

ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΚΡΗΤΗΣ

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων



Πτυχιακή Εργασία

**«InfoGraberCreator - Σχεδιασμός και Υλοποίηση εργαλείου
λογισμικού αναζήτησης δεδομένων στο web.»**

Ζαχαριουδάκης Εμμανουήλ

Επιβλέπων καθηγητής : Δρ Βιδάκης Νικόλαος

Επιτροπή Αξιολόγησης :

Ημερομηνία παρουσίασης:

Ηράκλειο Μάιος 2011

Περίληψη

Σκοπός της πτυχιακής αυτής ήταν η μελέτη, σχεδίαση και ανάπτυξη εργαλείου για την σάρωση διαφόρων πηγών δεδομένων και την αποθήκευση των δεδομένων που εμπεριέχονται, με σκοπό την εξαγωγή χρήσιμων, για το χρήστη, συμπερασμάτων.

Για το σκοπό αυτό μελετήθηκαν τέτοια συστήματα τα οποία λειτουργούν σαρώνοντας διάφορες πηγές δεδομένων. Κανένα από τα εργαλεία που μελετήθηκαν δεν προσφέρει την σάρωση ετερογενών πηγών δεδομένων. Πιο συγκεκριμένα, μελετήθηκαν συστήματα (Grabbers) τα οποία λειτουργούν για αποθήκευση προϊόντων εμπορικών καταστημάτων ηλεκτρονικής μορφής. Τα συστήματα αυτά προσφέρουν, αναζήτηση και αποθήκευση δεδομένων από xml (και μόνο) πηγές.

Το σύστημα που δημιουργήθηκε έχει την ικανότητα να διαχειρίζεται δεδομένα, τα οποία βρίσκονται σε αρχεία excels, xml, csv καθώς επίσης και την δημιουργία δεδομένων σε online μορφή (Digital Sources) χωρίς την παρουσία αρχείων. Επίσης, το σύστημα παρέχει διαφορετικές υπηρεσίες ανά χρήστη, σε web based interface, με απλοποιημένες διαδικασίες, τόσο όσο, ένας μη έμπειρος χρήστης του internet να μπορεί να χειριστεί.

Η συλλογή δεδομένων, και η παρουσίαση αυτών, γίνεται με διαφορετικούς τρόπους ανάλογα τις ανάγκες που παρουσιάζονται στον εκάστοτε χρήστη της εφαρμογής, με αποτέλεσμα, να ικανοποιεί και να παρέχει την λύση που ο χρήστης επιθυμεί.

Το σύστημα αναπτύχθηκε με χρήση γλώσσας PHP, JAVASCRIPT καθώς χρησιμοποιήθηκαν ακόμη τεχνολογίες Ajax, JQuery. Η βάση δεδομένων είναι mysql, καθώς για το Interface χρησιμοποιήθηκε html και css.

Abstract

The aim of this thesis was to design and develop a tool to be used for scanning various data sources, as well as for saving these data in order to be easy for the user to work on.

Several systems, which are used for scanning data were studied for that purpose, none of which provides scanning of heterogeneous data sources.

To be more precise we studied systems (Grabbers) which are used for storing the products of online shops. Those systems offer, searching and saving data from only xml sources.

The tool that was created, has the ability to handle data which are located in excel, xml and csv files.

In addition, it can create data in an online form (Digital Sources) without the need of files.

Moreover, the system provides different services for each user, in a web based interface, through simplified procedures, so as a non experienced internet user can operate it.

The collection and presentation of the data, can be illustrated through different ways, depending on the needs of each user.

The system was developed with the use of PHP and Javascript as well as Ajax and JQuery technologies. The data base is mysql the interface was created with html and css.

Table of Contents

ΠΕΡΙΛΗΨΗ	3
ABSTRACT	4
ΕΙΣΑΓΩΓΗ	7
1.1.	ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΠΤΥΧΙΑΚΗΣ 7
1.2.	ΟΡΓΑΝΩΣΗ ΤΟΥ ΤΟΜΟΥ 7
2.	ΧΡΗΣΙΜΟΙ ΟΡΙΣΜΟΙ 8
2.1.	ΤΙ ΕΙΝΑΙ ΔΕΔΟΜΕΝΑ 8
2.2.	ΤΙ ΕΙΝΑΙ ΣΥΛΛΟΓΗ ΔΕΔΟΜΕΝΩΝ 8
2.3.	ΤΙ ΕΙΝΑΙ ΑΥΤΟΜΑΤΟΠΟΙΗΣΗ ΣΥΛΛΟΓΗΣ ΔΕΔΟΜΕΝΩΝ 8
2.4.	ΤΙ ΕΙΝΑΙ WEB BASED APPLICATION 8
2.5.	ΤΙ ΕΙΝΑΙ XML..... 8
2.6.	ΤΙ ΕΙΝΑΙ EXCEL..... 9
2.7.	ΤΙ ΕΙΝΑΙ CSV 10
2.8.	ΠΑΡΑΔΕΙΓΜΑΤΑ ΧΡΗΣΗΣ 11
3.	ΤΕΧΝΟΛΟΓΙΑ 12
3.1.	ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ..... 12
3.1.1.	<i>MySQL</i> 12
3.2.	ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ..... 13
3.2.1.	<i>HTML</i> 13
3.2.2.	<i>Css</i> 17
3.2.3.	<i>Javascript</i> 18
3.2.4.	<i>AJAX</i> 19
3.2.5.	<i>JQuery</i> 21
3.2.6.	<i>PHP</i> 21
3.3.	MODULES 22
3.3.1.	<i>Live Search</i> 22
3.3.2.	<i>DataTable module</i> 22
3.4.	ΙΔΕΕΣ ΥΛΟΠΟΙΗΣΗΣ 23
3.4.1.	<i>Create xml</i> 23
3.4.2.	<i>Database</i> 23
3.5.	ΤΕΧΝΟΛΟΓΙΑ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΕ 24
4.	ΑΝΑΛΥΣΗ ΛΟΓΙΣΜΙΚΟΥ 25
4.1.	ΕΙΣΑΓΩΓΗ..... 25
4.2.	ΚΑΤΑΓΡΑΦΗ ΑΠΑΙΤΗΣΕΩΝ..... 25
4.2.1.	<i>Use Cases</i> 25
4.2.2.	<i>Σενάρια χρήσης (Scenarios of Use)</i> 26
4.2.3.	<i>Data Base Model</i> 27
4.2.4.	<i>User Interface</i> 27
4.3.	ΤΗ INFO GRABBER APPLICATION CASE STUDY 28
4.3.1.	<i>Περιγραφή συστήματος Info Grabber Application</i> 28
4.3.2.	<i>Γενικό σενάριο χρήσης του Info Grabber Application</i> 28
4.3.3.	<i>Καταγραφή απαιτήσεων Info Grabber Application</i> 29
4.3.4.	<i>Περιπτώσεις χρήσης και σενάρια Info Grabber Application</i> 33
4.3.5.	<i>Data model Info Grabber Application</i> 53
5.	ΔΙΕΠΑΦΗ INFO GRABBER APPLICATION 56
6.	ΣΥΜΠΕΡΑΣΜΑΤΑ 67
7.	ΒΙΒΛΙΟΓΡΑΦΙΑ 68

Table of Figures

Εικόνα 1: Δομή XML	9
Εικόνα 2: Δομή Excel.....	10
Εικόνα 3:Δομή CSV	10
Εικόνα 4:Ajax Model.....	21
Εικόνα 5:Data Table Model.....	22
Εικόνα 6: Create XML.....	23
Εικόνα 7: Database	23
Εικόνα 8: Παράδειγμα χρήσης κατηγοριών	29
Εικόνα 9:Data Base Screenshot.....	53
Εικόνα 10: Register	56
Εικόνα 11:Login	57
Εικόνα 12:Login	57
Εικόνα 13:New Source	58
Εικόνα 14:View sources	59
Εικόνα 15:New Field	59
Εικόνα 16:Source Connections.....	60
Εικόνα 17:New Category.....	60
Εικόνα 18: Node View.....	61
Εικόνα 19: Node Edit.....	61
Εικόνα 20:Live Search Module	62
Εικόνα 21: Live Search Module - Results	62
Εικόνα 22:Nodes From One Source	64

Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζουμε μια γενική περιγραφή του αντικειμένου της πτυχιακής αυτής και την οργάνωση των κεφαλαίων του υπόλοιπου τόμου.

1.1. Αντικείμενο της πτυχιακής

Στην παρούσα πτυχιακή ασχοληθήκαμε με την δημιουργία μιας διαδικτυακής πλατφόρμας, αυτοματοποιημένης διαδικασίας για δημιουργία συλλογών δεδομένων. Η πλατφόρμα υποστηρίζει σάρωση ετερογενών πηγών δεδομένων, με φιλικές για τον χρήστη διαδικασίες και έξυπνες μεθόδους (smart digital logic) για αποθήκευση αυτών, με σκοπό την δημιουργία συλλογών δεδομένων. Οι συλλογές δεδομένων ομαδοποιούνται και χρησιμοποιούνται για την εξαγωγή χρήσιμων συμπερασμάτων, διαφορετικών ανά χρήστη.

1.2. Οργάνωση του τόμου

Η συνέχεια της πτυχιακής έχει οργανωθεί στα ακόλουθα κεφάλαια:

- Στο Κεφάλαιο 2 παρουσιάζονται χρήσιμες έννοιες και ορισμοί, σχετικά με τις απαραίτητες γνώσεις που πρέπει να κατέχει κάποιος, για την παρουσίαση της πτυχιακής.
- Στο Κεφάλαιο 3, παρουσιάζεται η τεχνολογία που χρησιμοποιήθηκε για την ανάπτυξη του συστήματος
- Στο Κεφάλαιο 4, παρουσιάζονται γενικές αλλά και ειδικές πληροφορίες για την ανάλυση, σχεδίαση και υλοποίηση του συστήματος
- Στο Κεφάλαιο 5, παρουσιάζονται τα συμπεράσματα της έρευνας και της υλοποίησης
- Στο Κεφάλαιο 6, παρουσιάζονται τα παραρτήματα στα οποία αναφερόμαστε σε διάφορα σημεία της πτυχιακής
- Στο Κεφάλαιο 7, παρουσιάζεται όλη η σχετική βιβλιογραφία στην οποία βασιστήκαμε για τη μελέτη και εγγραφή της πτυχιακής.

2. Χρήσιμοι ορισμοί

2.1. Τι είναι δεδομένα

Σκοπός τα παρούσας πτυχιακής εργασίας, όπως προαναφέραμε είναι η συλλογή δεδομένων, για διαφορετικούς σκοπούς. Οτιδήποτε μπορεί να περιέχει ένα αρχείο, η εφαρμογή μας το θεωρεί δεδομένο. Σαν δεδομένο, στην πλατφόρμα μας, είναι οποιαδήποτε πληροφορία παρέχεται από τα υποστηριζόμενα data sources, καθώς και πληροφορίες αυτών, όπως ημερομηνία δημιουργίας, styling κ.α.. Οτιδήποτε από τα δεδομένα μας, μπορεί να εμπλακεί σε μια δομή, και να χρησιμοποιηθεί για την εξαγωγή συμπερασμάτων.

2.2. Τι είναι συλλογή δεδομένων

Συλλογή δεδομένων, είναι ένα σύνολο δεδομένων, οργανωμένα με ορισμένο από τον χρήστη, τρόπο, τα οποία, ο χρήστης χρησιμοποιεί. Την δομή με την οποία, η συλλογή θα χρησιμοποιείται από τον χρήστη, ορίζεται από τον ίδιο και μόνο, και είναι επεξεργάσιμη οποιαδήποτε στιγμή.

2.3. Τι είναι αυτοματοποίηση συλλογής δεδομένων

Τα δεδομένα, παρέχονται από πηγές δεδομένων, διαφορετικού τύπου (ετερογενείς). Οι πηγές, μπορεί να είναι αρχεία όπως xml, excel, csv, που ως γνωστό έχουν τελείως διαφορετική δομή μεταξύ τους. Τα δεδομένα σε κάθε πηγή παρέχονται με τελείως διαφορετικό τρόπο, όπως και σε κάθε είδους πηγή τα δεδομένα δεν είναι οργανωμένα ομοιογενώς. Για παράδειγμα δύο διαφορετικά xml είναι δομημένα τελείως διαφορετικά και οι πληροφορίες ομαδοποιημένες σε διαφορετικά tags (tag: Ονομασία τίτλου, μιας πληροφορίας που βρίσκεται μέσα σε xml file – Θα αναφερθούμε αναλυτικά παρακάτω). Η εφαρμογή, υποστηρίζει το data parsing από τελείως διαφορετικά αρχεία, καθώς επίσης και όταν αυτά έχουν τελείως διαφορετική δομή μεταξύ τους. Ο χρήστης μπορεί να ορίσει, σε ποιο σημείο βρίσκεται η εκάστοτε πληροφορία που τον ενδιαφέρει, και αυτόματα η εφαρμογή μας την «τραβάει» από το σημείο που ο χρήστης όρισε.

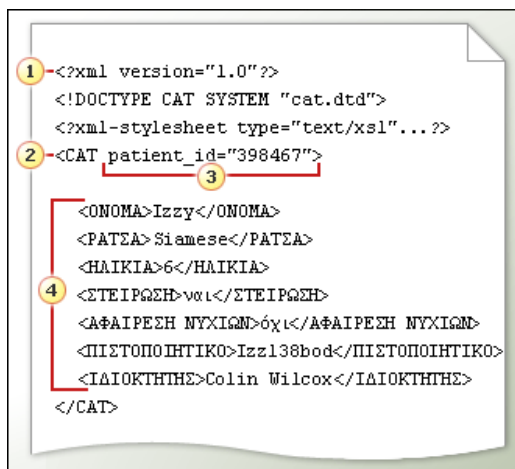
2.4. Τι είναι web based application

Ο τίτλος της πτυχιακής μας, αναφέρει την έκφραση web based application. Με τον όρο αυτό, γίνεται κατανοητό, ότι η εφαρμογή μας θα δουλεύει διαδικτυακά και όχι τοπικά σε κάθε υπολογιστή. Κάθε χρήστης μπορεί να έχει πρόσβαση, από οπουδήποτε στον κόσμο με την χρήση του διαδικτύου. Το interface της εφαρμογής, θα πρέπει να είναι σχεδιασμένο για να υποστηρίζεται από web browsers. Επίσης η εφαρμογή θα πρέπει να έχει προδιαγραφές για να δουλεύει μέσω web, δηλαδή ανταλλαγή δεδομένων server – client, με τέτοιο τρόπο ώστε να είναι εύχρηστο, γρήγορο και ασφαλές.

2.5. Τι είναι xml

Η XML, είναι μία γλώσσα που χρησιμοποιείται για την περιγραφή μίας σημειακής γλώσσας. Ο τεχνικός όρος μιας τέτοιας γλώσσας είναι μετα-γλώσσα. Χρησιμοποιώντας την XML ένας προγραμματιστής μπορεί να προσδιορίσει τις σημειακές γλώσσες που περιγράφουν ηλεκτρονικά κυκλώματα, πληροφορίες για ανταλλαγή ηλεκτρονικών δεδομένων, τα αρχεία που παράγονται από τους διακομιστές Web, μηχανικά μέρη αεροσκαφών και ούτω καθεξής. Ένας προγραμματιστής καθορίζει μία συγκεκριμένη γλώσσα χρησιμοποιώντας την XML κι ένα εργαλείο, στην συνέχεια παίρνει τα αρχεία XML που περιέχουν κείμενα σε αυτήν την γλώσσα και εκτελεί μία επεξεργασία όπως η μετατροπή τους σε αρχεία MS Word ή σε κάποια άλλη μορφή που μπορεί να παρουσιαστεί από προγράμματα γραφικών.

Περισσότερα μέρη ενός αρχείου δεδομένων XML



Εικόνα 1: Δομή XML

- 1 Δηλώσεις
- 2 Ριζικό στοιχείο
- 3 Ιδιότητα
- 4 Ετικέτες και δεδομένα

Εκτός από τα δεδομένα και τις ετικέτες, τα αρχεία δεδομένων XML περιέχουν δηλώσεις, ριζικά στοιχεία και ιδιότητες. Τα ονόματα μπορεί να ακούγονται μυστήρια, αλλά μετά από μερικούς σύντομους ορισμούς, θα δείτε ότι αυτά τα ονόματα είναι απλά και λογικά.

Οι δηλώσεις XML είναι προτάσεις στο αρχείο δεδομένων που το προσδιορίζουν και άλλα αρχεία με τα οποία συνεργάζεται. Σε αυτή την εικόνα, η πρώτη δήλωση αναφέρει ότι το αρχείο είναι XML. (Αν αυτό

δεν αναφέρεται, τα προγράμματα δεν γνωρίζουν τον τρόπο εργασίας με αυτό.) Η δεύτερη δήλωση ονομάζει ένα αρχείο το οποίο διέπεται από δομή δεδομένων για αυτό το αρχείο και άλλα αρχεία του είδους του. Η τρίτη δήλωση αναφέρει ένα αρχείο το οποίο χρησιμοποιείται για τον έλεγχο της εμφάνισης των δεδομένων και το οποίο επιτρέπει σε ένα σύστημα XML την αναγνώριση των δεδομένων και αυτό επιτρέπει σε ένα σύστημα XML για αναγνώριση των δεδομένων.

Ριζικό στοιχείο σημαίνει μία ετικέτα η οποία περιέχει όλες τις άλλες ετικέτες και όλα τα δεδομένα σε κάθε συγκεκριμένη καταχώρηση. Τα αρχεία XML απαιτούν ριζικές ετικέτες. Για τα αρχεία δεδομένων αυτού του κτηνιάτρου, η ριζική ετικέτα είναι ο μεμονωμένος πελάτης με όνομα.

Η επιλογή Ιδιότητες γίνεται αποθήκευση πληροφοριών σχετικά με τις ετικέτες και τα δεδομένα που υπάρχουν σε αυτές. Οι τύποι πληροφοριών στις ιδιότητες ορίζονται από την XML για κάθε είδος ετικέτας. Σε αυτό το παράδειγμα, κάθε πελάτης με ζώο πρέπει να έχει έναν αριθμό ασθενή ως μια ιδιότητα για το ριζικό στοιχείο.

Ετικέτες και δεδομένα είναι, βέβαια, τα βασικά στοιχεία του αρχείου. Μπορείτε, επίσης, να σκεφθείτε τις ετικέτες και τα δεδομένα ως εξαρτημένα στοιχεία τα οποία περιέχονται στο ριζικό στοιχείο <CAT> ... </CAT> .

2.6. Τι είναι excel

Το Excel είναι ένα πρόγραμμα επεξεργασίας λογιστικών φύλλων που μπορεί να κάνει πολύπλοκους μαθηματικούς υπολογισμούς, να δημιουργήσει και να επεξεργαστεί μια βάση δεδομένων, να κάνει γραφήματα, να επεξεργαστεί κείμενο κ.ά.

Με την έννοια *Λογιστικά Φύλλα (Spreadsheets)*, εννοούμε μια μεγάλη κατηγορία προγραμμάτων, τα οποία αποτελούν ένα πολύ ισχυρό εργαλείο σε πολλούς τομείς και όχι μόνο στις λογιστικές εφαρμογές. Στα προγράμματα αυτά, η μνήμη του Η/Υ προσομοιώνεται σαν μια μεγάλη ηλεκτρονική σελίδα, που είναι χωρισμένη σε γραμμές και στήλες.

Έτσι δημιουργούνται τα λεγόμενα *κώτταρα* ή *κελιά (cells)*, στα οποία μπορούμε να γράψουμε δεδομένα, όπως αριθμούς, κείμενα, τύπους υπολογισμών κ.ά. Από την επεξεργασία των δεδομένων του λογιστικού φύλλου, μπορούμε να εξάγουμε πολλά συμπεράσματα και να τα παρουσιάσουμε με διάφορους τρόπους, από απλούς αριθμούς έως και πολύπλοκα γραφήματα.

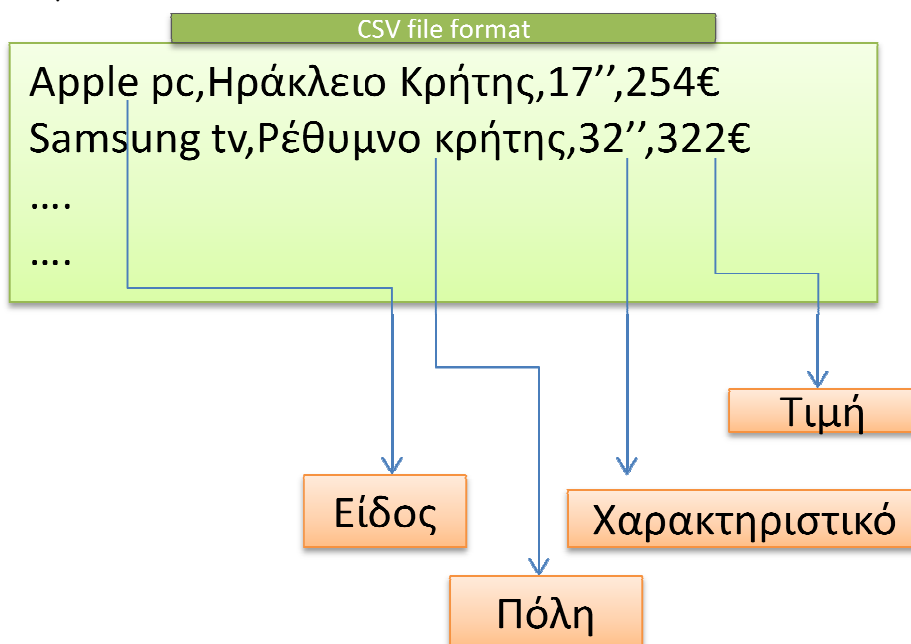
Ενδεικτικό screenshot

Peak	Location	Feet	Meters
1. Mount Everest	Nepal	29,035 ft.	8,850 m.
2. K2	Pakistan/China	28,250 ft.	8,611 m.
3. Kangchenjunga	Nepal/India	28,169 ft.	8,586 m.
4. Lhotse	Nepal	27,920 ft.	8,501 m.
5. Makalu	Nepal	27,765 ft.	8,462 m.
6. Cho Oyu	Nepal	26,906 ft.	8,201 m.
7. Dhaulagiri	Nepal	26,794 ft.	8,167 m.
8. Manaslu	Nepal	26,758 ft.	8,156 m.
9. Nanga Parbat	Pakistan	26,658 ft.	8,125 m.
10. Annapurna	Nepal	26,545 ft.	8,091 m.
11. Gasherbrum I	Pakistan/China	26,470 ft.	8,068 m.
12. Broad Peak	Pakistan/China	26,400 ft.	8,047 m.
13. Gasherbrum II	Pakistan/China	26,360 ft.	8,035 m.
14. Shisha Pangma	Nepal/Tibet	26,289 ft.	8,013 m.

Εικόνα 2: Δομή Excel

2.7. Τι είναι csv

Το csv είναι ένα αρχείο κειμένου, το οποίο περιέχει δεδομένα, χωρισμένα μεταξύ τους με ειδικά σύμβολα. Το πιο συνηθισμένο, και αυτό το οποίο χρησιμοποιούμε για την δική μας εφαρμογή είναι το σύμβολο « , » (comma). Τα δεδομένα δηλαδή τα οποία περιέχονται σε csv αρχεία και θα πρέπει η εφαρμογή μας να διαβάσει, είναι χωρισμένα με το σύμβολο comma. Επίσης κάθε σύνολο δεδομένων είναι γραμμένο σε μία γραμμή. Κάθε διαφορετικό σύνολο, τοποθετείται σε νέα γραμμή. Σε αυτού του είδους τα αρχεία δεν έχουμε ετικέτες, ούτε κάποιο άλλο χαρακτηριστικό που να μπορούμε να αναγνωρίσουμε τα δεδομένα. Το μόνο που, ξέρει η εφαρμογή μας, είναι η σειρά των δεδομένων.



Εικόνα 3:Δομή CSV

2.8. Παραδείγματα χρήσης

Παρακάτω σας περιγράφω μερικά παραδείγματα που η εφαρμογή μας, μπορεί να χρησιμοποιηθεί.

Παράδειγμα1

Σε όλους, είναι γνωστές, πλατφόρμες σύγκρισης τιμών προϊόντων, ή αναζήτησης κάποιας υπηρεσίας. Έστω ότι κάποιος χρήστης θέλει να αγοράσει, μέσω διαδικτύου, μια τηλεόραση. Αυτό που πρέπει να κάνει είναι η έρευνα αγοράς. Να ψάξει δηλαδή σε διάφορα ηλεκτρονικά καταστήματα, και να συγκρίνει τις τιμές που το διαθέτει το κάθε ένα ώστε να την αγοράσει όσο το δυνατόν σε χαμηλότερο κόστος.

Το πρόβλημα δημιουργείται όταν ο χρήστης αρχίσει να χάνεται μέσα στο πλήθος των καταστημάτων και στη σύγκριση τιμών και υπηρεσιών από το ένα στο άλλο. Η εφαρμογή μας, μπορεί να λύσει με πολύ εύκολο τρόπο, το πρόβλημα αυτό. Το μόνο που πρέπει να γίνει, είναι να μας προσφέρουν τα καταστήματα, κάποια πηγή στην οποία θα αναγράφονται τα απαραίτητα στοιχεία που ο χρήστης ζητάει.

Αφού γίνει η συλλογή των δεδομένων από την εφαρμογή μας, ο χρήστης θα μπορεί να κάνει μια αναζήτηση, με την λέξη κλειδί, και η εφαρμογή μας, θα του εμφανίσει όλα τα αντίστοιχα αποτελέσματα.

Η σειρά με την οποία τα αποτελέσματα εμφανίζονται, είναι στη διάθεση του χρήστη. Μπορεί να ταξινομήσει τα αποτελέσματα, με όποιο τρόπο επιθυμεί. Στο συγκεκριμένο παράδειγμα, η σωστή ταξινόμηση είναι βάση τιμής.

Παράδειγμα 2

Έστω ότι εργαζόμαστε σε μία μεγάλη εταιρία με πολλούς υπαλλήλους. Κάθε υπάλληλος συνεργάζεται με πολλούς, εργαζόμενους άλλων εταιριών και έχει στην κατοχή του, όλες τις επαφές των συνεργατών του (Όνοματεπώνυμο, εταιρία, τηλέφωνο, e-mail). Η εταιρία, αποφασίζει να συγκεντρώσει τις επαφές, όλων των υπαλλήλων, σε ένα μέρος, ώστε με μία αναζήτηση, να μπορούν να βρουν τα στοιχεία κάθε συνεργάτη, όλοι οι εργαζόμενοι της εταιρίας.

Υπάρχουν πολλοί τρόποι για να πραγματοποιηθεί κάτι τέτοιο, όμως χρειάζεται πολύς χρόνος από κάθε εργαζόμενο, να εισάγει τις δικές του επαφές, στο ηλεκτρονικό μέσο. Επίσης, κάθε εργαζόμενος, έχει τις επαφές του σε διαφορετική μορφή. Δηλαδή, άλλος τις έχει σε excel, άλλος σε csv αρχείο κτλ.

Η παρούσα πτυχιακή εργασία, μπορεί να χρησιμοποιηθεί για την υλοποίηση ενός τέτοιου μηχανισμού. Οι εργαζόμενοι, το μόνο που έχουν να κάνουν, είναι να κάνουν upload τα αρχεία τους. Με λίγα μόνο clicks, οι επαφές περνάνε στο σύστημα, και έτσι, μπορούν όλοι οι εργαζόμενοι να έχουν πρόσβαση, και να βρίσκουν εύκολα, τα στοιχεία του συνεργάτη που χρειάζονται.

Σημαντικό επίσης, πλεονέκτημα, είναι ότι όλοι οι εργαζόμενοι, της εταιρίας, έχουν πρόσβαση σε όλες τις επαφές. Έτσι μπορούν να αναζητήσουν και να βρουν στοιχεία επαφών, που δεν διέθεταν πριν, αρκεί κάποιος συνάδελφός τους, να τα έχει καταχωρίσει.

Παράδειγμα 3

Το γνωστό σε όλους μας, wikipedia.org, μια πλατφόρμα ανοιχτής ηλεκτρονικής εγκυκλοπαίδειας, θα μπορούσε να είχε φτιαχτεί με την εφαρμογή μας. Στο δικτυακό τόπο αυτό, μπορεί ο κάθε χρήστης να προσθέσει οποιαδήποτε πληροφορία, σχετική με το οτιδήποτε. Για παράδειγμα μπορεί να γράψει πληροφορίες για το Keyword (ΤΕΙ ΗΡΑΚΛΕΙΟΥ), περιγράφοντας το πρόγραμμα σπουδών την τοποθεσία και άλλες πληροφορίες. Έτσι, αν κάποιος χρήστης του internet, θελήσει να αναζητήσει πληροφορίες για την συγκεκριμένη λέξη κλειδί, θα του δοθούν οι πληροφορίες που έχουν καταχωρηθεί.

3. Τεχνολογία

3.1. Βάση δεδομένων

Η Βάση Δεδομένων που επιθυμούμε να χρησιμοποιήσουμε πρέπει να διαθέτει όλα εκείνα τα χαρακτηριστικά που θα την καθιστούν γρήγορη στην πρόσβαση, ασφαλή, αξιόπιστη και εύκολη στη χρήση. Γι' αυτό έγινε μια σχετική έρευνα στο Internet, τα αποτελέσματα της οποίας παρουσιάζονται στο Παράρτημα Β (Tables 10 έως 15), για τις διαθέσιμες Βάσεις Δεδομένων.

Μετά από τα συγκριτικά αποτελέσματα της έρευνας, που φαίνονται στο Παράρτημα Β (Tables 16 έως 24) καταλήξαμε στη MySQL ως τη βάση δεδομένων που θα υποστηρίξει το σύστημά μας, καθώς είναι ανοικτού κώδικα, γρήγορη, αξιόπιστη, εύκολη στη χρήση και απόλυτα συμβατή με την γλώσσα προγραμματισμού (php) που χρησιμοποιήσαμε.

3.1.1. MySQL

Η MySQL είναι το πιο δημοφιλές, ανοικτού κώδικα, σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων, το οποίο έχει δημιουργηθεί, διανέμεται και υποστηρίζεται από το MySQLAB.

Χαρακτηριστικά

Η MySQL είναι ένα σύστημα διαχείρισης βάσεων δεδομένων: Μια βάση δεδομένων είναι μια δομημένη συλλογή δεδομένων. Μπορεί να αφορά τα πάντα, από μια λίστα με ψώνια ως και μια έκθεση με εικόνες ή τεράστια μεγέθη πληροφορίας σε ένα εταιρικό δίκτυο. Για την προσθήκη, πρόσβαση και επεξεργασία των δεδομένων που είναι αποθηκευμένα σε μια βάση δεδομένων, απαιτείται ένα σύστημα διαχείρισης βάσεων δεδομένων όπως είναι η MySQL. Καθώς οι υπολογιστές είναι ικανοί να χειρίζονται μεγάλες ποσότητες δεδομένων, τα συστήματα διαχείρισης βάσεων δεδομένων παίζουν κεντρικό ρόλο στον προγραμματισμό, ως ανεξάρτητες υπηρεσίες ή ως μέρη άλλων εφαρμογών.

Η MySQL είναι ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων: Μια σχεσιακή βάση δεδομένων αποθηκεύει τα δεδομένα σε ξεχωριστούς πίνακες, αντί να τα τοποθετεί σε έναν ενιαίο χώρο. Το γεγονός αυτό προσθέτει ταχύτητα και ευελιξία. Το SQL μέρος από το «MySQL» αντιστοιχεί στο «Structured Query Language», δηλαδή Δομημένη Γλώσσα Επερωτήσεων. Η SQL είναι η πιο κοινά τυποποιημένη γλώσσα που χρησιμοποιείται για πρόσβαση στις βάσεις δεδομένων και προσδιορίζεται από το πρότυπο ANSI/ISO. Το SQL πρότυπο εξελίσσεται από το 1986 και υπάρχουν διάφορες εκδόσεις.

Η MySQL είναι ανοικτού κώδικα: Οποιοσδήποτε το επιθυμεί μπορεί να «κατεβάσει» το λογισμικό της MySQL από το Internet και να το χρησιμοποιήσει ή τροποποιήσει χωρίς κανένα κόστος. Ο χρήστης μπορεί να αλλάξει τον πηγαίο κώδικα του λογισμικού αυτού, εφόσον γνωρίζει τι επιτρέπεται και τι όχι να κάνει με το λογισμικό σύμφωνα με την άδεια GPL (GNU General Public Licence).

Ο MySQL Database Server είναι πολύ γρήγορος, αξιόπιστος και εύκολος στη χρήση: Ο MySQL Server είχε αρχικά αναπτυχθεί για να υποστηρίξει μεγάλες βάσεις δεδομένων πολύ πιο γρήγορα από τις υπάρχοντες λύσεις και έχει χρησιμοποιηθεί επιτυχώς σε περιβάλλοντα με υψηλά απαιτητική παραγωγή για πολλά χρόνια τώρα. Η συνδεσιμότητα, ταχύτητα και ασφάλεια καθιστά τον MySQL Server ιδιαίτερα κατάλληλο για πρόσβαση σε βάσεις δεδομένων μέσω Internet.

Ο MySQL Server λειτουργεί σε συστήματα πελάτη/εξυπηρετητή ή εμπεδωμένα συστήματα: Το λογισμικό της MySQL είναι ένα σύστημα πελάτη/εξυπηρετητή το οποίο αποτελείται από έναν multi-threaded SQL εξυπηρετητή που υποστηρίζει διαφορετικά backends, πολλαπλά διαφορετικά προγράμματα πελάτη και βιβλιοθήκες, εργαλεία διαχείρισης και μια ευρεία κλίμακα από διεπαφές προγραμματισμού εφαρμογών (Applications Programming Interfaces). Επίσης ο MySQL εξυπηρετητής παρέχεται ως εμπεδωμένη multi-threaded βιβλιοθήκη που μπορεί να συνδεθεί με τη εφαρμογή και να γίνει ένα μικρότερο, γρηγορότερο και ευκολότερο στη διαχείριση, ανεξάρτητο προϊόν.

Υπάρχει διαθέσιμη μια μεγάλη ποσότητα διαθέσιμου λογισμικού MySQL

3.2. Γλώσσες προγραμματισμού

Οι γλώσσες προγραμματισμού που μελετήθηκαν για την κατασκευή της εφαρμογής μας είναι οι PHP, Javascript, AJAX, JQuery, Html – Css.

3.2.1. HTML

Η HTML (HyperText Markup Language) είναι η επικρατέστερη markup γλώσσα για τη δημιουργία ιστοσελίδων. Παρέχει τα μέσα για την περιγραφή της δομής μιας πληροφορίας βασισμένης σε κείμενο μέσα σε ένα έγγραφο (δηλώνοντας συγκεκριμένο κείμενο ως επικεφαλίδες, παραγράφους, λίστες και λοιπά) και για τη συμπλήρωση αυτού του κειμένου με αλληλεπιδραστικές φόρμες, ενσωματωμένες εικόνες και άλλα αντικείμενα. Η HTML είναι γραμμένη με μορφή ετικετών (γνωστές ως tags) μέσα σε angle brackets, <>. Μπορεί επίσης να περιγράψει, ως ένα βαθμό, την παρουσίαση και τα semantics ενός εγγράφου και μπορεί να περιλαμβάνει ενσωματωμένο scripting language κώδικα, ο οποίος μπορεί να επηρεάσει τη συμπεριφορά των browser και άλλο HTML επεξεργαστών.

Προσδιορισμός του όρου HyperText Markup Language:

HyperText είναι η μέθοδος με την οποία κινείσαι μέσα στο web – κλικάροντας πάνω σε ειδικό κείμενο που ονομάζεται hyperlink, το οποίο σε πηγαίνει στην επόμενη σελίδα. Το γεγονός ότι είναι *hyper* απλά σημαίνει ότι δεν είναι γραμμική διαδικασία, δηλαδή, μπορείς να πας σε οποιοδήποτε μέρος στο Internet όποτε θελήσεις απλά κάνοντας κλικ στα αντίστοιχα link. Δεν υπάρχει συγκεκριμένη σειρά.

Markup καλείται αυτό που κάνουν οι HTML ετικέτες στο κείμενο που βρίσκεται μέσα σε αυτές. Το μαρκάρουν ως ένα συγκεκριμένο τύπο κειμένου (για παράδειγμα κείμενο σε *italics*).

Η HTML είναι γλώσσα, *Language*, καθώς έχει λέξεις κώδικα και σύνταξη όπως κάθε άλλη γλώσσα.

Πώς λειτουργεί

Η HTML αποτελείται από μια σειρά από σύντομους κώδικες που έχουν τυπωθεί, από τον δημιουργό της ιστοσελίδας, μέσα σε ένα αρχείο κειμένου και ονομάζονται ετικέτες (tags). Έπειτα το κείμενο σώζεται ως html αρχείο, και προβάλλεται μέσα από έναν browser, όπως είναι ο Internet Explorer. Αυτός ο browser διαβάζει το αρχείο και μεταφράζει το κείμενο σε ορατή μορφή. Υπάρχουν διαθέσιμοι είτε text-editors είτε προγράμματα με graphical editors για τη δημιουργία HTML αρχείων.

HTML markup

Η HTML περιέχεται από διάφορους τύπους οντοτήτων συμπεριλαμβανομένου των στοιχείων (elements), παραμέτρων (attributes), τύπων δεδομένων (data types), και αναφορών σε χαρακτήρες (character references).

Προσδιορισμός τύπου εγγράφου (Document Type Definition)

Το DTD προσδιορίζεται στην αρχή του εγγράφου γραμμένο σε HTML και περιέχει γραμματική η οποία προσδιορίζει το επιτρεπόμενο και απαγορευμένο περιεχόμενο αυτού του εγγράφου. Οι browsers, όταν διαβάσουν το HTML έγγραφο ελέγχουν τον τύπο εγγράφου έτσι ώστε να αποφασίσουν το στυλ διάταξης.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

Στοιχεία (Elements)

Τα στοιχεία είναι η βασική δομή για HTML markup και έχουν δύο βασικές ιδιότητες: τις παραμέτρους και το περιεχόμενο. Το περιεχόμενο κάθε παραμέτρου και κάθε στοιχείου έχει συγκεκριμένους περιορισμούς που πρέπει να ακολουθούνται έτσι ώστε να είναι έγκυρο το HTML έγγραφο. Ένα στοιχείο έχει μια ταμπέλα έναρξης (π.χ. <label>) και μια ταμπέλα τερματισμού (π.χ. </label>). Οι παράμετροι των στοιχείων περιέχονται στην ταμπέλα έναρξης και το περιεχόμενο τοποθετείται ανάμεσα στις ταμπέλες (π.χ. <label attribute="value">Content</label>). Μερικά στοιχεία. Όπως το
 δεν έχουν περιεχόμενο και έτσι δε χρειάζεται ταμπέλα τερματισμού.

Παρακάτω παρατίθενται μερικοί από τους τύπους markup στοιχείων που χρησιμοποιούνται από την HTML:

Το *markup* *δόμησης* (*structural markup*) περιγράφει το σκοπό του κειμένου. Για παράδειγμα, το `<h2>Golf</h2>` τοποθετεί το “Golf” ως τίτλο δευτέρου επιπέδου. Το markup δόμησης δε δηλώνει κάποιο συγκεκριμένο τύπο απόδοσης (rendering), αλλά οι περισσότεροι browser έχουν τυποποιήσει τον τρόπο που τα στοιχεία φορμάρονται.

Το *markup* *παρουσίασης* (*presentational markup*) περιγράφει την παρουσίαση του κειμένου ανεξάρτητα από τη λειτουργία του. Για παράδειγμα, το `boldface` επιδεικνύει ότι η συσκευές εξόδου πρέπει να παρουσιάσουν το κείμενο σε **bold** και το `<i>italic</i>` ότι το κείμενο πρέπει να παρουσιαστεί ως *italic*.

Το *markup* *υπερκειμένου* (*Hypertext markup*) συνδέει σημεία του εγγράφου με άλλα έγγραφα. Για παράδειγμα, το `Wikipedia` δηλώνει ότι η λέξη “Wikipedia” θα είναι υπερσύνδεσμος που όταν πατηθεί θα μεταφέρει τον χρήστη στο url που δηλώνεται στην παράμετρο href.

Παράμετροι (Attributes)

Οι παράμετροι ενός στοιχείου είναι ζεύγη ονόματος – τιμής που χωρίζονται με ίσον, “=”, και γράφονται μέσα στην ταμπέλα έναρξης του στοιχείου, μετά το όνομα του στοιχείου. Η τιμή πρέπει να εσωκλείεται μέσα σε μονά ή διπλά εισαγωγικά. Τα περισσότερα στοιχεία έχουν κοινές παραμέτρους όπως id, class, style και title και πολλά επίσης μπορούν να έχουν παραμέτρους σχετικές με τη γλώσσα όπως lang και dir.

Η παράμετρος id προσδίδει στο στοιχείο ένα χαρακτηριστικό που είναι μοναδικό σε μέσα σε όλο το έγγραφο. Αυτή η παράμετρος μπορεί να χρησιμοποιηθεί από τα style sheets για να αποδίδουν ιδιότητες παρουσίασης, από τους browser για να κάνουν focus στο συγκεκριμένο στοιχείο ή από τα scripts για να αλλάζουν τα περιεχόμενα ή την παρουσίαση του στοιχείου.

Η παράμετρος class παρέχει έναν τρόπο ταξινόμησης ίδιων στοιχείων για λόγους παρουσίασης. Για παράδειγμα, ένα έγγραφο HTML (ή ένα σύνολο από έγγραφα) μπορεί να χρησιμοποιούν τον προσδιορισμό class="notation" για να υποδεικνύουν ότι όλα τα στοιχεία με αυτή την τιμή κλάσης είναι υφιστάμενα του κυρίου κειμένου του εγγράφου (ή εγγράφων).

Η παράμετρος style χρησιμοποιείται για τον προσδιορισμό συγκεκριμένης παρουσίασης του στοιχείου, δηλαδή συνδέει το στοιχείο με ένα συγκεκριμένο style sheet.

Η παράμετρος title χρησιμοποιείται για να επισυνάπτεται subtextual επεξήγηση για ένα στοιχείο. Στους περισσότερους browser αυτή η παράμετρος επιδεικνύεται, όπως συχνά αναφέρεται, ως tooltip.

Το στοιχείο span μπορεί να χρησιμοποιηθεί για να περιγράψει αυτές τις διάφορες παραμέτρους:

```
<span id='anId' class='aClass' style='color:red;' title='Hypertext MarkupLanguage'>HTML</span>
```

Ως αποτέλεσμα παίρνουμε τη λέξη **HTML** και όταν ο κέρσορας «σημαδέψει» τη λέξη θα παρουσιάζεται ο τίτλος στους περισσότερους browser.

Άλλα markup

Η HTML προσδιορίζει ένα σύνολο από 256 αναφορές σε οντότητες χαρακτήρων και ένα σύνολο από 1,114,050 αναφορές σε αριθμητικούς χαρακτήρες. Και τα δύο αυτά σύνολα επιτρέπουν σε ξεχωριστούς χαρακτήρες να γραφτούν με απλό markup δηλαδή με γράμματα. Έτσι επιτρέπεται οι χαρακτήρες “<” και “&” να μεταφραστούν ως δεδομένα χαρακτήρων αντί για markup. Δηλαδή, το “<” κανονικά επιδεικνύει την αρχή της ταμπέλας και το “&” επιδεικνύει την αρχή μιας αναφοράς σε οντότητα χαρακτήρων, ενώ αν γραφτούν ως “<” και “&” στην έξοδο θα πάρουμε “<” και “&” αντίστοιχα.

Η HTML επίσης προσδιορίζει διάφορους τύπους δεδομένων για το περιεχόμενο του στοιχείου, όπως είναι δεδομένα σεναρίου (script data) και δεδομένα stylesheet, και μια πληθώρα από τύπους για τις τιμές των παραμέτρων όπως ID, ονόματα (names), URL, αριθμοί (numbers), μονάδες μήκους, γλώσσες (languages), media descriptors, κωδικοποιήσεις χαρακτήρων (character encodings), ημερομηνίες και ώρα, και πολλά άλλα. Όλοι αυτοί οι τύποι δεδομένων είναι εξειδικεύσεις των δεδομένων χαρακτήρων.

Χρήσιμα Tags της HTML

Βασική Δομή

<!DOCTYPE>	Το DTD (Δήλωση τύπου δεδομένων) λέει στον browser ποια έκδοση της HTML χρησιμοποιείται.
<html>... </html>	Οι σταθερές ετικέτες ανοίγματος και κλεισίματος για κάθε HTML σελίδα. Τα πάντα μπαίνουν ανάμεσα σε αυτές τις ετικέτες.
<!-- ... -->	Ό,τι τοποθετηθεί ανάμεσα θεωρείται σχόλιο και παραβλέπεται από τον browser.
<head> ... </head>	Ξεκινά την επικεφαλίδα του εγγράφου. Ό,τι τοποθετηθεί ανάμεσα βοηθά τον browser και τις μηχανές αναζήτησης να κατηγοριοποιήσουν την σελίδα.
<title> ... </title>	Περιέχεται στο <head> ... </head> και αφορά τον τίτλο που αποδίδεται στη σελίδα.
<meta>	Περιέχεται στο <head> ... </head>. Σύνολο από ετικέτες που δίνουν πληροφορίες σχετικά με τη σελίδα στις μηχανές αναζήτησης. <base>: Αλλάζει τον προκαθορισμένο στόχο συνδέσμου ή το σχετικό URL <link>: Βοηθά στο συσχετισμό stylesheet και icon στη σελίδα
<body> ... </body>	Οτιδήποτε ορατό στη σελίδα μπαίνει ανάμεσα σε αυτές τις ετικέτες

Σύνδεσμοι (Links)

<a> ... 	Κάνει το κείμενο που εσωκλείεται υπερσύνδεσμο σε ένα άλλο αρχείο
--------------	--

Λίστες (Lists)

 ... 	Δημιουργεί μια διατεταγμένη λίστα, όπου κάθε στοιχείο της αριθμείται. : Κάθε στοιχείο της λίστας ξεκινά με αυτή την ετικέτα.
 ... 	Δημιουργεί μια μη διατεταγμένη λίστα όπου κάθε στοιχείο της είναι bulleted. : Κάθε στοιχείο της λίστας ξεκινά με αυτή την ετικέτα.
<dl> ... </dl>	Δημιουργεί μια definition λίστα. <dt>: Δημιουργεί έναν definition όρο. <dd>: Δημιουργεί μια definition, που παρουσιάζεται κάτω από τον parent όρο και στοιχίζεται αριστερά.

Πολυμέσα (Multimedia)

	Τοποθετεί μια εικόνα στη σελίδα.
<embed>	Προσθέτει ένα πολυμεσικό στοιχείο κατευθείαν στη σελίδα, επιτρέποντας στον browser να το εκτελέσει με ένα plug-in.
<script>...</script>	Προσθέτει ένα script, συνήθως JavaScript, στη σελίδα. <noscript> ... </noscript>: Εσωκλείεται οτιδήποτε επιθυμεί να παρουσιάζεται στους browsers που δεν υποστηρίζει script.

Πίνακες (Tables)

<table> ... </table>	Τοποθετεί έναν πίνακα στη σελίδα. <caption> ... </caption>: Περιέχει το caption του πίνακα και εμφανίζεται στην κορυφή. <tr> ... </tr>: Ξεκινά μια νέα γραμμή πίνακα με κελιά <td> ... </td>: Περιέχει ένα κελί του πίνακα με περιεχόμενα. <th> ... </th>: Το ίδιο με τα κελιά του πίνακα, αλλά με τα περιεχόμενα bold και στοιχισμένα στο κέντρο. <thead> ... </thead>: Προσδιορίζει την επικεφαλίδα του πίνακα. <tbody> ... </tbody>: Προσδιορίζει το κυρίως μέρος του πίνακα. <tfoot> ... </tfoot>: Περικλείονται γύρω από το κάτω μέρος του πίνακα. <colgroup>: Επιτρέπει τον προσδιορισμό παραμέτρων για όλη τη στήλη, κάθε στήλη ξεχωριστά.
----------------------	---

Πλαίσια (Frames)

<frameset> ... </frameset>	Ξεκινά ένα νέο frame layout. <frame>: Προσδιορίζει ένα frame μέσα στο frameset. <noframes> ... </noframes>: Εάν ένας επισκέπτης έχει παλιό browser που δεν υποστηρίζει πλαίσια, μπορεί να τοποθετηθεί ένα μήνυμα ανάμεσα στις ετικέτες.
<iframe> ... </iframe>	Τοποθετεί ένα inline ή 'floating' πλαίσιο.

Φόρμες (Forms)

<form> ... </form>	Ξεκινά μια περιοχή φόρμας. <input>: Επιτρέπει να προστεθούν διάφορα πεδία εισόδου, όπως text-boxes, checkboxes, radio buttons, submit και reset buttons ανάλογα με την τιμή της παραμέτρου type. <textarea> ... </textarea>: Προσθέτει μια περιοχή με πολλαπλές γραμμές εισόδου. <select> ... </select>: Τοποθετεί ένα άδειο drop-down box. Μπορούν να προστεθούν επιλογές με τις ετικέτες <option>... </option> <fieldset> ... </fieldset>: Επιτρέπει την ομαδοποίηση στοιχείων φόρμας σε λογικές διατάξεις. <legend> ... </legend>: Βάζει τίτλο στο fieldset. <label> ... </label>: Τοποθετεί ετικέτα σε ένα στοιχείο της φόρμας.
--------------------	---

Τυποποίηση κειμένου (Text Formatting)

<address> ... </address>	Εσωκλείει την υπογραφή και διεύθυνση του συγγραφέα σε italics.
 ... 	Μετατρέπει το κείμενο σε bold.
<big> ... </big>	Κάνει το κείμενο ένα μέγεθος μεγαλύτερο.
<blockquote>...</blockquote>	Στοιχίζει το κείμενο και από τις δύο πλευρές.
 	Αλλάζει γραμμή.
<center> ... </center>	Κεντράρει ό,τι βρίσκεται ανάμεσα στις ετικέτες.
<cite> ... </cite>	Εσωκλείει μια αναφορά, συνήθως σε italics.
<code> ... </code>	Αφορά υπολογιστικό ή HTML κώδικα που υπάρχει στο κείμενο και παρουσιάζεται ως μικρό κείμενο.
<dfn> ... </dfn>	Ένας προσδιορισμός και προσδίδεται με italics.
<div> ... </div>	Περικλείει ευδιάκριτα μέρη της σελίδας κυρίως για τη στοίχιση, αλλά χρησιμοποιούνται και για layer setup.
 ... 	Δίνει έμφαση στο κείμενο σε italics.
 ... 	Καθορίζει τις ιδιότητες γραμματοσειράς για το κείμενο που περικλείεται.
<hx> ... </hx>	Βάζει το κείμενο επικεφαλίδα από τιμές 1 ως 6 (h1 ... h6)
<hr>	Βάζει μια γκρι οριζόντια γραμμή κατά μήκος της σελίδας.
<i> ... </i>	Μετατρέπει το κείμενο σε italics.
<kbd> ... </kbd>	Υπονοεί ότι το κείμενο πρέπει να εισαχθεί στο πληκτρολόγιο.
<nobr> ... </nobr>	Λέει στο κείμενο να μην αλλάξει γραμμή στο τέλος της οθόνης αλλά να συνεχίσει όσο χρειαστεί. <wbr>: Εισάγει linebreak στο κείμενο.
<p> ... </p>	Φτιάχνει παράγραφο.
<pre> ... </pre>	Εμφανίζει το κείμενο με ρυθμισμένο μήκος γραμματοσειράς και διατηρεί το φORMάρισμα του αρχικού κειμένου.
<q> ... </q>	Χρησιμοποιείται για παραπομπές.

<s>...</s> ή <strike>...</strike>	Εμφανίζει το κείμενο με strike-through.
<samp> ... </samp>	Υποδηλώνει sample έξοδο από μια φόρμα ή πρόγραμμα.
<small> ... </small>	Κάνει το κείμενο ένα μέγεθος μικρότερο.
 ... 	Δίνει έμφαση στο κείμενο κάνοντάς το bold.
_{...}	Κάνει το κείμενο subscript, δηλαδή ^{this} .
^{...}	Κάνει το κείμενο superscript, δηλαδή ^{this} .
<tt> ... </tt>	Κάνει το κείμενο fixed-width και mono-spaced γραμματοσειράς, όπως μια παλιά δακτυλογραφική μηχανή.
<u> ... </u>	Υπογραμμίζει το κείμενο

Συμπεράσματα

Η HTML χρησίμευσε ιδιαίτερα στην ανάπτυξη της εφαρμογής μας, καθώς είναι ο τρόπος με τον οποίο μπορέσαμε να κάνουμε την εφαρμογή μας web-based και συμβατή με τους browsers που σχεδόν όλοι οι χρήστες χρησιμοποιούν.

3.2.2. Css

3.2.2.1. Δημιουργία CSS

Η δημιουργία των επάλληλων φύλλων στυλ (Cascading Style Sheets) ήταν κίνηση του οργανισμού W3C. Η παλιά έκδοση της γλώσσας HTML, συνδύαζε το περιεχόμενο, την δομή, και τις οδηγίες μορφοποίησης σε ένα μόνο έγγραφο, κάτι που απλοποιούσε τα πράγματα αλλά δεν παρείχε και πολλές δυνατότητες. Οι άνθρωποι του W3C οραματίστηκαν ένα νέο σύστημα στο οποίο οι οδηγίες μορφοποίησης θα αποθηκεύονταν χωριστά από το περιεχόμενο και την δομή, έτσι ώστε να μπορούν να εφαρμοστούν όχι μόνο σε μία παράγραφο ή ιστοσελίδα αλλά και σε μια ολόκληρη τοποθεσία ιστού αν το επιθυμούσε ο σχεδιαστής της.

Οι αρχικές προδιαγραφές του CSS περιορίζονταν κυρίως στην επανάληψη των παλιών εφέ της HTML. Ωστόσο με το δεύτερο επίπεδο του συστήματος CSS, δηλαδή το CSS2 το οποίο δημοσιεύθηκε το 1998, προστέθηκαν νέες δυνατότητες - με σημαντικότερη τη δυνατότητα τοποθέτησης στοιχείων σε μια ιστοσελίδα με μεγάλη ακρίβεια.

3.2.2.2. CSS1 και CSS2

Το CSS σχεδιάστηκε για να χωρίσει την μορφοποίηση από τις προδιαγραφές του HTML. Όπως και τα στυλ σε ένα πρόγραμμα επεξεργασίας κειμένου, το CSS παρέχει ένα μηχανισμό για να ορίσει εύκολα και να αλλάξει την μορφοποίηση χωρίς να αλλάξει την υπόστρωμη δομή του κώδικα. Η λέξη "cascade" (επάλληλο) στο όνομα προέρχεται από το γεγονός πως οι προδιαγραφές επιτρέπουν πολλαπλά φύλλα στυλ να αλληλεπιδρούν, επιτρέποντας έτσι ανεξάρτητα Web κείμενα να μορφοποιούνται ελαφρώς διαφορετικά από τα συγγενικά τους.

Η δεύτερη έκδοση του CSS (CSS2) χτίζεται στις ικανότητες της πρώτης έκδοσης, προσθέτοντας περισσότερες ιδιότητες από τις οποίες μπορεί να επιλέξει ένας Web designer.

Η χρήση φύλλων στυλ CSS για την διάταξη των ιστοσελίδων παρέχει αρκετά πλεονεκτήματα σε σχέση με την χρήση άλλων μεθόδων.

Καταρχάς τα φύλλα στυλ CSS είναι ό,τι πρέπει για την δημιουργία ρευστών διατάξεων, οι οποίες αναπτύσσονται ή συρρικνώνονται ανάλογα με το μέγεθος της οθόνης του επισκέπτη. Εξάλλου η διατήρηση του κειμένου χωριστά από τις οδηγίες της διάταξης σημαίνει ότι μπορείτε πολύ εύκολα να εφαρμόσετε την ίδια διάταξη σε μια ολόκληρη τοποθεσία ιστού με μία μόνο κίνηση.

Έτσι θα μπορείτε αργότερα να αλλάξετε τη διάταξη ολόκληρης της τοποθεσίας τροποποιώντας απλώς το αρχείο CSS. Ο συνδυασμός φύλλων στυλ CSS και (X)HTML οδηγεί επίσης στην παραγωγή μικρότερων αρχείων, κάτι που σημαίνει ότι οι επισκέπτες μας δεν θα χρειάζεται να περιμένουν τόσο πού για να δουν την τοποθεσία μας. Τέλος επειδή τα φύλλα στυλ CSS και η (X)HTML αποτελούν αναμφισβήτητα τα ισχύοντα πρότυπα, οι σελίδες που συμμορφώνονται στους

κανόνες τους είναι εξασφαλισμένο ότι θα υποστηρίζονται στις μελλοντικές εκδόσεις των φυλλομετρητών (και θα ζητούνται όλο και περισσότερο από τους επαγγελματίες σχεδιαστές ιστοσελίδων).

Το κυριότερο μειονέκτημα των φύλλων στυλ CSS σε ότι αφορά τη διάταξη είναι ότι οι παλιότεροι φυλλομετρητές είτε δεν τα καταλαβαίνουν είτε τα παρερμηνεύουν.

3.2.3. Javascript

Η Javascript είναι μια γλώσσα σεναρίων που δημιουργήθηκε αποκλειστικά για χρήση σε web browsers έτσι ώστε να γίνουν οι ιστοσελίδες πιο δυναμικές. Η HTML από μόνη της είναι ικανή να παρουσιάζει στατικές σελίδες, που από τη στιγμή που δημοσιοποιηθούν δεν αλλάζουν και πολύ ωστόσο πατηθεί σε αυτές ένας σύνδεσμος για τη μετάβαση σε μια άλλη σελίδα. Προσθέτοντας Javascript κώδικα σε τέτοιες σελίδες, τότε γίνονται εφικτές αλλαγές στο έγγραφο, όπως αλλαγές στο κείμενο, στα χρώματα, στις επιλογές που είναι διαθέσιμες σε μια drop-down λίστα και πολλές άλλες. Η Javascript είναι client-side γλώσσα, που σημαίνει ότι οποιαδήποτε αλλαγή γίνεται στην πλευρά του client, δηλαδή του αναγνώστη. Έτσι, δεν απαιτούνται συνεχείς αιτήσεις στο server για να λειτουργήσουν τα JavaScripts, κάτι το οποίο θα καθυστερούσε υπερβολικά τη διαδικασία. Στην πραγματικότητα, η Javascript συχνά χρησιμοποιείται για την εκτέλεση λειτουργιών που διαφορετικά θα επιβάρυναν τον server.

Τα JavaScripts ολοκληρώνονται στο περιβάλλον του browsing και έτσι μπορούν να πάρουν πληροφορίες σχετικά με τον browser και την HTML σελίδα και να τις τροποποιήσουν, συνεπώς να αλλάζουν τον τρόπο που παρουσιάζονται τα πράγματα στην οθόνη. Επίσης μπορούν να αντιδράσουν σε γεγονότα (events), όπως όταν ο χρήστης κάνει κλικ με το ποντίκι ή δείχνει σε ένα συγκεκριμένο στοιχείο της σελίδας. Το πιο σημαντικό απ' όλα είναι ότι η Javascript είναι εύκολη στη μάθηση και πολύ χρήσιμη για τις ιστοσελίδες.

DOM (Document Object Model)

Το DOM είναι το πλαίσιο εργασίας στο οποίο δουλεύει η Javascript, καθώς η JavaScript είναι μια μορφή αντικειμενοστραφούς προγραμματισμού. Αυτό σημαίνει ότι μπορούμε να θεωρήσουμε όλα τα στοιχεία που χρειάζονται για τη δημιουργία της σελίδας ως αντικείμενα. Το ίδιο το έγγραφο είναι ένα αντικείμενο, το οποίο αποτελείται από άλλα αντικείμενα όπως οι φόρμες, οι εικόνες και οι πίνακες. Ακόμη, οι φόρμες αποτελούνται και αυτές με τη σειρά τους από αντικείμενα, όπως text boxes και κουμπιά.

Όλα αυτά τα αντικείμενα έχουν ιδιότητες, με τιμές που προσδιορίζουν το χρώμα, το μήκος τους, κ.α. Η JavaScript μπορεί να διαβάσει αυτές τις ιδιότητες και να τις αλλάξει, ή να αντιδράσει σε γεγονότα που συμβαίνουν σε αντικείμενα, που στιγμιαία αλλάζουν το αντικείμενο στο παράθυρο του browser. Ένα script μπορεί να αντιδράσει σε μια αλληλεπίδραση του χρήστη με τη σελίδα or it can run all by itself.

Οι ενέργειες που εκτελεί το script σε αντικείμενα ή μαζί με αυτά ονομάζονται *μέθοδοι (methods)*. Αυτές είναι συναρτήσεις (functions) που μετατρέπονται σε αντικείμενα. Η αντιμετώπιση των γεγονότων που προκαλούνται από τον χρήστη, όπως ένα κλικ ή μια κίνηση του ποντικιού, επιτυγχάνεται από εντολές που καλούνται *event handlers*. Αυτές οι δύο αρχές (methods και event handlers) αποτελούν τη βάση του προγραμματισμού σε JavaScript. Το DOM επιτρέπει την πρόσβαση σε αυτά τα αντικείμενα σελίδας.

Υλοποίηση (Implementation)

Ο JavaScript κώδικας μπορεί είτε να ενσωματωθεί στο ίδιο αρχείο που είναι γραμμένη και η HTML, είτε να γραφτεί σε ξεχωριστό αρχείο με την κατάληξη .js και να γίνει link. Παρακάτω φαίνεται ένα ενσωματωμένο script το οποίο γράφει στην έξοδο "Hello World":

```
<script type="text/javascript">
<!--
document.write("<i>Hello World!</i>");
//-->
</script>
```

Η ετικέτα <script> μπορεί να περιέχει κάθε είδους κώδικα, γι' αυτό στην παράμετρο "type" ορίζουμε ποιο συγκεκριμένο κώδικα χρησιμοποιούμε έτσι ώστε να διευκολύνεται η διαδικασία της μεταγλώττισης.

Τα σχόλια γύρω από τον script κώδικα χρησιμεύουν ώστε οι παλαιότεροι browser που δεν κατανοούν script να αγνοήσουν την ετικέτα <script>. Οποιοσδήποτε άλλος browser που μπορεί να διαβάσει script θα αγνοήσει τα σχόλια και θα εκτελέσει το script.

Εξωτερικά scripts (External scripts)

Για την εισαγωγή script από κάποιο εξωτερικό αρχείο, αρχικά χρειάζεται η δημιουργία αυτού του αρχείου χωρίς την ετικέτα <script> και χωρίς τα σχόλια και την αποθήκευσή του με κατάληξη .js . Έπειτα αυτό γίνεται link μέσω της παραμέτρου "src" στην ετικέτα <script> του HTML αρχείου. Έτσι, για το παραπάνω παράδειγμα το JS αρχείο θα είχε τη μορφή:

```
document.write("Hello World!");
```

και η ετικέτα <script> θα έπαιρνε τη μορφή:

```
<script type="text/javascript" src="theJS.js"></script>
```

Με αυτόν τον τρόπο και άλλες σελίδες μπορούν να έχουν πρόσβαση στο JS αρχείο, χωρίς να χρειάζεται να επαναλαμβάνουν τον κώδικα.

Είναι απαραίτητο η εισαγωγή ή εγγραφή script να γίνει μέσα στο <head> tag του αρχείου, το οποίο διαβάζει πρώτο ο browser, έτσι ώστε να γνωρίζει την ύπαρξή του προτού αυτό καλεστεί από π.χ. το πάτημα ενός κουμπιού.

```
<noscript>
```

Η ετικέτα <noscript> είναι ένας τρόπος για να προσδίδεται εναλλακτικό περιεχόμενο σε browser οι οποίοι δεν υποστηρίζουν JavaScript. Οι παλιοί browser δε θα καταλάβουν τη σημασία της ετικέτας και θα παρουσιάσουν ό,τι βρίσκεται μέσα σε αυτή, ενώ οι πιο καινούριοι browser θα το καταλάβουν και θα αγνοήσουν το περιεχόμενό της. Παρακάτω φαίνεται ένα παράδειγμα χρήσης της ετικέτας αυτής:

```
<noscript>
```

```
<p>Sorry, your browser does not support JavaScript.</p>
```

```
</noscript>
```

3.2.4. AJAX

Το ακρωνύμιο AJAX αντιστοιχεί στο «Asynchronous Javascript and XML» και είναι μια οικογένεια τεχνολογιών οι οποίες είναι διαθέσιμες εδώ και πολλά χρόνια.

Ο όρος *Asynchronous* σημαίνει ότι όταν στέλνεις μια αίτηση στο server περιμένεις την απάντηση, ενώ παράλληλα μπορείς να κάνεις και άλλα πράγματα. Πιθανόν, η απάντηση να μην έρθει αμέσως, έτσι ορίζεις μια συνάρτηση (function) η οποία θα περιμένει να επιστρέψει η απάντηση από τον server και να αντιδράσει ανάλογα όταν αυτό συμβεί.

Η *JavaScript* χρησιμοποιείται για να γίνει η αίτηση στο server. Μόλις επιστρέψει η απάντηση, θα χρειαστεί να χρησιμοποιήσετε περισσότερη JavaScript για τη διαμόρφωση του μοντέλου εγγράφου αντικειμένου (DOM) της τρέχουσας σελίδας, έτσι ώστε να δει ο χρήστης ότι η αίτηση ολοκληρώθηκε με επιτυχία.

Τα δεδομένα που επιστρέφουν από τον server πακετάρονται σε ένα snippet της XML, έτσι ώστε να μπορεί να γίνει επεξεργασία με Javascript. Αυτά τα δεδομένα μπορεί να είναι οτιδήποτε και με οποιοδήποτε μήκος.

Η όλη παραπάνω διαδικασία δεν είναι κάτι καινούριο. Γίνεται μια αίτηση αρχείου (το οποίο συχνά θα είναι ένα server-side script, κωδικοποιημένο με π.χ.PHP) και λήψη μιας σελίδας ως απάντηση. Έτσι λειτουργεί το web άλλωστε, η μόνη διαφορά είναι ότι τώρα μπορούμε να κάνουμε αιτήσεις χρησιμοποιώντας JavaScript.

Cross – browser Ajax

Δυστυχώς η τεχνολογία Ajax αντιμετωπίζεται ελαφρώς διαφορετικά από τον Internet Explorer, από τον Opera και από τους Mozilla-based browsers όπως ο Firefox. Αυτό αφήνει ανοικτές δύο επιλογές: χρήση code branching ώστε να στέλνεται ο σωστός κώδικας σε κάθε browser βασισμένος στο μοντέλο που υποστηρίζουν, ή χρήση βιβλιοθήκης Javascript η οποία μετατρέπει τον Ajax κώδικα σε ένα μόνο αντικείμενο που αντιμετωπίζει τις ασυμβατότητες στους browser.

Η δεύτερη επιλογή είναι η πιο προτιμητέα καθώς μια JavaScript βιβλιοθήκη, όπως η Sarissa, περιέχει τις μεθόδους για τη δημιουργία αιτήσεων και μεθόδους οι οποίες βοηθούν στην επεξεργασία XML που επιστρέφει ως απάντηση. Έτσι, απαλλασσόμαστε από τις περιπλοκότητες της Ajax και ο κώδικας γίνεται πιο “κομψός”. Για τη χρήση εξωτερικής βιβλιοθήκης Javascript απαιτείται η δήλωσή της στην αρχή του αρχείου, δηλαδή:

```
<script type="text/javascript" src="sarissa.js"></script>
```

Κάνοντας την αίτηση

Αρχικά, πρέπει να δημιουργηθεί η αίτηση (request) και η μέθοδος για να γίνει κάτι τέτοιο καλείται “XML HTTP Request”. Έτσι έχουμε:

```
var xmlhttp= new XMLHttpRequest();
```

Η παραπάνω εντολή θα ελέγξει τις συμβατότητες του browser και θα επιστρέψει ένα αντικείμενο, το οποίο θα λειτουργήσει ανάλογα με το τι μπορεί ο browser του χρήστη να υποστηρίξει. Έπειτα, καθορίζουμε τη σελίδα που αιτούμαστε από τον server:

```
xmlhttp.open('POST', 'submitPage.jsp', true);
```

Το πρώτο όρισμα της εντολής μπορεί να πάρει τις τιμές ‘GET’ ή ‘POST’, ανάλογα με το αν παίρνουμε δεδομένα από το server χωρίς να αλλάζουμε τίποτα σε αυτόν ή αν στέλνουμε σε αυτόν δεδομένα αντίστοιχα. Το δεύτερο όρισμα αφορά το URL του αρχείου στο οποίο αναφερόμαστε με την αίτησή μας και ενδεχομένως να περνάμε και κάποιες παραμέτρους. Το τρίτο όρισμα καθορίζει αν η αίτηση είναι ασύγχρονη ή σύγχρονη. Αν είναι *σύγχρονη (synchronous)* θέτουμε το όρισμα σε false, αν είναι *ασύγχρονη (asynchronous)* θέτουμε το όρισμα σε true.

Το επόμενο βήμα είναι να ορίσουμε τη συνάρτηση η οποία θα περιμένει να εκτελεστεί ωστόσο επιστρέψει η απάντηση από τον server. Αυτή η συνάρτηση αναλαμβάνει να δείξει στο χρήστη ότι έχει επιτευχθεί η παράδοση, συνήθως με ένα μικρό μήνυμα στη σελίδα. Η συνάρτηση αυτή καλείται *call-back function* και έχει την εξής δομή:

```
xmlhttp.onreadystatechange = function() {  
    if (xmlhttp.readyState == 4) {  
        // Εδώ μπαίνει ο κώδικας επιστροφής  
    }  
}
```

Από τη στιγμή που θα γίνει η αίτηση αυτή περνά από διάφορα στάδια. Στον παραπάνω κώδικα ελέγχουμε αν η παράμετρος *readyState* του αντικειμένου έχει φτάσει στο επίπεδο 4, δηλαδή αν έχει ολοκληρωθεί η απάντηση.

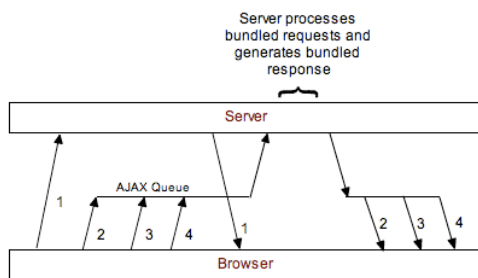
Στο τελευταίο βήμα, είναι η αποστολή της αίτησης:

```
xmlhttp.send(null);
```

Αυτή η εντολή θα στείλει την αίτηση και θα επιστρέψει αμέσως. Μετά περιμένει την απάντηση να φτάσει στο επίπεδο 4, και όταν γίνει αυτό καλείται η *call-back* συνάρτηση. Έτσι, οτιδήποτε δεδομένα στέλνονται πίσω με την απάντηση (αν στέλνονται), είναι διαθέσιμα στη μεταβλητή *xmlhttp.responseText*.

Συμπεράσματα

Η τεχνολογία Ajax είναι πάρα πολύ χρήσιμη, ειδικά όταν φτιάχνουμε σελίδες που περιέχουν φόρμες εγγραφής καθώς επιτρέπει την πλοήγηση μέσα στις φόρμες εγγραφής χωρίς να γίνεται refresh του web browser και έτσι δε χάνονται τα δεδομένα που έχει προηγουμένως εισάγει ο χρήστης.



Εικόνα 4: Ajax Model

3.2.5. JQuery

Η jQuery δεν είναι τίποτα περισσότερο από μία “βιβλιοθήκη” javascript η οποία μας δίνει τη δυνατότητα να δημιουργούμε διάφορα εφέ στην σελίδα μας χωρίς όμως να χρειάζεται η ποσότητα κώδικα που θα χρειαζόταν εάν χρησιμοποιούσαμε την παραδοσιακή javascript. Το μόνο που έχουμε να κάνουμε είναι να κατεβάσουμε τη jQuery ,να την εφαρμόσουμε στο site μας και είμαστε έτοιμοι να δημιουργήσουμε. Τα εφέ που μπορούμε να δημιουργήσουμε είναι πολλά και διάφορα. Για παράδειγμα μπορούμε να εμφανίζουμε ή να εξαφανίζουμε παραγράφους, να αλλάζουμε χρώματα γραμματοσειράς ,να ελέγχουμε δηλαδή με όλα τα html στοιχεία της σελίδας μας και να μπορούμε να ελέγχουμε ένα- ένα από αυτά ξεχωριστά. Η jQuery χρησιμοποιείται ευρέως σήμερα διότι δίνει τη δυνατότητα να προσφέρουμε στους χρήστες της ιστοσελίδας μας αλληλεπίδραση με αυτή.

3.2.6. PHP

Η PHP είναι μια γλώσσα προγραμματισμού ειδικά για την κατασκευή δυναμικών ιστοσελίδων. Με τον όρο δυναμική εννοείται μια ιστοσελίδα που αλλάζει αυτόματα, ανάλογα με τα στοιχεία του θεατή της. Στοιχεία όπως το λειτουργικό του σύστημα, η διεύθυνση IP του κ.ά. Η PHP χρησιμοποιείται όχι για την διακόσμηση μιας ιστοσελίδας αλλά για τον χειρισμό των λειτουργιών και εργασιών που θα διεκπεραιώνει. Συνεπώς, ο κώδικας που γράφεται για μια ιστοσελίδα σε γλώσσα PHP δεν γίνεται άμεσα αντιληπτός αλλά μετά απο την επέμβαση του θεατή στην ιστοσελίδα. Για να γίνει αυτό κατανοητό: η PHP χρησιμοποιείται ευρέως για τον χειρισμό ιστοσελίδων με δυνατότητες όπως η εγγραφή χρηστών (user registration), τα φόρουμ κ.ά. Λειτουργεί με την βοήθεια της HTML και πλέον και με την XHTML (νέα αναθεωρημένη έκδοση της HTML). Σε συνδυασμό και με την MySQL μπορεί να χρησιμοποιηθεί κάλλιστα για την διαχείριση δεδομένων μέσα σε βάσεις. Λόγω χάρη στο προηγούμενο παράδειγμα της εγγραφής των χρηστών η PHP μπορεί να αποθηκεύει τα ονόματα και τους κωδικούς χρηστών σε μια βάση δεδομένων.

Όσο αφορά την δικιά μας πλατφόρμα, οι λειτουργίες εκτελούνται με χρήση, php γλώσσας προγραμματισμού. Φυσικά το functionality που η ίδια γλώσσα παρέχει, είναι αδύνατον να σας το επισυνάψω στο παρών έγγραφο. Στην δικιά μας εφαρμογή χρησιμοποιήσα αντικειμενοστραφή προγραμματισμό της php, και δημιούργησα functionality, τέτοιο, το οποίο να είναι εύκολα αναπτύξιμο. Χρησιμοποιήσα δεκάδες από τις συναρτήσεις που παρέχει η php, συνδυάζοντας τις, για να υλοποιηθεί η εφαρμογή μας.

3.3. Modules

Για την ανάπτυξη της εφαρμογής μας, ήταν απαραίτητη η χρήση κάποιων έτοιμων επιπρόσθετων κομματιών κώδικα, τα οποία εύκολα ενοποιούνται με το σύνολο της εφαρμογής μας.

3.3.1. Live Search

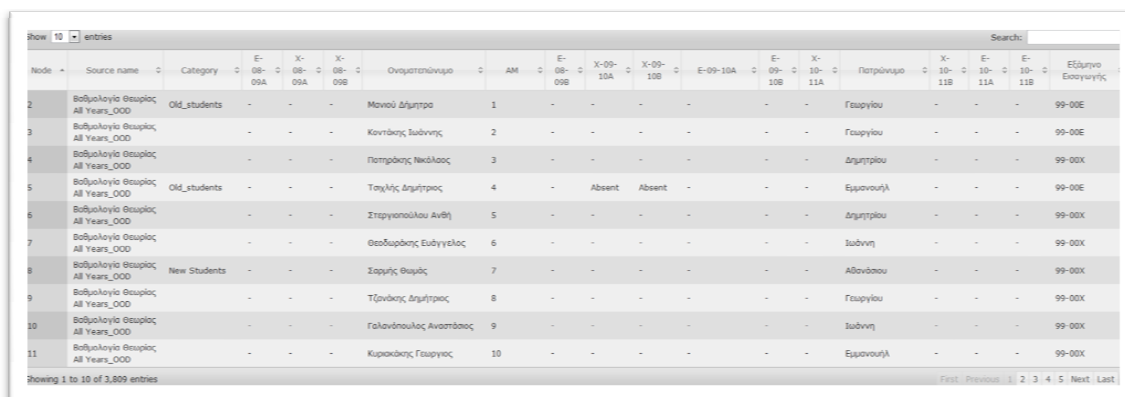
Το Live Search είναι ένα module, φτιαγμένο με php και JQuery, το οποίο κάνει αναζήτηση στην βάση δεδομένων, ώστε να βρεί την ζητούμενη τιμή. Επιστρέφει τα αποτελέσματα, χωρίς να κάνει επαναφόρτωση της σελίδας, κάτι ιδιαίτερα σημαντικό για το Bandwidth και την ταχύτητα της εφαρμογής. Τα αποτελέσματα, συντάσσονται με τρόπο που εγώ έχω ορίσει, και μπορεί ο χρήστης να μεταβεί στην αντίστοιχη σελίδα με μόνο ένα click. Είναι φιλικό προς τον χρήστη και προσφέρει ιδιαίτερη ακρίβεια στα αποτελέσματα.

Το Live Search module βρίσκεται σε δυο διαφορετικά σημεία στην εφαρμογή, εκ των οποίων το πρώτο είναι στο *dashboard*, και το δεύτερο στην σελίδα *Live Search*. Και στις δύο σελίδες τα αποτελέσματα είναι τα ίδια.

Όσο αφορά την μελλοντική ανάπτυξη του, είναι ιδιαίτερα εύκολη, αφού τα αρχεία είναι συντεταγμένα σε χωριστούς φακέλους, με ερωτήματα sql για την βάση δεδομένων.

3.3.2. DataTable module

Το DataTable module, είναι μια μορφή συντεταγμένων αποτελεσμάτων, με διάταξη την οποία ο χρήστης ορίζει. Έχει γίνει τελείως δυναμικό, δηλαδή εμφανίζει για κάθε χρήση διαφορετικά αποτελέσματα και διατάξεις, ανάλογα με τις προτιμήσεις του. Το πιο αξιόλογο όλων των πλεονεκτημάτων του, είναι ότι μπορείς να κάνεις αναζήτηση οποιαδήποτε λέξη, μέσα στα αποτελέσματα, και σου επιστρέφει με την ίδια διάταξη τα αποτελέσματα εκείνα τα οποία ταιριάζουν με την αναζήτηση. Τέλος το DataTable module υποστηρίζει ταξινόμηση των αποτελεσμάτων βάση οποιασδήποτε στήλης ο χρήστης επιθυμεί. Έτσι, αν ο χρήστης έχει πολλά αποτελέσματα, μπορεί να τα ταξινομήσει, κατά αύξουσα ή φθίνουσα σειρά, με βάση οποιαδήποτε στήλη χρειάζεται.



Node	Source name	Category	E-08-09A	X-08-09A	X-08-09B	Όνοματεπώνυμο	AM	E-08-09B	X-09-10A	X-09-10B	E-09-10A	E-09-10B	X-10-11A	Πατρώνυμο	X-10-11B	E-10-11A	E-10-11B	Εύρημα Επαγγελίας
2	Βαθμολογία Θεωρίας All Years_000	Old_students	-	-	-	Μονού Δημήτρα	1	-	-	-	-	-	-	Γεωργίου	-	-	-	99-00E
3	Βαθμολογία Θεωρίας All Years_000		-	-	-	Κωνσταντίνος Ιωάννης	2	-	-	-	-	-	-	Γεωργίου	-	-	-	99-00E
4	Βαθμολογία Θεωρίας All Years_000		-	-	-	Πατριάρχης Νικόλαος	3	-	-	-	-	-	-	Δημητρίου	-	-	-	99-00X
5	Βαθμολογία Θεωρίας All Years_000	Old_students	-	-	-	Τσιγκλής Δημήτρης	4	-	Absent	Absent	-	-	-	Εμμανουήλ	-	-	-	99-00E
6	Βαθμολογία Θεωρίας All Years_000		-	-	-	Σπυριγιόπουλου Ανθή	5	-	-	-	-	-	-	Δημητρίου	-	-	-	99-00X
7	Βαθμολογία Θεωρίας All Years_000		-	-	-	Θεοδωράκης Ευάγγελος	6	-	-	-	-	-	-	Ιωάννη	-	-	-	99-00X
8	Βαθμολογία Θεωρίας All Years_000	New Students	-	-	-	Σαράης Θεοδός	7	-	-	-	-	-	-	Αθανάσιου	-	-	-	99-00X
9	Βαθμολογία Θεωρίας All Years_000		-	-	-	Τζινιάκης Δημήτρης	8	-	-	-	-	-	-	Γεωργίου	-	-	-	99-00X
10	Βαθμολογία Θεωρίας All Years_000		-	-	-	Γαλιανόπουλος Αναστάσιος	9	-	-	-	-	-	-	Ιωάννη	-	-	-	99-00X
11	Βαθμολογία Θεωρίας All Years_000		-	-	-	Κυριακός Γεωργιος	10	-	-	-	-	-	-	Εμμανουήλ	-	-	-	99-00X

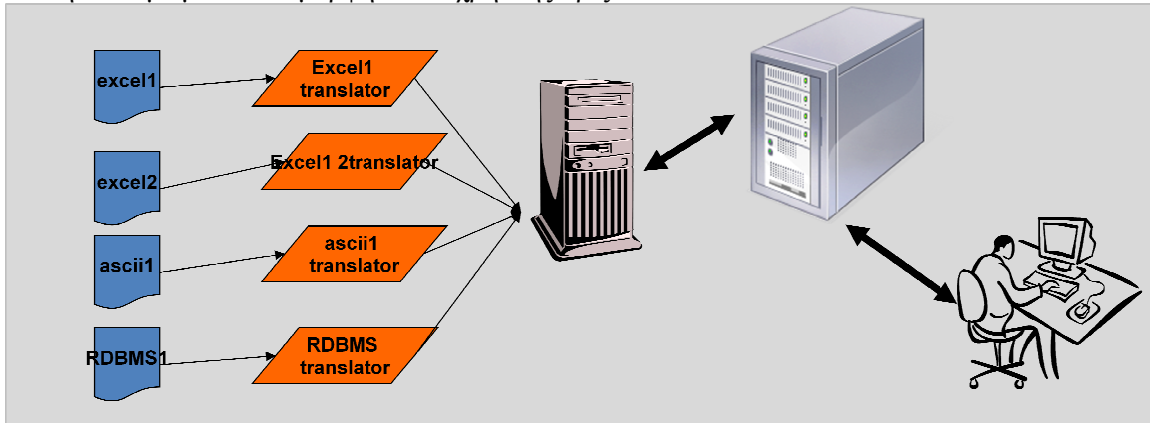
Εικόνα 5:Data Table Model

3.4. Ιδέες υλοποίησης

Για την υλοποίηση μιας τέτοιας εφαρμογής, χρειάζεται αρκετός προγραμματισμός των εργασιών και σκέψη, για το προς τα ποια πορεία θα ακολουθήσουμε, ώστε να φτάσουμε στην υλοποίηση. Ο βασικός στόχος, είναι να μπορέσουμε να συλλέξουμε τα δεδομένα από διάφορες πηγές, και να τα αποθηκεύσουμε σε σημείο και με τρόπο τέτοιο ώστε αργότερα να είναι εύκολα προσπελάσιμα.

3.4.1. Create xml

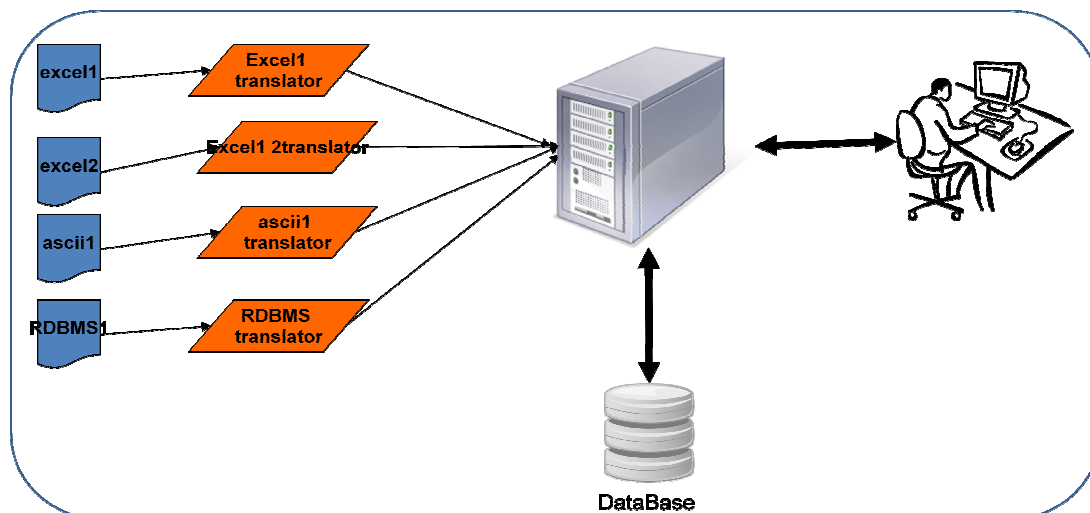
Το σενάριο της ιδέας αυτής, ήταν να επιλέγουμε τα δεδομένα από τις πηγές, που οι χρήστες όριζαν, και με την βοήθεια των files translators να κατασκευάζουμε xml αρχεία και να τα αποθηκεύουμε με τέτοια μορφή που ο χρήστης ορίζει.



Εικόνα 6: Create XML

3.4.2. Database

Το σενάριο αυτό, έχει σαν βάση μια βάση δεδομένων, στην οποία αποθηκεύονται τα δεδομένα, και από εκεί γίνεται η προσπέλασή τους. Τα δεδομένα, αντιγράφονται από την εκάστοτε πηγή, με τον αντίστοιχο translator, στη βάση δεδομένων, και η βάση δεδομένων τα προσφέρει στο χρήστη σε κάθε αίτημα του.



Εικόνα 7: Database

Τελικά, ακολουθήσαμε το τελευταίο σενάριο, σύμφωνα με το οποίο έπρεπε να φτιάξουμε διαφορετικούς data grabbers για κάθε είδους αρχείο. Κάθε αρχείο χρησιμοποιεί τον αντίστοιχο αποκωδικοποιητή για να αντιγραφούν τα δεδομένα του στην βάση δεδομένων.

3.5. Τεχνολογία που χρησιμοποιήθηκε

Ανακεφαλαιώνοντας, για το σύστημα, που από εδώ και στο εξής θα αναφέρω ως Data Grabber Application, χρησιμοποίησα ως βάση δεδομένων την MySQL η οποία προσφέρει γρήγορη πρόσβαση και ανάκτηση δεδομένων και συνοδεύεται από τα MySQL GUI Tools τα οποία προσφέρουν ένα προσιτό γραφικό περιβάλλον στο διαχειριστή για τον έλεγχο της βάσης και την εκτέλεση ερωτοαπαντήσεων. Για τον προγραμματισμό χρησιμοποίησα γλώσσα php και Javascript, γλώσσες συμβατές με τους browsers (φυλλομετρητές) , καθώς για την παρουσίαση – interface χρησιμοποίησα ajax, jquery, html, css.

4. Ανάλυση Λογισμικού

4.1. Εισαγωγή

Η ανάλυση λογισμικού είναι ένα δομημένο σύνολο δραστηριοτήτων που απαιτούνται για την ανάπτυξη ενός συστήματος λογισμικού. Οι δραστηριότητες αυτές αφορούν την: Ανάλυση (Specification), Σχεδίαση (Design), Τεκμηρίωση (Validation) και Εξέλιξη (Evolution) του συστήματος. Με την Ανάλυση προσδιορίζονται όλα όσα πρέπει να υλοποιεί το σύστημα (απαιτήσεις συστήματος) καθώς και εντοπίζονται οι περιορισμοί του, με την Σχεδίαση υλοποιείται το σύστημα, με την Τεκμηρίωση ελέγχεται αν το λογισμικό ανταπεξέρχεται στις αρχικές απαιτήσεις και τέλος στη φάση της Εξέλιξης το σύστημα εξελίσσεται για να καλύψει τυχόν μελλοντικές ανάγκες.

Για τη διεκπεραίωση των παραπάνω δραστηριοτήτων έχουν αναπτυχθεί διάφορα μοντέλα διαδικασίας λογισμικού, τα οποία είναι απλοποιημένες αναπαραστάσεις της διαδικασίας μέσα από συγκεκριμένη οπτική γωνία. Τέτοια μοντέλα είναι το Waterfall (σειριακό μοντέλο που προαπαιτεί πλήρως καταγεγραμμένες απαιτήσεις), το Evolutionary (εξελικτικό μοντέλο με πολλές ενδιάμεσες εκδόσεις) και το Reuse-based (μοντέλο που βασίζεται στην επαναχρησιμοποίηση υαρχόντων στοιχείων τα οποία ενοποιούνται για τη δημιουργία του νέου συστήματος).

Σε κάθε περίπτωση η επιλογή του μοντέλου εξαρτάται από τις απαιτήσεις τους συστήματος καθώς και από τη διάρκεια και το μέγεθος του έργου.

4.2. Καταγραφή Απαιτήσεων

Η μελέτη των απαιτήσεων ενός νέου συστήματος λογισμικού αποτελεί ιδιαίτερα σημαντικό τμήμα της συνολικής φάσης ανάπτυξης του συστήματος και έχει μελετηθεί από πολλές και διαφορετικές μεθοδολογικές προσεγγίσεις. Συνήθως οι απαιτήσεις λογισμικού διαχωρίζονται σε δύο κατηγορίες – αυτές που αφορούν τα λειτουργικά χαρακτηριστικά του συστήματος (λειτουργικές απαιτήσεις) και αυτές που συνιστούν ‘οριζόντια’ ποιοτικά χαρακτηριστικά (quality attributes) και που αναφέρονται ως μη λειτουργικές απαιτήσεις. Οι λειτουργικές απαιτήσεις καταγράφονται, αναλύονται και ενσωματώνονται σε διαδικασίες που συνιστούν το σύστημα λογισμικού. Υπάρχουν πληθώρα τεχνικών τόσο για τη καταγραφή όσο και για την ανάλυση και επικύρωση των λειτουργικών απαιτήσεων ενός συστήματος λογισμικού.

Σε αντίθεση με τις λειτουργικές απαιτήσεις λογισμικού, οι μη-λειτουργικές απαιτήσεις είναι λιγότερο γνωστές και περισσότερο σύνθετες στη μελέτη τους. Παραδείγματα μη-λειτουργικών απαιτήσεων για ένα σύστημα λογισμικού μπορεί να αφορούν μεγέθη όπως η χρηστικότητα του συστήματος, η μεταφερσιμότητά του από μια πλατφόρμα σε μια άλλη η δυνατότητα προσαρμογής του συστήματος, η ασφάλεια και η υποστήριξη χρηστών μετά την παράδοση.

Από τα παραπάνω προκύπτει ότι εξαιτίας των χαρακτηριστικών των μη-λειτουργικών απαιτήσεων (π.χ. αφηρημένες έννοιες, οριζόντια εφαρμογή, κλπ) η μελέτη και ανάλυσή τους δεν είναι ούτε εύκολη ούτε συγκρίσιμη με τη μελέτη και ανάλυση των λειτουργικών απαιτήσεων ενός συστήματος λογισμικού. Η δυσκολία συνίσταται σε διάφορους λόγους. Πρώτα από όλα, οι απαιτήσεις της κατηγορίας αυτής δεν προσδιορίζονται και δεν επικυρώνονται εύκολα. Δεύτερον, δεν υπάρχει ένας ενιαίος και πλήρης κατάλογος μη-λειτουργικών απαιτήσεων που να μπορούν να εξειδικευτούν ανάλογα με το εκάστοτε πρόβλημα. Τρίτον, στην πλειοψηφία των περιπτώσεων οι μη-λειτουργικές απαιτήσεις είναι αντιμαχόμενες η μία την άλλη με αποτέλεσμα να μην καθίσταται εφικτή η προοδευτική ενσωμάτωση και υποστήριξης τους σε ένα σύστημα κάτι που ισχύει για τις λειτουργικές απαιτήσεις. Τέλος, οι μη-λειτουργικές απαιτήσεις αφορούν κυρίως κατασκευαστές συνιστωσών πλατφόρμας (platform developers) με αποτέλεσμα συχνά να μην εξετάζονται κατά το σχεδιασμό ενός συστήματος λογισμικού.

4.2.1. Use Cases.

Περίπτωση χρήσης είναι η περιγραφή ενός σεναρίου ή ενός συνόλου σεναρίων τα οποία είναι στενά συνδεδεμένα μεταξύ τους. Μια περίπτωση χρήσης ορίζει και περιγράφει την αλληλεπίδραση ανάμεσα στους δράστες (actors) και συγκεκριμένα λειτουργικά τμήμα της εφαρμογής (υπηρεσίες του συστήματος). Δηλαδή μια περίπτωση χρήσης περιγράφει σε υψηλό επίπεδο, συστηματικά και μεθοδικά πώς μια εφαρμογή θα χρησιμοποιηθεί για την υλοποίηση

συγκεκριμένων στόχων. Πολλές περιπτώσεις χρήσεις καλύπτουν τελικά όλες τις απαιτήσεις και την επιθυμητή συμπεριφορά της εφαρμογής. Με τις περιπτώσεις χρήσης περιγράφουμε την συμπεριφορά του συστήματος χωρίς να μπαίνουμε σε λεπτομέρειες όπως τρόποι υλοποίησης.

Μια περίπτωση χρήσης συνήθως συμπεριλαμβάνει πολλά σενάρια και δίνει την δυνατότητα αποφυγής περιττών πληροφοριών και περιγραφών των λειτουργιών ενός συστήματος. Αυτό το οποίο προσπαθούμε να επιτύχουμε με τις περιπτώσεις χρήσης είναι μια ολοκληρωμένη άποψη των λειτουργιών και της συμπεριφοράς του λογισμικού του συστήματος.

Οι περιπτώσεις χρήσης είναι δομημένες με συγκεκριμένο και τυπικό τρόπο. Αυτό γίνεται μέσω δομημένης περιγραφής. Τα συστατικά αυτής της δομής είναι:

Όνομα / σύντομη περιγραφή

Χαρακτήρες (actors): που επικοινωνούν με την περίπτωση χρήσης

Προϋποθέσεις (Ποιες συνθήκες πρέπει να ικανοποιούνται στην έναρξη της περίπτωσης χρήσης)

Μέτα-Συνθήκη (Ποιες συνθήκες πρέπει να ικανοποιούνται στην λήξη της περίπτωσης χρήσης)

Ερέθισμα (Trigger): ένα συμβάν που προκαλεί την περίπτωση χρήσης

Σχέσεις: Συσχέτιση (association), Περιλαμβάνει (include), Επεκτείνει (extend), Εξειδικεύει (generalization) :

Ομαλή Ροή Γεγονότων (Περιγραφή της συμπεριφοράς βάση της περίπτωσης χρήσης)

Συνιστώσες ροές γεγονότων

Εναλλακτικές ροές – Εξαιρέσεις: αποκλίσεις από το κύριο σενάριο επιτυχίας

4.2.2. Σενάρια χρήσης (Scenarios of Use).

Τα σενάρια αποτελούν μια μικρογραφία των επιμέρους λειτουργιών ενός συστήματος και ανάλογα με την πιστότητά τους μπορεί να είναι είτε γενικά σενάρια που εμπεριέχουν επιλεγμένες περιπτώσεις χρήσης είτε σενάρια αλληλεπίδρασης που καταγράφουν τρόπους εκτέλεσης μιας περίπτωσης χρήσης. Η χρήση των σεναρίων κατά την διαδικασία ανάλυσης ενός συστήματος λογισμικού βασίζεται στην ιδιότητα που έχουν τα σενάρια να προσφέρουν αφενός λεπτομερείς καταγραφές πρακτικών (δηλαδή το τι κάνουν οι χρήστες ή το ίδιο το σύστημα) και ταυτόχρονα να συνεισφέρουν στον οραματισμό νέων λύσεων με τη χρήση προηγμένων τεχνολογικών εργαλείων.

Συγκεκριμένα, ένα σενάριο προσφέρεται τόσο για την αποκάλυψη καθηκόντων που εκτελούν οι χρήστες και που συνιστούν υπάρχουσες πρακτικές όσο και για τον οραματισμό λύσεων που γίνονται εφικτές με τη χρήση νέων τεχνολογιών και εργαλείων.

Τα σενάρια χρήσης θα πρέπει να ικανοποιούν συγκεκριμένα ποιοτικά χαρακτηριστικά που καθορίζουν την πληρότητα του σεναρίου.

Ένα σενάριο για να είναι πλήρες θα πρέπει να καταγράφεται με μια μορφή που να επιτρέπει περαιτέρω ανάλυση.

Το σενάριο θα πρέπει να προσδιορίζει το στόχο ή το σκοπό που εξυπηρετείται και να γίνεται σαφής αναφορά σε εμπλεκόμενους φορείς (συνιστώσες συστήματος και χρήστες).

Το σενάριο θα πρέπει να συνθέτεται από επιμέρους δραστηριότητες (βήματα) που όταν εκτελεστούν να αποφέρουν το επιθυμητό αποτέλεσμα.

Το σενάριο θα πρέπει να προσδιορίζει την ροή των δραστηριοτήτων (βημάτων) που πρέπει να εκτελεστούν

Για κάθε επιμέρους δραστηριότητα ή βήμα θα πρέπει να υπάρχουν ή να μπορούν να προσδιοριστούν όρια που εξασφαλίζουν την έναρξη και λήξη της δραστηριότητας και του σεναρίου γενικότερα.

Ένα σενάριο θα πρέπει να προσδιορίζει τις προϋποθέσεις που το ολοκληρώνουν και τυχόν εξαιρέσεις που συνήθως συνιστούν εναλλακτικά σενάρια.

Αναφορικά με τις τεχνικές καταγραφής, ένα σενάριο ανάλογα με το βαθμό ωριμότητάς του μπορεί να καταγράφεται με πολλούς διαφορετικούς τρόπους, όπως:

- κείμενο / αφήγηση γεγονότων,
- αριθμημένη ακολουθία βημάτων,
- φόρμα δομημένης περιγραφής
- διαχωρισμένη αφήγηση,
- μοντέλο περιπτώσεων χρήσης,
- διάγραμμα σεναρίου VORD
- μοντέλο ακολουθίας της UML, κλπ.

Επίσης, είναι δυνατόν ένα σενάριο αρχικά καταγεγραμμένο με ένα γενικό τύπο (π.χ. κείμενο) να εξειδικευτεί με τη χρήση μιας άλλης μορφής (π.χ. διαχωρισμένη ακολουθία βημάτων) προκειμένου να γίνει δυνατή η καλύτερη κατανόηση επιμέρους συστατικών του.

4.2.3. Data Base Model.

Το database model είναι μια θεωρία ή μια προδιαγραφή, που περιγράφει τον τρόπο που δομείται και χρησιμοποιείται μια βάση δεδομένων. Χρησιμοποιούνται διάφορες τεχνικές για τη μοντελοποίηση δομών δεδομένων. Τα περισσότερα συστήματα βάσεων δεδομένων χτίζονται γύρω από ένα συγκεκριμένο data model, αν και είναι σύνηθες τα προϊόντα να προσφέρουν υποστήριξη για περισσότερα από ένα μοντέλα. Για κάθε λογικό μοντέλο είναι δυνατές ποικίλες φυσικές τροποποιήσεις και τα περισσότερα προσφέρουν στο χρήστη, ως κάποιο σημείο, έλεγχο στη ρύθμιση των φυσικών τροποποιήσεων, καθώς οι επιλογές που γίνονται έχουν άμεσο αντίκτυπο στη λειτουργικότητα.

Ένα τέτοιο παράδειγμα είναι το σχεσιακό μοντέλο, όπου όλες οι σημαντικές τροποποιήσεις επιτρέπουν τη δημιουργία indexes, τα οποία παρέχουν γρήγορη πρόσβαση στις γραμμές (rows) ενός πίνακα (table), εάν οι τιμές των συγκεκριμένων στηλών είναι γνωστές.

Διάφορα τέτοια μοντέλα έχουν προταθεί εκ των οποίων τα πιο κοινά είναι: Ιεραρχικό μοντέλο (Hierarchical model), Δικτυακό μοντέλο (Network model), Σχεσιακό μοντέλο (Relational model), Entity-relationship, Object-Relational μοντέλο και μοντέλο αντικειμένων (Object model).

Άλλα μοντέλα περιλαμβάνουν : Associative, Concept-oriented, Entity-Attribute-Value, Multi-dimensional model, Semi-structured, Star schema και XML database.

4.2.4. User Interface.

Η διεπαφή χρήστη (User Interface) είναι το σύνολο των μέσων με τα οποία οι χρήστες αλληλεπιδρούν με ένα συγκεκριμένο μηχάνημα, συσκευή, υπολογιστικό πρόγραμμα ή άλλο περίπλοκο εργαλείο (το σύστημα). Η διεπαφή χρήστη παρέχει μέσα εισόδου (input), που επιτρέπουν στους χρήστες να χειρίζονται το σύστημα και μέσα εξόδου (output), που επιτρέπουν στο σύστημα να παράγουν τα εφέ από τη διαχείριση των χρηστών.

Στην επιστήμη των υπολογιστών και στην αλληλεπίδραση ανθρώπου-υπολογιστή, το user interface αναφέρεται στην πληροφορία που αφορά γραφικά, κείμενο και ήχο που παρουσιάζει ένα πρόγραμμα στο χρήστη καθώς και τις ακολουθίες ελέγχου (όπως πάτημα κουμπιών μέσω του πληκτρολογίου, κινήσεις μέσω του ποντικιού και επιλογές μέσω του touchscreen) τις οποίες απασχολεί ο χρήστης για τον έλεγχο του προγράμματος.

Τα πιο κοινά user interfaces είναι τα ακόλουθα δύο:

Graphical user interfaces (GUI): Τα γραφικά user interfaces δέχονται είσοδο από συσκευές όπως το πληκτρολόγιο του υπολογιστή και το ποντίκι και παρέχουν αρθρωτή γραφική έξοδο στην οθόνη του υπολογιστή. Υπάρχουν τουλάχιστον δύο διαφορετικές αρχές που χρησιμοποιούνται ευρέως στο σχεδιασμό GUI: Object-oriented user interfaces (OOUIs) και application oriented interfaces.

Web – based user interfaces : Οι διεπαφές αυτές δέχονται είσοδο και παρέχουν έξοδο παράγοντας ιστοσελίδες, οι οποίες μεταδίδονται μέσω του Internet και είναι ορατές από χρήστες που χρησιμοποιούν το πρόγραμμα κάποιου web browser.

Πιο πρόσφατες υλοποιήσεις αξιοποιούν Java, AJAX, Microsoft .NET ή παρόμοιες τεχνολογίες για να παρέχουν realtime έλεγχο σε ένα πρόγραμμα, εξαλείφοντας έτσι την ανάγκη για ανανέωση (refresh) του web browser βασισμένου σε HTML.

Γενικότερα, οι χρήστες ενός συστήματος επιθυμούν η διεπαφή με την οποία αλληλεπιδρούν να τους παρέχει οικείο γραφικό περιβάλλον, ακόμη και αν ορισμένα χαρακτηριστικά δεν προσδίδουν καμία λειτουργικότητα στο σύστημα (π.χ. εικόνες, βίντεο, κ.α.). Έτσι, η υποστήριξη γραφικών περιβαλλόντων χρήσης θεωρείται ένα δεδομένο χαρακτηριστικό όλων των εμπορικά διαθέσιμων συστημάτων διαχείρισης περιεχομένου. Το Σύστημα μας, καλείται να υποστηρίξει διαφορετικά templates, για κάθε υπηρεσία, ώστε να είναι user-friendly και όσο γίνεται απλούστερο στην χρήση του.

4.3. The info Grabber Application case study

Το πρόγραμμα data grabber application εστιάζει στη διερεύνηση, σχεδιασμό, κατασκευή και χρησιμοποίηση τεχνολογικών εργαλείων που αποσκοπούν στην επεξεργασία των ψηφιακών δεδομένων για την παραγωγή συμπερασμάτων από το χρήστη. Τέτοια συστήματα, παρέχουν την δυνατότητα σύγκρισης δεδομένων, εξαγωγή συμπερασμάτων και ιστορικό, ανεξαρτήτως του τύπου των δεδομένων, ή την πηγή από την οποία προέρχονται. Επίσης η μορφή στην οποία η εφαρμογή μας υλοποιείται, μας δίνει το πλεονέκτημα της ανταλλαγής δεδομένων από χρήστη σε χρήστη, ψηφιακά πλέον.

4.3.1. Περιγραφή συστήματος Info Grabber Application

Το Info Grabber Application βασίζεται στην ανάπτυξη προηγμένων εργαλείων που συνθέτουν ένα τεχνολογικό περιβάλλον βασισμένο στον παγκόσμιο ιστό, μέσω του οποίου μπορούν να γίνονται, μεταξύ άλλων, τα ακόλουθα:

Login/SignUp: Χρήστες μπορούν να εγγράφονται στην εφαρμογή μας, για την δημιουργία της δικιά τους data collection, με δικό τους username και password, χρησιμοποιώντας τα εργαλεία που αποκτούν μέσα από το interface της εφαρμογής, κερδίζοντας όλα τα προσφερόμενα πλεονεκτήματα.

i-Results: Οι χρήστες μπορούν να εξάγουν τα δεδομένα που έχουν καταγράψει από τις πηγές και να τα συγκρίνουν μεταξύ τους, καθώς μπορούν να δούν και συγκρίσεις δεδομένων που δεν ανήκουν στην ίδια ομάδα. (για παράδειγμα ένας φοιτητής με αριθμό μητρώου 1567 μπορεί να συγκριθεί με ιατρικά δεδομένα που ανήκουν σε κάποια ομάδα).

Categories: Τα δεδομένα μας, μπορούν να ομαδοποιηθούν και να δημιουργήσουν μια δομή (structure-node) ανάλογα τις προτιμήσεις του χρήστη. Επιπλέον, κάθε τέτοιο node μπορεί να ομαδοποιηθεί με αντίστοιχα άλλα, και να αποτελέσουν μια ομάδα. Έτσι μπορούμε να συγκρίνουμε 2 διαφορετικές ομάδες – κατηγορίες.

4.3.2. Γενικό σενάριο χρήσης του Info Grabber Application

Ας υποθέσουμε ότι ένας χρήστης έχει συγκεντρωμένα διάφορα δεδομένα σε διάφορα αρχεία. Συγκεκριμένα σας αναλύω ένα παράδειγμα χρήσης της εφαρμογής, που υλοποιείται από sites σύγκρισης τιμών προϊόντων, ώστε ο καταναλωτής να μπορέσει να βρεί την χαμηλότερη. Για να μπορέσει να υλοποιηθεί ένα τέτοιο site θα πρέπει να βρεθούν τα δεδομένα από τα καταστήματα, και να περαστούν σε μια μεγάλη βάση δεδομένων, ώστε οι χρήστες όταν αναζητούν ένα προϊόν να προκύπτει η σύγκριση του ίδιου προϊόντος από κάθε κατάσταση ταξινομημένο κατά αύξουσα σειρά βάσει τιμής. Ο χρήστης μας λοιπόν μπορεί να δημιουργήσει ένα λογαριασμό στην εφαρμογή μας, και να έχει πρόσβαση στο interface της.

Ο χρήστης θα πρέπει να ορίσει τα fields που πρόκειται να του χρησιμεύσουν, συγκεκριμένα στο παράδειγμα μας (έρευνα από <http://www.bestprice.gr>) θα πρέπει να ορίσει τα εξής fields:

- Όνομα προϊόντος
- Περιγραφή προϊόντος
- Τιμή προϊόντος
- Φωτογραφία προϊόντος

τα οποία αντιστοιχούν στις αντίστοιχες πληροφορίες που χρειαζόμαστε. Κάθε λοιπόν σύνολο των τεσσάρων παραπάνω πεδίων αποτελεί ένα Node, το οποίο ο καταναλωτής θα συγκρίνει με άλλα τέτοια, που διαθέτουν ίδια περιγραφή και όνομα (ώστε να είναι για το ίδιο προϊόν), ανάλογα την τιμή τους. Το σύστημα θα τους τα κατατάσει ανάλογα το πεδίο της τιμής και θα τους δίνει την πληροφορία, από ποιο κατάστημα μπορούν να το βρουν φτηνότερα.

Η αμέσως επόμενη αναγκαία διαδικασία που πρέπει να ακολουθηθεί, είναι ο χρήστης να εισάγει τις πηγές δεδομένων, από τις οποίες θα συλλέγει τα δεδομένα. Τέτοιες πηγές είναι αρχεία:

- xml
- excel
- csv

καθώς, μπορεί να εισάγει και digital local sources, δηλαδή πηγές οι οποίες δεν αντιστοιχούν σε κάποιο αρχείο, αλλά ο χρήστης κάνει εισαγωγή δεδομένων απ'ευθείας στην εφαρμογή.

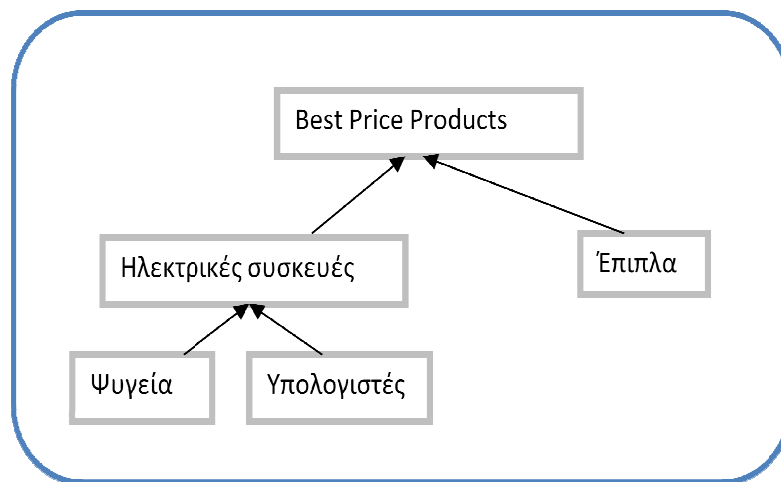
Είναι ευνόητο ότι κάθε κατάστημα θα παρέχει ένα από αυτά τα αρχεία στον χρήστη. Τώρα ο χρήστης θα πρέπει να αντιστοιχήσει για κάθε πηγή, τα fields του με την ανάλογη θέση. Αυτό συμβαίνει γιατί κάθε πηγή δεν δίνει το ίδιο όνομα σε κάθε πληροφορία. Για παράδειγμα την

πληροφορία της τιμής σε κάποια πηγή θα αναφέρεται ως price και σε άλλη product_price. Αφού γίνει η αντιστοίχιση της πηγής με τα αντίστοιχα fields μπορεί ο χρήστης να κάνει το data parsing από κάθε πηγή.

Αξίζει να σημειωθεί ότι κάθε πηγή μπορεί να είναι σε απομακρυσμένο υπολογιστή, ή να γίνει upload στο δικό μας σύστημα από τον τοπικό υπολογιστή του κάθε χρήστη. Τώρα ο χρήστης μας, είναι σε θέση, να κατηγοριοποιήσει τα nodes του, αναλόγως τις κατηγορίες που ο ίδιος θα θελήσει να φτιάξει. Συγκεκριμένα για το παράδειγμα, μπορεί να φτιάξει μια κατηγορία «ηλεκτρικές συσκευές» και μια άλλη «έπιπλα». Στις ηλεκτρικές συσκευές μπορεί να φτιάξει μια υποκατηγορία «ψυγεία» και μια «υπολογιστές» και αντίστοιχα την κατηγορία έπιπλα να την χωρίσει σε υποκατηγορίες όπως αυτός προτιμάει. Ενδεικτικά σας παραθέτω το παρακάτω σχήμα.

Οι κατηγορίες και οι υποκατηγορίες μπορούν να φτάσουν σε οποιοδήποτε βάθος ο χρήστης επιθυμεί καθώς επίσης και να επεξεργαστούν, δηλαδή να μετατεθούν ως παιδιά σε οποιαδήποτε άλλη κατηγορία, μετά την δημιουργία τους.

Γραφικό παράδειγμα:



4.3.3. Καταγραφή απαιτήσεων Info Grabber Application

Όπως προαναφέρθηκε, η καταγραφή των απαιτήσεων είναι ένα ιδιαίτερα σημαντικό κομμάτι στη διαδικασία της ανάπτυξης ενός συστήματος. Παρακάτω παρουσιάζουμε τις απαιτήσεις του συστήματος Info Grabber Application, που αφορούν την συγκεκριμένη αναφορά.

Κωδικό Όνομα Απαίτησης
R1
Προσδιορισμός / Τίτλος Απαίτησης Fields Creation
Περιγραφή Απαίτησης <ul style="list-style-type: none"> • Απαίτηση για δυναμική ταξινόμηση των δεδομένων • Δυναμική μετονομασία ενός field • Προσθήκη ή αφαίρεση πεδίων • Χρήση άπεριόριστων πεδίων ανα χρήστη

Κωδικό Όνομα Απαίτησης
R2
Προσδιορισμός / Τίτλος Απαίτησης Δημιουργία Πηγών
Περιγραφή Απαίτησης <ul style="list-style-type: none"> • Κατασκευή/Δημιουργία Source Instance • Upload Αρχείου • Υποστήριξη απομακρυσμένης πηγής

Κωδικό Όνομα Απαίτησης
R3
Προσδιορισμός / Τίτλος Απαίτησης Υποστήριξη ετερογενών πηγών
Περιγραφή Απαίτησης <ul style="list-style-type: none"> • Ο χρήστης μπορεί να αποθηκεύσει δεδομένα από διαφορετικές πηγές δεδομένων. • Ο χρήστης θα μπορεί να δημιουργήσει εγγραφές χωρίς να υπάρχει αρχείο

Κωδικό Όνομα Απαίτησης
R5
Προσδιορισμός / Τίτλος Απαίτησης Preview Source
Περιγραφή Απαίτησης Ο χρήστης για να μπορέσει να αντιστοιχήσει τα fields με τα ανάλογα πεδία της εκάστοτε πηγής θα πρέπει να μπορεί να έχει μια προεπισκόπηση των δεδομένων της πηγής

Κωδικό Όνομα Απαίτησης
R6
Προσδιορισμός / Τίτλος Απαίτησης Connections
Περιγραφή Απαίτησης Ο χρήστης με χρήση απλής φόρμας θα πρέπει να μπορεί να αντιστοιχήσει τα ορισμένα από αυτόν, fields με την αντίστοιχη θέση που έχουν μέσα στο αρχείο. Ο χρήστης δεν θα πρέπει να έχει δικαίωμα αλλαγής της πληροφορίας. Η μόνη του παρέμβαση, θα μπορεί να είναι στην κατηγοριοποίηση των δεδομένων.

Κωδικό Όνομα Απαίτησης
R7
Προσδιορισμός / Τίτλος Απαίτησης Nodes
Περιγραφή Απαίτησης Τα δεδομένα θα πρέπει να τακτοποιούνται και να ορίζουν μια οντότητα, αποτελούμενα από τις πληροφορίες που παρέχει η εκάστοτε πηγή. Το κάθε πακέτο πληροφοριών θα έχει ένα μοναδικό χαρακτηριστικό (node_id). Το κάθε node θα μπορεί να ανήκει σε μια από τις κατηγορίες που ο χρήστης έχει φτιάξει.

Κωδικό Όνομα Απαίτησης
R8
Προσδιορισμός / Τίτλος Απαίτησης Grouping
Περιγραφή Απαίτησης <ul style="list-style-type: none"> • Δυνατότητα δημιουργίας κατηγοριών. • Δυνατότητα δημιουργίας υποκατηγοριών • Οι υποκατηγορίες θα μπορούν να έχουν σαν parent οποιαδήποτε άλλη κατηγορία, εκτός από αυτές που είναι παιδιά τους. • Σε κάθε κατηγορία θα μπορεί να αντιστοιχηθούν nodes • Κάθε node θα πρέπει να ανήκει σε μία κατηγορία

Κωδικό Όνομα Απαίτησης
R9
Προσδιορισμός / Τίτλος Απαίτησης Αναζήτηση με κάθε πεδίο
Περιγραφή Απαίτησης Ο χρήστης θα μπορεί να αναζητήσει οποιαδήποτε τιμή επιθυμεί και σαν αποτέλεσμα να παίρνει οτιδήποτε είναι σχετικό με την αναζήτησή του.

Κωδικό Όνομα Απαίτησης
R10
Προσδιορισμός / Τίτλος Απαίτησης Dashboard
Περιγραφή Απαίτησης Θα πρέπει η εφαρμογή να έχει μια κεντρική σελίδα από την οποία ο χρήστης θα μπορεί να ανακατευθύνεται στην εκάστοτε υπηρεσία επιθυμεί. Στην κεντρική σελίδα θα μπορεί να παρακολουθεί την πορεία του project του.

Κωδικό Όνομα Απαίτησης
R11
Προσδιορισμός / Τίτλος Απαίτησης Compare Results
Περιγραφή Απαίτησης <ul style="list-style-type: none"> • Εισαγωγή λέξεων – κλειδιών. • Επισκόπηση των αποτελεσμάτων • Δυνατότητα πλοήγησης ανάμεσα στα αποτελέσματα • Δυνατότητα ταξινόμησης των αποτελεσμάτων όπως ο χρήστης επιθυμεί. Με οποιοδήποτε field και αύξουσα ή φθίνουσα τιμή. Η τιμή δεν περιορίζεται μόνο σε αριθμητικές μεταβλητές αλλά και σε αλφαριθμητικές.

Κωδικό Όνομα Απαίτησης
R12
Προσδιορισμός / Τίτλος Απαίτησης Online Results
Περιγραφή Απαίτησης Ο χρήστης θα πρέπει να μπορεί να δει τα δεδομένα του, σε πραγματικό χρόνο, ανάλογα με το ποιες πηγές έχει περάσει εντός της πλατφόρμας. Δεν θα πρέπει να χρησιμοποιηθεί cache

Κωδικό Όνομα Απαίτησης
R13
Προσδιορισμός / Τίτλος Απαίτησης Βάση δεδομένων improve
Περιγραφή Απαίτησης <ul style="list-style-type: none"> • Η βάση δεδομένων του συστήματος, θα πρέπει να είναι σχεσιακή, για εύκολη υλοποίηση της εφαρμογής. • Θα πρέπει να κρατάει μόνο την απαραίτητη πληροφορία. • Θα πρέπει να υπάρχουν indexes για την βελτιστοποίηση της βάσης δεδομένων και την όσο γίνεται μεγαλύτερη ταχύτητά της.

4.3.4. Περιπτώσεις χρήσης και σενάρια Info Grabber Application

Σε αυτή την ενότητα θα αναλύσουμε όλες τις περιπτώσεις χρήσης που προκύπτουν από την παραπάνω περιγραφή του γενικού σεναρίου έτσι ώστε να προσδιορίσουμε τα σενάρια χρήσης του Info grabber application και να καταγράψουμε όλες τις περιπτώσεις αλληλεπίδρασης των χρηστών με το σύστημα.

Στο Table 2 βλέπουμε όλα τα σενάρια ανά περίπτωση χρήσης.

Συγκεκριμένα, για την περίπτωση χρήσης «Εγγραφή και ηλεκτρονική παρουσίαση» έχουμε τα εξής σενάρια:

- Εγγραφή μέλους στο eKoNEΣ
- Login στο eKoNEΣ
- Εγγραφή επιχειρηματικού εταίρου στο eKoNEΣ
- Διαχείριση προφίλ χρήστη
- Διαχείριση εταιρικού προφίλ
- Για την περίπτωση χρήσης «Διαχείριση περιεχομένου eKoNEΣ» έχουμε τα εξής σενάρια:
- Αναζήτηση επιχειρηματικών εταίρων ανά γειτονιά
- Διαχείριση ανακοινώσεων (message board)
- Ανίχνευση και εμφάνιση πακέτων βάση κατάστασης πακέτου (Active/Inactive)
- Λειτουργία ανεύρεσης περιεχομένου
- Εισαγωγή περιεχομένου από επιχειρηματικό εταίρο
- Επισκόπηση έργου eKoNEΣ (eKoNEΣ Project)
- Δημοσιοποίηση Εγγράφων/Εικόνων (Document/Image Publication)

Table 1: Λίστα σεναρίων ανά περίπτωση χρήσης

Περίπτωση χρήσης	Σενάρια
Εγγραφή και Είσοδος στο σύστημα	<ul style="list-style-type: none">• Εγγραφή μέλους στην εφαρμογή• Login στην εφαρμογή
Διαχείριση περιεχομένου	<ul style="list-style-type: none">• Δημιουργία πεδίων• Δημιουργία εικονικών πηγών (source instance)• Αποθήκευση πηγών στο σύστημα• Αντιστοίχιση των πεδίων με τα πεδία της εκάστοτε πηγής• Προβολή της πηγής για την αποφυγή λαθών• Parsing της πηγής• Δημιουργία κατηγοριών• Προβολή των αποτελεσμάτων• Σύγκριση των αποτελεσμάτων• Κατηγοριοποίηση των αποτελεσμάτων

4.3.4.1. Λίστα σεναρίων περίπτωσης χρήσης: Εγγραφή και Είσοδος στο σύστημα

Σε περαιτέρω ανάλυση, βρίσκουμε τις δράσεις για κάθε σενάριο της περίπτωσης χρήσης «Εγγραφή και ηλεκτρονική παρουσίαση» ξεχωριστά, όπως φαίνεται στο Table 3.

Table 2: Δράσεις σεναρίων περίπτωσης χρήσης «Εγγραφή και ηλεκτρονική παρουσίαση»

Σύστημα: info grabber application	
User: Πελάτης	
Περίπτωση χρήσης: Εγγραφή και Είσοδος στο σύστημα	
Σενάρια	Δράσεις
Εγγραφή μέλους στο info grabber application	Εισαγωγή υποχρεωτικών προσωπικών στοιχείων χρήστη
	<ul style="list-style-type: none">Εισαγωγή προαιρετικών προσωπικών στοιχείων χρήστη
	<ul style="list-style-type: none">Υποβολή φόρμας εγγραφής
	<ul style="list-style-type: none">Εισαγωγή κωδικού
	Υποβολή φόρμας
Login στο info grabber application	Εισαγωγή username και password
	Υποβολή αιτήματος

Σενάριο: Εγγραφή μέλους στο Info Grabber Application

Περίπτωση χρήσης: Εγγραφή και Είσοδος στο σύστημα

Σενάριο: Εγγραφή μέλους στο info grabber application

Το info grabber application είναι προσβάσιμο από όλους όσους επιθυμούν να χρησιμοποιήσουν τέτοιου ήδους εφαρμογή. Για να μπορεί ένας χρήστης να συμμετέχει ενεργά στην εφαρμογή θα πρέπει πρώτα να εγγραφεί. Κάνοντας εγγραφή γίνεται μέλος και μπορεί πλέον να χρησιμοποιήσει τις παρεχόμενες υπηρεσίες. Για να γίνει κάποιος μέλος θα πρέπει να εκτελέσει τα παρακάτω βήματα.

Βήματα σεναρίου:

Εισαγωγή υποχρεωτικών προσωπικών στοιχείων χρήστη

Εισαγωγή προαιρετικών υποχρεωτικών στοιχείων χρήστη

Υποβολή φόρμας εγγραφής

Εισαγωγή κωδικού

Υποβολή φόρμας

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα 1: Εισαγωγή υποχρεωτικών προσωπικών στοιχείων χρήστη

Τα υποχρεωτικά στοιχεία του χρήστη περιλαμβάνουν όνομα, επώνυμο, e-mail και είναι χαρακτηρισμένα ως υποχρεωτικά στη φόρμα εγγραφής

Βήμα2: Εισαγωγή προαιρετικών προσωπικών στοιχείων χρήστη

Επιπλέον των υποχρεωτικών στοιχείων ο χρήστης προτρέπεται από την διαδικασία εγγραφής να συμπληρώσει προαιρετικά στοιχεία.

Βήμα 5: Εισαγωγή κωδικού

Ο χρήστης θα πρέπει να εισαγάγει ένα μυστικό κωδικό με τον οποίο θα μπορεί να έχει πρόσβαση στην εφαρμογή

Βήμα 4: Υποβολή φόρμας εγγραφής

Αφού συμπληρώσει όλα τα στοιχεία ο χρήστης υποβάλει τη φόρμα εγγραφής

Σχηματική απεικόνιση σεναρίου:
Βήματα 1-4:



The image shows a registration form titled "Register to System". It contains six input fields with labels and instructions in Greek, and a "SignUp" button at the bottom.

Field Label	Instruction
Όνομα:	Το ονόμά σας
Επώνυμο:	Το επώνυμό σας
Username:	Το username σας
Password	Συνθηματικό
Confirm Password	Επιβεβαιώστε
E-mail:	Το e-mail σας

SignUp

Σενάριο: Login

Περίπτωση χρήσης: Εγγραφή και Είσοδος στο σύστημα

Σενάριο: Login

Η διαδικασία Login αφορά τους ήδη εγγεγραμμένους χρήστες του συστήματος, δηλαδή προϋποθέτει την επιτυχή εγγραφή ενός επισκέπτη με το σενάριο που περιγράφηκε νωρίτερα. Για να κάνει κάποιος login στο σύστημα μας θα πρέπει να εκτελέσει τα παρακάτω βήματα.

Βήματα σεναρίου:

Εισαγωγή username και password

Υποβολή αιτήματος

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα 1: Εισαγωγή username και password

Το username και το password αφορούν το username και τον κωδικό αντίστοιχα, τα οποία έχει δηλώσει ο χρήστης κατά τη διαδικασία εγγραφής του.

Βήμα2: Υποβολή αιτήματος

Αφού έχει εισάγει τα απαραίτητα στοιχεία, ο χρήστης υποβάλει το αίτημα εισαγωγής στο Σύστημα.

Σχηματική απεικόνιση σεναρίου:

Βήματα 1-2:

Info Grabber

ptixiaki.manosz.gr/login.php

Username:
manos

Password:
••••

Login

[Register](#)

Info Grabber Application - Web based
Πτυχιακή Εργασία - Ζαχαριουδάκης Εμμ.

4.3.4.2. Λίστα σεναρίων περίπτωσης χρήσης: Διαχείριση περιεχομένου

Κάθε σενάριο της περίπτωσης χρήσης «Διαχείριση περιεχομένου» το αναλύουμε σε δράσεις για να κατανοήσουμε τη λειτουργία του. Στο Table 4 βλέπουμε τα σενάρια και τις αντίστοιχες δράσεις.

Table 3: Δράσεις σεναρίων περίπτωσης χρήσης «Διαχείριση περιεχομένου Info Grabber Application»

Σύστημα: Info Grabber Application	
User: Διαχειριστής Εφαρμογής	
Περίπτωση Χρήσης: Διαχείριση περιεχομένου	
Σενάρια	Δράσεις
Δημιουργία πεδίων	New Field Βλέπουμε τα ήδη καταχωρημένα fields Κάνουμε νέο field
Δημιουργία εικονικών πηγών (source instance)	Επιλογή New Source Επιλογή Ονόματος για την πηγή Επιλογή τύπου για την πηγή Εάν η πηγή είναι xml αρχείο, ορίζουμε το link και το encode Κάνουμε Save
Αντιστοίχιση των πεδίων με τα πεδία της εκάστοτε πηγής	Connections Επιλογή πηγής Αντιστοιχούμε κάθε field με την αντίστοιχη θέση Αποθήκευση
Parsing της πηγής	Από το menu, επιλέγει Run Source Επιλέγει την πηγή που επιθυμεί. Η πηγή σαρώνεται από το σύστημα, και εμφανίζει το αντίστοιχο μήνυμα.
Δημιουργία κατηγοριών	Από το menu επιλέγουμε το “categories->new Category”. Εισάγουμε το όνομα και το description της κατηγορίας Επιλέγουμε εάν η κατηγορία είναι master ή SubCategory Εάν είναι sub category επιλέγουμε σε ποια κατηγορία θα είναι υποκατηγορία. Αποθήκευση
Προβολή των αποτελεσμάτων	Από το menu, επιλέγουμε την επιλογή See nodes. Μας εμφανίζονται όλα τα δεδομένα, με επιλογή για αναζήτηση οτιδήποτε ο χρήστης επιθυμεί.
Κατηγοριοποίηση των αποτελεσμάτων	Επιλογή από το menu το “live search” Ο χρήστης πληκτρολογεί οτιδήποτε επιθυμεί Στα αποτελέσματα, μπορεί να μεταβεί στην σελίδα του κάθε node ή στη σελίδα με όλα τα nodes κάθε πηγής Επιλέγοντας ένα Node, μπορεί να του ορίσει την κατηγορία του.

Dashboard	Πατώντας από οπουδήποτε βρισκόμαστε το λογότυπο της εφαρμογής, μεταβαίνουμε στην σελίδα Dashboard
	Η κεντρική σελίδα είναι μια γρήγορη ανασκόπηση της χρήσης της πλατφόρμας του εκάστοτε χρήστη
	Μπορεί να εκτελέσει όλα τα παραπάνω σενάρια από την προβολή αυτή

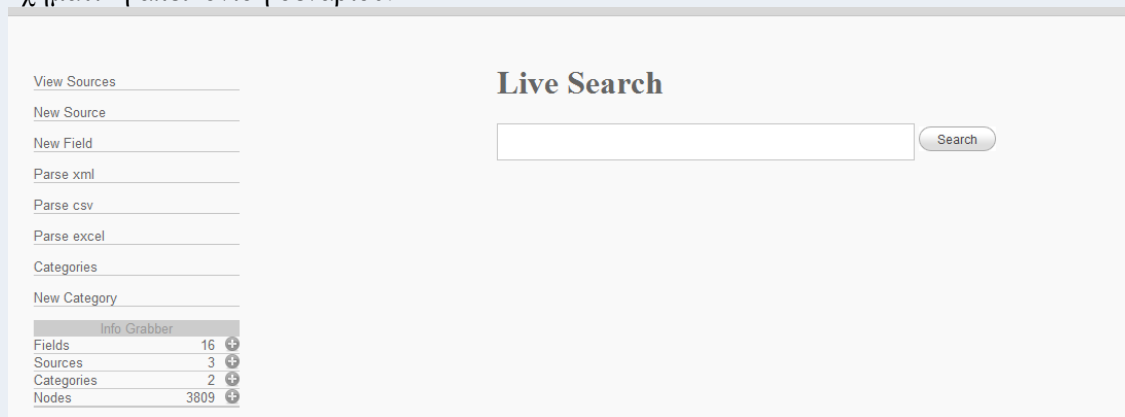
Σενάριο: Dashboard

Περίπτωση χρήσης: Διαχείριση περιεχομένου

Σενάριο: Dashboard

Στην εφαρμογή θα πρέπει να περιέχεται μια κεντρική σελίδα από την οποία ο χρήστης θα μπορεί να μετακινείται μέσα στην εφαρμογή, καθώς επίσης θα μπορεί να κάνει μια γρήγορη ανασκόπηση της χρήσης του, στην εφαρμογή.

Σχηματική απεικόνιση σεναρίου:



Σενάριο: Δημιουργία πεδίου

Περίπτωση χρήσης: Διαχείριση περιεχομένου

Σενάριο: Δημιουργία πεδίου

Ο χρήστης, μπορεί να φτιάξει όσα πεδία του είναι απαραίτητα για την υλοποίηση της εφαρμογής του. Για να γίνει αυτό θα πρέπει να ακολουθήσει τα παρακάτω βήματα.

Βήματα σεναρίου:

Από το dashboard ή το menu, πάμε στο New Field

Αν ο χρήστης έχει εισάγει κ άλλα fields μπορεί να τα δει, και πατάει Add Field

Ο χρήστης εισάγει ένα όνομα για το field που φτιάχνει και κάνει save

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα 1: Επιλογή New Field

Ο διαχειριστής του συστήματος επιλέγει από το dashboard ή την μπάρα του menu, το New Field

Βήμα 2: Add Field

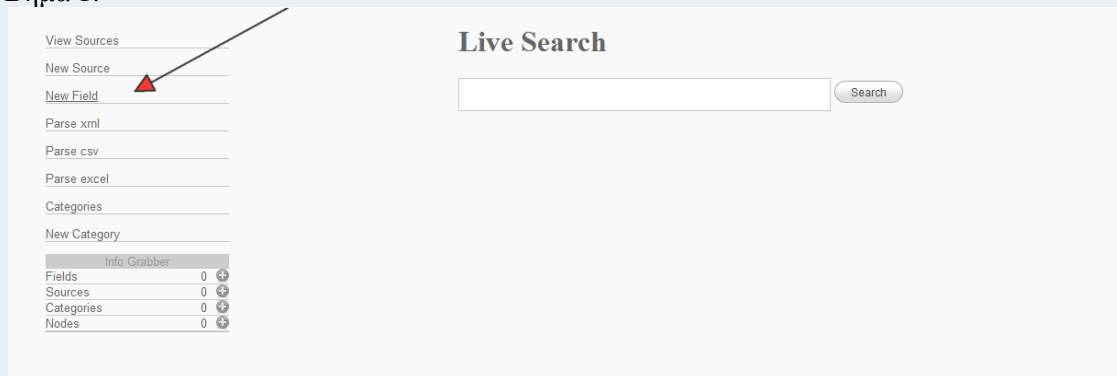
Το σύστημα ψάχνει να βρει όλα τα fields που ανήκουν στον χρήστη και τα παρουσιάζει. Για να προσθέσει νέο field ο χρήστης πατάει New field

Βήμα 3: Save Field

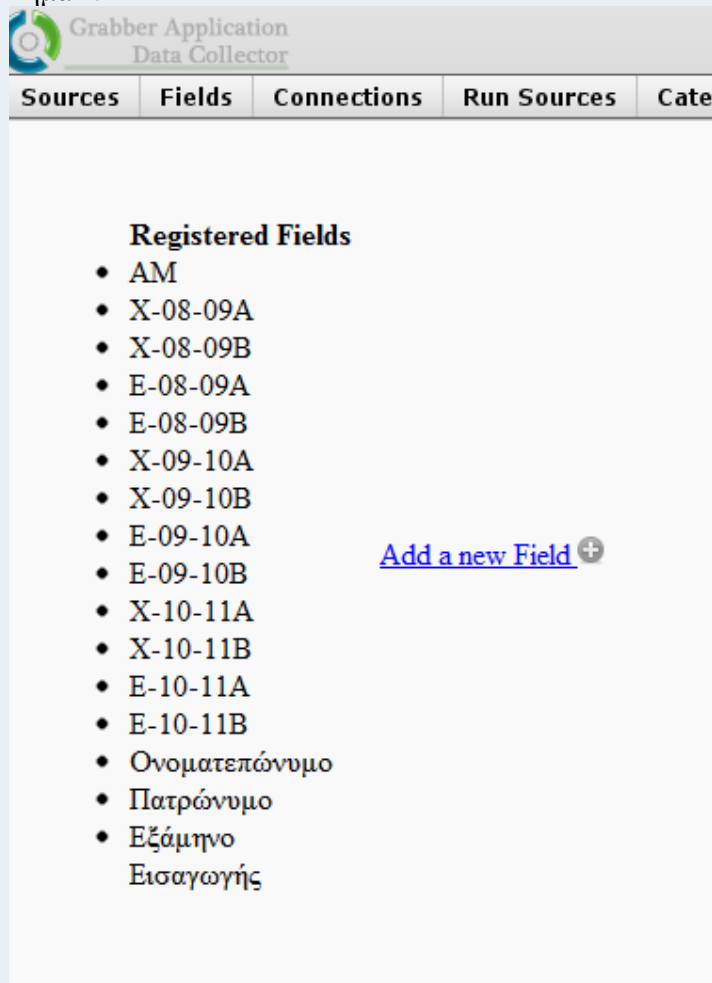
Αφού ο χρήστης εισάγει το όνομα του field κάνει save.

Σχηματική απεικόνιση σεναρίου:

Βήμα 1:



Βήμα 2:



Grabber Application
Data Collector

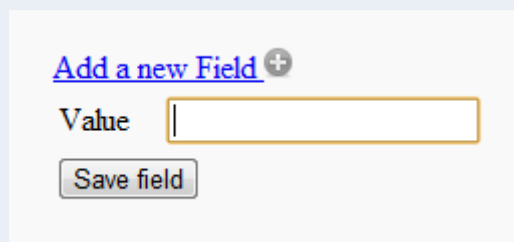
Sources Fields Connections Run Sources Cate

Registered Fields

- AM
- X-08-09A
- X-08-09B
- E-08-09A
- E-08-09B
- X-09-10A
- X-09-10B
- E-09-10A
- E-09-10B
- X-10-11A
- X-10-11B
- E-10-11A
- E-10-11B
- Ονοματεπώνυμο
- Πατρώνυμο
- Εξάμηνο
- Εισαγωγής

[Add a new Field](#) ⁺

Βήμα 3:



[Add a new Field](#) ⁺

Value

Σενάριο: Δημιουργία εικονικών πηγών (Source instance)

Περίπτωση χρήσης: Διαχείριση περιεχομένου

Σενάριο: Δημιουργία εικονικών πηγών (source instance)

Οι χρήστες της εφαρμογής μας θα πρέπει να φτιάξουν instance για κάθε πηγή από τις οποίες θα παίρνουν τα δεδομένα

Βήματα σεναρίου:

Από το menu ή το Dashboard κάνουμε new Source

Επιλέγουμε το όνομα και τον τύπο της πηγής

Κάνουμε Save

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα 1: Επιλογή New Source

Ο χρήστης επιλέγει από το μενού ή τον dashboard την επιλογή New Source

Βήμα2: Εισαγωγή Ονόματος

Καθορίζει ένα όνομα για την νέα πηγή που πρόκειται να φτιάξει

Βήμα 3: Καθορισμός τύπου

Καθορίζει το τύπο της.

Βήμα 4: Εισαγωγή επιπλέον χαρακτηριστικών

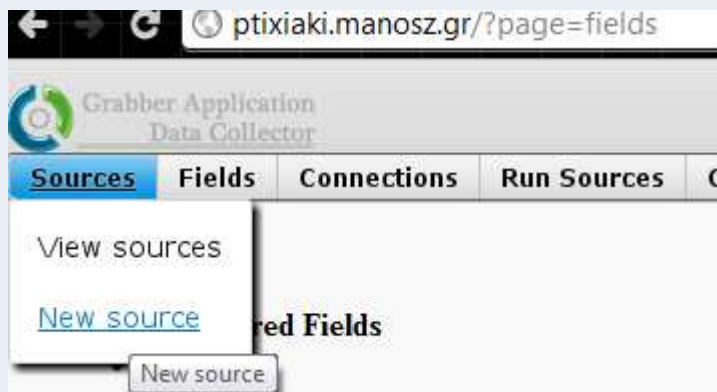
Πιθανόν να χρειαστεί να ορίσει απόμακρυσμένο υπολογιστή στον οποίο βρίσκεται η πηγή

Βήμα 5: Αποθήκευση

Πατάει την επιλογή SAVE

Σχηματική απεικόνιση σεναρίου:

Βήμα 1:



Βήματα 2:

Source Details

Όνομα

Τύπος
csv file
excel file
xml file
digital source

Βήμα 3:

Details for xml

Link

Encode

Βήμα 4:

<p>Source Details</p> <p>Όνομα <input type="text" value="Xml file"/></p> <p>Τύπος <input type="text" value="xml file"/></p> <p><input type="button" value="SAVE"/></p>	<p>Details for xml</p> <p>Link <input type="text" value="http://sports.pathfinder.gr/rss.php"/></p> <p>Encode <input type="text" value="UTF-8"/></p>
---	---

Σενάριο: Αντιστοίχιση των πεδίων με τα πεδία της εκάστοτε πηγής

Περίπτωση χρήσης: Διαχείριση περιεχομένου

Σενάριο: Αντιστοίχιση των πεδίων με τα πεδία της εκάστοτε πηγής

Η κάθε πηγή περιέχει την χρήσιμη πληροφορία, μέσα της, σε διάταξη τέτοια, η οποία είναι ακαθόριστη, και διαφέρει από πηγή σε πηγή. Ο χρήστης θα πρέπει να αντιστοιχήσει τα πεδία του, με την αντίστοιχη θέση, της πληροφορίας, μέσα στην εκάστοτε πηγή.

Βήματα σεναρίου:

Connections

Επιλογή πηγής

Αντιστοίχιση κάθε field με την αντίστοιχη θέση

Αποθήκευση

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα 1: Connections

Από το menu, πατάμε την επιλογή connections

Βήμα 2: Επιλογή πηγής

Στην εφαρμογή μας εμφανίζονται όλες οι διαθέσιμες πηγές, ανάλογα με τον τύπο τους.

Επιλέγοντας την πηγή που επιθυμεί, μεταβαίνει στη σελίδα που μπορεί να κάνει την αντιστοίχιση.

Βήμα 3: Αντιστοίχιση κάθε field με την αντίστοιχη θέση

Ο χρήστης κάνει την αντιστοίχιση, πεδία – θέση στην πηγή που προσφέρεται η αντίστοιχη πληροφορία.

Βήμα 4: Αποθήκευση

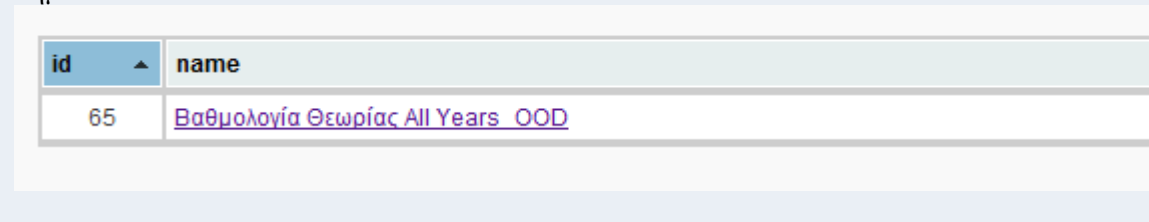
Ο χρήστης πατάει αποθήκευση

Σχηματική απεικόνιση σεναρίου:

Βήμα 1:



Βήμα 2:



Βήμα 3,4:

ΒΑΘΜΟΛΟΓΙΑ Diaxirisi ergon pliroforikis

E-08-09A	
X-08-09A	
X-08-09B	
Όνοματεπώνυμο	Name
AM	AM
E-08-09B	
X-09-10A	
X-09-10B	
E-09-10A	JuneTelikos
E-09-10B	SepTelikos
X-10-11A	
Πατρώνυμο	Father
X-10-11B	
E-10-11A	
E-10-11B	
Εξάμηνο Εισαγωγής	

Path to nodes:

```
SimpleXMLElement Object
(
    [attributes] => Array
        (
            [version] => 2.0
        )
    [channel] => SimpleXMLElement Object
        (
            [item] => Array
                (
                    [0] => SimpleXMLElement Object
                        (
                            [AM] => 193
                            [Name] => Γαλόπουλος Βασίλειος
                            [Father] => Κυριάκου
                            [xmino] => ΠΙΙΕ
                            [JuneExetasi] => 4,20
                        )
                )
            )
)
```


Σενάριο: Parsing της πηγής

Περίπτωση χρήσης: Διαχείριση περιεχομένου

Σενάριο: Parsing της πηγής

Ο χρήστης αφού υλοποιήσει τα παραπάνω βήματα τουλάχιστον για μία πηγή θα πρέπει να κάνει parsing, δηλαδή να διαβάσει την πηγή και να αποθηκεύσει την αντίστοιχη πληροφορία εντός του συστήματός μας.

Βήματα σεναρίου:

Από το menu, επιλέγει Run Source

Επιλέγει την πηγή που επιθυμεί.

Η πηγή σαρώνεται από το σύστημα, και εμφανίζει το αντίστοιχο μήνυμα

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα 1: Από το menu, επιλέγει Run Source

Ο χρήστης θα πρέπει να μεταβεί στην σελίδα για να κάνει parsing της κάθε πηγής. Για να μεταβεί στην αντίστοιχη θέση θα πρέπει να επιλέξει από το menu την επιλογή Run Source και να επιλέξει τον αντίστοιχο τύπο της πηγής.

Βήμα 2: Επιλογή της πηγής που επιθυμεί

Το σύστημα του εμφανίζει όλες τις πηγές που έχουν τύπο ίδιο με αυτό που επέλεξε στο Βήμα1. Ο χρήστης είναι ένα βήμα πριν από το Parsing. Θα πρέπει να γνωρίζει ότι ανάλογα με το πλήθος των πληροφοριών, το σύστημα θα του απαγορεύσει την οποιαδήποτε άλλη λειτουργία, για αντίστοιχο χρόνο.

Βήμα 3: Εμφάνιση αντίστοιχου μηνύματος

Καθώς το σύστημα σαρώνει την πηγή και αποθηκεύει την πληροφορία, ο χρήστης δεν μπορεί να χρησιμοποιήσει την εφαρμογή για οτιδήποτε άλλο. Αφου σαρωθεί η πηγή, εμφανίζεται στο χρήστη το αντίστοιχο μήνυμα.

Σχηματική απεικόνιση σεναρίου:

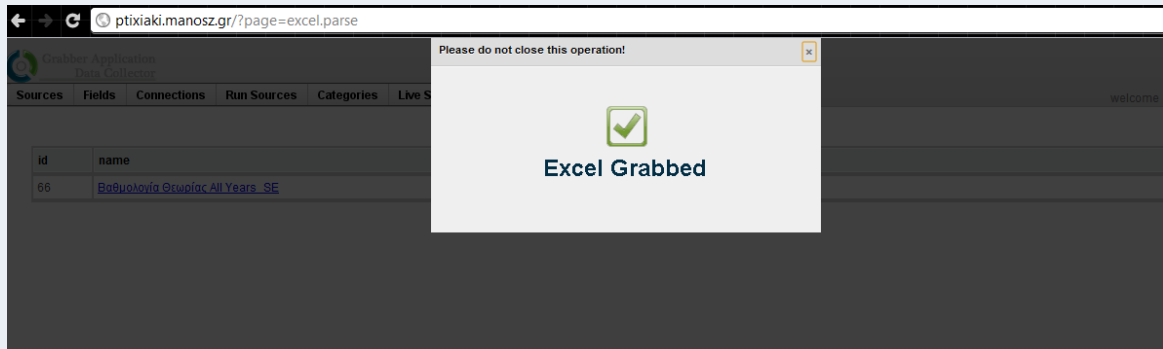
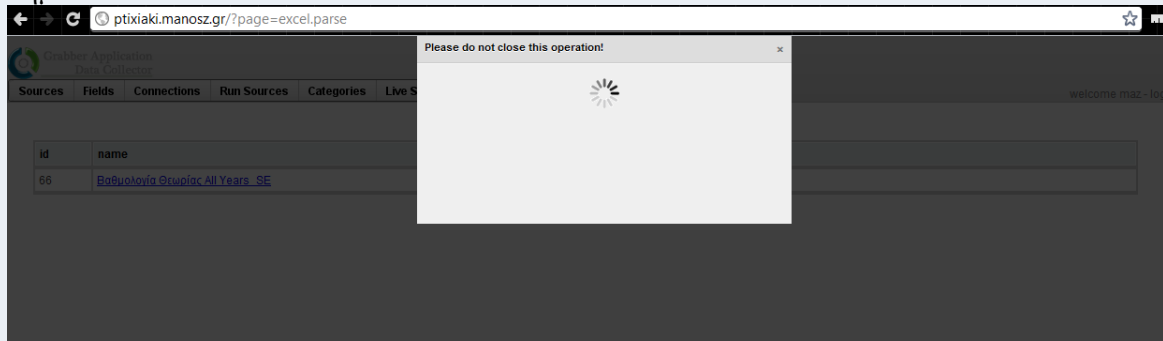
Βήμα 1:



Βήμα 2:

id	name
66	Βαθμολογία Θεωρίας All Years_SE

Βήμα 3:



Σενάριο: Δημιουργία κατηγοριών

Περίπτωση χρήσης: Διαχείριση περιεχομένου

Σενάριο: Δημιουργία κατηγοριών

Οι χρήστες της εφαρμογής έχουν την δυνατότητα να κατηγοριοποιήσουν τα δεδομένα, για εύκολη αναζήτησή τους μετά. Μπορούν δηλαδή να τοποθετήσουν σε ομάδες τα δεδομένα. Τις ομάδες αυτές οι χρήστες τις φτιάχνουν όπως αυτοί επιθυμούν.

Βήματα σεναρίου:

Μετάβαση στη σελίδα δημιουργίας νέας κατηγορίας

Εισάγουμε το όνομα και το description της κατηγορίας

Επιλέγουμε εάν η κατηγορία είναι master ή SubCategory

Εάν είναι sub category επιλέγουμε σε ποια κατηγορία θα είναι υποκατηγορία.

Αποθήκευση

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα 1: Από Μετάβαση στη σελίδα δημιουργίας νέας κατηγορίας

Ο χρήστης από το menu επιλέγει το Categories -> new Category

Βήμα 2: Εισαγωγή ονόματος και περιγραφής για την κατηγορία

Εισάγει ένα όνομα και μια σύντομη περιγραφή για την κατηγορία που πρόκειται να αποθηκεύσει.

Βήμα 3: Επιλογή Τύπου κατηγορίας

Ο χρήστης θα πρέπει να επιλέξει εάν η κατηγορία είναι κύρια ή υποκατηγορία κάποιας άλλης κατηγορίας.

Βήμα 4: Αποθήκευση

Ο χρήστης επιλέγει αποθήκευση

Σχηματική απεικόνιση σεναρίου:

Βήμα 1:



Βήμα 2-3-4:

The screenshot shows a form titled 'Category'. It has three input fields: 'name' with the value 'New Category', 'description' with the value 'Description for this', and a dropdown menu for 'Master Category' with 'Master Category' selected. A 'save' button is located at the bottom right of the form.

Σενάριο: Δημιουργία κατηγοριών

Περίπτωση χρήσης: Διαχείριση περιεχομένου

Σενάριο: Δημιουργία κατηγοριών

Οι χρήστες της εφαρμογής έχουν την δυνατότητα να κατηγοριοποιήσουν τα δεδομένα, για εύκολη αναζήτησή τους μετά. Μπορούν δηλαδή να τοποθετήσουν σε ομάδες τα δεδομένα. Τις ομάδες αυτές οι χρήστες τις φτιάχνουν όπως αυτοί επιθυμούν.

Βήματα σεναρίου:

Μετάβαση στη σελίδα δημιουργίας νέας κατηγορίας

Εισάγουμε το όνομα και το description της κατηγορίας

Επιλέγουμε εάν η κατηγορία είναι master ή SubCategory

Εάν είναι sub category επιλέγουμε σε ποια κατηγορία θα είναι υποκατηγορία.

Αποθήκευση

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα 1: Από Μετάβαση στη σελίδα δημιουργίας νέας κατηγορίας

Ο χρήστης από το menu επιλέγει το Categories -> new Category

Βήμα 2: Εισαγωγή ονόματος και περιγραφής για την κατηγορία

Εισάγει ένα όνομα και μια σύντομη περιγραφή για την κατηγορία που πρόκειται να αποθηκεύσει.

Βήμα 3: Επιλογή Τύπου κατηγορίας

Ο χρήστης θα πρέπει να επιλέξει εάν η κατηγορία είναι κύρια ή υποκατηγορία κάποιας άλλης κατηγορίας.

Βήμα 4: Αποθήκευση

Ο χρήστης επιλέγει αποθήκευση

Σχηματική απεικόνιση σεναρίου:

Βήμα 1:



Βήμα 2-3-4:

The image shows a form titled 'Category'. It contains three input fields: 'name' with the value 'New Category', 'description' with the value 'Description for this', and a dropdown menu for 'Master Category'. The dropdown menu is open, showing two options: 'Master Category' (highlighted in blue) and 'Subcategory'. A 'save' button is located at the bottom right of the form.

Σενάριο: Προβολή των αποτελεσμάτων

Περίπτωση χρήσης: Διαχείριση περιεχομένου

Σενάριο: Προβολή των αποτελεσμάτων

Ο κάθε χρήστης μπορεί να βλέπει τα δεδομένα του τακτοποιημένα από την επιλογή see nodes. Με αυτή την επιλογή εμφανίζονται στον χρήστη όλα τα δεδομένα του, σε στήλες, χωρισμένα ανα fields, και ο χρήστης μπορεί να κάνει διάφορες διατάξεις που επιθυμεί

Βήματα σεναρίου:
 Από το menu, επιλέγουμε την επιλογή See nodes.
 Μας εμφανίζονται όλα τα δεδομένα, με επιλογή για αναζήτηση οτιδήποτε ο χρήστης επιθυμεί.
Αναλυτική περιγραφή βημάτων σεναρίου:
Βήμα 1: Από το menu, επιλέγουμε την επιλογή See nodes
 Από το menu, ο χρήστης μπορεί να επιλέξει το See nodes
Βήμα 2: Εμφάνιση εικονικού χώρου της κοινότητας του eKoNES Project
 Στο χρήστη εμφανίζονται όλα τα τα δεδομένα, μπορεί να ταξινομήσει ανάλογα με τα fields ή την κατηγορία στην οποία ανήκουν.
Σχηματική απεικόνιση σεναρίου:



Βήμα 2:

Κόδος	Source name	Category	E-06-00A	X-06-00A	X-08-00B	Όνοματεπώνυμο	AM	E-08-00B	X-09-00A	X-09-00E	E-09-00A	E-09-00B	X-09-00A	X-09-00B	Πατρώνυμο	X-10-00E	E-10-00A	E-10-00B	Γύλινο (κατηγορία)
2	Επιθεωρήσεις Θεωρίας, All Nodes (000)	00_students	-	-	-	Μανωλ Αλέητρο	1	-	-	-	-	-	-	-	Γαωογίου	-	-	-	00-00E
3	Επιθεωρήσεις Θεωρίας, All Nodes (000)	-	-	-	-	Κωνσταντίνος Σωτήρης	2	-	-	-	-	-	-	-	Γαωογίου	-	-	-	00-00E
4	Επιθεωρήσεις Θεωρίας, All Nodes (000)	-	-	-	-	Πατριάρχης Νικόλαος	3	-	-	-	-	-	-	-	Δημητρίου	-	-	-	00-00X
5	Επιθεωρήσεις Θεωρίας, All Nodes (000)	00_students	-	-	-	Ταγιλς Δημήτριος	4	-	Absent	Absent	-	-	-	-	Εμμενοειλ	-	-	-	00-00E
6	Επιθεωρήσεις Θεωρίας, All Nodes (000)	-	-	-	-	Σπρινοπούλου Ανίτη	5	-	-	-	-	-	-	-	Δημητρίου	-	-	-	00-00X
7	Επιθεωρήσεις Θεωρίας, All Nodes (000)	-	-	-	-	Θεοδοσίδης Ειλιγγιλιος	6	-	-	-	-	-	-	-	Σωτήρη	-	-	-	00-00X
8	Επιθεωρήσεις Θεωρίας, All Nodes (000)	New Students	-	-	-	Σοφίης Θωμάς	7	-	-	-	-	-	-	-	Αλιάντρου	-	-	-	00-00X
9	Επιθεωρήσεις Θεωρίας, All Nodes (000)	-	-	-	-	Τιμόθεος Δημήτριος	8	-	-	-	-	-	-	-	Γαωογίου	-	-	-	00-00X
10	Επιθεωρήσεις Θεωρίας, All Nodes (000)	-	-	-	-	Τσιλινοπούλιος Αναστάσιος	9	-	-	-	-	-	-	-	Σωτήρη	-	-	-	00-00X
11	Επιθεωρήσεις Θεωρίας, All Nodes (000)	-	-	-	-	Κυριακίδης Γαωογιος	10	-	-	-	-	-	-	-	Εμμενοειλ	-	-	-	00-00X

Σενάριο: Κατηγοριοποίηση των αποτελεσμάτων

Περίπτωση χρήσης: Διαχείριση περιεχομένου

Σενάριο: Κατηγοριοποίηση των αποτελεσμάτων

Ο χρήστης, αφού έχει φτιάξει τις κατηγορίες τις οποίες επιθυμεί, μπορεί να κατηγοριοποιήσει τα δεδομένα του, όπως επιθυμεί. Κάθε ολόκληρο set από τιμές που αντιστοιχούν στα fields ονομάζεται node και κάθε node μπορεί να ανήκει σε μία και μόνο κατηγορία.

Βήματα σεναρίου:

Επιλογή από το menu το “live search”

Ο χρήστης πληκτρολογεί οτιδήποτε επιθυμεί

Στα αποτελέσματα, μπορεί να μεταβεί στην σελίδα του κάθε node ή στη σελίδα με όλα τα nodes κάθε πηγής

Επιλέγοντας ένα Node, μπορεί να του ορίσει την κατηγορία του.

Αναλυτική περιγραφή βημάτων σεναρίου:

Βήμα 1: Επιλογή από το menu το “live search”

Βήμα 2: Εισαγωγή τιμής για αναζήτηση

Ο χρήστης είναι πλέον σε θέση να κάνει την αναζήτησή του.

Βήμα 3: Μετάβαση στα Nodes

Αφού γίνει η αναζήτηση, ο χρήστης μπορεί να μεταβεί στην επεξεργασία των nodes, για να κατηγοριοποιήσει το κάθε ένα.

Η κατηγοριοποίηση μπορεί να γίνει και από την σελίδα που βλέπουμε όλα τα nodes που ανήκουν σε μια πηγή.

Σχηματική απεικόνιση:

Βήμα 1:

Sources	Fields	Connections	Run Sources	Categories	Live Search	See nodes
---------	--------	-------------	-------------	------------	-------------	-----------

Live Search

Σχηματική απεικόνιση σεναρίου:
Βήμα 2:

Live Search

Ζαχαριουδάκης

6 nodes

Ζαχαριουδάκης Εμμανουήλ

Source: Βαθμολογία Θεωρίας All Years_OOD

Ζαχαριουδάκης Διονύσιος

Source: Βαθμολογία Θεωρίας All Years_OOD

Ζαχαριουδάκης Εμμανουήλ

Source: Βαθμολογία Θεωρίας All Years_OOD

Ζαχαριουδάκης Νικόλαος

Source: Βαθμολογία Θεωρίας All Years_OOD

Ζαχαριουδάκης Ιωάννης

Source: Βαθμολογία Θεωρίας All Years_OOD

ZACHARIOYDAKHIS

Source: Βαθμολογία Θεωρίας All Years_SE

Βήμα 3:

Live Search

Ζαχαριουδάκης

6 nodes

Ζαχαριουδάκης Εμμανουήλ

Source: Βαθμολογία Θεωρίας All Years_OOD

Ζαχαριουδάκης Διονύσιος

Source: Βαθμολογία Θεωρίας All Years_OOD

Ζαχαριουδάκης Εμμανουήλ

Source: Βαθμολογία Θεωρίας All Years_OOD

Ζαχαριουδάκης Νικόλαος

Source: Βαθμολογία Θεωρίας All Years_OOD

Μετάβαση στην σελίδα για το συγκεκριμένο node

Μετάβαση στη σελίδα με όλα τα nodes της συγκεκριμένης πηγής

E-08-09A	
X-08-09A	
X-08-09B	
Όνοματεπώνυμο	Ζαχαριουδάκης Εμμανουήλ
ΑΜ	1567
E-08-09B	
X-09-10A	
X-09-10B	
E-09-10A	
E-09-10B	
X-10-11A	
Πατρώνυμο	Αριστείδη
X-10-11B	
E-10-11A	
E-10-11B	
Εξάμηνο Εισαγωγής	04-05X

▼

- Old_students
- New Students

Ο χρήστης μπορεί να μεταβεί και σε όλα τα nodes της συγκεκριμένης πηγής.

Set Category

E-08-09A:
X-08-09A:
X-08-09B:
Όνοματεπώνυμο: Μανιού Δήμητρα
ΑΜ: 1
E-08-09B:
X-09-10A:
X-09-10B:
E-09-10A:
E-09-10B:
X-10-11A:
Πατρώνυμο: Γεωργίου
X-10-11B:
E-10-11A:
E-10-11B:
Εξάμηνο 99-00E
Εισαγωγής:

4.3.5. Data model Info Grabber Application

Στον παρακάτω πίνακα παρατίθενται τα απαραίτητα δεδομένα ανά σενάριο για τις περιπτώσεις χρήσης «Εγγραφή και χρησιμοποίηση του συστήματος» καθώς και οι πίνακες των βάσεων δεδομένων που χρειάζονται.

Table 4: Necessary Data και Tables ανά σενάριο

Scenarios	Necessary data	Tables
Εγγραφή μέλους στο Info Grabber Application	Username, Password, lastname, name, email.	users
Login στο Info Grabber Application	Username, Password	Users
Δημιουργία Fields	User_id, name, visible, linkable	Fields
Δημιουργία Source	User_id, name, link, encode,type, template	Sources
Αντιστοιχήσεις	Source_id, field_id,position,tag,starttag,	Csvdetails,exceldetails,xmldetails,Sources
Κατηγοριοποίηση	Node_id, cat_id	NodesBelong
Τα κύρια δεδομένα μας	Node_id,source_id, field_id,value	Data

category

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
<u>id</u>	int(11)	Όχι	
user_id	int(11)	Όχι	
name	text	Όχι	
description	text	Όχι	
parent_id	int(11)	Όχι	
is_sub	int(11)	Όχι	

Εικόνα 9:Data Base Screenshot

csvdetails

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
<u>id</u>	int(11)	Όχι	
source_id	int(11)	Όχι	
field_id	int(11)	Όχι	
position	int(11)	Όχι	

data

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
<u>id</u>	int(11)	Όχι	
node_id	int(11)	Όχι	
source_id	int(11)	Όχι	
field_id	int(11)	Όχι	
value	text	Όχι	

exceldetails

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
<u>id</u>	int(11)	Όχι	
source_id	int(11)	Όχι	
field_id	int(11)	Όχι	
position	text	Όχι	

fields

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
<u>id</u>	int(11)	Όχι	
userid	int(11)	Όχι	
name	text	Όχι	
visible	int(11)	Όχι	
linkable	int(11)	Όχι	
type	text	Όχι	

nodesbelong

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
<u>node_id</u>	int(11)	Όχι	
<u>cat_id</u>	int(11)	Όχι	

sources

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
<u>id</u>	int(11)	Όχι	
userid	int(11)	Όχι	
name	text	Όχι	
link	text	Όχι	
encode	text	Όχι	
type	text	Όχι	
template	text	Όχι	

dbdetails

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
<u>id</u>	int(11)	Όχι	
<u>source_id</u>	int(11)	Όχι	
name	text	Όχι	
host	text	Όχι	
username	text	Όχι	
password	text	Όχι	

excel

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
<u>id</u>	int(11)	Όχι	
<u>source_id</u>	int(11)	Όχι	
layout	text	Όχι	
start	text	Όχι	
end	text	Όχι	

users

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
<u>id</u>	int(11)	Όχι	
name	text	Όχι	
lastname	text	Όχι	
password	text	Όχι	
email	text	Όχι	
permissions	int(11)	Όχι	
username	text	Όχι	

xmldetails

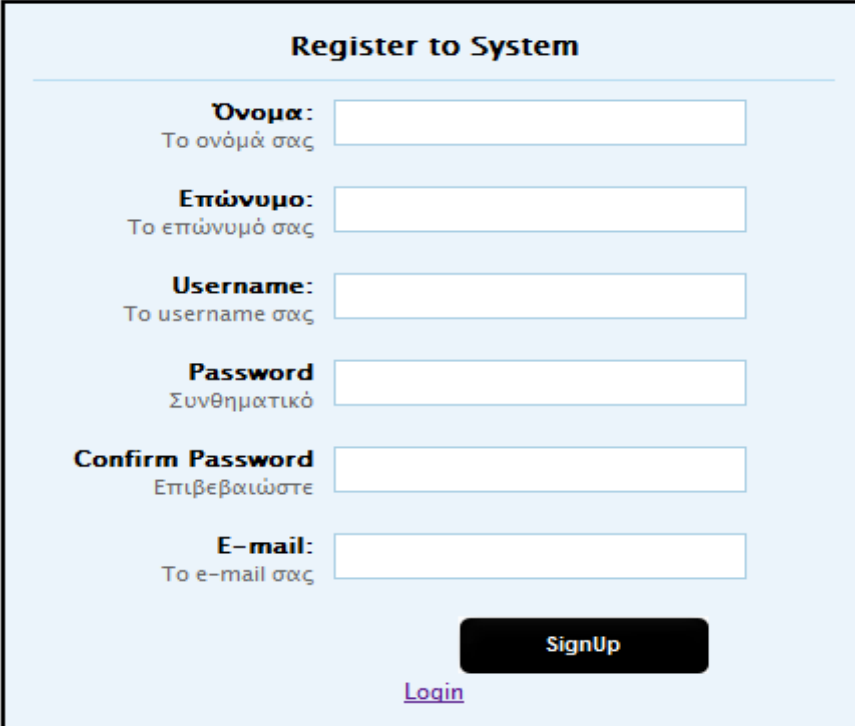
Πεδίο	Τύπος	Κενό	Προκαθορισμένο
<u>id</u>	int(11)	Όχι	
<u>source_id</u>	int(11)	Όχι	
<u>field_id</u>	int(11)	Όχι	
tag	text	Όχι	
starttag	tinyint(1)	Όχι	

5. Διεπαφή Info Grabber Application

Η διεπαφή ενός συστήματος παίζει πολύ σημαντικό ρόλο στη λειτουργικότητα αλλά και στην παρουσίασή του. Το Info Grabber Application καλείται να αναπτύξει μια τέτοια διεπαφή ώστε να μπορεί να υπάρχει λειτουργικότητα στο σύστημα, αλλά και να είναι εύχρηστο προς τους χρήστες.

Όσο αναφορά τα στοιχεία εκείνα της διεπαφής που προσδίδουν λειτουργικότητα στο σύστημα θα αναλύσουμε τη διεπαφή κάθε σεναρίου ξεχωριστά, όπως αυτή προκύπτει από τις απαιτήσεις του συστήματος και από τα εκάστοτε σενάρια χρήσης:

- Εγγραφή μέλους στο Info Grabber Application: Για την εγγραφή μέλους στο *Info Grabber Application* απαιτείται μια φόρμα εγγραφής με πεδία εισαγωγής κειμένου για την συμπλήρωση των προσωπικών στοιχείων του χρήστη (όνομα, επώνυμο, κ.τ.λ.), και ένα submit button για την υποβολή των παραπάνω στοιχείων. Σε περίπτωση που ο χρήστης δεν συμπληρώσει κάποιο υποχρεωτικό πεδίο, όπως το όνομα, το σύστημα τον ενημερώνει με μήνυμα λάθους και τον προτρέπει να συμπληρώσει σωστά το πεδίο για να προχωρήσει στην αποθήκευση των στοιχείων.



The image shows a registration form titled "Register to System". It contains the following fields and labels:

- Όνομα:** Το ονόμά σας
- Επώνυμο:** Το επώνυμό σας
- Username:** Το username σας
- Password:** Συνθηματικό
- Confirm Password:** Επιβεβαιώστε
- E-mail:** Το e-mail σας

At the bottom of the form, there is a black "SignUp" button and a blue "Login" link.

Εικόνα 10: Register

- Login στο Info Grabber Application: Η διαδικασία login απαιτεί το Username και τον κωδικό του χρήστη, γι' αυτό, όπως φαίνεται και στο παρακάτω σχήμα, χρειάζονται δύο πεδία εισαγωγής κειμένου (textfields) για την εισαγωγή των στοιχείων αυτών. Τέλος, το submit button είναι απαραίτητο για την υποβολή του αιτήματος εισαγωγή στην εφαρμογή.

A login form with two input fields: 'Username:' and 'Password:'. Below the password field is a dark 'Login' button. At the bottom left of the form area is a blue 'Register' link.

Εικόνα 11:Login

- *Dashboard:* Η διεπαφή προσφέρει την προβολή αρχικής σελίδας της εφαρμογής, από την οποία ο χρήστης μπορεί να δει συνοπτικά τα δεδομένα του, και τις ρυθμίσεις του. Επίσης του προσφέρεται ένα menu, για γρήγορη πρόσβαση στις υποσελίδες της εφαρμογής, και την γρήγορη πλοήγηση του.

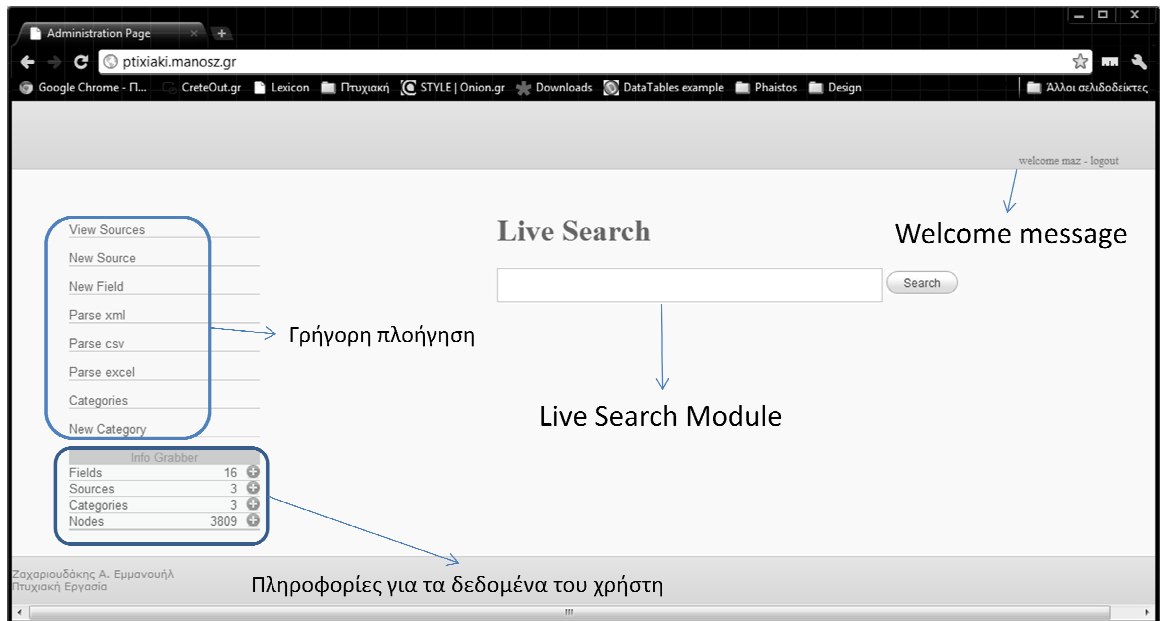
The dashboard features a top navigation bar with 'welcome maz - logout'. On the left is a menu with items: View Sources, New Source, New Field, Parse xml, Parse csv, Parse excel, Categories, and New Category. In the center is a 'Live Search' section with a search input field and a 'Search' button. On the right is an 'Info Grabber' table:

Info Grabber	
Fields	16
Sources	3
Categories	2
Nodes	3809

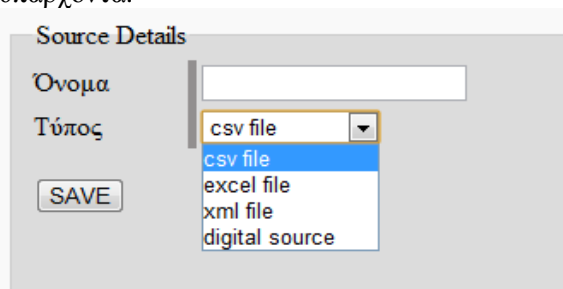
At the bottom left, it says 'Ζαχαριουδάκης Α. Εμμανουήλ Πτυχιακή Εργασία'.

Εικόνα 12:Login

Επίσης από το Dashboard, μπορεί να κάνει μια γρήγορη αναζήτηση με το Module Live Search, στα αποθηκευμένα σύνολα δεδομένων του.

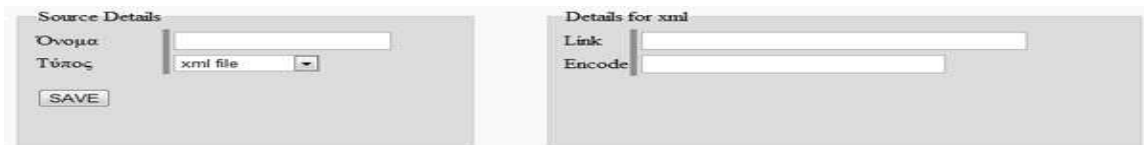


- *New Source*: Ο χρήστης έχει την επιλογή να αποθηκεύσει – δημιουργήσει ένα νέο instance της πηγής από την οποία θα σαρώσει τα δεδομένα για την σύζευξη τους, με τα ήδη υπάρχοντα.



Εικόνα 13: New Source

Βλέπουμε ότι από ένα drop down μενού, ο χρήστης μπορεί να επιλέξει τον τύπο της πηγής, καθώς μπορεί να ορίσει και το αντιπροσωπευτικό όνομα της. Με το κουμπί SAVE ο χρήστης, αποθηκεύει το instance της πηγής.



Στο παραπάνω screenshot βλέπουμε ότι ο χρήστης μπορεί να εισάγει το link στο οποίο βρίσκεται η απομακρυσμένη πηγή και την κωδικοποίηση των χαρακτήρων της.

- *View Sources*: Ο χρήστης μπορεί να βλέπει όλες τις πηγές που έχει δημιουργήσει, και να πλοηγηθεί μέσα σε αυτές, καθώς επίσης και να μεταβάλλει τα χαρακτηριστικά της κάθε μίας.

id	name	type
65	Βαθμολογία Θεωρίας All Years_OOD	csv
66	Βαθμολογία Θεωρίας All Years_SE	excel
67	ΒΑΘΜΟΛΟΓΙΑ Diaxirisi ergon pliroforikis	xml

Εικόνα 14:View sources

- *Add/View Fields*: Ο χρήστης της εφαρμογής μπορεί να ορίζει τα πεδία που θέλει να χρησιμοποιήσει, και να τα τροποποιεί. Παρακάτω βλέπετε μια εικονογράφηση της διαδικασίας.

Registered Fields

- AM
- X-08-09A
- X-08-09B
- E-08-09A
- E-08-09B
- X-09-10A
- X-09-10B
- E-09-10A
- E-09-10B
- X-10-11A
- X-10-11B
- E-10-11A
- E-10-11B
- Ονοματεπώνυμο
- Πατρώνυμο
- Εξάμηνο

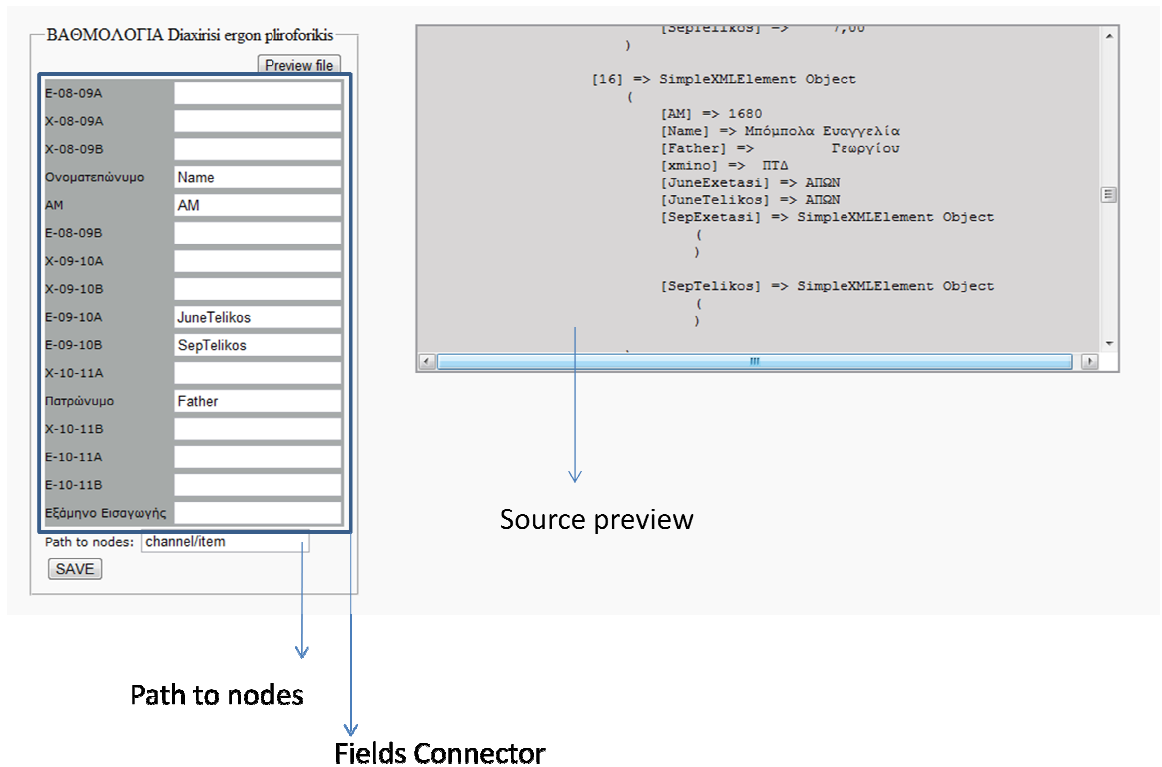
Εισαγωγής

[Add a new Field](#) ⁺

Value

Εικόνα 15:New Field

- *Source Connections & Preview*: Ο χρήστης, θα πρέπει να αντιστοιχίσει τα fields, με την ανάλογη θέση της εκάστοτε πηγής. Στο παράδειγμα μας, βλέπουμε ότι το Field «ονοματεπώνυμο» ο χρήστης το αντιστόχησε με το Keyword “name”. Συγκεκριμένα, έχουμε ένα xml αρχείο, στο οποίο το δεδομένο που αντιστοιχεί στο ονοματεπώνυμο βρίσκεται μέσα στα tags <name>, σύμφωνα με την θεωρία των xml και τις προδιαγραφές τις εφαρμογής μας.



Εικόνα 16:Source Connections

- Categories:** Ο χρήστης μπορεί να ομαδοποιήσει τα nodes που έχει αποθηκεύσει στην εφαρμογή, και να τα κατηγοριοποιήσει όπως αυτός επιθυμεί. Έχει την δυνατότητα να φτιάξει απεριόριστες κατηγορίες και υποκατηγορίες. Η δημιουργία και η επεξεργασία αυτών γίνεται μέσα από μια πολύ απλή διαδικασία, όπως φαίνεται παρακάτω:

The form is titled "Category" and contains the following fields:

- name:** Text input field containing "Test Category".
- description:** Text input field containing "Desc for SEO".
- Subcategory:** A dropdown menu currently showing "Subcategory".
- Parent:** A list box containing "Old_students" and "New Students".
- save:** A button at the bottom right.

Εικόνα 17:New Category

Ο χρήστης ορίζει το όνομα της κατηγορίας, μια σύντομη περιγραφή (για Search Engine Optimization κυρίως), ορίζει εάν είναι υποκατηγορία κάποιας κατηγορίας και αν είναι Subcategory, ορίζει ποια κατηγορία θα είναι ο γονέας της. Σαν γονέα της κατηγορίας δεν μπορούμε να ορίσουμε κάποια κατηγορία η οποία βρίσκεται σε χαμηλότερο επίπεδο από αυτήν, και ανήκουν στο ίδιο κλαδί του γράφου. Για παράδειγμα, δεν μπορεί μια κατηγορία A να είναι γονέας της B, και ταυτόχρονα παιδί κάποιας υποκατηγορίας της B. Αυτό γίνεται ώστε να αποφευχθούν κυκλικές ατέρμονες διαδικασίες.

E-08-09A	
X-08-09A	
X-08-09B	
Όνοματεπώνυμο	Ζαχαριουδάκης Εμμανουήλ
ΑΜ	1567
E-08-09B	
X-09-10A	
X-09-10B	
E-09-10A	
E-09-10B	
X-10-11A	
Πατρώνυμο	Αριστείδη
X-10-11B	
E-10-11A	
E-10-11B	
Εξάμηνο Εισαγωγής	04-05X

Old_students ▾ Save

Εικόνα 18: Node View

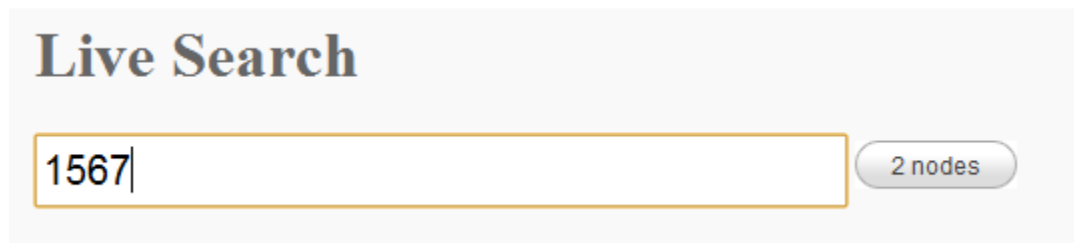
E-08-09A	
X-08-09A	
X-08-09B	
Όνοματεπώνυμο	Μανού Δήμητρα
ΑΜ	1
E-08-09B	
X-09-10A	
X-09-10B	
E-09-10A	
E-09-10B	
X-10-11A	
Πατρώνυμο	Γεωργίου
X-10-11B	
E-10-11A	
E-10-11B	
Εξάμηνο Εισαγωγής	99-00E

Old_students ▾ Save

- Old_students
- New Students
- Test Category

Εικόνα 19: Node Edit

Στη παραπάνω εικόνα βλέπουμε ένα Node. Ένα σύνολο δηλαδή, δεδομένων τα οποία έχουν συλλεχθεί από της πηγές. Στην αριστερή στήλη βλέπουμε τα πεδία που ο διαχειριστής έχει ορίσει και στη δεξιά στήλη, τα δεδομένα τα οποία έχει συλλέξει το σύστημά μας, αντίστοιχα. Στο τέλος βλέπουμε από ένα drop down menu, ο διαχειριστής της εφαρμογής μπορεί να κατηγοριοποιήσει το node, αυτό, σε μία από της κατηγορίες που έχει δημιουργήσει.



Εικόνα 20: Live Search Module

Το Module live search, αποτελεί μια πολύ καλή μηχανή αναζήτησης, στα δεδομένα του χρήστη. Στο παράδειγμά μας, αναζητάμε το Keyword “1567”. Δεξιά, μόλις από το Module, μας εμφανίζει σε real time τα αποτελέσματα τα οποία υπάρχουν στην βάση δεδομένων και ταιριάζουν με την λέξη που αναζητάμε. Πατώντας πάνω του, μας εμφανίζει την παρακάτω εικόνα.



Εικόνα 21: Live Search Module - Results

Εδώ βλέπουμε τα αποτελέσματα, της αναζήτησης, που μόλις κάναμε. σε πρώτη φάση βλέπουμε τα 2 group των αποτελεσμάτων. Το κάθε group αποτελείται από την λέξη κλειδί που αναζητήσαμε, και στη συνέχεια από την πηγή στην οποία ανήκει. Βλέπουμε λοιπόν, ότι το Keyword 1567 υπάρχει σε 2 διαφορετικές πηγές.

Έστω ότι μας ενδιαφέρει η πληροφορία που υπάρχει στο πρώτο group (εκεί που υπάρχει ο cursor). Μπορούμε να μεταβούμε σε 2 διαφορετικές σελίδες. Στο, αυτό καθ' αυτό Node, και να το επεξεργαστούμε (Εικόνα 18), ή σε προβολή όλων των Nodes, της πηγής που θα επιλέξουμε.

Πατώντας πάνω στην πηγή, οδηγούμαστε στην σελίδα, που φαίνεται παρακάτω, και βλέπουμε όλα τα nodes που υπάρχουν σε αυτή, με μορφή κατακόρυφης λίστας (Εικόνα 21).

Μπορούμε να μεταβούμε στην επεξεργασία κάθε Node πατώντας το SetCategory, πάνω αριστερά από κάθε node.

[Set Category](#)

E-08-09A:

X-08-09A:

X-08-09B:

Όνοματεπώνυμο: Μανιού Δήμητρα

ΑΜ: 1

E-08-09B:

X-09-10A:

X-09-10B:

E-09-10A:

E-09-10B:

X-10-11A:

Πατρώνυμο: Γεωργίου

X-10-11B:

E-10-11A:

E-10-11B:

Εξάμηνο 99-00E
Εισαγωγής:

[Set Category](#)

E-08-09A:

X-08-09A:

X-08-09B:

Όνοματεπώνυμο: Κοντάκης Ιωάννης

ΑΜ: 2

E-08-09B:

X-09-10A:

X-09-10B:

E-09-10A:

E-09-10B:

X-10-11A:

Πατρώνυμο: Γεωργίου

X-10-11B:

E-10-11A:

E-10-11B:

Εικόνα 22: Nodes From One Source

- *Nodes Preview:* Τέλος, προσφέρεται στον χρήστη, η επισκόπηση των Nodes, των οποίων η ταξινόμηση πρέπει να είναι με όλες τις δυνατές επιλογές.

Show 10 entries		Search:																								
Node	Source name	Category	E-09A	X-09A	Y-09A	AM	E-09B	X-10A	Y-10A	X-09B	Y-09B	E-09-10A	E-10B	X-10B	Y-10B	Parentesco	X-11B	Y-11B	E-11B	E-09-10B	E-09-11B	E-09-11A	E-09-11B	E-09-11A	E-09-11B	
2	Βιβλιογραφία Ευρωπαϊκής All Years_OOD	Old_students	-	-	-	1	-	-	-	-	-	-	-	-	-	Γεωργίου	-	-	-	-	-	-	-	-	-	99-00E
3	Βιβλιογραφία Ευρωπαϊκής All Years_OOD		-	-	-	2	-	-	-	-	-	-	-	-	-	Γεωργίου	-	-	-	-	-	-	-	-	-	99-00E
4	Βιβλιογραφία Ευρωπαϊκής All Years_OOD		-	-	-	3	-	-	-	-	-	-	-	-	-	Δαμπιέρου	-	-	-	-	-	-	-	-	-	99-00X
5	Βιβλιογραφία Ευρωπαϊκής All Years_OOD	Old_students	-	-	-	4	-	-	-	Absent	Absent	-	-	-	-	Εμμανουήλ	-	-	-	-	-	-	-	-	-	99-00E
6	Βιβλιογραφία Ευρωπαϊκής All Years_OOD		-	-	-	5	-	-	-	-	-	-	-	-	-	Δαμπιέρου	-	-	-	-	-	-	-	-	-	99-00X
7	Βιβλιογραφία Ευρωπαϊκής All Years_OOD		-	-	-	6	-	-	-	-	-	-	-	-	-	Ιωάννη	-	-	-	-	-	-	-	-	-	99-00X
8	Βιβλιογραφία Ευρωπαϊκής All Years_OOD	New Students	-	-	-	7	-	-	-	-	-	-	-	-	-	Αβωνάου	-	-	-	-	-	-	-	-	-	99-00X
9	Βιβλιογραφία Ευρωπαϊκής All Years_OOD		-	-	-	8	-	-	-	-	-	-	-	-	-	Γεωργίου	-	-	-	-	-	-	-	-	-	99-00X
10	Βιβλιογραφία Ευρωπαϊκής All Years_OOD		-	-	-	9	-	-	-	-	-	-	-	-	-	Ιωάννη	-	-	-	-	-	-	-	-	-	99-00X
11	Βιβλιογραφία Ευρωπαϊκής All Years_OOD		-	-	-	10	-	-	-	-	-	-	-	-	-	Εμμανουήλ	-	-	-	-	-	-	-	-	-	99-00X

Showing 1 to 10 of 2,809 entries

First Previous 1 2 3 4 5 Next Last

Στο Data Table που περιέχει η εφαρμογή, για την προβολή των nodes, υπάρχει η δυνατότητα ταξινόμησης των nodes, όπως είπαμε παραπάνω, αλλά και το φιλτράρισμα τους. Στον χρήστη παρέχεται η δυνατότητα, εισαγωγής κάποιας λέξης κλειδί, και φιλτράρισμα των αποτελεσμάτων βάση αυτού. Εξαιρετικά σημαντικό πλεονέκτημα του συγκεκριμένου module, είναι το ότι για το φιλτράρισμα των αποτελεσμάτων και για την αναταξινόμησή τους, δεν χρειάζεται η επαναφόρτωση

(reload) της σελίδας, κάτι που δεν καθυστερεί το server, και δεν τον επιβαρύνει με επιπλέον επεξεργασία.

6. Συμπεράσματα

Σημείο αναφοράς αυτής της πτυχιακής ήταν η έρευνα για τους τρόπους με τους οποίους μπορούμε να συλλέξουμε δεδομένα, σε μια ενιαία μορφή, ομογενής, από πολλές διαφορετικές πηγές και εταιρωγενείς μεταξύ τους. Επίσης στόχος της πτυχιακής, ήταν η υλοποίηση ενός τέτοιου συστήματος, το οποίο θα υποστηρίζει και πολλαπλούς χρήστες.

Σχετική έρευνα στο διαδύκτιο μας έδειξε, ότι τέτοια συστήματα, δεν παρέχονται, τουλάχιστον σχετικά με την δική μας υλοποίηση. Προφανώς απομονωμένα συστήματα υπάρχουν, όπως excel parsers ή xml parsers, αλλά όχι για τέτοια χρήση. Αξίζει να σημειωθεί ότι λόγω του όγκου των δεδομένων, χρειάζεται υψηλή υπολογιστική ισχύς, και μια τεράστια βάση δεδομένων. Αυτό έχει υψηλό κόστος, κάτι το οποίο σίγουρα είναι ένα εμπόδιο για την δημιουργία τέτοιων συστημάτων.

Η εφαρμογή μας, έχει την μορφή μιας σωστά δομημένης πλατφόρμας, εύκολα αναπτύξιμης και προσαρμόσιμης, στα εκάστοτε δεδομένα. Θα μπορούσε πολύ εύκολα να προστεθούν και άλλοι parsers, αν προκύψει η ανάγκη, όπως και η εξαγωγή των δεδομένων σε διάφορες μορφές. Η κατασκευή της εφαρμογής, περιέχει στον πυρήνα της, (brain όπως ονομάζω), functions σε php, που μπορούν να χρησιμοποιηθούν για την ευρύτερη ανάπτυξή της.

Κατά την δημιουργία της εφαρμογής, ήρθα αντιμέτωπος με πολλά προβλήματα, όπως η διαχείριση της ροής δεδομένων, και η ελαχιστοποίηση της υπολογιστικής ισχύς, προβλήματα που από το interface της εφαρμογής, δεν είναι ορατά. Προσωπικά θεωρώ ότι η εφαρμογή δεν θα έπρεπε να παρέχει την υποστήριξη πολλαπλών χρηστών, αλλά να μοιράζεται στον εκάστοτε υπονήφιο χρήστη, και να γίνεται εγκατάσταση στο δικό τοπικό τερματικό, για την διαχείριση των δικών του δεδομένων. Τέτοια συστήματα, υπάρχουν όπως εφαρμογές CMS (Διαχείρισης περιεχομένου) όπως το Joomla και το Wordpress.

Σε συνέχεια της προσωπικής μου άποψης, θεωρώ ότι εξελίξαμε το μοντέλο κατά ένα ακόμη βήμα, με το να μπορούμε να διαχειριστούμε απεριόριστα δεδομένα, αφού, μπορούμε να κατηγοριοποιήσουμε - ενοποιήσουμε τελείως διαφορετικά δεδομένα, διαφορετικών συλλογών, διαφορετικών χρηστών σε τελείως διαφορετικά επίπεδα. Σε μελλοντική, πιθανή ανάπτυξη της εφαρμογής, θα μπορούσε να επεκταθεί και στην έξυπνη εξαγωγή γραφημάτων για στατιστικούς λόγους.

7. Βιβλιογραφία

Μέρος του υλικού που παρουσιάστηκε σε αυτή την πτυχιακή βρέθηκε στο διαδίκτυο από τους ακόλουθους συνδέσμους:

<http://en.wikipedia.org>

<http://dev.mysql.com/>

<http://www.mysql.com>

<http://www.jboss.com>

<http://www.w3.org>

<http://www.yourhtmlsource.com>

<http://www.apl.jhu.edu>