



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή εργασία

ΣΥΣΤΗΜΑ ΟΡΓΑΝΩΣΗΣ ΚΑΙ ΕΠΙΛΟΓΗΣ ΣΥΓΓΡΑΜΜΑΤΩΝ



Του φοιτητή
Λάμπρου Αριστείδη
Αρ.Μητρώου 04/2487

Επιβλέπων καθηγητής
Βαφειάδης Αντώνης

Θεσσαλονίκη 2010

Πρόλογος

Η εφαρμογή αυτή δημιουργήθηκε το ακαδημαϊκό έτος 2009-2010 από τον Λάμπρου Αριστείδη, φοιτητή του τμήματος Πληροφορικής του ΑΤΕΙ, υπό την εποπτεία του καθηγητή της σχολής κ. Βαφειάδη Αντώνη, στα πλαίσια εκπόνησης της πτυχιακής εργασίας με τίτλο «Σύστημα οργάνωσης και επιλογής συγγραμμάτων» με χρήση PHP και MYSQL. Η επιλογή του θέματος αυτού αποφασίστηκε τον Οκτώβριο του 2009 και έγινε με γνώμονα την απόκτηση γνώσης που μπορεί να χρησιμοποιηθεί στο άμεσο μέλλον στο εργασιακό περιβάλλον. Η ραγδαία αύξηση των φοιτητών του τμήματος, καθώς περνούσαν τα χρόνια, σε συνδυασμό με την ανάγκη βελτίωσης της διαδικασίας διεξαγωγής των δηλώσεων των συγγραμμάτων για την εξυπηρέτηση των φοιτητών που βρίσκονταν εκτός Θεσσαλονίκης, αφενός για την καλύτερη εξυπηρέτησή τους και αφετέρου για την καλύτερη οργάνωση του τμήματος, οδήγησε στην δημιουργία αυτής της εφαρμογής, μέσω της οποίας οι φοιτητές θα μπορούν να πραγματοποιούν τις δηλώσεις των συγγραμμάτων, ηλεκτρονικά χωρίς να είναι αναγκασμένοι να παρευρίσκονται στο χώρο του ΑΤΕΙ. Έτσι σε περιόδους δηλώσεων η διαδικασία θα είναι ταχύτερη, θα γίνεται καλύτερος έλεγχος και φυσικά όλα αυτά θα πραγματοποιούνται απο τον υπολογιστή του κάθε φοιτητή ο οποίος θα βρίσκεται στον χώρο του. Αξίζει να τονίσουμε οτι με μερικές τροποποιήσεις, η εφαρμογή θα μπορεί να λειτουργήσει σε επαγγελματικό επίπεδο για τις καθημερινές ανάγκες της γραμματείας της σχολής και κατ'επέκταση του συνόλου των φοιτητών.

Περίληψη

Στα πλαίσια εκπόνησης της πτυχιακής αυτής, θα αναπτυχθεί μια διαδικτυακή εφαρμογή ενός συστήματος ηλεκτρονικής δήλωσης συγγραμμάτων, με χρήση των τεχνολογιών HTML, CSS, PHP, MySQL και JavaScript. Η εφαρμογή θα παρέχει σε κάθε χρήστη τη δυνατότητα να επιλέξει μέσα από μια λίστα μαθημάτων, τα συγγράμματα για το μάθημα που επιθυμεί, να στείλει ηλεκτρονικά την αίτηση με τις προτιμήσεις του στην γραμματεία του τμήματος και τέλος αφού εγκριθεί, να παραλάβει τα συγγράμματα απο τα αντίστοιχα βιβλιοπωλεία. Επιπλέον, η εφαρμογή αυτή θα δίνει την δυνατότητα στα αρμόδια άτομα, οι οποίοι θα είναι διαχειριστές του συστήματος, να επεξεργάζονται αυτά τα στοιχεία μέσα απο μια σελίδα διαχείρισης (Administrator page), να διεξάγουν τους απαραίτητους ελέγχους καθώς και να διαχειρίζονται την βάση δεδομένων. Τέλος η εφαρμογή θα αναρτηθεί σε μια πραγματική ιστοσελίδα για να είναι διαθέσιμη σε όλους.

Abstract

In the context of this dissertation, an internet application of a writings' electronic registration system will be developed making use of HTML, CSS, PHP, MySQL and JavaScript technologies. The application will provide to each user the possibility to choose within a list of courses the writings for a desired course, to send electronically his application about his or her preferences to the secretary of the department and finally, after approval, to collect the writings from the respective bookstores. In addition, this application will give the chance to the persons in charge, who will actually be the system's administrators, to assess these elements in an administrator page, to carry out the appropriate control procedures and to administer the database. Finally, the application will be posted on an actual webpage and be available to everyone.

Ευχαριστίες

Θέλω να ευχαριστήσω τον καθηγητή και επιβλέποντα της πτυχιακής μου κ.Βαφειάδη Αντώνη που με την εμπειρία του, την εκπαιδευτική του ικανότητα και την εξυπνάδα του με βοήθησε να οργανώσω την σκέψη μου, να εστιάσω πάνω στα προβλήματα που παρουσιάστηκαν κατά τη διαδικασία της ανάλυσης του συστήματος και να τα αντιμετωπίσω αποτελεσματικά από τη ρίζα τους. Τον ευχαριστώ επίσης για το κουράγιο που μου έδινε στις διάφορες φάσεις ανάλυσης και σχεδιασμού του συστήματος από την έναρξη του έργου μέχρι και το τέλος του.

Θέλω να ευχαριστήσω επίσης την Έλλη Παπαζαχαρίου, που με υπομονή και αφοσίωση μελέτησε την τεκμηρίωση της πτυχιακής μου εργασίας για τον πιθανό εντοπισμό συντακτικών και γραμματικών λαθών, τα περισσότερα από τα οποία έπρεπε να παραδεχτώ.

Τέλος, θέλω να ευχαριστήσω την οικογένεια μου που και αυτή συμμετείχε στο δύσκολο αλλά ανταποδοτικό έργο. Τόσο η ψυχολογική στήριξη όσο και η ανάγνωση του βιβλίου της πτυχιακής για τον εντοπισμό συντακτικών και γραμματικών λαθών συνέβαλαν στην τελειοποίηση της πτυχιακής με την οποία θα αποχαιρετίσω αυτό το κομμάτι της ζωής μου, το πανεπιστήμιο, προχωρώντας σε άλλο επίπεδο σπουδών, τις μεταπτυχιακές σπουδές ώστε να αποκτήσω εμπειρία και ειδικότητα.

Περιεχόμενα

Πρόλογος	3
Περίληψη	4
Abstract	5
Ευχαριστίες	6
Περιεχόμενα	7
Ευρετήριο σχημάτων και πινάκων	12
Εισαγωγή	14
ΠΡΩΤΟ ΜΕΡΟΣ - ΑΝΑΛΥΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	15
ΚΕΦΑΛΑΙΟ 1^ο	15
UML	15
1.1 Εισαγωγή	15
1.2 Ανάλυση	15
1.3 Όραμα	15
1.3.1 Σύντομη περιγραφή	15
1.3.2 Άποψη συστήματος	16
1.3.2.1 Χαρακτήρας εφαρμογής	16
1.3.3 Λειτουργίες εφαρμογής	17
1.3.3.1 Είσοδος στο σύστημα-Ταυτοποίηση στοιχείων	17
1.3.3.2 Αποσύνδεση απο το σύστημα	17
1.3.3.3 Επιλογή συγγραμμάτων	17
1.3.3.4 Δυνατότητα αλλαγής της επιλογής των συγγραμμάτων	17
1.3.3.5 Διατήρηση ιστορικών στοιχείων	17
1.3.3.6 Διαχείριση περιεχομένου	18
1.3.3.7 Απενεργοποίηση βιβλίων και μαθημάτων	18
1.3.3.8 Αλλαγή της θέσης του μαθήματος	18
1.3.3.9 Αντιστοίχιση των προτεινόμενων βιβλίων με τα μαθήματα	18
1.3.3.10 Ανάρτηση φωτογραφιών των συγγραμμάτων	18
1.3.3.11 Ενεργοποίηση/Απενεργοποίηση διαδικασίας επιλογής των βιβλίων	18
1.3.3.12 Εκκίνηση νέων δηλώσεων	18
1.3.3.13 Καταχώρηση φοιτητών	19
1.3.4 Επιχειρησιακό όφελος	19
1.4 Λειτουργικές απαιτήσεις	19
1.5 Χρήση εφαρμογής	19
1.6 Διερεύνηση υπαρχόντων συστημάτων	20
1.7 Πίνακας χρηστών-στόχων	20
1.8 Λεξικό όρων	20
1.9 Περιπτώσεις χρήσης - Διαγράμματα ακολουθίας συστήματος	21

1.9.1 Είσοδος στο σύστημα	25
1.9.2 Αποσύνδεση απο το σύστημα	26
1.9.3 Εγγραφή διαχειριστή	27
1.9.4 Αναζήτηση διαχειριστή	28
1.9.5 Ενημέρωση διαχειριστή	29
1.9.6 Διαγραφή διαχειριστή	31
1.9.7 Επιλογή συγγράμματος	33
1.9.8 Διαχείριση μαθημάτων - Εισαγωγή.....	34
1.9.9 Διαχείριση μαθημάτων - Αναζήτηση.....	35
1.9.10 Διαχείριση μαθημάτων - Ενημέρωση.....	36
1.9.11 Διαχείριση μαθημάτων - Διαγραφή.....	38
1.9.12 Διαχείριση συγγραμμάτων - Εισαγωγή.....	40
1.9.13 Διαχείριση συγγραμμάτων - Ενημέρωση.....	41
1.9.14 Διαχείριση συγγραμμάτων - Διαγραφή	43
1.9.15 Διαχείριση συγγραμμάτων - Αναζήτηση	44
1.9.16 Καταχώρηση φοιτητών	45
1.9.17 Ανοιγμα της διαδικασίας δηλώσεων	47
1.9.18 Κλείσιμο της διαδικασίας δηλώσεων	47
1.9.19 Εκκίνηση νέων δηλώσεων.....	48
1.10 Επίλογος	49
ΚΕΦΑΛΑΙΟ 2ο	50
Βάση δεδομένων	50
2.1 Εισαγωγή	50
2.2 Τι είναι MySQL	50
2.3 Κανονικοποίηση	50
2.3.1 Μη Κανονική Μορφή	50
2.3.2 Πρώτη Κανονική Μορφή (1NF)	51
2.3.3 Δεύτερη Κανονική Μορφή (2NF)	51
2.3.4 Τρίτη Κανονική Μορφή.....	51
2.4 Σχεσιακό Μοντέλο	52
2.5 Διάγραμμα Οντοτήτων – Συσχετίσεων (ER)	55
2.6 Τεκμηρίωση πινάκων	56
2.6.1 Περιγραφή πεδίων.....	56
2.7 SQL injection και προστασία βάσης δεδομένων.....	63
2.7.1 Αναποτελεσματικό φιλτράρισμα δεδομένων.....	63
2.7.2 Λανθασμένος χειρισμός τύπων δεδομένων.....	64
2.8 Αποφυγή SQL Injection.....	65
2.9 Επίλογος	65

ΔΕΥΤΕΡΟ ΜΕΡΟΣ - ΣΧΕΔΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	66
ΚΕΦΑΛΑΙΟ 3ο	66
HTML	66
3.1 Εισαγωγή	66
3.2 World Wide Web Consortium (W3C).....	66
3.3 Στοιχεία της HTML.....	66
3.4 Χρήση HTML στην εφαρμογή.....	67
3.4.1 Ετικέτα <html>	67
3.4.2 Ετικέτα <head>.....	67
3.4.3 Ετικέτα <link>	67
3.4.4 Ετικέτα <meta>	67
3.4.5 Ετικέτα <script>	67
3.4.6 Ετικέτα <style>	67
3.4.7 Ετικέτα <body>	68
3.4.8 Ετικέτα Κεφαλίδας	68
3.4.9 Ετικέτα <p>.....	68
3.4.10 Ετικέτα 	68
3.4.11 Ετικέτα 	68
3.4.12 Ετικέτα	68
3.4.13 Ετικέτα 	68
3.4.14 Ετικέτα <div>	69
3.4.15 Ετικέτα 	69
3.4.16 Ετικέτα 	69
3.4.17 Ετικέτα <a>.....	69
3.4.18 Ετικέτα <table>	69
3.4.19 Ετικέτα <form>.....	70
3.4.20 Ετικέτα <input>	70
3.4.21 Ετικέτα <select>	71
3.4.22 Ετικέτα <option>	71
3.5 Επίλογος	71
ΚΕΦΑΛΑΙΟ 4ο	72
CSS.....	72
4.1 Εισαγωγή	72
4.2 Διαχωρισμός HTML και εμφάνισης.....	72
4.3 Δομή – Παρουσίαση – Συμπεριφορά.....	72
4.4 Βασικές Αρχές του CSS	73
4.5 CSS Selectors	74
4.5.1 Παγκόσμιος Επιλογέας (Universal Selector)	74
4.5.2 Επιλογέας Στοιχείου (Element Selector)	74
4.5.3 Επιλογέας Κλάσης (Class Selector)	75

4.5.4 Επιλογέας Ταυτότητας (Id Selector)	75
4.5.5 Επιλογέας Ψευδοκλάσης (Pseudo Class Selector)	75
4.5.6 Επιλογέας απογόνου (Descendant Selector)	76
4.5.7 Συνδυασμός Επιλογέων (Combining Selectors).....	76
4.6 Επισύναψη Style Sheet στα έγγραφα.....	77
4.6.1 Ένθετα Στυλ (Inline Style)	77
4.6.2 Ενσωματωμένα Στυλ (Embedded Style Sheets)	77
4.6.3 Εξωτερικά Φύλλα Στυλ (External Style Sheet)	78
4.7 Επίλογος	78
ΚΕΦΑΛΑΙΟ 5ο	79
JAVASCRIPT	79
5.1 Εισαγωγή	79
5.2 JavaScript.....	79
5.3 Χρήσεις της JavaScript.....	80
5.4 Τοποθέτηση scripts σε ξεχωριστά αρχεία και ενσωμάτωση	81
5.5 Inline scripts	81
5.6 Επίλογος	82
ΚΕΦΑΛΑΙΟ 6ο	83
AJAX	83
6.1 Εισαγωγή	83
6.2 Τι είναι η τεχνολογία AJAX	83
6.3 Τρόπος λειτουργίας	83
6.4 Είναι όμως η τεχνολογία Ajax καινούρια;	86
6.5 Επίλογος	86
ΚΕΦΑΛΑΙΟ 7ο	87
PHP	87
7.1 Εισαγωγή	87
7.2 Ιστορία της PHP	87
7.3 Βασικά χαρακτηριστικά.....	87
7.4 Αρχιτεκτονική Βάσης Δεδομένων με PHP – MySQL	88
7.5 Apache Web Server	90
7.6 Hypertext Transfer Protocol (HTTP)	92
7.7 Εγκατάσταση εργαλείων.....	92
7.8 Επίλογος	92
ΚΕΦΑΛΑΙΟ 8ο	93
ΥΛΟΠΟΙΗΣΗ ΣΕ PHP	93
8.1 Εισαγωγή	93
8.2 Υπερκαθολικές μεταβλητές \$_GET και \$_POST.....	93
8.3 Η υπερκαθολική μεταβλητή \$_SESSION	93
8.4 Εκκίνηση μίας συνόδου λειτουργίας	94
8.5 Έλεγχος για το αν ο χρήστης είναι συνδεδεμένος.....	94
8.6 Σύνδεση με την βάση δεδομένων.....	95

8.7 Διαφυγή χαρακτήρων	96
8.8 Πρότυπο layout	97
8.9 Ακύρωση εγγραφής μεταβλητών και καταστροφή συνόδου	97
8.10 Διαγραφή μαθήματος	98
8.11 Διαγραφή διαχειριστή	98
8.12 Ανοιγμα/Κλείσιμο διαδικασίας δηλώσεων.....	99
8.13 Εκκίνηση νέων δηλώσεων.....	100
8.14 Εμφάνιση αποτελεσμάτων	101
8.15 Προσαρμογή της θέσης του μαθήματος στο εξάμηνο (SubjectRank)	101
8.16 Έλεγχος εγκυρότητας τιμών	103
8.17 Περιβάλλον – εφαρμογές – εργαλεία ανάπτυξης.....	103
8.18 Εναλλακτική υλοποίηση του συστήματος-Προτάσεις για μελλοντική βελτίωση	104
8.19 Αρχιτεκτονική MVC.....	104
8.20 Οφέλη.....	105
8.21 Βελτιώσεις – προσθήκες λειτουργιών στο υπάρχον σύστημα	105
8.22 Επίλογος	106
Βιβλιογραφία	107
ΠΑΡΑΡΤΗΜΑ Α - Εγκατάσταση των εργαλείων	108
ΠΑΡΑΡΤΗΜΑ Β - Εγχειρίδιο χρήσης εφαρμογής	114
ΠΑΡΑΡΤΗΜΑ Γ - Κώδικας SQL για την δημιουργία των πινάκων.....	126

Ευρετήριο Σχημάτων και Πινάκων

Πίνακες

Κεφάλαιο 1^ο

Πίνακας I:	Πίνακας χρηστών – στόχων	20
Πίνακας II:	Λεξικό όρων	20

Κεφάλαιο 5^ο

Πίνακας I:	Σύγκριση JavaScript και Java	79
------------	------------------------------------	----

Σχήματα

Κεφάλαιο 1^ο

Σχήμα I:	Διάγραμμα περιπτώσεων χρήσης εφαρμογής	22
Σχήμα II:	Διάγραμμα περίπτωσης χρήσης Επιλογή συγγραμμάτων	23
Σχήμα III:	Διάγραμμα περίπτωσης χρήσης Διαχείριση μαθημάτων	23
Σχήμα IV:	Διάγραμμα περίπτωσης χρήσης Διαχείριση συγγραμμάτων	24
Σχήμα V:	Διάγραμμα περίπτωσης χρήσης Διαχείριση διαχειριστών	24
Σχήμα VI:	Διάγραμμα ακολουθίας συστήματος Είσοδος στο σύστημα	26
Σχήμα VII:	Διάγραμμα ακολουθίας συστήματος Αποσύνδεση απο το σύστημα	27
Σχήμα VIII:	Διάγραμμα ακολουθίας συστήματος Εγγραφή διαχειριστή	28
Σχήμα IX:	Διάγραμμα ακολουθίας συστήματος Αναζήτηση διαχειριστή	29
Σχήμα X:	Διάγραμμα ακολουθίας συστήματος Ενημέρωση διαχειριστή	31
Σχήμα XI:	Διάγραμμα ακολουθίας συστήματος Διαγραφή διαχειριστή	32
Σχήμα XII:	Διάγραμμα ακολουθίας συστήματος Επιλογή συγγραμμάτων	34
Σχήμα XIII:	Διάγραμμα ακολουθίας συστήματος Εισαγωγή μαθήματος	35
Σχήμα XIV:	Διάγραμμα ακολουθίας συστήματος Αναζήτηση μαθήματος	36
Σχήμα XV:	Διάγραμμα ακολουθίας συστήματος Ενημέρωση μαθήματος	38
Σχήμα XVI:	Διάγραμμα ακολουθίας συστήματος Διαγραφή μαθήματος	39
Σχήμα XVII:	Διάγραμμα ακολουθίας συστήματος Εισαγωγή βιβλίου	41
Σχήμα XVIII:	Διάγραμμα ακολουθίας συστήματος Ενημέρωση βιβλίου	42
Σχήμα XIX:	Διάγραμμα ακολουθίας συστήματος Διαγραφή βιβλίου	44
Σχήμα XX:	Διάγραμμα ακολουθίας συστήματος Αναζήτηση βιβλίου	45
Σχήμα XXI:	Διάγραμμα ακολουθίας συστήματος Καταχώρηση φοιτητών	46
Σχήμα XXII:	Διάγραμμα ακολουθίας συστήματος Αλλαγή κατάστασης της διαδικασίας δηλώσεων	48
Σχήμα XXIII:	Διάγραμμα ακολουθίας συστήματος Εκκίνηση νέων δηλώσεων	49

Κεφάλαιο 2°

Σχήμα I:	Διάγραμμα ER εφαρμογής	55
Σχήμα II:	Η βάση δεδομένων, μέσα από το εργαλείο phpmyadmin	56

Κεφάλαιο 4°

Σχήμα I:	Τα τρία επίπεδα ενός Web εγγράφου	73
Σχήμα II:	Τα στοιχεία ενός CSS rule	73

Κεφάλαιο 6°

Σχήμα I:	Ο παραδοσιακός τρόπος αίτησης στον server.....	84
Σχήμα II:	Ο βασισμένος σε Ajax τρόπος αίτησης στον server.....	84
Σχήμα III:	Διαδοχικά requests/responses σε μία Ajax based εφαρμογή.....	85

Κεφάλαιο 7°

Σχήμα I:	Η σχέση browser / server.....	89
Σχήμα II:	Βασική Web αρχιτεκτονική βάσεων δεδομένων	89
Σχήμα III:	Μερίδιο ανάμεσα στους Server(Αύγουστος 1995 - Αύγουστος 2010)	91
Σχήμα IV:	Σύγκριση ανάλυση της χρήσης server ανά εταιρεία	91

Κεφάλαιο 8°

Σχήμα I:	Αναπαράσταση αρχιτεκτονικής MVC	105
----------	---------------------------------------	-----

Εισαγωγή

Με την πάροδο του χρόνου αναπτύσσεται ολοένα και περισσότερο το διαδίκτυο και οι υπηρεσίες που προσφέρει. Σημαντική είναι η εξέλιξη του διαδικτύου και στα εκπαιδευτικά ιδρύματα. Ολοένα και περισσότερα πανεπιστήμια χρησιμοποιούν την υπηρεσία του παγκόσμιου ιστού, προκειμένου να βελτιώνουν την εκπαιδευτική δραστηριότητα. Τηλεκπαίδευση, blogs, σελίδες ανακοινώσεων βαθμολογιών, ηλεκτρονικές δηλώσεις μαθημάτων και συγγραμμάτων είναι μερικές από τις υπηρεσίες που προσφέρονται δωρεάν στους φοιτητές για την καλύτερη εξυπηρέτησή τους.

Η συγκεκριμένη πτυχιακή αφορά στην δημιουργία ενός συστήματος ηλεκτρονικής δήλωσης των συγγραμμάτων του εκπαιδευτικού ιδρύματος.

Για τους φοιτητές του τμήματος Πληροφορικής οι οποίοι δεν μένουν στην πόλη που βρίσκεται το ΑΤΕΙ, είναι πολύ δύσκολο να κάνουν μια δήλωση βιβλίων ή ακόμα και να τροποποιήσουν μια υπάρχουσα. Για να γίνει κάτι τέτοιο θα πρέπει να απευθυνθούν αυτοπροσώπως στην γραμματεία ή κάποιο εξουσιοδοτημένο πρόσωπο να κάνει αίτηση για δήλωση βιβλίων. Κάτι τέτοιο όμως δεν είναι πάντα εύκολο ειδικά στην περίπτωση που πρέπει να γίνει άμεσα, καθώς απαιτούνται χρόνος και πιθανόν κάποια πιστοποιητικά. Παράλληλα, με τον τρόπο αυτό θα υπάρχει δραστική μείωση της απαιτούμενης εργασίας από τους υπαλλήλους της Γραμματείας, καθώς ένα αρκετά μεγάλο ποσοστό της γραφειοκρατικής εργασίας θα σταματήσει να υφίσταται.

Στόχος της εφαρμογής αυτής είναι η εξάλειψη όλων των παραπάνω προβλημάτων, καθώς και η εξυπηρέτηση με τον καλύτερο δυνατό τρόπο των αναγκών του εκπαιδευτικού ιδρύματος σε φοιτητές.

Στα κεφάλαια που ακολουθούν περιγράφεται η ανάλυση και ο σχεδιασμός του συστήματος από την πρώτη μέχρι και την τελευταία στιγμή. Η τεκμηρίωση χωρίζεται σε δύο μέρη.

Στο πρώτο μέρος, αυτό της ανάλυσης γίνεται λεπτομερής περιγραφή των διαφόρων λειτουργιών του συστήματος με διαγράμματα UML. Επίσης γίνεται ανάλυση της βάσης δεδομένων της εφαρμογής με την παρουσίαση των σχετικών πινάκων, την επεξήγησή τους και το διάγραμμα οντοτήτων συσχετίσεων για την καλύτερη κατανόηση της εφαρμογής.

Το δεύτερο μέρος, αφορά την σχεδίαση της εφαρμογής, όπου παρουσιάζονται οι διάφορες τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίησή της. Επίσης παρουσιάζονται τα σημαντικότερα τμήματα κώδικα με σχετική επεξήγηση για να γίνονται κατανοητά.

Τέλος ακολουθούν η εγκατάσταση των απαραίτητων εργαλείων για την λειτουργία της εφαρμογής και ο οδηγός χρήσης της.

ΠΡΩΤΟ ΜΕΡΟΣ - ΑΝΑΛΥΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

ΚΕΦΑΛΑΙΟ 1^ο

UML

1.1 Εισαγωγή

Στο πρώτο μέρος της τεκμηρίωσης γίνεται η παρουσίαση της διαδικασίας ανάλυσης του συστήματος. Το στάδιο της ανάλυσης είναι ίσως το βασικότερο σημείο κατά την διαδικασία ανάπτυξης ενός project, καθώς σε αυτό καθορίζονται οι βασικοί στόχοι της εφαρμογής και οι απαιτήσεις της. Σε αυτό το κεφάλαιο γίνεται ανάλυση του συστήματος με διαγράμματα UML, ώστε να γίνει κατανοητή η λειτουργία του.

1.2 Ανάλυση

Η κατανόηση των απαιτήσεων που οδηγούν στην δημιουργία ενός νέου συστήματος είναι το κυριότερο μέρος της διαδικασίας κατασκευής. Το κλειδί για την ανάπτυξη αξιόπιστων λειτουργικών εφαρμογών είναι η κατανόηση σε όσο δυνατόν μεγαλύτερο βαθμό των απαιτήσεων της εφαρμογής. Η ανάλυση των απαιτήσεων και ο πρότερος σχεδιασμός του συστήματος πριν από την κωδικοποίηση επιτρέπει την πλήρη κατανόηση των λειτουργικών και χρηστικών απαιτήσεων όπως επίσης και την ανακάλυψη και πρόβλεψη πιθανών λαθών και την επίλυσή τους πολύ πριν το στάδιο της συγγραφής του κώδικα, όπου η διόρθωσή τους πλέον καθίσταται ιδιαίτερα δαπανηρή.

Στο στάδιο της ανάλυσης ακολουθείται η μεθοδολογία RUP (Rational Unified Process) η οποία καθορίζει ένα ενιαίο πλαίσιο μοντελοποίησης με χρήση της γλώσσας UML και ευρέως χρησιμοποιούμενων διεργασιών που διευκολύνουν την διαδικασία της ανάλυσης και του σχεδιασμού του συστήματος.

1.3 Όραμα

Στο σημείο αυτό καθορίζονται οι πρωταρχικοί στόχοι της εφαρμογής.

1.3.1 Σύντομη περιγραφή

Η ζητούμενη εφαρμογή αφορά την διαχείριση των συγγραμμάτων του ΑΤΕΙ. Το σύστημα θα παρέχει την δυνατότητα σε κάθε χρήστη ο οποίος διαθέτει λογαριασμό, είναι δηλαδή ενεργός φοιτητής του ΑΤΕΙ, να πραγματοποιεί ηλεκτρονικά τις δηλώσεις των βιβλίων που δικαιούται. Πιο συγκεκριμένα σε περιόδους ηλεκτρονικών δηλώσεων οι οποίες ορίζονται από το εκπαιδευτικό ίδρυμα, ο φοιτητής θα έχει την δυνατότητα αφού ταυτοποιήσει τα στοιχεία του στο σύστημα, να επιλέγει μέσα από μια λίστα μαθημάτων τα συγγράμματα που επιθυμεί και να αποστέλει ηλεκτρονικά την δήλωση με τις προτιμήσεις του στην γραμματεία του τμήματος. Η γραμματεία με τη σειρά της αφού επιτελέσει όλους τους απαραίτητους ελέγχους σχετικά με τον εάν ο φοιτητής έχει δικαίωμα να παραλάβει τα βιβλία που δήλωσε θα στέλνει τα απαραίτητα έγγραφα στα αντίστοιχα βιβλιοπωλεία και ο φοιτητής θα μπορεί να τα παραλάβει εφόσον υπάρχει το όνομα του στις λίστες.

Σε περίπτωση που ο φοιτητής μετανιώσει για τις επιλογές του και εφόσον δεν έχει λήξει η περίοδος των δηλώσεων θα μπορεί να αλλάξει την επιλογή του και να ξαναστείλει νέα αίτηση στην γραμματεία του τμήματος.

Επίσης η εφαρμογή θα παρέχει τη δυνατότητα στα αρμόδια άτομα να διαχειρίζονται μέσω μιας κύριας σελίδας διαχείρισης (Administrator page) όλα τα συγγράμματα που διανέμονται για κάθε μάθημα. Επιπλέον επιτρέπει την διαχείριση των μαθημάτων του κάθε εξαμήνου καθώς και των διαχειριστών του συστήματος. Δοκιμαστικά για την σωστή λειτουργία και τον έλεγχο της εφαρμογής υπάρχει η δυνατότητα καταχώρησης φοιτητών.

1.3.2 Άποψη συστήματος

1.3.2.1 Χαρακτήρας εφαρμογής

Η εφαρμογή θα πρέπει να είναι διαδικτυακή. Οι λόγοι για τη συγκεκριμένη επιλογή είναι αρκετοί:

- **Απομακρυσμένη πρόσβαση:** οποιοσδήποτε χρήστης σε οποιοδήποτε μέρος κι αν βρίσκεται μπορεί να έχει πρόσβαση στην εφαρμογή και να χρησιμοποιεί τις υπηρεσίες που προσφέρει.
- **Παράκαμψη ζητημάτων εγκατάστασης:** η εφαρμογή εκτελείται μέσα από έναν απλό browser χωρίς να δημιουργεί προβλήματα ασυμβατότητας εφόσον είναι ανεξάρτητη από το λειτουργικό σύστημα ή τις δυνατότητες επεξεργασίας του εκάστοτε τερματικού.
- **Ταχύτητα συνδέσεων:** το “γρήγορο Internet” δεν αποτελεί πλέον ουτοπία καθώς οι συνδέσεις ADSL γίνονται όλο και περισσότερο δημοφιλείς και δεν συνιστούν εμπόδιο στην πρόσβαση στην εφαρμογή.
- **Καινούριες τεχνολογίες:** η αξιοπιστία και η λειτουργικότητα των διαδικτυακών εφαρμογών αυξάνει διαρκώς καθώς νέες τεχνολογίες κάνουν συνεχώς την εμφάνισή τους, προσφέροντας στους χρήστες την εμπειρία των πλούσιων, από άποψη παρεχόμενων υπηρεσιών, αλλά και εύκολων στην χρήση εφαρμογών. Ταυτόχρονα, από τη μεριά των προγραμματιστών οι νέες τεχνολογίες προσφέρουν την δυνατότητα της εύκολης συγγραφής κώδικα με παράλληλη αύξηση της εκμετάλλευσης ολοένα και περισσότερων δυνατοτήτων των υπολογιστικών συστημάτων.
- **Ευκολία μάθησης:** είναι γενικά παραδεκτό ότι υπάρχει μεγαλύτερη δυσκολία στην εκμάθηση μιας desktop εφαρμογής στην οποία συνήθως υπάρχουν πολλά μενού χωρίς επεξηγήσεις. Αντίθετα η πλοήγηση ανάμεσα σε ιστοσελίδες στις οποίες η ροή είναι καθορισμένη και σαφής και η λειτουργία της κάθε σελίδας είναι ξεκάθαρη, καθιστά τη χρήση της πιο εύκολη αλλά και πιο αποτελεσματική.
- **Ελκυστική διεπαφή:** συγκριτικά με τις desktop εφαρμογές, σε μια ιστοσελίδα υπάρχουν περισσότερες δυνατότητες για τη δημιουργία μιας αισθητικά ανώτερης διεπαφής.
- **Ενιαία διαχείριση:** σημαντικό κριτήριο αποτελεί η διαχείριση της εφαρμογής η οποία πρέπει να γίνεται μέσα από μια ενιαία διεπαφή που θα προσφέρει έλεγχο σε όλο το σύστημα και στις επιμέρους λειτουργίες. Η χρήση μιας διαδικτυακής εφαρμογής απαλλάσσει τον διαχειριστή από την διαχείριση κάθε τερματικού ξεχωριστά. Οι διαδικτυακές εφαρμογές φυσικά έχουν και μειονεκτήματα. Ένα μειονέκτημα είναι η ελάχιστη εκμετάλλευση των πόρων του τερματικού. Η μόνη απαίτηση για την εκτέλεση της εφαρμογής είναι ο browser του τερματικού, λογισμικό το οποίο δεν καταναλώνει ιδιαίτερους πόρους του συστήματος αλλά και δεν επιτρέπει την εκμετάλλευσή τους προς όφελος της εφαρμογής. Ακολουθώντας

τα παραπάνω, συνάγεται το συμπέρασμα ότι όλη η επεξεργαστική ισχύς που απαιτείται για την εκτέλεση της εφαρμογής πρέπει να παρέχεται από τον server. Οπότε ο server εξορισμού δέχεται μεγάλο φόρτο δεδομένων και η εφαρμογή πρέπει να περιορίζει αυτόν το φόρτο όσο το δυνατόν περισσότερο. Αναλογικά με τα μειονεκτήματα, τα προτερήματα μιας διαδικτυακής εφαρμογής δικαιολογούν απόλυτα την επιλογή της συγκεκριμένης στρατηγικής.

1.3.3 Λειτουργίες εφαρμογής

Παρακάτω παρουσιάζονται συνοπτικά οι σημαντικότερες λειτουργίες του συστήματος ώστε να μπορέσει ο αναγώστης να κατανοήσει ευκολότερα την λειτουργία του.

1.3.3.1 Είσοδος στο σύστημα-Ταυτοποίηση στοιχείων

Η πρώτη σελίδα που θα εμφανίζεται στον χρήστη του συστήματος αφορά την ταυτοποίηση των στοιχείων του. Θα ζητείται ένα όνομα χρήστη και ένας κωδικός πρόσβασης (username και password) για την είσοδο στο σύστημα. Σε περίπτωση που ο χρήστης δεν διαθέτει λογαριασμό που σημαίνει ότι δεν είναι φοιτητής του ΑΤΕΙ, δεν θα του επιτρέπεται η πρόσβαση. Την ταυτοποίηση των στοιχείων τους θα πρέπει να κάνουν και οι διαχειριστές του συστήματος με παρόμοιο τρόπο όπως ακριβώς συμβαίνει και με τους φοιτητές.

1.3.3.2 Αποσύνδεση από το σύστημα

Στο τέλος κάθε δραστηριότητας στο σύστημα οι φοιτητές και οι διαχειριστές μπορούν να αποσυνδέονται για μεγαλύτερη ασφάλεια.

1.3.3.3 Επιλογή συγγραμμάτων

Η αρχική σελίδα της εφαρμογής αφορά την επιλογή συγγραμμάτων. Μετά την ταυτοποίηση των στοιχείων του, ο φοιτητής θα έχει τη δυνατότητα να επιλέξει τα συγγράμματα που επιθυμεί για τα μαθήματα που έχει δηλώσει.

Η κύρια σελίδα θα περιέχει όλα τα μαθήματα ταξινομημένα ανά εξάμηνο. Αφού ο φοιτητής επιλέξει το εξάμηνο που επιθυμεί, θα μπορεί να βλέπει τα διαθέσιμα συγγράμματα για κάθε μάθημα. Με την βοήθεια radiobuttons θα επιλέγει τα βιβλία που επιθυμεί και θα ολοκληρώνει την διαδικασία πατώντας το κουμπί «Επιλογή βιβλίων». Οι προτιμήσεις του θα αποθηκεύονται στην βάση δεδομένων.

1.3.3.4 Δυνατότητα αλλαγής της επιλογής των συγγραμμάτων

Το σύστημα θα δίνει τη δυνατότητα στο χρήστη αφού έχει στείλει ήδη μια φορά την ηλεκτρονική αίτηση βιβλίων, να ξαναδηλώσει τα βιβλία που επιθυμεί (εφόσον βρίσκεται μέσα στην περίοδο των ηλεκτρονικών δηλώσεων) σε περίπτωση που μετανιώσει για κάποια επιλογή που έκανε, ή σε περίπτωση λάθους. Η παλιά του αίτηση θα διαγράφεται και θα αποθηκεύεται η νέα που θα είναι και η πιο πρόσφατη. Όταν κλείσει η διαδικασία των δηλώσεων ο φοιτητής θα μπορεί μόνο να βλέπει ποια βιβλία δικαιούται και δεν θα έχει τη δυνατότητα διόρθωσης.

1.3.3.5 Διατήρηση ιστορικών στοιχείων

Το σύστημα θα διατηρεί στην βάση δεδομένων ιστορικά στοιχεία που θα αφορούν ποιά βιβλία έχει πάρει ένας συγκεκριμένος φοιτητής στο παρελθόν, ώστε να ελέγχεται αν έχει το δικαίωμα να ξαναπάρει κάποιο βιβλίο. Έτσι ένας φοιτητής που έχει πάρει κάποιο σύγγραμμα στο παρελθόν δεν θα μπορεί να δηλώσει ξανά το

ίδιο.Ο έλεγχος θα γίνεται απο το σύστημα σαρώνοντας την βάση δεδομένων με τα ιστορικά στοιχεία και ανάλογα θα δίνεται το δικαίωμα ή όχι στον φοιτητή.

1.3.3.6 Διαχείριση περιεχομένου

Ο διαχειριστής της εφαρμογής έχει την δυνατότητα της διαχείρισης όλων των μονάδων της εφαρμογής με την εισαγωγή, ενημέρωση και διαγραφή αντίστοιχων οντοτήτων. Πιο συγκεκριμένα οι επιμέρους τομείς της διαχείρισης είναι:

1. Διαχείριση συγγραμμάτων
2. Διαχείριση μαθημάτων
3. Διαχείριση διαχειριστών
4. Καταχώρηση φοιτητών
5. Ενεργοποίηση/Απενεργοποίηση της διαδικασίας δήλωσης βιβλίων
6. Εκκίνηση νέων ηλεκτρονικών δηλώσεων

1.3.3.7 Απενεργοποίηση βιβλίων και μαθημάτων

Τα μαθήματα και τα βιβλία δεν διαγράφονται απο την βάση δεδομένων για λόγους ασφαλείας και διατήρησης ιστορικών στοιχείων.Ως εκ τούτου η εφαρμογή δίνει τη δυνατότητα απενεργοποίησης αυτών των οντοτήτων αν αποφασιστεί οτι δεν θα χρησιμοποιηθούν και επαναφοράς τους αν χρειασθούν ξανά στο μέλλον.

1.3.3.8 Αλλαγή της θέσης του μαθήματος

Μια από τις δυνατότητες της εφαρμογής είναι οτι επιτρέπει να αλλάζουμε την θέση των μαθημάτων ενός εξαμήνου ή ακόμα να αλλάξουμε και το εξάμηνο του μαθήματος.Ετσι αν υπάρξει κάποιο πιθανό λάθος κατα την εισαγωγή ενός μαθήματος,αυτό διορθώνεται εύκολα «τακτοποιώντας» τα μαθήματα με την σειρά που επιθυμούμε.

1.3.3.9 Αντιστοίχιση των προτεινόμενων βιβλίων με τα μαθήματα

Ο διαχειριστής αφού εισάγει στην βάση δεδομένων τα μαθήματα και τα αντίστοιχα συγγράμματα,θα έχει την δυνατότητα μέσα απο μια λίστα συγγραμμάτων να αντιστοιχίζει κάθε βιβλίο με κάθε μάθημα.

1.3.3.10 Ανάρτηση φωτογραφιών των συγγραμμάτων

Ο διαχειριστής του συστήματος έχει τη δυνατότητα να ανεβάζει φωτογραφίες με τα εξώφυλλα των συγγραμμάτων ώστε ο φοιτητής να έχει μια καλύτερη εικόνα για το βιβλίο που επιλέγει.

1.3.3.11 Ενεργοποίηση/Απενεργοποίηση διαδικασίας επιλογής των βιβλίων

Ο διαχειριστής του συστήματος έχει τη δυνατότητα να ενεργοποιεί ή να απενεργοποιεί αντίστοιχα την διαδικασία επιλογής των διαθέσιμων συγγραμμάτων σε περιόδους ηλεκτρονικών δηλώσεων και μή.Οταν η διαδικασία είναι ανοιχτή οι φοιτητές θα μπορούν να δηλώνουν όσες φορές θέλουν τα βιβλία που επιθυμούν.Μόλις η διαδικασία κλείσει οι φοιτητές μπορούν απλώς αφού εισέλθουν στο σύστημα να βλέπουν ποιά συγγράμματα δικαιούνται.

1.3.3.12 Εκκίνηση νέων δηλώσεων

Ο διαχειριστής έχει την δυνατότητα της εκκίνησης νέων ηλεκτρονικών δηλώσεων.Το σύστημα σε κάθε επανέναρξη της διαδικασίας αποθηκεύει ιστορικά

στοιχεία στην βάση δεδομένων και οι φοιτητές μπορούν να ξεκινήσουν την διαδικασία των ηλεκτρονικών δηλώσεων.

1.3.3.13 Καταχώρηση φοιτητών

Ο διαχειριστής της εφαρμογής έχει την δυνατότητα να καταχωρεί στην βάση δεδομένων, νέους φοιτητές. Σημειώνεται ότι αυτή η λειτουργία είναι «δοκιμαστική» έτσι ώστε να υπάρχουν στοιχεία φοιτητών στο σύστημα για να μπορεί να πραγματοποιηθεί η διαδικασία ηλεκτρονικών δηλώσεων. Σε άλλη περίπτωση τα στοιχεία των φοιτητών θα αντλούνται από την γραμματεία του τμήματος και η συγκεκριμένη εφαρμογή δεν θα παρέχει δυνατότητα διαχείρισης φοιτητών.

1.3.4 Επιχειρησιακό όφελος

Τα οφέλη που αποκομίζει το ΑΤΕΙ από την ανάπτυξη και λειτουργία μίας τέτοιου είδους εφαρμογής συνοψίζονται στα ακόλουθα:

- ✓ Ταχεία και αποτελεσματική εξυπηρέτηση φοιτητών: οι δυσαρεστημένοι φοιτητές αποτελούν ένα από τα βασικά μειονεκτήματα ενός εκπαιδευτικού ιδρύματος. Η δημιουργία και η χρήση μιας τέτοιας εφαρμογής ανεβάζει το επίπεδο του τμήματος, καθώς οι φοιτητές που ανήκουν σε αυτό εξυπηρετούνται με τον καλύτερο δυνατό τρόπο, ολοκληρώνοντας την διαδικασία των δηλώσεων από οποιοδήποτε μέρος και αν βρίσκονται αρκεί να υπάρχει ένας Η/Υ με σύνδεση στο δίκτυο.
- ✓ Μείωση λειτουργικών εξόδων: όλη η διαδικασία των δηλώσεων των συγγραμμάτων (αιτήσεις, αποδεικτικά αιτήσεων) ελέγχεται ηλεκτρονικά μειώνοντας το λειτουργικό κόστος που δημιουργεί η παραδοσιακή μέθοδος (εκτυπώσεις σε χαρτί κλπ).
- ✓ Μικρό κόστος αναβάθμισης και συντήρησης: η σχεδίαση της εφαρμογής επιτρέπει την εύκολη αναβάθμισή της σε περίπτωση επέκτασης αλλά και ο διαδικτυακός χαρακτήρας της διευκολύνει τη συντήρησή της.
- ✓ Αποτελεσματικότερος έλεγχος: Πιθανοί έλεγχοι νομιμότητας των επιλογών των φοιτητών τώρα καθίστανται ευκολότεροι και αυτόματοι, αφού πραγματοποιούνται πλέον από το σύστημα και όχι από την γραμματεία του πανεπιστημιακού ιδρύματος.
- ✓ Απομακρυσμένη πρόσβαση: Η πρόσβαση στην εφαρμογή και όλη η διαδικασία των ηλεκτρονικών δηλώσεων μπορεί να πραγματοποιηθεί από οποιοδήποτε μέρος αρκεί ο χρήστης να διαθέτει τον επιθυμητό κωδικό για την ταυτοποίηση των στοιχείων του και έναν Η/Υ με σύνδεση στο δίκτυο.

1.4 Λειτουργικές απαιτήσεις

Όσον αφορά τα τεχνικά ζητήματα ανάπτυξης της εφαρμογής, υπάρχει ιδιαίτερη προτίμηση στις ανοικτές τεχνολογίες. Αφενός το κόστος είναι ελάχιστο ή μηδενικό και αφετέρου υπάρχει η δυνατότητα εύκολης εύρεσης βοήθειας σε τεχνικά προβλήματα μέσα από την διαδικτυακή κοινότητα.

1.5 Χρήση εφαρμογής

Από την διαδικασία της ανάλυσης καθορίζονται δύο κατηγορίες χρηστών που αλληλεπιδρούν με την εφαρμογή:

- Ο φοιτητής του εκπαιδευτικού ιδρύματος ο οποίος έχει έναν προσωπικό λογαριασμό μέσω του οποίου υποβάλλει αιτήσεις ηλεκτρονικά προς την

γραμματεία του τμήματος, για την παραλαβή των συγγραμμάτων του κάθε εξαμήνου.

- Ο διαχειριστής του συστήματος ο οποίος είναι υπεύθυνος για την διαχείριση όλων των μονάδων της εφαρμογής με την εισαγωγή, ενημέρωση και διαγραφή αντίστοιχων οντοτήτων.

1.6 Διερεύνηση υπαρχόντων συστημάτων

Ο κύριος στόχος αυτής της δραστηριότητας είναι να μελετηθούν εφαρμογές και συστήματα τα οποία υπάρχουν ήδη στο εμπόριο, κι έτσι να γίνει ένας αρχικός καθορισμός των απαιτήσεων και λειτουργιών της εφαρμογής, ώστε να αποτελέσει το υπόβαθρο για το στάδιο της σχεδίασης και ανάλυσης. Έτσι συλλέγονται πολύτιμες πληροφορίες σχετικά με την τεχνική φύση του προβλήματος, όπως ποιές είναι οι τεχνολογίες που απαιτούνται, πώς είναι οργανωμένες παρόμοιες εφαρμογές, καθώς και τι θέση έχουν στην αγορά. Όλα τα παραπάνω θα βοηθήσουν πάρα πολύ σχετικά με την ανάλυση και σχεδίαση του συστήματος, αφού δίνουν μια πρώτη ιδέα για το πώς πρέπει να είναι η εφαρμογή και τι περίπου θα χρειαστεί για την ανάπτυξή της.

1.7 Πίνακας χρηστών-στόχων

Πίνακας I: Πίνακας χρηστών – στόχων

Κατηγορία χρήστη	Στόχος χρήστη
Φοιτητής	-Επιλογή συγγραμμάτων ανά μάθημα -Αποστολή ηλεκτρονικής αίτησης
Διαχειριστής	-Διαχείριση μαθημάτων -Διαχείριση συγγραμμάτων -Διαχείριση διαχειριστών -Καταχώρηση φοιτητών -Ανοιγμα/Κλείσιμο της διαδικασίας επιλογής βιβλίων -Εκκίνηση νέων ηλεκτρονικών δηλώσεων

1.8 Λεξικό όρων

Πίνακας II: Λεξικό όρων

Όρος	Περιγραφή
Χρήστης	Φοιτητής του εκπαιδευτικού ιδρύματος.
Διαχειριστής	Διαχειριστής της εφαρμογής. Επιβλέπει όλες τις διαδικασίες και ελέγχει τη σωστή λειτουργία του συστήματος. Μπορεί να υπάρχουν πολλοί.

Ορος	Περιγραφή
Σύγγραμμα	Βιβλίο που διανέμεται για κάποιο μάθημα του εκπαιδευτικού ιδρύματος.
Μάθημα	Μάθημα του εκπαιδευτικού ιδρύματος. Θεωρία ή εργαστήριο, υποχρεωτικό ή προαιρετικό.
Επιλογή συγγραμμάτων	Η διαδικασία επιλογής των διαθέσιμων βιβλίων του τμήματος, μέσα από μια λίστα μαθημάτων. Πραγματοποιείται από τον φοιτητή του τμήματος ηλεκτρονικά από την εφαρμογή.
Ταυτοποίηση στοιχείων	Μηχανισμός ελέγχου εγκυρότητας των στοιχείων (όνομα χρήστη, κωδικός πρόσβασης) του φοιτητή και του διαχειριστή.
Διαχείριση	Οποιαδήποτε διαδικασία εισαγωγής, ενημέρωσης ή διαγραφής κάποιων οντοτήτων.
Ανενεργό σύγγραμμα	Είναι το σύγγραμμα το οποίο είναι απενεργοποιημένο, είναι δηλαδή σε κατάσταση active='no'
Άνοιγμα/κλείσιμο/ Εκκίνηση νέας διαδικασίας ηλεκτρονικών δηλώσεων	Μια διαδικασία είναι ανοιχτή αν οι φοιτητές μπορούν να δηλώνουν βιβλία. Μια διαδικασία είναι κλειστή αν οι φοιτητές δεν έχουν το δικαίωμα δήλωσης αλλά μπορούν να παρακολουθούν ποιά βιβλία τους έχουν ανατεθεί. Η εκκίνηση νέας διαδικασίας αφορά την επανεκκίνηση της διαδικασίας των δηλώσεων και την συλλογή ιστορικών στοιχείων από παλαιότερες ηλεκτρονικές δηλώσεις.

1.9 Περιπτώσεις χρήσης - Διαγράμματα ακολουθίας συστήματος

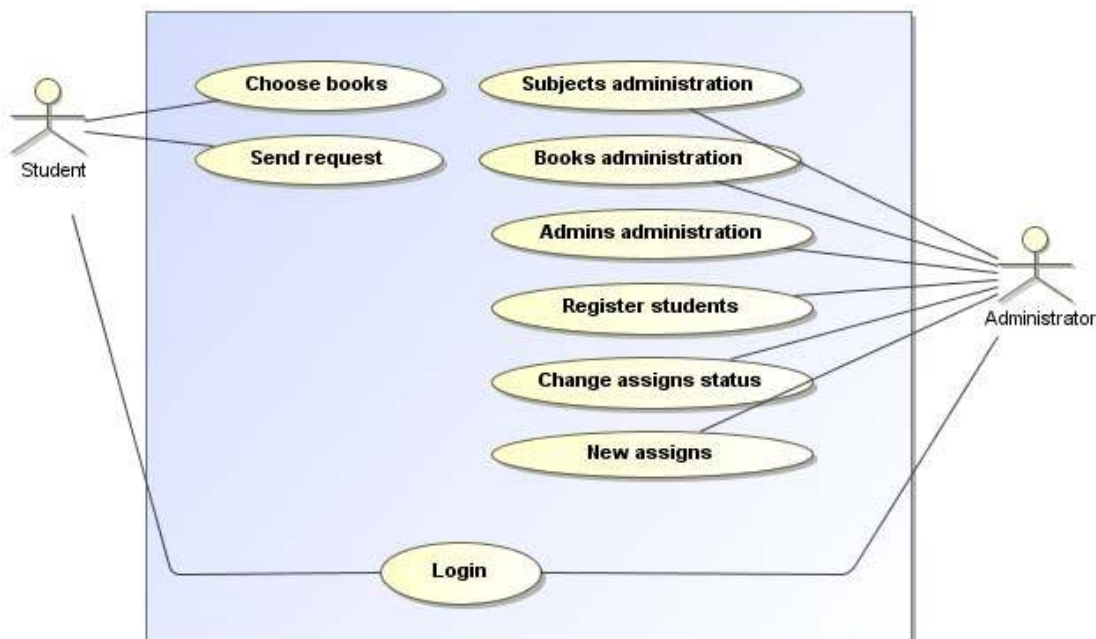
Σε οποιοδήποτε μοντέλο κύκλου ζωής λογισμικού, ένα από τα πρώτα στάδια είναι πάντοτε ο προσδιορισμός των απαιτήσεων. Στόχος είναι αφενός η καταγραφή και αφετέρου η λεπτομερής ανάλυση τους ώστε να προκύψει ένα έγγραφο με βάση το οποίο θα πραγματοποιηθεί η ανάπτυξη του λογισμικού. Το έγγραφο αυτό θα πρέπει να προσδιορίζει με όσο το δυνατό μεγαλύτερη σαφήνεια τις απαιτήσεις, να είναι πλήρες, να μην έχει αντιφάσεις και να είναι εύκολα κατανοητό και τροποποιήσιμο.

Τα διαγράμματα περιπτώσεων χρήσης (use case diagrams) στη UML χρησιμοποιούνται για τη μοντελοποίηση της συμπεριφοράς ενός συστήματος, υποσυστήματος ή κλάσης όπως αυτή γίνεται αντιληπτή από τον εξωτερικό χρήστη. Τα διαγράμματα αυτά διαμερίζουν τη λειτουργικότητα του συστήματος σε συναλλαγές που έχουν νόημα για τους χαρακτήρες (actors)-ιδανικούς χρήστες του συστήματος.

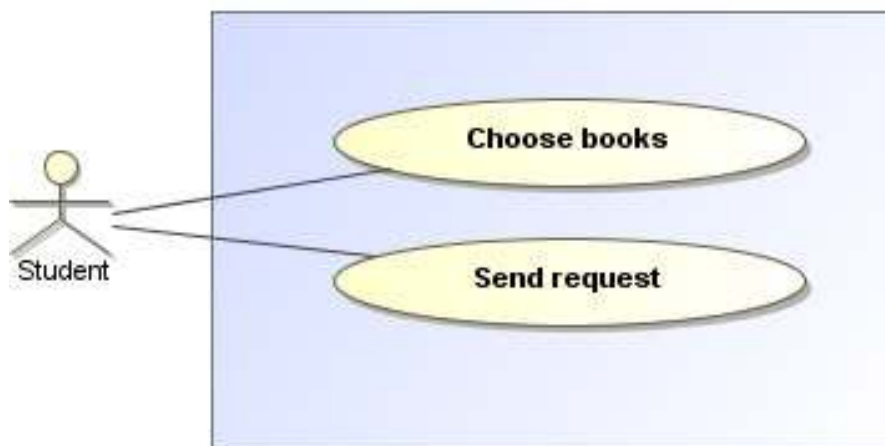
Η τεκμηρίωση κάθε περίπτωσης χρήσης για να είναι πλήρης, θα πρέπει να περιλαμβάνει μια ακολουθία γεγονότων που λαμβάνουν χώρα για την υλοποίηση της επιθυμητής συμπεριφοράς. Ένα τυπικό πρότυπο για την τεκμηρίωση μιας περίπτωσης χρήσης περιλαμβάνει τα ακόλουθα στοιχεία:

1. **Σύντομη περιγραφή:** Περιγραφή της επιθυμητής συμπεριφοράς των χρηστών της συγκεκριμένης περίπτωσης χρήσης.
2. **Προ-συνθήκες:** Συνθήκες που πρέπει να ισχύουν ώστε να είναι δυνατή η έναρξη της περίπτωσης χρήσης.
3. **Κύρια ροή γεγονότων:** Περιγραφή υπό μορφή κειμένου της ακολουθίας των γεγονότων που λαμβάνουν χώρα για την υλοποίηση της συμπεριφοράς.
4. **Εναλλακτική ροή γεγονότων:** Περιγραφή εξαιρέσεων ή λανθασμένων καταστάσεων.
5. **Μετά-συνθήκες:** Συνθήκες που θα ισχύουν μετά την ομαλή εκτέλεση της εν λόγω περίπτωσης χρήσης.

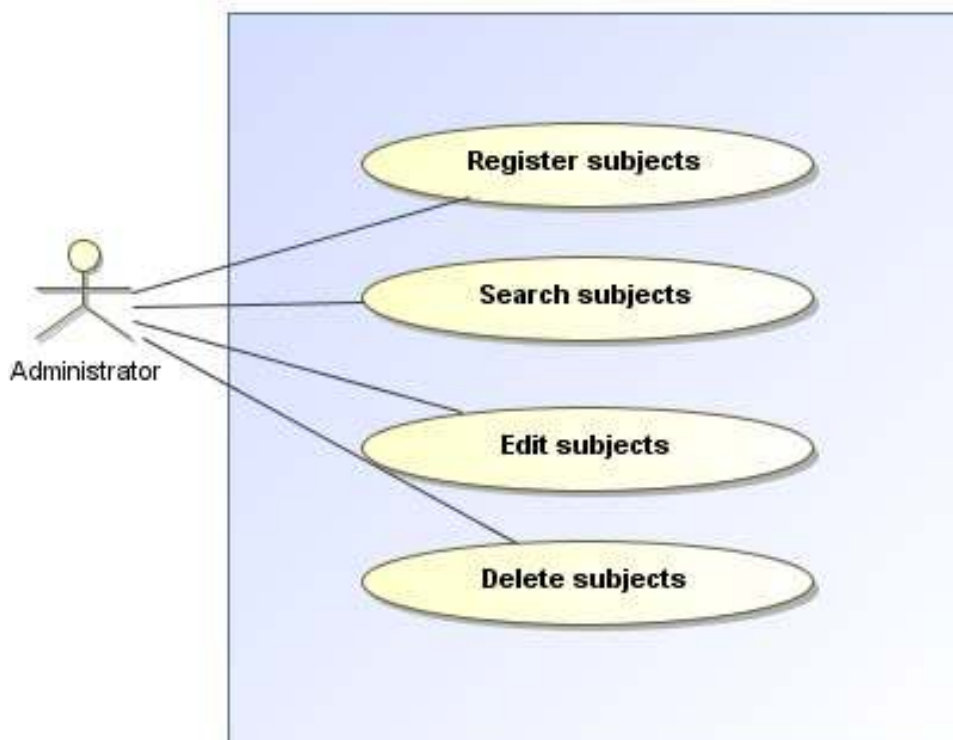
Σε αυτό το σημείο παρουσιάζονται οι περιπτώσεις χρήσης της εφαρμογής, τα σενάρια χρήσης για κάθε λειτουργία του συστήματος που θα βοηθήσουν στην πληρέστερη κατανόησή του. Μετά από την ανάλυση κάθε περίπτωσης χρήσης παραβάλλεται το διάγραμμα ακολουθίας συστήματος το οποίο παρουσιάζει την αλληλεπίδραση του χρήστη με την διεπαφή του συστήματος, όπως επίσης και την ανταπόκριση του δεύτερου στις ενέργειες που δέχεται από τον χρήστη.



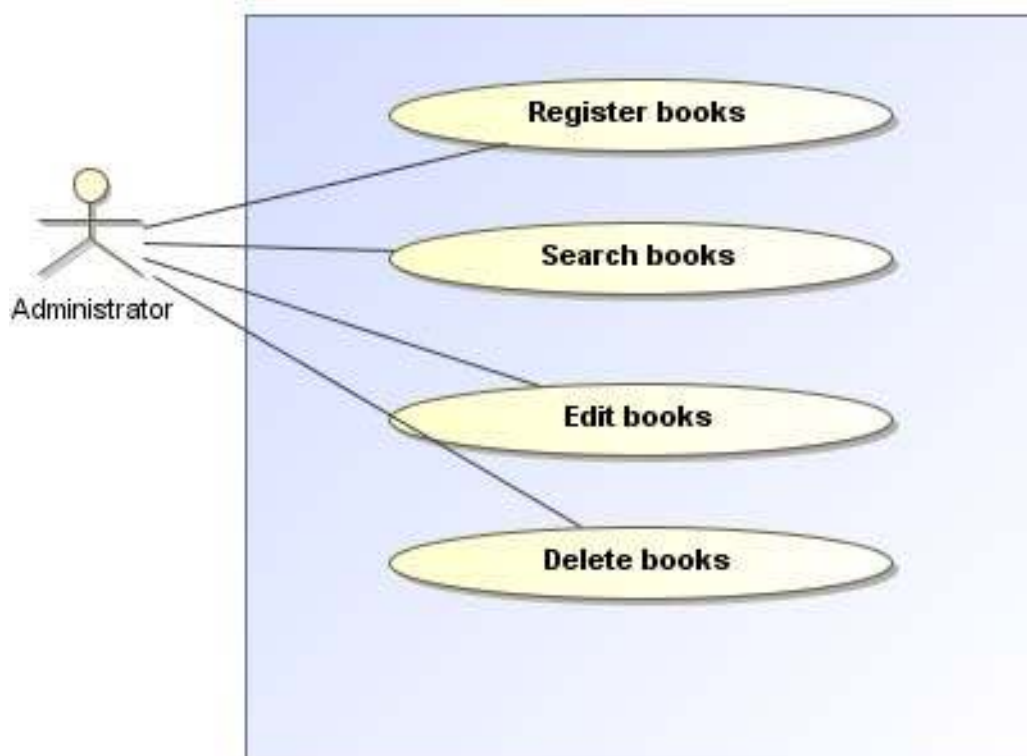
Σχήμα 1: Διάγραμμα περιπτώσεων χρήσης εφαρμογής



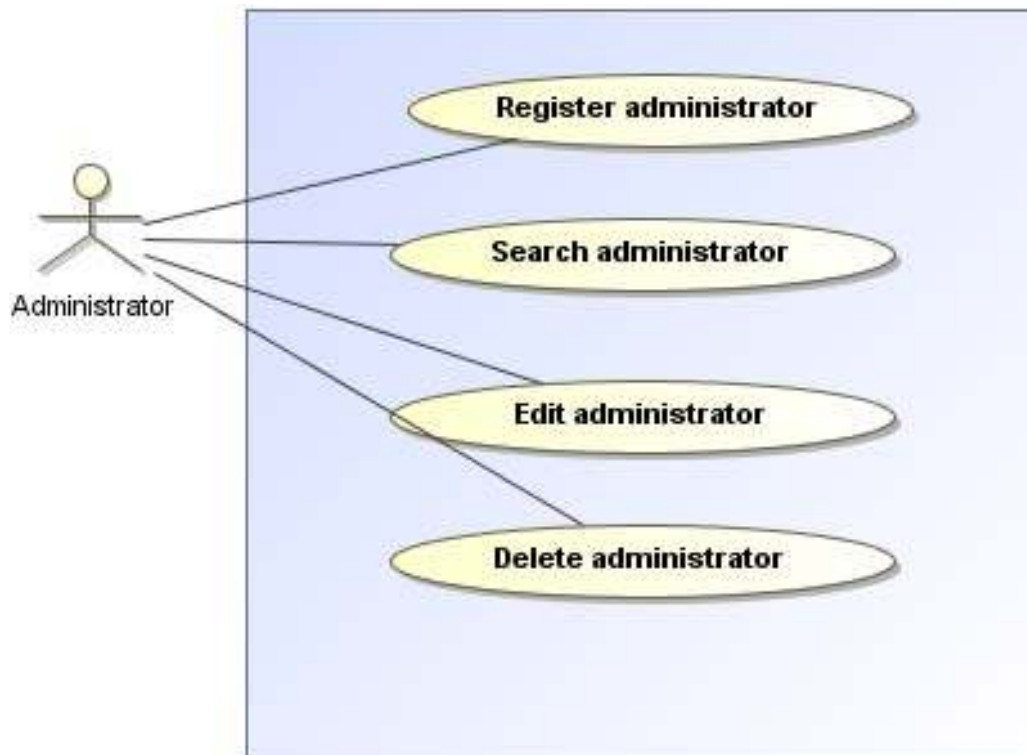
Σχήμα II: Διάγραμμα περίπτωσης χρήσης «Επιλογή συγγραμμάτων»



Σχήμα III: Διάγραμμα περίπτωσης χρήσης «Διαχείρισης μαθημάτων»



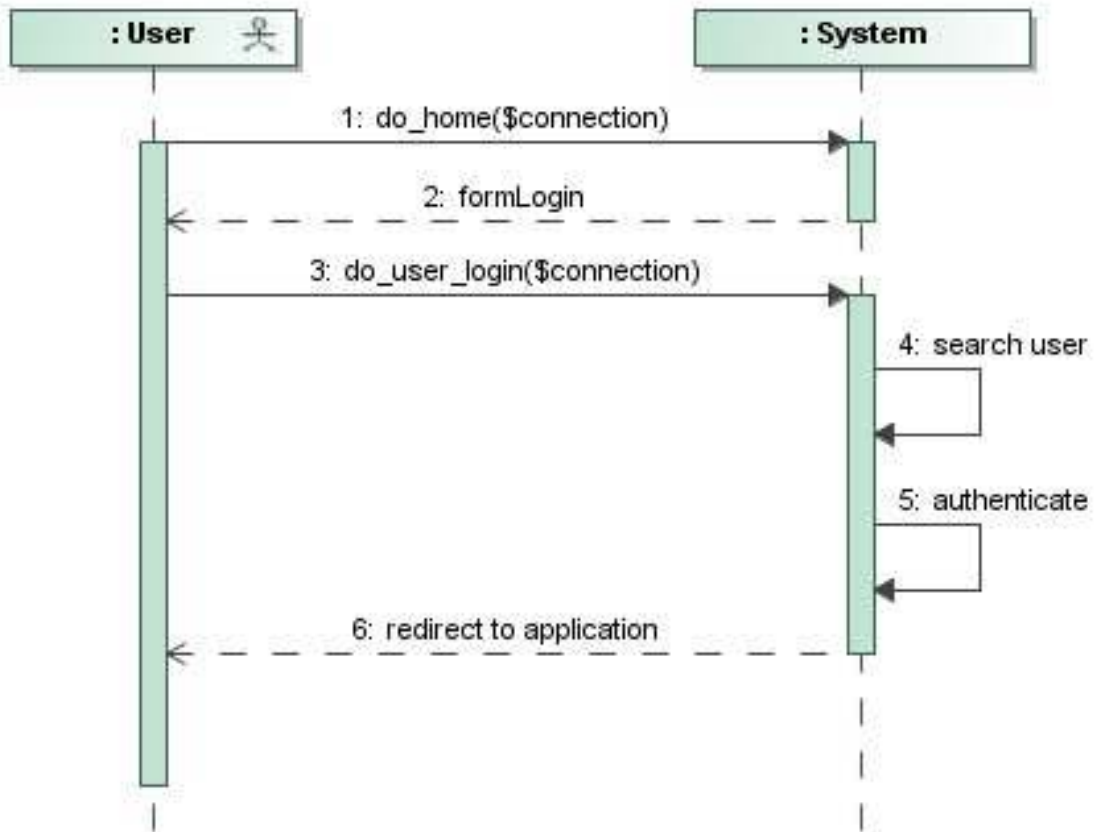
Σχήμα IV: Διάγραμμα περίπτωσης χρήσης «*Διαχείριση συγγραμμάτων*»



Σχήμα V: Διάγραμμα περίπτωσης χρήσης «*Διαχείριση διαχειριστών*»

1.9.1 Είσοδος στο σύστημα

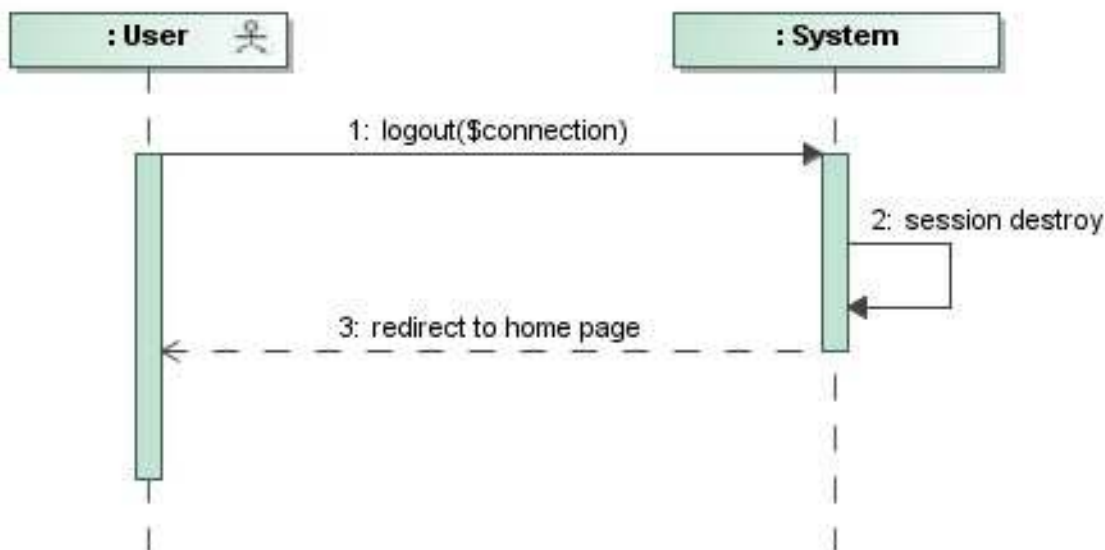
Όνομα σεναρίου χρήσης	Login
Βασικοί ενδιαφερόμενοι	Χρήστης, Διαχειριστής (Αναφέρονται με τον όρο «χρήστης» στην παρακάτω ΠΧ)
Προαπαιτούμενα	Ο χρήστης θα πρέπει να είναι εγγεγραμμένος στην κατάλληλη βάση και να έχει δυνατότητα πρόσβασης μέσω εξατομικευμένου username και password
Αποτέλεσμα επιτυχούς εκτέλεσης	Ο χρήστης συνδέεται στο σύστημα. Έχει προβληθεί η σελίδα εισόδου.
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο χρήστης εισάγει το όνομα χρήστη και τον κωδικό πρόσβασης 2. Το σύστημα ελέγχει αν ο χρήστης είναι καταχωρημένος στην βάση δεδομένων 3. Το σύστημα κατευθύνει το χρήστη στην σελίδα εισόδου
Εναλλακτικά σενάρια	<ol style="list-style-type: none"> 1. Ο χρήστης εισάγει λανθασμένο όνομα χρήστη ή κωδικό πρόσβασης <ol style="list-style-type: none"> a. Το σύστημα ενημερώνει το χρήστη για την λανθασμένη εισαγωγή 2. Ο χρήστης δεν είναι καταχωρημένος στην βάση δεδομένων <ol style="list-style-type: none"> a. Το σύστημα ενημερώνει το χρήστη για την ανάγκη δημιουργίας λογαριασμού.
Μετα-συνθήκες	-



Σχήμα VI: Διάγραμμα ακολουθίας συστήματος «Είσοδος στο σύστημα»

1.9.2 Αποσύνδεση απο το σύστημα

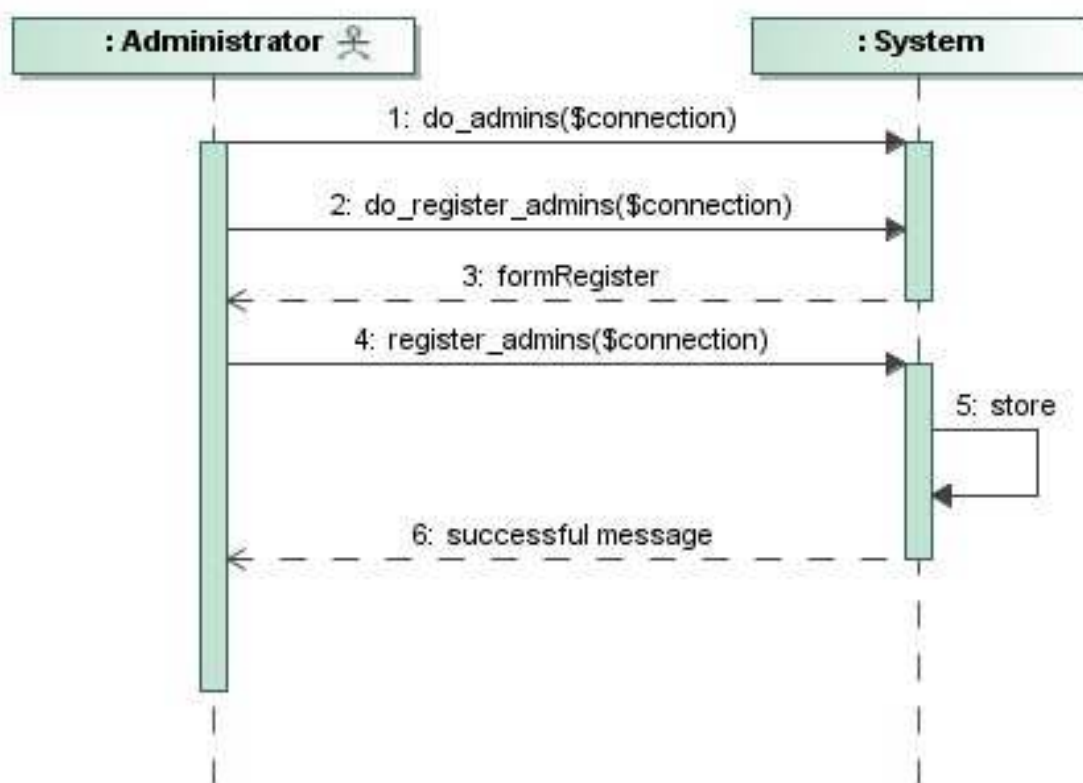
Όνομα σεναρίου χρήσης	Logout
Βασικοί ενδιαφερόμενοι	Χρήστης, Διαχειριστής (Αναφέρονται με τον όρο «χρήστης» στην παρακάτω ΠΧ)
Προαπαιτούμενα	Ο χρήστης είναι συνδεδεμένος στο σύστημα.
Αποτέλεσμα επιτυχούς εκτέλεσης	Ο χρήστης έχει αποσυνδεθεί από το σύστημα και βρίσκεται στην σελίδα εισόδου.
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο χρήστης κάνει κλικ στο κουμπί «αποσύνδεση» 2. Το σύστημα ανακατευθύνει το χρήστη στην σελίδα εισόδου.
Εναλλακτικά σενάρια	-
Μετα-συνθήκες	-



Σχήμα VII: Διάγραμμα ακολουθίας συστήματος «Αποσύνδεση απο το σύστημα»

1.9.3 Εγγραφή διαχειριστή

Όνομα σεναρίου χρήσης	Register admins
Βασικοί ενδιαφερόμενοι	Διαχειριστής
Προαπαιτούμενα	Ο διαχειριστής είναι συνδεδεμένος στο σύστημα
Αποτέλεσμα επιτυχούς εκτέλεσης	Ο διαχειριστής δημιουργεί νέο λογαριασμό και τα στοιχεία του καταχωρούνται στην βάση δεδομένων με επιτυχία.
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο διαχειριστής επιλέγει την εγγραφή νέου διαχειριστή από την κεντρική σελίδα διαχείρισης 2. Το σύστημα εμφανίζει μια φόρμα εισαγωγής στοιχείων 3. Ο διαχειριστής πληκτρολογεί τα στοιχεία 4. Ο διαχειριστής ολοκληρώνει τη διαδικασία 5. Το σύστημα ελέγχει την εγκυρότητα των στοιχείων 6. Το σύστημα καταχωρεί το νέο διαχειριστή στην βάση δεδομένων 7. Το σύστημα εμφανίζει μήνυμα ολοκλήρωσης της διαδικασίας
Εναλλακτικά σενάρια	<ol style="list-style-type: none"> 5. Ο διαχειριστής πληκτρολόγησε λανθασμένα στοιχεία ή δεν πληκτρολόγησε όλα τα υποχρεωτικά πεδία <ol style="list-style-type: none"> a. Το σύστημα ενημερώνει τον διαχειριστή για την λανθασμένη εισαγωγή ή την έλειψη κάποιου πεδίου
Μετα-συνθήκες	-

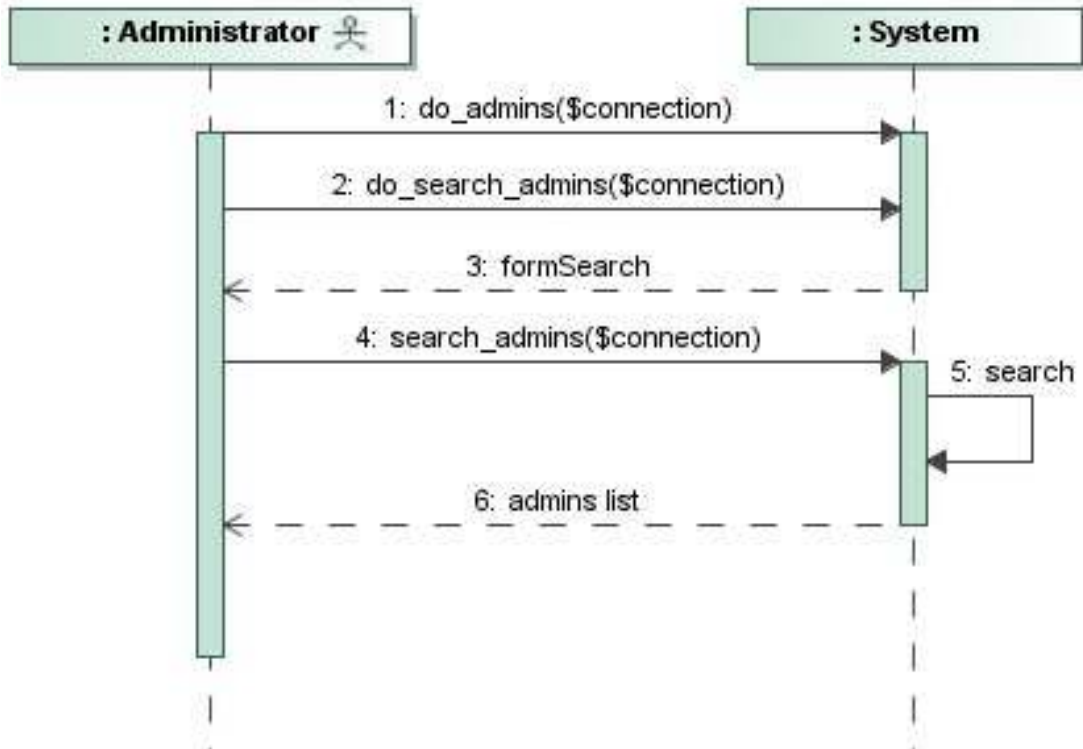


Σχήμα VIII: Διάγραμμα ακολουθίας συστήματος «Εγγραφή διαχειριστή»

1.9.4 Αναζήτηση διαχειριστή

Όνομα σεναρίου χρήσης	Search admins
Βασικοί ενδιαφερόμενοι	Διαχειριστής
Προαπαιτούμενα	Ο διαχειριστής είναι συνδεδεμένος στο σύστημα
Αποτέλεσμα επιτυχούς εκτέλεσης	Το σύστημα εμφανίζει τους προς αναζήτηση διαχειριστές
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο διαχειριστής επιλέγει την αναζήτηση διαχειριστή από την κεντρική σελίδα διαχείρισης 2. Το σύστημα εμφανίζει μια φόρμα εισαγωγής των κριτηρίων αναζήτησης 3. Ο διαχειριστής εισάγει το όνομα, το επίθετο ή το κινητό του διαχειριστή που αναζητά 4. Το σύστημα εκτελεί αναζήτηση και εμφανίζει τους διαθέσιμους διαχειριστές
Εναλλακτικά σενάρια	<ol style="list-style-type: none"> 4. Δεν υπάρχουν διαχειριστές με τα συγκεκριμένα κριτήρια <ol style="list-style-type: none"> a. Το σύστημα ενημερώνει τον διαχειριστή με σχετικό μήνυμα

Μετα-συνθήκες	-
---------------	---

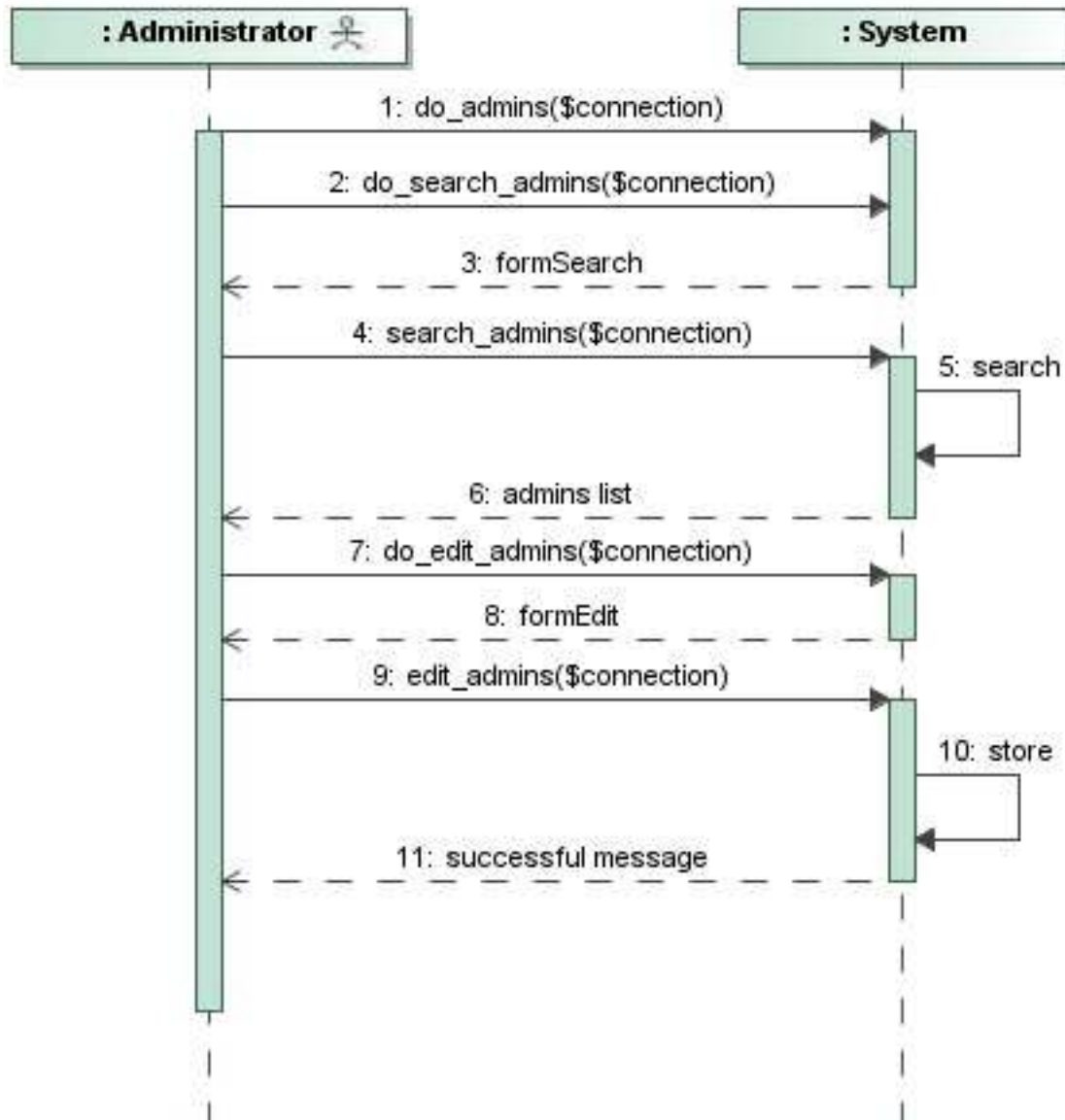


Σχήμα ΙΧ: Διάγραμμα ακολουθίας συστήματος «Αναζήτηση διαχειριστή»

1.9.5 Ενημέρωση διαχειριστή

Όνομα σεναρίου χρήσης	Edit admins
Βασικοί ενδιαφερόμενοι	Διαχειριστής
Προαπαιτούμενα	Ο διαχειριστής είναι συνδεδεμένος στο σύστημα
Αποτέλεσμα επιτυχούς εκτέλεσης	Ο διαχειριστής έχει ενημερωθεί με επιτυχία.
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο διαχειριστής επιλέγει την αναζήτηση διαχειριστή από την κεντρική σελίδα διαχείρισης 2. Το σύστημα εμφανίζει μια φόρμα εισαγωγής των κριτηρίων αναζήτησης 3. Ο διαχειριστής εισάγει το όνομα, το επίθετο ή το κινητό του διαχειριστή που πρόκειται να ενημερωθεί 4. Το σύστημα εκτελεί αναζήτηση και εμφανίζει τους διαθέσιμους διαχειριστές 5. Ο διαχειριστής επιλέγει τον διαχειριστή που θα

	<p>επεξεργαστεί</p> <ol style="list-style-type: none">6. Το σύστημα ανακατευθύνει το διαχειριστή στην σελίδα επεξεργασίας7. Ο διαχειριστής εισάγει το νέο όνομα, επίθετο, όνομα χρήστη, κωδικό πρόσβασης, σταθερό και κινητό τηλέφωνο ή το email8. Ο διαχειριστής ολοκληρώνει την διαδικασία9. Το σύστημα ελέγχει την εγκυρότητα των στοιχείων10. Το σύστημα ενημερώνει τον διαχειριστή
Εναλλακτικά σενάρια	<ol style="list-style-type: none">4. Δεν υπάρχουν διαχειριστές με τα συγκεκριμένα κριτήρια<ol style="list-style-type: none">a. Το σύστημα ενημερώνει το διαχειριστή με σχετικό μήνυμα9. Ο διαχειριστής πληκτρολόγησε λανθασμένα στοιχεία<ol style="list-style-type: none">a. Το σύστημα ενημερώνει το διαχειριστή για την λανθασμένη εισαγωγή
Μετα-συνθήκες	-

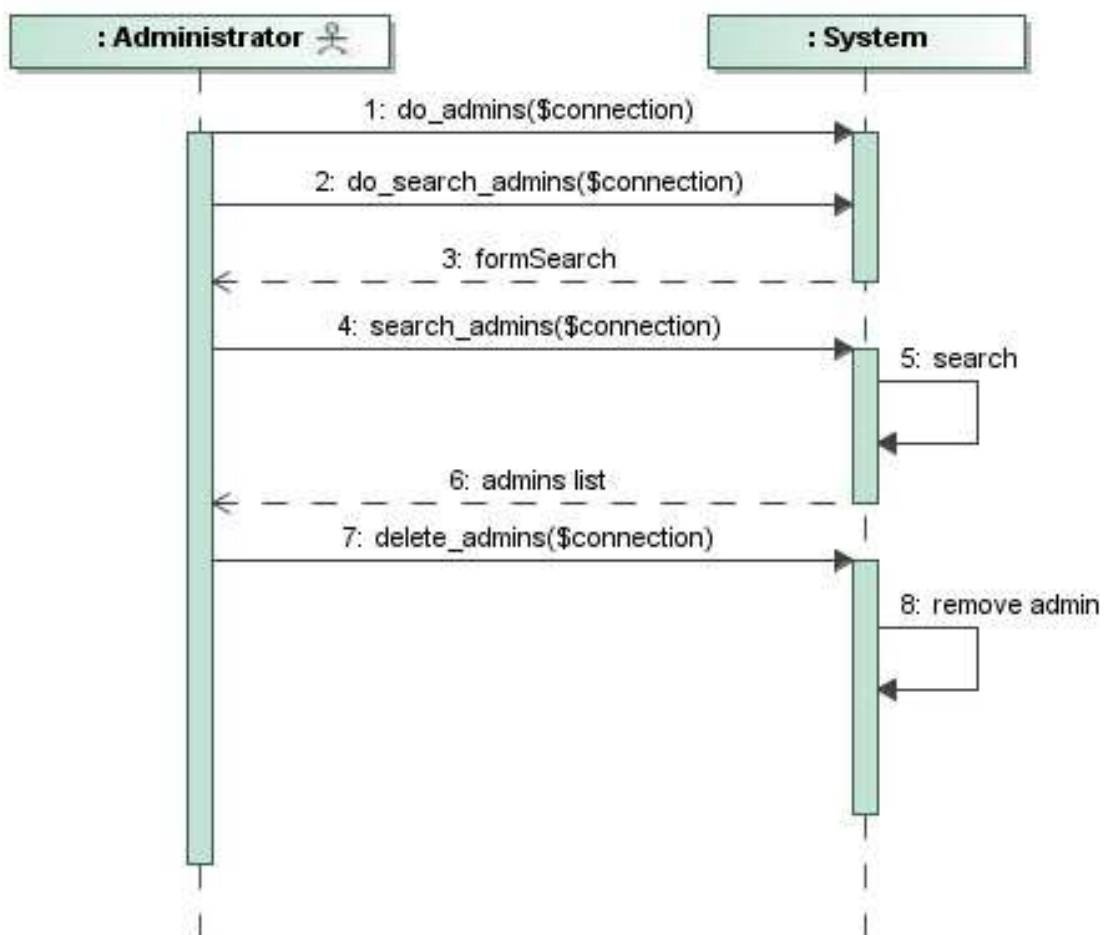


Σχήμα X: Διάγραμμα ακολουθίας συστήματος «Ενημέρωση διαχειριστή»

1.9.6 Διαγραφή διαχειριστή

Όνομα σεναρίου χρήσης	Delete admins
Βασικοί ενδιαφερόμενοι	Διαχειριστής
Προαπαιτούμενα	Ο διαχειριστής είναι συνδεδεμένος στο σύστημα
Αποτέλεσμα επιτυχούς εκτέλεσης	Ο διαχειριστής έχει διαγραφεί με επιτυχία από το σύστημα
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο διαχειριστής επιλέγει την αναζήτηση διαχειριστή από την κεντρική σελίδα διαχείρισης 2. Το σύστημα εμφανίζει μια φόρμα εισαγωγής των κριτηρίων αναζήτησης

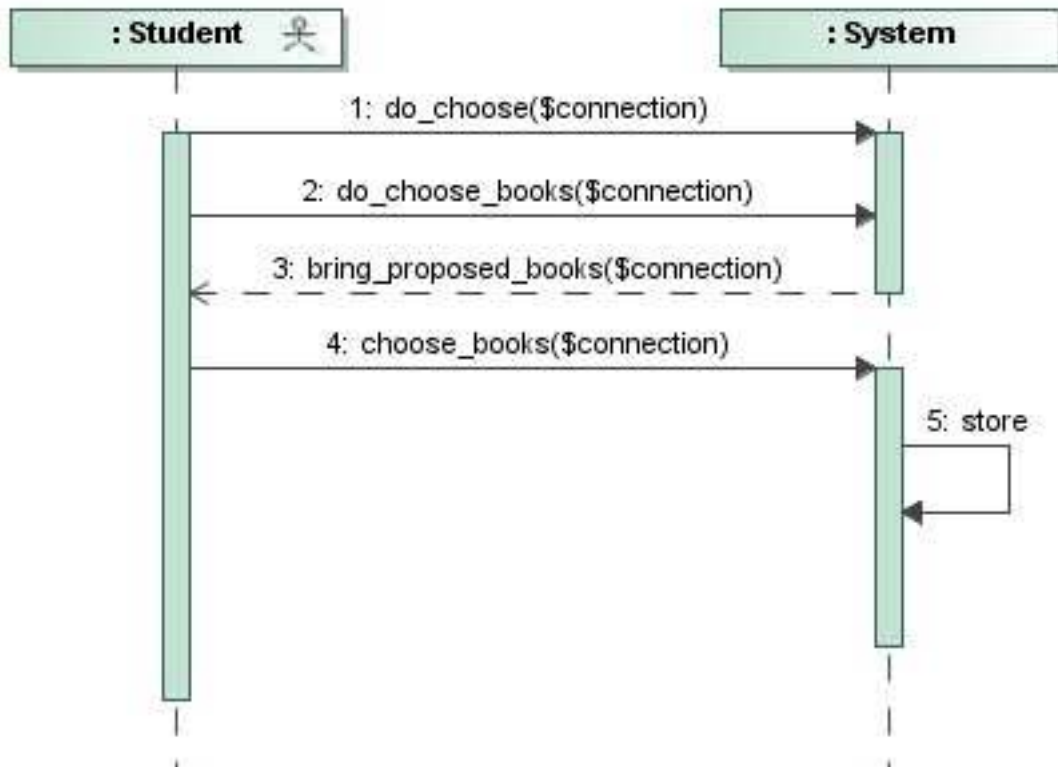
	<ol style="list-style-type: none"> 3. Ο διαχειριστής εισάγει το όνομα, το επίθετο ή το κινητό του διαχειριστή που πρόκειται να διαγράψει 4. Το σύστημα εκτελεί αναζήτηση και εμφανίζει τους διαθέσιμους διαχειριστές 5. Ο διαχειριστής επιλέγει τον διαχειριστή που θα διαγράψει 6. Ο διαχειριστής ολοκληρώνει την διαδικασία 7. Το σύστημα διαγράφει με επιτυχία τον διαχειριστή από τη βάση δεδομένων
Εναλλακτικά σενάρια	<ol style="list-style-type: none"> 4. Δεν υπάρχουν διαχειριστές με τα συγκεκριμένα κριτήρια <ol style="list-style-type: none"> a. Το σύστημα ενημερώνει το διαχειριστή με σχετικό μήνυμα
Μετα-συνθήκες	-



Σχήμα XI: Διάγραμμα ακολουθίας συστήματος «Διαγραφή διαχειριστή»

1.9.7 Επιλογή συγγράμματος

Όνομα σεναρίου χρήσης	Choose books
Βασικοί ενδιαφερόμενοι	Φοιτητής
Προαπαιτούμενα	-Ο φοιτητής είναι συνδεδεμένος στο σύστημα -Η κατάσταση των ηλεκτρονικών δηλώσεων είναι ανοιχτή ή έχει γίνει εκκίνηση νέας διαδικασίας
Αποτέλεσμα επιτυχούς εκτέλεσης	Ο χρήστης επιλέγει τα συγγράμματα για τα μαθήματα που έχει δηλώσει και η ηλεκτρονική του δήλωση αποθηκεύεται στην βάση δεδομένων για μελλοντική μελέτη απο την γραμματεία
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο φοιτητής επιλέγει την επιλογή των βιβλίων απο την κύρια σελίδα των δηλώσεων 2. Το σύστημα εμφανίζει όλα τα μαθήματα και τα συγγράμματα του που αντιστοιχούν σε αυτά 3. Το σύστημα ελέγχει την νομιμότητα της επιλογής του φοιτητή για τα συγκεκριμένα βιβλία 4. Ο φοιτητής επιλέγει το εξάμηνο που τον ενδιαφέρει 5. Ο φοιτητής επιλέγει τα συγγράμματα που επιθυμει για κάθε μάθημα 6. Ο φοιτητής ολοκληρώνει την διαδικασία 7. Το σύστημα καταχωρεί τις προτιμήσεις του φοιτητή στην βάση δεδομένων
Εναλλακτικά σενάρια	<ol style="list-style-type: none"> 1. Η διαδικασία των δηλώσεων έχει κλείσει <ol style="list-style-type: none"> a. Το σύστημα εμφανίζει τα βιβλία που δικαιούται ο φοιτητής 2. Δεν διανέμονται συγγράμματα για το συγκεκριμένο μάθημα <ol style="list-style-type: none"> a. Το σύστημα δεν εμφανίζει διαθέσιμα συγγράμματα 3. Ο φοιτητής έχει πάρει ξανά το ίδιο βιβλίο στο παρελθόν <ol style="list-style-type: none"> a. Το σύστημα κλειδώνει το βιβλίο b. Ο φοιτητής επιλέγει άλλο βιβλίο απο την λίστα
Μετα-συνθήκες	Μετά την επιτυχή πραγματοποίηση της επιλογής συγγραμμάτων και αφού η διαδικασία επιλογής έχει κλείσει, ο χρήστης δεν θα μπορεί να επιλέξει ξανά το ίδιο σύγγραμμα στο μέλλον.

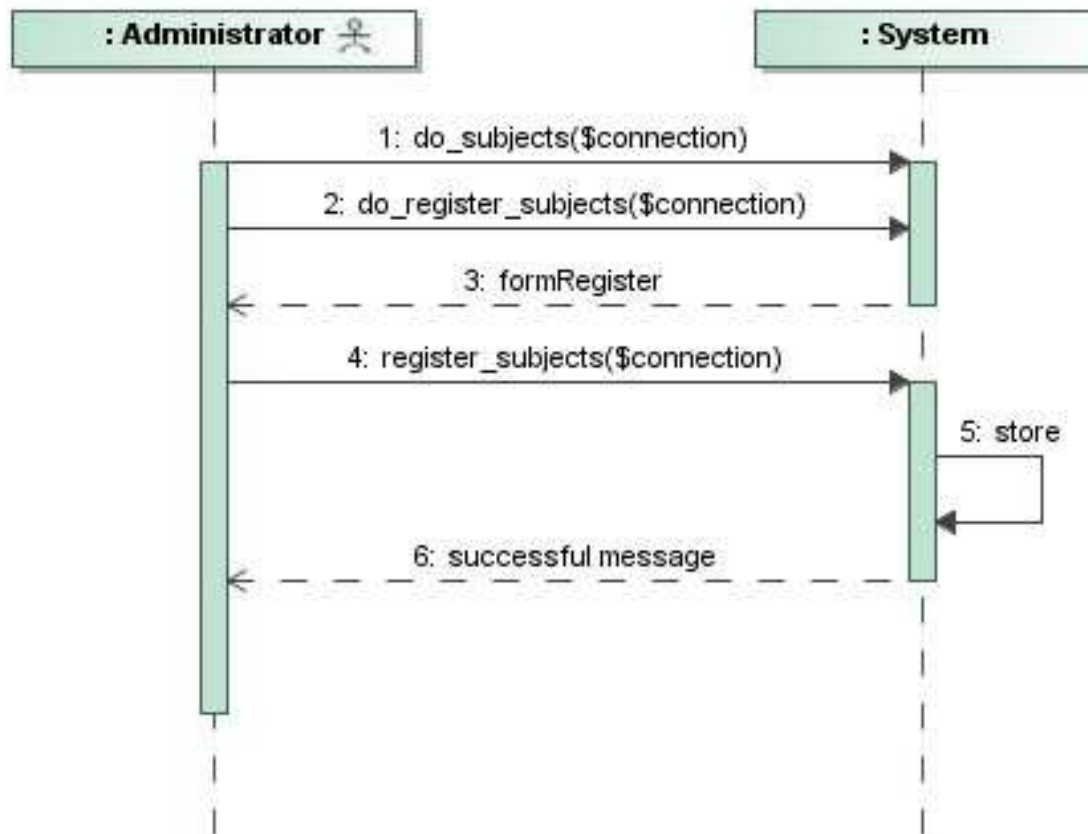


Σχήμα XII: Διάγραμμα ακολουθίας συστήματος «Επιλογή συγγραμμάτων»

1.9.8 Διαχείριση μαθημάτων - Εισαγωγή

Όνομα σεναρίου χρήσης	Register subjects
Βασικοί ενδιαφερόμενοι	Διαχειριστής
Προαπαιτούμενα	Ο διαχειριστής είναι συνδεδεμένος στο σύστημα.
Αποτέλεσμα επιτυχούς εκτέλεσης	Το νέο μάθημα έχει καταχωρηθεί στο σύστημα με επιτυχία.
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο διαχειριστής επιλέγει την εισαγωγή μαθήματος από την κεντρική σελίδα διαχείρισης. 2. Το σύστημα εμφανίζει μια φόρμα εισαγωγής στοιχείων 3. Ο διαχειριστής εισάγει τα στοιχεία του μαθήματος. 4. Ο διαχειριστής ολοκληρώνει την διαδικασία εισαγωγής 5. Το σύστημα ελέγχει την εγκυρότητα των στοιχείων 6. Το σύστημα καταχωρεί το νέο μάθημα. 7. Το σύστημα εμφανίζει μήνυμα
Εναλλακτικά σενάρια	<ol style="list-style-type: none"> 5. Κάποιο από τα πεδία δεν είναι σωστά ή είναι ελειπές. <ol style="list-style-type: none"> a. Το σύστημα ενημερώνει τον χρήστη για την αποτυχημένη εισαγωγή.

Μετα-συνθήκες	-
---------------	---

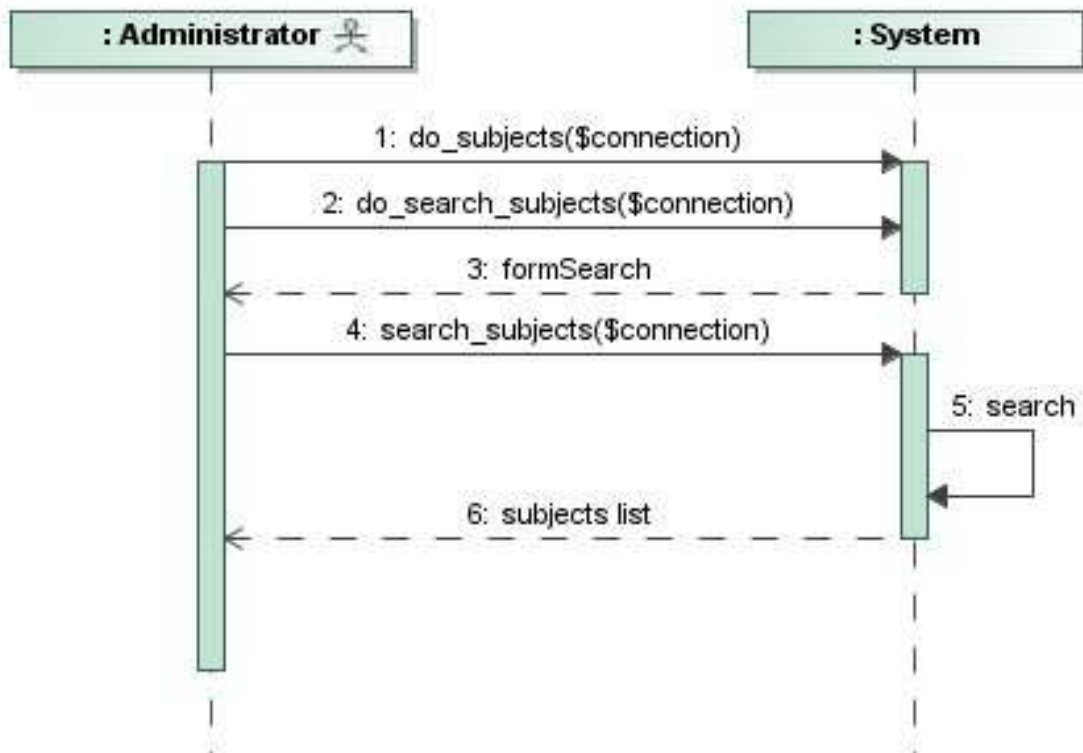


Σχήμα XIII: Διάγραμμα ακολουθίας συστήματος «Εισαγωγή μαθήματος»

1.9.9 Διαχείριση μαθημάτων - Αναζήτηση

Όνομα σεναρίου χρήσης	Search subjects
Βασικοί ενδιαφερόμενοι	Διαχειριστής
Προαπαιτούμενα	Ο διαχειριστής είναι συνδεδεμένος στο σύστημα.
Αποτέλεσμα επιτυχούς εκτέλεσης	Το σύστημα εμφανίζει τα διαθέσιμα μαθήματα
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο διαχειριστής επιλέγει την αναζήτηση μαθήματος απο την κεντρική σελίδα διαχείρισης 2. Το σύστημα εμφανίζει μια φόρμα εισαγωγής των κριτηρίων αναζήτησης 3. Ο διαχειριστής εισάγει τα κριτήρια αναζήτησης του μαθήματος που αναζητά 4. Το σύστημα εκτελεί αναζήτηση και εμφανίζει τα διαθέσιμα μαθήματα.

Εναλλακτικά σενάρια	4. Δεν υπάρχουν μαθήματα με τα κριτήρια που εισήγαγε ο διαχειριστής a. Το σύστημα εμφανίζει στον διαχειριστή σχετικό μήνυμα.
Μετα-συνθήκες	-

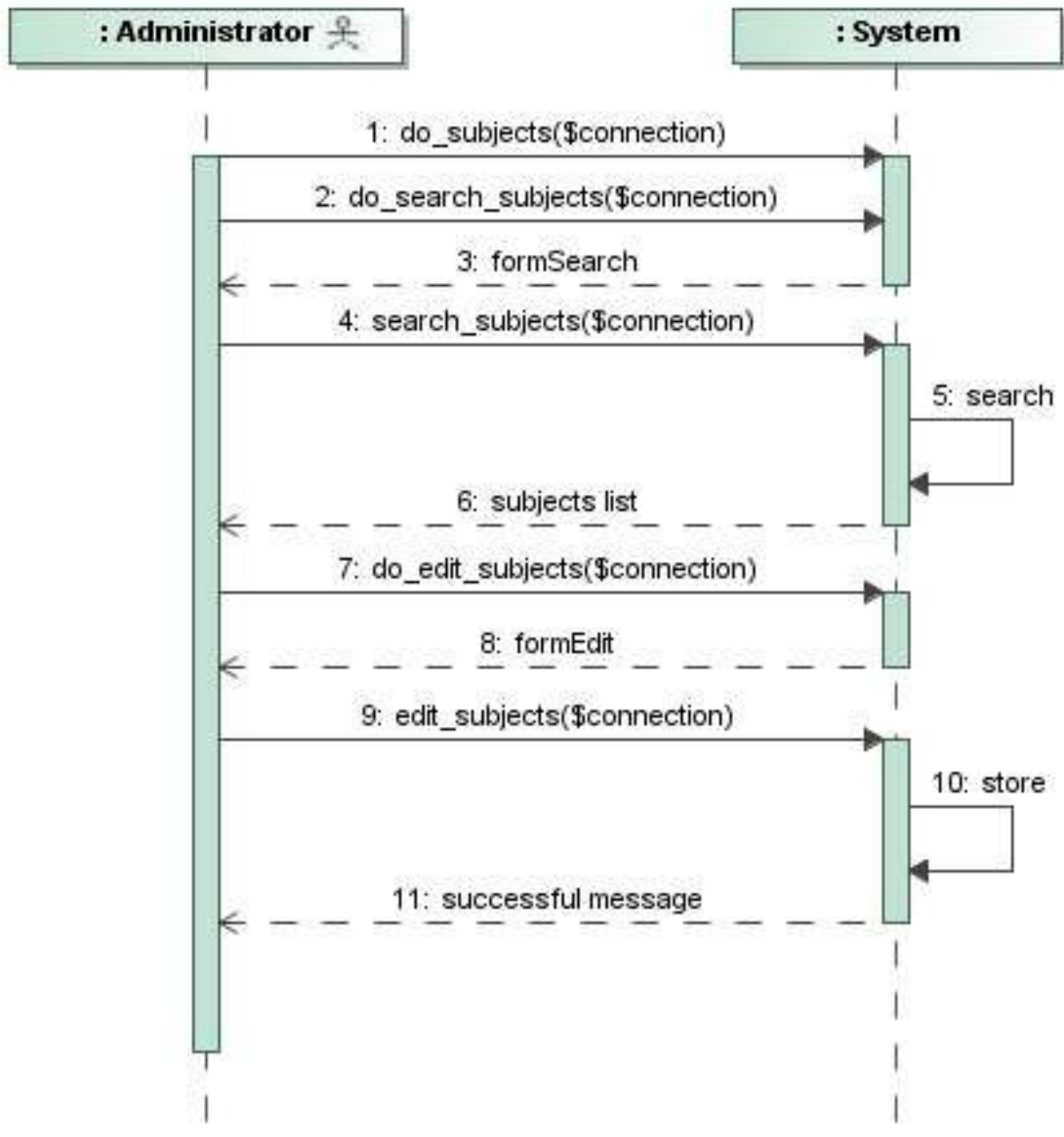


Σχήμα XIV: Διάγραμμα ακολουθίας συστήματος «Αναζήτηση μαθήματος»

1.9.10 Διαχείριση μαθημάτων - Ενημέρωση

Όνομα σεναρίου χρήσης	Edit subjects
Βασικοί ενδιαφερόμενοι	Διαχειριστής
Προαπαιτούμενα	Ο διαχειριστής είναι συνδεδεμένος στο σύστημα
Αποτέλεσμα επιτυχούς εκτέλεσης	Το μάθημα έχει ενημερωθεί με επιτυχία.
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο διαχειριστής επιλέγει την αναζήτηση μαθήματος από την κεντρική σελίδα διαχείρισης 2. Το σύστημα εμφανίζει μια φόρμα εισαγωγής των κριτηρίων αναζήτησης 3. Ο διαχειριστής εισάγει τα κριτήρια αναζήτησης του μαθήματος που πρόκειται να ενημερωθεί

	<ol style="list-style-type: none"> 4. Το σύστημα εκτελεί αναζήτηση και εμφανίζει τα διαθέσιμα μαθήματα. 5. Ο διαχειριστής επιλέγει το μάθημα που θα επεξεργαστεί 6. Το σύστημα ανακατευθύνει το διαχειριστή στην σελίδα επεξεργασίας 7. Ο διαχειριστής εισάγει τα νέα στοιχεία για το μάθημα 8. Ο διαχειριστής ολοκληρώνει την διαδικασία 9. Το σύστημα ελέγχει την εγκυρότητα των στοιχείων 10. Το σύστημα ενημερώνει τον διαχειριστή
Εναλλακτικά σενάρια	<ol style="list-style-type: none"> 4. Δεν υπάρχουν μαθήματα με τα συγκεκριμένα κριτήρια <ol style="list-style-type: none"> a. Το σύστημα ενημερώνει το διαχειριστή με σχετικό μήνυμα 9. Ο διαχειριστής πληκτρολόγησε λανθασμένα στοιχεία <ol style="list-style-type: none"> a. Το σύστημα ενημερώνει το διαχειριστή για την λανθασμένη εισαγωγή
Μετα-συνθήκες	-

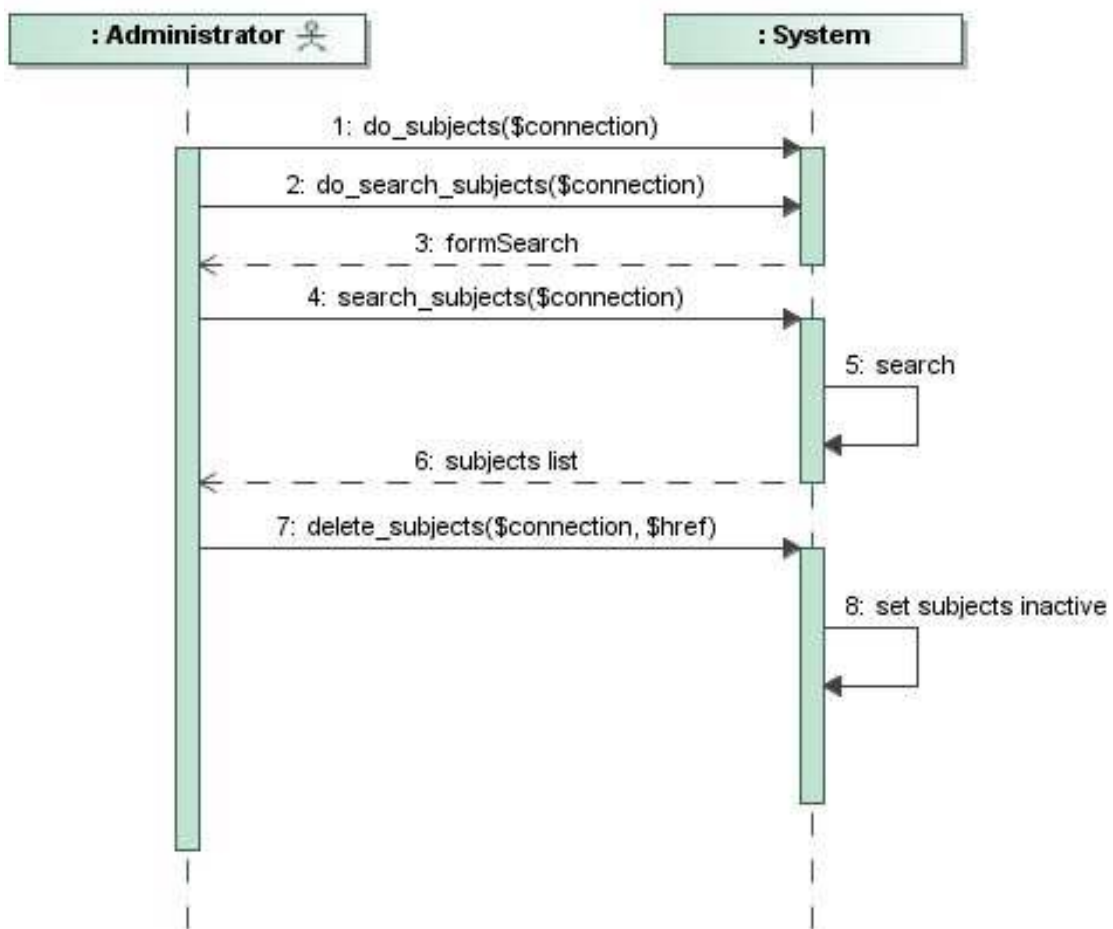


Σχήμα XV: Διάγραμμα ακολουθίας συστήματος «Ενημέρωση μαθήματος»

1.9.11 Διαχείριση μαθημάτων - Διαγραφή

Όνομα σεναρίου χρήσης	Delete subjects
Βασικοί ενδιαφερόμενοι	Διαχειριστής
Προαπαιτούμενα	Ο διαχειριστής είναι συνδεδεμένος στο σύστημα.
Αποτέλεσμα επιτυχούς εκτέλεσης	Το μάθημα έχει απενεργοποιηθεί με επιτυχία
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο διαχειριστής επιλέγει την αναζήτηση μαθήματος απο την κεντρική σελίδα διαχείρισης 2. Το σύστημα εμφανίζει μια φόρμα εισαγωγής των κριτηρίων αναζήτησης

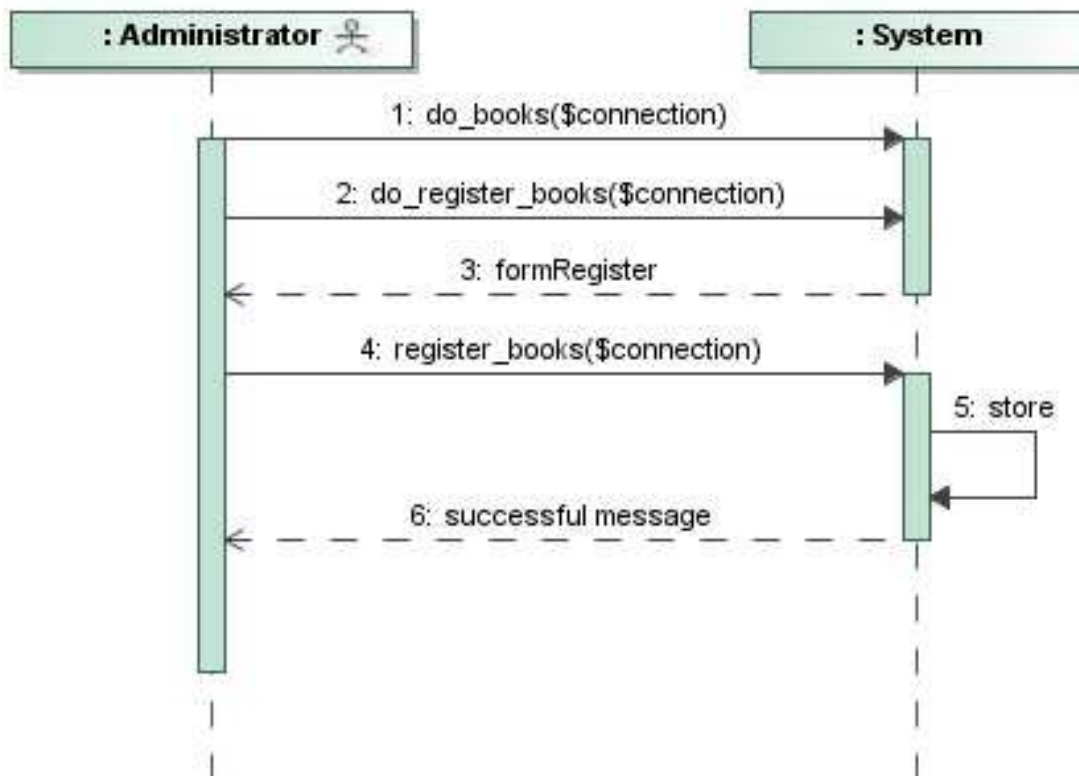
	<ol style="list-style-type: none"> 3. Ο διαχειριστής εισάγει τα κριτήρια αναζήτησης του μαθήματος που πρόκειται να απενεργοποιηθεί 4. Το σύστημα εκτελεί αναζήτηση και εμφανίζει τα διαθέσιμα μαθήματα. 5. Ο διαχειριστής επιλέγει το μάθημα που θα απενεργοποιήσει 6. Ο διαχειριστής ολοκληρώνει την διαδικασία 7. Το σύστημα απενεργοποιεί το μάθημα με επιτυχία
Εναλλακτικά σενάρια	<ol style="list-style-type: none"> 4. Δεν υπάρχουν μαθήματα με τα κριτήρια που εισήγαγε ο διαχειριστής <ol style="list-style-type: none"> a. Το σύστημα εμφανίζει στον διαχειριστή σχετικό μήνυμα. 7 Το μάθημα είναι ήδη απενεργοποιημένο <ol style="list-style-type: none"> a. Το σύστημα ενεργοποιεί το επιλεγμένο μάθημα
Μετα-συνθήκες	-



Σχήμα XVI: Διάγραμμα ακολουθίας συστήματος «Διαγραφή μαθήματος»

1.9.12 Διαχείριση συγγραμμάτων – Εισαγωγή

Όνομα σεναρίου χρήσης	Register books
Βασικοί ενδιαφερόμενοι	Διαχειριστής
Προαπαιτούμενα	Ο διαχειριστής είναι συνδεδεμένος στο σύστημα.
Αποτέλεσμα επιτυχούς εκτέλεσης	Το νέο σύγγραμμα έχει καταχωρηθεί στο σύστημα με επιτυχία.
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο διαχειριστής επιλέγει την καταχώρηση βιβλίου απο την κεντρική σελίδα διαχείρισης. 2. Το σύστημα εμφανίζει μια φόρμα εισαγωγής στοιχείων 3. Ο διαχειριστής εισάγει τα στοιχεία του βιβλίου. 4. Ο διαχειριστής ολοκληρώνει την διαδικασία εισαγωγής 5. Το σύστημα ελέγχει την εγκυρότητα των στοιχείων 6. Το σύστημα καταχωρεί το νέο βιβλίο. 7. Το σύστημα εμφανίζει μήνυμα ολοκλήρωσης της διαδικασίας εισαγωγής
Εναλλακτικά σενάρια	<ol style="list-style-type: none"> 5. Κάποιο απο τα πεδία δεν είναι σωστά ή είναι ελειπή. <ol style="list-style-type: none"> a. Το σύστημα ενημερώνει τον χρήστη για την αποτυχημένη εισαγωγή.
Μετα-συνθήκες	-

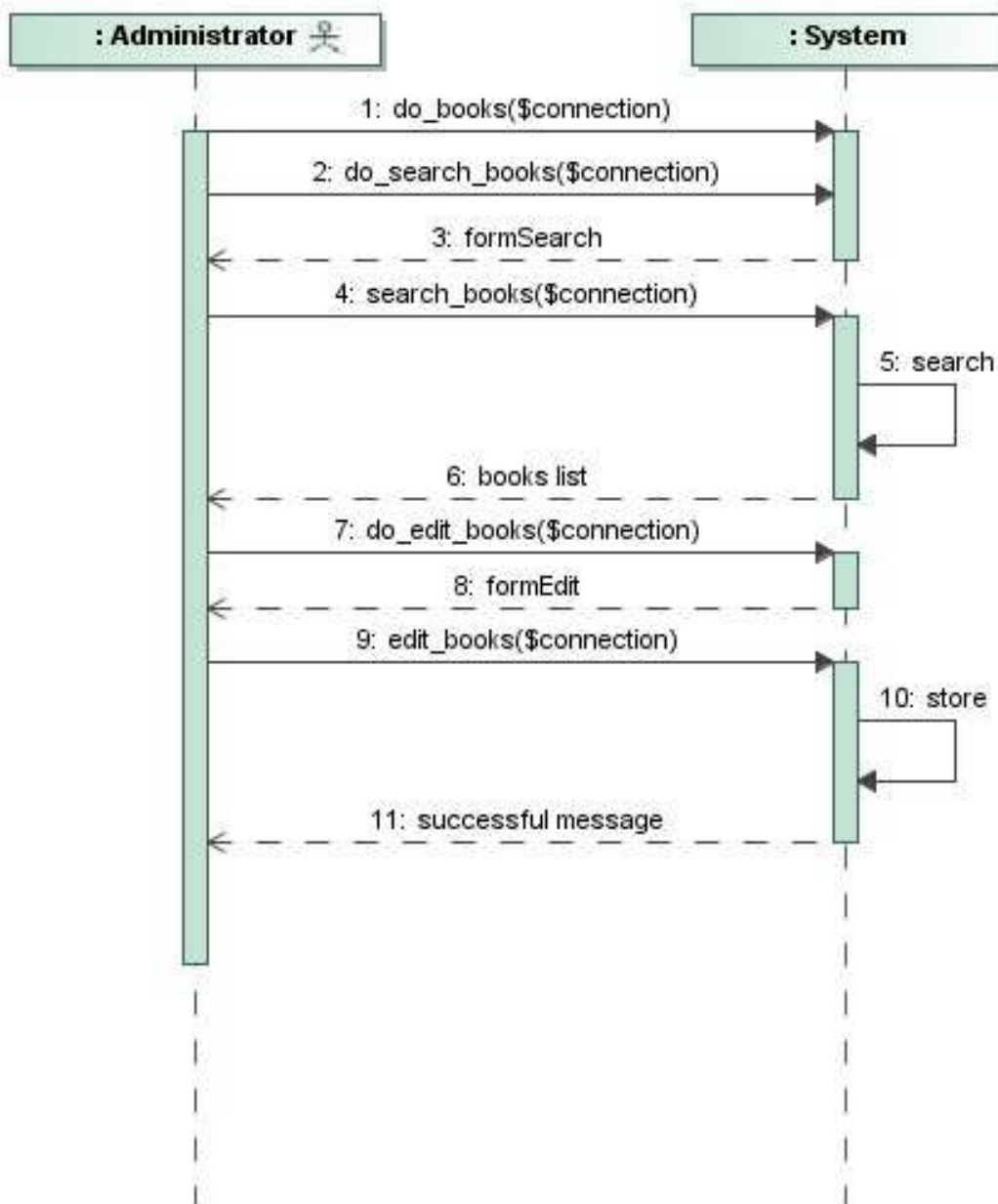


Σχήμα XVII: Διάγραμμα ακολουθίας συστήματος «Εισαγωγή βιβλίου»

1.9.13 Διαχείριση συγγραμμάτων – Ενημέρωση

Όνομα σεναρίου χρήσης	Edit books
Βασικοί ενδιαφερόμενοι	Διαχειριστής
Προαπαιτούμενα	Ο διαχειριστής είναι συνδεδεμένος στο σύστημα
Αποτέλεσμα επιτυχούς εκτέλεσης	Το βιβλίο έχει ενημερωθεί με επιτυχία.
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο διαχειριστής επιλέγει την αναζήτηση βιβλίου από την κεντρική σελίδα διαχείρισης 2. Το σύστημα εμφανίζει μια φόρμα εισαγωγής των κριτηρίων αναζήτησης 3. Ο διαχειριστής εισάγει τα κριτήρια αναζήτησης του βιβλίου που πρόκειται να ενημερωθεί 4. Το σύστημα εκτελεί αναζήτηση και εμφανίζει τα διαθέσιμα βιβλία. 5. Ο διαχειριστής επιλέγει το βιβλίο που θα επεξεργαστεί 6. Το σύστημα ανακατευθύνει το διαχειριστή στην σελίδα επεξεργασίας 7. Ο διαχειριστής εισάγει τα νέα στοιχεία για το βιβλίο 8. Ο διαχειριστής ολοκληρώνει την διαδικασία

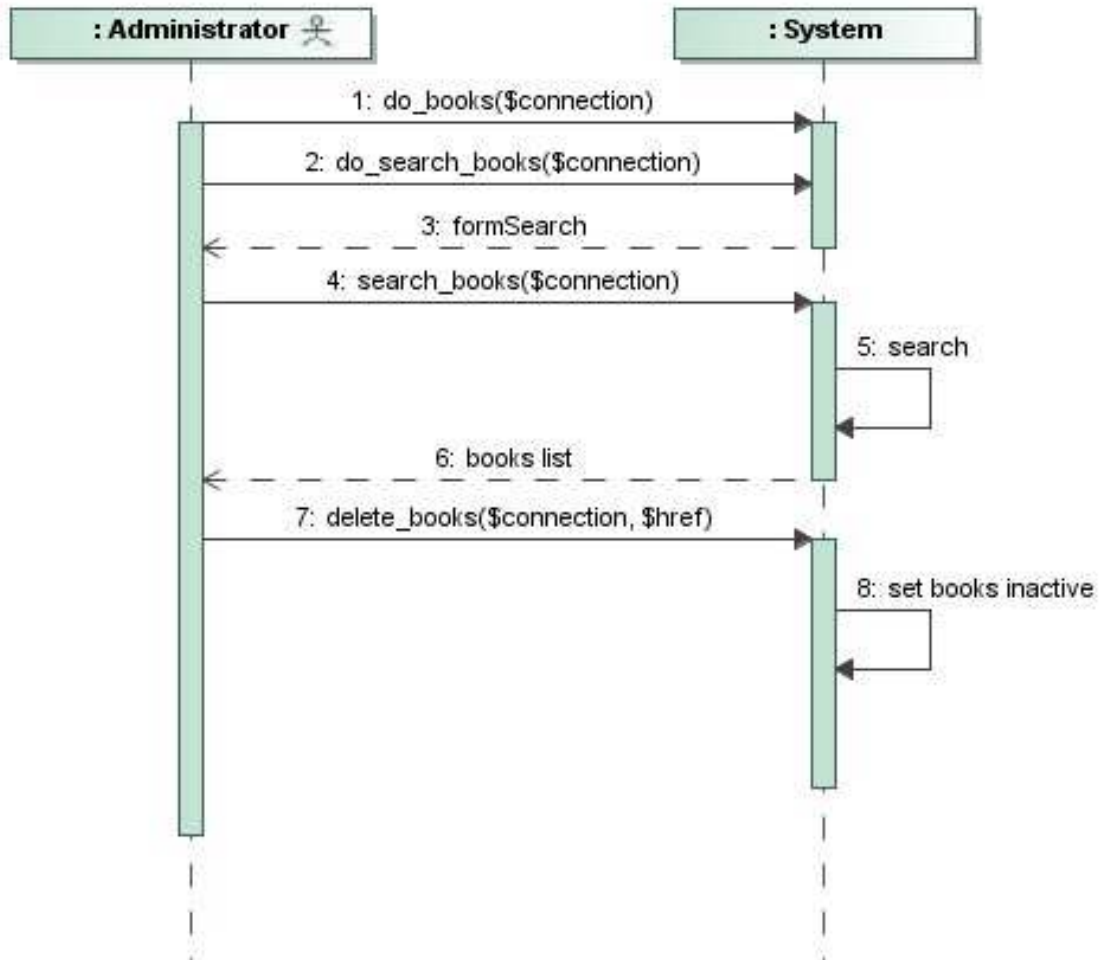
	9. Το σύστημα ελέγχει την εγκυρότητα των στοιχείων 10. Το σύστημα ενημερώνει τον διαχειριστή
Εναλλακτικά σενάρια	4. Δεν υπάρχουν βιβλία με τα συγκεκριμένα κριτήρια a. Το σύστημα ενημερώνει το διαχειριστή με σχετικό μήνυμα 9. Ο διαχειριστής πληκτρολόγησε λανθασμένα στοιχεία a. Το σύστημα ενημερώνει το διαχειριστή για την λανθασμένη εισαγωγή
Μετα-συνθήκες	-



Σχήμα XVIII: Διάγραμμα ακολουθίας συστήματος «Ενημέρωση βιβλίου»

1.9.14 Διαχείριση συγγραμμάτων – Διαγραφή

Όνομα σεναρίου χρήσης	Delete books
Βασικοί ενδιαφερόμενοι	Διαχειριστής
Προαπαιτούμενα	Ο διαχειριστής είναι συνδεδεμένος στο σύστημα.
Αποτέλεσμα επιτυχούς εκτέλεσης	Το βιβλίο έχει απενεργοποιηθεί με επιτυχία
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο διαχειριστής επιλέγει την αναζήτηση βιβλίου απο την κεντρική σελίδα διαχείρισης 2. Το σύστημα εμφανίζει μια φόρμα εισαγωγής των κριτηρίων αναζήτησης 3. Ο διαχειριστής εισάγει τα κριτήρια αναζήτησης του βιβλίου που πρόκειται να απενεργοποιηθεί 4. Το σύστημα εκτελεί αναζήτηση και εμφανίζει τα διαθέσιμα βιβλία. 5. Ο διαχειριστής επιλέγει το βιβλίο που θα απενεργοποιήσει 6. Ο διαχειριστής ολοκληρώνει την διαδικασία 7. Το σύστημα απενεργοποιεί το επιλεγμένο βιβλίο
Εναλλακτικά σενάρια	<ol style="list-style-type: none"> 4. Δεν υπάρχουν βιβλία με τα κριτήρια που εισήγαγε ο διαχειριστής <ol style="list-style-type: none"> a. Το σύστημα εμφανίζει στον διαχειριστή σχετικό μήνυμα. 7. Το βιβλίο είναι ήδη απενεργοποιημένο <ol style="list-style-type: none"> a. Το σύστημα ενεργοποιεί το βιβλίο
Μετα-συνθήκες	Το σύγγραμμα που απενεργοποιήθηκε δεν θα είναι ορατό στους φοιτητές κατά την περίοδο των ηλεκτρονικών δηλώσεων.

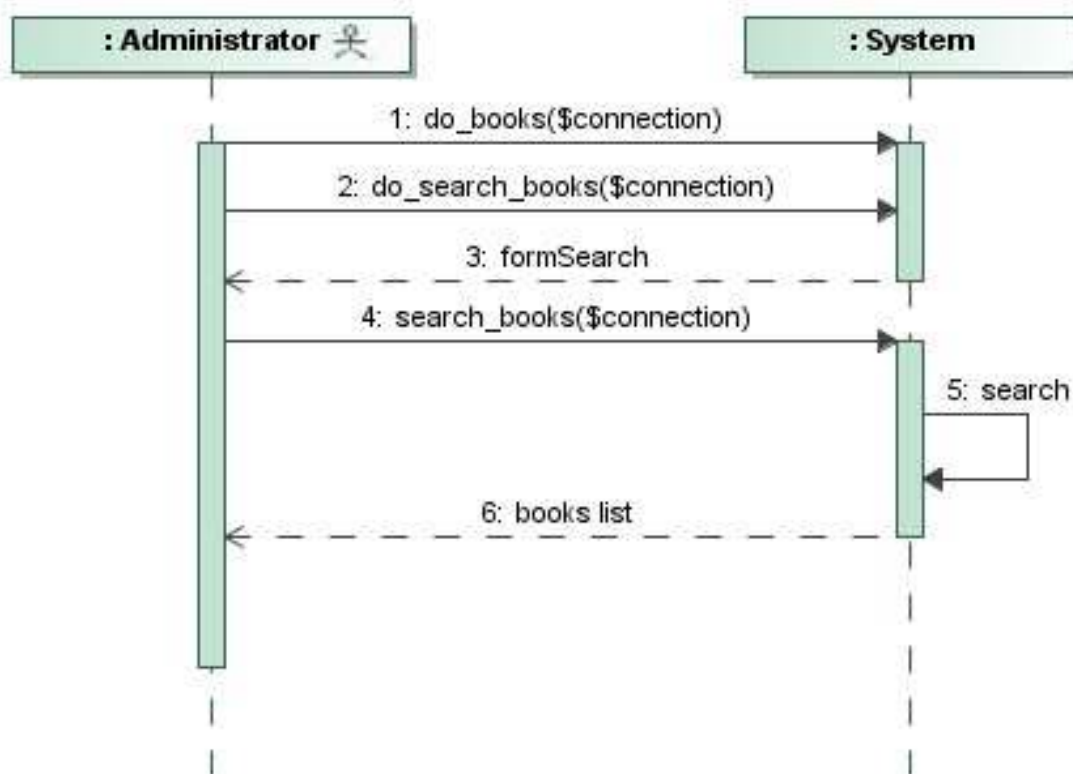


Σχήμα XIX: Διάγραμμα ακολουθίας συστήματος «Διαγραφή βιβλίου»

1.9.15 Διαχείριση συγγραμμάτων - Αναζήτηση

Όνομα σεναρίου χρήσης	Search books
Βασικοί ενδιαφερόμενοι	Διαχειριστής
Προαπαιτούμενα	Ο διαχειριστής είναι συνδεδεμένος στο σύστημα.
Αποτέλεσμα επιτυχούς εκτέλεσης	Το σύστημα εμφανίζει τα διαθέσιμα συγγράμματα
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο διαχειριστής επιλέγει την αναζήτηση βιβλίου από την κεντρική σελίδα διαχείρισης 2. Το σύστημα εμφανίζει μια φόρμα εισαγωγής των κριτηρίων αναζήτησης 3. Ο διαχειριστής εισάγει τα κριτήρια αναζήτησης του βιβλίου που αναζητά 4. Το σύστημα εκτελεί αναζήτηση και εμφανίζει τα διαθέσιμα βιβλία.

Εναλλακτικά σενάρια	4. Δεν υπάρχουν βιβλία με τα κριτήρια που εισήγαγε ο διαχειριστής a. Το σύστημα εμφανίζει στον διαχειριστή σχετικό μήνυμα.
Μετα-συνθήκες	-

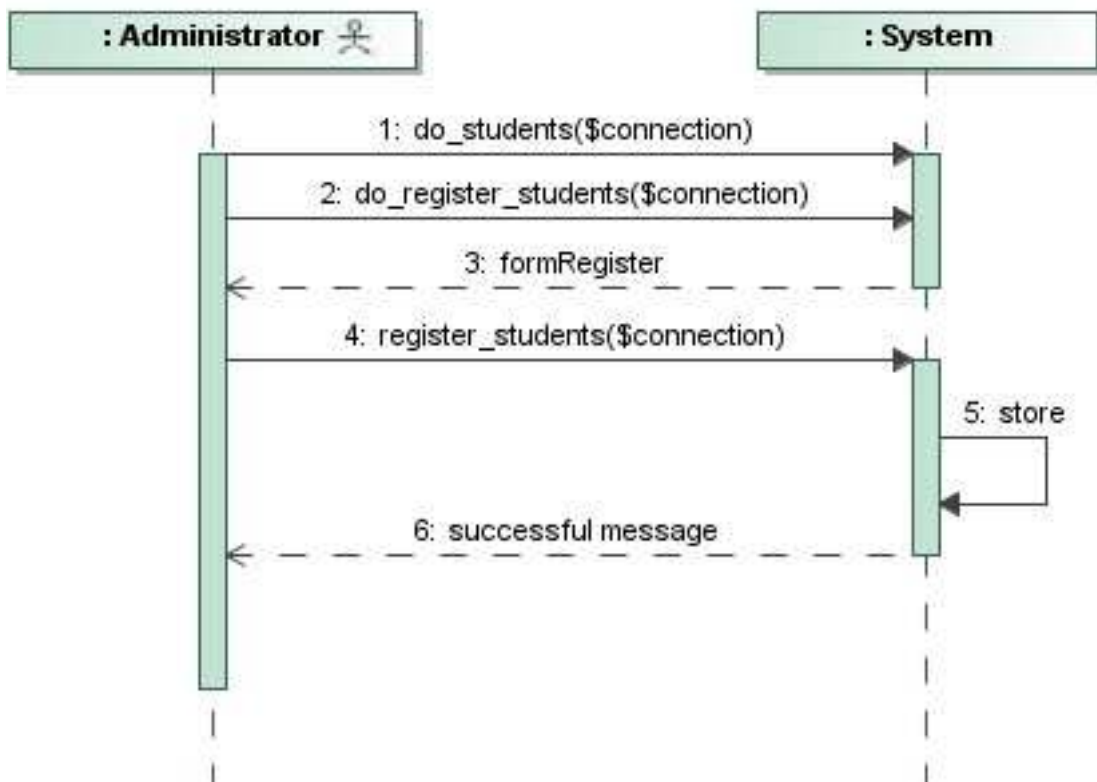


Σχήμα XX: Διάγραμμα ακολουθίας συστήματος «Αναζήτηση βιβλίου»

1.9.16 Καταχώρηση φοιτητών

Όνομα σεναρίου χρήσης	Register student
Βασικοί ενδιαφερόμενοι	Διαχειριστής
Προαπαιτούμενα	Ο διαχειριστής είναι συνδεδεμένος στο σύστημα.
Αποτέλεσμα επιτυχούς εκτέλεσης	Τα στοιχεία του νέου φοιτητή έχουν καταχωρηθεί στην βάση δεδομένων με επιτυχία.
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο διαχειριστής επιλέγει την καταχώρηση φοιτητή από την κεντρική σελίδα διαχείρισης 2. Το σύστημα εμφανίζει μια φόρμα εισαγωγής των στοιχείων του φοιτητή 3. Ο διαχειριστής εισάγει τα απαραίτητα στοιχεία

	<ol style="list-style-type: none"> 4. Ο διαχειριστής ολοκληρώνει την διαδικασία 5. Το σύστημα ελέγχει την εγκυρότητα των δεδομένων που εισήγαγε ο διαχειριστής 6. Το σύστημα καταχωρεί το νέο φοιτητή 7. Το σύστημα ενημερώνει τον διαχειριστή με κατάλληλο μήνυμα
Εναλλακτικά σενάρια	<ol style="list-style-type: none"> 5. Ο διαχειριστής δεν έχει συμπληρώσει όλα τα υποχρεωτικά πεδία ή έχει κάποιο λάθος <ol style="list-style-type: none"> a. Το σύστημα εμφανίζει στον διαχειριστή σχετικό μήνυμα.
Μετα-συνθήκες	-



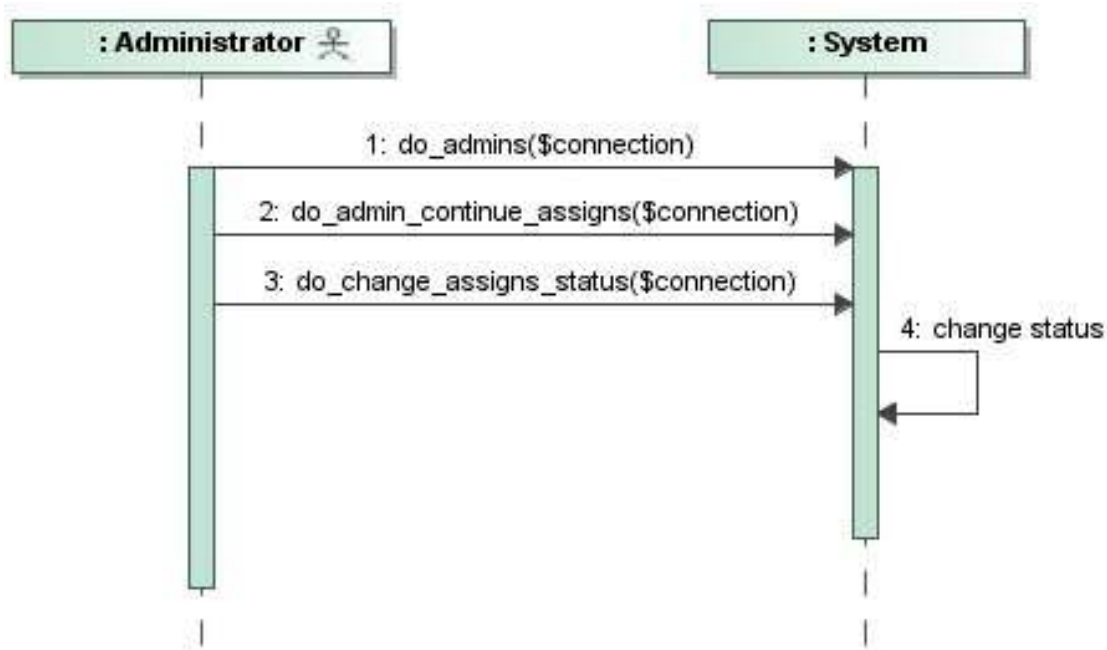
Σχήμα XXI: Διάγραμμα ακολουθίας συστήματος «Καταχώρηση φοιτητών»

1.9.17 Άνοιγμα της διαδικασίας δηλώσεων

Όνομα σεναρίου χρήσης	Change assigns status
Βασικοί ενδιαφερόμενοι	Διαχειριστής
Προαπαιτούμενα	Ο διαχειριστής είναι συνδεδεμένος στο σύστημα. Η διαδικασία των δηλώσεων είναι κλειστή.
Αποτέλεσμα επιτυχούς εκτέλεσης	Η διαδικασία των ηλεκτρονικών δηλώσεων έχει ανοίξει. Οι φοιτητές μπορούν να ξεκινήσουν την διαδικασία δηλώσεων.
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο διαχειριστής επιλέγει το άνοιγμα της διαδικασίας δηλώσεων από την κεντρική σελίδα διαχείρισης 2. Το σύστημα αλλάζει την κατάσταση της διαδικασίας
Εναλλακτικά σενάρια	-
Μετα-συνθήκες	Οι φοιτητές μπορούν να ξεκινήσουν την ηλεκτρονική διαδικασία των δηλώσεων

1.9.18 Κλείσιμο της διαδικασίας δηλώσεων

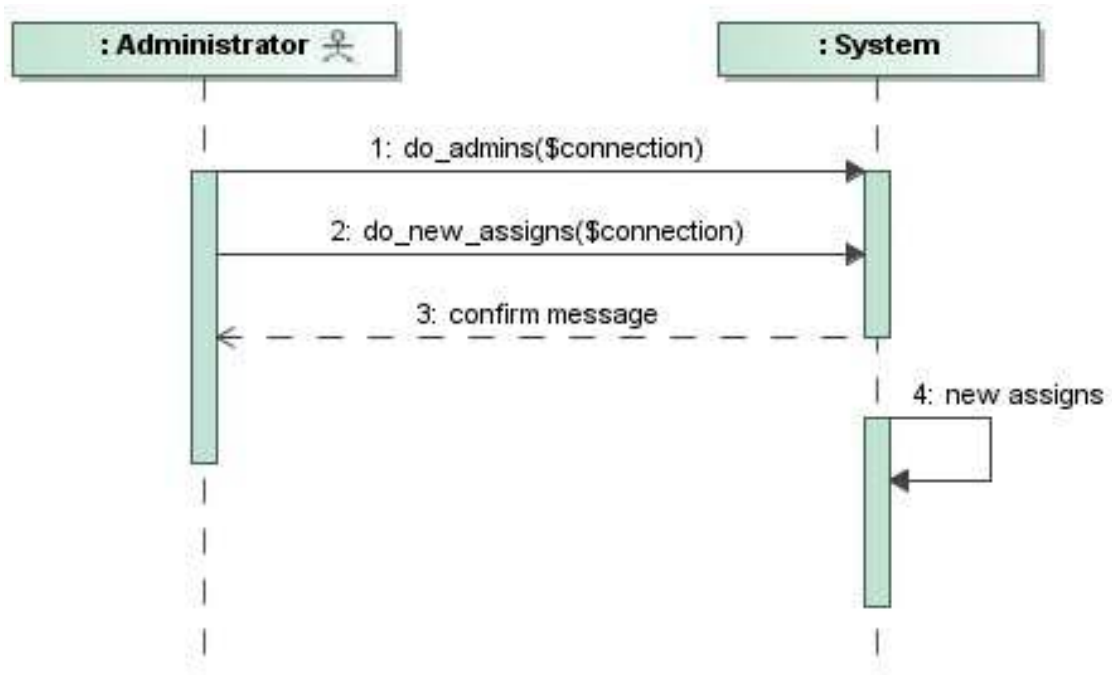
Όνομα σεναρίου χρήσης	Change assigns status
Βασικοί ενδιαφερόμενοι	Διαχειριστής
Προαπαιτούμενα	Ο διαχειριστής είναι συνδεδεμένος στο σύστημα. Η διαδικασία δηλώσεων είναι ανοιχτή.
Αποτέλεσμα επιτυχούς εκτέλεσης	Η διαδικασία των ηλεκτρονικών δηλώσεων έχει κλείσει. Οι φοιτητές δεν μπορούν να δηλώσουν βιβλία, μόνο να δούν τι βιβλία δικαιούνται.
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο διαχειριστής επιλέγει το κλείσιμο της διαδικασίας δηλώσεων από την κεντρική σελίδα διαχείρισης 2. Το σύστημα αλλάζει την κατάσταση της διαδικασίας
Εναλλακτικά σενάρια	-
Μετα-συνθήκες	Οι φοιτητές μπορούν να παρακολουθήσουν ποια βιβλία τους αντιστοιχούν.



Σχήμα XXII: Διάγραμμα ακολουθίας συστήματος «Αλλαγή κατάστασης της διαδικασίας δηλώσεων»

1.9.19 Εκκίνηση νέων δηλώσεων

Όνομα σεναρίου χρήσης	New assigns
Βασικοί ενδιαφερόμενοι	Διαχειριστής
Προαπαιτούμενα	Ο διαχειριστής είναι συνδεδεμένος στο σύστημα. Η διαδικασία των δηλώσεων είναι κλειστή.
Αποτέλεσμα επιτυχούς εκτέλεσης	Εγινε εκκίνηση της διαδικασίας των ηλεκτρονικών δηλώσεων με επιτυχία. Το σύστημα αποθήκευσε ιστορικά στοιχεία των προηγούμενων δηλώσεων
Σενάριο καλής εκτέλεσης	<ol style="list-style-type: none"> 1. Ο διαχειριστής επιλέγει την εκκίνηση νέων δηλώσεων από την κεντρική σελίδα διαχείρισης 2. Το σύστημα εμφανίζει μήνυμα επιβεβαίωσης 3. Το σύστημα ξεκινά νέα διαδικασία ηλεκτρονικών δηλώσεων
Εναλλακτικά σενάρια	2. Ο διαχειριστής ακυρώνει την εκκίνηση της διαδικασίας <ol style="list-style-type: none"> a. Το σύστημα διατηρεί την τρέχουσα κατάσταση της διαδικασίας δηλώσεων
Μετα-συνθήκες	Μετά την εκκίνηση της διαδικασίας των δηλώσεων οι φοιτητές μπορούν να ξεκινήσουν νέα ηλεκτρονική δήλωση.



Σχήμα XXIII: Διάγραμμα ακολουθίας συστήματος «Εκκίνηση νέων δηλώσεων»

1.10 Επίλογος

Μετά την παρουσίαση των βασικότερων διαγραμμάτων UML που πραγματοποιήθηκε στο πρώτο κεφάλαιο, ο αναγνώστης έχει σχηματίσει μια σαφέστερη εικόνα του συστήματος που αναπτύχθηκε, κατανοώντας τις βασικές του λειτουργίες. Στο κεφάλαιο που ακολουθεί γίνεται λεπτομερής περιγραφή της βάσης δεδομένων με σχετικούς πίνακες, διαγράμματα και την αντίστοιχη επεξήγηση.

Βάση δεδομένων

2.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζεται η ανάλυση της βάσης δεδομένων που χρησιμοποιήθηκε στην εφαρμογή καθώς και η τεκμηρίωσή της παραθέτοντας τους σχετικούς πίνακες και τα διαγράμματα.

2.2 Τι είναι MySQL

Η MySQL είναι ένα πολύ γρήγορο δυνατό σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Ακολουθεί το σχεσιακό μοντέλο (relational) και είναι συμβατή με ANSI-SQL. Ο MySQL διακομιστής ελέγχει την πρόσβαση στα δεδομένα ώστε να διασφαλίσει ότι πολλοί χρήστες θα μπορούν να δουλεύουν ταυτόχρονα, ώστε να επιτυγχάνεται γρηγορότερη πρόσβαση και να διασφαλιστεί η πιστοποίηση των χρηστών. Συνεπώς η MySQL είναι ένας πολυνηματικός διακομιστής πολλαπλών χρηστών. Χρησιμοποιεί την SQL (Structured Query Language), την τυπική γλώσσα ερωτημάτων για βάσεις δεδομένων. Η MySQL είναι διαθέσιμη από το 1996 αλλά η ιστορία της ξεκινάει από το 1979. Είναι παγκοσμίως η πιο δημοφιλής βάση δεδομένων ανοικτού κώδικα. Επίσης η έκδοση MySQL Server είναι κατάλληλη για τη διαχείριση μεγαλύτερων βάσεων δεδομένων και την παρουσίαση δεδομένων σε web site στο internet. Παρέχει υψηλή απόδοση και ασφάλεια. Συνοπτικά μπορεί κανείς να αναφέρει τα πλεονεκτήματα της MySQL:

- ✓ Είναι οικονομική (και μερικές φορές δωρεάν).
- ✓ Είναι πιο κατάλληλη για χρήση στο Internet.
- ✓ Είναι ιδιαίτερα βελτιστοποιημένη για ταχύτητα στην ανάκτηση δεδομένων.
- ✓ Παρέχει ευκολίες στο backup.
- ✓ Είναι συμβατή και μεταφέρσιμη σε διάφορες πλατφόρμες και για διάφορα εργαλεία ανάπτυξης.

2.3 Κανονικοποίηση

Η θεωρία της κανονικοποίησης αναπτύχθηκε με στόχο να ανιχνεύονται και να προλαμβάνονται προβληματικές καταστάσεις όταν η εφαρμογή είναι ακόμη στο στάδιο του σχεδιασμού της. Τέτοιες καταστάσεις είναι η διπλοεγγραφή δεδομένων, τα ασυνεπή δεδομένα κ.ά.

2.3.1 Μη Κανονική Μορφή Στη μη κανονική μορφή τα πεδία τοποθετούνται το ένα δίπλα στο άλλο με μόνο επιπλέον στοιχείο πληροφορίας την ομαδοποίηση πεδίων που αντιστοιχούν σε χαρακτηριστικά που παίρνουν πολλαπλές τιμές. Η ομαδοποίηση των τελευταίων αναπαριστάται με ζεύγη εσωτερικών παρενθέσεων στην περιγραφή της δομής του αντίστοιχου πίνακα .

2.3.2 Πρώτη Κανονική Μορφή (1NF) Για να έρθει ένας πίνακας σε πρώτη κανονική μορφή, θα πρέπει να απομακρυνθούν οι επαναλαμβανόμενες ομάδες πεδίων, έτσι ώστε η τομή μιας γραμμής και μιας στήλης του πίνακα, να αντιστοιχεί πάντα σε μια απλή τιμή. Η πρώτη κανονική μορφή δημιουργήθηκε για να αποτρέψει την εμφάνιση στα πεδία ενός πίνακα σύνθετων τιμών, πολλαπλών τιμών καθώς και συνδυασμούς αυτών των δύο. Ένας πίνακας βρίσκεται σε 1NF, όταν η τιμή του κάθε πεδίου σε κάθε εγγραφή, είναι ατομική δηλαδή δεν μπορεί να διασπαστεί σε μικρότερες μονάδες πληροφορίας. Δηλαδή δεν επιτρέπεται να υπάρχουν πεδία στον πίνακα που να παίρνουν πολλαπλές τιμές.

2.3.3 Δεύτερη Κανονική Μορφή (2NF) Η δεύτερη κανονική μορφή προκύπτει από την πρώτη κανονική μορφή εάν μετασχηματιστεί η δομή του πίνακα με τέτοιο τρόπο ώστε να απομακρυνθούν όλες οι μερικές συναρτησιακές εξαρτήσεις (partial dependencies) που υφίστανται ανάμεσα στα πεδία του. Με άλλα λόγια ένας πίνακας βρίσκεται σε δεύτερη κανονική μορφή όταν όλα τα πεδία που δεν ανήκουν στο πρωτεύον κλειδί του πίνακα εξαρτώνται συναρτησιακώς μόνο από τα πεδία του πρωτεύοντος κλειδιού και μάλιστα, μέσω πλήρους συναρτησιακής εξάρτησης (full dependency). Επειδή ο ορισμός της πλήρους και της μερικής συναρτησιακής εξάρτησης μεταξύ δύο πεδίων X και Y εφαρμόζεται μόνο όταν το X περιλαμβάνει περισσότερα από ένα πεδία, είναι προφανές πως η αναγωγή ενός πίνακα στη δεύτερη κανονική μορφή, έχει νόημα μόνο όταν το πρωτεύον κλειδί του είναι σύνθετο. Αντίθετα, όταν το κλειδί του πίνακα είναι ένα απλό πεδίο, η αναγωγή του πίνακα σε 1NF τον μετασχηματίζει αυτόματα και σε 2NF. Ένας πίνακας είναι σε δεύτερη κανονική μορφή όταν είναι σε πρώτη κανονική μορφή και επιπλέον το κάθε πεδίο του που δεν συμμετέχει στο σχηματισμό του κύριου κλειδιού εξαρτάται από το σύνολο του κυρίως κλειδιού και τίποτα λιγότερο.

2.3.4 Τρίτη Κανονική Μορφή Η τρίτη κανονική μορφή προκύπτει από τη δεύτερη κανονική μορφή εάν ο πίνακας μετασχηματιστεί με τέτοιο τρόπο ώστε να απομακρυνθούν όλες οι μεταβατικές συναρτησιακές εξαρτήσεις (transitive dependencies) που υφίστανται ανάμεσα στα πεδία του. Σημειώνεται ότι μια συναρτησιακή εξάρτηση $A \rightarrow B$ ονομάζεται μεταβατική, όταν υπάρχει ένα πεδίο C που δεν ανήκει στο πρωτεύον κλειδί του πίνακα, τέτοιο ώστε $C \rightarrow A$ και $B \rightarrow C$. Ένας πίνακας είναι σε τρίτη κανονική μορφή όταν είναι σε δεύτερη κανονική μορφή και επιπλέον δεν υπάρχει πεδίο που να μην συμμετέχει στο σχηματισμό του κύριου κλειδιού το οποίο να εξαρτάται συναρτησιακά από άλλο πεδίο που επίσης δεν συμμετέχει στο σχηματισμό του κύριου κλειδιού. Εξαίρεση είναι το εναλλακτικό κλειδί όταν αυτό υπάρχει. Παρακάτω, γίνεται η ανάλυση του κάθε πίνακα της εφαρμογής ξεχωριστά κι έπειτα παρουσιάζεται η ολοκληρωμένη βάση δεδομένων στην τρίτη κανονική μορφή της.

2.4 Σχεσιακό Μοντέλο Το σχεσιακό μοντέλο αναπαράστασης των δεδομένων μίας εφαρμογής (relational data model) καθιερώθηκε από τον Codd το 1970 και αποτέλεσε ένα από τα πιο απλά και ευέλικτα μοντέλα αυτού του είδους. Είναι ένα γράφημα που απεικονίζει τις συσχετίσεις ανάμεσα στις οντότητες της βάσης. Σε αυτό το μοντέλο, τα δεδομένα μιας εφαρμογής αναπαρίστανται ως ένα σύνολο από σχέσεις (relations) οι οποίες μπορεί να είναι πίνακες ή αρχεία. Στις πιο πολλές περιπτώσεις υιοθετείται η χρήση πινάκων (tables) που περιέχουν ένα πλήθος γραμμών (rows) και στηλών (columns). Η κάθε μια από αυτές τις γραμμές οι οποίες στην ορολογία του μοντέλου ονομάζονται και πλειάδες (tuples) περιέχει ένα σύνολο απλών πεδίων (attributes) τα οποία συσχετίζονται μεταξύ τους.

Κάθε πεδίο ή στήλη ενός πίνακα, δέχεται τιμές οι οποίες ανήκουν σε ένα συγκεκριμένο και καθορισμένο εκ των προτέρων σύνολο τιμών (domain). Το είδος των τιμών αυτού του συνόλου καθορίζεται από τον τύπο δεδομένων του πεδίου του πίνακα, ο οποίος με τη σειρά του ορίζεται κατά το στάδιο της λογικής σχεδίασης της εφαρμογής. Οι τιμές που καταχωρούνται σε μια πλειάδα, θα πρέπει να είναι ατομικές (atomic) να μην υποδιαιρούνται δηλαδή σε μικρότερες μονάδες πληροφορίας. Σύνθετες τιμές δεδομένων, δεν μπορούν να χρησιμοποιηθούν αλλά θα πρέπει να γραφούν με τρόπο που να μην παραβιάζει την παραπάνω αρχή. Παρακάτω παρατίθενται το σχεσιακό μοντέλο της εφαρμογής.

Admins (id, name, surname, username, password, level, email, landphone, mobile, active, last_seen)

Στον πίνακα admins αποθηκεύονται οι διαχειριστές της εφαρμογής οι οποίοι διαχειρίζονται όλες τις οντότητες του συστήματος.

assigns_is_continued (id, admins_id, active, date)

Ο πίνακας assigns_is_continued υποδηλώνει το αν η διαδικασία επιλογής βιβλίων είναι ακόμα ανοικτή ή όχι. Περιέχει το id του διαχειριστή που άλλαξε την κατάσταση της διαδικασίας.

Authors (id, name_gr)

Στον πίνακα authors καταχωρούνται τα ονόματα των συγγραφέων των συγγραμμάτων.

Books (id, name_gr, name_en, isbn, authors_id, publishers_id, publication_year, publication_places_id, fek, has_images, price, total_price, active)

Στον πίνακα books καταχωρούνται τα στοιχεία των συγγραμμάτων που διανέμονται.

books_images (books_id, type, size)

Για κάθε νέο βιβλίο που εισάγει ο διαχειριστής στην βάση δεδομένων υπάρχει η δυνατότητα να ανεβάσει ένα ενδεικτικό εξώφυλλο του συγγράμματος, προκειμένου ο φοιτητής να έχει μια καλύτερη εικόνα της επιλογής που πρόκειται να κάνει. Ο πίνακας books_images περιέχει πληροφορίες για τις εικόνες των συγγραμμάτων όπως ο τύπος της εικόνας (.jpg) και το μέγεθος της, καθώς και ένα πεδίο books_id που προσδιορίζει σε ποιο βιβλίο ανήκει η συγκεκριμένη εικόνα.

books_to_subjects_proposed (id, books_id, subjects_description_id)

Στον πίνακα books_to_subjects_proposed καταχωρούνται τα προτεινόμενα βιβλία.

publication_places (id, name_gr, name_en)

Στον πίνακα publication_places καταχωρούνται στοιχεία σχετικά με τον τόπο έκδοσης του συγγράμματος.

Publishers (id, name_gr)

Στον πίνακα publishers καταχωρούνται στοιχεία σχετικά με τον εκδοτικό οίκο που εκδίδεται το κάθε βιβλίο.

Schools (id, code, name_gr, name_en)

Ο πίνακας schools περιέχει το όνομα του τμήματος καθώς και τον κωδικό που είναι 3.

Semesters (id, name_gr, name_en)

Ο πίνακας semesters περιέχει όλα τα εξάμηνα.

Students (id, reg_num, name, surname, password, email, landphone, mobile, year_enrolled, semester_enrolled, active, last_seen)

Στον πίνακα students αποθηκεύονται τα στοιχεία όλων των φοιτητών της εφαρμογής οι οποίοι θα έχουν πρόσβαση στο σύστημα για την πραγματοποίηση της ηλεκτρονικής δήλωσης των συγγραμμάτων.

students_to_books_choises (students_id, books_to_subjects_id, date)

Στον πίνακα students_to_books_choises καταχωρούνται οι επιλογές των συγγραμμάτων που έχουν κάνει οι φοιτητές. Σε περίπτωση που κάποιος φοιτητής μετανιώσει για την επιλογή του και ξανα δηλώσει τα βιβλία, οι νέες του επιλογές καταχωρούνται πάλι στον πίνακα students_to_books_choises και οι παλιές διαγράφονται. Όλα αυτά συμβαίνουν με την προϋπόθεση, ότι η διαδικασία επιλογής των διαθέσιμων συγγραμμάτων είναι ανοιχτή. Σε κάθε επανέναρξη της διαδικασίας δηλώσεων τα περιεχόμενα του πίνακα students_to_books_choises μεταφέρονται στον πίνακα students_to_books_final και χρησιμοποιούνται σαν ιστορικά στοιχεία για τον έλεγχο της νομιμότητας της επιλογής των φοιτητών.

students_to_books_final (students_id, books_to_subjects_id, date)

Στον πίνακα students_to_books_final καταχωρούνται οι τελικές επιλογές των συγγραμμάτων που έχουν κάνει οι φοιτητές. Όταν κλείσει η διαδικασία των ηλεκτρονικών δηλώσεων καταχωρούνται οι τελικές προτιμήσεις των φοιτητών στα βιβλία.

Subjects(id, name_gr, name_en, semesters_id, rank, subjects_categories_id, subjects_fields_id)

Στον πίνακα Subjects καταχωρούνται όλα τα μαθήματα της εφαρμογής, τα οποία εμφανίζονται στον χρήστη, ώστε να πραγματοποιήσει την δήλωση των συγγραμμάτων.

subjects_categories (id, name_gr, name_en)

Στον πίνακα subjects_categories είναι καταχωρημένη η κατηγορία του μαθήματος, υποχρεωτικό, προαιρετικό, υποχρεωτικό επιλογής.

Subjects_description (id, subjects_id, subjects_type_id, code, ect, hours, active)

Στον πίνακα subjects description αποθηκεύονται πληροφορίες όπως ο κωδικός του μαθήματος (από τον πίνακα subjects), εάν το μάθημα είναι θεωρία ή εργαστήριο (subjects_types_id από τον πίνακα subjects_types), ο κωδικός του μαθήματος, οι ώρες που διδάσκεται, αν είναι ενεργό καθώς και οι etcs του μαθήματος.

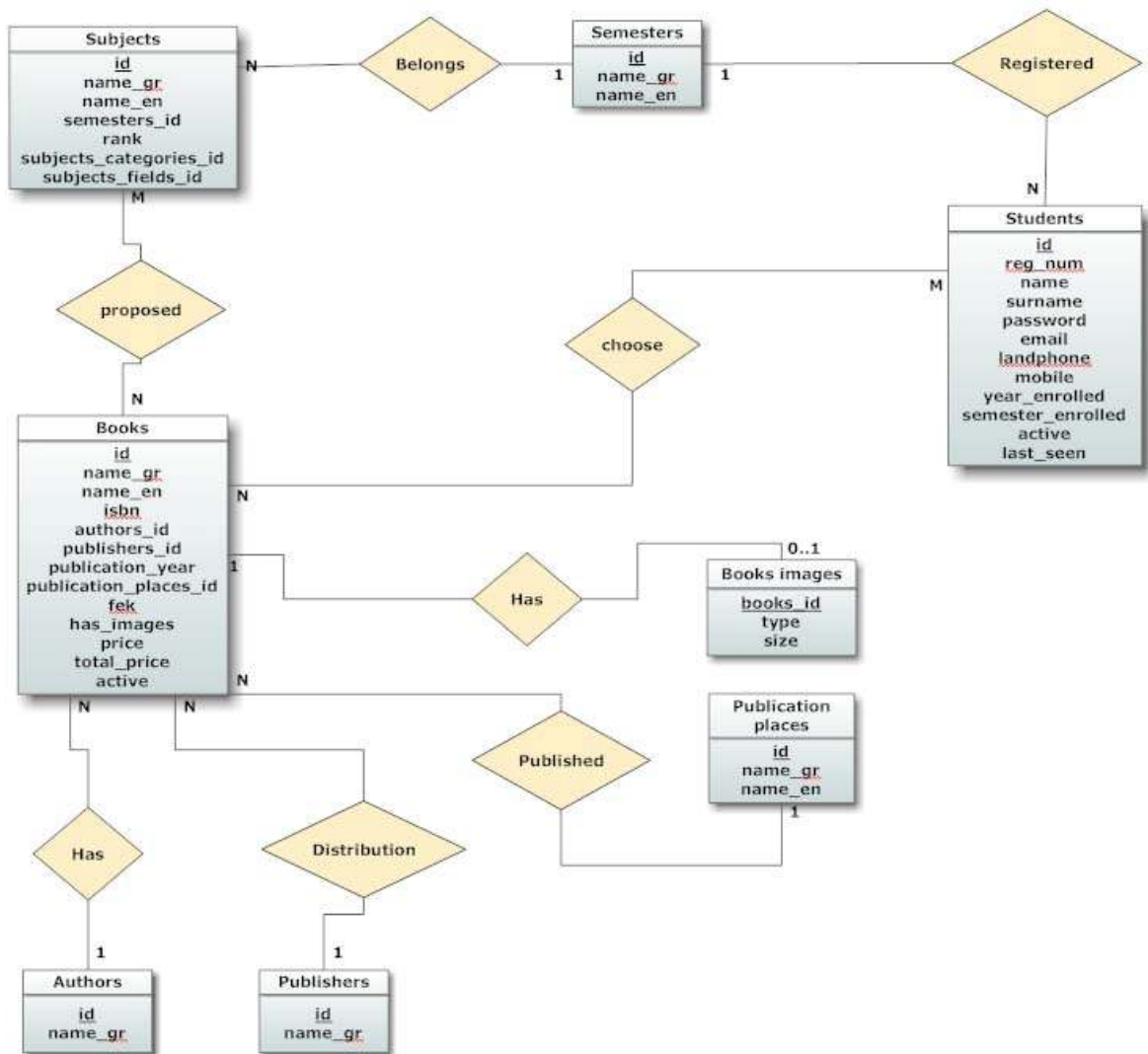
subjects_fields (id, name_gr, name_en)

Στον πίνακα subjects_fields είναι καταχωρημένα στοιχεία όπως εαν το μάθημα είναι ΓΥ,ΕΥ,Ε ή ΔΟΝΑ.

subjects_types (id, name_gr, name_en)

Στον πίνακα subjects_types είναι καταχωρημένος ο τύπος μαθήματος,δηλαδή θεωρία ή εργαστήριο.

2.5 Διάγραμμα Οντοτήτων - Συσχετίσεων (ER) Οντότητες και συσχετίσεις παρουσιάζονται διαγραμματικά , για ένα συγκεκριμένο μοντέλο εφαρμογής, με το λεγόμενο διάγραμμα ER (Entity - Relationship Diagram). Αυτή η διαγραμματική τεχνική, επιτρέπει τη σχεδίαση των οντοτήτων μιας βάσης και των συσχετίσεων που υφίστανται ανάμεσα τους χρησιμοποιώντας ειδικά σύμβολά.Στο Σχήμα Ι φαίνεται το βασικό διάγραμμα ER του συστήματος της εφαρμογής που αναπτύσσεται.



Σχήμα Ι:Διάγραμμα ER εφαρμογής

2.6 Τεκμηρίωση πινάκων Στην συγκεκριμένη εφαρμογή χρησιμοποιούνται οι δεκαοχτώ παραπάνω πίνακες οι οποίοι φαίνονται και στο Σχήμα II το οποίο δείχνει την βάση δεδομένων της εφαρμογής μέσα από το εργαλείο phpmyadmin:



Σχήμα II: Η βάση δεδομένων, μέσα από το εργαλείο phpmyadmin

2.6.1 Περιγραφή πεδίων Ακολουθεί η τεκμηρίωση των πινάκων με όλα τα πεδία και την περιγραφή τους .

Πίνακας: admins	
Πεδίο	Περιγραφή
<u>id</u>	Ο αύξων αριθμός του κάθε διαχειριστή.
name	Το όνομα του διαχειριστή.
surname	Το επίθετο του διαχειριστή.
username	Το όνομα χρήστη του διαχειριστή
password	Ο κωδικός του διαχειριστή.
level	Το επίπεδο εξουσιοδότησης του διαχειριστή.
email	Το email του διαχειριστή

Πεδίο	Περιγραφή
landphone	Το σταθερό τηλέφωνο του διαχειριστή.
mobile	Το κινητό τηλέφωνο του διαχειριστή.
active	Αν ο διαχειριστής είναι ενεργός ή όχι.
last_seen	Η ημερομηνία της τελευταίας εισόδου του διαχειριστή στο σύστημα.

Πίνακας: assigns_is_continued

Πεδίο	Περιγραφή
<u>id</u>	Ο αύξων αριθμός
admins_id	Το id του διαχειριστή που άλλαξε την κατάσταση της διαδικασίας
Active	Αν η διαδικασία είναι ανοιχτή ή κλειστή
date	Η ημερομηνία αλλαγής της κατάστασης

Πίνακας: authors

Πεδίο	Περιγραφή
<u>id</u>	Ο αύξων αριθμός του κάθε συγγραφέα.
name_gr	Το όνομα του συγγραφέα στα ελληνικά.

Πίνακας: books

Πεδίο	Περιγραφή
<u>id</u>	Ο αύξων αριθμός του κάθε βιβλίου.
name_gr	Το όνομα του βιβλίου στα ελληνικά.
name_en	Το όνομα του βιβλίου στα αγγλικά.
isbn	Το isbn του βιβλίου.

Πεδίο	Περιγραφή
authors_id	Ο κωδικός του συγγραφέα του κάθε βιβλίου.
publishers_id	Ο κωδικός του εκδοτικού οίκου που εξέδωσε το βιβλίο.
publication_year	Η χρονιά έκδοσης του βιβλίου.
publication_places_id	Ο κωδικός του τόπου έκδοσης του βιβλίου.
fek	Το ΦΕΚ που ενέκρινε το βιβλίο.
has_images	Εάν το βιβλίο έχει φωτογραφία.
price	Η τιμή του αντιτύπου που ορίζεται από το ΦΕΚ.
total_price	Η συνολική τιμή που εγκρίθηκε για όλα τα βιβλία.
active	Η κατάσταση του βιβλίου.Εάν είναι ενεργό ή όχι

Πίνακας: books_images

Πεδίο	Περιγραφή
<u>books_id</u>	Ο κωδικός που αντιστοιχεί στο κάθε βιβλίο.Χρησιμοποιείται για την αντιστοίχιση βιβλίου εικόνας.
type	Ο τύπος της εικόνας (.jpg .gif κτλ)
size	Το μέγεθος της εικόνας.

Πίνακας: books_to_subjects_proposed

Πεδίο	Περιγραφή
<u>id</u>	Ο αύξων κωδικός της αντιστοίχισης του βιβλίου με το κάθε τύπο μαθήματος.

Πεδίο	Περιγραφή
books_id	Ο κωδικός του βιβλίου που αντιστοιχεί στο μάθημα.
subjects_description_id	Ο κωδικός περιγραφής του μαθήματος.

Πίνακας: publication_places

Πεδίο	Περιγραφή
<u>id</u>	Ο αύξων αριθμός του κάθε τόπου έκδοσης.
name_gr	Το όνομα του τόπου έκδοσης στα ελληνικά.
name_en	Το όνομα του τόπου έκδοσης στα αγγλικά.

Πίνακας: publishers

Πεδίο	Περιγραφή
<u>id</u>	Ο αύξων αριθμός του κάθε εκδοτικού οίκου.
name_gr	Το όνομα του εκδοτικού οίκου στα ελληνικά.

Πίνακας: schools

Πεδίο	Περιγραφή
<u>id</u>	Ο αύξων αριθμός κάθε τμήματος.
code	Ο κωδικός που αντιστοιχεί στο τμήμα
name_gr	Το όνομα του κάθε τμήματος στα ελληνικά.
name_en	Το όνομα του κάθε τμήματος στα αγγλικά.

Πίνακας: semesters	
Πεδίο	Περιγραφή
<u>id</u>	Ο κωδικός κάθε εξαμήνου.
name_gr	Το όνομα του εξαμήνου στα ελληνικά.
name_en	Το όνομα του εξαμήνου στα αγγλικά.

Πίνακας: students	
Πεδίο	Περιγραφή
<u>id</u>	Ο αύξων αριθμός του κάθε φοιτητή.
reg_num	Ο αριθμός μητρώου του φοιτητή (πχ 2487).
name	Το όνομα του φοιτητή.
surname	Το επίθετο του φοιτητή.
password	Ο κωδικός πρόσβασης του φοιτητή.
email	Το email του φοιτητή.
landphone	Το σταθερό τηλέφωνο του φοιτητή.
mobile	Το κινητό τηλέφωνο του φοιτητή.
year_enrolled	Η χρονιά που εισήχθηκε στο τμήμα.
semester_enrolled	Το εξάμηνο που εισήχθηκε στο τμήμα (π.χ. Α ή Β).
active	Αν ο φοιτητής συνεχίζει να είναι εγγεγραμμένος.
last_seen	Η ημερομηνία της τελευταίας εισόδου του φοιτητή στο σύστημα.

Πίνακας: students_to_books_choises	
Πεδίο	Περιγραφή
<u>students_id</u>	Ο κωδικός του φοιτητή που επιλέγει το βιβλίο.
<u>books_to_subjects_id</u>	Ο κωδικός του βιβλίου που δίνεται στο συγκεκριμένο μάθημα.
date	Η ημερομηνία που δόθηκε το βιβλίο.

Πίνακας: students_to_books_final	
Πεδίο	Περιγραφή
<u>students_id</u>	Ο κωδικός του φοιτητή που επιλέγει το βιβλίο.
<u>books_to_subjects_id</u>	Ο κωδικός του βιβλίου που τελικά δόθηκε στο συγκεκριμένο μάθημα.
date	Η ημερομηνία που δόθηκε το βιβλίο.

Πίνακας: subjects	
Πεδίο	Περιγραφή
<u>id</u>	Ο αύξων αριθμός κάθε μαθήματος.
name_gr	Το όνομα του μαθήματος στα ελληνικά.
name_en	Το όνομα του μαθήματος στα αγγλικά.
semesters_id	Το εξάμηνο που διδάσκεται το μάθημα.
rank	Η θέση του μαθήματος στο εξάμηνο
subjects_categories_id	Αν το μάθημα είναι υποχρεωτικό ή επιλογής.
subjects_fields_id	Αν το μάθημα είναι ΓΥ,ΕΥ,Ε ή ΔΟΝΑ

Πίνακας: subjects_categories	
Πεδίο	Περιγραφή
<u>id</u>	Ο αύξων αριθμός της κάθε κατηγορίας μαθήματος.
name_gr	Το όνομα της κάθε κατηγορίας (υποχρεωτικό, επιλογής) στα ελληνικά.
name_en	Το όνομα της κάθε κατηγορίας (υποχρεωτικό, επιλογής) στα αγγλικά.

Πίνακας: subjects_description	
Πεδίο	Περιγραφή
<u>id</u>	Ο αύξων αριθμός που αντιστοιχεί στο κωδικό του μαθήματος και στον τύπο.
subjects_id	Ο κωδικός του μαθήματος.
subjects_type_id	Ο κωδικός του τύπου μαθήματος (αν είναι θεωρία ή εργαστήριο).
code	Ο κωδικός του μαθήματος (π.χ. 3 10 1 1).
ects	Οι διδακτικές μονάδες που αντιστοιχούν στο μάθημα.
hours	Οι ώρες που αντιστοιχούν στο μάθημα. Πόσες ώρες διδάσκεται δηλαδή.
active	Εάν το μάθημα είναι ενεργό ή όχι.

Πίνακας: subjects_fields	
Πεδίο	Περιγραφή
<u>id</u>	Ο αύξων αριθμός κάθε κατηγορίας μαθήματος
name_gr	Το όνομα της κατηγορίας στα ελληνικά
name_en	Το όνομα της κατηγορίας στα αγγλικά

Πίνακας: subjects_types	
Πεδίο	Περιγραφή
<u>id</u>	Ο αύξων αριθμός του κάθε τύπου μαθήματος.
name_gr	Το όνομα του κάθε τύπου μαθήματος (θεωρία, εργαστήριο) στα ελληνικά.
name_en	Το όνομα του κάθε τύπου μαθήματος (θεωρία, εργαστήριο) στα αγγλικά.

2.7 SQL injection και προστασία βάσης δεδομένων

Ο όρος SQL injection περιγράφει την τεχνική εισαγωγής κώδικα η οποία εκμεταλλεύεται πιθανά κενά ασφαλείας στην βάση δεδομένων μίας εφαρμογής. Το κενό στην ασφάλεια έγκειται στο μη αποτελεσματικό φιλτράρισμα δεδομένων από τον χρήστη με συνέπεια να υπάρχουν μέσα σε αυτά ένθετα SQL ερωτήματα ή σε περίπτωση που τα δεδομένα που εισάγει ο χρήστης δεν είναι ισχυρώς τυποποιημένα (strongly typed) υπάρξει απρόσμενη εκτέλεση κώδικα. Παρακάτω εξετάζονται τα δύο αυτά ενδεχόμενα.

2.7.1 Αναποτελεσματικό φιλτράρισμα δεδομένων

Η μορφή αυτής της SQL injection συμβαίνει όταν ο χρήστης εισάγει δεδομένα τα οποία δεν ελέγχονται για χαρακτήρες διαφυγής και έτσι περνούν όπως είναι μέσα σε μία SQL έκφραση. Αυτό έχει ως αποτέλεσμα πιθανή εισαγωγή κώδικα από τους χρήστες της εφαρμογής. Ο παρακάτω κώδικας δείχνει μία τέτοια περίπτωση:

```
Statement = "SELECT * FROM admins WHERE name = " + username + ";"
```

Ο κώδικας αυτός προορίζεται για να επιλέγει συγκεκριμένα ονόματα διαχειριστών (username) από τον πίνακα των διαχειριστών(admins) . Όμως, αν η μεταβλητή username πάρει κάποιες συγκεκριμένες τιμές όπως είναι η παρακάτω τότε θα υπάρχει απρόσμενη εκτέλεση κώδικα :

```
A' or 't'='t
```

Έτσι τώρα η έκφραση statement γίνεται :

```
SELECT * FROM admins WHERE name = 'a' OR't'='t';
```

Αν ο κώδικας αυτός χρησιμοποιείται σε μία συνάρτηση πιστοποίησης χρήστη τότε θα είχε ως αποτέλεσμα την επιλογή όλων των διαχειριστών αφού η έκφραση 't'='t' είναι πάντα αληθής . Αν και οι περισσότεροι SQL server επιτρέπουν την πολλαπλή εκτέλεση SQL εκφράσεων , αυτό δεν γίνεται από το ρηρ mysql_query για λόγους ασφαλείας. Αυτό αποτρέπει κακόβουλους χρήστες να εισάγουν εντελώς ξεχωριστά ερωτήματα προς την βάση αλλά παρόλα αυτά δεν αποτρέπει την τροποποίηση των ερωτημάτων. Η ακόλουθη τιμή της μεταβλητής username θα έχει ως αποτέλεσμα την διαγραφή του πίνακα admins καθώς και την επιλογή όλων των δεδομένων από τον πίνακα books.

```
A'; DROP TABLE admins; SELECT * FROM books WHERE name LIKE '%
```

Έτσι τώρα η έκφραση statement γίνεται:

```
SELECT * FROM admins WHERE name = 'a'; DROP TABLE admins; SELECT * FROM books WHERE name LIKE '%';
```

2.7.2 Λανθασμένος χειρισμός τύπων δεδομένων

Αυτή η μορφή της SQL injection συμβαίνει όταν ένα πεδίο εισαγωγής δεδομένων δεν ελέγχεται για περιορισμούς τύπων δεδομένων . Αυτό μπορεί να συμβεί όταν ένα αριθμητικό πεδίο πρόκειται να χρησιμοποιηθεί σε μία SQL έκφραση αλλά ο προγραμματιστής δεν ελέγχει τα δεδομένα που εισάγει ο χρήστης ώστε να διαπιστώσει αν αυτά είναι όντως αριθμητικά . Για παράδειγμα :

```
Statement = "SELECT * FROM books WHERE id = " + a_variable + ";
```

Είναι προφανές ότι από την δήλωση αυτή ότι ο προγραμματιστής περιμένει να εισαχθούν αριθμητικά δεδομένα για το πεδίο "id". Παρόλα αυτά , αν δοθούν δεδομένα τύπου string, τότε ο χρήστης μπορεί να χειριστεί την SQL έκφραση όπως αυτός επιλέξει . Για παράδειγμα , θέτοντας τη μεταβλητή a_variable σε :


```
1;DROP TABLE admins
```

θα διαγραφεί ο πίνακας admins από την βάση δεδομένων , αφού η SQL θα εκτιμήσει την έκφραση ως:

```
SELECT * FROM books WHERE id=1; DROP TABLE admins;
```

2.8 Αποφυγή SQL Injection

Ενας τρόπος για την αποφυγή SQL Injection attack είναι η αφαίρεση χαρακτήρων διαφυγής από πεδία εισαγωγής δεδομένων, παραμέτρους στο URL και τιμές από cookies. Στην εφαρμογή οι μέθοδοι `sanitize` και `mysql_prep` χρησιμοποιούνται για την αποφυγή τέτοιων περιπτώσεων. Οι `sanitize` ελέγχουν αν όντως τα δεδομένα που πληκτρολογεί ο χρήστης είναι αυτά που θεωρούμε εμείς ως λογικά (πχ σε αριθμητικό πεδίο αν είναι αριθμητικά δεδομένα).

2.9 Επίλογος

Με την παρουσίαση της βάσης δεδομένων ολοκληρώνεται το πρώτο μέρος της τεκμηρίωσης. Ο αναγνώστης πλέον έχει αποκτήσει μια ολοκληρωμένη εικόνα σχετικά με την λειτουργία του συστήματος.

Είναι εύκολα κατανοητό ότι η σχεδίαση της βάσης δεδομένων αποτελεί κλειδί κατά την διαδικασία της ανάλυσης καθώς το παραμικρό λάθος μπορεί να αλλοιώσει ολόκληρη τη λειτουργία της εφαρμογής. Η σωστή σχεδίαση του υπο ανάπτυξη συστήματος αποτελεί βασική προϋπόθεση, πριν ακολουθήσει η υλοποίηση του.

Στο δεύτερο μέρος της τεκμηρίωσης γίνεται η παρουσίαση της διαδικασίας σχεδίασης του συστήματος. Το στάδιο της σχεδίασης αφορά τη φυσική σχεδίαση του συστήματος δηλαδή την αξιοποίηση των διαθέσιμων τεχνολογιών για την κωδικοποίηση της εφαρμογής. Στα κεφάλαια που ακολουθούν γίνεται παρουσίαση των τεχνολογιών που χρησιμοποιούνται καθώς και μερικές αναφορές στα επιμέρους κομμάτια κάθε τεχνολογίας ώστε να μπορούν να γίνουν κατανοητές και αντιληπτές από τους αναγνώστες.

ΔΕΥΤΕΡΟ ΜΕΡΟΣ - ΣΧΕΔΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

ΚΕΦΑΛΑΙΟ 3^ο

HTML

3.1 Εισαγωγή

Ο βασικός πυρήνας των ιστοσελίδων που υπάρχουν στο Ίντερνετ είναι γραμμένος στην γλώσσα προγραμματισμού HTML (Hypertext Markup Language) . Η HTML η οποία όπως υποδηλώνει το όνομα της, αποτελεί μια markup γλώσσα για την περιγραφή hypertext κειμένων η οποία πρωτοεμφανίστηκε στις αρχές του 1990. Η γλώσσα αυτή χρησιμοποιείται στο WWW (World Wide Web) και αποτελεί υποσύνολο της γλώσσας SGML (Standard Generalized Markup Language) που επινοήθηκε από την IBM προκειμένου να λυθεί το πρόβλημα της μη τυποποιημένης εμφάνισης κειμένων σε διάφορα υπολογιστικά συστήματα. Η γλώσσα αυτή έχει τυποποιηθεί από τον παγκόσμιο οργανισμό τυποποίησης τεχνολογιών του Web, το World Wide Web Consortium (W3C).

3.2 World Wide Web Consortium (W3C)

Το World Wide Web Consortium (W3C) είναι μια διεθνής κοινοπραξία όπου οι Οργανισμοί Μέλη, το προσωπικό πλήρους απασχόλησης και το κοινό δουλεύουν μαζί για να αναπτύξουν πρότυπα του Παγκοσμίου Ιστού. Η αποστολή του W3C είναι να οδηγήσει τον Παγκόσμιο Ιστό στο μέγιστο των δυνατοτήτων του, αναπτύσσοντας πρωτόκολλα και οδηγίες που εξασφαλίζουν μακροπρόθεσμη ανάπτυξη του Παγκοσμίου Ιστού. Το W3C απαρτίζεται από οργανισμούς σε όλο τον κόσμο που ανήκουν σε ποικίλα πεδία και στόχος τους είναι να συμμετάσχουν σε έναν ουδέτερο τόπο συζήτησης για τη δημιουργία προτύπων του Παγκοσμίου Ιστού. Το W3C επιδιώκει την αποστολή του πρωταρχικά μέσα από τη δημιουργία προτύπων του Παγκοσμίου Ιστού και οδηγιών. Για να φτάσει ο Παγκόσμιος Ιστός στο μέγιστο των δυνατοτήτων του, οι πιο βασικές τεχνολογίες του Παγκοσμίου Ιστού θα πρέπει να είναι συμβατές μεταξύ τους και να επιτρέπουν στον εξοπλισμό (hardware) και στο λογισμικό που χρησιμοποιείται να έχουν πρόσβαση στον Παγκόσμιο Ιστό και να συνεργάζονται. Το W3C αναφέρεται σε αυτό το στόχο ως "δια-λειτουργικότητα στον Παγκόσμιο Ιστό" ("Web interoperability"). Με την έκδοση ανοιχτών προτύπων για τις γλώσσες και τα πρωτόκολλα του Παγκοσμίου Ιστού, το W3C επιδιώκει να αποφύγει κατακερματισμό της αγοράς και άρα τον κατακερματισμό του Παγκοσμίου Ιστού.

3.3 Στοιχεία της HTML

Όταν το 1989 ο Tim Berners Lee ανακάλυψε την Hypertext Markup Language, κανένας δεν μπορούσε να αναλογιστεί την αλματώδη ανάπτυξη στην οποία θα οδηγούσε. Τα πρώτα χρόνια, η HTML χρησιμοποιούνταν μόνο για την δημιουργία στατικών ιστοσελίδων όπως επίσης και για την διάταξη των εγγράφων. Η HTML συνεχίζει να παραμένει, 20 χρόνια περίπου μετά την δημιουργία της, ιεραρχικά δομημένη και αποτελούμενη από ετικέτες.

Οι ετικέτες (tags) αυτές περικλείουν και διαφοροποιούν τα bit κειμένου, υποδεικνύοντας την λειτουργία και τον σκοπό του κειμένου που βρίσκεται ανάμεσα στις ετικέτες.

Οι ετικέτες είναι γραμμένες απευθείας σε μορφή απλού κειμένου στο html έγγραφο όπου μπορούν να διερμηνευτούν από το λογισμικό του υπολογιστή.

Οι ετικέτες αυτές καθ' εαυτές δεν παρουσιάζονται στον browser και είναι ξεχωριστές από το περιεχόμενο που περικλείουν. Σημειώνεται ότι ανοίγουν με τη μορφή <tag> και κλείνουν με τη μορφή </tag>. Για παράδειγμα το <p> </p> ανοίγει και κλείνει μία παράγραφο.

3.4 Χρήση HTML στην εφαρμογή

Παρακάτω παρουσιάζονται οι βασικότερες HTML ετικέτες που χρησιμοποιήθηκαν στην εφαρμογή καθώς και η περίληψη των διαθέσιμων ιδιοτήτων τους όπου αυτό απαιτείται.

3.4.1 Ετικέτα <html>

Το στοιχείο αυτό είναι το πρώτο το οποίο χρησιμοποιείται στο έγγραφο και περικλείει όλα τα υπόλοιπα μέσα σε αυτό. Επίσης, είναι γνωστό ως στοιχείο ρίζα (root element) και δεν έχει ιδιότητες και αυτό που κάνει είναι να καθορίζει που θα αρχίζει και που θα τελειώνει το έγγραφο.

3.4.2 Ετικέτα <head>

Η ετικέτα αυτή περιέχει πληροφορίες σχετικά με το HTML έγγραφο, συμπεριλαμβάνει τις λέξεις κλειδιά που περιγράφουν την ιστοσελίδα καθώς και συνδέσμους σε άλλα αρχεία τα οποία χρησιμοποιεί το έγγραφο όπως τα αρχεία CSS και JavaScript. Τίποτα από ό,τι υπάρχει στο τμήμα του <head> δεν εμφανίζεται στον browser του χρήστη εκτός από τα περιεχόμενα της ετικέτας <title> όπου αυτά τυπώνονται στην αντίστοιχη μπάρα του browser.

3.4.3 Ετικέτα <link>

Η ετικέτα αυτή ορίζει τις σχέσεις μεταξύ δύο συνδεδεμένων εγγράφων. Η πιο κοινή του χρήση είναι στο να συνδέει εξωτερικά φύλλα στυλ (external style sheets) στο έγγραφο.

3.4.4 Ετικέτα <meta>

Η ετικέτα αυτή παρέχει πληροφορίες σχετικά με το έγγραφο. Οι μηχανές αναζήτησης συχνά χρησιμοποιούν αυτές τις πληροφορίες κατά τις αναζητήσεις στο Ίντερνετ. Με την χρήση αυτής της ετικέτας παρέχονται λέξεις κλειδιά και περιγραφές ώστε να ταξινομηθεί η τρέχουσα ιστοσελίδα κατά την αναζήτηση.

3.4.5 Ετικέτα <script>

Η ετικέτα <script> παίζει σημαντικό ρόλο στη δυναμικότητα της ιστοσελίδας. Επιτρέπει την προσθήκη γλωσσών script στην ιστοσελίδα όπως είναι η JavaScript.

3.4.6 Ετικέτα <style>

Η ετικέτα αυτή δημιουργεί ένα εσωτερικό στυλ (internal style) στο έγγραφο.

3.4.7 Ετικέτα <body>

Η ετικέτα <body> έρχεται αμέσως μετά την ετικέτα <head> και πρέπει να κλείσει ακριβώς πριν από την </html>. Το στοιχείο <body> είναι επιπέδου μπλοκ και μπορεί να περιέχει μόνο μπλοκ επιπέδου στοιχεία. Τα στοιχεία επιπέδου μπλοκ καταλαμβάνουν όλο το διαθέσιμο πλάτος της ιστοσελίδας κάτι που σημαίνει ότι δύο στοιχεία επιπέδου μπλοκ θα εμφανιστούν το ένα κάτω από το άλλο. Αντίθετα, τα ένθετα στοιχεία εμφανίζονται το ένα δίπλα στο άλλο. Σημειώνεται ότι οποιοδήποτε κείμενο ή ένθετο στοιχείο πρέπει να βρίσκεται σε κάποιο μπλοκ επιπέδου στοιχείο και όχι απευθείας στο <body>.

3.4.8 Ετικέτα Κεφαλίδας

Η HTML έχει έξι επίπεδα κεφαλίδων από το 1 μέχρι το 6 (<h1>, <h2>, <h3>, <h4>, <h5>, <h6>) με το πρώτο επίπεδο να είναι το ανώτερο. Οι κεφαλίδες τυπικά εμφανίζονται με μεγαλύτερες ή και πιο έντονες γραμματοσειρές από ότι το κανονικό κείμενο. Η πρώτη κεφαλίδα σε κάθε κείμενο θα πρέπει να έχει το σημάδι <h1>.

3.4.9 Ετικέτα <p>

Η ετικέτα αυτή χρησιμοποιείται για τον διαχωρισμό των προτάσεων κειμένου κατά την εμφάνισή τους από τον browser. Οι παράγραφοι είναι στοιχεία επιπέδου μπλοκ και πρέπει να περιέχουν μόνο κείμενο και ένθετα στοιχεία.

**3.4.10 Ετικέτα **

Μία λίστα χρησιμοποιείται σε περιπτώσεις όπου η σειρά των στοιχείων μέσα σε αυτήν δεν έχει κάποια σημασία. Με την σειρά τους, κάθε αντικείμενο της λίστας ορίζεται από μία ετικέτα που η αρχή και το τέλος της πρέπει να περικλείεται από την . Το στοιχείο είναι επιπέδου μπλοκ και μόνο στοιχεία επιτρέπεται να υπάρχουν μέσα σε αυτήν. Από προεπιλογή οι unordered list εμφανίζονται από τους browser σε εσοχή και με μία βούλα να βρίσκεται στα αριστερά κάθε .

**3.4.11 Ετικέτα **

Μία λίστα χρησιμοποιείται σε περιπτώσεις όπου η σειρά των στοιχείων μέσα σε αυτήν έχει σημασία. Με την σειρά τους, κάθε αντικείμενο της λίστας ορίζεται από μία ετικέτα που η αρχή και το τέλος της πρέπει να περικλείεται από την . Το στοιχείο είναι επιπέδου μπλοκ και μόνο στοιχεία επιτρέπεται να υπάρχουν μέσα σε αυτήν. Από προεπιλογή οι ordered list εμφανίζονται από τους browser σε εσοχή και με έναν αύξων αριθμό να βρίσκεται στα αριστερά κάθε .

**3.4.12 Ετικέτα **

Η ετικέτα αυτή δίνει έμφαση στο κείμενο το οποίο εμπεριέχεται σε αυτήν και αυτό εμφανίζεται σε έντονη γραφή.

**3.4.13 Ετικέτα
**

Η ετικέτα αυτή αναγκάζει τον browser να σταματήσει να τυπώνει στην τρέχουσα γραμμή και να συνεχίσει να τυπώνει στην αμέσως επόμενη. Το στοιχείο
 είναι ένα άδειο στοιχείο και αυτό σημαίνει ότι δεν έχει περιεχόμενο και κλείνει στην ίδια ετικέτα (
).

3.4.14 Ετικέτα <div>

Η ετικέτα <div> δημιουργεί μία λογική ενότητα στο έγγραφο η οποία ομαδοποιεί το συσχετιζόμενο με αυτήν περιεχόμενο. Είναι σημασιολογικά ουδέτερη και στην ουσία σημαίνει: “Ό,τι βρίσκεται μέσα σε αυτήν την περιοχή είναι μία ολότητα και είναι κάτι διαφορετικό από όλα τα υπόλοιπα στοιχεία.”.

Το στοιχείο <div> είναι εξαιρετικά χρήσιμο για την οργάνωση του περιεχομένου σε μεγάλα μπλοκ ώστε να μπορεί έπειτα να μορφοποιηθεί πιο εύκολα με τη χρήση CSS ή να χειριστεί με JavaScript ιδιαίτερα αν του έχει δοθεί μία ξεχωριστή ιδιότητα <id>. Ένα <div> είναι ένα στοιχείο επιπέδου μπλοκ, μπορεί να περιέχει κείμενο και οποιοδήποτε άλλο στοιχείο είτε αυτό είναι επιπέδου μπλοκ είτε ένθετο και το περιεχόμενό του ξεκινάει σε μία νέα γραμμή και καταλαμβάνει όλο το διαθέσιμο πλάτος.

**3.4.15 Ετικέτα **

Η ετικέτα αυτή είναι παρόμοια με την <div> με τη διαφορά ότι αυτή είναι ένθετη και χρησιμοποιείτε για να διαχωρίσει τμήματα κειμένου ώστε αυτά να μορφοποιηθούν διαφορετικά με την χρήση CSS.

**3.4.16 Ετικέτα **

Το ένθετο στοιχείο χρησιμοποιείται για να δηλώσει την εισαγωγή ενός αρχείου εικόνας στο HTML έγγραφο και απαιτεί τη χρήση της ιδιότητας src για να καθορίσει την θέση του αρχείου. Επίσης, απαιτείται η ιδιότητα alt η οποία παρέχει ένα εναλλακτικό κείμενο το οποίο προσδιορίζει την εικόνα. Το κείμενο αυτό εμφανίζεται στην περίπτωση που η εικόνα δεν είναι διαθέσιμη και δεν είναι δυνατό να εμφανιστεί στον browser.

Επίσης, χρησιμοποιείται για να αναγνωστεί από συσκευές ανάγνωσης οθόνης (screen readers) σε άτομα με προβλήματα όρασης.

3.4.17 Ετικέτα <a>

Η χρήση της ετικέτας <a> προκαλεί τον browser να μεταφερθεί σε μία διαφορετική τοποθεσία από την τρέχουσα όταν ο χρήστης κάνει κλικ πάνω στον σύνδεσμο που εμφανίζεται ανάμεσα στα <a> .

3.4.18 Ετικέτα <table>

Οι πίνακες χρησιμοποιούνται στις περιπτώσεις όπου υπάρχει ανάγκη για παρουσίαση δεδομένων όπου γίνονται πιο κατανοητά όταν εμφανίζονται σε σειρές και στήλες. Εδώ πρέπει να σημειωθεί ότι μέχρι πριν λίγα χρόνια οι πίνακες ήταν το κατεξοχήν εργαλείο στον σχεδιασμό και μορφοποίηση των ιστοσελίδων αλλά με την έλευση του CSS αυτό έχει περιοριστεί σε μεγάλο βαθμό. Ένας πίνακας έχει κεφαλές, όπου γίνεται η επεξήγηση της πληροφορίας των γραμμών και των στηλών καθώς και του κάθε κελιού. Οι ετικέτες <table> και </table> θα πρέπει να περιβάλουν τον ορισμό όλου του πίνακα. Το πρώτο στοιχείο εντός του πίνακα είναι η ετικέτα <caption> η χρήση του οποίου είναι προαιρετική. Έπειτα, με τη βοήθεια των ετικετών <tr> και </tr> μπορούν να οριστούν τόσες σειρές όσες απαιτούνται. Εντός μίας γραμμής μπορεί να οριστεί οποιοσδήποτε αριθμός κελιών που ορίζονται με τις ετικέτες <td> ... </td> και <th> ... </th>. Κάθε γραμμή ενός πίνακα μπορεί να μορφοποιηθεί ξεχωριστά και ανεξάρτητα από τις προηγούμενες και επόμενες γραμμές.

3.4.19 Ετικέτα <form>

Το στοιχείο <form> χρησιμοποιείται όπου απαιτείται η εισαγωγή δεδομένων από τον χρήστη. Όλη η λειτουργικότητα της φόρμας περικλείεται σε ένα και μόνο στοιχείο <form> το οποίο και στέλνει τα δεδομένα προς επεξεργασία.

Η φόρμα είναι στοιχείο επιπέδου μπλοκ και δεν μπορεί να έχει ένθετα περιεχόμενα. Στην περίπτωση που απαιτείται η εισαγωγή περισσότερων του ενός στοιχείου φόρμας σε ένα έγγραφο τότε κάθε ένα από αυτά πρέπει να είναι εντελώς ξεχωριστά - δεν επιτρέπεται να υπάρχει φόρμα μέσα σε μία άλλη φόρμα. Το στοιχείο φόρμα απαιτεί την ιδιότητα action της οποίας η τιμή είναι μία διεύθυνση URL. Με αυτόν τον τρόπο τα δεδομένα μπορούν να χρησιμοποιηθούν είτε από την πλευρά του πελάτη με κάποια γλώσσα όπως η JavaScript είτε από την πλευρά του εξυπηρετητή με κάποια γλώσσα όπως οι PHP, Ruby, ASP ή ASP.NET. Η ιδιότητα method είναι προαιρετική και μπορεί να δεχθεί μία από τις δύο τιμές get ή post για να υποδείξει τον τρόπο με τον οποίο θα υποβληθεί η φόρμα. Όταν η τιμή της ιδιότητας method είναι get, τα δεδομένα αποστέλλονται μέσω της URL με τη μορφή string το οποίο απαρτίζεται από ζεύγη ονόματος / τιμής για κάθε δεδομένο. Όταν η τιμή της ιδιότητας method είναι post, τα δεδομένα αποστέλλονται απευθείας στον χειριστή φόρμας της εφαρμογής για επεξεργασία στον εξυπηρετητή.

Η μέθοδος get χρησιμοποιείται συνήθως όταν απαιτούνται στατικά δεδομένα από τον εξυπηρετητή για προσωρινή χρήση ή όταν αυτά θα χρησιμοποιηθούν ξανά σύντομα. Πρέπει να σημειωθεί ότι η χρήση της μεθόδου get δεν πρέπει να γίνεται για ευαίσθητα δεδομένα καθώς αυτά είναι ορατά και από τρίτους.

Η μέθοδος post χρησιμοποιείται συνήθως όταν τα δεδομένα που αποστέλλονται πρέπει να αποθηκευτούν για μελλοντική χρήση ή όταν για λόγους ασφαλείας τα δεδομένα δεν πρέπει να είναι εμφανή.

Η προεπιλεγμένη μέθοδος είναι η get η οποία και υπονοείται όταν η ιδιότητα method παραλείπεται και δεν δηλώνεται καθόλου.

3.4.20 Ετικέτα <input>

Πολλά στοιχεία τα οποία ανήκουν σε φόρμα μπορούν να δημιουργηθούν με το ένθετο στοιχείο <input> και κάθε τέτοιο ορίζεται με την αντίστοιχη ιδιότητα type. Επειδή το στοιχείο <input> είναι ένθετο, πολλά τέτοια στοιχεία μπορούν να εμφανιστούν στην ίδια σειρά μέχρις ότου να καταληφθεί όλο το διαθέσιμο πλάτος, αλλά όλα πρέπει να ανήκουν σε έναν γονέα επιπέδου μπλοκ. Σημειώνεται επίσης ότι το στοιχείο <input> δεν έχει κείμενο δε μπορεί να περιέχει άλλα στοιχεία και πρέπει να κλείνει μόνο του (με />). Σημαντικές ιδιότητες του <input> είναι οι checked="checked", disabled="disabled", maxlength και type. Η πρώτη, όταν είναι δηλωμένη θέτει την αρχική τιμή για τα checkbox και τα radio buttons. Η δεύτερη απενεργοποιεί το αντίστοιχο στοιχείο ώστε να μη μπορεί να τροποποιηθεί. Η τιμή ενός απενεργοποιημένου στοιχείου δεν αποστέλλεται για επεξεργασία. Η ιδιότητα maxlength καθορίζει το μέγιστο αριθμό χαρακτήρων που μπορούν να εισαχθούν σε ένα πεδίο κειμένου. Η ιδιότητα αυτή υπάρχει μόνο για στοιχεία κειμένου και κωδικού (input type="text", input type="password"). Τέλος, η ιδιότητα type καθορίζει τον τύπο του αντίστοιχου στοιχείου της φόρμας. Ο προεπιλεγμένος τύπος είναι αυτός του κειμένου (text).

3.4.21 Ετικέτα <select>

Το ένθετο στοιχείο <select> δημιουργεί ένα πτυσσόμενο μενού επιλογών. Το μενού αυτό μπορεί να αναπτύσσεται είτε οριζόντια είτε κάθετα. Όταν αποστέλλεται η φόρμα τότε αποστέλλεται μόνο η τιμή που έχει επιλέξει ο χρήστης. Πρέπει να σημειωθεί ότι η ιδιότητα name είναι απαιτούμενη για το στοιχείο <select> έτσι ώστε να μπορεί να προσδιοριστεί. Τέλος, επειδή δε μπορεί να είναι άδειο, πρέπει να περιέχει τουλάχιστον μία τιμή.

3.4.22 Ετικέτα <option>

Κάθε επιλογή ενός στοιχείου <select> περιέχεται μέσα σε ένα στοιχείο <option>. Το στοιχείο αυτό δε μπορεί να είναι κενό αλλά μπορεί να περιέχει μόνο μία ετικέτα κειμένου (label). Μία επιλογή μπορεί να προεπιλεγεί συμπεριλαμβάνοντας την ιδιότητα selected="selected". Αν δεν έχει οριστεί η ιδιότητα value τότε η τιμή που θα αποσταλεί κατά την αποστολή της φόρμας είναι το κείμενο της επιλογής.

3.5 Επίλογος

Η HTML έχει αλλάξει ελάχιστα από τότε που δημιουργήθηκε. Αντίθετα, αυτό που έχει αλλάξει είναι η μορφοποίηση των στοιχείων της HTML κάτι που σήμερα γίνεται με την χρήση του CSS. Θα μπορούσε λοιπόν να ειπωθεί ότι τα στοιχεία της γλώσσας HTML χρησιμοποιούνται για την κατασκευή της δομής των ιστοσελίδων όμως αυτό που δίνει μορφή στις ιστοσελίδες αυτές είναι το CSS. Έτσι, η HTML θα μπορούσε πολύ εύστοχα να χαρακτηριστεί ως τα πινέλα με τα οποία δημιουργείται μία ιστοσελίδα ενώ το CSS ως τα χρώματα με τα οποία αυτή χρωματίζεται.

CSS

4.1 Εισαγωγή

Τα Φύλλα Διαμόρφωσης Στύλ (Cascading Style Sheets – CSS) είναι μια γλώσσα διαμόρφωσης ιστοσελίδων και χρησιμοποιείται για να περιγράψει την παρουσίαση ενός εγγράφου το οποίο έχει γραφτεί σε κάποια από τις γλώσσες σήμανσης (markup languages). Χρησιμοποιείται κυρίως σε εφαρμογές γραμμένες σε HTML και XHTML αλλά μπορεί να χρησιμοποιηθεί επίσης και σε οποιοδήποτε έγγραφο της XML για τη μορφοποίηση του.

4.2 Διαχωρισμός HTML και εμφάνισης

Το CSS έχει σχεδιαστεί ώστε να επιτρέπει τον διαχωρισμό του περιεχομένου του εγγράφου το οποίο συνήθως γράφεται σε HTML και της παρουσίασής του, η οποία περιλαμβάνει στοιχεία όπως τα χρώματα, οι γραμματοσειρές και η μορφή του. Ο διαχωρισμός αυτός μπορεί να βελτιώσει την λειτουργικότητα αλλά και την προσβασιμότητα παρέχοντας μεγαλύτερη ευελιξία και έλεγχο στον καθορισμό των χαρακτηριστικών εκείνων που άπτονται της μορφολογίας του εγγράφου, επιτρέποντας παράλληλα σε πολλά έγγραφα να μοιράζονται την ίδια μορφοποίηση με αποτέλεσμα τη μείωση της πολυπλοκότητας και της επανάληψης κώδικα. Επίσης η χρήση CSS επιτρέπει την παρουσίαση του ίδιου περιεχομένου με διαφορετικούς τρόπους όπως είναι η εμφάνιση σε οθόνη, η εκτύπωση, η αφήγηση καθώς και σε συσκευές ανάγνωσης βασισμένες στο σύστημα Braille.

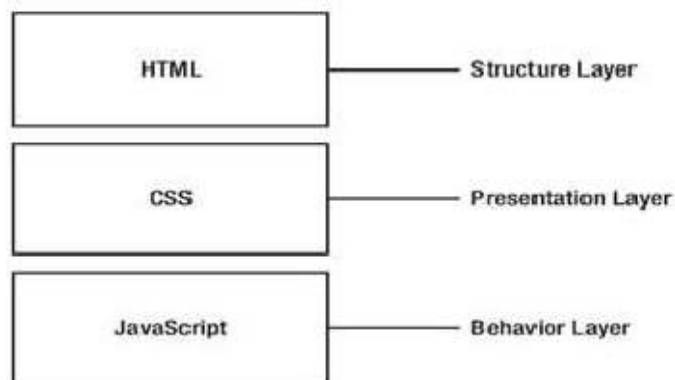
4.3 Δομή – Παρουσίαση – Συμπεριφορά

Τα τρία επίπεδα ενός Web εγγράφου είναι αυτά της Δομής, της Παρουσίασης και της Συμπεριφοράς (Structure Layer, Presentation Layer, Behavior Layer).

Το πρώτο επίπεδο, αυτό της Δομής, περιέχει το περιεχόμενο του εγγράφου μαζί με τις σημασιολογικές πληροφορίες. Είναι συνήθως γραμμένο στην γλώσσα HTML.

Το επίπεδο της παρουσίασης περιγράφει τον τρόπο με τον οποίο το έγγραφο πρέπει να παρουσιαστεί στον επισκέπτη της ιστοσελίδας, συμπεριλαμβάνοντας πληροφορίες όπως τα χρώματα, τις γραμματοσειρές, την μορφοποίηση των εικόνων κ.ά. Γενικά το επίπεδο της Παρουσίασης ενός εγγράφου Web, είναι γραμμένο με χρήση CSS.

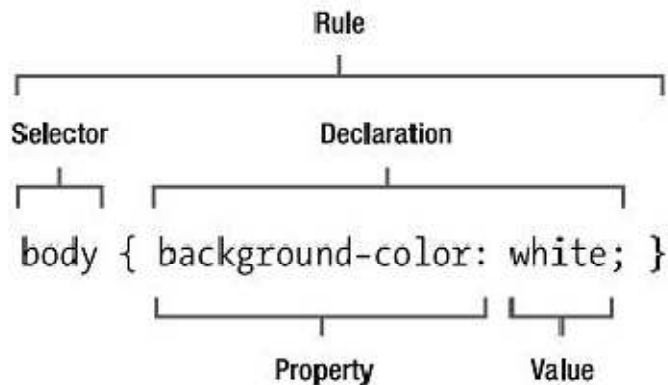
Τέλος το επίπεδο Συμπεριφοράς χρησιμοποιείται για την ανανέωση, πρόσθεση και αφαίρεση ενός Web εγγράφου σε συνάρτηση πάντα με τη συμπεριφορά του χρήστη. Το επίπεδο αυτό άπτεται της διαχείρισης του Document Object Model (DOM), συνήθως είναι γραμμένο στη γλώσσα JavaScript. Το Σχήμα 1 παρουσιάζει τα τρία αυτά επίπεδα.



Σχήμα I: Τα τρία επίπεδα ενός Web εγγράφου

4.4 Βασικές Αρχές του CSS

Το CSS επιτρέπει την προσθήκη στυλ στις ιστοσελίδες, ενσωματώνοντας και βελτιώνοντας την μορφή στο περιεχόμενο. Κάθε HTML στοιχείο προσδίδει κάποιο περιεχόμενο και οι ιδιότητες οι οποίες δηλώνονται με κάθε ένα από αυτά παρέχουν περισσότερη πληροφορία σε αυτά τα στοιχεία. Το δομικό στοιχείο του CSS είναι ο κανόνας (rule) ο οποίος είναι ένα σύνολο από εντολές τις οποίες ο browser ακολουθεί ώστε να αλλάξει την προεπιλεγμένη παρουσίαση των HTML στοιχείων ανάλογα με τις τιμές που έχουν οριστεί. Ένας CSS κανόνας αποτελείται από τα στοιχεία που απεικονίζονται στο Σχήμα II:



Σχήμα II: Τα στοιχεία ενός CSS rule

Ο επιλογέας (selector) είναι το κομμάτι αυτό του κανόνα το οποίο συγκεκριμενοποιεί το στοιχείο το οποίο πρόκειται να μορφοποιηθεί. Η σκοπιά του μπορεί να είναι μεγάλου εύρους, επηρεάζοντας τη μορφή κάθε στοιχείου, ή ακόμα και πολύ μικρή και συγκεκριμένη, επηρεάζοντας για παράδειγμα μόνο ένα στοιχείο. Κάθε δήλωση (declaration) αποτελείται από δύο επιμέρους στοιχεία τα οποία ακολουθούν συγκεκριμένη σειρά και είναι η ιδιότητα (property) και η τιμή (value). Η ιδιότητα είναι το τμήμα εκείνο του στοιχείου το οποίο πρόκειται να μορφοποιηθεί, όπως είναι το χρώμα, το πλάτος, το ύψος, η γραμματοσειρά ή ακόμα και η θέση όπου πρόκειται να εμφανιστεί το στοιχείο.

Η τιμή της ιδιότητας (property value) προσδίδει το συγκεκριμένο στυλ το οποίο πρόκειται να εφαρμοστεί στο επιλεγμένο στοιχείο. Οι τιμές οι οποίες είναι αποδεκτές εξαρτώνται από την κάθε ιδιότητα, και μερικές ιδιότητες δέχονται πολλαπλές τιμές οι οποίες χωρίζονται με κόμμα.

Κάθε δήλωση γίνεται ανάμεσα σε ένα ζεύγος άγκιστρων ({ και }) και υπάρχει η δυνατότητα να εφαρμοστούν πολλές δηλώσεις σε ένα συγκεκριμένο selector η κάθε μία από τις οποίες μπορεί να μορφοποιεί ξεχωριστά σημεία κάθε στοιχείου ή ακόμα και όλα εξαρχής. Κάθε ιδιότητα χωρίζεται από την τιμή της με μία άνω κάτω τελεία (:) και η δήλωση τελειώνει με ένα ελληνικό ερωτηματικό (;). Το ερωτηματικό αυτό είναι σημαντικό για το διαχωρισμό κάθε δήλωσης αλλά αν υπάρχει μόνο μία δήλωση στον κανόνα ή στην περίπτωση της τελευταίας δήλωσης τότε το ερωτηματικό είναι προαιρετικό. Καλό είναι πάντως αυτά να τοποθετούνται κάθε φορά.

4.5 CSS Selectors

Ένας selector, όπως φανερώνει και το όνομά του, επιλέγει ένα στοιχείο στο HTML έγγραφο. Οι διαθέσιμοι selector είναι σχετικά λίγοι και κάθε ένας τους έχει διαφορετικό βαθμό διάκρισης των στοιχείων που τον ακολουθούν. Διάκριση είναι ο τρόπος μέτρησης της σκοπιάς ενός selector, δηλαδή πόσα στοιχεία θα επιλεγούν από αυτόν. Το CSS είναι σχεδιασμένο έτσι ώστε ο πιο συγκεκριμένος γίνεται ένας selector να υπερβαίνει τον πιο γενικό. Η δυνατότητα αυτή της διάκρισης είναι ένα από τα πιο ισχυρά στοιχεία του CSS. Παρακάτω παρατίθενται με περισσότερες λεπτομέρειες, ορισμένοι από τους βασικότερους CSS Selectors.

4.5.1 Παγκόσμιος Επιλογέας (Universal Selector)

Ο universal selector εκφράζεται με το σημείο του αστερίσκου (*) και λειτουργεί ως wild card για την επιλογή κάθε στοιχείου στο έγγραφο. Για παράδειγμα, ο κανόνας:

```
* { color: blue; }
```

θα προσδώσει ένα μπλε υπόβαθρο σε όλα τα στοιχεία. Επικεφαλίδες, παράγραφοι, λίστες, κελιά πίνακα ακόμα και σύνδεσμοι θα μορφοποιηθούν με μπλε χρώμα. Αυτός είναι και ο λιγότερο συγκεκριμένος selector αφού στην ουσία επιλέγει όλα τα στοιχεία.

4.5.2 Επιλογέας Στοιχείου (Element Selector)

Ένας element selector επιλέγει όλα τα στιγμιότυπα ενός στοιχείου τα οποία καθορίζονται από το όνομα της ετικέτας τους. Ο selector αυτός χαρακτηρίζεται από μεγαλύτερη διακρίσιμότητα από ότι ο universal selector αλλά δεν είναι και πολύ συγκεκριμένος, αφού επιλέγει κάθε στοιχείο στο έγγραφο που έχει το ίδιο όνομα ετικέτας, όσες φορές κι αν εμφανιστεί αυτό. Για παράδειγμα, ο κανόνας:

```
em { color: red; }
```

προσδίνει σε κάθε στοιχείο με την ετικέτα em στο ίδιο κόκκινο χρώμα υποβάθρου ακόμα κι αν υπάρχουν χιλιάδες τέτοια στοιχεία σε όλο το έγγραφο. Πρέπει να σημειωθεί ότι οι element selectors ονομάζονται και type selectors.

4.5.3 Επιλογέας Κλάσης (Class Selector)

Ένας class selector στοχεύει σε κάθε στοιχείο το οποίο έχει μία συγκεκριμένη κλάση η οποία του έχει δοθεί πιο πριν με την χρήση της ιδιότητας class. Αφού η ιδιότητα αυτή μπορεί να δοθεί σε πολλά στοιχεία ο selector αυτός δεν είναι πολύ συγκεκριμένος αλλά καθορίζει τη διακρίτοτητα σε μεγαλύτερο βαθμό από ότι ο element selector.

Στο CSS, οι class selectors αναγνωρίζονται από την τελεία (.) που προηγείται πριν από αυτούς. Για παράδειγμα, ο παρακάτω κανόνας θα μορφοποιήσει κάθε στοιχείο που ανήκει στην κλάση "info":

```
.info {color: blue ;}
```

4.5.4 Επιλογέας Ταυτότητας (Id Selector)

Ένας id selector επιλέγει μόνο το στοιχείο που φέρει το αντίστοιχο id.

Συγκεκριμένα κάθε στοιχείο μπορεί να έχει κάποιο id αλλά μόνο ένα id του ίδιου ονόματος μπορεί να υπάρχει σε όλο το έγγραφο. Ο id selector στοχεύει σε ένα μόνο στοιχείο σε κάθε σελίδα και αυτό τον κάνει πολύ πιο συγκεκριμένο από τον class selector.

Στο CSS, οι id selectors αναγνωρίζονται από τον χαρακτήρα κάγκελο (#) που προηγείται πριν από αυτούς. Για παράδειγμα, ο παρακάτω κανόνας θα μορφοποιήσει το στοιχείο με id "introduction" και θα του δώσει ως χρώμα υποβάθρου το πράσινο:

```
#introduction {color: green; }
```

4.5.5 Επιλογέας Ψευδοκλάσης (Pseudo Class Selector)

Μία ψευδοκλάση είναι περίπου το ίδιο με τον επιλογέα κλάσης και έχει τον ίδιο βαθμό διακρίτοτητας με την κλάση αλλά επιλέγει μία συγκεκριμένη κατάσταση ενός στοιχείου. Πριν από μία ψευδοκλάση προηγείται μία άνω κάτω τελεία (:) και μόνο μερικές ψευδοκλάσεις είναι διαθέσιμες:

```
: link { color: blue; }  
: visited {color: purple ;}  
: active { color: red; }  
: hover {color: green ;}  
: focus { color: orange; }
```

Η ψευδοκλάση :link επιλέγει όλα τα στοιχεία τα οποία είναι σύνδεσμοι. Η ψευδοκλάση :visited επιλέγει τους συνδέσμους εκείνους οι οποίοι ο προορισμός που περιγράφουν έχει προηγουμένως επισκεφτεί. Η ψευδοκλάση :active επιλέγει εκείνους τους συνδέσμους οι οποίοι είναι σε ενεργή κατάσταση. Η ψευδοκλάση :hover επιλέγει όλα τα στοιχεία εκείνα στα οποία η θέση του mouse είναι από πάνω τους. Η ψευδοκλάση :focus επιλέγει κάθε στοιχείο σε κατάσταση προεπιλογής.

4.5.6 Επιλογέας απογόνου (Descendant Selector)

Ένας από τους πιο χρήσιμους και ισχυρούς selector είναι ο descendant selector ο οποίος συντελείται από δύο ή περισσότερους βασικούς selector (universal, element, class, id, pseudo class) χωρισμένους με κενά. Αυτό που κάνει είναι να επιλέγει τη θέση του συγκεκριμένου στοιχείου στο έγγραφο. Για παράδειγμα, ο selector:

```
#introduction em { color: yellow; }
```

θα προσδώσει το κίτρινο χρώμα σε οποιοδήποτε στοιχείο με ετικέτα em το οποίο υπάρχει μέσα στο στοιχείο με το id “introduction”.

Οι descendant selectors επιτρέπουν εξαιρετικά συγκεκριμένες επιλογές, όπως στο παρακάτω παράδειγμα:

```
#introduction .info p * {color: pink ;}
```

που θα επιλέξει όλα τα στοιχεία τα οποία είναι απόγονοι ενός στοιχείου p, το οποίο είναι απόγονος ενός στοιχείου που είναι της κλάσης info, το οποίο είναι απόγονος ενός στοιχείου που έχει το id introduction.

4.5.7 Συνδυασμός Επιλογέων (Combining Selectors)

Είναι εφικτός ο συνδυασμός δύο ή περισσότερων selector, όπως είναι ενός στοιχείου και ενός id ή ενός id και μίας κλάσης. Για παράδειγμα, ο παρακάτω κανόνας:

```
p.info { color: blue; }
```

θα επιλέξει μόνο τις παραγράφους (στοιχεία p) οι οποίες ανήκουν στην κλάση info. Όλα τα υπόλοιπα στοιχεία αυτής της κλάσης θα παραλειφθούν και όλες οι υπόλοιπες παράγραφοι οι οποίες δεν ανήκουν στην κλάση info θα μείνουν ανέπαφες. Χαρακτηριστικό παράδειγμα του πόσο ισχυροί μπορούν να γίνουν οι συνδυασμοί selector είναι το παρακάτω:

```
P#introduction a.info: hover {color: silver ;}
```

Ο κανόνας αυτός θα εφαρμοστεί μόνο σε hovered συνδέσμους (στοιχεία a), οι οποίοι ανήκουν στην κλάση info και οι οποίοι είναι απόγονοι της παραγράφου με id introduction.

4.6 Επισύναψη Style Sheet στα έγγραφα

Για την επισύναψη ενός style sheet σε κάποια σελίδα, πρέπει να γίνει η σύνδεση του εγγράφου με το συγκεκριμένο style sheet. Όταν ο browser κάνει download το HTML έγγραφο, τότε αυτόματα θα αναζητήσει τους CSS κανόνες που συμπεριλαμβάνουν τις οδηγίες σχετικά με τη μορφή που θα έχει το έγγραφο. Υπάρχουν διάφοροι τρόποι για να εισαχθούν αυτοί οι κανόνες οι οποίοι περιγράφονται παρακάτω.

4.6.1 Ένθετα Στυλ (Inline Style)

Οι δηλώσεις του CSS μπορούν να συμπεριληφθούν με τη χρήση της ιδιότητας style σε κάθε στοιχείο ξεχωριστά. Τα inline styles δεν γράφονται σαν κανόνες και δεν υπάρχει επιλογέας επειδή οι ιδιότητες και οι τιμές επισυνάπτονται απευθείας με το κάθε στοιχείο. Τα inline styles είναι τα πιο συγκεκριμένα από όλα επειδή εφαρμόζονται σε ακριβώς ένα στοιχείο και σε κανένα άλλο.

Παρόλα αυτά πρέπει να αποφεύγεται η χρήση ένθετων στυλ επειδή η μίξη της δομής με την παρουσίαση ακυρώνουν πολλά από τα πλεονεκτήματα που παρέχει η χρήση του CSS. Παράδειγμα ενός τέτοιου στυλ είναι το παρακάτω:

```
<p style="color: blue">text</p>
```

Με την χρήση αυτού του style η συγκεκριμένη παράγραφος θα μορφοποιηθεί με μπλε χρώμα. Τέτοιες μορφοποιήσεις μπορούμε να κάνουμε σχεδόν σε κάθε HTML στοιχείο.

4.6.2 Ενσωματωμένα Στυλ (Embedded Style Sheets)

Τα ενσωματωμένα στυλ δηλώνονται στο στοιχείο <head> του εγγράφου και οι κανόνες που εμπεριέχονται σε αυτό ακολουθούνται μόνο από το συγκεκριμένο έγγραφο. Ένα embedded style sheet περιέχεται μέσα στο στοιχείο style και διαχωρίζεται σε μεγάλο βαθμό τη δομή από την παρουσίαση του εγγράφου. Παράδειγμα ενός τέτοιου στυλ είναι το παρακάτω:

```
<html>
<head><title>CSS Example</title>
<style type="text /css">
p { color: green; }
a { color: blue; }
</style>
</head>
</html>
```

Το κομμάτι κώδικα αυτό θα κάνει όλες τις παραγράφους της HTML σελίδας πράσινες και όλους τους συνδέσμους της σελίδας μπλε. Όμως το στυλ αυτό έχει το μειονέκτημα ότι μπορεί να εφαρμοστεί μόνο σε μία σελίδα κι έτσι δεν είναι επαναχρησιμοποιήσιμο κάτι που οδηγεί στην επανάληψη του ίδιου κώδικα.

4.6.3 Εξωτερικά Φύλλα Στυλ (External Style Sheet)

Η τελευταία και καλύτερη επιλογή συγγραφής των CSS κανόνων είναι σε ένα ξεχωριστό, εξωτερικό αρχείου το οποίο συνδέεται απευθείας με τα έγγραφα όπου αυτό απαιτείται. Ένα external style sheet είναι ένα απλό αρχείο κειμένου το οποίο αποθηκεύεται με κατάληψη .css. Αυτή η προσέγγιση διαχωρίζει πλήρως την παρουσίαση από το περιεχόμενο, αφού αυτά δεν βρίσκονται στο ίδιο αρχείο.

Επίσης, ένα τέτοιο αρχείο μπορεί να συνδεθεί με πολλά HTML έγγραφα επιτρέποντας έτσι σε πλήθος ιστοσελίδων να ελέγχονται από ένα και μόνο κεντρικό αρχείο. Οι αλλαγές σε αυτό το αρχείο θα εφαρμοστούν προοδευτικά σε όλες τις σελίδες, οι οποίες είναι συνδεδεμένες με αυτό. Κάτι τέτοιο είναι πολύ ευέλικτο και δείχνει το πόσο ισχυρό και χρήσιμο είναι το CSS.

Ένα HTML έγγραφο συνδέεται με ένα external style sheet μέσω ενός συνδέσμου στο στοιχείο <head> του εγγράφου. Το παρακάτω τμήμα κώδικα δείχνει πως επιτυγχάνεται αυτό:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en"
xml:lang="en">
<head>
<title>Spaghetti and Cruft : Our Menu</title>
<link rel="stylesheet" type="text/css" href="styles.css" />
</head>
<body>
<h2>This is managable via a external stylesheet</h2>
<p>me too!>p>
</body>
</html>
```

4.7 Επίλογος

Όπως γράφτηκε πιο πάνω, τα επίπεδα του περιεχομένου και της παρουσίασης, τα οποία σε προγραμματιστικό επίπεδο μεταφράζονται σε HTML και CSS αντίστοιχα, είναι δύο από τα τρία μέρη από τα οποία αποτελείται μία σύγχρονη ιστοσελίδα. Το τρίτο μέρος είναι το επίπεδο της συμπεριφοράς το οποίο αναλύεται συνοπτικά στο επόμενο κεφάλαιο.

JAVASCRIPT

5.1 Εισαγωγή

Με την πάροδο του χρόνου, όσο περισσότερες ιστοσελίδες δημιουργούσαν οι σχεδιαστές και οι προγραμματιστές, τόσο μεγάλωνε η απαίτηση για δυναμικές ιστοσελίδες. Ένα από τα προβλήματα όμως που είχαν να αντιμετωπίσουν ήταν ο πόλεμος των browser που είχε σαν αποτέλεσμα την ασυμβατότητα των προγραμμάτων ανάμεσα σε browser διαφορετικών εταιρειών. Μέχρι πριν λίγο καιρό δεν ήταν εύκολο να δημιουργηθούν εφαρμογές JavaScript οι οποίες να εκτελούνταν παντού το ίδιο. Αυτό όμως άρχισε να αλλάζει με την χρήση της τεχνολογίας DOM.

5.2 JavaScript

Η γλώσσα προγραμματισμού JavaScript αναπτύχθηκε από την εταιρεία Netscape, σε συνεργασία με την Sun Microsystems και η πρώτη της έκδοση δημοσιεύτηκε το 1995. Ακολούθησε η αντίστοιχη γλώσσα της Microsoft η οποία ονομάστηκε Jscript και η επόμενη έκδοση της JavaScript που είχε το όνομα ECMAScript που αργότερα όμως καθιερώθηκε με το όνομα που είναι γνωστό μέχρι σήμερα. Η JavaScript είναι μία διερμηνευμένη (interpreted) γλώσσα προγραμματισμού με ιδιότητες αντικειμενοστραφούς γλώσσας προγραμματισμού, χωρίς όμως να μπορεί να χαρακτηριστεί ως πλήρης αντικειμενοστραφής. Η γλώσσα αυτή, χτίστηκε ουσιαστικά πάνω στο πρότυπο των γλωσσών C, C++ και Java. Από την άλλη όμως έχει μία πολύ σημαντική διαφορά, στο ότι διαχειρίζεται τους τύπους δεδομένων πιο χαλαρά (loosely typed) σε σχέση με τη σφικτή διαχείριση τύπων δεδομένων (strongly typed) που γίνεται στις προαναφερόμενες γλώσσες. Στην JavaScript οι μεταβλητές δεν είναι απαραίτητο να έχουν ένα συγκεκριμένο τύπο ή ακόμη είναι δυνατόν να αλλάζουν τύπο κατά τη διάρκεια της ζωής τους. Επίσης, δεν πρέπει να συγχέεται η JavaScript με την Java της Sun Microsystems. Η μία δεν έχει καμία σχέση με την άλλη. Η χρήση του ονόματος JavaScript έγινε για λόγους προώθησης της γλώσσας σε μία εποχή που η εξάπλωση της Java ήταν πολύ μεγάλη. Στον πίνακα 1 βλέπουμε τις βασικές διαφορές JavaScript και Java. (Salabasis, 2008) Ευκολα διαπιστώνει κανείς ότι οι 2 γλώσσες δεν έχουν καμία σχέση.

Πίνακας 1: Σύγκριση JavaScript και Java

JavaScript	Java
Ερμηνευμένη(Interpreted)απο τον client	Μεταγλωτισμένη(Compiled) απο τον προγραμματιστή,τρέχει στον client
Κώδικας ενσωματωμένος σε HTML σελίδες	Τα java Applets δεν είναι ενσωματωμένα σε HTML κείμενα
Χαλαροί τύποι δεδομένων(loose typing of data types)	Σφικτοί τύποι δεδομένων(strong typing of data types)
Δυναμική διασύνδεση(dynamic binding)	Στατική διασύνδεση(Static binding)
Περιορισμός σε λειτουργίες του Client browser	Εφαρμογές αυτόνομες και πλήρεις

JavaScript	Java
Λειτουργεί με τα HTML αντικείμενα (HTML elements)	Ξεπερνάει τον περιορισμένο ορίζοντα της HTML(παράδειγμα multimedia)
Προσπελαύνει αντικείμενα του browser και τη λειτουργικότητα του	Δεν έχει προσπέλαση στα αντικείμενα του browser και τις λειτουργίες του

5.3 Χρήσεις της JavaScript

Η γλώσσα JavaScript χρησιμοποιείται κυρίως για την εξυπηρέτηση των παρακάτω σκοπών:

- Λιγότερος φόρτος των server: Ο έλεγχος και η επικύρωση των δεδομένων που εισάγονται από τους χρήστες γίνεται από τη μεριά του browser κι έτσι δεδομένα τα οποία δεν είναι σε κατάλληλη μορφή δεν αποστέλλονται στον server. Αυτό όμως δεν σημαίνει ότι ο έλεγχος δεν πρέπει να γίνεται και στη μεριά των εξυπηρετητών, καθώς κάποιος χρήστης μπορεί να μην έχει ενσωματωμένη την JavaScript στον browser του ή υπάρχει πιθανότητα να την έχει απενεργοποιήσει.
- Άμεση αλληλεπίδραση με τους χρήστες: Με την χρήση της JavaScript για τον έλεγχο των δεδομένων μειώνονται οι χρόνοι αναμονής του χρηστών αφού αυτοί δεν χρειάζεται να περιμένουν μεγάλα χρονικά διαστήματα επαναφόρτωσης της σελίδας σε περίπτωση που έχουν ξεχάσει να εισάγουν κάποιο δεδομένο ή έχουν εισάγει κάτι λάθος.
- Αυτόματη διόρθωση λαθών: Ένα παράδειγμα που μπορεί να κάνει περισσότερο κατανοητό το πώς μπορεί να χρησιμοποιηθεί η JavaScript με αυτόν τον τρόπο είναι αυτό της ημερομηνίας. Πολλά συστήματα βάσεων δεδομένων αποθηκεύουν δεδομένα ημερομηνιών σε μορφή dd-mm-yyyy. Αν κάποιος χρήστης εισάγει κάποια ημερομηνία σε μορφή dd/mm/yyyy τότε κάτι τέτοιο θα μπορούσε να ανιχνευτεί αυτόματα από τον browser και να μετατραπεί στην σωστή μορφή πριν τα δεδομένα αποσταλούν στον server.
- Αυξημένη χρηστικότητα: Αυτό επιτυγχάνεται επιτρέποντας στον χρήστη, την αλλαγή και αλληλεπίδραση με το γραφικό περιβάλλον χωρίς την επαναφόρτωση της σελίδας. Ένα τέτοιο παράδειγμα είναι τα πτυσσόμενα μενού.
- Αυξημένη δυνατότητα αλληλεπίδρασης: Ένα τέτοιο παράδειγμα όπου κάτι τέτοιο επιτυγχάνεται είναι τα μενού τα οποία αλληλεπιδρούν όταν ο χρήστης περάσει το mouse πάνω από αυτά - η λειτουργία hover - κάτι το οποίο έχει ως αποτέλεσμα να δημιουργηθεί μία σειρά από γεγονότα τα οποία έχουν προγραμματιστεί να λειτουργούν με έναν συγκεκριμένο τρόπο.
- Καλύτερα γραφικά περιβάλλοντα: Χρησιμοποιώντας την JavaScript μπορούν να συμπεριληφθούν αντικείμενα με λειτουργίες drag-and-drop καθώς και plug-ins, όπως είναι το Flash.
- Ελαφρότερα περιβάλλοντα: Αντί της απαίτησης download ενός μεγάλου αρχείου Java applet ή ενός Flash movie, τα προγράμματα γραμμένα σε JavaScript είναι μικρά σε μέγεθος και αποθηκεύονται στη μνήμη του browser μόλις κατέβουν.

5.4 Τοποθέτηση scripts σε ξεχωριστά αρχεία και ενσωμάτωση

Εκτός απο τη δυνατότητα να γράφουμε τα scripts κατευθείαν μέσα στην HTML σελίδα, υπάρχει η δυνατότητα να γράφουμε script κώδικα σε ξεχωριστό αρχείο και να το ενσωματώσουμε μετέπειτα σε οποιοδήποτε HTML αρχείο επιθυμούμε. Η ενσωμάτωση κώδικα μπορεί να γίνει με τη χρήση της ιδιότητας SRC όπως φαίνεται παρακάτω.

```
<HTML>
<HEAD>
<TITLE>EXAMPLE 2</TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript" SRC="myscript.js">
</SCRIPT>
</BODY>
</HTML>
```

Ο script κώδικας βρίσκεται στο αρχείο myscript.js

Η ενσωμάτωση script είναι μια καλή πρακτική για διάφορους λόγους:

- ✓ Μας δίνει τη δυνατότητα να οργανώσουμε τον κώδικά μας καλύτερα.
- ✓ Εάν ο κώδικας αποθηκεύεται τοπικά απο τον browser τότε αν γίνεται ενσωμάτωση του script σε πολλά HTML αρχεία τότε το script «κατεβαίνει» μια φορά και μετά ανακτάται απο την cache μνήμη του browser. Αυτό δεν θα συνέβαινε εάν το script ήταν ενσωματωμένο σε κάθε HTML σελίδα και «κατέβαινε» εξ'ολοκλήρου μαζί με την κάθε σελίδα.
- ✓ Επίσης εάν θέλουμε να κάνουμε cross-browser ανάπτυξη να καλύπτουμε δηλαδή διάφορους browser συνήθως θα έχουμε δύο εκδόσεις του script και θα κατεβάζουμε αυτόν που θέλουμε ανάλογα με τον browser.

5.5 Inline scripts

Εκτός απο την ενσωμάτωση απο ξεχωριστά αρχεία και την τοποθέτηση κώδικα με την βοήθεια της σήμανσης <script> υπάρχει και η δυνατότητα άμεσης γραφής JavaScript κώδικα μέσα στον ορισμό κάποιου γεγονότος εάν ο κώδικας είναι μικρός. Παρακάτω βλέπουμε ένα παράδειγμα:

```
<HTML>
<HEAD>
<TITLE>EXAMPLE 3-Inline script</TITLE>
</HEAD>
<BODY>
<IMG ID="lmg" SRC="old.gif" onclick="lmg.src='new.gif'">
</BODY>
</HTML>
```

Ο τρόπος αυτός προτείνεται μόνο για μικρά scripts καθώς δεν είναι ευανάγνωστος.

5.6 Επίλογος

Η παρουσίαση της συμπεριφοράς ενός web εγγράφου καθορίζεται με την γλώσσα javascript.Τι γίνεται όμως στην περίπτωση που θέλουμε το περιεχόμενο μιας ιστοσελίδας να ανανεώνεται δυναμικά χωρίς να είναι απαραίτητη η ανανέωση ολόκληρης της σελίδας;Στο επόμενο κεφάλαιο παρουσιάζεται η τεχνολογία AJAX η οποία λύνει αυτό το πρόβλημα.

AJAX

6.1 Εισαγωγή

Ο όρος AJAX προέρχεται απο το Asynchronous JavaScript and XML. Ο όρος Ajax επινοήθηκε το 2005. Ο Jesse James Garrett σκέφτηκε τον όρο "Ajax", όταν συνειδητοποίησε την ανάγκη ενός σύντομου όρου για να παρουσιάζει την ακολουθία των τεχνολογιών που πρότεινε σε έναν client. Στις 5 Απριλίου του 2006, η W3C κυκλοφόρησε το πρώτο προσχέδιο για το αντικείμενο σε μια προσπάθεια να δημιουργήσει ένα επίσημο web πρότυπο.

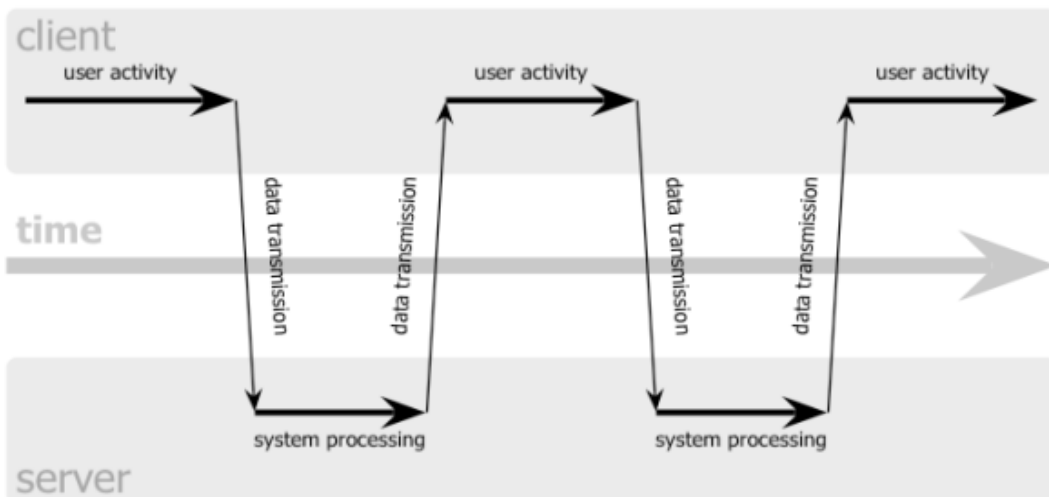
6.2 Τι είναι η τεχνολογία AJAX

Η τεχνολογία AJAX είναι ένας συνδυασμός υπαρχουσων τεχνολογιών που βοηθά κάποιον ευκολότερα να δημιουργήσει διαδραστικούς ιστοχώρους. Όταν αναπτύσσετε έναν ιστοχώρο χρησιμοποιώντας τεχνολογία AJAX, χρησιμοποιείτε τεχνολογίες όπως η HTML και τα CSS για την μορφοποίηση, JavaScript για να αλληλεπιδράσετε με το DOM απο την πλευρά του πελάτη, ένα νέο αντικείμενο XMLHttpRequest για να παρέχετε έναν αγωγό ροής δεδομένων μεταξύ του πελάτη και του κεντρικού υπολογιστή και τέλος την XML ως πρωτόκολλο μεταφοράς δεδομένων. Πλέον, υπάρχουν εφαρμογές οι οποίες είναι υλοποιημένες και βασίζονται πλήρως σε AJAX και είναι γεγονός ότι είναι πολύ περισσότερο φιλικές προς τον χρήστη και έχουν πάρα πολύ μεγάλο βαθμό επισκεψιμότητας. Ένα τέτοιο παράδειγμα είναι το Flickr (www.flickr.com). Ο λόγος για τον οποίο οι εφαρμογές βασισμένες σε AJAX είναι τόσο δημοφιλείς δεν είναι ότι οι λειτουργίες που παρέχει είναι ρηξικέλευθα καινοτομικές, αλλά το γεγονός ότι ο τρόπος που παρουσιάζονται οι πληροφορίες μέσα απο αυτές είναι πολύ αποτελεσματικός και εργονομικός.

6.3 Τρόπος λειτουργίας

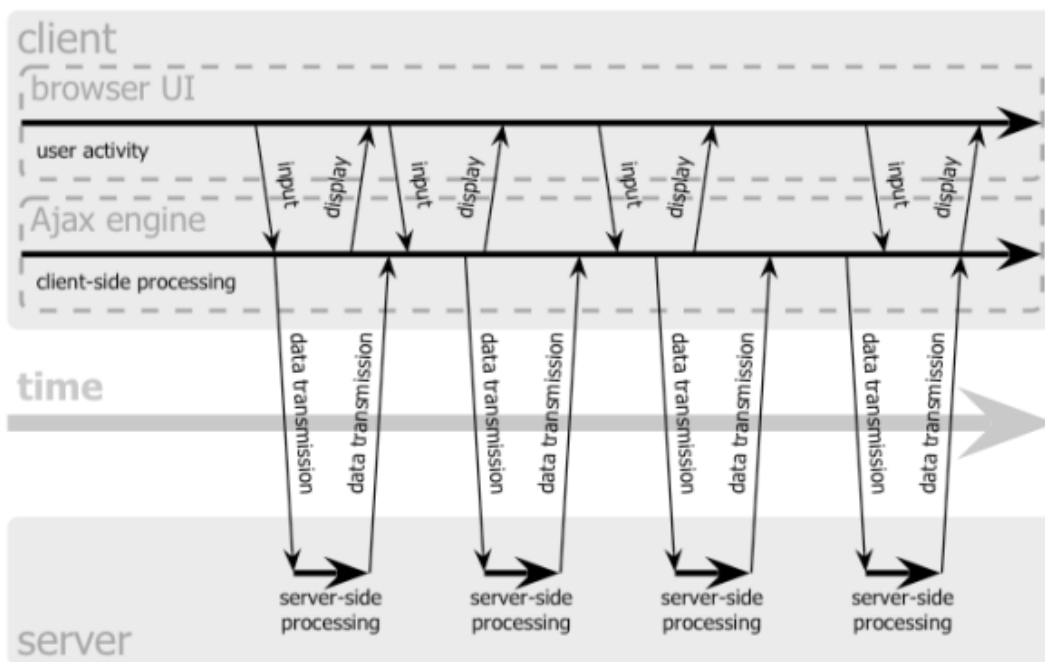
Βασικά, αυτό που κάνει το AJAX είναι να χρησιμοποιεί το αντικείμενο XMLHttpRequest της JavaScript το οποίο στέλνει ασύγχρονα τις αιτήσεις στον server χωρίς να απαιτείται η ανανέωση της σελίδας. Το σχήμα I και το σχήμα II παρουσιάζουν τις διαφορές ανάμεσα στο παραδοσιακό μοντέλο και σε αυτό που βασίζεται στο AJAX.

classic web application model (synchronous)



Σχήμα I: Ο παραδοσιακός τρόπος αίτησης στον server

Ajax web application model (asynchronous)



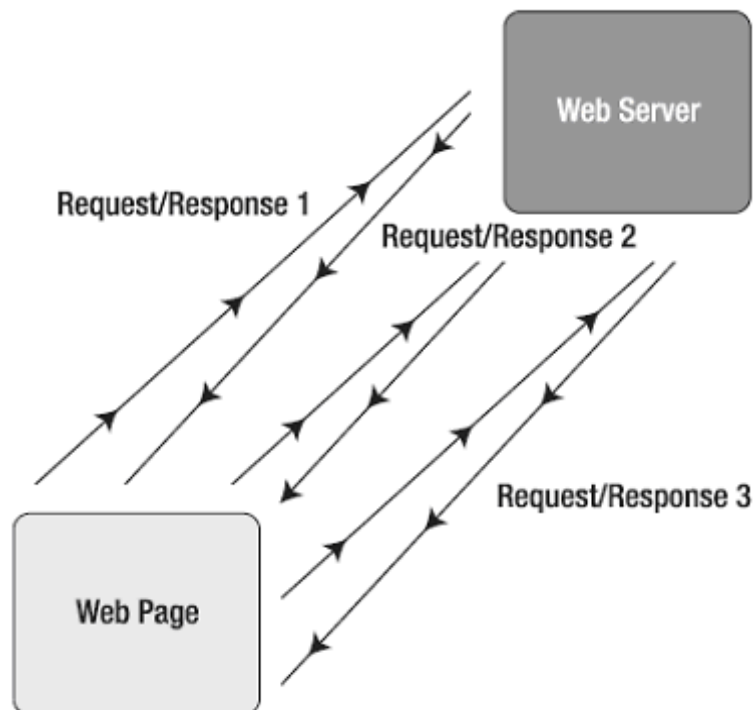
Σχήμα II: Ο βασισμένος σε Ajax τρόπος αίτησης στον server

Συνοπτικά το AJAX λειτουργεί ως εξής:

1. Στον browser γράφεται κώδικας σε JavaScript με τον οποίο και γίνεται η ανάκτηση των δεδομένων από τον server όποτε αυτό απαιτείται.
2. Σε περίπτωση που απαιτούνται από τον server περισσότερα δεδομένα, η JavaScript χρησιμοποιεί ένα ειδικό αντικείμενο (που υποστηρίζεται από τους browsers), το XMLHttpRequest object, για να στείλει παρασκησιακά

μία αίτηση (Request) στο server χωρίς να απαιτηθεί ή να προκληθεί ανανέωση της σελίδας. Αντιθέτως, η JavaScript δεν χρειάζεται να σταματήσει οποιαδήποτε άλλη εργασία που εκτελείται στον browser όσο περιμένει τα δεδομένα να σταλούν απο τον server, αφού μπορεί να περιμένει τα δεδομένα στο παρασκήνιο και να εκτελέσει κάποια λειτουργία όταν τα δεδομένα φτάσουν (για αυτόν τον λόγο λέγεται ασύγχρονη- asynhronous ανάκτηση δεδομένων).

3. Τα δεδομένα που στέλνονται απο τον server μπορούν να είναι σε μορφή XML ή ακόμα και σε απλό text. Ο JavaScript κώδικας στον browser μπορεί να διαβάσει δεδομένα, να τα επεξεργαστεί ή και να τα εμφανίσει αμέσως. Για παράδειγμα μία υποθετική εφαρμογή για τον υπολογισμό δανείων περιέχει μια φόρμα όπου εισάγεται το αρχικό ποσό των χρημάτων και καθώς γίνονται κλικ στις διάφορες παραμέτρους, η επεξεργασία γίνεται αυτόματα και τα αποτελέσματα παρουσιάζονται δυναμικά σε κάποιο σημείο της ιστοσελίδας χωρίς να απαιτείται κάποιο submit button ή κάποια ανανέωση ιστοσελίδας. Όπως γίνεται εύκολα κατανοητό, υπάρχει μεγάλη διαφορά απο τον παραδοσιακό τρόπο αποστολής της φόρμας όπου έπρεπε πρώτα να συμπληρωθούν όλα τα απαραίτητα πεδία και έπειτα να γίνει κλικ στο submit button. Τα διαδοχικά requests/responses φαίνονται στο σχήμα III:



Σχήμα III: Διαδοχικά requests/responses σε μία Ajax based εφαρμογή

6.4 Είναι όμως η τεχνολογία Ajax καινούργια;

Το AJAX δεν είναι μια καινούργια τεχνολογία και στην ουσία δεν είναι καν μια τεχνολογία. Το AJAX, όπως γράφτηκε και πιο πάνω είναι ένας όρος για να περιγράψει τις διαδικασίες που χρησιμοποιούνται από την JavaScript και του XMLHttpRequest αντικειμένου, για την ανάκτηση πληροφορίας με ασύγχρονο τρόπο από έναν server.

6.5 Επίλογος

Με την ολοκλήρωση αυτού του κεφαλαίου έχουμε καλύψει τα τρία επίπεδα ενός web εγγράφου τα οποία είναι η δομή, η παρουσίαση και η συμπεριφορά. Στο επόμενο κεφάλαιο παρουσιάζεται η γλώσσα σεναρίου PHP που χρησιμοποιήθηκε για την δημιουργία της εφαρμογής.

PHP

7.1 Εισαγωγή

Η PHP (Hypertext Preprocessor) είναι μία ευρέως χρησιμοποιούμενη, ανοιχτού κώδικα και γενικού σκοπού γλώσσα σεναρίου που είναι ειδικά σχεδιασμένη για την ανάπτυξη εφαρμογών διαδικτύου και μπορεί να ενσωματωθεί μέσα σε κώδικα HTML και να εκτελείται κάθε φορά που ο χρήστης επισκέπτεται την σελίδα. Ο PHP κώδικας μεταφράζεται στον Web διακομιστή και δημιουργεί κώδικα HTML ή άλλη έξοδο που θα δει ο επισκέπτης. Αυτό που διαχωρίζει την PHP από τα client-side JavaScripts είναι ότι ο κώδικας εκτελείται στον server (εξυπηρετητή). Αν υπήρχε ένα script PHP, ο browser θα έπαιρνε τα αποτελέσματα της εκτέλεσης αυτού του script, χωρίς να μπορεί να καταλάβει με κανένα τρόπο τι κώδικας υπάρχει από κάτω. Μπορούμε ακόμα να ρυθμίσουμε τον Web Server ώστε να χειρίζεται όλα τα HTML αρχεία με την PHP. Αν και η ανάπτυξη της PHP εστιάζεται σε server-side scripting (scripting στην πλευρά του διακομιστή), μπορούν να γίνουν πολύ περισσότερα με αυτήν. Σημειώνεται ότι η τελευταία έκδοση της PHP είναι η 5.3.0 και είναι αυτή η οποία χρησιμοποιείται στην εφαρμογή.

7.2 Ιστορία της PHP

Η ιστορία της PHP ξεκινά από το 1995, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό script με όνομα php.cgi, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6, για οποιονδήποτε προγραμματιστή θέλει να τη χρησιμοποιήσει. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP.

7.3 Βασικά χαρακτηριστικά

Η PHP ενσωματώνει την ισχύ και τη δυναμικότητα σχετικά παλαιότερων γλωσσών όπως η Perl αλλά καταργώντας τις αδυναμίες τους. Αναφέρουμε μερικά από τα βασικά χαρακτηριστικά της:

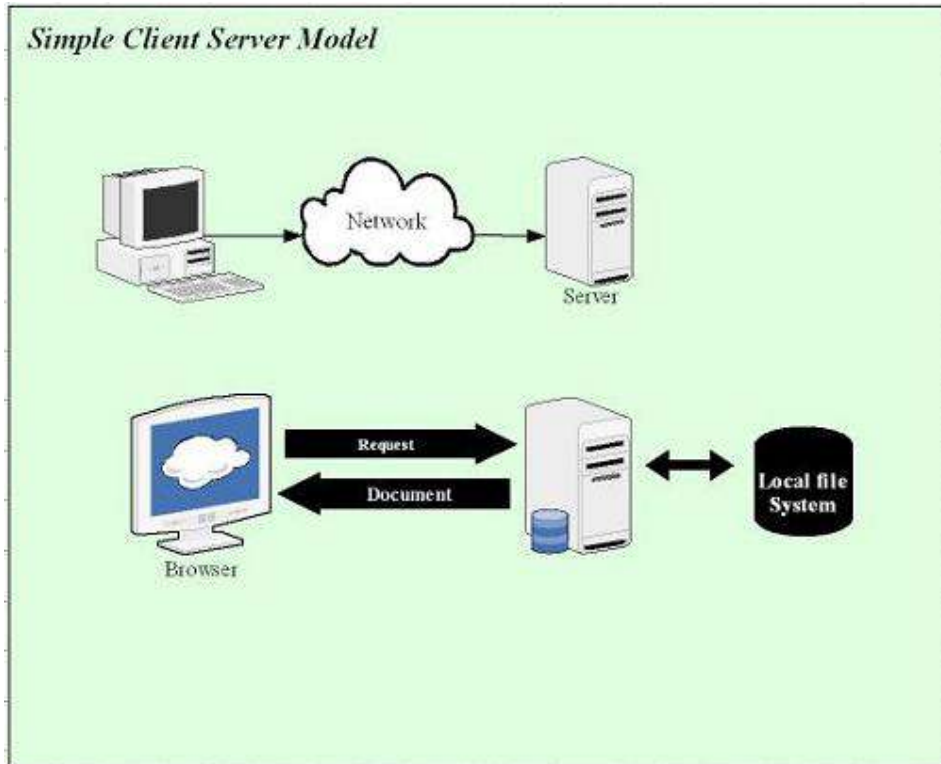
- Ο συντακτικός αναλυτής της, καθώς και ο πηγαίος κώδικάς της διανέμεται ελεύθερα στο διαδίκτυο, δίνοντας την δυνατότητα σε όποιον θέλει να κατασκευάζει και να διανέμει εφαρμογές για εμπορική και μη χρήση.

- Μπορεί να μεταφραστεί και να τρέξει στα περισσότερα λειτουργικά συστήματα που κυκλοφορούν στην αγορά (Microsoft Windows, Linux, BSD, Solaris, Macintosh OS X, και UNIX servers).
- Συνεργάζεται χωρίς προβλήματα με τους πιο δημοφιλείς Web Servers που κυκλοφορούν όπως τον Apache και τον Microsoft IIS. Διαθέτει ενσωματωμένες εντολές υποστήριξης για ένα μεγάλο αριθμό βάσεων δεδομένων όπως MySQL, Sybase, Oracle, Ingres. Προσφέρει ένα σύνολο από Database API's τις ενοποιημένες ODBC συναρτήσεις (unified ODBC functions), που εξασφαλίζουν την προσπέλαση σε μια υποκείμενη βάση δεδομένων, χρησιμοποιώντας τις εγγενείς μεθόδους της εκάστοτε βάσης για να μεγιστοποιήσουν την απόδοση (IBM DB2).
- Είναι πιο απλό να συντάξει κάποιος κώδικα PHP από ότι σε οποιαδήποτε άλλη γλώσσα σεναρίου.
- Μπορεί να χρησιμοποιηθεί στη δημιουργία εικόνων, ανάγνωση / εγγραφή σε αρχεία και για αποστολή email. Για να προσφέρει αυτές τις υπηρεσίες, η PHP επικοινωνεί με αρκετά πρωτόκολλα όπως: HTTP (Ιστοσελίδες), POP3 (e-mail), SNMP και LDAP.
- Υποστηρίζει τόσο τον διαδικαστικό προγραμματισμό όσο και τον αντικειμενοστραφή.

7.4 Αρχιτεκτονική Βάσης Δεδομένων με PHP – MySQL

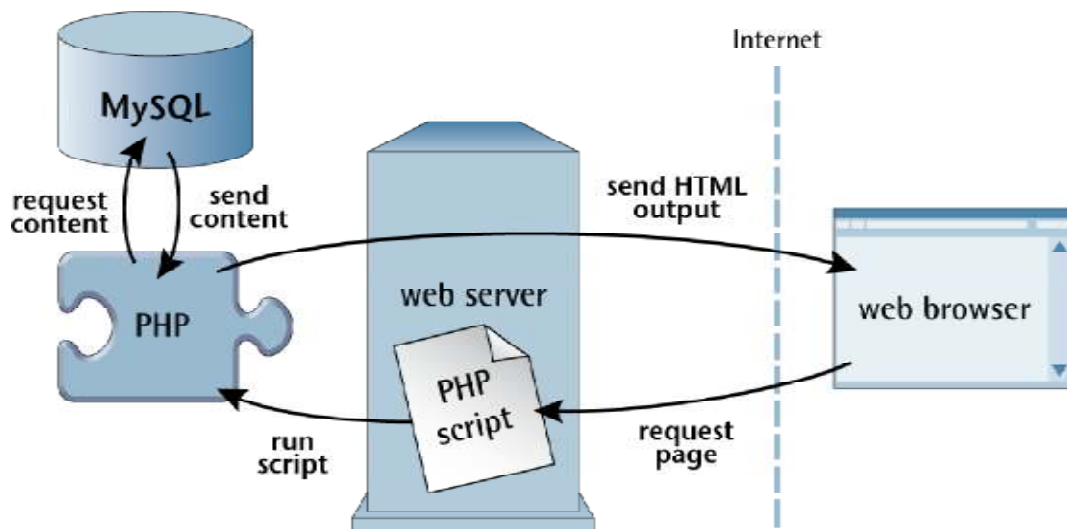
Η βασική λειτουργία ενός Web server φαίνεται στο Σχήμα Ι. Αυτό το σύστημα αποτελείται από δύο αντικείμενα από τα οποία το ένα είναι ο Web browser και το άλλο ο Web server. Απαιτείται μεταξύ τους μία σύνδεση επικοινωνίας. Ένας browser κάνει μία αίτηση στον server κι έπειτα ο server στέλνει πίσω μία απόκριση. Αυτή η αρχιτεκτονική εξυπηρετεί όταν ο διακομιστής παρέχει στατικές σελίδες. Σημειώνεται ότι οι διακομιστές είναι τα μηχανήματα που προσφέρουν υπηρεσίες ενώ οι πελάτες είναι τα μηχανήματα που ζητούν και δέχονται τις υπηρεσίες αυτές. Ένα μηχάνημα μπορεί να είναι οποιουδήποτε τύπου, ακόμα και των δύο τύπων ταυτόχρονα.

Επίσης, η γλώσσα μορφοποίησης που χρησιμοποιείται για τη δημιουργία ιστοσελίδων, είναι η HTML (Hypertext Markup Language) και το πρωτόκολλο το οποίο χρησιμοποιείται για την μεταφορά των σελίδων από τον διακομιστή στον πελάτη είναι το HTTP (Hypertext Transfer Protocol).



Σχήμα I: Η σχέση browser / server

Η αρχιτεκτονική που υποστηρίζει μία Web τοποθεσία με βάση δεδομένων είναι λίγο πιο περίπλοκη. Η Web εφαρμογή με βάση δεδομένων που αναπτύσσεται στην παρούσα εργασία ακολουθεί την γενική δομή που φαίνεται στο Σχήμα II.



Σχήμα II: Βασική Web αρχιτεκτονική βάσεων δεδομένων

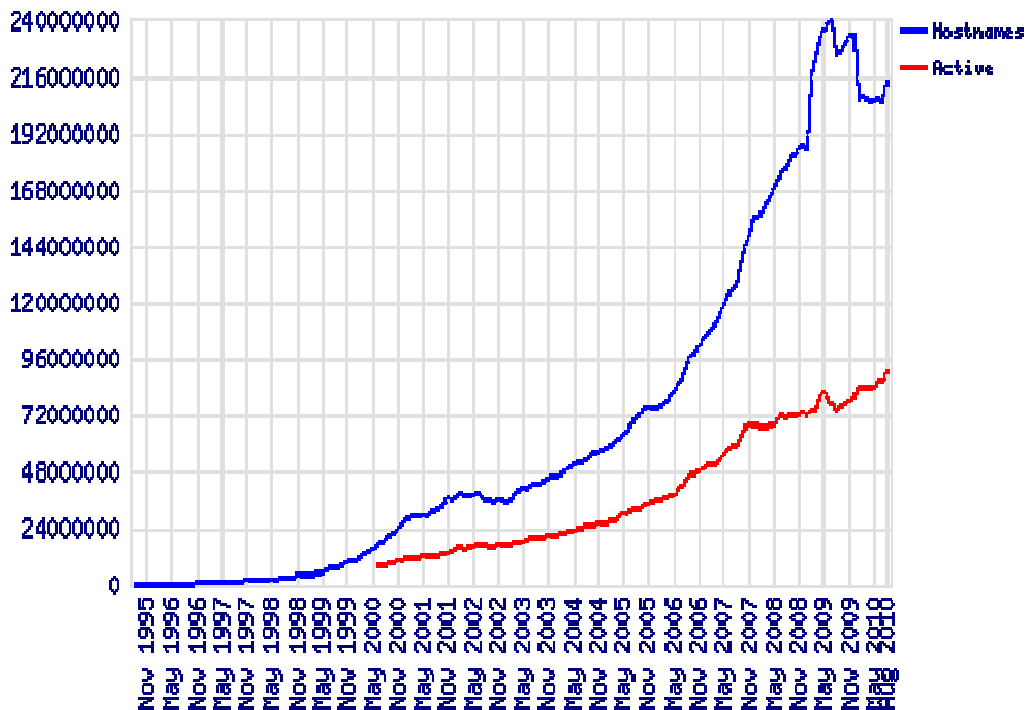
Μια τυπική Web συναλλαγή βάσεων δεδομένων αποτελείται από τις παρακάτω φάσεις:

1. Ο browser ενός χρήστη κάνει μια HTTP αίτηση για μια συγκεκριμένη σελίδα.
2. Ο server λαμβάνει την αίτηση για την συγκεκριμένη σελίδα, ανακαλεί το αρχείο και το περνά στην μηχανή PHP για επεξεργασία.
3. Η PHP μηχανή αρχίζει την ανάλυση του script. Μέσα στον κώδικα, υπάρχει μια εντολή που κάνει τη σύνδεση με την βάση δεδομένων και εκτελεί ένα ερώτημα. Η PHP ανοίγει μια σύνδεση με τον MySQL διακομιστή και στέλνει το κατάλληλο ερώτημα.
4. Ο MySQL διακομιστής λαμβάνει το ερώτημα της βάσης δεδομένων, το επεξεργάζεται και στέλνει τα αποτελέσματα ξανά στην PHP μηχανή.
5. Η PHP μηχανή σταματά την εκτέλεση του script, που συνήθως περιλαμβάνει την μορφοποίηση των αποτελεσμάτων του ερωτήματος σε HTML. Μετά, επιστρέφει την τελική HTML σελίδα στον server.
6. Ο server περνά την HTML σελίδα ξανά στον browser, όπου ο χρήστης μπορεί να δει τα αποτελέσματα. Η διαδικασία είναι η ίδια, ανεξάρτητα από το ποιά μηχανή script ή ποιός server βάσης δεδομένων χρησιμοποιείται. Συνήθως το πρόγραμμα του server, η PHP μηχανή και ο server της βάσης δεδομένων, βρίσκονται στον ίδιο υπολογιστή. Ωστόσο, είναι πολύ συνηθισμένο ο server της βάσης δεδομένων να βρίσκεται σε διαφορετικό υπολογιστή. Αυτό ,μπορεί να γίνει για λόγους ασφαλείας, για μεγαλύτερη χωρητικότητα ή για κατανομή του φόρτου.

7.5 Apache Web Server

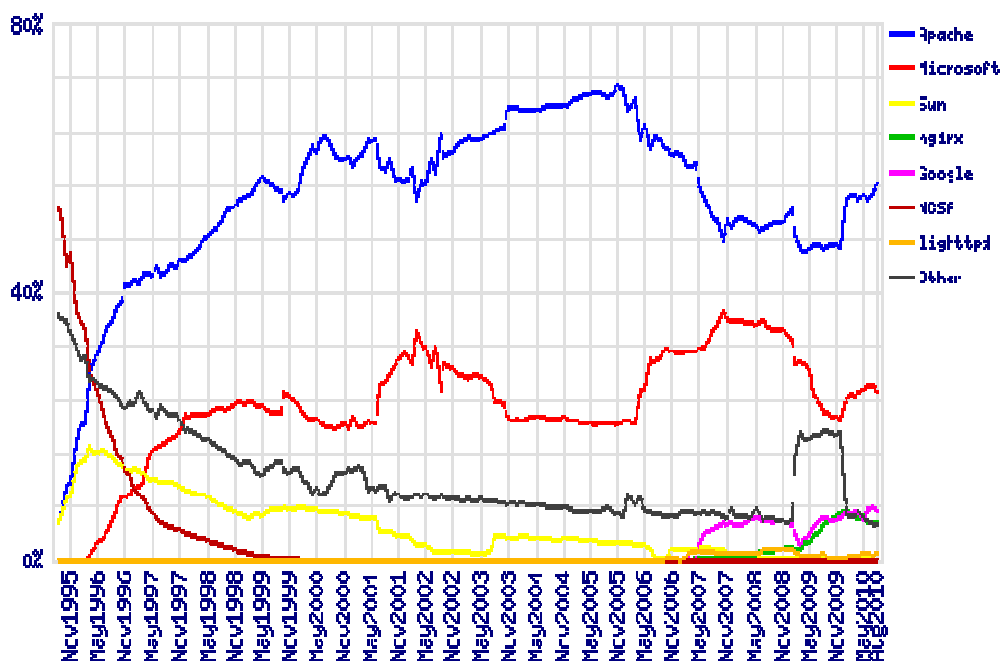
Ο Apache Web Server είναι ένας πολύ δημοφιλής διακομιστής διαδικτύου που διανέμεται ελεύθερα στο διαδίκτυο. Αναπτύχθηκε και συντηρείται από μια ομάδα εθελοντών που ήθελαν να υλοποιήσουν έναν εύρωστο κώδικα για διακομιστή δικτύου, που να είναι εμπορικός και να έχει πολλά χαρακτηριστικά. Σήμερα ο Apache θεωρείται από τους πιο σταθερούς διακομιστές δικτύου που κυκλοφορούν και θα πρέπει να τονίσουμε ότι αρκετοί εμπορικοί διακομιστές διαδικτύου, όπως ο HTTP Server της IBM, χρησιμοποιούν τον πυρήνα του Apache.

Στο σχήμα III φαίνεται ο αριθμός των συνολικών ιστοσελίδων που φιλοξενούνται από συστήματα που βασίζονται στον Apache κάνοντάς τον την πρώτη επιλογή ανάμεσα στους υπόλοιπους server(πηγή: <http://news.netcraft.com>).



Σχήμα III: Μεριδίο ανάμεσα στους Server (Σύνολο Ιστοσελίδων, Αύγουστος 1995 - Αύγουστος 2010)

Σημαντικά είναι και τα συγκριτικά στοιχεία που παρουσιάζονται στο σχήμα IV:



Σχήμα IV: Σύγκριση ανάλυση της χρήσης server ανά εταιρεία

Η συγκριτική αυτή ανάλυση δείχνει τη δυναμική του Apache καθώς διατηρεί με διαφορά την πρώτη θέση στην αγορά(πηγή: <http://news.netcraft.com>).

7.6 Hypertext Transfer Protocol (HTTP)

Το πρωτόκολλο HTTP καθορίζει τον τρόπο επικοινωνίας στο διαδίκτυο μεταξύ των διακομιστών και των πελατών (servers - clients). Είναι ένα γενικό, αντικειμενοστραφές πρωτόκολλο που μεταβιβάζει πληροφορία μεταξύ των διακομιστών και των πελατών. Ξεκίνησε από την έκδοση HTTP/0.9 κατά τη διάρκεια της πρώιμης ανάπτυξης του διαδικτύου και ακολούθησε η έκδοση HTTP/1.0 το 1995. Η πιο πρόσφατη έκδοσή του, HTTP/1.1, παρέχει περισσότερη λειτουργικότητα και υποστηρίζει πολλαπλές συναλλαγές μεταξύ πελάτη και διακομιστή κατά τη διάρκεια της ίδιας αίτησης.

7.7 Εγκατάσταση εργαλείων

Η εγκατάσταση των εργαλείων PHP, MySQL, Apache και PHPMyAdmin γίνεται πλέον πολύ εύκολα κατεβάζοντας ένα και μόνο αρχείο από την ιστοσελίδα <http://www.wampserver.com/en/download.php>

7.8 Επίλογος

Στα παραπάνω κεφάλαια έγινε η παρουσίαση όλων των τεχνολογιών που χρησιμοποιήθηκαν στην ανάπτυξη της εφαρμογής καθώς και μερικών τμημάτων της κάθε τεχνολογίας για γίνουν κατανοητές. Στο επόμενο κεφάλαιο επικεντρωνόμαστε στις τεχνολογίες php και mysql και συγκεκριμένα γίνεται ανάλυση του κώδικα που χρησιμοποιήθηκε για να υλοποιηθούν βασικές λειτουργίες του συστήματος.

ΥΛΟΠΟΙΗΣΗ ΣΕ PHP

8.1 Εισαγωγή

Στο παρόν κεφάλαιο παρατίθεται ο κώδικας των βασικότερων λειτουργιών της εφαρμογής, ο οποίος συνοδεύεται από κατάλληλη τεκμηρίωση ώστε να γίνει εύκολα κατανοητός από τον αναγνώστη.

8.2 Υπερκαθολικές μεταβλητές `$_GET` και `$_POST`

Επειδή η HTML είναι stateless δηλαδή χωρίς κατάσταση, πρέπει να βρεθεί κάποιος τρόπος ώστε να ελέγχονται οι διάφορες καταστάσεις και μεταβλητές των εφαρμογών. Αυτό γίνεται με τις υπερκαθολικές μεταβλητές `$_GET` και `$_POST`. Η php παίρνει από τη διεύθυνση URL ό,τι υπάρχει μετά το `?` και το τοποθετεί σε έναν πίνακα που είναι μία υπερκαθολική μεταβλητή, η `$_GET`. Με την εντολή `print_r($_GET)` μπορούμε δούμε τα περιεχόμενα της `$_GET`. Είναι πολύ χρήσιμο κατά το πέρασμα URL με PHP να χρησιμοποιούμε μία μεταβλητή για να πάρουμε την τοποθεσία URL όπου βρισκόμαστε. Για παράδειγμα ο κώδικας:

```
$id = $_GET['id'];
```

θα πάρει τη μεταβλητή `$id` από την URL και θα την αποθηκεύσει σε μία τοπική μεταβλητή `$id`. Με αυτόν τον τρόπο μπορούν να αποσταλούν τιμές ανάμεσα στις ιστοσελίδες.

8.3 Η υπερκαθολική μεταβλητή `$_SESSION`

Μία session είναι ένα αρχείο που αποθηκεύεται στον web server. Μπορούμε να βρούμε ποιο αρχείο ανήκει σε κάθε χρήστη κάνοντας χρήση των cookies. Θέτουμε ένα ειδικό session cookie στον browser και μετά κοιτάμε σε αυτό το cookie για να βρούμε το μέρος όπου βρίσκεται το αρχείο στον server μας και έπειτα ψάχνουμε στο αρχείο αυτό για τις πληροφορίες που θέλουμε. Οι πληροφορίες αυτές δεν μπορούν να αλλάξουν αλλά το μόνο που μπορούμε να κάνουμε είναι να αλλάξουμε το id number το οποίο είναι πολύ μεγάλο για λόγους ασφαλείας. Έτσι για να χρησιμοποιήσουμε μία session πρώτα δημιουργούμε ένα τέτοιο αρχείο για να θέσουμε το cookie στον υπολογιστή του χρήστη ή αν αυτό έχει ήδη γίνει να βρούμε αυτό το cookie και να βρούμε το αντίστοιχο αρχείο σε αυτόν τον υπολογιστή. Έτσι πρέπει να αρχίσουμε με μία `session_start()`. Αυτή πρέπει να κληθεί στην αρχή του αρχείου πριν εκτυπώσουμε οποιαδήποτε HTML πληροφορία ή ακόμα και κενό και πριν στείλουμε οποιαδήποτε πληροφορία στον browser του χρήστη. Έτσι η PHP βρίσκει από τον browser το id number του session cookie, βρίσκει το αρχείο και το ανοίγει ώστε να χρησιμοποιηθεί. Αν δεν βρει το id number ή αν δε μπορεί να βρει το session file τότε αρχίζει μία καινούρια session και διαμορφώνει το κατάλληλο cookie.

8.4 Εκκίνηση μίας συνόδου λειτουργίας

Πριν χρησιμοποιηθεί λειτουργικά μία σύνοδος, πρέπει πρώτα να ξεκινήσει. Υπάρχουν δύο τρόποι για να γίνει αυτό. Ο πρώτος και απλούστερος, είναι να ξεκινήσει ένα script με μία κλήση στην συνάρτηση `session_start()`. Αυτή η συνάρτηση ελέγχει αν υπάρχει ήδη μια τρέχουσα σύνοδος. Αν όχι, θα δημιουργήσει μία, παρέχοντας πρόσβαση στον υπερκαθολικό πίνακα `$_SESSION`. Αν υπάρχει ήδη μία σύνοδος, η `session_start()` φορτώνει τις εγγεγραμμένες μεταβλητές συνόδου ώστε να μπορούν να χρησιμοποιηθούν. Σημειώνεται ότι η `session_start()` πρέπει να είναι η πρώτη γραμμή που εκτελείται όταν χρειάζεται να χρησιμοποιηθεί μία σύνοδος λειτουργίας. Ο δεύτερος τρόπος για να ξεκινήσει μία σύνοδος είναι να οριστεί η PHP έτσι ώστε να ξεκινά μία σύνοδο αυτόματα όταν κάποιος έρχεται στην τοποθεσία της εφαρμογής. Αυτό μπορεί να γίνει με την επιλογή `session.auto_start` στο αρχείο `php.ini`. Αυτή η μέθοδος όμως έχει το μεγάλο μειονέκτημα ότι με την επιλογή `auto_start` ενεργοποιημένη, δεν μπορούν να χρησιμοποιηθούν αντικείμενα ως μεταβλητές συνόδου.

8.5 Έλεγχος για το αν ο χρήστης είναι συνδεδεμένος

Στην εφαρμογή χρησιμοποιείται ο παρακάτω κώδικας για τον έλεγχο αν κάποιος χρήστης είναι συνδεδεμένος ή όχι :

```
session_start ();
function logged_in() {
    return isset($_SESSION['session_key']);
}
function confirm_logged_in() {
    if (!logged_in()) {
        redirect_to ('index.php? mod=home');} }
function redirect_to($location = NULL) {
    if ($location != NULL) {
        header ("Location: {$location}");
        exit;
    }
}
```

Το πρώτο πράγμα που γίνεται όταν ο χρήστης περιηγείται στις σελίδες είναι να γίνει κλήση της συνάρτησης `confirm_logged_in()`, η οποία ελέγχει αν το 'session_key' του χρήστη βρίσκεται στον πίνακα `$_SESSION`. Το 'session_key' του κάθε μέλους είναι μοναδικό και αποθηκεύεται στον πίνακα αυτόν κατά την είσοδό του στην εφαρμογή. Έτσι, αν ο χρήστης είναι συνδεδεμένος τότε συνεχίζει την περιήγησή του στην εφαρμογή, διαφορετικά καλείται η συνάρτηση `redirect_to()` όπου τον μεταφέρει στην αρχική σελίδα. Εδώ σημειώνεται ότι η εντολή `header()` στέλνει HTTP header ώστε να μεταφερθεί η εκτέλεση σε άλλη ιστοσελίδα. Πολύ σημαντικό είναι ότι πρέπει να γίνει κλήση της `header()` πριν

σταλεί στην έξοδο οποιοσδήποτε HTML κώδικας (ετικέτες, κείμενο ή ακόμα και κενά) αλλιώς θα προκληθεί λάθος.

8.6 Σύνδεση με την βάση δεδομένων

Πριν αρχίσει κάποια εργασία με τη βάση δεδομένων, θα πρέπει πρώτα να υπάρξει σύνδεση με τον διακομιστή. Για το λόγο αυτό η PHP περιέχει τη ρουτίνα `mysql_connect()`. Η `mysql_connect()` δεν απαιτεί κανένα όρισμα αλλά δέχεται τρία strings : το `hostname`, ένα `username` και ένα `password`. Εάν δεν δοθεί κανένα από αυτά τα ορίσματα, η ρουτίνα υποθέτει ότι το `host` είναι `localhost` και ότι τα `username` και `password` δεν έχουν οριστεί στο `mysqluser` πίνακα, εκτός αν έχουν οριστεί στο αρχείο `php.ini`. Η `mysql_connect()` επιστρέφει έναν αναγνωριστή συνδέσμου (`link identifier`) αν η σύνδεση είναι επιτυχής. Η επιστρεφόμενη τιμή μπορεί να αποθηκευτεί σε μία μεταβλητή ώστε να μπορεί να προσπελαστεί ξανά χωρίς να χρειαστεί να εκτελεσθεί πάλι η μέθοδος `mysql_connect()`. Όταν γίνει η σύνδεση με τον διακομιστή βάσης δεδομένων θα πρέπει να επιλεγεί η βάση δεδομένων με την οποία θα λειτουργεί η εφαρμογή. Η δυνατότητα αυτή δίνεται από την συνάρτηση `mysql_select_db()`. Η συγκεκριμένη μέθοδος απαιτεί το όνομα μιας βάσης δεδομένων και προαιρετικά έναν αναγνωριστή συνδέσμου. Η `mysql_select_db()` επιστρέφει `true` αν η βάση δεδομένων υπάρχει και μπορεί να γίνει χειρισμός αυτής. Το επόμενο κομμάτι κώδικα χρησιμοποιεί τη `mysql_connect()` και `mysql_select_db()` για να συνδεθεί με τον διακομιστή βάσης δεδομένων MySQL και να χρησιμοποιήσει τη βάση δεδομένων που καθορίζει η μεταβλητή `MY_DATABASE`:

```
require_once ("constants.php");
function db_connection()
{
    $connection = mysqli_connect (SERVER, USER, MY_PASSWORD,
MY_DATABASE);
    $connection->query ("SET NAMES 'utf8'");
    if (!$connection)
    {
        die ("Database connection failed: " . mysqli_error());
    }
    return $connection;
}
function db_selection()
{
    $db_select = mysqli_select_db (MY_DATABASE, db_connection());
    if (!$db_select)
    {
        die ("Database selection failed: " . mysqli_error ());
    }
}
```

```
}  
    return $db_select;  
}
```

Σημειώνεται ότι για λόγους ασφαλείας καλό είναι το όνομα της βάσης, ο κωδικός αλλά και το όνομα χρήστη να αποθηκεύονται σε ξεχωριστό αρχείο στον server και να γίνεται εισαγωγή αυτού με την συνάρτηση include ή require.

8.7 Διαφυγή χαρακτήρων

Για την αποφυγή αυτών των προβλημάτων συνίσταται η χρήση μεθόδων διαφυγής χαρακτήρων. Στην PHP χρησιμοποιείται η μέθοδος `mysql_real_escape_string()`. Παρακάτω παρουσιάζεται η μέθοδος αυτή όπου εφαρμόζεται σε όλη την παρούσα εφαρμογή:

```
function mysql_prep($value, $connection){  
    $magic_quotes_active = get_magic_quotes_gpc(); $new_enough_php =  
        function_exists("mysql_real_escape_string");  
  
    if($new_enough_php){  
        if($magic_quotes_active){  
            $value = stripslashes($value);} $value = mysql_real_escape_string($connection,  
            $value);  
        } else {  
            if (!$magic_quotes_active) {  
                $value = addslashes($value);  
            }  
        }  
        return $value;  
    }  
}
```

Παρατηρείται η συνάρτηση `get_magic_quotes_gpc()`. Αυτό που κάνει η εντολή αυτή είναι να ελέγχει αν η οδηγία `magic_quotes_gpc` είναι ενεργοποιημένη ή όχι. Σε περίπτωση που η οδηγία αυτή είναι ενεργοποιημένη τότε η διαφυγή χαρακτήρων γίνεται αυτόματα προσθέτοντας όπου αυτό απαιτείται τον χαρακτήρα “ ` ” Όμως αυτή η οδηγία πρόκειται να αφαιρεθεί από τις μελλοντικές εκδόσεις της PHP και ήδη η χρήση της δεν συνίσταται. Έτσι , αφού ελεγχθεί αν η οδηγία αυτή είναι ενεργοποιημένη, τότε ελέγχεται αν υπάρχει η συνάρτηση

`mysql_real_escape_string` αφού αυτή προστέθηκε μετά την έκδοση 4. Αυτό γίνεται επειδή μερικές φορές μπορεί ο προγραμματιστής να μην γνωρίζει την έκδοση της PHP που χρησιμοποιεί ο server. Έπειτα γίνεται έλεγχος και γίνεται η διαφυγή χαρακτήρων όπου αυτό είναι απαραίτητο και επιστρέφεται η κατάλληλη τιμή με ασφαλή πλέον δεδομένα.

8.8 Πρότυπο layout

Για την επίτευξη κοινού layout σε όλες τις σελίδες της εφαρμογής χρησιμοποιείται στην αρχή κάθε σελίδας η μέθοδος όπου έχει ως όρισμα το τίτλο της σελίδας, το αρχείο με τους CSS κανόνες καθώς και τα αρχεία JavaScript. Αυτή είναι η:

```
function html_title($title, $stylesheet, $javascript) { ... ..  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
<title>'.$title.'</title> <link href='.$stylesheet.'" rel="stylesheet" type="text/css" />
```

Με τη μέθοδο αυτή επιτυγχάνεται η απευθείας ανάθεση του CSS καθώς και ενός προκαθορισμένου στυλ σε όλες τις σελίδες πριν ακόμα αρχίσει να εκτελείται οποιοδήποτε κομμάτι κώδικα.

8.9 Ακύρωση εγγραφής μεταβλητών και καταστροφή συνόδου

Η ακύρωση μίας μεταβλητής συνόδου γίνεται κατευθείαν ακυρώνοντας το κατάλληλο στοιχείο στον πίνακα `$_SESSION`. Για παράδειγμα :

```
unset ($_SESSION['session_key']);
```

Τέλος, όταν ακυρωθούν όλες οι μεταβλητές πρέπει να γίνει κλήση στην `session_destroy()` ώστε να καθαρίσει ο κωδικός της συνόδου. Ο παρακάτω κώδικας χρησιμοποιείται για την αποσύνδεση του χρήστη από την εφαρμογή.

```
function logout() {  
    $_SESSION = array ();  
    if (isset($_COOKIE[session_name()])) {  
        setcookie (session_name(), "", time()-42000, '/');  
    }  
    unset ($_SESSION['session_key']);  
    session_destroy ();  
    redirect_to ("index.php? mod=home");  
}
```

8.10 Διαγραφή μαθήματος

Η μέθοδος `delete_subjects` απενεργοποιεί ένα μάθημα από την βάση δεδομένων. Αφού πάρει τα δεδομένα από την φόρμα (αν έχουμε επιλέξει μάθημα προς διαγραφή δηλαδή), εκτελεί το sql ερώτημα για να ανακτήσει το πεδίο `active` από την βάση δεδομένων και αν το πεδίο αυτό είναι `yes` το αλλάζει σε `no` και αντιστρόφως. Αν το πεδίο `active` είναι `yes` τότε το μάθημα είναι ενεργοποιημένο.

```
function delete_subjects($connection, $href) {
    $s = "";
    if (!empty($_POST['checkSubjects'])) {
        $check = $_POST ['checkSubjects'];
        for ($i = 0; $i < count($check); $i++) {
            $del = mysql_prep ($connection, $check [$i]);
            $query1 = "SELECT active FROM subjects_description
WHERE id = '$del'";
            $result1 = q ($connection, $query1);
            $row1 = mysqli_fetch_array ($result1);
            $active = $row1 ['active'];
            if ($active == 'yes') {
                $change_active = 'no';
            } elseif ($active == 'no') {
                $change_active = 'yes';
            }
            $query1 = "UPDATE subjects_description SET active =
'$change_active' WHERE id = '$del'";
            $result1 = q ($connection, $query1);
        }
        $_SESSION ['dok'] = 'yes'; //Μήνυμα ότι το βιβλίο διαγράφηκε
        redirect_to ($href);
    } else {
        $s. = '<p class="warning">Παρακαλώ επιλέξτε μάθημα.</p>';
    }
    return $s;
}
```

8.11 Διαγραφή διαχειριστή

Η παρακάτω μέθοδος διαγράφει έναν διαχειριστή από την βάση δεδομένων. Ελέγχει αν έχουμε επιλέξει τον διαχειριστή προς διαγραφή και τον διαγράφει. Στην περίπτωση που επιχειρήσουμε να διαγράψουμε τον εαυτό μας ενώ είμαστε συνδεδεμένοι με τα στοιχεία μας το σύστημα δεν μας το επιτρέπει.

```
function delete_admins($connection, $href) {
    $s = "";
    if (!empty($_POST['checkAdmins'])) {
        $check = $_POST ['checkAdmins']; //Τα checkbox
        for ($i = 0; $i < count($check); $i++) {
            $del = mysql_prep ($connection, $check[$i]);
            if ($del == $_SESSION['id']) {
                $s. = '<p class="warning">Δε μπορείτε να διαγράψετε
τον εαυτό σας. </p>';
                continue;
            }
            $query1 = "DELETE FROM admins WHERE id = '$del'";
            $result1 = q ($connection, $query1);
        }
        $_SESSION ['dok'] = 'yes'; //Μήνυμα διαγραφής διαχειριστή
        redirect_to ($href);
    } else {
        $s. = '<p class="warning">Παρακαλώ επιλέξτε διαχειριστή. </p>';
    }
    return $s;
}
```

8.12 Ανοιγμα/Κλείσιμο διαδικασίας δηλώσεων

Η παρακάτω μέθοδος ανοίγει ή κλείνει αντίστοιχα την διαδικασία των ηλεκτρονικών δηλώσεων, αλλάζοντας το πεδίο active του πίνακα assigns_is_continued. Έπειτα ενημερώνει τον πίνακα για τις αλλαγές.

```
function do_change_assigns_status($connection) {
    $s = "";
    $admins_id = $_SESSION ['id'];
    $mod = sanatize_get ('mod', 'str', "");
    $url = ROOT.'mod='. $mod;
    $query1 = "SELECT * FROM assigns_is_continued;";
    $result1 = q ($connection, $query1); //Το αποτέλεσμα του query
    $row1 = mysqli_fetch_array ($result1);
    $active = $row1 ['active']; //Ανάκτηση πεδίου active απο την βάση
    if ($active == 'yes') { //Αλλαγή του πεδίου ανάλογα με την τιμή του
        $activation = 'no';
    } elseif ($active == 'no') {
        $activation = 'yes';
    } //Ενημέρωση του πίνακα στην βάση
}
```

```
$query2 = "UPDATE assigns_is_continued SET admins_id = '$admins_id',
active = '$activation';";
$result2 = q($connection, $query2);
redirect_to($url);
return $s;
}
```

8.13 Εκκίνηση νέων δηλώσεων

Ο παρακάτω κώδικας αφορά την εκκίνηση νέων δηλώσεων. Αφού μεταφέρει όλα τα περιεχόμενα του πίνακα `students_to_books_choises` στον πίνακα `students_to_books_final` για την διατήρηση ιστορικών στοιχείων, διαγράφει τα στοιχεία του πίνακα `students_to_books_choises` και ανανεώνει τον πίνακα `assigns_is_continued` αλλάζοντας το πεδίο `active` σε `yes`. Ο κώδικας αυτός εκτελείται όταν επιλέξουμε από την σελίδα διαχείρισης την επιλογή «Εκκίνηση νέων δηλώσεων».

```
function do_new_assigns($connection) {
    $s = "";
    $admins_id = $_SESSION ['id'];
    $mod = sanatzize_get ('mod', 'str', "");
    $url = ROOT.'mod='. $mod;
    $query1 = "SELECT * FROM students_to_books_choises;";
    $result1 = q ($connection, $query1);
    while ($row1 = mysqli_fetch_array($result1)) {
        $students_id = $row1 ['students_id'];
        $books_to_subjects_id = $row1 ["books_to_subjects_id"];
        $query2 = "INSERT INTO students_to_books_final (students_id,
books_to_subjects_id) VALUES ('$students_id', '$books_to_subjects_id');";
        $result2 = q($connection, $query2);
        $query3 = "DELETE FROM students_to_books_choises WHERE
students_id = '$students_id' AND books_to_subjects_id =
'$books_to_subjects_id';";
        $result3 = q($connection, $query3);}
    $query4 = "UPDATE assigns_is_continued SET active = 'yes',
has_assigned = 'no';";
    $result4 = q($connection, $query4);
    redirect_to ($url);
    return $s;}
}
```

8.14 Εμφάνιση αποτελεσμάτων

Ο συνδυασμός της μεθόδου `odd_even($mode)`

```
function odd_even($mode) {  
    if ($mode % 2 == 0) {  
        $cl = "even";  
    } else if ($mode % 2 != 0) {  
        $cl = "odd";  
    }  
    return $cl;  
}
```

και του κώδικα CSS του αρχείου `same_admin.css`

```
.odd {  
    width: 800px;  
    position: relative;  
    background-color: #C9C9C9;}  
  
.even {  
    width: 800px;  
    position: relative;  
    background-color: #E2E2E2;}
```

μας βοηθά να χειριστούμε την μεταβλητή `$mode` κατάλληλα έτσι ώστε η μεταβλητή `$cl` να πάρει μονές ή ζυγές τιμές και να εμφανιστεί το ανάλογο χρώμα. Αν ορίσουμε το `$mode` να είναι 1 και προσαυξήσουμε το `$mode++` τότε η μεταβλητή `$cl` θα παίρνει εναλλάξ τιμές και θα εκτελείται κάθε φορά ο αντίστοιχος κώδικας CSS. Έτσι πετυχαίνουμε μια ομορφότερη προβολή των αποτελεσμάτων μας στην εφαρμογή.

8.15 Προσαρμογή της θέσης του μαθήματος στο εξάμηνο (SubjectRank)

Κατά την εισαγωγή ενός νέου μαθήματος, το σύστημα θα εισάγει το νέο μάθημα στην επόμενη κενή θέση στο εξάμηνο. Για παράδειγμα έστω ότι ένα εξάμηνο περιέχει 4 μαθήματα και θέλουμε να εισάγουμε το μάθημα XX. Τότε αυτό θα τοποθετηθεί στην 5^η θέση γιατί είναι η επόμενη κενή. Αν όμως εμείς αποφασίσουμε ότι θέλουμε να το βάλουμε στην 2^η θέση επειδή είχαμε κάνει κάποιο λάθος σε προηγούμενη εισαγωγή, το σύστημα μας δίνει την δυνατότητα να το κάνουμε. Με απλά λόγια σε περίπτωση που στο εξάμηνο A έχουμε εισάγει 10 μαθήματα με οποιαδήποτε σειρά, μπορούμε να αναδιοργανώσουμε τον πίνακα μαθημάτων και να τοποθετήσουμε τα μαθήματα στην σωστή σειρά και με τους σωστούς κωδικούς, χωρίς να απαιτείται να διαγράψουμε όλον τον πίνακα και να εισάγουμε ξανά. Ο παρακάτω κώδικας επιστρέφει τον αριθμό των μαθημάτων προσαυξημένο κατά ένα, που υπάρχουν στο εξάμηνο που επιλέγουμε κατά την εισαγωγή μαθήματος.

```
include_once ("modules/session.php"); //Κώδικα 1
include_once ("admin/db_connection.php");
$connection = db_connection ();
$semesters_id = $_GET['semester'];
$query1 = "SELECT COUNT (*) AS total FROM subjects WHERE semesters_id
= '$semesters_id'";
$result1 = q($connection, $query1);
$row1 = mysqli_fetch_array ($result1);
$total = $row1 ['total'] + 1;
echo $total;
```

Η επόμενος κώδικας χρησιμεύει είτε στην περίπτωση που θέλουμε να κάνουμε τροποποίηση στο rank και να το βάλουμε στο ίδιο εξάμηνο, οπότε το total θα είναι ίσο με τον αριθμό των μαθημάτων του εξαμήνου, είτε αν θέλουμε να αλλάξουμε ένα μάθημα και να το βάλουμε σε άλλο εξάμηνο, οπότε το total θα είναι ίσο με τον αριθμό των μαθημάτων που βρίσκονται στο εξάμηνο προς εισαγωγή συν ένα.

```
include_once ("modules/session.php"); //Κώδικα 2
include_once ("admin/db_connection.php");
$connection = db_connection ();
$semesters_id = $_GET ['semester'];
$subjects_id = $_GET ['subjects_id'];
$query100 = "SELECT semesters_id FROM subjects WHERE id =
'$subjects_id'";
$result100 = q($connection, $query100);
$row100 = mysqli_fetch_array ($result100);
$initial_semesters_id = $row100 ['semesters_id'];

$query200 = "SELECT COUNT (*) AS total FROM subjects WHERE
semesters_id = '$initial_semesters_id'";
$result200 = q($connection, $query200);
$row200 = mysqli_fetch_array ($result200);
$total_initial_semester = $row200 ['total'];

$query1 = "SELECT COUNT (*) AS total FROM subjects WHERE semesters_id
= '$semesters_id'";
$result1 = q ($connection, $query1);
$row1 = mysqli_fetch_array ($result1);
$total = $row1 ['total'] + 1;

if ($initial_semesters_id == $semesters_id) {
```

```
        echo $total_initial_semester;
    } else {
        echo $total;
    }
```

Με το που φύγει ο κέρσορας (όταν γίνει onblur δηλαδή) από το πεδίο του εξαμήνου που επιθυμούμε να βάλουμε το μάθημα, τότε καλείται η συνάρτηση socketsetup(), παίρνει την τιμή του εξαμήνου και καλεί το αρχείο που περιέχει τον κώδικα 1 (ή το αρχείο που περιέχει τον κώδικα 2 αν είμαστε στην τροποποίηση μαθήματος). Με το που εκτελεστεί ο κώδικας 1 τότε καλείται η συνάρτηση socketreceive() και δημιουργεί τις ανάλογες τιμές του rank.Σημείωση:Το rank αφορά την θέση του μαθήματος στο εξάμηνο δηλαδή πρώτο,δεύτερο,τρίτο κ.ο.κ

8.16 Έλεγχος εγκυρότητας τιμών

Στο παρακάτω απόσπασμα κώδικα γίνεται έλεγχος αν στο πεδίο του κινητού τηλεφώνου έχουμε πληκτρολογήσει έγκυρες τιμές.Αν ξεκινά δηλαδή με 69 και αν αποτελείται απο 10 αριθμούς.

```
function is_mobile($phone) {
    if (utf8_strlen($phone) != 10 || !preg_match("/^69/", $phone)) {
        return false;
    }
    return true;
}
```

Με παρόμοιο τρόπο ελέγχονται και πεδία όπως το σταθερό τηλέφωνο και το email.

8.17 Περιβάλλον – εφαρμογές – εργαλεία ανάπτυξης

Στην παρούσα ενότητα εκτίθενται τα χαρακτηριστικά του υπολογιστικού συστήματος που αναπτύχθηκε η συγκεκριμένη εφαρμογή, καθώς και τα εργαλεία που χρησιμοποιήθηκαν κατά τη φάση της ανάπτυξης.

Τεχνικές προδιαγραφές υπολογιστικού συστήματος

- Επεξεργαστής: Intel® Core™ 2 Duo CPU E8500 3.16GHz
- Κεντρική μνήμη (RAM): 2 GB DDR3 1333 MHz
- Σκληρός δίσκος: 640 GB
- Λειτουργικό σύστημα : Microsoft Windows 7 Ultimate 64-bit Operating System

Περιβάλλον: WampServer 2.0i

- Apache 2.2.11
- MySQL 5.1.36
- PHP 5.3.0
- phpMyAdmin

Σύστημα Διαχείρισης Βάσης Δεδομένων

- MySQL 5.1.36

Εργαλείο οπτικής απεικόνισης Βάσης Δεδομένων

- phpMyAdmin

Εργαλείο ανάπτυξης-συγγραφής κώδικα(Σχεδίαση ιστοσελίδας)

- Dreamweaver CS4

Εργαλείο ανάλυσης - σχεδιασμού UML διαγραμμάτων

- MagicDraw 16.0

Εργαλείο σχεδιασμού διαγράμματος οντοτήτων-συσχετίσεων ER

- Smart Draw VP

Εργαλείο επεξεργασίας λογοτύπων, κουμπιών, banner, footer κ.ο.κ

- Adobe Photoshop CS3

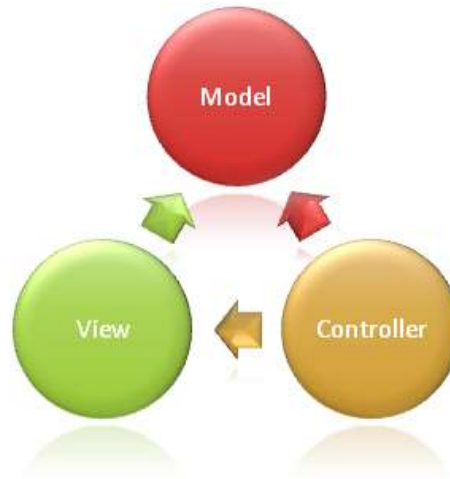
8.18 Εναλλακτική υλοποίηση του συστήματος-Προτάσεις για μελλοντική βελτίωση.

Τα τελευταία χρόνια η γλώσσα προγραμματισμού PHP η οποία χρησιμοποιείται κυρίως για την ανάπτυξη Web εφαρμογών έχει εκρηκτική ανάπτυξη σε σχέση με άλλες στον τομέα του Web. Επιπλέον μία από τις πιο διαδεδομένες αρχιτεκτονικές λογισμικού είναι αυτή του MVC (Model-View-Controller) η οποία χρησιμοποιείται και αυτή ευρέως στην ανάπτυξη λογισμικού Web. Ο συνδυασμός της αρχιτεκτονικής λογισμικού MVC και της γλώσσας PHP είχε ως αποτέλεσμα την ανάπτυξη των PHP Web Frameworks, τα οποία είναι εργαλεία προγραμματισμού γραμμένα στη γλώσσα PHP και επιτρέπουν στον Web Developer να αναπτύξει πολύ γρήγορα εφαρμογές Web και Web Sites. Αυτό γιατί εάν τα χρησιμοποιήσει σωστά τον βοηθούν να μην επαναλαμβάνει των κώδικα του, καθώς και να χρησιμοποιήσει ήδη έτοιμο κώδικα τον οποίο διαθέτουν αυτά τα frameworks. Ως εκ τούτου, η συγκεκριμένη εφαρμογή θα μπορούσε να αναπτυχθεί με την βοήθεια κάποιου php framework πχ cake php (ένα από τα πιο δημοφιλή και ολοκληρωμένα frameworks για php), το οποίο θα μείωνε σημαντικά την δουλειά ενός προγραμματιστή. Αναλυτικότερα πληροφορίες για τα διαθέσιμα php web frameworks υπάρχουν στην σελίδα: <http://www.phpframeworks.com/>.

8.19 Αρχιτεκτονική MVC

Η αρχιτεκτονική MVC (Model – View - Controller) αποτελεί την βάση για την ανάπτυξη των περισσότερων διαδικτυακών εφαρμογών σήμερα.

Η αρχή του MVC είναι να επιτυγχάνεται ο διαχωρισμός ανάμεσα στην επιχειρησιακή λογική της εφαρμογής και στο τμήμα που αφορά την παρουσίαση της. Με αυτό το διαχωρισμό είναι εύκολο πλέον να εκτελούνται αλλαγές σε ένα τμήμα χωρίς να επηρεάζεται το άλλο.



Σχήμα 1: Αναπαράσταση αρχιτεκτονικής MVC

Το MVC καθορίζει τρία τμήματα:

- Το μοντέλο (*Model*) της εφαρμογής το οποίο προσδιορίζει τα δεδομένα του επιπέδου της επιχειρησιακής λογικής καθώς και τους κανόνες που επιτρέπουν την αλληλεπίδραση με αυτά.
- Την παρουσίαση (*View*) της εφαρμογής που αναπαριστά τα δεδομένα του μοντέλου στις διεπαφές της εφαρμογής.
- Τον ελεγκτή (*Controller*) που αναλαμβάνει την σύνδεση του μοντέλου με την παρουσίαση καθώς και την μετατροπή των δεδομένων μεταξύ των επιπέδων ώστε να διατηρείται η επικοινωνία.

8.20 Οφέλη

Το MVC είναι ένα δοκιμασμένο και πραγματικό μοντέλο προγραμματισμού που κάνει μια εφαρμογή εύκολη στη συντήρηση, αρθρωτή και ένα γρήγορα αναπτυσσόμενο πακέτο. Η κατασκευή των μερών της εφαρμογής σε διαφορετικά μοντέλα, προβολές και ελεγκτές κάνει την εφαρμογή να στέκεται καλύτερα στα πόδια της. Νέα χαρακτηριστικά μπορούν να προστεθούν πολύ εύκολα.

Η αρθρωτή και διαμοιρασμένη αρχιτεκτονική επιτρέπει στους προγραμματιστές και σχεδιαστές να δουλεύουν ταυτόχρονα. Η μοιρασμένη αυτή αρχιτεκτονική επίσης επιτρέπει στους προγραμματιστές να πραγματοποιούν αλλαγές σε ένα μέρος του κώδικα χωρίς να επηρεάζουν άλλα.

(Πηγή: <http://book.cakephp.org/el/view/10/Understanding-Model-View-Controller>)

8.21 Βελτιώσεις – προσθήκες λειτουργιών στο υπάρχον σύστημα

Το υπάρχον σύστημα διαθέτει ένα σύνολο απο λειτουργίες οι οποίες το κάνουν λειτουργικό και καλύπτουν όσο το δυνατότερο περισσότερο τις ανάγκες του εκπαιδευτικού ιδρύματος. Αυτό δεν αποκλείει την περίπτωση βελτίωσης του συστήματος καθώς πάντα νέες λειτουργίες μπορούν να προστίθενται. Δεδομένου ότι η εκπόνηση της συγκεκριμένης πτυχιακής πραγματοποιήθηκε από ένα άτομο και σε καθορισμένο χρονικό διάστημα, πιθανώς ελείψεις μπορεί να υπάρχουν.

Τι γίνεται στην περίπτωση όπου ένας φοιτητής δεν δηλώσει ένα μάθημα και παρόλα αυτά κάνει ηλεκτρονική αίτηση για να πάρει βιβλίο; Μια αδυναμία της εφαρμογής αυτής είναι ότι δεν παρέχει έλεγχο, για το εάν ένας φοιτητής έχει δηλώσει προηγουμένως το μάθημα, ώστε να του δώσει δικαίωμα να πάρει το αντίστοιχο σύγγραμμα. Αυτό έχει ως αποτέλεσμα διάφοροι φοιτητές να μπαίνουν και να δηλώνουν βιβλία ακόμα και αν δεν τα δικαιούνται. Το σύστημα απλώς δέχεται τις αιτήσεις και η γραμματεία εκτελεί τον έλεγχο για το ποιοί φοιτητές δικαιούνται βιβλίο. Η εφαρμογή ελέγχει αν ένας φοιτητής έχει πάρει το ίδιο βιβλίο στο παρελθόν από ιστορικά στοιχεία που έχει συλλέξει. Μια προσθήκη - βελτίωση του υπάρχοντος συστήματος είναι ο έλεγχος νομιμότητας της επιλογής του φοιτητή για το συγκεκριμένο σύγγραμμα. Η ενσωμάτωση ουσιαστικά στο σύστημα της διαδικασίας ελέγχου που επιτελεί η γραμματεία της σχολής.

Το σύστημα θα ελέγχει εάν ο φοιτητής δήλωσε το μάθημα και θα του δίνει το δικαίωμα να προχωρήσει, διαφορετικά σε κάθε άλλη περίπτωση θα τον ενημερώνει με κατάλληλο μήνυμα.

Επιπλέον μπορεί να προστεθεί στην εφαρμογή η δυνατότητα συλλογής στατιστικών στοιχείων τα οποία θα προβάλλονται στην οθόνη του χρήστη. Τα στατιστικά στοιχεία αυτά μπορεί να αφορούν τον αριθμό των βιβλίων που έχει πάρει ένας συγκεκριμένος φοιτητής έως τώρα, τον αριθμό των αντιτύπων ανα έτος και ανά σύγγραμμα, την χρέωση του ΑΤΕΙ στα συγκεκριμένα βιβλία κ.ο.κ.

Τέλος μπορούν να προστεθούν δυνατότητες εκτύπωσης των απαραίτητων εγγράφων προς τους διανομείς των εκπαιδευτικών συγγραμμάτων.

8.22 Επίλογος

Με την ολοκλήρωση αυτού του κεφαλαίου ολοκληρώνεται το δεύτερο μέρος της πτυχιακής με την παρουσίαση των διάφορων τεχνολογιών που χρησιμοποιήθηκαν για την υλοποίησή της. Έτσι ο αναγνώστης τώρα έχει την απαραίτητη θεωρητική υποδομή για τα εργαλεία που μπορεί και ο ίδιος να χρησιμοποιήσει σε περίπτωση ενασχόλησης με κάποια web εφαρμογή.

Καταλυτική στην δημιουργία μιας τέτοιας εφαρμογής ήταν η γλώσσα προγραμματισμού php ενώ η συμμετοχή των φύλλων διαμόρφωσης στυλ CSS και της javascript αποτέλεσαν ένα βασικό μέρος του κορμού της εφαρμογής. Όλα αυτά όμως στηρίζονται στην βάση δεδομένων mysql η οποία αποτελεί τα θεμέλια της εφαρμογής.

Βιβλιογραφία

Σαλαμπάσης Μιχάλης (2008) *Εισαγωγή στον προγραμματισμό διαδικτυακών εφαρμογών*, Θεσσαλονίκη

Χατζηγεωργίου Ν. Αλέξανδρος (2005), *Αντικειμενοστραφής σχεδίαση*, Κλειδάριθμος, Αθήνα

Croft, J., Lloyd, I., Rubin, D., (2006), *Pro CSS Techniques*, Apress, Berkeley, USA.

Griffiths, P., (2007), *the Best-Practice Guide to XHTML & CSS*, HTML Dog, Berkeley, USA.

Heilmann, C., (2006), *Beginning JavaScript with DOM Scripting and Ajax: From Novice to Professional*, Apress, Berkeley, USA

Pfaffenberger, B., Shcafer, S., White, C., Karow, B., (2004), *HTML, XHTML, and CSS Bible, 3rd Edition*, Wiley Publishing, Inc., Indiana, USA.

Shultz, D., Cook, C., (2007), *Beginning HTML with CSS and XHTML: Modern Guide and Reference*, Apress, Berkeley, USA.

Welling, L., Laura, T., (2005), *PHP and MySQL Web Development, 3rd Edition*, Sams Publishing, Indiana, USA.

ΠΑΡΑΡΤΗΜΑ Α - Εγκατάσταση των εργαλείων

Εγκατάσταση εφαρμογής

Για την λειτουργία της εφαρμογής απαιτείται η εγκατάσταση επιπλέον λογισμικού το οποίο αναλύεται στη συνέχεια. Όλα τα απαιτούμενα αρχεία βρίσκονται στον φάκελο *setup* του CD εγκατάστασης.

Εγκατάσταση των εργαλείων

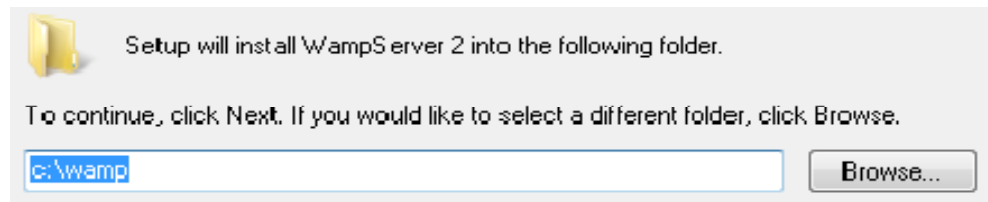
Ο WampServer (Windows, Apache, MySQL, PHP) είναι ένα δωρεάν, all-in-one πρόγραμμα το οποίο περιλαμβάνει ενσωματωμένα αντίγραφα των πιο πρόσφατων εκδόσεων του server Apache, της PHP και της MySQL. Ένα εναλλακτικό, all-in-one πρόγραμμα είναι το xampp. Η διαδικασία εγκατάστασής του είναι η ακόλουθη:

1. Κατεβάστε την τελευταία έκδοσή του WampServer από την ιστοσελίδα <http://www.wampserver.com/en/download.php>. Μόλις το αρχείο κατέβει (είναι περίπου 16MB), κάνετε διπλό κλικ στο αρχείο εγκατάστασης, όπως φαίνεται στο σχήμα I:



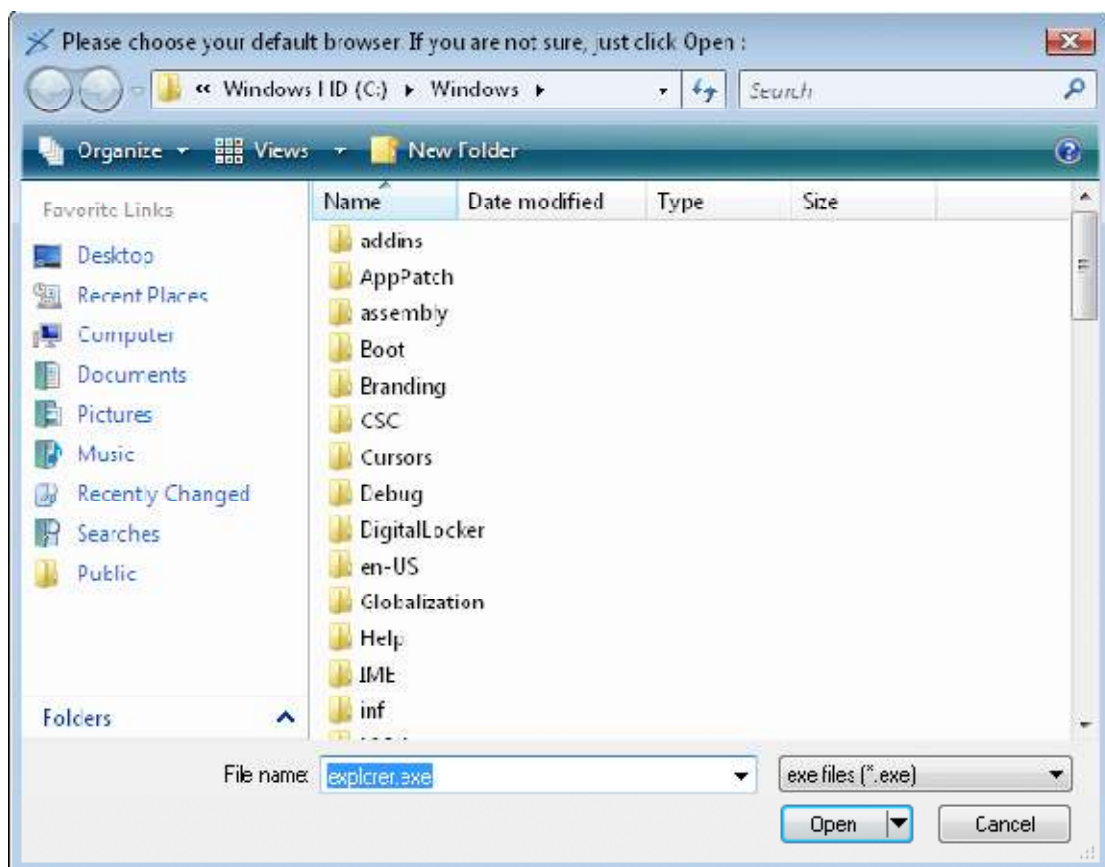
Σχήμα I: Η αρχική οθόνη του WAMP

2. Το αρχείο εγκατάστασης θα ζητήσει τη θέση όπου πρόκειται να εγκατασταθεί ο WampServer. Η προεπιλογή είναι η θέση *c:\wamp* η οποία φαίνεται στο σχήμα II, η οποία είναι συνήθως και η καλύτερη. Αν υπάρχει όμως κάποια άλλη θέση όπου επιθυμείτε να βρίσκονται όλα τα προγράμματα σας συγκεντρωμένα μπορείτε να την καθορίσετε στο σημείο αυτό.



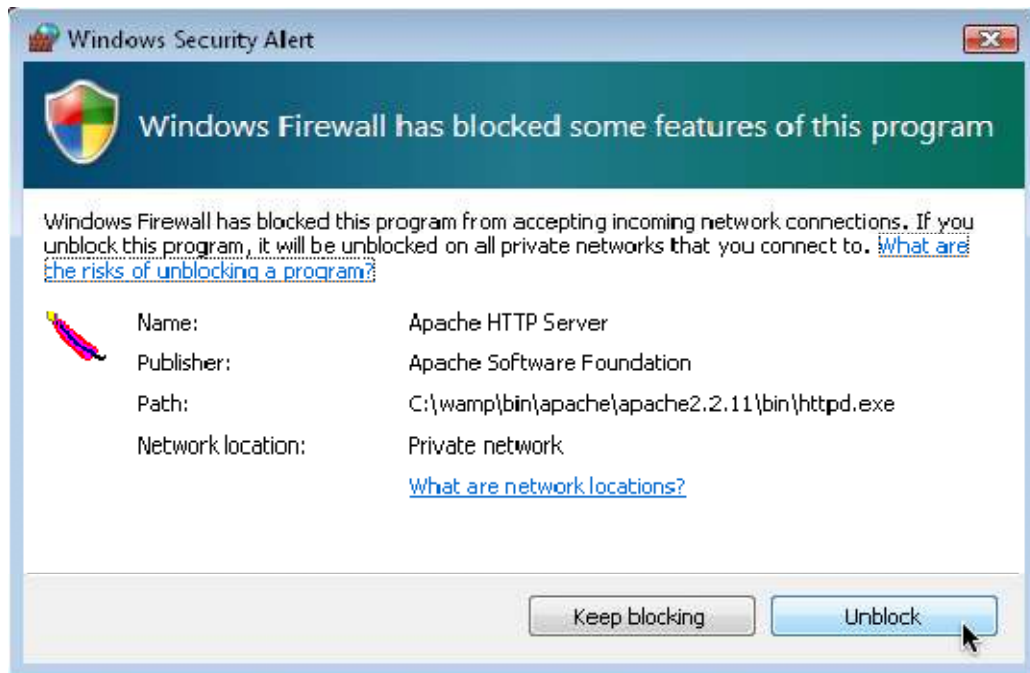
Σχήμα II: Η προεπιλεγμένη θέση εγκατάστασης του WAMP

3. Στο τέλος της εγκατάστασης, ο WampServer θα σας ζητήσει να επιλέξετε τον προεπιλεγμένο browser. Αυτός θα είναι και ο browser που θα ανοίγει όταν ελέγχετε τα project. Αν ο firefox είναι εγκατεστημένος στο σύστημα σας, θα εμφανιστεί η ερώτηση αν επιθυμείτε να τον ορίσετε ως τον default browser. Αν η απάντηση είναι No ή έχετε κάποιον άλλον browser εγκατεστημένο, τότε θα σας ζητηθεί να επιλέξετε ποιο θα είναι το εκτελέσιμο αρχείο του browser που επιθυμείτε να χρησιμοποιήσετε. Στο σχήμα III επιλέγεται ο Internet explorer (explorer.exe). Εναλλακτικά, μπορείτε να επιλέξετε τον Safari ή Opera.



Σχήμα III: Επιλογή προεπιλεγμένου browser

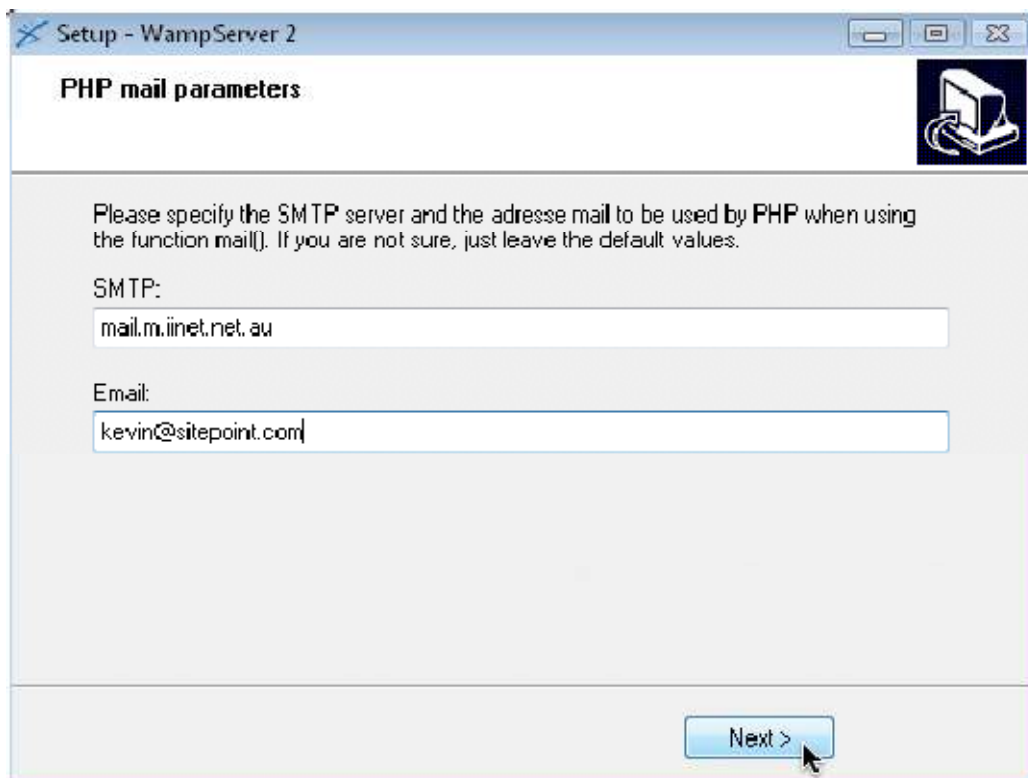
4. Μόλις ο WampServer εγκατασταθεί, θα εμφανιστεί το εικονίδιο του Apache HTTP Server. Τα Windows θα εμφανίσουν ένα μήνυμα ασφαλείας, όπως αυτό που φαίνεται στο σχήμα IV, αφού ο web server θα προσπαθήσει να έχει πρόσβαση στις αιτήσεις που κάνει ο browser.



Σχήμα IV: Μήνυμα ασφαλείας των Windows

Αν επιθυμείτε να είστε εντελώς σίγουροι ότι ο Apache θα απορρίπτει κάθε σύνδεση από το εξωτερικό περιβάλλον και ότι ο μόνος Web browser που θα εκτελείτε στον υπολογιστή σας θα μπορεί να προσπελαίνει σελίδες στον development server, τότε επιλέξτε “Keep blocking”. Ο WampServer έχει ενσωματωμένη την επιλογή να απορρίπτει τις συνδέσεις από το εξωτερικό περιβάλλον αν το επιθυμείτε, όμως συστήνεται να επιλεγεί το “unblock” ώστε να υπάρχει η ευελιξία της πρόσβασης στο development server.

5. Έπειτα, όπως φαίνεται στο σχήμα V, το πρόγραμμα εγκατάστασης του WampServer, θα ζητήσει τον SMTP server και τη διεύθυνση email. Αυτό γίνεται επειδή ένα PHP script έχει τη δυνατότητα αποστολής μηνυμάτων email και από αυτές τις ρυθμίσεις εξαρτώνται ο server αποστολής email αλλά και το προεπιλεγμένο πεδίο “from” του email. Εισάγετε την διεύθυνση email που επιθυμείτε και αν μπορείτε εισάγετε επίσης την διεύθυνση Service Provider’s SMTP. Πάντως, σε αυτό το σημείο μπορείτε να αφήσετε την προεπιλεγμένη τιμή και να τη θέτετε μέσα στο PHP script όταν κάτι τέτοιο απαιτείται.



Σχήμα V: Εισάγετε την διεύθυνση Service Provider's SMTP

Μόλις η εγκατάσταση ολοκληρωθεί μπορείτε να τρέξετε τον WampServer. Ένα εικονίδιο θα εμφανιστεί στο System Tray των windows. Κάντε κλικ σε αυτό και θα εμφανιστεί το μενου του WampServer, το οποίο φαίνεται στο σχήμα VI.



Σχήμα VI: Το μενού του WampServer

Απο προεπιλογή, ο server θα είναι προσβάσιμος από browser που τρέχουν μόνο στον δικό σας υπολογιστή. Αν κάνετε όμως κλικ στην επιλογή του menu “Put Online”, τότε ο server θα είναι προσβάσιμος από browser του εξωτερικού περιβάλλοντος.

Για να ελέγξετε αν ο WampServer λειτουργεί σωστά, κάντε κλικ στην επιλογή “localhost”, η οποία θα βρίσκεται στην κορυφή του menu του WampServer. Ο browser που θα ανοίξει πρέπει να εμφανίσει μια αρχική σελίδα όπως αυτή που φαίνεται στο σχήμα VII.



Σχήμα VII: Αρχική σελίδα του WampServer

Μόλις τελιώσετε με την εργασία σας στον WampServer, μπορείτε να τον κλείσετε κάνοντας δεξιά κλικ στο System Tray εικονίδιο και επιλέγοντας “exit”. Για να λειτουργήσει η MySQL σωστά, πρέπει να προστεθεί στο path των windows. Για την προσθήκη της MySQL command prompt θα πρέπει να ακολουθήσετε τα παρακάτω βήματα:

1. Ανοίγετε το Windows Control Panel. Εντοπίστε και κάντε διπλό κλικ στο εικονίδιο “System”.
2. Ακολουθήστε τα ακόλουθα βήματα για την έκδοση των Windows που χρησιμοποιείτε.
 - Στα Windows XP, πηγαίνετε στον Advanced tab στο παράθυρο “System properties”.
 - Στα Windows Vista ή στα Windows 7, κάντε κλικ στον σύνδεσμο “Advanced”.

3. Κάντε κλικ στο κουμπί “Environment Variables”.
4. Στη λίστα “User”, εντοπίστε την μεταβλητή με το όνομα “PATH”.
 - Αν υπάρχει τότε επιλέξτε το και κάντε κλικ στο κουμπί “Edit”.
 - Αν δεν υπάρχει αυτή η μεταβλητή, κάντε κλικ στο κουμπί “New” και εισάγετε το όνομα της μεταβλητής πληκτρολογώντας την λέξη “PATH”.
5. Προσθέστε το path στον κατάλογο directory2 WampServer της MySQL με την τιμή “Variable”:
 - Αν η τιμή “Variable” είναι κενή τότε εισάγετε το path.
 - Αν η υπάρχει ήδη κείμενο στο πεδίο τιμής “Variable”, προσθέστε ένα ελληνικό ερωτηματικό (;) στο τέλος της τιμής και έπειτα εισάγετε το path.
6. Κάντε κλικ στο κουμπί “Ok” σε όλα τα επομενα παράθυρα γαι να εφαρμοστούν οι αλλαγές σας.

Σε περίπτωση που υπάρξει κάποιο πρόβλημα με τον κωδικό του phpmysqladmin, θα χρειαστεί να καταχωρήσετε το username και το password της βάσης δεδομένων που χρησιμοποιείτε στο αρχείο C:\wamp\apps\phpmysqladmin3.2.0.1/config.inc.php.

ΠΑΡΑΡΤΗΜΑ Β - Εγχειρίδιο χρήσης εφαρμογής

Εγκατάσταση εφαρμογής

Η εφαρμογή αποτελείται από το αρχείο *Lamprou042487-source_code.rar*. Είναι συμπιεσμένο αρχείο που περιέχει τα ρηρ source files, την βάση δεδομένων, τα αρχεία css και ότι απαιτείται για την λειτουργία της εφαρμογής.

Για την εγκατάστασή της, αρκεί η αποσυμπίεση του αρχείου *Lamprou042487-source_code.rar* στον φάκελο *C:\wamp\www* του wamp. Πιό αναλυτικά το cd εγκατάστασης αποτελείται από:

- Τον κατάλογο “Setup” ο οποίος περιέχει το συμπιεσμένο αρχείο *Lamprou042487-source_code.rar* που είναι ο κώδικας της εφαρμογής και τον κατάλογο «Λογισμικό» που περιέχει το απαιτούμενο λογισμικό για να λειτουργήσει η εφαρμογή (Η εγκατάστασή του, αναλύεται στο Παράρτημα Α της τεκμηρίωσης).Ο φάκελος Setup δηλαδή, περιέχει ότι χρειάζεται για την λειτουργία της εφαρμογής.
- Το αρχείο «Πτυχιακή-ΛάμπρουΑριστείδης-042487.pdf» που είναι η τεκμηρίωση της εφαρμογής.
- Το αρχείο «Αριστείδης-Λάμπρου-Στοιχεία.txt» που περιέχει το ονοματεπώνυμο του φοιτητή, το θέμα της πτυχιακής και το ονοματεπώνυμο του καθηγητή που την επιβλέπει.

Σημαντικό: Για να λειτουργήσει η εφαρμογή θα πρέπει, αφού ανοίξετε τον WampServer να επιλέξετε από το system tray των windows την επιλογή ρηρMyAdmin (Παράρτημα Α-Σελίδα 111-Σχήμα VI). Στην σελίδα που θα ανοίξει, δημιουργήστε μια νέα βάση δεδομένων με όνομα database όπως φαίνεται στο Σχήμα I:



Σχήμα I: Δημιουργία βάσης δεδομένων στο ρηrmyadmin

Αφου φτιάξετε την βάση δεδομένων, κάντε import το αρχείο *database.txt* που περιέχει όλους τους πίνακες της εφαρμογής και βρίσκεται μέσα στον κατάλογο *Setup/Lamprou042487-source_code.rar* του cd. Αφού ολοκληρώσετε όλη την παραπάνω διαδικασία εφαρμογή θα είναι έτοιμη προς χρήση.

Εγχειρίδιο χρήσης

Είσοδος στην εφαρμογή

Αν είστε ο διαχειριστής της εφαρμογής, τότε για να εισέλθετε στην σελίδα διαχείρισης, πληκτρολογήστε στο URL την

διεύθυνση, <http://localhost/index.php?mod=domyadmin>, ενώ αν είστε φοιτητής την διεύθυνση <http://localhost/index.php?mod=home> .

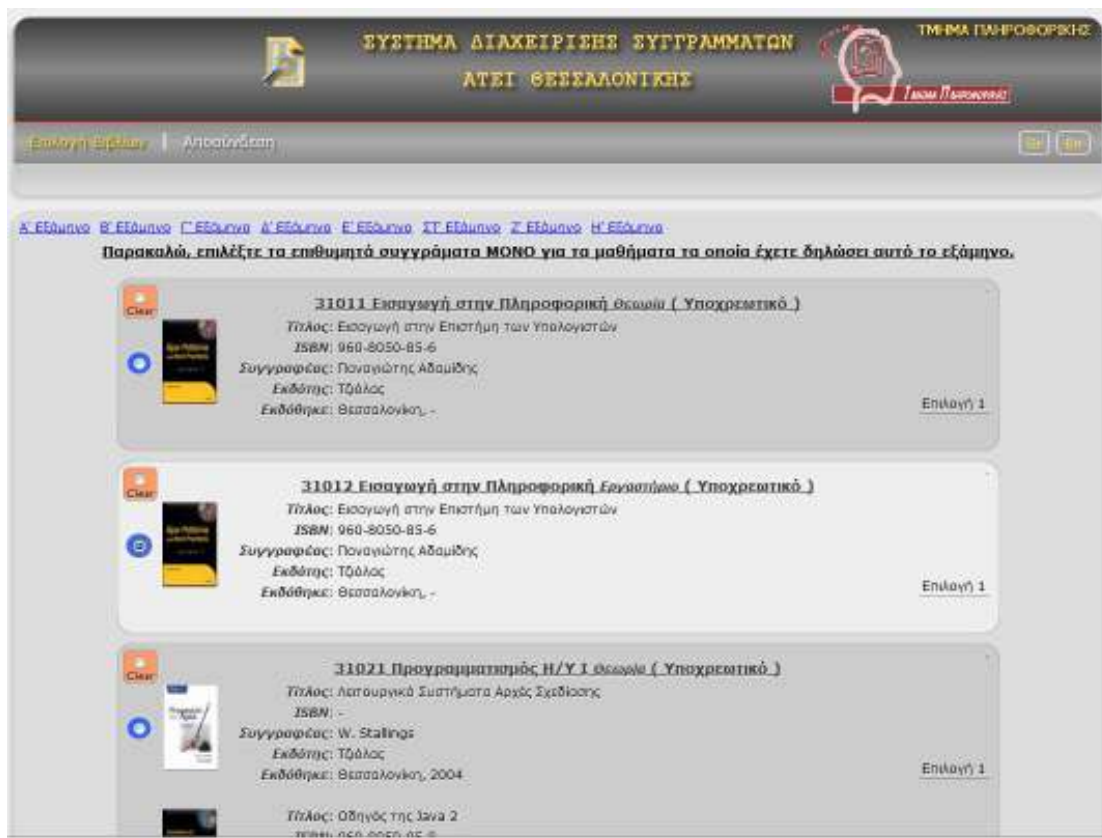
Για να συνδεθείτε στην εφαρμογή εισάγετε το Όνομα Χρήστη (Username) και τον Κωδικό Πρόσβασης (Password) στα αντίστοιχα πεδία της αρχικής σελίδας και πατήστε το κουμπί **Είσοδος** όπως φαίνεται στο σχήμα II. Το δοκιμαστικό username και password του διαχειριστή, είναι **admin** και **11111111**. Η εφαρμογή θα σας κατευθύνει στην ανάλογη σελίδα ανάλογα με το αν είστε ο διαχειριστής του συστήματος ή ο φοιτητής.

The screenshot shows the login interface of the 'ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΣΥΓΓΡΑΜΜΑΤΩΝ ΑΤΕΙ ΘΕΣΣΑΛΟΝΙΚΗΣ'. At the top, there is a header with the system name and the logo of the institution. Below the header, the text 'Επιλογή βιβλίων' is displayed. The main area contains two input fields: 'ΑΔ:' and 'Password:'. Below these fields is a button labeled 'Είσοδος'. On the left side, there is a small box with the text 'Η διαδικασία επιλογής είναι αντιστική'. At the bottom, there is a footer with copyright information: 'Copyright © ΑΤΕΙ Θεσσαλονίκης 2010. Γνωστική εργασία του Διονυσίου Λάμπρου υπό την επίβλεψη του καθηγητή Αγγελου Σαραδάκη.'

Σχήμα II:Είσοδος στην εφαρμογή

Αρχική σελίδα φοιτητή

Μετά την είσοδό σας στην εφαρμογή, προβάλλεται η αρχική σελίδα που αφορά τις δηλώσεις των βιβλίων (Σχήμα III) ,αν συνδεθείτε ως φοιτητής.



Σχήμα III:Δηλώσεις βιβλίων

Επιλέγοντας το εξάμηνο που επιθυμείτε μπορείτε να επιλέξετε βιβλία για όλα τα δηλωμένα μαθήματα.Για κάθε μάθημα μπορεί να υπάρχουν παραπάνω απο ένα βιβλία.Εχετε τη δυνατότητα να επιλέξετε ένα για κάθε μάθημα.Επιλέγετε τα βιβλία που θέλετε απλά τσεκάροντας στο αντίστοιχο radio button και ακυρώνετε την

επιλογή σας τσεκάροντας το radio button clear .Όταν ολοκληρώσετε την διαδικασία επιλογής βιβλίων, πατήστε το κουμπί **Επιλογή Βιβλίων** και οι προτιμήσεις σας θα αποθηκευτούν στην βάση δεδομένων.Όσο η διαδικασία επιλογής βιβλίων είναι ανοιχτή μπορείτε να αλλάξετε τις επιλογές σας.Μετά απο την είσοδό σας στην εφαρμογή μπορείτε να δείτε ποια βιβλία είχατε επιλέξει στην πρώτη σας δήλωση και να τα αλλάξετε.Οι επιλογές που κάνετε μένουν στα αντίστοιχα radio button.Αν έχετε ξανα πάρει κάποιο βιβλίο στο παρελθόν, το σύστημα δεν θα σας επιτρέψει να το επιλέξετε και θα είναι απενεργοποιημένο το αντίστοιχο radio button.Τέλος για την έξοδό σας απο το σύστημα, πατήστε το κουμπί «Αποσύνδεση» για μεγαλύτερη ασφάλεια.Το σύστημα θα σας ανακατευθύνει στην κύρια σελίδα ταυτοποίησης των στοιχείων σας. Στην

περίπτωση που είστε ο διαχειριστής της εφαρμογής, τότε αφού συνδεθείτε έχετε τη δυνατότητα διαχείρισης όλων των μονάδων της εφαρμογής.

Διαχείριση μαθημάτων

Απο την βασική σελίδα διαχείρισης επιλέξτε απο το menu μαθήματα την «Εισαγωγή μαθήματος». Θα εμφανιστεί μια φόρμα εισαγωγής των στοιχείων του μαθήματος όπως φαίνεται στο σχήμα IV. Εισάγετε τα στοιχεία για το νέο μάθημα. Προσοχή στην επιλογή θεωρίας ή εργαστηρίου. Επιλέγοντας το εξάμηνο του μαθήματος αυτόματα υπολογίζεται και η θέση του μαθήματος στο εξάμηνο. Αν το εξάμηνο έχει μαθήματα τότε μπορείτε να επιλέξετε εσείς την θέση που επιθυμείτε. Παράδειγμα εστω οτι στο Α εξάμηνο υπάρχουν καταχωρημένα 4 μαθήματα και θέλετε να καταχωρήσετε αλλο ένα. Επιλέγοντας το εξάμηνο η εφαρμογή θα το τοποθετήσει αυτόματα στην 5^η θέση. Αν επιθυμείτε διαφορετική θέση τότε αλλάζετε το πεδίο «Θέση» και επιλέγετε τη νέα θέση του μαθήματος. Αυτή η ιδιότητα χρησιμεύει, γιατί στην περίπτωση κάποιου λάθους στην σειρά των μαθημάτων θα έχετε την δυνατότητα να αναδιατάξετε το περιεχόμενο του πίνακα μαθημάτων και των κωδικών.

The screenshot shows a web interface for creating a new course. The title is "1. Δημιουργία νέου μαθήματος". Below the title, there is a note: "1. Εισάγετε στο σύστημα όλα τα απαραίτητα στοιχεία του νέου μαθήματος." The form contains the following fields:

- Όνομα GR: Λειτουργικό σύστημα I
- Όνομα EN: UNIX
- Εξάμηνο: Ε' Εξάμηνο
- Υποχρεωτικό: Υποχρεωτικό
- Ενότητα: Εκότητα...
- Θέση: Θέση...
- Θεωρία: [x]
- Εργαστήριο: [1]
- ECTS Θεωρίας: 4
- ECTS Εργαστηρίου: ECTS...
- Ωρες Θεωρίας: 3ωρες...
- Ωρες Εργαστηρίου: 3ωρες...

There are two dropdown menus for selecting textbooks:

- Βιβλία Θεωρίας: A list of books including "Αλγόριθμοι σε Java, Μέρη 1-4", "Ανάπτυξη και διακρίση εφαρμογών", "Ανάλυση Επικοινωνία", "Εισαγωγή στην Επιστήμη των Υπολογιστών", "Επισκευή Ανθρώπου - Υπολογιστή", and "Λειτουργικό Σύστημα Αρχές Εκπαίδευσης".
- Βιβλία Εργαστηρίου: A list of books including "Αλγόριθμοι σε Java, Μέρη 1-4", "Ανάπτυξη και διακρίση εφαρμογών", "Ανάλυση Επικοινωνία", "Εισαγωγή στην Επιστήμη των Υπολογιστών", "Επισκευή Ανθρώπου - Υπολογιστή", and "Λειτουργικό Σύστημα Αρχές Εκπαίδευσης".

A "Καταχώρηση" button is located at the bottom center of the form.

Σχήμα IV: Εισαγωγή μαθήματος

Τέλος αφού έχετε εισάγει όλα τα επιθυμητά στοιχεία, επιλέξτε τα βιβλία που αντιστοιχούν στον κάθε τύπο μαθήματος και θα δίνονται στους φοιτητές αν αυτά υπάρχουν και πατήστε το κουμπί **Καταχώρηση**.

Το σύστημα θα αποθηκεύσει το νέο μάθημα στην βάση δεδομένων και θα σας εμφανίσει μήνυμα ολοκλήρωσης.

Απο την επιλογή Αναζήτηση μαθήματος μπορείτε να αναζητήσετε τα μαθήματα που σας ενδιαφέρουν σύμφωνα με τα κριτήρια που επιθυμείτε (Σχήμα V).

1. Αναζήτηση Μαθήματος

Εισάγετε τα κριτήρια προς αναζήτηση για τα μαθήματα που επιθυμείτε να βρείτε.

Κωδικός:

Εξάμηνο:

Υποχρεωτικό:

Ενότητα:

Θ | Ε:

Ενεργό:

Αναζήτηση

Βρέθηκαν 11 αποτελέσματα σύμφωνα με τα κριτήρια αναζήτησης που τέθηκαν.

Κωδ.	Τίτλος	Εξάμηνο	Τύπος	Α.Δ.Μ.	Στες Διδασκ.	Κατηγορία Μαθήματος	Ενεργό
<input type="checkbox"/> 31011	Εισαγωγή στην Πληροφορική <i>Βιβλίο: Εισαγωγή στην Επιστήμη των Υπολογιστών</i>	A	Θεωρία	4	4	Υποχρεωτικό ΓΥ	no <input type="checkbox"/>
<input type="checkbox"/> 31012	Εισαγωγή στην Πληροφορική <i>Βιβλίο: Εισαγωγή στην Επιστήμη των Υπολογιστών</i>	A	Εργαστήριο	2	2	Υποχρεωτικό ΓΥ	no <input type="checkbox"/>
<input checked="" type="checkbox"/> 31021	Προγραμματισμός ΗΥ I <i>Βιβλίο: Οδηγός της Άσκας 2, Αλγόριθμοι σε Άσκας Μέρος 1-4, Λειτουργικό Σύστημα Άσκας Σχεδίασης</i>	A	Θεωρία	4	4	Υποχρεωτικό ΓΥ	yes <input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> 31022	Προγραμματισμός ΗΥ I <i>Βιβλίο: Οδηγός της Άσκας 2, Αλγόριθμοι σε Άσκας Μέρος 1-4, Λειτουργικό Σύστημα Άσκας Σχεδίασης</i>	A	Εργαστήριο	2	2	Υποχρεωτικό ΓΥ	yes <input checked="" type="checkbox"/>

Ενεργοποίηση / Απενεργοποίηση

Σχήμα V:Αναζήτηση μαθήματος


Αν δεν εισάγετε κριτήρια το σύστημα θα εμφανίσει όλα τα διαθέσιμα μαθήματα και τα χαρακτηριστικά τους.Κάτω απο κάθε μάθημα θα αναγράφεται και το αντίστοιχο βιβλίο που διανέμεται γι αυτό και είναι το βιβλίο που είχατε αντιστοιχήσει παραπάνω στην εισαγωγή του μαθήματος.Επιλέγοντας το checkbox του μαθήματος που θέλετε να απενεργοποιήσετε και πατώντας το κουμπί

Ενεργοποίηση / Απενεργοποίηση

το σύστημα απενεργοποιεί το επιλεγμένο μάθημα.Προσοχή ΔΕΝ το διαγράφει.Σε περίπτωση που επιθυμείτε ενεργοποίηση επιλέγετε ξανά το απενεργοποιημένο μάθημα και πατήστε το κουμπί

Ενεργοποίηση / Απενεργοποίηση

για να το ενεργοποιήσετε.Η πληροφορία «Ενεργό» στην στήλη μαθημάτων δείχνει εάν ένα μάθημα είναι ενεργό ή

ανενεργό.Πατώντας το κουμπί  επεξεργασία μπορείτε να επεξεργαστείτε τα στοιχεία ενός μαθήματος.

Διαχείριση βιβλίων

Επιλέγοντας απο το menu την «Καταχώρηση βιβλίου» μπορείτε να καταχωρήσετε ένα νέο βιβλιο στην βάση δεδομένων.Εμφανίζεται μια φόρμα εισαγωγής των στοιχείων του βιβλίου (Σχήμα VI)



Σχήμα VI:Εισαγωγή νέου βιβλίου

Πληκτρολογείτε τον τίτλο του βιβλίου το ISBN κτλ,Επιλέγετε τον συγγραφέα και την έκδοση του βιβλίου(Προσοχή:Αν ο συγγραφέας ή η έκδοση δεν περιέχεται στην λίστα τότε μπορείτε να εισάγετε τον δικό σας απλώς πληκτρολογώντας το εναλλακτικό πεδίο που βρίσκεται ακριβώς απο κάτω),και τέλος αν επιθυμείτε ανεβάζετε και φωτογραφία του εξώφυλλου του βιβλίου απο το πεδίο «Εξώφυλλο».Αφου ολοκληρώσετε την διαδικασία εισαγωγής πατήστε το κουμπί **Καταχώρηση** και το νέο σύγγραμμα θα καταχωρηθεί στην βάση δεδομένων.

Απο την επιλογή «Αναζήτηση βιβλίου» μπορείτε να αναζητήσετε τα βιβλία που επιθυμείτε απλώς εισάγωντας τα επιθυμητά κριτήρια και πατώντας το κουμπί **Αναζήτηση** όπως φαίνεται στο σχήμα VII.

1. Αναζήτηση Βιβλίου

ι. Εισάγετε τα **κριτήρια προς αναζήτηση** για το βιβλίο που επιθυμείτε να βρείτε.

Τίτλος GR:

ISBN: ΦΕΚ:

Συγγραφέας: Εκδόσεις:

Ενεργά:


Σχήμα VII:Αναζήτηση βιβλίου

Το σύστημα θα εμφανίσει όλα τα διαθέσιμα βιβλία σύμφωνα με τα κριτήρια που εισάγατε (Σχήμα VIII).Επιλέγοντας τα βιβλία και πατώντας το κουμπί **Ενεργοποίηση / Απενεργοποίηση** μπορείτε να ενεργοποιήσετε/απενεργοποιήσετε ένα βιβλίο.Προσοχή τα απενεργοποιημένα βιβλία δεν θα είναι ορατά στους φοιτητές κατά την φάση των δηλώσεων αφού είναι εικονικά διεγραμμένα.

<input type="checkbox"/>		Ανάπτυξη και Διαχείριση εφαρμογών	160-8050-85-5	Mikal	Κλαδάριβμος	2008	Αθήνα	910/Γ/2008	48	no	
Μαθήματα: Ανάπτυξη και Διαχείριση εφαρμογών Β, 7 Εξάμηνο											
<input type="checkbox"/>		Ανθρώπινη Επικοινωνία	010-8050-85-6	Βασίλειος Αντώνιος	Ζήτη	2008	Λάρισα	928/Β/2010	20,00	no	
Μαθήματα: Μαθηματικά Ανάλυσης Β, Α Εξάμηνο , Ανάλυση Ι Β, Α Εξάμηνο , Ανάλυση Ι Ε, Α Εξάμηνο , Διδάκτες Επικοινωνίας Β, Α Εξάμηνο , Διδάκτες Επικοινωνίας Ε, Α Εξάμηνο											
<input type="checkbox"/>		Εισαγωγή στην Επιστήμη των Υπολογιστών	960-8050-85-6	Παναγιώτης Αδαμίδης	Τζάλος	-	Βασιλειάδης	916/Β/2008	4,00	yes	
Μαθήματα: Ψηφιακό Συστήματα Β, Α Εξάμηνο , Ψηφιακό Συστήματα Ε, Α Εξάμηνο , Εισαγωγή στην Πληροφορική Β, Α Εξάμηνο , Εισαγωγή στην Πληροφορική Ε, Α Εξάμηνο , Οργάνωση και Αρχιτεκτονική Υπολ. Συστημάτων Β, Β Εξάμηνο , Οργάνωση και Αρχιτεκτονική Υπολ. Συστημάτων Ε, Β Εξάμηνο											

Σχήμα VIII:Αποτέλεσμα αναζήτησης βιβλίων

Κάτω απο κάθε σύγγραμμα εμφανίζονται και τα αντίστοιχα μαθήματα στα οποία δίνεται.Πατώντας σε ένα μάθημα μεταφερόμαστε στην σελίδα επεξεργασίας του

μαθήματος.Πατώντας την επεξεργασία  δίπλα απο κάθε βιβλίο, επεξεργαζόμαστε το αντίστοιχο βιβλίο.Εμφανίζεται μια οθόνη με όλα τα στοιχεία του επιλεγμένου βιβλίου (Σχήμα ΙΧ).

Σχήμα ΙΧ:Επεξεργασία βιβλίου

Αφου ολοκληρώσουμε την επεξεργασία του συγγράμματος πατάμε το κουμπί **Επεξεργασία** και ενημερώνεται η βάση δεδομένων.

Καταχώρηση φοιτητή

Δοκιμαστικά όπως προαναφέρα στο κύριο μέρος της τεκμηρίωσης υπάρχει η δυνατότητα καταχώρησης φοιτητών.Επιλέγοντας απο το menu την «Καταχώρηση φοιτητών» μπορούμε να εισάγουμε έναν νέο φοιτητή στο σύστημα όπως φαίνεται στο σχήμα Χ πατώντας το κουμπί **Καταχώρηση**.

ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΣΥΓΓΡΑΜΜΑΤΩΝ
ΑΤΕΙ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ Γ.Π.Φ.ΘΕΣΣΑΛΟΝΙΚΗΣ
ΑΝΩΝΥΜΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Καθήματα | Βιβλία | **Φοιτητές** | Διαχειριστές | Αναρτήσεις

1. Δημιουργία νέου φοιτητή

1. Εισάγετε στο σύστημα όλα τα απαραίτητα στοιχεία του νέου φοιτητή.

Αντώνης Βαρελόδης
Η διαδικασία επίλογής είναι αναγκαία
[Κλείσιμο Διαδικασίας](#)

Όνομα*: Επώνυμο*:
ΑΜ*: Κωδικός*:
Χρόνος Εισαγωγής: Χρόνος... Εξάμηνα Εισαγωγής: Εξάμηνα...
Σταθρό: Κινητό:
Email:

Copyright © ΑΤΕΙ Θεσσαλονίκης 2010. Πτυχιακή εργασία του Αριστείδη Λάμπρου υπό την εποπτεία του καθηγητή Αντώνη Βαρελόδη.

Σχήμα Χ:Εισαγωγή νέου φοιτητή

Διαχείριση διαχειριστών

Απο την επιλογή «Εγγραφή διαχειριστή» που θα βρείτε στο μενυ επιλογών μπορείτε εύκολα να εγγράψετε έναν νέο διαχειριστή στο σύστημα.Για να γίνει αυτο θα πρέπει να έχετε ήδη λογαριασμό ως διαχειριστής για να εγγράψετε έναν δεύτερο διαχειριστή.Εισάγοντας όλα τα απαραίτητα στοιχεία για τον νέο διαχειριστή στην φόρμα εισαγωγής που θα εμφανιστεί ,όπως φαίνεται στο σχήμα ΧΙ και στη συνέχεια πατώντας το κουμπί το σύστημα αποθηκεύει τα στοιχεία του στην βάση δεδομένων.

The screenshot shows the registration page for a new administrator. The page title is 'ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΣΥΓΓΡΑΜΜΑΤΩΝ ΑΤΕΙ ΘΕΣΣΑΛΟΝΙΚΗΣ'. The navigation menu includes 'Επιβάματα', 'Επίδομα', 'Φοιτητές', 'Διαχειριστές', and 'Αποσύνδεση'. The main heading is '1. Δημιουργία νέου διαχειριστή'. Below it, the instruction reads: '1. Εισάγετε στο σύστημα όλα τα απαραίτητα στοιχεία του νέου διαχειριστή:'. The form contains the following fields: 'Όνομα*', 'Επίθετο*', 'Username*', 'Κωδικός*', 'Σταθιρό*', 'Κινητό*', and 'Email'. A 'Καταχώριση' button is located at the bottom of the form. On the left side, there is a sidebar with links: 'Αντώνης Βασιλάκης', 'Η διαδικασία επιλογής έχει κλείσει !!', 'Αναζητήστε Διαχειριστή', and 'Επιστρέψτε στην Σελίδα Αρχής'.

Copyright © ΑΤΕΙ Θεσσαλονίκης 2010. Πτυχιακή εργασία του Αριστείδη Λάμπρου υπό την επίσημη καθοδήγηση Αντώνη Βασιλάκη.

Σχήμα XI:Εγγραφή νέου διαχειριστή




Απο την επιλογή «Αναζήτηση Διαχειριστή» μπορείτε να αναζητήσετε κάποιον διαχειριστή με τα κριτήρια αναζήτησης που επιθυμείτε πατώντας το κουμπί **Αναζήτηση**. Το σύστημα θα εμφανίσει τους διαθέσιμους διαχειριστές όπως φαίνεται στο σχήμα XII.

The screenshot shows the search page for an administrator. The page title is 'ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΣΥΓΓΡΑΜΜΑΤΩΝ ΑΤΕΙ ΘΕΣΣΑΛΟΝΙΚΗΣ'. The navigation menu is the same as in the previous screenshot. The main heading is '1. Αναζήτηση Διαχειριστή'. Below it, the instruction reads: '1. Εισάγετε τα κριτήρια προς αναζήτηση για το διαχειριστή που επιθυμείτε να βρείτε:'. The form contains the following fields: 'Όνομα: Αντ', 'Επίθετο: Βασ', and 'Κινητό:'. A 'Αναζήτηση' button is located at the bottom of the form. Below the search form, there is a table with the following data: 'Διαχειριστής: Αντώνης Βασιλάκης', 'Επικοινωνία: Τηλ: 6011111111, 2311111111, vaf@teko.gr.', and 'Last Access: 2010-08-10 16:23:16'. A 'Δεγγραφή' button is located at the bottom of the page.

Σχήμα XII:Αναζήτηση διαχειριστή

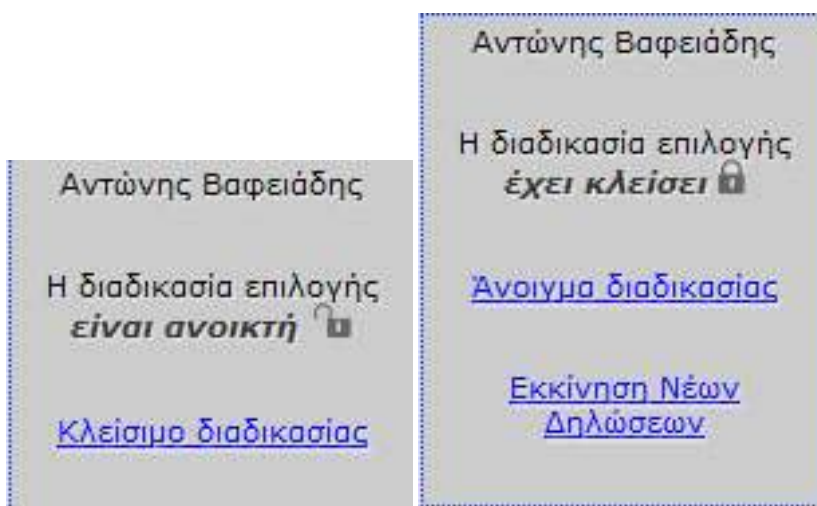
Μπορεί να είναι παραπάνω απο μία σελίδα ανάλογα με τα κριτήρια που δώσατε.Αν δεν εισάγετε κανένα κριτήριο προς αναζήτηση η εφαρμογή θα εμφανίσει όλους τους εγγεγραμμένους διαχειριστές του συστήματος.

Για να επεξεργαστείτε κάποιον διαχειριστή πατήστε το κουμπί «Επεξεργασία» που

βρίσκεται στα δεξιά .Θα εμφανιστεί μια φόρμα επεξεργασίας με τα στοιχεία του διαχειριστή που επιλέξατε να επεξεργαστείτε.Μόλις ολοκληρώσετε την διαδικασία πατήστε το κουμπί . Για να διαγράψετε έναν διαχειριστή απλώς τσεκάρετε στο αντίστοιχο checkbox και πατήστε το κουμπί .Προσοχή οι διαχειριστές διαγράφονται απο το σύστημα ΔΕΝ απενεργοποιούνται.Δεν μπορείτε να διαγράψετε τον εαυτό σας.

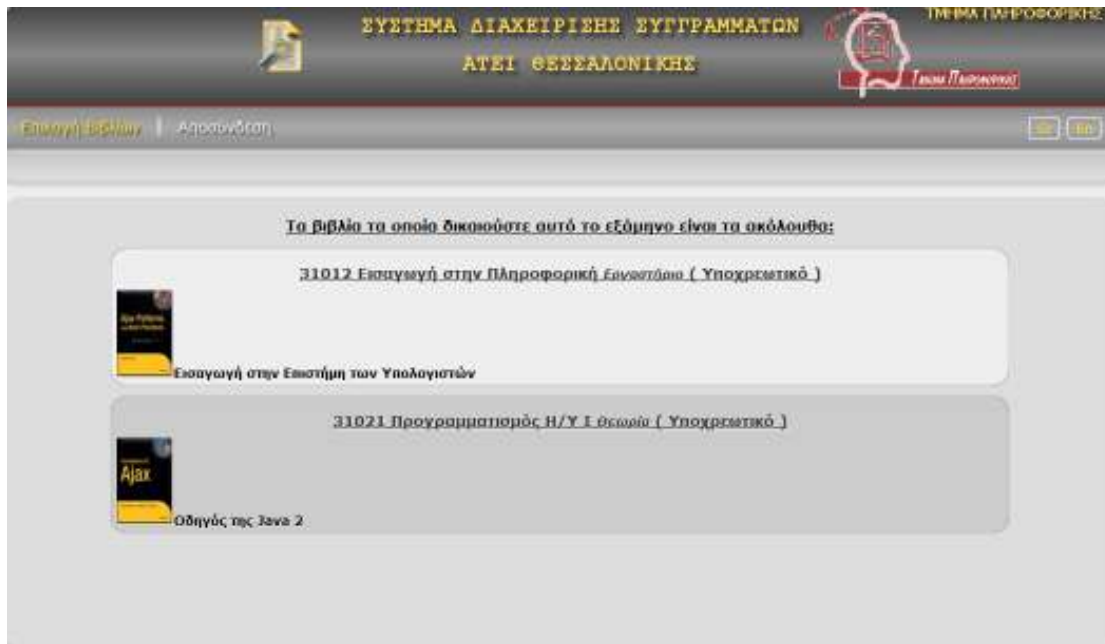
Ανοιγμα/Κλείσιμο της διαδικασίας δηλώσεων

Απο την επιλογή «Κλείσιμο διαδικασίας» (Σχήμα XIII) μπορείτε να κλείσετε την διαδικασία των δηλώσεων,ενώ απο την επιλογή «Ανοιγμα διαδικασίας» μπορείτε να ανοίξετε την διαδικασία των δηλώσεων.



Σχήμα XIII:Ανοιγμα/Κλείσιμο δηλώσεων-Εκκίνηση νέων δηλώσεων

Αν κλείσετε την διαδικασία των ηλεκτρονικών δηλώσεων,οι φοιτητές δεν θα μπορούν πλέον να δηλώνουν τα συγγράμματα που επιθυμούν,αλλά θα μπορούν να παρακολουθούν ποιά βιβλία τους έχουν ανατεθεί όπως φαίνεται στο σχήμα XIV.



Σχήμα XIV:Αποτέλεσμα μετά το κλείσιμο της διαδικασίας δηλώσεων.Τα βιβλία που δικαιούνται οι φοιτητές

Τέλος αν πατήσετε το κουμπί «Εκκίνηση νέων δηλώσεων» (Σχήμα XIII-Δεξιά) γίνεται επανεκκίνηση των ηλεκτρονικών δηλώσεων.Το κουμπί αυτό ουσιαστικά κρατά όλα τα στοιχεία με τις προτιμήσεις των φοιτητών απο τις προηγούμενες δηλώσεις σε έναν πίνακα για ιστορικά στοιχεία.Καθαρίζει τις τρέχουσες δηλώσεις δηλαδή και κρατά στοιχεία παλαιότερων δηλώσεων.Συνήθως την επιλογή αυτή την χρησιμοποιούμε σε κάθε νεα έναρξη των ηλεκτρονικών δηλώσεων των συγγραμμάτων.Για να αποσυνδεθείτε απο την εφαρμογή πατήστε το κουμπί «Αποσύνδεση» για μεγαλύτερη ασφάλεια και το σύστημα θα σας ανακατευθύνει στην σελίδα ταυτοποίησης των στοιχείων του φοιτητή.

ΠΑΡΑΡΤΗΜΑ Γ - Κώδικας SQL για την δημιουργία των πινάκων

Η εφαρμογή αποτελείται από 18 πίνακες που περιέχονται σε μια βάση δεδομένων(database) με όνομα database. Η δημιουργία όλων των πινάκων της εφαρμογής έγινε με τον SQL κώδικα που δίνεται παρακάτω.

```
Database: `database`
```

Table structure for table `admins`

```
CREATE TABLE IF NOT EXISTS `admins` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(40) collate utf8_unicode_ci NOT NULL,  
  `surname` varchar(40) collate utf8_unicode_ci NOT NULL,  
  `username` varchar(20) collate utf8_unicode_ci NOT NULL,  
  `password` varchar(40) collate utf8_unicode_ci NOT NULL,  
  `level` enum('1','2') collate utf8_unicode_ci NOT NULL default '1',  
  `email` varchar(40) collate utf8_unicode_ci NOT NULL,  
  `landphone` char(40) collate utf8_unicode_ci default NULL,  
  `mobile` char(40) collate utf8_unicode_ci default NULL,  
  `active` enum('yes','no') collate utf8_unicode_ci NOT NULL default 'yes',  
  `last_seen` timestamp NOT NULL default CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
AUTO_INCREMENT=7;
```

Table structure for table `assigns_is_continued`

```
CREATE TABLE IF NOT EXISTS `assigns_is_continued` (  
  `id` int(11) NOT NULL auto_increment,  
  `admins_id` int (11) default NULL,  
  `active` enum('yes','no') collate utf8_unicode_ci NOT NULL default 'yes',  
  `has_assigned` enum ('yes','no') collate utf8_unicode_ci NOT NULL default 'no',  
  `date` timestamp NOT NULL default CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
ROW_FORMAT=COMPACT AUTO_INCREMENT=2 ;
```

Table structure for table `authors`

```
CREATE TABLE IF NOT EXISTS `authors` (  
  `id` int(11) NOT NULL auto_increment,  
  `name_gr` varchar (40) collate utf8_unicode_ci NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
AUTO_INCREMENT=6 ;
```

Table structure for table `books`

```
CREATE TABLE IF NOT EXISTS `books` (  
  `id` int(11) NOT NULL auto_increment,  
  `name_gr` varchar (100) collate utf8_unicode_ci default NULL,  
  `name_en` varchar (100) collate utf8_unicode_ci default NULL,  
  `isbn` varchar(20) collate utf8_unicode_ci default NULL,  
  `authors_id` int (11) default NULL,  
  `publishers_id` int (11) default NULL,  
  `publication_year` char (4) collate utf8_unicode_ci default NULL,  
  `publication_places_id` int(11) default NULL,  
  `fek` varchar(40) collate utf8_unicode_ci default NULL,  
  `has_images` enum ('yes','no') collate utf8_unicode_ci default 'no',  
  `price` varchar(6) collate utf8_unicode_ci default '14.50',  
  `total_price` decimal (6,2) default '0.00',  
  `active` enum('yes','no') collate utf8_unicode_ci default 'yes',  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
ROW_FORMAT=COMPACT AUTO_INCREMENT=9 ;
```

Table structure for table `books_images`

```
CREATE TABLE IF NOT EXISTS `books_images` (  
  `books_id` int (11) NOT NULL,  
  `type` char(4) collate utf8_unicode_ci default NULL,  
  `size` int(11) default NULL,  
  PRIMARY KEY (`books_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
ROW_FORMAT=COMPACT;
```

Table structure for table `books_to_subjects_proposed`

```
CREATE TABLE IF NOT EXISTS `books_to_subjects_proposed` (  
  `id` int(11) NOT NULL auto_increment,  
  `books_id` int (11) NOT NULL,  
  `subjects_description_id` int(11) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
ROW_FORMAT=COMPACT AUTO_INCREMENT=132 ;
```

Table structure for table `publication_places`

```
CREATE TABLE IF NOT EXISTS `publication_places` (  
  `id` int(11) NOT NULL auto_increment,  
  `name_gr` varchar (40) collate utf8_unicode_ci default NULL,  
  `name_en` varchar (40) collate utf8_unicode_ci default NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
AUTO_INCREMENT=4 ;
```

Table structure for table `publishers`

```
CREATE TABLE IF NOT EXISTS `publishers` (  
  `id` int(11) NOT NULL auto_increment,  
  `name_gr` varchar(40) collate utf8_unicode_ci NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
AUTO_INCREMENT=4 ;
```

Table structure for table `schools`

```
CREATE TABLE IF NOT EXISTS `schools` (  
  `id` int(11) NOT NULL auto_increment,  
  `code` int(11) default NULL,  
  `name_gr` varchar (40) collate utf8_unicode_ci NOT NULL,  
  `name_en` varchar (40) collate utf8_unicode_ci NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
AUTO_INCREMENT=2;
```


Table structure for table `semesters`

```
CREATE TABLE IF NOT EXISTS `semesters` (  
  `id` int(11) NOT NULL auto_increment,  
  `name_gr` varchar(40) collate utf8_unicode_ci NOT NULL,  
  `name_en` varchar(40) collate utf8_unicode_ci NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
AUTO_INCREMENT=9 ;
```

Table structure for table `students`

```
CREATE TABLE IF NOT EXISTS `students` (  
  `id` int(11) NOT NULL auto_increment,  
  `reg_num` varchar(8) collate utf8_unicode_ci NOT NULL,  
  `name` varchar(40) collate utf8_unicode_ci NOT NULL,  
  `surname` varchar(40) collate utf8_unicode_ci NOT NULL,  
  `password` varchar(40) collate utf8_unicode_ci NOT NULL,  
  `email` varchar(40) collate utf8_unicode_ci default NULL,  
  `landphone` char(10) collate utf8_unicode_ci default NULL,  
  `mobile` char(10) collate utf8_unicode_ci default NULL,  
  `year_enrolled` char(4) collate utf8_unicode_ci default NULL,  
  `semester_enrolled` enum('1','2') collate utf8_unicode_ci default '1',  
  `active` enum('yes','no') collate utf8_unicode_ci NOT NULL default 'yes',  
  `last_seen` timestamp NOT NULL default CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
AUTO_INCREMENT=6 ;
```

Table structure for table `students_to_books_choises`

```
CREATE TABLE IF NOT EXISTS `students_to_books_choises` (  
  `students_id` int(11) NOT NULL,  
  `books_to_subjects_id` int(11) NOT NULL,  
  `date` timestamp NOT NULL default CURRENT_TIMESTAMP,  
  PRIMARY KEY (`students_id`,`books_to_subjects_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

Table structure for table `students_to_books_final`

```
CREATE TABLE IF NOT EXISTS `students_to_books_final` (  
  `students_id` int (11) NOT NULL,  
  `books_to_subjects_id` int (11) NOT NULL,  
  `date` timestamp NOT NULL default CURRENT_TIMESTAMP,  
  PRIMARY KEY (`students_id`,`books_to_subjects_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

Table structure for table `subjects`

```
CREATE TABLE IF NOT EXISTS `subjects` (  
  `id` int(11) NOT NULL auto_increment,  
  `name_gr` varchar (50) collate utf8_unicode_ci NOT NULL,  
  `name_en` varchar (50) collate utf8_unicode_ci NOT NULL,  
  `semesters_id` int (6) NOT NULL,  
  `rank` int(6) default '0',  
  `subjects_categories_id` int (6) NOT NULL,  
  `subjects_fields_id` int (11) default '0',  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
ROW_FORMAT=COMPACT AUTO_INCREMENT=11 ;
```

Table structure for table `subjects_categories`

```
CREATE TABLE IF NOT EXISTS `subjects_categories` (  
  `id` int(11) NOT NULL auto_increment,  
  `name_gr` varchar (40) collate utf8_unicode_ci NOT NULL,  
  `name_en` varchar (40) collate utf8_unicode_ci NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
ROW_FORMAT=COMPACT AUTO_INCREMENT=4 ;
```

Table structure for table `subjects_description`

```
CREATE TABLE IF NOT EXISTS `subjects_description` (  
  `id` int(11) NOT NULL auto_increment,  
  `subjects_id` int (11) NOT NULL,  
  `subjects_types_id` int (11) NOT NULL,  
  `code` varchar(6) collate utf8_unicode_ci default '0',  
  `ects` int(11) default '0',  
  `hours` int(11) default '0',  
  `active` enum('yes','no') collate utf8_unicode_ci NOT NULL default 'yes',  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
ROW_FORMAT=COMPACT AUTO_INCREMENT=18 ;
```

Table structure for table `subjects_fields`

```
CREATE TABLE IF NOT EXISTS `subjects_fields` (  
  `id` int(11) NOT NULL auto_increment,  
  `name_gr` varchar (30) collate utf8_unicode_ci NOT NULL,  
  `name_en` varchar (30) collate utf8_unicode_ci NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
ROW_FORMAT=COMPACT AUTO_INCREMENT=5 ;
```

Table structure for table `subjects_types`

```
CREATE TABLE IF NOT EXISTS `subjects_types` (  
  `id` int(11) NOT NULL auto_increment,  
  `name_gr` varchar (40) collate utf8_unicode_ci NOT NULL,  
  `name_en` varchar (40) collate utf8_unicode_ci NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
ROW_FORMAT=COMPACT AUTO_INCREMENT=3 ;
```