



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ  
ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Πανεπιστήμιο Δυτικής Μακεδονίας

Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών

# Σχεδιασμός πληροφοριακού συστήματος διαχείρισης μητρώου υλικών

---

Χριστοφορίδης Ιωάννης Ραφαήλ

Επιβλέπων καθηγητής : Δασυγένης Μηνάς

Κοζάνη, Ιούλιος 2014

## Περιεχόμενα

Περίληψη .....	5
Abstract.....	6
Εισαγωγή.....	7
Κεφάλαιο 1: Αρχιτεκτονική.....	10
1.1 HTML .....	11
1.2 CSS .....	12
1.3 PHP .....	13
1.4 MySQL .....	14
1.5 JavaScript.....	15
1.6 Κώδικας QR .....	16
Κεφάλαιο 2: Επισκόπηση δομής .....	18
2.1 Αρχεία.....	19
2.2 Σχεδιασμός Βάσης Δεδομένων .....	24
Κεφάλαιο 3: Εργαλεία .....	31
3.1 WinSCP .....	31
3.2 Notepad++ .....	33
3.3 Putty .....	34
Κεφάλαιο 4: Ανάλυση κώδικα .....	35
4.1 header.php & footer.php .....	35
4.2 functions.php .....	36
4.3 index.php.....	37
4.4 usercp.php.....	38
4.5 users.php.....	41

4.6 manage.php.....	45
4.7 permissions.php .....	50
4.8 addit.php .....	51
4.9 search.php.....	56
4.10 qrcode.php & showconfs.php.....	57
4.11 pdf.php .....	58
4.12 watchers.php & watchershelper.php .....	59
Κεφάλαιο 5: Οδηγίες χρήσης.....	60
5.1 Οδηγίες Εγκατάστασης .....	61
5.2 Προσθήκη Αντικειμένου .....	63
5.3 Εύρεση Αντικειμένου .....	65
5.4 Εκτύπωση Ετικετών.....	67
5.5 Χρήστες/Ομάδες .....	69
5.6 Διαχείριση Δομής.....	71
5.7 Δικαιώματα .....	74
5.8 Παραχώρηση Αντικειμένων .....	75
5.9 Ρυθμίσεις.....	77
5.10 Έξοδος .....	77
Κεφάλαιο 7: Συμπεράσματα και μελλοντικές χρήσεις.....	78
Βιογραφικό σημείωμα .....	81
Βιβλιογραφία .....	82

## Λίστα Εικόνων

Εικόνα 1: Παράδειγμα QR ετικέτας.....	17
Εικόνα 2: Αρχεία συστήματος διαχείρισης μητρώου υλικών .....	19
Εικόνα 3: Διάγραμμα βάσης δεδομένων.....	26

Εικόνα 4: Το περιβάλλον λειτουργίας του WinSCP .....	32
Εικόνα 5 Το περιβάλλον λειτουργίας του Notepad++ .....	33
Εικόνα 6: Το περιβάλλον λειτουργίας του PUTTY .....	34
Εικόνα 7: Διάγραμμα χρήσης .....	60
Εικόνα 8: Παράδειγμα περιεχομένου του αρχείου config.php που χρησιμοποιήθηκε στον server "zafora". .....	62
Εικόνα 9: Επιλογή υποκατηγορίας .....	64
Εικόνα 10: Προσθήκη στοιχείων αντικειμένου .....	64
Εικόνα 11: Παράδειγμα αναζήτησης αντικειμένου .....	66
Εικόνα 12: Παράδειγμα επεξεργασίας αντικειμένου .....	66
Εικόνα 13: Οδηγός χωροθέτησης ετικετών για εκτύπωση .....	68
Εικόνα 14: Παράδειγμα εξαναγκασμού χρήστη να αλλάξει τον κωδικό του. ....	70
Εικόνα 15: Παράδειγμα ορισμού κοινού στοιχείου .....	73
Εικόνα 16: Παράδειγμα προσθήκης νέας υποκατηγορίας .....	73
Εικόνα 17: Παράδειγμα προσθήκης δικαιώματος .....	74
Εικόνα 18: Παράδειγμα παραχώρησης αντικειμένου .....	76
Εικόνα 19: Ρυθμίσεις χρήστη.....	77

## Λίστα Πινάκων

Πίνακας 1:Ανάλυση περιεχομένου κώδικα .....	8
---	---

## Περίληψη

Η τεχνολογία και οι αυτοματισμοί είναι πλέον ένα κοινό στοιχείο στην κοινωνία μας. Είναι γεγονός πως οι υπολογιστές αποδίδουν γρηγορότερα και καλύτερα πολλές ενέργειες οι οποίες χειροκίνητα θα απαιτούσαν αρκετό χρόνο για να διεκπεραιωθούν. Παρόλη την εξέλιξη της τεχνολογίας ο αριθμός των μη-αυτοποιημένων συστημάτων, σε εργασίες που θα ήταν πολύ αποτελεσματικό να υπάρχει αυτοματοποιημένο σύστημα, παραμένει υψηλός.

Ο στόχος της εργασίας αυτής είναι η αυτοματοποίηση της διαχείρισης περιουσιακών στοιχείων κάποιου οργανισμού. Το σύστημα έχει σχεδιαστεί με σκοπό να παρέχει μεγάλη ελευθερία στον χρήστη, καθώς και ευκολία, ώστε να ολοκληρώνει εύκολα τις ενέργειες που χρειάζεται να κάνει. Το σύστημα χαρακτηρίζεται από τα εξής στοιχεία:

- Προσθήκη/Επεξεργασία/Αφαίρεση αντικειμένων στην βάση.
- Δυνατότητα αλλαγής δομής της βάσης δεδομένων των αντικειμένων.
- Εκτύπωση ετικετών για εύκολη και γρήγορη πρόσβαση στις πληροφορίες του αντικειμένου.
- Δυνατότητα παραχώρησης αντικειμένων σε τρίτα άτομα.
- Δυνατότητα διαχείρισης χρηστών και ομάδων.

## **Abstract**

The technology and automated systems is now a common thing in our society. It's a fact that computers can complete various tasks, faster and better than a human who would need much time in order to complete them. Despite the technological development, the number of the non-automated systems, in jobs that would be wise to have one, remains high.

The goal of this project is the automation of a property management system of an organization. The system has been developed in a way so it can provide much freedom to the user and also an easy to operate system so the user can complete his tasks easily. The system's qualities are the following:

- Adding/Editing/Deleting items in the database.
- Capability of changing the item's database.
- Printing QR labels for easy and fast access to the item's information.
- Capability of lending an item to a third person.
- Capability of managing users and groups.

## Εισαγωγή

Η διαχείριση των περιουσιακών στοιχείων αποτελεί πρόκληση για πολλούς οργανισμούς, καθώς αυτή απαιτεί αρκετή οργάνωση και επιμέλεια από το ανθρώπινο δυναμικό. Είναι επίσης μια εργασία που χρειάζεται να γίνει αρκετές φορές τον χρόνο, ώστε να υπάρχει συνέπεια. Αυτή η δουλειά συνήθως γίνεται χειροκίνητα και όχι με κάποιον αυτοματοποιημένο τρόπο και αυτό συνήθως επιφέρει λάθη στην καταγραφή των αντικειμένων που μπορεί να οδηγήσουν αρνητικές συνέπειες για τον οργανισμό αυτό.

Η εποχή μας χαρακτηρίζεται από την ραγδαία ανάπτυξη της τεχνολογίας. Η επεξεργαστική δύναμη των υπολογιστών βελτιώνεται συνέχεια και οι υπολογιστές πλέον είναι ικανοί να λύσουν προβλήματα που πριν μερικά χρόνια θεωρούταν αδύνατο. Έτσι, τα αυτοματοποιημένα συστήματα γίνονται όλο και πιο γνωστά και χρησιμοποιούνται σε περισσότερους τομείς. Οπότε, σκοπός της εργασίας αυτή είναι να σχεδιαστεί ένα σύστημα το οποίο να διαχειρίζεται τα περιουσιακά στοιχεία ενός οργανισμού και να παρέχει τρόπους εύκολους, γρήγορους και κατανοητούς προς τον χρήστη να ολοκληρώσει το έργο του.

Η εργασία αφορά τη διαχείριση των περιουσιακών στοιχείων μιας εταιρίας/οργανισμού. Πολλά πράγματα στην ζωή μας τείνουν να γίνονται με ηλεκτρονικό τρόπο γιατί έτσι επιταχύνονται οι διαδικασίες και ολοκληρώνονται με πιο άνετο και εύκολο τρόπο. Έτσι, το σύστημα αυτό έχει ως σκοπό την ευκολότερη κατανομή του μητρώου υλικών που όχι μόνο υλοποιεί τις ανάγκες κάποιου οργανισμού για διαχείριση υλικών, αλλά είναι ευέλικτη και αλλάζει ανάλογα με τις ρυθμίσεις που θα ορίσουν οι διαχειριστές. Αυτό φυσικά αποτελεί πρόκληση για κάποιον προγραμματιστή, καθώς κάθε ευκολία που παρέχεται στον χρήστη χρειάζεται αρκετές γραμμές κώδικα που να το υποστηρίζουν.

Οι απαιτήσεις σε λογισμικό για την διεκπεραίωση της εργασίας είναι ελάχιστες καθώς δεν καταναλώνει περισσότερα στοιχεία από έναν κανονικό ιστοχώρο. Συγκεκριμένα χρειάζεται έναν **web server** ο οποίος να έχει βάση δεδομένων **MySQL** που να δέχεται **PDO** εκφράσεις, και να υποστηρίζει **PHP**. Το μέγεθος των αρχείων που χρειάζεται για την εγκατάσταση δεν ξεπερνάει τα **4MB**. Βέβαια το σύστημα επιτρέπει ανέβασμα αρχείων και εικόνων έτσι είναι απαραίτητο να υπάρχει αρκετός ελεύθερος χώρος, ώστε να μπορούν να ανέβουν όσα αρχεία χρειαστεί. Επίσης υπάρχουν και απαιτήσεις όσον αφορά την εμφάνιση. Για αυτόν τον λόγο η κάθε ιστοσελίδα του συστήματος περιέχει κώδικα **XTML traditional** και ενσωματώνει **CSS** κώδικα, ώστε το περιεχόμενο της να πιο εμφανίσιμο.

Η ασφάλεια σε ένα τέτοιου είδους σύστημα είναι σαφώς σημαντική, για αυτό γίνονται έλεγχοι σε πολλά σημεία του κώδικα. Το σύστημα ελέγχει τα δικαιώματα των χρηστών και επίσης για κάθε σύνδεση με την βάση MySQL χρησιμοποιώντας **PDO** εκφράσεις που παρέχουν ασφάλεια από πολλών ειδών επιθέσεις όπως την διάσημη **sql injection**. Εκτός από ασφάλεια από εξωτερικούς παράγοντες και επιθέσεις, ελέγχεται και η ασφάλεια λανθασμένων επιλογών. Σε περίπτωση που κάποιος χρήστης πάει να κάνει κατά λάθος μια ενέργεια που κανονικά θα επέφερε σοβαρές επιπτώσεις στην βάση, το σύστημα τον αποτρέπει από το να ολοκληρώσει αυτήν την ενέργεια και του δίνει συμβουλές, ώστε τέτοιο λάθος να μην επαναληφθεί. Γενικότερα έχουν τοποθετηθεί σε πολλά σημεία του ιστοχώρου πολλά βοηθητικά μηνύματα, ώστε κανένας χρήστης να μην βρεθεί σε μια θέση που να μην ξέρει τι να κάνει.

Όπως αναφέρθηκε και προηγουμένως, όλες οι ευκολίες που παρέχονται από το σύστημα μεταφράζονται σε ανάγκη συγγραφής πολλών γραμμών κώδικα. Συγκεκριμένα στον πίνακα 1 αναλύονται τα αρχεία όσον αφορά τις γραμμές κώδικα και τους χαρακτήρες που έχουν. Συγκεκριμένα υπάρχουν 518 αρχεία και 23 φάκελοι και μόνο τα κύρια αρχεία φτάνουν τις 6449 γραμμές και ξεπερνάνε κατά πολύ τους 200.000 χαρακτήρες.

Αρχείο	Γραμμές κώδικα	Χαρακτήρες
addit.php	589	18208
chpas.php	49	1671
index.php	102	2944
main.php	19	544
manage.php	1883	62468
pdf.php	243	6364
permissions.php	353	10877
qrcode.php	249	7658
search.php	1111	33614
showconfs.php	39	1436
showsearch.php	175	4851
showsubs.php	30	934
usercp.php	88	3208
users.php	724	25384
watchers.php	862	33844
watchershelper.php	133	4648
Σύνολο	6649	218653

Πίνακας 1:Ανάλυση περιεχομένου κώδικα



Το σύστημα διαχείρισης περιουσιακών στοιχείων αποτελείται από πολλά μέρη. Αρχικά, οι διαχειριστές δημιουργούν κατηγορίες και υποκατηγορίες, ανάλογα με τα αντικείμενα που επιθυμούν να καταχωρηθούν στην βάση. Έπειτα προσθέτουν τις πληροφορίες που θα έχουν τα αντικείμενα ανάλογα με την κατηγορία ή την υποκατηγορία τους. Μπορούν να τοποθετήσουν κοινές μεταβλητές οι οποίες εφαρμόζονται είτε σε όλες τις υποκατηγορίες ή συγκεκριμένα στις υποκατηγορίες της κατηγορίας που θα επιλέξει ο διαχειριστής. Στην συνέχεια καθώς προστίθενται αντικείμενα, μπορούν να εκτυπώνονται ειδικές **QR ετικέτες** (1) οι οποίες θα επικολλούνται στο αντικείμενο. Οι ετικέτες αυτές περιέχουν κάποιες πληροφορίες που επιθυμεί ο χρήστης, τον σύνδεσμο για να δει κάποιος πληροφορίες σχετικά με το αντικείμενο και φυσικά την **QR ετικέτα** η οποία μπορεί να σαρωθεί από κάποιο κινητό με **QR reader**, ώστε ο χρήστης να μεταβεί απευθείας στην σελίδα όπου αναλύονται οι ιδιότητες του αντικειμένου.

Υπάρχουν επίσης πολλές δυνατότητες που βοηθούν στην καλύτερη διαχείριση του συστήματος. Οι διαχειριστές μπορούν να δημιουργήσουν χρήστες και ομάδες και να θέσουν τα δικαιώματα που επιθυμούν, ώστε να επιτρέπουν ορισμένες ενέργειες σε αυτούς. Μόνο οι διαχειριστές μπορούν να δημιουργήσουν νέους χρήστες και όταν μεταβούν σε μια τέτοια ενέργεια στέλνεται αυτομάτως ένα e-mail στον νέο χρήστη με τον προσωρινό κωδικό του. Ο χρήστης από την στιγμή που θα προσπαθήσει να μπει στο σύστημα θα αναγκαστεί από αυτό να αλλάξει υποχρεωτικά τον κωδικό του καθώς αυτός που έχει αρχικά δεν θεωρείται μεγάλης ασφάλειας καθώς είναι μια σειρά αριθμών και παράγεται τυχαία. Τέλος υπάρχει η δυνατότητα παραχώρησης αντικειμένων σε τρίτα άτομα. Το σύστημα καταγράφει όλες τις παραχωρήσεις καθώς και τα στοιχεία που χρειάζονται για κάθε παραχώρηση. Για μεγαλύτερη ευκολία καταγράφονται τα στοιχεία του ατόμου παραχώρησης, ώστε σε μελλοντική παραχώρηση αντικειμένου στο ίδιο άτομο τα στοιχεία του να συμπληρώνονται αυτομάτως από το σύστημα.

Τα κεφάλαια του κειμένου αναλύουν το σύστημα με αρκετές λεπτομέρειες και σχόλια. Συγκεκριμένα στο κεφάλαιο 1 αναλύονται οι γλώσσες προγραμματισμού που χρησιμοποιούνται από το σύστημα για τις λειτουργίες του. Στην συνέχεια στο κεφάλαιο 2 γίνεται μια σύντομη περιγραφή των αρχείων και ανάλυση της δομής της βάσης δεδομένων. Επιπλέον, στο κεφάλαιο 3 παρουσιάζονται τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση της εργασίας. Στο κεφάλαιο 4 αναλύονται σημαντικά κομμάτια του κώδικα του ιστοχώρου παραθέτοντας δείγματα κώδικα και επεξηγώντας την λειτουργία τους και αναλύοντας τη λογική που οδήγησε σε αυτήν την μορφή του κώδικα. Επιπλέον στο κεφάλαιο 5 δίνονται αναλυτικές εξηγήσεις εγκατάστασης και εξηγούνται όλες οι δυνατότητες που προσφέρουν. Τέλος, στο κεφάλαιο 7, υπάρχουν τα συμπεράσματα και πιθανές επεκτάσεις του συστήματος.

## Κεφάλαιο 1: Αρχιτεκτονική

Στο κεφάλαιο αυτό αναλύεται το θεωρητικό υπόβαθρο των στοιχείων που χρησιμοποιεί το σύστημα διαχώρισης μητρώου υλικών. Γίνεται επεξήγηση των διάφορων προγραμματιστικών γλωσσών καθώς και των QR ετικετών που χρησιμοποιούνται. Επίσης φανερώνεται μέρος της λογικής της δομής του κώδικα, ώστε αυτός να είναι καλύτερα κατανοητός από τον χρήστη.

Η αρχιτεκτονική του συστήματος είναι ιδιαίτερα σημαντική καθώς αποτελεί την βάση ολόκληρου του ιστοχώρου. Για την επίτευξη αυτού του έργου χρησιμοποιούνται 5 γλώσσες προγραμματισμού που συνεργάζονται μεταξύ τους για να πετύχουν το θεμιτό αποτέλεσμα. Συγκεκριμένα αναλύονται η **HTML**, η **CSS**, η **PHP**, η **MySQL** και η **JavaScript**. Η **HTML** αποτελεί το τελικό αποτέλεσμα που εμφανίζεται στον χρήστη, μεταφραζόμενο φυσικά από τον εκάστοτε περιηγητή που χρησιμοποιεί. Η **CSS** διαμορφώνει τα στοιχεία της **HTML** δίνοντάς τους διάφορες οπτικές ιδιότητες όπως μέγεθος και χρώμα. Η **PHP** δίνει τον δυναμικό χαρακτήρα κάθε σελίδας, ουσιαστικά εξάγει κάθε φορά διαφορετικό **HTML** κώδικα ανάλογα με τις επιλογές του χρήστη και τα δεδομένα από την βάση. Η **MySQL** είναι η βάση δεδομένων που χρησιμοποιεί το σύστημα και μέσω αυτής αποθηκεύονται, ανακτώνται και διαγράφονται ό,τι πληροφορίες χρειάζονται. Η **JavaScript** δίνει την δυνατότητα να αλλάξουν κάποια στοιχεία στην σελίδα χωρίς να χρειάζεται να γίνει επαναφόρτιση της σελίδας. Ο συνδυασμός όλων αυτών των στοιχείων μας δίνει ένα σύστημα πλήρως δυναμικό και φιλικό προς τον χρήστη με πολλές δυνατότητες και ευκολίες. Τέλος, χρησιμοποιείται η τεχνολογία των **QR ετικετών**. Η τεχνολογία αυτή μας επιτρέπει γρήγορη πρόσβαση σε οποιοδήποτε μέρος του συστήματος, ώστε να βελτιστοποιείται η λειτουργικότητα του συστήματος.

Μετά το θεωρητικό υπόβαθρο που αναλύεται σε αυτό το κεφάλαιο, ακολουθεί η δομή των αρχείων και της βάσης δεδομένων του συστήματος αναλύονται στο κεφάλαιο 2 όπου υπάρχει μια περιγραφή των σημαντικότερων αρχείων του συστήματος καθώς και πλήρης επεξήγηση των πινάκων της βάσης δεδομένων.

## 1.1 HTML

Η **HTML**<sup>a</sup> (1) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων .

Η **HTML** γράφεται υπό μορφή στοιχείων **HTML** τα οποία αποτελούνται από *ετικέτες (tags)*, οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα `<html>`), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες **HTML** συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα `<h1>` και `</h1>`), με την πρώτη να ονομάζεται *ετικέτα έναρξης* και τη δεύτερη *ετικέτα λήξης* (ή σε άλλες περιπτώσεις *ετικέτα ανοίγματος* και *ετικέτα κλεισίματος* αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

Ο σκοπός ενός **web browser** είναι να διαβάζει τα έγγραφα **HTML** και τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες **HTML**, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας.

Το σύστημα διαμορφώνει κατάλληλα την **HTML** της κάθε σελίδας, ώστε ο χρήστης να την αισθάνεται οικεία. Στην διαμόρφωση της ιστοσελίδας παίζουν χαρακτηριστικό ρόλο τα **css** τα οποία θα αναλυθούν στην συνέχεια.

Συγκεκριμένα το σύστημα περιέχει σελίδες που είναι συμβατές με το πρότυπο **XHTML** που είναι αυστηρότερο από το **HTML** και συμβατό με πολλές συσκευές. Οι σελίδες όλες είναι δυναμικού τύπου και αλλάζουν ανάλογα με τις ρυθμίσεις και τα δεδομένα της εκάστοτε στιγμής. Οι σελίδες XHTML μπορούν να εμφανιστούν χωρίς σφάλματα σε σχεδόν όλες τις συσκευές αλλά απαιτούν προσεκτικά γραμμένο κώδικα. Επίσης πολλές δυνατότητες που υπάρχουν στην HTML έχουν καταργηθεί στην **XHTML** για λόγους συμβατότητας.

Για τον έλεγχο της εγκυρότητας και έλεγχο του ιστοχώρου αν πληροί τις προϋποθέσεις που απαιτεί η **XHTML**, ο έλεγχος γίνεται από το **validator** της **W3schools**.

---

<sup>a</sup> ακρωνύμιο του αγγλικού **Hyper Text Markup Language**

## 1.2 CSS

Η **CSS<sup>b</sup>** (2) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες **HTML** και **XHTML**, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστότοπου. Η **CSS** είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στιλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την **HTML**. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της **CSS** κρίνεται ως απαραίτητη.

Κάθε σελίδα περιλαμβάνει τα δικά της **CSS**, ανάλογα με αυτά που χρειάζεται να δείξει. Συγκεκριμένα υπάρχουν 4 κύρια **CSS**. Το **menu** που διαμορφώνει την μπάρα διεύθυνσης σε κάθε σελίδα, το **global** που μπαίνει αυτόματα σε κάθε σελίδα και διαμορφώνει κατάλληλα τα **headers**, το **elegant aero** το οποίο δημιουργεί αισθητικά ωραίες φόρμες και το **table** το οποίο δημιουργεί τους πίνακες της σελίδας.

Τα **CSS**, όπως και κάθε άλλο προγραμματιστικό στοιχείο του ιστοχώρου αποτέλεσαν και αυτά την δική τους πρόκληση, καθώς αυτά έπρεπε να συνδυαστούν από διάφορες πηγές, να συνεργαστούν μεταξύ τους και επίσης να αλλαχθούν για να καλύψουν τις ανάγκες που δημιουργούνταν. Έγιναν αρκετές αλλαγές, ιδικά στα χρώματα, ώστε το τελικό αποτέλεσμα να είναι φιλικό προς τον χρήστη. Εφόσον όλα παίζουν ρόλο, (χρώμα, μέγεθος, στυλ, σχήμα) μερικά από τα αρχεία **CSS** άλλαξαν τόσο πολύ που πλέον δεν θυμίζουν καθόλου την αρχική τους μορφή. Αυτό φυσικά αποδεικνύει και πόσες αλλαγές χρειάστηκαν να γίνουν και σαφώς, αυτές οι αλλαγές δεν έγιναν όλες μαζί, αλλά σταδιακά καθώς η ιστοσελίδα εξελισσόταν όλο και παραπάνω.

Γενικότερα κάθε **CSS** έχει συγκεκριμένη μορφή και αποτελεί πρωταρχικό στοιχείο στην διαμόρφωση των στοιχείων του ιστοχώρου. Στον κώδικα **XHTML** δεν αναγράφονται στοιχεία αλλαγής εμφάνισης καθώς αυτή η δουλειά γίνεται αποκλειστικά από τα αρχεία **CSS** του συστήματος.

---

<sup>b</sup> *Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ* ή αλληλουχία φύλλων στυλ

## 1.3 PHP

Η **PHP** (3) είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML. Έχει ως βάση την γλώσσα **C** και υπάρχει στις περισσότερες ιστοσελίδες. Κάθε ιστοσελίδα που περιέχει **PHP** κώδικα συνήθως χαρακτηρίζεται από την επέκταση `.php`.

Κάθε σελίδα του συστήματος περιέχει κώδικα **PHP**. Η ιστοσελίδα έχει σχεδιαστεί, ώστε ο χρήστης να έχει μεγάλη διευκόλυνση στις ενέργειες που εκτελεί. Αυτό έχει ως αποτέλεσμα πολλές σειρές κώδικα ανά σελίδα που φυσικά δεν αποτελεί πρόβλημα εφόσον οι υπολογιστές σήμερα μπορούν να εκτελέσουν πολύ περισσότερο κώδικα χωρίς πρόβλημα.

Η γλώσσα **PHP**, όπως και κάθε γλώσσα προγραμματισμού, δίνει την ελευθερία για προγραμματισμό με μοναδικό τρόπο. Κάθε άνθρωπος σκέπτεται διαφορετικά, έτσι το δυσκολότερο πράγμα σε κάθε προγραμματιστή είναι να μπορέσει να διαβάσει τον κώδικα κάποιου άλλου καθώς αυτό βγαίνει από την δικιά του ροή σκέψης. Αυτό γίνεται λόγω του ότι μία ενέργεια μπορεί να γίνει με πολλούς τρόπους οπότε 10 διαφορετικοί προγραμματιστές μπορεί να γράψουν 10 διαφορετικά κομμάτια κώδικα που να κάνουν το ίδιο ακριβώς πράγμα.

Κάτι σημαντικό στον κώδικα είναι μερικά μοτίβα κώδικα που χρησιμοποιούνται σε όλο το μέρος του κώδικα. Καθώς τα σχόλια είναι λίγα, κάποιος που θα δει τον κώδικα, από τα ονόματα των μεταβλητών και των **functions** όπως και από πολλά άλλα θα καταλάβει διάφορα πράγματα όπως την πιθανή λειτουργία τους ή και την ιστορία του site όπως και διάφορες σκέψεις από τον συγγραφέα του κώδικα. Θα παρατηρήσετε για παράδειγμα πολλές φορές να εμφανίζεται η λέξη "**dx**" σε πολλά σημεία του κώδικα καθώς είναι μέρος του ψευδωνύμου του δημιουργού (**dx4**).

## 1.4 MySQL

Η **MySQL** (4) είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Έλαβε το όνομά της από την κόρη του Μόντυ Βιντένιους, τη Μάι<sup>c</sup>. Το πρόγραμμα τρέχει έναν εξυπηρετητή<sup>d</sup> παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων. Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU<sup>e</sup>, καθώς και μέσω ορισμένων ιδιόκτητων συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, η οποία σήμερα ανήκει στην Oracle.

Οι βάσεις **MySQL** χρησιμοποιούνται πολύ συχνά λόγω της μεγάλης ευκολίας που δίνουν όσον αφορά την αποθήκευση και ανάκτηση δεδομένων. Η **MySQL** συνδέεται εύκολα με κώδικα **PHP** για τις παραπάνω ενέργειες. Στην ιστοσελίδα η σύνδεση γίνεται μόνο με **PDO** εκφράσεις, κάτι που προστατεύει την σελίδα από διάφορες επιθέσεις.

Το σύστημα διαμορφώνει κατάλληλα τις βάσεις δεδομένων και τις χτίζει ή τις αλλάζει ανάλογα με τις εντολές που δίνει ο χρήστης μέσω της ιστοσελίδας. Αυτό περιλαμβάνει αλλά δεν περιορίζεται σε δημιουργία νέων πινάκων και εμφάνιση/αλλαγή δομής κάποιου υπάρχοντος πίνακα. Ο κώδικας **PHP** πολλές φορές μπορεί να περιορίζει, για λόγους ασφάλειας και ακεραιότητας του συστήματος, την πρόσβαση σε μερικά σημεία της βάσης. Επίσης το σύστημα διαθέτει στο **configuration** αρχείο να επιλεγεί κάποιο πρόθεμα βάσεις έτσι, ώστε να μην προκληθεί πρόβλημα με τις υπάρχουσες βάσεις δεδομένων. Υποστηρίζονται μέχρι στιγμής κανονική σύνδεση και σύνδεση μέσω **socket**. Όλοι οι κωδικοί αποθηκεύονται με **hash** μορφή.

Στο συγκεκριμένο σύστημα, κρίθηκε σημαντικό ότι έπρεπε να δημιουργηθούν αρκετές βάσεις δεδομένων, όπως επίσης και να υπάρχει τρόπος να δημιουργούνται και να διαγράφονται δυναμικά, σύμφωνα με τις προτιμήσεις των διαχειριστών. Κάθε πίνακας της βάση μπορούν να τροποποιηθούν σε μεγάλο βαθμό, αν και μερικά στοιχεία και έλεγχοι της βάσεις (όπως μοναδικότητα) γίνονται μέσω του κώδικα για διάφορους λόγους. Τις βάσεις του συστήματος θα τις αναλύσουμε στο κεφάλαιο 2.

---

<sup>c</sup> αγγλ. My

<sup>d</sup> server

<sup>e</sup> General Public License

## 1.5 JavaScript

Η **JavaScript** (5) είναι διερμηνευμένη γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές. Αρχικά αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται. Η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα, είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη **C**. Η **JavaScript** αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη **Java**, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της **JavaScript** προέρχονται από τις γλώσσες προγραμματισμού **Self** και **Scheme**. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα, υποστηρίζοντας αντικειμενοστραφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων — τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή. Το πρότυπο της γλώσσας κατά τον οργανισμό τυποποίησης **ECMA** ονομάζεται **ECMAScript**.

Η γλώσσα **Java** επίσης συνδυάζεται άριστα στις λειτουργίες μιας δυναμικής ιστοσελίδας. Μπορεί ανά πάσα στιγμή, ανάλογα με τις κινήσεις που γίνονται από τον χρήστη, χωρίς να χρειάζεται να ξαναφορτωθεί η σελίδα να αλλάξει το περιεχόμενό της.

Στο σύστημα εκτός από την κανονική μορφή της **JavaScript**, χρησιμοποιείται η **Ajax** και η **jquery**. Με την χρήση όλων αυτών των μεθόδων υπάρχει αρκετή δυναμικότητα στο περιβάλλον του χρήστη και τον καθοδηγούν σε πολλά σημεία όπου μπορεί να προκληθεί κάποιο πρόβλημα.

Εκτός από την δυναμικότητα που παρέχει, η **JavaScript** χρησιμοποιείται και για αισθητικούς λόγους. Για παράδειγμα εμφανίζει κουτί για επιλογή ώρας/ημερομηνίας στα κατάλληλα πεδία, ώστε ο χρήστης να μπορεί εύκολα να δίνει τα δεδομένα χωρίς να ανησυχεί για περίπτωση εσφαλμένης μορφής.

## 1.6 Κώδικας QR

Ο **κώδικας QR** (6) είναι ένας γραμμωτός κώδικας<sup>f</sup> δύο διαστάσεων, που δημιουργήθηκε από την ιαπωνική εταιρεία Denso-Wave το 1994. Το "QR" προέρχεται από τα αρχικά των λέξεων "Quick Response" (Γρήγορη Ανταπόκριση), γιατί οι δημιουργοί του είχαν ως κύριο σκοπό τα δεδομένα, που περιέχονται στον κώδικα, να αποκωδικοποιούνται με μεγάλη ταχύτητα. Ο Κώδικας QR είναι πολύ διαδεδομένος στην Ιαπωνία, όπου αποτελεί το πιο δημοφιλές είδος κώδικα δύο διαστάσεων.

Στην Ιαπωνία, όπου η χρήση των μονοδιάστατων barcode ήταν και είναι πολύ δημοφιλής, η πληροφορία που μπορεί να αποθηκεύσει ένας μονοδιάστατος ράβδο-κώδικας κάποια στιγμή δεν ήταν αρκετή, οπότε άρχισαν να ερευνούν την δυνατότητα κωδικοποίησης πληροφορίας σε σχήματα δύο διαστάσεων. Ο κώδικας QR είναι αποτέλεσμα τέτοιας έρευνας. Αρχικά χρησιμοποιήθηκε για τον εντοπισμό ανταλλακτικών οχημάτων σε τμήματα παραγωγής. Σήμερα χρησιμοποιείται σε πολύ πιο ευρύ φάσμα, στο εμπόριο, στη διαφήμιση και στη διαχείριση εμπορευμάτων μέχρι τη διασκέδαση καθώς και για προσωπικά μηνύματα.

Η σάρωση ενός κώδικα QR, γίνεται με την βοήθεια ενός έξυπνου κινητού. Για να γίνει αυτό, ο χρήστης χρειάζεται μία εφαρμογή η οποία μετατρέπει την κάμερα του κινητού σε σαρωτή κώδικα QR.

Μέσα σε ένα κώδικα QR μπορεί να αποθηκευτεί, π.χ., ένας σύνδεσμος προς μια ιστοσελίδα. Ο χρήστης σαρώνει με το κινητό του τηλέφωνο τον κώδικα QR και πλοηγείται αυτόματα στην ιστοσελίδα. Αυτή η πράξη της σύνδεσης από το φυσικό κόσμο είναι γνωστή ως *hard-link* ή υπερσύνδεση με φυσικό κόσμο. Οι χρήστες μπορούν, επίσης, να δημιουργήσουν και να εκτυπώσουν τους δικούς τους κώδικες QR με χρήση διάφορων ελεύθερων λογισμικών παραγωγής κώδικα QR που υπάρχουν στο δίκτυο. Μέσα μπορούν να αποθηκεύσουν όποιο μήνυμα θέλουν.

Στο σύστημα διαχείρισης περιουσιακών στοιχείων, οι QR ετικέτες περιέχουν έναν σύνδεσμο προς το σύστημα με παράμετρο το μοναδικό ID του αντικειμένου. Έτσι, όταν κάποιος χρήστης σαρώσει την ετικέτα, θα μεταβεί κατευθείαν στην σελίδα που αναγράφονται αναλυτικά τα στοιχεία του αντικειμένου. Φυσικά για να δει όλα τα στοιχεία και όχι μόνο τα δημόσια, ο χρήστης πρέπει να έχει κάνει είσοδο στο σύστημα και να έχει τα κατάλληλα

---

<sup>f</sup> barcode



δικαιώματα. Οι ετικέτες αυτές είναι σχεδιασμένες ώστε να προσαρμόζονται σε αυτοκόλλητα τα οποία θα τοποθετούνται στο αντικείμενο. Περισσότερα για το πως χρησιμοποιούνται οι QR ετικέτες θα δούμε στο κεφάλαιο 4 και 5. Στην Εικόνα 1 υπάρχει ένα παράδειγμα QR ετικέτας που περιέχει το μήνυμα "hello world".



Εικόνα 1: Παράδειγμα QR ετικέτας

## **Κεφάλαιο 2: Επισκόπηση δομής**

Στο κεφάλαιο αυτό υπάρχει μια περιγραφή των σημαντικότερων αρχείων που βρίσκονται στο σύστημα καθώς και μια πλήρης ανάλυση της δομής της βάσης δεδομένων.

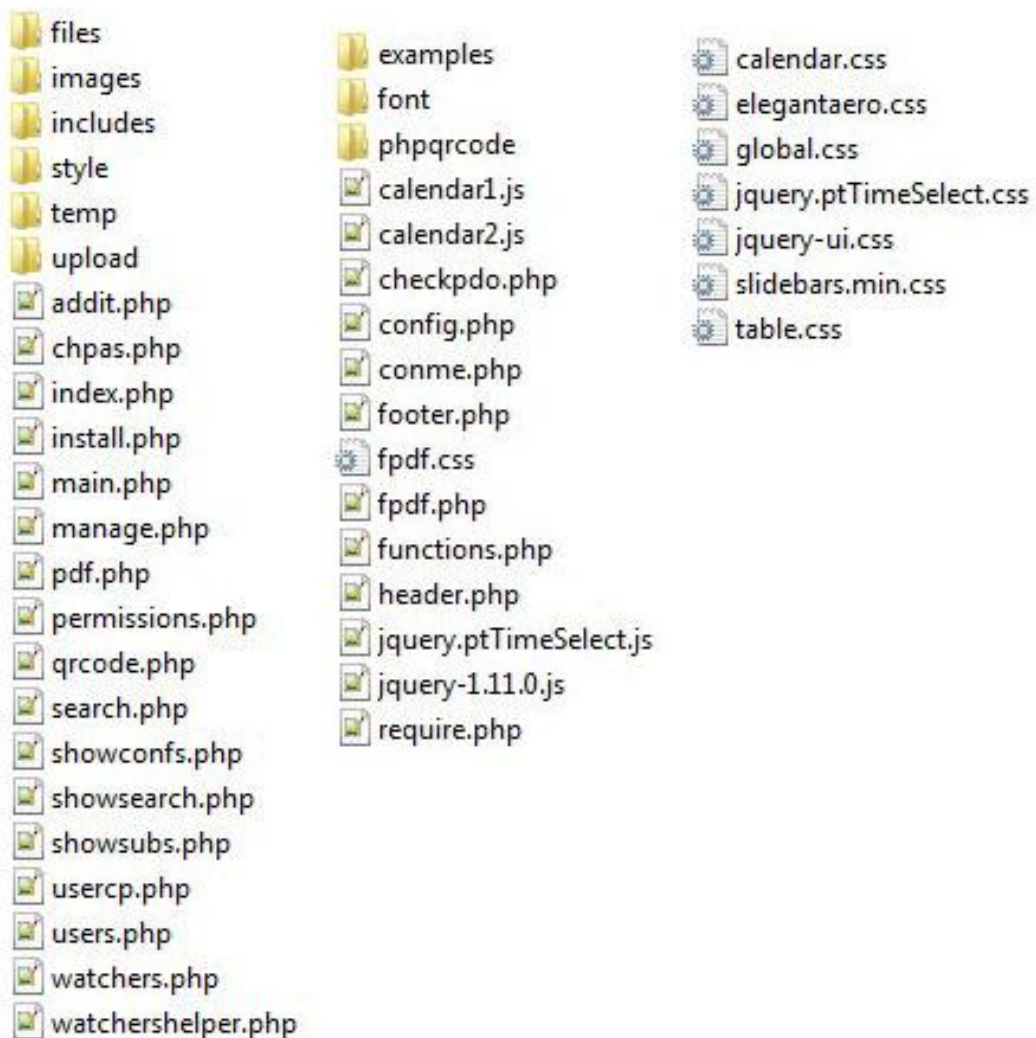
Το σύστημα αρχείων χωρίζεται σε 4 κυρίως μέρη: Τον κύριο κατάλογο που περιέχει τα κύρια αρχεία του συστήματος και τα μοναδικά που επισκέπτεται ο χρήστης, τον φάκελο "includes" που περιέχει όλα τα αρχεία που περιέχουν διάφορες λειτουργίες και ενσωματώνονται στις κύριες ιστοσελίδες του ιστοχώρου, τον φάκελο "style" που περιέχει όλα τα αρχεία CSS που ορίζουν το στυλ κάθε σελίδας, και τους λοιπούς φακέλους όπου περιέχουν τα δεδομένα των χρηστών ή πρόχειρα δεδομένα του συστήματος. Ο κώδικας των αρχείων αναλύεται στο κεφάλαιο 4.

Η βάση δεδομένων είναι τύπου MySQL και αναλύονται οι πίνακες με την σειρά που δημιουργούνται κατά την εγκατάσταση (περισσότερες πληροφορίες σχετικά με την εγκατάσταση υπάρχουν στο κεφάλαιο 5). Επίσης δίνονται λεπτομέρειες και στους πίνακες που δημιουργούνται από τους χρήστες μέσω καθοδήγησης φυσικά από το σύστημα. Συγκεκριμένα αναλύονται οι πίνακες καθώς και όλα τα στοιχεία που έχουν. Είναι σημαντικό να αναφερθεί πως οι περισσότεροι πίνακες δεν έχουν σταθερή μορφή καθώς αλλάζουν ανάλογα με τις απαιτήσεις των χρηστών.

Η δημιουργία, τοποθέτηση, επεξεργασία και λοιπές ενέργειες που χρειάστηκαν για να δημιουργηθούν όλα τα αρχεία, για να δημιουργηθεί η βάση δεδομένων και γενικότερα το σύστημα, απαιτούν ορισμένα εργαλεία. Τα εργαλεία που χρησιμοποιήθηκαν για την διεκπεραίωση της εργασίας αυτής αναλύονται στο κεφάλαιο 3.

## 2.1 Αρχεία

Για να καλυφθούν οι ανάγκες του συστήματος διαχείρισης μητρώου υλικών δημιουργήθηκαν τα αρχεία που φαίνονται στην Εικόνα 2 (αναλύονται οι φάκελοι **includes** και **style**):



Εικόνα 2: Αρχεία συστήματος διαχείρισης μητρώου υλικών

Για καλύτερη κατανομή των αρχείων κάθε λειτουργιά έχει το δικό της αρχείο PHP και μερικές συνδυάζονται με κάποιο βοηθητικό αρχείο. Επίσης, για να υπάρχει οργάνωση των αρχείων, δημιουργήθηκαν διάφοροι φάκελοι για τα αρχεία των χρηστών και φάκελοι που περιέχουν αρχεία του συστήματος. Έτσι, η δομή των αρχείων είναι η εξής:

- **files:** Ο φάκελος **files** περιέχει τα αρχεία που ανεβαίνουν από τον χρήστη. Τα αρχεία αυτά είναι τύπου **zip** και έχουν μοναδικό όνομα σύμφωνα με το είδος τους και την ώρα που δημιουργούνται.
- **images:** Ο φάκελος **images** περιέχει τις κυρίως εικόνες που χρησιμοποιούνται στην ιστοσελίδα.
- **includes:** Ο φάκελος **includes** περιέχει σημαντικά αρχεία που περιλαμβάνονται σε όλες τις σελίδες.
- **style:** Ο φάκελος **style** περιέχει τα σημαντικά **css** που εκχωρούνται στην ιστοσελίδα.
- **temp:** Ο φάκελος **temp** περιέχει τα προσωρινά αρχεία εικόνων **QR ετικετών** που δημιουργούνται όταν είναι να εκτυπωθούν οι ετικέτες.
- **upload:** Σε αυτόν τον φάκελο ανεβαίνουν οι εικόνες των χρηστών.
- **addit.php:** Το αρχείο το οποίο αφορά την προσθήκη αντικειμένου στην βάση δεδομένων
- **chpas.php:** Η σελίδα αυτή αφορά τους χρήστες που πρέπει υποχρεωτικά να αλλάξουν τον κωδικό τους.
- **index.php:** Η αρχική σελίδα που περιέχει μόνο το **login**. Αν ο χρήστης έχει ήδη κάνει είσοδο, τον στέλνει στην κατάλληλη σελίδα.
- **install.php:** Το αρχείο αυτό πρέπει να τρέξει **μόνο μία φορά**, κατά την εγκατάσταση, και έκτοτε να διαγραφεί ή να γίνει απροσπέλαστο.
- **main.php:** Είναι η κύρια σελίδα του συστήματος.
- **manage.php:** Εδώ ο χρήστης ορίζει την δομή που θα έχει το σύστημα, όπως είναι οι χρήστες και οι βάσεις δεδομένων.
- **pdf.php:** Χρησιμεύει στην εκτύπωση **pdf** των **QR code**.
- **permissions.php:** Εδώ ορίζονται οι επιτρεπτές πράξεις που μπορεί να κάνει κάποιος χρήστης.

- **qrcode.php**: Σε αυτή την σελίδα ο χρήστης θέτει τις παραμέτρους από τα αυτοκόλλητα, ώστε να εκτυπωθούν στα κατάλληλα σημεία της σελίδας.
- **search.php**: Εδώ γίνεται η αναζήτηση κάποιου αντικειμένου. Με την κατάλληλη παράμετρο εμφανίζει μόνο ένα αντικείμενο.
- **showconfs.php**: Δέχεται τις κατάλληλες παραμέτρους, ώστε να εμφανίσει τις παραμέτρους που χρειάζονται για την εκτύπωση των **QR ετικετών**.
- **showsearch.php**: Βοηθάει στην αναζήτηση φανερώνοντας τα κατάλληλα πεδία, ανάλογα με τις επιλογές που έχει κάνει ο χρήστης.
- **showsubs.php**: Φανερώνει όλες τις υποκατηγορίες της κατηγορίας που θα επιλέξει ο χρήστης σε μία λίστα.
- **usercp.php**: Δίνει την δυνατότητα στον χρήστη να αλλάξει τα στοιχεία του.
- **users.php**: Οι **admin** μπορούν να αλλάξουν/δουν τα στοιχεία κάποιου χρήστη.
- **watchers.php**: Εδώ γίνεται η διαχείριση των παραχωρήσεων των αντικειμένων προς τρίτα άτομα.
- **watchershelper.php**: Υλοποιεί διάφορες λειτουργίες **ajax** που ολοκληρώνει τις λειτουργίες των παραχωρήσεων των αντικειμένων.

Στην συνέχεια, στον φάκελο **includes**:

- **examples**: Στον φάκελο αυτό βρίσκονται ορισμένα παραδείγματα κώδικα που χρησιμεύουν στην καλύτερη κατανόηση του κώδικα που χρησιμοποιεί το σύστημα.
- **font**: Σε αυτόν το φάκελο υπάρχουν κάποιες γραμματοσειρές που χρησιμοποιείται για την δημιουργία του pdf.
- **phpqrcode**: Εδώ βρίσκονται τα αρχεία που χρειάζεται για να παραχθούν οι εικόνες του **QR code**. (7)
- **calendar1.js, calendar2.js**: Τα αρχεία αυτά χρειάζονται για την λειτουργία του ημερολογίου που βοηθάει στην κατάλληλη επιλογή στα πεδία ημερομηνίας.
- **checkpdo.php**: Το αρχείο αυτό ελέγχει κατά πόσο ο server υποστηρίζει PDO εκφράσεις, καθώς χρησιμοποιούνται ευρέως.
- **config.php**: Το αρχείο αυτό περιέχει τις ρυθμίσεις του **server**. Στο κεφάλαιο X θα αναλύσουμε πως πρέπει να αλλαχθούν για την εγκατάσταση.
- **conme.php**: Χρησιμεύει για την σύνδεση στην βάση δεδομένων. Το αρχείο αυτό παίρνει τα δεδομένα από το **config.php**, οπότε καμία αλλαγή δεν χρειάζεται σε περίπτωση αλλαγής κάποιου στοιχείου της βάσης δεδομένων.
- **footer.php**: Δείχνει το **footer** κάθε σελίδας και το ανάλογο helpline.
- **fpdf.css, fpdf.php**: Χρησιμεύει στην δημιουργία **pdf**. (8)
- **functions.php**: Περιέχει τα περισσότερα **functions**.
- **header.php**: Δημιουργεί το **header** κάθε σελίδας με τις κατάλληλες παραμέτρους.
- **jquery.ptTimeSelect.js**: Το αρχείο αυτό περιέχει τον **javascript jquery** κώδικα το οποίο έχει ως λειτουργία την εμφάνιση του πίνακα που βοηθάει στην επιλογή ώρας στα κατάλληλα πεδία
- **jquery-1.11.0.js**: Το αρχείο αυτό περιέχει τον κώδικα που χρειάζεται το σύστημα για να χρησιμοποιήσει τις δυνατότητες της **jquery**
- **require.php**: Καλεί όλα τα απαιτούμενα αρχεία

Τέλος, στον φάκελο **styles**:

- **calendar.css**: Το αρχείο αυτό περιέχει τον **CSS** κώδικα που ορίζει το στυλ στο ημερολόγιο που εμφανίζεται για την επιλογή της ημερομηνίας
- **elegantaero**: Εδώ βρίσκεται ο κώδικας **CSS** που ρυθμίζει τις οπτικές ιδιότητες των φορμών που χρησιμοποιούνται στο σύστημα
- **global.css**: Το αρχείο αυτό περιέχεται σε κάθε σελίδα καθώς περιέχει πληροφορίες για την οπτική βελτιστοποίηση πολλών **header** όπως και κάποιων άλλων στοιχείων του συστήματος
- **jquery.ptTimeSelect.css**, **jquery-ui**: Τα 2 αυτά αρχεία ρυθμίζουν το στυλ των λειτουργιών της **jquery**
- **slidebars.min.css**: Εδώ ορίζεται η μορφή του δυναμικού μενού του συστήματος (9)
- **table.css**: Σε αυτό το αρχείο υπάρχει ο κώδικας **CSS** που ορίζει το στυλ των πινάκων που χρησιμοποιούνται στο σύστημα

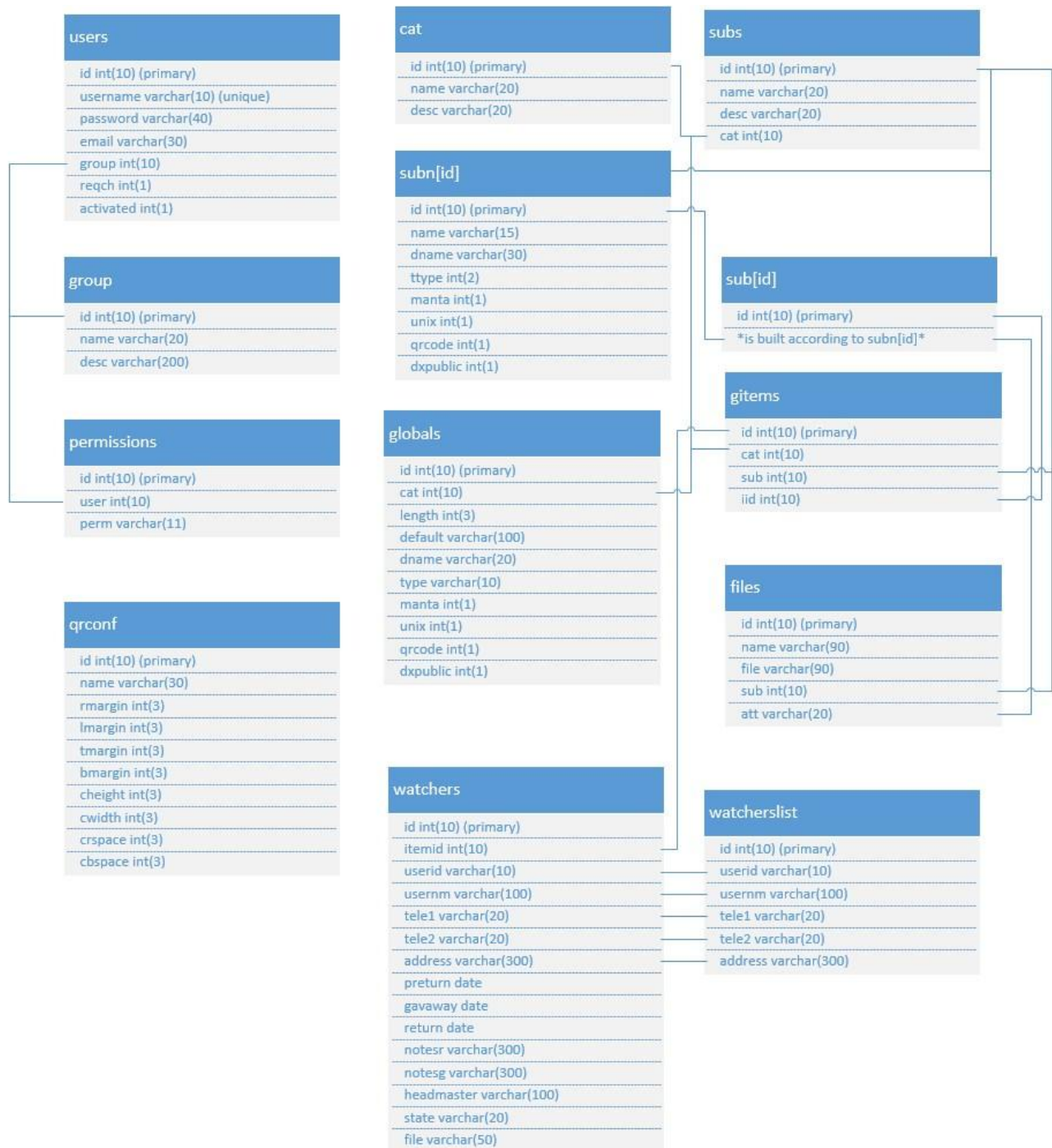
## 2.2 Σχεδιασμός Βάσης Δεδομένων

Η απαιτήσεις του συστήματος όσον αφορά την αποθήκευση των δεδομένων στην βάση είναι οι εξής:

- Στο σύστημα θα υπάρχουν χρήστες οι οποίοι θα συνδέονται με κάποιο όνομα χρήστη και κωδικό. Επίσης κάθε χρήστης θα πρέπει να έχει διάφορες άλλες πληροφορίες όπως e-mail.
- Το σύστημα θα πρέπει να κατατάσσει τους χρήστες σε ομάδες, ώστε να έχουν τα κατάλληλα δικαιώματα και να υπάρχει κατηγοριοποίηση χρηστών. Επιπλέον θα πρέπει να υπάρχει σύνδεση με τον πίνακα των χρηστών.
- Πρέπει επίσης να υπάρχει πίνακας που να περιέχει τα δικαιώματα κάθε ομάδας. Βεβαίως, αυτό απαιτεί σύνδεση με τον πίνακα των ομάδων.
- Φυσικά θα πρέπει να υπάρχουν δύο πίνακες που να περιέχουν τις κατηγορίες και τις υποκατηγορίες αντίστοιχα. Οι υποκατηγορίες, εκτός από όνομα και περιγραφή, οφείλουν να έχουν και ένα πεδίο που να δηλώνει την κατηγορία στην οποία ανήκει.
- Για κάθε υποκατηγορία, πρέπει να αντιστοιχηθεί ένας πίνακας που να περιέχει τα στοιχεία των αντικειμένων. Επειδή το σύστημα είναι δυναμικό οι ιδιότητες του πίνακα αυτού πρέπει να καθορίζονται από ξεχωριστό πίνακα.
- Το σύστημα διαθέτει κοινές μεταβλητές οι οποίες περιέχουν όλες τις πληροφορίες που χρειάζεται για να δημιουργήσουν το καινούργιο στοιχείο που θα δημιουργηθεί σε περίπτωση νέας υποκατηγορίας, όπως επίσης να ορίζεται σε ποια κατηγορία θα εφαρμόζεται.
- Λόγω του ότι τα αντικείμενα υπάρχουν σε διαφορετικούς πίνακες, πρέπει να υπάρχει ένας πίνακας ο οποίος να δίνει μοναδικό ID στα αντικείμενα, καθώς και την προέλευση του αντικειμένου.
- Επίσης χρειάζεται ένας πίνακας που να περιέχει την λίστα των αρχείων που ανεβαίνουν στον server, συνδέοντας το όνομα του αρχείου με το όνομα που επιθυμεί ο χρήστης καθώς και να περιέχει πληροφορίες που να εξασφαλίζουν από που προέρχεται.
- Όλες οι παραχωρήσεις αντικειμένων πρέπει να καταγράφονται στην βάση οπότε χρειάζεται ένας πίνακας που να περιέχει όλες τις παραχωρήσεις αντικειμένων.
- Δίνεται η δυνατότητα αποθήκευσης ρυθμίσεων των QR ετικετών όπως και των ατόμων που τους έχει παραχωρηθεί αντικείμενο. Για τον λόγο αυτό χρειάζεται να δημιουργηθεί κατάλληλος πίνακας για την αποθήκευση των πληροφοριών αυτών.



Μετά από την μελέτη των απαιτήσεων προσαρμόστηκε κατάλληλα το αρχείο install.php όπου δημιουργεί τις αρχικές βάσεις που χρειάζονται για να λειτουργήσει το σύστημα. Η ανάλυση της βάσης δεδομένων αναλύεται με την σειρά που δημιουργούνται οι πίνακες από το αρχείο αυτό. Αναλυτικό διάγραμμα υπάρχει στην Εικόνα 3.



Εικόνα 3: Διάγραμμα βάσης δεδομένων

Αρχικά δημιουργείται ένας πίνακας `users` όπου θα βρίσκονται όλοι οι χρήστες με τα στοιχεία τους. Στην εγκατάσταση δημιουργείται ο πρώτος χρήστης (**root**) ο οποίος έχει τον κωδικό που εμφανίζεται στην οθόνη. Το πεδίο **reqch** (require change) αναγκάζει τον χρήστη να αλλάξει κωδικό ενώ αν κάποιος αλλάξει το **activated** ο χρήστης δεν θα έχει πρόσβαση στον ιστοχώρο. Αναλυτικά ο πίνακας αποτελείται από τις εξής στήλες:

- `id`: μοναδικός αριθμός που χαρακτηρίζει κάθε γραμμή
- `username`: Το όνομα χρήστη
- `password`: Ο κωδικός χρήστη ο οποίος είναι κρυπτογραφημένος<sup>6</sup>
- `e-mail`: Το e-mail που έχει ορίσει ο χρήστης
- `group`: Το id της ομάδας που ανήκει ο χρήστης
- `reqch`: Σημαία που δηλώνει αν ο χρήστης πρέπει να αλλάξει κωδικό
- `activated`: Σημαία που δηλώνει αν ο χρήστης έχει πρόσβαση στο σύστημα

Στην συνέχεια, ο πίνακας `permissions` συνδέει σε κάθε στήλη το **id** του γκρουπ χρηστών (**user**) με το κατάλληλο `permission` (**perm**). Τα επιτρεπτά `permissions` είναι τα εξής:

- **-1** : Το γκρουπ γίνεται **admin γκρουπ**
- **aitems** : (Add items) Προσθήκη αντικειμένων
- **eitems** : (Edit/Delete items) Επεξεργασία/Διαγραφή αντικειμένων
- **sitems** : (Search/View items) Αναζήτηση/Προβολή αντικειμένων
- **users** : Αλλαγή στοιχείων χρήστη.

Επίσης δημιουργούνται οι πίνακες `cat` και `subs` οι οποίοι έχουν παρόμοια δομή. Ο πίνακας `cat` αποτελείται από το μοναδικό `id` του, το `name` που είναι το όνομά του και το `desc` που είναι η περιγραφή του. Ο πίνακας `sub` αναλυτικά αποτελείται από τα εξής:

- `id`: Κατά την δημιουργία κάποιας υποκατηγορίας δημιουργούνται δύο πίνακες οι οποίοι είναι οι `sub[id]` και `subn[id]`, οπότε το `id` αποτελεί σύνδεση με τους πίνακες αυτούς που θα αναλύσουμε στην συνέχεια
- `name`: Το όνομα της υποκατηγορίας
- `desc`: Η περιγραφή της υποκατηγορίας
- `cat`: Το αντίστοιχο `id` της κατηγορίας στην οποία ανήκει

---

<sup>6</sup> Πρώτα μέσω `md5` και μετά μέσω `sha1`

Έπειτα δημιουργείται ο πίνακας `group` ο οποίος περιέχει τα δεδομένα των ομάδων του συστήματος. Ο πίνακας `group` αποτελείται από το μοναδικό `id`, το οποίο συνδέεται με την στήλη `group` των χρηστών, το `name` το οποίο είναι το όνομα της βάσης, και το `desc` που είναι η περιγραφή της ομάδας.

Στην συνέχεια δημιουργείται ο πίνακας `globals` ο οποίος περιέχει όλες τις κοινές μεταβλητές μαζί με τα χαρακτηριστικά τους. Συγκεκριμένα, τα στοιχεία του πίνακα είναι τα εξής:

- `id`: Το `id` καθορίζει το όνομα του στοιχείου του πίνακα της υποκατηγορίας που θα δημιουργηθεί (`sub[id]`). Με αυτόν τον τρόπο εκτός από την σύνδεση εξασφαλίζεται και η μοναδικότητα κάθε στοιχείου των πινάκων.
- `cat`: Η κατηγορία στην οποία θα εφαρμοστεί το κοινό στοιχείο. Αν έχει την τιμή "-1" τότε σημαίνει ότι εφαρμόζεται σε όλες τις κατηγορίες
- `length`: Το μήκος του πεδίου
- `default`: Η προεπιλεγμένη τιμή του πεδίου
- `dname`: Το όνομα του πεδίου
- `type`: Ο τύπος του πεδίου
- `mandatory`: Δηλώνει αν το πεδίο θα είναι υποχρεωτικό
- `unix`: Δηλώνει αν το πεδίο θα είναι μοναδικό
- `qrcode`: Δηλώνει αν το πεδίο θα εκτυπώνεται σε καρτέλα
- `dxpublic`: Δηλώνει αν το πεδίο θα είναι δημόσιο

Ο πίνακας `gitems` που δημιουργείται στο επόμενο βήμα ο οποίος συνδέει όλα τα αντικείμενα του συστήματος παρέχοντάς τους ένα μοναδικό `id` στο καθένα, συγκεκριμένα τα στοιχεία του έχουν ως εξής:

- `id`: Το μοναδικό `id` κάθε αντικειμένου. Αυτό μπαίνει σαν παράμετρο στην σελίδα `search.php` που αναλύουμε παρακάτω
- `cat`: Το `id` της κατηγορίας που ανήκει το αντικείμενο
- `sub`: Το `id` της υποκατηγορίας που ανήκει το αντικείμενο
- `iid`: Το `id` που έχει το αντικείμενο στον πίνακα της υποκατηγορίας

Έπειτα δημιουργείται ο πίνακας `qrconf` ο οποίος έχει αποθηκευμένα όλα τα χωροθετικά στοιχεία που έχει αποθηκεύσει ο χρήστης. Αποτελείται από το `id` που χαρακτηρίζει κάθε γραμμή μοναδική, το `name` που είναι το όνομα των ρυθμίσεων που έχει αποθηκεύσει ο χρήστης και εν συνέχεια ακολουθούν όλα τα νούμερα που έχει αποθηκεύσει ο χρήστης.

Στην συνέχεια δημιουργείται ο πίνακας files που περιέχει τον κατάλογο με όλα τα αρχεία τα οποία έχει ανεβάσει ο χρήστης στο σύστημα. Συγκεκριμένα ο πίνακας αυτός αποτελείται από τα εξής στοιχεία:

- id: Μοναδικός αριθμός που χαρακτηρίζει κάθε καταχώρηση
- name: Το όνομα που έχει δώσει ο χρήστης στο αρχείο
- file: Το όνομα του αρχείου έτσι όπως έχει καταχωρηθεί στον server
- sub: Η υποκατηγορία του αντικειμένου το οποίο έχει το αρχείο αυτό
- att: Το όνομα του στοιχείου του πίνακα που έχει το αρχείο αυτό

Επίσης δημιουργείται ο πίνακας watchers ο οποίος περιέχει όλες τις καταχωρήσεις τα στοιχεία που δόθηκαν σε κάθε καταχώρηση. Αναλυτικότητα ο πίνακας αυτός έχει ως εξής:

- id: Μοναδικός αριθμός που χαρακτηρίζει κάθε καταχώρηση
- itemid: Το μοναδικό id<sup>h</sup> του αντικειμένου από την βάση δεδομένων
- userid: Η ταυτότητα του ατόμου στον οποίο γίνεται η παραχώρηση
- usernm: Το ονοματεπώνυμο του
- tele1: Το πρώτο τηλέφωνο του
- tele2: Το δεύτερο τηλέφωνο του
- address: Η διεύθυνσή του
- preturn: Εκτιμώμενη ημερομηνία επιστροφής
- gaway: Ημερομηνία παραχώρησης
- return: Ημερομηνία επιστροφής
- notesr: Σημειώσεις επιστροφής
- notesg: Σημειώσεις παραχώρησης
- headmaster: Υπεύθυνος παραχώρησης
- state: Κατάσταση αντικειμένου
- file: Κάποιο αρχείο zip

Για τον πίνακα watchers έχει δημιουργηθεί και ο βοηθητικός πίνακας watcherslist ο οποίος περιέχει τις πληροφορίες των ατόμων που έχει γίνει κάποια παραχώρηση. Έχει τα κοινά στοιχεία με τον πίνακα watchers που αφορούν το άτομο παραχώρησης, εκτός φυσικά από το id που είναι μοναδικό για κάθε καταχώρηση.

---

<sup>h</sup> από τον πίνακα gitems

Οι πίνακες που δεν δημιουργούνται κατά την εγκατάσταση αλλά δυναμικά, είναι οι sub[id] και subn[id]. Ο πίνακας subn[id] περιγράφει την δομή του πίνακα sub[id]. Η μορφή του πίνακα subn[id] έχει ως εξής:

- id: Μοναδικός αριθμός που χαρακτηρίζει κάθε καταχώρηση
- name: Το όνομα του στοιχείου του πίνακα sub[id]
- dname: Το όνομα που έχει δώσει ο χρήστης για το πεδίο αυτό
- ttype: Ο τύπος του πεδίου
- manpa: Δηλώνει αν το πεδίο είναι υποχρεωτικό
- unix: Δηλώνει αν το πεδίο είναι μοναδικό
- qrcode: Δηλώνει αν το πεδίο είναι προς εκτύπωση στην QR ετικέτα
- dxpublic: Δηλώνει αν το πεδίο είναι δημόσιο

Ο πίνακας sub[id] αποτελείται από:

- id: Που χαρακτηρίζει μοναδικά κάθε αντικείμενο σε αυτήν την υποκατηγορία
- Όλα τα στοιχεία που υπάρχουν στον πίνακα subn[id] στην θέση name και οι ιδιότητές τους ορίζονται από τα υπόλοιπα στοιχεία του πίνακα αυτού. Επίσης κατατάσσονται πρώτα τα κοινά στοιχεία και έπειτα αυτά που δημιουργήθηκαν χειροκίνητα

## Κεφάλαιο 3: Εργαλεία

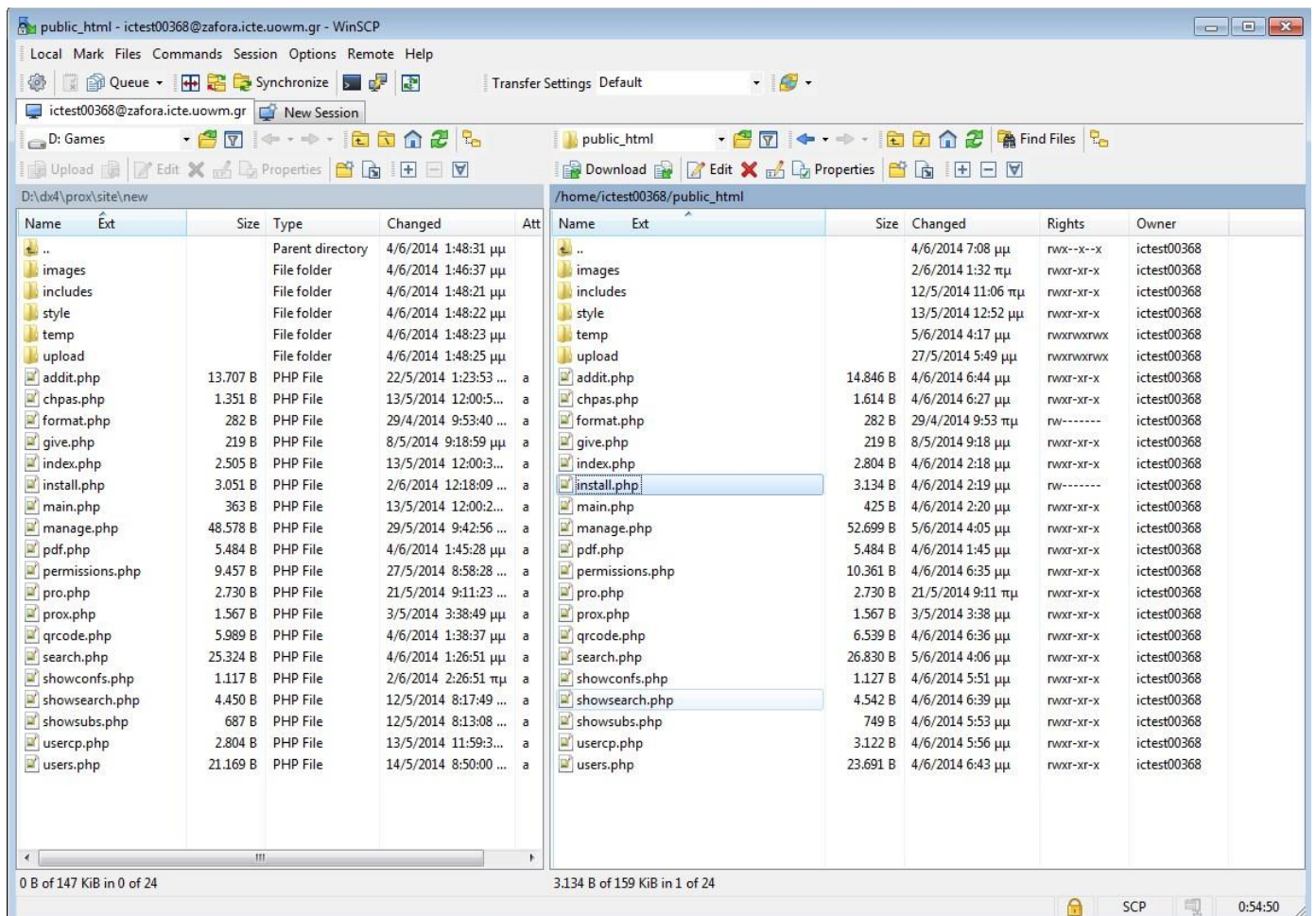
Για την κατασκευή του συστήματος, την συγγραφή κώδικα και την εγκατάσταση και επισκόπηση των στοιχείων που χρησιμοποιούνται, χρησιμοποιήθηκαν ορισμένα εργαλεία. Στο κεφάλαιο αυτό αναλύονται τα εργαλεία αυτά, οι δυνατότητές τους και οι ευκολίες τις οποίες παρείχαν και βοήθησαν στην υλοποίηση αυτής της εργασίας.

### 3.1 WinSCP

Το **WinSCP** (10) είναι ένα από τα καλύτερα προγράμματα για συνδέσεις **FTP/SCP** και λειτουργεί σε περιβάλλον **Windows**. Το συγκεκριμένο εργαλείο χρησιμοποιήθηκε στην διαχείριση των αρχείων που βρισκόταν στον **server**. Ο **server** που χρησιμοποιήθηκε ήταν ο **zafora**, που ανήκει στο τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών και διατίθεται στους φοιτητές για ακαδημαϊκούς σκοπούς.

Μία από τις καλύτερες και πιο χρήσιμες λειτουργίες του είναι η δυνατότητα του να συνδυαστεί με άλλα προγράμματα των **Windows** όπως είναι το **Notepad++**. Χάρη σε αυτό μπορεί να γίνεται η συγγραφή και τροποποίηση κώδικα εύκολα και γρήγορα, και σώζοντας το αρχείο ανεβαίνει αυτόματα στον **server** σε δευτερόλεπτα, κάνοντας την δοκιμή του αρχείου έτοιμη σε πολύ μικρό χρονικό διάστημα. Επίσης το **WinSCP** έχει και άλλες λειτουργίες ευκολίας όπως το σύστημα **drag & drop** που διαθέτει για την εύκολη και γρήγορη μεταφορά αρχείων από και προς τον **server**.

Το πρόγραμμα αυτό είναι αρκετά φιλικό προς τον χρήστη και παρέχει ευκολίες χρήσης σε κάθε κίνηση που επιθυμεί να κάνει ο χρήστης. Στην Εικόνα 4 φαίνεται το περιβάλλον εργασίας ενώ το πρόγραμμα είναι συνδεδεμένο με Επίσης, σε περίπτωση αποσύνδεσης από τον **server** για οποιονδήποτε λόγο, εκτός του ότι το ανοικτό αρχείο που έχει επεξεργαστεί και δεν έχει ανέβει ακόμα στον **server** μένει στο πρόγραμμα επεξεργασίας κειμένου, αποθηκεύεται αυτόματα και σε έναν πρόχειρο χώρο στον υπολογιστή. Έτσι ακόμα και ο υπολογιστής να κλείσει το **WinSCP** θα κάνει ανάκτηση του χαμένου αρχείου και η εργασία θα μπορέσει να συνεχιστεί χωρίς καθυστέρηση.



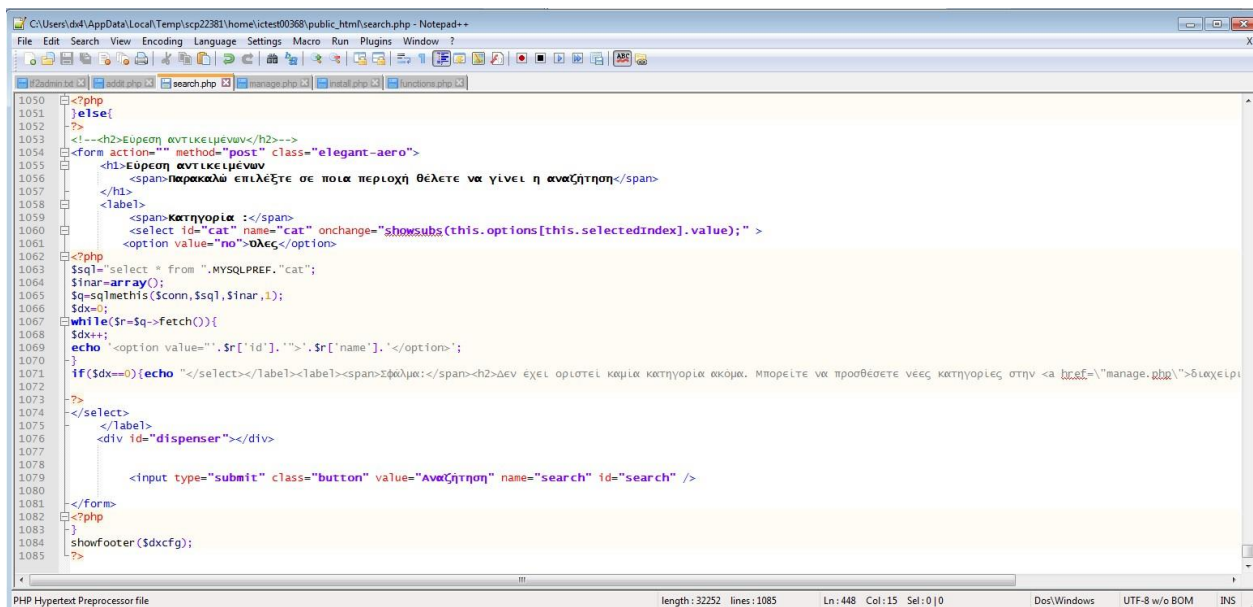
Εικόνα 4: Το περιβάλλον λειτουργίας του WinSCP



## 3.2 Notepad++

Το **notepad++** (11) σε γενικές γραμμές είναι μια αρκετά βελτιωμένη έκδοση του notepad που διαθέτουν τα **Windows**. Διαθέτει διάφορες λειτουργίες που βοηθάνε πολύ τους προγραμματιστές. Μία από αυτές είναι ο κατάλληλος χρωματισμός κώδικα, ανάλογα με την λειτουργία του και η υπόδειξη του που ανοίγει και που κλείνει κάποια **αγκύλη** ή **παρένθεση**. Αυτές οι λειτουργίες βοηθούν εξαιρετικά στην συγγραφή κώδικα, ειδικά αν είναι πολύ μεγάλος σε έκταση.

Το συγκεκριμένο εργαλείο λειτουργούσε σε συνδυασμό με το πρόγραμμα **WinSCP**, όπως αναφέρθηκε παραπάνω. Ο καλός συνδυασμός των 2 προγραμμάτων συντέλεσε στην γρηγορότερη και ευκολότερη ανάπτυξη του κώδικα, εφόσον απέτρεπε προγραμματιστικά λάθη (όπως παράληψη κάποιας αγκύλης ή τοποθέτηση αυτής σε λάθος σημείο).



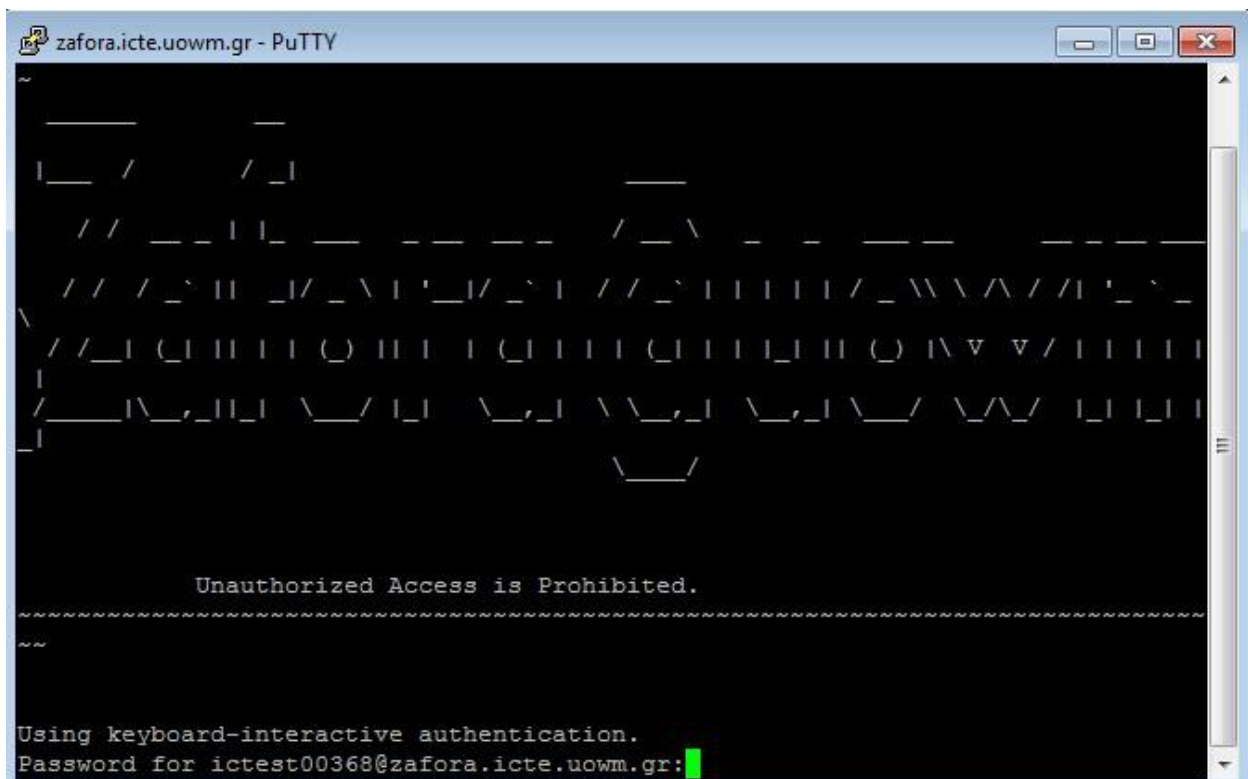
```
1050 <?php
1051 }elseif
1052 ?>
1053 <!--<h2>Εύρεση αντικειμένων</h2-->
1054 <form action="" method="post" class="elegant-aero">
1055 <h3>Εύρεση αντικειμένων
1056 <span>Παρακαλώ επιλέξτε σε ποια περιοχή θέλετε να γίνει η αναζήτηση</span>
1057 </h3>
1058 <label>
1059 <span>Κατηγορία :</span>
1060 <select id="cat" name="cat" onchange="showsubs(this.options[this.selectedIndex].value);" >
1061 <option value="no">Όλες</option>
1062 <?php
1063 $sql="select * from ".MYSQLPREFIX."cat";
1064 $inar=array();
1065 $q=$sqlmethis($conn,$sql,$inar,$i);
1066 $dx=0;
1067 while($r=$q->fetch()){
1068 $dx++;
1069 echo '<option value="'. $r['id']. "'>'. $r['name']. '</options>';
1070 }
1071 if($dx==0){echo "</select></label><span>Σφάλμα:</span><h2>δεν έχει οριστεί καμία κατηγορία ακόμα. Μπορείτε να προσέσετε νέες κατηγορίες στην <a href='\"manage.php\">διαχείρι
1072 ?>
1073 </select>
1074 </label>
1075 <div id="dispenser"></div>
1076 <input type="submit" class="button" value="Αναζήτηση" name="search" id="search" />
1077
1078 </form>
1081 <?php
1082 }
1083 showFooter($dxcfg);
1084 ?>
```

Εικόνα 5 Το περιβάλλον λειτουργίας του Notepad++

### 3.3 Putty

Το εργαλείο **Putty** (12) επιτρέπει να γίνουν **ssh** συνδέσεις σε απομακρυσμένα τερματικά και λειτουργεί σε περιβάλλον Windows. Πέραν των προτερημάτων διασύνδεσης με τον server που προσέφερε το **WinSCP**, χρειαζόταν να γίνει και σύνδεση με **ssh** για διάφορους σκοπούς, όπως εγκατάσταση **server**, αλλά κυρίως για αξιοποίηση της βάσης δεδομένων για έλεγχο της καλής λειτουργίας του συστήματος. Κάθε φορά που έπρεπε να αλλάξει κάτι στην βάση δεδομένων, αυτή η ενέργεια έπρεπε να επιβεβαιωθεί δίνοντας την κατάλληλη **MySQL** εντολή μέσω του εργαλείου αυτού.

Κατά την δημιουργία του server το **Putty** βοήθησε εκτός των εντολών δημιουργίας, και στην συγγραφή και ενεργοποίηση ενός κώδικα "**keep alive**" script το οποίο κρατούσε ζωντανή την βάση **MySQL** σε περίπτωση που αυτή κλείσει για κάποιον λόγο. Επίσης, εκτός των άλλων αποδείχτηκε χρήσιμο όταν χρειάστηκε να διαγραφούν κάτι αρχεία τα οποία δημιουργήθηκαν από το διαδίκτυο έτσι δεν άνηκαν στον χρήστη αλλά στο "**www**".



Εικόνα 6: Το περιβάλλον λειτουργίας του PUTTY

## Κεφάλαιο 4: Ανάλυση κώδικα

Στο κεφάλαιο αυτό γίνεται ανάλυση μέρος του κώδικα των σημαντικότερων αρχείων του συστήματος. Παραθέτονται αρκετές φορές σημεία κώδικα ως παράδειγμα για ανάλυση σε επόμενες γραμμές. Αυτό το κεφάλαιο είναι ιδιαίτερα σημαντικό καθώς φανερώνει την λογική που υπάρχει πίσω από τον κώδικα η οποία αν γίνει αντιληπτή από κάποιον προγραμματιστή θα τον βοηθήσει πολύ σε πιθανή επέκταση η τροποποίηση του προγράμματος.

Αυτό το κεφάλαιο δεν αναλύει τις δυνατότητες κάθε σελίδας αλλά τον κώδικα σε αυτήν. Μετά από αυτό το κεφάλαιο ακολουθεί το κεφάλαιο με τις οδηγίες τις οποίες θα χρειαστεί να διαβάσουν οι χρήστες προτού προβούν στην χρήση του συστήματος.

### 4.1 header.php & footer.php

Ως εισαγωγή ας αρχίσουμε από αυτά τα αρχεία μιας και που βρίσκεται στην αρχή κάθε PHP αρχείου και δημιουργεί το header κάθε σελίδας.

Αρχικά υπάρχει η μεταβλητή `dxcfg`<sup>i</sup>, η οποία είναι ένα array που αν έχει κάποιες τιμές κάνει τις κατάλληλες ρυθμίσεις. Για παράδειγμα αν έχουμε ορίσει `$dxcfg['sessionstart']=1;` στο PHP αρχείο μας, τότε στην αρχή θα τοποθετήσει το `session_start();`

Παρατηρούμε πως το `<link rel = "stylesheet" type = "text/css" media="all" href = "style/global.css" />` υπάρχει ανεξαρτήτως του `dxcfg` γιατί είναι απαραίτητο σε κάθε αρχείο ενώ το `headertitle` ρυθμίζει τον τίτλο της σελίδας.

Στο **footer** περιέχονται τα απαραίτητα tags τα οποία πρέπει να υπάρχουν στο τέλος κάθε ιστοσελίδας. Επίσης, ανάλογα με την σελίδα που βρίσκεται ο χρήστης, εμφανίζει κάποιο βοηθητικό κείμενο το οποίο περιέχει χρήσιμες πληροφορίες σχετικά με τις δυνατότητες της ενότητας που βρίσκεται ο χρήστης. Τέλος, προσθέτει διάφορα στοιχεία όπως είναι ο δημιουργός του συστήματος.

---

<sup>i</sup> dx configuration

## 4.2 functions.php

Η αρχική ιδέα ήταν να περιέχει όλα τα **functions** που μπορεί να χρησιμεύσουν. Στην συνέχεια όμως, λόγω του μεγάλου αριθμού των **functions** που χρησιμοποιήθηκαν, κάθε **function** μπήκε στο δικό του **PHP** αρχείο. Στο **functions.php** υπάρχουν κάποια **functions** γενικής χρήσης τα οποία είναι:

- **redme** (redirect me): Ανακατευθύνει τον χρήστη σε κάποια σελίδα για διάφορους λόγους. Δέχεται 23 παραμέτρους. Η πρώτη είναι η σελίδα η οποία θα σταλεί ο χρήστης. Αν η 1η παράμετρος είναι **selfy**, ο χρήστης είναι σαν να κάνει **reload** την σελίδα κρατώντας μόνο τις **GET** τιμές. Η δεύτερη παράμετρος είναι ο λόγος που γίνεται η ανακατεύθυνση. Η τρίτη ορίζει αν θα εκτυπωθεί στην οθόνη το **footer**.
- **sqlmethis**: Δέχεται ως παράμετρο την **σύνδεση**, το **sql query**, τις **παραμέτρους** και ένα **flag** που ορίζει αν θέλουμε έξοδο. Ως έξοδο δίνει το αποτέλεσμα του **query**.
- **checklogin**: Ελέγχει αν ο χρήστης έχει το κατάλληλο **permission** για να μπει στην σελίδα. Αν δεν το έχει τον ανακατευθύνει στην αρχική. Χαρακτηριστικό αυτής της λειτουργίας είναι ότι αν ο χρήστης δεν έχει κάνει **login**, θυμάται μέσω του session wanted την σελίδα που ήθελε να πάει ο χρήστης και ανάλογα τον ανακατευθύνει μετά το login.
- **megethos**: Ελέγχει το μέγεθος ενός string. Δέχεται 3 παραμέτρους εκ των οποίων η πρώτη είναι η μεταβλητή προς έλεγχο, η δεύτερη το μικρότερο δυνατό μέγεθος και η τρίτη το μέγιστο δυνατό μέγεθος. Αν δεν υπάρχει πρόβλημα επιστρέφει **true**.
- **elenxos**: Ελέγχει τον τύπο ενός string. Η πρώτη παράμετρος είναι η εν λόγω μεταβλητή και η δεύτερη είναι ένας αριθμός που ορίζει τον τύπο του ελέγχου : 1 για **αλφαριθμητικά**, 2 για **ακεραίους θετικούς αριθμούς**, 3 για **e-mail**, 4 για **όλους τους αριθμούς**, 5 για **ακαίραιο**, 6 για **ημερομηνία**, 7 για **ώρα**. Αν δεν υπάρχει πρόβλημα επιστρέφει **true**.
- **showmenu**: Εμφανίζει το **μενού**, ανάλογα με το που βρίσκεται ο χρήστης και τι **permissions** έχει. Αναλυτικότερα, αν κάποιος χρήστης, για παράδειγμα, δεν έχει την κατάλληλη πρόσβαση να προσθέσει κάποιο αντικείμενο, το **μενού** δεν θα του δίνει την δυνατότητα να πλοηγηθεί σε αυτήν την σελίδα.

## 4.3 index.php

Αρχικά ελέγχει αν ο χρήστης έχει ήδη κάνει είσοδο. Αν ναι, τότε τον ανακατευθύνει κατάλληλα. Αλλιώς εμφανίζει την φόρμα **login**. Ο κωδικός αφού περάσει πρώτα από md5 και μετά από **sha1** ελέγχεται με τον κωδικό που είναι αποθηκευμένος στην βάση δεδομένων και ανάλογα γίνεται η είσοδος. Κατά την είσοδο δημιουργείται το session του χρήστη που περιέχει τις πληροφορίες του.

Στις γραμμές 25-33 βλέπουμε τον κώδικα δημιουργίας του session:

```
$_SESSION["logged"]=4; <- flag ότι έχει γίνει είσοδος
```

```
$_SESSION["id"]=$_r['id'];
```

```
$_SESSION["password"]=$_r['password'];
```

```
$_SESSION["username"]=$_r['username'];
```

```
$_SESSION["email"]=$_r['email'];
```

```
$_SESSION["reqch"]=$_r['reqch'];
```

```
$_SESSION["activated"]=$_r['activated'];
```

```
$_SESSION["group"]=$_r['group'];
```

```
$_SESSION['page']=1;
```

Παρατηρούμε πως αποθηκεύονται χρήσιμες πληροφορίες: το id του χρήστη, ο κωδικός (σε κωδικοποιημένη μορφή) το username, το e-mail, αν χρειάζεται να αλλάξει κωδικό, αν είναι ενεργοποιημένος ο λογαριασμός, το group στο οποίο ανήκει και θέτει τον αριθμό της σελίδας (στις παραχωρήσεις αντικειμένου) στο 1.

## 4.4 usercp.php

Η σελίδα αυτή ήταν από τις πρώτες που δημιουργήθηκαν. Στην συγκεκριμένη σελίδα ο χρήστης μπορεί να αλλάξει τα στοιχεία του.

Ο παρακάτω κώδικας αποτελεί τον κατασκευαστή των header σε κάθε σελίδα του συστήματος:

```
require_once('includes/require.php');
```

```
$dxcfg['headertitle']="Αλλαγή Ρυθμίσεων";
```

```
$dxcfg['cssfile']="menu";
```

```
$dxcfg['helpline']="usercp";
```

```
$dxcfg['cssfilee']="elegantaero";
```

```
$dxcfg['sessionstart']=1;
```

```
$dxcfg['showerrors']=DISPERROR;
```

```
showheader($dxcfg);
```

```
$conn=connect();
```

```
checklogin();
```

```
showmenu("usercp");
```

Με λίγα λόγια, θέτει τον τίτλο της σελίδας, βάζει 2 **CSS** που χρειάζονται, θέτει το **helpline** που θα εμφανιστεί στο τέλος, δίνει το σήμα ότι χρειάζεται να είναι ενεργό το **session** και τέλος, εφόσον μπήκαν οι απαραίτητοι παράμετροι, εμφανίζει το **header** καλώντας την κατάλληλη **συνάρτηση**. Στο τέλος γίνεται η σύνδεση με την βάση δεδομένων, ελέγχει αν ο χρήστης είναι συνδεδεμένος και εμφανίζει το **menu** με την παράμετρο που δείχνει ότι βρισκόμαστε στο **usercp**.

Εδώ παρατηρούμε και την μορφή **html** που πρέπει να έχει η φόρμα **elegant aero**:

```
<form action="" method="post" class="elegant-aero">
  <h1>Change your settings
  <span>If you do not wish to change something leave it as it is</span> </h1>
  <label> <span>Username :</span>
  <input id="username" type="text" name="username" value="<?php echo
  $_SESSION["username"] ?>" />
  </label>
  <input type="submit" class="button" value="Submit Changes" name="submit" id="submit"
  />
</form>
```

Παρατηρείται πως στην αρχή έχει τίτλο (με το **h1 tag**) και υπότιτλο (με το **span** που βρίσκεται μέσα στο **h1 tag**). Μετά φαίνεται πως κάθε **input** τοποθετείται μέσα σε **label** αφού πρώτα προηγηθεί το **span** με την ονομασία/περιγραφή του **label**. Ακόμα και κάποια μικρή αλλαγή στην δομή, επιφέρει μεγάλες και άσχημες αλλαγές στο τελικό αποτέλεσμα. Το **elegant aero** πέρασε από πολλές τροποποιήσεις μέχρι την τελική μορφή του.

Στην συνέχεια, αφού ο χρήστης επιβεβαιώσει τις πληροφορίες, γίνονται οι απαραίτητοι έλεγχοι. Συγκεκριμένα για το πεδίο **e-mail**:

```
if (!elenxos($email,3)){redme("self","Please enter a valid e-mail.");}
```

```
if (!megethos($email,3,30)){redme("self","The email entered should be less than 30 characters.");}
```

Στο πρώτο **if** ελέγχεται αν το **e-mail** έχει την κατάλληλη μορφή. Σε περίπτωση σφάλματος η σελίδα επαναφορτώνεται. Στο δεύτερο **if** ελέγχεται αν το **e-mail** είναι πολύ μεγάλο για να χωρέσει στην βάση δεδομένων και ανάλογα γράφει το κατάλληλο μήνυμα. Μεγάλα **e-mail** (περισσότερο από 30 χαρακτήρες) δεν επιτρέπονται για διάφορους άλλους λόγους.

Σημαντική επίσης είναι και η μορφή του κώδικα που χρησιμοποιείται για την σύνδεση με την βάση δεδομένων:

```
$sql="UPDATE ".MYSQLPREFIX."users  
SET password=?, username=?, email=?  
WHERE id=?";  
$inar=array($pass,$username,$email,$userid);  
sqlmethis($conn,$sql,$inar);
```

Στην συγκεκριμένη εντολή δεν χρειαζόμαστε έξοδο καθώς απλά επεξεργαζόμαστε τα στοιχεία του χρήστη. Εφόσον λοιπόν γράψουμε το κατάλληλο **sql** μαζί με τις ανάλογες **παραμέτρους**, είμαστε έτοιμοι να εκτελέσουμε την συνάρτηση **sqlmethis** η οποία θα εκτελέσει την εντολή που δώθηκε.

Μετά τις αλλαγές ο χρήστης πρέπει να ξανασυνδεθεί για να οριστεί ξανά το **session**:

```
session_destroy();
```

```
redme("index.php","You now need to relogin.");
```



## 4.5 users.php

Είναι η δεύτερη σε σειρά σελίδα που δημιουργήθηκε και η πρώτη που χρειάστηκε πολλές γραμμές κώδικα για να εξασφαλιστεί η ευκολία και άνεση του χρήστη να κάνει τις ενέργειες που θέλει.

Αρχικά στις γραμμές **16-43**:

```
<script type="text/javascript">
//
function ssearchu() {
    document.getElementById('screateu').style.display = 'none';
    document.getElementById('screateg').style.display = 'none';
    document.getElementById('ssearchg').style.display = 'none';
    document.getElementById('ssearchu').style.display = "";
}
function ssearchg() {
    document.getElementById('screateu').style.display = 'none';
    document.getElementById('screateg').style.display = 'none';
    document.getElementById('ssearchg').style.display = "";
    document.getElementById('ssearchu').style.display = 'none';
}
...
</pre></div><div data-bbox="456 937 540 965" data-label="Page-Footer"><hr/><p>{ 41 }</p></div>
```

Στον παραπάνω κώδικα, βλέπουμε το **JavaScript** το οποίο εμφανίζει ή εξαφανίζει ανάλογα τα πεδία που θέλει να δει ο χρήστης. Η σελίδα αρχικά είχε ορατά όλα τα πεδία αλλά καθώς οι **4 λειτουργίες (αναζήτηση/επεξεργασία χρήστη, αναζήτηση/επεξεργασία group, προσθήκη χρήστη, προσθήκη group)** που εκτελεί ήταν μεγάλες σε όγκο αποφασίστηκε να χρησιμοποιηθεί αυτός ο τρόπος καθώς είναι εύκολος στον κώδικα και ωραίος αισθητικά στον χρήστη. Κάθε φόρμα φυσικά έχει και το ανάλογο **div**, για παράδειγμα:

```
<div id="screateg" style="display: none">
```

Η γραμμή αυτή δηλώνει το **div** που ενθυλακώνει την φόρμα για δημιουργία **group**. Ανάλογα είναι και τα **div** των άλλων φορμών.

Αν αναλύσουμε τον κώδικα σειριακά παρατηρούμε τα εξής :

```
if({$isAdmin==1}){die ("Cannot delete admin.");}
```

Εδώ αποτρέπουμε τον χρήστη από το να διαγράψει κάποιον **admin**. Αυτό γίνεται για πολλούς λόγους αλλά ο κυριότερος είναι για να μην διαγράψει τον εαυτό του. Στην **διαγραφή** κάποιου **γκρουπ**:

```
$sql="UPDATE ".MYSQLPREFIX."users
```

```
SET `group`=?
```

```
WHERE `group`=?";
```

```
$inar=array($usergid,$userid);
```

```
sqlmethis($conn,$sql,$inar,0);
```

Παρατηρούμε πως όταν διαγράψουμε κάποιο γκρουπ, το σύστημα αυτόματα τοποθετεί αυτόματα όλους τους χρήστες που άνηκαν σε εκείνο το **γκρουπ** στο **Default Use Group**. Αυτό γίνεται για να αποφύγουμε το γεγονός κάποιος χρήστης να είναι σε ένα **γκρουπ** που δεν υπάρχει. Επίσης διαγράψουμε όλα τα permissions που έχει το group που διαγράψαμε:

```
$sql="delete from ".MYSQLPREF."permissions WHERE user=?";
```

```
$inar=array($userid);
```

```
sqlmethis($conn,$sql,$inar,0);
```

Παρατηρείται για πρώτη φορά την μορφή που πρέπει να έχουν οι πίνακες του συστήματός μας. Η γενικότερη μορφή πρέπει να είναι η εξής:

```
<table id="rounded-corner" summary="Search Results">  
  <thead>  
    <tr>  
      <th scope="col" class="rounded-company">Column</th>  
      <th scope="col" class="rounded">Column</th>  
    </tr>  
  </thead>  
<tbody>  
  <tr>  
    <td>Data</td>  
    <td>Data</td>  
  </tr>  
</tbody>  
<tfoot>
```

```
<tr>
  <td class="rounded-foot-left">Data</td>
  <td class="rounded-foot-right">Data</td>
</tr>
</tfoot>
</table>
```

Γενικότερα θα δείτε πως υπάρχουν πολλές παραλλαγές αυτής της μορφής του πίνακα (για παράδειγμα αν δεν υπάρχουν δεδομένα) που χρησιμοποιούν άλλο **format**. Επειδή το **tfoot** είναι απαραίτητο στοιχείο του πίνακα και πρέπει πάντοτε να βρίσκεται στο τέλος και λόγω του ότι δεν γνωρίζουμε από πριν ποιο είναι το τέλος του πίνακα, η επεξεργασία της εξόδου γίνεται με έναν διαφορετικό τρόπο. Κάθε φορά αποθηκεύει τα δεδομένα **της τωρινής σειράς** και εκτυπώνει τα δεδομένα **της προηγούμενης**. Με αυτόν τον τρόπο όταν φεύγει από το **while** της **MySQL**, το σύστημα δεν έχει εκτυπώσει τα τελευταία δεδομένα, αλλά τα έχει αποθηκεύσει. Έτσι, αφού βγει τα εκτυπώνει χρησιμοποιώντας το **tfoot** αφού πρώτα κλείσει το **tbody**. Λόγω του ότι τα δεδομένα προ αποθηκεύονται, οι μεταβλητές έχουν το πρόθεμα **"pre"**.

## 4.6 manage.php

Το τρίτο σε σειρά να δημιουργηθεί και το μεγαλύτερο σε μέγεθος καθώς είναι υπεύθυνο για την δομή όλου του συστήματος βάσης δεδομένων. Αρχίζοντας, ας αναλύσουμε τα **functions** που χρησιμοποιούνται για την διευκόλυνση σε θέμα κώδικα:

- **damnihadtoaddtype**: Στην αρχική έκδοση, ο χρήστης δεν είχε την επιλογή να ορίσει σε ένα πεδίο τον τύπο του. Έπειτα κρίθηκε απαραίτητη αυτή η προσθήκη η οποία βέβαια έφερε πολλές αλλαγές στον κώδικα μιας και που μια τέτοια κίνηση ήταν μια ριζική αλλαγή (η οποίες συντέλεσε και στο όνομα της συνάρτησης). Η συγκεκριμένη συνάρτηση κάνει τους απαραίτητους ελέγχους στο πεδίο που θέλει να ορίσει ο χρήστης. Ελέγχει αν γίνεται, αν έχει βάλει σωστές τιμές και αν η **default** τιμή είναι σωστή (δεν γίνεται σε πεδίο αριθμού να μπει default “υπολογιστής”).
- **showtype**: Ανάλογα με τον **τύπο**, επιστρέφει την κατάλληλη ονομασία του για να είναι αναγνώσιμος από τον χρήστη.
- **MySQLtypec**: (c for coding) Μετατρέπει τον τύπο σε μορφή που καταλαβαίνει η **MySQL**, ώστε να δημιουργηθούν ή να επεξεργαστούν τα ανάλογα **πεδία**.
- **MySQLtype**: Μετατρέπει τον τύπο σε αριθμό, ώστε να εκχωρηθεί η κατάλληλη τιμή στον πίνακα **subn**. Οι τιμές είναι : 1 - **κείμενο**, 2 - **μεγάλο κείμενο** (Δημιουργείται textarea), 3 - **αριθμός**, 4 - **ακέραιος**, 5 - **Ημερομηνία**, 6 - **Ώρα**, 7 - **Εικόνα**
- **MySQLtyper**: (r for reverse) Μετατρέπει τον αριθμό, πίσω στην μορφή που του αναλογεί.

Προτού προχωρήσουμε, θα αναλύσουμε την δομή του πίνακα **subn**, αποτελείται από 8 πεδία τα οποία αναφέρονται στον πίνακα sub:

- 1) Το **id** που έχουν όλοι οι πίνακες.
- 2) Το **name** που είναι το όνομα του πεδίου.
- 3) Το **dname** που είναι το όνομα που βλέπει ο χρήστης
- 4) Το **default**, η τιμή που θα πάρει αν ο χρήστης το αφήσει κενό.
- 5) Το **ttype** που είναι ο τύπος του πεδίου (1-7)
- 6) Το **mant**a (mandatory) που συμβολίζει πως το πεδίο είναι **υποχρεωτικό**.
- 7) Το **unix** (unique) που συμβολίζει πως το πεδίο είναι **μοναδικό**.
- 8) Το **qrcode** που συμβολίζει πως η τιμή του θα αναγράφεται πάνω στο **QR Label**.

Επίσης χρειάζεται να αναλυθεί και η μορφή και χρησιμότητα του πίνακα **globals**:

Ο πίνακας **globals** έχει παρόμοια δομή, μόνο που αυτός έχει 2 παραπάνω πεδία, το **cat** (category) και το **length**. Φυσικά, το **length** κάθε κατηγορίας φαίνεται χάρη στην εντολή **describe**. Όπως έχει αναφερθεί παραπάνω, ο πίνακας αυτός είναι χρήσιμος στην εύκολη δημιουργία πινάκων **νέων υποκατηγοριών**, αλλά και στην δημιουργία **νέων στοιχείων πίνακα** σε πίνακες πολλαπλών **υποκατηγοριών**. Έτσι οι 2 λειτουργίες της **global** εκτελούνται με διαφορετικό τρόπο. Η πρώτη γίνεται αυτόματα με την δημιουργία **νέας κατηγορίας** και η δεύτερη γίνεται από τον χρήστη με την επιλογή “**εφαρμογή**”. Ο κώδικας της εφαρμογής υπάρχει στην γραμμή 273 και μετά. Γενικά στον κώδικα παρατηρούμε πως συλλέγει το **id** των υποκατηγοριών (τα ονομάζει **dogs**) ανάλογα με την κατηγορία (cat) και όλα τα στοιχεία για την δημιουργία του τοποθετούνται σε μεταβλητές. Αξίζει να σημειωθεί μέρος του κώδικα:

```
if($mahflag){$mahprox="AFTER `.$r['name']`";}
```

Υπήρξε ένα πρόβλημα με τα **global** στοιχεία πίνακα, όταν αυτά έμπαιναν εκπρόθεσμα στους πίνακες μέσω της εντολής “**εφαρμογή**”. Αναλυτικότερα, τα νέα στοιχεία πίνακα εισχωρούνταν στο τέλος κάθε πίνακα, έχοντας ως αποτέλεσμα την μη σωστή ταξινόμηση των στοιχείων κάθε πίνακα. Στο συγκεκριμένο κομμάτι (μετά τον απαραίτητο έλεγχο) το νέο στοιχείο πίνακα τοποθετείται μετά το τελευταίο **global**, ώστε να υπάρχει μια καλύτερη κατανομή. Βέβαια ο πίνακας **subn** είναι μια τελείως διαφορετική ιστορία γιατί ουσιαστικά γίνεται sort σύμφωνα με το **id**. Για τον λόγο αυτό παρατηρούμε σε πολλά σημεία εντολών της **MySQL** την παράμετρο:

```
ORDER BY `name` ASC
```

Η τελική μορφή του πίνακα σχηματίζεται από την παρακάτω εντολή (όχι πάντα):

```
$sql="ALTER TABLE ".$MYSQLPREFIX."sub".$dogs[$x]." ADD `globalat` $type( $length ) NULL$default $mahprox";
```

Παρατηρείται πως δεν έχει κάποια άλλη ιδιότητα (όπως NOT NULL [**mandatory**] ή **Unique**). Αυτό γίνεται για διάφορους λόγους, κυρίως ο έλεγχος των τιμών γίνεται από το σύστημα μέσω κώδικα.

Στην συνέχεια αξίζει να σημειωθεί πως εδώ βλέπουμε για πρώτη φορά την ανάγκη επιβεβαίωσης για κάποια ενέργεια του χρήστη. Προχωράμε στην διαγραφή κατηγορίας που αρχίζει από την γραμμή 511. Αυτή η διαδικασία αποδείχθηκε αρκετά περίπλοκη λόγω του μεγάλου πλήθους των λειτουργιών της κατηγορίας που έπρεπε να διαγραφούν. Οι γραμμές

μέχρι την **458** αφορούν το σημάδεμα των στοιχείων που είναι προς διαγραφή και ανάλογα διαγράφονται όποιες εικόνες αντιστοιχούν εκεί (θα αναλύσουμε τις εικόνες παρακάτω). Στον παρακάτω κώδικα έχουμε την εξής ανάλυση:

```

$sql="delete from ".MYSQLPREFIX."subs WHERE id=?"; <- Διαγράφουμε όλες τις υποκατηγορίες
$inar=array($prox);
sqlmethis($conn,$sql,$inar,0);
$sql="drop table ".MYSQLPREFIX."sub$prox"; <- Διαγράφονται όλοι οι πίνακες των
υποκατηγοριών, όπως επίσης και
$inar=array($userid);
sqlmethis($conn,$sql,$inar,0);
$sql="drop table ".MYSQLPREFIX."subn$prox"; <- οι πίνακες που ορίζουν τις ιδιότητές τους
$inar=array($userid);
sqlmethis($conn,$sql,$inar,0); }
$sql="delete from ".MYSQLPREFIX."globals WHERE cat=?"; <- Διαγράφονται όλα τα global που
σχετίζονται με την κατηγορία που είναι να διαγραφεί
$inar=array($userid);
sqlmethis($conn,$sql,$inar,0);
$sql="delete from ".MYSQLPREFIX."gitems WHERE cat=?"; <- Διαγράφονται όλα τα αντικείμενα
που ανήκουν στην κατηγορία από το ευρετήριο
$inar=array($userid);
sqlmethis($conn,$sql,$inar,0);
$proxuserid="c".$userid;
$sql="delete from ".MYSQLPREFIX."permissions WHERE perm=?"; <- Κώδικας από παλιά έκδοση
που είχε άλλες λειτουργίες.
$inar=array($proxuserid);
sqlmethis($conn,$sql,$inar,0);
$sql="delete from ".MYSQLPREFIX."cat WHERE id=?"; <- Τέλος, διαγράφεται η κατηγορία
    Στην συνέχεια παρατηρείται ο κώδικας:
$global[$viewglobals][1]=$r['id'];
//$global[$viewglobals][2]=$r['name'];
$global[$viewglobals][3]=$r['type'];
$global[$viewglobals][4]=$r['length'];
//$global[$viewglobals][5]=$r['textbox'];
$global[$viewglobals][6]=$r['default'];
$global[$viewglobals][7]=$r['dname'];
$global[$viewglobals][8]=$r['manta'];
$global[$viewglobals][9]=$r['unix'];
$global[$viewglobals][10]=$r['qrcode'];

```

Αφορά την αποθήκευση των **global** μεταβλητών για περαιτέρω ενέργειες. Ο κώδικας αυτός είναι σημαντικός και από πλευράς ιστορίας, μιας και που ο σχολιασμός μερικών από τα πεδία υποδεικνύει παλαιές εκδόσεις του συστήματος. Γενικότερα υπάρχουν αρκετά κομμάτια κώδικα (σε σχόλια) από προηγούμενες εκδόσεις και αυτό γίνεται εκτός των ιστορικών λόγων, και για λόγους ότι όταν κάποιος επιχειρήσει να αλλάξει την μορφή του συστήματος και να προσθέσει κάποια λειτουργία, αυτά τα σχόλια θα τον βοηθήσουν αρκετά.

Επίσης, έχουμε τον κώδικα:

```
$sql="CREATE TABLE `".MYSQLPREFIX."subn".$idme."` (
`id` INT( 10 ) NOT NULL AUTO_INCREMENT ,
`name` varchar( 15 ) NOT NULL ,
`dname` varchar( 30 ) NOT NULL ,
`ttype` int( 2 ) NOT NULL ,
`manta` int( 1 ) NOT NULL ,
`unix` int( 1 ) NOT NULL ,
`qrcode` int( 1 ) NOT NULL ,
PRIMARY KEY ( `id` )) CHARACTER SET utf8 COLLATE utf8_unicode_ci";
```

Βλέπουμε την κώδικα που δημιουργεί την μορφή των πινάκων subn. Σημαντικό κομμάτι που στην αρχή παραλείφθηκε είναι το **CHARACTER SET utf8 COLLATE utf8\_unicode\_ci** το οποίο είναι για να υποστηρίζει **ελληνικά**.

Στην συνέχεια:

```
$sql="INSERT INTO `".MYSQLPREFIX."subn".$idme."
VALUES (NULL";
$mplampla=")";
$mpladiblah="";
if($viewglobals>0){$dos=$global[$x][3];
```



```
$uno=$global[$x][7];
```

```
$tre=$global[$x][1];
```

```
$abra=$global[$x][8];
```

```
$katabra=$global[$x][9];
```

```
$alakazam=$global[$x][10];
```

```
$mpladiblah=$mpladiblah.",'gatt$tre','$uno','$dos','$abra','$katabra','$alakazam'; }
```

```
$sql=$sql.$mpladiblah.$mplampla;
```

Υπάρχει ο κώδικας που εκχωρεί τις κατάλληλες τιμές στον πίνακα που μόλις δημιουργήθηκε. Αξίζει να σημειωθεί πως το τελευταίο id το ανακτάμε με την εντολή:

```
$idme = $conn->lastInsertId();
```

Στις **σειρές 901-960** βλέπουμε με ποιον τρόπο το σύστημα χειρίζεται τα δεδομένα που αντλεί από την εντολή **describe**. Στις πρώτες εκδόσεις του συστήματος, ο πίνακας **subn** δεν υπήρχε έτσι όλες οι πληροφορίες **αντλώνταν** χάρη σε αυτήν την εντολή. Βέβαια καθώς οι απαιτήσεις αυξήθηκαν, ο πίνακας **subn** ήταν πλέον απαραίτητος και φυσικά για ευκολία όλα τα στοιχεία εκτός από το **length** μπήκαν σε αυτόν. Το **describe** κρίθηκε σημαντικό να κρατηθεί, όχι μόνο για το **length** αλλά και για βοήθεια του προγραμματιστή αν χρειαστεί να κάνει κάποια πρόσθετη **λειτουργία** που χρειάζεται την εντολή **describe**.

## 4.7 permissions.php

Στην συνέχεια και τέταρτο στην δημιουργία του, συναντάμε την λειτουργία που ρυθμίζει τα δικαιώματα. Στον κώδικα αν δεν υπάρχουν παράμετροι γίνεται η εμφάνιση όλων των δικαιωμάτων που έχουν οριστεί. Επειδή ο πίνακας permissions έχει μόνο το id της ομάδας, χρειάζεται να γίνει σύνδεση με τον πίνακα group για να ανακτηθούν τα ονόματα των ομάδων. Ο τρόπος που χρησιμοποιείται για αυτόν τον σκοπό είναι να δηλωθεί ένας πίνακας όπου το index του να είναι το id της ομάδας, και η τιμή του το όνομα της ομάδας. Αυτό το παρατηρούμε στο παρακάτω κομμάτι κώδικα:

```
$sql="select * from ".MYSQLPREFIX."group";  
  
$inar=array();  
  
$q=sqlmethis($conn,$sql,$inar,1);  
  
while($r=$q->fetch()){ $gid=$r['id'];  
  
$showmethis[$gid]=$r['name'];}
```

Αξίζει να σημειώσουμε πως εδώ συναντάμε και εκτέλεση MySQL κώδικα μέσα στην εκτέλεση άλλου MySQL κώδικα. Αυτό φυσικά απαιτεί την χρήση άλλων μεταβλητών, ώστε να μην υπάρχει σύγκρουση με τις μεταβλητές που ήδη χρησιμοποιούνται. Ένα παράδειγμα κώδικα είναι:

```
$sqll="select * from ".MYSQLPREFIX."subs where id=?";  
  
$inarr=array($me);  
  
$qq=sqlmethis($conn,$sqll,$inarr,1);  
  
while($rr=$qq->fetch()){ $name=$rr['name'];}
```

## 4.8 addit.php

Εδώ ο χρήστης μπορεί να προσθέσει κάποιο νέο αντικείμενο στην βάση δεδομένων. Για πρώτη φορά παρατηρούμε την χρήση του ajax:

```
<script type="text/javascript">
//
function showsubs(str){var xmlhttp;
if (window.XMLHttpRequest) {<i>// code for IE7+, Firefox, Chrome, Opera, Safari</i>
xmlhttp=new XMLHttpRequest(); }else {<i>// code for IE6, IE5</i>
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP"); }
xmlhttp.onreadystatechange=function() {
if (xmlhttp.readyState==4 &amp;&amp; xmlhttp.status==200) {
document.getElementById("dispenser").innerHTML=xmlhttp.responseText; } }
xmlhttp.open("GET","showsups.php?sub="+str,true);
xmlhttp.send();}]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="112 672 889 732" data-label="Text"><p>Ο συγκεκριμένος κώδικας εμφανίζει τις υποκατηγορίες της κατηγορίας που θα επιλέξει ο χρήστης. Αυτό γίνεται σε πραγματικό χρόνο χωρίς να χρειαστεί η επαναφόρτωση της σελίδας. Ο κώδικας html είναι ο εξής:</p></div><div data-bbox="112 749 851 766" data-label="Text"><pre>&lt;select id="cat" name="cat" onchange="showsups(this.options[this.selectedIndex].value)" &gt;</pre></div><div data-bbox="112 782 889 885" data-label="Text"><p>Αναλυτικότερα, κάθε φορά που αλλάζει η τιμή του <b>select (onchange)</b> καλεί την συνάρτηση με παράμετρο την τιμή που επέλεξε ο χρήστης από την <b>dropdown λίστα</b> με τις κατηγορίες. Αν ο χρήστης δεν επιλέξει κατηγορία, η επιλογή υποκατηγορίας εξαφανίζεται. Ο χρήστης δεν έχει δικαίωμα να προσθέσει αντικείμενο σε μια κατηγορία αν δεν επιλέξει υποκατηγορία. Αυτό γίνεται για να εξασφαλιστεί η καλή δομή του συστήματος.</p></div><div data-bbox="456 937 540 965" data-label="Page-Footer"><hr/><p>{ 51 }</p></div>
```

Στο αρχείο παρατηρούμε πολλές γνώριμες συναρτήσεις αλλά και μια νέα. Προτού προχωρήσουμε στην ανάλυση της συνάρτησης, σημαντικό είναι να αναλύσουμε τον κώδικα που χειρίζεται το **ανέβασμα** της εικόνας στον **server** και την δημιουργία **thumbnail** που χρειάζεται στην προβολή. Στον κώδικα μπορούμε να τον δούμε το εξής:

```
if(($type=='image')and(isset($_FILES[$prox]["name"]))){$success=1;
$allowedExts = array("gif","jpeg","bmp","jpg","png");
$temp = explode(".", $_FILES[$prox]["name"]);
$extension = strtolower(end($temp));
if (in_array($extension, $allowedExts)){
...
if($success==1){
$toinput="dximage".uniqid().substr(sha1($prox),5,8);
move_uploaded_file($_FILES[$prox]["tmp_name"],"upload/".$toinput."n".$extension);
chmod("upload/".$toinput."n".$extension, 0777);
$sec=$extension;
if($sec=='jpg'){ $sec='jpeg';}
$t = 'imagecreatefrom'.$sec;
$t = str_replace('.',",$t);
$source_image = $t('upload/".$toinput."n".$extension);
$width = imagesx($source_image);
$height = imagesy($source_image);
$desired_width=min(100,$width);
//$desired_height = floor($height * ($desired_width / $width));
$desired_height = min(100,floor($height * ($desired_width / $width)));
$virtual_image = imagecreatetruecolor($desired_width, $desired_height);
imagecopyresampled($virtual_image, $source_image, 0, 0, 0, 0, $desired_width,
$desired_height, $width, $height);
imagejpeg($virtual_image, "upload/".$toinput."t".$extension);
chmod("upload/".$toinput."t".$extension, 0777);
$toinput=$toinput."$extension";
$damnoob++;
$nowihavetodeletethem[$damnoob]=$toinput;
}elseif($success==9){
echo "Error : File is not an allowed type of image.";
learntoread();
die ();}}
<form enctype="multipart/form-data" action="" method="post" class="elegant-aero">
```

Καταρχάς, στο **html** μέρος του κώδικα, η φόρμα πρέπει να έχει την μορφή `enctype="multipart/form-data"`, ώστε να μπορεί να δεχθεί δεδομένα (εικόνα). Όταν ο χρήστης ανεβάσει την εικόνα μπορούμε να την επεξεργαστούμε. Αρχικά ελέγχουμε αν αυτό που έστειλε ο χρήστης είναι εικόνα από τον υπολογιστή του ή **url** με την εικόνα (εκεί δεν την ανεβάζουμε στον server). Αφού γίνει αυτό ελέγχουμε αν είναι αποδεκτός τύπος εικόνας. Οι αποδεκτοί τύποι είναι: **gif, jpeg, bmp, jpg, png**

Εφόσον γίνουν οι απαραίτητοι έλεγχοι, ορίζουμε το όνομα με το οποίο θα ανέβει η εικόνα. Το format που χρησιμοποιεί το σύστημα είναι **dximage + ένα αλφαριθμητικό που εξάγει η php ανάλογα με την ώρα + ένα κομμάτι από το hash του ονόματος του πεδίου που είναι να μπει**. Αυτό γίνεται για να μην υπάρχουν 2 εικόνες με το ίδιο όνομα. Στην συνέχεια αποθηκεύεται με την κατάληξη **n** (normal) και προχωράμε στην δημιουργία **thumbnail**. Ο κώδικας αναλύει προσεκτικά τις διαστάσεις και αναλογίες της εικόνας. Ανάλογα με το ύψος και το μήκος της, την διαμορφώνει κατάλληλα, ώστε να έχει μέγιστο ύψος **100 pixels** και μέγιστο μήκος **100 pixels**. Τέλος αποθηκεύει και το **thumbnail** με κατάληξη **t**. Το **thumbnail** δημιουργείται για να γίνεται γρηγορότερη η φόρτωση των εικόνων όταν ο χρήστης βλέπει ένα ή περισσότερα αντικείμενα. Βέβαια, αν ο χρήστης θελήσει να δει ολόκληρη την εικόνα μπορεί να πατήσει πάνω στο **thumbnail** και να μεταβεί εκεί που βρίσκεται η κανονική εικόνα. Σε περίπτωση που ο χρήστης βάλει κάποιο λάθος πεδίο, όλες οι ανεβασμένες εικόνες στην φόρμα αυτή πρέπει να διαγραφούν, από όπου βγαίνει και η συνάρτηση *learntoread*:

```
function learntoread(){global $damnoob,$nowihavetodeletethem;
if($damnoob==0){return;}
for($x=1;$x<=$damnoob;$x++){$delete=$nowihavetodeletethem[$x];
$pieces = explode(".", $delete);
if($pieces[1]=="zip"){unlink('files/.$delete');}else{$proxnormal=$pieces[0].".n.". $pieces[1];
$proxthumb=$pieces[0].".t.jpeg";
unlink('upload/.$proxnormal');
unlink('upload/.$proxthumb');}}return;}
```

Εδώ χρησιμοποιούνται οι global μεταβλητές που σε αυτές είναι αποθηκευμένες τα στοιχεία των εικόνων που μόλις ανέβηκαν. Ο κώδικας τις διαγράφει με την εντολή unlink.

Στην συνέχεια υπάρχει ο εξής κώδικας:

```
<script type="text/javascript">
$(document).ready(function(){ $("#<?php echo $id; ?>").hover(function(){
    $("#<?php echo "pok".$id; ?>").show(); },function(){
    $("#<?php echo "pok".$id; ?>").hide(); });});
</script>
```

Βλέπουμε το **JavaScript** (με την βοήθεια της **jquery**) να κρύβει και να δείχνει πληροφορίες σχετικά με το πεδίο που ο χρήστης τις βλέπει όταν έχει τον δείκτη του ποντικιού πάνω από το πεδίο (hover). Στην συνέχεια αναλύεται το παρακάτω **script**:

```
<script type="text/javascript">
//
$(function() { $( "#&lt;?php echo $id; ?&gt;" ).datepicker({dateFormat: 'yyyy-mm-dd'}); });
//]]&gt;
&lt;/script&gt;</pre>
</div>
<div data-bbox="111 598 889 659" data-label="Text">
<p>Υπάρχει το <b>JavaScript</b> που βοηθάει στην επιλογή ημερομηνίας. Παρατηρούμε ότι το τελικό format είναι <b>χρόνος-μήνας-ημέρα</b> καθώς σε αυτήν την μορφή τα δέχεται η <b>MySQL</b>. Στην συνέχεια υπάρχει ο κώδικας:</p>
</div>
<div data-bbox="130 675 772 849" data-label="Text">
<pre>&lt;script type="text/javascript"&gt;
//<![CDATA[
$(document).ready(function(){
    $('#&lt;?php echo $id; ?&gt;').ptTimeSelect({    }); //end ptTimeSelect()    });
//]]&gt;
&lt;/script&gt;</pre>
</div>
<div data-bbox="456 937 540 965" data-label="Page-Footer">
<hr/>
<p>{ 54 }</p>
</div>
```

Βλέπουμε το **JavaScript** που δημιουργεί την ώρα. Το τελικό αποτέλεσμα είναι **ώρα:λεπτό** με το **κρυφό** πεδίο “δευτερόλεπτο” καθώς αυτή είναι η μορφή που δέχεται η **MySQL**. Επίσης έχουμε τον εξής κώδικα:

```
<script type="text/javascript">
//
function fundx&lt;?php echo$cid ?&gt;() {
    var input = document.getElementById("&lt;?php echo$cid ?&gt;");
    var input2 = document.createElement('input');
    input2.id = input.id;
    input2.name = input.name;
    input2.value = input.value;
    if (document.getElementById("but&lt;?php echo$cid ?&gt;").value == 'Upload') {
        input2.type = 'file';
        input.parentNode.replaceChild(input2, input);
        document.getElementById("but&lt;?php echo$cid ?&gt;").value = 'URL'; } else {
        input2.type = 'text';
        input.parentNode.replaceChild(input2, input);
        document.getElementById("but&lt;?php echo$cid ?&gt;").value = 'Upload'; } }
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="112 754 889 836" data-label="Text"><p>Παρατηρείται πως το <b>JavaScript</b> το οποίο δίνει την επιλογή στον χρήστη σε κάθε πεδίο εικόνας να μπορεί να διαλέξει αν θα βάλει κάποιο <b>url</b> εικόνας από το διαδίκτυο, ή αν θα <b>ανεβάσει</b> δικιά του εικόνα από τον υπολογιστή του. Η διαδικασία δεν είναι και τόσο απλή καθώς τελικά χρειάστηκε να γίνει η χρήση της λειτουργίας <i>replaceChild</i> της <b>JavaScript</b>.</p></div><div data-bbox="112 852 889 891" data-label="Text"><p>Τα πεδία χαρακτηρίζονται από τα χρώματα που τα χαρακτηρίζουν. Περισσότερα για τα χρώματα θα αναλύσουμε στο επόμενο υποκεφάλαιο.</p></div><div data-bbox="456 937 541 966" data-label="Page-Footer"><hr data-bbox="145 930 855 935"/> { 55 }</div>
```

## 4.9 search.php

Το αρχείο search.php αρχικά περιέχει κώδικα παρόμοιο με του αρχείου add.php. Η επιλογή κατηγορίας και υποκατηγορίας γίνεται με παρόμοιο τρόπο, βέβαια στον συγκεκριμένο ο κώδικας επιτρέπει κενές εκφράσεις και αναζήτηση σε όλες τις κατηγορίες ή υποκατηγορίες. Χαρακτηριστικό του κώδικα είναι το παρακάτω JavaScript:

```
<script type="text/javascript">  
  
function showtotal(str)  
  
{ document.getElementById("totalres").innerHTML="Σύνολο: "+str; }  
  
</script>
```

Ο συγκεκριμένος κώδικας συλλέγει όλους τους αριθμούς που αποτελούν το σύνολο των αποτελεσμάτων που βρέθηκαν και τους προσθέτει, ώστε να βγάλει το τελικό αποτέλεσμα. Σαφώς επειδή ο κώδικας περιέχει δυνατότητα επεξεργασίας περιέχει την συνάρτηση *learnread* όπως επίσης την συνάρτηση *showtype* μαζί με τις βοηθητικές της. Εδώ συναντάμε και την συνάντηση *genius* η οποία δέχεται ως παραμέτρους τις ιδιότητες του πεδίου και το εμφανίζει ανάλογα. Η μεταβλητή *therightclass* δηλώνει το χρώμα που θα έχει το όνομα ενός πεδίου λόγω των ιδιοτήτων του, επίσης κάθε ιδιότητα έχει το δικό της σήμα κάτω από το όνομα του πεδίου τοποθετημένο με JavaScript, ώστε να εμφανίζεται μόνο όταν το δείκτης του χρήστη πάει πάνω από το πεδίο. Το υποχρεωτικό συμβολίζεται με (M) και χρώμα κόκκινο, το μοναδικό με (U) και χρώμα μπλε, το πεδίο που δηλώνει ότι θα εκτυπωθεί στην QR ετικέτα με (Q) και χρώμα πράσινο, και τέλος το δημόσιο συμβολίζεται με (P) και χρώμα βιολετί.

Επίσης για κάθε αντικείμενο υπάρχει και το αντίστοιχο checkbox το οποίο δημιουργείται από τον παρακάτω κώδικα:

```
echo '<td><input type="checkbox" name="pt[]" value="'. $trueid. "'></td>';
```

Ανάλογα με το ποια αντικείμενα θα επιλέξει ο χρήστης, στέλνεται και η αντίστοιχη παράμετρος στην σελίδα qrcode.php που θα αναλύσουμε στην συνέχεια.



## 4.10 qrcode.php & showconfs.php

Η σελίδα qrcode.php συνεργάζεται με την σελίδα pdf.php στην οποία στέλνει τις κατάλληλες παραμέτρους, ώστε αυτή να εκτυπώσει με την σειρά της τις QR ετικέτες. Η σελίδα αυτή χαρακτηρίζεται από τον συνεχή έλεγχο των τιμών που εισχωρούνται από τον χρήστη με την χρήση κώδικα JavaScript. Συγκεκριμένα το script αρχίζει από την γραμμή 41 όπου λαμβάνει τις τιμές, τις ελέγχει και ανάλογα παράγει το κατάλληλο μήνυμα. Για την ανάκτηση των ήδη υπάρχοντων ρυθμίσεων χρησιμοποιείται η βοηθητική σελίδα showconfs.php.

Καθώς πρέπει να υπάρχουν τα στοιχεία, ώστε να γίνει η τοποθέτηση των τιμών στα πεδία, η σελίδα showconfs.php εξάγει τα δεδομένα σε μορφή div που δεν είναι ορατή, και εμφανίζει 2 κουμπιά τα οποία ενεργοποιούν τα functions που είναι δηλωμένα στην qrcode.php. Εδώ παρατηρείτε σε τι βαθμό μπορούν να συνεργαστούν 2 δυναμικές σελίδες όπου η μία συμπληρώνει την άλλη. Παρόμοιους συνδυασμούς έχουμε αναφέρει ξανά σε προηγούμενο κεφάλαιο αλλά ο συγκεκριμένος ήταν από τους πιο δύσκολους καθώς απαιτεί στοιχεία τις άλλης σελίδας αντί του απλά να δίνει κάποιες πληροφορίες.

Στον παρακάτω κώδικα φαίνεται η δυναμικότητα της σελίδας, ανάλογα με την προέλευση του χρήστη:

```
if(isset($_GET['sub'])){  
...  
}else{  
?>  
  
    <label> <span>Κατηγορία :</span>  
...  

```

Ο κώδικας αυτός φανερώνει πως η επιλογή κατηγορίας, μαζί με την επιλογή υποκατηγορίας δεν εμφανίζεται στον χρήστη αν υπάρχει είσοδος από την σελίδα search.php. Αντί αυτού, το σύστημα βάζει κρυφά πεδία τα οποία με την σειρά τους θα λειτουργήσουν ως είσοδο στο αρχείο pdf.php.

## 4.11 pdf.php

Το αρχείο αυτό παράγει το pdf με τις QR ετικέτες. Για να γίνει αυτό χρησιμοποιήθηκε η τεχνολογία FPDF η οποία εισάγεται στο σύστημα. Αξιοσημείωτο είναι πως μόνο το 20% του κώδικα σε αυτήν την σελίδα ασχολείται με την εκτύπωση των ετικετών χρησιμοποιώντας την βιβλιοθήκη της FPDF. Το υπόλοιπο 80% ασχολείται με την ανάκτηση λεπτομερειών από την βάση όπως επίσης και προετοιμάζει το έδαφος για την δημιουργία του pdf αφού βεβαίως κάνει έλεγχο ξανά των διαστάσεων που της ζητήθηκε να έχει.

Η προ-εγκατάσταση της σελίδας υπάρχει στον παρακάτω κώδικα:

```
$pdf = new PDF();  
  
$pdf->SetLeftMargin($margin);  
  
$pdf->SetRightMargin($margin);  
  
$pdf->SetTopMargin($margin);  
  
$pdf->AddPage();  
  
$pdf->SetAutoPageBreak(true,$margin);  
  
$pdf->AddFont('DxArial','', 'arial.php');
```

Για να εμφανίζονται κατάλληλα οι ελληνικοί χαρακτήρες δημιουργήθηκε με τα εργαλεία της FPDF κατάλληλη γραμματοσειρά η οποία εισάχθηκε στον server<sup>j</sup>. Έπειτα παρατηρείται πως ο κώδικας έχει διάφορων ειδών ευελιξίες και διαμορφώνει κατάλληλα τις ετικέτες για καλύτερη οπτική παρουσίαση. Να σημειωθεί πως η εικόνα της QR ετικέτας έχει ορισθεί στην μέγιστη ποιότητα για να βεβαιωθεί πως ακόμα και να αλλοιωθεί η ετικέτα, ο κώδικας θα είναι πιθανώς ακόμα δυνατόν να σαρωθεί.

---

<sup>j</sup> αρχεία arial.php & arial.z

## 4.12 watchers.php & watchershelper.php

Οι παραχωρήσεις αντικειμένων είναι αποτέλεσμα του αρχείου `watchers.php` με το βοηθητικό αρχείο `watchershelper.php`. Το βοηθητικό αρχείο όχι μόνο βοηθάει στην εμφάνιση της σελίδας πρόσφατων παραχωρήσεων, αλλά παρέχει και την δυνατότητα εμφάνισης των στοιχείων του ατόμου που έχει γίνει η παραχώρηση. Είναι με λίγα λόγια μια βελτιωμένη έκδοση του βοηθητικού αρχείου `showconfs.php` που είδαμε σε προηγούμενο υποκεφάλαιο.

Σε αυτό το αρχείο γίνεται η χρήση της μεταβλητής `session "page"` η οποία και δηλώνει την σελίδα στην οποία βρίσκεται ο χρήστης στις πρόσφατες παραχωρήσεις. Ανάλογα με αυτήν την τιμή το βοηθητικό αρχείο επιστρέφει την ανάλογη σελίδα με παραχωρήσεις.

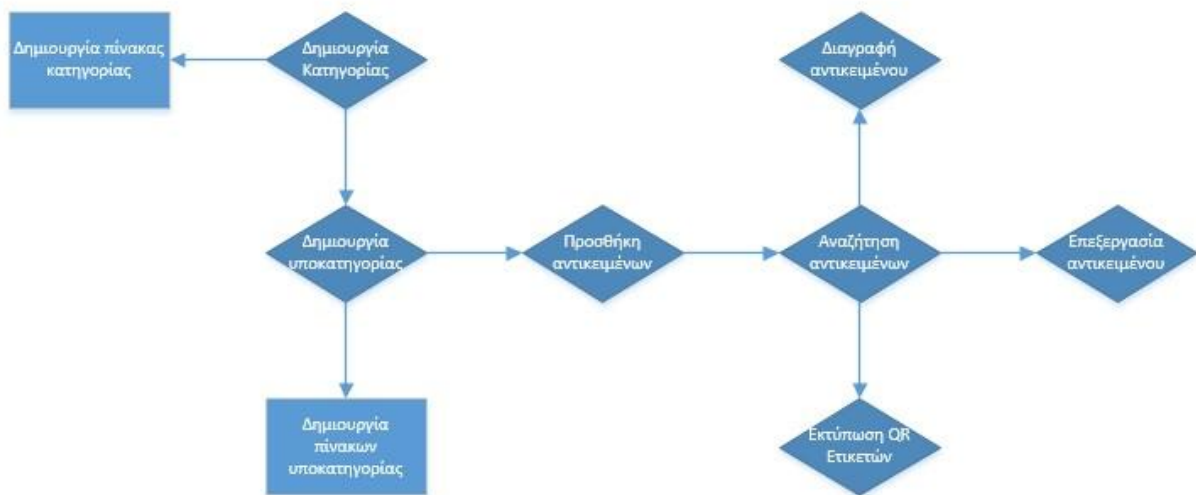
Χαρακτηριστικό των αρχείων αυτών είναι ο τρόπος με τον οποίον γίνεται η σύγκριση των 2 ημερομηνιών. Ο κώδικας με τον οποίον γίνεται αυτή η πράξη είναι ο εξής:

```
date_default_timezone_set('UTC');  
  
if(strtotime($r['return'])>strtotime($r['gavaway']))
```

Η πρώτη γραμμή δεν είναι απαραίτητη αλλά εμφανίζει μήνυμα σφάλματος ότι πρέπει να δηλωθεί. Ουσιαστικά η δεύτερη γραμμή αποτελεί την σύγκριση των 2 ημερομηνιών. Η κάθε ημερομηνία μετατρέπεται από αλφαριθμητικό σε έναν αριθμό που συμβολίζει αυτήν την ημερομηνία και έπειτα οι 2 αυτές τιμές συγκρίνονται και ανάλογα προχωράει ο κώδικας.

## Κεφάλαιο 5: Οδηγίες χρήσης

Στο κεφάλαιο αυτό αναλύονται όλες οι δυνατότητες που έχουν οι χρήστες και οι διαχειριστές στο σύστημα. Το κεφάλαιο αρχίζει με τα βήματα που πρέπει να ακολουθήσει κάποιος για να εγκαταστήσει το σύστημα σε κάποιον server, και μετά ακολουθούν, με την σειρά που εμφανίζονται στο μενού, διευκρινήσεις για κάθε τομέα του ιστοχώρου. Στην Εικόνα 7 φαίνεται το διάγραμμα χρήσης του συστήματος όσον αφορά την διαχείριση των αντικειμένων.



Εικόνα 7: Διάγραμμα χρήσης

## 5.1 Οδηγίες Εγκατάστασης

Αρχικά πρέπει όλα τα αρχεία του συστήματος (Εικόνα 2) να ανέβουν στο φάκελο `html_public` του `server`. Τα δικαιώματα<sup>k</sup> πρέπει να είναι όλα 0755 εκτός από τους φακέλους `upload`, `files`, `temp` και `phpcode` που βρίσκεται στον φάκελο `"includes"` οι οποίοι θέλουν 0777. Έπειτα θα πρέπει να επεξεργαστεί το αρχείο `config.php` που βρίσκεται στον φάκελο `includes`. Στην Εικόνα 8 φαίνεται πως περίπου πρέπει να είναι το περιεχόμενο του `config.php`. Αναλυτικότερα, οι σταθερές του έχουν ως εξής<sup>l</sup>:

- `MYSQLHOST`: Ορίζεται ο MySQL host που έχει η βάση δεδομένων. Παράδειγμα σωστής παραμετροποίησης αν χρησιμοποιείται `socket` είναι η: `/zstorage/home/icctest00368/mysql/run/mysql.sock`
- `MYSQLUSER`: Ορίζεται ο χρήστης της βάσης που θα χρησιμοποιηθεί. Ο χρήστης πρέπει να έχει όλα τα δικαιώματα, ώστε να λειτουργήσει σωστά η βάση. Παράδειγμα χρήσης του διαχειριστή της βάσης δεδομένων: `root`
- `MYSQLPASS`: Ορίζεται ο κωδικός του χρήστη που δηλώσατε παραπάνω και θα χρησιμοποιείται για την σύνδεση στην βάση δεδομένων.
- `MYSQLPREF`: Ορίζεται το πρόθεμα που θα έχουν οι πίνακες της βάσης, ώστε να μην έρχονται σε σύγκρουση με άλλες βάσεις που υπάρχουν πιθανώς στον `server`. Για παράδειγμα αν μπει η παράμετρος `"dx"`, τότε θα δημιουργηθεί ο πίνακας `"dxusers"` αντί για `"users"`.
- `MYSQLBASE`: Ορίζεται το όνομα της βάσης δεδομένων που θα χρησιμοποιηθεί.
- `MYSQLHTYP`: Ορίζεται ο τύπος της σύνδεσης της βάσης δεδομένων. Παράδειγμα σωστής παραμετροποίησης αν η βάση MySQL λειτουργεί με `sockets` είναι: `unix_socket`
- `HTMLHOME`: Ορίζεται ο σύνδεσμος που οδηγεί στην αρχική σελίδα του συστήματος. Συγκεκριμένα στον `server` `zafora` χρησιμοποιήθηκε η παρακάτω ρύθμιση: `zafora.icte.uowm.gr/~icctest00368/`
- `DISPERROR`: Φανερώνει διάφορα σφάλματα κώδικα αν υπάρχουν. Χρησιμοποιείται κυρίως για ανάγκες `debugging`.
- `HELPON`: Αν είναι `TRUE` εμφανίζει βοηθητικά κείμενα στο τέλος της σελίδας.

Στην συνέχεια ανοίγουμε σε κάποιον `browser` το `install.php`. Αν το `config.php` έχει οριστεί σωστά τότε δεν εμφανίζονται σφάλματα ενδιάμεσα στα μηνύματα κατά την διάρκεια

<sup>k</sup> `Chmod`

<sup>l</sup> Δίνονται παραδείγματα για τον `server` του τμήματος `zafora.icte.uowm.gr`

της εγκατάστασης της βάσης δεδομένων. Αν υπάρχει το οποιοδήποτε σφάλμα, η εγκατάσταση πρέπει να επαναληφθεί αφού πρώτα διορθωθεί το αρχείο config.php. Μόλις ολοκληρωθεί η εγκατάσταση το αρχείο install.php πρέπει είτε να διαγραφεί, είτε να γίνει μη προσπελάσιμο από τρίτους.

Αμέσως μόλις ολοκληρωθεί η εγκατάσταση πρέπει να γίνει είσοδος με όνομα χρήστη root και κενό κωδικό. Το σύστημα αυτομάτως θα απαιτήσει αλλαγή κωδικού και όταν ολοκληρωθεί αυτή η ενέργεια ο διαχειριστής θα πρέπει να εισέλθει με τα κανονικά του στοιχεία αν θέλει να αλλάξει τα λοιπά στοιχεία του από τις ρυθμίσεις.

Στην συνέχεια θα αναλυθεί η κάθε λειτουργία και δυνατότητα του συστήματος με την σειρά που εμφανίζονται στο κυρίως μενού.

```
define("MYSQLHOST", "/zstorage/home/icctest00368/mysql/run/mysql.sock");
define("MYSQLUSER", "root");
define("MYSQLPASS", "526052");
define("MYSQLPREF", "dx");
define("MYSQLBASE", "diploma");
define("MYSQLHTYP", "unix_socket");
define("HTMLHOME", "zafora.icte.uowm.gr/~icctest00368/");
define("DISPERROR", "1");
define("HELPON", TRUE);
```

Εικόνα 8: Παράδειγμα περιεχομένου του αρχείου config.php που χρησιμοποιήθηκε στον server "zafora".

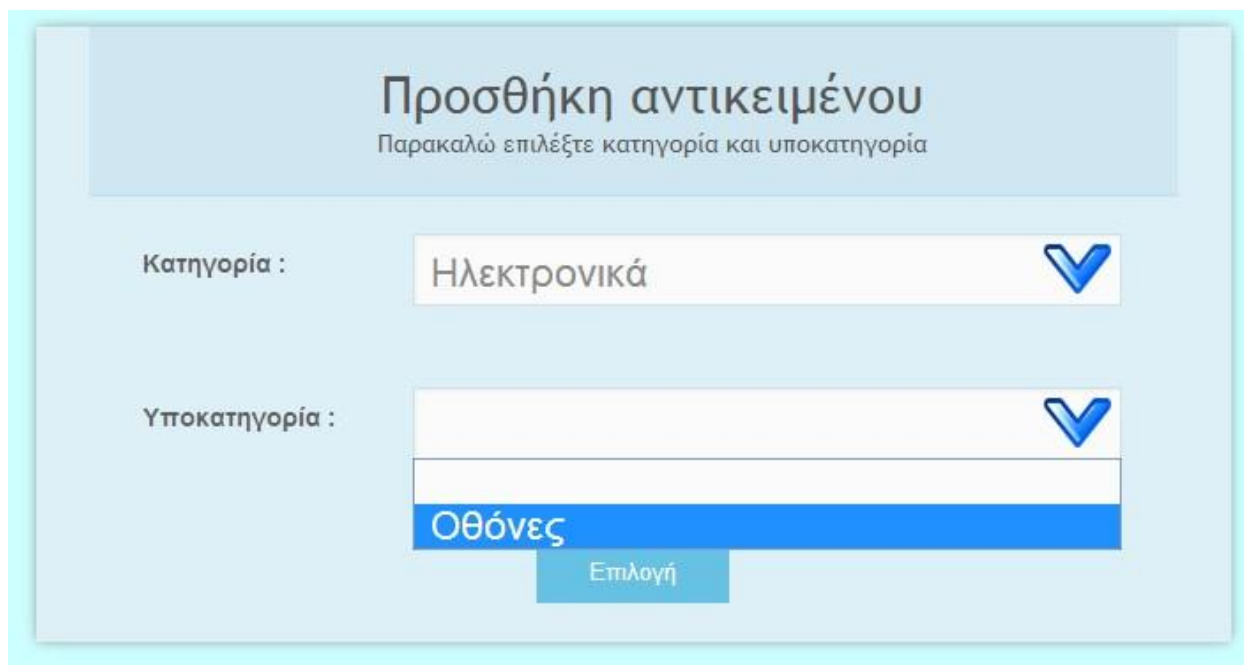
## 5.2 Προσθήκη Αντικειμένου

Αυτός ο τομέας αφορά την προσθήκη αντικειμένων στην βάση δεδομένων. Για να μπορέσει κάποιος να μεταβεί στις σχετικές ενέργειες θα πρέπει να έχει και το δικαίωμα προσθήκης αντικειμένων στην βάση. Επίσης πρέπει να υπάρχει τουλάχιστον μία κατηγορία η οποία να έχει τουλάχιστον μία υποκατηγορία που με την σειρά της να έχει τουλάχιστον ένα πεδίο. Ο χρήστης επιλέγει από το μενού την κατηγορία που θέλει να προσθέσει το αντικείμενο και τότε αυτόματα εμφανίζεται μία ακόμα λίστα η οποία περιέχει τις υποκατηγορίες της κατηγορίας που επέλεξε ο χρήστης. Αν ο χρήστης αλλάξει κατηγορία, η λίστα ανανεώνεται ανάλογα. Για να μπορέσει ο χρήστης να προχωρήσει στην προσθήκη του αντικειμένου, πρέπει να έχει επιλέξει και κατηγορία και υποκατηγορία. Επιλέγοντας το κουμπί "επιλογή" οριστικοποιεί την επιλογή του και μεταβαίνει να συμπληρώσει τα στοιχεία του αντικειμένου που επιθυμεί να προσθέσει. Η Εικόνα 9 παρουσιάζει ένα παράδειγμα επιλογής.

Αφού πατήσει το κουμπί επιλογή, εμφανίζεται μια φόρμα με όλα τα στοιχεία που έχει ορίσει ο χρήστης για αυτή την υποκατηγορία, όπως φαίνεται στην Εικόνα 10. Το σύστημα τροποποιεί τα χρώματα της ονομασίας των πεδίων, ανάλογα με το αν το πεδίο είναι υποχρεωτικό, μοναδικό, αν θα είναι στην QR ετικέτα και αν είναι δημόσιο. Τα χρώματα είναι κόκκινο, μπλε, πράσινο και βιολετί με την αντίστοιχη σειρά. Αν κάποιο πεδίο έχει παραπάνω από μία τέτοιες ιδιότητες, τότε η σειρά προτίμησης είναι η σειρά που αναφέρθηκαν προηγουμένως. Επίσης οι ιδιότητες που πρέπει να έχει το πεδίο γίνονται εύκολα αντιληπτές γιατί εμφανίζονται κάτω από το όνομα καθώς το ποντίκι βρίσκεται πάνω από το επιθυμητό πεδίο.

Σε περίπτωση που υπάρχει πεδίο εικόνας, ο χρήστης έχει την επιλογή να διαλέξει αν θα βάλει εικόνα από κάποια εξωτερική πηγή ή αν θα ανεβάσει ο ίδιος κάποια εικόνα από τον υπολογιστή του. Στην δεύτερη περίπτωση η εικόνα πρέπει να είναι gif, jpeg, bmp, jpg, ή png για να τις δεχθεί το σύστημα. Αν το πεδίο είναι αρχείο, τότε το μόνο αποδεκτό αρχείο που μπορεί να ανεβάσει είναι zip.

Αν ο χρήστης πληκτρολογήσει κάτι που δεν είναι αποδεκτό από το σύστημα, εμφανίζεται το κατάλληλο μήνυμα λάθους και δίνεται η επιλογή στον χρήστη να επιστρέψει στην φόρμα και να αλλάξει το λάθος του, χωρίς βέβαια να χρειαστεί να συμπληρώσει ξανά τα υπόλοιπα πεδία καθώς αυτά έχουν αποθηκευτεί προσωρινά στην μνήμη του περιηγητή.



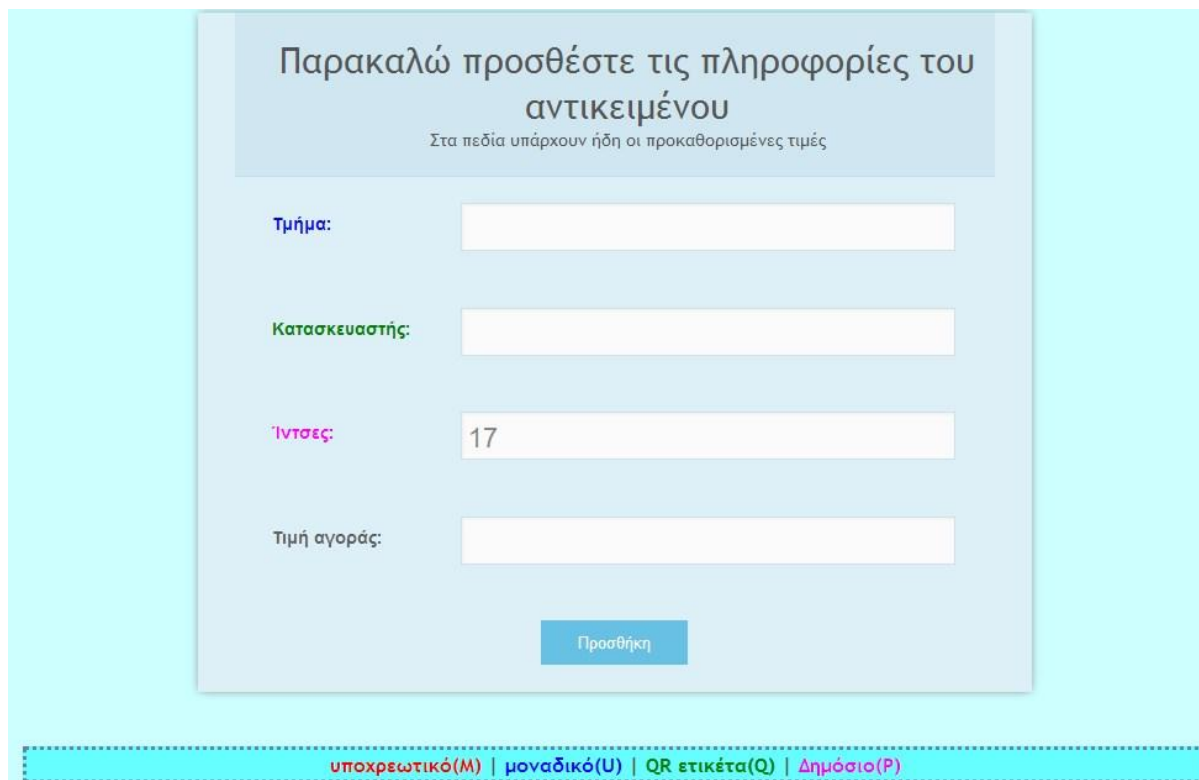
**Προσθήκη αντικειμένου**  
Παρακαλώ επιλέξτε κατηγορία και υποκατηγορία

Κατηγορία : Ηλεκτρονικά

Υποκατηγορία : Οθόνες

Επιλογή

Εικόνα 9: Επιλογή υποκατηγορίας



**Παρακαλώ προσθέστε τις πληροφορίες του αντικειμένου**  
Στα πεδία υπάρχουν ήδη οι προκαθορισμένες τιμές

Τμήμα:

Κατασκευαστής:

Ίντσες: 17

Τιμή αγοράς:

Προσθήκη

υποχρεωτικό(M) | μοναδικό(U) | QR ετικέτα(Q) | Δημόσιο(P)

Εικόνα 10: Προσθήκη στοιχείων αντικειμένου



## 5.3 Εύρεση Αντικειμένου

Ο τομέας αυτός βοηθά τον χρήστη να βρει ένα ή περισσότερα αντικείμενα που αναζητά. Όπως έχουμε αναφέρει σε προηγούμενο κεφάλαιο, χρειάζεται να έχει το κατάλληλο δικαίωμα προκειμένου να μπορεί να κάνει αναζήτηση αντικειμένου. Όπως και στην προθήκη αντικειμένου το αρχικό μενού αποτελείται αρχικά μόνο με την λίστα των κατηγοριών, πλην όμως, ο χρήστης στην αναζήτηση δεν χρειάζεται να κάνει αναζήτηση ούτε σε κάποια συγκεκριμένη υποκατηγορία, ούτε σε κάποια κατηγορία. Ανάλογα με την επιλογή που θα κάνει ο χρήστης εμφανίζονται και τα ανάλογα πεδία. Σε περίπτωση που ο χρήστης επιλέξει να κάνει αναζήτηση σε μία ή όλες τις κατηγορίες, τότε τα πεδία που εμφανίζονται είναι οι κοινές μεταβλητές των υποκατηγοριών των κατηγοριών που επιλέχθηκαν.

Όταν ο χρήστης επιλέξει σε ποια περιοχή επιθυμεί να κάνει αναζήτηση, η φόρμα που εμφανίζεται είναι παρόμοια πάλι με αυτήν στην προσθήκη αντικειμένου. Πάλι υπάρχουν χρώματα, σημειώσεις όταν ο χρήστης βάζει το δείκτη του ποντικιού του πάνω από το πεδίο και τα σχετικά. Βέβαια ο χρήστης δεν υποχρεούται να συμπληρώσει κανένα πεδίο, μπορεί να συμπληρώσει όσα επιθυμεί. Αν δεν συμπληρώσει κανένα πεδίο και επιλέξει το κουμπί "αναζήτηση" τότε ως αποτέλεσμα θα έχει όλα τα αντικείμενα της υποκατηγορίας, ή των υποκατηγοριών που επέλεξε.

Μετά την εκτέλεση της αναζήτησης τα αποτελέσματα χωρίζονται ανά υποκατηγορία. Ένα παράδειγμα φαίνεται στην Εικόνα 11. Υπάρχει δείκτης των συνολικών αποτελεσμάτων, όπως και δείκτης των αποτελεσμάτων ανά κατηγορία. Εκτός από τις λεπτομέρειες κάθε αντικειμένου, δίνονται στον χρήστη οι εξής 3 επιλογές:

- **Εκτύπωση:** Η εκτύπωση αποτελείται από πολλά κουτάκια επιλογής στα οποία ο χρήστης επιλέγει τα αντικείμενα στα οποία θέλει να εκτυπώσει QR ετικέτες. Το σύστημα καταγράφει την επιλογή του και επιλέγοντας το κουμπί "εκτύπωση" στέλνονται οι κατάλληλες παράμετροι στην σελίδα για εκτύπωση ετικετών. Περισσότερα για τις QR ετικέτες θα αναλυθούν στο υποκεφάλαιο 5.4
- **Επεξεργασία:** Δίνεται η δυνατότητα στον χρήστη να αλλάξει κάποιες πληροφορίες του αντικειμένου. Φυσικά για να μεταβεί σε μια τέτοια πράξη χρειάζεται περαιτέρω δικαιώματα. Υπάρχει παράδειγμα επεξεργασίας στην Εικόνα 12
- **Διαγραφή:** Μέσω αυτής τις επιλογής ο χρήστης μπορεί να αφαιρέσει ένα αντικείμενο από την βάση. Αυτή η επιλογή θέλει επιβεβαίωση (όπως και κάθε διαγραφή) και είναι οριστική και μη αναστρέψιμη

## Αποτελέσματα

Σύνολο: 3

Ηλεκτρονικά > Οθόνες (βρέθηκαν 3)

ID	Τμήμα	Κατασκευαστής	Ίντσες	Τιμή αγοράς	Εκτύπωση	Επεξεργασία	Διαγραφή
10	Θεσσαλονίκης	Samsung	15	102.25	<input type="checkbox"/>	<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>
11	Αθήνας	LD	17	170.25	<input type="checkbox"/>	<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>
29	Λαμίας	Samsung	20	199.99	<input type="checkbox"/>	<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>

Εκτύπωση

Εικόνα 11: Παράδειγμα αναζήτησης αντικειμένου

### Επεξεργασία αντικειμένου

Αν δεν επιθυμείτε να αλλάξετε κάτι, αφήστε το όπως είναι

Τμήμα:

Κατασκευαστής:

Ίντσες:

Τιμή αγοράς:

Εικόνα 12: Παράδειγμα επεξεργασίας αντικειμένου

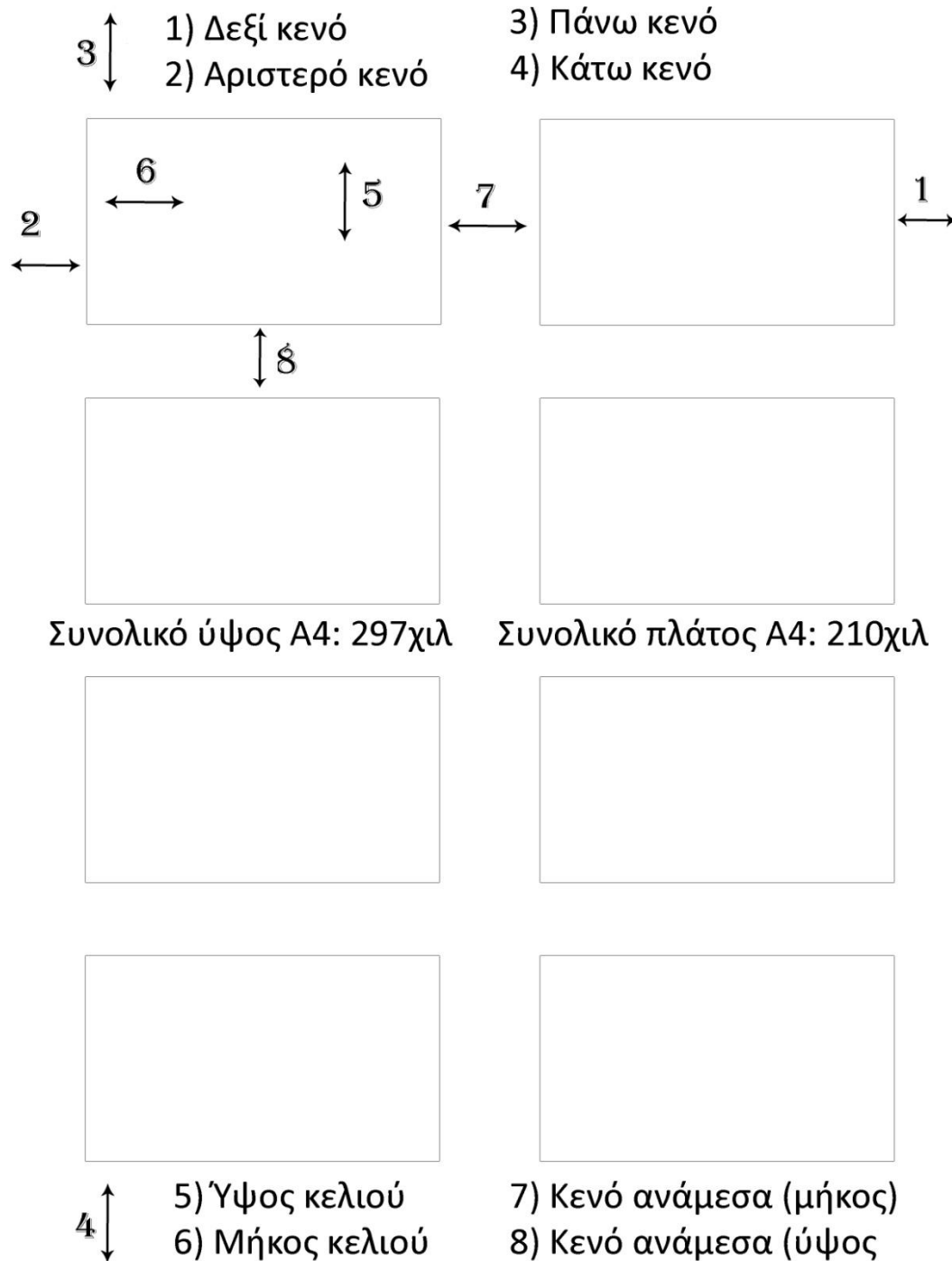
## 5.4 Εκτύπωση Ετικετών

Σε αυτόν τον τομέα μπορεί ο χρήστης να μεταβεί με 2 τρόπους. Ο ένας είναι να μεταβεί επιλέγοντας το "Εκτύπωση Ετικετών" από το μενού και η δεύτερη είναι μέσω της εύρεσης αντικειμένων, πράγμα το οποίο αναφέραμε στο προηγούμενο κεφάλαιο. Αν ο χρήστης μεταβεί σε αυτόν τον τομέα από το μενού τότε έχει την επιλογή να επιλέξει αν θα εκτυπωθούν ετικέτες από μία ή περισσότερες κατηγορίες ή από μία ή περισσότερες υποκατηγορίες. Στην δεύτερη περίπτωση το σύστημα έχει ήδη μαρκάρει από μόνο του τα αντικείμενα για εκτύπωση δεν δίνεται κάποια επιλογή στον χρήστη για άλλη κατηγορία/υποκατηγορία.

Η εκτύπωση των ετικετών γίνεται με αρκετή ελευθερία του χρήστη να ρυθμίσει ακριβώς τις ετικέτες, ώστε να εφαρμόζουν απόλυτα στα αυτοκόλλητα. Η Εικόνα 13 βοηθάει τον χρήστη να καταλάβει πως γίνεται η χωροθέτηση των ετικετών. Ο χρήστης υποχρεούται να βάλει τιμές οι οποίες να εφαρμόζουν ακριβώς ανά σελίδα. Για αυτόν τον λόγο, σε πραγματικό χρόνο ενώ συμπληρώνει τις τιμές γίνεται έλεγχος αν έχει βάλει κατάλληλες τιμές και εμφανίζεται το σχετικό μήνυμα ανάλογα με το αν οι τιμές ταιριάζουν στις διαστάσεις μιας σελίδας ή όχι. Για μεγαλύτερη ευκολία, ο χρήστης μπορεί να αποθηκεύσει τις ρυθμίσεις του, ώστε σε μελλοντική εκτύπωση ετικετών τα πεδία να συμπληρώνονται αυτόματα από το σύστημα.

Υπάρχει επίσης ένα ειδικό πεδίο που αν το συμπληρώσει ο χρήστης εμφανίζεται σε κάθε ετικέτα ασχέτως με τις πληροφορίες των αντικειμένων. Αυτό μπορεί να βοηθήσει τον χρήστη να συμπληρώσει κάτι επιπλέον που μπορεί να θέλει να εμφανίζεται στην ετικέτα ενώ κανονικά δεν θα εμφανιζόταν. Αν δεν συμπληρώσει τίποτα στο συγκεκριμένο πεδίο, το σύστημα αγνοεί αυτήν την επιλογή. Όσον αφορά τα πεδία που έχουν επισημανθεί για εκτύπωση στην QR ετικέτα, αν είναι λίγα τότε το σύστημα βάζει μεγάλο φόντο, εφόσον υπάρχει πολύς χώρος, αν όμως είναι πολλά, τότε χρησιμοποιείται μικρότερο φόντο έτσι ώστε να ξεχωρίζουν το ένα από το άλλο. Φυσικά αν υπάρχουν πάρα πολλές πληροφορίες ή το μήκος τους είναι υπερβολικά μεγάλο, υπάρχει κίνδυνος οι πληροφορίες να ξεπεράσουν το μέγεθος των QR ετικετών.

Το τελικό αποτέλεσμα είναι ένα pdf αρχείο που παράγεται από το σύστημα και ο χρήστης μπορεί να το αποθηκεύσει και να το στείλει για εκτύπωση. Κάθε αυτοκόλλητο περιέχει τις πληροφορίες που έχει επιλέξει ο χρήστης να εκτυπωθούν, το link που οδηγεί στην σελίδα που περιέχει τις πληροφορίες αυτού του αντικειμένου, και φυσικά το QR code που οδηγεί στο ίδιο μέρος.



Εικόνα 13: Οδηγός χωροθέτησης ετικετών για εκτύπωση

## 5.5 Χρήστες/Ομάδες

Ο τομέας αυτός είναι προσπελάσιμος μόνο από τους διαχειριστές του συστήματος και προσφέρει τις εξής 4 δυνατότητες:

- Εύρεση χρήστη
- Εύρεση ομάδας
- Δημιουργία χρήστη
- Δημιουργία ομάδας

Στην εύρεση χρήστη ο διαχειριστής μπορεί να συμπληρώσει όσα πεδία επιθυμεί. Αν δεν συμπληρώσει κανένα πεδίο τότε θα του εμφανιστεί μια λίστα με όλους τους χρήστες του συστήματος. Στην λίστα των αποτελεσμάτων ο διαχειριστής έχει δύο επιλογές. Η πρώτη είναι να επεξεργαστεί τα στοιχεία του χρήστη και όλες τις ιδιότητές του, και η δεύτερη είναι να διαγράψει τον χρήστη. Αν ο διαχειριστής επιλέξει να απενεργοποιήσει κάποιον χρήστη, αυτός ο χρήστης δεν θα έχει πρόσβαση στο σύστημα και ο μόνος τρόπος για να την αποκτήσει είναι να τον ενεργοποιήσει με τον ίδιο τρόπο κάποιος διαχειριστής. Ο μόνος χρήστης που δεν μπορεί να πειράξει κανείς είναι ο πρώτος χρήστης που δημιουργείται κατά την εγκατάσταση. Ο διαχειριστής μπορεί επίσης να αναγκάσει τον χρήστη να αλλάξει τον κωδικό του. Ένα παράδειγμα μιας τέτοιας ενέργειας υπάρχει στην Εικόνα 14.

Η εύρεση ομάδας λειτουργεί με παρόμοιο τρόπο· οι διαχειριστές μπορούν πάλι να αφήσουν όσα πεδία επιθυμούν κενά και αν τα αφήσουν όλα εμφανίζονται σε αυτούς όλες οι ομάδες. Δίνεται και εδώ η δυνατότητα επεξεργασίας και διαγραφής κάποιου γκρουπ. Να σημειωθεί όμως πως ούτε εδώ μπορεί κάποιος να αλλάξει τα στοιχεία του πρώτου γκρουπ (Χρήστες).

Για την δημιουργία χρήστη συμπληρώνεται μόνο το όνομα χρήστη και το e-mail. Το σύστημα ρυθμίζει μόνο του τις υπόλοιπες παραμέτρους και στέλνεται ένα e-mail στον χρήστη με έναν προσωρινό κωδικό τον οποίο θα πρέπει υποχρεωτικά να αλλάξει κατά την είσοδό του στο σύστημα.


Στην δημιουργία ομάδας οι διαχειριστές πρέπει να δώσουν ένα όνομα σε αυτήν που να αποτελείται τουλάχιστον από 3 χαρακτήρες και αν θέλουν μπορούν να δώσουν και μία περιγραφή, ώστε να ξέρουν τον λόγο ύπαρξης αυτής της ομάδας. Κάθε νέα ομάδα δεν έχει κανένα χρήστη και κανένα δικαίωμα, αυτά πρέπει να προστεθούν από τους διαχειριστές μετά

την δημιουργία ομάδας. Περισσότερες πληροφορίες σχετικά με την δυνατότητα παραχώρησης δικαιωμάτων υπάρχουν στο υποκεφάλαιο 5.7.

**Αλλαγή Ρυθμίσεων Χρήστη**  
Αν δεν επιθυμείτε να αλλάξετε κάτι, αφήστε το όπως είναι

Όνομα χρήστη:

Κωδικός:

Ομάδα:  

E-mail:

Ενεργοποιημένος:

Εξανάγκασε την αλλαγή κωδικού:

Εικόνα 14: Παράδειγμα εξαναγκασμού χρήστη να αλλάξει τον κωδικό του.

## 5.6 Διαχείριση Δομής

Αυτόν τον τομέα μπορούν να τον δουν μόνο οι διαχειριστές του συστήματος. Αφορά την διαχώριση της δομής της βάσης δεδομένων. Κάθε ενέργεια επιφέρει αρκετές αλλαγές στους πίνακες της βάσης δεδομένων καθώς προστίθενται, αφαιρούνται και επεξεργάζονται αρκετά στοιχεία ανάλογα με την επιθυμία των διαχειριστών. Συγκεκριμένα η διαχείριση δομής προσφέρει 2 δυνατότητες: Διαχείριση Κατηγοριών και Διαχείριση Κοινών Στοιχείων.

Τα κοινά στοιχεία είναι πεδία τα οποία εκχωρούνται αυτόματα σε κάθε νέα υποκατηγορία της κατηγορίας που έχουν οριστεί. Κάθε κοινό στοιχείο μπορεί να καλύπτει μία ή όλες τις κατηγορίες. Με αυτόν τον τρόπο ο χρήστης μπορεί να προσθέσει ένα πεδίο σε ολόκληρη την βάση δεδομένων, χωρίς ιδιαίτερο κόπο. Στην κύρια σελίδα εμφανίζεται η λίστα με όλα τα κοινά στοιχεία και στην συνέχεια η φόρμα για υποβολή νέου κοινού στοιχείου. Για την δημιουργία κοινού στοιχείου έχουμε τα εξής πεδία:

- **Κατηγορία:** Επιλέγεται αν το κοινό στοιχείο θα εφαρμόζεται σε κάποια συγκεκριμένη κατηγορία, ή σε όλες.
- **Όνομα:** Εδώ ορίζεται το όνομα που θα έχει το πεδίο που θα δημιουργείται.
- **Τύπος:** Δίνεται μια λίστα με τους διαθέσιμους τύπους που μπορεί να είναι το πεδίο. Ανάλογα με την επιλογή, κάποια επόμενα στοιχεία μπορεί να μην χρειάζονται οπότε και ακυρώνονται αυτόματα.
- **Μήκος:** Το μήκος ορίζει το μέγιστο αριθμό χαρακτήρων που μπορεί να έχει το πεδίο.
- **Προεπιλεγμένη τιμή:** Στην προσθήκη αντικείμενου η αρχική φόρμα συμπληρώνεται αυτόματα με την προεπιλεγμένη τιμή που έχει οριστεί.
- **Δημόσιο:** Αν επιλεγεί, η πληροφορία του πεδίου θα είναι ορατή προς όλους.
- **Υποχρεωτικό:** Αν επιλεγεί, απαγορεύει τον χρήστη από το να αφήσει το συγκεκριμένο πεδίο κενό
- **Μοναδικό:** Αν επιλεγεί, δεν επιτρέπει στον χρήστη να βάλει στο πεδίο κάποια τιμή που ήδη υπάρχει σε άλλο αντικείμενο της υποκατηγορίας αυτής.
- **Ετικέτα QR:** Αν επιλεγεί, η πληροφορία αυτή θα εκτυπωθεί στο αυτοκόλλητο ως επιπρόσθετη πληροφορία για το αντικείμενο αυτό.

Ένα παράδειγμα υπάρχει στην Εικόνα 15. Κάθε νέο κοινό στοιχείο δεν εφαρμόζεται αυτόματα σε καμία υποκατηγορία, μόνο σε μελλοντικές που μπορεί δημιουργηθούν. Ο χρήστης έχει βέβαια την δυνατότητα να επιλέξει από την λίστα να εφαρμόσει το κοινό αυτό

στοιχείο σε όσες υποκατηγορίες δεν έχει ακόμα εφαρμοστεί. Οι άλλες 2 επιλογές που του παρέχει η λίστα είναι η επεξεργασία του κοινού στοιχείου και η διαγραφή του. Και οι 2 αυτές ενέργειες δεν επηρεάζουν τα πεδία που έχουν ήδη εκχωρηθεί στις υποκατηγορίες.

Οι κατηγορίες αποτελούν την βάση των υποκατηγοριών που είναι να δημιουργηθούν. Κάθε κατηγορία μπορεί να έχει όσες υποκατηγορίες χρειάζεται αλλά για προσθήκη αντικειμένου πρέπει να έχει τουλάχιστον μία και αυτή να έχει τουλάχιστον ένα πεδίο. Στις κατηγορίες εμφανίζεται η λίστα με όλες τις τρέχουσες κατηγορίες που υπάρχουν στην βάση, όπως επίσης και η δυνατότητα δημιουργίας νέας κατηγορίας. Οι διαχειριστές μπορούν να δουν την δομή μιας κατηγορίας επιλέγοντας το όνομά της. Δίνεται επίσης η δυνατότητα επεξεργασίας ή διαγραφής μιας κατηγορίας. Η διαγραφή μιας κατηγορίας οδηγεί στην διαγραφή όλων των υποκατηγοριών με αποτέλεσμα την διαγραφή όλων των αντικειμένων των υποκατηγοριών αυτών.

Στην επισκόπηση της δομής της κατηγορίας, εμφανίζεται η λίστα με όλα τα δηλωμένα κοινά στοιχεία που αντιστοιχούν στην κατηγορία αυτή. Επίσης υπάρχει η λίστα με τις υποκατηγορίες που ανήκουν σε αυτήν την κατηγορία, όπως επίσης και η δυνατότητα προσθήκης νέας κατηγορίας. Παράδειγμα προσθήκης νέας κατηγορίας υπάρχει στην Εικόνα 16. Η λίστα με τις κατηγορίες παρέχει την δυνατότητα επεξεργασίας της δομής της υποκατηγορίας ή διαγραφής της υποκατηγορίας. Όπως και πριν αν διαγραφεί μια υποκατηγορία, διαγράφονται επίσης όλα τα αντικείμενα της κατηγορίας αυτής.

Στην επισκόπηση της δομής της υποκατηγορίας εμφανίζεται η λίστα όλων των πεδίων της υποκατηγορίας. Επιπλέον δίνεται η δυνατότητα αλλαγής του ονόματος και της περιγραφής της υποκατηγορίας, όπως επίσης και η προσθήκη κάποιου νέου πεδίου για την κατηγορία αυτή. Η λίστα με τα πεδία δίνει με την σειρά τις 2 επιλογές: Επεξεργασία και Διαγραφή.



### Ορισμός νέου κοινού στοιχείου

Τα νέα κοινά στοιχεία θα εφαρμόζονται αυτόματα σε όλες τις νέες υποκατηγορίες που θα επιλέξετε

Κατηγορία : Όλες

Όνομα : Κατασκευαστής

Τύπος : Κείμενο

Μήκος : 65

Προεπιλεγμένη τιμή : Samsung

Δημόσιο :

Υποχρεωτικό :

Μοναδικό :

Ετικέτα QR :

Δημιουργία

Εικόνα 15: Παράδειγμα ορισμού κοινού στοιχείου

Όνομα	Περιγραφή	Επεξεργασία	Διαγραφή
Οθόνες	CRT, LCD, TFT	Επεξεργασία	Διαγραφή

### Προσθήκη νέας υποκατηγορίας

Οι κοινές μεταβλητές που έχουν οριστεί θα προστεθούν αυτόματα

Όνομα : Ηχεία

Περιγραφή :

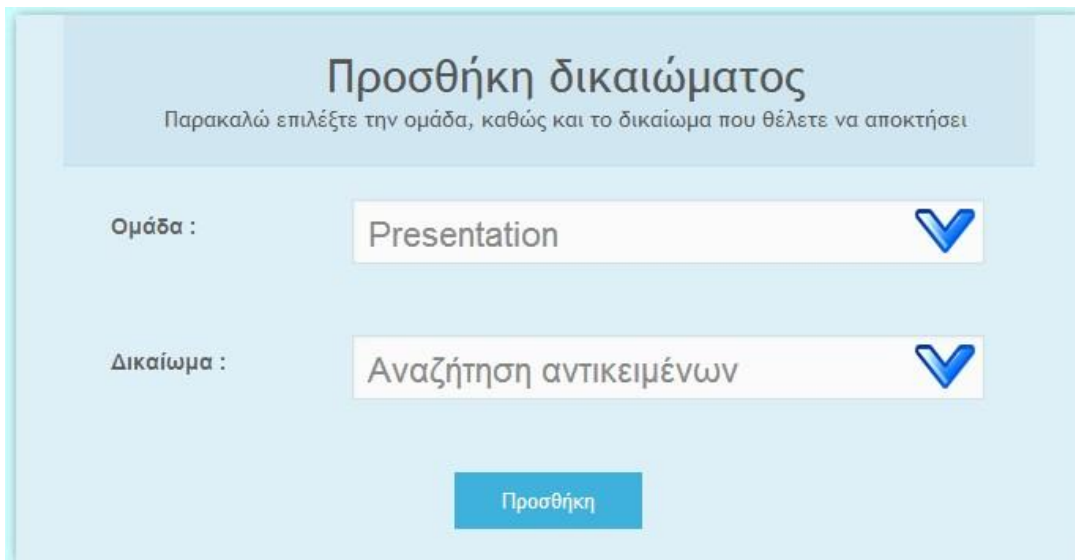
Δημιουργία

Εικόνα 16: Παράδειγμα προσθήκης νέας υποκατηγορίας

## 5.7 Δικαιώματα

Τα δικαιώματα αφορούν τις επιτρεπόμενες ενέργειες που μπορεί να πραγματοποιήσουν οι χρήστες της ομάδας που ανήκουν. Τα δικαιώματα εφαρμόζονται μόνο σε ομάδες και όχι ξεχωριστά σε κάποιον χρήστη. Όπως έχουμε αναλύσει στο υποκεφάλαιο 2.2, τα δικαιώματα είναι αναζήτηση αντικειμένου, προβολή αντικειμένου, επεξεργασία<sup>m</sup> αντικειμένου καθώς και παραχώρηση δικαιωμάτων διαχειριστή. Ένα παράδειγμα προσθήκης δικαιώματος υπάρχει στην Εικόνα 17.

Στην σελίδα εμφανίζονται τα δικαιώματα που είναι σε ισχύ καθώς και δίνεται η δυνατότητα προσθήκης νέου δικαιώματος. Από την λίστα οι διαχειριστές μπορούν, αν επιθυμούν, να καταργήσουν κάποιο δικαίωμα κάποιας ομάδας. Οποιαδήποτε αλλαγή εφαρμόζεται στους χρήστες στην επόμενη τους σύνδεση στο σύστημα.



Εικόνα 17: Παράδειγμα προσθήκης δικαιώματος

---

<sup>m</sup> και διαγραφή

## 5.8 Παραχώρηση Αντικειμένων

Η παραχώρηση κάποιου αντικειμένου μπορεί να γίνει μόνο από κάποιον διαχειριστή του συστήματος. Ο τομέας αυτός δίνει αρχικά 3 δυνατότητες στον διαχειριστή οι οποίες είναι οι εξής:

- Πρόσφατες παραχωρήσεις: Εδώ οι διαχειριστές μπορούν να δουν τις πιο πρόσφατες παραχωρήσεις. Για εξοικονόμηση χώρου οι παραχωρήσεις χωρίζονται ανά 10 και ο διαχειριστής μπορεί να επιλέξει ποια σελίδα επιθυμεί να δει. Το αντικείμενο εμφανίζεται "Υπό παραχώρηση" αν η ημερομηνία επιστροφής είναι πριν την ημερομηνία παραχώρησης, αλλιώς εμφανίζεται ως "Έχει επιστραφεί".
- Αναζήτηση παραχωρήσεων: Σε περίπτωση που οι παραχωρήσεις είναι πάρα πολλές, ή ο διαχειριστής επιθυμεί να ψάξει μία παλαιά παραχώρηση, συμπληρώνοντας τα πεδία που επιθυμεί, εκτελεί έρευνα στις παραχωρήσεις και του εμφανίζονται τα σχετικά αποτελέσματα.
- Παραχώρηση αντικειμένου: Στην παραχώρηση αντικειμένου, τα μόνα υποχρεωτικά πεδία είναι τα το ID του αντικειμένου που πρέπει να βρίσκεται στην βάση καθώς και το ID του ατόμου που είναι να δεχθεί την παραχώρηση. Τα δεδομένα των ατόμων καταγράφονται σε ένα πίνακα, ώστε σε επόμενη πιθανή παραχώρηση προς το ίδιο άτομο τα στοιχεία να συμπληρώνονται άμεσα.

Ένα παράδειγμα παραχώρησης κάποιου αντικειμένου υπάρχει στην Εικόνα 18.

### Παραχώρηση αντικειμένου

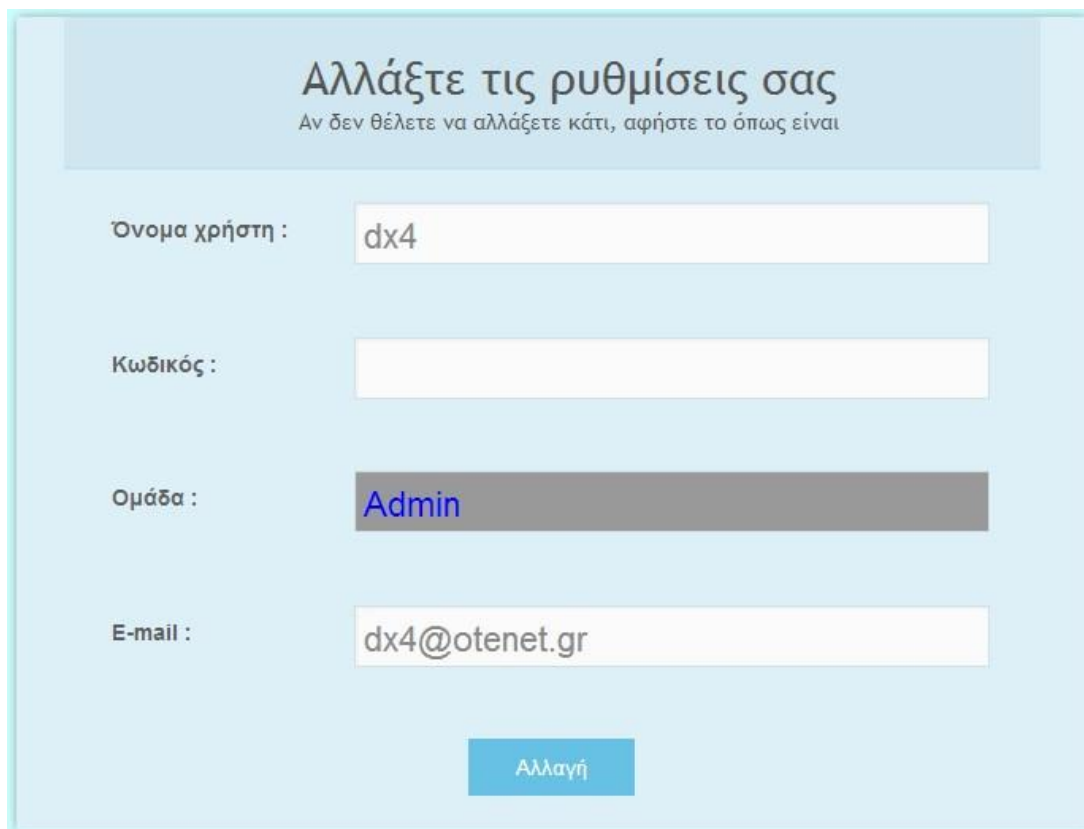
Παρακαλώ συμπληρώστε τα απαραίτητα στοιχεία

ID Αντικείμενου:	29
ID απόρου παραχώρησης:	ΑΞ 401295
Όνοματεπώνυμο:	Παπαδόπουλος Γιώργος
Τηλέφωνο 1:	2310111222
Τηλέφωνο 2:	6977669669
Διεύθυνση:	Μητροπόλεως 15
Εκπλήρωση Ημερομηνία επιστροφής:	2014-07-31
Ημερομηνία παραχώρησης:	2014-07-02
Σημείωση παραχώρησης:	Παραχωρείται για λόγους διεκπεραίωσης διπλωματικής εργασίας.
Υπόθετος παραχώρησης:	Αναστασόπουλος Δημήτρης
Κατάσταση αντικείμενου:	Καινούργιο
Επιπρόσθετο αρχείο:	<input type="button" value="Choose File"/> No file chosen <input type="button" value="Προσθήκη"/>

Εικόνα 18: Παράδειγμα παραχώρησης αντικειμένου

## 5.9 Ρυθμίσεις

Στις ρυθμίσεις ο κάθε χρήστης μπορεί να αλλάξει το όνομα χρήστη, τον κωδικό του ή το e-mail του αν επιθυμεί. Όλοι οι χρήστες του συστήματος έχουν πρόσβαση σε αυτό το μενού. Κανένας χρήστης όμως δεν μπορεί να αλλάξει την ομάδα του από αυτό το μενού. Αυτό είναι φανερό και στην Εικόνα 19.



**Αλλάξτε τις ρυθμίσεις σας**  
Αν δεν θέλετε να αλλάξετε κάτι, αφήστε το όπως είναι

Όνομα χρήστη : dx4

Κωδικός :

Ομάδα : Admin

E-mail : dx4@otenet.gr

Αλλαγή

Εικόνα 19: Ρυθμίσεις χρήστη.

## 5.10 Έξοδος

Στην έξοδο αναφέρεται το όνομα χρήσης και κατά την έξοδο διαγράφονται ολόκληρο το session. Μετά την έξοδο ο χρήστης ανακατευθύνεται στην σελίδα σύνδεσης.

## Κεφάλαιο 7: Συμπεράσματα και μελλοντικές χρήσεις

Το σύστημα που δημιουργήθηκε πέτυχε τους στόχους που είχαμε θέσει. Παρέχει ένα μεγάλο εύρος δυνατοτήτων στον χρήστη και του παρέχει βοήθεια και συμβουλές, όπως επίσης και τα κατάλληλα μηνύματα λάθους, ανάλογα πάντα με τις πράξεις που κάνει. Οι διαχειριστές έχουν πλήρη έλεγχο των βάσεων δεδομένων των αντικειμένων και μπορούν να τις διαμορφώσουν όποια στιγμή θέλουν. Επίσης το πλήθος των ομάδων που μπορούν να δημιουργηθούν δίνει την δυνατότητα να υπάρχει μεγάλο εύρος χρήσης του συστήματος χωρίς να υπάρχει ρίσκο να χαθούν πολύτιμα αρχεία ή να γίνει κάποιο λάθος.

Φυσικά όλες αυτές οι δυνατότητες και οι ευκολίες αποτέλεσαν μια μεγάλη προγραμματιστική πρόκληση, καθώς για κάθε ελευθερία και βοήθεια που παρέχεται στον χρήστη ο προγραμματιστής πρέπει να γράψει πολλές γραμμές κώδικα. Επίσης χρειάστηκε αρκετή μελέτη, ώστε να εξερευνηθούν οι πιθανές κινήσεις των χρηστών, ώστε το σύστημα να προβλέπει λάθη που μπορούν να γίνουν και να τα αποτρέπει δίνοντας πάντα την σωστή καθοδήγηση. Για αυτούς τους λόγους παρατηρούμε πως έχουμε τόσο πολύ κώδικα σε κάθε αρχείο που φυσικά αυτό δεν αποτελεί κανένα πρόβλημα για τον server, εφόσον οι υπολογιστές ολοκληρώνουν τις ενέργειες αυτές σε δέκατα του δευτερολέπτου.

Η ασφάλεια ήταν σαφώς ένα δύσκολο κομμάτι της εργασίας αυτής. Ένα τέτοιο σύστημα πρέπει να παρέχει προστασία στα δεδομένα από εσωτερικά λάθη όσο και από εξωτερικές επιθέσεις. Για αυτόν τον λόγο δόθηκε μεγάλη σημασία στους ελέγχους που γίνονται σε ολόκληρο το σύστημα. Σε κάθε σελίδα γίνονται πολλοί έλεγχοι όσον αφορά το αν ο χρήστης είναι συνδεδεμένος, τι δικαιώματα έχει, τι παραμέτρους έχει θέσει και τα σχετικά. Επίσης γίνεται έλεγχος των παραμέτρων που δίνει ο χρήστης, ώστε να μην περαστεί κάποια τιμή που κανονικά δεν γίνεται και προκληθεί κάποιο λάθος. Όλες οι συνδέσεις με την βάση δεδομένων γίνονται μόνο με PDO εκφράσεις προστατεύοντας οποιονδήποτε χρήστη να ανακτήσει δεδομένα από την βάση με κάποιο αθέμιτο μέσο. Επιπλέον, στην βάση δεδομένων οι κωδικοί των χρηστών είναι σε μορφή hash, ώστε να μην μπορεί κάποιος να αποκρυπτογραφήσει εύκολα κάποιον κωδικό. Συγκεκριμένα οι κωδικοί περνάνε από 2 προγράμματα hash, το md5 και το sha1 που μας εξασφαλίζει περισσότερη ασφάλεια στο σύστημα.

Ο έλεγχος των παραμέτρων και των τιμών των πεδίων που έχει συμπληρώσει ο χρήστης γίνεται με δύο τρόπους. Ο πρώτος είναι ο έλεγχος που γίνεται στα πεδία που είναι προκαθορισμένα από το σύστημα και οι τιμές που είναι να πάρουν είναι προβλέψιμες. Σε αυτήν την περίπτωση ο έλεγχος γίνεται με προκαθορισμένο τρόπο και οι τιμές ελέγχονται μαζί με τις υπόλοιπες παραμέτρους. Το σύστημα ελέγχει κατά πόσο είναι σωστές οι τιμές και αντιστοιχούν στην βάση δεδομένων ή γενικότερα στις λειτουργίες του συστήματος. Αν κάποιος χρήστης επιχειρήσει να περάσει παράμετρο η οποία θα κατέστρεφε την ευστάθεια του ιστοχώρου, το σύστημα αυτομάτως θα αποτρέψει αυτήν την ενέργεια και θα σταματήσει την περαιτέρω εκτέλεση κώδικα.

Ο δεύτερος τρόπος ελέγχου αφορά τα πεδία που καθορίζονται από τον χρήστη. Σε αυτήν την περίπτωση οι ιδιότητες κάθε πεδίο καθορίζονται σε έναν πίνακα. Το σύστημα αναλύει προσεκτικά τις ιδιότητες αυτές και εφαρμόζει τους κατάλληλους ελέγχους, βεβαιώνοντας πως οι τιμές που έδωσε ο χρήστης είναι αποδεκτές και από το σύστημα. Φυσικά για αποτροπή λάθος τιμών υπάρχουν βοηθητικά στοιχεία, όπως είναι το ημερολόγιο, που καθοδηγούν τον χρήστη να επιλέξει την τιμή που θέλει, εύκολα και γρήγορα, εφόσον το σύστημα μετά την επιλογή εμφανίζει την σωστή μορφή που θα έπρεπε να έχει, ώστε να γίνει αποδεκτή από το σύστημα.

Ως προς τις μελλοντικές επεκτάσεις μπορούμε να αναφέρουμε τα παρακάτω: Το σύστημα έχει σχεδιαστεί, ώστε να υποστηρίζει κάθε είδους προθήκη. Με λίγες αλλαγές στον κώδικα μπορούν να προστεθούν και άλλοι τύποι πεδίων. Επίσης υπάρχει η δυνατότητα αντί για QR ετικέτες να εκτυπώνεται RFID ετικέτες που πιθανώς να είναι πιο εύχρηστες σε κάποια εταιρία. Σαφώς μπορεί κάποιος, με πολύ εύκολο τρόπο, να προσθέσει μία καινούργια καταχώρηση στο μενού, καθώς και να γράψει τον κώδικα της σελίδας. Αυτό γίνεται γρήγορα, διότι έχει αναλυθεί ο κατασκευαστής του header και footer και κάποιος ο οποίος θα ήθελε μία καινούργια σελίδα στο σύστημα, το μόνο που έχει να κάνει είναι να αντιγράψει την παραμετροποίηση από κάποια άλλη σελίδα και να αλλάξει τις παραμέτρους κατά όπως επιθυμεί.

Μια ακόμα μελλοντική επέκταση που μπορεί να προστεθεί στο σύστημα είναι η λειτουργία του συστήματος σε κάποια βιβλιοθήκη. Οι αλλαγές που χρειάζονται για μία τέτοια ενέργεια είναι ελάχιστες. Ουσιαστικά πρέπει να προσαρμοστεί ο κώδικας που αφορά τις παραχωρήσεις των αντικειμένων για να καλύψει τις ανάγκες της. Οι QR ετικέτες αποδεικνύονται ιδιαίτερα χρήσιμες σε μια τέτοια επέκταση, καθώς οι χρήστες θα μπορούν να βλέπουν και να αναζητούν πληροφορίες για τα βιβλία γρήγορα και εύκολα.

Το σύστημα επίσης μπορεί να προσαρμοστεί ώστε κατά την παραχώρηση κάποιου αντικείμενου να υπάρχει χρέωση για κάθε μέρα που ο χρήστης έχει υπό την κατοχή του το αντικείμενο. Φυσικά για αυτό χρειάζεται αλλαγή στον κώδικα των καταχωρήσεων και πρέπει να προσαρμοστεί ώστε να υπολογίζει την διαφορά των ημερών παραχώρησης και επιστροφής και ανάλογα να δείχνει το κόστος. Επίσης θα μπορούσε να ειδοποιήσει τους διαχειριστές με e-mail αν κάποιο αντικείμενο έχει ξεπεράσει την ημερομηνία επιστροφής κατά κάποιες ημέρες και δεν έχει επιστραφεί ακόμα.

Γενικότερα κάθε σελίδα ενσωματώνει τα ίδια αρχεία για να υλοποιήσει τις λειτουργίες της. Αυτό σημαίνει πως αν ο χρήστης επιθυμεί να αλλάξει μια λειτουργία μπορεί να επεξεργαστεί μόνο τα αρχεία αυτά. Τροποποιώντας τα θα έχει καταφέρει να αλλάξει την λειτουργία όλου του συστήματος κατά όπως επιθυμεί. Αυτό φυσικά τον απαλλάσσει από την ανάγκη να χρειαστεί να επεξεργαστεί κάθε αρχείο ξεχωριστά.



## Βιογραφικό σημείωμα



- Επίθετο: Χριστοφορίδης
- Όνομα: Ιωάννης Ραφαήλ
- E-mail: dx4@otenet.gr

### Λίγα λόγια

Ο Χριστοφορίδης Ιωάννης Ραφαήλ σπούδασε στο Πανεπιστήμιο Δυτικής Μακεδονίας στην Κοζάνη, στο τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών. Τελείωσε τις σπουδές του με 64 περασμένα μαθήματα από τα 58 που χρειαζόταν για πτυχίο. Έδειξε μεγάλο ενδιαφέρον στα μαθήματα προγραμματισμού διαδικτύου και βάσεων δεδομένων. Ασχολήθηκε με το αντικείμενο αναπτύσσοντας δικιές του τεχνικές και μεθόδους προγραμματισμού, πράγμα που κάνει τα προγράμματά του μοναδικά. Επιπλέον, χρησιμοποιεί και κάποια μοτίβα όπως είναι ο "χιουμοριστικός" τρόπος ονομασίας των μεταβλητών και των συναρτήσεων που δίνει μια ιδιαιτερότητα στα προγράμματα του. Επίσης έχει δημιουργήσει και έναν ιστοχώρο που χρησιμοποιείται από το πανεπιστήμιο για την αυτόματη εξέταση φοιτητών στα μαθήματα του κ. Μηνά Δασυγένη, με το όνομα iexams. Αυτή η διπλωματική εργασία αποτέλεσε πρόκληση καθώς έπρεπε να διευρυνθούν οι γνώσεις του φοιτητή, να αναπτυχθούν οι τεχνικές του και οι μέθοδοί του, όπως και να χρησιμοποιήσει και να βελτιώσει τις ήδη υφιστάμενες γνώσεις.

## Βιβλιογραφία

1. HTML. *Βικιπαίδεια*. [Ηλεκτρονικό] <http://el.wikipedia.org/wiki/HTML>.
2. CSS. *Βικιπαίδεια*. [Ηλεκτρονικό] <http://el.wikipedia.org/wiki/CSS>.
3. PHP. *Βικιπαίδεια*. [Ηλεκτρονικό] <http://el.wikipedia.org/wiki/PHP>.
4. MySQL. *Βικιπαίδεια*. [Ηλεκτρονικό] <http://el.wikipedia.org/wiki/MySQL>.
5. JavaScript. *Βικιπαίδεια*. [Ηλεκτρονικό] <http://el.wikipedia.org/wiki/JavaScript>.
6. Κώδικας QR. *Βικιπαίδεια*. [Ηλεκτρονικό] [http://el.wikipedia.org/wiki/%CE%9A%CF%8E%CE%B4%CE%B9%CE%BA%CE%B1%CF%82\\_QR](http://el.wikipedia.org/wiki/%CE%9A%CF%8E%CE%B4%CE%B9%CE%BA%CE%B1%CF%82_QR).
7. PHP - Qr Code. *sourceforge*. [Ηλεκτρονικό] <http://phpqrcode.sourceforge.net/>.
8. *FPDF*. [Ηλεκτρονικό] <http://www.fpdf.org/>.
9. *Slidebars*. *Adam Charles Smith*. [Ηλεκτρονικό] <http://plugins.adchsm.me/slidebars/>.
10. *WinSCP*. [Ηλεκτρονικό] <http://winscp.net/eng/index.php>.
11. *Notepad++*. [Ηλεκτρονικό] <http://notepad-plus-plus.org/>.
12. *Putty*. [Ηλεκτρονικό] <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>.
13. *FPDF*. *FPDF*. [Ηλεκτρονικό] <http://www.fpdf.org/>.