ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

ΟΙΚΟΝΟΜΙΚΩΝ ΚΑΙ ΚΟΙΝΩΝΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΜΕΤΑΠΤΥΧΙΑΚΟ ΣΤΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

ΣΥΓΧΡΟΝΙΣΜΟΣ ΑΠΟΘΗΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΟΥ ΚΑΤΑΣΤΗΜΑΤΟΣ ΜΕ WEB SERVICES.

Η ΠΕΡΙΠΤΩΣΗ ΤΟΥ webERP ΚΑΙ ΤΟΥ osCommerce

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Του

ΜΙΣΙΡΚΑ ΑΘΑΝΑΣΙΟΥ
ΑΜ:33/05

Επιβλέπων :ΝΙΚΟΛΑΟΣ ΠΡΩΤΟΓΕΡΟΣ
Επίκουρος Καθηγητής

Θεσσαλονίκη, Φεβρουάριος 2009

Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η θεωρητική και πρακτική μελέτη και ανάλυση μιας σχετικά καινούριας τεχνολογίας : των Web Services. Στην ουσία, βέβαια, δεν πρόκειται για μια τεχνολογία αλλά για ένα σύνολο επιμέρους τεχνολογιών που αρμονικά συνδυασμένες δίνουν το επιθυμητό αποτέλεσμα. Πίσω από όλες αυτές βρίσκεται η XML μια μετα-γλώσσα η οποία μπορεί να χρησιμοποιηθεί για να περιγράψει έννοιες και δεδομένα.

Το σημαντικότερο πλεονέκτημα των Web Services είναι η διαλειτουργικότητά τους: το γεγονός ότι μπορούν να γεφυρώσουν συστήματα δομημένα σε διαφορετικές γλώσσες προγραμματισμού που τρέχουν σε διαφορετικές πλατφόρμες. Στην συγκεκριμένη εργασία, ωστόσο, απλά θα εξετάσουμε τη λειτουργία τους ελέγχοντας την επικοινωνία ανάμεσα σε δύο συστήματα, ένα ηλεκτρονικό κατάστημα (osCommerce) και ένα ERP (webERP), αμφότερα δομημένα σε γλώσσα PHP.

Λέξεις Κλειδιά: XML, Web Services, Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL),Universal Discovery Description and Integration (UDDI), osCommerce, webERP.

ΠΕΡΙΕΧΟΜΕΝΑ

1 ΕΙΣΑΓΩΓΗ…….. 3

1.1 Γενικά…… 3

1.2 Περιγραφή προβλήματος……. 3

2 ΤΙ ΕΙΝΑΙ ΕΝΑ ERP ΣΥΣΤΗΜΑ……. 5

2.1 Ορισμοί ERP……. 5

2.2. Ιστορική εξέλιξη των ERP…….. 6

2.3 Αρχιτεκτονική των ERP……... 8

2.4 Πλεονεκτήματα-μειονεκτήματα……... 10

2.5 webERP…….. 12

2.5.1 Τι είναι το webERP……..... 12

2.5.2 Η αξιοπιστία σαν παράγοντας ποιότητας και σχεδιασμού……... 12

2.5.3 Πλεονεκτήματα webERP……... 13

2.5.4 Χαρακτηριστικά webERP……. 14

3 ΤΟ osCommerce……. 18

3.1 Τι είναι το osCommerce…… 18

4 ΤΑ WEB SERVICES……... 21

4.1 Ορισμός για τα web services…… 21

4.2 Το παρελθόν των web services……. 22

4.3 Το σήμερα…….. 23

4.4 Πίσω από τα web services……. 25

4.4.1 XML (Extensible Markup Language)…….. 27

4.4.2 SOAP (Simple Object Access Protocol)……... 29

4.4.3 WSDL (Web Service Description Language)……. 30

4.4.4 UDDI (Universal Discovery Description and Integration)…….. 32

4.4.5 Συνδυάζοντας όλα τα παραπάνω…….. 33

4.4.6 Πλεονεκτήματα των web services……. 34

4.4.7 Μειονεκτήματα των web services……. 35

5 ΥΛΟΠΟΙΗΣΗ……. 36

5.1 Μια πρώτη προσέγγιση…… 36

5.2 Μελέτη εφικτότητας……. 38

5.3 NuSoap – REST……. 42

5.4 Σενάριο χρήσης…….. 43

6 ΣΥΜΠΕΡΑΣΜΑΤΑ…….. 50

7 ΠΑΡΑΡΤΗΜΑ Α…… 52

7.1 Επικοινωνία από webERP προς osCommerce…… 52

7.1.1 Ο κώδικας του server…… 52

7.1.2 Το WSDL αρχείο…… 54

7.1.3 Τα μηνύματα SOAP……... 57

7.1.4 Ο κώδικας του client…….. 59

7.2 Επικοινωνία από osCommerce προς webERP…… 62

7.2.1 Ο κώδικας του server…… 62

7.2.2 Το WSDL αρχείο…… 64

7.2.3 Τα μηνύματα SOAP……... 67

7.2.4 Ο κώδικας του client……. 69

8 ΠΑΡΑΡΤΗΜΑ Β…… 71
8.1 Εγκατάσταση του osCommerce…….. 71
8.2 Εγκατάσταση του webERP…….. 75
9 ΒΙΒΛΙΟΓΡΑΦΙΑ…….……... 76

1. ΕΙΣΑΓΩΓΗ

1.1. Γενικά

Είναι αλήθεια ότι ζούμε στην κοινωνία της πληροφορίας που στην πράξη μεταφράζεται σε εύκολη και ταχύτατη πρόσβαση σε δεδομένα χωρίς η απόσταση να αποτελεί περιορισμό. Αυτό είναι δυνατό κατά κύριο λόγο μέσα από την εξέλιξη των τεχνολογιών των υπολογιστών και των δικτύων. Το πρόβλημα της εργασίας αυτής εντάσσεται σε αυτό το γενικότερο πλαίσιο. Αποτελεί μια προσπάθεια πρόσβασης σε πληροφορία από ένα σύστημα απομακρυσμένο σε ένα άλλο.

1.2 Περιγραφή προβλήματος

Σαν γενική περιγραφή του προβλήματος μπορούμε να θεωρήσουμε την προσπάθεια συγχρονισμού δύο βάσεων δεδομένων. Ειδικότερα, η εργασία αφορά τον συγχρονισμό της βάσης δεδομένων ενός ERP συστήματος με αυτήν ενός ηλεκτρονικού καταστήματος μιας επιχείρησης. Θεωρούμε ότι η επιχείρηση διατηρεί δύο βάσεις δεδομένων, μια για το ηλεκτρονικό κατάστημα και μια για το ERP σύστημα. Κάτω από αυτές τις συνθήκες όταν εκτελείται μια αγορά μέσω Internet γίνεται χρήση της βάσης δεδομένων του ηλεκτρονικού καταστήματος ενώ αντίστοιχα όταν ο πελάτης επισκέπτεται το «φυσικό» κατάστημα η αγορά γίνεται μέσω της βάσης δεδομένων του ERP. Γίνεται, επομένως, αντιληπτό ότι αν δεν ενημερώνονται και οι δύο βάσεις σε κάθε δοσοληψία, είτε γίνεται μέσω ERP είτε γίνεται μέσω Internet, το αποτέλεσμα θα είναι η κάθε μια να έχει τα δικά της δεδομένα που σε καμία περίπτωση δεν θα αντιστοιχούν στα πραγματικά.

Στην παρούσα εργασία επικεντρωνόμαστε στο να συγχρονιστούν οι ποσότητες των προϊόντων. Επομένως για κάθε ενημέρωση της ποσότητας ενός χ προϊόντος στην βάση δεδομένων του ERP (που σημαίνει ότι ο πελάτης αγόρασε το προϊόν από το φυσικό κατάστημα) πρέπει να γίνεται και μία ενημέρωση της ποσότητας του ίδιου προϊόντος στην βάση δεδομένων του ηλεκτρονικού καταστήματος και αντίστροφα. Κατ' αυτόν τον τρόπο και οι δύο βάσεις παραμένουν ενημερωμένες και σύμφωνες με τις πραγματικές ποσότητες στην αποθήκη.

Για την υλοποίηση της εργασίας χρησιμοποιήθηκαν προγράμματα ανοιχτού κώδικα. Συγκεκριμένα, σαν ERP σύστημα χρησιμοποιήθηκε το WebERP το οποίο είναι ένα

ολοκληρωμένο σύστημα λογιστικών – ERP. Έχει web interface και απαιτεί, επομένως , μόνο έναν web browser για να τρέξει ενώ ο προγραμματισμός του έγινε σε γλώσσα php. Για το ηλεκτρονικό κατάστημα έγινε χρήση του os-commerce ενός online καταστήματος το οποίο διανέμεται δωρεάν με την άδεια GNU (general public license). Το osCommerce είναι δομημένο με την γλώσσα php. Στην πράξη, λοιπόν, στην εργασία αυτή γίνεται μια προσπάθεια συντονισμού των δύο αυτών συστημάτων. Ερχόμενος ο πελάτης στο φυσικό κατάστημα και αγοράζοντας ένα προϊόν, με το που κόβεται το σχετικό παραστατικό ενημερώνεται και η βάση του WebERP. Ταυτόχρονα πρέπει να γίνει και η ενημέρωση της βάσης του osCommerce ώστε να ανταποκρίνεται και αυτή στις πραγματικές ποσότητες. Για τη λύση του προβλήματος αυτού χρησιμοποιήθηκε η τεχνολογία των web services. Χρησιμοποιώντας τα web services θα ενημερώνεται αυτόματα η βάση δεδομένων του ηλεκτρονικού καταστήματος κάθε φορά που ενημερώνεται η βάση του ERP.

2.ΤΙ ΕΙΝΑΙ ΕΝΑ ERP ΣΥΣΤΗΜΑ

2.1 Ορισμοί ERP

Τα γράμματα E R P αποτελούν την αρχή των λέξεων Enterprise Resource Planning ελληνιστί διαχείριση επιχειρηματικών πόρων. Είναι ένας όρος που χρησιμοποιείται για να περιγράψει ένα σύστημα λογισμικού το οποίο αποτελείται από ένα σύνολο επιμέρους τμημάτων (modules) που καλύπτουν μια ευρεία γκάμα επιχειρησιακών λειτουργιών όπως είναι οι χρηματοοικονομικές λειτουργίες – λογιστικά, η παραγωγή, η διαχείριση ανθρωπίνων πόρων και τα logistics

Ας δούμε πώς ορίζουν το ERP σύστημα διάφοροι συγγραφείς :

- Ένα ERP σύστημα μπορεί να θεωρηθεί ως ένα εταιρικό και ευρέων πληροφοριών σύστημα που ολοκληρώνει όλες τις πλευρές της επιχειρηματικής δραστηριότητας. Προϋποθέτει μια βάση δεδομένων, μια εφαρμογή, και μια ενιαία διεπαφή για όλη την επιχείρηση. (Bingi et al.,1999, σελ.8)
- Τα ERP συστήματα είναι εξαιρετικά ολοκληρωμένα και μεγάλου εύρους εταιρικά πληροφοριακά συστήματα (πακέτα λογισμικού) που αυτοματοποιούν εταιρικές λειτουργίες πυρήνα όπως χρηματοοικονομικές, ανθρωπίνων πόρων, παραγωγής και εφοδιασμού και διανομής. Holland et al., 1999a, σελ,289 και Holland et al., 1999b, σελ273.
- ERP είναι ένα ολοκληρωμένο πακέτο εφαρμογών λογισμικού σχεδιασμένο για να αυτοματοποιήσει και να ολοκληρώσει όλες τις επιχειρηματικές δραστηριότητες μιας εταιρείας σε όλη την αλυσίδα εφοδιασμού και να παρέχει άμεση πρόσβαση στην πληροφορία που έχει η επιχείρηση στη διάθεσή της. Τα ERP συστήματα μπορούν να θεωρηθούν σαν ευρείας χρήσης και γενικού σκοπού πληροφοριακά συστήματα διοίκησης (MIS-management Information Systems) για τις επιχειρήσεις. Maher, 1999, σελ.36.

- Τα ERP συστήματα, ένα είδος ευρέων εταιρικών πληροφοριακών συστημάτων (EWIS- Enterprise-Wide Information System) αντιπροσωπεύουν σύνολα επιχειρηματικών εφαρμογών που επιτρέπουν σε έναν οργανισμό μια μεγάλη γκάμα λειτουργιών διαχείρισης. Τα ERP συστήματα μπορούν να θεωρηθούν σαν εργαλεία βελτιστοποίησης και ολοκλήρωσης των επιχειρηματικών διαδικασιών κατά μήκος της αλυσίδας εφοδιασμού (μέσα και έξω από τον οργανισμό) υλοποιημένα με βάση τα σύγχρονα πληροφοριακά συστήματα διοίκησης. Al-Mashari, 2000, σελ.3.
- Τα ERP συστήματα είναι μεγάλης κλίμακας διαλειτουργικά και ολοκληρωμένα συστήματα. Brown et al., 2000, σελ.1029
- Τα ERP συστήματα είναι πακέτα λογισμικού που ενσωματώνουν πληροφορία που υπάρχει σε ολόκληρη την επιχείρηση. Αυτή η ενσωμάτωση αποτρέπει την ύπαρξη αντιφάσεων στην επιχειρηματική πληροφορία και επιτρέπει στον οργανισμό να έχει συνεκτικές και ολοκληρωμένες αναφορές και εκθέσεις. Shakir, 2000, σελ.1033.

2.2 Ιστορική εξέλιξη των ERP

Η εξέλιξη των ERP συστημάτων ακολούθησε στενά την εξέλιξη του υλικού και του λογισμικού στους υπολογιστές. Στην δεκαετία του 60’ οι περισσότεροι οργανισμοί σχεδίαζαν, ανέπτυσσαν και εφάρμοζαν υπολογιστικά συστήματα, αυτοματοποιώντας κατά κύριο λόγο τη λειτουργία ελέγχου των αποθεμάτων, με τη χρήση πακέτων ελέγχου αποθεμάτων (IC – Inventory Control). Αυτά τα συστήματα ήταν βασισμένα σε γλώσσες προγραμματισμού όπως η COBOL, η ALGOL και η FORTRAN. Τα συστήματα σχεδιασμού απαιτήσεων υλικών (MRP – Material Requirements Planning) αναπτύχθηκαν την δεκαετία του 70’ και περιελάμβαναν κυρίως τον προγραμματισμό της παραγωγής ή απαραίτητων εξαρτημάτων της, σύμφωνα πάντα με το βασικό πρόγραμμα παραγωγής. Ακολουθώντας αυτή την πορεία νέα συστήματα λογισμικού που ονομάζονταν συστήματα σχεδιασμού πόρων παραγωγής (MRP II – Manufacturing Resources Planning) εισήχθηκαν την δεκαετία του 80’. Αυτά έδιναν ιδιαίτερη έμφαση στην βελτιστοποίηση των διαδικασιών παραγωγής συγχρονίζοντας τις ποσότητες στις πρώτες ύλες με τις ανάγκες τις παραγωγής. Τα MRP II συστήματα περιελάμβαναν στο πακέτο και λειτουργίες όπως η διαχείριση των διανομών (distribution management), του έργου (project management), των ανθρωπίνων πόρων

(human resources) καθώς και οικονομική διαχείριση. Τα ERP συστήματα πρωτοεισήχθησαν στα τέλη την δεκαετίας του 80’ και στην αρχή του 90’ με το πλεονέκτημα της ολοκλήρωσης και του κεντρικού ελέγχου μιας ευρείας γκάμας λειτουργιών της επιχείρησης. Βασισμένα στα τεχνολογικά θεμέλια των συστημάτων MRP και MRP II τα ERP ολοκλήρωναν σε ένα σύστημα επιχειρησιακές λειτουργίες όπως αυτή της παραγωγής, της διανομής, των οικονομικών, της διαχείρισης ανθρωπίνων πόρων, έργου και αποθεμάτων παρέχοντας πρόσβαση και συνέπεια στην πληροφορία σε ολόκληρη την επιχείρηση. Κατά την διάρκεια της δεκαετίας του 90’ οι κατασκευαστές των ERP πρόσθεσαν επιπλέον λειτουργίες με τη μορφή add-on στον πυρήνα των λειτουργιών δημιουργώντας τα extended ERP. Αυτά τα συστήματα περιελάμβαναν δυνατότητες προχωρημένου σχεδιασμού και προγραμματισμού, λύσεις e-bussiness όπως η διαχείριση σχέσεων πελατών (CRM – Customer Relationship Management) και η διαχείριση αλυσίδας εφοδιασμού (SCM – Supply Chain Management).

Σχήμα 1

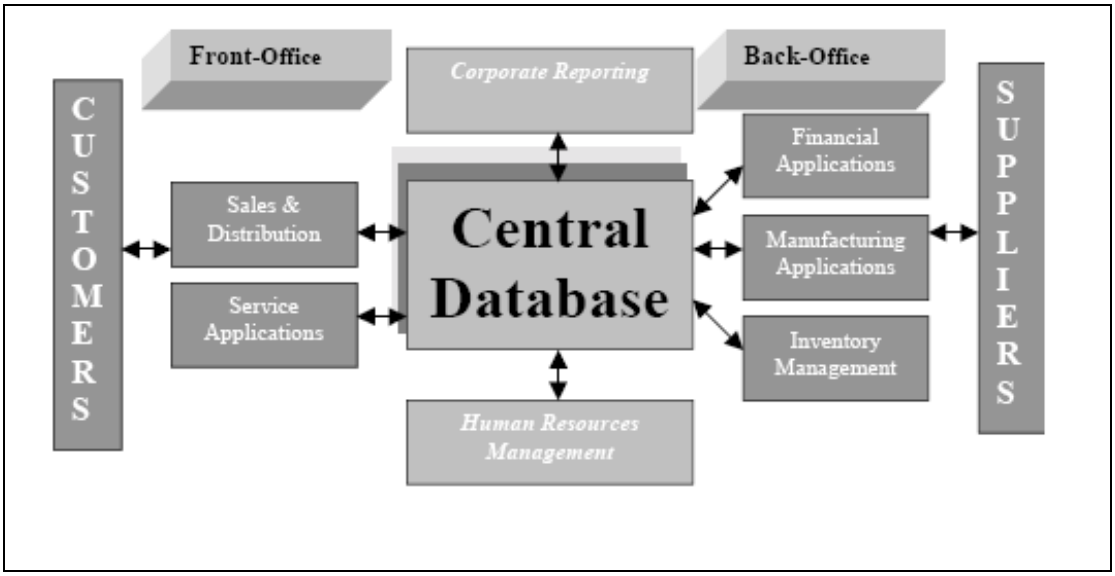
2.3 Αρχιτεκτονική των ERP

Τα ERP συστήματα έχουν τα εξής χαρακτηριστικά :

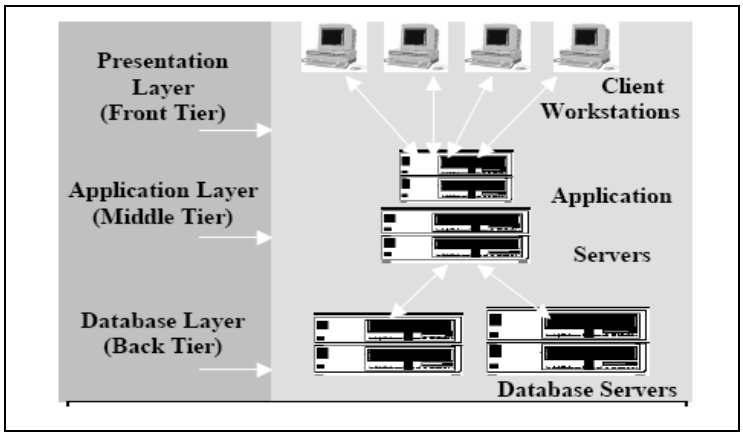
- Αρθρωτό σχεδιασμό που να περιλαμβάνει πολλά διαφορετικά τμήματα βάση των επιχειρησιακών λειτουργιών όπως οικονομικά, παραγωγή, διανομή, λογιστικά κτλ.
- Χρήση ενός κεντρικού και κοινού συστήματος βάσεων δεδομένων
- Τα διάφορα κομμάτια τους παρέχουν τη δυνατότητα ανεμπόδιστης ροής της πληροφορίας μεταξύ τους, αυξάνοντας την λειτουργική διαπερατότητα μέσα από τη χρήση συγκεκριμένων διεπαφών.
- Είναι συνήθως πολύπλοκα συστήματα μεγάλου κόστους
- Είναι ευέλικτα και βελτιώνουν τις λειτουργίες της επιχείρησης
- Απαιτούν πολύ χρόνο για την παραμετροποίηση και προσαρμογή τους στις επιχειρηματικές λειτουργίες της εταιρίας
- Εκμεταλλεύονται τις δυνατότητες του διαδικτύου

Οι κατασκευαστές ERP προσφέρουν συστήματα με κάποιο βαθμό διαφορετικότητας αλλά ο βασικός πυρήνας λειτουργιών είναι σχεδόν ίδιος για όλα. Μερικές από τις βασικές λειτουργίες που εμπεριέχουν σχεδόν όλα τα ERP είναι :

- Λογιστική διαχείριση
- Χρηματοοικονομική διαχείριση
- Διαχείριση παραγωγής
- Διαχείριση μεταφορών
- Διαχείριση δικτύου πωλήσεων και διανομών
- Διαχείριση ανθρωπίνων πόρων
- Διαχείριση αλυσίδας εφοδιασμού
- Διαχείριση σχέσεων πελατών (CRM – customer relationship system)
- E-bussiness



Τα εταιρικά συστήματα εφαρμόζουν τεχνολογίες είτε Thin Client/Server (C/S) είτε Client/Fat Server (C/Fs) δημιουργώντας ένα αποκεντρωτικό υπολογιστικό περιβάλλον. Σε ένα C/S σύστημα ένας αριθμός τερματικών συσκευών (desktop pc) αιτούνται υπηρεσίες στους Application Servers οι οποίοι με τη σειρά τους παίρνουν τα απαιτούμενα για την υπηρεσία δεδομένα από τους Database Servers. Η γενική πρακτική είναι να ακολουθείται αρχιτεκτονική 3-tier όπως φαίνεται και στο σχήμα 2.



Σχήμα 2

Στα συστήματα Client/Server οι λειτουργίες επιτελούνται ακολουθώντας 3 επίπεδα λογικής όπως φαίνεται και από το σχήμα 2:

- Το επίπεδο παρουσίασης:Το γραφικό περιβάλλον διεπαφής για εισαγωγή δεδομένων και πρόσβαση στις λειτουργίες του συστήματος.

- Το επίπεδο εφαρμογής: Οι επιχειρηματικοί κανόνες, οι λειτουργίες, η λογική και τα προγράμματα για την ανταλλαγή δεδομένων από και προς τους Database Servers.
- Το επίπεδο βάσης δεδομένων: Η διαχείριση των λειτουργικών δεδομένων της επιχείρησης, συμπεριλαμβανομένου και των μετά-δεδομένων. Κατά κύριο λόγο γίνεται χρήση RDBMS και SQL.

2.4 Πλεονεκτήματα –μειονεκτήματα

Γενικά είναι λάθος η αντίληψη ότι με την εφαρμογή ενός ERP συστήματος θα βελτιωθούν οι λειτουργίες της επιχείρησης από την μια στιγμή στην άλλη. Οι υψηλές αυτές απαιτήσεις που θέλουν την μείωση την εξόδων και την βελτιστοποίηση των υπηρεσιών εξαρτώνται από το πόσο καλά το επιλεγμένο ERP θα «ταιριάξει» με τις επιχειρησιακές δραστηριότητες και από το πόσο καλά η παραμετροποίηση και το πατρονάρισμα του συστήματος θα συμπνεύσουν με την κουλτούρα, τη στρατηγική, και την δομή του οργανισμού. Συνολικά ένα ERP σύστημα αναμένεται να βελτιώσει τόσο τις εσωτερικές λειτουργίες (μέσα στην επιχείρηση) όσο και τις εξωτερικές, που σχετίζονται με τους πελάτες-προμηθευτές και γενικά την εικόνα της επιχείρησης προς τα έξω. Οι επιχειρήσεις επιλέγουν να εφαρμόσουν ERP συστήματα για πολλά απτά και μη οφέλη και στρατηγικούς λόγους. Σε πολλές περιπτώσεις ο υπολογισμός της απόδοσης της επένδυσης (ROI – return on investment) σε ένα ERP έχει μεγαλύτερη βαρύτητα και έρχεται κόντρα στα πολλά μη απτά και στρατηγικά οφέλη. Τα τελευταία φαίνονται παρακάτω.

Πλεονεκτήματα μπορούν να θεωρηθούν τα παρακάτω :

- Αξιόπιστη πρόσβαση στην πληροφορία. Η ύπαρξη μιας κεντρικής βάσης δεδομένων συνεπάγεται πληροφορία ακριβής και συνεπής και άρα βελτιωμένες αναφορές.
- Αποφυγή επανάληψης πληροφορίας αλλά και λειτουργιών. Τα διάφορα τμήματα του ERP έχουν πρόσβαση σε μια κοινή κεντρική βάση δεδομένων κάτι που σημαίνει αποφυγή πολλαπλών εισόδων δεδομένων.

- Μείωση του χρόνου παράδοσης αναφορών καθώς ελαχιστοποιείται ο χρόνος ανάκτησης των απαραίτητων δεδομένων για τη δημιουργία αυτών.
- Μείωση κόστους μέσα από τη δυνατότητα ευρείας ανάλυσης των επιχειρησιακών αποφάσεων.
- Ευκολία προσαρμογής. Αλλαγές στις επιχειρησιακές διαδικασίες είναι εύκολο να υιοθετηθούν στη δομή του συστήματος.
- Βελτιωμένη ικανότητα προσθήκης νέων λειτουργιών μέσα από τον αρθρωτό τρόπο δόμησης τους και την χρήση add-ons
- Βελτιωμένη δυνατότητα συντήρησης καθώς οι κατασκευαστές των συστημάτων προσφέρουν μακροχρόνια συμβόλαια υποστήριξης.
- E-commerce, E-bussiness

Για να γευτούν τα οφέλη ενός ERP συστήματος, όμως, οι επιχειρήσεις χρειάζεται να ξεπεράσουν συγκεκριμένα προβλήματα και μειονεκτήματα όπως αυτά φαίνονται παρακάτω. Η αγορές ERP συστημάτων υπολογίστηκαν το 1998 γύρω στα 17 δις. δολάρια ακολουθώντας ετήσιο ρυθμό αύξησης από 30% ως 50%. Ακόμα, οι εταιρίες ξοδεύουν πολλαπλάσια του κόστους των αδειών χρήσης σε υπηρεσίες σχετικές με την εφαρμογή και συντήρηση του λογισμικού. Τα έσοδα παγκοσμίως από τέτοιου είδους υπηρεσίες σε ERP συστήματα ήταν 21,5δις. δολάρια το 2000, κάτι που αποτελεί αύξηση της τάξης του 13,1% σε σχέση με την αγορά αξίας 19δις. δολαρίων του 1999 (Broach, 2001). Η συνεχόμενη αύξηση της αγοράς των ERP οφείλετε στο γεγονός ότι οι κατασκευαστές προσθέτουν συνεχώς εφαρμογές στα πακέτα που προσφέρουν.

Μειονεκτήματα μπορούν να θεωρηθούν τα παρακάτω:

- Το κόστος, το οποίο ποικίλει από χιλιάδες ευρώ μέχρι και εκατομμύρια.
- Η συμμόρφωση των διαφόρων τμημάτων του συστήματος με τις επιχειρηματικές διαδικασίες την κουλτούρα και τους στρατηγικούς στόχους της επιχείρησης.
- Η εξάρτηση από τον κατασκευαστή του ERP
- Οι λειτουργίες του ERP και η πολυπλοκότητα. Ένα ERP μπορεί να έχει πάρα πολλές λειτουργίες και τμήματα οπότε ο χρήστης θα πρέπει να επιλέξει προσεκτικά και να υλοποιήσει μόνο τις απαραίτητες σε αυτόν.

- Χρονοβόρα εφαρμογή καθώς χρειάζεται γενική ομοφωνία από τα διάφορα τμήματα της επιχείρησης.

2.5 WebERP

2.5.1 Τι είναι το WebERP

Το WebERP είναι ένα ανοιχτού κώδικα Web-Based ERP σύστημα. Η τρέχουσα έκδοσή του είναι η 3.0.9 και εκδόθηκε στις 27 Ιουνίου 2008. Πρόκειται για ένα λογιστικό/ERP σύστημα που απαιτεί αποκλειστικά έναν φυλλομετρητή διαδικτύου (Web Browser) και έναν PDF Reader για να λειτουργήσει. Οι διαθέσιμες λειτουργίες του συνεχώς αυξάνονται καθώς νέες επιχειρήσεις και προγραμματιστές το υιοθετούν. Η εξέλιξη της χρήσης του WebERP στηρίζεται αποκλειστικά στον «δια στόματος» καλό λόγο μεταξύ των χρηστών καθώς δεν έχει γίνει ποτέ ως τώρα προσπάθεια διαφήμισης ή marketing για την αύξηση της διάθεσης του συστήματος στον κόσμο. Φυσικά δεν υπάρχουν οικονομικοί πόροι ούτε εμπορικά κίνητρα για να γίνει κάτι τέτοιο σε ένα λογισμικό που διανέμεται δωρεάν. Αντ'αυτού η ανάπτυξη της διάθεσής του στηρίζεται αποκλειστικά στη φήμη του και στη στιβαρή λειτουργικότητα του όπως αυτή έχει δοκιμαστεί στην πράξη από έναν αυξανόμενο αριθμό επιχειρήσεων.

2.5.2 Η αξιοπιστία σαν παράγοντας ποιότητας και σχεδιασμού

Ένα λογιστικό σύστημα, περισσότερο από κάθε άλλο σύστημα χρειάζεται να είναι αξιόπιστο. Ο κώδικας του WebERP έχει σε μεγάλο βαθμό αναθεωρηθεί αλλά κυρίως έχει δοκιμαστεί μέσα από την χρήση του προγράμματος σε πραγματικές συνθήκες. Το αποτέλεσμα είναι ένας κώδικας σταθερός και ένα πρόγραμμα που κάνει ακριβώς αυτό που αναμένεται να κάνει σε κάθε περίπτωση. Έμφαση δόθηκε σε χαρακτηριστικά του προγράμματος που οι επιχειρήσεις πραγματικά χρειάζονται. Για να παραχθεί μια στιβαρή εφαρμογή είναι απαραίτητο να βασιστεί σε στιβαρά θεμέλια. Το WebERP έχει αναπτυχθεί σε γλώσσα PHP. Και η PHP είναι ένα ανοιχτού κώδικα περιβάλλον ανάπτυξης εφαρμογών ιδιαίτερα δημοφιλές στην κατασκευή web-based εφαρμογών. Η επιδόσεις και η αξιοπιστία Web-Based εφαρμογών γραμμένων σε PHP είναι αδιαμφισβήτητές. Πέρα από αυτό η PHP μπορεί να παραμετροποιηθεί εύκολα σε οποιοδήποτε λειτουργικό σύστημα και οι απαιτήσεις της σε υπολογιστική ισχύ είναι εξαιρετικά χαμηλές.

2.5.3 Πλεονεκτήματα webERP

Δεν έχει και πολύ καιρό από τότε που οι λογιστικές εφαρμογές εξελίχτηκαν από DOS-based εφαρμογές σε παραθυρικές που εκμεταλλεύονταν τις σύγχρονες γραφικές δυνατότητες των υπολογιστών. Παρόμοια, χρειάστηκε αρκετός χρόνος για τις λογιστικές εφαρμογές για να περάσουν από την X-tier Client/Server τεχνολογία που αντικατέστησε τα παλιά DOS συστήματα και να εκμεταλλευτούν τα πλεονεκτήματα των αληθινά Web-Based εφαρμογών. Αν και τα περισσότερα συστήματα σήμερα δίνουν την επιλογή για Web μορφή του Front-End τμήματος του συστήματος, παραμένουν ουσιαστικά Client/Server εφαρμογές. Λόγω της αυξημένης υπολογιστικής ισχύς που απαιτείται για τέτοιου είδους εφαρμογές οι υποδομές που απαιτούνται από τις επιχειρήσεις μπορεί να περιλαμβάνουν :

- Μια σύνθετη φάρμα από Server για την εξισορρόπηση του φορτίου με εξειδικευμένο λογισμικό διαχείρισης Server.
- Ακριβές άδειες χρήσης λειτουργικών συστημάτων
- Μεγάλης χωρητικότητας συνδέσεις οπτικών ινών
- Προσωπικό εξειδικευμένο και αφοσιωμένο στη διαχείριση τέτοιων πληροφοριακών εφαρμογών αλλά και υποδομών.

Όλη αυτή η απαιτούμενη υποδομή αποτελεί σοβαρό περιοριστικό παράγοντα για πολλές επιχειρήσεις. Το webERP τα αντιπαρέρχεται όλα αυτά με τη χρήση απλής Html για τα γραφικά του συστήματος με μηδαμινό κόστος αλλά και με την ακεραιότητα μια κεντρικής βάσης δεδομένων για την επεξεργασία και αποθήκευση των δεδομένων. Προσφέρει ασφαλή και σε πραγματικό χρόνο πρόσβαση στις εγγραφές της εταιρείας οπουδήποτε και οποτεδήποτε δίνοντας πραγματικά συγκριτικό πλεονέκτημα σε μια εταιρεία. Το μόνο που απαιτείται είναι μια συσκευή με πρόσβαση στο διαδίκτυο.

Επιπλέον, το webERP είναι ευέλικτο ως προς την πλατφόρμα λειτουργίας του και την παραμετροποίηση του. Μπορεί θεωρητικά να παραμετροποιηθεί σε οποιοδήποτε web-server, λειτουργικό σύστημα ή υπολογιστή. Είναι ακόμα εξαιρετικά ευέλικτο γιατί είναι πολύ εύκολο να τροποποιηθεί η λειτουργικότητά του ώστε να

ανταποκρίνεται ακριβώς στις ανάγκες της εκάστοτε επιχείρησης. Οι δυνατότητες πρόσβασης στις λειτουργίες του webERP είναι απεριόριστες χωρίς να επηρεάζουν σε καμία περίπτωση την απόδοσή του : Laptop-pc, Desctop-pc, Pda ή ακόμα και από PC σε Intenet Café.

Το webERP είναι ένα project ανοιχτού κώδικα. Ως τέτοιο δεν απαιτούνται αμοιβές σχετιζόμενες με τη χρήση του ενώ ένας προγραμματιστής με γνώση σε PHP είναι σε θέση να αναγνωρίσει τι είδους επεξεργασία συμβαίνει σε κάθε περίπτωση και να την προσαρμόσει στις επιχειρηματικές ανάγκες. Αυτό είναι κάτι που δεν συμβαίνει σε συστήματα κλειστού κώδικα όπου ο πάροχος υποστήριξης πρέπει να επικοινωνήσει με τους πωλητές του συστήματος.

2.5.4 Χαρακτηριστικά webERP

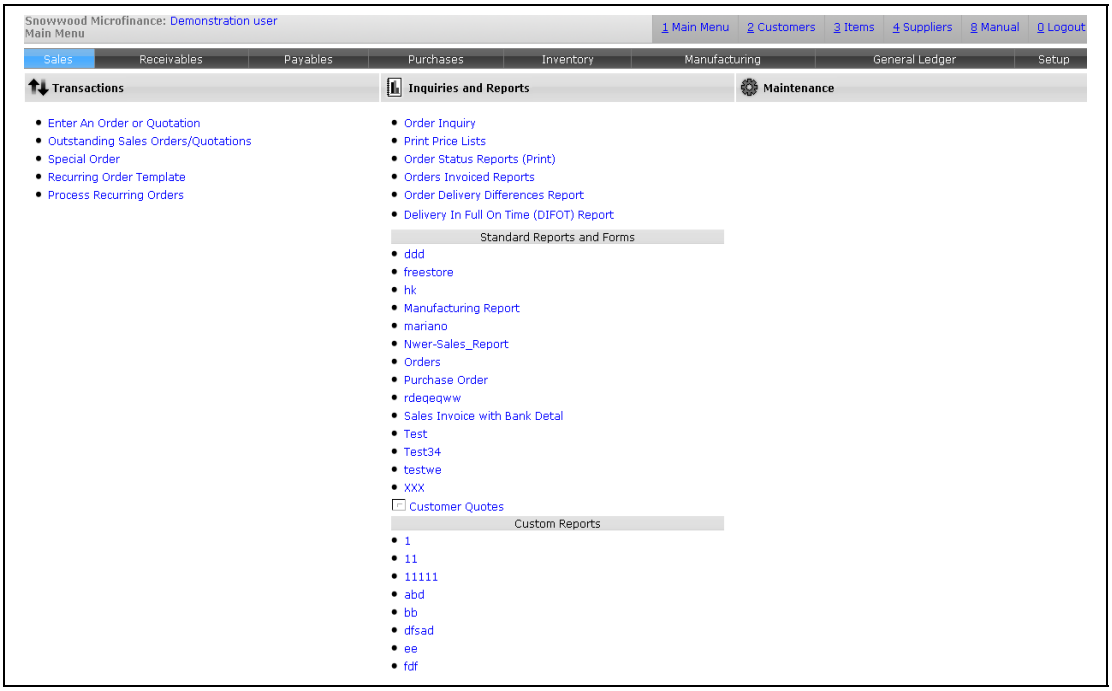
Γενικά χαρακτηριστικά :

- Web-Based
- Τρέχει σε οποιονδήποτε Web-Server που μπορεί να «στεγάσει» και να εξυπηρετήσει PHP αρχεία. Μπορεί δε, να χρησιμοποιηθεί και ένας ISP (Intenet Service Provider) αντί της διατήρησης ιδιόκτητου Server.
- Ελαχιστοποιημένη χρήση Java-Script για μέγιστη συμβατότητα με όλους τους web browsers και συσκευές.
- Παράγει αναφορές σε μορφή PDF για ακριβή στοιχειοθέτηση του κειμένου
- Όλες οι αναφορές και τα script εύκολα μεταβάλλονται για να ικανοποιήσουν τις ανάγκες της επιχείρησης.
- Όλη η επεξεργασία γίνεται στον server και δεν απαιτείται καμιά εγκατάσταση στην πλευρά του Client.
- Υποστήριξη πολλών γλωσσών. Κάθε χρήστης μπορεί να επιλέξει την γλώσσα προτίμησής του.
- Υποστήριξη πολλών θεμάτων αναφορικά με τα γραφικά της διεπαφής. Κάθε χρήστης μπορεί να επιλέξει το θέμα της προτίμησής του.

Ειδικότερα σε ότι αφορά την ασφάλεια :

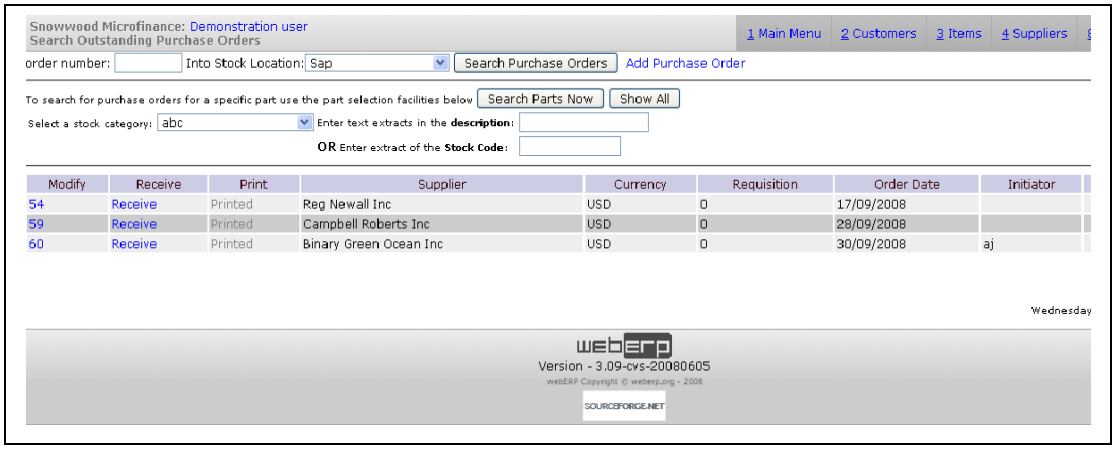
- Οι χρήστες μπορεί να οριστούν με συγκεκριμένη πρόσβαση σε συγκεκριμένες επιλογές.
- Λανθασμένη εισαγωγή κωδικού (πάνω από 3 φορές) έχει σαν αποτέλεσμα το κλείδωμα του λογαριασμού. Το ξεκλείδωμα μπορεί να γίνει μόνο από τον διαχειριστή του συστήματος. Αυτό εμποδίζει τους Crackers να παρακάμψουν την ασφάλεια του συστήματος.
- Οι σελίδες μπορούν να κωδικοποιηθούν με τη χρήση του SSL και το webWRP μπορεί να παραμετροποιηθεί κατά τέτοιο τρόπο που να εμφανίζει τις σελίδες με τη χρήση του SSL εξασφαλίζοντας ότι όλη η πληροφορία που μεταδίδεται μέσω διαδικτύου είναι κωδικοποιημένη.
- Οι χρήστες μπορούν να οριστούν με τέτοιο τρόπο ώστε να έχουν πρόσβαση σε συγκεκριμένες λειτουργίες του κυρίως μενού, ενώ οι επιλογές για τις οποίες δεν έχουν πρόσβαση δεν εμφανίζονται καν στην οθόνη.
- Η ασφάλεια του HTTP μπορεί να παραμετροποιηθεί πάνω από την εσωτερική ασφάλεια του συστήματος παρέχοντας δύο επίπεδα ασφάλειας : το πρώτο είναι η πρόσβαση στον Web-Server και το δεύτερο είναι η πρόσβαση στο ίδιο το webERP.

Ακολουθούν μερικά screen shot του προγράμματος



Κεντρική οθόνη

Οθόνη αναζήτησης είδους



Οθόνη παρουσίασης παραγγελιών

Οθόνη εισαγωγής παραγγελίας

3. ΤΟ osCommerce

3.1 Τι είναι το osCommerce

Το osCommerce είναι μια λύση για Online ηλεκτρονικό κατάστημα που προσφέρει μια ευρεία γκάμα χαρακτηριστικών, που το επιτρέπουν να στηθεί σχετικά γρήγορα και εύκολα. Το πιο σημαντικό από όλα, όμως, είναι ότι είναι δωρεάν ως λύση ανοιχτού κώδικα και διανέμεται υπό την άδεια GNU (General Public License). Η εκκίνηση της εξέλιξης του έγινε τον Μάρτιο του 2000 και από τότε αποτελεί μια δοκιμασμένη λύση που χρησιμοποιείται σε 14060 καταγεγραμμένα ηλεκτρονικά καταστήματα σε όλο τον κόσμο. Σήμερα, το osCommerce έχει πάει στο επόμενο επίπεδο, δίνοντας μια επιλογή που όχι μόνο εξακολουθεί να είναι εύκολη στην εγκατάσταση και στην συντήρηση, αλλά και διευκολύνει τους διαχειριστές των καταστημάτων να παρουσιάζουν τα καταστήματά τους στους πελάτες θέτοντας τις δικές τους μοναδικές απαιτήσεις. Η επιτυχία του osCommerce είναι εξασφαλισμένη από την μεγάλη και ενεργή κοινότητα που το στηρίζει στην οποία τα μέλη βοηθούν το ένα το άλλο και συμμετέχουν σε θέματα ανάπτυξης .

Το osCommerce δεν ευαγγελίζεται ότι είναι η απόλυτη λύση στο ηλεκτρονικό εμπόριο. Αντί να προσπαθεί να προσφέρει όλα τα πράγματα σε όλους τους ανθρώπους, παρέχει ένα βασικό σύνολο λειτουργιών ώστε να ικανοποιεί τις πιο κοινές ανάγκες των περισσοτέρων σχεδόν επιχειρήσεων. Ακολουθώντας αυτή την προσέγγιση το osCommerce απέφυγε το συνηθισμένο λάθος από το οποίο οι περισσότερες εφαρμογές υποφέρουν: υπερβολική πολυπλοκότητα η οποία απαιτεί υπερβολικά σύνθετη υλοποίηση. Αυτό, βέβαια, δεν σημαίνει σε καμία περίπτωση ότι το osCommerce προσφέρει μόνο τα βασικά. Η ικανότητα του να εμπλουτίζεται μέσα από add-ons ή κοινό κώδικα, σημαίνει ότι μπορεί να παραμετροποιηθεί για να ικανοποιήσει τις ανάγκες της κάθε επιχείρησης.

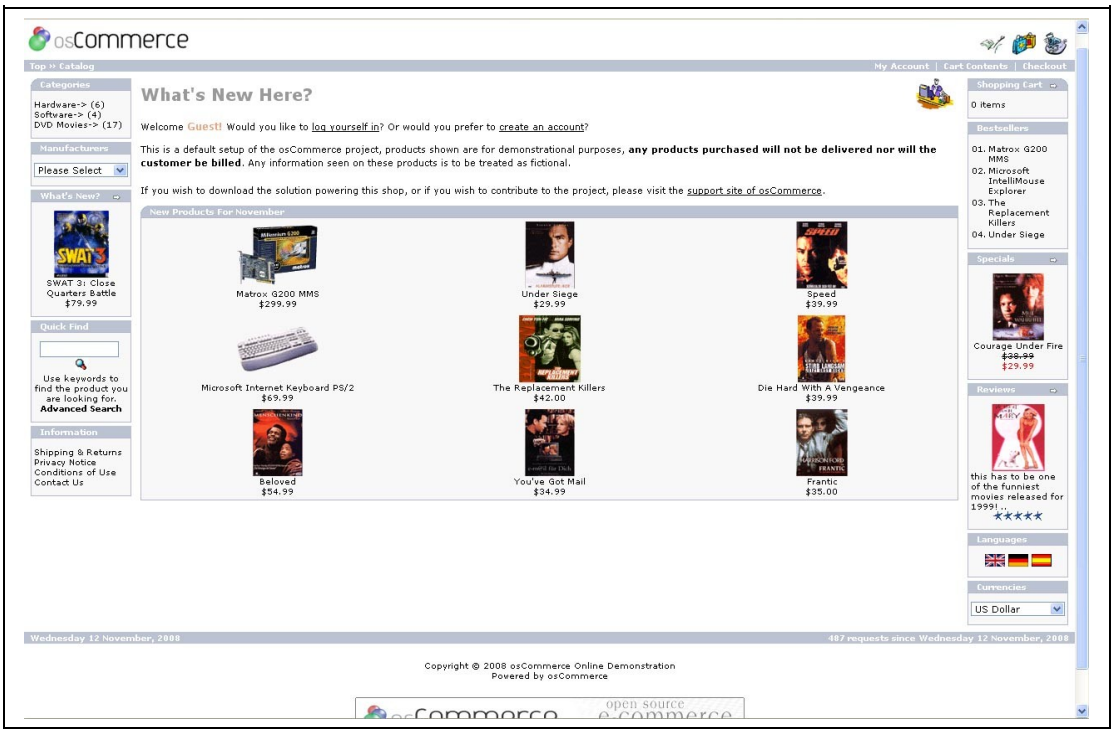
Ακολουθεί μια λίστα χαρακτηριστικών του osCommerce όπως αυτά αναφέρονται στη σελίδα του προγράμματος:

1. Συμβατό με PHP4 και mySQL ενώ μεταφέρεται εύκολα σε έκδοση 5
2. Συμβατό με όλους τους μεγάλους Browser

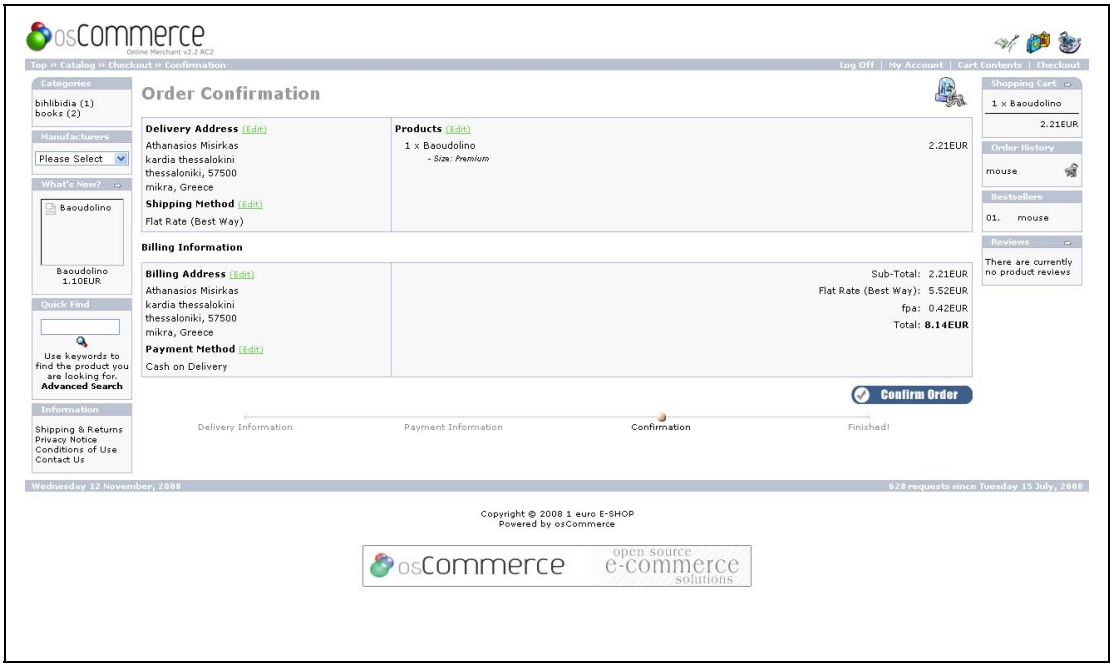
3. Πολύγλωσση υποστήριξη, αγγλικά, γερμανικά, ισπανικά
4. Αυτόματη Web-based εγκατάσταση
5. Απεριόριστος αριθμός προϊόντων και κατηγοριών
6. Αυτόματη εγκατάσταση συστημάτων πληρωμής ηλεκτρονικού εμπορίου: Authorize.net, 2Checkout.com, iPayment.de, PayPal.com, PsiGate.com, SECPay.com και TrustCommerce.com καθώς και πολλά άλλα add-ons συστημάτων πληρωμής.
7. Υποστήριξη συναλλάγματος μέσα από χειροκίνητη ενημέρωση μέσω server ισοτιμιών
8. Υποστήριξη δυναμικών εικόνων
9. Υποστήριξη φυσικών προϊόντων όσο και λογικών προϊόντων που κατεβαίνουν στον υπολογιστή (downloads).
10. Web-based τμήμα του προγράμματος για τον διαχειριστή
11. Αντίγραφα ασφαλείας και επαναφορά της βάσης δεδομένων
12. Προσωρινά και μόνιμα καροτσάκια
13. Ασφαλείς συναλλαγές με το SSL (Secure Sockets Layer)
14. Επιλογές αποστολής βάση κιλών, τιμής, προορισμού
15. Επιλογές αποστολής προϊόντων βασισμένη σε ζώνες καθώς και επιλογή για δωρεάν αποστολή με βάση το σύνολο της παραγγελίας
16. Πλήρης λειτουργικότητα σε ότι αφορά τους φόρους, καθώς υποστηρίζει διαφορετικές ζώνες για διαφορετικά προϊόντα.

Στην υλοποίηση ενός Site βασισμένου στο osCommerce πολύ σημαντική απόφαση είναι ο σχεδιασμός του περιβάλλοντος. Το osCommerce δεν διατίθεται με σύστημα διαθέσιμων Templates αν και γενικά υπάρχουν διαθέσιμα Templates από τρίτους. Αυτό μπορεί να ειδωθεί σαν αρνητικό. Σημαίνει, ωστόσο, ότι ο σχεδιασμός του περιβάλλοντος μπορεί να γίνει είτε με κώδικα είτε με την εγκατάσταση κάποιου Template. Η μεγαλύτερη κριτική που δέχεται το osCommerce, όμως, αφορά τις διάφορες αδυναμίες που παρουσιάζει σε θέματα ασφαλείας όπως είναι το SQL Injection με κώδικα PHP που δεν έχει ελεγχθεί και οι χαλαρές ρυθμίσεις σε θέματα δικαιωμάτων χρήσης αρχείων και καταλόγων που είναι απαραίτητη για τη λειτουργικότητα του προγράμματος.

Ακολουθούν μερικά screen shots του προγράμματος



Κεντρική σελίδα



Σελίδα για την επικύρωση παραγγελίας

4. ΤΑ WEB SERVICES

4.1 Ορισμός για τα web services

Υπάρχουν πολλοί ορισμοί για το τι είναι Web Services, περίπου όσοι και οι εταιρίες πληροφορικής που αναπτύσσουν εργαλεία για τα Web Services.

Το W3C (World Wide Web Consortium) ορίζει την υπηρεσία ιστού (Web Service) σαν ένα σύστημα λογισμικού σχεδιασμένο για αλληλεπίδραση από υπολογιστή σε υπολογιστή πάνω από ένα δίκτυο. Τα Web Services είναι συνήθως απλά Web APIs που μπορούν να προσπελαστούν μέσω ενός δικτύου όπως το Internet ενώ εκτελούνται σε ένα απομακρυσμένο σύστημα που φιλοξενεί την υπηρεσία. Ο ορισμός του W3C για τα web services περιλαμβάνει πολλά διαφορετικά συστήματα αλλά σε κοινή χρήση ο όρος αναφέρεται σε Clients και Servers που επικοινωνούν μέσω του πρωτοκόλλου HTTP το οποίο και χρησιμοποιείται ευρέως στο Internet. Ο ορισμός που δίνει το WebServices.org είναι :

Τα web services είναι συναρτήσεις οι οποίες παρέχονται μέσα από τυποποιημένα πρωτόκολλα και είναι :

0. Encapsulated (Ενθυλακωμένες) : με την έννοια ότι η υλοποίηση της συνάρτησης δεν φαίνεται από έξω.
1. Loosely coupled (χαλαρής συνδεσιμότητας): με την έννοια ότι αλλάζοντας την υλοποίηση μιας συνάρτησης δεν σημαίνει ότι απαιτείται και αλλαγή της καλούσας συνάρτησης.
2. Contracted : με την έννοια ότι υπάρχουν δημόσια διαθέσιμες περιγραφές με τον τρόπο λειτουργίας της συνάρτησης, πώς να γίνει η κλήση της, τι δεδομένα εισόδου απαιτεί καθώς και τι δεδομένα εξόδου παράγει.

Ένας πολύ καλός ορισμός έρχεται από την IBM : *Τα Web Services είναι μια τεχνολογία που επιτρέπει στις εφαρμογές να επικοινωνούν μεταξύ τους ανεξαρτήτως πλατφόρμας και γλώσσας προγραμματισμού. Ένα Web Service είναι μια διεπαφή λογισμικού (Software Interface) που περιγράφει μια συλλογή από λειτουργίες οι οποίες μπορούν να προσεγγιστούν από το δίκτυο μέσω πρότυπων*

μηνυμάτων XML. Χρησιμοποιεί πρότυπα βασισμένα στη γλώσσα XML για να περιγράψει μία λειτουργία (Operation) προς εκτέλεση και τα δεδομένα προς ανταλλαγή με κάποια άλλη εφαρμογή. Μια ομάδα από Web Services οι οποίες αλληλεπιδρούν μεταξύ τους καθορίζει μια εφαρμογή Web Services.

Η Microsoft μέσα από το MSDN της καταλήγει ότι όλα τα web services έχουν τρία (3) κοινά χαρακτηριστικά :

- *Τα Web Services εκθέτουν χρήσιμη λειτουργικότητα σε χρήστες του διαδικτύου μέσα από ένα πρότυπο δικτυακό πρωτόκολλο. Στις περισσότερες περιπτώσεις αυτό το πρωτόκολλο είναι το SOAP (Simple Object Access Protocol).*
- *Τα 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).*
- *Τα Web Services καταχωρούνται ώστε οι δυνητικοί χρήστες να μπορούν να τα βρουν εύκολα. Αυτό γίνεται με το UDDI (Universal Discovery Description and Integration).*

Τα Web Services λοιπόν αποτελούν μία αρχιτεκτονική κατανεμημένων συστημάτων κατασκευασμένη από πολλά διαφορετικά υπολογιστικά συστήματα τα οποία επικοινωνούν μέσω του δικτύου ώστε να δημιουργήσουν ένα σύστημα. Αποτελούνται από ένα σύνολο από πρότυπα τα οποία επιτρέπουν στους προγραμματιστές - Developers να υλοποιήσουν κατανεμημένες εφαρμογές χρησιμοποιώντας διαφορετικά εργαλεία από διαφορετικούς προμηθευτές. Αυτά τα προγράμματα χρησιμοποιούν ένα συνδυασμό από ενότητες λογισμικού (Software Modules) οι οποίες καλούνται από συστήματα που ανήκουν σε διαφορετικά τμήματα ενός οργανισμού ή σε διαφορετικούς οργανισμούς.

4.2 Το παρελθόν των web services

Με την ανάπτυξη των δικτύων υπολογιστών γεννήθηκε και η θεωρία γύρω από τα κατανεμημένα συστήματα. Οι εφαρμογές χωρίστηκαν αρχικά σε δύο κατηγορίες: τους Clients, οι οποίοι και ξεκινούσαν την κατανεμημένη δραστηριότητα, και τους

Server οι οποίοι εκτελούσαν και την δραστηριότητα. Αυτού του είδους η αποκεντροποίηση ελαχιστοποιούσε τις συμφορήσεις κατανέμοντας το φόρτο εργασίας. Προσέφερε ευελιξία στον σχεδιασμό των εφαρμογών η οποία ήταν άγνωστη μέχρι τότε. Αλλά αυτή η αρχιτεκτονική 2-tier είχε και περιορισμούς. Για λόγους ευελιξίας και αντιμετώπισης λαθών, προστέθηκε και ένα τρίτο Tier χωρίζοντας τις εφαρμογές στο κομμάτι της παρουσίασης-διεπαφής, στο Tier που περιείχε το Bussiness Logic, και σε αυτό που είχε να κάνει με τα δεδομένα. Αυτό το μοντέλο τριών επιπέδων έγινε ο πιο δημοφιλής τρόπος κατανεμημένων εφαρμογών.

Το θεμέλιο για την επικοινωνία μεταξύ κατανεμημένων τμημάτων μιας εφαρμογής είναι το RPC – Remote Procedure Call. Προκειμένου να μην ασχολούνται οι προγραμματιστές με καθήκοντα χαμηλού επιπέδου, όπως ο μετασχηματισμός των δεδομένων και η σειρά των bytes των διαφόρων μηχανών ένα νέο στρώμα λογισμικού εισήχθηκε στην αγορά. Αυτό το ενδιάμεσο είχε σκοπό να απαλείφει τις διαφορές μεταξύ των Host. «Καθότανε» πάνω από το λειτουργικό σύστημα του Host και τις υπηρεσίες δικτύου και προσέφερε τις υπηρεσίες του στις εφαρμογές από κάτω του. Τα πρώτα τέτοια ενδιάμεσα, όπως, το DCE βασίζονταν στον διαδικασιακό προγραμματισμό και υπερκεράστηκαν με την εισαγωγή του αντικειμενοστραφούς προγραμματιστικού μοντέλου με ενδιάμεσα όπως τα CORBA, DCOM, RMI τα οποία είναι και τα πιο δημοφιλή ως και σήμερα.

4.3 Το σήμερα

Όπως προαναφέρθηκε υπάρχουν τρία ενδιάμεσα διαθέσιμα: DCOM, CORBA, RMI. Το καθένα έχει τα πλεονεκτήματα και τα μειονεκτήματά του.

- **CORBA** : Είναι ένα ανοιχτό πρότυπο για κατανεμημένα συστήματα. Το πλεονέκτημα του CORBA είναι ότι τόσο οι Clients όσο και οι Servers μπορούν να συνταχθούν σε οποιαδήποτε προγραμματιστική γλώσσα. Αυτό είναι δυνατό γιατί τα αντικείμενα ορίζονται σε υψηλό εννοιολογικό επίπεδο το οποίο εξασφαλίζεται με το IDL (Interface Definition Language). Αφού οριστεί ένα αρχείο IDL ο compiler φροντίζει την αντιστοίχισή του με μια συγκεκριμένη γλώσσα προγραμματισμού. Για κατανεμημένες εφαρμογές υψηλής απόδοσης το CORBA είναι το

καλύτερο ενδιάμεσο. Η ανάπτυξη, όμως, μιας κατανεμημένης εφαρμογής με CORBA είναι μια αρκετά περίπλοκη διαδικασία. Προκειμένου το IDL να μπορεί να μεταφράζεται σε διάφορες προγραμματιστικές γλώσσες, χρειάζεται να περιοριστεί σε γενικές ιδέες που μπορούν να υλοποιηθούν σε όλες τις υποστηριζόμενες γλώσσες προγραμματισμού.

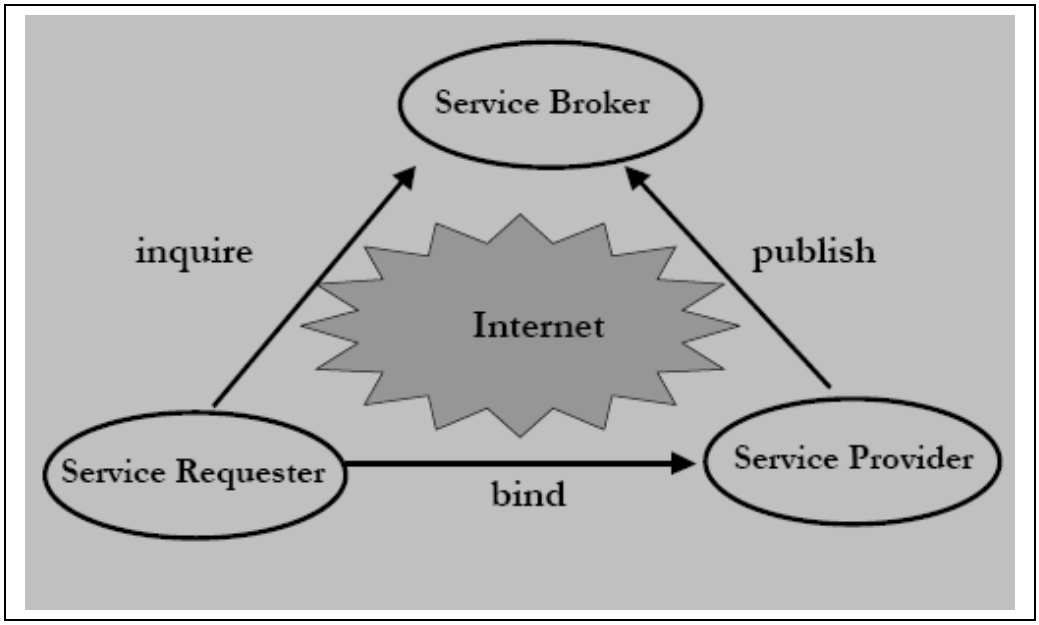
- **RMI** : Remote Method Iinvocation, επιτρέπει τη δημιουργία κατανεμημένων από java σε java εφαρμογές, στις οποίες οι μέθοδοι των απομακρυσμένων Java αντικειμένων γίνεται από άλλες Java Virtual Machines, που ενδεχομένως βρίσκονται σε άλλους hosts. Ένα πρόγραμμα Java μπορεί να κάνει μια κλήση σε ένα απομακρυσμένο αντικείμενο, μόλις αποκτήσει μια αναφορά στο απομακρυσμένο αντικείμενο. Ένας Client μπορεί να καλέσει ένα απομακρυσμένο αντικείμενο σε έναν Server. Ο Server μπορεί με τη σειρά του να είναι Client για άλλα απομακρυσμένα αντικείμενα. Το πρωτόκολλο που χρησιμοποιείται για την επικοινωνία είναι το Java Remote Method Protocol (JRMP). Ο προγραμματισμός με το RMI είναι άμεσος αν ο προγραμματιστής αποκτήσει μια εμπειρία με τη γλώσσα Java και τις κατανεμημένες εφαρμογές. Δεν απαιτείται κάποια αφαιρετική γλώσσα όπως η IDL για την περιγραφή των απομακρυσμένων αντικειμένων. Από την άλλη, μπορεί να χρησιμοποιηθεί μόνο για επικοινωνία από Java σε Java. Και η ευκολία στη χρήση επιτυγχάνεται κρατώντας το ενδιάμεσο αρκετά απλό.
- **DCOM** : Το Distributed Component Object Model της Microsoft επιτρέπει κλήσεις σε απομακρυσμένα αντικείμενα. Ένας DCOM server δημοσιεύει τις μεθόδους του σε Clients μέσα από πολλές διεπαφές. Αυτές είναι γραμμένες σε IDL που είναι παρόμοια με τη C++. Ο IDL compiler του DCOM λειτουργεί παρόμοια με τον compiler του CORBA. Επιπλέον, δημιουργούνται βιβλιοθήκες τύπων. Αυτές είναι αρχεία που περιγράφουν το απομακρυσμένο αντικείμενο και τα οποία μπορούν να ερωτηθούν από διεπαφές που παρέχει ο μηχανισμός COM. Το πρωτόκολλο επικοινωνίας που χρησιμοποιείται εδώ είναι το Object Remote Procedure Call (ORPC). Το DCOM επιτρέπει τη χρήση διαφόρων γλωσσών για τον προγραμματισμό των αντικειμένων του server.

Και οι τρεις μεθοδολογίες που περιγράφηκαν λειτουργούν με παρόμοιο τρόπο. Οι διαφορές βρίσκονται στις επιμέρους λεπτομέρειες που κάθε ενδιάμεσο παρέχει καθώς και στο βαθμό πολυπλοκότητας που έχει το καθένα για να υλοποιηθεί. Αυτό, όμως, που ισχύει και για τα τρία είναι το λεγόμενο Tight Coupling ή στα ελληνικά ισχυρή συνδεσιμότητα μεταξύ Client και Server. Εξαιτίας των διαφορετικών πρωτοκόλλων που χρησιμοποιούνται σε κάθε περίπτωση δεν είναι δυνατή η κλήση ενός έστω DCOM Server από έναν RMI Client. Επιπλέον, αυτά τα ενδιάμεσα που συνήθως χρησιμοποιούνται για εφαρμογές Intranet εγείρουν ένα σημαντικό θέμα, αυτό της προσπέλασης του Firewall καθώς χρησιμοποιούν μη πρότυπες πόρτες για την επικοινωνία τους. Για το λόγο αυτό και τα τρία παρέχουν δυνατότητα HTTP Tunneling προκειμένου να είναι ο Server ασφαλής πίσω από το Firewall.

4.4 Πίσω από τα web services

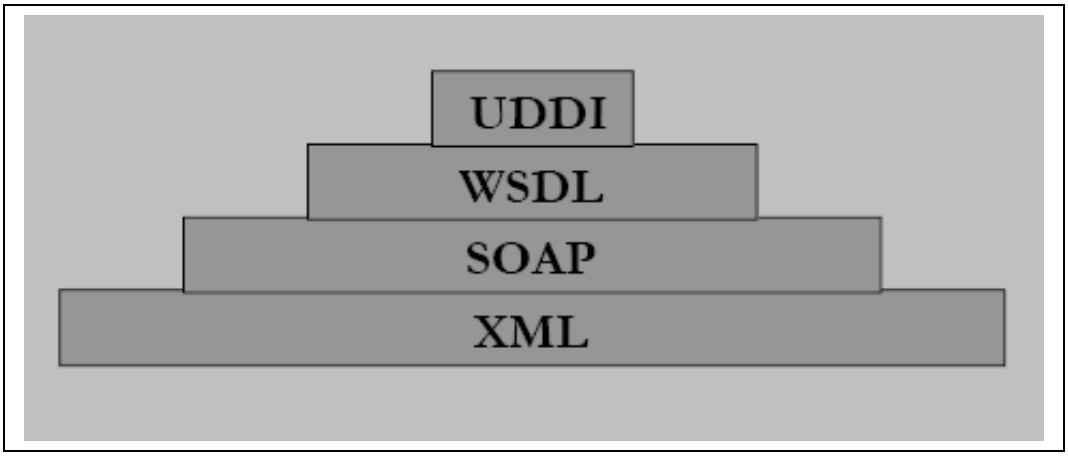
Στην παραδοσιακή περίπτωση Client/Server, υπήρχε ο Server ο οποίος παρείχε κάποια λειτουργία που μπορούσε να κληθεί ή να χρησιμοποιηθεί από τον Client. Κάποιο είδος υπηρεσίας αναζήτησης λειτουργούσε σαν «μεσίτης»(Broker) ανάμεσα στον Client και στον Server. Μιας και τα Web Services εντάσσονται εννοιολογικά στις κατανεμημένες εφαρμογές αποτελούνται από τα ίδια τρία συστατικά (Σχήμα 4):

- Μια υπηρεσία «μεσίτη» (Broker) που θα έχει τη λειτουργία αναζήτησης υπηρεσιών ανάμεσα στον πάροχο (Provider) της υπηρεσίας και στον αιτούντα(Requester) της υπηρεσίας.
- Τον πάροχο της υπηρεσίας που δημοσιεύει τις υπηρεσίες του στον μεσίτη
- Τον αιτούντα της υπηρεσίας (Service Requestor) που ζητά από τον μεσίτη που μπορεί να βρει τον κατάλληλο πάροχο (Service Provider) και δεσμεύει τον εαυτό του με τον πάροχο.



Σχήμα 4

Για να επιτευχθεί το παραπάνω πρότυπο επικοινωνίας επιστρατεύονται μια σειρά από τεχνολογίες. Στη βάση όλων βρίσκεται η XML (Extensible Markup Language) η οποίο είναι μια ευρέως αποδεκτή μορφή ανταλλαγής πληροφοριών και της αντίστοιχης σημασιολογίας τους. Αποτελεί το ακρογωνιαίο λίθο για κάθε άλλη τεχνολογία που χρησιμοποιείται από τα Web Services. Στη συνέχεια έχουμε το πρωτόκολλο SOAP (Simple Object Access Protocol) το οποίο επιτρέπει τη μεταφορά δεδομένων μεταξύ ετερογενών συστημάτων. Ο ορισμός-περιγραφή του Web Service γίνεται με τη γλώσσα WSDL (Web Services Definition Language). Η γλώσσα αυτή ουσιαστικά περιγράφει την υπηρεσία χρησιμοποιώντας γραμματική της XML. Στην κορυφή έχουμε το UDDI (Universal Description, Discovery and Intergration) το οποίο αποτελεί ένα πρότυπο, σχεδιασμένο για να παρέχει έναν κατάλογο με τις επιχειρήσεις και τα Web Services που διαθέτουν. Κατ' αυτόν τον τρόπο καταλήγουμε στην «πυραμίδα» των Web Services η οποία παρουσιάζεται στο σχήμα 5.



Σχήμα 5

Στην συνέχεια εξετάζεται επιμέρους η κάθε μία από τις παραπάνω τεχνολογίες.

4.4.1 XML (Extensible Markup Language)

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

Οι ρίζες της XML μπορούν ν' αναζητηθούν στην εκρηκτική ανάπτυξη του Παγκόσμιου Ιστού στα μέσα της δεκαετίας του 1990 και στους πολέμους των Browser που έλαβαν χώρα μεταξύ της Microsoft Corporation και της Netscape Corporation, όπου καθεμιά από αυτές πάλευε για την απόλυτη κυριαρχία.

Καθώς το Web γινόταν όλο και πιο μεγάλο και όλο και περισσότεροι χρήστες το χρησιμοποιούσαν, άρχισαν ν' ανακαλύπτονται από τους προγραμματιστές, που χρησιμοποιούσαν HTML διάφορα προβλήματα:

Το γεγονός ότι ο ίδιος πόρος HTML εμφανιζόταν με διάφορες μορφές, ανάλογα με τον Browser που χρησιμοποιούνταν σήμαινε ότι, οι σχεδιαστές των ιστοσελίδων

έπρεπε το λιγότερο να διπλασιάσουν τις προσπάθειές τους. Ορισμένοι κατασκευαστές Browser ανέπτυξαν εργαλεία HTML που δεν ήταν αναγνωρίσιμα από άλλους Browser.

Ήταν σχεδόν αδύνατον να διακριθεί οποιαδήποτε σημαντική έννοια μέσα σε μια ιστοσελίδα: πουθενά αλλού αυτό δεν ήταν πιο προφανές από ότι στην χρήση των μηχανών αναζήτησης. Ακόμη και στα μέσα της δεκαετίας του 1990 πολλοί χρήστες εξέφραζαν την απογοήτευσή τους σε ότι αφορούσε αυτά τα προγράμματα εξαιτίας του όγκου των ανακτημένων αρχείων που επέστρεφαν οι μηχανές, τα οποία, στην καλύτερη περίπτωση, σχετίζονταν οριακά με την αναζήτηση. Η αιτία αυτής της "φτωχής" απόδοσης δεν ήταν η τεχνολογία των μηχανών αναζήτησης - στην πραγματικότητα επρόκειτο για εξεζητημένα προγράμματα που αποτελούν αποδεικτικό της εξυπνάδας των προγραμματιστών - αλλά το ότι τα δεδομένα που επεξεργάζονταν, η σελίδες HTML, δεν έδιναν πολλά στοιχεία για το περιεχόμενό τους.

Εξαιτίας αυτών των προβλημάτων η Κοινοπραξία Παγκόσμιου Ιστού, η ομάδα που ελέγχει την διαδικασία τυποποίησης του Ιστού, αποφάσισε το 1996 ν' αναπτύξει μία σημειακή γλώσσα που μελλοντικά θα υποσκέλιζε την HTML. Οι στόχοι αυτής της γλώσσας ήταν:

- Να χρησιμοποιείται εύκολα στο Internet.
- Να μπορεί να υποστηρίζει πολλές εφαρμογές οι οποίες θα κυμαίνονται από Browser μέχρι βάσεις δεδομένων μηχανών αναζήτησης.
- Να είναι συμβατή με την SGML, την γλώσσα επεξεργασίας κειμένου που αποτέλεσε την έμπνευση για την HTML.
- Να μην αποτελεί πολύπλοκη διαδικασία η ανάπτυξη επεξεργαστών κειμένων γραμμένων σε γλώσσες που θα βασίζονταν σε XML. Για παράδειγμα θα έπρεπε να είναι εύκολη η εγγραφή ενός προγράμματος για τον έλεγχο της σαφήνειας ενός κειμένου.
- Ο αριθμός των προαιρετικών εργαλείων της γλώσσας να είναι χαμηλός.
- Να είναι εύκολη η ανάγνωση και κατανόηση των αρχείων XML.
- Να είναι εύκολο ν' αναπτυχθούν, με την χρήση απλών συντακτών, αρχεία γραμμένων σε γλώσσα βασιζόμενη σε XML.

Το 1998, η γλώσσα XML παρουσιάστηκε παγκόσμια ως μία τελευταία υπόδειξη της Κοινοπραξίας Παγκόσμιου Ιστού. Σαν αποτέλεσμα, έγινε μία σταθερά για το

Internet. Βασιζόταν στην γλώσσα επεξεργασίας κειμένων SGML, η οποία αποτέλεσε την έμπνευση για την HTML.

Στην πράξη η XML όπως και η HTML είναι μια μορφή κειμένου που βασίζεται σε ζευγάρια ετικετών. Παραδείγματος χάρη :

```
<Authors>
  <Author>Luke Welling</Author>
  <Author>Laura Thomson</Author>
</Authors>
```

Η κύρια διαφορά ανάμεσα στην XML και στην HTML είναι ότι στην XML μπορεί ο καθένας να δημιουργήσει τις δικές του ετικέτες. Κατ' αυτόν τον τρόπο μπορεί να δομήσει τα έγγραφά του ώστε να ταιριάζουν απόλυτα στο περιεχόμενο της πληροφορίας που θέλει να αποθηκεύσει.

4.4.2 SOAP (Simple Object Access Protocol)

Το Soap είναι ένα ελαφρύ πρωτόκολλο για ανταλλαγή πληροφορίας σε αποκεντρωμένα και κατανεμημένα συστήματα. Στηρίζεται στη γλώσσα XML και αποτελείται κυρίως από τρία μέρη.

- **Τον φάκελο (Envelope)** που καθορίζει ένα πλαίσιο το οποίο περιγράφει τι περιέχει το μήνυμα και πώς να επεξεργαστεί.
- **Ένα σύνολο από κανόνες κωδικοποίησης** για την έκφραση στιγμιότυπων των τύπων δεδομένων που ορίζονται από την εφαρμογή.
- **Μια σύμβαση** για την αναπαράσταση κλήσεων και απαντήσεων απομακρυσμένων διαδικασιών.

Ο φάκελος (Envelope) : Ένα Soap μήνυμα είναι ουσιαστικά ένα έγγραφο XML που αποτελείται υποχρεωτικά από το Soap Envelope, προαιρετικά από μια επικεφαλίδα (Soap Header) και υποχρεωτικά από το κυρίως σώμα (Soap Body). Ο φάκελος είναι κύριο στοιχείο του XML εγγράφου και αναπαριστά το μήνυμα. Η επικεφαλίδα (Header) είναι ένα γενικός μηχανισμός που προσθέτει στοιχεία σε ένα

μήνυμα SOAP με τρόπο αποκεντρωμένο και χωρίς προηγούμενη συμφωνία μεταξύ των μερών που επικοινωνούν. Το σώμα είναι το στοιχείο που περιέχει την πληροφορία που αποσκοπεί στον τελικό παραλήπτη του μηνύματος.

Κανόνες κωδικοποίησης : Ο τρόπος κωδικοποίησης στο SOAP βασίζεται σε ένα απλό σύστημα τύπων που αποτελεί γενίκευση των πιο κοινών στοιχείων που βρίσκονται στα συστήματα τύπων των προγραμματιστικών γλωσσών, βάσεων δεδομένων και ημιδομημένων πληροφοριών. Σε αυτό το κομμάτι ορίζονται οι κανόνες για τον τρόπο που θα κωδικοποιηθεί-αποκωδικοποιηθεί η πληροφορία σύμφωνα με την γραμματική της XML.

Κλήσεις απομακρυσμένων διαδικασιών : Ένας από τους σκοπούς της δημιουργίας του SOAP ήταν η ενθυλάκωση και η ανταλλαγή RPC κλήσεων χρησιμοποιώντας την ευελιξία και τη δυνατότητα επέκτασης που παρέχει η XML. Έδω καθορίζεται ο τρόπος λειτουργίας των κλήσεων και απαντήσεων των απομακρυσμένων διαδικασιών. Για να γίνει μια κλήση σε μια μέθοδο οι ακόλουθες πληροφορίες απαιτούνται.

- Το URI του επιθυμητού αντικειμένου.
- Το όνομα της μεθόδου.
- Μια προαιρετική μέθοδο υπογραφής
- Τις παραμέτρους της μεθόδου.
- Μια προαιρετική κεφαλίδα για τα δεδομένα.

4.4.3 WSDL (Web Service Description Language)

Σύμφωνα με το w3.org ο ορισμός για την WSDL είναι ο εξής:

"Η WSDL είναι ένα σχήμα XML για την περιγραφή δικτυακών υπηρεσιών σαν ένα σύνολο από τελικά σημεία που λειτουργούν σε μηνύματα τα οποία περιέχουν πληροφορία είτε προσανατολισμένη στα έγγραφα είτε πρσανατολισμένη στις διαδικασίες.

Οι λειτουργίες και τα μηνύματα περιγράφονται περιληπτικά, και τότε δένονται σε ένα συγκεκριμένο πρωτόκολλο δικτύων και μορφή μηνυμάτων για να καθορίσουν ένα τελικό σημείο. Πολλά σχετικά τελικά σημεία συνδυάζονται σε υπηρεσίες (services). Η WSDL είναι επεκτάσιμη στο να επιτρέπει την περιγραφή τελικών σημείων και των μηνυμάτων τους άσχετα από τη μορφή των μηνυμάτων και των πρωτοκόλλων δικτύων που χρησιμοποιούνται για την επικοινωνία. Παρόλα αυτά, αυτή τη στιγμή στην προδιαγραφή της WSDL οι μόνες συνδέσεις που περιλαμβάνονται περιγράφουν πώς μπορούμε να χρησιμοποιήσουμε την WSDL σε συνδυασμό με το SOAP 1.1, το HTTP GET/POST και το MIME."

Με πιο απλά λόγια η WSDL μας βοηθάει να περιγράψουμε ένα σύνολο από μηνύματα και το πώς αυτά τα μηνύματα ανταλλάσσονται.
Η σήμανση που χρησιμοποιείται σε ένα αρχείο WSDL για να περιγράψει μορφές μηνυμάτων βασίζεται στο πρότυπο του XML Schema το οποίο σημαίνει ότι είναι ταυτόχρονα ανεξάρτητη από γλώσσα προγραμματισμού και βασισμένη σε πρότυπα. Αυτό το γεγονός την κάνει κατάλληλη για να περιγράψει διεπαφές Web Services οι οποίες είναι προσβάσιμες από μία μεγάλη ποικιλία πλατφόρμων και γλωσσών προγραμματισμού. Επιπλέον, εκτός του ότι περιγράφει τα περιεχόμενα των μηνυμάτων, η WSDL ορίζει πού είναι διαθέσιμη μία υπηρεσία και ποια πρωτόκολλα επικοινωνίας χρησιμοποιούνται για να επικοινωνήσουμε με αυτή την υπηρεσία. Αυτό σημαίνει ότι ένα αρχείο WSDL ορίζει όλα όσα χρειάζονται για να γράψουμε ένα πρόγραμμα το οποίο να λειτουργεί με ένα Web Service

Ένα έγγραφο WSDL χρησιμοποιεί τα παρακάτω στοιχεία για τον ορισμό δικτυακών υπηρεσιών :

- **Types** - ένα περίβλημα για ορισμούς τύπων δεδομένων χρησιμοποιώντας ένα σύστημα τύπων (όπως για παράδειγμα το XML Schema).
- **Message** - ένας περιγραφικός ορισμός των δεδομένων που ανταλλάσσονται.
- **Operation** - μία περιγραφή μίας λειτουργίας που υποστηρίζεται από μία υπηρεσία
- **Port Type** - ένα περιγραφικό σύνολο από λειτουργίες που υποστηρίζονται από ένα ή περισσότερα τελικά σημεία.

- **Binding** - ένα συγκεκριμένο πρωτόκολλο και μορφή δεδομένων για ένα συγκεκριμένο τύπο τελικών σημείων (Port Type).
- **Port** - ένα μοναδικό τελικό σημείο που ορίζεται σαν συνδυασμός μίας σύνδεσης (Binding) και μιας διεύθυνσης δικτύου.
- **Service** - μία συλλογή από σχετικά τελικά σημεία.

4.4.4 UDDI (Universal Discovery Description and Integration)

Ο σκοπός που δίνει το OASIS για το Universal Description, Discovery and Integration είναι ο παρακάτω :

"*Τα Web Services έχουν νόημα μόνο όταν δυνητικοί χρήστες μπορούν να βρουν πληροφορίες ικανές ώστε να επιτρέψουν την εκτέλεσή τους. Το Universal Description Discovery & Integration (UDDI) εστιάζει στον καθορισμό ενός συνόλου από υπηρεσίες που θα υποστηρίζουν την περιγραφή και την ανακάλυψη:*

1. *Των εταιριών, των οργανισμών και άλλων παρόχων Web Services*
2. *Των Web Services, που είναι διαθέσιμες*
3. *Και των τεχνικών διεπαφών οι οποίες μπορούν να χρησιμοποιηθούν ώστε να έχει κάποιος χρήστης πρόσβαση σε αυτές τις υπηρεσίες.*

Βασισμένο σε ένα κοινό σύνολο από βιομηχανικά πρότυπα, συμπεριλαμβανομένων των HTTP, XML, XML Schema και SOAP το UDDI παρέχει μία διαλειτουργική, θεμελιώδη υποδομή για ένα περιβάλλον λογισμικού προσανατολισμένο στις υπηρεσίες τόσο για δημόσια διαθέσιμες υπηρεσίες όσο και για υπηρεσίες που εκτίθενται μόνο εσωτερικά ενός οργανισμού."

Με πιο απλά λόγια το UDDI είναι ο «χρυσός οδηγός» των Web Services. Όπως σε ένα χρυσό οδηγό, μπορούμε να αναζητήσουμε μια εταιρία που προσφέρει τις υπηρεσίες που χρειαζόμαστε, να διαβάσουμε για μια προσφερόμενη υπηρεσία και να επικοινωνήσουμε με κάποιον για περισσότερες λεπτομέρειες. Φυσικά μπορούμε να προσφέρουμε ένα Web Service χωρίς να το καταχωρίσουμε στο UDDI, όπως αν ανοίγαμε μία επιχείρηση στο υπόγειο του σπιτιού μας και βασιζόμασταν στη

διαφήμιση από «στόμα σε στόμα». Αλλά αν θέλουμε να αγγίξουμε το ευρύ κοινό, θα χρειαστούμε το UDDI ώστε οι δυνητικοί μας πελάτες να μπορέσουν να μας βρουν.

Η οργάνωση του UDDI

- **Λευκές σελίδες**
  - Όπως και σε κάθε τηλεφωνικό κατάλογο οι λευκές σελίδες παρέχουν ένα ονομαστικό ευρετήριο έτσι και εδώ οι λευκές σελίδες παρέχουν ένα ονομαστικό ευρετήριο των επιχειρήσεων. Περιέχουν πληροφορίες όπως το όνομα της επιχείρησης, κάποια πληροφορία συνδέσμου, διεύθυνση και τηλέφωνο. Επίσης μπορεί να περιέχουν και αναγνωριστικά των επιχειρήσεων.
- **Κίτρινες σελίδες**
  - Αυτές οι σελίδες χωρίζουν τις επιχειρήσεις ή τις υπηρεσίες ανάλογα με την κατηγορία στην οποία ανήκουν. Για παράδειγμα μπορεί να περιέχονται δεδομένα όπως βιομηχανίες, προϊόντα ή γεωγραφικούς κώδικες, όλα ομαδοποιημένα σύμφωνα με καθορισμένες κατηγορίες.
- **Πράσινες σελίδες**
  - Αυτές οι σελίδες περιλαμβάνουν την τεχνική περιγραφή μιας Web Service. Αυτό συνιστά έναν δείκτη σε μια περιγραφή της Web Service και μια διεύθυνση για να την καλέσεις.

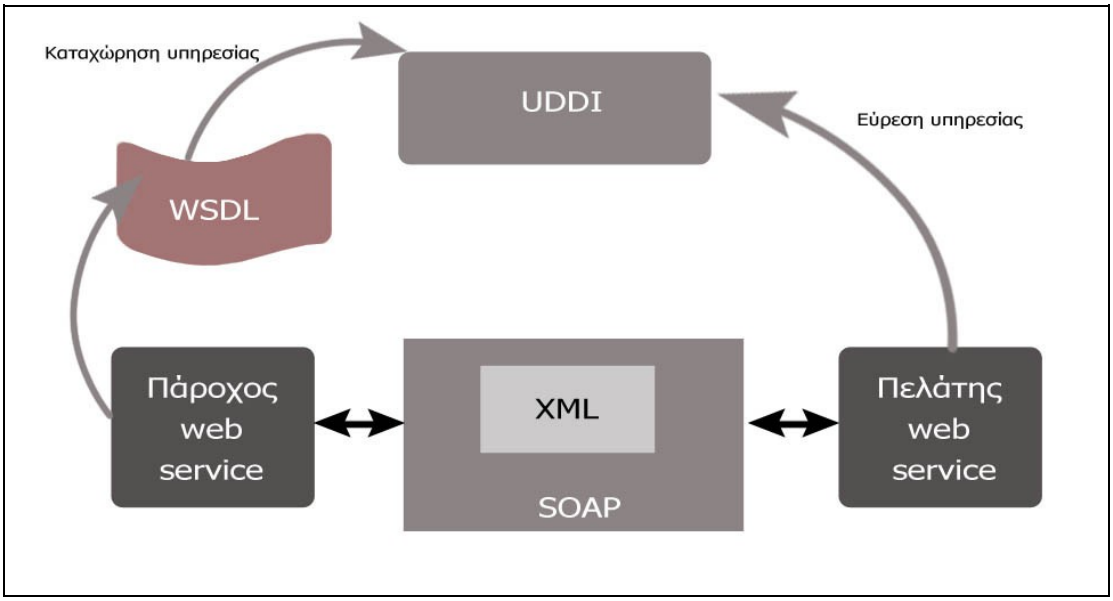
4.4.5 Συνδυάζοντας όλα τα παραπάνω

Συνοψίζοντας τα παραπάνω για τα web services καταλήγουμε στα εξής:

- Το SOAP (Simple Object Access Protocol) είναι η μέθοδος με την οποία μπορούμε να στέλνουμε μηνύματα σε διαφορετικές ενότητες λογισμικού (software modules)..
- Το UDDI (Universal Description, Discovery and Integration) είναι η καθολική βάση αναζήτησης για τον εντοπισμό των υπηρεσιών. Είναι ανάλογη της υπηρεσίας ευρετηρίου της μηχανής αναζήτησης, στην οποία όλες οι ιστοσελίδες καταχωρούνται και σχετίζονται με λέξεις κλειδιά.

- Η WSDL (Web Services Definition Language) είναι ο τρόπος με τον οποίο διαφορετικές υπηρεσίες περιγράφονται στο UDDI
- Όλα τα παραπάνω χρησιμοποιούν την XML σαν γλώσσα σύνταξης.

Στο σχήμα 6 φαίνεται σχηματικά ο τρόπος λειτουργίας των web services



Σχήμα 6

4.4.6 Πλεονεκτήματα των Web Services

Οι υπηρεσίες δικτύου παρουσιάζουν πολλά πλεονεκτήματα

- Είναι ανεξάρτητα πλατφόρμας όπως αυτό προκύπτει από την χρήση προτύπων XML. Αυτό επιτρέπει σε έναν οργανισμό να σχεδιάσει και να εφαρμόσει το Web Service σε πλατφόρμα Server της δικής του επιλογής. Από εκεί και πέρα η κωδικοποίηση του Client μπορεί να γίνει σε οποιαδήποτε γλώσσα προγραμματισμού : Java, PHP, VB. Η μόνη απαίτηση είναι να μπορεί η γλώσσα αυτή να χειριστεί δεδομένα XML
- Ανεξάρτητα γλώσσας προγραμματισμού όπως αυτό προκύπτει από την χρήση προτύπων XML

- Δεν βασίζεται σε προγραμματιστικά μοντέλα τύπων δεδομένων
- Τα Web Services στηρίζονται σε πρότυπα (HTTP, XML, SOAP) τα οποία είναι ήδη καθιερωμένα, έχουν επικρατήσει ευρέως και έχουν αποδείξει την αξία τους μέσα από την χρήση τους.
- Η χρήση του πρωτοκόλλου HTTP αποτελεί από μόνη της ένα πλεονέκτημα για τα Web Services καθώς τα Firewall δεν μπλοκάρουν τις τυποποιημένες θύρες που χρησιμοποιεί
- Το μεγαλύτερο πλεονέκτημα των web services αποτελεί το λεγόμενο Loose Coupling (χαλαρή συνδεσιμότητα) ανάμεσα στον Client και στον Server. Ο Client δεν χρειάζεται να γνωρίζει τίποτα για τον Server και για τον τρόπο που αυτός θα εκτελέσει την υπηρεσία παρά μόνο τα ορίσματα που αυτός χρειάζεται.

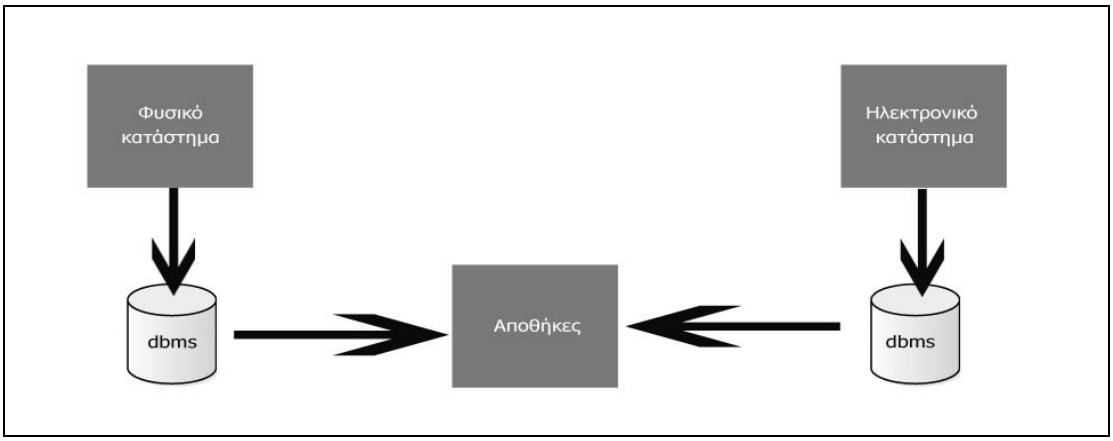
4.4.7 Μειονεκτήματα των Web Services

- Ένα βασικό μειονέκτημα των web services αποτελεί ο χρόνος απόκρισης. Η μετάδοση των δεδομένων σε μορφή XML δεν μπορεί να είναι το ίδιο αποδοτική όπως με τη χρήση δυαδικού κώδικα. Αυτό που κερδίζεται σε διαλειτουργικότητα χάνεται σε απόδοση. Βέβαια, παρόλα αυτά ο επιπλέον αυτός χρόνος απόκρισης είναι αποδεκτός για τις περισσότερες εφαρμογές αν και πολύ σπάνια αν όχι ποτέ επιλέγεται η χρήση των web service σε εφαρμογές πραγματικού χρόνου (Real-Time)
- Η έλλειψη πολλαπλότητας στην εφαρμογή τους. Μέχρι σήμερα τα Web Services έχουν περιορισμένη εφαρμογή και επιτρέπουν πολύ βασικές μορφές κλήσης υπηρεσιών. Ευτυχώς, βέβαια, εμφανίζονται στο προσκήνιο πολλές τυποποιήσεις που θα βοηθήσουν τα Web Services να γίνουν όλο και πιο πολύπλευρα.

5.ΥΛΟΠΟΙΗΣΗ

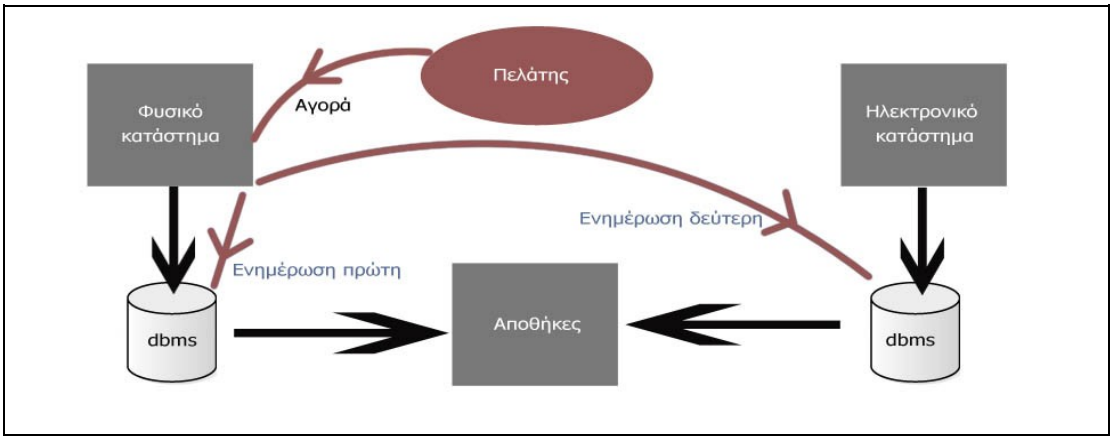
5.1 Μια πρώτη προσέγγιση

Όπως αναφέρθηκε και σε προηγούμενα κεφάλαια το υπόβαθρο του εγχειρήματος είναι το εξής : μια εταιρία διατηρεί κατάστημα και εμπορεύεται διάφορα αντικείμενα. Ο πελάτης μπορεί να περάσει από τα κατάστημα, να αγοράσει το προϊόν της αρεσκείας του, να κοπεί το σχετικό παραστατικό και να φύγει. Ταυτόχρονα η εταιρεία διατηρεί και ένα ηλεκτρονικό κατάστημα. Ο εκάστοτε πελάτης μπορεί να αγοράσει ένα προϊόν και από εκεί. Τα δύο συστήματα διατηρούν δύο βάσεις δεδομένων. Οι αποθήκες, όμως, της εταιρίας είναι κοινές και για το ηλεκτρονικό και για το φυσικό κατάστημα Αυτή η κατάσταση παρουσιάζεται σχηματικά αμέσως παρακάτω. (Σχήμα 7).



Σχήμα 7

Επομένως, μια συναλλαγή με το ένα σύστημα χωρίς την αντίστοιχη ενημέρωση του άλλου θα του δημιουργούσε πρόβλημα στην ακεραιότητα των δεδομένων του. Αυτό που χρειάζεται είναι όταν ένας πελάτης αγοράζει ένα προϊόν στο φυσικό κατάστημα να ενημερώνεται αυτόματα και χωρίς καμιά επιπλέον ενέργεια με την ίδια ποσότητα η βάση δεδομένων και του ηλεκτρονικού καταστήματος (Σχήμα 8). Τα αντίστοιχα είναι απαραίτητο να συμβαίνουν στην αντίστροφη περίπτωση που η συναλλαγή του πελάτη γίνεται από το ηλεκτρονικό κατάστημα. Είναι φανερό ότι ένα τέτοιο σύστημα με δύο βάσεις δεδομένων των οποίων τα δεδομένα περιγράφουν ένα κοινό αντικείμενο του φυσικού κόσμου, την αποθήκη, μπορεί να χαρακτηριστεί σαν κατανεμημένο.



Σχήμα 8

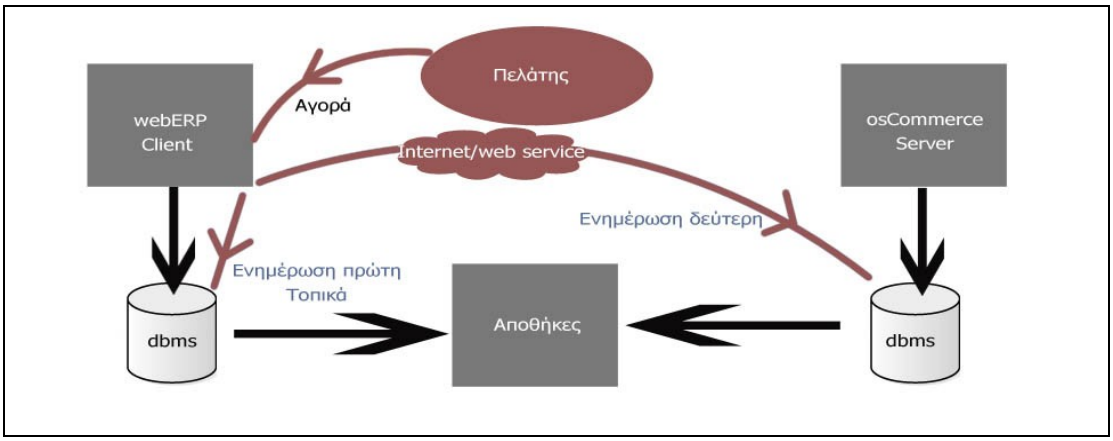
Πρώτα από όλα χρειάζεται να εξεταστεί αν υπάρχει η διαθέσιμη τεχνολογία για την απαίτηση που τέθηκε, δηλαδή την αυτόματη ενημέρωση της βάσης του ηλεκτρονικού καταστήματος κάθε φορά που γίνεται μια συναλλαγή από το φυσικό κατάστημα και αντίστροφα. Αναφέρθηκε και προηγουμένως ότι πρόκειται για μια μορφή κατανεμημένου συστήματος. Φυσικά στον τομέα αυτό η επιστήμη των υπολογιστών έχει αρκετές τεχνολογίες διαθέσιμες και οι οποίες παρουσιάστηκαν σε προηγούμενο κεφάλαιο. Πλεονεκτήματα των Web Services, όπως η χαλαρή συνδεσιμότητα μεταξύ Server και Client προσφέρουν σημαντική ευελιξία σε περίπτωση που αλλάξει η υλοποίηση στον Server. Το σημαντικότερο πλεονέκτημα, βέβαια, των Web Services που είναι η διαλειτουργικότητα ανεξαρτήτου πλατφόρμας και γλώσσας προγραμματισμού, δεν ξεχωρίζει στην προκειμένη περίπτωση καθώς και τα δύο συστήματα είναι δομημένα σε PHP.

Για να επιτευχθεί η ενημέρωση μπορεί κανείς να σκεφτεί πολλές περιπτώσεις. Θα μπορούσε θεωρητικά να γίνει μέχρι και χειροκίνητα παρεμβαίνοντας ο άνθρωπος : κάθε φορά που γίνεται μια συναλλαγή στο φυσικό κατάστημα κάποιος να αναλαμβάνει τις απαραίτητες ενέργειες για να ενημερωθεί η βάση του ηλεκτρονικού καταστήματος. Φυσικά κάτι τέτοιο μόνο σαν απαράδεκτη λύση μπορεί να χαρακτηριστεί γιατί καταρχήν δεν γίνεται αυτόματα οπότε συνεπάγεται καθυστερήσεις, αλλά και γιατί ο άνθρωπος έχει μεγάλη πιθανότητα να κάνει κάποιο λάθος στην εισαγωγή των δεδομένων στη βάση είτε ενημερώνοντας με λάθος ποσότητα τον σωστό κωδικό προϊόντος, είτε ενημερώνοντας με σωστή ποσότητα τον λάθος κωδικό προϊόντος είτε συνδυάζοντας και τα δύο παραπάνω σφάλματα.

Προκείμενου να αυτοματοποιηθεί, λοιπόν, η διαδικασία αλλά και να εξαλειφθούν τα σφάλματα χρειάζεται να χρησιμοποιηθούν τεχνολογίες όπως αυτές των Web Services που να επιτρέπουν σε δύο συστήματα τα οποία είναι απομακρυσμένα μεταξύ τους να επικοινωνούν και να ανταλλάσσουν δεδομένα.

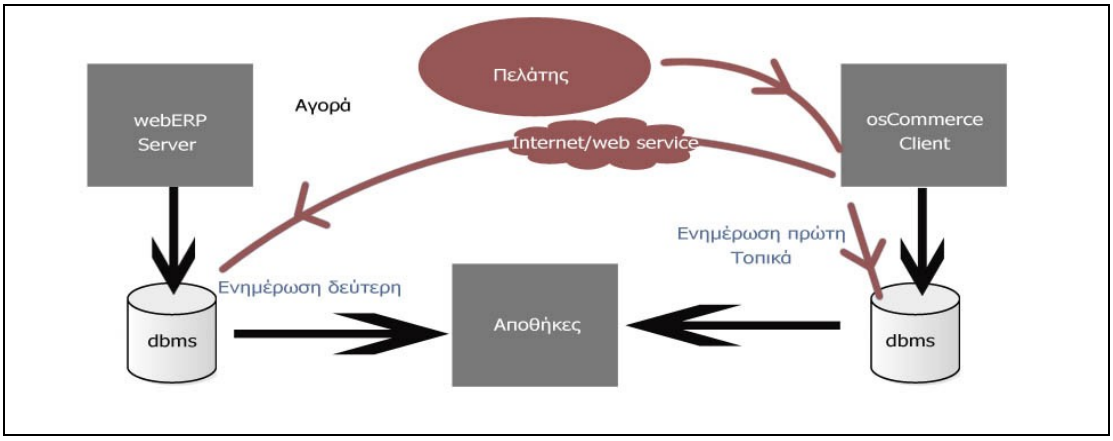
5.2 Μελέτη εφικτότητας

Αρχικά, χρειάζεται να ξεχωρίσουμε τα δύο είδη επικοινωνίας που λαμβάνουν χώρα κατά την εκτέλεση μιας συναλλαγής. Αν αυτή εκτελείται από την πλευρά του ERP συστήματος τότε η ενημέρωση ξεκινάει από αυτό με κατεύθυνση το ηλεκτρονικό κατάστημα. Αυτό σημαίνει ότι σε αυτή την περίπτωση τον ρόλο του Client τον παίζει το ERP σύστημα γιατί αυτό είναι που αιτείται μια ενημέρωση της βάσης του ηλεκτρονικού καταστήματος το οποίο αποτελεί τον Server επειδή αυτό εκτελεί την υπηρεσία ενημέρωσης της βάσης δεδομένων του (Σχήμα 9).



Σχήμα 9

Τα δεδομένα αλλάζουν, όμως, όταν η συναλλαγή εκτελεστεί από το ηλεκτρονικό κατάστημα. Η διαδικασία της ενημέρωσης ακολουθεί την αντίστροφη πορεία. Και οι ρόλοι αντιστρέφονται. Client αποτελεί το ηλεκτρονικό κατάστημα γιατί αιτείται την ενημέρωση της βάσης δεδομένων του ERP και Server γίνεται το ERP το οποίο πρέπει να εκτελέσει την υπηρεσία (Σχήμα 10). Για την αμφίδρομη επομένως λειτουργία της διαδικασίας ενημέρωσης χρειάζεται και τα δύο συστήματα να είναι σε θέση να διαδραματίσουν και τους δύο ρόλους, τόσο του Server όσο και του Client.

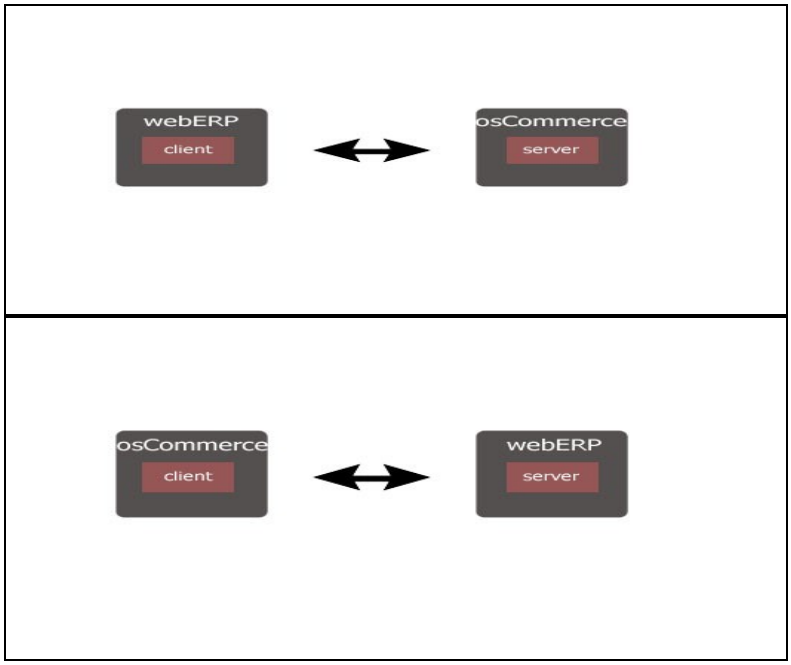


Σχήμα 10

Όπως αναφέρθηκε σε προηγούμενα κεφάλαια τόσο το WebERP όσο και το Oscommerce είναι γραμμένα στη γλώσσα προγραμματισμού PHP. Τα Web Services επιτρέπουν την επικοινωνία μεταξύ συστημάτων διαφορετικών πλατφόρμων αλλά και διαφορετικών γλωσσών. Επομένως θα μπορούσε το ένα σύστημα να ήταν δομημένο σε διαφορετική γλώσσα από το άλλο. Επειδή αυτό δεν ισχύει στην εξεταζόμενη περίπτωση το Web Service τόσο από την πλευρά του Server όσο και του Client θα υλοποιηθεί με την γλώσσα PHP. Στην συγκεκριμένη εργασία θα δούμε και τις δύο περιπτώσεις είτε ο πελάτης επισκέπτεται το «φυσικό» κατάστημα οπότε το webERP αποτελεί τον Client ενώ το osCommerce αποτελεί τον Server, είτε η συναλλαγή γίνεται από το ηλεκτρονικό κατάστημα οπότε το osCommerce αποτελεί τον Client και το webERP τον Server.

Γενικός σχεδιασμός

Για την επικοινωνία μεταξύ webERP και osCommerce θα ενσωματωθεί στον υπάρχοντα κώδικα των δύο προγραμμάτων ένας Server και ένας Client.



Σχήμα 11

Ο σκοπός του Server θα είναι να ενημερώνει την βάση του ενός συστήματος με τη σωστή ποσότητα ενός συγκεκριμένου είδους για το οποίο έγινε συναλλαγή στο άλλο σύστημα.

Επομένως, ξεχωρίζουμε δύο δεδομένα εισόδου τα οποία χρειάζεται ο Server για να εκτελέσει την λειτουργία του:

- **Ποσότητα προϊόντος όπως προέκυψε μετά την ενημέρωση του εκάστοτε συστήματος**
- **Κωδικό προϊόντος στη βάση του συστήματος**

Μεταφράζοντάς τα σε ονόματα μεταβλητών θα έχουμε :

Για τον Server στο osCommerce :

- **stock_left (int)**
- **product_id (int)**

Για τον Server στο webERP :

- **stock_left(int)**

- **product_id(string)**

Η λειτουργία που θα επιτελείται από τον εκάστοτε Server είναι μία και είναι η ενημέρωση της βάσης δεδομένων του. Ορίζεται, συνεπώς, μια συνάρτηση η οποία θα ονομάζεται **update_quanti** η οποία θα δέχεται σαν ορίσματα τα προηγούμενα δεδομένα.

Σαν έξοδο θα παράγεται μια μεταβλητή που θα περιέχει την ποσότητα στην βάση μετά την ενημέρωση από το web service. Η ποσότητα αυτή στέλνεται πίσω στον client ο οποίος την συγκρίνει με την ποσότητα στη δική του βάση. Αν είναι ίδιες η ενημέρωση έγινε επιτυχώς και στα δύο συστήματα.

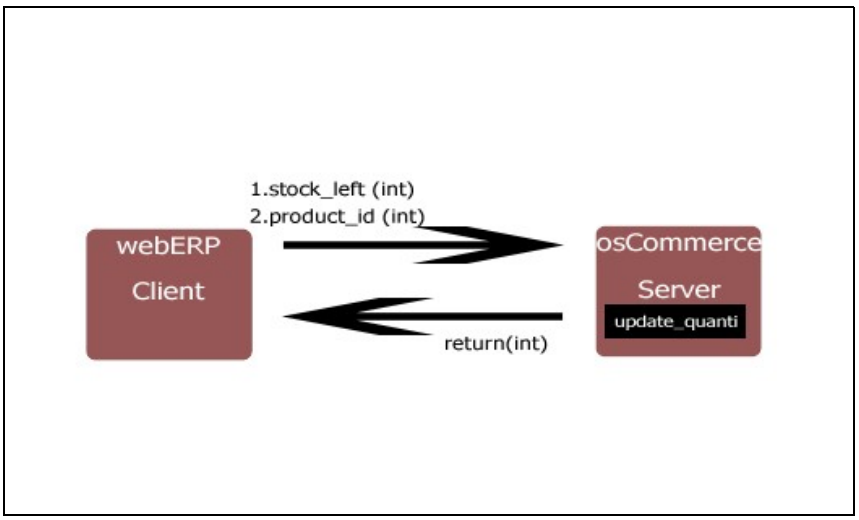
Άρα, ξεχωρίζουμε ένα δεδομένο εξόδου το οποίο παράγει ο Server

- **Η ποσότητα στη βάση δεδομένων του απομακρυσμένου συστήματος όπως αυτή προέκυψε μετά την ενημέρωση από το web service**

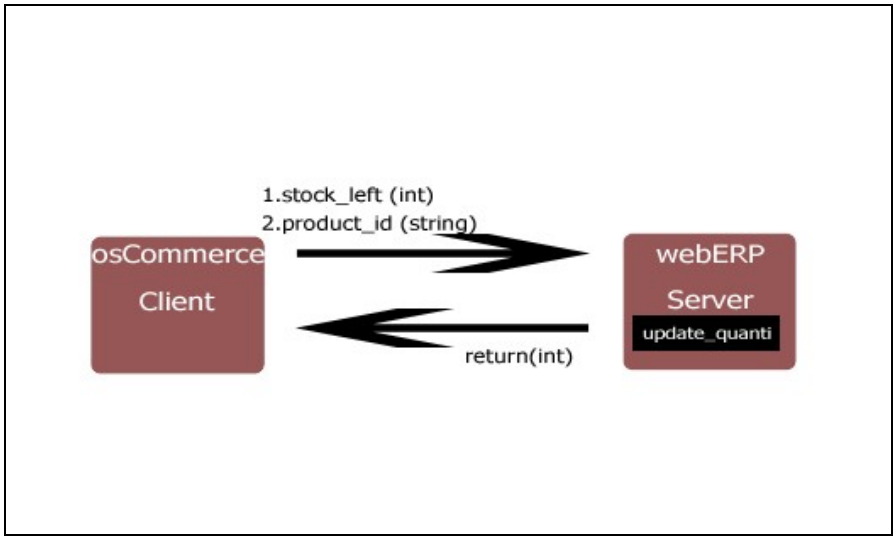
Μεταφράζοντάς το σε όνομα μεταβλητής θα έχουμε :

- **return (int)**

Όλα τα παραπάνω παρουσιάζονται σχηματικά στο Σχήμα 12



Σχήμα 12α



Σχήμα 12β

Η ποσότητα που επιστρέφεται από το web service συγκρίνεται με την ποσότητα στο σύστημα που ξεκίνησε την επικοινωνία. Στην περίπτωση που η έναρξη έγινε από το webERP ο χρήστης λαμβάνει ένα μήνυμα που τον ενημερώνει αν η ενημέρωση είναι επιτυχής (οι δύο ποσότητες είναι ίσες) ή ανεπιτυχής (οι δύο ποσότητες είναι άνισες). Αν η έναρξη έγινε από το osCommerce τότε δεν έχει νόημα να επιστρέψουμε κάποιο μήνυμα στον χρήστη-πελάτη ότι ενημερώθηκε επιτυχώς ή ανεπιτυχώς η βάση του webERP. Μία τέτοια πληροφορία του είναι αδιάφορη. Στην περίπτωση αυτή, λοιπόν, ενσωματώνουμε μια δομή επανάληψης η οποία επαναλαμβάνει την κλήση στο web service μέχρι η ποσότητα που θα επιστραφεί από αυτό να είναι ίδια με αυτή του osCommerce.

5.3 **NuSoap - REST**

Όπως αναφέρθηκε και παραπάνω χρειάζεται να υλοποιηθεί ένας Server και ένας Client τα οποία θα τρέχουν τόσο στο webERP όσο και στο osCommerce. Και τα δύο θα υλοποιηθούν σε PHP καθώς αμφότερα είναι δομημένα σε αυτή την γλώσσα προγραμματισμού. Όταν ο πελάτης θα μπαίνει στο κατάστημα για να αγοράσει ένα προϊόν με την έκδοση του σχετικού τιμολογίου ο Client που τρέχει στο webERP θα κάνει μια αίτηση στον Server του osCommerce για να ενημερώσει αυτός την βάση δεδομένων του. Αντίστοιχα, όταν, ο ηλεκτρονικός επισκέπτης του καταστήματος

πραγματοποιεί μια αγορά μέσω του osCommerce τότε ο Client που τρέχει σε αυτό ειδοποιεί τον Server του webERP να ενημερώσει την βάση του με τη σωστή ποσότητα.

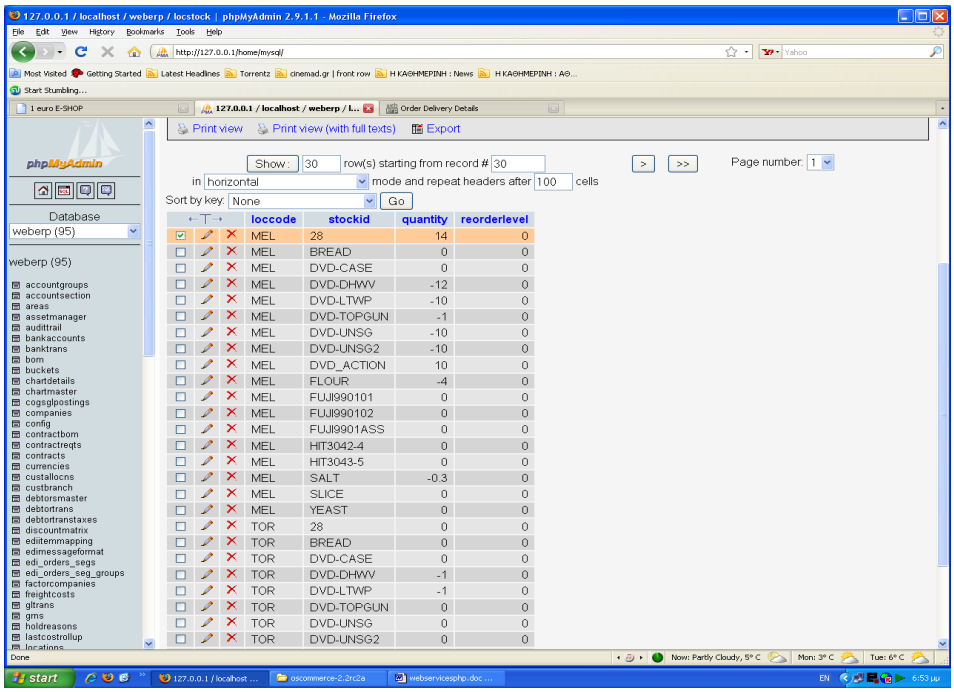
Χρειάζεται, επομένως, να εξεταστούν οι διαθέσιμες τεχνολογίες που προσφέρει η PHP στο θέμα της υλοποίησης των web services. Τα μηνύματα που ανταλλάσσουν Server-Client δύναται να γίνεται μέσω HTTP. Η μέθοδος αυτή που χρησιμοποιεί το πρωτόκολλο HTTP για τη μεταφορά των XML μηνυμάτων ονομάζεται REST. Η λέξη REST αποτελεί το ακρωνύμιο για το Representational State Transfer. Το REST δεν αποτελεί κάποιο πρότυπο. Για αυτό και οργανισμοί όπως ο W3C δεν έχουν εκδώσει κάποια τυποποίηση για αυτό. Το REST αποτελεί μια αρχιτεκτονική για τον σχεδιασμό των web services η οποία χρησιμοποιεί υπάρχοντα πρότυπα, όπως HTTP, XML, URL κτλ. Στην PHP η μέθοδος REST υλοποιείται με την ενσωματωμένη βιβλιοθήκη SimpleXML η οποία παρέχει μεθόδους και συναρτήσεις για τον χειρισμό της XML και την μετατροπή των εγγράφων XML σε δεδομένα που μπορούν να χρησιμοποιηθούν από μια γλώσσα προγραμματισμού.

Ο άλλος τρόπος για την επικοινωνία μεταξύ Server και Client είναι η χρήση του πρωτοκόλλου SOAP. Για αυτή την περίπτωση έχει δημιουργηθεί η εξωτερική βιβλιοθήκη NuSOAP. Η NuSOAP είναι ένα σύνολο από κλάσεις που επιτρέπει στους προγραμματιστές να δημιουργούν και να καταναλώνουν SOAP Web Services. Δεν απαιτεί κάποιες ιδιαίτερες PHP επεκτάσεις. Η πιο πρόσφατη έκδοση της βιβλιοθήκης είναι η 0.7.1 (27-7-2005) και υποστηρίζει τα περισσότερα χαρακτηριστικά του SOAP 1.1. Μπορεί να αναπαράγει αλλά και να «καταναλώσει» WSDL 1.1 . Διανέμεται με την άδεια GNU ή LGPL (Lesser General Public Liscense)

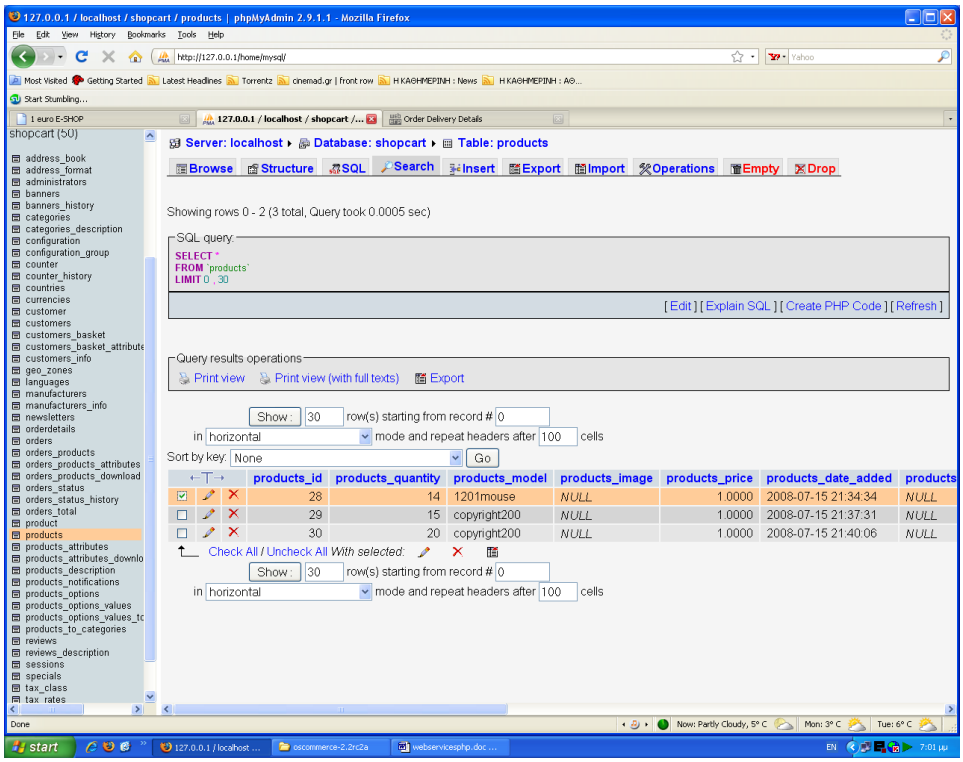
Στην συγκεκριμένη εργασία χρησιμοποιήθηκε η εξωτερική βιβλιοθήκη NuSOAP για την υλοποίηση των Web Services.

5.4Σενάριο χρήσης

Ας εξετάσουμε τώρα ένα σενάριο χρήσης του συστήματος. Στο ακόλουθο σενάριο υποθέτουμε ότι κάποιος πελάτης επισκέπτεται το φυσικό μαγαζί και εκτελεί μια αγορά έστω δέκα ποντίκια για τον υπολογιστή. Μετά την εκτέλεση της αγοράς και της ενημέρωσης της βάσης του ERP συστήματος χρειάζεται, να ενημερωθεί και η βάση του ηλεκτρονικού καταστήματος. Ας δούμε βήμα βήμα τη διαδικασία που ακολουθείται.

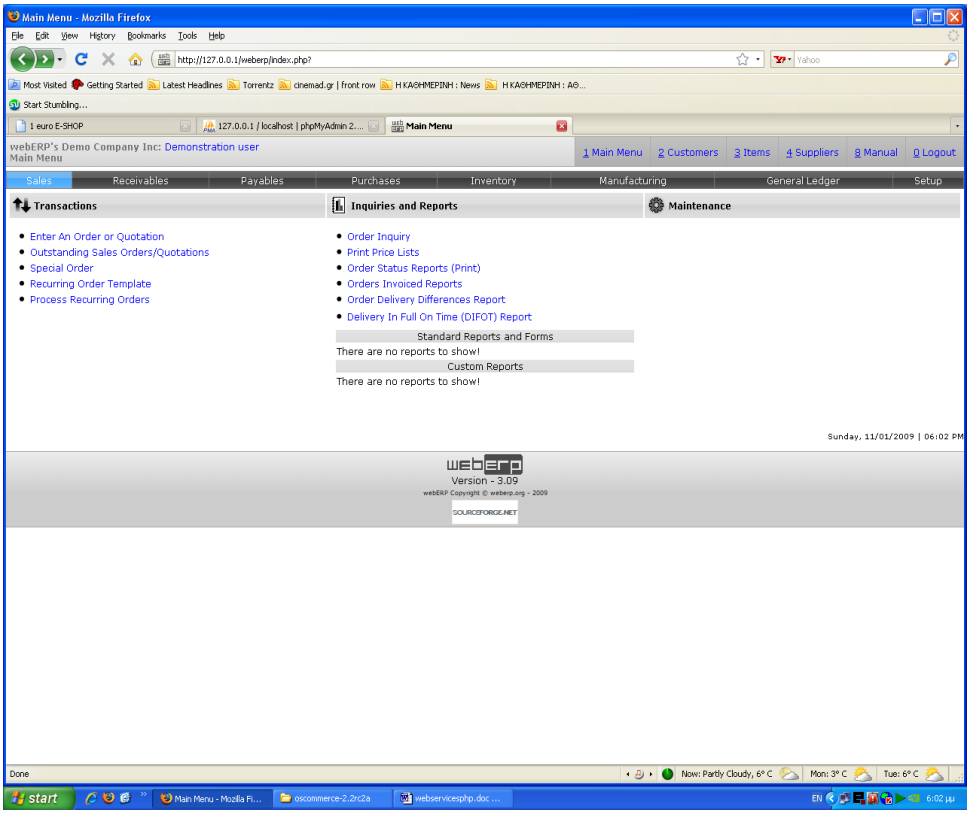


Ο πίνακας lockstock της βάσης δεδομένων του webERP. Διακρίνεται ότι το είδος με stockid 28 (το ποντίκι δηλαδή) έχει ποσότητα 14 τεμάχια.

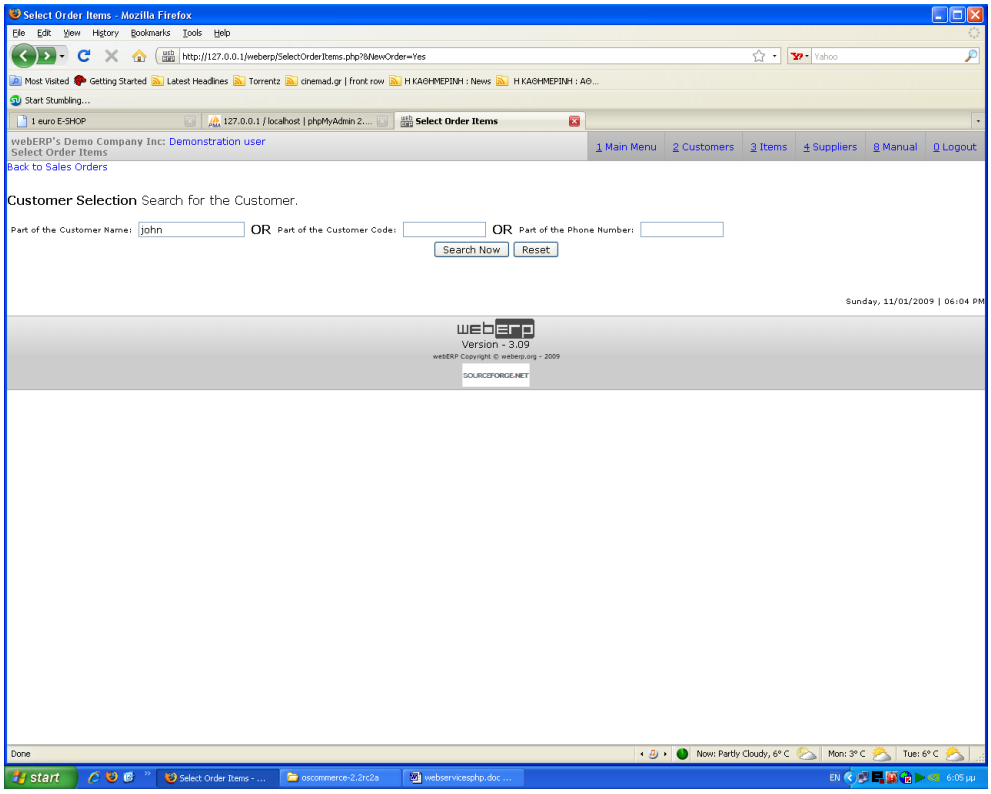


Αντίστοιχα ο πίνακας products του osCommerce που διακρίνεται ότι το είδος με product_id 28 (το ποντίκι) έχει ποσότητα διαθέσιμη 14 τεμάχια.

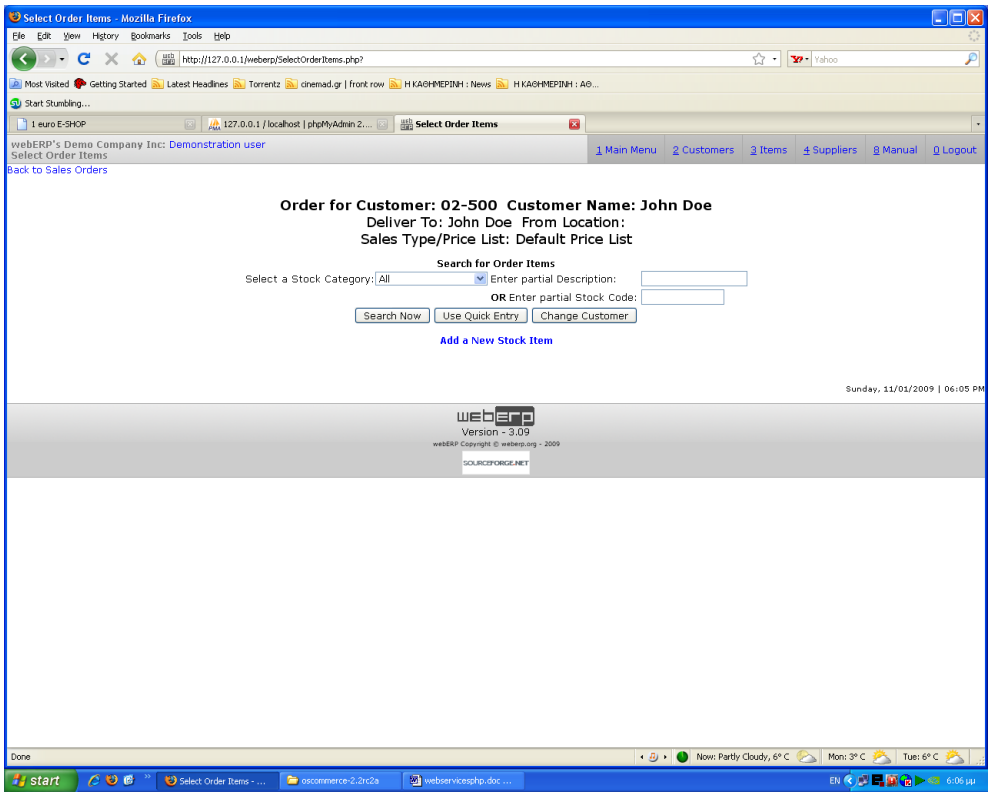
Ας περάσουμε στην εισαγωγή της παραγγελίας στο webERP. Στη βασική οθόνη του webERP πατάμε στο **Enter An Order or Quotation** για να εισάγουμε μια παραγγελία πελάτη.



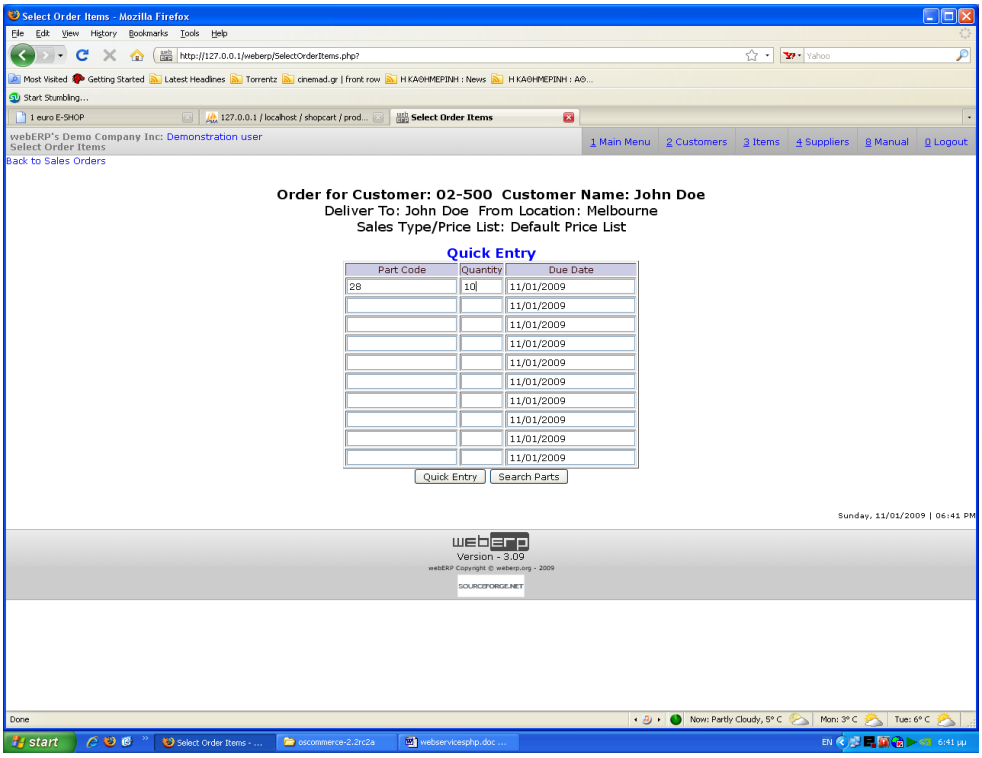
Στην επόμενη οθόνη δίνουμε το όνομα του πελάτη έστω john Doe και πατάμε το **Search Now**.



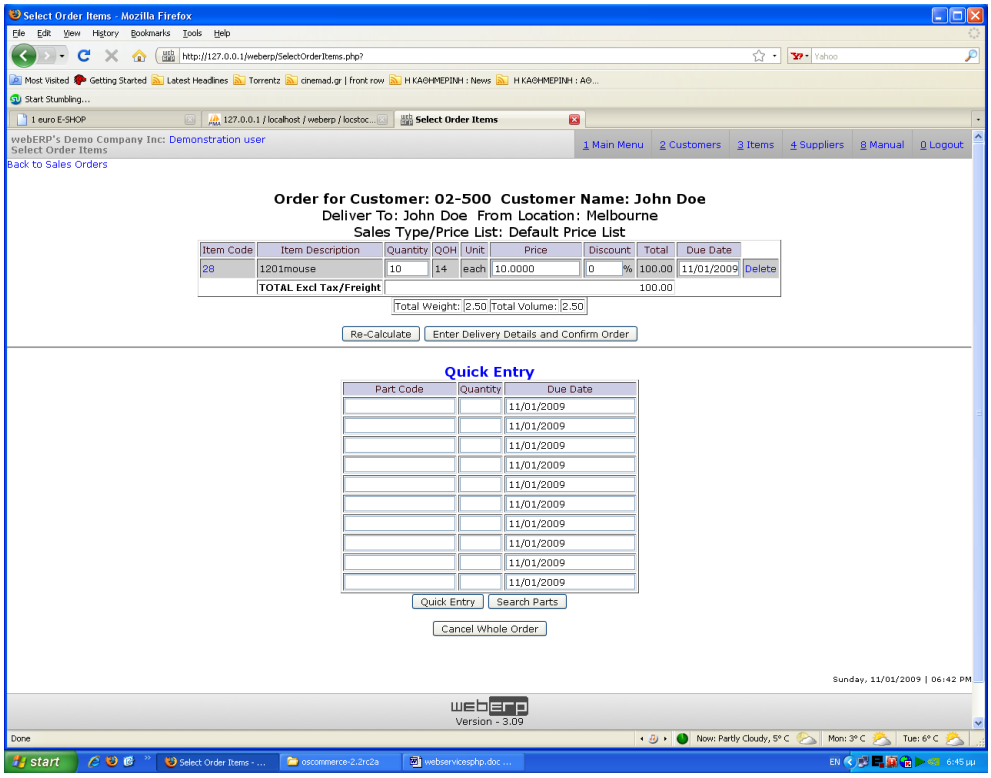
Η αναζήτηση μας αποδίδει τα στοιχεία του ζητούμενου πελάτη.



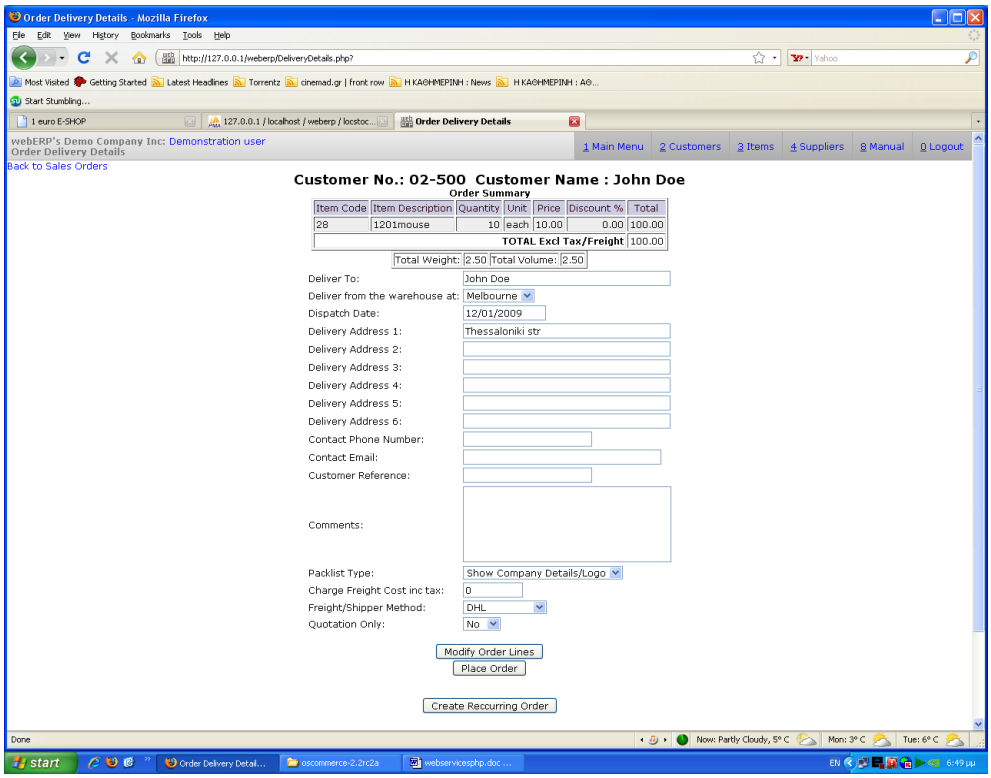
Πατώντας τώρα στο **Use quick Entry** εισάγουμε τον κωδικό του είδους που επιθυμεί να αγοράσει ο πελάτης και την αντίστοιχη ποσότητα. Στην περίπτωσή μας ο κωδικός είδους για το ποντίκι είναι 28 και η ποσότητα είναι 10.



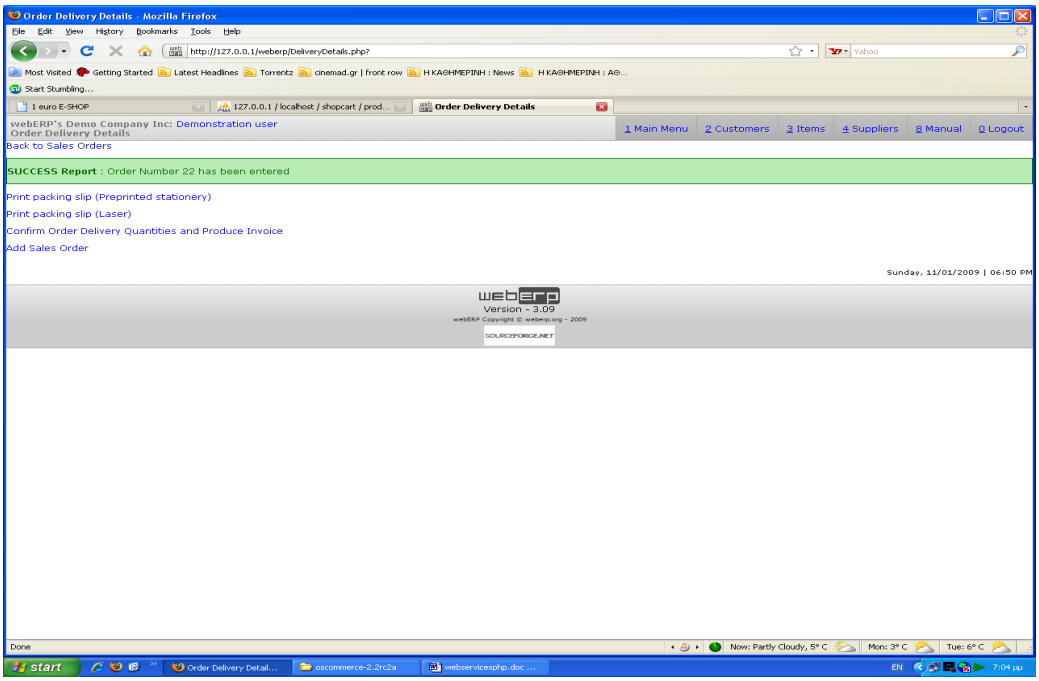
Πατάμε το **Quick Entry** και εμφανίζεται η παραγγελία μας



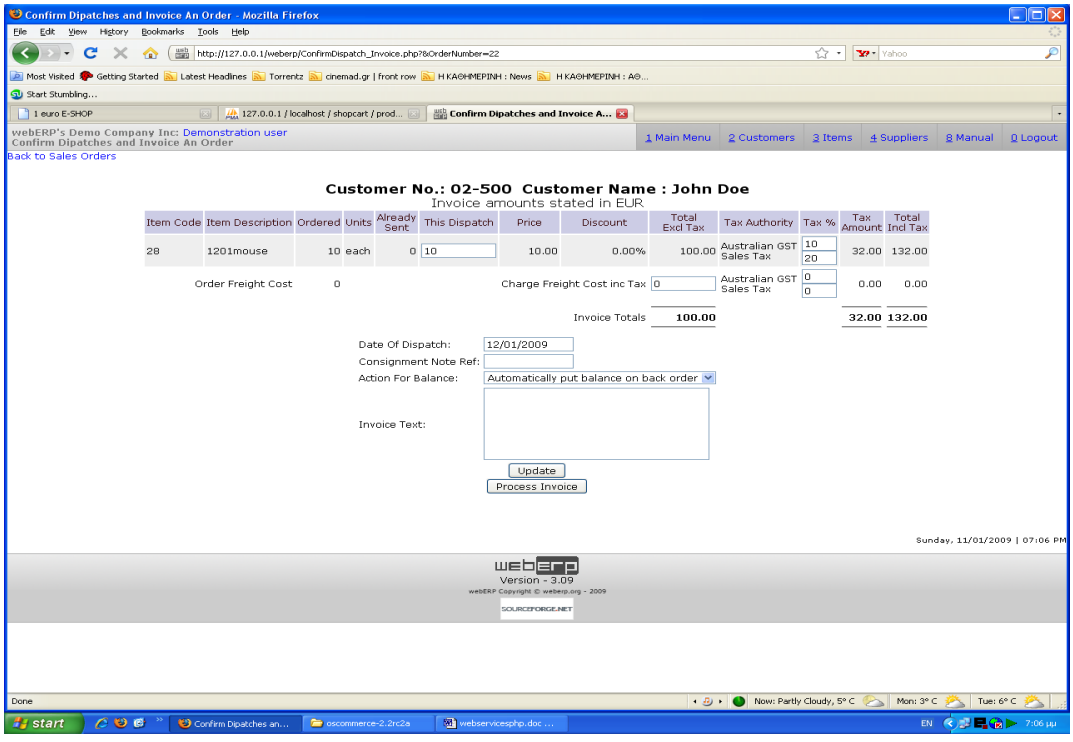
Στη συνέχεια πατάμε στο **Enter Delivery Details and Confirm Order** για να δώσουμε τα στοιχεία παράδοσης και να επικυρώσουμε την παραγγελία.



Αφού κάνουμε τις απαραίτητες αλλαγές πατάμε στο **Place Order.**



Η παραγγελία δρομολογήθηκε. Πατάμε στο **Confirm Order Delivery Quantities and Produce Invoice** για να γίνει η τιμολόγηση.



Αν δεν κάνουμε καμιά αλλαγή πατάμε στο **Process Invoice** και ολοκληρώνεται η παραγγελία. Με μια αναζήτηση στους αντίστοιχους πίνακες της κάθε βάσης δεδομένων θα διαπιστώσουμε ότι η ποσότητα και στα δύο συστήματα μειώθηκε κατά 10.

6. ΣΥΜΠΕΡΑΣΜΑΤΑ

Στα πλαίσια της εργασίας έγινε μια προσπάθεια συγχρονισμού δύο βάσεων δεδομένων. Η μια βάση ήταν μέρος ενός ERP συστήματος και η άλλη ενός ηλεκτρονικού καταστήματος. Για να επιτευχθεί ο σκοπός χρησιμοποιήθηκε τεχνολογία των Web Servces. Κατ' αυτόν τον τρόπο κάθε φορά που ενημερώνεται η βάση δεδομένων του ERP συστήματος μέσω του Web Service που τρέχει στο ηλεκτρονικό κατάστημα ενημερώνεται και η βάση του τελευταίου. Η χρήση των Web Service σαν τεχνολογία κατανεμημένων συστημάτων κρίνεται επιτυχής. Ο στόχος που ήταν η αυτόματη ενημέρωση της βάσης του ηλεκτρονικού καταστήματος, χωρίς την ανθρώπινη παρέμβαση επετεύχθη εξασφαλίζοντάς την από τυχόν λάθη που θα προέκυπταν σε περίπτωση καταχώρισης από κάποιο άνθρωπο.

Τόσο ο Client όσο και ο Server κωδικοποιήθηκε σε γλώσσα PHP. Αυτό έγινε γιατί τόσο το ERP όσο και το ηλεκτρονικό κατάστημα είναι δομημένα σε PHP. Τα Web Services, βέβαια, θα επέτρεπαν την επικοινωνία ανάμεσα στα δύο συστήματα ακόμα και αν ήταν δομημένα σε διαφορετικές γλώσσες. Η PHP παρέχει δύο τρόπους για την υλοποίηση των Web Service. Η μία είναι η REST, η οποία είναι γνωστή και ως Web Service μέσω HTTP μιας και τα αιτήματα αλλά και οι XML αποκρίσεις στέλνονται μέσω HTTP. Η άλλη μέθοδος είναι χρήση του τυπικού για τα Web Services πρωτοκόλλου SOAP. Για την υλοποίηση αυτής της μεθόδου υπάρχει διαθέσιμη η εξωτερική βιβλιοθήκη NuSOAP. Στην παρούσα εργασία χρησιμοποιήθηκε η δεύτερη μέθοδος και η βιβλιοθήκη NuSOAP.

Για την λειτουργία του Web Service υλοποιήθηκε μια συνάρτηση η οποία δέχεται σαν ορίσματα την νέα διαθέσιμη ποσότητα και τον κωδικό του είδους και ενημερώνει τη βάση του ηλεκτρονικού καταστήματος. Η συνάρτηση αυτή ενθυλακώθηκε στο Web Service με τη βοήθεια του NuSOAP. Το NuSOAP παρέχει πολλές ευκολίες στον χρήστη που υλοποιεί ένα Web Service με τη βοήθεια του. Μια από τις πιο σημαντικές είναι ότι αναλαμβάνει να δημιουργήσει το WSDL του Web Service χωρίς την παραμικρή ανάμιξη του χρήστη. Όσο πιο πολύπλοκο γίνεται το web service τόσο πιο χρήσιμη αναδεικνύεται αυτή η δυνατότητα.

Τα προγράμματα που χρησιμοποιήθηκαν για την πραγματοποίηση του στόχου, το webERP και το osCommerce, χαρακτηρίζονται σίγουρα καλά δομημένα και συνεπώς ο χρήστης/ προγραμματιστής εύκολα μπορεί να μπει στην λογική τους και να καταλάβει τι είδους διαδικασίες συμβαίνουν κάθε στιγμή. Σε αυτό συμβάλλει και

η ύπαρξη της απαραίτητης γραπτής τεκμηρίωσης (Documantation). Αυτό αποτελεί ιδιαίτερο πλεονέκτημα μιας και είναι προγράμματα ανοιχτού κώδικα. Συνεπώς,ο τελευταίος παρέχεται στους χρήστες οι οποίοι μπορούν να τον προσαρμόσουν σύμφωνα με τις ανάγκες της εκάστοτε επιχείρησης. Οι ενεργές κοινότητες που στηρίζουν τα δύο αυτά συστήματα, άλλωστε, δίνουν επιπλέον βοήθεια σε πιθανά προβλήματα που μπορεί να ανακύψουν κατά την προσαρμογή των προγραμμάτων, μέσα από Forum αλλά και βάσεις με τη συσσωρευμένη γνώση που προέκυψε από τις διαχρονικές προσπάθειες των χρηστών/ προγραμματιστών.

7. ΠΑΡΑΡΤΗΜΑ Α

7.1 Επικοινωνία από webERP προς osCommerce

7.1.1 Ο κώδικας του osCommerce server

Ο κώδικας του Server τοποθετήθηκε σε ένα αρχείο με το όνομα soap_serv.php. Είναι το αρχείο που θα αναλάβει να εξυπηρετήσει τυχών αιτήσεις ενημέρωσης της βάσης δεδομένων του ηλεκτρονικού καταστήματος και βρίσκεται φυσικά στον κατάλογο του osCommerce με διαδρομή **oscommerce-2.2rc2a/catalog/includes/functions/soap_serv.php** και είναι ο ακόλουθος :

Ο PHP κώδικας του server :

```
<?php

// Εισάγεται η βιβλιοθήκη NuSOAP στο αρχείο για μπορούν να χρησιμοποιηθούν οι
//συναρτήσεις της
require_once('lib/nusoap.php');

//Εισάγεται το αρχείο με τις συναρτήσεις χειρισμού της βάσης δεδομένων
require('database.php');

//Εισάγεται το αρχείο με τα ονόματα των πινάκων της βάσης
include('../database_tables.php');

// Include application configuration parameters
require('../configure.php');

// Δημιουργούμε ένα στιγμιότυπο του αντικειμένου soap_server
$server = new soap_server();

//Αρχικοποιούμε την υποστήριξη WSDL
$server->configureWSDL('soap_serv', 'urn:soap_serv');
```

```
// Εγγράφουμε την μέθοδο που θα δημοσιεύσουμε για χρήση
//Δηλώνουμε επίσης τα δεδομένα εισόδου αλλά και την έξοδο
//της μεθόδου. Βλέπουμε στην συγκεκριμένη περίπτωση η μέθοδος
//έχει όνομα  update_quanti, δέχεται 2 ορίσματα σαν είσοδο που
//είναι ακέραιοι αριθμοί (int) και επιστρέφει την έξοδο σε μορφή
//ακεραίου (int)
$server->register('update_quanti',
        array('product_id' => 'xsd:int','stock_left'=>'xsd:int'),
                array('return'=>'xsd:int'),
                'urn:soap_serv',
                'urn:soap_serv#update_quanti',
                'rpc',
                'encoded',
                'update quantity to db');
// Ορίζουμε την μέθοδο σαν μια απλή συνάρτηση PHP
function update_quanti($product_id,$stock_left) {
//Χρησιμοποιούμε τις διαθέσιμες συναρτήσεις από το πρόγραμμα που εισαχθήκανε
// με το αρχείο database.php
  tep_db_connect() or die('Unable to connect to database server!');
  //Χρησιμοποιούμε τις ενσωματωμένες συναρτήσεις (tep_db_query) για την πράξη
//του update στον πίνακα της βάσης. Την ποσότητα και τον κωδικό προϊόντος την
//παίρνουμε απο τα ορίσματα της συνάρτησης

  tep_db_query("update " . TABLE_PRODUCTS . " set products_quantity = '" .
$stock_left . "' where products_id = '" . $product_id . "'");

//Κάνουμε το ερώτημα στη βάση για να βρούμε την ποσότητα μετά την ενημέρωση
$return_query= tep_db_query("select products_quantity from " .
TABLE_PRODUCTS . " where products_id =  '" . $product_id . "'");

  $return=tep_db_fetch_array($return_query);
  tep_db_close();
//Το αποτέλεσμα του ερωτήματος είναι αυτό που επιστρέφει το web service
```

```
 return $return['products_quantity'];
  }
$HTTP_RAW_POST_DATA = isset($HTTP_RAW_POST_DATA) ?
$HTTP_RAW_POST_DATA : '';
$server->service($HTTP_RAW_POST_DATA);
?>
```

7.1.2 Το WSDL αρχείο

Δημιουργώντας το αρχείο soap_serv.php αυτομάτως η βιβλιοθήκη NuSOAP αναγνωρίζει ότι πρόκειται για Web Service και παράγει χωρίς καμιά περαιτέρω ενέργεια το WSDL αρχείο του Web Service το οποίο και παρατίθεται παρακάτω . Απλά με τη βοήθεια ενός Browser χρειάζεται να δείξουμε στο αρχείο που έχει τον Server και θα εμφανιστεί η ακόλουθη εικόνα (Σχήμα 13).

Σχήμα 13

WSDL :

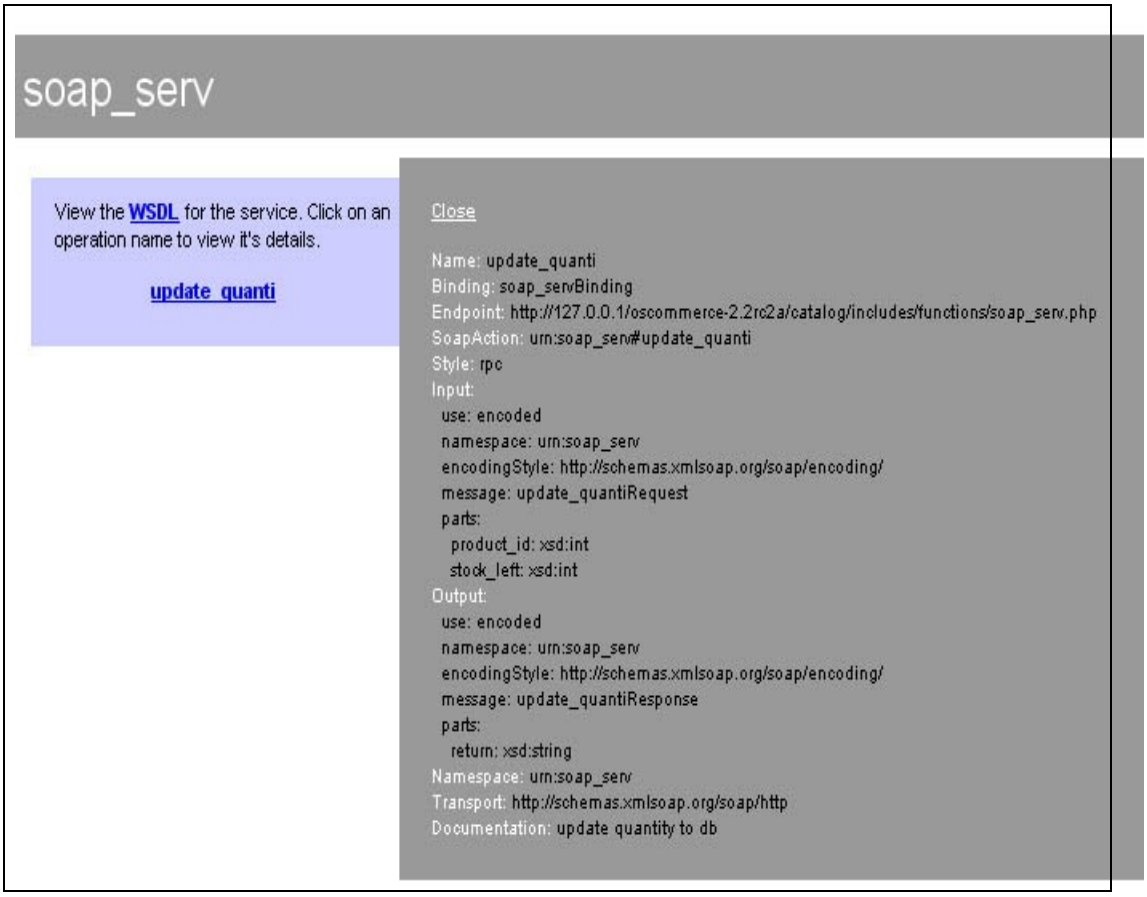
```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

```
- <definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
    ENC="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:tns="urn:soap_serv"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    targetNamespace="urn:soap_serv">
  - <types>
    - <xsd:schema targetNamespace="urn:soap_serv">
        <xsd:import
          namespace="http://schemas.xmlsoap.org/soap/encoding/"
          />
        <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
      </xsd:schema>
    </types>
  //Τα δεδομένα που στέλνονται με το request
  - <message name="update_quantiRequest">
      <part name="product_id" type="xsd:int" />
      <part name="stock_left" type="xsd:int" />
    </message>
  //Τα δεδομένα που στέλνονται με το response
  - <message name="update_quantiResponse">
      <part name="return" type="xsd:int" />
    </message>

//Περιγράφει το web service τις λειτουργίες που επιτελεί και τα μηνύματα που
//ανταλλάσσονται
  - <portType name="soap_servPortType">
    - <operation name="update_quanti">
        <documentation>update quantity to db</documentation>
        <input message="tns:update_quantiRequest" />
        <output message="tns:update_quantiResponse" />
      </operation>
    </portType>
//Τα πρωτόκολλα για επικοινωνία που χρησιμοποιούνται από το
```

```
//web service
- <binding name="soap_servBinding" type="tns:soap_servPortType">
    <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http" />
  - <operation name="update_quanti">
      <soap:operation soapAction="urn:soap_serv#update_quanti"
        style="rpc" />
    - <input>
        <soap:body use="encoded" namespace="urn:soap_serv"
          encodingStyle="http://schemas.xmlsoap.org/soap/encodi
          ng/" />
      </input>
    - <output>
        <soap:body use="encoded" namespace="urn:soap_serv"
          encodingStyle="http://schemas.xmlsoap.org/soap/encodi
          ng/" />
      </output>
    </operation>
  </binding>
- <service name="soap_serv">
  - <port name="soap_servPort" binding="tns:soap_servBinding">
      <soap:address location="http://127.0.0.1/oscommerce-
        2.2rc2a/catalog/includes/functions/soap_serv.php" />
    </port>
  </service>
</definitions>
```

Ταυτόχρονα το NuSOAP μας δίνει και ένα πίνακα με πληροφορίες σχετικά με τις μεθόδους του Web Service τις παραμέτρους που δέχεται, το τι επιστρέφει (Σχήμα 14).



Σχήμα 14

7.1.3 Τα μηνύματα SOAP

Ακολουθούν τα μηνύματα SOAP που ανταλλάσσονται μεταξύ Client και Server σε μια υποθετική συναλλαγή όπου ένας πελάτης αγοράζει το προϊόν με κωδικό 28 και ποσότητα 1 κάτι που θέτει την διαθέσιμη ποσότητα του προϊόντος από 15 τεμ. που ήταν σε 14τεμ.:

Request

```
POST /oscommerce-
2.2rc2a/catalog/includes/functions/soap_serv.php HTTP/1.0
Host: 127.0.0.1
User-Agent: NuSOAP/0.7.3 (1.114)
Content-Type: text/xml; charset=ISO-8859-1
```

```
SOAPAction: "urn:soap_serv#update_quanti"
Content-Length: 580

<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-
ENV:Envelope SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encodi
ng/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="urn:soap_serv"><SOAP-
ENV:Body><tns:update_quanti
xmlns:tns="urn:soap_serv"><product_id
xsi:type="xsd:int">28</product_id><stock_left
xsi:type="xsd:int">14</stock_left></tns:update_quanti></S
OAP-ENV:Body></SOAP-ENV:Envelope>
```

Response

```
HTTP/1.1 200 OK
Date: Tue, 17 Feb 2009 19:30:39 GMT
Server: Apache/2.2.3 (Win32) PHP/5.2.0
X-Powered-By: PHP/5.2.0
X-SOAP-Server: NuSOAP/0.7.3 (1.114)
Content-Length: 449
Connection: close
Content-Type: text/xml; charset=ISO-8859-1

<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-
ENV:Envelope xmlns:SOAP-
```

```
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-
ENV:Body><ns1:update_quantiResponse
xmlns:ns1="urn:soap_serv"><return
xsi:type="xsd:int">14</return></ns1:update_quantiResponse
></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

7.1.4 Ο κώδικας του Client

Ο Client είναι αυτός που αιτείται την ενημέρωση της βάσης του osCommerce. Βρίσκεται μέσα στα αρχεία του webERP και συγκεκριμένα στο αρχείο ConfirmDispatch_Invoice.php. Το αρχείο αυτό έχει διαδρομή καταλόγου την : **webERP\ConfirmDispatch_Invoice.php** βρίσκεται, συνεπώς, μέσα στον κεντρικό κατάλογο του webERP. Ο client «ενεργοποιείται» μόλις γίνει η ενημέρωση της βάσης δεδομένων του webERP.

Ο κώδικας του client :

```
//Πρώτα από όλα γίνεται ένα sql ερώτημα στη βάση δεδομένων του webERP για να
//βρούμε την ποσότητα του είδους όπως αυτή διαμορφώθηκε μετά την ενημέρωση.
//Ο λόγος είναι ότι οι συναρτήσεις ενημέρωσης της βάσης του osCommerce απαιτούν
//την ποσότητα αποθέματος που προκύπτει μετά την συναλλαγή
//Αντίθετα το webERP χρησιμοποιεί  την ποσότητα μεταβολής (πόσο μεταβλήθηκε το
//απόθεμα)


$SQL="SELECT quantity
FROM locstock
WHERE locstock.stockid = '" . $OrderLine->StockID . "'
AND loccode = '" . $_SESSION['Items']->Location . """;
```

```
//Παίρνουμε τα αποτελέσματα του ερωτήματος χρησιμοποιώντας τις συναρτήσεις
//χειρισμού της βάσης δεδομένων που είναι διαθέσιμες από το σύστημα
$Result = DB_query($SQL,$db);

//Βάζω στην μεταβλητή $stock_left την  ποσότητα του είδους όπως αυτή
//διαμορφώθηκε μετά το update
if (DB_num_rows($Result)==1){
$LocQtyRow = DB_fetch_row($Result);
$stock_left = $LocQtyRow[0];
}
//Σηκώνω τον client ο οποίος κοιτά στην διεύθυνση του server
$client = new soapclient('http://127.0.0.1/oscommerce-
2.2rc2a/catalog/includes/functions/soap_serv.php?wsdl', true);

// Ελέγχω για τυχόν λάθη
$err = $client->getError();
if ($err) {
// Εμφανίζω το σφάλμα
echo '<h2>Constructor error</h2><pre>' . $err . '</pre>';
// Η κλήση έχει αποτύχει
}

// Καλώ την Soap μέθοδο και περνάω τα απαραίτητα ορίσματα (ποσότητα και κωδικό
//προϊόντος)
$result = $client->call('update_quanti',array('product_id'=>$OrderLine-
>StockID,'stock_left'=>$stock_left));

//Ελέγχω για τυχόν σφάλματα
if ($client->fault) {
echo '<h2>Fault</h2><pre>';
print_r($result);
echo '</pre>';
}
```

```
//Κοιτάω για λάθη
else {
$err = $client->getError();
if ($err) {

// Αν υπάρχουν λάθη τα δείχνω
echo '<h2>Error</h2><pre>' . $err . '</pre>';
} else {
//Αλλιώς δείχνω το αποτέλεσμα της ενημέρωσης
echo '<h2>osCommerce update information</h2><pre>';
//Αν η ποσότητα που επέστρεψε το osCommerce είναι ίδια με αυτή που έχω στη
//βάση η ενημέρωση ήταν επιτυχής
if ($result==$stock_left) {
print_r('osCommerce was updated correctly');
} else { print_r('osCommerce was not updated correctly');}    }
}
```

7.2 Επικοινωνία από osCommerce προς webERP

7.2.1 Ο κώδικας του server

Ο κώδικας του server τοποθετήθηκε σε ένα αρχείο με το όνομα CommerceToWeberpServer.php. Είναι το αρχείο που θα αναλάβει να εξυπηρετήσει τυχών αιτήσεις ενημέρωσης της βάσης δεδομένων του ERP συστήματος και βρίσκεται φυσικά στον κατάλογο του webERP με διαδρομή **webERP/ CommerceToWeberpServer.php** και είναι ο ακόλουθος :

Ο PHP κώδικας του server :

```
<?php
// Εισάγεται η βιβλιοθήκη NuSOAP στο αρχείο για μπορούν να χρησιμοποιηθούν οι
//συναρτήσεις της
require_once('lib/nusoap.php');

//Εισάγεται το αρχείο gettext.inc καθώς υπάρχουν αναφορές σε συναρτήσεις αυτού
του αρχείου
include('includes/php-gettext/gettext.inc');

// Δημιουργούμε ένα στιγμιότυπο του αντικειμένου soap_server
$server = new soap_server();

//Αρχικοποιούμε την υποστήριξη WSDL
$server->configureWSDL('CommerceToWeberpServer',
'urn:CommerceToWeberpServer');

// Εγγράφουμε την μέθοδο που θα δημοσιεύσουμε για χρήση
//Δηλώνουμε επίσης τα δεδομένα εισόδου αλλά και την έξοδο
//της μεθόδου. Βλέπουμε στην συγκεκριμένη περίπτωση η μέθοδος
//έχει όνομα  update_quanti, δέχεται 2 ορίσματα σαν είσοδο που
//είναι το ένα αλφαριθμητικού τύπου (string)
//και το άλλο ακέραιος αριθμός (int).Η έξοδος είναι τύπου ακέραιου (int)
$server->register('update_quanti',
        array('product_id' => 'xsd:string','stock_left'=>'xsd:int'),
```

```
                array('return'=>'xsd:int'),
                'urn:CommerceToWeberpServer',
                'urn:CommerceToWeberpServer#update_quanti',
                'rpc',
                'encoded',
                'update quantity to db');



// Ορίζουμε την μέθοδο σαν μια απλή συνάρτηση PHP
function update_quanti($product_id,$stock_left) {

//Επειδή η ενημέρωση γίνεται χωρίς να προηγηθεί login χρειάζεται να δηλώσουμε
//στον κώδικα το όνομα της βάσης με την οποία θα συνδεθούμε
$DatabaseName='weberp';

//Συντάσσουμε το sql ερώτημα για την ενημέρωση της βάσης
//με τις σωστές ποσότητες
 $SQL = "UPDATE locstock
            SET quantity = " . $stock_left . "
            WHERE locstock.stockid = '" . $product_id . "'
           AND locstock.loccode ='MEL'";
//Εισάγουμε το αρχείο config.php το οποίο περιέχει τα όνομα χρήστη και τον
//κωδικό για την σύνδεση στη βάση.
   include('config.php');

//Εισάγουμε το αρχείο ConnectDB.inc που περιέχει τις συναρτήσεις του συστήματος
//για την σύνδεση με τη βάση και την εκτέλεση sql ερωτημάτων
  include('includes/ConnectDB.inc');

  //Χρησιμοποιώ τη συνάρτηση DB_query του συστήματος για την εκτέλεση του
// ερωτήματος
  $Result = DB_query($SQL,$db);
//Συντάσσω ένα νέο ερώτημα για να δω την ποσότητα στην βάση μετά την
```

```
//ενημέρωση
$SQL="SELECT quantity FROM locstock WHERE  locstock.stockid= '" .
$product_id . "' AND locstock.loccode='MEL'";
$Result_query = DB_query($SQL,$db);
$Result= DB_fetch_array($Result_query);
//Η επιστροφή του web service είναι το αποτέλεσμα του ερωτήματος
return $Result['quantity'];
}
// Use the request to (try to) invoke the service
$HTTP_RAW_POST_DATA = isset($HTTP_RAW_POST_DATA) ?
$HTTP_RAW_POST_DATA : '';
$server->service($HTTP_RAW_POST_DATA);
?>
```

7.2.2 Το WSDL αρχείο

Δημιουργώντας το αρχείο CommerceToWeberp.php αυτομάτως η βιβλιοθήκη NuSOAP αναγνωρίζει ότι πρόκειται για Web Service και παράγει χωρίς καμιά περαιτέρω ενέργεια το WSDL αρχείο του Web Service το οποίο και παρατίθεται παρακάτω . Απλά με τη βοήθεια ενός Browser χρειάζεται να δείξουμε στο αρχείο που έχει τον server και θα εμφανιστεί η ακόλουθη εικόνα (Σχήμα 15)

Σχήμα 15

WSDL :
<?xml version="1.0" encoding="ISO-8859-1" ?>

```
- <definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
      ENC="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:tns="urn:CommerceToWeberpServer"
      xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
      xmlns="http://schemas.xmlsoap.org/wsdl/"
      targetNamespace="urn:CommerceToWeberpServer">
   - <types>
      - <xsd:schema targetNamespace="urn:CommerceToWeberpServer">
            <xsd:import
               namespace="http://schemas.xmlsoap.org/soap/encoding/" />
            <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
         </xsd:schema>
      </types>
   //Τα δεδομένα που στέλνονται με το request
   - <message name="update_quantiRequest">
         <part name="product_id" type="xsd:string" />
         <part name="stock_left" type="xsd:int" />
      </message>
   //Τα δεδομένα που στέλνονται με το response
     <message name="update_quantiResponse">
        <part name="return" type="xsd:int" />
      </message>

//Περιγράφει το web service τις λειτουργίες που επιτελεί και τα μηνύματα που
//ανταλλάσσονται
   - <portType name="CommerceToWeberpServerPortType">
      - <operation name="update_quanti">
            <documentation>update quantity to db</documentation>
            <input message="tns:update_quantiRequest" />
            <output message="tns:update_quantiResponse" />
         </operation>
      </portType>
//Τα πρωτόκολλα για επικοινωνία που χρησιμοποιούνται από το
//web service
```

```
- <binding name="CommerceToWeberpServerBinding"
    type="tns:CommerceToWeberpServerPortType">
    <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http" />
  - <operation name="update_quanti">
      <soap:operation
        soapAction="urn:CommerceToWeberpServer#update_quanti"
        style="rpc" />
    - <input>
        <soap:body use="encoded"
          namespace="urn:CommerceToWeberpServer"
          encodingStyle="http://schemas.xmlsoap.org/soap/encodin
          g/" />
      </input>
    - <output>
        <soap:body use="encoded"
          namespace="urn:CommerceToWeberpServer"
          encodingStyle="http://schemas.xmlsoap.org/soap/encodin
          g/" />
      </output>
    </operation>
  </binding>
- <service name="CommerceToWeberpServer">
  - <port name="CommerceToWeberpServerPort"
      binding="tns:CommerceToWeberpServerBinding">
      <soap:address
        location="http://127.0.0.1/webERP/CommerceToWeberpServer
        .php" />
    </port>
  </service>
</definitions>
```

Ταυτόχρονα το NuSOAP μας δίνει και ένα πίνακα με πληροφορίες σχετικά με τις μεθόδους του Web Service τις παραμέτρους που δέχεται, την έξοδο που παράγει. (Σχήμα 16).

Σχήμα 16

7.2.3 Τα μηνύματα SOAP

Ακολουθούν τα μηνύματα SOAP που ανταλλάσσονται μεταξύ Client και Server σε μια υποθετική συναλλαγή όπου ένας πελάτης αγοράζει το προϊόν με κωδικό 28 και ποσότητα 1 κάτι που θέτει την διαθέσιμη ποσότητα του προϊόντος από 13 τεμ. που ήταν σε 12τεμ.:

Request

```
POST /webERP/CommerceToWeberpServer.php HTTP/1.0
Host: 127.0.0.1
User-Agent: NuSOAP/0.7.3 (1.114)
Content-Type: text/xml; charset=ISO-8859-1
SOAPAction: "urn:CommerceToWeberpServer#update_quanti"
Content-Length: 609

<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-
ENV:Envelope SOAP-
```

```
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encodi
ng/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="urn:CommerceToWeberpServer"><SOAP-
ENV:Body><tns:update_quanti
xmlns:tns="urn:CommerceToWeberpServer"><product_id
xsi:type="xsd:string">28</product_id><stock_left
xsi:type="xsd:int">12</stock_left></tns:update_quanti></S
OAP-ENV:Body></SOAP-ENV:Envelope>
```

Response

```
HTTP/1.1 200 OK
Date: Tue, 17 Feb 2009 19:55:23 GMT
Server: Apache/2.2.3 (Win32) PHP/5.2.0
X-Powered-By: PHP/5.2.0
X-SOAP-Server: NuSOAP/0.7.3 (1.114)
Content-Length: 462
Connection: close
Content-Type: text/xml; charset=ISO-8859-1

<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-
ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"><SOAP-
ENV:Body><ns1:update_quantiResponse
xmlns:ns1="urn:CommerceToWeberpServer"><return
```

```
xsi:type="xsd:int">12</return></ns1:update_quantiResponse
></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

7.2.4 Ο κώδικας του Client

Ο Client είναι αυτός που αιτείται την ενημέρωση της βάσης του webERP. Βρίσκεται μέσα στα αρχεία του osCommerce και συγκεκριμένα στο αρχείο **checkout_process.php**. Το αρχείο αυτό έχει διαδρομή καταλόγου την : **oscommerce-2.2rc2a\catalog\ checkout_process.php**. Ο Client «ενεργοποιείται» μόλις γίνει η ενημέρωση της βάσης δεδομένων του osCommerce.

Ο κώδικας του client :

```
//Ορίζω ένα flag που σηματοδοτεί την έξοδο από το loop. Γίνεται αληθές μόνο όταν
//η ποσότητα που επιστρέφει το webERP είναι ίδια με την ποσότητα στη βάση  του
//osCommerce
$exodos = false;
 while ($exodos == false) {
//Σηκώνω τον client ο οποίος κοιτά στην διεύθυνση του server
$client = new
soapclient('http://127.0.0.1/webERP/CommerceToWeberpServer.php?wsdl', true);
          // Check for an error
   $err = $client->getError();
// Ελέγχω για τυχόν λάθη
   if ($err) {
   // Εμφανίζω το σφάλμα
   echo '<h2>Constructor error</h2><pre>' . $err . '</pre>';
   // Η κλήση έχει αποτύχει
    }

 //κάνω casting στη μεταβλητή $product_id και την μετατρέπω από integer που είναι
//σε string.Ο λόγος είναι ότι στη βάση δεδομένων του webERP ο κωδικός είδους
//αποθηκεύεται σε τύπο δεδομένων string ενώ στο osCommerce σε integer
   $product_id = (string)(tep_get_prid($order->products[$i]['id']));
```

```
 // Καλώ την Soap μέθοδο και περνάω τα απαραίτητα ορίσματα (ποσότητα και
//κωδικό προϊόντος)
          $result = $client-
>call('update_quanti',array('product_id'=>$product_id,'stock_left'=>$stock_left));
//Ελέγχω για τυχόν σφάλματα
   if ($client->fault) {
   echo '<h2>Fault</h2><pre>';
   print_r($result);
   echo '</pre>';
} else {
//Κοιτάω για λάθη
   $err = $client->getError();
   if ($err) {
// Αν υπάρχουν λάθη τα δείχνω
       echo '<h2>Error</h2><pre>' . $err . '</pre>';
   } else {
         if ($result==$stock_left) {
          //Η ενημέρωση ήταν επιτυχής μπορείς να βγείς από το loop
                    $exodos=true;    }
}
}//τέλος while
```

8.ΠΑΡΑΡΤΗΜΑ Β

8.1 Εγκατάσταση του osCommerce

Ένα από τα πιο σημαντικά χαρακτηριστικά του osCommerce είναι η δυνατότητα που παρέχει στον ιδιοκτήτη του καταστήματος να στήσει, να τρέξει και να διαχειριστεί τα Online καταστήματά του με ελάχιστη προσπάθεια και χωρίς κάποιο επιπλέον κόστος. Το osCommerce συνδυάζει λύσεις ανοιχτού κώδικα για να παρέχει μια δωρεάν και ανοιχτή πλατφόρμα ανάπτυξης που περιλαμβάνει σαν γλώσσα ανάπτυξης την PHP τον σταθερό Apache web server και τον ταχύτατο MySQL Database Server. Χωρίς κάποιον επιπλέον περιορισμό, το osCommerce μπορεί να εγκατασταθεί σε οποιονδήποτε web server που υποστηρίζει την PHP3 ή PHP4 και σε οποιοδήποτε περιβάλλον που υποστηρίζει η PHP – MySQL που περιλαμβάνει λειτουργικά όπως Linux, Solaris, BSD και Microsoft Windows.

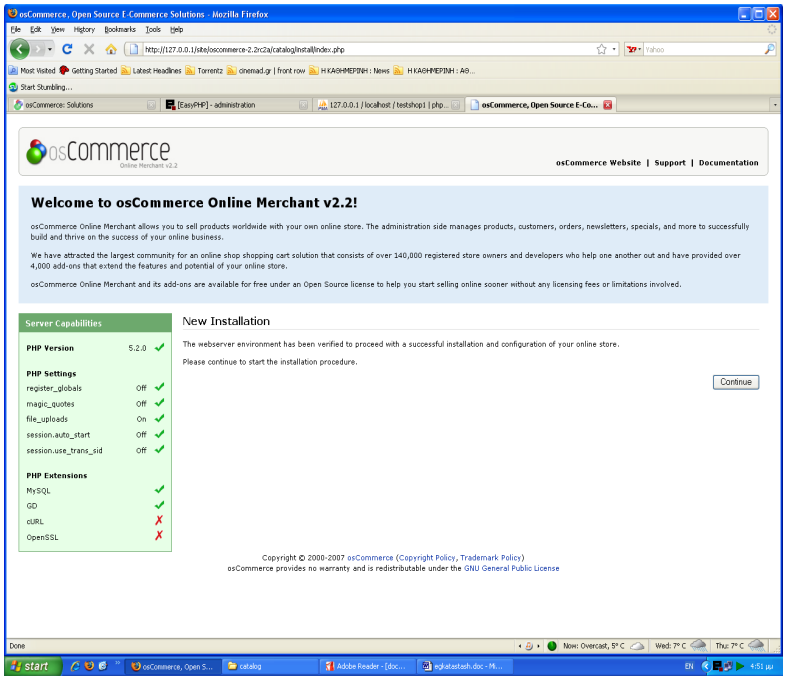
Πρώτα από όλα ανεβάζουμε τον κατάλογο catalog στον server χρησιμοποιώντας ένα πρόγραμμα FTP. Η δομή του osCommerce είναι :

- Catalog
- Catalog/admin
- Extras

Ανεβάζουμε τον φάκελο Catalog, ο οποίος περιλαμβάνει και τον φάκελο Admin στον κατάλογο δημοσίευσης του Server (public_html,www,htdocs).

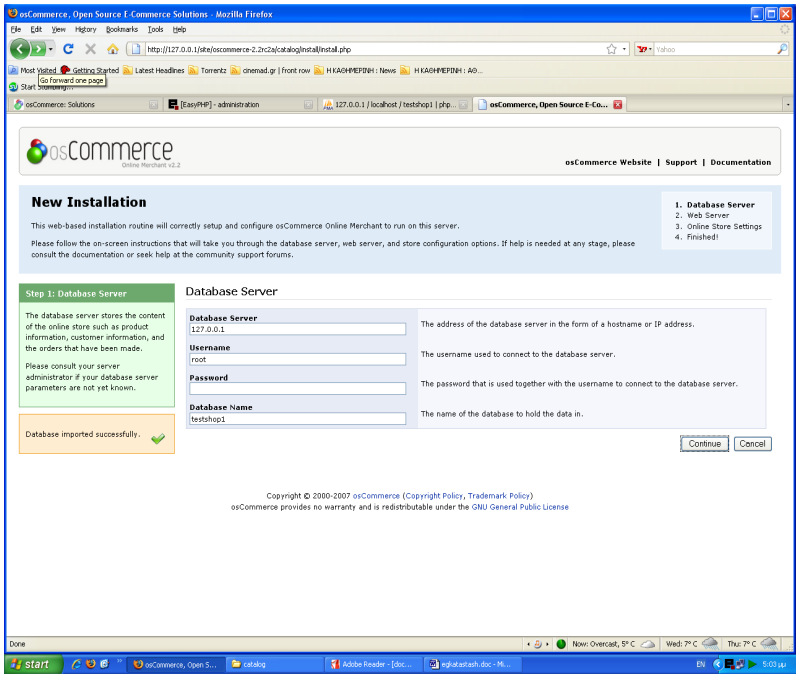
Χρησιμοποιώντας το phpMyAdmin ή κάποιο άλλο αντίστοιχο εργαλείο δημιουργούμε την βάση και τον χρήστη και αναθέτουμε τον χρήστη αυτό στην βάση. Καταγράφουμε το όνομα της βάσης, το username και τον κωδικό.

Ανοίγουμε έναν Browser και πληκτρολογούμε την διεύθυνση http://τοΌνομαΤουServer/catalog/install και πατάμε το κουμπί **Continue**.

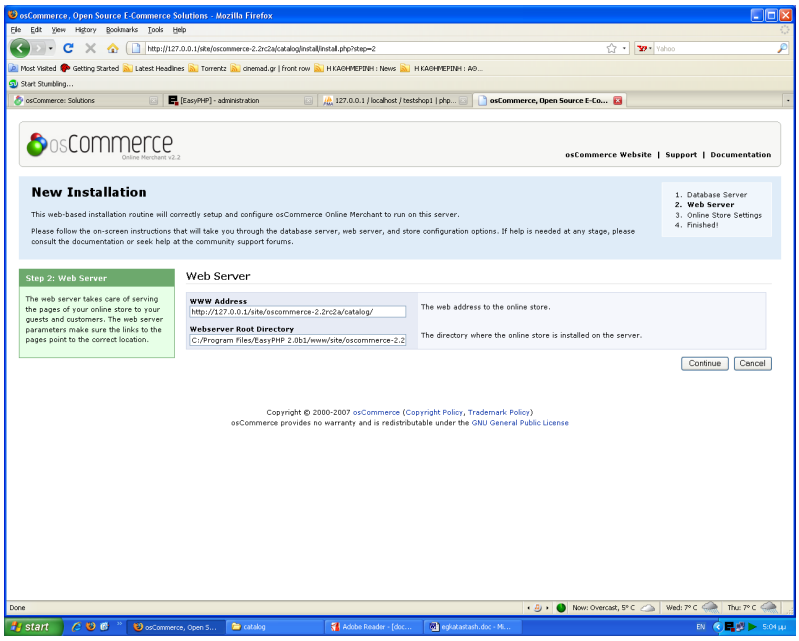


1. Ορίζουμε τον Database Server είτε με την μορφή Hostname είτε με την μορφή IP διεύθυνσης.
2. Βάζουμε το Username που ορίσαμε για τη σύνδεση στη βάση
3. Βάζουμε το Password που ορίσαμε για τη σύνδεση στη βάση
4. Βάζουμε το όνομα της βάσης που δημιουργήσαμε.

Πατάμε το κουμπί **Continue** οπότε και γίνεται η εισαγωγή της βάσης με τα ορίσματα που δώσαμε.



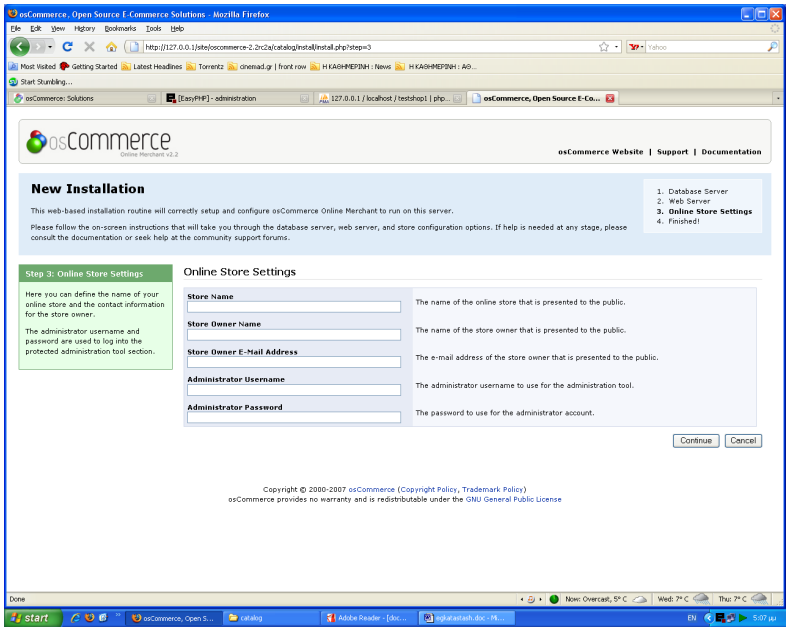
Στη συνέχεια παρέχονται πληροφορίες για την Web Address του καταστήματος και το Directory που θα εγκατασταθεί το κατάστημα στον server. Πατάμε **Continue**.



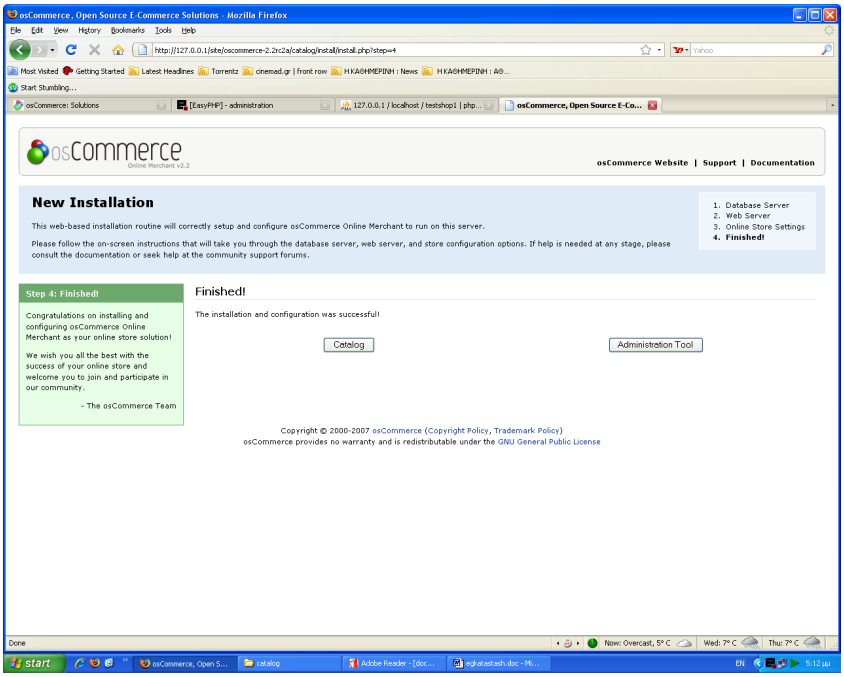
Στην επόμενη οθόνη ορίζουμε τις εξής πληροφορίες :

1. Το όνομα του καταστήματος
2. Το όνομα του κατόχου του καταστήματος που θα φαίνεται στο κοινό
3. Το Email επικοινωνίας που θα φαίνεται στο κοινό
4. Το Username του Administrator για τη χρήση του Administrator Account
5. Το Password του Administrator για τη χρήση του Administrator Account

Πατάμε το **Continue**



Η εγκατάσταση ολοκληρώθηκε



8.2 Εγκατάσταση του webERP

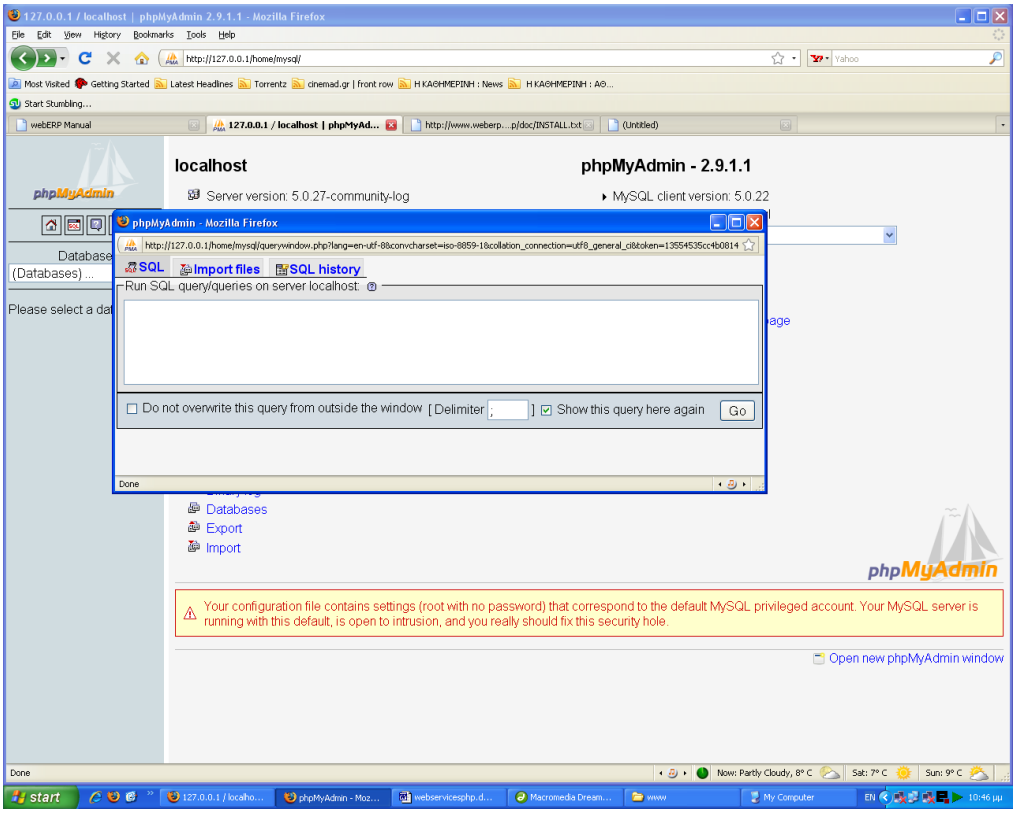
Το webERP είναι ένα ERP σύστημα οποίο βασίζεται στην γλώσσα PHP (έκδοση 4.2 και μετά) και στον MySQL Database Server. Σαν Web Server προτείνεται ο Apache αλλά ουσιαστικά λειτουργεί με οποιονδήποτε Web Server υποστηρίζεται από την PHP.

Πρώτα από όλα χρειάζεται να μεταφέρουμε όλα τα Scripts σε έναν κατάλογο στο Root Directory του Web Server.

Το επόμενο βήμα είναι να δημιουργήσουμε τη βάση δεδομένων. Παρέχονται δύο Scripts(βρίσκονται στον κατάλογο webERP/sql/mysql) :

1. weberp-demo.sql το οποίο περιέχει κάποια ελάχιστα δεδομένα και μια «στημένη» εταιρεία για τον έλεγχο λειτουργίας του συστήματος.
2. weberp-new.sql το οποίο περιέχει τις βασικές πληροφορίες για το «στήσιμο» μιας νέας εταιρείας. Αυτό το Script δημιουργεί μια webERP βάση δεδομένων με μόνο τα αρχικά δεδομένα.

Μέσα από το εργαλείο PhpMyAdmin εγκαθιστούμε το Script που επιθυμούμε και δημιουργούμε την βάση.



Η εγκατάσταση ολοκληρώθηκε. Πληκτρολογώντας την διεύθυνση http://τοΟνομαΤουServer/weberp εμφανίζεται η σελίδα εισαγωγής στο σύστημα.

9. ΒΙΒΛΙΟΓΡΑΦΙΑ

[1]. http://www.igi-pub.com/downloads/excerpts/193070836XExcerp.pdf

[2]. http://www.weberp.org

[3]. http://www.it.uom.gr/project/soap/Theory/introduction.html

[4]. http://www.itwizard.info/technology/webservice/webservices.pdf

[5]. http://www.webservices.org

[6]. http://www.scottnichol.com/nusoapprogwsdl.htm

[7]. http://www.w3.org/TR/REC-xml

[8]. http://www.w3.org/TR/SOAP

[9]. http://www.w3.org/TR/WSDL

[10]. http://www.service-architecture.com/

[11]. http://gdp.globus.org/gt4-tutorial/multiplehtml/ch01s02.html

[12]. http://msdn.microsoft.com/en-us/library/ms996507.aspx

[13]. http://static.userland.com/xmlRpcCom/soap/SOAPv11.htm#_Toc478383494

[14]. www.oscommerce.com

[15]. Luke Welling, Laura Thomson (2005). *Ανάπτυξη Web εφαρμογών με PHP και MySQL*

[16]. Bingi, P., Sharma, M. K. and Godla, J. K. (1999). *Critical issues affecting an ERP implementation*. *Information Systems Management*.

[17]. Al-Mudimigh A., Zairi M., Al-Mashari. M. European Journal of Information Systems, Volume 10, Number 4, 1 April 2001. *ERP software implementation: an integrative framework*

[16]. Al-Mashari, M. (2000). *Constructs of Process Change Management in ERP Context: A Focus on SAP R/3.*

[17]. Holland, C. P., Light, B., & Gibson, N. (1999a). *A critical success factors model for enterprise resource planning implementation*

[18]. Holland, C., Light, B., & Kawalek, P. (1999b). *Beyond enterprise resource planning projects*

[19]. Maher, J. (1999). *ERP in industry: Automate and integrate*

[20]. Brown, C., Vessey, I., & Powell, A. (2000). *The ERP purchase decision: Influential business and IT factors.*

[21]. Shakir, M. (2000). *Decision making in evaluation, selection and implementation of ERP systems*