



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ &

ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

## **ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

Αυτοματοποιημένη καταγραφή συμπεριφοράς χρηστών για την αξιολόγηση διεπαφής εμπορικής εφαρμογής χρησιμοποιώντας κατακευματισμένη NoSQL βάση δεδομένων.

Νικολαΐδου Φανή

Επιβλέπων καθηγητής: Δρ. Αντώνιος Πρωτοψάλτης

Δρ. Παναγιώτης Σαρηγιαννίδης

Οκτώβριος 2019, Κοζάνη

## Ευχαριστίες

Θα ήθελα να ευχαριστώ τον καθηγητή μου κ. Δρ. Αντώνιο Πρωτοψάλτη για την βοήθεια και την καθοδήγηση που μου παρείχε κατά την εκπόνηση της διπλωματικής μου εργασίας.

Επίσης, θα ήθελα να ευχαριστώ τον κ. Γιάννη Γιουρούκη και την εταιρία Semantic για την προσφορά των υποδομών λογισμικού και τον χρόνο που αφιέρωσε για την περάτωση της παρούσας διπλωματικής.

Τέλος, θα ήθελα να εκφράσω ένα μεγάλο ευχαριστώ στην οικογένεια μου για την στήριξη που έδειξε.

## Περίληψη

Η ανάγκη για εύχρηστα συστήματα αυξάνεται συνεχώς, για αυτό και η συνεχής συλλογή πληροφοριών για την σωστή αξιολόγηση αυτών κρίνεται πολύ σημαντική.

Αντικείμενο της παρούσας διπλωματικής είναι η μελέτη, ανάπτυξη και σχεδίαση αυτοματοποιημένης καταγραφής πληροφοριών με βάση τη συμπεριφορά του χρήστη με σκοπό την αξιολόγηση της εμπορικής ERP εφαρμογής Business Evolution.

Η καταγραφή των δεδομένων γίνεται σε πραγματικό χρόνο και ο όγκος των δεδομένων προβλέπεται μεγάλος. Για αυτόν τον λόγο, η καταγραφή των πληροφοριών γίνεται με την κατάλληλη για αυτή την χρήση, CouchDB, η οποία είναι μια κατανεμημένη NoSQL βάση δεδομένων.

Η συλλογή των δεδομένων γίνεται με σκοπό την στατιστική ανάλυση της οποίας η σημασία είναι μεγάλη στις περιπτώσεις αξιολόγησης διαδραστικών συστημάτων και έχει ως στόχο την εξαγωγή συμπερασμάτων ως προς την επαρκή σχεδίαση ενός εύχρηστου συστήματος για τον γενικότερο πληθυσμό.

Για την ερμηνεία των δεδομένων, υλοποιήθηκαν ερωτήματα που αφορούν την ευχρηστία του συστήματος και έγινε στατιστική ανάλυση και οπτικοποίηση των αποτελεσμάτων ώστε να δοθεί η δυνατότητα κατανόησης συμπερασμάτων της ευχρηστίας του συστήματος αλλά και η δυνατότητα χρήσης των αποτελεσμάτων για περαιτέρω έρευνα.

## **Abstract**

The need for easy-to-use systems is constantly increasing, so a continuous collection of information to properly evaluate these systems, is very important.

The purpose of this thesis is to study, develop and design automated logging of user behavior for the purpose of evaluating the commercial ERP application, Business Evolution.

The data is recorded in real time and the volume of data is enormous. For this reason, the information is recorded using the appropriate CouchDB, which is a distributed Nosql database.

The data is collected for the purpose of statistical analysis, which is of a great importance in the case of interactive systems evaluation concludes to the adequate design of a user-friendly system for the general population.

For the interpretation of the data, questions, we considered to be related to system usability, were implemented and the results were statistically analyzed and visualized in order to enable us understand the system usability and use the results for further research.

# Περιεχόμενα

Ευχαριστίες.....	2
Περίληψη.....	3
Abstract.....	4
Κατάλογος εικόνων/σχημάτων.....	8

## ΚΕΦΑΛΑΙΟ 1<sup>ο</sup> : Εισαγωγή

1.1 Προσδιορισμός Προβλήματος.....	10
1.2 Κίνητρο.....	11
1.3 Σκοπός.....	12
1.4 Δομή.....	13

## ΚΕΦΑΛΑΙΟ 2<sup>ο</sup> : Αλληλεπίδραση Ανθρώπου-Υπολογιστή

2.1 Εισαγωγή.....	14
2.1.1 Χρήστης.....	14
2.1.2 Υπολογιστής.....	15
2.1.3 Αλληλεπίδραση.....	15
2.2 Στόχοι έρευνας της Αλληλεπίδραση Ανθρώπου Υπολογιστή.....	17
2.3 Τομείς που συμβάλλουν στην αλληλεπίδραση Ανθρώπου-Υπολογιστή.....	18
2.4 Μοντέλα ανθρώπινης συμπεριφοράς που σχετίζονται με την ευχρηστία.....	20
2.4.1 Το μοντέλο του ανθρώπινου επεξεργαστή.....	21
2.5 Η προέλευση και οι στόχοι της Αλληλεπίδραση Ανθρώπου-Υπολογιστή.....	22
2.5.1 Οι στόχοι της Αλληλεπίδραση Ανθρώπου-Υπολογιστή.....	23
2.6 Η Ευχρηστία.....	24
2.6.1 Αξιολόγηση Ευχρηστίας.....	25
2.6.2 Συστήματα καταγραφής της συμπεριφοράς χρηστών.....	27
2.6.3 Από την ευχρηστία στην εμπειρία χρήστη.....	28
2.7 Η Χρηστικότητα.....	28
2.7.1 Αξιολόγηση της Χρηστικότητας.....	29
2.7.2 Γιατί είναι σημαντική η χρηστικότητα;.....	30

## ΚΕΦΑΛΑΙΟ 3<sup>ο</sup> : NoSql Βάσεις Δεδομένων

3.1 Οι NoSQL Βάσεις Δεδομένων.....	31
3.2 Οι ιδιότητες ACID, BASE και το θεώρημα CAP.....	32

3.3	Θετικά χαρακτηριστικά NoSql Βάσεων Δεδομένων.....	34
3.3.1	<i>Πλεονεκτήματα συγκριτικά με τις SQL βάσεις δεδομένων</i> .....	35
3.4	Κατηγορίες NoSQL Βάσεων Δεδομένων .....	37
3.5	Συγκριτικός πίνακας Σχεσιακών- μη σχεσιακών βάσεων δεδομένων.....	40

## ΚΕΦΑΛΑΙΟ 4<sup>ο</sup> : CouchDB

4.1	Τι είναι η CouchDB; .....	41
4.2	Θεώρημα CAP .....	42
4.3	Λειτουργίες της CouchDB.....	43
4.3.1	<i>Αναπαραγωγή</i> .....	43
4.3.2	<i>Master - Master και Master Slave Αναπαραγωγή</i> .....	43
4.3.3	<i>Υποστήριξη για κινητά</i> .....	44
4.3.4	<i>MapReduce</i> .....	44
4.3.5	<i>Views</i> .....	45
4.3.6	<i>Έλεγχος Ταυτοχρονισμού Πολλαπλών Εκδόσεων (MVCC)</i> .....	46
4.3.7	<i>Ασφάλεια και επικύρωση</i> .....	46
4.4	Πλεονεκτήματα της CouchDB .....	47
4.4.1	<i>Πίνακας Σύγκρισης Σχεσιακής Βάσης Δεδομένων και CouchDB</i> .....	48
4.5	Έγγραφο CouchDB .....	49
4.5.1	<i>Fauxton</i> .....	50
4.6	Ιστορία της CouchDB.....	51

## ΚΕΦΑΛΑΙΟ 5<sup>ο</sup> : Περιγραφή του Project

5.1	Σκοπός .....	53
5.2	Εισαγωγή .....	53
5.3	ERP σύστημα Business Evolution.....	53
5.3.1	<i>Σύστημα ERP</i> .....	54
5.3.2	<i>Business Evolution</i> .....	54
5.3.3	<i>Περιγραφή φόρμας «Είδη»</i> .....	55
5.3.4	<i>Προφίλ Χρήστη</i> .....	58
5.4	Αρχιτεκτονική συστήματος .....	59
5.4.1	<i>Υποσύστημα Καταγραφής συμπεριφοράς χρήστη</i> .....	60
5.4.2	<i>Υποσύστημα Καταγραφής στατιστικής ανάλυσης</i> .....	62

## **ΚΕΦΑΛΑΙΟ 6<sup>ο</sup> : Υποσύστημα καταγραφής συμπεριφοράς χρήστη**

6.1 Το εργαλείο Delphi .....	64
6.1.1 Προγραμματισμός σε Delphi .....	65
6.2 Εγκαταστάση LAMP και CouchDB σε Ubuntu .....	65
6.3 Σύνδεση με την Βάση δεδομένων CouchDB .....	66
6.3.1 cURL.....	67
6.3.2 Εγκατάσταση CouchDB και Σύνδεση μέσω cURL .....	67
6.3.3 Η σύνδεση του Business Evolution με την CouchDB μέσω cURL .....	68
6.4 Περιγραφή και επεξήγηση ενός CouchDB document .....	71

## **ΚΕΦΑΛΑΙΟ 7<sup>ο</sup> : Υποσύστημα στατιστικής ανάλυσης**

7.1 Σχεδιασμός υποσυστήματος .....	76
------------------------------------	----

## **ΚΕΦΑΛΑΙΟ 8<sup>ο</sup> : Αξιολόγηση Ευχρηστίας**

8.1 Ο Σκοπός για την εξαγωγή συμπερασμάτων της στατιστικής ανάλυσης .....	80
8.2 Συμπερασματικές δοκιμές ευχρηστίας .....	80
8.3 Στατιστική ανάλυση, συμπεράσματα και μελλοντικές εργασίες .....	81

## **Κεφάλαιο 9<sup>ο</sup> : Συμπεράσματα**

9.1 Σύνοψη .....	91
9.2 Μελλοντικές Επεκτάσεις.....	92

## **Πηγές και Βιβλιογραφία .....**

94

### **Παράρτημα**

A Εγκατάσταση LAMP και CouchDB σε Ubuntu .....	97
A.1 Εισαγωγή και Προϋποθέσεις.....	97
A.2 Εγκατάσταση του Apache .....	97
A.3 Εγκατάσταση της PHP .....	101
A.4 Εγκατάσταση της CouchDB.....	103
B Δημιουργία requests στη Βάση Δεδομένων μέσω cURL.....	105
Γ Διάγραμμα καταστάσεων της φόρμας «Είδη» .....	108
Δ Κώδικας (Script) Εφαρμογής .....	113
E Υλοποίηση Υποσυστήματος Στατιστικής Ανάλυσης .....	126

## Κατάλογος Εικόνων

Εικόνα 2.1 Πολυεπιστημονικός τομέας της HCI

Εικόνα 2.2 Χρηστικότητα εκτυπωτή

Εικόνα 4.1: Συμμόρφωση της CouchDB στο θεώρημα Cap

Εικόνα 4.2: Τυπικό έγγραφο Couch

Εικόνα 4.3: Αποθήκευση εγγράφων Couch

Εικόνα 5.1: Φόρμα Είδη του Business Evolution

Εικόνα 5.2 :Διάγραμμα Ροής εργασιών του χρήστη

Εικόνα 5.3 : Διάγραμμα Ροής ενεργειών του χρήστη σε κατάσταση επεξεργασίας μιας εγγραφής

Εικόνα 6.1: Business Evolution

Εικόνα 6.2: CouchDB 2.1.1

Εικόνα 6.3: cURL

Εικόνα 6.4: Fauxton Program-Web GUI

Εικόνα 6.5: cmd cURL –version

Εικονά 6.7: Μήνυμα λάθους

Εικόνα 6.8: Fauxton GUI

Εικόνα 6.9: Η Βάση «ux\_tracker» στο Fauxton GUI

Εικόνα 6.10: Εγγραφο Couch

Εικόνα 7.1: Queries

Εικόνα 7.2: Queries:Day

Εικόνα 7.3: Γράφημα

Εικόνα 7.4: Previous Field

Εικόνα 7.5: User Actions

Εικόνα 7.6: Track way 1

Εικόνα 7.7: Track way 2

Εικόνα 7.8: Track Way 3

Εικόνα 7.9: Insert

Εικόνα 7.10: Save

Εικόνα 7.11: Tabsheet

Εικόνα 7.12: Blank Field 1

Εικόνα 7.13: Blank Field 2



Εικόνα 7.14: Print  
Εικόνα 7.15: Checkbox  
Εικόνα 7.16: Day  
Εικόνα 7.17: Time  
Εικόνα 7.18: incorrect actions per user  
Εικόνα 7.19: wasted time per user  
Εικόνα 7.8: Fraction Right/Wrong  
Εικόνα 7.21: Time for an action 1  
Εικόνα 7.22: Time for an action 2  
Εικόνα 7.23: Completed Record  
Εικόνα 7. 24: Διάγραμμα αρχιτεκτονικής συστήματος  
Εικόνα 7.25 : Διάγραμμα ροής δεδομένων  
Εικόνα 7.26: Διάγραμμα υποσυστήματος

## **Κατάλογος Πινάκων**

Πίνακας 3.1: Τύποι NoSQL Βάσης δεδομένων

Πίνακας 3.2: SQL vs NoSQ

Πίνακας 4.1: Πίνακας Σύγκρισης Σχεσιακής Βάσης Δεδομένων με Βάση δεδομένων  
CouchDB

# Κεφάλαιο 1

## Εισαγωγή

Η θεματική ενότητα των έξυπνων στατιστικών αποτελεί ένα διαρκώς διερευνώμενο αντικείμενο που απασχολεί ιδιαίτερα τον ακαδημαϊκό και ερευνητικό τομέα. Αφορά την δυνατότητα απόκτησης πολύ μεγάλου όγκου δεδομένων μέσα από τη χρήση προσωπικών ηλεκτρονικών συσκευών, την επεξεργασία τους και την κατανόηση συμπεριφορών και τάσεων που υποδεικνύουν αλλαγές.

Κύριο χαρακτηριστικό που διαχωρίζει την παραγωγή στατιστικών δεικτών επί αυτών των δεδομένων από άλλες παραδοσιακές στατιστικές μεθόδους, είναι η δυνατότητα επεξεργασίας αυτών των μεγάλου όγκου δεδομένων ανά πάσα στιγμή. Ακριβώς επειδή υπάρχει το πληροφοριακό υπόβαθρο ως υποδομή (χρήση προσωπικών ηλεκτρονικών συσκευών), η συλλογή των δεδομένων γίνεται αυτοματοποιημένα, ενώ οι κύκλοι συλλογής μπορεί να είναι έως και διαρκείς. Αυτή η διαρκής συλλογή δεδομένων μετατρέπει το απλό δεδομένο σε πληροφορία επί του φαινομένου – εξού και η μετεξέλιξη των στατιστικών πρακτικών σε "έξυπνες"<sup>[1]</sup>

### 1.1 Προσδιορισμός προβλήματος

Η συνεχής τεχνολογική εξέλιξη είναι ανάγκη να ακολουθείται και από αντίστοιχα τεχνολογικά εξελιγμένα συστήματα. Καθώς η χρήση των υπολογιστών επεκτείνεται και οι υπολογιστές υπεισέρχονται ολοένα και περισσότερο στη ζωής μας στις σύγχρονες κοινωνίες, αλλάζουν και αυξάνονται κατά συνέπεια και οι απαιτήσεις των χρηστών. Η ευχρηστία, ή αλλιώς η ευκολία στην χρήση, έχει αναγνωριστεί ως σημαντική παράμετρος για τη σχεδίαση και αξιολόγησης προϊόντων, συστημάτων κ.α.

Το πόσο εύκολο στην χρήση είναι ένα σύστημα το κρίνουν επί της ουσίας οι ίδιοι οι χρήστες ακολουθώντας κάποιους κανόνες και μεθόδους κατά την δοκιμαστική εκτέλεση εργασιών . Όσο περισσότεροι χρήστες πετυχαίνουν στην εκτέλεση εργασιών σε σύντομο χρόνο, τόσο πιο εύχρηστο είναι το προϊόν που αξιολογείται.

Αυτό που αποκτά ενδιαφέρον για την επεξεργασία των δεδομένων που αφορούν τον χρόνο επίτευξης εργασίας είναι ο μέσος χρόνος στον οποίο όλοι οι χρήστες πέτυχαν την εργασία, στο διάστημα εμπιστοσύνης που προκύπτει.

Η αξιολόγηση ευχρηστίας ορίζεται ως «η μέτρηση των ιδιοτήτων της ευχρηστίας (ή του προσδιορισμού πιθανών προβλημάτων) ενός συστήματος ή μιας εφαρμογής, σε σχέση με συγκεκριμένους χρήστες, που εκτελούν συγκεκριμένες διεργασίες, σε προκαθορισμένα πλαίσια»<sup>[41]</sup>. Οι μέθοδοι αξιολόγησης ευχρηστίας περιγράφουν με γενικό τρόπο τις προϋποθέσεις, διαδικασίες και εργαλεία με τα οποία μπορεί να αποκτηθεί γνώση για την ευχρηστία της Αλληλεπίδρασης Ανθρώπου Υπολογιστή (ΑΑΥ). [2]

Η εύρεση και η ανάλυση των δεδομένων σήμερα είναι ένα πολύ σημαντικό θέμα για την αξιολόγηση της ευχρηστίας ενός συστήματος. Η συμπεριφορά των χρηστών και τα κύρια πορίσματα υπό μορφή εμπειρικών κανόνων και μεθόδων, έχουν ως στόχο να υποστηρίξουν τη σχεδίαση υπολογιστικών συστημάτων τα οποία παρέχουν θετική εμπειρία χρήσης, και είναι εύχρηστα και προσαρμοσμένα στις ανάγκες και τα χαρακτηριστικά των χρηστών τους.<sup>[3]</sup>

Ιδιαίτερα όμως περίπλοκη είναι η ανάλυση δεδομένων τα οποία είτε μεταβάλλονται με την πάροδο του χρόνου είτε διαφοροποιούνται ανάλογα τον χρηστή και τις δεξιότητες του. Συνεπώς, είναι απαραίτητη η συλλογή δεδομένων και πληροφοριών της καθημερινής συμπεριφοράς του χρήστη για ορισμένο χρονικό διάστημα. Η καθημερινή παρακολούθηση της συμπεριφοράς των χρηστών μίας εφαρμογής και η καταχώριση δεδομένων σε βάση δεδομένων είναι επίπονη και δαπανηρή διαδικασία. Έτσι δημιουργείται η απαίτηση εύρεσης ενός αυτοματοποιημένου τρόπου συλλογής της συμπεριφοράς των χρηστών μίας εφαρμογής. Για μεγάλες εφαρμογές με μεγάλο πλήθος χρηστών τα σύνολα δεδομένων μπορεί να είναι τόσο μεγάλα ώστε οι παραδοσιακές μέθοδοι αποθήκευσης και επεξεργασίας είναι ανεπαρκείς.

## 1.2 Κίνητρο

Η εξέλιξη των τεχνολογιών και η ταχύτατη διείσδυση των χρηστών στις τεχνολογίες πληροφορικής ίσως δημιουργεί προβλήματα στα ήδη υπάρχοντα σύστημα όσον αφορά την αλληλεπίδραση του ανθρώπου με τον υπολογιστή καθιστώντας τις διεπαφές αυτών ‘ξεπερασμένες’ ή λιγότερο εύχρηστες. Υπάρχουσες τεχνικές αξιολόγησης (π.χ. ερωτηματολόγια) μπορούν και καταγράφουν τη συμπεριφορά των χρηστών με μη αυτοματοποιημένο τρόπο ενώ η καταχώριση των δεδομένων αυτών σε βάσεις δεδομένων είναι μια χρονοβόρα διαδικασία, ειδικά όταν τα δεδομένα πρέπει να ενημερώνονται καθημερινά και για μεγάλο πλήθος χρηστών.

Πρόβλημα όμως αποτελεί και η εξαγωγή γνώσης από ένα τεράστιο πλήθος δεδομένων. Για να προκύψουν αντικειμενικά αποτελέσματα είναι απαραίτητος ο υπολογισμός στατιστικών μεγεθών όπου υπολογίζεται από τον μέσο όρο των συλλεγμένων στοιχείων. Η συμπεριφορά του χρήστη αφορά συνήθειες, ευκολία χρήσης ακόμα και την άγνοια βασικών πληροφοριών που αφορούν το σύστημα. Για το λόγο αυτό πραγματοποιείται στατιστική ανάλυση δεδομένων της συμπεριφοράς του χρήστη που υπάρχουν στη βάση και οπτικοποίηση των αποτελεσμάτων με τη βοήθεια στατιστικών οργάνων όπως γραφήματα πίτας.

Η παρούσα πτυχιακή στοχεύει να συμβάλει στην επίλυση του παραπάνω προβλήματος με την ανάπτυξη μιας εφαρμογής για την ανάλυση της συμπεριφοράς των χρηστών ενός συστήματος με αυτοματοποιημένη καταγραφή και ενημέρωση της βάσης δεδομένων. Τα συμπεράσματα αυτής της ανάλυσης θα βοηθήσουν στην καλύτερη κατανόηση της συμπεριφοράς του χρήστη με βάση τον “περίπατο” του στο σύστημα με στόχο την βελτίωση του ήδη υπάρχοντος συστήματος σε επίπεδο ευχρηστίας. Επιπλέον η χρήση αυτών των πληροφοριών για μελλοντικό Project θα είναι πολύ χρήσιμη.

### **1.3 Σκοπός**

Ο σκοπός της παρούσας εργασίας είναι η μελέτη και ανάπτυξη μιας εφαρμογής μέσω της οποίας θα καταγράφεται και θα γίνεται η αποθήκευση δεδομένων και πληροφοριών για την συμπεριφορά όλων των χρηστών που χειρίζονται το σύστημα για συγκεκριμένο χρονικό διάστημα.

Ο όγκος των δεδομένων που πρόκειται να συλλεχθεί προβλέπεται να είναι τεράστιος (πρόβλημα με σχεσιακές βάσεις) γι’ αυτόν τον λόγο η αποθήκευση δεδομένων γίνεται σε NoSQL βάση Δεδομένων η οποία είναι μια ευέλικτη βάση δεδομένων που χρησιμοποιείται για μεγάλα δεδομένα (big data) και για real-time διαδικτυακές εφαρμογές.

Εκτός από την συμπεριφορά του χρήστη θα αποθηκεύονται και άλλες πληροφορίες που βοηθήσουν στην εξαγωγή επιπλέον χρήσιμων συμπερασμάτων. Επίσης τα δεδομένα θα μελετηθούν και θα αναλυθούν και με τη χρήση κατάλληλης στατιστικής βιβλιοθήκης θα γίνει εξαγωγή συμπερασμάτων με μορφή γραφημάτων. Με αυτό τον τρόπο θα δημιουργηθούν γραφικές αναπαραστάσεις των συλλεγμένων πληροφοριών που υπάρχουν στη βάση, ώστε η ομάδα σχεδιασμού της διεπαφής ενός συστήματος να εξάγει ασφαλή

συμπεράσματα σχετικά με την συμπεριφορά των χρηστών της εφαρμογής με στόχο την βελτίωση της ευχρηστίας.

## 1.4 Δομή

- Κεφάλαιο 1: Στο πρώτο κεφάλαιο προσδιορίζεται το πρόβλημα και το κίνητρο και παρουσιάζεται ο σκοπός και η δομή της πτυχιακής εργασίας.
- Κεφάλαιο 2: Στο δεύτερο κεφάλαιο αναλύονται βασικές έννοιες της Αλληλεπίδραση Ανθρώπου Υπολογιστή (Human Computer Interaction )
- Κεφάλαιο 3: Στο τρίτο κεφάλαιο αναλύονται οι NoSql Βάσεις Δεδομένων για να γίνει κατανοητή η σύγκριση με σχεσιακές βάσεις δεδομένων και να τονιστεί η σημαντικότητα τους.
- Κεφάλαιο 4: Το τέταρτο κεφάλαιο παρουσιάζει τη βάση δεδομένων CouchDB δίνοντας ιδιαίτερη έμφαση στα πλεονεκτήματα της.
- Κεφάλαιο 5: Στο πέμπτο κεφάλαιο παρουσιάζεται το Business Evolution ERP, ο σχεδιασμός της εφαρμογής και των υποσυστημάτων της.
- Κεφάλαιο 6: Στο έκτο κεφάλαιο παρουσιάζεται η υλοποίηση του υποσυστήματος καταγραφής συμπεριφοράς του χρήστη.
- Κεφάλαιο 7: Στο έβδομο κεφάλαιο παρουσιάζεται η υλοποίηση του υποσυστήματος στατιστικής ανάλυσης των δεδομένων
- Κεφάλαιο 8: Στο τελευταίο κεφάλαιο η αξιολόγηση της ευχρηστίας του συστήματος με ανάλυση ερωτημάτων.
- Κεφάλαιο 9: Στο τελευταίο κεφάλαιο παρουσιάζονται τα συμπεράσματα της πτυχιακής εργασίας και οι μελλοντικές επεκτάσεις.

Ακολουθούν :Πηγες-Βιβλιογραφία, Παράρτημα

## **Κεφάλαιο 2**

### **Αλληλεπίδραση Ανθρώπου-Υπολογιστή** (Human-Computer Interaction)

#### **2.1 Εισαγωγή**

**Αλληλεπίδραση Ανθρώπου Υπολογιστή - ΑΑΥ (Human-Computer Interaction -HCI)** είναι η μελέτη του πώς οι άνθρωποι αλληλεπιδρούν με τους υπολογιστές και σε ποιο βαθμό οι υπολογιστές αναπτύσσονται ή δεν αναπτύσσονται για επιτυχή αλληλεπίδραση με τον άνθρωπο.[4]

Η επιστήμη αυτή, έχει ως αντικείμενο αφενός τη μελέτη των φαινομένων που σχετίζονται με την αλληλεπίδραση ανθρώπων και υπολογιστών. Αφετέρου αφορά την ανάπτυξη μεθόδων και εργαλείων για το σχεδιασμό, ανάπτυξη και αξιολόγηση εύχρηστων Διαδραστικών Υπολογιστικών Συστημάτων (interactive computer systems) που αλληλεπιδρούν σε μεγάλο βαθμό με τους χρήστες και να μπορούν να ανταποκρίνονται στις απαιτήσεις τους. Ένα εύχρηστο σύστημα προκύπτει ως αποτέλεσμα της εφαρμογής τόσο της θεωρίας και των κανόνων που διέπουν την αλληλεπίδραση ανθρώπου-υπολογιστή όσο και μιας επίπονης διαδικασίας ανάλυσης και αξιολόγησης.[3]

Όπως υπονοεί το όνομά της αλληλεπίδρασης ανθρώπου-υπολογιστή αποτελείται από τρία μέρη: τον χρήστη, τον ίδιο τον υπολογιστή και τους τρόπους με τους οποίους αλληλεπιδρούν.[6]

##### **2.1.1 Χρήστης**

Με τον όρο «χρήστη» εννοείται ένας μεμονωμένος χρήστης ή μια ομάδα χρηστών που συνεργάζονται. Για να γίνει κατανοητό ποιος θα είναι ο χρήστης του συστήματος που σχεδιάζεται θα πρέπει να προσδιοριστεί η ομάδα χρηστών που θα αλληλεπιδράσει με το σύστημα και να γίνει καταγραφή των βασικών χαρακτηριστικών τους. Έτσι, θα αποκωδικοποιηθεί όχι μόνο η συμπεριφορά του χρήστη αλλά και τι σκοπεύει να κάνει με το σύστημα και τις συνθήκες κάτω από τις οποίες θα το χρησιμοποιήσει.

Τα αισθητήρια συστήματα των ανθρώπων (όραση, ακοή, αφή) είναι απαραίτητα για να έρθει ο χρήστης σε επαφή με ένα υπολογιστικό σύστημα. Επίσης, οι διαφορετικοί χρήστες διαμορφώνουν διαφορετικές αντιλήψεις ή διανοητικά μοντέλα για τις αλληλεπιδράσεις τους και έχουν διαφορετικούς τρόπους μάθησης και διατήρησης της γνώσης. Επιπλέον, οι πολιτιστικές και εθνικές διαφορές παίζουν σημαντικό ρόλο.

Σκοπός της μελέτης αποτελεί η μέτρηση της αποτελεσματικότητας, της αποδοτικότητας και της ικανοποίησης των χρηστών από την εμπειρία της αλληλεπίδρασης.<sup>[3][6]</sup>

### **2.1.2 Ο Υπολογιστής**

Ως Υπολογιστή αναφέρεται οποιαδήποτε διεπαφή χρήστη (user interface), δηλαδή το σύνολο των στοιχείων του υπολογιστικού συστήματος με τα οποία ο χρήστης έρχεται σε επαφή και με τα οποία αλληλεπιδρά. Ένα τέτοιο στοιχείο αποτελεί για παράδειγμα, η οθόνη αφής που ανήκει στις μονάδες εισόδου-εξόδου, δηλαδή μέσα παρουσίασης πληροφοριών προς τον χρήστη, αλλά και μέσα με τα οποία ο χρήστης δίνει εντολή προς τη συσκευή, με την εισαγωγή κείμενου ή με την χρήση των δάκτυλων του στην επιφάνεια της οθόνης.

Ο καλός σχεδιασμός της διεπαφής χρήστη των σύγχρονων υπολογιστών αποτελεί σημαντική προϋπόθεση τόσο για την επιτυχή ενσωμάτωσή τους σε παραγωγικές διαδικασίες όσο και για την αποδοχή τους από τους χρήστες τους.

Πιο ειδικά, ο σχεδιαστής των σύγχρονων εφαρμογών και συστημάτων οφείλει να ενσωματώσει τη διεπαφή χρήστη στη σχεδίαση της αλληλεπίδρασης (δηλαδή την επικοινωνία χρήστη-διεπαφής) με τελικό στόχο, η εμπειρία του χρήστη να συνάδει με τις ανάγκες του και τα χαρακτηριστικά του.

Για αυτόν τον λόγο, σήμερα η διαδικασία της σχεδίασης των σύγχρονων διεπαφών χρήστη εντάσσει όχι μόνο τη σχεδίαση της διεπαφής σε αυτή, αλλά και τη συνολική εμπειρία χρήσης.<sup>[3][6]</sup>

### **2.1.3 Η Αλληλεπίδραση**

Υπάρχουν προφανείς διαφορές μεταξύ ανθρώπων και υπολογιστών. Παρόλα αυτά, η ΑΑΥ προσπαθεί να εξασφαλίσει ότι και οι δύο θα αλληλεπιδρούν μεταξύ τους με επιτυχία. Προκειμένου να επιτευχθεί ένα εύχρηστο σύστημα, πρέπει να εφαρμοστεί ό, τι θεωρείται γνωστό για τους ανθρώπους και τους υπολογιστές και να υπάρχει η συμβολή πιθανών χρηστών σε όλη τη διαδικασία σχεδιασμού.

Σε πραγματικά συστήματα, το χρονοδιάγραμμα και ο προϋπολογισμός είναι σημαντικά και είναι ζωτικής σημασίας να βρεθεί μια ισορροπία μεταξύ του τι θα ήταν ιδανικό για τους χρήστες και τι είναι εφικτό στην πραγματικότητα.[6]

Τα πληροφοριακά συστήματα ξεκίνησαν από τα συστήματα ομαδικής επεξεργασίας (batch systems) και οδηγήθηκαν στα συστήματα αλληλεπίδρασης (Interactive systems). Η βασική συνθήκη ήταν πάντα: “Κάθε υπολογιστής που δεν δίνει έξοδο (output), είναι άχρηστος”.[7]

Επίσης, χρειάζεται να παίρνει είσοδο (input). Η ιδέα του “φιλικού προς το χρήστη”(user friendly), ήταν πάντοτε μια βασική αρχή για την ανάπτυξη καλών συστημάτων υπολογιστών. Η σημασία του είναι το πόσο εύκολη είναι για τον άνθρωπο η αλληλεπίδραση με τον υπολογιστή. Αργότερα ο όρος “φιλικό προς το χρήστη” αντικαταστάθηκε από δυο άλλους όρους: τον όρο *χρησιμότητα* (usability) και τον όρο *ευχρηστία* (usability).

Η *χρησιμότητα* αφορά την λειτουργικότητα του συστήματος, κατά πόσο δηλαδή το σύστημα επιτρέπει να πραγματοποιούνται οι εργασίες που σκοπεύει να κάνει ο χρήστης. Με λίγα λόγια αν το σύστημα παρέχει τις σωστές ευκολίες.

Η *ευχρηστία* έχει να κάνει με το πόσο εύκολα ο χρήστης χρησιμοποιεί την λειτουργικότητα. Με άλλα λόγια αν είναι εύκολο να πραγματοποιήσει τα πράγματα που σκοπεύει να κάνει.

Συγκριτικά, η χρησιμότητα είναι ευκολότερο να εκτιμηθεί ενώ η ευχρηστία εξαρτάται από τα προβλήματα που συναντώνται κατά την διάρκεια της χρήσης.

Η αλληλεπίδραση ανθρώπου υπολογιστή είναι ένα παρακλάδι του κλάδου αλληλεπίδρασης ανθρώπου μηχανής - AAM (Human Machine Interaction - HMI).

Η AAM είναι ο κλάδος της επιστήμης που έχει να κάνει με την επίδραση της μηχανής στον άνθρωπο. Είναι μια ευρεία έννοια αφού στον όρο μηχανή περιλαμβάνεται κάθε μηχανή που μπορεί να φανταστεί ο ανθρώπινος νους.



## 2.2 Στόχοι έρευνας της Αλληλεπίδραση Ανθρώπου-Υπολογιστή

Μεγάλο μέρος της έρευνας στον τομέα της ΑΑΥ επιδιώκει να βελτιώσει την αλληλεπίδραση ανθρώπου-υπολογιστή βελτιώνοντας τη ευχρηστία των διεπαφών των υπολογιστών. Ο τρόπος με τον οποίο γίνεται κατανοητή η έννοια ευχρηστία, ο συσχετισμός της με κοινωνικές και πολιτιστικές αξίες, και οι περιπτώσεις κατά τις οποίες ένα συγκεκριμένο χαρακτηριστικό της διεπαφής είναι επιθυμητό συζητείται όλο και περισσότερο στην επιστημονική κοινότητα

Μεγάλο μέρος της έρευνας στον τομέα της αλληλεπίδρασης ανθρώπου-υπολογιστή αφορά:

- Μεθόδους για το σχεδιασμό νέων διεπαφών υπολογιστών. Με αυτό τον τρόπο βελτιστοποιείται η δυνατότητα εκμάθησης και η αποτελεσματικότητα της χρήσης.
- Μεθόδους για την υλοποίηση διεπαφών, π.χ. μέσω βιβλιοθηκών λογισμικού .
- Μεθόδους αξιολόγησης και σύγκρισης των διεπαφών σε σχέση με τη ευχρηστία τους και άλλες επιθυμητές ιδιότητες.
- Μεθόδους για τη μελέτη της χρήσης υπολογιστών από τους ανθρώπους και των κοινωνικοπολιτισμικών τους επιπτώσεων ευρύτερα.
- Μεθόδους οι οποίοι προσδιορίζουν αν ο χρήστης είναι άνθρωπος ή υπολογιστής.
- Μοντέλα και θεωρίες ανθρώπινης χρήσης ηλεκτρονικών υπολογιστών, καθώς και μοντέλα σχετικά με το σχεδιασμό διεπαφών ηλεκτρονικών υπολογιστών.
- Απόψεις οι οποίες αφορούν τις βασικές αρχές που αποτελούν τη βάση του υπολογιστικού σχεδιασμού, της χρήσης υπολογιστών και της πρακτικής έρευνας του ΑΑΥ.

Τα οράματα για το τι επιδιώκουν να επιτύχουν οι ερευνητές στον τομέα αυτόν, ποικίλλουν. Όταν επιδιώκουν μια γνωστική οπτική, οι ερευνητές της ΑΑΥ μπορούν να επιδιώξουν να συσχετίσουν τις διεπαφές των υπολογιστών με τις λειτουργίες που έχουν οι άνθρωποι για τις δραστηριότητές τους. Κάποιες από αυτές είναι η λειτουργία της μνήμης, προσοχής, αντίληψης, αναπαράστασης της γνώσης, σκέψης δημιουργικότητας και επίλυσης προβλημάτων, υποστηρίζοντας ότι έτσι μπορεί να εξηγηθεί η ανθρώπινη συμπεριφορά.

Όταν επιδιώκουν μια μετα-γνωστική οπτική, οι ερευνητές της ΑΑΥ μπορούν να επιδιώξουν την συσχέτιση των διεπαφών με τις υπάρχουσες κοινωνικές συνήθειες ή τις υπάρχουσες κοινωνικοπολιτιστικές αξίες. Σκοπός είναι να διατυπωθούν ομοιότητες και

διαφορές μεταξύ μερικών από τις κύριες «μετα-γνωσιακές» ιδιότητες: γλώσσα / δράση, θεωρία δραστηριότητας και κατανεμημένη γνώση.

Οι ερευνητές της ΑΑΥ ενδιαφέρονται να αναπτύξουν μεθοδολογίες σχεδίασης, να πειραματιστούν με συσκευές, να δημιουργήσουν πρωτότυπο υλικό και λογισμικό σύστημα, να διερευνήσουν παραδείγματα αλληλεπίδρασης και να αναπτύξουν μοντέλα και θεωρίες αλληλεπίδρασης.

## **2.3 Τομείς που συμβάλλουν στην Αλληλεπίδραση Ανθρώπου-Υπολογιστή**

Το πεδίο της ΑΑΥ καλύπτει ένα ευρύ φάσμα θεμάτων και η ανάπτυξη του βασίζεται σε συμβολές πολλών επιστημονικών κλάδων. Μερικοί από τους κύριους κλάδους που συνέβαλαν στο ΑΑΥ είναι:

### **Επιστήμη Υπολογιστών**

- Σχεδιασμός, ανάπτυξη και συντήρηση λογισμικού
- Συστήματα Διαχείρισης Διεπαφών Χρήστη (UIMS) και περιβάλλοντα ανάπτυξης περιβάλλοντος χρήστη (UIDE)
- Εργαλεία πρωτοτύποποίησης
- Γραφικά

### **Γνωστική Ψυχολογία**

- Επεξεργασία πληροφοριών
- Δυνατότητες
- Περιορισμοί
- Συνεργατική εργασία
- Πρόβλεψη Απόδοσης

### **Τεχνητή Νοημοσύνη**

- Έξυπνο λογισμικό

## Κοινωνική Ψυχολογία

- Κοινωνικές και οργανωτικές δομές

## Εργονομία/ Ανθρώπινοι παράγοντες

- Σχεδιασμός υλικού
- Αναγνωσιμότητα οθόνης[6]



Εικόνα 2.1 Πολυεπιστημονικός τομέας της ΑΑΥ[8]

## 2.4 Μοντέλα ανθρώπινης συμπεριφοράς που σχετίζονται με την ευχρηστία

Ένας σημαντικός αριθμός μεγάλων εταιρειών και ακαδημαϊκών ιδρυμάτων μελετά τη ΑΑΥ. Ιστορικά και με κάποιες εξαιρέσεις, οι προγραμματιστές συστημάτων υπολογιστών δεν έχουν δώσει μεγάλη προσοχή στην ευκολία χρήσης του υπολογιστή. Πολλοί χρήστες ηλεκτρονικών υπολογιστών σήμερα θα υποστήριζαν ότι οι κατασκευαστές υπολογιστών εξακολουθούν να μην δίνουν αρκετή προσοχή ώστε να κάνουν τα προϊόντα τους "φιλικά προς το χρήστη".

Ωστόσο, οι προγραμματιστές συστημάτων υπολογιστών υποστηρίζουν ότι η δημιουργία και η σχεδίαση υπολογιστών είναι εξαιρετικά πολύπλοκη και ότι η ζήτηση για τις υπηρεσίες που μπορούν να παρέχουν οι υπολογιστές ξεπερνά πάντα τη ζήτηση για ευκολία χρήσης.

Ένας σημαντικός παράγοντας ΑΑΥ είναι ότι οι διαφορετικοί χρήστες διαμορφώνουν διαφορετικές αντιλήψεις ή διανοητικά μοντέλα για τις αλληλεπιδράσεις τους. Κάθε χρήστης έχει διαφορετικούς τρόπους μάθησης και διατήρησης της γνώσης και των δεξιοτήτων.[4] Χαρακτηριστικό παράδειγμα αποτελεί ο τρόπος μάθησης των μαθητών. Κατά το σχεδιασμό μαθησιακού υλικού υπάρχει συχνά η υπόθεση ότι όλοι οι μαθητές θα επεξεργάζονται και θα οργανώνουν πληροφορίες με παρόμοιο τρόπο.

Ωστόσο, ο τρόπος παρουσίασης πληροφοριών μπορεί να αλληλεπιδράσει διαφορετικά με κάθε μαθητή, ανάλογα τη διάσταση της φρασεογραφίας.

Πιο συγκεκριμένα υποστηρίζεται ότι, σε γενικές γραμμές, κάποιος που προτιμά την γραπτή απεικόνιση πληροφοριών θα μάθει καλύτερα από τις πληροφορίες κειμένου, ενώ κάποιος που προτιμά τις απεικονίσεις θα μάθουν καλύτερα από τις εικονογραφικές πληροφορίες.

Σε κάθε περίπτωση ο φόρτος επεξεργασίας ελαχιστοποιείται εφόσον ο τρόπος παρουσίασης είναι σύμφωνος με την εκάστοτε συνήθη λειτουργία του ατόμου. [11]

### 2.4.1 Το μοντέλο του ανθρώπινου επεξεργαστή

Γνωστική συμπεριφορά είναι η συνήθεια του ατόμου ή προτιμώμενος τρόπος επεξεργασίας πληροφοριών και θεωρείται ως ένας τρόπος που το άτομο μαθαίνει, σκέφτεται, αντιλαμβάνεται και συμπεριφέρεται [10]

Η αρχική εστίαση στην μελέτη της ΑΑΥ λοιπόν, ήταν σε μεγάλο βαθμό γνωστική, εστιάζοντας στις αιτιώδεις σχέσεις μεταξύ των χαρακτηριστικών διεπαφής χρήστη και της ανθρώπινης απόδοσης, αλλά με διαφορετικές απόψεις για το πώς θα αλληλεπιδρούσαν μεταξύ τους. Εάν τα ανθρώπινα γνωστικά χαρακτηριστικά είναι σταθερά και καθολικά, δηλαδή αν όλη η ομάδα χρηστών αντιλαμβάνεται την γνώση με τα ίδια χαρακτηριστικά και αφομοίωναν την πληροφορία με τον ίδιο τρόπο, τότε τα χαρακτηριστικά διεπαφής χρήστη μπορούν να είναι εγγενώς χρήσιμα ή άχρηστα, καθιστώντας την ευχρηστία μια εγγενή δυαδική ιδιότητα του διαδραστικού λογισμικού, δηλαδή ένα διαδραστικό σύστημα απλά είναι ή δεν είναι εύχρηστο.

Ωστόσο, εάν οι ανθρώπινες γνωστικές ιδιότητες ποικίλλουν όχι μόνο μεταξύ ατόμων, αλλά και σε διαφορετικές συνθήκες, τότε η ευχρηστία γίνεται μια αναδυόμενη ιδιότητα που εξαρτάται όχι μόνο από τα χαρακτηριστικά και τις ιδιότητες ενός αλληλεπιδραστικού συστήματος αλλά και από το ποιος το χρησιμοποίησε και από το τι προσπάθησε να κάνει με αυτό. Η τελευταία αυτή θέση ενισχύθηκε σημαντικά στη δεκαετία του 1990 και παρουσιάζει ένα θεωρητικό μοντέλο, σύμφωνα με το οποίο ο άνθρωπος επεξεργαστής περιλαμβάνει:

- Αισθητηριακό σύστημα (perceptual system): επιτρέπει την κατανόηση αισθητήριων ερεθισμάτων από τον εξωτερικό κόσμο και περιλαμβάνει αισθητηριακά υποσυστήματα.
- Σύστημα κατανόησης (cognitive system): κάνει την απαραίτητη επεξεργασία για συσχετίσεις εννοιών.
- Κινητικό σύστημα (motor system): ελέγχει κινήσεις του σώματος (δράσεις και αντιδράσεις)

Η χρησιμότητα του παραπάνω μοντέλου δεν καλύπτει μια συνολική θεωρία που αφορά κάθε μορφή ΑΑΥ (Παρόλο που σε αυτό αποσκοπούσε). Υπάρχουν αμέτρητοι παράγοντες που μπορούν να επηρεάσουν τη χρήση ενός συστήματος όπως είναι για παράδειγμα η αντίληψή

του χρήστη, ή η συναισθηματική και ψυχολογική του κατάσταση. Επίσης, ο χρόνος αντίδρασής του σε κάποιο ερέθισμα επηρεάζεται από το περιβάλλον.

Επιπλέον, οι σχεδιαστές και αξιολογητές των διαδραστικών συστημάτων υπογραμμίζουν πως τα λάθη αποτελούν χαρακτηριστικό της ανθρώπινης φύσης και συχνά συμβαίνουν ανεξαρτήτως της ανθρώπινης εμπειρίας ή ευφυΐας (παράγοντες που μπορούν να προκαλέσουν λάθη μπορεί να είναι η λήθη, η κούραση, η σωματική διάπλαση, η ηλικία, οι εξωτερικοί θόρυβοι, η απόσπαση της προσοχής, κ.ά.). Ωστόσο, δεν κάνουν μόνο οι χρήστες λάθη, αλλά και οι σχεδιαστές. Επομένως, πρέπει να αποδεχτούμε ως δεδομένο ότι θα προκληθούν λάθη, ακόμα και για τα πιο απλά πράγματα.[2][3]

Τέλος, οι πολιτιστικές και εθνικές διαφορές παίζουν εξίσου σημαντικό ρόλο. Ένας άλλος παράγοντας που απαιτεί μελέτη της ΑΑΥ είναι ότι η τεχνολογία διεπαφής χρήστη αλλάζει ταχύτατα, προσφέροντας νέες δυνατότητες αλληλεπίδρασης οι οποίες, ενδεχομένως να μην συνάδουν με τα προηγούμενα ερευνητικά ευρήματα. Καθώς επίσης παρατηρείται πως οι προτιμήσεις των χρηστών αλλάζουν καθώς αλληλεπιδρούν με νέες διεπαφές.

## **2.5 Η Ιστορία και οι Στόχοι της Αλληλεπίδρασης Ανθρώπου Υπολογιστή**

Για τις τρεις πρώτες δεκαετίες της πληροφορικής, σχεδόν όλοι οι χρήστες ήταν εξειδικευμένοι στον ακριβό συγκεντρωτικό εξοπλισμό. Μια τάση προς λιγότερο εκπαιδευμένους χρήστες άρχισε τη δεκαετία του 1960 με την εισαγωγή της χρονομεριστικής μίσθωσης και των μικρών υπολογιστών. Με τη χρήση υπολογιστών τη δεκαετία του '80, οι χρήστες ηλεκτρονικών υπολογιστών είχαν μόνο, ή και καθόλου, βασική κατάρτιση σε λειτουργικά συστήματα και εφαρμογές λογισμικού.

Ωστόσο, οι πρακτικές σχεδιασμού λογισμικού συνέχισαν να υποθέτουν έμμεσα πως οι χρηστές ήταν έμπειροι και ικανοί, και ήταν εξοικειωμένοι με τα τεχνικά λεξιλόγια και τις αρχιτεκτονικές συστημάτων και επίσης είχαν την ικανότητα να επιλύουν προβλήματα που προκύπτουν από τη χρήση του υπολογιστή. Αυτές οι υποθέσεις γρήγορα δεν έγιναν αποδεκτές. Για τον τυπικό χρήστη, ο διαδραστικός υπολογισμός συσχετίστηκε με συνεχείς απογοητεύσεις και επακόλουθες ανησυχίες. Οι υπολογιστές ήταν προφανώς πολύ δύσκολο να χρησιμοποιηθούν για τους περισσότερους χρήστες και συχνά απολύτως άχρηστοι. [9]

Η ΑΑΥ και η ευχρηστία έχουν τις ρίζες τους στις πτωτικές τιμές των υπολογιστών τη δεκαετία του 1980, όταν για πρώτη φορά ήταν εφικτό για πολλούς υπαλλήλους να έχουν δικό τους προσωπικό υπολογιστή (γνωστός και ως PC).

Η **ευχρηστία** έγινε έτσι ένας βασικός στόχος για το σχεδιασμό οποιουδήποτε διαδραστικού λογισμικού που δεν θα χρησιμοποιηθεί από εκπαιδευμένους τεχνικούς υπολογιστών. Δημοφιλείς όροι όπως "φιλικό προς το χρήστη" εισήχθησαν καθημερινά. Τόσο η ευχρηστία όσο και η φιλικότητα προς το χρήστη αντιλήφθηκαν αρχικά ως ιδιότητα διαδραστικού λογισμικού. Το λογισμικό ήταν είτε εύχρηστο είτε όχι. Το άχρηστο λογισμικό θα μπορούσε να γίνει χρήσιμο μέσω του επανασχεδιασμού.

### **2.5.1 Οι στόχοι της Αλληλεπίδρασης Ανθρώπου-Υπολογιστή**

Οι στόχοι της ΑΑΥ είναι να παράγει χρήσιμα και ασφαλή συστήματα, καθώς και λειτουργικά συστήματα. Προκειμένου να παράγουν συστήματα ηλεκτρονικών υπολογιστών με ευχρηστία, οι προγραμματιστές πρέπει να προσπαθούν:

- Να κατανοήσουν τους παράγοντες που καθορίζουν τον τρόπο με τον οποίο οι άνθρωποι χρησιμοποιούν την τεχνολογία.
- Να αναπτύσσουν εργαλεία και τεχνικές που επιτρέπουν την κατασκευή συστημάτων κατάλληλων για χρήση.
- Να επιτεύξουν αποδοτική, αποτελεσματική και ασφαλή αλληλεπίδραση.
- Να θέτουν ως προτεραιότητα τους ανθρώπους.

Η βασική αρχή της ΑΑΥ είναι η άποψη πως ως προτεραιότητα τίθενται οι άνθρωποι που χρησιμοποιούν ένα σύστημα ηλεκτρονικών υπολογιστών. Οι ανάγκες, οι δυνατότητές και οι προτιμήσεις των χρηστών για τη διεξαγωγή διαφόρων εργασιών πρέπει να κατευθύνουν τους προγραμματιστές για τον τρόπο που σχεδιάζουν τα συστήματα.

Οι άνθρωποι δεν πρέπει να αλλάζουν τον τρόπο με τον οποίο χρησιμοποιούν ένα σύστημα για να ταιριάζουν με αυτό. Αντ' αυτού, το σύστημα θα πρέπει να σχεδιαστεί έτσι ώστε να ανταποκρίνεται στις απαιτήσεις τους.[6]

## 2.6 Η Ευχρηστία

Η ευχρηστία (usability), που στα Ελληνικά συχνά αποδίδεται και ως ευκολία στη χρήση, είναι μία από τις βασικές έννοιες της ΑΑΥ. Με τον όρο ευχρηστία αναφέρεται η ευκολία εκμάθησης και χρήσης ενός διαδραστικού συστήματος από τον άνθρωπο, ενώ ο έλεγχος ευχρηστίας κατά κύριο λόγο, γίνεται κάποιας μορφής δοκιμών της χρήσης. Επιπλέον υπογραμμίζει το γεγονός πως η ευχρηστία είναι σημαντική προϋπόθεση για την αποδοχή του συστήματος από τον χρήστη.

Έχουν αποδοθεί αρκετοί ορισμοί για τον όρο ευχρηστία καθώς κάποιοι θεωρούν ότι η ευχρηστία είναι μια σύνθετη έννοια, δηλαδή περιλαμβάνει και άλλους παράγοντες.

Το πρότυπο ISO 9241 ορίζει την έννοια της ευχρηστίας ως την «έκταση στην οποία ένα προϊόν μπορεί να χρησιμοποιηθεί από προσδιορισμένους χρήστες ώστε να πετύχουν συγκεκριμένους στόχους με αποτελεσματικότητα, αποδοτικότητα και ικανοποίηση, σε συγκεκριμένο πλαίσιο χρήσης».[40]

Με απλά λόγια, η ευχρηστία δεν σημαίνει τίποτε άλλο από το να είναι ο χρήστης σε θέση να χρησιμοποιήσει γρήγορα και εύκολα, αποτελεσματικά και αποδοτικά ένα σύστημα,. Σε επίπεδο συστημάτων Η/Υ, η ευχρηστία έχει αναδειχθεί ως σημαντικός στόχος σχεδίασης και αξιολόγησής τους και η σημασία της μεγαλώνει όσο η χρήση των συστημάτων γίνεται συχνότερη στην καθημερινότητα μας. [2]

Ένα εύχρηστο σύστημα είναι:

- Εύκολο να το μάθει ο χρήστης (εύκολο να κατανοηθεί)
- Εύκολο να θυμάται ο χρήστης πώς να το χρησιμοποιεί
- Αποτελεσματικό στην χρήση
- Αποδοτικό στη χρήση
- Ασφαλές στη χρήση
- Ευχάριστο στην χρήση[6]



### 2.6.1 Αξιολόγηση Ευχρηστίας

Η αξιολόγηση ευχρηστίας ορίζεται ως «η μέτρηση των ιδιοτήτων της ευχρηστίας (ή του προσδιορισμού πιθανών προβλημάτων) ενός συστήματος ή μιας εφαρμογής, σε σχέση με συγκεκριμένους χρήστες, που εκτελούν συγκεκριμένες διεργασίες, σε προκαθορισμένα πλαίσια».[41]

Σήμερα, η ευχρηστία αποτελεί έναν από τους βασικούς παράγοντες αξιολόγησης οποιουδήποτε διαδραστικού συστήματος. Άλλωστε, υπάρχουν πλέον συγκεκριμένες μέθοδοι για την αξιολόγηση της.

Είναι ιδιαίτερη η σημαντικότητα της αξιολόγησης της ευχρηστίας γιατί η προσέγγιση της στηρίζεται στην συμμετοχή των χρηστών κατά την διαδικασία, σε αντίθεση με άλλες προϋποθέσεις αποδοχής των διαδραστικών συστημάτων. Οι μέθοδοι αξιολόγησης της ευχρηστίας κρίνονται απαραίτητες για τον έλεγχο της ποιότητας όλων των διαδραστικών συστημάτων και η εφαρμογή των μεθόδων αυτών αποτελεί πλέον καθιερωμένη πρακτική στην ανάπτυξη συστημάτων πληροφορικής και γενικά διαδραστικών προϊόντων.

#### Μέθοδοι αξιολόγησης ευχρηστίας (usability evaluation methods):

Η θεωρητική αντιμετώπιση της ευχρηστίας περιλαμβάνει τα εξής:

- “Νόμους” για τις δυνατότητες ανθρώπινης επεξεργασίας και την ανθρώπινη συμπεριφορά, οι οποίοι επηρεάζουν τον σχεδιασμό συστημάτων και την ευχρηστία τους, και οι οποίοι έχουν προκύψει από πειραματικές διαδικασίες και παρατήρηση.
- Αρχές ευχρηστίας (usability principles): Αποτελούν βασικές «αλήθειες», ή «υποθέσεις» για την ισχύ της ευχρηστίας.
- Οδηγίες ευχρηστίας (usability guidelines): Αποτελούν δηλώσεις που συγκεκριμενοποιούν τις αρχές για την υποστήριξη της σχεδίασης και αξιολόγησης των διαδραστικών συστημάτων
- Πρότυπα (standards): Ορίζουν με συστηματικό τρόπο αρχές, οδηγίες, μεθόδους και κριτήρια αξιολόγησης.[2]

Η πρακτική αντιμετώπιση της ευχρηστίας περιλαμβάνει τα εξής:

### **Επιθεωρήσεις ευχρηστίας**

Η επιθεώρηση ευχρηστίας πραγματοποιείται κατά βάση από τους ειδικούς ευχρηστίας. Επειδή συνήθως δεν συμμετέχουν χρήστες, οι επιθεωρήσεις ευχρηστίας αποκαλούνται και ως μια προσέγγιση προβλεπτικής αξιολόγησης ευχρηστίας και αφορά την:

- Ευρετική αξιολόγηση. Είναι μια συστηματική και ποιοτική μέθοδος επιθεώρησης ευχρηστίας, όπου ειδικοί ευχρηστίας κρίνουν αν η διεπαφή ακολουθεί καθιερωμένες και κοινά αποδεκτές αρχές.
- Γνωστική περιδιάβαση. Το σημαντικό ζητούμενο που απαιτείται σε μια γνωστική περιδιάβαση είναι να προσδιοριστεί το κατά πόσον οι χρήστες θα μπορέσουν να μάθουν να χρησιμοποιούν το σύστημα μέσα από την εξερεύνηση τους από τον ίδιους.<sup>[2]</sup>

### **Δοκιμές ευχρηστίας**

Οι δοκιμές ευχρηστίας αποτελούν κάποιου είδους πειράματος στο οποίο συμμετέχουν χρήστες, γι' αυτό τα θέματα των μεθόδων αξιολόγησης που προκύπτουν είναι πολλά και πολύπλοκα. Ταυτόχρονα υπάρχουν και σημαντικά ηθικά θέματα όπως η αξιοπιστία (reliability) και η εγκυρότητα (validity). Οι δοκιμές ευχρηστίας, αναλόγως τον σκοπό τους, διακρίνονται σε διαμορφωτικές και συμπερασματικές.

- Οι διαμορφωτικές δοκιμές ευχρηστίας (formative usability tests) έχουν ως σκοπό τη συγκέντρωση χρήσιμων σχολίων και διορθώσεων των χρηστών, δηλαδή ευρημάτων ευχρηστίας επί ενός, ίσως μη ολοκληρωμένου, πρωτοτύπου του συστήματος.
- Οι συμπερασματικές δοκιμές ευχρηστίας (summative usability tests) έχουν ως σκοπό την εξαγωγή συμπερασμάτων για το εάν το σύστημα είναι αποδεκτό ή όχι. Το σύστημα θα πρέπει να είναι πλήρως λειτουργικό/<sup>[2]</sup>

### **Ερωματολογία διερεύνησης της ικανοποίησης των χρηστών.**

Η μέθοδος των ερωματολογίων αφορά την καταγραφή των αντιδράσεων των χρηστών. Τα συμπεράσματα που προκύπτουν αφορούν την αντίληψη των χρηστών για την ευχρηστία. Ο τύπος που γίνεται η διαδικασία του ερωματολογίου δεν είναι συγκεκριμένη. Επίσης,

διαδικασία μπορεί να συμβεί οποιαδήποτε χρονική στιγμή θεωρήσουν οι αξιολόγησες πως είναι κατάλληλη για την εξαγωγή χρήσιμων δεδομένων. Ως προς την χρονική περίοδο, οι επιλογές είναι τρεις: (α) κατά την εκτελεί εργασιών, (β) στο τέλος κάθε εργασίας (γ) στο τέλος της διαδικασίας δοκιμής.

Τα ευρήματα ευχρηστίας, που καλούνται συχνά και ως προβλήματα ευχρηστίας είναι το βασικό συμπέρασμα όχι μόνο της ευρετικής αξιολόγησης, αλλά και των δοκιμών ευχρηστίας. Πρόκειται για δηλώσεις που προσδιορίζουν κάποια δυσκολία ή κάποιο πρόβλημα που αντιμετωπίζουν οι χρήστες κατά την αλληλεπίδραση τους με το σύστημα.

Τα ευρήματα ευχρηστίας μπορεί να είναι πολύ συγκεκριμένα ή πιο αφηρημένα. Μπορεί να είναι αληθή ή ψευδή και απαιτείται να ιεραρχηθούν, όχι μόνο σε σχέση με τις ευρετικές αρχές, αλλά κυρίως με βάση τη σημαντικότητά τους για την εμπειρία του χρήστη.

Υπάρχουν διάφορα ερωτήματα ως κριτήρια, από τα οποία τα πιο σημαντικά είναι τα εξής:

- Πόσο σημαντικό θεωρείται το εύρημα από τους ίδιους τους χρήστες;
- Πόσοι χρήστες επηρεάζονται από αυτό;
- Πόσο σύντομα μπορεί να διορθωθεί<sup>[2]</sup>

## **2.6.2 Συστήματα καταγραφής της συμπεριφοράς χρηστών**

Για να επιτευχθεί η αξιολόγηση της ευχρηστίας ενός συστήματος κρίνεται απαραίτητο το πλήθος πληροφοριών που αφορούν την συμπεριφορά του χρήστη. Όσο περισσότερη πληροφορία διατίθεται τόσο πιο αντικειμενικά μπορεί να είναι τα συμπεράσματα που προκύπτουν.

Η καταγραφή της συμπεριφοράς του χρήστη μπορεί να επιτευχθεί με την συλλογή των πληροφοριών που αναφέρονται στον τρόπο που διαχειρίζονται μια διεπαφή. Ο τρόπος χρήσης των περιφερειακών συσκευών όπως τα “κλικς” του ποντικιού, ο χρόνος ολοκλήρωσης των ενεργειών, τα λάθη, ο τρόπος περιήγησης του χρήστη στο σύστημα αποτελούν χαρακτηριστικά παραδείγματα της συμπεριφοράς του χρήστη. Η δημιουργία συστημάτων που θα καταγράφουν αυτή την συμπεριφορά διευκολύνουν την διαδικασία της αξιολόγησης ποσό μάλλον όταν η καταγραφή γίνεται με αυτοματοποιημένο τρόπο. Τα συστήματα αυτοματοποιημένης καταγραφής δεδομένων συλλέγουν δεδομένα διαρκώς, ανανεώνοντας έτσι συνεχώς τα συμπεράσματα που προκύπτουν. Επιπλέον επιτρέπουν την αξιολόγηση σε πραγματικές συνθήκες εργασίας των χρηστών καθιστώντας τα αποτελέσματα της αξιολόγησης πιο αξιόπιστα με μειωμένο κόπο και κόστος. Παρόλα αυτά απαιτείται σημαντική προετοιμασία και διαδικτυακή υποδομή.

### 2.6.3 Από την ευχρηστία στην εμπειρία χρήστη.

Με την έρευνα επικεντρωμένη όλο και περισσότερο στην προσαρμογή μεταξύ του διαδραστικού λογισμικού και του περιβάλλοντος χρήσης του, η ευχρηστία έπαψε να είναι η κυρίαρχη ιδέα της ΑΑΥ. Η **ποιότητα στη χρήση** δεν φαίνεται πλέον να είναι ένα απλό ζήτημα για το πόσο εύχρηστο ήταν ένα διαδραστικό σύστημα εκ φύσεως, αλλά πόσο καλά προσαρμόστηκε στο περιβάλλον χρήσης του.

Η ποιότητα κατά τη χρήση έγινε ένας δημοφιλέστερος εναλλακτικός όρος χρήσης σε διεθνή πρότυπα, καθώς αποφεύγεται η ευχρηστία να είναι μια απόλυτη αμετάβλητη ιδιότητα ενός αλληλεπιδραστικού συστήματος. Γύρω στα τέλη του αιώνα, η άνοδος των δικτυωμένων ψηφιακών μέσων (π.χ. διαδίκτυο, κινητά, διακρατικές τηλεοράσεις, δημόσιες εγκαταστάσεις) προσέθεσαν νέες ανησυχίες για την ΑΑΥ, δημιουργώντας έναν ακόμη πιο ελκυστικό όρο από τη ευχρηστία: **την εμπειρία του χρήστη**.

Οι τρέχουσες αντιλήψεις ευχρηστίας σήμερα είναι διαφορετικές από τις αυτές της ΑΑΥ στη δεκαετία του '80. Από τότε, η **ευκολία στη χρήση** έχει βελτιωθεί, τόσο και η σχεδίαση των αλληλεπιδράσεων όσο και τα βελτιωμένα επίπεδα γλωσσομάθειας σε μεγάλο μέρος του πληθυσμού των προηγμένων οικονομιών. Αυτό σημαίνει πως η εξοικείωση με τις βασικές λειτουργίες των ηλεκτρονικών υπολογιστών είναι πλέον ευρέως διαδεδομένη. Η ευχρηστία δεν είναι πλέον η κυρίαρχη ανησυχία στο σχεδιασμό αλληλεπίδρασης. Παραμένει όμως σημαντική, με τις απογοητευτικές εμπειρίες των δύσκολων στη χρήση ψηφιακών τεχνολογιών να είναι ακόμη συνηθισμένες. [12]

Η κακή ευχρηστία λοιπόν, εξακολουθεί να είναι μαζί μας, αλλά έχουμε προχωρήσει.

## 2.7 Η Χρηστικότητα

Όπως αναφέρθηκε παραπάνω, η χρηστικότητα (usability) αφορά την λειτουργικότητα του συστήματος, κατά πόσο δηλαδή το σύστημα επιτρέπει να πραγματοποιούνται οι εργασίες που σκοπεύει να κάνει ο χρήστης. Με λίγα λόγια αν το σύστημα παρέχει τις σωστές ευκολίες.

Η στενή αλληλεπίδραση με την κοινότητα των χρηστών οδηγεί στη δημιουργία ενός καλά επιλεγμένου συνόλου εργασιών συγκριτικής αξιολόγησης που αποτελεί την βάση για τους στόχους και τα μετρά της χρηστικότητας.[13]

### 2.7.1 Αξιολόγηση της χρηστικότητας

Για κάθε τύπο και κάθε εργασία ακριβείς μετρήσιμοι αντικειμενικοί στόχοι καθοδηγούν το σχεδιαστή κατά την διάρκεια της διαδικασίας ελέγχου. Το πρότυπο ISO 9241 εργονομία επικοινωνίας Ανθρώπου-συστήματος εστιάζει στους αξιοθαύμαστους στόχους της αποτελεσματικότητας της αποδοτικότητας και της ικανοποίησης.[40] Όμως τα ακόλουθα μέτρα χρηστικότητας, που εστιάζουν στους τελευταίους δυο στόχους οδηγούν πιο άμεσα σε πρακτική αξιολόγηση:

1. Χρόνος εκμάθησης (Time to learn). Πόσο χρόνο χρειάζονται τα συνήθη μέρη της οομάδας των χρηστών για να μάθουν με ποιον τρόπο πρέπει να χρησιμοποιούν τις ενέργειες που έχουν σχέση με κάποιο σύνολο εργασιών:
2. Ταχύτητα απόδοσης (Speed of performance). Πόσος χρόνος χρειάζεται για την εκτέλεση των εργασιών συγκριτικής αξιολόγησης.
3. Ρυθμός σφαλμάτων των χρηστών(Rate of errors by users). Σε πόσα και σε τι είδους λάθη υποπίπτουν οι άνθρωποι όταν εκτελούν εργασίες συγκριτικής αξιολόγησης;  
Ο χρόνος που χρειάζεται για την διενέργεια και την διόρθωση των σφαλμάτων μπορεί να ενσωματωθεί στην ταχύτητα της απόδοσης. Ο χειρισμός των λαθών είναι ένα τόσο κρίσιμο στοιχείο της χρήσης της διεπαφής, που χρειάζεται εκτεταμένη μελέτη.
4. Απομνημόνευση μέσα στον χρόνο(Retention over time). Πόσο καλά διατηρούν οι χρηστές τη γνώση τους μετά από μια ώρα, μια μέρα ή μια εβδομάδα; Η απομνημόνευση μπορεί να συνδέεται στενά με τον χρόνο εκμάθησης ενώ η συχνότητα χρήσης παίζει σημαντικό ρόλο.
5. Υποκειμενική ικανοποίηση (Subjective satisfaction). Σε τι βαθμό ικανοποίησε τους χρηστές, με βάση διαφόρους παράγοντες, η διεπαφή; Η απάντηση μπορεί να εξακριβωθεί μέσω συνεντεύξεων η γραπτών εκθέσεων που περιλαμβάνουν κλίμακες ικανοποίησης και διαθέτουν χώρο για τη συμπλήρωση σχόλιων.[13]

## 2.7.2 Γιατί είναι σημαντική η χρηστικότητα;

Πολλά καθημερινά συστήματα και προϊόντα φαίνεται να έχουν σχεδιαστεί με ελάχιστη προσοχή στη χρηστικότητα. Αυτό οδηγεί σε απογοήτευση, σπατάλη χρόνου και λάθη.

Παραδείγματα διαδραστικών προϊόντων αποτελούν: κινητό τηλέφωνο, ηλεκτρονικός υπολογιστής, τηλεχειριστήριο, αυτόματος πωλητής, μηχανή καφέ, ATM, μηχανή εισιτηρίων, βιβλιοθήκη, web, φωτοαντιγραφικό, ρολόι, εκτυπωτής, στερεοφωνικό κ.α. *Πόσα είναι πραγματικά εύκολα, αβίαστα και ευχάριστα στη χρήση;*

Για παράδειγμα, ένα φωτοτυπικό μπορεί να έχει κουμπιά όπως αυτά στον πίνακα ελέγχου του. Τοποθετείται ένα έγγραφο στο φωτοτυπικό μηχάνημα και έχει ρυθμιστεί το φωτοτυπικό για να δημιουργήσει 15 αντίγραφα, ταξινομημένα και συναφθέντα. Στη συνέχεια ο χρήστης πιέζει το κουμπί με το "C" για να ξεκινήσει η διαδικασία των αντιγράφων.

*Τι έχει συμβεί;*

Ενώ ο χρήστης περιμένει πως το φωτοαντιγραφικό θα κάνει αντίγραφα σωστά, οι ρυθμίσεις φωτοτυπικού μηχανήματος διαγράφονται και δεν εκδίδονται αντίγραφα. Το κουμπί φωτοτύπησης αντιγράφου είναι στην πραγματικότητα το κουμπί με σύμβολο το ρόμβο. Αυτό το σύμβολο χρησιμοποιείται ευρέως σε φωτοτυπικά μηχανήματα, αλλά δεν βοηθά πολύ κάποιον που δεν είναι εξοικειωμένος με αυτό. [6]



Εικόνα 4.2 Χρηστικότητα εκτυπωτή [14]

## Κεφάλαιο 3

### Οι NoSQL Βάσεις Δεδομένων

Οι SQL βάσεις δεδομένων που ονομάζονται κυρίως σχεσιακές (relational) βάσεις δεδομένων (RDBMS). Διαχειρίζονται τα δεδομένα σε μορφή πίνακα που απαρτίζεται από συγκεκριμένο πλήθος γραμμών σε σχήμα πίνακα και έχουν ορισμένο σχήμα. Αυτό σημαίνει πως η μεταβολή της δομής και της ποσότητας των δεδομένων τους και δεν είναι ευέλικτη.

Οι SQL βάση δεδομένων διαθέτουν δομημένη γλώσσα ερωτημάτων (Structured Query Language) για την διαχείριση των δεδομένων, η οποία είναι πολύ ισχυρή. Οι σχεσιακές βάσεις δεδομένων είναι κατάλληλες για την διαχείριση περίπλοκων ερωτημάτων και με πολύπλοκες επεμβάσεις στα δεδομένα ωστόσο, δεν αποτελεί την πλέον ικανοποιητική για ιεραρχική αποθήκευση δεδομένων.

Κατά κανόνα, οι SQL βάσεις δεδομένων ακολουθούν κατακόρυφη κλιμάκωση. Η αύξηση του όγκου δεδομένων μπορεί να διαχειριστεί αυξάνοντας την CPU, RAM, SSD κ.α., σε έναν μονάχα διακομιστή, καθιστώντας την κλιμάκωση αρκετά ακριβή.

Οι SQL βάσεις δεδομένων είναι σταθερές και συμμορφώνονται στις ιδιότητες ACID (Ατομικότητα, Συνάφεια, Απομόνωση και Ανθεκτικότητα), γι'αυτό είναι οι πλέον κατάλληλες για εφαρμογές που αφορά καθήκοντα συναλλαγής, αφού προσφέρουν τις ιδιότητες ατομικότητας και ακεραιότητας των δεδομένων .

Όλες οι SQL βάσεις δεδομένων παρέχουν εξαιρετική υποστήριξη από τους προμηθευτές τους. Υπάρχουν επίσης, πολλές ανεξάρτητες πηγές που μπορεί να συμβουλευτεί ο χρήστης για να αποκτήσει εξοικείωση με τις SQL βάσεις δεδομένων για εφαρμογές πολύ μεγάλης κλίμακας.

Τα ελαττώματα του σχεσιακού μοντέλου οδήγησαν στην ανάγκη για μελέτη και σχεδίαση ενός πιο ευέλικτου μοντέλου το οποίο αναφέρεται ως NoSQL (Not Only SQL - Όχι μόνο SQL)

#### 3.1 Οι NoSQL Βάσεις Δεδομένων

Ο όρος NoSQL χρησιμοποιείται για να χαρακτηρίσει μία ομάδα συστημάτων βάσεων δεδομένων που το κύριο χαρακτηριστικό του είναι η μη τήρηση του μοντέλου RDBMS (Relational DataBase Management System ) το οποίο αφορά τα παραδοσιακά συστήματα σχεσιακών βάσεων δεδομένων. Η NoSQL βάση δεδομένων δεν είναι

κατάλληλη για περίπλοκα ερωτήματα, καθώς δεν έχει συγκεκριμένες διεπαφές για την εκτέλεση τέτοιων ερωτημάτων. Συγκριτικά, τα ίδια ερωτήματα, δεν είναι το ίδιο ισχυρά στη NoSQL όσο σε γλώσσα ερωτημάτων SQL

Η NoSQL βάση δεδομένων ωστόσο, είναι καταλληλότερη για την ιεραρχική αποθήκευση δεδομένων μιας και η αποθήκευση των δεδομένων πραγματοποιείται σε ζευγάρια κλειδιών-τιμών για την αποθήκευση δεδομένων που μοιάζουν με τα JSON δεδομένα. Η NoSQL βάση δεδομένων ενδείκνυται για την διαχείριση μεγάλου όγκου δεδομένων.

Οι μη σχεσιακές βάσεις δεδομένων ακολουθούν οριζόντια κλιμάκωση, δηλαδή η μεγάλη κίνηση δεδομένων, μπορεί να διαχειριστεί εύκολα αυξάνοντας ακόμα περισσότερο τους διακομιστές στην υποδομή της NoSQL βάσης δεδομένων ελαχιστοποιώντας το κόστος.

Παρ'όλο που μπορεί να χρησιμοποιηθεί η NoSQL βάση δεδομένων για σκοπούς συναλλαγής, συνεχίζει να μην μπορεί να συγκριθεί με τις ιδιότητες των σχεσιακών βάσεων σε επίπεδο πολύπλοκων συναλλαγών για μεγάλο όγκο δεδομένων.

Βάσεις δεδομένων NoSQL χρησιμοποιούνται όλο και περισσότερο σε εφαρμογές μεγάλου όγκου δεδομένων (big data) και σε πραγματικό χρόνο (real time).

Για κάποιες NoSQL βάσεις δεδομένων η υποστήριξη από ειδήμονες στην εγκατάσταση και την ανάπτυξη NoSQL εφαρμογών μεγάλης κλίμακας είναι ακόμα σε πρώιμο στάδιο

[24]

### 3.2 Οι ιδιότητες ACID, BASE και το θεώρημα CAP

Ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων (RDBMS) πρέπει να εξασφαλίζει τις παρακάτω ιδιότητες **ACID**:

**Ατομικότητα** (Atomicity) : Πρέπει να εκτελεστούν είτε όλες οι ενέργειες της συναλλαγής είτε καμία.

**Συνέπεια** (Consistency) : Η εκτέλεση μιας συναλλαγής θα πρέπει να διατηρεί τη συνέπεια της βάσης δεδομένων

**Απομόνωση** (Isolation) : Κάθε συναλλαγή πρέπει να θεωρείται ότι τρέχει μόνη της δηλαδή Μια συναλλαγή δεν μπορεί να επηρεάσει την άλλη



**Μονιμότητα (Durability)** : Αν μια συναλλαγή ολοκληρωθεί με επιτυχία, οι αλλαγές που έχει κάνει θα πρέπει να παραμένουν στη βάση δεδομένων, ακόμα και αν το σύστημα αποτύχει

Μπορεί να φαίνονται αρκετά απαραίτητες αυτές οι ιδιότητες, δεν είναι συμβατές με τις διαδικτυακές εφαρμογές όσο αφορά την απόδοση.

Καθώς ο όγκος δεδομένων, των ερωτημάτων αλλά και των χρηστών γίνεται μεγάλος, αρχίζουν να παρουσιάζουν αδυναμίες στις επιδόσεις τους. Οι αυξημένες απαιτήσεις αύξησαν την πολυπλοκότητα και το κόστος καθώς επίσης σε αρκετές περιπτώσεις η δομή των δεδομένων δεν ταιριάζει με την δομή των πινάκων των σχεσιακών βάσεων δεδομένων.

Ένα NoSQL σύστημα βάσης δεδομένων υποστηρίζει τις λεγόμενες **BASE** ιδιότητες στις συναλλαγές που πραγματοποιούνται στη βάση

**Basic Availability**, σημαίνει ότι ανά πάσα στιγμή το σύστημα είναι διαθέσιμο αλλά όχι κατ' ανάγκη όλα τα στοιχεία του

**Soft-state**, σημαίνει ότι τα δεδομένα δεν είναι πάντα αποθηκευμένα σε δίσκο, αλλά αν χρειαστεί μπορούν να ξαναδημιουργηθούν παρέχουν εγγυήσεις μονιμότητας στα δεδομένα χωρίς να επηρεάζουν σημαντικά την απόδοση του συστήματος. Μια διαφορετική ερμηνεία είναι ότι το σύστημα μπορεί να αλλάξει χωρίς κάποια παρεμβολή από τον χρήστη.

**Eventual-consistency**, σημαίνει ότι κατά βάση, δεν υπάρχει ανά πάσα στιγμή συνέπεια στα δεδομένα αλλά εν τέλει.

Ο όρος BASE είναι επινοημένος από τον Eric Brewer και επειδή είναι γενικός, κάθε σύστημα θα πρέπει να εξετάζεται ξεχωριστά για τις ιδιότητες που έχει και τις δυνατότητες που δίνει. [17]

Ο Eric Brewer παρατήρησε επίσης ότι είναι αδύνατο για ένα κατακευματισμένο σύστημα ηλεκτρονικών υπολογιστών να παρέχει συγχρόνως συνέπεια, διαθεσιμότητα και ανοχή στις κατατμήσεις. Αυτό συνηθέστερα αναφέρεται ως θεώρημα CAP.

( Consistency - Availability – Partitions Tolerance) [42]

Η CouchDB αποτελεί ένα παράδειγμα NoSQL βάσης δεδομένων που παρέχει ισχυρή συνέπεια και είναι συμβατή με τις ιδιότητες ACID. [35]

### 3.3 Θετικά χαρακτηριστικά NoSQL Βάσεων Δεδομένων

Οι βάσεις δεδομένων NoSQL προσφέρουν σημαντικά πλεονεκτήματα στις επιχειρήσεις σε σχέση με τα παραδοσιακά RDBMS, τα πλεονεκτήματα αυτά συμπεριλαμβάνουν:

#### Επεκτασιμότητα

Οι βάσεις δεδομένων NoSQL χρησιμοποιούν μια οριζόντια κλιμάκωση που καθιστά εύκολη και γρήγορη την συνεχή προσθήκη ή τη μείωση των πόρων. Αυτό εξαλείφει το τεράστιο κόστος και την πολυπλοκότητα κατά την αύξηση της υποδύναμης του υλικού που είναι απαραίτητη όταν επιχειρείται η κατακόρυφη κλιμάκωση σε ένα RDBMS σύστημα.

#### Αποδοτικότητα

Με την κατανεμημένη αρχιτεκτονική των NoSQL βάσεων δεδομένων οι επιχειρήσεις μπορούν να αυξήσουν την απόδοσή τους απλά με την συνεχή προσθήκη νέων πόρων καθώς τα δεδομένα αυξάνονται. Αυτό επιτρέπει στους εκάστοτε οργανισμούς να συνεχίζουν να παρέχουν στους χρήστες αξιοπιστία και υψηλές ταχύτητες χωρίς επιβάρυνση στην απόδοση, -και πάλι, χωρίς το κόστος που σχετίζεται με το μη κατακόρυφη κλιμάκωση στα RDBMS συστήματα.

#### Μεγάλη διαθεσιμότητα

Οι βάσεις δεδομένων NoSQL σχεδιάστηκαν γενικά για να εξασφαλίζουν υψηλή διαθεσιμότητα και να αποφύγουν την πολυπλοκότητα που επέρχεται με μια τυπική αρχιτεκτονική RDBMS που βασίζεται σε πρωτεύοντες και δευτερεύοντες κόμβους. Ορισμένες κατανεμημένες NoSQL βάσεις δεδομένων χρησιμοποιούν μια λιγότερο “αυστηρή” αρχιτεκτονική η οποία κατανέμει αυτόματα τα δεδομένα εξίσου μεταξύ πολλών πόρων, έτσι ώστε η εφαρμογή να παραμένει διαθέσιμη για τις λειτουργίες ανάγνωσης και γραφής, ακόμη και όταν αποτύχει ένας κόμβος.

#### Παγκόσμια διαθεσιμότητα

Με την αυτόματη αναπαραγωγή δεδομένων σε πολλούς διακομιστές, κέντρα δεδομένων ή διακομιστές υπολογιστικού νέφους (cloud), οι NoSQL κατανεμημένες βάσεις δεδομένων

μπορούν να ελαχιστοποιήσουν την καθυστέρηση και να εξασφαλίσουν στους χρήστες σταθερή διαθεσιμότητα όπου κι αν βρίσκονται. Ένα πρόσθετο πλεονέκτημα είναι τα αδόμητα δεδομένα που αποθηκεύονται σε πολλαπλούς κόμβους απαλλάσσουν τον χρήστη από τον κόπο χειροκίνητης οργάνωσης των δεδομένων στην βάση, όπως συμβαίνει στα RDBMS. Με αυτόν τον τρόπο επιτρέπει στις ομάδες επιχειρήσεων να επικεντρωθούν σε άλλες προτεραιότητες.

### Ευέλικτο μοντέλο δεδομένων

Οι NoSQL προσφέρουν τη δυνατότητα εφαρμογής ευέλικτων μοντέλων δεδομένων. Τα εν λόγω συστήματα δεν συμμορφώνονται πιστά στους κανόνες ACID των RDBMS, έτσι ώστε να ανταποκρίνονται σε συνθήκες στις οποίες η απόδοση ενός ερωτήματος είναι πιο σημαντική από την απόλυτη συνέπεια των δεδομένων στο σχήμα της βάσης. Το αποτέλεσμα είναι μια ταχύτερη και με ελαστικότητα αλληλεπίδραση μεταξύ της εφαρμογής και της βάσης δεδομένων αλλά απλούστερη σε απαιτήσεις. [22]

Η παραπάνω λίστα με τα πλεονεκτήματα των NoSQL βάσεων φυσικά δεν είναι με κανένα τρόπο πλήρης. Ένα επιπλέον θετικό σημείο που πρέπει να προσθέσουμε είναι ότι οι περισσότερες από τους NoSQL βάσεις δεδομένων είναι ΕΛ/ΛΑΚ (Ελεύθερο Λογισμικό / Λογισμικό Ανοιχτού Κώδικα) και η εξέλιξή τους υποστηρίζεται από την παγκόσμια κοινότητα προγραμματιστών.

### **3.3.1 Πλεονεκτήματα συγκριτικά με τις SQL βάσεις δεδομένων**

Μερικές από τις πιο αξιοσημείωτες διαφορές των NoSQL βάσεων συγκριτικά με τις σχεσιακές αποτελούν:

- Η απουσία σχήματος (shemafree), η οποία επιτρέπει ευκολία σχεδιασμού.
- Η αποφυγή ενώσεων (joins) που στοχεύει σε αύξηση της απόδοσης.
- Η ικανότητα οριζόντιας επέκτασης (horizontal scaling) στοχεύοντας στην παροχή διαθεσιμότητας και μείωση του κόστους.
- Η υποστήριξη αυτόματης αναπαραγωγής η οποία βελτιώνει τη διαθεσιμότητα.

Πιο Αναλυτικά,

- ✓ τα συστήματα NoSQL αναπτύχθηκαν και εξελίχθηκαν παράλληλα με τις πιο μεγάλες εταιρίες πληροφορικής του κόσμου, όπως είναι η Google και η Amazon. Τέτοιες εταιρείες είχαν την ανάγκη να διαχειρίζονται έναν τεράστιο ποσότητες δεδομένων, κάτι στο οποίο δεν μπορούσαν να ανταποκριθούν τα κλασσικά μέχρι τότε συστήματα RDBMS.
- ✓ Τα συστήματα NoSQL σχεδιάστηκαν έτσι ώστε να μπορούν να διαχειριστούν μεγάλο όγκο δεδομένων χωρίς να διατηρούν υποχρεωτικά μία συγκεκριμένη δομή.
- ✓ Οι μέθοδοι υλοποίησης και εφαρμογής τους αξιοποιούν μια αρχιτεκτονική που επιτρέπει (και ταυτόχρονα διευκολύνει) την κατανομημένη δομή του συστήματος. Έτσι, με αυτόν τον τρόπο το σύστημα μπορεί θεωρητικά να «απογειωθεί» σε επιδόσεις, από την στιγμή που μπορούν να προστεθούν θεωρητικά άπειροι διακομιστές οι οποίοι θα επεξεργάζονται κατανομημένα τα δεδομένα του συστήματος. Ακόμη, η πιθανή αδυναμία λειτουργίας ενός διακομιστή του συστήματος μπορεί να αντιμετωπισθεί πιο εύκολα.
- ✓ Ο τύπος των NoSQL βάσεων δεδομένων αναπτύσσεται συνεχώς οριζόντια, καθώς αυξάνονται οι απαιτήσεις για αποθήκευση δεδομένων, ενώ είναι κρίνονται πιο σημαντικές οι επιδόσεις προσπέλασης των δεδομένων σε αντίθεση της συνοχής που παρουσιάζουν.[21]

Αφού αναφέρθηκαν μερικά στοιχεία που αφορούν την λειτουργία των συστημάτων NoSQL σε θεωρητικό επίπεδο, ακολουθούν τα πλεονεκτήματά τους σε πρακτικό πια επίπεδο.

- ✓ Παρουσιάζουν μεγάλη «ελαστικότητα» στην βελτίωση των επιδόσεών τους. Τις περισσότερες φορές η υλοποίησή τους γίνεται σε υποδομές υπολογιστικού νέφους ή σε εικονικό περιβάλλον, συμπεραίνοντας πως συμφέρουν αρκετά από οικονομική άποψη, σε αντίθεση με τα κλασσικά σχεσιακά συστήματα.
- ✓ Για την επίτευξη επεξεργασίας ακόμη περισσότερων δεδομένων ταυτόχρονα στα NoSQL συστήματα, απλά προσθέτουμε κόμβους. Στην αντίθετη περίπτωση τα σχεσιακά συστήματα, απαιτούν συνεχώς καλύτερες υποδομές όπως περισσότερη μνήμη ή καλύτερους επεξεργαστές για την βελτίωση τους.
- ✓ Επίσης, η συντήρηση ενός “κορυφαίου” συστήματος RDBMS που διαχειρίζεται τεράστιο όγκο δεδομένων απαιτεί εξίσου μεγάλο κόστος.

- ✓ Σε αντίθεση με τα σχεσιακά συστήματα, τα NoSQL συστήματα εξ'ορισμού είναι σχεδιασμένα ώστε να απαιτούν τη λιγότερη δυνατή διαχείριση από τον ανθρώπινο παράγοντα. [21]

Συνοψίζοντας, τα NoSQL συστήματα βάσεων δεδομένων κρίνονται ιδιαίτερα χρήσιμα κατά την διαχείριση τεράστιου όγκου δεδομένων και η φύση των δεδομένων αυτών δεν απαιτεί μια δομημένη δομή των δεδομένων.

### 3.4 Κατηγορίες NoSQL Βάσεων Δεδομένων

Οι NoSQL βάσεις δεδομένων δεν περιορίζονται σε μια προσέγγιση γραμμών και στηλών. Είναι σχεδιασμένες να χειρίζονται μια μεγάλη ποικιλία δεδομένων. Σε αυτά συμπεριλαμβάνονται και τα δεδομένα των οποίων η δομή αλλάζει με την πάροδο του χρόνου και δεδομένων των οποίων οι αλληλεπιδράσεις δεν είναι ακόμα γνωστές.

Οι NoSQL βάσεις δεδομένων κατηγοριοποιούνται ανάλογα με τον τύπο δεδομένων που πρόκειται να διαχειριστεί/αποθηκεύσει η βάση δεδομένων. Η ανάγκη να παρέχουν πληροφορίες που προέρχονται από επεξεργασία μεγάλων όγκων δεδομένων σε πραγματικό χρόνο, οδηγεί συνήθως τις NoSQL βάσεις να ακολουθούν μια οριζόντια δομή.

Οι πιο συνηθισμένοι τύποι που χρησιμοποιούνται είναι οι:

- Στηλοκεντρικές (Column-oriented)
- Εγγραφοκεντρικές (Document Stores)
- Ζεύγη κλειδιού – τιμής (Key Value Stores)
- Γράφοι (Graph)

- **Στηλοκεντρικές (Column-oriented) βάσεις δεδομένων**

Οι Στηλοκεντρικές βάσεις δεδομένων αποθηκεύουν τα δεδομένα κατά στήλη, σε αντίθεση με τις κλασσικές σχεσιακές στις οποίες τα δεδομένα αποθηκεύονται σε γραμμές. Η βασική του διαφορά έγκειται στην διάσταση που χρησιμοποιούν για την αποθήκευση των εγγράφων. Έτσι, αυτή η κατηγορία μη σχεσιακών βάσεων μπορεί να χρησιμοποιήσει την SQL ως γλώσσα ερωτημάτων για την ανάκτηση και επεξεργασία των δεδομένων.

- **Εγγραφοκεντρικές (Document Stores)**

Οι εγγραφοκεντρικές βάσεις δεδομένων, οι οποίες καλούνται και ως αποθήκες εγγράφων παρέχουν τρόπους για την εισαγωγή, την εξόρυξη και την επεξεργασία ημι-δομημένων δεδομένων. Οι περισσότερες βάσεις δεδομένων αυτού του τύπου χρησιμοποιούν έγγραφα της μορφής JSON. Τα έγγραφα αυτών των βάσεων είναι οι αντίστοιχες “γραμμές” στις σχεσιακές βάσεις δεδομένων.

- **Αποθήκες ζευγών κλειδιών-τιμών (Key Value Stores)**

Οι βάσεις δεδομένων ζεύγους “κλειδιού-τιμής” μοιάζουν με έναν πίνακα με δύο στήλες. Μια στήλη που αποτελείται από κλειδιά και μια δεύτερη που αποτελείται από τιμές. Κάθε ένα και μοναδικό κλειδί αντιστοιχεί σε μια μόνο τιμή. Η τιμή μπορεί να είναι οποιουδήποτε τύπου δεδομένων και φυσικά δεν απαιτείται συγκεκριμένο σχήμα για αυτή την κατηγορία βάσεων. Ευρετήριο (index) παρέχεται μόνο σε επίπεδο κλειδιών. Η βασική τους χρήση αφορά την υλοποίηση συστημάτων προσωρινής μνήμης.

- **Βάσεις δεδομένων γράφων (graph)**

Οι βάσεις δεδομένων γράφων σχεδιάστηκαν για δεδομένα όπου οι συσχετίσεις των οντοτήτων αποδίδονται καλύτερα με την αναπαράσταση γραφημάτων. Στα γραφήματα υπάρχουν πολλαπλές ακμές ανάμεσα σε δύο κόμβους, δίνοντας την δυνατότητα επεκτασιμότητας και αναπαράστασης πολλαπλών σχέσεων μεταξύ των οντοτήτων που συνδέονται.<sup>[16],[19]</sup>

ΤΥΠΟΣ	Επίδοση	Επεκτασιμότητα	Προσαρμοστικότητα	Πολυπλοκότητα
<b>Ζεύγη-Κλειδιά</b>	Υψηλή	Υψηλή	Υψηλή	Καθόλου
<b>Στήλες</b>	Υψηλή	Υψηλή	Μεσαία	Χαμηλή
<b>Έγγραφα</b>	Υψηλή	Μεταβλητή	Υψηλή	Χαμηλή
<b>Γράφοι</b>	Μεταβλητή	Μεταβλητή	Υψηλή	Υψηλή

Πίνακας 3.1: Τύποι NoSQL Βάσης δεδομένων

Υπάρχουν πάνω από 250 βάσεις δεδομένων που περιγράφονται από αναλυτές ως NoSQL. Παρακάτω παρουσιάζεται μια συνοπτική λίστα με τις πιο βασικές NoSQL βάσεις δεδομένων ανά κατηγορία :

**Key-value:** Basho Riak, Redis, Voldemort, Aerospike, Oracle NoSQL

**Columnar:** DataStax, Apache Cassandra, HBase, Apache Accumulo, Hypertable

**Document:** MongoDB, MarkLogic, CouchDB, FoundationDB, IBM Cloudant, Couchbase

**Triple/graph:** Neo4j, Ontotext's GraphDB (formerly OWLIM), MarkLogic, OrientDB,  
AllegroGraph, YarcData [38]

Περισσότερο διαδεδομένα κρίθηκαν τα συστήματα των κατηγοριών **στηλοκεντρικές** και **Εγγράφου**, για κάθε μία εκ των οποίων επιλέχθηκαν δύο εκπρόσωποι. Από την πρώτη κατηγορία οι **Cassandra** και **HBase**, ενώ από τη δεύτερη οι **CouchDB** και **MongoDB**

### 3.5 Συγκριτικός πίνακας Σχεσιακών και μη σχεσιακών βάσεων δεδομένων

Ακολουθεί συνοπτικός πίνακας με τις διαφορές μεταξύ των μη σχεσιακών και σχεσιακών Βάσεων δεδομένων

Οι βάσεις δεδομένων NoSQL :		Οι βάσεις δεδομένων SQL:
1	Κατηγοριοποιούνται ως Μη Σχεσιακό ή κατακευματισμένο σύστημα Βάσης Δεδομένων	Κατηγοριοποιούνται ως Σχεσιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων (RDBMS).
2	Έχουν δυναμικό σχήμα.	Έχουν σταθερό ή στατικό ή προκαθορισμένο σχήμα.
3	Απεικονίζουν δεδομένα ως συλλογή κλειδιών-τιμών, εγγράφων, γράφων ή στηλών.	Απεικονίζουν δεδομένα σε μορφή πινάκων
4	Είναι οριζόντια κλιμακούμενες	Είναι κατακόρυφα κλιμακούμενες
5	Η αποθήκευση εγγράφων χρησιμοποιείται για την συλλογή των δεδομένων. Έχει μη δομημένο σχήμα. Διαφέρει από βάση δεδομένων σε βάση δεδομένων.	Χρησιμοποιούν ισχυρή "Δομημένη Γλώσσα Ερωτημάτων" για τον καθορισμό και τον χειρισμό των δεδομένων.
6	Δεν είναι κατάλληλες για σύνθετα ερωτήματα, επειδή δεν είναι τόσο ισχυρά όσο τα ερωτήματα SQL	Είναι οι πλέον κατάλληλες για σύνθετα ερωτήματα.
7	Οι βάσεις δεδομένων NoSQL ταιριάζουν καλύτερα στην ιεραρχική αποθήκευση δεδομένων. διαχειρίζεται τεράστιο όγκο δεδομένων	Δεν είναι οι πλέον κατάλληλες για ιεραρχική αποθήκευση δεδομένων.
8	Το κόστος είναι μειωμένο	Απαιτεί μεγάλο κόστος

Πίνακας 3.2: SQL vs NoSQ [25]



## Κεφάλαιο 4

### Η βάση δεδομένων CouchDB

#### 4.1 Τι είναι η βάση δεδομένων CouchDB ;

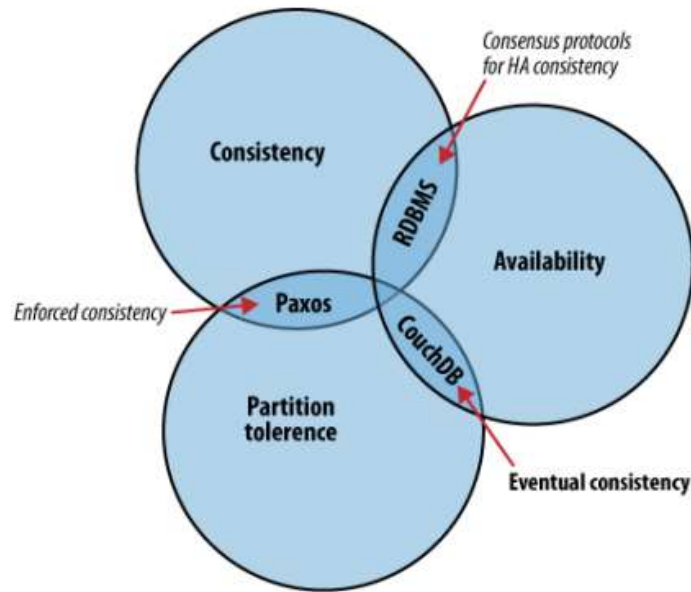
Η **CouchDB** ή Apache CouchDB είναι μία NoSQL βάση δεδομένων ανοιχτού κώδικα που εστιάζει στην ευχρηστία και διαθέτει κλιμακωτή αρχιτεκτονική. Κυκλοφόρησε για πρώτη φορά το 2005 και αργότερα το 2008 έγινε το Apache Software Foundation project. Βασίζεται σε ένα ελεύθερο σχήμα δεδομένων, που δε συμβαδίζει με την κλασική έννοια των πεδίων πίνακα μίας σχεσιακής βάσης δεδομένων.

Έχει συγγραφεί σε γλώσσα προγραμματισμού Erlang, μια συναρτησιακή γλώσσα προγραμματισμού, που χαρακτηρίζεται από την ιδιαίτερη έμφαση στην παραγωγή αξιόπιστων εφαρμογών. Η CouchDB αποτελεί μια εγγραφο-κεντρική (Document oriented) NoSQL βάση δεδομένων και θέτει ως προτεραιότητα την αποδοτική αποθήκευση μεγάλου όγκου δεδομένων και την επεξεργασία κατανεμημένων δεδομένων.

Η ονομασία της συνιστά ακρωνύμιο της φράσης **cluster of unreliable commodity hardware**, η οποία αντανακλά τους βασικούς στόχους της **CouchDB** να είναι εξαιρετικά επεκτάσιμη και να προσφέρει υψηλή διαθεσιμότητα και αξιοπιστία, ακόμη και όταν λειτουργεί σε υλικό το οποίο είναι επιρρεπές σε σφάλματα.

Πιο συγκεκριμένα η CouchDB είναι μια αξιόπιστη βάση δεδομένων, καθώς κάθε αλλαγή αποθηκεύεται ως αναθεώρηση εγγράφου. Η επικοινωνία με τη βάση δεδομένων επιτελείται μέσω ενός RESTfull API βασισμένο στο πρωτόκολλο HTTP , γεγονός που υποδεικνύει τη κατεύθυνση του συστήματος προς δικτυακές εφαρμογές. Επίσης, χρησιμοποιεί το JSON για την αποθήκευση δεδομένων, και τη JavaScript ως τη γλώσσα ερωτημάτων.[28][30][35]

## 4.2 Θεώρημα CAP



Εικόνα 4.1: Συμμόρφωση της CouchDB στο θεώρημα Cap [17]

Σύμφωνα με το θεώρημα CAP<sup>[43]</sup> υπάρχουν τρία επιθυμητά χαρακτηριστικά για κάθε σύστημα δεδομένων που έχει αναπτυχθεί σε κατακευματισμένο περιβάλλον.

- Συνέπεια (Consistency) : κάθε κόμβος στο σύστημα περιέχει τα ίδια δεδομένα.
- Διαθεσιμότητα (Availability): κάθε αίτηση σε ένα λειτουργικό κόμβο στο σύστημα επιστρέφει μια απάντηση.
- Ανοχή στο διαχωρισμό (Partition tolerance) : οι ιδιότητες του συστήματος (η συνέπεια και / ή η διαθεσιμότητα) έχουν ισχύ ακόμη και ακόμα και αν μεμονωμένα συστατικά του δεν είναι διαθέσιμα. [17]

Το θεώρημα αναφέρει ότι κάθε κατακευματισμένο σύστημα μπορεί να υποστηρίξει το πολύ, δύο από αυτές τις ιδιότητες. Στην παραπάνω εικόνα φαίνονται τα πιο δημοφιλή προϊόντα λογισμικού NoSQL βάσεων τοποθετημένα στις τομές PA (Partition-Availability), CP (Consistency-Partition) και AC (Availability-Consistency). Παρατηρείται το γεγονός ότι δεν υπάρχει μία και μοναδική λύση για την επίλυση των προβλημάτων μεγάλου όγκου Δεδομένων (Big Data). Αντίθετα υπάρχει πληθώρα διαφορετικών μοντέλων βάσεων δεδομένων καθένα από τα οποία είναι πιο ειδικό και πιο κατάλληλο για τη διαχείριση συγκεκριμένων ειδών προβλημάτων.

## 4.3 Λειτουργίες της CouchDB

### 4.3.1 Αναπαραγωγή

Ένα από τα δυνατά σημεία της CouchDB είναι η δυνατότητα να συγχρονίζει δυο αντίγραφα της ίδιας βάσης δεδομένων. Αυτό επιτρέπει στους χρήστες να διανέμουν δεδομένα μεταξύ των διακομιστές, αλλά και να έχουν πρόσβαση και να ενημερώνουν τα ίδια κοινόχρηστα δεδομένα ενώ αποσυνδέονται.

Η αναπαραγωγή αποτελείται από μια βάση δεδομένων προέλευσης και μια βάση δεδομένων προορισμού, η οποία μπορεί να είναι στον ίδιο ή σε διαφορετικό CouchDB διακομιστή. Ο στόχος της αναπαραγωγής είναι, στο τέλος της διαδικασίας, όλα τα ενεργά έγγραφα της πηγής να βρίσκονται επίσης στη βάση προορισμού και όλα τα έγγραφα που διαγράφηκαν από την πηγή να διαγράφονται επίσης από τον προορισμό. (ακόμα κι αν ήδη υπάρχουν μετά την αναπαραγωγή).

Η διαδικασία αναπαραγωγής είναι σταδιακή. Σε επίπεδο βάσης δεδομένων, η αναπαραγωγή εξετάζει μόνο τα έγγραφα που έχουν ενημερωθεί (Changes Documents) από την τελευταία αναπαραγωγή. Αυτό συμβαίνει συγκρίνοντας τις ενημερώσεις της βάση προέλευσης με τα έγγραφα του προορισμού. Επειδή η CouchDB έχει ενημερώσεις μόνο στη βάση προέλευσης, έτσι μειώνει τον κίνδυνο συγκρούσεων.

Τα έγγραφα που υπάρχουν ήδη στην βάση δεδομένων του προορισμού στην ίδια αναθεώρηση (Revision), δεν μεταφέρονται. Η διαγραφή των εγγράφων αναπαρίσταται από την δημιουργία μιας νέας αναθεώρησης. Ένα έγγραφο που διαγράφεται στην πηγή θα διαγραφεί επίσης στο προορισμό. Η διαδικασία αναπαραγωγής θα ολοκληρωθεί μόλις τελειώσει η ροή των ενημερώσεων (Changes). Εάν η αναπαραγωγή αποτύχει σε οποιοδήποτε βήμα, εξαιτίας προβλημάτων δικτύου ή διακοπής, για παράδειγμα, η επόμενη αναπαραγωγή επανεκκινείται στο τελευταίο σημείο ελέγχου.<sup>[28][31]</sup>

### 4.3.2 Master - Master και Master Slave replication

Η αναπαραγωγή μπορεί να διανέμει δεδομένα μόνο προς μια κατεύθυνση. Η CouchDB εκτός από Master slave (μονοκατευθυντική) αναπαραγωγή υποστηρίζει και Master – Master αναπαραγωγή όπου όλοι οι διακομιστές λειτουργούν αμφίδρομα.

Αυτό μπορεί να συμβεί αν ενεργοποιήσει και την αναπαραγωγή στον απομακρυσμένο εξυπηρετητή προορισμού. Δηλαδή να ορίσει δυο διαδικασίες αναπαραγωγής με αντίθετες κατευθύνσεις. Κατά την πρώτη διαδικασία αναπαραγωγής, μια ενημέρωση στην βάση A

αναπαράγεται στην βάση B. Κατά την αντίστροφη αναπαραγωγή (B->A), η B θα ανακαλύψει πως η ενημέρωση που δέχτηκε υπάρχει ήδη στην A και θα περιμένει περαιτέρω ενημερώσεις από την A. [32][31]

### 4.3.3 Υποστήριξη για κινητά

Η CouchDB ξεχωρίζει, δεδομένου ότι μπορεί να «τρέξει» σε κινητή συσκευή Android ή iOS. Επιπρόσθετα, το ότι είναι κινητή η βάση δεδομένων μπορεί επίσης να συγχρονιστεί με μια απομακρυσμένη κύρια βάση δεδομένων, επιτρέποντας την εύκολη κοινή χρήση των δεδομένων μεταξύ κινητών συσκευών και διακομιστών.

Μπορούν επίσης, να δημιουργηθούν και να συντηρηθούν επιμέρους αντίγραφα. Η αναπαραγωγή μπορεί να φιλτραριστεί με την λειτουργία Map Reduce, έτσι ώστε να αναπαραχθούν μόνο συγκεκριμένα έγγραφα ή εκείνα που πληρούν συγκεκριμένα κριτήρια. Αυτό μπορεί να επιτρέψει στους χρήστες να λαμβάνουν υποσύνολα μιας μεγάλης κοινής εφαρμογής βάσης δεδομένων χωρίς σύνδεση για δική τους χρήση, διατηρώντας ταυτόχρονα κανονική αλληλεπίδραση με την εφαρμογή και το υποσύνολο δεδομένων. [18][28]

### 4.3.4 MapReduce

Οι παραδοσιακές σχεσιακές βάσεις δεδομένων επιτρέπουν την εκτέλεση οποιοδήποτε ερωτήματος (query), εφόσον τα δεδομένα είναι σωστά δομημένα. Αντίθετα, στη CouchDB, τα ερωτήματα χρησιμοποιούν λειτουργίες MapReduce, η οποία μπορεί να είναι μια δύσκολη ιδέα για άτομα εξοικειωμένα με την SQL.

Η CouchDB χρησιμοποιεί τις συναρτήσεις Map και Reduce για να δημιουργήσει μια λειτουργία MapReduce. Το CouchDB χρησιμοποιεί το MapReduce, μια διαδικασία δύο βημάτων που εξετάζει όλα τα έγγραφα και δημιουργεί ένα αποτέλεσμα χάρτη που αποτελείται από μια ταξινομημένη λίστα ζευγών κλειδιού / τιμής. Η λειτουργία MapReduce πραγματοποιείται μία φορά μετά την δημιουργία ενός εγγράφου. Μετά από αυτό, δεν αλλάζει εκτός εάν το έγγραφο ενημερωθεί.

Αυτές οι λειτουργίες παρέχουν μεγάλη ευελιξία επειδή προσαρμόζονται στην ακαθόριστη δομή εγγράφων. Ο συνδυασμός των συναρτήσεων Map και Reduce ονομάζεται Προβολή (View) στην ορολογία CouchDB.[31]

### 4.3.5 Προβολές (Views)

Οι ιδιότητες των ACID ασχολούνται αποκλειστικά με την αποθήκευση και τις ενημερώσεις, αλλά χρειαζόμαστε επίσης τη δυνατότητα να παρουσιάζουμε τα δεδομένα μας με ενδιαφέροντα και χρήσιμα μέσα. Σε αντίθεση με τις βάσεις δεδομένων SQL, όπου τα δεδομένα πρέπει να αποσυντίθενται προσεκτικά σε πίνακες, τα δεδομένα στο CouchDB αποθηκεύονται σε ημιδομημένα έγγραφα. Τα έγγραφα CouchDB είναι ευέλικτα και το καθένα έχει τη δική του δομή, η οποία επιλύει τα πιο δύσκολα προβλήματα και τις παγίδες των αμφίδρομων αντιγράφων των σχεσιακών βάσεων καθώς και των δεδομένων που περιέχουν.

Αλλά, πέρα από τη λειτουργία τους ως διακομιστής αρχείων, ένα απλό μοντέλο εγγράφων για την αποθήκευση και την κοινή χρήση δεδομένων είναι πολύ απλό για την κατασκευή πραγματικών εφαρμογών - απλά δεν πραγματοποιεί πολλά από τα αναμενόμενα. Ο χρήστης επιθυμεί να δει τα δεδομένα με πολλούς διαφορετικούς τρόπους. Αυτό που χρειάζεται είναι ένας τρόπος να φιλτράρονται, να οργανώνονται και να ανακτώνται δεδομένα που δεν έχουν αποσυντεθεί σε πίνακες.

Οι προβολές (Views) είναι το κύριο εργαλείο εγγράφων CouchDB που χρησιμοποιείται για την δημιουργία ερωτημάτων (queries) και την ανάκτηση τους.

Οι προβολές είναι χρήσιμες για πολλούς σκοπούς.

Μια προβολή:

- ✓ Φιλτράρει τα δεδομένα της βάσης και βρίσκει αυτά που σχετίζονται με συγκεκριμένα φίλτρα.
- ✓ Εξάγει δεδομένα από την βάση και τα ανακτά με συγκεκριμένη σειρά.
- ✓ Δημιουργεί αποτελεσματικά ευρετηρίων, ώστε ο χρήστης να μπορεί να βρει έγγραφα με οποιαδήποτε τιμή που υπάρχει μέσα σε.
- ✓ Χρησιμοποιεί αυτά τα ευρετήρια για να αναπαραστήσει τις σχέσεις μεταξύ των δεδομένων.
- ✓ Τέλος, με τις προβολές ο χρήστης μπορεί να κάνει κάθε είδους υπολογισμούς στα δεδομένα των εγγράφων σας. Για παράδειγμα, εάν τα έγγραφα αντιπροσωπεύουν τις οικονομικές συναλλαγές της εταιρείας σας, μια προβολή μπορεί να απαντήσει στην ερώτηση για το ποια ήταν η δαπάνη την τελευταία εβδομάδα, μήνα ή έτος.

Τα μοντέλα αποθήκευσης, προβολής και ασφαλείας CouchDB εγγράφων έχουν σχεδιαστεί για να συνεργάζονται για αμφίδρομη αποτελεσματική και αξιόπιστη αναπαραγωγή. Τόσο τα έγγραφα όσο και τα δεδομένα σχεδιασμού (Design Document), μπορούν να αναπαραχθούν, επιτρέποντας την αναπαραγωγή ολοκληρωμένων εφαρμογών βάσεων δεδομένων (συμπεριλαμβανομένου του σχεδιασμού της λογικής και των δεδομένων των εφαρμογών) σε φορητούς υπολογιστές για χρήση εκτός σύνδεσης ή αναπαραγωγής σε διακομιστές σε απομακρυσμένα γραφεία όπου οι αργές ή αναξιόπιστες συνδέσεις καθιστούν δύσκολη την ανταλλαγή δεδομένων. [28][31][32]

#### **4.3.6 Έλεγχος Ταυτοχρονισμού Πολλαπλών Εκδόσεων (MVCC)**

Η CouchDB διατηρεί πολλαπλές εκδόσεις για κάθε αντικείμενο που απεικονίζεται στη βάση δεδομένων. Συγκεκριμένα, κάθε έγγραφο περιέχει ένα αναγνωριστικό αναθεώρησης εγγράφου `_rev` το οποίο υποδεικνύει την έκδοση της συγκεκριμένης εγγραφής, δηλαδή πόσες φορές έχει ενημερωθεί το συγκεκριμένο έγγραφο. Ο ρόλος του συγκεκριμένου πεδίου σχετίζεται με το γεγονός ότι η CouchDB δεν χρησιμοποιεί κλειδιά, αλλά βασίζεται στον Έλεγχο Ταυτοχρονισμού Πολλαπλών Εγγραφών. Ο MVCC επιβάλλει το να κρατούνται όλες οι αναθεωρήσεις για κάθε εγγραφή συμπεραίνοντας πως κάθε εισαγωγή πραγματοποιείται στο τέλος του αρχείου.[31]

#### **4.3.7 Ασφάλεια και επικύρωση**

Προκειμένου να προστατευτεί ο χρήστης που μπορεί να διαβάσει και να ενημερώσει τα έγγραφα, το CouchDB διαθέτει ένα απλό μοντέλο επικύρωσης πρόσβασης και επικαιροποίησης που μπορεί να επεκταθεί για την εφαρμογή προσαρμοσμένων μοντέλων ασφαλείας

## 4.4 Πλεονεκτήματα της χρήσης του CouchDB

- Ανοιχτή πηγή κώδικα
- Υπακούει στις ιδιότητες ACID (Ατομικότητα, Συνέπεια, Απομόνωση, Ανθεκτικότητα)
- Μπορεί να χειριστεί έναν μεγάλο όγκο ταυτόχρονων αναγνωστών και συγγραφέων χωρίς σύγκρουση, δηλ. MVCC (έλεγχος συνδρομικότητας πολλαπλών εκδόσεων)
- Υποστηρίζει πρωτόκολλα HTTP και REST API
- Διπλή αναπαραγωγή, συνεχής όταν χρειάζεται, με ανίχνευση συγκρούσεων
- Διατίθενται όλες οι προηγούμενες εκδόσεις των εγγράφων
- Οι προβολές μπορούν να δημιουργηθούν χρησιμοποιώντας ενσωματωμένες λειτουργίες μείωσης χάρτη (Map Reduce)
- Επαλήθευση ταυτότητας και χειρισμός συνημμένων.
- Αναβαθμίσεις σε πραγματικό χρόνο μέσω των ενημερώσεων ('\_changes'),<sup>[27]</sup>

Σε αντίθεση με μια σχεσιακή βάση δεδομένων ,

Μία βάση δεδομένων CouchDB :

- Δεν αποθηκεύει δεδομένα και σχέσεις σε πίνακες.
- Αντ 'αυτού, κάθε βάση δεδομένων είναι μια συλλογή από ανεξάρτητα έγγραφα.
- Κάθε έγγραφο διατηρεί τα δικά του δεδομένα και αυτόνομο σχήμα.
- Μια εφαρμογή μπορεί να έχει πρόσβαση σε πολλές βάσεις δεδομένων, όπως μία που είναι αποθηκευμένη στο κινητό τηλέφωνο ενός χρήστη και η άλλη σε έναν διακομιστή.
- Τα μεταδεδομένα εγγράφων περιέχουν πληροφορίες αναθεώρησης, καθιστώντας δυνατή τη συγχώνευση τυχόν διαφορών που ενδεχομένως εμφανίστηκαν κατά την αποσύνδεση των βάσεων δεδομένων.

#### 4.4.1 Πίνακας Σύγκρισης Σχεσιακής Βάσης Δεδομένων και CouchDB

Πίνακας 4.1: Πίνακας Σύγκρισης Σχεσιακής Βάσης Δεδομένων με Βάση δεδομένων CouchDB

	Σχεσιακή βάση δεδομένων	Βάση δεδομένων CouchDB
<ul style="list-style-type: none"> <li>• αποθηκεύει δεδομένα και σχέσεις σε πίνακες.</li> </ul>	✓	✗
<ul style="list-style-type: none"> <li>• κάθε βάση δεδομένων είναι μια συλλογή από ανεξάρτητα έγγραφα.</li> </ul>	✗	✓
<ul style="list-style-type: none"> <li>• Κάθε έγγραφο διατηρεί τα δικά του δεδομένα και αυτόνομο σχήμα.</li> </ul>	✗	✓
<ul style="list-style-type: none"> <li>• Μια εφαρμογή μπορεί να έχει πρόσβαση σε πολλές βάσεις δεδομένων</li> </ul>	✗	✓
<ul style="list-style-type: none"> <li>• Τα μεταδεδομένα εγγράφων περιέχουν πληροφορίες αναθεώρησης</li> </ul>	✗	✓
<ul style="list-style-type: none"> <li>• εκτέλεση ερωτημάτων (Queries) με JSON</li> </ul>	✗	✓
<ul style="list-style-type: none"> <li>• εκτέλεση ερωτημάτων(Queries) με Javascript</li> </ul>	✓	✓



## 4.5 Έγγραφο CouchDB

Κάθε αποθηκευμένο έγγραφο περιέχει μεταδεδομένα,(metadata) όπως μοναδικό κλειδί(\_uuid) και την αριθμό αναθεώρησης(\_rev).

Τα έγγραφα couch που είναι αποθηκευμένα σε μια CouchDB έχουν αναγνωριστικό εγγράφου (\_id). \_id είναι αναγνωριστικά συμβολοσειρών ευαίσθητων σε πεζά-κεφαλαία που προσδιορίζουν με μοναδικό τρόπο ένα έγγραφο.

Ο χρήστης είναι ελεύθερος να επιλέξει οποιαδήποτε συμβολοσειρά ως αναγνωριστικό, αλλά για καλύτερα αποτελέσματα συνίσταται ένα UUID (ή GUID), δηλαδή ένα Παγκόσμια Μοναδικό αναγνωριστικό.[32] Κάθε ενημέρωση σε μια βάση δεδομένων δημιουργεί έναν νέο αναγνωριστικό.

Δύο έγγραφα couch δεν μπορούν να έχουν το ίδιο αναγνωριστικό στην ίδια βάση δεδομένων διότι θεωρούνται ως το ίδιο έγγραφο.

Συνιστάται η χρήση μη ειδικών χαρακτήρων για αναγνωριστικά εγγράφων.

Τα UUIDs είναι τυχαίοι (32-bit) αριθμοί, οι οποίοι έχουν μια τόσο χαμηλή πιθανότητα σύγκρουσης που όλοι μπορούν να κάνουν χιλιάδες UUIDs ανά λεπτό για εκατομμύρια χρόνια χωρίς ποτέ να δημιουργήσουν ένα αντίγραφο. Αυτός είναι ένας πολύ καλός τρόπος για να εξασφαλιστεί ότι δύο ανεξάρτητοι άνθρωποι δεν μπορούν να δημιουργήσουν δύο διαφορετικά έγγραφα με το ίδιο αναγνωριστικό.

Απαιτείται προσοχή ως προς τον χρήστη, πρώτον, γιατί κάποιος άλλος θα μπορούσε να είναι στην θέση του αργότερα ή σε διαφορετικό υπολογιστή. Δεύτερον, η αναπαραγωγή του CouchDB επιτρέπει να μοιράζεται έγγραφα με άλλους χρήστες και χρησιμοποιώντας τα UUIDs διασφαλίζει ότι όλα λειτουργούν.

Κάθε ενημέρωση που πραγματοποιείται σε ένα έγγραφο αποθηκεύεται σε ένα νέο έγγραφο αναθεώρησης. Σε κάθε έγγραφο περιέχεται και ένα πεδίο \_rev (revision) το οποίο υποδεικνύει την έκδοση της συγκεκριμένης εγγραφής. Η αναθεώρηση εγγράφου αποτελεί ένα χαρακτηριστικό ασφαλείας κατά την αποθήκευση ενός εγγράφου.[31]

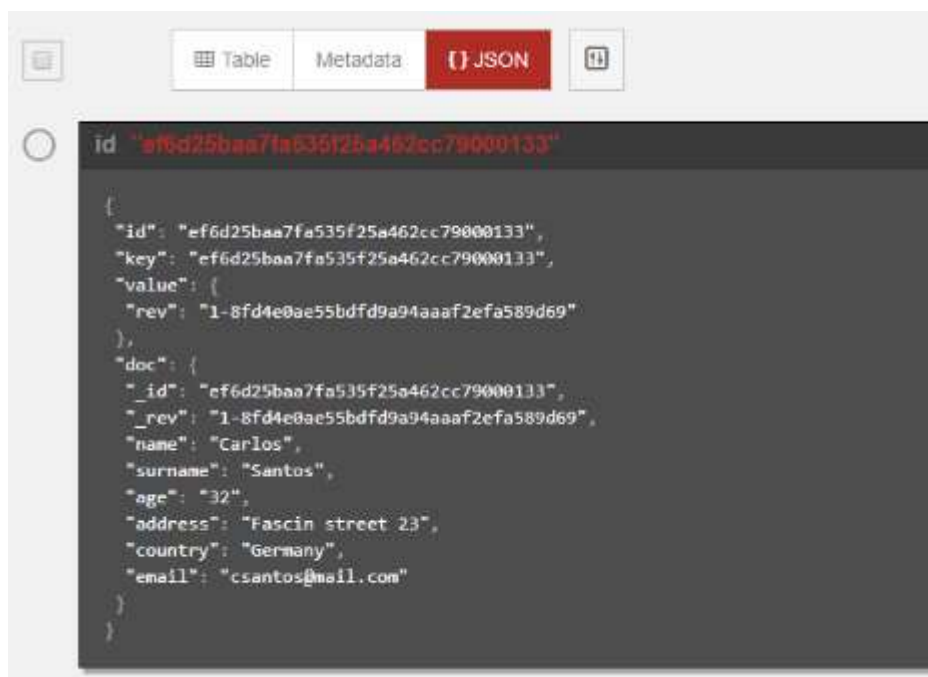
Αν λοιπόν το πεδίο \_rev έχει την μορφή: 2-xxxxxxxx(..)(32-bit συμβολοσειρά) σημαίνει πως είναι η 2<sup>η</sup> αναθεωρημένη έκδοση του συγκεκριμένου εγγράφου.

### 4.5.1 Fauxton

Η CouchDB προσφέρει μία διαδικτυακή διεπαφή (web interface) που ονομάζεται Fauxton, απλοποιώντας σημαντικά την επεξεργασία των δεδομένων από τον χρήστη, παρέχει πλήρη πρόσβαση σε όλες λειτουργίες του CouchDB και κάνει εφικτές και τις πιο περίπλοκες ιδέες. Με το Fauxton ο χρήστης μπορεί να δημιουργήσει και να καταργήσει βάσεις δεδομένων. Μπορεί να δημιουργήσει προβολές (views) των δεδομένων και να ανακτά έγγραφα. Μπορεί να συντάξει και να εκτελέσετε προβολές MapReduce και να ενεργοποιήσει την αναπαραγωγή δεδομένων μεταξύ βάσεων δεδομένων. [31]

Η σύνδεση του Fauxton στον περιηγητή, επιτυγχάνεται με τον παρακάτω σύνδεσμο: [http://127.0.0.1:5984/\\_utils/](http://127.0.0.1:5984/_utils/)

Πώς θα φαίνεται η δομή ενός τυπικού έγγραφου Couch στο Fauxton UI:

The image shows a screenshot of the Fauxton web interface. At the top, there are navigation tabs: 'Table', 'Metadata', and 'JSON' (which is highlighted in red). Below the tabs, there is a circular refresh icon and a document icon. The main content area displays a JSON document with the following structure:

```
id "ef6d25baa7fa535f25a462cc79000133"
{
  "id": "ef6d25baa7fa535f25a462cc79000133",
  "key": "ef6d25baa7fa535f25a462cc79000133",
  "value": {
    "rev": "1-8fd4e0ae55bdfd9a94aaaf2efa589d69"
  },
  "doc": {
    "_id": "ef6d25baa7fa535f25a462cc79000133",
    "_rev": "1-8fd4e0ae55bdfd9a94aaaf2efa589d69",
    "name": "Carlos",
    "surname": "Santos",
    "age": "32",
    "address": "Fascin street 23",
    "country": "Germany",
    "email": "csantos@mail.com"
  }
}
```

Εικόνα 4.2: Τυπικό έγγραφο Couch [29]

Πως θα φαίνεται η συλλογή των εγγράφων μιας Couch Βάσης Δεδομένων στο Fauxton UI:



doc.name	doc._id	doc.address	doc.age	doc.country	doc.name
Carlos	e96d25baa7fa335025a462cc79000123	Fascin street 23	32	Germany	carlos@i
Mike	e96d25baa7fa335025a462cc79000044	Halloween street 23	32	Germany	mike@i
Susan	e96d25baa7fa335025a462cc79001ae3	Music street 23	34	Germany	susan@i
Bruce	e96d25baa7fa335025a462cc790038af	Iron street 23	33	Germany	bruce@i

Εικόνα 4.3: Αποθήκευση εγγράφων Couch [16]

Η συνήθης τακτική είναι κάθε βάση δεδομένων να συλλέγει παρόμοιες εγγραφές, ωστόσο χάρη στο δυναμικό της σχήμα, οι εγγραφές της ίδιας βάσης δεδομένων μπορούν να διαφοροποιούνται σε ορισμένα πεδία ή και εξ' ολοκλήρου χωρίς κανένα σφάλμα.

## 4.6 Ιστορία της CouchDB

Στόχος της ήταν να αποτελέσει τη βάση δεδομένων του διαδικτύου, καθώς ο σχεδιασμός του είχε ως βασικό αντικείμενο να εξυπηρετήσει τις δικτυακές εφαρμογές.

Αρχικά είχε συγγραφεί σε C ++, αλλά στην πορεία μεταφέρθηκε στην πλατφόρμα Erlang OTP, λόγω των πλεονεκτημάτων της τελευταίας στη διαχείριση σφαλμάτων.

Το 2008 δόθηκε ως λογισμικό ανοικτού πηγαίου κώδικα με άδεια του Apache Software Foundation50, με την πρώτη έκδοση να διατίθεται τον Ιούλιο του 2010 . Έκτοτε έχει χρησιμοποιηθεί σε πληθώρα λογισμικών και υπηρεσιών, δραστηριοποιούμενων κατά κύριο λόγο στο διαδίκτυο

Η **CouchDB** δημιουργήθηκε τον Απρίλιο του 2005 από τον Damien Katz, πρώην προγραμματιστή της Lotus Notesστην IBM . Στόχος του ήταν να αποτελέσει τη βάση δεδομένων του διαδικτύου, καθώς ο σχεδιασμός του είχε ως βασικό αντικείμενο να εξυπηρετήσει τις δικτυακές εφαρμογές. Αρχικά είχε συγγραφεί σε C ++, αλλά στην πορεία μεταφέρθηκε στην πλατφόρμα Erlang OTP, λόγω των πλεονεκτημάτων της τελευταίας στη διαχείριση σφαλμάτων.

Κυκλοφόρησε ως έργο ανοικτού κώδικα υπό τη Γενική Άδεια Δημόσιας Χρήσης του GNU. Τον Φεβρουάριο του 2008, δημιουργήθηκε το Apache Incubator πρότζεκτ και

προσφέρθηκε με την Άδεια Apache . Αυτό οδήγησε στην έκδοση της πρώτης σταθερής έκδοσης τον Ιούλιο του 2010. Έκτοτε έχει χρησιμοποιηθεί σε πληθώρα λογισμικών και υπηρεσιών, δραστηριοποιούμενων κατά κύριο λόγο στο διαδίκτυο

Στις αρχές του 2012, ο Katz άφησε το πρότζεκτ να επικεντρωθεί στον Couchbase Server . Τον Ιούλιο του 2013, η κοινότητα CouchDB συγχώνευσε το κώδικα βάσης για το BigCouch , την cloud έκδοση του CouchDB του Cloudbant, στο πρόγραμμα Apache. Το πλαίσιο συσσώρευσης BigCouch περιλαμβάνεται στην τρέχουσα έκδοση του Apache CouchDB.

Το CouchDB είναι διαθέσιμο για Android, BSD, iOS, Linux, OS X, Solaris και Windows. Έχει υποστήριξη για πολλές γλώσσες προγραμματισμού, όπως: C, C #, ColdFusion, Erlang, Haskell, Java, JavaScript, Lisp, Lua, Objective-C, OCaml, Perl, PHP, PL / SQL, Python, Ruby και Smalltalk.[33]

## Κεφάλαιο 5

### Περιγραφή του Project

#### 5.1 Σκοπός

Σκοπός της παρούσας διπλωματικής είναι η αυτοματοποιημένη καταγραφή της συμπεριφοράς των χρηστών μιας εφαρμογής μέσα από την ενσωμάτωση ειδικών σεναρίων καταγραφής συμβάντων της διεπαφής σε επίπεδο πληκτρολογήσεων. Απώτερος στόχος είναι η εξαγωγή συμπερασμάτων για παραμέτρους ευχρηστίας αναλύοντας τα δεδομένα πληκτρολογήσεων των χρηστών.

#### 5.2 Εισαγωγή

Η αυτοματοποιημένη συλλογή γεγονότων και η ανάλυσή τους με κατάλληλες στατιστικές μπορούν να δώσουν χρήσιμες πληροφορίες σχετικά με τη συμπεριφορά των χρηστών για μια διεπαφή για την αξιολόγηση της ευχρηστίας της. Απώτερος στόχος της αξιολόγησης ευχρηστίας ενός λογισμικού είναι η βέλτιστη σχεδίαση των διεπαφών του για την μεγιστοποίηση της αποδοτικότητας και της προσωπικής ικανοποίησης των χρηστών.

Η συγκεκριμένη μελέτη θα υλοποιηθεί πάνω σε υπάρχων ERP σύστημα (Business Evolution) το οποίο χρησιμοποιείται ευρέως στην Ελληνική αγορά.

Το σύστημα θα αποτελείται από δύο υποσυστήματα:

**Το σύστημα καταγραφής συμπεριφοράς χρήστη.** Η συμπεριφορά του χρήστη αντικατοπτρίζεται από τη σειρά των ενεργειών του στη χρήση μιας φόρμας.

**Το σύστημα Στατιστικής ανάλυσης.** Αφορά την μελέτη της συλλογή των δεδομένων της συμπεριφοράς της χρήσης με σκοπό την εξαγωγή συμπερασμάτων για την ευχρηστία του συστήματος.

#### 5.3 ERP σύστημα Business Evolution

Για την υλοποίηση του συστήματος καταγραφής της συμπεριφοράς του χρήστη επιλέχθηκε ένα ήδη υπάρχων σύστημα ERP που μετρά ήδη δεκάδες εγκαταστάσεις, με εκατοντάδες χρήστες και χιλιάδες καθημερινές εγγραφές ώστε να διενεργήσουμε αξιολόγηση της

διεπαφής του με πραγματικά δεδομένα και πραγματικού χρόνου εργασιών ώστε τα συμπεράσματα που θα προκύψουν να είναι αντικειμενικά στην πραγματικότητα.

### 5.3.1 Σύστημα ERP

Ένα σύστημα ERP είναι ένα ολοκληρωμένο πληροφοριακό σύστημα που παρέχει τη δυνατότητα επικοινωνίας μεταξύ όλων των τμημάτων μιας επιχείρησης. Ένα σύστημα ERP, επιτελεί ουσιαστικά προσομοίωση της πραγματικότητας των καθημερινών πρακτικών. Οι λόγοι που οι επιχειρήσεις προμηθεύονται συστήματα ERP είναι οι εξής:

- Συγκρότηση όλων των μηχανογραφικών διαδικασιών κάτω από ένα ενιαίο μηχανογραφικό σύστημα.
- Αύξηση της παραγωγικότητας και της αποδοτικότητας.
- Βελτίωση της ποιότητας.
- Ολοκλήρωση επιχειρησιακών διαδικασιών.
- Μείωση κόστους σε όλη την εφοδιαστική αλυσίδα (αποθεμάτων, προμηθειών, logistics, πληροφοριακών πόρων).
- Ακεραιότητα και ακρίβεια πληροφοριών
- Ταχύτητα
- Βασική υποδομή για διευρυμένη επιχείρηση (extended enterprise)<sup>[36]</sup>

### 5.3.2 Business Evolution

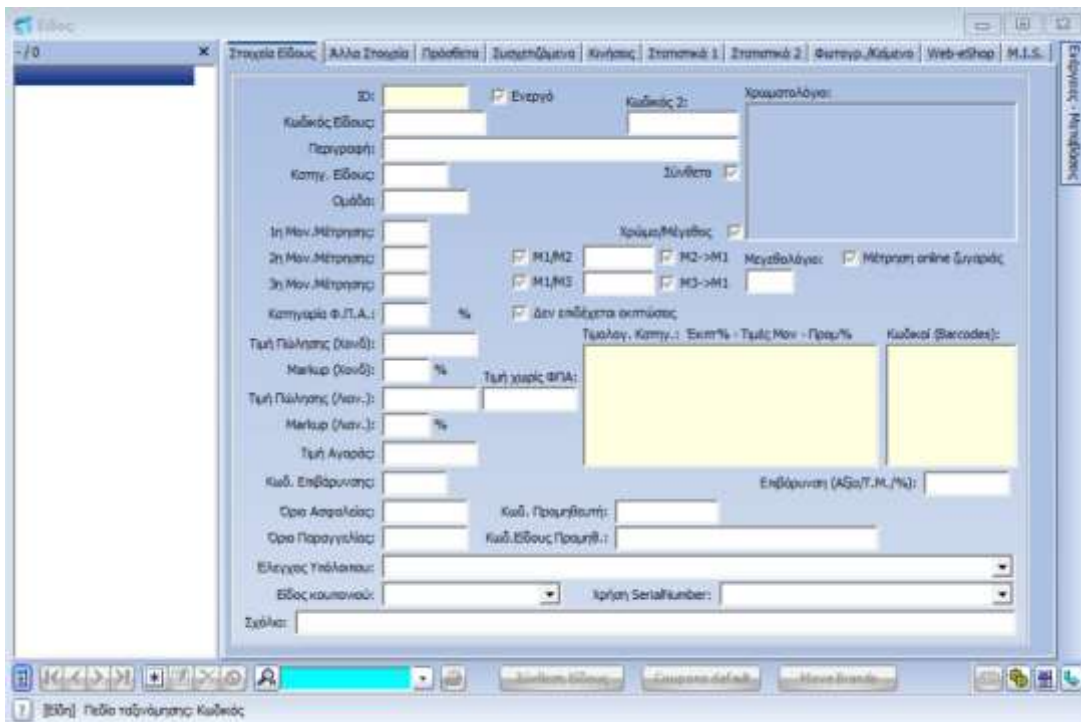
Το Business Evolution είναι ένα ERP μηχανογραφικό πρόγραμμα, εμπορικής και λογιστικής διαχείρισης που επεξεργάζεται και αξιοποιεί όλα τα δεδομένα της επιχείρησης. Υποστηρίζει όλες τις λειτουργίες και όλες τις επιχειρηματικές δραστηριότητες μιας επιχείρησης. Πρωτοπαρουσιάστηκε στην Ελληνική αγορά το 2001 και από τότε αριθμεί πάνω από 1000 εγκαταστάσεις τόσο στο cloud όσο και on premise σε όλη την Ελλάδα και περισσότερους από 2000 καθημερινούς χρήστες. Διαθέτει μεγάλη κοινότητα τεχνικών και μεταπωλητών που αναπτύσσουν custom modules για κάθε πελάτη.

Είναι Windows desktop εφαρμογή, ανεπτυγμένη σε Delphi. Διαθέτει web modules για εξειδικευμένες εργασίες. Έχει περίπου 70 φόρμες καταχώρησης δεδομένων και η βάση δεδομένων έχει περίπου 500 πίνακες. Ενσωματώνει ένα πλήρες ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) που περιλαμβάνει form designer, editor, debugger. Το Business Evolution ERP διαθέτει ενσωματωμένη scripted γλώσσα η οποία μπορεί και επεμβαίνει στις ιδιότητες

(properties) και μεθόδους (methods) των αντικειμένων μίας φόρμας στη διεπαφή του. Ο κώδικας της scripted γλώσσας του Business Evolution ERP, αποθηκεύεται στη Βάση Δεδομένων του BE (SQL Server), και είναι διαθέσιμος και εκτελέσιμος από όλες τις θέσεις εργασίας. Υπάρχουν υλοποιημένα scripts με events service routines για τα πεδία της κάθε φόρμας της διεπαφής του τα οποία αναπτύχθηκαν χρησιμοποιώντας τη scripting γλώσσα. Ως εκ τούτου, είναι ανοικτό για παραμετροποίηση ανάλογα με τον πελάτη και τις απαιτήσεις του. Τέλος, το BE ERP, διαθέτει Business Intelligence υποσύστημα (Cubes).

### 5.3.3 Περιγραφή διεπαφής φόρμας "Είδη".

Για τις ανάγκες της παρούσας διπλωματικής επιλέχθηκε μία αντιπροσωπευτική φόρμα της διεπαφής του BE για τη διενέργεια αξιολόγησης ευχρηστίας. Η προτεινόμενη μέθοδος θα εφαρμοστεί στην φόρμα «Είδη». Η φόρμα “Είδη” είναι η πιο συχνά επεξεργάσιμη σε επίπεδο καταγραφής πληροφοριών προϊόντων. Η φόρμα “Είδη” αφορά την καταγραφή των πληροφοριών των προϊόντων. Στην εικόνα 5.1 παρουσιάζεται η φόρμα στο σύνολό της ενώ παρακάτω περιγράφονται οι βασικές λειτουργίες που μπορεί ο χρήστης να πραγματοποιήσει κατά τη χρήση της.



Εικόνα 5.1: Η Φόρμα “Είδη” του Business Evolution

Η φόρμα Είδη του Business Evolution έχει την ίδια μορφή για όλους του χρήστες. Επιτρέπει στον χρήστη να αναζητεί να επεξεργάζεται να ενημερώνει και να αποθηκεύει πληροφορίες των προϊόντων που καταχωρεί.

Κατά το άνοιγμα της φόρμας το σύστημα καθοδηγεί τον χρήστη να προχωρήσει σε δυο ενέργειες. Την άμεση καταγραφή ενός προϊόντος, ή την αναζήτηση ενός προϊόντος προς ενημέρωση μιας ήδη υπάρχουσας καταγραφής.

### **Εισαγωγή Νέας Εγγραφής**

Για να δημιουργήσει μια νέα εγγραφή ο χρήστη μπορεί να “κλικάρει” το κουμπί με το σύμβολο “\*” του μενού της φόρμας, είτε να χρησιμοποιήσει το πλήκτρο “Insert” του πληκτρολογίου.

Σε αυτό το επίπεδο η φόρμα δίνει την άδεια για περαιτέρω επεξεργασία. Ο χρήστης ξεκινά την καταγραφή των πληροφοριών με το προεπιλεγμένο πεδίο Κωδικός Είδους. Το πεδίο αυτό μαζί με το πεδίο “Περιγραφή” και το πεδίο “Κατηγορία Είδους” αποτελούν τα πιο σημαντικά πεδία της φόρμας αφού δεν μπορεί να κάνει αποθήκευση της εγγραφής με λανθασμένα ή κενά αυτά τα πεδία. Αυτή η περίπτωση λάθους γίνεται αντιληπτή από τον χρήστη με την εμφάνιση μηνύματων λάθους.

### **Αποθήκευση Εγγραφής**

Η Αποθήκευση μιας εγγραφής σημαίνει την ολοκλήρωση της διαδικασίας καταχώρησης δεδομένων. Η αποθήκευση μπορεί να πραγματοποιηθεί με το πλήκτρο F2 του πληκτρολογίου ή χρησιμοποιώντας το κλικ του ποντικιού στο κουμπί του μενού εντολών της φόρμας.

### **Προβολή Εγγραφής**

Μετά την αποθήκευση η φόρμα επιστρέφει σε κατάσταση προβολής. Σε αυτή την φάση ο χρήστης μπορεί να επεξεργαστεί την ίδια φόρμα, να την εκτυπώσει ή να την διαγράψει. Διαφορετικά μπορεί να συνεχίσει με νέα εγγραφή ή αναζήτηση διαφορετικής εγγραφής.

### **Αναζήτηση Εγγραφής**

Όταν ο χρήστης επιθυμεί να αναζητήσει μια εγγραφή για προβολή ή ανανέωση αυτής μπορεί να το κάνει με δυο τρόπους. Με τη χρήση φίλτρου, χρησιμοποιώντας λέξεις κλειδιά ή χαρακτήρες του προϊόντος (κωδικός) και με το άνοιγμα της λίστας όλων των



προϊόντων επιλέγοντας ποια εγγραφή τον ενδιαφέρει. Το άνοιγμα της λίστας μπορεί να ενεργοποιηθεί και με την συντόμευση πληκτρολογίου F5.

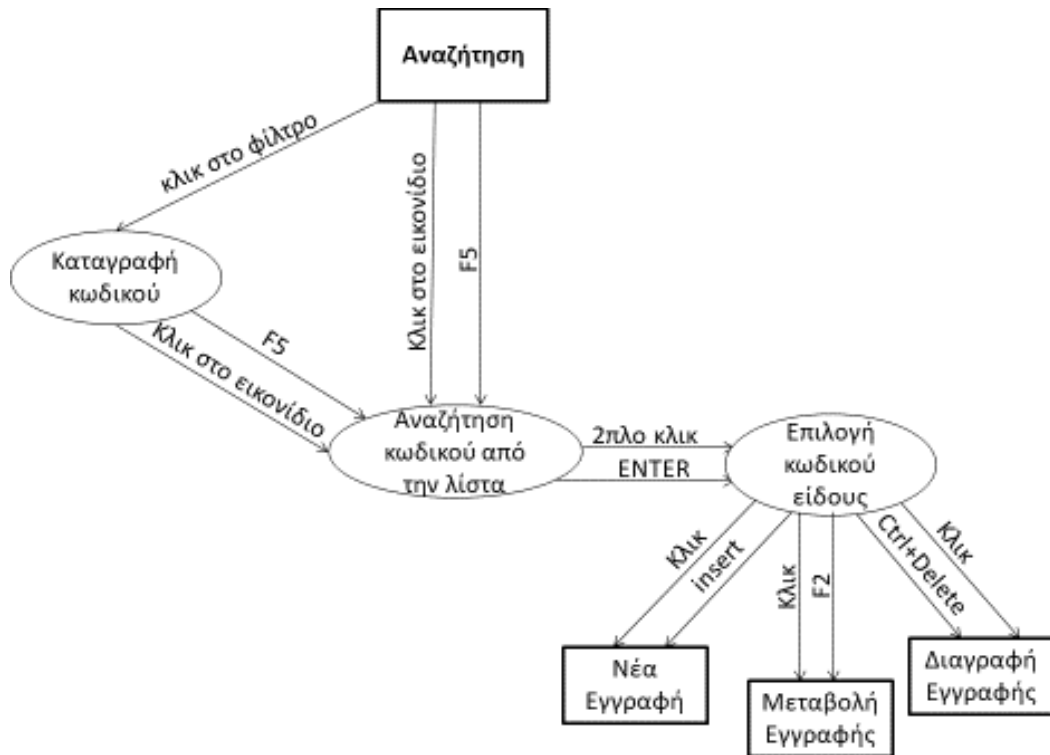
### Ενημέρωση Εγγραφής

Η επιλεγμένη εγγραφή προβάλλεται δίνοντας την δυνατότητα επεξεργασίας της προς ενημέρωση ή και διαγραφής της.

### Διαγραφή Εγγραφής

Αν ο χρήστης επιθυμεί την διαγραφή μιας εγγραφής, χρειάζεται να επιβεβαιώσει την ενέργεια του στο μήνυμα επιβεβαίωσης που λαμβάνει στην οθόνη του.

Ακολουθεί το διάγραμμα καταστάσεων των εργασιών από τους χρήστες:



Εικόνα 5.2 :Διάγραμμα καταστάσεων εργασιών του χρήστη

### Κατάσταση Επεξεργασίας

Η Φόρμα που επεξεργάζεται ο χρήστης αποτελείται από πεδία τύπου:

**Text:** Η πληροφορία που δέχεται προέρχεται από συμβολοσειρά από το πληκτρολόγιο

**List:** Η πληροφορία που δέχεται προκύπτει από την επιλογή στοιχείου από λίστα αρχείου

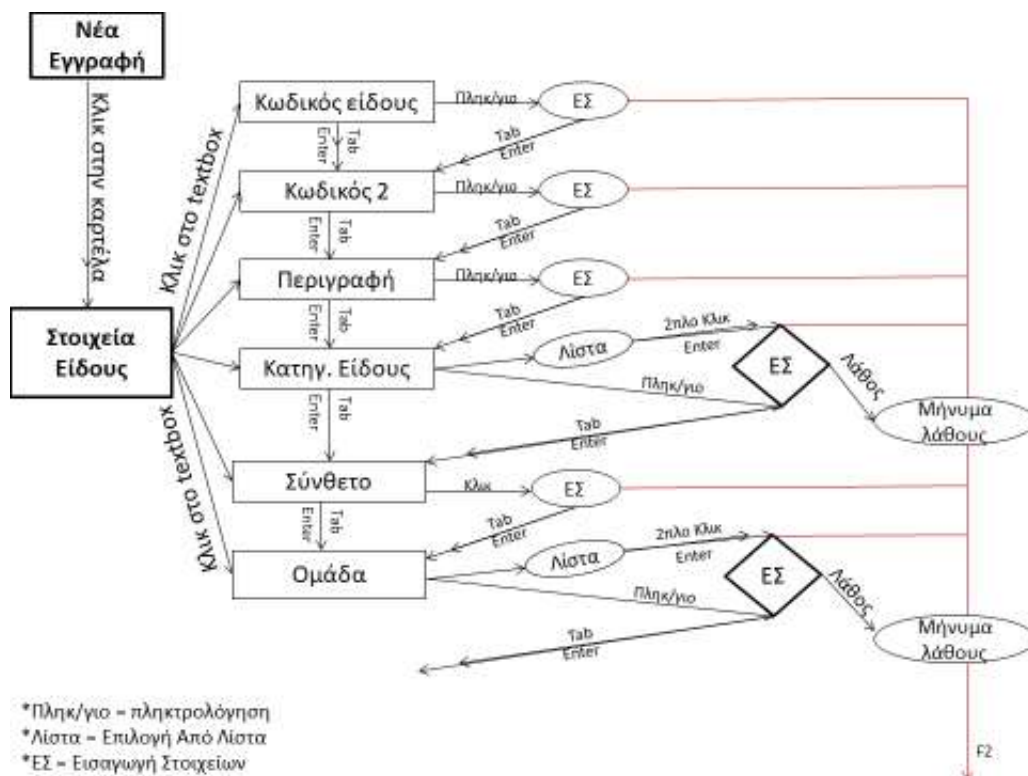
**Checkbox :** Η πληροφορία που δέχεται προκύπτει από το κλικ του ποντικιού ή του πλήκτρου “Space” για Checked/Unchecked.

**Drop down Menu:** Η πληροφορία που δέχεται προκύπτει από την επιλογή στοιχείου από μενού λίστας.

**Buttons:** Πληροφορία αποτελούν επίσης, και κλικς από τα κουμπιά των μενού της φόρμας όπως είναι το Save, Insert κτλ.

Όλες οι παραπάνω ενέργειες μαζί με τον τρόπο που κινείται ο χρήστης μέσα στην φόρμα σε κατάσταση επεξεργασίας αφορούν την συμπεριφορά του χρήστη.

Ακολουθεί διάγραμμα καταστάσεων ενεργειών του χρήστη σε κατάσταση επεξεργασίας μιας εγγραφής:



Εικόνα 5. : Διάγραμμα καταστάσεων ενεργειών σε κατάσταση επεξεργασίας

*\*Το Ολοκληρωμένο διάγραμμα καταστάσεων των ενεργειών ολόκληρης της φόρμας “Είδη” βρίσκονται στο παράρτημα.*

### 5.3.4 Προφίλ Χρηστών

Οι στόχοι ευχρηστίας είναι σε απόλυτη συνάρτηση με το προφίλ των χρηστών ενός διαδραστικού συστήματος καθώς και το πλαίσιο χρήσης του. Τα συστήματα σχεδιάζονται

ανάλογα τον χρήστη που θα τα διαχειριστεί. Άλλα αφορούν έμπειρους χρήστες, αρχάριους, ευκαιριακούς ή των συνδυασμό αυτών. Ανάλογα με τις γνώσεις και τις απαιτήσεις των χρηστών τα συστήματα δίνουν την κατάλληλη έμφαση ευχρηστίας στα διάφορα σημεία του συστήματος.

Χαρακτηριστικό διαχωρισμού των χρηστών είναι η εξοικείωση τους με τη τεχνολογία.

Με βάση αυτή την παράμετρο, οι χρήστες διακρίνονται από έμπειρους μέχρι άπειρους.

Στη περίπτωση του συγκεκριμένου συστήματος οι χρήστες δεν είναι αντιπροσωπευτικοί αλλά πραγματικοί χρήστες των οποίων η εργασία τους, τους επιβάλλει να κάνουν χρήση πληροφοριακών συστημάτων και να είναι εξοικειωμένοι με παρόμοια περιβάλλοντα. Δεν ανήκουν στην κατηγορία των άπειρων χρηστών αλλά συχνά χρειάζονται κατάλληλη καθοδήγηση για την πιο γρήγορη και εύχρηστη διαχείριση του συστήματος.

Μια εργασία μπορεί να έχει περισσότερους από έναν τρόπους και χρόνους ολοκλήρωσης. Πολλές φορές ο χρήστης επιλέγει τον τρόπο με τον μεγαλύτερο χρόνο ολοκλήρωσης μιας εργασίας και αυτό συμβαίνει γιατί δεν έχει την κατάλληλη τεχνολογική κατάρτιση και εξοικείωση ώστε να σκεφτεί ή να απομνημονεύσει τον "γρήγορο τρόπο".

Είναι σημαντικό να αναφέρουμε πως κατά την διεξαγωγή της δοκιμής ευχρηστίας, η διαδικασία της δοκιμής είναι ίδια για κάθε συμμετέχοντα.

#### **5.4 Αρχιτεκτονική συστήματος**

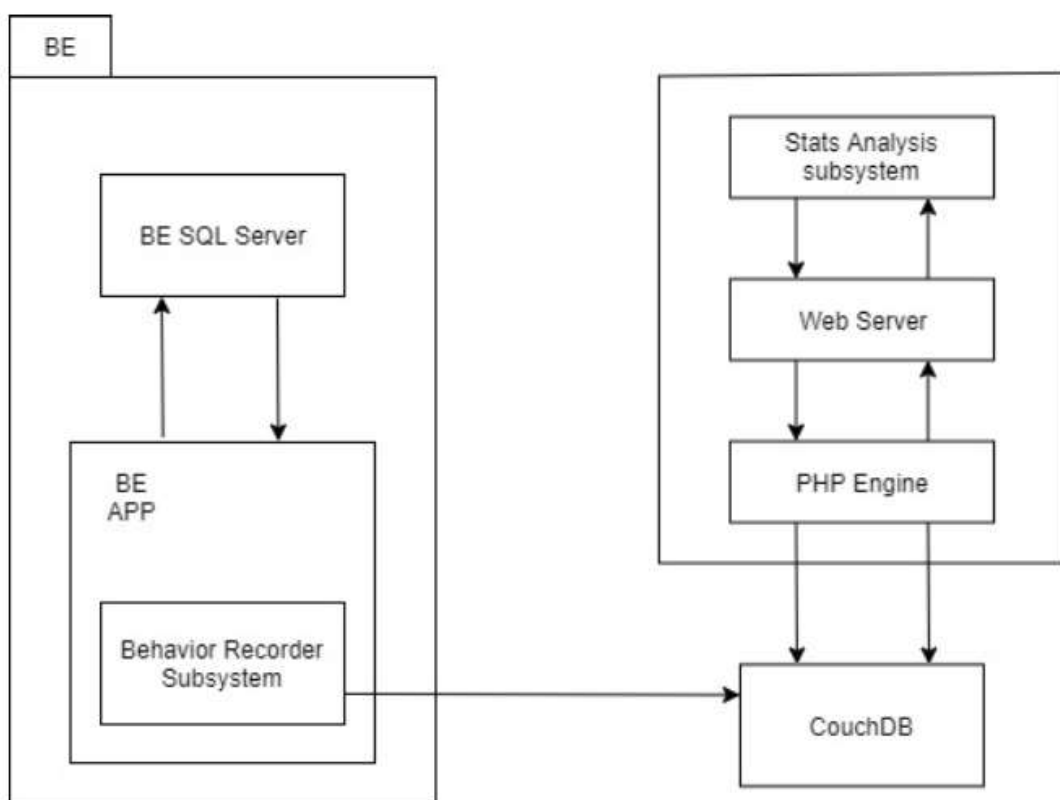
Η φόρμα έχει δημιουργηθεί κατά βάση με σενάρια του εργαλείου Delphi. Η Delphi παρέχει την δυνατότητα για δημιουργία οποιουδήποτε σεναρίου απαιτεί ο προγραμματιστής καθώς του επιτρέπει να επέμβει εμβόλιμα και να εισάγει οποιοδήποτε σενάριο που πυροδοτείται από συμβάντα της διεπαφής της κάθε φόρμας. Με την χρήση τέτοιων σεναρίων υλοποιήθηκε το script το οποίο αναγνωρίζει και καταγράφει σε βάση δεδομένων την συμπεριφορά του χρήστη πάνω στην επιλεγμένη φόρμα. Το σύστημα που αναπτύχθηκε αποτελείται από δύο υποσυστήματα : Το υποσύστημα καταγραφής συμπεριφοράς του χρήστη και το υποσύστημα στατιστικής ανάλυσης.

Το υποσύστημα καταγραφής της συμπεριφοράς του χρήστη αναπτύχθηκε χρησιμοποιώντας σενάρια DelphiScript τα οποία ενσωματώθηκαν στα σενάρια της επιλεγμένης φόρμας του BE. Το συγκεκριμένο υποσύστημα αφού αναγνωρίσει τις ενέργειες του χρήστη, πάνω στη φόρμα έως ότου ολοκληρώσει την ενέργεια του, αποστέλλει την

εγγραφή στη διαδικτυακή NOSQL βάση δεδομένων CouchDB μέσω του πρωτοκόλλου cURL.

Το υποσύστημα της στατιστικής ανάλυσης αναπτύχθηκε ως διαδικτυακή εφαρμογή χρησιμοποιώντας HTML, PHP, Javascript και βρίσκεται ενσωματωμένο στον web server του BE. Τα PHP scripts που αναπτύχθηκαν δημιουργούν και αποστέλλουν ερωτήματα στην CouchDB. Η απαντήσεις της CouchDB προσαρμόζονται κατάλληλα σε μία σελίδα δυναμικού περιεχομένου με γραφήματα, με την χρήση της βιβλιοθήκης Jcanvas, και παρέχονται στην ομάδα σχεδιασμού των διεπαφών του BE.

Στο παρακάτω διάγραμμα παρουσιάζεται η αρχιτεκτονική του συστήματος.



Εικόνα 7. 24: Διάγραμμα αρχιτεκτονικής συστήματος

#### 5.4.1 Υποσύστημα Καταγραφή συμπεριφοράς χρήστη

Οι φόρμες καταχώρησης δεδομένων ενός **ERP** συστήματος Enterprise Resource Planning (Διαχείριση Επιχειρησιακών Πόρων) παράγουν ένα μεγάλο αριθμό συμβάντων, η καταγραφή των οποίων μπορεί να δώσει πλούσιο υλικό για τη μελέτη της συμπεριφοράς των χρηστών. Για την υλοποίηση της καταγραφής του υποσυστήματος θα πρέπει να έχουμε τη δυνατότητα επέμβασης στα συμβάντα της κάθε φόρμας. Η συγκεκριμένη εφαρμογή είναι

ανοικτή για παραμετροποίηση γεγονός που επιτρέπει την επέμβαση στα συμβάντα των φορμών και των controls τους.

**User\_Actions :**

- Enter
- Tab
- Mouse click
- Keyboard shortcuts
- Space
- Insert
- Save(f2)

Οι ενέργειες του χρήστη πυροδοτούν συμβάντα πάνω στην φόρμα ή στα Form controls

Form controls	Form Events :	Script Events
<ul style="list-style-type: none"><li>• (Textbox) Tedit Tdbedit</li><li>• Checkbox</li><li>• Button</li><li>• Insert</li><li>• Save</li></ul>	<ul style="list-style-type: none"><li>• Textbox Get_Focus</li><li>• Textbox Lost_Focus</li><li>• Checkbox checked</li><li>• Checkbox unchecked</li><li>• Button pressed</li><li>• Insert</li><li>• Save</li></ul>	<ul style="list-style-type: none"><li>• OnMouseDown</li><li>• OnKeyUp</li><li>• OnEnter</li><li>• OnExit</li><li>• OnNewRecord</li><li>• OnSavingRecord</li></ul>

Πρόβλημα 1: Η καταγραφή κάθε συμβάντος μιας φόρμας στη βάση δεδομένων ξεχωριστά θα δημιουργούσε μεγάλο όγκο δεδομένων στο δίκτυο και ενδεχόμενες καθυστερήσεις στην λειτουργία της εφαρμογής.

Λύση 1: Για τη βέλτιστη καταγραφή της συμπεριφοράς του χρήστη, δημιουργήθηκε το διάγραμμα καταστάσεων (state chart diagram) της φόρμας Είδη, το οποίο ουσιαστικά περιγράφει τη λειτουργία της. Στο διάγραμμα καταστάσεων υπάρχει μία τελική κατάσταση (final state) η οποία ενεργοποιεί την αποστολή δεδομένων στη βάση.

Η ακολουθία των ενεργειών του χρήστη καταγράφεται σε μία μεταβλητή **Tracker** (τύπου array), ορατή σε όλα τα συμβάντα. Το περιεχόμενο αυτής της μεταβλητής είναι ουσιαστικά ο περίπατος του χρήστη μέσα στο διάγραμμα καταστάσεων της φόρμας.

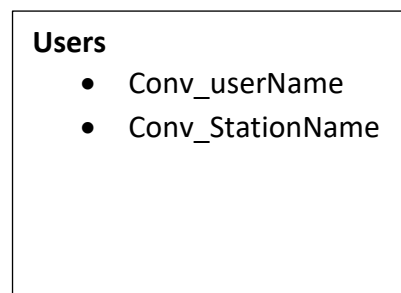
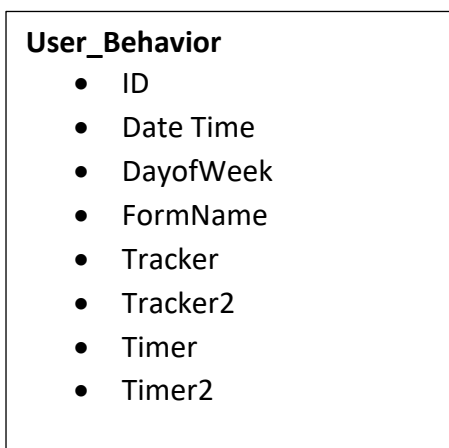
Όταν η ενεργή κατάσταση της φόρμας φτάσει στην τελική κατάσταση του διαγράμματος καταστάσεων θα αποστέλλεται το **Tracker** (μαζί με τις υπόλοιπες πληροφορίες) στη βάση.

**Πρόβλημα 2:** Ο μεγάλος αριθμός καθημερινών του ERP θα δημιουργήσει υλικό μεγάλου όγκου το οποίο δύσκολα θα μπορούσε να διαχειριστεί από μία σχεσιακή βάση δεδομένων

**Λύση 2:** Σε αντίθεση με μια σχεσιακή Βάση δεδομένων, μια NoSQL βάση δεδομένων (CouchDB):

- Μπορεί να χειρίζεται μεγάλο όγκο
- Δεν υπάρχει προκαθορισμένο σχήμα για τα μοντέλα δεδομένων.
- Χειρίζεται δεδομένα τα οποία συχνά έχουν μικρή έως και καθόλου δομή.
- Είναι φθηνότερο να διαχειριστεί.
- Λειτουργεί με Οριζόντια κλιμάκωση.

Για την καταγραφή της συμπεριφοράς του χρήστη οι ακόλουθες πληροφορίες θα πρέπει να αποστέλλονται στη βάση δεδομένων:



### 5.4.2 Υποσύστημα στατιστικής ανάλυσης

Το σύστημα, μέσω στατιστικής ανάλυσης των καταγεγραμμένων δεδομένων, έχει τη δυνατότητα να μελετάει τη συμπεριφορά των χρηστών όσο αναφορά την φόρμα “Είδη” σε σχέση με τον τύπο χρήστη, και τις συνήθειες του.

Η ανάλυση των καταγεγραμμένων δεδομένων είναι επίπονη διαδικασία. Για το λόγο αυτό δημιουργήθηκε εργαλείο στατιστικής ανάλυσης των δεδομένων με το οποίο θα είναι δυνατή η εξαγωγή ποικίλων μετρήσεων. Οι μετρήσεις έχουν μεγάλη συνεισφορά στην σωστή αξιολόγηση της ευχρηστίας του συστήματος.

Το υποσύστημα στατιστικής ανάλυσης έχει τη δυνατότητα σύγκρισης της συμπεριφοράς των χρηστών

- μεταξύ διαφορετικής διαχείρισης της φόρμας από έναν χρήστη,
- μεταξύ διαφορετικών τύπων χρηστών,
- μεταξύ διαφορετικών χρόνων ολοκλήρωσης μιας εργασίας.

#### **Ερωτήματα που μετρήθηκαν και αναλύθηκαν αφορούν**

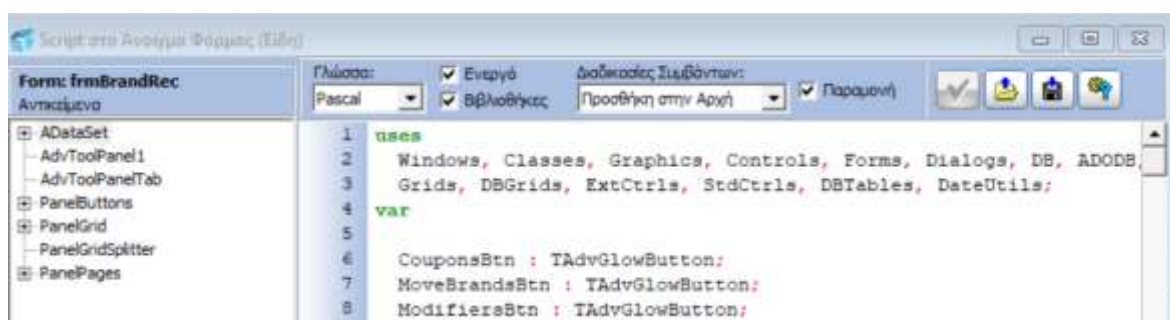
- Τις ενέργειες του κάθε χρήστη
- Τον τρόπο με τον οποίο περιηγείται μέσα στην φόρμα ο χρήστης
- Τις πιο συνηθισμένες μέρες και ώρες που χειρίζεται ο χρήστης την φόρμα
- Το Ποσοστό επιτυχούς ολοκλήρωσης εργασίας
- Τον Χρόνος επιτυχούς ολοκλήρωσης εργασίας
- Το Ποσοστό ολοκλήρωσης εργασίας χρησιμοποιώντας ένα συγκεκριμένο τρόπο εργασίας

## ΚΕΦΑΛΑΙΟ 6

### Υποσύστημα καταγραφής συμπεριφοράς χρήστη

Για την ανάπτυξη του συστήματος καταγραφής χρήστη χρησιμοποιήθηκε το εργαλείο Delphi και οι πληροφορίες που καταγράφονται αποθηκεύονται στην CouchDB μέσω του πρωτοκόλλου cURL. Οι τεχνολογίες που χρησιμοποιήθηκαν αναπτύσσονται παρακάτω:

Ο προγραμματισμός της εφαρμογής έγινε με το εργαλείο **Delphi**.



Εικόνα 6.1: Business Evolution

#### 6.1 Το εργαλείο Delphi

Είναι ένα εργαλείο προγραμματισμού, το οποίο μοιάζει πολύ στη Visual Basic. Η ανάπτυξη εφαρμογών γίνεται μέσω του ολοκληρωμένου περιβάλλοντος (IDE) που διαθέτει. Βασίζεται στη γλώσσα προγραμματισμού Object Pascal, η οποία είναι αντικειμενοστραφής γλώσσα. Το Delphi είναι ευέλικτο και έχει πολλά πλεονεκτήματα σε όλα τα στάδια ανάπτυξης εφαρμογών, καθώς και στην εγκατάσταση των εφαρμογών στους τελικούς χρήστες, αφού η εφαρμογή μπορεί να αποτελείται από ένα αυτόνομο εκτελέσιμο αρχείο (EXE), χωρίς να βασίζεται στην ύπαρξη άλλων αρχείων. Ο μεταγλωττιστής παράγει πολύ γρήγορες εφαρμογές. Με το Delphi είναι πολύ εύκολη η επαναχρησιμοποίηση των στοιχείων και του κώδικα καθώς και η ενσωμάτωση άλλων βιβλιοθηκών, που κάνει πολύ γρήγορη και εύκολη τη δουλειά κάθε προγραμματιστή.

Υπάρχουν πολλές έτοιμες σπουδαίες βιβλιοθήκες που μπορεί κάποιος να αγοράσει ή να κατεβάσει από το Internet, που θα του λύσουν τα χέρια στη δουλειά του. Είναι πολύ ισχυρό στην ανάπτυξη εφαρμογών με βάσεις δεδομένων και έχει wizard για τη γρήγορη ανάπτυξη εφαρμογών αυτού του είδους. Το Delphi σε κάθε έκδοση εμφανίζεται όλο και πιο ισχυρό.



Έτσι, από την έκδοση 3 και μετά μπορεί κάποιος να δημιουργήσει client και server εφαρμογές για το Internet<sup>[36]</sup>

### **6.1.1 Προγραμματισμός σε Delphi**

Κατά την ανάπτυξη μιας εφαρμογής σε Delphi ο χρήστης δημιουργεί φόρμες, οι οποίες αποτελούνται από μηχανισμούς αντικείμενα και σε κάθε μηχανισμό αντιστοιχίζει τον κώδικα που χρειάζεται για να λειτουργεί η εφαρμογή όπως πρέπει. Βέβαια υπάρχουν έτοιμες φόρμες που παρέχονται μαζί με το εργαλείο, οι οποίες μπορούν να καλύψουν πολλές από τις ανάγκες του προγραμματιστή.

Επίσης υπάρχουν βιβλιοθήκες με πάρα πολλές φόρμες έτοιμες για χρήση που μπορεί να προμηθευτεί από τρίτους. Συνήθως, ο χρήστης πρέπει να δημιουργήσει από μόνος του φόρμες που ανταποκρίνονται σε συγκεκριμένες απαιτήσεις της εφαρμογής. Σε κάθε φόρμα που δημιουργεί τοποθετεί ένα πλήθος μηχανισμών. Μπορεί να επιλέξει μέσα από τα διαθέσιμα συστατικά ότι χρειάζεται. <sup>[37]</sup>

*\*Στο παράρτημα ο κώδικας*

## **6.2 Εγκαταστάση LAMP και CouchDB σε Ubuntu**

### **Εισαγωγή και Προϋποθέσεις**

Μια στοίβα "LAMP" είναι μια ομάδα λογισμικού ανοιχτού κώδικα που συνήθως εγκαθίσταται μαζί για να επιτρέψει σε έναν διακομιστή να φιλοξενήσει δυναμικές ιστοσελίδες και εφαρμογές ιστού. Αυτός ο όρος είναι στην πραγματικότητα ένα ακρωνύμιο που αντιπροσωπεύει το λειτουργικό σύστημα Linux, με τον διακομιστή ιστού Apache.

### **Το λειτουργικό σύστημα Ubuntu**

Ο Ubuntu είναι ένα πλήρες λειτουργικό σύστημα βασισμένο όμως στο Linux. Είναι αφάνταστα απλό και εύκολο στην χρήση του. Το περιβάλλον εργασίας του είναι πλήρως εξελληνισμένο και εξαιρετικά φιλικό προς τον χρήστη.

### **Ο εξυπηρετητής Apache**

Ο Apache HTTP γνωστός και απλά σαν Apache είναι ένας εξυπηρετητής του παγκόσμιου ιστού (web). Όποτε ένας χρήστης επισκέπτεται ένα ιστότοπο το πρόγραμμα πλοήγησης (browser) επικοινωνεί με έναν διακομιστή (server) μέσω του πρωτοκόλλου HTTP, ο οποίος

παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης. Ο Apache είναι ένας από τους δημοφιλέστερους εξυπηρετητές ιστού, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες όπως τα Windows, το Linux, το Unix και το Mac OS X.

*\* Εγκατάσταση LAMP CouchDB στο παράρτημα*

## 6.3 Σύνδεση με την Βάση δεδομένων Couchdb

### 6.3.1 cURL

Η CouchDB Χρησιμοποιεί Couch Replication Protocol το οποίο είναι πρωτόκολλο που συγχρονίζει JSON δεδομένα μεταξύ δυο peers μέσω HTTP

Τα στοιχεία της διαδρομής της διεύθυνσης API καθορίζουν το τμήμα του διακομιστή CouchDB στον οποίο γίνεται πρόσβαση. Το αποτέλεσμα είναι ότι η δομή του αιτήματος URL προσδιορίζει και περιγράφει αποτελεσματικά την περιοχή της βάσης δεδομένων στην οποία έχει γίνει πρόσβαση. Όπως συμβαίνει με όλες τις διευθύνσεις URL, τα μεμονωμένα στοιχεία διαχωρίζονται με μια κάθετο.

Κατά γενικό κανόνα, τα στοιχεία URL και τα πεδία JSON που αρχίζουν με τον `_` χαρακτήρα (υποσημείωση) αντιπροσωπεύουν ένα ειδικό στοιχείο ή οντότητα στο διακομιστή ή στο αντικείμενο που επιστρέφεται. Για παράδειγμα, το κομμάτι διευθύνσεων URL `/_all_dbs` περιλαμβάνει μια λίστα με όλες τις βάσεις δεδομένων σε έναν διακομιστή CouchDB.

- Είναι ουσιαστικά μια βάση δεδομένων noSQL που είναι διαθέσιμη στον χρήστη μέσω της δημιουργίας HTTP requests ώστε να μπορεί να συνδεθεί με αυτήν μέσω CURL ή σχεδόν οτιδήποτε που μας επιτρέπει να υποβάλουμε requests.
- Η CouchDB προσφέρει ένα RESTful HTTP API για ανάγνωση, προσθήκη, επεξεργασία και διαγραφή εγγράφων βάσης δεδομένων

(GET, PUT, POST, DELETE.) [31]

Ο χρήστης μπορεί να χρησιμοποιήσει το βοηθητικό πρόγραμμα Curl για να εργαστεί με την CouchDB από τη γραμμή εντολών έχοντας πρόσβαση στο API του μέσω πρωτόκολλου HTTP στη θύρα :5984. Το API είναι ο πιο ευέλικτος, γρήγορος και ισχυρός τρόπος εκτέλεσης των λειτουργιών CRUD (Δημιουργία, ανάγνωση, ενημέρωση, διαγραφή) στη CouchDB χρησιμοποιώντας τα τυπικά αιτήματα PUT, GET, POST, DELETE. [34]

*\*Λειτουργίες CRUD στο παράρτημα*

### Παράδειγμα:

Το cURL μπορεί να χρησιμοποιηθεί για να προβάλει ο χρήστης όλα τα έγγραφα της βάσης δεδομένων που ονομάζεται *mybooks* χρησιμοποιώντας το HTTP:

`curl -X GET http://127.0.0.1:5984/mybooks/ all_docs`

### 6.3.2 Εγκατάσταση CouchDB σε μηχανήμα Windows και Σύνδεση με cURL

Download CouchDB 2.1.1 Windows (x64)

Download CURL windows 64-bit

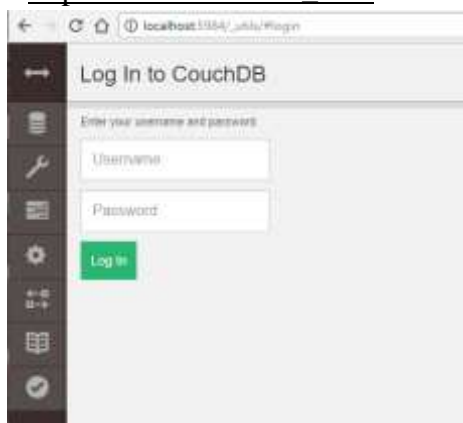


Εικόνα 6.2: CouchDB 2.1.1



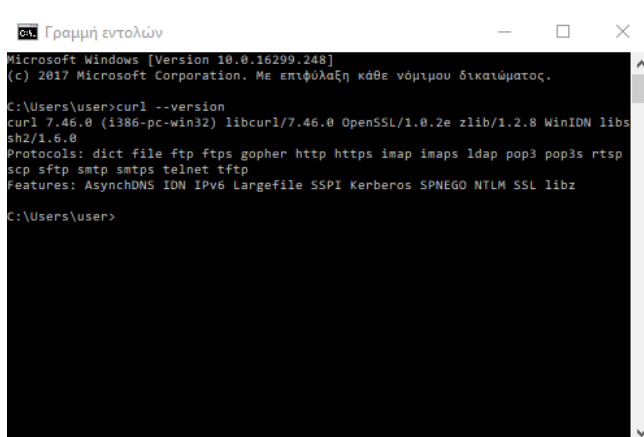
Εικόνα 6.3: cURL

Γραφικό Περιβάλλον  
Fauxton Program- Web GUI  
[http://localhost:5984/\\_utils](http://localhost:5984/_utils)



Εικόνα 6.4: Fauxton Program-Web GUI

Περιβάλλον cmd cURL -version



Εικόνα 6.5: cmd cURL -version

### 6.3.3 Σύνδεση Business Evolution με την CouchDB μέσω cURL

Σε αυτή την φάση η καταγραφή των πληροφοριών του χρήστη, ας πούμε καλύτερα ο “περίπατος” του χρήστη, έχει ξεκινήσει ήδη με το άνοιγμα της φόρμας Είδη.

Στην συνέχεια οι ενέργειές του καταγράφονται διαδοχικά και η αποθήκευση αυτών των πληροφοριών στην βάση δεδομένων CouchDB θα επιτευχθεί με την ενέργεια της Αποθήκευσης Εγγραφής.

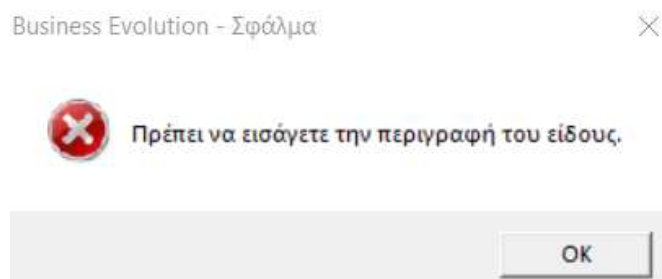
Σε περίπτωση ακύρωσης, τα δεδομένα δεν αποθηκεύονται. Κατά συνέπεια δεν λαμβάνονται υπόψη.

Με λίγα λόγια λοιπόν, η πληροφορία που θα λάβει η βάση δεδομένων θα αφορά την συμπεριφορά ενός συγκεκριμένου χρήστη για μια συγκεκριμένη χρονική στιγμή, από το άνοιγμα της φόρμα Είδη μέχρι την αποθήκευση της εγγραφής.

Για την αποθήκευση απαραίτητη είναι ενημέρωση των τριών βασικών πεδίων

**Κωδικός Είδους, Περιγραφή, Κατηγορία είδους** καθώς και η τήρηση κανόνων του κάθε πεδίου για παράδειγμα το πεδίο Κωδικός Είδους πρέπει να έχει μοναδικό κωδικό για αυτό και απαγορεύει τον διπλότυπο κωδικό.

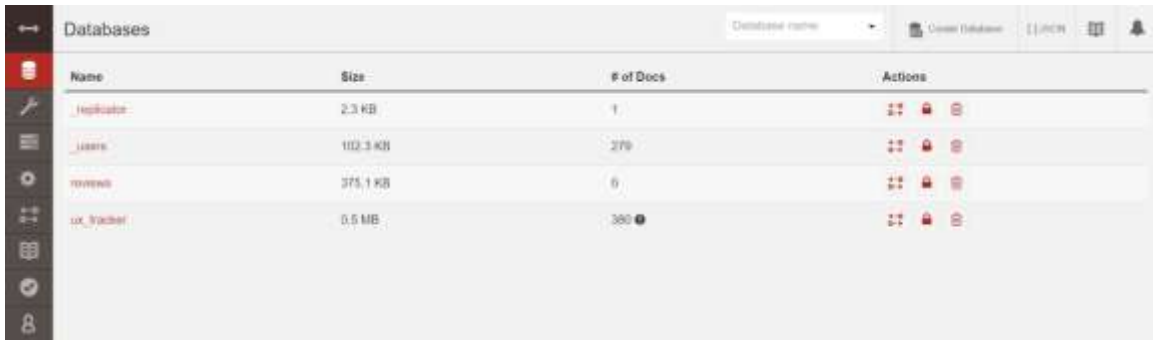
Διαφορετικά δεν επιτυγχάνεται η ενεργεία της αποθήκευσης λαμβάνοντας μήνυμα λάθους.



Εικόνα 6.7 Μήνυμα λάθους

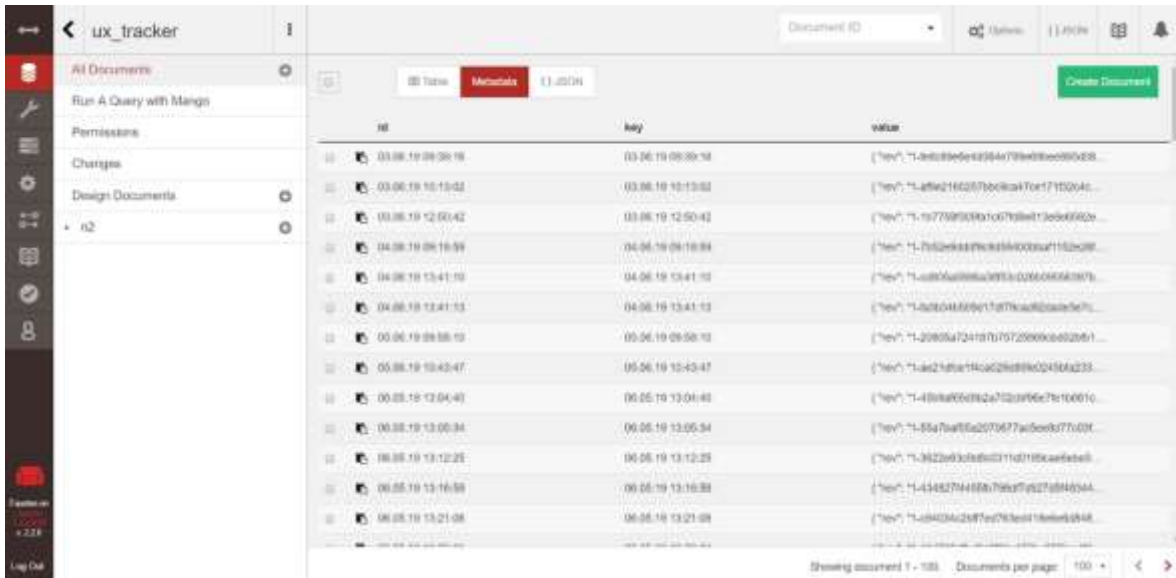
Η αποστολή και η αποθήκευση των δεδομένων θα γίνεται στην μη σχεσιακή βάση δεδομένων CouchDB. Η CouchDB χρησιμοποιεί το Fauxton ως διεπαφή του χρήστη και κάνει την διαχείριση των δεδομένων πολύ πιο εύκολη και κατανοητή.

Η βάση δεδομένων στο εργαλείο Fauxton GUI της CouchDB εμφανίζεται κάπως έτσι:



Εικόνα 6.8 Fauxton GUI

Στη συγκεκριμένη περίπτωση, η βάση δεδομένων που αποθηκεύονται τα δεδομένα για την συμπεριφορά του χρήστη για το Business Evolution ERP είναι η **ux\_tracker** και τα δεδομένα έχουν την παρακάτω μορφή:



Εικόνα 6.9 Η Βάση «ux\_tracker» στο Fauxton GUI

- Η εντολή **executeprocess**

```

executeprocess('curl -X PUT
http://195.201.222.232:5984/ux_tracker/' +
stringreplace(DT, ' ', '%20', setof(['rfReplaceAll']))) +
' -d ' +
''{
"UserName\": \"' + conv_UserName + '\",
"StationName\": \"' + conv_StationName + '\",
"DateTime\": \"' + DT + '\",
"DayofWeek\": \"' + Day + '\",
"FormName\": \"' + Self.Name + '\",
"Tracker\": \"' + mystring + '\",

```

```
\\"Tracker2\\": \\" + str + '\\",  
\\"Timer\\": \\" + timerstring + 'End' + '\\",  
\\"Timer2\\": \\" + timer2 + '\\"  
}\" ,false);
```

Με την `executeprocess` επιτυγχάνεται η σύνδεση μεταξύ Business Evolution και CouchDB

Το document Couch δεν δέχεται σαν δεδομένα το κενό. Γι' αυτόν τον λόγο αντικαθιστούμε το κενό με τον χαρακτήρα του κενού (Space %20).

```
stringreplace(DT, ' ', '%20', setof(['rfReplaceAll'
```

Επιπλέον, το Couch document δεν δέχεται σαν δεδομένα τους ελληνικούς χαρακτήρες επομένως οι μεταβλητές με τυχόν ελληνικούς χαρακτήρες, όπως για παράδειγμα το `username` και το `StationName` δεν θα περνούσαν στο Couch document και η μεταβλητή θα παρέμενε κενή.

Για τον λόγο αυτό έχει γίνει μετατροπή των ελληνικών χαρακτήρων με του αντίστοιχους αγγλικούς.

Το `conv_UserName` **Φανή** θα αντιστοιχιστεί με αυτό του **Fani**.

Το `conv_StationName` **αβγ** θα αντιστοιχιστεί με αυτό του **abg**

#### Σημείωση:

Η CouchDB υποστηρίζει utf-8 ενώ το cmd 1253. Έτσι χρησιμοποιήσαμε τη `dbo.fn_converset` η οποία μετατρέπει τους ελληνικούς χαρακτήρες σε αγγλικούς.

## 6.4 Περιγραφή και επεξήγηση ενός CouchDB document

Η CouchDB χρησιμοποιεί ένα χώρο αποθήκευσης εγγράφων με δεδομένα που παρουσιάζονται στη μορφή JSON.

Ένα τυπικό έγγραφο της Couch έχει αυτή την μορφή:

Εικόνα 6.10 έγγραφο Couch

```
1 {  
2   "_id": "14-06-19 01:22:28",  
3   "_rev": "1-13f09d9e39e502e3440550880621ac3",  
4   "UserName": "ADMINISTRATOR",  
5   "StationName": "BePAC3",  
6   "DateTime": "14-06-19 01:22:28",  
7   "DayOfWeek": "Friday",  
8   "FormName": "CmsBrandRec",  
9   "Tracker": "Start @ 0:01 @ 1:11 @ 0:20 @ 0:10 @ 2:21 @ 2:31 @ 2:42 @ 5:45 @ 1:20 @ 0:50 @ 0:10 @ 0:1 @ final state",  
10  "Tracker2": "Stokodino @ Invert @ TabSheet1 @ TabSheet5 @ TabSheet4 @ TabSheet3 @ TabSheet2 @ Save",  
11  "Timer": "Start @ 0:7 @ 1:2 @ 2:2 @ 2:2 @ 3:2 @ 4:3 @ 20:2 @ 38:0 @ 40:1 @ 0:1 @ end",  
12  "Timer2": "00 seconds"  
13 }
```

Σε αυτό το document καταγράφονται χρήσιμες πληροφορίες με απώτερο σκοπό την συλλογή στατιστικών στοιχείων για την εξαγωγή πρακτικών συμπερασμάτων.

Πιο αναλυτικά, οι πληροφορίες αυτές είναι:

**Username:** Το Ψευδώνυμο που χρησιμοποιεί ο κάθε χρήστης.

Το οποίο θα μας βοηθήσει να κατηγοριοποιήσουμε τα δεδομένα που συγκεντρώνουμε ανά χρήστη

**StationName:** Το όνομα του μηχανήματος από όπου πραγματοποιείται η εγγραφή.

**DateTime:** Η ακριβής ημερομηνία και ώρα που καταγράφεται η εγγραφή.

**DayOfWeek :** Η ημέρα της Εβδομάδας που καταγράφεται η εγγραφή.

**FormName:** Το όνομα της φόρμας που επεξεργάζεται ο χρήστης.

### **Tracker**

Καθώς ο χρήστης περιηγείται στην φόρμα και βρίσκεται σε κατάσταση επεξεργασίας και εισαγωγής δεδομένων, καταγράφεται σταδιακά η συμπεριφορά του σε μια μεταβλητή

**Tracker** και δημιουργείται μια ακολουθία, τύπου String, από τριπλέτες. (N<sub>1</sub>:N<sub>2</sub>:N<sub>3</sub>) όπου N -> Number .

Το περιεχόμενο αυτής της μεταβλητής είναι ουσιαστικά ο περίπατος του χρηστή μέσα στην φόρμα. Ενεργοποιείται (Start) με την έναρξη της φόρμας και τερματίζει (Final State) με την ενεργεία της αποθήκευσης.

Η κάθε τριπλέτα μεταξύ **Start & Final state** δηλώνει τα εξής:

- **N1:** τον τρόπο με τον οποίο έχεις ενεργοποιηθεί ένα πεδίο.

Μπορεί να πάρει τις εξής τιμές :

**0** Για αυτόματη- προεπιλεγμένη ενεργοποίηση

**1** Για ενεργοποίηση με **Mouse Click**

**2** Για ενεργοποίηση με **Enter**

**3** Για ενεργοποίηση με **Tab**

**4** Για ενεργοποίηση με **Shift\_Tab**

**5** Για ενεργοποίηση με **Space**

- **N2:** τον αριθμό του πεδίου (Τα πεδία είναι αριθμημένα διαδοχικά κατά το πάτημα του Tab)

Μπορεί να πάρει τις τιμές από **1-73**, όσα δηλαδή και τα επεξεργάσιμα πεδία της φόρμας “Είδη”.

- **N3:**

- Textbox:

- Αν έχει γίνει καταγραφή δεδομένων σε “Textbox” πεδίο αυτό τότε παίρνει την τιμή **1**
- Αν το “Textbox” πεδίο παραμένει κενό τότε παίρνει την τιμή **0**.

- Checkbox:

- Αν το πεδίο είναι Checked τότε παίρνει την τιμή **2**.
- Αν το πεδίο είναι Unchecked τότε παίρνει την τιμή **3**.

Για να είναι εμφανής ο διαχωρισμός μεταξύ των τριπλέτων εισάγεται το σύμβολο **&** μετά το πέρας κάθε ενέργειας.



Παράδειγμα:

**"Tracker": "Start & 0:0:1 & 3:1:1 & 3:2:0 & 4:1:0 & 2:2:1 & 2:3:1 & 2:4:2 & 5:4:3 & 1:28:0 & 0:30:1 & 0:46:0 & 0:0:1 & Final State",**

Μετάφραση της Ακολουθίας:

Start

&

Αυτόματη εισαγωγή στο πεδίο 0 (Κωδικός Είδους), πραγματοποιείται εισαγωγή δεδομένων

&

Εισαγωγή στο πεδίο 1 (Κωδικός 2) με το πλήκτρο Tab, πραγματοποιείται εισαγωγή δεδομένων

&

Εισαγωγή στο πεδίο 2 (Περιγραφή) με το πλήκτρο Tab, το πεδίο παραμένει κενό

&

Επιστροφή στο πεδίο 1 (Κωδικός 2) με το πλήκτρο Shift\_Tab, το πεδίο γίνεται κενό

&

Εισαγωγή στο πεδίο 2 (Περιγραφή) με το πλήκτρο Enter, πραγματοποιείται εισαγωγή δεδομένων

&

Εισαγωγή στο πεδίο 3 (Κατηγορία Είδους) με το πλήκτρο Enter, πραγματοποιείται εισαγωγή δεδομένων

&

Εισαγωγή στο πεδίο 4 (Σύνθετο) με το πλήκτρο Enter, το checkbox είναι unchecked

&

Αλλαγή κατάστασης στο πεδίο 4 (Σύνθετο) με το πλήκτρο Space, το checkbox είναι checked

&

Εισαγωγή στο πεδίο 28 (Σχόλια) με Mouse Click, το πεδίο παραμένει κενό

&

Αυτόματη εισαγωγή στο πεδίο 30 (Κωδικός Εμπορ. Διαχείρισης), πραγματοποιείται εισαγωγή δεδομένων

&

Αυτόματη εισαγωγή στο πεδίο 46 (Ακέραιος 1), πραγματοποιείται εισαγωγή δεδομένων

&

Αυτόματη εισαγωγή στο πεδίο 0 (Κωδικός Είδους), πραγματοποιείται εισαγωγή δεδομένων  
&

Final State

... Τέλος Ενεργειών

## **Tracker2**

Όσο αφορά τις ενέργειες που γίνονται εκτός των πεδίων, όπως για παράδειγμα Νέα Εγγραφή, Αναζήτηση, Αποθήκευση κτλ, Όσο, δηλαδή, ο χρήστης βρίσκεται σε κατάσταση προβολής της φόρμας οι ενέργειες αυτές καταγράφονται σταδιακά σε μια άλλη μεταβλητή Tracker2 (τύπου String) και έτσι δημιουργείται μια δεύτερη ακολουθία από ενέργειες του χρήστη. Αυτή την φορά δεν αποτελείται από τριπλέτες, μιας και ο χρήστης δεν περιηγείται μεταξύ των πεδίων.

Παράδειγμα:

**"Tracker2": "BtnRecIns & Insert & TabSheet1 & TabSheet5 & TabSheet4 & TabSheet1 & btnRecOK & Save",**

Μετάφραση της Ακολουθίας:

Ενεργοποίηση της κατάστασης Εισαγωγή δεδομένων με το button Insert από το menu της φόρμας

&

Περιήγηση στην Καρτέλα *Στοιχεία Είδους* (TabSheet1)

&

Περιήγηση στην Καρτέλα *Άλλα Στοιχεία* (TabSheet5)

&

Περιήγηση στην Καρτέλα *Πρόσθετα* (TabSheet4)

&

Επιστροφή στην Καρτέλα *Στοιχεία Είδους* (TabSheet1)

&

Αποθήκευση με το button Insert από το menu της φόρμας.

## **Timer**

Η τελευταία ακολουθία μετρά τον χρόνο περαίωσης της κάθε ενέργειας.

- **N1:** τον αριθμό του πεδίου (Τα πεδία είναι αριθμημένα διαδοχικά κατά το πάτημα του Tab)  
Μπορεί να πάρει τις τιμές από **1-73**, όσα δηλαδή και τα επεξεργάσιμα πεδία της φόρμας “Είδη”.
- **N2:** Ο χρόνος περαίωσης την ενέργειες σε second.

Παράδειγμα:

"Timer": "Start & 0:7 & 1:3 & 2:1 & 1:2 & 2:1 & 3:2 & 4:3 & 28:1 & 30:8 & 46:1 & 0:5 & End",

Μετάφραση:

Για να ολοκληρωθεί η ενεργεία του πεδίου “0” πέρασες 7 δευτερόλεπτα

&

Για να ολοκληρωθεί η ενεργεία του πεδίου “1” πέρασαν 3 δευτερόλεπτα

&

.

.

(κοκ)

**Timer2**

Ο χρόνος περαίωσης όλης της καταγραφής σε second. Με αυτό εννοούμε από την ενεργοποίηση κάποιας ενεργείας στην φόρμα μέχρι την αποθήκευση της εγγραφής.

Παράδειγμα:

"Timer2": "40 seconds"

Αυτό σημαίνει πως η εγγραφή περατώθηκε σε 40 δευτερόλεπτα.

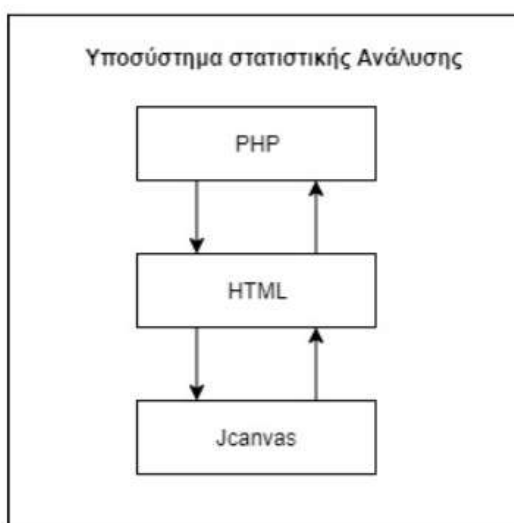
Παρατήρηση: Συνήθως συμπίπτει με το άθροισμα των χρόνων της ακολουθίας Tracker2, κατά προσέγγιση στο δευτερόλεπτο.

## ΚΕΦΑΛΑΙΟ 7

### Υποσύστημα στατιστικής ανάλυσης

#### 7.1 Σχεδιασμός Υποσυστήματος

Για την επεξεργασία και την οπτικοποίηση των δεδομένων της συμπεριφοράς του χρήστη σε πραγματικό χρόνο χρησιμοποιήθηκαν οι παρακάτω τεχνολογίες λογισμικού:



Εικόνα 7.26: Διάγραμμα υποσυστήματος

#### HTML

Η HTML είναι μια γλώσσα προγραμματισμού και τα αρχικά HTML είναι το ακρωνύμιο των λέξεων Hyper Text Markup Language (γλώσσα μορφοποίησης υπερκειμένου) και είναι η βασική γλώσσα δόμησης σελίδων του Web. Χρησιμοποιείται για να βελτιώνει την εικόνα-εμφάνιση τμημάτων κειμένου. Αρχικά δημιουργήθηκε απλά για τη μορφοποίηση κειμένου αλλά πλέον εξελίχθηκε και με τις σχεδιαστικές ικανότητες που παρέχει δίνει τη δυνατότητα προσθήκης εικόνων βίντεο και ήχου σε αυτά. Αυτά τα κείμενα είναι οι σελίδες ιστού. Για τη μορφοποίηση των σελίδων ιστού η γλώσσα αυτή χρησιμοποιεί έναν αριθμό από tags. Αυτά τα tags μεταφράζονται σε κατάλληλα χαρακτηριστικά από τους διακομιστές ιστού κατά το άνοιγμα ενός αρχείου HTML ώστε να έχουμε το επιθυμητό αποτέλεσμα τόσο στη λειτουργικότητα όσο και στην εμφάνιση της σελίδας.[44]

## **PHP**

Η υλοποίηση των scripts στηρίζεται στην δυναμική γλώσσα προγραμματισμού PHP. Η PHP αποτελεί μια γλώσσα προγραμματισμού για τη δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο σχεδιασμένη ειδικά για το Web. Τα αρχεία της PHP μπορεί να περιέχουν κώδικα HTML, CSS, JavaScript και PHP, ο κώδικας εκτελείται στον διακομιστή ιστού και το αποτέλεσμα επιστρέφει στον πρόγραμμα περιήγησης ιστού ως απλές σελίδες HTML.[45]

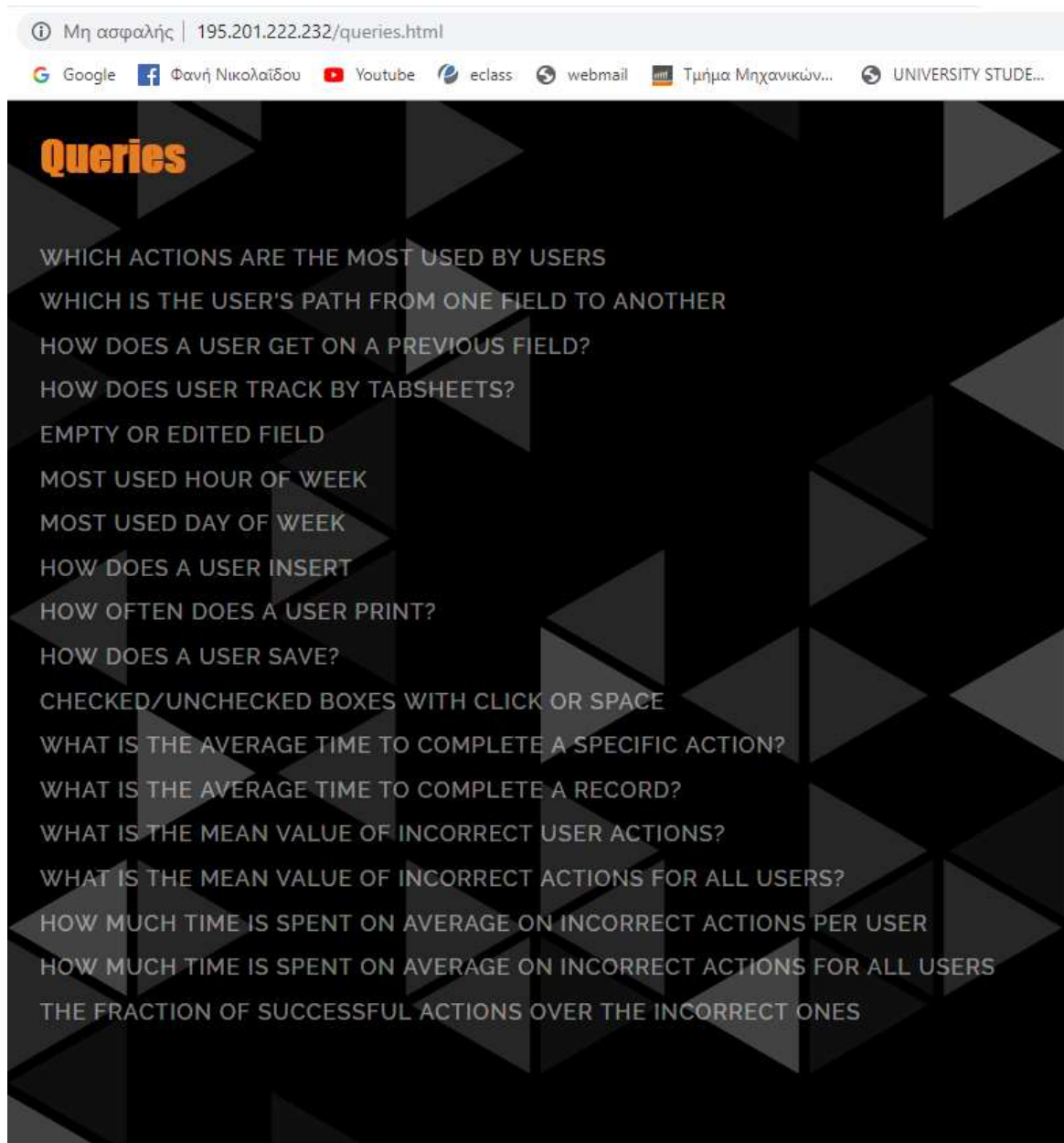
## **Jcanvas**

Η βιβλιοθήκη Jcanvas αποτελεί στοιχείο της HTML και χρησιμοποιείται για να σχεδιάζει γραφήματα μέσω της γλώσσας προγραμματισμού JavaScript.

Για την διαχείριση των εγγράφων που επεξεργάστηκαν χρησιμοποιήθηκε το εργαλείο WinSCP. Το **WinSCP** ( **Win Dows Secure Copy** ) είναι ένα ελεύθερο λογισμικό ανοικτού κώδικα SFTP για Microsoft Windows . Η κύρια λειτουργία του είναι η ασφαλής μεταφορά αρχείων μεταξύ τοπικού και απομακρυσμένου υπολογιστή . Πέρα από αυτό, το WinSCP προσφέρει βασική λειτουργία διαχείρισης αρχείων και συγχρονισμού αρχείων . Για ασφαλείς μεταφορές, χρησιμοποιεί το Secure Shell ( SSH ) και υποστηρίζει το πρωτόκολλο SCP εκτός από το SFTP.

Δημιουργήθηκε μια Html σελίδα όπου συγκεντρώσαμε όλα τα Javascript ερωτήματα που κρίθηκαν κατάλληλα για την εξαγωγή σημαντικών συμπερασμάτων. Τα παρακάτω ερωτήματα γράφτηκαν σε Html και PHP και με την βοήθεια της βιβλιοθήκης JScanvas οπτικοποιήσαμε τις μετρήσεις και δημιουργήσαμε γραφήματα για την καλύτερη ανάγνωση και κατανόηση των αποτελεσμάτων.

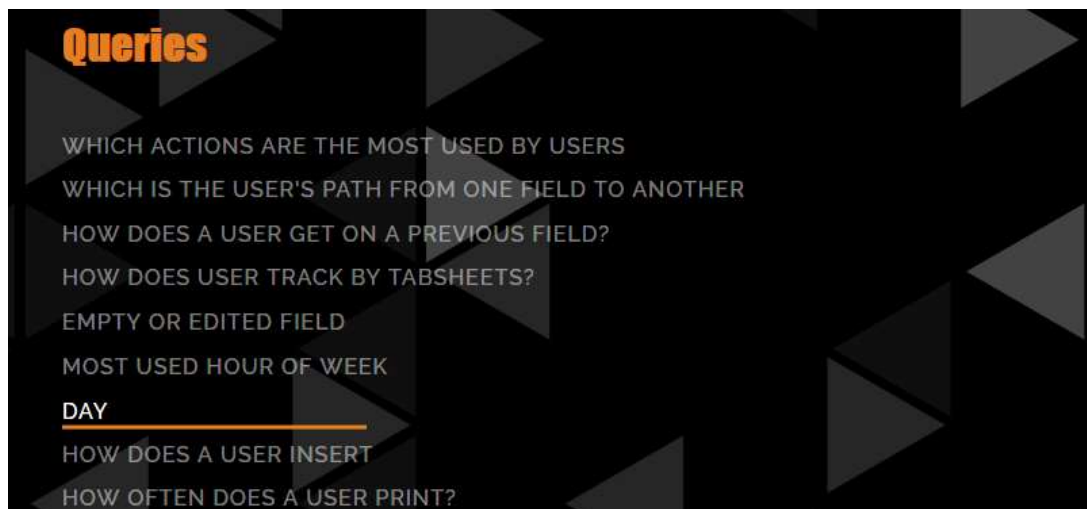
Η ηλεκτρονική διεύθυνση των ερωτημάτων: [195.201.222.232/queries.html](http://195.201.222.232/queries.html)



Εικόνα 7.1: Queries

Με ένα κλικ σε κάποιο από τα ερωτήματα θα παρουσιαστεί η απάντηση του με μορφή γραφήματος.

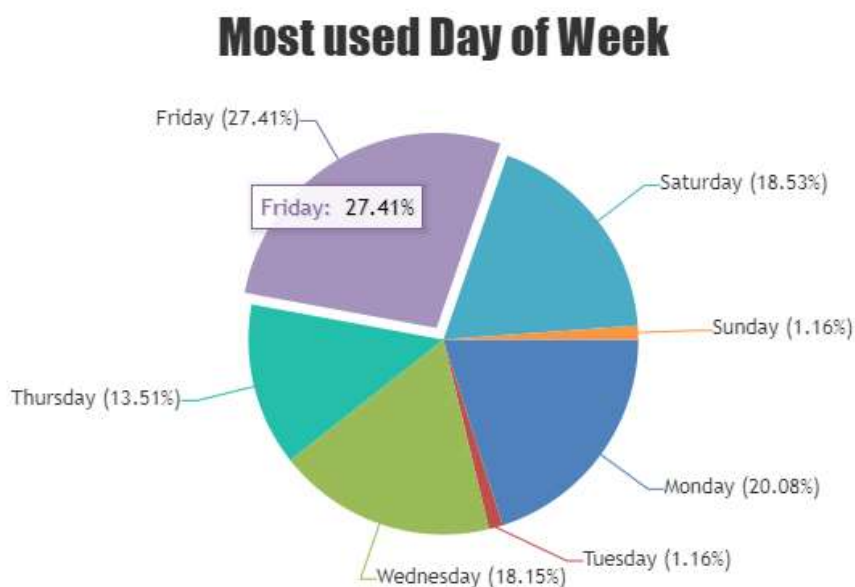
Παράδειγμα : Κλικ στο ερώτημα «Most used Day of Week»



Εικόνα 7.2: Queries:Day

*\*Παράδειγμα υλοποίησης ερωτήματος υπάρχει στο παράρτημα*

Τα αποτελέσματα των μετρήσεων δείχνουν πως η πιο “απασχολημένη” μέρα είναι η **Παρασκευή με ποσοστό 27,41%** και ακολουθεί η **Δευτέρα με 20,08%**



Εικόνα 7.3: Γράφημα

Τα δεδομένα φυσικά συνεχώς ανανεώνονται και τα στατικά στοιχεία αλλάζουν μιας και οι μετρήσεις είναι σε πραγματικό χρόνο.

## Κεφάλαιο 8

### Αξιολόγηση Ευχρηστίας

Στο κεφάλαιο αυτό παρουσιάζονται τα βασικότερα συμπεράσματα της παρούσας διπλωματικής εργασίας καθώς επίσης προτείνονται και ορισμένες επιπρόσθετες μελλοντικές επεκτάσεις του έργου που πραγματοποιήθηκε στα πλαίσια της εργασίας

#### 8.1 Ο Σκοπός για την εξαγωγή συμπερασμάτων της στατιστικής ανάλυσης

Σκοπός αυτής της διπλωματικής είναι η αυτοματοποιημένη ενημέρωση και καταγραφή δεδομένων της συμπεριφοράς του χρήστη στη βάση δεδομένων, σε πραγματικό χρόνο και στην συνέχεια μετά από συμπερασματικές δοκιμές να πραγματοποιείται στατιστική ανάλυση των δεδομένων.

Η NoSQL βάση δεδομένων CouchDB κάνει ευκολότερο τον χειρισμό μεγάλου πλήθους δεδομένων (Big Data). Έτσι συμπεραίνουμε την σημασία και την αξία των NoSQL βάσεων δεδομένων στην εποχή των Big Data, όπου η ανάγκη διαχείρισης μεγάλου όγκου δεδομένων είναι τεράστια ειδικά στο κομμάτι της αποθήκευσης δεδομένων.

Ο προσδιορισμός μέτρων ευχρηστίας της φόρμας και η μέτρηση αυτών αποσκοπούν στην σωστή αξιολόγηση της ευχρηστίας της φόρμας “Είδη” του συστήματος Business Evolution ERP. Ο τελικός στόχος της διπλωματικής είναι η στατιστική ανάλυση των δεδομένων που περιγράφουν την συμπεριφορά του χρήστη στην φόρμα “Είδη” ώστε να εξαχθούν ορισμένα συμπεράσματα σχετικά με το επίπεδο ευχρηστίας της φόρμας κατά την χρήση της και τον χρόνο ολοκλήρωσης μιας εργασίας της.

#### 8.2 Συμπερασματικές δοκιμές ευχρηστίας

Σε προηγούμενο κεφάλαιο αναφερθήκαμε στις μεθόδους αξιολόγησης της ευχρηστίας που περιλαμβάνει τις Επιθεωρήσεις ευχρηστίας, Δοκιμές ευχρηστίας και τα Ερωτηματολόγια διερεύνησης της ικανοποίησης των χρηστών.



Η μέθοδος με την οποία πραγματοποιείται η αξιολόγηση του συγκεκριμένου συστήματος εντάσσεται στις Δοκιμές ευχρηστίας και πιο συγκεκριμένα στις Συμπερασματικές δοκιμές ευχρηστίας.

Σκοπός των συμπερασματικών δοκιμών ευχρηστίας είναι να καταλήξουμε σε κάποια απόφαση για την αποδοχή (ή μη) του συστήματος ως προς την ευχρηστία του. Κατά κανόνα αυτές οι δοκιμές χρησιμοποιούνται στο τελικό στάδιο ενός έργου ανάπτυξης, ώστε να πιστοποιήσουν ότι το σύστημα ικανοποιεί τις απαιτήσεις των χρηστών. Στις συμπερασματικές δοκιμές ευχρηστίας, εξετάζεται, κατά κανόνα, η αποδοτικότητα, αποτελεσματικότητα και προσωπική ικανοποίηση του χρήστη, τα τρία χαρακτηριστικά της ευχρηστίας σύμφωνα με τον ορισμό ISO 9241.[40]

Οι συμπερασματικές δοκιμές βασίζονται σε μετρικές (ή μέτρα) ευχρηστίας (χρόνος ολοκλήρωσης εργασίας, λάθη, κ.α.), οι οποίες είναι μετρήσιμες ποσότητες της αλληλεπίδρασης των χρηστών κατά την δοκιμαστική χρήση του συστήματος. Οι μετρικές προσδιορίζονται από τον αξιολογητή, με βάση τους στόχους της αξιολόγησης, το είδος των εργασιών που θα ζητηθούν από τους χρήστες και το στατιστικό έλεγχο που θα χρησιμοποιηθεί για την εξαγωγή συμπερασμάτων

Τέλος, ένα τελευταίο πρακτικό ζήτημα είναι ότι οι μετρικές ευχρηστίας επηρεάζουν τη παρουσίαση των αποτελεσμάτων, η οποία είναι σημαντικό να είναι επίσης εύχρηστη. Οι μετρικές ευχρηστίας παρουσιάζονται με περιγραφική στατιστική και πίνακες, διαγράμματα και γραφικές παραστάσεις.

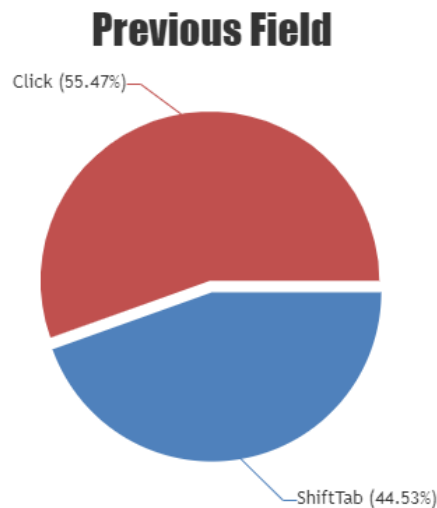
### **8.3 Στατιστική ανάλυση, συμπεράσματα και μελλοντικές εργασίες**

Στο συγκεκριμένο σύστημα στα πλαίσια της Διπλωματικής, καταγράφονται δυο εγκαταστάσεις με εικοσιπέντε χρήστες συνολικά. Η κίνηση των δεδομένων είναι καθημερινή και σε ικανοποιητικό όγκο ώστε να προκύψουν χρήσιμα αποτελέσματα. Η συλλογή δεδομένα που αποθηκεύονται στην βάση δεδομένων CouchDB βρίσκεται σε εφαρμογή περίπου έναν μήνα.

Από την ανάλυση των δεδομένων με την οπτικοποίηση τους σε γραφήματα προκύπτουν τα εξής συμπεράσματα:

- **Στο ερώτημα:** Με ποιον τρόπο ο χρήστης πηγαίνει σε προηγούμενο πεδίο, τα στατιστικά δείχνουν πως το 55,47% πηγαίνει με Click έναντι του 44.53% που χρησιμοποιεί την συντόμευση Shift\_Tab.

**Συμπέρασμα:** Αυτό βέβαια μπορεί να οφείλεται σε αρκετά πράγματα. Για παράδειγμα, είτε όντως, είναι ο ποιος εύκολος τρόπος για τους χρήστες να χρησιμοποιήσουν το ποντίκι για την επιστροφή σε προηγούμενο πεδίο, είτε ο χρήστης δεν γνωρίζει για την συντόμευση Shift\_Tab



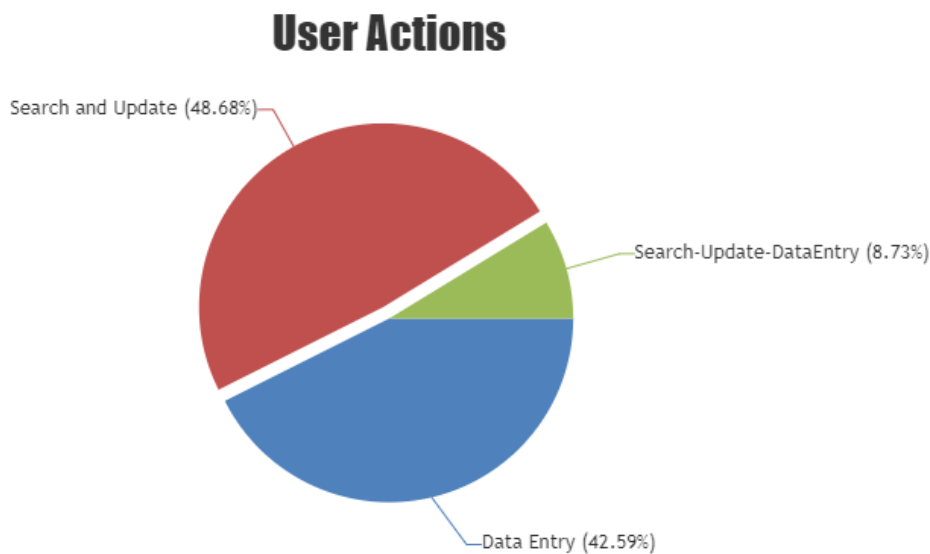
Εικόνα 7.4: Previous Field

**Ο Προγραμματιστής μελλοντικά μπορεί:**

- να χρησιμοποιήσει μια πιο χρηστική συντόμευση για την επιστροφή σε προηγούμενο πεδίο
- να επισημάνει με ένα μήνυμα πως υπάρχει δυνατότητα επιστροφής σε προηγούμενο πεδίο με το Shift\_Tab.
- είτε να καταργήσει την συντόμευση αν κρίνει πως δεν είναι χρηστική.

- **Στο ερώτημα :** Ποια ενέργεια είναι πιο συνηθισμένη χρήση της φόρμας για τους χρήστες, τα στατιστικά δείχνουν πως το 48,68% αναζητεί και επεξεργάζεται τα υπάρχοντα προϊόντα, το 42.59% καταχωρεί προϊόντα στην φόρμα, ενώ το υπόλοιπο 8,73% κάνει και τις τρεις ενέργειες πριν μια αποθήκευση.

Συμπέρασμα: αυτό σημαίνει και οι τρεις αυτές ενέργειες είναι εξίσου σημαντικές. Συνήθως όμως ο χρήστης επιλέγει χρησιμοποιήσει την φόρμα για έναν από τους δυο σκοπούς αποκλειστικά. Είτε για καταχώρηση προϊόντων είτε για την ενημέρωσή τους, όχι τόσο συχνά όμως και για τα δυο ταυτόχρονα.



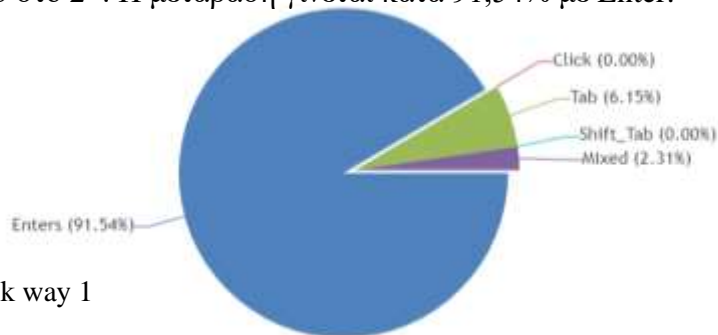
Εικόνα 7.5: User Actions

Ο Προγραμματιστής μελλοντικά μπορεί:

- Να διαμορφώσει την φόρμα ώστε να είναι πιο χρηστική προβλέποντας τον λόγο που θέλει να την χρησιμοποιήσει ο χρήστης
- Να δημιουργήσει ενέργεια μόνο για την προβολή των πληροφοριών των προϊόντων με έναν πιο ενδιαφέρον τρόπο

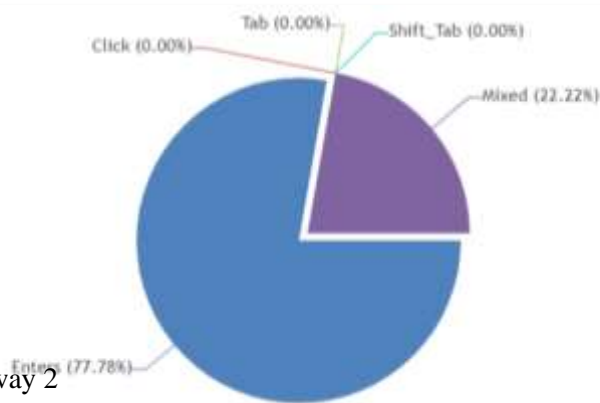
- **Ερώτημα :** Με ποιον τρόπο γίνεται η μετάβαση από ένα πεδίο σε ένα άλλο (χωρίς να έχει επεξεργαστεί αλλά πεδία στον προορισμό του);

Από το 1<sup>ο</sup> πεδίο στο 2<sup>ο</sup> : Η μετάβαση γίνεται κατά 91,54% με Enter.



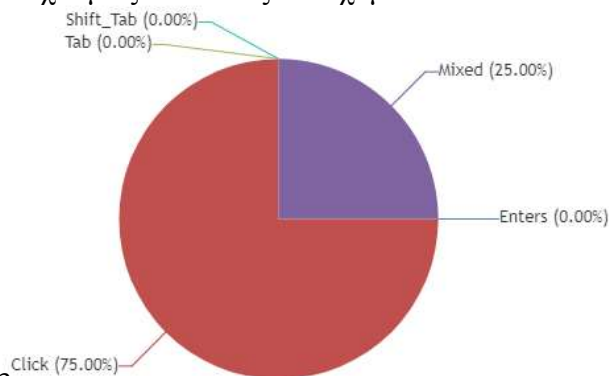
Εικόνα 7.6: Track way 1

Από το 4<sup>ο</sup> πεδίο στο 8<sup>ο</sup> : Το ποσοστό όπου η μετάβαση γίνεται με Enter μειώθηκε



Εικόνα 7.7: Track way 2

Από το 1<sup>ο</sup> πεδίο στο 30<sup>ο</sup> : η μετάβαση γίνεται συνήθως με το Click του ποντικιού, είτε και με τους δυο τρόπους, (δηλαδή χρησιμοποιώντας αρχικά το πλεκτό Enter και στην συνέχεια το κλικ ή αντίστροφα). Όχι όμως πατώντας συνεχόμενα το enter.



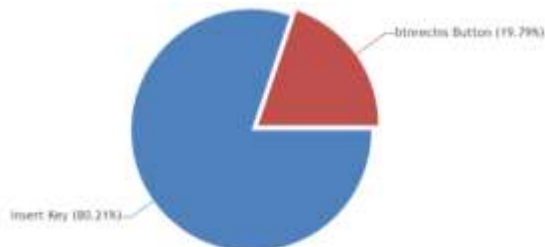
Εικόνα 7.8: Track Way 3

Προκύπτει το λογικό Συμπέρασμα, ότι όταν ο χρήστης θέλει να “καλύψει” μια μεγάλη απόσταση πεδίων χρησιμοποιεί το κλικ. Για πεδία με μικρότερη απόσταση κάνει χρήση του πλήκτρου Enter. Συμπεραίνοντας πως ο χρήστης προτιμά την χρήση του πληκτρολογίου κατά την διαδικασία επεξεργασίας της φόρμας. Ο Προγραμματιστής μελλοντικά μπορεί: Να εντάξει συντομεύσεις πληκτρολογίου για περισσότερες ενέργειες.

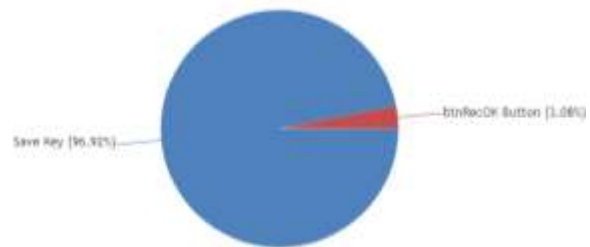
- **Ερώτημα:** Με ποιον τόπο κάνει Νέα εγγραφή και αποθήκευση δεδομένων ο χρήστης;

Ο χρήστης έχει από δυο επιλογές:

Για Νέα Εγγραφή: Το πλήκτρο “INSERT” η το button Insert του μενού της φόρμας  
Για την αποθήκευση: Το πλήκτρο F2 η το button Save του μενού της φόρμας..



Εικόνα 7.9: Insert



Εικόνα 7.10: Save

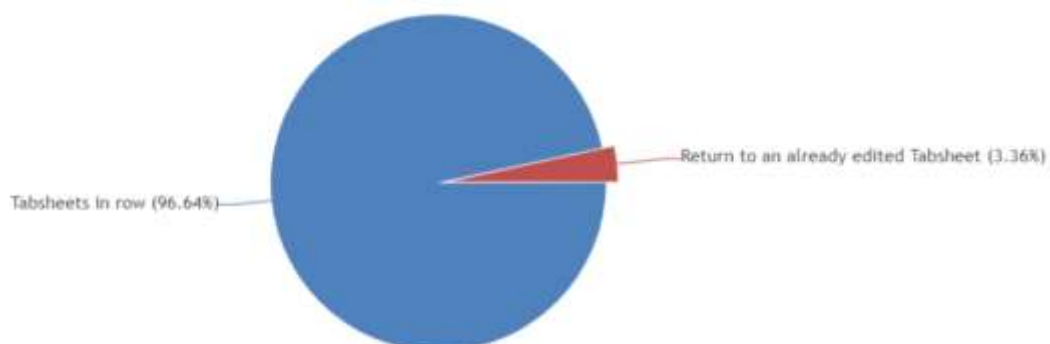
**Συμπέρασμα:** Όπως και στο προηγούμενο ερώτημα ο χρήστης και εδώ προτιμάει τις συντομεύσεις πληκτρολογίου από το κλικ των κουμπιών του μενού.

- **Ερώτημα:** Πως κινείται ο χρήστης ανά τις καρτέλες της φόρμας.

Το Ποσοστό που περιηγείται ανά των καρτελών είναι μόλις 3,36% αυτό σημαίνει πως ο χρήστης προτίμα μεταβαίνει στην επόμενη καρτέλα αφού ολοκληρώσει τις επιθυμητές ενέργειες στη τρέχουσα

**Συμπέρασμα:** Το ποσοστό αυτό μπορεί να δείξει πόσο καλά ομαδοποιημένα (ανά καρτέλες) είναι τα πεδία προς επεξεργασία. Η περίπτωση αυτή μοιάζει να είναι ιδανική.

Σε διαφορετική περίπτωση, ο Προγραμματιστής μελλοντικά μπορεί: Να διαμορφώσει καλύτερα την δομή των καρτελών της φόρμας χωρίς να προκαλεί σύγχυση στον χρήστη



Εικόνα 7.11: Tabsheet

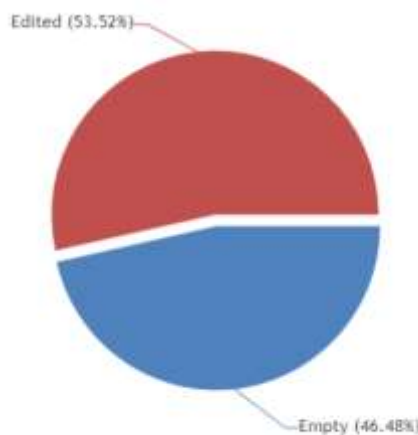
- **Ερώτημα:** Σε τι ποσοστό παραμένει κενό ένα συγκεκριμένο πεδίο;

Στο 1<sup>ο</sup> Πεδίο που αντιστοιχεί στο πεδίο “Κωδικός Είδους” γίνεται σχεδόν πάντα καταχώρηση



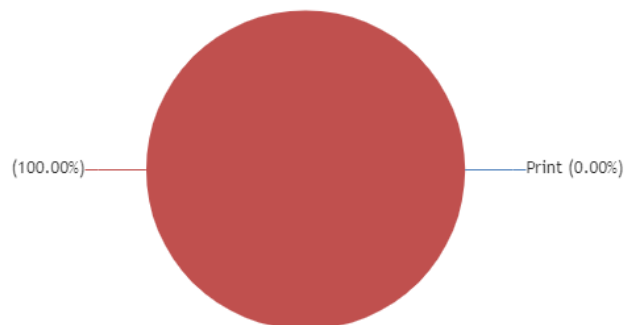
Εικόνα 7.12: Blank Field 1

Στο 5<sup>ο</sup> Πεδίο που αντιστοιχεί στο πεδίο “Ομάδα” η καταχώρηση γίνεται περίπου τις μισές φορές



Εικόνα 7.13: Blank Field 2

Λειτουργία Εκτύπωσης: Η Εκτύπωση φαίνεται δεν είναι αρκετά σημαντική για τον χρήστη



Εικόνα 7.14: Print

**Συμπέρασμα:** Από αυτά τα στατιστικά μπορούμε να εξάγουμε χρήσιμες πληροφορίες για το ποσό σημαντικό ή όχι θεωρεί ο χρήστης ένα πεδίο ή μια λειτουργία.

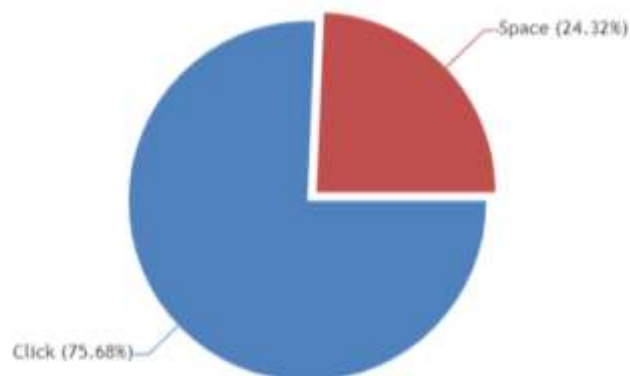
**Ο Προγραμματιστής μελλοντικά μπορεί:**

- Να διαμορφώσει την δομή της φόρμας ανάλογα την σημαντικότητα των πεδίων και των λειτουργιών ώστε να απαλλάξει τον χρήστη από επιπλέον χρόνο και κόπο.
- Να εξάγει συμπεράσματα προς χρήση άλλου project

- **Ερώτημα:** Με ποιον τρόπο ο χρήστης ενεργοποιεί/απενεργοποιεί ένα Checkbox;

Ο χρήστης έχει πάλι δυο επιλογές. Αυτή του πληκτρολογίου με το πλήκτρο Space η το αριστερό κλικ του ποντικιού. Σε προηγούμενα ερωτήματα συμπεράναμε πως ο χρήστης προτιμάς τη χρήση του πληκτρολογίου όταν έχει την δυνατότητα επιλογής. Εδώ παρατηρείται πως το μεγαλύτερο ποσοστό των checks σε checkbox γίνεται με κλικ.

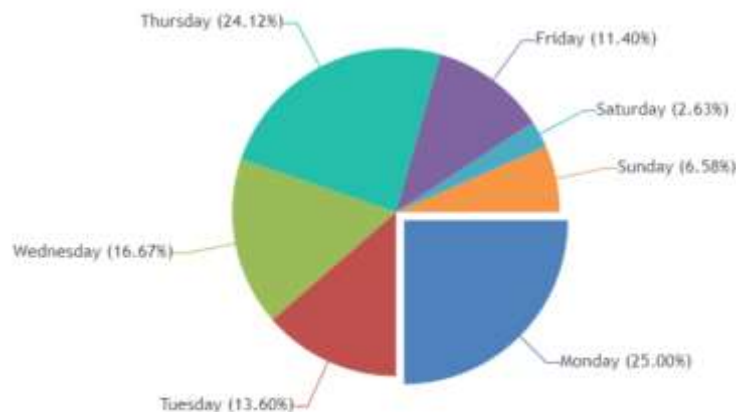
Συμπεράσματα: Ίσως σε αυτή την περίπτωση ο χρήστης να προτιμάει το κλικ. Ίσως να έχει άγνοια της λειτουργίας του πλήκτρου Space.



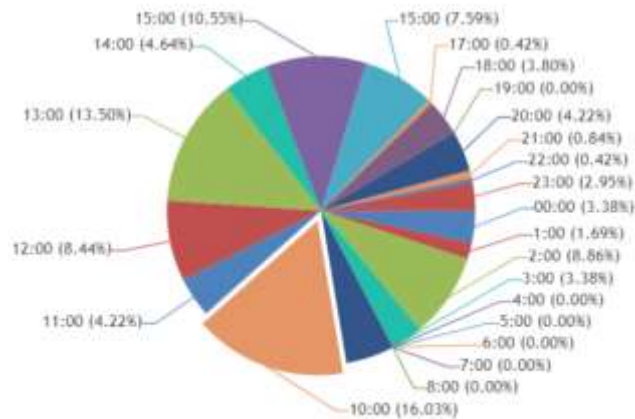
Εικόνα 7.15: Checkbox

Ο Προγραμματιστής μελλοντικά μπορεί: Να ενημερώνει τον χρήστη για την λειτουργία των συντομεύσεων καθώς ο χρήστης θα κάνει focus σε ένα πεδίο ή μια λειτουργία.

- **Ερώτημα:** Ποια είναι η πιο δημοφιλής ώρα και μέρα που περιηγείται ο χρήστης στην φόρμα



Εικόνα 7.16: Day



Εικόνα 7.17: Time

Η πιο δημοφιλής μέρα είναι η Δευτέρα και η πιο δημοφιλής ώρα είναι 10:00 π.μ. Αυτά τα στατιστικά ίσως να αφορούν κάποιον άλλον τομέα πέραν την ευχρηστίας του συστήματος. Δεν παύουν όμως να είναι χρήσιμα.

Στον τομέα τις ευχρηστίας θα ήταν πολύ χρήσιμο να εκτιμηθεί ο απαιτούμενος χρόνος για την ολοκλήρωση των επιθυμητών ενεργειών κατά την διάρκεια της χρήσης ενός συστήματος, αλλά και να υπολογιστεί ο άσκοπα δαπανημένος χρόνος και κόπος σε λανθασμένες ενέργειες. Αυτό μπορεί να προκύψει είτε από ένα δύσκολο στην χρήση σύστημα το οποίο θα είναι επιρρεπές στα λάθη, είτε από ένα χρονοβόρο στην χρήση του σύστημα. Αυτό θα χτίσει τις βάσεις για τον εντοπισμό και την κατανόηση των προβλημάτων και των εμποδίων που αντιμετωπίζει ο χρήστης.

- Ερώτημα:** Ποιος είναι ο μέσος Όρος λανθασμένων χειρισμών ανά χρήστη;
  - the average incorrect actions of user system administrator: 34
  - the average incorrect actions of user kikh: 4
  - the average incorrect actions of user apostolhw lagoylow: 7
  - the average incorrect actions of user kvsths: 0
  - the average incorrect actions of user ADMINISTRATOR: 60
  - the average incorrect actions of user administrator: 5
  - the average incorrect actions of user retail aulhtika: 14
  - the average incorrect actions of user retail edesshs 29: 3
  - the average incorrect actions of user System Administrator: 11
  - the average incorrect actions of user nikos: 2
  - the average incorrect actions of user Nikos: 4
  - the average incorrect actions of user ueopisth: 2

Εικόνα 7.18: incorrect actions per user



- **Ερώτημα:** Πόσος χρόνος δαπανάται κατά μέσο όρο σε λανθασμένες ενέργειες ανά χρήστη;

time wasted on average on incorrect actions for user system administrator: 944 seconds  
time wasted on average on incorrect actions for user kikh: 177 seconds  
time wasted on average on incorrect actions for user apostolhw lagopoylow: 55 seconds  
time wasted on average on incorrect actions for user kvsths: 25 seconds  
time wasted on average on incorrect actions for user ADMINISTRATOR: 1108 seconds  
time wasted on average on incorrect actions for user administrator: 35 seconds  
time wasted on average on incorrect actions for user retail aulhtika: 174 seconds  
time wasted on average on incorrect actions for user retail edesshs 29: 19 seconds  
time wasted on average on incorrect actions for user System Administrator: 91 seconds  
time wasted on average on incorrect actions for user nikos: 10 seconds  
time wasted on average on incorrect actions for user Nikos: 4 seconds  
time wasted on average on incorrect actions for user ueopisth: 10 seconds

Εικόνα 7.19: wasted time per user

*Σημείωση: Στην προκειμένη περίπτωση λάθος χειρισμός θεωρείται η επιστροφή του χρήστη σε κάποιο πεδίο προς διόρθωση, πριν την αποθήκευση των πληροφοριών.*

- **Ερώτημα:** Ποιος λόγος που προκύπτει από τις σωστές ενέργειες προς τις λανθασμένες;

---

right over wrong actions: 20.09

Εικόνα 7.20: Fraction Right/Wrong

- **Ερώτημα:** Μέσα σε πόσο χρόνος ολοκληρώνεται μια ενέργεια κατά μέσο όρο; Ολοκλήρωση καταχώρησης του πεδίου Περιγραφή. Δέχεται τιμές τύπου “text”

---

Enter a number of field:   
The average time to complete this action is: 11.2 seconds

Εικόνα 7.21: Time for an action 1

- **Ερώτημα:** Ολοκλήρωση καταχώρησης του πεδίου Ομάδα. Δέχεται τιμές από λίστα.

---

Enter a number of field:   
The average time to complete this action is: 4.56 seconds

Εικόνα 7.22: Time for an action 2

- **Ερώτημα:** Μέσα σε πόσο χρόνο ολοκληρώνεται μια εγγραφή κατά μέσο όρο;

---

The average time to complete a record is: 132.54 sec

Εικόνα 7.23: Completed Record

Συμπέρασμα: Τα πεδία με προκαθορισμένες τιμές τύπου λίστας ελαχιστοποιούν το χρόνο διεκπεραίωσης μιας εργασίας έναντι των χειρόγραφων συμβολοσειρών.

## Κεφάλαιο 9

### Συμπεράσματα

#### 9.1 Σύνοψη

Στην παρούσα διπλωματική μελετήθηκε η αξιολόγηση της ευχρηστίας της διεπαφής ενός εμπορικού ERP συστήματος μέσα από την αυτοματοποιημένη καταγραφή συμπεριφοράς των χρηστών του. Αρχικά μελετήθηκε η έννοια της ευχρηστίας και καθορίστηκε το πόσο σημαντικός είναι ο ρόλος της αξιολόγησης της για τον σχεδιασμό εύκολων στην χρήση συστημάτων. Στην συνέχεια εξετάστηκαν οι τεχνολογίες με τις οποίες υλοποιήθηκε το σύστημα για την πιο αξιόπιστη απόδοση των αποτελεσμάτων και τονίστηκε η ανάγκη για διαχείριση μεγάλου όγκου δεδομένων σε πραγματικό χρόνο. Στο πρακτικό επίπεδο αναλύθηκε η αρχιτεκτονική της φόρμας του εμπορικού ERP συστήματος και έγινε κατανοητός ο τρόπος με τον οποίο περιηγείται και επεξεργάζεται ο χρήστης μία επιλεγμένη φόρμα. Η καταγραφή των πληροφοριών για την συμπεριφορά των χρηστών έγινε υπό πραγματικές συνθήκες και δημιούργησε ερωτήματα τα οποία και μελετήθηκαν μέσω στατιστικής ανάλυσης.

Η οπτικοποίηση των στατιστικών που υλοποιήθηκε έκανε κατανοητά τα αποτελέσματα που προέκυψαν, κατανοώντας με αυτόν τον τρόπο την συμπεριφορά του χρήστη κατά τη χρήση της διεπαφής αλλά και το επίπεδο ευχρηστίας του περιβάλλοντος. Απώτερος στόχος της αξιολόγησης ευχρηστίας του συστήματος είναι η βέλτιστη ανασχεδίαση της διεπαφής του για την μεγιστοποίηση της αποδοτικότητας και της προσωπικής ικανοποίησης των χρηστών.

Τα συμπεράσματα που αφορούν τον χρήστη υποστηρίζουν πως όταν δίνεται στον χρήστη η επιλογή για τον τρόπο που θα ολοκληρώσει μια ενέργεια συνήθως προτιμάει να χρησιμοποιεί το πληκτρολόγιο έναντι της συσκευής του ποντικιού όταν βρίσκεται σε κατάσταση επεξεργασίας της φόρμας. Παρατηρήθηκε επίσης πως παρόλο που χρησιμοποιεί αρκετά το πληκτρολόγιο, δεν κάνει συχνή χρήση κάποιων συντομεύσεων πληκτρολογίου. Παράλληλα, υποδείχτηκε το αναμενόμενο συμπέρασμα πως ο χρήστης θα χρησιμοποιήσει το ποντίκι όταν θέλει να μεταβεί σε ένα “μακρινό” πεδίο.

Για να βελτιστοποιηθεί το επίπεδο ευχρηστία σε αυτόν το τομέα, θα ήταν αρκετά χρήσιμο να δημιουργηθούν περισσότερες συντομεύσεις πληκτρολογίου αφού ο χρήστης έδειξε την προτίμηση του στη χρήση του πληκτρολογίου. Εξίσου σημαντική είναι όμως και ενημέρωση και εκπαίδευση των χρηστών με την δυνατότητα χρήσης τέτοιων

συντομεύσεων. Αυτό μπορεί να συμβεί με την εμφάνιση μηνυμάτων καθώς ο χρήστης πλησιάζει σε μια ενέργεια (κάνει focus).

Ο χρήστης χρησιμοποιεί την φόρμα πιο συχνά για προβολή και ενημέρωση μιας υπάρχουσας εγγραφής πάρα για την καταχώρηση μιας νέας εγγραφής. Ίσως να ήταν πιο εύχρηστο το σύστημα αν γινόταν διαχωρισμός των ενεργειών και διαμορφωνόταν η φόρμα με βάση το τί θέλει ο χρήστης να κάνει. Χρήσιμη θα ήταν η προβολή των πληροφοριών με ένα πιο ενδιαφέρον τρόπο, χωρίς τα περιττά κενά πεδία. Σε αντιδιαστολή με τις συνήθειες του χρήστη, αναδείχτηκαν επίσης και ενέργειες οι οποίες δεν συμβαίνουν σχεδόν ποτέ. Οι ενέργειες αυτές μπορούν να απομακρυνθούν από πρώτο πλάνο.

Όσον αφορά το περιβάλλον της φόρμας “Είδη” φαίνεται πως τα πεδία της είναι αρκετά καλά ομαδοποιημένα με τα πιο πολύ-επεξεργασμένα πεδία να είναι γειτονικά. Παράλληλα παρατηρείται πως ο τρόπος με τον οποίο γίνεται η συμπλήρωση των στοιχείων επηρεάζει αρκετά τον χρόνο ολοκλήρωσης μιας ενέργειας. Σε περιπτώσεις όπου η τιμή του πεδίου είναι τύπου λίστας ο χρόνος ολοκλήρωσης της είναι αρκετά μικρότερος από των πεδίων τύπου “κείμενο”. Μειώνοντας τον χρόνο διεκπεραίωσης κάθε ενέργειας συνεπάγεται και η συνολική ελαχιστοποίηση του χρόνου για ολόκληρη την εγγραφή πράγμα που πρέπει να γίνεται αυτοσκοπός των προγραμματιστών.

Τέλος, παρατηρήθηκε πως κάποιοι χρήστες υποπίπτουν πιο εύκολα σε λάθη από κάποιους άλλους, διαχωρίζοντας κάποιους χρήστες πιο αρχάριους στη χρήση του συστήματος. Έτσι, δημιουργείται η ανάγκη για εξοικείωση με το σύστημα. Αυτό μπορεί να συμβεί με κάποιου είδους εκπαίδευση του χρήστη από το ίδιο το σύστημα ή και με την συγγραφή αναλυτικών οδηγιών.

## **9.2 Μελλοντικές Επεκτάσεις**

Αποτέλεσμα αυτής της συνεχούς παρακολούθησης εξέλιξης ενός φαινομένου είναι η σε βάθος κατανόηση του. Γίνεται αντιληπτό ότι αυτός ο βαθμός κατανόησης της εξέλιξης ενός φαινομένου δίνει τη δυνατότητα να ληφθούν περισσότερο τεκμηριωμένες αποφάσεις. Ανάλογα με την οπτική και την σημασία των αποτελεσμάτων για τον καθένα, αυτά τα στατιστικά στοιχεία μπορούν να εξηγήσουν αντικειμενικά και χωρίς υποθέσεις την συμπεριφορά των χρηστών και κατ’ επέκταση να χρησιμοποιηθούν για την μελλοντική

βελτιστοποίηση του ήδη υπάρχοντος συστήματος έχοντας κατανοήσει πλήρως την πληροφορία.

Το σύστημα που αναπτύχθηκε στα πλαίσια διπλωματικής εργασίας δεν είναι μία πλήρης αξιολόγηση του συγκεκριμένου ERP συστήματος καθώς έγινε στα πλαίσια μίας συγκεκριμένης φόρμας της διεπαφής του. Ως μελλοντική εργασία μπορεί η συγκεκριμένη εργασία να επεκταθεί στο σύνολο της διεπαφής. Επίσης, η καταγραφή της συμπεριφοράς των χρηστών έγινε για ένα μικρό χρονικό διάστημα το οποίο θα μπορούσε να επεκταθεί σε αρκετά μεγαλύτερο χρονικό διάστημα για τη λήψη όσο το δυνατόν περισσότερων δεδομένων χρήσης. Τέλος, τα αποτελέσματα της συγκεκριμένης στατιστικής ανάλυσης θα πρέπει να ενσωματωθούν σε μία ευρύτερη και πολύπλευρη αξιολόγηση ευχρηστίας του σύμφωνα με το πρότυπο ISO 9241.

## Πηγές – Βιβλιογραφία

- [1] National Documentation Centre | National Documentation Centre,  
Η σημασία των έξυπνων στατιστικών στη δημιουργία έξυπνων και καινοτόμων πόλεων  
Ιστότοπος: <https://www.ekt.gr/en/node/22626>
- [2] Κουτσαμπάσης, Π. (2015) Αξιολόγηση Διαδραστικών Συστημάτων με Επίκεντρο τον χρήστη.  
Δράση Κάλλιπος - Ελληνικά Ακαδημαϊκά Ηλεκτρονικά Συγγράμματα
- [3] Κουτσαμπάσης, Π. (2011) Αλληλεπίδραση Ανθρώπου-Υπολογιστή. Κλειδάριθμος
- [4] Software Quality information, news and tips: What is HCI (human-computer interaction)?,  
Ιστότοπος:<https://searchsoftwarequality.techtarget.com/definition/HCI-human-computer-interaction>
- [6] University of Birmingham - A leading global university, Introduction to HCI,  
Ιστότοπος:<https://www.cs.bham.ac.uk/~rx/Teaching/HCI%20II/intro.html>
- [7] Dix, Alan J, 2003 Human-Computer Interaction Handbook
- [8] Interaction Design Foundation: What is Human-Computer Interaction (HCI)?  
Ιστότοπος:<https://www.interaction-design.org/literature/topics/human-computer-interaction>
- [9] The Encyclopedia of Human-Computer Interaction, 2nd Ed, by JOHN M. CARROLL  
Ιστότοπος:<https://www.interaction-design.org/literature/book/the-encyclopedia-of-humancomputer-interaction-2nd-ed/human-computer-interaction-brief-intro>
- [10] E.R. Peterson et al. / Personality and Individual Differences 38 (2005) 1269–1281  
A new measure of Verbal–Imagery Cognitive Style: VICS
- [11] Interaction with Mode and Structure of Presentation  
Ιστότοπος:<http://www.elsinnet.org.uk/abstracts/aom/sad-aom.htm>
- [12] The Encyclopedia of Human-Computer Interaction, 2nd Ed, by JOHN M. CARROLL,  
Context of Use,  
Ιστότοπος: <https://www.interaction-design.org/literature/topics/context-of-use>
- [13] Διεπαφή χρήστη-υπολογιστή : μια σύγχρονη προσέγγιση / Δημοσθένης Ακουμιανάκης  
Αθήνα : Κλειδάριθμος , c2006
- [14] Ιστότοπος: [https://www.123rf.com/photo\\_32778766\\_copy-machine-start-button-close-up.html](https://www.123rf.com/photo_32778766_copy-machine-start-button-close-up.html)
- [16] Padhy, R. P., Patra, M. R., & Satapathy, S. C. (2011). RDBMS to NoSQL: Reviewing Some Next-Generation Non-Relational Databases. International Journal of Advanced Engineering Science and Technologies, 15-20
- [17] E.A.Brewer, (2000, Jul) Towards robust distributed systems  
Ιστότοπος: <http://www.cs.berkeley.edu/~brewer/cs262b-2004/PODC-keynote.pdf>
- [18] MongoDB: relational-vs-non-relational-database,  
Ιστότοπος: <https://www.mongodb.com/scale/relational-vs-non-relational-database>.
- [19] Upwork. Wodehouse, Carey. sql vs nosql databases.

- Ιστότοπος: <https://www.upwork.com/hiring/data/sql-vs-nosql-databases-whats-the-difference>
- [20] ReadWrite: When NoSQL Databases Are — Yes — Good For You And Your Company  
 Ιστότοπος: <https://readwrite.com/2013/03/25/when-nosql-databases-are-good-for-you/>
- [21] Ιστότοπος: <https://www.linuxinsider.gr/forum/programmatismos/8873/eisagogi-stis-nosql-baseis-dedomenon-hrisi-toy-cassandra>
- [22] International Journal of Research in Advent Technology, Vol.6, No.2, February 2018  
 A Survey Paper on NoSQL Databases: Key-Value Data Stores and Document Stores  
<http://www.ijrat.org/downloads/Vol-6/feb-2018/paper%20ID-62201812.pdf>
- [23] Gaurav Vaish, 2013. Getting Started with NoSQL. Packt Publishing,  
 An Overview of NoSQL, Comparative Study of NoSQL Products, 67
- [24] SQL vs NoSQL Database Differences Explained with few Example DB  
 Ιστότοπος: <https://www.thegeekstuff.com/2014/01/sql-vs-nosql-db/>
- [25] SQL vs NoSQL  
 Ιστότοπος: <https://www.javatpoint.com/sql-vs-nosql>
- [27] Trigent Software: Understanding CouchDB, a Database That Embraces the Web,  
 Ιστότοπος: <https://blog.trigent.com/understanding-couchdb-a-database-that-embraces-the-web>
- [28] Apache CouchDB® 2.3 Documentation, Technical Overview  
 Ιστότοπος: <http://docs.couchdb.org/en/stable/intro/overview.html>
- [29] SharePoint and Office 365 Data Integration and File Sync Solutions,  
 How to connect CouchDB with SharePoint., A Step-by-step intro.  
 Ιστότοπος: <https://www.layer2solutions.com/how-to-connect/couchdb>
- [30] J. Chris Anderson 2010, CouchDB The Definitive Guide, The CAP Theorem  
 Ιστότοπος: <http://guide.couchdb.org/>
- [31] Apache CouchDB® Release 2.3.1, 2019  
 Ιστότοπος: <https://buildmedia.readthedocs.org/media/pdf/couchdb/latest/couchdb.pdf>
- [33] Simple Data Management for Analytics | Panoply: CouchDB vs MongoDB,  
 CouchDB vs MongoDB. Which Database Is Right For Your Business?  
 Ιστότοπος: <https://blog.panoply.io/couchdb-vs-mongodb>
- [34] Ionos, Working With CouchDB From the Command Line - 1&1 Hosting (US),  
 The Basics of Working With CouchDB  
 Ιστότοπος: <https://www.ionos.com/community/hosting/couchdb/working-with-couchdb-from-the-command-line/>
- [35] Gaurav Vaish 2013, Getting Started with NoSQ
- [36] Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον, ΟΡΓΑΝΙΣΜΟΣ ΕΚΔΟΣΕΩΣ ΔΙΔΑΚΤΙΚΩΝ ΒΙΒΛΙΩΝ - ΑΘΗΝΑ, 228
- [37] Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον, ΟΡΓΑΝΙΣΜΟΣ ΕΚΔΟΣΕΩΣ ΔΙΔΑΚΤΙΚΩΝ ΒΙΒΛΙΩΝ - ΑΘΗΝΑ, 228
- [38] Adam Fowler, 2015, NoSQL for Dummies - Getting started with NoSQL

- [39] How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu 18.04  
Ιστότοπος:<https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04>
- [40] ISO 9241-171:2008 Ergonomics of human-system interaction Guidance on software Accessibility
- [41] Extracting Usability Information from User Interface Events DAVID M. HILBERT AND DAVID F. REDMILES
- [42] The CAP-Theorem - Brewer, Towards Robust Distributed System, 2000
- [44] Laura Lemay, Πλήρες εγχειρίδιο της HTML4, 2008
- [45] Luke Welling, Laura Thomson - PHP and MySQL Web Development, Fifth Edition



## Παράρτημα

### A) Εγκατάσταση LAMP και CouchDB σε Ubuntu

#### A.1) Εισαγωγή και Προϋποθέσεις

##### Εισαγωγή

Μια στοίβα "LAMP" είναι μια ομάδα λογισμικού ανοιχτού κώδικα που συχνά εγκαθίστανται ταυτόχρονα με σκοπό να επιτρέψουν σε έναν διακομιστή να φιλοξενήσει δυναμικές ιστοσελίδες και εφαρμογές ιστού. Αυτός ο όρος είναι στην πραγματικότητα ένα ακρωνύμιο που αντιπροσωπεύει το λειτουργικό σύστημα Linux, με τον διακομιστή ιστού Apache. Η αποθήκευση των δεδομένων γίνεται σε μια MySQL βάση και η επεξεργασία του δυναμικού περιεχομένου γίνεται από την PHP. Σε συγκεκριμένη την περίπτωση η βάση δεδομένων όμως θα είναι η CouchDB κρίνοντας περιττή την εγκατάσταση της MySQL.

Σε αυτόν τον οδηγό, θα έχουμε μια στοίβα LAMP εγκατεστημένη σε ένα Ubuntu 16.04 Droplet. Το Ubuntu θα εκπληρώσει την πρώτη προϋπόθεση: ένα λειτουργικό σύστημα Linux.

##### Προϋποθέσεις

Πριν την εκκίνηση αυτού του οδηγού, θα πρέπει να υπάρχει ένας διαφορετικός λογαριασμός χρήστη πέραν του root με τα δικαιώματα sudo που έχουν οριστεί στον διακομιστή σας. Για να πραγματοποιηθεί αυτό θα πρέπει να γίνουν τα βήματα 1-4 στην αρχική εγκατάσταση του διακομιστή για το Ubuntu 16.04.

#### A.2) Εγκατάσταση του Apache

Ο διακομιστής ιστού Apache είναι ένας από τους πιο δημοφιλείς διακομιστές ιστού στον κόσμο. Είναι καλά τεκμηριωμένο και έχει χρησιμοποιηθεί ευρέως για μεγάλο μέρος της ιστορίας του ιστού, γεγονός που την καθιστά μια μεγάλη προεπιλεγμένη επιλογή για τη φιλοξενία ενός ιστοτόπου.

Η Εγκατάσταση του Apache μπορεί να γίνει εύκολα με την χρήση του διαχειριστή πακέτων του Ubuntu, απτ. πληκτρολογώντας τις εξής εντολές:

```
$ sudo apt-get update
$ sudo apt-get install apache2
```

Do you want to continue? [Y/n]

Πατώντας Y και στη συνέχεια Enter και η εγκατάσταση θα συνεχιστεί

Ορίζοντας την επιλογή Global ServerName θα καταργηθούν όλες οι προειδοποιήσεις σύνταξης

Στη συνέχεια, προσθέτοντας μια μόνο γραμμή στο αρχείο `/etc/apache2/apache2.conf` θα καταργηθεί ένα προειδοποιητικό μήνυμα. Ενώ είναι αβλαβές, εάν δεν οριστεί ServerName σε παγκόσμιο επίπεδο, θα εμφανιστεί η ακόλουθη προειδοποίηση κατά τον έλεγχο της διαμόρφωσης Apache για σφάλματα σύνταξης:

```
$ sudo apache2ctl configtest
```

Output

```
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using
Syntax OK
```

Το κύριο αρχείο ρυθμίσεων με την επεξεργασία κειμένου θα ανοίξει με της εξής εντολή:

```
$ sudo nano /etc/apache2/apache2.conf
```

Στο εσωτερικό, στο κάτω μέρος του αρχείου, η προσθήκη ServerName οδηγεί στο κύριο όνομα τομέα σας. Εάν δεν υπάρχει όνομα τομέα συνδεδεμένο με τον διακομιστή, μπορείτε να γίνει χρήση της δημόσιας διεύθυνση IP του διακομιστή:

```
/etc/apache2/apache2.conf
```

```
. . .
ServerName server_domain_or_IP
```

Ακολουθεί αποθήκευση και κλείσιμο του αρχείου.

Στη συνέχεια, γίνεται έλεγχος σφάλματων σύνταξης με την εντολή:

```
$ sudo apache2ctl configtest
```

Δεδομένου ότι προστέθηκε το global ServerName, θα πρέπει να δείτε όλα τα εξής:

Output

```
Syntax OK
```

Για να εφαρμοστούν οι αλλαγές θα πρέπει να γίνει επανεκκίνηση του Apache:

```
$ sudo systemctl restart apache2
```

Τώρα μπορεί να ξεκινήσει η προσαρμογή του τείχους προστασίας.

**Ρυθμίστε το τείχος προστασίας ώστε να επιτρέπεται η κυκλοφορία στον ιστό.**

Στη συνέχεια, υποθέτοντας ότι έχουν γίνει σωστά οι οδηγίες εγκατάστασης του αρχικού διακομιστή για να ενεργοποιηθεί το τείχος προστασίας UFW, θα πρέπει να επιτρέπεται η επισκεψιμότητα HTTP και HTTPS. Επιβεβαιώνοντας ότι το UFW έχει ένα προφίλ εφαρμογής για το Apache όπως αυτό:

```
$ sudo ufw app list
```

Output

Available applications:

```
Apache
Apache Full
Apache Secure
OpenSSH
```

Παρατηρώντας το πλήρες προφίλ Apache, θα πρέπει να φαίνεται ότι επιτρέπεται η κυκλοφορία στις θύρες 80 και 443:

```
$ sudo ufw app info "Apache Full"
```

Output

Profile: Apache Full

Title: Web Server (HTTP,HTTPS)

Description: Apache v2 is the next generation of the omnipresent Apache web server.

Ports:

```
80,443/tcp
```

Να επιτρέπεται η εισερχόμενη επισκεψιμότητα για αυτό το προφίλ:

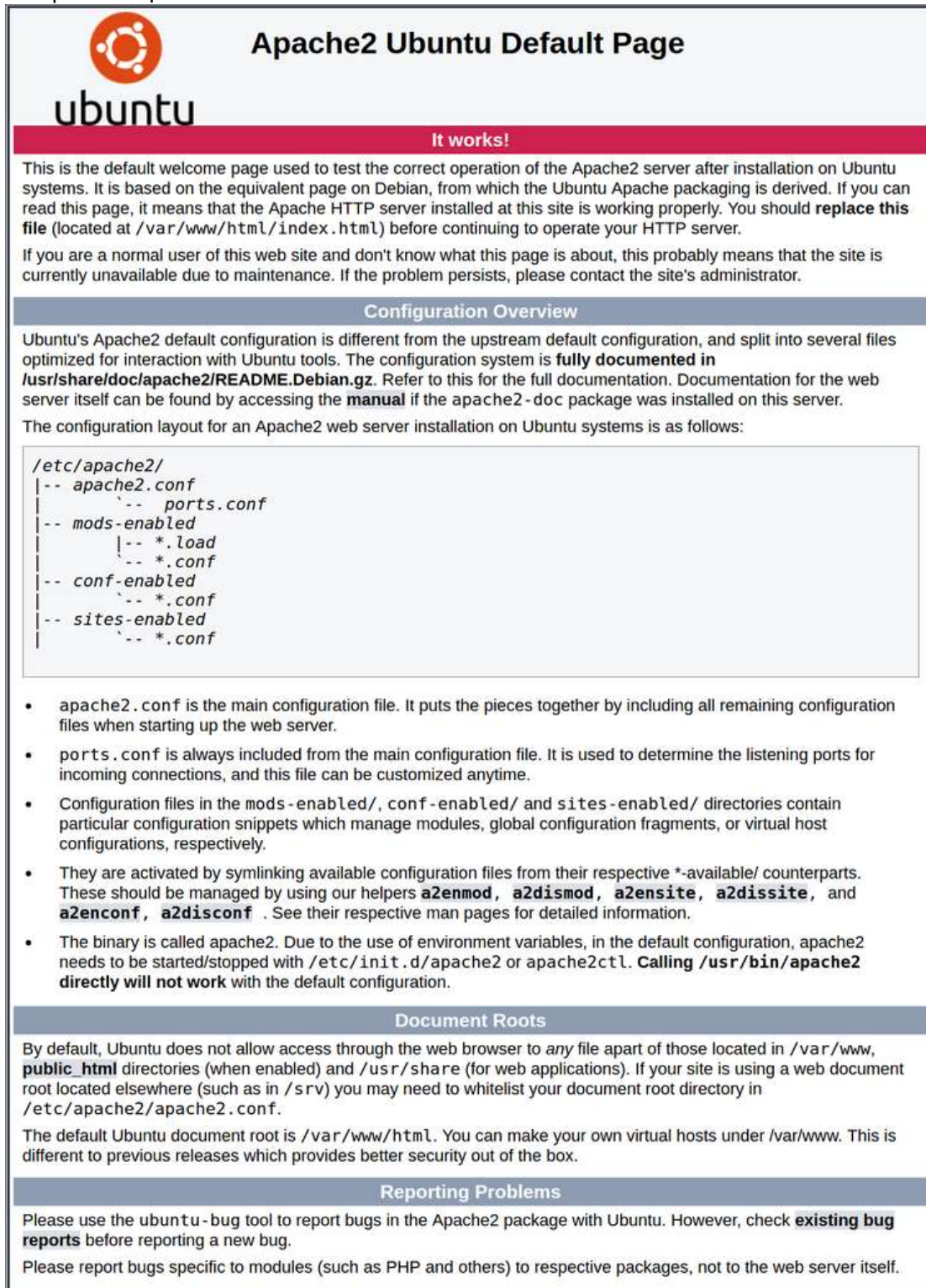
```
$ sudo ufw allow in "Apache Full"
```


Για να γίνει άμεσος έλεγχος επιβεβαιώνοντας ότι όλα πήγαν όπως προβλεπόταν με την επίσκεψη της δημόσιας διεύθυνσης IP του διακομιστή στο πρόγραμμα περιήγησης ιστού.

```
http://your_server_IP_address
```

Η προεπιλεγμένη ιστοσελίδα Ubuntu 16.04 Apache, η οποία υπάρχει για λόγους πληροφόρησης και δοκιμής, θα πρέπει να φαίνεται κάτι τέτοιο:

Θα πρέπει να φαίνεται κάτι τέτοιο:



 **Apache2 Ubuntu Default Page**

**It works!**

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

### Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

### Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, `public_html` directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

### Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

Εφόσον αυτή η ιστοσελίδα είναι ορατή, ο διακομιστής ιστού είναι πλέον σωστά εγκατεστημένος και προσβάσιμος μέσω του τείχους προστασίας.

### A.3) Εγκατάσταση PHP

#### Βήμα 1

Η PHP είναι το στοιχείο της ρύθμισης για την επεξεργασία κώδικα και την δημιουργία δυναμικού περιεχομένου ιστού. Εκτελεί αλυσίδες ενεργειών, συνδέετε με τις βάσεις δεδομένων για να αντλήσει πληροφορίες, τις επεξεργάζεται και να παραδίδει το δυναμικό περιεχόμενο στον διακομιστή ιστού μας.

```
$ sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql
```

Έτσι θα έχει πραγματοποιηθεί η εγκατάσταση της PHP χωρίς κανένα πρόβλημα.

Ο τρόπος που ο Apache εξυπηρετεί αρχεία κατά την αναζήτηση ενός καταλόγου μπορεί να αλλάξει. Επί του παρόντος, αν ο χρήστης αναζητήσει έναν κατάλογο από το διακομιστή, το Apache θα αναζητήσει πρώτα ένα αρχείο που ονομάζεται index.html. Αν ο χρήστης επιθυμεί, μπορεί να δηλώσει στον διακομιστή τα αρχεία PHP ως αρχεία της προτίμησης του ώστε ο Apache να αναζητήσει πρώτα ένα αρχείο με κατάληξη index.php

Αυτό μπορεί να συμβεί πληκτρολογώντας την παρακάτω εντολή για να ανοίξει το αρχείο dir.conf σε ένα πρόγραμμα επεξεργασίας κειμένου με δικαιώματα root:

```
$ sudo nano /etc/apache2/mods-enabled/dir.conf
```

Μοιάζει κάπως έτσι:

```
/etc/apache2/mods-enabled/dir.conf

<IfModule mod_dir.c>
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>
```

Τοποθέτηση του αρχείου index.php στην πρώτη θέση:

```
/etc/apache2/mods-enabled/dir.conf

<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>
```

Τέλος, ακολουθεί αποθήκευση και κλείσιμο του αρχείου με τον συνδυασμό πλήκτρων **Ctrl-X**. Η επιβεβαίωση της αποθήκευσης γίνεται πληκτρολογώντας το **Y** και πατώντας το πλήκτρο Enter για να επιβεβαιωθεί η τοποθεσία αποθήκευσης αρχείων.

Για να ισχύσουν οι αλλαγές πρέπει να γίνει επανεκκίνηση του διακομιστή Apache, με την εξής εντολή:

```
$ sudo systemctl restart apache2
```

Επίσης γίνεται έλεγχος της κατάστασης της υπηρεσίας apache2 χρησιμοποιώντας το systemctl

```
$ sudo systemctl status apache2
```

[39]

### Βήμα 3: Δοκιμή επεξεργασίας PHP στο διακομιστή σας Web

Για τον έλεγχο της διαμόρφωσης του συστήματος δημιουργείται το αρχείο info.php. και αποθηκεύεται στον κατάλογο "root web".

Στο Ubuntu 16.04, ο κατάλογος αυτός βρίσκεται στο / var / www / html /.

```
$ sudo nano /var/www/html/info.php
```

Το αρχείο θα είναι κενό και συγγράφεται ο παρακάτω έγκυρος κώδικα PHP:

```
info.php

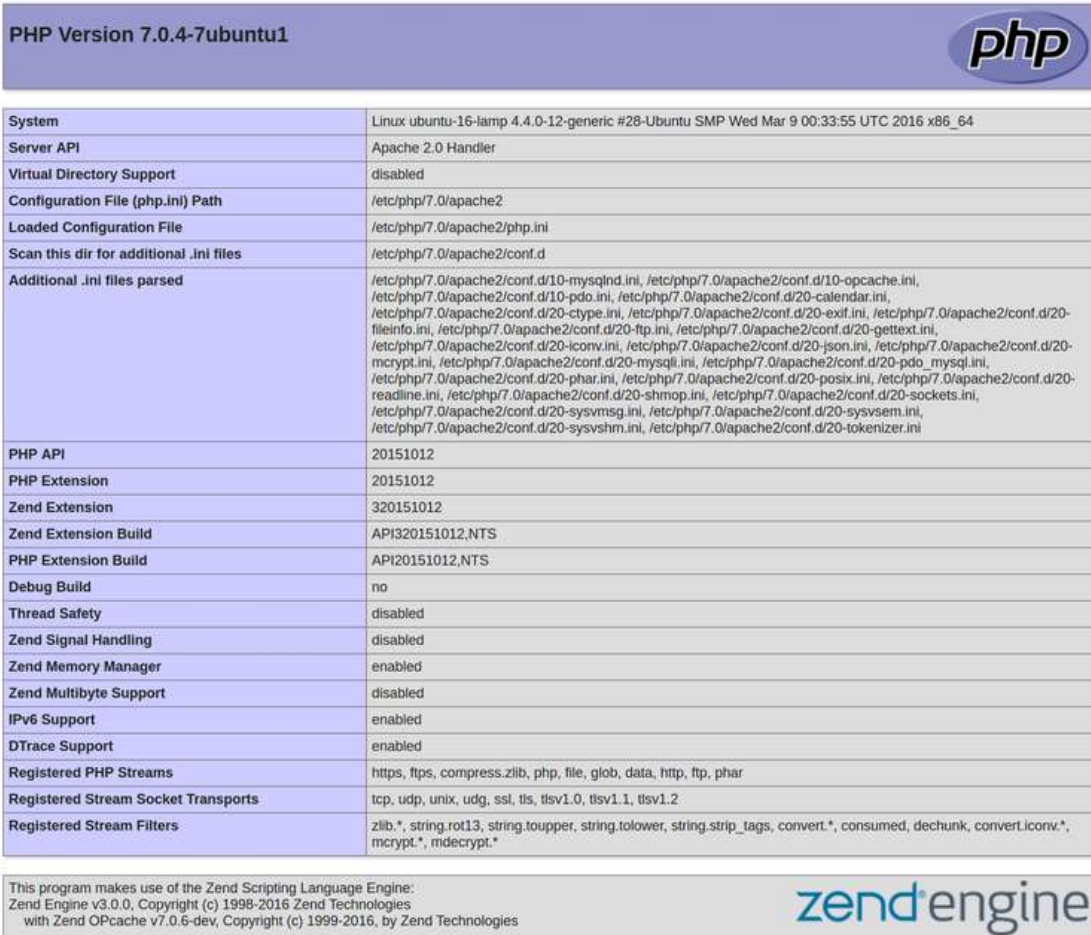
<?php
phpinfo();
?>
```

Ακολουθεί αποθήκευση και κλείσιμο του αρχείου.

Για την επαλήθευση αρκεί η επίσκεψη σε αυτή τη σελίδα στο πρόγραμμα περιήγησης ιστού

```
http://your_server_IP_address/info.php
```

Η σελίδα που γίνεται η μετάβαση θα πρέπει να μοιάζει με αυτήν:



The screenshot shows the output of the phpinfo() function. At the top, it displays "PHP Version 7.0.4-7ubuntu1" and the PHP logo. Below this is a table with various system and configuration details:

System	Linux ubuntu-16-lamp 4.4.0-12-generic #28-Ubuntu SMP Wed Mar 9 00:33:55 UTC 2016 x86_64
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-mcrypt.ini, /etc/php/7.0/apache2/conf.d/20-mysqli.ini, /etc/php/7.0/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.0/apache2/conf.d/20-sysvsem.ini, /etc/php/7.0/apache2/conf.d/20-sysvshm.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012.NTS
PHP Extension Build	API20151012.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*, mcrypt.*, mdecrypt.*

At the bottom, there is a footer with the Zend Engine logo and copyright information: "This program makes use of the Zend Scripting Language Engine: Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2016, by Zend Technologies".

## A.4) Εγκατάσταση Couchdb

**Debian/Ubuntu:** Εκτέλεση της εντολής:

```
$ echo "deb https://apache.bintray.com/couchdb-deb {distribution} main" \  
| sudo tee -a /etc/apt/sources.list
```

και αντικατάσταση {**distribution**} με την κατάλληλη επιλογή για την έκδοση λειτουργικού συστήματος:

- Debian 8: **jessie**
- Debian 9: **stretch**
- Ubuntu 14.04: **trusty**
- Ubuntu 16.04: **xenial**
- Ubuntu 18.04: **bionic**

### Εγκατάσταση των πακέτων Apache CouchDB

**Debian/Ubuntu:**

```
$ curl -L https://couchdb.apache.org/repo/bintray-pubkey.asc \  
| sudo apt-key add -
```

Ενημέρωση της προσωρινής μνήμης του και εγκατάσταση του πακέτου:

```
$ sudo apt-get update && sudo apt-get install couchdb
```

### Καθορισμός ιδιοκτησίας και άδειας

Για να αλλάξει η ιδιοκτησία, θα χρειαστεί να εκτελεστεί η ακόλουθη εντολή.

```
$ sudo chown -R couchdb:couchdb /usr/bin/couchdb /etc/couchdb/usr/share/couchdb
```

Αφού γίνει, θα πρέπει τώρα να διορθωθεί η άδεια αυτών των αρχείων και καταλόγων εκτελώντας τα παρακάτω.

```
$ sudo chmod -R 0770 /usr/bin/couchdb/etc/couchdb/usr/share/couchdb
```

Επανεκκίνηση το CouchDB

```
$ sudo systemctl restart couchdb
```

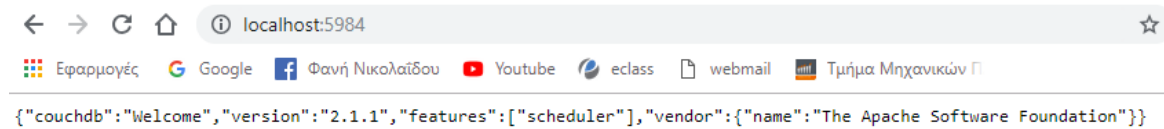
### Πρόσβαση στο Futon

```
$ ssh -L 5984: 127.0.0.1: 5984 root@ip-address
```

\*Εδώ θα χρειαστεί να αντικατασταθεί το root και το ip-address με το όνομα χρήστη και τη διεύθυνση ip του αντίστοιχου διακομιστή.

Τώρα, καθώς έχει ρυθμιστεί επιτυχώς η σήραγγα SSH, είναι δυνατή η πρόσβαση στην διεπαφή ιστού της CouchDB υποδεικνύοντας σε ένα πρόγραμμα περιήγησης ιστού το

**http: // localhost: 5984.**



Εξωτερική πρόσβαση

Αν χρειαστεί να γίνει το CouchDB προσιτό και διαθέσιμο εκτός του τοπικού δικτύου ή της τοπικής μηχανής, τότε θα πρέπει να έχουν ολοκληρωθεί τα παραπάνω βήματα για την εξασφάλιση του Fauxton.

Στη συνέχεια, θα πρέπει να προστεθεί το 0.0.0.0 στη διεύθυνση bind-address που μπορεί να τροποποιηθεί στο αρχείο /opt/couchdb/etc/local.ini κάτω από το [httpd] block. Για να γίνει αυτό, θα πρέπει να ακολουθήσει σύνδεση ως χρήστης root και να ανοιχτεί το αρχείο χρησιμοποιώντας ένα πρόγραμμα επεξεργασίας κειμένου.

### Αρχείο local.ini

```
nano /opt/couchdb/etc/local.ini
```

Τα παρακάτω δεν θα πρέπει να εμφανίζονται σαν σχόλια:

```
[chttpd]
```

```
port = 5984
```

```
bind_address = 0.0.0.0
```

Και στο τέλος του αρχείου:

```
[admins]
```

```
admin = mysecretpassword
```

Example: Fani = 123456

Τώρα, για να εφαρμοστούν οι αλλαγές, θα πρέπει να ξαναγίνει εκκίνηση των υπηρεσιών CouchDB χρησιμοποιώντας την εντολή systemctl.

```
$ sudo systemctl restart couchdb
```



## B) Δημιουργία requests στη Βάση Δεδομένων μέσω cURL

Αφού γίνει εγκατάσταση της CouchDB, η πρόσβαση API είναι δυνατή μέσω HTTP στη θύρα 5984, χρησιμοποιώντας τα τυπικά αιτήματα GET και PUT.

Από τη γραμμή εντολών χρησιμοποιώντας το Curl είναι δυνατή η έκδοση αυτών των αιτημάτων.

Η δοκιμή γίνεται με την εντολή:

```
curl http://127.0.0.1:5984/
```

Ο διακομιστής θα απαντήσει με ένα μήνυμα καλωσορίσματος:

```
{"couchdb":"Welcome","uuid":"f33e87d034bb8c1227f866445a977555","version":"1.6.1", "vendor":{"version":"16.04","name":"Ubuntu"}}
```

Χρησιμοποιώντας την `-X` σημαία καθορίζεται μια μέθοδος (όπως GET ή PUT) για το Curl. Για παράδειγμα, η εντολή για την εμφάνιση μιας λίστας με όλες τις βάσεις δεδομένων στην εγκατάσταση CouchDB είναι:

```
curl -X GET http://127.0.0.1:5984/\_all\_dbs
```

### Create

Χρησιμοποιώντας τη μέθοδο PUT δημιουργείται μια βάση δεδομένων ή ένα έγγραφο. Για παράδειγμα, για την δημιουργία μιας βάσης δεδομένων που ονομάζεται `reviews` η εντολή είναι:

```
curl -X PUT http://127.0.0.1:5984/reviews
```

Για την προσθήκη ενός έγγραφο στη βάση δεδομένων, ο χρήστης χρησιμοποιήσει `curl -X PUT` το όνομα της βάσης δεδομένων και το μοναδικό αναγνωριστικό της εγγραφής που θέλει να δημιουργήσει. Στη συνέχεια, χρησιμοποιεί το `-d` για να μεταφέρει τις πληροφορίες της εγγραφής σε μορφή JSON.

Για παράδειγμα, η ακόλουθη εντολή θα εισαγάγει μια εγγραφή με αναγνωριστικό 01 στη `reviews` βάση δεδομένων:

```
curl -X PUT http://127.0.0.1:5984/reviews/01 -d '{"reviewer_name":"Ben", "stars":"4", "details":"Love the calzone!"}'
```

## Read

Χρησιμοποιώντας τη μέθοδο GET είναι εφικτή η ανάκτηση και η ανάγνωση πληροφοριών σχετικά με μια βάση δεδομένων ή ένα έγγραφο. Για την απόκτησή ενός εγγράφου, πρέπει να καθοριστεί το αναγνωριστικό του εγγράφου στο τέλος της διεύθυνσης URL.

Για παράδειγμα, η εντολή για την ανάκτηση της εγγραφής που καταχωρίσαμε στο παραπάνω παράδειγμα είναι:

```
curl -X GET http://127.0.0.1:5984/reviews/01
```

Ο διακομιστής θα απαντήσει με τα περιεχόμενα του εγγράφου σε μορφή JSON:

```
{
  "_id": "01",
  "_rev": "1-8ce1d23b7455705c3c2cbbeb86d8ccf5",
  "reviewer_name": "Ben",
  "stars": "4",
  "details": "Love the calzone!"
}
```

## Update

Η CouchDB έχει αυτόματη παρακολούθηση και έλεγχο των αναθεωρήσεων. Στο παραπάνω παράδειγμα, υπάρχει ένα `_rev` πεδίο με αριθμό αναθεώρησης, παρόλο που δεν είχε εισαχθεί στο αρχικό μας αίτημα PUT. Αυτό το πεδίο αριθμού αναθεώρησης δημιουργείται αυτόματα κάθε φορά που εισάγετε μια εγγραφή.

Για να τροποποιηθεί ένα έγγραφο, θα πρέπει να ενημερωθεί η CouchDB για την αναθεώρηση που ενημερώνεται. Εάν συμφωνεί ο αριθμός αναθεώρησης, το CouchDB θα ενημερώσει τις πληροφορίες και θα δημιουργήσει έναν νέο αριθμό αναθεώρησης. (Εάν ο αριθμός αναθεώρησης δεν ταιριάζει, το CouchDB θα επιστρέψει ένα σφάλμα σύγκρουσης 409.)

Για να την ενημέρωση ενός εγγράφου, χρησιμοποιείται η μέθοδος PUT και συμπεριλαμβάνει τον αριθμό της αναθεώρησης. Μπορεί να ενημερωθεί ολόκληρο το έγγραφο ή μόνο ένα πεδίο.

Σε αυτό το παράδειγμα, ενημερώνεται ο αριθμός των αστεριών στην βαθμολογία από 4 σε 5:

```
curl -X PUT http://127.0.0.1:5984/reviews/01 -d '{"_id": "01", "stars": "5", "_rev": "1-8ce1d23b7455705c3c2cbbeb86d8ccf5"}'
```

Ο διακομιστής θα επιστρέψει μια απάντηση που περιλαμβάνει "ok" και τον νέο αριθμό αναθεώρησης:

```
{"ok": true, "id": "01", "rev": "2-55b1c6edb9a0fd8eb58bd73b8e0058d5"}
```

## Delete

Η διαγραφή μιας εγγραφής, γίνεται με τη χρήση της μεθόδου DELETE και συμπεριλαμβανομένου του αναγνωριστικού της και του πιο πρόσφατου αριθμού αναθεώρησης. Για παράδειγμα, η εντολή διαγραφής της εγγραφής που δημιουργήθηκε πιο παραπάνω είναι:

```
curl -X DELETE
http://127.0.0.1:5984/reviews/01?rev=255b1c6edb9a0fd8eb58bd73b8e0058d5
```

Ο διακομιστής θα επιστρέψει μια απάντηση που περιλαμβάνει "ok" και τον νέο αριθμό αναθεώρησης:

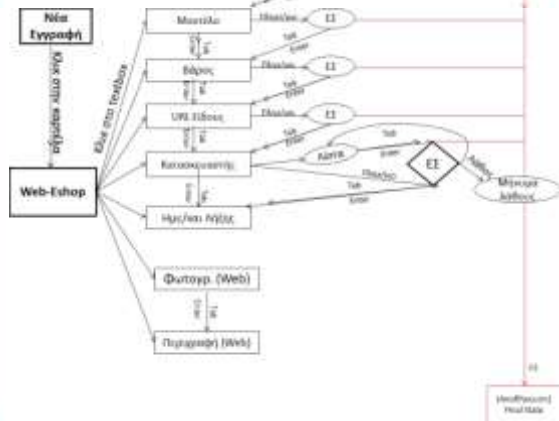
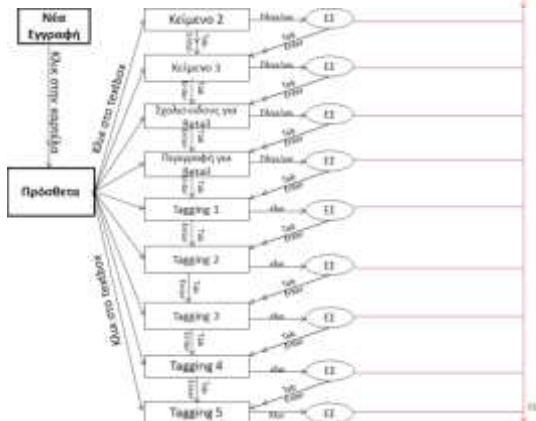
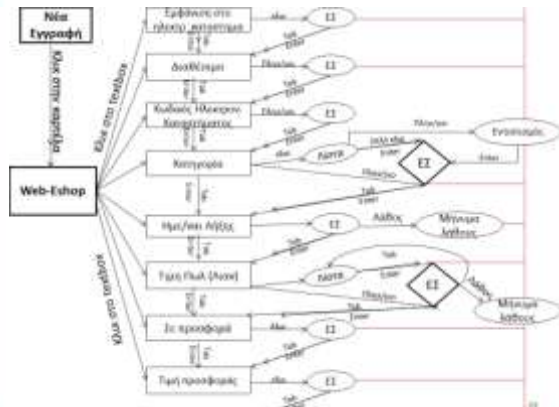
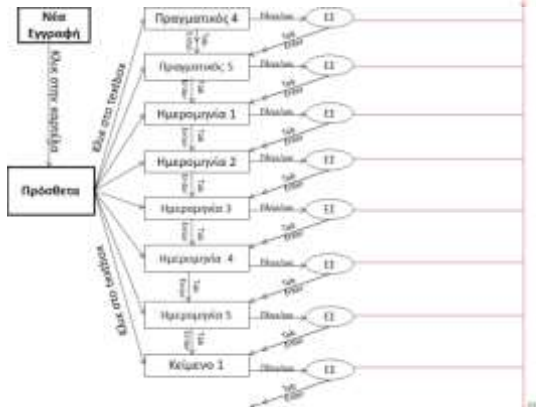
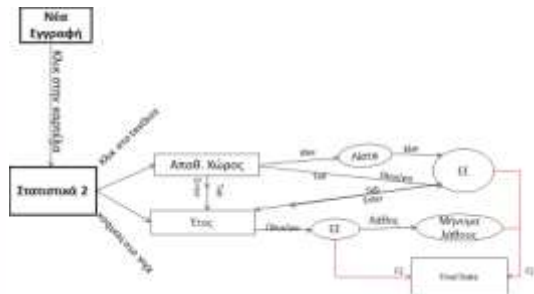
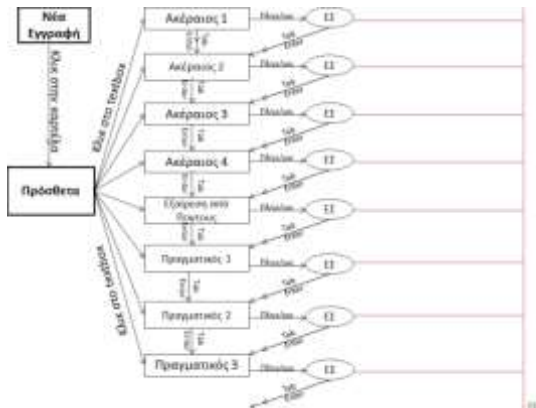
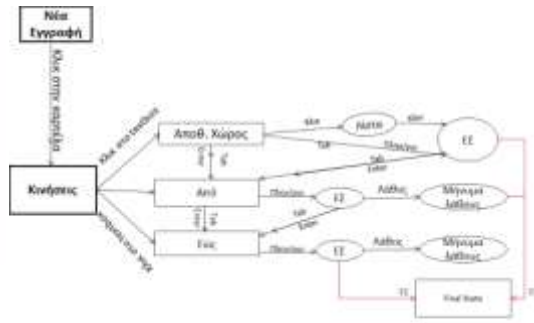
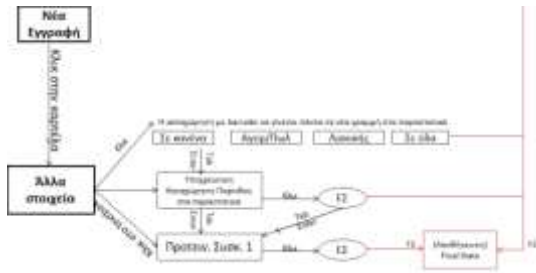
```
{"ok":true,"id":"01","rev":"3-ec3f3491444854d08aaa7dd6ffe68670"}
```

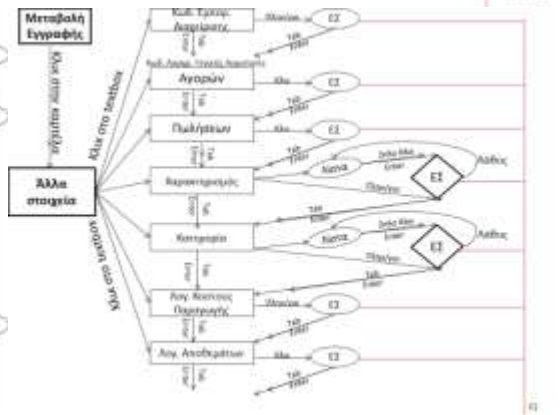
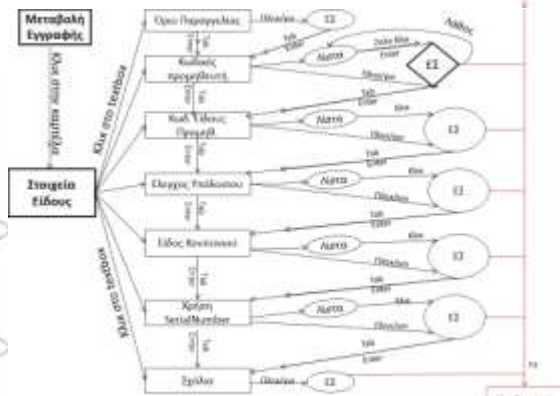
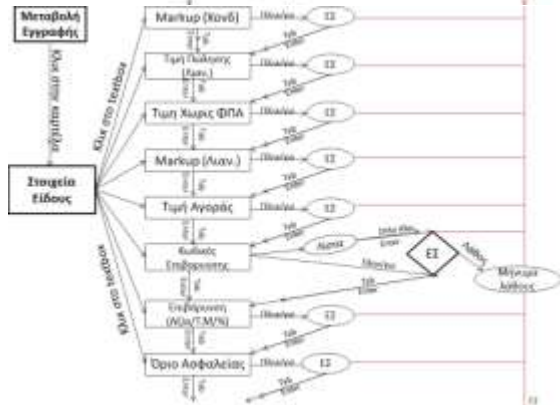
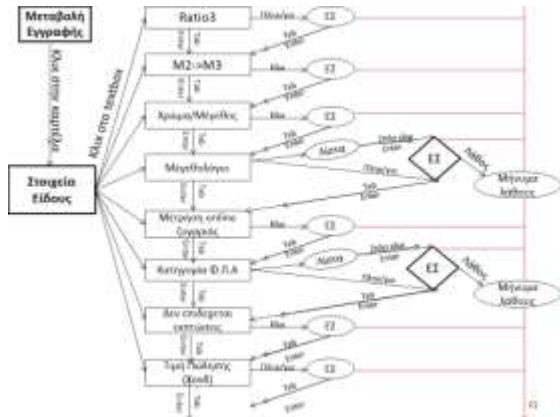
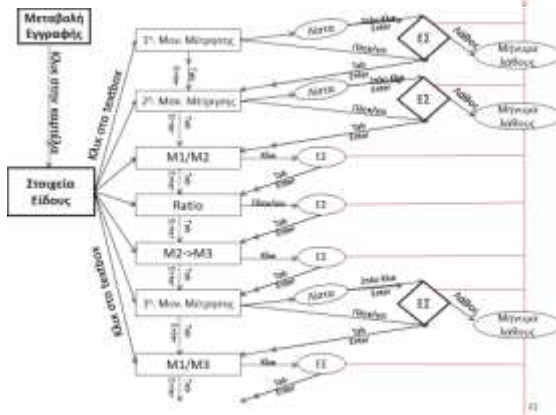
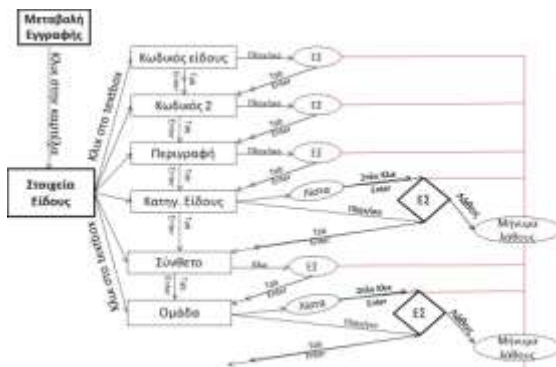
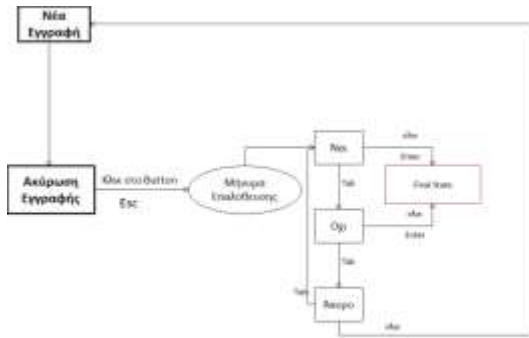
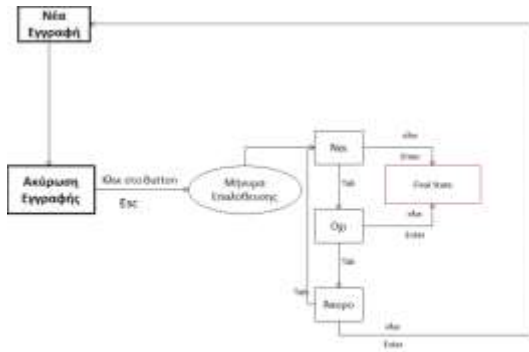
Αυτό συμβαίνει επειδή, για να διατηρηθούν οι λειτουργίες αναπαραγωγής του CouchDB, τα αρχεία δεν διαγράφονται πραγματικά. Αντ' αυτού, προστίθεται μια αναθεώρηση που περιλαμβάνει πληροφορίες σχετικά με τη διαγραφή και η εγγραφή δεν θα είναι προσβάσιμη με ένα τυποποιημένο αίτημα.

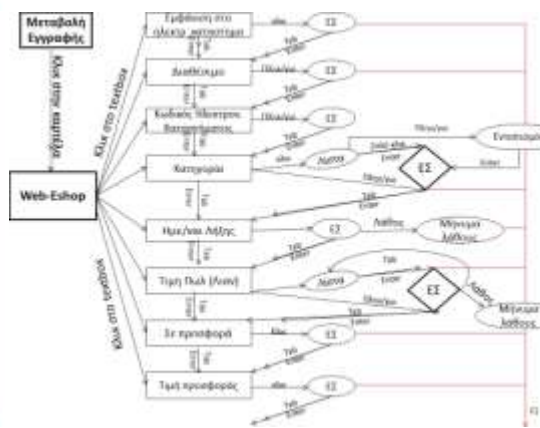
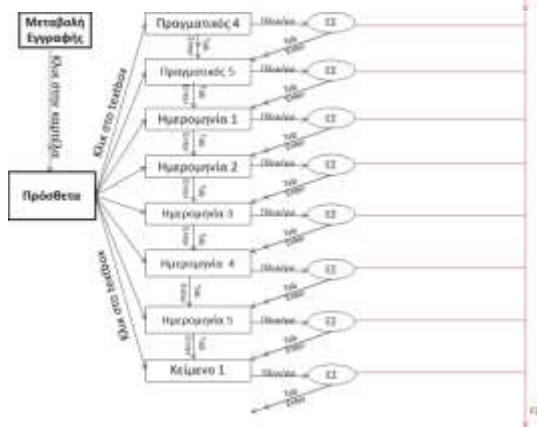
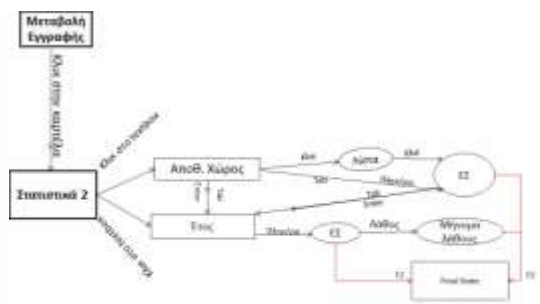
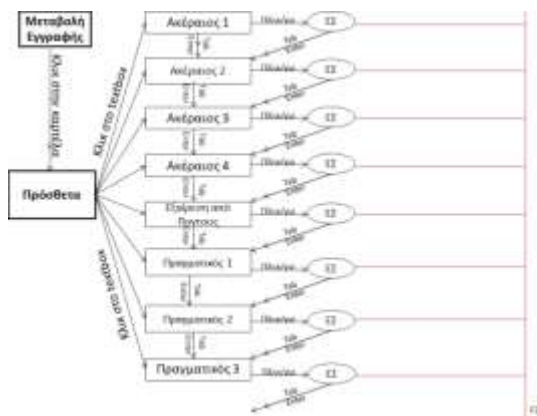
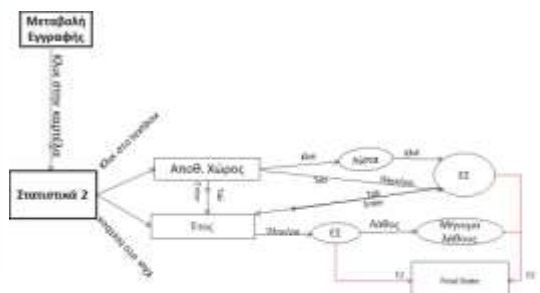
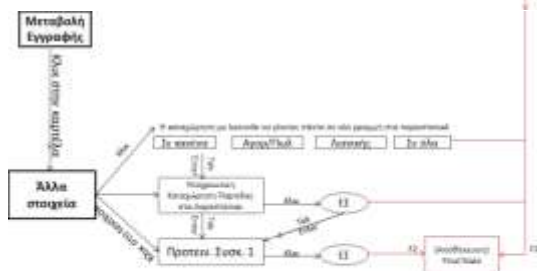
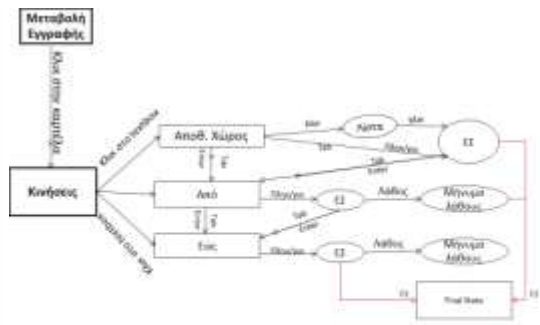
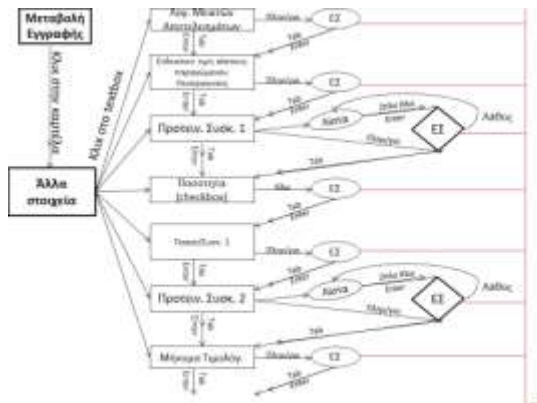
Για παράδειγμα, αν ο χρήστης προσπαθήσει να δει ξανά αυτή την εγγραφή, ο διακομιστής θα απαντήσει με "not found":

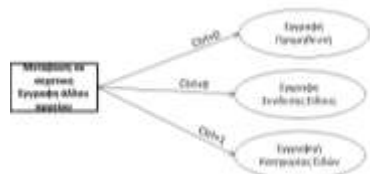
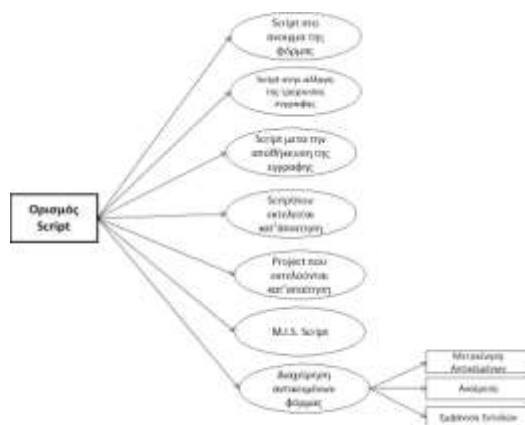
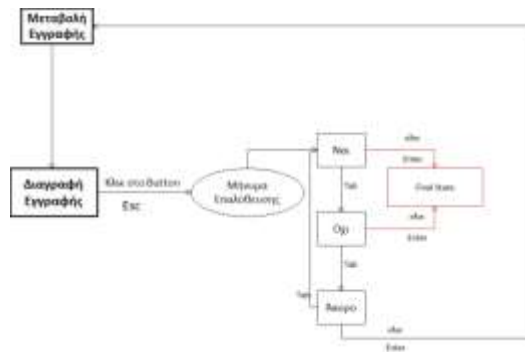
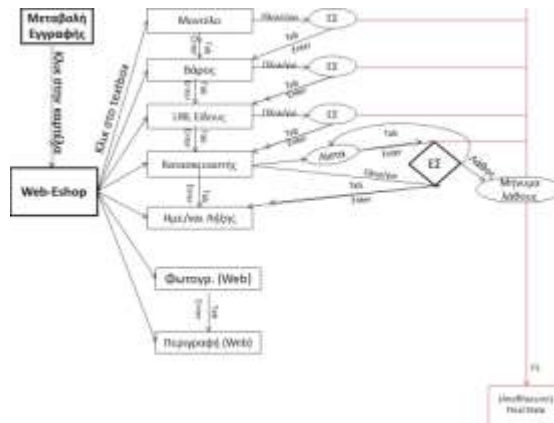
```
user@localhost:$ curl -X GET http://127.0.0.1:5984/reviews/01
{"error":"not_found","reason":"deleted"}
```













## Δ) Υλοποίηση Υποσυστήματος Καταγραφής Συμπεριφοράς (Delphi-Script)

```
uses
  Windows, Classes, Graphics, Controls, Forms, Dialogs, DB, ADODB,
  Grids, DBGrids, ExtCtrls, StdCtrls, DBTables, DateUtils;
var

  CouponsBtn : TAdvGlowButton;
  MoveBrandsBtn : TAdvGlowButton;
  ModifiersBtn : TAdvGlowButton;
  File1 : events;
  k : integer;
  p : integer;
  m : integer;
  g : integer;
  flag1 : integer;
  n : integer;
  v : integer;
  activity : TStringList;
  activity2 : TStringList;
  NameField : TStringList;
  NumField : TStringList;
  Index : integer;
  tabsheet : TStringList;
  sheetnumber : TStringList;
  button1 : AdvGlowButton;
  button2 : AdvGlowButton;
  button3 : AdvGlowButton;
  button4 : AdvGlowButton;
  button5 : AdvGlowButton;
  button6 : AdvGlowButton;
  button7 : AdvGlowButton;
  button8 : AdvGlowButton;
  button9 : AdvGlowButton;
  button10 : AdvGlowButton;
  button11 : AdvGlowButton;
  button12 : AdvGlowButton;
  button13 : AdvGlowButton;
  buttons : TStringList;
  numbers : TStringList;
  f5 : TStringList;
  str : string;
  buttonArr : Variant;
  find : TEdit;
  page : string;
  s : string;
  flag5, flag32, myflag : integer;
  box : TDBCheckbox;
  DataEntry : string;
  boxes : TStringList;
  checks : TStringList;
  flag50 : integer;
  start1, stop1, elapsed1, start2, stop2, elapsed2 : cardinal;
  timer, pedio : TStringList;
  mystring, name1, name2, timerstring, timer2 : string;
```

```

procedure BtnEnabled;
begin
if (ADataSet.FieldByName('COUPONKIND').Value = 2) then CouponsBtn.Enabled:=True
else CouponsBtn.Enabled:=False;
MoveBrandsBtn.Enabled:=True;
ModifiersBtn.Enabled:=True;
end;

```

```

Procedure SavingRecord (Sender: TObject; var Key: Word; Shift: TShiftState);
var
thttp : TIdHTTP;
DayOfWeek: string;

```

```

begin
stop2:=GetTickCount;
elapsed2:=stop2 - start2; //timer
elapsed2:=FloatToStr(round(elapsed2/1000));
timer2:=elapsed2 + ' seconds';

stop1:=GetTickCount;
elapsed1:=stop1 - start1; //timer2
elapsed1:=FloatToStr(round(elapsed1/1000));
timer.Add(elapsed1);

str:=str + 'Save';
DayOfWeek:="";
DT:=formatdatetime('dd.mm.yy hh:nn:ss', now);
Day:=formatdatetime('dddd', now);
if not (AnsiPos('day',Day)) then
begin
if (Day='Κυριακή') then Day:=('Sunday');
if (Day='Δευτέρα') then Day:=('Monday');
if (Day='Τρίτη') then Day:=('Tuesday');
if (Day='Τετάρτη') then Day:=('Wednesday');
if (Day='Πέμπτη') then Day:=('Thursday');
if (Day='Παρασκευή') then Day:=('Friday');
if (Day='Σάββατο') then Day:=('Saturday');
end

mystring:=mystring + 'Final State';
executeprocess('curl -X PUT http://195.201.222.232:5984/ux_tracker/' +
stringreplace(DT,' ','%20',setof(['rfReplaceAll'])) +
'-d ' +
""{
"UserName\": \"\" + UserName + "\",
"StationName\": \"\" + StationName + "\",
"DateTime\": \"\" + DT + "\",
"DayofWeek\": \"\" + Day + "\",
"FormName\": \"\" + Self.Name + "\",
"Tracker\": \"\" + mystring + "\",
"Tracker2\": \"\" + str + "\",
"Timer\": \"\" + timerstring + 'End' + "\",
"Timer2\": \"\" + timer2 + \"
}\"",false);
start2:=GetTickCount;
timerstring:='Start & ';
mystring:='Start & ';
str:="";
myflag:=0;

```

```

    flag32:=0;
    name1:="";
    name2:="";
    mystring:="";
    checks.Clear;
    boxes.Clear;
end;

```

```

Procedure EnterClick(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y);
Begin

```

```

    if (name1 <> TComponent(Sender).Name) then //για να μην αναγραφονται τα συνεχομενα κλικ
στην ιδια τριπλετα
    begin
        flag32:=1;
        mystring:=mystring + '1: ';
        name1:=TComponent(Sender).Name;
    end

```

```

checkflag:=0;
if (TComponent(Sender) is tdbcheckbox) then
begin
    if ( flag32 = 0 ) then //για να δινει αριθμο/ονομα σε συνεχομενα κλικ πανω σε checkbox
begin
    mystring:=mystring + '1: ';
end
flag32:=0;
for p :=0 to NameField.Count -1 do
begin
    if TComponent(Sender).Name = NameField.Strings[p].Name then
mystring:=mystring + inttostr(p);
end
cb:=TdbCheckbox(Sender);
set:= cb.datafield;

```

```

boxes.Add(TComponent(Sender).Name);
if(checks.Count > 0 )then
begin
    g:=boxes.Count - 1;
    for v:=g downto 1 do
begin
    if(boxes.Strings[v-1] = TComponent(Sender).Name) then
begin
    if(strtoint(checks.Strings[v-1]) = '3') then
begin
mystring:=mystring + ':2 & ';
checks.Add('2');
end
else
begin
mystring:=mystring + ':3 & ';
checks.Add('3');
end
checkflag:=1; // ιδιο checkbox
break;
end
end
end
if(checkflag=0) then // καινουργιο checkbox

```

```

begin
  if (Adataset[set]= cb.ValueChecked) then
  begin
    mystring:=mystring + ':2 & ';
    checks.Add('2');
  end
  else
  begin
    mystring:=mystring + ':3 & ';
    checks.Add('3');
  end
  end
end
else
begin
  if (Adataset[set]= cb.ValueChecked) then
  begin
    mystring:=mystring + ':2 & ';
    checks.Add('2');
  end
  else
  begin
    mystring:=mystring + ':3 & ';
    checks.Add('3');
  end
  end
end
end

for p := 0 to NameField.Count -1 do
  begin
    if TComponent(Sender).Name = NameField.Strings[p].Name then activity2.Add(inttostr(p));
  end
  if(page <> PageControl.ActivePage.Name) then
  begin
    page := PageControl.ActivePage.Name;
    showmessage (page);
    str := str + page + ' & ';
  end
end;

Procedure exiting(Sender: TObject)
begin
  stop1:=GetTickCount;
  elapsed1:=stop1 - start1;
  elapsed1:=FloatToStr(round(elapsed1/1000));

  for p :=0 to NameField.Count -1 do
  begin
    if TComponent(Sender).Name = NameField.Strings[p].Name then
    timerstring:=timerstring + intostr(p);
    end
    timerstring:=timerstring + ':' + elapsed1 + ' & ';
  end

for p :=0 to NameField.Count -1 do
  begin
    if TComponent(Sender).Name = NameField.Strings[p].Name then
    begin
    if (TComponent(Sender) is TDBGrid )then mystring:=mystring + intostr(p) + ':0 & ';
    if ((TComponent(Sender) is TDBedit) or (TComponent(Sender) is TEdit) ) then

```

```

begin
  if (NameField.Strings[p].Text > '') then mystring:=mystring + inttostr(p) + ':1 & ' else
mystring:=mystring + inttostr(p) + ':0 & ';
  end
end
end
end;

```

Procedure entering(Sender: TObject)

```

begin
start1:=GetTickCount;
  if(name2 = TComponent(Sender).Name ) then mystring:=mystring + '1:'; //τα TEdit κανουν onexit
για να διαλεξεις απο την λιστα κ ξανα onenter στο ιδιο πεδιο
  name2:=TComponent(Sender).Name;

  if (myflag = 0 ) then //1η φορα σε Tabsheet
begin
  page:=PageControl.ActivePage.Name;
  myflag:=1;
end;

  if(page <> PageControl.ActivePage.Name) then //προεπιλεγμενα Fields
begin
  mystring:=mystring + '0:';
  page:=PageControl.ActivePage.Name;
  str:=str + page + ' & ';
  end
end

```

Procedure KeyUP(Sender: TObject; var Key: Word; Shift: TShiftState)

```

begin
  checkflag:=0;
  if (Key = 32 ) then
begin
  if (TComponent(Sender) is tdbcheckbox) then
begin
  mystring:=mystring + '5:';
  for p :=0 to NameField.Count -1 do
begin
  if TComponent(Sender).Name = NameField.Strings[p].Name then mystring:=mystring +
inttostr(p);
  end
  cb:=TdbCheckbox(Sender);
  set:= cb.datafield;
boxes.Add(TComponent(Sender).Name);
if(checks.Count > 0 )then
begin
  g:=boxes.Count - 1;
  for v :=g downto 1 do
begin
  if(boxes.Strings[v-1] = TComponent(Sender).Name) then
begin
  if(strtoint(checks.Strings[v-1]) = '3') then

```

```

begin
mystring:=mystring + ':2 & ';
checks.Add('2');
end
else
begin
mystring:=mystring + ':3 & ';
checks.Add('3');
end
    checkflag:=1;
    break;
end
end
if(checkflag=0) then
begin
if (Adataset[set]= cb.ValueChecked) then
begin
mystring:=mystring + ':2 & ';
checks.Add('2');
end
else
begin
mystring:=mystring + ':3 & ';
checks.Add('3');
end
end
end
else
begin
if (Adataset[set]= cb.ValueChecked) then
begin
mystring:=mystring + ':2 & ';
checks.Add('2');
end
else
begin
mystring:=mystring + ':3 & ';
checks.Add('3');
end
end
end
end
if (Key = 13) or (Key = 9) or (Key = 32 and TComponent(Sender) is tdbcheckbox) then
begin
    if(page <> PageControl.ActivePage.Name) then
begin
page := PageControl.ActivePage.Name;
str := str + page + ' & ';

end
if ((inttostr(Key) = 09) and (Shift = SetOf([]))) then
begin
mystring:=mystring + '3: ';

    if (TComponent(Sender) is tdbcheckbox) then
begin
for p :=0 to NameField.Count -1 do
begin

```

```

        if TComponent(Sender).Name = NameField.Strings[p].Name then mystring:=mystring +
inttostr(p);
        end
        cb:=TdbCheckbox(Sender);
        set:= cb.datafield;
        boxes.Add(TComponent(Sender).Name);
        if(checks.Count > 0 )then
        begin
            g:=boxes.Count - 1;
            for v :=g downto 1  do
            begin
                if(boxes.Strings[v-1] = TComponent(Sender).Name) then
                begin
                    if(strtoint(checks.Strings[v-1]) = '2') then
                    begin
                        mystring:=mystring + ':3 & ';
                        checks.Add('3');
                    end
                    else
                    begin
                        mystring:=mystring + ':2 & ';
                        checks.Add('2');
                    end
                    checkflag:=1;
                    break;
                end
            end
            if(checkflag=0) then
            begin
                if (Adataset[set]= cb.ValueChecked) then
                begin
                    mystring:=mystring + ':3 & ';
                    checks.Add('3');
                end
                else
                begin
                    mystring:=mystring + ':2 & ';
                    checks.Add('2');
                end
            end
        end
        else
        begin
            if (Adataset[set]= cb.ValueChecked) then
            begin
                mystring:=mystring + ':3 & ';
                checks.Add('3');
            end
            else
            begin
                mystring:=mystring + ':2 & ';
                checks.Add('2');
            end
        end
    end
end
end
else if (inttostr(Key) = 09) then
    begin
        mystring:=mystring + '4: ';
    end

```

```

        if (TComponent(Sender) is tdbcheckbox) then
begin
    for p :=0 to NameField.Count -1 do
begin
    if TComponent(Sender).Name = NameField.Strings[p].Name then mystring:=mystring +
inttostr(p);
end
    cb:=TdbCheckbox(Sender);
    set:= cb.datafield;
boxes.Add(TComponent(Sender).Name);
if(checks.Count > 0 )then
begin
    g:=boxes.Count - 1;
    for v :=g downto 1 do
begin
    if(boxes.Strings[v-1] = TComponent(Sender).Name) then
begin
    if(strtoint(checks.Strings[v-1]) = '3') then begin
mystring:=mystring + ':3 & ';
checks.Add('3');
end
else
begin
mystring:=mystring + ':2 & ';
checks.Add('2');
end
        checkflag:=1;
        break;
end
end
if(checkflag=0) then
begin
if (Adataset[set]= cb.ValueChecked) then
begin
mystring:=mystring + ':3 & ';
checks.Add('3');
end
else
begin
mystring:=mystring + ':2 & ';
checks.Add('2');
end
end
end
else
begin
if (Adataset[set]= cb.ValueChecked) then
begin
mystring:=mystring + ':3 & ';
checks.Add('3');
end
else
begin
mystring:=mystring + ':2 & ';
checks.Add('2');
end
end
end
end
end
end

```



```

        end
    else if (inttostr(Key) = 13) then
        begin
            mystring:=mystring + '2: ';

            if (TComponent(Sender) is tdbcheckbox) then
            begin
                for p :=0 to NameField.Count -1 do
                begin
                    if TComponent(Sender).Name = NameField.Strings[p].Name then mystring:=mystring +
                    intostr(p);
                    end
                    cb:=TdbCheckbox(Sender);
                    set:= cb.datafield;
                    boxes.Add(TComponent(Sender).Name);
                    if(checks.Count > 0 )then
                    begin
                        g:=boxes.Count - 1;
                        for v :=g downto 1 do
                        begin
                            if(boxes.Strings[v-1] = TComponent(Sender).Name) then
                            begin
                                if(strtoint(checks.Strings[v-1]) = '3') then
                                begin
                                    mystring:=mystring + ':3 & ';
                                    checks.Add('3');
                                end
                                else
                                begin
                                    mystring:=mystring + ':2 & ';
                                    checks.Add('2');
                                end
                                checkflag:=1;
                                break;
                            end
                            end
                            if(checkflag=0) then
                            begin
                                if (Adataset[set]= cb.ValueChecked) then
                                begin
                                    mystring:=mystring + ':3 & ';
                                    checks.Add('3');
                                end
                                else
                                begin
                                    mystring:=mystring + ':2 & ';
                                    checks.Add('2');
                                end
                            end
                        end
                    end
                else
                begin
                    if (Adataset[set]= cb.ValueChecked) then
                    begin
                        mystring:=mystring + ':3 & ';
                        checks.Add('3');
                    end
                    else
                    begin
                        mystring:=mystring + ':2 & ';
                    end
                end
            end
        end
    end

```

```

    mystring:=mystring + ':2 & ';
    checks.Add('2');
end
end
end
end
end
end;

```

Procedure NewRecordEventHandler;

```

begin
str := str + 'Insert' + ' & ';
name1:=EditCODE.Name;
mystring := mystring + '0: ';
if (myflag=1) then str := str + page + ' & ';
end;

```

Procedure openform;

```

begin
RunProject(4,0,ADataset.FieldByName('ID').Value);
end;

```

Procedure openmovebrandsform

```

begin
RunProject(5,0,ADataset.FieldByName('ID').Value);
end;

```

Procedure openmodifiersform

```

begin
RunProject(10,0,ADataset.FieldByName('ID').Value);
end;

```

Procedure myfunc1(Sender: TObject);

```

begin
if (TComponent(Sender).Name = BtnRecCanc.Name ) then
begin
start2:=GetTickCount;
timerstring:='Start & ';
mystring:='Start & ';
myflag:=0;
flag32:=0;
name1:="";
name2:="";
str:="";
checks.Clear;
boxes.Clear;
end
if (TComponent(Sender).Name = btnRecUpd.Name ) then mystring:=mystring + '0: ';
str := str +TComponent(Sender).Name + ' & ';
end

```

begin

```

start2:=GetTickCount;
timerstring:='Start & ';

```

```

flag32:=0; //checkbox name
name1:=""; //συνεχομενα κλικ
name2:=""; //TEdit
mystring:='Start & ';

```

```

pedio:= TStringList.create;
timer:= TStringList.create;
checks:= TStringList.create;
boxes:= TStringList.create;
str := "";
myflag:=0; //page
button1 := BtnRecIns;
button2 := BtnRecCanc;
button3 := BtnRecFind;
button4 := btnRecOk;
button5 := btnActions;
button6 := BtnRecDel;
button7 := btnScripts;
button8 := BtnRecFirst;
button9 := BtnRecPrev;
button10 := BtnRecNext;
button11 := BtnRecLast;
button12 := btnRecPrint;
button13 := btnRecUpd;

button1.OnClick := 'myfunc1';
button2.OnClick := 'myfunc1';
button3.OnClick := 'myfunc1';
button4.OnClick := 'myfunc1';
button5.OnClick := 'myfunc1';
button6.OnClick := 'myfunc1';
button7.OnClick := 'myfunc1';
button8.OnClick := 'myfunc1';
button9.OnClick := 'myfunc1';
button10.OnClick := 'myfunc1';
button11.OnClick := 'myfunc1';
button12.OnClick := 'myfunc1';
button13.OnClick := 'myfunc1';

f5 := TStringList.create;
numbers := TStringList.create;
buttons := TStringList.create;
sheetnumber := TStringList.create;
tabsheet := TStringList.create;
activity2 := TStringList.create;
activity := TStringList.create;
NameField := TStringList.create;
NumField := TStringList.create;

NameField.Add(EditCODE); //0 Κωδικος Ειδους
NameField.Add(EditCODE2); //1 Κωδικος2
NameField.Add(EditDESCR); //2 Περιγραφη
NameField.Add(PRODUCTCODE); //3 Κατηγ Ειδους
NameField.Add(EditCOMPOSITE); //4 Συνθετο
NameField.Add(BRANDGROUPCODE); //5 Ομάδα
NameField.Add(MEASURECODE); //6 1η MM
NameField.Add(MEASURE2CODE); //7 2η MM
NameField.Add(EditUSEMMRATIO); //8 M1/M2
NameField.Add(EditMMRATIO); //9 Αναλογια
NameField.Add(EditM2TOM1); //10 M2->M1
NameField.Add(MEASURE3CODE); //11 3η MM
NameField.Add(EditUSEMMRATIO3); //12 M1/M3
NameField.Add(EditMMRATIO3); //13 Αναλογία2

```

```

NameField.Add(EditM3TOM1); //14 M3->M1
NameField.Add(DBCheckBox1); //15 Χρώμα/Μεγεθος
NameField.Add(BRANDSIZESCODE); //16 Μεγεθολογιο
NameField.Add(DBCheckBox4); //17 μετρησηζυγαριας
NameField.Add(DBCheckBox5); //18 Δεν επιδεχεται εκπτωσεις
NameField.Add(EditAVALSALES); //19 Τιμη Πώλησης (Χονδ)
NameField.Add(EditAVALSALES2); //20 Τιμη Πώλησης (Λιαν)
NameField.Add(EditAValNoFPA); //21 Τιμη Χωρις ΦΠΑ
NameField.Add(EditLASTVALBUY); //22 Τιμη Αγορας
NameField.Add(EditGChargeAVal); //23 Επιβαρυνση (Αξια)
NameField.Add(EditQTYLIMIT1); //24 Οριο Ασφαλείας
NameField.Add(EditQTYLIMIT2); //25 Οριο Παραγγελίας
NameField.Add(SUPPLIERCODE); //26 Κωδ.Προμηθευτη
NameField.Add(EditCODESUPPL); //27 Κωδ.Ειδους Προμηθ.
NameField.Add(EditCOMMENTS); //28 Σχόλια
NameField.Add(DBCheckBox2); //28 Ενεργό

```

//Alla stoixeia

```

NameField.Add(EditCMCODE);
NameField.Add(EditGLPURCHASE);
NameField.Add(EditGLLOCAL);
NameField.Add(BRANDALTYPEPCODE);
NameField.Add(BRANDALCATEGCODE);
NameField.Add(EditGLPRODCOST);
NameField.Add(EditGLSTOCK);
NameField.Add(EditGLRESULTS);
NameField.Add(EditBYPRODVAL);
NameField.Add(PACK1CODE);
NameField.Add(DBCheckBox3);
NameField.Add(EditM1P1RATIO);
NameField.Add(PACK2CODE);
NameField.Add(EditMSGPARREC);
NameField.Add(EditLOTINDOCS);
NameField.Add(EditTAGLIST);

```

//Prostheta

```

NameField.Add(DBEditINT1);
NameField.Add(DBEditINT2);
NameField.Add(DBEditINT3);
NameField.Add(DBEditINT4);
NameField.Add(DBEditINT5);
NameField.Add(DBEditFLOAT1);
NameField.Add(DBEditFLOAT2);
NameField.Add(DBEditFLOAT3);
NameField.Add(DBEditFLOAT4);
NameField.Add(DBEditFLOAT5);
NameField.Add(DBEditDATE1);
NameField.Add(DBEditDATE2);
NameField.Add(DBEditDATE3);
NameField.Add(DBEditDATE4);
NameField.Add(DBEditDATE5);
NameField.Add(DBEditSTR1);
NameField.Add(DBEditSTR2);
NameField.Add(DBEditSTR3);
NameField.Add(DBEditSTR4);
NameField.Add(DBEditSTR5);

```

//Sysxetizomenoi

```

NameField.Add(EditBSFilter);

```

```
NameField.Add(DBGridBrandSelf);

//Kinhseis
NameField.Add(DBGrid);

//Statistika1
NameField.Add(DBGrid1);

//Statistika12
NameField.Add(DBGrid2);

//Fwto/keimeno
NameField.Add(EditInfo);

for k := 0 to NameField.Count -1 do
begin
  NumField.Add(inttostr(k));
  NameField.Strings[k].onmousedown := 'EnterClick';
  NameField.Strings[k].onkeyup := 'KeyUP';
  NameField.Strings[k].onexit := 'exiting';
  NameField.Strings[k].onenter := 'entering';
end;
page:="";
host.keypreview := True;
// self.onkeydown:= 'f12';
Host.OnNewRecord := 'NewRecordEventHandler';
Host.OnSavingRecord := 'SavingRecord';
```

## E) Υλοποίηση Υποσυστήματος Στατιστικής Ανάλυσης

Παρακάτω παρουσιάζονται τρία scripts που συνεργάζονται για την υλοποίηση του εξής ερωτήματος:

Σε τι ποσοστό παραμένει κάποιο πεδίο κενό;

Το αρχείο html δημιουργεί ιστοσελίδα για την εισαγωγή τιμής από τον χρήστη, σε αυτή την περίπτωση ζητείται να εισαχθεί ο αριθμός του πεδίου που για να αναλυθεί κατά ποσό παραμένει κενό.

### Αρχείο searchblank.html

```
<html>
<head></head>
<body>
<form method="get" action="searchblank1.php">
Give Number of Field</input></br>
<input type="number" name="pedio">
<input type="submit" value="search"/>
</form>
</body>
</html>
```

Το αρχείο blank.php συνδέει την βάση δεδομένων CouchDB μέσω curl για να μπορεί να έχει πρόσβαση στα δεδομένα της και τα επεξεργάζεται.

### Αρχείο blank.php

```
<?php
$ch = curl_init();

curl_setopt($ch, CURLOPT_URL,
'http://195.201.222.232:5984/ux_tracker/_design/n2/_view/allmine');
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'GET');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
    'Content-type: application/json',
    'Accept: */*'
));

curl_setopt($ch, CURLOPT_USERPWD, 'admin:faniflorina');
$response = curl_exec($ch);
$a=json_decode($response, true);

$pedio=":".$_GET['pedio'].":";

$q=0;
$q1=0;
foreach($a[rows] as $value){
    $b=explode(' ', $value[key][Tracker]);
    for($i=0;$i<count($b);$i++){
        if (strstr($b[$i], $pedio)){
            $q++;
            if(substr($b[$i], -1)=="0") $q1++;
        }
    }
}
```

```

    }
}
$pos= round((( $q1/($q1+$q))*100),2);
echo "This Field is Empty: $pos%";

```

```

curl_close($ch);
?>

```

Το αρχείο searchblank.php αφορά την βιβλιοθήκη Jcanvas που αναπαριστά τα ποσοστά με γράφημα.

### searchblank.php

```

<?php
    include 'blank.php';

    $dataPoints = array(
        array("label"=>"Empty", "y"=>(( $q1/($q1+$q))*100) ),
        array("label"=>"Edited", "y"=>(100-(( $q1/($q1+$q))*100)))
    )

    ?>
    <!DOCTYPE HTML>
    <html>
    <head>
    <script>
    window.onload = function() {

        var chart = new CanvasJS.Chart("chartContainer", {
            animationEnabled: true,
            title: {
                text: "Empty or Edited Field"
            },
            // subtitles: [{
            //     text: "November 2017"}],
            data: [{
                type: "pie",
                yValueFormatString: "#,##0.00\ \"%\\"",
                indexLabel: "{label} ({y})",
                dataPoints: <?php echo json_encode($dataPoints,
JSON_NUMERIC_CHECK); ?>
            }
        ]
    });
    chart.render();

    }
</script>
</head>
<body>
<div id="chartContainer" style="height: 370px; width: 100%;"></div>
<script
src="https://canvasjs.com/assets/script/canvasjs.min.js"></script>
</body>
</html>

```

Όλα τα υπόλοιπα ερωτήματα υλοποιήθηκαν κατά παρόμοιο τρόπο.