

Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Διαδικτυακό σύστημα διαχείρισης δεδομένων για την απογραφή εργαστηριακού εξοπλισμού An on line database management system for laboratory equipment inventory
Όνοματεπώνυμο Φοιτητή	Σωτήριος Μουστακίδης
Πατρώνυμο	Αντώνιος
Αριθμός Μητρώου	ΜΠΠΛ/ 11019
Επιβλέπων	Χρήστος Δουληγέρης, Καθηγητής

Ημερομηνία Παράδοσης **Μάρτιος 2016**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Όνομα Επώνυμο
Βαθμίδα

Περίληψη

Η συγκεκριμένη εργασία επικεντρώνεται στην υλοποίηση και εφαρμογή ενός διαδικτυακού συστήματος βάσεων δεδομένων. Το συγκεκριμένο σύστημα – εφαρμογή εξυπηρετεί στην διαχείριση και διοίκηση ενός υποθετικού εργαστηρίου. Όπως είναι γνωστό ένα εργαστήριο, είτε βιολογικό, είτε τοξικολογικό, είτε μηχανικό, αποτελείται όχι μόνο από το ανθρώπινο δυναμικό που το απαρτίζει, αλλά και από εξοπλισμό. Ως εξοπλισμός εργαστηρίου ορίζεται οτιδήποτε προϊόν χρησιμοποιείται για την λειτουργία του εργαστηρίου εκτός από το ανθρώπινο δυναμικό και τις κτιριακές εγκαταστάσεις. Ο εξοπλισμός μπορεί να διαχωριστεί βάσει πολλών κριτηρίων, όπως για παράδειγμα η τοποθεσία, οι προμηθευτές, οι χρήστες κλπ.

Η σωστή απογραφή του εξοπλισμού ενός εργαστηρίου αποτελεί σημαντικό βοήθημα για την διαχείριση και τον συντονισμό του εργαστηρίου. Σε αρχικό στάδιο, κατά την διάρκεια σχεδιασμού ενός νέου εργαστηρίου όπως είναι λογικό θα υπάρξει ένα πλήθος από παραγγελίες εξοπλισμού, η απογραφή του εξοπλισμού είναι απαραίτητη ώστε να αποφευχθούν διπλές παραγγελίες του ίδιου εξοπλισμού ή να ελεγχθεί καλύτερα το συνολικό κόστος αν είναι στα πλαίσια του προϋπολογισμού. Σε επόμενο στάδιο, κατά την λειτουργία του εργαστηρίου, η απογραφή του εξοπλισμού μπορεί να φανεί επίσης χρήσιμο εργαλείο για την συντήρηση. Σχεδόν κάθε προϊόν, είτε ηλεκτρονική συσκευή είτε ηλεκτρονικό εργαλείο έχει και ημερομηνία εγγύησης, αλλά και πιθανή ημερομηνία ανανέωσης. Προκειμένου να ελέγχουμε καλύτερα τις αντίστοιχες ημερομηνίες το εργαλείο της απογραφής θα μπορούσε να φανεί χρήσιμο.

Το εργαλείο απογραφής που υλοποιήσαμε είναι ένα σύγχρονο σύστημα διαχείρισης δεδομένων. Οι χρήστες του συστήματος είτε είναι εξοικειωμένοι με τις βάσεις δεδομένων είτε όχι μπορούν εύκολα να μελετούν τον εξοπλισμό του εργαστηρίου, να προσθέτουν ή να αφαιρούν νέα στοιχεία, να παρακολουθούν τις ημερομηνίες και σε γενικές γραμμές να διαχειρίζονται τον εξοπλισμό του εργαστηρίου. Η εφαρμογή που υλοποιήθηκε είναι διαδικτυακή εφαρμογή ώστε να είναι εύκολα προσβάσιμη από διαφορετικά σημεία ή και από διαφορετικούς χρήστες. Τέλος η εφαρμογή είναι εύχρηστη, πράγμα που σημαίνει ο χρήστης είναι σε θέση να βρει εύκολα την πληροφορία που τον ενδιαφέρει και με ανάλογη ευκολία να την επεξεργαστεί.

Abstract

This thesis focuses on the development of an online database system. This system can be applied to the management and administration of a hypothetical laboratory. As it is known, a laboratory, either biological or toxicological or mechanical, is composed not only of human resources but also of equipment. As laboratory equipment, is defined any product is used for the operation of the laboratory outside of the human resources and the building facilities. The equipment can be separated based on several criteria such as location, suppliers, users etc.

The efficient inventory management of the laboratory equipment is an important tool for the management and coordination of the laboratory. Initially, during the design of a new laboratory, a number of equipment orders may exist and inventory management of equipment may be needed not only to avoid double orders and excessive inventory, but also to better control the total cost accordingly to the budget. In the next stage, during the operation of the laboratory, the management of the equipment inventory can also be a useful tool for maintenance. Almost every product or electronic device or electronic tool has a guarantee date and a possible renewal date. In order to better control the corresponding dates, the inventory management tool could be helpful.

However the inventory management tool should be a modern data management system. The users of the system, either familiar with databases or not, should be able to easily study the equipment of the laboratory, to add or to remove new elements, to monitor dates and to administer the lab equipment. The application that will be implemented should be a web application in order to be easily accessible from different locations and by the different users. Finally, the application should be both user-friendly and usable, which means that users are able to easily find and process the information required.

Περιεχόμενα

Περίληψη.....	3
Abstract	4
Κεφάλαιο 1	6
Εισαγωγή	6
Κεφάλαιο 2	8
Συστήματα βάσεων δεδομένων	8
Ανάλυση απαιτήσεων.....	11
Λειτουργικές απαιτήσεις συστήματος.....	11
Μη λειτουργικές απαιτήσεις συστήματος.....	12
UML	13
Διάγραμμα Use – Case.....	14
Διάγραμμα sequence.....	15
Κεφάλαιο 3	16
Αρχιτεκτονικός σχεδιασμός	16
Μοντέλο client – server	16
Τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής.....	17
CSS- Web design	17
HTML.....	20
PHP	23
Joomla.....	24
Πλεονεκτήματα του Joomla	24
Joomla 3 – Responsitive Design	25
SQL.....	26
Περιγραφή της εφαρμογής.....	27
Λειτουργικότητα της ιστοσελίδας.....	28
Φόρμα εισόδου χρηστών / Login form	28
Φόρμα επικοινωνίας	29
Φόρμα εισαγωγής δεδομένων	30
Κώδικας για φόρμα εισαγωγής δεδομένων	31
Πίνακας απογραφής	33
Κεφάλαιο 4	37
Ασφάλεια συστήματος	37
Κεφάλαιο 5	39

Αξιολόγηση συστήματος.....	39
Αποτελέσματα ερωτηματολογίου χρηστών.....	43
Συμπεράσματα	44
Βιβλιογραφία.....	46

Κεφάλαιο 1

Εισαγωγή

Η αλματώδης ανάπτυξη της επιστήμης της πληροφορικής και των επικοινωνιών τα τελευταία χρόνια έχει καταστήσει την πληροφορία ως ένα από τα βασικότερα και πολυτιμότερα αγαθά. Είναι κοινός τόπος σήμερα η εκτίμηση ότι το αγαθό της πληροφορίας είναι επιθυμητό απ' όλους τους εργαζόμενους αλλά και τους εκπαιδευόμενους, ώστε να είναι πιο αποδοτικοί, ανταγωνιστικοί αλλά και παραγωγικοί στην εργασία τους.

Τα δεδομένα στις μέρες μας καταγράφονται και χρησιμοποιούνται σε κάθε δραστηριότητα επιστημονική, εμπορική, παραγωγική, κυβερνητική, στρατιωτική κλπ. Οι μεγάλες ποσότητες δεδομένων επιβάλλουν την εύρεση αποτελεσματικών μεθόδων αποθήκευσης. Τα δεδομένα πρέπει να είναι οργανωμένα με τέτοιο τρόπο ώστε να διευκολύνεται η αναζήτηση και η ενημέρωσή τους και αυτό ακριβώς είναι το αντικείμενο της συγκεκριμένης διπλωματικής εργασίας.

Ένας από τους βασικούς λόγους για τους οποίους αναθέτουμε σε υπολογιστές να επιλύουν προβλήματα είναι η δυνατότητα τους να αποθηκεύουν μεγάλο όγκο δεδομένων με ταχύτητα ακρίβεια και ασφάλεια. Αυτό είχε ως αποτέλεσμα να αναπτυχθούν συστήματα όπου τα δεδομένα αποθηκεύονταν με τη μορφή αρχείων σε ψηφιακά μέσα, όπως μαγνητικές ταινίες και μαγνητικούς δίσκους, με τη χρήση κατάλληλων εφαρμογών λογισμικού. Οι εφαρμογές αυτές, εκτός από δυνατότητες εισαγωγής και αποθήκευσης δεδομένων στο σύστημα, έδιναν επιπλέον στον χρήστη τη ευκαιρία να ανακαλεί, ενημερώνει, διαγράφει και γενικά να επεξεργάζεται και να αξιοποιεί με πολλούς τρόπους τα δεδομένα των αρχείων. Η διαχείριση των δεδομένων ανήκει στην ευθύνη του προγραμματιστή της εφαρμογής.

Τα συστήματα βάσεων δεδομένων τα χρησιμοποιούμε για να μπορούμε να αποθηκεύσουμε, να επεξεργαστούμε αλλά και να εκμεταλλευτούμε αποδοτικά αυτόν τον τεράστιο όγκο των πληροφοριών που αυξάνονται με αλματώδεις ρυθμούς καθημερινά. Τα συστήματα βάσεων δεδομένων ξεκίνησαν σαν ειδικές πληροφοριακές εφαρμογές και εξελίχθηκαν ως ένα από τα πιο βασικά τμήματα της επιστήμης των υπολογιστών. Ένα σύστημα βάσης δεδομένων αποτελείται από μια συλλογή συσχετισμένων δεδομένων και ένα σύνολο προγραμμάτων για την πρόσβαση και διαχείρισή τους. Το σύνολο των δεδομένων αυτών, είναι αυτό που συνήθως αναφέρεται ως "βάση δεδομένων", ενώ ο βασικός σκοπός ενός τέτοιου συστήματος είναι η αποτελεσματική αποθήκευση, πρόσβαση, οργάνωση και αναζήτηση των δεδομένων που περιέχει.

Η συγκεκριμένη εργασία επικεντρώθηκε στην υλοποίηση ενός συστήματος διαχείρισης δεδομένων για την καταγραφή εξοπλισμού εργαστηρίου. Στην συγκεκριμένη παράγραφο θα αναλύσουμε την σημασία του συστήματος. Κατά την διάρκεια της απασχόλησής μου σε δύο ερευνητικά εργαστήρια (Eurl-Ecvam - European Union Reference Laboratory for alternatives to animal testing, και SGI Lab - Smart Grid and Interoperability Lab) διαπίστωσα ότι η απογραφή του εργαστηριακού εξοπλισμού αποτελεί σημαντικό εργαλείο για τη διαχείριση του εργαστηρίου. Το στάδιο της δημιουργίας ενός εργαστηρίου προϋποθέτει ένα πλήθος από παραγγελίες τόσο τεχνικού εξοπλισμού όσο και λογισμικού. Στο στάδιο των παραγγελιών ο κάθε project leader καλείται να δηλώσει τι εξοπλισμός χρειάζεται να παραγγελθεί για το κάθε project του και να προκηρύξει διαγωνισμό προκειμένου να δηλώσουν οι προμηθευτές τις προσφορές τους. Το γεγονός ότι υπάρχουν πολλοί προμηθευτές αλλά κυρίως αρκετοί project leaders κάνει την απογραφή πιο δύσκολη. Το σύστημα απογραφής εξοπλισμού βοηθάει στο να αποφύγουμε μία διπλή παραγγελία με το ίδιο περιεχόμενο. Για

παράδειγμα είναι πολύ σύνηθες να χρησιμοποιείται το λειτουργικό Matlab σε ένα project, προκειμένου, όμως, να εγκατασταθεί στους υπολογιστές του εργαστηρίου, απαιτείται να έχουν αγοραστεί τα πνευματικά δικαιώματα του. Με την αγορά των δικαιωμάτων ο αγοραστής μπορεί να το χρησιμοποιήσει σε πέντε διαφορετικούς υπολογιστές. Σε περίπτωση που ένας project leader χρειάζεται το συγκεκριμένο λειτουργικό μπορεί μέσω του συστήματος απογραφής εξοπλισμού να δει, πριν προχωρήσει στην παραγγελία, αν κάποιο άλλο μέλος του εργαστηρίου έχει ήδη αγοράσει το συγκεκριμένο προϊόν. Με αυτόν τον τρόπο μειώνεται το κόστος και διευκολύνεται ο συντονισμός. Από την άλλη μεριά κατά την διάρκεια της συντήρησης του εργαστηρίου, κάθε project leader καλείται στο τέλος της χρονιάς να καθορίσει και να δικαιολογήσει το χρηματικό ποσό για τον προϋπολογισμό του επόμενου έτους. Τα περισσότερα προϊόντα προϋποθέτουν και ένα ετήσιο κόστος συντήρησης. Σε αυτήν την φάση το σύστημα απογραφής μπορεί με ένα ερώτημα να ενημερώσει τον project leader, ποιες από τις παραγγελίες του πρόκειται να λήξουν στο προσεχές έτος και πόσο είναι το συνολικό κόστος τους.

Για την υλοποίηση του συγκεκριμένου συστήματος διαχείρισης δεδομένων χρησιμοποιήθηκε το CMS Joomla και το component Tabulizer. Ωστόσο η επέκταση του συγκεκριμένου component κρίθηκε αναγκαία για τις ανάγκες του συγκεκριμένου συστήματος. Η δυνατότητα που παρέχει το Tabulizer είναι η απλή προβολή δεδομένων με την μορφή πινάκων και η δυνατότητα μορφοποίησης πινάκων. Η διαδικασία για την λειτουργία του Tabulizer είναι αφού ο χρήστης συνδεθεί στο διαχειριστικό του Joomla να δημιουργήσει ένα νέο πίνακα, να καταγράψει τα δεδομένα και τέλος να προβάλλει τον πίνακα σε ένα άρθρο του Joomla. Τα προβλήματα που εντοπίσαμε κατά την μελέτη του Tabulizer για τις ανάγκες του συγκεκριμένου συστήματος διαχείρισης δεδομένων είναι τα εξής:

- Τα δεδομένα που καταχωρεί ο χρήστης δεν αποθηκεύονται πουθενά, ούτε καν στην βάση δεδομένων του Joomla, αποθηκεύονται μόνο στο διαχειριστικό του Joomla. Αυτό αποτελεί σημαντικό μειονέκτημα για την διαδικασία backup.
- Μόνο ο διαχειριστής του Joomla μπορεί να καταχωρίσει δεδομένα στους πίνακες. Σκοπός του δικού μας συστήματος είναι να δοθεί η δυνατότητα σε ένα συγκεκριμένο πλήθος χρηστών (project leaders, dataproviders) να προσθέτουν τα δικά τους δεδομένα χωρίς να απαιτείται να συνδεθούν στο διαχειριστικό του Joomla.
- Μέσα από το συγκεκριμένο component δίνεται η δυνατότητα προβολής δεδομένων ωστόσο δεν δίνεται η δυνατότητα τροποποίησης ή διαγραφής ήδη υπάρχουσας εγγραφής.

Προκειμένου να επιλύσουμε τα συγκεκριμένα προβλήματα, υλοποιήσαμε τις συγκεκριμένες ενέργειες.

- Χρησιμοποιήσαμε μια ξεχωριστή βάση δεδομένων μόνο για την απογραφή του εξοπλισμού και συνδέσαμε το συγκεκριμένο component με την συγκεκριμένη βάση δεδομένων. Καθορίσαμε με αυτόν τον τρόπο ένα data source το οποίο δεν προϋπήρχε. Με την ύπαρξη της βάσης δεδομένων δίνεται η δυνατότητα back up.
- Δημιουργήσαμε μέσω του Joomla ιεραρχία χρηστών (dataproviders, dataviewers) και υλοποιήσαμε μια φόρμα εισαγωγής δεδομένων σε php – javascript. Η συγκεκριμένη φόρμα είναι ορατή μόνο στους dataproviders. Μεσώ της συγκεκριμένης φόρμας οι dataproviders καταχωρούν στην ανεξάρτητη βάση δεδομένων τα δεδομένα τους και στην συνέχεια αυτά προβάλλονται μέσω του Tabulizer.

Δώσαμε την δυνατότητα επεξεργασίας και διαγραφής δεδομένων στους dataproviders. Όπως φαίνεται στην εικόνα 16 και αναλύεται παρακάτω προσθέσαμε μια νέα στήλη στον πίνακα δεδομένων με όνομα action, με την δυνατότητα διαγραφής και επεξεργασίας δεδομένων. Η προσθήκη της στήλης action έγινε στο php αρχείο `id2links {joomla root directory} /templates/tabulizer/modify/id2links.php` με την προσθήκη δύο links για την διαγραφή ή επεξεργασία δεδομένων. Με το που ο χρήστης επιλέξει μια από τις δύο επιλογές καλείται το αρχείο `inventorytask.php`. Το συγκεκριμένο αρχείο δεν προϋπήρχε στο component Tabulizer και το υλοποιήσαμε προκειμένου να ρυθμίσουμε τις λειτουργίες εγγραφής διαγραφής και επεξεργασίας δεδομένων με την βοήθεια των φορμών εισαγωγής, επεξεργασίας και του πίνακα απογραφής.

Τέλος να διευκρινιστεί ότι προκειμένου να ελέγξουμε την ιεραρχία χρηστών και να αποτρέψουμε την πρόσβαση των dataviewers στην τελευταία στήλη του πίνακα απογραφής προσθέσαμε php κώδικα

στο αρχείο id2links.php που αναφέρθηκε παραπάνω και μέσω του διαχειριστικού του cms Joomla ορίσαμε τις ομάδες χρηστών

Στο επόμενο κεφάλαιο θα ασχοληθούμε με τα συστήματα διαχείρισης βάσεων δεδομένων, με την δομή τους, τον τρόπο λειτουργίας τους και με την ανάλυση των απαιτήσεων (λειτουργικές και μη λειτουργικές) του συστήματος που υλοποιήσαμε.

Το τρίτο κεφάλαιο επικεντρώνεται στην ανάπτυξη της διαδικτυακής εφαρμογής και στον αρχιτεκτονικό σχεδιασμό του συστήματος. Αναλύεται το μοντέλο client server και στην συνέχεια αναλύονται οι τεχνολογίες διαδικτύου που χρησιμοποιήσαμε και ο λόγος που επιλέχθηκαν.

Το τέταρτο κεφάλαιο ασχολείται με την ασφάλεια διαδικτυακών συστημάτων γενικώς, ωστόσο αναφέρονται όλα τα μέτρα ασφάλειας που πήραμε προκειμένου να παρέχουμε ένα υψηλό επίπεδο ασφάλειας στην εφαρμογή μας.

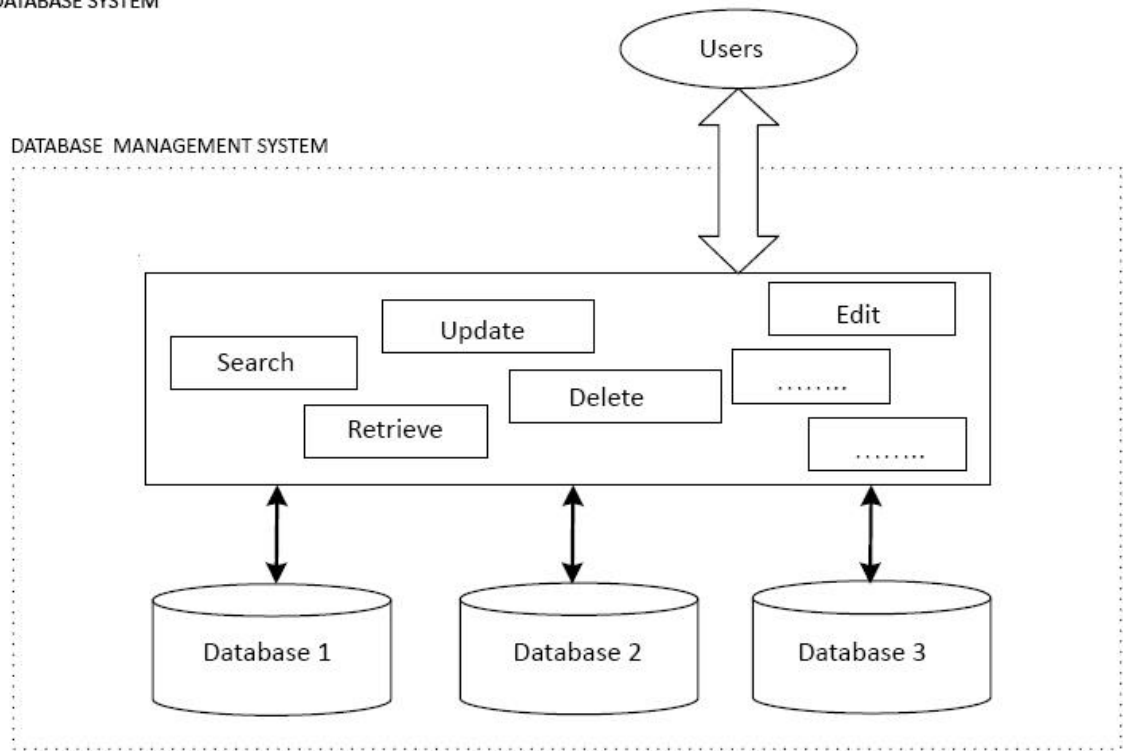
Κεφάλαιο 2

Συστήματα βάσεων δεδομένων

Ένα σύστημα Βάσης Δεδομένων αναλαμβάνει αποκλειστικά την αποθήκευση , προστασία και επεξεργασία των δεδομένων , παρέχοντας εξελιγμένους μηχανισμούς πρόσβασης , δικαιωμάτων και ενημέρωσης δεδομένων. Ένα σύστημα βάσης δεδομένων αποτελείται από τα εξής:

- Τη βάση δεδομένων, η οποία αποτελεί μια συλλογή στοιχείων μεταξύ τους, που είναι δομημένα και καταχωρημένα με κατάλληλο τρόπο.
- Το σύστημα διαχείρισης Βάσης δεδομένων το οποίο είναι ένα λογισμικό σύστημα που υλοποιεί όλες τις λειτουργίες που πρέπει να υποστηριχθούν , όπως αναζήτηση εισαγωγή, διαγραφή , συγχρονισμός προσπελάσεων , προστασία και πολλές ακόμη . Το σύστημα διαχείρισης βάσης δεδομένων είναι σε θέση να διαχειρίζεται πολλές βάσεις δεδομένων ταυτόχρονα , ανάλογα με τις απαιτήσεις των χρηστών.[6]

DATABASE SYSTEM



Εικόνα 1: Δομή συστήματος διαχείρισης δεδομένων

Στο σχήμα παρουσιάζεται η σχέση μεταξύ της βάσης δεδομένων και του συστήματος διαχείρισης βάσεων δεδομένων. Όπως προκύπτει οι χρήστες δεν έχουν άμεση επαφή με τα δεδομένα. Ο έλεγχος των δεδομένων ανήκει στο σύστημα διαχείρισης δεδομένων και μόνο αυτό είναι υπεύθυνο για τη διαχείριση και την προστασία τους.

Σε γενικές γραμμές κάθε χρήστης της εφαρμογής έχει διαφορετικά δικαιώματα ως προς την προσπέλαση των δεδομένων. Για παράδειγμα, μια ομάδα χρηστών μπορεί να έχει μόνο δικαιώματα ανάγνωσης δεδομένων, ενώ μια άλλη ομάδα χρηστών μπορεί να έχει δικαιώματα ανάγνωσης, διαγραφής αλλά και εισαγωγής νέων στοιχείων. Η εφαρμογή μπορεί να είναι ταυτόχρονα συνδεδεμένη με περισσότερες από μια βάσεις δεδομένων και να επεξεργάζεται ή να συνδυάζει δεδομένα που ανήκουν σε διαφορετικές βάσεις.

Με τον όρο πληροφορία αναφερόμαστε συνήθως σε ειδήσεις, γεγονότα και έννοιες που αποκτάμε από την καθημερινή μας επικοινωνία και τα θεωρούμε ως αποκτηθείσα γνώση, ενώ τα δεδομένα μπορούν να είναι μη κατάλληλα επεξεργασμένα και μη ταξινομημένα σύνολα πληροφοριών. Ένας αυστηρός ορισμός για το τι είναι δεδομένα και τι είναι πληροφορία, σύμφωνα με την επιτροπή ANSI των ΗΠΑ, είναι ο εξής :

- Δεδομένα είναι μια παράσταση, όπως γράμματα, αριθμοί, σύμβολα κ.ά. στα οποία μπορούμε να δώσουμε κάποια σημασία (έννοια).
- Πληροφορία είναι η σημασία που δίνουμε σ' ένα σύνολο από δεδομένα, τα οποία μπορούμε να επεξεργαστούμε βάσει προκαθορισμένων κανόνων και να βγάλουμε έτσι κάποια χρήσιμα συμπεράσματα. Με τις πληροφορίες περιορίζεται η αβεβαιότητα που έχουμε για διάφορα πράγματα και βοηθούμε έτσι στο να λάβουμε σωστές αποφάσεις.

Τα δεδομένα μπορούν να θεωρηθούν ως τρόποι αναπαράστασης εννοιών και γεγονότων που μπορούν να υποστούν διαχείριση και επεξεργασία. Η συλλογή και αποθήκευση ενός τεράστιου όγκου δεδομένων όπως απαιτούν οι κοινωνικές συνθήκες σήμερα, δεν λύνει τελείως το πρόβλημα της σωστής οργάνωσης και ταξινόμησης των δεδομένων. Τα δεδομένα θα πρέπει να οργανωθούν με τέτοιο τρόπο έτσι ώστε να μπορούμε να τα εντοπίζουμε και να τα αξιοποιούμε εύκολα και γρήγορα και τη στιγμή που τα χρειαζόμαστε.[3]

Ένα κλασικό παράδειγμα μη σωστής οργάνωσης δεδομένων θα ήταν για παράδειγμα ο τηλεφωνικός κατάλογος της πόλης της Θεσσαλονίκης, όπου οι συνδρομητές δεν θα ήταν καταχωρημένοι αλφαβητικά σύμφωνα με το επώνυμο και το όνομά τους, αλλά εντελώς τυχαία. Ένας τέτοιος τηλεφωνικός κατάλογος θα περιείχε μια τεράστια ποσότητα δεδομένων αλλά θα ήταν ουσιαστικά άχρηστος.

Ένα άλλο παράδειγμα μη σωστής οργάνωσης δεδομένων θα ήταν μια πολύ μεγάλη βιβλιοθήκη με χιλιάδες τόμους βιβλίων και χωρίς να διαθέτει κάποιο υποτυπώδες σύστημα οργάνωσης και ταξινόμησης των βιβλίων. Ούτε οι υπάλληλοι της βιβλιοθήκης θα μπορούσαν να κάνουν τη δουλειά τους αλλά ούτε και οι επισκέπτες θα μπορούσαν να αξιοποιήσουν την πληθώρα των πληροφοριών που περιέχονται στα βιβλία. Εκτός λοιπόν από τη μόνιμη αποθήκευση των δεδομένων, χρειαζόμαστε και κάποιους τρόπους ευέλικτης και αποδοτικής οργάνωσής τους.

Χαρακτηριστικά παραδείγματα δεδομένων που απαιτούν σωστή και αποδοτική οργάνωση είναι τα εξής :

- Τα στοιχεία υπαλλήλων, πελατών, προμηθευτών και παραγγελιών μιας εμπορικής επιχείρησης.
- Τα στοιχεία υλικών μιας αποθήκης.
- Τα στοιχεία ταινιών, πελατών και δανεισμών μιας βιντεολέσχης.
- Τα στοιχεία υπαλλήλων, γιατρών, ασθενών αλλά και υλικών ενός νοσοκομείου.
- Τα στοιχεία βιβλίων, χρηστών (δανειστών) και δανεισμών μιας βιβλιοθήκης.

Για να δοθεί μια λύση σ' όλα τα παραπάνω προβλήματα, και με βάση το γεγονός ότι η χρήση των ηλεκτρονικών υπολογιστών και συνεπώς η ηλεκτρονική καταχώρηση και επεξεργασία δεδομένων αυξήθηκε κατακόρυφα ήδη από τη δεκαετία του '70 στις μεγάλες επιχειρήσεις και άρα είχαμε πάρα πολλές εφαρμογές να επεξεργάζονται δεδομένα σε πάρα πολλά αρχεία ταυτόχρονα, προτάθηκε η συνένωση όλων των αρχείων μιας εφαρμογής. Εκτός, όμως, από τη συνένωση των αρχείων, απαιτείτο και μια σωστή οργάνωσή τους. Δημιουργήθηκαν έτσι οι Τράπεζες Πληροφοριών ή Βάσεις Δεδομένων (Data Bases).

Οι χρήστες των εφαρμογών αντλούν τα στοιχεία που τους ενδιαφέρουν από τη βάση δεδομένων χωρίς να είναι σε θέση να γνωρίζουν με ποιο τρόπο είναι οργανωμένα τα δεδομένα σ' αυτήν. Το ΣΔΒΔ παίζει τον ρόλο του μεσάζοντα ανάμεσα στον χρήστη και τη βάση δεδομένων και μόνο μέσω του ΣΔΒΔ μπορεί ο χρήστης να αντλήσει πληροφορίες από τη βάση δεδομένων. Ένα ΣΔΒΔ μπορεί να είναι εγκατεστημένο σ' έναν μόνο υπολογιστή ή και σ' ένα δίκτυο υπολογιστών και μπορεί να χρησιμοποιείται από έναν χρήστη ή και από πολλούς χρήστες.

Ένα Σύστημα Βάσης Δεδομένων (ΣΒΔ) ή DBS (Data Base System) αποτελείται από το υλικό, το λογισμικό, τη βάση δεδομένων και τους χρήστες. Είναι δηλαδή ένα σύστημα με το οποίο μπορούμε να αποθηκεύσουμε και να αξιοποιήσουμε δεδομένα με τη βοήθεια ηλεκτρονικού υπολογιστή. Αναλυτικά:

- Το υλικό (hardware) αποτελείται όπως είναι γνωστό από τους ηλεκτρονικούς υπολογιστές, τα περιφερειακά, τους σκληρούς δίσκους, τις μαγνητικές ταινίες κ.ά., όπου είναι αποθηκευμένα τα αρχεία της βάσης δεδομένων αλλά και τα προγράμματα που χρησιμοποιούνται για την επεξεργασία τους.
- Το λογισμικό (software) είναι τα προγράμματα που χρησιμοποιούνται για την επεξεργασία των δεδομένων (στοιχείων) της βάσης δεδομένων.
- Η βάση δεδομένων (data base) αποτελείται από το σύνολο των αρχείων όπου είναι αποθηκευμένα τα δεδομένα του συστήματος. Τα στοιχεία αυτά μπορεί να βρίσκονται αποθηκευμένα σ' έναν φυσικό υπολογιστή αλλά και σε περισσότερους. Όμως, στον

χρήστη δίνεται η εντύπωση ότι βρίσκονται συγκεντρωμένα στον ίδιο υπολογιστή. Τα δεδομένα των αρχείων αυτών είναι ενοποιημένα (data integration), δηλ. δεν υπάρχει πλεονασμός (άσκοπη επανάληψη) δεδομένων και μερισμένα (data sharing), δηλ. υπάρχει δυνατότητα ταυτόχρονης προσπέλασης των δεδομένων από πολλούς χρήστες. Ο κάθε χρήστης έχει διαφορετικά δικαιώματα και βλέπει διαφορετικό κομμάτι της βάσης δεδομένων, ανάλογα με τον σκοπό για τον οποίο συνδέεται.

- Οι χρήστες (users) μιας βάσης δεδομένων χωρίζονται στις εξής κατηγορίες :
 - Τελικοί χρήστες (end users). Χρησιμοποιούν κάποια εφαρμογή για να παίρνουν στοιχεία από μια βάση δεδομένων, έχουν τις λιγότερες δυνατότητες επέμβασης στα στοιχεία της βάσης δεδομένων, χρησιμοποιούν ειδικούς κωδικούς πρόσβασης και το σύστημα τους επιτρέπει ανάλογα πρόσβαση σε συγκεκριμένο κομμάτι της βάσης δεδομένων.
 - Προγραμματιστές εφαρμογών (application programmers). Αναπτύσσουν τις εφαρμογές του ΣΒΔ σε κάποια από τις γνωστές γλώσσες προγραμματισμού.
 - Διαχειριστής δεδομένων (data administrator – DA). Έχει τη διοικητική αρμοδιότητα και ευθύνη για την οργάνωση της βάσης δεδομένων και την απόδοση δικαιωμάτων πρόσβασης στους χρήστες.
 - Διαχειριστής βάσης δεδομένων (database administrator – DBA). Λαμβάνει οδηγίες από τον διαχειριστή δεδομένων και είναι αυτός που διαθέτει τις τεχνικές γνώσεις και αρμοδιότητες για τη σωστή και αποδοτική λειτουργία του ΣΒΔ.

Ανάλυση απαιτήσεων

Αφού αναλύσαμε την δομή των συστημάτων διαχείρισης δεδομένων και πιο συγκεκριμένα στο διαδίκτυο, θα επικεντρωθούμε στην ανάλυση απαιτήσεων. Με την ανάλυση απαιτήσεων δεν εννοούμε τίποτα άλλο παρά μόνο έναν κατάλογο όπου γίνεται πλήρης καταγραφή όλων των υπηρεσιών που πρέπει να παρέχει το σύστημα που πρόκειται να δημιουργηθεί. Ο προγραμματιστής θα πρέπει να εργαστεί με τους πελάτες για να διαπιστώσει ποιες θα πρέπει να παρέχει το σύστημα για να είναι λειτουργικό.

Όπως καταλαβαίνουμε η ανάλυση απαιτήσεων είναι μία διαδικασία συνεργασίας όπου διαφορετικά άτομα συναντιούνται και ανταλλάζουν απόψεις. Σε αυτή την φάση έρχεται η βοήθεια του αναλυτή προς τον προγραμματιστή ώστε να δημιουργήσει κάτι εύκολο και όχι κάτι δυσνόητο προς το χρήστη.

Λειτουργικές απαιτήσεις συστήματος

- Ταυτοποίηση χρηστών: προκειμένου να εισαχθεί ένα χρήστης στην ιστοσελίδα θα πρέπει πρώτα να ταυτοποιηθεί. Το περιεχόμενο της συγκεκριμένης ιστοσελίδας δεν είναι κοινού ενδιαφέροντος καθώς αφορά ένα συγκεκριμένο γκρουπ ατόμων , επίσης για λόγους ασφαλείας είναι προτιμότερο να μην είναι προσβάσιμα τα δεδομένα εύκολα.
- Ομαδοποίηση χρηστών: στην συγκεκριμένη εφαρμογή θα πρέπει να υπάρχει μια ιεραρχία χρηστών καθώς δεν έχουν πρόσβαση σε ίδια πληροφορία όλοι οι χρήστες. Υπάρχουν ομάδες χρηστών με περισσότερα δικαιώματα από άλλους.
- Αποθήκευση-Εισαγωγή δεδομένων: υπάρχει μια ομάδα χρηστών η οποία είναι αρμόδια για να παρέχει και να αποθηκεύει δεδομένα που κατέχει στην βάση δεδομένων του συστήματος. Προκειμένου να είναι εύχρηστη η εφαρμογή θα πρέπει να περιέχει μια φόρμα για την εισαγωγή δεδομένων.
- Μορφοποίηση δεδομένων: Είναι σημαντικό ο χρήστης που έχει δικαίωμα να καταχωρεί δεδομένα να μπορεί να τα διορθώνει σε περίπτωση που αντιληφθεί ένα λάθος ή να μπορεί να τα αλλάξει σε περίπτωση ανανέωσης.[2]
- Διαγραφή δεδομένων: Σε περίπτωση που τα ήδη καταχωρημένα δεδομένα έχουν καταστραφεί ή έχουν αντικατασταθεί είναι απαραίτητη η δυνατότητα διαγραφής τους από την βάση δεδομένων

- Προβολή δεδομένων: Τα δεδομένα της βάσης θα πρέπει να παρουσιάζονται με την μορφή πίνακα σε μία καρτέλα τις ιστοσελίδας ώστε οι χρήστες να μπορούν να καταλάβουν εύκολα τι έχει καταχωρηθεί στην βάση δεδομένων.
- Αναζήτηση δεδομένων . Μέσω μίας φόρμας αναζήτησης δίνεται η δυνατότητα στους χρήστες να αναζητήσουν δεδομένα εάν γνωρίζουν ένα στοιχείο που τους ενδιαφέρει. Για παράδειγμα μπορούν να βρουν τα μηχανήματα που βρίσκονται στο χ δωμάτιο αν αναζητήσουν με κριτήριο την τοποθεσία κλπ.
- Δυνατότητα αποθήκευσης αρχείων σε μορφή excel ώστε ο χρήστης να μπορεί να δει η να επεξεργαστεί τα δεδομένα του και χωρίς σύνδεση στην εφαρμογή.[4]
- Επικοινωνία με τον διαχειριστή: Είναι σημαντικό ο χρήστης να μπορεί να επικοινωνήσει με τον διαχειριστή του συστήματος προκειμένου να ζητήσει βοήθεια, να ρωτήσει ή ακόμη και να προτείνει διορθώσεις.

Μη λειτουργικές απαιτήσεις συστήματος

Οι περιορισμοί στις λειτουργίες και τις υπηρεσίες που προσφέρονται από το σύστημα αποτελούν τις μη λειτουργικές απαιτήσεις του συστήματος. Αναλυτικότερα :

- Υψηλή ταχύτητα ανάκτησης δεδομένων και ταχεία απόκριση του συστήματος στις απαιτήσεις του χρήστη για παροχή υπηρεσιών. Με αυτό τον τρόπο το σύστημα εγγυάται την έγκαιρη ενημέρωση του χρήστη (όλων των επιπέδων) σχετικά με τις πληροφορίες που του ζητούνται.
- Το σύστημα πρέπει να είναι φιλικό προς το χρήστη. Αυτό επιτυγχάνεται με τη δημιουργία απλών και περιεκτικών διεπαφών που δεν απαιτούν ειδικές γνώσεις στη χρήση ηλεκτρονικών υπολογιστών και γενικότερα ιδιαίτερη εξοικείωση με την τεχνολογία. Το σύστημα πρέπει να παρέχει ασφάλεια. Οι χρήστες του συστήματος ομαδοποιούνται ανάλογα με τα δικαιώματα πρόσβασης που έχουν στο σύστημα ώστε να αποφεύγεται η χρήση πληροφοριών (προσωπικά στοιχεία) για ανεπιθύμητες ενέργειες.
- Το κόστος του συστήματος πρέπει να περιορίζεται από τις οικονομικές απαιτήσεις που έχει καθορίσει ο πελάτης. Αυτό επιτυγχάνεται με τη χρήση ήδη υπάρχοντος εγκατεστημένου υλικού και την εξασφάλιση οικονομικών πόρων.
- Η υλοποίηση και η εγκατάσταση του συστήματος πρέπει να τηρεί τα αρχικά χρονοδιαγράμματα που συμφωνήθηκαν με τον πελάτη και να καλύπτει τους στόχους του.
- Το σύστημα οφείλει να είναι αξιόπιστο. Δηλαδή δεν πρέπει να επηρεάζεται από ανωμαλίες του δικτύου ηλεκτροδότησης. Επομένως, απαιτείται η χρήση συστήματος UPS για να ελαχιστοποιηθούν οι απώλειες σε περίπτωση πτώσης της τάσης του ρεύματος.
- Το σύστημα διατηρεί αντίγραφα ασφαλείας ανά τακτά χρονικά διαστήματα. Σε περίπτωση που χαθεί μέρος των δεδομένων του συστήματος για οποιοδήποτε λόγο, πρέπει να είναι δυνατή η επανάκτηση του. Προτείνεται η χρήση βοηθητικού δίσκου στον οποίο αποθηκεύονται τα περιεχόμενα της βάσης του server.
- Δυνατότητες συντήρησης και επέκτασης του συστήματος κρίνονται απαραίτητες ώστε να μπορεί το σύστημα να προσαρμόζεται στις αλλαγές των δεδομένων
- Η εγκατάσταση του συστήματος πρέπει να είναι γρήγορη και εύκολη ώστε να τεθεί σύντομα σε λειτουργία και να αντιμετωπίσει τα προβλήματα που καλείται να λύσει.
- Το σύστημα πρέπει να χαρακτηρίζεται από μεταφερσιμότητα. Η αποθήκευση των δεδομένων της βάσης θα πρέπει να είναι όσο το δυνατόν πιο απλή. Η μορφή των δεδομένων πρέπει να είναι τέτοια ώστε να καθίσταται εφικτή η διαχείρισή τους από διάφορα συστήματα διαχείρισης βάσεων δεδομένων.
- Απαιτείται αποθηκευτικός χώρος 10MB για την αποθήκευση του κώδικα Οι ελάχιστες απαιτήσεις που χρειάζονται για να λειτουργήσει το σύστημα είναι μνήμη 2 RAM chips 512MB και cache memory 1024KB Το σύστημα πρέπει να τρέχει πάνω σε πλατφόρμα PC με επεξεργαστές τεχνολογίας Pentium VI
- Χρήση Apache μέσω ενός browser προκειμένου να εξασφαλίζεται η επικοινωνία με ένα διακομιστή HTTP.Επιπλέον απαιτείται η χρήση μιας βάσης δεδομένων MySQL.
- Το σύστημα πρέπει να είναι on line ώστε να έχουν ευκολότερη πρόσβαση όλοι οι χρήστες , και να γίνεται καλύτερη διαχείριση των δεδομένων της βάσης δεδομένων.[3]

UML

Η ενοποιημένη γλώσσα σχεδιασμού (*unified modelling language*) (UML) είναι μια γραφική γλώσσα για την οπτική παράσταση, τη διαμόρφωση προδιαγραφών και την τεκμηρίωση συστημάτων που βασίζονται σε λογισμικό. Η UML στοχεύει στο σχεδιασμό αντικειμενοστραφών συστημάτων. Το σχέδιο είναι μια απλοποιημένη παράσταση της πραγματικότητας.

Σχεδιάζουμε για να μπορέσουμε να καταλάβουμε το σύστημα που αναπτύσσουμε. Έτσι, δημιουργώντας ένα σχέδιο επιτυγχάνουμε τέσσερις στόχους:

- παριστάνουμε οπτικά το σύστημα που έχουμε ή θέλουμε να κατασκευάσουμε,
- προσδιορίζουμε τη δομή και τη συμπεριφορά του συστήματος,
- δημιουργούμε ένα πρότυπο για να βασίσουμε την κατασκευή του συστήματος,
- τεκμηριώνουμε τις αποφάσεις που λάβαμε.

Σε όλους τους τεχνολογικούς τομείς ο σχεδιασμός βασίζεται σε τέσσερις βασικές αρχές:

- η επιλογή του είδους του σχεδίου έχει επίπτωση στον τρόπο και την μορφή επίλυσης του προβλήματος,
- όλα τα σχέδια εκφράζονται σε διαφορετικές βαθμίδες ακρίβειας,
- τα καλύτερα σχέδια σχετίζονται με την πραγματικότητα,
- ένα είδος σχεδίων δεν είναι ποτέ αρκετό.

Η UML περιλαμβάνει τρία βασικά στοιχεία:

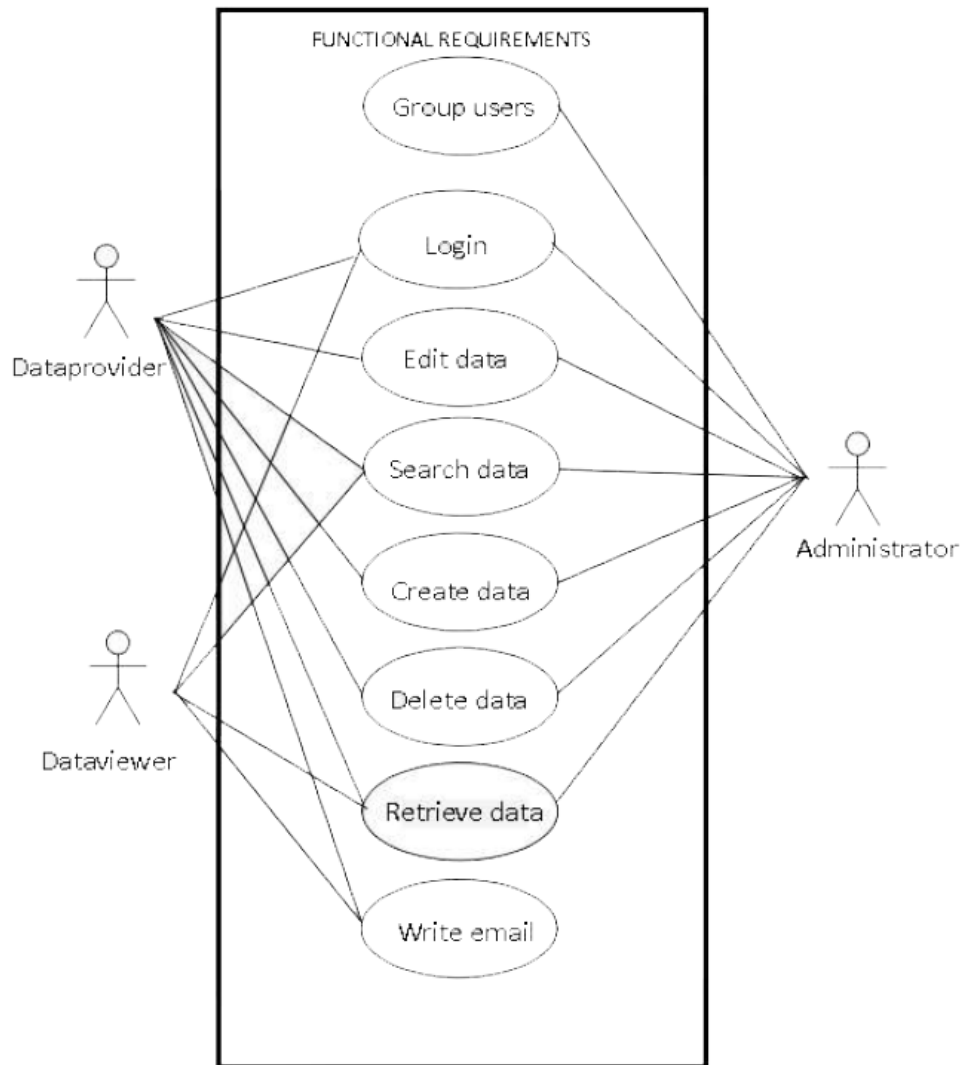
Οντότητες

Σχέσεις

Διαγράμματα

Η UML είναι μια πλήρης και πλούσια γλώσσα με εξαιρετικά ευρύ πεδίο εφαρμογής. Δύο από τα πιο χαρακτηριστικά διαγράμματα θα παρουσιάσουμε σε αυτήν την ενότητα.

Διάγραμμα Use – Case

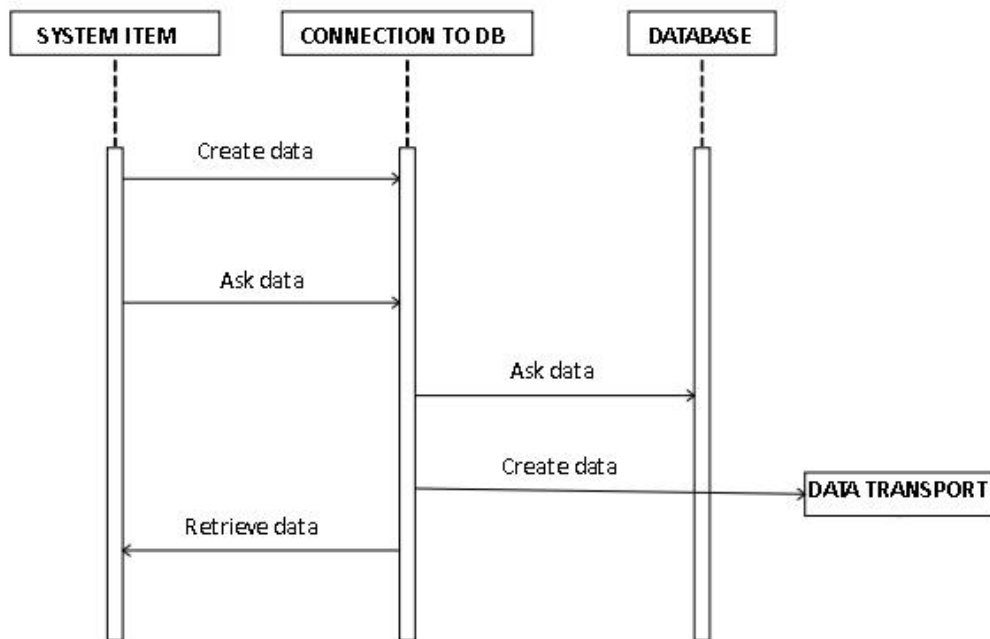


Εικόνα 2: Διάγραμμα Use Case

Όπως διαφαίνεται στην εικόνα 2 έχουμε στην συγκεκριμένη διαδικτυακή εφαρμογή μια ιεραρχία χρηστών. Οι κατηγορίες χρηστών είναι τρεις και κάθε χρήστης ανάλογα με την κατηγορία που ανήκει αποκτά τα αντίστοιχα δικαιώματα πρόσβασης.

Στο πλαίσιο του διαγράμματος απεικονίζονται οι λειτουργικές απαιτήσεις που παρουσιάστηκαν παραπάνω. Η γραμμή μεταξύ λειτουργιών του συστήματος με χρήστες δείχνουν ότι ο χρήστης έχει πρόσβαση στην συγκεκριμένη λειτουργία. Για παράδειγμα όλοι οι χρήστες του συστήματος έχουν δικαίωμα πρόσβασης στην προβολή των δεδομένων ωστόσο οι dataviewers δεν έχουν δικαίωμα διαγραφής ή επεξεργασίας δεδομένων.

Διάγραμμα sequence



Εικόνα 3: Διάγραμμα sequence

Ένα διάγραμμα ακολουθίας αναπαριστά την αλληλεπίδραση, δηλαδή μία σειρά από μηνύματα ανάμεσα σε μια συνεργασία ώστε να πραγματοποιηθεί μία επιθυμητή πράξη ή αποτέλεσμα στην εφαρμογή.

Ένα διάγραμμα ακολουθίας έχει δύο διαστάσεις

- 1) την κάθετη διάσταση που αναπαριστά το χρόνο και
- 2) την οριζόντια διάσταση που αναπαριστά τα διάφορα αντικείμενα. Κανονικά ο χρόνος προχωρά προς τα κάτω, Αν είναι επιθυμητό οι διαστάσεις μπορούν να αλλάξουν θέση. Συνήθως έχει σημασία μόνο η ακολουθία του χρόνου, αλλά σε εφαρμογές πραγματικού χρόνου ο άξονας του χρόνου μπορεί να είναι οποιοδήποτε μέτρο. Δεν υπάρχει καμία σημασία στην ταξινόμηση των αντικειμένων στον οριζόντιο άξονα. Τα αντικείμενα μπορούν να ομαδοποιηθούν σε "swimlanes" σε ένα διάγραμμα.

Οι βασικές λειτουργίες και η προτεραιότητα τους φαίνονται στο παραπάνω σχήμα. Αρχικά ο χρήστης επιλέγει να δημιουργήσει ένα αντικείμενο του συστήματος (π.χ. έργο, εργασία, χρήστης, επαφή κλπ.) και μέσω της διασύνδεσης βάσης-συστήματος ζητά τα αντίστοιχα δεδομένα από την ΒΔ (πλατφόρμες συμπλήρωσης κ δημιουργίας). Όταν δημιουργηθούν νέα δεδομένα αποθηκεύονται στη βάση δεδομένων. Μια άλλη ενέργεια του χρήστη είναι να ζητήσει δεδομένα από το σύστημα (π.χ. προβολή προφίλ του, προβολή έργου κλπ.). Η διαδικασία αυτή προϋποθέτει την απαίτηση δεδομένων από την ΒΔ και την επιστροφή δεδομένων από την ΒΔ.

Κεφάλαιο 3

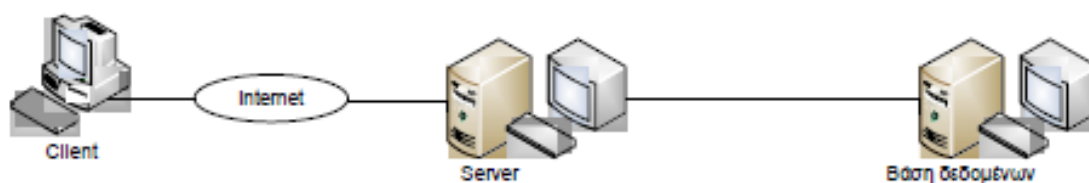
Αρχιτεκτονικός σχεδιασμός

Μοντέλο client – server

Η δημιουργία ενός ενιαίου πληροφοριακού συστήματος, με δυνατότητα καταχώρισης του μέγιστου αριθμού χρήσιμων πληροφοριών για την καταγραφή εξοπλισμού, απαιτεί την εφαρμογή ομοιόμορφης, ανοικτής και εύληπτης αρχιτεκτονικής, όπου θα αποδίδεται επίκαιρη εικόνα της υφιστάμενης κατάστασης, παρέχοντας τη δυνατότητα για σύνθεση πληροφοριών και επεξεργασία σεναρίων. Το πλήθος και η πολυπλοκότητα των αναγκών των χρηστών καθώς και η ανάγκη υποστήριξης ερωτήσεων τύπου what's if και what's up, επιβάλλει όσο το δυνατόν πιο σαφή και οργανωμένο τρόπο καταχώρισης των δεδομένων. Αυτό απαιτεί μια εύχρηστη, κατανοητή, ανοικτή και κοινά αποδεκτή αρχιτεκτονική του πληροφοριακού συστήματος, λαμβάνοντας υπόψη όλα τα στοιχεία που χαρακτηρίζουν μια πληροφορία. Η διαχείριση των ερωτήσεων τύπου what's up στο προτεινόμενο πληροφοριακό σύστημα εμφανίζεται στο διάγραμμα που ακολουθεί.

Τα δεδομένα που χρησιμοποιούνται θα βρίσκονται καταχωρημένα, είτε στη βάση δεδομένων του συστήματος, είτε σε εξωτερικές βάσεις δεδομένων. Έτσι, το πληροφοριακό σύστημα πρέπει να περιλαμβάνει κατ' ελάχιστο την περιοχή με τις πληροφορίες δεδομένων, μια περιοχή επεξεργασίας και σύνθεσης πληροφοριών που αντλεί ο χρήστης για τις ανάγκες του και μια περιοχή αποθήκευσης των επεξεργασμένων δεδομένων για επαναχρησιμοποίησή τους στο μέλλον. Στην αρχιτεκτονική του προτεινόμενου πληροφοριακού συστήματος πρέπει να καθοριστούν τα ελάχιστα επίπεδα κατάταξης των πληροφοριών, δημιουργώντας τους κατάλληλους κόμβους επιλογής για την ανάκτησή τους.

Το σύστημα που θα προσομοιώσουμε θα ακολουθήσει το μοντέλο client – server. Ένας κεντρικός server παρέχει τις υπηρεσίες του σε ένα σύνολο πελατών (clients) και ουσιαστικά έχει τον πλήρη έλεγχο του συστήματος και αναλαμβάνει να παράγει τις απαραίτητες αποφάσεις τις οποίες στη συνέχεια τις υλοποιούν οι πελάτες. Ο server επικοινωνεί με τους clients μέσω ενός δικτύου ανταλλάσσοντας δεδομένα με τη μορφή μηνυμάτων. Η μεταφορά των δεδομένων διέπεται από ένα σύνολο κανόνων που αποτελούν το πρωτόκολλο μεταφοράς. Ο server αναλαμβάνει να διαχειρίζεται τη κεντρική βάση δεδομένων του συστήματος. Αυτό το μοντέλο προωθεί την αποτελεσματική χρήση των υποσυστημάτων που αποτελούν το έργο όπως επίσης και μειωμένο κόστος εγκατάστασης και συντήρησης. Είναι αρκετά πιο οικονομική και σύντομη η εισαγωγή, η διαγραφή ή η αναβάθμιση ενός ή περισσότερων clients ανάλογα με τις ανάγκες του συστήματος οι οποίοι δε διατηρούν κάποια βάση δεδομένων ούτε έχουν απαιτούν τον αποθηκευτικό χώρο ενός server. Η υλοποίηση ενός τέτοιου συστήματος είναι αρκετά ευκολότερη από ένα κατανεμημένο σύστημα με μεγάλο αριθμό server γιατί υπάρχει μία κεντρική βάση και γενικά ένα κέντρο αποφάσεων. Αυτό κάνει πιο γρήγορη την λήψη αποφάσεων και την απόκριση του συστήματος αλλά κρύβει τον κίνδυνο ολικής κατάρρευσης του συστήματος σε περίπτωση σοβαρής βλάβης του server. Κατά συνέπεια, διατηρούνται επιπρόσθετοι εφεδρικοί χώροι αποθήκευσης δεδομένων και εφαρμόζονται τεχνικές ελαχιστοποίησης εμφάνισης σφάλματος από εξωγενείς παράγοντες όπως πτώση τάσης του ηλεκτρικού ρεύματος.



Εικόνα 4: Μοντέλο Client - Server

Στο δίκτυο υπάρχει ο εξυπηρετητής (server), που διαχειρίζεται αρχεία και δικτυακές υπηρεσίες, και οι πελάτες (client) που έχουν την δυνατότητα να τρέχουν εφαρμογές και να αποκτούν πρόσβαση στον εξυπηρετητή (server). Οι πελάτες (clients), προφανώς βασίζονται στον εξυπηρετητή (server).

Τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής

CSS- Web design

Τα Διαδοχικά Φύλλα Στυλ (CSS, Cascading Style Sheets) αποτελούν ένα πολύ καλό εργαλείο για να μπορούμε να αλλάζουμε την εμφάνιση και τη διάταξη (layout) των ιστοσελίδων μας. Μπορούν να μας γλυτώσουν από πολύ χρόνο και κόπο και μας δίνουν τη δυνατότητα να σχεδιάζουμε τις ιστοσελίδες μας με μια εντελώς καινούργια φιλοσοφία. Η κατανόηση των CSS απαιτεί να υπάρχει κάποια βασική εμπειρία με την HTML.

Για να δουλέψουμε με τα CSS δεν χρειάζεται να χρησιμοποιήσουμε κάποιο πρόγραμμα δημιουργίας ιστοσελίδων, όπως είναι τα γνωστά FrontPage, DreamWeaver ή και το Word, καθώς θα σταθούν εμπόδιο στην κατανόηση των CSS. Αυτό που χρειαζόμαστε είναι ένας απλός επεξεργαστής κειμένου, όπως είναι το Σημειωματάριο (Notepad) των Windows.

Μπορούμε να χρησιμοποιήσουμε όποιον φυλλομετρητή (browser) επιθυμούμε για να βλέπουμε πώς θα εμφανίζονται οι ιστοσελίδες που θα δημιουργούμε. Συνεπώς αυτό που χρειαζόμαστε είναι ένας φυλλομετρητής (browser) και ένας απλός επεξεργαστής κειμένου (text editor).

Στο ερώτημα τι μπορούμε να κάνουμε με τα CSS, θα πρέπει να γνωρίζουμε ότι τα CSS είναι μια γλώσσα στυλ (style language) που ορίζουν τη διάταξη (layout) των HTML εγγράφων. Για παράδειγμα, τα CSS έχουν να κάνουν με γραμματοσειρές (fonts), με χρώματα (colours), με περιθώρια (margins), με εικόνες φόντου (background images) και με πολλά άλλα. Με την HTML θα δυσκολευτούμε να αλλάξουμε τη διάταξη των ιστοσελίδων μας, αλλά τα CSS προσφέρουν πολλές επιλογές και είναι πολύ πιο συγκεκριμένα στις λεπτομέρειες. Επιπλέον, υποστηρίζονται απ' όλους τους φυλλομετρητές.

Στο ερώτημα ποια είναι η διαφορά ανάμεσα στα CSS και την HTML, μπορούμε να πούμε ότι η HTML χρησιμοποιείται για να δομηθεί το περιεχόμενο (content), ενώ τα CSS χρησιμοποιούνται για τη διαμόρφωση ή μορφοποίηση (formatting) του δομημένου περιεχομένου. Σύντομα θα γίνει σαφές το τι εννοούμε. Από το ξεκίνημά της, η HTML χρησιμοποιείτο μόνο για να μπορούμε να προσθέτουμε δομή στο κείμενο, όπως για να επισημαίνουμε ποια είναι κεφαλίδα (headline) ή ποια είναι παράγραφος (paragraph) με τα γνωστά tags (ετικέτες) της HTML, όπως είναι τα <h1> και <p>.

Όμως, καθώς η δημοτικότητα του Web αύξανε συνεχώς, οι σχεδιαστές των ιστοσελίδων άρχισαν να ψάχνουν για επιπλέον δυνατότητες, όπως προσθήκη διάταξης (layout) στα έγγραφα. Για να γίνει αυτό, οι φυλλομετρητές επινόησαν καινούργια HTML tags, όπως για παράδειγμα το , τα οποία διέφεραν από τα αρχικά HTML tags καθώς όριζαν τη διάταξη και όχι τη δομή μιας ιστοσελίδας.

Στο ερώτημα τι έχουν να μου προσφέρουν τα CSS, μπορούμε να πούμε ότι τα CSS αποτέλεσαν μια επανάσταση στον κόσμο του Web design και τα μεγάλα πλεονεκτήματά τους είναι τα εξής :

- Δυνατότητα για τον ταυτόχρονο έλεγχο της διάταξης (layout) πολλών εγγράφων από ένα μόνο φύλλο στυλ (style sheet).
- Πιο ακριβής έλεγχος της διάταξης (layout) των ιστοσελίδων.
- Εφαρμογή διαφορετικής διάταξης σε διαφορετικές εξόδους, όπως screen, print κ.ά.
- Αναρίθμητες προηγμένες και εξεζητημένες τεχνικές.

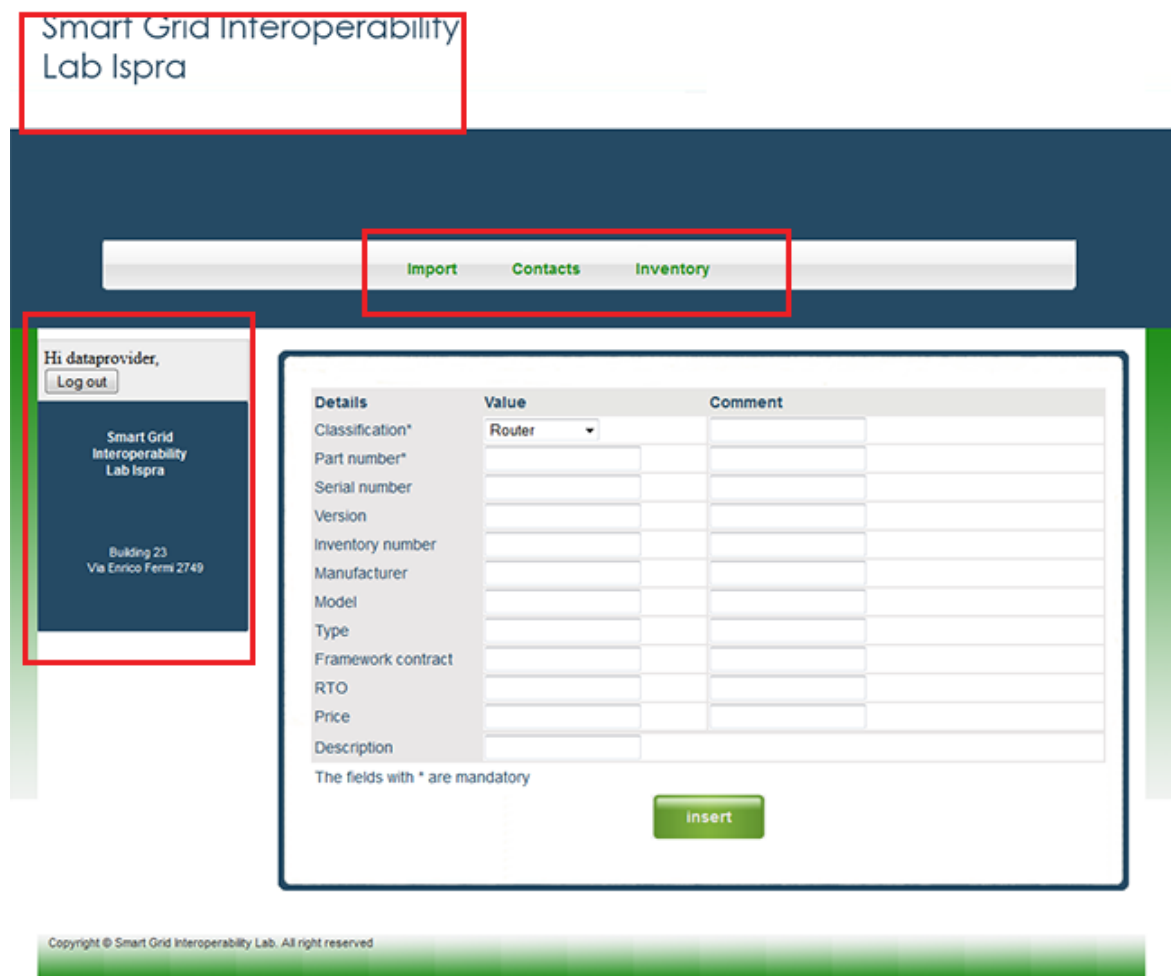
Για την συγκεκριμένη διαδικτυακή εφαρμογή δεν χρησιμοποιήθηκε κάποιο έτοιμο template , ούτε κάποιο από τα προεπιλεγμένα templates του CMS Joomla.

Δημιούργησα ένα αρχείο .css , το οποίο είναι εύκολα προσαρμόσιμο όχι μόνο σε άλλες ιστοσελίδες αλλά είναι προσαρμόσιμο επίσης σε κινητά τηλέφωνα . (mobile friendly). Ο κώδικας του αρχείου layout_css.css με την μορφοποίηση καθώς επίσης και το αρχείο index.html με την αρχιτεκτονική του

template. Οι εικόνες που χρησιμοποιήθηκαν για την μορφοποίηση της ιστοσελίδας δημιουργήθηκαν στο Adobe Photoshop.

Το template που δημιουργήθηκε διατηρεί τα βασικά χαρακτηριστικά Συγκεκριμένα

- Το λογότυπο : είναι τοποθετημένο στην αριστερή πάνω μεριά της ιστοσελίδας. Ο λόγος που τοποθετείται εκεί είναι όχι μόνο το να είναι ευδιάκριτο αλλά κυρίως στην περίπτωση που οι χρήστες αποστέλλουν ή δημοσιεύουν την ιστοσελίδα σε μέσα κοινωνικής δικτύωσης ή προσωπικά Blogs , να εμφανίζεται η φωτογραφία του λογότυπου.
- Κεντρικό μενού : Στην συγκεκριμένη ιστοσελίδα περιοριστήκαμε σε ένα μενού επιλογών . Ο λόγος είναι πρώτον ότι ο αριθμός των menu-items είναι πολύ μικρός και δεύτερον το περιεχόμενο της ιστοσελίδας είναι φόρμες και πίνακες, δεν είναι μια ιστοσελίδα που στηρίζεται σε μακροσκελές κείμενο ή σε πολλά άρθρα , οπότε ένα πάνω μενού και ένα κάτω μενού με ακριβώς τα ίδια menu-items δεν θα εξυπηρετούσε.
- Αριστερή στήλη : στην συγκεκριμένη στήλη είναι η φόρμα εισόδου χρηστών και ένα πλαίσιο με στοιχεία επικοινωνίας –τοποθεσίας του εργαστηρίου.



Εικόνα 5: Τα βασικά στοιχεία του template

Κώδικας του css – web design

```
1 body{
2 margin-top: -15px;
3 margin-left: 0px;
4 margin-right: 0px;
5 margin-bottom:0px;
6 }
7 .input{
8 background-image: url(../images/input.png);
9 width: 104px;
10 height: 40px;
11 font-family: Arial, Helvetica, sans-serif;
12 font-size: 14px;
13 font-weight: bold;
14 color: #e9e7e7;
15 border-width: 0px;
16 }
17 .tooltip {
18 display:none!important;
19 }
20 .userdata {
21 border-width: 0px;
22 }
23 .input:hover{
24 color: #254ae4;
25 }
26 .clearboth {
27 clear:both;
28 }
29 #all {
30 margin-top: 0px;
31 background-image: url(../images/bg.png);
32 background-repeat: repeat-x;
33 }
34 #wrapper {
35 width: 1004px;
36 margin-right: auto;
37 margin-left: auto;
38 background-color: #FFFFFF;
39 }
40 #wrapper #content {
41 width: 771px;
42 float: right;
43 padding-top: 20px;
44 padding-right: 15px;
45 padding-bottom: 20px;
46 }
47 #wrapper #up_menu li:hover{
48 text-decoration:underline;
49 height: 60px;
50 border-bottom: 4px solid;
51 }
52 #wrapper #left_col {
53 float: left;
54 padding-left:0px;
55 padding-top: 10px;
56 }
57 #login-form {
58 -moz-border-radius: 0px;
59 -webkit-border-radius: 0px;
60 -khtml-border-radius: 0px;
61 border-radius: 0px;
62 background:#eee;
63 border:1px solid #ccc;
64 padding:5px;
65 margin:0px 0px 0px 0px;
66 width:180px;
67 }
68 #wrapper #left_col #left_menu ul {
69 height: 128px;
70 width: 145px;
71 background-image: url(../images/epikoinonia.png);
72 padding-right: 0px;
73 padding-left: 0px;
74 }
75 #wrapper #left_col #left_menu a {
76 text-decoration:none;
77 color: #f0ffff;
78 }
79 #wrapper #left_col #left_menu li {
80 font-size: 13px;
81 font-weight: bold;
82 text-decoration: none;
83 padding-top: 10px;
84 padding-bottom:10px;
85 font-family: Arial, Helvetica, sans-serif;
86 border-bottom: 3px solid #0b8403;
87 list-style: none;
88 width:100px;
89 margin-right: auto;
90 margin-left: auto;
91 }
92 #wrapper #left_col #left_menu a:visited{
93 color: #f0ffff;
94 }
95 #wrapper #content a {
96 text-decoration:none;
97 border-top-width: 0px;
98 border-right-width: 0px;
99 border-bottom-width: 0px;
100 border-left-width: 0px;
101 border: 0;
102 }
103 #wrapper #header {
104 margin-top: 0px;
105 background-image: url(../images/header.png);
106 height: 300px;
107 }
108 #wrapper #up_menu {
109 position:relative;
110 background-image: url(../images/up_menu.png);
111 background-repeat: no-repeat;
112 top: 220px;
113 height: 70px;
114 font-family: Arial, Helvetica, sans-serif;
115 font-size: 14px;
116 font-weight: bold;
117 color: #238f1d;
118 text-decoration: none;
119 }
120 #wrapper #up_menu a {
121 text-decoration:none;
122 font-family: Arial, Helvetica, sans-serif;
123 color: #238f1d;
124 }
125 #wrapper #up_menu li{
126 text-decoration:none;
127 display:inline;
128 padding-right:25px;
129 padding-left:25px;
130 height: 60px;
131 color: #238f1d;
132 }
133 #wrapper #up_menu ul{
134 height:70px;
135 padding-top:17px;
136 padding-left: 310px;
137 color: #238f1d;
138 }
139 }
140 #wrapper #left_col #left_menu a:hover{
141 color: #0b8403;
142 }
143 #wrapper #left_col #left_epik {
144 height: 209px;
145 background-image: url(../images/epikoinonia.png);
146 width: 191px;
147 }
148 #left_epik #eponumia {
149 font-family: Arial, Helvetica, sans-serif;
150 font-size: 12px;
151 font-style: normal;
152 padding-top: 25px;
153 color: #f0ffff;
154 font-weight: bold;
155 }
156 #left_epik #eponumia #kwmnos {
157 padding-left: 62px;
158 }
159 #address #ad_ath {
160 padding-left: 65px;
161 }
162 #left_epik #address {
163 font-family: Arial, Helvetica, sans-serif;
164 font-size: 11px;
165 color: #f0ffff;
166 padding-top: 60px;
167 }
168 #address #ath {
169 padding-left: 80px;
170 }
171 #address #addr_york {
172 padding-left: 45px;
173 }
174 #left_epik #tel {
175 font-family: Arial, Helvetica, sans-serif;
176 font-size: 12px;
177 font-weight: bold;
178 color: #f0ffff;
179 padding-top: 10px;
180 padding-left: 35px;
181 }
182 #left_epik #til {
183 font-family: Arial, Helvetica, sans-serif;
184 font-size: 12px;
185 font-weight: bold;
186 color: #f0ffff;
187 padding-left: 35px;
188 }
```

```

189 #address #york {
190     padding-left: 73px;
191 }
192 #left_epik #eponymia #so {
193     margin-right: auto;
194     margin-left: auto;
195     padding-left: 50px;
196 }
197 #left_epik #eponymia #office {
198     padding-left: 62px;
199 }
200
201 #wrapper #footer {
202     background-image: url(../images/footer.png);
203     height: 45px;
204     margin-top: 20px;
205 }
206 #wrapper #footer #Konso {
207     height: 45px;
208     margin-top: 0px;
209     margin-right: 7px;
210     padding-top: 0px;
211     padding-right: 7px;
212     float: right;
213     width: 180px;
214 }
215 #wrapper #footer #copyright {
216     height: 45px;
217     font-family: Arial, Helvetica, sans-serif;
218     font-size: 11px;
219     margin-top: 0px;
220     margin-left: 5px;
221     padding-top: 0px;
222     padding-left: 5px;
223     float: left;
224     color: #333333;
225 }
226 #wrapper #footer #Konso #design {
227     font-family: Arial, Helvetica, sans-serif;
228     font-size: 12px;
229     font-style: normal;
230     color: #999999;
231     float: left;
232 }
233
234 #wrapper #footer #Konso #link {
235     float: right;
236     font-family: Arial, Helvetica, sans-serif;
237     font-size: 12px;
238     font-weight: bold;
239     color: #666666;
240 }
241 #wrapper #footer #Konso #link a:hover{
242     color: #9966FF;
243 }
244 #wrapper #footer #Konso #link a {
245     text-decoration:none;
246     color: #666666;
247 }
248
249 #wrapper #footer #Konso #link :visited{
250     color: #666666;
251 }
252
253 #content .module_menu #articles {
254     padding-top: 20px;
255     padding-right: 45px;
256     padding-left: 25px;
257     font-family: Arial;
258     font-size: 14px;
259     color: #333333;
260     line-height: 20px;
261     width: 730px;
262     padding-bottom: 30px;
263 }
264

```

Εικόνα 6: Αρχείο css

HTML

Κάθε σελίδα που εμφανίζεται στο Internet είναι ένα αρχείο γραμμένο με τη γλώσσα *HTML (HyperText Markup Language, Γλώσσα Χαρακτηρισμού Υπερ-Κειμένου)*, που περιλαμβάνει το κείμενο της σελίδας, τη δομή της και τους συνδέσμους προς άλλα έγγραφα, εικόνες ή άλλα μέσα.

Ο φυλλομετρητής (Web browser) παίρνει τις πληροφορίες από τον Web server, τις μορφοποιεί και τις εμφανίζει κατάλληλα για το σύστημά μας. Διαφορετικά προγράμματα φυλλομετρητή μπορεί να μορφοποιούν και να εμφανίζουν το ίδιο αρχείο με διαφορετικό τρόπο, ανάλογα με τις δυνατότητες του συστήματος στο οποίο τρέχουν και τις επιλογές διαμόρφωσης του προγράμματος του φυλλομετρητή.

Μια *Web σελίδα* ή *ιστοσελίδα (Web page)* είναι ένα μεμονωμένο στοιχείο μιας παρουσίασης για το Web και περιέχεται σ' ένα αρχείο στον δίσκο, το οποίο ανακτάται από έναν Web server και μορφοποιείται μέσω ενός φυλλομετρητή.

Η *αρχική σελίδα (home page)* είναι η πρώτη ή κορυφαία σελίδα μιας παρουσίασης για το Web, είναι δηλαδή το σημείο εισόδου ή εκκίνησης για τις υπόλοιπες σελίδες της παρουσίασης και η πρώτη σελίδα που θα συναντήσουν οι αναγνώστες της παρουσίασής μας. Η αρχική σελίδα περιέχει συνήθως μια σύνοψη του περιεχομένου της παρουσίασης με τη μορφή ενός πίνακα περιεχομένων ή μιας ομάδας εικονιδίων.

Η HTML είναι το ακρωνύμιο των λέξεων *HyperText Markup Language*, δηλ. *Γλώσσα Χαρακτηρισμού Υπερ-Κειμένου* και βασίζεται στη γλώσσα *SGML, Standard Generalized Markup Language*, που είναι ένα πολύ μεγαλύτερο σύστημα επεξεργασίας εγγράφων.

Η HTML ορίζει ένα σύνολο κοινών στυλ για τις Web σελίδες, όπως τίτλοι (titles), επικεφαλίδες (headings), παράγραφοι (paragraphs), λίστες (lists) και πίνακες (tables). Ορίζει επίσης στυλ χαρακτήρων, όπως η έντονη γραφή (boldface) και οι ενότητες κώδικα.

Κάθε στοιχείο έχει ένα όνομα και περιέχεται μέσα στα σύμβολα <>, που αποκαλούνται *tags (ετικέτες)*. Όταν γράφουμε μια Web σελίδα με την HTML, στην ουσία δίνουμε τίτλους στα διάφορα στοιχεία της σελίδας μ' αυτά τα tags.

Οι φυλλομετρητές, μαζί με τη δυνατότητά τους να ανακτούν σελίδες από το Web, λειτουργούν επίσης και σαν μορφοποιητές για την HTML. Όταν διαβάζουμε μια σελίδα γραμμένη με την HTML σ' έναν φυλλομετρητή, ο φυλλομετρητής διαβάζει (διερμηνεύει) τα tags της HTML και μορφοποιεί το κείμενο και τις εικόνες στην οθόνη.

Διαφορετικοί φυλλομετρητές, οι οποίοι τρέχουν σε διαφορετικούς υπολογιστές, μπορεί να αντιστοιχίζουν διαφορετικά στυλ σε κάθε στοιχείο μιας σελίδας. Αυτό σημαίνει ότι οι σελίδες που δημιουργούμε με την HTML μπορεί να δείχνουν εντελώς διαφορετικές από σύστημα σε σύστημα και από φυλλομετρητή σε φυλλομετρητή. Δηλαδή, οι πραγματικές πληροφορίες και οι σύνδεσμοι που περιέχουν οι σελίδες μας θα είναι πάντα εκεί, αλλά η εμφάνιση των σελίδων στην οθόνη θα είναι διαφορετική.

Η Δομή μιας HTML Σελίδας

Οι σελίδες που γράφουμε με την HTML είναι απλά αρχεία κειμένου σε μορφή ASCII, που σημαίνει ότι δεν περιέχουν πληροφορίες για κάποιο λειτουργικό σύστημα ή πρόγραμμα, αλλά μπορούν να διαβαστούν από οποιονδήποτε συντάκτη υποστηρίζει απλό κείμενο.

Τα αρχεία της HTML περιέχουν τα ακόλουθα :

- Το κείμενο της σελίδας.
- Τα tags της HTML, τα οποία υποδεικνύουν τα στοιχεία, τη δομή και τη μορφοποίηση των σελίδων, καθώς επίσης και τους συνδέσμους υπερ-κειμένου προς άλλες σελίδες ή προς αρχεία άλλων μορφών (πολυμέσα).

Τα περισσότερα tags της HTML έχουν την εξής μορφή :

`<ΌνομαTag> επηρεαζόμενο κείμενο </ΌνομαTag>`

Τα tags της HTML έχουν γενικά ένα tag αρχής και ένα tag τέλους ή ένα tag ανοίγματος και ένα tag κλεισίματος, τα οποία περικλείουν το κείμενο που επηρεάζουν. Το tag αρχής ενεργοποιεί μια λειτουργία ή ένα χαρακτηριστικό, όπως είναι για παράδειγμα η έντονη γραφή, ενώ το tag τέλους την απενεργοποιεί. Τα tags τέλους έχουν το ίδιο όνομα με τα tags αρχής, αλλά με πρόθεμα τον χαρακτήρα /.

Δεν αποτελούν ζευγάρι όλα τα tags της HTML, καθώς ορισμένα είναι "μονομελή", ενώ άλλα περιέχουν επιπλέον πληροφορίες και κείμενο μέσα στα σύμβολα <>. Ακόμη, όλα τα tags της HTML δεν κάνουν διάκριση μεταξύ κεφαλαίων και πεζών γραμμάτων, δηλ. μπορούμε να τα γράφουμε είτε με κεφαλαίους είτε με πεζούς χαρακτήρες είτε με οποιονδήποτε συνδυασμό τους. Συνήθως τα γράφουμε κεφαλαία για να μπορούμε να τα ξεχωρίζουμε ευκολότερα από το κείμενο της σελίδας.

Τα αρχεία της HTML πρέπει να τα γράφουμε σε συντάκτες κειμένων, δηλ. προγράμματα που μπορούν να αποθηκεύσουν αρχεία κειμένου σε μορφή ASCII. Τέτοια προγράμματα είναι τα Notepad, Wordpad και Write των Windows, το Edit του DOS και τα vi, emacs και pico του UNIX.

PHP

Η *PHP*, όπου τα αρχικά σημαίνουν *Hypertext PreProcessor*, είναι μια γλώσσα συγγραφής σεναρίων (scripting language) που ενσωματώνεται μέσα στον κώδικα της HTML και εκτελείται στην πλευρά του server (server-side scripting). Ανταγωνιστικές της τεχνολογίας PHP είναι οι εξής γλώσσες προγραμματισμού : *ASP (Active Server Pages)* της εταιρείας Microsoft, *CFML (ColdFusion Markup Language)* της εταιρείας Allaire και *JSP (JavaServer Pages)* της εταιρείας Sun.

Το μεγαλύτερο μέρος της σύνταξής της, η PHP το έχει δανειστεί από την C, την Java και την Perl και διαθέτει και μερικά δικά της μοναδικά χαρακτηριστικά. Ο σκοπός της γλώσσας είναι να δώσει τη δυνατότητα στους web developers να δημιουργούν δυναμικά παραγόμενες ιστοσελίδες.

Ακολουθεί ένα εισαγωγικό παράδειγμα :

```
<html>

  <head>

    <title> Παράδειγμα </title>

  </head>

  <body>

    <?php echo "Γεια σας, είμαι ένα script της PHP!"; ?>

  </body>

</html>
```

Προσέξτε πόσο διαφέρει από ένα CGI script που γράφεται σ' άλλες γλώσσες, όπως η Perl ή η C, όπου αντί να γράψουμε ένα πρόγραμμα με πολλές εντολές για να δημιουργήσουμε κώδικα HTML, γράψουμε ένα HTML script με κάποιον ενσωματωμένο κώδικα για να κάνει κάτι, όπως στην συγκεκριμένη περίπτωση να εμφανίσει κάποιο κείμενο (μήνυμα). Ο κώδικας της PHP περικλείεται με ειδικά tags αρχής και τέλους για να μπορούμε να εισερχόμαστε και να εξερχόμαστε από το PHP mode.

Αυτό που ξεχωρίζει την PHP από μια γλώσσα όπως η JavaScript, η οποία εκτελείται στην πλευρά του χρήστη (client-side), είναι ότι ο κώδικάς της εκτελείται στον server. Αν είχαμε σ' έναν server ένα script παρόμοιο με το παραπάνω, ο χρήστης (client) θα λάμβανε το αποτέλεσμα της εκτέλεσης αυτού του script, χωρίς να είναι σε θέση να γνωρίζει ποιος μπορεί να είναι ο αρχικός κώδικας.

Μπορούμε ακόμη να ρυθμίσουμε (configure) τον web server ώστε να επεξεργάζεται όλα τα HTML αρχεία με την PHP και τότε δεν θα υπάρχει πράγματι κανένας τρόπος να μάθουν οι χρήστες τον κώδικά μας.

Στο πιο βασικό επίπεδο, η PHP μπορεί να κάνει ό,τι και τα άλλα προγράμματα της τεχνολογίας CGI, όπως επεξεργασία των δεδομένων μιας φόρμας, δημιουργία δυναμικού περιεχομένου ιστοσελίδων ή αποστολή και λήψη cookies.

Ίσως το δυνατότερο και πιο σημαντικό χαρακτηριστικό της PHP είναι η υποστήριξη που παρέχει σε μια ευρεία γκάμα από βάσεις δεδομένων. Έτσι, το να δημιουργήσουμε μια ιστοσελίδα που να παρέχει υποστήριξη σε βάσεις δεδομένων είναι απίστευτα απλό. Υποστηρίζει τις εξής βάσεις δεδομένων :

Adabas D	dBase	Empress	FilePro	Informix	InterBase	mSQL
MySQL	Oracle	PostgreSQL	Solid	Sybase	Velocis	Unix dbm

Η PHP παρέχει επίσης υποστήριξη για συνομιλία μ' άλλες υπηρεσίες, χρησιμοποιώντας πρωτόκολλα όπως τα IMAP, SNMP, NNTP, POP3 ή και το HTTP.

Μια Σύντομη Ιστορία της PHP

Η ιδέα για την δημιουργία της PHP ελήφθη το φθινόπωρο του 1994 από τον *Rasmus Lerdorf*. Οι πρώτες ανεπίσημες εκδόσεις (versions) της PHP χρησιμοποιήθηκαν στην αρχική του σελίδα (home page) για να μπορεί να παρακολουθεί αυτούς που έμπαιναν στην σελίδα. Η πρώτη έκδοση που δόθηκε για χρήση στο κοινό ήταν διαθέσιμη στις αρχές του 1995 με το όνομα *Personal Home Page Tools*.

Αποτελείτο από μια πολύ απλοϊκή μηχανή ανάλυσης (parser engine) η οποία καταλάβαινε λίγες μόνο ειδικές μακροεντολές (macros) και έναν αριθμό από utilities που βρίσκονταν σε κοινή χρήση στις home pages εκείνη την εποχή. Ένα guestbook, ένας μετρητής (counter) και κάποιο άλλο υλικό. Ο αναλυτής (parser) ξαναγράφηκε στα μέσα του 1995 και ονομάστηκε *PHP/FI Version 2*.

Το όνομα FI προέρχεται από ένα άλλο πακέτο που είχε γράψει ο Rasmus και το οποίο διερμήνευε (interpreted) τα δεδομένα από φόρμες της HTML. Συνδύασε τα εργαλεία scripts της Personal Home Page με τον *Form Interpreter* και πρόσθεσε υποστήριξη για mSQL. Έτσι γεννήθηκε η *PHP/FI*, η οποία αναπτύχθηκε αλματωδώς και διάφοροι χρήστες άρχισαν να συνεισφέρουν κώδικα σ' αυτήν.

Υπολογίζεται ότι μέχρι τα τέλη του 1996, η PHP/FI χρησιμοποιείτο σε τουλάχιστον 15.000 web sites σ' όλον τον κόσμο και στα μέσα του 1997 αυτός ο αριθμός είχε ξεπεράσει τις 50.000. Στα μέσα του 1997 είχαμε επίσης μια αλλαγή στην ανάπτυξη της PHP.

Σταμάτησε να αποτελεί το αγαπημένο αντικείμενο του Rasmus και έγινε ο στόχος μιας πιο καλά οργανωμένης ομαδικής εργασίας. Ο αναλυτής (parser) ξαναγράφηκε από την αρχή από τους Zeev Suraski και Andi Gutmans και αυτός ο νέος parser αποτέλεσε τη βάση για την PHP Version 3. Ένα μεγάλο μέρος του utility code μεταφέρθηκε από την PHP/FI στην PHP3 και ένα μεγάλο μέρος του ξαναγράφηκε από την αρχή.

Σήμερα, η PHP/FI και η PHP3 έρχονται μ' έναν αριθμό εμπορικών προϊόντων όπως ο web server C2 StrongHold και το RedHat Linux. Σύμφωνα με μια συντηρητική εκτίμηση, η PHP χρησιμοποιείται από περισσότερα από 150.000 sites σ' όλον τον κόσμο.

Joomla

Το Joomla είναι ένα λογισμικό διαχείρισης περιεχομένου ανοικτού κώδικα με αρκετές διακλαδώσεις από το Mambo . Είναι μία από τις πιο δημοφιλείς λύσεις CMS στον κόσμο και μπορεί να υπερηφανεύεται για τα πάνω από 30 εκατ. downloads μέχρι σήμερα. Θα πρέπει να σημειωθεί κάτι το οποίο δεν είναι τυχαίο το Cloud.com και το Linux.com όπως και άλλα πολύ μεγάλα site είναι κατασκευασμένα σε Joomla.

Πλεονεκτήματα του Joomla

- **Φιλικό προς το χρήστη:** Το Joomla δεν είναι WordPress , αλλά είναι ακόμα σχετικά εύκολο στη χρήση.
- **Ισχυρή Κοινότητα προγραμματιστών:** Όπως το WordPress , έτσι και το Joomla έχει μια ισχυρή κοινότητα προγραμματιστών. Η βιβλιοθήκη plugin (που ονομάζεται « extensions ») είναι πολύ μεγάλη, με έναν τόνο από δωρεάν προς χρήση, plugins.
- **Επέκταση και Μεταβλητότητα:** Οι επεκτάσεις Joomla χωρίζονται σε πέντε κατηγορίες - components, plugins, templates, ενότητες και γλώσσες. Κάθε ένα από αυτά διαφέρει σε λειτουργία, τη

δύναμη και την ικανότητα. Τα Components, για παράδειγμα, λειτουργούν ως « μίνι - εφαρμογές », που μπορούν να αλλάξουν την εγκατάσταση του Joomla συνολικά. Οι Ενότητες, από την άλλη πλευρά, προσθέτουν δευτερεύουσες δυνατότητες όπως δυναμικό περιεχόμενο, RSS feeds, και λειτουργία αναζήτησης στην ιστοσελίδα σας.

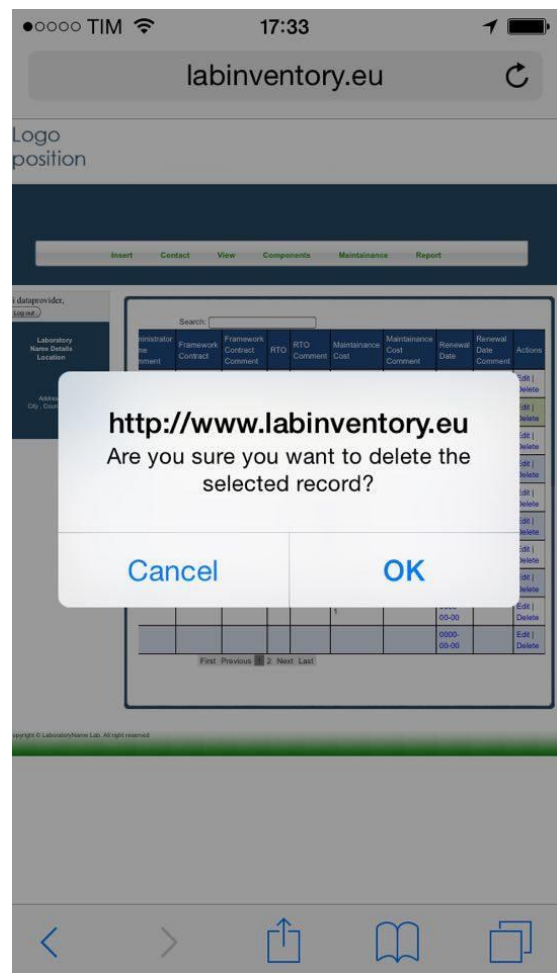
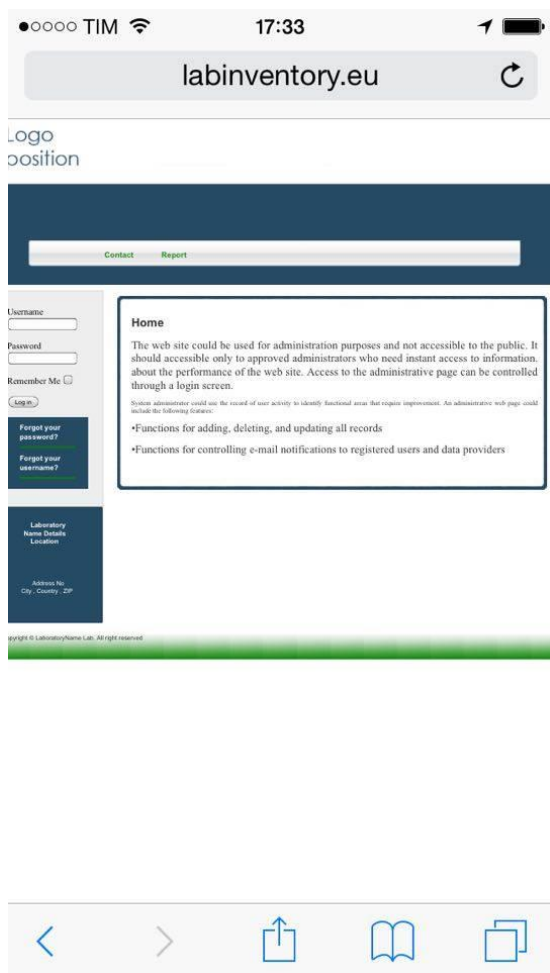
• **Ισχυρές δυνατότητες διαχείρισης περιεχομένου:** Σε αντίθεση με WordPress, το Joomla αρχικά είχε σχεδιαστεί ως ένα επιχειρησιακό CMS όπως το drupal. Αυτό το καθιστά πολύ πιο ικανό στο χειρισμό μεγάλου όγκου των αντικειμένων από WordPress.

Joomla 3 – Responsitive Design

Το μεγαλύτερο ίσως από τα πλεονεκτήματα της συγκεκριμένης έκδοσης που χρησιμοποιήθηκε για την κατασκευή της συγκεκριμένης εφαρμογής είναι το Responsive Design Η **κατασκευή ιστοσελίδων με responsive design** (προσαρμοστική σχεδίαση ιστοσελίδων) είναι η πλέον σύγχρονος, απαιτητικός και απαραίτητος τρόπος σχεδίασης ιστοσελίδων. Το responsive design ήρθε στο προσκήνιο με την καθιέρωση των φορητών συσκευών, smartphone και tablet, ως συσκευές περιήγησης στο Internet, συσκευές που πλέον χρησιμοποιούνται από την πλειονότητα των υποψηφίων πελατών σας.

Πολύ απλά θα μπορούσαμε να ορίσουμε το **responsive design** ως εκείνη την μέθοδο για την **κατασκευή ιστοσελίδων** που επιτρέπει στην ιστοσελίδα να αναγνωρίζει με τι είδους συσκευή επισκέπτεται το site ο χρήστης και να προσαρμόζει την εμφάνιση της ανάλογα. Αν έχετε χρησιμοποιήσει smartphone ή tablet για περιήγηση στο internet θα έχετε δει ιστοσελίδες στις οποίες βλέπετε ένα μόνο τμήμα της ιστοσελίδας ή χρειάζεται να κάνετε ζουμ για να μπορέσετε να διαβάσετε το περιεχόμενο. Αυτό συμβαίνει γιατί **δεν έγινε κατασκευή ιστοσελίδων με responsive design** και αυτό έχει ως αποτέλεσμα την δυσκολία περιήγησης στην ιστοσελίδα.

Καθώς ο διαθέσιμος χώρος στην οθόνη ενός smartphone είναι αρκετά μικρότερος από την οθόνη ενός desktop υπολογιστή, η παρουσίαση μιας ιστοσελίδας σχεδιασμένης για μεγάλες αναλύσεις (οθόνες laptop και desktop) είναι συχνά προβληματική στην οθόνη ενός smartphone ή tablet. Επίσης μην ξεχνάμε τον διαφορετικό τρόπο πλοήγησης: μια συσκευή αφής δεν έχει ποντίκι αλλά χρησιμοποιεί τα δάχτυλα, που συχνά δυσκολεύονται να επιλέξουν με ακρίβεια μικρά στοιχεία πλοήγησης (συνδέσμους σε κείμενο, μικρά μενού κλπ.). Είναι σαφές πως μια εικαστική και λειτουργική προσέγγιση που λειτουργεί άριστα σε desktop περιβάλλον συχνά έχει σοβαρά ζητήματα χρηστικότητας σε κινητές συσκευές.



Εικόνα 9: Η εφαρμογή απο mobile browser

Αν δούμε τα στατιστικά στοιχεία, τον τελευταίο χρόνο, οι φορητές συσκευές (smartphone και tablet) χρησιμοποιήθηκαν για το 15% περίπου όλης της περιήγησης στο internet παγκοσμίως (gs statcounter com). Με την χρήση **προσαρμοστικής σχεδίασης στην κατασκευή ιστοσελίδων** μπορείτε να μπείτε σε μια δυναμική νέα αγορά στην οποία πολλοί ανταγωνιστές σας δεν έχουν την ίδια καλή απόδοση. Ο σκοπός του **responsive design** είναι να προσφέρει στους χρήστες της ιστοσελίδας σας την καλύτερη δυνατή εμπειρία περιήγησης.

SQL

Η SQL αποτελεί μια στάνταρτ γλώσσα του ANSI για να μπορούμε να έχουμε πρόσβαση σε βάσεις δεδομένων.

Τα αρχικά SQL σημαίνουν *Structured Query Language*, δηλ. *Δομημένη Γλώσσα Ερωτημάτων*.

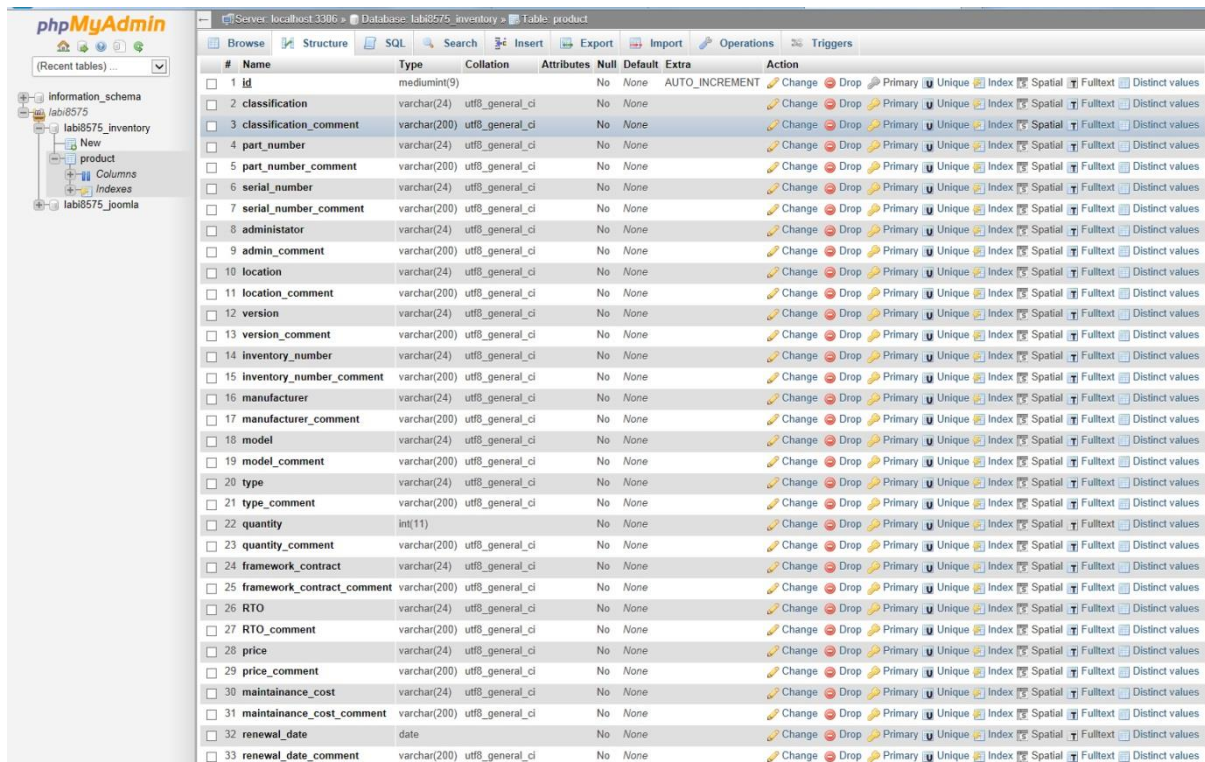
- Η SQL μάς δίνει τη δυνατότητα να έχουμε πρόσβαση σε μια βάση δεδομένων (database).
- Η SQL αποτελεί μια στάνταρτ γλώσσα του ANSI (ANSI standard language).
- Η SQL μπορεί να εκτελέσει ερωτήματα (queries) σχετικά με μια βάση δεδομένων.
- Η SQL μπορεί να ανακτήσει δεδομένα από μια βάση δεδομένων.
- Η SQL μπορεί να εισαγάγει νέες εγγραφές σε μια βάση δεδομένων.
- Η SQL μπορεί να διαγράψει εγγραφές από μια βάση δεδομένων.
- Η SQL μπορεί να ενημερώσει εγγραφές σε μια βάση δεδομένων.
- Η SQL είναι πολύ εύκολη στην εκμάθηση.

Η SQL αποτελεί ένα στάνταρτ του ANSI (American National Standards Institute) για να μπορούμε να έχουμε πρόσβαση σε συστήματα βάσεων δεδομένων. Οι εντολές της SQL χρησιμοποιούνται για να ανακτήσουμε (retrieve) και να ενημερώσουμε (update) δεδομένα σε μια βάση δεδομένων (database).

Η SQL συνεργάζεται με προγράμματα βάσεων δεδομένων όπως είναι τα εξής : Access, Informix, Microsoft SQL Server, Oracle, Sybase και πολλά άλλα.

Οι βάσεις δεδομένων (databases) περιέχουν αντικείμενα (objects) που ονομάζονται Πίνακες (Tables). Οι Εγγραφές (Records) των δεδομένων αποθηκεύονται σ' αυτούς τους πίνακες. Οι Πίνακες αναγνωρίζονται με τα ονόματά τους, όπως "Persons", "Orders", "Suppliers" κ.ά.

Οι Πίνακες περιέχουν Στήλες (Columns) και Γραμμές (Rows) με δεδομένα. Οι Γραμμές (Rows) περιέχουν εγγραφές (records), όπως μία εγγραφή για κάθε άτομο. Οι Στήλες (Columns) περιέχουν δεδομένα, όπως First Name, Last Name, Address και City.



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	mediumint(9)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	classification	varchar(24)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	classification_comment	varchar(200)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	part_number	varchar(24)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
5	part_number_comment	varchar(200)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
6	serial_number	varchar(24)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
7	serial_number_comment	varchar(200)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
8	administrator	varchar(24)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
9	admin_comment	varchar(200)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
10	location	varchar(24)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
11	location_comment	varchar(200)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
12	version	varchar(24)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
13	version_comment	varchar(200)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
14	inventory_number	varchar(24)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
15	inventory_number_comment	varchar(200)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
16	manufacturer	varchar(24)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
17	manufacturer_comment	varchar(200)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
18	model	varchar(24)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
19	model_comment	varchar(200)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
20	type	varchar(24)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
21	type_comment	varchar(200)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
22	quantity	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
23	quantity_comment	varchar(200)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
24	framework_contract	varchar(24)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
25	framework_contract_comment	varchar(200)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
26	RTO	varchar(24)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
27	RTO_comment	varchar(200)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
28	price	varchar(24)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
29	price_comment	varchar(200)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
30	maintenance_cost	varchar(24)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
31	maintenance_cost_comment	varchar(200)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
32	renewal_date	date			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
33	renewal_date_comment	varchar(200)	utf8_general_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Εικόνα 10: Στιγμιότυπο από την βάση δεδομένων της εφαρμογής

Περιγραφή της εφαρμογής.

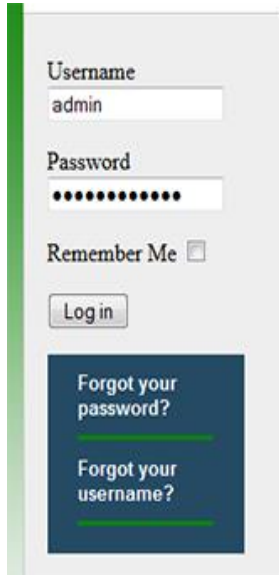
Η ηλεκτρονική διεύθυνση της εφαρμογής είναι labinventory.eu. Το πρώτο πράγμα που βλέπει ο χρήστης με το που πληκτρολογεί την ηλεκτρονική διεύθυνση είναι μια σύντομη περιγραφή για τον σκοπό της εφαρμογής.

Λειτουργικότητα της ιστοσελίδας.

Με το που μπαίνει ο χρήστης στην αρχική σελίδα , διαβάζει μια σύντομη περιγραφή με τον σκοπό και την λειτουργία της ιστοσελίδας. Εν συντομία ο σκοπός λειτουργίας της συγκεκριμένης ιστοσελίδας είναι να διευκολύνει στην καταγράψει εξοπλισμού ενός εργαστηρίου., είτε λογισμικού είτε υλικού.

Ωστόσο η εφαρμογή έχει σχεδιαστεί ώστε να προσαρμόζεται εύκολα και για άλλες εφαρμογές διαχείρισης δεδομένων.

Φόρμα εισόδου χρηστών / Login form



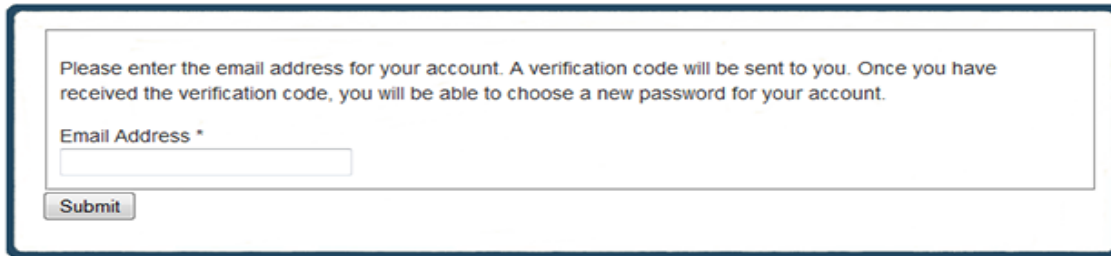
Στην αριστερή πλευρά της ιστοσελίδας ο χρήστης μπορεί να δει την φόρμα εισόδου χρηστών. Να διευκρινιστεί ότι η συγκεκριμένη εφαρμογή σχεδιάστηκε για τις ανάγκες συγκεκριμένου εργαστηρίου. Οπότε η απογραφή του εξοπλισμού αφορά πολύ συγκεκριμένο αριθμό χρηστών , που παράλληλα είναι οι εργαζόμενοι στο συγκεκριμένο εργαστήριο. Συνεπώς ο αριθμός των χρηστών είναι ήδη γνωστός και ο διαχειριστής της εφαρμογής μπορεί να δημιουργήσει αντίστοιχους χρήστες .

Στην συγκεκριμένη εφαρμογή υπάρχει ιεραρχία χρηστών ,οι τρεις κατηγορίες χρηστών είναι

- Ο administrator : είναι υπεύθυνος για την διαχείριση όχι μόνο του Frontend αλλά και του Backend. Είναι επίσης αυτός που λαμβάνει τα email-απορίες των χρηστών μέσω της φόρμας επικοινωνίας. Τέλος είναι ο μόνος που είναι αρμόδιος για την προσθήκη νέων σελιδων ή άρθρων στην εφαρμογή και αρμόδιος για τυχόν υλοποίηση επέκταση της εφαρμογής και συντήρηση
- Ο dataprovider: η βασική αρμοδιότητα του είναι να παρέχει πληροφορία – δεδομένα. Με το που εισάγει τα στοιχεία του σωστά μπορεί να διακρίνει στο πάνω μενού δύο νέες καρτέλες Import και Inventory. Στην καρτέλα Import υπάρχει μια φόρμα εισαγωγής δεδομένων , ο χρήστης γράφει τα δεδομένα και πατώντας το κουμπί Import τα δεδομένα του καταχωρούνται στην βάση δεδομένων. Η διπλανή καρτέλα είναι η Inventory , όπου με την μορφή πίνακα προβάλλονται όλα τα δεδομένα που έχουν καταχωρηθεί ήδη στην βάση δεδομένων. Ωστόσο υπάρχει μια διαφοροποίηση για τους dataprovider σε σχέση με τους απλούς χρήστες όσον αφορά τις δύο τελευταίες στήλες του πίνακα. Συγκεκριμένα, μπορείτε να διακρίνετε δυο στήλες σε κάθε πλειάδα του πίνακα την διαγραφή και την τροποποίηση. Επιλέγοντας την διαγραφή η συγκεκριμένη γραμμή – καταχώρηση διαγράφεται ενώ πατώντας κλικ στην μορφοποίηση ο χρήστης μεταφέρεται σε μια νέα καρτέλα, παρόμοια με την φόρμα εισαγωγής που αναλύθηκε πριν, με την διαφορά ότι είναι καταγεγραμμένα τα πεδία που έχουν ήδη συμπληρωθεί , ωστόσο ο χρήστης μπορεί είτε να τα τροποποιήσει είτε να τα διαγράψει είτε να συμπληρώσει νέα πεδία που ήταν κενά.
- Ο dataviewer : ο συγκεκριμένος χρήστης αφού συνδεθεί μπορεί να δει τα δεδομένα που έχουν καταχωρηθεί στην βάση δεδομένων , ωστόσο δεν μπορεί να τα τροποποιήσει να τα διαγράψει ή να εισάγει νέα δεδομένα.

Σε περίπτωση που ο χρήστης έχει ξεχάσει το όνομα χρήστη ή τον κωδικό του. Μπορεί να πατήσει ένα από τα βοηθητικά links που βρίσκονται στο κάτω μέρος της εφαρμογής.

Αυτομάτως ο χρήστης μεταφέρεται σε μία νέα σελίδα όπου καλείται να συμπληρώσει το email με το οποίο έχει εγγραφεί σαν χρήστης.



Please enter the email address for your account. A verification code will be sent to you. Once you have received the verification code, you will be able to choose a new password for your account.

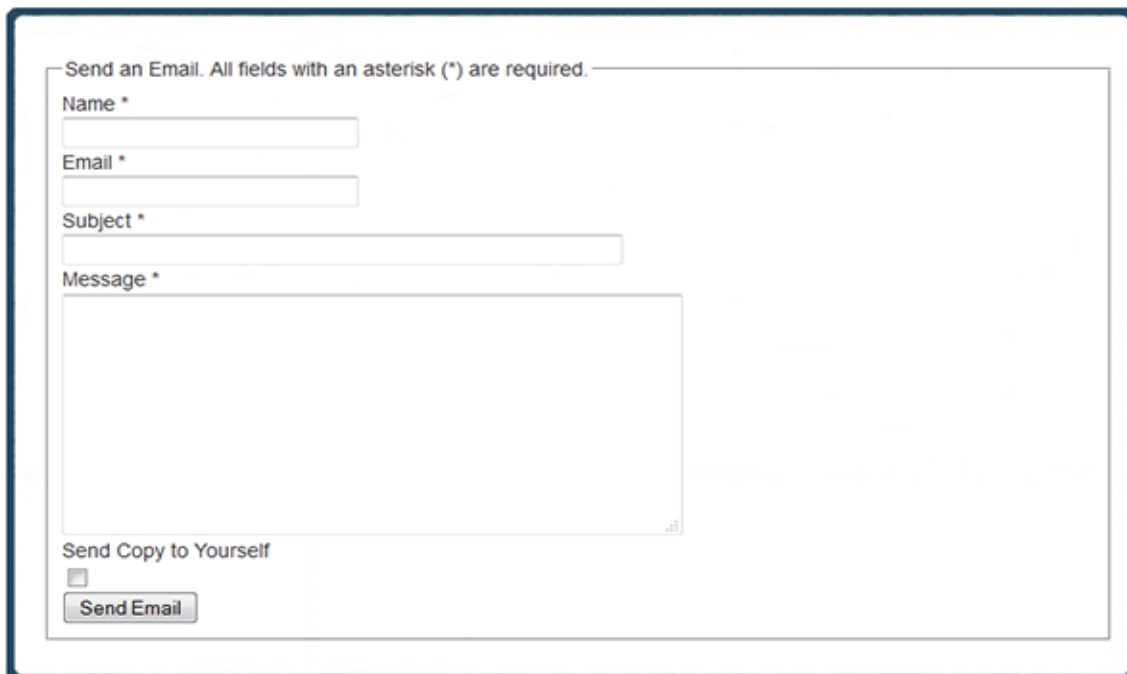
Email Address *

Submit

Εικόνα 11: Φόρμα δημιουργίας νέου χρήστη

Φόρμα επικοινωνίας

Μέσω της φόρμας επικοινωνίας δίνεται η δυνατότητα όχι μόνο στους ήδη καταγεγραμμένους χρήστες της εφαρμογής αλλά επίσης σε όλους όσους γνωρίζουν την ιστοσελίδα να αποστείλουν email στον διαχειριστή προκειμένου να εκφράσουν απορίες , να ζητήσουν βοήθεια ή διευκρινίσεις ή ακόμη και να σχολιάσουν ή να προτείνουν διορθώσεις. Ο χρήστης μπορεί να μεταβεί στην φόρμα επικοινωνίας ακόμη και όταν δεν είναι συνδεδεμένος , πατώντας το κουμπί Contacts στο κεντρικό μενού της ιστοσελίδας.



Send an Email. All fields with an asterisk (*) are required.

Name *

Email *

Subject *

Message *

Send Copy to Yourself

Send Email

Εικόνα 12: Φόρμα επικοινωνίας

Με το που αποστέλλει ο χρήστης ένα email ο διαχειριστής λαμβάνει ένα email στο προσωπικό του ηλεκτρονικό ταχυδρομείο . Ο λόγος που τα λαμβάνει στο προσωπικό του ταχυδρομείο είναι η πιο εύκολη , γρήγορη και άμεση πρόσβαση του για τον καλύτερο συντονισμό.

Φόρμα εισαγωγής δεδομένων

Μέσω της συγκεκριμένης φόρμας , οι dataproviders καταχωρούν τα δεδομένα τους σε μια συνολική βάση δεδομένων . Σε αυτό το σημείο να διευκρινιστεί ότι αν και η ιστοσελίδα έχει διαμορφωθεί ώστε το back end – διαχειριστικό κομμάτι της να υποστηρίζεται από το CMS Joomla , ;έχει δημιουργηθεί μια εξωτερική βάση δεδομένων και όχι αυτήν του Joomla. Με αυτόν τον τρόπο μπορούμε να έχουμε άμεσα Back up της βάσης καθώς επίσης οι καταχωρήσεις είναι ξεκάθαρες και διαχωρισμένες από τα υπόλοιπα δεδομένα της εφαρμογής και του συστήματος διαχείρισης Joomla.

Η φόρμα εισαγωγής δεδομένων αποτελείται από 12 πεδία και για κάθε πεδίο υπάρχει και ένα πεδίο σχολιασμού. Η συμπλήρωση των περισσότερων πεδίων δεν είναι υποχρεωτική , μόνο δύο πεδία είναι υποχρεωτικά (classification , part number) το part number συγκεκριμένα λειτουργεί και σαν κλειδί της οντότητας.

Details	Value	Comment
Classification*	Router	
Part number*		
Serial number		
Version		
Inventory number		
Manufacturer		
Model		
Type		
Framework contract		
RTO		
Price		
Description		

The fields with * are mandatory

Εικόνα 13: Φόρμα εισαγωγής δεδομένων

Κώδικας για φόρμα εισαγωγής δεδομένων

```
2 <script type="text/javascript">
3   function open_page()
4   {
5
6       window.open('http://labinventory.eu/import-control.php', '_self', true)
7
8   }
9
10  function check(){
11      perfect=true;
12
13      par=document.getElementById("par").value;
14      if (descr.length ==0){
15          alert("The field Classification is mandatory!");
16          perfect=false;
17      }
18      if (val.length ==0){
19          alert("The field Part number is mandatory");
20          perfect=false;
21      }
22
23
24      return perfect;
25  }
26  </script>
27  <form method = "post" action = "http://labinventory.eu/import-control.php" >
28  <table style="border-color: #218e1b; border-width: 0px; background-color: #e9e7e7; ; width: 720px;" border="0" align="center">
29  <tbody>
30  <tr>
31  <td style="width: 150px;"><span style="color: #254a64;"><strong>Details</strong></span></td>
32  <td style="background-color: #e7e7e4;"><span style="color: #254a64;"><strong>Value</strong></span></td>
33  <td style="background-color: #e7e7e4;"><span style="color: #254a64;"><strong>Comment</strong></span></td>
34  </tr>
35  <tr>
36  <td style="width: 150px;"><span style="color: #254a64;">Classification*</span></td>
37  <td style="background-color: #ffffff; width: 200px;">
38  <select name="cla">
39      <option> Router</option>
40      <option> Switch</option>
41      <option> Server</option>
42      <option> UPS</option>
43      <option> Module Card</option>
44      <option> Software</option>
45      <option> Rack</option>
46      <option> Other</option>
47  </select>
48
49  </td>
```

```

50 <td style="background-color: #ffffff;"><input id="cla_com" name="comment" type="text" maxlength="230"/> </td>
51 </tr>
52 <tr>
53 <td style="width: 150px;"><span style="color: #254a64;">Part number</span></td>
54 <td style="background-color: #ffffff; width: 200px;"><input id="par" name="par" type="text" maxlength="230"/> </td>
55 <td style="background-color: #ffffff;"><input id="par_com" name="par_com" type="text" maxlength="230"/> </td>
56 </tr>
57 <tr>
58 <td style="width: 150px;"><span style="color: #254a64;">Serial number </span></td>
59 <td style="background-color: #ffffff; width: 200px;"><input id="ser" name="ser" type="text" maxlength="230"/> </td>
60 <td style="background-color: #ffffff;"><input id="ser_com" name="ser_com" type="text" maxlength="230"/> </td>
61 </tr>
62 <tr>
63 <td style="width: 150px;"><span style="color: #254a64;">Version</span></td>
64 <td style="background-color: #ffffff; width: 200px;"><input id="ver" name="ver" type="text" maxlength="230"/> </td>
65 <td style="background-color: #ffffff;"><input id="ver_com" name="ver_com" type="text" maxlength="230"/> </td>
66 </tr>
67 <tr>
68 <td style="width: 150px;"><span style="color: #254a64;">Inventory number</span></td>
69 <td style="background-color: #ffffff; width: 200px;"><input id="inv" name="inv" type="text" maxlength="230"/> </td>
70 <td style="background-color: #ffffff;"><input id="inv_com" name="inv_com" type="text" maxlength="230"/> </td>
71 </tr>
72 <tr>
73 <td style="width: 150px;"><span style="color: #254a64;">Manufacturer</span></td>
74 <td style="background-color: #ffffff; width: 200px;"><input id="man" name="man" type="text" maxlength="230"/> </td>
75 <td style="background-color: #ffffff;"><input id="man_com" name="man_com" type="text" maxlength="230"/> </td>
76 </tr>
77 <tr>
78 <td style="width: 150px;"><span style="color: #254a64;">Model</span></td>
79 <td style="background-color: #ffffff; width: 200px;"><input id="mod" name="mod" type="text" maxlength="230"/> </td>
80 <td style="background-color: #ffffff;"><input id="mod_com" name="mod_com" type="text" maxlength="230"/> </td>
81 </tr>
82 <tr>
83 <td style="width: 150px;"><span style="color: #254a64;">Type</span></td>
84 <td style="background-color: #ffffff; width: 200px;"><input id="typ" name="typ" type="text" maxlength="230"/> </td>
85 <td style="background-color: #ffffff;"><input id="typ_com" name="typ_com" type="text" maxlength="230"/> </td>
86 </tr>
87 <tr>
88 <td style="width: 150px;"><span style="color: #254a64;">Framework contract</span></td>
89 <td style="background-color: #ffffff; width: 200px;"><input id="fra" name="fra" type="text" maxlength="230"/> </td>
90 <td style="background-color: #ffffff;"><input id="fra_com" name="fra_com" type="text" maxlength="230"/> </td>
91 </tr>
92 <tr>
93 <td style="width: 150px;"><span style="color: #254a64;">RTO</span></td>
94 <td style="background-color: #ffffff; width: 200px;"><input id="rto" name="rto" type="text" maxlength="230"/> </td>
95 <td style="background-color: #ffffff;"><input id="rto_com" name="rto_com" type="text" maxlength="230"/> </td>
96 </tr>
97 <tr>
98 <td style="width: 150px;"><span style="color: #254a64;">Price</span></td>
99 <td style="background-color: #ffffff;"><input id="pri" name="pri" type="text" maxlength="230"/> </td>
100 <td style="background-color: #ffffff;"><input id="pri_com" name="pri_com" type="text" maxlength="230"/> </td>
101 </tr>
102 </tbody>
103 </table>
104
105 <table style="border-color: #218e1b; border-width: 0px; background-color: #e9e7e7; width: 720px;" border="0" align="center">
106 <tbody>
107 <tr>
108 <td style="width: 150px;"><span style="color: #254a64;">Description</span></td>
109 <td style="background-color: #ffffff;"><input id="typ_com" name="typ_com" type="text" maxlength="400"/> </td>
110 </tr>
111 </tbody>
112 </table>
113 <table style="border-color: #218e1b; border-width: 0px; background-color: #ffffff; width: 720px;" border="0" align="center">
114 <tbody>
115 <tr>
116 <td style="background-color: #ffffff;"><span style="color: #254a64;">The fields with * are mandatory</span></td>
117 </tr>
118 </tbody>
119 </table>
120 <table style="border-color: #218e1b; border-width: 0px; background-color: #ffffff; width: 100px;" border="0" align="center">
121 <tbody>
122 <tr>
123 <td style="background-color: #ffffff;"><span style="color: #254a64;" align="center"> <input type="submit" value="import" class="input" /> </td>
124 </tr>
125 </tbody>
126 </table>
127 </form>

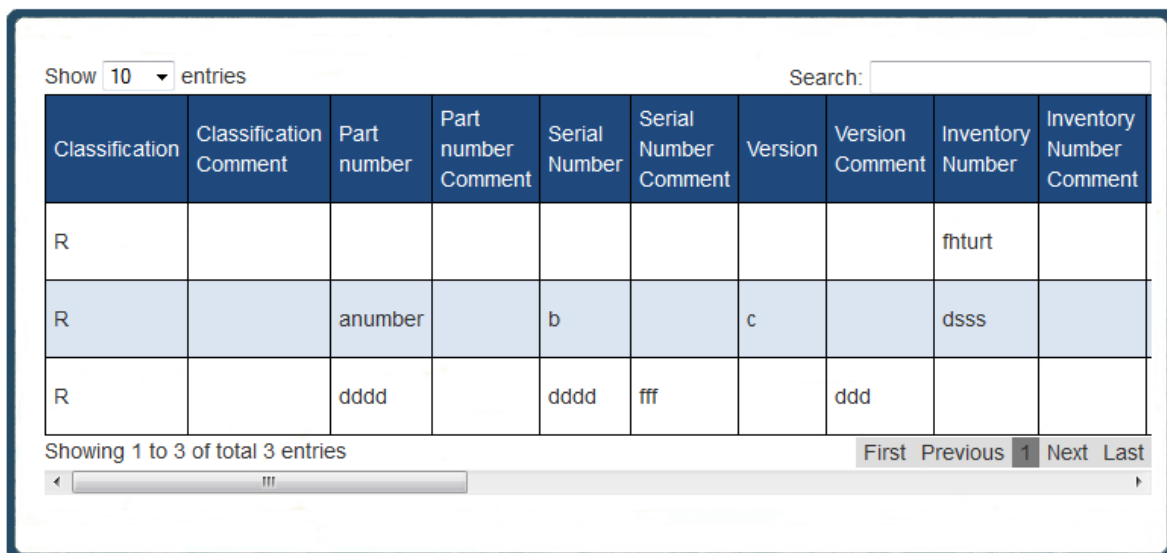
```

Εικόνα 14: Κώδικας εισαγωγής δεδομένων

Πίνακας απογραφής

Στην καρτέλα Inventory προβάλλονται όλες οι εγγραφές που έχουν καταχωρηθεί στην βάση δεδομένων.

Στην πάνω δεξιά μεριά του πίνακα βρίσκεται ένα εργαλείο αναζήτησης, προκειμένου να διευκολύνει την αναζήτηση συγκεκριμένων εγγραφών ή ομάδων εγγραφών με συγκεκριμένα κοινά χαρακτηριστικά. Ενώ στην πάνω αριστερή μεριά του πίνακα δίνεται η δυνατότητα να επιλέξει ο χρήστης το όριο το γραμμών του πίνακα



Classification	Classification Comment	Part number	Part number Comment	Serial Number	Serial Number Comment	Version	Version Comment	Inventory Number	Inventory Number Comment
R								fhturt	
R		anumber		b		c		dsss	
R		dddd		dddd	fff		ddd		

Showing 1 to 3 of total 3 entries

First Previous 1 Next Last

Εικόνα 15: Πίνακας απογραφής δεδομένων

Η τελευταία στήλη του πίνακα ονομάζεται Action και είναι ορατή μόνο στους χρήστες με δικαιώματα dataprovider. Στην συγκεκριμένη στήλη αναγράφονται δύο links ένα διαγραφή και ένα τροποποίηση. Επιλέγοντας διαγραφή, διαγράφεται από τον πίνακα απογραφής και από την βάση δεδομένων η συγκεκριμένη εγγραφή και εμφανίζεται μήνυμα επιβεβαίωσης. Επιλέγοντας την επιλογή τροποποίησης ο χρήστης μεταφέρεται σε μια νέα καρτέλα ίδια με την φόρμα εισαγωγής δεδομένων που αναλύσαμε προηγουμένως με την διαφορά ότι είναι ήδη συμπληρωμένη με τα πεδία της συγκεκριμένης εγγραφής. Από εκεί ο χρήστης μπορεί είτε να προσθέσει νέα , είτε να διαγράψει , είτε να τροποποιήσει τα δεδομένα.

Model Comment	Type	Type Comment	Framework Contract	Framework Contract Comment	RTO	RTO Comment	Price	Price Comment	Description	Actions
			jtjtjktjtj							Edit Delete
	guuuyu		h		i		j		last	Edit Delete
										Edit Delete

Εικόνα 16: Δυνατότητα επεξεργασίας / διαγραφής στοιχείων του πίνακα

Να διευκρινιστεί ότι προκειμένου να ελέγξουμε την ιεραρχία χρηστών και να αποτρέψουμε την πρόσβαση των dataviewers στην τελευταία στήλη του πίνακα απογραφής προσθέσαμε php κώδικα στο αρχείο id2links.php που αναφέρθηκε παραπάνω και μέσω του διαχειριστικού του cms Joomla ορίσαμε τις ομάδες χρηστών. Το σημείο του κώδικα όπου καθορίζεται ο αριθμός των στηλών του πίνακα απογραφής ανάλογα με την ιεραρχία χρηστών που αναλύθηκε προηγουμένως είναι το εξής.

```

1 <?php
2 class TabulizerModifyId2links {
3     function modify(&$rows, $args = null) {
4         if (empty($rows)) return;
5
6         $num_of_rows = count($rows);
7         $num_of_columns = count($rows[1]);
8
9         $edit_link = 'http://www.labinventory.eu/index.php/import';
10        $delete_link = 'http://www.labinventory.eu/inventory_task.php?inventory_task=delete';
11
12        $id = $num_of_columns;
13        $header_size = (empty($args)) ? 0 : intval($args);
14        if (($id < 1) || ($id > $num_of_columns)) return;
15        $fields = array('classification', 'classification_comment', 'part_number', 'part_number_comment',
16        'serial_number', 'serial_number_comment', 'version', 'version_comment', 'inventory_number',
17        'inventory_number_comment', 'manufacturer', 'manufacturer_comment', 'model', 'model_comment', 'type',
18        'type_comment', 'framework_contract', 'framework_contract_comment', 'RTO', 'RTO_comment', 'price', 'price_comment',
19        'description');
20        $delete_confirmation_msg = 'Are you sure you want to delete the selected record?';
21        for ($i=1; $i <= $num_of_rows; $i++) {
22            if ($i > $header_size) {
23                $record_id = $rows[$i][$id];
24                $fields_str = '';
25                foreach ($fields as $field_id => $field_name) {
26                    if (isset($rows[$i][$field_id+1])) {
27                        $fields_str .= '&'.$field_name.'='.urlencode($rows[$i][$field_id+1]);
28                    }
29                }
30            }
31        }

```

```

26 // $userId = $user->id;
27 // jimport( 'joomla.access.access' );
28 // $groups = JFactory::getGroupsByUser($userId);
29
30 $user = JFactory::getUser();
31 if ($user->guest) {
32     // the user is not logged in
33 } else {
34     // user is logged in
35     $fullname = $user->name;
36     $email = $user->email;
37     $username = $user->username;
38     $user_id = $user->id;
39     $group_ids = $user->get('groups');
40     // print_r($group_ids);
41
42
43 }
44
45     if (in_array(11,$group_ids)){
46         $rows[$i][$id] = '<a href="'. $edit_link.'?record_id='.$record_id.$fields_str.'">Edit</a> | <a
href="'. $delete_link.'&record_id='.$record_id.'" onclick="return confirm('\'. $delete_confirmation_msg.'\')">
Delete</a>';
47     } else {
48         $rows[$i][$id] = 'N/A';
49     }
50 }
51 }
52 }
53 ?>

```

Εικόνα 17: Κώδικας για την επεξεργασία και διαγραφή στοιχείων

Όπως φαίνεται στην εικόνα 16 προσθέσαμε μια νέα στήλη στον πίνακα δεδομένων με όνομα action, με την δυνατότητα διαγραφής και επεξεργασίας δεδομένων. Η προσθήκη της στήλης action έγινε στο ρηρ αρχείο `id2links {joomla root directory}/templates/tabulizer/modify/id2links.php` με την προσθήκη δύο links για την διαγραφή ή επεξεργασία δεδομένων. Με το που ο χρήστης επιλέξει μια απο τις δύο επιλογές καλείται το αρχείο `inventorytask.php`. Το συγκεκριμένο αρχείο δεν προϋπήρχε στο component Tabulizer και το υλοποιήσαμε προκειμένου να ρυθμίσουμε τις λειτουργίες εγγραφής, διαγραφής και επεξεργασίας δεδομένων με την βοήθεια των φορμών εισαγωγής, επεξεργασίας και του πίνακα απογραφής.

Στην εικόνα 18 που ακολουθεί γίνεται σαφής διαχωρισμός των τριών περιπτώσεων . Αρχικά μελετάται η περίπτωση ο χρήστης επιλέξει την να διαγραφή όπου με ένα απλό ερώτημα sql αναζητούμε την συγκεκριμένη εγγραφή στην βάση δεδομένων με κριτήριο το `record_id` που αποτελεί πρωτεύον κλειδί , διαγράφουμε την εγγραφή και ενημερώνουμε τον χρήστη για την με κατάλληλο μήνυμα επιβεβαίωσης ή λάθους.

Στην συνέχεια υλοποιήσαμε την λειτουργία εισαγωγής δεδομένων στην βάση δεδομένων. Όπως αναφέρθηκε παραπάνω ο χρήστης μέσω της φόρμας εισαγωγής δεδομένων που υλοποιήσαμε σε ρηρ εισάγει τα δεδομένα του . Οι προσωρινές μεταβλητές περνάνε από την φόρμα εισαγωγής στο αρχείο `id2links` όπου γίνεται η αντιστοιχία μεταβλητών με πίνακα στην βάση δεδομένων και τελικά εισάγονται τα δεδομένα στην βάση.

Τέλος για την λειτουργία τροποποίησης δεδομένων , με το που ο χρήστης επιλέξει να τροποποιήσει μια ήδη υπάρχουσα εγγραφή , εγγραφή αναζητείται στην βάση δεδομένων βάσει του πρωτεύοντος κλειδιού και αφού βρεθεί ο χρήστης μεταφέρεται σε μια φόρμα ίδια με την φόρμα εγγραφής με την διαφορά ότι είναι ήδη συμπληρωμένα τα πεδία. Ο χρήστης κάνει τις αλλαγές που επιθυμεί και στην συνέχεια οι νέες προσωρινές μεταβλητές μεταφέρονται από την φόρμα εισαγωγής στο αρχείο `inventorytask.php` γίνεται η αντιστοίχιση και τέλος τα δεδομένα αποθηκεύονται εκ νέου στην βάση δεδομένων.

```

356 // Construct query
357 switch ($task) {
358
359     case 'delete':
360
361         $sql = 'DELETE FROM product WHERE id = '.$record_id;
362         $success_msg = "Record deleted successfully!";
363         break;
364
365     case 'insert':
366
367         $sql = "INSERT INTO product(classification, classification_comment, part_number,
368 part_number_comment, serial_number, serial_number_comment, administrator, admin_comment, location,
location_comment, version, version_comment, inventory_number, inventory_number_comment, manufacturer,
manufacturer_comment, model, model_comment, type, type_comment, quantity, quantity_comment,
framework_contract, framework_contract_comment, RTO, RTO_comment, price, price_comment, maintenance_cost,
maintenance_cost_comment, renewal_date, renewal_date_comment, description) VALUES('".
mysql_real_escape_string ( $conn, $cla)."',".mysql_real_escape_string ( $conn, $cla_com)."',".
mysql_real_escape_string ( $conn, $par)."',".mysql_real_escape_string ( $conn, $par_com)."',".
mysql_real_escape_string ( $conn, $ser)."',".mysql_real_escape_string ( $conn, $ser_com)."',".
mysql_real_escape_string ( $conn, $adm)."',".mysql_real_escape_string ( $conn, $adm_com)."',".
mysql_real_escape_string ( $conn, $loc)."',".mysql_real_escape_string ( $conn, $loc_com)."',".
mysql_real_escape_string ( $conn, $ver)."',".mysql_real_escape_string ( $conn, $ver_com)."',".
mysql_real_escape_string ( $conn, $inv)."',".mysql_real_escape_string ( $conn, $inv_com)."',".
mysql_real_escape_string ( $conn, $man)."',".mysql_real_escape_string ( $conn, $man_com)."',".
mysql_real_escape_string ( $conn, $mod)."',".mysql_real_escape_string ( $conn, $mod_com)."',".
mysql_real_escape_string ( $conn, $typ)."',".mysql_real_escape_string ( $conn, $typ_com)."',".
mysql_real_escape_string ( $conn, $qua)."',".mysql_real_escape_string ( $conn, $qua_com)."',".
mysql_real_escape_string ( $conn, $fra)."',".mysql_real_escape_string ( $conn, $fra_com)."',".
mysql_real_escape_string ( $conn, $rto)."',".mysql_real_escape_string ( $conn, $rto_com)."',".
mysql_real_escape_string ( $conn, $pri)."',".mysql_real_escape_string ( $conn, $pri_com)."',".
mysql_real_escape_string ( $conn, $cos)."',".mysql_real_escape_string ( $conn, $cos_com)."',".
mysql_real_escape_string ( $conn, $dat)."',".mysql_real_escape_string ( $conn, $dat_com)."',".
mysql_real_escape_string ( $conn, $des)."')";
369         $success_msg = "New record added successfully!";
370         break;
371
372     case 'update':
373
374         $record_id = intval($_POST['record_id']);
375
376         $sql = "UPDATE product SET classification = '".
mysql_real_escape_string ( $conn, $cla)."', classification_comment = '".
mysql_real_escape_string ( $conn, $cla_com)."', part_number = '".
mysql_real_escape_string ( $conn, $par)."', part_number_comment = '".
mysql_real_escape_string ( $conn, $par_com)."', serial_number = '".
mysql_real_escape_string ( $conn, $ser)."', serial_number_comment = '".
mysql_real_escape_string ( $conn, $ser_com)."', administrator = '".
mysql_real_escape_string ( $conn, $adm)."', admin_comment = '".
mysql_real_escape_string ( $conn, $adm_com)."', location = '".
mysql_real_escape_string ( $conn, $loc)."', location_comment = '".
mysql_real_escape_string ( $conn, $loc_com)."', version = '".
mysql_real_escape_string ( $conn, $ver)."', version_comment = '".
mysql_real_escape_string ( $conn, $ver_com)."', inventory_number = '".
mysql_real_escape_string ( $conn, $inv)."', inventory_number_comment = '".
mysql_real_escape_string ( $conn, $inv_com)."', manufacturer = '".
mysql_real_escape_string ( $conn, $man)."', manufacturer_comment = '".
mysql_real_escape_string ( $conn, $man_com)."', model = '".
mysql_real_escape_string ( $conn, $mod)."', model_comment = '".
mysql_real_escape_string ( $conn, $mod_com)."', type = '".
mysql_real_escape_string ( $conn, $typ)."', type_comment = '".
mysql_real_escape_string ( $conn, $typ_com)."', quantity = '".
mysql_real_escape_string ( $conn, $qua)."', quantity_comment = '".
mysql_real_escape_string ( $conn, $qua_com)."', framework_contract = '".
mysql_real_escape_string ( $conn, $fra)."', framework_contract_comment = '".
mysql_real_escape_string ( $conn, $fra_com)."', RTO = '".
mysql_real_escape_string ( $conn, $rto)."', RTO_comment = '".
mysql_real_escape_string ( $conn, $rto_com)."', price = '".
mysql_real_escape_string ( $conn, $pri)."', price_comment = '".
mysql_real_escape_string ( $conn, $pri_com)."', maintenance_cost = '".
mysql_real_escape_string ( $conn, $cos)."', maintenance_cost_comment = '".
mysql_real_escape_string ( $conn, $cos_com)."', renewal_date = '".
mysql_real_escape_string ( $conn, $dat)."', renewal_date_comment = '".
mysql_real_escape_string ( $conn, $dat_com)."', description = '".
mysql_real_escape_string ( $conn, $des)."') WHERE id = ".$record_id;
377         $success_msg = "Record updated successfully!";
378         break;
379

```

Εικόνα 18: Περιπτώσεις διαγραφής, εγγραφής και επεξεργασίας δεδομένων στο αρχείο inventorytask.php

Κεφάλαιο 4

Ασφάλεια συστήματος

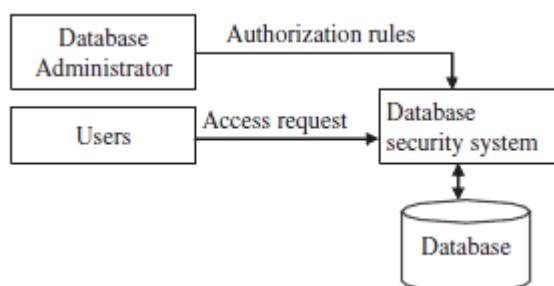
Η κοινωνία μας στηρίζεται όλο και περισσότερο στα ηλεκτρονικά δίκτυα και τα πληροφοριακά συστήματα. Ταυτόχρονα, έχουν αυξηθεί τα κρούσματα ηλεκτρονικής απάτης και πλαστογραφίας. Η κλοπή ταυτότητας, οι πλαστές τραπεζικές ιστοσελίδες και η βιομηχανική κατασκοπεία, αποτελούν μόνο μερικά από τα εγκλήματα που διαπράττονται στο διαδίκτυο και έχουν στόχο προσωπικούς λογαριασμούς, επιχειρήσεις και δημόσιες αρχές.

Σύμφωνα με έρευνα του Ευρωβαρόμετρου για το 2015, οι ευρωπαίοι πολίτες, σε μεγάλο ποσοστό, εκφράζουν ανησυχίες για την ασφάλεια στο διαδίκτυο. Το 89% του συνόλου των χρηστών του διαδικτύου αποφεύγουν την καταγραφή προσωπικών πληροφοριών στο διαδίκτυο, ενώ το 85% συμφωνεί ότι ο κίνδυνος να πέσει θύμα επίθεσης στο διαδίκτυο, αυξάνεται.

Με τις νέες τεχνολογίες να κυριαρχούν τα τελευταία χρόνια καθώς και τη συνεχή άνοδο του Internet of Things (το Διαδίκτυο των Πραγμάτων), η αξία της πληροφορίας αυξάνεται διαρκώς και ταυτόχρονα η ζήτηση και η απόκτησή της. Σημείο ενδιαφέροντος αποτελεί το γεγονός ότι οι προσωπικές προτιμήσεις και οι συνήθειες του καθενός μπορούν να γίνουν ευκαιρία κέρδους. Με τα ατομικά δεδομένα να διακινούνται και να αποθηκεύονται σε πολλαπλές συσκευές συνδεδεμένες μεταξύ τους, όπως κινητές συσκευές, wearables (υπολογιστές - αξεσουάρ) και sensors (αισθητήρες), η πληροφορία γίνεται ευάλωτη σε πιθανές υποκλοπές και παραβιάσεις.

Διαπιστώνουμε λοιπόν ότι είναι επιτακτική η ανάγκη δημιουργίας Συστημάτων Βάσεων Δεδομένων τα οποία όχι μόνο θα αποθηκεύουν, αλλά θα διασφαλίζουν ακόμα και την απόκρυψη πληροφορίας από μη εξουσιοδοτημένους χρήστες. Αναφερόμαστε λοιπόν σε θέματα ασφάλειας. Για την προστασία των δεδομένων σε ένα τέτοιο σύστημα, είναι απαραίτητοι συγκεκριμένοι μηχανισμοί ελέγχου πρόσβασης σε αυτά. Με τον τρόπο αυτό, εξασφαλίζεται η εμπιστευτικότητα (confidentiality) των δεδομένων, ενώ παράλληλα είναι η απαραίτητη η διαρκής διαθεσιμότητά τους στους εξουσιοδοτημένους χρήστες οι οποίοι έχουν το δικαίωμα προσπέλασης σε αυτά.

Κατά την διάρκεια υλοποίησης του συστήματος διαχείρισης δεδομένων, έπρεπε να ληφθούν υπόψη μέτρα για την ασφάλεια των δεδομένων. Τα δεδομένα που αποθηκεύονται στην βάση είναι δεδομένα ενός συγκεκριμένου εργαστηρίου, οπότε η συγκεκριμένη πληροφορία αφορά μόνο το διοικητικό προσωπικό του εργαστηρίου.



Εικόνα 19: Δομή ασφαλούς συστήματος διαχείρισης δεδομένων

Σε γενικές γραμμές, η έννοια της ασφάλειας μίας Βάσης Δεδομένων σε ένα Πληροφοριακό Σύστημα περιλαμβάνει την προστασία από τρεις βασικές απειλές

- *Μη εξουσιοδοτημένη παρακολούθηση δεδομένων (unauthorized data observation)* : Αναφέρεται σε δυνατότητες πρόσβασης σε πληροφορία από χρήστες οι οποίοι δεν έχουν το αντίστοιχο δικαίωμα.
- *Εσφαλμένη τροποποίηση δεδομένων (incorrect data modification)* : Αναφέρεται σε περιπτώσεις που μία μεταβολή στα δεδομένα οδηγεί τη Βάση Δεδομένων σε μία μη έγκυρη κατάσταση, όσον αφορά την πληροφορία που αυτή περιέχει. Λανθασμένα δεδομένα σε έναν οργανισμό μπορεί να οδηγήσουν δε μεγάλες απώλειες.
- *Μη διαθεσιμότητα δεδομένων (data unavailability)* : Όταν τα δεδομένα είναι μη διαθέσιμα, η πρόσβαση στην επιθυμητή πληροφορία τη χρονική στιγμή που κρίνεται αναγκαίο είναι αδύνατη.

Για την αποφυγή τέτοιων καταστάσεων και προστασία των δεδομένων, ο σχεδιασμός της ασφάλειας για ένα σύστημα πρέπει να περιλαμβάνει αντιστοίχως συγκεκριμένες απαιτήσεις όπως:

- *Εμπιστευτικότητα (confidentiality)* : Αναφέρεται στην προστασία των δεδομένων μίας Βάσης από την προβολή τους σε μη εξουσιοδοτημένα άτομα.
- *Ακεραιότητα (integrity)* : Αναφέρεται στην παρεμπόδιση της μη εξουσιοδοτημένης μεταβολής δεδομένων και στη διατήρηση της ακεραιότητάς τους.
- *Διαθεσιμότητα (availability)* : Αναφέρεται στην ύπαρξη μηχανισμών παρεμπόδισης και επαναφοράς του συστήματος από σφάλματα, όσο και από σκόπιμες επιθέσεις που έχουν στόχο τη μη διαθεσιμότητα του συστήματος.

Τα μέτρα που λήφθηκαν για την ασφάλεια της διαδικτυακής εφαρμογής είναι τα εξής:

1. Backup! Πριν κάνουμε οποιαδήποτε άλλη ενέργεια αναβάθμισης , ασφάλειας κλπ. θα πήραμε backup την ιστοσελίδα μας. Ακόμα και να χτυπήσουν την σελίδα η επαναφορά από ένα προηγούμενο ασφαλές αντίγραφο είναι πολύ σημαντική . Η καταλληλότερη εφαρμογή για αυτό το λόγο είναι το akeeba backup , το οποίο και χρησιμοποίησαμε.

2. Χρησιμοποιήσαμε την τελευταία έκδοση του CMS Joomla. Πολλές από τις αναβαθμίσεις είναι για λόγους ασφαλείας.

3. Αναβάθμιση των extension και κυρίως του editor. Οι περισσότερες επιθέσεις που έχουμε δει ξεκινάνε με τρύπες στον editor...

4. Μετονομασία του htaccess.txt σε .htaccess. Πέρα του mod rewrite για sef url υπάρχει μέσα και κώδικας που ασφαρίζει την σελίδα.

5. Κλείδωμα του /administrator με έξτρα κωδικό με χρήση htaccess. Στο cpanel αυτό γίνεται πολύ εύκολα

6. Αλλαγή του url του /administrator . Υπάρχουν διάφορα extension που το κάνουν αυτό. JLSecure My Site , Admin tools pro κλπ

7. Δύσκολος κωδικός χρήστη και σύνδεσης με την βάση δεδομένων

8. Κλείσαμε το ftp layer του joomla.

9. Αποφύγαμε να χρησιμοποιήσουμε Super user Με όνομα χρήστη admin , administrator

10. Υλοποιήσαμε ένα αρχείο php.ini με το παρακάτω περιεχόμενο

```
register_globals=Off  
safe_mode=Off
```

```
allow_url_fopen=Off
allow_url_include=Off
disable_functions=show_source, system, shell_exec, passthru, exec, phpinfo, popen, proc_open
open_basedir=/home/username/public_html:/home/username/tmp
```

11. Αλλάξαμε το Id του admin για να μην χρησιμοποιήσουμε το default (42 ή 62)

12. Αλλάξαμε το επίθεμα της βάσης δεδομένων για να μην χρησιμοποιήσουμε το default (jos_)

13 Αλλάξαμε τα δικαιώματα του configuration.php σε 400 (για να αποθηκεύσετε αλλαγές θα πρέπει να το αλλάζετε σε 644)

Σε μία Βάση Δεδομένων είναι ιδιαίτερα σημαντική η παρακολούθηση όλων των ενεργειών που κάνουν οι χρήστες στο σύστημα, όπως για παράδειγμα κάθε εισαγωγή/τροποποίηση/διαγραφή, ακόμα και ανάγνωση δεδομένων. Για την καταγραφή κάθε ενημέρωσης δεδομένων καθώς και του χρήστη ο οποίος την έκανε, χρησιμοποιείται το *ημερολόγιο* συστήματος. Αυτό περιλαμβάνει μία καταχώρηση για κάθε πράξη/ενέργεια που εφαρμόζεται στη Βάση Δεδομένων, με κύριο σκοπό τη δυνατότητα ανάκαμψης από αποτυχιές δοσοληψιών ή κατάρρευση συστήματος. Στα πλαίσια της ασφάλειας του συστήματος από πιθανές ενέργειες χρηστών, υπάρχει η δυνατότητα επέκτασης του ημερολογίου έτσι ώστε να καταγράφεται τόσο ο χρήστης ο οποίος έκανε κάθε ενέργεια, όσο και η ημερομηνία και ώρα που αυτή πραγματοποιήθηκε. Ένα τέτοιο ημερολόγιο ονομάζεται συνήθως *αρχείο επιθεώρησης*.

Κεφάλαιο 5

Αξιολόγηση συστήματος

Όσον αφορά την αξιολόγηση, οι στόχοι των ατόμων που εμπλέκονται στην ιστοσελίδα διαφέρουν και καθιστούν έτσι διαφορετικούς ρόλους ανάμεσα σε σχεδιαστές, managers, νόμιμους κάτοχους ιστοσελίδας και χρήστες. Όλα αυτά αναφέρονται λαμβάνοντας υπόψη ένα απλοποιημένο μοντέλο στο οποίο η αξιολόγηση μπορεί να βασιστεί σ' έναν τρισδιάστατο χώρο. Με άλλα λόγια, αυτό που πρέπει να αξιολογηθεί είναι η ικανοποίηση του χρήστη από την αποτελεσματικότητα σε σχέση με την κατηγορία της ιστοσελίδας και τους στόχους του σχεδιαστή της, παρατηρώντας ότι αυτές οι διαστάσεις είναι αλληλεξαρτώμενες. Στην πραγματικότητα, το αίσθημα της ικανοποίησης των χρηστών εξαρτάται, επίσης, από το πόσο μια ιστοσελίδα είναι συμβατή με την αντίληψη που αυτοί έχουν για την κατηγορία στην οποία ανήκει. Η επίτευξη των στόχων του σχεδιαστή της εξαρτάται από την ικανοποίηση του χρήστη, επειδή όσο περισσότερο είναι αυτός ικανοποιημένος, τόσο περισσότερο είναι πιθανό να επισκεφθεί την ιστοσελίδα κατά τον τρόπο που αρχικά ο σχεδιαστής της είχε προγραμματίσει και σχεδιάσει.

Η συμφωνία μιας ιστοσελίδας με μια δεδομένη κατηγορία δικτυακών τόπων, επίσης, εξαρτάται από τον τρόπο με τον οποίο ο σχεδιαστής έχει σχεδιάσει τους στόχους του με βάση τα χαρακτηριστικά αυτής της κατηγορίας, αν και ενδέχεται να σχετίζεται έμμεσα με το αίσθημα ικανοποίησης του χρήστη. Η αξιολόγηση των χαρακτηριστικών μιας ιστοσελίδας μπορεί να αναπαρασταθεί με ένα σημείο στον τρισδιάστατο αυτό χώρο. Ωστόσο, με βάση ό,τι έχει ήδη αναφερθεί, η ικανοποίηση του χρήστη επηρεάζει τόσο έντονα τις άλλες δύο παραμέτρους ώστε μπορεί να θεωρηθεί ως ένα γενικό μέτρο αξιολόγησης της επιτυχίας μιας ιστοσελίδας



Εικόνα 20: Τρεις άξονες για την αξιολόγηση συστήματος

Δύο είναι τα βασικά ζητήματα που αναδύονται από την παρούσα ανάλυση σχετικά με την αξιολόγηση μιας ιστοσελίδας. Πρώτον, οι μέθοδοι που χρησιμοποιούνται έχουν να κάνουν κυρίως με την προσβασιμότητα, και τη συμβατότητα, και επίσης με τα συντακτικά και δομικά χαρακτηριστικά. Οι εργασίες που εκτελεί ένας χρήστης μιας ιστοσελίδας μπορεί να είναι σύνθετες. Επιπλέον δεν είναι στατικές, αλλά μπορούν να εξελιχθούν δυναμικά κάτω από την επίδραση των μεταβολών των αναγκών των χρηστών. Το στοιχείο αυτό οδηγεί σε ένα δεύτερο ζήτημα. Η αξιολόγηση μιας ιστοσελίδας έχει δύο ιδιάζοντα χαρακτηριστικά.

Ακόμη κι αν η κατηγορία της ιστοσελίδας είναι προσδιορισμένη με σαφήνεια, είναι δύσκολο να διατυπωθούν με ακρίβεια όλα τα χαρακτηριστικά που αυτή έχει, δηλαδή όλες οι διάφορες εργασίες που οι χρήστες αναμένουν να επιτελέσουν κάνοντας χρήση της συγκεκριμένης ιστοσελίδας. Διαφορετικοί χρήστες μπορεί να έχουν ανάγκη διαφορετικής πληροφόρησης και διαφορετικών τρόπων πλοήγησης, καθώς και διαφορετικών λογικών δομών. Επιπλέον, είναι δύσκολο να διασαφηνιστεί εκ των προτέρων ποιος είναι ακριβώς ο χρήστης όσον αφορά το πολιτισμικό του επίπεδο, την εμπειρία του, τις προτεραιότητές του, καθώς και τι είδος σύνδεσης με το internet έχει. Συνήθως, όταν μια ιστοσελίδα είναι καλά σχεδιασμένη δεν λαμβάνονται υπόψη τα κίνητρα μιας θετικής αξιολόγησης. Αντίστοιχα, όταν μια ιστοσελίδα δεν ικανοποιεί τους χρήστες της είναι καίριας σημασίας να κατανοήσουμε τα πραγματικά προβλήματα που υφίστανται.

Τα ερωτηματολόγια είναι μια από τις έμμεσες τεχνικές δοκιμασίες σχεδιασμένες για την αξιολόγηση του βαθμού χρηστικότητας. Μπορούν να προσφέρουν πολύτιμες πληροφορίες αν και θα πρέπει να ικανοποιούν συγκεκριμένες προδιαγραφές. Ως ένα γενικό κανόνα, τα ερωτήματα θα πρέπει να είναι καλά διατυπωμένα, δηλαδή σαφή και σημαντικά ως προς το πλαίσιο αξιολόγησης. Επιπλέον, τα αποτελέσματα θα πρέπει να αναλυθούν και να ερμηνευθούν προσεκτικά. Ένα καλά σχεδιασμένο ερωτηματολόγιο μπορεί να δώσει μια ικανοποιητική παρουσίαση των προβλημάτων της εφαρμογής του, ακόμα κι αν είναι δύσκολο να αποκτηθεί πρόσβαση σε πολύ λεπτομερειακές πληροφορίες. Ο βαθμός εφικτότητας και η ακρίβεια της ανάλυσης, επίσης, εξαρτάται από τον τρόπο με τον οποίο παρουσιάζονται οι ερωτήσεις. Τα αποτελέσματα που βασίζονται σε κατάλληλα σχεδιασμένες κλίμακες Likert μπορούν να διαχειριστούν και να συνοψισθούν σε ικανοποιητικό βαθμό ενώ οι απαντήσεις στα ανοιχτά ερωτήματα μπορούν εν γένει να είναι σημαντικά, αν και είναι δύσκολο να αναλυθούν από ποιοτικής πλευράς.

Για τις ανάγκες αξιολόγησης του συγκεκριμένου συστήματος διαχείρισης βάσεων δεδομένων δημιουργήσαμε ένα ερωτηματολόγιο το οποίο στάλθηκε σε εργαζόμενους εργαστηρίων. Τα άτομα, σύνολο 18 ατόμων, που απάντησαν στις ερωτήσεις είναι εργαζόμενοι σε εργαστήρια διάφορων κατηγοριών και διάφορων ειδικοτήτων. Το ενδιαφέρον είναι ότι κάποιοι από αυτούς έχουν μια διοικητική θέση και εμπειρία σε ήδη υπάρχοντα συστήματα διαχείρισης εξοπλισμού, ενώ άλλοι ασχολούνται με ερευνητικές ή πειραματικές εργασίες του εργαστηρίου.

Στην συγκεκριμένη ενότητα θα προσπαθήσουμε να ξεκαθαρίσουμε το προφίλ των ανθρώπων που αξιολόγησαν το συγκεκριμένο σύστημα. Να διευκρινιστεί ότι εφόσον το σύστημα διαχείρισης δεδομένων για τις ανάγκες απογραφής εργαστηριακού εξοπλισμού αφορά συγκεκριμένα πλήθος ατόμων, που ασχολείται και εργάζεται σε εργαστήριο, η διαδικασία της αξιολόγησης δεν θα μπορούσε να είναι ανοιχτή αλλά θα πρέπει να στοχεύει σε συγκεκριμένο προφίλ χρηστών, οι οποίοι θα μπορούσαν να είναι μελλοντικοί χρήστες του συγκεκριμένου συστήματος ή έχουν ήδη εμπειρία σε παρόμοια συστήματα.

Αναλυτικότερα το πλήθος των ατόμων που προηγήθηκαν στο σύστημα και στη συνέχεια απάντησαν στο ερωτηματολόγιο είναι 18 και συγκεκριμένα :

- 3 προγραμματιστές εφαρμογών διαδικτύου. Το συγκεκριμένο σύνολο ατόμων το χρειαστήκαμε προκειμένου να αξιολογήσουν την εφαρμογή από μία άλλη οπτική γωνία και όχι αυτήν της λειτουργικότητας για την απογραφή δεδομένων αλλά για την αξιολόγηση όσον αφορά την ευχρηστία ,την ταχύτητα κλπ. Το συγκεκριμένο δείγμα θα μπορούσε να αποτελεί την κατηγορία διαχειριστών συστήματος.
- 15 εργαζομένων σε ερευνητικά εργαστήρια από τους οποίους
 - ο 4 είναι αποκλειστικά διοικητικοί υπάλληλοι. Το συγκεκριμένο δείγμα θα μπορούσε να αποτελεί την κατηγορία χρηστών dataprovider αλλά και το dataviewers.
 - ο 5 είναι ερευνητές με καμία διοικητικοί αρμοδιότητα , εργάζονται στο εργαστήριο για διδακτορικό, μεταδιδακτορικό και χρησιμοποιούν το εργαστήριο για ερευνητικούς λόγους. Το συγκεκριμένο δείγμα θα μπορούσε να αποτελεί την κατηγορία χρηστών dataviewers
 - ο 6 είναι ερευνητές με διοικητικές αρμοδιότητες οι περισσότεροι από αυτούς και εμπειρία στην απογραφή εργαστηριακού εξοπλισμού. Το συγκεκριμένο δείγμα θα μπορούσε να αποτελεί την κατηγορία χρηστών dataprovider

Όλοι από τους παραπάνω εργάζονται στο ερευνητικό κέντρο JRC της ευρωπαϊκής επιτροπής και συγκεκριμένα σε δύο ερευνητικά εργαστήρια (Eurl-Ecvam - European Union Reference Laboratory for alternatives to animal testing και SGI Lab - Smart Grid and Interoperability Lab)

Η δομή των ερωτήσεων πρέπει να είναι απλή και η έκταση των ερωτήσεων μικρή. Ο χρήστης θα πρέπει να έχει ολοκληρώσει το ερωτηματολόγιο σε περίπου 3 λεπτά ώστε να το θεωρήσει μια ευχάριστη και όχι κουραστική εμπειρία. Έχει παρατηρηθεί ότι σε περιπτώσεις μακροσκελών ερωτηματολογίων ο χρήστης κουράζεται και εκνευρίζεται και αυτός ο εκνευρισμός είναι εμφανής και στον τρόπο που απαντάει. Από την άλλη μεριά η διατύπωση των ερωτήσεων θα πρέπει να είναι κατανοητή και απλή. Από την πρώτη ανάγνωση ο χρήστης θα πρέπει να έχει καταλάβει ακριβώς σε τι καλείται να απαντήσει. Το εργαλείο που χρησιμοποιήθηκε για την υλοποίηση του ερωτηματολογίου είναι ένα εργαλείο ανοιχτού λογαριασμού, το EU-Survey. Η επιλογή του συγκεκριμένου ερωτηματολογίου βασίστηκε όχι μόνο στην ευκολία ανάλυσης των δεδομένων αλλά και στην ευκολία κατανόησης από την μεριά των χρηστών.

Επιπλέον ,η δομή του ερωτηματολογίου δεν είναι τόσο απλή καθώς διαθέτει ερωτήσεις διαφορετικού τύπου , όπως ερωτήσεις πολλαπλής επιλογής , ερωτήσεις μίας επιλογής και ερωτήσεις κλειμένου. Το ερωτηματολόγιο περιέχει και σύνθετες ερωτήσεις , κάποιες ερωτήσεις εξαρτώνται και εμφανίζονται ανάλογα με τις απαντήσεις των χρηστών σε παραπάνω ερωτήσεις.

Η υλοποίηση του ερωτηματολογίου ήταν εύκολη με την βοήθεια του εργαλείου EU-Survey.

Ο σύνδεσμος για το ερωτηματολόγιο είναι

<https://ec.europa.eu/eusurvey/runner/2aa137fd-ef8a-a6f0-4fbf-2aad1ee333cc>

Κωδικός εισόδου : SOMOUS

Lab Inventory - Evaluation

Τα πεδία με αστερίσκο (*) είναι υποχρεωτικά.

*To which category do you belong?

- Administrator
 Dataprovider
 Dataviewer

*Do you have already experience in laboratory inventory?

- Yes
 No

*How many years?

*How did you visit the online lab inventory?

- Mobile phone
 Desktop/Laptop
 Tablet/Ipad

*Are you a programmer/ web developer ?

- Yes
 No

*Do you think that it is easy to modify this database management system for other purposes?

- Yes
 No

*Did you find easily the information that you were searching?

- Yes
 No

*How did you feel about the web design?

- Satisfied
 Dissatisfied

please explain the reasons

*Did you use the contact form in order to communicate with the system administrator?

- Yes
 No

Εικόνα 21: Ερωτηματολόγιο αξιολόγησης

Αποτελέσματα ερωτηματολογίου χρηστών

To which category do you belong?

	Answers	Ratio
Administrator	6	33.33 %
Dataprovider	6	33.33 %
Dataviewer	6	33.33 %
No Answer	0	0.00 %

Do you have already experience in laboratory inventory?

	Answers	Ratio
Yes	12	66.67 %
No	6	33.33 %
No Answer	0	0.00 %

How did you visit the online lab inventory?

	Answers	Ratio
Mobile phone	9	50.00 %
Desktop/Laptop	15	83.33 %
Tablet/Ipad	9	50.00 %
No Answer	0	0.00 %

Are you a programmer/ wed developer ?

	Answers	Ratio
No	15	83.33 %
Yes	3	16.67 %
No Answer	0	0.00 %

Do you think that it is easy to modify this database management system for other purposes?

	Answers	Ratio
Yes	9	50.00 %
No	6	33.33 %
No Answer	3	16.67 %

Did you find easily the information that you were searching?

	Answers	Ratio
Yes	18	100.00 %
No	0	0.00 %
No Answer	0	0.00 %

How did you feel about the web design?

	Answers	Ratio
Satisfied	18	100.00 %
Disatisfied	0	0.00 %
No Answer	0	0.00 %

Did you use the contact form in order to communicate with the system administrator?

	Answers	Ratio
Yes	9	50.00 %
No	9	50.00 %
No Answer	0	0.00 %

Συμπεράσματα

Γενικά λοιπόν, όπως παρουσιάστηκε στα προηγούμενα κεφάλαια, η εφαρμογή, έχοντας χαρακτηριστικά που την καθιστούν σύστημα διαχείρισης δεδομένων, μπορεί να ενσωματωθεί στα εργαλεία ενός εργαστηρίου και να αποτελέσει μια ικανοποιητική λύση για τη διαχείριση δεδομένων, η οποία επιτρέπει στους χρήστες του να διαχειρίζονται τα δεδομένα τους με τρόπο αποδοτικό σε οργάνωση, χρόνο και χρήμα.

Η ιδέα του συγκεκριμένου θέματος προέκυψε κατά την απασχόληση μου σε δύο ερευνητικά εργαστήρια. Σκοπός της υλοποίησης ενός συγκεκριμένου συστήματος για την απογραφή είναι η διευκόλυνση για την λειτουργία του εργαστηρίου. Μέσα από την εμπειρία μου διαπίστωσα ότι όλοι οι εμπλεκόμενοι χρήστες έχουν συνειδητοποιήσει την σημασία της απογραφής εξοπλισμού, σε κάποιες περιπτώσεις μάλιστα έχουν προχωρήσει στην απογραφή του εργαστηριακού εξοπλισμού με διάφορες μορφές όπως απογραφή με την μορφή αρχείου. Ελάχιστοι όμως έχουν προχωρήσει στην χρήση ενός συστήματος για την απογραφή εξοπλισμού, χρησιμοποιώντας ένα ήδη υπάρχον σύστημα διαχείρισης δεδομένων και τέλος κανένας δεν έχει προχωρήσει στην υλοποίηση ενός εξειδικευμένου συστήματος για την απογραφή εξοπλισμού. Ο λόγος προφανώς είναι το κόστος, ωστόσο το συμπέρασμα όλων είναι ότι μια τέτοια εφαρμογή μακροπρόθεσμα θα βοηθούσε σημαντικά στην διοίκηση και στο κόστος.

Οι επεκτάσεις και οι βελτιώσεις που μπορούν να γίνουν στο σύστημα, έτσι ώστε αυτό να γίνει ολοκληρωμένο είναι πάρα πολλές και απαιτούν για την υλοποίησή τους αρκετές ώρες εργασίας και προσπάθειας. Ωστόσο, αναφέρονται στη συνέχεια μερικές από αυτές που τις θεωρούμε πιο σημαντικές. Το πιο σημαντικό αφορά το θέμα της επεκτασιμότητας και προσθήκης νέων στοιχείων. Στην εφαρμογή μας τα δεδομένα αποθηκεύονται σε μια ενιαία βάση δεδομένων. Εκεί συγκεντρώνονται όλα τα δεδομένα των χρηστών. Ένας διαφορετικός, περισσότερο αποδοτικός τρόπος θα ήταν η αρχειοθέτηση σε ομαδοποιημένους φακέλους ανάλογα με τους χρήστες.

Μία άλλη πιθανή επέκταση του συστήματος θα μπορούσε να είναι η δυνατότητα κάποιων απλών ερωτήσεων – queries από την μεριά των χρηστών. Σε αυτήν την φάση του συστήματος ο χρήστης μπορεί να εισάγει, να διαγράψει, να τροποποιήσει και να δει τα δεδομένα που έχουν καταχωρηθεί. Για να μελετήσει όμως τα δεδομένα ο χρήστης στη συγκεκριμένη φάση καλείται να εξάγει τον αντίστοιχο πίνακα που τον αφορά σε ένα αρχείο excel και να το επεξεργαστεί ή να ρωτήσει τον διαχειριστή συστήματος, μέσω της φόρμας επικοινωνίας, που έχει πρόσβαση στην βάση δεδομένων και γνώση της sql το ερώτημα που τον ενδιαφέρει και αυτός με την σειρά του να του επιστρέψει τα αποτελέσματα.

Για παράδειγμα δύο σημαντικά ερωτήματα – queries που θα μπορούσαν να ενδιαφέρουν τους μελλοντικούς χρήστες θα μπορούσαν να είναι τα εξής.

- Στο συγκεκριμένο κτίριο τι εξοπλισμός βρίσκεται?
- Ποιες ανανεώσεις εξοπλισμού που ανήκουν σε εμένα λήγουν τον επόμενο εξάμηνο και πόσο είναι το συνολικό κόστος ανανέωσης ?

Τα ερωτήματα αυτά είναι απλά και μπορούν να απαντηθούν εύκολα με μια ερώτηση στην βάση δεδομένων ωστόσο οι χρήστες από την μια μεριά δεν έχουν πρόσβαση στην βάση δεδομένων για λόγους ασφαλείας από την άλλη δεν έχουν γνώση της mysql. Μια πιθανή λύση θα μπορούσε να είναι η δημιουργία ενός νέου menu item, άρθρου στο cms Joomla με το όνομα Report σε αυτό θα είναι ήδη διατυπωμένα κάποια βασικά ερωτήματα queries , ο χρήστης ανάλογα με το ερώτημα θα καθορίζει την ημερομηνία , το όνομα κατόχου ή την τοποθεσία που τον ενδιαφέρει και θα εκτελεί το ερώτημα. Θα του επιστρέφεται ένας πίνακας με τα αποτελέσματα ή ένα μήνυμα λάθους.

Μια επιπλέον έλλειψη που θα μπορούσε να αποτελέσει μελλοντική επέκταση του τωρινού συστήματος είναι η δυνατότητα εισαγωγής δεδομένων μέσω ενός excel αρχείου. Στην προκειμένη φάση ο χρήστης μπορεί να εισάγει δεδομένα στην βάση μέσω της φόρμας εισαγωγής δεδομένων και του δίνεται η δυνατότητα να τα εξάγει και να τα αποθηκεύσει με την μορφή αρχείου excel. Ωστόσο θα ήταν αρκετά χρήσιμο ειδικά στο αρχικό στάδιο κατά το οποίο οι χρήστες καλούνται να εισάγουν πολλά δεδομένα και όχι τόσο κατά την διάρκεια συντήρησης και διαχείρισης.

Για παράδειγμα υπάρχουν πολλές εγγραφές που έχουν ακριβώς τα ίδια χαρακτηριστικά και διαφέρουν μόνο στο χαρακτηριστικό serial number, για παράδειγμα τα καλώδια. Σε αυτήν την περίπτωση η δυνατότητα εισαγωγής από excel αρχείο θα βοηθούσε αρκετά σε χρόνο. Από την άλλη μεριά το αρχείο excel θα πρέπει να έχει συγκεκριμένη μορφή και είναι δύσκολο να εγγραφθούμε τι είδος αρχείο θα εισάγει ο χρήστης. Στην συγκεκριμένη φάση μόνο ο διαχειριστής συστήματος μπορεί να εισάγει δεδομένα μέσω ενός αρχείου excel καθώς μόνο αυτός έχει πρόσβαση στο διαχειριστικό της βάσης δεδομένων.

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν μεταξύ άλλων νέες τεχνολογίες όπως HTML, CSS και jQuery. Οι τεχνολογίες αυτές λύνουν προβλήματα που παλαιότερες τεχνολογίες όπως η flash μπορούσαν μεν να λύσουν αλλά με αρκετά πιο δύσκολο τρόπο. Ο συνδυασμός των τριών παραπάνω τεχνολογιών δίνει τη δυνατότητα στον προγραμματιστή πρακτικά να κάνει τα πάντα. Το μεγαλύτερο πλεονέκτημά τους είναι ότι μπορούν να εφαρμοστούν πάνω σε οποιαδήποτε συσκευή, σταθερή ή φορητή, ενώ πλέον οι ενημερωμένες εκδόσεις όλων των φυλλομετρητών τις υποστηρίζουν.

Η λειτουργία της εφαρμογής σε φορητές συσκευές επέκτεινε εξαιρετικά το φάσμα των χρηστών της. Παρεμβάσεις στο σχεδιαστικό κομμάτι της, όσων αφορά δηλαδή τη CSS και το χειρισμό της με jQuery, την κατέστησαν λειτουργική σε φορητές συσκευές και έδωσαν πλήθος νέων δυνατοτήτων στη χρήση της.

Από τη σχεδίαση ακόμη της εφαρμογής βασικός στόχος ήταν να υπάρξει η δυνατότητα προσθήκης νέων στοιχείων. Τόσο η διεπαφή χρήστη της εφαρμογής όσο και η βάση δεδομένων επιτρέπουν την

εύκολη σύνδεση νέων ή υπαρχόντων εργαλείων, τα οποία με αυτόν τον τρόπο θα αποκτούν χαρακτηριστικά ολοκληρωμένης εφαρμογής.

Βιβλιογραφία

- 1] Bielawski, L., Boyle, J., Electronic Document Management Systems: A user centered Approach for Creating, Distributing and Managing On-line publications, Copyright © 1999 Booknews, Inc., Portland.
- [2] City of Spokane, Best Management Practices for Laboratories, October 1995
- [3] Μουζακίτης, Σ., Σχεδιασμός και υλοποίηση συστήματος διαχείρισης εγγράφων Σύστημα Διαχείρισης εγγράφων. Δακτυλογραφημένη διπλωματική εργασία, Ε.Μ.Π., Αθήνα 2004
- [4] King County Local Hazardous Waste Management Program, Waste Management Guidelines for King County Laboratories, September 1994
- [5] Castro, E., Εισαγωγή στην HTML για τον παγκόσμιο ιστό με εικόνες, Copyright © 2000 Κλειδάριθμος.
- [6] Date, C.J., Introduction to Database Systems (7th Edition) Addison Wesley, 2000
- [7] S. Sarawagi, S. Thomas, and R. Agrawal. Integrating mining with relational database systems: Alternatives and implications. In Haas and Tiwary
- [8] J. J. Elam, J. C. Henderson, and L. W. Miller, "Model management systems: An approach to decision support in complex organizations," in Proc. Conf. Inform. Syst., 1980, pp. 98-110.
- [9] C Coronel, S Morris , Database Systems: Design, Implementation, & Management
- [10] Yetgin, E., Jensen, M., & Shaft, T. (2015). Complacency and intentionality in IT use and continuance. AISTransactions on Human-Computer Interaction, 7(1), 17-42.
- [11] G. Ioannou, K. C. Pramataris and G.P. Prastacos, " A Quality Function Deployment Approach to Web Site Development: Applications for Electronic Retailing" Les Cahiers du Management Technologique, vol. 13, no. 3, 2004.
- [12] [Online] www.joomla.org
- [13] [Online] www.w3schools.com
- [14] [Online] www.apache.com
- [14] [Online] www.mysql.com