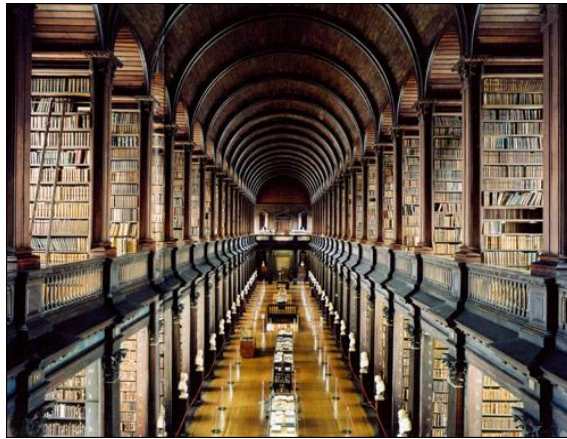


Τεχνολογικό Εκπαιδευτικό Ίδρυμα Πάτρας
Τμήμα Εφαρμογών Πληροφορικής στην Διοίκηση και Οικονομία

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ



Ανάλυση σχεδιασμός και ανάπτυξη μιας web εφαρμογής ψηφιακής βιβλιοθήκης e-library

Analysis, design and development of a web application of an e-library

**Σπουδαστές: Μαυρικάκη Μαρία
Αντωνίου Αντώνης**

Εισηγητής: Χόχολης Διονύσιος

ΑΜΑΛΙΑΔΑ 2010

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΥΧΑΡΙΣΤΙΕΣ.....	5
ΠΕΡΙΛΗΨΗ.....	6
ABSTRACT	7
1 ΕΙΣΑΓΩΓΗ.....	8
1.1 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ.....	8
1.2 ΠΡΟΤΕΙΝΟΜΕΝΗ ΛΥΣΗ.....	10
1.3 ΜΕΘΟΔΟΛΟΓΙΑ	11
1.4 ΠΕΡΙΓΡΑΦΗ ΚΕΦΑΛΑΙΩΝ	12
2 ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ	14
2.1 ΜΟΝΤΕΛΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ.....	14
2.1.1 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΟΥ ΜΟΝΤΕΛΟΥ 16	
2.1.2 ΛΟΓΟΣ ΧΡΗΣΙΜΟΠΟΙΗΣΗΣ ΜΟΝΤΕΛΟΥ ΚΑΤΑΡΡΑΚΤΗ.....	17
2.2 ΔΙΑΔΙΚΤΥΑΚΕΣ ΕΦΑΡΜΟΓΕΣ	17
2.2.1 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ	18
2.2.2 ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ	19
2.2.3 ΑΣΦΑΛΕΙΑ ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ.....	20
2.3 ΕΠΙΘΕΩΡΗΣΗ ΑΣΦΑΛΕΙΑΣ (SECURITY AUDIT):.....	21
2.3.1 ΑΥΤΟ-ΑΞΙΟΛΟΓΗΣΗ ΑΣΦΑΛΕΙΑΣ (SECURITY SELF- ASSESSMENT):	21
2.3.2 ΔΟΚΙΜΗ ΔΙΕΙΣΔΥΣΗΣ (PENETRATION TESTING Η “ETHICAL HACKING”):.....	22
2.4 WEB SERVER	23
2.4.1 ΙΣΤΟΡΙΑ.....	23
2.5 APACHE.....	24
2.5.1 ΙΣΤΟΡΙΑ APACHE SERVER	24
2.5.2 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ APACHE WEB SERVER	25
2.6 CLIENT/SERVER ΑΡΧΙΤΕΚΤΟΝΙΚΗ	26
2.7 HTML	28
2.7.1 ΙΣΤΟΡΙΑ HTML	29

2.7.2	ΔΟΜΗ HTML	30
2.8	PHP	31
2.8.1	ΙΣΤΟΡΙΑ PHP	33
2.9	JAVASCRIPT.....	34
2.9.1	ΙΣΤΟΡΙΑ JAVASCRIPT	34
2.9.2	ΔΕΙΓΜΑ ΚΩΔΙΚΑ JAVASCRIPT.....	35
2.10	ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ MYSQL.....	35
2.10.1	ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ.....	35
2.10.2	ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ	36
2.10.3	ΙΣΤΟΡΙΑ ΤΗΣ MYSQL	37
2.10.4	ΠΩΣ ΛΕΙΤΟΥΡΓΕΙ Η SQL.....	38
2.10.5	ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ MYSQL.....	39
2.10.6	ΚΥΡΙΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ MYSQL	41
2.10.7	ΙΣΤΟΡΙΑ ΤΗΣ ΒΑΣΗΣ MYSQL	42
2.10.8	ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΗΣ MYSQL:	42
2.10.9	ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΗΣ MYSQL:	45
3	ΚΥΚΛΟΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΗΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΒΙΒΛΙΟΘΗΚΗΣ	47
3.1	ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ.....	47
3.1.1	ΕΦΑΡΜΟΓΗ ΧΡΗΣΤΩΝ	48
3.1.1.1	ΧΡΗΣΤΗΣ ΜΕ ΑΠΕΡΙΟΡΙΣΤΗ ΠΡΟΣΒΑΣΗ.....	49
3.1.1.2	ΧΡΗΣΤΗΣ ΜΕ ΠΕΡΙΟΡΙΣΜΕΝΗ ΠΡΟΣΒΑΣΗ	49
3.1.1.3	ΔΟΚΙΜΑΣΤΙΚΟΣ ΧΡΗΣΤΗΣ.....	50
3.1.2	ΕΦΑΡΜΟΓΗ ΕΚΔΟΤΙΚΩΝ ΟΙΚΩΝ.....	50
3.1.3	ΕΦΑΡΜΟΓΗ ΔΙΑΧΕΙΡΙΣΗΣ.....	50
3.2	ΣΧΕΔΙΑΣΜΟΣ ΛΟΓΙΣΜΙΚΟΥ.....	51
3.3	ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ.....	54
3.4	ΔΟΚΙΜΗ ΛΟΓΙΣΜΙΚΟΥ.....	55
3.5	ΛΕΙΤΟΥΡΓΙΑ ΚΑΙ ΣΥΝΤΗΡΗΣΗ.....	56
4	ΚΑΤΑΓΡΑΦΗ ΑΠΑΙΤΗΣΕΩΝ ΕΦΑΡΜΟΓΗΣ	57
4.1	ΟΡΙΣΜΟΣ.....	57
4.2	ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ	57
4.2.1	ΔΙΑΔΙΚΑΣΙΑ ΕΠΙΛΟΓΗΣ ΚΑΙ ΑΠΟΚΤΗΣΗΣ ΤΟΥ ΒΙΒΛΙΟΥ.....	57
4.3	ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ.....	58

4.3.1	ΑΠΑΙΤΗΣΕΙΣ ΧΡΗΣΗΣ	58
4.3.2	ΑΠΑΙΤΗΣΕΙΣ ΑΞΙΟΠΙΣΤΙΑΣ	58
4.3.3	ΑΠΑΙΤΗΣΕΙΣ ΕΠΙΔΟΣΕΩΝ.....	59
4.3.4	ΑΠΑΙΤΗΣΕΙΣ ΥΠΟΣΤΗΡΙΞΗΣ.....	59
4.3.5	ΑΠΑΙΤΗΣΕΙΣ ΥΛΟΠΟΙΗΣΗΣ	59
4.3.6	ΑΠΑΙΤΗΣΕΙΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ	59
4.3.7	ΦΥΣΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ	59
5	ΜΟΝΤΕΛΟΠΟΙΗΣΗ.....	60
5.1.1	ΕΙΣΑΓΩΓΙΚΑ.....	60
5.1.2	ΔΙΑΓΡΑΜΜΑ ΧΡΗΣΗΣ (USE CASE).....	61
5.1.3	ΔΙΑΓΡΑΜΜΑΤΑ ΑΚΟΛΟΥΘΙΑΣ (SEQUENCE)	68
6	ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	71
6.1	ΠΙΝΑΚΕΣ ΚΑΙ ΠΕΔΙΑ.....	73
6.1.1	ΠΙΝΑΚΑΣ ΧΡΗΣΤΕΣ.....	73
6.1.2	ΠΙΝΑΚΑΣ ΒΙΒΛΙΑ	75
6.1.3	ΠΙΝΑΚΑΣ ΚΑΤΗΓΟΡΙΕΣ ΒΙΒΛΙΩΝ.....	76
6.1.4	ΠΙΝΑΚΑΣ ΣΥΓΓΡΑΦΕΙΣ	77
6.1.5	ΠΙΝΑΚΑΣ ΜΕΤΑΦΟΡΤΩΣΗΣ ΑΡΧΕΙΩΝ	77
6.1.6	ΠΙΝΑΚΑΣ ΕΚΔΟΤΙΚΟΙ ΟΙΚΟΙ	78
6.1.7	ΠΙΝΑΚΑΣ ΒΙΒΛΙΑ ΣΥΓΓΡΑΦΕΑ	79
6.1.8	ΠΙΝΑΚΑΣ ΒΙΒΛΙΑ ΕΚΔΟΤΙΚΟΥ ΟΙΚΟΥ	80
7	ΠΕΡΙΓΡΑΦΗ ΕΡΓΑΛΕΙΩΝ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ	82
7.1	WAMP SERVER 2.0.....	82
7.2	PHPMYADMIN	82
7.3	HOMESITE	83
8	ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ	84
8.1	ΔΙΑΚΙΩΜΑΤΑ ΜΕΛΟΥΣ.....	84
8.2	ΚΑΙΝΟΥΡΓΙΑ ΜΕΛΗ	84
8.3	ΕΓΓΕΓΡΑΜΜΕΝΑ ΜΕΛΗ	88
8.4	ΕΚΔΟΤΙΚΟΣ ΟΙΚΟΣ.....	93
8.5	ΑΡΜΟΔΙΟΤΗΤΕΣ ΕΛΕΓΚΤΗ ΠΕΡΙΕΧΟΜΕΝΟΥ ΒΙΒΛΙΩΝ	94
8.6	ΑΡΜΟΔΙΟΤΗΤΕΣ ADMINISTRATOR.....	96
9	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	100

10	ΜΕΛΛΟΝΤΙΚΕΣ ΒΕΛΤΙΩΣΕΙΣ	101
	ΠΑΡΑΡΤΗΜΑ	103
11	ΠΗΓΕΣ-ΒΙΒΛΙΟΓΡΑΦΙΑ.....	109

ΕΥΧΑΡΙΣΤΙΕΣ

Θα θέλαμε να ευχαριστήσουμε τον καθηγητή μας κ. Χόχολη Διονύσιο που μας καθοδήγησε και μας βοήθησε να υλοποιήσουμε την πτυχιακή εργασία που μας είχε ανατεθεί.

Αντώνης Αντωνίου – Μαρία Μαυρικάκη

Θα ήθελα να ευχαριστήσω τους γονείς μου που με στήριξαν όλο αυτό το διάστημα των σπουδών μου.

Αντώνης Αντωνίου

Θα ήθελα να ευχαριστήσω τους γονείς μου και ιδιαίτερα τον πατέρα μου που με στήριξε και με στηρίζει σε κάθε μου επιλογή.

Μαρία Μαυρικάκη

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή αποσκοπεί στην ανάλυση, τον σχεδιασμό και την ανάπτυξη μιας εφαρμογής διαχείρισης ψηφιακής βιβλιοθήκης.

Η εφαρμογή θα υποστηρίζει τόσο τις βασικές εργασίες που θα εκτελεί ο γενικός διαχειριστής, όσο και τις υπηρεσίες των μελών-χρηστών της βιβλιοθήκης.

Στην βάση δεδομένων θα καταγράφονται το σύνολο των βιβλίων της βιβλιοθήκης, με αναλυτικές πληροφορίες για καθένα από αυτά.

Θα παρέχεται δυνατότητα αναζήτησης στην βάση δεδομένων καθώς επίσης και διαχείρισης των βιβλίων (εισαγωγή, τροποποίηση πληροφοριών των βιβλίων, διαγραφή των βιβλίων).

Στη βάση δεδομένων θα καταχωρούνται και τα προσωπικά στοιχεία των μελών-χρηστών που ζητούνται κατά την εγγραφή τους καθώς και τα στοιχεία πρόσβασης που είναι απαραίτητα για την είσοδο του στην εφαρμογή.

Κάθε μέλος θα έχει τον δικό του λογαριασμό και δυνατότητα πρόσβασης στην εφαρμογή. Θα μπορεί να :

1. κάνει αναζήτηση βιβλίων,
2. να κατεβάσει τα βιβλία που τον ενδιαφέρουν,
3. καθώς και να δει την καρτέλα με τα βιβλία που έχει κατεβάσει.

Ανάλογα με την ιδιότητα του κάθε χρήστη (απλός χρήστης, εκδοτικός οίκος, ελεγκτής περιεχομένου βιβλίων, γενικός διαχειριστής) θα έχει και τα αντίστοιχα δικαιώματα πρόσβασης στις λειτουργίες της εφαρμογής.

ABSTRACT

This present study aims to analyze, design and develop an application for the management of a digital library.

The application will support the basic tasks that the general administrator will execute, as well as the services of the member-users of the library.

All the books of the library will be recorded on the database, with detailed information for each one. The capability of searching to the database as well to the management of the books (introduction, amendment of the information, deletion of the books), will be provided.

On the database will also be recorded the personal information of the member-users that is necessary for their registration, as well as the access codes that are necessary in order to access the application. Each member will maintain its account and will have the capability of accessing the application. It will be possible for the members to:

1. Search for books.
2. Download interesting books.
3. View the relevant card of the downloaded books.

Depending on the status of each user (simple user, publishing house, checker of book content, general administrator), each one will have the appropriate authority to access the application.

1 ΕΙΣΑΓΩΓΗ

1.1 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Η μνήμη των ανθρώπων είναι υψίστης σημασίας για την διατήρηση της πνευματικής προσωπικότητας, για το γεφύρωμα του παρελθόντος με το παρόν και για τη διαμόρφωση του μέλλοντος. Η καταγεγραμμένη κληρονομιά αποτελεί το κύριο μέρος αυτής της μνήμης και αντανακλά στην ποικιλία των ανθρώπων, των γλωσσών και των πολιτισμών.

Η μνήμη αυτή όμως είναι εύθραυστη. Πολύτιμα και μοναδικά τεκμήρια, καμιά φορά και ολόκληρες συλλογές βιβλιοθηκών και αρχείων, μπορούν να χαθούν πολύ περισσότερο στην σημερινή εποχή της δημιουργίας ψηφιακών βιβλιοθηκών. Τα ψηφιακά τεκμήρια όπως και τα συμβατικά έχουν ανάγκη διατήρησης και οι βιβλιοθήκες, έχουν χρέος να φροντίζουν για τη σωστή διατήρησή τους.

Οι βιβλιοθήκες συνεπικουρούμενες από την ανάπτυξη της τεχνολογίας σήμερα, αποτελούν σύνθετους οργανισμούς με πολλαπλούς στόχους που δεν έχουν προηγούμενο. Αυτό δεν συνέβαινε ανέκαθεν. Επί αιώνες δημιουργούσαν έντυπες συλλογές και έντυπες βάσεις δεδομένων, τους γνωστούς δελτίο - καταλόγους.

Όταν οι τεχνολογίες των οπτικοακουστικών αναπτύχθηκαν και διαδόθηκαν ευρέως, οι βιβλιοθήκες απέκτησαν μεγάλες συλλογές για εκπαιδευτικούς και ψυχαγωγικούς σκοπούς. Ακόμη και στην εκπαίδευση των χρηστών, ακόμη και στην απομακρυσμένη εκπαίδευση έπαιξαν πρωτεύοντα ρόλο τα οπτικοακουστικά μέσα. Μέχρι αυτή τη στιγμή η οργάνωση και η εξέλιξη των βιβλιοθηκών κινείτο κανονικά σε κάθε στάδιο, χωρίς χάσματα, ραγδαίες αλλαγές ή ανατροπές στη σκέψη και τη πράξη. Οι βιβλιοθήκες δρούσαν με επίκεντρο τη συλλογή τους και

αποφάσιζαν την οργάνωσή τους με κριτήριο τη διαφύλαξη και διατήρησή της.

Αλλά κατά τη δεκαετία του '80, η νέα τεχνολογία με την επωνομασία “συστήματα βιβλιοθηκών” ήρθε στο προσκήνιο. Έγινε κατανοητό ότι οι βιβλιοθήκες θα μπορούσαν να απλοποιήσουν και να επεκτείνουν τη δουλειά τους και ξεκίνησαν να υιοθετούν τα συστήματα αυτά μαζικά. Ήταν η πρόκληση των μεγάλων προγραμμάτων αυτοματοποίησης των βιβλιοθηκών. Η υλοποίηση της νέας τεχνολογίας έφερε και επίδραση στην οργάνωση των βιβλιοθηκών. Πέραν των ανωτέρω, το διαδίκτυο, το ηλεκτρονικό ταχυδρομείο και ο παγκόσμιος ιστός εισέβαλαν στη ζωή μας.

Με την πληθώρα νέων πηγών, τοπικών και απομακρυσμένων, τις νέες δυνατότητες για παραλαβή υλικού, την απουσία ομοιομορφίας στα διάφορα interfaces των διαρκώς μεταβαλλόμενων μηχανών αναζήτησης και τις τεχνικές δεξιότητες που απαιτούντο, το προσωπικό των βιβλιοθηκών και των κέντρων πληροφόρησης δυσκολεύτηκε να αφυπνισθεί, πράγμα που ενέτεινε η αίσθηση ότι είναι ευάλωτο και ανεπαρκές.

Η αβεβαιότητα που συνδυάζεται με την οργανωτική αλλαγή είναι προφανώς αναπόφευκτη, αφού ο εργαζόμενος αισθάνεται να διακυβεύεται ο επαγγελματισμός του, στις μεταβολές. Πάντως, αυτό που έχει διαπιστωθεί από καιρού εις καιρόν είναι ότι η τεχνολογία αποτελεί μόνο το πρόσχημα της οργανωτικής αλλαγής.

Σήμερα, οι στόχοι και οι σκοποί των βιβλιοθηκών έχουν μάλλον εμπλουτιστεί σε σύγκριση με το παρελθόν, η ταυτότητα της βιβλιοθήκης έχει τροποποιηθεί, όχι όμως και ο σκοπός και η αποστολή της.

Οι βιβλιοθήκες, εκτός από τεχνολογικά εξακολουθούν να είναι κοινωνικά συστήματα, να επικοινωνούν με ανθρώπους, να υποστηρίζουν τις ανάγκες τους σε εκπαίδευση και ψυχαγωγία. Εξακολουθούν να

προστατεύουν την ανθρώπινη ιστορία και σκέψη δια μέσου των αιώνων και να εκπαιδεύουν το κοινό για το πώς θα χρησιμοποιεί το υλικό που χρειάζεται. Το υλικό αυτό η βιβλιοθήκη το έχει πριν από αυτούς αναζητήσει, εντοπίσει, αξιολογήσει, αποκτήσει, οργανώσει και διαμοιράσει.

Επομένως, η τεχνολογία δεν αποτελεί από μόνη της το θεμελιώδες χαρακτηριστικό της σύγχρονης βιβλιοθήκης, απλά εμπλουτίζει τις μορφές και τον όγκο των πηγών που η τελευταία προσφέρει. Συνεκδοχικά, η τεχνολογία προκαλεί ραγδαίες οργανωτικές αλλαγές και αποκαλύπτει πώς αυτές υλοποιούνται και πώς δρουν στο περιβάλλον της βιβλιοθήκης.⁵

1.2 ΠΡΟΤΕΙΝΟΜΕΝΗ ΛΥΣΗ

Όπως ήδη σημειώθηκε, οι βιβλιοθήκες ανταποκρίνονται στις προκλήσεις της κοινωνίας της πληροφορίας και στις απαιτήσεις μιας νέας γενιάς χρηστών εξοικειωμένων με τις νέες τεχνολογίες, επομένως κρίθηκε απαραίτητη η ανάπτυξη μιας διαδικτυακής εφαρμογής ψηφιακής βιβλιοθήκης εκσυγχρονίζοντας τα βιβλία που διαθέτουν από παραδοσιακά σε ψηφιακά.

Για την είσοδο του χρήστη στην ψηφιακή βιβλιοθήκη είναι απαραίτητη η εγγραφή του σε αυτήν και η απόκτηση κωδικού πρόσβασης. Εφόσον συνδεθεί με την εφαρμογή, θα του δίνεται η δυνατότητα να αναζητήσει το βιβλίο που επιθυμεί μέσα από ένα σύνολο κατηγοριών με τρόπο διαφορετικό και σαφώς ευκολότερο από μια παραδοσιακή βιβλιοθήκη. Η αναζήτηση αυτή έχει ως στόχο την εύρεση του βιβλίου και την ανάγνωση του, προϋποθέτοντας ότι ο χρήστης έχει κάνει ήδη “download” το ψηφιακό περιεχόμενο του βιβλίου.

Τα ψηφιακά βιβλία που διαθέτει η ψηφιακή βιβλιοθήκη στέλνονται από διάφορους εκδοτικούς οίκους στον ελεγκτή βιβλίων της βιβλιοθήκης, ο ελεγκτής είναι υπεύθυνος για τον έλεγχο των βιβλίων που στέλνονται από τους εκδοτικούς οίκους. Επίσης, είναι το άτομο που αποφασίζει ποια από τα βιβλία αυτά θα καταχωρηθούν στην βιβλιοθήκη. Για την μεταφόρτωση των ψηφιακών βιβλίων είναι αρμόδιος ο γενικός διαχειριστής όπως επίσης και για την γενικότερη ομαλή λειτουργία της ψηφιακής βιβλιοθήκης .

Για την ανάλυση, τον σχεδιασμό και την ανάπτυξη της διαδικτυακής εφαρμογής ψηφιακής βιβλιοθήκης θα μελετήσουμε τις φάσεις του κύκλου ανάπτυξης της εφαρμογής καθώς και τις λειτουργικές και μη απαιτήσεις της. Θα παρουσιάσουμε τα διαγράμματα χρήσης (use case) και τα διαγράμματα ακολουθίας (sequence) που δημιουργήσαμε κατά την φάση του σχεδιασμού της εφαρμογής, όπως και την αναλυτική περιγραφή της βάσης δεδομένων. Επίσης θα σας αναφέρουμε τα συμπεράσματα στα οποία καταλήξαμε καθώς και της μελλοντικές βελτιώσεις τις οποίες μπορεί να δεχτεί η εφαρμογή.

1.3 ΜΕΘΟΔΟΛΟΓΙΑ

Για την ανάπτυξη της εφαρμογής κρίθηκε αρχικά απαραίτητη η βιβλιογραφική έρευνα σε διεθνείς και ελληνικές πηγές (μελέτες, έρευνες) καθώς και η μελέτη παρόμοιων διαδικτυακών εφαρμογών που υπάρχουν στο διαδίκτυο. Ακολούθησε ο καθορισμός και η ανάλυση των απαιτήσεων της εφαρμογής με σκοπό την κάλυψη των στόχων και των αναγκών που απαιτούνται για την ανάπτυξη της. Για την σχεδίαση μιας εφαρμογής η οποία θα εξυπηρετεί τις απαιτήσεις των χρηστών, δημιουργήθηκαν διαγράμματα χρήσης (use case) και διαγράμματα ακολουθίας (sequence), περιγράφοντας τις δυνατότητες που παρέχει στους χρήστες η εφαρμογή.

Για την υλοποίηση της εφαρμογής και την δημιουργία του κώδικα ήταν απαραίτητη η χρήση κάποιων εργαλείων (web server, phpmyadmin, homesite). Με την ολοκλήρωση της εφαρμογής πραγματοποιήθηκε δοκιμή για να διαπιστωθεί αν λειτουργεί χωρίς σφάλματα. Σε περίπτωση που παρατηρηθεί ότι δεν εμφανίζονται τα επιθυμητά αποτελέσματα ή προκύψουν νέες απαιτήσεις η εφαρμογή θα πρέπει να τροποποιηθεί.

1.4 ΠΕΡΙΓΡΑΦΗ ΚΕΦΑΛΑΙΩΝ

Στο 2 κεφάλαιο αναφέρονται οι κυριότερες θεωρητικές έννοιες πάνω στις οποίες βασίστηκε η υλοποίηση της πτυχιακής εργασίας. Στο συγκεκριμένο κεφάλαιο μελετήθηκαν τα κυριότερα μοντέλα ανάπτυξης λογισμικού με ιδιαίτερη έμφαση στο μοντέλο καταρράκτη, το οποίο κρίθηκε κατάλληλο για την ανάπτυξη του λογισμικού μας. Επίσης μελετώνται και αναφέρονται η αρχιτεκτονική καθώς και οι βασικότερες τεχνολογίες των διαδικτυακών εφαρμογών.

Στο 3 κεφάλαιο που ακολουθεί αναλύεται ο κύκλος ανάπτυξης του λογισμικού όπως επίσης και οι πέντε φάσεις από τις οποίες αποτελείται.

Στο κεφάλαιο 4 γίνεται καταγραφή των λειτουργικών και μη λειτουργικών απαιτήσεων.

Στο κεφάλαιο 5 γίνεται η ανάλυση της εφαρμογής με τη βοήθεια των διαγραμμάτων χρήσης (use case) και των διαγραμμάτων ακολουθίας (sequence diagrams).

Στο κεφάλαιο 6 γίνεται παρουσίαση της βάσης δεδομένων, των πινάκων, των πρωτεύοντων και δευτερευόντων κλειδιών.

Στο κεφάλαιο 7 περιγράφονται τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής.

Στο κεφάλαιο 8 γίνεται η παρουσίαση της εφαρμογής.

Στο 10 αναφέρονται τα συμπεράσματα στα οποία καταλήξαμε.

Στο κεφάλαιο 11 αναφέρονται προτάσεις για μελλοντικές βελτιώσεις και προσθήκες για την εφαρμογή.

2 ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

2.1 ΜΟΝΤΕΛΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ

Με στόχο την περιγραφή των διαφόρων σταδίων του κύκλου ανάπτυξης του λογιστικού, έχουν αναπτυχθεί διάφορα μοντέλα ανάπτυξης του. Τα μοντέλα αυτά περιγράφουν κατά κανόνα γενικές κατηγορίες και περιπτώσεις ανάπτυξης λογισμικού παρακάμπτοντας λεπτομερείς αναφορές σε συγκεκριμένες περιπτώσεις. Στην πραγματικότητα τα μοντέλα αυτά καλύπτουν τις γενικές γραμμές ανάπτυξης λογισμικού, ενώ η ανάπτυξη και συντήρηση ενός προϊόντος λογισμικού είναι προσαρμοσμένα στις συνθήκες ανάπτυξής του (χρήστες, ομάδα ανάπτυξης, εμπειρία αναλυτών, υφιστάμενος εξοπλισμός). Σε κάθε φάση ανάπτυξης εφαρμόζονται οι τεχνικές που καλύπτονται από την επιστήμη της τεχνολογίας λογισμικού. Αναφέρονται τα σημαντικότερα μοντέλα ανάπτυξης λογισμικού:

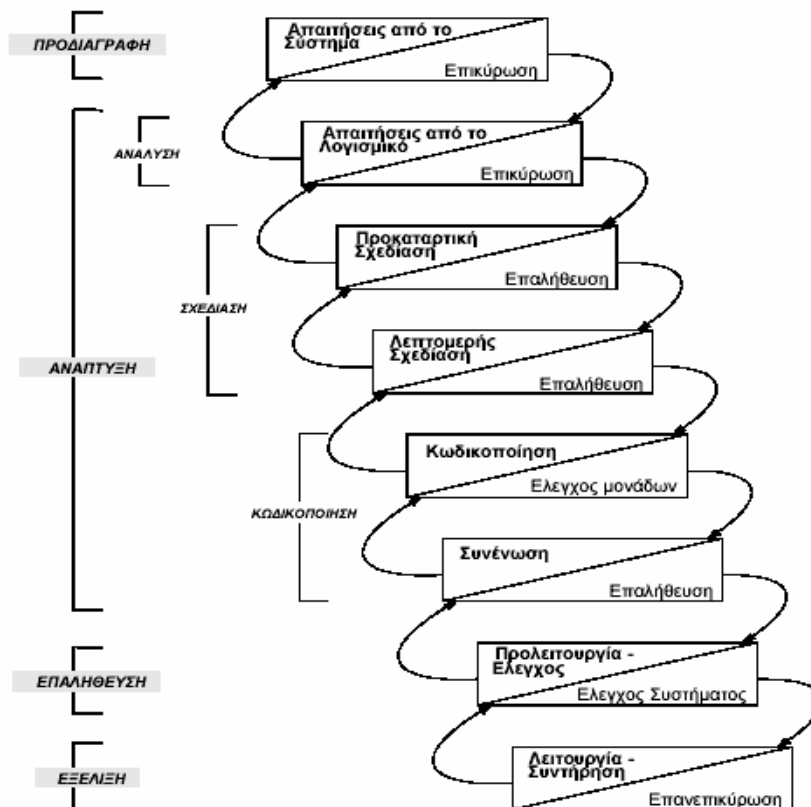
- § Μοντέλο Αυτόματου Προγραμματισμού
- § Συμβατικά και Εναλλακτικά Μοντέλα Ανάπτυξης Λογισμικού
- § Μοντέλο Καταρράκτη
- § Μοντέλο Προτυποποίησης
- § Μοντέλο Λειτουργικής Επαύξησης
- § Λειτουργικό Μοντέλο
- § Μοντέλο Επαναχρησιμοποίησης Λογισμικού
- § Αντικειμενοστραφές Μοντέλο
- § Σπειροειδές μοντέλο

Από τα προαναφερθέντα μοντέλα εμείς θα σταθούμε σε αυτό του καταρράκτη. Το μοντέλο καταρράκτη (waterfall model ή linear sequential model) αναπτύχθηκε το 1970 από τον Royce.

Ήταν το πρώτο μοντέλο που δημιουργήθηκε και έγινε ευρέως αποδεκτό, ενώ ακόμα παραμένει δημοφιλές ιδιαίτερα για μικρά ή μεσαία μεγέθη εφαρμογών αφού συμβάλλει στην επιτυχή κατασκευή αξιόπιστων προϊόντων σε μικρό χρονικό διάστημα.

Στο μοντέλο αυτό οι διάφορες φάσεις διαχωρίζονται και ακολουθούνται σειριακά. Η κάθε φάση παράγει ενδιάμεσα προϊόντα τα οποία χρησιμοποιούνται από τις επόμενες φάσεις και κορυφώνεται από μια διαδικασία επικύρωσης ή επαλήθευσης των προϊόντων που παράγονται, με σκοπό να απαλειφθούν τυχόν σφάλματα.

Οι παραλλαγές αυτές διαφοροποιούνται περισσότερο στον τρόπο που γίνεται ο έλεγχος, η διόρθωση, η επαλήθευση, η επικύρωση καθώς και ο τρόπος που θα πραγματοποιηθούν αυτές δηλαδή σε ποια φάση θα πρέπει να επιστρέψουμε αν χρειαστεί ενώ παράλληλα οι φάσεις που περιλαμβάνονται χαρακτηρίζονται είτε από την σύμπτυξη είτε από τον επιμέρους διαχωρισμό των αντίστοιχων φάσεων του αντίστοιχου μοντέλου του καταρράκτη.



Σχήμα 2.1 (Μοντέλο καταρράκτη)

2.1.1 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΟΥ ΜΟΝΤΕΛΟΥ

Το μοντέλο καταρράκτη έχει πολλά ελκυστικά χαρακτηριστικά γνωρίσματα:

- § Σαφώς καθορισμένα παραδοτέα στο τέλος κάθε φάσης, έτσι ώστε ο χρήστης μπορεί να πάρει αποφάσεις σχετικά με τη συνέχιση ή όχι του προγράμματος
- § Επαυξητική δέσμευση πόρων. Ο χρήστης δεν είναι απαραίτητο να κάνει πλήρη δέσμευση όλων των πόρων στο έργο από την αρχή

Εντούτοις, έχει και μερικά μειονεκτήματα:

- § Απαιτεί μια οριστική προσέγγιση στην ανάπτυξη του συστήματος και δεν επιτρέπει τη επαυξητική ανάπτυξη
- § Απαιτεί την πολύ πρόωρη απομόνωση του προβλήματος. Στον πραγματικό κόσμο, συχνά τα προβλήματα αποκαλύπτονται στο στάδιο της σχεδίασης και κατασκευής του συστήματος

§ Ο χρήστης/πελάτης σπάνια γνωρίζει από την αρχή όλες του τις ανάγκες και έτσι πάντα οι απαιτήσεις του για το σύστημα είναι ελλιπείς. Το μοντέλο αυτό επομένως ξεκινάει πολύ συχνά από μια λάθος υπόθεση.

2.1.2 ΛΟΓΟΣ ΧΡΗΣΙΜΟΠΟΙΗΣΗΣ ΜΟΝΤΕΛΟΥ ΚΑΤΑΡΡΑΚΤΗ

Ο λόγος που χρησιμοποιήσαμε το μοντέλο καταρράκτη είναι επειδή χρησιμοποιείται σε μικρές και μεσαίες εφαρμογές και δεν είναι ιδιαίτερα δημοφιλές στις εμπορικές εφαρμογές. Δεν χρησιμοποιήσαμε κάποιο άλλο μοντέλο για διάφορους λόγους όπως η έλλειψη ετοιμών κομματιών κώδικα λογισμικού σε συνδυασμό με την ανυπαρξία εργαλείων και των κατάλληλων τεχνικών ενσωμάτωσης. Επίσης το μεγάλο κόστος και οι υψηλές γνώσεις πληροφορικής που χρειαζόταν δεν μας επέτρεπαν την χρήση άλλων μοντέλων όπως της λειτουργικής επαύξησης ή πρωτυποποίησης. Τέλος, ένας λόγος ήταν ότι η εφαρμογή μας δεν έχει εμπορικούς σκοπούς, απευθύνεται ως ελεύθερο λογισμικό σε οργανισμούς και ιδιώτες.

2.2 ΔΙΑΔΙΚΤΥΑΚΕΣ ΕΦΑΡΜΟΓΕΣ

Η τεράστια ανάπτυξη του διαδικτύου την δεκαετία του `90 δημιούργησε ένα νέο πεδίο στην επιστήμη των υπολογιστών και κυρίως την μηχανική λογισμικού, την ανάπτυξη διαδικτυακών εφαρμογών. Η ανάπτυξη διαδικτυακών εφαρμογών διαφέρει σε πολλά σημεία από την ανάπτυξη λογισμικού σε κλασικότερες μορφές, όπως εφαρμογών για χρήση σε desktop περιβάλλον. Οι διαφορές μιας διαδικτυακής σε σύγκριση με τις κλασικότερες μορφές οφείλονται κατά κύριο λόγο στο γεγονός ότι η πλατφόρμα εγκατάστασης και εφαρμογής της είναι το διαδίκτυο.

2.2.1 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ

Οι διαδικτυακές εφαρμογές έχουν αρκετά πλεονεκτήματα, μερικά από τα οποία αναφέρονται παρακάτω:

- § Είναι συμβατές με όλες τις πλατφόρμες (υποστηρίζουν όλα τα Λειτουργικά Συστήματα).
- § Υπάρχει συνεχής ενημέρωση των εφαρμογών (χωρίς να απαιτείται καμιά ενέργεια από τους χρήστες).
- § Παρέχουν αμεσότητα πρόσβασης των δεδομένων (δεν χρειάζονται εγκατάσταση και διαμόρφωση).
- § Ευκολία δοκιμής και χρήσης (υπάρχουν δοκιμαστικές εκδόσεις, ειδικά για ακριβά λογισμικά).
- § Μικρότερες απαιτήσεις μνήμης (από ότι τα τοπικά εγκατεστημένα προγράμματα).
- § Λιγότερα τεχνικά προβλήματα (όσον αφορά το software αλλά και το hardware, και τυχόν προβλήματα που παρουσιάζονται λύνονται πολύ πιο γρήγορα από ότι αν ήταν εγκατεστημένες τοπικά).
- § Χαμηλότερα κόστη (δεν απαιτείται διανομή, τεχνική υποστήριξη και υποδομή μάρκετινγκ του λογισμικού).
- § Συγκεντρωμένα δεδομένα, κάτι που κάνει πιο εύκολη την αποθήκευση τους και την δημιουργία αντιγράφων ασφαλείας (εταιρίες αποθηκεύουν τα δεδομένα με σύγχρονα μέσα και μεθόδους σε server farms όπως λέγονται, μειώνοντας τον κίνδυνο απώλειας αυτών, από ότι ήταν τοπικά σε κάθε χρήστη).
- § Οι πληροφορίες είναι προσβάσιμες από περισσότερους χρήστες (σε όλο το κόσμο).
- § Η πρόσβαση των πληροφοριών είναι συνεχής.
- § Υπάρχει ταυτόχρονη εξυπηρέτηση πολλών χρηστών.

§ Η ανάπτυξή τους μπορεί να γίνει σε οποιαδήποτε γλώσσα επιθυμούμε από τη στιγμή που δεν υπάρχουν οι περιορισμοί του Λειτουργικού Συστήματος, λόγω της ανεξαρτησίας τους από αυτό.

2.2.2 ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ

Στην προηγούμενη παράγραφο αναφέραμε τα πλεονεκτήματα που έχουν οι διαδικτυακές εφαρμογές. Στη συνέχεια καταγράφονται κάποια μειονεκτήματα τους τα οποία αναφέρονται κυρίως στο σχεδιασμό και στις γλώσσες που χρησιμοποιούν:

§ Περιορισμένη ευελιξία: Οι διαδικτυακές εφαρμογές, είναι εξαιρετικές σε ένα βασικό επίπεδο, όμως δεν μπορούν να φτάσουν σε ποιότητα εφαρμογές που αποθηκεύονται τοπικά στους προσωπικούς υπολογιστές.

§ Τρόπος διαχείρισης: Δημιουργούνται ερωτηματικά όσον αφορά τον όγκο των προσωπικών δεδομένων στα οποία θα έχει πρόσβαση ο διαχειριστής καθώς όλα τα δεδομένα θα αποθηκεύονται στα δίκτυα και τους server της εταιρείας.

§ Κόστος: Ένας server έχει μεγάλο κόστος, το οποίο αυξάνεται αναλόγως από την ύπαρξη διαφόρων εργαλείων που ενσωματώνει, όπως ύπαρξη επιπλέον βάσεων δεδομένων ή υποστήριξη διαφόρων τεχνολογιών.

§ Επεκτασιμότητα: Συνήθως, για λόγους εξοικονόμησης πόρων, κάθε εφαρμογή στο Internet αρχίζει τη λειτουργία της με μια φθηνή πλατφόρμα υλοποίησης με πρόβλεψη αναβάθμισής της, αν αυτή αποδειχθεί δημοφιλής. Όμως υπάρχουν πολλές περιπτώσεις όπου οι αρχικές επιλογές δεσμεύουν την υπηρεσία και περιορίζουν τις δυνατότητες επέκτασής της.

- § Ειδικευμένο προσωπικό: Εννοείται ότι αν η εταιρεία διαθέτει ήδη τεχνογνωσία σε μια πλατφόρμα είναι προτιμότερο να επενδύσει εκεί όπου υπερέχει ήδη παρά να υποχρεωθεί σε επανεκπαίδευση του προσωπικού της ή σε πρόσληψη νέου.
- § Ανοιχτή πλατφόρμα υλοποίησης: Τόσο στο hardware όσο και στο software υπάρχουν κάποια προϊόντα και εφαρμογές που "δένουν" τον πελάτη με τον αρχικό προμηθευτή του, ενώ άλλα του παρέχουν μεγαλύτερη ελευθερία όπως είναι οι εφαρμογές ανοικτού κώδικα.
- § Το σημαντικότερο όμως μειονέκτημα των διαδικτυακών εφαρμογών αφορά την ασφάλεια, αφού σχετίζονται άμεσα με το WEB.

2.2.3 ΑΣΦΑΛΕΙΑ ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΦΑΡΜΟΓΩΝ

Οι διαδικτυακές εφαρμογές δέχονται απειλές που κλονίζουν την ασφάλεια τους. Με τον όρο απειλή εννοείται οποιοδήποτε πιθανό περιστατικό, κακόβουλο ή μη, που μπορεί να βλάψει την εφαρμογή. Λόγω αδυναμιών στη σχεδίαση, λάθη στη διαμόρφωση ή λόγω ακατάλληλων και επισφαλών τεχνικών κωδικοποίησης η εφαρμογή γίνεται περισσότερο ευπαθής, με αποτέλεσμα αυτές οι αδυναμίες να κάνουν δυνατή την απειλή. Για το λόγο αυτό γίνεται η χρήση εργαλείων για την εύρεση αδυναμιών της εφαρμογής, τα οποία αναφέρονται παρακάτω:

- § Τα “Web scanning” τα οποία μπορούν να μας βοηθήσουν να βρούμε αυτές τις αδυναμίες, ειδικά αν είναι γνωστά bugs. Όμως, για να βρούμε όλες τις αδυναμίες που πιθανόν να υπάρχουν θα πρέπει να γίνει καλή “επιθεώρηση” του κώδικα της εφαρμογής, όπως επίσης και έλεγχος διείσδυσης στην εφαρμογή (application penetration test). Αυτές οι ενέργειες

πρέπει να εκτελούνται από τους υπεύθυνους ανάπτυξης της εφαρμογής, πριν από την κυκλοφορία κάθε εφαρμογής.

§ Πρέπει να γίνονται συχνές επιθεωρήσεις της διαμόρφωσης των υποδομών της εφαρμογής και να επιβάλλεται η κατάλληλη ασφάλεια.

§ Οι διαχειριστές των συστημάτων πρέπει να “σκανάρουν” περιοδικά τους web servers με διάφορα vulnerability scanners, ιδιαίτερα αν οι servers χρησιμοποιούν μεγάλο σε αριθμό και ποικιλία σύνολο scripts.

2.3 ΕΠΙΘΕΩΡΗΣΗ ΑΣΦΑΛΕΙΑΣ (SECURITY AUDIT):

Ένα σύστημα ελέγχεται με βάση ένα σύνολο από πίνακες ελέγχου (checklists), οι οποίοι διαμορφώνονται με βάση διεθνή πρότυπα σχετικά με την ασφάλεια και τις κατάλληλες πολιτικές ασφαλείας ενός οργανισμού. Οι ελεγκτές εκτελούν την εργασία τους μέσα από προσωπικές συνεντεύξεις, ανιχνεύσεις αδυναμιών, εξετάσεις των ρυθμίσεων, αναλύσεις των διαμοιρασμένων πόρων δικτύου και μελέτες των ιστορικών στοιχείων (log files).

2.3.1 ΑΥΤΟ-ΑΞΙΟΛΟΓΗΣΗ ΑΣΦΑΛΕΙΑΣ (SECURITY SELF-ASSESSMENT):

Δεν υπάρχουν συγκεκριμένα standards ως προς τα οποία θα μετρηθεί το σύστημα, αλλά ο στόχος προσδιορίζεται από την περιοχή που χρειάζεται διερεύνηση και βελτίωση στη θωράκιση της. Ξεπερνά τους πίνακες ελέγχου (checklists) και επεκτείνεται, σε ένα πιο λεπτομερή έλεγχο για εντοπισμό αδυναμιών αλλά και σε συστάσεις για επιδιορθώσεις και βελτιώσεις. Πλεονέκτημα της είναι η δυνατότητα να οριστούν επίπεδα προτεραιότητας σε κάθε συστατικό που αξιολογείται, έτσι ώστε με την ολοκλήρωση της να δοθεί μια σειρά προτεραιότητας στην επιδιόρθωση των αδυναμιών.

2.3.2 ΔΟΚΙΜΗ ΔΙΕΙΣΔΥΣΗΣ (PENETRATION TESTING Η “ETHICAL HACKING”):

Είναι η ελεγχόμενη προσομοίωση μιας επίθεσης, προκειμένου να επιτευχθεί ένας προκαθορισμένος στόχος. Σκοπός της είναι να εντοπιστούν συγκεκριμένες πληροφορίες (σχετικές με την ύπαρξη αδυναμιών) και να διερευνηθεί κατά πόσο είναι δυνατόν ένας ξένος, κάνοντας χρήση αυτών των πληροφοριών, να μπορέσει να δημιουργήσει προβλήματα. Δεν έχει σκοπό να εντοπίσει όλες τις αδυναμίες, αλλά να υποδείξει ότι η ασφάλεια του συστήματος μπορεί να διακυβευτεί. Η δοκιμή μπορεί να πραγματοποιηθεί με βάση μηδενική γνώση (zero knowledge), ή με πλήρη γνώση (full knowledge) του συστήματος που δοκιμάζεται. Χρησιμοποιείται για να καθορίσει την αξιοπιστία και τη δύναμη των μέτρων ασφαλείας που παίρνουμε. Οι “ethical hackers” προσπαθούν να υιοθετήσουν τις επιθέσεις και τις τεχνικές των hackers, για να μετρήσουν το επίπεδο ασφαλείας.

Η ασφάλεια των διαδικτυακών εφαρμογών διακρίνεται σε δυο τεχνικές:

§ **BLACK BOX:** είναι η εφαρμογή δοκιμαστικών δεδομένων που έχουν προέλθει από καθορισμένες λειτουργικές απαιτήσεις, χωρίς να λαμβάνουν υπόψη τη δομή της εφαρμογής στην οποία εφαρμόζονται. Η εφαρμογή εξετάζεται χρησιμοποιώντας την εξωτερική της διεπαφή, αυτή που χρησιμοποιούν οι απλοί χρήστες. Μιμούνται την ακολουθία αλληλεπιδράσεων χρήστη-εφαρμογής και κάθε αποτυχία δείχνει ότι ο χρήστης έλαβε ανεπαρκή υπηρεσία

§ **WHITE BOX:** εξετάζεται η δομή της εφαρμογής και βάσει αυτής καθορίζονται τα δεδομένα της δοκιμής. Εξετάζεται η εσωτερική δομή της εφαρμογής, χρησιμοποιώντας τη διεπαφή προγραμματισμού εφαρμογών.

2.4 WEB SERVER

Ο web server είναι ένας υπολογιστής που επιτρέπει σε άλλους υπολογιστές, να έχουν πρόσβαση στα αρχεία που ο ίδιος ο web server διαχειρίζεται, με τη βοήθεια του HTTP πρωτοκόλλου (Hyper Text Transfer Protocol). Να σημειωθεί ότι ο βασικός διαχειριστής του web server έχει τον απόλυτο έλεγχο του διακομιστή σε αντίθεση με ένα απλό χρήστη.

Συνήθως, ο απλός χρήστης περιορίζεται μόνο στο να ζητήσει (request), από τον web server, να δει μια συγκεκριμένη ιστοσελίδα στην οποία η πρόσβαση ελέγχεται από το διαχειριστή της ιστοσελίδας ή/και από το διαχειριστή ολόκληρου του διακομιστή.

Οι διακομιστές χωρίζονται σε δύο είδη, τους “dedicated servers” που ασχολούνται αποκλειστικά με τον έλεγχο των ιστοσελίδων και στους “interactive servers” που πέρα από τη δυνατότητα διαχείρισης web αρχείων εξυπηρετούν και άλλους σκοπούς.

2.4.1 ΙΣΤΟΡΙΑ

Το 1989 ο Tim Berners-Lee πρότεινε στον εργοδότη του CERN (Ευρωπαϊκός Οργανισμός Πυρηνικής Έρευνας), ένα νέο σχέδιο, το οποίο είχε ως στόχο τη διευκόλυνση της ανταλλαγής πληροφοριών μεταξύ των επιστημόνων, χρησιμοποιώντας ένα σύστημα υπερκειμένου. Ως αποτέλεσμα της υλοποίησης αυτού του έργου, το 1990 ο Berners-Lee έγραψε δυο προγράμματα:

§ Ένα browser που ονομάζεται WorldWideWeb και

§ Τον πρώτο web server, αργότερα έγινε γνωστό ως CERN httpd.

Μεταξύ του 1991 και του 1994, η απλότητα και η αποτελεσματικότητα της πρόωρης τεχνολογίας που χρησιμοποιούνται για την πλοήγηση και την ανταλλαγή των δεδομένων μέσω του World Wide Web βοήθησε σε πολλά διαφορετικά λειτουργικά συστήματα και την

εξάπλωση της χρήσης τους μεταξύ των τμημάτων διαφορετικών κοινωνικών ομάδων, πρώτα στους επιστημονικούς οργανισμούς, στη συνέχεια στα πανεπιστήμια και, τέλος στη βιομηχανία.

Το 1994 ο Tim Berners-Lee αποφάσισε να δημιουργήσει το World Wide Web Consortium (W3C) ώστε να διαχειρίζονται την μεγάλη ανάπτυξη των τεχνολογιών που εμπλέκονται (HTTP, HTML, κλπ) μέσω μιας διαδικασίας τυποποίησης.

2.5 APACHE

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 ένα εύρωστο κώδικα για διακομιστή δικτύου, που να μην είναι εμπορικός αλλά να υποστηρίζει πολλά χαρακτηριστικά. Ο Apache όπως έχει αποδειχτεί είναι ο πιο γρήγορος, σταθερός, ασφαλής και υποστηρίζει τα περισσότερα χαρακτηριστικά από οποιονδήποτε άλλο διακομιστή δικτύου.

Ο Apache είναι εγκατεστημένος στο 80% των διακομιστών παγκοσμίως (πάνω από έξι εκατομμύρια διακομιστές). Πάνω του είναι εγκατεστημένα εκατομμύρια sites που δέχονται εκατομμύρια hits καθημερινά χωρίς να παρουσιάζει κανένα απολύτως πρόβλημα. 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.

2.5.1 ΙΣΤΟΡΙΑ APACHE SERVER

Εδώ είναι λίγη ιστορία για τον Apache. Τις πρώτες ημέρες του web, το National Center for Super Computing Applications (NCSA) δημιούργησαν έναν web server που έγινε ο αριθμός ένα web server στις αρχές του 1995. Εντούτοις, ο αρχικός υπεύθυνος για την ανάπτυξη του

Ιστού του NCSA web server άφησε τον NCSA σχεδόν τον ίδιο χρόνο, και το πρόγραμμα web server άρχισε να χρονοτριβεί. Στο μεταξύ, άνθρωποι που χρησιμοποιούσαν τον web server του NCSA άρχισαν να ανταλλάσσουν μεταξύ τους μπαλώματα (patches) για τον web server και σύντομα συνειδητοποίησαν ότι ένα φόρουμ για να διαχειριστεί τα μπαλώματα (patches) ήταν απαραίτητος. Η ομάδα Apache γεννήθηκε.

Η ομάδα χρησιμοποίησε τον web server του NCSA και έναν μεγάλο αριθμό από patches, κωδικοποίησε και γέννησε έναν νέο web server αποκαλούμενο Apache (A patchy server). Αρχικά προερχόμενος από το κώδικα πυρήνων του web server δικτύου NCSA και μιας δέσμης των μπαλωμάτων (patches), ο web server Apache είναι τώρα η συζήτηση της κοινότητας κεντρικών υπολογιστών δικτύου. Σε τρία έτη, απέκτησε το κεντρικό ρόλο κεντρικών υπολογιστών στην αγορά. Η πρώτη έκδοση (0,6,2) δημόσια διανεμημένου Apache κυκλοφόρησε τον Απρίλιο 1995. Η έκδοση 1,0 κυκλοφόρησε την 1η Δεκεμβρίου, το 1995. Η ομάδα Apache έχει διαμορφωθεί και οργανωθεί ως μη κερδοσκοπική ομάδα. Η ομάδα λειτουργεί εξ ολοκλήρου μέσω του Διαδικτύου. Επίσης η ομάδα του Apache δίνει την δυνατότητα σε όποιον έχει την τεχνογνωσία να συμμετάσχει με οποιοδήποτε τρόπο στην ανάπτυξη του web server.

2.5.2 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ APACHE WEB SERVER

Ο Apache Web Server διαθέτει ποικιλία χαρακτηριστικών και μπορεί να υποστηρίξει μια μεγάλη γκάμα εφαρμογών με τις οποίες και συνεργάζεται. Ορισμένα από τα χαρακτηριστικά του αναφέρονται παρακάτω:

§ Unix treading: Υποστήριξη συστημάτων Unix με νήματα POSIX, όπου ο Apache μπορεί να “τρέχει” πολλές διεργασίες ταυτόχρονα.

- § Υποστήριξη πολλαπλών πρωτοκόλλων: Υποστηρίζει πιο γρήγορα και πιο σταθερά λειτουργικά όπως BeOS, OS/2 και Windows.
- § Φίλτρα: Υποστήριξη φίλτρων που διανέμονται από και προς τους διακομιστές.
- § Λάθη: Τα μηνύματα λαθών μπορούν να εμφανίζονται σε διαφορετικές γλώσσες.
- § Απλοποιημένη παραμετροποίηση: Έχουν απλοποιηθεί κάποια directives που ως τώρα ήταν κάπως μπερδεμένα.
- § Υποστήριξη Unicode: Ο Apache 2 σε Windows NT χρησιμοποιεί μόνο utf-8 κωδικοποίηση.
- § Κανονικές εκφράσεις: Υποστήριξη της βιβλιοθήκης δηλαδή όλες οι κανονικές εκφράσεις που υποστηρίζει η Perl 5.



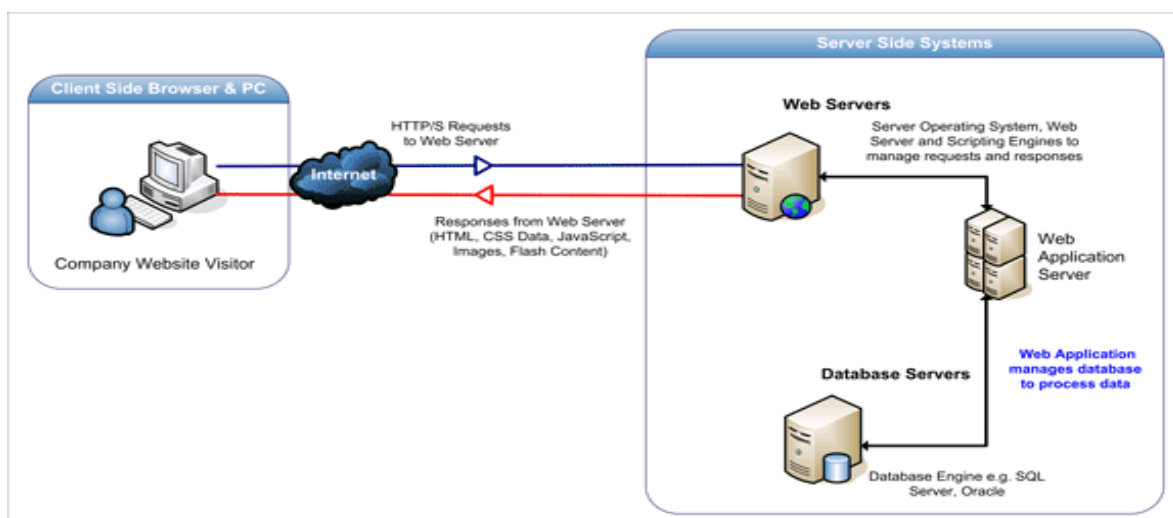
Εικόνα 2.2 (Λογότυπο του Apache)

2.6 CLIENT/SERVER ΑΡΧΙΤΕΚΤΟΝΙΚΗ

Εφόσον το διαδίκτυο στηρίζεται στην αρχιτεκτονική πελάτη/εξυπηρετητή (client/server) κάθε διαδικτυακή εφαρμογή θα λειτουργεί με βάση αυτό το πρότυπο. Αυτό το πρότυπο ουσιαστικά ορίζει ότι τα συνεργαζόμενα κομμάτια μιας διαδικτυακής εφαρμογής μπορούν να κατηγοριοποιηθούν ως client ή ως server. Η client εφαρμογή ζητά υπηρεσίες και δεδομένα από το server και η εφαρμογή server απαντάει στις αιτήσεις του client. Στην περίπτωση που ένας web client (π.χ Mozilla Firefox, Internet Explorer) επικοινωνεί με το web server χρησιμοποιώντας κάποιο πρωτόκολλο (HTTP πρωτόκολλο) κάνοντας

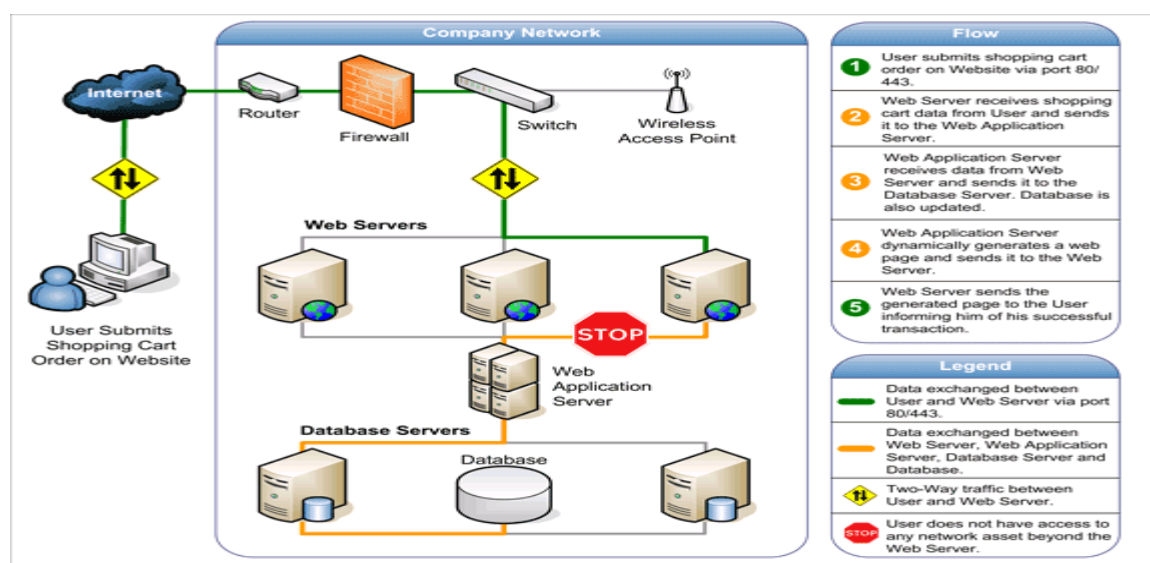
κάποια αίτηση για δεδομένα ή υπηρεσία. Ο web server λαμβάνει την αίτηση του περιηγητή (web browser) και φροντίζει να την εξυπηρετήσει στέλνοντας τα δεδομένα (π.χ στέλνοντας ένα HTML αρχείο στο περιηγητή (web browser) ή τρέχοντας κάποιο πρόγραμμα στο server και αποστέλλοντας τα αποτελέσματα κάποιας επεξεργασίας (π.χ η άντληση δεδομένων από μια βάση δεδομένων).

Το απλό πρότυπο client/server που περιγράφεται παραπάνω ονομάζεται και 2-tier αρχιτεκτονική καθώς υπάρχουν δυο μέρη που παίζουν ρόλο στην λειτουργία ενός συστήματος. Η αρχιτεκτονική 2-tier σε μεγάλες εφαρμογές παρουσιάζει ορισμένα μειονεκτήματα. Υπάρχει η περίπτωση της υπερφόρτωσης του server από τις αιτήσεις του μεγάλου αριθμού client που μπορεί να υπάρχουν. Επίσης πιθανές αλλαγές στην επιχειρησιακή λογική που υλοποιεί μια εφαρμογή, απαιτούν υψηλό κόστος για αλλαγή σε όλη την σειρά διαδικτυακών εφαρμογών όπου αυτή υλοποιείται. Η αντιμετώπιση αυτών των μειονεκτημάτων γίνεται με την εφαρμογή 3-tier (εικόνα που ακολουθεί) ή n-tier αρχιτεκτονικών όπου η υλοποίηση της επιχειρησιακής λογικής και η διαχείριση των δεδομένων γίνεται από δυο ή περισσότερα διαφορετικά συστατικά στοιχεία της διαδικτυακής εφαρμογής.



Εικόνα 2.3 (Αρχιτεκτονική client/server)

Στην εικόνα που ακολουθεί βλέπουμε τον τρόπο που ενεργοποιείται η αρχική αίτηση από τον χρήστη μέσω του περιηγητή (web browser) με τον application server. Η πρόσβαση του server στη βάση δεδομένων για την εκτέλεση των αιτημάτων ενημέρωσης και ανάκτησης βρίσκεται μέσα σε αυτή. Στη συνέχεια η διαδικτυακή εφαρμογή παρουσιάζει τα αποτελέσματα στον χρήστη μέσω του περιηγητή (web browser).



Εικόνα 2.4 (Λειτουργία-επικοινωνία client/server)

2.7 HTML

Η HTML προέρχεται από τις λέξεις HyperText Markup Language (Γλώσσα Χαρακτηρισμού Υπέρ-κειμένου) και βασίζεται στην γλώσσα SGML (Standard Generalized Markup Language) που είναι ένα σύστημα επεξεργασίας εγγράφων.

Η HTML ορίζει κάποια κοινά στυλ για διαδικτυακές σελίδες, όπως επικεφαλίδες (headings), λίστες (lists), τίτλους (titles), παραγράφους (paragraphs) και πίνακες (tables). Ορίζει και κάποια άλλα στυλ χαρακτήρων όπως οι ενότητες κώδικα και η έντονη γραφή (bold).

Τα στοιχεία έχουν από ένα όνομα και τα βάζουμε μέσα στα σύμβολα < > που λέγονται ετικέτες (tags). Με αυτά τα tags δίνουμε

τίτλους στα στοιχεία όταν γράφουμε μια σελίδα με την HTML. Οι διάφοροι φυλλομετρητές λειτουργούν και σαν μορφοποιητές στην HTML με τη δυνατότητα που έχουν να ανακτούν σελίδες από το διαδίκτυο. Όταν ανοίγουμε μια σελίδα που γράφτηκε σε HTML από ένα φυλλομετρητή, ο φυλλομετρητής επεξεργάζεται τα tags της σελίδας και μας εμφανίζει τις εικόνες και το κείμενο στην οθόνη. Τα στυλ σε κάθε στοιχείο μιας σελίδας μπορεί να εμφανίζονται διαφορετικά από διαφορετικούς φυλλομετρητές οι οποίοι βρίσκονται σε διαφορετικά υπολογιστικά συστήματα. Δηλαδή, οι σελίδες που φτιάξαμε σε HTML μπορεί να εμφανίζονται εντελώς διαφορετικά από φυλλομετρητή σε φυλλομετρητή και από υπολογιστικό σύστημα σε υπολογιστικό σύστημα. Ότι γράψαμε θα υπάρχει εκεί αλλά η εμφάνιση της σελίδας στην οθόνη θα είναι διαφορετική.

2.7.1 ΙΣΤΟΡΙΑ HTML

Το 1990 ο Tim Berners-Lee από το Cern, το εργαστήριο φυσικής της Γενεύης, δημιούργησε ένα νέο πρωτόκολλο με το οποίο θα μπορούσαν να μεταφέρονται κάθε είδος αρχείων και αντικειμένων μέσα από το Internet. Το πρωτόκολλο αυτό ονομάστηκε HTTP (HyperText Transfer Protocol) και σηματοδότησε την αρχή του WWW όπως το ξέρουμε σήμερα. Οι σελίδες που ήταν η βάση του WWW ήταν γραμμένες στην πρώτη έκδοση της γλώσσας HTML.

Το 1994 αναπτύσσεται το πρότυπο HTML 2.0 από ένα διεθνή οργανισμό (Internet Engineering Task Force). Η επόμενη έκδοση η 3.0 δεν έγινε αποδεκτή από τις Microsoft και Netscape οπότε γρήγορα αντικαταστάθηκε από την έκδοση 3.2 (1996). Η τελευταία περιελάμβανε πολλές από τις ετικέτες (tags) που είχαν εισάγει οι δυο εταιρίες. Η έκδοση 4.0 παρουσιάστηκε τον Ιούνιο του 1997.

2.7.2 ΔΟΜΗ HTML

Τις σελίδες τις οποίες γράφουμε με HTML είναι απλά αρχεία κειμένου που έχουν την μορφή ASCII, και αυτό σημαίνει ότι δεν έχουμε πληροφορίες για κάποιο λειτουργικό ή πρόγραμμα και μπορούμε να το διαβάσουμε με οτιδήποτε υποστηρίζει ανάγνωση απλού κειμένου.

Τα αρχεία της HTML έχουν τα ακόλουθα:

- § Το κείμενο το οποίο γράφουμε στην σελίδα και
- § Τα tags που μας υποδεικνύουν την δομή και την μορφοποίηση της σελίδας, τα στοιχεία και τους υπερσυνδέσμους που μας μεταφέρουν σε άλλες σελίδες ή σε άλλων μορφών αρχεία (πολυμέσα).

Τα tags έχουν την μορφή:

<tag> το κείμενο μας </tag>

Τα tags έχουν ένα στην αρχή και ένα στο τέλος που συμβολίζουν το άνοιγμα και το κλείσιμο του κειμένου που υπάρχει ανάμεσα τους. Το πρώτο tag ενεργοποιεί την λειτουργία για το οποίο το βάλαμε (όπως να κάνει τα γράμματα μας σε πλάγια μορφή) και το tag τέλους απενεργοποιεί την συγκεκριμένη λειτουργία. Το tag της αρχής είναι το ίδιο με το tag του τέλους με μόνη διαφορά ότι στο τέλος υπάρχει με πρόθεμα ο χαρακτήρας /.

Τα tags δεν αποτελούν πάντα ζευγάρι γιατί υπάρχουν και “μονομελή” αλλά και άλλα tags που μπορούμε να βάλουμε περισσότερα στοιχεία μέσα στα σύμβολα < >. Τα tags μας αφήνουν να γράφουμε με μικρά αλλά και με κεφαλαία χωρίς καμιά διάκριση αλλά συνήθως τα γραφούμε με κεφαλαία για να ξεχωρίζουν από το κείμενο που γράφουμε στην σελίδα μας. Τα διάφορα αρχεία που γράφουμε σε HTML πρέπει να είναι σε επεξεργαστές κειμένου οι οποίοι μπορούν να αποθηκεύσουν σε

μορφή ASCII. Τέτοιοι επεξεργαστές είναι το Notepad, WordPad και Write που χρησιμοποιούμε στα Windows. Υπάρχει και ο Edit που χρησιμοποιείται σε διάφορα λειτουργικά συστήματα όπως DOS και τα vi, Pico και emacs για τα UNIX. Η επέκταση που θα πρέπει να έχει ένα αρχείο HTML θα πρέπει να είναι .html ή htm σε λειτουργικά συστήματα Windows ή DOS που υποστηρίζουν επέκταση τριών χαρακτήρων. Χρησιμοποιεί τρία tags για να περιγράψει την συνολική δομή μιας σελίδας και αυτά προσδιορίζουν την σελίδα μας στους φυλλομετρητές και δίνουν επίσης απλές πληροφορίες όπως τον τίτλο ή τον συγγραφέα πριν την ολική φόρτωση της σελίδας.

2.8 PHP

Η PHP που τα αρχικά της αντιπροσωπεύουν “PHP: Hypertext Preprocessor” είναι μια ευρέως χρησιμοποιούμενη ανοιχτού κώδικα scripting γλώσσα προγραμματισμού και είναι κατάλληλη για ανάπτυξη διαδικτυακών εφαρμογών και μπορεί να ενσωματωθεί στην HTML. Η σύνταξή της παίρνει στοιχεία από διάφορες άλλες γλώσσες όπως οι Java, Perl και C και είναι εύκολη στην μάθηση. Ο κυριότερος λόγος αυτής της γλώσσας είναι να δίνει την δυνατότητα στους web developers να μπορούν να γράφουν δυναμικές σελίδες αλλά ακόμα μπορούν να κάνουν πολύ περισσότερα με αυτή την γλώσσα.

Η PHP επικεντρώνεται κυρίως στο server-side scripting και μπορεί να παράγει δυναμικές σελίδες, να μαζεύει δεδομένα, ή να στείλει και να πάρει cookies.

Υπάρχουν τρεις κύριοι τομείς για τη χρησιμοποίηση ενός PHP Script:

§ Server-side scripting: Είναι το κύριο πεδίο και το πιο παραδοσιακό για την PHP. Χρειάζεται τρία πράγματα για να μπορέσει να δουλέψει, ένα web browser, ένα web server και ένα PHP μεταγλωττιστή. Θα πρέπει να ανοίξουμε τον web

server μαζί με την PHP και μπορούμε να προσπελάσουμε τα αποτελέσματα της PHP χρησιμοποιώντας ένα περιηγητή (web browser) βλέποντας την σελίδα της PHP από τον server.

§ Command line scripting: Μπορεί να φτιαχτεί ένα PHP script χωρίς να έχουμε server ή περιηγητή (web browser) για να το τρέξουμε. Χρειαζόμαστε απλά ένα μεταγλωττιστή και αυτός είναι ένας ιδανικός τρόπος για script που εκτελούνται με χρήση cron σε Linux ή σε Task Scheduler σε Windows. Μπορούν ακόμα αυτά τα script να χρησιμοποιηθούν για απλές εργασίες σε διάφορους επεξεργαστές κειμένου.

§ Εγγραφή client-side GUI εφαρμογών (Γραφικά περιβάλλοντα χρηστών): Η PHP δεν είναι και η καλύτερη γλώσσα για να γράψει κανείς παραθυριακές εφαρμογές αλλά αν ξέρει πολύ καλά την PHP θα μπορεί να χρησιμοποιήσει κάποια προχωρημένα χαρακτηριστικά της στις client-side εφαρμογές.

Την PHP μπορούμε να την χρησιμοποιήσουμε σε όλα τα κύρια λειτουργικά συστήματα (Linux, Microsoft Windows, Unix, MAC OS X, και πιθανώς σε άλλα). Υποστηρίζει τους Apache, Microsoft Internet Information Server, Personal Web Server, Netscape και πολλούς άλλους web server. Υπάρχει η ελευθερία στην επιλογή διαφόρων λειτουργικών συστημάτων και web server.

Η υποστήριξη για ένα μεγάλο σύνολο βάσεων δεδομένων είναι ένα από τα πιο σημαντικά και δυνατά χαρακτηριστικά της PHP, όταν η συγγραφή μιας σελίδας που υποστηρίζει βάσεις δεδομένων είναι εξαιρετικά απλή. Ορισμένες από τις βάσεις δεδομένων που υποστηρίζονται είναι:

§ MySQL

§ mSQL

§ Oracle

§ Solid

§ Sybase

2.8.1 ΙΣΤΟΡΙΑ PHP

Η ιστορία της PHP ξεκινά με ένα φοιτητή τον Rasmus Lerdorf το 1995 που δημιούργησε ένα απλό script χρησιμοποιώντας την γλώσσα Perl με το όνομα `php.cgi`, για προσωπική χρήση. Αυτό το script κατέγραφε από τα άτομα διάφορα στατιστικά που έμπαιναν στο online βιογραφικό του σημείωμα. Με την διάθεση αυτού του script σε φίλους του, οι φίλοι του του ζητούσαν να προσθέσει και άλλες δυνατότητες στο script. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Η PHP/FI έφτασε στην έκδοση 2.0 το 1997 και αυτή την φορά ήταν βασισμένη στην γλώσσα C και είχε περισσότερους από 50.000 ιστότοπους που το χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά ο Zeen Suraski και ο Andi Gutmans έγραψαν ξανά την γλώσσα από την αρχή και έδωσαν περισσότερη σημασία στην PHP/FI 2.0. Η PHP έφτασε στο 3.0 που μοιάζει με την σημερινή της μορφή. Ο Zeen και ο Andi έφτιαξαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους) η οποία εξελίσσει και αναπτύσσει ακόμα και σήμερα την γλώσσα της PHP. Η έκδοση 4 της εμφανίστηκε το 1998, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5 ενώ αυτή την περίοδο διατέθηκαν και πρώτες δοκιμαστικές εκδόσεις της PHP 6 για τον προγραμματιστή που θέλει να την χρησιμοποιήσει. Οι πιο πολλοί ιστότοποι χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5.



Εικόνα 2.5 (Λογότυπο της PHP)

2.9 JAVASCRIPT

Η Javascript είναι μια γλώσσα προγραμματισμού η οποία παράγει δυναμικού περιεχομένου ιστοσελίδες. Ξεκίνησε από την ECMAScript από την οποία επεκτάθηκε με περισσότερες πρόσθετες δυνατότητες.

Όπως και η PHP, έτσι και η Javascript βασίζεται στον τρόπο σύνταξης της στην γλώσσα προγραμματισμού C, οι οποίες έχουν πολλά κοινά. Η επεξεργασία του κώδικα της Javascript και η παρουσίαση του τελικού περιεχομένου δεν γίνεται στον server όπως στην PHP αλλά στον περιηγητή (web browser) του επισκέπτη. Αυτό φυσικά θα είχε και πλεονεκτήματα και μειονεκτήματα για την καθεμιά γλώσσα. Η Javascript δεν έχει ανάγκη τις δυνατότητες που έχει ο server για να εκτελέσει κάτι (όπως ισχύ, συμβατό λογισμικό), αλλά στηρίζεται στις δυνατότητες του περιηγητή (web browser) του επισκέπτη. Ακόμα η Javascript θα μπορούσε να ενσωματωθεί και σε στατικές σελίδες HTML. Οι δυνατότητες της είναι πολύ μικρότερες από αυτές που θα μπορούσε να κάνει η PHP όπως για παράδειγμα η σύνδεση με βάσεις δεδομένων.

Η Javascript δεν θα πρέπει να συνδέεται με την Java γιατί είναι διαφορετικές γλώσσες προγραμματισμού και με διαφορετικές εφαρμογές. Ο σωστός τρόπος γραφής της είναι “Javascript” και όχι ‘Java script’ όπως λανθασμένα γράφεται από πολλούς ορισμένες φορές.

2.9.1 ΙΣΤΟΡΙΑ JAVASCRIPT

Η Javascript αρχικά αναπτύχθηκε από τον Brendan Eich της Netscape με το όνομα Mocha, η οποία στην συνέχεια μετονομάστηκε Livescript και μετά πήρε το τελικό της όνομα σε Javascript. Η JavaScript μπήκε για πρώτη φορά στον περιηγητή (web browser) του Netscape το Δεκέμβριο του 1995 και είναι αποκλειστικό προϊόν της Sun Microsystems. Χρησιμοποιούνται κατόπιν αδείας επειδή εφηύρε την τεχνολογία και εφαρμόζεται από την Netscape Communications και από

το Mozilla Foundation. Η Javascript έχει γίνει πλέον μια από τις πιο διαδεδομένες γλώσσες προγραμματισμού στον παγκόσμιο ιστό. Αρχικά έγινε πολλή δυσφήμιση στην Javascript από πολλούς επαγγελματίες προγραμματιστές επειδή στόχος της ήταν οι web editors και άλλοι ερασιτέχνες. Με την έλευση της Ajax επανήλθε και η Javascript στα φώτα της δημοσιότητας με πιο επαγγελματικό προγραμματισμό. Το αποτέλεσμα ήταν η εξάπλωση των frameworks, η βελτίωση του προγραμματισμού της Javascript, και η αυξημένη χρήση της εκτός από τους περιηγητές (web browsers). Τον Ιανουάριο του 2009, η CommonJS project ιδρύθηκε με στόχο την δημιουργία κοινοποιημένων υποδομών στήριξης βιβλιοθήκης με προδιαγραφές βασισμένες κυρίως στις λειτουργίες της JavaScript και σε άλλες κατηγορίες εκτός από τους περιηγητές.

2.9.2 ΔΕΙΓΜΑ ΚΩΔΙΚΑ JAVASCRIPT

Ο κώδικας της Javascript μπαίνει μέσα σε HTML tags `<script language=javascript>` και `</script>`. Θα σας δείξουμε ένα απλό δείγμα της Javascript με αποτέλεσμα την προβολή της φράσης Hello World!.

```
<script language=javascript>  
    document.write('Hello world!');  
</script>
```

Αποτέλεσμα:

Hello world!

2.10 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ MYSQL

2.10.1 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Μια βάση δεδομένων είναι μια δομημένη συλλογή από δεδομένα. Μπορεί να είναι οτιδήποτε από ένα απλό κατάλογο αγορών ως μια

συλλογή φωτογραφιών ή τα απέραντα ποσά πληροφοριών σε ένα εταιρικό δίκτυο. Βάση δεδομένων είναι η οργάνωση και καταχώριση της πληροφορίας σε τρόπο τέτοιο ώστε να ενημερώνεται και να ανακαλείται με τον πλέον ευέλικτο τρόπο. Αποτελεί δηλ. την συλλογή δεδομένων τα οποία έχουν να κάνουν με τις δραστηριότητες ενός οργανισμού. Για παράδειγμα, βάση δεδομένων αποτελούν τα επιχειρησιακά δεδομένα ενός πανεπιστημίου που καταχωρούν πληροφορία για :

- § Οντότητες στο χώρο του συστήματος, όπως φοιτητές, ακαδημαϊκοί, μαθήματα και αίθουσες διδασκαλίας.
- § Συσχετίσεις μεταξύ οντοτήτων, όπως η συμμετοχή φοιτητών σε μαθήματα, η διδασκαλία μαθημάτων από καθηγητές και η χρήση των αιθουσών διδασκαλίας.

2.10.2 ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Η εξέλιξη των Βάσεων Δεδομένων και οι ανάγκες για δημιουργία όλο και περισσότερων Βάσεων Δεδομένων, οδήγησαν στη δημιουργία των Συστημάτων Διαχείρισης Βάσεων Δεδομένων (Database Management Systems ή DBMS). Το Σύστημα Διαχείρισης Βάσεων Δεδομένων είναι ένα εργαλείο το οποίο διευκολύνει τους χρήστες να εργάζονται με Βάσεις Δεδομένων. Με τη χρήση του Συστήματος Διαχείρισης Βάσεων Δεδομένων οι χρήστες μπορούν να κατασκευάσουν και να χρησιμοποιήσουν Βάσεις Δεδομένων. Ακολουθεί ο ορισμός του Συστήματος Διαχείρισης Βάσεων Δεδομένων.

Ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων κατά κανόνα «φιλοξενεί» πολλές Βάσεις Δεδομένων που έχουν κατασκευαστεί από διαφορετικούς χρήστες. Οι δυνατότητες που παρέχει ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων στους χρήστες συνοψίζονται στις παρακάτω:

- § Ορισμός της Βάσης Δεδομένων

§ Κατασκευή της Βάσης Δεδομένων

§ Διαγραφή της Βάσης Δεδομένων

§ Χρήση της Βάσης Δεδομένων

Στον **ορισμό** της Βάσης Δεδομένων ο χρήστης μπορεί να καθορίσει το μοντέλο της Βάσης Δεδομένων, να ορίσει τους τύπους δεδομένων που θα χρησιμοποιήσει και να ελέγξει τη Βάση Δεδομένων χωρίς να προχωρήσει σε κατασκευή της. Στην **κατασκευή** της Βάσης Δεδομένων οι τύποι των δεδομένων και τα δεδομένα αποθηκεύονται στο υλικό (hardware) του ηλεκτρονικού υπολογιστή με διαδικασίες που ελέγχονται από το Σύστημα Διαχείρισης Βάσεων Δεδομένων και δεν απασχολούν τον χρήστη. Στη **διαγραφή** της Βάσης Δεδομένων ο χρήστης αποφασίζει τον τερματισμό μιας Βάσης Δεδομένων και την απομάκρυνση των δεδομένων από το υλικό. Τέλος, κατά τη **χρήση** της Βάσης Δεδομένων ο χρήστης, είτε χειρίζεται τα δεδομένα (εισάγει νέα δεδομένα, τροποποιεί δεδομένα, ή διαγράφει δεδομένα), είτε υποβάλει ερωτήσεις στη Βάση Δεδομένων με στόχο την εξαγωγή πληροφοριών.

2.10.3 ΙΣΤΟΡΙΑ ΤΗΣ MYSQL

Η IBM ανέπτυξε την SQL με τους Donald C. Messerly, Raymond F. Boyce και Andrew Richardson, στις αρχές της δεκαετίας του 1970. Αυτή η έκδοση ονομαζόταν στην αρχή SEQUEL και είχε ως στόχο να διαχειριστεί και να ανακτήσει τα διάφορα στοιχεία που είχαν αποθηκευτεί στο πρώτο RDBMS της IBM, το System R.

Το πρώτο σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων ήταν το RDBMS το οποίο αναπτύχθηκε στο MIT ('Massachusetts Institute of Technology' και στα ελληνικά 'Τεχνολογικό Ινστιτούτο Μασαχουσέτης') στις αρχές της δεκαετίας του 1970 και η Ingres που αναπτύχθηκε το 1974 στο Πανεπιστήμιο Μπέρκλεϋ. Η Ingres

χρησιμοποιούσε στην αρχή μια γλώσσα διατύπωσης ερωτήσεων με το όνομα QUEL, το οποίο μετά αντικαταστάθηκε από τη SQL.

Η Relation Software που τώρα είναι η Oracle Corporation στα τέλη της δεκαετίας του 70 είδαν τις δυνατότητες που περιέγραψαν οι Chamberlin, Codd και Boyce και ανέπτυξε την SQL βασισμένο στο RDBMS, με φιλοδοξίες πώλησης στην Κεντρική Υπηρεσία Πληροφοριών, στο Αμερικάνικο Ναυτικό και σε άλλες Αμερικάνικες Υπηρεσίες.

Η Relation Software το καλοκαίρι του 1979 έγινε κάτοχος της πρώτης εφαρμογής της SQL που διατέθηκε στο εμπόριο και κατάφερε και νίκησε την IBM για μερικές βδομάδες με την διάθεση του εμπορικού RDBMS.

2.10.4 ΠΩΣ ΛΕΙΤΟΥΡΓΕΙ Η SQL

Η SQL, παρέχει οφέλη και δυνατότητες προς όλους τους τύπους των χρηστών, συμπεριλαμβανομένων των προγραμματιστών εφαρμογών, των διαχειριστών βάσεων δεδομένων, των διευθυντών και των τελικών χρηστών. Από τεχνικής απόψεως, η SQL η γλώσσα που χρησιμοποιείται για να χειριστεί και να ανακτήσει δεδομένα από τις σχεσιακές βάσεις δεδομένων. Σκοπός της είναι να παρέχει μια διεπαφή σε μια σχεσιακή βάση δεδομένων όπως η MySQL και όλες οι SQL δηλώσεις είναι οδηγίες προς τη βάση δεδομένων, κάτι που τη διαφοροποιεί από τις γενικής χρήσεως γλώσσες προγραμματισμού, όπως η C και η Basic. Μεταξύ των χαρακτηριστικών γνωρισμάτων της SQL είναι και τα εξής :

- § Επεξεργάζεται τα σύνολα των δεδομένων ως ομάδες παρά ως μεμονωμένες μονάδες.
- § Παρέχει αυτόματη περιήγηση στα δεδομένα.
- § Χρησιμοποιεί δηλώσεις που είναι πολύπλοκες και ισχυρές χωριστά και για αυτό και στέκονται από μόνα τους.

Ουσιαστικά η SQL, επιτρέπει την εργασία με τα δεδομένα στο λογικό επίπεδο. Για παράδειγμα, η ανάκτηση ενός συνόλου γραμμών από ένα πίνακα, ορίζει έναν κανόνα που χρησιμοποιείται για να φιλτράρει το σύνολο των γραμμών του πίνακα. Όλες οι γραμμές που ικανοποιούν τον κανόνα αυτό, ανακτώνται σε ένα και μόνο βήμα και μπορούν να περαστούν ως μονάδα στο χρήστη, σε μια άλλη SQL δήλωση ή σε μια εφαρμογή. Δεν χρειάζεται να εξεταστούν οι γραμμές μια προς μια, αλλά δεν υπάρχει και ανησυχία για την αποθήκευση των δεδομένων με όποιο τρόπο και αν γίνεται αυτή, φυσικά ή με ανάκτηση. Η SQL παρέχει δηλώσεις για μια ποικιλία από εργασίες, όπως:

- § Διατύπωση ερωτήσεων δεδομένων.
- § Εισαγωγή, ανανέωση και διαγραφή γραμμών σε έναν πίνακα.
- § Δημιουργία, αντικατάσταση, τροποποίηση και διαγραφή αντικειμένων.

Επίσης επιτρέπει σε έναν προγραμματιστή ή σε ένα διαχειριστή της βάσεως να κάνει τα παρακάτω :

- § Να τροποποιήσει το σχήμα της βάσης δεδομένων.
- § Να αλλάξει τις ρυθμίσεις ασφάλειας του συστήματος.
- § Να προσθέσει δικαιώματα χρηστών στις βάσεις ή στους πίνακες.
- § Να διατυπώσει ερωτήματα για την ανάκτηση πληροφοριών.
- § Να ανανεώσει τα περιεχόμενα της βάσης.

2.10.5 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ MYSQL

Η αποθήκευση και ανάκτηση δεδομένων είναι ένα βασικό στοιχείο των περισσότερων εφαρμογών σήμερα. Στις παλιότερες ημέρες ανάπτυξης λογισμικού, οι προγραμματιστές έγραφαν το δικό τους κώδικα χαμηλού επιπέδου για να το καταφέρουν. Εντούτοις, γρήγορα αντιλήφθηκαν ότι σε κάθε εφαρμογή εφεύρισκαν πάλι ουσιαστικά, τη

ρόδα. Μέσω του συνηθισμένου κύκλου της δοκιμής, του λάθους και του επόμενου καθορισμού, μια λύση αναπτύχθηκε : η μηχανή αποθήκευσης και ανάκτησης των δεδομένων ενσωματώθηκε σε έναν αυτόνομο server βάσεως δεδομένων με τους πελάτες να συνδέονται σε αυτό και στέλνοντας αιτήματα σε μια γλώσσα την SQL. Σήμερα οι προγραμματιστές μπορούν να επιλέξουν μέσα από πολλά προϊόντα αποθήκευσης και ανάκτησης δεδομένων που χρησιμοποιούν τη γλώσσα SQL. Αυτά τα προϊόντα αναφέρονται συνήθως ως server βάσεων δεδομένων SQL ή μερικές φορές ως σχεσιακά συστήματα διαχείρισης βάσεων δεδομένων. Μια σχεσιακή βάση δεδομένων αποθηκεύει τα δεδομένα σε ξεχωριστούς πίνακες αντί να τα βάζει σε μια μεγάλη αποθήκη.

Αυτό προσθέτει ταχύτητα. Οι πίνακες συνδέονται μεταξύ τους κάνοντας εφικτό το συνδυασμό δεδομένων από πολλούς πίνακες κατόπιν αιτήσεως. Αντιθέτως με τους περισσότερους server βάσεων δεδομένων, η MySQL είναι ένα προϊόν ανοικτού κώδικα : ο πηγαίος κώδικας του είναι ελεύθερα διαθέσιμος για κατέβασμα στο καθένα. Οι προγραμματιστές μπορούν να αλλάξουν τον πηγαίο κώδικα έτσι ώστε να προσαρμόσουν τη MySQL στις ανάγκες τους. Ένα από τα καλά των προϊόντων ανοικτού κώδικα είναι ότι ένα ευρύ φάσμα προγραμματιστών και χρηστών συμβάλλουν την εμπειρία τους στο λογισμικό, κάνοντας το καλύτερο. Η MySQL έχει μια μεγάλη κοινότητα πιστών υποστηρικτών.

Η πιο κοινή ανησυχία για τα προϊόντα ανοιχτού κώδικα είναι ότι δεν έχουν μια εμπορική οντότητα από πίσω τους που θα έχει την ευθύνη για το λογισμικό. Από αυτή την άποψη, η MySQL είναι ένα από τα λίγα προϊόντα ανοιχτού λογισμικού. Η MySQL AB είναι μια ολοκληρωμένη επιχείρηση, που απασχολεί 50 άτομα σε όλο τον κόσμο τα οποία είναι υπεύθυνα για την ανάπτυξη, την υποστήριξη, τις πωλήσεις, την κατάρτιση, την τεκμηρίωση και τις επιχειρησιακές λειτουργίες. Το

σύστημα βάσεων δεδομένων της MySQL χρησιμοποιεί την αρχιτεκτονική πελάτη-εξυπηρετητή που συγκεντρώνεται γύρω από τον server, *myaql*, υποστηρίζει πολλά διαφορετικά προγράμματα και βιβλιοθήκες χρηστών, εργαλεία διαχείρισης και πολλές προγραμματιστικές διεπαφές. Ο server είναι στη πραγματικότητα το πρόγραμμα που διαχειρίζεται τις βάσεις δεδομένων. Τα προγράμματα των πελατών δεν το κάνουν αυτό άμεσα, μάλλον, διαβιβάζουν την πρόθεσή τους στο server με την βοήθεια ερωτήσεων γραμμένων σε SQL. Το πρόγραμμα ή τα προγράμματα των χρηστών μπορούν να αποθηκευτούν τοπικά στη μηχανή, από την οποία θα έχουν πρόσβαση στη MySQL. Αντιθέτως server μπορεί να εγκατασταθεί οπουδήποτε αρκεί οι χρήστες να μπορούν να συνδεθούν σε αυτό. Η MySQL είναι ένα εγγενώς δικτυωμένο σύστημα βάσεων δεδομένων, με το οποίο οι χρήστες μπορούν να επικοινωνήσουν με τον server είτε αυτός είναι εγκατεστημένος τοπικά στο μηχάνημα τους είτε σε κάποιο άλλο μηχάνημα, ακόμη και στην άλλη μεριά του πλανήτη. Όταν η χρήση είναι αμφίδρομη, η MySQL διενεργεί ερωτήματα τα οποία αποστέλλει στη MySQL server για εκτέλεση και έπειτα εμφανίζει τα αποτελέσματα. Η MySQL μπορεί να χρησιμοποιηθεί και μη-αμφίδρομα, παραδείγματος χάριν, να διαβάσει τις ερωτήσεις από ένα αρχείο ή από άλλα προγράμματα.

2.10.6 ΚΥΡΙΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ MYSQL

Η MySQL έχει κάποια σημαντικά χαρακτηριστικά που την διακρίνουν όπως η φορητότητα, δηλ. η δυνατότητα της εύκολης μεταφοράς των αρχείων από βάση σε βάση, με μηδαμινή διαρροή μνήμης. Επίσης έχει την δυνατότητα να δουλέψει σε διαφορετικές πλατφόρμες. Μπορεί να δουλέψει σε γλώσσες όπως η C, C++, Java, Perl, Python και TCL. Αυτό σε συνδυασμό ότι μπορεί εύκολα να

χρησιμοποιήσει πολλαπλές CPU την καθιστούν ένα πολύ δυνατό εργαλείο. Επίσης διακρίνεται για το πολύ γρήγορο σύστημα κατανομής μνήμης και για την πολύ γρήγορη δενδρική αναζήτηση (B-tree) στους πίνακες τις με την μέθοδο της συμπίεσης δεικτών. Τέλος οι SQL λειτουργίες εφαρμόζονται μέσω μιας ιδιαίτερα βελτιστοποιημένης βιβλιοθήκης και πρέπει να είναι όσο το δυνατό γρήγορη.

2.10.7 ΙΣΤΟΡΙΑ ΤΗΣ ΒΑΣΗΣ MYSQL

Η ομάδα που δημιούργησε τη MySQL ξεκίνησε με την πρόθεση να χρησιμοποιήσει τη mSQL για να συνδεθούν σε πίνακες χρησιμοποιώντας τις χαμηλού επιπέδου ρουτίνες τους. Ωστόσο καταλήξανε στο ότι η mSQL δεν είναι αρκετά γρήγορη για τις ανάγκες τους. Αυτό τους οδήγησε σε μια νέα SQL διεπαφή στη βάση δεδομένων αλλά με σχεδόν ίδια API διεπαφή όπως η mSQL, η νέα αυτή διεπαφή ονομάστηκε MySQL.

2.10.8 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΗΣ MYSQL:

Είναι αρκετά και σημαντικά τα πλεονεκτήματα που προσφέρει η χρήση της βάσης MySQL. Πιο συγκεκριμένα τα πλεονεκτήματα της MySQL παραθέτονται παρακάτω:

- § Ταχύτητα: Ο κώδικας του πυρήνα της MySQL γράφτηκε από την αρχή μέχρι το τέλος με άριστη απόδοση ως αρχικός στόχος.
- § Αξιοπιστία: Η MySQL έχει κερδίσει τη φήμη του ότι είναι σε θέση να τρέχει αφύλακτο για μέρες ακόμη και μήνες μετά από την αρχική οργάνωση.
- § Χαμηλές απαιτήσεις σε πόρους συστήματος: Η MySQL είναι ικανό να κάνει το καλύτερο με τους πόρους που του δίνεται. Φυσικά, όσο περισσότεροι πόροι, τόσο καλύτερη απόδοση να περιμένετε, αλλά ελάχιστοι πόροι δεν θα βγάλουν τη MySQL εκτός λειτουργίας.

- § **Εξελιξιμότητα:** Η εμπειρία έδειξε ότι η MySQL αποδίδει καλά σε συστήματα με μέχρι 4 επεξεργαστές και μνήμη μέχρι 4GB εκμεταλλεύόμενο πλήρως τους πόρους του συστήματος.
- § **Ποικιλομορφία πλατφόρμων:** Τρέχει σε μια ευρεία ποικιλία υπολογιστικών συστημάτων. Ανάμεσα στα οποία τα πιο δημοφιλή είναι τα Linux, Windows, Solaris και FreeBSD.
- § **Υποστήριξη για έναν μεγάλο αριθμό γλωσσών :** Όταν αναπτύσσετε μια εφαρμογή με την χρήση της βάσης δεδομένων, μια από τις αρχικές ανησυχίες είναι η διασύνδεση με τον server βάσεων δεδομένων χρησιμοποιώντας κάποια γλώσσα προγραμματισμού.
- § **Αυτός είναι ακόμη ένας τομέας της δύναμης της MySQL.** Οι προγραμματιστές μπορούν να επικοινωνήσουν με τη MySQL χρησιμοποιώντας C/C++, PHP, Perl, Java, Python, TCL, Ruby και Eiffel.
- § **Υποστήριξη ODBC:** Περιλαμβάνει οδηγό ODBC. Αυτό δίνει τη δυνατότητα στους προγραμματιστές να γράψουν εφαρμογές χρησιμοποιώντας Open Database Connectivity. Το ODBC επιτρέπει στη MySQL να χρησιμοποιηθεί στο Microsoft Access, Excel και άλλα. Επίσης επιτρέπει στη MySQL να χρησιμοποιηθεί σε εφαρμογές σε Visual Basic και Delphi, στην ASP και σε άλλα εργαλεία και περιβάλλοντα ανάπτυξης εφαρμογών.
- § **Δωρεάν ή χαμηλού κόστους χορήγηση αδειών:** Η MySQL διανέμεται κάτω από την άδεια GPL (General Public License). Μέσω της άδειας αυτής μπορεί να χρησιμοποιηθεί και για εμπορικό και για μη εμπορικό σκοπό υπό την προϋπόθεση ότι οποιοδήποτε παραγόμενο προϊόν πρέπει να διανεμηθεί με

ολόκληρο τον πηγαίο κώδικα κάτω από τους όρους της ίδιας άδειας.

- § Ανέξοδη εμπορική υποστήριξη: Για όσους σκέφτονται να τρέξουν τη MySQL σε ένα περιβάλλον, το ζήτημα υψηλής ποιότητας εμπορικής υποστήριξης είναι πολύ σημαντικό. Η MySQL AB παρέχει ένα ευρύ φάσμα εμπορικής υποστήριξης σε λογική τιμή, συμπεριλαμβανομένου 24x7 τηλεφωνική υποστήριξη.
- § Ισχυρή υποστήριξη κοινοτήτων χρηστών: Όπως αναφέρθηκε πιο πάνω η MySQL δεν είναι απλά μια βάση δεδομένων. Οι ιδρυτές της MySQL έχουν εστιάσει τη προσοχή τους στο να δίνουν στη κοινότητα περισσότερα από αυτά που παίρνουν από αυτήν. Η κοινότητα ανταποκρίθηκε με αφοσίωση, σκληρή δουλειά και συναδελφικότητα. Έτσι μπορεί κάποιος χρήστης να βρει τις απαντήσεις στα προβλήματα που δεν μπορεί να επιλύσει λαμβάνοντας βοήθεια από τη κοινότητα, πέρα από την βοήθεια που υποστηρίζει η MySQL. Μια άλλη πτυχή της ισχυρής κοινότητας είναι το ότι μπορεί να βρει κανείς ένα ειδικό στη MySQL για να εργαστεί για εσάς.
- § Διαθεσιμότητα του πηγαίου κώδικα: Η πρόσβαση στον πηγαίο κώδικα είναι κάτι πολύ σημαντικό για τις επιχειρήσεις που απασχολούν πεπειραμένους C/C++ προγραμματιστές. Παρέχει την ευκαιρία να γίνουν διάφορες προσαρμογές, βελτιώσεις, επεκτάσεις και διόρθωση λαθών, χωρίς να περιμένεις των προμηθευτή να το κάνει.

2.10.9 ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΗΣ MYSQL:

Στην προηγούμενη παράγραφο αναφέραμε τα πλεονεκτήματα της βάσης δεδομένων MySQL. Στη συνέχεια καταγράφονται κάποια μειονεκτήματα της:

- § Έλλειψη ορισμένων χαρακτηριστικών γνωρισμάτων της SQL:
Η πιο σοβαρή αδυναμία της MySQL είναι ότι στην συγκεκριμένη έκδοση δεν υπάρχει η δυνατότητα ενσωμάτωσης ερωτημάτων, όψεων, αποθηκευμένων διαδικασιών, εναυσμάτων και επιβολής ξένου κλειδιού.
- § Έλλειψη λεπτομερούς δοκιμής σε συγκεκριμένες πλατφόρμες:
Η MySQL AB μπορεί να εξασφαλίσει ως ένα επίπεδο την ποιότητα της MySQL σε μια πλατφόρμα από εκεί και μετά, η ποιότητα αυτή θα αυξάνεται όσοι περισσότεροι χρησιμοποιούν τη MySQL στη πλατφόρμα αυτή και αποστέλλουν τα σφάλματα που βρίσκουν. Επομένως η εξασφάλιση της ποιότητας εξαρτάται κατά ένα μεγάλο μέρος από τις δοκιμές που κάνουν οι χρήστες, οπότε όσοι περισσότεροι χρήστες τόσο λιγότερα σφάλματα θα περάσουν απαρατήρητα, για μεγάλο χρονικό διάστημα. Βέβαια αυτό δεν σημαίνει ότι σε πλατφόρμες που δεν χρησιμοποιούν πολλοί χρήστες τη MySQL δεν θα τρέχει και μάλιστα καλά, αλλά ίσως να υπάρξουν κάποια προβλήματα.
- § Δυσκολία εργασίας με τον πηγαίο κώδικα του server: Το δύσκολο του να δουλεύει κανείς με τον πηγαίο κώδικα που έγραψε κάποιος άλλος είναι ότι πρέπει να τον καταλάβει για να υπάρχει κάποια πρακτική αξία. Τον κώδικα της MySQL server είναι πολύ δύσκολο να τον καταλάβει ακόμη και ένας ειδικευμένος πεπειραμένος προγραμματιστής της C/C++. Υπάρχουν δυο κύριοι λόγοι της δυσκολίας αυτής. Πρώτα, είναι ένας server βάσεων δεδομένων, το οποίο σημαίνει ότι πρέπει αν

έχει το κώδικα για να οργανώσει τα δεδομένα στο σκληρό δίσκο και να επιλέξει μια στρατηγική για την επίλυση ενός συγκεκριμένου ερωτήματος.



Εικόνα 2.6 (Λογότυπο της MySQL)

3 ΚΥΚΛΟΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΗΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΒΙΒΛΙΟΘΗΚΗΣ

Λογισμικό είναι ένα σύνολο από εντολές προς τον υπολογιστή και κατά την εκτέλεσή τους παράγουν επιθυμητές αποδόσεις και λειτουργίες, επιτρέποντας την ικανοποιητική διαχείριση πληροφοριών με τις δομές δεδομένων και τα έγγραφα τα οποία περιγράφουν την χρήση και την λειτουργία διαφόρων προγραμμάτων.²²

Τεχνολογία λογισμικού είναι ένας τομέας που εξετάζει τις μεθοδολογίες, τις πρακτικές, τις τεχνικές και τα εργαλεία για την μεθοδική, την συστηματική και ποσοτικοποιημένη προδιαγραφή, τη σχεδίαση, την ανάπτυξη, τον έλεγχο και τη συντήρηση συστημάτων λογισμικού.

§ Μεθοδολογία είναι μια συλλογή από τεχνικές που εφαρμόζονται κατά την διάρκεια των φάσεων ενός έργου και συνδυάζονται με κάποιο πλάνο και μια γενική πρακτική.

§ Τεχνική είναι φορμαλιστικές διαδικασίες με τη χρήση ορισμένων συστημάτων παρουσίασης και κωδικοποίησης για την επίλυση διαφόρων προβλημάτων.

§ Εργαλεία είναι τα αυτοματοποιημένα συστήματα που επιλύουν μια τεχνική.

Ο κύκλος ανάπτυξης λογισμικού χωρίζεται σε πέντε φάσεις τις οποίες

αναλύουμε παρακάτω:

3.1 ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ

Η ανάλυση απαιτήσεων καλύπτει τους στόχους για τον καθορισμό των αναγκών που απαιτούνται για ένα νέο ή υπάρχον προϊόν και είναι πολύ κρίσιμη για την ανάπτυξη κάποιου έργου. Οι απαιτήσεις πρέπει να

είναι σχετικές με τις ευκαιρίες ή τις ανάγκες και ικανοποιητικά λεπτομερείς για το σχεδιασμό της εφαρμογής.

Η ανάλυση απαιτήσεων χωρίζεται σε τρεις τύπους:

- § Απόκτηση απαιτήσεων: η εύρεση των απαιτήσεων από τον σχεδιαστή του λογισμικού σύμφωνα με την εφαρμογή που του έχει ανατεθεί για ανάπτυξη
- § Ανάλυση απαιτήσεων: καθοριστικό κομμάτι ειδικά αν οι απαιτήσεις είναι ελλιπείς ή ασαφείς και ο τρόπος που θα πρέπει να εξεταστούν αυτές
- § Καταγραφή απαιτήσεων: οι απαιτήσεις μπορούν να τεκμηριωθούν σε διάφορες μορφές, όπως σε φυσική γλώσσα ή σε πιο εξειδικευμένη μορφή

3.1.1 ΕΦΑΡΜΟΓΗ ΧΡΗΣΤΩΝ

Η εφαρμογή των χρηστών αποτελεί την διεπαφή των συνδρομητών προς τις υπηρεσίες που παρέχει η ψηφιακή βιβλιοθήκη που είναι οι εξής:

- § Ο χρήστης πρέπει να έχει τη δυνατότητα να μπορεί να επιλέξει το βιβλίο που τον ενδιαφέρει μέσα από ένα σύνολο κατηγοριών έτσι ώστε να γίνεται ευκολότερη η περιήγησή του.
- § Για την διευκόλυνση του χρήστη στην εύρεση του βιβλίου που τον ενδιαφέρει θα υπάρχει μια αναζήτηση, όπου ο χρήστης θα μπορεί να το αναζητήσει είτε γράφοντας τον τίτλο του βιβλίου, είτε τον συγγραφέα ή ακόμα και επιλέγοντας την γενικότερη κατηγορία που ανήκει το βιβλίο.
- § Πρέπει να υπάρχει ένα σύστημα εγγραφής και σύνδεσης με διαφορετικά δικαιώματα για κάθε χρήστη. Συγκεκριμένα κάθε χρήστης που εγγράφεται θα έχει ένα διαφορετικό ένα username και ένα password που με αυτό θα μπορεί να συνδεθεί με το

σύστημα επιτρέποντάς του να προβεί στις ενέργειες που επιθυμεί.

Κατά την εγγραφή του ο κάθε χρήστης θα μπορεί να επιλέξει ανάμεσα από τρεις επιλογές. Κάθε επιλογή θα παραχωρεί στο χρήστη και διαφορετικά δικαιώματα αναλόγως του ποσού που θα επιθυμεί να καταβάλει ως συνδρομή. Για να επιλέξει μια από τις τρεις επιλογές ο χρήστης θα πρέπει να συμπληρώσει μια φόρμα με τα προσωπικά του στοιχεία καθώς και τα στοιχεία της πιστωτικής του κάρτας για να πληρώσει το ποσό που αντιστοιχεί στην συνδρομή της επιλογής που έχει επιλέξει.

Οι τρεις επιλογές του χρήστη είναι οι ακόλουθες:

3.1.1.1 ΧΡΗΣΤΗΣ ΜΕ ΑΠΕΡΙΟΡΙΣΤΗ ΠΡΟΣΒΑΣΗ

Ο Χρήστης με απεριόριστη πρόσβαση έχει πρόσβαση στις υπηρεσίες της ψηφιακής βιβλιοθήκης χωρίς περιορισμούς .Συγκεκριμένα έχει τα εξής δικαιώματα:

- § Ο χρήστης μπορεί να έχει πλήρη πρόσβαση σε όλα τα βιβλία καθώς και γρήγορη μετακίνηση για να βρει τις σχετικές πληροφορίες για τις οποίες ενδιαφέρεται.
- § Μπορεί να κατεβάσει ολόκληρο το βιβλίο σε μορφή PDF για να το διαβάσει.
- § Ο χρήστης θα μπορεί να κατεβάσει όσα βιβλία επιθυμεί το μήνα σε μορφή PDF.
- § Ο χρήστης θα πρέπει να πληρώσει το αντίστοιχο χρηματικό ποσό.

3.1.1.2 ΧΡΗΣΤΗΣ ΜΕ ΠΕΡΙΟΡΙΣΜΕΝΗ ΠΡΟΣΒΑΣΗ

Ο Χρήστης με περιορισμένη πρόσβαση έχει πρόσβαση στις υπηρεσίες της ψηφιακής βιβλιοθήκης με ορισμένους περιορισμούς .Συγκεκριμένα έχει τα εξής δικαιώματα:

- § Μπορεί να αναζητήσει οποιοδήποτε βιβλίο τον ενδιαφέρει αλλά θα μπορεί να κατεβάσει μόνο 10 βιβλία το μήνα σε μορφή PDF.
- § Ο χρήστης θα πρέπει να πληρώσει το αντίστοιχο χρηματικό ποσό.

3.1.1.3 ΔΟΚΙΜΑΣΤΙΚΟΣ ΧΡΗΣΤΗΣ

Ο Δοκιμαστικός Χρήστης έχει πρόσβαση στις υπηρεσίες της ψηφιακής βιβλιοθήκης δοκιμαστικά με αρκετούς περιορισμούς. Συγκεκριμένα έχει τα εξής δικαιώματα:

- § Ο χρήστης θα μπορεί να κατεβάσει μόνο 3 βιβλία το μήνα.
- § Ο χρήστης κατά την εγγραφή του χρειάζεται να δώσει μόνο το e-mail του.
- § Ο χρήστης δεν χρειάζεται να πληρώσει κάποιο χρηματικό ποσό.

3.1.2 ΕΦΑΡΜΟΓΗ ΕΚΔΟΤΙΚΩΝ ΟΙΚΩΝ

Η Εφαρμογή Εκδοτικών Οίκων αποτελεί την διεπαφή της ψηφιακής βιβλιοθήκης με τους εκδοτικούς οίκους για την αυτοματοποίηση της διαδικασίας αποστολής και έγκρισης καινούριων βιβλίων. Συγκεκριμένα οι εκδοτικοί οίκοι θα έχουν πρόσβαση στην εφαρμογή μέσω του χρήστη “Χρήστης Εκδοτικών Οίκων” που έχει τα εξής δικαιώματα:

- § Θα κάνει εγγραφή και θα έχει ένα password και ένα username.
- § Θα έχει πρόσβαση μόνο στο δικό του κομμάτι που θα είναι να στέλνει βιβλία προς έγκριση.
- § Θα ενημερώνει για κάθε καινούργια παραλαβή.

3.1.3 ΕΦΑΡΜΟΓΗ ΔΙΑΧΕΙΡΙΣΗΣ

Η Εφαρμογή Διαχείρισης παρέχει υποστηρικτικές υπηρεσίες διαχείρισης και στην Εφαρμογή Χρηστών αλλά και στην Εφαρμογή των Εκδοτικών Οίκων.

Για την εφαρμογή των Εκδοτικών Οίκων παρέχει τον μηχανισμό έγκρισης των περιεχομένων των νέων βιβλίων που έχουν αποσταλεί από τους Εκδοτικούς Οίκους μέσω του χρήστη **“Ελεγκτής Περιεχομένων Βιβλίων”** που έχει τα εξής δικαιώματα:

- § Ελέγχει την καταλληλότητα των βιβλίων που στέλνονται από τον υπεύθυνο των εκδοτικών οίκων και αποφασίζει για την έγκριση τους ή όχι.
- § Διαγράφει τα βιβλία που θεωρεί πως δεν πρέπει να καταχωρηθούν.

Η Εφαρμογή διαχείρισης Για την εφαρμογή των Εκδοτικών Οίκων παρέχει τις απαραίτητες διαδικασίες για την ομαλή λειτουργία της όλων υποσυστημάτων της ψηφιακής βιβλιοθήκης καθώς και τις απαραίτητες διαδικασίες για την ανανέωση του περιεχομένου της μέσω του χρήστη **“Γενικός Διαχειριστής”** που έχει τα εξής δικαιώματα:

- § Εισάγει τα βιβλία που έχει στείλει ο εκδοτικός οίκος κι έχει εγκρίνει ο ελεγκτής περιεχομένου βιβλίων.
- § Είναι υπεύθυνος για την ομαλή λειτουργία της ηλεκτρονικής βιβλιοθήκης.
- § Θα πρέπει να ανανεώνει το site σε τακτά χρονικά διαστήματα

3.2 ΣΧΕΔΙΑΣΜΟΣ ΛΟΓΙΣΜΙΚΟΥ

Ο σχεδιασμός ενός λογισμικού αποτελείται από χαμηλού επιπέδου κώδικες, (ονομασία αλγορίθμων) και την αρχιτεκτονική δομή του λογισμικού. Ο σχεδιασμός αυτός είναι μια διαδικασία με την οποία οι σχεδιαστές έχουν να αντιμετωπίσουν τις απαιτήσεις του ευρέως κοινού και να δημιουργήσουν μια σειρά από κώδικες οι οποίοι θα εφαρμοστούν σε ένα λογισμικό το οποίο θα εξυπηρετεί τις ανάγκες του κάθε χρήστη.

Ο σχεδιασμός μιας εφαρμογής διέπει σε βασικές θεμελιώδεις αρχές οι οποίες εξαπλώνονται ως δομή, εργασίες, αρχιτεκτονική και διαμερισμό

των λειτουργιών αυτής. Στόχος μας είναι η ευχρηστία, ο καταμερισμός ιεραρχίας και η σωστή δομή, από τις διαδικασίες που αφορούν:

- § Δυνατότητα επεξεργασίας δεδομένων
- § Δυνατότητα αφαίρεσης δεδομένων
- § Ευχρηστία και λειτουργικότητα
- § Τη δομή του λογισμικού ως σύνολο και τη γενική κωδικοποίηση των τμημάτων αυτού με σκοπό υψηλής ποιότητας εργασιών και απόδοσης όπως και ακεραιότητα του συστήματος
- § Τον καταμερισμό ιεραρχίας ο οποίος δίνει ή και αφαιρεί εξουσιοδότηση και χωρίζει σε βαθμίδες τους χρήστες του λογισμικού από τον προγραμματιστή του λογισμικού, τον διαχειριστή, μέχρι και τον απλό χρήστη
- § Τον διαμερισμό των λειτουργιών ο οποίος χωρίζει την δομή σε κάθετες και οριζόντιες βαθμίδες, όπου στις κάθετες καθορίζουμε ξεχωριστές ενέργειες τις εφαρμογής μας, οι οποίες λειτουργούν ανεξάρτητα. Οι οριζόντιες βαθμίδες είναι ως παρακλάδια μιας συγκεκριμένης ενέργειας και καθορίζουν εναλλακτικές λειτουργίες που προσφέρει η κάθε βαθμίδα.
- § Τη βάση δεδομένων, στην οποία μπορούμε να αποθηκεύσουμε δεδομένα που αφορούν την εφαρμογή που λειτουργούμε και μπορούμε μέσω αυτής να τα ανακτήσουμε και να τα επεξεργαστούμε μέσω της εφαρμογής αυτής χρησιμοποιώντας κώδικες.
- § Τη διαδικασία εφαρμογής λογισμικού όπου έχουμε τη δυνατότητα να τυπώσουμε, να επεξεργαστούμε και να καταχωρήσουμε την κάθε ενότητα της εφαρμογής μας ξεχωριστά.

§ Τη δυνατότητα απόκρυψης δεδομένων με την οποία δίδουμε ξεχωριστά δικαιώματα στην ιεραρχία των χρηστών. Υπάρχουν οι διαχειριστές, οι οποίοι έχουν τον απεριόριστο έλεγχο στην εφαρμογή και χωρίζονται σε γενικό διαχειριστή και ελεγκτή περιεχομένου βιβλίων, τους εκδοτικούς οίκους οι οποίοι έχουν μόνο τη δυνατότητα να ανεβάσουν δεδομένα στην βάση δεδομένων αλλά όχι στην εφαρμογή όπου και θα ελέγχονται από τον ελεγκτή περιεχομένου βιβλίων, και τους απλούς χρήστες (χρήστες με απεριόριστη πρόσβαση, χρήστες με περιορισμένη πρόσβαση και δοκιμαστικούς χρήστες), όπου σε αυτούς δίδουμε τη δυνατότητα επεξεργασίας περιορισμένου υλικού το οποίο έχει σχέση μόνο με λειτουργίες κατεβάσματος δεδομένων.

Βασική πτυχή στον σχεδιασμό ενός λογισμικού είναι η μελέτη σχεδίου. Στόχος του είναι η τελική επιτυχία του λογισμικού. Η μελέτη αυτή είναι ένα προσχέδιο του λογισμικού που έχουμε να φέρουμε εις πέρας και σε αυτή θα επικεντρωθούμε σε ότι αφορά την δομή της εφαρμογής μας. Τα βασικότερα σημεία που έχουμε να αντιμετωπίσουμε είναι τα εξής:

- § Να είναι συμβατό με άλλα λογισμικά που αφορούν τις ίδιες ή ίδιου τύπου λειτουργίες που έχουν σχέση με το λογισμικό μας
- § Να μπορούμε να αναβαθμίζουμε το λογισμικό μας με νέες λειτουργίες, χωρίς να είναι απαραίτητο να επεμβαίνουμε στη βασική δομή και να αλλάζουμε την αρχιτεκτονική του αρχικού μας σχεδίου
- § Να παρέχουμε ανθεκτικότητα, όσον αφορά πιθανά σφάλματα που μπορεί να θέσουν το λογισμικό μας εκτός λειτουργίας

- § Να δημιουργήσουμε τη δυνατότητα του να μπορούμε να επαναφέρουμε το λογισμικό μας σε μια προϋπάρχουσα κατάσταση για κάποιο χρονικό διάστημα
- § Οφείλουμε από πριν να συνθέσουμε ένα ολοκληρωμένο προϊόν, το οποίο θα θέσουμε υπό δοκιμή ως προς τη λειτουργικότητα και την ευχρηστία του
- § Το λογισμικό μας πρέπει να είναι αξιόπιστο ως προς τις λειτουργίες που παρέχει
- § Σημεία της δομής του λογισμικού μας να είναι δυνατόν να επαναχρησιμοποιηθούν σε άλλες εφαρμογές, όπου μπορεί να μας φανούν χρήσιμες
- § Να είναι σε θέση το λογισμικό να λειτουργεί υπό πίεση χαμηλής μνήμης και να είναι ανθεκτικό σε εισαγωγή λάθος δεδομένων
- § Μια από τις πιο σημαντικές δυσχέρειες που έχουμε να αντιμετωπίσουμε είναι η παροχή ασφάλειας στο λογισμικό, ως προς τις κακόβουλες ενέργειες που μπορεί να τύχουν. Πρέπει να δημιουργήσουμε ανάλογες δικλίδες ασφαλείας, ώστε να προφυλάξουμε τα δεδομένα μας ηλεκτρονικής μορφής και μη, όπως και τους χρήστες που λειτουργούν την εφαρμογή μας

Στην τελειοποιημένη δομή θα θέσουμε παραμέτρους, με τις οποίες θα λειτουργεί η εφαρμογή με την μεγαλύτερη δυνατή χρησιμότητα, για την πλειοψηφία των χρηστών που έχουμε ως στόχο να προσκαλέσουμε.

3.3 ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ

Η τρίτη φάση του κύκλου ανάπτυξης λογισμικού είναι η ίδια η ανάπτυξη του λογισμικού και αυτή είναι η πιο σύντομη φάση όλου του κύκλου. Σε αυτή τη φάση υπάρχουν οι εντολές για τον σχηματισμό του κύριου μέρους του λογισμικού. Κατά την διάρκεια της ανάπτυξης και με

τις απαιτήσεις που ετοιμάσαμε σε προηγούμενη φάση σχεδιάζουμε την κωδικοποίηση του λογισμικού μας. Με τα διάφορα στοιχεία που είχαμε στην φάση του σχεδιασμού ετοιμάζουμε τα διάφορα κομμάτια του λογισμικού ώστε στο τέλος να τα ενώσουμε και να έχουμε ένα λογισμικό χωρίς σφάλματα.

3.4 ΔΟΚΙΜΗ ΛΟΓΙΣΜΙΚΟΥ

Σε αυτή τη φάση της δοκιμής θα πρέπει να ελέγξουμε το λογισμικό μας για να διαπιστώσουμε αν εμφανίζει τα αποτελέσματα που πρέπει. Οι δοκιμές θα πρέπει να απαντούν στο ερώτημα «Το λογισμικό θα εμφανίζει τα ζητούμενα αποτελέσματα κάτω από τις απαιτούμενες συνθήκες;»

Η διάρκεια του χρόνου που χρειαζόμαστε για αυτό το ερώτημα συνήθως είναι περισσότερος από όσο υποθέτουμε. Οι δοκιμές είναι οι πιο χρονοβόρες διαδικασίες εφόσον θα πρέπει να ελέγξουμε τα αποτελέσματα που εμφανίζει διορθώνοντας το λογισμικό και υπάρχουν και ορισμένα κομμάτια που θα πρέπει να σχεδιαστούν από την αρχή.

Υπάρχουν τρεις ενέργειες που θα πρέπει να εξετάσουμε για την δοκιμή του λογισμικού:

- § Δοκιμή μονάδας: σε αυτή την ενέργεια ελέγχουμε κάθε κομμάτι του λογισμικού ξεχωριστά για να δούμε αν υπάρχουν σφάλματα και αν εντοπιστούν να διορθωθούν.
- § Δοκιμή συστήματος: ελέγχουμε το λογισμικό ως σύνολο και ελέγχουμε και ένα ένα το κάθε κομμάτι της εφαρμογής για να διαπιστώσουμε αν υπάρχουν τυχόν σφάλματα ή όχι. Εξετάζουμε ακόμη το χρόνο εμφάνισης των αποτελεσμάτων και την χωρητικότητα που έχουμε για αποθήκευση των αρχείων μας.
- § Δοκιμή αποδοχής: είναι όταν το σύστημα είναι έτοιμο για την χρησιμοποίηση του χωρίς κανένα πρόβλημα. Ελέγχεται και αν

μείνουν όλοι ικανοποιημένοι τότε η εφαρμογή είναι έτοιμη για χρήση.

3.5 ΛΕΙΤΟΥΡΓΙΑ ΚΑΙ ΣΥΝΤΗΡΗΣΗ

Σε αυτό το στάδιο η εφαρμογή ελέγχεται από τους τεχνικούς και τους τελικούς χρήστες ώστε να δουν αν λειτουργεί σωστά όπως το ζήτησαν ή αν χρειάζεται κάποιες αλλαγές. Οι διάφορες αλλαγές στο λογισμικό, στο υλικό ή στις νέες απαιτήσεις που θα εμφανιστούν ονομάζεται συντήρηση.

Έγιναν μελέτες που εξέτασαν το χρόνο που απαιτείται για τις εργασίες συντήρησης (Lientz και Swanson, 1980). Το 20% περίπου σε προβλήματα παραγωγής, ένα άλλο 20% σε αλλαγές υλικού, αναφορών, αρχείων ή λογισμικού της εφαρμογής. Το 60% αφορά διάφορες βελτιώσεις που ζητάνε οι χρήστες και αλλαγή για καλύτερη απόδοση επεξεργασίας. Μπορεί να μειωθεί σημαντικά ο χρόνος για την επίλυση των προβλημάτων με καλύτερο σχεδιασμό και καλύτερες πρακτικές ανάλυσης.

Οι εφαρμογές διαφέρουν η μια με την άλλη ως προς τα προβλήματα που έχουν να λύσουν και ως προς την τεχνολογία τους και για αυτό έχουν δημιουργηθεί διαφορές μέθοδοι ανάπτυξης τους.

4 ΚΑΤΑΓΡΑΦΗ ΑΠΑΙΤΗΣΕΩΝ ΕΦΑΡΜΟΓΗΣ

4.1 ΟΡΙΣΜΟΣ

Μια λειτουργία που θα πρέπει το λογισμικό να επιτελεί ή μια συνθήκη που θα πρέπει να ικανοποιεί όταν θα έχει ολοκληρωθεί η κατασκευή του αφορούν τη συμπεριφορά του λογισμικού προς το εξωτερικό του περιβάλλον (χρήστης, άλλες εφαρμογές, λογισμικού) και όχι εσωτερικά του στοιχεία.

§ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ: περιγράφουν τις εργασίες (λειτουργίες) που θα πρέπει να εκτελεί το λογισμικό, καθορίζουν τη συμπεριφορά του συστήματος, δηλ. την απόκριση που πρέπει να εμφανίζει στο περιβάλλον του όταν ισχύουν συγκεκριμένες συνθήκες.

§ ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ: περιγράφουν χαρακτηριστικά του λογισμικού τα οποία δεν αφορούν την εκτέλεση κάποιας λειτουργίας από αυτό. Όπως για παράδειγμα ο καθορισμός ιδιωμάτων εμφάνισης (αισθητική, επικοινωνία με το χρήστη), επιδόσεων (αξιοπιστία, χρόνος εκτέλεσης, χρήση πόρων), υλοποίησης, κ.τ.λ.

4.2 ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

Οι λειτουργικές απαιτήσεις αφορούν στην ανάπτυξη μίας εφαρμογής η οποία θα ικανοποιεί τις ακόλουθες απαιτήσεις κατά την διαδικασία επιλογής και “download” του βιβλίου.

4.2.1 ΔΙΑΔΙΚΑΣΙΑ ΕΠΙΛΟΓΗΣ ΚΑΙ ΑΠΟΚΤΗΣΗΣ ΤΟΥ ΒΙΒΛΙΟΥ

Περιλαμβάνει τρία βήματα:

1. ΕΙΣΑΓΩΓΗ ΕΠΙΘΥΜΙΑΣ

Ο χρήστης-μέλος θέτει την επιθυμία του για ένα βιβλίο. Η

επιθυμία του πελάτη αφορά την επιλογή ενός βιβλίου το οποίο ανήκει σε έναν συγκεκριμένο συγγραφέα και μια συγκεκριμένη κατηγορία (λογοτεχνία, πληροφορική κ.α.)

2. ΠΡΟΤΑΣΗ ΑΠΟΚΤΗΣΗΣ ΤΟΥ ΒΙΒΛΙΟΥ

Το σύστημα αποκρίνεται στην επιθυμία του χρήστη-μέλους, οπότε υπάρχουν οι εξής περιπτώσεις:

- είναι δυνατόν να ικανοποιηθεί το αίτημα του χρήστη-μέλους όπως είναι, άρα υπάρχει το συγκεκριμένο βιβλίο που επιθυμεί, το οποίο στη συνέχεια μπορεί να επιλεγεί,
- δεν ικανοποιείται το αίτημα του χρήστη-μέλους (π.χ. δεν υπάρχει το συγκεκριμένο βιβλίο που να ανταποκρίνεται στα κριτήρια που έχει θέσει) και δεν υπάρχει δυνατότητα “download” του συγκεκριμένου βιβλίου.

3. ΑΠΟΚΤΗΣΗ ΤΟΥ ΒΙΒΛΙΟΥ

Στην περίπτωση που ο χρήστης επιλέγει ένα βιβλίο που ανήκει σε μια συγκεκριμένη κατηγορία και σε έναν συγκεκριμένο συγγραφέα ολοκληρώνεται η απόκτηση του κατεβάζοντας το ψηφιακό περιεχόμενο του βιβλίου.

4.3 ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

4.3.1 ΑΠΑΙΤΗΣΕΙΣ ΧΡΗΣΗΣ

Το λογισμικό θα πρέπει να περιέχει φιλικό περιβάλλον προς το χρήστη με υποστήριξη γραφικών και συνδυασμό χρήσης ποντικιού και πληκτρολογίου. Θα υπάρχουν φόρμες καταχώρησης στοιχείων με αυτοματοποίηση πεδίων όπου αυτό επιτρέπεται.

4.3.2 ΑΠΑΙΤΗΣΕΙΣ ΑΞΙΟΠΙΣΤΙΑΣ

Η πρόσβαση από πολλούς χρήστες ταυτόχρονα προστατεύεται με δυνατότητα να χρησιμοποιούν την εφαρμογή με δικό τους κωδικό χρήστη

και password. Το λογισμικό θα υποστηρίζει «κλείδωμα» των εγγραφών κάθε χρήστη ώστε να μην υπάρχουν προβλήματα λάθους κατά τη διάρκεια ενημέρωσης από άλλο χρήστη.

4.3.3 ΑΠΑΙΤΗΣΕΙΣ ΕΠΙΔΟΣΕΩΝ

Θα υπάρχει αξιοποίηση των πόρων του συστήματος, όποιων κι αν είναι αυτοί. Επιπλέον, η απόκριση από το σύστημα βάσεων δεδομένων δεν θα ξεπερνά τα 2 δευτερόλεπτα.

4.3.4 ΑΠΑΙΤΗΣΕΙΣ ΥΠΟΣΤΗΡΙΞΗΣ

Το λογισμικό θα διατίθεται σε περιβάλλον Windows και θα εγκαθίσταται ιδιαίτερα εύκολα σε οποιοδήποτε μηχάνημα με υποστήριξη αρχείου README το οποίο θα επεξηγεί και θα διευκρινίζει τα θέματα εγκατάστασης και λειτουργίας της εφαρμογής.

4.3.5 ΑΠΑΙΤΗΣΕΙΣ ΥΛΟΠΟΙΗΣΗΣ

Θα χρησιμοποιηθεί η γλώσσα PHP λόγω σε συνδυασμό με χρήση SQL για υποβολή ερωτημάτων προς τη Βάση Δεδομένων.

4.3.6 ΑΠΑΙΤΗΣΕΙΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Θα χρησιμοποιηθεί σύστημα Βάσεων Δεδομένων με τους ακόλουθους βασικούς πίνακες: Πίνακα users (με τα στοιχεία των μελών), Πίνακας βιβλίων (με τα στοιχεία του κάθε βιβλίου).

4.3.7 ΦΥΣΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

Το λογισμικό αρχικά θα εγκατασταθεί σε έναν server με ελάχιστη απαίτηση Pentium IV, 128 MB RAM και θα έχει την δυνατότητα εγκατάστασης σε πολλά τερματικά PCs (Pentium II) με λειτουργικό Windows NT στα οποία θα έχει εγκατασταθεί MySQL και Php.

5 ΜΟΝΤΕΛΟΠΟΙΗΣΗ

5.1.1 ΕΙΣΑΓΩΓΙΚΑ

Η ανάπτυξη ενός μοντέλου για ένα σύστημα λογισμικού που θα μπορέσει να χρησιμοποιηθεί στη βιομηχανία, προτού δομηθεί ή επανασχεδιαστεί το σύστημα, είναι τόσο απαραίτητο όσο το να υπάρχουν τα αρχιτεκτονικά σχέδια για ένα μεγάλο κτίσμα. Τα καλά μοντέλα είναι απαραίτητα για την επικοινωνία ανάμεσα στις ομάδες ενός έργου και επίσης για να υπάρχει μία ορθή αρχιτεκτονική του συστήματος. Δημιουργούμε μοντέλα των σύνθετων συστημάτων γιατί δεν μπορούμε να τα κατανοήσουμε στην ολότητά τους. Όσο αυξάνει η πολυπλοκότητα των συστημάτων, αυξάνει και η σημασία της ύπαρξης καλών τεχνικών μοντελοποίησης. Υπάρχουν και άλλοι παράγοντες για την επιτυχία ενός έργου, αλλά η ύπαρξη μίας πρότυπης, πλούσιας γλώσσας μοντελοποίησης είναι απαραίτητος παράγοντας. Μία γλώσσα μοντελοποίησης πρέπει να περιλαμβάνει:

- § Στοιχεία μοντέλων – στοιχειώδεις έννοιες μοντελοποίησης και τη σημασιολογία τους.
- § Συμβολισμό – οπτικές αναπαραστάσεις των στοιχείων μοντελοποίησης.
- § Οδηγίες – τρόπους χρήσης στην πράξη.

Στην περίπτωση των όλο και πιο πολύπλοκων συστημάτων η οπτική αναπαράσταση και η μοντελοποίηση γίνονται απαραίτητες.

Στη εφαρμογή μας ο χρήστης πρέπει να ακολουθήσει μια συγκεκριμένη σειρά βημάτων μέχρι να φτάσει στο αποτέλεσμα που επιθυμεί. Υπάρχουν διάφορα είδη διαγραμμάτων που αναπαριστούν οπτικά την σειρά των βημάτων που ακολουθεί ο χρήστης. Αρχικά, δημιουργήσαμε ένα γενικό διάγραμμα χρήσης (use case) το οποίο απεικονίζει τις τρεις εφαρμογές από τις οποίες αποτελείται ουσιαστικά η εφαρμογή μας. Στη συνέχεια,

δημιουργήσαμε διαγράμματα χρήσης (use case) με τις ορατές λειτουργίες του συστήματος. Αυτό ανήκει στην φάση της προκαταρκτικής σχεδίασης στο μοντέλο καταρράκτη. Εφόσον παρουσιάσουμε τις βασικές λειτουργίες του συστήματος μας, θα πρέπει να προχωρήσουμε σε μια πιο λεπτομερή σχεδίαση δηλαδή πως τα διάφορα αντικείμενα αλληλεπιδρούν μεταξύ τους, καθώς και τα βήματα που πρέπει να ακολουθήσει ο χρήστης αναλυτικότερα. Σε αυτό θα μας βοηθήσουν τα διαγράμματα ακολουθίας (sequence).

5.1.2 ΔΙΑΓΡΑΜΜΑ ΧΡΗΣΗΣ (USE CASE)

Διάγραμμα που αποτελείται από περιπτώσεις χρήσης, δράστες, και σχέσεις συνάφειας (relationships) μεταξύ των δραστών και των υπηρεσιών του συστήματος .

Για τα Διαγράμματα Χρήσης επιλέγουμε μόνο τις υπηρεσίες που είναι «ορατές» από ένα εξωτερικό σύστημα (δράστη)

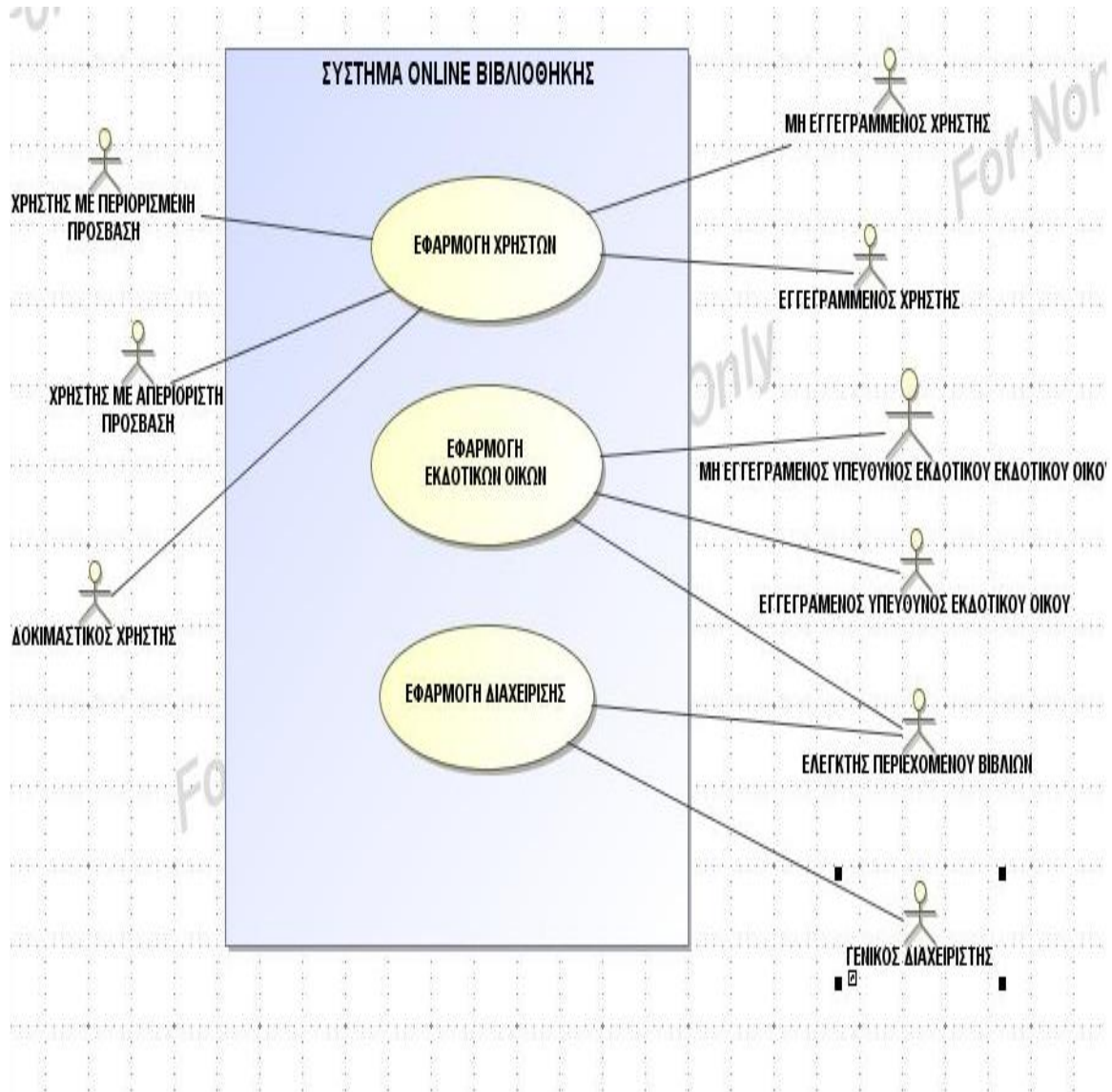
Τα Διαγράμματα Χρήσης μπορούν να εφαρμοσθούν με δύο τρόπους

Για να μοντελοποιήσουν το πλαίσιο (context) του συστήματος

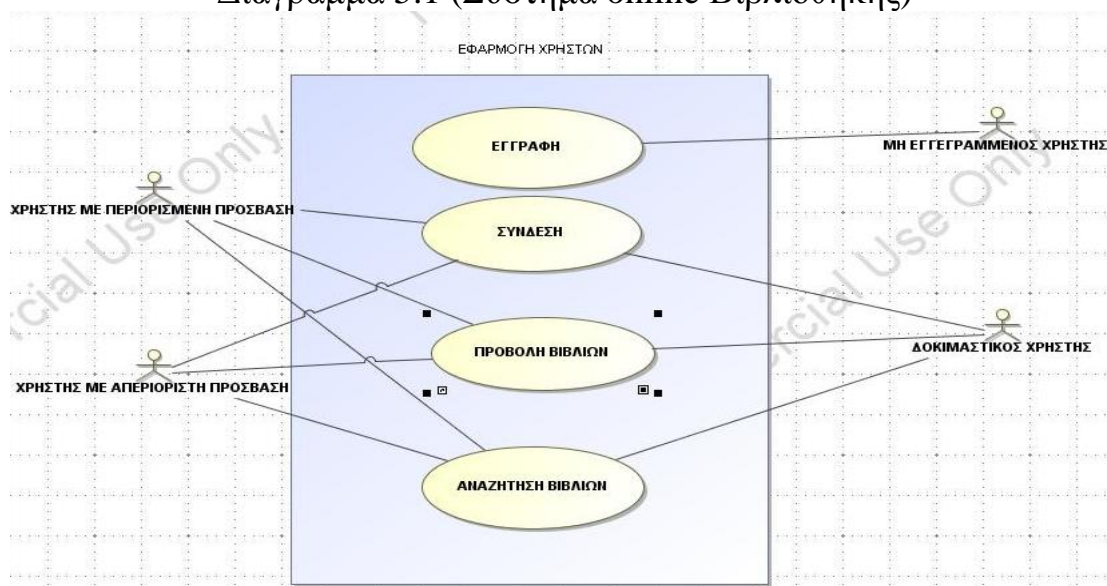
§ Ποιοι χρήστες αλληλεπιδρούν με ποια σημεία του συστήματος.

Μοντελοποίηση των απαιτήσεων του συστήματος.

§ Ποιες είναι οι λειτουργίες του συστήματος

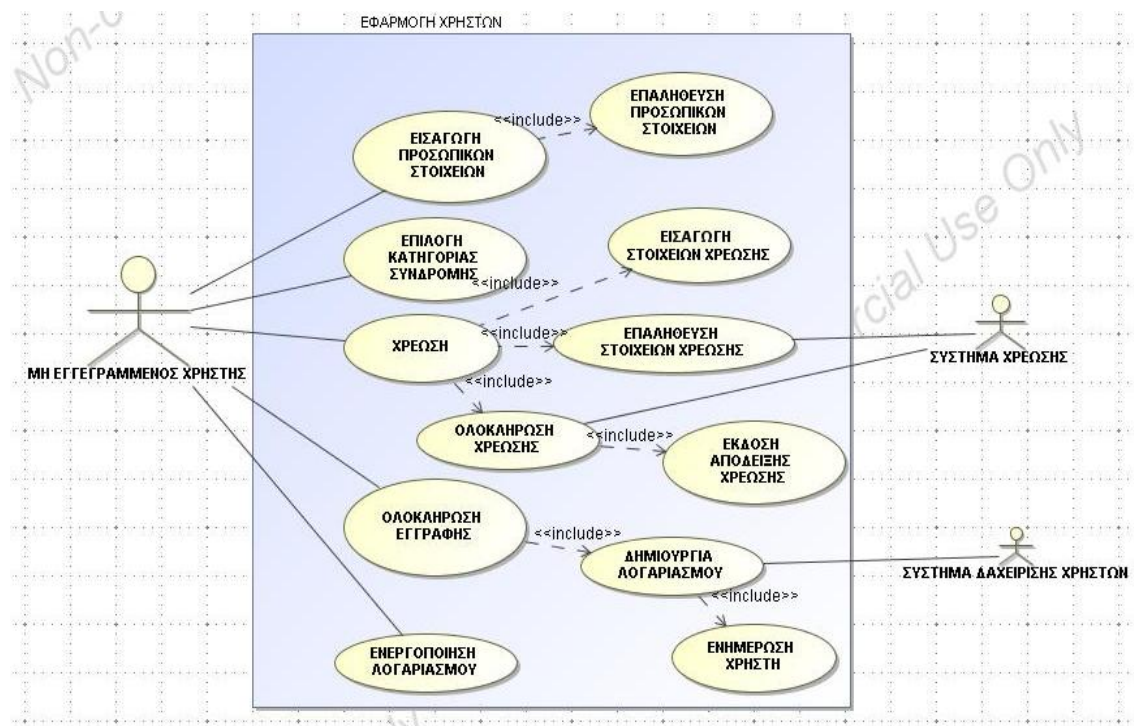


Διάγραμμα 5.1 (Σύστημα online Βιβλιοθήκης)



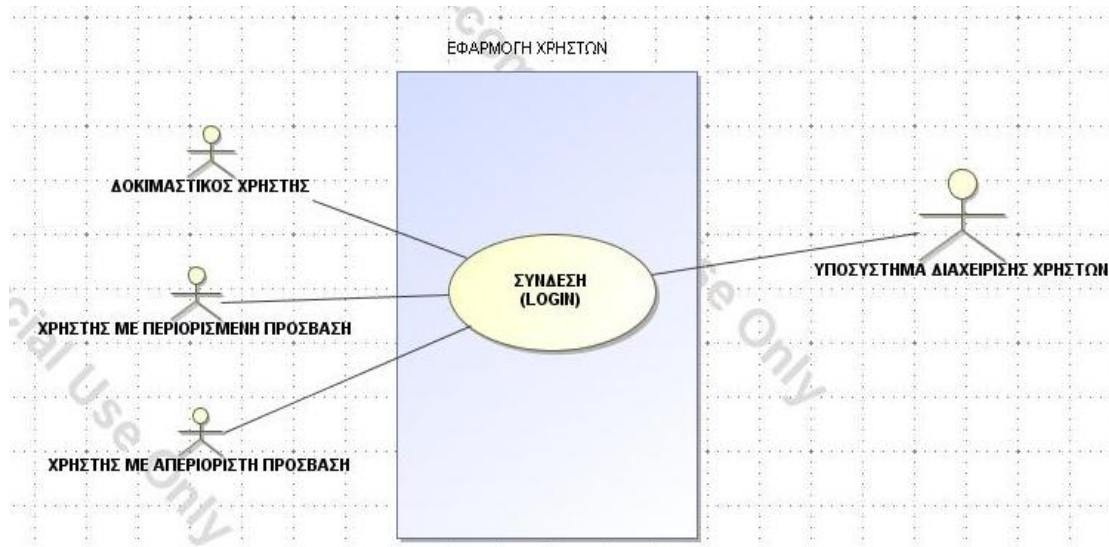
Διάγραμμα 5.2 (Εφαρμογή χρηστών)

Ο μη εγγεγραμμένος χρήστης θα πρέπει να κάνει εγγραφή στην εφαρμογή για να εισέλθει σε αυτήν. Ο δοκιμαστικός χρήστης, ο χρήστης με περιορισμένη πρόσβαση και ο χρήστης με απεριόριστη πρόσβαση θα μπορούν να κάνουν σύνδεση στην εφαρμογή και να δουν τα ήδη υπάρχοντα βιβλία καθώς και να κάνουν αναζήτηση για το βιβλίο που χρειάζονται.



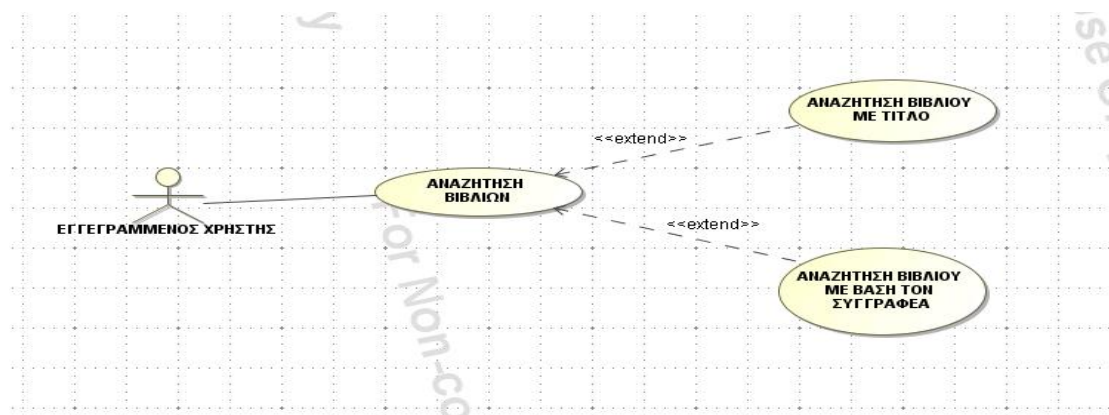
Διάγραμμα 5.3 (Λειτουργία εγγραφής χρηστών)

Ο μη εγγεγραμμένος χρήστης θα πρέπει να επιλέξει ποια κατηγορία συνδρομής επιθυμεί και να εισάγει τα προσωπικά του στοιχεία και στην συνέχεια θα επαληθευτούν. Ο χρήστης θα χρεωθεί με την αξία της συνδρομής και το σύστημα χρέωσης θα επαληθεύει αν τα στοιχεία που έδωσε είναι ορθά. Μετά την ολοκλήρωση της χρέωσης θα λαμβάνει ένα μήνυμα ότι η χρέωση έγινε κανονικά. Με την ολοκλήρωση της εγγραφής θα δημιουργείται ένας λογαριασμός από το σύστημα διαχείρισης χρηστών με τα στοιχεία που έδωσε ο χρήστης και θα ενεργοποιείται αυτόματα.



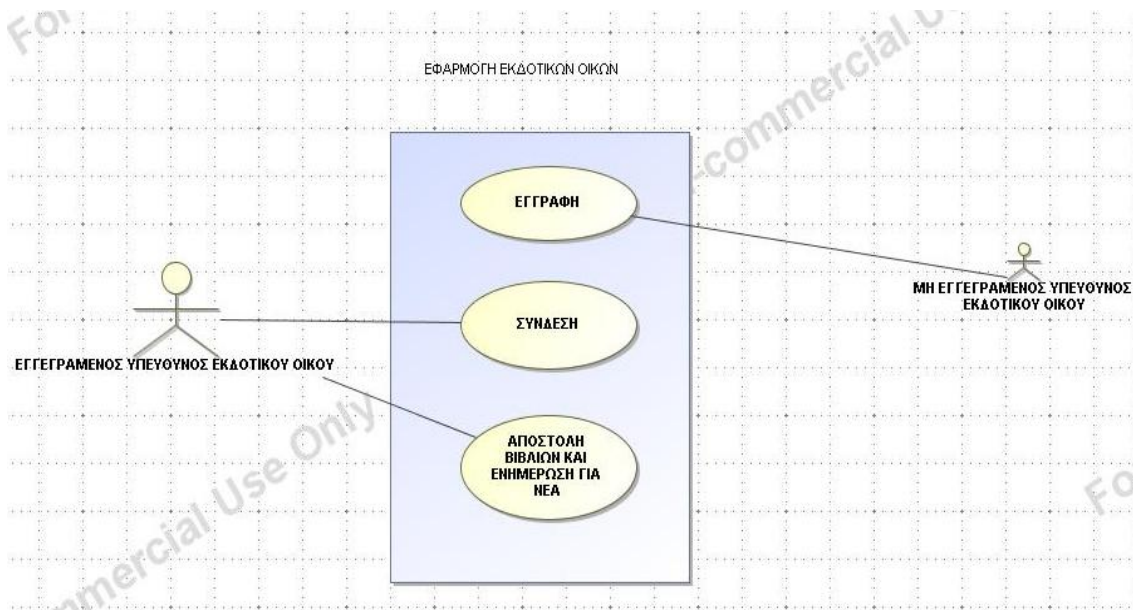
Διάγραμμα 5.4 (Λειτουργία σύνδεσης)

Ο δοκιμαστικός χρήστης, ο χρήστης με περιορισμένη πρόσβαση και ο χρήστης με απεριόριστη πρόσβαση θα συνδέονται με την εφαρμογή εφόσον έχουν προηγουμένως πραγματοποιήσει εγγραφή, αυτό θα το ελέγξει το υποσύστημα διαχείρισης χρηστών.



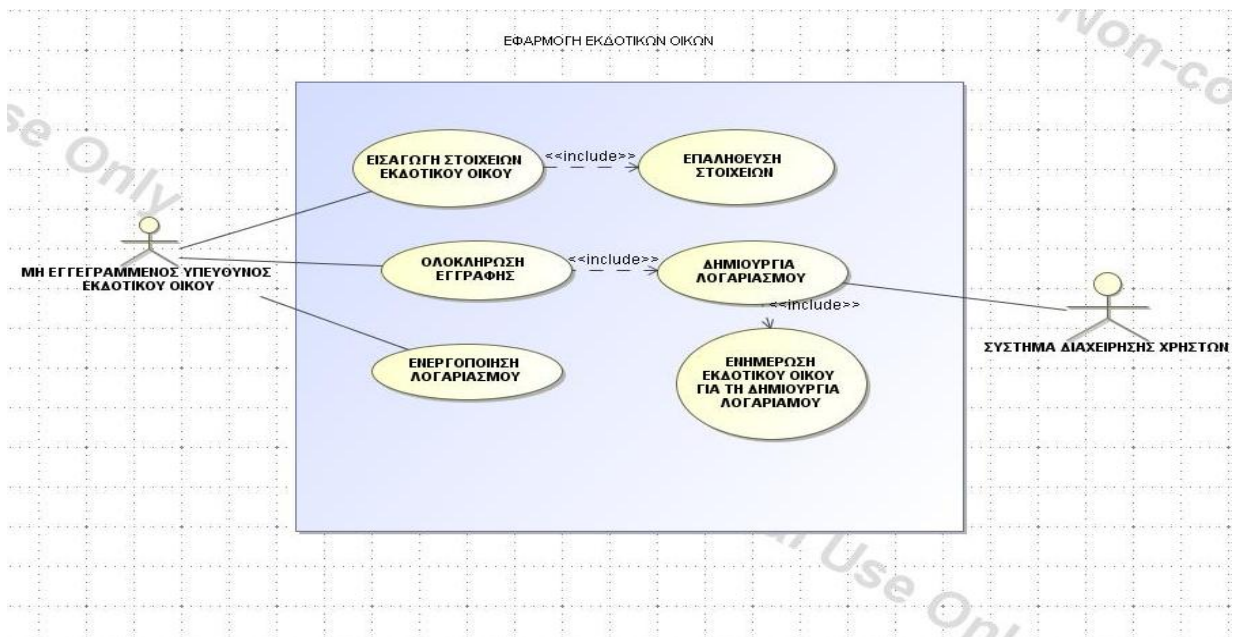
Διάγραμμα 5.5 (Λειτουργία αναζήτησης χρηστών)

Σε αυτό το διάγραμμα βλέπουμε τους εγγεγραμμένους χρήστες να μπορούν κάνουν αναζήτηση με δυο διαφορετικούς τρόπους (αναζήτηση με βάση τον τίτλο του βιβλίου και αναζήτηση με βάση τον συγγραφέα του βιβλίου)



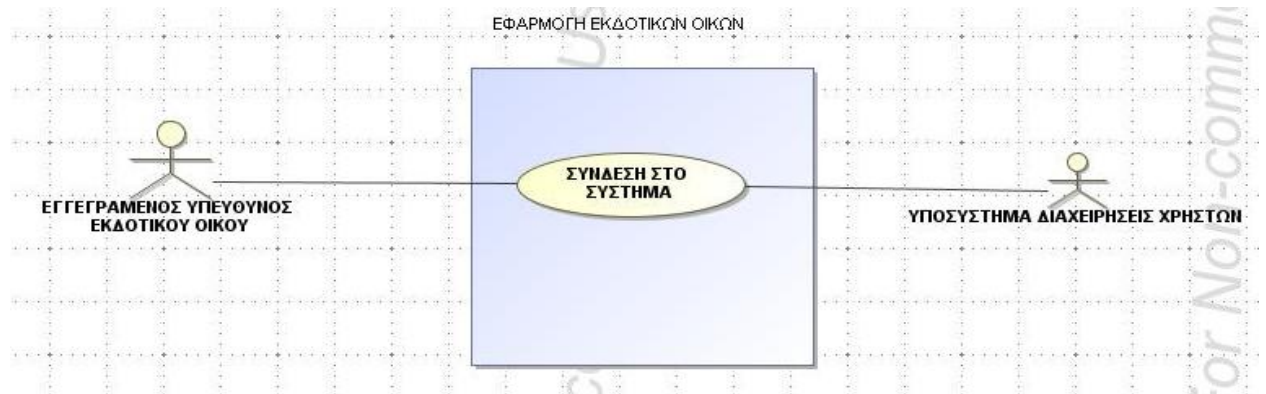
Διάγραμμα 5.6 (Εφαρμογής εκδοτικών οίκων)

Ο μη εγγεγραμμένος υπεύθυνος εκδοτικού οίκου θα πρέπει αναγκαστικά να κάνει εγγραφή στην εφαρμογή. Ο εγγεγραμμένος υπεύθυνος εκδοτικού οίκου θα κάνει σύνδεση στην εφαρμογή και θα μπορεί να στέλνει βιβλία προς έγκριση καθώς και να ενημερώνει την εφαρμογή με καινούρια.



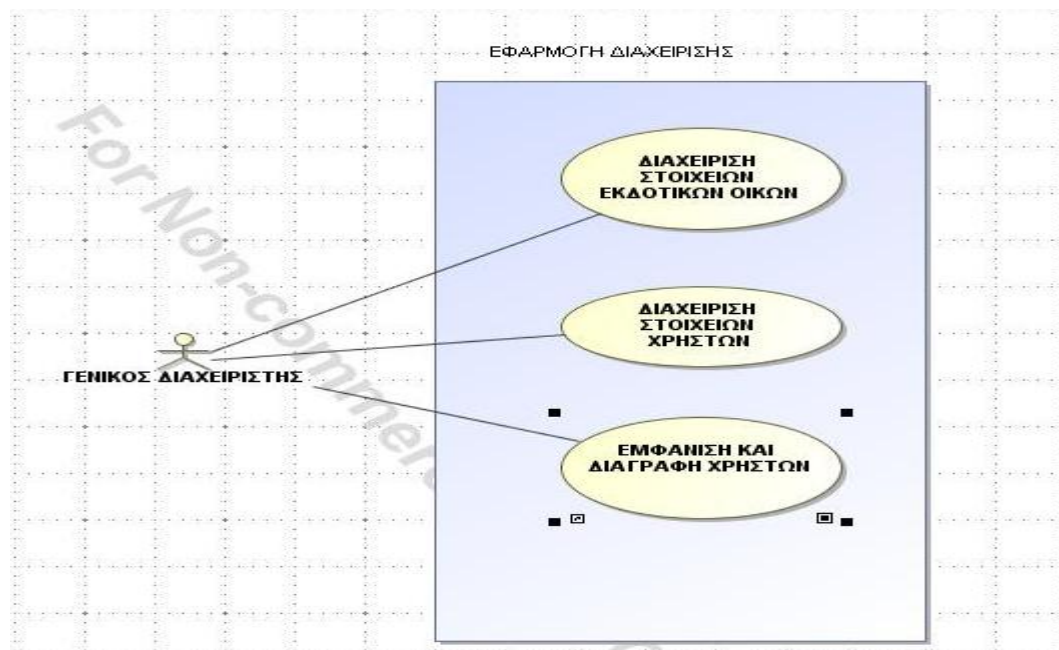
Διάγραμμα 5.7 (Λειτουργία εγγραφής εκδοτικών οίκων)

Ο μη εγγεγραμμένος υπεύθυνος εκδοτικού οίκου θα μπορεί να εισάγει τα στοιχεία του και να γίνεται η επαλήθευση αυτών. Στην συνέχεια θα ολοκληρώνεται η εγγραφή και θα δημιουργείται ο λογαριασμός του από το σύστημα διαχείρισης χρηστών που θα τον ενημερώνει αμέσως εάν η εγγραφή του είναι επιτυχής. Ο λογαριασμός του εκδοτικού οίκου θα ενεργοποιείται αυτόματα.



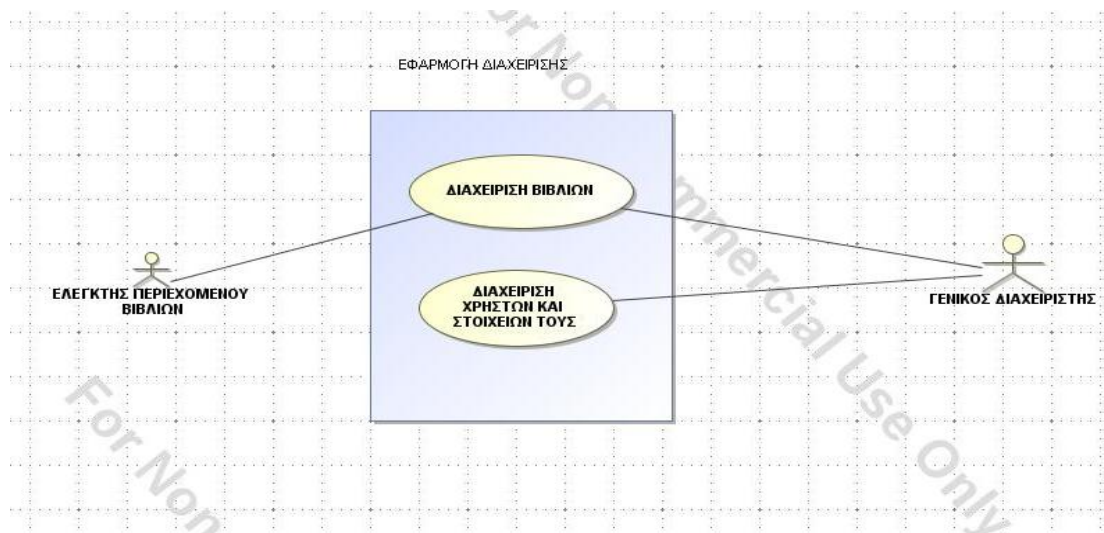
Διάγραμμα 5.8 (λειτουργία σύνδεσης εκδοτικών οίκων)

Ο εγγεγραμμένος υπεύθυνος εκδοτικού οίκου θα μπορεί να κάνει σύνδεση στην εφαρμογή εφόσον έχει κάνει πρώτα εγγραφή, αυτό θα το ελέγξει από το υποσύστημα διαχείρισης χρηστών.



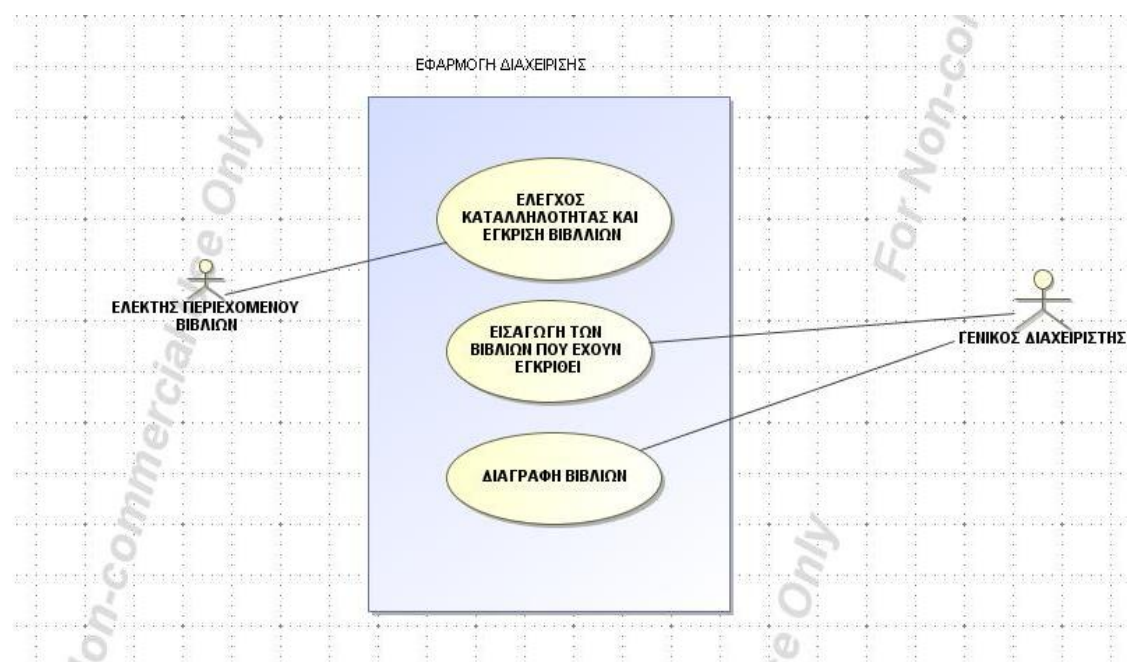
Διάγραμμα 5.9 (Εφαρμογή διαχείρισης)

Ο γενικός διαχειριστής έχει την δυνατότητα να επεξεργάζεται τα στοιχεία των εκδοτικών οίκων και τα στοιχεία των χρηστών. Επίσης μπορεί να δει τους εγγεγραμμένους χρήστες της εφαρμογής και να διαγράψει κάποιον εφόσον το κρίνει απαραίτητο.



Διάγραμμα 5.10 (Εφαρμογή διαχείρισης)

Εδώ βλέπουμε τους διαχειριστές της εφαρμογής (ελεγκτή περιεχομένου βιβλίων και γενικό διαχειριστή). Ο ελεγκτής περιεχομένου βιβλίων μπορεί να διαχειρίζεται τα βιβλία σε συνεργασία με τον γενικό διαχειριστή. Ο γενικός διαχειριστής επίσης διαχειρίζεται τους χρήστες και τα στοιχεία των χρηστών.



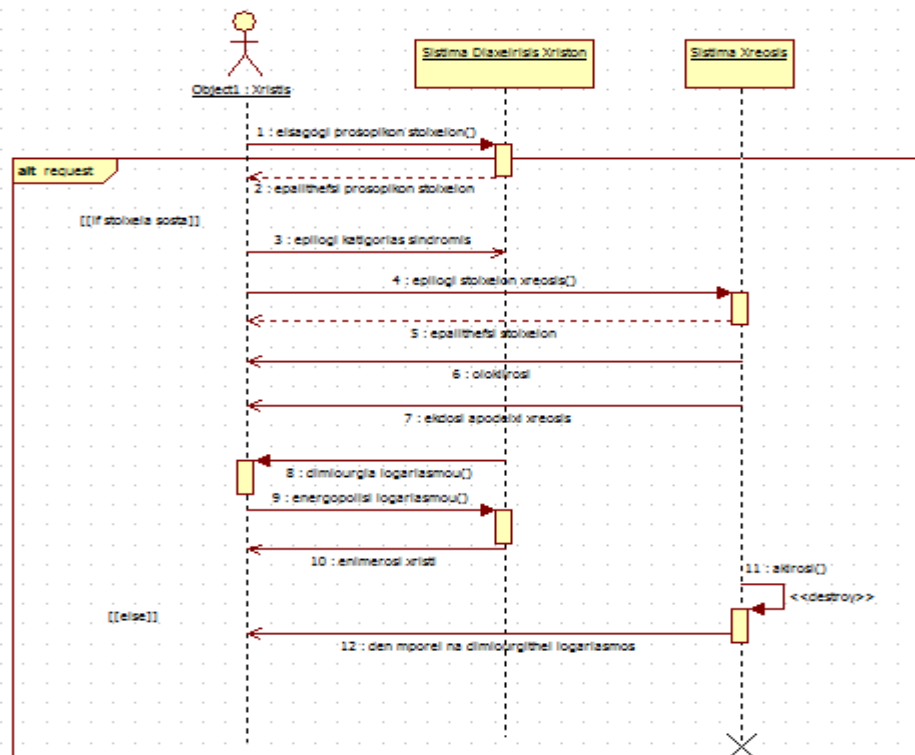
Διάγραμμα 5.11 (Εφαρμογής διαχείρισης)

Ο ελεγκτής περιεχομένου βιβλίων θα ελέγχει την καταλληλότητα των βιβλίων που έχουν σταλεί από τον υπεύθυνο του εκδοτικού οίκου. Στην συνέχεια, ο γενικός διαχειριστής θα προσθέσει τα βιβλία που έχει εγκρίνει προηγουμένως ο ελεγκτής. Ακόμα μπορεί να διαγράψει τα βιβλία από την εφαρμογή.

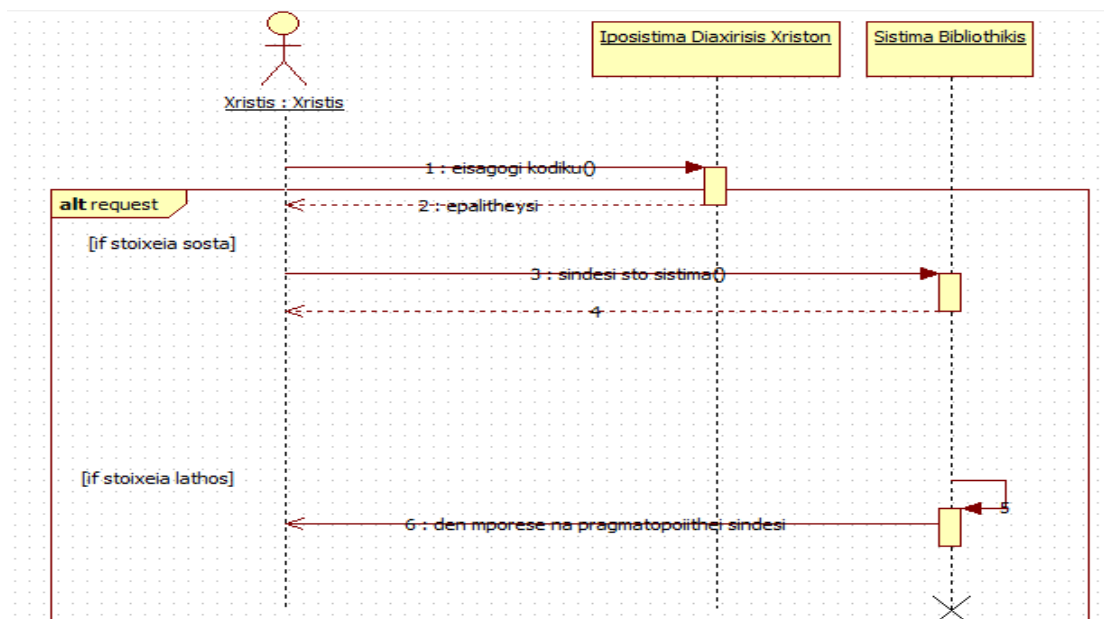
5.1.3 ΔΙΑΓΡΑΜΜΑΤΑ ΑΚΟΛΟΥΘΙΑΣ (SEQUENCE)

Περιγράφουν πώς τα αντικείμενα αλληλεπιδρούν μεταξύ τους, εστιάζοντας περισσότερο σε θέματα χρόνου.

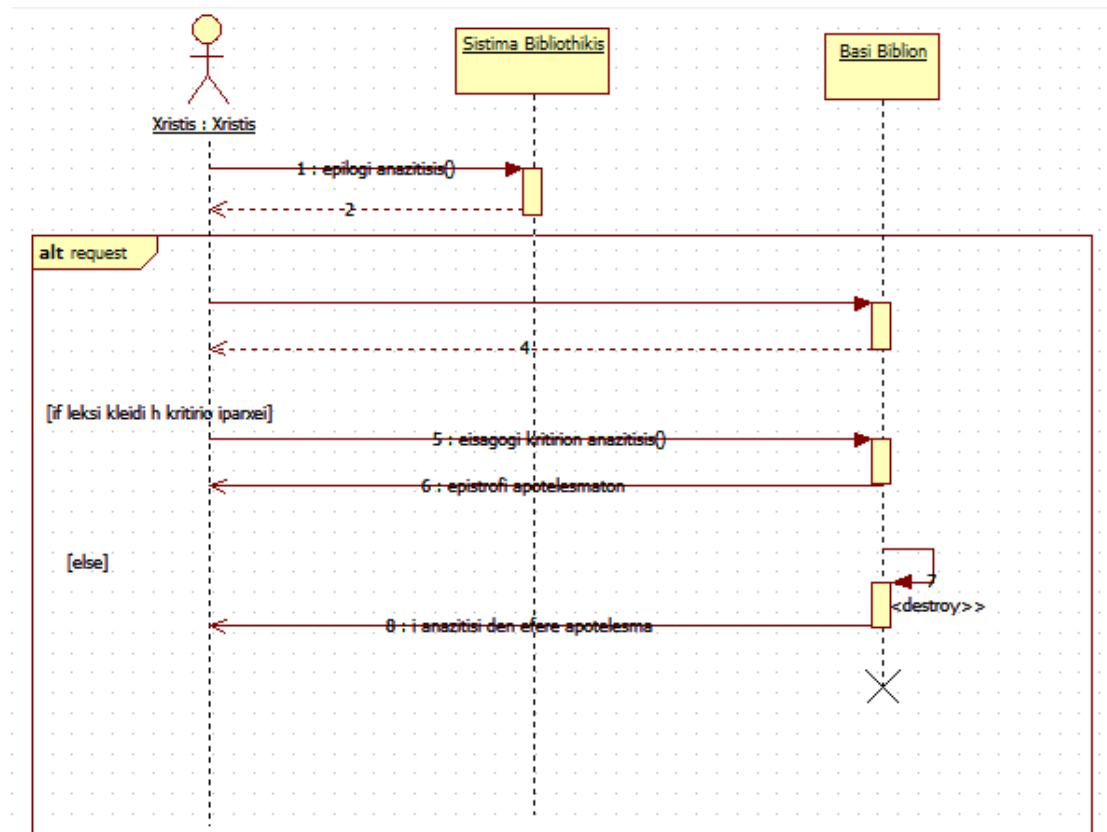
Στα **διαγράμματα ακολουθίας (sequence)** τα αντικείμενα παριστάνονται με κουτιά ή/και ρόλους τα οποία παρατάσσονται στο χώρο (συνήθως από αριστερά προς τα δεξιά). Τα αντικείμενα ανταλλάσσουν μηνύματα τα οποία παριστάνονται με κατευθυνόμενα βέλη των οποίων η φορά δείχνει την κατεύθυνση του μηνύματος (ποιο αντικείμενο κάνει την κλήση και σε ποιο αντικείμενο γίνεται η κλήση). Ο χρόνος παριστάνεται από πάνω προς τα κάτω στο διάγραμμα. Δηλαδή τα μηνύματα που βρίσκονται πιο πάνω στο διάγραμμα προηγούνται χρονικά αυτών που βρίσκονται πιο χαμηλά.



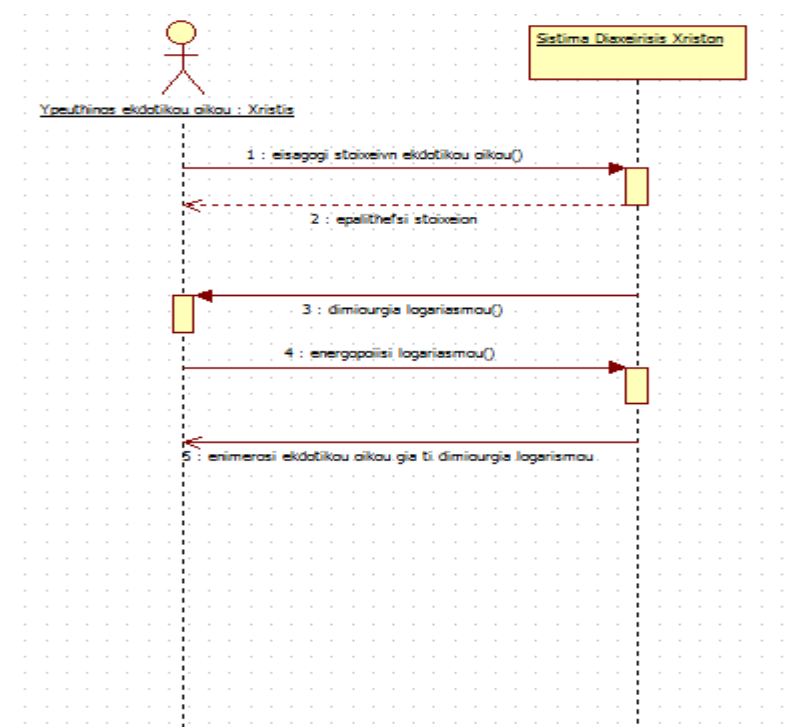
Διάγραμμα 5.12 (Λειτουργία εγγραφής χρηστών)



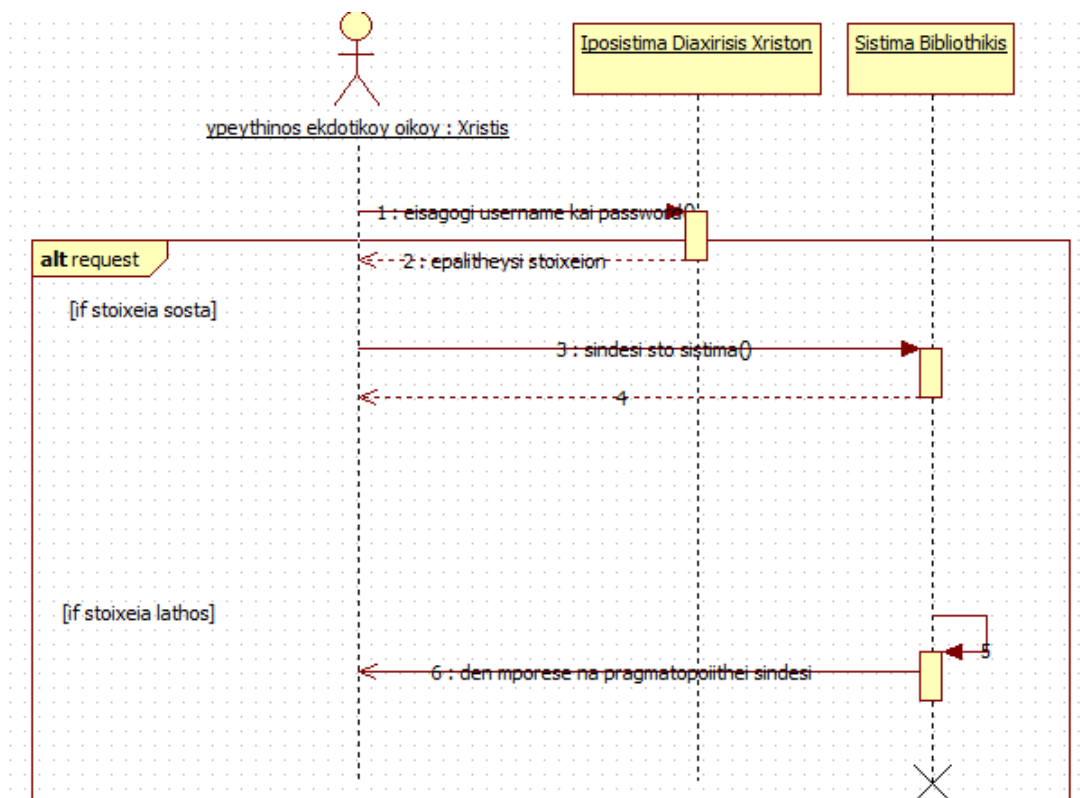
Διάγραμμα 5.13 (Λειτουργία σύνδεσης χρηστών)



Διάγραμμα 5.14 (Λειτουργία αναζήτησης χρηστών)



Διάγραμμα 5.15 (Εγγραφής εκδοτικών οίκων)



Διάγραμμα 5.16 (Σύνδεση εκδοτικών οίκων)

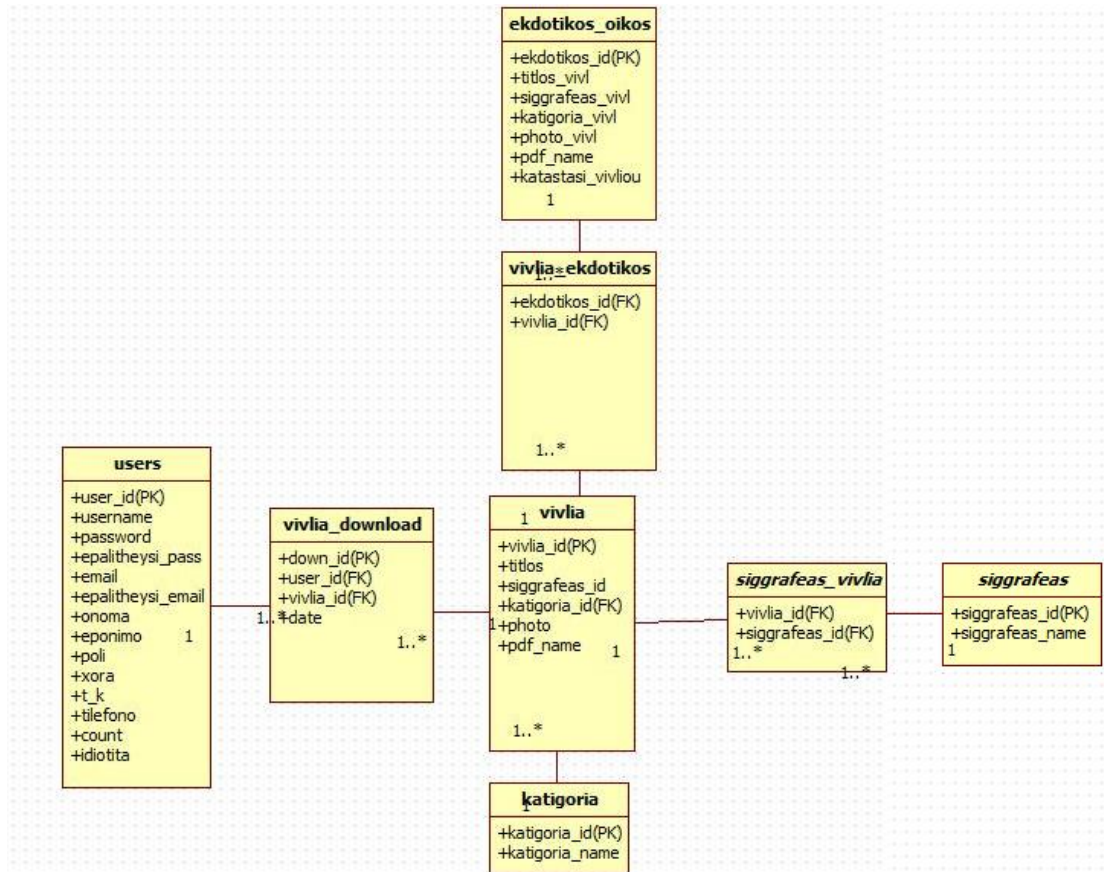
6 ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Η πτυχιακή εργασία βασίζεται πάνω σε php και MySQL. Περιλαμβάνει μια βάση δεδομένων με το όνομα «vivia» η οποία περιέχει πίνακες δεδομένων. Μέσα σε κάθε πίνακα δηλώνουμε τον τύπο κάθε πεδίου (π.χ. varchar, int) και το πρωτεύον του κλειδί (primary key).

Οι πίνακες μέσα στη βάση συνδέονται μεταξύ τους με τα πρωτεύοντα κλειδιά, έτσι όταν ένας πίνακας αλλάξει δεδομένα, να ενημερώνονται και οι υπόλοιποι πίνακες που είναι συνδεδεμένοι με αυτόν. Επίσης, μέσω των ξένων κλειδιών (foreign keys), τα οποία αποτελούνται από πεδία-ορόσημα για την περάτωση του σκοπού μας, καταφέρνουμε να συνδέσουμε δυναμικά ή στατικά τους πίνακες, κυρίως

οπτικά για να εμφανίζουμε δηλαδή τα πεδία ή τους πίνακες που επιθυμούμε κατά την περιήγηση μας στο site.

Παρακάτω παραθέτουμε την βάση μας:



Εικόνα 6.1 (Διάγραμμα βάσης)

Πίνακας	Ενέργεια	Εγγραφές ¹	Τύπος	Collation	Μέγεθος	Επιβάρυνση
ekdotikos_oikos		4	InnoDB	utf8_unicode_ci	16,0 KB	-
katigoria		17	InnoDB	utf8_unicode_ci	16,0 KB	-
siggrafeas		22	InnoDB	utf8_unicode_ci	16,0 KB	-
siggrafeas_vivlia		0	MyISAM	latin1_swedish_ci	1,0 KB	-
users		10	InnoDB	utf8_unicode_ci	16,0 KB	-
vivlia		17	InnoDB	utf8_unicode_ci	48,0 KB	-
vivlia_download		25	InnoDB	utf8_unicode_ci	64,0 KB	-
vivlia_ekdotikos		0	InnoDB	utf8_unicode_ci	48,0 KB	-
8 Πίνακας/Πίνακες	Σύνολο	95	MyISAM	latin1_swedish_ci	225,0 KB	0 Bytes

Εικόνα 6.2

Η βάση μας λοιπόν αποτελείται από τον πίνακα ‘users’ στον οποίο καταχωρούμε τους διάφορους χρήστες της εφαρμογής έχοντας τους μια ιδιότητα (Δοκιμαστικός, χρήστης με περιορισμένη πρόσβαση, εκδοτικός οίκος, administrator κ.α.), τον πίνακα ‘vivlia’ στον οποίο καταχωρούμε

τα βιβλία μας ανάλογα με την κατηγορία, συγγραφέα. Στην συνέχεια συναντάμε τον πίνακα 'katigoria' στον οποίο καταχωρούμε τις κατηγορίες βιβλίων. Ακολουθώντας την ροή των πινάκων μας συναντάμε τον πίνακα 'siggrafeas' στον οποίο καταχωρούμε του συγγραφείς των βιβλίων της εφαρμογής μας.

Έπειτα βρίσκουμε τον πίνακα 'ekdotikos_oikos' όπου και καταχωρούνται στην βάση μας τα στοιχεία των βιβλίων (π.χ. τίτλος, όνομα συγγραφέας, κατηγορία) που στέλνουν οι εκδοτικοί οίκοι προς έγκριση. Ακολουθώντας συναντάμε τον πίνακα 'vivlia_ekdotikos' ο οποίος είναι ο πίνακας που συνδέει τον πίνακα 'ekdotikos_oikos' με τον πίνακα 'vivlia' και τον πίνακα siggrafeas_vivlia ο οποίος είναι ο πίνακας που συνδέει τον πίνακα 'vivlia' με τον πίνακα 'siggrafeas' .

Ο τελευταίος πίνακας που συναντάμε είναι ο πίνακας 'vivlia_download' ο οποίος συνδέει τον πίνακα 'users' με τον πίνακα 'vivlia'.

6.1 ΠΙΝΑΚΕΣ ΚΑΙ ΠΕΔΙΑ

6.1.1 ΠΙΝΑΚΑΣ ΧΡΗΣΤΕΣ

Ο πίνακας **Χρήστες(users)** είναι ο πίνακας που είναι υπευθυνος για την διαχείριση των χρηστών και των δικαιωμάτων τους. Μέσω αυτού του πίνακα γίνεται το login και το register (το οποίο γίνεται στην σελίδα του login και ο κάθε χρήστης συμπληρώνει μια φόρμα με τα στοιχεία που του ζητούνται και είναι απαραίτητα για την εγγραφή).

Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/> user_id	smallint(4)		UNSIGNED	Όχι	None	auto_increment	
<input type="checkbox"/> username	varchar(50)	utf8_unicode_ci		Όχι			
<input type="checkbox"/> password	varchar(50)	utf8_unicode_ci		Όχι			
<input type="checkbox"/> epalithefsi_pass	varchar(50)	utf8_unicode_ci		Όχι			
<input type="checkbox"/> email	varchar(50)	utf8_unicode_ci		Όχι			
<input type="checkbox"/> epalithefsi_email	varchar(50)	utf8_unicode_ci		Όχι			
<input type="checkbox"/> onoma	varchar(50)	utf8_unicode_ci		Ναι	NULL		
<input type="checkbox"/> eponimo	varchar(50)	utf8_unicode_ci		Ναι	NULL		
<input type="checkbox"/> poli	varchar(50)	utf8_unicode_ci		Ναι	NULL		
<input type="checkbox"/> xora	varchar(50)	utf8_unicode_ci		Ναι	NULL		
<input type="checkbox"/> t_k	varchar(50)	utf8_unicode_ci		Ναι	NULL		
<input type="checkbox"/> telefono	int(10)			Ναι	NULL		
<input type="checkbox"/> count	int(10)			Όχι	None		
<input type="checkbox"/> idiotita	varchar(50)	utf8_unicode_ci		Όχι	None		

Εικόνα 6.3

Πρωτεύον κλειδί: user_id .











































Όνομα πεδίου	Τύπος πεδίου	Σύντομη περιγραφή	Αναλυτική περιγραφή	Κλειδί
User_id	Numeric(4)	Κωδικός χρήστη	Ο κάθε χρήστης έχει τον δικό του	1
username	Varchar(50)	Username του χρήστη	Το username που θα χρησιμοποιεί ο χρήστης για να κάνει login	
password	Varchar(50)	Password του χρήστη	Το password που θα χρησιμοποιεί ο χρήστης για να κάνει login	
Epalithefsi_pass	Varchar(50)	Επαλήθευση του Password του χρήστη	Επαλήθευση του password που θα χρησιμοποιεί ο χρήστης για να κάνει login	
Email	Varchar(50)	Email χρήστη	Το email του χρήστη	
Epalithefsi_email	Varchar(50)	Επαλήθευση του email του χρήστη	Επαλήθευση του email του χρήστη	
onoma	Varchar(50)	Όνομα χρήστη	Το όνομα του χρήστη	
eponimo	Varchar(50)	Επώνυμο χρήστη	Το επώνυμο του χρήστη	
poli	Varchar(50)	Πόλη διαμονής	Η πόλη διαμονής του χρήστη	
xora	Varchar(50)	Χώρα διαμονής	Η χώρα διαμονής του χρήστη	
t_k	Varchar(50)	Ταχυδρομικός κώδικας		
telefono	Int(10)	Τηλέφωνο	Το τηλέφωνο του	

		χρήστη	χρήστη σε περίπτωση που χρειαστεί	
count	Int(10)	Σύνολο βιβλίων download	Το σύνολο των βιβλίων που κατέβασε ο χρήστης-μέλος	
idiotita	Varchar(50)	Ιδιότητα χρήστη	Αν είναι δοκιμαστικός χρήστης, χρήστης με απεριόριστη ή περιορισμένη πρόσβαση, εκδοτικός οίκος.	

Πίνακας 6-1 Users(XΡΗΣΤΕΣ)

6.1.2 ΠΙΝΑΚΑΣ ΒΙΒΛΙΑ

Με τον πίνακα **Βιβλία(vivlia)** εισάγουμε στη βάση τα βιβλία της βιβλιοθήκης. Εισάγουμε στοιχεία όπως τίτλο, συγγραφέα, κατηγορία βιβλίου.

Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/> vivlia_id	smallint(4)		UNSIGNED	Όχι	None	auto_increment	      
<input type="checkbox"/> titlos	varchar(50)	utf8_unicode_ci		Όχι			      
<input type="checkbox"/> siggrafeas_id	smallint(4)		UNSIGNED	Όχι	None		      
<input type="checkbox"/> katigoria_id	smallint(4)		UNSIGNED	Όχι	None		      
<input type="checkbox"/> photo	varchar(50)	utf8_unicode_ci		Όχι			      
<input type="checkbox"/> pdf_name	varchar(50)	utf8_unicode_ci		Όχι			      

Εικόνα 6.4

Πρωτεύον κλειδί:vivlia_id.

Ξένα κλειδιά:

§ **siggrafeas_id** που αναφέρεται στον πίνακα siggrafeas,

§ **katigoria_id** που αναφέρεται στον πίνακα katigoria.

Όνομα πεδίου	Τύπος πεδίου	Σύντομη περιγραφή	Αναλυτική περιγραφή	Κλειδί	Πίνακας που αναφέρεται
Vivlia_id	Numeric(4)	A/A Βιβλίου	A/A Βιβλίου ξεχωριστός για κάθε βιβλίο	1	
titlos	Varchar(50)	Τίτλος βιβλίου	Ο τίτλος του βιβλίου		

Siggrafeas_id	Numeric(4)	A/A συγγραφέα	A/A συγγραφέα μοναδικός για τον καθένα		Siggrafeas
Kategoria_id	Numeric(4)	A/A κατηγορίας	A/A κατηγορίας ξεχωριστός για κάθε κατηγορία		Kategoria
photo	Varchar(50)	Εξώφυλλο βιβλίου	Ονομασία φωτογραφίας, εξώφυλλου του βιβλίου		
Pdf_name	Varchar(50)	Βιβλίο	Όνομα του βιβλίου σε μορφή pdf		

Πίνακας 6-2 Βιβλία(νινλία)

6.1.3 ΠΙΝΑΚΑΣ ΚΑΤΗΓΟΡΙΕΣ ΒΙΒΛΙΩΝ

Με τον πίνακα **Κατηγορίες Βιβλίων(kategoria)** εισάγουμε στην βάση μας τις κατηγορίες βιβλίων. Ο συγκεκριμένος πίνακας περιλαμβάνει το όνομα της κατηγορίας και συνδέεται άμεσα με τον πίνακα νινλία (δίνει ουσιαστικά την κατηγορία στην οποία ανήκει το κάθε βιβλίο).

Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
kategoria_id	smallint(4)		UNSIGNED	Όχι	None	auto_increment	      
kategoria_name	varchar(50)	utf8_unicode_ci		Όχι			      

Εικόνα 6.5















Πρωτεύον κλειδί: kategoria_id.

Όνομα πεδίου	Τύπος	Σύντομη περιγραφή	Αναλυτική περιγραφή	Κλειδί
Kategoria_id	Numeric(4)	A/A κατηγορίας	A/A κατηγορίας ξεχωριστός για κάθε κατηγορία	1
Kategoria_name	Varchar(50)	Όνομα κατηγορίας	Γενική ονομασία κατηγορίας	

Πίνακας 6-3 Κατηγορίες Βιβλίων(kategoria)

6.1.4 ΠΙΝΑΚΑΣ ΣΥΓΓΡΑΦΕΙΣ

Με τον πίνακα **Συγγραφείς(siggrafeas)** καταχωρούμε τους διάφορους συγγραφείς στην βάση της βιβλιοθήκης (Όνομα συγγραφέα) και καταχωρούμε μια καινούργια εγγραφή η οποία χρησιμοποιείται στον πίνακα *vivlia* που είδαμε προηγουμένως.

Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/> siggrafeas_id	smallint(4)		UNSIGNED	Όχι	None	auto_increment	      
<input type="checkbox"/> siggrafeas_name	varchar(50)	utf8_unicode_ci		Όχι			      

Εικόνα 6.6

Πρωτεύον κλειδί:siggrafeas_id.

Όνομα πεδίου	Τύπος	Σύντομη περιγραφή	Αναλυτική περιγραφή	Κλειδί
Siggrafeas_id	Numeric(4)	A/A συγγραφέα	A/A συγγραφέα για τον καθένα	1
Siggrafeas_name	Varchar(50)	Όνομα συγγραφέα	Το όνομα του εκάστοτε συγγραφέα	

Πίνακας 6-4 Συγγραφείς (siggrafeas)

6.1.5 ΠΙΝΑΚΑΣ ΜΕΤΑΦΟΡΤΩΣΗΣ ΑΡΧΕΙΩΝ

Με τον πίνακα **Μεταφόρτωσης Αρχείων (vivlia_download)** εισάγουμε στην βάση τα βιβλία που κατέβασε το κάθε μέλος. Ο πίνακας αυτός συνδέεται άμεσα με τους πίνακες *vivlia* και *users*.

Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/> down_id	smallint(4)		UNSIGNED	Όχι	None	auto_increment	      
<input type="checkbox"/> vivlia_id	smallint(4)		UNSIGNED	Όχι	None		      
<input type="checkbox"/> user_id	smallint(4)		UNSIGNED	Όχι	None		      
<input type="checkbox"/> date	datetime			Όχι	None		      

Εικόνα 6.7

Πρωτεύον κλειδί:down_id

Ξένα κλειδιά:

§ **vivlia_id** που αναφέρεται στον πίνακα *vivlia*

§ **user_id** που αναφέρεται στον πίνακα users.

Όνομα πεδίου	Τύπος	Σύντομη περιγραφή	Αναλυτική περιγραφή	Κλειδί	Πίνακας που αναφέρεται
Down_id	Numeric(4)	A/A download βιβλίων	A/A download Βιβλίων ξεχωριστό για το κάθε μέλος	1	
Vivlia_id	Numeric(4)	A/A Βιβλίου	A/A Βιβλίου ξεχωριστές για κάθε βιβλίο		vivlia
User_id	Numeric(4)	A/A χρήστη	A/A χρήστη ξεχωριστός για τον κάθε χρήστη		users
date	Datetime	Ημερομηνία download	Ημερομηνία που κατέβασε το βιβλίο		

Πίνακας 6-5 Μεταφορτώσεις Αρχείων (vivlia_download)

6.1.6 ΠΙΝΑΚΑΣ ΕΚΔΟΤΙΚΟΙ ΟΙΚΟΙ

Στον πίνακα **Εκδοτικοί Οίκοι (ekdotikos_oikos)** εισάγονται τα στοιχεία των βιβλίων (Τίτλος, όνομα συγγραφέα, κατηγορία βιβλίου κ.α.) που στέλνουν οι εκδοτικοί οίκοι προς έγκριση.

Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
ekdotikos_id	smallint(4)		UNSIGNED	Όχι	None	auto_increment	     
titlos	varchar(50)	utf8_unicode_ci		Όχι	None		     
siggrafeas_name	varchar(50)	utf8_unicode_ci		Όχι	None		     
katigoria_name	varchar(50)	utf8_unicode_ci		Όχι	None		     
photo	varchar(50)	utf8_unicode_ci		Όχι	None		     
pdf_name	varchar(50)	utf8_unicode_ci		Όχι	None		     
katastasi_vivliou	varchar(50)	utf8_unicode_ci		Όχι	None		     

Εικόνα 6.8

Πρωτεύον κλειδί:ekdotikos_id















Όνομα πεδίου	Τύπος	Σύντομη περιγραφή	Αναλυτική περιγραφή	Κλειδί

Ekdotikos_id	Numeric(4)	A/A καταχώρησης βιβλίου εκδοτικού οίκου	A/A καταχώρησης βιβλίου εκδοτικού οίκου ξεχωριστός κάθε φορά	1
Titlos	Varchar(50)	Τίτλος βιβλίου	Ο τίτλος του βιβλίου	
Siggrafeas_name	Varchar(50)	Όνομα συγγραφέα	Το όνομα του συγγραφέα	
Kategoria_name	Varchar(50)	Όνομα κατηγορίας	Γενική ονομασία κατηγορίας	
Photo	Varchar(50)	Εξώφυλλο βιβλίου	Ονομασία φωτογραφίας, εξώφυλλου του βιβλίου	
Pdf_name	Varchar(50)	Βιβλίο	Όνομα του βιβλίου σε μορφή pdf	
Katastasi_vivliou	Varchar(50)	Κατάσταση βιβλίου	Εάν το βιβλίο έχει εγκριθεί από τον ελεγκτή περιεχομένου βιβλίου ή όχι	

Πίνακας 6-6 Εκδοτικοί Οίκοι (ekdotikos_oikos)

6.1.7 ΠΙΝΑΚΑΣ ΒΙΒΛΙΑ ΣΥΓΓΡΑΦΕΑ

Ο πίνακας **Βιβλία Συγγραφέα (Siggrafeas_vivlia)** συνδέει τους πίνακες vivlia και siggrafeas.

	Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
<input type="checkbox"/>	vivlia_id	smallint(6)			Όχι	None		      
<input type="checkbox"/>	siggrafeas_id	smallint(6)			Όχι	None		      

Εικόνα 6.9

Πρωτεύον κλειδί:

§ vivlia_id

§ siggrafeas_id

Ξένα κλειδιά:

§ vivlia_id που αναφέρεται στον πίνακα vivlia

§ siggrafeas_id που αναφέρεται στον πίνακα siggrafeas.

Όνομα πεδίου	Τύπος	Σύντομη περιγραφή	Αναλυτική περιγραφή	Κλειδί	Πίνακας που αναφέρεται
Vivlia_id	Numeric(4)	A/A Βιβλίου	A/A Βιβλίου ξεχωριστές για κάθε βιβλίο		vivlia
siggrafeas_id	Numeric(4)	A/A Συγγραφέα	A/A συγγραφέα ξεχωριστό για τον καθένα		siggrafeas

Πίνακας 6-7 Βιβλία Συγγραφέα (siggrafeas_vivlia)

6.1.8 ΠΙΝΑΚΑΣ ΒΙΒΛΙΑ ΕΚΔΟΤΙΚΟΥ ΟΙΚΟΥ

Ο πίνακας **Βιβλία Εκδοτικού Οίκου (vivlia_ekdotikos)** συνδέει τους πίνακες vivlia και ekdotikos_oikos.

Πεδίο	Τύπος	Collation	Χαρακτηριστικά	Κενό	Προκαθορισμένο	Πρόσθετα	Ενέργεια
vivlia_id	smallint(4)		UNSIGNED	Όχι	None		      
ekdotikos_id	smallint(4)		UNSIGNED	Όχι	None		      

Εικόνα 6.10

Πρωτεύον κλειδί:

§ vivlia_id

§ ekdotikos_id

Ξένα κλειδιά:

§ vivlia_id που αναφέρεται στον πίνακα vivlia

§ ekdotikos_id που αναφέρεται στον πίνακα ekdotikos.

Όνομα πεδίου	Τύπος	Σύντομη περιγραφή	Αναλυτική περιγραφή	Κλειδί	Πίνακας που αναφέρεται
Vivlia_id	Numeric(4)	A/A Βιβλίου	A/A Βιβλίου		vivlia

			ξεχωριστές για κάθε βιβλίο		
ekdotikos_id	Numeric(4)	A/A Εκδοτικού οίκου	A/A εκδοτικού οίκου ξεχωριστός κάθε φορά		Ekdotikos_oikos

Πίνακας 6-8 Βιβλία Εκδοτικού Οίκου (vivlia_ekdotikos)

Παραπάνω παρατέθηκαν αναλυτικά οι πίνακες της βάσης μας. Στην πορεία θα δούμε πως χρησιμοποιούνται αυτοί οι πίνακες στην εφαρμογή μας.

7 ΠΕΡΙΓΡΑΦΗ ΕΡΓΑΛΕΙΩΝ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ

7.1 WAMP SERVER 2.0

Το wamp είναι ένα ολοκληρωμένο πακέτο Server που μπορεί να λειτουργεί σχεδόν σε οποιοδήποτε λειτουργικό σύστημα Windows. Το wamp περιλαμβάνει Apache, PHP, MySQL και όλα αυτά με αυτοματοποιημένη εγκατάσταση και ρύθμιση. Στην εφαρμογή μας χρησιμοποιήσαμε το WampServer 2.0 (PHP 5.3.0, Apache 2.2.11 και MySQL 5.1.36)



Εικόνα 7.1 (wampserver)

7.2 PHPMYADMIN

Το PhpMyAdmin είναι ένα εργαλείο γραμμένο στην γλώσσα PHP το οποίο διαχειρίζεται την MySQL στο δίκτυο. Μπορεί να χειρίζεται βάσεις δεδομένων, πίνακες, σχέσεις, πεδία και μπορεί να εκτελέσει ακόμα οτιδήποτε από SQL. Υποστηρίζει πάρα πολλές γλώσσες και μέσα σε αυτές είναι και τα Ελληνικά.

Το PhpMyAdmin έχει τις εξής δυνατότητες:

§ Δημιουργεί και διαγράφει βάσεις δεδομένων

- § Δημιουργεί, τροποποιεί, διαγράφει, αντιγράφει και μετονομάζει πίνακες
- § Προσθέτει, διαγράφει και τροποποιεί πεδία πινάκων
- § Εκτελεί SQL ερωτήματα
- § Διαχειρίζεται κλειδιά σε πεδία
- § Διαχειρίζεται πολλούς διακομιστές
- § Διαχειρίζεται τους χρήστες MySQL και τα δικαιώματά τους
- § Εκτελεί αναζητήσεις σε όλη την βάση ή μέρος από αυτήν
- § Εξάγει δεδομένα σε μορφή CSV, Latex και XML
- § Υποστηρίζει πίνακες InnoDB και ξένα κλειδιά
- § Φορτώνει αρχεία κειμένου σε πίνακες



Εικόνα 7.2 (Λογότυπο phpmyadmin)

7.3 HOMESITE

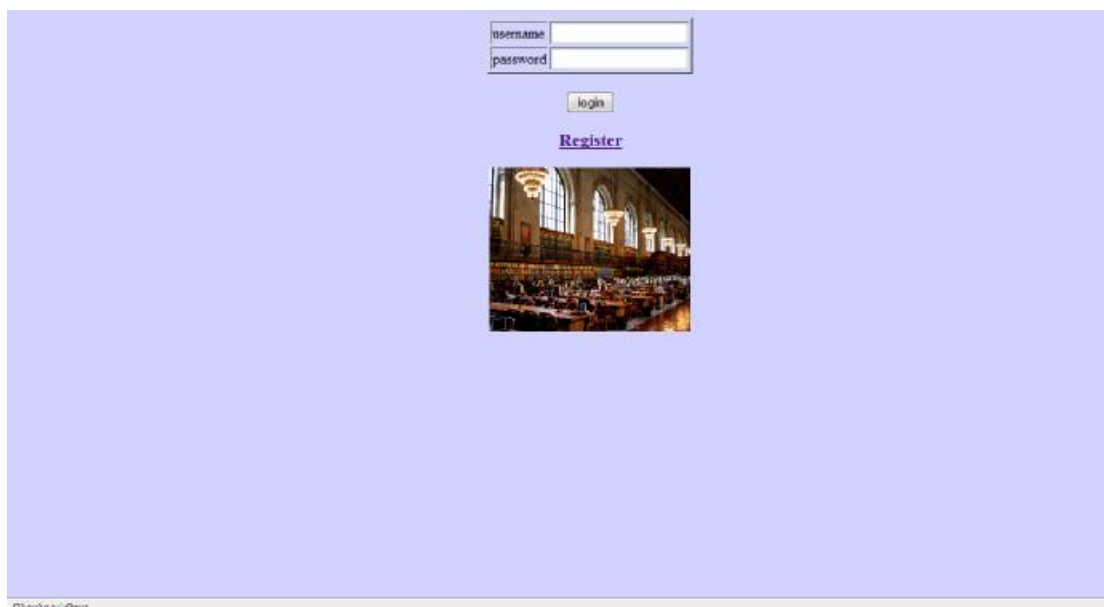
Το Homesite είναι ένα εργαλείο σχεδιασμού HTML για προγραμματιστές. Το Homesite είναι λιτό και επιτρέπει τη χρήση κώδικα για την ανάπτυξη ιστοσελίδων. Παρέχει επιπλέον προηγμένες λειτουργίες κωδικοποίησης που επιτρέπουν την άμεση δημιουργία και τροποποίηση HTML, CFML, JSP και τις ετικέτες XHTML. Το Homesite ήταν διαθέσιμο μόνο για πλατφόρμες Windows.



Εικόνα 7.3 (Λογότυπο homesite)

8 ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ

Για να εισέλθουμε στην εφαρμογή μας ανοίγουμε την σελίδα start όπως φαίνεται παρακάτω.



Εικόνα 8.1 (Σελίδα login εφαρμογής)

Έτσι λοιπόν ξεκινάει η εφαρμογή μας, η e-library υποδέχεται το χρήστη με μια φόρμα login, μέσω της οποίας μπορεί να εισέλθει ο χρήστης συμπληρώνοντας το δικό του username και password ή αν δεν ανήκει ακόμα στην βιβλιοθήκη, μπορεί απλώς να κάνει register.

8.1 ΔΙΑΚΙΩΜΑΤΑ ΜΕΛΟΥΣ

Αρχικά λοιπόν θα μιλήσουμε για τις αρμοδιότητες του κάθε μέλους καθώς εισέρχεται στην εφαρμογή.

Θα χωρίσουμε τα μέλη μας σε 2 κατηγορίες. Έχουμε λοιπόν:

- § Καινούργια μέλη-Μη εγγεγραμμένα μέλη
- § Εγγεγραμμένα μέλη

8.2 ΚΑΙΝΟΥΡΓΙΑ ΜΕΛΗ

Σ' αυτήν την κατηγορία συναντάμε τα μέλη εκείνα τα οποία έρχονται για πρώτη φορά σε επαφή με την εφαρμογή μας. Ανοίγοντας

λοιπόν την σελίδα start (βλέπουμε σχήμα 10.1) ξεκινάει η εφαρμογή μας η οποία υποδέχεται το χρήστη με μια φόρμα login.

Εδώ λοιπόν τα μέλη μας σαν καινούργια που είναι, θα πρέπει να συμπληρώσουν τα στοιχεία τους σε μια φόρμα και να αποκτήσουν έτσι με αυτό τον τρόπο τους κωδικούς τους για να έχουν πρόσβαση στην εφαρμογή. Η φόρμα η οποία συμπληρώνουν ονομάζεται register και βρίσκεται κάτω από την φόρμα στη σελίδα login. Πατώντας λοιπόν πάνω σε αυτό το link εμφανίζεται η εξής φόρμα:

	ΧΡΗΣΤΗΣ ΜΕ ΑΠΕΡΙΟΡΙΣΤΗ ΠΡΟΣΒΑΣΗ	ΧΡΗΣΤΗΣ ΜΕ ΠΕΡΙΟΡΙΣΜΕΝΗ ΠΡΟΣΒΑΣΗ	ΔΟΚΙΜΑΣΤΙΚΟΣ ΧΡΗΣΤΗΣ
Πλήρη πρόσβαση σε όλα τα βιβλία	X	-	-
Κατέβασμα 10 βιβλίων το μήνα σε μορφή pdf	-	X	-
Κατέβασμα 3 βιβλίων το μήνα σε μορφή pdf	-	-	x
Κατέβασμα ολόκληρου του βιβλίου σε μορφή pdf	X	X	X
Για την εγγραφή απαραίτητο είναι μόνο το e-mail	-	-	X
Χρηματικό ποσό	€20.00	€15.00	-
			
Για εγγραφή εκδοτικού οίκου 			

Εικόνα 8.2 (Σελίδα εγγραφής χρήστη)

Κατά την εγγραφή του ο κάθε χρήστης-μέλος θα μπορεί να επιλέξει ανάμεσα από 3 επιλογές. Κάθε επιλογή θα παραχωρεί στο χρήστη-μέλος και διαφορετικά δικαιώματα αναλόγως του ποσού που θα επιθυμεί να καταβάλει ως συνδρομή.

Εδώ επίσης υπάρχει και η επιλογή για την εγγραφή των εκδοτικών οίκων, που εφόσον εγγραφούν και αποκτήσουν username και password θα μπορούν να εισέρχονται στην εφαρμογή και να έχουν πρόσβαση μόνο στο δικό τους κομμάτι που θα είναι να στέλνουν βιβλία προς έγκριση. Εάν επιλέξει την πρώτη επιλογή (χρήστης με απεριόριστη πρόσβαση) εμφανίζεται η εξής φόρμα:

Για την εγγραφή σας πρέπει να πληρώσετε €20.00

Συμπληρώστε τα υποχρεωτικά πεδία:

Username *	<input type="text"/>
Password *	<input type="password"/>
Επιλογή Password *	<input type="text"/>
Όνομα:	<input type="text"/>
Επώνυμο:	<input type="text"/>
E-mail *	<input type="text"/>
Επιλογή E-mail *	<input type="text"/>
Πόλη:	<input type="text"/>
Χώρα:	<input type="text"/>
T.K.:	<input type="text"/>
Τηλέφωνο:	<input type="text"/>

Στοιχεία Πιστωτικής Κάρτας:

Τύπος πιστωτικής κάρτας:	Visa
Όνομα Κατόχου Κάρτας:	<input type="text"/>
Αριθμός πιστωτικής κάρτας:	<input type="text"/>
Ημερομηνία Λήξης:	01 2009
Κωδικός επιλέξεως πιστωτικής:	<input type="text"/>

Επιβεβαίωση Ακύρωση επιλογών

Εικόνα 8.3 (Φόρμα εγγραφής χρήστη με απεριόριστη πρόσβαση)

Εάν επιλέξει την δεύτερη επιλογή (χρήστης με περιορισμένη πρόσβαση) εμφανίζεται η εξής φόρμα:

Για την εγγραφή σας πρέπει να πληρώσετε €15.00

Συμπληρώστε τα υποχρεωτικά πεδία:

Username *	<input type="text"/>
Password *	<input type="password"/>
Επιλογή Password *	<input type="text"/>
Όνομα:	<input type="text"/>
Επώνυμο:	<input type="text"/>
E-mail *	<input type="text"/>
Επιλογή E-mail *	<input type="text"/>
Πόλη:	<input type="text"/>
Χώρα:	<input type="text"/>
T.K.:	<input type="text"/>
Τηλέφωνο:	<input type="text"/>

Στοιχεία Πιστωτικής Κάρτας:

Τύπος πιστωτικής κάρτας:	Visa
Όνομα Κατόχου Κάρτας:	<input type="text"/>
Αριθμός πιστωτικής κάρτας:	<input type="text"/>
Ημερομηνία Λήξης:	01 2009
Κωδικός επιλέξεως πιστωτικής:	<input type="text"/>

Επιβεβαίωση Ακύρωση επιλογών

Εικόνα 8.4 (Φόρμα εγγραφής χρήστη με περιορισμένη πρόσβαση)

Εάν επιλέξει την τρίτη επιλογή (δοκιμαστικός χρήστης) εμφανίζεται η εξής φόρμα:

Συμπληρώστε τα υποχρεωτικά πεδία:

Username *	<input type="text"/>
Password *	<input type="password"/>
Επαλήθευση password *	<input type="password"/>
E-mail *	<input type="text"/>
Επαλήθευση e-mail *	<input type="text"/>

Εικόνα 8.5 (Φόρμα εγγραφής δοκιμαστικού χρήστη)

Ενώ εάν ένας εκδοτικός οίκος θέλει να εγγραφεί επιλέγει την επιλογή εγγραφής εκδοτικών οίκων στη σελίδα register όπως είδαμε παραπάνω.

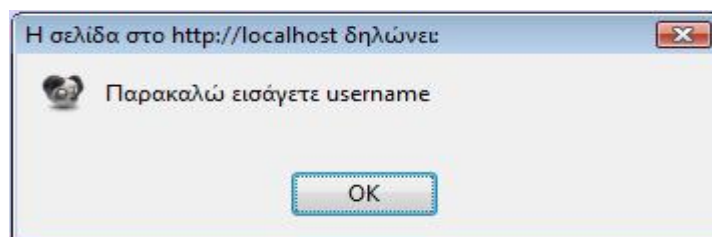
Οπότε εμφανίζεται η ακόλουθη φόρμα:

Συμπληρώστε τα υποχρεωτικά πεδία:

Username *	<input type="text"/>
Password *	<input type="password"/>
Επαλήθευση password *	<input type="password"/>
Επωνυμία *	<input type="text"/>
E-mail *	<input type="text"/>
Επαλήθευση e-mail *	<input type="text"/>
Πόλη:	<input type="text"/>
Χώρα:	<input type="text"/>
T.K.:	<input type="text"/>
Τηλέφωνο:	<input type="text"/>

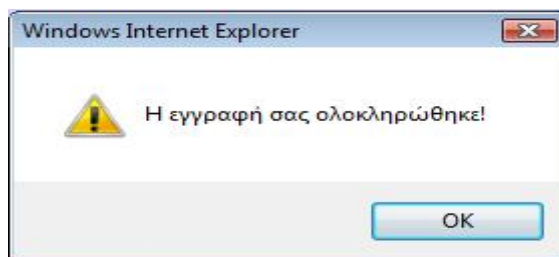
Εικόνα 8.6 (Φόρμα εγγραφής εκδοτικού οίκου)

Τα πεδία με το "*" είναι τα υποχρεωτικά πεδία της φόρμας, τα οποία αν δεν συμπληρωθούν δεν γίνεται το register. Σε περίπτωση που τα μέλη ξεχάσουν να συμπληρώσουν ένα πεδίο, το οποίο είναι υποχρεωτικό τους εμφανίζεται μήνυμα λάθους από την εφαρμογή (βλέπετε το παρακάτω σχήμα).



Εικόνα 8.7

Αφού έχουν συμπληρώσει λοιπόν σωστά τα στοιχεία τους και δεν έχουν κανένα λάθος πατάνε το κουμπί (Επιβεβαίωση). Εμφανίζεται το ακόλουθο μήνυμα.



Εικόνα 8.8

Με αυτόν τον τρόπο δημιουργούνται οι κωδικοί τους (τους οποίους έχουν επιλέξει οι ίδιοι συμπληρώνοντας την φόρμα) για να έχουν πρόσβαση στην εφαρμογή.

Έτσι λοιπόν δημιουργούνται τα καινούργια μέλη. Στην επόμενη παράγραφο θα δούμε τι συμβαίνει με τα εγγεγραμμένα μέλη.

8.3 ΕΓΓΕΓΡΑΜΜΕΝΑ ΜΕΛΗ

Κάνοντας λοιπόν login μπαίνουμε στην εφαρμογή. Η πρώτη σελίδα που συναντάμε είναι η εξής:



Εικόνα 8.9 (Αρχική σελίδα χρήστη)

Πάνω ψηλά στη σελίδα βλέπουμε το λογότυπο της βιβλιοθήκης. Από κάτω από το λογότυπο βλέπουμε το όνομα του χρήστη που έχει συνδεθεί με την εφαρμογή. Αριστερά είναι μια στήλη με links, μέσω των οποίων το μέλος μας θα περιηγηθεί όπου επιθυμεί μέσα στην εφαρμογή.

Αμέσως μετά ακολουθούν αναλυτικά οι εγγραφές που βρίσκονται στο κέντρο της σελίδας, ανάλογα με το ποιο site βλέπουμε. Πάνω από τις εγγραφές υπάρχει μια αναζήτηση με την οποία ψάχνουμε λέξεις για γρήγορη αναζήτηση ανάλογα με την εγγραφή που θέλουμε να μας εμφανίσει.

Με την ιδιότητα του μέλους μπορούμε να κάνουμε τα εξής:

- § Να κατεβάσουμε τα βιβλία που επιθυμούμε σε μορφή PDF.
- § Μπορούμε να κάνουμε περιήγηση σε όλα τα site αλλά δεν μπορούμε να πειράξουμε κάτι παρά μόνο να τα δούμε.

Είδαμε λοιπόν τις αρμοδιότητες που έχει το μέλος ας κάνουμε μια περιήγηση στην εφαρμογή για να το εξηγήσουμε καλύτερα. Ξεκινώντας από τα 'βιβλία' εμφανίζεται η παρακάτω φόρμα:



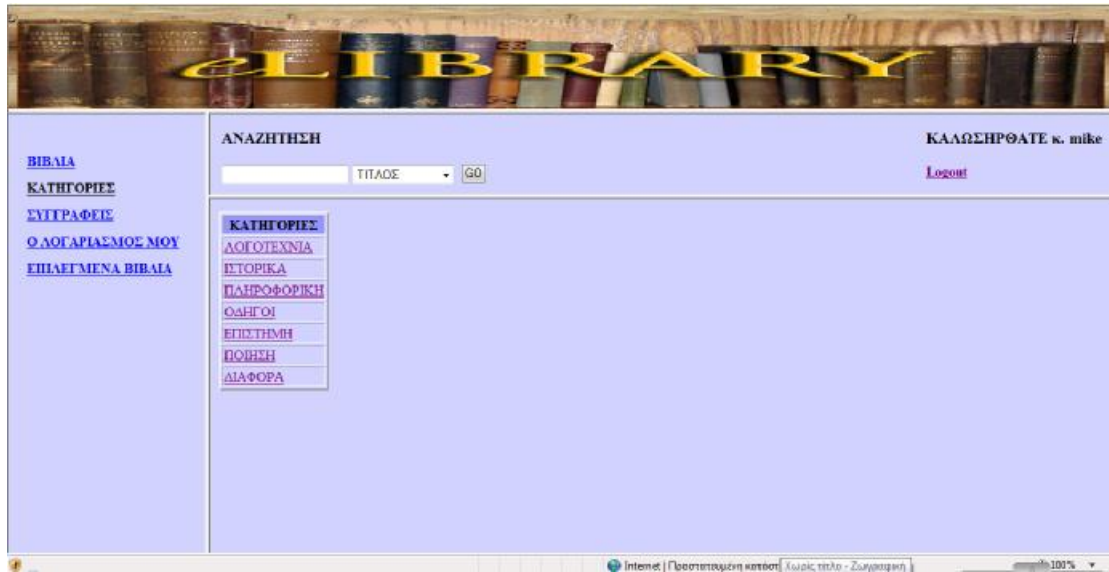
Εικόνα 8.10 (Σελίδα βιβλία)

Εδώ το μέλος έχει τη δυνατότητα να δει ποια βιβλία υπάρχουν στη βάση και αν τον ενδιαφέρει κάποιο από τα βιβλία αυτά να το προσθέσει πατώντας το πλήκτρο add ώστε να το κατεβάσει αργότερα. Πατώντας το πλήκτρο add του δίνεται η επιλογή να δει ανοίγοντας του μια καινούρια σελίδα το βιβλίο ή τα βιβλία που έχει επιλέξει (βλέπετε παρακάτω).



Εικόνα 8.11 (Προσθήκη βιβλίων για Μεταφόρτωση)

Αριστερά υπάρχει ένα link με τις κατηγορίες που χωρίζονται τα βιβλία της εφαρμογής, πατώντας το εμφανίζεται ένας πίνακας με τις υπάρχουσες κατηγορίες, (βλέπετε παρακάτω).



Εικόνα 8.12 (Κατηγορίες εφαρμογής)

Δίνοντας του την δυνατότητα επιλέγοντας μια κατηγορία να του εμφανίσει τις εγγραφές που ανήκουν στην κατηγορία αυτή (βλέπετε παρακάτω).



Εικόνα 8.13 (Βιβλία κατηγορίας Λογοτεχνία)

Έχει επίσης την δυνατότητα μέσω της αναζήτησης που υπάρχει ψηλά στη σελίδα να ψάξει ανάλογα με τον συγγραφέα ή τον τίτλο του βιβλίου για να βρει το βιβλίο που τον ενδιαφέρει. Αυτό γίνεται ως εξής:

§ Επιλεγεί από την αναδιπλούμενη λίστα αυτό που τον ενδιαφέρει π.χ. επιλέγει τον Τίτλο και γράφει στο κενό πεδίο τον Τίτλο που τον ενδιαφέρει για παράδειγμα 'PHP' και εμφανίζονται οι εγγραφές που υπάρχουν με τα στοιχεία αυτά (βλέπετε το σχήμα).



Εικόνα 8.14 (Αποτέλεσμα αναζήτησης με κριτήριο τον τίτλο)

- § Αν πάλι επιλέξουμε ανά συγγραφέα βλέπουμε πάλι τις εγγραφές εκείνες ανάλογα με τα στοιχεία που δώσαμε. Π.χ. αν επιλέξουμε σαν συγγραφέα τον 'ΠΑΠΑΔΙΑΜΑΝΤΗ'.
- § Στη συνέχεια της περιήγησής μας, συναντάμε τους 'συγγραφείς' όπου μπορούμε να δούμε τους συγγραφείς που υπάρχουν αυτή την στιγμή στην εφαρμογή.



Εικόνα 8.15 (Συγγραφείς της εφαρμογής)

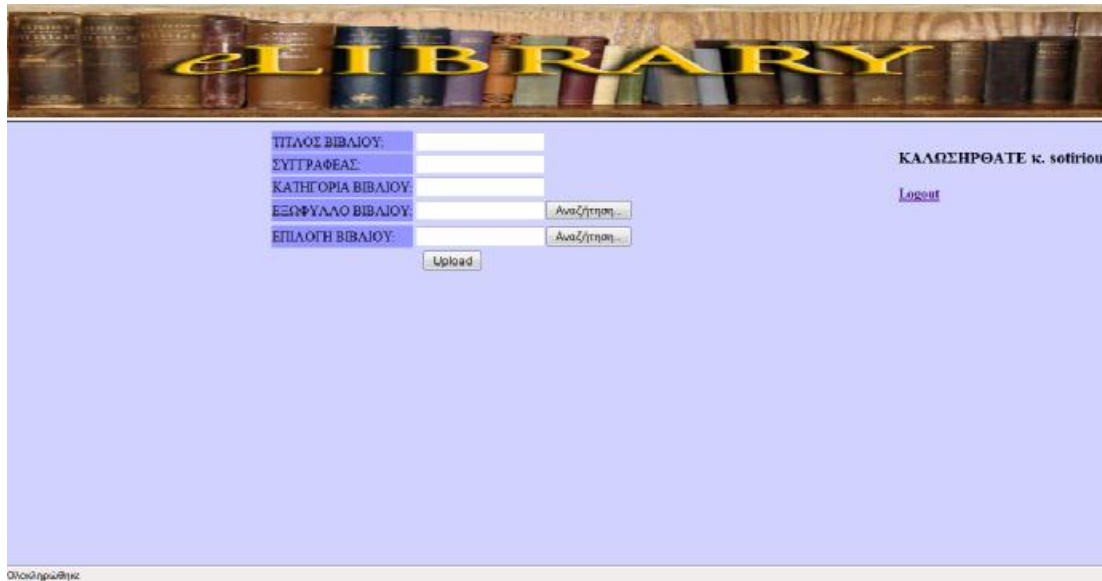
Τέλος, ακολουθεί η επιλογή 'Ο ΛΟΓΑΡΙΑΣΜΟΣ ΜΟΥ' όπου ο χρήστης μπορεί να δει κάποια προσωπικά του στοιχεία όπως επίσης και πόσα βιβλία έχει κατεβάσει μέχρι την συγκεκριμένη στιγμή (βλέπετε παρακάτω).



Εικόνα 8.16 (Εμφάνιση των στοιχείων του χρήστη)

8.4 ΕΚΔΟΤΙΚΟΣ ΟΙΚΟΣ

Συνδεδεμένοι στην εφαρμογή σαν εκδοτικός οίκος βλέπουμε την εξής σελίδα.



Εικόνα 8.17 (Αρχική σελίδα εκδοτικού οίκου)

Ο εκδοτικός οίκος έχει πρόσβαση μόνο στο δικό του κομμάτι που είναι η αποστολή βιβλίων προς έγκριση, για την έγκριση των βιβλίων αποφασίζει ο ελεγκτής περιεχομένου βιβλίων. Μέσω της εφαρμογής ο εκδοτικός οίκος μπορεί να στείλει τα βιβλία που επιθυμεί στον ελεγκτή περιεχομένου βιβλίων, σε αυτό τον βοηθάει η φόρμα που βλέπουμε παραπάνω. Ο εκδοτικός οίκος συμπληρώνει τα στοιχεία του βιβλίου και μαζί με το pdf αρχείο του βιβλίου τα αποστέλλει στον ελεγκτή περιεχομένου βιβλίων.

8.5 ΑΡΜΟΔΙΟΤΗΤΕΣ ΕΛΕΓΚΤΗ ΠΕΡΙΕΧΟΜΕΝΟΥ ΒΙΒΛΙΩΝ

Συνδεδόμενοι στην εφαρμογή σαν ελεγκτής περιεχομένου βιβλίων βλέπουμε την εξής σελίδα.



Εικόνα 8.18 (Αρχική σελίδα ελεγκτή περιεχομένων βιβλίων)

Σαν ελεγκτής περιεχομένου βιβλίων, όπως και σαν μέλος, βλέπουμε σχεδόν τα ίδια πράγματα στην πρώτη σελίδα. Αυτό που αλλάζει εδώ είναι οι αρμοδιότητες που έχει ο ελεγκτής περιεχομένων βιβλίων σε σχέση με αυτές του μέλους.

Ξεκινώντας πάλι την περιήγησή μας στην εφαρμογή με την ιδιότητα του ελεγκτή περιεχομένου βιβλίων αυτή την φορά συναντάμε την διαχείριση βιβλίων.

Εδώ ο ελεγκτής περιεχομένου βιβλίων έχει την αρμοδιότητα να ελέγχει την καταλληλότητα των βιβλίων που στέλνονται από τον υπεύθυνο των εκδοτικών οίκων και αποφασίζει ποια από τα βιβλία θα καταχωρηθούν στην βάση μας, τα υπόλοιπα έχει την δυνατότητα να τα διαγράψει.

Επίσης, μέσω της αναζήτησης ανάλογα με τις επιλογές που έχει βάλει μπορεί να βρει τις εγγραφές που τον ενδιαφέρουν χωρίς να ψάχνει κάθε μια ξεχωριστά. Για παράδειγμα εισάγει στο κενό πεδίο αυτό που τον ενδιαφέρει να ψάξει και επιλέγει από τη αναδιπλούμενη λίστα, συγγραφέα ή τίτλο ανάλογα το που ανήκει η επιλογή που έχει εισάγει στο κενό πεδίο και εμφανίζονται οι εγγραφές που επιθυμεί.

8.6 ΑΡΜΟΔΙΟΤΗΤΕΣ ADMINISTRATOR

Μέχρι στιγμής είδαμε τις αρμοδιότητες του μέλους και του ελεγκτή περιεχομένου βιβλίου ήρθε λοιπόν η στιγμή να μιλήσουμε και για τις αρμοδιότητες του administrator. Συνδέοντας στην εφαρμογή σαν administrator βλέπουμε τα εξής:



Εικόνα 8.19 (Αρχική σελίδα administrator)

Όπως και στα μέλη και στον ελεγκτή περιεχομένου βιβλίων η πρώτη σελίδα είναι ίδια, με τη διαφορά ότι σαν administrator βλέπουμε όλα τα πιθανά sites της εφαρμογής.

Όπως είναι λογικό, ο ρόλος του administrator είναι πολύ σημαντικός στην εφαρμογή. Επιλέγοντας από την αριστερή στήλη 'ΧΡΗΣΤΕΣ' οδηγούμαστε στο site με όλα τα μέλη της εφαρμογής έχοντας την δυνατότητα να δει τα στοιχεία των χρηστών και να διαγράψει κάποιο χρήστη αν αυτό είναι απαραίτητο.

Username	Password	ΟΝΟΜΑ	ΕΠΩΝΥΜΟ	ΛΙΟΤΗΤΑ	Email	ΠΟΛΗ	ΧΩΡΑ	T.K.	ΤΗΛΕΦΩΝΟ	
Antonis_Antoniou	admin	Αντώνης	Αντωνίου	Administrator	admin@hotmail.com	Λεμεσός	Κύπρος	11111	2147483647	<input type="checkbox"/>
Mavrikaki	m	Μαρία	Μαυρικάκη	ΕΛΕΓΚΤΗΣ	mavrikaki@hotmail.com	Ρόδος	Ελλάδα	85101	693456788	<input type="checkbox"/>
Mike	Mike	Μιχάλης	Πέττας	ΔΟΚΙΜΑΣΤΙΚΟΣ	mike@hotmail.com	Ρόδος	Ελλάδα	85101	697456789	<input type="checkbox"/>
irine	12345	Ειρήνη	Μαυρικάκη	ΠΛΗΡΗ_ΠΡΟΣΒΑΣΗ	irine@hotmail.com	Αθήνα	Ελλάδα	45678	2147483647	<input type="checkbox"/>
eustathiou	eustathiou	Ευσταθίου		ΕΚΔΟΤΙΚΟΣ_ΟΙΚΟΣ	eustathiou@hotmail.com	Αθήνα	Ελλάδα	12654	2147483647	<input type="checkbox"/>
sotiriou	sotiriou	ΣΩΤΗΡΙΟΥ		ΕΚΔΟΤΙΚΟΣ_ΟΙΚΟΣ	sotiriou@hotmail.com	Κρήτη	Ελλάδα	90789	2147483647	<input type="checkbox"/>
chris	mav	Χρήστος	Αλεξίου	ΠΛΗΡΗ_ΠΡΟΣΒΑΣΗ	chris@hotmail.com	Πάτρα	Ελλάδα	27200	262207896	<input type="checkbox"/>
kkk	kkkk	kkkk	kk	ΠΛΗΡΗΣ_ΠΡΟΣΒΑΣΗ	kkk				0	<input type="checkbox"/>

Delete Records

Εικόνα 8.20 (Οι χρήστες της εφαρμογής)

Επιλέγοντας από την αριστερή στήλη ‘ΒΙΒΛΙΑ ΕΛΕΓΚΤΗ’ οδηγούμαστε στο site με τα βιβλία που έχει εγκρίνει ο ελεγκτής περιεχομένου βιβλίων.

The screenshot shows a web application interface for a library. At the top, there's a header with the word 'LIBRARY' in large yellow letters over a background of books. Below the header, there's a search bar with a dropdown menu for 'ΤΙΤΛΟΣ' and a 'GO' button. On the right side of the header, it says 'ΚΑΛΩΣΗΡΘΑΤΕ κ. antonis' and a 'Logout' link. On the left side, there's a sidebar with navigation links: 'ΚΑΤΗΓΟΡΙΕΣ', 'ΒΙΒΛΙΑ', 'ΣΥΓΓΡΑΦΕΙΣ', 'ΚΑΤΗΓΟΡΙΕΣ', 'ΠΡΟΣΘΕΣΗ ΚΑΤΗΓΟΡΙΑΣ', 'ΔΙΑΓΡΑΦΗ ΚΑΤΗΓΟΡΙΑΣ', 'ΕΝΗΜΕΡΩΣΗ ΒΙΒΛΙΩΝ', 'ΒΙΒΛΙΑ ΕΛΕΓΚΤΗ', 'UPLOAD', and 'ΧΡΗΣΤΕΣ'. The main content area is a table with columns: 'ΕΞΟΦΥΛΛΙΟ', 'ΤΙΤΛΟΣ', 'ΣΥΓΓΡΑΦΕΑΣ', and 'ΚΑΤΗΓΟΡΙΑ'. There are four rows of books listed, each with a small book cover icon, a title, an author name, a category, and a 'Download' button with a downward arrow.

Εικόνα 8.21 (Εγκεκριμένα βιβλία από ελεγκτή)

Τα βιβλία που έχει εγκρίνει ο ελεγκτής περιεχομένου βιβλίων τα προσθέτει μέσω της επιλογής upload ο administrator στη βάση της εφαρμογής (βλέπετε το ακόλουθο σχήμα).

Εικόνα 8.22 (Upload φόρμα βιβλίου)

Συμπληρώνοντας τα απαραίτητα πεδία που αφορούν το βιβλίο, προσθέτουν ένα καινούργιο βιβλίο στη βάση. Από εδώ ο administrator τοποθετεί τα βιβλία σε μια κατηγορία, με αυτόν τον τρόπο ουσιαστικά ταξινομεί τα βιβλία. Επίσης επιλέγει τον αντίστοιχο συγγραφέα. Με αυτόν λοιπόν τον τρόπο ολοκληρώνεται η ταξινόμηση των βιβλίων, τα οποία ανήκουν πλέον σε μια συγκεκριμένη κατηγορία, με έναν συγκεκριμένο συγγραφέα.

Επιπλέον ο administrator έχει τη δυνατότητα να προσθέτει νέες κατηγορίες και να διαγράφει κατηγορίες επιλέγοντας τα αντίστοιχα links από το αριστερό menu. Για να προσθέσει μια νέα κατηγορία συμπληρώνει το όνομά της στην φόρμα που ακολουθεί.

Εικόνα 8.23 (Φόρμα προσθήκης κατηγορίας)

9 ΣΥΜΠΕΡΑΣΜΑΤΑ

Με την παρούσα πτυχιακή πιστεύουμε ότι πετύχαμε τον αρχικό στόχο, δηλαδή την ανάλυση, τον σχεδιασμό και την ανάπτυξη μιας web εφαρμογής ψηφιακής βιβλιοθήκης e-library.

Η εφαρμογή αυτή θα καλύπτει τις βασικές λειτουργίες που γίνονται με σκοπό να ολοκληρωθεί το “download” ενός βιβλίου. Επομένως, διευκολύνθηκε η διαδικασία αναζήτησης και η δυνατότητα ανάγνωσης ενός βιβλίου που μέχρι στιγμής ήταν απαραίτητη η φυσική παρουσία του προσώπου στο χώρο της βιβλιοθήκης. Αυτό έχει ως αποτέλεσμα την αύξηση της παραγωγικότητας, της επιχειρηματικότητας καθώς και την διευκόλυνση και επιτάχυνση των διαδικασιών.

10 ΜΕΛΛΟΝΤΙΚΕΣ ΒΕΛΤΙΩΣΕΙΣ

Μελλοντικά, η εφαρμογή μπορεί να δεχτεί κάποιες βελτιώσεις που αφορούν κυρίως το marketing και την καλύτερη εξυπηρέτηση των χρηστών. Οι βελτιώσεις αυτές, όπως και κάποιες δυνατότητες που μπορούν να προστεθούν στην εφαρμογή για την καλύτερη εξυπηρέτηση των χρηστών αναφέρονται παρακάτω:

- § Συλλογή πληροφοριών όπως για παράδειγμα ιστορικό download βιβλίων και προτιμήσεων κ.α. και αξιοποίησή τους για την παροχή όσο το δυνατόν καλύτερων υπηρεσιών προς τους χρήστες-μέλη.
- § Διατήρηση πελατών μέσω καλύτερης εξυπηρέτησης. Όπως για παράδειγμα
- § Απαντήσεις που πρέπει να δίνονται στον χρήστη-μέλος σε κάθε πιθανή ερώτηση ή παράπονο του.
- § Μετατροπή των επισκεπτών σε αγοραστές. Έχοντας τη δυνατότητα καταγραφής του χρόνου και των επισκέψεων που κάνει ένα μέλος σε μια κατηγορία βιβλίων και εφόσον το ίδιο μέλος έχει πραγματοποιήσει και στο παρελθόν αγορά από την ίδια κατηγορία μπορεί να του σταλεί ένα ηλεκτρονικό πάντα εκπωτικό κουπόνι για ένα άλλο βιβλίο της αντίστοιχης κατηγορίας.
- § Συνεργασία με άλλες, μη ανταγωνιστικές εταιρείες και αγορά ή ανταλλαγή δεδομένων.
- § Ανάλυση εκστρατείας (ποιοι πελάτες ανταποκρίνονται, πόσο συχνά (ηλεκτρονικά κουπόνια, προσφορές) .

§ Αξιοποίηση του «ιογενούς» (viral) marketing η οποία δίνει τη δυνατότητα στα μέλη να στέλνουν σε γνωστούς τους e-mails, προτείνοντας τους συγκεκριμένα βιβλία.

Επίσης μελλοντικά θα μπορούσαν να προστεθούν δυνατότητές όπως:

§ Newsletters για την ενημέρωση των μελών.

§ Διαφημιστικές εκστρατείες βασισμένες σε συγκεκριμένες εκδηλώσεις ή ημερομηνίες για παράδειγμα ειδικές προσφορές για βιβλία μια συγκεκριμένη ημέρα του έτους.

§ Εφαρμογές αποστολής απαντητικών e-mail για συγκεκριμένες ερωτήσεις ή περιστάσεις (π.χ. επιβεβαίωση εγγραφής και αποστολή των κωδικών πρόσβασης στην εφαρμογή στο email που έχει δηλώσει ο χρήστης-μέλος.

ΠΑΡΑΡΤΗΜΑ

Πληροφορίες βάσης δεδομένων, πίνακες βάσης, πρωτεύοντα και δευτερεύοντα κλειδιά.

1. Database

Database Description

Database Annotation

Database Name vivlia

Database Character Set

Database Collate

Comment

Database Create SQL CREATE DATABASE IF NOT EXISTS `vivlia1`;
USE `vivlia1`;

2. Tables

2.1 ekdotikos_oikos

Table Description

Table Annotation

Param Name Param Value

Table Type INNODB

Row Format DEFAULT

Temporary NO

Row Checksum NO

Column Name	Data Type	Primary Key	Not Null	AutoInc	Flags	Default Value	Comment
ekdotikos_id	smallint(4)	YES	YES	YES	UNSIGNED		
titlos	varchar(50)	NO	YES	NO			
siggrafeas_name	varchar(50)	NO	YES	NO			
katigoria_name	varchar(50)	NO	YES	NO			
photo	varchar(50)	NO	YES	NO			
pdf_name	varchar(50)	NO	YES	NO			

Index Name Columns

PRIMARY ekdotikos_id

Partition Name Partition Comment Subpartitions

Trigger Name Trigger Time Trigger Event Definer

Table Create SQL CREATE TABLE `ekdotikos_oikos` (
`ekdotikos_id` smallint(4) UNSIGNED NOT NULL
AUTO_INCREMENT,
`titlos` varchar(50) CHARACTER SET utf8 COLLATE
utf8_unicode_ci NOT NULL,
`siggrafeas_name` varchar(50) CHARACTER SET utf8 COLLATE
utf8_unicode_ci NOT NULL,
`katigoria_name` varchar(50) CHARACTER SET utf8 COLLATE


```

utf8_unicode_ci NOT NULL,
  `photo` varchar(50) CHARACTER SET utf8 COLLATE
utf8_unicode_ci NOT NULL,
  `pdf_name` varchar(50) CHARACTER SET utf8 COLLATE
utf8_unicode_ci NOT NULL,
  PRIMARY KEY(`ekdotikos_id`)
)
ENGINE=INNODB
COMMENT = 'InnoDB free: 9216 kB';

```

2.2 *katigoria*

Table Description

Table Annotation

Param Name	Param Value
Table Type	INNODB
Row Format	DEFAULT
Temporary	NO
Row Checksum	NO

Column Name	Data Type	Primary Key	Not Null	AutoInc	Flags	Default Value	Comment
katigoria_id	smallint(4)	YES	YES	YES	UNSIGNED		
katigoria_name	varchar(50)	NO	YES	NO			

Index Name	Columns
PRIMARY	katigoria_id

Partition Name	Partition Comment	Subpartitions
----------------	-------------------	---------------

Trigger Name	Trigger Time	Trigger Event	Definer
--------------	--------------	---------------	---------

Table Create SQL CREATE TABLE `katigoria` (
 `katigoria_id` smallint(4) UNSIGNED NOT NULL
 AUTO_INCREMENT,
 `katigoria_name` varchar(50) CHARACTER SET utf8 COLLATE
 utf8_unicode_ci NOT NULL,
 PRIMARY KEY(`katigoria_id`)
)
 ENGINE=INNODB
 COMMENT = 'InnoDB free: 9216 kB';

2.3 *siggrafeas*

Table Description

Table Annotation

Param Name	Param Value
Table Type	INNODB
Row Format	DEFAULT
Temporary	NO
Row Checksum	NO

Column Name	Data Type	Primary Key	Not Null	AutoInc	Flags	Default Value	Comment
siggrafeas_id	smallint(4)	YES	YES	YES	UNSIGNED		
siggrafeas_name	varchar(50)	NO	YES	NO			

Index Name	Columns
PRIMARY	siggrafeas_id

Partition Name	Partition Comment	Subpartitions
----------------	-------------------	---------------

Trigger Name	Trigger Time	Trigger Event	Definer
--------------	--------------	---------------	---------

Table Create SQL CREATE TABLE `siggrafeas` (
 siggrafeas_id smallint(4) UNSIGNED NOT NULL
 AUTO_INCREMENT,
 siggrafeas_name varchar(50) CHARACTER SET utf8 COLLATE
 utf8_unicode_ci NOT NULL,
 PRIMARY KEY(siggrafeas_id)
)
 ENGINE=INNODB
 COMMENT = 'InnoDB free: 9216 kB';

```

        `siggrafeas_id` smallint(4) UNSIGNED NOT NULL
        AUTO_INCREMENT,
        `siggrafeas_name` varchar(50) CHARACTER SET utf8 COLLATE
        utf8_unicode_ci NOT NULL,
        PRIMARY KEY(`siggrafeas_id`)
    )
    ENGINE=INNODB
    COMMENT = 'InnoDB free: 9216 kB';

```

2.4 users

Table Description

Table Annotation

Param Name	Param Value
Table Type	INNODB
Row Format	DEFAULT
Temporary	NO
Row Checksum	NO

Column Name	Data Type	Primary Key	Not Null	AutoInc	Flags	Default Value	Comment
user_id	smallint(4)	YES	YES	YES	UNSIGNED		
username	varchar(50)	YES	YES	NO			
password	varchar(50)	NO	YES	NO			
epalithefsi_pass	varchar(50)	NO	YES	NO			
email	varchar(50)	NO	YES	NO			
epalithefsi_email	varchar(50)	NO	YES	NO			
onoma	varchar(50)	NO	NO	NO			
eponimo	varchar(50)	NO	NO	NO			
poli	varchar(50)	NO	NO	NO			
xora	varchar(50)	NO	NO	NO			
t_k	varchar(50)	NO	NO	NO			
telefono	int(10)	NO	NO	NO			
count	int(10)	NO	NO	NO			
idiotita	varchar(50)	NO	NO	NO			

Index Name	Columns
PRIMARY	user_id,username

Partition Name	Partition Comment	Subpartitions
Trigger Name	Trigger Time	Trigger Event Definer

```

Table Create SQL CREATE TABLE `users` (
    `user_id` smallint(4) UNSIGNED NOT NULL AUTO_INCREMENT,
    `username` varchar(50) CHARACTER SET utf8 COLLATE
    utf8_unicode_ci NOT NULL,
    `password` varchar(50) CHARACTER SET utf8 COLLATE
    utf8_unicode_ci NOT NULL,
    `epalithefsi_pass` varchar(50) CHARACTER SET utf8 COLLATE
    utf8_unicode_ci NOT NULL,
    `email` varchar(50) CHARACTER SET utf8 COLLATE
    utf8_unicode_ci NOT NULL,
    `epalithefsi_email` varchar(50) CHARACTER SET utf8 COLLATE
    utf8_unicode_ci NOT NULL,
    `onoma` varchar(50) CHARACTER SET utf8 COLLATE
    utf8_unicode_ci,
    `eponimo` varchar(50) CHARACTER SET utf8 COLLATE
    utf8_unicode_ci,

```

```

    `poli` varchar(50) CHARACTER SET utf8 COLLATE
utf8_unicode_ci,
    `xora` varchar(50) CHARACTER SET utf8 COLLATE
utf8_unicode_ci,
    `t_k` varchar(50) CHARACTER SET utf8 COLLATE
utf8_unicode_ci,
    `telefono` int(10),
    `count` int(10),
    `idiotita` varchar(50) CHARACTER SET utf8 COLLATE
utf8_unicode_ci,
    PRIMARY KEY(`user_id`, `username`)
)
ENGINE=INNODB
COMMENT = 'InnoDB free: 9216 kB';

```

2.5 vivlia

Table Description

Table Annotation

Param Name	Param Value
Table Type	INNODB
Row Format	DEFAULT
Temporary	NO
Row Checksum	NO

Column Name	Data Type	Primary Key	Not Null	AutoInc	Flags	Default Value	Comment
vivlia_id	smallint(4)	YES	YES	YES	UNSIGNED		
siggrafeas_id	smallint(4)	YES	YES	NO	UNSIGNED		
katigoria_id	smallint(4)	YES	YES	NO	UNSIGNED		
titlos	varchar(50)	NO	YES	NO			
photo	varchar(50)	NO	YES	NO			
pdf_name	varchar(50)	NO	YES	NO			

Index Name	Columns
PRIMARY	vivlia_id,siggrafeas_id,katigoria_id
siggrafeas_id	siggrafeas_id
katigoria_id	katigoria_id

Partition Name	Partition Comment	Subpartitions
Trigger Name	Trigger Time	Trigger Event Definer

```

Table Create SQL CREATE TABLE `vivlia` (
    `vivlia_id` smallint(4) UNSIGNED NOT NULL AUTO_INCREMENT,
    `siggrafeas_id` smallint(4) UNSIGNED NOT NULL,
    `katigoria_id` smallint(4) UNSIGNED NOT NULL,
    `titlos` varchar(50) CHARACTER SET utf8 COLLATE
utf8_unicode_ci NOT NULL,
    `photo` varchar(50) CHARACTER SET utf8 COLLATE
utf8_unicode_ci NOT NULL,
    `pdf_name` varchar(50) CHARACTER SET utf8 COLLATE
utf8_unicode_ci NOT NULL,
    PRIMARY KEY(`vivlia_id`, `siggrafeas_id`, `katigoria_id`),
    INDEX `siggrafeas_id`(`siggrafeas_id`),
    INDEX `katigoria_id`(`katigoria_id`),
    CONSTRAINT `vivlia_ibfk_1` FOREIGN KEY (`siggrafeas_id`)
REFERENCES `siggrafeas`(`siggrafeas_id`)
ON DELETE RESTRICT
ON UPDATE RESTRICT,
    CONSTRAINT `vivlia_ibfk_2` FOREIGN KEY (`katigoria_id`)
REFERENCES `katigoria`(`katigoria_id`)

```

```

        ON DELETE RESTRICT
        ON UPDATE RESTRICT
    )
ENGINE=INNODB
COMMENT = 'InnoDB free: 9216 kB; (`siggrafeas_id`) REFER
`vivlial/siggrafeas`(`siggrafeas_i`';

```

2.6 vivlia_download

Table Description

Table Annotation

Param Name	Param Value
Table Type	INNODB
Row Format	DEFAULT
Temporary	NO
Row Checksum	NO

Column Name	Data Type	Primary Key	Not Null	AutoInc	Flags	Default Value	Comment
down_id	smallint(4)	YES	YES	YES	UNSIGNED		
vivlia_id	smallint(4)	NO	YES	NO	UNSIGNED		
user_id	smallint(4)	NO	YES	NO	UNSIGNED		
date	timestamp	NO	YES	NO		CURRENT_TIMESTAMP	
username	varchar(50)	NO	YES	NO			

Index Name	Columns
PRIMARY	down_id
vivlia_id	vivlia_id
user_id	user_id
username	username

Partition Name	Partition Comment	Subpartitions
----------------	-------------------	---------------

Trigger Name	Trigger Time	Trigger Event	Definer
--------------	--------------	---------------	---------

```

Table Create SQL CREATE TABLE `vivlia_download` (
    `down_id` smallint(4) UNSIGNED NOT NULL AUTO_INCREMENT,
    `vivlia_id` smallint(4) UNSIGNED NOT NULL,
    `user_id` smallint(4) UNSIGNED NOT NULL,
    `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `username` varchar(50) CHARACTER SET utf8 COLLATE
utf8_unicode_ci NOT NULL,
    PRIMARY KEY(`down_id`),
    INDEX `vivlia_id`(`vivlia_id`),
    INDEX `user_id`(`user_id`),
    INDEX `username`(`username`)
)
ENGINE=INNODB
COMMENT = 'InnoDB free: 9216 kB';

```

2.7 vivlia_ekdotikos

Table Description

Table Annotation

Param Name	Param Value
Table Type	INNODB
Row Format	DEFAULT
Temporary	NO

Row Checksum NO

Column Name	Data Type	Primary Key	Not Null	AutoInc	Flags	Default Value	Comment
vivlia_id	smallint(4)	NO	YES	NO	UNSIGNED		
ekdotikos_id	smallint(4)	NO	YES	NO	UNSIGNED		

Index Name	Columns
vivlia_id	vivlia_id
ekdotikos_id	ekdotikos_id

Partition Name	Partition Comment	Subpartitions
----------------	-------------------	---------------

Trigger Name	Trigger Time	Trigger Event	Definer
--------------	--------------	---------------	---------

```
Table Create SQL CREATE TABLE `vivlia_ekdotikos` (  
    `vivlia_id` smallint(4) UNSIGNED NOT NULL,  
    `ekdotikos_id` smallint(4) UNSIGNED NOT NULL,  
    INDEX `vivlia_id`(`vivlia_id`),  
    INDEX `ekdotikos_id`(`ekdotikos_id`)  
)  
ENGINE=INNODB  
COMMENT = 'InnoDB free: 9216 kB';
```

3. References

3.1 vivlia_ibfk_1

Reference Description

Reference Annotation

Source Table	Target Table	Delete Action	Update Action	Link
siggrafeas	vivlia	RESTRICT	RESTRICT	siggrafeas_id=siggrafeas_id

3.2 vivlia_ibfk_2

Reference Description

Reference Annotation

Source Table	Target Table	Delete Action	Update Action	Link
katigoria	vivlia	RESTRICT	RESTRICT	katigoria_id=katigoria_id

11 ΠΗΓΕΣ-ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλιογραφία

1. Ian Sommerville, M. K. (2008). *Software engineering*. Pearson Education Limited.
2. Madachy, R. J. (2008). *Software Process Dynamics*. Wiley-IEEE Press.
3. Madnick, T. K.-H. (1991). *Software Project Dynamics: An Integrated Approach*. Prentice-Hall.
4. Melony, J. C.(2000) *PHP, MySql και Apache*. Μ.Γκιούρδας.
5. Murr, L. E. και Williams, J. B. (1987): The Roles of the Future Library.
Library Hi Tech

Πηγές από διαδίκτυο

6. http://en.wikipedia.org/wiki/Web_application
7. <http://www.w3schools.com>
8. <http://greetuts.net>
9. <http://www.techteam.gr>
10. <http://php.net>
11. <http://www.php.net/manual/en/book.mysql.php>
12. <http://sql.org/sql-database/mysql/>
13. <http://www.apache.org>
14. <http://www.phpmyadmin.net/documentation>
15. <http://www.mysql.com>
16. <http://www.freestuff.gr/>
17. <http://www.it.uom.gr/project/html2/main.html>

18. <http://www.tizag.com/phpT/>
19. <http://en.wikipedia.org/wiki/JavaScript>
20. <http://www.insomnia.gr/forum/>
21. <http://giorgosbiz.wordpress.com/2009/07/11/>
22. http://www.ergasya.tuc.gr/Users/matsatsinis/courses/tech_log_slides/lectures.ppt