουθα πιο ευέλικτο.

## **4.3 Χρήση τεχνολογίας JSP**

Όλα τα παραπάνω δεν θα μπορούσαν να συμβάλλουν στην αποδοτική λειτουργία του συστήματος εγγραφών, αν δεν υπήρχε ένας ακόμα καθοριστικός παράγοντας στην όλη σχεδίαση. Και αυτός δεν είναι άλλος από τη χρήση δυναμικών web σελίδων με την τεχνολογία JSP, που εγγυάται ταχύτερη πρόσβαση στην πληροφορία που ενδιαφέρει τον χρήστη και μόνο εκείνη, καθώς και προσωποποιημένη κατά κάποιον τρόπο πρόσβαση, παρουσιάζοντάς του κάθε φορά στοιχεία και περιεχόμενο, ανάλογα τις προηγούμενες επιλογές που έχει διαπράξει στο σύστημα ο ίδιος. Ακόμα για τον προγραμματιστή παρέχει, όπως είδαμε, διαχωρισμό business logic από presentation

logic σε επίπεδο που ο ίδιος αποφασίζει, μετατρέποντας τη διαδικασία παραγωγής και σχεδίασης σε μία σαφή και ξεκάθαρη εργασία.

#### 4.4 Περιγραφή του Προγράμματος Σπουδών

Είναι λογικό το σύστημά μας να παρέχει όλες εκείνες τις πληροφορίες που θεωρούνται απαραίτητες, έτσι ώστε οποιοσδήποτε φοιτητής να είναι σε θέση να πραγματοποιήσει με επιτυχία την εγγραφή του στο εξάμηνο που επιθυμεί/μπορεί από το πρόγραμμα σπουδών. Παρέχει λοιπόν, από κάθε σημείο της εφαρμογής τη δυνατότητα στο χρήστη να μεταφερθεί στο αντίστοιχο κείμενο βοήθειας που εξηγεί τη λογική και τις δυνατότητες της διαδικασίας εγγραφής. Επίσης, συγκεκριμένο τμήμα του προσφέρει τα αντίστοιχα επεξηγηματικά τμήματα του Οδηγού Σπουδών (Ο.Σ.) σε ηλεκτρονική μορφή, έτσι ώστε να βρίσκονται σε απόσταση μερικών μόνο click με το ποντίκι κατά τη διαδικασία εγγραφής των φοιτητών, λειτουργώντας το ίδιο συμβουλευτικά όπως ο ίδιος ο έντυπος Ο.Σ..

#### 4.5 Αναζήτηση - Εγγραφή Μαθημάτων ακαδημαϊκού εξαμήνου

Το πλέον σημαντικό στοιχείο που θα ενδιέφερε τους περισσότερους χρήστες του ΣΗΕ είναι η γρήγορη και αποδοτική αναζήτηση των μαθημάτων των ακαδημαϊκών εξαμήνων και η εύκολη εγγραφή σε αυτά. Προς το σκοπό αυτό, το κύριο τμήμα της εφαρμογής ασχολείται με την παροχή τέτοιου είδους πληροφορίας και δυνατότητας στο χρήστη του. Θα πρέπει να σημειώσουμε εδώ ότι σε κάθε σελίδα που παρουσιάζονται στοιχεία πληροφόρησης από τη βάση δεδομένων, έχει υλοποιηθεί μία εύχρηστη μπάρα πλοήγησης:



Εικόνα 4.5.α

, όπου όπως παρατηρούμε ο χρήστης έχει τη δυνατότητα για έξι (6) διαφορετικά set πλοήγησης («**αρχική σελίδα**», «**προηγούμενη σελίδα**», «**προηγούμενο record**», «**επόμενο record**», «**επόμενη σελίδα**» και «**τελευταία σελίδα**») και δύο (2) διαφορετικά είδη πλοήγησης («**πλοήγηση records**» και «**μεμονωμένη απεικόνιση record**», ανάλογα του mode στο οποίο βρίσκεται κάθε χρονική στιγμή). Αναλυτικότερα και ξεκινώντας από το υποσύστημα που προορίζεται για πρόσβαση από τους προπτυχιακούς φοιτητές υπάρχουν δυνατότητες για:

- παρουσίαση μόνο των μαθημάτων κορμού, διατεταγμένα αλφαβητικά ανά εξάμηνο, τόσο σε μεμονωμένη απεικόνιση συγκεκριμένου μαθήματος, όσο και σε απεικόνιση όλων των μαθημάτων μαζί.

- παρουσίαση μόνο των μαθημάτων ροών, διατεταγμένα αλφαβητικά ανά εξάμηνο και ροή, τόσο σε μεμονωμένη απεικόνιση συγκεκριμένου μαθήματος, όσο και σε απεικόνιση όλων των μαθημάτων μαζί.
- παρουσίαση μόνο των υποχρεωτικών μαθημάτων, είτε αυτά ανήκουν στον κορμό, είτε στις ροές, διατεταγμένα αλφαβητικά ανά εξάμηνο και (εφόσον υπάρχει) κατεύθυνση/ροή, τόσο σε μεμονωμένη απεικόνιση συγκεκριμένου μαθήματος, όσο και σε σελίδα πλοήγησης όλων των μαθημάτων μαζί.
- παρουσίαση μόνο των μαθημάτων επιλογής, διατεταγμένα αλφαβητικά ανά εξάμηνο και (εφόσον υπάρχει) κατεύθυνση/ροή, είτε αυτά ανήκουν στον κορμό, είτε στις ροές. Και εδώ είναι δυνατή διπλή απεικόνισή τους σε επίπεδο συγκεκριμένου μαθήματος και σε επίπεδο συνόλου.
- παρουσίαση σε νέο παράθυρο για εύκολη και ανεξάρτητη πρόσβαση ανά πάσα στιγμή των μαθημάτων συγκεκριμένου εξαμήνου, υλοποιημένου με εύκολο μενού με καρτέλες.
- εισαγωγή επιλεγμένων από το φοιτητή μαθημάτων, ανάλογα το ονομαστικό εξάμηνο εγγραφής του και τους κανόνες που καθορίζει ο Οδηγός Σπουδών σε μια διαδικασία 3 + 1 βημάτων.
- εισαγωγή επιλεγμένης από το φοιτητή κατεύθυνσης σπουδών και ανάλογα δυνατού συνδυασμού ροών, με βάση τον παραπάνω πίνακα του Οδηγού Σπουδών, σε μια διαδικασία 4 + 1 βημάτων.
- ταυτόχρονη με την εισαγωγή, παρουσίαση καρτέλας μαθήματος με πλήρη περιγραφή του, στη μορφή του υπάρχοντος Οδηγού Σπουδών.

## 4.6 Τμήμα εγγραφής μαθημάτων

Ουσιαστικά πρόκειται για ένα από τα ημιαυτόνομα τμήματα της εφαρμογής ΣΗΕ, το οποίο επιτρέπει στο χρήστη της να εγγραφεί εύκολα, γρήγορα και με απλό τρόπο στα μαθήματα του ακαδημαϊκού εξαμήνου που επιθυμεί, προσομοιώνοντας τη μέχρι σήμερα γνωστή έντυπη αίτηση εγγραφής του Τμήματος. Η λογική της αίτησης αυτής έχει κατά το δυνατόν διατηρηθεί και αποτυπωθεί σε αυτό το σύστημα, ενώ έχει ακολουθηθεί μια διαδικασία 3 + 1 βημάτων, η οποία και προήλθε κυρίως από την μέχρι σήμερα εμπειρία που αποκομίσαμε από τη σκοπιά των φοιτητών που εγγράφονται στο Τμήμα.

Ο χρήστης καλείται αρχικά να επιλέξει το ονομαστικό εξάμηνο εγγραφής στο οποίο επιθυμεί να εγγραφεί κατά την περίοδο εγγραφών (**βήμα 1**). Με βάση αυτή την επιλογή του διαθέτει και ανάλογες δυνατότητες επιλογής μαθημάτων, σύμφωνα με τους κανόνες του Οδηγού Σπουδών. Έτσι, στη συνέχεια και αφού επιβεβαιώσει οπτικά την επιλογή του, μεταφέρεται στην - φαινομενικά όμοια για όλους και ανά χειμερινή ή θερινή περίοδο κατάλληλη - σελίδα επιλογής μαθημάτων. Στη σελίδα αυτή συμπληρώνει τα επιθυμητά μαθήματά του, ενώ ταυτόχρονα, διαφόρων ειδών συναρτήσεις ελέγχουν ως προς την ορθότητα τους τις επιλογές του (**βήμα 2**). Κατόπιν ο χρήστης μεταφέρεται σε μια άλλη σελίδα, όπου και παρουσιάζονται

συγκεντρωμένες οι μέχρι τώρα επιλογές μαθημάτων του (χωρίς να διαθέτει πλέον δυνατότητα αλλαγής τους κατά την παρούσα αρχικοποιημένη διαδικασία) και καλείται να συμπληρώσει τα προσωπικά του στοιχεία (**βήμα 3**). Και σε αυτό το βήμα πραγματοποιούνται διάφοροι έλεγχοι προς επιβεβαίωση της ορθότητας των εισαγομένων στοιχείων. Τέλος, ακριβώς πριν την τελική εισαγωγή των συνολικών στοιχείων του στη βάση, παρουσιάζεται μια σελίδα επιβεβαίωσης αυτών (**βήμα 4**) και ο χρήστης καλείται είτε να τα επιβεβαιώσει οριστικά και αμετάκλητα, πραγματοποιώντας με επιτυχία την εγγραφή του, είτε να τα απορρίψει ολοκληρωτικά, ξεκινώντας τη συναλλαγή του με την εφαρμογή ΣΗΕ από την αρχή, για λόγους εγκυρότητας και authentication των προσωπικών του στοιχείων.

#### **4.7 Τμήμα επιλογής κατεύθυνσης / ροών**

Το δεύτερο ημιαυτόνομο τμήμα της εφαρμογής ΣΗΕ αποτελείται από τη διαδικασία επιλογής κατεύθυνσης σπουδών και συνδυασμού ροών που θα ακολουθήσει ο φοιτητής. Ενσωματώνει όλη τη λογική του συστήματος των ροών και της επιλογής των συνδυασμών, επιτρέπει δε μέσα από τη διαδραστικότητά του, την κατανόηση και ευκολότερη επομένως επιλογή.

Ο χρήστης καλείται αρχικά να εισάγει τον Αριθμό Μητρώου του και τον ειδικό κωδικό του συστήματος (**βήμα 1**), τα οποία και διασταυρώνονται στη βάση<sup>5</sup>, έτσι ώστε να περιοριστεί το φορτίο αυτού του κομματιού, αλλά και αν αποφύγουμε λανθασμένες ή κακόβουλες εισαγωγές στοιχείων. Ακολούθως και εφόσον επιτύχει το προηγούμενο βήμα, καλείται να επιλέξει από έναν πίνακα επιθυμητή κατεύθυνση σπουδών (**βήμα 2**). Ανάλογα την επιλογή του και για κάθε μια από αυτές, εμφανίζεται ο γνωστός πίνακας με τους συνδυασμούς των ροών. Εκεί συμπληρώνει τον συνδυασμό που επιθυμεί (**βήμα 3**) (με δυνατότητα προς το παρόν να αλλάξει γνώμη και επιλογές) και κατόπιν υποβάλει τη φόρμα, οπότε και του παρουσιάζεται μια σελίδα (**βήμα 4**) με τις έως τώρα επιλογές του (χωρίς πλέον δυνατότητα διόρθωσης/αλλαγής στην παρούσα αρχικοποιημένη διαδικασία). Σε καθένα από τα παραπάνω βήματα πραγματοποιούνται διάφοροι έλεγχοι προς επιβεβαίωση της ορθότητας των εισαγομένων στοιχείων. Τέλος, ακριβώς πριν την τελική εισαγωγή των συνολικών παραπάνω στοιχείων στη βάση, παρουσιάζεται μια σελίδα επιβεβαίωσης αυτών (**βήμα 5**) και ο χρήστης καλείται είτε να τα επιβεβαιώσει οριστικά και αμετάκλητα, πραγματοποιώντας με επιτυχία την επιλογή κατεύθυνσης/ροών, είτε να τα απορρίψει ολοκληρωτικά, ξεκινώντας τη συναλλαγή του με την εφαρμογή ΣΗΕ από την αρχή, για λόγους και εδώ εγκυρότητας και authentication των προσωπικών του στοιχείων.

---

<sup>5</sup> Δικαίωμα επιλογής κατεύθυνσης/ροών διαθέτουν κάθε φορά μόνο οι φοιτητές που εγγράφονται στο 6<sup>ο</sup> εξάμηνο σπουδών τους, σύμφωνα με τον Οδηγό Σπουδών.

## 4.8 Υποσύστημα καθηγητών

Ο χώρος αποκλειστικής πρόσβασης των καθηγητών αποτελεί ένα εντελώς αυτόνομο υποσύστημα του ΣΗΕ. Με κατάλληλο authentication διατίθενται πληροφορίες για τους καθηγητές, όπως η παροχή στατιστικών στοιχείων για τα μαθήματα τα οποία διδάσκουν, αλλά και γενικότερου ενδιαφέροντος, όπως για ολόκληρα γνωστικά αντικείμενα.

Στην παρούσα φάση δεν έχει ολοκληρωθεί η ανάπτυξή του, καθώς θεωρήθηκε από τους επιβλέποντες ως αντικείμενο μελλοντικής υλοποίησης.

## 4.9 Υποσύστημα γραμματείας

Στο υποσύστημα της γραμματείας αποκλειστική πρόσβαση διαθέτουν οι κατάλληλα πιστοποιημένοι χρήστες της. Η λογική που χρησιμοποιείται στην πλοήγηση είναι παρόμοια με του υποσυστήματος των φοιτητών, με την ουσιώδη διαφορά της ύπαρξης πολύ περισσότερων δυνατοτήτων χειρισμού και ελέγχου της διαδικασίας εγγραφών και σημαντικά εμπλουτισμένης λειτουργικής μπάρας πλοήγησης σε κάθε σελίδα:



Εικόνα 4.9.α

Συγκεκριμένα υπάρχουν δυνατότητες για:

- απεικόνιση με τρεις διαφορετικούς τρόπους των records που έχουν εισάγει οι φοιτητές μέσω της διαδικασίας που περιγράφηκε παραπάνω
- πλοήγηση με δύο τρόπους σε όλες τις εγγραφές των φοιτητών
- εισαγωγή εγγραφής φοιτητή σε μαθήματα.
- αναζήτηση εγγραφών, τόσο με βάση συγκεκριμένα κριτήρια (όπως π.χ. ο «Α.Μ.» ή ο «Κωδικός Συστήματος»), όσο και με χρήση απευθείας SQL queries προς τη βάση. Επίσης στο ίδιο υπομενού παρέχεται η δυνατότητα για άμεση διόρθωση του ευρεθέντος record.
- απεικόνιση, πλοήγηση και χαρακτηρισμό ήδη υπάρχουσων εγγραφών φοιτητών ως «καθαρών» (και συνακόλουθα έτοιμων προς αποστολή στο φοιτητολόγιο) ή «βρώμικων» (και χρήζουσων διορθώσεων ή συμπληρώσεων).
- απεικόνιση και πλοήγηση των εγγραφών των φοιτητών σε επίπεδο κατευθύνσεων και ροών.

- εισαγωγή εγγραφής φοιτητή σε ροές και/ή κατευθύνσεις.
- αναζήτηση εγγραφών κατευθύνσεων/ροών, τόσο με βάση συγκεκριμένα κριτήρια (όπως π.χ. ο «Α.Μ.» ή ο «Κωδικός Συστήματος»), όσο και με χρήση απευθείας SQL queries προς τη βάση. Και σε αυτό το υπομενού παρέχεται η δυνατότητα για άμεση διόρθωση του ευρεθέντος record.

#### **4.10 Υποσύστημα administrator**

Όπως έγινε κατανοητό είναι επιτακτική η ανάγκη ενός administrator του όλου συστήματος. Μέσω αυτού του υποσυστήματος μπορεί ο τελευταίος να ασκεί έναν εποπτικό ρόλο στο όλο σύστημα, παρακολουθώντας μια σειρά παραμέτρων και πληροφοριών του, όπως:

- πληροφορίες σχετικές με το application module pool.
- πληροφορίες σχετικές με τα υπάρχοντα http sessions.
- παραμέτρους εκτέλεσης Java (Java runtime parameters).
- παραμέτρους εκτέλεσης των Business Components for Java (BC4J runtime parameters).
- πλήρη database report, βασισμένη σε πρόσφατα updated στοιχεία.
- πληροφορίες για τα View Objects της εφαρμογής.
- συνολικές πληροφορίες για την εφαρμογή.





# 5

## *Σχεδίαση του συστήματος HE*

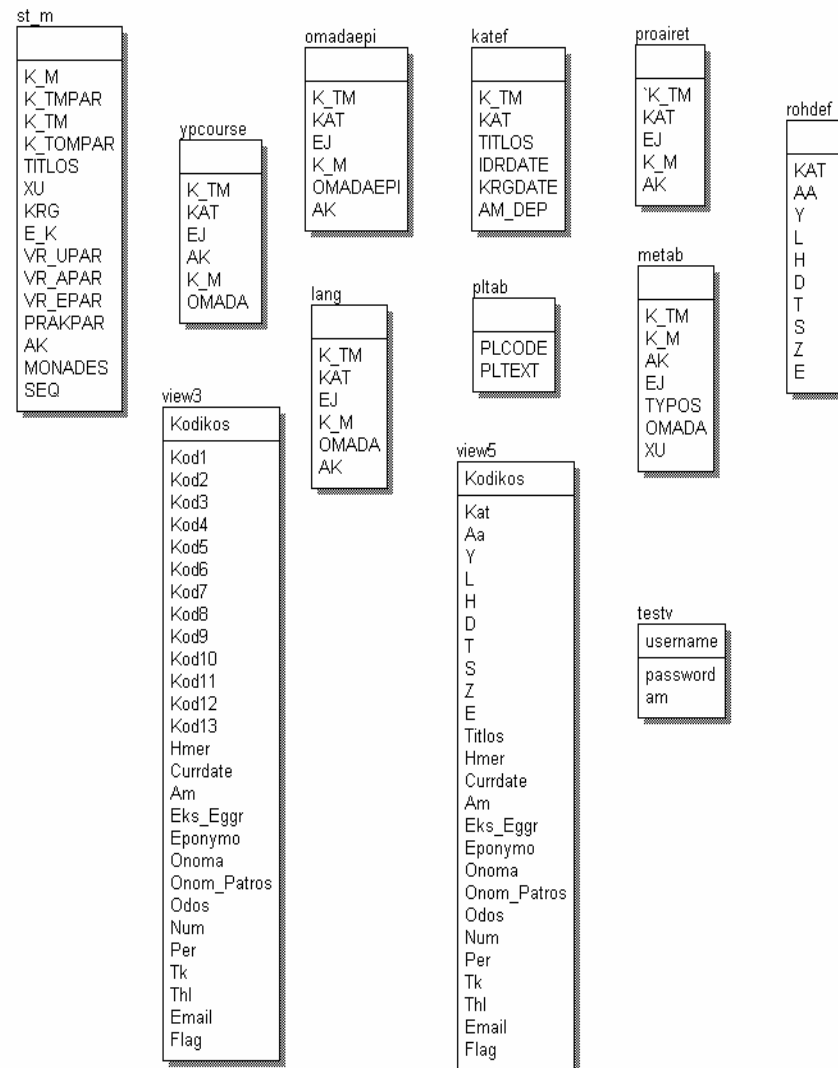
### *5.1 Σχεδίαση της Βάσης Δεδομένων*

Ένα από τα πιο σημαντικά βήματα κατά την υλοποίηση ενός project λογισμικού είναι η σχεδιάσή του, ιδιαίτερα αν αυτό περιλαμβάνει μία βάση δεδομένων, που αποτελεί τη ραχοκοκαλιά του και κυριολεκτικά τη βάση πάνω στην οποία χτίζεται ολόκληρο. Η ορθή και προσεκτική σχεδίαση του σχήματος της βάσης έχει ιδιαίτερη σημασία για το πρόγραμμα της εφαρμογής, καθώς μια καλά δομημένη βάση είναι σε θέση να δώσει σημαντική ώθηση σε αυτή.

Μία καλή σχεδίαση της βάσης δεδομένων συμβάλει αποτελεσματικά κατά τη συγγραφή του κώδικα προσπέλασής της, στην ταχύτητα εκτέλεσης των queries που υποβάλλονται, στην επεκτασιμότητα του σχήματος σε μελλοντικό χρόνο και όταν παρουσιαστούν καινούριες ανάγκες, όπως επίσης και στην αναγνωσιμότητα και συντήρησή του. Εν τέλει, μια καλή σχεδίαση της υποκείμενης βάσης δεδομένων αποτελεί κριτήριο ποιότητας του συνολικού συστήματος. Είναι βέβαια προφανή τα μειονεκτήματα που προκύπτουν από μία πρόχειρα οργανωμένη σχεδίαση και τα οποία μπορούν να οδηγήσουν ακόμη και σε πλήρη αχρήστευση του κώδικα που θα τη χρησιμοποιεί.

Η σχεδίαση του λογικού σχήματος της βάσης δεδομένων της εφαρμογής HE πραγματοποιήθηκε με χρήση του εργαλείου ERWin® 3.52 της Platinum©, και με γνώμονα την ικανοποίηση όλων των προδιαγραφών, οι οποίες τέθηκαν κατά το στάδιο της ανάλυσης του προβλήματος. Λήφθηκε δε υπόψη κατά μεγάλο ποσοστό η σχεδίαση της βάσης του φοιτητολογίου, πάνω και στην οποία στηρίχθηκε το όλο εγχείρημα, ιδιαίτερα για τη φάση της άντλησης read-only πληροφορίας προς

παρουσίαση. Έτσι, το μοντέλο Οντοτήτων-Συσχετίσεων του συστήματος έχει ως εξής:



### 5.1.1 Οντότητες

Βασική οντότητα (entity) του σχήματος είναι η **St\_m**. Κατά βάση όλη η λειτουργία του συστήματος περιστρέφεται γύρω από τα μαθήματα. Συνεπώς είναι απαραίτητη μία κεντρική οντότητα που θα έχει να κάνει με τα μαθήματα και τις όποιες πληροφορίες αυτά περικλείουν, όπως ο τίτλος του μαθήματος, ο κωδικός αριθμός του, η κατηγορία που ανήκει και άλλες. Όπως είναι αναμενόμενο, σχεδόν όλα τα SQL queries της εφαρμογής προσπελαίνουν και αντλούν στοιχεία από τον πίνακα αυτό. Συγκεκριμένα, ο πίνακας της οντότητας αυτής έχει ως εξής:

St_m
K_M
K_TMPAR
K_TM
K_TOMPAR
TITLOS
XU
KRG
E_K
VR_UPAR
VR_APAR
VR_EPAR
PRAKPAR
AK
MONADES
SEQ

Ακολουθούν οι πίνακες **Ypcourse**, **Omadaepi** και **Lang** των αντίστοιχων οντοτήτων, που περιέχουν αντίστοιχα πληροφορίες για τα υποχρεωτικά μαθήματα του προγράμματος σπουδών (μαζί με πληροφορία κατεύθυνσης για καθένα από αυτά), την αντιστοίχιση κωδικών ροής-κωδικών μαθημάτων και τις ξένες γλώσσες του προγράμματος σπουδών.

Ypcourse
K_TM
KAT
EJ
K_M
OMADA
AK

και

Omadaepi
K_TM
KAT
EJ
K_M
OMADAEPI
AK

Lang
K_TM
KAT
EJ
K_M
OMADA
AK

Δύο ακόμα βασικές οντότητες της βάσης δεδομένων είναι η **Katef** και η **Pltab**. Η πρώτη περιέχει την αντιστοίχιση των κωδικών κατεύθυνσης του συστήματος με τα ονόματα των κατευθύνσεων, ενώ η δεύτερη «ζευγαρώνει» κάθε κωδικό ροής με το αντίστοιχο γράμμα της ή την ένδειξη μισής ροής για το γράμμα αυτό, καθώς και

χαρακτηρίζει ένα μάθημα επιλογής του κορμού, αποτελώντας ουσιαστικά ένα λεκτικό pool για τις περιγραφές τους στη βάση. Δηλαδή:

Katef
K_TM
KAT
TITLOS
IDRDATE
KRGDATE
AM_DEP

Pltab
PLCODE
PLTEXT

Ακολουθούν κάποιες βοηθητικές/συμπληρωματικές οντότητες, όπως οι **Proairet** και η **Metab**, με την πρώτη να περιγράφει τα προαιρετικά μαθήματα του Οδηγού Σπουδών και τη δεύτερη να αναφέρεται στα μεταβατικά, σε περιπτώσεις αλλαγών του Προγράμματος Σπουδών, μαθήματα:

Proairet
K_TM
KAT
EJ
K_M
AK

Metab
K_TM
K_M
AK
EJ
TYPOS
OMADA
XU

Τέλος, ένας ιδιαίτερα σημαντικός πίνακας για την παρουσίαση των δυνατών συνδυασμών ανά κατεύθυνση είναι ο **Rohdef**, ο οποίος κρατάει όλους αυτούς δυνατούς συνδυασμούς ανά κατεύθυνση και ροή.

Rohdef
KAT
AA
Y
L
H
D
T
S
Z
E

Όσον αφορά τώρα την διαδικασία εισαγωγής στοιχείων από τους φοιτητές κατά τη φάση εκτέλεσης της εφαρμογής, διακρίνουμε δύο απαραίτητους πίνακες, με ονομασίες **View3** και **View5**, οι οποίοι και δέχονται όλα τα στοιχεία εισαγωγής απευθείας από τους φοιτητές μέσω web. Ο View3 κρατάει όλα τα στοιχεία εκείνα που

απαιτούνται κατά τη διαδικασία εγγραφής μαθημάτων, ενώ ο View5 διατηρεί τα στοιχεία εκείνα που χρειάζονται να περάσουν στο σύστημα κατά την επιλογή κατεύθυνσης και ροών από τους φοιτητές. Αναλυτικά, οι οντότητες αυτές έχουν ως εξής:

View3	View5
Kodikos (PK)	Kat
Kod1	Aa
Kod2	Y
Kod3	L
Kod4	H
Kod5	D
Kod6	T
Kod7	S
Kod8	Z
Kod9	E
Kod10	Titlos
Kod11	Hmer
Kod12	Kodikos (PK)
Kod13	Currdate
Hmer	Am
Currdate	Eks_Eggr
Am	Eponymo
Eks_Eggr	Onoma
Eponymo	Onom_Patros
Onoma	Odos
Onom_Patros	Num
Odos	Per
Num	Tk
Per	Thl
Tk	Email
Thl	Flag
Email	
Flag	

Υπάρχει, τέλος, ακόμα ένας πίνακας όπου βρίσκονται αποθηκευμένα τα στοιχεία authentication των χρηστών, με usernames, encrypted στη βάση passwords και Αριθμούς Μητρώων, τόσο για το γενική είσοδο στο σύστημα, όσο και για τη διαδικασία επιλογής ροής, ως εξής:

TESTV
Kodikos (PK)
\$password\$
Am

Σημειώνουμε, δε, ότι πολλά από τα παραπάνω πεδία των πινάκων έχουν συμπεριληφθεί στο σχήμα της βάσης, για λόγους πληρότητας και συμβατότητας με το σύστημα του φοιτητολογίου και της εκεί εβρισκόμενης βάσης δεδομένων, καθώς και για μελλοντικές χρήσεις και επεκτάσεις του Συστήματος Ηλεκτρονικών Εγγραφών.

### **5.1.2 Κρυπτογράφηση passwords:**

Ιδιαίτερη αναφορά θα πρέπει να γίνει στη χρήση κρυπτογραφημένων passwords στους δύο (2) πίνακες που χρησιμοποιούνται για το authentication των χρηστών του συστήματος. Η κρυπτογραφημένη αποθήκευση passwords στη βάση δεδομένων θεωρείται και είναι σήμερα απαραίτητη για λόγους ασφάλειας οποιουδήποτε συστήματος, ενώ ο συνδυασμός τους με τη χρήση SSL ή s-http για την προσπέλαση μέσω web αποτελεί μια άκρως αξιόπιστη λύση. Απλή αποθήκευση των passwords με τη μορφή plain text, είτε απευθείας στο σύστημα αρχείων, είτε πιο σύνθετα σε μία DB, τα καθιστά αυτόματα ευπαθή σε οποιονδήποτε έχει νόμιμη ή παράνομη πρόσβαση στο σύστημα.

Στο Σύστημα Ηλεκτρονικών Εγγραφών ακολουθήσαμε μια πολιτική υλοποίησης, η οποία θεωρεί την βάση και την εφαρμογή ως εντελώς ανεξάρτητες οντότητες, για προφανείς λόγους scalability και ευελιξίας. Συνακόλουθα, οι χρήστες της εφαρμογής δεν σχετίζονται με τους χρήστες της βάσης και οι πρώτοι θεωρούνται απλά «δεδομένα» αποθηκευμένα σε κάποιους πίνακες στη βάση, οπότε και όποια μέτρα για ασφάλεια χρηστών της βάσης (που εν προκειμένω χρειάζονται απαραίτητα για την ομαλή λειτουργία του όλου συστήματος) δεν βρίσκουν εφαρμογή στους χρήστες της εφαρμογής.

Ακολουθήθηκε, λοιπόν, η λύση ενός «εξωτερικού» PL/SQL πακέτου, το οποίο έρχεται μαζί με τη βάση δεδομένων της Oracle και ονομάζεται DBMS\_OBFUSCATION\_TOOLKIT. Το πακέτο αυτό υποστηρίζει μαζική κρυπτογράφηση και αποκρυπτογράφηση αποθηκευμένων δεδομένων χρησιμοποιώντας τους αλγόριθμους **Data Encryption Standard (DES)** και **Triple DES**<sup>6</sup>, ενώ παρέχει και δυνατότητα για την παραγωγή ενός MD5 checksum. Ο αναγνώστης μπορεί να ανατρέξει στον επισυναπτόμενο κώδικα για τη δήλωση σε PL/SQL του πακέτου αυτού.

Τέλος, να σημειώσουμε ότι με την παραπάνω διαδικασία η κρυπτογράφηση των δεδομένων πραγματοποιείται πριν από την αποθήκευση στη βάση. Κατά το authentication ενός χρήστη πραγματοποιείται κατάλληλο encryption και σύγκριση με τα ήδη κρυπτογραφημένα στοιχεία του αντίστοιχου πίνακα της βάσης δεδομένων.

---

<sup>6</sup> Εξ ορισμού, το πακέτο δεν επιτρέπει την χρήση διαφορετικού αλγόριθμου κρυπτογράφησης από τους παραπάνω ή/και την πολλαπλή κρυπτογράφηση με πολλαπλό πέρασμα των data από την ίδια function, λόγω των ισχυρών περιορισμών που θέτουν οι αμερικανικοί νόμοι περί εξαγωγής κρυπτογραφημένου υλικού!

## **5.2 Σχεδίαση του προγράμματος ΗΕ**






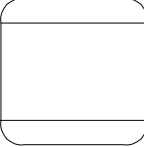
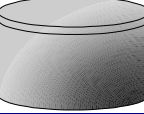

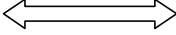

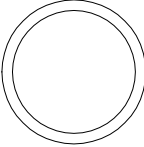
Στη συνέχεια ακολουθεί η αναλυτική περιγραφή της σχεδίασης του ίδιου του Σ.Η.Ε.. Θα πραγματοποιηθεί ιδιαίτερη εστίαση στην presentation logic του project, με την ιδιαίτερη διαδοχή των σελίδων του και τη γενικότερη πλοήγηση του χρήστη σε αυτό. Στα σχεδιαγράμματα που ακολουθούν χρησιμοποιήθηκε το πρόγραμμα της εταιρίας Microsoft® “Visio 2001 Technical Edition”. Αρχικά πραγματοποιείται μια μικρής έκτασης εισαγωγή των χρησιμοποιούμενων διαγραμμάτων, συμβόλων και σχημάτων και ακολουθούν μερικά ενδεικτικά διαγράμματα, που στόχο έχουν να δείξουν τη φιλοσοφία πίσω από τη σχεδίαση και υλοποίηση του συστήματος.

### **5.2.1 Εισαγωγή – επεξήγηση διαγραμμάτων**

Λαμβάνοντας υπόψη ότι για τέτοιου είδους web-based εφαρμογές δεν υπάρχει μέχρι σήμερα μια κοινά αποδεκτή και προτυποποιημένη μεθοδολογία σχεδίασης, σε αντίθεση με τον καθαρό διαδικασιακό ή αντικειμενοστραφή προγραμματισμό και την πληθώρα επιλογών τους, καταλήξαμε στη λύση της χρήσης custom-made διαγραμμάτων στο γενικότερο πλαίσιο των διαγραμμάτων ροής. Στα διαγράμματα αυτά θα προσπαθήσουμε να κάνουμε κατανοητές και να παρουσιάσουμε με σχετική λεπτομέρεια, τις δυνατότητες πλοήγησης των φοιτητών στα διάφορα τμήματά του συστήματος, είτε αυτά πρόκειται για στατικά html κομμάτια, είτε για δυναμικά παραγόμενες jsp σελίδες, με ύπαρξη αλληλεπίδρασης ή όχι με τη βάση δεδομένων. Επίσης, θα γίνει παρουσίαση ορισμένων χρήσιμων λειτουργιών του συστήματος, όπως διάφορες toolbars, κ.λ.π..

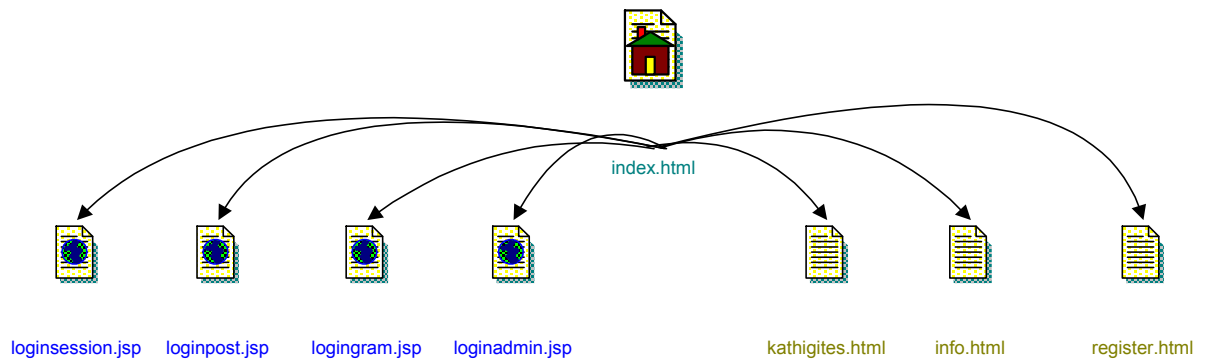
Θα παρουσιαστούν, επίσης, ενδεικτικά στα διαγράμματα αυτά δευτερεύοντα και «κρυμμένα» από τον χρήστη χαρακτηριστικά της εφαρμογής, όπως η διαδικασία του να τίθεται ένα cookie ή το url rewriting σε περίπτωση μη χρήσης cookies. Επίσης οτιδήποτε έχει να κάνει με session variables και διατήρησή τους ή expiration τους φαίνεται με κατάλληλα σύμβολα στο διάγραμμα, ενώ αναλύεται και η θέση και εμφάνιση των μηνυμάτων λάθους, τόσο απ’ ευθείας από τη βάση δεδομένων, όσο και από τα διάφορα components της εφαρμογής. Παρόλα αυτά, τυχόν μηνύματα λάθους, που εμφανίζονται στη θέση της κανονικής απόκρισης του συστήματος, ή του κώδικα της Javascript, που εκτελείται σε μερικές από τις σελίδες του συστήματος δεν είναι δυνατόν να παρασταθούν αναλυτικά

Για τα σχέδια, λοιπόν, του συστήματος, θα χρησιμοποιήσουμε το παρακάτω σύνολο σχημάτων, τα οποία συνοδεύονται από μια στοιχειώδη επεξήγηση της λειτουργίας τους:

Σύμβολο	Επεξήγηση
	αρχική σελίδα εφαρμογής
	στατική html σελίδα.
	δυναμική JSP σελίδα.
	μονόδρομη επικοινωνία με το χρήστη
	interaction με το χρήστη, εισαγωγή στοιχείων
	authentication
	βάση δεδομένων
	cookie ή session parameter setting
	αμφίδρομο query στη βάση δεδομένων
	μονόδρομο query στη βάση δεδομένων
	περιορισμένη πρόσβαση

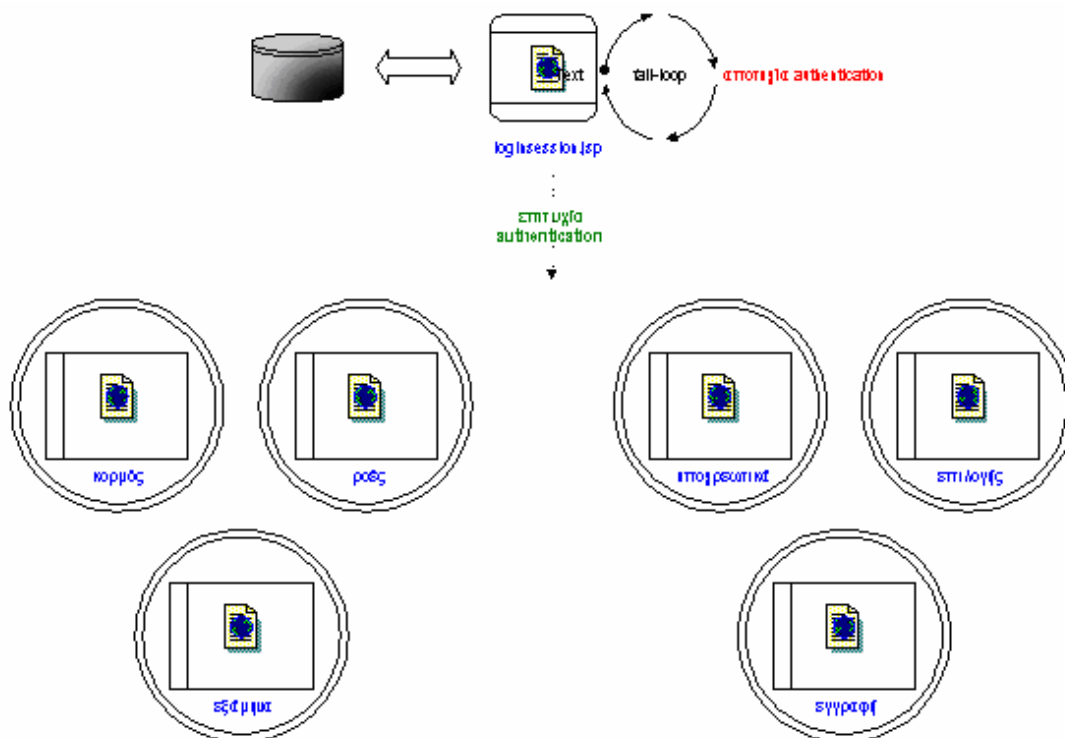
Στο πρώτο διάγραμμα που ακολουθεί, διαφαίνεται η αρχική ακτινωτή δομή του Συστήματος Ηλεκτρονικών Εγγραφών. Σκοπός του είναι να καταδείξει τις δυνατότητες πλοήγησης υποσυστημάτων από την αρχική σελίδα [index.html](#) του συστήματος, ενώ πραγματοποιείται και μία οπτική κατηγοριοποίηση σε [δυναμικές](#) και [στατικές](#) «πρώτες» σελίδες σε καθένα από αυτά.





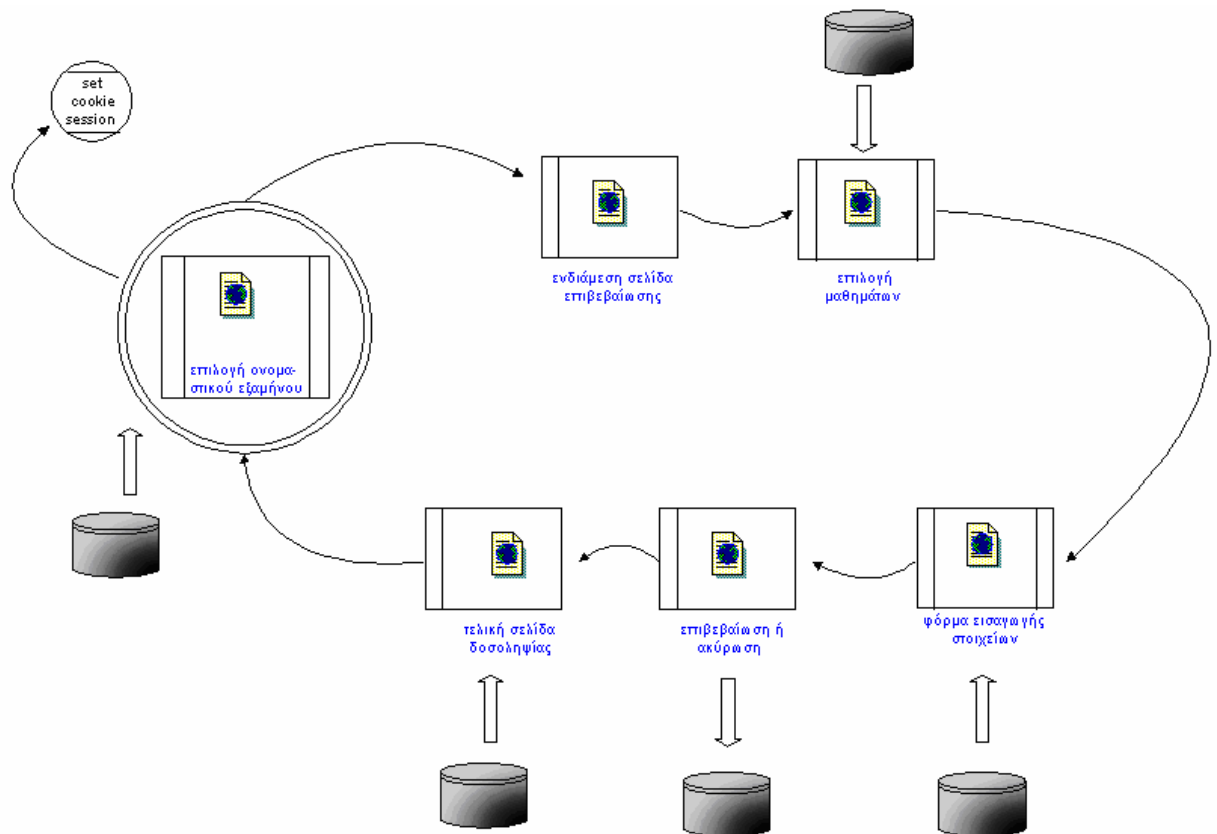
Διάγραμμα 5.2.1.α

Το επόμενο διάγραμμα παρουσιάζει την φιλοσοφία της πολιτικής πρόσβασης στο σύνολο του υποσυστήματος που προορίζεται για χρήση από τους φοιτητές. Υπάρχει μία αρχική σελίδα authentication που επικοινωνεί αμφίδρομα με τη βάση δεδομένων προκειμένου να πιστοποιήσει τον εκάστοτε χρήστη. Σε περίπτωση **αποτυχίας** εισέρχεται κανείς σε loop, ενώ σε περίπτωση **επιτυχούς authentication**, ο χρήστης έχει επτά ομάδες περιορισμένης πρόσβασης να επιλέξει. Ακριβώς, δε, επειδή είναι τέτοιου είδους ομάδες, παράκαμψη της αρχικής σελίδας του υποσυστήματος για απευθείας πρόσβαση σε κάποια από τις εσωτερικές δεν είναι δυνατή.



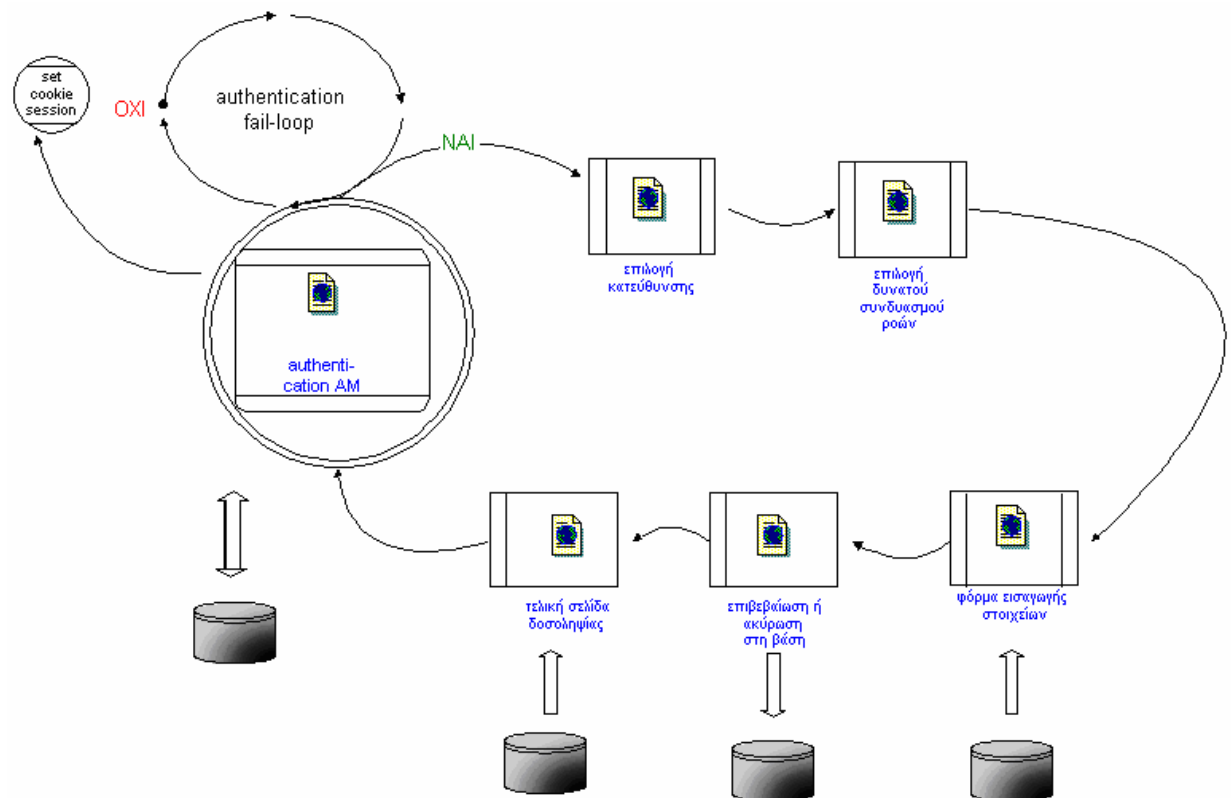
Διάγραμμα 5.2.1.β

Το τρίτο ενδεικτικό του συστήματος διάγραμμα ασχολείται με την παρουσίαση της λογικής πίσω από τη διαδικασία εγγραφής σε μαθήματα ονομαστικού εξαμήνου. Παρατηρούμε ότι σχεδόν σε κάθε σελίδα, την οποία προσπελαίνει ο χρήστης διαδοχικά και μέσω συγκεκριμένου προκαθορισμένου μονοπατιού, υπάρχει αφενός μεν database interaction και αφετέρου δε αλληλεπίδραση με αυτόν. Να σημειώσουμε επίσης ότι όλες ανεξαιρέτως οι σελίδες που διακρίνονται στο παρακάτω διάγραμμα βρίσκονται υπό την προστασία του ασφαλούς container περιορισμένης πρόσβασης που ορίζεται με την αρχική σελίδα επιλογής ονομαστικού εξαμήνου.



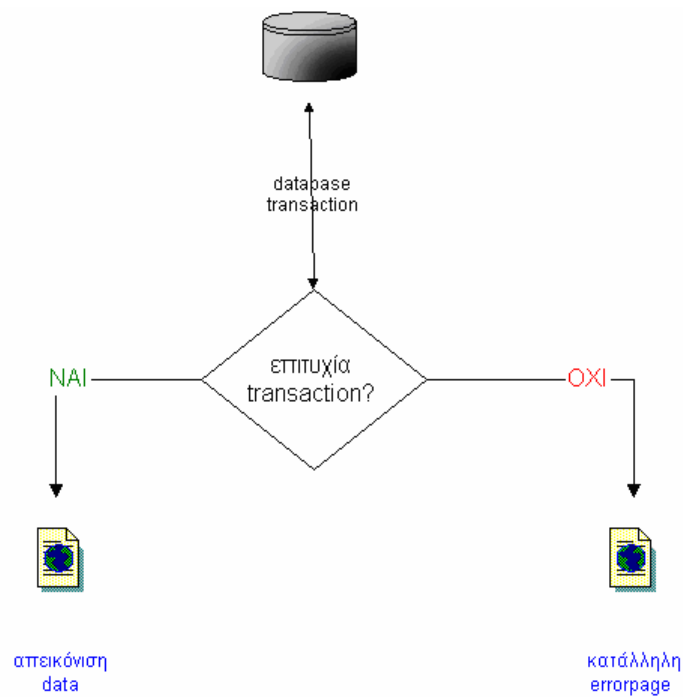
Διάγραμμα 5.2.1.γ

Η διαδικασία εγγραφής σε κατεύθυνση και επιτρεπτούς συνδυασμούς ροών διαφαίνεται στο διάγραμμα 5.2.1.δ που ακολουθεί. Η κύρια διαφορά σε σχέση με προηγουμένως έγκειται στο γεγονός της ύπαρξης δεύτερου, σε σχέση με το καθολικό πρώτο, authentication loop, ουσιαστικά για την εξακρίβωση του εξαμήνου του υποψήφιου χρήστη του κομματιού αυτού του συστήματος, καθότι ενδιαφέρει η συνολική μείωση του φόρτου του προς database transactions και http overhead. Κατά τα υπόλοιπα, η λογική είναι η ίδια με αυτή που εξετάσαμε προηγουμένως.



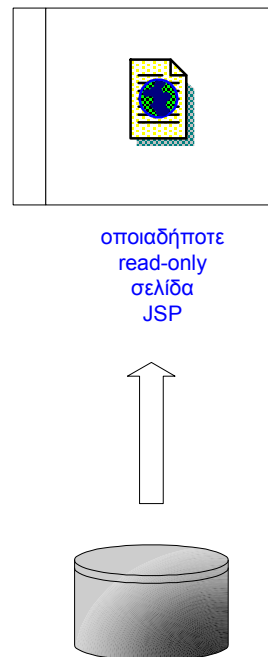
Διάγραμμα 5.2.1.δ

Ακολουθεί το διάγραμμα που διέπει τη λογική πίσω από κάθε transaction του συστήματος, όπου βλέπουμε ότι ανάλογα την επιτυχία ή την αποτυχία του, παρουσιάζεται και στο χρήστη κατάλληλα διαμορφωμένη δυναμική σελίδα:



Διάγραμμα 5.2.1.δ

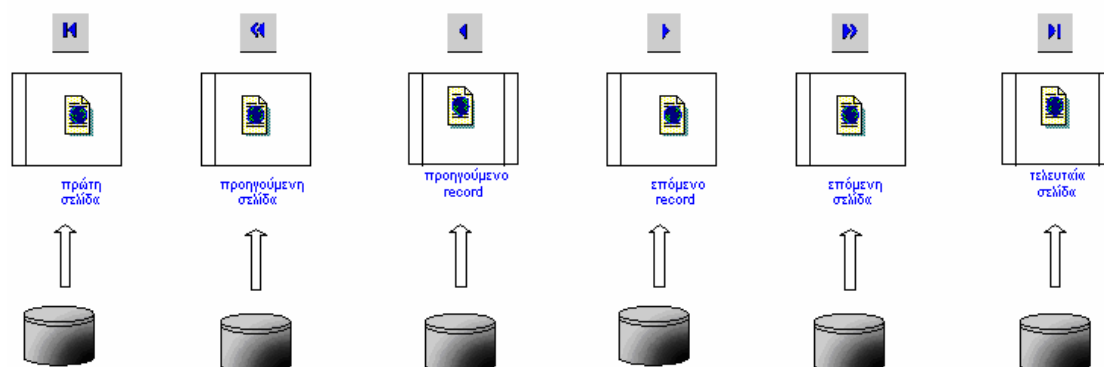
Κάθε read-only JSP σελίδα ολόκληρου του συστήματος, στην οποία εμπεριέχονται πληροφορίες από τη βάση, όπως για παράδειγμα όλες οι σελίδες παρουσίασης εξαμήνων ή/και επιλογής μαθημάτων (checkbox list), έχει την εξής μορφή:



Διάγραμμα 5.2.1.ε

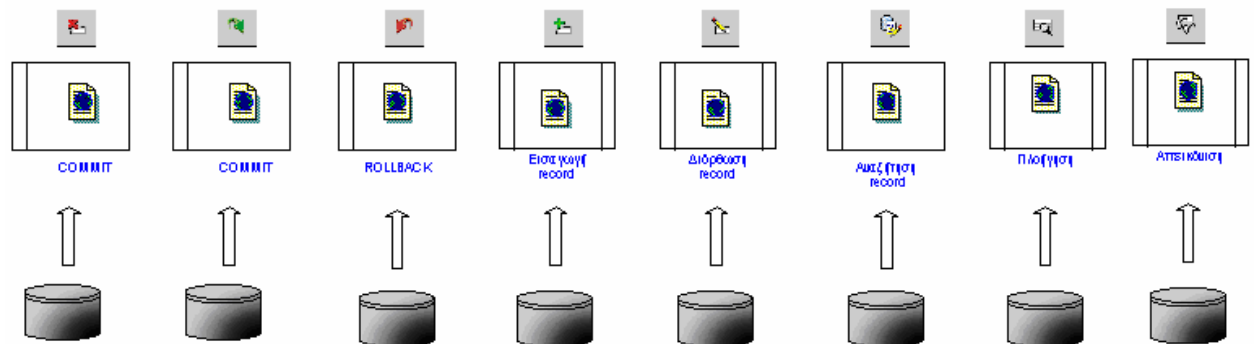
Στα επόμενα δύο διαγράμματα καταδεικνύεται η λειτουργικότητα της υλοποίησης μπάρας πλοήγησης (navigation toolbar) και μπάρας εργαλείων ελέγχου (control toolbar). Οι μπάρες αυτές είναι θεμιτό και εύκολο να επαναχρησιμοποιηθούν κατά την ανάπτυξη της εφαρμογής, καθώς της προσδίδουν ευελιξία και λειτουργικότητα, ενώ απλοποιούν σημαντικά την πλοήγηση στα σημεία της στα οποία παρέχουν πρόσβαση. Κάθε τμήμα της εκάστοτε μπάρας επικοινωνεί αμφίδρομα ή μονόδρομα με τη βάση δεδομένων, ανάλογα με την ίδια την υπηρεσία που αυτό παρέχει στο τελικό χρήστη.

### Navigation Toolbar



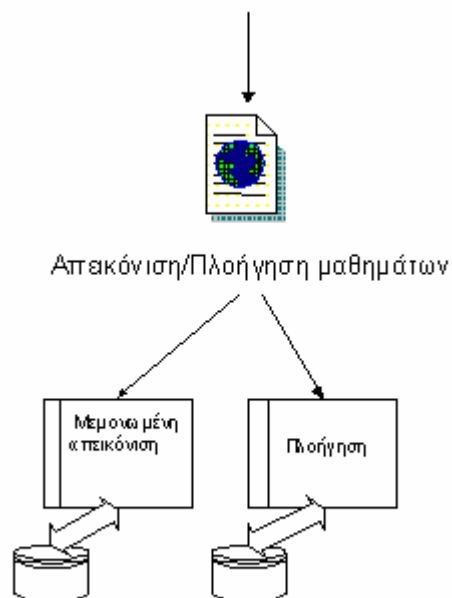
Διάγραμμα 5.2.1.στ

### Control Toolbar



Διάγραμμα 5.2.1.η

Τέλος, παρουσιάζεται μέσω διαγράμματος και η βασική μορφή όλων των απεικονίσεων και πλοηγήσεων του συστήματος. Από μία εισαγωγική δυναμική JSP σελίδα μπορούμε να έχουμε τόσο απλή απεικόνιση μεμονωμένης πληροφορίας, όσο και πιο σύνθετη πλοήγηση σε αυτή και στις γειτονικές της. Παρατηρούμε ότι πρόκειται για σελίδες στις οποίες η επικοινωνία με το χρήστη είναι μονόδρομη, σε αντίθεση με τα queries στη βάση δεδομένων, τα οποία μπορούν σε ορισμένες περιπτώσεις να είναι και αμφίδρομα.



Διάγραμμα 5.2.1.θ

### 5.2.2 Σχεδίαση πολιτικής πρόσβασης:

Όλη η φιλοσοφία γύρω από τη σχεδίαση του συστήματος αναφορικά με την κατηγοριοποίηση των χρηστών και τις δυνατότητες πιστοποίησής τους, έχει δύο διακριτά επίπεδα. Ένα είναι το επίπεδο authentication του χρήστη, το οποίο έχει αναλάβει όπως είδαμε ένας πίνακας στη βάση με κρυπτογραφημένα passwords, που καθορίζει έμμεσα και τι είναι σε θέση να προσπελάσει ο εκάστοτε χρήστης ανάλογα με την ομάδα στην οποία ανήκει, και ένα άλλο είναι το επίπεδο όπου καθορίζεται τι είναι θεμιτό να γνωρίζει ο χρήστης για το τμήμα του συστήματος στο οποίο του έχει παραχωρηθεί πρόσβαση.

Με άλλα λόγια στο σύστημά μας ο χρήστης δεν θα πρέπει για προφανείς λόγους ασφαλείας να έχει πρόσβαση στο document root του Apache, ή να είναι σε θέση να δει έστω τη δομή του web site, πέρα από τη λογική κατευθυντήρια γραμμή που του ορίζεται με τη συμβατική πιστοποιημένη πλοήγησή του. Στα παραπάνω πλαίσια θα υλοποιηθούν διάφοροι μηχανισμοί περιορισμού του, από τον πιο απλό, που αποτελεί η απενεργοποίηση σε συγκεκριμένα σημεία του back button του browser, έτσι ώστε να μην μπορεί να γυρίσει σε μια σελίδα που π.χ. έχει ήδη δει/συμπληρώσει κατά τη διαδικασία εγγραφής, μέχρι το πιο σύνθετο, που είναι η διάφανη παρουσίαση των σελίδων της εφαρμογής μέσω ενός μοναδικού url στην address bar του browser και τη χρήση frames.

Επίσης, μια σημαντικότερη παράμετρος στο σύστημά μας αποτελεί η παρεμπόδιση της αποθήκευσης με τη μορφή bookmarks σελίδων πέρα από την αρχική πριν το login του κάθε χρήστη. Κι αυτό διότι κάτι τέτοιο θα καταργούσε ως ένα βαθμό την λειτουργικότητα του. Έτσι προς το σκοπό αυτό θα χρησιμοποιηθούν συγκεκριμένες directives [L25], όπως οι ακόλουθες:

```
<%  
    response.setHeader("Cache-Control","no-store");           //HTTP 1.1  
    response.setHeader("Pragma","no-cache");                 //HTTP 1.0  
    response.setDateHeader ("Expires", 0);                   //prevents caching  
στον proxy server  
%>
```

και:

```
<meta http-equiv="Pragma" content="no-cache">
```

οι οποίες και καθοδηγούν τον browser του χρήστη να μην αποθηκεύει στην cache οποιαδήποτε πληροφορία της σελίδας, ενώ οι επόμενες γραμμές κώδικα αναγκάζουν το χρήστη σε redirect στην login page εφόσον δεν έχει κάνει ήδη login (περίπτωση bookmark):

```
<%  
Object obj_getdata = session.getAttribute("Testing");  
String getdata = (String)obj_getdata;  
  
if ( getdata == null ) {  
    response.sendRedirect  
(response.encodeRedirectUrl("loginsession.jsp"));  
}  
  
else if ( getdata == session.getId() ) {  
}  
%>
```





# 6

## *Υλοποίηση του συστήματος ΗΕ*

### *6.1 Επιλογή Πλατφόρμας και Εργαλείων*

Η σωστή επιλογή πλατφόρμας και εργαλείων συγγραφής και ανάπτυξης λογισμικού είναι ιδιαίτερα σημαντική για την ανάπτυξη ενός τέτοιου συστήματος ως προς την αποδοτικότητα, την επεκτασιμότητα και την μεταφερσιμότητα της εφαρμογής. Συνάμα, απλοποιείται κατά πολύ η διαδικασία του προγραμματισμού και ο προγραμματιστής επικεντρώνεται και εστιάζει στη σχεδίαση και τη βελτιστοποίηση του παραγόμενου κώδικα.

Θεωρώντας ότι έχουμε καλύψει τα πεδία των ουσιαστών επιλογών της βάσης δεδομένων στο κεφάλαιο 2.11 (**Oracle8i**), της κύριας γλώσσας προγραμματισμού στα κεφάλαια 2.13-2.14 (**Java-JSPs**) και του εργαλείου ανάπτυξης λογισμικού στο κεφάλαιο 2.17 (**Oracle Jdeveloper**), θα επικεντρωθούμε στο κεφάλαιο αυτό στα υπόλοιπα εργαλεία που χρησιμοποιήθηκαν κατά την ανάπτυξη της εφαρμογής του Συστήματος Ηλεκτρονικών Εγγραφών.

Ακολουθώντας, λοιπόν, τα υψηλά standard όλων των σύγχρονων web-based εφαρμογών βάσεων δεδομένων επιλέχθηκε ο **Apache Web Server** (έκδοση 1.3.14) καθώς είναι από τους πλέον διαδεδομένους, αξιόπιστους, γρήγορους αλλά και συνεχώς αναπτυσσόμενους που υπάρχουν σήμερα. Επίσης επιτρέπει την αναδιανομή και χρήση τόσο του source όσο και των binaries του κάτω από ιδιαίτερα ελεύθερες συνθήκες και δωρεάν, ενώ αποτελεί εδώ και χρόνια και την επιλογή του Εργαστηρίου Βάσεων Γνώσεων και Δεδομένων στο οποίο και εκπονήθηκε η διπλωματική.

Προκειμένου να κατασταθεί ο Apache “Java enabled web server” χρησιμοποιήθηκε το **Tomcat** (έκδοση 3.2), το οποίο αποτελεί την επίσημη Reference Implementation

του για τα Java Servlets και τα JavaServer Pages. Το Tomcat αναπτύσσεται σε ένα ανοιχτό περιβάλλον πολλαπλών συμμετοχών και διέπεται από την [Apache Software License](#).

Από τα συστήματα dynamic load balancing που εξετάσαμε στο κεφάλαιο 2.1.5.3, επιλέχθηκε τελικά για το Σύστημα Ηλεκτρονικών Εγγραφών η λύση του DNS-based συστήματος με adaptive TTL αλγόριθμο και **Round-Robin DNS**, κυρίως για λόγους απόδοσης, μιας και απαιτείται πλήρης διαφάνεια σε επίπεδο URL για τους χρήστες

Όσον αφορά το πεδίο του web caching, για τις ανάγκες του συστήματος μας, θα βασιστούμε προς το παρόν στο ήδη υπάρχον σύστημα web cache του δικτύου δεδομένων του Εθνικού Μετσόβιου Πολυτεχνείου, το οποίο όπως είδαμε χρησιμοποιεί την βέλτιστη αυτή τη στιγμή στον τομέα αυτό λύση, το **Squid**.

Από τους 3 τρόπους διασύνδεσης στην Oracle8i database που προαναφέρθηκαν στο κεφάλαιο 2.13, επιλέχθηκε η λύση με τον **JDBC Thin Client-Side Driver**. Έτσι δεν χρειαζόμαστε browsers που να δέχονται συγκεκριμένα plug-ins, κάτι που επιβαρύνει σημαντικά τον client, αλλά ούτε και ιδιαίτερα πολύπλοκη διαχείριση του web server, συντελώντας προφανώς στην μεταφερσιμότητα και επεκτασιμότητα της εφαρμογής μας.

## 6.2 Υλοποίηση βάσης δεδομένων:

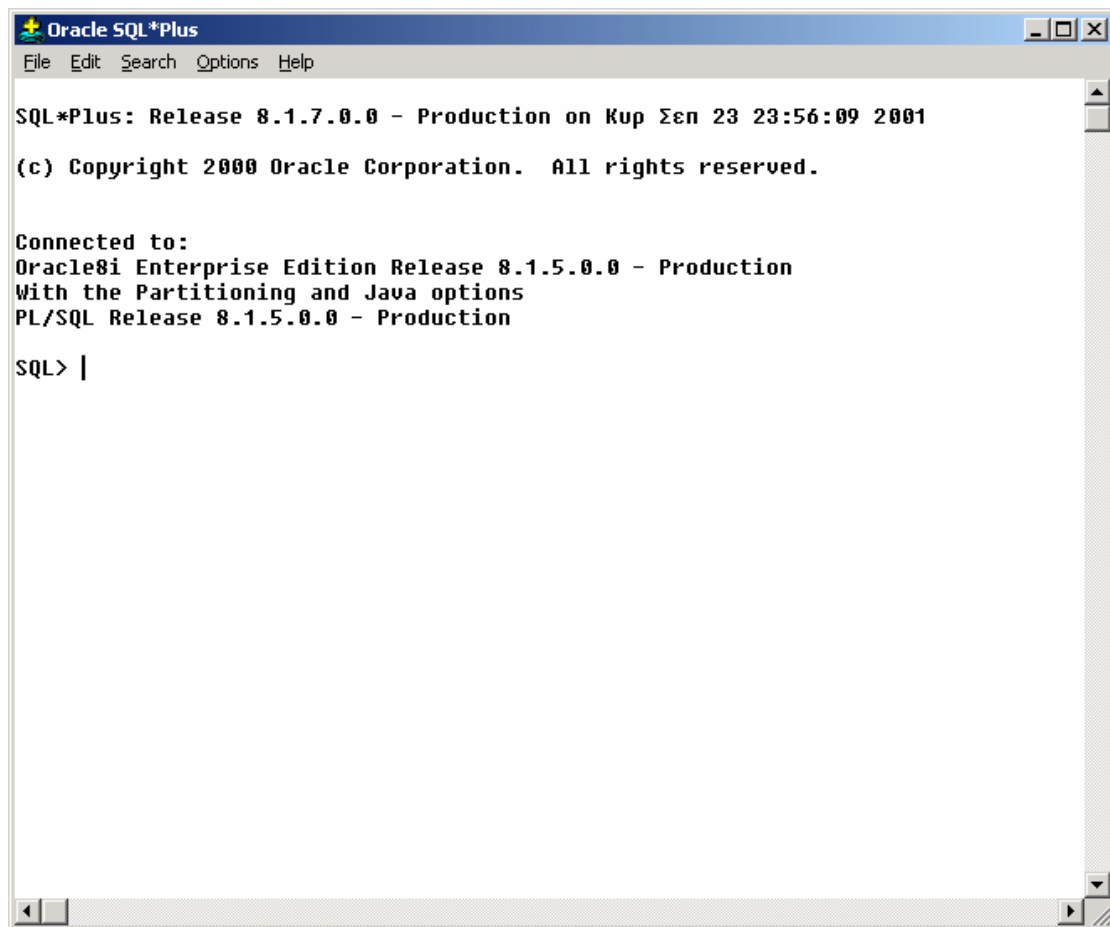
Η αρχιτεκτονική της βάσης δεδομένων που χρησιμοποιήθηκε (Oracle8i) αναλύθηκε διεξοδικά στο κεφάλαιο 2.11. Εδώ θα αναφερθούμε στα επιμέρους εργαλεία και χαρακτηριστικά του DBMS που συνέβαλαν στην ολοκλήρωση της εφαρμογής του Συστήματος Ηλεκτρονικών Εγγραφών.

Το βασικό command line εργαλείο που χρησιμοποιήθηκε κατά κόρον για raw SQL queries στη βάση του συστήματος ήταν το SQL \*Plus, έκδοση 8.1.7.0.0:



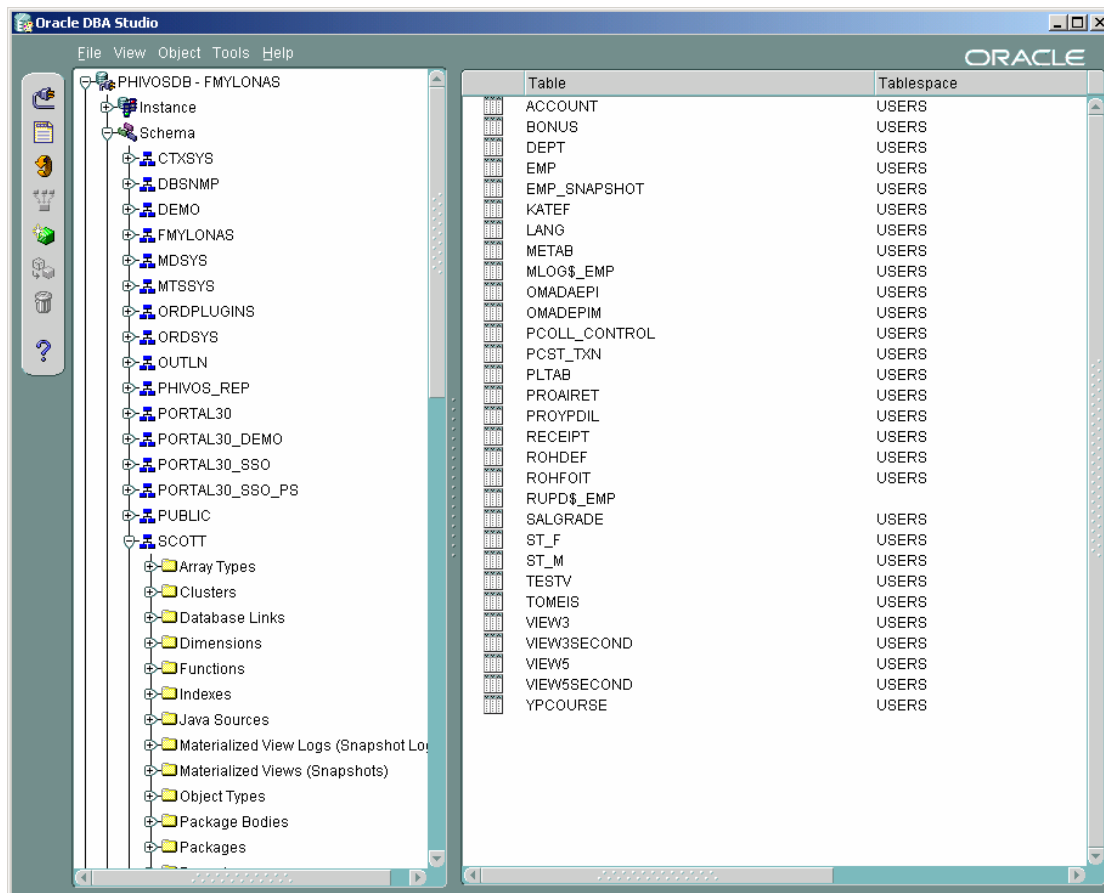
Εικόνα 6.2.α

Μέσω ενός ιδιαίτερα λιτού και απλού interface ήταν δυνατή η εκτέλεση σύνθετων SQL queries και η ταυτόχρονη συγκομιδή των αποτελεσμάτων τους είτε στην οθόνη, είτε σε flat text files.



*Εικόνα 6.2.β*

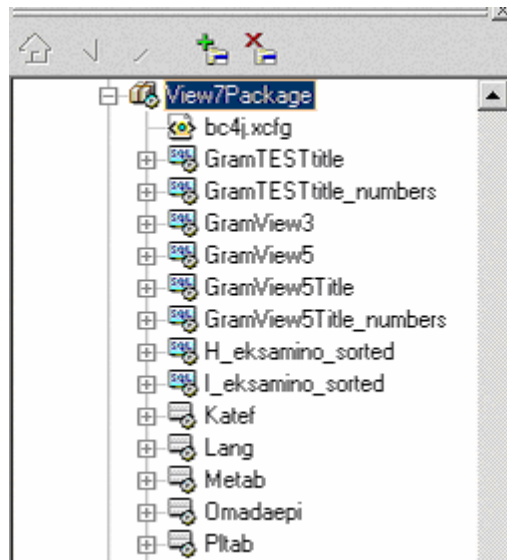
Θεωρητικά είναι δυνατή και η έως ένα βαθμό εποπτεία της βάσης και η διαχείρισή της μέσα από τον client αυτόν, αν θέσει κανείς τα παραπάνω σε επίπεδο «καθαρής» SQL. Για το σκοπό αυτό όμως προτιμήθηκε η λύση του visual εργαλείου, το οποίο ακούει στο όνομα «DBA Studio» και ακολουθώντας την κατηγοριοποίηση και τη δενδρική δομή του Explorer των MS Windows® παρείχε άμεσα εποπτεία στη βάση του συστήματός μας.



Εικόνα 6.2.γ)

### 6.3 Υλοποίηση BC4J:

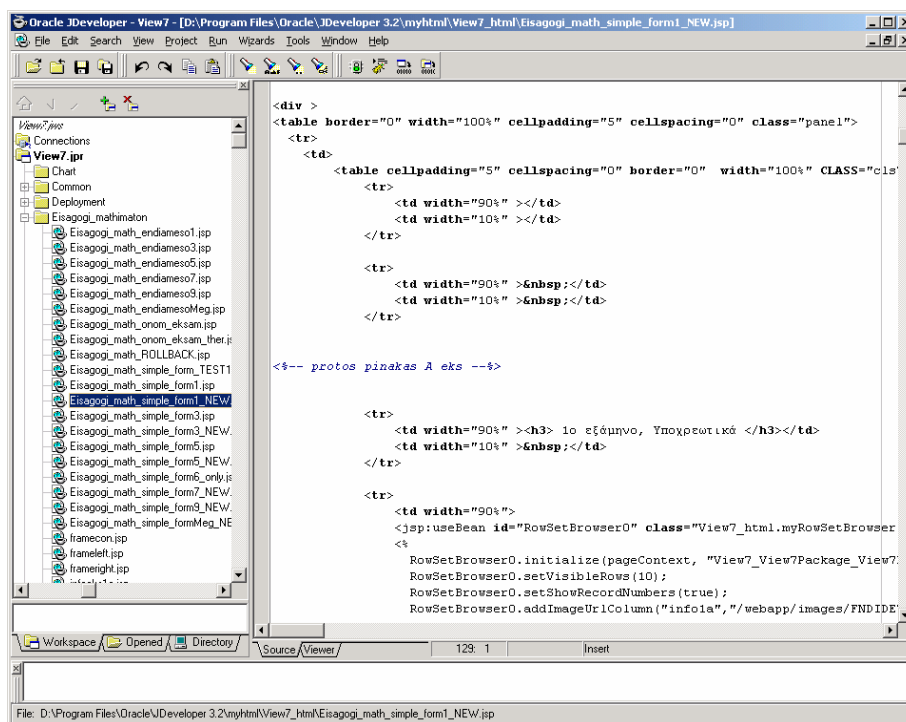
Κατασκευάστηκε μια πληθώρα από **Entity Objects** (~40) και **View Objects** (~70) μέσω του εργαλείου JDeveloper προκειμένου να ικανοποιηθούν οι εξειδικευμένες ανάγκες των επιμέρους τμημάτων του Συστήματος Ηλεκτρονικών Εγγραφών. Σε κάθε περίπτωση διαφορετικού query στη βάση για οποιοδήποτε λόγο, απαιτείται προφανώς, και με βάση τη λογική που αναπτύχθηκε διεξοδικά στο κεφάλαιο 2.16, και διαφορετικό VO πάνω στο ίδιο υποκείμενο EO. Για παράδειγμα ένα μέρος μόνο των EO και VO που χρησιμοποιήσαμε για το υποσύστημα των προπτυχιακών φοιτητών διακρίνεται στην παρακάτω εικόνα, με τα EO να χαρακτηρίζονται από άσπρο χρώμα, ενώ τα VO έχουν μπλε:



Εικόνα 6.3.α

## 6.4 Υλοποίηση JSP files:

Συνολικά πραγματοποιήθηκε συγγραφή περίπου 93000 γραμμών JSP σελίδων. Σε αυτές συμπεριλαμβάνεται, λόγω της ίδιας της μορφής των JSPs, τόσο η «καθαρή» γλώσσα html, όσο και ο embedded κώδικας γλώσσας Java σε αυτές. Ως επί το πλείστον η υλοποίησή τους έγινε μαζικά μέσα στο περιβάλλον του εργαλείου JDeveloper, στο προγραμματιστικό περιβάλλον που διακρίνεται στην παρακάτω εικόνα. Περαιτέρω εκλέπτυνση και τυχούσα αισθητική αναβάθμισή τους πραγματοποιήθηκε με χρήση διαφόρων διαθέσιμων εργαλείων editor, όπως το Notepad, το UltraEdit και το Dreamweaver.

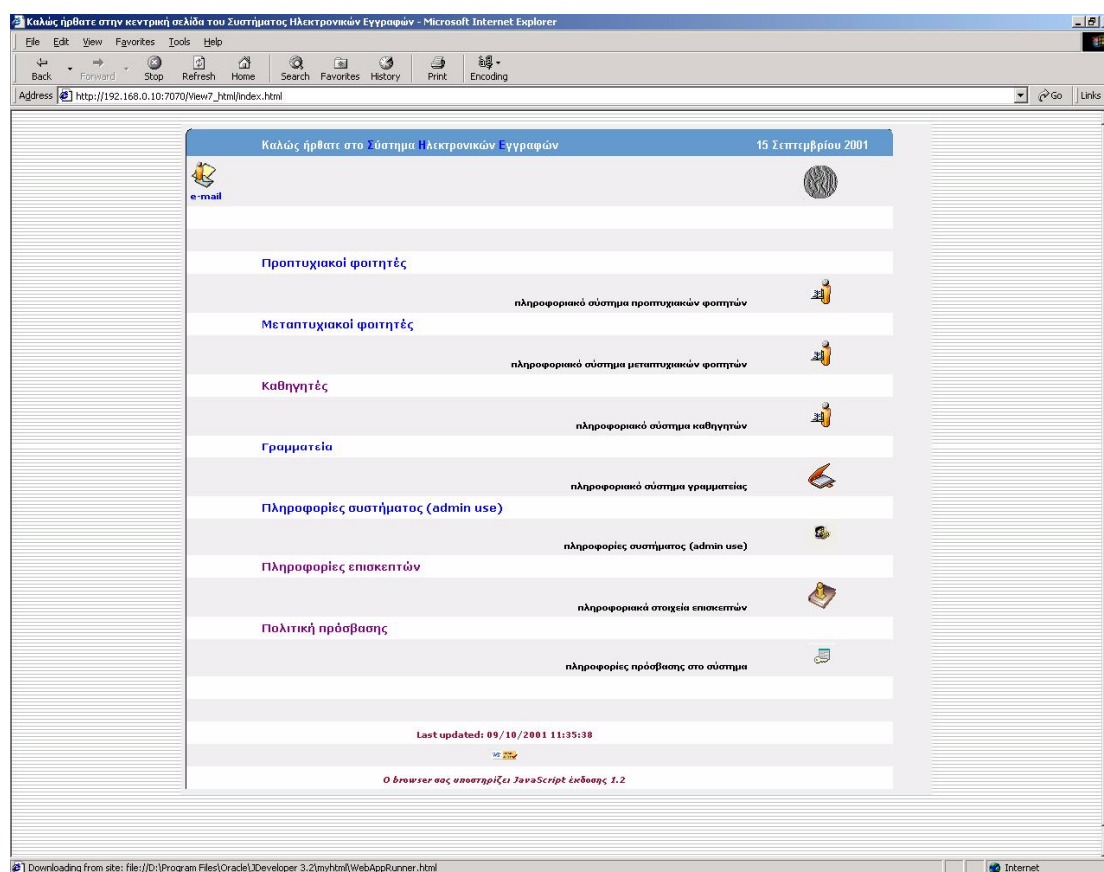


Εικόνα 6.4.α

Να σημειώσουμε εδώ ότι ακολουθήθηκε πιστά η λίστα των διαθέσιμων tags του κεφαλαίου 2.14, προκειμένου να υπάρχει πλήρης συμφωνία με το JSP 1.2 standard [L26].

## 6.5 Υλοποίηση static html:

Τα στατικά χαρακτηριστικά μιας html σελίδας, όπως αυτά εξηγούνται στο κεφάλαιο 2.6, μπορούν πολύ εύκολα να κωδικοποιηθούν μέσα από οποιονδήποτε από τους γνωστούς εξειδικευμένους html editors, αλλά και μέσα από οποιονδήποτε text editor, όπως το **Notepad**, το **UltraEdit 8.0**, κ.ο.κ.. Αυτό που γίνεται είναι, είτε η συγγραφή html tags (<body>, <form>, <table>, <br> κ.λ.π.) «με το χέρι», είτε η χρήση κάποιου γραφικού περιβάλλοντος (όπως π.χ. το **Macromedia Dreamweaver 4.0**), το οποίο και χρησιμοποιείται κυρίως για την αισθητική αναβάθμιση μιας σελίδας. Παράδειγμα υλοποίησης μιας τέτοιας σελίδας στατικού περιεχομένου αποτελεί η αρχική σελίδα του Συστήματος Ηλεκτρονικών Εγγραφών index.html (**Εικόνα 6.5.α**), ο κώδικας της οποίας μπορεί να αναζητηθεί στο συνοδευτικό παράρτημα:



Εικόνα 6.5.α

## 6.6 Υλοποίηση *dynamic html* από Javascript:

Σε ορισμένες περιπτώσεις και ιδιαίτερα κατά την διαδικασία επιλογής μαθημάτων, κάποιες html σελίδες παράγονται μεν δυναμικά, αλλά από συναρτήσεις Javascript και όχι μέσω JSPs. Συγκεκριμένα, με τον τρόπο αυτό γίνεται το πέραςμα των επιλογών του χρήστη από τα checkboxes επιλογής μαθημάτων (Εικόνα 6.6.α) στη φόρμα εισαγωγής προσωπικών στοιχείων (Εικόνα 6.6.γ). Ανάμεσα στα άλλα πραγματοποιείται χρήση της συνάρτησης write της Javascript σε νέο παράθυρο (Εικόνα 6.6.β), σε συνδυασμό με τις δυνατότητες που παρέχουν οι html φόρμες με χρήση των http-request parameters στον προγραμματιστή. Η όλη διαδικασία πραγματοποιείται μόνο εφόσον ο client προσπελάσει τη συγκεκριμένη σελίδα και υποβάλλει τη φόρμα των checkboxes και δεν επιβαρύνει καθόλου τον server, αλλά πρακτικά ούτε τον client, καθότι η δυναμικά παραγόμενη πληροφορία είναι η δυνατόν ελάχιστη, με τον server να έχει αναλάβει όλη την υπόλοιπη. Έχουμε, δηλαδή:

Σύστημα Ηλεκτρονικών Εγγραφών

Περιεχόμενα:

- Μαθήματα κορμού
- Μαθήματα ρούιν
- Υποχρεωτικά μαθήματα
- Μαθήματα επιλογής
- Α εξαμήνο
- Διαδικασία Εγγραφής

Φόρμα εγγραφής Α' ονομ. εξ. εγγραφής

1ο εξάμηνο, Υποχρεωτικά

Αριθμός record	Κωδικός μαθήματος	Τίτλος μαθήματος	Ώρες θεωρίας	Ακαδημαϊκό έτος	Επιλογή μαθήματος	info1a
1	3132	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	5	2001	<input checked="" type="checkbox"/>	
2	3035	ΗΛΕΚΤΡΟΛΟΓΙΚΟ ΣΧΕΔΙΟ	4	2001	<input checked="" type="checkbox"/>	
3	3131	ΜΑΘΗΜΑΤΑ ΑΝΑΛΥΣΗ (ΣΥΝΑΡΤΗΣΕΙΣ ΜΙΑΣ ΜΕΤΑΒΛΗΤΗΣ)	6	2001	<input checked="" type="checkbox"/>	
4	3020	ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΗΛΕΚΤΡ. ΥΠΟΛΟΓΙΣΤΩΝ	5	2001	<input checked="" type="checkbox"/>	
5	3049	ΦΥΣΙΚΗ(ΜΗΧΑΝΙΚΗ)	5	2001	<input checked="" type="checkbox"/>	

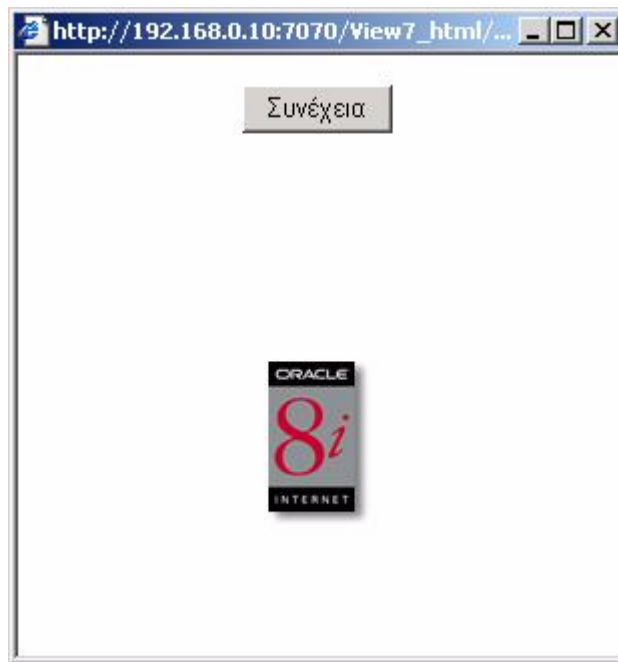
☐ Επιλέξτε όλα τα υποχρεωτικά μαθήματα του 1ου εξαμήνου

1ο εξάμηνο, κατ' επιλογήν υποχρεωτικά

Αριθμός record	Κωδικός μαθήματος	Τίτλος μαθήματος	Ώρες θεωρίας	Ακαδημαϊκό έτος	Επιλογή μαθήματος	info1b
1	3027	ΙΣΤΟΡΙΑ ΤΩΝ ΕΠΙΣΤΗΜΟΝΙΚΩΝ ΚΑΙ ΦΙΛΟΣΟΦΙΚΩΝ ΙΔΕΩΝ	2	2001	<input type="checkbox"/>	
2	3028	ΣΤΟΙΧΕΙΑ ΚΟΙΝΩΝΙΟΛΟΓΙΑΣ	2	2001	<input type="checkbox"/>	
3	3048	ΦΙΛΟΣΟΦΙΑ	2	2001	<input type="checkbox"/>	

Τα εδάφη Προσέλευση

Εικόνα 6.6.α



Εικόνα 6.6.β

http://192.168.0.10:7070/View7\_html/loginsession.jsp - Microsoft Internet Explorer

Address http://192.168.0.10:7070/View7\_html/loginsession.jsp

## Σύστημα Ηλεκτρονικών Εγγραφών

**Περιεχόμενα:**

- Μαθήματα κορμού
- Μαθήματα ροών
- Υποχρεωτικά μαθήματα
- Μαθήματα επιλογής
- Α εξάμηνο
- Διαδικασία Εγγραφής

### Εισαγωγή στοιχείων

Κωδικός συστήματος	*****
1. μάθημα	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ
2. μάθημα	ΗΛΕΚΤΡΟΛΟΓΙΚΟ ΣΧΕΔΙΟ
3. μάθημα	ΜΑΘΗΜΑΤ. ΑΝΑΛΥΣΗ (ΣΥΝΑΡΤΗΣΕΙΣ ΜΙΑΣ ΜΕΤΑΒΛΗΤΗΣ)
4. μάθημα	ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΗΛΕΚΤΡ. ΥΠΟΛΟΓΙΣΤΩΝ
5. μάθημα	ΦΥΣΙΚΗ (ΜΗΧΑΝΙΚΗ)
6. μάθημα	ΣΤΟΙΧΕΙΑ ΚΟΙΝΩΝΙΟΛΟΓΙΑΣ
7. μάθημα	---μη επιλεγμένο---
8. μάθημα	---μη επιλεγμένο---
9. μάθημα	---μη επιλεγμένο---
10. μάθημα	---μη επιλεγμένο---
11. μάθημα	---μη επιλεγμένο---
12. μάθημα	---μη επιλεγμένο---
13. μάθημα	---μη επιλεγμένο---
Ημερομηνία εγγραφής *	2001-09-27
Ημερομηνία υποβολής αίτησης	2001-09-15 18:48:56.0
A.M. φοιτητή	03096001
Ονομαστικό εξάμηνο εγγραφής	Κανονικό εξ. εγγραφής: 1
Επώνυμο	ΜΥΛΟΝΑΣ
Όνομα	ΦΟΙΒΟΣ
Όνομα Πατρός	ΙΩΑΝΝΗΣ

Εικόνα 6.6.γ



## 6.7 Υλοποίηση Javascript:

Η υλοποίηση όλων των client-side ελέγχων, όπως η ορθή εισαγωγή στοιχείων authentication, η ορθή και με το θεμιτό format εισαγωγή προσωπικών στοιχείων, η ορθή διαχείριση των records από τη γραμματεία, καθώς και της ίδιας της διαδικασίας πλοήγησης στο κάθε υποσύστημα, καθίσταται δυνατή με την κλήση συναρτήσεων Javascript σε αυτές. Έγινε συγγραφή περίπου 39000 γραμμών κώδικα Javascript σε διάφορα αρχεία, ως επί το πλείστον στο UltraEdit. Όπου απαιτείται τέτοιου είδους έλεγχος, δηλαδή σε οποιαδήποτε περίπτωση ύπαρξης φόρμας ή πεδίου προς συμπλήρωση από το χρήστη του συστήματος,, συμπεριλαμβάνονται κατάλληλα tags της μορφής:

```
<script src="/webapp/cabo/jslib/filename.js" language="Javascript">
</script>
```

στο head της html ή συνηθέστερα JSP σελίδας. Παράδειγμα υλοποίησης μιας τέτοιας σελίδας με κλήση συνάρτησης Javascript, αποτελεί κάθε σελίδα εισαγωγής προσωπικών στοιχείων του Συστήματος Ηλεκτρονικών Εγγραφών, όπως η παρακάτω (Εικόνα 6.7.α):

http://127.0.0.1:7070/View7\_html/loginsession.jsp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Print Encoding

Address http://127.0.0.1:7070/View7\_html/loginsession.jsp Go Links

### Σύστημα Ηλεκτρονικών Εγγραφών

- Υποχρεωτικά μαθήματα
- Μαθήματα επιλογής
- 1ο εξάμηνο
- 2ο εξάμηνο
- 3ο εξάμηνο
- 4ο εξάμηνο
- 5ο εξάμηνο
- 6ο εξάμηνο
- 7ο εξάμηνο
- 8ο εξάμηνο
- 9ο εξάμηνο
- Διαδικασία Εγγραφής

Ημερομηνία εγγραφής \*

Ημερομηνία υποβολής αίτησης 2001-09-29 18:43:40.0

A.M. φοιτητή 03096150

Ονομαστικό εξάμηνο εγγραφής Κανονικό εξ. εγγραφής: 1

Επώνυμο

Όνομα

Όνομα Πατρός

Οδός

Αριθμός

Περιοχή

Ταχ. κώδικας

Τηλέφωνο

e-mail

\* Σημείωση: Για εισαγωγή ημερομηνίας χρησιμοποιήστε το κουμπί δεξιά από το πεδίο

Υποβολή εγγραφής Καθάρισμα φόρμας

Done Internet

Εικόνα 6.7.α

## 6.8 Υλοποίηση Java:

Τέλος, το δυσκολότερο και παραγωγικότερο κομμάτι υλοποίησης της διπλωματικής εργασίας αποτέλεσε η συγγραφή περίπου 26000 γραμμών κώδικα Java, μη συμπεριλαμβανομένου ενός μεγάλου αριθμού υπάρχουσων βιβλιοθηκών ή imported αρχείων. Η Java χρησιμοποιήθηκε ως επί το πλείστον για τη διαδικασία πιστοποίησης των εκάστοτε χρηστών στη βάση του συστήματος, την ορθή πρόσβαση σε αυτή και την ορθή, λιτή και αισθητικά προσεγμένη απεικόνιση των δυναμικά παραγόμενων αποτελεσμάτων από την προηγούμενη διαδικασία πρόσβασης. Χαρακτηριστικό του μεγέθους εμπλοκής της στο όλο σύστημα είναι το γεγονός, ότι σχεδόν όλοι οι πίνακες μιας οποιασδήποτε δυναμικής σελίδας JSP παράγονται από κατάλληλες μεθόδους Java. Το τελευταίο προσδίδει όπως είναι φυσικό ύψιστη ευελιξία στο όλο σύστημα, μεταφερσιμότητα και βέλτιστη απόδοση, με εύκολα παραμετροποιήσιμους παράγοντες και δυνατότητες εύκολης αλλαγής υποκείμενων καίριων στοιχείων, όπως οι οντότητες της βάσης, χωρίς να απαιτείται ιδιαίτερη προγραμματιστική παρέμβαση κατά την περίπτωση αλλαγών ή /και προσθηκών στο όλο σύστημα μελλοντικά.

Επίσης, είναι προφανές ότι ανάμεσα στους πρωταρχικούς στόχους κατά την υλοποίηση του όλου εγχειρήματος ήταν και η κατά το δυνατόν επαναχρησιμοποίηση κώδικα κατά τη φάση ανάπτυξης του συστήματος, προκειμένου να επιταχυνθεί η παραγωγική διαδικασία και η όλη εξέλιξή του. Με τη χρήση της Java το παραπάνω επετεύχθη σε πολύ μεγάλο βαθμό. Ενδεικτικά και μόνο παρατίθεται η κλήση ενός τέτοιου component από μια JSP σελίδα και η υλοποίηση του κώδικα που παράγει το αποτέλεσμα. Το τελευταίο είναι η σχεδίαση ενός πίνακα με τα αποτελέσματα από τη βάση σε κάθε σχεδόν περίπτωση read-only πρόσβασης σε κάποια από αυτά (εκτός ίσως από κάποιες εντελώς εξειδικευμένες περιπτώσεις πρόσβασης που είχαν ελαφρά διαφορετικές απαιτήσεις και για τις οποίες χρησιμοποιήθηκε κατά κόρον η τεχνική του subclassing υπαρχόντων κλάσεων και η ιδιότητα της κληρονομικότητας). Παρατηρούμε στο αρχείο υλοποίησης του κώδικα ότι είναι ιδιαίτερα εμφανής η επαναχρησιμοποίηση κώδικα για την ανάπτυξη περαιτέρω components. Για περισσότερες πληροφορίες ο αναγνώστης καλείται να ανατρέξει στον κώδικα που επισυνάπτεται.

- *κλήση μέσω Java scriptlets στο περιεχόμενο της σελίδας*

```
<jsp:useBean id="RowSetBrowser" class="View7_html.myRowSetBrowser"
scope="request">
<%
    RowSetBrowser.initialize(pageContext,
"OLA_html_Olamazi_Package_Olamazi_PackageModule.Ola_kormou");
    RowSetBrowser.setVisibleRows(15);
    RowSetBrowser.setShowRecordNumbers(true);

    RowSetBrowser.setAttributeTitle("KM","Εὐἄἔἔῦῶ ἰᾶἔῖᾶῶἰῶ");
    RowSetBrowser.setAttributeTitle("Titlos","Ὀβῶἔἰῶ ἰᾶἔῖᾶῶἰῶ");
    RowSetBrowser.setAttributeTitle("Omada","Εὐἄἔἔῦῶ ἰῖῦᾶᾶῶ
'ἰᾶἔῖᾶῶἰῶ');
    RowSetBrowser.setAttributeTitle("Ej","Ἄἰῦἰῑῑἰ");

    RowSetBrowser.setReleaseApplicationResources(true);
    RowSetBrowser.render();
%>
```

</jsp:useBean>

- **κώδικας υλοποίησης**

```
// Author: Phivos Mylonas - © 2001
package View7_html;
```

```
import java.io.PrintWriter;
import java.io.OutputStream;
import java.util.Vector;
import java.util.Enumeration;
import java.util.StringTokenizer;
import java.util.Hashtable;
import oracle.jbo.*;
import java.lang.reflect.*;
import oracle.jbo.domain.TypeFactory;
import oracle.jdeveloper.html.*;
import oracle.jdeveloper.jsp.wb.*;
import oracle.jbo.html.databeans.*;

public class myRowSetBrowser extends DataWebBeanImpl
{
    protected int                nVisibleRows = 10;
    protected boolean            bAlternateColors = true;
    protected String             sRowUrl = null;
    protected boolean            bShowCurrentRow = true;
    protected String             sRowUrlTarget = null;
    protected Vector             extraUrlCols = new Vector();
    protected boolean            useRoundedCorners = true;
    protected boolean            showRecordNumber = false;
    protected Hashtable colTitles = new Hashtable();
    protected String             sRowEditUrl = null;
    protected String             ImageDir = "/webapp/images/";
    protected String             sRowEditTarget = null;
    protected boolean            nVisibleRowsWasChanged = false;
    protected TableControl aTable = null;

    public myRowSetBrowser()
    {
    }

    public void setImageDir(String sDir)
    {
        ImageDir = sDir;
    }

    public String getImageDir()
    {
        return ImageDir;
    }

    public void setRowEditUrl(String sUrl)
    {
        sRowEditUrl = sUrl;
    }

    public void setRowEditTargetWindow(String sName)
    {
    }
}
```

```

        sRowEditTarget = sName;
    }

    public void setAttributeTitle(String sAttribute , String sTitle)
    {
        colTitles.put(sAttribute , sTitle);
    }

    public void setShowRecordNumbers(boolean bShow)
    {
        showRecordNumber = bShow;
    }

    public void setUseRoundedCorners(boolean bSet)
    {
        useRoundedCorners = bSet;
    }

    public void addTextUrlColumn(String sTitle, String sText, String
sUrl , String sTarget)
    {
        extraUrlCols.addElement(new TextColumnUrlInfo(sTitle, sText,
sUrl , sTarget));
    }

    public void addTextUrlColumn(String sTitle, String sText, String
sUrl)
    {
        addTextUrlColumn(sTitle , sText, sUrl , null);
    }

    public void addImageUrlColumn(String sTitle, String sImageFile,
String sUrl , String sTarget)
    {
        extraUrlCols.addElement(new ImageUrlInfo(sTitle, sImageFile,
sUrl , sTarget));
    }

    public void addImageUrlColumn(String sTitle, String sImageFile,
String sUrl)
    {
        addImageUrlColumn(sTitle, sImageFile, sUrl , null);
    }

    public void setRowUrlTargetWindow(String sUrl)
    {
        sRowUrlTarget = sUrl;
    }

    public String getRowUrlTargetWindow()
    {
        return sRowUrlTarget;
    }

    public void setRowUrl(String sUrl)
    {
        sRowUrl = sUrl;
    }

```

```

public String getRowUrl()
{
    return sRowUrl;
}

public void setShowCurrentRow(boolean bSet)
{
    bShowCurrentRow = bSet;
}

public boolean getShowCurrentRow()
{
    return bShowCurrentRow;
}

public void setVisibleRows(int nRows)
{
    nVisibleRows = nRows;
    nVisibleRowsWasChanged = true;
}

public int getVisibleRows()
{
    return nVisibleRows;
}

public void setAlternateColors(boolean bSet)
{
    bAlternateColors = bSet;
}

public TableControl getTableControl()
{
    if(aTable == null)
        aTable = new TableControl();
    return aTable;
}

protected void renderTableHeaders() throws Exception
{
    int          attrNo;
    String       sHeader;
    TableControl aTable = getTableControl();
    AttributeDef[] dattrs = getDisplayAttributeDefs();
    int          RowTag;

    if(showRecordNumber)
        aTable.addHeader("Ñéèìüò record");

    if(sRowEditUrl != null)
    {
        aTable.addHeader("Action");
    }

    for(attrNo = 0; attrNo < dattrs.length ; attrNo++)
    {
        if(!shouldDisplayAttribute(dattrs[attrNo]))
            continue;

        sHeader = dattrs[attrNo].getName();
    }
}

```

```

        sHeader = getAttributeLabel(qView, dattrs[attrNo]);

        // αν ο χρήστης θέλει διαφορετικό τίτλο
        if(colTitles.get(sHeader) != null)
            sHeader = (String)colTitles.get(sHeader);

        aTable.addHeader(sHeader);
    }

    // προσθήκη headers για παραπάνω στήλες
    for(int nExtra = 0 ; nExtra < extraUrlCols.size() ; nExtra++)
    {
        TextColumnUrlInfo info =
(TextColumnUrlInfo)extraUrlCols.elementAt(nExtra);
        aTable.addHeader(info.getTitle());
    }
}

protected boolean initializeNewTableRow(HTMLTableRow htmlRow,
Row[] drows, int rowno, boolean isEven)
{
    String sRowKey;
    int RowTag;
    Row currentRow = qView.getCurrentRow();

    sRowKey = getRowKey(drows[rowno]);

    if(bAlternateColors)
    {
        if(isEven)
        {
            htmlRow.setCSSClassName("clsEvenTableRow");
            isEven = false;
        }
        else
        {
            htmlRow.setCSSClassName("clsOddTableRow");
            isEven = true;
        }
    }

    if(bShowCurrentRow && currentRow == drows[rowno])
    {
        htmlRow.setCSSClassName("clsCurrentTableRow");
    }

    if(showRecordNumber)
    {
        if(sRowUrl != null)
        {
            RowTag = qView.getRangeStart() + rowno + 1;

            int nParamIndex = sRowUrl.indexOf('?');
            String sUrl = sRowUrl;

            sUrl = fixupUrl(sRowUrl, qView.getName() + "_ROWKEY="
+ sRowKey);

            HTMLTextURL txtUrl = new HTMLTextURL(" " + RowTag,
sUrl);

```

```

        if(sRowUrlTarget != null)
        {
            txtUrl.setTarget(sRowUrlTarget);
        }

        htmlRow.addTextURL(txtUrl);
    }
    else
    {
        RowTag = qView.getRangeStart() + rowno + 1;

        htmlRow.addFormattedTextCell("" + RowTag);
    }
} //if(showRecordNumber)

if(sRowEditUrl != null)
{
    String sImage = null;

    if(qView.getViewObject().isReadOnly())
    {
        sImage = ImageDir + "/readonly.gif";
    }
    else
    {
        sImage = ImageDir + "/editrec.gif";
    }

    String sUrl = fixupUrl(sRowEditUrl, qView.getName() +
"_ROWKEY=" + sRowKey);

    HTMLImageURL imgUrl = new HTMLImageURL(sImage, sUrl);

    if(sRowEditTarget != null)
    {
        imgUrl.setTarget(sRowEditTarget);
    }
    htmlRow.addImageURL(imgUrl);
}
return isEven;
}

public void render() throws Exception
{
    int attrNo;
    String sHeader;
    TableControl aTable = getTableControl();
    AttributeDef[] dattrs =
getDisplayAttributeDefs();
    int RowTag;

    aTable.initialize(application, session, request,
response, out);
    aTable.setUseRoundedCorners(useRoundedCorners);
    aTable.setImageBase(this.getImageDir());

    renderTableHeaders();

    int nRangeSize = qView.getRangeSize();

```

```

        // override μόνο αν το θέλει ο χρήστης
        if(nVisibleRowsWasChanged    &&    qView.getRangeSize()    !=
nVisibleRows)
        {
            qView.setRangeSize(nVisibleRows);
        }

        Row[]    drows = qView.getAllRowsInRange();
        Row    currentRow = qView.getCurrentRow();
        String    sValue = null;
        boolean isEven = true;
        String    sRowKey;

        for(int rowno = 0; rowno < drows.length; rowno++)
        {
            sRowKey = getRowKey(drows[rowno]);

            HTMLTableRow    htmlRow = new HTMLTableRow();

            aTable.addRow(htmlRow);

            isEven = initializeNewTableRow(htmlRow,    drows,    rowno,
isEven);

            for (int j = 0; j < dattrs.length; j++)
            {
                if(!shouldDisplayAttribute(dattrs[j]))
                    continue;

                sValue = null;

                if(drows[rowno] != null)
                {
                    Object    attrObj    =    drows[rowno].getAttribute(
dattrs[j].getIndex());

                    if( attrObj != null)
                        sValue = attrObj.toString();
                }

                //htmlRow.addCell(aRenderer.getAttributeHtml(dattrs[j].getName(),
sValue));

                HTMLFieldRenderer    rField    =
this.getDisplayFieldRenderer(drows[rowno],dattrs[j]);
                String sOutput = rField.renderToString(ctx,    qView,
drows[rowno] ,    dattrs[j].getName());
                htmlRow.addCell(sOutput);
            }

            // προσθήκη extra columns
            int RowRangeIndex = qView.getRangeIndexof(drows[rowno]);

            for(int nExtra = 0 ; nExtra < extraUrlCols.size() ;
nExtra++)
            {
                TextColumnUrlInfo    info    =
(TextColumnUrlInfo)extraUrlCols.elementAt(nExtra);
                String sUrl = fixupUrl(info.sUrl,    qView.getName() +
"_ROWKEY=" + sRowKey);

```



```

        htmlRow.addElement(info.getColumnElement(sUrl));
    }

    htmlRow = new HTMLTableRow();
}

aTable.render();

// restore προηγούμενο rangesize
if(nVisibleRowsWasChanged)
    qView.setRangeSize(nRangeSize);

releaseApplicationResources();
}

public String fixupUrl(String sUrl , String sExtra)
{
    int    nParamIndex = sUrl.indexOf('?');
    String sNewUrl = sUrl;

    if(nParamIndex != -1)
    {
        sNewUrl = sUrl + "&" + sExtra;
    }
    else
    {
        sNewUrl = sUrl + "?" + sExtra;
    }
    return sNewUrl;
}
}

```

Αριθμός record	Κωδικός μαθήματος	Εξάμηνο	Τίτλος μαθήματος	Κωδικός ομάδας 'μαθήματος'
1	3037	01	ΑΓΓΛΙΚΗ ΓΛΩΣΣΑ	
2	3038	01	ΓΑΛΛΙΚΗ ΓΛΩΣΣΑ	
3	3038	01	ΓΑΛΛΙΚΗ ΓΛΩΣΣΑ ΚΑΙ ΤΕΧΝΙΚΗ ΟΡΟΛΟΓΙΑ	
4	3086	01	ΓΕΡΜΑΝΙΚΗ ΓΛΩΣΣΑ	
5	3132	01	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	
6	3035	01	ΗΛΕΚΤΡΟΛΟΓΙΚΟ ΣΧΕΔΙΟ	
7	3027	01	ΙΣΤΟΡΙΑ ΤΩΝ ΕΠΙΣΤΗΜΟΝΙΚΩΝ ΚΑΙ ΦΙΛΟΣΟΦΙΚΩΝ ΙΔΕΩΝ	2011
8	3277	01	ΙΣΤΟΡΙΚΗ ΜΟΥΣΙΚΟΛΟΓΙΑ ΚΑΙ ΘΕΩΡΗΤΙΚΑ ΜΟΥΣΙΚΗΣ	
9	3087	01	ΙΤΑΛΙΚΗ ΓΛΩΣΣΑ	
10	3131	01	ΜΑΘΗΜΑΤ ΑΝΑΛΥΣΗ (ΣΥΝΑΡΤΗΣΕΙΣ ΜΙΑΣ ΜΕΤΑΒΛΗΤΗΣ)	
11	3020	01	ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΗΛΕΚΤΡ. ΥΠΟΛΟΓΙΣΤΩΝ	
12	3028	01	ΣΤΟΙΧΕΙΑ ΚΟΙΝΩΝΙΟΛΟΓΙΑΣ	2011
13	3048	01	ΦΙΛΟΣΟΦΙΑ	2011
14	3049	01	ΦΥΣΙΚΗ(ΜΗΧΑΝΙΚΗ)	
15	3163	01	ΧΗΜΕΙΑ	

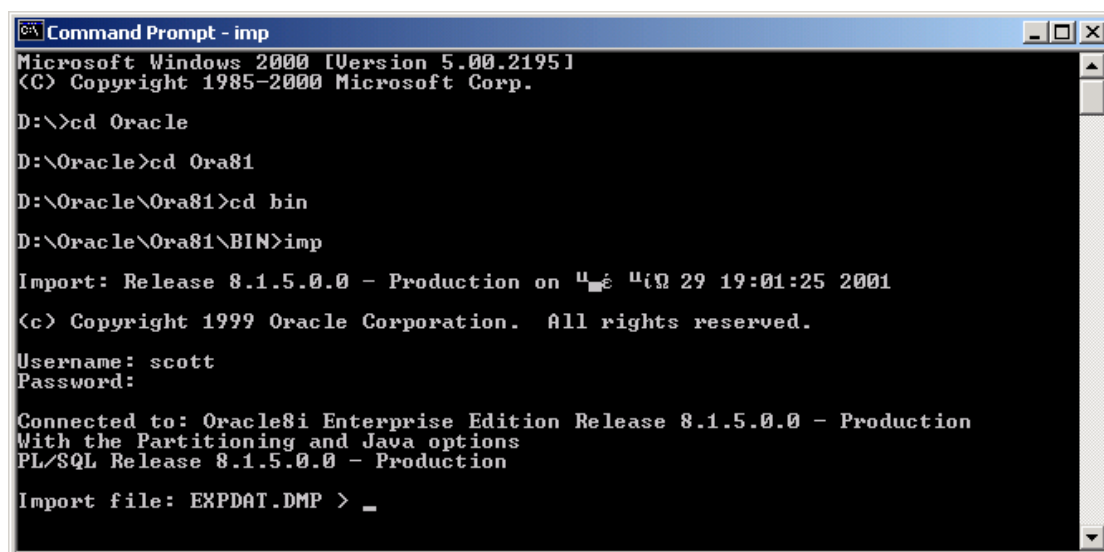
Εικόνα 6.8.α

## 6.9 Υλοποίηση επικοινωνίας με βάση φοιτητολογίου

Στην πρώτη φάση ανάπτυξης του Συστήματος Ηλεκτρονικών Εγγραφών η απαραίτητη επικοινωνία με τη βάση του φοιτητολογίου πραγματοποιείται με τη χρήση εργαλείων export και import της Oracle8i, όπως τα **exp** και **imp**, τα οποία και έρχονται ενσωματωμένα στο βασικό πακέτο της. Απαιτείται συνεπώς η ύπαρξη ενός database administrator, ο οποίος ανάμεσα στα λοιπά καθήκοντα του έχει αναλάβει και την παραπάνω διαδικασία. Η τελευταία σε γενικές γραμμές συνοψίζεται στη χρήση ενός αρχείου ελέγχου (π.χ. oratables.dat), το οποίο «διαβάζει» το utility και ακολουθεί γραμμή προς γραμμή τις εντολές που αυτό περικλείει. Παράδειγμα τέτοιου αρχείου αποτελεί το επόμενο:

```
FILE=orawin2k_scott_tables.dmp
LOG=orawin2k_scott_tables.dat.log
TABLES=(KATEF, LANG, METAB, OMADAEP1, OMADPEM, PLTAB, PROAIRET, PROYPDIL, RO
HDEF, ROHFOIT, ST_F, ST_M, TESTV, TOMEIS, VIEW3, VIEW3SECOND, VIEW5, VIEW5SECO
ND, YPCOURSE)
```

Παρατηρούμε ότι καθορίζεται το αρχείο orawin2k\_scott\_tables.dmp στο οποίο εν προκειμένω θα πραγματοποιηθεί το dump των επιλεγμένων πινάκων (καθώς πρόκειται για αρχείο ελέγχου export utility), το αρχείο orawin2k\_scott\_tables.dat.log που αποτελεί το αρχείο log της όλης διαδικασίας export και ακολουθούν διάφορες παράμετροι πληροφοριών προς το utility, οι οποίες εδώ συνοψίζονται στην εξής μία, η οποία αφορά τους πίνακες που θα γίνουν export<sup>7</sup>.



Εικόνα 6.9.α

<sup>7</sup> Στο παραπάνω παράδειγμα, πραγματοποιήθηκε import των exported tables σε ανάλογη βάση Oracle8i, οπότε δεν ήταν απαραίτητος ο ορισμός επιπλέον παραμέτρων στο αρχείο ελέγχου. Σε περίπτωση όμως flat text file, προοριζόμενου για το φοιτητολόγιο, είναι αυτονόητο ότι απαιτείται πιο λεπτομερής ανάλυση και περιγραφή χαρακτηριστικών στο αρχείο ελέγχου.

Αντίστοιχη διαδικασία ακολουθείται και στην περίπτωση import στοιχείων στη βάση μας, όπως π.χ. έγινε κατά την αρχική είσοδο των απαιτούμενων data από το φοιτητολόγιο στη βάση.

## 6.10 Υλοποίηση hardware-OS:

Οι λειτουργικές παράμετροι ανάπτυξης και ελέγχου που χρησιμοποιήθηκαν συνοψίζονται στα εξής:

	Φάση ανάπτυξης λογισμικού συστήματος	Τελική φάση ελέγχου ορθής λειτουργίας
<b>processor</b>	Intel Pentium III 800 MHz	Intel Pentium III 800 MHz
<b>RAM</b>	512 MB	512 MB
<b>disk space</b>	5 GB	5 GB
<b>VGA card</b>	Matrox DH 32 MB	Matrox DH 32 MB
<b>monitor</b>	Nokia 446 Pro 19"	Nokia 446 Pro 19"
<b>OS</b>	Windows 2000®	Redhat Linux 6.2

## 6.11 Υλοποίηση BIND:

Αρχικά γίνεται το request σε μορφή URL από τον client (<http://www.dblab.ntua.gr/>), αυτό περνάει από διάφορους name-servers και τελικά φτάνει στον cluster-DNS-server του συστήματος (IP num), ο οποίος με βάση κατάλληλο αλγόριθμο επιλογής server πραγματοποιεί το address mapping του URL στην τελική IP διεύθυνση του server που θα εξυπηρετήσει την αίτηση, στέλνοντας πίσω στον client την πληροφορία. Κατόπιν ο client ξεκινάει νέο request απευθείας στον server, την IP του οποίου πλέον γνωρίζει. Σε επίπεδο URL λοιπόν επιτυγχάνεται η θεμιτή διαφάνεια και η όλη διαδικασία «ασχολείται» μόνο με IP διευθύνσεις.

Σε επίπεδο υλοποίησης επιλέχθηκε η επιλογή του Berkley Internet Name Daemon (B.I.N.D.), με υλοποίησή του σε εξειδικευμένο server, που θα αναλάβει τη διαδικασία dispatching των requests στους τέσσερις web-servers, καθένας από τους οποίους θα διαθέτει το δικό του ξεχωριστό IP address. Το BIND χρησιμοποιεί Round-Robin αλγόριθμο επιλογής server και υλοποιείται ορίζοντας πολλαπλά canonical names, μετατρέποντας το π.χ. [www.foo.dom](http://www.foo.dom) σε alias στο wwwX.foo.dom, όπως φαίνεται και στο παρακάτω παράδειγμα:

```
www.foo.dom.      IN  CNAME  www1.foo.dom
                  IN  CNAME  www2.foo.dom
                  IN  CNAME  www3.foo.dom
                  IN  CNAME  www4.foo.dom
                  IN  CNAME  www5.foo.dom
                  IN  CNAME  www6.foo.dom
```

Η μοναδική δύσκολη απόφαση στο όλο configuration είναι η επιλογή του ιδανικού TTL. Κι αυτό γιατί πολύ μεγάλο TTL θα είχε ως αποτέλεσμα αφενός μεν την ελάττωση της κίνησης στο δικό μας σύνδεσμο, αφετέρου δε την διατήρηση για υπερβολικό χρόνο στις caches των άλλων DNS servers της πληροφορίας μας, με αποτέλεσμα να έχουμε κακή κατανομή της στο «web cluster». Αν πάλι τεθεί ιδιαίτερα μικρή τιμή στο TTL θα έχουμε αύξηση του φορτίου και του χρόνου απόκρισης στο request του πελάτη, καθότι οι υπόλοιποι DNS servers θα κάνουν expire πολύ σύντομα της πληροφορίας για το δικό μας σύστημα, οπότε θα πρέπει να την ανακαλύπτουν και συχνότερα, αλλά θα έχουμε σαφώς καλύτερη κατανομή της http κίνησης σε αυτό. Σε πρακτικά συστήματα του μεγέθους του Σ.Η.Ε. ένα TTL της τάξεως της 1 ώρας θεωρείται ικανοποιητικό.

## 6.12 Υλοποίηση firewall:

Το firewall του όλου συστήματος στηρίχθηκε αποκλειστικά στη χρήση κανόνων ipchains, μέσω των οποίων επιτυγχάνεται η χρήση συγκεκριμένων μόνο ports του μηχανήματος που είναι συνδεδεμένο στο δίκτυο δεδομένων. Παράδειγμα τμήματος τέτοιων κανόνων αποτελούν οι παρακάτω, οι οποίοι και μπλοκάρουν το μεγαλύτερο μέρος της κίνησης σε unknown και άχρηστα για το παρεχόμενο service ports, και αφήνουν ελεύθερα να περάσουν μόνο συγκεκριμένα πακέτα σε συγκεκριμένες χρήσιμες για το σύστημα πόρτες, όπως η πόρτα του απλού http και του https. Επιπρόσθετα, γίνεται καταγραφή όλης των εισερχόμενων και εξερχόμενων συμβάντων, ενεργοποιώντας κατάλληλη παράμετρο του ipchains, όπως μπορεί κανείς να παρατηρήσει στην τελευταία σειρά.

```
[root@mpriki firewall]# cat rules
#!/bin/sh

COMMAND=/sbin/ipchains
MYIP=147.102.12.2/32
MYNET=147.102.12.0/24

# deny everything

$COMMAND -F input
$COMMAND -F output
$COMMAND -F forward

$COMMAND -P input REJECT
$COMMAND -P output ACCEPT
$COMMAND -P forward DENY

#loopback interface
$COMMAND -A input -i lo -j ACCEPT

# www outgoing
$COMMAND -A input -p tcp -d $MYIP -s 0.0.0.0/0 www -j ACCEPT

# www incoming
$COMMAND -A input -p tcp -d $MYIP www -j ACCEPT

# https incoming
$COMMAND -A input -p tcp -d $MYIP https -j ACCEPT
```

```
# logging
$COMMAND -A input -l
```

## 6.13 Υλοποίηση SSL:

Όπως ήδη έχει αναφερθεί το όλο σύστημα στηρίχθηκε από πλευράς web server σε έναν Apache + Tomcat με δυνατότητες SSL κρυπτογράφησης. Το κύριο έργο της SSL μετατροπής αναλαμβάνει ο Apache web server με τη χρήση κατάλληλων modules, όπως τα mod\_ssl και mod\_jk, και tags στο configuration file του Apache. Παράδειγμα τμήματος τέτοιου setup του Apache αποτελεί και το ακόλουθο:

```
JkWorkersFile /etc/httpd/conf/workers.properties
JkLogFile /var/log/httpd/mod_jk.log
JkLogLevel warn

<VirtualHost _default_:443>
    SSLEngine on
    SSLCipherSuite
    ALL:!ADH:!EXP56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL

    # SSL stuff

    Alias /BC4J "/usr/local/tomcat/webapps/BC4J"
    <Directory "/usr/local/tomcat/webapps/BC4J">
        Options Indexes FollowSymLinks
    </Directory>
    JkMount /BC4J/servlet/* ajp13
    JkMount /BC4J/*.jsp ajp13

    <Location "/BC4J/WEB-INF/">
        AllowOverride None
        Deny from all
    </Location>

</VirtualHost>
```

Τα παραπάνω θα έχουν ως αποτέλεσμα το σερβίρισμα όλων των σελίδων της εφαρμογής κάτω από το συγκεκριμένο virtual host (εν προκειμένω της εφαρμογής ΣΗΕ) στον συγκεκριμένο server (χρήση actual paths), κρυπτογραφημένες πάνω από SSL και συγκεκριμένα με αλλαγή των urls τους σε «https://» αντί για «http://».

Να σημειώσουμε εδώ ότι η χρήση SSL πάνω από http απαιτεί εξ ορισμού την ύπαρξη ενός SSL certificate. Η παραγωγή ενός τέτοιου certificate μπορεί να γίνει με απλό τρόπο μέσω του Tomcat και με χρήση των [OpenSSL](#) (OpenSource SSL implementation) ή/και [ModSSL](#) (SSL support for Apache) projects.



# 7

## *Εγχειρίδιο χρήσης - Manual*

### *7.1 Εισαγωγή*

Το Σύστημα Ηλεκτρονικών Εγγραφών (ΣΗΕ) είναι ένα πληροφοριακό σύστημα που παρέχει στους αντίστοιχους χρήστες τη δυνατότητα για πλήρη απεικόνιση, πλοήγηση, αναζήτηση και εγγραφή μαθημάτων του Τμήματος Ηλεκτρολόγων Μηχανικών & Μηχανικών Η/Υ, τη δυνατότητα πλήρους ελέγχου της όλης διαδικασίας και παρεμβάσεων σε αυτή, καθώς και τη δυνατότητα εποπτικού ελέγχου του ίδιου του συστήματος.

Το σύστημα είναι προσβάσιμο στον καθένα μέσω του Internet στην ηλεκτρονική διεύθυνση:

<http://www.dblab.ntua.gr/~she>

ή εναλλακτικά:

[http://jazz.dblab.ntua.gr:8080/examples/BC4J/View7\\_html/index.html](http://jazz.dblab.ntua.gr:8080/examples/BC4J/View7_html/index.html)

Στην παρούσα φάση του συστήματος, ανάλογα την ομάδα στην οποία ανήκει ο εκάστοτε χρήστης, είναι σε θέση να προσπελάσει και αντίστοιχες κατηγορίες περιεχομένου:

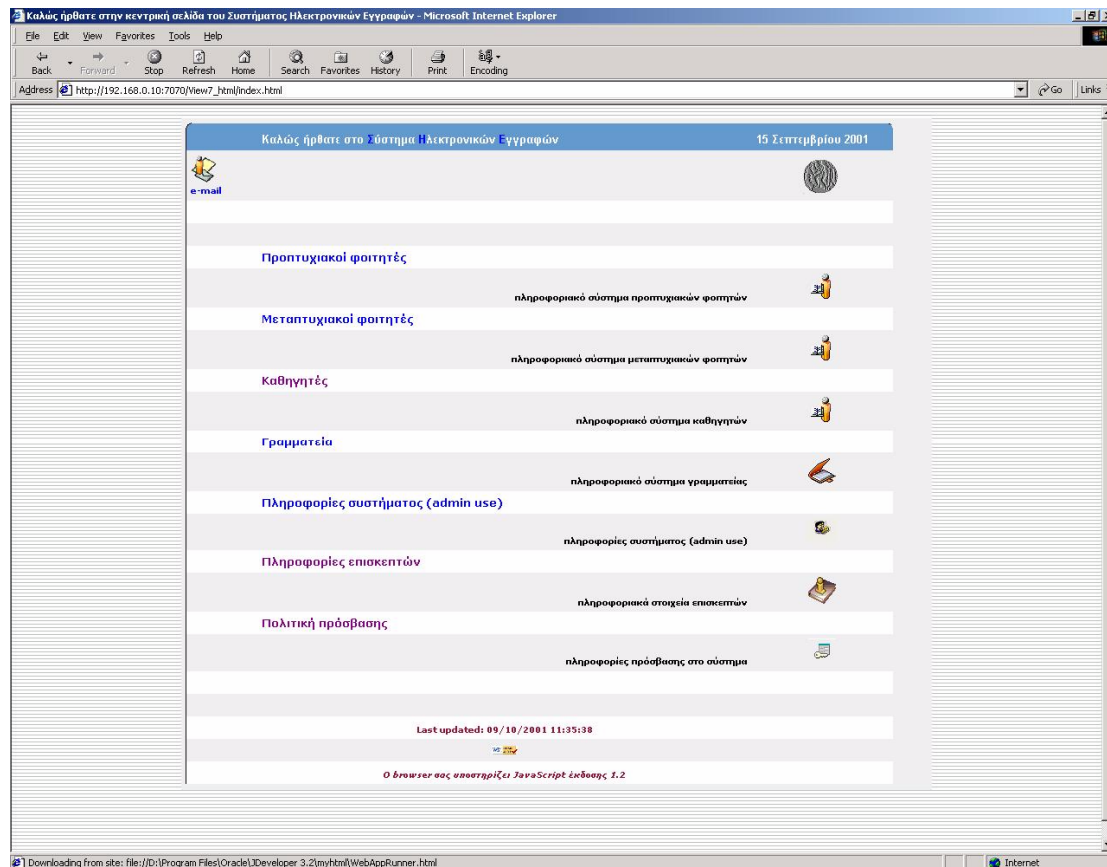
- Οι προπτυχιακοί φοιτητές μπορούν να πληροφορηθούν αναλυτικά στοιχεία για οποιοδήποτε προπτυχιακό μάθημα του Τμήματος, για το προπτυχιακό πρόγραμμα σπουδών, για τη διάρθρωση των ροών και των κατευθύνσεων, καθώς και άλλες γενικές και οργανωτικές πληροφορίες.

- Το προσωπικό της γραμματείας δύναται να προσπελάσει όλα τα παραπάνω στοιχεία και επιπλέον ξεχωριστά μενού αναζητήσεων, διορθώσεων και ελέγχου των εγγραφών, τόσο σε μαθήματα εξαμήνων, όσο και σε περιπτώσεις επιλογής ροών και/ή κατευθύνσεων.
- Ο administrator του όλου συστήματος έχει πρόσβαση σε εποπτικού περιεχομένου πληροφορίες της βάσης δεδομένων που χρησιμοποιείται, αλλά και των επιμέρους συστημάτων υλοποίησής του.
- Μελλοντικά ουσιαστική πρόσβαση στο σύστημα αναμένεται να έχουν οι μεταπτυχιακοί φοιτητές, καθώς και οι καθηγητές του Τμήματος, οι τελευταίοι κυρίως προς άντληση στατιστικών στοιχείων.

Ακολουθεί μια πλήρης περιγραφή των λειτουργιών του συστήματος, καθώς και τον τρόπο πλοήγησης στα διάφορα υπομενού του.

## 7.2 Είσοδος στο Σύστημα – κεντρική Σελίδα

Η είσοδος στο ΣΗΕ επιτυγχάνεται με την εισαγωγή της διεύθυνσης [http://jazz.dblab.ntua.gr:8080/examples/BC4J/View7\\_html/index.html](http://jazz.dblab.ntua.gr:8080/examples/BC4J/View7_html/index.html) σε έναν οποιοδήποτε web browser. Για περαιτέρω πλοήγηση στο σύστημα απαιτείται η χρήση Javascript (ενημερώνεται κατάλληλα ο χρήστης για την έκδοση που υποστηρίζει ο browser του) και ως εκ τούτου προτείνουμε τη χρήση ενός εκ των Internet Explorer 5.5 και Netscape 6.1.

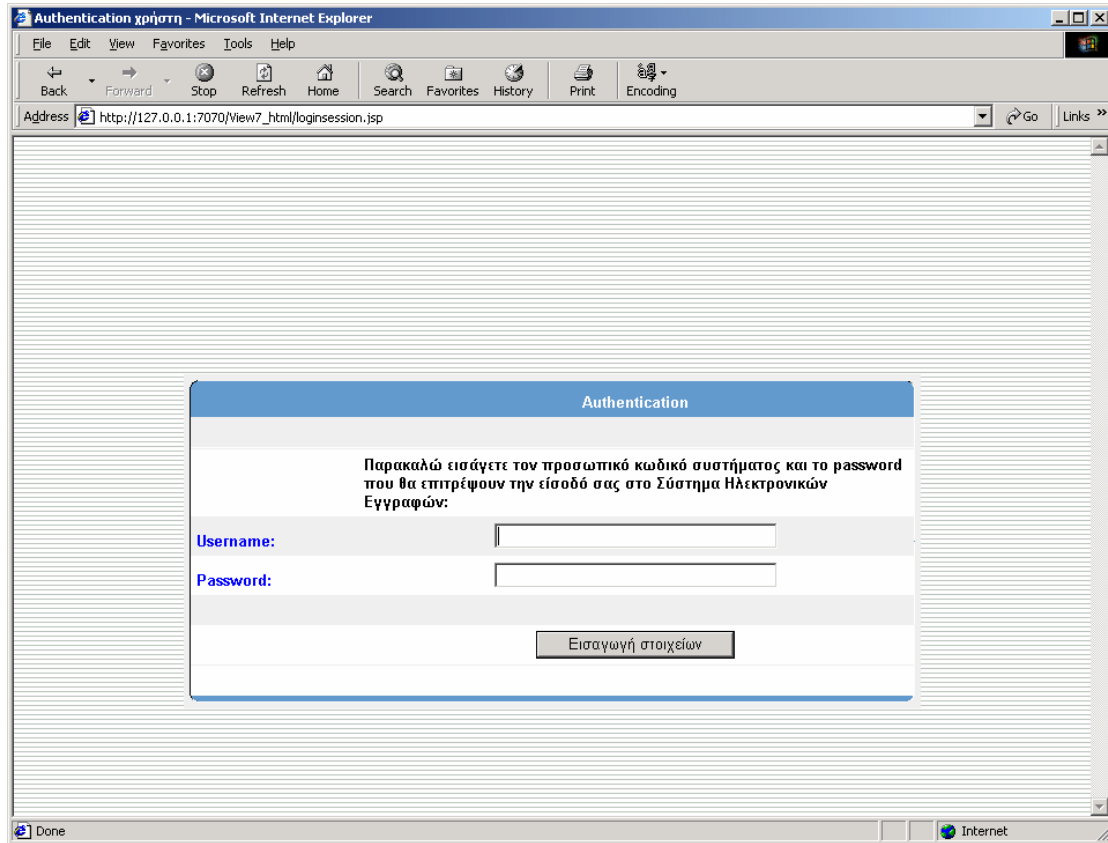


Εικόνα 7.2.α



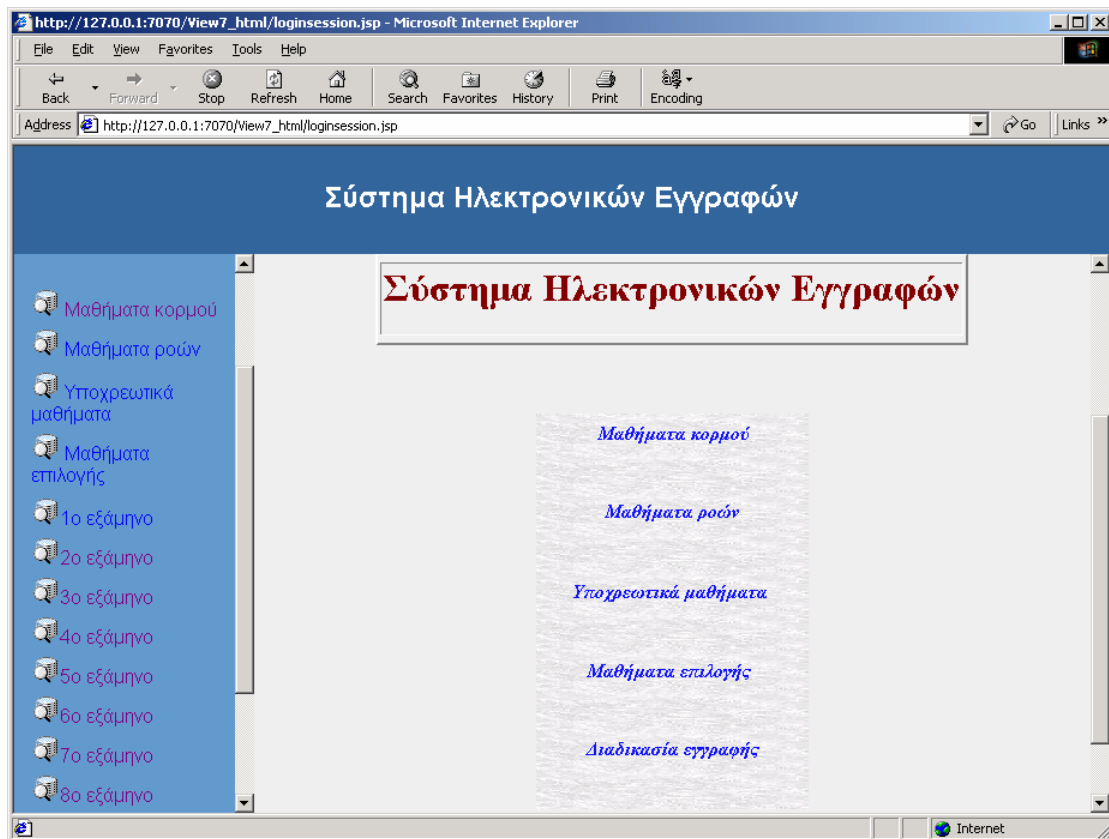
### 7.3 Υποσύστημα προπτυχιακών φοιτητών

Ας θεωρήσουμε αρχικά την περίπτωση που πραγματοποιεί login ένας προπτυχιακός φοιτητής.



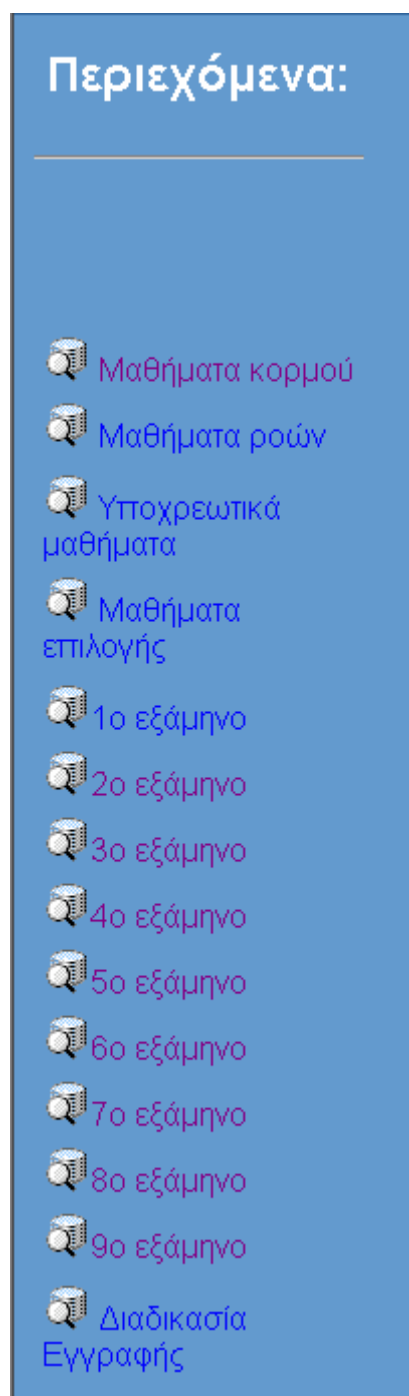
Εικόνα 7.3.α

Αμέσως μετά το επιτυχημένο login του θα βρεθεί στην επόμενη σελίδα:



Εικόνα 7.3.β

Στο αριστερό από τα δύο κάθετα frames ο χρήστης έχει τη δυνατότητα να επιλέξει links από τα περιεχόμενα που θα τον οδηγήσουν στις αντίστοιχες σελίδες του υποσυστήματος στο δεξιό frame. Με τον παραπάνω τρόπο καθίσταται ελεύθερη η βούληση του χρήστη, ανά πάσα στιγμή να προσπελάσει οποιαδήποτε σελίδα από τα περιεχόμενα επιθυμεί.



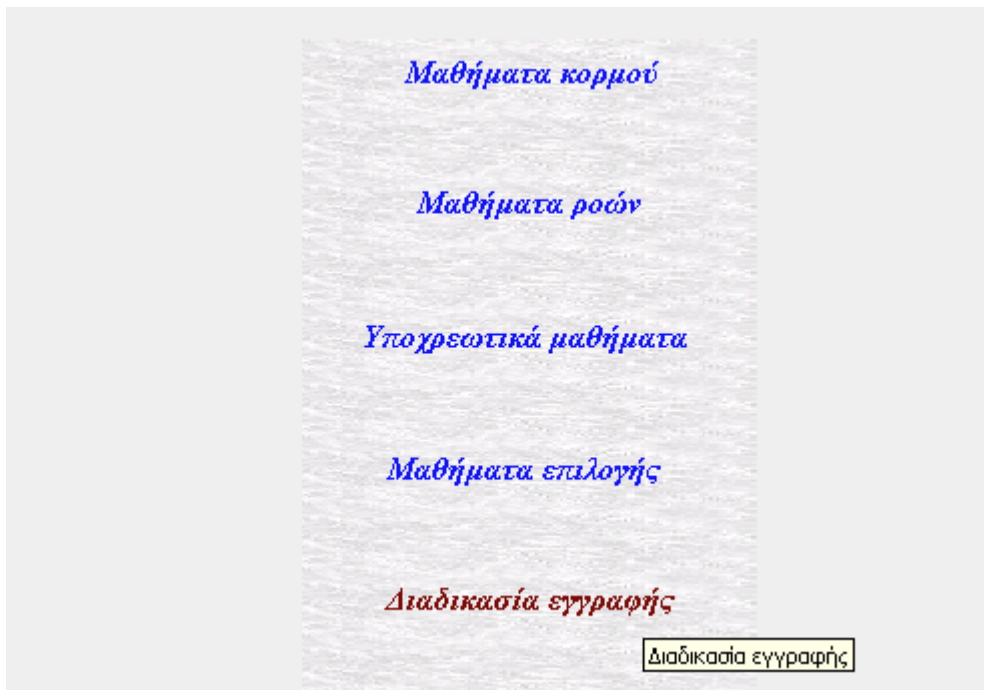
Εικόνα 7.3.γ

Στο δεξιό frame παρατηρούμε τις ίδιες δυνατότητες πλοήγησης με κάποια επιπλέον πληροφοριακά και αισθητικά στοιχεία. Το κεντρικό menu επιλογών του συστήματος είναι ορατό στο κέντρο της σελίδας.



Εικόνα 7.3.δ

Ο χρήστης μπορεί, μετακινώντας το δείκτη του ποντικιού στην επιθυμητή περιοχή, να επιλέξει να επισκεφτεί μια από τις πέντε θεματικές περιοχές. Για παράδειγμα, στην παρακάτω εικόνα φαίνεται να επιλέγεται η περιοχή της διαδικασίας εγγραφής:



Εικόνα 7.3.ε

Παρατηρούμε επίσης πως η επιλογή αλλάζει χρώμα, υποδεικνύοντας ότι έχει επιλεγθεί.

Ξεκινώντας από πάνω προς τα κάτω ο χρήστης έχει τη δυνατότητα να προσπελάσει με οποιαδήποτε σειρά τις εξής σελίδες-θεματικές περιοχές:

- Μαθήματα κορμού
- Μαθήματα ρωών
- Υποχρεωτικά μαθήματα
- Μαθήματα επιλογής
- Διαδικασία εγγραφής

Τέλος, μέσω του αριστερού frame δίνεται η δυνατότητα στο χρήστη να ανοίξει σε καινούριο παράθυρο το εξάμηνο που τον ενδιαφέρει, όπου και με τη μορφή καρτελών είναι σε θέση να προσπελάσει διάφορες χρήσιμες, κατά τη διαδικασία πλοήγησής του στο σύστημα, πληροφορίες για τα μαθήματά του:

The screenshot shows a web browser window with the address bar displaying 'http://127.0.0.1:7070/View7\_html/Eksamino\_A\_JSTab.jsp - Microsoft Internet Explorer'. The page has a blue header with navigation tabs: 'Α εξάμηνο:', 'Υποχρεωτικά μαθήματα', 'Κατ' επιλογήν υποχρεωτικά μαθήματα', 'Προαιρετικά Α' εξαμήνου', and 'Όλα μαζί'. Below the tabs, a message reads 'Κείμενο βοήθειας: Βρίσκεστε στο Α εξάμηνο!'. The main content area is titled 'Υποχρεωτικά μαθήματα Α εξαμήνου' and contains a table with the following data:

Κωδικός μαθήματος	Τίτλος μαθήματος	Ώρες θεωρίας	Ακαδημαϊκό έτος
3132	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	5	2001
3035	ΗΛΕΚΤΡΟΛΟΓΙΚΟ ΣΧΕΔΙΟ	4	2001
3131	ΜΑΘΗΜΑΤ ΑΝΑΛΥΣΗ (ΣΥΝΑΡΤΗΣΕΙΣ ΜΙΑΣ ΜΕΤΑΒΛΗΤΗΣ)	6	2001
3020	ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΗΛΕΚΤΡ. ΥΠΟΛΟΓΙΣΤΩΝ	5	2001
3049	ΦΥΣΙΚΗ(ΜΗΧΑΝΙΚΗ)	5	2001

At the bottom right of the page, there is a button labeled 'Κλείσιμο'.

Εικόνα 7.3.στ

### 7.3.1 Μαθήματα κορμού:

Επιλέγοντας το link [Μαθήματα κορμού](#) από οποιοδήποτε σημείο της αρχικής (μετά το επιτυχές login σελίδας), ο χρήστης περνάει στη σελίδα *Kormos\_ola.jsp*, η οποία και περιέχει την πλήρη περιγραφή όλων των μαθημάτων κορμού του προγράμματος σπουδών του Τμήματος, διατεταγμένα με τη σειρά ανά εξάμηνο και αλφαβητικά. Παρακάτω φαίνεται ένα μέρος αυτής της σελίδας:

Σύστημα Ηλεκτρονικών Εγγραφών

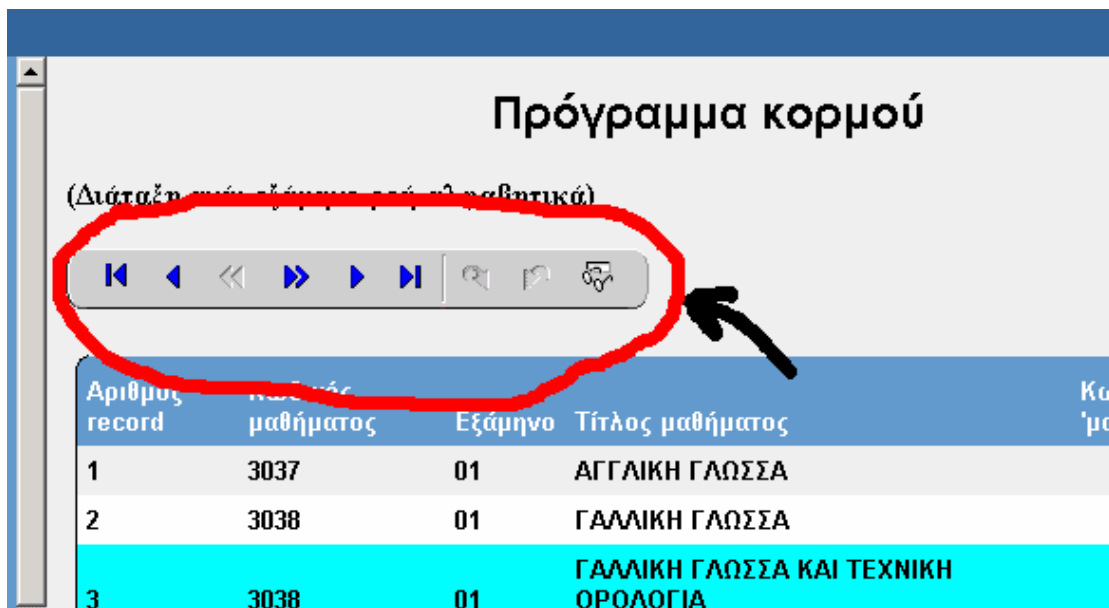
Πρόγραμμα κορμού

(Διάταξη ανά: εξάμηνο-ροή-αλφαβητικά)

Αριθμός record	Κωδικός μαθήματος	Εξάμηνο	Τίτλος μαθήματος	Κωδικός ομάδας 'μαθήματος'
1	3037	01	ΑΓΓΛΙΚΗ ΓΛΩΣΣΑ	
2	3038	01	ΓΑΛΛΙΚΗ ΓΛΩΣΣΑ	
3	3038	01	ΓΑΛΛΙΚΗ ΓΛΩΣΣΑ ΚΑΙ ΤΕΧΝΙΚΗ ΟΡΟΛΟΓΙΑ	
4	3086	01	ΓΕΡΜΑΝΙΚΗ ΓΛΩΣΣΑ	
5	3132	01	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	
6	3035	01	ΗΛΕΚΤΡΟΛΟΓΙΚΟ ΣΧΕΔΙΟ	
7	3027	01	ΙΣΤΟΡΙΑ ΤΩΝ ΕΠΙΣΤΗΜΟΝΙΚΩΝ ΚΑΙ ΦΙΛΟΣΟΦΙΚΩΝ ΙΔΕΩΝ	2011
8	3277	01	ΙΣΤΟΡΙΚΗ ΜΟΥΣΙΚΟΛΟΓΙΑ ΚΑΙ ΘΕΩΡΗΤΙΚΑ ΜΟΥΣΙΚΗΣ	

Εικόνα 7.3.1.α

Παρατηρούμε στην παραπάνω εικόνα την ύπαρξη μιας μπάρας εργαλείων-κουμπιών,



Εικόνα 7.3.1.β

η οποία και αποτελεί την κύρια βοήθεια πλοήγησης του χρήστη του συστήματος στα records και το κύριο εργαλείο μέσω του οποίου είναι σε θέση να πραγματοποιήσει διάφορες λειτουργίες ανάλογα με το επίπεδο πρόσβασης που διαθέτει.



Εικόνα 7.3.1.γ

Η μπάρα αυτή διαθέτει πλήθος κουμπιών, καθένα από τα οποία έχει και διαφορετική λειτουργία ανάλογα τη σελίδα στην οποία βρισκόμαστε τη δεδομένη στιγμή. Συγκεκριμένα για τη σελίδα «Μαθήματα κορμού» ο χρήστης μπορεί να πλοηγηθεί σε 6 διαφορετικές κατευθύνσεις (6 βελάκια) και να δει είτε την μεμονωμένη απεικόνιση μιας καρτέλας μαθήματος (Εικόνα 7.3.1.δ), είτε τον πίνακα πλοήγησης όλων των επιλεγμένων μαθημάτων (Εικόνα 7.3.1.ε).

Κωδικός μαθήματος	3038
Εξάμηνο	01
Τίτλος μαθήματος	ΓΑΛΛΙΚΗ ΓΛΩΣΣΑ ΚΑΙ ΤΕΧΝΙΚΗ ΟΡΟΛΟΓΙΑ
Κωδικός ομάδας μαθήματος	

Εικόνα 7.3.1.δ

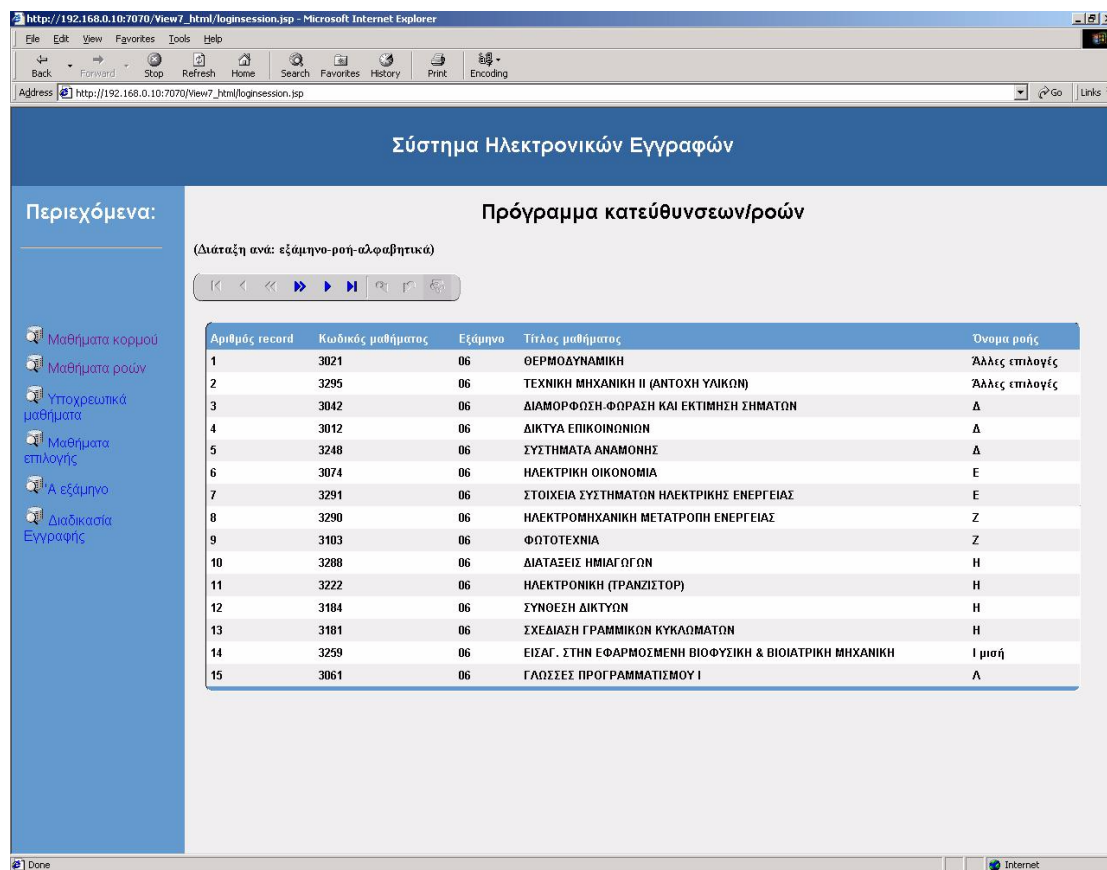
Αριθμός record	Κωδικός μαθήματος	Εξάμηνο	Τίτλος μαθήματος	Κωδικός ομάδας 'μαθήματος'
1	3037	01	ΑΓΓΛΙΚΗ ΓΛΩΣΣΑ	
2	3038	01	ΓΑΛΛΙΚΗ ΓΛΩΣΣΑ	
3	3038	01	ΓΑΛΛΙΚΗ ΓΛΩΣΣΑ ΚΑΙ ΤΕΧΝΙΚΗ ΟΡΟΛΟΓΙΑ	
4	3086	01	ΓΕΡΜΑΝΙΚΗ ΓΛΩΣΣΑ	
5	3132	01	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	
6	3035	01	ΗΛΕΚΤΡΟΛΟΓΙΚΟ ΣΧΕΔΙΟ	
7	3027	01	ΙΣΤΟΡΙΑ ΤΩΝ ΕΠΙΣΤΗΜΟΝΙΚΩΝ ΚΑΙ ΦΙΛΟΣΟΦΙΚΩΝ ΙΔΕΩΝ	2011
8	3277	01	ΙΣΤΟΡΙΚΗ ΜΟΥΣΙΚΟΛΟΓΙΑ ΚΑΙ ΘΕΩΡΗΤΙΚΑ ΜΟΥΣΙΚΗΣ	
9	3087	01	ΙΤΑΛΙΚΗ ΓΛΩΣΣΑ	
10	3131	01	ΜΑΘΗΜΑΤ ΑΝΑΛΥΣΗ (ΣΥΝΑΡΤΗΣΕΙΣ ΜΙΑΣ ΜΕΤΑΒΛΗΤΗΣ)	
11	3020	01	ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΗΛΕΚΤΡ. ΥΠΟΛΟΓΙΣΤΩΝ	
12	3028	01	ΣΤΟΙΧΕΙΑ ΚΟΙΝΩΝΙΟΛΟΓΙΑΣ	2011
13	3048	01	ΦΙΛΟΣΟΦΙΑ	2011
14	3049	01	ΦΥΣΙΚΗ(ΜΗΧΑΝΙΚΗ)	
15	3163	01	ΧΗΜΕΙΑ	

Εικόνα 7.3.1.ε

### 7.3.2 Μαθήματα ροών:

Επιλέγοντας το link [Μαθήματα ροών](#) από οποιοδήποτε σημείο της αρχικής, ο χρήστης περνάει στη σελίδα *Roes\_ola\_titles.jsp*, η οποία και περιέχει την πλήρη περιγραφή όλων των μαθημάτων ροών του προγράμματος σπουδών του Τμήματος, διατεταγμένα με τη σειρά ανά εξάμηνο, ροή στην οποία ανήκουν και αλφαβητικά. Παρακάτω φαίνεται ένα μέρος αυτής της σελίδας:



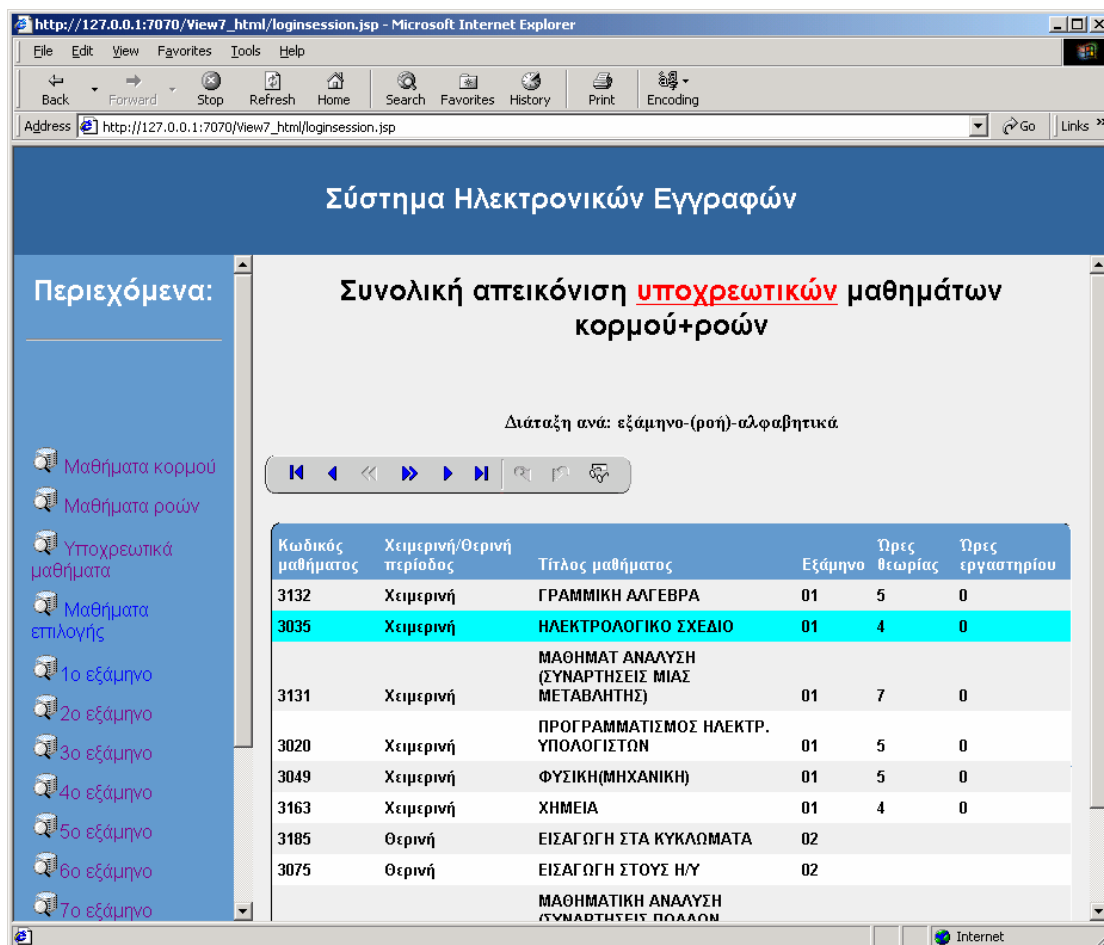


Εικόνα 7.3.2.α

Όπως και προηγουμένως, η φιλοσοφία πλοήγησης είναι η ίδια, με την toolbar να οδηγεί στα αντίστοιχα links και διέπει όλες τις παρεμφερείς σελίδες πλοήγησης στα μαθήματα, όπως θα δούμε και παρακάτω.

### 7.3.3 Υποχρεωτικά μαθήματα:

Επιλέγοντας το link [Υποχρεωτικά μαθήματα](#) από οποιοδήποτε σημείο της αρχικής, ο χρήστης περνάει στη σελίδα *Yproxreotika.jspr*, η οποία και περιέχει την πλήρη περιγραφή όλων των υποχρεωτικών μαθημάτων του προγράμματος σπουδών του Τμήματος, ανεξάρτητα αν αυτά ανήκουν στον κορμό ή στις ροές και διατεταγμένα με τη σειρά ανά εξάμηνο, τυχούσα ροή στην οποία ανήκουν (αν ανήκουν σε ροή) και αλφαβητικά. Παρακάτω φαίνεται ένα μέρος αυτής της σελίδας:



Εικόνα 7.3.3.α

#### 7.3.4 Μαθήματα επιλογής:

Επιλέγοντας το link [Μαθήματα επιλογής](#), ο χρήστης περνάει στη σελίδα *Epilogis.jsp*, η οποία και περιέχει την πλήρη περιγραφή όλων των μαθημάτων επιλογής του προγράμματος σπουδών του Τμήματος, ανεξάρτητα αν αυτά ανήκουν στον κορμό ή στις ροές και διατεταγμένα με τη σειρά ανά εξάμηνο, τυχούσα ροή στην οποία ανήκουν (αν ανήκουν σε ροή) και αλφαβητικά. Παρακάτω φαίνεται ένα μέρος αυτής της σελίδας:

Σύστημα Ηλεκτρονικών Εγγραφών

Μαθήματα **επιλογής**, ανεξαρτήτως εξαμήνου

Διάταξη ανά: εξάμηνο-(ροή)-αλφαβητικά

Κωδικός μαθήματος	Χειμερινή/Θερινή περίοδος	Τίτλος μαθήματος	Εξάμηνο	Ώρες θεωρίας	Ώρες εργαστηρίου
3027	Χειμερινή	ΙΣΤΟΡΙΑ ΤΩΝ ΕΠΙΣΤΗΜΟΝΙΚΩΝ ΚΑΙ ΦΙΛΟΣΟΦΙΚΩΝ ΙΔΕΩΝ	01	2	0
3028	Χειμερινή	ΣΤΟΙΧΕΙΑ ΚΟΙΝΩΝΙΟΛΟΓΙΑΣ	01	2	0
3048	Χειμερινή	ΦΙΛΟΣΟΦΙΑ	01	2	0
3243	Θερινή	ΑΞΙΟΠΙΣΤΙΑ ΚΑΙ ΕΛΕΓΧΟΣ ΠΟΙΟΤΗΤΑΣ ΣΥΣΤΗΜΑΤΩΝ	04	4	0
3209	Θερινή	ΔΙΑΚΡΙΤΕΣ ΜΕΘΟΔΟΙ ΓΙΑ ΤΗΝ ΕΠΙΣΤΗΜΗ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ	04	5	0
3115	Θερινή	ΕΡΓΑΣΤΗΡΙΑΚΗ ΦΥΣΙΚΗ	04	4	0
3018	Θερινή	ΗΛΕΚΤΡΙΚΕΣ ΜΕΤΡΗΣΕΙΣ (ΗΛΕΚΤΡΟΝ-ΨΗΦΙΑΚΕΣ)	04	4	0
3283	Θερινή	ΤΕΧΝΙΚΗ ΜΗΧΑΝΙΚΗ Ι	04	6	0
3161	Θερινή	ΤΕΧΝΟΛΟΓΙΚΗ ΟΙΚΟΝΟΜΙΚΗ	04	4	0
3050	Θερινή	ΦΥΣΙΚΗ/ΔΟΜΗ ΤΗΣ ΓΛΩΣΣΗΣ	04	5	0

Εικόνα 7.3.4.α

### 7.3.5 Links εξαμήνων:

Ακολουθούν τα links των επιμέρους εξαμήνων, τα οποία και ανοίγουν σε ξεχωριστό παράθυρο (ένα σε κάθε περίπτωση, στο οποίο μπορούν να εναλλαχθούν εξάμηνα) τις καρτέλες των μαθημάτων του εξαμήνου, ανάλογα την κατηγοριοποίησή τους σε αυτό.

Προκειμένου να γίνουν κατανοητά τα παραπάνω παρατίθενται ακολούθως όλες οι καρτέλες του Α' εξαμήνου:

http://127.0.0.1:7070/View7\_html/Eksamino\_A\_JSTab.jsp - Microsoft Internet Explorer

Α εξάμηνο: Υποχρεωτικά μαθήματα Κατ' επιλογήν υποχρεωτικά μαθήματα Προαιρετικά Α' εξαμήνου Όλα μαζί

Κείμενο βοήθειας: Βρίσκεστε στο Α' εξάμηνο!

### Υποχρεωτικά μαθήματα Α' εξαμήνου

◀ ◁ ▷ ▶ ▶▶ 🔍 🖨️ 🗑️

Κωδικός μαθήματος	Τίτλος μαθήματος	Ώρες θεωρίας	Ακαδημαϊκό έτος
3132	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	5	2001
3035	ΗΛΕΚΤΡΟΛΟΓΙΚΟ ΣΧΕΔΙΟ	4	2001
3131	ΜΑΘΗΜΑΤ ΑΝΑΛΥΣΗ (ΣΥΝΑΡΤΗΣΕΙΣ ΜΙΑΣ ΜΕΤΑΒΛΗΤΗΣ)	6	2001
3020	ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΗΛΕΚΤΡ. ΥΠΟΛΟΓΙΣΤΩΝ	5	2001
3049	ΦΥΣΙΚΗ(ΜΗΧΑΝΙΚΗ)	5	2001

Κλείσιμο

Εικόνα 7.3.5.α

http://127.0.0.1:7070/View7\_html/Eksamino\_A\_JSTab.jsp - Microsoft Internet Explorer

Α εξάμηνο: Υποχρεωτικά μαθήματα Κατ' επιλογήν υποχρεωτικά μαθήματα Προαιρετικά Α' εξαμήνου Όλα μαζί

Κείμενο βοήθειας: Βρίσκεστε στο Α' εξάμηνο!

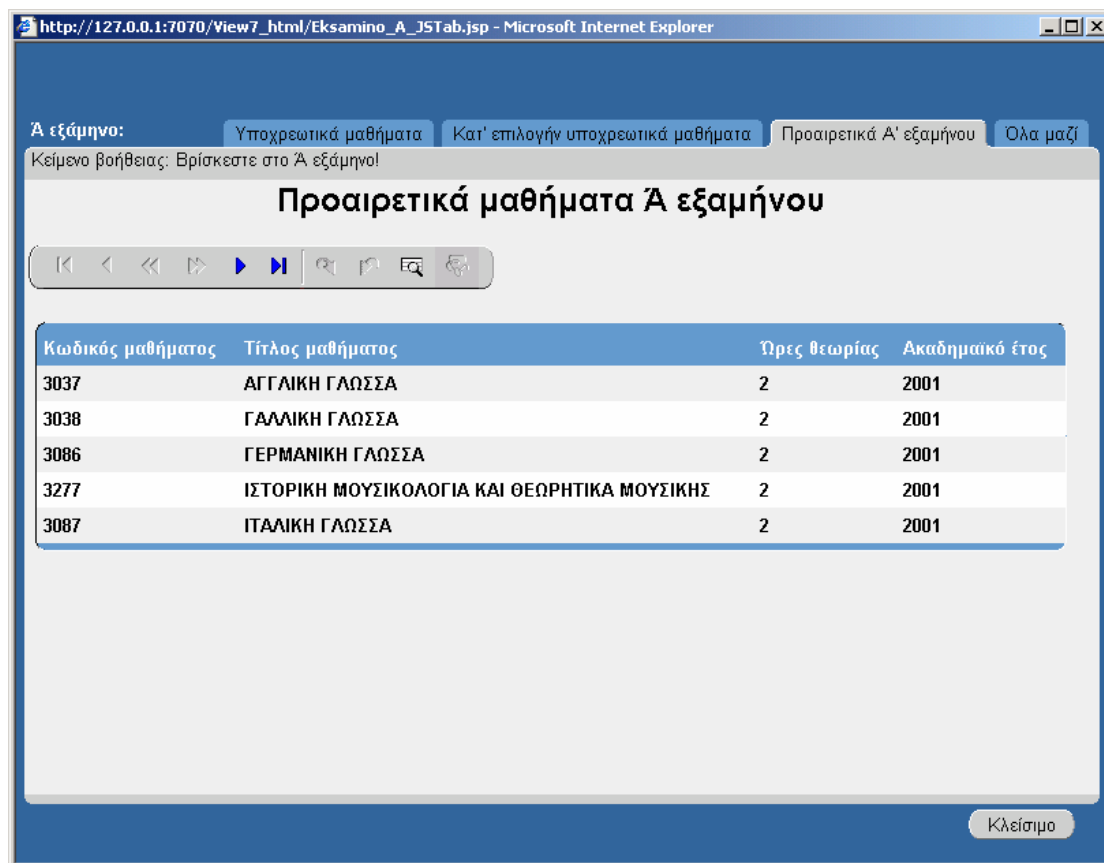
### Κατ' επιλογήν υποχρεωτικά μαθήματα Α' εξαμήνου

◀ ◁ ▷ ▶ ▶▶ 🔍 🖨️ 🗑️

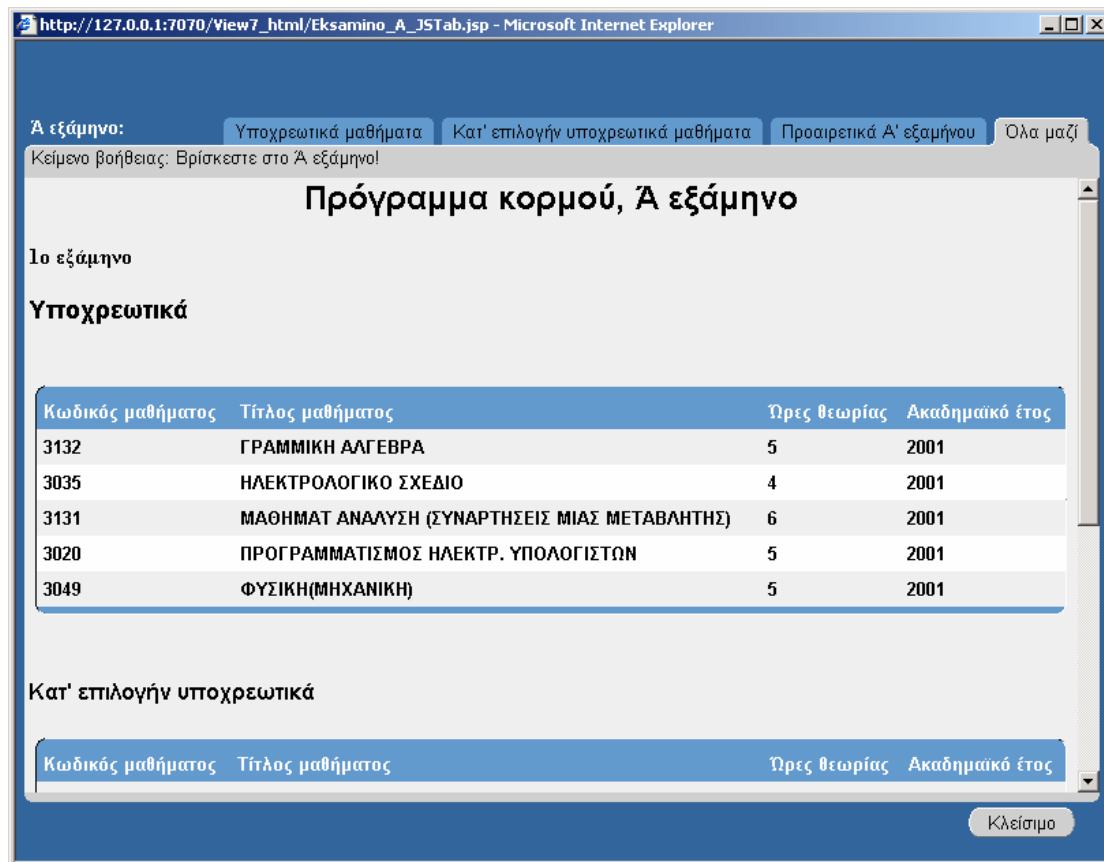
Κωδικός μαθήματος	Τίτλος μαθήματος	Ώρες θεωρίας	Ακαδημαϊκό έτος
3027	ΙΣΤΟΡΙΑ ΤΩΝ ΕΠΙΣΤΗΜΟΝΙΚΩΝ ΚΑΙ ΦΙΛΟΣΟΦΙΚΩΝ ΙΔΕΩΝ	2	2001
3028	ΣΤΟΙΧΕΙΑ ΚΟΙΝΩΝΙΟΛΟΓΙΑΣ	2	2001
3048	ΦΙΛΟΣΟΦΙΑ	2	2001

Κλείσιμο

Εικόνα 7.3.5.β



Εικόνα 7.3.5.γ



Εικόνα 7.3.5.δ

### 7.3.6 Διαδικασία εγγραφής:

Τελευταία δυνατότητα και ίσως η πιο σημαντική που δίνεται στους φοιτητές-χρήστες του συστήματος είναι η εγγραφή τους, είτε μόνο σε μαθήματα εξαμήνου («μονού» ή «ζυγού»), είτε σε κατεύθυνση και συνακόλουθα επιτρεπτού συνδυασμού ροών και μαθημάτων του έκτου εξαμήνου σπουδών του Τμήματος. Προκειμένου να γίνει πλήρως κατανοητή η παραπάνω διαδικασία παρατίθενται αναλυτικά όλα τα δυνατά βήματα κατά την εγγραφή ενός φοιτητή, αρχικά στην περίπτωση επιλογής κατεύθυνσης/ροών (Εικόνες 7.6.α – 7.6. ) και στη συνέχεια μόνο σε μαθήματα κάποιου οποιουδήποτε εξαμήνου (Εικόνες 7.6. – 7.6. ).

Στην αρχική σελίδα στην οποία και μεταφέρεται ο χρήστης μετά το επιτυχές login θα πρέπει να επιλεγεί το link [Διαδικασία Εγγραφής](#), οπότε και ανοίγει η ακόλουθη εισαγωγική σελίδα:

http://127.0.0.1:7070/View7\_html/loginsession.jsp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Print Encoding

Address http://127.0.0.1:7070/View7\_html/loginsession.jsp

## Σύστημα Ηλεκτρονικών Εγγραφών

- Μαθήματα κορμού
- Μαθήματα ροών
- Υποχρεωτικά μαθήματα
- Μαθήματα επιλογής
- 1ο εξάμηνο
- 2ο εξάμηνο
- 3ο εξάμηνο
- 4ο εξάμηνο
- 5ο εξάμηνο
- 6ο εξάμηνο
- 7ο εξάμηνο
- 8ο εξάμηνο
- 9ο εξάμηνο
- Διαδικασία Εγγραφής

### Γενικές απαιτήσεις

Προκειμένου να επιλέξετε κατεύθυνση και ροές θα πρέπει να εκπληρούνται οι προϋποθέσεις που περιγράφονται αναλυτικά παρακάτω, αλλά και στον Οδηγό Σπουδών του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών.

Παρακαλείστε για την σωστή χρήση τους.

Σε περίπτωση που θέλετε να επιλέξετε κατεύθυνση σπουδών και ροές μαθημάτων, τσεκάρετε την πρώτη επιλογή.

Σε αντίθετη περίπτωση που επιθυμείτε μόνο την δήλωση μαθημάτων, τσεκάρετε την κατάλληλη επιλογή, ανάλογα του εάν θέλετε να εγγραφείτε σε χειμερινό ή θερινό εξάμηνο αντίστοιχα.

Επιλογή	check:
1. Κατευθύνσεις	<input type="checkbox"/>
2. Μαθήματα χειμερινών εξαμήνων	<input type="checkbox"/>
3. Μαθήματα θερινών εξαμήνων	<input type="checkbox"/>

Επιλογή

Εικόνα 7.6.α

Εδώ καλείται ο χρήστης να επιλέξει σε ποια από τις 3 διαδικασίες εγγραφής θέλει να εμπλακεί.

### 7.3.6.1 Διαδικασία εγγραφής – Κατευθύνσεις:

Ας υποθέσουμε ότι ο χρήστης επιλέγει να εγγραφεί σε κατεύθυνση σπουδών, οπότε συνακόλουθα υποχρεούται να επιλέξει κάποιον από τους δυνατούς συνδυασμούς ροών που περιέχει η κατεύθυνση αυτή, ενώ ως επιπλέον δυνατότητα του συστήματος παρέχεται και η δυνατότητα να γραφτεί σε μαθήματα του έκτου εξαμήνου σπουδών, καθόσον τότε πραγματοποιείται η επιλογή κατεύθυνσης από τους φοιτητές σύμφωνα με τις υπάρχουσες οδηγίες του Οδηγού Σπουδών του Τμήματος.

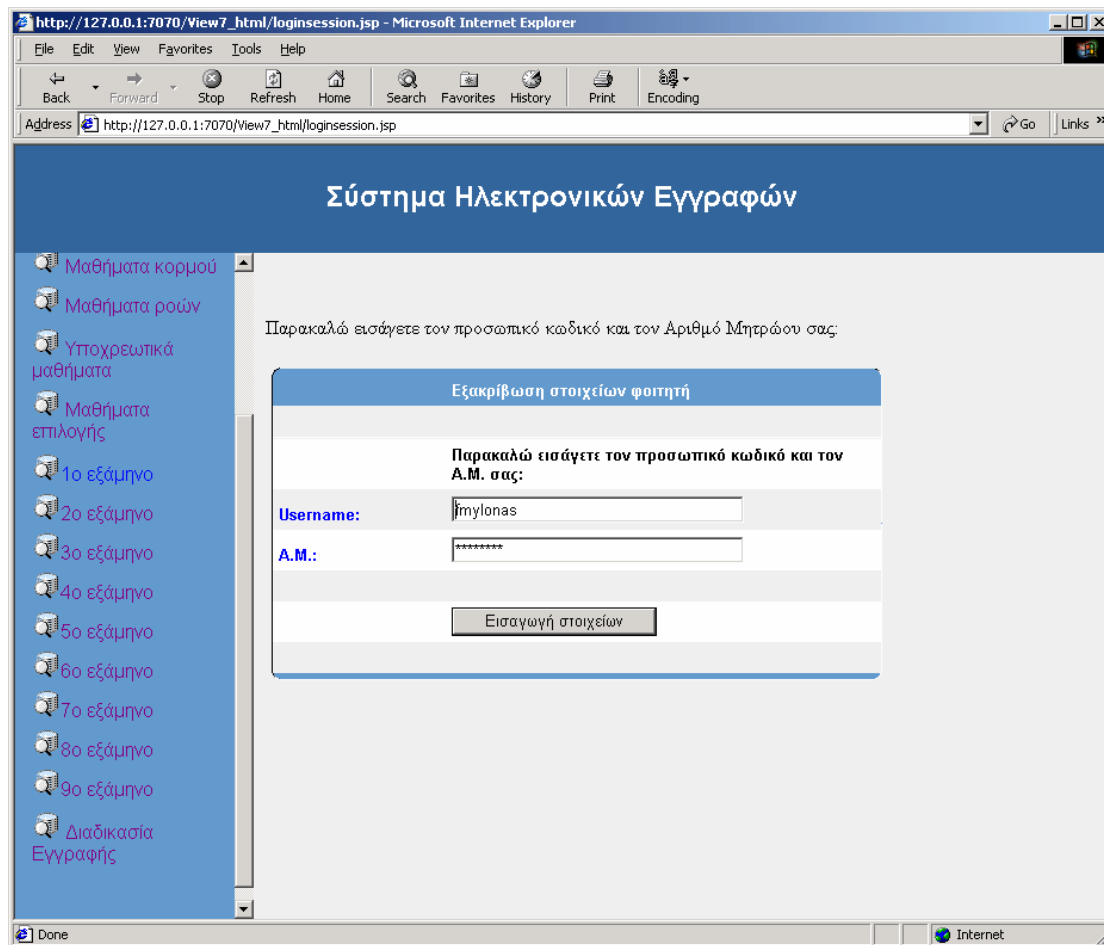
Τσεκάροντας, λοιπόν, την επιλογή **Κατευθύνσεις** στην παραπάνω αρχική σελίδα και υποβάλλοντας την (Εικόνα 7.3.6.1.α)

Επιλογή	check:
1. Κατευθύνσεις	<input checked="" type="checkbox"/>
2. Μαθήματα χειμερινών εξαμήνων	<input type="checkbox"/>
3. Μαθήματα θερινών εξαμήνων	<input type="checkbox"/>

Επιλογή

Εικόνα 7.3.6.1.α

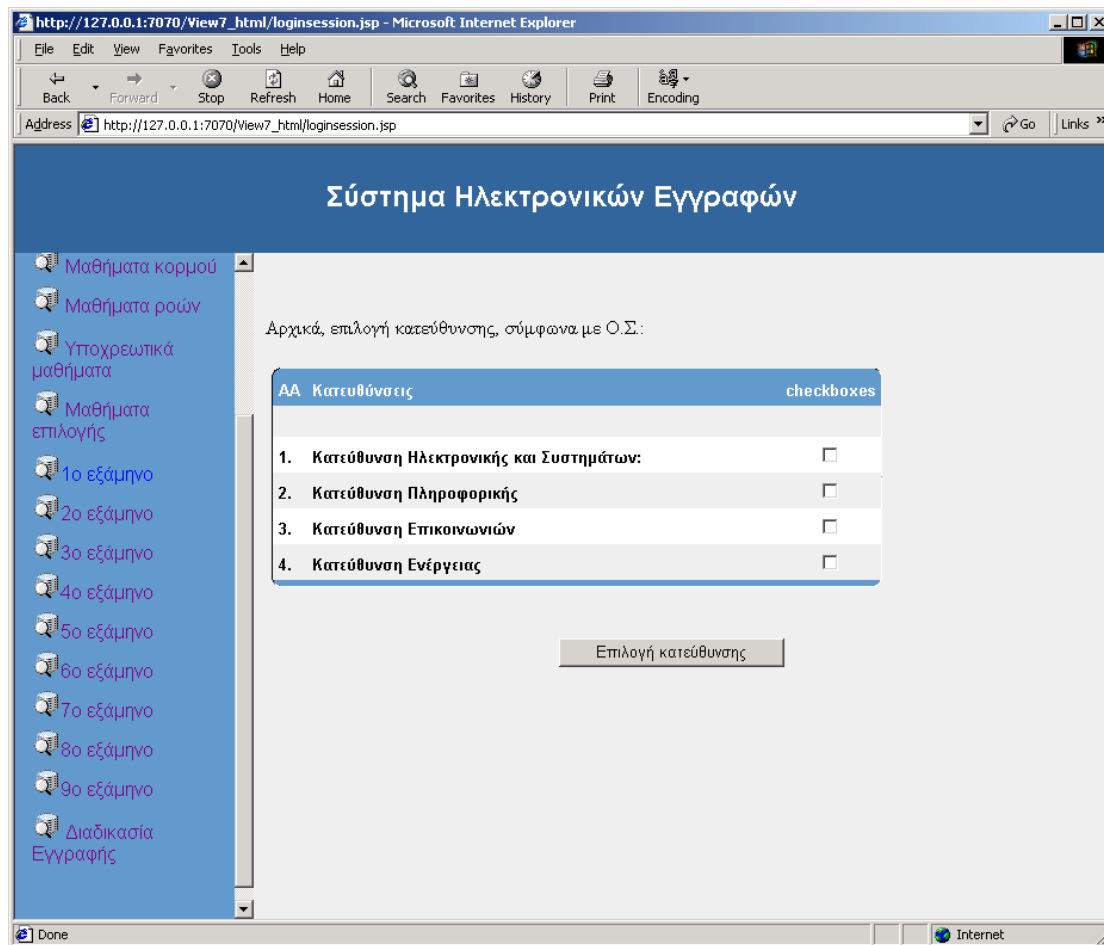
, ο φοιτητής μεταφέρεται στην επόμενη σελίδα εξακρίβωσης των στοιχείων του, όσον αφορά το αν αυτός ανήκει στο έκτο εξάμηνο σπουδών ή όχι, για λόγους ελάττωσης του φορτίου στον κόμβο από το σημείο αυτό και πέρα. Τα στοιχεία του φοιτητή είναι προσυμπληρωμένα από το σύστημα, καθόσον αυτός έχει ήδη κάνει login σε αυτό και το σύστημα γνωρίζει ποιος είναι καθόλη τη διάρκεια που αυτός πλοηγείται στο web site (μέσω της διατήρησης του id του στο http session – τεχνολογία http session tracking):



Εικόνα 7.3.6.1.β

Υποβάλλοντας τα στοιχεία του και εφόσον αυτά είναι σωστά ανοίγει η σελίδα επιλογής κατεύθυνσης. Να σημειώσουμε εδώ ότι όλη η φιλοσοφία στα βήματα επιλογής κατεύθυνσης/ρουν του ΣΗΕ ακολουθεί πιστά τη φιλοσοφία της μέχρι σήμερα υπάρχουσας έντυπης αίτησης εγγραφής με τα αντίστοιχα βήματα που αυτή προτείνει. Έτσι, θα έχουμε:





Εικόνα 7.3.6.1.γ

Ανάλογα τώρα του ποια θα είναι η επιλογή του φοιτητή στα παραπάνω checkboxes, του παρέχονται στην ίδια σελίδα δυναμικά οι επιτρεπόμενοι δυνατοί συνδυασμοί ρούων της κάθε κατεύθυνσης. Στο παράδειγμα μας, ο φοιτητής επέλεξε την κατεύθυνση Πληροφορικής:

Αρχικά, επιλογή κατεύθυνσης, σύμφωνα με Ο.Σ.:

ΑΑ	Κατευθύνσεις	checkboxes
1.	Κατεύθυνση Ηλεκτρονικής και Συστημάτων:	<input type="checkbox"/>
2.	Κατεύθυνση Πληροφορικής	<input type="checkbox"/>
3.	Κατεύθυνση Επικοινωνιών	<input type="checkbox"/>
4.	Κατεύθυνση Ενέργειας	<input type="checkbox"/>

Επιλογή κατεύθυνσης

Συνδυασμοί Κατεύθυνσης Πληροφορικής	Υ	Λ	Η	Δ	Τ	Σ	Ζ	Ε	Ο	Ι	Μ	Φ
1. <input type="checkbox"/>	✓	✓										
2. <input type="checkbox"/>	✓	½	✓									
3. <input type="checkbox"/>	✓	½		✓								
4. <input type="checkbox"/>	✓	½				✓						
5. <input type="checkbox"/>	½	✓		✓								
6. <input type="checkbox"/>	½	✓				✓						

Επιλογή συνδυασμού ροών

Εικόνα 7.3.6.1.δ

Παρατηρούμε ότι πριν επιλέξει συνδυασμό ο φοιτητής δεν μπορεί να υποβάλει τη φόρμα, καθότι το κουμπί υποβολής παραμένει ανενεργό, όπως άλλωστε και τα προηγούμενα checkboxes και κουμπιά της επιλογής κατεύθυνσης. Επιλέγοντας, λοιπόν, συνδυασμό ροών, αυτόματα μεταφέρεται στη σελίδα εισαγωγής στοιχείων στη βάση, όπου και καλείται, πέρα από τα προσυμπληρωμένα, με βάση τις προηγούμενες επιλογές του, ή με βάση τα χαρακτηριστικά του login του στο σύστημα, στοιχεία, να δώσει κάποια επιπλέον, ώστε να καταστεί δυνατή η ολοκληρωμένη αρχειοθέτηση στη βάση του συστήματος της εγγραφής του.

Στο παράδειγμά μας επιλέγεται ο συνδυασμός 2 της κατεύθυνσης Πληροφορικής, που περιλαμβάνει υποχρεωτικά δύο ολόκληρες και μία μισή ροή ενώ απαιτείται ο εμπλουτισμός της με μισή ακόμα ροή.

Κωδικός συστήματος: \*\*\*\*\*

Επιλεγμένη κατεύθυνση: Κατεύθυνση Πληροφορικής

Επιλεγμένος συνδυασμός ρωών: Συνδυασμός ρωών: 2

ροή Υ: Ολόκληρη

ροή Α: Μισή

ροή Η: Ολόκληρη

ροή Δ:

ροή Τ:

ροή Σ:

ροή Ζ:

ροή Ε:

Ημερομηνία εγγραφής:

Ημερομηνία υποβολής αίτησης: 2001-09-29 23:27:42.0

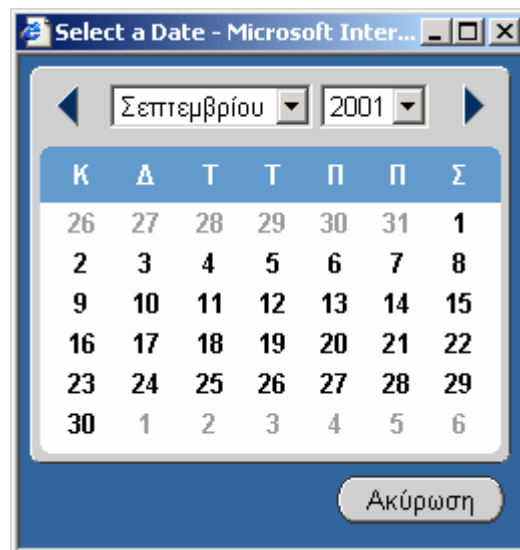
Α.Μ. φοιτητή: 03096150

Ονομαστικό εξάμηνο εγγραφής: Κανονικό εξ. εγγραφής: 6

Υποβολή εγγραφής Reset φόρμας εγγραφής

Εικόνα 7.3.6.1.ε

Το πεδίο της *ημερομηνίας εγγραφής* είναι read-only και πρέπει να συμπληρωθεί μέσω του κουμπιού που υπάρχει στα δεξιά του και ανοίγει ένα παραθυράκι ημερολογίου, το οποίο και επιτρέπει την εισαγωγή ημερομηνιών στο κατάλληλο format που απαιτεί η βάση δεδομένων του συστήματος.

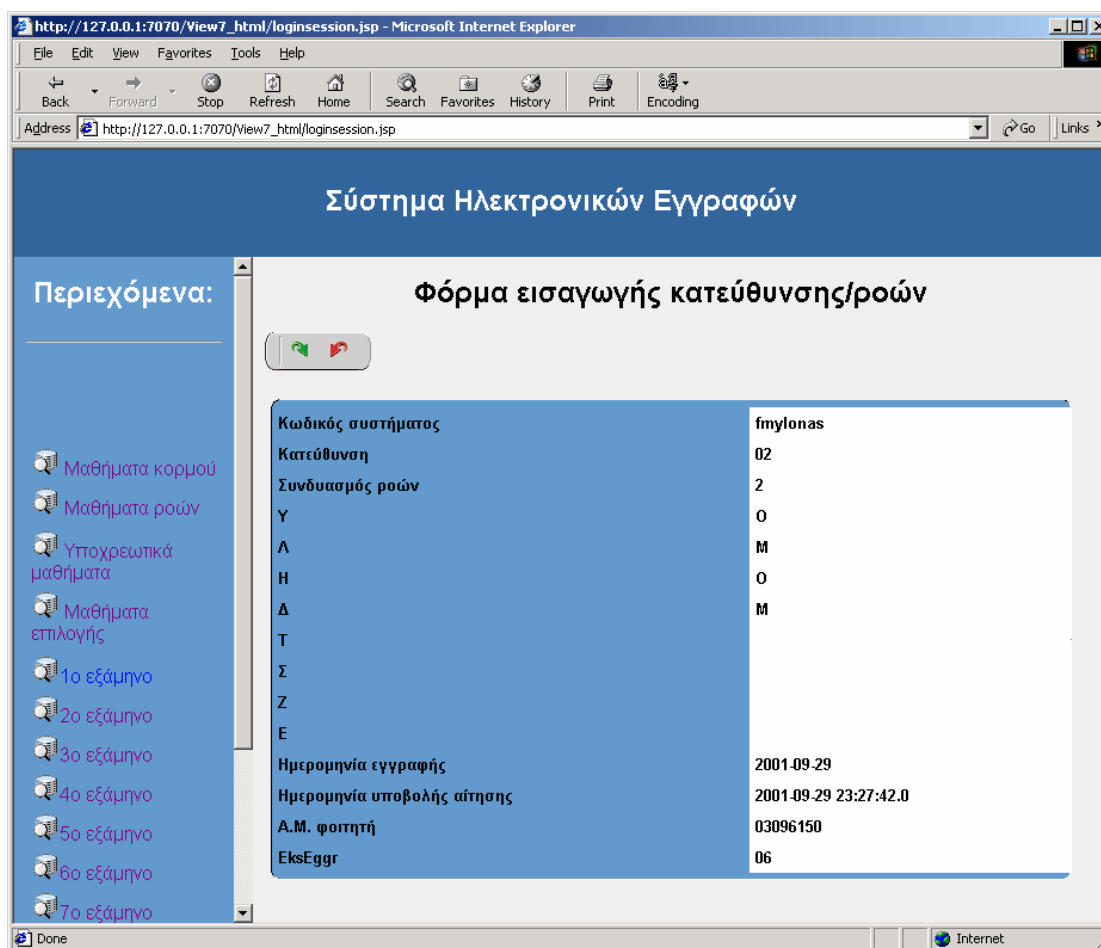


Εικόνα 7.3.6.1.στ

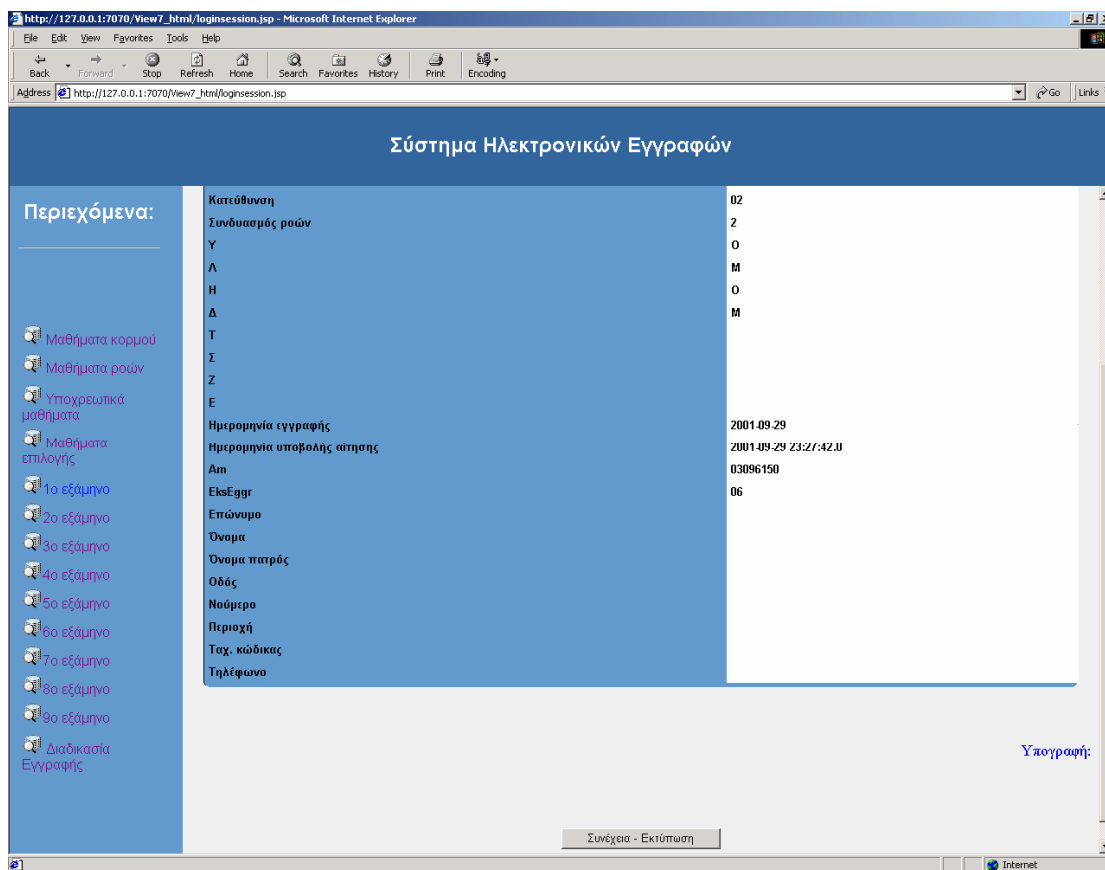
Να σημειώσουμε εδώ ότι υπάρχει δυνατότητα καθαρισμού των μεταβαλλόμενο πεδίων που συμπλήρωσε ο φοιτητής πριν από την υποβολή της φόρμας, σε περίπτωση εύρεσης κάποιου λάθους, μέσω του κουμπιού **Reset φόρμας εγγραφής**.

Μετά την ορθή υποβολή των παραπάνω ο φοιτητής καλείται να ρίξει μια τελευταία ματιά στις επιλογές του στην ακόλουθη σελίδα και να αποφασίσει οριστικά και

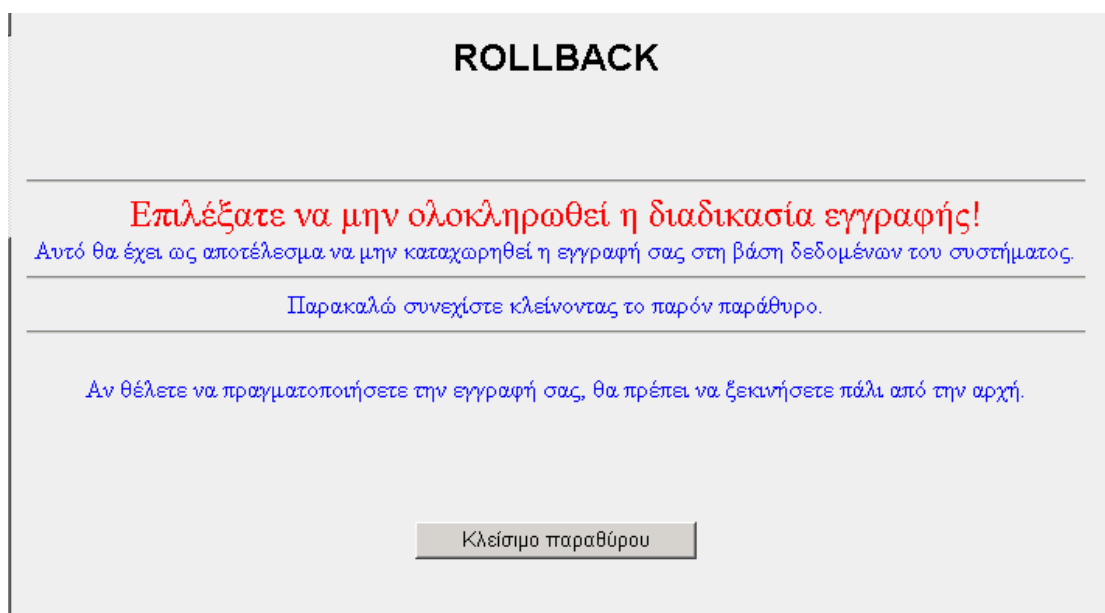
αμετάκλητα αν θα τις καταχωρήσει στη βάση του συστήματος ή όχι. Αν μεν, επιλέξει να τις καταχωρήσει μέσω του **πράσινου** βέλους-κουμπιού, δεν έχει καμία δυνατότητα διόρθωσής τους με ηλεκτρονικό τρόπο, αφού το σύστημα θεωρεί ότι έχει εγγραφεί επιτυχώς σε αυτό και συνεπώς δεν τον αφήνει να προσπελάσει ξανά την εγγραφή του ή ακόμα χειρότερα να επαναεγγραφεί, και καλείται να την εκτυπώσει υποχρεωτικά και να την υπογράψει, προκειμένου να προχωρήσει στο σύστημα (Εικόνα 7.3.6.1.η). Αν δε, για οποιοδήποτε λόγο, επιλέξει το **κόκκινο** βέλος-κουμπί της ακύρωσης της εγγραφής στη βάση (Εικόνα 7.3.6.1.θ), τότε θα πρέπει να ξεκινήσει όλη τη διαδικασία εγγραφής του από την αρχική σελίδα επιλογής (Εικόνα 7.3.6.1.α), ούτως ώστε το σύστημα να επανέλθει σε ετοιμότητα εισαγωγής στοιχείων από τον ίδιο χρήστη, καθαρίζοντας όλες τις cache του και διατηρώντας την ταυτότητά του. Με κατάλληλο prompt (Εικόνα 7.3.6.1.ι) ερωτάται ο χρήστης αν θέλει να πραγματοποιήσει κάτι τέτοιο ή αν απλώς θέλει να περιηγηθεί στο υπόλοιπο σύστημα με την επιλογή της διαδικασίας εγγραφής απενεργοποιημένη. (Εικόνα 7.3.6.1.κ).



Εικόνα 7.3.6.1.ζ



Εικόνα 7.3.6.1.η



Εικόνα 7.3.6.1.θ

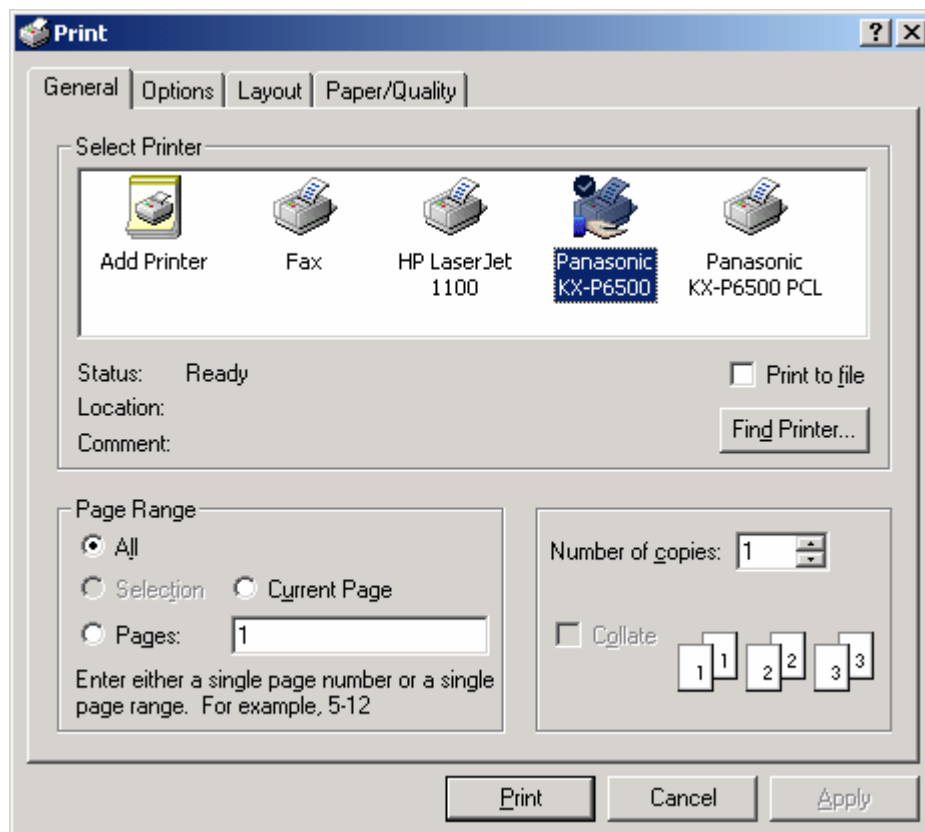


Εικόνα 7.3.6.1.ι

**Η επιλογή "Διαδικασία εγγραφής" απενεργοποιήθηκε.**

Εικόνα 7.3.6.1.κ

Μετά το ακόλουθο παράθυρο εκτύπωσης, ακολουθεί αυτόματο redirect του χρήστη στη σελίδα εισαγωγής μαθημάτων του έκτου εξαμήνου σπουδών (μέρος της οποίας φαίνεται στην Εικόνα 7.3.6.1.μ), καθώς σε αυτό θεωρείται από το σύστημα ότι θέλει να εγγραφεί, εφόσον έχει ακολουθήσει διαδικασία επιλογής κατεύθυνσης/ροών.



Εικόνα 7.3.6.1.λ

http://127.0.0.1:7070/View7\_html/loginsession.jsp - Microsoft Internet Explorer

Address http://127.0.0.1:7070/View7\_html/loginsession.jsp

## Σύστημα Ηλεκτρονικών Εγγραφών

**Περιεχόμενα:**

- Μαθήματα κορμού
- Μαθήματα ρούιν
- Υποχρεωτικά μαθήματα
- Μαθήματα επιλογής
- 1ο εξάμηνο
- 2ο εξάμηνο
- 3ο εξάμηνο
- 4ο εξάμηνο
- 5ο εξάμηνο
- 6ο εξάμηνο
- 7ο εξάμηνο
- 8ο εξάμηνο
- 9ο εξάμηνο
- Διαδικασία Εγγραφής

### Φόρμα εγγραφής ΣΤ' ονομ. εξ. εγγραφής

**2ο εξάμηνο, Υποχρεωτικά**

Αριθμός record	Κωδικός μαθήματος	Τίτλος μαθήματος	Πίρες θεωρίας	Ακαδημαϊκό έτος	Επιλογή μαθήματος	info
1	3185	ΕΙΣΑΓΩΓΗ ΣΤΑ ΚΥΚΛΩΜΑΤΑ		2000	<input type="checkbox"/>	
2	3075	ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ Η/Υ		2000	<input type="checkbox"/>	
3	3137	ΜΑΘΗΜΑΤΙΚΗ ΑΝΑΛΥΣΗ (ΣΥΝΑΡΤΗΣΕΙΣ ΠΟΛΛΩΝ ΜΕΤΑΒΛ.ΔΙΑΝΥΣΜΑΤΙΚΗ ΑΝΑΛΥΣΗ)		2000	<input type="checkbox"/>	
4	3079	ΠΟΛΙΤΙΚΗ ΟΙΚΟΝΟΜΙΑ		2000	<input type="checkbox"/>	
5	3138	ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΕΣ ΤΕΧΝΙΚΕΣ		2000	<input type="checkbox"/>	
6	3017	ΦΥΣΙΚΗ (ΗΛΕΚΤΡΟΜΑΓΝΗΤΙΣΜΟΣ)		2000	<input type="checkbox"/>	





**2ο εξάμηνο, κατ' επιλογήν υποχρεωτικά**

Αριθμός record	Κωδικός μαθήματος	Τίτλος μαθήματος	Πίρες θεωρίας	Ακαδημαϊκό έτος	Επιλογή μαθήματος	info
1	3039	ΑΓΓΛΙΚΗ ΓΛΩΣΣΑ		2000	<input type="checkbox"/>	
2	3088	ΓΑΛΛΙΚΗ ΓΛΩΣΣΑ		2000	<input type="checkbox"/>	
3	3118	ΓΕΡΜΑΝΙΚΗ ΓΛΩΣΣΑ		2000	<input type="checkbox"/>	
4	3005	ΙΤΑΛΙΚΗ ΓΛΩΣΣΑ		2000	<input type="checkbox"/>	

Εικόνα 7.3.6.1.μ

Παρατηρούμε ότι στη σελίδα του έκτου εξαμήνου αναγράφονται προφανώς όλα τα «ζυγά» μαθήματα του Προγράμματος Σπουδών, καθότι ο Οδηγός Σπουδών του Τμήματος δίνει τη δυνατότητα στο φοιτητή να εγγραφεί, υπακούοντας βέβαια σε κάποιους κανόνες, οι οποίοι και υλοποιούνται στο παρασκήνιο με ελέγχους των εισαγομένων στοιχείων, σε όσα μαθήματα της περιόδου αυτός επιθυμεί ανεξάρτητα του ονομαστικού εξαμήνου εγγραφής του. Επίσης, στην περίπτωση των εξαμήνων ρούιν, έχει πραγματοποιηθεί οπτικός διαχωρισμός των μαθημάτων, ανάλογα αν αυτά θεωρούνται από τον Οδηγό Σπουδών ως υποχρεωτικά και στην μισή επιλεγμένη από το φοιτητή ροή και στην ολόκληρη (καφέ χρώμα), ή ως υποχρεωτικά μόνο στην ολόκληρη ροή (πράσινο χρώμα), όπως φαίνεται και στην παρακάτω εικόνα:

#### 6ο εξάμηνο, υποχρεωτικά

Αριθμός record	Κωδικός μαθήματος	Τίτλος μαθήματος	Ώρες θεωρίας	Ακαδημαϊκό έτος	Ομάδα	Pltext	Επιλογή μαθήματος	info
1	3143	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ	4	2000	8001	Υ	<input type="checkbox"/>	
2	3236	ΕΡΓΑΣΤΗΡΙΟ ΛΟΓΙΚΩΝ ΚΥΚΛΩΜΑΤΩΝ	3	2000	8001	Υ	<input type="checkbox"/>	
3	3061	ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι	4	2000	8002	Λ	<input type="checkbox"/>	
4	3165	ΓΡΑΦΙΚΗ ΜΕ ΥΠΟΛΟΓΙΣΤΕΣ	4	2000	8002	Λ	<input type="checkbox"/>	
5	3288	ΔΙΑΤΑΞΕΙΣ ΗΜΙΑΓΩΓΩΝ	3	2000	8003	Η	<input type="checkbox"/>	
6	3222	ΗΛΕΚΤΡΟΝΙΚΗ (ΤΡΑΝΖΙΣΤΟΡ)	3	2000	8003	Η	<input type="checkbox"/>	
7	3184	ΣΥΝΘΕΣΗ ΔΙΚΤΥΩΝ	3	2000	8003	Η	<input type="checkbox"/>	
8	3181	ΣΧΕΔΙΑΣΗ ΓΡΑΜΜΙΚΩΝ ΚΥΚΛΩΜΑΤΩΝ	3	2000	8003	Η	<input type="checkbox"/>	
9	3042	ΔΙΑΜΟΡΦΩΣΗ-ΦΩΡΑΣΗ ΚΑΙ ΕΚΤΙΜΗΣΗ ΣΗΜΑΤΩΝ	4	2000	8004	Δ	<input type="checkbox"/>	
10	3012	ΔΙΚΤΥΑ ΕΠΙΚΟΙΝΩΝΙΩΝ	4	2000	8004	Δ	<input type="checkbox"/>	
11	3248	ΣΥΣΤΗΜΑΤΑ ΑΝΑΜΟΝΗΣ	3	2000	8004	Δ	<input type="checkbox"/>	
12	3296	ΗΛΕΚΤΡΟΜΑΓΝΗΤΙΚΑ ΠΕΔΙΑ II	4	2000	8005	Τ	<input type="checkbox"/>	
13	3057	ΜΙΚΡΟΚΥΜΑΤΑ	5	2000	8005	Τ	<input type="checkbox"/>	
14	3338	ΣΥΣΤΗΜΑΤΑ ΔΙΑΜΟΡΦΩΣΗΣ ΚΑΙ ΜΕΤΑΔΟΣΗΣ	4	2000	8005	Τ	<input type="checkbox"/>	

Εικόνα 7.3.6.1.ν

Από το σημείο αυτό και πέρα η διαδικασία είναι κοινή με την απλή διαδικασία εγγραφής φοιτητών σε μαθήματα εξαμήνου, η οποία και περιγράφεται αναλυτικά παρακάτω.

#### 7.3.6.2 Διαδικασία εγγραφής – Χειμερινά μαθήματα:

Η επιλογή στην περίπτωση αυτή στην αρχική σελίδα είναι η ακόλουθη:

Επιλογή

check:

1. Κατευθύνσεις

☐

2. Μαθήματα χειμερινών εξαμήνων

☒

3. Μαθήματα θερινών εξαμήνων

☐

Επιλογή

Εικόνα 7.3.6.2.α

Ακολουθεί εδώ μια σελίδα όπου ο χρήστης καλείται να επιλέξει τη βασική πληροφορία του συστήματος, το ονομαστικό εξάμηνο εγγραφής στο οποίο επιθυμεί



να εγγραφεί σε μαθήματα, και το οποίο καθορίζει σειρά παραμέτρων στα μετέπειτα βήματα.

## Επιλογή ονομαστικού εξαμήνου εγγραφής

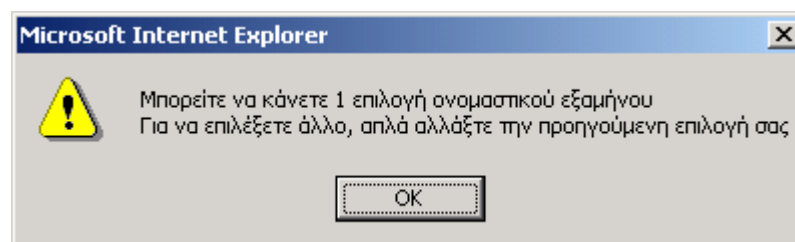
Αρχικά, επιλογή ονομαστικού εξαμήνου σπουδών, σύμφωνα με Ο.Σ.:

ΑΑ	Ονομαστικό εξάμηνο εγγραφής	Επιλογή
1.	1ο. εξάμηνο	<input type="checkbox"/>
2.	3ο. εξάμηνο	<input type="checkbox"/>
3.	5ο. εξάμηνο	<input type="checkbox"/>
4.	7ο. εξάμηνο	<input type="checkbox"/>
3.	9ο. εξάμηνο	<input type="checkbox"/>
4.	Εξάμηνο μεγαλύτερο του 9ου	<input type="checkbox"/>

Επιλογή ονομαστικού εξαμήνου

Εικόνα 7.3.6.2.β

Να σημειώσουμε εδώ ότι όπως και σε κάθε περίπτωση επιλογής checkboxes από το χρήστη, έτσι και τώρα είναι δυνατή η επιλογή μόνο ενός από αυτά, σε αντίθετη περίπτωση προκύπτει κατάλληλο μήνυμα λάθους:



Εικόνα 7.3.6.2.γ

Μετά την επιλογή του ονομαστικού εξαμήνου εγγραφής από το φοιτητή και ανάλογα πάντα του τι επέλεξε, μεταφέρεται στην κατάλληλη σελίδα επιλογής μαθημάτων με τους κατάλληλους ελέγχους να τρέχουν στο παρασκήνιο.

## Φόρμα εγγραφής Α' ονομ. εξ. εγγραφής

### 1ο εξάμηνο, Υποχρεωτικά

Αριθμός record	Κωδικός μαθήματος	Τίτλος μαθήματος	Ώρες θεωρίας	Ακαδημαϊκό έτος	Επιλογή μαθήματος	info1a
1	3132	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	5	2001	<input checked="" type="checkbox"/>	
2	3035	ΗΛΕΚΤΡΟΛΟΓΙΚΟ ΣΧΕΔΙΟ	4	2001	<input checked="" type="checkbox"/>	
3	3131	ΜΑΘΗΜΑΤ ΑΝΑΛΥΣΗ (ΣΥΝΑΡΤΗΣΕΙΣ ΜΙΑΣ ΜΕΤΑΒΛΗΤΗΣ)	6	2001	<input checked="" type="checkbox"/>	
4	3020	ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΗΛΕΚΤΡ. ΥΠΟΛΟΓΙΣΤΩΝ	5	2001	<input checked="" type="checkbox"/>	
5	3049	ΦΥΣΙΚΗ(ΜΗΧΑΝΙΚΗ)	5	2001	<input checked="" type="checkbox"/>	

☐ Επιλέξτε όλα τα υποχρεωτικά μαθήματα του 1ου εξαμήνου

### 1ο εξάμηνο, κατ' επιλογήν υποχρεωτικά

Αριθμός record	Κωδικός μαθήματος	Τίτλος μαθήματος	Ώρες θεωρίας	Ακαδημαϊκό έτος	Επιλογή μαθήματος	info1b
1	3027	ΙΣΤΟΡΙΑ ΤΩΝ ΕΠΙΣΤΗΜΟΝΙΚΩΝ ΚΑΙ ΦΙΛΟΣΟΦΙΚΩΝ ΙΔΕΩΝ	2	2001	<input type="checkbox"/>	
2	3028	ΣΤΟΙΧΕΙΑ ΚΟΙΝΩΝΙΟΛΟΓΙΑΣ	2	2001	<input type="checkbox"/>	
3	3048	ΦΙΛΟΣΟΦΙΑ	2	2001	<input type="checkbox"/>	

Εικόνα 7.3.6.2.δ

Η φόρμα επιλογής περιλαμβάνει όλα τα μονά εξάμηνα, καθώς στο παράδειγμα επιλέξαμε το πρώτο ως ονομαστικό εξάμηνο, έχει **κόκκινα** τα υποχρεωτικά μαθήματα του ονομαστικού,

### 1ο εξάμηνο, Υποχρεωτικά

Αριθμός record	Κωδικός μαθήματος	Τίτλος μαθήματος	Ώρες θεωρίας	Ακαδημαϊκό έτος	Επιλογή μαθήματος	info1a
1	3132	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	5	2001	<input checked="" type="checkbox"/>	
2	3035	ΗΛΕΚΤΡΟΛΟΓΙΚΟ ΣΧΕΔΙΟ	4	2001	<input checked="" type="checkbox"/>	
3	3131	ΜΑΘΗΜΑΤ ΑΝΑΛΥΣΗ (ΣΥΝΑΡΤΗΣΕΙΣ ΜΙΑΣ ΜΕΤΑΒΛΗΤΗΣ)	6	2001	<input checked="" type="checkbox"/>	
4	3020	ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΗΛΕΚΤΡ. ΥΠΟΛΟΓΙΣΤΩΝ	5	2001	<input checked="" type="checkbox"/>	
5	3049	ΦΥΣΙΚΗ(ΜΗΧΑΝΙΚΗ)	5	2001	<input checked="" type="checkbox"/>	

☐ Επιλέξτε όλα τα υποχρεωτικά μαθήματα του 1ου εξαμήνου

Εικόνα 7.3.6.2.ε

παρέχει δυνατότητα συνολικής επιλογής τους με ένα μόνο click,

☐ Επιλέξτε όλα τα υποχρεωτικά μαθήματα του 1ου εξαμήνου

Εικόνα 7.3.6.2.στ

διαθέτει κατάλληλους χρωματισμούς των μαθημάτων ροών, όπως είδαμε και στο προηγούμενο κεφάλαιο,

**7ο εξάμηνο, υποχρεωτικά**

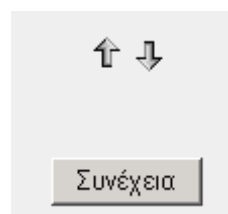
Αριθμός record	Κωδικός μαθήματος	Τίτλος μαθήματος	Ώρες θεωρίας	Ακαδημαϊκό έτος	Κωδικός ροής	Όνομα ροής	Επιλογή μαθήματος	info
1	3136	ΛΕΙΤΟΥΡΓ.ΣΥΣΤ.ΥΠΟΛΟΓ.	4	2001	8001	Υ		
2	3046	ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΩΝ	4	2001	8001	Υ		
3	3297	ΤΕΧΝΟΛΟΓΙΑ ΠΟΛΥΜΕΣΩΝ	3	2001	8001	Υ		
4	3330	ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΚΑΙ ΑΝΑΛΥΣΗ ΕΙΚΟΝΩΝ	4	2001	8001	Υ		
5	3105	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ	4	2001	8002	Λ		
6	3123	ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ	4	2001	8002	Λ		

Εικόνα 7.3.6.2.ζ

βελάκια πλοήγησης κάθετα στη σελίδα, και με μια χρονική καθυστέρηση της τάξεως των 15 δευτερολέπτων για την αποφυγή λάθους και την προετοιμασία του συστήματος για τη δημιουργία των παρακάτω βημάτων, επιτρέπει στο χρήστη την υποβολή των επιλογών του.

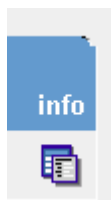


Εικόνα 7.3.6.2.η



Εικόνα 7.3.6.2.θ

Επίσης, δεξιά από κάθε μάθημα παρέχεται η δυνατότητα επισκόπησης της καρτέλας μέσω ενός click στην παρακάτω εικόνα:



Εικόνα 7.3.6.2.ι

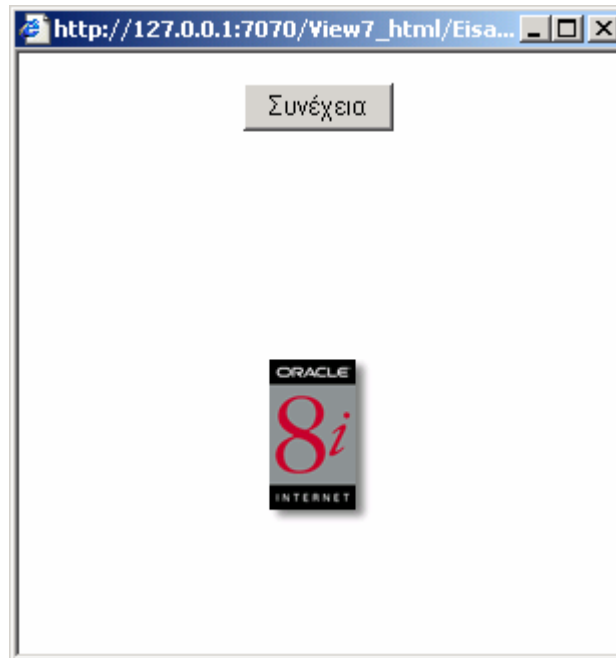
οπότε και παρουσιάζεται αυτή σε νέο παράθυρο προς άντληση πληροφοριών:

### Υποχρεωτικά μαθήματα Α εξ. (μεμονωμένη απεικόνιση)

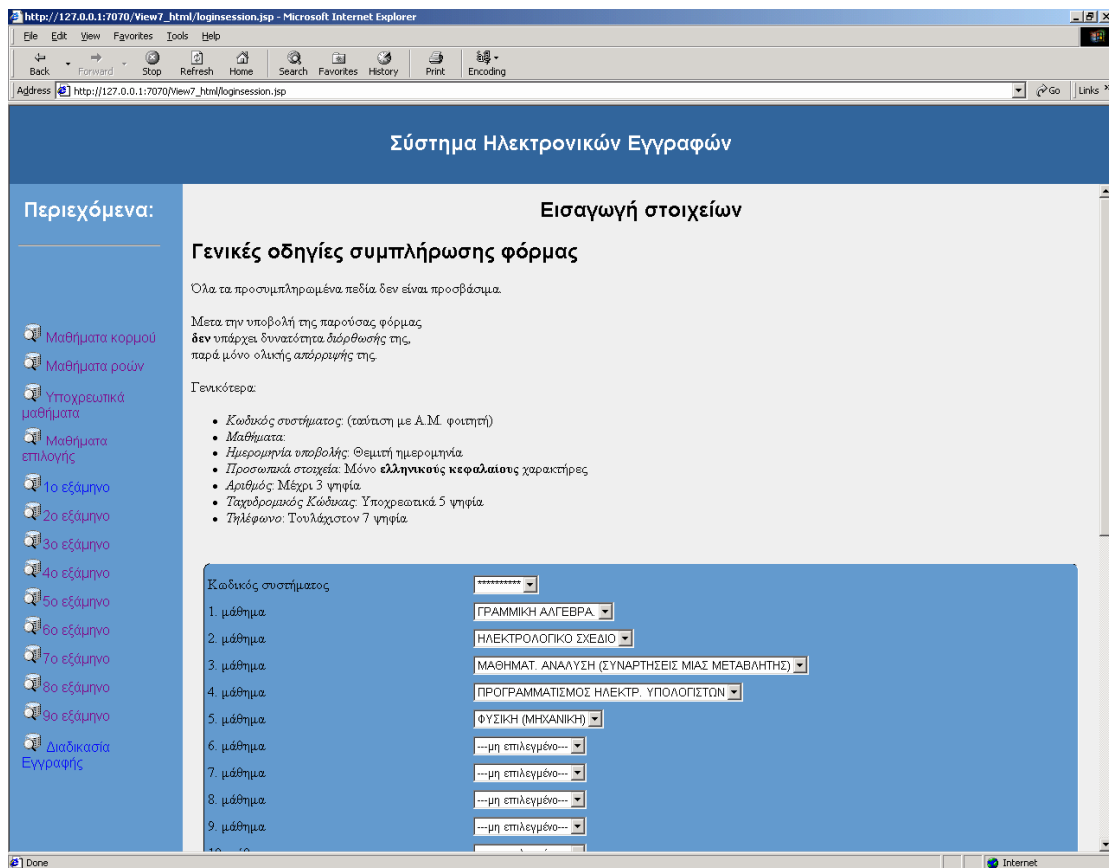
Κωδικός μαθήματος	3132
Τίτλος μαθήματος	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ
Ώρες θεωρίας	5
Ακαδημαϊκό έτος	2001

Εικόνα 7.3.6.2.κ

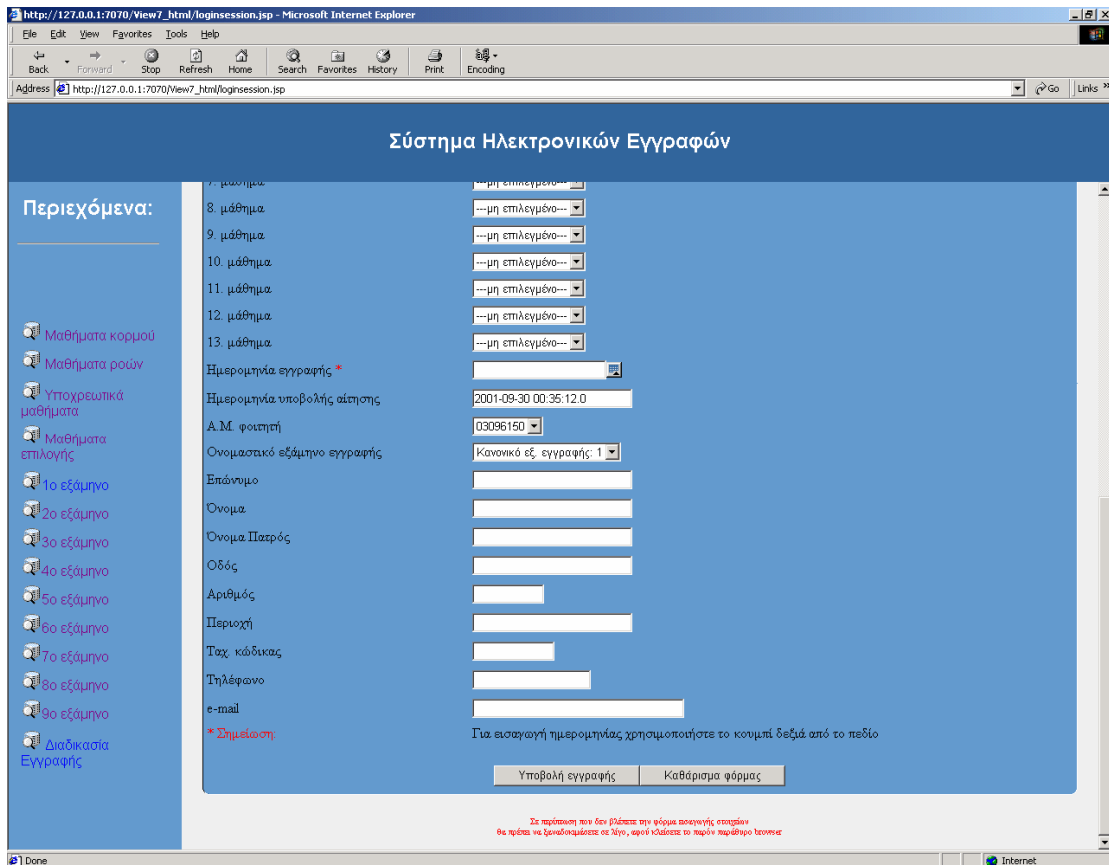
Αφού πραγματοποιηθούν οι επιλογές του χρήστη, υποβάλλεται η φόρμα και ανοίγει ένα pop-up window, το οποίο περιέχει ένα μήνυμα προς το χρήστη, δεν είναι δυνατόν να το κλείσει αυτός και παραμένει πάντα στο foreground, ούτως ώστε να εξαναγκασθεί ο χρήστης να συνεχίσει την εγγραφή του υποβάλλοντάς το, πράγμα ουσιώδες για το σύστημα.



Εικόνα 7.3.6.2.λ



Εικόνα 7.3.6.2.μ-1



Εικόνα 7.3.6.2.μ-2

Στις δύο παραπάνω εικόνες φαίνεται η συνολική φόρμα εισαγωγής στοιχείων του φοιτητή στο σύστημα, η οποία και παρέχει όλες τις απαραίτητες πληροφορίες προκειμένου να πραγματοποιήσει απροβλημάτιστα την εγγραφή του σε αυτό.

Όπως και στην περίπτωση επιλογής κατευθύνσεων/ροών και εδώ υπάρχει σελίδα με δυνατότητα ολικής απόρριψης ή ολικής επιβεβαίωσης των στοιχείων που εισάγει ο χρήστης, δυνατότητα καθάρισμα της φόρμας πριν την υποβολή της, και ανάλογη μεταφορά του χρήστη είτε στη σελίδα Rollback, είτε στη σελίδα επιτυχίας-εκτύπωσης της αίτησής του. Αναλυτικά τα παραπάνω παρατίθενται στις παρακάτω εικόνες:



Κωδικός συστήματος	fmylonas
1. μάθημα	3132
2. μάθημα	3035
3. μάθημα	3131
4. μάθημα	3020
5. μάθημα	3049
6. μάθημα	3028
7. μάθημα	3087
8. μάθημα	3145
9. μάθημα	
10. μάθημα	
11. μάθημα	
12. μάθημα	
13. μάθημα	
Ημερομηνία	2001-09-19
Ημερομηνία υποβολής αίτησης	2001-09-30 00:43:56.0
A.M. φοιτητή	03096150
Ονομαστικό εξάμηνο εγγραφής	01
Επώνυμο	ΜΥΛΩΝΑΣ
Όνομα	ΦΟΙΒΟΣ
Όνομα Πατρός	ΙΩΑΝΝΗΣ
Οδός	ΒΗΣΣΑΡΙΩΝΟΣ
Αριθμός	6
Περιοχή	ΚΟΛΩΝΑΚΙ
Ταχ. κώδικας	10672
Τηλέφωνο	3636887
Email	fmylonas@dblab.ntua.gr

Εικόνα 7.3.6.2.ν

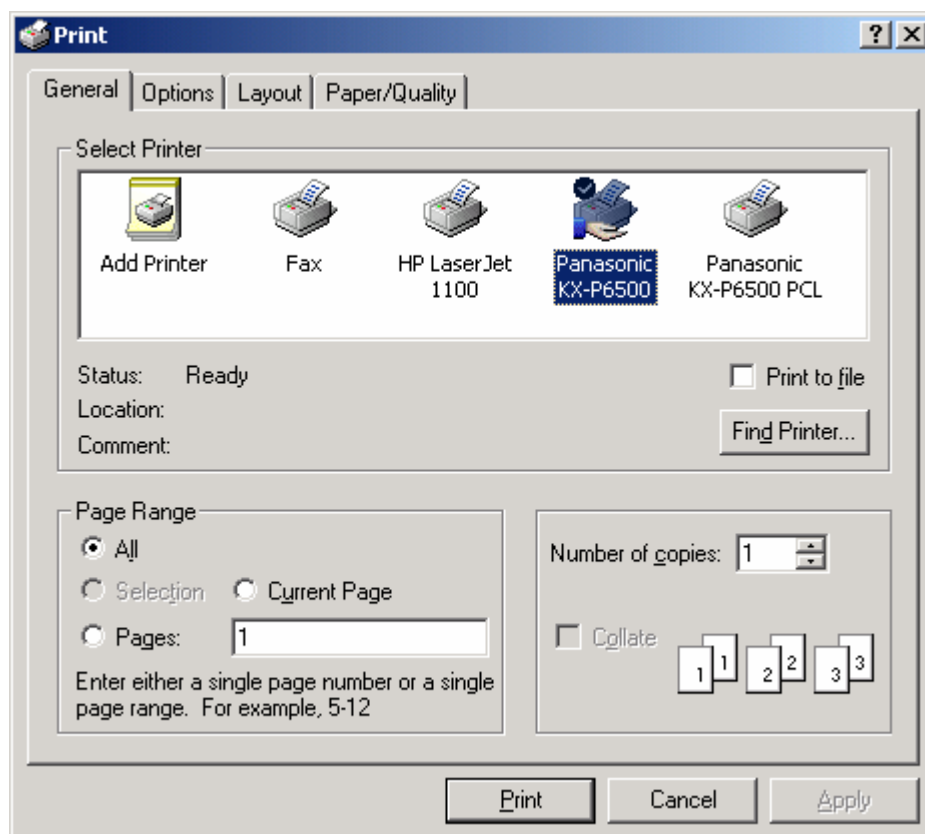
- Πράσινο βέλος:

Ημερομηνία υποβολής αίτησης	2001-09-30 00:43:56.0
A.M. φοιτητή	03096150
Ονομαστικό εξάμηνο εγγραφής	01
Επώνυμο	ΜΥΛΩΝΑΣ
Όνομα	ΦΟΙΒΟΣ
Όνομα Πατρός	ΙΩΑΝΝΗΣ
Οδός	ΒΗΣΣΑΡΙΩΝΟΣ
Αριθμός	6
Περιοχή	ΚΟΛΩΝΑΚΙ
Ταχ. κώδικας	10672
Τηλέφωνο	3636887

Υπογραφή:

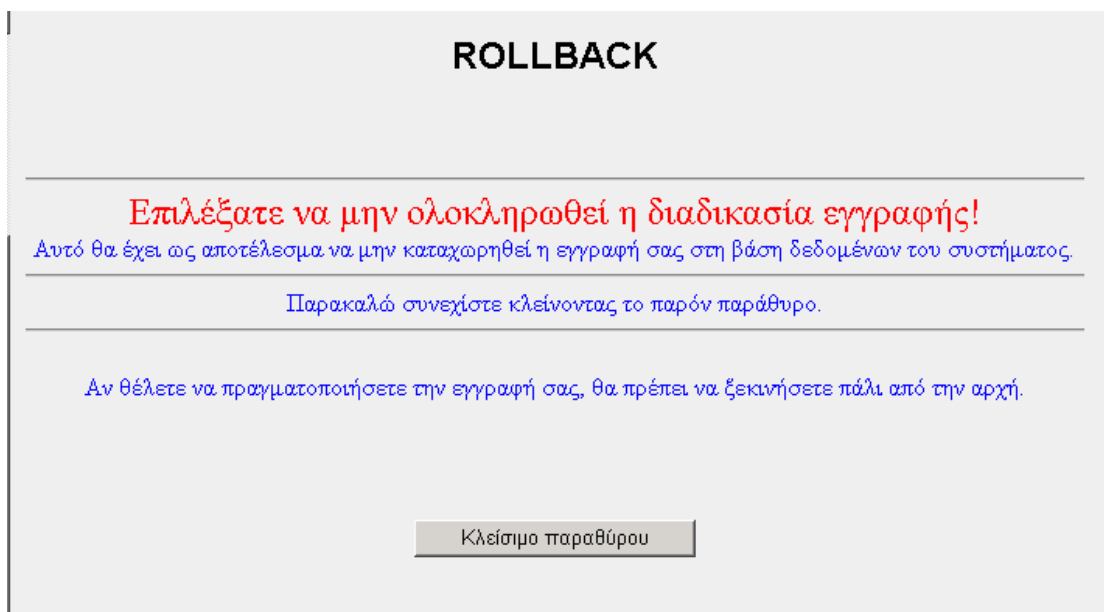
Εκτύπωση

Εικόνα 7.3.6.2.ξ



Εικόνα 7.3.6.2.ο

- **Κόκκινο** βέλος:

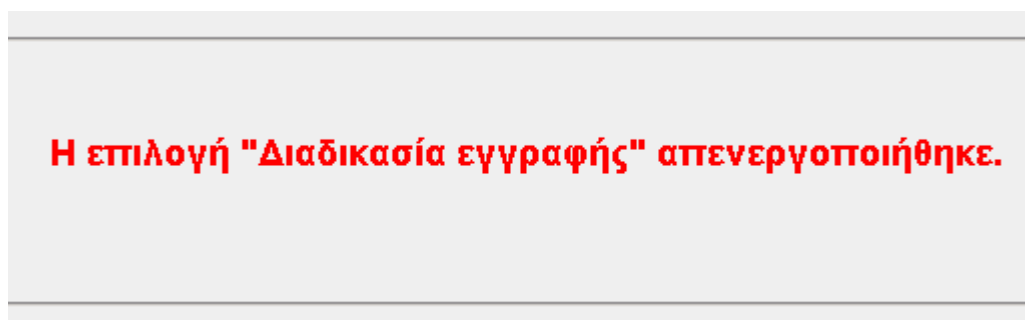


Εικόνα 7.3.6.2.π



Εικόνα 7.3.6.2.ρ

Σε περίπτωση, τέλος, που επιχειρηθεί να επαναεγγραφεί κακόβουλα κάποιος χρήστης του συστήματος μετά την επιτυχή ολοκλήρωση της εγγραφής του, αυτό καθίσταται αδύνατο μέσω της «παρακολούθησης» του καθόλη τη διάρκεια της πλοήγησής του στο σύστημα, οπότε και του παρέχεται μία διαφορετική σελίδα προειδοποίησης αντί της κανονικής αρχικής σελίδας επιλογής τύπου εγγραφής στο σύστημα.



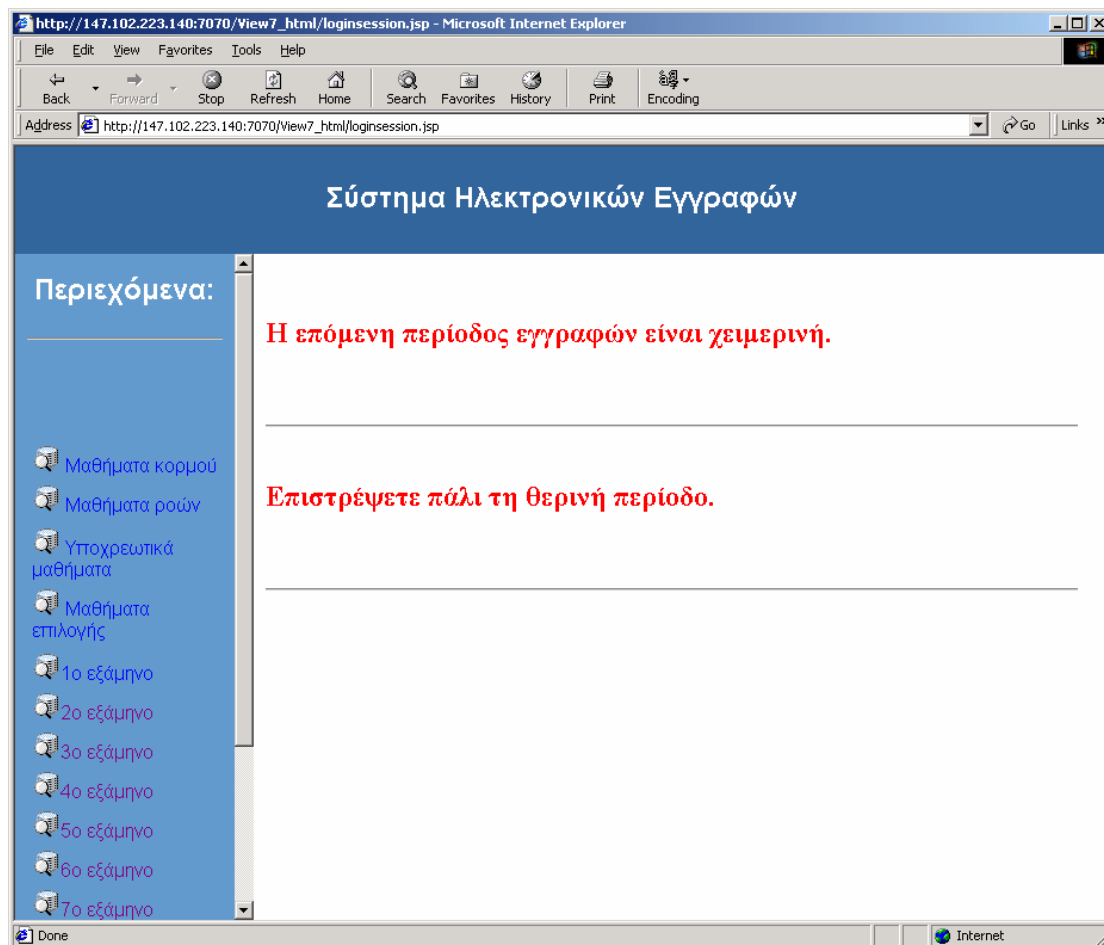
Εικόνα 7.3.6.2.σ



### 7.3.6.3 Διαδικασία εγγραφής – **Θερινά** μαθήματα:

Εντελώς ανάλογες διαδικασίες, με διαφορετικούς ελέγχους βέβαια, πραγματοποιούνται και κατά την διαδικασία εγγραφής σε θερινά μαθήματα (υποσύνολο τω οποίων αποτελεί το 6<sup>ο</sup> εξάμηνο το οποίο είδαμε στην διαδικασία επιλογής κατεύθυνσης/ροών.

Σημαντική λεπτομέρεια του όλου συστήματος αποτελεί το γεγονός ότι προσβάσιμο θεωρητικά κατά τη χειμερινή περίοδο εγγραφών είναι μόνο το τμήμα των χειμερινών μαθημάτων και vice-versa.



Εικόνα 7.3.6.3.a

## 7.4 Υποσύστημα γραμματείας:

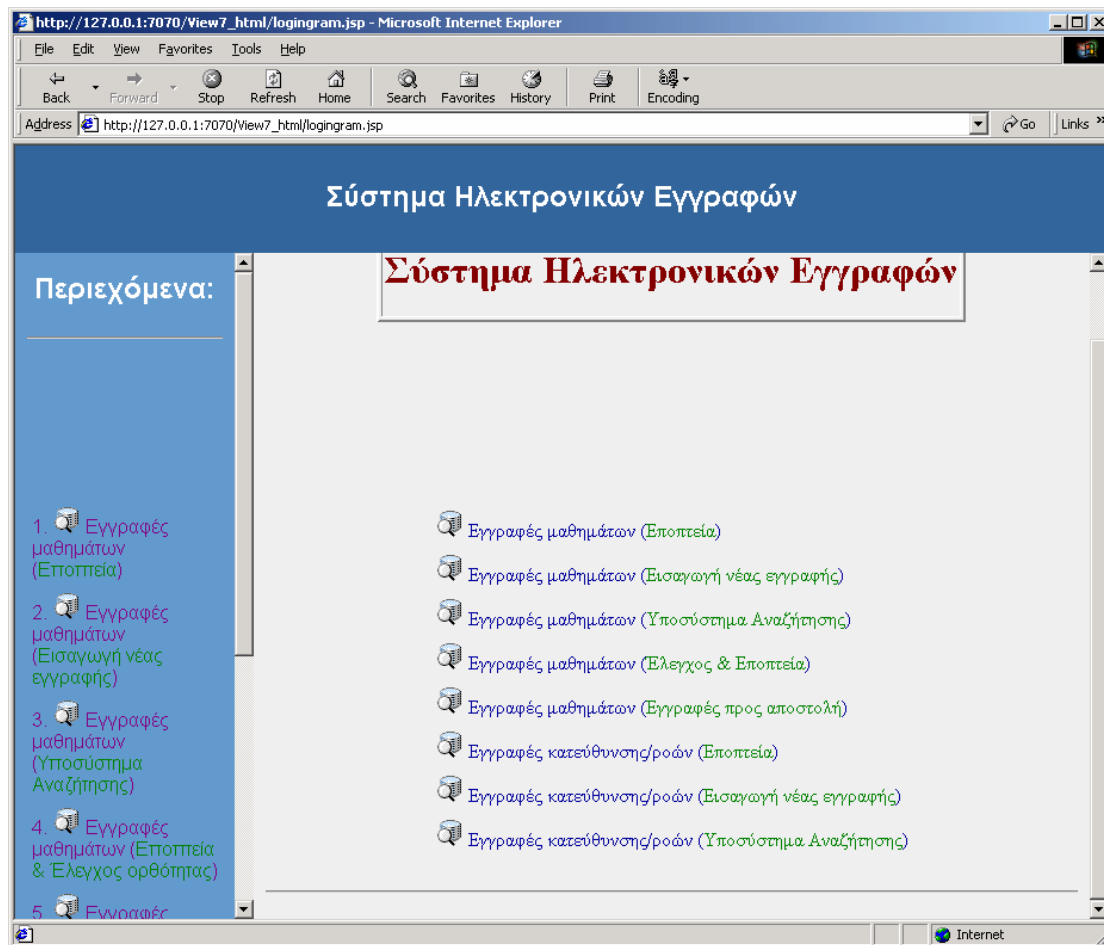
Αμέσως μετά το ακόλουθο login prompt,

The screenshot shows a Microsoft Internet Explorer window with the title 'Authentication χρήστη - Microsoft Internet Explorer'. The address bar contains the URL 'http://147.102.223.140:7070/View7\_html/logingram.jsp'. The main content area displays a login form with the following elements:

- Title:** Εξακρίβωση στοιχείων
- Text:** Παρακαλώ εισάγετε τον προσωπικό κωδικό συστήματος και το password που θα επιτρέψουν την είσοδό σας στο Σύστημα Ηλεκτρονικών Εγγραφών:
- Username:** A text input field.
- Password:** A text input field.
- Submit Button:** Εισαγωγή στοιχείων

Εικόνα 7.4.α

ο χρήστης της γραμματείας θα βρεθεί στην εξής σελίδα,



Εικόνα 7.4.β

απ' όπου είναι σε θέση να έχει full access πάνω στους πίνακες της βάσης που εισάγουν οι φοιτητές και μόνο σε αυτούς. Η πρόσβαση μπορεί να πραγματοποιηθεί είτε μέσω των απλών links του αριστερού frame, είτε μέσω της ανάλογης με προηγούμενως αλλά πολύ περισσότερο εμπλουτισμένης toolbar.



Εικόνα 7.4.γ

Μέσω αυτής μπορούν να πραγματοποιηθούν ολικές διαγραφές εγγραφών φοιτητών, εισαγωγές νέων εγγραφών, αναζητήσεις εγγραφών με διάφορα κριτήρια, συνολική απεικόνιση όλων των εγγραφών, διόρθωση συγκεκριμένης εγγραφής και προφανώς απεικόνιση μεμονωμένης εγγραφής.

Οι δυνατότητες για πλοήγηση του χρήστη της γραμματείας ομαδοποιούνται στα εξής 8 links:

- Εγγραφές μαθημάτων – Εποπτεία
- Εγγραφές μαθημάτων – Εισαγωγή νέας εγγραφής
- Εγγραφές μαθημάτων – Υποσύστημα αναζήτησης
- Εγγραφές μαθημάτων – Έλεγχος & Εποπτεία
- Εγγραφές μαθημάτων – Εγγραφές προς αποστολή
- Εγγραφές κατεύθυνσης/ροών – Εποπτεία
- Εγγραφές κατεύθυνσης/ροών – Εισαγωγή νέας εγγραφής
- Εγγραφές κατεύθυνσης/ροών – Υποσύστημα αναζήτησης

#### **7.4.1 Εγγραφές μαθημάτων - Εποπτεία:**

Αποτελεί την πρώτη ομάδα links όπου παρουσιάζονται εισαγωγικά οι εγγραφές των φοιτητών και υπάρχει εποπτική και μεμονωμένη δυνατότητα απεικόνισής τους. Υπάρχουν 3 δυνατές απεικονίσεις των επιλογών των φοιτητών προς τη γραμματεία, η απλή χρήση κωδικών για τα μαθήματα, η απλή χρήση ονομάτων για τα μαθήματα και η σύνθετη ταυτόχρονη απεικόνιση και των κωδικών και των ονομάτων των επιλεγμένων μαθημάτων προς ιδιαίτερη διευκόλυνση του προσωπικού, όπως άλλωστε παρατηρούμε και στις 5 εικόνες που ακολουθούν (Εικόνες **7.4.1.α – 7.4.1.ε**):

http://127.0.0.1:7070/View7\_html/loggingram.jsp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Print Encoding

Address http://127.0.0.1:7070/View7\_html/loggingram.jsp Go Links >>

## Σύστημα Ηλεκτρονικών Εγγραφών

### Γραμματεία - Πλοήγηση στις εγγραφές

Περιεχόμενα:

1. Πίνακας φοιτητών (Εγγραφές μαθημάτων) - (Μεμονωμένη απεικόνιση)
3. Πίνακας φοιτητών (Εισαγωγή εγγραφής μαθημάτων) - (Νέο record)
4. Πίνακας φοιτητών (Υποσύστημα Αναζήτησης) - (SQL/A.M./Κωδικός)
5. Πίνακας φοιτητών (Εγγραφές)

Αριθμός record	Κωδικός συστήματος	Κωδικός 1. μαθήματος	Κωδικός 2. μαθήματος	Κωδικός 3. μαθήματος	Κωδικός 4. μαθήματος	Κωδικός 5. μαθήματος	Κωδικός 6. μαθήματος	Κωδικός 7. μαθήματος
1	45639	3035	1000	1000	1000	1000	1000	1000
2	98746	3132	3035	3131	3020	3049	3104	3303
3	65897	3145	3083	3102	3167	3307	3224	3255
4	65785							
5	456354	3037						
6	7890							
7	321456789							
8	123456789	3145	1000					
9	2365	3208						
10	3256	3256						
11	6545	3208						
12	5554	3208						
13	4545	3145						

Internet

Εικόνα 7.4.1.α

http://127.0.0.1:7070/View7\_html/loggingram.jsp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Print Encoding

Address http://127.0.0.1:7070/View7\_html/loggingram.jsp Go Links >>

## Σύστημα Ηλεκτρονικών Εγγραφών

### Γραμματεία - Μεμονωμένη απεικόνιση εγγραφών

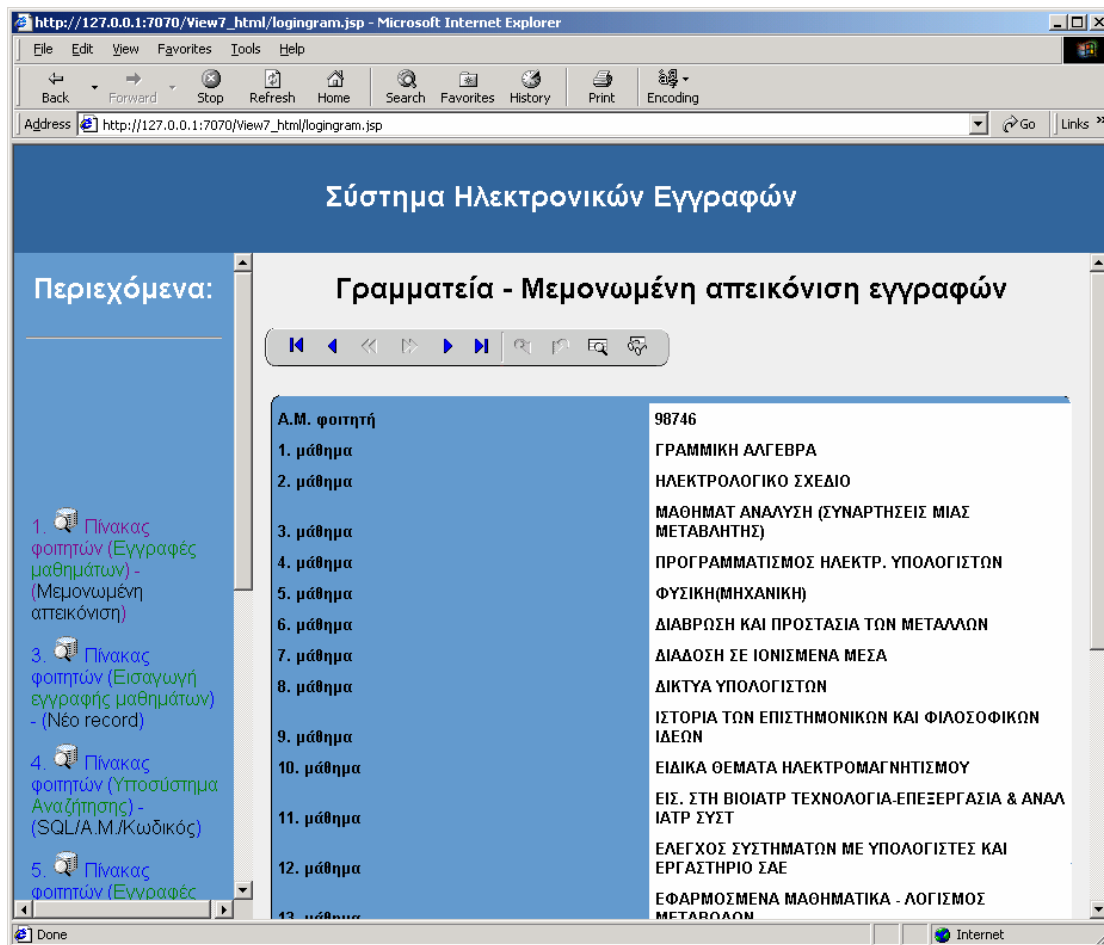
Περιεχόμενα:

1. Πίνακας φοιτητών (Εγγραφές μαθημάτων) - (Μεμονωμένη απεικόνιση)
3. Πίνακας φοιτητών (Εισαγωγή εγγραφής μαθημάτων) - (Νέο record)
4. Πίνακας φοιτητών (Υποσύστημα Αναζήτησης) - (SQL/A.M./Κωδικός)
5. Πίνακας φοιτητών (Εννοαφές)

A.M. φοιτητή	98746
1. μάθημα	3132
2. μάθημα	3035
3. μάθημα	3131
4. μάθημα	3020
5. μάθημα	3049
6. μάθημα	3104
7. μάθημα	3303
8. μάθημα	3298
9. μάθημα	3027
10. μάθημα	3301
11. μάθημα	3331
12. μάθημα	3304
13. μάθημα	3167
Ημερομηνία εγγραφής	2001-07-09
Ημερομηνία εγγραφής (αυτόματη εισαγωγή από το σύστημα)	2001-07-01 23:16:28.0
Αριθμός Μητρώου	03098745

Done Internet

Εικόνα 7.4.1.β



Εικόνα 7.4.1.γ

http://127.0.0.1:7070/View7\_html/loggingram.jsp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Print Encoding

Address http://127.0.0.1:7070/View7\_html/loggingram.jsp Go Links >>

## Σύστημα Ηλεκτρονικών Εγγραφών

### Γραμματεία - Πλοήγηση στις εγγραφές

Περιεχόμενα:

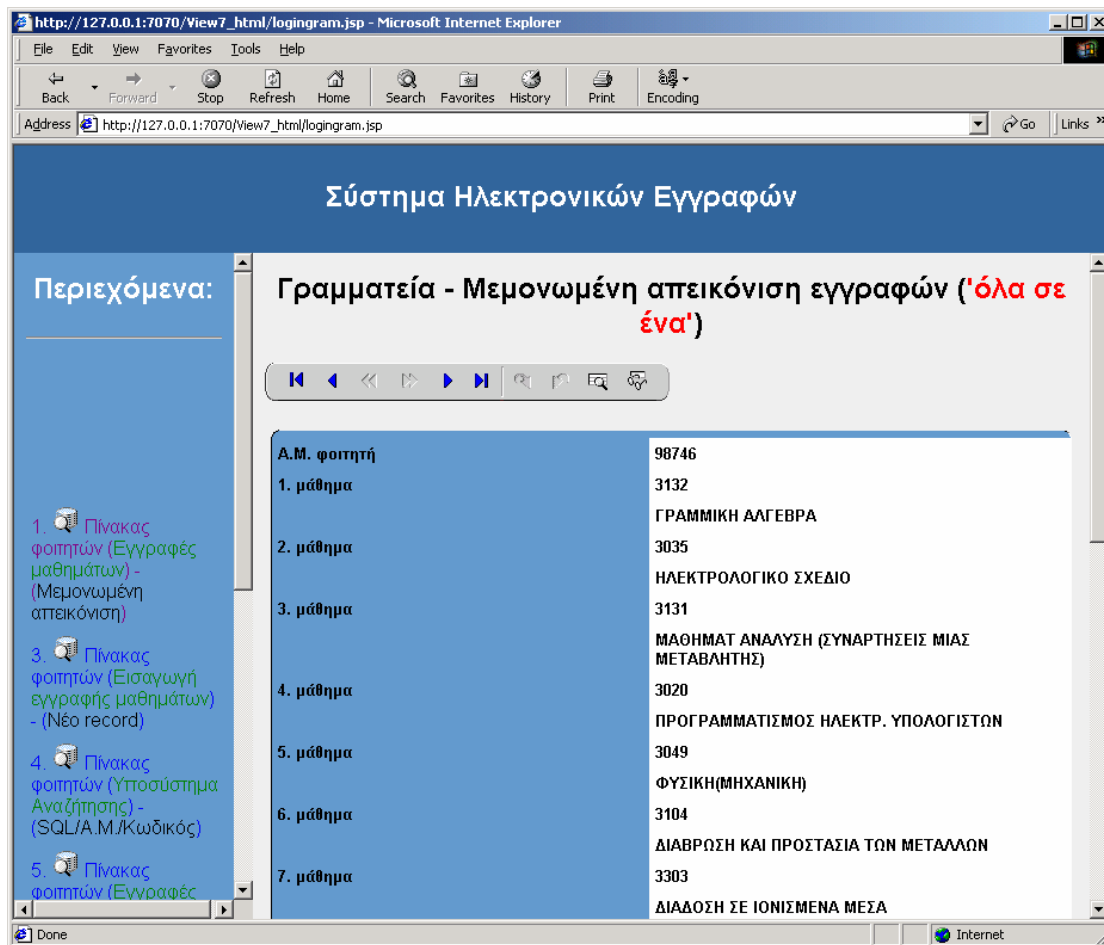
1. Πίνακας φοιτητών (Εγγραφές μαθημάτων) - (Μεμονωμένη απεικόνιση)
3. Πίνακας φοιτητών (Εισαγωγή εγγραφής μαθημάτων) - (Νέο record)
4. Πίνακας φοιτητών (Υποσύστημα Αναζήτησης) - (SQL/Α.Μ./Κωδικός)
5. Πίνακας φοιτητών (Εγγραφές)

Αριθμός record	Κωδικός συστήματος	1. μάθημα	2. μάθημα	3. μάθημα	4. μάθημα	5. μάθημα
1	45639	ΗΛΕΚΤΡΟΛΟΓΙΚΟ ΣΧΕΔΙΟ	ΔΕΝ ΕΧΕΙ ΕΠΙΛΕΓΕΙ	ΔΕΝ ΕΧΕΙ ΕΠΙΛΕΓΕΙ	ΔΕΝ ΕΧΕΙ ΕΠΙΛΕΓΕΙ	ΔΕΙ
2	98746	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	ΗΛΕΚΤΡΟΛΟΓΙΚΟ ΣΧΕΔΙΟ	ΜΑΘΗΜΑΤ ΑΝΑΛΥΣΗ (ΣΥΝΑΡΤΗΣΕΙΣ ΜΙΑΣ ΜΕΤΑΒΛΗΤΗΣ)	ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΗΛΕΚΤΡ. ΥΠΟΛΟΓΙΣΤΩΝ	ΦΥ (ΜΗ
3	877	ΦΙΛΟΣΟΦΙΑ	ΓΑΛΛΙΚΗ ΓΛΩΣΣΑ	ΟΠΤΟΗΛΕΚΤΡΟΝΙΚΗ	ΔΙΚΤΥΑ ΕΥΡΕΙΑΣ ΖΩΝΗΣ	ΔΟ ΕΠΙ

Internet

Εικόνα 7.4.1.δ





Εικόνα 7.4.1.ε

#### 7.4.2 Εγγραφές μαθημάτων - Εισαγωγή νέας εγγραφής:

Η δεύτερη ομάδα links αποτελείται από την ιδιαίτερη σελίδα εισαγωγής μιας εντελώς νέας εγγραφής σ τους πίνακες, λειτουργία που έχει όπως είδαμε ενσωματωθεί και στην toolbar στις σελίδες απεικόνισης.

http://127.0.0.1:7070/View7\_html/loggingram.jsp - Microsoft Internet Explorer

Address http://127.0.0.1:7070/View7\_html/loggingram.jsp

Σύστημα Ηλεκτρονικών Εγγραφών

Γραμματεία - Εισαγωγή νέας εγγραφής

Επιστροφή στην προηγούμενη σελίδα

Κωδικός φοιτητή (συστήματος ΗΕ)

1. μάθημα

2. μάθημα

3. μάθημα

4. μάθημα

5. μάθημα

6. μάθημα

7. μάθημα

8. μάθημα

9. μάθημα

10. μάθημα

11. μάθημα

12. μάθημα

13. μάθημα

ΔΙΑΧΕΙΡΙΣΗ ΕΝΕΡΓΕΙΑΣ & ΠΕΡΙΒΑΛΛΟΝΤΙΚΗ ΠΟΛΙΤΙΚΗ  
ΠΡΟΣΤΑΣΙΑ ΣΥΣΤΗΜ. ΗΛΕΚΤΡ. ΕΝΕΡΓΙΑΣ  
ΠΑΙΓΝΙΑ ΑΠΟΦΑΣΕΩΝ  
ΣΥΣΤΗΜΑΤΑ ΑΣΙΩΣΗΣ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗΣ  
ΕΠΕΚΑΤΑΣΤΑΣΗ ΔΙΑΧΕΙΡΙΣΗΣ & ΠΟΙΟΤ. ΕΛ. ΙΑΤΡ. & ΝΟΣΟΚΟΜΕΙΑΚΩΝ  
ΗΛΕΚΤΡΟΜ. & ΜΗΧΑΝΙΚΗ ΠΡΟΣΟΜΟΙΩΣΗ ΦΥΣΙΟΛ. ΣΥΣΤΗΜΑΤΩΝ  
ΦΥΣΙΚΗ ΤΩΝ ΔΙΑΗΛΕΚΤΡΙΚΩΝ ΥΛΙΚΩΝ  
ΣΤΟΙΧΕΙΑ ΓΕΩΡΓΙΑΣ ΑΡΙΘΜΩΝ & ΕΦΑΡΜΟΓΕΣ ΣΤΗΝ ΚΡΥΠΤΟΓΡΑΦΙΑ  
ΣΤΟΙΧΕΙΑ ΔΙΚΑΙΟΥ ΚΑΙ ΤΕΧΝΙΚΗΣ ΝΟΜΟΘΕΣΙΑΣ  
ΣΥΣΤΗΜΑΤΑ ΕΙΔΩΚΩΝ ΗΛΕΚΤΡΙΚΩΝ ΚΙΝΗΤΗΡΩΝ

Ημερομηνία εγγραφής \*

Ημερομηνία υποβολής αίτησης 2001-09-30 02:01:56.0

Α.Μ. φοιτητή

Ονομαστικό εξάμηνο εγγραφής

Επώνυμο

Όνομα

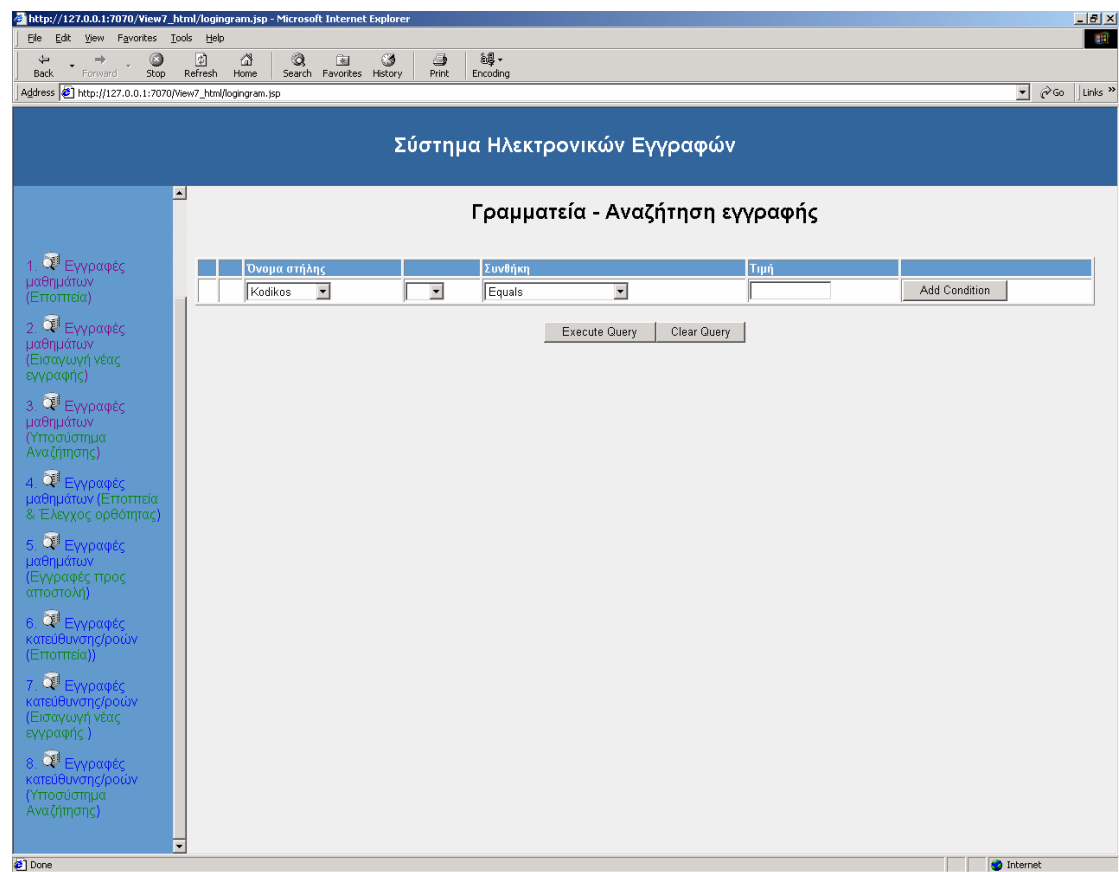
Όνομα Πατρός

Εικόνα 7.4.2.α

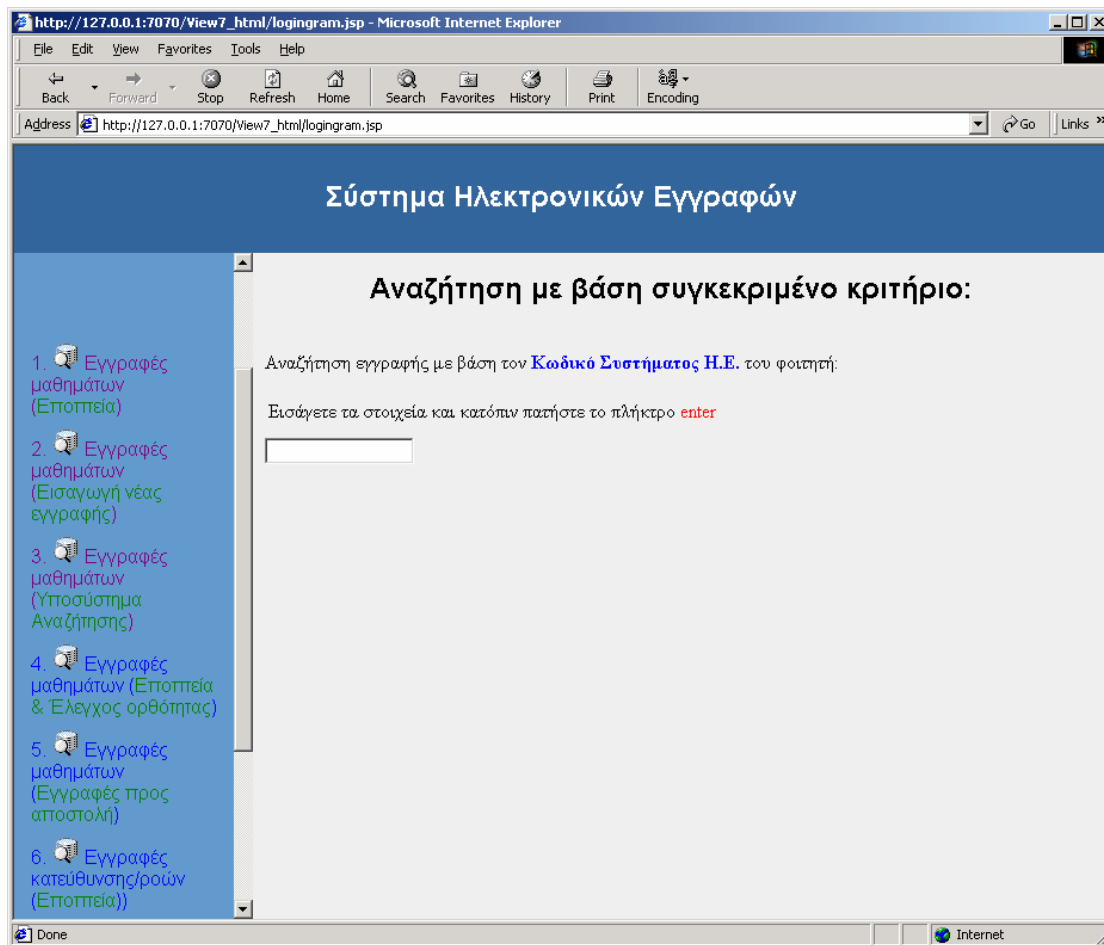
#### 7.4.3 Εγγραφές μαθημάτων – Υποσύστημα Αναζήτησης:

Το τρίτο group links αφορά τις δυνατότητες αναζήτησης εγγραφών στο σύστημα ΣΗΕ από την πιστοποιημένη ομάδα χρηστών της γραμματείας. Το σύστημα σε πρώτη φάση παρέχει 2 διαφορετικά είδη αναζήτησης, την «καθαρή» SQL αναζήτηση και την αναζήτηση με βάση συγκεκριμένο κριτήριο τιμής πεδίου, π.χ. με βάση τον αριθμό μητρώου ενός φοιτητή. Δίνεται επίσης η δυνατότητα να εμπλουτιστεί η παραπάνω διαδικασία με άμεση διόρθωση της εγγραφής που ανευρέθη και καταχώρησή της στη βάση του συστήματος<sup>8</sup>.

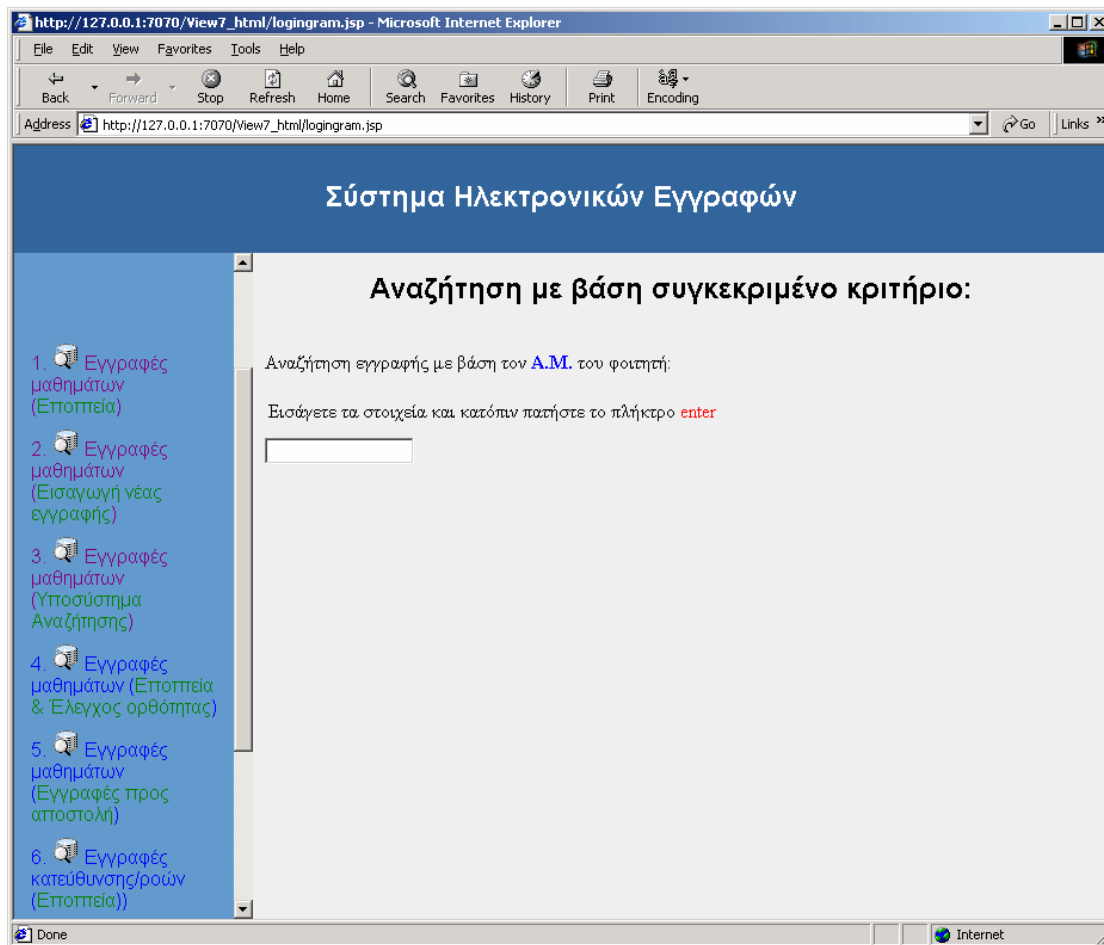
<sup>8</sup> Να σημειώσουμε εδώ, ότι η γραμματεία είναι σε θέση να διορθώσει οποιοδήποτε πεδίο της εγγραφής του φοιτητή, εκτός από την **Ημερομηνία υποβολής αίτησης**, η οποία και διαβάζει την ημερομηνία του συστήματος κατά την πραγματοποίησή του transaction.



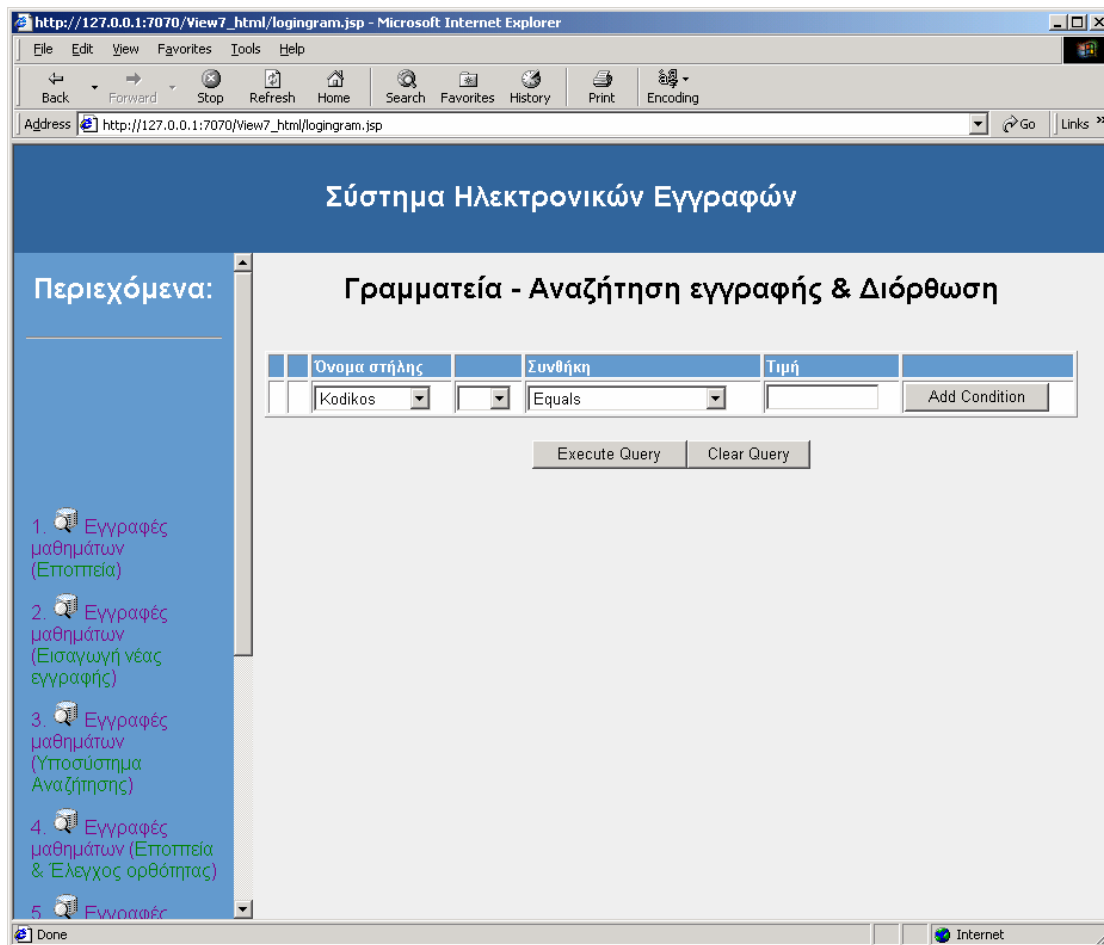
Εικόνα 7.4.3.α



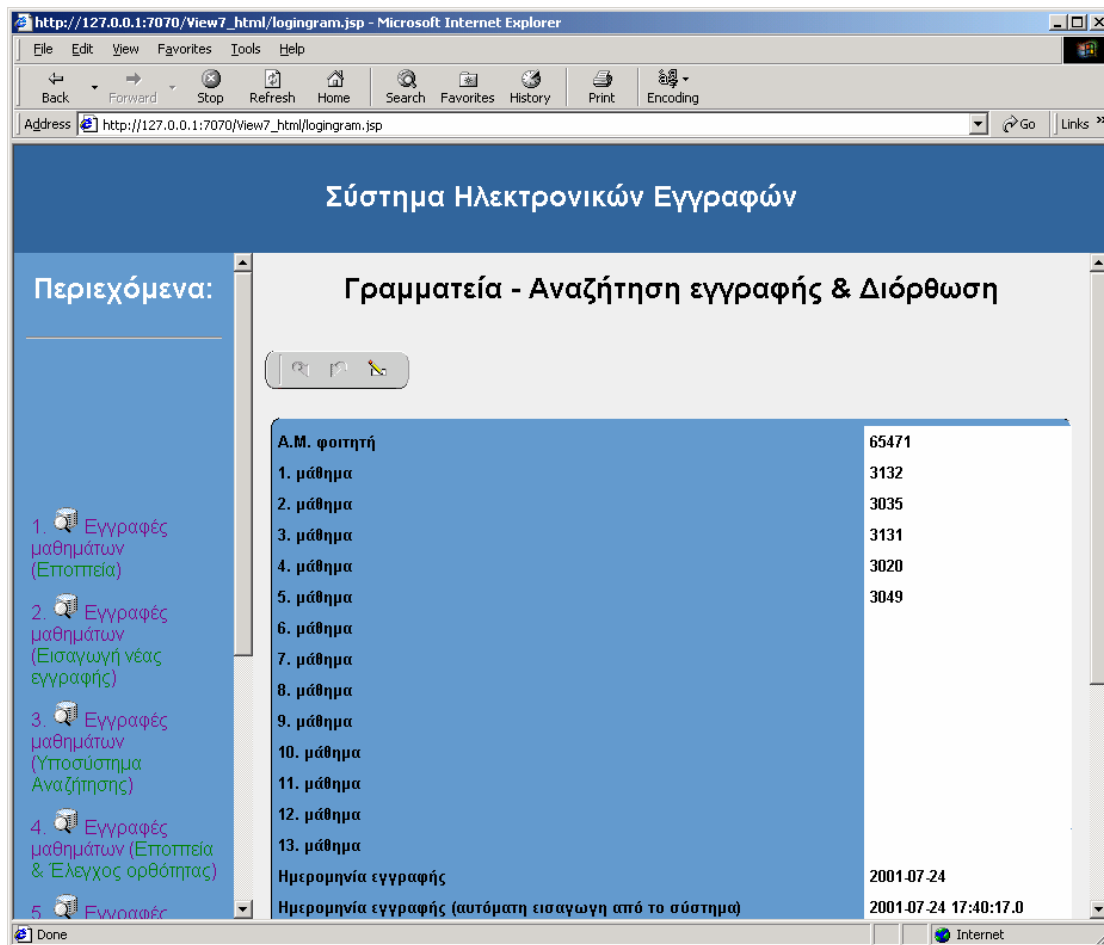
Εικόνα 7.4.3.β



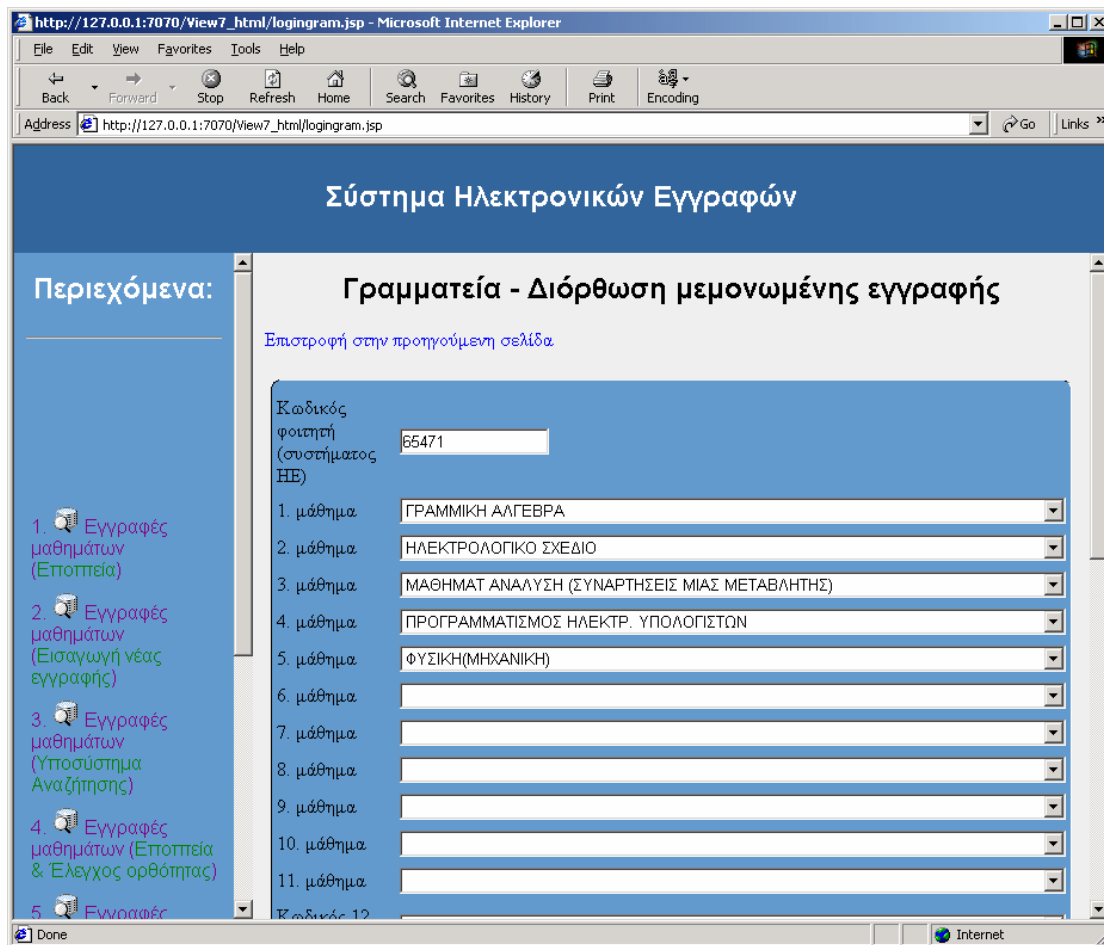
Εικόνα 7.4.3.γ



Εικόνα 7.4.3.δ

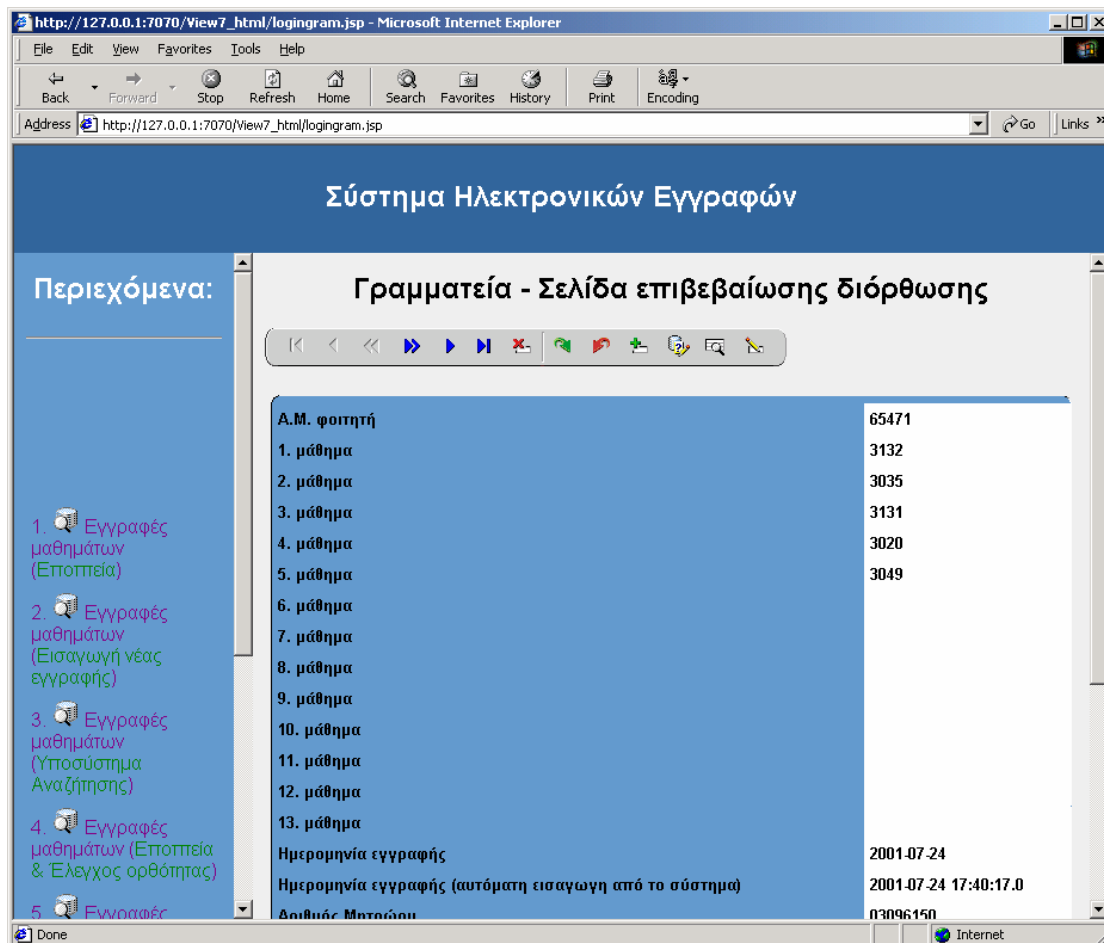


Εικόνα 7.4.3.ε



Εικόνα 7.4.3.στ

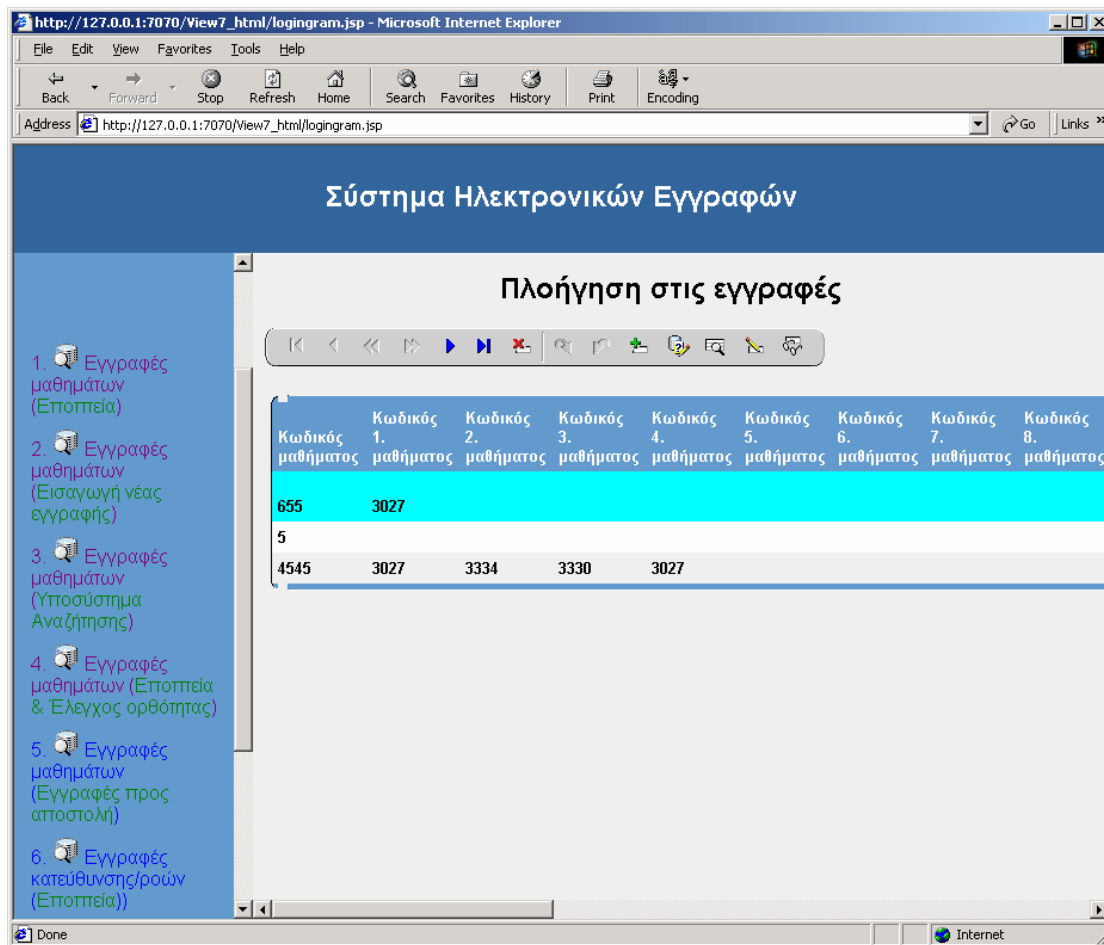




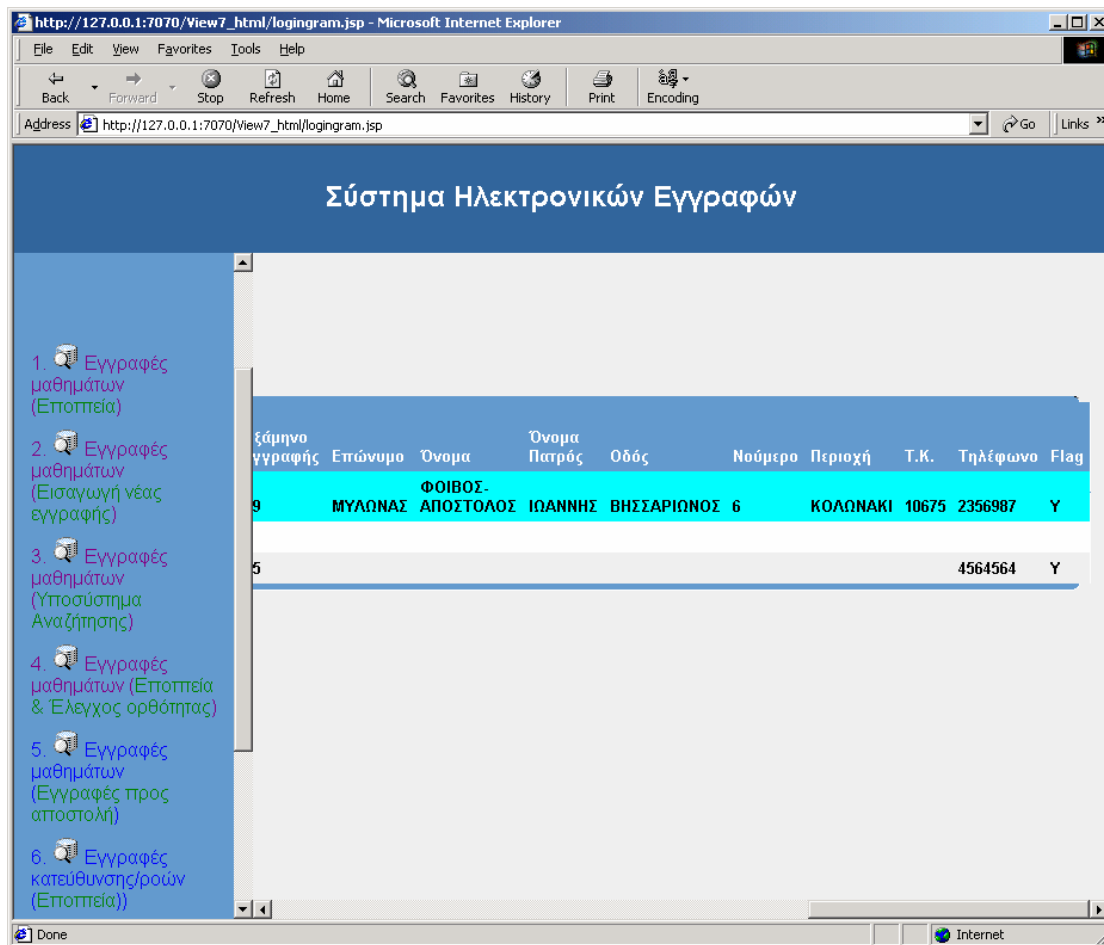
Εικόνα 7.4.3.η

#### 7.4.4 Εγγραφές μαθημάτων – Έλεγχος & Εποπτεία:

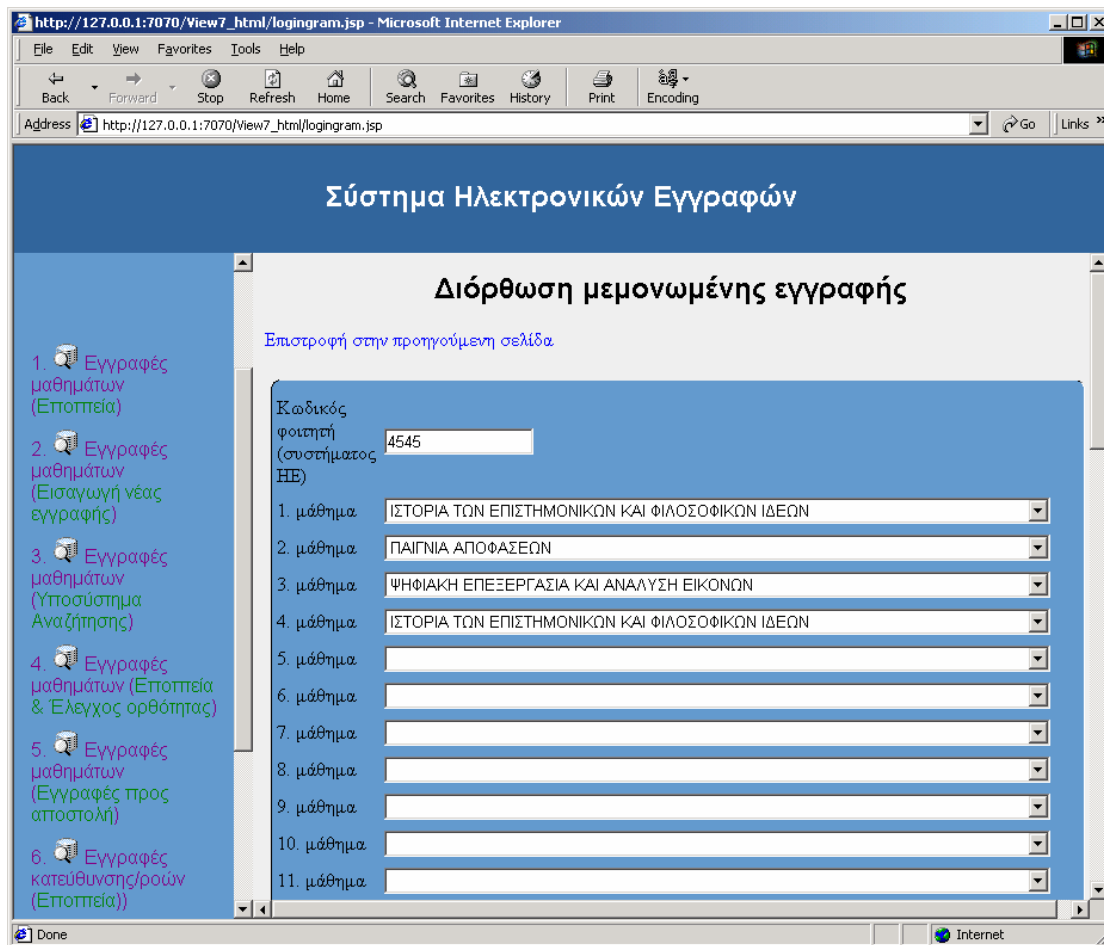
Στα πλαίσια της παρέμβασης της γραμματείας στις εγγραφές των φοιτητών, πριν αυτές καταλήξουν προς αποστολή στο σύστημα του φοιτητολογίου, πρέπει να υπάρχει μια διαδικασία ελέγχου και έγκρισής τους. Αυτή η διαδικασία πραγματοποιείται στην ομάδα των σελίδων αυτών που βρίσκονται πίσω από το link αυτό. Αναλυτικά, μπορούμε να έχουμε πλοήγηση στα πραγματικά records των πινάκων στη βάση, τα οποία διαθέτουν ακόμα ένα - κρυφό προς τους φοιτητές - πεδίο flag, το οποίο και καθορίζει αν μια εγγραφή είναι έγκυρη και έτοιμη προς αποστολή στο φοιτητολόγιο, ή όχι (Εικόνες 7.4.4.α & 7.4.4.β). Επίσης, από το σημείο αυτό μπορούμε να έχουμε ολική διόρθωση ενός τέτοιου record (και με διόρθωση του flag του δηλαδή) (Εικόνα 7.4.4.γ), σε αντίθεση με προηγούμενες προσεγγίσεις διόρθωσης, για λόγους π.χ. πολιτικής και αρμοδιοτήτων συγκεκριμένων χρηστών της γραμματείας. Είναι δυνατός ο έλεγχος μόνο του flag (Εικόνα 7.4.4.δ), σε συνδυασμό με το κλειδί του κάθε record, το οποίο εν προκειμένω είναι ο κωδικός συστήματος, ενώ το παραπάνω συνδυάζεται άμεσα και με browsing ή simple view των records (Εικόνα 7.4.4.ε). Τέλος, υπάρχει η πολύ σημαντική δυνατότητα απεικόνισης (μεμονωμένης ή/και σύνθετης) εγγραφών και ταυτόχρονη διόρθωσή τους «on the fly» στην ίδια σελίδα (Εικόνες 7.4.4.στ & 7.4.4.η).



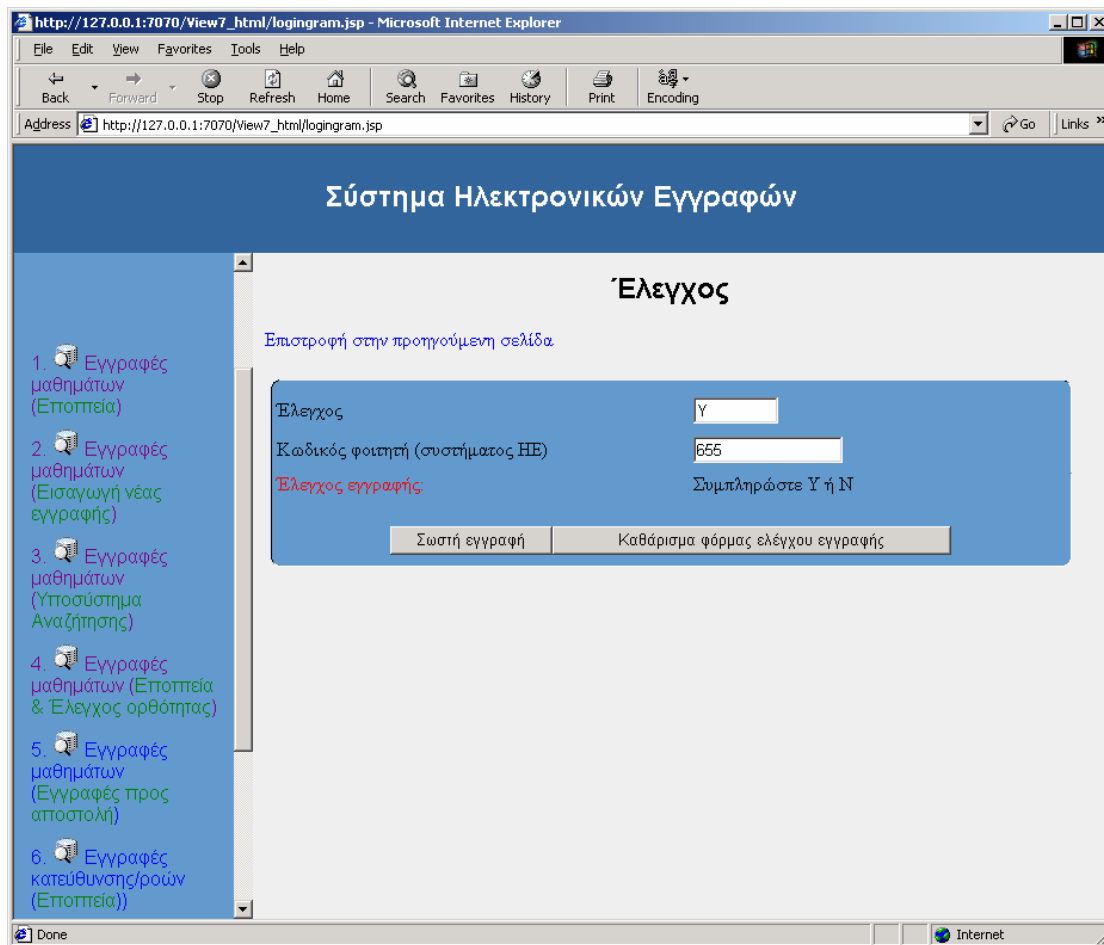
Εικόνα 7.4.4.α



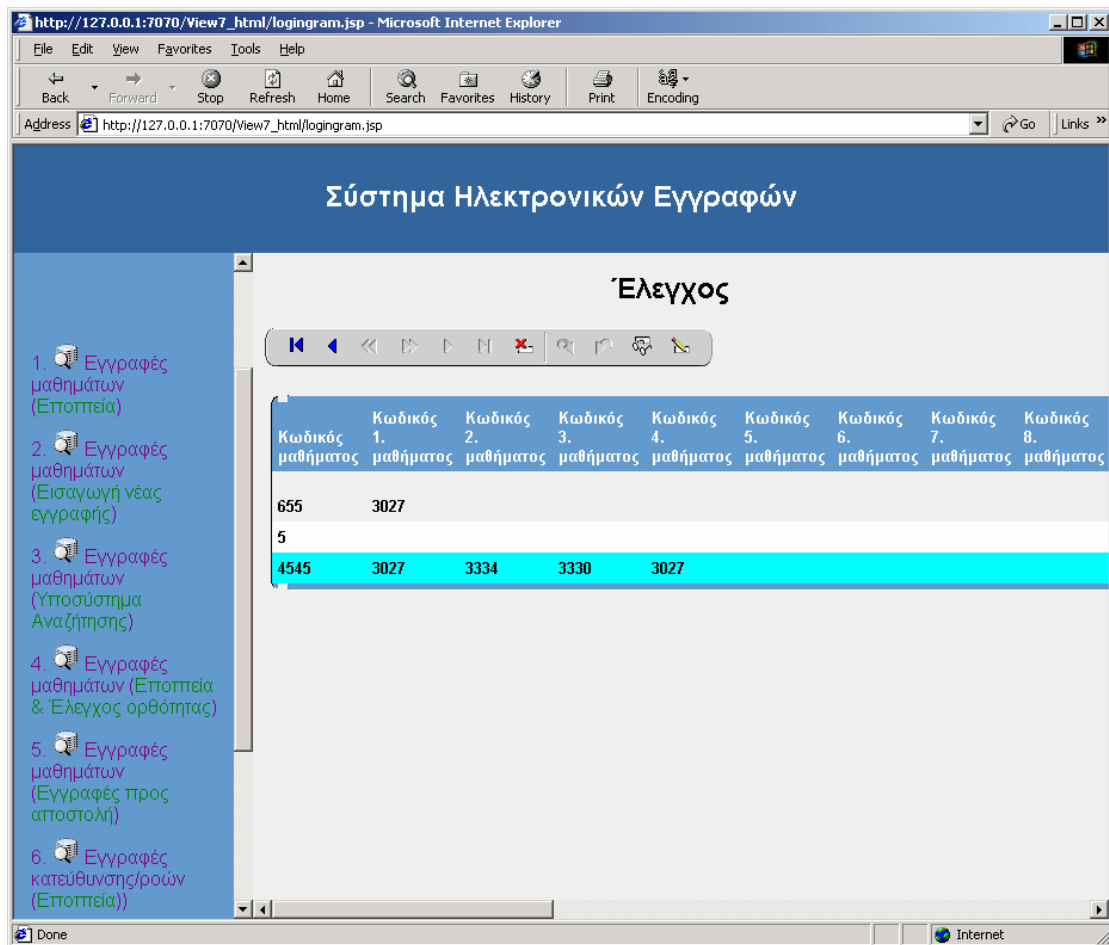
Εικόνα 7.4.4.β



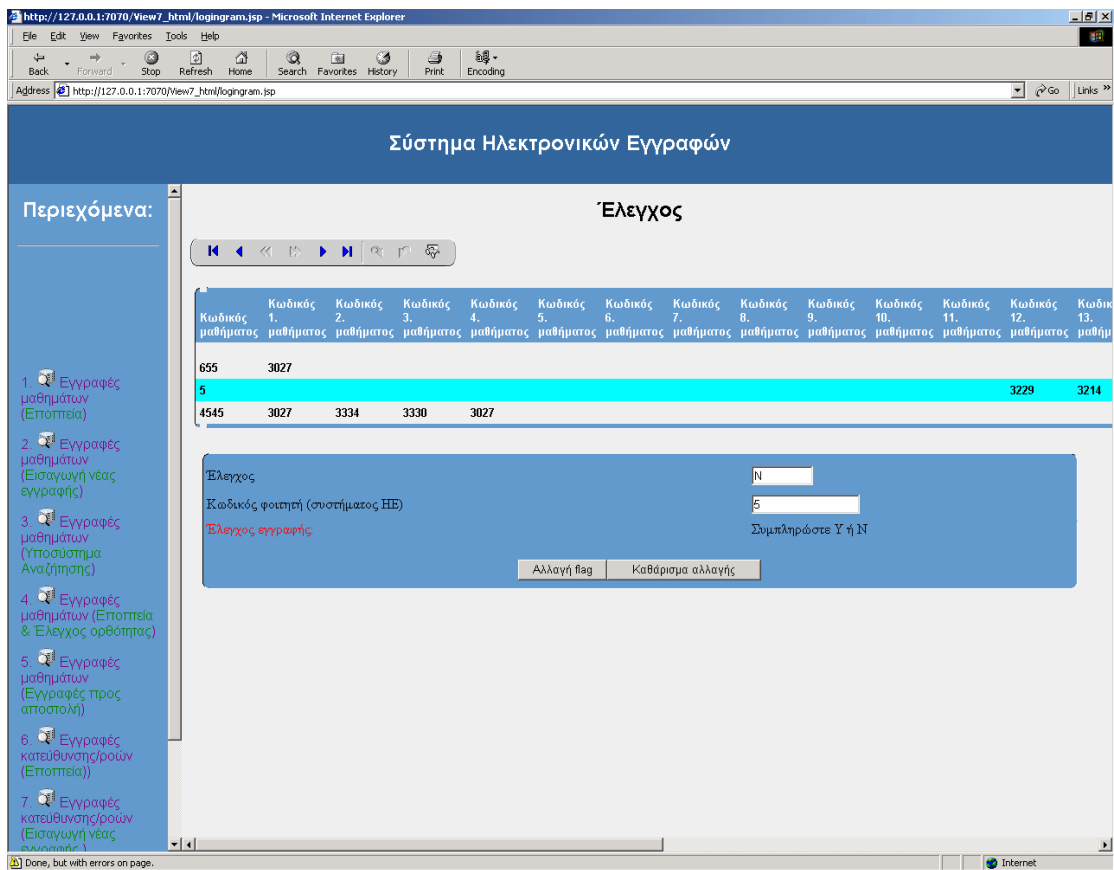
Εικόνα 7.4.4.γ



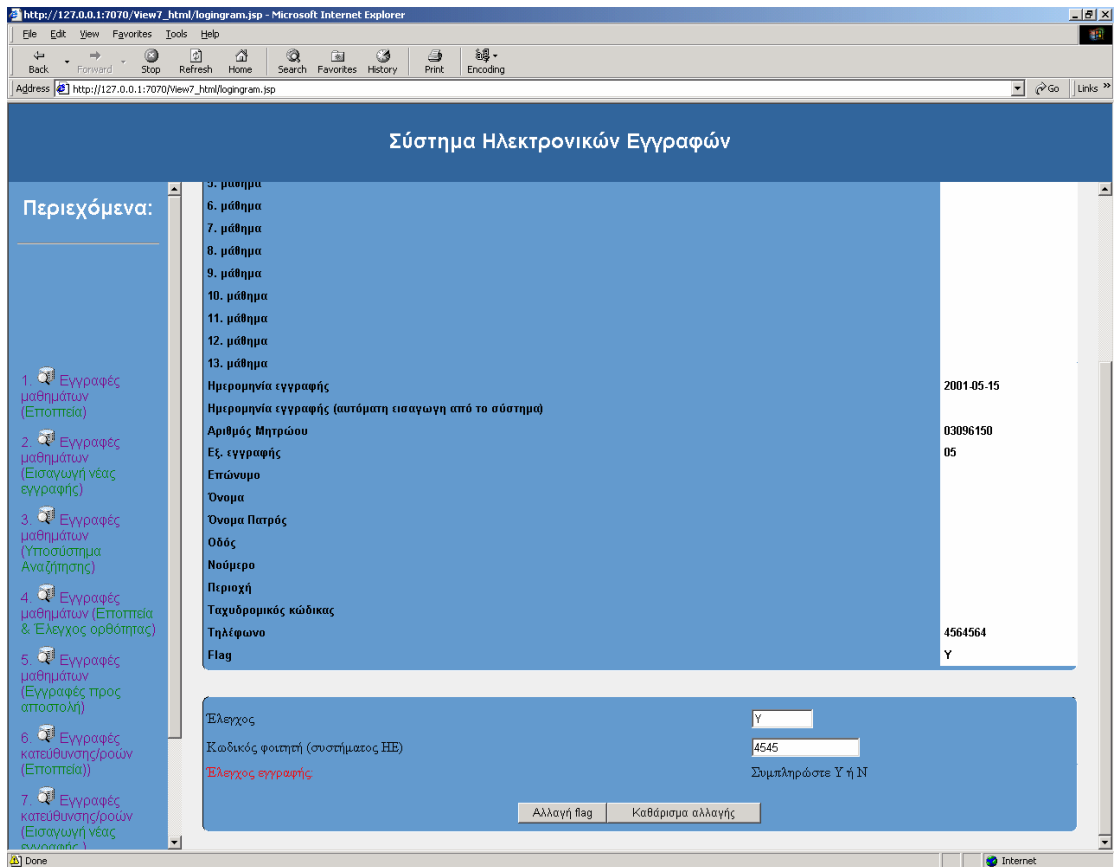
Εικόνα 7.4.4.δ



Εικόνα 7.4.4.ε



Εικόνα 7.4.4.στ



Εικόνα 7.4.4.η

#### 7.4.5 Εγγραφές μαθημάτων – Εγγραφές προς αποστολή:

Το πέμπτο link του αριστερού frame περιεχομένων δίνει στη γραμματεία την εποπτεία των εγγραφών των φοιτητών που έχουν ελεγχθεί και είναι έτοιμες προς αποστολή στο φοιτητολόγιο. Πρόκειται προφανώς για τις εγγραφές στις οποίες έχει παρέμβει η γραμματεία και έχει μετατρέψει το πεδίο flag τους, από «κενό» που τυχόν ήταν μετά τη δημιουργία του από τους φοιτητές, ή ακόμα και «N» μετά από προηγούμενη παρέμβασή της, σε «Y» που υποδηλώνει το έγκυρο της εγγραφής. Υπάρχει δυνατότητα τόσο απλής, όσο και σύνθετης συνολικής απεικόνισής τους.

Σύστημα Ηλεκτρονικών Εγγραφών

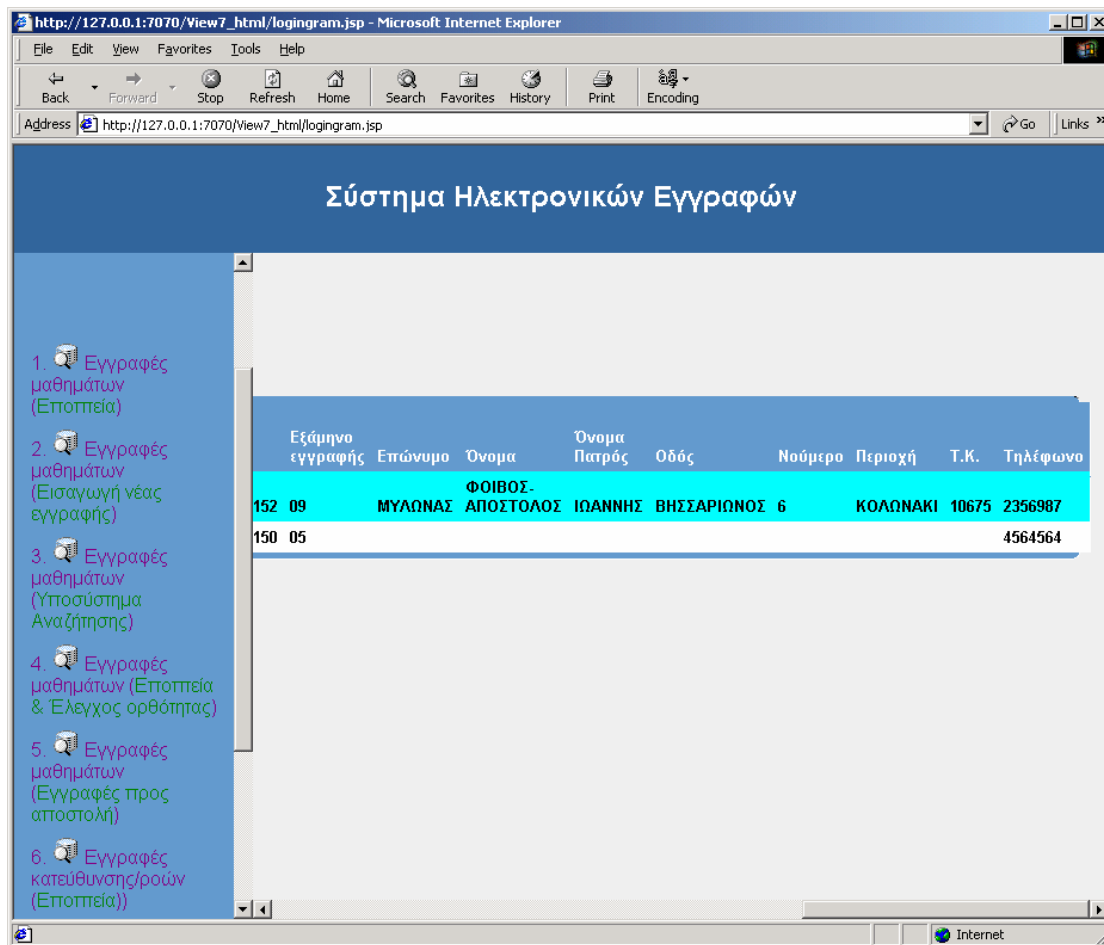
Πλοήγηση στις εγγραφές

Κωδικός μαθήματος	Κωδικός 1. μαθήματος	Κωδικός 2. μαθήματος	Κωδικός 3. μαθήματος	Κωδικός 4. μαθήματος	Κωδικός 5. μαθήματος	Κωδικός 6. μαθήματος	Κωδικός 7. μαθήματος	Κωδικός 8. μαθήματος
655	3027							
4545	3027	3334	3330	3027				

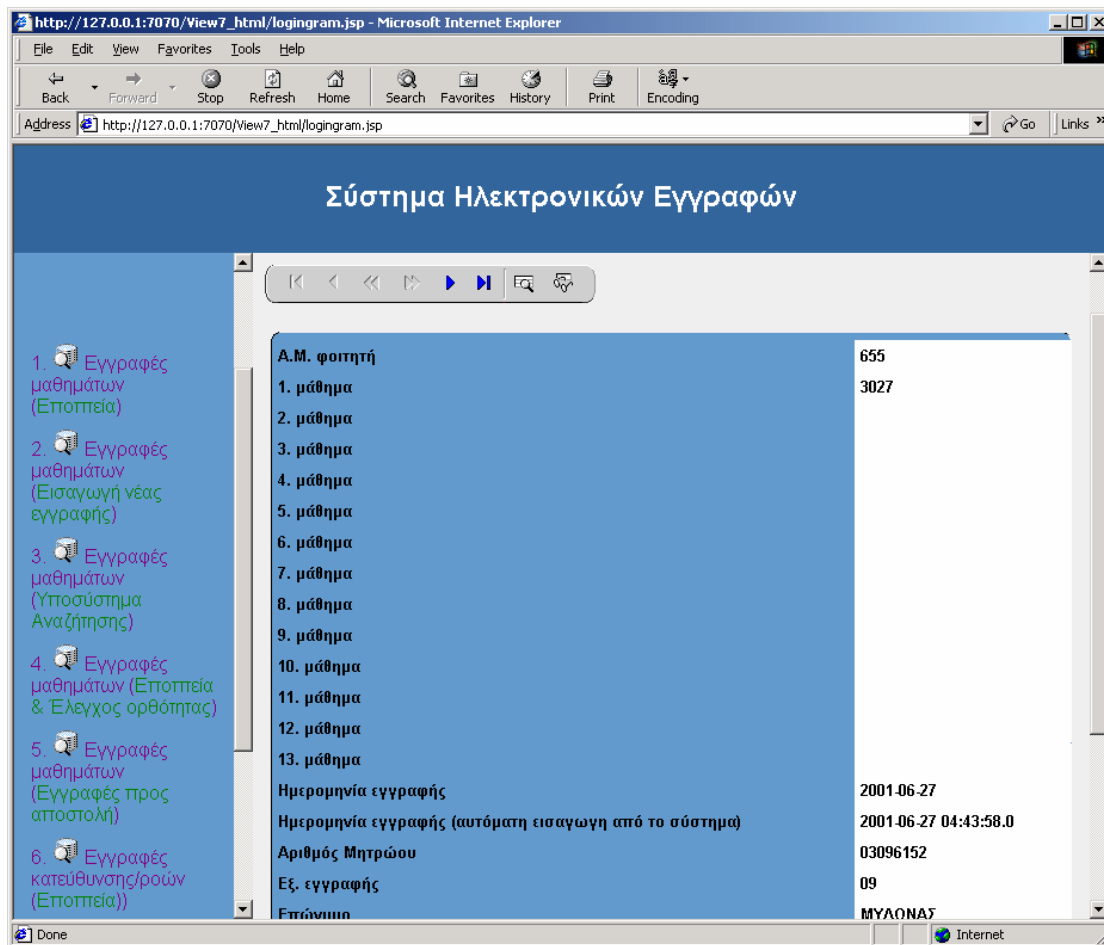
1. Εγγραφές μαθημάτων (Εποπτεία)  
2. Εγγραφές μαθημάτων (Εισαγωγή νέας εγγραφής)  
3. Εγγραφές μαθημάτων (Υποσύστημα Αναζήτησης)  
4. Εγγραφές μαθημάτων (Εποπτεία & Έλεγχος ορθότητας)  
5. Εγγραφές μαθημάτων (Εγγραφές προς αποστολή)  
6. Εγγραφές κατεύθυνσης/ρoών (Εποπτεία)

Εικόνα 7.4.5.α

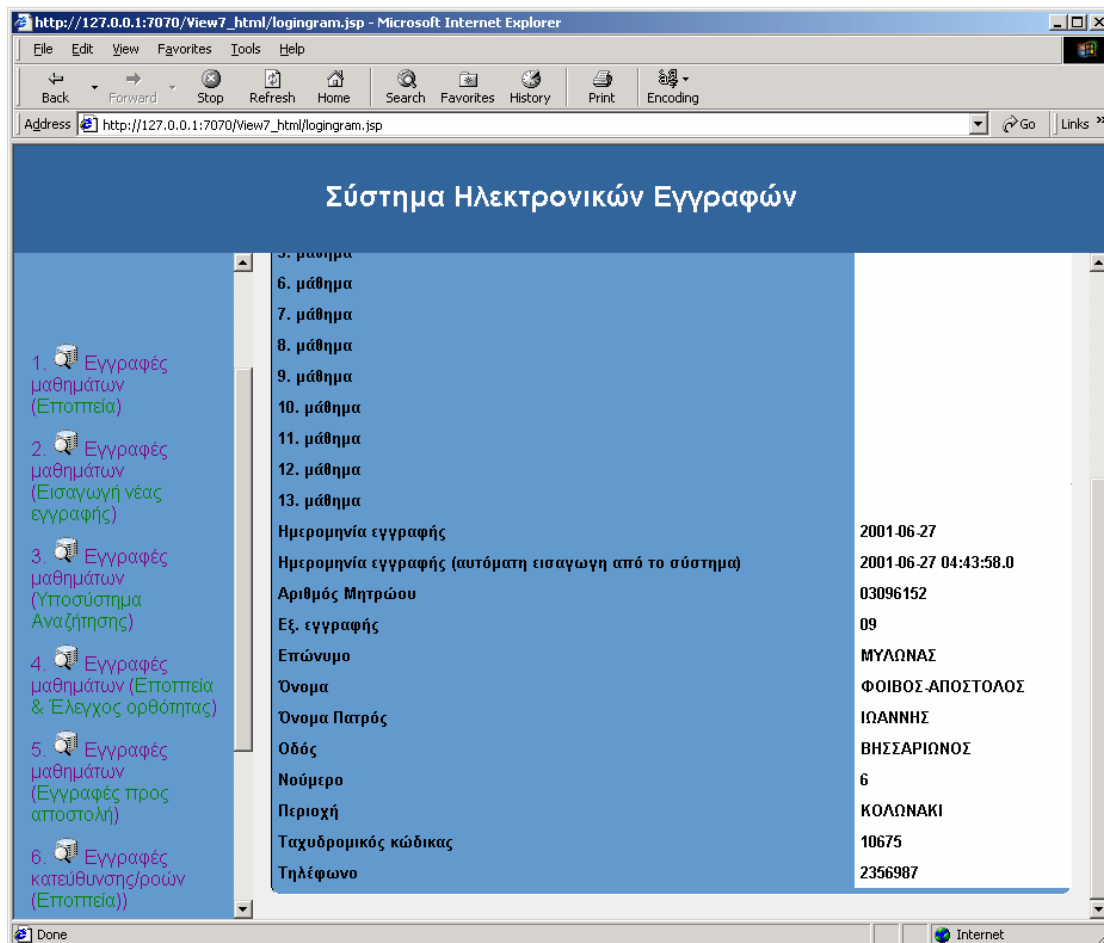




Εικόνα 7.4.5.β



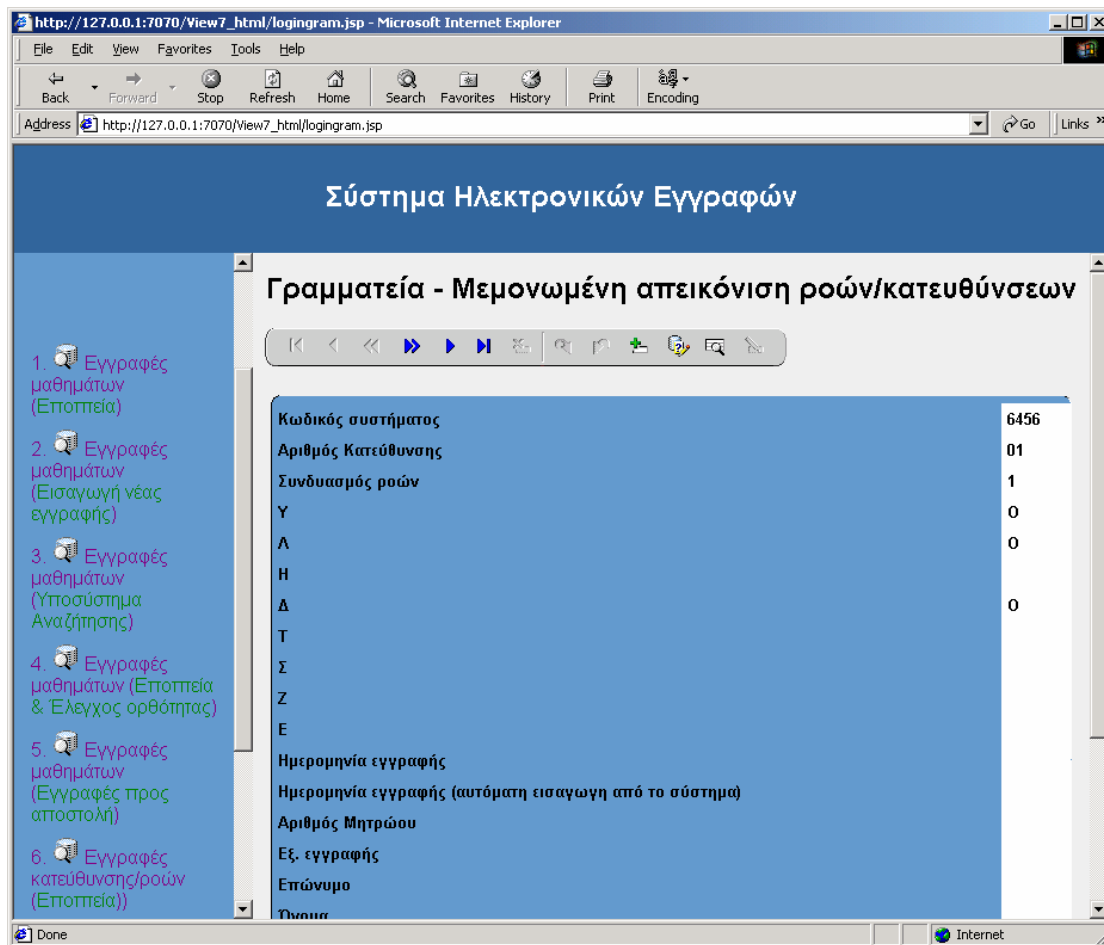
Εικόνα 7.4.5.γ



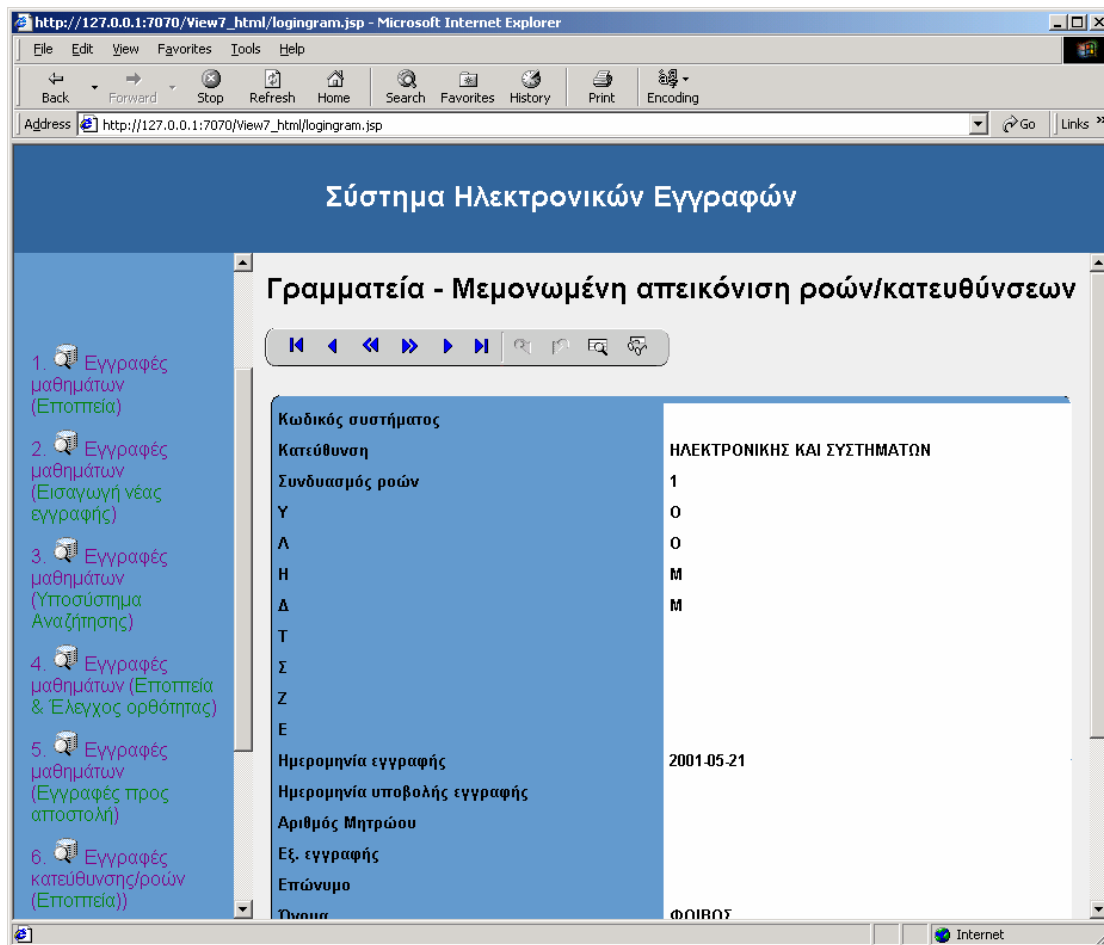
Εικόνα 7.4.5.δ

#### 7.4.6 Εγγραφές κατεύθυνσης/ροών - Εποπτεία:

Ανάλογη με προηγουμένως είναι και η φιλοσοφία των τελευταίων 3 links, το πρώτο από τα οποία οδηγεί στην read-only απεικόνιση των εισαγωγών των φοιτητών όσον αφορά την κατεύθυνση και τις ροές που επιλέγουν στο έκτο εξάμηνο. Και εδώ, δύο δυνατότητες απεικόνισης υπάρχουν (μεμονωμένη και/ή συνολική), ενώ πάλι υπάρχουν 3 διαφορετικά modes: μόνο με κωδικούς, μόνο με τίτλους και με ταυτόχρονη προβολή κωδικών και τίτλων.



Εικόνα 7.4.6.α



Εικόνα 7.4.6.β

http://127.0.0.1:7070/View7\_html/loggingram.jsp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Print Encoding

Address http://127.0.0.1:7070/View7\_html/loggingram.jsp Go Links >>

## Σύστημα Ηλεκτρονικών Εγγραφών

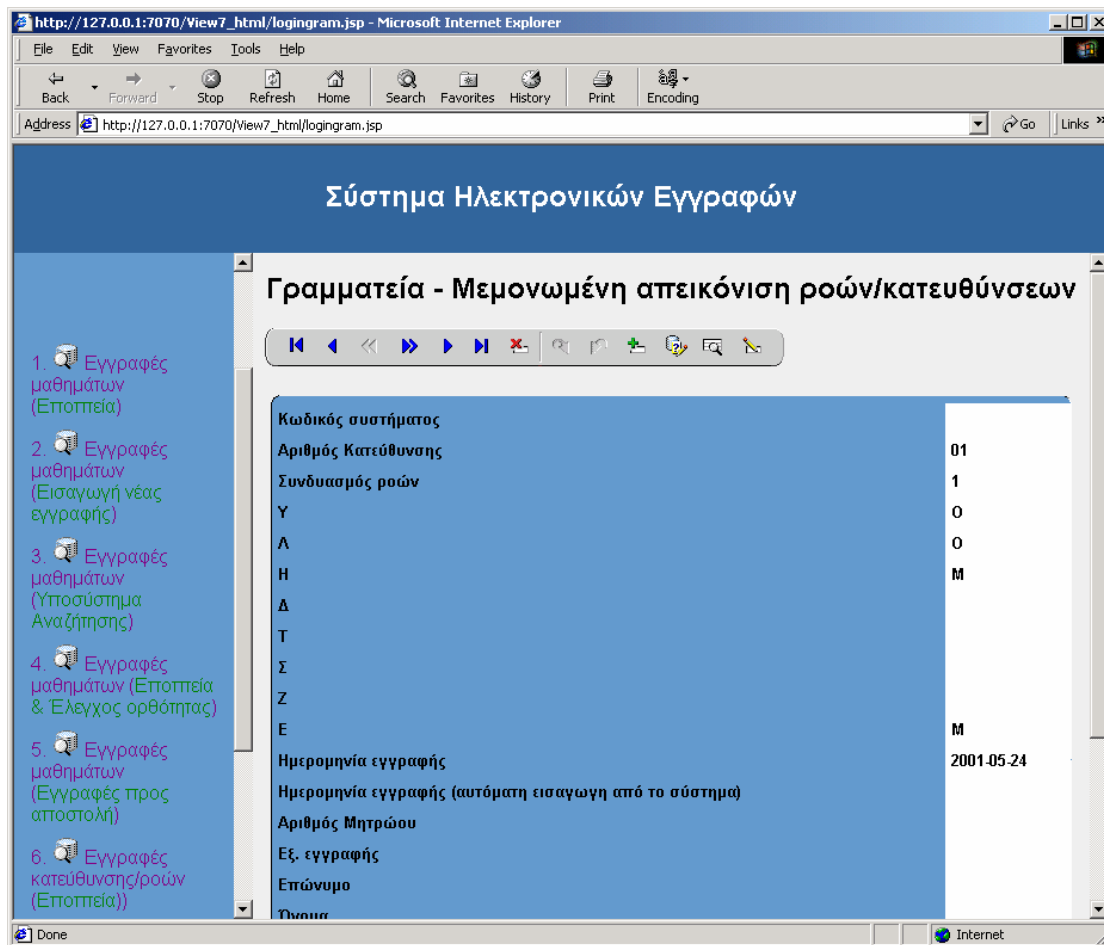
### Γραμματεία - Πλοήγηση ροών/κατευθύνσεων

1. Εγγραφές μαθημάτων (Εποπτεία)  
2. Εγγραφές μαθημάτων (Εισαγωγή νέας εγγραφής)  
3. Εγγραφές μαθημάτων (Υποσύστημα Αναζήτησης)  
4. Εγγραφές μαθημάτων (Εποπτεία & Έλεγχος ορθότητας)  
5. Εγγραφές μαθημάτων (Εγγραφές προς αποστολή)  
6. Εγγραφές κατεύθυνσης/ροών (Εποπτεία))

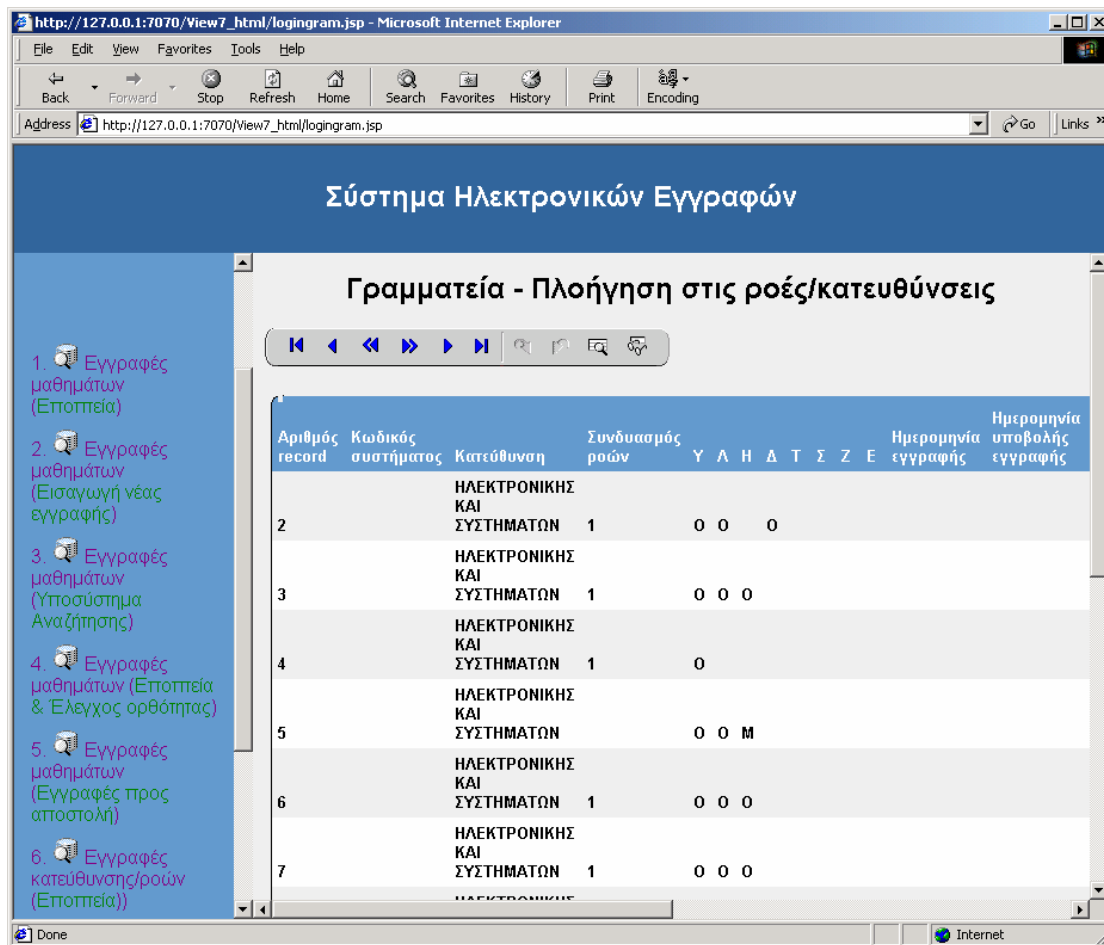
Αριθμός record	Κωδικός συστήματος	Αριθμός Κατεύθυνσης	Συνδυασμός ροών	Υ	Λ	Η	Δ	Τ	Σ	Ζ	Ε	Ημερομηνία εγγραφής	Ημερομηνία εγγραφής	Αρ
1	6456	01	1	0	0	0								
2		01	1	0	0	0								
3		01	1	0	0	0								
4		01	1	0										
5		01		0	0	M								
6		01	1	0	0	0								
7		01	1	0	0	0								
8		01	1	0	0	M						2001-05-28		
9		01	1	0	0	M				M		2001-05-24		
10		01	1	0	M	O		M				2001-05-31		
11		01	1	0	0	M	M					2001-05-21		
12		01	1	0	0	O	M	M	O	M		2001-05-24		
13		01	1	0	0	O	O		M			2001-05-31		
14		01	1	0	0	O	M	O				2001-05-22		
15		01	1	0	0	O	M	M	M	M		2001-05-28		

Done Internet

Εικόνα 7.4.6.γ

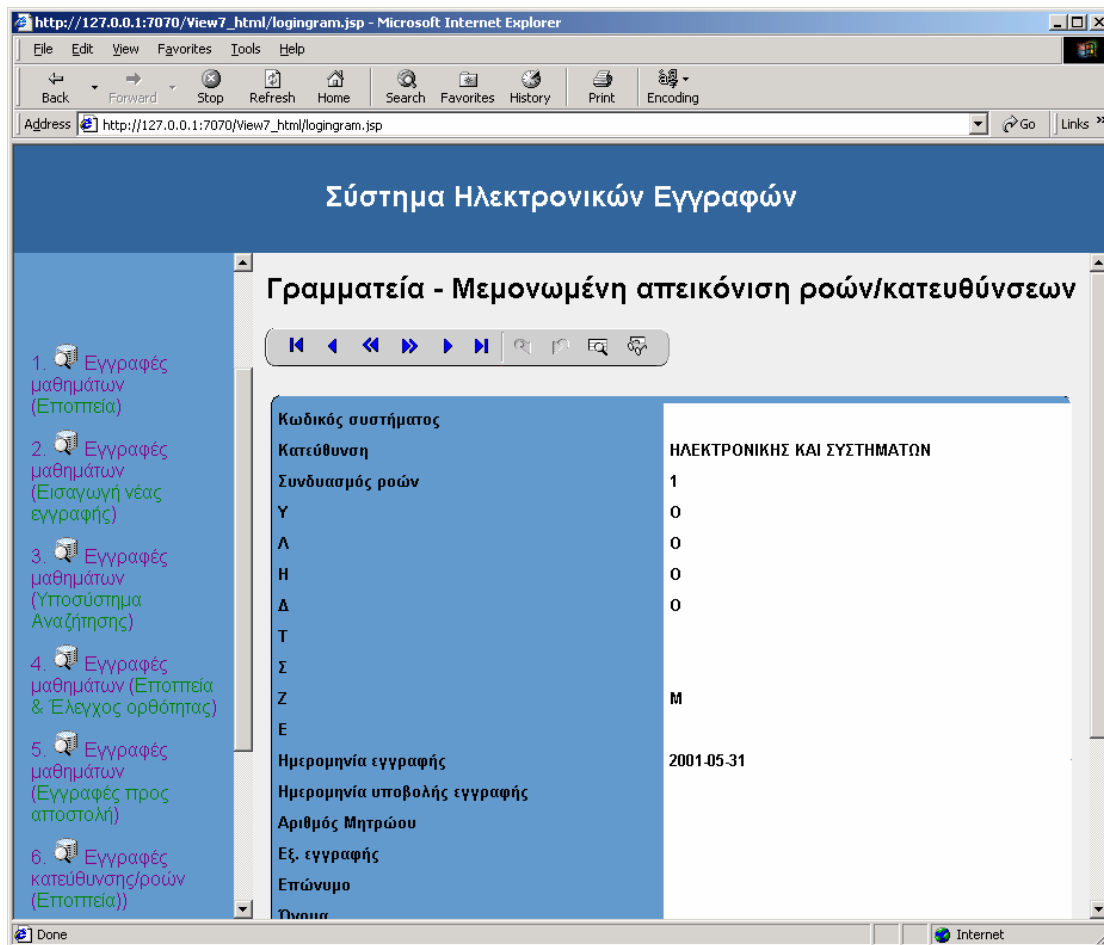


Εικόνα 7.4.6.δ



Εικόνα 7.4.6.ε





Εικόνα 7.4.6.στ

http://127.0.0.1:7070/View7\_html/loggingram.jsp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Print Encoding

Address http://127.0.0.1:7070/View7\_html/loggingram.jsp Go Links >>

## Σύστημα Ηλεκτρονικών Εγγραφών

### Γραμματεία - Πλοήγηση στις ροές/κατευθύνσεις (όλα σε ένα)

1. Εγγραφές μαθημάτων (Επτοπεία)

2. Εγγραφές μαθημάτων (Εισαγωγή νέας εγγραφής)

3. Εγγραφές μαθημάτων (Υποσύστημα Αναζήτησης)

4. Εγγραφές μαθημάτων (Επτοπεία & Έλεγχος ορθότητας)

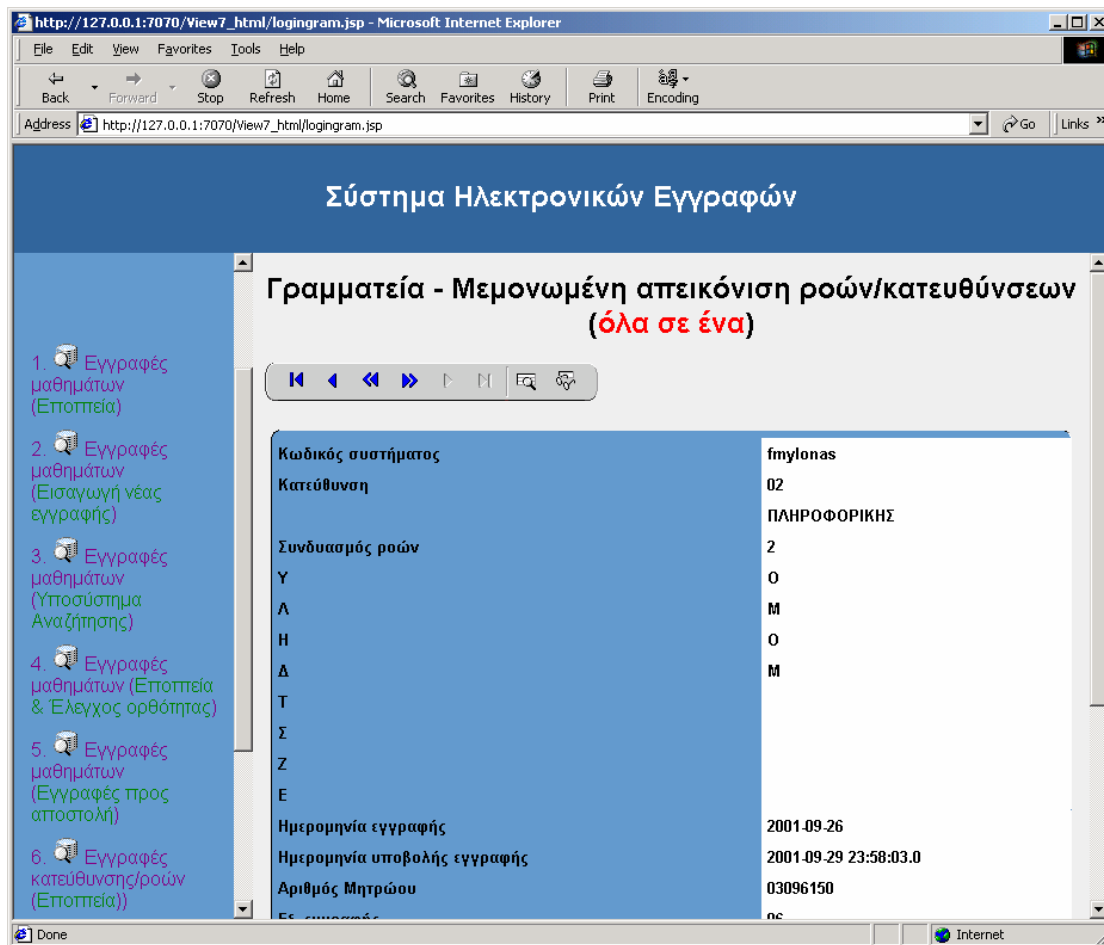
5. Εγγραφές μαθημάτων (Εγγραφές προς αποστολή)

6. Εγγραφές κατεύθυνσης/ροών (Επτοπεία)

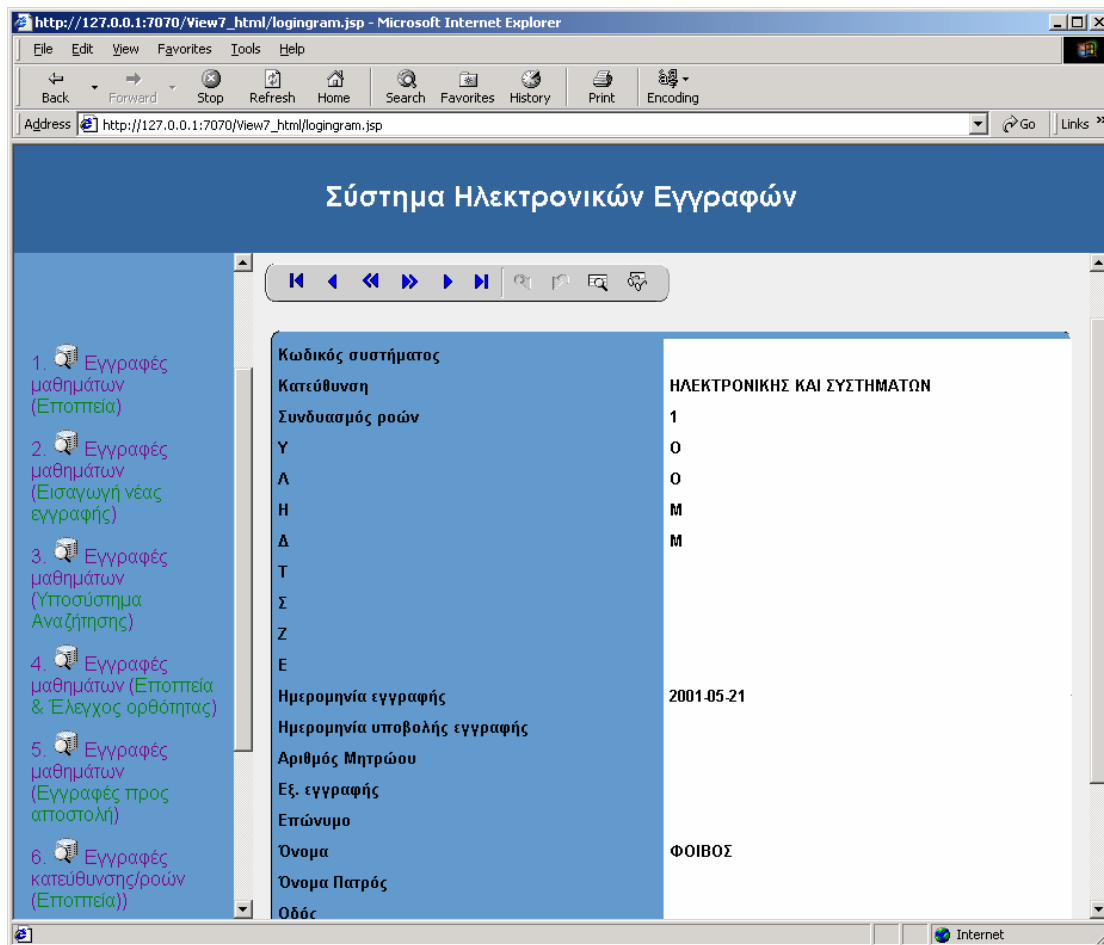
Αριθμός record	Κωδικός συστήματος	Κατεύθυνση	Συνδυασμός ροών	Υ	Λ	Η	Δ	Τ	Σ	Ζ	Ε	Ημερομηνία εγγραφής
1	6456	01	ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ	1	0	0	0					
2		01	ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ	1	0	0	0					
3		01	ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ	1	0	0	0					
4		01	ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ	1	0							
5		01	ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ		0	0	0	0				
			ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ									

Done Internet

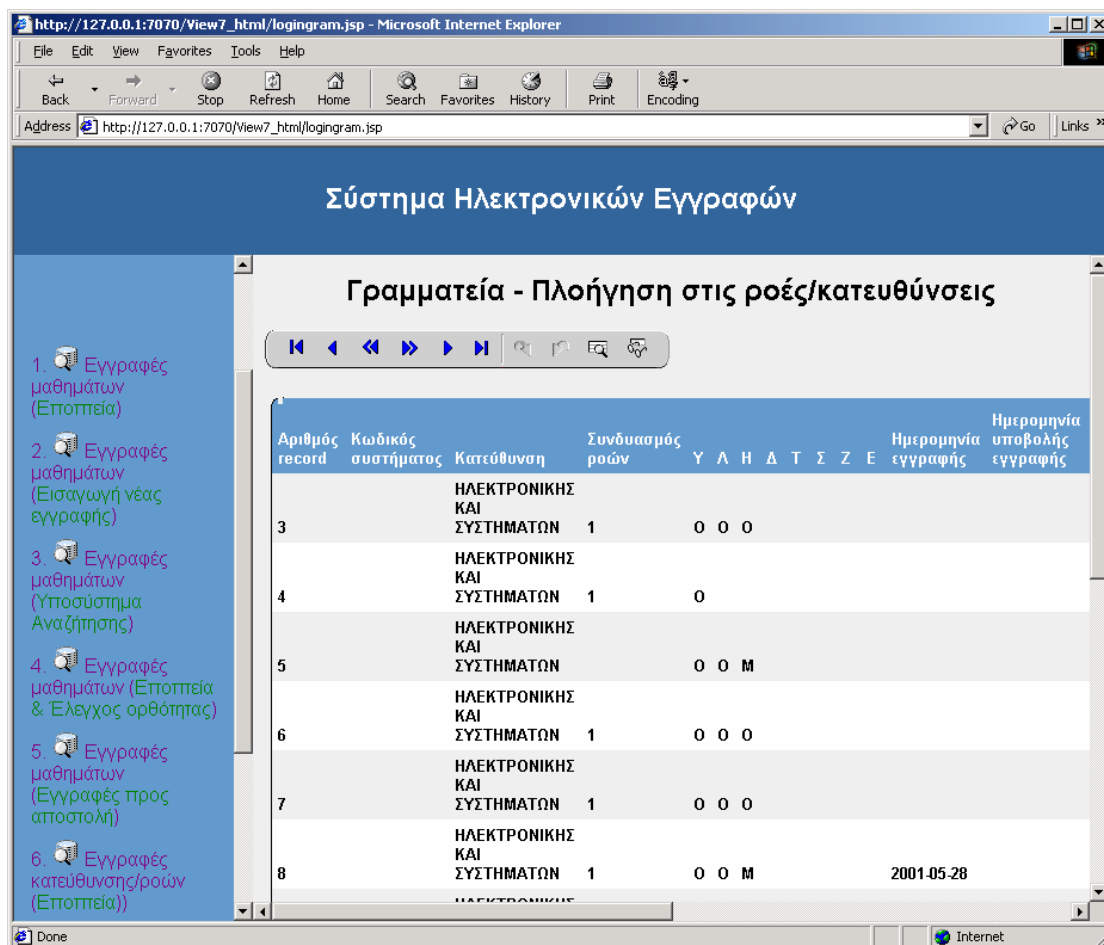
Εικόνα 7.4.6.η



Εικόνα 7.4.6.θ



Εικόνα 7.4.6.1



Εικόνα 7.4.6.κ

http://127.0.0.1:7070/View7\_html/logingram.jsp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Print Encoding

Address http://127.0.0.1:7070/View7\_html/logingram.jsp Go Links >>

## Σύστημα Ηλεκτρονικών Εγγραφών

### Γραμματεία - Πλοήγηση στις ροές/κατευθύνσεις (όλα σε ένα)

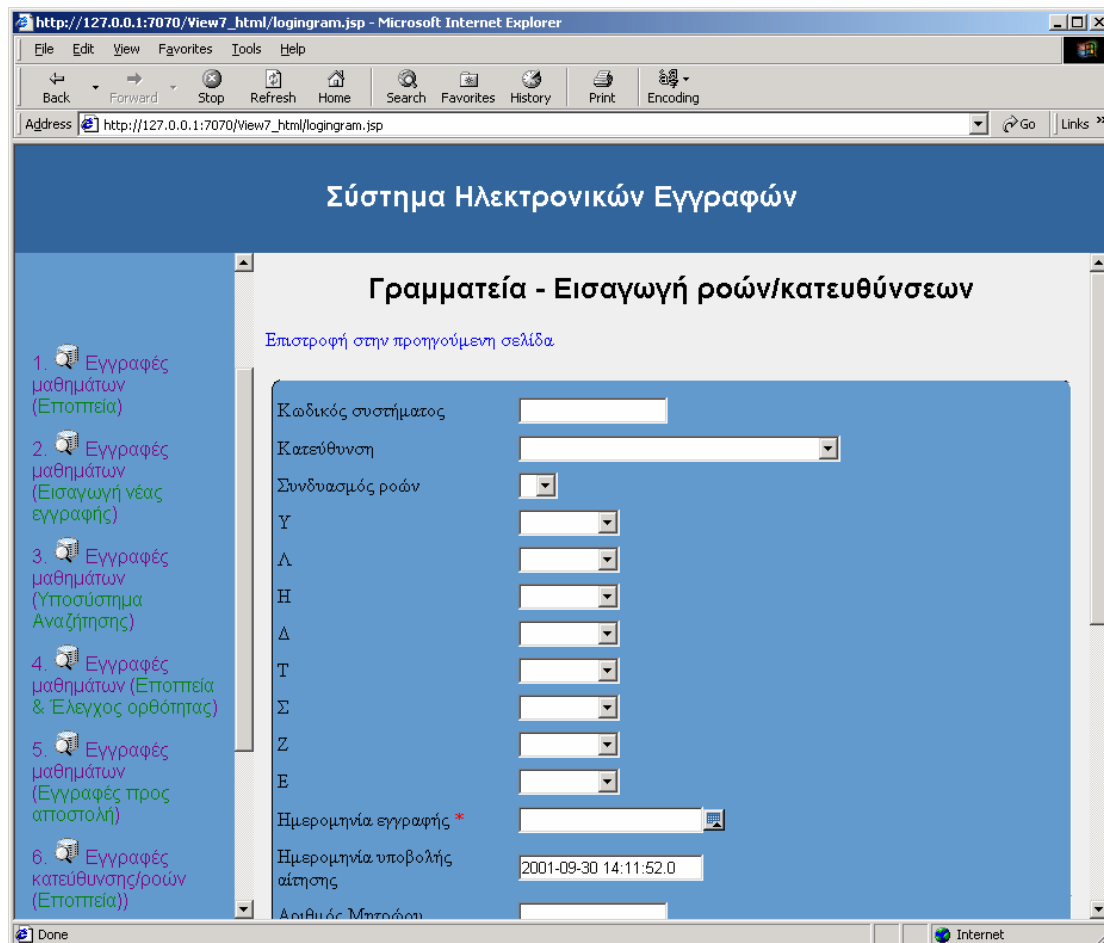
1. Εγγραφές μαθημάτων (Εποπτεία)
2. Εγγραφές μαθημάτων (Εισαγωγή νέας εγγραφής)
3. Εγγραφές μαθημάτων (Υποσύστημα Αναζήτησης)
4. Εγγραφές μαθημάτων (Εποπτεία & Έλεγχος ορθότητας)
5. Εγγραφές μαθημάτων (Εγγραφές προς αποστολή)
6. Εγγραφές κατεύθυνσης/ροών (Εποπτεία)

Αριθμός record	Κωδικός συστήματος	Κατεύθυνση	Συνδυασμός ροών	Υ	Λ	Η	Δ	Τ	Σ	Ζ	Ε	Ημερομηνία εγγραφής
1	6456	01	ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ	1	0	0	0					
2		01	ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ	1	0	0	0					
3		01	ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ	1	0	0	0					
4		01	ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ	1	0							
5		01	ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ		0	0	0	M				
			ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ									

Εικόνα 7.4.6.λ

#### 7.4.7 Εγγραφές κατεύθυνσης/ροών - Εισαγωγή νέας εγγραφής:

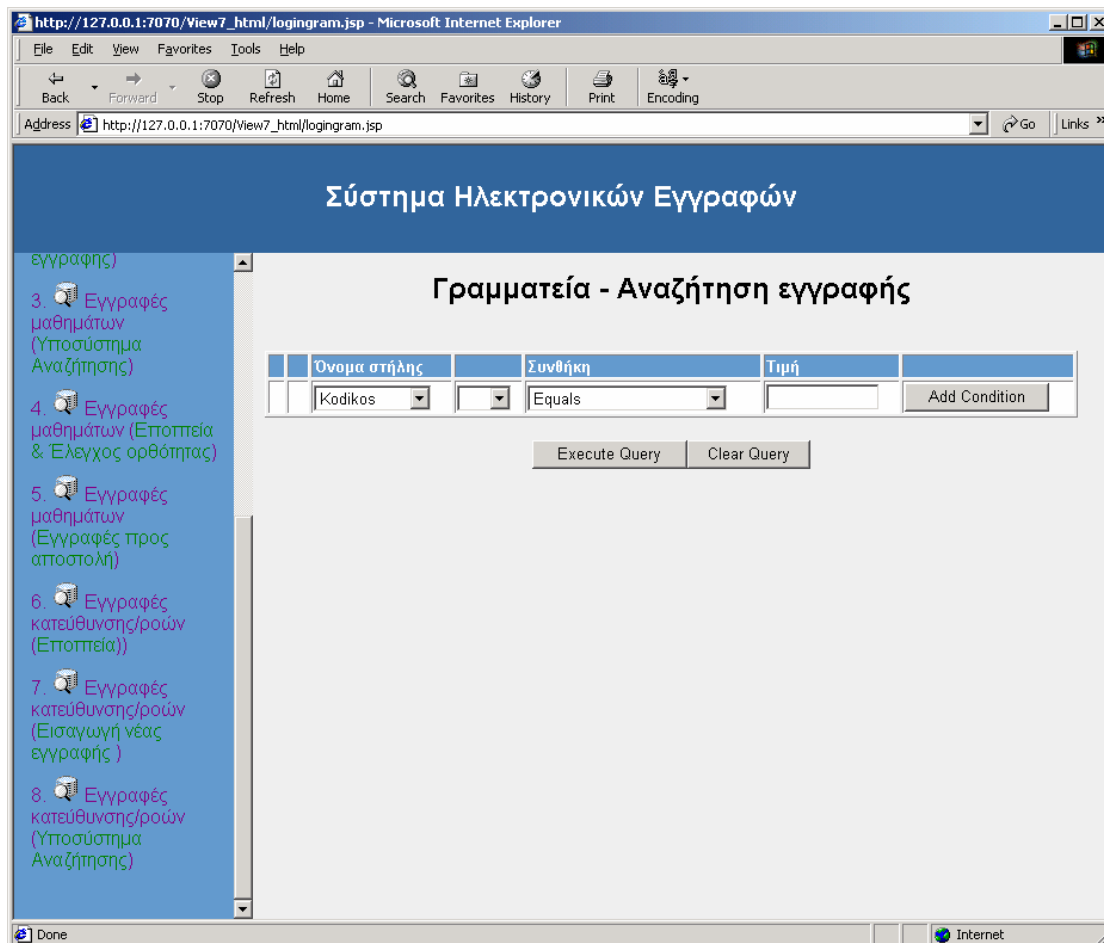
Η δυνατότητα εισαγωγής εντελώς νέας εγγραφής στον πίνακα κατευθύνσεων ροών ακολουθεί στο επόμενο link:



Εικόνα 7.4.7.α

#### 7.4.8 Εγγραφές κατεύθυνσης/ροών – Υποσύστημα Αναζήτησης:

Τέλος, έχουμε το υποσύστημα αναζήτησης του πίνακα κατευθύνσεων/ροών, με παρόμοια χαρακτηριστικά με το αντίστοιχο υποσύστημα για τον πίνακα εγγραφών μαθημάτων. Υπάρχουν 2 είδη αναζήτησης, SQL based και με βάση συγκεκριμένο κριτήριο, όπως ο *Α.Μ.* και ο *Κωδικός συστήματος*, ενώ συνοδευτικά υπάρχουν η δυνατότητα άμεσης διόρθωσης μετά την εποπτεία ενός record και η ταυτόχρονη «on the fly» διόρθωση κατά την πλοήγηση.



Εικόνα 7.4.8.α

## Αναζήτηση με βάση συγκεκριμένο κριτήριο:

Αναζήτηση εγγραφής με βάση τον **A.M.** του φοιτητή:

Εισάγετε τα στοιχεία και κατόπιν πατήστε το πλήκτρο **enter**

Εικόνα 7.4.8.β

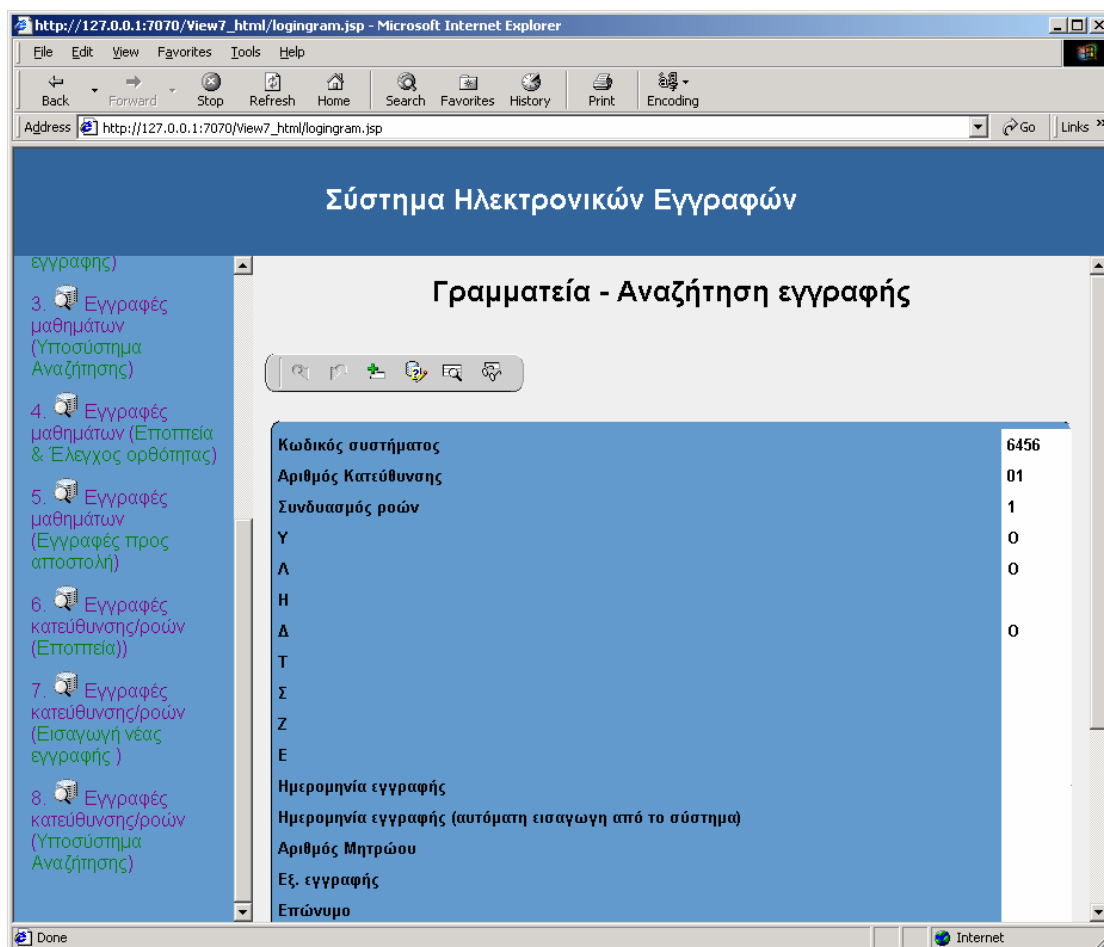
## Αναζήτηση με βάση συγκεκριμένο κριτήριο:

Αναζήτηση εγγραφής με βάση τον **Κωδικό Συστήματος H.E.** του φοιτητή:

Εισάγετε τα στοιχεία και κατόπιν πατήστε το πλήκτρο **enter**

Εικόνα 7.4.8.γ

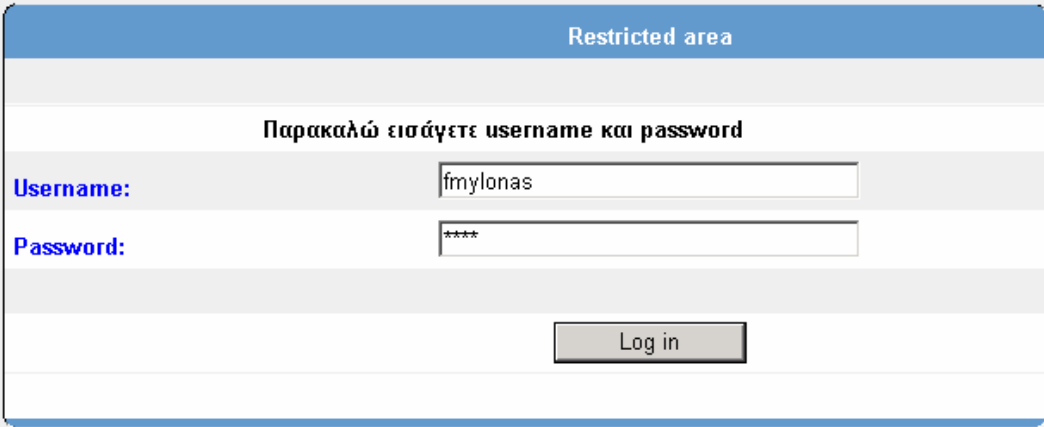




Εικόνα 7.4.8.δ

## 7.5 Υποσύστημα administrator::

Το τελευταίο άμεσα υλοποιημένο μέρος του Συστήματος Ηλεκτρονικών Εγγραφών είναι το τμήμα του Administrator του όλου συστήματος, ο οποίος δεν έχει κάποιο ρόλο στη μέχρι τώρα διαδικασία, όπως την περιγράψαμε παραπάνω, αλλά ενδιαφέρεται μόνο για το τεχνικό κομμάτι του και έχει access σε αυτό και μόνο. Αρχικά, μετά το παρακάτω login page (Εικόνα 7.5.α), που γίνεται πιστοποίησή του στη βάση, μεταφέρεται σε μια σελίδα με τελείως διαφορετικό format από τις προηγούμενες, προκειμένου και να τονιστεί η ιδιαιτερότητα του χρήστη αυτού από τους υπόλοιπους που διαθέτουν πρόσβαση στο σύστημα.



Restricted area

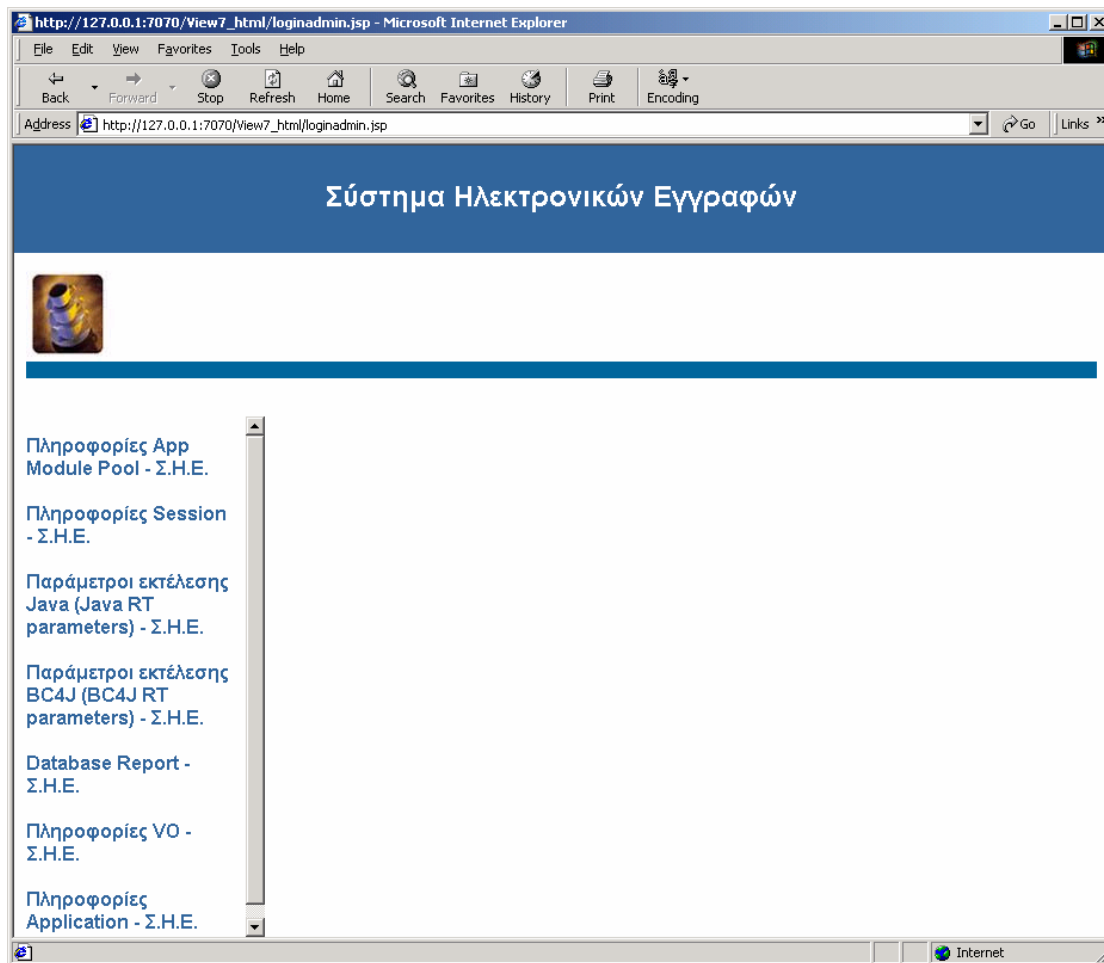
Παρακαλώ εισάγετε username και password

Username: fmylonas

Password: \*\*\*\*

Log in

Εικόνα 7.5.α



Εικόνα 7.5.β

Στο αριστερό frame υπάρχει το κλασικό προσφιλές μενού περιεχομένων:

Πληροφορίες App  
Module Pool - Σ.Η.Ε.

Πληροφορίες Session  
- Σ.Η.Ε.

Παράμετροι εκτέλεσης  
Java (Java RT  
parameters) - Σ.Η.Ε.

Παράμετροι εκτέλεσης  
BC4J (BC4J RT  
parameters) - Σ.Η.Ε.

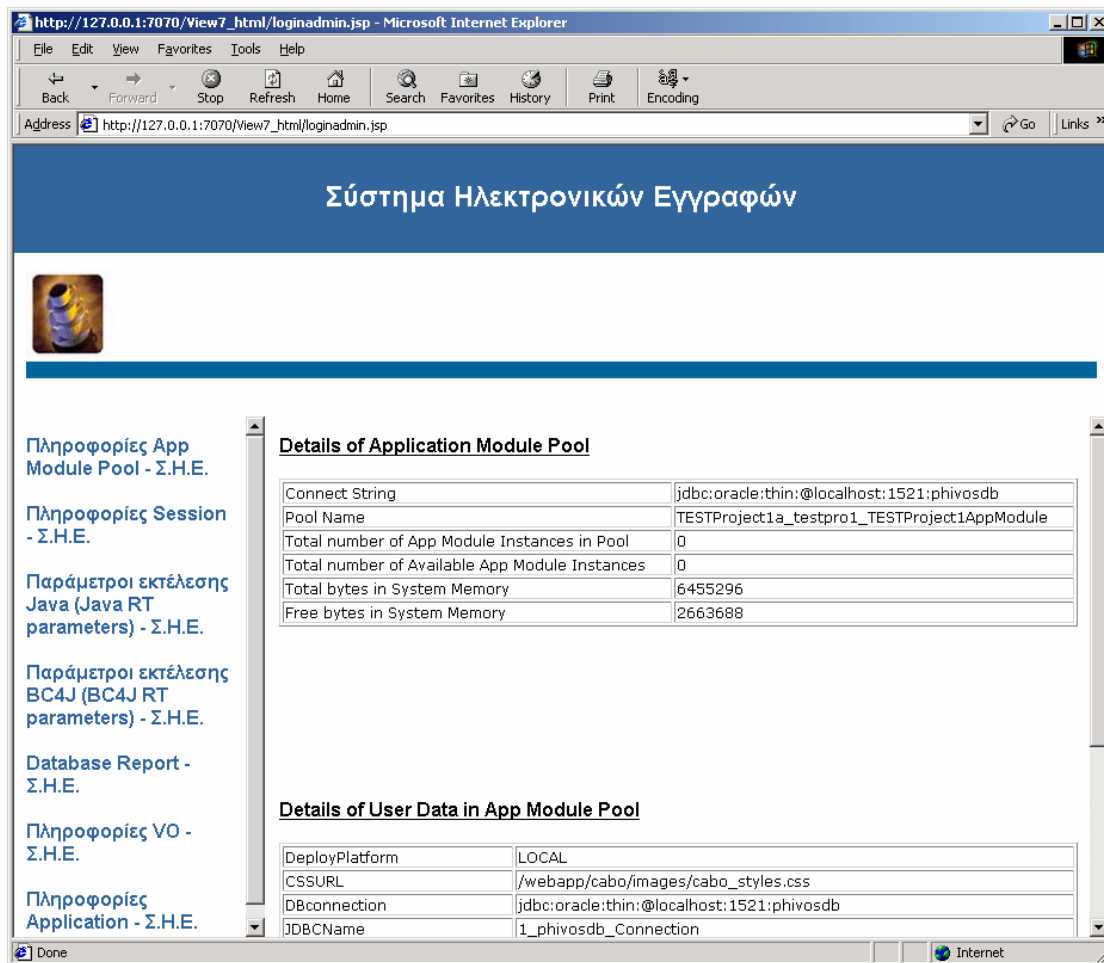
Database Report -  
Σ.Η.Ε.

Πληροφορίες VO -  
Σ.Η.Ε.

Πληροφορίες  
Application - Σ.Η.Ε.

*Εικόνα 7.5.γ*

, το οποίο ανοίγει στο δεξιό frame έναν αριθμό από σελίδες, αρχικά πληροφοριακού και μόνο περιεχομένου για τον administrator (Εικόνες 7.5.δ – 7.5.ι).



Εικόνα 7.5.δ


http://127.0.0.1:7070/View7\_html/loginadmin.jsp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Print Encoding

Address http://127.0.0.1:7070/View7\_html/loginadmin.jsp Go Links >>

## Σύστημα Ηλεκτρονικών Εγγραφών



**Πληροφορίες App  
Module Pool - Σ.Η.Ε.**

**Πληροφορίες Session  
- Σ.Η.Ε.**

**Παράμετροι εκτέλεσης  
Java (Java RT  
parameters) - Σ.Η.Ε.**

**Παράμετροι εκτέλεσης  
BC4J (BC4J RT  
parameters) - Σ.Η.Ε.**

**Database Report -  
Σ.Η.Ε.**

**Πληροφορίες VO -  
Σ.Η.Ε.**

**Πληροφορίες  
Application - Σ.Η.Ε.**

### Details of HTTP Session

Session Creation Time	Sun Sep 30 14:26:54 GMT+03:00 2001
Session Last Accessed Time	Sun Sep 30 14:49:02 GMT+03:00 2001
Session Maximum Inactive level	28800
Is Session is new ?	false

### Details of HTTP Session Object Table

Object Name	Object Value
admin	cjm2forc
adminbean	View7_html.login.adminbean@7269
cssurl	/webapp/cabo/images/cabo_styles.css
imagebase	/webapp/jsimages
oracle.jsp.jspsessionmgr	oracle.jsp.app.JspApplication@6792
oracle_ord_im_orclaudiodomain_editrender	oracle.ord.html.FileUploadField
oracle_ord_im_orclaudiodomain_render	oracle.ord.html.OrdBuildURL
oracle_ord_im_ordimagedomain_editrender	oracle.ord.html.FileUploadField
oracle_ord_im_ordimagedomain_render	oracle.ord.html.OrdBuildURL

Done Internet

Εικόνα 7.5.ε


http://127.0.0.1:7070/View7\_html/loginadmin.jsp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Print Encoding

Address http://127.0.0.1:7070/View7\_html/loginadmin.jsp Go Links >>

## Σύστημα Ηλεκτρονικών Εγγραφών



[Πληροφορίες App Module Pool - Σ.Η.Ε.](#)

[Πληροφορίες Session - Σ.Η.Ε.](#)

[Παράμετροι εκτέλεσης Java \(Java RT parameters\) - Σ.Η.Ε.](#)

[Παράμετροι εκτέλεσης BC4J \(BC4J RT parameters\) - Σ.Η.Ε.](#)

[Database Report - Σ.Η.Ε.](#)

[Πληροφορίες VO - Σ.Η.Ε.](#)

[Πληροφορίες Application - Σ.Η.Ε.](#)

### Details of Java Run Time Parameters

Name	Value
java.specification.name	Java Platform API Specification
awt.toolkit	sun.awt.windows.WToolkit
java.version	1.2.2-JDeveloper
java.awt.graphicsenv	sun.awt.Win32GraphicsEnvironment
user.timezone	Europe/Athens
java.specification.version	1.2
java.vm.vendor	Oracle Corp.
java.vm.specification.version	1.0
user.home	D:\Documents and Settings\Administrator
os.arch	x86
java.awt.fonts	
java.vendor.url	http://java.sun.com/
file.encoding.pkg	sun.io
user.region	GR
java.vm.vendor.url	http://www.oracle.com/
java.home	D:\Program Files\Oracle\JDeveloper 3.2\java1.2\jre
line.separator	

Done Internet

Εικόνα 7.5.στ


http://127.0.0.1:7070/View7\_html/loginadmin.jsp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Print Encoding

Address http://127.0.0.1:7070/View7\_html/loginadmin.jsp Go Links >>

## Σύστημα Ηλεκτρονικών Εγγραφών



Πληροφορίες App  
Module Pool - Σ.Η.Ε.

Πληροφορίες Session  
- Σ.Η.Ε.

Παράμετροι εκτέλεσης  
Java (Java RT  
parameters) - Σ.Η.Ε.

Παράμετροι εκτέλεσης  
BC4J (BC4J RT  
parameters) - Σ.Η.Ε.

Database Report -  
Σ.Η.Ε.

Πληροφορίες VO -  
Σ.Η.Ε.

Πληροφορίες  
Application - Σ.Η.Ε.

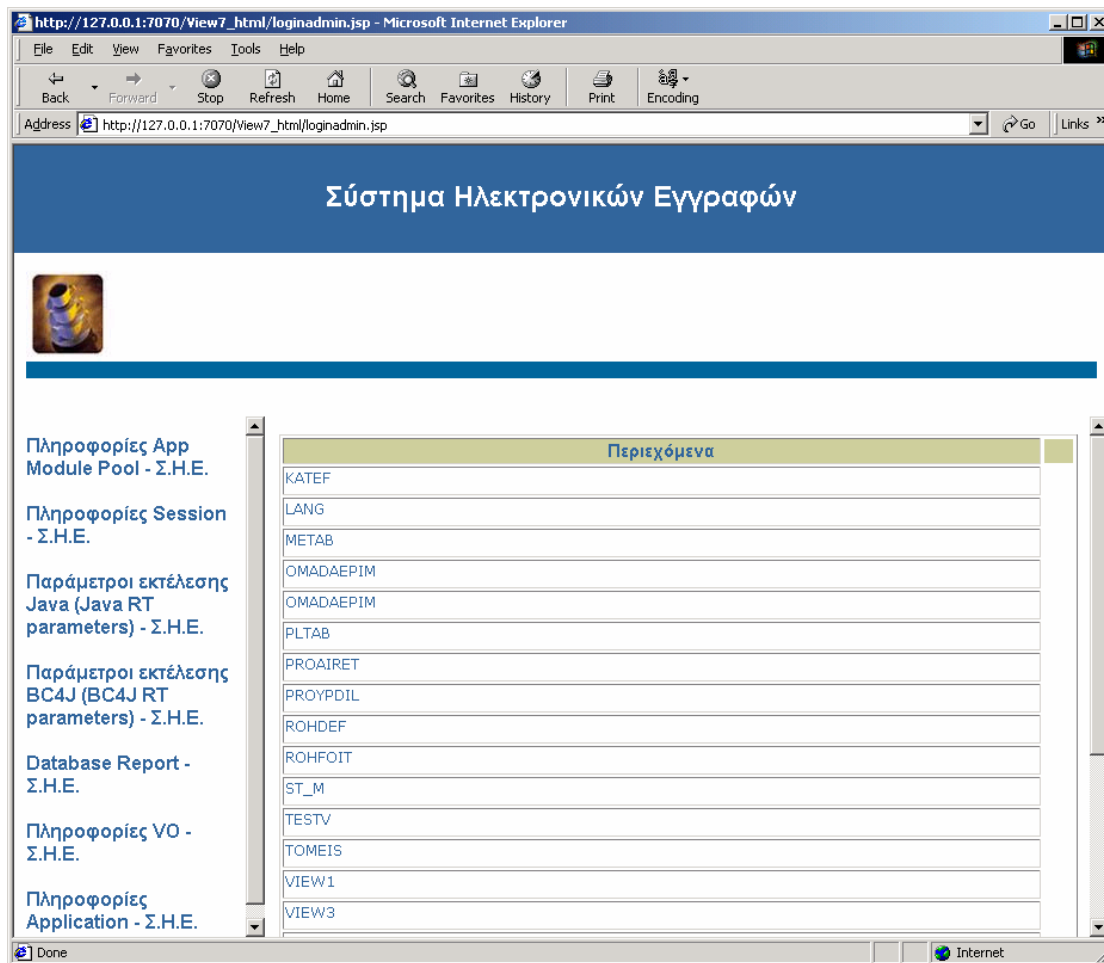
### Details of BC4J Run Time Parameters

Name	Value
jbo.default.language	['en'] - The default BC4J session language - part of the Locale
jbo.default.country	['US'] - The default BC4J session country - part of the Locale
DeployPlatform	['LOCAL'] - Deployment Platform
ConnectionMode	[''] - Connection Mode used for Visigenic ORB (1 2 3 4)
HostName	[''] - Name of the host machine
ConnectionPort	[''] - Connection Port
ApplicationPath	[''] - Application Path
java.naming.security.principal	[''] - JNDI: Identity of principal (e.g. user) for the authentication scheme
java.naming.security.credentials	[''] - JNDI: Principals credentials for the authentication scheme
jbo.use.pers.coll	['false'] - Use persistent collection for view row spill-over?
jbo.pers.max.rows.per.node	['70'] - Maximum number of rows in a persistent collection node
jbo.pers.max.active.nodes	['10'] - Maximum number of active (in-memory) nodes in a persistent collection
jbo.pcoll.mgr	['oracle.jbo.pcoll.OraclePersistManager'] - Persistent collection manager
jbo.fetch.mode	['AS.NEEDED'] - Control fetch behavior of View Objects (AS.NEEDED ALL)
JBODynamicObjectsPackage	[''] - Name of package to be loaded dynamically (no default)
MetaObjectContextFactory	['oracle.jbo.server.xml.DefaultMOMContextFactory'] -

Done Internet

Εικόνα 7.5.η





Εικόνα 7.5.θ


http://127.0.0.1:7070/View7\_html/loginadmin.jsp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Print Encoding

Address http://127.0.0.1:7070/View7\_html/loginadmin.jsp Go Links >>

## Σύστημα Ηλεκτρονικών Εγγραφών



Πληροφορίες App  
Module Pool - Σ.Η.Ε.

Πληροφορίες Session  
- Σ.Η.Ε.

Παράμετροι εκτέλεσης  
Java (Java RT  
parameters) - Σ.Η.Ε.

Παράμετροι εκτέλεσης  
BC4J (BC4J RT  
parameters) - Σ.Η.Ε.

Database Report -  
Σ.Η.Ε.

Πληροφορίες VO -  
Σ.Η.Ε.

Πληροφορίες  
Application - Σ.Η.Ε.

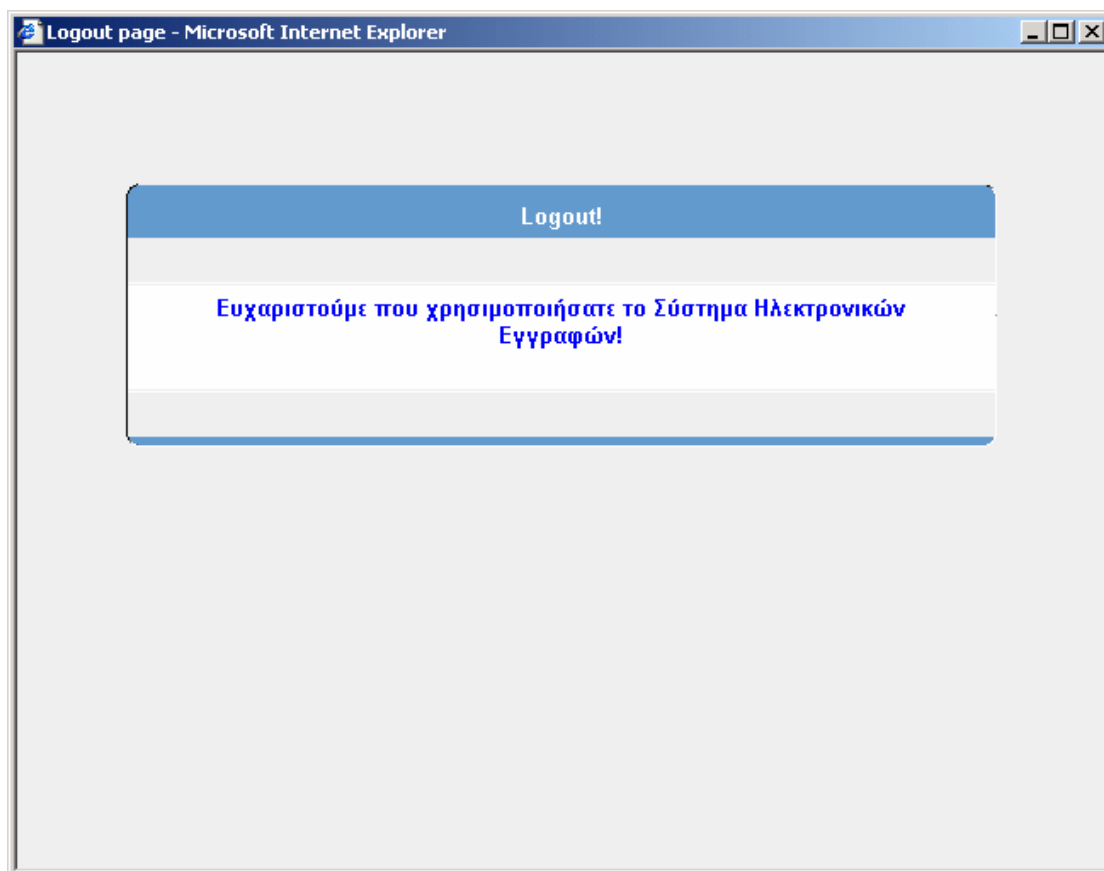
KODIKOS			VARCHAR2	10	0	No	No			
KOD1			NUMBER	4	0	No	Yes			
KOD2			NUMBER	4	0	No	Yes			
KOD3			NUMBER	4	0	No	Yes			
KOD4			NUMBER	4	0	No	Yes			
KOD5			NUMBER	4	0	No	Yes			
KOD6			NUMBER	4	0	No	Yes			
KOD7			NUMBER	4	0	No	Yes			
KOD8			NUMBER	4	0	No	Yes			
KOD9			NUMBER	4	0	No	Yes			
KOD10			NUMBER	4	0	No	Yes			
KOD11			NUMBER	4	0	No	Yes			
KOD12			NUMBER	4	0	No	Yes			
KOD13			NUMBER	4	0	No	Yes			
HMER			DATE	7	0	No	Yes			
CURRDATE			DATE	7	0	No	Yes			
AM			VARCHAR2	10	0	No	Yes			
EKS, EGGR			VARCHAR2	2	0	No	Yes			

Internet

Εικόνα 7.5.1

## 7.6 Έξοδος από το Σύστημα – *logout page*

Η έξοδος από το Σύστημα Ηλεκτρονικών Εγγραφών πραγματοποιείται πάντα μέσω της σελίδας logout, στην οποία και μεταφέρεται αυτόματα ο χρήστης μόλις κλείσει το παράθυρο του browser του ή μόλις επισκεφθεί μία σελίδα στο Internet έξω από το ΣΗΕ. Στη σελίδα αυτή πραγματοποιούνται διάφοροι έλεγχοι και καθάρισμα των παραμέτρων identification του χρήστη, όπως είδαμε στο κεφάλαιο 2.14.6), οπότε και είναι απολύτως απαραίτητη σε περιβάλλον πολλαπλών χρηστών. Πέρα από τα παραπάνω, και με γνώση ότι πρόκειται για ένα fully web based σύστημα, δεν είναι ορατή ούτε απαιτείται κάποια άλλη διαδικασία από το χρήστη ή το σύστημα για την έξοδο, ενισχύοντας έτσι την απλότητα, την απόδοση και την λειτουργικότητά του.



Εικόνα 7.6.α



# 8

## *Απαιτήσεις Συστήματος*

### *8.1 Απαιτήσεις client-side:*

Οι απαιτήσεις του Συστήματος Ηλεκτρονικών Εγγραφών όσον αφορά την πλευρά του οποιοδήποτε client, πέρα από τις αυτονόμετες της ύπαρξης ηλεκτρονικού υπολογιστή και σύνδεσης/πρόσβασης στο Internet, συνοψίζονται στην ύπαρξη **Javascript enabled web browser** στο σύστημά του. Όλοι οι σύγχρονοι browsers τελευταίας γενιάς υποστηρίζονται, όπως ο Internet Explorer 5.5 και εξής και ο Netscape 6.1 και εξής. Σε περίπτωση πρόσβασης χρήστη στην αρχική σελίδα με browser έκδοσης μικρότερης από τις παραπάνω ή με την επιλογή της Javascript απενεργοποιημένη, εμφανίζεται κατάλληλο μήνυμα, το οποίο τον προτρέπει, είτε να αναβαθμίσει το software του, είτε να ενεργοποιήσει την κατάλληλη επιλογή σε αυτό.

Η χρήση της Javascript είναι απαραίτητη κυρίως για τη διαδικασία ελέγχων των επιλογών/εισαγωγών του χρήστη, αλλά και κατά την ίδια την πλοήγηση στο site, όπως έγινε κατανοητό στο κεφάλαιο 6.6. Σημειώνουμε, δε, ότι παρόλο που η αναγνώριση της ταυτότητας του χρήστη πραγματοποιείται κατά την είσοδό του στο σύστημα και διατηρείται καθόλη τη διάρκεια της πλοήγησής του στο site, δεν απαιτείται η ενεργοποίηση των cookies από τη πλευρά του. Κι αυτό, γιατί χρησιμοποιείται η **session tracking** τεχνική των JSPs, η οποία και πραγματοποιεί μέσω κατάλληλων κλάσεων και κλήσεων σε μεθόδους διατήρηση του session ενός χρήστη, είτε μέσω **cookies**, αν αυτά είναι ενεργοποιημένα, είτε μέσω **url-rewriting**, πέρασμα δηλαδή των χαρακτηριστικών του session στο url κάθε σελίδας, αν τα cookies είναι disabled. Η όλη διαδικασία πραγματοποιείται με πλήρη διαφάνεια στο χρήστη και περιγράφηκε αναλυτικότερα στο κεφάλαιο 2.14.6.

Επιπρόσθετα στα παραπάνω, συγκεκριμένες ομάδες χρηστών θα πρέπει να διαθέτουν πρόσβαση σε εκτυπωτή, είτε άμεσα στο ίδιο το σύστημά τους, είτε έμμεσα στο περιβάλλον τοπικού δικτύου (LAN) τους. Η απαίτηση αυτή έχει να κάνει περισσότερο με την λειτουργικότητα του συστήματος, παρά με συγκεκριμένα requirements από τη πλευρά του χρήστη, καθώς η χρήση εκτυπωτή αποτελεί αναγκαίο μέτρο για την επιτυχή ολοκλήρωση της διαδικασίας εγγραφής σε μαθήματα και κατευθύνσεις/ροές.

## **8.2 Απαιτήσεις server-side:**

Τέλος, αν θέλουμε να συνοψίσουμε σε μερικές γραμμές τις λειτουργικές απαιτήσεις του συστήματος από τη πλευρά του καθενός από τους servers που θα το σηκώσουν, αυτές δεν θα διέφεραν ιδιαίτερα από τα τεχνικά χαρακτηριστικά του μηχανήματος στο οποίο πραγματοποιήθηκε η φάση της ανάπτυξης και περιγράφηκαν στην ενότητα 6.10 και τις εξειδικευμένες ανάγκες λογισμικού που είδαμε στις ενότητες 2.4 και 2.5. Κι αυτό, διότι ουσιαστικά αυτό που απαιτείται για την εύρυθμη λειτουργία του, είναι ένα καταναμημένο περιβάλλον web servers με δυνατότητες DNS. Και ένα τέτοιο περιβάλλον είναι σχετικά εύκολο να στηθεί, στηριζόμενοι στην υπολογιστική ισχύ που περιγράψαμε και χρησιμοποιώντας ένα Unix-οειδές περιβάλλον, όπως το RedHat Linux.

Επίσης, από τη πλευρά της υλικοτεχνικής δικτυακής υποδομής, απαιτείται σαφώς broadband σύνδεση με το Internet, έτσι ώστε το σύστημα να είναι σε θέση να αντεπεξέλθει σε μεγάλο αριθμό requests, και βεβαίως ορθό configuration του για την ένταξή του στο υποδίκτυο του Εθνικού Μετσόβιου Πολυτεχνείου.

# 9

## *Επίλογος*

### *9.1 Μελλοντικές Επεκτάσεις*

Το Σύστημα Ηλεκτρονικών Εγγραφών στην παρούσα έκδοσή του αναπτύχθηκε με κύριο γνώμονα την άμεση, απροβλημάτιστη και εύκολη χρησιμοποίησή του από τους προπτυχιακούς φοιτητές και το προσωπικό της γραμματείας για την διευκόλυνση και απλοποίηση της διαδικασίας εγγραφής των πρώτων στο Τμήμα. Τα βήματα επιλογής μαθημάτων αποτέλεσαν την κατά το δυνατόν ηλεκτρονική υλοποίηση του υπάρχοντος εγγράφου αίτησης εγγραφής και της νοητής συμπλήρωσής του, και συνακόλουθα προσπάθησε να λύσει τα όποια προβλήματα ή δυσλειτουργίες αυτό παρουσιάζει. Ο αλγόριθμος επιλογής κατευθύνσεων και ροών είναι ρητά ορισμένος στον Οδηγό Σπουδών του Τμήματος και ακολουθήθηκε πιστά. Τέλος, στηριζόμενο στα παραπάνω αναπτύχθηκε ένα πλήρες υποσύστημα ελέγχου της διαδικασίας από το προσωπικό της γραμματείας και πιο συγκεκριμένα ένα σύστημα εισαγωγής, μετατροπής και διαγραφής των στοιχείων που είναι καταχωρημένα στη βάση δεδομένων, καθώς και ένα εποπτικό υποσύστημα για τον εκάστοτε administrator του.

Αν κανείς δοκιμάσει να ξεφύγει από την αυστηρή οριοθέτηση του υπάρχοντος εγγράφου αίτησης εγγραφής και θελήσει να παρουσιάσει ένα αυτόνομο σύστημα εγγραφών, θα μπορούσε να υλοποιήσει μια προσωποποιημένη διαδικασία εγγραφής, με βάση τα προσωπικά στοιχεία κάθε φοιτητή, όπως βαθμοί, δικαιώματα/υποχρεώσεις σε μαθήματα, προσωπικό ιστορικό κ.ο.κ., συλλέγοντας και διατηρώντας στη βάση του συστήματος τα απαιτούμενα records. Η όλη μετατροπή του συστήματος δεν απαιτεί ιδιαίτερα κοπιαστική δουλειά, καθώς μεγάλο μέρος μιας τέτοιας προσέγγισης θα στηριχθεί στην υπάρχουσα υλοποίηση, η οποία είναι σχεδιασμένη με ευέλικτα κριτήρια και επιτρέπει τέτοιου είδους μετατροπές, ξεκινώντας από την ίδια τη βάση δεδομένων έως το web layout.

Επίσης, ένα άλλο πιθανό σημείο μελλοντικής επέκτασης αποτελεί ο τρόπος μεταφοράς των δεδομένων που σχετίζονται με τη λειτουργία του συστήματος από και προς το φοιτητολόγιο, ο οποίος και πραγματοποιείται προς το παρόν με batch imports και exports data από τη μία βάση στην άλλη. Απαιτείται δηλαδή αυτή τη στιγμή η ύπαρξη database administrator για την εκτέλεση SQL εντολών στον εκάστοτε database server. Για να γίνει το σύστημα πιο φιλικό και προσβάσιμο σε μη εξειδικευμένους χρήστες, θα πρέπει να αναπτυχθεί ένα διαχειριστικό περιβάλλον, πιθανώς επίσης μέσω web access.

Ανάγκη μελλοντικής επέκτασης χρήζει οπωσδήποτε το τμήμα του ΣΗΕ που απευθύνεται στους καθηγητές του Τμήματος, οι οποίοι αναμένονται να βρουν ιδιαίτερα ενδιαφέρον και χρηστικό το κομμάτι αυτό. Στατιστικά στοιχεία από τις εγγραφές των φοιτητών μπορούν να παραχθούν με αυτοματοποιημένο τρόπο, απευθείας από τη βάση δεδομένων του συστήματος.

Θα πρέπει να τονιστεί, ότι με τη δομή που έχει αναπτυχθεί το ΣΗΕ, είναι ιδιαίτερα εύκολη και απλή η ενσωμάτωση και άλλων υποσυστημάτων, όπως για παράδειγμα του συστήματος παρουσίασης των βαθμών εκάστοτε φοιτητή στο web (<http://www.central.ntua.gr/>), ή/και η ανάπτυξη εξ ολοκλήρου νέων υποσυστημάτων, όπως αυτού της συμπλήρωσης και αποστολής κάθε λογής επίσημων εγγράφων προς (και γιατί όχι και από) τη γραμματεία.

Τέλος, σε καθαρά τεχνικό επίπεδο μπορεί να πραγματοποιηθεί βελτίωση της παρεχόμενης ποιότητας υπηρεσίας με χρήση εξειδικευμένου proxy server για το στατικό κομμάτι του web site και ιδιαίτερων πολυπλοκότερων τεχνικών load balancing και web caching.

## **9.2 Σύνοψη και Συμπεράσματα**

Με την ολοκλήρωση της εφαρμογής του Συστήματος Ηλεκτρονικών Εγγραφών ικανοποιείται πλήρως η ανάγκη για μεταφορά της διαδικασίας εγγραφής των φοιτητών, από το στενό χώρο και χρόνο της αίθουσας της γραμματείας, στον απέραντο «χώρο» του Internet και στην απεριόριστη πρόσβαση 24ώρες/24ώρο και για τις 14 ημέρες που διαρκούν επίσημα οι εγγραφές. Παρέχεται πλέον η δυνατότητα στους φοιτητές να εγγράφονται γρήγορα και προπάντων εύκολα στα μαθήματα και την κατεύθυνση της επιλογής τους, και στο προσωπικό της γραμματείας μία απεριόριστη βοήθεια και ελάφρυνση φόρτου εργασίας κατά την περίοδο αυτή. Επίσης, ελαχιστοποιούνται τα λάθη, οι παραλείψεις και οι καθυστερήσεις στη διαδικασία εγγραφών, καθώς ελαχιστοποιείται ο ανθρώπινος παράγοντας παρέμβασης σε αυτή. Προσφέρονται, τέλος, γρήγορα εύκολα ανανεώσιμες πληροφορίες για οποιοδήποτε μάθημα του Τμήματος, ενώ η ομαδοποίησή τους στις κατηγορίες που έχουν αναφερθεί, δίνει μια πλήρη και ξεκάθαρη εικόνα του Προγράμματος Σπουδών του. Επίσης, το όλο σύστημα είναι ιδιαίτερα ασφαλές, με τη χρήση κρυπτογράφησης στην ίδια τη βάση και SSL για την πρόσβαση, αξιόπιστο και εύπλαστο, ενώ υπάρχει και πλήρης εποπτεία του από τον technical administrator του.



Όσον αφορά τα συμπεράσματα τα οποία εξήχθησαν από τη διπλωματική αυτή εργασία, μπορούμε να τα συνοψίσουμε στα ακόλουθα:

- Η πρόσβαση στην πληροφορία μιας Βάσης Δεδομένων απλοποιείται με τη χρήση του Internet, ενώ μέσω του www το όλο αποτέλεσμα είναι ιδιαίτερα «ελαφρύ».
- Ο συνδυασμός web interface με τη γλώσσα προγραμματισμού Java (JavaServer Pages) αποτελεί μια ιδιαίτερα σταθερή, αξιόπιστη και ανεξάρτητη πλατφόρμας λύση, διευκολύνοντας έτσι κατά πολύ τη μεταφερσιμότητα της εφαρμογής.
- Το www αποτελεί μια κατανεμημένη πηγή πληροφόρησης, με διαρκώς αυξανόμενη χρήση, γεγονός που δικαιολογεί την επιλογή του ως περιβάλλοντος ανάπτυξης και παρουσίασης οποιουδήποτε συστήματος ανάλογου περιεχομένου και προοπτικής.



# 10

## *Βιβλιογραφία*

Κατά τη συγγραφή της διπλωματικής μου εργασίας, πολύτιμες πληροφορίες, γνώσεις και σχήματα χρησιμοποιήθηκαν από τις ακόλουθες πηγές:

### *10.1 Βιβλία:*

- [Βασ99] I. Βασιλείου. Σημειώσεις στο μάθημα Βάσεων Δεδομένων, Ε.Μ.Π., Αθήνα 1999
- [ΒΣ99] Π. Βασιλειάδης, Τ. Σελλής. Προχωρημένα Θέματα Βάσεων Δεδομένων – Συμπληρωματικές Σημειώσεις, Ε.Μ.Π., Αθήνα 1999
- [ΜΧΚΣ94] Β. Μάγκλαρης, Τ. Χιώτης, Θ. Καρούνος, Φ. Σταματελόπουλος. Διαχείριση Δικτύων Υπολογιστών – Σημειώσεις για το μάθημα: Διαχείριση Δικτύων, Ε.Μ.Π., Αθήνα 1994
- [Σκο91] Ε.Σκορδαλάκης. Εισαγωγή στην Τεχνολογία Λογισμικού, Αθήνα 1991
- [Eck98] B. Eckel. Thinking in Java - Prentice Hall PTR 1998
- [EN96] R. Elmasri, S.B. Navathe. Θεμελιώδεις Αρχές Συστημάτων Βάσεων Δεδομένων – τόμος Α, Δίαυλος, Αθήνα 1996
- [EN98] R. Elmasri, S.B. Navathe. Θεμελιώδεις Αρχές Συστημάτων Βάσεων Δεδομένων – τόμος Β, Δίαυλος, Αθήνα 1998
- [NSSH95] E. Nemeth, G. Snyder, S. Seebass, T. R. Hein. Unix® System Administration Handbook – Second Edition, Prentice Hall PTR 1995

## **10.2 Links:**

- [L1] <http://www.ietf.org> [IETF web site]
- [L2] <http://www.ipv6.org> [IPv6 web site]
- [L3] <http://public.web.cern.ch/Public/ACHIEVEMENTS/WEB/history.html>  
[CERN web site]
- [L4] <http://www.internet2.edu> [Internet 2 web site]
- [L5] <http://www.david-guerrero.com/papers/squid/squid.htm>  
[web caching techniques]
- [L6] <http://www.squid.com> [Squid web site]
- [L7] <http://computer.org/internet/> [Dynamic load balancing on web-server systems]
- [L8] <http://www.w3.org> [W3C web site]
- [L9] <http://www.w3.org/TR/html4/> [HTML 4.01 web site]
- [L10] <http://developer.netscape.com/docs/manuals/communicator/jsref/index.htm>  
[Javascript reference]
- [L11] <http://developer.netscape.com/docs/javascript/>  
[Netscape Javascript reference]
- [L12] <http://www.w3.org/XML/1999/XML-in-10-points> [W3C xml web site]
- [L13] <http://www.w3.org/TR/REC-xml> [W3C xml reference]
- [L14] [http://www.jcc.com/SQLPages/jccs\\_sql.htm](http://www.jcc.com/SQLPages/jccs_sql.htm) [info about SQL standards]
- [L15] [http://www.oracle.com/corporate/investor\\_relations/news/index.html?history.html](http://www.oracle.com/corporate/investor_relations/news/index.html?history.html)  
[Oracle history]
- [L16] <http://java.sun.com/> [Java web site]
- [L17] <http://java.sun.com/nav/whatis/?frontpage-shortcuts> [JSP site]
- [L18] <http://java.sun.com/products/jsp> [JSP web site]
- [L19] <http://java.sun.com/products/jsp/index.html> [JSP main reference]
- [L20] <http://technet.oracle.com/tech/java/jroadmap/jsp/listing.htm> [JSP reference]
- [L21] <http://java.sun.com/products/javabeans/> [JavaBeans reference]
- [L22] <http://technet.oracle.com/tech/java/jroadmap/javabean/listing.htm>

[JavaBeans web site]

[L23] <http://technet.oracle.com/tech/java/jroadmap/bc4j/listing.htm> [BC4J web site]

[L24] <http://technet.oracle.com/products/jdev/content.html> [JDeveloper web site]

[L25] <http://java.sun.com/products/servlet/2.2/javadoc/javax/servlet/http/HttpSession.html> [HttpSession interface info]

[L26] <http://java.sun.com/products/jsp/?frontpage-javaplatform> [JSP web site]

[L27] <http://java.sun.com/products/jsp/technical.html> [JSP reference]

[L28] <http://java.sun.com/products/jsp/tags/12/syntaxref12.html>  
[JSP1.2 syntax reference]

### **10.3 RFCs:**

[1] RFC-2616: Hypertext Transfer Protocol -- HTTP/1.1

[2] RFC-2774: An HTTP Extension Framework

[3] RFC-2660: The Secure HyperText Transfer Protocol

[4] RFC-2186: Internet Cache Protocol (ICP), version 2

[5] RFC-2187: Application of Internet Cache Protocol (ICP), version 2

[6] RFC-2376: XML Media Types

[7] RFC-1766: Tags for the Identification of Languages

### **10.4 Περιοδικά:**

[1] Oracle magazine, January/February 2001

[2] Oracle magazine, March/April 2001

[3] Oracle magazine, May/June 2001

[4] Oracle magazine, July/August 2001

[5] Οδηγός Σπουδών, Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών  
Υπολογιστών, Ε.Μ.Π. Αθήνα 2000-2001

## ***10.5 pdf:***

- [1] «Everybody has secrets: New solutions in stored data encryption»  
[Oracle corporation]
  - [2] «WEB interface to Corporate Knowledge»  
[Unixfor company]
  - [3] «Dynamic load balancing on web-server systems»  
[IEEE computer society]
-