

ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Πτυχιακή Εργασία

«Ηλεκτρονική Διαχείριση Εργαστηρίου»



**Δοξαστάκη Μαρία
Κορνελάκης Μιχαήλ**

Επιβλέπων καθηγητής:
Βιδάκης Νικόλαος, Καθηγητής Εφαρμογών

ΗΡΑΚΛΕΙΟ ΙΟΥΝΙΟΣ 2007

Πίνακας Περιεχομένων

1. ΕΙΣΑΓΩΓΗ	5
1. ΕΙΣΑΓΩΓΗ	5
2. ΤΕΧΝΟΛΟΓΙΑ.....	6
2.1 PHP	6
2.1.1 Ιστορική αναδρομή.....	6
2.1.2 Τι είναι η PHP.....	7
2.1.3 Διαφορές με Client Side Scripting.....	7
2.2 MySQL.....	8
2.2.1 Ιστορική αναδρομή.....	8
2.2.2 Σχεσιακές βάσεις δεδομένων.....	9
2.2.3 Τι είναι η MySQL.....	9
2.2.4 Τρόπος λειτουργίας MySQL Βάσης Δεδομένων.....	10
2.3 Apache.....	11
2.3.1 Εισαγωγή.....	11
2.3.2 Χαρακτηριστικά.....	11
2.3.3 Χρήση.....	12
3. ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ ΣΥΣΤΗΜΑΤΟΣ.....	13
3.1 ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ.....	13
3.2 ΚΕΝΤΡΙΚΕΣ ΕΝΟΤΗΤΕΣ ΣΥΣΤΗΜΑΤΟΣ.....	13
3.2.1 Διαχείριση Εξαμήνου.....	13
3.2.2 Διαχείριση Μαθημάτων.....	14
3.2.3 Διαχείριση Τμημάτων.....	14
3.2.4 Διαχείριση Ομάδων Εργασίας.....	15
3.2.5 Διαχείριση Εργασιών και Project.....	15
3.2.6 Τήρηση Απουσιολογίου Εργαστηρίου.....	15
3.2.7 Διαχείριση Βαθμολογίας Εργαστηρίου.....	16
3.2.8 Διαχείριση Παραμέτρων Συστήματος.....	16
3.3 ΔΙΑΓΡΑΜΜΑΤΑ ΑΝΑΛΥΣΗΣ ΣΥΣΤΗΜΑΤΟΣ.....	16
3.3.1 Μοντέλο περιεχομένου.....	16
3.3.2 Σενάρια Χρήσης βάση VORD.....	22
4. ΣΧΕΔΙΑΣΜΟΣ ΣΥΣΤΗΜΑΤΟΣ.....	28
4.1 ΔΙΑΓΡΑΜΜΑΤΑ ΡΟΗΣ ΔΕΔΟΜΕΝΩΝ (D.F.D).....	28
4.2 ΣΧΕΔΙΑΣΜΟΣ DATA MODEL	32
4.2.1 Λίστα δεδομένων ανά απαίτηση.....	33
4.2.2 Έλεγχος επαναλαμβανόμενων δεδομένων.....	35
4.2.3 Ομαδοποίηση δεδομένων σε πίνακες.....	35
4.2.4 Διαγραμματική απεικόνιση Data Model.....	36
5. ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ.....	37

5.1 ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ.....	37
5.1.1 Πίνακες.....	37
5.1.2 Σχέσεις πινάκων.....	38
5.2 ΥΛΟΠΟΙΗΣΗ ΙΣΤΟΣΕΛΙΔΑΣ.....	40
5.2.1 Εγγραφή χρήστη-καθηγητή.....	40
5.2.2 Εισαγωγή χρήστη.....	41
5.2.3 Διαχείριση εξαμήνου.....	42
5.2.4 Διαχείριση μαθημάτων.....	45
5.2.5 Διαχείριση τμημάτων.....	48
5.2.6 Διαχείριση ομάδων εργασίας.....	50
5.2.7 Διαχείριση εργασιών/project.....	52
5.2.8 Διαχείριση βαθμολογίας.....	53
5.2.9 Διαχείριση απουσιολογίου.....	55
5.2.10 Διαχείριση παραμέτρων συστήματος.....	57
6. ΣΥΜΠΕΡΑΣΜΑΤΑ.....	59
7. ΠΑΡΑΡΤΗΜΑ Α.....	60
8. ΠΑΡΑΡΤΗΜΑ Β.....	65
9. ΓΛΩΣΣΑΡΙΟ.....	66
10. ΒΙΒΛΙΟΓΡΑΦΙΑ.....	67

Πίνακας Σχημάτων & Εικόνων

ΣΧΗΜΑ 1: ΜΟΝΤΕΛΟ ΠΕΡΙΕΧΟΜΕΝΟΥ ΣΥΣΤΗΜΑΤΟΣ ELABMANAGE.....	17
ΣΧΗΜΑ 2: ΜΟΝΤΕΛΟ ΠΕΡΙΕΧΟΜΕΝΟΥ ΔΙΑΧΕΙΡΙΣΗΣ ΕΞΑΜΗΝΟΥ	18
ΣΧΗΜΑ 3: ΜΟΝΤΕΛΟ ΠΕΡΙΕΧΟΜΕΝΟΥ ΔΙΑΧΕΙΡΙΣΗΣ ΜΑΘΗΜΑΤΩΝ	18
ΣΧΗΜΑ 4: ΜΟΝΤΕΛΟ ΠΕΡΙΕΧΟΜΕΝΟΥ ΔΙΑΧΕΙΡΙΣΗΣ ΤΜΗΜΑΤΩΝ	19
ΣΧΗΜΑ 5: ΜΟΝΤΕΛΟ ΠΕΡΙΕΧΟΜΕΝΟΥ ΔΙΑΧΕΙΡΙΣΗΣ ΟΜΑΔΩΝ ΕΡΓΑΣΙΑΣ.....	19
ΣΧΗΜΑ 5: ΜΟΝΤΕΛΟ ΠΕΡΙΕΧΟΜΕΝΟΥ ΔΙΑΧΕΙΡΙΣΗΣ ΕΡΓΑΣΙΩΝ/PROJECT	20
ΣΧΗΜΑ 6: ΜΟΝΤΕΛΟ ΠΕΡΙΕΧΟΜΕΝΟΥ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΘΜΟΛΟΓΙΑΣ	20
ΣΧΗΜΑ 7: ΜΟΝΤΕΛΟ ΠΕΡΙΕΧΟΜΕΝΟΥ ΔΙΑΧΕΙΡΙΣΗΣ ΑΠΟΥΣΙΟΛΟΓΙΟΥ.....	21
ΣΧΗΜΑ 8: ΜΟΝΤΕΛΟ ΠΕΡΙΕΧΟΜΕΝΟΥ ΠΑΡΑΜΕΤΡΩΝ ΣΥΣΤΗΜΑΤΟΣ	21
ΣΧΗΜΑ 9: ΣΕΝΑΡΙΟ ΕΙΣΑΓΩΓΗΣ ΧΡΗΣΤΗ	22
ΣΧΗΜΑ 10: ΣΕΝΑΡΙΟ ΠΡΟΒΟΛΗΣ ΤΜΗΜΑΤΩΝ ΒΑΣΗ ΕΠΙΛΟΓΗΣ ΣΥΓΚΕΚΡΙΜΕΝΟΥ ΕΞΑΜΗΝΟΥ ΚΑΙ ΜΑΘΗΜΑΤΟΣ.....	23
ΣΧΗΜΑ 11: ΣΕΝΑΡΙΟ ΣΥΜΠΛΗΡΩΣΗΣ ΚΑΙ ΑΠΟΣΤΟΛΗΣ ΑΙΤΗΣΗΣ ΕΓΓΡΑΦΗΣ ΝΕΟΥ ΧΡΗΣΤΗ..	24
ΣΧΗΜΑ 12: ΣΕΝΑΡΙΟ ΔΗΜΙΟΥΡΓΙΑΣ ΝΕΟΥ ΜΑΘΗΜΑΤΟΣ.....	25
ΣΧΗΜΑ 13: ΣΕΝΑΡΙΟ ΔΙΟΡΘΩΣΗΣ ΣΤΟΙΧΕΙΩΝ ΕΡΓΑΣΙΑΣ.....	26
ΣΧΗΜΑ 14: ΣΕΝΑΡΙΟ ΔΙΑΧΕΙΡΙΣΗΣ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	27
ΣΧΗΜΑ 15: D.F.D ΕΙΣΑΓΩΓΗΣ ΧΡΗΣΤΗ ΣΤΗΝ ΙΣΤΟΣΕΛΙΔΑ	29
ΣΧΗΜΑ 16: D.F.D ΠΡΟΒΟΛΗΣ ΑΠΟΥΣΙΩΝ	29
ΣΧΗΜΑ 17: D.F.D ΕΓΓΡΑΦΗΣ ΝΕΟΥ ΧΡΗΣΤΗ-ΚΑΘΗΓΗΤΗ	30
ΣΧΗΜΑ 18: D.F.D ΔΙΑΓΡΑΦΗΣ ΜΑΘΗΜΑΤΟΣ	31
ΣΧΗΜΑ 19: D.F.D ΕΛΕΓΧΟΥ ΥΠΑΡΞΗΣ USERNAME ΣΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΚΑΤΑ ΤΗ ΔΙΑΔΙΚΑΣΙΑ ΤΗΣ ΝΕΑΣ ΕΓΓΡΑΦΗΣ.....	31
ΣΧΗΜΑ 20: D.F.D ΚΑΤΑΧΩΡΗΣΗΣ ΑΤΟΜΙΚΗΣ ΒΑΘΜΟΛΟΓΙΑΣ.....	32
ΣΧΗΜΑ 21: DATA MODEL.....	36
ΣΧΗΜΑ 22: ΣΧΕΣΕΙΣ ΠΙΝΑΚΩΝ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	39
ΕΙΚΟΝΑ 1: ΟΘΟΝΗ ΕΓΓΡΑΦΗΣ ΝΕΟΥ ΧΡΗΣΤΗ-ΚΑΘΗΓΗΤΗ	40
ΕΙΚΟΝΑ 2: ΟΘΟΝΗ ΕΙΣΑΓΩΓΗΣ ΧΡΗΣΤΗ (LOG IN).....	41
ΕΙΚΟΝΑ 3: ΟΘΟΝΗ ΔΙΑΧΕΙΡΙΣΗΣ ΕΞΑΜΗΝΟΥ.....	43
ΕΙΚΟΝΑ 4: ΟΘΟΝΗ ΔΙΑΧΕΙΡΙΣΗΣ ΑΡΓΙΩΝ ΕΞΑΜΗΝΟΥ	44
ΕΙΚΟΝΑ 5: ΟΘΟΝΗ ΔΗΜΙΟΥΡΓΙΑΣ ΝΕΟΥ ΜΑΘΗΜΑΤΟΣ	46
ΕΙΚΟΝΑ 6: ΟΘΟΝΗ ΑΝΤΙΣΤΟΙΧΗΣΗΣ ΣΠΟΥΔΑΣΤΩΝ ΣΕ ΤΜΗΜΑΤΑ	49
ΕΙΚΟΝΑ 7: ΟΘΟΝΗ ΔΗΜΙΟΥΡΓΙΑΣ ΝΕΑΣ ΟΜΑΔΑΣ ΕΡΓΑΣΙΑΣ	51
ΕΙΚΟΝΑ 8: ΟΘΟΝΗ ΕΜΦΑΝΙΣΗΣ ΕΡΓΑΣΙΩΝ/PROJECT ΓΙΑ ΤΟ ΜΑΘΗΜΑ ‘ΔΙΑΧΕΙΡΙΣΗ ΣΥΣΤΗΜΑΤΩΝ’	52
ΕΙΚΟΝΑ 8: ΟΘΟΝΗ ΕΜΦΑΝΙΣΗΣ ΒΑΘΜΟΛΟΓΙΩΝ	53
ΕΙΚΟΝΑ 9: ΟΘΟΝΗ ΚΑΤΑΧΩΡΗΣΗΣ ΝΕΩΝ ΠΑΡΟΥΣΙΩΝ	56
ΕΙΚΟΝΑ 10: ΗΜΕΡΟΛΟΓΙΟ	57
ΕΙΚΟΝΑ 11: ΟΘΟΝΗ ΔΙΑΧΕΙΡΙΣΗΣ ΠΑΡΑΜΕΤΡΩΝ ΣΥΣΤΗΜΑΤΟΣ	58

1. Εισαγωγή

Σκοπός αυτής της πτυχιακής εργασίας είναι η ανάλυση, σχεδίαση και υλοποίηση ενός ηλεκτρονικού συστήματος Διαχείρισης των Εργαστηριακών μαθημάτων της σχολής (eLabManage).

Το σύστημα αυτό έχει ενδεικτικά τις παρακάτω δυνατότητες:

1. δημιουργεί νέα εξάμηνα και επεξεργάζεται τα στοιχεία του,
2. διαχειρίζεται τα μαθήματα-εργαστήρια της σχολής,
3. διαχειρίζεται τα τμήματα που δημιουργούνται σε κάθε εργαστήριο,
4. διαχειρίζεται τις ομάδες εργασίας που υπάρχουν σε κάθε τμήμα,
5. διαχειρίζεται το απουσιολόγιο των τμημάτων,
6. διαχειρίζεται τις εργασίες - project που δίδονται στο εργαστήριο,
7. διαχειρίζεται την βαθμολογία του εργαστηρίου (εργασίες & τελική εξέταση),
8. συντηρεί τα στοιχεία των εργαστηρίων σε βάση δεδομένων (MySQL) για χρήση από τους καθηγητές.

Μέσα στα πλαίσια της εργασίας ασχοληθήκαμε με:

- θέματα ανάλυσης & σχεδίασης εφαρμογών,
- θέματα web design & development,
- θέματα online databases.

Για το σχεδιασμό της διεπαφής χρησιμοποιήθηκε η γλώσσα προγραμματισμού PHP. Η PHP είναι μια γλώσσα προγραμματισμού με αντικειμενοστραφή χαρακτηριστικά η οποία είναι ιδανική για την ανάπτυξη εφαρμογών για το περιβάλλον του διαδικτύου όπως εφαρμογές διαχείρισης δεδομένων, όπως αυτή που παρουσιάζεται σε αυτήν την πτυχιακή εργασία.

Πέρα από την PHP χρησιμοποιήθηκαν και άλλες τεχνολογίες για την ανάπτυξη της εφαρμογής όπως την HTML, τα Cascading Stylesheets, την Javascript καθώς και βάσεις δεδομένων MySQL που χρησιμοποιήθηκαν ως υπόβαθρο της εφαρμογής. Επίσης έγινε χρήση και του Apache Server.

2. Τεχνολογία

Με απλά λόγια μπορούμε να πούμε ότι, τεχνολογία είναι τα μέσα που χρησιμοποιούμε για την υποβοήθηση της διαδικασίας λύσης ενός προβλήματος που προκύπτει κατά την ανάπτυξη ενός έργου. Τα μέσα αυτά, αποτελούνται από μια σειρά εργαλείων και τεχνικών που χρησιμοποιούνται από αναλυτές και προγραμματιστές για τον παραπάνω σκοπό.

Σε αυτή την ενότητα θα αναφερθούμε στις τρεις βασικότερες τεχνολογίες που χρησιμοποιήθηκαν για την επίτευξη αυτής της πτυχιακής. Την Php για το προγραμματιστικό μέρος, τη MySQL για τη δημιουργία της δυναμικής βάσης δεδομένων και τον Apache ως server του συστήματος.

2.1 PHP

2.1.1 Ιστορική αναδρομή

Η PHP ξεκίνησε ως ένα σύνολο έτοιμων συναρτήσεων γραμμένων σε C (το ακρώνυμο είχε τη σημασία Personal Home Page). Η PHP έγινε τόσο δημοφιλής που το 1997, 2 προγραμματιστές του Israel Institute of Technology, ο Zeev Suraski και ο Andi Gutmans, έγραψαν από την αρχή των διερμηνευτή (parser) αλλάζοντας και τη σημασία του ονόματος της γλώσσας στο αναδρομικό ακρώνυμο PHP: Hypertext Preprocessor (Προεπεξεργαστής υπερκειμένου) και δημιουργώντας της βάση για την PHP 3.

Το 1998 λίγο μετά την κυκλοφορία της PHP 3 άρχισε η ανάπτυξη ενός νέου πυρήνα με στόχο την βελτίωση της απόδοσης σύνθετων εφαρμογών. Ο νέος αυτός πυρήνας ονομάστηκε Zend (το όνομα προέρχεται από τα αρχικά των ονομάτων των δημιουργών της PHP) και παρουσιάστηκε στα μέσα του 1999. Η PHP 4 η οποία βασίστηκε σε αυτόν τον νέο πυρήνα κυκλοφόρησε το Μάιο του 2000. Η νέα αυτή έκδοση εκτός από τη βελτιωμένη απόδοση προσέφερε και μια σειρά νέων δυνατοτήτων όπως την υποστήριξη πολλών νέων διακομιστών (Web Servers), συνόδους HTTP και βελτιωμένη ασφάλεια.

Τέλος το 2004 κυκλοφόρησε η PHP 5 στηριγμένη στον πυρήνα Zend 2 η οποία προσέφερε μια πληθώρα νέων δυνατοτήτων και μια νέα βελτιωμένη υποστήριξη για αντικειμενοστραφή προγραμματισμό. Τα

τελευταία χρόνια η PHP έχει αναπτυχθεί αρκετά ώστε να θεωρείται μια από τις πιο δυνατές γλώσσες που στηρίζουν τον παγκόσμιο ιστό.

2.1.2 Τι είναι η PHP

Η PHP είναι μια γλώσσα προγραμματισμού ανοικτού κώδικα (Open Source) η οποία διερμηνεύεται (Interpreted Programming Language) στη μεριά του διακομιστή (server side scripting) και η οποία αναπτύχθηκε για χρήση στον παγκόσμιο ιστό. Το κυριότερο ατού της PHP σε αντίθεση με άλλες γλώσσες για παραγωγή δυναμικού υπερκειμένου είναι ότι ενσωματώνεται στον κώδικα HTML. Αυτό σημαίνει ότι αντί ο προγραμματιστής να γράψει ένα πρόγραμμα που έχει ως έξοδο κώδικα HTML, γράφει κώδικα HTML στο οποίο παρεμβάλετε ο κώδικας PHP. Ο κώδικας PHP παρεμβάλλεται σε ένα ειδικό HTML tag του οποίου τα περιεχόμενα αναλύονται από τον Parser της PHP. Το tag αυτό ξεκινά ως `<?PHP` και καταλήγει ως `?>`. Για να γίνει κατανοητό τι σημαίνει το παραπάνω, θεωρείστε το εξής αρχείο `index.php`:

```
<html>
<body>
<?php echo "Welcome"; ?>
</body>
</html>
```

Προτού ο διακομιστής στείλει το αρχείο στον πελάτη (client) θα αναγνωρίσει το tag `<?php ... ?>` και θα το αναλύσει χρησιμοποιώντας των parser της php. Το αποτέλεσμα της εξόδου θα είναι «*Welcome*» μιας και η εντολή `echo` είναι η εντολή της php για εκτύπωση συμβολοσειρών.

2.1.3 Διαφορές με Client Side Scripting

Πολλοί αναρωτιούνται γιατί είναι απαραίτητη μια γλώσσα η οποία να διερμηνεύεται στον διακομιστή και όχι στον υπολογιστή του πελάτη όπως η JavaScript ή η HTML. Οι απαντήσεις στην παραπάνω ερώτηση είναι πολλές. Η βασική διαφορά της HTML είναι ότι μεταφράζεται από τον browser, δεν εκτελείται στον server. Γράφοντας κώδικα ο οποίος να εκτελείται στον server μπορεί να επιτευχθούν πολλά περισσότερα πράγματα. Ακολουθούν μερικά παραδείγματα:

- Μπορούμε να φτιάξουμε μια ιστοσελίδα η οποία θα ανιχνεύει τον τύπο του browser που χρησιμοποιεί ο χρήστης και έπειτα να

βελτιστοποιήσουμε τις απαιτούμενες πληροφορίες για αυτόν τον browser.

- Είναι πιο εύκολο να τροποποιήσουμε περιεχόμενα μιας ιστοσελίδας, ενημερώνοντας τα περιεχόμενα μιας βάσης δεδομένων, από το να αλλάξουμε τον ίδιο τον HTML κώδικα.
- Έχουμε τη δυνατότητα να δημιουργήσουμε σελίδες οι οποίες μπορούν να είναι πλήρως παραμετροποιήσιμες ώστε να εμφανίζουν μόνο αυτά που ενδιαφέρουν τον κάθε χρήστη.
- Μπορούμε να εμφανίσουμε και να ενημερώσουμε βάσεις δεδομένων, που περιέχονται σε μια ιστοσελίδα, καθώς και να διαχειριστούμε πληροφορίες σε αυτές, ταξινομώντας τις εγγραφές με οποιαδήποτε σειρά ή εμφάνιση θέλουμε.
- Δημιουργούμε ιστοσελίδες οι οποίες εναλλάσσουν σειρές διαφορετικών γραφημάτων.
- Λαμβάνουμε πληροφορίες από το χρήστη και επιστρέφουμε πληροφορία βασισμένη σε αυτή.

Ο πελάτης δεν έχει καμία πρόσβαση στον κώδικα PHP. Ο κώδικας PHP που εκτελείται στον διακομιστή μπορεί να έχει ως σκοπό τη συλλογή δεδομένων από τοπικές βάσεις δεδομένων ή την επεξεργασία μιας εικόνας. Τα δεδομένα αυτά δεν μπορούν με κανένα τρόπο να είναι άμεσα προσβάσιμα από τον οποιονδήποτε όχι μόνο για τους προφανείς λόγους ασφαλείας αλλά και γιατί είναι αδύνατον να ανταλλάσσονται αυτά τα δεδομένα ανάμεσα στον πελάτη και στον εξυπηρετητή.

Ο Parser της PHP έχει άμεση πρόσβαση στο σύστημα αρχείων, στους πόρους και στο υλικό του server και μπορεί να συλλέγει και να εμφανίζει πληροφορίες από κάθε στοιχείο του server στον πελάτη.

2.2 MySQL

2.2.1 Ιστορική αναδρομή

Η MySQL είναι διαθέσιμη από το 1996, αλλά η ιστορία της ξεκινά από το 1979. Κέρδισε το βραβείο Choice Award του Linux Journal Readers, για τρία χρόνια συνεχεία. Είναι πλέον διαθέσιμη με άδεια ανοικτού κώδικα, αλλά και με εμπορικές άδειες. Αυτή τη στιγμή είναι η δημοφιλέστερη σχεσιακή βάση δεδομένων ανοικτού κώδικα που κυριαρχεί στον τομέα της ανάπτυξης δικτυακών εφαρμογών.

2.2.2 Σχεσιακές βάσεις δεδομένων

Οι σχεσιακές βάσεις δεδομένων έχουν τα εξής χαρακτηριστικά:

1. Παρέχουν γρηγορότερη πρόσβαση στα δεδομένα από τα επίπεδα αρχεία.
2. Μπορούν να δημιουργηθούν ερωτήματα για να εξαχθούν σύνολα από δεδομένα που ανταποκρίνονται σε κάποια κριτήρια.
3. Έχουν ενσωματωμένους μηχανισμούς για τον χειρισμό ταυτόχρονης πρόσβασης, ώστε ο προγραμματιστής να μην ανησυχεί για αυτό.
4. Παρέχουν τυχαία πρόσβαση στα δεδομένα.
5. Έχουν ενσωματωμένα συστήματα δικαιωμάτων.

Σε πιο αυστηρούς όρους η χρήση μιας σχεσιακής βάσης δεδομένων μας επιτρέπει γρήγορα και εύκολα να απαντάμε σε διάφορα ερωτήματα (για παράδειγμα, ποιοι σπουδαστές είναι εγγεγραμμένοι σε μια εργαστηριακή ομάδα).

Οι σχεσιακές βάσεις δεδομένων αποτελούνται από σχέσεις, που συνήθως ονομάζονται πίνακες. Ένας πίνακας είναι ακριβώς αυτό που υποδεικνύει το όνομα του, δηλαδή είναι δεδομένα. Κάθε πίνακας αποτελείται από γραμμές και στήλες. Κάθε γραμμή αντιπροσωπεύει μια διαφορετική εγγραφή. Κάθε στήλη του πίνακα έχει μοναδικό όνομα και περιέχει διαφορετικά δεδομένα.

2.2.3 Τι είναι η MySQL

Η MySQL είναι ένα πολύ γρήγορο και δυνατό σύστημα διαχείρισης βάσεων δεδομένων. Μια βάση δεδομένων επιτρέπει να αποθηκεύουμε, να αναζητάμε, να ταξινομούμε και να ανακαλούμε τα δεδομένα αποτελεσματικά. Ο MySQL διακομιστής ελέγχει την πρόσβαση στα δεδομένα, για να μπορούν να δουλεύουν πολλοί χρήστες ταυτόχρονα, για να παρέχει γρήγορη πρόσβαση και να διασφαλίσει ότι μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση. Συνεπώς είναι ένας πολυνηματικός διακομιστής πολλαπλών χρηστών. Χρησιμοποιεί την SQL (Structured Query Language), την τυπική γλώσσα ερωτημάτων για βάσεις δεδομένων, παγκόσμια.

Ένα από τα μεγαλύτερα πλεονεκτήματα της MySQL είναι η δυνατότητα που δίνει για σύνδεση σε πολλές διαφορετικές βάσεις δεδομένων. Οι βάσεις δεδομένων που υποστηρίζονται περιλαμβάνουν τις: Adabas D, InterBase, PostgreSQL, dBase, FrontBase, SQLite, Empress,

mSQL, Solid, FilePro (read-only), Direct MS-SQL, Sybase, Hyperwave, MySQL, Velocis, IBM, ODBC.

2.2.4 Τρόπος λειτουργίας MySQL Βάσης Δεδομένων

Ο τρόπος λειτουργίας της MySQL είναι ίδιος με αυτόν που ακολουθούν όλες οι Web βάσεις δεδομένων. Τα βήματα λειτουργίας των αρχιτεκτονικών των Web Βάσεων Δεδομένων μπορούν να συνοψιστούν στα παρακάτω:

1. Ο web browser ενός χρήστη κάνει μια HTTP αίτηση για μια συγκεκριμένη ιστοσελίδα. Για παράδειγμα, μπορεί να κάνει μια αναζήτηση για όλους τους σπουδαστές οι οποίοι είναι εγγεγραμμένοι σε μια συγκεκριμένη εργαστηριακή ομάδα. Έστω ότι η σελίδα των αποτελεσμάτων αναζήτησης ονομάζεται results.php
2. Ο web διακομιστής λαμβάνει την αίτηση για τη σελίδα results.php, ανακαλεί το αρχείο και το περνά στην μηχανή PHP για επεξεργασία.
3. Η μηχανή PHP αρχίζει την ανάλυση του script. Μέσα στο script, υπάρχει μια εντολή που συνδέει την βάση δεδομένων και εκτελεί ένα ερώτημα (την αναζήτηση των σπουδαστών). Η PHP ανοίγει μια σύνδεση με τον MySQL διακομιστή και στέλνει το κατάλληλο ερώτημα.
4. Ο MySQL διακομιστής λαμβάνει το ερώτημα της βάσης δεδομένων και το επεξεργάζεται και στέλνει τα αποτελέσματα (μια λίστα σπουδαστών), ξανά στη μηχανή PHP.
5. Η μηχανή PHP σταματά την εκτέλεση του script, που συνήθως περιλαμβάνει την μορφοποίηση των αποτελεσμάτων του ερωτήματος σε HTML. Επιστρέφει μετά την τελική HTML σελίδα στο web διακομιστή.
6. Ο web διακομιστής περνά την HTML σελίδα ξανά στο browser, όπου ο χρήστης μπορεί να δει τη λίστα των σπουδαστών που ζήτησε.

2.3 Apache

2.3.1 Εισαγωγή

Ο Apache HTTP Server (ή εν συντομία Apache) είναι ένας πασίγνωστος web server ο οποίος έπαιξε βασικό ρόλο στην αρχική ανάπτυξη του διαδικτύου. Ο Apache ήταν η πρώτη εφαρμόσιμη εναλλακτική του Netscape Communications Corporation web server (πλέον γνωστός σαν Sun Java System Web Server) και από τότε εξελίχθηκε σε ανταγωνιστή και των βασισμένων σε unix web server όσον αφορά θέματα λειτουργικότητας και επιδόσεων. Από τον Απρίλιο του 1996 ο Apache είναι ο πιο δημοφιλής HTTP Server του διαδικτύου και σύμφωνα με έρευνες το Μάρτιο του 2007 εξυπηρετεί το 58% όλων των ιστοσελίδων του διαδικτύου!

Ο Apache αναπτύσσεται και συντηρείται από μια ανοικτή κοινότητα δημιουργών υπό την «κηδεμονία» της Apache Software Foundation. Είναι διαθέσιμος για μια μεγάλη γκάμα λειτουργικών συστημάτων συμπεριλαμβανόμενων των Windows, Linux, Mac OS. Και το κυριότερο είναι μια δωρεάν εφαρμογή ανοικτού κώδικα.

Βασικός ανταγωνιστής του Apache σήμερα, είναι ο Microsoft Internet Information Services (IIS).

2.3.2 Χαρακτηριστικά

Ο Apache υποστηρίζει μια ποικιλία χαρακτηριστικών, πολλά από τα οποία εκτελούνται σαν μεταγλωττισμένες υπομονάδες (modules) που επεκτείνουν τη λειτουργικότητα του πυρήνα. Αυτές μπορούν να περιλαμβάνουν από υποστήριξη για server-side γλώσσες προγραμματισμού (όπως η php) μέχρι διαγράμματα πιστοποίησης (authentication schemes). Ένα δείγμα άλλων χαρακτηριστικών, περιλαμβάνει την υποστήριξη SSL και TLS (mod_ssl), ένα proxy module, ένα χρήσιμο URL κειμενογράφο (που εκτελείται από το mod_rewrite), τα αρχεία καταγραφής (log files εκτελούνται από το mod_log_config), και την υποστήριξη φιλτραρίσματος (mod_include και mod_ext_filter). Τα logs του Apache μπορούν να αναλυθούν μέσω ενός web browser.

Η εικονική φιλοξενία (virtual hosting) επιτρέπει σε μια εγκατάσταση Apache να εξυπηρετήσει πολλούς διαφορετικούς ιστοσελίδες. Παραδείγματος χάριν, μια μηχανή, με μια εγκατάσταση Apache θα μπορούσε ταυτόχρονα να εξυπηρετήσει www.example.com, www.test.com,

test47.test-server.test.com, κ.λπ... Το Apache χαρακτηρίζεται από διαμορφώσιμα μηνύματα λάθους (error messages), βάσεις δεδομένων επικύρωσης βασισμένες στο DBMS (πρόγραμμα διαχείρισης βάσεων δεδομένων), και τη διαπραγμάτευση περιεχομένου (content negotiation).

Υποστηρίζεται επίσης από διάφορες γραφικές αλληλεπιδράσεις με τον χρήστη (graphical user interfaces-GUIs) που επιτρέπουν την ευκολότερη διαμόρφωση του server.

2.3.3 Χρήση

Ο Apache χρησιμοποιείται για να εξυπηρετεί και στατικές αλλά και δυναμικές ιστοσελίδες. Πολλές εφαρμογές δικτύου σχεδιάζονται σύμφωνα με το περιβάλλον και τα χαρακτηριστικά που παρέχει ο Apache.

Ο Apache χρησιμοποιείται και για πολλές άλλες εργασίες, όπου το περιεχόμενο πρέπει να είναι διαθέσιμο με έναν ασφαλή και αξιόπιστο τρόπο. Ένα παράδειγμα είναι το μοίρασμα αρχείων από ένα προσωπικό υπολογιστή, στο διαδίκτυο. Ένας χρήστης ο οποίος έχει τον Apache εγκατεστημένο στον υπολογιστή του, μπορεί να βάλει ελεύθερα αρχεία στο document root του Apache, τα οποία μπορούν να διαμοιραστούν.

Οι προγραμματιστές που αναπτύσσουν ιστοσελίδες συχνά χρησιμοποιούν μια τοπικά εγκατεστημένη έκδοση του Apache για να προεπισκοπήσουν και να ελέγξουν τον κώδικα της ιστοσελίδας όποτε θέλουν.

3. Ανάλυση Απαιτήσεων Συστήματος

Πριν σχεδιάσουμε ένα σύστημα, θα πρέπει να γνωρίζουμε τις λειτουργίες που πρέπει να έχει το σύστημα. Γι' αυτό το λόγο ξεκινάμε τη διαδικασία ανάπτυξης του συστήματος με την εξέταση των απαιτήσεων. Οι απαιτήσεις είναι μία περιγραφή του τι μπορεί το σύστημα να κάνει έτσι ώστε να ικανοποιεί τον σκοπό για τον οποίο αναπτύσσεται. Οι απαιτήσεις εστιάζουν μόνο στις λειτουργίες του συστήματος και όχι στον τρόπο υλοποίησης αυτών.

3.1 Γενική Περιγραφή Συστήματος

Το σύστημα αναπτύσσεται με σκοπό να δίνει τη δυνατότητα στους καθηγητές του Τ.Ε.Ι. να διαχειρίζονται ηλεκτρονικά τα εργαστήρια που διεξάγονται στη σχολή. Μέσα στα πλαίσια της διαχείρισης των εργαστηρίων περιλαμβάνονται οι ενότητες του εξαμήνου, των μαθημάτων, των τμημάτων, των ομάδων εργασιών, των εργασιών/project, της βαθμολογίας, του απουσιολογίου καθώς επίσης και των παραμέτρων του συστήματος.

3.2 Κεντρικές Ενότητες Συστήματος

3.2.1 Διαχείριση Εξαμήνου

Το eLabManage θα πρέπει να δίνει την δυνατότητα στον χρήστη:

- Να προσδιορίζει τα στοιχεία του νέου εξαμήνου και να διαχειρίζεται τα στοιχεία προηγούμενων εξαμήνων.
- Να προσδιορίζει τις επίσημες αργίες του εξαμήνου.
- Τα στοιχεία των παλιών εξαμήνων θα πρέπει να αποθηκεύονται μόνιμα στη βάση δεδομένων του προγράμματος και μετά τη λήξη του εξαμήνου να κλειδώνονται, έτσι ώστε να διασφαλίζεται η αυθεντικότητα και ακεραιότητά τους. Θα υπάρχει η δυνατότητα ανάγνωσης και αλλαγής στοιχείων αλλά όχι η διαγραφή τους.
- Όταν ο χρήστης θελήσει να δημιουργήσει ξανά ένα εξάμηνο το οποίο ήδη υπάρχει (παλιό ή τρέχον), το πρόγραμμα θα τον

προειδοποιεί μέσω ενός μηνύματος ότι η ενέργεια αυτή δεν μπορεί να πραγματοποιηθεί. (ο έλεγχος θα γίνεται με κώδικα php).

- Για να ολοκληρωθεί η διαχείριση εξαμήνου θα δοθεί δυνατότητα επεξεργασίας των στοιχείων στο χρήστη με την μορφή φόρμας java scripts ή php.

3.2.2 Διαχείριση Μαθημάτων

Το eLabManage θα πρέπει να δίνει την δυνατότητα στον χρήστη:

- Να δημιουργεί νέα μαθήματα.
- Να επεξεργάζεται τα στοιχεία των μαθημάτων που είδη υπάρχουν και να έχει τη δυνατότητα αλλαγής αυτών ή ακόμα και διαγραφής του μαθήματος.

3.2.3 Διαχείριση Τμημάτων

Το eLabManage θα πρέπει να δίνει την δυνατότητα στον χρήστη:

- να δημιουργεί νέα τμήματα σε οποιαδήποτε τάξη (τάξη= συγκεκριμένο εξάμηνο και εργαστηριακό μάθημα).
- Με τη δημιουργία του νέου τμήματος θα ελέγχεται αυτόματα η ύπαρξη της τάξης που έχει επιλέξει ο χρήστης, κι αν δεν υπάρχει θα την δημιουργεί χωρίς να τον ρωτάει.
- Θα χρειάζεται το σύστημα να ελέγχει και να απαγορεύει διπλοεγγραφή σπουδαστή σε μία τάξη.
- Ο χρήστη θα πρέπει να μπορεί να εισάγει στο τμήμα εγγεγραμμένους από εξωτερικά αρχεία δικά του, όπως για παράδειγμα από μια λίστα αναμονής.
- Η εγγραφή των σπουδαστών στο μάθημα γίνεται από το αρμόδιο τμήμα της σχολής μέσω ενός προγράμματος εγγραφών το οποίο έχει την δυνατότητα εξαγωγής στοιχείων σε μορφή κειμένου (text).
- Το eLabManage θα πρέπει να δίνει την δυνατότητα στον χειριστή να εισαγάγει τα στοιχεία από το πρόγραμμα των εγγραφών.
- Για να ολοκληρωθεί η διαχείριση τμημάτων θα δοθεί στο χρήστη η δυνατότητα επεξεργασίας (προβολής, διαγραφή, αλλαγή κτλ.) των στοιχείων των τμημάτων με την μορφή φόρμας java scripts ή php.

3.2.4 Διαχείριση Ομάδων Εργασίας

Το eLabManage θα πρέπει να δίνει την δυνατότητα στον χρήστη:

- να δημιουργεί νέες ομάδες εργασίας και να προσδιορίζει τα στοιχεία τους.
- Ο διαχωρισμός των σπουδαστών ενός τμήματος σε ομάδες εργασίας θα προσδιορίζεται από τον υπεύθυνο καθηγητή.
- Για να ολοκληρωθεί η διαχείριση ομάδων εργασίας θα δοθεί στο χρήστη η δυνατότητα επεξεργασίας, δηλαδή προβολής και διαγραφής των στοιχείων των ομάδων, με την μορφή φόρμας java scripts ή php.

3.2.5 Διαχείριση Εργασιών και Project

Το eLabManage θα πρέπει να δίνει την δυνατότητα στον χρήστη:

- να διαχειρίζεται το περιεχόμενο των εργασιών,
- να διαχειρίζεται το ποσοστό συμμετοχής τους στο τελικό βαθμό,
- να συνδέει τις εργασίες με το κάθε εξάμηνο,
- να διαχειρίζεται το περιεχόμενο των project,
- να διαχειρίζεται το ποσοστό συμμετοχής τους στο τελικό βαθμό,
- να συνδέει τα project με το κάθε εξάμηνο,
- να γίνεται διαχωρισμός μεταξύ project εξαμήνου και project για φοιτητές που έχουν κατοχυρώσει το μάθημα.
- Με τη μορφή μιας φόρμας-ερωτηματολόγιο (java scripts ή php) θα δίνεται η δυνατότητα στο χρήστη να επιλέξει για όλα τα παραπάνω, προσδιορίζοντας πλήρως τις απαιτήσεις και τον τρόπο διεξαγωγής του τμήματος για το συγκεκριμένο εξάμηνο.

3.2.6 Τήρηση Απουσιολογίου Εργαστηρίου

Το eLabManage θα πρέπει να δίνει την δυνατότητα στον χρήστη:

- Να ελέγχει αν οι επίσημες αργίες συμπίπτουν ημερολογιακά με το μάθημα και να προειδοποιεί τον χρήστη όταν προσπαθεί να κάνει καταχώρηση παρουσιών σε ημερομηνία επίσημης αργίας.
- Να τηρεί απουσιολόγιο για κάθε τμήμα και σπουδαστή.
- Να ενημερώνει με κατάλληλο τρόπο για τις απουσίες κάθε φοιτητή και να προειδοποιεί έγκαιρα σε περίπτωση κινδύνου απώλειας του μαθήματος λόγω απουσιών.

- Να προβάλλει και να εκτυπώνει συγκεντρωτικούς πίνακες παρουσιών ανά τμήμα.

3.2.7 Διαχείριση Βαθμολογίας Εργαστηρίου

Το eLabManage θα πρέπει να δίνει την δυνατότητα στον χρήστη:

- Να καταγράφει την βαθμολογία σε επίπεδο εργασίας και ομάδας.
- Να συνδέει την βαθμολογία κάθε εργασίας με το ποσοστό συμμετοχής της εργασίας και να δίνει προοδευτική συνολική βαθμολογία.
- Να προβάλλει και να εκτυπώνει συγκεντρωτικούς πίνακες βαθμολογιών εργασιών και project ανά τμήμα.
- Οι εργασίες και τα project οι οποίες θα εμφανίζονται σε κάθε ομάδα για βαθμολόγηση θα είναι μόνο αυτές οι οποίες έχουν συνδεθεί με το τμήμα και συνεπώς με το εξάμηνο στην διαδικασία «Διαχείριση Εργασιών και Project».

3.2.8 Διαχείριση Παραμέτρων Συστήματος

Το eLabManage θα πρέπει να δίνει την δυνατότητα στον χρήστη:

- Να διαχειρίζεται όλες τις παραμέτρους του συστήματος.
- Να βλέπει τις επιλογές που υπάρχουν για κάθε παράμετρο και εισάγει νέες εγγραφές σε κάθε μία από αυτές.
- Οι νέες επιλογές των παραμέτρων να χρησιμοποιούνται άμεσα στη λειτουργία του προγράμματος, χωρίς καμία άλλη παρέμβαση του χρήστη.
- Να μην δίνετε η δυνατότητα διαγραφής παραμέτρων που υπάρχουν ήδη, για να μην υπάρχουν αλλοιώσεις σε στοιχεία που υπάρχουν στη βάση δεδομένων.

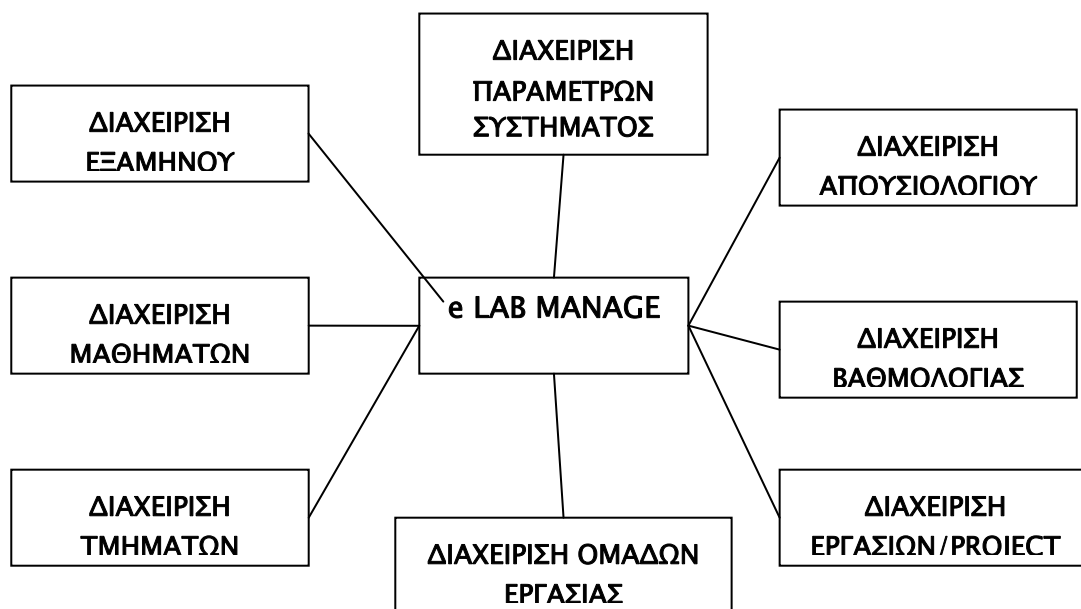
3.3 Διαγράμματα Ανάλυσης συστήματος

3.3.1 Μοντέλο περιεχομένου

Τα μοντέλα περιεχομένου είναι ένα είδος διαγραμμάτων που χρησιμοποιούνται για να προσδιορίζουν τα όρια του συστήματος. Υπάρχει όμως πιθανότητα τα όρια αυτά να επηρεασθούν από κοινωνικούς ή επιχειρησιακούς παράγοντες.

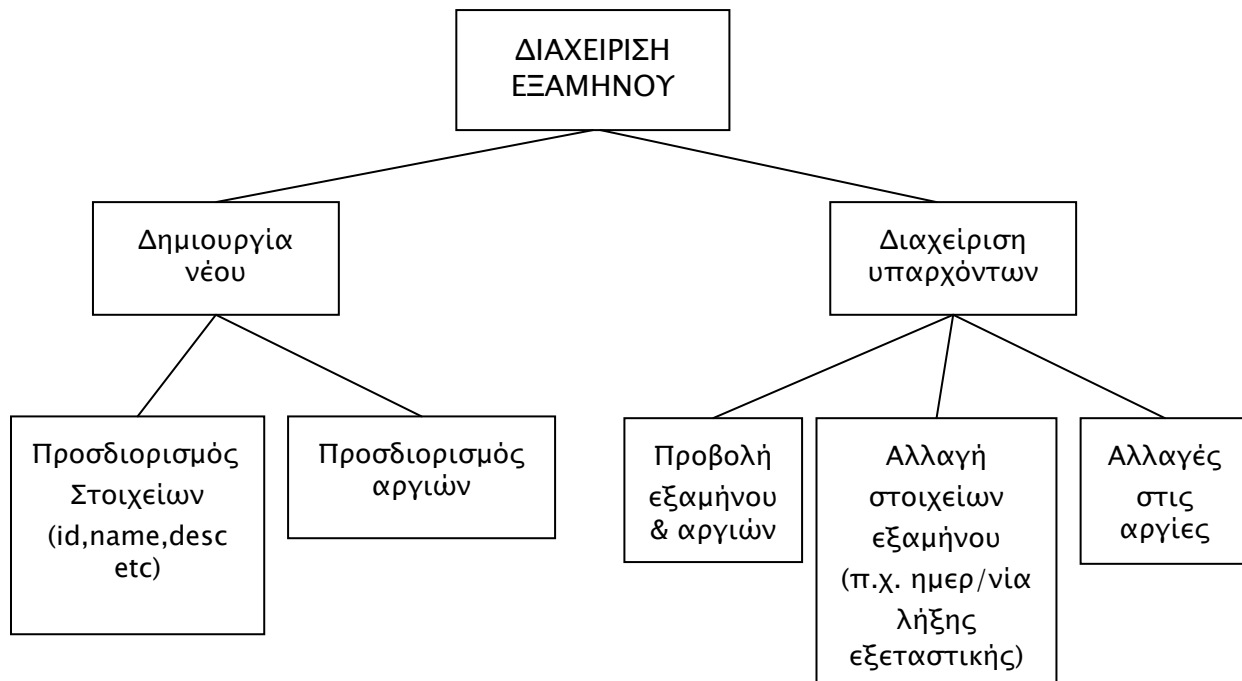
Τα παρακάτω διαγράμματα προσδιορίζουν την ιστοσελίδα «Ηλεκτρονική Διαχείριση Εργαστηρίων» (“eLabManage”) σε ανάλυση τριών επιπέδων.

Μοντέλο Περιεχομένου 1^{ου} επιπέδου

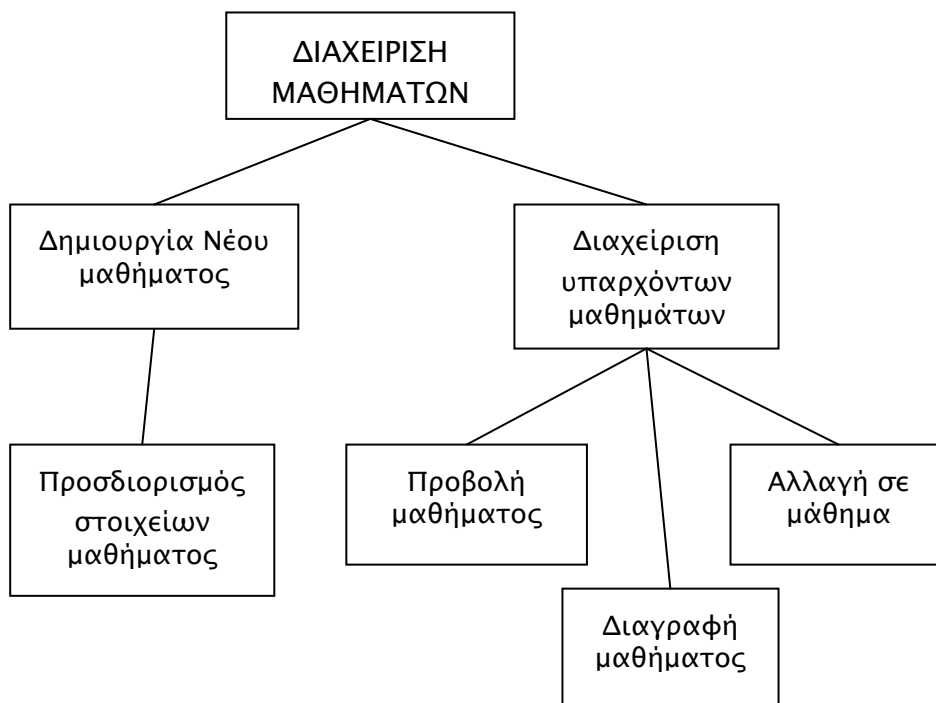


Σχήμα 1: Μοντέλο περιεχομένου συστήματος eLabManage

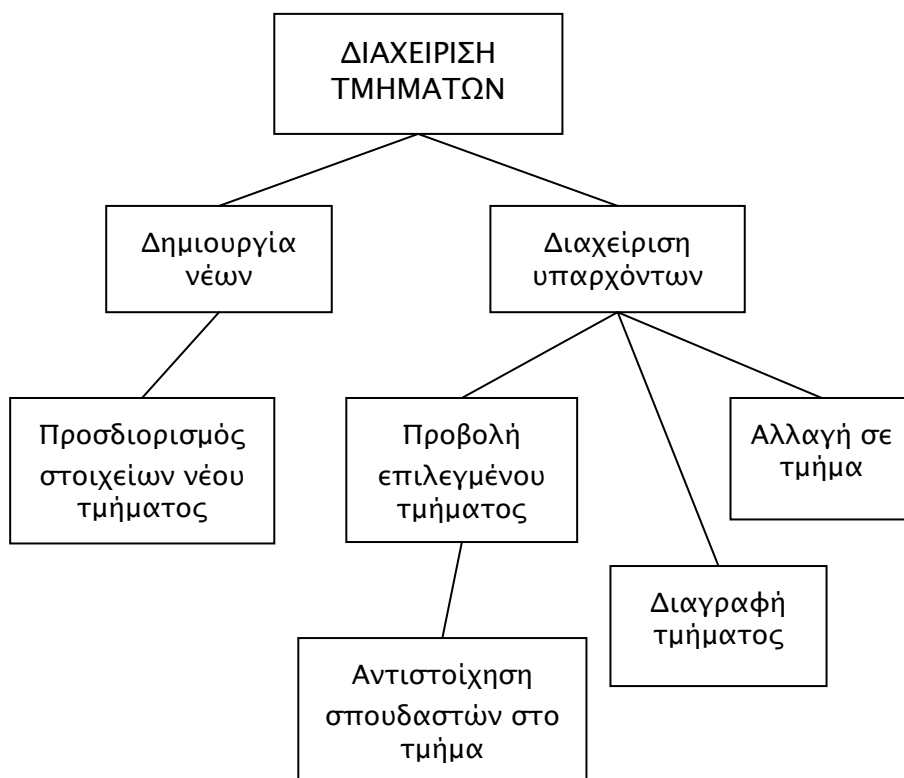
Μοντέλα Περιεχομένου 2^{ου} και 3^{ου} επιπέδου



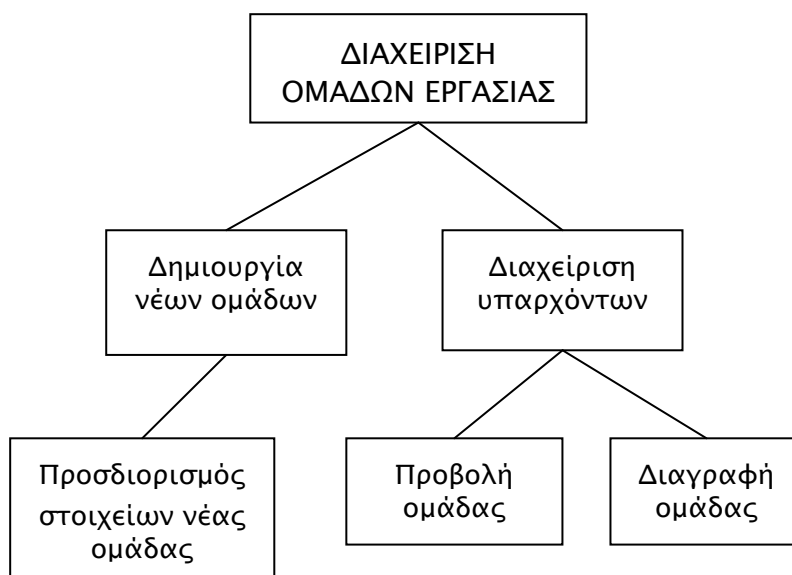
Σχήμα 2: Μοντέλο περιεχομένου διαχείρισης εξαμήνου



Σχήμα 3: Μοντέλο περιεχομένου διαχείρισης μαθημάτων



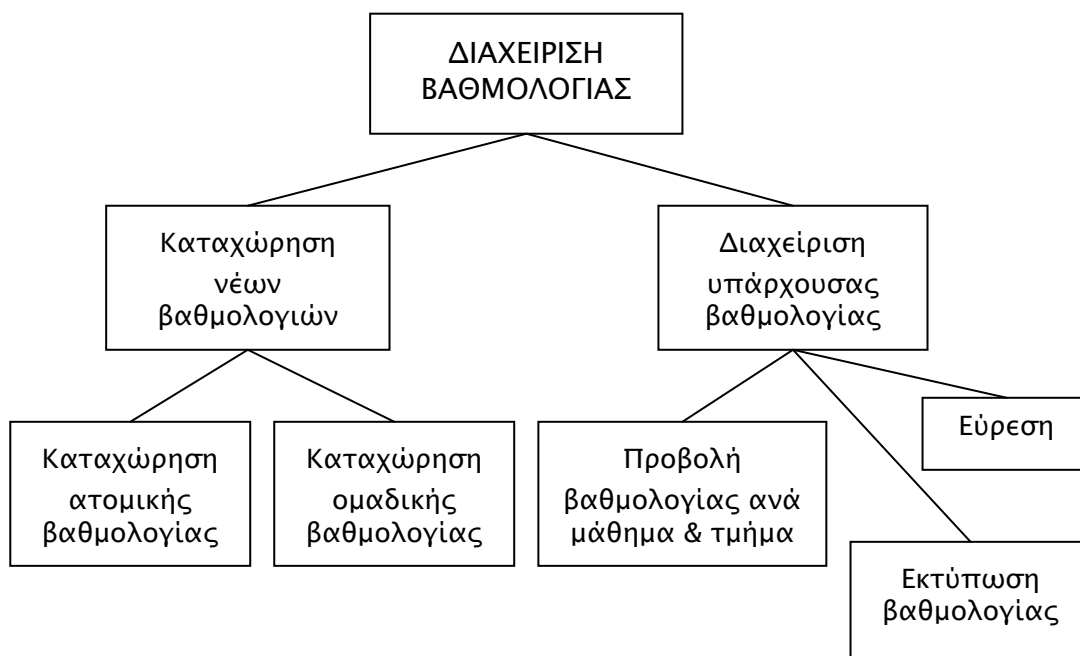
Σχήμα 4: Μοντέλο περιεχομένου διαχείρισης τμημάτων



Σχήμα 5: Μοντέλο περιεχομένου διαχείρισης ομάδων εργασίας



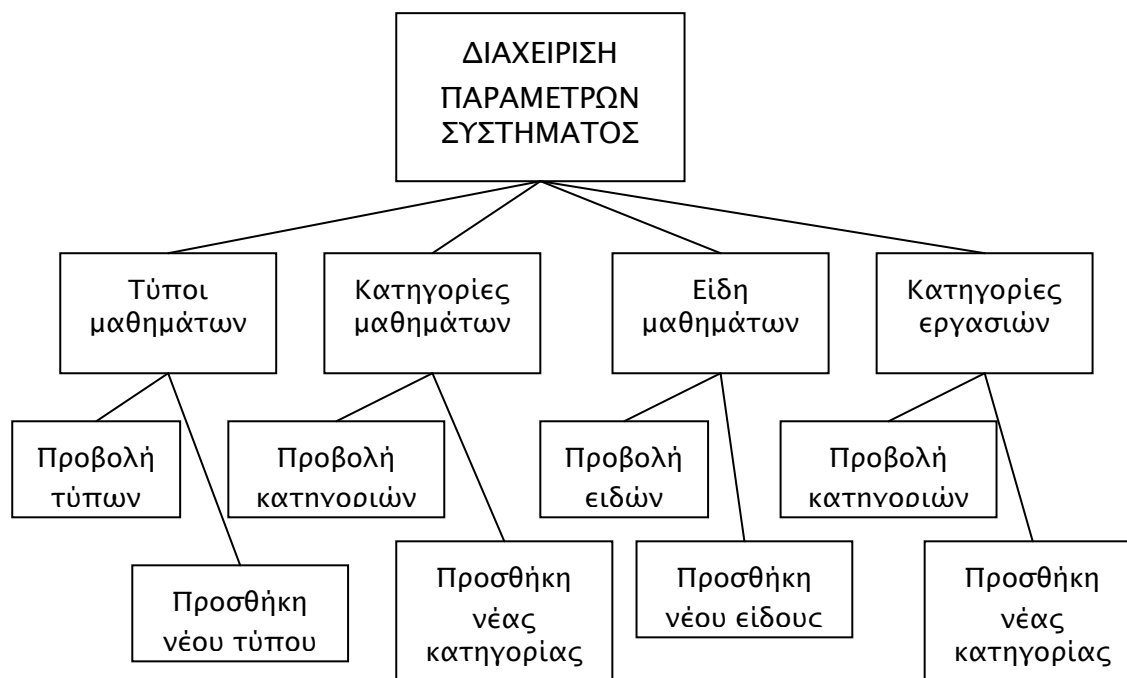
Σχήμα 5: Μοντέλο περιεχομένου διαχείρισης εργασιών/project



Σχήμα 6: Μοντέλο περιεχομένου διαχείρισης βαθμολογίας



Σχήμα 7: Μοντέλο περιεχομένου διαχείρισης απουσιολογίου



Σχήμα 8: Μοντέλο περιεχομένου παραμέτρων συστήματος

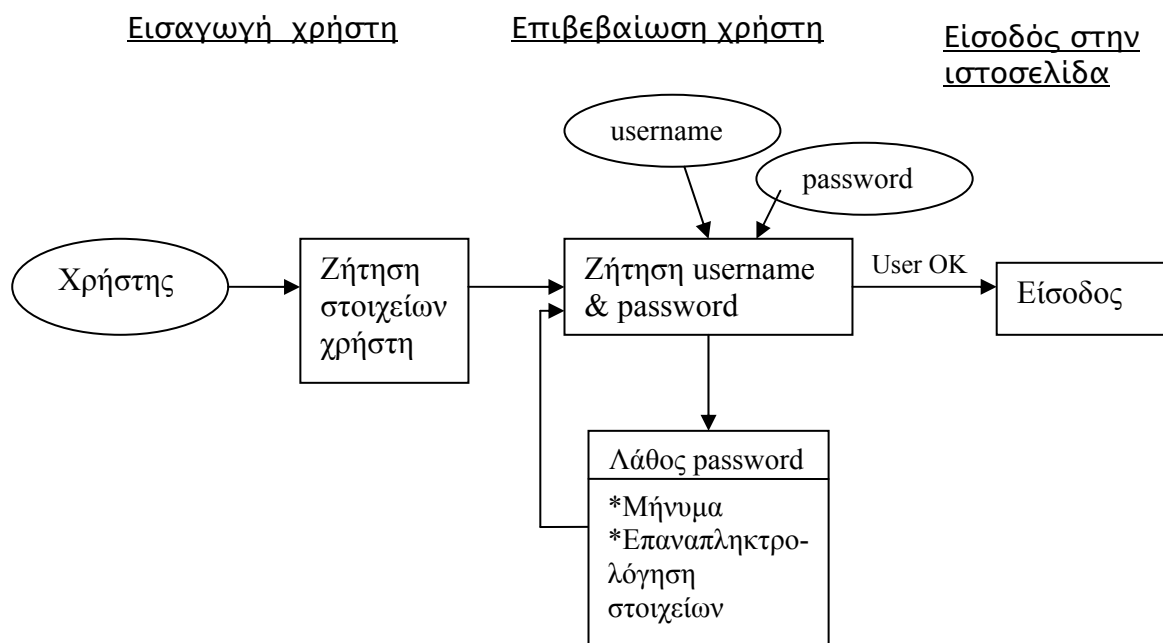
3.3.2 Σενάρια Χρήσης βάση VORD

Τα σενάρια είναι περιγραφές του πως λειτουργεί ένα σύστημα στην πράξη και είναι χρήσιμα στην επιτευξιμότητα των απαιτήσεων. Με τα σενάρια μπορούμε εύκολα να προσθέσουμε λεπτομέρειες σε γενικές περιγραφές απαιτήσεων.

Αν αναλύαμε ένα σενάριο θα παρατηρούσαμε ότι στην αρχή του σεναρίου περιγράφεται η κατάσταση του συστήματος, μετά ακολουθεί η ροή των γεγονότων μαζί με τις εξαιρέσεις που μπορεί να προκύπτουν από αυτά, οι άλλες εργασίες που γίνονται παράλληλα και στο τέλος του σεναρίου, περιγράφεται ξανά η κατάσταση του συστήματος.

Στη συνέχεια, εικονίζονται ενδεικτικά μερικά από τα σενάρια που προκύπτουν από τις απαιτήσεις του συστήματος.

Σενάριο 1^ο: Εισαγωγή χρήστη στην ιστοσελίδα



Σχήμα 9: Σενάριο εισαγωγής χρήστη

Σενάριο 2ο: Προβολή τμημάτων

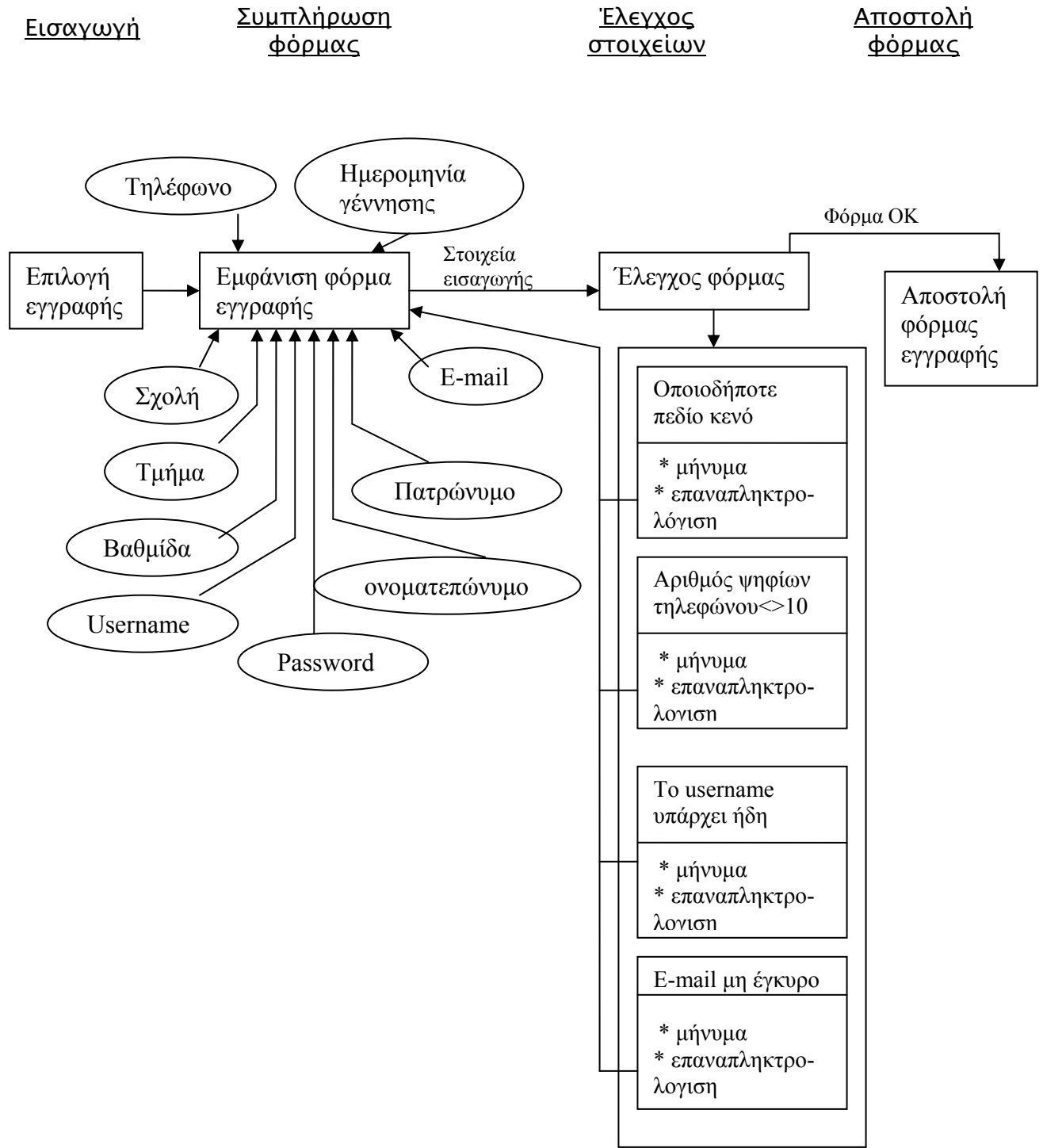
Επιλογή εξαμήνου & μαθήματος

Προβολή τμημάτων



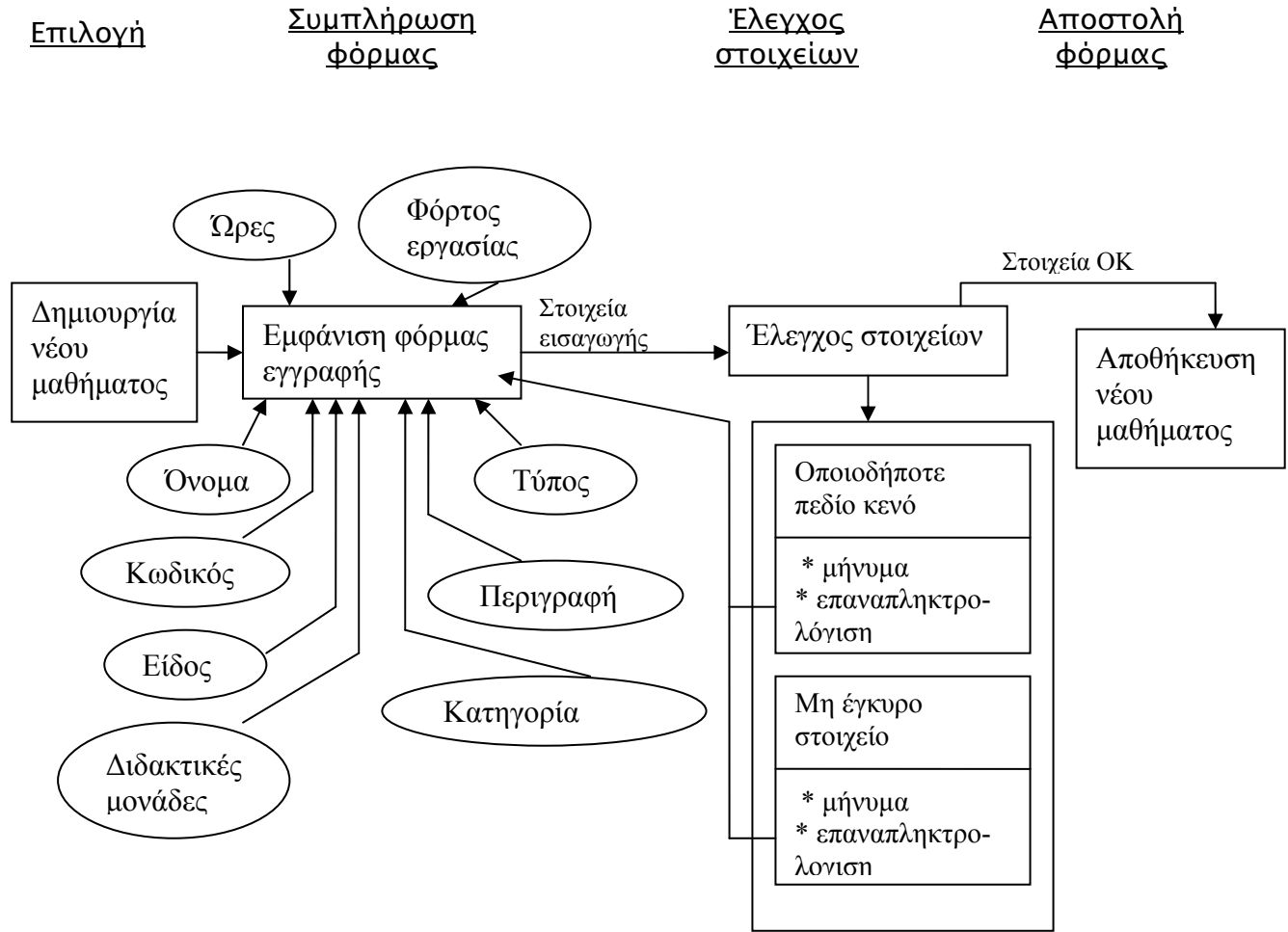
Σχήμα 10: Σενάριο προβολής τμημάτων βάση επιλογής συγκεκριμένου εξαμήνου και μαθήματος

Σενάριο 3^ο: Εγγραφή νέου χρήστη



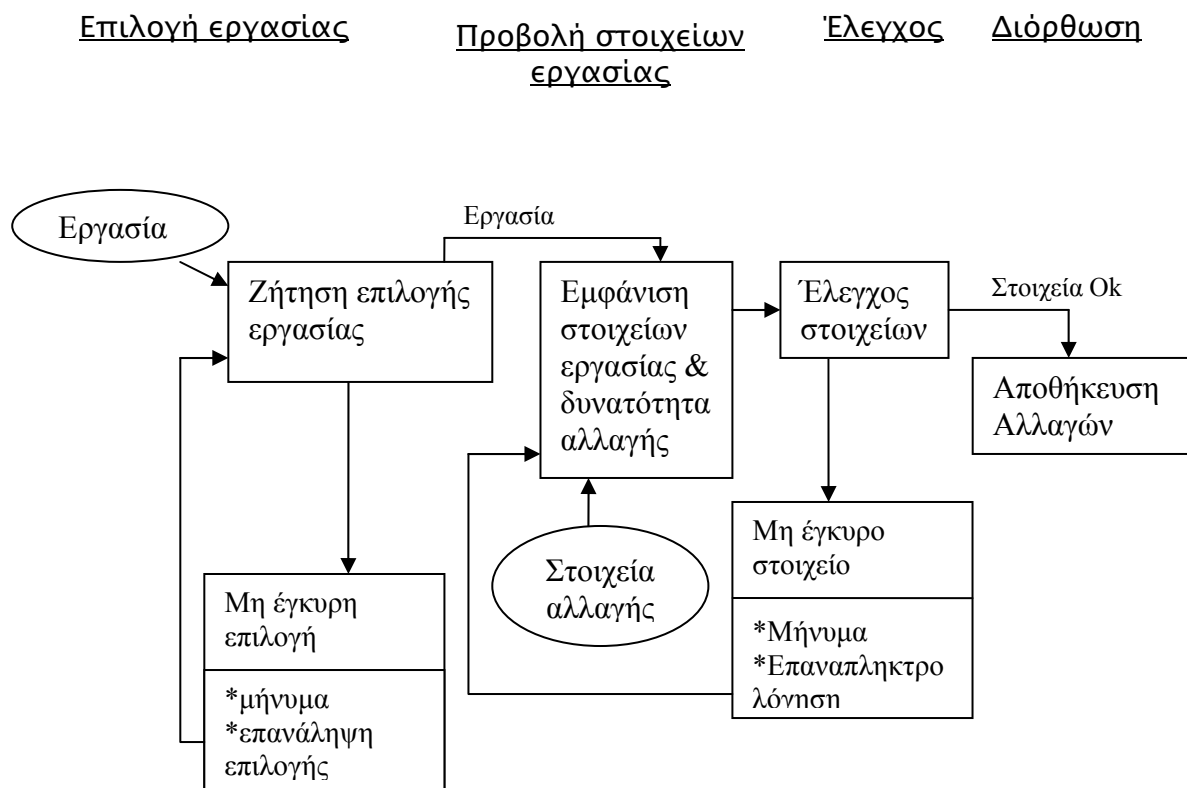
Σχήμα 11: Σενάριο συμπλήρωσης και αποστολής αίτησης εγγραφής νέου χρήστη

Σενάριο 4^ο: Δημιουργία νέου μαθήματος



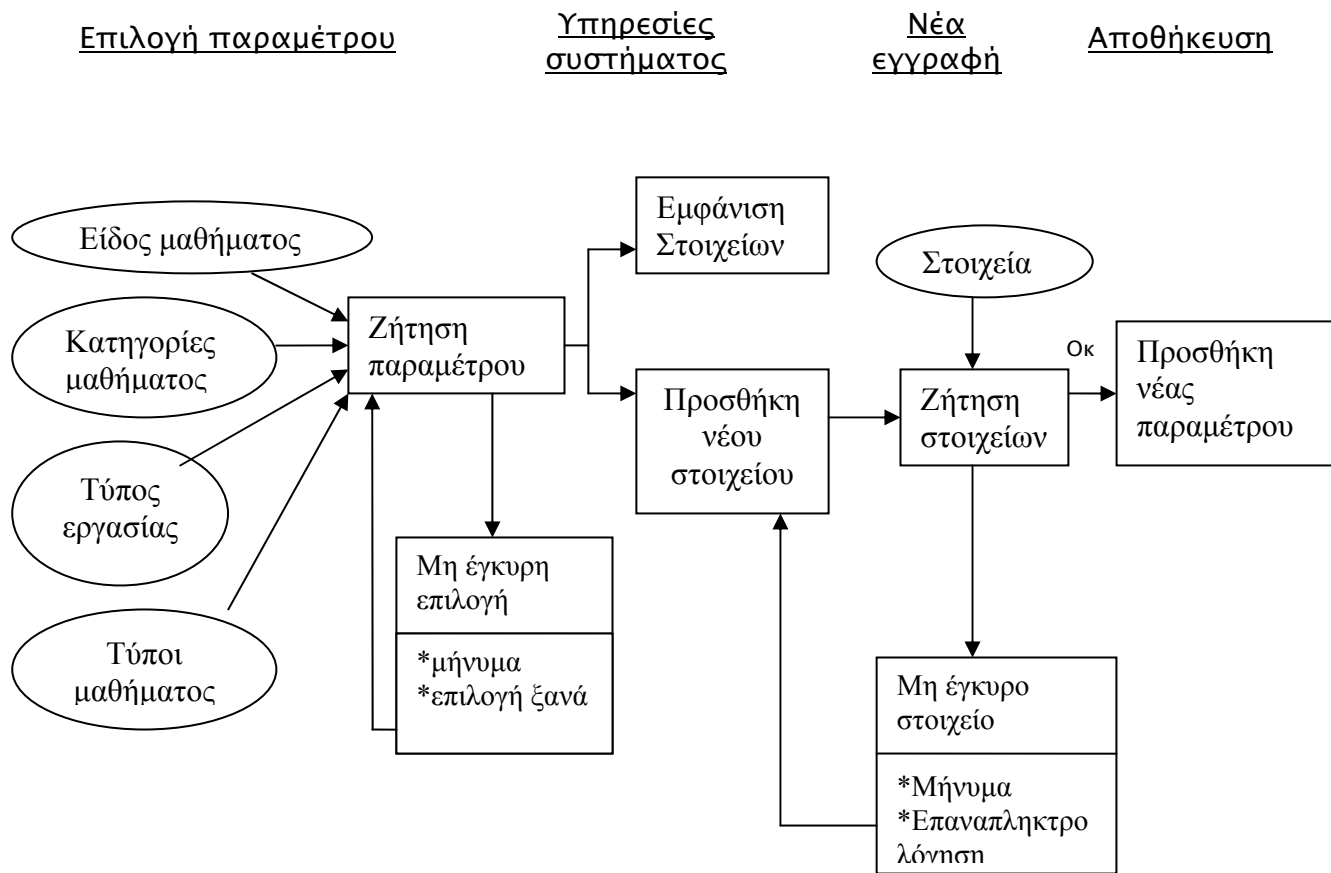
Σχήμα 12: Σενάριο δημιουργίας νέου μαθήματος

Σενάριο 5^ο: Διόρθωση εργασίας



Σχήμα 13: Σενάριο διόρθωσης στοιχείων εργασίας

Σενάριο 6ο: Διαχείριση παραμέτρων συστήματος



Σχήμα 14: Σενάριο διαχείρισης των παραμέτρων του συστήματος

4. Σχεδιασμός Συστήματος

Εφόσον έχουν προσδιοριστεί πλήρως οι απαιτήσεις της ιστοσελίδας, μπορεί να ακολουθήσει η διαδικασία του σχεδιασμού. Οι δραστηριότητες της διαδικασίας αυτής είναι:

- Ο αρχιτεκτονικός σχεδιασμός, ο οποίος περιλαμβάνει πρωτόκολλα επικοινωνίας, αρχιτεκτονικές δικτύου, κτλ.
- Ο αφηρημένος προσδιορισμός, που σημαίνει χωρισμός του προγράμματος σε διάφορους τομείς/τμήματα.
- Ο σχεδιασμός διεπιφάνειας, δηλαδή ο σχεδιασμός της διεπαφής μεταξύ χρήστη και προγράμματος.
- Ο σχεδιασμός κομματιών, ο οποίος περιλαμβάνει τη διαδικασία της σχεδίασης των τμημάτων του προγράμματος και στη συνέχεια τον συνδυασμό αυτών.
- Ο σχεδιασμός δομών δεδομένων, συμπεριλαμβανομένου όλες τις λειτουργίες του απαιτεί το πρόγραμμα.
- Ο σχεδιασμός αλγορίθμων.

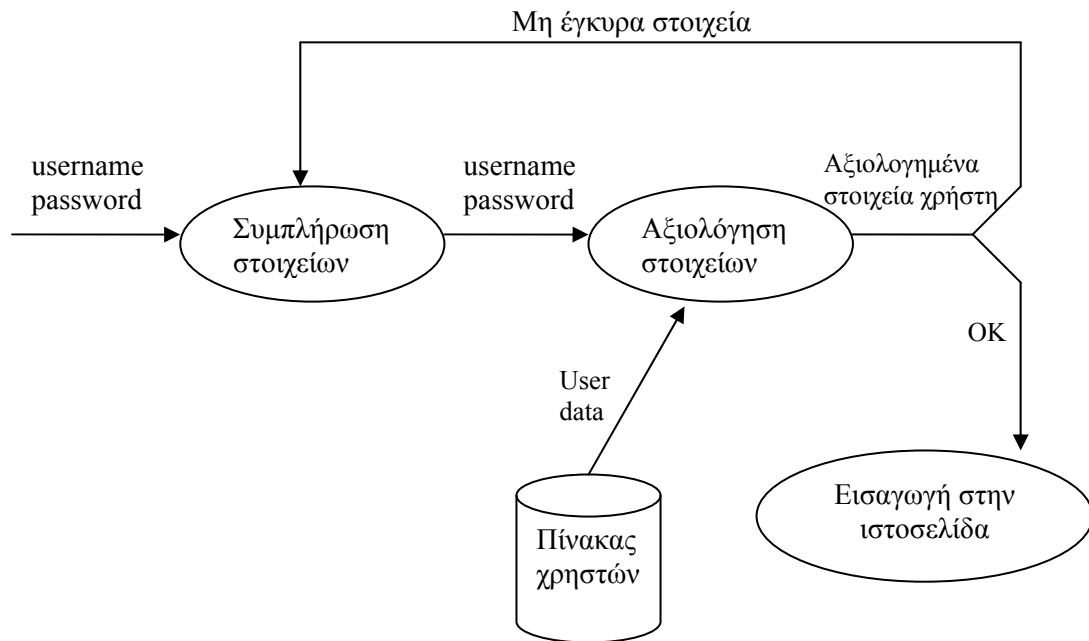
Ο σχεδιασμός συνήθως τεκμηριώνετε με διάφορα γραφικά μοντέλα, εκ των οποίων τα πιο διαδεδομένα είναι το Μοντέλο Ροής Δεδομένων (Data Flow Diagram model) και το Μοντέλο Δεδομένων (Data Model).

4.1 Διαγράμματα ροής δεδομένων (D.F.D)

Τα Διαγράμματα Data Flow ανήκουν στη κατηγορία των μοντέλων Data Processing και ένα γενικό χαρακτηριστικό της χρησιμότητάς τους είναι ότι παρουσιάζουν το σύστημα από λειτουργικής άποψης. Αναλυτικότερα, εντοπίζουν και τεκμηριώνουν τον τρόπο με τον οποίο τα δεδομένα που αφορούν μια διαδικασία μπορούν να βοηθήσουν στην κατανόηση του συστήματος. Παράλληλα, προσδιορίζουν την ανταλλαγή δεδομένων μεταξύ διαφορετικών συστημάτων.

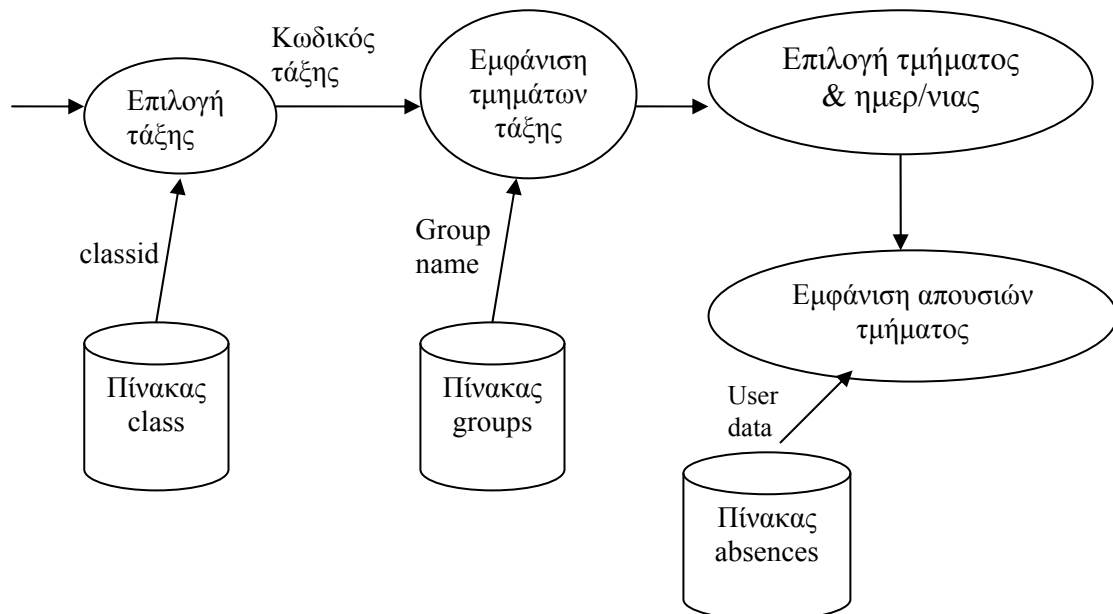
Στα παρακάτω σχήματα, απεικονίζονται ενδεικτικά έξι από τα βασικότερα διαγράμματα ροής δεδομένων (Data Flow Diagrams) που περιγράφουν κεντρικές λειτουργίες του συστήματος.

D.F.D 1^ο: Είσοδος ενός χρήστη στην ιστοσελίδα (Log In)



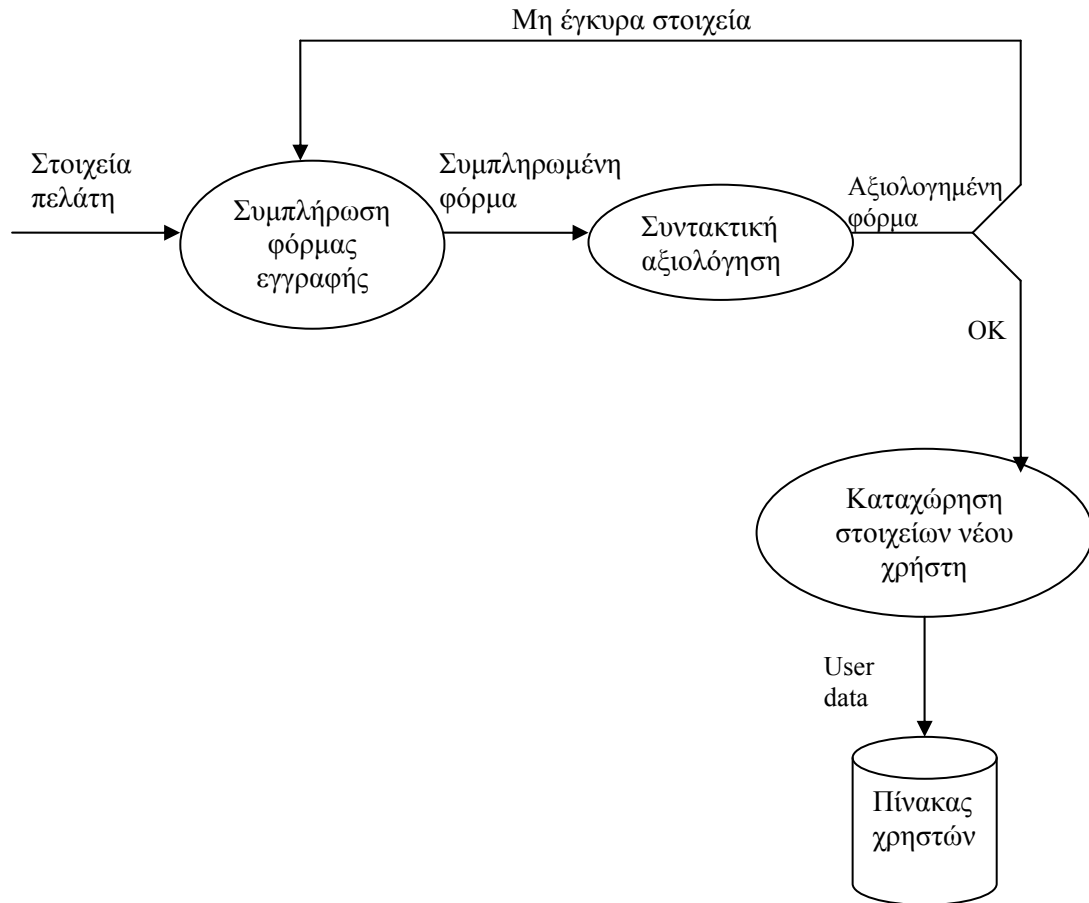
Σχήμα 15: D.F.D εισαγωγής χρήστη στην ιστοσελίδα

D.F.D 2^ο: Προβολή απουσιών ανά τμήμα



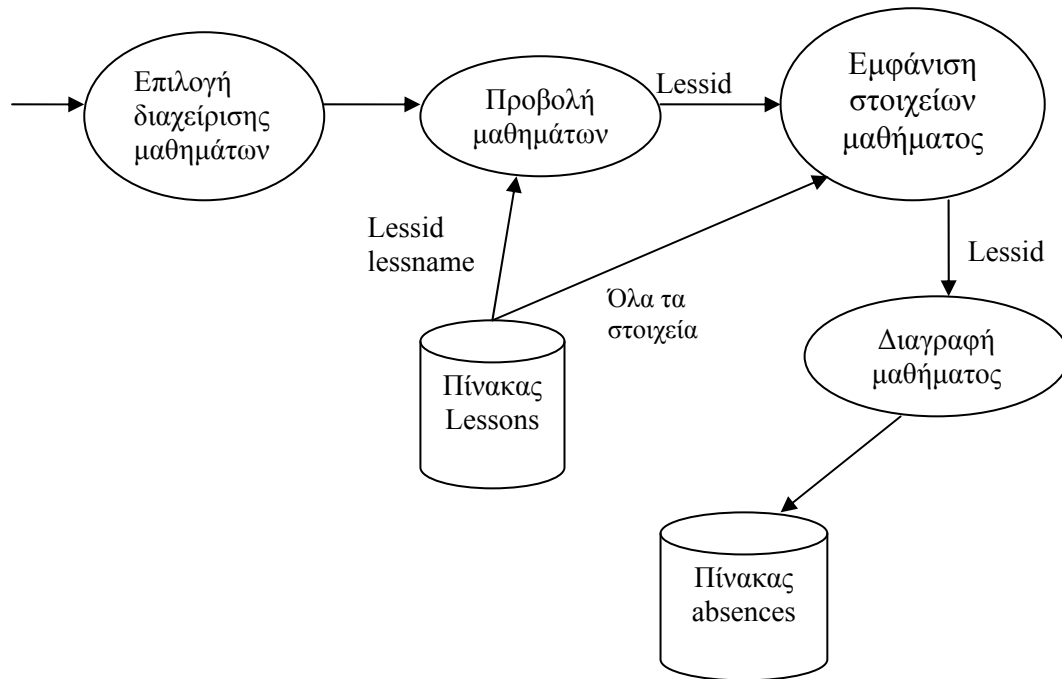
Σχήμα 16: D.F.D προβολής απουσιών

D.F.D 3^ο: Εγγραφή νέου χρήστη-καθηγητή



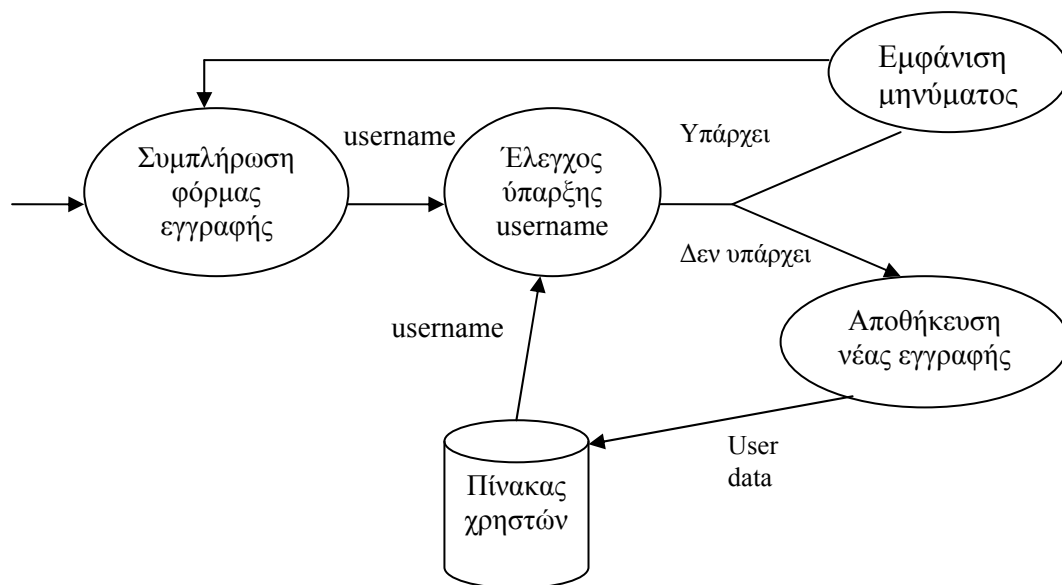
Σχήμα 17: D.F.D εγγραφής νέου χρήστη-καθηγητή

D.F.D 4^ο: Διαγραφής μαθήματος

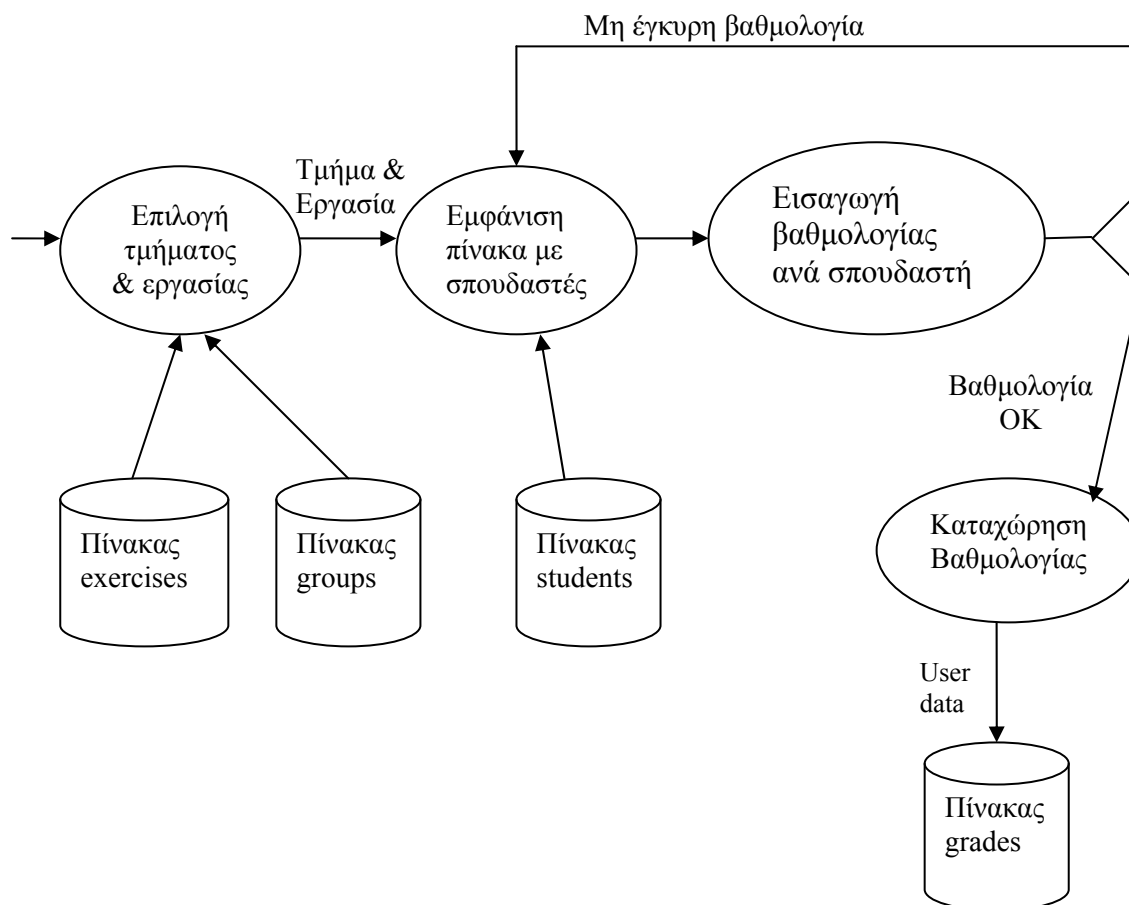


Σχήμα 18: D.F.D διαγραφής μαθήματος

D.F.D 5^ο: Έλεγχος ύπαρξης username στη Β.Δ. κατά τη νέα εγγραφή



Σχήμα 19: D.F.D ελέγχου ύπαρξης username στη βάση δεδομένων κατά τη διαδικασία της νέας εγγραφής

D.F.D 6^ο: Καταχώρηση ατομικής βαθμολογίας

Σχήμα 20: D.F.D καταχώρησης ατομικής βαθμολογίας

4.2 Σχεδιασμός Data Model

Το Data Model είναι ένα απεικονιστικό μοντέλο που περιγράφει πως τα δεδομένα εμφανίζονται και χρησιμοποιούνται. Ο όρος Data Model έχει δύο γενικά αποδεκτές έννοιες:

1. Μία θεωρία Data Model. Για παράδειγμα, μια τυπική περιγραφή του τρόπου με τον οποίο τα δεδομένα δομούνται και χρησιμοποιούνται.
2. Μία υπόδειξη Data Model. Για παράδειγμα, εφαρμόζοντας μία θεωρία Data Model για να δημιουργήσουμε ένα πρακτικό παράδειγμα για κάποιες συγκεκριμένες εφαρμογές.

4.2.1 Λίστα δεδομένων ανά απαίτηση

Εισαγωγή χρήστη:

Username

Password

Εγγραφή νέου χρήστη / καθηγητή:

Professor id

Professor name

Professor birthday

Professor mail

Professor telephone

Professor bathmida

Professor sxoli

Professor tmima

Username

Password

Διαχείριση εξαμήνου:

Semester id

Semester name

Semester description

Semester start date

Semester end date

Semester start examinations A

Semester end examinations A

Semester start examinations B

Semester end examinations B

Holiday id

Holiday date

Holiday description

Διαχείριση μαθημάτων:

Lesson id

Lesson name

Lesson code

Lesson description

Lesson type

Lesson category

Lesson eidos

Lesson hours

Lesson fortos ergasias
Lesson didaktikes monades

Διαχείριση τμημάτων:

Group id
Group name
Group day
Group time
Group capacity
Group room
Group team
Group teacher A
Group teacher B
Semester id
Lesson id

Διαχείριση ομάδων εργασίας:

Team id
Team name
Team students
Group id

Διαχείριση εργασιών / project:

Exercise id
Exercise title
Exercise description
Exercise type
Exercise percent
Lesson id

Διαχείριση βαθμολογίας:

Student am
Semester id
Lesson id
Group id
Exercise id
Professor id
grade

Διαχείριση απουσιολογίου:

Absence date

Group id
Student am
Absence check

4.2.2 Έλεγχος επαναλαμβανόμενων δεδομένων

Από τις παραπάνω λίστες παρατηρούμε ότι δεδομένα όπως: username, password, professor id, semester id, lesson id, group id και exercise id, επαναλαμβάνονται σε διάφορες ενότητες. Κάθε ένα όμως από αυτά θα διατηρείται σε έναν μόνο πίνακα. Δηλαδή, το username, το password όπως και το professor id θα υπάρχει στον πίνακα των καθηγητών-χρηστών “Professors”. Το semester id θα υπάρχει στον πίνακα των εξαμήνων “Semester”, το lesson id στον πίνακα μαθημάτων “Lessons”, το group id στον πίνακα τμημάτων “Groups” και τέλος, το exercise id θα υπάρχει στον πίνακα των εργασιών “Exercise”.

4.2.3 Ομαδοποίηση δεδομένων σε πίνακες

Οι πίνακες που ξεκάθαρα έχουν προκύψει από τις λίστες των δεδομένων ανά απαίτηση και από τον έλεγχο των επαναλαμβανόμενων δεδομένων είναι οι ακόλουθοι: Professors, Semester, lessons, groups, team, exercise, grades, absences και holidays.

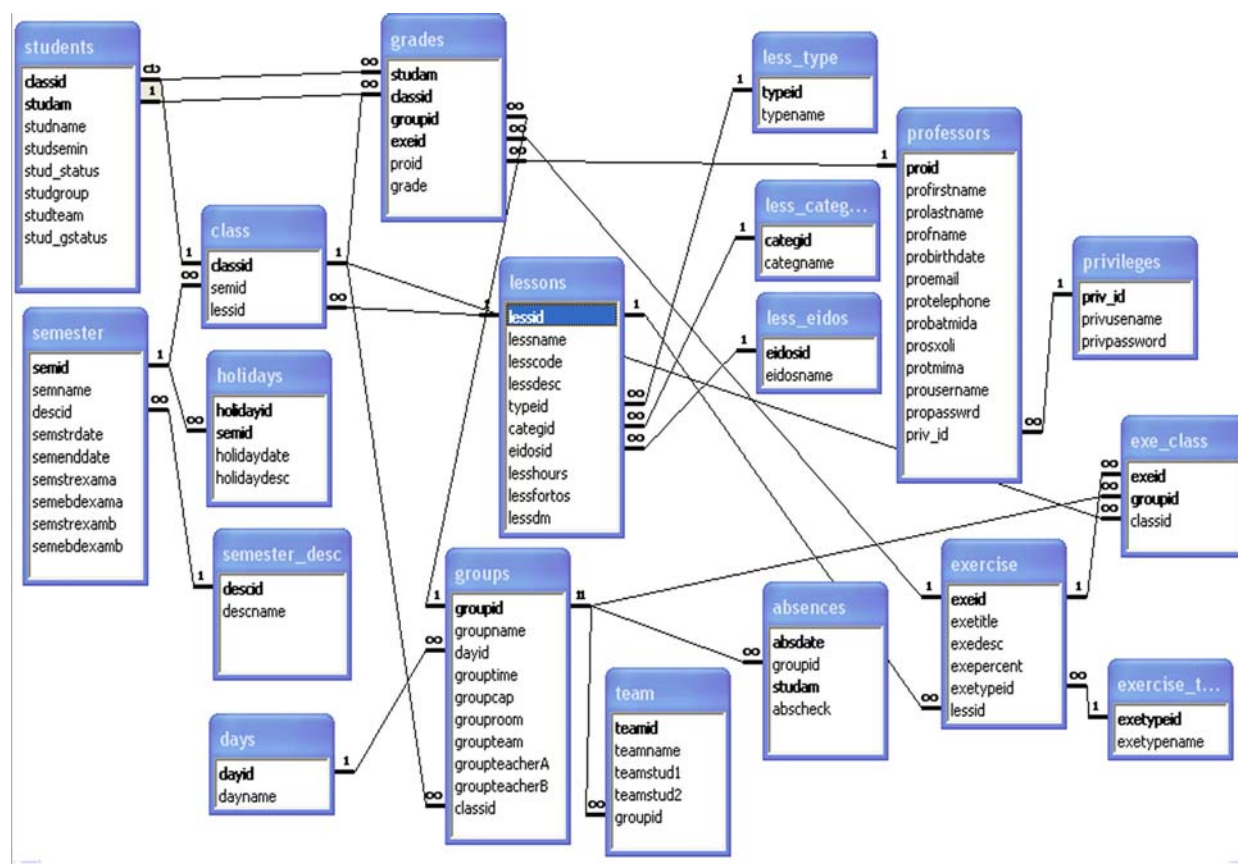
Ο συνδυασμός του κωδικού του εξαμήνου με τον κωδικό του μαθήματος προσδιορίζει την έννοια της τάξης, συνεπώς προκύπτει άλλος ένας πίνακας “Class” με πεδία το class id ως πρωτεύον κλειδί, το semester id και το lesson id. Επίσης αφού σε πολλούς πίνακες χρειαζόμαστε τα ονόματα και τα Α.Μ. των σπουδαστών θα πρέπει να φτιάξουμε και έναν πίνακα σπουδαστών “Students” ο οποίος θα ενημερώνεται κάθε εξάμηνο για τους νέους σπουδαστές που γράφτηκαν στα εργαστηριακά μαθήματα. Ο πίνακας “Students” θα περιέχει όλα τα στοιχεία που αφορούν τον σπουδαστή, για παράδειγμα: ονοματεπώνυμο, αριθμό μητρώου, εξάμηνο εισαγωγής, τμήμα και ομάδα στην οποία ανήκει, κ.τ.λ.

Για την τελειοποίηση του σχεδιασμού της βάσης δεδομένων, θα χρειαστούμε ακόμα έξι πίνακες, οι οποίοι θα περιλαμβάνουν βασικούς παραμέτρους του συστήματος. Οι πίνακες αυτοί είναι:

1. Ο πίνακας “Days”, που θα περιλαμβάνει τις ημέρες της εβδομάδας με σκοπό στο σύστημα να εμφανίζονται δυναμικά, μέσω μιας λίστας.

2. Ο πίνακας “ Semester_desc”, που θα περιέχει τις δύο τιμές που περιγράφουν το εξάμηνο, Χειμερινό και Εαρινό.
3. Ο πίνακας “Less_category”, που θα περιλαμβάνει τις κατηγορίες του μαθήματος, όπως για παράδειγμα: Μάθημα γενικής υποδομής, Μάθημα ειδικότητας, κ.τ.λ.
4. Ο πίνακας “Less_tyre”, που θα περιλαμβάνει τους τύπους του μαθήματος, όπως για παράδειγμα: Εργαστήριο, Θεωρία και Φροντιστήριο.
5. Ο πίνακας “Less_eidos”, που θα περιλαμβάνει τα είδη του μαθήματος, όπως για παράδειγμα: Υποχρεωτικό, Προαιρετικό και Επιλογής Υποχρεωτικό κ.τ.λ.
6. Ο πίνακας “Exercise_type”, που θα περιλαμβάνει τους τύπους των εργασιών, όπως για παράδειγμα: Εβδομαδιαία εργασία, Μηνιαία εργασία, Project εξαμήνου και Project κατοχύρωσης.

4.2.4 Διαγραμματική απεικόνιση Data Model



Σχήμα 21: Data Model

5. Υλοποίηση Συστήματος

Μετά από την πλήρη σχεδίαση της αρχιτεκτονικής του έργου, ακολουθεί η τμηματική υλοποίηση αυτού. Τα δύο βασικά κατασκευαστικά μέρη του έργου μας είναι η ασφαλής δυναμική βάση δεδομένων, καθώς επίσης και η λειτουργικότητα σε συνδυασμό με τη διεπαφή της ιστοσελίδας.

5.1 Υλοποίηση της βάσης δεδομένων

Χρησιμοποιώντας τα εργαλεία της MySQL και το γενικό πλάνο της βάσης δεδομένων από το σχεδιασμό, προχωράμε στην υλοποίηση αυτής.

5.1.1 Πίνακες

Το έργο «Ηλεκτρονική Διαχείριση Εργαστηρίου» απαιτεί για τη σωστή λειτουργία του δεκαεννιά (19) πίνακες, εκ των οποίων έντεκα (11) είναι βασικοί, δύο (2) ενδιάμεσοι και έξι (6) βοηθητικοί.

Αναλυτικότερα, τα ονόματα των Βασικών Πινάκων είναι:

- i. Semester
- ii. Lessons
- iii. Groups
- iv. Team
- v. Exercise
- vi. Holidays
- vii. Grades
- viii. Absences
- ix. Professors
- x. Privileges
- xi. Students

Τα ονόματα των ενδιάμεσων πινάκων είναι:

- i. Class
- ii. Exe_class

Τα ονόματα των βοηθητικών πινάκων είναι:

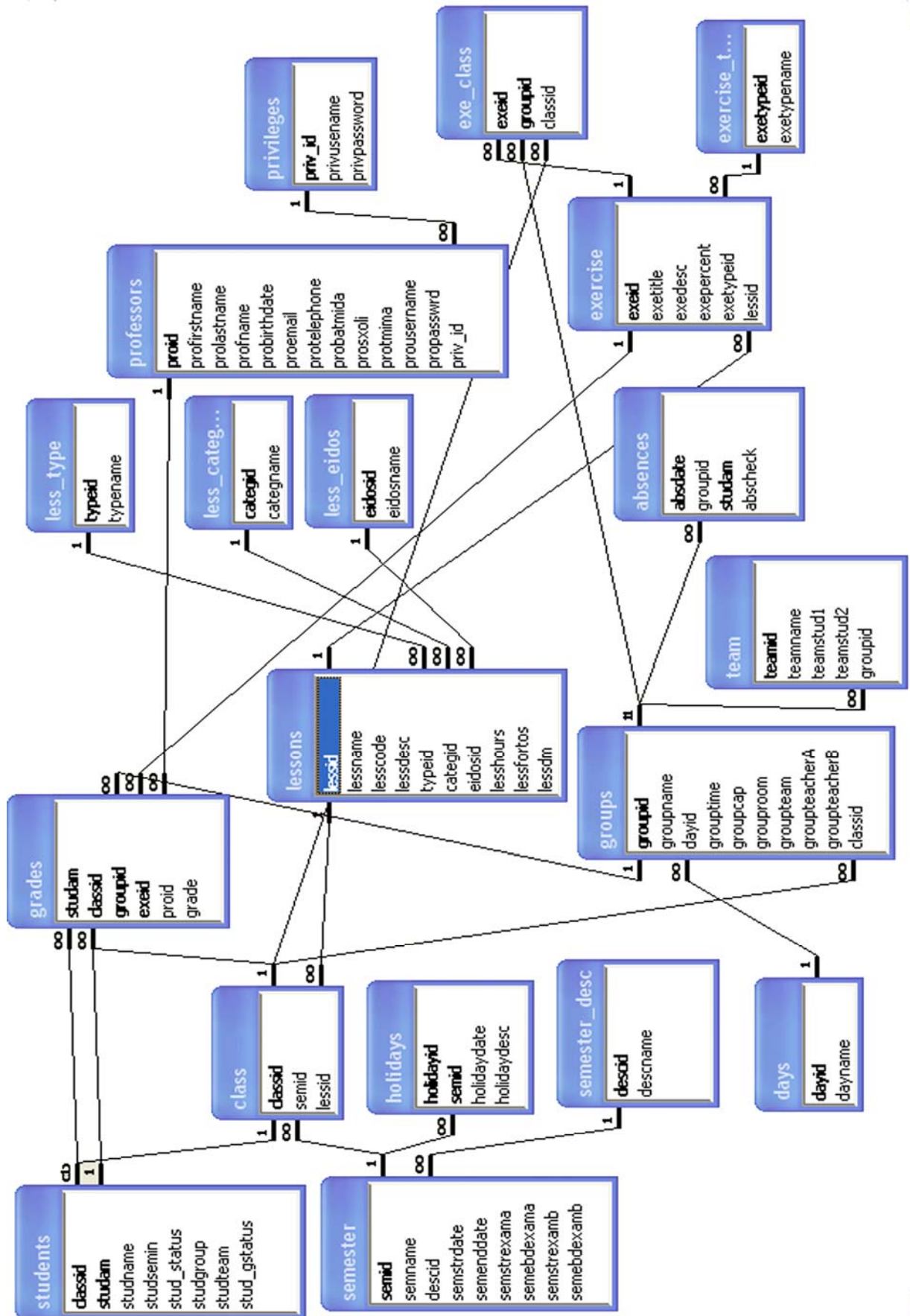
- i. Days
- ii. Semester_desc

- iii. Less_category
- iv. Less_type
- v. Less_eidos
- vi. Exercise_type

Ο κώδικας του sql αρχείου που δημιούργησε τους παραπάνω πίνακες, χωρίς φυσικά τις εγγραφές, βρίσκεται στο [Παράρτημα Α](#).

5.1.2 Σχέσεις πινάκων

Οι σχέσεις μεταξύ των πινάκων που αποτελούν τη βάση δεδομένων eLabManage φαίνονται στο παρακάτω σχήμα:



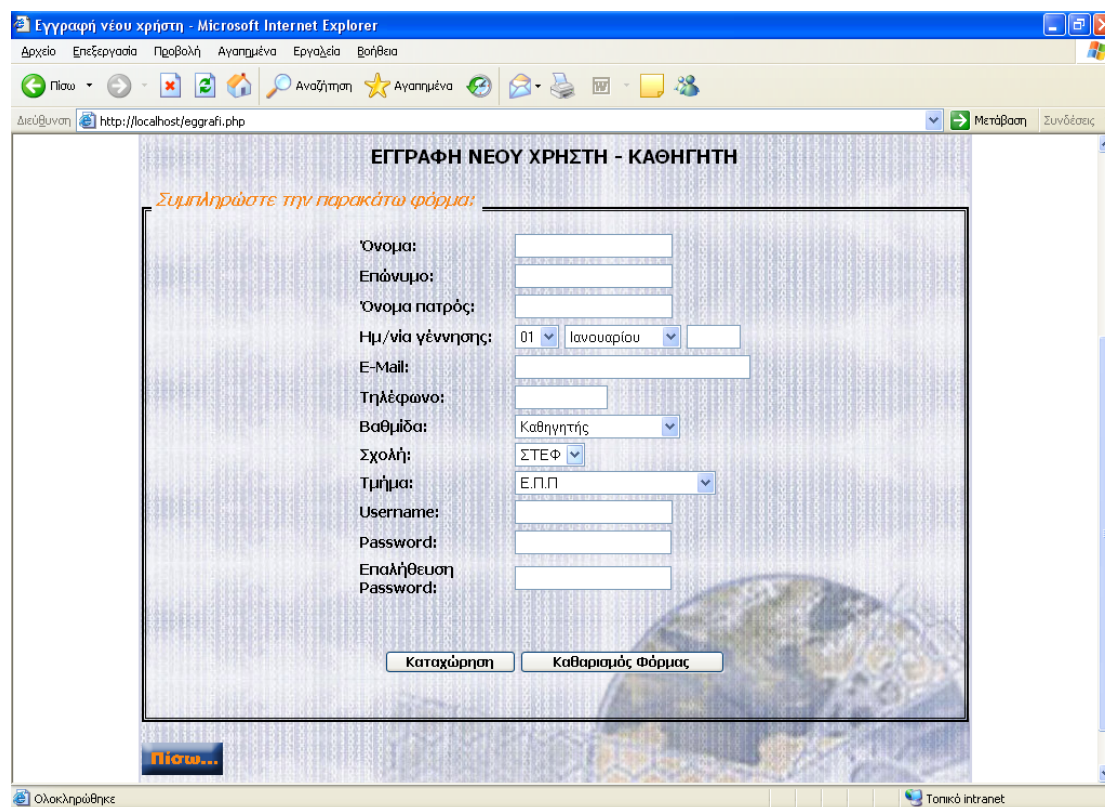
Σχήμα 22: Σχέσεις πινάκων βάσης δεδομένων

5.2 Υλοποίηση ιστοσελίδας

Ένα χρήσιμο εργαλείο για τη δημιουργία μιας ιστοσελίδας και συγκεκριμένα της διεπαφής αυτής, είναι το Dreamweaver. Με τη χρήση, λοιπόν, αυτού του προγράμματος υλοποιήσαμε τις λειτουργίες του συστήματος βάση των απαιτήσεων που έχουμε καταγράψει. Στην συνέχεια του κεφαλαίου αναλύουμε την υλοποίηση των ενοτήτων του συστήματος.

5.2.1 Εγγραφή χρήστη-καθηγητή

Το βασικότερη ενέργεια που πρέπει να κάνει ο χρήστης-καθηγητής που επισκέπτεται για πρώτη φορά την ιστοσελίδα, είναι η εγγραφή του σε αυτό. Η διαδικασία της εγγραφής είναι πολύ απλή και περιορίζεται στην σωστή συμπλήρωση της παρακάτω φόρμας:



The screenshot shows a web browser window titled "Εγγραφή νέου χρήστη - Microsoft Internet Explorer". The address bar shows "http://localhost/eggrafi.php". The main content area displays a registration form titled "ΕΓΓΡΑΦΗ ΝΕΟΥ ΧΡΗΣΤΗ - ΚΑΘΗΓΗΤΗ". The form includes the following fields and options:

- Όνομα: [Text input]
- Επώνυμο: [Text input]
- Όνομα πατρός: [Text input]
- Ημ/νία γέννησης: [Date picker] (01 / Ιανουαρίου / [Year])
- E-Mail: [Text input]
- Τηλέφωνο: [Text input]
- Βαθμίδα: [Dropdown menu] (Καθηγητής)
- Σχολή: [Dropdown menu] (ΣΤΕΦ)
- Τμήμα: [Dropdown menu] (Ε.Π.Π)
- Username: [Text input]
- Password: [Text input]
- Επαλήθευση Password: [Text input]

At the bottom of the form, there are two buttons: "Καταχώρηση" and "Καθαρισμός Φόρμας". A "Πισω..." button is also visible at the bottom left of the form area.

Εικόνα 1: Οθόνη εγγραφής νέου χρήστη-καθηγητή

Τα στοιχεία που εισάγονται στη φόρμα εγγραφής, αφού πρώτα ελεγχθούν, αποθηκεύονται στον πίνακα professors της MySQL βάσης δεδομένων. Για να γίνει οποιαδήποτε εργασία στη βάση δεδομένων,

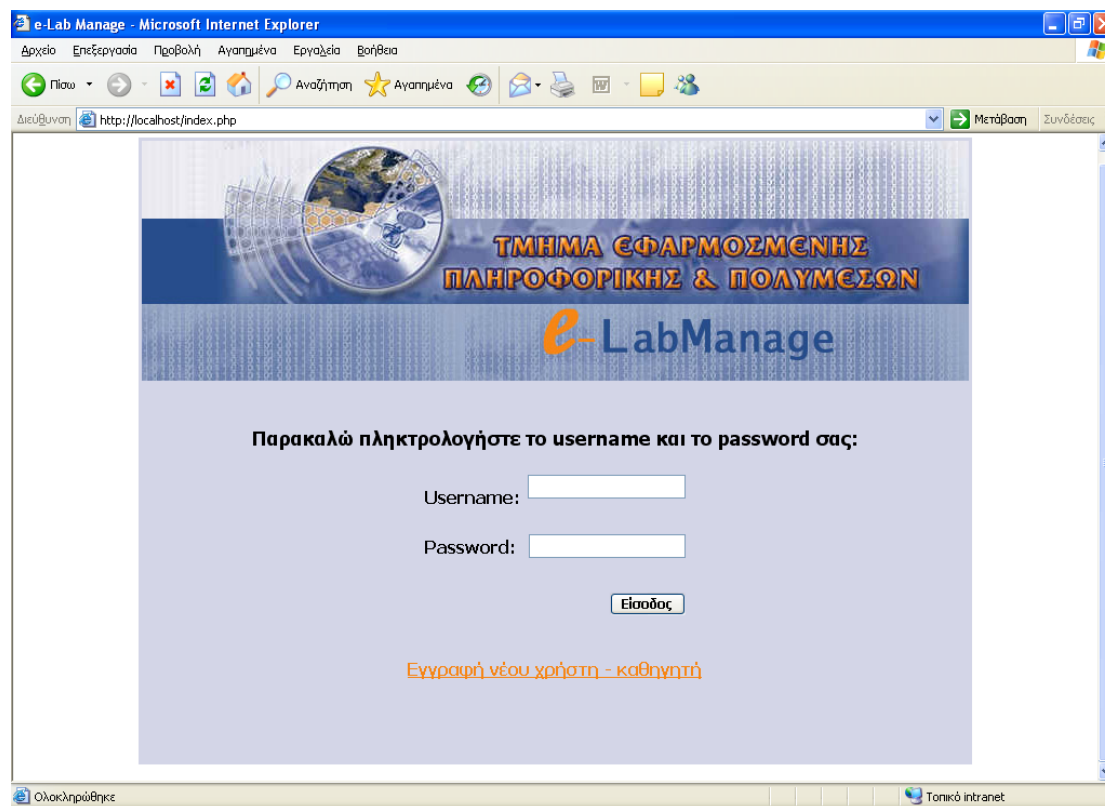
πρέπει πρώτα να γίνει η σύνδεση σε αυτήν. Για την πραγματοποίηση αυτής της σύνδεσης είναι απαραίτητη η παρακάτω γραμμή κώδικα:

```
$db = mysql_connect($host,$username,$password);
```

Για μεγαλύτερη ασφάλεια των προσωπικών δεδομένων του χρήστη-καθηγητή, χρησιμοποιούμε τη συνάρτηση `crypt('password')` για να κρυπτογραφήσουμε το password πριν προβούμε στην αποθήκευση αυτού στη βάση δεδομένων.

5.2.2 Εισαγωγή χρήστη

Το πρώτο πράγμα που ζητείται από το χρήστη όταν μπαίνει στην ιστοσελίδα Ηλεκτρονικής Διαχείρισης Εργαστηρίων, όπως και στα περισσότερα site αυτού του είδους, είναι να πιστοποιήσει την ταυτότητά του με τη χρήση του μοναδικού ονόματος (username) και του κωδικού πρόσβασης (password) που επέλεξε κατά την εγγραφή του.



Εικόνα 2: Οθόνη εισαγωγής χρήστη (Log in)

Στη συνέχεια, το σύστημα ελέγχει αν το username και το password που έδωσε ο χρήστης είναι σωστά και αν αντιστοιχούν σε κάποια εγγραφή στον πίνακα των χρηστών-καθηγητών. Αν υπάρχουν στον πίνακα “professors”, του επιτρέπει να μπει στην κεντρική σελίδα του site αλλιώς, τον παραπέμπει να ξαναπροσπαθήσει. Ο έλεγχος αυτός πραγματοποιείται με το παρακάτω κώδικα php:

```
$query = "select * from professors where pusername='$logname' and
propasswrд='$logpass'";
$result = mysql_query($query);
$pin=mysql_fetch_array($result);
$privilege=$pin["priv_id"];

if (!$result)
    echo '<center>Υπάρχει Σφάλμα!</center>';
if (mysql_num_rows($result)>0)
{
    echo '<p align="left" class="style3"><b>ΚΑΛΩΣ ΟΡΙΣΕΣ
'.$pin["pusername"].' </b></p><br>';
    echo '<center><b><a href="Start.php">Πάτησε εδώ </a>για να μπείς
στην αρχική σελίδα του e-LabManage.</b><br></center>';
    $_SESSION['userid']=stripslashes($pin["proid"]);
}
else
    echo '<center><h4>Ο συνδυασμός username και password δεν είναι
σωστός.</h4></center>';

}
```

Με την εντολή “\$_SESSION['userid'] = stripslashes(\$pin["proid"]);” γίνεται η αποθήκευση του μοναδικού κωδικού του χρήστη (id) σε μία γενική μεταβλητή \$_SESSION, έτσι ώστε να είναι διαθέσιμη οποιαδήποτε άλλη στιγμή κατά την περιήγηση του χρήστη στην ιστοσελίδα.

5.2.3 Διαχείριση εξαμήνου

Το πρόγραμμα δίνει τη δυνατότητα στο χρήστη να διαχειριστεί τα εξάμηνα που υπάρχουν, να δημιουργήσει νέα, να κάνει αλλαγές στα ήδη υπάρχοντα αλλά αποκλείει τη δυνατότητα διαγραφής αυτών.

Προεπιλεγμένο εξάμηνο για διαχείριση από το χρήστη είναι το τρέχον, δηλαδή το τελευταίο που έχει δημιουργηθεί. Μπορεί όμως ο χρήστης να επιλέξει οποιοδήποτε άλλο εξάμηνο για να επεξεργαστεί. Για κάθε

επιλεγμένο εξάμηνο προβάλλονται αναλυτικά όλα τα στοιχεία του, όπως κωδικός, όνομα, περιγραφή, ημερομηνίες έναρξης και λήξης εξαμήνου, κ.τ.λ. Τα στοιχεία αυτά εμφανίζονται με σκοπό να διορθωθούν από το χρήστη ή απλά να διαβαστούν.

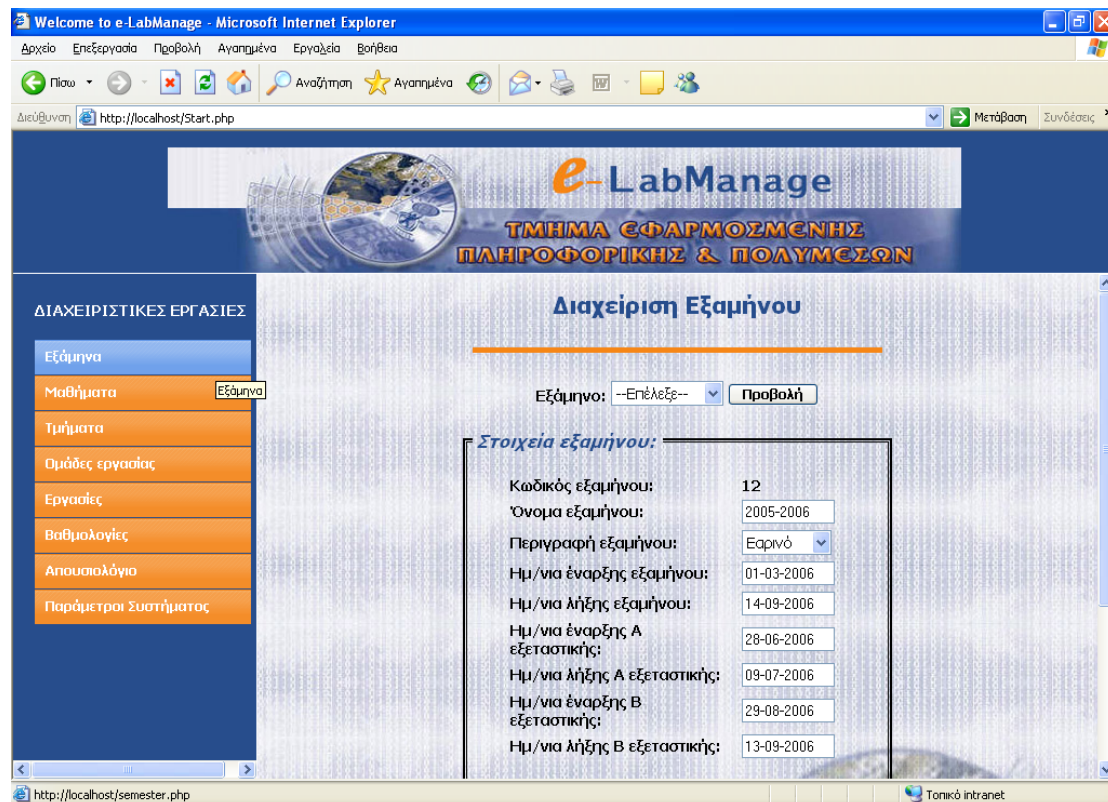
Ο ερώτημα (query) που επιλέξει τα κατάλληλα στοιχεία από τον πίνακα 'semester' για να προβληθούν είναι:

```
$query1="select * from semester where semid = '$edit'";
$result=mysql_query($query1);
$semester=mysql_fetch_array($result);
```

όπου στη μεταβλητή \$edit είναι αποθηκευμένο το id του εξαμήνου που έχει επιλέξει ο χρήστης, και το \$semester είναι ένας πίνακας που περιέχει τα στοιχεία της γραμμής του πίνακα 'semester' που προέκυψε από την εκτέλεση του ερωτήματος.

Τα στοιχεία αυτά του πίνακα \$semester εμφανίζονται ένα ένα σε text fields με την παρακάτω γραμμή κώδικα:

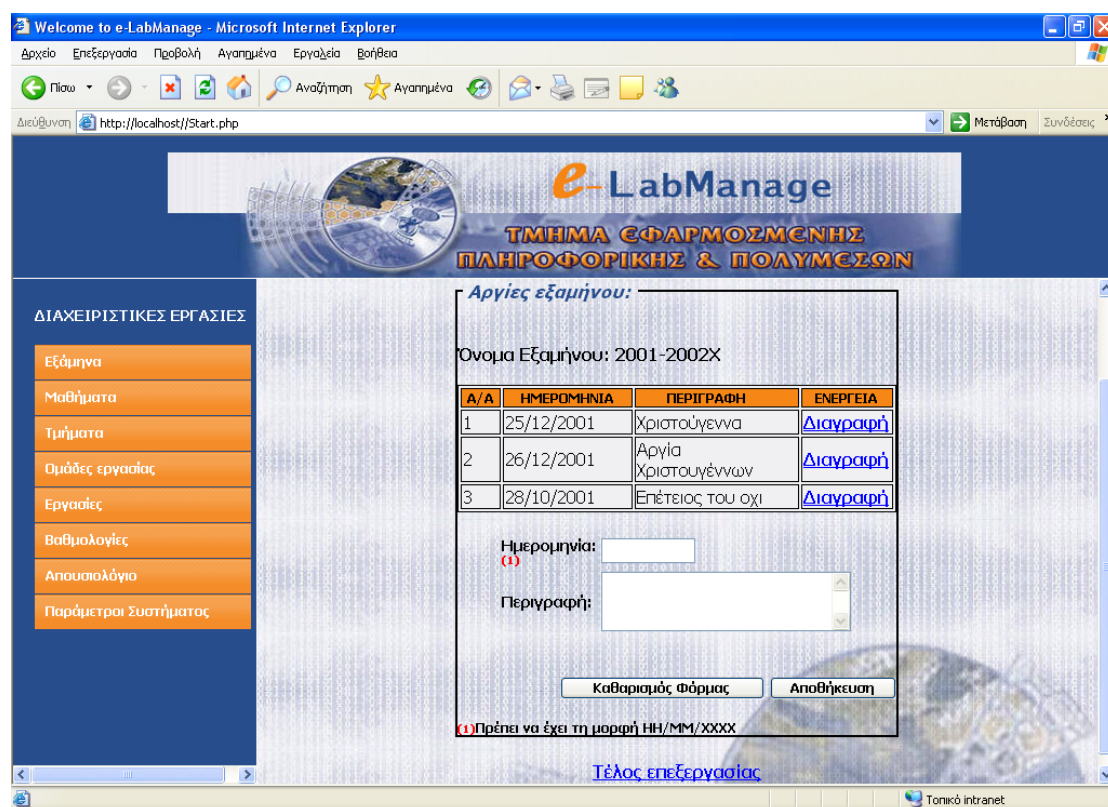
```
<input name="sem3" type="text" size="10" maxlength="10" value="<? echo ($semester["semstrdate"]); ?>" />
```



Εικόνα 3: Οθόνη διαχείρισης εξαμήνου

Ο χρήστης επίσης, μπορεί πολύ εύκολα να δημιουργήσει ένα νέο εξαμήνο συμπληρώνοντας απλά τα στοιχεία του νέου εξαμήνου στη σχετική φόρμα.

Η έννοια της διαχείρισης του εξαμήνου περιλαμβάνει επίσης και τον καθορισμό των αργιών ανά εξαμήνο, τις ημέρες δηλαδή όπου η σχολή θα είναι κλειστή και συνεπώς δεν θα διεξαχθούν μαθήματα. Η διαχείριση των αργιών κάθε εξαμήνου, εκτός από την προσθήκη νέων αργιών, δίνει επίσης τη δυνατότητα στο χρήστη να διαγράψει τις αργίες που πιθανότατα έχουν καταχωρηθεί λάθος. Η διεπαφή της παραπάνω διαδικασίας φαίνεται στην παρακάτω εικόνα:



Εικόνα 4: Οθόνη διαχείρισης αργιών εξαμήνου

Κάτι άλλο που αξίζει να σημειώσουμε για την υλοποίηση της διαχείριση των αργιών είναι η επανάληψη της σύνδεσης (link) 'Διαγραφή' στον πίνακα προβολής των ήδη καταχωρημένων αργιών. Η σύνδεση αυτή επειδή μεταφέρει μαζί την τιμή μιας μεταβλητής η οποία είναι διαφορετική για κάθε μία από αυτές, επιλέξαμε να υλοποιείται μέσω μιας συνάρτησης με όνομα `do_html_URL()` της οποίας ο κώδικας είναι:

```
<?
function do_html_URL($url)
{
?>
<a href="<?=$url?>">Διαγραφή</a>
```

```
<?
}
?>
```

Η συνάρτηση αυτή καλείται μέσα από τον βρόγχο της επανάληψης που δημιουργεί τον πίνακα, με τις ακόλουθες γραμμές:

```
$url="removeargia.php?id=".$row["holidayid"];
do_html_URL($url);
```

Η μεταβλητής \$_GET[id] είναι αυτή που θα μεταφερθεί μέσω της σύνδεσης στο αρχείο removeargia.php και η τιμή της θα είναι ή ίδια με της \$row["holidayid"].

5.2.4 Διαχείριση μαθημάτων

Ένα άλλο τμήμα της Ηλεκτρονικής Διαχείρισης Εργαστηρίων είναι τα μαθήματα. Η βασική σελίδα των μαθημάτων περιλαμβάνει έναν πίνακα με όλα τα μαθήματα που υπάρχουν στον πίνακα 'lessons' της βάση δεδομένων, καθώς επίσης και τη δυνατότητα προβολής περισσότερων λεπτομερειών για κάθε μάθημα ξεχωριστά, εφ' όσον βέβαια το επιλέξει ο χρήστης. Στην εμφάνιση αυτών των λεπτομερειών, το πρόγραμμα δίνει τη δυνατότητα διόρθωσης αυτών αλλά και διαγραφής του μαθήματος.

Για την υλοποίηση της διόρθωσης του μαθήματος χρησιμοποιείται ο παρακάτω κώδικας:

```
<?php
...
include("connectdb.php");
```

```
$query = "UPDATE lessons SET lessname='$name', lesscode='$code',
lessdesc='$desc', lesstype='$type', lesscategory='$category', lesseidos='$eidos',
lesshours='$hours', lessfortos='$fortos', lessdm='$dm' WHERE lessid='$idless' ";
```

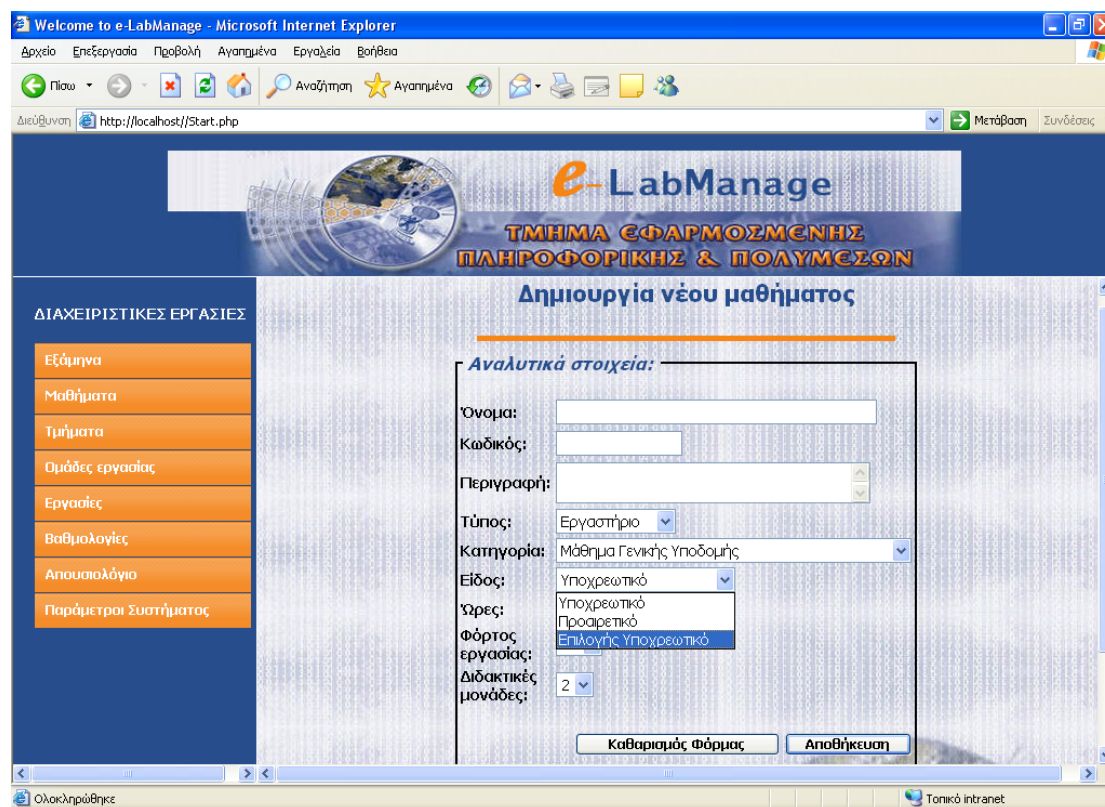
```
$result = mysql_query($query);
if ($result)
{
    echo '<h3>Η ΔΙΟΡΘΩΣΗ ΠΡΑΓΜΑΤΟΠΟΙΗΘΗΚΕ ΕΠΙΤΥΧΩΣ!</h3><br>';
    ...
}
...
?>
```

Η εντολή include() επιτρέπει την ενσωμάτωση αρχείων σε PHP έγγραφα. Ο PHP κώδικας που περιέχουν αυτά τα αρχεία μπορεί να

εκτελείται σαν να ήταν μέρος του κύριου εγγράφου. Αυτό είναι πολύ χρήσιμο όταν θέλουμε να συμπεριλάβουμε τον ίδιο κώδικα σε πολλές σελίδες. Στη συγκεκριμένη σελίδα, με την `include()` ενσωματώνουμε το αρχείο `connectdb.php` που περιέχει τον κώδικα που χρειάζεται για την σύνδεση με τη βάση δεδομένων. Αυτή η διαδικασία της σύνδεσης με τη βάση δεδομένων, όπως είπαμε και προηγουμένως, απαιτείται σε κάθε σελίδα για να είναι εφικτή η ανταλλαγή πληροφοριών μεταξύ βάσης δεδομένων και συστήματος.

Το ερώτημα αποθηκεύεται στη μεταβλητή `$query` και στη συνέχεια περνάει με τη μορφή ορίσματος από τη συνάρτηση `mysql_query()`, η οποία εκτελεί το ερώτημα. Αν το ερώτημα εκτελεστεί σωστά επιστρέφει μια τιμή και αποθηκεύεται στη μεταβλητή `$result`. Από αυτήν τη μεταβλητή ελέγχουμε αν η διόρθωση εκτελέστηκε επιτυχώς, ή όχι.

Άλλη μία βασική λειτουργία της Διαχείρισης μαθημάτων είναι η δημιουργία ενός νέου μαθήματος. Η υλοποίησή της βασίζεται σε τρία πράγματα. Πρώτον, στη ύπαρξη μιας φόρμας με καλή διεπαφή έτσι ώστε ο χρήστης να μπορεί εύκολα να καταλάβει τι είδους στοιχεία απαιτεί η συμπλήρωση αυτής. Δεύτερον, στο διάβασμα των στοιχείων του μαθήματος που ο χρήστης έχει εισάγει στη φόρμα και τρίτον, ο έλεγχος αυτών των στοιχείων και η αποθήκευσή τους στη βάση δεδομένων.



Εικόνα 5: Οθόνη δημιουργίας νέου μαθήματος

Η δημιουργία της φόρμας υλοποιείται με αρκετά απλό κώδικα βασισμένο σε html. Το τμήμα του κώδικα όμως, που αξίζει να τονίσουμε είναι αυτό της λίστας (menu-list box), που υπάρχει για παράδειγμα στον 'Τύπο' του μαθήματος τον οποίο καλείται ο χρήστης να επιλέξει (βλέπε Εικόνα 5). Οι αλφαριθμητικές τιμές αυτής της λίστας εισάγονται δυναμικά από τον πίνακα less_type με τον ακόλουθο κώδικα php:

```
<select name="type" class="fortext" id="type">
  <?
  include('connectdb.php');
  $query="select * from less_type";
  $result=mysql_query($query);
  $row=mysql_num_rows($result);

  for ($i=0; $i<$row; $i++)
  {
    $pin=mysql_fetch_array($result);
    echo '<option value="'. $pin["typeid"].">'. $pin["typename"].'
  </option><br>';
  }
  ?>
</select>
```

Η αποθήκευση των δεδομένων που εισήγαγε ο χρήστης στη φόρμα καταλήγουν σε μια νέα εγγραφή του πίνακα lessons, αφού πρώτα περάσουν από τη συνάρτηση addslashes(), η οποία πάρνει μια συμβολοσειρά ως όρισμα και επιστέφει την μορφοποιημένη συμβολοσειρά. Η μορφοποίηση αυτή προκύπτει από την προσθήκη μιας ανάποδης καθέτου (\) μπροστά από κάθε ειδικό χαρακτήρα, για να μην μεταφράσει η βάση δεδομένων τους χαρακτήρες αυτούς ως χαρακτήρες ελέγχου.

Κάποια δεδομένα απ' αυτά περνάνε επίσης και από άλλες συναρτήσεις όπως οι strtolower(), ucwords() και ucfirst(), των οποίων οι λειτουργίες είναι η μετατροπή ενός αλφαριθμητικό σε κεφαλαίους χαρακτήρες, η μετατροπή του πρώτου γράμματος κάθε λέξης σε κεφαλαίο και η μετατροπή του πρώτου γράμματος του αλφαριθμητικού σε κεφαλαίο, αντίστοιχα. Π.χ. \$name = strtolower(\$name);

Το ερώτημα που εκτελεί τη αποθήκευση των στοιχείων στη βάση δεδομένων είναι το ακόλουθο:

```
INSERT INTO lessons (lessname, lesscode, lessdesc, lesstype, lesscategory, lesseidos, lesshours, lessfortos, lessdm) values (".$name.", ".$code.", ".$desc.", ".$type.", ".$category.", ".$eidos.", ".$hours.", ".$fortos.", ".$dm.")
```

5.2.5 Διαχείριση τμημάτων

Κατά την υλοποίηση της διαχείρισης των τμημάτων προέκυψε η ανάγκη για μία συνδυαστική έννοια μεταξύ εξαμήνου και μαθήματος. Έτσι δημιουργήσαμε την έννοια 'Τάξη' και παράλληλα τον ενδιάμεσο πίνακα 'class'. Με την επιλογή ενός εξαμήνου και ενός μαθήματος δημιουργείται μια 'τάξη' με ένα μοναδικό κωδικό (id). Πάνω σε αυτές τις 'τάξεις' που δημιουργούνται στηρίζεται η διαχείριση των τμημάτων.

Σε κάθε τάξη μπορούν να δημιουργηθούν όσα τμήματα θελήσει ο χρήστης, ενώ σε κάθε τμήμα αντιστοιχείται συγκεκριμένος αριθμός σπουδαστών. Επίσης, υπάρχει η δυνατότητα προβολής, διόρθωσης αλλά και διαγραφής οποιουδήποτε τμήματος.

Η υλοποίηση της δημιουργίας νέου τμήματος είναι βασισμένη στο πρότυπο υλοποίησης της δημιουργίας νέου μαθήματος που περιγράψαμε στην προηγούμενη ενότητα. Η μόνη διαφορά είναι στη διαδικασία αποθήκευσης. Όταν ο χρήστης πατάει 'Αποθήκευση', το πρόγραμμα ελέγχει πρώτα αν για το συγκεκριμένο εξάμηνο και μάθημα έχει δημιουργηθεί 'τάξη'. Αν έχει δημιουργηθεί, βρίσκει τον κωδικό (id) της 'τάξης' και έπειτα προχωράει στην αποθήκευση των στοιχείων της φόρμας στη βάση δεδομένων. Στην περίπτωση όμως που δεν δημιουργηθεί, αρχικά γίνεται μια εγγραφή στον πίνακα 'class', κρατάει σε μία μεταβλητή τον κωδικό (id) της νέας εγγραφής και στη συνέχεια, χρησιμοποιώντας αυτό το id προχωράει στη αποθήκευση των στοιχείων του νέου τμήματος.

Ο κώδικας που ελέγχει την ύπαρξη της 'τάξης' και δημιουργεί νέα αν δεν υπάρχει, φαίνεται στις παρακάτω γραμμές:

```
$query2 = "SELECT * from class where lessid='$less' and semid='$sem'";
$result2 = mysql_query($query2);
$rows = mysql_num_rows($result2);
if ($rows==0)
{
    $query3 = "INSERT INTO class (lessid, semid) values (".$less.", ".$sem.")";
    $result3 = mysql_query($query3);
    $classid = mysql_insert_id();
}
else if ($rows==1)
```



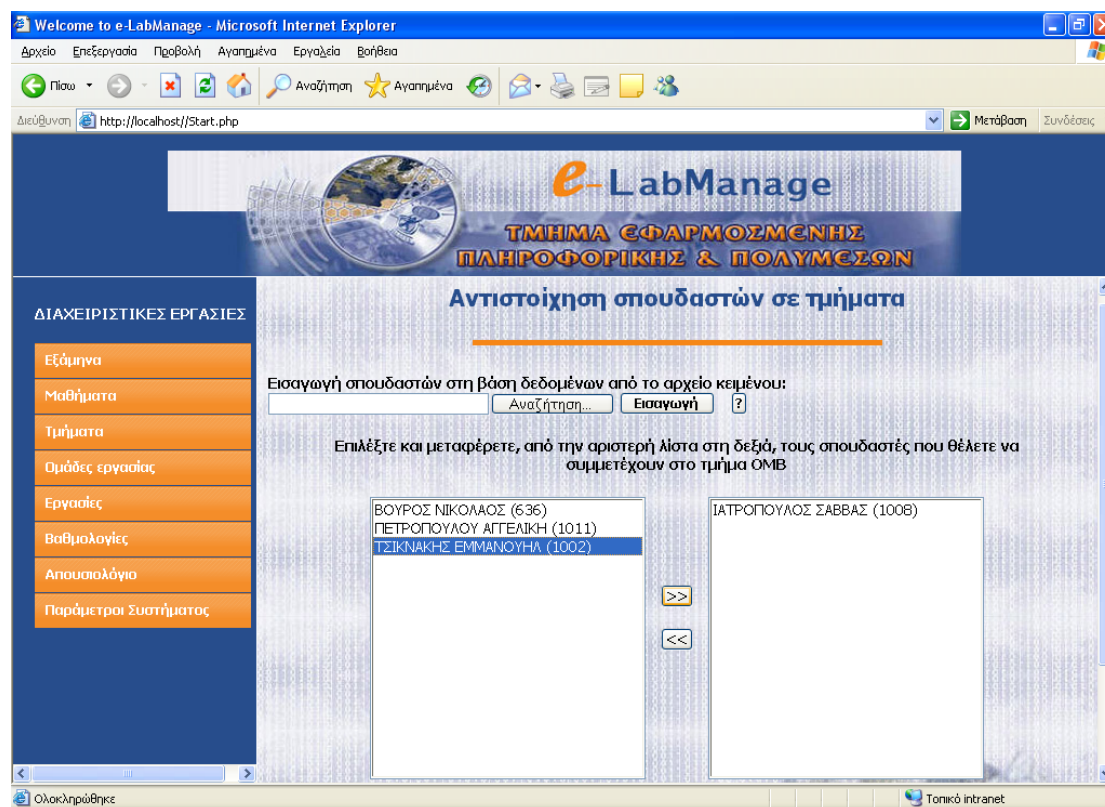
```

{
    $pin = mysql_fetch_array($result2);
    $classid = $pin["classid"];
}
else
{
    echo "<H4>ΠΡΟΣΟΧΗ: ΥΠΑΡΧΕΙ ΔΙΠΛΗ ΕΓΓΡΑΦΗ ΣΤΗ ΒΑΣΗ
ΔΕΔΟΜΕΝΩΝ!</H4><br>";
    exit;
}

```

Η συνάρτηση `mysql_insert_id()` χρησιμοποιείται για την ανάκτηση της τιμής του πρωτεύοντος κλειδιού (id) της τελευταίας εγγραφής που έχει εισαχθεί στην βάση δεδομένων από το συγκεκριμένο script.

Για την υλοποίηση της αντιστοίχισης των σπουδαστών σε κάθε τμήμα ξεχωριστά, χρησιμοποιήσαμε εκτός από php και κώδικα javascript για να κάνουμε τη διεπαφή αυτής της σελίδας ευκολότερη.



Εικόνα 6: Οθόνη αντιστοίχισης σπουδαστών σε τμήματα

Όπως βλέπεται στην παραπάνω εικόνα υπάρχουν δύο λίστες. Η πρώτη φορτώνει δυναμικά από τα βάση δεδομένων τους σπουδαστές που δεν έχουν καταχωρηθεί ακόμα σε τμήματα για το συγκεκριμένο εξάμηνο και μάθημα, ενώ η δεύτερη είναι αρχικά άδεια και περιμένει

από τον χρήστη να περάσει σε αυτήν, τους σπουδαστές που θέλει να καταχωρήσει στο συγκεκριμένο τμήμα, από την πρώτη λίστα. Όταν χρήστης διαλέξει τους σπουδαστές που θέλει για το τμήμα αυτό, προχωράει στην αποθήκευση. Η διαδικασία υλοποίησης της αποθήκευσης είναι ακριβώς η ίδια που περιγράψαμε στην ενότητα 5.2.4 *Διαχείριση Μαθημάτων*.

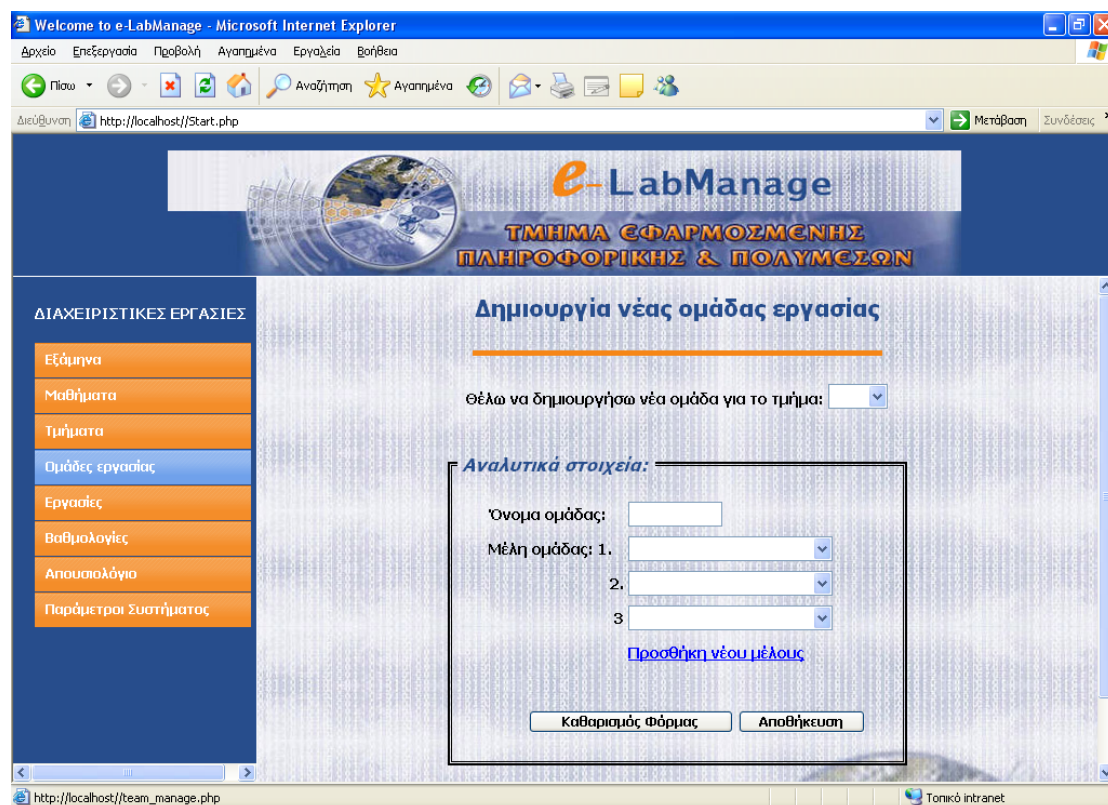
Μια από τις σημαντικότερες δυνατότητες που προσφέρει στο χρήστη η Διαχείριση Τμημάτων είναι αυτή της εισαγωγής νέων σπουδαστών στην βάση δεδομένων από ένα εξωτερικό αρχείο κειμένου (*.txt) που ο χρήστης επιλέγει και φυσικά έχει διαμορφώσει κατάλληλα. Για την αναλυτικότερη περιγραφή του αρχείου αυτού, παρακαλώ δείτε το [Παράρτημα Β](#).

Στον κώδικα που χρησιμοποιήσαμε για την μεταφορά αυτών των στοιχείων από το αρχείο κειμένου στη βάση δεδομένων, συμπεριλάβαμε κάποιες συναρτήσεις όπως, τη file(), τη count() και την explode(). Η συνάρτηση file() διαβάζει τα στοιχεία του αρχείου και τα επιστρέφει σε ένα πίνακα. Κάθε στοιχείο του πίνακα αυτού περιέχει μία γραμμή του αρχείου. Η συνάρτηση count() μετράει το πλήθος των στοιχείων μιας διάταξης. Τέλος, η explode() διαχωρίζει ένα αλφαριθμητικό σε μία διάταξη η οποία μπορεί στη συνέχεια να αποθηκευτεί, να ταξινομηθεί ή να εξεταστεί με οποιοδήποτε άλλο τρόπο.

5.2.6 Διαχείριση ομάδων εργασίας

Άλλο ένα τμήμα της Ηλεκτρονικής Διαχείρισης Εργαστηρίων είναι οι ομάδες εργασίας. Η διαχείριση ομάδων εργασίας περιλαμβάνει τις λειτουργίες προβολής, δημιουργία νέων και διαγραφής αυτών. Η υλοποίηση των παραπάνω λειτουργιών έχει ήδη περιγραφεί στις προηγούμενες ενότητες. Το μόνο σημείο που διαφέρει στην υλοποίηση είναι αυτό της δημιουργία νέας ομάδας.

Στην αρχική φόρμα που συμπληρώνει ο χρήστης για να δημιουργήσει μια νέα ομάδα εργασίας υπάρχουν θέσεις για την εισαγωγή δύο (2) μελών. Δύνεται όμως η επιπλέον επιλογή στο χρήστη να προσθέσει κι άλλα μέλη, πατώντας στη σύνδεση (link) «Προσθήκη νέου μέλους». Η σύνδεση αυτή παραπέμπει σε ένα script το οποίο προσθέτει ένα επιπλέον μενού για την επιλογή μέλους, κάθε φορά.



Εικόνα 7: Οθόνη δημιουργίας νέας ομάδας εργασίας

Με αυτήν την αλλαγή αυτή γίνεται αυτόματα και η αλλαγή του κώδικα αποθήκευσης των δεδομένων, όπως επίσης και ο αριθμός των στηλών του πίνακα 'team', όπου αποθηκεύονται οι νέες ομάδες. Το πιο σημαντικό κομμάτι κώδικα, που υλοποιούν τα παραπάνω είναι η ακόλουθη συνάρτηση, η οποία δημιουργεί τις επιπλέον στήλες του πίνακα:

```
<?php
function new_melos($w)
{
    $a=$w-2;
    $t=3;
    for ($y=0; $y<$a; $y++)
    {
        $query2="ALTER table team ADD teamstud$t bigint(20) unsigned";
        $result2=mysql_query($query2);
        ...
        ...
        $t=$t+1;
    }
?>
```

5.2.7 Διαχείριση εργασιών/project

Η επόμενη ενότητα της ιστοσελίδας αφορά τη διαχείριση εργασιών/project. Η κατασκευή της ενότητας αυτής ακολουθεί τα χνάρια των προηγούμενων ενότητων. Περιλαμβάνει εμφάνιση των εργασιών/project ανά μάθημα αλλά και ανά τμήμα, δημιουργία νέων καθώς και διαγραφή υπαρχόντων, όπως επίσης και δυνατότητα σύνδεσης εργασιών/ project με μαθήματα ή τμήματα.

The screenshot shows a web browser window displaying the 'e-LabManage' interface. The main content area is titled 'Εμφάνιση Εργασιών/ Project μαθήματος' (Display Projects/Project of course). Below the title, it specifies the course: 'Μάθημα: Διαχείριση Συστημάτων' (Course: System Management). A table lists the projects with columns for ID, Type, Description, Percentage, Type, and Status. Below the table, there is a section for displaying a list of projects for the semester 2000-2001E, with a dropdown menu for the department and a 'View' button.

Α/Α	ΤΥΠΟΣ	ΠΕΡΙΓΡΑΦΗ	ΠΟΣΟΣΤΟ	ΤΥΠΟΣ	ΔΙΑΓΡΑΦΗ
1	Project planning	Κατανόηση μοντέλων ανάπτυξης λογισμικού.	5%	Εβδομαδιαία εργασία	Ναι
2	Project Scheduling	Προσδιορισμός δραστηριοτήτων και χρόνου για καθένα απ' αυτές.	10%	Εβδομαδιαία εργασία	Ναι
3	Cost Estimation	Υπολογισμός κόστους υλοποίησης του έργου.	10%	Εβδομαδιαία εργασία	Ναι
4	Τελική Εξέταση	Ερωτήσεις επί' όλης της ύλης.	75%	Τελική εξέταση	Ναι

Εικόνα 8: Οθόνη εμφάνισης εργασιών/project για το μάθημα 'Διαχείριση συστημάτων'

Για την υλοποίηση της διαγραφής δεν έχει αναφερθεί καμιά λεπτομέρεια έως τώρα. Οι παρακάτω γραμμές κώδικας περιγράφουν την διαδικασία διαγραφής μιας εργασίας ή ενός project:

```
<?php
include('connectdb.php');

$delete_item = "delete from exercise where exeid = $_GET[id]";
$result=mysql_query($delete_item);
if ($result)
{
    echo '<center><h3>Η ΔΙΑΓΡΑΦΗ ΠΡΑΓΜΑΤΟΠΟΙΗΘΗΚΕ!</h3> <br><br>';
    ...
}
```

```

}
...
?>

```

Αν μπορούσαμε να δώσουμε μια «μετάφραση» στα ερωτήματα που χρησιμοποιούνται, το ερώτημα 'delete from exercise where exeid = \$_GET[id]' θα «μεταφραζόταν» ως εξής: *Διέγραψε από τον πίνακα exercise την γραμμή εκείνη όπου το exeid είναι ίσο με την τιμή της \$_GET[id]*.

5.2.8 Διαχείριση βαθμολογίας

Μία άλλη ενότητα λειτουργιών που προσφέρει η ιστοσελίδα, είναι αυτή της διαχείρισης των βαθμολογιών για τις εργασίες και τα projects. Μέσω αυτής της ενότητας λειτουργιών, δίνεται η δυνατότητα στο χρήστη να δει αναλυτικά τις βαθμολογίες των εργασιών/project που έχουν καταχωρηθεί για όλους του σπουδαστές του τμήματος. Παράλληλα, εμφανίζει για κάθε σπουδαστή τη συγκεντρωτική βαθμολογία των ήδη καταχωρημένων εργασιών του, καθώς επίσης και τον καθηγητή που έχει βάσει τις βαθμολογίες ανά εργασία.

Διαχείριση Βαθμολογίας

Εξάμηνο: 2000-2001E Μάθημα: Ψηφιακή Σχεδίαση Τμήμα: OMB
Εργασία: Υλοποίηση των πιλών NOT, AND

A.M.	ΕΠΩΝΥΜΟ	ΟΝΟΜΑ	ΟΜΑΔΑ	ΒΑΘΜΟΣ	Μ.Ο. ΕΡΓΑΣΙΩΝ	ΚΑΘΗΓΗΤΗΣ
872	ΔΟΞΑΣΤΑΚΗ	ΜΑΡΙΑ	A	9	1.27	ΣΤΡΑΤΑΚΗΣ
661	ΚΟΡΝΕΛΑΚΗΣ	ΜΙΧΑΗΛ	A	9	1.27	ΣΤΡΑΤΑΚΗΣ
840	ΠΑΠΑΔΑΚΗ	ΕΛΕΝΗ	B	7	1.30	ΣΤΡΑΤΑΚΗΣ
604	ΚΟΥΤΑΝΤΟΥ	ΕΛΕΥΘΕΡΙΑ	B	7	1.30	ΣΤΡΑΤΑΚΗΣ
1010	ΚΑΛΑΘΑΚΗΣ	ΙΩΑΝΝΗΣ	Γ	6	1.33	ΣΤΡΑΤΑΚΗΣ
1268	ΕΛΕΥΘΕΡΑΚΗ	ΠΙΝΕΛΟΠΗ	Γ	6	1.33	ΣΤΡΑΤΑΚΗΣ
990	ΑΛΕΞΑΚΗ	ΖΩΗ	Δ	8.5	1.53	ΣΤΡΑΤΑΚΗΣ
1177	ΒΑΡΔΗΣ	ΣΤΕΦΑΝΟΣ	Δ	8.5	1.53	ΣΤΡΑΤΑΚΗΣ

Εκτύπωση

Εικόνα 8: Οθόνη εμφάνισης βαθμολογιών

Για την υλοποίηση του υπολογισμού της συγκεντρωτικής βαθμολογίας κάθε σπουδαστή, χρειάστηκαν πολύ απλά μαθηματικά και φυσικά προγραμματιστικές γνώσεις. Οι παρακάτω γραμμές κώδικα είναι αυτές που βγάζουν το επιθυμητό αποτέλεσμα:

```
$q_mo = "select * from grades where studam = '$stud_am' and classid =
_SESSION[gr_classid] and groupid = $_GET[group] ";
$r_mo = mysql_query($q_mo);
$x = mysql_num_rows($r_mo);
$sum=0;
for ($y=0; $y<$x; $y++)
{
    $grad=mysql_fetch_array($r_mo);
    $exer=$grad["exeid"];
    $q3="select exepercent from exercise where exeid='$exer'";
    $r3=mysql_query($q3);
    $pososto=mysql_fetch_array($r3);
    $bathmos=($grad["grade"]*($pososto["exepercent"]/100));
    $sum=$sum+$bathmos;
}
printf("%.2f", $sum);
```

Αρχικά εκτελείται ένα ερώτημα για να βρεθούν πόσες βαθμολογίες έχουν καταχωρηθεί για συγκεκριμένο σπουδαστή του τμήματος. Στη συνέχεια, με την συνάρτηση `mysql_num_rows()` μετρούνται οι γραμμές του αποτελέσματος του ερωτήματος, έτσι ώστε η τιμή αυτή να χρησιμοποιηθεί σαν μέτρο για τις επαναλήψεις που βρόγχου που ακολουθεί. Μέσα στον επαναληπτικό βρόγχο εκτελείται άλλο ένα ερώτημα, το οποίο βρίσκει το ποσοστό βαρύτητας που έχει κάθε εργασία ή project για το γενικό μέσο όρο των εργασιών. Τέλος, γίνεται ο πολλαπλασιασμός του βαθμού επί του ποσοστού κάθε εργασίας και προστίθεται στη μεταβλητή `$sum`, της οποίας η τιμή είναι αυτή που εμφανίζεται στην οθόνη ως συγκεντρωτική βαθμολογία του σπουδαστή.

Η οθόνη προβολής των βαθμολογιών υποστηρίζει επίσης τη λειτουργία της άμεσης εκτύπωσης αυτής, με το απλό πάτημα ενός κουμπιού.

Μία λειτουργία που απαραίτητως συμπεριλαμβάνεται στην διαχείριση των βαθμολογιών είναι αυτή της καταχώρησης νέων βαθμολογιών. Η υλοποίηση αυτής είναι βασισμένη στην εκτέλεση ενός ερωτήματος που εισάγει στοιχεία στη βάση δεδομένων, όπως φυσικά όλων των ειδών οι καταχωρήσεις. Το πρόγραμμα προσφέρει δύο επιλογές καταχώρησης νέων βαθμολογιών. Την επιλογή της ατομικής και

της ομαδικής καταχώρησης. Στην πρώτη περίπτωση ο χρήστης εισάγει βαθμολογίες ανά σπουδαστή, ενώ στη δεύτερη ανά ομάδες εργασίας. Λόγω αυτών, προκύπτει μια ιδιαιτερότητα στον κώδικα της αποθήκευσης της ομαδικής βαθμολογίας. Υπάρχουν δύο επαναληπτικοί βρόγχοι, ο πρώτος αναζητά τα Α.Μ. των σπουδαστών που αποτελούν την ομάδα εργασίας, ενώ ο δεύτερος αποθηκεύει την ίδια βαθμολογία σε όλα τα μέλη της ομάδας, κάνοντας διαφορετική εγγραφή για κάθε μέλος στον πίνακα 'grades'. Ο κώδικας αυτών των βρόγχων είναι ο ακόλουθος:

```
for ($i=0; $i<$rows; $i++)
{

    $gr2 = addslashes($gr[$i]);
    $team = addslashes($teams[$i]);

    $query_am="select studam from students where studteam='$team'";
    $result_am = mysql_query($query_am);
    $num=mysql_num_rows($result_am);

    for ($j=0; $j<$num; $j++)
    {
        $student=mysql_fetch_array($result_am);
        $am=$student["studam"];

        $query = "INSERT INTO grades values (". $am. ", ". $idclass. ",
        ". $idgroup. ", ". $idexe. ", ". $user. ", ". $gr2. ")";
        $result = mysql_query($query);
    }
}
```

5.2.9 Διαχείριση απουσιολογίου

Η προ τελευταία διαχειριστική ενότητα της Ηλεκτρονικής Διαχείρισης Εργαστηρίων είναι η διαχείριση του απουσιολογίου. Το εύρος της υλοποίησης της περιορίζεται στην προβολή των παρουσιών ανά τμήμα και ημερομηνία, καθώς επίσης και στην καταχώρηση νέων παρουσιών. Στις παραπάνω ενότητες έχει περιγραφεί η διαδικασία υλοποίησης προβολής στοιχείων από τη βάση δεδομένων, όπως επίσης και η διαδικασία υλοποίησης νέων καταχωρήσεων σε αυτήν.

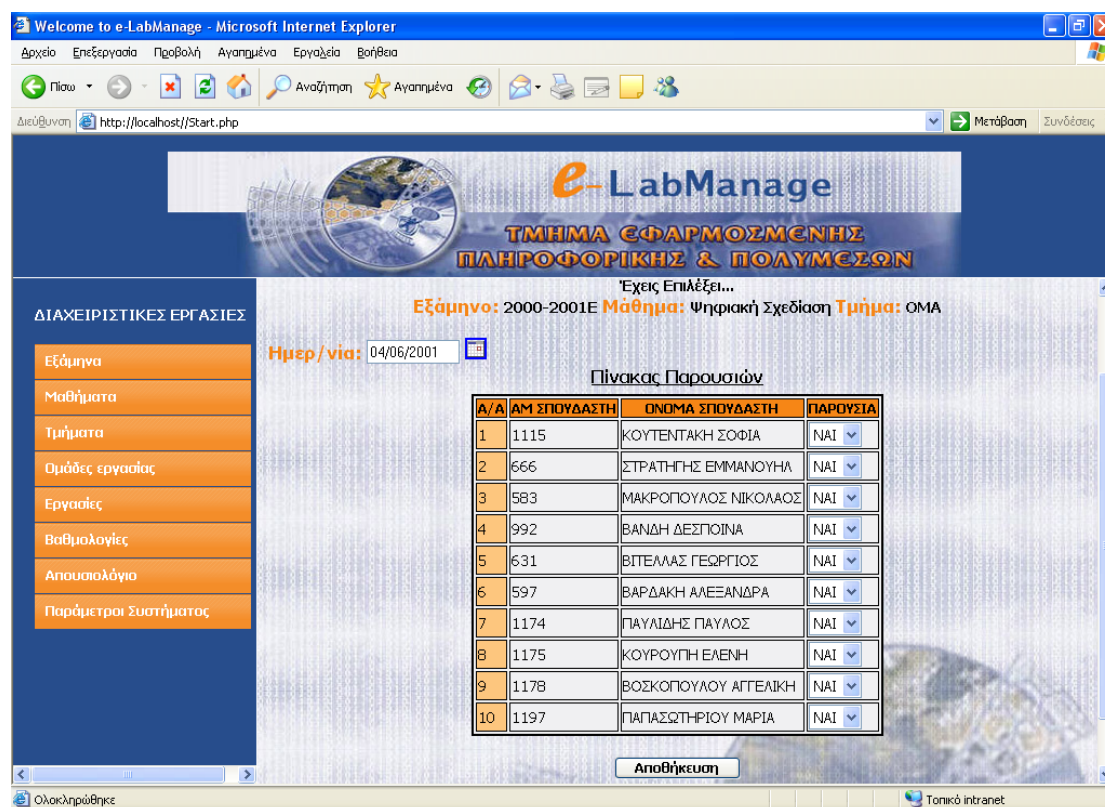
Μια μικρή λεπτομέρεια που αξίζει να σημειωθεί είναι ο τρόπος αποθήκευσης της παρουσίας κάθε σπουδαστή, σε έναν κοινό πίνακα.

Όπως βλέπετε στην Εικόνα 9, η παρουσία επιλέγεται μέσω μιας λίστας με δύο επιλογές, ΝΑΙ ή ΟΧΙ. Ο κώδικας που της λίστας είναι:

```
<select name="abs[]" size="1" class="fortext">
  <option value="ΠΑΡΟΥΣΙΑ">ΝΑΙ</option>
  <option value="ΑΠΟΥΣΙΑ">ΟΧΙ</option>
</select>
```

Όπου, abs[] είναι ο πίνακας που αποθηκεύονται οι επιλογές του χρήστη. Κατά την αποθήκευση των παρουσιών ο κώδικας που διαβάζει κάθε στοιχείο αυτού του πίνακα είναι ο ακόλουθος:

```
for ($i=0; $i<$rows; $i++)
{
    $abs1 = addslashes($abs[$i]);
    ...
    ...
}
```



The screenshot shows the 'e-LabManage' web application interface. The main content area displays the 'Πίνακας Παρουσιών' (Attendance Table) for the semester 'Εξάμηνο: 2000-2001Ε' and subject 'Μάθημα: Ψηφιακή Σχεδίαση Τμήμα: ΟΜΑ'. The date is set to '04/06/2001'. The table lists 10 students with their IDs, AM numbers, names, and attendance status (all 'ΝΑΙ').

A/A	ΑΜ ΣΠΟΥΔΑΣΤΗ	ΟΝΟΜΑ ΣΠΟΥΔΑΣΤΗ	ΠΑΡΟΥΣΙΑ
1	1115	ΚΟΥΤΕΝΤΑΚΗ ΣΟΦΙΑ	ΝΑΙ
2	666	ΣΤΡΑΤΗΓΗΣ ΕΜΜΑΝΟΥΗΛ	ΝΑΙ
3	583	ΜΑΚΡΟΠΟΥΛΟΣ ΝΙΚΟΛΑΟΣ	ΝΑΙ
4	992	ΒΑΝΔΗ ΔΕΣΠΟΙΝΑ	ΝΑΙ
5	631	ΒΙΤΕΛΛΑΣ ΓΕΩΡΓΙΟΣ	ΝΑΙ
6	597	ΒΑΡΔΑΚΗ ΑΛΕΞΑΝΔΡΑ	ΝΑΙ
7	1174	ΠΑΥΛΙΔΗΣ ΠΑΥΛΟΣ	ΝΑΙ
8	1175	ΚΟΥΡΟΥΠΗ ΕΛΕΝΗ	ΝΑΙ
9	1178	ΒΟΣΚΟΠΟΥΛΟΥ ΑΓΓΕΛΙΚΗ	ΝΑΙ
10	1197	ΠΑΠΑΣΩΤΗΡΙΟΥ ΜΑΡΙΑ	ΝΑΙ

Εικόνα 9: Οθόνη καταχώρησης νέων παρουσιών

Ένα άλλο πρωτοεμφανιζόμενο στοιχείο στην ιστοσελίδα είναι το ημερολόγιο. Αυτό υπάρχει σε όλα τα σημεία που ζητείται από τον χρήστη να επιλέξει ημερομηνία. Το ημερολόγιο που εμφανίζεται εφόσον

ο χρήστης το θελήσει (πατώντας το σχετικά κουμπί), φαίνεται στην ακόλουθη εικόνα:



Εικόνα 10: Ημερολόγιο

Το ημερολόγιο αυτό υλοποιήθηκε με κώδικα javascript, ο οποίος περιλαμβάνεται στον αρχείο calendar.js, με σκοπό τη διευκόλυνση του χρήστη.

5.2.10 Διαχείριση παραμέτρων συστήματος

Η τελευταία διαχειριστική ενότητα της ιστοσελίδας είναι η διαχείριση των παραμέτρων του συστήματος. Αυτή ασχολείται με τους τύπους, τις κατηγορίες και τα είδη των μαθημάτων, όπως επίσης και για τις κατηγορίες των εργασιών ή projects. Η λειτουργίες των παραπάνω περιλαμβάνουν την προβολή των στοιχείων και την προσθήκη νέων. Η υλοποίηση αυτών, είναι η ίδια με αυτήν των λειτουργιών των προαναφερθέντων διαχειριστικών ενοτήτων.

Στην Εικόνα 11 που ακολουθεί φαίνεται η διεπαφή της σελίδας για τις κατηγορίες μαθημάτων. Η διεπαφή των άλλων παραμέτρων του συστήματος είναι σχεδόν η ίδια με την υποφαινόμενη, με τη μόνη διαφορά τους τίτλους και τα περιεχόμενα των πινάκων.

Welcome to e-LabManage - Microsoft Internet Explorer

Διεύθυνση http://localhost/Start.php

e-LabManage

ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΠΟΛΥΜΕΣΩΝ

Παράμετροι Συστήματος

Οι Κατηγορίες Μαθημάτων που υπάρχουν είναι οι παρακάτω:

A/A	ΚΑΤΗΓΟΡΙΑ ΜΑΘΗΜΑΤΟΣ
1	Μάθημα Γενικής Υποδομής
2	Μάθημα Ειδικής Υποδομής
3	Μάθημα Ειδικότητας
4	Διοίκησης Οικονομίας & Ανθρωπιστικών Σπουδών

Προσθήκη νέας κατηγορίας :

Όνομα κατηγορίας:

Αποθήκευση

Ολοκληρώθηκε

Τοπικό Intranet

Εικόνα 11: Οθόνη διαχείρισης παραμέτρων συστήματος

6. Συμπεράσματα

Η ανάθεση αυτής της πτυχιακής εργασίας έγινε με σκοπό την κατασκευή ενός μέσου διαχείρισης των εργαστηριακών μαθημάτων. Το μέσο που επιλέγαμε να φτιάξουμε είναι μια ιστοσελίδα, την οποία ονομάσαμε “e Lab Manage”. Η ιστοσελίδα αυτή διευκολύνει τους χρήστες, που στην προκειμένη περίπτωση είναι μόνο καθηγητές, να κάνουν όλες τις απαραίτητες ενέργειες που απαιτούνται για την διεκπεραίωση των εργαστηριακών μαθημάτων.

Εφόσον αναλύθηκαν οι τεχνολογίες και τα προγράμματα που χρησιμοποιήθηκαν για την υλοποίηση της ιστοσελίδας ηλεκτρονικής διαχείρισης εργαστηρίων (“e Lab Manage”), στη συνέχεια έγινε μια «ξενάγηση» στο αποτέλεσμα αυτής της δουλειάς. Οι δυνατότητες που προσφέρει το “e Lab Manage” σε συνδυασμό με τη δυναμική βάση δεδομένων που βρίσκεται πίσω από αυτό, καλύπτουν πλήρως τις απαιτήσεις των καθηγητών σε όλα τα εργαστήρια.

Για οποιοδήποτε πρόγραμμα που κατασκευάζεται είτε μέσω μιας πτυχιακή εργασία, όπως αυτή, είτε μέσω μιας επαγγελματικής δουλειάς, υπάρχουν προοπτικές βελτίωσης και αναβάθμισης. Μέσα στα πλαίσια της βελτίωσης ενός προγράμματος, μπορούν να προκύψουν διάφορων ειδών απαιτήσεις που θα πρέπει να ικανοποιήσει το σύστημα. Όσον αφορά τη συγκεκριμένη πτυχιακή, πιθανές βελτιώσεις θα ήταν οι ακόλουθες: Πρώτον, στη διαχείριση τμημάτων η δυνατότητα ανάκτησης και αποθήκευσης νέων στοιχείων στη βάση δεδομένων μέσω, εκτός από αρχεία κειμένου (*.txt), και άλλου τύπου αρχεία όπως excel, acrobat reader, κ.ά. Δεύτερον, στη διαχείριση των τμημάτων στην ενότητα αντιστοίχισης σπουδαστών σε τμήματα, όπου υπάρχουν δύο λίστες που δίνουν τη δυνατότητα μεταφοράς στοιχείων από την μία στη άλλη. Όταν σε αυτό το σημείο του προγράμματος ο χρήστης προβεί σε αποθήκευση, τότε αποθηκεύεται μόνο μία γραμμή από τη λίστα κάθε φορά. Αυτό σίγουρα είναι ένα αρκετά σημαντικό πρόβλημα του “e Lab Manage” το οποίο όμως στα πλαίσια αυτής της πτυχιακής δεν μπορέσαμε να αντιμετωπίσουμε. Το ίδιο ακριβώς πρόβλημα εμφανίζεται και στη διαχείριση των εργασιών και συγκεκριμένα στην αντιστοίχιση των εργασιών σε τμήματα.

Από τα παραπάνω συμπεραίνουμε ότι το αποτέλεσμα αυτής της πτυχιακής εργασίας, παρ’ όλα τα μειονεκτήματα που αναφέραμε, καλύπτει κατά το μέγιστο τους αρχικούς της στόχους.

7. Παράρτημα Α

Sql κώδικας για τη δημιουργία των πινάκων:

```
CREATE DATABASE /*!32312 IF NOT EXISTS*/ elabmanage;
USE elabmanage;
```

```
CREATE TABLE `absences` (
  `absdate` varchar(10) NOT NULL default "",
  `groupid` bigint(20) unsigned NOT NULL default '0',
  `studam` bigint(20) unsigned NOT NULL default '0',
  `abscheck` varchar(8) NOT NULL default "" COMMENT 'ΠΑΡΟΥΣΙΑ Η ΑΠΟΥΣΙΑ',
  PRIMARY KEY (`absdate`, `studam`)
) ENGINE=MyISAM DEFAULT CHARSET=greek;
```

```
CREATE TABLE `class` (
  `classid` bigint(20) unsigned NOT NULL auto_increment,
  `semid` bigint(20) unsigned NOT NULL default '0',
  `lessid` bigint(20) unsigned NOT NULL default '0',
  PRIMARY KEY (`classid`)
) ENGINE=MyISAM DEFAULT CHARSET=greek ROW_FORMAT=DYNAMIC;
```

```
CREATE TABLE `days` (
  `dayid` tinyint(3) unsigned NOT NULL auto_increment,
  `dayname` varchar(10) NOT NULL default "",
  PRIMARY KEY (`dayid`)
) ENGINE=MyISAM DEFAULT CHARSET=greek;
```

```
CREATE TABLE `exe_class` (
  `classid` bigint(20) unsigned NOT NULL default '0',
  `groupid` bigint(20) unsigned NOT NULL default '0',
  `exeid` bigint(20) unsigned NOT NULL default '0',
  PRIMARY KEY (`groupid`, `exeid`)
) ENGINE=MyISAM DEFAULT CHARSET=greek;
```

```
CREATE TABLE `exercise` (
  `exeid` bigint(20) unsigned NOT NULL auto_increment,
  `exetitle` varchar(45) NOT NULL default "",
  `exedesc` text NOT NULL,
  `exepercent` smallint(5) unsigned NOT NULL default '0',
  `exetype` varchar(45) NOT NULL default 'Εργασία Εβδομάδας' COMMENT 'Εργασία
Εβδομάδας ή Εργασία μήνα ή Project εξαμήνου ή Project κατοχύρωσης',
```

```
`lessid` bigint(20) unsigned NOT NULL default '0',  
PRIMARY KEY (`exeid`)  
) ENGINE=MyISAM DEFAULT CHARSET=greek;
```

```
CREATE TABLE `exercise_type` (  
  `exetypeid` smallint(5) unsigned NOT NULL auto_increment,  
  `exetypename` varchar(45) NOT NULL default "",  
  PRIMARY KEY (`exetypeid`)  
) ENGINE=MyISAM DEFAULT CHARSET=greek;
```

```
CREATE TABLE `grades` (  
  `studam` bigint(20) unsigned NOT NULL default '0',  
  `classid` bigint(20) unsigned NOT NULL default '0',  
  `groupid` bigint(20) unsigned NOT NULL default '0',  
  `exeid` bigint(20) unsigned NOT NULL default '0',  
  `proid` bigint(20) unsigned NOT NULL default '0',  
  `grade` double NOT NULL default '0',  
  PRIMARY KEY (`studam`, `groupid`, `classid`, `exeid`)  
) ENGINE=MyISAM DEFAULT CHARSET=greek;
```

```
CREATE TABLE `groups` (  
  `groupid` bigint(20) unsigned NOT NULL auto_increment,  
  `groupname` varchar(5) NOT NULL default "",  
  `groupday` varchar(10) NOT NULL default "",  
  `grouptime` varchar(11) NOT NULL default "",  
  `groupcap` tinyint(3) unsigned NOT NULL default '0',  
  `grouproom` varchar(6) NOT NULL default "",  
  `groupteam` varchar(3) NOT NULL default " COMMENT 'TO TMHMA ΘΑ ΧΩΡΙΖΕΤΑΙ  
ΣΕ ΟΜΑΔΕΣ; ΝΑΙ 'Η ΟΧΙ',  
  `groupteachera` varchar(45) NOT NULL default "",  
  `groupteacherb` varchar(45) default "",  
  `classid` bigint(20) unsigned NOT NULL default '0',  
  PRIMARY KEY (`groupid`)  
) ENGINE=MyISAM DEFAULT CHARSET=greek;
```

```
CREATE TABLE `holidays` (  
  `holidayid` bigint(20) unsigned NOT NULL auto_increment,  
  `holidaydate` varchar(10) NOT NULL default "",  
  `holidaydesc` varchar(45) default "",  
  `semid` bigint(20) unsigned NOT NULL default '0',  
  PRIMARY KEY (`holidayid`, `semid`)  
) ENGINE=MyISAM DEFAULT CHARSET=greek;
```

```
CREATE TABLE `less_category` (  
  `categid` int(10) unsigned NOT NULL auto_increment,
```

```
`catename` varchar(45) NOT NULL default "",
PRIMARY KEY (`catid`)
) ENGINE=MyISAM DEFAULT CHARSET=greek;

CREATE TABLE `less_eidos` (
  `eidosid` int(10) unsigned NOT NULL auto_increment,
  `eidosname` varchar(45) NOT NULL default "",
  PRIMARY KEY (`eidosid`)
) ENGINE=MyISAM DEFAULT CHARSET=greek;

CREATE TABLE `less_type` (
  `typeid` int(10) unsigned NOT NULL auto_increment,
  `typename` varchar(45) NOT NULL default "",
  PRIMARY KEY (`typeid`)
) ENGINE=MyISAM DEFAULT CHARSET=greek;

CREATE TABLE `lessons` (
  `lessid` bigint(20) unsigned NOT NULL auto_increment,
  `lessname` varchar(45) NOT NULL default "",
  `lesscode` varchar(10) NOT NULL default "",
  `lessdesc` text,
  `lessstype` smallint(5) unsigned NOT NULL default '0',
  `lesscategory` smallint(5) unsigned NOT NULL default '0',
  `lesseidos` smallint(5) unsigned NOT NULL default '0',
  `lesshours` smallint(5) unsigned NOT NULL default '0',
  `lessfortos` smallint(5) unsigned NOT NULL default '0',
  `lessdm` smallint(5) unsigned NOT NULL default '0',
  PRIMARY KEY (`lessid`)
) ENGINE=MyISAM DEFAULT CHARSET=greek;

CREATE TABLE `privileges` (
  `priv_id` int(10) unsigned NOT NULL auto_increment,
  `priv_name` varchar(20) NOT NULL default "",
  `priv_password` varchar(20) NOT NULL default "",
  PRIMARY KEY (`priv_id`)
) ENGINE=MyISAM DEFAULT CHARSET=greek;

CREATE TABLE `professors` (
  `proid` bigint(20) unsigned NOT NULL auto_increment,
  `profirstname` varchar(25) NOT NULL default "",
  `prolastname` varchar(35) NOT NULL default "",
  `profname` varchar(25) NOT NULL default "",
  `probirthdate` varchar(10) NOT NULL default "",
  `proemail` varchar(45) NOT NULL default "",
  `protelephone` varchar(10) NOT NULL default "",
```

```

`probathmida` varchar(45) NOT NULL default "",
`prosxoli` varchar(10) NOT NULL default "",
`protmima` varchar(30) NOT NULL default "",
`prouername` varchar(20) NOT NULL default "",
`propasswrđ` varchar(20) NOT NULL default "",
`priv_id` int(10) unsigned default '1',
PRIMARY KEY (`proid`)
) ENGINE=MyISAM DEFAULT CHARSET=greek;

CREATE TABLE `semester` (
  `semid` bigint(20) unsigned NOT NULL auto_increment,
  `semname` varchar(10) NOT NULL default "",
  `semdesc` tinyint(3) unsigned default '0' COMMENT 'χειμερινό ή εαρινό',
  `semstrdate` varchar(10) NOT NULL default '',
  `semenddate` varchar(10) NOT NULL default '',
  `semstrexama` varchar(10) NOT NULL default '',
  `semendexama` varchar(10) NOT NULL default '',
  `semstrexamb` varchar(10) NOT NULL default "",
  `semendexamb` varchar(10) NOT NULL default "",
  PRIMARY KEY (`semid`)
) ENGINE=MyISAM DEFAULT CHARSET=greek COMMENT='InnoDB free: 4096 kB';

CREATE TABLE `semester_desc` (
  `descid` tinyint(3) unsigned NOT NULL auto_increment,
  `descname` varchar(45) NOT NULL default "",
  PRIMARY KEY (`descid`)
) ENGINE=MyISAM DEFAULT CHARSET=greek;

CREATE TABLE `students` (
  `studam` bigint(20) unsigned NOT NULL default '0',
  `studlastname` varchar(30) NOT NULL default "",
  `studfirstname` varchar(15) NOT NULL default "",
  `studsemin` varchar(10) NOT NULL default "",
  `stud_status` tinyint(3) unsigned default '0' COMMENT '0=μη
επιλεγμένος\r\n1=επιλεγμενος',
  `studgroup` bigint(20) unsigned default '0',
  `studteam` bigint(20) unsigned default '0',
  `classid` bigint(20) unsigned NOT NULL default '0',
  `stud_gstatus` tinyint(3) unsigned default '0',
  PRIMARY KEY (`studam`, `classid`)
) ENGINE=MyISAM DEFAULT CHARSET=greek;

CREATE TABLE `team` (
  `teamid` bigint(20) unsigned NOT NULL auto_increment,
  `teamname` varchar(10) NOT NULL default "",

```

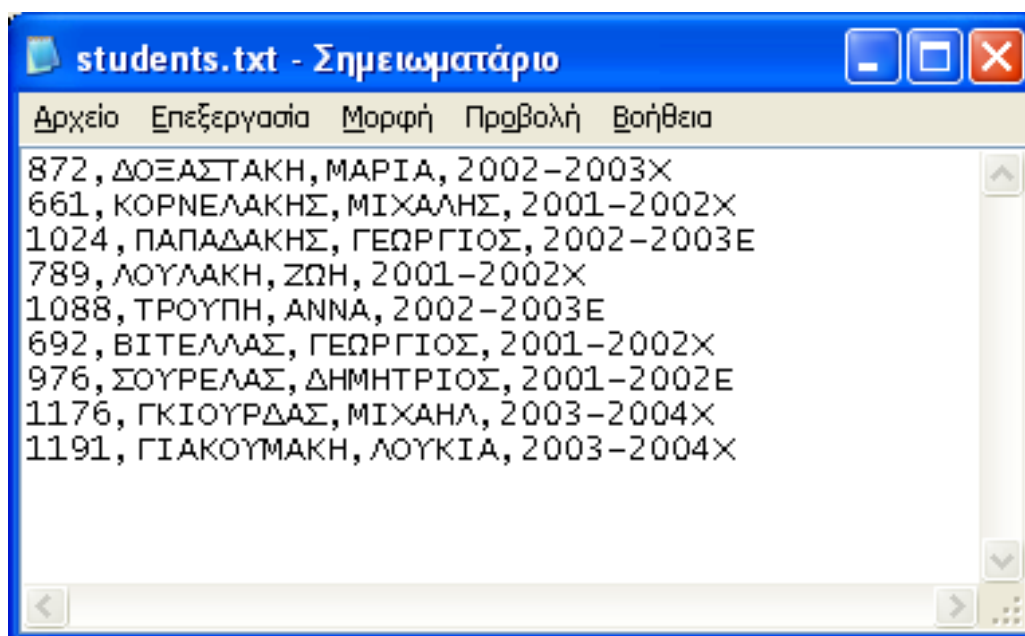
```
`teamstud1` bigint(20) unsigned NOT NULL default '0',  
`teamstud2` bigint(20) unsigned NOT NULL default '0',  
`teamstud3` bigint(20) unsigned default NULL,  
`teamstud4` bigint(20) unsigned default NULL,  
`groupid` bigint(20) unsigned default '0',  
`teamstud5` bigint(20) unsigned default NULL,  
`teamstud6` bigint(20) unsigned default NULL,  
PRIMARY KEY (`teamid`)  
) ENGINE=MyISAM DEFAULT CHARSET=greek;
```


8. Παράρτημα Β

Το αρχείο κειμένου (*.txt) που ο χρήστης μπορεί να επιλέξει, στο μέρος των Τμημάτων της Ηλεκτρονικής Διαχείρισης Εργαστηρίων, θα πρέπει να τηρεί συγκεκριμένες προϋποθέσεις.

Θα πρέπει να περιέχει σε κάθε γραμμή μία μόνο εγγραφή της μορφής: ΑΜ,ΕΠΩΝΥΜΟ,ΟΝΟΜΑ,ΕΞΑΜΗΝΟ όπου ΑΜ = ο αριθμός μητρώου του σπουδαστή, ΕΠΩΝΥΜΟ = το επώνυμο του σπουδαστή, ΟΝΟΜΑ = το όνομα του σπουδαστή, ΕΞΑΜΗΝΟ = εξάμηνο εισαγωγής του σπουδαστή στη σχολή με μορφή π.χ. 2002-2003Χ.

Υπόδειγμα αρχείου κειμένου:



9. Γλωσσάριο

Apache: Ένας από τους πιο διαδεδομένους διακομιστές περιεχομένου για το World Wide Web.

Client-Side Scripting: Μια μέθοδος προγραμματισμού για το Web όπου οι εντολές που δίνουμε στέλνονται και μεταφράζονται από το πρόγραμμα - φυλλομετρητή του πελάτη (client).

CSS (Cascading Style Sheets): Μια μέθοδος με την οποία δημιουργούμε ενιαία στυλ όπως χρώματα, γραμματοσειρές, borders κ.λπ για ιστοσελίδες.

HTML (Hypertext Markup Language): Η γλώσσα προγραμματισμού για την εμφάνιση περιεχομένου στο World Wide Web.

HTTP (Hypertext Transfer Protocol): Μια υπηρεσία του Internet (ένα πρωτόκολλο επικοινωνιών) που βρίσκεται στο επίπεδο εφαρμογής και χρησιμοποιείται για την αποστολή υπερκειμένων μέσα από το δίκτυο.

JavaScript: Μια client-side scripting γλώσσα που αναπτύχθηκε από τη Netscape και που μπορεί να ενσωματωθεί σε μια σελίδα HTML για να προσθέσει λειτουργικότητα σε αυτήν. Δεν πρέπει να συγχέεται με τη Java.

MySQL: Μια ελεύθερη ανοικτού κώδικα σχεσιακή βάση δεδομένων.

PHP (PHP Hypertext Preprocessor): Μια γλώσσα προγραμματισμού ευρέως διαδεδομένη στον τομέα της ανάπτυξης εφαρμογών για το World Wide Web.

Server-Side Scripting: Μια μέθοδος προγραμματισμού για το Web όπου οι εντολές που δίνουμε μεταφράζονται από τον διακομιστή (server) και το αποτέλεσμα τους (outrput) στέλνεται κατευθείαν στο πρόγραμμα - φυλλομετρητή του πελάτη.

Server (Διακομιστής): Ένας υπολογιστής που τρέχει το κατάλληλο λογισμικό έτσι ώστε να μπορεί να προσφέρει υπηρεσίες (services) σε υπολογιστές-πελάτες (clients) όπως περιεχόμενο WWW, υπηρεσίες νοητού τερματικού (telnet) κ.α.

Φυλλομετρητής (Browser): Ένα πρόγραμμα ερμηνεύσης κώδικα HTML, CSS και Client-Side script.

10. Βιβλιογραφία

Εμμ. Α. Γιακουμάκης (1994), «Τεχνολογία Λογισμικού» τόμος Α & Β, Εκδόσεις Α. Σταμούλης

Julie C. Meloni (2004), “Μάθετε PHP, MySQL και Apache - Όλα σε ένα”, Εκδόσεις Μ. Γκιούρδας.

Luke Welling and Laura Thomson (2001), “PHP and MySQL Web Development”, SAMS Publishing.

Jay Greenspan and Brad Bulger (2001), “MySQL/PHP Database Applications”, MRT Books.

Jesus Castagnetto, Harish Rawat, Sascha Schumann, Chris Scollo and Deepak Veliath (1999), “Professional PHP Programming”, Wrox Press Ltd.

Janet Valade (2004), “PHP & MySQL For Dummies” 2nd Edition, Wiley Publishing, Inc.

Mat Zandstra (2000), “Teach Yourself PHP4 in 24Hours”, SAMS Publishing.

Mohammed J. Kabir (2003), “Secure PHP Development: Building 50 Practical Applications”, Wiley Publishing, Inc.