



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Πτυχιακή Εργασία

Υλοποίηση διαδικτυακής εφαρμογής διαχείρισης του Προγράμματος Μαθημάτων του Τμήματος Πληροφορικής



Φοιτητής
Σέργιος Γαβριηλίδης
Αρ. Μητρώου Φοιτητή: 05/2954

Επιβλέπων Καθηγητής
Ευκλείδης Κεραμόπουλος

Θεσσαλονίκη 2012

Ευχαριστίες

Θα ήθελα να ευχαριστήσω όλους τους καθηγητές μου στο ΑΤΕΙ τμήματος πληροφορικής Θεσσαλονίκης και κυρίως τους Κεραμόπουλο Ευκλείδη και Δέρβο Δημήτριο για τις γνώσεις και την αγάπη για αναζήτηση που μου εμφύσησαν πάνω στο πεδίο της Βάσεις Δεδομένων.

Τέλος, και πιο σημαντικά, θα ήθελα να ευχαριστήσω τους γονείς μου που με στήριξαν με όλους τους τρόπους κατά την πορεία μου σε αυτόν τον κόσμο.

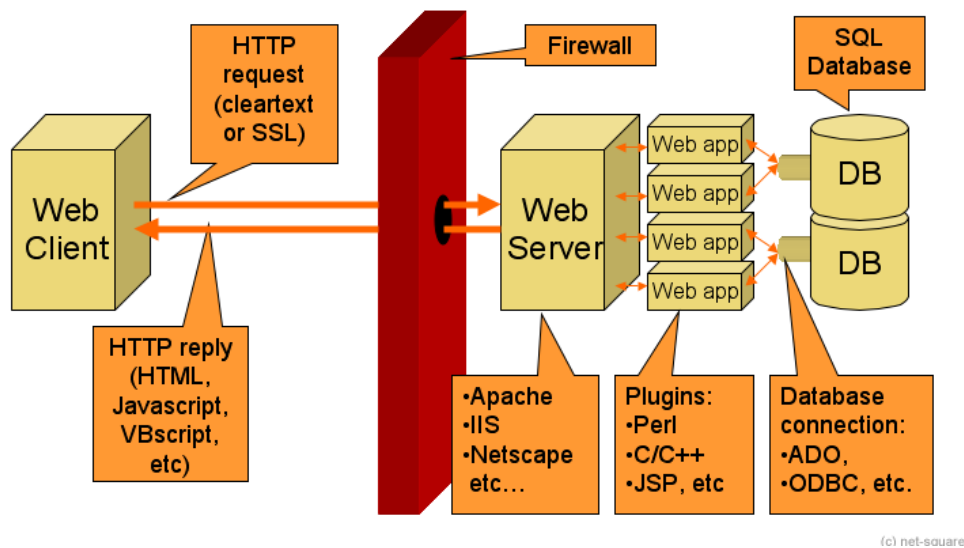
" Μηδένα προ του τέλους μακάριζε "

Σόλων, 630-560 π χ., Αθηναίος νομοθέτης & φιλόσοφος

Abstract

Web Services and Generally Web Applications are continuing to enter our lives on every platform possible [1]. It isn't uncommon for every major software developer to port some, or all of its product functionality to web and internet generally. For that reason tools as IBM DB2, RESTFULL services, SOAP and others have been extended beyond imaginations from their start roughly 5 years ago making web application programming easier and more powerful than ever before.

In this theses we will examine some of these tools and we will explain how these technologies can be combined in order to create a Web application that is comparable with any desktop, but has also all the advantages that web can give. At the following project we created a php-based webpage which connects to a database and is used to manage the education procedure. The main technologies used to achieve this goal are HTML, PHP and JavaScript for the development of our website and also IBM DB2 with Web Services for database management.



Generic web app use

Περίληψη

Οι υπηρεσίες Ιστού και γενικά οι εφαρμογές Ιστού συνεχίζουν να μπαίνουν στις ζωές μας σε κάθε πιθανή πλατφόρμα. Δεν είναι ασυνήθιστο για κάθε καλό προγραμματιστή λογισμικού να μετατρέψει μερικώς, ή ολοκληρωτικά την λειτουργία προϊόντων του σε υπηρεσίες Ιστού και του Διαδικτύου γενικά. Για αυτόν τον λόγο εργαλεία σαν το IBM DB2, υπηρεσίες RESTFULL, SOAP και άλλα, έχουν επεκταθεί πέρα από κάθε φαντασία, από την εποχή της έναρξής τους, πριν κατά προσέγγιση 5 έτη, καθιστώντας τον προγραμματισμό κάποιας εφαρμογής ιστού ευκολότερη και ισχυρότερη από ποτέ.

Σε αυτήν την πτυχιακή θα εξετάσουμε μερικά από αυτά τα εργαλεία και θα εξηγήσουμε πώς αυτές οι τεχνολογίες μπορούν να συνδυαστούν προκειμένου να δημιουργηθεί μια εφαρμογή Ιστού που είναι συγκρίσιμη με οποιαδήποτε υπολογιστή γραφείου, αλλά έχει επίσης όλα τα πλεονεκτήματα που ο Ιστός μπορεί να παρέχει.

Στην παρακάτω εργασία αναλύεται η δημιουργία μιας ιστοσελίδας η οποία συνδέεται με μια βάση δεδομένων και χρησιμοποιείται για την διαχείριση των μαθημάτων, αιθουσών, καθηγητών και της λειτουργίας του τμήματος Πληροφορικής. Οι βασικές τεχνολογίες που χρησιμοποιήθηκαν για την ολοκλήρωση της ιστοσελίδας είναι οι HTML, PHP και JavaScript για την ανάπτυξη της ιστοσελίδας καθώς και η IBM DB2 με Web Services για την διαχείριση της βάσης δεδομένων.

Ευρετήριο Περιεχομένων

Ευχαριστίες	2
Abstract	3
Περίληψη	4
Θέμα Εργασίας	11
1.Εισαγωγή	12
1.1. Περίληψη.....	12
1.2. Κίνητρο για την Εκπόνηση της Εργασίας.....	12
1.3. Σκοπός και Στόχος Εργασίας.....	12
1.4. Δομή Εργασίας.....	13
2.PHP	15
2.1 Εισαγωγή.....	15
2.2 Ορισμός.....	15
2.3 Επεκτάσεις αρχείων και διακομιστές.....	16
2.4 Εναλλακτικός τρόπος εκτέλεσης ιστοσελίδων.....	16
2.5 Ιστορία της PHP.....	17
2.6 Χρήση της γλώσσας PHP.....	18
2.7 Ασφάλεια.....	20
2.8 Ταχύτητα βελτιστοποίηση.....	21
2.9 Επίλογος.....	22
3.XAMMP	23
3.1 Ορισμός.....	23
3.2 Χρήση.....	23
4. APACHE	24
4.1 Εισαγωγή.....	24
4.2 Ορισμός.....	24
4.3 Απόδοση.....	25
4.4 Επίλογος.....	25
5. IBM DB2	28
5.1 Εισαγωγή.....	28
5.1.1 Βασικές Έννοιες.....	28
5.1.2 Ιστορία – Εξέλιξη βάσης δεδομένων.....	29

5.2 Ιστορία της IBM DB2.....	31
5.3 Τα χαρακτηριστικά και οφέλη χρήσης DB2 10:.....	33
5.4 Τεχνικές πληροφορίες IBM DB2	33
5.5 Επίλογος	34
6. WEB SERVICES.....	35
6.1 Εισαγωγή	35
6.2 Ιστορική αναδρομή.....	35
6.3 Ορισμός Web Services	36
6.4 Οι ανάγκες που οδήγησαν στις υπηρεσίες ιστού.....	39
6.5 Χαρακτηριστικά	39
6.6 Αρχιτεκτονική υπηρεσιών ιστού	40
6.7 Είδη interface Υπηρεσιών.....	41
6.8 SOAP.....	42
6.8.1 Διεργασιακό Μοντέλο (processing model).....	45
6.8.2 Δομή Μηνυμάτων Σφάλματος	47
6.9 REST	47
6.10 XML	48
6.10.1 Το ιεραρχικό μοντέλο της XML	49
6.10.2 Ορθώς ορισμένο XML έγγραφο.....	50
6.10.3 Έγκυρο XML έγγραφο	50
6.10.4 XML schema	51
6.11 JSON	51
6.11.1 Values	52
6.11.2 Object.....	52
6.11.3 Arrays.....	54
6.11.4 JSONP	55
6.11.5 Θέματα ασφαλείας	56
6.12 Επίλογος.....	56
7 ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΗΣ ΚΑΙ ΔΙΑΓΡΑΜΜΑΤΑ	58
7.1 Εισαγωγή	59
7.2 Μοντέλο Περίπτωσης Χρήσης.....	62
7.3 Περίπτωση Χρήσης: Εισαγωγή στο Σύστημα	63
7.4 Περίπτωση Χρήσης: Δημιουργία, Διαγραφή, Τροποποίηση Καθηγητή/Μαθήματος/Απαλλαγών	64

7.5 Περίπτωση Χρήσης: Δημιουργία Μαθήματος στο Πρόγραμμα.....	65
7.6 Περίπτωση Χρήσης: Τροποποίηση Μαθήματος στο Πρόγραμμα.....	66
7.7 Περίπτωση Χρήσης: Διαγράφη Μαθήματος στο Πρόγραμμα	67
7.8 Επίλογος	68
8.ΟΔΗΓΟΣ ΧΡΗΣΗΣ ΛΟΓΙΣΜΙΚΟΥ	69
8.1 Εισαγωγή	69
8.2 Είσοδος χρήστη.....	70
8.3 Διαχείριση “Νέο Εξάμηνο”.....	75
8.4 Διαχείριση “BackUp Προηγούμενου Εξαμήνου”	76
8.5 Διαχείριση Αιθουσών	78
8.6 Διαχείριση Μαθημάτων	79
8.7 Διαχείριση Απαλλαγών.....	81
8.8 Διαχείριση Επιτροπών	81
8.9 Διαχείριση Καθηγητών	82
8.10 Διαχείριση Βαθμίδων.....	85
8.11 Διαχείριση “Πρόγραμμα”	85
8.12 Διαχείριση «Αναφορές»	89
9.Συμπεράσματα - Προτάσεις	95
10. Βιβλιογραφία	98

Ευρετήριο Εικόνων

Εικόνα 2.1 Τυπική χρήση ρηρ.....	18
Εικόνα 2.2 Διάφορες εκδόσεις Χampp.....	20
Εικόνα 2.3 Γενική ασφάλεια της ρηρ.....	21
Εικόνα 2.4 Λογότυπο PHP Speedy.....	22
Εικόνα 6.1 Η προσέγγιση σύνδεση/δημοσίευση/εντοπισμός	41
Εικόνα 6.2 Η δομή ενός μηνύματος SOAP	43
Εικόνα 6.3 Οι δυνατές τιμές των βασικών τύπων σε JSON	52
Εικόνα 6.4 Η μορφή του αντικειμένου σε JSON.....	53
Εικόνα 6.5 Η γενική μορφή ενός πίνακα στη JSON.....	54
Εικόνα 7.1 Βάση Δεδομένων	61
Εικόνα 8.1 Είσοδος χρήστη.....	70
Εικόνα 8.2 Λάθος στοιχεία	71
Εικόνα 8.3 Επιτυχής είσοδος χρήστη	71
Εικόνα 8.4 Αποσύνδεση χρήστη	75
Εικόνα 8.5 Ενεργοποιώντας cookies	75
Εικόνα 8.6 Νέο Εξάμηνο.....	76
Εικόνα 8.7 BackUp Προηγούμενου Εξαμήνου	76
Εικόνα 8.8 BackUp κάθε πίνακα δεδομένων	77
Εικόνα 8.9 Αποθήκευση της βάσης	78
Εικόνα 8.10 Διαχείριση «Αίθουσα»	79
Εικόνα 8.11 Διαχείριση Μαθήματος.....	80
Εικόνα 8.12 Εισαγωγή είδους μαθήματος.....	80
Εικόνα 8.13 Διαχείριση Απαλλαγών	81
Εικόνα 8.14 Διαχείριση Επιτροπών	82
Εικόνα 8.15 Διαχείριση Καθηγητών	83
Εικόνα 8.16 Εισαγωγή τηλεφώνου.....	84
Εικόνα 8.17 Εισαγωγή ηλεκτρονικής διεύθυνσης	84
Εικόνα 8.18 Διαγραφή τηλεφώνου	84
Εικόνα 8.19 Διαχείριση Βαθμίδων.....	85
Εικόνα 8.20 Διαχείριση 'Πρόγραμμα'	86
Εικόνα 8.21 Αποτέλεσμα αναζήτησης	87

Εικόνα 8.22 Εισαγωγή μάθημα.....	87
Εικόνα 8.23 Εισαγωγή μαθήματος.....	88
Εικόνα 8.24 Γενικές πληροφορίες μαθήματος.....	88
Εικόνα 8.25 Ειδικές πληροφορίες μαθήματος.....	89
Εικόνα 8.26 Αναφορές.....	91
Εικόνα 8.28 Παράδειγμα αναφοράς 'Εξαμηνολόγιο'.....	92
Εικόνα 8.29 Εκτύπωση όλων των στοιχείων.....	92
Εικόνα 8.30 Αναφορά 'Συνδιδασκαλία ανά μάθημα'.....	93

Θέμα Εργασίας

Τα IBM DB2 Data Web Services διευκολύνουν σημαντικά την ανάπτυξη, εγκατάσταση, και τη διαχείριση των Web Services με βάση την πρόσβαση σε DB2 διακομιστές. Με την χρήση των Data Web Services μας παρέχονται εργαλεία για τη δημιουργία Web Services τα οποία βασίζονται σε SQL εντολές και Stored Procedures. Η υλοποίηση των Data Web Services στο περιβάλλον IBM Data Studio γίνεται με απλές λειτουργίες «drag and drop».

Στόχος της συγκεκριμένης πτυχιακής είναι η χρήση της παραπάνω τεχνολογίας για τη δημιουργία μίας διαδικτυακής εφαρμογής διαχείρισης του προγράμματος μαθημάτων ενός Τμήματος Ανώτατου Εκπαιδευτικού Ιδρύματος. Η εφαρμογή θα υλοποιηθεί λαμβάνοντας υπόψη τις ιδιαιτερότητες του Τμήματος Πληροφορικής.

1.Εισαγωγή

1.1. Περίληψη

Το Διαδίκτυο και η τεράστια ανάπτυξη του, προσέδωσε απρόσμενη ώθηση σ' όλες τις ανθρώπινες δραστηριότητες, Η πτυχιακή αυτή αποσκοπεί στην μελέτη, σχεδίαση και ανάπτυξη μιας ιστοσελίδας για την ευκολότερη και γρηγορότερη διαχείριση των θεμάτων του ΑΤΕΙ. Παρουσιάζονται σ' αυτή την εργασία οι υποδομές και η πορεία ανάπτυξης των τεχνολογιών του Διαδικτύου και πως αυτές μας οδήγησαν στα συστήματα που θα παρουσιαστούν εδώ . Όλες αυτές οι τεχνολογίες έχουν παραδείγματα και οι βασικές τους λειτουργίες επεκτάθηκαν και μέσα σε αυτή την πτυχιακή οπου παρουσιάζονται με συνεχή τρόπο στον αναγνώστη.

Πιο συγκεκριμένα, χρησιμοποιήσαμε την γλώσσα διαδικτυακών σελίδων PHP για την βασική ανάπτυξη, την DB2 για την αποθήκευση των δεδομένων σε βάση, το XAMMP και WebSphere για Web server, και το NETBEANS για εργαλείο ανάπτυξης.

1.2. Κίνητρο για την Εκπόνηση της Εργασίας

Ο βασικότερος λόγος που επέλεξα αυτή την πτυχιακή είναι το προσωπικό ενδιαφέρον που είχα για τις υπηρεσίες του ΑΤΕΙ και συγκεκριμένα την πορεία των δηλώσεων, μαθημάτων και βαθμών, αλλά και για την απόκτηση εμπειρίας στον τομέα αυτόν. Επιπρόσθετος, γνωρίζοντας ότι στην αγορά εργασίας υπάρχει ζήτηση σε διαχείριση βάσεων δεδομένων καθώς και ότι πλέον είναι απαραίτητη η ικανότητα ανάπτυξης ιστοσελίδων, πιστεύω ότι θα με βοηθήσει για την μετέπειτα επαγγελματική μου αποκατάσταση.

1.3. Σκοπός και Στόχος Εργασίας

Σκοπός της πτυχιακής εργασίας είναι η χρησιμοποίηση των τελευταίων τεχνολογιών για την ανάπτυξη μιας δυναμικής ιστοσελίδας, η οποία θα μπορεί να

περιέχει και να διαχειρίζεται μέσα από την χρήση Web Services το περιεχόμενο μιας σχεσιακής βάσης δεδομένων βασισμένη στην IBM DB2. Θα πρέπει να έχει ένα ευανάγνωστο και εύκολα διαχειρήσιμο γραφικό περιβάλλον αλλά και επίσης να είναι διαδραστικό σε κάθε τομέα, γρήγορο όσο γίνεται και σταθερό.

1.4. Δομή Εργασίας

Η πτυχιακή μου εργασία χωρίζεται σε 9 κεφάλαια μετά την εισαγωγή με βάση το περιεχόμενο του κάθε τμήματος ανάπτυξης του προγράμματος της πτυχιακής Ξεκινώντας την ανάλυση, το πρώτο κεφάλαιο περιστρέφεται στην Θεωρητική παρουσίαση των σύγχρονων τεχνολογιών του διαδικτύου, που σχετίζονται κυρίως με δυναμικές ιστοσελίδες και web προγράμματα όπου χρησιμοποιήθηκαν για την συγκεκριμένη εργασία, καθώς και με την τεχνική παρουσίαση τους και τους λόγους για τους οποίους επιλέχτηκαν οι συγκεκριμένες. Συνεχίζοντας αναπτύξαμε τις λειτουργίες της δυναμικής ιστοσελίδας μας σε PHP. Δηλαδή, το πως μπορούμε να την χρησιμοποιήσουμε για να φτιάξουμε μια κλασσική ιστοσελίδα τύπου HTML από δυναμικό περιεχόμενο που έχουμε πάρει, από βάσεις δεδομένων και στοιχεία χρηστών.

Στο επόμενο κεφάλαιο αναπτύσσουμε την σημασία που έχει η νέα σχετικά τεχνολογία των Web Services σε αυτή την πτυχιακή καθώς και πώς γενικότερα λειτουργούν αυτές αλλά και ειδικότερες αρχιτεκτονικές σχεδίασης και ανάπτυξης τους. Όστε συνεπώς να καταφέρουμε να ενώσουμε την IBM DB2 βάση δεδομένων με το πρόγραμμα μας.

Στο προτελευταίο κεφάλαιο εξηγούμε το πώς θα είναι τα διαγράμματα του προγράμματος αλλά και τις διαφορετικές πιθανές εκβάσεις που μπορεί να πάρει η πορεία του (User Cases), και στη συνέχεια το κεφάλαιο «Οδηγός χρήσης Λογισμικού». Στο τέλος βρίσκουμε τον επίλογο της πτυχιακής μας εργασίας αναπαράγοντας εν συντομία τα παραπάνω.



"The hardest part of design ... is keeping features out."

Donald Norman

2.PHP

2.1 Εισαγωγή

Κάθε γλώσσα προγραμματισμού διαδικτυακών, δυναμικών ιστοσελίδων έχει τα υπέρ και τα κατά της. Αναφορικά μπορώ να παρουσιάσω τις γλώσσες JSP, ASP, PHP οι οποίες είναι και οι πιο γνωστές στον χώρο της πληροφορικής. Μετά από σχετική έρευνα αποφασίστηκε η χρήση της PHP για τους ακόλουθους λόγους:

1. Η Γλώσσα είναι ανοικτού κώδικα και δωρεάν για οποιαδήποτε χρήση.
2. Έχει μεγάλη αποδοχή και υποστήριξη από όλα τα γνωστά προϊόντα πληροφορικής, όποτε εύκολα βρίσκεις κάποιο plugin για σύνδεση με βάση ή λύση σε κάποιο πρόβλημα.
3. Υποστηρίζεται από όλους τους web server, οπότε και μετά το πέρας της ολοκλήρωσης της εργασίας είναι σχεδόν σίγουρο ότι μπορεί το project αυτό να ανέβει σε οποιοδήποτε hosting πρόγραμμα.

Αυτό το κεφάλαιο περιγράφει μερικές από τις βασικές έννοιες της php καθώς και όλα όσα χρειάζονται στον αναγνώστη ώστε να μπορέσει να κατανοήσει σε βάθος το νόημα και τον ρόλο της χρήσης της στην συγκεκριμένη εργασία.

Ξεκινάμε με τον ορισμό της γλώσσας και συνεχίζουμε με τον τρόπο που αυτή χρησιμοποιείται και το πώς αναγνωρίζεται από κάποιο σύστημα έτσι ώστε να μπορέσει να μεταφραστεί από αυτό σαν γλώσσα, παρουσιάζοντας παράλληλα κάποιους εναλλακτικούς τρόπους υλοποίησης της. Προχωράμε με την παρουσίαση των ιστορικών στοιχείων της και την γενικότερη χρήση της, παρουσιάζοντας έπειτα και την υποστήριξη ασφάλειας που παρέχει αλλά και το πώς βελτιστοποιείται γενικότερα. Τέλος κλείνουμε με τον επίλογο, στον οποίο περιγράφουμε την γενικότερη εικόνα που μας έδωσε κατά την υλοποίηση του προγράμματος της πτυχιακής.

2.2 Ορισμός

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία διαδικτυακών σελίδων (web) με δυναμικό περιεχόμενο.

Μια σελίδα γραμμένη σε PHP περνά από την επεξεργασία του ενός συμβατού web server του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML (π.χ. Firefox, Google Chrome, Explorer).

2.3 Επεκτάσεις αρχείων και διακομιστές

Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (π.χ. *.php, *.php4, *.phtml κ.ά.). Το να γράψουμε κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στον browser του χρήστη τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME (Multipurpose Internet Mail Extension) types του server [2].

Επίσης, ακόμη και όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται και να μεταγλωττίζει τον κώδικα PHP σε HTML που θα μπορέσει να καταλάβει και να εμφανίσει το πρόγραμμα πελάτη (Internet Browser).

Ο διακομιστής Apache, που χρησιμοποιείται σήμερα ευρέως σε συστήματα με τα λειτουργικά συστήματα GNU/Linux, Microsoft Windows, Mac OS X υποστηρίζει εξ ορισμού την εκτέλεση κώδικα PHP.

2.4 Εναλλακτικός τρόπος εκτέλεσης ιστοσελίδων

Ο συνδυασμός Linux/Apache/PHP/MySQL, που είναι η πιο δημοφιλής πλατφόρμα εκτέλεσης ιστοσελίδων είναι γνωστός και με το ακρωνύμιο LAMP. Παρόμοια, ο συνδυασμός */Apache/PHP/MySQL ονομάζεται *AMP, όπου το πρώτο αρχικό αστεράκι αντιστοιχεί στην πλατφόρμα, στην οποία εγκαθίστανται ο Apache, η PHP και η MySQL (π.χ. Windows, Mac OS X).

Ο LAMP συνήθως εγκαθίσταται και ρυθμίζεται στο Linux με τη βοήθεια του διαχειριστή πακέτων της εκάστοτε διανομής [3]. Στην περίπτωση άλλων λειτουργικών συστημάτων, επειδή το κατέβασμα και η ρύθμιση των ξεχωριστών προγραμμάτων

μπορεί να είναι πολύπλοκη, υπάρχουν έτοιμα πακέτα προς εγκατάσταση, όπως το XAMPP [4] και το WAMP [5] για τα Windows και το MAMP [6] για το Mac OS X.

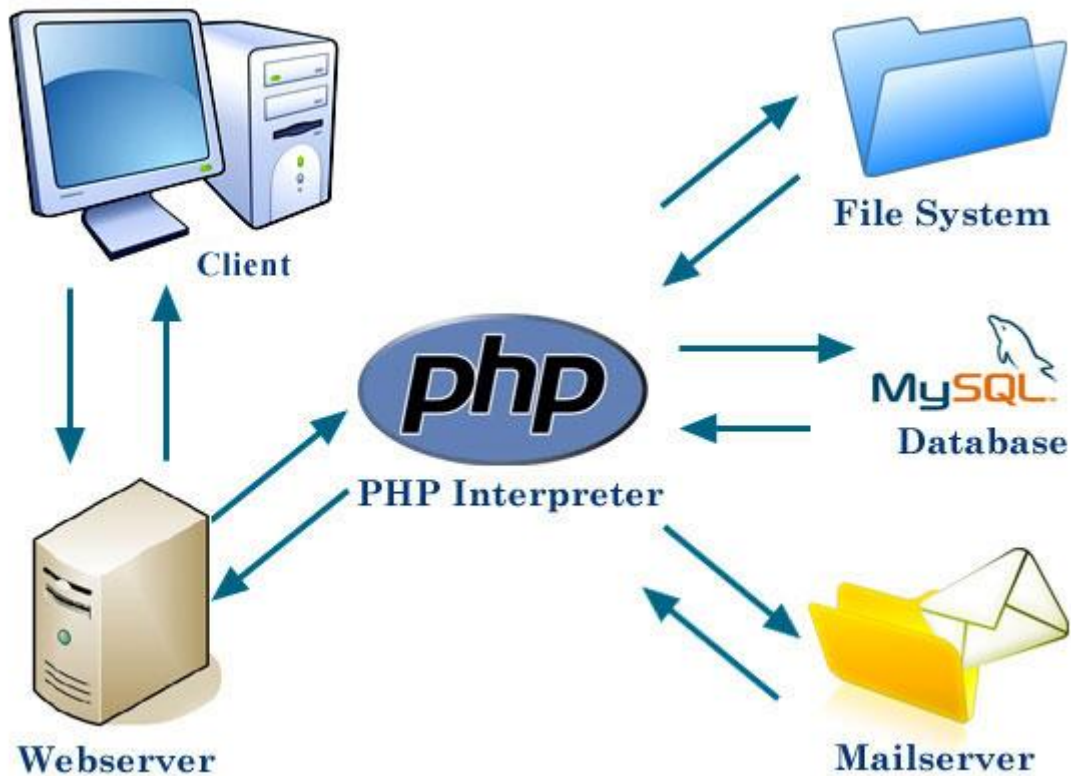
2.5 Ιστορία της PHP

Η ιστορία της PHP ξεκινά από το 1994, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό script με όνομα php.cgi, για προσωπική χρήση.

Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες.

Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστοτόπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της.

Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6, για οποιονδήποτε προγραμματιστή θέλει να τη χρησιμοποιήσει. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP.



Εικόνα 2.1 Τυπική χρήση php

2.6 Χρήση της γλώσσας PHP

Η PHP είναι μια γενικού σκοπού scripting γλώσσα προγραμματισμού, που είναι ιδιαίτερα κατάλληλη για server-side ανάπτυξη ιστοσελίδων, όπου η PHP τρέχει γενικά σε ένα διακομιστή Web.

Μπορεί επίσης να χρησιμοποιηθεί από την γραμμή εντολών, αλλά και με client-side GUI εφαρμογές. Η PHP μπορεί να εγκατασταθεί σε διάφορους διακομιστές Web, πολλά λειτουργικά συστήματα και πλατφόρμες, μπορεί επίσης να χρησιμοποιηθεί και με πολλά σχεσιακά συστήματα διαχείρισης βάσεων δεδομένων (RDBMS). Η PHP λειτουργεί ως ένα φίλτρο, λαμβάνοντας δεδομένα από το αρχείο ή ρεύμα που περιέχει το κείμενο και / ή τις PHP οδηγίες και αναλαμβάνοντας την αποθήκευση ή αποστολή σε μια άλλη ροή δεδομένων. Πιο συχνά το αποτέλεσμα θα είναι μια HTML ιστοσελίδα.

Τα **αντικείμενα**, η βασική λειτουργία του αντικειμενοστρεφούς προγραμματισμού, προστέθηκαν στην PHP 3 και βελτιώθηκαν στην PHP 4. Η λειτουργία αυτή ξαναγράφηκε από την αρχή για την PHP 5, επεκτείνοντας το σύνολο των χαρακτηριστικών και την αύξηση της απόδοσης τους. Σε προηγούμενες εκδόσεις της PHP, τα αντικείμενα αντιμετωπίζονται σαν είδη μεταβλητών [7].

Το μειονέκτημα εκείνης της μεθόδου ήταν ότι το σύνολο του αντικειμένου έπρεπε να αντιγραφεί όταν μια μεταβλητή ορίζονταν ή μεταβιβάζονταν ως παράμετρος σε μια μέθοδο. Στη νέα προσέγγιση, τα αντικείμενα χρησιμοποιούνται από αναφορά και όχι με βάση την τιμή τους. Η PHP 5 εισήγαγε `private` και `protected` μεταβλητές και μεθόδους, μαζί με αφηρημένες τάξεις και τάξεις τύπου `final` καθώς και `abstract` των μεθόδων και μεθόδους `final`.

Επίσης, εισήγαγε έναν τυποποιημένο τρόπο δήλωσης των κατασκευαστών (`constructor`) και καταστροφών (`destroyers`), μέθοδοι για την δημιουργία και καταστροφή αντικειμένων αντίστοιχα, παρόμοια με εκείνη των άλλων αντικειμενοστρεφών γλωσσών, όπως η C++ και JAVA, και ένα πρότυπο μοντέλο χειρισμού εξαιρέσεων.

Η PHP χρησιμοποιείται ως γλώσσα προγραμματισμού `server-side` στο 75% του συνόλου όλων των `Web servers`. Σχεδόν όλα τα `Web` συστήματα διαχείρισης περιεχομένου είναι γραμμένα σε PHP, περιλαμβανομένων των `MediaWiki`, `Joomla`, `EZ publish`, `WordPress`, του `Drupal` και του `Moodle`. Όλες οι τοποθεσίες `Web` που χρησιμοποιούν αυτά τα εργαλεία είναι γραμμένες σε PHP, συμπεριλαμβανομένου και γνωστών `site` όπως τα `Wikipedia`, `Facebook`, και `Digg`.

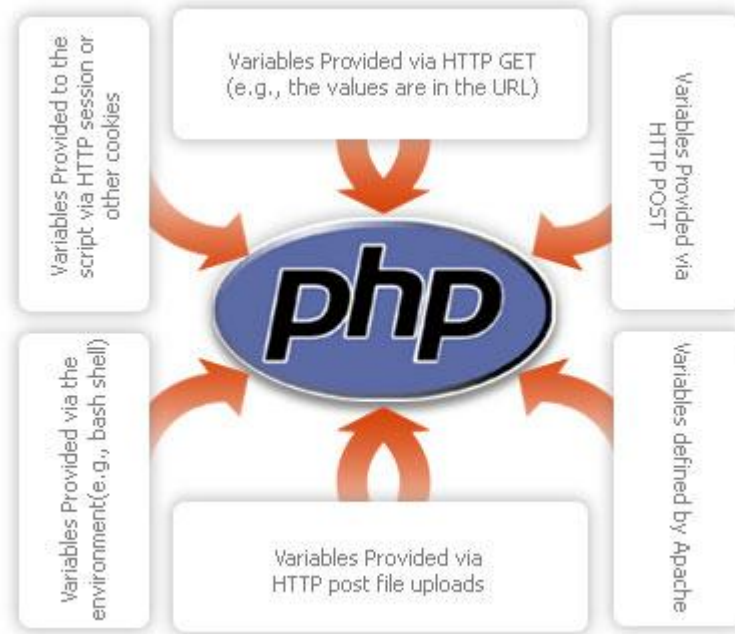


Εικόνα 2.2 διάφορες εκδόσεις Χαμπρ

2.7 Ασφάλεια

Οι Αδυναμίες στην ασφάλεια προκαλούνται κυρίως από τη μη τήρηση των απλών κανόνων προγραμματισμού για καλύτερη πρακτική εφαρμογή. Τεχνικές ατέλειες της ασφάλειας της ίδιας της γλώσσας ή των βιβλιοθηκών πυρήνα της δεν εμφανίζονται συχνά (23 το 2008, περίπου 1% του συνόλου). Γενικότερα, αναγνωρίζοντας ότι οι προγραμματιστές κάνουν λάθη, ορισμένες γλώσσες περιλαμβάνουν τον έλεγχο για να ανιχνεύσουν αυτόματα την έλλειψη επικύρωσης των δεδομένων εισόδου που προκαλεί πολλά προβλήματα (πχ SQL Injection) [8]. Ένα τέτοιο χαρακτηριστικό αναπτύσσεται και για την PHP, αλλά η ένταξη του σε ένα εξωτερικό πακέτο επέκτασης έχει απορριφθεί αρκετές φορές στο παρελθόν.

Υπάρχει γενικότερα προηγμένη προστασία, μπαλώματα (patches), ειδικά σχεδιασμένα για περιβάλλοντα Web hosting. Η PHPIDS προσθέτει την ασφάλεια σε οποιαδήποτε εφαρμογή PHP και τις δίνει την δυνατότητα να υπερασπιστεί κατά των επιθέσεων [9]. Η PHPIDS ανιχνεύει cross-site scripting (XSS), SQL injection , header injection , directory traversal , την απομακρυσμένη εκτέλεση αρχείων και Denial of Service (DoS) επίθεση [10].



Εικόνα 2.3 γενική ασφάλεια της php

2.8 Ταχύτητα βελτιστοποίηση

Ο PHP κώδικας είναι δημιουργημένος on-the-fly σε μια μορφή που μπορεί να εκτελεστεί από τη μηχανή της PHP. Προκειμένου να επιταχυνθεί ο χρόνος εκτέλεσης οι χρήστες δεν υποχρεούνται να δημιουργούν τον PHP κώδικα κάθε φορά που κάποιος μπαίνει στην ιστοσελίδα. Τα PHP script μπορούν επίσης να χρησιμοποιηθούν σε εκτελέσιμη μορφή χρησιμοποιώντας ένα εκτελέσιμο μεταγλωττιστή PHP .

Η Βελτιστοποίηση κώδικα γενικά αποσκοπεί στη βελτίωση της απόδοσης του μεταγλωττισμένου κώδικα, μειώνοντας το μέγεθός του, συγχωνεύοντας περιττές οδηγίες και κάνοντας άλλες αλλαγές που μπορούν να μειώσουν το χρόνο εκτέλεσης ενός script.

Μια άλλη προσέγγιση για τη μείωση των γενικών εξόδων κατάρτισης των PHP servers που χρησιμοποιούν PHP είναι η χρήση των σελίδων από την προσωρινή αποθήκευση της PHP (cache) σε μεταγλωττισμένη μορφή για να αποφευχθεί η επιβάρυνση της μεταγλώττισης του κώδικα κάθε φορά που τρέχει το script. Η μνήμη opcode cache, APC , θα κατασκευαστεί σε μια επερχόμενη έκδοση της PHP και συγκεκριμένα από την PHP 5.4 και οι βελτιστοποιήσεις κώδικα μπορούν να συνδυαστούν για καλύτερη

συνολικά απόδοση, εκτός όμως από τις τροποποιήσεις που δεν συσχετίζονται μεταξύ τους στην λειτουργία τους και συνεπώς δημιουργούν προβλήματα η μία στην άλλη (που αυτά τα προβλήματα συμβαίνουν σε διαφορετικά στάδια της δημιουργίας της σελίδας).



Εικόνα 2.4 Λογότυπο PHP Speedy

2.9 Επίλογος

Η PHP είναι μια γλώσσα που συντροφεύει το διαδίκτυο σχεδόν με την αλλαγή του από μέσο για λίγους σε μέσο επικοινωνίας. Έχει πολλές φορές ξαναγραφτεί με βάση τις ανάγκες κάθε εποχής και γρήγορα εμφανίζει λειτουργικότητα που είναι γενικά αποδεκτή και αναμενόμενη από το επίπεδο μιας τέτοιας γλώσσας. Η Δημοσιότητα της και το ότι είναι η πρώτη επιλογή πολλών μεγάλων εταιριών της δίνει την τωρινή της δύναμη και την ασφάλεια να πούμε ότι θα είναι το standard στην ανάπτυξη δυναμικών ιστοσελίδων για πολύ καιρό ακόμα.

3.ΧΑΜΜΡ

3.1 Ορισμός

Το ΧΑΜΡΡ είναι ακρωνύμιο και αναφέρεται στα παρακάτω αρχικά:

- Χ (αναφέρεται στο "cross-platform" που σημαίνει λογισμικό ανεξάρτητο πλατφόρμας)
- Apache HTTP [11]
- MySQL [12]
- PHP [13]
- Perl

Το ΧΑΜΡΡ είναι ένα ελεύθερο λογισμικό το οποίο περιέχει ένα εξυπηρετητή ιστοσελίδων το οποίο μπορεί να εξυπηρετεί και δυναμικές ιστοσελίδες τεχνολογίας PHP/MySQL. Είναι ανεξάρτητο πλατφόρμας και τρέχει σε Microsoft Windows, Linux, Solaris, and Mac OS X και χρησιμοποιείται ως πλατφόρμα για την σχεδίαση και ανάπτυξη ιστοσελίδων με την τεχνολογίες όπως PHP, JSP και Servlets.

3.2 Χρήση

Επίσημα οι σχεδιαστές του ΧΑΜΡΡ προόριζαν το λογισμικό ως εργαλείο ανάπτυξης και δοκιμής ιστοσελίδων τοπικά στον υπολογιστή χωρίς να είναι απαραίτητη η σύνδεση στο διαδίκτυο. Για να είναι δυνατή η χρήση του, πολλές σημαντικές λειτουργίες ασφάλειας έχουν απενεργοποιηθεί. Στην πράξη το ΧΑΜΡΡ ορισμένες φορές χρησιμοποιείται και για την φιλοξενία ιστοσελίδων (hosting). Υπάρχει ειδικό εργαλείο το οποίο περιέχεται στο ΧΑΜΡΡ για την προστασία με κωδικό των σημαντικών μερών και αρχείων μίας ιστοσελίδας. Το ΧΑΜΡΡ υποστηρίζει την δημιουργία και διαχείριση βάσεων δεδομένων τύπου MySQL και SQLite.

4. APACHE

4.1 Εισαγωγή

Αν η επιλογή γλώσσας προγραμματισμού δυναμικών ιστοσελίδων είναι δύσκολη κατά έναν βαθμό από τις πολλές επιλογές που παρέχονται, τότε η επιλογή web server είναι ακόμα πιο απαιτητική από πλευράς έρευνας, γιατί υπάρχουν κυριολεκτικά απεριόριστες επιλογές με σχεδόν την ισότιμη δυναμικότητα. Για αυτό ο σωστός τρόπος να επιλέξεις έναν είναι να μελετήσεις την υποστήριξη που δίνει, την ασφάλεια, την γενικότερη αποδοχή και ταυτόχρονα το κόστος. Σε αυτούς τους τομείς μόνο ο Apache HTTP SERVER ξεχωρίζει σαν την πρώτη επιλογή μεταξύ πολλών web server και πολλές φορές αξιολογείται ως καλύτερος και από τους εμπορικούς (π.χ. Microsoft IIS).

4.2 Ορισμός

Ο Apache http, γνωστός και απλά σαν Apache, είναι ένας εξυπηρετητής του παγκόσμιου ιστού (web). Όποτε ένας χρήστης επισκέπτεται ένα ιστότοπο το πρόγραμμα πλοήγησης (browser) επικοινωνεί με έναν διακομιστή (server) μέσω του πρωτοκόλλου HTTP, ο οποίος παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης. Ο Apache είναι ένας από τους δημοφιλέστερους, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες όπως τα Windows, το Linux, το Unix και το Mac OS X. Συντηρείται τώρα από μια κοινότητα ανοικτού κώδικα με επιτήρηση από το Ίδρυμα Λογισμικού Apache (Apache Software Foundation).

Ο Apache χρησιμοποιείται και σε τοπικά δίκτυα σαν διακομιστής συνεργαζόμενος με συστήματα διαχείρισης Βάσεων Δεδομένων π.χ. Oracle, MySQL.

Η πρώτη του έκδοση, γνωστή ως NCSA HTTPd, δημιουργήθηκε από τον Robert McCool και κυκλοφόρησε το 1993. Θεωρείται ότι έπαιξε σημαντικό ρόλο στην αρχική επέκταση του παγκόσμιου ιστού. Ήταν η πρώτη βιώσιμη εναλλακτική επιλογή που παρουσιάστηκε απέναντι στον εξυπηρετητή http της εταιρείας Netscape και από τότε έχει εξελιχθεί στο σημείο να ανταγωνίζεται με άλλους εξυπηρετητές βασισμένους στο Unix. Από το 1996 ήταν από τους πιο δημοφιλείς, όμως, από τον Μάρτιο του 2006 έχει

μειωθεί το ποσοστό της εγκατάστασής του κυρίως από τον Microsoft Internet Information Services και την πλατφόρμα .NET . Τον Οκτώβριο του 2007 το μερίδιο του ήταν 47.73% από όλους τους ιστοτόπους.

4.3 Απόδοση

Παρά το γεγονός ότι ο κύριος στόχος του σχεδιασμού του Apache δεν είναι να είναι ο "ταχύτερος" web server, ο Apache έχει επιδόσεις παρόμοιες με τους άλλους "υψηλής απόδοσης" web servers. Αντί της εφαρμογής μιας ενιαίας αρχιτεκτονικής, ο Apache παρέχει μια ποικιλία δυνατοτήτων πολυεπεξεργασίας ενοτήτων (MPMs), που του επιτρέπουν να τρέξει με τον τρόπο που χρειάζεται ώστε να ταιριάζει καλύτερα στις απαιτήσεις της κάθε συγκεκριμένης υποδομής.

Αυτό σημαίνει ότι η επιλογή της σωστής MPM και η σωστή ρύθμιση της είναι σημαντική. Όταν πρέπει να αυξηθεί η απόδοση ενός συστήματος, ο σχεδιασμός του Apache για να το επιτύχει αυτό είναι η μείωση των καθυστερήσεων και η αύξηση της απόδοσης ανά εντολή, σε σχέση με το χειρισμό απλά περισσότερων αιτήσεων που προτιμούν άλλοι server, εξασφαλίζοντας έτσι συνεπή και αξιόπιστη επεξεργασία των αιτήσεων σε εύλογο χρονικό διάστημα.

4.4 Επίλογος

Ο Apache κατά την διάρκεια της συγγραφής αυτής της πτυχιακής ποτέ δεν αποτέλεσε μέρος κάποιου προβλήματος σε οποιοδήποτε βαθμό και πάντα επιτελούσε το έργο του προβλεπόμενα. Επίσης είναι από τα λίγα κομμάτια του project της πτυχιακής από τα οποία δεν μπορούν να αντικατασταθούν με άλλη μορφής/παρόμοιων λειτουργιών πρόγραμμα, καθώς οι εγγενείς του λειτουργίες χρησιμοποιούνται πολλές φορές στην εργασία αυτή και συνεπώς έχει γίνει αναπόσπαστο μέρος του όχι όμως σε καμία περίπτωση με την έννοια της δέσμευσης σε συγκεκριμένο προϊόν, αλλά αντίθετα την εξύψωση του τελικού λογισμικού. Συνοψίζοντας, το γεγονός αυτό της κατά μία έννοια αρμονικής συνύπαρξης στον server αυτό και τις λειτουργίες που προσφέρει είναι κάτι

που συμβαίνει γενικά στην ανάπτυξη εφαρμογών σε apache web server και είναι αυτό που του χαρίζει και την διάρκεια στον χρόνο καθώς και την επιτυχία που έχει μέχρι τώρα.



"Software and cathedrals are much the same – first we build them, then we pray."

Sam Redwine.

5. IBM DB2

5.1 Εισαγωγή

Σε μεγάλα project όπως αυτό της πτυχιακής χρειάζεται να υπάρχει μια δομή η οποία είναι ικανή να στηρίξει την διαχείριση μεγάλων ποσών δεδομένων χωρίς να υπάρχει ο κίνδυνος αυτά να χαθούν ή να καταστραφούν κατά την χρησιμοποίηση. Το θέμα αυτό έχει αντιμετωπιστεί με την χρήση των βάσεων δεδομένων, οι οποίες αναλαμβάνουν την αποθήκευση και την ανάκτηση τους, όποτε το θελήσει ο προγραμματιστής, και έχουν επεκτείνει και την χρήση τους σαν ενδιάμεσοι



του προγράμματος και των δεδομένων οι οποίοι μπορούν να κατανοήσουν βασικές εντολές όπως ταξινόμηση, φιλτράρισμα και εισαγωγή/διαγραφή κάποιου στοιχείου τους. Μια από τις πιο ολοκληρωμένες λύσεις βάσεων δεδομένων είναι η IBM DB2 η οποία και χρησιμοποιείται στην συγκεκριμένη πτυχιακή και αφορά λύση που περιέχει όλα τα απαραίτητα στοιχεία και λειτουργικότητες για την ορθή της ολοκλήρωση. Στο συγκεκριμένο κεφάλαιο παρουσιάζουμε την IBM db2 και αναφερόμαστε σε όλα αυτά τα στοιχεία που την έκαναν τόσο δυνατή και ολοκληρωμένη, καθώς και μερικές της αδυναμίες.

5.1.1 Βασικές Έννοιες

1) Τι είναι μια βάση δεδομένων

Μια βάση δεδομένων είναι μια συλλογή από καλά οργανωμένα δεδομένα τα οποία σχετίζονται μεταξύ τους. Μια τέτοια βάση είναι για παράδειγμα τηλεφωνικός κατάλογος ο οποίος περιέχει αποθηκευμένα δεδομένα όπως το όνομα το επίθετο και ο αριθμός τηλεφώνου. Όταν στις έννοιες της πληροφορικής αναφερόμαστε στον όρο βάση δεδομένων εννοούμε ένα σύνολο σχετιζόμενων δεδομένων (τμημάτων πληροφορίας) τα οποία αποθηκεύονται ηλεκτρονικά.

2) Συστήματα διαχείρισης Βάσεων Δεδομένων

Οι βάσεις δεδομένων οι οποίες είναι αποθηκευμένες στο σκληρό δίσκο του υπολογιστή ή ακόμη και σε ένα δίκτυο συστημάτων διαχειρίζονται από ένα πρόγραμμα λογισμικού υπολογιστών το οποίο ονομάζεται Database Management System (DBMS) ή Σύστημα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ). Μέσω αυτού του λογισμικού τα προγράμματα των υπολογιστών μπορούν να δημιουργούν, να συντηρούν και να χρησιμοποιούν μια βάση δεδομένων. Έτσι, ένας οργανισμός μπορεί να αναπτύσσει εύκολα βάσεις δεδομένων για οποιαδήποτε εφαρμογή χρειαστεί.

3) Διαχειριστής Βάσεων Δεδομένων

Ο υπεύθυνος για τη διασφάλιση της ορθής, αξιόπιστης και αποδοτικής λειτουργίας, μιας βάσης δεδομένων ονομάζεται διαχειριστής Βάσης Δεδομένων ή Database Administrator (DBA). Κάποιες από τις αρμοδιότητες του είναι οι εξής:

- Επιλογή είδους της πληροφορίας που πρόκειται να αποθηκευτεί
- Επιλογή τρόπου αποθήκευσης και πρόσβασης σε αυτές τις πληροφορίες
- Επιτυχής συνεργασία με τον χρήστη
- Επιλογή τρόπου απόκτησης πληροφορίας
- Διατήρηση αντιγράφων ασφαλείας (backup)
- Επιτήρηση ορθής λειτουργίας της βάσης δεδομένων όπως και η απαιτούμενη προσαρμογή της σε πιθανά σφάλματα.

5.1.2 Ιστορία – Εξέλιξη βάσης δεδομένων

- **Δεκαετία του 1950**

Για την διαχείριση και επεξεργασία μεμονωμένων αρχείων χρησιμοποιούνται διάτρητες κάρτες και μαγνητικές ταινίες (Σειριακή αποθήκευση). Οι εξελίξεις στις συσκευές μαζικής αποθήκευσης τυχαίας πρόσβασης και αύξηση της υπολογιστικής ισχύος καθιστούν δυνατή την ανάπτυξη των συστημάτων διαχείρισης δεδομένων.

- **Δεκαετία του 1960**

Την δεκαετία του 1960 δημιουργούνται πρώτα συστήματα της διαχείρισης βάσεων δεδομένων (ΣΔΒΔ) στηριγμένα σε δικτυωτό μοντέλο –network data model- το οποίο δημιουργήθηκε από των Charles Bachman (1st Turing Award, 1973) της General Electric.

Το 1965 η εταιρία IBM σε συνεργασία με την Διεύθυνση Διαστήματος της North American Aviation ανέπτυξαν ιεραρχικό μοντέλο δεδομένων (hierarchical data model) το χαρακτηριστικό του οποίου ήταν ότι τα δεδομένα παριστάνουν ως δενδροειδείς δομές μέσα σε μια ιεραρχία εγγράφων. Το Σύστημα Διαχείρισης Πληροφοριών (Information Management System -IMS) της IBM που κυκλοφόρησε για πρώτη φορά στην αγορά το 1969, ήταν βασισμένο στο ιεραρχικό μοντέλο δεδομένων και παραμένει σε χρήση μέχρι και σήμερα.

- **Δεκαετία του 1970**

Εμφάνιση του Σχεσιακού Μοντέλου Δεδομένων (Relation Data Model) το 1970 από τον Δρ. Edgar Codd (Turing Award, 1981).

Αρχή ανάπτυξης ερευνητικών προγραμμάτων σχεσιακών συστημάτων System R (IBM) και INGRESS (Berkeley) καθώς και οι σχεσιακές γλώσσες SEQUEL, QBE και QUEL. Το 1979 ιδρύθηκε η εταιρία Relational Software Incorporated (RSI) η οποία κυκλοφόρησε στην αγορά τη σχεσιακή βάση δεδομένων Oracle v2.

- **Δεκαετία του 1980**

Ραγδαία αύξηση των εγκαταστάσεων των πρώτων συστημάτων Oracle, SQL Server, Informix, DB2. Αρχή εμφάνισης των σχεσιακών συστημάτων διαχείρισης βάσεων δεδομένων σε προσωπικούς υπολογιστές DBase 2 (με κωδική ονομασία Ashton Tate) και εξελίχτηκε μέχρι τις μέρες μας ως Paradox, Microsoft ACCESS.

- **Δεκαετία του 1990**

Εμφανίζονται τα πρώτα εμπορικά αντικειμενοστρεφή και αντικείμενο-σχεσιακά συστήματα ΒΔ. Η σύνδεση σε μια Βάση Δεδομένων καθίσταται δυνατή μέσω TCP/IP και κατ' επέκταση μέσω διαδικτύου.

5.2 Ιστορία της IBM DB2

Η DB2 έχει μια μακροχρόνια ιστορία και θεωρείται από ορισμένους ως το πρώτο προϊόν βάσης δεδομένων που χρησιμοποίησε τη γλώσσα ερωτημάτων SQL. Η IBM άλλωστε ξεκίνησε την ανάπτυξη της SQL η οποία αργότερα εξελίχθηκε σε πρότυπο. Παράλληλα η Oracle κυκλοφόρησε παράλληλα ένα εμπορικό προϊόν διαχείρισης βάσης δεδομένων SQL λίγο νωρίτερα από ό, τι έκανε η IBM.

Το όνομα DB2 δόθηκε για πρώτη φορά στο Σύστημα Διαχείρισης Βάσεων Δεδομένων(ΣΔΒΔ) το 1983 όταν η IBM κυκλοφόρησε την DB2 για το MVS mainframe platform. Πριν όμως από αυτό, κυκλοφόρησε ένα παρόμοιο προϊόν που ονομάστηκε SQL / DS για την εμπορική εικονική πλατφόρμα μηχανής της (VM).

Πηγαίνοντας ακόμα πιο πίσω στον χρόνο, η IBM στα μέσα της δεκαετίας του 1970 κυκλοφόρησε το QBE ως προϊόν σχεσιακής βάσης δεδομένων για την VM πλατφόρμα της, που περιείχε έναν πίνακα με κύριο προσανατολισμό "Ερώτημα Με Παράδειγμα", που παρήγαγε ένα γραμμικό συντακτικό της γλώσσας, το οποίο ήταν ένας αναγνωρίσιμος πρόδρομος της QBE (Question by example) και οδήγησε σε συναλλαγές με μια σχεσιακή βάση δεδομένων, έναν πρόδρομο της σημερινής χρήσης των βάσεων δεδομένων. Αργότερα εμφανίστηκε η QMF, χαρακτηριστικό της DB2 SQL, όπου παράγεται σε πραγματικό χρόνο και έφερε την ίδια εμφάνιση και αίσθηση με την QBE και DB2.

Όταν η IBM κυκλοφόρησε το πρώτο ολοκληρωμένο προϊόν σχεσιακής βάσης δεδομένων, ήθελε να έχει μια εμπορικής ποιότητας γλώσσα αιτημάτων , η οποία και στην συνέχεια μετονομάστηκε σε μια νέα γλώσσα (Structured Query Language) SQL για να την διαφοροποιήσει από την sequel που προϋπήρχε και εμφανίστηκε από την παραπάνω αναφερόμενη ομάδα. Η IBM έπειτα συνέχισε με το να δημιουργήσει τη διεπαφή (GUI) και να ενσωματώσει σε αυτή την SQL πλατφόρμα, η οποία απο μόνη της είχε ήδη αρχίσει να είναι σε χρήση από τα μέσα της δεκαετίας 1980 από τα μέχρι τότε προϊόντα της.

Με τα χρόνια η DB2 έχει αξιοποιηθεί όλο και περισσότερο και έχουν δρομολογηθεί πολλές βελτιώσεις υλικού, ιδιαίτερα σε IBM System z με χαρακτηριστικά όπως η παράλληλη ανταλλαγή δεδομένων, που θα την κάνει στην θεωρία ακόμα πιο γρήγορη.

Το 2009, ανακοινώθηκε ότι η MySQL μπορεί να αποθηκεύσει τα δεδομένα της σε IBM DB2 storage engine. Αυτό επιτρέπει στους χρήστες για το System IBM platform να έχουν πρόσβαση εγγενώς στο DB2 σύμφωνα με το σύστημα IBM (παλαιότερα ονομαζόταν OS/400), και για τους χρήστες που υπάρχουν σε άλλες πλατφόρμες να αποκτήσουν πρόσβαση σε αυτά τα αρχεία μέσα από το περιβάλλον της MySQL. Το System I και οι προκάτοχοί του, το AS/400 και το System/38, καθώς και η DB2 έχουν καλά ενσωματωθεί στο λειτουργικό σύστημα, και έρχονται, πια, ως μέρος του.

Βασικά Στοιχεία :

- Αυτοματοποιεί πολλές λειτουργίες του συστήματος, παρέχοντας στους διαχειριστές της βάσης ευελιξία και απαλλαγή από την εκτέλεση λειτουργιών καθημερινής ρουτίνας, κάτι το οποίο απαιτείται από άλλες βάσεις δεδομένων.
- Μείωση του όγκου των δεδομένων με χρήση της υψηλής συμπίεσης (deep compression) και ως συνέπεια την μείωση αποθηκευτικών συσκευών και της ποσότητας ηλεκτρικής ενέργειας.
- Με την λειτουργία DB2 pure scale (US), παρέχει μια clustered και shared-disk αρχιτεκτονική, για εύκολη επεκτασιμότητα και υψηλή διαθεσιμότητα (high availability). Με τη λειτουργία DB2 pure scale γίνεται εύκολη και προσιτή η δημιουργία συστοιχίας βάσεων.
- Οι επιδόσεις είναι κορυφαίες σε πολλαπλά φορτία, ως συνέπεια απαιτεί μικρότερη υπολογιστική ισχύ για να «τρέξει» η βάση δεδομένων, λιγότερες άδειες χρήσης και μικρότερα κόστη συντήρησης.
- Τον Οκτώβριο του 2010, η IBM ανακοίνωσε τη γενική διαθεσιμότητα της DB2 10 για z / OS. DB2 επεκτείνοντας την αξία του λογισμικού της που παραδίδεται στις επιχειρήσεις με ηγετική θέση στη βιομηχανία μέσω καινοτομιών σε τομείς-κλειδιά:
- Η ταχεία εφαρμογή και ανάπτυξη DATA WAREHOUSE για την επιχειρηματική ανάπτυξη
- Βελτιωμένη λειτουργική αποδοτικότητα για "out-of-the-box" DB2
- Ενισχυμένη επιχειρηματικών αναλύσεων και λύσεις απεικόνισης δεδομένων με QMF
- Αξεπέραστη ανθεκτικότητα για σημαντικές επιχειρηματικές πληροφορίες

5.3 Τα χαρακτηριστικά και οφέλη χρήσης DB2 10

Η IBM DB2 έχει κάποια χαρακτηριστικά που την κάνουν καλή επιλογή έναντι σε άλλες βάσεις δεδομένων. Αυτές είναι:

- Μειωμένη χρήση της CPU
- Από πέντε έως δέκα φορές περισσότερους χρήστες ταυτόχρονα σε ένα ενιαίο υποσύστημα, αποφεύγοντας περιορισμούς μνήμης
- Υποστηρίζει Σχεσιακή, Αντικειμενοσχεσιακή και XML (native) οργάνωση των δεδομένων.
- Υποστηρίζει εγγενώς την XQuery.
- Πρόσθετες αλλαγές σε απευθείας σύνδεση για τους ορισμούς των δεδομένων, τις υπηρεσίες και τα υποσυστήματα
- Παρέχει ασφάλεια με δυνατότητα υποδιαίρεσης των χρηστών σε επίπεδα και ομάδες.
- Qmf Classic Edition, για ένα προαιρετικό χαρακτηριστικό χρέωση, παρέχοντας μεγαλύτερη διαλειτουργικότητα με άλλα προγράμματα και βελτιωμένα ερωτήματα, φόρμες, τη διάγνωση, την απόδοση και τον έλεγχο των πόρων
- Qmf Enterprise Edition ένα βελτιωμένο μοντέλο ασφαλείας, υποστήριξη για HTML, PDF, ή Flash

Η IBM και η SAP έχουν συνεργαστεί πολύ στενά με DB2 10 για z / OS, έτσι και τώρα οι SAP χρήστες μπορούν να επωφεληθούν από τη δυνατότητα κλιμάκωσης της DB2 και τις βελτιώσεις που ωφελούν σημαντικά την απόδοση και επιτρέπουν την περαιτέρω ανάπτυξη των εφαρμογών SAP και την ενοποίηση του τοπικού υλικού ταυτόχρονα.

5.4 Τεχνικές πληροφορίες IBM DB2

Η DB2 μπορεί να προσπελασθεί είτε από την γραμμή εντολών είτε από ένα γραφικό περιβάλλον. Η διασύνδεση γραμμής εντολών απαιτεί περισσότερη γνώση του

προϊόντος. Το GUI είναι μια πολυ-πλατφόρμα Java που περιέχει μια ποικιλία από οδηγούς κατάλληλους για αρχάριους χρήστες. Η DB2 υποστηρίζει τόσο SQL και XQuery και έχει εγγενή εφαρμογή της αποθήκευσης δεδομένων XML, όπου τα δεδομένα αποθηκεύονται ως XML αρχείο (δεν είναι τόσο κοινό για την χρήση σχεσιακών δεδομένων ή δεδομένων CLOB) για την ταχύτερη πρόσβαση χρησιμοποιώντας XQuery.

Τέλος η DB2 παρέχει API και drivers για πολλές γλώσσες και πλατφόρμες προγραμματισμού όπως REXX, COBOL , RPG , FORTRAN , C + + , C , Delphi , .NET CLI , Java (μέσω JDBC) , Python , Perl , PHP , Ruby. Παρέχει επίσης UI εργαλεία για διαχείριση και χρήση ως plug in σε γνωστά IDE όπως το Eclipse, Netbeans και Visual Studio.

5.5 Επίλογος

Η IBM DB2 είναι από τα παλιότερα εργαλεία βάσης δεδομένων που υπάρχουν αυτή τη στιγμή, και εάν δεν ήταν τόσο ολοκληρωμένη όσο είναι τώρα θα είχε σβήσει πολύ νωρίτερα από τον χάρτη λόγω της αυξανόμενης πίεσης από τα καινούργια εργαλεία. Γενικά είναι μια πολύ δυνατή λύση στο πρόβλημα αποθήκευσης πληροφορίας και διαθέτει όλα τα παραπάνω με κατανοητό τρόπο με μοναδικά μειονεκτήματα την αδύναμη υποστήριξη Unicode χαρακτήρων που περνάνε μέσα από web services και την σχετικά βάρια εφαρμογή διαχείρισης Data Studio που ο ίδιος αντιμετώπισα. Παρόλα αυτά δεν θα υπάρξει κάποιο κώλυμα στην λειτουργία του προγράμματος εάν όλα έχουν στηθεί σωστά, και αυτό είναι που στην τελική κάνει ένα λογισμικό συνετό. Συνοψίζοντας, η DB2 κάλυψε απόλυτα τις ανάγκες της εργασίας που παρουσιάζεται εδώ και είναι ικανή να γίνει σταθερή βάση σε ακόμα μεγαλύτερα project.

6. WEB SERVICES

6.1 Εισαγωγή

Σε όλη την πτυχιακή εργασία γίνεται αναφορά για τα web services, το τί είναι ακριβώς, θα αναπτυχθεί στο συγκεκριμένο κεφάλαιο εκτενέστερα. Πιο συγκεκριμένα, θα παρουσιάσω όσο ποιο συνολικά γίνεται την τεχνολογία των web services από πλευράς αρχιτεκτονικής ενός προγράμματος, και το πώς αυτές μπορούν να χρησιμοποιηθούν στην πτυχιακή εργασία. Παράλληλα με αυτή την ανάλυση θα αναλυθούν και λεπτομερώς όλα τα βήματα που οδηγούν στην σωστή χρήση των υπηρεσιών αυτών.

6.2 Ιστορική αναδρομή

Τα τελευταία χρόνια έχει επικρατήσει στην αγορά ένα δοκιμασμένος μηχανισμός για την πρόσβαση, διαχείριση και ανταλλαγή δεδομένων τα οποία είναι Web Services.

Πολλά από τα web services που προσανατολίζονται στα δεδομένα (data oriented) έχουν μια και μοναδική λειτουργία: Μπορούν να εκτελούν ένα ερώτημα SQL σε μια βάση δεδομένων ή να καλούν μια stored procedure το οποίο ωθεί στο να ενσωματώνουμε όλες τις συνδιαλλαγές με την βάση δεδομένων μέσα στη λογική του κώδικα του προγράμματος. Έτσι, η υλοποίηση μιας εφαρμογής ιστού προσανατολισμένης στα δεδομένα αποτελεί πρόκληση. Συνδυάζει γνώσεις πάνω στις βάσεις δεδομένων, σε web προγραμματισμό αλλά και την αρμονική ζεύξη τους προκειμένου να παραχθεί ένα ολοκληρωμένο αποτέλεσμα με διακριτές λειτουργίες.

Πολλοί από τους προγραμματιστές που δημιουργούν data oriented εφαρμογές παραδέχονται ότι η διαδικασία δημιουργίας τους περιλαμβάνει πολλά DML (Data Manipulation language) ερωτήματα τα οποία πρέπει να τοποθετηθούν μέσα στον κώδικα. Επιπλέον, η δημιουργία μιας σύνδεσης με τη βάση, η εκτέλεση ενός DML, DDL ή SQL statement, η επεξεργασία του συνόλου των αποτελεσμάτων και ο τερματισμός της σύνδεσης αποτελεί μια επαναληπτική διαδικασία η οποία καταλαμβάνει πολλές γραμμές κώδικα και απαιτεί ιδιαίτερη προσοχή και αυτοσυγκέντρωση. Αυτό που θέλουμε να γίνεται είναι να μπορούμε να χρησιμοποιούμε τις λειτουργίες της βάσης

δεδομένων και να προβάλλουμε τα δεδομένα τους στο Web χωρίς να μας ταλαιπωρεί η δομή του προγράμματος που θα το υλοποιήσει. Για να επιτευχτεί ο στόχος αυτός, η IBM προτείνει τις Data Web Services με βάση τις οποίες δημιουργήσαμε το project της διαχείρισης μαθημάτων του τμήματος μας ΑΤΕΙ.

6.3 Ορισμός Web Services

Σύμφωνα με το W3C, «.. Υπηρεσία ιστού ονομάζεται ένα σύστημα λογισμικού που υποστηρίζει την διαλειτουργικότητα μεταξύ μηχανών οι οποίες επικοινωνούν μέσω δικτύου. Κάθε υπηρεσία ιστού έχει μια διεπαφή επικοινωνίας η οποία περιγράφεται από μια γλώσσα κατανοητή στα μηχανήματα και είναι γνωστή ως WSDL (web service description language). ..» Μια υπηρεσία ιστού μπορεί να κληθεί από ένα οποιοδήποτε υπολογιστικό σύστημα μέσω του πρωτοκόλλου SOAP ή επικοινωνίας REST ανταλλάσσοντας μηνύματα με βάση την WSDL περιγραφή της. Τα μηνύματα αυτά μεταφέρονται μέσω του HTTP, μορφοποιημένα σε XML σε συνδυασμό με άλλα πρότυπα στον παγκόσμιο ιστό (web).

Στην πράξη μια υπηρεσία ιστού είναι ένα πρόγραμμα γραμμένο σε κάποια γλώσσα ή πλατφόρμα προγραμματισμού (Java, .NET, PHP κ.ο.κ) το οποίο εκθέτει τις λειτουργίες του στον παγκόσμιο ιστό. Με τον όρο λειτουργίες εννοούμε τις μεθόδους που υλοποιεί και είναι προσβάσιμες μέσω πρωτοκόλλων SOAP ή της αρχιτεκτονικής REST. Μια απλή web service μπορεί να είναι μια εφαρμογή που μετατρέπει θερμοκρασία από βαθμούς Κελσίου σε βαθμούς Fahrenheit και το αντίστροφο. Για να την υλοποιήσουμε χρειαζόμαστε δύο μεθόδους μια που μετατρέπει σε βαθμούς Κελσίου και μια που μετατρέπει σε βαθμούς Fahrenheit. Έτσι έχουμε την web service με όνομα temperatureConverter που εκθέτει (exposes) προς ανάλωσιν δύο μεθόδους: την celsiusToFahrenheit και την fahrenheitToCelsius. Μπορείτε να δοκιμάσετε αυτήν την υπηρεσία στο <http://www.w3schools.com/webservices>

Ο κώδικας που υλοποιεί αυτή την απλή μετατροπή, γραμμένος σε Visual Basic .NET είναι ο ακόλουθος:

```
1. <%@ WebService Language="VBScript" Class="TempConvert" %>
2.
3. Imports System
4. Imports System.Web.Services
5.
6. Public Class TempConvert :Inherits WebService
7.
8. <WebMethod()> Public Function FahrenheitToCelsius
9. (ByVal Fahrenheit As String) As String
10.     dim fahr
11.     fahr=trim(replace(Fahrenheit,",","."))
12.     if fahr="" or IsNumeric (fahr)=false then return "Error"
13.     return (((fahr) - 32) / 9) * 5)
14. end function
15.
16. <WebMethod()> Public Function CelsiusToFahrenheit
17. (ByVal Celsius As String) As String
18.     dim cel
19.     cel=trim(replace(Celsius,",","."))
20.     if cel="" or IsNumeric(cel)=false then return "Error"
21.     return (((cel) * 9) / 5) + 32)
22. end function
23.
24. end class
```

Μια διαδικτυακή υπηρεσία αποτελείται από πολλές συσχετιζόμενες τεχνολογίες που τοποθετούνται σε διαφορετικό επίπεδο. Ξεκινώντας από κάτω προς τα πάνω: Αρχικά, απαιτείται ένα πρωτόκολλο για μεταφορά πληροφοριών μέσω δικτύου, όπως το http (HyperText Transfer Protocol), το SMTP (Simple Mail Transport Protocol) ή το FTP (File Transfer Protocol). Κάθε κλήση και απόκριση της υπηρεσίας θα πρέπει να «συσκευαστεί» σε ένα μήνυμα SOAP (Simple Object Access Protocol), το οποίο μπορεί να υφίσταται επεξεργασία από κάποιες επεκτάσεις SOAP (SOAP Extensions) πριν αποσταλεί από την αιτούντα υπηρεσία (request agent) και παραδοθεί στον πάροχο υπηρεσίας (provider agent) και αντίστροφα. Τα μηνύματα που ανταλλάσσονται και ο τρόπος που γίνεται η ανταλλαγή περιγράφονται λεπτομερώς στο αρχείο WSDL (Web Services Description Language). Επόμενο βήμα είναι η ανακάλυψη των υπηρεσιών, για την οποία υπάρχουν τρεις προσεγγίσεις : η ύπαρξη ενός καταλόγου καταγραφής και δημοσίευσης των υπηρεσιών με τη μορφή υπηρεσίας καταγραφής (registry), όπως το UDDI (Universal Description, Definition and Integration) και το DISCO (Microsoft Discovery), που αποτελεί την πιο διαδεδομένη λύση, η ύπαρξη ιστοσελίδων παραπομπής σε υπηρεσίες, τύπου <http://www.amazon.com> και πρότυπα SOAP, WSDL, UDDI και DISCO έχουν γραφεί σε γλώσσα επισήμανσης XML

(eXtensible Markup Language) και ενδεχομένως υπακούουν σε κάποιο έγγραφο DTD (Document Type Definition) ή XML Schema.

Σύμφωνα με τον ιστοτόπο MSDN της Microsoft οι web services παρουσιάζουν τα εξής τρία (3) κοινά χαρακτηριστικά :

- Οι Web Services εκθέτουν τη λειτουργικότητα τους σε χρήστες του διαδικτύου μέσα από ένα δικτυακό πρωτόκολλο. Στις περισσότερες περιπτώσεις αυτό το πρωτόκολλο είναι το SOAP (Simple Object Access Protocol) το οποίο έχει προτυποποιηθεί από το W3C.
- Οι Web Services παρέχουν ένα τρόπο να περιγράψουν τις διεπαφές τους με αρκετή λεπτομέρεια ώστε να επιτρέψουν στο χρήστη τους να χτίσει μια εφαρμογή πελάτη η οποία να επικοινωνήσει μαζί τους. Η περιγραφή συνήθως παρέχεται σε ένα έγγραφο XML το οποίο ονομάζεται έγγραφο WSDL (Web Services Description Language).
- Οι Web Services καταχωρούνται ώστε οι δυνητικοί χρήστες να μπορούν να τις ανακαλύψουν και να χρησιμοποιήσουν εύκολα. Αυτό γίνεται με το UDDI (Universal Discovery Description and Integration).

Οι Web Services λοιπόν αποτελούν μία αρχιτεκτονική κατανεμημένων συστημάτων κατασκευασμένη από ετερογενή υπολογιστικά συστήματα, τα οποία επικοινωνούν μέσω του δικτύου ώστε να δημιουργήσουν ένα σύστημα. Αποτελούνται από ένα σύνολο προτύπων τα οποία επιτρέπουν στους προγραμματιστές (developers) να υλοποιήσουν κατανεμημένες εφαρμογές χρησιμοποιώντας εργαλεία από διαφορετικούς προμηθευτές - ώστε να κατασκευάσουν εφαρμογές που χρησιμοποιούν ένα συνδυασμό από ενότητες λογισμικού (software modules) οι οποίες καλούνται από συστήματα που ανήκουν σε διαφορετικά τμήματα ενός οργανισμού ή σε διαφορετικούς οργανισμούς.

6.4 Οι ανάγκες που οδήγησαν στις υπηρεσίες ιστού

Στην δεκαετία του '90 οι επιχειρήσεις προκειμένου να περικόψουν έξοδα που αφορούσαν σε μηχανογραφικό εξοπλισμό, υπαλλήλους και υλοποίηση εφαρμογών που επιλύουν θέματα ή εκτελούν διαδικασίες, άρχισαν να αναθέτουν μέρος ή ολόκληρες τις διαδικασίες σε εξειδικευμένους εξωτερικούς συνεργάτες. Οι συνεργάτες οι οποίοι ονομάστηκαν πάροχοι υπηρεσιών εφαρμογών (application service providers) αναλαμβάνουν την περαίωση των δραστηριοτήτων και διαδικασιών που αφορούν τις επιχειρήσεις – πελάτες τους με την παραγωγή προγραμμάτων – υπηρεσιών για αυτές. Με την εξάπλωση του παγκοσμίου ιστού και την δημιουργία των standards για τα πρωτόκολλα επικοινωνίας (όπως το HTTP) τα προγράμματα – υπηρεσίες έγιναν διαθέσιμα σε ένα συγκεκριμένο ακραίο κόμβο του διαδικτύου. Με αυτόν τον τρόπο άρχισε να υφαίνεται η έννοια της αρχιτεκτονικής υπηρεσιών ιστού (SOA: Service Oriented Architecture).

Για την επίτευξη της επικοινωνίας μέσω υπηρεσιών, οι οποίες ενδεχομένως εντοπίζονται δυναμικά, απαιτείται χαλαρή σύζευξη (loose coupling) μεταξύ εκείνου που κάνει την αίτηση και της υπηρεσίας που θα χρησιμοποιηθεί. Η έννοια της χαλαρής σύζευξης αποκλείει οποιαδήποτε γνώση ή υποθέσεις σχετικά με τις συγκεκριμένες πλατφόρμες όπου εκτελείται ο αιτών ή ο πάροχος υπηρεσιών, την υλοποίηση που χρησιμοποιεί ο εκάστοτε πάροχος, ή τις μορφές και τα πρωτόκολλα που χρησιμοποιούν για τη μεταξύ τους επικοινωνία.

6.5 Χαρακτηριστικά

Οι υπηρεσίες ιστού έχουν κάποια βασικά στοιχεία τα οποία τις χαρακτηρίζουν:

- Οι υπηρεσίες είναι οντότητες λογισμικού διαθέσιμες σε ένα συγκεκριμένο ακραίο σημείο-κόμβο του δικτύου (ένας server). Λαμβάνουν και στέλνουν μηνύματα σύμφωνα με τη συμπεριφορά που έχουν ορίσει στις προδιαγραφές τους μέσω της WSDL (web service description language).

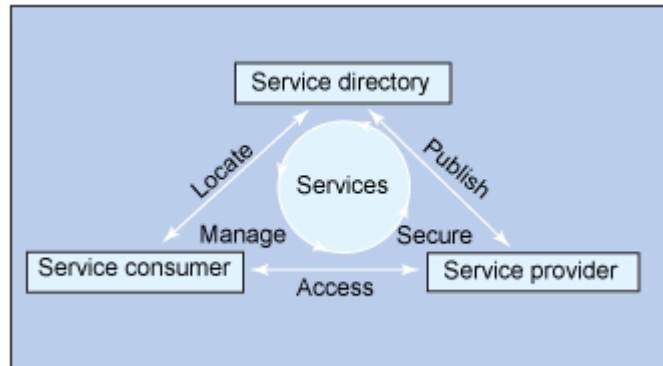
- Οι υπηρεσίες κρύβουν μέσα τους το μοντέλο προγραμματισμού και τις λεπτομέρειες της υλοποίησης τους. Εκθέτουν μόνο ότι είναι χρήσιμο στους πελάτες για να τις καλέσουν και να τις χρησιμοποιήσουν.
- Οι υπηρεσίες ιστού μπορούν να κληθούν από οποιαδήποτε πλατφόρμα λογισμικού ή υλικού. Τα πρωτόκολλα επικοινωνίας τους αποτελούν πρότυπα που υλοποιούνται από όλους τους κατασκευαστές. Έτσι όταν καλούμε μια web service δεν γνωρίζουμε τι λειτουργικό σύστημα κρύβεται από πίσω αλλά ούτε μας ενδιαφέρει αν το λειτουργικό μας σύστημα είναι συμβατό.
- Οι πληροφορίες που δημοσιεύονται για μια υπηρεσία μέσω της WSDL (web service description language) αρκούν για την περιγραφή της και τον τρόπο ανάληψης της (αίτηση προς αυτήν, λήψη απάντησης, δικλείδες ασφαλείας). Όλα αυτά ισχύουν υπό την προϋπόθεση ότι ο αιτών γνωρίζει πώς να τις χρησιμοποιήσει.
- Μια υπηρεσία ιστού μπορεί να ενσωματώσει ή να χρησιμοποιεί άλλες υπηρεσίες.
- Κάθε υπηρεσία ιστού έχει υποδοχείς (endpoints) οι οποίοι δέχονται τα μηνύματα των αιτήσεων προς την υπηρεσία και αναλαμβάνουν να στείλουν τις απαντήσεις στους πελάτες. Τα endpoints είναι δύο κατηγοριών, SOAP αν χρησιμοποιείται το πρωτόκολλο soap για την επικοινωνία ή REST HTTP GET ή POST αν η υπηρεσία χρησιμοποιείται σαν πόρος του δικτύου.

6.6 Αρχιτεκτονική υπηρεσιών ιστού

Η αρχιτεκτονική υπηρεσιών ιστού ή αρχιτεκτονική προσανατολισμένη στις υπηρεσίες (SOA) αποτελεί ένα συγκεκριμένο αρχιτεκτονικό ύφος το οποίο έχει ως στόχο τη χαλαρή σύνδεση και δυναμική σύζευξη μεταξύ των υπηρεσιών. Για να λειτουργήσει αποδοτικά η SOA πρέπει να διαθέτει ορισμένους συγκεκριμένους παράγοντες.

Καταρχάς είναι απαραίτητο να παρασχεθεί ένας αφηρημένος ορισμός της υπηρεσίας, ο οποίος θα συμπεριλαμβάνει τις λεπτομέρειες που επιτρέπουν σε οποιονδήποτε θέλει να χρησιμοποιήσει την υπηρεσία να συνδεθεί (bind) καταλλήλως με εκείνη. Δεύτερον οι πάροχοι των υπηρεσιών πρέπει να δημοσιεύουν (publish) τις λεπτομέρειες των υπηρεσιών έτσι ώστε οι πελάτες τους να μπορούν να κατανοήσουν τον σκοπό, τον τρόπο χρήσης και σύνδεσης με την υπηρεσία. Τρίτον, εκείνη που ζητούν υπηρεσίες,

πρέπει να έχουν κάποιο τρόπο να εντοπίζουν (find) ποιες υπηρεσίες είναι διαθέσιμες και να εξυπηρετούν τις ανάγκες τους. Για να λειτουργήσει η προσέγγιση «σύνδεση/δημοσίευση/εντοπισμός» πρέπει να καθοριστούν τα πρότυπα τα οποία διέπουν τι θα δημοσιεύεται και πώς, τον τρόπο που θα εντοπίζονται οι πληροφορίες καθώς και τον τρόπο σύνδεσης σε αυτές.



Εικόνα 6.1 Η προσέγγιση σύνδεση/δημοσίευση/εντοπισμός

Ο πάροχος μιας υπηρεσίας περιγράφει τη σημασιολογία της, δηλαδή τις λειτουργίες που υποστηρίζει και τα χαρακτηριστικά σχετικά με τον τρόπο προσπέλασής της σε ένα έγγραφο WSDL (web service description language). Οι πάροχοι των υπηρεσιών δημοσιεύουν τα wsdl έγγραφα των υπηρεσιών τους σε καταλόγους υπηρεσιών ιστού UDDI όπου οι επιχειρήσεις μπορούν να εγγραφούν και να αναζητούν υπηρεσίες ιστού ανάλογα με τις ανάγκες τους.

6.7 Είδη interface Υπηρεσιών

Οι υπηρεσίες ιστού χωρίζονται σε δύο μεγάλες κατηγορίες ανάλογα με την αρχιτεκτονική υλοποίησης που χρησιμοποιούν ως προς τον τρόπο επικοινωνίας με τους πελάτες τους. Η έννοια τρόπος επικοινωνίας συχνά συναντάται στη βιβλιογραφία ως το interface ή τα endpoints της υπηρεσίας ιστού. Η υπηρεσία μπορεί να είναι μία αλλά να έχει παραπάνω από μία υλοποίηση ως προς τον τρόπο επικοινωνίας. Υπάρχουν οι υπηρεσίες που επικοινωνούν μέσω του πρωτοκόλλου SOAP και οι

υπηρεσίες που υλοποιούν τα πρότυπα της αρχιτεκτονικής REST στην οποία βασίζεται ολόκληρος ο παγκόσμιος ιστός. Μια υπηρεσία μπορεί να επικοινωνεί με έναν από τους δύο τρόπους ή και με τους δύο. Αυτό εναπόκειται στην ευχέρεια του προγραμματιστή και τις ανάγκες των πελατών του. Στις επόμενες δύο ενότητες εξηγούμε τα βασικά χαρακτηριστικά του πρωτοκόλλου SOAP και της αρχιτεκτονικής REST.

6.8 SOAP

Το πρωτόκολλο SOAP (το οποίο ξεκίνησε ως ακρώνυμο: Simple Object Access Protocol) παρέχει ένα πλαίσιο επικοινωνίας μεταξύ κόμβων (SOAP nodes) στον παγκόσμιο ιστό μέσω ανταλλαγής μηνυμάτων σε XML. Τέτοιοι κόμβοι μπορούν να είναι οι Υπηρεσίες Ιστού οι οποίες μπορούν να επικοινωνούν μέσω του πρωτοκόλλου SOAP. Ένα μήνυμα που αποστέλλεται από έναν Αρχικό Αποστολέα κόμβο (initial sender) σε έναν Τελικό Παραλήπτη κόμβο (ultimate receiver) μπορεί να παραληφθεί, επεξεργαστεί και προωθηθεί από έναν αριθμό Ενδιάμεσων κόμβων (intermediaries).

Το SOAP [14] είναι αποτέλεσμα τυποποίησης από τον οργανισμό W3C (World Wide Web Consortium <http://www.w3.org>).

Το SOAP αναφέρεται κυρίως ως ένα πρωτόκολλο που επιτρέπει την ανταλλαγή μηνυμάτων μονόδρομα (one-way), αλλά η υποστήριξη επιπλέον μοντέλων ανταλλαγής μηνυμάτων όπως αμφίδρομη επικοινωνία (request/response) μπορεί να υποστηριχθεί (α) σε επίπεδο εφαρμογής που κάνει χρήση του SOAP ή (β) σε επίπεδο πρωτοκόλλου που υλοποιεί τη μεταφορά των μηνυμάτων SOAP (transport).

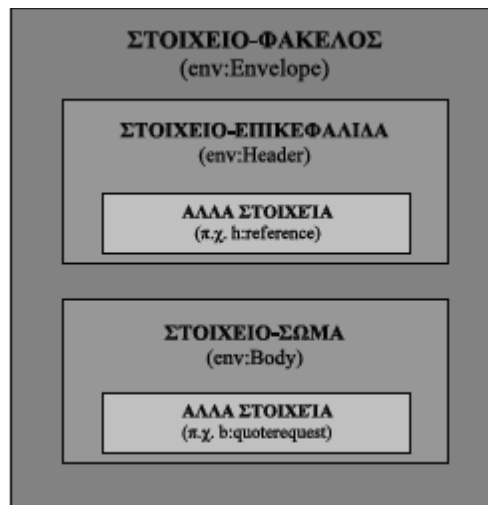
Τα μηνύματα που ανταλλάσσονται στα πλαίσια του SOAP προβλέπεται να έχουν συγκεκριμένη δομή (message structure) σύμφωνα με το πρότυπο όπως επίσης προβλέπεται ένα διεργασιακό μοντέλο (processing model) δηλ. ένας συγκεκριμένος τρόπος επεξεργασίας μηνυμάτων από τους κόμβους που εμπλέκονται στην ανταλλαγή ή τη μετάδοσή τους. Επίσης προβλέπεται η μεταφορά αυτών των μηνυμάτων χρησιμοποιώντας πρωτόκολλα όπως το HTTP ή το SMTP αν και προς το παρόν έχουν τυποποιηθεί οι προδιαγραφές για μεταφορά μέσω HTTP μόνο.

Εν συντομία τα βασικά χαρακτηριστικά του SOAP είναι τα εξής:

- Είναι ένα πρωτόκολλο για την επικοινωνία με web services
- Ανήκει στο application layer του OSI
- Χρησιμοποιεί μηνύματα γραμμένα σε XML
- Χρησιμοποιεί το πρωτόκολλο HTTP ως φέρον
- Είναι ανεξάρτητο πλατφόρμας λογισμικού
- Είναι ανεξάρτητο γλώσσας προγραμματισμού
- Είναι απλό και επεκτάσιμο
- Αποτελεί πρότυπο του W3C

Η δομή του έχει ως εξής:

Το μήνυμα SOAP έχει (Εικόνα 6.2) ως στοιχείο ρίζα τον φάκελο (envelope), ο οποίος περιέχει μια κεφαλίδα (header) με μεταδεδομένα και ένα σώμα (body) με τις κλήσεις των μεθόδων μιας web service και τις παραμέτρους αν είναι μήνυμα αίτησης (request) ή την απάντηση από αυτήν αν είναι μήνυμα απάντησης (response message).



Εικόνα 6.2 Η δομή ενός μηνύματος SOAP

Η κεφαλίδα είναι ένα προαιρετικό στοιχείο και περιέχει πληροφορίες σχετικά με την εφαρμογή που θα εκτελεστεί, όπως κρυπτογράφηση, σύνδεση μέσω λογαριασμού (authentication).

Το σώμα μπορεί να περιέχει κλήση προς μεθόδους της web service αν το μήνυμα είναι αίτηση προς τη web service. Στο παρακάτω κομμάτι κώδικα έχουμε το μήνυμα αίτησης προς την data web service “Τμήματα” το οποίο ζητά την κλήση της μεθόδου GETKathigites (γραμμή 3) η οποία δέχεται ως όρισμα ένα string (σαν παράμετρος) και εμείς περνούμε σε αυτό το string την τιμή «PLHROFORIKH» (γραμμή 4). Στην πράξη, ζητούμε από την web service να μας εμφανίσει όλους τους καθηγητές του τμήματος Πληροφορικής.

```

1.<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:q0="http://example.com/tmhmata"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
2.   <soapenv:Body>
3.     <q0:GETKathigites>
4.       <TMHMA>PLHROFORIKH</TMHMA>
5.     </q0: GETKathigites >
6.   </soapenv:Body>
</soapenv:Envelope>

```

Το σώμα ενός μηνύματος SOAP μπορεί να περιέχει την απάντηση από την web service. Στο παρακάτω κομμάτι κώδικα βλέπουμε την απάντηση της data web service.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <GETKATHIGITESResponse xmlns:ns1="http://web.GETKATHIGITES">
      <rowset>
        <row>
          <NAME>Βασίλειος</NAME>
          <SURNAME>Βίτσας</SURNAME>
          <TOMEAS>ΣΥΣΤ & ΤΕΧΝΟΛ</TOMEAS>
        </row>
        <row>
          <NAME>Αλεξάνδρα</NAME>
          <SURNAME>Παπαδοπούλου</SURNAME>
          <TOMEAS>ΑΝΑΛ & ΠΡΟΓΡ</TOMEAS>
        </row>
      </rowset>
    </GETKATHIGITESResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

Παρατηρούμε ότι τόσο η αίτηση, όσο και η απάντηση είναι μορφοποιημένες σε XML, κάτι το οποίο μας βοηθά στην κλήση του. Οι περισσότερες αντικειμενοστρεφείς γλώσσες προγραμματισμού (Java, C++, C#, Visual Basic, PHP, Ruby) έχουν

υλοποιήσει κλάσεις προκειμένου να μπορούν να στέλνουν και να λαμβάνουν μηνύματα SOAP μέσω του HTTP πρωτοκόλλου.

6.8.1 Διεργασιακό Μοντέλο (processing model)

Ένα άλλο παράδειγμα μηνύματος SOAP.

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
<env:Header>
  <h:reference xmlns:h="http://atei.example.com/atei_reference
    env:mustUnderstand="true">
    <h:number> 123456 </h:number>
  </h:reference>
</env:Header>
  <env:Body>
    <b:quoteresponse xmlns:b="http://atei.example.com/atei_stoixeia">
      <b:surname_exchange> Βίτσας </b:surname_exchange>
      <b:name> Βασίλειος </b:name>
    </b:quoteresponse>
  </env:Body>
</env:Envelope>
```

Αυτό το μήνυμα αποστέλλεται σε μια υπηρεσία ιστού την `atei.example.com/atei_reference` η οποία για έναν συγκεκριμένο καθηγητή (Βίτσας Βασίλειος) επιστρέφει το τηλέφωνο του. Οι παράμετροι που απαιτούνται για αυτήν τη λειτουργία δίνονται στο στοιχείο-σώμα του μηνύματος (`env:Body`). Η πληροφορία στο στοιχείο-επικεφαλίδα είναι ένα συγκεκριμένο νούμερο αναφοράς (`reference number`) το οποίο πιστοποιεί τον πελάτη ή χρήστη που καλεί την υπηρεσία ιστού και επιτρέπει στην υπηρεσία, με βάση αυτό το νούμερο, να δώσει την απάντηση.

Το W3C έχει τυποποιήσει συγκεκριμένους SOAP ρόλους που μπορεί να παίξει κάποιος κόμβος όταν λάβει ένα μήνυμα SOAP. Ένας τέτοιος «κόμβος» μπορεί να είναι μία υπηρεσία ιστού.

Ένας κόμβος SOAP μπορεί να είναι ο Αρχικός Αποστολέας (`initial sender`), ο Τελικός Παραλήπτης (`ultimate receiver`) ή ένας Ενδιάμεσος (`intermediary`) προκειμένου να γίνει η αποστολή και πιθανή επεξεργασία ενός μηνύματος SOAP που εστάλη από τον αρχικό αποστολέα μέχρι να φθάσει στον τελικό παραλήπτη. Τα στοιχεία που βρίσκονται στο στοιχείο-κεφαλίδα μπορούν να χρησιμοποιήσουν μία από τις τρεις τυποποιημένες τιμές

που μπορεί να έχει η τυποποιημένη ιδιότητα `env:role` προκειμένου να υποδείξουν στους Ενδιάμεσους και στον Τελικό Παραλήπτη αν και πως επιτρέπεται να επεξεργαστούν το συγκεκριμένο στοιχείο. Έτσι, μπορεί να ζητηθεί η επεξεργασία του στοιχείου αυτού μόνο από τον τελικό παραλήπτη, από τους ενδιάμεσους και τον τελικό παραλήπτη ή από κανένα.

Η προεπιλεγμένη συμπεριφορά ενός κόμβου που επεξεργάζεται κάποιο στοιχείο επικεφαλίδας ενός SOAP μηνύματος είναι να αφαιρέσει το συγκεκριμένο στοιχείο από την επικεφαλίδα πριν προωθήσει το μήνυμα στον επόμενο κόμβο. Διατηρείται όμως η δυνατότητα να επαναπροσθέσει το στοιχείο που επεξεργάστηκε πριν την προώθηση. Υπάρχουν όμως περιπτώσεις όπου θέλουμε να σταλεί ένα μήνυμα SOAP σε έναν αριθμό κόμβων και μόνον όσοι επιλέγουν να το επεξεργαστούν να αφαιρέσουν το αντίστοιχο στοιχείο επικεφαλίδας αλλιώς να το προωθήσουν αναλλοίωτο. Σε αυτές τις περιπτώσεις μπορεί να γίνει χρήση της `env:relay` ιδιότητας η οποία μπορεί να τεθεί σε κατάσταση 'true'. Εάν όμως παράλληλα γίνει χρήση της ιδιότητας `env:mustUnderstand` σε κατάσταση 'true' ένας κόμβος είναι υποχρεωμένος να επεξεργαστεί ένα στοιχείο επικεφαλίδας εάν το κατανοεί. Σε απάντηση του μηνύματος που δίνεται στο πιο πάνω παράδειγμα ο τελικός παραλήπτης μπορεί να απαντήσει με ένα άλλο SOAP μήνυμα όπως το εξής:

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <h:reference xmlns:h="http://atei.example.com/atei_reference"
    env:mustUnderstand="true">
    <h:number> 123456 </h:numberE>
  </h:reference>
</env:Header>
  <env:Body>
    <b:quoteresponse xmlns:b="http://atei.example.com/atei_stoixeia">
      <b:surname_exchange> Βίτσας </b:surname_exchange>
      <b:name> Βασίλειος </b:name>
      <b:phone> 699-99-99999 </b:phone>
      <b:eidos> Mobile </b:eidos>
    </b:quoteresponse>
  </env:Body>
</env:Envelope>
```

Αυτό το μήνυμα βλέπουμε ότι περιλαμβάνει τα ίδια στοιχεία στο στοιχείο-επικεφαλίδα και επιτρέπει το συσχετισμό του αρχικού μηνύματος με το απαντητικό μήνυμα με βάση το νούμερο αναφοράς (`h: reference 123456`). Βλέπουμε επίσης ότι μαζί με τα στοιχεία του καθηγητή επιστρέφεται το τηλέφωνο του και το είδος σε οποίο το νούμερο αυτό ανήκει.

6.8.2 Δομή Μηνυμάτων Σφάλματος

Σε κάθε εκτέλεση κώδικα μπορούν να προκύψουν ανεπιθύμητες καταστάσεις όπως σφάλματα ή Exceptions που μπορούν να οφείλονται σε σφάλμα υλικού, δικτύου ή κάποια λανθασμένη τιμή παραμέτρου. Το πρωτόκολλο προβλέπει συγκεκριμένη δομή και για τα μηνύματα σφάλματος τα οποία μπορούν να σημειωθούν κατά την επεξεργασία του στοιχείου-επικεφαλίδα ή του στοιχείου-σώμα ενός μηνύματος SOAP από τον τελικό παραλήπτη ή τους ενδιάμεσους. Τα μηνύματα είναι γνωστά και ως SOAP Fault. Το κανάλι επικοινωνίας για την γνωστοποίηση του σφάλματος μπορεί να είναι το κανονικό κανάλι που διατίθεται από το πρωτόκολλο μεταφοράς ή το κανάλι σφάλματος του πρωτοκόλλου αναφοράς. Τα μηνύματα σφάλματος επεξεργασίας του στοιχείου-σώματος περιλαμβάνουν ένα στοιχείο-σφάλμα (env:Fault) στο στοιχείο-σώμα του απαντητικού SOAP μηνύματος το οποίο περιέχει τυποποιημένα στοιχεία τα οποία καθορίζουν τον κωδικό λάθους (env:Code), την αιτία (env:Reason) και περαιτέρω λεπτομέρειες (env:Detail).

Μηνύματα σφάλματος που αφορούν την επεξεργασία στοιχείων του στοιχείου-επικεφαλίδα περιέχουν ένα στοιχείο-σφάλμα (env:Fault) πάλι στο στοιχείο-σώμα του μηνύματος σφάλματος αλλά επίσης υποδεικνύουν το στοιχείο του στοιχείου-επικεφαλίδα που προκάλεσε το σφάλμα.

6.9 REST

Μία εναλλακτική υλοποίηση έναντι του πρωτοκόλλου SOAP είναι η υλοποίηση των Restful web services. Η βασική έννοια στην αρχιτεκτονική REST (Representational State Transfer) είναι ο **πόρος**. Ένας πόρος είναι ένα οποιοδήποτε κομμάτι πληροφορίας, το οποίο μπορεί να προσδιοριστεί με μοναδικό τρόπο. Στην αρχιτεκτονική REST, οι αιτούντες και οι υπηρεσίες ανταλλάσσουν μηνύματα τα οποία μπορούν να έχουν τόσο δεδομένα όσο και μεταδεδομένα. Για να αναπαρασταθεί ένας πόρος με μοναδικό τρόπο χρησιμοποιούμε URIs. Για παράδειγμα ένας πόρος μπορεί να είναι ο <http://www.go.gr/kilometers/count>.

Οι υπηρεσίες που είναι υλοποιημένες με REST δεν διατηρούν την κατάσταση μιας αλληλεπίδρασης (stateless). Αυτό βοηθά τη γρήγορη ανάκαμψή τους σε περίπτωση

σφάλματος. Σε περίπτωση που η υπηρεσία που βρίσκεται στο uri <http://www.go.gr/kilometers/count> παρουσιάσει σφάλμα ενώ την καλούμε, μπορούμε να την ξανακαλέσουμε χωρίς να μας περιορίσει κάποιος. Το γεγονός ότι οι RESTful web services δεν διατηρούν την κατάσταση της αλληλεπίδρασης πελάτη – υπηρεσίας, μας υποχρεώνει να αποστέλλουμε ότι είναι απαραίτητο στην κάθε αίτηση προς αυτές. Κάθε μήνυμα πρέπει να είναι αυτοπεριγραφικό, δηλαδή να περιέχει όλες τις πληροφορίες που χρειάζεται η web service για να εκτελεστεί.

Τα βασικά χαρακτηριστικά των υπηρεσιών REST είναι τα εξής:

- Η web service είναι προσβάσιμη μέσω του πρωτοκόλλου HTTP με τις μεθόδους GET ή POST ή PUT ή DELETE.
- Κάθε web service έχει ένα μοναδικό base uri στο οποίο προστίθενται στο τέλος το όνομα της μεθόδου που θέλουμε να καλέσουμε, μαζί με τις παραμέτρους της.
- Κάθε web service υποστηρίζει δεδομένα τα οποία είναι κωδικοποιημένα με βάση ένα MIME type (JSON, XML, YAML ή άλλο).

Η χρήση των RESTful web services έχει αυξηθεί δραματικά εξαιτίας του γεγονότος ότι είναι απλές. Αρκεί μόνο το URL της υπηρεσίας και η γνώση των παραμέτρων που πρέπει να περαστούν σε αυτήν. Μια υπηρεσία με REST endpoint με μέθοδο HTTP GET μπορεί να κληθεί ακόμη και από έναν web browser. Η απλότητα της την έχουν κάνει τόσο δημοφιλή ώστε μεγάλες εταιρίες έχουν δημιουργήσει REST endpoints για τις ήδη υπάρχουσες web services που υποστήριζαν μόνο SOAP.

Η Google έχει αναπτύξει το AJAX API το οποίο είναι μια RESTful υλοποίηση των προϊόντων της. Μέσω αυτού μπορούμε να χρησιμοποιήσουμε υπηρεσίες όπως το Google Search, Google Maps, Google Translate, Google Ajax Feed και άλλες.

6.10 XML

Η γλώσσα Extensible Markup Language (XML) είναι ένα πρότυπο για τη δόμηση και την ανταλλαγή δεδομένων μέσω ετερογενών συσκευών, πλατφορμών και λειτουργικών

συστημάτων. Αποτελεί πρότυπο του W3C και χρησιμοποιείται ευρέως με κύρια χρήση στον παγκόσμιο ιστό. Δημιουργήθηκε με σκοπό να παρέχει έναν κοινό τρόπο ανταλλαγής δεδομένων μεταξύ διαφορετικών συσκευών που βασίζονται σε διαφορετικά λειτουργικά συστήματα..

Οι web services χρησιμοποιούν την XML για τη μεταφορά δεδομένων. Το πρωτόκολλο SOAP είναι ένα πρωτόκολλο που βασίζεται και δομείται με XML. Οι REST style web services χρησιμοποιούν την XML ή τη JSON για την μορφοποίηση των δεδομένων εντός του HTTP πακέτου.

6.10.1 Το ιεραρχικό μοντέλο της XML

Η γλώσσα XML χρησιμοποιείται για να περιγράψει τη δομή που έχουν τα δεδομένα και κατ' επέκτασιν τα δεδομένα που περιέχονται σε έγγραφα. Δεν έχει καμία σχέση με τον τρόπο αναπαράστασης ενός εγγράφου, το οποίο είναι δουλειά HTML. Η ευρεία χρήση της έχει οδηγήσει στην αποθήκευση XML εγγράφων μέσα σε βάσεις δεδομένων και τη χρήση ερωτημάτων πάνω σε αυτά μέσω της γλώσσας αιτημάτων XQuery.

“.. Βασική μονάδα δόμησης ενός XML εγγράφου είναι το element (στοιχείο) το οποίο αποτελείται από ετικέτες (tags) και οι ιδιότητες - γνωρίσματα (attributes). Οι ετικέτες μπορούν να έχουν οποιοδήποτε όνομα θέλουμε. Με αυτόν τον τρόπο μπορούμε να δημιουργήσουμε όποια ετικέτα θέλουμε, γι' αυτό και η XML χαρακτηρίζεται ως αυτοπεριγραφόμενη. ..” Στον αντίποδα, η HTML χρησιμοποιεί μόνο έναν πεπερασμένο αριθμό ετικετών που έχουν οριστεί από το W3C.

- Κάθε στοιχείο της XML αποτελείται από μια ετικέτα αρχής, τις ιδιότητές της (αν υπάρχουν) και την ετικέτα τέλους.
- Τα ονόματα των ετικετών αρχής περιβάλλονται από γωνιακές αγκύλες (π.χ. <ptyxiaki>)
- Τα ονόματα των ετικετών τέλους περιβάλλονται από γωνιακές αγκύλες και μια slash (π.χ. </ptyxiaki>)
- Τα σχόλια περιβάλλονται από τα σύμβολα <!--και --> όπως και στην HTML.
- Τα πολύπλοκα στοιχεία κατασκευάζονται από άλλα στοιχεία, ιεραρχικώ τω τρόπω.
- Τα απλά στοιχεία περιέχουν τιμές δεδομένων (αριθμοί, strings, Boolean values κλπ)

- Στην XML τα ονόματα των ετικετών περιγράφουν τη σημασία των δεδομένων (π.χ. <teaches>), ενώ στην HTML κάθε στοιχείο περιγράφει τη μορφή εμφάνισης του κειμένου.
- Το μοντέλο που υλοποιεί η XML ονομάζεται ιεραρχικό ή δενδρικό.

Η XML έχει τα εξής χαρακτηριστικά:

- Είναι case-sensitive γι' αυτό η ετικέτα αρχής και τέλους πρέπει να γραμμένες ακριβώς με τον ίδιο τρόπο. Το <Name> είναι διαφορετική ετικέτα από το <name>.
- Οι τιμές κειμένου των στοιχείων (strings) δεν περικλείονται από εισαγωγικά.
- Οι τιμές των ιδιοτήτων (attribute) περικλείονται από μονά (π.χ. 'velos') ή διπλά εισαγωγικά (π.χ. "velos")

6.10.2 Ορθώς ορισμένο XML έγγραφο

Ένα ορθώς ορισμένο (well formed) XML θα πρέπει να ακολουθεί τις ακόλουθες συνθήκες:

- Να αναφέρει στην πρώτη γραμμή την έκδοση της XML που ακολουθεί.
- Να ξεκαθαρίζει με το γνώρισμα standalone αν το έγγραφο ακολουθεί κάποιο schema ή όχι.
- standalone="yes" δεν ακολουθεί κάποιο σχήμα
- standalone="no" ακολουθεί συγκεκριμένο σχήμα
- Θα πρέπει να ακολουθεί το συντακτικό του δενδρικού μοντέλου:
- Θα πρέπει να υπάρχει ένα στοιχείο (element) ρίζα
- Κάθε στοιχείο να περιλαμβάνει ένα ζεύγος ετικετών αρχής και τέλους
- Κάθε στοιχείο να περικλείεται μέσα στις ετικέτες του κόμβου γονέα.
- Δεν υπάρχει προκαθορισμένο σύνολο στοιχείων.

6.10.3 Έγκυρο XML έγγραφο

Ένα XML έγγραφο είναι έγκυρο (valid) όταν:

- Είναι ορθώς ορισμένο (well formed)
- Τα ονόματα των στοιχείων του ακολουθούν κάποιο XML schema.

6.10.4 XML schema

Ένα XML schema (σχήμα) είναι ένα πρότυπο για τον προσδιορισμό της δομής των XML εγγράφων. Κάθε σχήμα είναι γραμμένο σε XML και αποτελεί ένα έγγραφο XML. Κάθε μήνυμα αίτησης ή απάντησης του πρωτοκόλλου SOAP ακολουθεί ένα schema τα οποίο έχει οριστεί από το W3C.

6.11 JSON

Η JSON ή αλλιώς JavaScript Object Notation αποτελεί ένα «ελαφρύ» πρότυπο ανταλλαγής δεδομένων μεταξύ γλωσσών προγραμματισμού. Βασίζεται σε ένα υποσύνολο της γλώσσας JavaScript σύμφωνα με το πρότυπο ECMA-262 3rd Edition - December 1999. Ο δημιουργός της JSON είναι ο Douglas Crockford, ο οποίος είναι μηχανικός πληροφορικής στη Yahoo και ιδρυτής της ιστοσελίδας json.org

Με την εξάπλωση της χρήσης του παγκοσμίου ιστού και την εμφάνιση του AJAX και των Web Services η χρήση της JSON έχει αυξηθεί ραγδαία και αποτελεί μια πιο ελαφριά (lightweight) εναλλακτική για την XML. Ένα αρχείο με δεδομένα μορφοποιημένα σε JSON είναι πιο μικρό σε σχέση με ένα αντίστοιχο XML, εξαιτίας του γεγονότος ότι τα δεδομένα σε JSON δεν περικλείονται από tags.

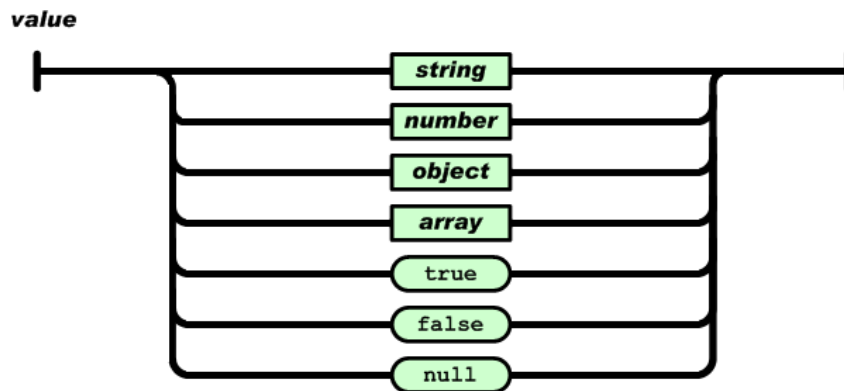
Η Yahoo την υιοθέτησε το 2005 για την ανάκτηση πληροφοριών στις Web Services της που παρέχονται από το Yahoo! API. Ακολούθησε η Google το 2006 η οποία το χρησιμοποιεί στα διάφορα APIs του Google Code.

Στην πράξη η JSON είναι μια μορφή κειμένου για την σειριοποίηση (serialization) δομημένων δεδομένων.

Η JSON ορίζει ένα μικρό σύνολο κανόνων μορφοποίησης για την αναπαράσταση δομημένων δεδομένων. Σύμφωνα με το RFC 4627 μπορούν να αναπαρασταθούν τέσσερις βασικοί τύποι δεδομένων (strings, Boolean, null, numbers) και δύο δομημένοι τύποι δεδομένων (αντικείμενα και πίνακες).

6.11.1 Values

Οι βασικοί τύποι αποκαλούνται values (τιμές) και μπορούν να πάρουν τις ακόλουθες μορφές όπως αυτές αναπαρίστανται στο Σχήμα 6.3



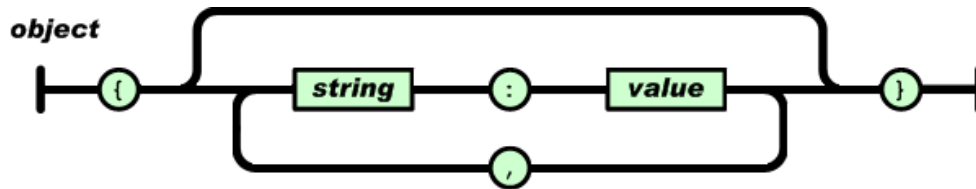
Εικόνα 6.3 Οι δυνατές τιμές των βασικών τύπων σε JSON

Ειδικά για τα strings αυτά πρέπει να περικλείονται από διπλά εισαγωγικά " και μπορούν να έχουν ως τιμή οποιονδήποτε χαρακτήρα που περιλαμβάνεται στο πρότυπο Unicode εκτός από τους χαρακτήρες " (διπλά εισαγωγικά) ή \ (backslash). Με τη χρήση της backslash μπορούν να αναπαρασταθούν ειδικοί χαρακτήρες όπως το backspace (\b) ή το carriage return (\r).

6.11.2 Object

Ένα αντικείμενο είναι ένα σύνολο από ζευγάρια ονομάτων (names) και τιμών (values) . Περικλείεται από άγκιστρα { και }. Κάθε όνομα ακολουθείται από τον χαρακτήρα : (διπλή τελεία) και την τιμή του (value). Οι τιμές που μπορεί να δεχτεί είναι αυτές που έχουν

οριστεί για τους βασικούς τύπους δεδομένων (values). Κάθε ζεύγος ονόματος/τιμής διαχωρίζεται με κόμμα. Η γενική μορφή του αντικειμένου δίνεται στο Σχήμα 6.4



Εικόνα 6.4 Η μορφή του αντικειμένου σε JSON

Τα αντικείμενα υλοποιούνται χωρίς να υφίσταται κάποια κλάση που τα ορίζει, όπως είναι συνηθισμένο στις γλώσσες C++ και Java. Απλώς τα υλοποιούμε με τη δομή και το περιεχόμενο που θέλουμε και στην περίπτωση της JavaScript τα αναθέτουμε σε μία μεταβλητή έτσι ώστε να μπορούμε να τα προσπελάσουμε.

Για παράδειγμα αν θέλουμε να γράψουμε ένα αντικείμενο που αναπαριστά ένα πρόσωπο (Person) μπορούμε να επιλέξουμε την ακόλουθη μορφή:

```
var ob1 = {  
    "name": "Βασίλειος",  
    "surname": "Βίτσας",  
    "tomeas": " ΣΥΣΤ & ΤΕΧΝΟΛ"  
}
```

Για να προσπελάσουμε ή να αλλάξουμε την ιδιότητα name του αντικειμένου ob1, αρκεί να γράψουμε σε JavaScript: ob1.name

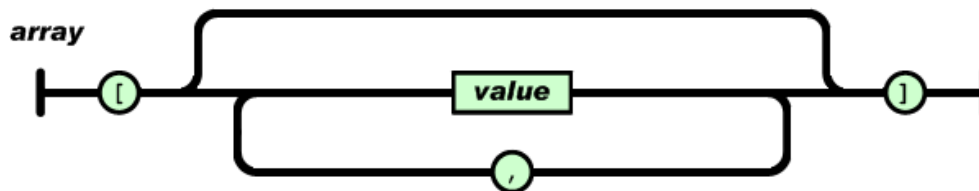
Ένα αντικείμενο μπορεί να είναι σύνθετο και να περιλαμβάνει άλλα αντικείμενα όπως για παράδειγμα

```
var ob2 = {  
    "name": "Βασίλειος",  
    "surname": "Βίτσας",  
    "data": {  
        "email": "vitcas@gmail.com",  
        "phone": 123456789,  
        "id": 100  
    }  
}
```

Για να προσπελάσουμε ή να αλλάξουμε την ιδιότητα `id` του αντικειμένου `ob2`, αρκεί να γράψουμε σε JavaScript: `ob2.data.id = 200`

6.11.3 Arrays

Οι πίνακες είναι δομές δεδομένων που αναπαρίστανται από συλλογές τιμών (values) ή αντικειμένων. Ένας πίνακας ξεκινά με αριστερή αγκύλη `[` και τελειώνει με δεξιά αγκύλη `]`. Κάθε τιμή διαχωρίζεται με κόμμα `,` από μία άλλη (Σχήμα 6.5).



Εικόνα 6.5 Η γενική μορφή ενός πίνακα στη JSON

Η JSON επιτρέπει να έχουμε πίνακες τιμών, πίνακες αντικειμένων και μεικτούς πίνακες (με τιμές και αντικείμενα). Δεν υπάρχει περιορισμός στον τύπο των τιμών και των αντικειμένων. Ένας πίνακας μπορεί να περιέχει δεδομένα διαφορετικού τύπου.

```
Var arr1 = [2,3,4,5,6]; //πίνακας ακεραίων
```

```
Var arr2 = ["vasileos","vitsas","volotnikov"] //πίνακας strings
```

```
Var arr3 = [3,4,5, "sergios", [12,34,5.4], 3] //πίνακας δεδομένων διαφορετικών τύπων
```

```
//πίνακας αντικειμένων
```

```
var ob1 = [ {
    "surname": "Βίτσας ",
    "name": "Βασίλειος",
    "height": 175
},
{
    "surname": "Παπαδόπουλος ",
    "name": "Στέφανος",
    "height": 170
}
]
```

6.11.4 JSONP

Η JSONP (json with padding) είναι μια επέκταση στο ήδη υπάρχον πρότυπο. Δημιουργήθηκε από τον Bob Ippolito το 2005 και δεν αποτελεί πρότυπο όπως η JSON αλλά τεχνική. Βασική της λειτουργία είναι ότι επιτρέπει ένα client side script (JavaScript) που έχει φορτωθεί από μια σελίδα να ζητά (request) δεδομένα σε μορφή JSON από έναν server διαφορετικό από αυτόν που είναι αποθηκευμένη η σελίδα.

Οι web browsers ακολουθούν την πολιτική ασφαλείας **same origin policy**. Σύμφωνα με αυτήν ένα client side script που βρίσκεται σε μια σελίδα μπορεί να ανακτά ή να μεταχειρίζεται μέσω AJAX δεδομένα μόνο από σελίδες και scripts που βρίσκονται στο ίδιο domain. Έτσι αν το script βρίσκεται στο domain example1.com, δεν μπορεί να ανακτήσει δεδομένα από ιστοσελίδες που βρίσκονται στο domain example2.com. Το θέμα αυτό περιορίζει τους προγραμματιστές που θέλουν να λάβουν δεδομένα από Web Services ή ιστοσελίδες τρίτων με τη χρήση AJAX.

Το JSONP είναι δεδομένα μορφοποιημένα σε JSON που περιβάλλονται από παρενθέσεις () και αποτελούν ορίσματα μιας συνάρτησης επιστροφής (callback function). Έτσι τα δεδομένα φορτώνονται από το client script δυναμικά σαν κλήση της συνάρτησης που έχουμε ορίσει ως callback με όρισμα τα δεδομένα που αιτούμαστε από το δικό μας ή ένα ξένο domain.

Με τον όρο συνάρτηση callback εννοούμε τη συνάρτηση η οποία θα κληθεί όταν τα δεδομένα τα οποία αιτηθήκαμε φτάσουν στον client. Τα δεδομένα αυτά θα περάσουν ως ορίσματα στην συνάρτηση αυτή, από όπου μπορούμε να τα επεξεργαστούμε και να τα μετασχηματίσουμε κατά το δοκούν.

Έτσι αν μέσα στον κώδικά μας θέλουμε να καλέσουμε μια web service μέσω JavaScript με μέθοδο HTTP GET (XMLHttpRequest object) που δέχεται ως όρισμα ένα όνομα και επιστρέφει αντικείμενα που έχουν αυτό το όνομα σε μορφοποίηση JSON π.χ.

```
{ "name": "Sergios", "height": 1.78 }
```

και έστω ότι το URL της web service είναι <http://www.wstest/getobject> . Το url πέρνει ως παραμέτρους το callback και το όνομα namein που ψάχνουμε. Για να βρούμε όλα τα

αντικείμενα με όνομα Sergios θα πρέπει στο url της web service να ορίσουμε ένα όνομα για την callback function π.χ. ptyxiaki και το κλειδί αναζήτησης π.χ. namein=Sergios.

Το url θα είναι το <http://www.wstest/getobject?callback=ptyxiaki&namein=Sergio> και η απάντηση θα έχει τη μορφή:

```
ptyxiaki ({ "name": "Sergios", "surname": "Gavrihidhs", "age": 20})
```

Ο πλήρης κώδικας με τη χρήση της JQuery θα έχει την εξής μορφή, όπου ως callback βάζουμε τον χαρακτήρα ?:

```
jQuery.getJSON("http://www.wstest/getobject&callback=?namein=Sergios",  
function(data) {  
    alert("Name: " + data.name + ", Surname: " + data.surname + " Age:  
    " + data.age});
```

6.11.5 Θέματα ασφαλείας

Η JSONP μας βοηθά να καλέσουμε REST web services εύκολα και γρήγορα, αλλά έχει «τρύπες» ασφαλείας που την καθιστούν ευάλωτη σε επιθέσεις.

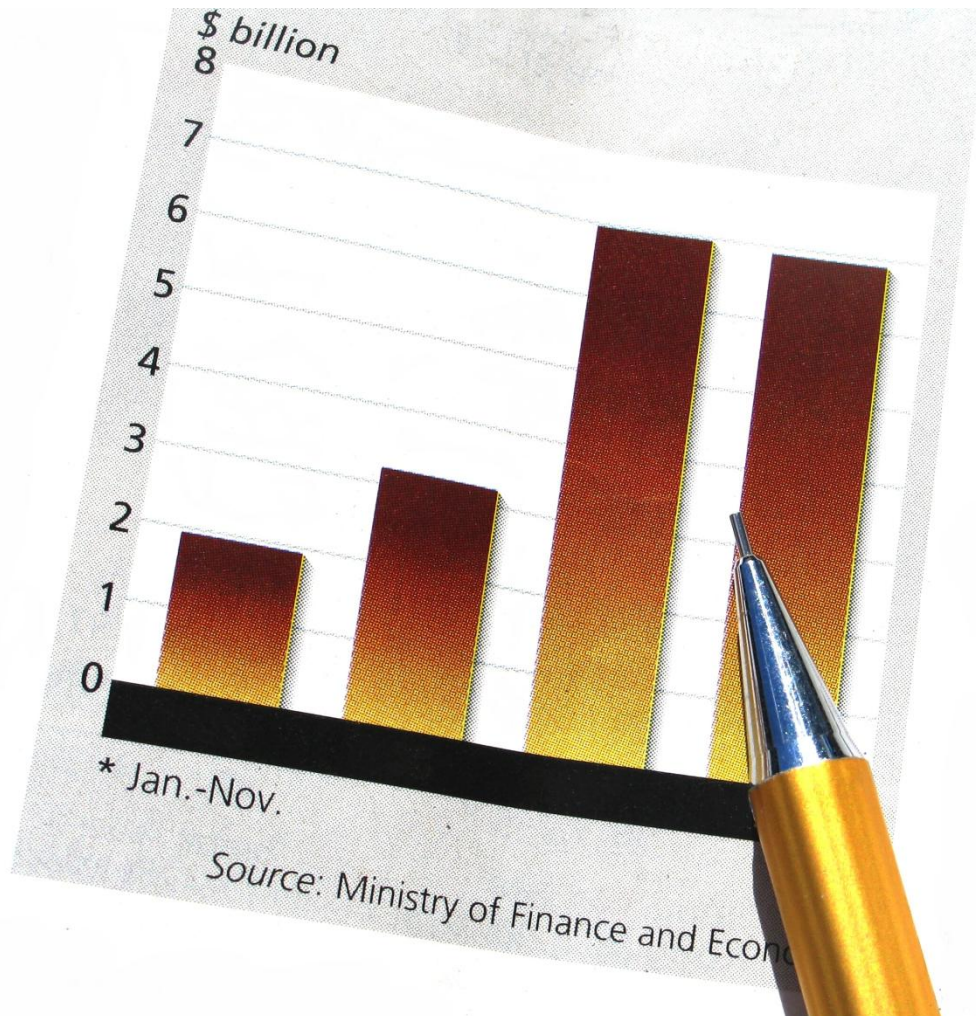
Το γεγονός ότι χρησιμοποιεί το tag <script> και δεν σέβεται την πολιτική same origin policy την καθιστά ευάλωτη σε επιθέσεις τύπου cross site request forgery (CSRF). Έτσι μια κακόβουλη σελίδα μπορεί να ζητήσει δεδομένα μέσω JSONP από μια άλλη σελίδα και να αποκαλύψει ευαίσθητα δεδομένα που μπορεί να έχει αποθηκευμένα ο χρήστης στην κληθείσα δεύτερη σελίδα. Για αυτόν το λόγο συστήνεται προσοχή στη χρήση της JSONP και κρίνεται καλό να μην χρησιμοποιείται για τη μεταφορά ευαίσθητων δεδομένων.

6.12 Επίλογος

Στο κεφάλαιο αυτό αναλύθηκαν λεπτομερώς τα web services σε βαθμό που μπορεί να γίνει κατανοητό το πως αυτές ήταν η πρώτη επιλογή μας στην πτυχιακή αυτή και με ποιόν τρόπο χρησιμοποιηθήκαν για να επιτύχουμε το καλύτερο αποτέλεσμα, όσων

αφορά την δύναμη που μπορούν να προσφέρουν και τις δυνατότητες που μεταλαμπαδεύουν στην αρχιτεκτονική του τελικού προϊόντος. Παράλληλα παρουσιάστηκαν παρακλάδια των τεχνολογιών που υποστηρίζονται από τα web services και που μπορούν να χρησιμοποιηθούν σε άλλου είδους προγράμματα, αποδεικνύοντας το σύνολο των δυνατοτήτων τους και το πως θα μπορούν οι υπηρεσίες αυτές να είναι μέρος οποιουδήποτε προγράμματος, είτε στο internet είτε σε desktop εφαρμογές.

7 ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΗΣ ΚΑΙ ΔΙΑΓΡΑΜΜΑΤΑ



"Once a new technology starts rolling, if you're not part of the steamroller, you're part of the road."

Stewart Brand.

7.1 Εισαγωγή

Κάθε λειτουργικό που υπάρχει έχει δημιουργηθεί για να καλύψει μία ανάγκη. Για την σωστή σχεδίαση κάποιου λογισμικού λοιπόν, οποιοδήποτε τύπου θα πρέπει να συναντηθούμε με τον κάθε ενδιαφερόμενο και να μπορέσουμε να καταγράψουμε τους τρόπους με τους οποίους θα πρέπει να λειτουργεί το πρόγραμμα το οποίο γράφουμε, και αυτό για να μπορέσει να ανταποκριθεί όσον το δυνατό καλύτερα στις ανάγκες του.

Ένας από τους πιο ορθούς τρόπους να γίνει αυτό είναι να δημιουργηθούν οι περιπτώσεις χρήσης και να μπορέσει ο κάθε ενδιαφερόμενος να δει κάθε βήμα της διαδικασίας που αποτελεί λύση στο πρόβλημα του και να εκτιμήσει το κατά πόσο αυτό αντιπροσωπεύει την καθημερινότητα του. Αυτό έχει σαν κύριο στόχο την δημιουργία ενός κοινού τρόπου επικοινωνίας μεταξύ του δημιουργού και ενδιαφερομένου ώστε να μπορέσουν αν επιτύχουν μαζί το καλύτερο δυνατό αποτέλεσμα.

Πέρα όμως από το ότι αποτελούν σημείο επικοινωνίας, οι περιπτώσεις χρήσης είναι και βοήθημα για τον δημιουργό καθώς μπορεί ανά πάσα στιγμή να ελέγχει σε ποιο σημείο βρίσκεται η ανάπτυξη του λογισμικού χωρίς να πρέπει να ανατρέχει συνεχώς στα αρχεία ανάπτυξης του project και να ξέρει ακριβώς τι θα πρέπει να κάνει ολοκληρωμένα η λειτουργία που υλοποιεί.

Στο κεφάλαιο που θα αναπτύξω εδώ θα παρουσιαστούν όλες οι περιπτώσεις χρήσης και τα στοιχεία που τις έκαναν να υλοποιηθούν στο πρόγραμμα της πτυχιακής. Θα αναφερθούν όλα τα βήματα που οδηγούν από την ανάγκη του χρήστη, πχ η εισαγωγή κάποιου δεδομένου στο σύστημα του ΑΤΕΙ Θεσσαλονίκης, μέχρι το αποτέλεσμα που θα είναι η επιτυχής ικανοποίηση της ανάγκης.

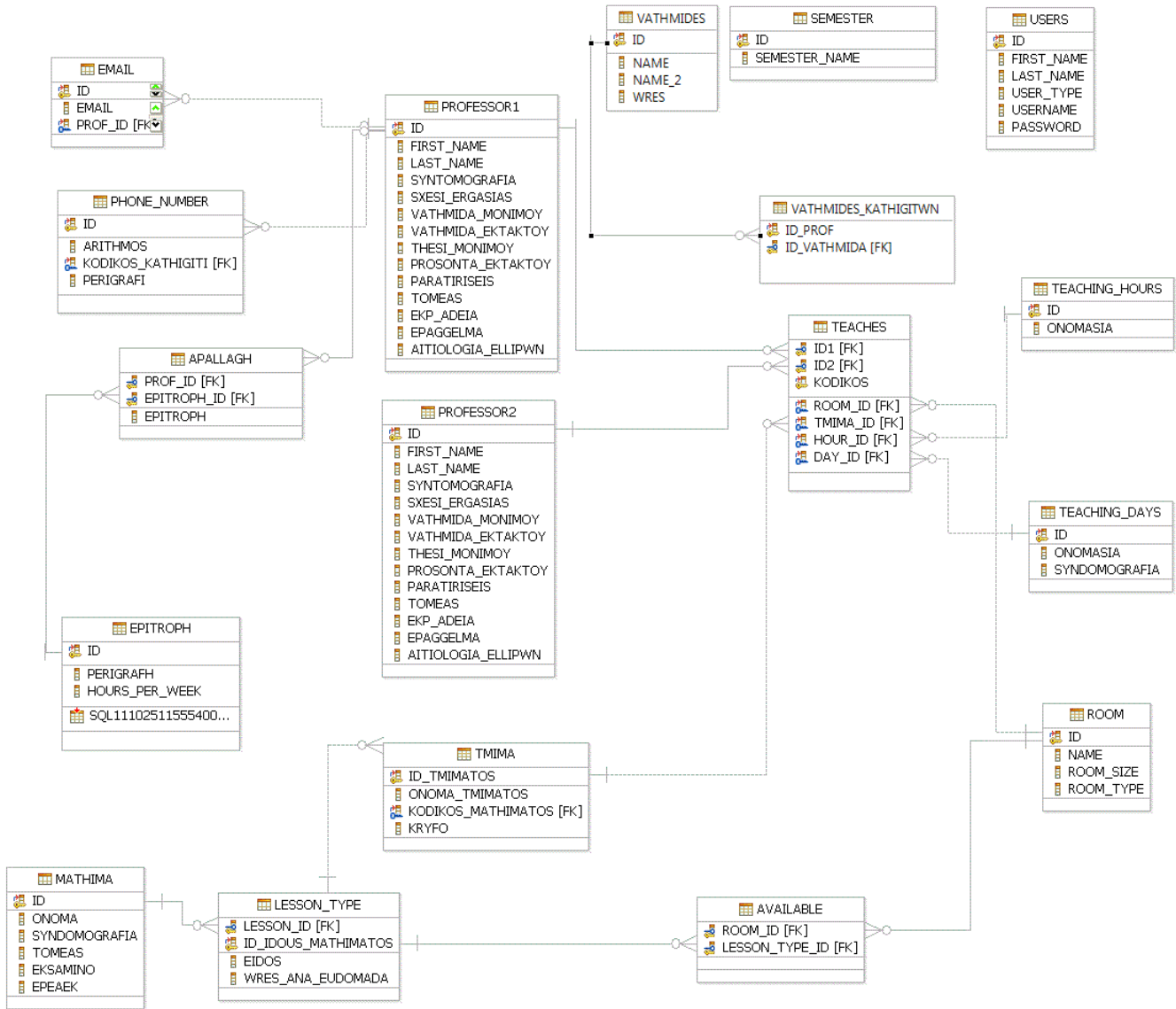
Πάντα οι περιπτώσεις χρήσης είναι πολλές και αφορούν κυρίως τις βασικές χρήσεις του συστήματος. Δεν παρουσιάζουν δηλαδή λανθασμένες υποδοχές ή διαγράμματα όπου περιέχουν πάνω από μια χρήσεις. Αυτό έχει σαν αποτέλεσμα, να μην την απόλυτη συγκέντρωση σε μια χρήση αλλά θα μπορούσαν να εμφανίζουν το πρόβλημα της ασυμβατότητας μεταξύ τους.

Γι αυτό και όταν ολοκληρώνουμε ένα λογισμικό δεν μπορούμε να στηριχθούμε μόνο σε αυτές, και παρουσιάζονται εδώ σαν, ξεχωριστό και ικανό τρόπο εμφάνισης

μεμονωμένων και βασικών ικανοτήτων του προγράμματος, που συνδυασμένων αρμονικά παρέχουν στον ενδιαφερόμενο καθηγητή (σε αυτή τη περίπτωση) το καλύτερο και ποιο άρτιο αποτέλεσμα για την ικανοποίηση των αναγκών του.

Πριν την υλοποίηση οποιασδήποτε εφαρμογής λογισμικού είναι απαραίτητος ο σωστός σχεδιασμός όλων των υποσυστημάτων και των τεχνολογιών που χρησιμοποιούνται. Ο σχεδιασμός αρχίζει με την συλλογή προδιαγραφών. Για την καταγραφή τους είναι απαραίτητες κάποιες συνεντεύξεις με όλους τους πιθανούς χρήστες. Ποιο συγκεκριμένα όταν η εφαρμογή περιλαμβάνει μια βάση δεδομένων είναι σκόπιμο να αποτυπώνονται οι σχέσεις μεταξύ οντοτήτων, καθώς και οι περιορισμοί. Αυτό γίνεται φτιάχνοντας ένα διάγραμμα της βάσης δεδομένων. Στο παρακάτω σχήμα στηρίχθηκε ο σχεδιασμός της Βάσης Δεδομένων του Συστήματος που αναπτύχθηκε στην πτυχιακή αυτή καθώς και η υλοποίηση της διαδικτυακής εφαρμογής του τμήματος πληροφορικής για τις ανάγκες διαχείρισης των μαθημάτων, των καθηγητών, των αιθουσών κ.τ.λ.

Πτυχιακή εργασία του Σέργιου Γαβριηλίδη



Εικόνα 7.1 Βάση Δεδομένων

Σε μια πρώτη εξερευνητική φάση χρησιμοποιήσαμε την διαδικασία ανάπτυξης καταρράκτη (waterfall process) ώστε να διαμορφώσουμε μια συνολική άποψη για τις απαιτήσεις και την κατανομή τους στα μέρη του έργου προκειμένου να προκύψουν ορισμένες υψηλού επιπέδου σχεδιαστικές αποφάσεις. Ακολούθως, εφαρμόσαμε την ευέλικτη (agile) επαναληπτική διαδικασία ανάπτυξης (iterative process) για την παραγωγή της τελικής απόφασης σχεδιασμού του προγράμματος της πτυχιακής σε ελεγχόμενα και επαναλαμβανόμενα υπό-στάδια.

Η υλοποίηση της εφαρμογής έγινε με χρήση php , javascript, και απλής html σε περιβάλλον ανάπτυξης NetBeans. Χρησιμοποιήθηκαν επίσης web τεχνολογίες όπως οι DHTML, CSS, AJAX 3.5, JQuery, JavaScript, κ.ά. κυρίως για να αυξήσουν την διαδραστικότητα, όπως επίσης την εμφάνιση και τη φιλικότητα προς τους χρήστες. Για τη σχεδίαση του σχήματος της βάσης δεδομένων χρησιμοποιήθηκε το IBM DataStudio. Το γραφικό περιεχόμενο των ιστοσελίδων σχεδιάστηκε με Adobe Photoshop CS3.

7.2 Μοντέλο Περίπτωσης Χρήσης

Μια περίπτωση χρήσης περιγράφει μια σειρά από ενέργειες που εκτελεί το σύστημα ώστε να αποδώσει χρήσιμα αποτελέσματα σε ένα ρόλο του συστήματος (χρήστη, άλλο συνεργαζόμενο λογισμικό). Ρόλος είναι ένας χρήστης ή ένα εξωτερικό σύστημα (λογισμικό) που αλληλεπιδρά με το λογισμικό ώστε να λάβει κάποια χρήσιμη πληροφορία. Οι περιπτώσεις χρήσης περιγράφουν σενάρια αλληλεπίδρασης μεταξύ των ρόλων και του συστήματος.

7.3 Περίπτωση Χρήσης: Εισαγωγή στο Σύστημα

Κύριος Χειριστής: Προγραμματιστής, Τομεάρχης , Προϊστάμενος , Καθηγητής

Περιγραφή: Εισαγωγή χρήστη στο Σύστημα

Εμπλεκόμενοι: Προγραμματιστής, Τομεάρχης , Προϊστάμενος , Καθηγητής

Προϋποθέσεις: Να έχει δημιουργηθεί το όνομα και ο κωδικός του χρήστη Να έχει ταυτοποιηθεί ο χρήστης

Κατάσταση Εξόδου: Ο χρήστης έχει εισέλθει στο Σύστημα και είναι έτοιμος να το χρησιμοποιήσει.

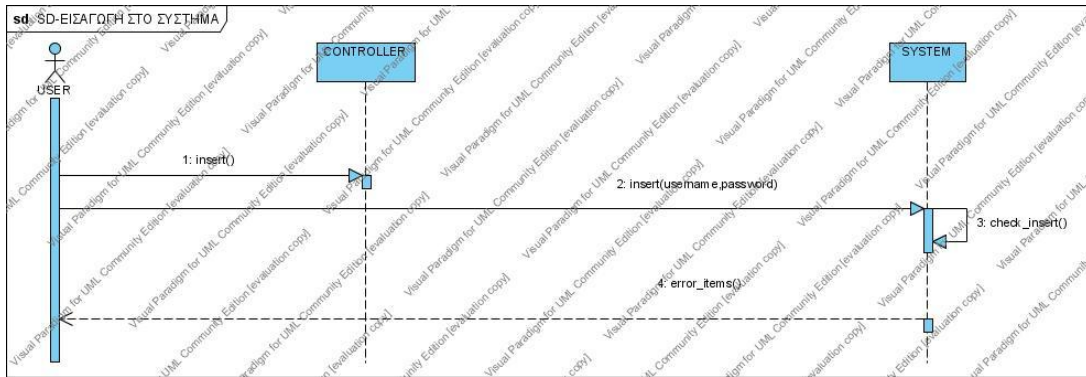
Βασικό Σενάριο:

1. Προγραμματιστής, Τομεάρχης , Προϊστάμενος , Καθηγητής Καθηγητής εισάγει το όνομα και τον κωδικό χρήστη
2. Το Σύστημα ελέγχει την εγκυρότητα των στοιχείων του χρήστη
3. Το Σύστημα εισάγει τον χρήστη στην εφαρμογή

Εναλλακτικό Σενάριο:

2. Το όνομα και ο κωδικός χρήστη είναι λανθασμένος
 - 2.1. Το Σύστημα εμφανίζει μήνυμα λάθους και ζητά από τον χρήστη να επανεισάγει τα στοιχεία του
 - 2.1.1. Επιστροφή στο βήμα 1

Διάγραμμα ακολουθίας



Εικόνα 7.2 Διάγραμμα Ακολουθίας (Εισαγωγή στο Σύστημα)

7.4 Περίπτωση Χρήσης: Δημιουργία, Διαγραφή, Τροποποίηση Καθηγητή/Μαθήματος/Απαλλαγών

Κύριος Χειριστής : Προγραμματιστής, Τομεάρχης

Περιγραφή : Δημιουργία, τροποποίηση, Διαγραφή : Καθηγητή, Μαθήματος, Απαλλαγής

Εμπλεκόμενοι : Προγραμματιστής, Τομεάρχης

Προϋποθέσεις: Να έχει ταυτοποιηθεί ο : Προγραμματιστής, Τομεάρχης στο σύστημα.

Κατάσταση Εξόδου: Ο : Προγραμματιστής, Τομεάρχης έχει δημιουργήσει (τροποποιήσει) ένα νέο μάθημα, καθηγητή κ.λ.π.

Βασικό Σενάριο:

1. Ο Προγραμματιστής, Τομεάρχης ζητά από το Σύστημα να δημιουργήσει ένα νέο μάθημα, καθηγητή κ.λ.π.
2. Το Σύστημα ζητά το όνομα του μαθήματος, καθηγητή
3. Ο Προγραμματιστής, Τομεάρχης συμπληρώνει το όνομα του μαθήματος, καθηγητή
4. Το Σύστημα δημιουργεί ένα νέο μάθημα, καθηγητή

Εναλλακτικό Σενάριο

3. Το όνομα που εισήγαγε ο Προγραμματιστής, Τομεάρχης υπάρχει ήδη

3.1. Το Σύστημα ενημερώνει τον Προγραμματιστής, Τομεάρχης ότι το όνομα που έδωσε υπάρχει ήδη και ζητά να δώσει ένα άλλο όνομα

3.2. Επιστροφή στο βήμα 3

3. Τα δεδομένα που εισήγαγε ο Προγραμματιστής, Τομεάρχης δεν είναι επιτρεπτά από το πρόγραμμα.

3.1. Το Σύστημα απορρίπτει την εγγραφή στην βάση

3.2. Επιστροφή στο βήμα 3

7.5 Περίπτωση Χρήσης: Δημιουργία Μαθήματος στο Πρόγραμμα

Κύριος Χειριστής : Προγραμματιστής

Περιγραφή : Δημιουργία Μαθήματος στο Πρόγραμμα

Εμπλεκόμενοι: : Προγραμματιστής

Προϋποθέσεις: Να έχει ταυτοποιηθεί ο Προγραμματιστής στο σύστημα.

Κατάσταση Εξόδου: Ο Προγραμματιστής έχει δημιουργήσει (τροποποιήσει) ένα νέο μάθημα, καθηγητή κ.λ.π.

Βασικό Σενάριο:

1. Ο Προγραμματιστής ζητά από το Σύστημα να δημιουργήσει, διαγράψει, τροποποιήσει ένα μάθημα.

2. Το Σύστημα ζητά το όνομα του μαθήματος (τύπος, είδος), μέρα, ώρα, αίθουσα, καθηγητή (καθηγητή 2 αν διδάσκουν μαζί).
3. Ο Προγραμματιστής συμπληρώνει (επιλέγει) όλα τα απαραίτητα στοιχεία.
4. Το Σύστημα μετά από απαραίτητους ελέγχους δημιουργεί ένα μάθημα (μαζί με άλλα στοιχεία : μέρα, ώρα κλπ.) και γίνεται εγγραφή στην βάση.
5. Γίνεται εμφάνιση αντίστοιχης εγγραφής στο “Πρόγραμμα”.

Εναλλακτικό Σενάριο

3. Οι επιλογές που έκανε ο Προγραμματιστής δεν είναι επιτρέπονται από το πρόγραμμα.

3.1. Το Σύστημα ενημερώνει τον Προγραμματιστή ότι οι επιλογές έχουν γίνει λανθασμένα και με αντίστοιχο μήνυμα ενημερώνει τον χειριστή να κάνει άλλες επιλογές

3.2. Επιστροφή στο βήμα 3

7.6 Περίπτωση Χρήσης: Τροποποίηση Μαθήματος στο Πρόγραμμα

Κύριος Χειριστής : Προγραμματιστής

Περιγραφή : Τροποποίηση ήδη υπάρχουσας εγγραφής μαθήματος ή διδασκαλίας μαθήματος από τον έναν συγκεκριμένο καθηγητή στο πρόγραμμα

Εμπλεκόμενοι : Προγραμματιστής

Προϋποθέσεις: Να έχει ταυτοποιηθεί ο Προγραμματιστής στο σύστημα.

Κατάσταση Εξόδου: Ο Προγραμματιστής έχει τροποποιήσει ένα μάθημα, καθηγητή .

Βασικό Σενάριο:

1. Ο Προγραμματιστής ζητά από το Σύστημα να τροποποιήσει ένα μάθημα.
2. Το Σύστημα εμφανίζει όλα τα στοιχεία του μαθήματος (μέρα, ώρα, καθηγητές κ.λ.π.)
3. Ο Προγραμματιστής κάνει τροποποιήσεις.
4. Το Σύστημα μετά από απαραίτητους ελέγχους αποθηκεύει τροποποιημένο μάθημα (εγγραφή) στην βάση.
5. Γίνεται εμφάνιση αντίστοιχης εγγραφής στο “Πρόγραμμα”.

Εναλλακτικό Σενάριο

3. Επιλογές που έκανε ο Προγραμματιστής δεν έγιναν δεκτές

3.1. Το Σύστημα ενημερώνει τον Προγραμματιστή ότι τροποποιήσεις έχουν γίνει λανθασμένα και με αντίστοιχο μήνυμα ενημερώνει τον χειριστή να κάνει άλλες επιλογές

3.2. Επιστροφή στο βήμα 3

7.7 Περίπτωση Χρήσης: Διαγράφη Μαθήματος στο Πρόγραμμα

Κύριος Χειριστής : Προγραμματιστής

Περιγραφή : Διαγραφή μαθήματος

Εμπλεκόμενοι : Προγραμματιστής

Προϋποθέσεις: Να έχει ταυτοποιηθεί ο Προγραμματιστής στο σύστημα.

Κατάσταση Εξόδου: Ο Προγραμματιστής έχει διαγράψει συγκεκριμένη εγγραφή από την βάση

Βασικό Σενάριο:

1. Ο Προγραμματιστής ζητά από το Σύστημα να διαγράψει ένα συγκεκριμένο μάθημα.
2. Το Σύστημα εμφανίζει όλα τα στοιχεία του μαθήματος (μέρα, ώρα, καθηγητές κλπ.)
3. Ο Προγραμματιστής κάνει διαγραφή πατώντας αντίστοιχο κουμπί.
4. Το Σύστημα διαγράφει αντίστοιχη εγγραφή (με βάση του id του μαθήματος) από την βάση.
5. Το “Πρόγραμμα” ξανά φορτώνεται από την βάση και εμφανίζεται εκ νέου.

7.8 Επίλογος

Σε αυτό το κεφάλαιο αναπτύξαμε τις ποιό βασικές περιπτώσεις χρήσης, οι οποίες και θα μπορούσαν σε συνδυασμό να παρέχουν την λύση το θέμα της πτυχιακής μου εργασίας. Πιο συγκεκριμένα αναλύσαμε κάθε περίπτωση χρήσης με σκοπιά την ανάγκη του καθηγητή να διαπράττει λειτουργίες μέσα στο σύστημα. Αξίζει να αναφερθεί ότι όλες οι λειτουργίες που αναφέρθηκαν πιο πάνω έχουν ως κύριο κοινό στοιχείο το ότι η κάθε μία από αυτές είναι ήδη υπάρχουσα ανάγκη στο τμήμα πληροφορικής του ΑΤΕΙ Θεσσαλονίκης, η οποία έχει καταγραφεί ως περίπτωση χρήσης. Κάτι που συμπερασματικά σημαίνει ότι δεν έχει σημασία η παλαιότητα κάποιας ανάγκης καθώς όλες, ακόμα και οι πιο παλιές ή οι πιο νέες θα μπορούσαν να αναπτυχθούν με έναν τέτοιο τρόπο και να αποσαφηνίζουν όλα τα βήματα με τα οποία αυτή τη στιγμή, αυτό το σύστημα, ή ένα άλλο αντιμετωπίζει κώλυμα.

8.ΟΔΗΓΟΣ ΧΡΗΣΗΣ ΛΟΓΙΣΜΙΚΟΥ

8.1 Εισαγωγή

Για να μπορέσει να καταλάβει ο αναγνώστης ακριβώς το πώς δουλεύει το σύστημα το οποίο σχεδιάστηκε και υλοποιήθηκε στην συγκεκριμένη εργασία, πέρα από την αναφορά στις τεχνολογίες και στην αναφορά για το πώς επιμέρους κομμάτια της λειτουργούν, θα πρέπει να εισάγουν τον αναγνώστη και στο πώς το ίδιο τι πρόγραμμα λειτουργεί και το πώς θα μπορέσει να φέρει σε πέρας το τελικό του σκοπό, δηλαδή την διαχείριση των μαθημάτων/καθηγητών και την οργάνωση των ωρών/ μαθημάτων σε ένα τμήμα μιας ανωτάτου επιπέδου σχολής.

Το πρόγραμμα που έχει υλοποιηθεί, μετά από σαφής κατανόηση των απαιτήσεων κάποιας ανώτατης σχολής, και συγκεκριμένα εδώ το ΑΤΕΙ Θεσσαλονίκης, έχει σχεδιαστεί με τρόπο τον οποίο οι ήδη υπάρχοντες χρήστες δεν θα βρουν δύσκολο να αφομοιώσουν. Πιο συγκεκριμένα υπάρχει μια διαχείριση χρηστών, με ιεραρχία και ταυτόχρονα ένα σύστημα σύνδεσης χρηστών με βάση τον γνωστό τρόπο ονόματος/συνθηματικού. Έπειτα κάθε διεργασία της σχολής είναι διαχωρισμένη ξεκάθαρα σε μενού και υπό-μενού ώστε να είναι απόλυτα κατανοητό από τον χρήστη το πώς μπορεί να χειριστεί την αρμοδιότητα που του έχει οριστεί, και έχει δικαίωμα σε αυτή. Είναι σαφές λοιπόν πως υπάρχει ένα σύστημα διαχείρισης χρηστών όπου ο κάθε ένας έχει πρόσβαση σε αρμοδιότητες που επιτρέπει το επίπεδο του σαν χρήστης, και οι οποίες είναι ξεχωριστές για τον καθένα.

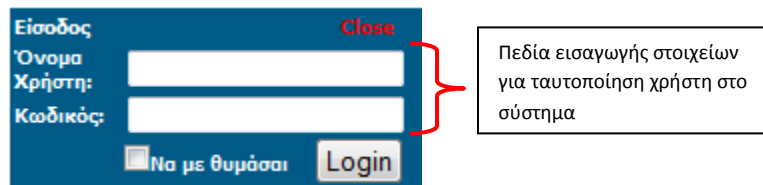
Στο συγκεκριμένο κεφάλαιο θα αναπτύξουμε τις λειτουργικότητες του προγράμματος με την ακόλουθη σειρά. Πρώτα θα αναφερθούμε στον τρόπο με τον οποίο ο χρήστης μπορεί να συνδεθεί και να εμφανίσει τα πρωταρχικά του δεδομένα στην αρχική του σελίδα, έπειτα θα εξηγήσουμε λεπτομερώς την λειτουργικότητα όλων των τύπων, των μενού και των υπό-μενού με σκοπό την παρουσίαση της πλήρους ικανότητας του προγράμματος για οποιονδήποτε χρήστη. Επιπρόσθετος θα αναφερθούν και μερικά παραδείγματα χρήσης από αυτά, συνεχίζοντας θα αναφερθούμε στις δυναμικές αναφορές των λειτουργιών και το πώς αυτές μπορούν να εμφανιστούν στον χρήστη με σκοπό την εκτύπωση και την απλή εμφάνιση. Τέλος θα παρουσιάσουμε

τα συμπεράσματα μας στην χρήση των λειτουργιών αυτών και το γιατί πιστεύεται ότι είναι ικανές να καλύψουν τις ανάγκες ενός συστήματος διαχείρισης του τμήματος πληροφορικής του ΑΤΕΙ Θεσσαλονίκης και κατ'επέκταση περισσότερων τμημάτων βάσει της δυναμικότητας του προγράμματος αυτού.

Γενικότερα, η χρήση ενός συστήματος διαχείρισης από κάποιον καθηγητή υπήρξε το πρωταρχικό στόχο του προγράμματος αυτού, λήφθηκαν υπόψη οι διαφορετικές ανάγκες του κάθε τύπου χρήστη και έχουν υλοποιηθεί στον μεγαλύτερο δυνατό βαθμό, χρησιμοποιώντας όλες τις τεχνολογίες που παρουσιάστηκαν. Το αποτέλεσμα έχει καταφέρει να καλύψει το μεγαλύτερο μέρος αν όχι το σύνολο από αυτές τις ανάγκες και έχει σχεδιαστεί με τέτοιο τρόπο με τον οποίο θα μπορέσει να καλύψει ακόμα και νεοεμφανιζόμενες, κάτι που οφείλει κυρίως, στην χρήση αντικειμενοστρεφών γλωσσών προγραμματισμού και δυναμικών εν γένει βάσεων δεδομένων.

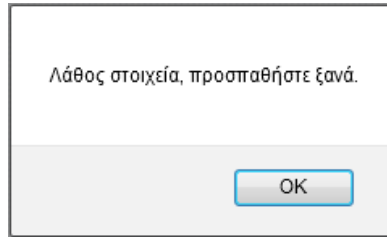
8.2 Είσοδος χρήστη

Στην αρχική σελίδα ο χρήστης πρέπει να εισάγει τα στοιχεία του, όνομα και κωδικό χρήστη, για να μπορέσει να εισέλθει στο σύστημα.

The image shows a login form with a blue background. It has two input fields: 'Όνομα Χρήστη:' and 'Κωδικός:'. Below the fields is a checkbox labeled 'Να με θυμάσαι' and a 'Login' button. A red 'Close' button is in the top right corner of the form. A red bracket on the right side of the form points to a text box that says: 'Πεδία εισαγωγής στοιχείων για ταυτοποίηση χρήστη στο σύστημα'.

Εικόνα 8.1 Είσοδος χρήστη

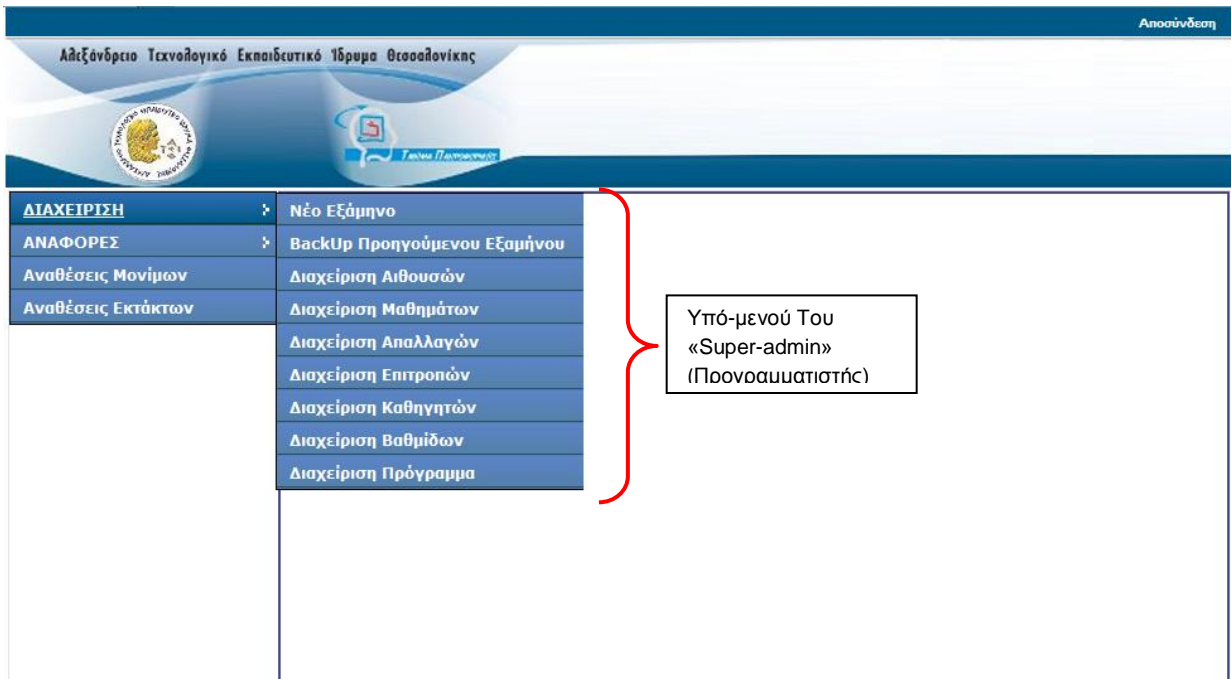
Σε περίπτωση που εισαχθούν λάθος στοιχεία, τότε ένα μήνυμα λάθους θα εμφανιστεί στην οθόνη ενημερώνοντας το χρήστη.



Εικόνα 8.2 Λάθος στοιχεία

Όταν εισάγει ο χρήστης τα σωστά στοιχεία εισόδου, τότε εμφανίζεται ένα μήνυμα υποδοχής με το ονοματεπώνυμό του και τον εισάγει στο σύστημα.

Όταν εισάγει ο χρήστης τα έγκυρα στοιχεία εισόδου, τότε αυτομάτως μπαίνει στην ιστοσελίδα διαχείρισης ανάλογα με τα δικαιώματά του.



Εικόνα 8.3 Επιτυχής είσοδος χρήστη

Υπάρχουν 4 διαχειριστές (super_admin, admin, primary_user_a, primary_user_b) που έχουν διαφορετικά μεταξύ τους δικαιώματα διαχείρισης της βάσης δεδομένων, με συνέπεια να έχουν διαφορετικό υπό-μενού:

Το **default** υπό-μενού έχει εξής επιλογές :

- Εξαμηνολόγιο
- Αιθουσιολόγιο
- Αιθουσιολόγιο(γενικό)
- Μαθητολόγιο
- Καθηγητολόγιο
- Διδασκαλία ανά Μάθημα
- Διδασκαλία ανά Καθηγητή
- Ελεύθερες Αίθουσες

Ενώ ο **Primary_user_b** θα έχει εκτός από το default υπό-μενού και άλλες επιλογές:

- Καθηγητές ανά Μάθημα
- Συνδιδασκαλίες

Ο **Primary_user_a** Θα έχει περισσότερα δικαιώματα από τον **Primary_user_b** που θα εμφανίζονται στα μενού. Συμπληρωματικά σε αυτά θα είναι:

- Απαλλαγές Μονίμων
- **Αναλυτικά Στατ. Διδασκαλίας**
 - Θεωρίες Μονίμων (αναλυτικά)
 - Θεωρίες Εκτάκτων (αναλυτικά)
 - Εργαστήρια Μονίμων (αναλυτικά)
 - Εργαστήρια Εκτάκτων (αναλυτικά)
- **Συνοπτικά Στατιστικά**
 - Θεωρίες ανά βαθμίδα Καθηγητή
 - Θεωρίες ανά βαθμίδα Αναπληρωτή Καθηγητή
 - Θεωρίες ανά βαθμίδα Επίκουρου
 - Θεωρίες ανά βαθμίδα Καθηγητή Εφαρμογών

- Θεωρίες ανά βαθμίδα Επιστημονικού Συνεργάτη
 - Εργαστήρια ανά βαθμίδα Καθηγητή
 - Εργαστήρια ανά βαθμίδα Αναπληρωτή Καθηγητή
 - Εργαστήρια ανά βαθμίδα Επίκουρου
 - Εργαστήρια ανά βαθμίδα Καθηγητή Εφαρμογών
 - Εργαστήρια ανά βαθμίδα Επιστημονικού Συνεργάτη
 - Εργαστήρια ανά βαθμίδα Εργαστηριακό Συνεργάτη
 - Αναθέσεις Μονίμων
 - Αναθέσεις Εκτάκτων
-

Ο **Admin**, ως ακόμα ποιο πάνω στην λίστα δικαιωμάτων, θα έχει ακόμη πιο πολλές επιλογές σε σύγκριση με **Primary_user_a**. Συμπληρωματικά, αυτές είναι:

- Διαχείριση Επιτροπών
 - Διαχείριση Καθηγητών
 - Προσωπικά στοιχεία Καθηγητών
-

Όλο το πλήρες μενού θα το έχει ο **Super_admin**, που συνήθως θα είναι αυτός, που εγκαθιστά το πρόγραμμα:

- ΔΙΑΧΕΙΡΙΣΗ
 - Νέο Εξάμηνο
 - BackUp Προηγούμενου Εξαμήνου
 - Διαχείριση Αιθουσών
 - Διαχείριση Μαθημάτων
 - Διαχείριση Απαλλαγών
 - Διαχείριση Επιτροπών
 - Διαχείριση Καθηγητών
 - Διαχείριση Βαθμίδων
 - Διαχείριση Πρόγραμμα
 - ΑΝΑΦΟΡΕΣ
 - Εξαμηνολόγιο
 - Αιθουσιολόγιο
-

- Αιθουσιολόγιο(check)
- Μαθητολόγιο
- Καθηγητολόγιο
- Διδασκαλία ανά Μάθημα
- Καθηγητές ανά Μάθημα
- Διδασκαλία ανά Καθηγητή
- Ελεύθερες Αίθουσες
- Απαλλαγές Μονίμων
- Συνδιδασκαλίες
- Προσωπικά στοιχεία Καθηγητών
- **Αναλυτικά Στατ. Διδασκαλίας**
 - Θεωρίες Μονίμων (αναλυτικά)
 - Θεωρίες Εκτάκτων (αναλυτικά)
 - Εργαστήρια Μονίμων (αναλυτικά)
 - Εργαστήρια Εκτάκτων (αναλυτικά)
- **Συνοπτικά Στατιστικά**
 - Θεωρίες ανά βαθμίδα Καθηγητή
 - Θεωρίες ανά βαθμίδα Αναπληρωτή Καθηγητή
 - Θεωρίες ανά βαθμίδα Επίκουρο
 - Θεωρίες ανά βαθμίδα Εφαρμογών
 - Θεωρίες ανά βαθμίδα Επιστημονικό Συνεργάτη
 - Εργαστήρια ανά βαθμίδα Καθηγητή
 - Εργαστήρια ανά βαθμίδα Αναπληρωτή Καθηγητή
 - Εργαστήρια ανά βαθμίδα Επίκουρο
 - Εργαστήρια ανά βαθμίδα Εφαρμογών
 - Εργαστήρια ανά βαθμίδα Επιστημονικό Συνεργάτη
 - Εργαστήρια ανά βαθμίδα Εργαστηριακό Συνεργάτη
- Αναθέσεις Μονίμων
- Αναθέσεις Εκτάκτων

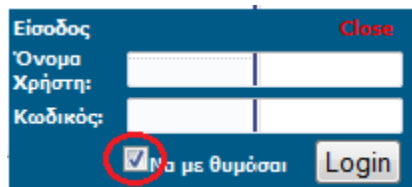
Έξοδος από την σελίδα διαχείρισης μπορεί να γίνει πατώντας σχετικό σύνδεσμο αποσύνδεσης που βρίσκεται πάνω δεξιά στη σελίδα (εκεί απ' όπου κάνετε είσοδος (login))



Εικόνα 8.4 Αποσύνδεση χρήστη

Μετά την αποσύνδεση σας, η ιστοσελίδα φορτώνει τα default υπό-μενού.

Υπάρχει δυνατότητα να μη συμπληρώνεται κάθε φορά το username και password για να εισέρθετε στη σελίδα διαχείρισης σας ενεργοποιώντας (check) το σχετικό τετραγωνάκι δίπλα στο κουμπί login.



Εικόνα 8.5 Ενεργοποιώντας cookies

Τώρα, κάθε φορά που κλείνετε – μπαίνετε από τον ίδιο υπολογιστή στη σελίδα διαχείρισης, ο υπολογιστής θα σας “θυμάται” και θα σας απαλλάσσει από περιττές ενέργειες συμπλήρωσης στοιχείων. Αυτό παρέχεται με την βοήθεια των cookies [15]

8.3 Διαχείριση: Νέο Εξάμηνο

Κάθε φορά που θέλουμε να εισάγουμε ένα νέο εξάμηνο κάνουμε σχετικές ενέργειες από την Διαχείριση – νέο Εξάμηνο. Μας εμφανίζεται σελίδα στην οποία μας παρέχεται δυνατότητα από list box να επιλέξουμε εξάμηνο και να εισάγουμε τον χρόνο. Με το πάτημα του κουμπιού «Εισαγωγή» το νέο εξάμηνο εισάγεται στην βάση και αυτομάτως ενημερώνονται αντίστοιχα πεδία σε όλο το project μας.

ΕΞΑΜΗΝΟ

--Επιλογή Εξαμήνου-- Χειμερινό --Χρόνου-- Εισαγωγή

Εξάμινο

Generated by "ATEI"

Εικόνα 8.6 Νέο Εξάμηνο

8.4 Διαχείριση: BackUp Προηγούμενου Εξαμήνου

Όταν ο χρήστης επιλέξει το «BackUp Προηγούμενου Εξαμήνου», τότε πραγματοποιείται το Backup(αντίγραφο) της βάσης.

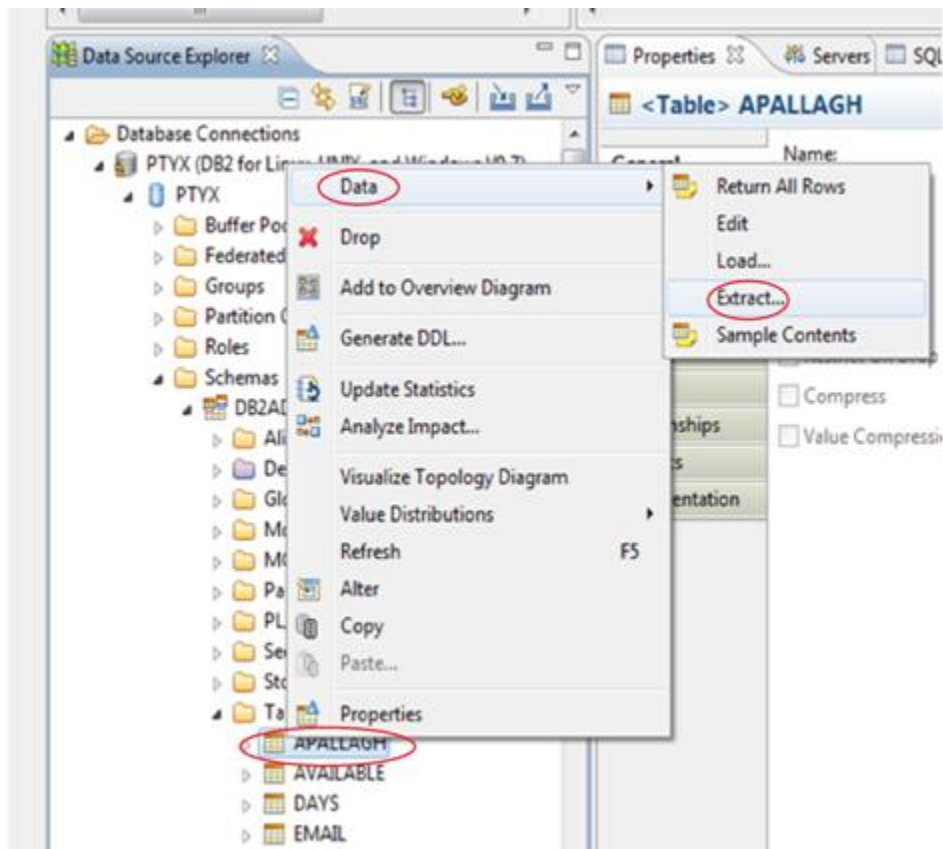


Εικόνα 8.7 BackUp Προηγούμενου Εξαμήνου

Υπάρχει ακόμη μια πολύ βολική και ασφαλή μέθοδο αντιγραφής της βάσης, την οποία μας παρέχει η Data Studio, πιο συγκεκριμένα, μας δίνει δυνατότητα να κάνουμε αντιγραφή (export) πινάκων της βάσης σε μορφή .data. Αυτό μπορούμε να κάνουμε με λίγα και πολύ απλά βήματα.

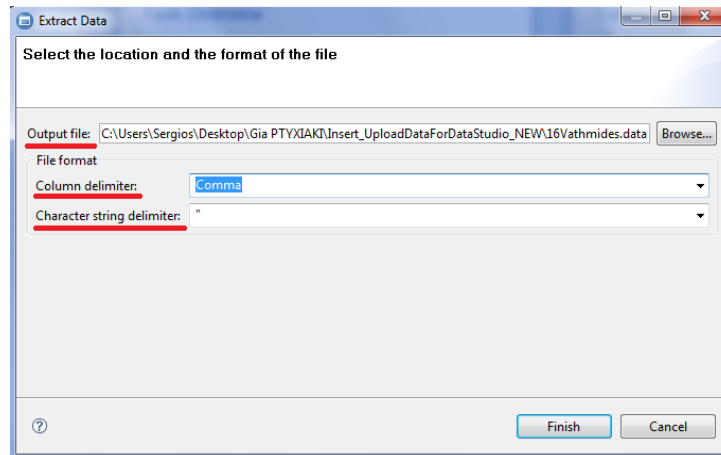
1. Συνδεόμαστε στη βάση μας

2. Επιλέγουμε το SCHEMAS (στο δικό μας παράδειγμα έχουμε schemas “DB2ADMIN”)
3. Βρίσκουμε το φάκελο “Tables”
4. Μας εμφανίζονται πίνακες της βάσης. Από εδώ, με πάτημα του δεξιού κουμπιού του ποντικιού επιλέγουμε Data -> Extract (σχήμα 8.8)



Εικόνα 8.8 BackUp κάθε πίνακα δεδομένων

5. Θα μας εμφανιστεί ένα παράθυρο με τις επιλογές (σχήμα 8.9)
 - Output file (Που θα γίνει αποθήκευση)
 - Column Delimiter (στήλη οριοθέτη)
 - Character string delimiter (συμβολοσειρά οριοθέτη)



Εικόνα 8.9 Αποθήκευση της βάσης

6. Κάνουμε της επιλογές μας και πατάμε “Finish”

Θέλω να σημειώσω ότι αυτή η ιδιαιτερότητα της Data Studio να κάνει Export των δεδομένων των πινάκων σε μορφή .data δεν μας επηρεάζει στην μετά - ανάγνωση εφόσον μπορούμε να ανοίγουμε αυτού του είδους τα αρχεία με προγράμματα όπως TextPad, NotePad τα οποία είναι δωρεάν και χρησιμοποιούνται ευρέως στον κόσμο.

Στο Data Studio υπάρχει δυνατότητα και να εισάγουμε τα δεδομένα στους πίνακες επιλέγοντας αντί Extract το Load [16]. και επίσης τροποποίηση δεδομένων μέσα στους πίνακες (Edit).

8.5 Διαχείριση Αιθουσών

Για την διαχείριση των αιθουσών επιλέγουμε την καρτέλα ΑΙΘΟΥΣΑ, όπου υπάρχουν οι εξής επιλογές: Ονομασία, Χωρητικότητα, Είδος Αίθουσας, Κτίριο, καθώς και επιλογή αίθουσας. Ανάλογα με το τι επιθυμούμε, μπορούμε να εισάγουμε μια νέα αίθουσα (Εισαγωγή), να ανανεώσουμε μια ήδη υπάρχουσα αίθουσα (Ενημέρωση), καθώς και να διαγράψουμε οριστικά μια αίθουσα (Διαγραφή).

ΑΙΘΟΥΣΑ

--Επιλογή αίθουσας-- 101

ΟΝΟΜΑΣΙΑ 101

Χωρητικότητα 63

Είδος Αίθουσας ΕΡΓ

Κτίριο

Εισαγωγή Ενημέρωση Διαγραφή

Εικόνα 8.10 Διαχείριση «Αίθουσα»

8.6 Διαχείριση Μαθημάτων

Στη καρτέλα διαχείρισης των μαθημάτων, μας παρέχονται οι δυνατότητες της Εισαγωγής, Ενημέρωσης και Διαγραφής ενός συγκεκριμένου μαθήματος, και της πλήρους διαχείρισης του είδους του, των τμημάτων του και των αιθουσών του. Όσο αφορά τη διαχείριση των τμημάτων και των αιθουσών κάνουμε ανάλογες επιλογές από τα αντίστοιχα drop-down μενού. Με το πάτημα του κουμπιού Προσθήκη ή Διαγραφή γίνεται εισαγωγή ή διαγραφή στη βάση. Κάθε φορά που επιλέγουμε ένα από τα Εισαγωγή, Ενημέρωση ή Διαγραφή, γίνεται αυτόματη ενημέρωση της σελίδας με χρήση των τεχνολογιών web services της IBM και της Ajax. Αυτές οι τεχνολογίες μας απαλλάσσουν από την επαναφόρτωση ολόκληρης της σελίδας, κάνοντας την σελίδα ταχύτερη στην ανταπόκριση και στην λειτουργικότητα.

<select> με επιλογές των μαθημάτων

ΜΑΘΗΜΑ

--Επιλογή μαθήματος-- Βάσεις Δεδομένων Ι Κουμπιά Προσθήκης/Διαγραφής των ειδών Μαθημάτων

Κωδικός 3402

Όνομα Βάσεις Δεδομένων Ι Είδος ΕΡΓ Προσθήκη Διαγραφή Ωρες 2

Τομέας ΣΥΣΤ & ΤΕΧΝΟΛ

Εξάμηνο Δ Τμήματα T1 Αίθουσες 101

Συντομογραφία ΒαΔει

ΕπεΑεκ

Εισαγωγή Ενημέρωση Διαγραφή

--Διαχρήρηση Τμημάτων-- T3

T1 Κρυφό Προσθήκη Διαγραφή

T2 Κρυφό Προσθήκη Διαγραφή

T3 Κρυφό Προσθήκη Διαγραφή Κουμπιά Προσθήκης/Διαγραφής Τμημάτων

--Διαχρήρηση Αιθουσών-- 101 ΑΙΘΟΥΣΑ 101 Προσθήκη Διαγραφή Κουμπιά Προσθήκης/Διαγραφής Αιθουσών

Εικόνα 8.11 Διαχείριση Μαθήματος

Είδος ΕΡΓ Προσθήκη Διαγραφή Ωρες 2

ΘΕΩΡ Ωρες 6 Προσθήκη

Εικόνα 8.12 Εισαγωγή είδους μαθήματος

8.7 Διαχείριση Απαλλαγών

Η καρτέλα «Απαλλαγή» μας δίνει δυνατότητα να ορίζουμε και να βλέπουμε τις απαλλαγές που έχει ο κάθε καθηγητής. Ανάθεση της επιτροπής στον Καθηγητή γίνεται επιλέγοντας «Επιτροπή», «Καθηγητή», «Εξαμήνου» και πατώντας κουμπί «Υποβολή».

Ενώ για διαγραφή της απαλλαγής απλά πατάμε κουμπί «Διαγραφή» κατόπιν από της επιλογές «Επιτροπή», «Καθηγητή» και «Εξάμηνο».

Εικόνα 8.13 Διαχείριση Απαλλαγών

8.8 Διαχείριση: Επιτροπών

Η καρτέλα «Επιτροπή» διαχειρίζεται όλες τις επιτροπές του τμήματος μας.

Έχουμε δυνατότητα να κάνουμε Εισαγωγή, Ενημέρωση και Διαγραφή (Εικόνα 8.14).

ΕΠΙΤΡΟΠΗ

--Επιλογή επιτροπής-- Διευθυντής

Ονομασία Διευθυντής

Ωρες εβδομάδας 6

Εισαγωγή Ενημέρωση Διαγραφή

Εικόνα 8.14 Διαχείριση Επιτροπών

8.9 Διαχείριση Καθηγητών

Όλη την διαχείριση των καθηγητών (προσωπικά στοιχεία ,τηλέφωνα, ηλεκτρονική διεύθυνση) την κάνουμε από την καρτέλα «Διαχείριση καθηγητών».

Για να εισάγουμε έναν νέο καθηγητή, ακολουθούμε τα εξής βήματα:

- Πατάμε το κουμπί «Νέος Καθηγητής» (σχήμα 8.15)
- Αφού δημιουργηθεί ένας «κενός» καθηγητής (θα εμφανιστεί το σχετικό μήνυμα) συμπληρώνουμε τα στοιχεία του: όνομα, επίθετο, σχέση εργασίας, τομέας κλπ.
- Στο τέλος όταν συμπληρώσουμε τα στοιχεία, πατάμε το κουμπί «Εισαγωγή» για να ενημερωθεί η βάση μας με καινούργια δεδομένα.

Για να κάνουμε αλλαγές στα υπάρχοντα στοιχεία των καθηγητών γράφουμε στα αντίστοιχα πεδία τα νέα δεδομένα (η μέσω drop-down μενού όπου μπορούμε να κάνουμε επιλογές). Ύστερα πατάμε κουμπί «Ενημέρωση» για να αποθηκευτούν οι αλλαγές.

Για να διαγράψουμε έναν καθηγητή πατάμε το κουμπί «Διαγραφή».

ΚΑΘΗΓΗΤΗΣ

--Επιλογή Καθηγητή-- Αγάς Κωνσταντίνος

ΕΠΩΝΥΜΟ: Αγάς
ΟΝΟΜΑ: Κωνσταντίνος
ΣΥΝΤΟΜΟΓΡΑΦΙΑ: ΑγασΚ
ΠΑΡΑΤΗΡΗΣΕΙΣ: Kalos
ΣΧΕΣΗ ΕΡΓΑΣΙΑΣ: ΜΟΝΙΜΗ
ΒΑΘΜΙΔΑ ΜΟΝΙΜΟΥ: ΚΑΘΗΓΗΤΗΣ
ΤΟΜΕΑΣ: ΑΝΑΛ & ΠΡΟΓΡ
ΕΚΠ ΑΔΕΙΑ/ ΔΙΜ. ΥΠΑΛΛ.

Εισαγωγή Ενημέρωση
Διαγραφή Νέος Καθηγητής

ΑΠΑΛΛΑΓΗ
Διευθυντής

ΤΗΛΕΦΩΝΟ
4324 - 234 - 242

ΗΛ.ΔΙΕΥΘΥΝΣΗ
sfetsos@it.teithe.gr

ΘΕΣΗ ΜΟΝΙΜΟΥ: ΠΡΟΣΩΠΟΠΑΓΗΣ

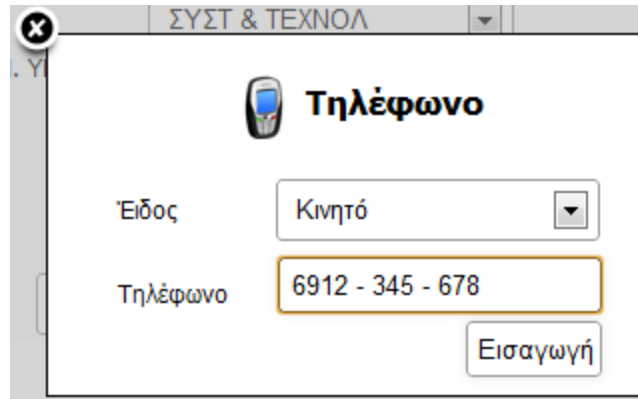
Διαγραφή Τηλεφώνου
Εισαγωγή Τηλεφώνου
Εισαγωγή email
Διαγραφή email

Εικόνα 8.15 Διαχείριση Καθηγητών

Κάθε καθηγητής συνολικά μπορεί να έχει 6 τηλέφωνα (2 κινητά, 2 του σπιτιού και 2 της εργασίας), τα οποία τα εμφανίζουμε με το πάτημα του αντίστοιχου εικονιδίου.

Για να εισάγουμε ένα καινούργιο αριθμό τηλεφώνου ακολουθούμε τα εξής βήματα:

- Πατάμε «Insert».
- Κάνουμε την αντίστοιχη επιλογή (Εργασία, Κατοικία, Κινητό).
- Συμπληρώνουμε το αντίστοιχο πεδίο για το τηλέφωνο.
- Πατάμε «Εισαγωγή» (Εικόνα 8.16)
- Κλίνουμε το παράθυρο.



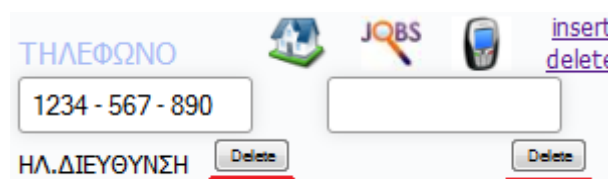
Εικόνα 8.16 Εισαγωγή τηλεφώνου

Για να εισάγουμε μια ηλεκτρονική διεύθυνση πατάμε «Insert» , όπου εμφανίζεται ένα πεδίο με κουμπί. Γράφουμε την ηλεκτρονική διεύθυνση και πατάμε «Insert» ακριβώς δίπλα δεξιά.



Εικόνα 8.17 Εισαγωγή ηλεκτρονικής διεύθυνσης

Η διαγραφή του τηλεφώνου (σχήμα 8.18) και της ηλεκτρονικής διεύθυνσης γίνεται πατώντας «Delete»



Εικόνα 8.18 Διαγραφή τηλεφώνου

8.10 Διαχείριση Βαθμίδων

Η πλήρης διαχείριση Βαθμίδων των καθηγητών μπορεί να γίνει από την καρτέλα «Διαχείριση Βαθμίδων» όπου μπορούμε να κάνουμε Εισαγωγή, Ενημέρωση και Διαγραφή (σχήμα 8.19). Θέλω να σημειώσω, ότι δεν μπορεί να γίνει διαγραφή μιας βαθμίδας, αν έστω και ένας καθηγητής έχει συγκεκριμένη βαθμίδα την οποία θέλουμε να διαγράψουμε από την βάση δεδομένων μας.

The screenshot displays the 'ΔΙΑΧΕΙΡΙΣΗ ΒΑΘΜΙΔΩΝ' (Grade Management) interface. At the top, there is a dropdown menu labeled '-Επιλογή Βαθμίδας-' with the selected value 'ΕΠΙΣΤΗΜΟΝΙΚΟΣ ΣΥΝ | ΠΛΗΡΗ'. Below this, there are three input fields: 'Βαθμίδα' (Grade) with 'ΕΠΙΣΤΗΜΟΝΙΚΟΣ ΣΥΝ' and 'ΠΛΗΡΗ', and 'Ωρες εβδομάδας' (Hours per week) with '14'. At the bottom of the form, three buttons are visible: 'Εισαγωγή' (Add), 'Ενημέρωση' (Update), and 'Διαγραφή' (Delete). The 'Εισαγωγή', 'Ενημέρωση', and 'Διαγραφή' buttons are circled in red in the original image.

Εικόνα 8.19 Διαχείριση Βαθμίδων

8.11 Διαχείριση: «Πρόγραμμα»

Όλη η βάση και η καρδιά της πτυχιακής εργασίας βρίσκεται στην «Διαχείριση Πρόγραμμα» όπου κάνουμε εισαγωγή, τροποποίηση (μετακίνηση) και διαγραφή μαθημάτων τα οποία διεξάγονται στο τμήμα μας (εικόνα 8.20).

Πτυχιακή εργασία του Σέργιου Γαβριηλίδη

ΕΞΑΜΗΝΟ * ΑΙΘΟΥΣΑ * ΚΑΘΗΓΗΤΗΣ * ΜΑΘΗΜΑ *

08-12		12-16		16-20																					
ΔΕΥΤΕΡΑ		ΤΡΙΤΗ		ΤΕΤΑΡΤΗ		ΠΕΜΠΤΗ		ΠΑΡΑΣΚΕΥΗ																	
Αιθ	Εξ	Μάθημα	Είδ	Τμ	Καθηγ 1	Καθηγ 2	Αιθ	Εξ	Μάθημα	Είδ	Τμ	Καθηγ 1	Καθηγ 2												
08-09																									
102	Δ	Βαδει	ΘΕΩΡ	T2	ΑγασΚ	Ed	101	Γ	ΑΡΑΠΕΕ	ΘΕΩΡ	T5	Λάγκθ	ΚαγκαΑ	Ed											
109	Ε	ΑνδΧρ	ΘΕΩΡ	T2	ΝιταΗ	Ed	102	Ζ	Νευδικ	ΑΣ/ΠΡ	ΑΣ/ΠΡ	ΠουΛΓ	Ed	101	ΣΤ	ΛεΣυλ	ΑΣ/ΠΡ	Θ1	Λάγκθ	ΠουΛΓ	Ed				
201	Α	ΠρΥνη	ΕΡΓ	T1	ΒαφειΑ	Ed	109	Ε	ΑνδΧρ	ΘΕΩΡ	T3	ΒαφειΑ	Ed	102	Β	ΑγγΟρ	ΘΕΩΡ	Θ1	ΒαφειΑ	Ed					
ΑΜΦ	Ε	ΑνδΧρ	ΘΕΩΡ	T1	ΛόταΑ	ΚαγκαΑ	Ed							101	Ε	ΤεχΝοη	ΘΕΩΡ	T4	ΠαπΧκω	Ed					
														109	Β	ΠρΥνη	ΕΡΓ	T4	ΚατωκΚ	Ed					
														201	ΣΤ	ΠλΚοιν	ΘΕΩΡ	T1	Αντωλ	Ed					
														201	ΣΤ	ΠλΚοιν	ΘΕΩΡ	T1	Αντωλ	Ed					
														202	Β	ΟρΑρΥνη	ΕΡΓ	T3	ΧρσοΕ	Ed					
														208	Ζ	ΗλΕμτ	ΕΡΓ	Θ1	ΜπαργΜ	Ed					
														210	Δ	ΜεΠρι	ΕΡΓ	ΑΣ/ΠΡ	ΑρολδΑ	ΔασυΜ	Ed				
														211	Χ	Μηδισθ	Χ	Θ1	ΣιαΚ	Ed					
														ΑΜΦ	Α	ΠρΥνη	ΕΡΓ	T3	ΑγασΚ	Ed					
09-10																									
101	Α	ΕισΠλη	ΘΕΩΡ	T1	ΠουΛΓ	Ed	102	Α	ΔεΞεπ	ΕΡΓ	T3	ΑδαμΠ	Ed	101	Α	ΞεΓλη	ΑΣ/ΠΡ	ΑΣ/ΠΡ	ΒαφειΑ	ΝιταΗ	Ed				
102	Α	ΕισΠλη	ΘΕΩΡ	T4	ΚαγκαΑ	Ed	201	Ζ	ΗλΕμτ	ΘΕΩΡ	ΑΣ/ΠΡ	Πέτρικ	Ed	102	Ζ	Νευδικ	ΑΣ/ΠΡ	ΑΣ/ΠΡ	ΒίτσαΒ	Ed					
109	Α	Βαδει	ΘΕΩΡ	T3	ΑραμΠ	Ed	210	Α	ΨηφΣυσ	ΘΕΩΡ	Θ1	ΑγασΚ	ΣιαΚ	Ed	109	Δ	ΜεΠρι	ΘΕΩΡ	T4	ΚουλοΒ	Ed				
201	Α	ΠρΥνη	ΕΡΓ	T2	ΒαφειΑ	Ed								101	Ε	ΤεχΝοη	ΘΕΩΡ	T4	ΠαπΧκω	Ed					
														109	Β	ΠρΥνη	ΕΡΓ	T4	ΚατωκΚ	Ed					
														201	ΣΤ	ΠλΚοιν	ΘΕΩΡ	T1	Αντωλ	Ed					
														202	Β	ΟρΑρΥνη	ΕΡΓ	T3	ΧρσοΕ	Ed					
														208	Ζ	ΗλΕμτ	ΕΡΓ	Θ1	ΜπαργΜ	Ed					
														210	Δ	ΜεΠρι	ΕΡΓ	ΑΣ/ΠΡ	ΑρολδΑ	ΔασυΜ	Ed				
														211	Χ	Μηδισθ	Χ	Θ1	ΣιαΚ	Ed					
														ΑΜΦ	Α	ΠρΥνη	ΕΡΓ	T3	ΑγασΚ	Ed					
10-11																									
102	Β	ΠρΥνη	ΕΡΓ	T1	ΒαφειΑ	Ed	102	Γ	ΑΡΑΠΕΕ	ΘΕΩΡ	T5	ΑδαμΠ	Ed	101	Ε	ΑνδΧρ	ΑΣ/ΠΡ	Θ1	ΠουΛΓ	Ed					
202	Α	ΠρΥνη	ΕΡΓ	T2	ΣιαΚ	Ed	202	ΣΤ	ΛεΣυλ	ΘΕΩΡ	ΑΣ/ΠΡ	ΧαρναΣ	ΠαπΧκω	Ed	109	Ε	ΜεΠρι	ΘΕΩΡ	T3	ΣαλαμΜ	Ed				
301	Α	ΕισΠλη	ΘΕΩΡ	T1	ΒίτσαΒ	Ed	208	Ζ	Νευδικ	ΘΕΩΡ	Θ1	ΚουλοΒ	Μάστθ	Ed	202	Ζ	ΕΒΒαδς	ΑΣ/ΠΡ	ΑΣ/ΠΡ	ΣταμάΔ	Ed				
							211	Γ	ΘεΠιστ	ΘΕΩΡ	T1	ΧρσοΕ	ΚοταΕ	Ed	211	ΣΤ	Πρωιε	ΑΣ/ΠΡ	T3	Εολ	ΚασΘστ	Ed			
							301	Α	ΨηφΣυσ	ΑΣ/ΠΡ	ΑΣ/ΠΡ	ΣιαΚ	ΑγασΚ	Ed	301	Ζ	Νευδικ	ΕΡΓ	T1	ΚεραμΕ	Ed				
							3019	Α	ΜαθΑν	ΑΣ/ΠΡ	ΑΣ/ΠΡ	ΚιωνσΚ	Ed	ΑΜΦ	Ζ	ΓρωΥτο	ΑΣ/ΠΡ	ΑΣ/ΠΡ	ΧατζόΚ	Ed					
														102	Δ	ΜεΠρι	ΘΕΩΡ	T4	ΚουλοΒ	Ed					
														109	Β	ΞεΓλη	ΑΣ/ΠΡ	Θ1	ΒαφειΑ	Ed					
														201	ΣΤ	ΠλΚοιν	ΘΕΩΡ	T2	ΜάργαΑ	Αντωλ	Ed				
														202	Α	ΨηφΣυσ	ΕΡΓ	T1	ΣιαΚ	Ed					
														208	Ε	ΑνδΧρ	ΘΕΩΡ	T4	Ψαρκων	ΦραγκΕ	Ed				
														210	Δ	Βαδει	ΘΕΩΡ	T5	ΜπαργΜ	ΣμαΑπ	Ed				
														ΑΜΦ	Α	ΠρΥνη	ΕΡΓ	T4	ΑγασΚ	Ed					
														211	Δ	ΜεΠρι	ΘΕΩΡ	T1	ΑρολδΑ	ΔασυΜ	Ed				

Εικόνα 8.20 Διαχείριση 'Πρόγραμμα'

Ενσωματώνονται πολλοί έλεγχοι για την ακεραιότητα των δεδομένων πριν εισαχθούν στο «πρόγραμμα». Έτσι, το μάθημα εξασφαλίζει ότι είναι το μοναδικό που μπορεί να διεξαχθεί στην συγκεκριμένη αίθουσα, ώρα και μέρα καθώς και το ότι ο καθηγητής δεν μπορεί να κάνει ταυτόχρονα 2 μαθήματα στην ίδια ώρα και μέρα. Όλοι αυτοί οι έλεγχοι υλοποιούνται με την χρήση των κατάλληλων SQL ερωτημάτων και Web Services ώστε ο έλεγχος να γίνεται παράλληλα μαζί με τις επιλογές μας. Επίσης, δίνεται η δυνατότητα να κάνουμε αναζήτηση με βάση κάποιων κριτηρίων , πχ με βάση το Εξάμηνο, την Αίθουσα, τον Καθηγητή και το Μάθημα, με 1, 2 ,3 ή και με τα 4 κριτήρια.

Στην παρακάτω εικόνα βλέπετε το αποτέλεσμα της αναζήτησης του μαθήματος «Προγραμματισμός Υπολογιστών 1» που διδάσκει ο καθηγητής «Βαφειάδης» στην αίθουσα 201 και εξαμήνου «Α». (εικόνα 8.21)Έτσι, με αυτόν τον τρόπο, αναζητούμε εύκολα τα δεδομένα.

Πτυχιακή εργασία του Σέργιου Γαβριηλίδη

ΕΞΑΜΗΝΟ	A	ΑΙΘΟΥΣΑ	201	ΚΑΘΗΓΗΤΗΣ	Βαφειάδης Αντώνιο	ΜΑΘΗΜΑ	Προγρ Υπολογιστών I		
08-12	12-16	16-20	ΔΕΥΤΕΡΑ		ΤΡΙΤΗ	ΤΕΤΑΡΤΗ	ΠΕΜΠΤΗ	ΠΑΡΑΣΚΕΥΗ	
Αιθ Εξ Μάθημα Ειδ Τμ Καθηγ 1 Καθηγ 2		Αιθ Εξ Μάθημα Ειδ Τμ Καθηγ 1 Καθηγ 2		Αιθ Εξ Μάθημα Ειδ Τμ Καθηγ 1 Καθηγ 2		Αιθ Εξ Μάθημα Ειδ Τμ Καθηγ 1 Καθηγ 2		Αιθ Εξ Μάθημα Ειδ Τμ Καθηγ 1 Καθηγ 2	
08-09		08-09		08-09		08-09		08-09	
201 Α ΠρΥπι ΕΡΓ Τ1 ΒαφειΑ Ed									
09-10		09-10		09-10		09-10		09-10	
201 Α ΠρΥπι ΕΡΓ Τ2 ΒαφειΑ Ed									
10-11		10-11		10-11		10-11		10-11	

Εικόνα 8.21 Αποτέλεσμα αναζήτησης

Κάτω από το πρόγραμμα υπάρχει κουμπί «ΕΙΣΑΓΩΓΗ ΜΑΘΗΜΑ» (Εικόνα 8.22)

11-12							11-12							
102	Z	ΕΘΔικι	ΕΡΓ	T2	ΜπαργΜ	Ed	109	E	Τεχ					
202	Δ	ΜηχΛογ	ΘΕΩΡ	ΑΣ/ΠΡ	Εκλ	Ed	202	ΣΤ	Λεζ					
210	B	ΠρΥπιII	ΕΡΓ	T4	ΒαφειΑ	Ed	208	Z	Νευ					
211	ΣΤ	ΔικτΗΥ	ΑΣ/ΠΡ	T3	ΧρυσοΕ	Νιτση	Ed	211	Γ	Θεφ				
ΤΗΛ	Z	ΕυφΣυ	ΘΕΩΡ	Θ1	ΚλεφτΔ	Ed	301	A	Ψηφ					
							3019	A	Μα					
<div style="border: 2px solid red; padding: 5px; display: inline-block;">ΕΙΣΑΓΩΓΗ ΜΑΘΗΜΑ</div>														

Εικόνα 8.22 Εισαγωγή μάθημα

με το πάτημα του οποίου εμφανίζεται ένα παράθυρο στο οποίο κάνουμε επιλογές να κάνουμε εισαγωγή στο «Πρόγραμμα» μας (Εικόνα 8.23). Σε περίπτωση που θέλουμε να επιστρέψουμε στις αρχικές επιλογές, δηλαδή όλα να είναι κενά, πατάμε το κουμπί

«Καθαρισμός». Αν δεν μας εμφανίσει κανένα μήνυμα λάθους όταν κάνουμε τις επιλογές μας, μπορούμε να πατήσουμε «Εισαγωγή» για να εισαχθεί το μάθημα στο «Πρόγραμμα». Αμέσως μετά , με χρήση Web Services η ιστοσελίδα θα φορτωθεί με τα καινούργια στοιχεία.

Προσοχή! Για να γίνει εισαγωγή, τα στοιχεία πρέπει να είναι μοναδικά.

Εικόνα 8.23 Εισαγωγή μαθήματος

Για την διευκόλυνση σας, δίπλα από κάθε μάθημα υπάρχει ένα μικρό υπογραμμισμένο κειμενάκι «Ed» (Εικόνα 8.24)πράσινου χρώματος.

11-12						11-12			
102	Z	ΕΘΔικι	ΕΡΓ	T2	ΜπαργΜ	Ed	109	E	Τεχ
202	Δ	Μηχ/λογ	ΘΕΩΡ	ΑΣ/ΠΡ	Εκλ	Ed	202	ΣΤ	Λε'
210	B	ΠρΥπιII	ΕΡΓ	T4	ΒαφειΑ	Ed	208	Z	Νευ

Εικόνα 8.24 Γενικές πληροφορίες μαθήματος

Πατώντας σε αυτό, βλέπουμε όλα τα στοιχεία του μαθήματος (Εικόνα 8.25)

The screenshot shows a window titled "ΔΙΔΑΣΚΕΙ" with the following fields and values:

- Καθηγητής 1: Αγάς Κωνσταντίνος
- Καθηγητής 2: *
- Αίθουσα: 102
- Ωρα: 08-09
- Μέρα: ΔΕΥΤΕΡΑ
- Μάθημα: Βάσεις Δεδομένων Ι
- Κωδικός: 3402
- Είδος: ΘΕΩΡ
- Εξάμηνο: Δ
- Όνομα Τμήματος: T2 Κρυφό
- Τομέας: ΣΥΣΤ & ΤΕΧΝΟΛ

Additional fields on the right side:

- Νέα Αίθουσα: *
- Ωρα: *
- Μέρα: *

Buttons at the bottom right: Διαγραφή, Ανανέωση, Έξοδος.

Εικόνα 8.25 Ειδικές πληροφορίες μαθήματος

Από εδώ μπορούμε να ανανεώσουμε μια εγγραφή (πχ αλλαγή καθηγητή), να διαγράψουμε η να μετακινήσουμε σε μια άλλη αίθουσα κάποιο μάθημα, επιλέγοντας ώρα και μέρα και βέβαια, περνώντας αυτομάτως από έλεγχο μοναδικότητας εγγραφής*.

(Μάθημα – μέρα – ώρα – αίθουσα είναι μοναδική εγγραφή στην βάση).

8.12 Διαχείριση: «Αναφορές»

Στον χρήστη του προγράμματος παρέχεται η δυνατότητα χρήσης “Αναφορών”. Οι αναφορές αυτές επιλέγονται από Drop Down Menu και είναι οι εξής (εικόνα 8.26) :

- Εξαμηνολόγιο
- Αιθουσιολόγιο
- Αιθουσιολόγιο(check)
- Μαθητολόγιο

- Καθηγητολόγιο
- Διδασκαλία ανά Μάθημα
- Καθηγητές ανά Μάθημα
- Διδασκαλία ανά Καθηγητή
- Ελεύθερες Αίθουσες
- Απαλλαγές Μονίμων
- Συνδιδασκαλίες
- Προσωπικά στοιχεία Καθηγητών
- Θεωρίες Μονίμων (αναλυτικά)
- Θεωρίες Εκτάκτων (αναλυτικά)
- Εργαστήρια Μονίμων (αναλυτικά)
- Εργαστήρια Εκτάκτων (αναλυτικά)
- Θεωρίες ανά βαθμίδα Καθηγητή
- Θεωρίες ανά βαθμίδα Αναπληρωτή Καθηγητή
- Θεωρίες ανά βαθμίδα Επίκουρο
- Θεωρίες ανά βαθμίδα Εφαρμογών
- Θεωρίες ανά βαθμίδα Επιστημονικό Συνεργάτη
- Εργαστήρια ανά βαθμίδα Καθηγητή
- Εργαστήρια ανά βαθμίδα Αναπληρωτή Καθηγητή
- Εργαστήρια ανά βαθμίδα Επίκουρο
- Εργαστήρια ανά βαθμίδα Εφαρμογών
- Εργαστήρια ανά βαθμίδα Επιστημονικό Συνεργάτη
- Εργαστήρια ανά βαθμίδα Εργαστηριακό Συνεργάτη
- Αναθέσεις Μονίμων
- Αναθέσεις Εκτάκτων

ΔΙΑΧΕΙΡΙΣΗ		
ΑΝΑΦΟΡΕΣ	Εξαμηνολόγιο	
Αναθέσεις Μονίμων	Αιθουσιολόγιο	
Αναθέσεις Εκτάκτων	Αιθουσιολόγιο(check)	
	Μαθητολόγιο	
	Καθηγητολόγιο	
	Διδασκαλία ανά Μάθημα	
	Καθηγητές ανά Μάθημα	
	Διδασκαλία ανά Καθηγητή	
	Ελεύθερες Αίθουσες	
	Απαλλαγές Μονίμων	
	Συνδιδασκαλίες	
	Προσωπικά στοιχεία Καθηγητών	
	Αναλυτικά Στατ. Διδασκαλίας	Θεωρίες Μονίμων (αναλυτικά)
	Συνοπτικά Στατιστικά	Θεωρίες Εκτάκτων (αναλυτικά)
		Εργαστήρια Μονίμων (αναλυτικά)
		Εργαστήρια Εκτάκτων (αναλυτικά)

Εικόνα 8.26 Αναφορές

Πρέπει να διευκρινιστεί ότι οι επιλογές των αναφορών διαφέρουν από τον χρήστη σε χρήστη. Έτσι, για παράδειγμα, ο “Προγραμματιστής” (super_admin) μπορεί να βλέπει όλες τις Αναφορές ενώ ο χρήστης χωρίς δικαιώματα (default) μόνο (εικόνα 8.27):

Εξαμηνολόγιο

Αιθουσιολόγιο

Αιθουσιολόγιο(check)

Μαθητολόγιο

Καθηγητολόγιο

Διδασκαλία ανά Μάθημα


Διδασκαλία ανά Καθηγητή

Ελεύθερες Αίθουσες


Εξαμηνολόγιο
Αιθουσιολόγιο
Αιθουσιολόγιο(check)
Μαθητολόγιο
Καθηγητολόγιο
Διδασκαλία ανά Μάθημα
Διδασκαλία ανά Καθηγητή
ΕΛΕΥΘΕΡΕΣ ΑΙΘΟΥΣΕΣ

Εικόνα 8.27 Default Αναφορές

Σε κάθε μια από αυτές τις “Αναφορές” παρέχεται δυνατότητα να αποθηκεύσει σε διάφορες μορφές (Adobe Acrobat Reader(.pdf), Microsoft Word (.doc), Microsoft Excel (.excel)) την πληροφορία απλά πατώντας το αντίστοιχο κουμπί (εικόνα 8.28).



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΕΞΑΜΗΝΟ: Χειμερινό 11

Εξάμηνο

	ΔΕΥΤΕΡΑ	ΤΡΙΤΗ	ΤΕΤΑΡΤΗ	ΠΕΜΠΤΗ	ΠΑΡΑΣΚΕΥΗ
08-09	Βάσεις Δεδομένων Ι ΘΕΩΡ 102 Αγάς Κωνσταντίνος			Μεθοδολογίες Προγραμματισμού Ι ΕΡΓ Τμήμα: ΑΣ/ΠΡ 210 Αρσλάνογλου Αχιλλέας Δασυγένης Μηνάς	Τηλεπικοινωνίες και Δίκτυα Υπολ. ΑΣ/ΠΡ 210 Μπαργούλη Μαρία Καθ. Γενικού Τμ. Άγνωστος
09-10	Βάσεις Δεδομένων Ι ΘΕΩΡ 109 Αραμπατζή Πηνελόπη		Μεθοδολογίες Προγραμματισμού Ι ΘΕΩΡ 109 Κουλούκος Βησσαρίων	Μεθοδολογίες Προγραμματισμού Ι ΕΡΓ Τμήμα: ΑΣ/ΠΡ 210 Αρσλάνογλου Αχιλλέας Δασυγένης Μηνάς	Τηλεπικοινωνίες και Δίκτυα Υπολ. ΑΣ/ΠΡ 210 Μπαργούλη Μαρία Καθ. Γενικού Τμ. Άγνωστος

Εικόνα 8.28 Παράδειγμα αναφοράς ‘Εξαμηνολόγιο’



Σε περίπτωση που θέλετε να κάνετε εξαγωγή όλων των στοιχείων (πχ. Όλα εξάμηνα ή τα στοιχεία διδασκαλίας σε όλες τις αίθουσες του τμήματος) και όχι μόνο με την βάση κάποιου κριτηρίου, επιλέγουμε το select μας να έχει αστεράκι « * », και πατάμε το αντίστοιχο κουμπί (εικόνα 8.29).

Γεια να εκτυπώσουμε όλα τα δεδομένα, προσέχουμε το <select> μας να έχει αστεράκι

Καθηγητής

Εικόνα 8.29 Εκτύπωση όλων των στοιχείων

Κάποιες Αναφορές δεν έχουν <select> γιατί οι πληροφορίες που μας παρέχουν δεν εξαρτούνται από την επιλογή κάποιου κριτηρίου (Εικόνα 8.30).


ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΞΑΜΗΝΟ: Χειμερινό 11

ΣΥΝΔΙΔΑΣΚΑΛΙΑ ΑΝΑ ΜΑΘΗΜΑ

[Εξαγωγή σε Pdf](#) [Εξαγωγή σε Word](#) [Εξαγωγή σε Excel](#)

Ανάπτυξη Διεπιφανειών Χρήστη				Εξάμηνο: Ε		Κωδικός: 3502
<u>ΕΙΔΟΣ</u>	<u>ΤΜΗΜΑ</u>	<u>ΜΕΡΑ</u>	<u>ΩΡΑ</u>	<u>ΑΙΘ.</u>	<u>ΚΑΘΗΓΗΤΗΣ 1</u>	<u>ΚΑΘΗΓΗΤΗΣ 2</u>
ΘΕΩΡ	T1	ΔΕΥΤΕΡΑ	08-09	ΑΜΦ	Λότσιου Αντωνία	Καγκάνη Αικατερίνη
ΘΕΩΡ	T2	ΔΕΥΤΕΡΑ	12-13	208	Ψαρράς Κωνσταντίνος	Φραγκιαδάκη Ελένη
ΘΕΩΡ	T2	ΔΕΥΤΕΡΑ	13-14	208	Ψαρράς Κωνσταντίνος	Φραγκιαδάκη Ελένη

Εικόνα 8.30 Αναφορά 'Συνδιδασκαλία ανά μάθημα'

Αυτές είναι :

- Διδασκαλία ανά Μάθημα
- Καθηγητές ανά Μάθημα
- Διδασκαλία ανά Καθηγητή
- Ελεύθερες Αίθουσες
- Απαλλαγές Μονίμων
- Συνδιδασκαλίες
- **Αναλυτικά Στατ. Διδασκαλίας**
 - Θεωρίες Μονίμων (αναλυτικά)
 - Θεωρίες Εκτάκτων (αναλυτικά)
 - Εργαστήρια Μονίμων (αναλυτικά)
 - Εργαστήρια Εκτάκτων (αναλυτικά)
- **Συνοπτικά Στατιστικά**

- Θεωρίες ανά βαθμίδα Καθηγητή
- Θεωρίες ανά βαθμίδα Αναπληρωτή Καθηγητή
- Θεωρίες ανά βαθμίδα Επίκουρο
- Θεωρίες ανά βαθμίδα Εφαρμογών
- Θεωρίες ανά βαθμίδα Επιστημονικό Συνεργάτη
- Εργαστήρια ανά βαθμίδα Καθηγητή
- Εργαστήρια ανά βαθμίδα Αναπληρωτή Καθηγητή
- Εργαστήρια ανά βαθμίδα Επίκουρο
- Εργαστήρια ανά βαθμίδα Εφαρμογών
- Εργαστήρια ανά βαθμίδα Επιστημονικό Συνεργάτη
- Εργαστήρια ανά βαθμίδα Εργαστηριακό Συνεργάτη
- Αναθέσεις Μονίμων
- Αναθέσεις Εκτάκτων

9. Συμπεράσματα - Προτάσεις

Η χρήση του IBM Data Studio είναι γενικά βατή και χωρίς ιδιαίτερα σημεία προσοχής επιλογών, με αποτέλεσμα να διευκολύνει τη δημιουργία των Data Web Services.

Παρόλα αυτά όμως το γεγονός ότι δεν γράφουμε αναγκαστικά τον κώδικα για κάθε ένα από αυτά και, το ίδιο σημαντικό, δεν μπορούμε να τον δούμε μας περιορίζει σε συγκεκριμένους τύπους Data Web Service, η χρήση τους δηλαδή είναι πάντα με τον προβλεπόμενο τρόπο του Data Studio.

Έχω καταλήξει στο συμπέρασμα ότι ήταν αναγκαίο ο προγραμματιστής να έχει τη δυνατότητα να τροποποιεί τον κώδικα ώστε να μπορεί να κάνει αλλαγές ή ακόμα και να προσθέσει επιπλέον λειτουργικότητα σε κάποια Data Web Service που έφτιαξε.

Πιο συγκεκριμένα, αντιμετώπισα πρόβλημα κατά την δημιουργία κάποιου ιδιαίτερα πολύπλοκου Web Service από μεριάς διακλαδώσεων αποφάσεων (if then else), καθώς η SQL εντολή που θα ολοκληρωνόταν και εκτελούνταν ορθά σε ένα SQL σύστημα, στο Web Service παρήγαγε λάθος με αποτέλεσμα την αποφυγή χρήσης διακλαδώσεων if then καθώς δεν μπορούσα να δώ, ούτε να κάνω debug το σημείο που προκαλούσε το λάθος. Για να γίνει ποιο κατανοητή η φύση του προβλήματος με μια απλή if then το Web Service λειτουργούσε, με μια σύνθετη - όχι.

Ένα άλλο σοβαρό πρόβλημα είναι η ίσως μέχρι και σημείου παραμέλησης των character encoding υποστήριξης όσων αφορά τα Web Services. Αυτό παρατηρήθηκε όταν έπρεπε να δημιουργήσω συσχετίσεις και να πάρω αποτελέσματα από ελληνικά κείμενα. Πιο συγκεκριμένα ο τύπος UTF-8 της βάσης δεν βοηθούσε όταν δεδομένα που πέρασαν από POST ή GET είχαν υποστεί ISO encoding, καθώς και με το γεγονός ότι δεν υπάρχει σαφής τρόπος στον προγραμματισμό που να μπορεί να ξεχωρίζει 2 διαφορετικά encoding που εμφανίζουν σωστά μια λέξη σε άλλη, πέρα της αγγλικής γλώσσας. Και εδώ προς αποφυγή παρατηρήσεων ότι το λάθος αυτό οφείλεται σε απειρία προγραμματιστή, να δηλώσω ότι με τον standard connector της php IBM DB2 τα δεδομένα ήταν άψογα, που συμπερασματικά οδηγεί στις παραπάνω παρατηρήσεις μου.

Ένα άλλο σημείο είναι η χρήση των Web Services με γλώσσες όπως η PHP όπου τα δεδομένα δεν έχουν συγκεκριμένο τύπο, π.χ. String, varchar κ.λ.π, αυτό έχει σαν αποτέλεσμα να χάνεται η πληροφορία αυτή κατά την διάρκεια που επιστρέφονται αποτελέσματα από την βάση. Αυτό μπορεί μερικές φορές να οδηγήσει σε καταστροφικά λάθη καθώς και είναι από μόνο του θέμα ασφάλειας που μπορεί να δώσει δυνατότητα σε κάποιον γνώστη να περάσει λανθασμένα ή και κακόβουλα στοιχεία σε έναν πίνακα. Για αυτό πρέπει ο προγραμματιστής, είτε να έχει πάντα hardcoded ελέγχους για κάθε στοιχείο κάθε πίνακα, είτε να ζητάει και να αποθηκεύει τα metadata του πίνακα για να μπορέσει να ελέγχει την πορεία του προγράμματος.

Συνεχίζοντας στο θέμα τις ασφάλειας, οι υπηρεσίες που δημιουργούνται με το Data Studio δεν μπορούν να υποστηρίξουν εγγενή ασφάλεια (Kerberos πχ). Επίσης, προκαλεί και πρόβλημα η επίσης εγγενής έλλειψη δυνατότητας απάντησης ενός Web Service σε μορφή JSONP, γεγονός που αναγκάζει έναν προγραμματιστή να δημιουργήσει ο ίδιος έναν JSONP parser για να τις καλέσει μέσω AJAX. Προσωπικά όπως και έκανα. Σίγουρα είναι μια έλλειψη που θα έπρεπε να υπάρχει.

Γενικά η χρήση των Data Web Services μας κάνει την ζωή πιο εύκολη καθώς ο προγραμματιστής δεν χρειάζεται να βάζει τον κώδικα της σύνδεσης σε μια βάση δεδομένων όταν δημιουργεί μια ιστοσελίδα που τα χρησιμοποιεί. Κάθε συναλλαγή με τη βάση γίνεται με κλήση μεθόδων μιας web service.

Δυστυχώς όμως η δημιουργία ενός Web Service δεν είναι πανάκεια για αυτό το πρόβλημα, καθώς πολλές φορές τα statement, και execute που θα αναγκαζόταν να γράψει κάποιος θα δούλευαν πολύ καλύτερα, και ακόμα και αν μιλούσαμε για κάποιο μεγαλύτερο πρόγραμμα από αυτό της πτυχιακής μου ή για πιο σωστά δομημένη βάση. Ακόμα και τότε σύνθετα Web Services θα προκαλούσαν μεγαλύτερο πονοκέφαλο από αυτόν που λύνουν, καθώς σχεδόν ποτέ δεν θα λειτουργούσαν στο ακέραιο, οδηγώντας ίσως το ίδιο το πρόγραμμα σε αχρηστία, κάτι που αναγκάζει τον προγραμματιστή να δημιουργήσει πολλά και μικρά Web Services, με αποτέλεσμα να προσθέτει overhead στο δίκτυο του χρήστη. Παράδειγμα είναι η εισαγωγή ενός % σε οποιοδήποτε κείμενο, η format, το οποίο οδηγεί ακόμα και το πιο απλό Web Service σε απραξία, ενώ είναι προφανές Escape Char που χρησιμοποιείται σε ερωτήματα SQL.

Οι Data Web Services εμπίπτουν σε νόμους περί προστασίας προσωπικών δεδομένων και αυτό γιατί δεν υποστηρίζονται ακόμα πολιτικές ασφαλείας από το ίδιο το Data Studio εγγενώς. Θα μπορούσαν να χρησιμοποιηθούν για το TEI λοιπόν, σε δεδομένα που δεν εμπεριέχουν πολιτική απορρήτου όπως π.χ. οι μέσοι όροι μαθημάτων ή το πρόγραμμα μαθημάτων. Θα ήταν απαγορευτικές όμως για δεδομένα όπως προσωπικά στοιχεία φοιτητών, βαθμολογίες και απουσίες που είναι προσωπικά δεδομένα κάθε φοιτητή.

Τελική παρατήρηση όσον αφορά την δημιουργία ενός Web Service, είναι το κατά πόσο σε αναγκάζει να κάνεις δομές στην βάση από την οποία χρειάζεσαι τα δεδομένα. Συγκεκριμένα ίσως χρειαστεί να δημιουργήσεις Data Structures για να χειριστείς 2 πίνακες, κάτι που επαγγελματικά θα πρόσδιδε κόστος και χρόνο σε ένα Project. Υπάρχει συνεπώς η πιθανότητα να προσθέσεις Overhead και στο ίδιο το Project εάν επιλέξεις το Data Web Services για υπηρεσία Web Services.

Συνοψίζοντας, η IBM κατά τη γνώμη μου απαιτεί από τον προγραμματιστή να είναι αρκετά σχετικός με τις υπηρεσίες που προσφέρει και σίγουρα να έχει εμπειρία στον χειρισμό δεδομένων με οποιονδήποτε τρόπο. Προσφέρει κάτι που είναι διαφορετικό από άλλες λύσεις και το κάνει πολύ καλά μέχρι ένα σημείο, αλλά θα χρειαστεί κάποιος πολύ καλός γνώστης χειρισμού των προγραμμάτων της για να ανταπεξέλθει σε πραγματικές συνθήκες λειτουργίας, η οποία όπως έχει αποδείξει η DB2 μπορεί να γίνει πάρα πολύ σταθερή.

10. Βιβλιογραφία

- 1) Κεραμόπουλος Ευκλείδης (2009) , Σημειώσεις μαθήματος Βάσεις Δεδομένων II, ΑΤΕΙ Θεσσαλονίκης
- 2) Πτυχιακή Εργασία του Μάνιου Χρήστο (2011)
- 3) Adam Faeth, Dimple Bhatia, Vinod Chirayath, Praveen Ghantasala, Hassi Norlen, Hardik Patel & Daniel Zilio. Getting Started with IBM Data Studio 3.1 for DB2
<http://www.ibm.com/developerworks/wikis/display/db2oncampus/FREE+ebook+-+Getting+started+with+IBM+Data+Studio+for+DB2>
- 4) Build a Support Knowledge Base using DB2 pureXML and PHP.19 Nov ember 2009.
<http://www.ibm.com/developerworks/data/tutorials/dm-0911db2purexmlphp/section2.html>
- 5) Bruce Schneier . (15 Ινπλίνπ 2000). Security Risks of Unicode. In Crypto-Gram Newsletter. . Retrieved July 15, 2000, from
<http://www.schneier.com/crypto-gram-0007.html#9>
- 6) Connecting to a DB2 database with PHP (ibm_db2)
<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.apdv.php.doc/doc/t0023132.htm>
- 7) Eric Newcomer Understanding Web Services: XML, WSDL, SOAP, and UDD
Publisher: Addison-Wesley Professional; 1 edition (May 23, 2002)
- 8) Justin Whitney. Making the Most of PHP with DB2 (September 30, 2009)
<http://www.devx.com/ibm/Article/29366#>
- 9) Dan Scott .(03 Jul 2001).Connecting PHP applications to IBM DB2 Universal Database
http://www.ibm.com/developerworks/data/library/techarticle/scott/0614_scott.html
- 10) Josh Bush.Masked Input Plugin
<http://digitalbush.com/projects/masked-input-plugin/>
- 11) John Resig. Tutorials:How jQuery Works. In jQuery Documentation. Retrieved 15 december 2011, from
http://docs.jquery.com/How_jQuery_Works
- 12) Michael “BuzzLY” Glass, Yann “Bunkermaster” Le Scouarnec , Elizabeth Naramore , Gary “trooper” Mailer,Jeremy “stolzboy” Jason “goldbug” Gerner. Beginning PHP, Apache, MySQL Web Development. Publisher: Wrox (March 12, 2004)
- 13) Michael L. Pauser.(Dec 2008)IBM Data Studio Data Web Services, Part 1: IBM Data Studio: Get started with Data Web Services
<http://www.ibm.com/developerworks/data/tutorials/dm0711pauser/index.html>

- 14) Marc von Brockdorff. How to make your site Multilingual using XML and jQuery
<http://www.webgeekly.com/tutorials/jquery/how-to-make-your-site-multilingual-using-xml-and-jquery/>
- 15) Paul C. Zikopoulos. Featured Database Articles(March 22, 2006)
<http://www.databasejournal.com/features/article.php/3593431/DB2-Series.htm>
- 16) Seda Özses. Very simple login using Perl, jQuery, Ajax, JSON and MySQL. Retrieved 25 Jan 2011, from
<http://www.ibm.com/developerworks/webservices/library/ws-simplelogin/index.html?ca=drs->
- 17) IBM DB2, WebSphere, IMS, U2, etc (forum)
<http://www.sql.ru/forum/>
- 18) SQL Tutorial
<http://www.w3schools.com/sql/default.asp>
- 19) Introducing JSON
<http://json.org/>
- 20) jQuery Tutorials and examples
<http://www.roseindia.net/ajax/jquery/>
- 21) TCPDF is a FLOSS PHP class for generating PDF documents.
<http://www.tcpdf.org/index.php>
- 22) wkhtmltopdf -Convert html to pdf using webkit (qtwebkit)
<http://code.google.com/p/wkhtmltopdf/>
- 23) createDocx | PHPDOCX
<http://www.phpdocx.com/demo>
- 24) PHPExcel
<http://phpexcel.codeplex.com/>

Αναφορές

- [1] <http://www.v7n.com/forums/marketing-forum/25602-web-applications-vs-desktop-applications.html>
- [2] https://developer.mozilla.org/en/Properly_Configuring_Server_MIME_Types
- [3] <https://help.ubuntu.com/8.04/serverguide/C/apt-get.html>
- [4] <http://sourceforge.net/projects/xampp/>
- [5] <http://www.wampserver.com/en/>
- [6] <http://www.mamp.info/en/index.html>
- [7] <http://www.tutorialchip.com/php/history-of-php-objects-f1-to-php5/>

[8] <http://php.net/manual/en/security.database.sql-injection.php>

[9] <https://phpids.org/>

[10] <http://phpids.org/docs/>

[11] www.apachefriends.org

[12] www.mysql.net

[13] www.php.net

[14] <http://www.w3.org/TR/soap/>

[15] <http://help.pathfinder.gr/account/security/cookies2.html>

[16] <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.dm.doc/doc/t0004575.html>