



## Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών  
«Προηγμένα Συστήματα Πληροφορικής»

### Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	<b>Διαδικτυακός Γεννήτορας επεξεργασίας Πολλαπλών ERP Συστημάτων</b>
Όνοματεπώνυμο Φοιτητή	<b>Ζάρας Αχιλλέας του Παναγιώτη</b>
Αριθμός Μητρώου	<b>Π/08062</b>
Κατεύθυνση	<b>Συστήματα Υποστήριξης Αποφάσεων</b>
Επιβλέπων	<b>Ι.Χ. Παναγιωτόπουλος, Καθηγητής</b>

Πανεπιστήμιο Πειραιώς - Τμήμα Πληροφορικής  
Πρόγραμμα Μεταπτυχιακών Σπουδών στο  
Προηγμένα Συστήματα Πληροφορικής

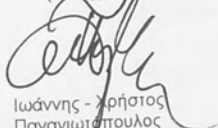
10/02/2010

**Φεβρουάριος 2010**

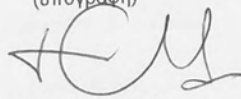
Πανεπιστήμιο Πειραιώς

**Τριμελής Εξεταστική Επιτροπή**

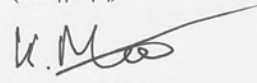
(υπογραφή)

  
Ιωάννης - Χρήστος  
Παναγιωτόπουλος  
Καθηγητής

(υπογραφή)

  
Ιωάννης Θεοδορίδης  
Αναπληρωτής Καθηγητής

(υπογραφή)

  
Κωνσταντίνος  
Μεταξιώτης  
Επίκουρος Καθηγητής

## Περιεχόμενα

Περιλήψη .....	5
1. Εισαγωγή.....	5
1.1 Η πορεία της ενσωμάτωσης (integration) στα ERP συστήματα μέχρι σήμερα και ο στόχος της παρούσας εργασίας.....	6
2. ERP Συστήματα (Enterprise Resource Planning Systems) .....	7
2.1 Τι είναι τα ERP συστήματα?.....	7
2.2 Συνοπτική περιγραφή των συστημάτων ERP (Enterprise Resource Planning) 8	
2.3 Η δομή και οι λειτουργίες ενός Συστήματος ERP.....	9
2.3.1 Εμπορικές Συναλλαγές.....	9
2.3.2 Χρηματοοικονομική διαχείριση (Accounting and Finance).....	13
2.3.3 Λογιστικές ενότητες (Accounting and Finance) .....	16
2.3.4 Διαχείριση Παραγωγικής Διαδικασίας/Αποθήκης.....	17
2.3.5 Διαχείριση Ανθρώπινου Δυναμικού (Human Resources).....	22
2.3.5 CRM (Customer Relationship Management) .....	24
2.3.6 Αναφορές και παραμετροποίηση (Reports & Customization).....	25
3. Web Services .....	26
3.1 Εισαγωγή .....	26
3.2. Σύντομη περιγραφή των κύριων τεχνολογιών των υπηρεσιών ιστού . 27	
3.3 Σύγχρονες Τάσεις.....	27
3.4 Extensible Markup Language (XML) .....	28
3.5 Simple Object Access Protocol (SOAP) .....	29
3.6 Web Services Definition Language (WSDL).....	29

3.7 Universal Description, Discovery and Integration (UDDI) .....	30
4. Ανάπτυξη του γεννήτορα .....	31
4.1 Εισαγωγή – Στόχος, δομή του συστήματος .....	31
4.2 Ανάλυση μιας βάσης δεδομένων ERP συστήματος .....	34
4.3 Διαδικτυακός Γεννήτορας Πολλαπλών ERP.....	41
4.3.1 Λειτουργίες της Υπηρεσίας Ιστού .....	41
4.3.2..... Κώδικας της Υπηρεσίας Ιστού	60
4.3.2.....Το αρχείο περιγραφής της Υπηρεσίας Ιστού (WSDL)	72
4.4 Διαδικτυακή εφαρμογή, που καλεί την Υπηρεσία Ιστού.....	85
Συμπεράσματα .....	106
Βιβλιογραφία .....	106

## Περίληψη

Σήμερα, σε αντίθεση με την προηγούμενη δεκαετία, το μεγαλύτερο ποσοστό των επιχειρήσεων στις ανεπτυγμένες χώρες χρησιμοποιούν κάποιο πληροφοριακό σύστημα. Πλέον, μία από τις καινούργιες πληροφοριακές απαιτήσεις των εταιριών, είναι η συνεργασία του ήδη εγκατεστημένου πληροφοριακού τους συστήματος, είτε με κάποιο νέο που θα καλύψει διαφορετικό φάσμα λειτουργιών απ' το ήδη εγκατεστημένο, είτε μ' ένα πληροφοριακό σύστημα κάποιας τρίτης συνεργαζόμενης εταιρίας, με σκοπό την αυτοματοποίηση των διαδικασιών που λαμβάνουν χώρα μεταξύ τους.

Η παρούσα εργασία, μελετά το θέμα της συνεργασίας μεταξύ δύο ή περισσότερων λογισμικών, ανεπτυγμένα από διαφορετικούς κατασκευαστές. Η κατηγορία των πληροφοριακών συστημάτων που μελετάται - αφού πρώτα αναλύεται - είναι αυτή των ERP (Enterprise Resource Planning systems), μια κατηγορία που κατέχει σημαντικό μερίδιο στην αγορά λογισμικού. Η τεχνολογία που χρησιμοποιείται για την επικοινωνία μεταξύ διαφορετικών ERP συστημάτων είναι οι Υπηρεσίες Ιστού (web services), που αποτελεί παγκόσμιο standard και είναι η μόνη αξιόπιστη λύση στο πρόβλημα της χρήσης διαφορετικών τεχνολογιών ανάπτυξης.

## Abstract

Nowadays, in contrast with the past decades, most of the companies around the world use at least one Information System. Henceforth, one of the new upcoming needs concerning their IT departments is the integration between a new Information System with a legacy system (that is already running in the company) or with a Partner's Information System (in order to accelerate the procedures that take place between these two or more partners).

This study examines the issue of integration between two or more Information Systems developed by different software houses. The main software category that is examined is the one of Enterprise Resource Planning systems (a category that holds a huge portion in software market). Web Services are used in order to make different ERP integrate with each other – the most efficient and reliable solution for such problems, which is also a worldwide standard.

## Εισαγωγή

Τη σημερινή εποχή, έχοντας ξεπεράσει χρονικά την έκρηξη της ανάπτυξης του κλάδου της πληροφορικής - που έλαβε χώρα από τέλη της δεκαετίας του 90' μέχρι και τις αρχές της δεκαετίας που διανύουμε - παρατηρούμε μερικές αλλαγές στους στόχους και τον προσανατολισμό της αγοράς που σχετίζεται με την επιστήμη των υπολογιστών.

Στο τομέα του υλικού (hardware), βλέπουμε να κλονίζεται ο νόμος του Moore (η χωρητικότητα των επεξεργαστών σε transistors - συνελπώς και η ταχύτητά τους - διπλασιάζεται κάθε 18 μήνες). Παρατηρεί κανείς, πως οι συχνότητες τους τα τελευταία τρία χρόνια κυμαίνονται σε παρόμοια επίπεδα και το μόνο που αλλάζει είναι ο αριθμός των πυρήνων τους και οι ταχύτητες των διαύλων επικοινωνίας (bus speed) με τα υπόλοιπα μέρη του υπολογιστή.

Στον τομέα του λογισμικού (στον οποίο και επικεντρώνεται η παρούσα εργασία) μπορούμε να κάνουμε διάφορες παραδοχές. Μία από αυτές, είναι πως σε πολλές κατηγορίες έχει επέλθει κορεσμός, λόγω της ύπαρξης πολλών εταιριών παραγωγής προϊόντων λογισμικού. Η κατηγορία των συστημάτων, η οποία απασχόλησε την παρούσα εργασία είναι αυτή των E.R.P. (Enterprise Resource Planning, Συστήματα Διαχείρισης Εταιρικών Πόρων), η οποία αναλύεται λεπτομερώς στην επόμενη ενότητα. Μπορούμε να πούμε πως μόνο στην ελληνική αγορά, κυκλοφορούν πάνω από 50 τέτοια προϊόντα λογισμικού, αριθμός υπερβολικός αν λάβουμε υπ' όψη την πολυπλοκότητα (τεκμηριώνεται σε επόμενες ενότητες) των συστημάτων αυτών.

Μία ακόμα παραδοχή είναι πως εκτός από πολλές εταιρίες παραγωγής λογισμικού - συμπεριλαμβανομένης και της κατηγορίας των ERP - υφίστανται σήμερα και πολλές πλατφόρμες ανάπτυξης λογισμικού (Java, .Net, PHP, Delphi). Κάθε εταιρία, χρησιμοποιεί την πλατφόρμα από την οποία θα αποκομίσει τα μεγαλύτερα οφέλη (π.χ. είναι δωρεάν; οι υπάρχοντες προγραμματιστές τη γνωρίζουν;). Αυτό το γεγονός, μπορεί στις περισσότερες των περιπτώσεων να μην επηρεάζει τα οφέλη μιας επιχείρησης από την ανάπτυξη, προώθηση και μαζική πώληση ενός προϊόντος ERP για παράδειγμα, αλλά τίθεται επί τάπητος όταν έρθει η ώρα της υλοποίησης ενός συστήματος βασισμένου σε πολλαπλά προγράμματα (software), τα οποία είναι ανεπτυγμένα πάνω σε διαφορετικές πλατφόρμες. Ένα κλασικό παράδειγμα, είναι η 'συνεννόηση' του ERP μιας ανεπτυγμένης με το ERP του προμηθευτή της, ώστε να υπάρχει δυνατότητα αυτόματης δημιουργίας παραγγελιών προς τον προμηθευτή, σε προϊόντα όπου το stock της πρώτης έχει πέσει κάτω από κάποια συγκεκριμένα επίπεδα. Όταν οι δύο αυτές εταιρίες διαθέτουν το ίδιο ERP σύστημα, η διαδικασία είναι σχετικά απλή (π.χ. το SAP υποστηρίζει τέτοιου είδους λειτουργίες). Όταν όμως, λόγω της ύπαρξης αναρίθμητων ERP στο εμπόριο, διαθέτουν συστήματα διαφορετικών κατασκευαστών, η αυτοματοποίηση αυτή δυσκολεύει κατά πολύ.

Υπάρχει λοιπόν μία τάση, ως συνέπεια των παραπάνω δύο φαινομένων (πολλά λογισμικά, πολλές πλατφόρμες ανάπτυξης) να αναλώνεται πολύς από τον χρόνο των προγραμματιστών, όχι προς την κατεύθυνση της ανάπτυξης νέου λογισμικού, αλλά προς την κατασκευή ενδιάμεσων λογισμικών (middleware), τα οποία έχουν ως στόχο την 'γεφύρωση' δύο ή περισσότερων software μεταξύ τους, πάντα πάνω σε συγκεκριμένες από τις λειτουργίες τους.

Στην παρούσα εργασία, αναλύεται πλήρως μια ανάλογη περίπτωση πάνω στην κατηγορία λογισμικού των ERP. Αρχικά, παρουσιάζονται πλήρως οι λειτουργίες αυτών των συστημάτων, η δομή τους, οι στόχοι τους και τα οφέλη που μπορούν να αποκομίσουν οι εταιρίες από τη χρήση τους. Στη συνέχεια παρουσιάζεται η πιο συνήθης τεχνολογία, η οποία χρησιμοποιείται για την ολοκλήρωση (integration) δύο διαφορετικών συστημάτων λογισμικού. Αυτή η τεχνολογία, είναι οι Υπηρεσίες Ιστού (Web Services). Στη συνέχεια αναπτύσσεται ένα σύστημα, το οποίο έχει ως στόχο την διαλειτουργικότητα μεταξύ διαφορετικών ERP Συστημάτων.

1.1 Η πορεία της ενσωμάτωσης (integration) στα ERP συστήματα μέχρι σήμερα και ο στόχος της παρούσας εργασίας.

Το ζήτημα της συνεργασίας ενός ERP με κάποιο άλλο σύστημα (το οποίο μπορεί να είναι και αυτό ERP), υφίσταται από τότε που υπάρχει και η συγκεκριμένη κατηγορία λογισμικού. Επειδή πολύ συχνά, το ERP δεν ήταν το πρώτο Πληροφοριακό Σύστημα που έκανε την εμφάνισή του σε πολλές επιχειρήσεις, γεννήθηκε η ανάγκη συνεργασίας του ('καινούργιου') ERP με τα παλαιότερα Πληροφοριακά Συστήματα που ήδη

λειτουργούσαν, ούτως ώστε να μην χαθεί η αρχική επένδυση της εταιρίας. Οι πρώτες, λοιπόν, προσπάθειες προς στην κατεύθυνση της ενσωμάτωσης του ERP με άλλα συστήματα είχαν αυτόν τον προσανατολισμό.

Οι περιπτώσεις διαλειτουργικότητας είναι πολλές. Ξεκινώντας από το χαμηλότερο επίπεδο (low level), οι σύγχρονες σουίτες ERP παρέχουν κάποια υπο-εφαρμογή (low level API's) η οποία βοηθάει την διαδικασία μεταφοράς δεδομένων από και προς το πρόγραμμα. Αυτές οι υπο-εφαρμογές, όμως, λειτουργούν στο χαμηλότερο επίπεδο, αυτό των δεδομένων, και δεν ενσωματώνουν καμία έννοια επιχειρηματικής λογικής (business logic) και διαδικασιών (business processes).

Ένα επίπεδο πιο μακριά μπορούμε να πούμε πως έχει φτάσει το – κορυφαίο σε πωλήσεις σήμερα – SAP. Το SAP παρέχει τα λεγόμενα 'BAPIS' (Business Application Interfaces), τα οποία αποτελούν διεπαφές (interfaces) μέσω των οποίων εξωτερικές εφαρμογές μπορούν να έχουν πρόσβαση σε λειτουργίες του SAP. Τα BAPIS, ενσωματώνουν μεν επιχειρηματική λογική, αλλά περιορίζουν τις εξωτερικές εφαρμογές σε μια μικρή γκάμα λειτουργιών, αναγκάζοντάς τες να καταφύγουν στις εξ' αρχής (custom) υλοποιήσεις. Μια ανάλογη τεχνολογία που χρησιμοποιεί το SAP είναι τα IDocs. Τα τελευταία, αποτελούν μηχανισμούς που επιτρέπουν την απομακρυσμένη κλήση διαδικασιών από τρίτες εφαρμογές, καθώς και την μεταφορά δεδομένων από και προς το SAP.

Πέρα από τις ενσωματωμένες εφαρμογές που παρέχουν οι κατασκευαστές των ERP, υπάρχουν στην αγορά και τα πακέτα EAI (Enterprise Application Integration). Τα πακέτα αυτά, είναι έτοιμες υλοποιήσεις – οι οποίες δεν χρειάζονται την παρέμβαση κάποιου προγραμματιστή όπως στα BAPIS – που βοηθούν στην κοινή χρήση δεδομένων μεταξύ συγκεκριμένων ERP. Παραδείγματα τέτοιων προϊόντων είναι το DataMirror Corp.'s Transformation Server, το οποίο μπορεί να πραγματοποιήσει εξαγωγή και διαμοιρασμό δεδομένων μεταξύ συγκεκριμένων ERP συστημάτων και το OnDisplay's CenterStage reIntegrate της Vignette Corporation το οποίο διασυνδέει ERP συστήματα με άλλες εταιρικές εφαρμογές (οι οποίες μπορεί να είναι και διαδικτυακές).

Το πιο πολύπλοκο κομμάτι στον τομέα της ενσωμάτωσης (integration) στα ERP είναι η ταυτόχρονη και αρμονική διαλειτουργικότητα μεταξύ πολλών διαφορετικών τέτοιων προϊόντων (έχει θεθεί ως σχετικός όρος η έννοια transactional integration). Υπάρχουν σήμερα εταιρίες που ασχολούνται με αυτό το κομμάτι (όπως η BEA Systems και η webMethods), οι οποίες όμως πραγματοποιούν ad-hoc έργα, δηλαδή την δημιουργία 'γεφυρών' μεταξύ ERP συστημάτων κατ' απαίτηση των πελατών τους και πάνω σε συγκεκριμένες ενότητες.

Η παρούσα εργασία, μελετά τον τρόπο με τον οποίο θα μπορούσε να δημιουργηθεί ένα ενιαίο πλαίσιο (ή σύστημα) συνεργασίας όλων των ERP. Γύρω από το πλαίσιο αυτό, θα πρέπει να υπάρξουν κοινές θεωρήσεις (ίσως και μέσω συνεννοήσεων με τους κατασκευαστές) για πολλές εκ των εννοιών που διαχειρίζεται ένα ERP σύστημα (π.χ. με ποιο τρόπο υπολογίζονται τα κέρδη?). Εφόσον το πλαίσιο αυτό δημιουργηθεί, (ένα παράδειγμα πραγματοποιείται στην εργασία αυτή, με την δημιουργία του Web Service στο κεφάλαιο 4.3) στη συνέχεια θα πρέπει όλοι οι κατασκευαστές να προσαρμοστούν στο πλαίσιο αυτό και να δώσουν την τεχνονομία τους, γύρω από τον τρόπο με τον οποίο κάθε λειτουργία πραγματοποιείται στο δικό τους ERP. Στη συνέχεια, στόχος είναι να δημιουργηθεί ένα και μόνο ενιαίο περιβάλλον, το οποίο βρίσκεται ένα επίπεδο πάνω από το προαναφερθέν σύστημα και θα είναι το μοναδικό με το οποίο θα έρχεται σε επαφή ο χρήστης. Μέσω του περιβάλλοντος αυτού, θα μπορεί κάποιος να πραγματοποιεί τις διαδικασίες που επιθυμεί, χωρίς να απασχολείται με το ποια εφαρμογή ERP θα τις εκτελέσει στο παρασκήνιο. Στην ενότητα 4.4 δημιουργείται μία τέτοια διαδικτυακή εφαρμογή η οποία αποτυπώνει τη συγκεκριμένη θεώρηση.

## 2. ERP Συστήματα (Enterprise Resource Planning Systems)

Στο κεφάλαιο αυτό θα παρουσιαστούν όλα τα χαρακτηριστικά των συστημάτων ERP. Αρχικά θα γίνει μια σύντομη επεξήγηση του τι είναι ένα ERP σύστημα. Ακολούθως, θα αναλυθούν πλήρως οι λειτουργίες και ο ρόλος ενός τέτοιου πληροφοριακού συστήματος σε μια επιχείρηση.

### 2.1 Τι είναι τα ERP συστήματα?

Τα ERP συστήματα αποτελούν εργαλεία λογισμικού τα οποία χρησιμοποιούνται για την διαχείριση επιχειρηματικών δεδομένων. Τα ERP συστήματα βοηθούν τις επιχειρήσεις να μηχανογραφίσουν διαδικασίες όπως η διαχείριση της αποθήκης τους, οι εμπορικές συναλλαγές, η χρηματοοικονομική διαχείριση, η διαχείριση ανθρώπινων πόρων και πολλές ακόμα εταιρικές λειτουργίες.

Σύμφωνα με την Deloitte Consulting (μεγάλη συμβουλευτική εταιρία), ένα ERP σύστημα είναι ένα πακέτο επιχειρηματικού λογισμικού που επιτρέπει στην εταιρία να 'αυτοματοποιήσει και να ενσωματώσει όλες τις επιχειρηματικές διαδικασίες. Να διαμοιράσει κοινής μορφής δεδομένα και πρακτικές σε όλα τα τμήματα μιας επιχείρησης. Επίσης, να παράγει και να έχει πρόσβαση σε πληροφορίες πραγματικού χρόνου'. Ένα ERP σύστημα, συνήθως διαθέτει μία ενιαία βάση δεδομένων στην οποία όλες οι επιχειρηματικές διαδικασίες καταχωρούνται, επεξεργάζονται, παρακολουθούνται και εξάγονται σε αναφορές.

## 2.2 Συνοπτική περιγραφή των συστημάτων ERP (Enterprise Resource Planning)

Τα συστήματα ERP μπορούν να υποστηρίξουν και να απεικονίσουν τις λειτουργίες οποιουδήποτε είδους επιχείρησης. Ο λόγος, φυσικά, για τον οποίο ξεκίνησε η ανάπτυξη τέτοιων συστημάτων είναι ο – όσο το δυνατόν περισσότερο – ακριβής προγραμματισμός των πόρων σε μεγάλες βιομηχανίες και εργοστάσια. Η επιχειρηματική επιτυχία στο συγκεκριμένο είδος επιχειρήσεων, εξαρτάται άμεσα από τον προγραμματισμό των πόρων, (είτε αυτός αφορά πρώτες ύλες, ανθρώπινο δυναμικό, πάγια, μηχανήματα), καθώς θα πρέπει να υπάρχει η σωστή διαθεσιμότητα σε μια δεδομένη χρονική στιγμή – αυτή μπορεί να είναι είτε αυξημένη είτε μειωμένη, ανάλογα με τα ερεθίσματα που λαμβάνει η επιχείρηση από το εξωτερικό της περιβάλλον. Υπάρχει λοιπόν, μεγάλη ανάγκη σωστής απεικόνισης της επιχείρησης τόσο στο κομμάτι των υπαρχόντων πόρων, όσο και των μελλοντικών αναγκών σε αυτούς, που θα προκύψουν με βάση τον ρυθμό της παραγωγής και των πωλήσεων. Συμπεραίνουμε, λοιπόν, πως οι μεγάλες βιομηχανίες με τις αυξημένες πληροφοριακές ανάγκες τους, σήμαναν την αρχή της ενοποίησης όλων των επιμέρους λογισμικών μιας επιχείρησης, σε ένα ενιαίο λογισμικό, με κοινή διεπαφή (interface) και αρχιτεκτονική, στο οποίο καταγράφεται οτιδήποτε εισάγεται, μεταποιείται και εξάγεται από αυτήν. Οι λειτουργίες που υποστηρίζονται από ένα τέτοιο σύστημα είναι ποικίλες. Παρακάτω, ακολουθεί μία σύνοψη των δυνατοτήτων ενός ERP, μέσω της οποίας γίνεται ακόμα πιο σαφές ότι η ποικιλία των μηχανισμών και λειτουργιών μιας επιχείρησης, συμβαδίζει με την γκάμα των δυνατοτήτων ενός συστήματος ERP.

Σαν μια πρώτη, λοιπόν, προφανή λειτουργία ενός τέτοιου συστήματος μπορούμε να θεωρήσουμε την αποθήκευση όλων των οικονομικών συναλλαγών (εισπράξεις μετρητών, επιταγών, πληρωμές προμηθευτών, καταθέσεις σε λογαριασμούς). Ο επιμέρους στόχος αυτής της καθημερινής διαδικασίας, είναι η εξαγωγή – ανά πάσα στιγμή που είναι αυτό επιθυμητό – μιας πλήρους οικονομικής εικόνας της επιχείρησης προς τα υψηλόβαθμα στελέχη. Επίσης, η καλύτερη παρακολούθηση του συνόλου των συναλλαγών με τους πελάτες και με τους προμηθευτές της, κάνει μια εταιρία ταχύτερη σε απόκριση και αποδοτικότητα όσο αναφορά το εξωτερικό της περιβάλλον. Το ERP αποθηκεύει κάθε συναλλαγή με τους προμηθευτές, κάνοντας τις πληρωμές τους πιο γρήγορες. Επίσης, βελτιώνονται οι σχέσεις με τους πελάτες της επιχείρησης. Η επιχείρηση αποκτά καλύτερη εικόνα γύρω από τη δυναμική του κάθε πελάτη και μπορεί να ξεχωρίσει και να διαχειριστεί με βέλτιστο τρόπο τους καλύτερους εξ' αυτών.

Επίσης, συνυφασμένη με τη προηγούμενη λειτουργία, είναι η διαχείριση των εμπορικών συναλλαγών της επιχείρησης. Σε ένα τέτοιο σύστημα μπορούν να διατηρούνται όλες οι παραγγελίες, οι αγορές, οι πωλήσεις που θα βοηθήσουν στην δημιουργία συνολικών στατιστικών στοιχείων για την επιχείρηση. Επίσης, είναι δυνατή η δυναμική κατάρτιση τιμοκαταλόγων (price lists) ανάλογα με την χρονική περίοδο και σε συντονισμό με το τμήμα marketing.

Μια τρίτη επιμέρους λειτουργία ενός συστήματος ERP είναι η καταγραφή των υλικών (ή των υπηρεσιών) που αγοράζει ή πουλάει η επιχείρηση. Ο περιορισμός του stock, είναι στατιστικά το μεγαλύτερο όφελος που αποκομίζουν οι επιχειρήσεις από την αγορά και την υλοποίηση ενός ERP συστήματος.

Απαρτία – ειδικά για το περίπλοκο ελληνικό φορολογικό σύστημα – είναι η διαχείριση των λογιστικών στοιχείων την επιχείρησης. Οι ελληνικές εταιρίες έχουν αυξημένες υποχρεώσεις προς την εφορία και τα συστήματα ERP μπορούν να ενημερώνουν αυτόματα τα λογιστικά δεδομένα της επιχείρησης κατά την διάρκεια των καθημερινών συναλλαγών (π.χ. έκδοση αποδείξεων λιανικής). Επίσης, υπάρχει δυνατότητα λήψης των λογιστικών στατιστικών της επιχείρησης, τα οποία ανάλογα το φορολογικό σύστημα μιας χώρας, είναι υποχρεωτικά (ισολογισμός, μηνιαίο Φ.Π.Α.).

Αναπόσπαστο στοιχείο ενός ERP συστήματος, αποτελεί σίγουρα το εργαλείο αναφορών (reporting tool) που ενσωματώνει. Και αυτό γιατί, και αυτό πρέπει να ληφθούν στατιστικά στοιχεία γύρω από την εικόνα της επιχείρησης, θα πρέπει να γίνει εξαγωγή των δεδομένων σε προσεγγμένες και ακριβείς αναφορές (reports). Οι αναφορές αυτές, θα βοηθήσουν τα υψηλόβαθμα στελέχη να λάβουν καλύτερες αποφάσεις.



Βασικό στοιχείο ενός σύγχρονου ERP συστήματος, είναι η αυξημένες δυνατότητες παραμετροποίησης. Η διαφορετικότητα των ειδών των επιχειρήσεων και των επιχειρηματικών διαδικασιών που καθημερινά πραγματοποιούνται σε αυτές, επιβάλλει σε ένα σύστημα με το μέγεθος του ERP, να είναι όσο το δυνατόν πιο ευέλικτο. Με αυτόν τον τρόπο επιτυγχάνεται η προσαρμογή του σε κάθε είδους επιχείρηση.

Ακόμα μία σημαντική λειτουργία ενός συστήματος ERP είναι η διαχείριση ανθρώπινων πόρων. Μέσω του λογισμικού, αποθηκεύονται τα στοιχεία όλων των εργαζομένων. Τα στοιχεία αυτά μπορεί να είναι στατικά (όνομα, επώνυμο, ημερομηνία γέννησης) είτε δυναμικά, όπως η μισθοδοσία των εργαζομένων, η ειδικότητα εργασίας κτλ. Με αυτήν την διαδικασία, μπορεί να απεικονιστεί πλήρως το μερίδιο που έχει το εργατικό δυναμικό στα συνολικά έξοδα της επιχείρησης. Αυτό οδηγεί στο να βελτιωθούν οι αποφάσεις γύρω από την μείωση ή την αύξηση του προσωπικού, ανάλογα με τις ανάγκες που προκύπτουν από τα στοιχεία αυτά.

Η σωστή διαχείριση των σχέσεων με τους πελάτες της, είναι ζωτικής σημασίας για κάθε επιχείρηση. Η ικανοποίηση του πελάτη – χωρίς να αποτελεί μετρήσιμο μέγεθος – αποτελεί πρώτης προτεραιότητας στόχο – μαζί με την αύξηση των εσόδων της. Χωρίς αυτήν, επέρχεται μείωση του πελατολογίου. Για το λόγο αυτό, δημιουργήθηκαν τα συστήματα CRM (Customer Relationship Management), τα οποία καταγράφουν την κάθε επαφή που έχει οποιοδήποτε στέλεχος της επιχείρησης με οποιοδήποτε πελάτη της. Μέσω της καταγραφής και της εξαγωγής των στοιχείων αυτών σε αναφορές (reports), μπορεί με έμμεσο τρόπο να μετρηθεί η ικανοποίηση του πελάτη που είναι σημαντική τόσο για μην τον χάσει η εταιρία, όσο και για την επέκταση των πωλήσεων μέσω της εκμετάλλευσης της ικανοποίησης αυτής (συστάσεις σε άλλους, δυνητικούς πελάτες από το περιβάλλον του ενεργού πελάτη). Οι επιμέρους λειτουργίες της ενότητας CRM ενός ERP λογισμικού περιγράφονται στην επόμενη ενότητα της παρούσης.

Τέλος, ο λόγος για τον οποίο τα συστήματα ERP, έλαβαν την συγκεκριμένη ονομασία (Enterprise Resource Planning) είναι κυρίως λόγω της υποστήριξης της Παραγωγής. Με βάση την πρόβλεψη των πωλήσεων, καταστρώνονται τα σχέδια παραγωγής και καθορίζονται με βάση αυτά, η ποσότητα των πόρων που είναι απαραίτητη, καθώς και ο χρονικός ορίζοντας που θα πρέπει αυτοί να είναι διαθέσιμοι. Σε περίπτωση που παρουσιαστούν ελλείψεις, δημιουργούνται παραγγελίες υλικών. Επίσης, παρακολουθείται το χρονοδιάγραμμα της παραγωγικής διαδικασίας και κατά πόσο αυτό τηρείται. Επίσης, έχοντας όλα τα οικονομικά δεδομένα στη διάθεσή μας (κόστος υλικών, τιμή πώλησης τελικών προϊόντων, κόστος εργατικού δυναμικού), μπορούμε να έχουμε σαφή εικόνα των εσόδων και των εξόδων της παραγωγικής διαδικασίας.

## 2.3 Η δομή και οι λειτουργίες ενός Συστήματος ERP

Στην ενότητα αυτή, θα γίνει λεπτομερής περιγραφή των δυνατοτήτων και των λειτουργιών των συστημάτων ERP. Οι λειτουργίες, όπως και ο τρόπος δομής των ERP (σε ενότητες – modules), ανεπτυγμένων από διαφορετικούς κατασκευαστές, έχουν πολλά κοινά σημεία μεταξύ τους. Στην παρούσα μελέτη, θα χρησιμοποιηθεί ως σημείο αναφοράς το ERP της εταιρίας Softone. Οι ενότητες αυτές, αναλύονται σε ένα ή περισσότερα κυκλώματα στην εφαρμογή. Με την έννοια κύκλωμα, εννοούμε μια θεματική ενότητα η οποία αντιστοιχεί – συνήθως – σε μια επιλογή στο κεντρικό μενού επιλογών της εφαρμογής και αποτελεί θεμελιώδη οντότητα για την επιχείρηση γενικότερα. Παράδειγμα ενός κυκλώματος, αποτελούν οι 'Πελάτες' στο Softone ERP ή οι 'Πωλήσεις'. Τα κυκλώματα αυτά, δεν αποτελούν ενότητα (module) της εφαρμογής, αλλά είναι συστατικά τους (π.χ. η ενότητα 'Πωλήσεις' αποτελεί συστατικό της ενότητας 'Εμπορικές Συναλλαγές' που αναλύεται παρακάτω).

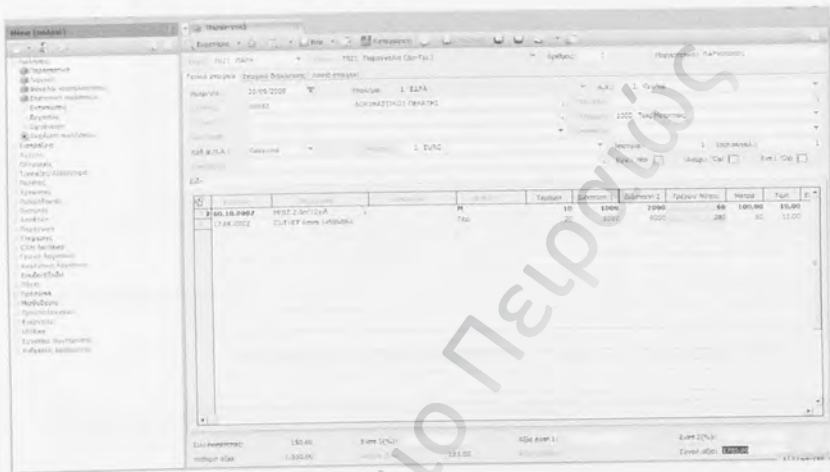
### 2.3.1 Εμπορικές Συναλλαγές

Η διαχείριση των εμπορικών συναλλαγών μιας επιχείρησης αποτελεί την κυριότερη καθημερινή διαδικασία ενός συστήματος ERP. Η συνήθης κύκλος ενεργειών που πραγματοποιείται σε μια επιχείρηση, ξεκινά με την παραγγελία κάποιου προϊόντος από ένα πελάτη. Η παραγγελία αυτή, μπορεί να είναι τηλεφωνική, γραπτή ή και να προέρχεται από το Internet (σε περίπτωση διασύνδεσης του ERP με το e-shop της επιχείρησης). Το σύστημα ενημερώνεται, και αυτόματα δεσμεύονται τα προϊόντα τα οποία παρήγγειλε ο πελάτης. Σε περίπτωση που δεν υπάρχει στην αποθήκη μας ο αριθμός των προϊόντων που παρήγγειλε ο πελάτης, ενεργοποιούνται διαφορετικές διαδικασίες, ανάλογα με το είδος της επιχείρησης. Είτε πραγματοποιείται παραγγελία – και κατόπιν αγορά – σε προμηθευτή (σε περίπτωση απλής εμπορικής επιχείρησης), είτε ενεργοποιούνται οι μηχανισμοί της παραγωγικής διαδικασίας (σε περίπτωση που η επιχείρηση διαθέτει παραγωγικό τμήμα). Όταν το προϊόν, είναι πλέον διαθέσιμο, πραγματοποιείται η παράδοση του στον πελάτη

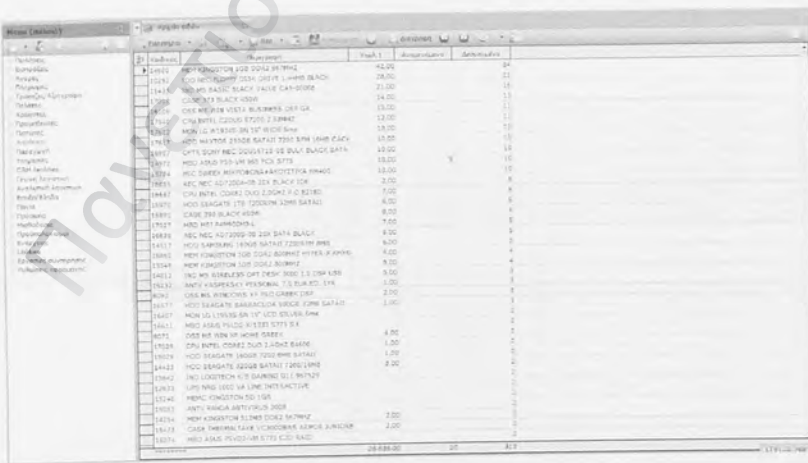
και η έκδοση των απαραίτητων παραστατικών (Απόδειξη Λιανικής, Τιμολόγιο). Τα παραστατικά αυτά ενημερώνουν:

1. Το υπόλοιπο του Πελάτη
2. Το υπόλοιπο του προϊόντος στην Αποθήκη
3. Τα λογιστικά κυκλώματα του ERP (χρέωση, πίστωση λογαριασμών λογιστικής)
4. Τα στατιστικά πρόβλεψης Πωλήσεων και Αγορών.

Στο Softone ERP, η διαδοχή των προαναφερθέντων ενεργειών φαίνεται στις παρακάτω οθόνες:  
Καταχώρηση παραγγελίας – Δέσμευση Προϊόντος



Εικόνα 1: Καταχώρηση Παραγγελίας



Εικόνα 2: Δέσμευση Προϊόντος μετά την καταχώρηση Παραγγελίας ("Δεσμευμένα")

i) Καταχώρηση Αγοράς Προϊόντος



Εικόνα 3: Αγορά Προϊόντος

ii) Πώληση Προϊόντος-Διαμόρφωση Υπολοίπου Πελάτη/Αποθήκης



Εικόνα 4: Πώληση Προϊόντος



Εικόνα 5: Διαμόρφωση Υπολοίπου αποθήκης του πωληθέντος προϊόντος

Παραπάνω, περιγράφηκε μια απλή διαδικασία πώλησης σε μια εμπορική επιχείρηση. Οι θόνοι της εφαρμογής, όπως βλέπουμε, είναι παρόμοιες είτε βρισκόμαστε στο κύκλωμα των αγορών είτε στο κύκλωμα των πωλήσεων. Το γεγονός αυτό, όπως αναφέρθηκε και στην εισαγωγή του κεφαλαίου αυτού, είναι πολύ σημαντικό σε ένα σύγχρονο ERP. Η διαεπαφή πρέπει να είναι κοινή, ανεξαρτήτου κυκλώματος και module της εφαρμογής.

Προηγουμένως, αναπαράσταθηκε ένα στιγμιότυπο καθημερινών λειτουργιών που πραγματοποιούνται από τα μέλη μιας εταιρίας. Τα μέλη που χειρίζονται το σύστημα, εκτελώντας αυτές τις καθημερινές ενέργειες ρουτίνας, ταυτόχρονα τροφοδοτούν την κεντρική βάση δεδομένων του ERP με στοιχεία τα οποία εφόσον εξαχθούν μέσω έγκυρων και ευπαρουσιαστων αναφορών (reports), μπορούν να βοηθήσουν τα υψηλόβαθμα στελέχη στη λήψη σημαντικών αποφάσεων. Μερικά από τα στατιστικά τα οποία μπορεί να λάβει ένα στέλεχος των πωλήσεων μπορεί να είναι:

- i) Στατιστικά πωλήσεων ανά προϊόν / κατηγορία προϊόντος
- ii) Στατιστικά πωλήσεων ανά πελάτη / κατηγορία πελατών
- iii) Στατιστικά πωλήσεων ανά πωλητή
- iv) Λίστα των πιο επικερδών προϊόντων ανά περιοχή
- v) Λίστα των πιο επικερδών πελατών ανά πωλητή

Αυτά είναι μερικά από τα στατιστικά που δύναται να λάβει ένα στέλεχος όπως ο Διευθυντής Πωλήσεων. Αυτά, θα τον βοηθήσουν να λάβει στρατηγικές αποφάσεις όπως π.χ. ποιά προϊόντα πρέπει να προωθηθούν περισσότερο με βάση την πορεία που ακολουθούν στην αγορά ή πως πρέπει να κατανεμηθούν οι πωλητές με βάση την απόδοσή τους σε συγκεκριμένα προϊόντα.

Τέλος, παρατηρούμε πως η ενότητα της Εμπορικής Διαχείρισης αλληλεπιδρά με τις εξής:

- i) Διαχείριση Αποθήκης (Stock Management)

Μετά από κάθε αγοροπωλησία, ενημερώνονται κατάλληλα τα υπόλοιπα των προϊόντων που υφίστανται στην αποθήκη, ώστε να υπάρχει σωστή εικόνα διαθεσιμότητας αυτών.

## ii) Διαχείριση Πελατών/Προμηθευτών (CRM)

Μετά από κάθε συναλλαγή, ενημερώνεται κατάλληλα το υπόλοιπο του πελάτη (ή του προμηθευτή, ανάλογα με το είδος του συναλλασσόμενου) ώστε να απεικονίζονται σωστά στο σύστημα οι οφειλές από και προς την επιχείρηση.

## iii) Λογιστικές Ενότητες (Accounting)

Η καταχώρηση κάποιας συναλλαγής μπορεί να κατανέμει αυτόματα το ποσό της συναλλαγής στους κατάλληλους λογαριασμούς λογιστικής (περισσότερα για τις λογιστικές ενότητες ακολουθούν παρακάτω) σύμφωνα με το λογιστικό σχέδιο της επιχείρησης.

### 2.3.2 Χρηματοοικονομική διαχείριση (Accounting and Finance)

Ένας από τους σημαντικότερους στόχους ενός συστήματος ERP – όπως προαναφέρθηκε – είναι να μπορεί να παρέχει μία πλήρη οικονομική εικόνα της επιχείρησης προς τα υψηλόβαθμα στελέχη της (οικονομικούς διευθυντές, διευθυντές πωλήσεων). Ο στόχος αυτός επιτυγχάνεται μέσω της αξιοποίησης της ενότητας της χρηματοοικονομικής διαχείρισης που παρέχουν όλα τα ERP συστήματα. Η ενότητα αυτή καταγράφει την ροή των χρημάτων από και προς την επιχείρηση. Επίσης, αποθηκεύονται όλες οι τραπεζικές συναλλαγές.

Το αποτέλεσμα αυτής της διαδικασίας, κρατά ενήμερα τα αρμόδια στελέχη γύρω από την ρευστότητα της επιχείρησης σε μια δεδομένη χρονική στιγμή, πράγμα το οποίο βοηθά στην λήψη αποφάσεων σχετικά με την περαιτέρω ανάπτυξη της. Αναλυτικά, τα κυκλώματα που αφορούν την ενότητα αυτή στο Softone ERP είναι:

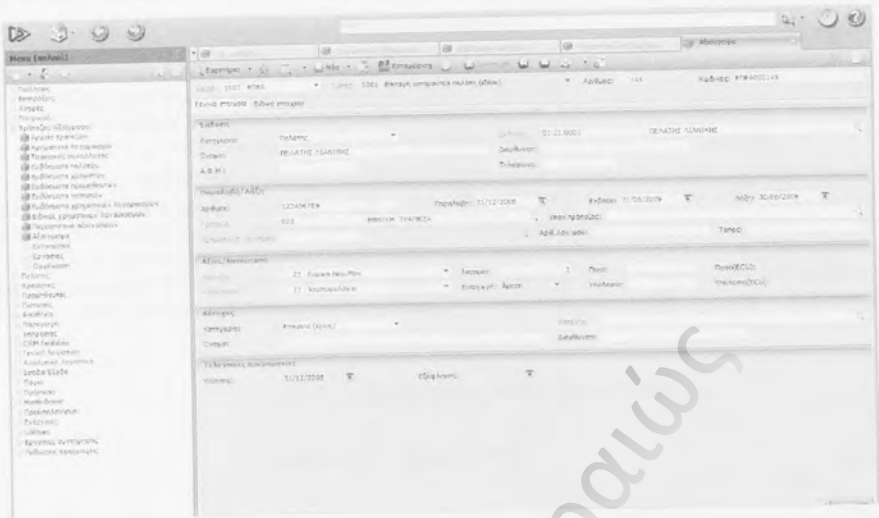
## i) Διαχείριση Εισπράξεων

Στο συγκεκριμένο κύκλωμα, αποθηκεύονται όλες οι εισπράξεις που πραγματοποιούνται σε καθημερινή βάση. Οι εισπράξεις αυτές, αφορούν κυρίως μετρητά. Μετά την καταχώρηση μιας εισπράξης, ενημερώνεται κατάλληλα το – τυχών - υπόλοιπο του συναλλασσόμενου με την επιχείρηση. Η παρακάτω εικόνα, δείχνει την καταχώρηση μιας εισπράξης και η αμέσως επόμενη, τη διαμόρφωση του υπολοίπου του πελάτη (ή του προμηθευτή) έπειτα από αυτό.

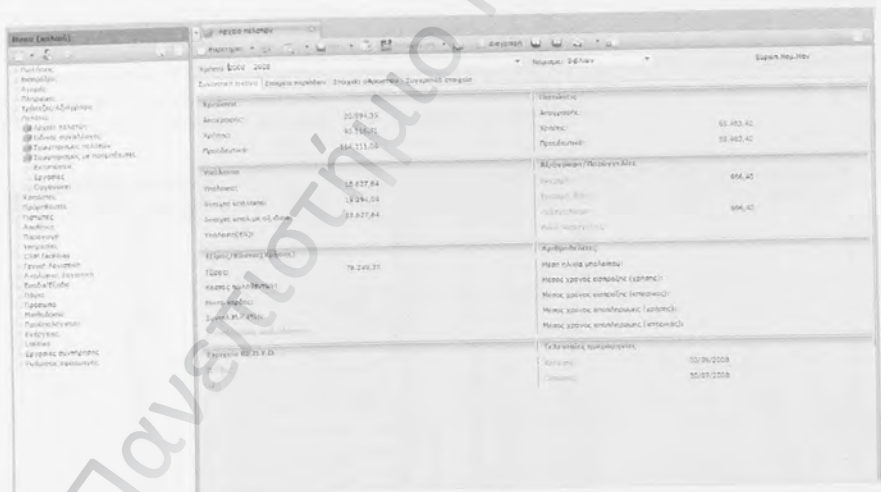


Εικόνα 6: Εισπραξη Μετρητών





Εικόνα 8: Καταχώρηση Αξιόγραφων



Εικόνα 9: Κατανομή του υπολοίπου σε επιμέρους ποσά

Τέλος, στηριζόμενοι σε όλα αυτά τα δεδομένα που βρίσκονται αποθηκευμένα στο ERP σύστημα, η επιχείρηση μπορεί να εξάγει κάθε είδους σχετικό εκτυπωτικό έγγραφο. Αυτό μπορεί να αφορά τα χρήματα που βρίσκονται στους λογαριασμούς της, τα ανοικτά υπόλοιπα προς τους προμηθευτές της, την κατάσταση των αξιόγραφων που διακινήθηκαν από και προς αυτή και όποιο συνδυασμό αυτών είναι επιθυμητός και θα την διευκόλυνε στο σχηματισμό της πλήρους οικονομικής εικόνας της επιχείρησης – που είναι το κύριο ζητούμενο της ενότητας «Χρηματοοικονομική Διαχείριση»

### 2.3.3 Λογιστικές ενότητες (Accounting and Finance)

Οι λογιστικές ενότητες αποκτούν βαρύνουσα σημασία κυρίως στα ελληνικά ERP. Η πολυπλοκότητα του φορολογικού συστήματος της Ελλάδας, αποτελεί μία πραγματικότητα η οποία είναι αδύνατο να αφήσει ανεπηρεάστη μια κατηγορία λογισμικών η οποία είναι υπεύθυνη για την καθολική καταγραφή των πόρων μιας επιχείρησης – πόσο μάλλον των χρηματικών, που αποτελούν δείκτη της υγείας μιας εταιρίας.

Οι λογιστικές ενότητες λοιπόν, θα πρέπει να εξυπηρετούν και να κάνουν ταχύτερη και αποδοτικότερη την δουλειά των λογιστών μιας επιχείρησης (είτε αυτοί είναι εξωτερικοί συνεργάτες, είτε εργάζονται σε εσωτερικό λογιστήριο). Τα περισσότερα ελληνικά ERP έχουν την δυνατότητα αυτόματης κατανομής του ποσού μιας συναλλαγής σε συγκεκριμένους λογαριασμούς λογιστικής. Αυτό, εξάλλου, είναι και η κυριότερη εργασία ενός λογιστή. Η κατανομή δηλαδή όλων των ποσών όλων των συναλλαγών μιας επιχείρησης σε συγκεκριμένες κατηγορίες έτσι ώστε να σχηματίζεται η τελική αναφορά, η οποία θα μεταφράζει τα εξής – κυριότερες σημασία – ποσά:

- i) Το ισοζύγιο μιας επιχείρησης για ένα χρονικό διάστημα
- ii) Τις υποχρεώσεις σε φόρους (ΦΠΑ)
- iii) Τις υποχρεώσεις προς ασφαλιστικά ταμεία (ΦΜΥ)

Η εξαγωγή αυτών των ποσών/αναφορών εξυπηρετεί δύο σκοπούς: Πρώτα απ' όλα με ταχύ τρόπο, και όχι με την παραδοσιακή μέθοδο της συλλογής όλων των αντιγράφων των παραστατικών και την εκτέλεση πράξεων, κατανοούνται τα ποσά των συναλλαγών και εξάγονται τα ποσά καταβολής προς την εφορία. Κατά δεύτερο, ο λογιστής μέσω του ERP εξάγει το ισοζύγιο (ή στο τέλος κάθε χρόνου τον ισολογισμό) της εταιρίας, που αποτελεί την οικονομική 'φωτογραφία' της για ένα χρονικό διάστημα. Επίσης εξάγονται και τα ποσά που πρέπει να κατατεθούν στα ασφαλιστικά ταμεία για τους εργαζομένους της.

Παρακάτω ακολουθούν εικόνες, όπου φαίνεται ένα πρότυπο λογιστικό σχέδιο μιας επιχείρησης, στη συνέχεια ένα παράδειγμα ισολογισμού, και τέλος μια 'γέφυρα', η οποία αναπαριστά με ποιόν τρόπο συνδέονται οι εμπορικές συναλλαγές με κατάλληλες λογιστικές χρεοπιστώσεις.

Κωδικός	Περιγραφή	Τύπος Λογαριασμού	Ποσοστό	Κύριο	Βαθμ.	Τέλος Βιβλίου
10	ΕΛΛΗΝΟΚΑΤΑΝΟΜΗΤΕΣ ΕΣΟΔΑ	ΕΣΟΔΑ	40	100	2	1/1/1990
11	ΕΠΙΧΕΙΡΗΣΙΑΚΑ ΕΣΟΔΑ	ΕΣΟΔΑ	40	200	2	1/1/1990
12	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΠΑΡΑΓΩΓΗΣ	ΕΣΟΔΑ	40	300	2	1/1/1990
13	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΕΜΠΟΡΕΥΜΑΤΟΣ	ΕΣΟΔΑ	40	400	2	1/1/1990
14	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΠΑΡΑΧΕΙΡΑΣΜΟΥ	ΕΣΟΔΑ	40	500	2	1/1/1990
15	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	600	2	1/1/1990
16	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	700	2	1/1/1990
17	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	800	2	1/1/1990
18	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	900	2	1/1/1990
19	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
20	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
21	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
22	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
23	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
24	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
25	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
26	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
27	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
28	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
29	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
30	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
31	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
32	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
33	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
34	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
35	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
36	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
37	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
38	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
39	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990
40	ΕΣΟΔΑ ΑΠΟ ΠΡΟΪΟΝΤΩΝ ΑΝΕΚΑΤΑΡΤΗΣ	ΕΣΟΔΑ	40	000	2	1/1/1990

Εικόνα 10: Πρότυπο Ελληνικό Λογιστικό Σχέδιο



Κωδ.	Παραγωγή	Παραγωγός Α.Δ.	Σταθ. Α.Δ. ΕΣ	Σύστημα	Πόσιον	Αποθήκη
01	ΠΡΑΚΤΙΚΟ			01011		01011-01011001
02	ΠΡΑΚΤΙΚΟ			01011		01011-01011002
03	ΠΡΑΚΤΙΚΟ			01011		01011-01011003
04	ΠΡΑΚΤΙΚΟ			01011		01011-01011004
05	ΠΡΑΚΤΙΚΟ			01011		01011-01011005
06	ΠΡΑΚΤΙΚΟ			01011		01011-01011006
07	ΠΡΑΚΤΙΚΟ			01011		01011-01011007
08	ΠΡΑΚΤΙΚΟ			01011		01011-01011008
09	ΠΡΑΚΤΙΚΟ			01011		01011-01011009
10	ΠΡΑΚΤΙΚΟ			01011		01011-01011010
11	ΠΡΑΚΤΙΚΟ			01011		01011-01011011
12	ΠΡΑΚΤΙΚΟ			01011		01011-01011012
13	ΠΡΑΚΤΙΚΟ			01011		01011-01011013
14	ΠΡΑΚΤΙΚΟ			01011		01011-01011014
15	ΠΡΑΚΤΙΚΟ			01011		01011-01011015
16	ΠΡΑΚΤΙΚΟ			01011		01011-01011016
17	ΠΡΑΚΤΙΚΟ			01011		01011-01011017
18	ΠΡΑΚΤΙΚΟ			01011		01011-01011018
19	ΠΡΑΚΤΙΚΟ			01011		01011-01011019
20	ΠΡΑΚΤΙΚΟ			01011		01011-01011020

Εικόνα 11: Γέφυρα Λογιστικής με το Εμπορικό Κύκλωμα

### 2.3.4 Διαχείριση Παραγωγικής Διαδικασίας/Αποθήκης

#### Διαχείριση Παραγωγής

Όπως τονίστηκε και στην εισαγωγή της παρούσας εργασίας, ο λόγος για τον οποίο υλοποιήθηκαν τα ERP συστήματα, ήταν η εξυπηρέτηση των μηχανογραφικών αναγκών βιομηχανιών και εργοστασίων μέσω ενός ενιαίου πληροφοριακού συστήματος. Το σύστημα αυτό, είναι ενιαίο με την έννοια ότι διαχειρίζεται μία κεντρική βάση δεδομένων και καταγράφονται σε αυτή όλες οι επιμέρους διαδικασίες μιας τέτοιας επιχείρησης, που αφορούν τους πόρους της (υλικά, ανθρώπινο δυναμικό, πάγια, μηχανήματα/εξοπλισμός, εγκαταστάσεις, τελικά προϊόντα).

Οι σχετικές διαδικασίες, λοιπόν, που μπορούν να υποστηριχθούν από ένα τέτοιο σύστημα χωρίζονται σε λειτουργικές και μη λειτουργικές και είναι οι εξής:

#### Λειτουργικές Διαδικασίες

- i) Αγορά Α' Υλών
- ii) Παραλαβή Α' Υλών
- iii) Ποιοτικός Έλεγχος Α' Υλών
- iv) Κοστολόγηση Α' Υλών
- v) Διαχείριση Α' Υλών / Προϊόντων
- vi) Διαχείριση/Έλεγχος Αποθέματος Αποθήκης

Μη λειτουργικές Διαδικασίες (Διοικητικού Ελέγχου)

- i) Κύριο Πρόγραμμα Παραγωγής (Master Production Schedule)
- ii) Χρονοπρογραμματισμός Παραγωγής (Production scheduling)
- iii) Σχεδιασμός Προϊόντων (Product Design)
- iv) Σχεδιασμός Πληρότητας/Χωρητικότητας (Capacity Planning)

Τα ERP συστήματα, επεκτείνουν τα παλαιότερα συστήματα διαχείρισης της παραγωγικής διαδικασίας (τα οποία εξυπηρετούσαν μόνο αυτό το σκοπό) μέσω της διασύνδεσης της ενότητας της Παραγωγής με ενότητες όπως η Χρηματοοικονομική Διαχείριση, το CRM (κυρίως με τις λειτουργίες πάνω στο κομμάτι του Marketing), τη Διαχείριση Αποθήκης και τη Διαχείριση Ανθρώπινου Δυναμικού. Η σημαντικότερη αλληλεπίδραση της Παραγωγής με τα υπόλοιπα module ενός ERP, είναι αυτή με τα οικονομικό μέρος, όπου ανακτώνται οι τιμές Ά Γλών μέσω των συναλλαγών που πραγματοποιούνται και σχηματίζεται το ακριβές κόστος ενός τελικού προϊόντος.

Η ακολουθία των λειτουργιών που πραγματοποιείται σε ένα σύγχρονο ERP και σχετίζεται με την παραγωγή είναι οι εξής: Αρχικά χρησιμοποιούνται οι δυνατότητες πρόβλεψης πωλήσεων, οι οποίες καθορίζουν τα επίπεδα στα οποία πρέπει να φτάσει ο όγκος της παραγωγής ώστε να καλύψει τη ζήτηση της αγοράς. Έπειτα, τα τελικά δεδομένα των προβλέψεων, εισάγονται στο κύκλωμα του Προγραμματισμού Πωλήσεων και Λειτουργίας. Το κύκλωμα αυτό είναι υπεύθυνο στο να καθορίσει εάν οι δυναμική της βιομηχανίας και των εγκαταστάσεών της είναι τέτοια, ώστε να καλύψει την – υπολογισμένη απ’ το προηγούμενο βήμα – ζήτηση. Στη συνέχεια, σειρά παίρνει η Διαχείριση Ζήτησης η οποία είναι υπεύθυνη στο να δημιουργήσει το Κύριο Πρόγραμμα Παραγωγής. Το Κύριο Πρόγραμμα Παραγωγής, καθορίζει ποιες θα είναι οι ημερομηνίες καθώς και οι ποσότητες των τελικών προϊόντων που θα πρέπει να παρασκευαστούν. Στη συνέχεια, βάσει του Κύριου Προγράμματος Παραγωγής, ενεργοποιείται κατάλληλα η ενότητα του Προγραμματισμού Απαιτούμενων Υλικών (Material Requirement Planning). Η ενότητα αυτή, είναι υπεύθυνη να καθορίσει ποια υλικά χρειάζονται, σε ποιές ποσότητες και σε ποιές ημερομηνίες πρέπει να εισέλθουν στην βιομηχανία ούτως ώστε να ικανοποιηθούν οι απαιτήσεις που έχουν τεθεί από το Κύριο Πρόγραμμα Παραγωγής. Επίσης, η ενότητα αυτή, δημιουργεί τις κατάλληλες παραγγελίες υλικών. Οι παραγγελίες αυτές αποστέλλονται επίσης και στην παραγωγή ως Εντολές Εργασίας. Κάθε εντολή εργασίας, περιλαμβάνει μια λίστα υλικών από τις προδιαγραφές παραγωγής, καθώς και τις απαραίτητες εργασίες οι οποίες πρέπει να γίνουν πάνω στα υλικά αυτά, προκειμένου να γίνει κατάλληλα ο μετασχηματισμός σε μέρος τελικού προϊόντος.

Παρακάτω, ακολουθεί ένας πίνακας, όπου φαίνονται οι ενότητες που διαθέτει το SAP (κυρίαρχο στον τομέα της Παραγωγής, γι αυτό και επιλέχθηκε για την παρουσίαση της συγκεκριμένης ενότητας) που αφορούν την παραγωγική διαδικασία.

<b>Sales And Operation Planning</b>	Καθορίζει το βαθμό σύμφωνα με τον οποίο η εταιρία μπορεί να υποστηρίξει την παραγωγή που έχει θέσει η πρόβλεψη πωλήσεων, ανάλογα με τις δυνατότητές της σε μηχανήματα, δυναμικό, και χρηματικούς πόρους
<b>Demand Management</b>	Συνδέει τις προβλέψεις πωλήσεων με το Κύριο Πρόγραμμα Παραγωγής και τον Προγραμματισμό Απαιτούμενων υλικών

<p><b>MPS</b></p> <p>(Master Production Plan)</p>	<p>Δημιουργεί παραγγελίες υλικών σύμφωνα με τις απαιτήσεις της Διαχείρισης Ζήτησης</p>
<p><b>MRP</b></p> <p>(Material Requirements Planning)</p>	<p>Καθορίζει ποια υλικά χρειάζονται, σε ποιές ποσότητες και πότε. Χρονοπρογραμματίζει την ανανέωση των ποσοτήτων των υλικών ανάλογα με το ρυθμό παραγωγής.</p>
<p><b>Production Execution</b></p>	<p>Δημιουργία παραγγελιών παραγωγής σύμφωνα με τις προγραμματισμένες παραγγελίες. Οι παραγγελίες παραγωγής περιέχουν δεδομένα σχετικά με τους στόχους παραγωγής, τα υλικά, τους απαιτούμενους πόρους και τα κόστη</p>
<p><b>Order Settlement</b></p>	<p>Εγκρίνει τις παραγγελίες της Παραγωγής. Συλλέγει δεδομένα για τις ποσότητες προϊόντων που παρήχθησαν και τις ημερομηνίες περάτωσης</p>

### Διαχείριση Αποθήκης

Όπως αναφέραμε στην αρχή της περιγραφής της ενότητας αυτής, οι μεγάλες βιομηχανίες ήταν ο λόγος για τον οποίο δημιουργήθηκαν τα ERP συστήματα. Όμως πέρα από αυτό το είδος επιχειρήσεων, υπάρχει μια τεράστια γκάμα εταιριών οι οποίες δεν παράγουν από μόνες τους τα προϊόντα (ή τις υπηρεσίες) τα οποία εμπορεύονται, αλλά τα αγοράζουν έτοιμα (ή τα μεταποιούν ελαφρώς). Οι επιχειρήσεις αυτές είναι εμπορικές και έχουν εξίσου μεγάλες ανάγκες όσο αναφορά την μηχανοργάνωσή τους και πιο συγκεκριμένα στο κομμάτι της διαχείρισης της αποθήκης τους (είτε πραγματική αποθήκη με εμπορεύματα, είτε εικονική αποθήκη π.χ. με προϊόντα λογισμικού).

Για αυτό το είδος επιχειρήσεων, η ενότητα ενός ERP συστήματος που θα ικανοποιήσει τις προαναφερθείσες απαιτήσεις, είναι η Διαχείριση Αποθήκης. Η κύρια αρμοδιότητα της ενότητας αυτής, είναι η ενημέρωση του χρήστη σε πραγματικό χρόνο σχετικά με το απόθεμα της επιχείρησης σε κάποιο συγκεκριμένο προϊόν (ή την συνολική ποσότητα/αξία των προϊόντων που φυλάσσονται στους αποθηκευτικούς της χώρους). Οι καθημερινές εφαρμογές της πολίτημας αυτής πληροφορίας είναι ποικίλες.

Η κυριότερη εξ' αυτών, αφορά τα στελέχη που είναι υπεύθυνα να προμηθευτούν τα είδη που εταιρία εμπορεύεται. Παρακολουθούν το απόθεμα και κάνουν τις κατάλληλες παραγγελίες στους προμηθευτές της επιχείρησης ανάλογα με τις παραγγελίες των πελατών. Παρακάτω ακολουθεί μία εικόνα που δείχνει την οθόνη που παρακολουθεί ο αρμόδιος υπάλληλος καθημερινά (εκτός από το πραγματικό υπόλοιπο, εμφανίζεται και ο αριθμός των Δεσμευμένων - από παραγγελίες πελατών - προϊόντων, καθώς και των Αναμενόμενων - από παραγγελίες σε Προμηθευτές)



παραμετροποίησης του ERP (π.χ. ένα Τιμολόγιο Αγοράς, ποια τιμή πρέπει να ενημερώνει, την μέση σταθμική, την τελευταία τιμή αγοράς ή και τις 2:). Στη συνέχεια μπορεί να εκτυπωθεί η συγκεντρωτική κατάσταση ποσοτήτων και αντίστοιχων αξιών άλλων των προϊόντων. Αυτή η διαδικασία, βοηθιά μία επιχείρηση σε περιόδους απογραφών (και όχι μόνο) να αποκτήσει μία εικόνα γύρω από τον πλούτο της σε ήδη αγορασμένα προϊόντα (stock). Σε περιπτώσεις όπως τα Ελληνικά ERP, η ετήσια απογραφή των προϊόντων είναι και υποχρέωση της επιχείρησης προς την εφορία (πέρα δηλαδή από το χαρακτήρα του στατιστικού, η συγκεκριμένη διαδικασία έχει τον χαρακτήρα υποχρέωσης για πολλές επιχειρήσεις στην Ελλάδα). Ακολουθεί μία εικόνα, όπου φαίνεται ένα εκτυπωτικό αποθήκης (Βιβλίο Αποθήκης), μέσω του οποίου πληροφορείται κάποιος συνολικά για τις ποσότητες και τις αξίες των προϊόντων που φυλάσσονται στους χώρους τις επιχείρησης.

Εικόνα 14: Βιβλίο Αποθήκης

Πέρα από τις τρεις αυτές βασικές χρήσεις της ενότητας της Διαχείρισης της Αποθήκης, υπάρχουν πολλές ακόμα. Απαριθμούνται παρακάτω και βρίσκουν εφαρμογή, η καθεμία σε διαφορετικό είδος επιχείρησης.

i) Πολλαπλοί Αποθηκευτικοί Χώροι

Η δυνατότητα παρακολούθησης πολλαπλών Αποθηκευτικών Χώρων, είναι συνήθως επιπλέον των βασικών σε ένα ERP. Βρίσκει εφαρμογή μόνο σε επιχειρήσεις που έχουν πολλές αποθήκες στην διάθεσή τους (ακόμα και ένα υποκατάστημα ή ένα σημείο πώλησης, στο ERP σύστημα θεωρείται ως αποθήκη) και υπάρχει ανάγκη παρακολούθησης του αποθέματος συνολικά.

ii) Θέσεις Αποθήκευσης (Ράφια)

Δίδεται – ως εξτρά κύκλωμα – η δυνατότητα παρακολούθησης συγκεκριμένων τομέων σε ένα αποθηκευτικό χώρο, κάνοντας έτσι την οργάνωση μιας αποθήκης καλύτερη. Βρίσκει εφαρμογή σε μεγάλους αποθηκευτικούς χώρους, όπου ο κάθε τομέας (ή ράφι) κρατά συγκεκριμένη κατηγορία προϊόντων.

### iii) Σειριακοί Αριθμοί (Serial Numbers)

Τα serial numbers ουσιαστικά ταυτοποιούν ένα προϊόν που αγοράζει ένα πελάτης. Με αυτόν τον τρόπο, μπορεί να γίνει καλύτερα διαχείριση της εγγύησης του σε περίπτωση βλάβης. Βρίσκει εφαρμογή μόνο σε επιχειρήσεις που εμπορεύονται προϊόντα με σειριακούς αριθμούς, όπως για παράδειγμα τα καταστήματα ηλεκτρονικών ειδών.

### iv) Παρτίδες (lots)

Σε πολλές επιχειρήσεις υπάρχει η ανάγκη παρακολούθησης των προϊόντων που αγοράζονται από αυτήν σε παρτίδες (γκρουπ προϊόντων με ορισμένα κοινά στοιχεία, όπως η ημερομηνία παραγωγής τους ή η ημερομηνία λήξης τους). Αυτό βοηθάει την επιχείρηση σε περίπτωση που εντοπιστεί ένα ελαττωματικό προϊόν (ή ένα προϊόν του οποίου η ημερομηνία λήξης έχει παρέλθει), να εντοπιστούν αυτόματα και τα υπόλοιπα, ώστε να αποσυρθούν από τη διαδικασία της πώλησης. Η μεγαλύτερη εφαρμογή του κυκλώματος αυτού γίνεται από επιχειρήσεις οι οποίες εμπορεύονται τρόφιμα.

### v) Χρώμα και Μέγεθος

Δίδεται μέσω του κυκλώματος αυτού, η δυνατότητα παρακολούθησης των προϊόντων σε διαφορετικά χρώματα και μεγέθη. Ένα είδος επιχειρήσεων – προφανώς - στο οποίο βρίσκει εφαρμογή είναι τα καταστήματα ρούχων. Δίδεται λοιπόν η δυνατότητα παρακολούθησης τόσο του αποθέματος (ώστε να αναπληρωθεί ένα είδος στα σημεία πώλησης, στα χρώματα και τα μεγέθη που υπάρχει ανάγκη) όσο και των κινήσεων που έχει ένα είδος με βάση τα χαρακτηριστικά αυτά.

### vi) Group και Set Προϊόντων

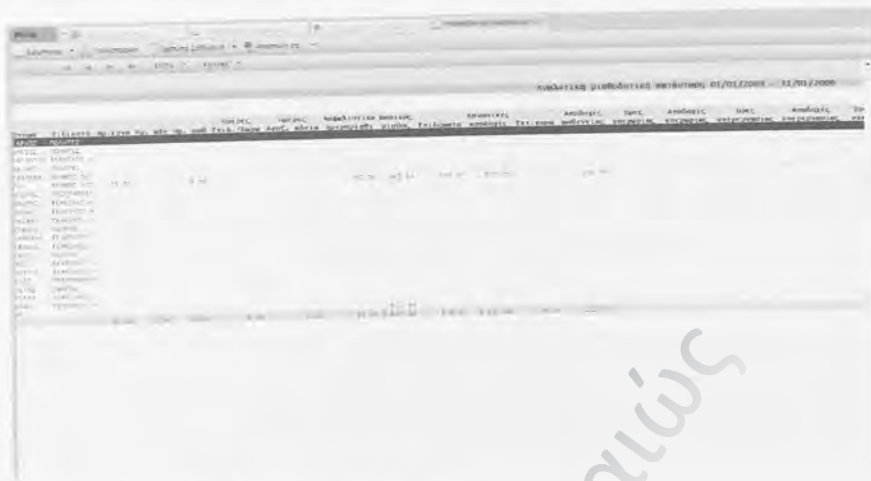
Σε πολλά είδη επιχειρήσεων, υπάρχει η ανάγκη αυτοματισμών στον τομέα της σύνθεσης προϊόντων από άλλα. Ένα κλασικό παράδειγμα, είναι η σύνθεση των μερών ενός ηλεκτρονικού υπολογιστή. Σε πολλά καταστήματα ηλεκτρονικών υπολογιστών, υπάρχουν έτοιμες συνθέσεις προς πώληση. Έτσι λοιπόν, στο ERP θα καταχωρηθούν μία φορά μόνο τα υπό-προϊόντα του προϊόντος 'Ηλεκτρονικός Υπολογιστής X'. Από εκείνο το σημείο και μετά, δεν χρειάζεται ο χειριστής ενός τερματικού να επιλέγει ένα ένα τα είδη που αγοράζει ο πελάτης. Επιλέγει απλά τη σύνθεση και το σύστημα εμφανίζει αυτόματα τα μέρη από τα οποία αποτελείται αυτή.

## 2.3.5 Διαχείριση Ανθρώπινου Δυναμικού (Human Resources)

Μία ακόμα σημαντική δυνατότητα ενός ERP συστήματος, είναι η διαχείριση του ανθρώπινου δυναμικού μιας επιχείρησης. Όπως ξέρουμε, ένα σημαντικό έξοδο για την επιχείρηση είναι η μισθοδοσία των υπαλλήλων της. Υπάρχει μεγάλη ανάγκη σε όλες τις επιχειρήσεις (ειδικά εάν ο αριθμός των εργαζομένων είναι αυξημένος) να καταγράφονται όλα τα σχετικά έξοδα της εταιρίας πάνω στο κομμάτι αυτό και να μπορούν να συγκριθούν με τα συνολικά οικονομικά δεδομένα της. Η διαδικασία αυτή, υποστηρίζει τις αποφάσεις των διοικούντων γύρω από ενδεχόμενες προσλήψεις ή απολύσεις προσωπικού, καθώς ο αριθμός και το μισθολόγιο των εργαζομένων θα πρέπει να συμβαδίζει με την ρευστότητά της.

Μέσω της ενότητας αυτής, δίδονται πολλές δυνατότητες. Αρχικά, μπορεί να καταχωρηθούν τα πλήρη στοιχεία των εργαζομένων. Αυτά μπορεί να είναι σταθερά (Όνομα, Επώνυμο, Ημερ/νία Γέννησης) είτε μεταβλητά (ειδικότητα, ημερομηνία απόλυσης). Ακολουθεί μια εικόνα, όπου φαίνεται η καρτέλα ενός εργαζομένου.





Εικόνα 16: Αναλυτική Μισθοδοτική Κατάσταση

Λαμβάνοντας υπόψη και τα ελληνικά δεδομένα, υπολογίζονται με αυτόν τον τρόπο και οι απαραίτητες εισφορές προς τα ασφαλιστικά ταμεία. Έτσι, σχηματίζεται μια ολοκληρωμένη εικόνα μέσω συγκεντρωτικών εκτυπώσεων, για τα συνολικά έξοδα μίας επιχείρησης που αφορούν τις οικονομικές υποχρεώσεις που σχετίζονται με τους υπαλλήλους της.

### 2.3.5 CRM (Customer Relationship Management)

Τα περισσότερα σύγχρονα ERP συστήματα περιλαμβάνουν την ενότητα CRM. Όπως είναι γνωστό στο χώρο της ευρύτερης αγοράς λογισμικού, τα CRM συστήματα υφίστανται και ως ανεξάρτητα συστήματα (π.χ. Microsoft Dynamics CRM) πράγμα το οποίο οφείλεται στο πλήθος των λειτουργιών τις οποίες διαθέτουν. Στη συγκεκριμένη ενότητα της εργασίας, θα παρουσιαστούν ορισμένες από τις κυριότερες λειτουργίες τους και στη συνέχεια θα εξηγηθεί πως οι λειτουργίες αυτές αλληλεπιδρούν με τις υπόλοιπες ενότητες του ERP.

Ο αποστολή της ενότητας του CRM (ή του ανεξάρτητου συστήματος αναλόγως με ποιά μορφή έχει υλοποιηθεί σε μια επιχείρηση) είναι η αποθήκευση όλων των επαφών που έχουν τα στελέχη μιας επιχείρησης με τον πελάτη. Επαφή μπορεί να θεωρήσουμε ένα απλό τηλεφώνημα προς τον πελάτη από τη γραμματεία της εταιρίας, έως την έκδοση τιμολογίου προς αυτόν. Ο στόχος της διαδικασίας αυτής, είναι να μετρηθεί – όσο αυτό είναι δυνατό - ο βαθμός ικανοποίησης που έχει ο πελάτης απέναντι στην εταιρία. Αυτό πραγματοποιείται είτε με άμεσο τρόπο π.χ. σε ένα ραντεβού ενός πωλητή το οποίο θα καταχωρηθεί στο ιστορικό του πελάτη ως μία εγγραφή, μπορεί να εισαχθεί στο σχετικό πεδίο η – κατά την κρίση του πωλητή – ικανοποίηση του πελάτη (χαμηλή, μέτρια, υψηλή), είτε με έμμεσο π.χ. μετρώντας τα ραντεβού που έχει πραγματοποιήσει το τεχνικό τμήμα μιας εταιρίας, σε συνάρτηση με την αποτελεσματικότητα και την χρέωση προς τον πελάτη (μεγάλη αποτελεσματικότητα και μικρή χρέωση είναι το ιδανικό από πλευράς πελάτη).

Πέρα από τις ενέργειες σε υπάρχοντες πελάτες που μπορεί να αποθηκεύσει και στη συνέχεια να εξάγει ένα CRM, υπάρχουν και αυτές προς υποψήφιους πελάτες. Συνεπώς, μέσω του CRM υποστηρίζονται διαδικασίες marketing. Οι διαδικασίες αυτές μπορούν να αφορούν είτε κάποια ενέργεια μαζικής αποστολής e-mail, είτε κάποια διαφημιστική καμπάνια σε κάποιο μέσο. Μέσω, λοιπόν, της ενότητας αυτής, μπορούν να μετρηθούν και οι επιδόσεις ενός τμήματος marketing μιας εταιρίας (μ' ένα εκτυπωτικό για παράδειγμα, ποιά είναι το ποσοστό των δυνητικών πελατών, που μετατράπηκαν σε ενεργούς έπειτα από μια ακολουθία από campaigns στο ραδιόφωνο).

Επίσης, μέσω του CRM, μπορούν να καταχωρηθούν όλες οι ενέργειες του τμήματος πωλήσεων μιας επιχείρησης. Και αυτό, μπορεί να συμβεί από την πρώτη επαφή που έχει η επιχείρηση με τον δυνητικό πελάτη (lead, όπως είναι η Αγγλική ορολογία), δηλαδή την απλή σύσταση του πελάτη με κάποιο στέλεχος των πωλήσεων, έως και το τελευταίο στάδιο των πωλήσεων που είναι η προσφορά και η παραγγελία κάποιων



προϊόντων ή υπηρεσιών από τον πελάτη. Φυσικά μέσω των κατάλληλων εκτυπωτικών, μπορεί να μετρηθεί η αποδοτικότητα του τμήματος αυτού.

Το μερίδιο των εξόδων μιας επιχείρησης που αναλογούν στα τμήματα πωλήσεων και marketing είναι συνήθως σημαντικά μεγάλο. Εκτός από χρηματικούς πόρους, τα τμήματα αυτά χρειάζονται και το κατάλληλο προσωπικό. Άρα λοιπόν, επιβάλλεται όχι μόνο η μέτρηση των συνολικών πόρων που δαπανούνται σε αυτά τα τμήματα, αλλά και η μέτρηση των αποτελεσμάτων που αυτά έχουν. Επειδή η φιλοσοφία μέτρησης των αποτελεσμάτων αλλάζει ριζικά σε σχέση με τις υπόλοιπες ενότητες ενός ERP (τα μετρούμενα μεγέθη δεν αποτελούν άμεσα οφέλη προς στην εταιρία – όπως π.χ. τα χρηματικά -, αλλά έμμεσα –όπως π.χ. η ικανοποίηση του πελάτη), η χρησιμοποίηση της ενότητας του CRM κρίνεται απαραίτητη.

### 2.3.6 Αναφορές και παραμετροποίηση (Reports & Customization)

Ο σημαντικότερος λόγος για τον οποίο ένας επιχειρηματίας θα επενδύσει ένα σημαντικό ποσό που απαιτείται για την υλοποίηση του ERP στην δικιά του εταιρία (ή στη δικιά τους, αν πρόκειται για πολλούς) είναι – πέρα απ' την καθημερινή λειτουργικότητα που προσφέρει – οι εξαγόμενες αναφορές (Reports). Οι αναφορές αυτές, απεικονίζουν ουσιαστικά την αποτελεσματικότητα της επιχείρησης και μπορούν να τον βοηθήσουν να πάρει καλύτερες αποφάσεις για το μέλλον της επιχείρησης σε σχέση με το παρελθόν. Επίσης, πολύ συχνά μπορούν να εκτιμηθούν μελλοντικές καταστάσεις της επιχείρησης, μέσω της ανάλυσης δεδομένων του παρελθόντος π.χ. προβλέψεις πωλήσεων για το επόμενο έτος. Ακολουθεί μια εικόνα που δείχνει την πορεία των πωλήσεων μιας εικονικής εταιρίας και πιο κάτω μία ακόμα, η οποία δείχνει ένα χρήσιμο εκτυπωτικό πωλήσεων ανά κατηγορία προϊόντων.



Εικόνα 17: Εκτύπωση Ανάλυσης Πωλήσεων



διαδικτυακής σύνδεσης (Web Connection) δεν είναι απλώς ο πάροχος ή ο δέκτης της πληροφορίας, αλλά μπορεί να συμμετέχει σε ένα περιβάλλον κατανεμημένων εφαρμογών και να πραγματοποιήσει απομακρυσμένα αιτήματα για κάποια συγκεκριμένη διαδικασία (remote procedure call) μιας Υπηρεσίας Ιστού. Η χρήση τυποποιημένων πρωτοκόλλων Διαδικτύου και άλλων προτύπων των Υπηρεσιών Ιστού επιτρέπει στις υπηρεσίες να εργαστούν πάνω σε διαφορετικά περιβάλλοντα, πράγμα το οποίο λύνει τα όποια θέματα διαλειτουργικότητας και συμβατότητας που θα μπορούσαν να ανακύψουν.

Οι εφαρμογές τοπικών δικτύων (Intranets) όπως και οι εφαρμογές που χρησιμοποιούνται μέσω extranets, δείχνουν σήμερα να επωφελούνται περισσότερο από τις λειτουργίες των Υπηρεσιών Ιστού. Οι εφαρμογές που απευθύνονται σε απλούς χρήστες, δεν χρειάζεται να εμπεριέχουν τεχνικές κατανεμημένης επεξεργασίας, όπως αυτές των Υπηρεσιών Ιστού. Η κατανεμημένη επεξεργασία απαιτείται πιο συχνά στις μεγάλες σε μέγεθος εταιρικές εφαρμογές. Επιπλέον, για λόγους ασφάλειας, οι επιχειρήσεις δεν έχουν κανένα λόγο να επιτρέπουν στον οποιοδήποτε ανώνυμο χρήστη την πρόσβαση σε ευαίσθητες πληροφορίες - οι οποίες διακινούνται κατά κόρων μέσω των Υπηρεσιών Ιστού - εκτός άμα υφίσταται σε μεγάλο βαθμό το μέτρο της εμπιστευσιμότητας, το οποίο φυσικά αποκτάται -μόναχα - μέσω των ποικίλων πιστοποιήσεων που παρέχουν οι τεχνικές ασφάλειας των Υπηρεσιών Ιστού.

Οι υπηρεσίες Ιστού διευκολύνουν τη δημιουργία και την εξυπηρέτηση αιτημάτων τρίτων είτε αυτοί αποτελούν κάποια διαφορετική - αλλά συνεργάτιδα - επιχείρηση είτε κάποιο άλλο τμήμα της συγκεκριμένης επιχείρησης. Το αίτημα αυτό περιλαμβάνεται μέσα σε ένα XML μήνυμα, το οποίο μπορεί να υπογραφεί για πιστοποίηση προκειμένου να διασφαλιστεί η ασφάλεια του περιεχομένου του.

### 3.2. Σύνομη περιγραφή των κύριων τεχνολογιών των υπηρεσιών ιστού

Η XML θεωρείται ένας πολύτιμος μηχανισμός/γλώσσα προγραμματισμού για την ανταλλαγή δεδομένων μέσω του διαδικτύου. Το SOAP αποτελεί ένα πρωτόκολλο για απόστολή μηνυμάτων XML - που συγκαταλέγεται σημασιολογικά στην ίδια κατηγορία με το HTTP και το FTP - και διευκολύνει τη διαδικασία ενδοεπικοινωνίας με τρόπους που δεν ήταν μέχρι τώρα δυνατοί, ενώ το UDDI αποτελεί το πρότυπο για τη συγκέντρωση όλων των πληροφοριών σχετικά με τις υπηρεσίες ιστού και έχει πολλές ομοιότητες με την αντίστοιχη Διαδικτυακή υπηρεσία Internet Domain Network Service (DNS) όπου καταχωρούνται - με αντίστοιχο περίπου τρόπο - όλα τα Domain Names. Οι ίδιες οι υπηρεσίες περιγράφονται από την XML σε WSDL έγγραφα. Χωρίς την XML, αυτή η ευελιξία δεν θα ήταν εφικτή.

### 3.3 Σύγχρονες Τάσεις

Στη σημερινή παγκόσμια αγορά, το Διαδίκτυο δεν χρησιμοποιείται πλέον μόνο για την αξιοποίηση του ηλεκτρονικού ταχυδρομείου και την απλή περιήγηση σε ιστοσελίδες από καθημερινούς χρήστες. Το Διαδίκτυο έχει γίνει ένα κρίσιμο σταυροδρόμι, που τροφοδοτεί μια αναπτυσσόμενη λίστα προσοδοφόρων εφαρμογών ηλεκτρονικού επιχειρείν - από το ηλεκτρονικό εμπόριο μέχρι την on-line συνεργασία εφαρμογών ηλεκτρονικών καταστημάτων και προμηθευτών. Οι υπηρεσίες ιστού (web services) εκμεταλλεύονται την απανταχού παρουσία του Διαδικτύου και τη δυνατότητα που έχει να κάνει εφικτή την επικοινωνία μεταξύ on-line εφαρμογών, πληροφοριακών συστημάτων και πόρων που ανήκουν σε διαφορετικές επιχειρήσεις, αναβαθμίζοντας έτσι τις επιχειρηματικές διαδικασίες και τις σχέσεις ανάμεσα σε πελάτες, προμηθευτές και συνεργάτες παγκοσμίως.

Ενώ οι πρόωρες εφαρμογές Διαδικτύου βοήθησαν τα άτομα να επικοινωνήσουν το ένα με το άλλο, οι υπηρεσίες Ιστού στρέφονται στην παροχή βοήθειας των εφαρμογών ώστε να μπορούν να επικοινωνήσουν η μία με την άλλη.

Παλαιότερα, ένας πωλητής έπρεπε να στείλει μια εντολή αγοράς σε έναν προμηθευτή μέσω του ηλεκτρονικού ταχυδρομείου. Στην εποχή των υπηρεσιών Ιστού, η εφαρμογή διαχείρισης αγορών (συνήθως ERP σύστημα) μιας επιχείρησης επικοινωνεί μέσω του Διαδικτύου άμεσα με την εφαρμογή Διαχείρισης Αποθήκης (Warehouse Management Application) ενός προμηθευτή για να ελεγχθεί η τιμή και η διαθεσιμότητα, υποβάλλοντας εντολή αγοράς στην εφαρμογή διαχείρισης παραγγελιοληψιών του προμηθευτή (επίσης κάποιο ERP σύστημα) - αυτόματα, με ελάχιστη ή καμία ανθρώπινη επέμβαση.

Τα οφέλη των υπηρεσιών Ιστού δεν περιορίζονται στις αλληλεπιδράσεις μεταξύ των διαφορετικών επιχειρήσεων. Οι επιμέρους μονάδες μέσα στην ίδια επιχείρηση χρησιμοποιούν συχνά ποικίλα διαχειριστικά περιβάλλοντα. Κάθε τμήμα, και κατ' επέκταση οι πολιτικές διαχειρίσεις (π.χ. πολιτικές ασφάλειας) από τις

οποίες το κάθε ένα από αυτά διέπεται, συχνά ποικίλουν. Αυτό που, επίσης, κάνει τις υπηρεσίες Ιστού τόσο ενδιαφέρουσες είναι ότι μπορούν να παρέχουν συμβατότητα ανάμεσα σε διαφορετικού τύπου πολιτικές ασφάλειας.

Όμως, ενώ οι υπηρεσίες Ιστού προσφέρουν ελκυστικά πλεονεκτήματα, δημιουργούνται πολλά ερωτηματικά και ζητήματα σχετικά με την εμπιστευτικότητα και την ασφάλεια που μπορούν να εξασφαλίσουν. Δεν υφίσταται πλέον το "back office" που σφραγίζεται ερμητικά από τον εξωτερικό κόσμο. Με την πραγματοποίηση κρίσιμων επιχειρηματικών λειτουργιών μέσω Διαδικτύου, οι υπηρεσίες Ιστού μπορούν να εκθέσουν πολύτιμα εταιρικά δεδομένα, εφαρμογές και συστήματα σε αναρίθμητες εξωτερικές απειλές. Αυτές οι απειλές δεν είναι φανταστικές. Κυμαίνονται από τυχαίες πράξεις Διαδικτυακού βανδαλισμού μέχρι και περίπλοκες, στοχοθετημένες πράξεις κλοπής, απάτης, ή δολιοφθοράς πληροφοριών. Σημασία, λοιπόν, δεν έχει να αναφέρουμε τους πιθανούς κινδύνους που ελλοχεύουν στο Διαδίκτυο, αλλά να επισημάνουμε το μέγεθος της καταστροφής που θα μπορούσε να επιφέρει μια τέτοια επίθεση. Σύμφωνα, λοιπόν, με τα δεδομένα που επικρατούν, η ασφάλεια πρέπει να είναι ζήτημα κεντρικής σημασίας κατά την σχεδίαση και υλοποίηση οποιασδήποτε εφαρμογής Υπηρεσιών Ιστού.

Με τον ίδιο τρόπο, λοιπόν, που οι επιχειρήσεις Πληροφορικής ακολουθούν τεχνικές ανάπτυξης συμβατότητας μεταξύ εφαρμογών (Enterprise Application Integration-EAI), με τον ίδιο τρόπο μπορούν να εκμεταλλευτούν τις τεχνικές συμβατότητας ασφαλείας εφαρμογών (Enterprise Application Security Integration - EASI) ώστε να διαθέτουν ολοκληρωμένες και εξελισσόμενες εφαρμογές.

### 3.4 Extensible Markup Language (XML)

Η γλώσσα XML αναπτύχθηκε από μια ερευνητική ομάδα εργασίας κάτω από την επίβλεψη του διεθνούς οργανισμού World Wide Web Consortium (W3C) το 1996. Εδραιώθηκε από τον John Bosak της Sun Microsystems.

Η XML σχεδιάστηκε για να ξεπεράσει τους περιορισμούς της HyperText Markup Language (HTML) και ειδικότερα να υποστηρίξει καλύτερα τη δημιουργία και τη διαχείριση δυναμικού περιεχομένου. Επιπλέον δίνει στα έγγραφα ένα μεγαλύτερο επίπεδο προσαρμοστικότητας στη μορφή και τη δομή από αυτό που υπήρχε παλαιότερα στην HTML. Η XML προσφέρει στους σχεδιαστές της, τη δυνατότητα να προσθέτουν περισσότερα στοιχεία στη γλώσσα. Στην HTML οι ετικέτες (tags) είναι προκαθορισμένες, ενώ η XML παρέχει τη δυνατότητα στους χρήστες να καθορίζουν τις ετικέτες.

Η XML είναι markup γλώσσα για έγγραφα που περιέχουν δομημένες πληροφορίες. Η markup γλώσσα είναι ένας μηχανισμός που καθορίζει δομές σε ένα έγγραφο. Οι δομημένες πληροφορίες περιλαμβάνουν περιεχόμενο και κάποιες διευκρινίσεις για το ρόλο του περιεχομένου. Σχεδόν όλα τα έγγραφα έχουν την ίδια δομή. Στην πραγματικότητα, η XML είναι κάτι περισσότερο από markup γλώσσα, είναι μεταγλώσσα, δηλαδή μια γλώσσα που χρησιμοποιείται για να καθορίσει νέες markup γλώσσες.

Όλο και περισσότερές εφαρμογές χρησιμοποιούν XML για να αποθηκεύσουν πληροφορίες λόγω των πλεονεκτημάτων της, κάποια εκ των οποίων είναι:

- Η δομή είναι καθορισμένη με σαφήνεια και μπορεί να περάσει μεταξύ διαφορετικών υπολογιστικών συστημάτων, που ειδικά θα ήταν ανίκανα να επικοινωνήσουν.
- Το «ωφέλιμο φορτίο» των δεδομένων είναι εμφωλευμένα σε ετικέτες και επομένως αναγνώσιμο από τους χρήστες.
- Λόγω της κειμενικής φύσης τους, τα αρχεία XML είναι δεν εξαρτώνται από την πλατφόρμα του συστήματος.

Τα πλεονεκτήματα αυτά, έκαναν την XML το πλέον κατάλληλο πρότυπο για επικοινωνία μεταξύ υπηρεσιών Ιστού. Για να εξασφαλιστεί η ευρύτερη χρήση ανεξαρτήτου πλατφόρμας και γλώσσας προγραμματισμού, κάθε υπηρεσία Ιστού, αναπτύχθηκε το SOAP, το οποίο είναι μια XML υλοποίηση με καθορισμένα στοιχεία και μια προκαθορισμένη δομή. Στην επόμενη ενότητα αναλύεται το πρωτόκολλο αυτό.

### 3.5 Simple Object Access Protocol (SOAP)

Η μεταφορά δεδομένων έχει κεντρική σημασία στο κατακευκτικό περιβάλλον του παγκόσμιου ιστού. Καθώς η XML έχει προκύψει ως η καταλληλότερη μορφή δεδομένων, η πρόκληση για τον αποστολέα και για τον παραλήπτη είναι να συμφωνήσουν στο πρωτόκολλο μεταφοράς, είτε αυτή πρόκειται να γίνει ανάμεσα σε προγράμματα λογισμικού, είτε ανάμεσα σε μηχανήματα ή οργανισμούς.

Το SOAP είναι ένα πρωτόκολλο σχεδιασμένο για την ανταλλαγή XML εγγράφων μέσω διαφορετικών πρότυπων τεχνολογιών διαδικτύου, συμπεριλαμβανομένων των HTTP, Simple Mail Transfer Protocol (SMTP) και File Transfer Protocol (FTP).

Το SOAP είναι βασικά ένα μοντέλο μονόδρομης επικοινωνίας, το οποίο εγγυάται ότι ένα μήνυμα μεταφέρεται από τον αποστολέα στον παραλήπτη, ενδεχομένως περιλαμβάνοντας ενδιάμεσους σταθμούς που μπορούν να επεξεργαστούν μέρος του μηνύματος ή να το μεταβάλουν.

Ένα μήνυμα SOAP είναι ένα συνηθισμένο XML έγγραφο, το οποίο περιέχει τα ακόλουθα στοιχεία :

- **Envelope**, το οποίο προσδιορίζει ότι το έγγραφο XML είναι ένα μήνυμα SOAP. Καθορίζει την αρχή και το τέλος του μηνύματος.
- **Header**, το οποίο περιέχει πληροφορίες επικεφαλίδας. Είναι ένας ευέλικτος μηχανισμός για πρόσθεση χαρακτηριστικών στο SOAP μήνυμα.
- **Body**, το οποίο παρέχει έναν απλό μηχανισμό για ανταλλαγή υποχρεωτικών πληροφοριών που προορίζονται για τον τελικό αποδέκτη του μηνύματος.
- **Fault**, το οποίο περιέχει πληροφορίες για λάθη που τυχόν εμφανίστηκαν κατά την επεξεργασία του μηνύματος. Αυτό το στοιχείο εμφανίζεται μόνο σε απαντητικά μηνύματα και δεν πρέπει να εμφανίζεται πάνω από μία φορά μέσα στο Body του μηνύματος.

Συνοψίζοντας, το SOAP είναι ένα πρωτόκολλο βασισμένο σε XML για την αποστολή μηνυμάτων και την πραγματοποίηση κλήσεων απομακρυσμένων διαδικασιών μέσα σε ένα κατακευκτικό περιβάλλον.

### 3.6 Web Services Definition Language (WSDL)

Το επόμενο βήμα για να ολοκληρωθεί η αρχιτεκτονική επικοινωνίας των υπηρεσιών ιστού, είναι να καθοριστεί το πώς οι χρήστες θα έχουν πρόσβαση σε μία υπηρεσία, μόλις αυτή τεθεί σε εφαρμογή. Σε αυτό το σημείο παρεμβαίνει η WSDL προδιαγραφή. Η τελευταία, παρέχει ένα συμβόλαιο μεταξύ του αιτούντος και του παροχέα της υπηρεσίας.

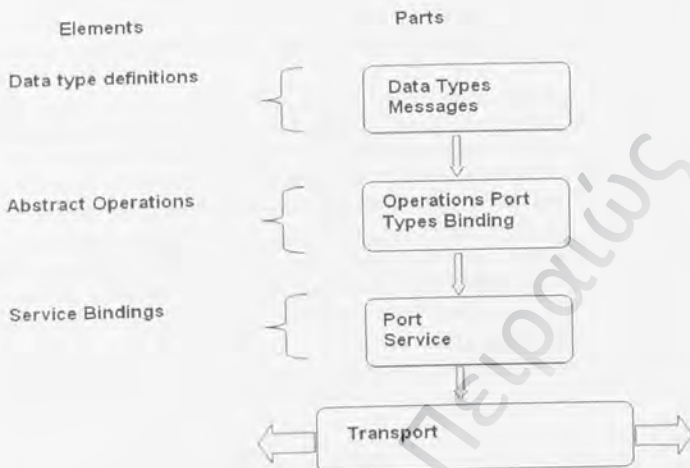
Η WSDL αναπτύχθηκε αρχικά από τη Microsoft και την IBM και υποβλήθηκε στο W3C από 25 εταιρείες. Βρίσκεται στην «καρδιά» του μοντέλου των υπηρεσιών του ιστού, παρέχοντας ένα κοινό τρόπο με τον οποίο παρουσιάζονται οι τύποι των δεδομένων που λαμβάνουν χώρα στα μηνύματα, οι λειτουργίες οι οποίες πρόκειται να εκτελεστούν στα μηνύματα και η αντιστοίχιση των μηνυμάτων πάνω σε συναλλαγές του δικτύου. Η WSDL έχει XML μορφή, η οποία περιγράφει τι κάνει μια υπηρεσία, πώς υλοποιεί τις λειτουργίες της και πού θα τη βρούμε.

Η WSDL διατρέχεται σε τρία βασικά στοιχεία και επτά τμήματα τα όποια είναι τα παρακάτω:

- i. data type definitions
- ii. abstract operations
- iii. service bindings

Κάθε βασικό στοιχείο μπορεί είτε να καθοριστεί σε ένα ξεχωριστό XML έγγραφο και να εισαχθεί σε διαφορετικούς συνδυασμούς για να δημιουργήσει μια τελική περιγραφή υπηρεσιών ιστού, είτε να υπάρξει

συνδυασμός και των τριών μαζί σε ένα μόνο έγγραφο. Τα data type definitions (Data types, Messages) προσδιορίζουν τη δομή και το περιεχόμενο των μηνυμάτων. Τα Abstract Operations (Operations, Port Types, Binding) προσδιορίζουν τις λειτουργίες που εκτελούνται στο περιεχόμενο του μηνύματος και τα Service Bindings (Port, Service) προσδιορίζουν τη μετάδοση δεδομένων, η οποία θα μεταφέρει το μήνυμα στον προορισμό του.



Εικόνα 19: Τα τρία βασικά στοιχεία και τα επτά τμήματα του WSDL.

Το μόνο πρόβλημα που παραμένει, είναι το πώς ένας ενδεχόμενος χρήστης μπορεί να βρει την αντίστοιχη περιγραφή, δηλαδή το WSDL έγγραφο. Τη λύση σ' αυτό το πρόβλημα έρχεται να δώσει η προδιαγραφή UDDI, η οποία αναλύεται στην επόμενη ενότητα.

### 3.7 Universal Description, Discovery and Integration (UDDI)

Το UDDI ως τεχνική προδιαγραφή, παρέχει μια μέθοδο για δημοσίευση και εύρεση της περιγραφής μιας υπηρεσίας ιστού. Είναι μια κεντρική υπηρεσία καταλόγου, όπου οι υπηρεσίες ιστού μπορούν να καταχωρηθούν και να προσδιοριστούν σε έναν παροχέα υπηρεσιών. Είναι μια πρωτοβουλία των εταιρειών IBM, Arriba και Microsoft που το Σεπτέμβριο του 2000 εξέδωσαν την έκδοση 1.0 του UDDI, η οποία επέτρεπε στις επιχειρήσεις τη γρήγορη και δυναμική εύρεση καθώς και τη συναλλαγή με κάθε άλλη υπηρεσία. Ακολούθησε το UDDI 2.0 το Μάιο του 2001, με επιπλέον χαρακτηριστικά, όπως η υποστήριξη μιας υπηρεσίας ιστού από πολλές γλώσσες διεθνώς και το βελτιωμένο σύνολο επιλογών αναζήτησης. Σήμερα το UDDI βρίσκεται στην έκδοση 3.0.2.

Η δομή των δεδομένων τα οποία αποθηκεύονται στον κατάλογο είναι σε μορφή XML. Τα δεδομένα τα οποία συλλέγονται εντός του καταλόγου χωρίζονται σε τρεις κατηγορίες:

- ✓ λευκές σελίδες (white pages)
- ✓ κίτρινες σελίδες (yellow pages)
- ✓ πράσινες σελίδες (green pages)

Οι λευκές σελίδες περιέχουν γενικές πληροφορίες όπως το όνομα, η περιγραφή, η διεύθυνση επικοινωνίας και μοναδικούς 'προσδιοριστές' όπως είναι οι D-Y-N-S αριθμοί ή τα ID's για την εταιρεία που προσφέρει την υπηρεσία. Αυτές οι πληροφορίες επιτρέπουν σε άλλους να ανακαλύψουν την υπηρεσία ιστού της εταιρείας βασισμένοι πάνω σε κάποιο στοιχείο αναγνώρισης. Οι κίτρινες σελίδες περιέχουν πληροφορίες οι οποίες περιγράφουν μια υπηρεσία χρησιμοποιώντας ποικίλες κατηγοριοποιήσεις. Αυτές οι πληροφορίες επιτρέπουν στους άλλους χρήστες να ανακαλύψουν την υπηρεσία βάσει της κατηγοριοποίησής της (π.χ. είναι μια εταιρεία πώλησης ή κατασκευής αυτοκινήτων). Οι πράσινες σελίδες περιλαμβάνουν λεπτομερείς τεχνικές πληροφορίες για μια υπηρεσία ιστού, επιτρέποντας κάποιον να υλοποιήσει μια εφαρμογή, χρησιμοποιώντας παράλληλα και την υπηρεσία ιστού. Αυτές οι τρεις κατηγορίες καταφέρνουν να κάνουν εύκολη την αναζήτηση για συγκεκριμένες υπηρεσίες ιστού.

Η UDDI προδιαγραφή λειτουργεί όπως το Internet Domain Network Service (DNS). Οι εταιρείες μπορούν να καταχωρηθούν σε οποιονδήποτε ξενιστή (host), όπως IBM, HP, SAP, ή Microsoft και οι πληροφορίες που παρέχουν να τοποθετηθούν στην αντίστοιχη βάση του ξενιστή. Οι ξενιστές αναρτούν WSDL περιγραφές των υπηρεσιών ιστού για καταχώρηση και ανακάλυψη. Η WSDL παρέχει ξεχωριστά αρχεία για καταχώριση και ανακάλυψη υπηρεσιών, χρησιμοποιώντας την δική της XML μορφή εγγράφου.

Κάθε χρήστης μπορεί να διαβάσει τον κατάλογο, να ερευνά για μια επιθυμητή υπηρεσία και να φορτώνει την περιγραφή, σε περίπτωση που του ταιριάζει, από οποιονδήποτε ξενιστή. Οι χρήστες δεν θα διαβάσουν απευθείας μια UDDI καταχώρηση από τη στιγμή που η πληροφορία, η οποία είναι αποθηκευμένη εντός του καταλόγου δεν είναι απαραίτητα φιλική προς τον αναγνώστη.

Όταν χρειαστεί να γίνει ενημέρωση των δεδομένων, ο χρήστης της εταιρείας θα πρέπει να επιστρέψει στον ξενιστή όπου έγινε η αρχική καταχώρηση των δεδομένων, έτσι ώστε ο τελευταίος να μπορέσει να εκτελέσει τη λειτουργία της ενημέρωσης.

## 4. Ανάπτυξη του γεννήτορα

### 4.1 Εισαγωγή – Στόχος, δομή του συστήματος

Στην ενότητα αυτή, θα παρουσιαστεί το πρακτικό μέρος της παρούσας μελέτης. Πριν ξεκινήσει η ανάλυση του συστήματος που αναπτύχθηκε (ή των συστημάτων, γιατί δεν είναι μόνο ένα), θα πρέπει να αναφέρουμε ορισμένες προϋποθέσεις που υφίστανται, προκειμένου δύο ή περισσότερα μεγάλα συστήματα να καταφέρουν να επικοινωνήσουν με ένα τρίτο ή μεταξύ τους.

Για κάθε πληροφοριακό σύστημα, το οποίο παρουσιάζει πολυπλοκότητα ανάλογη με αυτή των συστημάτων ERP (έγινε αντιληπτή στην δεύτερη ενότητα της παρούσας εργασίας), και χρειάζεται να λειτουργήσει σε συνδυασμό με κάποιο άλλο σύστημα, θα πρέπει στην ομάδα υλοποίησης του ενδιάμεσου συστήματος (middleware) να συμμετέχουν προγραμματιστές ή παραμετροποιητές όλων των συστημάτων που πρόκειται να συνεργαστούν. Χωρίς την ύπαρξη ανθρώπων, οι οποίοι είναι εξειδικευμένοι σε τέτοιου είδους συστήματα, είναι αδύνατο να κατασκευαστεί ο «ενδιάμεσος κρίκος» επικοινωνίας, το καινούργιο ενδιάμεσο σύστημα δηλαδή.

Επίσης, προκειμένου ένα σύστημα Α, να επικοινωνήσει με ένα σύστημα Β, είτε μέσω ενός συστήματος Γ, είτε απ' ευθείας, τότε θα πρέπει στα δύο πρώτα να γίνουν προσθήκες στο κώδικα (ή και στην βάση δεδομένων ακόμα). Είναι αρκετά δύσκολο να επικοινωνήσουν σωστά και αρμονικά δύο διαφορετικά συστήματα, σε αρκετά μεγάλο εύρος λειτουργιών, χωρίς να αλλάξουν καθόλου.

Τελευταία από τις παραδοχές που σφειλούμε να κάνουμε είναι η εξής: Από την τεράστια ποικιλία των λειτουργιών ενός ERP συστήματος (παρουσιάστηκαν εκτενώς στην ενότητα 2), καταλαβαίνουμε πως είναι σχεδόν αδύνατο να αλληλεπιδράσει κάθε τύπου ERP (στην αγορά είναι αναρίθμητα) με όλα τα υπόλοιπα και σε όλες τις λειτουργίες τους. Άρα λοιπόν, η διαλειτουργικότητα μεταξύ διαφορετικών συστημάτων ERP, αφορά συγκεκριμένες ενότητες κάθε φορά και για κάθε μία από αυτές, μπορούμε να πούμε πως η διασύνδεση τους αποτελεί από μόνη της ένα ξεχωριστό έργο (project).

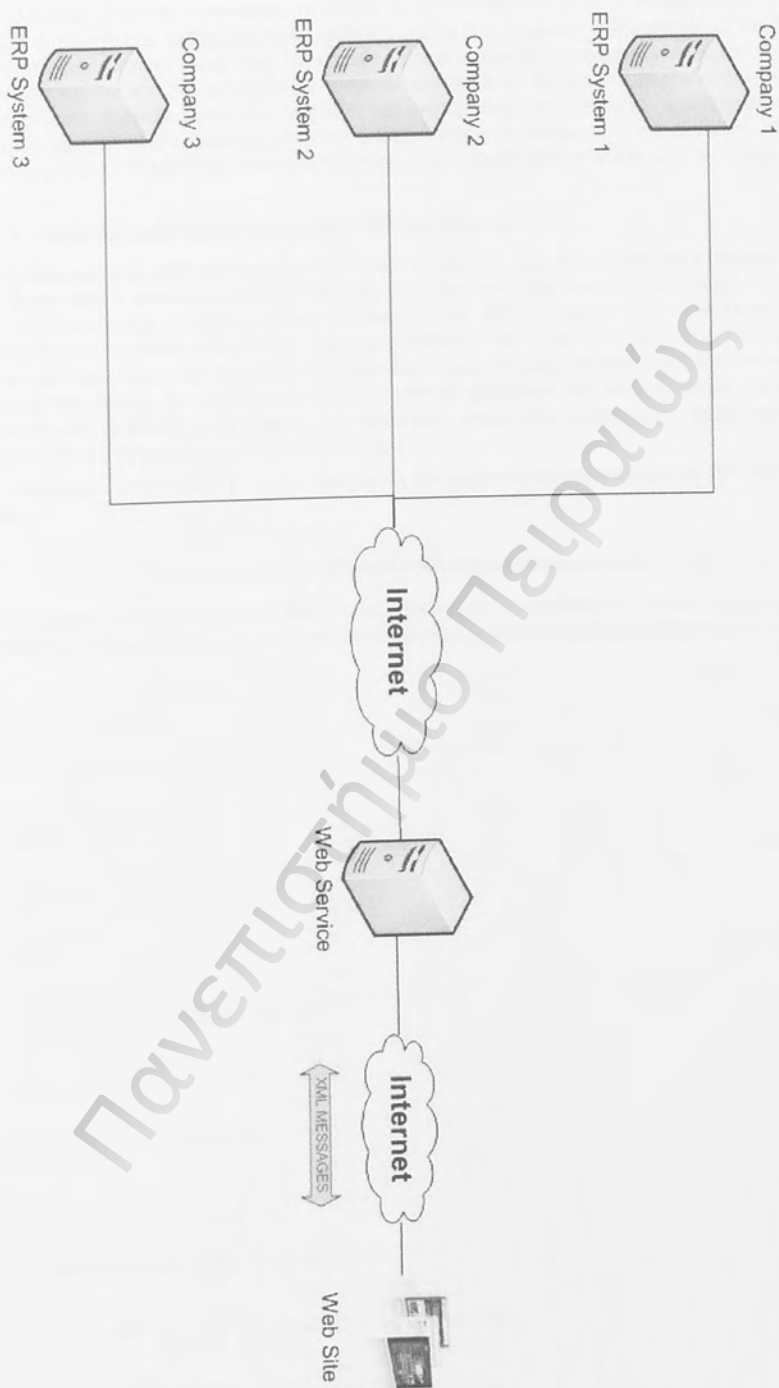
Στην παρούσα εργασία θα επικεντρωθούμε στην διαλειτουργικότητα μεταξύ ξεχωριστών ERP συστημάτων, και πιο συγκεκριμένα πάνω στην ενότητα της λογιστικής/χρηματοοικονομικής διαχείρισης. Η προσέγγιση που πραγματοποιείται, δεν είναι μόνο σε θεωρητικό επίπεδο, αλλά υπάρχει πιθανότητα κατάληξης σε χειροπιαστά οφέλη.

Ένας από τους επαγγελματικούς κλάδους, ο οποίος μπορούμε να πούμε ότι πλήττεται από την ύπαρξη πολλών ERP προϊόντων στην αγορά, είναι ο κλάδος των λογιστών. Και αυτό, γιατί ένα λογιστικό γραφείο διαχειρίζεται περισσότερες από μία επιχειρήσεις, η οποίες συνήθως διαθέτουν διαφορετικό σύστημα ERP η μία από την άλλη. Οπότε ο λογιστής είναι υποχρεωμένος (προκειμένου να εκτελέσει όλες τις απαραίτητες περιοδικές ενέργειες που αφορούν τα λογιστικά της κάθε επιχείρησης και να λάβει τις όποιες εκτυπώσεις θέλει) να χειρίζεται καθημερινά ποικιλία διαφορετικών τέτοιων συστημάτων. Φυσικά, το ιδανικό για έναν τέτοιο επαγγελματία (και όχι μόνο) θα ήταν η ύπαρξη ενός μόνο συστήματος ERP, με το οποίο θα εξοικειωνόταν και θα απέφευγε να χειρίζεται διαφορετικά λογισμικά, με διαφορετικές διεπαφές (interfaces) και φυσικά, διαφορετικό τρόπο λειτουργίας.

Για την προσέγγιση μιας λύσης στο συγκεκριμένο πρόβλημα (που εντάσσεται γενικά στο πρόβλημα της διαλειτουργικότητας μεταξύ εφαρμογών) στην παρούσα μελέτη αναπτύσσεται μια υπηρεσία ιστού και μία διαδικτυακή εφαρμογή. Η υπηρεσία ιστού έχει ως αποστολή την εκτέλεση διαφορετικών λειτουργιών πάνω σε διαφορετικά ERP, με σκοπό την εξαγωγή και αποστολή των αποτελεσμάτων σε μια διαδικτυακή εφαρμογή η οποία τροφοδοτείται από την υπηρεσία ιστού. Λειτουργίες όπως π.χ. ο μηνιαίος υπολογισμός του ΦΠΑ, μπορούν να εκτελεστούν από το σπίτι του εκάστοτε λογιστή μέσω της σύνδεσης σε μία και μόνο ιστοσελίδα. Τα στοιχεία που θα λάβει, μπορεί να αφορούν εταιρίες οι οποίες διαθέτουν διαφορετικά ERP, παρόλα αυτά ο χρήστης συνδέεται σε μία και μόνο ιστοσελίδα, η οποία συνεργάζεται στο παρασκήνιο με την υπηρεσία ιστού. Ακολουθεί σχηματικά η αρχιτεκτονική της προσέγγισης αυτής.

Πανεπιστήμιο Πειραιώς





Πανεπιστήμιο Πειραιώς

Βλέπουμε από την παραπάνω σχηματική αναπαράσταση, πως τα – διαφορετικού τύπου – ERP συστήματα συνεργάζονται με την υπηρεσία ιστού και αυτή με τη σειρά της συνεργάζεται με την διαδικτυακή εφαρμογή (web application / web site). Στη συνέχεια παρουσιάζονται και τεκμηριώνονται (με την σειρά που αναφέρονται) καθένα από τα τρία μέρη του παραπάνω σχήματος. Αρχικά, μία βάση δεδομένων (τα όποια μέρη της αφορούν την μελετώμενη ενότητα) ενός ERP συστήματος – και οι αλλαγές που πραγματοποιήθηκαν προκειμένου να δουλέψει η διαλειτουργικότητα. Στη συνέχεια, η υπηρεσία ιστού που είναι υπεύθυνη να επικοινωνήσει με τα ERP και τέλος η διαδικτυακή εφαρμογή η οποία τροφοδοτείται από την υπηρεσία ιστού.

#### 4.2 Ανάλυση μιας βάσης δεδομένων ERP συστήματος

Η βάση δεδομένων ενός ERP συστήματος, ακολουθεί σε μέγεθος και πολυπλοκότητα τα υπόλοιπα επίπεδα του λογισμικού αυτού (επίπεδο διεπαφής-interface layer, business logic layer). Είναι, δηλαδή, πολύπλοκη και μεγάλη. Η ανάλυση μιας ολοκληρωμένης βάσης δεδομένων ενός ERP συστήματος (ή γενικά όλων των βάσεων δεδομένων ERP συστημάτων, βάσει των κοινών χαρακτηριστικών τους) είναι πέρα από τους σκοπούς της παρούσας μελέτης. Όμως, θα αναφερθούν παρακάτω, ποιοί πίνακες παίζουν ρόλο στις λειτουργίες που θέλουμε να εκτελέσουμε και ποιές επιπλέον όψεις (Views) χρειάστηκε να δημιουργηθούν, προκειμένου να είναι εύκολη και ασφαλής η πρόσβαση της υπηρεσίας ιστού στη συγκεκριμένη βάση δεδομένων (θα παρουσιαστούν οι ενέργειες που αφορούν το Softone).

Οι κυριότεροι πίνακες που έχουν σχέση με τις χρηματοοικονομικές ενότητες του Softone ERP, είναι οι παρακάτω.

##### - acct (πίνακας αποθήκευσης λογαριασμών λογιστικής)

Ο πίνακας αυτός αποθηκεύει όλους τους λογαριασμούς λογιστικής, οι οποίοι έχουν δημιουργηθεί κατά βούληση του εκάστοτε λογιστή. Το script δημιουργίας του πίνακα, ακολουθεί παρακάτω.

```
USE [demo_erp]

GO

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [dbo].[ACNT] (

    [ACNT] [int] IDENTITY(1,1) NOT NULL,

    [ACNSHEMA] [smallint] NOT NULL,
```

```
[ISACTIVE] [smallint] NOT NULL CONSTRAINT  
[DF__ACNT__ISACTIVE__6C390A4C] DEFAULT (1),  
  
[SODTYPE] [smallint] NOT NULL,  
  
[CODE] [varchar] (25) COLLATE Greek_CI_AS NOT NULL,  
  
[CODE1] [varchar] (25) COLLATE Greek_CI_AS NULL,  
  
[NAME] [varchar] (64) COLLATE Greek_CI_AS NOT NULL,  
  
[NAME1] [varchar] (64) COLLATE Greek_CI_AS NULL,  
  
[ACNGRADE] [smallint] NOT NULL CONSTRAINT  
[DF__ACNT__ACNGRADE__6D2D2E85] DEFAULT (1),  
  
[ACNTYPE] [smallint] NOT NULL,  
  
[SOBEHAVIOUR] [smallint] NULL CONSTRAINT  
[DF__ACNT__SOBEHAVIOU__6E2152BE] DEFAULT (1),  
  
[ACNMOVING] [smallint] NOT NULL CONSTRAINT  
[DF__ACNT__ACNMOVING__6F1576F7] DEFAULT (0),  
  
[BALTRANSFER] [smallint] NOT NULL CONSTRAINT  
[DF__ACNT__BALTRANSFE__70099B30] DEFAULT (0),  
  
[SOCURRENCY] [smallint] NULL,  
  
[ACNCATEGORY] [smallint] NULL,  
  
[ACNGROUP] [smallint] NULL,  
  
[ACNISOLOG] [smallint] NULL,  
  
[MINVAL] [float] NULL,  
  
[MAXVAL] [float] NULL,  
  
[ACNCHK] [smallint] NOT NULL CONSTRAINT  
[DF__ACNT__ACNCHK__70FDBF69] DEFAULT (0),  
  
[CHKACTION] [smallint] NULL,  
  
[VATKEPYO] [int] NULL,  
  
[KEPYOSTS] [smallint] NOT NULL CONSTRAINT  
[DF__ACNT__KEPYOSTS__71F1E3A2] DEFAULT (0),
```

```

[ACNKEPYO] [smallint] NOT NULL CONSTRAINT
[DF__ACNT__ACNKEPYO__72E607DB] DEFAULT (0),

[ACNTVATID] [int] NULL,

[VAT] [smallint] NULL,

[ACNTAL] [int] NULL,

[ACNMODEL] [smallint] NULL,

[ISANAL] [smallint] NOT NULL CONSTRAINT
[DF__ACNT__ISANAL__73DA2C14] DEFAULT (0),

[REMARKS] [varchar](2000) COLLATE Greek_CI_AS NULL,

[ABCTPRMS] [smallint] NULL,

[ABCTPRMS1] [smallint] NULL,

[ABCMDL] [smallint] NULL,

[INSDATE] [datetime] NULL,

[INSUSER] [smallint] NULL,

[IASACN] [smallint] NOT NULL DEFAULT (0),

[UPDDATE] [datetime] NULL,

[UPDUSER] [smallint] NULL,

[AEM] [varchar](25) COLLATE Greek_CI_AS NULL,

[keypyomd] [smallint] NOT NULL DEFAULT (1),

CONSTRAINT [XP_ACNT] PRIMARY KEY CLUSTERED

(

[ACNT] ASC

) WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY],

CONSTRAINT [XU_ACNT_CODE] UNIQUE NONCLUSTERED

(

```

```
[ACNSHEMA] ASC,  
  
[SODTYPE] ASC,  
  
[CODE] ASC  
  
) WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]  
  
) ON [PRIMARY]  
  
GO  
  
USE [demo_erp]  
  
GO  
  
ALTER TABLE [dbo].[ACNT] WITH NOCHECK ADD CONSTRAINT  
[XF_ACNT_ACNMODEL] FOREIGN KEY ([ACNSHEMA], [ACNMODEL])  
  
REFERENCES [dbo].[ACNMODEL] ([ACNSHEMA], [ACNMODEL])  
  
GO  
  
ALTER TABLE [dbo].[ACNT] CHECK CONSTRAINT [XF_ACNT_ACNMODEL]  
  
GO  
  
ALTER TABLE [dbo].[ACNT] WITH NOCHECK ADD CONSTRAINT  
[XF_ACNT_ACNSHEMA] FOREIGN KEY ([ACNSHEMA])  
  
REFERENCES [dbo].[ACNSHEMA] ([ACNSHEMA])  
  
GO  
  
ALTER TABLE [dbo].[ACNT] CHECK CONSTRAINT [XF_ACNT_ACNSHEMA]  
  
GO  
  
ALTER TABLE [dbo].[ACNT] WITH NOCHECK ADD CONSTRAINT  
[XF_ACNT_ACNTAL] FOREIGN KEY ([ACNTAL])  
  
REFERENCES [dbo].[ACNT] ([ACNT])  
  
GO  
  
ALTER TABLE [dbo].[ACNT] CHECK CONSTRAINT [XF_ACNT_ACNTAL]
```

## - acnbalsheet (πίνακας αποθήκευσης κινήσεων λογαριασμών λογιστικής)

Κάθε συναλλαγή μιας επιχείρησης, αποτυπώνεται σε μια ή περισσότερες κινήσεις ενός ή περισσότερων λογαριασμών λογιστικής. Ο παρακάτω πίνακας, αποθηκεύει τις τελευταίες

```
USE [demo_erp]
```

```
GO
```

```
/***** Object: Table [dbo].[ACNBALSHEET]    Script Date: 11/04/2009  
09:56:11 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[ACNBALSHEET] (  
    [COMPANY] [smallint] NOT NULL,  
    [FISCPRD] [smallint] NOT NULL,  
    [PERIOD] [smallint] NOT NULL,  
    [ACNT] [int] NOT NULL,  
    [LDEBIT] [float] NOT NULL DEFAULT (0),  
    [LCREDIT] [float] NOT NULL DEFAULT (0),  
    [ADEBIT] [float] NOT NULL DEFAULT (0),  
    [ACREDIT] [float] NOT NULL DEFAULT (0),  
    [LDEBITTMP] [float] NOT NULL DEFAULT (0),  
    [LCREDITTMP] [float] NOT NULL DEFAULT (0),  
    [ADEBITTMP] [float] NOT NULL DEFAULT (0),  
    [ACREDITTMP] [float] NOT NULL DEFAULT (0),  
    [LDEBITIAS] [float] NOT NULL DEFAULT (0),
```

```

[LCREDITIAS] [float] NOT NULL DEFAULT (0),
[ADEBITIAS] [float] NOT NULL DEFAULT (0),
[ACREDITIAS] [float] NOT NULL DEFAULT (0),
[LDEBITMPIAS] [float] NOT NULL DEFAULT (0),
[LCREDITMPIAS] [float] NOT NULL DEFAULT (0),
[ADEBITMPIAS] [float] NOT NULL DEFAULT (0),
[ACREDITMPIAS] [float] NOT NULL DEFAULT (0).
CONSTRAINT [XP_ACNBALSHEET] PRIMARY KEY CLUSTERED
(
    [ACNT] ASC,
    [COMPANY] ASC,
    [FISCPRD] ASC,
    [PERIOD] ASC
) WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
USE [demo_erp]
GO
ALTER TABLE [dbo].[ACNBALSHEET] WITH NOCHECK ADD CONSTRAINT
[XD_ACNBALSHEET_ACNT] FOREIGN KEY([ACNT])
REFERENCES [dbo].[ACNT] ([ACNT])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[ACNBALSHEET] CHECK CONSTRAINT
[XD_ACNBALSHEET_ACNT]

```

Από εκεί και πέρα, για την διευκόλυνσή μας κατά την λήψη συγκεντρωτικών στοιχείων, δημιουργήθηκε το ακόλουθο view:

- acnt\_acnbalsheet (συνδυασμός των δύο προηγούμενων πινάκων)

```
USE [demo_erp]
```

```
GO
```

```
/****** Object: View [dbo].[acnt_acnbalsheet] Script Date:
11/04/2008 10:03:47 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE VIEW [dbo].[acnt_acnbalsheet]
```

```
AS
```

```
SELECT      dbo.ACNBALSHEET.FISCPRD, dbo.ACNBALSHEET.PERIOD,
dbo.ACNBALSHEET.COMPANY, dbo.ACNT.CODE, dbo.ACNT.NAME,
            dbo.ACNBALSHEET.LDEBIT,
dbo.ACNBALSHEET.LCREDIT, dbo.ACNBALSHEET.ADEBIT, dbo.ACNBALSHEET.ACREDIT,
dbo.ACNBALSHEET.LDEBITTMP,
```

```
            dbo.ACNBALSHEET.LCREDITTMP,
dbo.ACNBALSHEET.ADEBITTMP, dbo.ACNBALSHEET.ACREDITTMP,
dbo.ACNBALSHEET.LDEBITIAS,
```

```
            dbo.ACNBALSHEET.LCREDITIAS,
dbo.ACNBALSHEET.ADEBITIAS, dbo.ACNBALSHEET.ACREDITIAS,
dbo.ACNBALSHEET.LDEBITMPIAS,
```

```
            dbo.ACNBALSHEET.LCREDITMPIAS,
dbo.ACNBALSHEET.ADEBITMPIAS, dbo.ACNBALSHEET.ACREDITMPIAS
```

```
FROM        dbo.ACNBALSHEET INNER JOIN
```

```
            dbo.ACNT ON dbo.ACNBALSHEET.ACNT =
```

```
dbo.ACNT.ACNT
```

```
GO
```



Αυτά είναι τα κυριότερα μέρη της βάσης δεδομένων που παίρνουν μέρος στην διαδικασία μας. Υπάρχει, φυσικά και ένα ενιαίο business logic, το οποίο διακατέχει όλες τις βάσεις δεδομένων των ERP το οποίο, όπως προαναφέραμε, είναι πέρα από τους σκοπούς της παρούσας μελέτης - και τίθεται εις γνώση της ομάδας ανάπτυξης από τους αρμόδιους προγραμματιστές/παραμετροποιητές των ERP συστημάτων.

#### 4.3 Διαδικτυακός Γεννήτορας Πολλαπλών ERP

Στην ενότητα αυτή, θα παρουσιαστεί η «καρδιά» του όλου συστήματος, δηλαδή η Υπηρεσία Ιστού που αναπτύχθηκε προκειμένου να εκτελέσει όλες τις εργασίες που θα παρουσιαστούν στη συνέχεια. Οι λειτουργίες αυτές είναι μερικές από τις κυριότερες που εκτελεί κάποιος λογιστής ανά τακτά χρονικά διαστήματα. Κατά την παρουσίασή τους, παρατίθενται τα μηνύματα SOAP αίτησης και απάντησης (επεξηγήθηκαν σε προηγούμενη ενότητα) καθώς και ένα ενδεικτικό XML αρχείο που παράγεται από την εκτέλεσή τους. Ο συνολικός κώδικας (σε περιβάλλον Microsoft Studio 2008) σε γλώσσα C#, παρατίθεται στην ενότητα 4.3.3. Στην ενότητα 4.3.2 θα παρουσιαστεί το αρχείο αναπαράστασης της Υπηρεσίας Ιστού, το WSDL αρχείο (επεξηγήθηκε σε προηγούμενη ενότητα).

##### 4.3.1 Λειτουργίες της Υπηρεσίας Ιστού

###### 4.3.1.1 Λειτουργία 1: Αγορές Προϊόντων και Υπηρεσιών (third\_category\_GetProductPurchases)

Η λειτουργία αυτή, υπολογίζει τις αγορές που πραγματοποίησε μια εταιρία, είτε σε προϊόντα είτε σε υπηρεσίες για κάποιο συγκεκριμένο έτος. Δέχεται ως παραμέτρους την εταιρία και το φορολογικό έτος. Τα μηνύματα αίτησης και απάντησης της λειτουργίας αυτής φαίνονται παρακάτω.

###### - Μήνυμα Αίτησης

```
POST /WebSite3/Service.asmx HTTP/1.1
```

```
Host: localhost
```

```
Content-Type: application/soap+xml; charset=utf-8
```

```
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
```

```
<soap12:Body>
```

```
<third_category_GetProductPurchases xmlns="http://tempuri.org/">
```

```
<COMPANY>int</COMPANY>
```

```
<FISCPRD>int</FISCPRD>
```

```

    </third_category_GetProductPurchases>

</soap12:Body>

</soap12:Envelope>

```

- Μήνυμα Απάντησης

HTTP/1.1 200 OK

Content-Type: application/soap+xml; charset=utf-8

Content-Length: length

```

<?xml version="1.0" encoding="utf-8"?>

<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">

  <soap12:Body>

    <third_category_GetProductPurchasesResponse
xmlns="http://tempuri.org/">

      <third_category_GetProductPurchasesResult>

<xsd:schema>schema</xsd:schema><xml/>third_category_GetProductPurchasesResul
t>

    </third_category_GetProductPurchasesResponse>

  </soap12:Body>

</soap12:Envelope>

```

Ένα ενδεικτικό XML αρχείο που παράγεται καλώντας την μέθοδο αυτή φαίνεται παρακάτω.

- Ενδεικτικό XML αρχείο

```

<DataSet>

  <xs:schema id="NewDataSet">

```

```

    <xs:element          name="NewDataSet"          msdata:IsDataSet="true"
msdata:UseCurrentLocale="true">

    <xs:complexType>

    <xs:choice minOccurs="0" maxOccurs="unbounded">

    <xs:element name="Table">

    <xs:complexType>

    <xs:sequence>

    <xs:element name="Μήνας" type="xs:string" minOccurs="0"/>

    <xs:element          name="Αγορές_Εμπορευμάτων"          type="xs:double"
minOccurs="0"/>

    <xs:element name="Αγορές_Υπηρεσιών" type="xs:double" minOccurs="0"/>

    </xs:sequence>

    </xs:complexType>

    </xs:element>

    </xs:choice>

    </xs:complexType>

    </xs:element>

    </xs:schema>

    <diffgr:diffgram>

    <NewDataSet>

    <Table diffgr:id="Table1" msdata:rowOrder="0">

    <Μήνας>Ιανουάριος</Μήνας>

    <Αγορές_Εμπορευμάτων>72591.8</Αγορές_Εμπορευμάτων>

    <Αγορές_Υπηρεσιών>0</Αγορές_Υπηρεσιών>

    </Table>

```

```
<Table diffgr:id="Table2" msdata:rowOrder="1">
<Μήνας>Φεβρουάριος</Μήνας>
<Αγορές_Εμπορευμάτων>49093.600000000006</Αγορές_Εμπορευμάτων>
<Αγορές_Υπηρεσιών>0</Αγορές_Υπηρεσιών>
</Table>
<Table diffgr:id="Table3" msdata:rowOrder="2">
<Μήνας>Μάρτιος</Μήνας>
<Αγορές_Εμπορευμάτων>53247.8</Αγορές_Εμπορευμάτων>
<Αγορές_Υπηρεσιών>0</Αγορές_Υπηρεσιών>
</Table>
<Table diffgr:id="Table4" msdata:rowOrder="3">
<Μήνας>Απρίλιος</Μήνας>
<Αγορές_Εμπορευμάτων>50748.29</Αγορές_Εμπορευμάτων>
<Αγορές_Υπηρεσιών>0</Αγορές_Υπηρεσιών>
</Table>
<Table diffgr:id="Table5" msdata:rowOrder="4">
<Μήνας>Μαΐος</Μήνας>
<Αγορές_Εμπορευμάτων>35166.020000000004</Αγορές_Εμπορευμάτων>
<Αγορές_Υπηρεσιών>0</Αγορές_Υπηρεσιών>
</Table>
<Table diffgr:id="Table6" msdata:rowOrder="5">
<Μήνας>Ιούνιος</Μήνας>
<Αγορές_Εμπορευμάτων>40200.4</Αγορές_Εμπορευμάτων>
```

```

<Αγορές_Υπηρεσιών>0</Αγορές_Υπηρεσιών>
</Table>
<Table diffgr:id="Table7" msdata:rowOrder="6">
<Μήνας>Ιούλιος</Μήνας>
<Αγορές_Εμπορευμάτων>-1000</Αγορές_Εμπορευμάτων>
<Αγορές_Υπηρεσιών>0</Αγορές_Υπηρεσιών>
</Table>
</NewDataSet>
</diffgr:diffgram>
</DataSet>

```

#### 4.3.1.2 Λειτουργία 2: Πωλήσεις Προϊόντων και Υπηρεσιών (third\_category\_GetProductSales)

Η επόμενη λειτουργία που εκτελείται μέσα από την Υπηρεσία Ιστού, είναι ο μηνιαίος υπολογισμός των πωλήσεων προϊόντων και υπηρεσιών της εταιρίας ανά έτος, πληροφορία χρήσιμη είτε για τον επιχειρηματία, είτε για τον εκάστοτε λογιστή. Τα μηνύματα SOAP τα οποία αντιστοιχούν στην αίτηση και στην απάντηση της συγκεκριμένης λειτουργίας φαίνονται παρακάτω. Στη συνέχεια ακολουθεί και ένα ενδεικτικό XML Αρχείο.

##### - Μήνυμα Αίτησης

```
POST /WebSite3/Service.asmx HTTP/1.1
```

```
Host: localhost
```

```
Content-Type: application/soap+xml; charset=utf-8
```

```
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">

```

```
<soap12:Body>
```

```
<third_category_GetProductSales xmlns="http://tempuri.org/">
```

```
<COMPANY>int</COMPANY>
```

```

    <FISCPRD>int</FISCPRD>

  </third_category_GetProductSales>

</soap12:Body>

</soap12:Envelope>

```

- Μήνυμα Απάντησης

HTTP/1.1 200 OK

Content-Type: application/soap+xml; charset=utf-8

Content-Length: length

<?xml version="1.0" encoding="utf-8"?>

```

<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">

```

```

  <soap12:Body>

```

```

    <third_category_GetProductSalesResponse
xmlns="http://tempuri.org/">

```

```

      <third_category_GetProductSalesResult>

```

```

<xsd:schema>schema</xsd:schema>xml</third_category_GetProductSalesResult>

```

```

    </third_category_GetProductSalesResponse>

```

```

  </soap12:Body>

```

```

</soap12:Envelope>

```

- Ενδεικτικό XML αρχείο

```

<DataSet>

```

```

  <xs:schema id="NewDataSet">

```

```

    <xs:element          name="NewDataSet"          msdata:IsDataSet="true"
msdata:UseCurrentLocale="true">

```

```

<xs:complexType>
<xs:choice minOccurs="0" maxOccurs="unbounded">
<xs:element name="Table">
<xs:complexType>
<xs:sequence>
<xs:element name="Μήνας" type="xs:string" minOccurs="0"/>
<xs:element name="Πωλήσεις_Εμπορευμάτων" type="xs:double"
minOccurs="0"/>
<xs:element name="Πωλήσεις_Υπηρεσιών" type="xs:double"
minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>
<diffgr:diffgram>
<NewDataSet>
<Table diffgr:id="Table1" msdata:rowOrder="0">
<Πωλήσεις_Εμπορευμάτων>0</Πωλήσεις_Εμπορευμάτων>
<Πωλήσεις_Υπηρεσιών>0</Πωλήσεις_Υπηρεσιών>
</Table>
<Table diffgr:id="Table2" msdata:rowOrder="1">
<Μήνας>Ιανουάριος</Μήνας>

```

```
<Πωλήσεις_Εμπορευμάτων>98776.34000000004</Πωλήσεις_Εμπορευμάτων>  
  
<Πωλήσεις_Υπηρεσιών>0</Πωλήσεις_Υπηρεσιών>  
  
</Table>  
  
<Table diffgr:id="Table3" msdata:rowOrder="2">  
  
<Μήνας>Φεβρουάριος</Μήνας>  
  
<Πωλήσεις_Εμπορευμάτων>34198.79</Πωλήσεις_Εμπορευμάτων>  
  
<Πωλήσεις_Υπηρεσιών>0</Πωλήσεις_Υπηρεσιών>  
  
</Table>  
  
<Table diffgr:id="Table4" msdata:rowOrder="3">  
  
<Μήνας>Μάρτιος</Μήνας>  
  
<Πωλήσεις_Εμπορευμάτων>38588.82</Πωλήσεις_Εμπορευμάτων>  
  
<Πωλήσεις_Υπηρεσιών>0</Πωλήσεις_Υπηρεσιών>  
  
</Table>  
  
<Table diffgr:id="Table5" msdata:rowOrder="4">  
  
<Μήνας>Απρίλιος</Μήνας>  
  
<Πωλήσεις_Εμπορευμάτων>40047.4899999999991</Πωλήσεις_Εμπορευμάτων>  
  
<Πωλήσεις_Υπηρεσιών>0</Πωλήσεις_Υπηρεσιών>  
  
</Table>  
  
<Table diffgr:id="Table6" msdata:rowOrder="5">  
  
<Μήνας>Μαΐος</Μήνας>  
  
<Πωλήσεις_Εμπορευμάτων>42484.2299999999996</Πωλήσεις_Εμπορευμάτων>  
  
<Πωλήσεις_Υπηρεσιών>0</Πωλήσεις_Υπηρεσιών>
```



```

</Table>

<Table diffgr:id="Table7" msdata:rowOrder="6">
<Μήνας>Ιούνιος</Μήνας>
<Πωλήσεις_Εμπορευμάτων>50756.86</Πωλήσεις_Εμπορευμάτων>
<Πωλήσεις_Υπηρεσιών>0</Πωλήσεις_Υπηρεσιών>
</Table>

<Table diffgr:id="Table8" msdata:rowOrder="7">
<Μήνας>Οκτώβριος</Μήνας>
<Πωλήσεις_Εμπορευμάτων>0</Πωλήσεις_Εμπορευμάτων>
<Πωλήσεις_Υπηρεσιών>0</Πωλήσεις_Υπηρεσιών>
</Table>

</NewDataSet>

</diffgr:diffgram>

</DataSet>

```

#### 4.3.1.3 Λειτουργία 3: Υπολογισμός ετήσιου κέρδους ανά μήνα (third\_category\_get\_profit)

Από τον τίτλο της συγκεκριμένης εργασίας, καταλαβαίνουμε πως αφορά στον υπολογισμό των μηνιαίων κερδών μιας επιχείρησης σε βάθος ενός χρόνου. Ακολουθούν τα μηνύματα SOAP και ένα ενδεικτικό XML.

##### - Μήνυμα Αίτησης

```

POST /WebSite3/Service.asmx HTTP/1.1
Host: localhost

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction: "http://tempuri.org/third_category_get_profit"

<?xml version="1.0" encoding="utf-8"?>

```

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
```

```
<soap:Body>
```

```
<third_category_get_profit xmlns="http://tempuri.org/">
```

```
<COMPANY>int</COMPANY>
```

```
<FISCPRD>int</FISCPRD>
```

```
</third_category_get_profit>
```

```
</soap:Body>
```

```
</soap:Envelope>
```

- Μήνυμα Απάντησης

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

<?xml version="1.0" encoding="utf-8"?>

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
```

```
<soap:Body>
```

```
<third_category_get_profitResponse xmlns="http://tempuri.org/">
```

```
<third_category_get_profitResult>
```

```
<xsd:schema>schema</xsd:schema>xml</third_category_get_profitResult>
```

```
</third_category_get_profitResponse>
```

```
</soap:Body>
```

```
</soap:Envelope>
```

## - Ενδεικτικό XML αρχείο

```

<DataSet>
  <xs:schema id="NewDataSet">
    <xs:element name="NewDataSet" msdata:IsDataSet="true"
msdata:UseCurrentLocale="true">
      <xs:complexType>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element name="Table">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Μήνας" type="xs:string" minOccurs="0"/>
                <xs:element name="Τζίρος_Πωλήσεων" type="xs:double" minOccurs="0"/>
                <xs:element name="Μικτό_Κέρδος" type="xs:double" minOccurs="0"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
  <diffgr:diffgram>
    <NewDataSet>
      <Table diffgr:id="Table1" msdata:rowOrder="0">
        <Τζίρος_Πωλήσεων>0</Τζίρος_Πωλήσεων>

```

<Μικτό\_Κέρδος>0</Μικτό\_Κέρδος>

</Table>

<Table diffgr:id="Table2" msdata:rowOrder="1">

<Μήνας>Ιανουάριος</Μήνας>

<Τζίρος\_Πωλήσεων>98776.3400000004</Τζίρος\_Πωλήσεων>

<Μικτό\_Κέρδος>26184.54000000037</Μικτό\_Κέρδος>

</Table>

<Table diffgr:id="Table3" msdata:rowOrder="2">

<Μήνας>Φεβρουάριος</Μήνας>

<Τζίρος\_Πωλήσεων>34198.79</Τζίρος\_Πωλήσεων>

<Μικτό\_Κέρδος>-14894.81000000005</Μικτό\_Κέρδος>

</Table>

<Table diffgr:id="Table4" msdata:rowOrder="3">

<Μήνας>Μάρτιος</Μήνας>

<Τζίρος\_Πωλήσεων>38588.82</Τζίρος\_Πωλήσεων>

<Μικτό\_Κέρδος>-14658.98000000003</Μικτό\_Κέρδος>

</Table>

<Table diffgr:id="Table5" msdata:rowOrder="4">

<Μήνας>Απρίλιος</Μήνας>

<Τζίρος\_Πωλήσεων>40047.48999999991</Τζίρος\_Πωλήσεων>

<Μικτό\_Κέρδος>-10700.8000000001</Μικτό\_Κέρδος>

</Table>

<Table diffgr:id="Table6" msdata:rowOrder="5">

```

<Μήνας>Μαΐος</Μήνας>
<Τζίρος_Πωλήσεων>42484.2299999999996</Τζίρος_Πωλήσεων>
<Μικτό_Κέρδος>7318.20999999999919</Μικτό_Κέρδος>
</Table>
<Table diffgr:id="Table7" msdata:rowOrder="6">
<Μήνας>Ιούνιος</Μήνας>
<Τζίρος_Πωλήσεων>50756.86</Τζίρος_Πωλήσεων>
<Μικτό_Κέρδος>10556.46</Μικτό_Κέρδος>
</Table>
<Table diffgr:id="Table8" msdata:rowOrder="7">
<Μήνας>Οκτώβριος</Μήνας>
<Τζίρος_Πωλήσεων>0</Τζίρος_Πωλήσεων>
<Μικτό_Κέρδος>0</Μικτό_Κέρδος>
</Table>
</NewDataSet>
</diffgr:diffgram>
</DataSet>

```

#### 4.3.1.4 Λειτουργία 4: Υπολογισμός εισφορών προς τα ασφαλιστικά ταμεία ανά μήνα (third\_category\_get\_public\_insurance\_obligations)

Η λειτουργία αυτή, υπολογίζει τις μηνιαίες εισφορές της επιχείρησης προς τα ασφαλιστικά ταμεία (ΙΚΑ κτλ).

##### - Μήνυμα Αίτησης

```

POST /WebSite3/Service.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8

```

Content-Length: length

SOAPAction:

"http://tempuri.org/third\_category\_get\_public\_insurance\_obligations"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
 xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<third\_category\_get\_public\_insurance\_obligations  
 xmlns="http://tempuri.org/">

<COMPANY>int</COMPANY>

<FISCPRD>int</FISCPRD>

</third\_category\_get\_public\_insurance\_obligations>

</soap:Body>

</soap:Envelope>

- **Μήνυμα Απάντησης**

HTTP/1.1 200 OK

Content-Type: text/xml; charset=utf-8

Content-Length: length

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
 xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

<third\_category\_get\_public\_insurance\_obligationsResponse  
 xmlns="http://tempuri.org/">

<third\_category\_get\_public\_insurance\_obligationsResult>

```
<xsd:schema>schema</xsd:schema>xml</third_category_get_public_insurance_obli
gationsResult>
```

```
</third_category_get_public_insurance_obligationsResponse>
```

```
</soap:Body>
```

```
</soap:Envelope>
```

- Ενδεικτικό XML αρχείο

```
<DataSet>
```

```
<xs:schema id="NewDataSet">
```

```
<xs:element name="NewDataSet" msdata:IsDataSet="true"
msdata:UseCurrentLocale="true">
```

```
<xs:complexType>
```

```
<xs:choice minOccurs="0" maxOccurs="unbounded">
```

```
<xs:element name="Table">
```

```
<xs:complexType>
```

```
<xs:sequence>
```

```
<xs:element name="PERIOD" type="xs:short" minOccurs="0"/>
```

```
<xs:element name="μηνιαίο_IKA" type="xs:double" minOccurs="0"/>
```

```
<xs:element name="FISCPRD" type="xs:short" minOccurs="0"/>
```

```
<xs:element name="COMPANY" type="xs:short" minOccurs="0"/>
```

```
<xs:element name="Minas" type="xs:string" minOccurs="0"/>
```

```
</xs:sequence>
```

```
</xs:complexType>
```

```
</xs:element>
```

```
</xs:choice>
```

```

</xs:complexType>

</xs:element>

</xs:schema>

<diffgr:diffgram/>

</DataSet>

```

Στο παραπάνω XML αρχείο, παρατηρούμε μόνο το σχήμα του XML αρχείου, καθώς στη συγκεκριμένη βάση δεν υπάρχουν σχετικά δεδομένα.

#### 4.3.1.5 Λειτουργία 5: Υπολογισμός μηνιαίας απόδοσης του ΦΠΑ (third\_category\_get\_public\_tax\_obligations)

Η συγκεκριμένη λειτουργία, υπολογίζει τις μηνιαίες υποχρεώσεις της εταιρίας προς την εφορία (αυτές που αφορούν τον ΦΠΑ τουλάχιστον). Η πληροφορία αυτή είναι ακόμα πιο χρήσιμη για ένα λογιστή, όταν πλησιάζει η μέρα απόδοσης του ποσού.

##### - Μήνυμα Αίτησης

```

POST /WebSite3/Service.asmx HTTP/1.1

Host: localhost

Content-Type: text/xml; charset=utf-8

Content-Length: length

SOAPAction:
"http://tempuri.org/third_category_get_public_tax_obligations"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

  <soap:Body>

    <third_category_get_public_tax_obligations
xmlns="http://tempuri.org/">

      <COMPANY>int</COMPANY>

      <FISCPRD>int</FISCPRD>

    </third_category_get_public_tax_obligations>

```



```
</soap:Body>
```

```
</soap:Envelope>
```

- Μήνυμα Απάντησης

```
HTTP/1.1 200 OK
```

```
Content-Type: text/xml; charset=utf-8
```

```
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
```

```
<soap:Body>
```

```
<third_category_get_public_tax_obligationsResponse
xmlns="http://tempuri.org/">
```

```
<third_category_get_public_tax_obligationsResult>
```

```
<xsd:schema>schema</xsd:schema>xml</third_category_get_public_tax_obligatio
nsResult>
```

```
</third_category_get_public_tax_obligationsResponse>
```

```
</soap:Body>
```

```
</soap:Envelope>
```

- Ενδεικτικό XML αρχείο

```
<DataSet>
```

```
<xs:schema id="NewDataSet">
```

```
<xs:element name="NewDataSet" msdata:IsDataSet="true"
msdata:UseCurrentLocale="true">
```

```
<xs:complexType>
```

```
<xs:choice minOccurs="0" maxOccurs="unbounded">
```

```
<xs:element name="Table">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Μήνας" type="xs:string" minOccurs="0"/>
      <xs:element name="Μηνιαίο_ΦΠΑ" type="xs:double" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>
<diffgr:diffgram>
  <NewDataSet>
    <Table diffgr:id="Table1" msdata:rowOrder="0">
      <Μήνας>Ιανουάριος</Μήνας>
      <Μηνιαίο_ΦΠΑ>4975.2699999999822</Μηνιαίο_ΦΠΑ>
    </Table>
    <Table diffgr:id="Table2" msdata:rowOrder="1">
      <Μήνας>Φεβρουάριος</Μήνας>
      <Μηνιαίο_ΦΠΑ>-2829.8400000000029</Μηνιαίο_ΦΠΑ>
    </Table>
    <Table diffgr:id="Table3" msdata:rowOrder="2">
```

```
<Μήνας>Μάρτιος</Μήνας>  
<Μηνιαίο_ΦΠΑ>-2785.010000000002</Μηνιαίο_ΦΠΑ>  
</Table>  
<Table diffgr:id="Table4" msdata:rowOrder="3">  
<Μήνας>Απρίλιος</Μήνας>  
<Μηνιαίο_ΦΠΑ>-2032.9300000000058</Μηνιαίο_ΦΠΑ>  
</Table>  
<Table diffgr:id="Table5" msdata:rowOrder="4">  
<Μήνας>Μαΐος</Μήνας>  
<Μηνιαίο_ΦΠΑ>1390.6699999999983</Μηνιαίο_ΦΠΑ>  
</Table>  
<Table diffgr:id="Table6" msdata:rowOrder="5">  
<Μήνας>Ιούνιος</Μήνας>  
<Μηνιαίο_ΦΠΑ>2005.8599999999942</Μηνιαίο_ΦΠΑ>  
</Table>  
<Table diffgr:id="Table7" msdata:rowOrder="6">  
<Μήνας>Ιούλιος</Μήνας>  
<Μηνιαίο_ΦΠΑ>190</Μηνιαίο_ΦΠΑ>  
</Table>  
<Table diffgr:id="Table8" msdata:rowOrder="7">  
<Μήνας>Οκτώβριος</Μήνας>  
<Μηνιαίο_ΦΠΑ>0</Μηνιαίο_ΦΠΑ>  
</Table>
```

```

</NewDataSet>

</diffgr:diffgram>

</DataSet>

```

#### 4.3.2 Κώδικας της Υπηρεσίας Ιστού

Κατά την ανάπτυξη της Υπηρεσίας Ιστού δημιουργηθήκαν δύο κλάσεις. Η μία (SqlHelper), βοηθά στην εκτέλεση ερωτημάτων σε μια βάση δεδομένων SQL Server. Η άλλη είναι η κύρια κλάση της Υπηρεσίας Ιστού (Service) που είναι υπεύθυνη να εκτελέσει τις πέντε λειτουργίες που παρουσιάστηκαν προηγουμένως.

##### - Η κλάση SQLHelper

```

using System;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

/// <summary>

/// Summary description for SqlHelper

/// </summary>

public class SqlHelper

{

```

```
private static string strConn;

static SqlHelper()

{

    strConn = "Data Source=ANONYMOUS;Initial
Catalog=demo_erp;User ID=softone;Password=softone";

    //
ConfigurationManager.ConnectionStrings["connectionstring"].ConnectionString
;

}

public static int ExecuteNonQuery(string sql, SqlParameter[] p)

{

    SqlConnection cnn = new SqlConnection(strConn);

    SqlCommand cmd = new SqlCommand(sql, cnn);

    for (int i = 0; i < p.Length; i++)

    {

        cmd.Parameters.Add(p[i]);

    }

    cnn.Open();

    int retval = cmd.ExecuteNonQuery();

    cnn.Close();

    return retval;

}

public static object ExecuteScalar(string sql, SqlParameter[] p)

{

    SqlConnection cnn = new SqlConnection(strConn);
```

```
SqlCommand cmd = new SqlCommand(sql, cnn);

for (int i = 0; i < p.Length; i++)

{

    cmd.Parameters.Add(p[i]);

}

cnn.Open();

object obj = cmd.ExecuteScalar();

cnn.Close();

return obj;

}

public static DataSet GetDataSet(string sql, SqlParameter[] p)

{

    SqlConnection cnn = new SqlConnection(strConn);

    SqlCommand cmd = new SqlCommand(sql, cnn);

    if (p != null)

    {

        for (int i = 0; i < p.Length; i++)

        {

            cmd.Parameters.Add(p[i]);

        }

    }

    SqlDataAdapter da = new SqlDataAdapter();

    da.SelectCommand = cmd;
```

```

        DataSet ds = new DataSet();

        da.Fill(ds);

        return ds;

    }
}

```

- Η κλάση Service

```

using System;

using System.Linq;

using System.Web;

using System.Web.Services;

using System.Web.Services.Protocols;

using System.Xml.Linq;

using System.Data;

using System.Data.SqlClient;

[WebService(Namespace = "http://tempuri.org/")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]

// To allow this Web Service to be called from script, using ASP.NET
AJAX, uncomment the following line.

// [System.Web.Script.Services.ScriptService]

public class Service : System.Web.Services.WebService

{

    public Service () {

        //Uncomment the following line if using designed components

        //InitializeComponent();
    }
}

```

```

    ]

    #region Softone Web Methods

    #region product sales

    [WebMethod(Description = " Returns all sales of products for the
    selected company and fiscal year")]

    public DataSet third_category_GetProductSales(int COMPANY, int
    FISCPRD)

    {

        string sql = "SELECT      TOP (100) PERCENT "

            + " CASE acnt_comp_Period fiscper.PERIOD WHEN 1
    THEN 'Ιανουάριος' WHEN 2 THEN 'Φεβρουάριος' WHEN 3 THEN 'Μάρτιος' WHEN 4
    THEN 'Απρίλιος' WHEN "

            + " 5 THEN 'Μάιος' WHEN 6 THEN 'Ιούνιος' WHEN 7
    THEN 'Ιούλιος' WHEN 8 THEN 'Αύγουστος' WHEN 9 THEN 'Σεπτέμβριος' WHEN 10
    THEN 'Οκτώβριος' WHEN "

            + " 11 THEN 'Νοέμβριος' WHEN 12 THEN
    'Δεκέμβριος' END AS Μήνας,
    ISNULL(dbo.acnt_Poiliseis_Eporeymaton.Poilisies_Emporeymaton, 0) "

            + " AS Πωλήσεις_Εμπορευμάτων,
    ISNULL(dbo.acnt_Poilisies_Ypiresion.Poilisies_Ypiresion, 0) AS
    Πωλήσεις_Υπηρεσιών "

            + " FROM      dbo.acnt_comp_period_fiscper

    LEFT OUTER JOIN "

            + "  dbo.acnt_Agores_Eporeymaton ON
    dbo.acnt_comp_period_fiscper.PERIOD = dbo.acnt_Agores_Eporeymaton.PERIOD
    AND "

            + "  dbo.acnt_comp_period_fiscper.FISCPRD =
    dbo.acnt_Agores_Eporeymaton.FISCPRD AND "

            + "  dbo.acnt_comp_period_fiscper.COMPANY =
    dbo.acnt_Agores_Eporeymaton.COMPANY LEFT OUTER JOIN "

            + "  dbo.acnt_Agores_Ypiresion ON
    dbo.acnt_comp_period_fiscper.PERIOD = dbo.acnt_Agores_Ypiresion.PERIOD AND
    "
    }

```



```

        + " dbo.acnt_comp_period_fiscper.FISCPRD =
dbo.acnt_Agores_Ypiresion.FISCPRD AND "

        + " dbo.acnt_comp_period_fiscper.COMPANY =
dbo.acnt_Agores_Ypiresion.COMPANY LEFT OUTER JOIN "

        + " dbo.acnt_Policies_Ypiresion ON
dbo.acnt_comp_period_fiscper.PERIOD = dbo.acnt_Policies_Ypiresion.PERIOD
AND "

        + " dbo.acnt_comp_period_fiscper.FISCPRD =
dbo.acnt_Policies_Ypiresion.FISCPRD AND "

        + " dbo.acnt_comp_period_fiscper.COMPANY =
dbo.acnt_Policies_Ypiresion.COMPANY LEFT OUTER JOIN "

        + " dbo.acnt_Poiliseis_Eporeymaton ON
dbo.acnt_comp_period_fiscper.COMPANY =
dbo.acnt_Poiliseis_Eporeymaton.COMPANY AND "

        + " dbo.acnt_comp_period_fiscper.FISCPRD =
dbo.acnt_Poiliseis_Eporeymaton.FISCPRD AND "

        + " dbo.acnt_comp_period_fiscper.PERIOD =
dbo.acnt_Poiliseis_Eporeymaton.PERIOD "

        + " Where acnt_Poiliseis_Eporeymaton.COMPANY =
@COMPANY AND acnt_Poiliseis_Eporeymaton.FISCPRD = @FISCPRD "

        + " ORDER BY
dbo.acnt_comp_period_fiscper.FISCPRD, dbo.acnt_comp_period_fiscper.PERIOD";

    SqlParameter[] p = new SqlParameter[2];

    p[0] = new SqlParameter("@COMPANY", COMPANY);
    p[1] = new SqlParameter("@FISCPRD", FISCPRD);

    DataSet ds = SqlHelper.GetDataSet(sql, p);

    return ds;

}

#endregion

#region product purchases

```

[WebMethod(Description = " Returns all purchases of products for the selected company and fiscal year")]

```
public DataSet third_category_GetProductPurchases(int COMPANY,int
FISCPRD)
```

```
{
```

```
    string sql = "SELECT      TOP (100) PERCENT "
```

```
                + " CASE acnt_comp_period_fiscper.PERIOD WHEN 1
THEN 'Ιανουάριος' WHEN 2 THEN 'Φεβρουάριος' WHEN 3 THEN 'Μάρτιος' WHEN 4
THEN 'Απρίλιος' WHEN "
```

```
                + " 5 THEN 'Μάιος' WHEN 6 THEN 'Ιούνιος' WHEN 7
THEN 'Ιούλιος' WHEN 8 THEN 'Αύγουστος' WHEN 9 THEN 'Σεπτέμβριος' WHEN 10
THEN 'Οκτώβριος' WHEN "
```

```
                + " 11 THEN 'Νοέμβριος' WHEN 12 THEN
'Δεκεμβριος' END AS Μηνας, "
```

```
                + "
ISNULL(dbo.acnt_Agores_Eporeymaton.Agores_Emporeymaton, 0) AS
Αγορές_Εμπορευμάτων, ISNULL(dbo.acnt_Agores_Ypiresion.Agores_Ypiresion, "
```

```
                + " 0) AS Αγορές_Υπηρεσιών"
```

```
                + " FROM      dbo.acnt_comp_period_fiscper
LEFT OUTER JOIN "
```

```
                + "  dbo.acnt_Agores_Eporeymaton ON
dbo.acnt_comp_period_fiscper.PERIOD = dbo.acnt_Agores_Eporeymaton.PERIOD
AND "
```

```
                + "  dbo.acnt_comp_period_fiscper.FISCPRD =
dbo.acnt_Agores_Eporeymaton.FISCPRD AND "
```

```
                + "  dbo.acnt_comp_period_fiscper.COMPANY =
dbo.acnt_Agores_Eporeymaton.COMPANY LEFT OUTER JOIN "
```

```
                + "  dbo.acnt_Agores_Ypiresion ON
dbo.acnt_comp_period_fiscper.PERIOD = dbo.acnt_Agores_Ypiresion.PERIOD AND
"
```

```
                + "  dbo.acnt_comp_period_fiscper.FISCPRD =
dbo.acnt_Agores_Ypiresion.FISCPRD AND "
```

```
                + "  dbo.acnt_comp_period_fiscper.COMPANY =
dbo.acnt_Agores_Ypiresion.COMPANY LEFT OUTER JOIN "
```

```

        + " dbo.acnt_Polisies_Ypiresion ON
dbo.acnt_comp_period_fiscper.PERIOD = dbo.acnt_Polisies_Ypiresion.PERIOD
AND "

        + " dbo.acnt_comp_period_fiscper.FISCPRD =
dbo.acnt_Polisies_Ypiresion.FISCPRD AND "

        + " dbo.acnt_comp_period_fiscper.COMPANY =
dbo.acnt_Polisies_Ypiresion.COMPANY LEFT OUTER JOIN "

        + " dbo.acnt_Poiliseis_Eporeymaton ON
dbo.acnt_comp_period_fiscper.COMPANY =
dbo.acnt_Poiliseis_Eporeymaton.COMPANY AND "

        + " dbo.acnt_comp_period_fiscper.FISCPRD =
dbo.acnt_Poiliseis_Eporeymaton.FISCPRD AND "

        + " dbo.acnt_comp_period_fiscper.PERIOD =
dbo.acnt_Poiliseis_Eporeymaton.PERIOD "

        + " where acnt_Agores_Eporeymaton.COMPANY =
@COMPANY AND acnt_Agores_Eporeymaton.FISCPRD = @FISCPRD "

        + " ORDER BY
dbo.acnt_comp_period_fiscper.FISCPRD, dbo.acnt_comp_period_fiscper.PERIOD
";

```

```
SqlParameter[] p = new SqlParameter[2];
```

```
p[0] = new SqlParameter("@COMPANY", COMPANY);
```

```
p[1] = new SqlParameter("@FISCPRD", FISCPRD);
```

```
DataSet ds = SqlHelper.GetDataSet(sql, p);
```

```
return ds;
```

```
)
# endregion
```

```
#region public insurance organizations
```

```
[WebMethod(Description = " Returns monthly obligations of the
selected company against the public insurance organizations")]
```

```
public DataSet
third_category_get_public_insurance_obligations(int COMPANY, int FISCPRD)
```

```

{
    string sql = " SELECT      TOP (100) PERCENT
dbo.ACNBALSHEET.PERIOD, SUM(dbo.ACNBALSHEET.LDEBIT) AS μηνιαίο_ΙΚΑ,
dbo.ACNBALSHEET.FISCPRD, "

        +" dbo.ACNBALSHEET.COMPANY, "

        +" CASE ACNBALSHEET.PERIOD WHEN 1 THEN
'Ιανουάριος' WHEN 2 THEN 'Φεβρουάριος' WHEN 3 THEN 'Μάρτιος' WHEN 4 THEN
'Απρίλιος' WHEN 5 THEN "

        +" 'Μάιος' WHEN 6 THEN 'Ιούνιος' WHEN 7 THEN
'Ιούλιος' WHEN 8 THEN 'Αύγουστος' WHEN 9 THEN 'Σεπτέμβριος' WHEN 10 THEN
'Οκτώβριος' WHEN 11 THEN "

        +" 'Νοέμβριος' WHEN 12 THEN 'Δεκεμβριος' END AS
Μηνιαίο "

        +" FROM      dbo.ACNBALSHEET INNER JOIN "

        +" dbo.ACNT ON dbo.ACNBALSHEET.ACNT =
dbo.ACNT.ACNT "

        +" WHERE      (dbo.ACNT.CODE LIKE '55.00*') "

        +" AND      dbo.ACNBALSHEET.FISCPRD=@FISCPRD and
dbo.ACNBALSHEET.COMPANY=@COMPANY"

        +" GROUP BY dbo.ACNBALSHEET.PERIOD,
dbo.ACNBALSHEET.FISCPRD, dbo.ACNBALSHEET.COMPANY "

        +" ORDER BY dbo.ACNBALSHEET.PERIOD";

    SqlParameter[] p = new SqlParameter[2];
    p[0] = new SqlParameter("@COMPANY", COMPANY);
    p[1] = new SqlParameter("@FISCPRD", FISCPRD);

    DataSet ds = SqlHelper.GetDataSet(sql, p);

    return ds;
}

#endregion

```

```

#region public tax organizations

[WebMethod(Description = " Returns monthly obligations of the
selected company against the public tax organizations")]

public DataSet third_category_get_public_tax_obligations(int
COMPANY, int FISCPRD)

{

    string sql = " SELECT      TOP (100) PERCENT "

        + " CASE ACNBALSHEET.PERIOD WHEN 1 THEN
'Ιανουάριος' WHEN 2 THEN 'Φεβρουάριος' WHEN 3 THEN 'Μάρτιος' WHEN 4 THEN
'Απρίλιος' WHEN 5 THEN "

        + " 'Μάιος' WHEN 6 THEN 'Ιούνιος' WHEN 7 THEN
'Ιούλιος' WHEN 8 THEN 'Αυγούστος' WHEN 9 THEN 'Σεπτέμβριος' WHEN 10 THEN
'Οκτώβριος' WHEN 11 THEN "

        + " 'Νοέμβριος' WHEN 12 THEN 'Δεκέμβριος' END AS
Μήνας, "

        + " SUM(dbo.ACNBALSHEET.LCREDIT) -
SUM(dbo.ACNBALSHEET.LDEBIT) AS Μηνιαίο_ΦΠΑ "

        + " FROM      dbo.ACNBALSHEET INNER JOIN "

        + " dbo.ACNT ON dbo.ACNBALSHEET.ACNT =
dbo.ACNT.ACNT "

        + " WHERE      (dbo.ACNT.CODE LIKE '54.00.%') "

        + " AND      dbo.ACNBALSHEET.FISCPRD=@FISCPRD and
dbo.ACNBALSHEET.COMPANY=@COMPANY"

        + " GROUP BY dbo.ACNBALSHEET.PERIOD,
dbo.ACNBALSHEET.FISCPRD, dbo.ACNBALSHEET.COMPANY "

        + " ORDER BY dbo.ACNBALSHEET.PERIOD ";

    SqlParameter[] p = new SqlParameter[2];

    p[0] = new SqlParameter("@COMPANY", COMPANY);

    p[1] = new SqlParameter("@FISCPRD", FISCPRD);

    DataSet ds = SqlHelper.GetDataSet(sql, p);

```

```

return ds;

}

#endregion

#region profits

[WebMethod(Description = " Returns all profits for the selected
company and fiscal year")]

public DataSet third_category_get_profit(int COMPANY, int
FISCPERD)
{
    string sql = "SELECT      TOP (100) PERCENT "
        + " CASE acnt_comp_period_fiscper.PERIOD WHEN 1
THEN 'Ιανουάριος' WHEN 2 THEN 'Φεβρουάριος' WHEN 3 THEN 'Μάρτιος' WHEN 4
THEN 'Απρίλιος' WHEN "
        + " 5 THEN 'Μάιος' WHEN 6 THEN 'Ιούνιος' WHEN 7
THEN 'Ιούλιος' WHEN 8 THEN 'Αύγουστος' WHEN 9 THEN 'Σεπτέμβριος' WHEN 10
THEN 'Οκτώβριος' WHEN "
        + " 11 THEN 'Νοέμβριος' WHEN 12 THEN
'Δεκέμβριος' END AS Μήνας,
ISNULL(dbo.acnt_Polisies_Eporeymaton.Polisies_Eporeymaton, 0) "
        + "+
ISNULL(dbo.acnt_Polisies_Ypiresion.Polisies_Ypiresion, 0) AS
Τζίρος_Πωλήσεων,
(ISNULL(dbo.acnt_Polisies_Eporeymaton.Polisies_Eporeymaton, "
        + " 0) +
ISNULL(dbo.acnt_Polisies_Ypiresion.Polisies_Ypiresion, 0)) -
(ISNULL(dbo.acnt_Agores_Eporeymaton.Agores_Eporeymaton, 0) "
        + " +
ISNULL(dbo.acnt_Agores_Ypiresion.Agores_Ypiresion, 0)) AS Μικτό_Κέρδος "
        + " FROM      dbo.acnt_comp_period_fiscper
LEFT OUTER JOIN "
        + " dbo.acnt_Agores_Eporeymaton ON
dbo.acnt_comp_period_fiscper.PERIOD = dbo.acnt_Agores_Eporeymaton.PERIOD
AND "

```

```

        + " dbo.acnt_comp_period_fiscper.FISCPRD =
dbo.acnt_Agores_Eporeymaton.FISCPRD AND "

        + " dbo.acnt_comp_period_fiscper.COMPANY =
dbo.acnt_Agores_Eporeymaton.COMPANY LEFT OUTER JOIN "

        + " dbo.acnt_Agores_Ypiresion ON
dbo.acnt_comp_period_fiscper.PERIOD = dbo.acnt_Agores_Ypiresion.PERIOD AND
"

        + " dbo.acnt_comp_period_fiscper.FISCPRD =
dbo.acnt_Agores_Ypiresion.FISCPRD AND "

        + " dbo.acnt_comp_period_fiscper.COMPANY =
dbo.acnt_Agores_Ypiresion.COMPANY LEFT OUTER JOIN "

        + " dbo.acnt_Poliseis_Ypiresion ON
dbo.acnt_comp_period_fiscper.PERIOD = dbo.acnt_Poliseis_Ypiresion.PERIOD
AND "

        + " dbo.acnt_comp_period_fiscper.FISCPRD =
dbo.acnt_Poliseis_Ypiresion.FISCPRD AND "

        + " dbo.acnt_comp_period_fiscper.COMPANY =
dbo.acnt_Poliseis_Ypiresion.COMPANY LEFT OUTER JOIN "

        + " dbo.acnt_Poiliseis_Eporeymaton ON
dbo.acnt_comp_period_fiscper.COMPANY =
dbo.acnt_Poiliseis_Eporeymaton.COMPANY AND "

        + " dbo.acnt_comp_period_fiscper.FISCPRD =
dbo.acnt_Poiliseis_Eporeymaton.FISCPRD AND "

        + " dbo.acnt_comp_period_fiscper.PERIOD =
dbo.acnt_Poiliseis_Eporeymaton.PERIOD "

        + " where acnt_Poiliseis_Eporeymaton.COMPANY =
@COMPANY AND acnt_Poiliseis_Eporeymaton.FISCPRD = @FISCPRD "

        + " ORDER BY
dbo.acnt_comp_period_fiscper.FISCPRD, dbo.acnt_comp_period_fiscper.PERIOD
";

```

```
SqlParameter[] p = new SqlParameter[2];
```

```
p[0] = new SqlParameter("@COMPANY", COMPANY);
```

```
p[1] = new SqlParameter("@FISCPRD", FISCPRD);
```

```
DataSet ds = SqlHelper.GetDataSet(sql, p);
```

```

    return ds;
}

#endregion

#endregion

#region Singular Logic Control Web Methods

#endregion

```

#### 4.3.2 Το αρχείο περιγραφής της Υπηρεσίας Ιστού (WSDL)

Ακολουθεί το XML αρχείο που περιγράφει τη δομή και τις λειτουργίες της Υπηρεσίας Ιστού που αναπτύχθηκε (WSDL προδιαγραφή).

```

<?xml version="1.0" encoding="utf-8"?>

<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns="http://tempuri.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  targetNamespace="http://tempuri.org/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"

  <wsdl:types>

    <s:schema elementFormDefault="qualified"
      xmlns="http://tempuri.org/">

      <s:element name="third_category_GetProductSales">

        <s:complexType>

          <s:sequence>

            <s:element minOccurs="1" maxOccurs="1" name="COMPANY"
              type="s:int" />

            <s:element minOccurs="1" maxOccurs="1" name="FISCPRD"
              type="s:int" />

          </s:sequence>

        </s:complexType>

```



```
</s:element>

<s:element name="third_category_GetProductSalesResponse">

  <s:complexType>

    <s:sequence>

      <s:element minOccurs="0" maxOccurs="1"
name="third_category_GetProductSalesResult">

        <s:complexType>

          <s:sequence>

            <s:element ref="s:schema" />

            <s:any />

          </s:sequence>

        </s:complexType>

      </s:element>

    </s:sequence>

  </s:complexType>

</s:element>

<s:element name="third_category_GetProductPurchases">

  <s:complexType>

    <s:sequence>

      <s:element minOccurs="1" maxOccurs="1" name="COMPANY"
type="s:int" />

      <s:element minOccurs="1" maxOccurs="1" name="FISCPRD"
type="s:int" />

    </s:sequence>

  </s:complexType>


```

```
</s:element>

<s:element name="third_category_GetProductPurchasesResponse">

  <s:complexType>

    <s:sequence>

      <s:element minOccurs="0" maxOccurs="1"
name="third_category_GetProductPurchasesResult">

        <s:complexType>

          <s:sequence>

            <s:element ref="s:schema" />

            <s:any />

          </s:sequence>

        </s:complexType>

      </s:element>

    </s:sequence>

  </s:complexType>

</s:element>

<s:element
name="third_category_get_public_insurance_obligations">

  <s:complexType>

    <s:sequence>

      <s:element minOccurs="1" maxOccurs="1" name="COMPANY"
type="s:int" />

      <s:element minOccurs="1" maxOccurs="1" name="FISCPRD"
type="s:int" />

    </s:sequence>

  </s:complexType>


```

```
</s:element>

<s:element
name="third_category_get_public_insurance_obligationsResponse">

  <s:complexType>

    <s:sequence>

      <s:element minOccurs="0" maxOccurs="1"
name="third_category_get_public_insurance_obligationsResult">

        <s:complexType>

          <s:sequence>

            <s:element ref="s:schema" />

            <s:any />

          </s:sequence>

        </s:complexType>

      </s:element>

    </s:sequence>

  </s:complexType>

</s:element>

<s:element name="third_category_get_public_tax_obligations">

  <s:complexType>

    <s:sequence>

      <s:element minOccurs="1" maxOccurs="1" name="COMPANY"
type="s:int" />

      <s:element minOccurs="1" maxOccurs="1" name="FISCPRD"
type="s:int" />

    </s:sequence>

  </s:complexType>


```

```

    </s:element>

    <s:element
name="third_category_get_public_tax_obligationsResponse">

    <s:complexType>

    <s:sequence>

    <s:element minOccurs="0" maxOccurs="1"
name="third_category_get_public_tax_obligationsResult">

    <s:complexType>

    <s:sequence>

    <s:element ref="s:schema" />

    <s:any />

    </s:sequence>

    </s:complexType>

    </s:element>

    </s:sequence>

    </s:complexType>

</s:element>

<s:element name="third_category_get_profit">

    <s:complexType>

    <s:sequence>

    <s:element minOccurs="1" maxOccurs="1" name="COMPANY"
type="s:int" />

    <s:element minOccurs="1" maxOccurs="1" name="FISCPRD"
type="s:int" />

    </s:sequence>

    </s:complexType>

```

```

</s:element>

<s:element name="third_category_get_profitResponse">
  <s:complexType>
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="1"
name="third_category_get_profitResult">
        <s:complexType>
          <s:sequence>
            <s:element ref="s:schema" />
            <s:any />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:sequence>
  </s:complexType>
</s:element>
</s:schema>
</wsdl:types>
<wsdl:message name="third_category_GetProductSalesSoapIn">
  <wsdl:part name="parameters"
element="tns:third_category_GetProductSales" />
</wsdl:message>
<wsdl:message name="third_category_GetProductSalesSoapOut">
  <wsdl:part name="parameters"
element="tns:third_category_GetProductSalesResponse" />

```

```
</wsdl:message>

<wsdl:message name="third_category_GetProductPurchasesSoapIn">
  <wsdl:part name="parameters"
  element="tns:third_category_GetProductPurchases" />
</wsdl:message>

<wsdl:message name="third_category_GetProductPurchasesSoapOut">
  <wsdl:part name="parameters"
  element="tns:third_category_GetProductPurchasesResponse" />
</wsdl:message>

<wsdl:message
name="third_category_get_public_insurance_obligationsSoapIn">
  <wsdl:part name="parameters"
  element="tns:third_category_get_public_insurance_obligations" />
</wsdl:message>

<wsdl:message
name="third_category_get_public_insurance_obligationsSoapOut">
  <wsdl:part name="parameters"
  element="tns:third_category_get_public_insurance_obligationsResponse" />
</wsdl:message>

<wsdl:message
name="third_category_get_public_tax_obligationsSoapIn">
  <wsdl:part name="parameters"
  element="tns:third_category_get_public_tax_obligations" />
</wsdl:message>

<wsdl:message
name="third_category_get_public_tax_obligationsSoapOut">
  <wsdl:part name="parameters"
  element="tns:third_category_get_public_tax_obligationsResponse" />
</wsdl:message>
```

```

    <wsdl:message name="third_category_get_profitSoapIn">
      <wsdl:part name="parameters"
        element="tns:third_category_get_profit" />
    </wsdl:message>

    <wsdl:message name="third_category_get_profitSoapOut">
      <wsdl:part name="parameters"
        element="tns:third_category_get_profitResponse" />
    </wsdl:message>

    <wsdl:portType name="ServiceSoap">
      <wsdl:operation name="third_category_GetProductSales">
        <wsdl:documentation
          xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"> Returns all sales of
          products for the selected company and fiscal year</wsdl:documentation>
        <wsdl:input message="tns:third_category_GetProductSalesSoapIn"
          />
        <wsdl:output
          message="tns:third_category_GetProductSalesSoapOut" />
        </wsdl:operation>

        <wsdl:operation name="third_category_GetProductPurchases">
          <wsdl:documentation
            xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"> Returns all purchases of
            products for the selected company and fiscal year</wsdl:documentation>
          <wsdl:input
            message="tns:third_category_GetProductPurchasesSoapIn" />
          <wsdl:output
            message="tns:third_category_GetProductPurchasesSoapOut" />
          </wsdl:operation>

          <wsdl:operation
            name="third_category_get_public_insurance_obligations">
            <wsdl:documentation
              xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"> Returns monthly obligations

```

of the selected company against the public insurance organizations</wsdl:documentation>

```
<wsdl:input
  xmlns:tns="tns:third_category_get_public_insurance_obligationsSoapIn" />
```

```
<wsdl:output
  xmlns:tns="tns:third_category_get_public_insurance_obligationsSoapOut" />
```

```
</wsdl:operation>
```

```
<wsdl:operation name="third_category_get_public_tax_obligations">
```

```
<wsdl:documentation
  xmlns:tns="http://schemas.xmlsoap.org/wsdl/"> Returns monthly obligations
  of the selected company against the public tax
  organizations</wsdl:documentation>
```

```
<wsdl:input
  xmlns:tns="tns:third_category_get_public_tax_obligationsSoapIn" />
```

```
<wsdl:output
  xmlns:tns="tns:third_category_get_public_tax_obligationsSoapOut" />
```

```
</wsdl:operation>
```

```
<wsdl:operation name="third_category_get_profit">
```

```
<wsdl:documentation
  xmlns:tns="http://schemas.xmlsoap.org/wsdl/"> Returns all profits for the
  selected company and fiscal year</wsdl:documentation>
```

```
<wsdl:input message="tns:third_category_get_profitSoapIn" />
```

```
<wsdl:output message="tns:third_category_get_profitSoapOut" />
```

```
</wsdl:operation>
```

```
</wsdl:portType>
```

```
<wsdl:binding name="ServiceSoap" type="tns:ServiceSoap">
```

```
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
```

```
<wsdl:operation name="third_category_GetProductSales">
```

```
<soap:operation
  xmlns:tns="http://tempuri.org/third_category_GetProductSales"
  style="document" />
```



```
<wsdl:input>

  <soap:body use="literal" />

</wsdl:input>

<wsdl:output>

  <soap:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="third_category_GetProductPurchases">

  <soap:operation
soapAction="http://tempuri.org/third_category_GetProductPurchases"
style="document" />

  <wsdl:input>

    <soap:body use="literal" />

  </wsdl:input>

  <wsdl:output>

    <soap:body use="literal" />

  </wsdl:output>

</wsdl:operation>

<wsdl:operation
name="third_category_get_public_insurance_obligations">

  <soap:operation
soapAction="http://tempuri.org/third_category_get_public_insurance_obligati
ons" style="document" />

  <wsdl:input>

    <soap:body use="literal" />

  </wsdl:input>
```

```
<wsdl:output>
    <soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="third_category_get_public_tax_obligations">
    <soap:operation
soapAction="http://tempuri.org/third_category_get_public_tax_obligations"
style="document" />
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="third_category_get_profit">
    <soap:operation
soapAction="http://tempuri.org/third_category_get_profit" style="document"
/>
    <wsdl:input>
        <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal" />
    </wsdl:output>
</wsdl:operation>
```

```
</wsdl:binding>

<wsdl:binding name="ServiceSoap12" type="tns:ServiceSoap">

  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http"
/>

  <wsdl:operation name="third_category_GetProductSales">

    <soap12:operation
soapAction="http://tempuri.org/third_category_GetProductSales"
style="document" />

    <wsdl:input>

      <soap12:body use="literal" />

    </wsdl:input>

    <wsdl:output>

      <soap12:body use="literal" />

    </wsdl:output>

  </wsdl:operation>

  <wsdl:operation name="third_category_GetProductPurchases">

    <soap12:operation
soapAction="http://tempuri.org/third_category_GetProductPurchases"
style="document" />

    <wsdl:input>

      <soap12:body use="literal" />

    </wsdl:input>

    <wsdl:output>

      <soap12:body use="literal" />

    </wsdl:output>

  </wsdl:operation>
```

```
<wsdl:operation
name="third_category_get_public_insurance_obligations">

  <soap12:operation
soapAction="http://tempuri.org/third_category_get_public_insurance_obligati
ons" style="document" />

  <wsdl:input>

    <soap12:body use="literal" />

  </wsdl:input>

  <wsdl:output>

    <soap12:body use="literal" />

  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="third_category_get_public_tax_obligations">

  <soap12:operation
soapAction="http://tempuri.org/third_category_get_public_tax_obligations"
style="document" />

  <wsdl:input>

    <soap12:body use="literal" />

  </wsdl:input>

  <wsdl:output>

    <soap12:body use="literal" />

  </wsdl:output>
</wsdl:operation>

<wsdl:operation name="third_category_get_profit">

  <soap12:operation
soapAction="http://tempuri.org/third_category_get_profit" style="document"
/>

  <wsdl:input>
```

```

        <soap12:body xmlns="literal" />
    </wsdl:input>
    <wsdl:output>
        <soap12:body xmlns="literal" />
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="Service">
    <wsdl:port name="ServiceSoap" binding="tns:ServiceSoap">
        <soap:address
            address="http://localhost:4902/WebSite3/Service.asmx" />
    </wsdl:port>
    <wsdl:port name="ServiceSoap12" binding="tns:ServiceSoap12">
        <soap12:address
            address="http://localhost:4902/WebSite3/Service.asmx" />
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

#### 4.4 Διαδικτυακή εφαρμογή, που καλεί την Υπηρεσία Ιστού.

Προκειμένου να παρουσιαστεί στην πράξη η χρησιμότητα της Υπηρεσίας Ιστού που αναπτύχθηκε στην προηγούμενη ενότητα, κατασκευάστηκε μια διαδικτυακή εφαρμογή, η οποία αντλεί όλα της τα δεδομένα από την πρώτη.

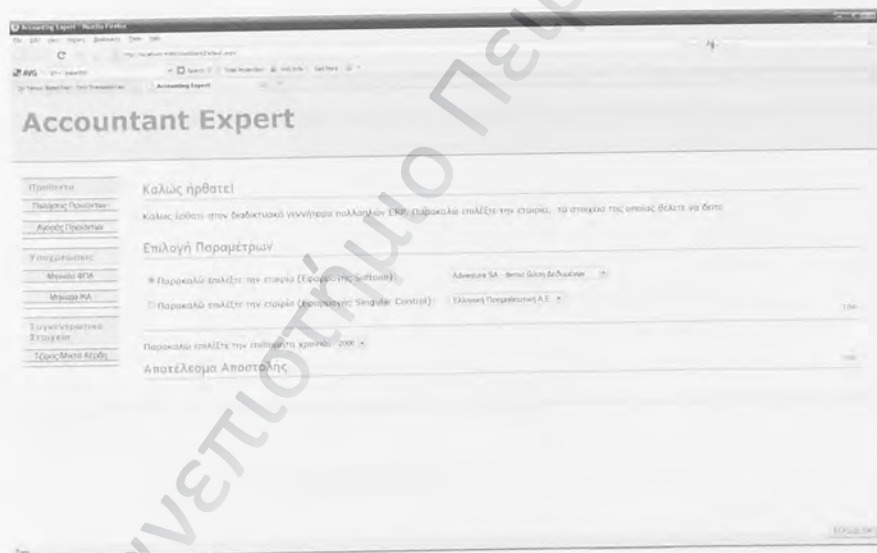
Δηλαδή, ο τελικός χρήστης, έρχεται σε επαφή μόνο με αυτή την ιστοσελίδα, η οποία όμως στο παρασκήνιο καλεί όλες τις μεθόδους που παρουσιάστηκαν προηγουμένως κατευθείαν πάνω στην Υπηρεσία Ιστού και εμφανίζει τα αποτελέσματα στον χρήστη.

Όπως θα δούμε στις επόμενες 2 εικόνες, η εφαρμογή αποτελείται αρχικά από μία οθόνη σύνδεσης, στην οποία πιστοποιούνται το όνομα χρήστη και το συνθηματικό του (η εφαρμογή καλύπτεται από τους κανόνες ασφαλείας του login control της Microsoft). Στη συνέχεια, φαίνεται η κεντρική ιστοσελίδα. Στην αριστερή μεριά φαίνονται όλες οι λειτουργίες που περιγράφηκαν προηγουμένως (και αποτελούν στην ιστοσελίδα τις επιλογές του χρήστη). Στο κέντρο και ψηλά της ιστοσελίδας, ο χρήστης μπορεί να επιλέξει με ένα radio button πάνω σε ποιά εφαρμογή θα εκτελέσει τις λειτουργίες του γεννήτορα (εφαρμογής Softone

,εφαρμογής Singular Control) και δίπλα ακριβώς επιλέγει την εταιρία στην οποία θα εκτελεστεί η όποια εργασία. Τέλος, παρακάτω επιλέγεται η χρονιά για την οποία θα εκτελέσουμε τη λειτουργία.

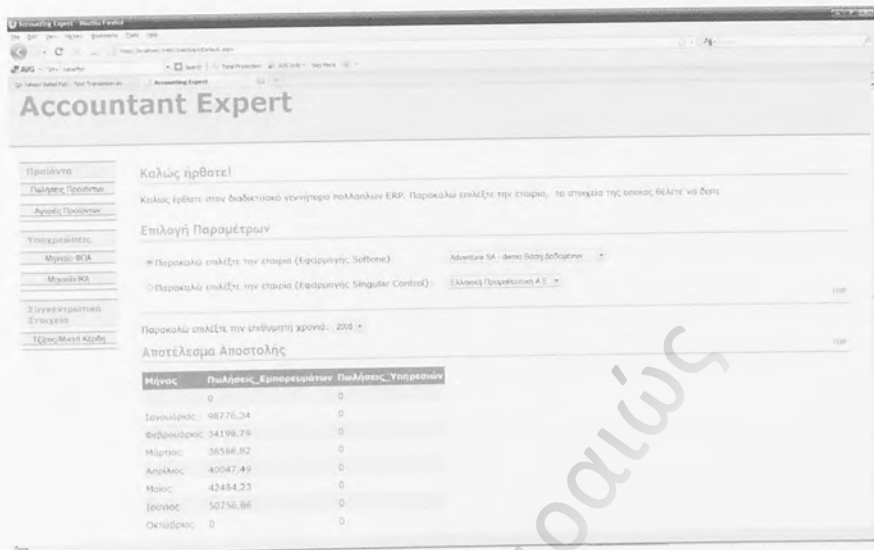


Εικόνα 20: Οθόνη εισαγωγής

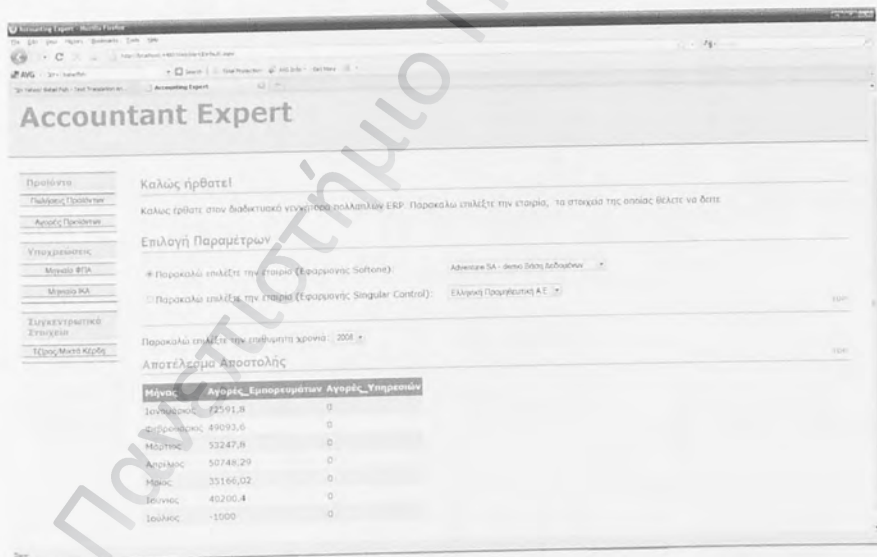


Εικόνα 21: Κεντρική σελίδα Διαδικτυακής Εφαρμογής

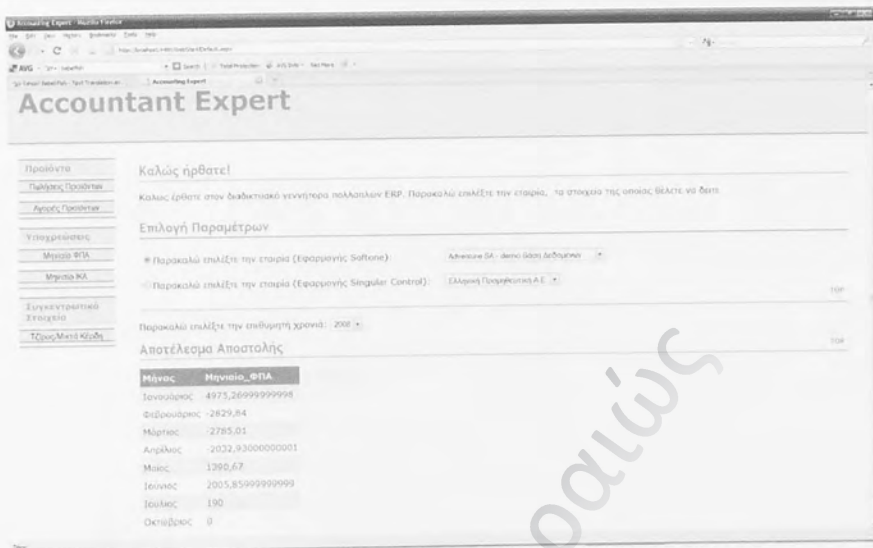
Έστω τώρα, ότι θέλουμε να τρέξουμε και τις πέντε λειτουργίες του μενού (Πωλήσεις Προϊόντων, Αγορές Προϊόντων, Μηνιαίο ΦΠΑ, Μηνιαίο ΙΚΑ, Τζίρος/Μικτά Κέρδη) για την εταιρία «Adventure S.A.» που τρέχει σε Softone, για το έτος 2008. Την επιλέγουμε από τη λίστα των εταιριών 'εφαρμογής Softone' αρχικά, επιλέγουμε με το radio button την κατηγορία 'Εφαρμογής Softone', και τέλος επιλέγουμε και το έτος '2008' στην επιλογή έτους. Επιλέγοντας διαδοχικά, από πάνω προς τα κάτω, όλες τις δυνατότητες του αριστερού μενού, μας έρχονται τα ακόλουθα αποτελέσματα:



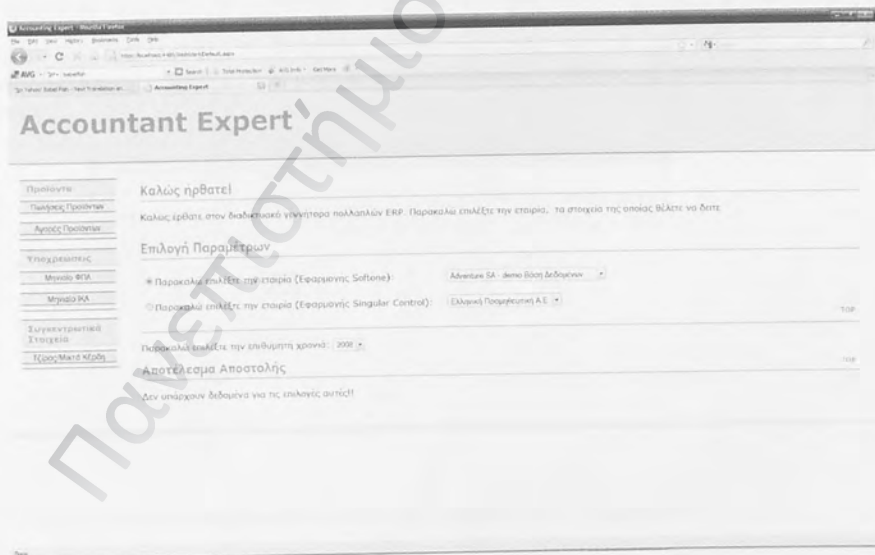
Εικόνα 22: Πωλήσεις Εμπορευμάτων



Εικόνα 23: Αγορές Εμπορευμάτων



Εικόνα 24: Μηνιαίο ΦΠΑ



Εικόνα 25: Μηνιαίο ΙΚΑ





```
using System;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Web.Services;

using System.Web.Services.Description;

using localhost;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {

        if (SoftoneRadioButton.Checked == true)
```

```
{  
  
    int company_chosen =  
Convert.ToInt32(DropDownList2.SelectedValue);  
  
    int fiscal_chosen =  
Convert.ToInt32(DropDownList1.SelectedValue);  
  
    localhost.Service proxy = new localhost.Service();  
  
    DataSet ds =  
proxy.third_category_GetProductSales(company_chosen, fiscal_chosen);  
  
    GridView1.DataSource = ds.Tables[0].DefaultView;  
  
    GridView1.DataBind();  
  
    if (ds.Tables[0].Rows.Count < 1)  
    {  
        Label3.Visible = true;  
        Label3.Text = "Δεν υπάρχουν δεδομένα για τις επιλογές  
αυτές!!";  
    }  
    else  
    {  
        Label3.Visible = false;  
    }  
}  
  
else  
  
    if (SingularControlRadioButton.Checked == true)  
  
    {
```

```
        int company_chosen =
Convert.ToInt32(DropDownList2.SelectedValues);

        int fiscal_chosen =
Convert.ToInt32(DropDownList1.SelectedValues);

        // Within this section

        // we must call the web methods implemented for
Singular Control ERP

        //

    }

    //the other ERP Systems Connection that will be implemented,
will follow here

}

protected void Button2_Click(object sender, EventArgs e)
{
    if (SoftoneRadioButton.Checked == true)
    {
        int company_chosen =
Convert.ToInt32(DropDownList2.SelectedValues);

        int fiscal_chosen =
Convert.ToInt32(DropDownList1.SelectedValues);

        localhost.Service proxy = new localhost.Service();

        DataSet ds =
proxy.third_category_GetProductPurchases(company_chosen, fiscal_chosen);

        GridView1.DataSource = ds.Tables[0].DefaultView;

        GridView1.DataBind();

        if (ds.Tables[0].Rows.Count < 1)
        {
```

```

        Label3.Visible = true;

        Label3.Text = "Δεν υπάρχουν δεδομένα για τις επιλογές
αυτές!!";

    }

    else

    {

        Label3.Visible = false;

    }

}

else

    if (SingularControlRadioButton.Checked == true)

    {

        int company_chosen =
Convert.ToInt32(DropDownList2.SelectedValues);

        int fiscal_chosen =
Convert.ToInt32(DropDownList1.SelectedValues);

        // Within this section

        // we must call the web methods implemented for
Singular Control ERP

        //

    }

    //the other ERP Systems Connection that will be implemented,
will follow here

}

protected void Button3_Click(object sender, EventArgs e)

{

```

```
if (SoftoneRadioButton.Checked == true)

{

    int company_chosen =
Convert.ToInt32(DropDownList2.Selected.Value);

    int fiscal_chosen =
Convert.ToInt32(DropDownList1.Selected.Value);

    localhost.Service proxy = new localhost.Service();

    DataSet ds =
proxy.third_category_get_public_tax_obligations(company_chosen,
fiscal_chosen);

    GridView1.DataSource = ds.Tables[0].DefaultView;

    GridView1.DataBind();

    if (ds.Tables[0].Rows.Count < 1)

    {

        Label3.Visible = true;

        Label3.Text = "Δεν υπάρχουν δεδομένα για τις επιλογές
αυτές!!";

    }

    else

    {

        Label3.Visible = false;

    }

}

else

    if (SingularControlRadioButton.Checked == true)

    {
```

```
        int company_chosen =
Convert.ToInt32(DropDownList2.Selected.Value);

        int fiscal_chosen =
Convert.ToInt32(DropDownList1.Selected.Value);

        // Within this section

        // we must call the web methods implemented for
Singular Control ERP

        //

    }

    //the other ERP Systems Connection that will be implemented,
will follow here

}

protected void Button4_Click(object sender, EventArgs e)
{

    if (SoftoneRadioButton.Checked == true)
    {

        int company_chosen =
Convert.ToInt32(DropDownList2.Selected.Value);

        int fiscal_chosen =
Convert.ToInt32(DropDownList1.Selected.Value);

        localhost.Service proxy = new localhost.Service();

        DataSet ds =
proxy.third_category_get_public_insurance_obligations(company_chosen,
fiscal_chosen);

        GridView1.DataSource = ds.Tables[0].DefaultView;

        GridView1.DataBind();

        if (ds.Tables[0].Rows.Count < 1)

        {
```

```
Label3.Visible = true;

Label3.Text = "Δεν υπάρχουν δεδομένα για τις επιλογές
αυτές!!";

    }

else

{

    Label3.Visible = false;

}

}

else

if (SingularControlRadioButton.Checked == true)

{

    int company_chosen =
Convert.ToInt32(DropDownList2.SelectedValues);

    int fiscal_chosen =
Convert.ToInt32(DropDownList1.SelectedValues);

    // Within this section

Singular Control ERP // we must call the web methods implemented for
//

}

//the other ERP Systems Connection that will be implemented,
will follow here

}

protected void Button5_Click(object sender, EventArgs e)

{
```



```
        if (SoftoneRadioButton.Checked == true)
        {
            int company_chosen =
Convert.ToInt32(DropDownList2.SelectedValue);

            int fiscal_chosen =
Convert.ToInt32(DropDownList1.SelectedValue);

            localhost.Service proxy = new
localhost.Service();

            DataSet ds =
proxy.third_category_get_profit(company_chosen, fiscal_chosen);

            GridView1.DataSource = ds.Tables[0].DefaultView;

            GridView1.DataBind();

            if (ds.Tables[0].Rows.Count < 1)
            {
                Label3.Visible = true;

                Label3.Text = "Δεν υπάρχουν δεδομένα για τις
επιλογές αυτές!!";
            }
            else
            {
                Label3.Visible = false;
            }
        }
    }
else

        if (SingularControlRadioButton.Checked == true)
```

```
{  
  
        int company_chosen =  
Convert.ToInt32(DropDownList2.SelectedValues);  
  
        int fiscal_chosen =  
Convert.ToInt32(DropDownList1.SelectedValues);  
  
        // Within this section  
  
        // we must call the web methods implemented for  
Singular Control ERP  
  
        //  
  
    }  
  
    //the other ERP Systems Connection that will be implemented,  
will follow here  
  
    }  
  
    protected void GridView1_RowDataBound(object sender,  
GridViewRowEventArgs e)  
  
    {  
  
        //if (e.Row.RowType == DataControlRowType.DataRow)  
  
        //{  
  
        //    decimal price =  
(decimal)DataBinder.Eval(e.Row.DataItem, "Πωλήσεις_Εμπορευμάτων");  
  
        //    if (price > 50)  
  
        //    {  
  
        //        e.Row.BackColor = System.Drawing.Color.Maroon;  
  
        //        e.Row.ForeColor = System.Drawing.Color.White;  
  
        //    }  
  
        //    }  
  
    }  
  
}
```

```
//          e.Row.Font.Bold = true;  
  
//      )  
  
//)  
  
)  
  
)
```

- Markup της Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"  
CodeFile="Default.aspx.cs" Inherits="_Default" %>  
  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
  
<head runat="server">  
  
    <title>Accounting Expert</title>  
  
</head>  
  
<form id="form1" runat="server">  
  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">  
  
<head>  
    <meta http-equiv="content-type" content="application/xhtml+xml;  
charset=utf-8" />  
  
    <meta name="author" content="haran" />  
  
    <meta name="generator" content="Windows Notepad" />  
  
    <link rel="stylesheet" type="text/css" href="bluehaze.css"  
title="Blue Haze stylesheet" />
```

```
<link rel="stylesheet" type="text/css" href="color-scheme.css"
title="Blue Haze stylesheet" />

<title>Blue Haze</title>

</head>

<style type="text/css">

    .style1

    {

        width: 525px;

    }

</style>

<body>

<div id="top"></div>

<!-- ##### Header ##### -->

<div id="header">

    <span class="headerTitle">Accountant Expert</span>

</div>

<!-- ##### Side Boxes ##### -->

<div class="sideBox LHS">

<div >Προϊόνια</div>

    <asp:Button

        ID="Button1" runat="server" onclick="Button1_Click"

        Text="Πωλήσεις Προϊόντων" Width="165px" CssClass="sideBox"

    />
```

```
<asp:Button
    ID="Button2" runat="server"
    Text="Αγορές Προϊόντων" Width="165px" CssClass="sideBox"
    onclick="Button2_Click"/>
</div>

<div class="sideBox LHS">
    <div>Υποχρεώσεις</div>
    <asp:Button
        ID="Button3" runat="server"
        Text="Μηνιαίο ΦΠΑ" Width="165px" CssClass="sideBox"
        onclick="Button3_Click" />
    <asp:Button
        ID="Button4" runat="server"
        Text="Μηνιαίο ΙΚΑ" Width="165px" CssClass="sideBox"
        onclick="Button4_Click"/>
</div>

<div class="sideBox LHS">
    <div>Συγκεντρωτικά Στοιχεία</div>
```



```

AutoPostBack="True"
        GroupName="RadioButtonList1" Checked="True"
    />
</td>
<td>
    <asp:DropDownList ID="DropDownList2"
runat="server"
        DataSourceID="SqlDataSource2"
        DataTextField="NAME"
        DataValueField="COMPANY"
        AutoPostBack="True">
    </asp:DropDownList>
</td>
</tr>
<tr>
<td class="style1">
    <asp:RadioButton ID="SingularControlRadioButton"
runat="server"
        Text="Παρακαλώ επιλέξτε την εταιρία
(Εφαρμογής Singular Control): "
        GroupName="RadioButtonList1"
        AutoPostBack="True"
    />
</td>

```

```

        <td>

                <asp:DropDownList ID="DropDownList3"
                runat="server"

                Visible="True" AutoPostBack="True" >

                <asp:ListItem>Ελληνική Προμηθευτική

                A.E.</asp:ListItem>

                <asp:ListItem>Ψηφιακή Συνεργασία

                E.Π.Ε.</asp:ListItem>

                </asp:DropDownList>

        </td>

</tr>

</table>

<h1 id="H1"></h1>

<dl>

        <asp:Label ID="Label2" runat="server" Text="Παρακαλώ
        επιλέξτε την επιθυμητή χρονιά:"></asp:Label>

        <asp:DropDownList ID="DropDownList1" runat="server"

        DataSourceID="SqlDataSource1" DataTextField="FISCPRD"
        DataValueField="FISCPRD">

        </asp:DropDownList>

        <asp:SqlDataSource ID="SqlDataSource2" runat="server"

        ConnectionString="<%$
        ConnectionStrings:demo_erpConnectionString %>"

        SelectCommand="SELECT [COMPANY], [NAME] FROM
        [COMPANY]"></asp:SqlDataSource>

</dl>

```



```
<a class="topOfPage" href="#top" title="Top Of Page">top</a>

<h1 id="older_virtues">Αποτέλεσμα Αποστολής</h1>

<dl>

    <asp:GridView ID="GridView1" runat="server" CellPadding="4"
    ForeColor="#333333"

        GridLines="None"
    onrowdatabound="GridView1_RowDataBound">

        <RowStyle BackColor="#EFF3FB" />

        <FooterStyle BackColor="#507CD1" Font-Bold="True"
    ForeColor="White" />

        <PagerStyle BackColor="#2461BF" ForeColor="White"
    HorizontalAlign="Center" />

        <SelectedRowStyle BackColor="#D1DDF1" Font-Bold="True"
    ForeColor="#333333" />

        <HeaderStyle BackColor="#507CD1" Font-Bold="True"
    ForeColor="White" />

        <EditRowStyle BackColor="#2461BF" />

        <AlternatingRowStyle BackColor="White" />

    </asp:GridView>

    <asp:Label ID="Label3" runat="server" Text="Label"
    Visible="False"></asp:Label>

    <asp:SqlDataSource ID="SqlDataSource1" runat="server"

        ConnectionString="<%$
    ConnectionStrings:demo_erpConnectionString %>"

        SelectCommand="SELECT DISTINCT [FISCPRD] FROM
    [acct_sigentrotiki]">

</asp:SqlDataSource>
```

```

</div>
<div></div>
<!-- ##### Footer ##### -->
<div id="footer">
</div>
</div>
</body>
</html>

```

## Συμπεράσματα

Στόχος της συγκεκριμένης μελέτης ήταν να επιτευχθεί ένα είδος επικοινωνίας μεταξύ διαφορετικών ERP συστημάτων, χρησιμοποιώντας την πλέον συνήθη τεχνολογία για τον συγκεκριμένο σκοπό, τις Υπηρεσίες Ιστού (αφού πρώτα αναλύθηκαν τόσο τα ERP συστήματα, όσο και οι Υπηρεσίες Ιστού). Προφανώς, το ζήτημα της συνεργασίας δύο τόσο πολύπλοκων συστημάτων σε όλο το φάσμα των λειτουργιών τους, αποτελεί μία μεγάλη πρόκληση. Αυτή ενισχύεται απ' το γεγονός ότι στην παγκόσμια αγορά δεν υπάρχει – μέχρι στιγμής – κάποιο έτοιμο 'πακέτο' λογισμικού μιας τέτοιας λύσης. Επίσης, όπως έγινε αντιληπτό, είναι αδύνατη η 'γεφύρωση' δύο τέτοιων συστημάτων, χωρίς την συμμετοχή προγραμματιστών από όλες τις κατασκευάστριες εταιρίες λογισμικού ERP. Τα οφέλη από ένα τέτοιο πιθανό έργο θα ήταν πολλαπλά. Εξάλλου, έχουν επιβεβαιωθεί απ' τις ουκ ολίγες υλοποιήσεις διαλειτουργικότητας λογισμικών που έχουν πραγματοποιηθεί μέχρι σήμερα (π.χ. η ενιαία αντιμετώπιση των αεροπορικών εταιριών στις κρατήσεις εισιτηρίων). Μελλοντικές μελέτες, θα μπορούσαν να αυξήσουν τις λειτουργίες που θα λειτουργούν παράλληλα σε διαφορετικά ERP (η παρούσα επικεντρώθηκε στις λογιστικές ενότητες) και γιατί όχι, να υπάρξει κάποια στιγμή ενιαία αντιμετώπιση των θεμάτων αυτής της κατηγορίας λογισμικού από όλες τις κατασκευάστριες εταιρίες και για όλες τις ενότητες – τουλάχιστον στις βασικές έννοιες της καθημίας.

## Βιβλιογραφία

- enterprise resource planning / Mary Summers
- Second-wave enterprise resource planning systems : implementing for effectiveness / Cambridge University Press, 2003.
- Web services platform architecture : SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and more / Prentice Hall PTR, c2005

- Service-oriented architecture : concepts, technology, and design / Prentice Hall Professional Technical Reference, c2005, 2008.
- Understanding enterprise SOA / Manning Publ., 2006
- Web services : a technical introduction / Prentice Hall, 2003
- Beginning XML with C# 2008 - From Novice to Professional / Bipin Josi
- Accelerated C# 2008 / Weldon W. Nash, III
- C# 3.0 Cookbook / Jay Hilyard and Stephen Teilhet
- XML Programming Bible / Brian Benz with John R. Durant
- Executive's Guide to Web Services / Eric A. Marks and Mark J. Werrell
- Programming Web Services with SOAP / Doug Tidwell, James Snell, Pavel Kulchenko
- Enterprise systems integration / Judith M. Myerson
- [http://www.ebizq.net/topics/erp\\_integration/features/2336.html](http://www.ebizq.net/topics/erp_integration/features/2336.html)  
/ Patrick Coleman
- <http://assets.emeraldinsight.com/Insight/viewContentItem.do?contentType=Article&hdAction=lnkhtml&contentId=843509&history=false>  
/ Sergio Beretta