

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανικών Πληροφορικής

Πτυχιακή εργασία

**Τίτλος: Ανάπτυξη διαδικτυακής εφαρμογής παρακολούθησης
συμβάντων σε υπολογιστικά συστήματα.**

Νικόλαος Βενέτης (ΑΜ:591)

Επιβλέπων καθηγητής: Παρασκευή Φραγκοπούλου

Επιτροπή Αξιολόγησης: Χαράλαμπος Παπαδάκης – Αθανάσιος Μαλάμος

Ημερομηνία παρουσίασης: 27/6/14

Abstract

The increasingly volume of information today causes a variety of issues for businesses concerning the organizations' already highly abstract and complex systems hard to manage. Decision taking in this kind of context might lead to structural errors by simply not fully understanding the function of each subsystem and dependencies between them. A hierarchical structure as far organizational structure is concerned comes to rescue as it allows each subsystem to be managed separately, a structure known as a decision tree. The great complexion of the general organizational structure can be radically reduced. Eventually, each node of the decision tree acts as a part of the whole organizational formation. Thus, it allows the decisions to be taken for each autonomic subsystem.

Σύνοψη

Ο ολοένα αυξανόμενος όγκος της πληροφορίας σήμερα φέρνει στην επιφάνεια ποικιλία προβλημάτων για τους οργανισμούς, τα οποία αφορούν στα αφηρημένα και πολύπλοκα συστήματά τους, τα οποία είναι ήδη δύσκολα στη διαχείρισή τους. Η λήψη αποφάσεων σε αυτό το πλαίσιο είναι ικανή να οδηγήσει σε δομικά και θεμελιώδη λάθη, για το λόγο ότι δεν είναι εύκολη η πλήρης κατανόηση της λειτουργίας κάθε ξεχωριστού υποσυστήματος και των αλληλεξαρτήσεων μεταξύ των. Το ιεραρχικό δένδρο αποφάσεων είναι η μία λύση στο συγκεκριμένο πρόβλημα και διευκολύνει τη διαχείρισή του κάθε υποσυστήματος ξεχωριστά. Η μεγάλη πολυπλοκότητα της γενικής οργανωτικής δομής μπορεί να μειωθεί δραστικά. Τελικά, ο κάθε κόμβος του δένδρου αποφάσεων αποτελεί ως ένα μέρος της ολότητας των συστημάτων του οργανισμού και έτσι οι αποφάσεις λαμβάνονται για κάθε αυτόνομο υποσύστημα.

Πίνακας περιεχομένων

Πτυχιακή εργασία.....	1
Abstract	2
Σύνοψη	3
1 Εισαγωγή.....	7
1.1 Περίληψη.....	7
1.2 Σκοπός και στόχοι της εργασίας.....	8
1.3 Δομή εργασίας.....	8
2 Μεθοδολογία υλοποίησης	9
2.1 Μέθοδος ανάλυσης & ανάπτυξης πτυχιακής	9
2.2 Θεωρία.....	10
3 Κύριο μέρος του προβλήματος.....	16
3.1 Ανάλυση προβλήματος.....	16
3.1.2 Απαιτήσεις συστήματος	16
3.2 Σχεδιασμός υλοποίησης	17
3.3 Υλοποίηση.....	18
3.4 Εισαγωγή στις διαδικτυακές εφαρμογές	18
3.4.1 Το μοντέλο client-server.....	18
3.4.2 Το επίπεδο client	19
3.4.3 HyperText Markup Language (HTML)	19
3.4.4 Cascading Style Sheets (CSS).....	19
3.4.5 Javascript	19
3.4.6 jQuery.....	20
3.4.7 Asynchronous JavaScript and XML (AJAX).....	20
3.4.8 Το επίπεδο server.....	21
3.4.9 Hypertext Processor (PHP)	21
3.4.10 Database	21
3.4.11 Σύστημα διαχείρισης βάσης δεδομένων (DBMS).....	21
3.4.12 Περιηγητές Ιστού (Web browsers).....	22
3.4.13 HTML5	22
3.4.14 MySQL.....	22
3.4.15 Open-source software (OSS).....	23
3.4.16 SQL Injection (Επίθεση SQL).....	23
3.5.1 Η βάση δεδομένων treesystem.....	23
3.5.2 Δομή αρχείων και φακέλων.....	29
3.5.3 Είσοδος στο σύστημα.....	29
3.5.4 Η σελίδα χρήστη (user).....	30
3.5.5 Η σελίδα διαχειριστή (administrator).....	34
4 Εγχειρίδιο χρήσης της εφαρμογής.....	36
6 Βιβλιογραφία.....	48

Πίνακας Εικόνων

Εικόνα 2.1: Διάγραμμα ER στη βάση δεδομένων με τους πίνακες που σχηματίζουν την ιεραρχική δομή10	
graphics1	10
Εικόνα 2.2: Κανονική συστημική δομή.....	12
graphics16	12
Εικόνα 2.3: Ιεραρχική δομή στην απλή μορφή της.....	13
graphics3	13
Εικόνα 2.4: Αφηρημένη ιεραρχική δομή που αναπαριστάται από τη βάση δεδομένων.....	14
graphics2	14
Εικόνα 3.1: Πλήρες διάγραμμα ER στη βάση δεδομένων.....	25
graphics4	25
Εικόνα 3.2: Ο πίνακας computers: περιλαμβάνει τους υπολογιστές που ανήκουν στα Υ/Σ	26
graphics17	26
Εικόνα 3.3: Ο πίνακας departments: περιλαμβάνει τα τμήματα του οργανισμού	26
graphics18	26
Εικόνα 3.4: Ο πίνακας events: περιλαμβάνει τα συμβάντα που καταγράφονται	27
graphics19	27
Εικόνα 3.5: Ο πίνακας halls: περιλαμβάνει τα υπολογιστικά συστήματα.....	27
graphics20	27
Εικόνα 3.6: Ο πίνακας premises: περιλαμβάνει τις εγκαταστάσεις της επιχείρησης	28
graphics21	28
Εικόνα 3.7: Ο πίνακας users: περιλαμβάνει όλους τους χρήστες που χρησιμοποιούν το σύστημα	28
graphics22	28
Εικόνα 3.8: Η αρχική διαδρομή αρχείων και φακέλων της εφαρμογής-ιστοσελίδας.....	29
graphics5	29
Εικόνα 4.1: Η είσοδος στο σύστημα(index.php)	36
graphics6	36
Εικόνα 4.2: Η σελίδα του χρήστη (user.php).....	37
graphics7	37
Εικόνα 4.3: Η αναφορά του συμβάντος που επέλεξε ο χρήστης.....	38
graphics8	38
Εικόνα 4.4: Η διάρκεια των συμβάντων στην προβολή του ημερολογίου ανά εβδομάδα	39
graphics9	39
Εικόνα 4.5: Η φόρμα που συμπληρώνει ο χρήστης για την καταχώρηση του συμβάντος.....	40
graphics10	40
Εικόνα 4.6: Ο πίνακας των συντηρήσεων του Τμήματος 5 για το 2014	41
graphics11.....	41
Εικόνα 4.7: Το διάγραμμα των συντηρήσεων του Τμήματος 5 για το 2014	42
graphics12	42
Εικόνα 4.8: Καταχώρηση Υ/Σ και υπολογιστών στη βάση δεδομένων.....	43
graphics13	43
Εικόνα 4.9: Εισαγωγή νέων χρηστών στη βάση δεδομένων	44
graphics14	44
Εικόνα 4.10: Τα αναλυτικά στατιστικά για το διαχειριστή.....	45
graphics15	45
Εικόνα 4.11: Αποσύνδεση του χρήστη από το σύστημα πατώντας πάνω δεξιά τον υπερσύνδεσμο Αποσύνδεση	46
graphics23	46
Εικόνα 4.12: Μήνυμα επιτυχίας στο διαχειριστή έπειτα από εισαγωγή υπολογιστικού συστήματος....	46
graphics24	46

1 Εισαγωγή

Η παρούσα πτυχιακή εργασία έχει ως αντικείμενο την ανάλυση και υλοποίηση λογισμικού που θα μπορεί να χρησιμοποιηθεί από το τμήμα Πληροφορικής μιας εταιρείας και να αποτελέσει δομικό στοιχείο του πληροφοριακού συστήματός της. Ο οργανισμός θεωρούμε ότι δραστηριοποιείται στη διαχείριση υπολογιστικών συστημάτων (Υ/Σ) και απαιτεί τη συστηματική καταγραφή και στατιστική ανάλυση των συμβάντων που πραγματοποιούνται επί αυτών. Η εργασία θα εστιάσει στη σχεδίαση του λογισμικού δείχνοντας ότι η δενδροειδής οργάνωση των συστημάτων της είναι μία αποτελεσματική λύση ενώπιον της πολύπλοκης διαχείρισής τους.

1.1 Περίληψη

Σκοπός της παρούσας πτυχιακής είναι η μελέτη, σχεδίαση και ανάπτυξη εφαρμογής για την καταγραφή και ανάλυση των συμβάντων (βλαβών, προβλημάτων, εργασιών συντήρησης) που δύνανται να πραγματοποιηθούν σε υπολογιστικά συστήματα και συγκεκριμένα σε εξαρτήματα (hardware, peripherals), τα οποία διαχειρίζεται και συντηρεί μία εταιρεία. Ένα υπολογιστικό σύστημα καλείται σύμφωνα με την πτυχιακή εργασία μία οντότητα που περιλαμβάνει μεμονωμένους υπολογιστές. Το πλήθος των υπολογιστών ποικίλλει ανάλογα με το υπολογιστικό σύστημα. Με άλλα λόγια, είναι ένας φυσικός χώρος εντός του οποίου είναι τοποθετημένοι υπολογιστές.

Μελετήθηκαν υπάρχουσες προτάσεις στη διαχείριση των υπολογιστικών συστημάτων δίνοντας ιδιαίτερη έμφαση στην αρχιτεκτονική με την οποία σχεδιάστηκαν. Η μελέτη ανέδειξε ορισμένες πτυχές της κάθε αρχιτεκτονικής, με τη δενδροειδή δομή να θεωρείται ο καταλληλότερος υποψήφιος εξ' αυτών.

Η αρχιτεκτονική σε τέτοιου είδους εφαρμογές αποτελεί θεμελιώδη λίθο, αφού πρωτεύων στόχος του συστήματος που θα αναπτυχθεί είναι η αποδοτική και χωρίς δυσκολίες διαχείριση του πολύπλοκου δικτύου συστημάτων του οργανισμού. Είναι αναγκαίο να εφευρεθεί ένας τρόπος που θα αυτοματοποιήσει τις λειτουργίες καταγραφής των ενεργειών επί των συστημάτων, ο οποίος θα επιτρέψει τη στατιστική ανάλυση της πληροφορίας που εξάγεται από το πρωταρχικό στάδιο της καταγραφής. Η ανάλυση της πληροφορίας με τη σειρά της θα δώσει μία άμεση και ευπαρουσίαστη εικόνα στο προσωπικό διαχείρισης και θα βοηθήσει αλλά και θα επιταχύνει τη λήψη αποφάσεων όσον αφορά τα συστήματα, όπως για παράδειγμα τη συχνότητα συντήρησής των.

Οι τεχνολογίες που επελέγησαν για τη γραφή της εφαρμογής είναι σύγχρονες και open-source, ενώ χρησιμοποιούνται επί πολλά έτη κυριαρχώντας σε πολλά σημεία έναντι εναλλακτικών τεχνολογιών. Ο server που φιλοξενεί την εφαρμογή-ιστοσελίδα είναι ο Apache, η γλώσσα προγραμματισμού που επικοινωνεί με το server είναι η PHP και η βάση δεδομένων είναι η MySQL. Στο επίπεδο πελάτη (client layer) χρησιμοποιούνται HTML, Javascript και JSON. Επιπλέον open-source βιβλιοθήκες ολοκληρώνουν την εμπειρία του πελάτη-χρήστη της εφαρμογής. Το περιβάλλον γραφής-κειμενογράφος (IDE) του λογισμικού είναι το Eclipse.

1.2 Σκοπός και στόχοι της εργασίας

Η εργασία αποσκοπεί στο να δώσει μία πρόταση στο τι θεωρεί αποτελεσματική διαχείριση

των υπολογιστικών συστημάτων ενός οργανισμού. Η αποτελεσματική διαχείριση οδηγεί σε ταχύτερη και καθαρότερη λήψη αποφάσεων. Η σωστή υπολογιστική οργάνωση μεταθέτει το θέμα που συζητείται στην πτυχιακή εργασία στο διοικητικό προσωπικό μίας εταιρείας και σχετίζεται με έρευνες που συναντώνται στο επιστημονικό πεδίο της λήψης αποφάσεων (decision tree structure). Παράλληλα, με τη χρήση εργαλείων ανοιχτού κώδικα η αποτελεσματική διαχείριση συνδυάζεται και με μικρό κόστος χρησιμοποίησής τους.

1.3 Δομή εργασίας

Ολοκληρώνοντας την εισαγωγή θα περιγραφεί σε επόμενα κεφάλαια συνοπτικά η δομή της εργασίας. Αρχικά θα περιγραφούν οι μέθοδοι υλοποίησης που εφαρμόστηκαν στο λογισμικό. Σειρά έχει το κύριο μέρος της πτυχιακής, όπου θα αναλυθεί το πρόβλημα, οι απαιτήσεις συστήματος, ο σχεδιασμός υλοποίησης και η τελικά η υλοποίηση. Το κεφάλαιο αυτό οργανώνεται ως εξής:

Τα δύο βασικά μέρη της εφαρμογής αφορούν το περιβάλλον του χρήστη (user) και το περιβάλλον του διαχειριστή (administrator). Οι χρήστες και οι διαχειριστές είναι εργαζόμενοι στον ίδιο όμιλο εταιρειών. Για κάθε μέρος θα συζητηθούν οι αλγόριθμοι που χρησιμοποιήθηκαν, οι τεχνολογίες που εφαρμόστηκαν για να ενσαρκώσουν τους παραπάνω αλγόριθμους, οι λειτουργίες που επιτελούνται σε κάθε μέρος καθώς και το γραφικό περιβάλλον κάθε χρήστη. Τέλος, θα ακολουθήσουν το βιβλιογραφικό μέρος και η μελλοντική εργασία και επεκτάσεις.

2 Μεθοδολογία υλοποίησης

Η εφαρμογή είναι δικτυακή (web-based), δηλαδή υπάρχει επικοινωνία μεταξύ client και server. Τα δεδομένα της εφαρμογής αποθηκεύονται σε βάση δεδομένων στο server, τη MySQL. Το μοντέλο αυτό είναι ευρέως γνωστό στο Web και χρησιμοποιείται από την πλειοψηφία (80%) των διαδικτυακών εφαρμογών. Η χρήση της εφαρμογής απαιτεί δίκτυο αλλά όχι απαραίτητα Διαδίκτυο, καθώς μπορεί να γίνει και ενδο-εταιρικά (intranet).

Σε αυτήν την ενότητα θα περιγραφούν τα θέματα που προκύπτουν κατά τη σχεδίαση και ανάπτυξη του βασικού συστατικού του λογισμικού, την οργάνωση των Υ/Σ. Όπως αναφέρθηκε η δενδροειδής δομή επιλέχθηκε για να στηρίζει το νεοσυσταθέν σύστημα της εταιρείας αλλά και να αφαιρέσει μέρος της πολυπλοκότητάς του, ενώ το γραφικό περιβάλλον των χρηστών να παραμένει απλό και στη διαισθητικό (intuitive) στη χρήση.

2.1. Μέθοδος ανάλυσης & ανάπτυξης πτυχιακής

Το πρώτο θέμα που προκύπτει είναι το πώς θα οργανωθούν τα Υ/Σ σε μορφή δένδρου. Αυτή η οργάνωση θα αναπαρασταθεί σε σχεσιακή βάση δεδομένων. Το γεγονός αυτό δε θα απασχολεί ιδιαίτερα τον τελικό χρήστη, διότι θα είναι κάτι για το οποίο θα είναι “ενήμερη” μόνο η βάση δεδομένων. Κρίθηκε λοιπόν ότι στη ΒΔ είναι αναγκαίο να φτιαχτεί ένας πίνακας για κάθε κλαδί του δένδρου. Σχηματίζεται έτσι μια μορφή ιεραρχίας ξεκινώντας από τον κορμό (root) και συνεχίζοντας στα υπόλοιπα κλαδιά του δένδρου, με την πληροφορία από γενική να εξελίσσεται σε ειδική.

Σε όρους πραγματικών δεδομένων, θεωρούμε ότι η εταιρεία ή αλλιώς όμιλος εταιρειών έχει μία εγκατάσταση ή πολλαπλές εγκαταστάσεις. Σε κάθε εγκατάσταση ανήκουν διάφορα τμήματα, όπως το τμήμα πληροφορικής και το τμήμα αποθήκης. Αυτά τα τμήματα περιλαμβάνουν αίθουσες ή υπολογιστικά συστήματα και δύναται ένα τμήμα να έχει παραπάνω από μία αίθουσες. Εδώ μπαίνει ο παράγοντας χρήστη (user), όπου και καθίσταται υπεύθυνος. Τα καθήκοντα του χρήστη είναι να επιβλέπει τους υπολογιστές και τα υποσυστήματα στην αίθουσα για την οποία είναι υπεύθυνος. Σε χαμηλότερο κλαδί ή φύλλο (leaf) του δένδρου ακολουθούν οι υπολογιστές, οι οποίοι όπως αναφέρθηκε βρίσκονται σε μία αίθουσα. Το τελευταίο παιδί του δένδρου είναι το υποσύστημα του υπολογιστή. Παράδειγμα αποτελεί το πληκτρολόγιο ή μία μητρική κάρτα εγκατεστημένη σε έναν υπολογιστή.

Σχηματίζονται, λοιπόν, τέσσερις πίνακες στη ΒΔ που αναπαριστούν την παραπάνω περιγραφείσα ιεραρχία: ο πίνακας premises, ο πίνακας departments, ο πίνακας halls και ο πίνακας computers. Στην παρακάτω εικόνα φαίνεται το διάγραμμα ER (entity-relationship). Οι οντότητες εννοιολογικά συνδέονται μεταξύ τους περιλαμβάνοντας μία στήλη που οδηγεί προς τον root κόμβο (με εξαίρεση τον root κόμβο).

Με αυτό το σχήμα (scheme) γνωρίζουμε τα παιδιά του δένδρου και μπορούμε τώρα να “περιηγηθούμε” προγραμματιστικά με ευκολία στη ΒΔ εκτελώντας τα ερωτήματα που απαιτούνται για τη γραφή της εφαρμογής. Επιπρόσθετα, διευκολύνεται η λήψη αποφάσεων για το λόγο ότι υπάρχει απλή και σαφής οργάνωση των τμημάτων και έτσι συγκεκριμένες αποφάσεις μπορούν να ληφθούν για συγκεκριμένα τμήματα.

2.2 Θεωρία

Σε αυτή την ενότητα θα περιγραφεί σχετική έρευνα που έγινε στη θεωρία των πληροφοριακών συστημάτων, στο θέμα των δένδροειδών ή ιεραρχικών δομών και στο αντικείμενο της διαχείρισης καταγραφής συμβάντων ως λογισμικό. Από την παρούσα πτυχιακή εργασία ζητείται η κατασκευή ενός πληροφοριακού συστήματος σε μια πολύ βασική και απλή μορφή του. Αρχικά, οφείλουμε να προσδιορίσουμε τι εστί ένα πληροφοριακό σύστημα και πώς χρησιμοποιείται σήμερα από τους οργανισμούς.

Υπάρχουν τρεις δομικές ιδιότητες σε ένα σύστημα. Πρώτον, ένα σύστημα έχει ένα σκοπό, όπως η διανομή νερού που θα ποτίσει ένα φυτό. Δεύτερον ένα σύστημα ομαδοποιεί δύο ή περισσότερα συστατικά, τα οποία συνδέονται μεταξύ τους μέσω ενός κοινού και συνεκτικού δεσμού. Στο παραπάνω παράδειγμα, ο δεσμός είναι το νερό στο σύστημα ύδρευσης. Και τρίτον, ένα σύστημα λειτουργεί καθημερινά και επαναληπτικά με την έννοια ότι είναι προβλέψιμος ο τρόπος λειτουργίας και παραγωγής του. Όλα τα συστήματα μοιράζονται αυτά τα βασικά χαρακτηριστικά. Η απουσία ενός από αυτά δεν αρκεί για να προσδιορίσει ένα σύστημα.

Η ανάπτυξη μίας διάταξης ή η ομαδοποίηση των συστατικών έχει ως στόχο την παραγωγή πληροφορίας για την υποστήριξη των ενεργειών και των αποφάσεων ενός οργανισμού. Τα δεδομένα είναι η πρώτη ύλη και είναι απαραίτητα για την παραγωγή της πληροφορίας. Χωρίς την πληροφορία τα δεδομένα δεν έχουν λόγο ύπαρξης. Είναι απλά τα στοιχεία που χρησιμοποιούνται για να αναγνωρίσουν, περιγράψουν ή ποσοτικοποιήσουν ένα αντικείμενο που χρησιμοποιείται από έναν οργανισμό.

Όταν τα δεδομένα παρουσιάζονται στον άνθρωπο σε μία συγκεκριμένη διάταξη για χρήση από αυτόν, τότε μετατρέπονται σε πληροφορία. Εάν ο άνθρωπος δεν είναι σε θέση να ενεργήσει ή να λάβει αποφάσεις από αυτά, τα δεδομένα αποτελούν απλά ωμή πληροφορία. Αυτό υποδηλώνει ότι τα δεδομένα αποθηκεύονται και η πληροφορία, όπως αναφέρθηκε, παράγεται. Αυτή εξαρτάται από τις επιθυμίες και τις ανάγκες του ανθρώπου, όπως για παράδειγμα ο πελάτης. Η πληροφορία, λοιπόν, μπορεί να χαρακτηριστεί ως η ευφυΐα ή διορατικότητα που έχει αποκτηθεί από την επεξεργασία/ανάλυση των δεδομένων. Όσον αφορά τα πληροφοριακά συστήματα επιχειρήσεων, η πληροφορία είναι σηματικός πόρος στη λειτουργία και στη διαχείρισή τους. Η χρονική διαθεσιμότητα μιας πληροφορίας είναι ζωτική για την αποτελεσματικότητα των διαχειριστικών λειτουργιών, όπως η μεθόδευση, η οργάνωση, η ηγεσία και ο έλεγχος.

Μία επιπλέον μεταβλητή που προκύπτει από τα παραπάνω είναι η “επεξεργασία”, η οποία καθορίζει πώς τα δεδομένα συλλέγονται και ανακτώνται με σκοπό την παραγωγή πληροφορίας. Αυτή η διαδικασία λαμβάνει χώρα όταν ο άνθρωπος χρειάζεται να ενεργήσει ή να πάρει αποφάσεις πάνω σε αυτήν την πληροφορία.

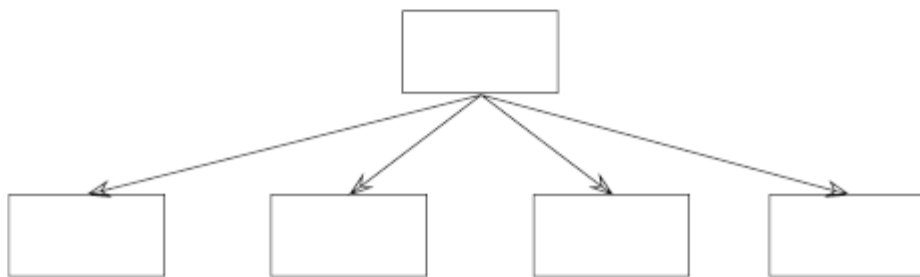
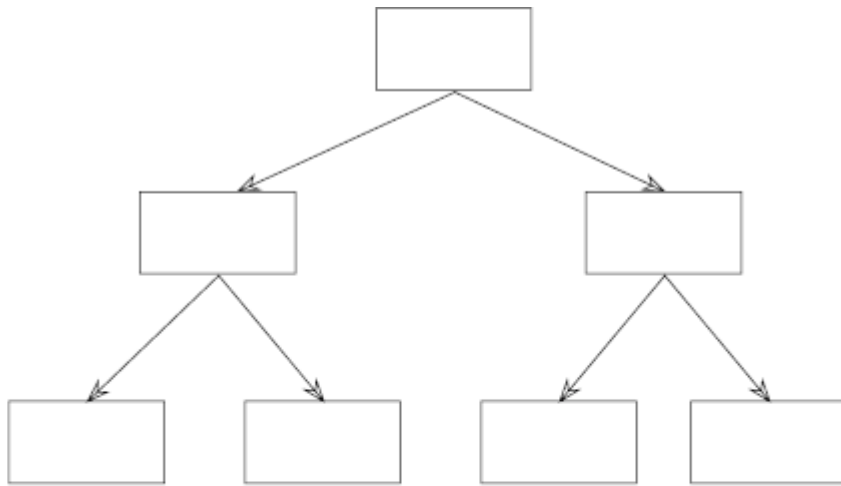
Ένα πληροφοριακό σύστημα είναι μία συλλογή από επεξεργασίες (υπο-συστήματα) για συλλογή, αποθήκευση και ανάκτηση πληροφορίας. Ο συνεκτικός δεσμός ανάμεσα σε αυτές τις επεξεργασίες είναι τα δεδομένα, τα οποία πρέπει να διανεμηθούν και να επαναχρησιμοποιηθούν μέσα από ένα σύστημα. Με άλλα λόγια, σε αυτό το σημείο τα υπο-συστήματα του συνολικού συστήματος ορίζουν ποια δεδομένα πρέπει να επεξεργαστούν και ποιος θα καταναλώσει την πληροφορία (π.χ. τελικοί χρήστες), αλλά δεν ορίζει πώς το σύστημα θα υλοποιηθεί.

Ακολουθώς, οι προγραμματιστές καθορίζουν μία κατάλληλη προσέγγιση για τη φυσική υλοποίηση κάθε υπο-συστήματος. Η απόφαση αυτή βασίζεται στην υλοποιησιμότητά του και στην αποδοτικότητά του. Τα υπο-συστήματα δύνανται να υλοποιηθούν χρησιμοποιώντας χειροκίνητες ή υπολογιστικές διαδικασίες, ή εργασίες αυτοματισμού γραφείου ή συνδυάζοντας και τα τρία. Ανάλογα την πολυπλοκότητα του κάθε υπο-συστήματος, προκύπτουν ορισμένες διαδικασίες. Ανεξάρτητα από τις διαδικασίες που θα ακολουθηθούν οι προγραμματιστές πρέπει να ορίσουν τις συσχετίσεις κατά την εκτέλεση των διαδικασιών είτε διαδοχικά είτε επαναληπτικά αφήνοντας χώρο και για εναλλακτικές λύσεις. Ορίζοντας με αυτόν τον τρόπο τις διαδικασίες, από την αρχή ως το τέλος, οι προγραμματιστές καθορίζουν την “ροή εργασιών” κάθε υπο-συστήματος, το οποίο προσδιορίζει πώς τα δεδομένα δημιουργούνται, επεξεργάζονται, ενημερώνονται και αναφέρονται.

Ένα πληροφοριακό σύστημα είναι ένα προϊόν που σχεδιάζεται και κατασκευάζεται σαν οποιοδήποτε άλλο προϊόν. Μπορεί να αναπαρασταθεί ως μία τεσσάρων επιπέδων ιεραρχία, γνωστή ως κανονική συστημική δομή. Κάθε επίπεδο αναπαριστά ένα διαφορετικό επίπεδο αφαιρετικότητας για κάθε σύστημα, προχωρώντας από το γενικό στο ειδικό, δηλαδή η σχεδίαση είναι από πάνω προς τα κάτω.

- Επίπεδο 1 – Σύστημα
- Επίπεδο 2 – Υπο-συστήματα: 2 ή περισσότερες
- Επίπεδο 3 - Διαδικασίες (χειροκίνητες, υπολογιστικές, αυτοματισμού γραφείου): 1 ή περισσότερες για κάθε υπο-σύστημα
- Επίπεδο 4 – Προγράμματα (για υπολογιστικές διαδικασίες) και Βήματα για τα υπόλοιπα: 1 ή περισσότερα για κάθε διαδικασία

Η ιεραρχία σύμφωνα με τον Arthur M. Langer είναι αρχικά η γραφική απεικόνιση των λειτουργικών οντοτήτων του μοντέλου που έχει εφαρμογή. Οι ιεραρχικές δομές λέγονται τυπικά δένδρα. Η έννοια του δένδρου είναι ότι καθώς κατευθυνόμαστε από πάνω προς τα κάτω στην ιεραρχία, ο αριθμός των φύλλων-παιδιών αυξάνεται και το δένδρο μεγαλώνει οριζόντια. Η γενική εικόνα ομοιάζει με ένα φυσικό δένδρο. Στην παρούσα εργασία η κίνηση προς τα εμπρός από τον έναν πατέρα-πίνακα στο παιδί-πίνακα αναπαριστάται καθαρά, ενώ η κίνηση προς τα πίσω είναι αντίθετη κάνοντας σε κάθε περίπτωση μικρά και σίγουρα βήματα. Ένα διάγραμμα στην απλή μορφή του που συναντάται στη βιβλιογραφία είναι το παρακάτω (Analysis and design of Information Systems by Arthur M. Langer).

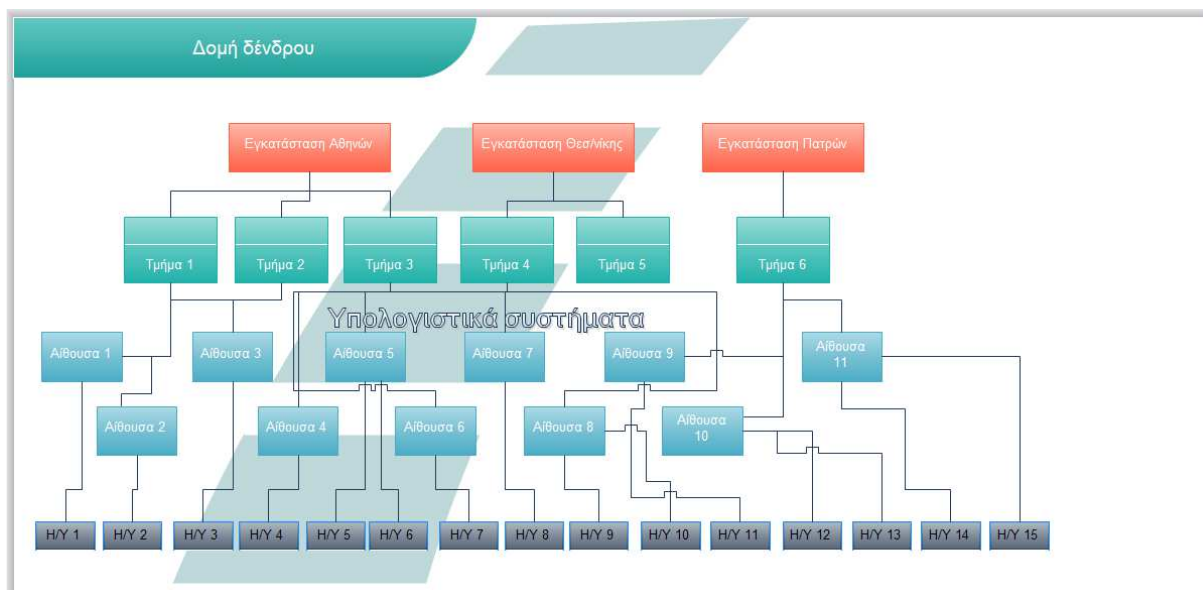


δομή στην απλή μορφή της

Εικόνα 2.3: Ιεραρχική

Εξειδικεύοντας την παραπάνω έννοια της ιεραρχίας η παρούσα πτυχιακή επικεντρώνεται στην παροχή εργαλείων στο διοικητικό προσωπικό ενός οργανισμού για τη δημιουργία ενός πλαισίου λήψης αποφάσεων (decision tree). Σύμφωνα με δημοσίευση στο περιοδικό *Organizational Dynamics* η ιεραρχία στα συστήματα επιτρέπει στο διαχειριστή που θα λάβει τις αποφάσεις για την εταιρεία να κάνει διάγνωση του περιβάλλοντος αποφάσεων (decision environment). Στόχος της ιεραρχικής δομής είναι ο διαχωρισμός των τμημάτων σε υπό-τμήματα για αφαίρεση της πολυπλοκότητας του συνολικού συστήματος. Δίνει τη δυνατότητα στο διαχειριστή να λαμβάνει αποφάσεις για ένα σύστημα δίχως η αλληλεξάρτηση των υπό-τμημάτων να στέκεται εμπόδιο στην κατανόησή τους. Οι λειτουργίες ενός οργανισμού διανέμονται απροβλημάτιστα στα υπό-τμήματα, με το κάθε ένα από αυτά να εκπληρώνει το μεμονωμένο σκοπό για το οποίο δημιουργήθηκε. Ο έλεγχος της συνεχώς αυξανόμενης πληροφορίας τη σημερινή εποχή έχει γίνει ιδιαίτερα δύσκολος. Η διεύθυνση ενός οργανισμού, λοιπόν, θα δημιουργήσει τα δένδρα αποφάσεων για τα υποσυστήματά της, τα οποία θα ρυθμίζει αυτόνομα και θα διευκολύνει τη βελτίωσή τους στα πλαίσια των αναγκών του οργανισμού.

Παρακάτω παρατίθεται ένα διάγραμμα που απεικονίζει την αντίστοιχη δενδροειδή δομή που έχει σχεδιαστεί στη βάση δεδομένων για την παρούσα εργασία. Ο κορμός του δένδρου είναι οι εγκαταστάσεις της εταιρείας και η κατεύθυνση προς τη συγκεκριμενοποίηση των λειτουργιών του κάθε κόμβου-παιδί είναι προς τα κάτω.



Εικόνα 2.4: Αφηρημένη ιεραρχική δομή που αναπαριστάται από τη βάση δεδομένων

Σημαντικό μέρος της πτυχιακής εργασίας ασχολείται με τη διαχείριση καταγραφής συμβάντων. Στην επιστήμη της πληροφορικής, η διαχείριση καταγραφών (log management) εν γένει και η ανάλυσή τους (ή αλλιώς η καταγραφή συστημάτων και δικτύων) είναι μία τέχνη και επιστήμη που επιχειρεί να καταλάβει τις εγγραφές που παράγονται αυτόματα από τους υπολογιστές. Η διαδικασία παραγωγής τέτοιου είδους εγγραφών ονομάζεται καταγραφή δεδομένων (data logging). Παρακάτω αναφέρονται ενδεικτικοί λόγοι για τους οποίους εκτελείται ανάλυση καταγραφών:

- Συμμόρφωση με πολιτικές ασφαλείας
- Συμμόρφωση με ελέγχους (audits) και κανονισμούς (regulations)
- Εντοπισμός βλαβών σε συστήματα
- Απάντηση σε συμβάντα (incidents) ασφαλείας

Οι καταγραφές παράγονται από συσκευές δικτύου, λειτουργικά συστήματα, εφαρμογές και από κάθε άλλη προγραμματιζόμενη συσκευή. Μία ροή μηνυμάτων σε χρονική ακολουθία συχνά συνθέτει μία καταγραφή. Η καταγραφή μπορεί να κατευθύνεται σε αποθηκευμένα αρχεία στο δίσκο ή σε ένα δικτυακό συλλέκτη πληροφοριών. Τα μηνύματα καταγραφής πρέπει να ερμηνευτούν λαμβάνοντας υπόψη την εσωτερική κατάσταση της πηγής (όπως μία εφαρμογή) και αναφέρουν συμβάντα σχετικά με την ασφάλεια ή τις λειτουργίες-δραστηριότητες (όπως η εισαγωγή χρηστών σε ένα σύστημα ή ένα σφάλμα συστήματος).

Οι καταγραφές δημιουργούνται από μηχανικούς λογισμικού στα πλαίσια της αποσφαλμάτωσης μίας λειτουργίας σε ένα πρόγραμμα. Η σύνταξη και η σημασιολογία των δεδομένων που περιλαμβάνονται στα μηνύματα καταγραφής συνήθως σχετίζονται με εφαρμογές του ίδιου του οργανισμού ή με εφαρμογές τρίτων. Η ορολογία των μηνυμάτων ανάμεσα σε διαφορετικές εφαρμογές μπορεί να ποικίλλει. Για παράδειγμα, η ταυτοποίηση ενός χρήστη σε ένα πρόγραμμα μπορεί να περιγραφεί ως είσοδος ή σύνδεση ή ταυτοποίηση. Για αυτό το λόγο η ανάλυση καταγραφών πρέπει να ερμηνεύει τα μηνύματα στα πλαίσια (context) κάθε εφαρμογής, συστήματος, ή ρύθμισης για να δημιουργηθούν χρήσιμες συγκρίσεις για τα μηνύματα από τις διαφορετικές πηγές καταγραφής.

Η μορφή ή το περιεχόμενο των μηνυμάτων καταγραφής ενδέχεται να μην είναι πάντοτε τεκμηριωμένη. Ένα από τα καθήκοντα του αναλυτή καταγραφών είναι να προτρέψει (induce) το σύστημα να εξάγει μεγάλη κλίμακα μηνυμάτων έτσι ώστε να κατανοήσει το ολοκληρωμένο πεδίο ορισμού εντός του οποίου τα μηνύματα πρέπει να ερμηνευθούν. Ο αναλυτής καταγραφών μπορεί να αντιστοιχίσει την ορολογία από τις διάφορες πηγές καταγραφών σε μία ομοιόμορφη, ενοποιημένη και κανονικοποιημένη ορολογία έτσι ώστε οι αναφορές και τα στατιστικά να προέρχονται από ετερογενή περιβάλλοντα. Για παράδειγμα, τα μηνύματα καταγραφής από λειτουργικά συστήματα όπως τα Windows και τα Linux αλλά και από βάσεις δεδομένων μπορούν να συναθροιστούν σε μία κανονικοποιημένη αναφορά για κάποιον ελεγκτή (auditor) ή διευθυντή. Υπάρχουν, επίσης, συστήματα τα οποία εξάγουν μηνύματα προτεραιοτήτων, όπως σφάλμα, προειδοποίηση κλπ. με σκοπό τη διευκόλυνση του έργου του αναλυτή. Συνεπώς, οι τεχνικές ανάλυσης καταγραφών συναντώνται συχνά και ως τεχνικές ανάκτησης κειμένου ή ως αντίστροφη μηχανική λογισμικού (reverse engineering software).

Το λογισμικό που χρησιμοποιείται για να αυτοματοποιήσει τις άνω πρακτικές ονομάζεται διαχείριση καταγραφών συμβάντων (Event log management software – ELMS). Πρόκειται για εφαρμογές που χρησιμοποιούνται για την παρακολούθηση στην αλλαγή διαχείρισης και στην προετοιμασία της συμμόρφωσης των ελέγχων στις επιχειρήσεις. Το λογισμικό αυτό αποτελεί ένα εργαλείο για ένα IT διαχειριστή (IT administrator), ο οποίος καλείται να επιδείξει στους διοικητικούς υπάλληλους ότι ένας οργανισμός είναι προετοιμασμένος στα πλαίσια του ελέγχου συμμόρφωσης. Ως δυνατότητες προγράμματος αναφέρονται:

- Δημιουργία αυτοματοποιημένων αντιγράφων ασφαλείας (backups)
- Δημιουργία και διαγραφή χρηστών
- Παρακολούθηση ειδοποιήσεων σε πραγματικό χρόνο
- Σύνθετες δυνατότητες αναφορών

Οι IT διαχειριστές μπορούν να πληροφορηθούν στιγμιαία για πιθανές παραβιάσεις ασφαλείας, όπως ένας πρώην εργαζόμενος που επιχειρεί να διαγράψει σημαντικά δεδομένα ή να ανακτήσει ήδη διαγεγραμμένα δεδομένα. Ζωτικής σημασίας χαρακτηριστικό σε ένα πρόγραμμα αποτελεί η μη μεταβολή στα μηνύματα καταγραφής. Οι ελεγκτές συμμόρφωσης απαιτούν τα δεδομένα να προέρχονται κατευθείαν από την εφαρμογή για την οποία γίνεται καταγραφή για να αποδειχθεί ότι οι εγγραφές που καταγράφονται δεν τροποποιούνται, παραποιούνται ή διαγράφονται.

Η παρούσα πτυχιακή εργασία δεν εστιάζει στην ανάλυση των αρχείων καταγραφής, λαθόν τα συμβάντα καταγράφονται χειροκίνητα από το προσωπικό και σε μορφή κατανοητή από τον άνθρωπο. Όποια ανάλυση γίνεται πάνω στα δεδομένα πραγματοποιείται από τη βάση δεδομένων και θεωρείται στοιχειώδης, για το λόγο ότι το σύστημα επιτρέπει την καταγραφή μηνυμάτων με συγκεκριμένο τρόπο που κατονοούνται τόσο από τον άνθρωπο όσο και από τον υπολογιστή. Οι αναφορές και τα αναλυτικά στατιστικά που εξάγει το σύστημα που έχει δημιουργηθεί δεν απαιτούν την παρουσία αναλυτή, δηλαδή είναι αυτοματοποιημένες.

3 Κύριο μέρος του προβλήματος

Σε αυτό το κεφάλαιο θα συζητηθούν η ανάλυση του προβλήματος, οι απαιτήσεις συστήματος, ο σχεδιασμός υλοποίησης και τελικά η υλοποίηση. Σημειώνεται ότι η γραμματοσειρά που χρησιμοποιείται σε ορισμένα σημεία του κεφαλαίου (Courier) αφορά σε αρχεία και φακέλους της εφαρμογής για αποφυγή αντίστοιχων τεχνολογικών όρων. Παρόμοια, χρησιμοποιείται για να επισημανθούν μέρη του κώδικα, όπως συναρτήσεις, μέθοδοι και μεταβλητές, αλλά και πίνακες και στήλες στη βάση δεδομένων.

3.1 Ανάλυση προβλήματος

Το λογισμικό αναγνωρίζει δύο ομάδες χρηστών για τους οποίους θα σχεδιαστούν διαφορετικές σελίδες (views), μία για τους χρήστες (users) και μία για τους διαχειριστές (administrators). Οι πρώτοι χρησιμοποιούν την εφαρμογή για να καταγράφουν τα συμβάντα που προκύπτουν στα Υ/Σ, έχοντας παράλληλα τη δυνατότητα να βλέπουν στατιστικά των συμβάντων του τμήματος-υποκαταστήματος (branch) του οργανισμού στον οποίο ανήκουν. Και οι δύο ομάδες μοιράζονται ένα αναλυτικό ημερολόγιο για τα συμβάντα που καταγράφονται. Εκτός από τη γραφική απεικόνιση του χρόνου που έλαβε χώρα το συμβάν, εμφανίζεται και η αναφορά κάθε συμβάντος, η οποία συνοδεύει την καταγραφή του. Οι δεύτεροι (διαχειριστές) έχουν τη δυνατότητα να εμφανίσουν τη στατιστική ανάλυση όλων των συμβάντων για όλα τα τμήματα της εταιρείας. Εκτός αυτού, οι διαχειριστές έχουν δικαιώματα εισαγωγής νέων χρηστών στη βάση δεδομένων (ΒΔ) (παράδειγμα αποτελεί μία νέα πρόσληψη εργαζομένου) και η ανάθεσή τους αιθουσών υπό την ευθύνη τους, αλλά και εισαγωγή νέων υπολογιστών και Υ/Σ στη ΒΔ.

3.1.2 Απαιτήσεις συστήματος

Κύριο μέλημα στην εφαρμογή αποτελεί η υλοποίηση του ημερολογίου. Αυτό θα επιτελεί εκτός από τις κοινές λειτουργίες ενός ημερολογίου και λειτουργίες απεικόνισης των συμβάντων που καταγράφονται καθημερινά σε σχέση με το οποίο ανήκει ο χρήστης. Κάθε συμβάν, είτε πρόκειται για βλάβη είτε πρόκειται για συντήρηση πρέπει να έχει μια χρωματική ταυτότητα και θα εμφανίζεται στην ημέρα και ώρα που καταγράφη στο ημερολόγιο. Επιπλέον, θα εμφανίζεται το Υ/Σ και ο υπολογιστής τον οποίο αφορά το συμβάν καθώς και η σχετική αναφορά που δόθηκε στην καταγραφή.

Η επόμενη λειτουργία που θα υποστηρίζει η σελίδα (view) του χρήστη είναι ασφαλώς η καταγραφή των συμβάντων. Ο χρήστης πρέπει να επιλέγει το Υ/Σ και τον υπολογιστή στον οποίο ανήκει και θα καταγράφει το είδος του συμβάντος μαζί με την αναφορά. Θα έχει επίσης τη δυνατότητα να εμφανίσει αναλυτικά στατιστικά των συμβάντων σε εύρος ενός χρόνου του τμήματος που ανήκει ο χρήστης. Την ίδια λειτουργία θα μοιράζεται και ο διαχειριστής.

Οι διαχειριστές πρέπει να έχουν αναλυτικό ημερολόγιο παρόμοια με τους χρήστες, αλλά θα είναι πιο γενικό δείχνοντας τα συμβάντα που έχουν λάβει χώρα για όλα τα τμήματα. Επίσης, οι διαχειριστές έχουν καθήκοντα εισαγωγής νέων χρηστών στη ΒΔ, νέα υπολογιστικών συστημάτων και νέων υπολογιστών. Παρατίθεται το έγγραφο τεχνικών απαιτήσεων του συστήματος:

1. Υλοποίηση εισόδου/εξόδου των χρηστών από το σύστημα.
2. Δημιουργία δύο ομάδων χρηστών που θα χρησιμοποιούν το σύστημα.

3. Καταγραφή των συμβάντων στα υπολογιστικά συστήματα.
4. Απεικόνιση των συμβάντων στο αναλυτικό ημερολόγιο.
5. Εμφάνιση αναλυτικών στατιστικών για τα συμβάντα σε μορφή πινάκων και διαγραμμάτων.
6. Καταχώρηση νέων χρηστών στο σύστημα.
7. Καταχώρηση νέων υπολογιστικών συστημάτων και των υπολογιστών που τα απαρτίζουν.

3.2 Σχεδιασμός υλοποίησης

Αναφέρθηκε ότι οι τεχνολογίες που χρησιμοποιούνται είναι web και open-source. Για αυτό το λόγο θα χρησιμοποιηθεί Apache server σε Linux Operating system. Η γλώσσα επικοινωνίας με το διακομιστή (server) είναι PHP 5.4. Η βάση δεδομένων είναι MySQL 5. Στο επίπεδο του πελάτη θα χρησιμοποιηθεί HTML5, πράγμα που σημαίνει ότι υπάρχει υποστήριξη μόνο για HTML5 browsers, γλώσσα Javascript και ανοιχτού κώδικα βιβλιοθήκες Javascript. Αναφέρονται οι βιβλιοθήκες: jQuery για ευκολότερο και ταχύτερο έλεγχο του Document Object Model (DOM) της σελίδας, bootstrap για τη μορφοποίηση και το responsiveness της ιστοσελίδας, fullcalendar για το αναλυτικό ημερολόγιο και highcharts για τη διαγραμματική παρουσίαση των στατιστικών.

Οι σελίδες και των δύο χρηστών είναι ισοδύναμα χωρισμένες στα δύο. Αριστερά υπάρχει το ημερολόγιο, ενώ δεξιά υπάρχουν οι συγκεκριμένες λειτουργίες για κάθε ομάδα χρήστη. Οι διαφορετικές λειτουργίες διαιρούνται με τη σειρά τους σε ετικέτες (tabs) με κάθε ετικέτα να καταλαμβάνει κάθε φορά ολόκληρη τη δεξιά περιοχή αποκρύπτοντας τις υπόλοιπες. Κάθε φορά είναι ορατή μία ετικέτα, η οποία δείχνει το περιεχόμενο της ανάλογα με την επιλογή του χρήστη.

Στο αριστερό μέρος ο χρήστης (user) βλέπει το ημερολόγιο και συγκεκριμένα όλες τις ημέρες μέσα στο μήνα. Επιτρέπεται με βέλη η πλοήγηση πίσω ή μπροστά ένα μήνα. Υπάρχει η επιλογή η εμφάνιση του ημερολογίου είτε ανά μήνα είτε ανά εβδομάδα. Στην παρουσίαση ανά μήνα εμφανίζονται οι μέρες του μήνα με τετράγωνα. Για τις ημέρες που έχουν καταγραφεί συμβάντα εμφανίζεται εντός του αντίστοιχου τετραγώνου ένα μικρότερο σε διαστάσεις ορθογώνιο παραλληλόγραμμο και αυτό περιλαμβάνει την ώρα του συμβάντος. Πράσινο παραλληλόγραμμο σημαίνει ότι κατεγράφη συντήρηση, ενώ κόκκινο σημαίνει καταγραφή βλάβης. Η παρουσίαση του ημερολογίου ανά εβδομάδα δίνει μία παραπάνω πληροφορία, τη διάρκεια του συμβάντος, αφού παράλληλα με τις ημέρες εμφανίζονται και οι ώρες της ημέρας. Η διάρκεια του συμβάντος συμπεραίνεται από το ύψος του παραλληλόγραμμου, για το λόγο ότι οι ώρες της ημερών είναι τοποθετημένες σε κάθετη σειρά.

Στο πρώτο tab ο χρήστης έχει μια φόρμα για συμπλήρωση και καταγραφή των συμβάντων. Ένας χρήστης ανήκει σε ένα τμήμα και είναι υπεύθυνος για μία αίθουσα. Σε αυτήν την αίθουσα βρίσκονται οι υπολογιστές τους οποίους επιβλέπει. Ο χρήστης μέσα από μία λίστα που του δίνεται μπορεί να επιλέγει τον υπολογιστή που χρειάζεται να καταγράψει. Η καταγραφή της αναφοράς γίνεται σε μία περιοχή κειμένου. Ο χρήστης επιλέγει το είδος του συμβάντος (συντήρηση ή βλάβη) και τα λεπτά που χρειάστηκαν για την αποκατάσταση της βλάβης ή τη διάρκεια της συντήρησης. Η αναφορά μπορεί να περιέχει και εξάρτημα ή περιφερειακό του υπολογιστή στο οποίο εντοπίστηκε η βλάβη.

Στο δεύτερο tab ο χρήστης μπορεί να εμφανίσει στατιστικά των συμβάντων στο τμήμα του. Μέσα από ένα μενού μπορεί να επιλέξει να δει τα στατιστικά των συμβάντων για το τμήμα του είτε μέσω διαγραμμάτων είτε μέσω πινάκων. Τα στατιστικά που προβάλλει ο χρήστης είναι ετήσια και απεικονίζουν τα Υ/Σ (και το πλήθος τους) για τα οποία έγινε καταγραφή για κάθε μήνα μέσα στο έτος.

Συνεχίζουμε με τη σελίδα του διαχειριστή (administrator). Στο αριστερό μέρος εμφανίζεται το ημερολόγιο περιλαμβάνοντας τα συμβάντα για όλα τα τμήματα. Στο δεξί μέρος ο διαχειριστής βλέπει τρεις ετικέτες (tabs) που περιλαμβάνει το καθένα μία φόρμα. Στο πρώτο tab ο διαχειριστής εισάγει νέο υπολογιστικό σύστημα και επιλέγει τον υπολογιστή στον οποίο θα ανήκει και τι εξαρτήματα και περιφερειακά θα περιλαμβάνει ο υπολογιστής. Μπορεί, επίσης, να προσθέσει υπολογιστές σε ήδη

υπάρχον υπολογιστικό σύστημα. Στο επόμενο tab δίνεται η δυνατότητα στο διαχειριστή να εισάγει νέο χρήστη (user). Η εισαγωγή απαιτεί το username του χρήστη και το password που αυτός θα χρησιμοποιεί. Μέσα από τη φόρμα ο διαχειριστής επιλέγει το τμήμα στο οποίο θα ανήκει ο χρήστης. Δοθέντων όλων των παραπάνω στοιχείων γίνεται η εισαγωγή του χρήστη. Ο διαχειριστής μπορεί να εμφανίσει τα στατιστικά των συμβάντων στο τρίτο tab, παρόμοια με το χρήστη. Υπάρχει μια διαφορά όμως σε σχέση με τα στατιστικά που εμφανίζονται στο χρήστη. Ο διαχειριστής επιπλέον μπορεί να εμφανίσει και στατιστικά όχι μόνο ανά είδος συμβάντος αλλά και ανά τμήμα, δηλαδή πόσο “ενεργό” είναι ένα τμήμα για μια κατηγορία συμβάντος μέσα σε μία περίοδο ενός έτους.

3.3 Υλοποίηση

Σε αυτήν την ενότητα θα συζητηθεί αναλυτικά η υλοποίηση της εφαρμογής. Συγκεκριμένα θα γίνει μια εισαγωγή στη θεωρία και στον τρόπο που λειτουργεί το HTTP πρωτόκολλο και οι εφαρμογές που αναπτύσσονται με αυτό. Έπειτα, θα γίνει παρουσίαση του σχεδιασμού της βάσης δεδομένων και πώς αυτός ο σχεδιασμός συμβάλλει στην εύκολη διαχείριση και αποτελεσματικότητα της εφαρμογής. Στη συνέχεια θα συζητηθούν τα μεμονωμένα αρχεία (script files) στα επίπεδα διακομιστή και πελάτη και πώς αυτά συνθέτουν τον τελικό αποτέλεσμα. Θα παρασχεθεί πλήρες εγχειρίδιο χρήσης στο τέλος του κεφαλαίου.

3.4 Εισαγωγή στις διαδικτυακές εφαρμογές

3.4.1 Το μοντέλο client-server

Η εφαρμογή που αναπτύχθηκε στα πλαίσια της πτυχιακής εργασίας και συγκεκριμένα οι τεχνολογίες που κάνει χρήση είναι δομημένες στο μοντέλο client-server. Πρόκειται για ένα υπολογιστικό μοντέλο που ενεργεί ως εφαρμογή-διανεμητής, η οποία τεμαχίζει τις εργασίες μεταξύ των παρόχων πόρων ή υπηρεσιών (servers – εξυπηρετητές) και αυτών που ζητάνε τις υπηρεσίες (clients – πελάτες). Στον εξυπηρετητή “τρέχουν” ένα ή περισσότερα προγράμματα τα οποία μοιράζονται πόρους με τους πελάτες. Ο πελάτης δεν μοιράζεται κανέναν από τους ίδιους πόρους, αλλά ζητά ή καλύτερα πραγματοποιεί αιτήσεις για περιεχόμενο από τον εξυπηρετητή. Οι πελάτες εκκινούν συνεδρίες επικοινωνίας με τους εξυπηρετητές, οι οποίοι αναμένουν για εισερχόμενες αιτήσεις. Το μοντέλο client-server περιγράφει τη σχέση που έχουν συνεργαζόμενα υπό-προγράμματα σε μία κύρια εφαρμογή. Τα υπό-προγράμματα είναι οι web τεχνολογίες που χρησιμοποιεί η εφαρμογή της εργασίας και θα εξηγηθούν σε επόμενες ενότητες.

3.4.2 Το επίπεδο client

Ένας πελάτης (client) είναι τυπικά μία εφαρμογή υπολογιστή, όπως ένας περιηγητής ιστοσελίδων, που “τρέχει” σε τοπικούς σταθμούς εργασίας χρηστών και συνδέεται με έναν εξυπηρετητή όταν είναι απαραίτητο. Οι λειτουργίες εκτελούνται στην πλευρά του client για το λόγο ότι απαιτούν πρόσβαση σε πληροφορία η οποία είναι διαθέσιμη στον πελάτη και όχι στον εξυπηρετητή. Τέτοιου είδους λειτουργίες είναι συνήθως η απόδοση κειμένου σε μία ιστοσελίδα και η αναμονή για εισαγωγή πληροφορίας από το χρήστη.

3.4.3 HyperText Markup Language (HTML)

Η HTML “τρέχει” στην πλευρά του client και είναι η κύρια γλώσσα για την εμφάνιση ιστοσελίδων και άλλων πληροφοριών μέσω ενός περιηγητή. Συντάσσεται με τη μορφή HTML στοιχείων τα οποία αποτελούνται από ετικέτες (tags) που περικλείονται από γωνιακές αγκύλες, όπως <html>, εντός του περιεχομένου της ιστοσελίδας. Οι HTML ετικέτες συναντώνται σε ζεύγη διαθέτοντας ετικέτες αρχής και τέλους, όπως <h1> και </h1>, ενώ μερικά στοιχεία δεν έχουν ετικέτα τέλους, π.χ. το . Ανάμεσα στις ετικέτες τοποθετείται κείμενο, σχόλια και εμφωλευμένες ετικέτες. Ο περιηγητής ιστοσελίδων “διαβάζει” τα έγγραφα HTML και τις μετατρέπει σε οπτικές ή ακουστικές ιστοσελίδες. Οι HTML ετικέτες δεν εμφανίζονται, αλλά ο περιηγητής τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της ιστοσελίδας.

3.4.4 Cascading Style Sheets (CSS)

Τα φύλλα ή στυλ CSS είναι μία γλώσσα κανόνων και χρησιμοποιείται για να περιγράψει τη μορφή των αρχείων που έχουν συνταχθεί με μία γλώσσα σήμανσης, όπως την HTML. Κύρια λειτουργία της είναι να μορφοποιεί έγγραφα ιστοσελίδων, προσθέτοντας γραμματοσειρές, χρώμα, στοίχιση και άλλες επιλογές. Τα στυλ CSS δημιουργήθηκαν για να διαφοροποιήσουν το περιεχόμενο των εγγράφων από την αντίστοιχη μορφοποίηση τους έτσι ώστε να παρέχουν μεγαλύτερη ευελιξία και έλεγχο στην παρουσίαση των στοιχείων των εγγράφων και δίνοντας τη δυνατότητα σε πολλαπλές ιστοσελίδες να μοιράζονται μορφοποιήσεις, μειώνοντας με αυτό τον τρόπο την πολυπλοκότητα και την επανάληψη κώδικα. Τα CSS περιγράφουν ένα σχέδιο (scheme) προτεραιοτήτων για να καθορίσουν ποιοι κανόνες στυλ εφαρμόζονται σε ένα έγγραφο στην περίπτωση που ισχύουν περισσότεροι του ενός κανόνες. Οι προτεραιότητες, τα βάρη υπολογίζονται και ανατίθενται σε κανόνες και ισχύουν σε ένα έγγραφο διαδοχικά (cascade).

3.4.5 Javascript

Η γλώσσα client Javascript είναι μία αντικειμενοστραφής δυναμική, χαλαρών τύπων γλώσσα σεναρίων βασισμένη στο Prototype (οι κλάσεις δεν είναι ορατές στον προγραμματιστή) και υποστηρίζει διαδικαστικά και συναρτησιακά προγραμματιστικά στυλ. Το Javascript κώδικα εκτελούν οι περιηγητές ιστού και χρησιμοποιείται για να προσδώσει σε ένα έγγραφο ιστοσελίδας εμπλουτισμένη διεπαφή με το χρήστη και δυναμικότητα. Η δημοτικότητα της Javascript ήρθε με τον καιρό καθώς ολοένα και περισσότεροι προγραμματιστές επιλέγουν να τρέχουν σεναρία ιστοσελίδων στον υπολογιστή του χρήστη μοιράζοντας έτσι το φόρτο εργασίας ανάμεσα στον server και στον client δεδομένου ότι η ισχύς και οι δυνατότητες των υπολογιστών από τη μεριά των χρηστών συνεχώς αυξάνονται. Η συγκεκριμένη γλώσσα σε μερικές περιπτώσεις καθίσταται απαραίτητη, για το λόγο ότι το γραφικό περιβάλλον χρήστη, το οποίο χειρίζεται η πλειοψηφία των συναρτήσεων, καθίσταται όλο και πιο σημαντικό, αφού σε πολλές περιπτώσεις μία ιστοσελίδα “κρίνεται” όχι μόνο μέσα από το περιεχόμενο αλλά και από την εμφάνιση.

3.4.6 jQuery

Το 99% του κώδικα της εφαρμογής στη διατριβή χρησιμοποιεί τη βιβλιοθήκη της Javascript, την jQuery. Πρόκειται για μία ελεύθερη, ανοικτού κώδικα γλώσσα που σχεδιάστηκε το 2006 για να απλοποιήσει τη γραφή σεναρίων στην πλευρά του client. Στατιστικά χρησιμοποιείται από το 55% των 10.000 ιστοσελίδων με τη μεγαλύτερη επισκεψιμότητα. Ένα από τα πλεονεκτήματά της είναι η ερμηνεία των σεναρίων το ίδιο από όλους τους περιηγητές, δίνοντας λύση στο πρόβλημα οι εντολές της Javascript (χωρίς τη βιβλιοθήκη jQuery) να ερμηνεύονται διαφορετικά. Η jQuery βιβλιοθήκη σταδιακά γίνεται πιο επιθυμητή από τους προγραμματιστές σε σχέση με την απλή Javascript για το λόγο ότι προσφέρει μεγαλύτερη ευελιξία και έλεγχο στα στοιχεία της HTML, των συμβάντων, τη δημιουργία animations και γενικότερα καλύτερη εμπειρία χρήσης και διαδραστικότητα. Η συγγραφή σεναρίων θεωρείται υψηλού επιπέδου και είναι περισσότερο κατανοητή από τον προγραμματιστή, απαιτεί λιγότερο κώδικα, ο οποίος είναι και ευκολότερα συντηρήσιμος είναι μερικά από τα πλεονεκτήματα της βιβλιοθήκης. Παρέχει δυνατότητες για επέκταση βιβλιοθηκών αλλά και για δημιουργία plugins (παρόμοια με τις έτοιμες κλάσεις που συναντώνται σε άλλες γλώσσες προγραμματισμού), τα οποία απλοποιούν ορισμένες επιλογές της Javascript αποκρύπτοντας το δυσνόητο κώδικα για τον προγραμματιστή.

3.4.7 Asynchronous JavaScript and XML (AJAX)

Το ακρωνύμιο AJAX είναι ένα σύνολο αλληλοσυνδεόμενων τεχνικών ανάπτυξης στο Διαδίκτυο που χρησιμοποιείται στην πλευρά του client για τη δημιουργία ασύγχρονων web εφαρμογών. Χάρη στο AJAX οι web εφαρμογές δύνανται να στείλουν και να λάβουν δεδομένα από ένα server ασύγχρονα (στο παρασκήνιο) χωρίς να επηρεάσουν την παρουσίαση και τη συμπεριφορά μίας ιστοσελίδας. Η πιο συχνή χρήση τους αφορά στην ανταλλαγή δεδομένων από τον client στο server και αντίστροφα χωρίς την απαίτηση για ανανέωση της ιστοσελίδας. Επιτρέπει τη διενέργεια αιτημάτων (requests) προς τον διακομιστή από τον περιηγητή ιστού στο παρασκήνιο. Θεωρείται μία σημαντική τεχνική που χρησιμοποιείται ευρέως από τα web προγράμματα: προσδίδει δυναμικότητα σε μια ιστοσελίδα, εξοικονομεί χρόνο και στο χρήστη και στον προγραμματιστή ανταλλάσσοντας μόνο τα χρήσιμα δεδομένα και αυξάνει την παραγωγικότητα του τελευταίου.

3.4.8 Το επίπεδο server

Το επίπεδο server περιγράφει τις λειτουργίες που εκτελούνται από έναν εξυπηρετητή σε ένα μοντέλο client-server στη διασύνδεση υπολογιστών. Τυπικά, ένας server είναι ένα λογισμικό, το οποίο προσπελάζεται από σταθμούς εργασίας χρηστών. Οι διάφορες λειτουργίες επιλέγονται να γίνουν στην πλευρά του server, επειδή απαιτούν πρόσβαση σε πληροφορία ή λειτουργικότητα, τα οποία είτε δεν είναι διαθέσιμα στην πλευρά του πελάτη είτε η πλευρά client δε θεωρείται αξιόπιστη στην προκειμένη περίπτωση. Πρόκειται για λειτουργίες που περιλαμβάνουν επεξεργασία και αποθήκευση δεδομένων από έναν client σε ένα server και διασύνδεση με βάση δεδομένων, τα αποτελέσματα των οποίων παρουσιάζονται στους χρήστες με τη μορφή ιστοσελίδων.

3.4.9 Hypertext Processor (PHP)

Μία γλώσσα προγραμματισμού server αποτελεί και η PHP με τα στατιστικά χρήσης της σε ιστοσελίδες να είναι πάνω από το 77% για το 2012. Είναι μία ανοικτού κώδικα, αντικειμενοστραφής, διαδικαστική, ανεξάρτητη από πλατφόρμα, γενικού σκοπού γλώσσα σεναρίων που σχεδιάστηκε για την ανάπτυξη διαδικτυακών και δυναμικών εφαρμογών. Ο κώδικάς της μεταγλωττίζεται σε έναν web server που διαθέτει μία υπό-μονάδα PHP επεξεργαστή, ο οποίος παράγει το τελικό αποτέλεσμα σε μία

ιστοσελίδα. Κύριες λειτουργίες της είναι η διαχείριση αρχείων και η εισαγωγή από το χρήστη, αλλά και η σύνδεση με βάσεις δεδομένων. Ενσωματώνεται σε έγγραφα HTML, είναι “ελαφριά” στη χρήση και σε πολλά σημεία γρηγορότερη σε σχέση με τους ανταγωνιστές της, όπως ASP και Coldfusion.

3.4.10 Database

Μία βάση δεδομένων είναι μία οργανωμένη συλλογή δεδομένων σε ψηφιακή μορφή. Τα δεδομένα είναι οργανωμένα σε ένα μοντέλο που αναπαριστά σχετικές όψεις της πραγματικότητας, π.χ. η διαθεσιμότητα των δωματίων σε ένα ξενοδοχείο, με τρόπο που υποστηρίζει επεξεργασίες που απαιτούν πληροφορίες, όπως η αναζήτηση ξενοδοχείων που έχουν διαθεσιμότητα. Ο όρος βάση δεδομένων εφαρμόζεται στα δεδομένα και τις δομές δεδομένων τους. Η χρήση τους είναι ευρέως διαδεδομένη σε σχεδόν κάθε τεχνολογία και όλες οι γλώσσες server τις υποστηρίζουν εγγενώς διαθέτοντας συναρτήσεις για διάφορα συστήματα βάσεων δεδομένων.

3.4.11 Σύστημα διαχείρισης βάσης δεδομένων (DBMS)

Πρόκειται για ένα πακέτο λογισμικού με προγράμματα που ελέγχουν τη δημιουργία, τη συντήρηση και τη χρήση βάσης δεδομένων. Δίνει τη δυνατότητα σε οργανισμούς να αναπτύσσουν με ευκολία βάσεις δεδομένων με διάφορες εφαρμογές και σε διαφορετικά προγράμματα χρηστών να έχουν ταυτόχρονη πρόσβαση στην ίδια βάση δεδομένων. Το DBMS χρησιμοποιεί πληθώρα από μοντέλα βάσεων δεδομένων, όπως το σχεσιακό (RDBMS) ή το αντικειμενοστραφές (OODBMS) για την καταλληλότερη περιγραφή των εφαρμογών που αναπτύσσουν. Τυπικά υποστηρίζει γλώσσες ερωτημάτων (SQL), οι οποίες είναι γλώσσες υψηλού επιπέδου που απλοποιούν αισθητά τη συγγραφή εφαρμογών σε βάσεις δεδομένων, αλλά και την οργάνωση όπως και την ανάκτηση και την παρουσίαση της πληροφορίας. Παρέχει δυνατότητες για έλεγχο πρόσβασης και ασφάλειας στα δεδομένα ενισχύοντας την ακεραιότητά τους διαχειρίζοντας τις ταυτόχρονες λειτουργίες και ανακτώντας τη ΒΔ ύστερα από σφάλματα στο λογισμικό.

3.4.12 Περιηγητές Ιστού (Web browsers)

Ένας περιηγητής ιστού είναι μία εφαρμογή για ανάκτηση, παρουσίαση και διανομή πληροφοριακών πηγών στο World Wide Web. Μία πληροφοριακή πηγή προσδιορίζεται από ένα καθολικό αναγνωριστικό πηγής ή τοποθεσίας (Uniform Resource Identifier - URL/URI), το οποίο μπορεί να είναι μία ιστοσελίδα, εικόνα, βίντεο ή οποιοδήποτε άλλο κομμάτι που περιέχει πληροφορία. Οι υπερσύνδεσμοι (hyperlinks) που εντοπίζονται σε αυτές τις πηγές επιτρέπουν στους χρήστες να περιηγηθούν σε άλλες τοποθεσίες και αυτό δείχνει το συσχετισμό των τοποθεσιών ή κόμβων μεταξύ τους. Παρόλο που οι περιηγητές (ιστού) προορίζονται κυρίως για το World Wide Web, μπορούν επίσης να χρησιμοποιηθούν για πρόσβαση σε πληροφορίες που παρέχονται από τους εξυπηρετητές σε ιδιωτικά δίκτυα (private networks) ή αρχείων σε συστήματα αρχείων (file systems). Ο περιηγητής ιστού είναι το λογισμικό που επεξεργάζεται και προβάλλει τις πληροφορίες που παρουσιάζονται στην πτυχιακή εργασία.

3.4.13 HTML5

Η HTML5 είναι μία γλώσσα σήμανσης που χρησιμοποιείται για προβολή πληροφορίας στο World Wide Web και είναι μία θεμελιώδης τεχνολογία του Διαδικτύου. Βρίσκεται στην πέμπτη του HTML προτύπου και θεωρείται υποψήφια για υιοθέτηση το World Wide Consortium (W3C). Οι κύριοι στόχοι της τεχνολογίας έχουν να κάνουν με τη βελτίωση της γλώσσας και για υποστήριξη των τελευταίων πολυμέσων, ενώ παράλληλα να διατηρεί την άνετη αλληλεπίδραση τόσο από ανθρώπους όσο και από υπολογιστές, όπως οι περιηγητές ιστού. Σχεδόν όλοι οι περιηγητές τον υποστηρίζουν, αλλά από συγκεκριμένες εκδόσεις και πάνω, ενώ υπάρχει ολοένα αυξανόμενη υποστήριξη της νέας τεχνολογίας. Ο λόγος αναφοράς της HTML5 τεχνολογίας είναι ότι η εφαρμογή που αναπτύχθηκε υποστηρίζεται μόνο από HTML5 browsers. Σε παλαιότερες εκδόσεις ενδέχεται το πρόγραμμα να λειτουργεί με προβλήματα.

3.4.14 MySQL

Η MySQL είναι ένα σχεσιακό σύστημα διαχείρισης βάσης δεδομένων (RDBMS) που “τρέχει” ως εξυπηρετητής παρέχοντας πρόσβαση σε πλειάδα χρηστών σε βάσεις δεδομένων. Είναι το πιο διαδεδομένο ανοικτού κώδικα σύστημα διαχείρισης βάσης δεδομένων και ανήκει στην οικογένεια δομημένων ερωτημάτων (SQL). Θεωρείται ως ένα αρκετά ισχυρό, γρήγορο, αξιόπιστο και ευέλικτο RDBMS, παρέχοντας υψηλή απόδοση με πολυνηματικές λειτουργίες. Χρησιμοποιείται για διασύνδεση με web εφαρμογές και προτιμάται για την ταχύτητα και την ασφάλειά του ικανό να ανταπεξέρχεται σε ερωτήματα με μεγάλα μεγέθη δεδομένων.

3.4.15 Open-source software (OSS)

Παραπάνω περιγράφηκαν με συντομία όλες οι βασικές τεχνολογίες που κάνει χρήση το λογισμικό που έχει αναπτυχθεί. Αυτές οι τεχνολογίες ανήκουν στην κατηγορία open-source λογισμικό. Αυτό είναι ένα πρόγραμμα υπολογιστή, το οποίο είναι διαθέσιμο και παρέχεται με άδεια (copyright) του κατόχου του λογισμικού και αφορά στα δικαιώματα της μελέτης, αλλαγής και διανομής λογισμικού σε οποιονδήποτε και για κάθε σκοπό. Συναντάται και ως μοντέλο παραγωγής λογισμικού, το οποίο αναπτύσσεται με δημόσιο και συνεργατικό τρόπο.

3.4.16 SQL Injection (Επίθεση SQL)

Η επίθεση τύπου SQL είναι μία τεχνική κατά την οποία κώδικας-σενάριο επιτίθεται σε μία βάση δεδομένων. Η χρήση της κατευθύνεται σε εφαρμογές που περιέχουν δεδομένα. Κακόβουλες προτάσεις (statements) SQL εισάγονται σε ένα πεδίο φόρμας HTML για εκτέλεση με σκοπό την προβολή των περιεχομένων από τη ΒΔ, τη διαγραφή πινάκων και άλλων επικίνδυνων ενεργειών. Αυτού του είδους η επίθεση εκμεταλλεύεται μία ευπάθεια σε θέμα ασφάλειας σε ένα λογισμικό, λόγω χάρη όταν η είσοδος ενός χρήστη σε ένα πεδίο κειμένου εσφαλμένα φιλτράρεται για κυριολεκτικές διαφυγόντες (escape) συμβολοσειρές που ενσωματώνονται (embedded) σε προτάσεις SQL ή όταν η είσοδος του χρήστη εκτελείται απρόβλεπτα ως πρόγραμμα.

3.5.1 Η βάση δεδομένων treesystem

Υπάρχουν 6 πίνακες στη τους οποίους απαιτεί η εφαρμογή για τη λειτουργία της. Ξεκινώντας με τον πρώτο πίνακα σημειώνουμε ότι είναι απαραίτητη η είσοδος στο σύστημα (login), έτσι ώστε αυτό να αναγνωρίσει την ομάδα χρηστών που ανήκει αυτός που πραγματοποιεί είσοδο. Συνεπώς, ο πίνακας users περιλαμβάνει 6 στήλες: τη στήλη id, τη στήλη user, τη στήλη pass, τη στήλη type, τη στήλη department και τη στήλη premise. Η στήλη user περιέχει το όνομα χρήστη και η στήλη pass περιέχει τον αντίστοιχο κωδικό. Η στήλη type πληροφορεί για την ομάδα χρήστη (user ή administrator), η στήλη department φανερώνει το τμήμα στο οποίο ανήκει ο χρήστης και η στήλη premise είναι η εγκατάσταση ή αλλιώς το υποκατάστημα της εταιρείας.

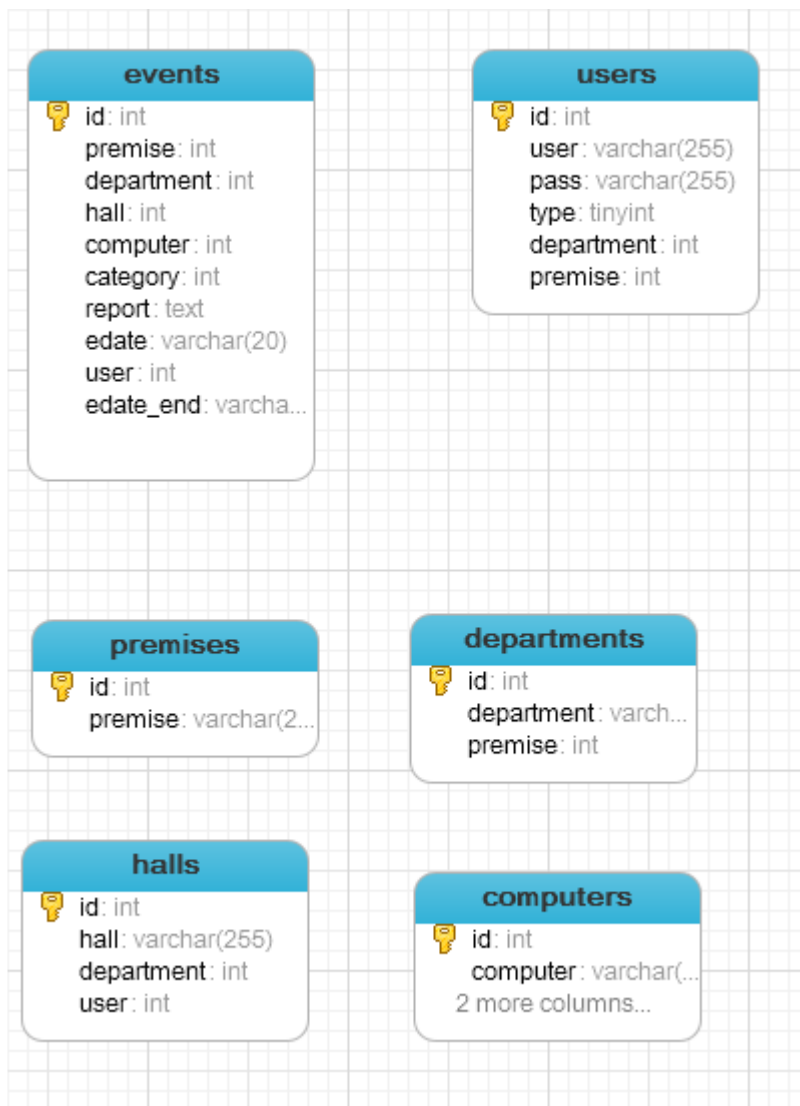
Ο πίνακας premises (εννοιολογικά είναι ο root κόμβος) περιέχει 2 στήλες: τη στήλη id και τη στήλη premise. Σε κάθε πίνακα που αναπαριστά τη δένδροειδή δομή, η στήλη (εδώ premise) μετά τη στήλη id περιέχει τα ονόματα των εγγραφών του πίνακα. Έτσι, η στήλη premise περιέχει τις εγκαταστάσεις (Αθηνών, Θεσ/νίκης και Πάτρας).

Ο πίνακας departments περιέχει τις στήλες id, department και premise. Η τελευταία στήλη φανερώνει ότι είναι πατέρας για τον οικείο πίνακα, δηλαδή υψηλότερος στην ιεραρχία. Αυτό ισχύει και για τους πίνακες halls και computers.

Ο πίνακας halls ακολουθεί ιεραρχικά και περιέχει τις στήλες id, hall, department και user. Δεν απαιτείται σχολιασμός για τις τρεις πρώτες στήλες, αλλά για τη στήλη user πρέπει να επισημανθεί ότι εννοιολογικά συνεχίζει τη δομή του δένδρου που συζητείται στην πτυχιακή εργασία, επεκτείνοντας το δένδρο οριζόντια. Κάθε χρήστης είναι υπεύθυνος για μία αίθουσα, αλλά το δένδρο δε συνεχίζει την πορεία του προς τα κάτω (ειδίκευση) από τον πίνακα των χρηστών (users), αλλά από τον πίνακα halls.

Σε κάθε αίθουσα ή Υ/Σ (hall) ανήκουν υπολογιστές (computers) και έτσι ο ομώνυμος πίνακας περιλαμβάνει τις στήλες id, computer, subsystems και hall. Η τρίτη στήλη περιλαμβάνει όλο το υλικό που είναι εγκατεστημένο ή συνδεδεμένο με τον υπολογιστή, ενώ η τελευταία στήλη στον πίνακα είναι ο πατέρας για το συγκεκριμένο κλαδί-πίνακα.

Ο πίνακας events ολοκληρώνει το μοντέλο της ΒΔ και περιέχει τις στήλες id, premise, department, hall, computer, category, report, edate, user και edate_end. Για κάθε συμβάν (event) καταγράφεται η εγκατάσταση, το τμήμα, η αίθουσα και ο υπολογιστής στον οποίο εντοπίστηκε. Η στήλη category δίνει το είδος του συμβάντος (συντήρηση ή βλάβη), η στήλη report είναι η αναφορά που καταχωρεί ο χρήστης, η στήλη edate είναι η ώρα που καταγράφη το συμβάν, η στήλη edate_end πληροφορεί πότε ολοκληρώθηκε η αποκατάσταση ή η συντήρηση της βλάβης ή της συντήρησης αντίστοιχα. Τέλος, η στήλη user δίνει το id του χρήστη που καταχώρησε το συμβάν. Σημειώνεται ότι όλες οι στήλες για τις οποίες συναντάται η αντίστοιχη στήλη στον πίνακα που αναφέρονται περιλαμβάνουν τα id των στηλών των τελευταίων. Σε όρους πληροφορικής και αρχείων, γίνεται προσπάθεια να αποθηκεύεται όσο το δυνατόν λιγότερη πληροφορία στη βάση δεδομένων.



Εικόνα 3.1: Πλήρες διάγραμμα

ER στη βάση δεδομένων

Στη συνέχεια θα παρουσιαστούν οι εγγραφές των πινάκων της βάσης δεδομένων. Αυτές καταχωρήθηκαν με τη βοήθεια ενός εργαλείου-προγράμματος, το Navicat. Τα δεδομένα της ΒΔ, δηλαδή οι εγγραφές οι οποίες τα περιέχουν τα παρέχει στο διακομιστή Ιστού κατόπιν αιτήματος του τελευταίου. Αφορά τις εγγραφές που έχουν δημιουργηθεί κατά την ανάπτυξη του προγράμματος. Οι παρακάτω πίνακες δύνανται να συμπληρωθούν με επιπλέον δεδομένα κατά τη χρήση της εφαρμογής και μάλιστα αρκετά μεγάλης κλίμακας και μεγέθους για το λόγο ότι έχει καταβληθεί προσπάθεια τα πεδία των πινάκων να είναι βελτιστοποιημένα για αποθήκευση των μεγαλύτερων σε μήκος δυνατών εγγραφών με δεδομένα που λαμβάνονται από το server.

id	computer	hall	subsystems
1	HY1	1	(Null)
2	HY2	1	(Null)
3	HY3	2	(Null)
4	HY4	6	(Null)
5	HY5	7	(Null)
6	HY6	1	(Null)
7	HY7	8	(Null)
8	HY8	6	(Null)

Εικόνα 3.2:

Ο πίνακας *computers*: περιλαμβάνει τους υπολογιστές που ανήκουν στα Υ/Σ

id	department	premise
5	Τμήμα 5	1
6	Τμήμα 1	1
7	Τμήμα 2	1
8	Τμήμα 3	1
9	Τμήμα 4	1
10	Τμήμα 6	2
11	Τμήμα 7	2
12	Τμήμα 8	2
13	Τμήμα 9	2
14	Τμήμα 10	3

Εικόνα 3.3:

Ο πίνακας *departments*: περιλαμβάνει τα τμήματα του οργανισμού

id	premise	department	hall	computer	category	report	edate	user	edate_end
1	1	5	1	2	1	συντήρηση1	2014-04-04T06:30:00	2	2014-04-04T07:30:00
2	1	5	1	2	1	συντήρηση3	2014-04-05T06:30:00	2	2014-04-05T07:30:00
3	1	5	1	2	1	συντήρηση2	2014-05-07T08:00:00	2	2014-05-07T09:00:00
4	1	5	1	2	1	test	2014-04-09T06:30:00	2	2014-04-09T07:30:00
50	1	5	1	1	2	χαλασμένο	2014-05-05T07:30:00	2	2014-05-05T09:30:00
80	1	5	2	1	1	test1	2014-04-27T19:37:27	2	2014-04-27T20:37:27
81	1	5	2	1	2	χάλασε	2014-05-09T19:10:38	2	2014-05-09T20:10:38

κόνα 3.4: Ο πίνακας events: περιλαμβάνει τα συμβάντα που καταγράφονται

id	hall	department	user
1	Υ/Σ1	5	2
2	Υ/Σ2	5	2
3	Υ/Σ3	5	2
4	Υ/Σ4	5	2
5	Υ/Σ5	5	2
6	Υ/Σ6	6	1
7	Υ/Σ7	6	1
8	Υ/Σ8	2	3

όνα 3.5: Ο πίνακας halls: περιλαμβάνει τα υπολογιστικά συστήματα

Εικ

id	premise
1	ΕΓΚΑΤΑΣΤΑΣΗ ΑΘΗΝΩΝ
2	ΕΓΚΑΤΑΣΤΑΣΗ ΘΕΣ/ΝΙΚΗΣ
3	ΕΓΚΑΤΑΣΤΑΣΗ ΠΑΤΡΩΝ

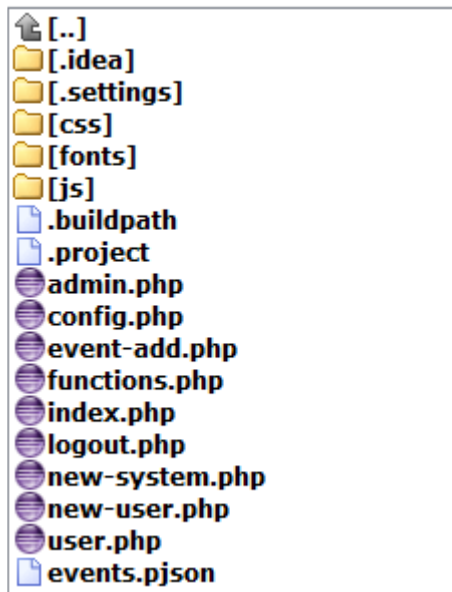
Εικ

όνα 3.6: Ο πίνακας premises: περιλαμβάνει τις εγκαταστάσεις της επιχείρησης

id	user	pass	type	department	premise
1	user	pass	1	1	1
2	user2	pass2	1	5	1
3	user3	pass3	1	5	2
4	user4	pass	1	5	3
5	admin	password	2	1	1
6	admin2	password2	2	1	2

Εικόνα 3.7: Ο πίνακας users: περιλαμβάνει όλους τους χρήστες που χρησιμοποιούν το σύστημα.

3.5.2 Δομή αρχείων και φακέλων



εφαρμογής-ιστοσελίδας

Εικόνα 3.8: Η αρχική διαδρομή αρχείων και φακέλων της

Το λογισμικό που δημιουργήθηκε στα πλαίσια της πτυχιακής εργασία δοκιμάστηκε σε προσωπικό εικονικό server (localhost). Παραπάνω είναι η διαδρομή root από την οποία ξεκινάνε όλες οι αποκρίσεις (HTTP responses) από το διακομιστή. Οι φάκελοι και τα αρχεία των οποίων το όνομα ξεκινάει με τελεία (π.χ. .idea) είναι τα αρχεία που εισάγει το Eclipse IDE στο φάκελο του project και δε χρησιμοποιούνται κατά την εκτέλεση της εφαρμογής.

Κάθε ένα PHP script έχει μία συγκεκριμένη λειτουργία επιστρέφοντας μία απάντηση στον πελάτη (client). Ορισμένα αρχεία είναι βοηθητικά (helper files) και είτε καλούνται από διαφορετικά PHP scripts είτε χρησιμοποιούνται για ασύγχρονη απάντηση (AJAX) από το server. Εκτός από τα PHP αρχεία της εφαρμογής θα συζητηθούν και τα Javascript scripts (βρίσκονται στο φάκελο js), τα οποία διευκολύνουν, βελτιώνουν και κάνουν πιο διαδραστική την εμπειρία χρήστη στον πελάτη-χρήστη. Οι φάκελοι css και fonts συμβάλλουν στη μορφοποίηση των σελίδων και δεν αποτελούν σημαντικό αντικείμενο συζήτησης.

3.5.3 Είσοδος στο σύστημα

Η είσοδος (login) στην εφαρμογή γίνεται μέσω της σελίδας index.php. Η συγκεκριμένη σελίδα είναι η πρώτη σελίδα την οποία αναζητά ο server στη διαδικασία της απόκρισης από αυτόν. Η διαδικασία εισόδου στο σύστημα “επικυρώνεται” χρησιμοποιώντας PHP sessions. Αυτά είναι cookies που αποθηκεύονται στο επίπεδο πελάτη (client layer) και έχουν ισχύ για μία συνεδρία ή αλλιώς για όσο διάστημα είναι ενεργός ο περιηγητής του χρήστη. Μετά το πέρας κάθε συνεδρίας τα αρχεία-cookies διαγράφονται από το προσωρινό φάκελο χρήστη.

Ο χρήστης παρέχει το username (και το password) του κατά την είσοδο στο σύστημα και αυτό ανακατευθύνει (redirects ή header στην HTTP ορολογία) ανάλογα την ομάδα χρήστη με την οποία είναι συνδεδεμένος στη βάση δεδομένων. Για κάθε ομάδα γίνεται ανακατεύθυνση σε διαφορετική PHP σελίδα. Ο διαχειριστής θα εισέλθει στη σελίδα `admin.php`, ενώ ο χρήστης θα εισέλθει στην `user.php`.

Το αρχείο `index.php` στο ξεκίνημα διαδικασία εισόδου θα ελέγξει εάν ήδη υπάρχει ενεργή συνεδρία (session) για το χρήστη που επιχειρεί να εισέλθει. Εάν υπάρχει, τότε θα ανακατευθύνει άμεσα το χρήστη στην αντίστοιχη ιστοσελίδα. Ειδικότερα, αφού πρώτα γίνει έλεγχος για συμπληρωμένο username και password και ότι έχει γίνει υποβολής της φόρμας εισόδου, τότε γίνεται κλήση (include) του script `config.php`, το οποίο παρέχει στη σελίδα τα διαπιστευτήρια εισόδου (connection credentials) για τη σύνδεση με τη MySQL. Ακολούθως, πραγματοποιείται ερώτημα (query) στη ΒΔ, το οποίο επιλέγει (select) όλες τις στήλες από τον πίνακα `users` δίνοντας ως παραμέτρους το username και το password του χρήστη. Σε περίπτωση επιστροφής object από τη MySQL στο παραπάνω ερώτημα, δηλαδή επιτυχίας εισόδου, γίνεται ορισμός (assign) των επόμενων συνεδριών:

```
$_SESSION['type'], $_SESSION['userid'],  
$_SESSION['departmentid']$_SESSION['user'], $_SESSION['premiseid'].
```

Η συνεδρία `type` πληροφορεί για την ομάδα χρήστη που επιχειρεί είσοδο στο σύστημα. Η συνεδρία `userid` λαμβάνει το `id` του χρήστη που είναι αποθηκευμένο στη ΒΔ. Η συνεδρία `departmentid` φανερώνει το `id` του τμήματος στο οποίο ανήκει ο χρήστης και η συνεδρία `user` κρατά το όνομα χρήστη, με το οποίο πραγματοποιήσε είσοδο. Αυτά τα sessions είναι σημαντικά και θα χρησιμοποιηθούν από άλλα PHP scripts στη συνέχεια εκτέλεσης του προγράμματος.

Τελευταίο βήμα είναι η ανακατεύθυνση του χρήστη στην ανάλογη σελίδα. Γίνεται πάντα πρόνοια για κλείσιμο της MySQL σύνδεσης έτσι ώστε να μην είναι πλέον ενεργή. Σημειώνεται ότι όλες οι συνδέσεις και τα ερωτήματα στη ΒΔ γίνεται μέσω των `mysqli_` συναρτήσεων (internal functions) της PHP. Οι συναρτήσεις αυτές παρέχουν μεταξύ άλλων αυξημένη ασφάλεια σε σχέση με τις αντίστοιχες `mysql` συναρτήσεις, ενώ παράλληλα επιτυγχάνεται ένα optimization στον τρόπο με τον οποίο η MySQL επεξεργάζεται και εκτελεί τα ερωτήματα.

Πρέπει να σημειωθεί ότι η είσοδος στο σύστημα αλλά και η περιήγηση στο πρόγραμμα και επικοινωνία με τη βάση δεδομένων ακολουθούν βασικές πρακτικές ασφαλείας. Η είσοδος του χρήστη, για παράδειγμα, σε ένα πεδίο κειμένου φόρμας φιλτράρεται από το σύστημα και απομακρύνει όποιους χαρακτήρες που πιθανόν να βρουν τρύπες ασφαλείας (security holes) στο σύστημα και να προκαλέσουν απρόβλεπτα επικίνδυνα αποτελέσματα.

Επιπρόσθετα, έχει καταβληθεί προσπάθεια έτσι ώστε η φόρτωση των σελίδων της εφαρμογής να επιτυγχάνεται ταχύτερα αλλά και η εκτέλεση σεναρίων στο επίπεδο πελάτη να είναι το κατά δύναμιν ελάχιστη και αριστοποιημένη. Έτσι, τα αρχεία της εφαρμογής είναι όσο το δυνατόν μικρά σε μέγεθος, από τα οποία γίνεται κλήση σε γνωστές ανοιχτού κώδικα βιβλιοθήκες, οι οποίες είναι επαληθευμένο ότι δουλεύουν σωστά ακολουθώντας αυστηρούς κανόνες και πρότυπα που υπαγορεύουν τη σωστή ανάπτυξη σεναρίων σε ιστοσελίδες.

3.5.4 Η σελίδα χρήστη (user)

Η σελίδα `user.php` είναι η σελίδα που εκτελεί την καθημερινή του εργασία ο χρήστης (user). Η αρχιτεκτονική και η λειτουργία της HTML σελίδας υπαγορεύεται από τη jQuery και από το Bootstrap. Η jQuery βιβλιοθήκη έχει εξηγηθεί παραπάνω. Το Bootstrap είναι επίσης ανοιχτού κώδικα και ελεύθερη βιβλιοθήκη. Το αρχείο `user.js` που βρίσκεται στο φάκελο `js` είναι αρχείο που γράφτηκε από τον προγραμματιστή και κάνει χρήση και των δύο προαναφερθέντων βιβλιοθηκών-αρχείων.

Το Bootstrap (συνοδεύεται με MIT άδεια και το copyright έχει γίνει από το Twitter), μία open-source μηχανή για μορφοποίηση, ταχύτερης απόκρισης, βελτίωση της HTML σελίδας, ενώ παράλληλα επιτρέπει την προβολή της ιστοσελίδας τόσο σε μικρές (mobile) όσο και πολύ μεγάλες (television) αναλύσεις. Το όφελος στον προγραμματιστή είναι αρκετά σημαντικό, διότι του παρέχει μία άνετη και ανέξοδη γραφή HTML, CSS και Javascript σεναρίων στο επίπεδο του πελάτη. Ίσως το πιο σημαντικό πλεονέκτημα χρήσης του Bootstrap είναι η συγγραφή cross-browser applications, δηλαδή εφαρμογών που εκτελούνται ισοδύναμα σε όλους μοντέρνους περιηγητές ιστού. Η βιβλιοθήκη της εξεταζόμενης σελίδας είναι το αρχείο bootstrap.min.js και το CSS αρχείο το bootstrap.min.css. Και τα δύο αρχεία καλούνται από τους φακέλους js και css αντίστοιχα.

Στα CSS αρχεία που καλούνται συγκαταλέγονται και τα user.css (είναι το αρχείο που έχει γραφεί από τον προγραμματιστή για τη μορφοποίηση της σελίδας χρήστη) και το fullcalendar.css (χρησιμοποιείται για τη μορφοποίηση του αναλυτικού ημερολογίου).

Η σελίδα του χρήστη διαιρείται σε 4 νοητές περιοχές. Η πρώτη περιοχή που τοποθετείται πάνω δεξιά φανερώνει την εγκατάσταση και το τμήμα στο οποίο ανήκει ο χρήστης. Επιπλέον, δείχνει το username του και το σύνδεσμο Αποσύνδεση για έξοδο του χρήστη από το σύστημα. Αυτές τις πληροφορίες το πρόγραμμα τις λαμβάνει μέσω των μεταβλητών-συνεδριών που έχουν τεθεί στην αρχική σελίδα (index.php).

Σειρά έχει η δεύτερη περιοχή που καταλαμβάνει το αριστερό μέρος της ιστοσελίδας. Εδώ έχει τοποθετηθεί το ημερολόγιο συμβάντων. Ο προγραμματισμός του ημερολογίου έχει γίνει με τη βοήθεια του FullCalendar το οποίο παρέχεται ελεύθερα με MIT άδεια και πρόκειται για ένα jQuery πρόσθετο (plugin). Χρησιμοποιεί τεχνολογία AJAX για να φορτώσει δεδομένα-συμβάντα για κάθε μήνα ξεχωριστά και μπορεί εύκολα να προσαρμοστεί για χρήση σε ιστοσελίδες. Παρέχει στον προγραμματιστή πληθώρα μεθόδων (functions) και ιδιοτήτων (properties) για ίδια χρήση και καταργεί την εκ νέου ανάπτυξη ενός ημερολογίου συμβάντων. Πιο τεχνικά, δημιουργεί δυναμικά στην οθόνη ένα δύο-διαστάσεων τετραγωνικό ημερολόγιο που απεικονίζει τις ημέρες του μήνα σε αντίστοιχα μεμονωμένα τετράγωνα. Προβλέπεται η πλοήγηση κατά μήνα μπροστά ή πίσω και η εμφάνιση του ημερολογίου είτε κατά μήνα είτε κατά εβδομάδα.

Το ημερολόγιο δέχεται δεδομένα σε μορφή JSON (Javascript Object Notation). Ένα JSON είναι ένα αρχείο (στην εφαρμογή καλείται events.pjson) και περιέχει μία συμβολοσειρά, η οποία μετατρέπεται (parsed) από το πρόγραμμα σε Javascript object. Αυτό περιέχει την κατηγορία του συμβάντος, την αρχή και το τέλος του ωρολογιακά και τη χρωματική του ταυτότητα. Παρακάτω παρατίθεται μία γραμμή που περιέχει ένα συμβάν:

```
{ "title": "Συντήρηση", "start": "2014-04-04T06:30:00", "end": "2014-04-04T07:30:00", "allDay": "", "backgroundColor": "green" }
```

Ολόκληρο το αρχείο που περιέχει όλα τα συμβάντα που έχουν μέχρι στιγμής καταγραφεί στο σύστημα είναι το παρακάτω.

```
[ { "title": "Συντήρηση", "start": "2014-04-04T06:30:00", "end": "2014-04-04T07:30:00", "allDay": "", "backgroundColor": "green" }, { "title": "Συντήρηση", "start": "2014-04-05T06:30:00", "end": "2014-04-05T07:30:00", "allDay": "", "backgroundColor": "green" }, { "title": "Συντήρηση", "start": "2014-05-07T08:00:00", "end": "2014-05-07T09:00:00", "allDay": "", "backgroundColor": "green" }, { "title": "Συντήρηση", "start": "2014-04-09T06:30:00", "end": "2014-04-09T07:30:00", "allDay": "", "backgroundColor": "green" }, { "title": "Βλάβη", "start": "2014-05-05T07:30:00", "end": "2014-05-05T09:30:00", "allDay": "", "backgroundColor": "crimson" } ]
```

Κάθε συμβάν-γραμμή χωρίζεται με άγκιστρα ({,}). Κατά το ξεκίνημα της εφαρμογής το αρχείο αυτό διαβάζεται και χρησιμοποιείται από το FullCalendar. Η ανάγνωση του αρχείου πραγματοποιείται ασύγχρονα μέσω τεχνολογίας AJAX. Το αίτημα της ανάγνωσης JSON έπεται της δημιουργίας του JSON αρχείου. Αυτό επιτυγχάνεται με τη βοήθεια της PHP συνάρτησης `createJSON`, που βρίσκεται στο βοηθητικό αρχείο `functions.php`. Η συνάρτηση λαμβάνει ως παραμέτρους το `mysql connection` και το `id` του χρήστη, που έχει αποθηκευτεί προηγουμένως σε συνεδρία. Όταν γίνει κλήση της συνάρτησης από τη σελίδα του χρήστη, τότε το ερώτημα που γίνεται στη ΒΔ, αφορά κυρίως τον πίνακα `events` επιλέγοντας τις στήλες που χρειάζεται να δείξει το ημερολόγιο. Το ερώτημα έχει ως παράμετρο το `id` του χρήστη και επιλέγει τα συμβάντα που σχετίζονται με το τμήμα του χρήστη. Η συμβολοσειρά που θα μετατραπεί σε JSON object σχηματίζεται δυναμικά σε ένα βρόγχο όπου κάθε μία επανάληψη περιλαμβάνει ένα συμβάν που έχει επιστρέψει το ερώτημα στη ΒΔ. Παράλληλα με την εκτέλεση του βρόγχου εκτυπώνεται στο HTML αλλά δεν είναι ορατή στο χρήστη μία συμβολοσειρά που θα χρησιμοποιηθεί από τη Javascript την ημερομηνία του συμβάντος, το Υ/Σ και τον υπολογιστή που αφορά καθώς και τη σχετική αναφορά. Τελικά, το αρχείο `events.pjson` δημιουργείται ή επικαλύπτεται (`overwrite`) εάν υπάρχει ήδη στο server με την εντολή `file_put_contents`.

Η συνάρτηση Javascript `readJSON` κάνει ανάγνωση του αρχείου `events.pjson` και καλεί με τη σειρά της τη μέθοδο `calendar` και της παρέχει η πρώτη μία παράμετρο με το JSON object. Η δεύτερη συνάρτηση αρχικοποιεί το ημερολόγιο περνώντας τις παραμέτρους που χρειάζεται και καλώντας τη συνάρτηση `fullCalendar` από το αρχείο `fullcalendar.js`.

Η λειτουργία του ημερολογίου ολοκληρώνεται με την εμφάνιση της αναφοράς καταγραφής ακριβώς κάτω από το ημερολόγιο. Η αναφορά έχει το δικό της χώρο και αναπαριστάται με ένα ορθογώνιο παραλληλόγραμμο με μέγεθος ίσο με το ημερολόγιο και δίνει πληροφορίες για το Υ/Σ και τον υπολογιστή για το οποίο κατεγράφη η αναφορά. Ο χώρος της αναφοράς είναι κενός για ημέρα για την οποία δεν έχει καταχωρηθεί συμβάν, ενώ “γεμίζει” (`populates`) όταν ο χρήστης πατήσει κατά τη χρήση του προγράμματος σε μία μέρα με καταχωρηθέν συμβάν δείχνοντας την αντίστοιχη πληροφορία.

Σημαντικό χαρακτηριστικό του ημερολογίου είναι ότι όταν ο χρήστης επιλέξει την εμφάνισή ανά εβδομάδα βλέπει ακόμη τη διάρκεια του συμβάντος που συμπεραίνεται από το ύψος του ορθογωνίου παραλληλογράμμου που αντιπροσωπεύει κάθε συμβάν. Υπεύθυνη για αυτή τη λειτουργία είναι η συνάρτηση Javascript `showReport`, κατά τη λειτουργία της οποίας όταν γίνει “κλικ” σε μία ημέρα, η μέθοδος τότε αναζητά τη συμβολοσειρά που έχει εκτυπωθεί νωρίτερα χάρη στη συνάρτηση PHP `createJSON`, διαβάζει την ημερομηνία και τη συσχετίζει-ισοδυναμεί με την ημερομηνία που αναπαριστά το FullCalendar για κάθε μεμονωμένη ημερομηνία για την οποία έχει καταχωρηθεί συμβάν. Αυτή η συνάρτηση δέχεται μία παράμετρο η οποία πληροφορεί για την περιοχή στην οποία πάτησε ο χρήστης με το ποντίκι. Έτσι, η περιοχή εμφάνισης της αναφοράς θα έχει τώρα την αναφορά, η οποία όπως αναφέρθηκε έχει προέλθει από την αντίστοιχη κρυφή εκτυπωθείσα αναφορά.

Συνεχίζουμε με την τρίτη νοητή περιοχή, την καταγραφή των συμβάντων από το χρήστη. Αυτή η περιοχή είναι το πρώτο tab του δεξιού μέρος, όπως συζητήθηκε σε προηγούμενα κεφάλαια. Είναι ορατή στο ξεκίνημα της εφαρμογής χρήστη, ενώ το ίδιο χώρο “καταλαμβάνει” και η τέταρτη περιοχή, αλλά προσωρινά δεν είναι ορατή από το χρήστη μέχρι να το επιλέξει. Η τρίτη περιοχή μεταφέρει στο χρήστη την πληροφορία της αίθουσας-Υ/Σ για την οποία είναι υπεύθυνος. Από κάτω αναφέρονται όλοι οι υπολογιστές που είναι τοποθετημένοι σε αυτή την αίθουσα σε διακριτά κουτιά-κουμπιά. Μέσω ενός ερωτήματος στη ΒΔ επιλέγονται τα Υ/Σ και οι υπολογιστές τους οποίους επιβλέπει ο χρήστης και αποθηκεύονται σε PHP μήτρες-μεταβλητές (`arrays`), οι οποίες με τη σειρά τους εκτυπώνονται στο HTML έγγραφο.

Όταν ο χρήστης πατήσει ένα από τα κουμπιά-υπολογιστές θα κυλήσει (`animate`) προς τα κάτω μία νέα περιοχή-φόρμα, προσφέροντας μία βασική διαδραστική εμπειρία στο χρήστη (Javascript συνάρτηση `doReport`) εντός της οποίας ο χρήστης επιλέγει το είδος του συμβάντος που επιθυμεί να καταχωρίσει και τη διάρκεια αποκατάστασης ή βλάβης. Η αναφορά πραγματοποιείται μέσα από μία

περιοχή κειμένου και αφορά τον επιλεγθέντα υπολογιστή που έγινε σε νωρίτερο στάδιο. Τη φόρμα ολοκληρώνει το κουμπί Καταγραφή και θα εκτελέσει τη μέθοδο `submit`. Η διαδικασία υποβολής μίας HTML φόρμας είναι η ανάγνωση των τιμών των στοιχείων της (για παράδειγμα πεδία κειμένου, dropdown menus κλπ.) και θα σταλούν ως παράμετροι ασύγχρονα ως αίτημα μέσω AJAX στο διακομιστή. Το αρχείο PHP υπεύθυνο για την επεξεργασία του παραπάνω αιτήματος είναι το `event-add.php`. Μετά το πέρας της περιγραφείσας διαδικασίας, και εφόσον η καταγραφή υπήρξε επιτυχής, εμφανίζεται ένα κινούμενο (animated) μήνυμα στο χρήστη πληροφορώντας τον/την για την επιτυχία της καταγραφής. Αμέσως μετά, η σελίδα κάνει επαναφόρτωση έτσι ώστε να δείξει τις αλλαγές στο ημερολόγιο.

Το αρχείο `event-add.php` που λαμβάνει το αίτημα AJAX σύμφωνα με την παραπάνω διαδικασία, συνοπτικά δέχεται τις παραμέτρους του αιτήματος και τις αποθηκεύει στις αντίστοιχες μεταβλητές, μορφοποιεί την τρέχουσα ημερομηνία και ώρα στην επιθυμητή μορφή που χρειάζεται το FullCalendar και υπολογίζει την ώρα που έλαβε τέλος η αποκατάσταση ή συντήρηση με βάση τη διάρκεια που όρισε προηγουμένα στη φόρμα καταγραφής. Έχοντας διεκπεραιώσει τους άνω υπολογισμούς, το πρόγραμμα είναι έτοιμο να εκτελέσει ερώτημα εισαγωγής (insert) στη ΒΔ παρέχοντας τις παραμέτρους που υπολογίστηκαν νωρίτερα. Μία νέα εγγραφή προστίθεται στη βάση δεδομένων και περιλαμβάνει αναλυτικά τις πληροφορίες του συμβάντος.

Καταλήγουμε στην τέταρτη νοητή περιοχή, η οποία είναι ορατή όταν επιλέξει το αντίστοιχο tab ο χρήστης. Σε αυτήν την περίπτωση το τρίτο tab θα εξαφανιστεί χάριν την ορατότητας του τέταρτου. Η συγκεκριμένη περιοχή δείχνει στο χρήστη ένα πτυσσόμενο μενού (dropdown menu) από το οποίο μπορεί να επιλέξει να εμφανίσει διαγράμματα ή πίνακες για βλάβες και συντηρήσεις που καταγράφηκαν στο τμήμα του χρήστη.

Η εκτύπωση της περιοχής στο HTML έγγραφο γίνεται με σε συνεργασία της PHP και της Javascript. Πρώτα καλείται η PHP συνάρτηση `statsSystems` μέσα από το βοηθητικό αρχείο `functions.php`. Αυτή δέχεται ως παραμέτρους το `mysqli object connection`, το `id` του `department` στο οποίο ανήκει ο χρήστης, και μία παράμετρο που πληροφορεί εάν ο χρήστης είναι `user` ή `administrator`. Η τελευταία παράμετρος χρησιμοποιείται για να γίνει διαφορετική κλήση του ερωτήματος στη ΒΔ ανάλογα με την ομάδα χρήστη. Το ερώτημα αυτό επιλέγει όλα τα συμβάντα αναφορικά με την ομάδα χρήστη και τα εκτυπώνει στο HTML δίχως αυτά να είναι ορατά. Σχηματίζεται μία ιεραρχία HTML κόμβων, οι οποίοι περιέχουν ανά μήνα πόσες βλάβες και πόσες συντηρήσεις πραγματοποιήθηκαν για ποιο Υ/Σ. Στη Javascript αυτή η ιεραρχία αναλύεται και μετατρέπεται σε JSON object. Τα αντικείμενα (objects) χρησιμοποιούνται για τη δημιουργία κατανοητών και εύληπτων πινάκων και διαγραμμάτων από το χρήστη.

Η διαγραμματική απεικόνιση γίνεται με τη βοήθεια της ανοιχτού κώδικα βιβλιοθήκης Highcharts. Αυτή προσφέρει μεγάλη ποικιλία από HTML διαγράμματα, ενώ η ενσωμάτωσή τους στην ιστοσελίδα έρχεται χωρίς κανένα κόστος από μέρος του προγραμματιστή. Το `highcharts.js` απαιτεί τα δεδομένα που θα εμφανίσει στο διάγραμμα να είναι JSON object. Κάθε στοιχείο του object περιλαμβάνει μία ετικέτα, πχ. Υ/Σ 1 και επιπλέον 12 τιμές ανά μήνα που αφορούν για παράδειγμα τις βλάβες μέσα σε ένα χρόνο. Η συνάρτηση που επιτυγχάνει αυτή τη λειτουργία είναι η `renderHighCharts` που δέχεται μία παράμετρο που πληροφορεί για την κατηγορία του συμβάντος που ο χρήστης επιθυμεί να εμφανίσει.

Η εμφάνιση των συμβάντων σε μορφή πίνακα ολοκληρώνει τη στατιστική ανάλυση των συμβάντων που δείχνει μία πρωταρχική αλλά πλήρως κατανοητή πληροφορία σχετικά με το πόσα και ποια Υ/Σ εμφάνισαν βλάβη ή πραγματοποιήθηκε συντήρηση για κάθε μήνα. Η συνάρτηση `renderTable` είναι υπεύθυνη για την παραπάνω λειτουργία και δέχεται μία παράμετρο που αναφέρει την κατηγορία του συμβάντος που ο χρήστης επιθυμεί να εμφανίσει.

3.5.5 Η σελίδα διαχειριστή (administrator)

Η σχεδίαση της σελίδας `admin.php` βασίστηκε πάνω στα πρότυπα και `scripts` παρόμοια με τη σελίδα χρήστη. Η σελίδα διαχειριστή διαιρείται σε 5 νοητές περιοχές. Η πρώτη πληροφορεί για το `username` του διαχειριστή, την εγκατάσταση και το τμήμα στο οποίο ανήκει καθώς και το σύνδεσμο Αποσύνδεση. Η δεύτερη περιλαμβάνει το ημερολόγιο που δείχνει τα συμβάντα σε όλο τον όμιλο εταιρειών.

Η τρίτη περιοχή διαιρείται σε τρία `tabs`. Έχοντας εξηγήσει τη λειτουργία των `tabs`, το πρώτο επιτρέπει στο διαχειριστή να προσθέσει νέο Υ/Σ στη ΒΔ ή να εγκαταστήσει νέους υπολογιστές σε ήδη υπάρχον Υ/Σ. Αρχικά ο διαχειριστής μέσα από ένα πτυσσόμενο μενού θα επιλέξει είτε Νέο Υ/Σ είτε υπάρχον Υ/Σ. Τα υπάρχοντα Υ/Σ έχουν προέλθει κάνοντας να ερωτήματα στη ΒΔ για τα Υ/Σ που ανήκουν στο τμήμα στο οποίο υπάγεται ο διαχειριστής. Η ίδια διαδικασία έχει ακολουθηθεί και για το πτυσσόμενο μενού των χρηστών, αφού για κάθε νέο υπολογιστικό σύστημα που προστίθεται στη βάση ορίζεται και ένας χρήστης υπεύθυνος γι αυτό. Είτε πρόκειται για νέο είτε για υπάρχον Υ/Σ εγκαθίστανται υπολογιστές σε αυτά. Έτσι προκύπτει η ανάγκη εισαγωγής ή επέκτασης Υ/Σ.

Για κάθε υπολογιστή ο διαχειριστής μπορεί να επιλέξει τα εξαρτήματα και περιφερειακά που είναι εγκατεστημένα ή συνδεδεμένα αντίστοιχα. Ορισμένοι υπολογιστές ενδέχεται να είναι ολοκληρωμένοι και να απαιτείται από το διαχειριστή να εισάγει όλα τα υλικά από τη λίστα, ενώ άλλοι υπολογιστές να μη διαθέτουν όλα τα υλικά κι έτσι ο διαχειριστής πρέπει να επιλέξει ανάλογα. Οι επιλογές του διαχειριστή όσον αφορά τα κομμάτια του υπολογιστή γίνεται μέσω μίας πολλαπλής επιλογής λίστα (`multi dropdown list`) και επιτυγχάνεται μέσω ενός `jQuery plugin: choses.jQuery.min.js`. Τα `jQuery plugins` είναι Javascript `scripts`, τα οποία διευκολύνουν την ανάπτυξη προγραμμάτων στο HTML document και τα οποία απαιτούν την ύπαρξη της `jQuery` βιβλιοθήκης στο HTML έγγραφο.

Έχοντας κάνει ο διαχειριστής όλες τις επιλογές που απαιτούνται για την προσθήκη ή επέκταση Υ/Σ μπορεί να πατήσει το κουμπί Εισαγωγή και αυτή η κίνηση θα στείλει νέο αίτημα στο `server` και θα ζητήσει το αρχείο `new-system.php`. Πρέπει να αναφερθεί ότι γίνεται κάποια επεξεργασία από τη μεριά της Javascript έτσι ώστε η νέα σελίδα `new-system.php` που θα αναλάβει την επεξεργασία της φόρμας που υποβάλλεται να λάβει με συγκεκριμένο τρόπο τιμές των στοιχείων της.

Ο κύριος έλεγχος στο νέο PHP αρχείο γίνεται για να διαπιστωθεί εάν ο διαχειριστής κάνει προσθήκη ή επέκταση Υ/Σ. Στην περίπτωση που επιλέξει να προσθέσει Υ/Σ, τότε γίνεται καταμέτρηση των ήδη υπαρχόντων Υ/Σ και υπολογιστών και προστίθεται μία νέα εγγραφή στον πίνακα `halls` και μία νέα εγγραφή στον πίνακα `computers`, όπου γίνεται και συσχέτιση του υπολογιστή με το Υ/Σ. Σε αντίθετη περίπτωση προστίθεται μία νέα εγγραφή στον πίνακα `computers` και συσχετίζεται με τον Υ/Σ που επέλεξε ο διαχειριστής. Μετά το τέλος της παραπάνω διαδικασίας γίνεται ανακατεύθυνση (`redirect`) στη σελίδα `admin.php` πληροφορώντας με ένα κινούμενο μήνυμα το διαχειριστή για την επιτυχία υποβολής της φόρμας.

Στα καθήκοντα του διαχειριστή περιλαμβάνεται και η εισαγωγή νέου χρήστη (`user`) στο σύστημα. Παράδειγμα αποτελεί η νέα πρόσληψη ενός χρήστη-εργαζομένου. Ο διαχειριστής καλείται να εισάγει το `username` και το `password` του χρήστη και να επιλέξει το τμήμα στο οποίο θα ανήκει και βρίσκεται στην ίδια εγκατάσταση με το διαχειριστή μέσα από μία λίστα (`dropdown menu`). Σημειώνεται ότι η λίστα έχει δημιουργηθεί με δυναμικό τρόπο έχοντας εκτελέσει πρώτα ένα ερώτημα στη ΒΔ επιλέγοντας τα τμήματα που υπάρχουν στην ίδια εγκατάσταση με το διαχειριστή.

Εδώ θα κληθεί το `script new-user.php` όταν ο διαχειριστής κάνει υποβολή της άνω φόρμας. Το σενάριο αυτό επιτελεί δύο βασικές λειτουργίες. Ελέγχει πρώτα εάν υπάρχει ήδη χρήστης στο σύστημα με το `username` που έχει δώσει ο χρήστης. Σε αυτήν την περίπτωση τότε ανακατευθύνει το διαχειριστή στη σελίδα υποβολής της φόρμας ενημερώνοντας με αντίστοιχο μήνυμα. Σε περίπτωση που ο πρώτος έλεγχος είναι ψευδής προστίθεται μία νέα εγγραφή στη ΒΔ με τα στοιχεία που έδωσε ο διαχειριστής. Γίνεται ανακατεύθυνση στη σελίδα `admin.php` και ενημερώνει το διαχειριστή για την επιτυχία εισαγωγής χρήστη στο σύστημα.

Η τελευταία νοητή περιοχή που προβάλλεται στο διαχειριστή είναι τα στατιστικά. Η λειτουργία τους είναι παρόμοια με αυτήν των στατιστικών που προβάλλονται στο χρήστη (user) με τη διαφορά ότι ο διαχειριστής βλέπει και στατιστικά για τα συμβάντα που έχουν λάβει χώρα ανά τμήμα. Επιτρέπει, δηλαδή, στο διοικητικό προσωπικό να λάβει αποφάσεις σύμφωνα με την ενεργητικότητα για παράδειγμα ενός τμήματος. Η PHP συνάρτηση που υπολογίζει και προβάλλει τα στατιστικά ανά τμήμα είναι `statsDepartments` που βρίσκεται στο βοηθητικό αρχείο `functions.php`. Επιτελεί παρόμοια λειτουργία με τη μέθοδο `statsSystems`.

4 Εγχειρίδιο χρήσης της εφαρμογής

Η εφαρμογή που αναπτύχθηκε υπάρχει online και βρίσκεται σε αυτήν την τοποθεσία: <http://tree-system.comze.com>. Παρακάτω είναι η πρώτη εικόνα που προβάλλεται στο χρήστη όταν θα επιχειρήσει να κάνει είσοδο στο σύστημα. Η διαδικασία είναι απλή και ο χρήστης πληκτρολογεί το username και το password και ύστερα πατάει το κουμπί Είσοδος για την είσοδο στο



Εικόνα 4.1: Η είσοδος

στο σύστημα(*index.php*)
σύστημα.

Ας υποθέσουμε ότι πραγματοποιεί είσοδο ο χρήστης. Θα εμφανιστεί η νέα σελίδα `user.php`. Αριστερά φαίνεται το ημερολόγιο των συμβάντων του τμήματος στο οποίο ανήκει. Μέσα στις αντίστοιχες ημέρες μπορεί να δει τα συμβάντα σε χρωματιστά ορθογώνια. Το κόκκινο σημαίνει ότι κατεγράφη βλάβη, ενώ το πράσινο σημαίνει ότι έχει καταγραφεί συντήρηση.

Εγκατάσταση Αθηνών
 Τμήμα 5 - user2 Αποσύνδεση

<
σήμερα
>

Μάιος 2014

μήνας
εβδομάδα

Κυρ	Δευ	Τρι	Τετ	Πεμ	Παρ	Σαβ
27	28	29	30	1	2	3
4	5 7:30α Βλάβη	6	7 8α Συντήρηση	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Αναφορά:

Υ/Σ Στατιστικά Τμήματος

Υ/Σ 1

HY1
HY2
HY6

Υ/Σ 2

HY3

E

ικώνα 4.2: Η σελίδα του χρήστη (`user.php`)

Κάτω από το ημερολόγιο βλέπει την περιοχή της αναφοράς όταν ο χρήστης επιλέξει μία ημέρα που έχει καταγραφεί συμβάν. Η αναφορά καλύπτει το υπολογιστικό σύστημα και το μεμονωμένο υπολογιστή που κατεγράφη το συμβάν.

Εγκατάσταση Αθηνών
 Τμήμα 5 - user2 [Αποσύνδεση](#)

<
σήμερα
>

Μάιος 2014

μήνας
εβδομάδα

Κυρ	Δευ	Τρι	Τετ	Πεμ	Παρ	Σαβ
27	28	29	30	1	2	3
4	5 7:30α Θιάβη	6	7 8α Συντήρηση	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Αναφορά:

Υ/Σ 1-ΗΥ1: χαλασμένο

Υ/Σ
Στατιστικά Τμήματος

Υ/Σ 1

HY1
HY2
HY6

Υ/Σ 2

HY3

Ει

κόνα 4.3: Η αναφορά του συμβάντος που επέλεξε ο χρήστης

Στην παρακάτω εικόνα ο χρήστης έχει τη δυνατότητα να προβάλλει το ημερολόγιο ανά εβδομάδα. Αυτή η προβολή επιτρέπει στο χρήστη να δει τη διάρκεια ενός συμβάντος, που συμπεραίνεται από του ύψος του ορθογωνίου.

Εγκατάσταση Αθηνών
Τμήμα 5 - user2 Αποσύνδεση

<
σήμερα
>
Μαρ 30 — Απρ 5 2014
μήνας
εβδομάδα

Υ/Σ
Στατιστικά Τμήματος

	Κυρ 3/30	Δευ 3/31	Τρι 4/1	Τετ 4/2	Πεμ 4/3	Παρ 4/4	Σαβ 4/5
all-day							
6am						6:30 - 7:30 Συντήρηση	6:30 - 7:30 Συντήρηση
7am							
8am							
9am							
10am							
11am							
12pm							
1pm							
2pm							

Αναφορά:

Υ/Σ 1-HY2: συντήρηση2

Υ/Σ 1

HY1
HY2
HY6

Υ/Σ 2

HY3

Ει

κόνα 4.4: Η διάρκεια των συμβάντων στην προβολή του ημερολογίου ανά εβδομάδα

Η παρακάτω εικόνα αφορά στην καταχώρηση ενός συμβάντος. Στο χρήστη προβάλλονται τα υπολογιστικά συστήματα υπό την ευθύνη του και από κάτω οι υπολογιστές που ανήκουν σε αυτά. Όταν ο χρήστης πατήσει σε ένα κουμπί-υπολογιστή, εμφανίζεται μία φόρμα που επιτρέπει στο χρήστη να καταχωρήσει την αναφορά, το είδος του συμβάντος και τη συντήρηση ή αποκατάσταση του οικείου συμβάντος. Ύστερα ο χρήστης πατάει στο κουμπί Καταγραφή για την υποβολή της φόρμας.

Εγκατάσταση Αθηνών
Τμήμα 5 - user2 [Αποσύνδεση](#)

Υ/Σ Στατιστικά Τμήματος

Υ/Σ 1

HY1 HY2 HY6

Υ/Σ 2

HY3

HY3 Συντήρηση Συντηρ./Αποκατ. λεπτά

Αναφορά

Καταγραφή

Εικόνα 4.5:

Η φόρμα που συμπληρώνει ο χρήστης για την καταχώρηση του συμβάντος

Παρακάτω βλέπουμε το επόμενο tab-καρτέλα όπου δίνεται στο χρήστη ένα πτυσσόμενο μενού για να επιλέξει να προβάλλει αναλυτικά στατιστικά για τα συμβάντα που έχουν καταγραφεί στο τμήμα του. Οι επιλογές αφορούν στην προβολή πίνακα ή διαγράμματος για βλάβες και συντηρήσεις σε ξεχωριστή προβολή για ένα έτος. Στην επόμενη εικόνα ο χρήστης επέλεξε να προβάλλει πίνακα για τις συντηρήσεις του τμήματός του.

Εγκατάσταση Αθηνών
Τμήμα 5 - user2 [Αποσύνδεση](#)

Υ/Σ Στατιστικά Τμήματος

Πίνακας - Συντηρήσεις Τμήματος 2014

Μάρτιος 2014

Υ/Σ 2	1 συντηρ.
Υ/Σ 1	3 συντηρ.

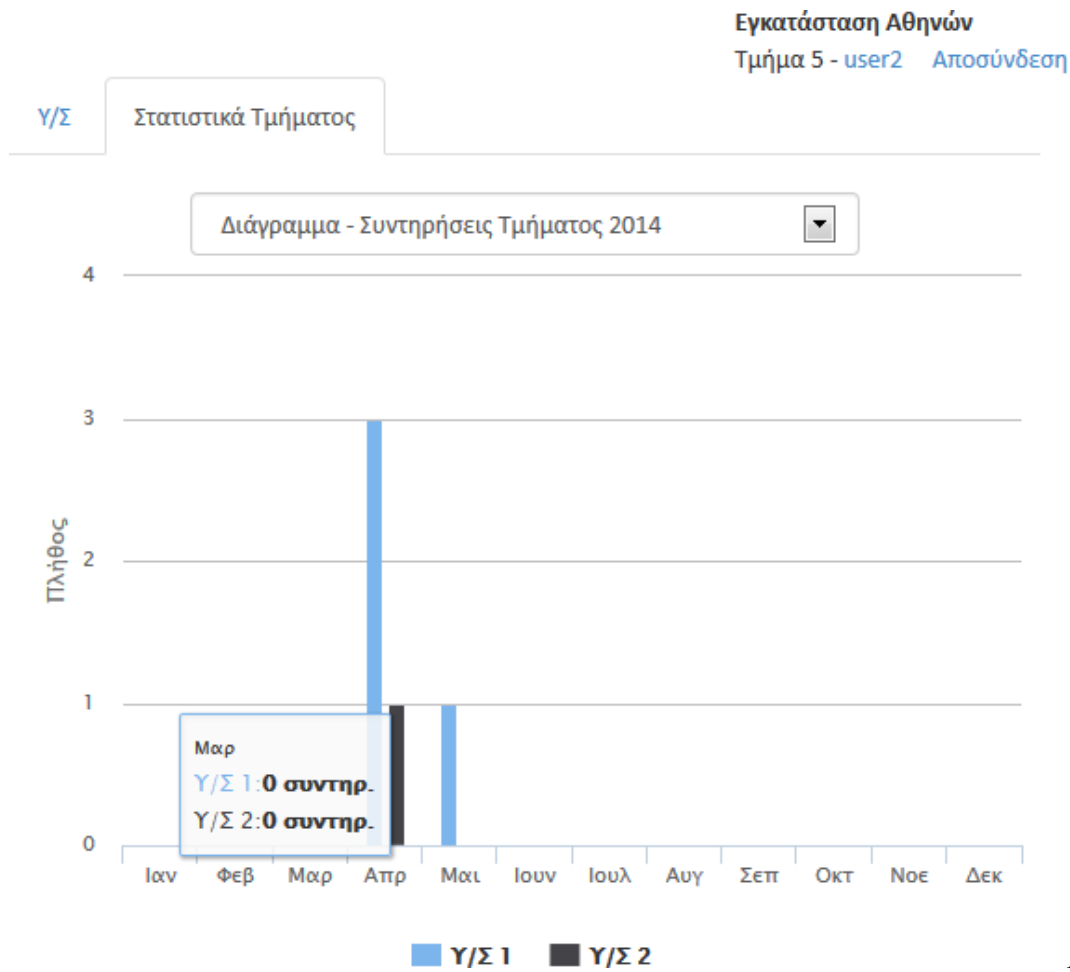
Απρίλιος 2014

Υ/Σ 1	1 συντηρ.
-------	-----------

Εικόνα

4.6: Ο πίνακας των συντηρήσεων του Τμήματος 5 για το 2014

Στην επόμενη εικόνα ο χρήστης επέλεξε να δει στατιστικά για τις συντηρήσεις του 2014 σε μορφή



Εικόνα 4.7:

Το διάγραμμα των συντηρήσεων του Τμήματος 5 για το 2014 διαγράμματος.

Στη συνέχεια θα παρουσιαστεί η σελίδα χρήσης (admin.php) της σελίδας του διαχειριστή. Το ημερολόγιο παραλείπεται να εξηγηθεί, καθώς εμφανίζεται και έχει την ίδια λειτουργία όπως του χρήστη. Παρακάτω φαίνεται μόνο το δεξί μέρος της σελίδας, όπου ο διαχειριστής έχει τη δυνατότητα να εισάγει νέο υπολογιστικό σύστημα στη βάση δεδομένων. Ένα Υ/Σ που αποτελείται από ομάδα υπολογιστών καταλαμβάνει φυσικό χώρο και πρέπει να καταχωρηθεί στη ΒΔ από το διαχειριστή. Στο πρώτο μενού αριστερά επιλέγει εάν πρόκειται για νέο ή υπάρχον Υ/Σ (έχει ήδη καταχωρηθεί – όπου και επιλέγει ένα υπάρχον Υ/Σ). Σε κάθε περίπτωση, επιλέγει από το δεύτερο μενού τα εξαρτήματα κάθε υπολογιστή που θα απαρτίζουν το Υ/Σ. Έπειτα, ο διαχειριστής οφείλει να αναθέσει το Υ/Σ υπό την ευθύνη ενός χρήστη εάν πρόκειται για νέο Υ/Σ και να πατήσει το κουμπί Εισαγωγή για την υποβολή της φόρμας.

Εγκατάσταση Αθηνών
Τμήμα 1 - admin [Αποσύνδεση](#)

Υ/Σ Χρήστης Στατιστικά

Νέο Υ/Σ -Χρήστης- Εισαγωγή

Οθόνη × |

- Οθόνη
- Πληκτρολόγιο
- Ποντίκι
- Ηχεία
- Εκτυπωτής
- Επεξεργαστής
- Μητρική κάρτα
- Μνήμη RAM
- Σκληρός δίσκος
- Κάρτα

Καταχώρηση Υ/Σ και υπολογιστών στη βάση δεδομένων

Εικόνα 4.8:

Εδώ ο διαχειριστής θα εισάγει ένα νέο χρήστη στη βάση δεδομένων πληκτρολογώντας το username και το password που θα έχει. Στη συνέχεια, θα ορίσει το τμήμα στο οποίο θα ανήκει ο χρήστης από το παρακάτω μενού, επιλέγοντας ένα από τα υπάρχοντα τμήματα και πατώντας στο κουμπί Δημιουργία

Εγκατάσταση Αθηνών
Τμήμα 1 - [admin](#) [Αποσύνδεση](#)

[Υ/Σ](#) [Χρήστης](#) [Στατιστικά](#)

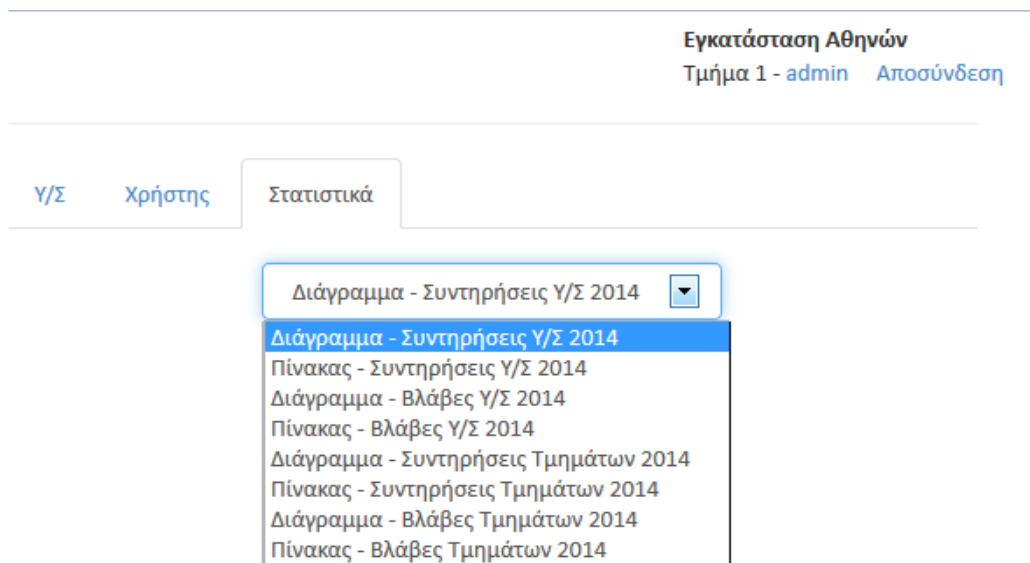
Νέος χρήστης

Κωδικός

Εικόνα 4.9:

Εισαγωγή νέων χρηστών στη βάση δεδομένων

Τα αναλυτικά στατιστικά του διαχειριστή δε διαφέρουν αρκετά από τα αντίστοιχα του χρήστη. Ο διαχειριστής, όμως, έχει επιπλέον επιλογές από το μενού και μπορεί να προβάλλει τις συντηρήσεις και τις βλάβες σε Υ/Σ για ένα έτος συγκριτικά για τα τμήματα του οργανισμού.



Τα αναλυτικά στατιστικά για το διαχειριστή

Εικόνα 4.10:

Μάιος 2014

Κυρ	Δευ	Τρι	Τετ	Πεμ	Παρ	Σαβ
27	28	29	30	1	2	3

[Υ/Σ](#)
[Χρήστης](#)
[Στατιστικά](#)

Εικόνα 4.11: Αποσύνδεση του χρήστη από το σύστημα πατώντας πάνω δεξιά τον υπερσύνδεσμο Αποσύνδεση

Παρακάτω βλέπουμε ένα μήνυμα επιτυχίας που ενημερώνει ότι με επιτυχία έχει εισάγει στο σύστημα ένα νέο Υ/Σ σύστημα. Όλα τα μηνύματα επιτυχίας και αποτυχίας εμφανίζονται πάνω στο κέντρο της σελίδας και διαρκούν 3 δευτερόλεπτα μέχρι να εξαφανιστούν.

Επιτυχής εισαγωγή υπολογιστικού συστήματος.

Μάιος 2014

Κυρ	Δευ	Τρι	Τετ	Πεμ	Παρ	Σαβ
27	28	29	30	1	2	3
4	5	6	7	8	9	10
	7:30α Βλάβη		8α Συντήρηση			
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Αναφορά:

Εγκατάσταση Αθηνών
Τμήμα 1 - admin Αποσύνδεση

[Υ/Σ](#)
[Χρήστης](#)
[Στατιστικά](#)

Εικόνα 4.12: Μήνυμα επιτυχίας στο διαχειριστή έπειτα από εισαγωγή υπολογιστικού συστήματος

5 Συμπεράσματα

Η οργανωτική ιεραρχική δομή είναι μία παραδοσιακή προσέγγιση για την οργάνωση μίας επιχείρησης όπου η έμφαση τοποθετείται από πάνω προς τα κάτω (top-down) όσον αφορά τους συσχετισμούς μεταξύ των τμημάτων που απαρτίζουν την επιχείρηση. Τα πλεονεκτήματα περιγράφηκαν στα προηγούμενα κεφάλαια. Στην ιεραρχική δομή υπάρχουν ξεκάθαροι ρόλοι ανάμεσα στα υπό-συστήματα και στο προσωπικό που τα διαχειρίζεται. Η διαχείριση προσδιορίζει διαφορετικά επίπεδα ηγεσίας και ευθυνών μέσα στον οργανισμό. Επιτρέπει παράλληλα την εξειδίκευση σε ένα τμήμα όσον αφορά το προσωπικό.

Το πληροφοριακό σύστημα που κατασκευάστηκε στα πλαίσια της πτυχιακής εργασίας με βάση το παραπάνω μοντέλο εκπληρώνει το σκοπό για τον οποίο σχεδιάστηκε έχοντας υλοποιήσει προγραμματιστικά κάθε απαίτηση κατά τον καθορισμό των προδιαγραφών. Η εφαρμογή δοκιμάστηκε επιτυχώς κάτω από ιδανικές συνθήκες σε διακομιστή (web server) με ελάχιστο φόρτο εργασίας. Το σύστημα μπορεί να έχει πρακτική και εμπορική εφαρμογή σε οργανισμούς που διαχειρίζονται και συντηρούν μεσαίο αριθμό υπολογιστών.

6 Βιβλιογραφία

- Τεχνολογία Λογισμικού, Μανόλης Γιακουμάκης, Νίκος Διαμαντίδης, Εκδόσεις Αθ. Σταμούλης
- PHP and MySQL Web Development, Luke Welling, Laura Thomson, Developer's Library, Fourth Edition
- jQuery in Action, Second Edition, Bear Bibeault and Yehuda Katz, Manning
- Twitter Bootstrap Web Development How-To, David Kochran
- AJAX and PHP: Building Modern Web Applications 2nd Edition, Bogdan Brinzarea-Iamandi, Christian, Darie, Audra Hendrix, Packt Publishing
- Essential PHP Security, Chris Shiflett, O'Reilly
- Professional JavaScript for Web Developers, Nicholas C. Zakas, 3rd Edition, Wrox,
- PHP Documentation, <http://www.php.net/>,
- MySQL Documentation, <http://www.mysql.com/>
- JQuery Documentation, <http://jquery.com/>
- Bootstrap Documentation, <http://getbootstrap.com/getting-started/>
- FullCalendar Documentation <http://arshaw.com/fullcalendar/docs/>
- Event monitoring, http://en.wikipedia.org/wiki/Event_monitoring
- Log management and intelligence, http://en.wikipedia.org/wiki/Log_management_and_intelligence
- JQuery, Wikipedia, <http://en.wikipedia.org/wiki/Jquery>
- PHP, Wikipedia, <http://en.wikipedia.org/wiki/Php>
- JavaScript, Wikipedia ,<http://en.wikipedia.org/wiki/JavaScript>
- Ajax, Wikipedia, <http://en.wikipedia.org/wiki/Ajax>
- MySQL, Wikipedia, <http://en.wikipedia.org/wiki/Mysql>
- HTML5, <http://en.wikipedia.org/wiki/HTML5>
- JSON, <http://en.wikipedia.org/wiki/JSON>
- Web service, http://en.wikipedia.org/wiki/Web_service
- Chosen jQuery plugin, <http://harvesthq.github.io/chosen/>
- Highcharts, <http://www.highcharts.com/docs>
- Open-source software, http://en.wikipedia.org/wiki/Open-source_software
- SQL injection, Wikipedia, http://en.wikipedia.org/wiki/SQL_injection
- Database, Wikipedia, <http://en.wikipedia.org/wiki/Database>
- Database management system, Wikipedia, <http://en.wikipedia.org/wiki/Dbms>
- HTML, Wikipedia, <http://en.wikipedia.org/wiki/Html>
- Cascading Style Sheets, Wikipedia, <http://en.wikipedia.org/wiki/Css>
- Client–server model, Wikipedia, <http://en.wikipedia.org/wiki/Client-server>
- Client (computing), Wikipedia, http://en.wikipedia.org/wiki/Client_%28computing%29
- Server (computing), Wikipedia, http://en.wikipedia.org/wiki/Server_%28computing%29
- Hierarchical database model, http://en.wikipedia.org/wiki/Hierarchical_database_model
- Analysis and design of Information Systems by Arthur M. Langer

- Flat Vs. Hierarchical Organizational Structure, <http://smallbusiness.chron.com/flat-vs-hierarchical-organizational-structure-724.html>
- HIERARCHICAL ORGANISATION, <http://www.learnmanagement2.com/hierarchical%20structure.htm>
- Decision making techniques, <http://businesscasestudies.co.uk/cima/decision-making-techniques/introduction.html>
- Decision tree, http://en.wikipedia.org/wiki/Decision_tree
- Decision Trees for Decision Making, John F. Magee
- How Gerber Used a Decision Tree in Strategic Decision-Making, <http://gbr.pepperdine.edu/2010/08/how-gerber-used-a-decision-tree-in-strategic-decision-making/>
- Decision trees, <http://www.accaglobal.com/zw/en/student/acca-qual-student-journey/qual-resource/acca-qualification/f5/technical-articles/decision-trees.html>
- Introduction to Decision Trees, <http://www.treeplan.com/chapters/introduction-to-decision-trees.pdf>
- DECISION TREES, Lior Rokach
- Advantages Business Hierarchy, <http://www.hierarchystructure.com/advantages-business-hierarchy>
- Pros & Cons of a Hierarchical Organizational Structure, <http://yourbusiness.azcentral.com/pros-cons-hierarchical-organizational-structure-1208.html>
- Management Information Systems (MIS), <http://www.inc.com/encyclopedia/management-information-systems-mis.html>
- Introduction to Management Information Systems (MIS), <http://www.mu.ac.in/mis.pdf>
- A Theory of Theories in Information Systems, <http://www.caseykings.net/wp-content/uploads/2012/01/Gregor-ISF-theory-paper-final-t.pdf>
- Information Systems:towards a new theory, Rodrigo Manuel Oliveira da Silva Ponciano de Magalhães
- INFORMATION SYSTEMS THEORY 101, Tim Bryce
- Management information system, http://en.wikipedia.org/wiki/Management_information_system
- What is the right organization structure, Organizational Dynamics
- The Inherent Complexity of Software, <http://www.informit.com/articles/article.aspx?p=726130&seqNum=2>

Γλωσσάρι ξενικών όρων

Event	συμβάν
Work flow	ροή εργασιών
Script	σενάριο
Header	επικεφαλίδα
Root directory	κύριος φάκελος
Plugins	πρόσθετα
Format	μορφή
Hardware	υλικό υπολογιστή
peripherals	περιφερειακά υπολογιστή
client-layer	επίπεδο πελάτη
client	πελάτης
server	εξυπηρετητής, διακομιστής
open-source	ανοιχτού κώδικα
decision tree structure	δενδροειδής δομή
user	χρήστης
administrator	διαχειριστής
web-based	βασισμένο στον Ιστό
intranet	ενδοεταιρικό δίκτυο
root node	κύριος κόμβος
leave	φύλλο, κλαδί
node	κόμβος
scheme	σχήμα
decision environment	περιβάλλον απόφασης
view	σελίδα
responsiveness	ανταποκρισιμότητα
insert	εισάγω
dropdown menu	πτυσσόμενο μενού
tab	καρτέλα
client-server model	μοντέλο πελάτη-διακομιστή
cascade	διαδοχικός
request	αίτημα
jQuery plugin	πρόσθετο για τη βιβλιοθήκη jQuery
filesystems	αρχεία συστημάτων
private networks	ιδιωτικά δίκτυα

hyperlink	υπερσύνδεσμος
copyright	πνευματικά δικαιώματα
localhost	εικονικός server
redirect	αναδρομολογώ
include	περιλαμβάνω (αρχεία)
connection credentials	πιστοποιητικά εισόδου
cross-browser	εκτελείται σε όλους τους περιηγητές Ιστού
web browser	περιηγητής Ιστού
populate	γεμίζω
animated	κινούμενο
object connection	αντικείμενο που κρατά τη σύνδεση (με τη βάση δεδομένων)
SQL injection	επίθεσης της μορφής SQL
security hole	τρύπα ασφαλείας
log management	διαχείριση καταγραφής
log analysis	ανάλυση καταγραφής
audit	έλεγχος
auditor	ελεγκτής
data logging	καταγραφή δεδομένων
regulation	κανονισμός
incident	συμβάν, περιστατικό
context	πλαίσιο
induce	επιφέρω
reverse engineering software	αντίστροφη μηχανική λογισμικού
Event logging management software	Λογισμικό διαχείρισης καταγραφών συμβάντων
IT administrator	διαχειριστής πληροφορικής

Συντμήσεις

ΒΔ	Βάση δεδομένων
DB	Database
XML	Extensible Markup Language
SQL	Structured Query Language
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
AJAX	Asynchronous JavaScript and XML
PHP	Hypertext Processor
URI	Uniform resource identifier
URL	Uniform Resource Locator
GUI	Graphical User interface
Υ/Σ	Υπολογιστικό σύστημα
JSON	Graphical User interface
ER	Entity-relationship
DOM	Document Object Model
HTTP	Hypertext Transfer Protocol
MIT	Massachusetts Institute of Technology
ELMS	Event logging management software