



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

**ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΤΗΝ ΟΡΓΑΝΩΣΗ
ΚΑΙ ΔΙΟΙΚΗΣΗ ΒΙΟΜΗΧΑΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΕΙΔΙΚΕΥΣΗ: LOGISTICS**

Διπλωματική Εργασία

**Θεωρητική προσέγγιση της Ιχνηλασιμότητας και
πρακτική εφαρμογή της στα πλαίσια ανάλυσης,
σχεδίασης και μερικής υλοποίησης ενός
λογισμικού ιχνηλασιμότητας**

Logistics Δ'

Επιβλέπων: Μοσχούρης Σωκράτης

Φοιτητές: Καλκαβούρας Κων/νος ΜΠΛ/0518

Κατσικάρης Δημήτρης ΜΠΛ/0538

Πειραιάς, 2007

Αντί Προλόγου

Με την παρούσα μελέτη και με ιδιαίτερη χαρά σας παρουσιάζουμε την προσπάθειά μας για μια ολοκληρωμένη προσέγγιση της ιδιαίτερας σημαντικής, κατά την κρίση μας, και εξαιρετικά επίκαιρης έννοιας της «ιχνηλασιμότητας».

Το κίνητρο για τη μελέτη αυτή προέκυψε από τη διαπίστωση της έλλειψης ελληνικής βιβλιογραφίας στον τομέα της ιχνηλασιμότητας, αλλά και της σημασίας που προσλαμβάνει πλέον ως έννοια, μέσω της ραγδαίας ένταξης της στις ελληνικές επιχειρήσεις ως αναγκαία δραστηριότητα.

Στόχος αυτής της εργασίας είναι η δημιουργία ενός οδηγού ιχνηλασιμότητας, ο οποίος θα παρέχει στον αναγνώστη όλες τις πληροφορίες που απαιτούνται, ώστε να αποκτήσει μια πλήρη εικόνα και κατανόηση του όρου «ιχνηλασιμότητα», αλλά και να τον βοηθήσει να αντιληφθεί τον ρόλο της μέσα στο πλαίσιο του συνόλου των επιχειρησιακών διαδικασιών.

Επιπλέον, γίνεται εκτενής αναφορά στο θέμα της ανάπτυξης του κατάλληλου λογισμικού που θα προσφέρει τη δυνατότητα να παρακολουθείται η διαδικασία της ιχνηλασιμότητας. Η μελέτη αυτή της εφαρμογής της ιχνηλασιμότητας σε πρακτικό επίπεδο πραγματώνεται σε τρία στάδια: της ανάλυσης, της σχεδίασης και της υλοποίησης λογισμικού ιχνηλασιμότητας.

Με την ελπίδα να αποτελέσει ένα επιπλέον βοήθημα για όσους ενδιαφέρονται να μελετήσουν περαιτέρω την έννοια της ιχνηλασιμότητας, καταθέτουμε την παρούσα εργασία που είναι απόρροια της ισότιμης ερευνητικής συνεργασίας μας και καρπός των κοινών ακαδημαϊκών και προσωπικών μας ενδιαφερόντων.

Καλκαβούρας Κωνσταντίνος

Κατσικάρης Δημήτρης

Συνοπτικά Περιεχόμενα

Αντί Προλόγου	2
Συνοπτικά Περιεχόμενα	3
Αναλυτικά Περιεχόμενα	4
Πίνακας Σχημάτων	12
Πίνακας Διαγραμμάτων.....	14
Πίνακας Διαγραμμάτων.....	14
Περίληψη	16
Μέρος 1ο: Θεωρητική προσέγγιση της Ιχνηλασιμότητας.....	17
1.1. ΕΙΣΑΓΩΓΗ	18
1.2. ΑΝΑΓΚΗ ΙΧΝΗΛΑΣΙΜΟΤΗΤΑΣ	21
1.3. ΑΡΧΕΣ ΣΩΣΤΗΣ ΙΧΝΗΛΑΣΙΜΟΤΗΤΑΣ	23
1.4. ΕΦΑΡΜΟΖΟΜΕΝΕΣ ΤΕΧΝΟΛΟΓΙΕΣ ΣΤΗΝ ΙΧΝΗΛΑΣΙΜΟΤΗΤΑ	24
1.5. ΛΕΙΤΟΥΡΓΙΕΣ ΣΥΣΤΗΜΑΤΟΣ ΙΧΝΗΛΑΣΙΜΟΤΗΤΑΣ.....	88
1.6. ΕΦΑΡΜΟΓΗ ΣΥΣΤΗΜΑΤΟΣ ΙΧΝΗΛΑΣΙΜΟΤΗΤΑΣ	97
1.7. ΕΠΟΜΕΝΑ ΒΗΜΑΤΑ ΣΤΗΝ ΙΧΝΗΛΑΣΙΜΟΤΗΤΑ.....	99
1.8. ΣΥΜΠΕΡΑΣΜΑΤΑ	103
Μέρος 2ο: Πρακτική εφαρμογή στα πλαίσια ανάλυσης, σχεδίασης και μερικής υλοποίησης ενός λογισμικού ιχνηλασιμότητας.....	104
2.1. ΟΡΙΣΜΟΣ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ	105
2.2. ΕΓΓΡΑΦΟ ΠΡΟΔΙΑΓΡΑΦΩΝ ΤΩΝ ΑΠΑΙΤΗΣΕΩΝ ΑΠΟ ΤΟ ΛΟΓΙΣΜΙΚΟ.....	107
2.3. ΜΟΝΤΕΛΑ ΠΑΡΑΣΤΑΣΗΣ ΛΟΓΙΣΜΙΚΟΥ.....	125
2.4. ΕΓΓΡΑΦΟ ΠΕΡΙΓΡΑΦΗΣ ΤΟΥ ΣΧΕΔΙΟΥ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ.....	153
Βιβλιογραφία.....	261
Παράρτημα	264

Αναλυτικά Περιεχόμενα

Μέρος 1ο: Θεωρητική προσέγγιση της Ιχνηλασιμότητας	17
1.1. ΕΙΣΑΓΩΓΗ	18
1.2. ΑΝΑΓΚΗ ΙΧΝΗΛΑΣΙΜΟΤΗΤΑΣ.....	21
1.3. ΑΡΧΕΣ ΣΩΣΤΗΣ ΙΧΝΗΛΑΣΙΜΟΤΗΤΑΣ	23
1.4. ΕΦΑΡΜΟΖΟΜΕΝΕΣ ΤΕΧΝΟΛΟΓΙΕΣ ΣΤΗΝ ΙΧΝΗΛΑΣΙΜΟΤΗΤΑ.....	24
1.4.1. Αυτοματοποιημένη Αναγνώριση.....	24
1.4.1.1. GTIN (Global Trade Item Number)	26
1.4.1.1.1. Προϊόντα Σταθερής Μέτρησης	27
1.4.1.1.2. Προϊόντα μεταβλητής μέτρησης	29
1.4.1.1.3. Αλλαγή κωδικού Μονάδας Εμπορίας	32
1.4.1.1.4. Επιλογή μεταξύ των δομών κωδικοποίησης GTIN	33
1.4.1.1.5. Υπεύθυνος για την κωδικοποίηση των Μονάδων Εμπορίας	34
1.4.1.2. GLN (Global Location Number)	35
1.4.1.3. SSCC (Serial Shipping Container Code)	36
1.4.1.3.1. Ετικέτες Μονάδων Logistics.....	38
1.4.2. Αυτόματη Συλλογή Δεδομένων (Automatic Data Capture - ADC)	41
1.4.2.1. Barcode	41
1.4.2.1.1. Λειτουργία.....	42
1.4.2.1.2. Μεγέθη.....	43
1.4.2.1.3. Ανοιχτόχρωμα Περιθώρια (Ελεύθερες Ζώνες).....	44
1.4.2.1.4. Χρώματα και Αντίθεση (contrast)	45
1.4.2.1.5. Τρόποι Εκτύπωσης και Επικόλλησης	47
1.4.2.1.6. Ποιότητα Εκτύπωσης	53
1.4.2.1.7. Θέσεις barcode	53
1.4.2.1.8. Barcode που χρησιμοποιούνται στο Σύστημα GS1	60
1.4.2.1.9. Στοιχεία που πρέπει να λαμβάνονται υπ' όψιν για τη χρήση των συμβολογίων	68
1.4.2.1.10. Επιλογή μεταξύ των barcode.....	68
1.4.2.2. Radio Frequency Tags (RFID).....	70
1.4.2.2.1. Αρχιτεκτονική.....	71

1.4.2.2.2. Λειτουργία συστήματος RFID	72
1.4.2.2.3. Βασικά στοιχεία συστήματος RFID	73
1.4.2.2.4. Πρότυπα	80
1.4.2.2.5. Πλεονεκτήματα Τεχνολογίας RFID	81
1.4.2.2.6. Μειονεκτήματα Τεχνολογίας RFID.....	82
1.4.2.2.7. Σύγκριση RFID & Barcode	82
1.4.2.2.8. Θέματα Ασφαλείας Τεχνολογίας RFID	83
1.4.3. Ηλεκτρονική Επεξεργασία Δεδομένων (Electronic Data Processing - EDP).....	84
1.4.4. Ηλεκτρονική Ανταλλαγή Δεδομένων (Electronic Data Interchange - EDI).....	85
1.5. ΛΕΙΤΟΥΡΓΙΕΣ ΣΥΣΤΗΜΑΤΟΣ ΙΧΝΗΛΑΣΙΜΟΤΗΤΑΣ	88
1.5.1. Απόσυρση - Ανάκληση Προϊόντων	88
1.5.1.1. Υποχρεώσεις επιχειρήσεων περί μη ασφαλών τροφίμων - ζωοτροφών	89
1.5.1.1.1. Κοινοτική Νομοθεσία (Κανονισμός 178/2002/ΕΚ).....	89
1.5.1.1.2. Εθνική Νομοθεσία (ΚΥΑ 121/2003 ΦΕΚ 685 Β)	92
1.5.2. Διαχείριση αποθηκών.....	96
1.5.3. Ποιοτικός έλεγχος προϊόντων και πληροφόρηση καταναλωτή	96
1.6. ΕΦΑΡΜΟΓΗ ΣΥΣΤΗΜΑΤΟΣ ΙΧΝΗΛΑΣΙΜΟΤΗΤΑΣ	97
1.7. ΕΠΟΜΕΝΑ ΒΗΜΑΤΑ ΣΤΗΝ ΙΧΝΗΛΑΣΙΜΟΤΗΤΑ	99
1.7.1. EPC και EPCglobal Network.....	99
1.7.1.1. Ηλεκτρονικός Κωδικός Προϊόντος	101
1.8. ΣΥΜΠΕΡΑΣΜΑΤΑ.....	103
Μέρος 2ο: Πρακτική εφαρμογή στα πλαίσια ανάλυσης, σχεδίασης και μερικής υλοποίησης ενός λογισμικού ιχνηλασιμότητας.....	104
2.1. ΟΡΙΣΜΟΣ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ	105
2.1.1. Παρουσίαση του προβλήματος	105
2.1.2. Περιληπτική σκιαγράφηση της λύσης	105
2.2. ΕΓΓΡΑΦΟ ΠΡΟΔΙΑΓΡΑΦΩΝ ΤΩΝ ΑΠΑΙΤΗΣΕΩΝ ΑΠΟ ΤΟ ΛΟΓΙΣΜΙΚΟ	107
2.2.1. Εισαγωγή.....	107
2.2.1.1. Ταυτότητα του εγγράφου	107
2.2.1.2. Σκοπός	107

2.2.1.3. Εμβέλεια	107
2.2.1.4. Ορισμοί, ακρωνύμια, συντομογραφίες	107
2.2.1.5. Πηγές αναφορών	107
2.2.1.6. Περίληψη	108
2.2.2. Γενική περιγραφή του λογισμικού.....	108
2.2.2.1. Στίγμα	108
2.2.2.2. Προοπτική	109
2.2.2.3. Γενικές λειτουργίες του λογισμικού.....	109
2.2.2.4. Χαρακτηριστικά χρηστών.....	109
2.2.2.5. Περιορισμοί.....	109
2.2.2.6. Παραδοχές και εξαρτήσεις	110
2.2.2.6.1. Παραδοχές και εξαρτήσεις κατά την ανάπτυξη	110
2.2.2.6.2. Παραδοχές και εξαρτήσεις κατά την χρήση	110
2.2.3. Ειδικές απαιτήσεις	110
2.2.3.1. Απαιτήσεις εξωτερικών διαπροσωπειών	110
2.2.3.1.1. Διαπροσωπίες χρήστη	110
2.2.3.1.2. Διαπροσωπίες υλικού.....	110
2.2.3.1.3. Διαπροσωπίες λογισμικού	111
2.2.3.1.4. Διαπροσωπίες επικοινωνιών	111
2.2.3.2. Λειτουργικές απαιτήσεις.....	111
2.2.3.2.1. Τρόπος λειτουργίας 1 (Διαχείριση Πρώτων Υλών)	111
2.2.3.2.2. Τρόπος λειτουργίας 2 (Διαχείριση Προϊόντων)	112
2.2.3.2.3. Τρόπος λειτουργίας 3 (Διαχείριση Πελατών).....	112
2.2.3.2.4. Τρόπος λειτουργίας 4 (Διαχείριση Αποθεμάτων Πρώτων Υλών). 113	
2.2.3.2.5. Τρόπος λειτουργίας 5 (Διαχείριση Αποθεμάτων Προϊόντων).....	113
2.2.3.2.6. Τρόπος λειτουργίας 6 (Παραγωγή Προϊόντων)	114
2.2.3.2.7. Τρόπος λειτουργίας 7 (Παραγγελία Προϊόντων).....	114
2.2.3.2.8. Τρόπος λειτουργίας 8 (Ιχνηλασιμότητα)	116
2.2.3.2.9. Τρόπος λειτουργίας 9 (Κεντρική Διαχείριση Εφαρμογής)	116
2.2.3.2.10. Τρόπος λειτουργίας 10 (Διαχείριση Χρηστών)	117
2.2.3.2.11. Τρόπος λειτουργίας 11 (Συντήρηση)	118
2.2.3.3. Απαιτήσεις επιδόσεων	118

2.2.3.4. Περιορισμοί σχεδίασης.....	118
2.2.3.4.1. Περιορισμοί από το υλικό	118
2.2.3.4.2. Συμμόρφωση με πρότυπα	119
2.2.3.5. Χαρακτηριστικά του λογισμικού.....	119
2.2.3.5.1. Αξιοπιστία	119
2.2.3.5.2. Διαθεσιμότητα	119
2.2.3.5.3. Ασφάλεια	119
2.2.3.5.4. Χαρακτηριστικά συντήρησης	119
2.2.3.5.5. Μεταφερισιμότητα	120
2.2.3.6. Άλλες απαιτήσεις	120
2.2.3.6.1. Φυσικές απαιτήσεις	120
2.2.3.6.2. Απαιτήσεις Βάσεων Δεδομένων	120
2.2.3.6.3. Απαιτήσεις Υλοποίησης.....	124
2.3. ΜΟΝΤΕΛΑ ΠΑΡΑΣΤΑΣΗΣ ΛΟΓΙΣΜΙΚΟΥ	125
2.3.1. Διαγράμματα ροής δεδομένων	125
2.3.1.1. 1 ^{ος} Τρόπος λειτουργίας (Διαχείριση πρώτων υλών)	125
2.3.1.1.1. Επίπεδο 1.....	125
2.3.1.1.2. Επίπεδο 2.....	125
2.3.1.1.3. Επίπεδο 3.....	126
2.3.1.2. 2 ^{ος} Τρόπος λειτουργίας (Διαχείριση προϊόντων)	127
2.3.1.2.1. Επίπεδο 1.....	127
2.3.1.2.2. Επίπεδο 2.....	127
2.3.1.2.3. Επίπεδο 3.....	128
2.3.1.3. 3 ^{ος} Τρόπος λειτουργίας (Διαχείριση πελατών).....	129
2.3.1.3.1. Επίπεδο 1.....	129
2.3.1.3.2. Επίπεδο 2.....	129
2.3.1.3.3. Επίπεδο 3.....	130
2.3.1.4. 4 ^{ος} Τρόπος λειτουργίας (Διαχείριση αποθεμάτων πρώτων υλών) .	131
2.3.1.4.1. Επίπεδο 1.....	131
2.3.1.4.2. Επίπεδο 2.....	131
2.3.1.4.3. Επίπεδο 3.....	132
2.3.1.5. 5 ^{ος} Τρόπος λειτουργίας (Διαχείριση αποθεμάτων προϊόντων)	133

2.3.1.5.1. Επίπεδο 1.....	133
2.3.1.5.2. Επίπεδο 2.....	133
2.3.1.5.3. Επίπεδο 3.....	134
2.3.1.6. 6 ^{ος} Τρόπος λειτουργίας (Παραγωγή προϊόντων)	135
2.3.1.6.1. Επίπεδο 1.....	135
2.3.1.6.2. Επίπεδο 2.....	135
2.3.1.6.3. Επίπεδο 3.....	136
2.3.1.7. 7 ^{ος} Τρόπος λειτουργίας (Παραγγελίες προϊόντων).....	137
2.3.1.7.1. Επίπεδο 1.....	137
2.3.1.7.2. Επίπεδο 2.....	137
2.3.1.7.3. Επίπεδο 3.....	138
2.3.1.8. 8 ^{ος} Τρόπος λειτουργίας (Ικνηλασιμότητα)	139
2.3.1.8.1. Επίπεδο 1.....	139
2.3.1.8.2. Επίπεδο 2.....	139
2.3.1.8.3. Επίπεδο 3.....	140
2.3.1.9. 9 ^{ος} Τρόπος λειτουργίας (Κεντρική διαχείριση εφαρμογής)	141
2.3.1.9.1. Επίπεδο 1.....	141
2.3.1.9.2. Επίπεδο 2.....	141
2.3.1.9.3. Επίπεδο 3.....	142
2.3.1.10. 10 ^{ος} Τρόπος λειτουργίας (Διαχείριση χρηστών).....	143
2.3.1.10.1. Επίπεδο 1	143
2.3.1.10.2. Επίπεδο 2	143
2.3.1.10.3. Επίπεδο 3	144
2.3.1.11. 11 ^{ος} Τρόπος λειτουργίας (Συντήρηση).....	144
2.3.2. Διάγραμμα οντοτήτων - συσχετίσεων	145
2.3.3. Διαγράμματα μετάβασης καταστάσεων	146
2.3.4. Λεξικό δεδομένων.....	148
2.4. ΕΓΓΡΑΦΟ ΠΕΡΙΓΡΑΦΗΣ ΤΟΥ ΣΧΕΔΙΟΥ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ	153
2.4.1. Εισαγωγή.....	153
2.4.1.1. Γενική περιγραφή	153
2.4.1.2. Εμβέλεια	153
2.4.1.3. Ορισμοί και ακρωνύμια	153

2.4.1.4. Αναφορές	153
2.4.2. Περιγραφή αποσύνθεσης του λογισμικού	153
2.4.2.1. Αποσύνθεση σε μονάδες	153
2.4.2.1.1. Περιγραφή μονάδας 1	158
2.4.2.1.2. Περιγραφή μονάδας 2	158
2.4.2.1.3. Περιγραφή μονάδας 3	158
2.4.2.1.4. Περιγραφή μονάδας 4	158
2.4.2.1.5. Περιγραφή μονάδας 5	158
2.4.2.1.6. Περιγραφή μονάδας 6.	158
2.4.2.1.7. Περιγραφή μονάδας 7.	158
2.4.2.1.8. Περιγραφή μονάδας 8.	158
2.4.2.1.9. Περιγραφή μονάδας 9.	159
2.4.2.1.10. Περιγραφή μονάδας 10.	159
2.4.2.1.11. Περιγραφή μονάδας 11.	159
2.4.2.1.12. Περιγραφή μονάδας 12.	159
2.4.2.1.13. Περιγραφή μονάδας 13.	159
2.4.2.1.14. Περιγραφή μονάδας 14.	159
2.4.2.1.15. Περιγραφή μονάδας 15.	159
2.4.2.1.16. Περιγραφή μονάδας 16.	159
2.4.2.1.17. Περιγραφή μονάδας 17	159
2.4.2.1.18. Περιγραφή μονάδας 18.	160
2.4.2.1.19. Περιγραφή μονάδας 19.	160
2.4.2.1.20. Περιγραφή μονάδας 20.	160
2.4.2.1.21. Περιγραφή μονάδας 21.	160
2.4.2.1.22. Περιγραφή μονάδας 22.	160
2.4.2.1.23. Περιγραφή μονάδας 23.	160
2.4.2.1.24. Περιγραφή μονάδας 24.	160
2.4.2.1.25. Περιγραφή μονάδας 25.	160
2.4.2.1.26. Περιγραφή μονάδας 26.	161
2.4.2.1.27. Περιγραφή μονάδας 27.	161
2.4.2.1.28. Περιγραφή μονάδας 28.	161
2.4.2.1.29. Περιγραφή μονάδας 29.	161

2.4.2.1.30. Περιγραφή μονάδας 30.	161
2.4.2.1.31. Περιγραφή μονάδας 31	161
2.4.2.1.32. Περιγραφή μονάδας 32	161
2.4.2.1.33. Περιγραφή μονάδας 33	161
2.4.2.1.34. Περιγραφή μονάδας 34	161
2.4.2.1.35. Περιγραφή μονάδας 35	162
2.4.2.1.36. Περιγραφή μονάδας 36.	162
2.4.2.1.37. Περιγραφή μονάδας 37.	162
2.4.2.1.38. Περιγραφή μονάδας 38.	162
2.4.2.1.39. Περιγραφή μονάδας 39.	162
2.4.2.1.40. Περιγραφή μονάδας 40.	162
2.4.2.1.41. Περιγραφή μονάδας 41.	162
2.4.2.1.42. Περιγραφή μονάδας 42.	162
2.4.2.1.43. Περιγραφή μονάδας 43.	163
2.4.2.1.44. Περιγραφή μονάδας 44.	163
2.4.2.1.45. Περιγραφή μονάδας 45.	163
2.4.2.1.46. Περιγραφή μονάδας 46.	163
2.4.2.1.47. Περιγραφή μονάδας 47.	163
2.4.2.1.48. Περιγραφή μονάδας 48.	163
2.4.2.1.49. Περιγραφή μονάδας 49.	163
2.4.2.1.50. Περιγραφή μονάδας 50.	163
2.4.2.1.51. Περιγραφή μονάδας 51.	163
2.4.2.1.52. Περιγραφή μονάδας 52.	164
2.4.2.1.53. Περιγραφή μονάδας 53.	164
2.4.2.1.54. Περιγραφή μονάδας 54.	164
2.4.2.1.55. Περιγραφή μονάδας 55.	164
2.4.2.2. Αποσύνθεση σε ταυτόχρονες διεργασίες	164
2.4.2.3. Αποσύνθεση δεδομένων	164
2.4.3. Περιγραφή εξαρτήσεων	164
2.4.3.1. Εξαρτήσεις μεταξύ μονάδων	164
2.4.3.2. Εξαρτήσεις μεταξύ διεργασιών	164
2.4.3.3. Εξαρτήσεις μεταξύ δεδομένων	165

2.4.4. Περιγραφή διαπροσωπειών	165
2.4.4.1. Διαπροσωπείες μονάδων	165
2.4.4.2. Διαπροσωπείες διεργασιών	165
2.4.4.3. Διαπροσωπείες χρήστη	165
2.4.4.4. Εξωτερικές διαπροσωπείες.....	174
2.4.4.4.1. Συστήματα λογισμικού.....	174
2.4.4.4.2. Συσκευές	174
2.4.5. Λεπτομερές σχέδιο μονάδων	175
2.4.6. Λεπτομερές σχέδιο δεδομένων	260

Πίνακας Σχημάτων

Σχήμα 1. Μονάδες εμπορίας	26
Σχήμα 2. GIN-13	27
Σχήμα 3. GTIN -12.....	27
Σχήμα 4. GTIN - 8	28
Σχήμα 5. GTIN - 14.....	28
Σχήμα 6. Δομή κωδικού προϊόντος μεταβλητού βάρους από παραγωγό.....	30
Σχήμα 7. Σχήμα 6. Δομή κωδικού προϊόντος μεταβλητού βάρους από λιανέμπορο	30
Σχήμα 8. GTIN - 14 με τον δείκτη «9»	31
Σχήμα 9. Τυπικοί δείκτες εφαρμογής (AI)	32
Σχήμα 10. Επιλογή μεταξύ των δομών κωδικοποίησης GTIN	33
Σχήμα 11. Διάγραμμα επιλογής κωδικών	34
Σχήμα 12. GLN (Global Location Number)	36
Σχήμα 13. Μορφή του σειριακού κωδικού SSCC	38
Σχήμα 14. Πρότυπη ετικέτα Μονάδας Logistics GS1	40
Σχήμα 15. Ετικέτα με τμήματα προμηθευτή, πελάτη και μεταφορέα	41
Σχήμα 16. Προσπίπτουσα και ανακλώμενη δέσμη φωτός	43
Σχήμα 17. Περιγραφή δομής barcode	43
Σχήμα 18. Αναγνώσιμα barcode.....	46
Σχήμα 19. Μη αναγνώσιμα barcode	47
Σχήμα 22. Ετικετέζα με προσαρμοσμένο θερμικό εκτυπωτή τύπου SmartDate	50
Σχήμα 24. Εκτυπωτής εκτόξευσης μελάνης (ink-jet) και δείγματα εφαρμογών.....	52
Σχήμα 25. Εκτυπωτής εκτόξευσης μελάνης (ink-jet) υψηλής ανάλυσης και δείγματα εφαρμογών	53
Σχήμα 26. Θέση barcode στα είδη λιανικής	54
Σχήμα 27. Θέση barcode στα είδη λιανικής 2	54
Σχήμα 28. Θέση barcode στα είδη λιανικής 3	55
Σχήμα 29. Θέση barcode στα είδη λιανικής 4	55
Σχήμα 30. Θέση barcode στα είδη λιανικής 5	56
Σχήμα 31. Barcode σε χαρτοκιβώτια	57
Σχήμα 32. Barcode σε παλέτες	58
Σχήμα 33. Barcode σε μικρά κιβώτια.....	59
Σχήμα 34. Μη αναγνώσιμο barcode 1	60

Σχήμα 35. Μη αναγνώσιμο barcode 2	60
Σχήμα 36. . Μη αναγνώσιμο barcode 3	60
Σχήμα 37. Σύμβολο EAN - 8.....	61
Σχήμα 38. Σύμβολο EAN - 13	62
Σχήμα 39. Σύμβολο UPC - A.....	63
Σχήμα 40. Σύμβολο UPC - E.....	64
Σχήμα 41. 1.4.2.1.8.2. Σύμβολο ITF - 14	65
Σχήμα 42. 1.4.2.1.8.3. Σύμβολο GS1 - 128	66
Σχήμα 43. Reduced Space Symbology (RSS)	67
Σχήμα 44. 1.4.2.1.8.5. RSS & COMPOSITE	68
Σχήμα 45. 1.4.2.1.10. Επιλογή μεταξύ των barcode	70
Σχήμα 46. Αρχιτεκτονική του συστήματος RFID.....	72
Σχήμα 47. Διαφορές μεταξύ παθητικών και ενεργητικών ετικετών.....	75
Σχήμα 48. Διαφορές μεταξύ αναγνώσιμων και επανεγγράψιμων ετικετών.....	76
Σχήμα 49. Barcode αναγνώστης	77
Σχήμα 50. Ιδιότητες για κάθε μία από τις κατηγορίες των αναγνωστών.....	78
Σχήμα 51. Ιδιότητες για κάθε μία από τις κατηγορίες των αναγνωστών 2.....	79
Σχήμα 52. RFID και πρότυπα	81
Σχήμα 53. Ανάκληση προϊόντων 1	94
Σχήμα 54. Ανάκληση προϊόντων 2	94
Σχήμα 55. Ανάκληση με διαχείριση παρτίδων.....	95
Σχήμα 56. RFID and EPC.....	100
Σχήμα 57. EPC Number	101
Σχήμα 58. EPC Number 2.....	101
Σχήμα 59. GTIN integration with the EPC	101
Σχήμα 60. Λειτουργία εφοδιαστικής αλυσίδας με RF και EPC.....	102

Πίνακας Διαγραμμάτων

Διάγραμμα 1. 1ος Τρόπος Λειτουργίας - Επίπεδο 1	125
Διάγραμμα 2. 1ος Τρόπος Λειτουργίας - Επίπεδο 2	125
Διάγραμμα 3. 1ος Τρόπος Λειτουργίας - Επίπεδο 3	126
Διάγραμμα 4. 2ος Τρόπος Λειτουργίας - Επίπεδο 1	127
Διάγραμμα 5. 2ος Τρόπος Λειτουργίας - Επίπεδο 2	127
Διάγραμμα 6. 2ος Τρόπος Λειτουργίας - Επίπεδο 3	128
Διάγραμμα 7. 3ος Τρόπος Λειτουργίας - Επίπεδο 1	129
Διάγραμμα 8. 3ος Τρόπος Λειτουργίας - Επίπεδο 2	129
Διάγραμμα 9. 3ος Τρόπος Λειτουργίας - Επίπεδο 3	130
Διάγραμμα 10. 4ος Τρόπος Λειτουργίας - Επίπεδο 1	131
Διάγραμμα 11. 4ος Τρόπος Λειτουργίας - Επίπεδο 2	131
Διάγραμμα 12. 4ος Τρόπος Λειτουργίας - Επίπεδο 3	132
Διάγραμμα 13. 5ος Τρόπος Λειτουργίας - Επίπεδο 1	133
Διάγραμμα 14. 5ος Τρόπος Λειτουργίας - Επίπεδο 2	133
Διάγραμμα 15. 5ος Τρόπος Λειτουργίας - Επίπεδο 3	134
Διάγραμμα 16. 6ος Τρόπος Λειτουργίας - Επίπεδο 1	135
Διάγραμμα 17. 6ος Τρόπος Λειτουργίας - Επίπεδο 2	135
Διάγραμμα 18. 6ος Τρόπος Λειτουργίας - Επίπεδο 3	136
Διάγραμμα 19. 7ος Τρόπος Λειτουργίας - Επίπεδο 1	137
Διάγραμμα 20. 7ος Τρόπος Λειτουργίας - Επίπεδο 2	137
Διάγραμμα 21. 7ος Τρόπος Λειτουργίας - Επίπεδο 3	138
Διάγραμμα 22. 8ος Τρόπος Λειτουργίας - Επίπεδο 1	139
Διάγραμμα 23. 8ος Τρόπος Λειτουργίας - Επίπεδο 2	139
Διάγραμμα 24. 8ος Τρόπος Λειτουργίας - Επίπεδο 3	140
Διάγραμμα 25. 9ος Τρόπος Λειτουργίας - Επίπεδο 1	141
Διάγραμμα 26. 9ος Τρόπος Λειτουργίας - Επίπεδο 2	141
Διάγραμμα 27. 9ος Τρόπος Λειτουργίας - Επίπεδο 3	142
Διάγραμμα 28. 10ος Τρόπος Λειτουργίας - Επίπεδο 1	143
Διάγραμμα 29. 10ος Τρόπος Λειτουργίας - Επίπεδο 2	143
Διάγραμμα 30. 10ος Τρόπος Λειτουργίας - Επίπεδο 3	144
Διάγραμμα 31. Οντοτήτων - Συσχετίσεων	145
Διάγραμμα 32. Μετάβαση καταστάσεων (Παραγωγή)	146
Διάγραμμα 33. Μετάβαση καταστάσεων (Παραγγελία)	146

Διάγραμμα 34. Μετάβαση καταστάσεων (Εύρεση ΠΡ., ΠΛ., Π.Υ.) 147

Διάγραμμα 35. Μετάβαση καταστάσεων (Ανάκληση) 147

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΔΙΑ

Περίληψη

Η παρούσα διπλωματική εργασία διερευνά την έννοια της «ιχνηλασιμότητας». Τον όρο αυτό τον εξετάζει υπό το πρίσμα τόσο της θεωρητικής προσέγγισης όσο και της πρακτικής εφαρμογής του, σε μια προσπάθεια σφαιρικής διερεύνησης της έννοιας, αλλά και του τρόπου με τον οποίο υλοποιείται ένα σύστημα ιχνηλασιμότητας.

Στο θεωρητικό μέρος περιγράφεται ο όρος «ιχνηλασιμότητα», η ανάγκη για την ύπαρξη της, οι αρχές που την διέπουν, οι τεχνολογίες που εφαρμόζονται κ.α. Στο πρακτικό μέρος περιγράφονται και αναλύονται οι απαιτήσεις ενός λογισμικού που θα εκπληρώνει τη διαδικασία της ιχνηλασιμότητας, σχεδιάζεται το λογισμικό αναλυτικά, και παρακολουθείται η προσπάθεια υλοποίησης του λογισμικού αυτού.

Μέρος 1ο:
Θεωρητική προσέγγιση της Ιχνηλασιμότητας

1.1. ΕΙΣΑΓΩΓΗ

Τα προβλήματα που εντοπίστηκαν στην τροφική αλυσίδα τα τελευταία χρόνια στον Ευρωπαϊκό χώρο, όπως για παράδειγμα το σκάνδαλο της νόσου των τρελών αγελάδων με τα κρεατάλευρα, το επεισόδιο των διοξινών, η χρήση ορμονών στην πάχυνση των ζώων, κατέστησαν επιτακτική την ανάγκη της ασφαλούς και υγιεινής παραγωγής αγροτικών προϊόντων και τροφίμων σε όλα τα στάδια παραγωγής, σε κοινοτικό και σε εθνικό επίπεδο.

Η Ευρωπαϊκή Ένωση έλαβε τα μέτρα της για τη λύση των προβλημάτων που παρουσιάζονται στην τροφική αλυσίδα και στην παραγωγή τροφίμων. Εξέδωσε τον Κανονισμό 178/2002 για την ασφάλεια των τροφίμων και των ζωοτροφών που άρχισε να ισχύει την 1η Ιανουαρίου 2005. Οι βασικές υποχρεώσεις των επιχειρήσεων του τομέα τροφίμων και ζωοτροφών, που απορρέουν από τον Κανονισμό, είναι οι ακόλουθες:

- ✓ Ασφάλεια: Οι επιχειρήσεις οφείλουν να μη διαθέτουν στην αγορά μη ασφαλή τρόφιμα και ζωοτροφές.
- ✓ Ευθύνη: Οι επιχειρήσεις είναι υπεύθυνες για την ασφάλεια των τροφίμων και ζωοτροφών που παράγουν, μεταφέρουν αποθηκεύουν ή πωλούν.
- ✓ Ικνηλασιμότητα: Η ικνηλασιμότητα των τροφίμων είναι η δυνατότητα εξακρίβωσης της προέλευσης των προϊόντων από άκρη σε άκρη της τροφικής αλυσίδας. Οι νέες υποχρεωτικές απαιτήσεις για την ικνηλασιμότητα εφαρμόζονται σε όλα τα τρόφιμα, τις ζωοτροφές, τα παραγωγικά ζώα και όλες τις κατηγορίες υπευθύνων επιχειρήσεων της τροφικής αλυσίδας από το αγρόκτημα έως τη μεταποίηση, τη μεταφορά, την αποθήκευση, τη διανομή και τη λιανική πώληση στον καταναλωτή. Στο σύστημα ικνηλασιμότητας που εφαρμόζει κάθε υπεύθυνος επιχείρησης του τομέα πρέπει να καταγράφονται συστηματικά στοιχεία για το όνομα, τη διεύθυνση του παραγωγού, το είδος των προϊόντων και την ημερομηνία συναλλαγής. Τα στοιχεία αυτά πρέπει να φυλάσσονται για περίοδο 5 ετών και, εφόσον ζητηθούν, πρέπει να διατίθενται αμέσως στις αρμόδιες αρχές. Οι επιχειρήσεις πρέπει να είναι σε θέση να ταυτοποιούν ταχέως κάθε προμηθευτή ή παραλήπτη.
- ✓ Διαφάνεια: Οι επιχειρήσεις οφείλουν να ενημερώνουν αμέσως τις αρμόδιες αρχές, αν έχουν κάποιο λόγο να πιστεύουν ότι τα τρόφιμα ή οι ζωοτροφές τους δεν είναι ασφαλή.
- ✓ Κατάσταση Έκτακτης Ανάγκης: Οι επιχειρήσεις οφείλουν να αποσύρουν αμέσως τα τρόφιμα ή τις ζωοτροφές από την αγορά, αν έχουν κάποιο λόγο να πιστεύουν ότι αυτά δεν είναι ασφαλή.

- ✓ Πρόληψη: Οι επιχειρήσεις οφείλουν να εντοπίζουν και να επανεξετάζουν τακτικά τα κρίσιμα σημεία στις διαδικασίες τους και να μεριμνούν για τη διενέργεια ελέγχων στα σημεία αυτά.
- ✓ Συνεργασία: Οι επιχειρήσεις οφείλουν να συνεργάζονται με τις αρμόδιες αρχές σε ενέργειες που αναλαμβάνονται για τη μείωση των κινδύνων.
- ✓ Η συμμόρφωση των επιχειρήσεων του τομέα τροφίμων και ζωοτροφών με τις Νομοθετικές απαιτήσεις της Ε.Ε. για την ασφάλεια των τροφίμων θα συμβάλει σημαντικά στην εθνική και κοινοτική οικονομία. Ως γνωστό η βιομηχανία τροφίμων και ποτών είναι ένας σημαντικότερος τομέας στην Ε.Ε. με ετήσια παραγωγή αξίας 600 δισ. ευρώ ή 15% της συνολικής βιομηχανικής παραγωγής. Ο γεωργικός τομέας επίσης έχει ετήσια παραγωγή αξίας 220 δισ. ευρώ. Οι εξαγωγές των γεωργικών προϊόντων, των τροφίμων και των ποτών αξίζουν 50 δισ. ευρώ ετησίως. Η ευρωπαϊκή νομοθεσία για την ασφάλεια των τροφίμων συνηγορεί στη διασφάλιση της ποιότητας και της υγιεινής των τροφίμων και κατ' επέκταση της υγείας των καταναλωτών. Η συμμόρφωση με τις απαιτήσεις της Νομοθεσίας άρχισε ήδη να απασχολεί τις επιχειρήσεις τροφίμων, εκείνων που θέλουν να πρωτοστατούν και να υπερέχουν στην ανταγωνιστική αγορά.

Όπως αναφέραμε παραπάνω αρκετά από τα μέτρα που έλαβε η Ε.Ε., για τη λύση των προβλημάτων που παρουσιάζονται στην τροφική αλυσίδα και στην παραγωγή τροφίμων, αφορούν την ικνηλασιμότητα. Ακολούθως παραθέτουμε δυο ορισμούς της έννοιας ικνηλασιμότητας, με σκοπό να γίνει κατανοητότερο το αντικείμενο που πραγματεύεται η συγκεκριμένη εργασία.

1ος ορισμός

Η ικανότητα να παρακολουθεί κανείς, μέσα από όλες τις φάσεις της παραγωγής, της μεταποίησης και της διανομής, τη διαδρομή ενός τροφίμου, μιας ζωοτροφής ή ενός ζώο από το οποίο εξάγονται βρώσιμες ουσίες ή ουσίες που πρόκειται ή αναμένεται να ενσωματωθούν σε τρόφιμο που προορίζεται για άνθρωπο ή ζώο.

2ος ορισμός

Η δυνατότητα για πλήρες tracking και tracing, σε κάθε σημείο της εφοδιαστικής αλυσίδας.

Tracking: Η δυνατότητα παρακολούθησης της διαδρομής ενός προϊόντος κατά μήκος της εφοδιαστικής αλυσίδας. Ιδιαίτερα σημαντικό όταν πρόκειται για ανακλήσεις προϊόντων.

Tracing: Η δυνατότητα προσδιορισμού της προέλευσης και των χαρακτηριστικών ενός προϊόντος κατά μήκος της εφοδιαστικής αλυσίδας. Σκοπός είναι ο εντοπισμός των πηγών των προβλημάτων.

ΓΑΛΕΡΓΙΣΤΗΜΟ ΓΕΡΑΛΑ

1.2. ΑΝΑΓΚΗ ΙΧΝΗΛΑΣΙΜΟΤΗΤΑΣ

Σχεδόν σε όλες τις επιχειρήσεις τροφίμων - ζωοτροφών είναι αναγκαία η ύπαρξη ενός συστήματος ιχνηλασιμότητας. Αυτή η επιτακτική ανάγκη αιτιολογείται από τα ακόλουθα:

- ✓ Αριστοποίηση και εκλογίκευση της φυσικής ροής των εμπορευμάτων και της ροής της πληροφορίας.
- ✓ Εξασφάλιση της ασφάλειας των καταναλωτών.
- ✓ Νομοθεσιακές απαιτήσεις:
 - ✎ Οδηγία 2001/CE.
 - ✎ Κανονισμός 178/2002.
- ✓ Ταχεία αντίδραση, για να συνεισφέρουμε στο σεβασμό της υποχρέωσης ασφάλειας.
- ✓ Συμμετοχή στην παρακολούθηση της ασφάλειας των εμπορευμάτων.
- ✓ Μετάδοση στις επίσημες αρχές τις πληροφορίες που αφορούν τον κίνδυνο από προϊόν.
- ✓ Καταγραφή και παροχή των απαραίτητων εγγράφων για τον εντοπισμό της προέλευσης των προϊόντων.
- ✓ Ενδυνάμωση της διατροφικής αλυσίδας.
- ✓ Άμεση αντίδραση και απόσυρση σε κάθε απαίτηση απόσυρσης προϊόντος.
- ✓ Δυνατότητα εντοπισμού ελαττωματικών παρτίδων προϊόντων τα οποία ήδη έχουν αποσταλεί προς τρίτους. Σε περιπτώσεις αδυναμίας εντοπισμού των προϊόντων αυτών, η νομοθεσία προβλέπει την επιβολή κυρώσεων.
- ✓ Δυνατότητα παρακολούθησης της ποιότητας των παραλαμβανόμενων από προμηθευτές προϊόντων, με αποτέλεσμα την εύρεση των πιο αξιόπιστων προμηθευτών.
- ✓ Διασφάλιση των δικαιωμάτων της επιχείρησης από αγωγές πελατών και τρίτων καταναλωτών. Η δυνατότητα Ιχνηλασιμότητας μπορεί να αποφέρει άμεσα κέρδη στην επιχείρηση από την αποφυγή επιβολής κυρώσεων από αρμόδιους φορείς αλλά και καταγγελίες και αγωγές τρίτων.

- ✓ Βελτίωση της κοινωνικής, επαγγελματικής και οικονομικής εικόνας της επιχείρησης.
- ✓ Ορθολογική τήρηση και παρακολούθηση αποθεμάτων και αναγκών αποθήκης.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ

1.3. ΑΡΧΕΣ ΣΩΣΤΗΣ ΙΧΝΗΛΑΣΙΜΟΤΗΤΑΣ

Για να μπορέσει να πετύχει ένα σύστημα ιχνηλασιμότητας, δηλαδή να είναι σε θέση να αναγνωρίσει και να εντοπίσει όλα τα "βήματα" ενός παρακολουθούμενου προϊόντος πρέπει να ακολουθούνται κάποιες συγκεκριμένες αρχές. Αυτές είναι οι εξής:

1. Μοναδική κωδικοποίηση: Κάθε κρίκος αποδίδει ένα μοναδικό κωδικό αριθμό / ταυτότητα.
2. Συνεχή συλλογή και καταγραφή δεδομένων: Κάθε κρίκος συλλέγει και καταγράφει δεδομένα που είτε έχει αποδώσει ο ίδιος είτε κάποιος προηγούμενος κρίκος.
3. Διαχείριση διασυνδέσεων: Οι διασυνδέσεις ανάμεσα στα εισερχόμενα και στα εξερχόμενα προϊόντα πρέπει να καταγράφονται σε βάσεις δεδομένων.
4. Μεταφορά δεδομένων: Τα δεδομένα πρέπει να κοινοποιούνται. Κάθε κρίκος είναι υπεύθυνος για την ορθότητα αλλά και την κοινοποίηση τους.

Με την τήρηση των παραπάνω αρχών είναι σχεδόν σίγουρο πως ένα σύστημα ιχνηλασιμότητας θα λειτουργεί σωστά, χωρίς να χάνονται τα μονοπάτια της αλυσίδας των προϊόντων.

1.4. ΕΦΑΡΜΟΖΟΜΕΝΕΣ ΤΕΧΝΟΛΟΓΙΕΣ ΣΤΗΝ ΙΧΝΗΛΑΣΙΜΟΤΗΤΑ

1.4.1. Αυτοματοποιημένη Αναγνώριση

Τον Φεβρουάριο του 2005 ο EAN International μετονομάστηκε σε GS1. Κατ' επέκταση όλοι οι Εθνικοί Οργανισμοί υιοθέτησαν τη νέα επωνυμία (GS1 + όνομα χώρας), δίνοντας έμφαση στον διεθνή χαρακτήρα του Οργανισμού. Τα αρχικά GS υπογραμμίζουν το ρόλο του οργανισμού στη διαμόρφωση διεθνών προτύπων και συμβολίζουν την αρμονική συνύπαρξη των εμπορικών διαδικασιών με ένα διεθνές σύστημα προτύπων, ανοικτό σε όλους. Το "1" δηλώνει ΕΝΑ διεθνές πρότυπο, ΕΝΑ διεθνή οργανισμό και ΕΝΑ όνομα, δίνοντας έμφαση στο γεγονός ότι οι υπηρεσίες, οι λύσεις, και τα πρότυπα GS1 είναι ολοκληρωμένα και πλήρως συμβατά μεταξύ τους.

Το σύστημα GS1 είναι ένα σύνολο διεθνών προτύπων, το οποίο επιτρέπει την αποτελεσματική διαχείριση των διεθνών εφοδιαστικών αλυσίδων που καλύπτουν πολλούς τομείς της βιομηχανίας, μέσω της αναγνώρισης με μοναδικό τρόπο των προϊόντων / υπηρεσιών, των μονάδων Logistics, των παγίων και των θέσεων. Διευκολύνει τις διαδικασίες του ηλεκτρονικού εμπορίου, συμπεριλαμβανομένης της δυνατότητας πλήρους παρακολούθησης και ιχνηλασιμότητας.

Οι κωδικοί αναγνώρισης μπορούν να απεικονίζονται μέσω συμβόλων barcode, επιτρέποντας έτσι την ανάγνωση τους με ηλεκτρονικά μέσα (scanner) στα σημεία πώλησης, στα σημεία παραλαβής των αποθηκών ή σε οποιαδήποτε άλλα σημεία αυτό απαιτείται, σύμφωνα με τις εμπορικές διαδικασίες, και αποτελούν τα «κλειδιά» για την πρόσβαση σε βάσεις δεδομένων Η/Υ και την αδιαμφισβήτητη αναγνώριση των προϊόντων, σε όλα τα μηνύματα που ανταλλάσσονται σε μια συναλλαγή.

Οι συγκεκριμένοι κωδικοί αναγνώρισης των προϊόντων, υπηρεσιών ή θέσεων είναι:

1. Μοναδικοί: Σε κάθε προϊόν, υπηρεσία ή θέση αντιστοιχεί ένας διαφορετικός μοναδικός κωδικός ανεξάρτητα από την προέλευση ή τον προορισμό του.
2. Μη σημαντικοί: Δεν περιέχουν καμία πληροφορία, αποτελούν όμως το κλειδί για την πρόσβαση σε μια βάση δεδομένων που περιέχει ακριβείς πληροφορίες σχετικά με το προϊόν, υπηρεσία ή θέση.
3. Ασφαλείς: Οι κωδικοί GS1 είναι αριθμητικοί και περιλαμβάνουν ένα ψηφίο ελέγχου που εξασφαλίζει την ορθή ανάκτηση των δεδομένων.
4. Διεθνείς: Οι κωδικοί GS1 είναι διεθνείς και αποδεκτοί οπουδήποτε στον κόσμο και από πολλούς τομείς της βιομηχανίας.

Εκτός από την παροχή μοναδικών κωδικών αριθμών αναγνώρισης, το σύστημα GS1 προβλέπει και την κωδικοποίηση επιπλέον πληροφοριών, όπως οι ημερομηνίες ανάλωσης, οι σειριακοί κωδικοί, οι κωδικοί παρτίδας, κλπ. που μπορούν επίσης να απεικονίζονται με μορφή barcode. Οι πληροφορίες αυτές είναι ιδιαίτερα σημαντικές για την επίτευξη της ικνηλασιμότητας.

Το σύστημα GS1 καλύπτει την κωδικοποίηση και σήμανση των παρακάτω:

- ✓ Μονάδες Εμπορίας
- ✓ Μονάδες Logistics
- ✓ Θέσεις
- ✓ Πάγια
- ✓ Υπηρεσίες

Οι εφαρμογές αυτές εξαρτώνται από τυποποιημένες δομές κωδικοποίησης, μέσω των οποίων μπορούν να αναγνωρίζονται όλα τα είδη και τα αντίστοιχα δεδομένα τους.

Τα τρία κύρια στοιχεία του συστήματος κωδικοποίησης που καλύπτονται στην εργασία αυτή είναι τα εξής:

1. Ο Διεθνής Κωδικός Μονάδων Εμπορίας (GTIN)
2. Ο Σειριακός Κωδικός Μονάδων Logistics (SSCC)
3. Ο Διεθνής Κωδικός Θέσης (GLN)

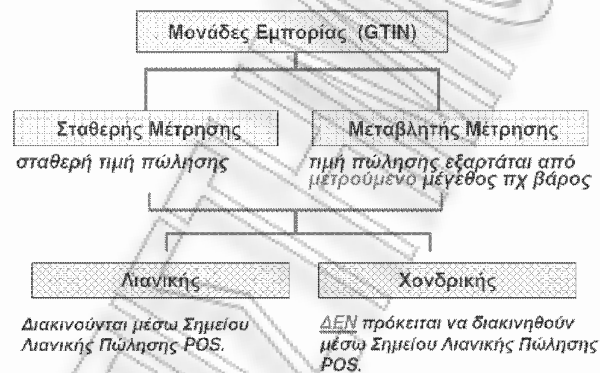
Ακολουθεί λεπτομερέστερη ανάλυση κάθε ενός από τα τρία παραπάνω κύρια στοιχεία του συστήματος κωδικοποίησης GS1.

1.4.1.1. GTIN (Global Trade Item Number)

Οποιαδήποτε «Μονάδα Εμπορίας*», που αποτελεί αντικείμενο εμπορικής συναλλαγής και μπορεί να παραγγελθεί, πωληθεί ή τιμολογηθεί σε οποιοδήποτε σημείο της εφοδιαστικής αλυσίδας και για την οποία υπάρχει ανάγκη πρόσβασης σε προκαθορισμένες πληροφορίες, φέρει ένα μοναδικό κωδικό αριθμό.

Οι κωδικοί αυτοί χρησιμοποιούνται σε όλα τα επίπεδα ιεραρχίας των προϊόντων όπως: μονάδες λιανικής, μονάδες χονδρικής (π.χ. χαρτοκιβώτια) και είναι διαφορετικοί για προϊόντα σταθερής μέτρησης (σταθερή τιμή πώλησης) και για προϊόντα μεταβλητής μέτρησης (η τιμή πώλησης εξαρτάται από μετρούμενο μέγεθος π.χ. βάρος, διαστάσεις, κλπ.).

Η αναγνώριση και η σήμανση των Μονάδων Εμπορίας με σύμβολα barcode επιτρέπει την αυτοματοποίηση των λειτουργιών στα σημεία λιανικής πώλησης και παραλαβής προϊόντων, της διαχείρισης απογραφών, των διαδικασιών ανεφοδιασμού, της ανάλυσης πωλήσεων, της ικνηλασιμότητας και ενός μεγάλου εύρους άλλων επιχειρηματικών εφαρμογών.



Σχήμα 1. Μονάδες εμπορίας

Οι Μονάδες Εμπορίας κωδικοποιούνται μέσω ενός GTIN με χρήση τεσσάρων δομών κωδικοποίησης:

- ✓ GTIN - 13
- ✓ GTIN - 8
- ✓ GTIN - 12
- ✓ GTIN - 14

* Ως Μονάδα Εμπορίας ορίζεται οποιοδήποτε είδος (προϊόν ή υπηρεσία) για το οποίο χρειάζεται να ληφθούν προκαθορισμένες πληροφορίες, είναι αντικείμενο εμπορίας και επομένως μπορεί να τιμολογηθεί ή να παραγγελθεί.

Αν αυτές οι δομές περιλαμβάνονται σε κάποια βάση δεδομένων Η/Υ, αποθηκεύονται όλες σε ένα πεδίο 14 ψηφίων. Η επιλογή της δομής κωδικοποίησης εξαρτάται από τη φύση του είδους και από το πεδίο εφαρμογών του χρήστη.

Στη συνέχεια περιγράφονται οι τέσσερις δομές κωδικοποίησης που προαναφέρθηκαν. Αφού γίνει η επιλογή της δομής για κάποιο είδος και αποδοθεί σε αυτό κάποιος κωδικός, δεν επιτρέπεται η απόδοση κάποιου άλλου κωδικού ή κάποιας άλλης δομής στο ίδιο είδος.

1.4.1.1.1. Προϊόντα Σταθερής Μέτρησης

Η συγκεκριμένη κωδικοποίηση αφορά προϊόντα των οποίων η τιμή πώλησης είναι σταθερή.

1.4.1.1.1.1. Κωδικοποίηση Μονάδας Εμπορίας Λιανικής

GTIN - 13

Εταιρικό πρόθεμα GS1 και Κωδικός αναφοράς είδους	Ψηφίο ελέγχου
$N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12}$	C

Σχήμα 2. GTIN-13

Οι κωδικοί αυτοί αποτελούνται από δεκατρία ψηφία και συμβολίζονται με την συμβολογία EAN - 13.

GTIN - 12

Εταιρικό πρόθεμα UPC και Κωδικός αναφοράς είδους	Ψηφίο ελέγχου
$N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11}$	C

Σχήμα 3. GTIN -12

Οι κωδικοί αυτοί αποτελούνται από δώδεκα ψηφία και συμβολίζονται με τις συμβολογίες UPC - A ή UPC - E.

GTIN - 8

Πρόθεμα χώρας GS1 και Κωδικός αναφοράς είδους	Ψηφίο ελέγχου
$N_1 N_2 N_3 N_4 N_5 N_6 N_7$	C

Σχήμα 4. GTIN - 8

Οι κωδικοί αυτοί αποτελούνται από οκτώ ψηφία και συμβολίζονται με την συμβολογία EAN - 8.

1.4.1.1.1.2. Κωδικοποίηση μονάδας εμπορίας χονδρικής**GTIN - 13**

Η ανάλυση του συγκεκριμένου δομής κωδικοποίησης έγινε παραπάνω.

GTIN - 14

Δείκτης	Κωδικός GTIN των περιεχόμενων ειδών (χωρίς το ψηφίο ελέγχου)	Ψηφίο ελέγχου
I	$N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12}$	C

Σχήμα 5. GTIN - 14

Οι συγκεκριμένοι κωδικοί συμβολίζονται με μια από τις συμβολογίες ITF-14 ή GS - 128.

1.4.1.1.1.3. Ανάλυση Δομών Κωδικοποίησης Μονάδας Εμπορίας Σταθερής Μέτρησης**Δείκτης**

Χρησιμοποιείται μόνο στον κωδικό GTIN - 14. Παίρνει τιμές από 1 έως 8 για Μονάδες Εμπορίας σταθερής μέτρησης και την τιμή 9 για Μονάδες Εμπορίας μεταβλητής μέτρησης. Ο απλούστερος τρόπος είναι να αποδίδεται ο δείκτης σειριακά, δηλαδή 1,2,3,4..., σε κάθε ομαδοποίηση μιας Μονάδας Εμπορίας.

Εταιρικό πρόθεμα GS1

Τα πρώτα δυο ή τρία ψηφία N_1 , N_2 , N_3 , συνιστούν το GS1 πρόθεμα χώρας, το οποίο χορηγείται από τον GS1 σε κάθε Εθνικό Οργανισμό GS1.

Δεν δηλώνει κατ' ανάγκη ότι το είδος παράγεται ή κυκλοφορεί στη χώρα που αντιστοιχεί στο πρόθεμα, αλλά καθορίζει μόνο τον Εθνικό Οργανισμό GS1 που έχει χορηγήσει τον κωδικό εταιρίας.

Το GS1 πρόθεμα χώρας και ο κωδικός εταιρίας σχηματίζουν το εταιρικό πρόθεμα GS1, το οποίο χορηγείται σε κάθε χρήστη του συστήματος. Γενικά, αποτελείται από 6 έως 10 ψηφία, ανάλογα με τις ανάγκες της εταιρίας. Για παράδειγμα, στην εταιρία Α ανάλογα με τις ανάγκες της μπορεί να αποδοθεί το εταιρικό πρόθεμα GS1 520 1234, στην εταιρία Β το 520 12345 και στην εταιρία Γ το 520 123456.

Κωδικός αναφοράς προϊόντος

Ο κωδικός αναφοράς προϊόντος έχει συνήθως 1 έως 6 ψηφία και είναι μη σημαντικός, πράγμα που σημαίνει ότι τα μεμονωμένα ψηφία του κωδικού δεν σχετίζονται με κάποια ταξινόμηση και δεν μεταφέρουν συγκεκριμένες πληροφορίες. Ο απλούστερος τρόπος απόδοσης των κωδικών αναφοράς είδους είναι σειριακά, δηλαδή 000, 001, 002, 003 κλπ.

Ψηφίο ελέγχου

Το ψηφίο ελέγχου είναι το τελευταίο ψηφίο (στα δεξιά) του GTIN. Υπολογίζεται με βάση όλα τα υπόλοιπα ψηφία του κωδικού και χρησιμοποιείται ώστε να εξασφαλίζει τη σωστή ανάγνωση του barcode από τις συσκευές σάρωσης και τη σωστή σύνθεση του κωδικού.

1.4.1.1.2. Προϊόντα μεταβλητής μέτρησης

Ο όρος προϊόντα «μεταβλητής μέτρησης» χρησιμοποιούνται για την περιγραφή προϊόντων τα οποία πωλούνται, παραγγέλλονται ή παράγονται σε ποσότητες οι οποίες μπορούν να μεταβάλλονται συνεχώς, όπως φρούτα, λαχανικά, ψάρια, υφάσματα και μοκέτες σε ρολά, κλπ.

1.4.1.1.2.1. Κωδικοποίηση μονάδας εμπορίας λιανικής

Για τις καταναλωτικές Μονάδες Εμπορίας μεταβλητής μέτρησης, θα πρέπει στο barcode της Μονάδας Εμπορίας να περιέχονται και πληροφορίες σχετικά με το βάρος, το πλήθος ή την τιμή πώλησης, ώστε να αναγνωσθούν στο ταμείο. Στα σύμβολα EAN/UPC όμως δεν υπάρχει αρκετός χώρος για όλες τις πληροφορίες που απαιτούνται για τις Μονάδες Εμπορίας μεταβλητής μέτρησης και έτσι πρέπει να χρησιμοποιηθεί ένας μικρότερος κωδικός για την αναγνώριση του προϊόντος.

Η μέτρηση / διάσταση ή η τιμή πώλησης μπορούν να αποτελούνται από 4 ή 5 ψηφία, ανάλογα με το νομισματικό σύστημα και το σύστημα μονάδων, ενώ μπορεί να περιλαμβάνεται και ένα ειδικό ψηφίο ελέγχου για την τιμή.

Ένας τέτοιος κωδικός για την αναγνώριση προϊόντων μεταβλητής μέτρησης μπορεί να αποδίδεται από:

1. Τον λιανέμπορο (κωδικοποίηση στο σημείο πώλησης).
2. Τον προμηθευτή (κωδικοποίηση στην πηγή).

Τον Εθνικό Οργανισμό GS1, στην περίπτωση που έχει αποφασιστεί ο καθορισμός ενός ενιαίου κωδικού, σε εθνικό επίπεδο, για συγκεκριμένο τύπο προϊόντων, π.χ. όλα τα τυριά του ίδιου τύπου φέρουν τον ίδιο κωδικό ανεξαρτήτως του παραγωγού (στην Ελλάδα δεν ακολουθείται αυτή η λύση).

Για την Ελλάδα, η δομή ενός τέτοιου κωδικού προϊόντος μεταβλητού βάρους, όταν κωδικοποιείται από τον παραγωγό, είναι η ακόλουθη:

Ειδικό Πρόθεμα	Ειδικό εταιρικό πρόθεμα GS1 και Κωδικός αναφοράς είδους	Βάρος	Ψηφίο ελέγχου
29	N ₁ N ₂ N ₃ N ₄ N ₅	B ₁ B ₂ B ₃ B ₄ B ₅	C

Σχήμα 6. Δομή κωδικού προϊόντος μεταβλητού βάρους από παραγωγό

Το ειδικό εταιρικό πρόθεμα GS1 αποδίδεται από τον GS1 Ελλάς και είναι κατά κανόνα τριψήφιο (N₁, N₂, N₃), ενώ ο κωδικός αναφοράς του είδους αποδίδεται από τον παραγωγό και είναι κατά κανόνα διψήφιος (N₄, N₅).

Το βάρος που περιέχεται στον κωδικό μπορεί να είναι έως 99 κιλά (B₁, B₂) και 999 γραμμάρια (B₃, B₄, B₅).

Στην περίπτωση όπου το προϊόν κωδικοποιείται από τον λιανέμπορο, η δομή είναι η ακόλουθη:

Ειδικό Πρόθεμα	Κωδικός αναφοράς είδους	Τιμή πώλησης ή Βάρος	Ψηφίο ελέγχου
02	N ₁ N ₂ N ₃ N ₄ N ₅	T ₁ T ₂ T ₃ T ₄ T ₅	C
20 & 21	N ₁ N ₂ N ₃ N ₄ N ₅	B ₁ B ₂ B ₃ B ₄ B ₅	C

Σχήμα 7. Σχήμα 6. Δομή κωδικού προϊόντος μεταβλητού βάρους από λιανέμπορο

Η τιμή πώλησης που περιέχεται στον κωδικό μπορεί να είναι έως 999 € (T₁, T₂, T₃) και 99 λεπτά (T₄, T₅).

1.4.1.1.2.2. Κωδικοποίηση μονάδας εμπορίας χονδρικής

Για την αναγνώριση μιας Μονάδας Εμπορίας μεταβλητής μέτρησης που δεν προορίζεται για λιανική πώληση, είναι υποχρεωτική η παρουσία της συγκεκριμένης μέτρησης / διάστασης

του είδους. Για την αναγνώριση μιας τέτοιας Μονάδας μεταβλητής μέτρησης χρησιμοποιείται ο κωδικός αναγνώρισης GTIN - 14 με τον δείκτη «9».

Το ψηφίο «9» στην πρώτη θέση αποτελεί αναπόσπαστο τμήμα του 14 - ψηφίου κωδικού των Μονάδων Εμπορίας μεταβλητής μέτρησης. Καθορίζει ότι κάποια πρόσθετη πληροφορία σχετικά με την μέτρηση / διάσταση ή το πλήθος έχει επισυναφθεί στον κωδικό αναγνώρισης ώστε να τον συμπληρώσει.

Όταν για κάποιο συγκεκριμένο είδος μεταβλητής μέτρησης προοριζόμενο για λιανική πώληση υπάρχουν πολλές διαφορετικές Μονάδες Εμπορίας Χονδρικής μεταβλητής μέτρησης, σε κάθε μια από αυτές πρέπει να αποδίδεται ο δικός της GTIN, ο οποίος αρχίζει με το ψηφίο «9».

Η σύνθεση του κωδικού έχει την ακόλουθη δομή, για μια μέτρηση που αντιστοιχεί σε χιλιόγραμμα:

AI	GTIN	AI	Μέτρηση
01	9 N ₁ N ₂ N ₃ N ₄ N ₅ N ₆ N ₇ N ₈ N ₉ N ₁₀ N ₁₁ N ₁₂ C	310x	M ₁ M ₂ M ₃ M ₄ M ₅ M ₆

Σχήμα 8. GTIN – 14 με τον δείκτη «9»

Οι πληροφορίες αυτές απεικονίζονται σε μορφή barcode με έναν από τους παρακάτω δυο τρόπους:

1. Κατά προτίμηση, μέσω ενός συμβόλου GS1 - 128 ώστε ο κωδικός αναγνώρισης (GTIN) και η μέτρηση να περιέχονται σ' ένα μόνο σύμβολο, με χρήση του δείκτη εφαρμογής (01) για τον GTIN και ενός δείκτη εφαρμογής από το (3100) έως το (3169) ή του (8001) για τη μέτρηση.
2. Επίσης, επιτρέπεται η απεικόνιση του GTIN μέσω ενός συμβόλου ITF - 14 και της μέτρησης μέσω ενός συμβόλου GS1 - 128.

Η μέτρηση εκφράζεται πάντοτε μέσω 6 ψηφίων, στη μονάδα μέτρησης που καθορίζεται από τον αντίστοιχο δείκτη εφαρμογής (AI). Ένας Δείκτης Εφαρμογής είναι ένα πεδίο δυο ή περισσότερων χαρακτήρων στην αρχή μιας σειράς δεδομένων. Οι Δείκτες Εφαρμογής είναι «προθέματα» τα οποία προσδιορίζουν κατά μοναδικό τρόπο τα δεδομένα που ακολουθούν, το νόημα τους, το μήκος τους και την εσωτερική δομή τους. Οι δείκτες εφαρμογής έχουν αρκετά πλεονεκτήματα όπως: εύκολη και ακριβής απόκτηση πληροφοριών, ένα σύμβολο για όλες τις πληροφορίες, καλύτερη ανταλλαγή πληροφοριών, εύκολη ανίχνευση θέσης προϊόντων, μοναδικός προσδιορισμός συνδυασμών προϊόντων, βελτιωμένη διαχείριση αποθήκευσης και διανομής. Η θέση της υποδιαστολής υποδεικνύεται από το τελευταίο ψηφίο (x) του δείκτη εφαρμογής. Αν έχει τιμή 0 σημαίνει ότι δεν υπάρχει υποδιαστολή, ενώ αν έχει την τιμή 3 σημαίνει ότι υπάρχουν τρία δεκαδικά ψηφία.

Παρακάτω αναφέρουμε ορισμένους από τους τυπικούς δείκτες εφαρμογής (AI).

AI	Πλήρης Τίτλος	Μορφή*
00	Σειριακός Κωδικός Μονάδων Logistics - SSCC	$n2+n18$
01	Διεθνής Κωδικός Μονάδων Εμπορίας - GTIN	$n2+n14$
02	GTIN Μονάδων Εμπορίας που περιλαμβάνονται σε μια Μονάδα Logistics	$n2+n14$
10	Αριθμός Παρτίδας (Batch/Lot)	$n2+an..20$
11	Ημερομηνία παραγωγής (EEMMHH)	$n2+n6$
15	Ημερομηνία ανάλωσης κατά προτίμηση (EEMMHH)	$n2+n6$
17	Ημερομηνία λήξης (EEMMHH)	$n2+n6$
21	Σειριακός αριθμός	$n2+an..20$
310 X**	Καθαρό Βάρος (kg)	$n4+n6$
37	Πλήθος Μονάδων Εμπορίας που περιλαμβάνονται σε μια Μονάδα Logistics	$n2+n..8$
401	Κωδικός Φορτίου	$n3+an..30$
420	Ταχυδρομικός κωδικός "Αποστολή προς - Παράδοση σε"	$n3+an..20$

Σχήμα 9. Τυπικοί δείκτες εφαρμογής (AI)

Στον παραπάνω πίνακα το n δηλώνει αριθμητικούς χαρακτήρες, ενώ το an δηλώνει αλφαριθμητικούς χαρακτήρες.

1.4.1.1.3. Αλλαγή κωδικού Μονάδας Εμπορίας

Ο γενικός κανόνας ορίζει ότι, σε περίπτωση που αλλάξει οποιοδήποτε από τα προκαθορισμένα χαρακτηριστικά μιας Μονάδας Εμπορίας που σχετίζονται με την εμπορική διαδικασία, απαιτείται ένας διαφορετικός, μοναδικός GTIN. Αυτό συνεπάγεται ότι σε μια παραλλαγή κάποιου προϊόντος θα πρέπει να αποδίδεται ένας διαφορετικός κωδικός, όταν αυτή η παραλλαγή είναι κατά οποιονδήποτε τρόπο εμφανής και σημαντική για οποιονδήποτε εταίρο στην εφοδιαστική αλυσίδα, έως τον τελικό χρήστη ή τον καταναλωτή.

Η έννοια της προφανούς και σημαντικής παραλλαγής μπορεί να διαφέρει από βιομηχανία σε βιομηχανία. Παρ' όλα αυτά, υπάρχουν κανόνες οι οποίοι θα πρέπει να τηρούνται.

Νέα κωδικοποίηση σε μια Μονάδα Εμπορίας γίνεται όταν αλλάζει κάποιο από τα βασικά χαρακτηριστικά της. Ορισμένα από τα χαρακτηριστικά αυτά είναι:

- ✓ Η εμπορική επωνυμία / μάρκα και η περιγραφή του προϊόντος.
- ✓ Ο τύπος του προϊόντος.
- ✓ Οι διαστάσεις της συσκευασίας και η φύση της.

- ✓ Η ποσότητα του προϊόντος.
- ✓ Αν η Μονάδα Εμπορίας είναι μια ομαδοποίηση, το πλήθος των περιεχόμενων στοιχειωδών ειδών και η υποδιαίρεση τους σε επιμέρους Μονάδες συσκευασίας, φύση της ομαδοποίησης (χαρτοκιβώτιο, παλέτα - κιβώτιο, επίπεδη παλέτα κλπ.).

Όπως αναφέραμε παραπάνω μια σημαντική μετατροπή σε κάποιο από τα βασικά στοιχεία που χαρακτηρίζουν τη Μονάδα Εμπορίας οδηγεί συνήθως σε απόδοση νέου κωδικού, παράδειγμα:

- ✓ Απόδοση νέου κωδικού - αυξήθηκε η καθαρή ποσότητα ενός μεταλλικού νερού από 250ml σε 500ml.
- ✓ Διατήρηση του ίδιου κωδικού - το χρώμα μιας ετικέτας σε συσκευασία χυμού αλλάζει από ανοικτό ροζ σε σκούρο ροζ.

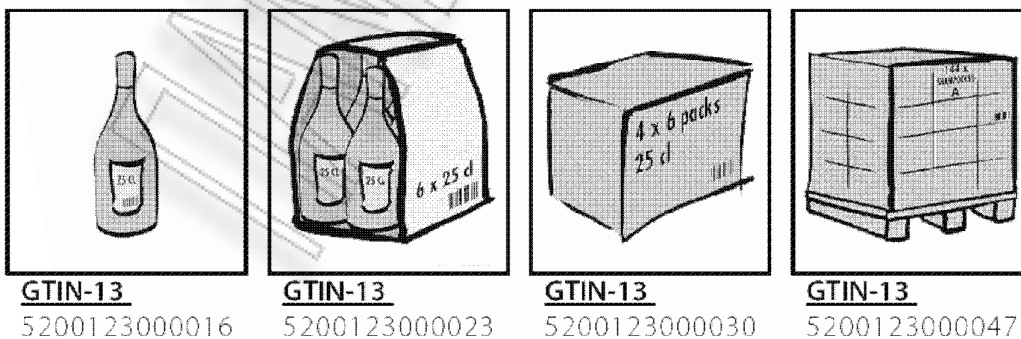
Από τη στιγμή που θα αποδοθεί ένας νέος GTIN σε ένα είδος, τότε θα πρέπει να δοθεί και ένας νέος κωδικός σε κάθε ομαδοποιημένη συσκευασία που περιέχει το συγκεκριμένο είδος.

Εντούτοις, θα πρέπει να σημειωθεί ότι:

- ✓ Αλλαγή της τιμής πώλησης δεν αποτελεί κριτήριο για την αλλαγή ενός GTIN, εκτός εάν αυτή είναι απ' ευθείας εκτυπωμένη στη συσκευασία του προϊόντος.
- ✓ Κρατικοί, ομοσπονδιακοί ή τοπικοί κανονισμοί, έχουν προτεραιότητα ως προς αυτές τις οδηγίες. Για παράδειγμα, σε βιομηχανίες όπως αυτή της υγείας, η νομοθεσία ή άλλες απαιτήσεις ίσως υπαγορεύουν την απόδοση νέου GTIN για οποιαδήποτε αλλαγή στο είδος.

1.4.1.1.4. Επιλογή μεταξύ των δομών κωδικοποίησης GTIN

Μια Μονάδα Εμπορίας μπορεί να περιέχεται σε κάποια άλλη. Κάθε επίπεδο συσκευασίας Μονάδων Εμπορίας θα πρέπει να διαθέτει τον δικό του GTIN.

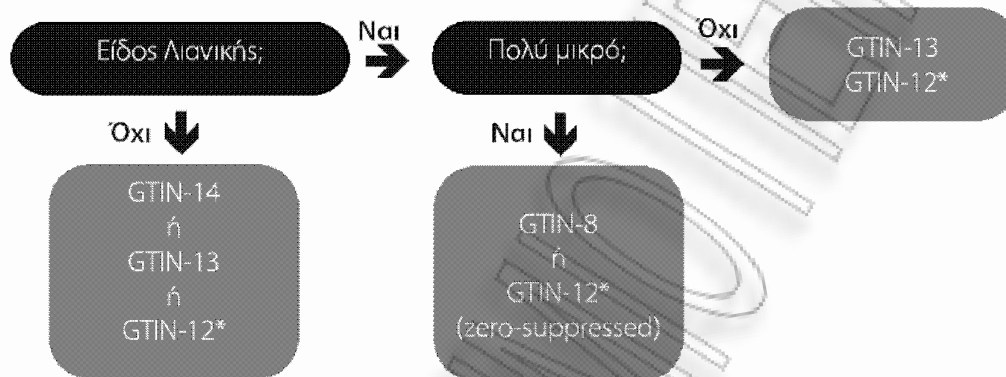


Σχήμα 10. Επιλογή μεταξύ των δομών κωδικοποίησης GTIN

Μπορούν να χρησιμοποιηθούν κωδικοί GTIN - 13, GTIN - 12, GTIN - 8 ή GTIN - 14, ανάλογα με το αν είναι καταναλωτικό είδος ή όχι και στην περίπτωση που δεν είναι, ανάλογα με τη μέθοδο κωδικοποίησης που έχει υιοθετηθεί από την εταιρία.

Οι εταιρίες θα πρέπει να λαμβάνουν υπ' όψιν τους τα κανάλια μέσω των οποίων πωλούνται τα είδη και τις υπάρχουσες προϋποθέσεις.

Το δενδροειδές διάγραμμα που ακολουθεί, παρουσιάζει τις πιθανές επιλογές για έναν κωδικό.



Σχήμα 11. Διάγραμμα επιλογής κωδικών

*Αποδίδεται από τους Εθνικούς Οργανισμούς GS1 των ΗΠΑ και του Καναδά.

1.4.1.1.5. Υπεύθυνος για την κωδικοποίηση των Μονάδων Εμπορίας

Η εταιρία στην οποία ανήκει η εμπορική επωνυμία ενός προϊόντος, ασχέτως με το που και από ποιόν κατασκευάζεται το προϊόν, είναι υπεύθυνη για την απόδοση του GTIN.

Κάθε Εθνικός Οργανισμός GS1 εκδίδει ένα εταιρικό πρόθεμα στην εταιρία - ιδιοκτήτη της επωνυμίας του προϊόντος για αποκλειστική της χρήση. Το πρόθεμα αυτό δεν μπορεί να πωληθεί, ενοικιαστεί ή παραχωρηθεί, ολόκληρο ή μέρος του, για χρήση από οποιαδήποτε άλλη εταιρία.

Ως «ιδιοκτήτης της εμπορικής επωνυμίας» νοείται:

1. Ο κατασκευαστής ή ο προμηθευτής. Αν κατασκευάζει ο ίδιος το προϊόν ή αναθέτει την παραγωγή του σε οποιαδήποτε άλλη εταιρία, αλλά το πωλεί με κάποια εμπορική επωνυμία η οποία ανήκει στον ίδιο.
2. Ο εισαγωγέας ή ο έμπορος χονδρικής. Αν ο εισαγωγέας ή ο έμπορος χονδρικής αναθέτει την παραγωγή του προϊόντος σε οποιαδήποτε χώρα και το πωλεί με κάποια εμπορική επωνυμία η οποία ανήκει σε αυτόν.

3. Ο λιανέμπορος. Αν ο λιανέμπορος αναθέτει την παραγωγή του προϊόντος σε οποιαδήποτε χώρα και το πωλεί με κάποια εμπορική επωνυμία που ανήκει σε αυτόν.

Στο σημείο αυτό αξίζει να αναφέρουμε κάποιες εξαιρέσεις στους παραπάνω κανόνες για το ποιος είναι υπεύθυνος απόδοσης των κωδικών GTIN:

1. Αν σε κάποιο είδος δεν αποδοθεί κάποιος GTIN «στην πηγή», ο εισαγωγέας ή ο διακινητής μπορεί, κατ' απαίτηση των πελατών του, να του αποδώσει έναν προσωρινό GTIN. Είναι όμως προτιμότερο, η απόδοση του κωδικού να γίνεται από τον κατασκευαστή.
2. Ο λιανέμπορος μπορεί να αποδώσει έναν εσωτερικό κωδικό, για χρήση εντός του καταστήματος, σε κάποιο είδος στο οποίο δεν έχει αποδοθεί ακόμη κάποιος GTIN.
3. Τα είδη που δεν διαθέτουν εμπορική επωνυμία και τα generic items κωδικοποιούνται από τον κατασκευαστή τους.

1.4.1.2. GLN (Global Location Number)

Ο Διεθνής Κωδικός Θέσης (GLN), καθιστά εφικτή τη μοναδική και αδιαμφισβήτητη αναγνώριση φυσικών, λειτουργικών ή νομικών οντοτήτων.

Μια εμπορική συναλλαγή περιλαμβάνει διάφορους εμπλεκόμενους όπως, προμηθευτές, πελάτες, πιθανόν έναν παροχέα υπηρεσιών logistics, κλπ. Παράλληλα, σε κάθε εταιρία, μπορεί να εμπλέκονται διάφορα τμήματα της στη συγκεκριμένη εμπορική συναλλαγή.

Οι εμπορικοί εταίροι λοιπόν, χρειάζεται να προσδιορίζουν με ακρίβεια στα αντίστοιχα αρχεία τους, όλες τις «θέσεις» και όλες τις λειτουργίες που σχετίζονται με αυτή την εμπορική συναλλαγή. Για το σκοπό αυτό χρησιμοποιείται η τυποποιημένη δομή κωδικοποίησης GTIN -13 όπως και στην περίπτωση της κωδικοποίησης των Μονάδων Εμπορίας. Δεν υπάρχει όμως περιορισμός ως προς την απόδοση του ίδιου κωδικού αναγνώρισης GTIN -13 σε κάποιο προϊόν και σε κάποια «θέση» διότι οι εφαρμογές είναι εντελώς ξεχωριστές και δεν υπάρχει καμία σύγχυση. Παρ' όλα αυτά όμως συνίσταται να αποφεύγεται η χρήση του ίδιου 13 - ψήφιου αριθμού για την κωδικοποίηση ενός προϊόντος και μιας «θέσης».

Κάθε εταιρία ή οργανισμός που είναι κάτοχος ενός εταιρικού προθέματος GS1 μπορεί να αποδίδει Διεθνείς Κωδικούς Θέσης (GLN) στις δικές της «θέσεις». Σε κάθε διαφορετική

διεύθυνση, σε κάθε λειτουργία η οποία μπορεί να διακρίνεται από άλλες, θα πρέπει να αποδίδεται ένας διαφορετικός κωδικός GLN.

Η πληροφόρηση όλων των εμπορικών εταιριών για όλους τους χρησιμοποιούμενους κωδικούς και τις σχετικές τους λεπτομέρειες αποτελεί ευθύνη της εταιρίας που αποδίδει τους κωδικούς.

Οι GLNs χρησιμοποιούνται σε μορφή barcode μόνο με τη συμβολογία GS1 - 128, με δείκτες εφαρμογής (AI) που έχουν καθοριστεί για GLNs, όπως:

1. «Παράδοση σε...» AI (410).
2. «Τιμολόγηση προς...» AI (411).
3. «Αγορά από...» AI (412).
4. «Αποστολή για... - Παράδοση για... - Προώθηση σε...» AI (413).
5. «Φυσική θέση» AI (414).
6. «Κωδικός θέσης του τιμολογούντος» AI (415).

Η σύνθεση του κωδικού έχει την ακόλουθη δομή:

Εταιρικό πρόθεμα GS1 και Κωδικός αναφοράς είδους	Ψηφίο ελέγχου
$N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12}$	C

Σχήμα 12. GLN (Global Location Number)

Η χρήση των κωδικών θέσης αποτελεί προϋπόθεση για μια αποτελεσματική συναλλαγή μέσω EDI (Ηλεκτρονική Ανταλλαγή Δεδομένων).

1.4.1.3. SSCC (Serial Shipping Container Code)

Ο SSCC είναι ένας τυποποιημένος κωδικός αναγνώρισης, ο οποίος χρησιμοποιείται για την αναγνώριση με μοναδικό τρόπο των Μονάδων Logistics[†], απεικονίζεται δε με την χρήση της συμβολογίας GS1 - 128.

[†] Μια Μονάδα Logistics είναι ένα είδος οποιασδήποτε σύνθεσης, το οποίο δεν αποτελεί αντικείμενο εμπορίας αλλά προορίζεται για μεταφορά και / ή αποθήκευση και η διαχείριση του οποίου μπορεί να γίνεται μέσα σε όλη την εφοδιαστική αλυσίδα.

Ο κωδικός αυτός είναι μοναδικός για κάθε συγκεκριμένη Μονάδα Logistics και είναι επαρκής για όλες τις εφαρμογές logistics.

Ο κωδικός SSCC είναι ο μόνος απολύτως απαραίτητος κωδικός για μια Μονάδα Logistics, υπό την προϋπόθεση ότι όλοι οι εμπορικοί εταίροι, συμπεριλαμβανομένων των μεταφορέων και των εμπλεκόμενων τρίτων, έχουν την δυνατότητα ανάγνωσής του. Ταυτόχρονα ανταλλάσσουν μηνύματα EDI που δίνουν πλήρεις περιγραφές των Μονάδων Logistics, έχουν δε on - line πρόσβαση στο σχετικό αρχείο κατά την ανάγνωση των SSCCs. Όμως, επειδή σπάνια πληρούνται όλες αυτές οι συνθήκες, είναι γενικά αποδεκτό ότι πέραν του SSCC είναι χρήσιμο να υπάρχουν πάνω στις Μονάδες Logistics και πρόσθετες πληροφορίες σε μορφή barcode.

Επειδή σε κάθε Μονάδα Logistics πρέπει να αποδίδεται ο δικός της μοναδικός SSCC, η προεκτύπωση του στη συσκευασία της Μονάδας δεν είναι εφικτή. Αντιθέτως, θα πρέπει να δημιουργείται κάποια ετικέτα, η οποία θα επικολλάται στη Μονάδα Logistics την ώρα της δημιουργίας της π.χ. του palletisation.

Είναι επίσης πιθανό μια Μονάδα Logistics να είναι και Μονάδα Εμπορίας, οπότε θα υπόκειται στις προδιαγραφές GS1 για Μονάδες Εμπορίας. Σε αυτή την περίπτωση, είναι λογικό να δημιουργείται μια ετικέτα η οποία θα περιλαμβάνει όλες τις απαιτούμενες πληροφορίες σε μορφή barcode.

Ο GS1, σε συνεργασία με εκπροσώπους των κατασκευαστών, λιανέμπορους, μεταφορείς και με τους κατά τόπους Εθνικούς Οργανισμούς GS1 έχουν αναπτύξει ένα εθελοντικό πρότυπο για τις εφαρμογές των ετικετών barcode στις Μονάδες Logistics. Το πρότυπο αυτό ονομάζεται «Πρότυπη Ετικέτα Logistics GS1» και αποτελεί τμήμα των Γενικών Προδιαγραφών GS1. Ο SSCC, και η εφαρμογή του στις Μονάδες Logistics, είναι το πιο σημαντικό στοιχείο της Πρότυπης Ετικέτας Logistics GS1.

Η ανάγνωση του SSCC επιτρέπει την παρακολούθηση και τον εντοπισμό των μεμονωμένων Μονάδων, όταν αυτές διακινούνται, παρέχοντας έναν «σύνδεσμο» μεταξύ της φυσικής ροής των προϊόντων και της ροής των σχετικών πληροφοριών. Επίσης, παρέχει την ευκαιρία υλοποίησης μια ευρείας κλίμακας εφαρμογών, όπως το cross docking, τη δρομολόγηση των αποστολών, την αυτόματη παραλαβή κλπ.

Παρακάτω απεικονίζεται και αναλύεται η μορφή του σειριακού κωδικού SSCC.

Δ.Ε.	Σειριακός Κωδικός Μονάδων Logistics		
	Ψηφίο επέκτασης	Εταιρικό πρόθεμα GS1 & αύξων αριθμός Μονάδας Logistics	Ψηφίο ελέγχου
00	N ₁	N ₂ N ₃ N ₄ N ₅ N ₆ N ₇ N ₈ N ₉ N ₁₀ N ₁₁ N ₁₂ N ₁₃ N ₁₄ N ₁₅ N ₁₆ N ₁₇	C

Σχήμα 13. Μορφή του σειριακού κωδικού SSCC

Το ψηφίο επέκτασης χρησιμοποιείται για την αύξηση του πλήθους των διαθέσιμων SSCCs. Αποδίδεται από την εταιρία που καθορίζει τον SSCC. Μέσω αυτού, χρησιμοποιώντας τιμές από 0 έως και 9, δεκαπλασιάζονται οι διαθέσιμοι SSCCs για την κάθε εταιρία.

Το εταιρικό πρόθεμα GS1 χορηγείται από έναν Εθνικό Οργανισμό GS1 στον χρήστη του συστήματος, ο οποίος είναι συνήθως η εταιρία που δημιουργεί τη Μονάδα Logistics. Αυτό κάνει τον κωδικό παγκοσμίως μοναδικό, αλλά δεν προσδιορίζει την προέλευση της Μονάδας Logistics.

Ο αύξων αριθμός Μονάδας Logistics είναι ένας σειριακός αριθμός που δημιουργείται από την εταιρία στην οποία έχει χορηγηθεί το εταιρικό πρόθεμα, ώστε να συμπληρώσει την σειρά των ψηφίων N₂ έως N₁₇. Ο απλούστερος τρόπος καθορισμού των αριθμών αναφοράς είναι με αύξουσα αρίθμηση δηλαδή 000, 001, 002, 003..., κλπ.

1.4.1.3.1. Ετικέτες Μονάδων Logistics

1.4.1.3.1.1. Απεικόνιση των πληροφοριών

Οι πληροφορίες που περιλαμβάνονται σε ετικέτες Logistics παίρνουν δυο βασικές μορφές:

1. Πληροφορίες αναγνώσιμες από τον άνθρωπο οι οποίες αποτελούνται από κείμενο και γραφικά.
2. Πληροφορίες αναγνώσιμες από τα μηχανήματα σχεδιασμένες για συστήματα αυτόματης συλλογής δεδομένων.

Τα barcode, ως σύμβολα αναγνώσιμα από μηχανές, είναι μια ασφαλής και αποδοτική μέθοδος μεταφοράς δομημένων δεδομένων. Τα σύμβολα αυτά, όπως και το αναγνώσιμο από τον άνθρωπο κείμενο επιτρέπουν γενική πρόσβαση σε βασικές πληροφορίες σε κάθε σημείο της εφοδιαστικής αλυσίδας. Και οι δυο αυτές μέθοδοι απεικόνισης των πληροφοριών προσθέτουν αξία στις ετικέτες logistics, και συχνά συνυπάρχουν στην ίδια ετικέτα.

Η πρότυπη ετικέτα logistics GS1 χωρίζεται σε τρία τμήματα. Το επάνω τμήμα της περιέχει ελεύθερο κείμενο. Το μεσαίο, πληροφορίες σε μορφή κειμένου και ερμηνεία των barcode σε μορφή αναγνώσιμη από τον άνθρωπο. Το κάτω τμήμα περιλαμβάνει τα barcode και τις αντίστοιχες πληροφορίες.

1.4.1.3.1.2. Σχεδιασμός της ετικέτας

Η διάταξη της ετικέτας logistics υποστηρίζει την επεξεργασία των πληροφοριών μέσα στην εφοδιαστική αλυσίδα, ομαδοποιώντας τις πληροφορίες σε τρία λογικά τμήματα, για τον προμηθευτή, τον πελάτη και τον μεταφορέα. Κάθε τμήμα της ετικέτας μπορεί να επικολλλάται σε διαφορετικές χρονικές στιγμές καθώς γίνονται γνωστές οι πληροφορίες. Επιπλέον, σε κάθε τμήμα τα barcode είναι διαχωρισμένα από τις πληροφορίες σε μορφή κειμένου, ώστε να διευκολύνεται η ξεχωριστή επεξεργασία από ανθρώπους και μηχανήματα.

Ένα τμήμα είναι μια λογική ομαδοποίηση πληροφοριών οι οποίες είναι γενικά γνωστές σε μια δεδομένη στιγμή. Υπάρχουν τρία τμήματα στις ετικέτες, κάθε ένα από τα οποία απεικονίζει μια ομάδα πληροφοριών. Γενικά η σειρά αυτών των τμημάτων, από πάνω προς τα κάτω είναι: Μεταφορέας. Πελάτης, Προμηθευτής. Όμως, αυτή η σειρά και η στοίχιση πάνω / κάτω μπορεί να ποικίλει ανάλογα με το μέγεθος της Μονάδας Logistics και την εξυπηρετούμενη διαδικασία. Ακολουθεί λεπτομερή ανάλυση κάθε ενός από τα τρία τμήματα της ετικέτας Μονάδας Logistics.

Τμήμα προμηθευτή

Οι πληροφορίες που περιλαμβάνονται σε αυτό το τμήμα είναι γενικά γνωστές την ώρα της συσκευασίας από τον προμηθευτή. Ο υποχρεωτικός SSCC απεικονίζεται εδώ, σαν κωδικός αναγνώρισης της Μονάδας Logistics. Ο κωδικός αναγνώρισης της Μονάδας Εμπορία (GTIN) θα πρέπει επίσης να απεικονίζεται εδώ, όταν χρησιμοποιείται.

Στο τμήμα αυτό μπορούν να συμπεριληφθούν και άλλες πρόσθετες πληροφορίες, οι οποίες παρουσιάζουν ιδιαίτερο ενδιαφέρον για τον προμηθευτή, αλλά είναι εξίσου χρήσιμες στους πελάτες και στους μεταφορείς. Οι πληροφορίες αυτές μπορεί να είναι σχετικές με τα προϊόντα, όπως παραλλαγή προϊόντος, ημερομηνίες παραγωγής, συσκευασία, λήξης και ανάλωσης, καθώς και κωδικοί παρτίδας και σειράς.

Τμήμα πελάτη

Οι πληροφορίες που περιλαμβάνονται σε αυτό το τμήμα είναι γενικά γνωστές την ώρα της παραγγελίας και της επεξεργασίας της παραγγελίας από τον προμηθευτή. Περιλαμβάνουν τον προορισμό της αποστολής, τον κωδικό παραγγελίας και πληροφορίες που αφορούν κάθε πελάτη ξεχωριστά και σχετίζονται με τα δρομολόγια και τον τρόπο χειρισμού των αγαθών.

Τμήμα μεταφορέα

Οι πληροφορίες που περιλαμβάνονται σε αυτό το τμήμα είναι γενικά γνωστές την ώρα της μεταφοράς και συνήθως σχετίζονται με αυτήν. Περιλαμβάνουν ταχυδρομικούς κωδικούς αποστολής, κωδικούς φορτίου και πληροφορίες που αφορούν τον μεταφορέα και σχετίζονται με τα δρομολόγια και τον τρόπο χειρισμού των αγαθών.

1.4.1.3.1.3. Παράδειγμα ετικετών



Σχήμα 14. Πρότυπη ετικέτα Μονάδας Logistics GS1



Σχήμα 15. Ετικέτα με τμήματα προμηθευτή, πελάτη και μεταφορέα

Ο υπεύθυνος για τις ετικέτες (η εταιρία που έχει την ευθύνη της εκτύπωσης και της επικόλλησης των ετικετών) καθορίζει το περιεχόμενο, τη μορφή και τις διαστάσεις της ετικέτας.

Ο SSCC είναι το μόνο στοιχείο που πρέπει να περιλαμβάνεται υποχρεωτικά σε όλες τις ετικέτες logistics GS1. Άλλες πληροφορίες μπορούν να προστεθούν, όταν απαιτείται, προς συμμόρφωση με τις Γενικές Προδιαγραφές του Συστήματος GS1.

1.4.2. Αυτόματη Συλλογή Δεδομένων (Automatic Data Capture - ADC)

1.4.2.1. Barcode

Ο γραμμωτός κώδικας που διεθνώς ονομάζεται barcode είναι ένα σύστημα αυτόματης οπτικής αναγνώρισης κωδικών αριθμών που αντιστοιχούν σε περιγραφή των χαρακτηριστικών των προϊόντων. Πλεονεκτήματα του συγκεκριμένου συστήματος είναι:

1. Οικονομία.
2. Ευκολία χρήσης.
3. Ακρίβεια και ασφάλεια.
4. Συνεχής παρακολούθηση των μετακινήσεων των αντικειμένων στις διάφορες φάσεις της εφοδιαστικής αλυσίδας.
5. Γνώση της θέσης τους ανά πάσα χρονική στιγμή.

6. Μείωση του χρόνου συλλογής πληροφοριών, γεγονός που συνεπάγεται αύξηση της ταχύτητας διέλευσης των αντικειμένων.
7. Ακριβής πληροφόρηση, καθώς εκλείπουν τελείως τα λάθη αντιγραφής και πληκτρολόγησης.
8. Αύξηση του βαθμού λεπτομέρειας της πληροφορίας.
9. Άμεση μεταβίβαση (real time) της πληροφορίας, εάν αυτό απαιτείται.
10. Αυτοματοποίηση αναφορών (reporting).

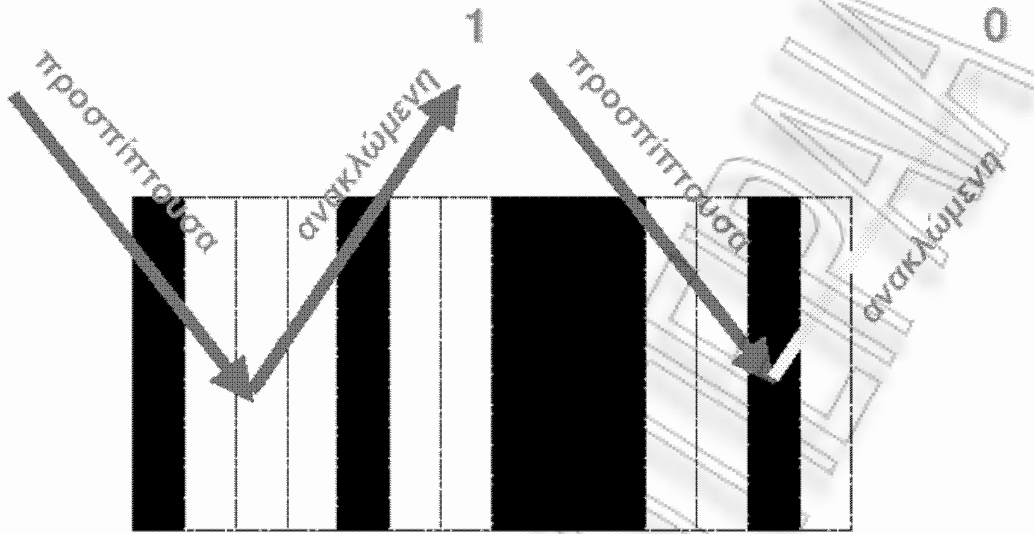
Ενδεικτικά αναφέρουμε ότι κατά την πληκτρολόγηση υπάρχει πιθανότητα λάθους 1/300, ενώ κάνοντας χρήση των barcode η πιθανότητα λάθους μειώνεται στο 1/5000000.

1.4.2.1.1. Λειτουργία

Κατ' αρχάς, στην πιο συνήθη του μορφή, κάθε barcode συνίσταται από μια διαδοχή μαύρων και λευκών λωρίδων (bars) τυπωμένων πάνω σε κάποιο προϊόν / συσκευασία. Ανάμεσα στις λωρίδες παρεμβάλλονται κενά διαστήματα ποικίλου μεγέθους. Η σχεδίαση, εκτύπωση και απεικόνιση των λωρίδων ακολουθούν συγκεκριμένους κανόνες και κάθε αλληλουχία λωρίδων αντιστοιχεί αμφιμονοσήμαντα σε κάποιον αριθμό. Αυτό στην πράξη σημαίνει ότι οι αριθμοί μπορούν να συμβολίζονται από διαφορετική κάθε φορά αλληλουχία λωρίδων: άλλοτε μία μαύρη λεπτή λωρίδα μαζί με μία λευκή παχιά λωρίδα μπορούν να αντιστοιχούν στον αριθμό 6, άλλοτε συμβαίνει ο ίδιος συνδυασμός να αντιστοιχεί στον αριθμό 8 κ.ο.κ. Ο πιο κοινός τύπος γραμμωτού κώδικα είναι ο EAN (European Article Numbering), ο οποίος αποτελείται από αριθμοσειρά 13 ψηφίων και απαντάται στα περισσότερα καταναλωτικά είδη. Ωστόσο, υπάρχουν και άλλοι τύποι barcode με περισσότερα ή και λιγότερα ψηφία από τα 13 του προτύπου EAN, τα οποία βρίσκουν εφαρμογή σε εξειδικευμένους επιχειρηματικούς κλάδους, όπως λ.χ. στη σιδηροδρομική βιομηχανία.

Η ανάγνωση του barcode συντελείται ως εξής: ο κωδικός EAN μεταφράζεται μέσα σε κλάσματα δευτερολέπτου από κάποιο scanner σε γλώσσα Η/Υ (δηλ. 0, 1 - δυαδικό σύστημα). Η ανάγνωση του barcode συνίσταται στην αποκωδικοποίηση της ανάκλασης μιας δέσμης ακτίνων laser που "πέφτει" πάνω στην ετικέτα barcode. Ο βαθμός ανάκλασης είναι μεταβλητός (άρα ξεχωριστός για κάθε προϊόν) γιατί η δέσμη laser συναντά διαδοχικά τις λευκές και μαύρες λωρίδες. Τα scanners αποκωδικοποιούν τη μεταβλητή ανάκλαση (σήμα) και τη μετατρέπουν σε αριθμούς ή γράμματα, τα οποία ταυτίζονται ως προς το περιεχόμενο με τους χαρακτήρες που κωδικοποιήθηκαν με μορφή barcode. Έτσι, όταν ένα προϊόν περνά από το ταμείο λαμβάνει χώρα η εξής διαδικασία: αναγνώριση, ανάγνωση, μετάφραση και αποθήκευση, που σημαίνεται από το χαρακτηριστικό "μπιπ" που ακούγεται κάθε φορά που η ταμίας χρησιμοποιεί το scanner. Τη συγκεκριμένη στιγμή ανιχνεύεται το είδος από τη βάση δεδομένων και διατίθενται για εκτύπωση τα επιμέρους προσδιοριστικά στοιχεία του, όπως η περιγραφή του προϊόντος, η τιμή πώλησής του, η ποσότητα κ.λ.π., στοιχεία που

καταγράφονται στην ταμειακή μηχανή. Συγχρόνως, αφαιρείται το προϊόν από τη βάση δεδομένων της αποθήκης και των αποθεμάτων.



Σχήμα 16. Προσπίπτουσα και ανακλώμενη δέσμη φωτός



Σχήμα 17. Περιγραφή δομής barcode

1.4.2.1.2. Μεγέθη

Τα barcode μπορούν να εκτυπώνονται σε διάφορα μεγέθη. Το περιβάλλον ανάγνωσης και οι συνθήκες εκτύπωσης καθορίζουν το μέγεθος του συμβόλου. Μια καλής ποιότητα εκτύπωση σε συνδυασμό με ένα καλής ποιότητας υπόστρωμα, επιτρέπει να χρησιμοποιηθεί barcode μικρού μεγέθους.

Δεν είναι σωστή η επιλογή ενός αυθαίρετου μεγέθους, με μοναδικό κριτήριο το σύμβολο να ταιριάζει σε κάποιον προκαθορισμένο χώρο πάνω στη συσκευασία δεν είναι σωστή.

Για κάθε τύπο barcode, το μέγεθος μπορεί να ποικίλει μεταξύ μιας ελάχιστης και μιας μέγιστης τιμής. Για την άμεση εκτύπωση, το μέγεθος καθορίζεται από τον υπεύθυνο εκτύπωσης μετά από δοκιμές. Τα μηχανήματα που εκτυπώνουν barcode μέσω pixels ή κουκίδων δεν είναι σε θέση να παράγουν barcode σε όλη την κλίμακα των επιτρεπόμενων μεγεθών.

Ένας άλλος παράγοντας που πρέπει να λαμβάνεται υπ' όψιν στην απόφαση για το μέγεθος του barcode, θα πρέπει να είναι το περιβάλλον στο οποίο το σύμβολο αυτό θα διαβαστεί. Σύμβολα που προορίζονται για το λιανικό εμπόριο μπορούν να είναι όσο μικρά επιτρέπει η ποιότητα εκτύπωσης, ενώ αυτά που προορίζονται για περιβάλλοντα αποθήκης θα πρέπει να είναι όσο το δυνατόν μεγαλύτερα ώστε να επιτρέπουν την ανάγνωση από μεγαλύτερη απόσταση, π.χ. από έναν χειριστή περνοφορού.

Μέχρι το 1999, το μέγεθος ενός συμβόλου barcode στο σύστημα GS1 εκφραζόταν ως Συντελεστής Μεγέθυνσης. Ο συντελεστής αυτός προϋπέθετε μια σταθερή αναλογία μεταξύ της διάστασης X ενός συμβόλου (δηλαδή τη μικρότερη σε πλάτος μπάρα του) και του ύψους του, γεγονός που δεν ισχύει για κάποιες συμβολογίες.

Γι' αυτό το λόγο, ο όρος Συντελεστής Μεγέθυνσης αποσύρθηκε από τις γενικές προδιαγραφές GS1 και το μέγεθος του συμβόλου barcode καθορίζεται πλέον θέτοντας την ελάχιστη, προτεινόμενη και μέγιστη Διάσταση - X για κάθε συμβολογία, ανάλογα με την εφαρμογή στην οποία πρόκειται να χρησιμοποιηθεί το συγκεκριμένο σύμβολο. Επιπλέον, πρέπει να τηρείται το ελάχιστο ύψος των συμβόλων. Αξίζει να σημειωθεί, ότι το αυξημένο ύψος των συμβόλων barcode οδηγεί σε καλύτερα ποσοστά ανάγνωσης με την πρώτη απόπειρα.

1.4.2.1.3. Ανοιχτόχρωμα Περιθώρια (Ελεύθερες Ζώνες)

Όλοι οι τύποι barcode πρέπει να έχουν ανοιχτόχρωμα περιθώρια, πριν από την πρώτη και μετά την τελευταία μπάρα.

Αυτό το ανοιχτόχρωμο περιθώριο είναι εξαιρετικά σημαντικό και πρέπει να τηρείται. Το μέγεθος της επιφάνειας του ανοιχτόχρωμου περιθωρίου ποικίλλει, ανάλογα με το μέγεθος και τον τύπο του barcode. Οτιδήποτε εκτυπώνεται μέσα σε αυτά τα ανοιχτόχρωμα περιθώρια μπορεί να εμποδίσει την ανάγνωση από τους scanners.

1.4.2.1.4. Χρώματα και Αντίθεση (contrast)

Οι scanners λειτουργούν μετρώντας την ανακλαστικότητα. Επομένως θα πρέπει να υπάρχει αρκετή αντίθεση μεταξύ των σκουρόχρωμων μπάρων και των ανοιχτόχρωμων κενών. Το μελάνι των μπάρων θα πρέπει να έχει αρκετή πυκνότητα ώστε να μην δημιουργούνται κενά.

Τα σύνθετα χρώματα (CMYK) δεν είναι κατάλληλα για την εκτύπωση των barcodes, το καλύτερο είναι να χρησιμοποιούνται συμπαγή χρώματα.

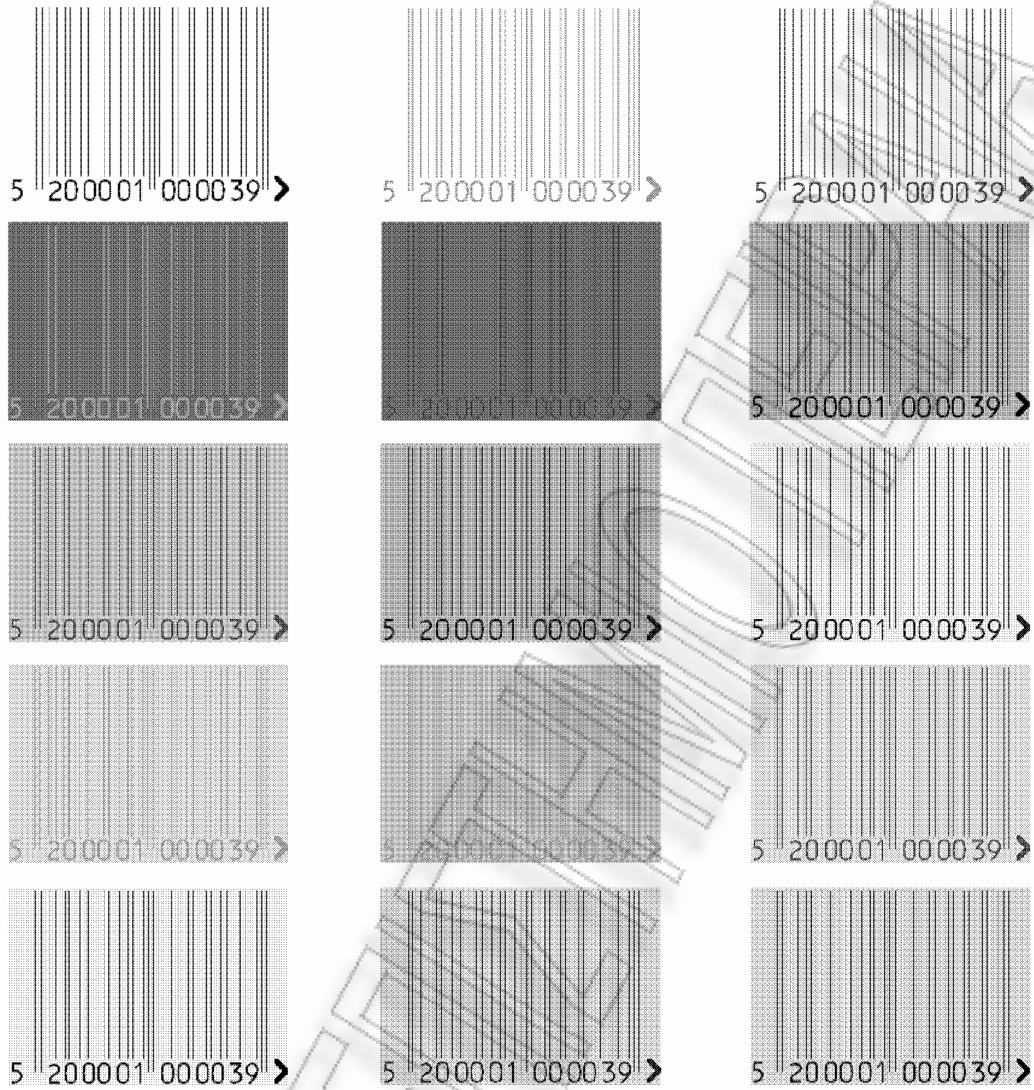
Οι scanners χρησιμοποιούν μια δέσμη κόκκινου φωτός. Μια αντίθεση η οποία ίσως φαίνεται ικανοποιητική για το ανθρώπινο μάτι, μπορεί να μην επαρκεί για τον scanner.

Τα barcodes μπορούν να εκτυπώνονται σε διάφορα χρώματα. Ένας γενικός κανόνας είναι ότι τα ανοιχτόχρωμα - θερμά χρώματα, συμπεριλαμβανομένου του λευκού, του κόκκινου, του κίτρινου και του πορτοκαλί, είναι κατάλληλα για τις ανοιχτόχρωμες γραμμές (κενά) και για τα περιθώρια. Αντιθέτως, τα σκούρα - ψυχρά χρώματα, συμπεριλαμβανομένου του μαύρου, του μπλε, του καφέ και του πράσινου, είναι κατάλληλα για τις γραμμές του συμβόλου.

Τέλος, τα γυαλιστερά υποστρώματα μπορεί να μεταβάλλουν την ανακλαστικότητα, γι' αυτό θα πρέπει να γίνονται έλεγχοι πριν από την εκτύπωση. Τα διαφανή περιτυλίγματα μπορεί επίσης να μειώνουν την αντίθεση, γι' αυτό αν χρησιμοποιείται διαφανές περιτύλιγμα, θα πρέπει να γίνονται έλεγχοι στις τελικές ολοκληρωμένες συσκευασίες.

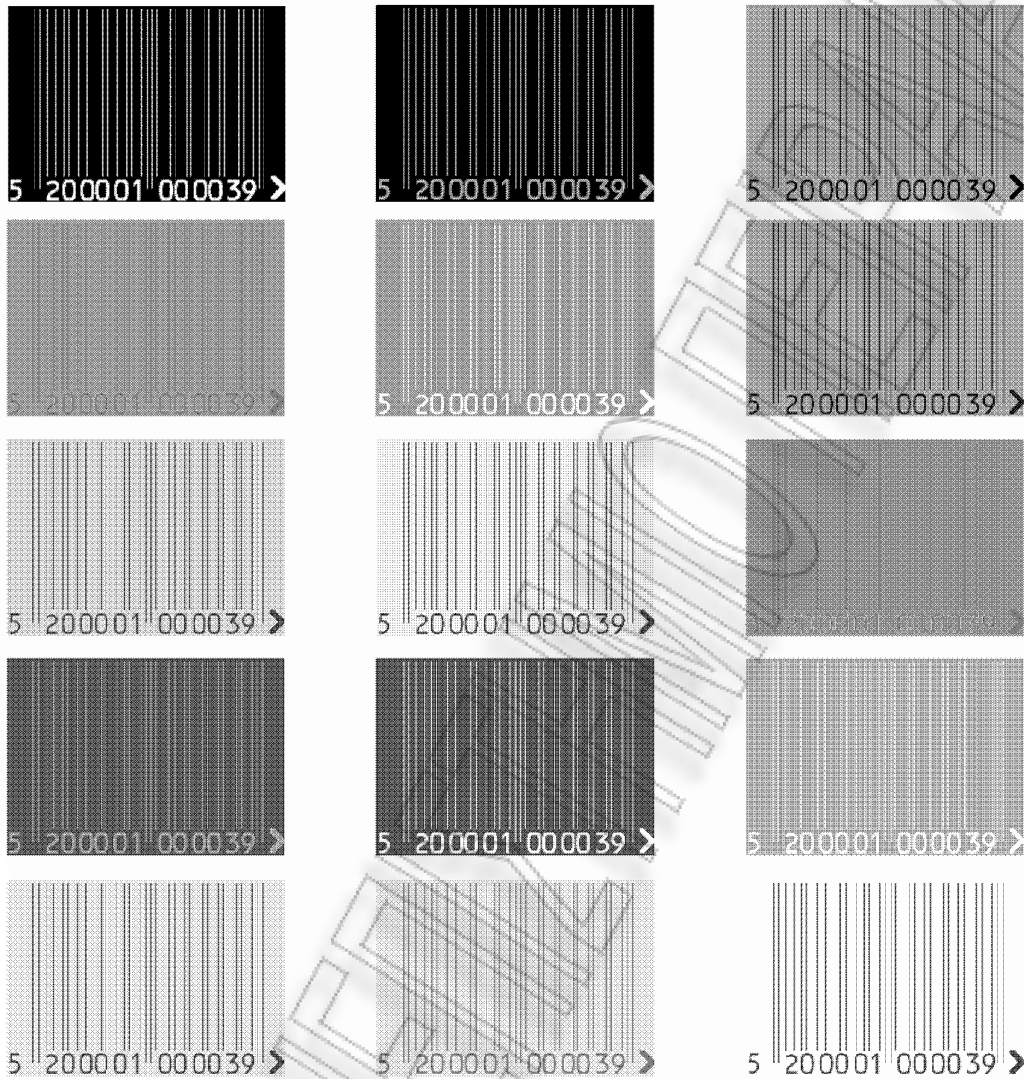
Ακολουθούν ορισμένα παραδείγματα αναγνώσιμων και μη αναγνώσιμων χρωματικών συνδυασμών για barcode.

Αναγνώσιμοι



Σχήμα 18. Αναγνώσιμα barcode

Μη αναγνώσιμοι



Σχήμα 19. Μη αναγνώσιμα barcode

1.4.2.1.5. Τρόποι Εκτύπωσης και Επικόλλησης

Οι μέθοδοι και οι τεχνολογίες που χρησιμοποιούνται σήμερα για την εκτύπωση και επικόλληση ετικετών, είτε on-line είτε off-line, είναι αρκετές, οι σημαντικότερες από τις οποίες παρουσιάζονται παρακάτω.

1.4.2.1.5.1. Θερμικοί εκτυπωτές ετικετών

Είναι οι πιο διαδεδομένοι εκτυπωτές, οι οποίοι λειτουργούν τόσο με τη μέθοδο θερμικής μεταφοράς, όσο και με την άμεση θερμική εκτύπωση. Χρησιμοποιούνται για την off-line εκτύπωση διαφόρων σταθερών ή / και μεταβλητών πληροφοριών (barcodes, περιγραφές, ημερομηνίες παραγωγής και λήξης, κωδικούς παραγωγής (Lot No), λογότυπα, κλπ) σε κάθε είδους ετικέτα, με τη βοήθεια του προσωπικού σχεδιαστικού προγράμματος του κάθε εκτυπωτή.

Στην περίπτωση που η εκτύπωση γίνεται με τέτοιου είδους θερμικό εκτυπωτή, η επικόλληση γίνεται είτε με το χέρι είτε με χρήση ετικετέζας και αφού οι εκτυπωμένες ετικέτες έχουν επανατυλιχθεί σε ρολλό, με τη βοήθεια ενός συστήματος ανατύλιξης.

1.4.2.1.5.2. Συστήματα Αυτόματης Εκτύπωσης και Επικόλλησης (PRINT & APPLY)

Αποτελούν συνδυασμό θερμικού εκτυπωτή ετικετών και ετικετέζας (Print & Apply). Τόσο το σύστημα εκτύπωσης ετικετών, όσο και το σύστημα επικόλλησης (Applicator) συνδυάζονται άψογα σε μια ολοκληρωμένη μονάδα, που ελέγχεται από ένα κεντρικό ενσωματωμένο μικροϋπολογιστή. Επιπλέον, σε μια βασική μονάδα εκτύπωσης μπορούν να προσαρμοστούν εύκολα πολλά διαφορετικά applicators που καλύπτουν κάθε ανάγκη επικόλλησης ετικετών σε οποιαδήποτε συσκευασία (χαρτοκιβώτια, παλέτες, ατομικό προϊόν, πολυσυσκευασίες, κλπ).

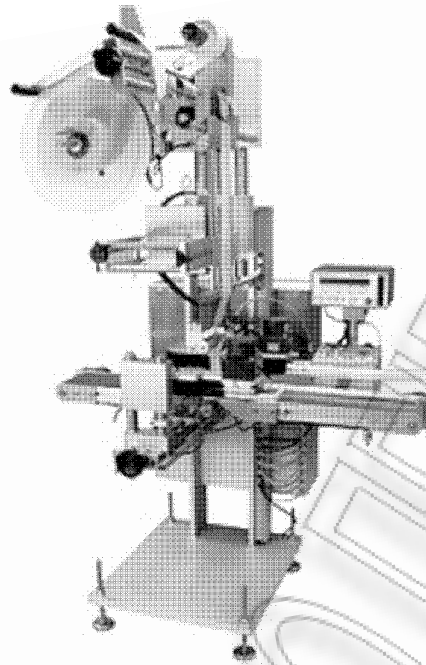
Ο εκτυπωτής εκτυπώνει on-line την ετικέτα με τις απαραίτητες πληροφορίες (σταθερές και μεταβλητές) και στην συνέχεια την τοποθετεί αυτόματα στο προϊόν. Το βασικό χαρακτηριστικό τέτοιων συστημάτων print & apply είναι η compact κατασκευή τους ενώ υποστηρίζουν ταχύτητες εκτύπωσης και επικόλλησης που φθάνουν έως και τα 120 προϊόντα / λεπτό.

Χρησιμοποιείται κυρίως σε αυτόματες γραμμές συσκευασίας για κωδικοποίηση τόσο των ίδιων των προϊόντων, όσο και των χαρτοκιβωτίων, shrink-wraps, παλετών, κλπ.

1.4.2.1.5.3. Ετικετέζα με προσαρμοσμένο θερμικό εκτυπωτή τύπου SmartDate

Αποτελούνται από δύο ξεχωριστά συστήματα, μία ετικετέζα και ένα θερμικό εκτυπωτή τύπου SmartDate, κατάλληλα συνδυασμένα ώστε να εκτυπώνουν on-line τις απαιτούμενες σταθερές και μεταβλητές πληροφορίες και να επικολλούν ετικέτες κυρίως σε προϊόντα που η μορφή και το σχήμα τους δεν είναι ομοιόμορφα, π.χ. σκαφάκια, εύκαμπτες συσκευασίες τροφίμων, κλπ.

Τα βασικά χαρακτηριστικά ενός τέτοιου υβριδικού συστήματος είναι οι πολύ υψηλές ταχύτητες επικόλλησης (έως και 220 προϊόντα/λεπτό), η ευελιξία και η εξαιρετική ακρίβεια επικόλλησης. Οι περισσότερες εφαρμογές τέτοιων συστημάτων εμφανίζονται στις βιομηχανίες συσκευασίας αγροτικών προϊόντων, στις βιομηχανίες τροφίμων και στις φαρμακοβιομηχανίες.



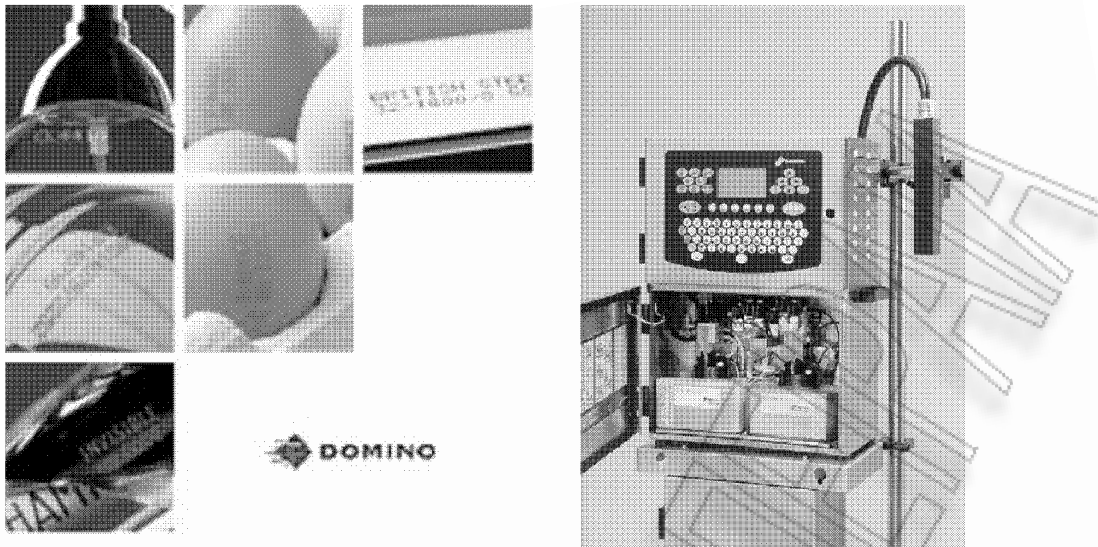
Σχήμα 22. Ετικετέζα με προσαρμοσμένο θερμικό εκτυπωτή τύπου SmartDate

1.4.2.1.5.4. Εκτυπωτές θερμικής μεταφοράς για εύκαμπτα υλικά συσκευασίας

Οι βιομηχανικοί εκτυπωτές θερμικής μεταφοράς τύπου SmartDate προσφέρουν αξιόπιστες λύσεις κωδικοποίησης για κάθε συσκευαστική μηχανή. Χρησιμοποιούν τεχνολογία εκτύπωσης θερμικής μεταφοράς και μπορούν να εκτυπώνουν on-line και απευθείας σε οποιοδήποτε υλικό συσκευασίας ημερομηνίες παραγωγής και λήξης, κωδικούς (Lot, batch, κλπ), barcodes, λογότυπα και διάφορα κείμενα, αντικαθιστώντας την ανάγκη εκτύπωσης και επικόλλησης ετικετών. Είναι, επίσης, ειδικά σχεδιασμένοι για να μπορούν να αντικαθιστούν με απλό τρόπο τους παλαιάς τεχνολογίας μηχανικούς εκτυπωτές επαφής. Τα σημαντικότερα πλεονεκτήματά τους είναι η υψηλή ταχύτητα εκτύπωσης που μπορεί να φθάσει και τα 1.800mm/sec, όπως και το χαμηλό κόστος λειτουργίας, δεδομένου ότι το μοναδικό αναλώσιμο είναι η μελανοταινία. Επιπλέον, οι βιομηχανικοί εκτυπωτές SmartDate μπορούν να προσαρμοστούν εύκολα και σε οποιαδήποτε ετικετέζα. Τέτοια συστήματα βρίσκουν εφαρμογή σε βιομηχανίες συσκευασίας ξηρών καρπών, καφέ, οσπρίων, σταφίδας, σνακ, ζαχαρωδών, ζύμης, ζυμαρικών, πλαστικών, κλπ, καθώς και φαρμακοβιομηχανίες και βιομηχανίες καλλυντικών.

1.4.2.1.5.5. Εκτυπωτές ink-jet

Οι βιομηχανικοί εκτυπωτές εκτόξευσης μελάνης ή κοινώς ink-jets χρησιμοποιούνται σε γραμμές παραγωγής για την on-line εκτύπωση κυρίως μεταβλητών πληροφοριών, όπως ημερομηνίες παραγωγής και λήξεως, κωδικούς παραγωγής, κλπ. Η εκτύπωση αυτή μπορεί να γίνει είτε απευθείας στο προϊόν είτε επάνω στην ετικέτα του προϊόντος που έχει σημάνει με τα σταθερά στοιχεία το προϊόν. Τα σημαντικότερα πλεονεκτήματα αυτών των συστημάτων είναι η μεγάλη ευελιξία και ταχύτητα εκτύπωσης, ιδιαίτερα σε γρήγορες γραμμές παραγωγής, σε σχέση με τη περισσότερο χρονοβόρα διαδικασία της εκτύπωσης και επικόλλησης μιας ετικέτας, καθώς και το πολύ χαμηλότερο κόστος ανά εκτύπωση. Τέτοια συστήματα βρίσκουν εφαρμογή σε όλες τις βιομηχανίες τροφίμων, ποτών, φαρμάκων, καλλυντικών, πλαστικών, μετάλλου, κλπ.



Σχήμα 24. Εκτυπωτής εκτόξευσης μελάνης (ink-jet) και δείγματα εφαρμογών

1.4.2.1.5.6. Εκτυπωτές ink-jet υψηλής ανάλυσης

Οι βιομηχανικοί εκτυπωτές εκτόξευσης μελάνης υψηλής ανάλυσης χρησιμοποιούνται σε γραμμές παραγωγής για την on-line εκτύπωση είτε σταθερών είτε μεταβλητών πληροφοριών, όπως ημερομηνίες παραγωγής και λήξεως, κωδικούς παραγωγής, περιγραφές, λογότυπα, barcode, κλπ. Η ικανότητα αυτών των συστημάτων να εκτυπώνουν εξαιρετικής ποιότητας πληροφορίες οφείλεται στη χρήση θερμοπλαστικής μελάνης (Touch Dry), η οποία σε κανονικές συνθήκες είναι σε στερεά κατάσταση. Τη στιγμή της εκτύπωσης, ο εκτυπωτής ανεβάζει τη θερμοκρασία της μελάνης, η οποία μεταπίπτει σε υγρή κατάσταση και στη συνέχεια γίνεται η εκτόξευση από την εκτυπωτική κεφαλή. Όταν η μελάνη φθάσει στην επιφάνεια του χαρτοκιβωτίου στερεοποιείται, προσφέροντας υψηλής ποιότητας εκτυπούμενο μήνυμα. Τέτοια συστήματα βρίσκουν κυρίως εφαρμογή σε χαρτοκιβώτια, συσκευασίες διογκωμένης πολυστερίνης, κλπ, εξαλείφοντας το κόστος των προτυπωμένων υλικών.



Σχήμα 25. Εκτυπωτής εκτόξευσης μελάνης (ink-jet) υψηλής ανάλυσης και δείγματα εφαρμογών

1.4.2.1.6. Ποιότητα Εκτύπωσης

Οι συνθήκες εκτύπωσης θα πρέπει να ελέγχονται τακτικά σε όλη την περίοδο εκτύπωσης ώστε να εξασφαλίζεται ότι δεν έχουν χειροτερέψει μετά από την αρχική αξιολόγηση.

Υπάρχουν διάφοροι τρόποι για την αξιολόγηση της ποιότητας ενός barcode. Μπορούν να χρησιμοποιηθούν πιο σύνθετες ή πιο απλές οπτικές μέθοδοι προκειμένου να πετύχουμε την αξιολόγηση της ποιότητας ενός συμβόλου. Ένα απλό παράδειγμα είναι η εκτύπωση ενός Η με δεδομένες διαστάσεις στο εσωτερικό της περιβάλλουσας γραμμής ενός συμβόλου ITF - 14.

Για τον προσδιορισμό του προσανατολισμού εκτύπωσης του συμβόλου, θα πρέπει να λαμβάνεται πάντα υπ' όψιν η μέθοδος εκτύπωσης που θα χρησιμοποιηθεί. Για παράδειγμα, αν χρησιμοποιείται φλεξογραφία, θα πρέπει το barcode να εκτυπώνεται παράλληλα προς τη διεύθυνση εκτύπωσης λόγω του «απλώματος» της μελάνης που προκύπτει από αυτή την μέθοδο. Στην περίπτωση λιθογραφίας ή offset, το άπλωμα είναι συνήθως αμελητέο. Σε κάθε περίπτωση, θα πρέπει να συμβουλευόμαστε τον υπεύθυνο εκτύπωσης.

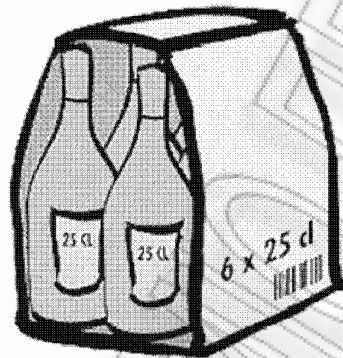
1.4.2.1.7. Θέσεις barcode

Η παραγωγικότητα και η ακρίβεια κατά την ανάγνωση (scanning) του συμβόλου αυξάνονται σημαντικά όταν η θέση του barcode είναι προβλέψιμη. Με την τοποθέτηση του σε κάποια αναμενόμενη θέση, επιτυγχάνεται η μέγιστη παραγωγικότητα σε οποιοδήποτε περιβάλλον.

1.4.2.1.7.1. Θέσεις του barcode στα είδη λιανικής πώλησης

Το barcode, συμπεριλαμβανομένων των ψηφίων που είναι αναγνώσιμα από τον άνθρωπο και απεικονίζονται στο κάτω μέρος, θα πρέπει να είναι ευδιάκριτο και ευανάγνωστο.

Δεν πρέπει ποτέ να αφήνουμε δυο barcode με διαφορετικούς κωδικούς αναγνώρισης να είναι ορατά πάνω στην ίδια συσκευασία. Αυτό ισχύει ιδιαίτερα για τις πολυσυσκευασίες, ειδικά εκείνες που έχουν διαφανές περιτύλιγμα. Οι συσκευασίες αυτές θα πρέπει να φέρουν ένα ξεχωριστό μοναδικό κωδικό και όλα τα υπόλοιπα barcode (των περιεχόμενων ειδών) θα πρέπει να είναι καλυμμένα.



Σχήμα 26. Θέση barcode στα είδη λιανικής

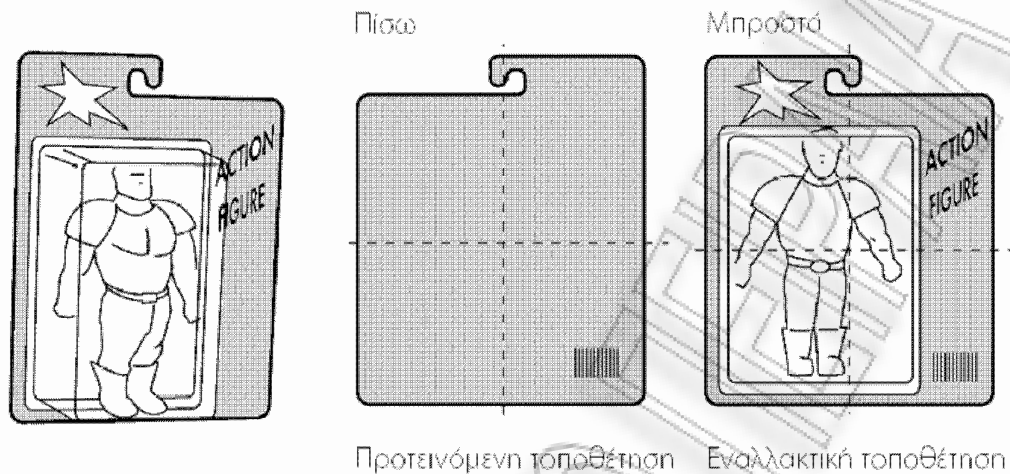
Αν το είδος έχει τυχαία περιτύλιξη (αυτό συμβαίνει συνήθως σε προϊόντα των οποίων η συσκευασία προέρχεται από συνεχές ρολό film), είναι αποδεκτό να υπάρχουν δυο ή περισσότερα ίδια barcode τυπωμένα στο υλικό του περιτυλίγματος. Αυτό εξασφαλίζει ότι ένα τουλάχιστον πλήρες barcode είναι πάντα ορατό.



Σχήμα 27. Θέση barcode στα είδη λιανικής 2

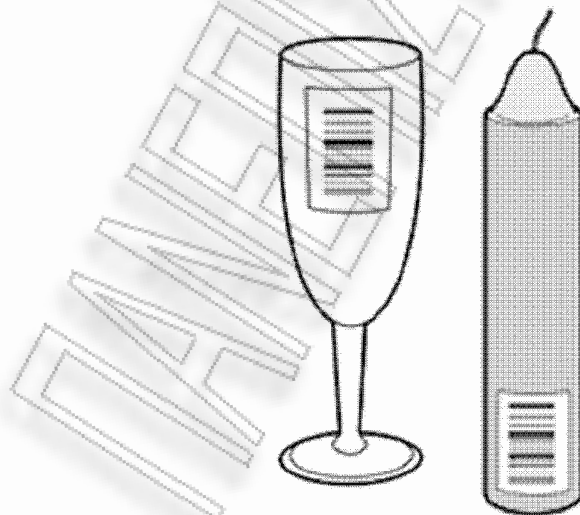
Η ανάγνωση έχει μεγάλη επιτυχία όταν το barcode είναι τυπωμένο πάνω σε μια αρκετά λεία επιφάνεια. Συνιστάται να αποφεύγονται οι εκτυπώσεις γύρω από γωνίες, σε σημεία που διπλώνουν ή πάνω σε οποιαδήποτε άλλη ανώμαλη επιφάνεια της συσκευασίας.

Μερικές φορές, το ακανόνιστο σχήμα μιας συσκευασίας εμποδίζει την άμεση επαφή του scanner με το σύμβολο. Αυτό ισχύει ειδικά για τα είδη με συσκευασία σε κάρτα, συσκευασία τύπου blister ή συσκευασία με καμπυλότητες / κοιλότητες.



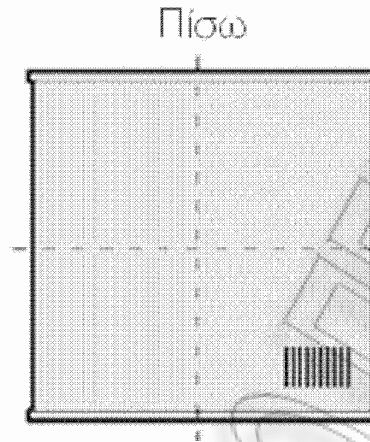
Σχήμα 28. Θέση barcode στα είδη λιανικής 3

Στα προϊόντα κυλινδρικού σχήματος, όταν το επιτρέπει η διεύθυνση εκτύπωσης, είναι γενικά προτιμότερο οι γραμμές του barcode να έχουν οριζόντια διεύθυνση (τύπου κλίμακας) όταν το είδος στηρίζεται στη βάση του, έτσι ώστε η ακτίνα του scanner να σαρώνει το σύμβολο σε όσο το δυνατόν πιο επίπεδη επιφάνεια του. Αυτό αντισταθμίζει τα προβλήματα που σχετίζονται με τις καμπύλες τέτοιων ειδών, όπως οι κονσέρβες και οι φιάλες. Ο προσανατολισμός «τύπου κλίμακας» είναι υποχρεωτικός για τις καμπυλωτές επιφάνειες με μικρή ακτίνα.



Σχήμα 29. Θέση barcode στα είδη λιανικής 4

Γενικά, η προτεινόμενη θέση για το σύμβολο barcode είναι στο κάτω δεξιό τεταρτημόριο του πίσω μέρους, τηρώντας τις απαιτούμενες Ελεύθερες Ζώνες γύρω από το barcode και τον «κανόνα τον ακμών[‡]». Η εναλλακτική θέση είναι στο κάτω τεταρτημόριο κάποιας άλλης πλευράς της συσκευασίας.



Σχήμα 30. Θέση barcode στα είδη λιανικής 5

1.4.2.1.7.2. Θέσεις του barcode σε είδη που δεν προορίζονται για λιανική πώληση

Για κάθε Μονάδα Εμπορίας και κάθε Μονάδα Logistics η ελάχιστη απαίτηση είναι να φέρουν τουλάχιστον ένα barcode.

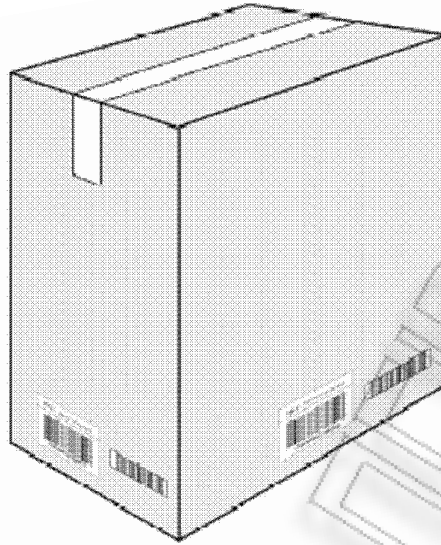
Εν τούτοις, η βέλτιστη προτεινόμενη πρακτική είναι να επικολλώνται δυο ετικέτες σε διαδοχικές πλευρές των Μονάδων Logistics. Αυτές οι πλευρές θα πρέπει να είναι η στενή και η φαρδιά (δεξιά της στενής). Έτσι εξασφαλίζεται ότι με την κατάλληλη στροφή του αντικειμένου θα είναι πάντα ορατό το ένα σύμβολο barcode.

1.4.2.1.7.3. Χαρτοκιβώτια

Το κάτω άκρο των γραμμών του barcode θα πρέπει να απέχει 32mm από την ακμή της βάσης του κιβωτίου. Συμπεριλαμβανόμενων των Ελεύθερων Ζωνών, το barcode θα πρέπει να απέχει τουλάχιστον 19mm από τις κατακόρυφες ακμές των πλευρών του κιβωτίου.

Όταν χρησιμοποιείται κάποιο σύμβολο barcode ITF - 14, τα εξωτερικά άκρα της αριστερής ή της δεξιάς περιβάλλουσας γραμμής θα πρέπει να απέχουν τουλάχιστον 19mm από τις κατακόρυφες ακμές των πλευρών του κιβωτίου.

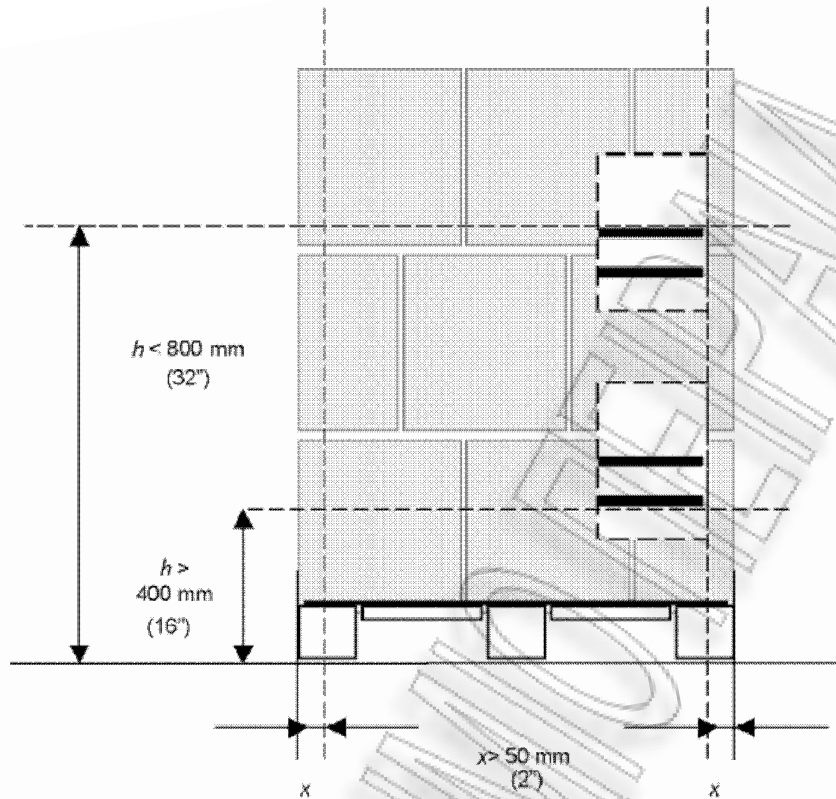
[‡] Το barcode δεν πρέπει να τοποθετείται σε απόσταση μικρότερη των 8mm ή μεγαλύτερη των 102mm από οποιαδήποτε ακμή / άκρο της συσκευασίας.



Σχήμα 31. Barcode σε χαρτοκιβώπια

1.4.2.1.7.4. Παλέτες

Στις παλέτες οι ετικέτες θα πρέπει να τοποθετούνται έτσι ώστε όλα τα σύμβολα να βρίσκονται σε ύψος 400mm έως 800mm από την επιφάνεια στήριξης της παλέτας, περιλαμβανομένης της βάσης, ενώ δεν θα πρέπει να απέχουν λιγότερο από 50mm από την κατακόρυφη ακμή.

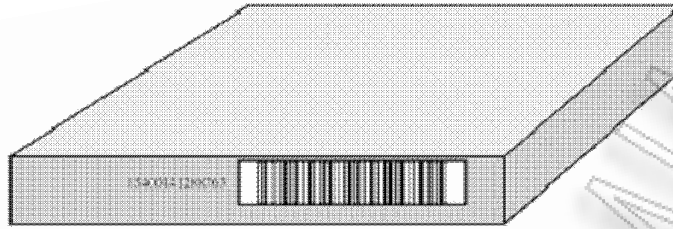


Σχήμα 32. Barcode σε παλέτες

1.4.2.1.7.5. Μικρά κιβώτια

Αν το ύψος ενός κιβωτίου ή ενός δίσκου είναι μικρότερο από 50mm και είναι αδύνατη η εκτύπωση ενός barcode πλήρους ύψους μαζί με τους χαρακτήρες OCR που βρίσκονται κάτω από τις μπάρες, ή αν η φύση της Μονάδας είναι τέτοια που να απαγορεύει την τοποθέτηση ενός συμβόλου πλήρους ύψους, τότε θα πρέπει να ληφθούν υπόψη οι παρακάτω επιλογές κατά σειρά προτεραιότητας:

1. Τοποθετούμε τους χαρακτήρες OCR δίπλα από το barcode, προσέχοντας να μην παραβιάσουμε τις Ελεύθερες Ζώνες.
2. Όταν το ύψος της Μονάδας είναι μικρότερο από 32mm, τότε το barcode μπορεί να εκτυπωθεί στην επάνω επιφάνεια της συσκευασίας. Το σύμβολο πρέπει να τοποθετηθεί έτσι ώστε οι μπάρες να είναι κάθετες στην στενή πλευρά και να βρίσκεται σε απόσταση μεγαλύτερη των 19mm από οποιαδήποτε ακμή.



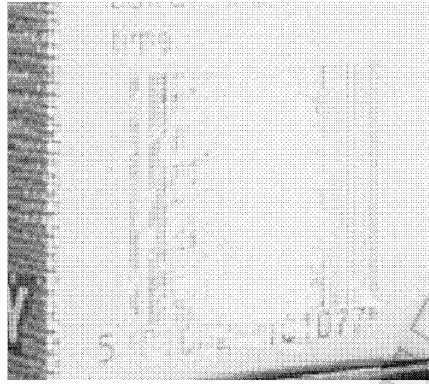
Σχήμα 33. Barcode σε μικρά κιβώτια

Στο σημείο αυτό, πριν αναφέρουμε τα barcode που χρησιμοποιούνται στο Σύστημα GS1, είναι αναγκαίο να παρουσιάσουμε τα στοιχεία που επηρεάζουν την αναγνωσιμότητα των barcode καθώς και κάποιες φωτογραφίες μη αναγνώσιμων barcode.

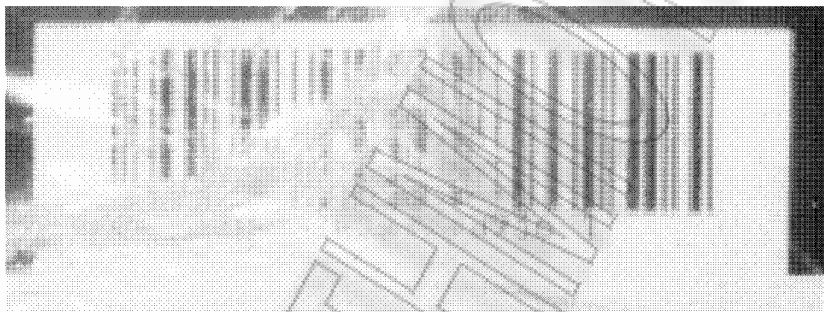
Ορισμένα από τα στοιχεία τα οποία επηρεάζουν την σωστή ανάγνωση των barcode είναι τα ακόλουθα:

- ✓ Ποιότητα εκτύπωσης
- ✓ Θέση συμβόλου
- ✓ Μέγεθος συμβόλου
- ✓ Ελεύθερες ζώνες
- ✓ Επικαλύψεις
- ✓ Συνθήκες περιβάλλοντος
- ✓ Είδος scanner (σταθερός - φορητός)

Ακολουθούν φωτογραφίες μη αναγνώσιμων barcode.



Σχήμα 34. Μη αναγνώσιμο barcode 1



Σχήμα 35. Μη αναγνώσιμο barcode 2

Σχήμα 36. Μη αναγνώσιμο barcode 3

1.4.2.1.8. Barcode που χρησιμοποιούνται στο Σύστημα GS1

Το σύστημα GS1 υποστηρίζει τις εξής τρεις διαφορετικές συμβολογίες:

1. EAN/UPC

2. ITF - 14

3. GS1 - 128

1.4.2.1.8.1. Σύμβολα EAN/UPC

Οι Μονάδες Εμπορίας που πωλούνται σε σημεία λιανικής πώλησης θα πρέπει να φέρουν ένα από τα εξής σύμβολα barcode EAN/UPC: EAN - 13, EAN - 8, UPC - A ή UPC - E. Τα σύμβολα αυτά μπορούν επίσης να χρησιμοποιούνται σε είδη εμπορίας που δεν πωλούνται σε σημεία λιανικής πώλησης.

Αν οι συνθήκες εκτύπωσης και / ή η ποιότητα του υποστρώματος δεν είναι ικανοποιητικές για την απ' ευθείας εκτύπωση στη συσκευασία, τότε θα πρέπει να χρησιμοποιούνται ετικέτες barcode.

Παρακάτω απεικονίζονται τα barcode EAN/UPC στις ονομαστικές τους διαστάσεις (συντελεστής μεγέθυνσης 100%), συμπεριλαμβανομένων των Ελεύθερων Ζωνών. Για κάθε τύπο barcode δίδεται το ελάχιστο και το μέγιστο μέγεθος.

Σύμβολο EAN - 8



Σχήμα 37. Σύμβολο EAN - 8

Ελάχιστο μέγεθος: 21,38mm x 17mm

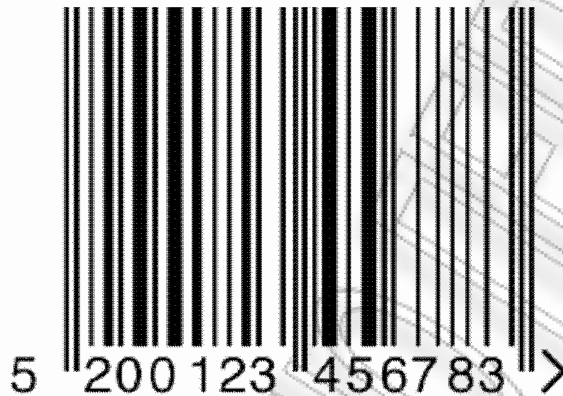
Μέγιστο μέγεθος: 53,46mm x 43mm

Ονομαστικό μέγεθος: 26,73mm x 21mm

Διάσταση - Χ[§]: 0,330mm

Σημείωση: το ύψος του barcode έχει στρογγυλοποιηθεί.

Σύμβολο EAN - 13



Σχήμα 38. Σύμβολο EAN – 13

Ελάχιστο μέγεθος: 29,83mm x 21mm

Μέγιστο μέγεθος: 74,58mm x 52mm

Ονομαστικό μέγεθος: 37,29mm x 26mm

Διάσταση - Χ: 0,330mm

Σημείωση: το ύψος του barcode έχει στρογγυλοποιηθεί.

Κάποια από τα βασικά χαρακτηριστικά των συμβολογίων EAN - 8 και EAN - 13 είναι τα ακόλουθα:

1. Αριθμητικές συμβολογίες
2. Σταθερό μήκος
3. Μονοδιάστατες και γραμμικές συμβολογίες

§ Η Διάσταση - Χ είναι το προκαθορισμένο πλάτος της στοιχειώδους μπάρας ενός barcode. Αυτό το μέγεθος διαφέρει για κάθε είδος barcode.

4. Διαβάζονται κάτω από οποιαδήποτε διεύθυνση (omnidirectionally)

Σύμβολο UPC - A



Σχήμα 39. Σύμβολο UPC - A

Ελάχιστο μέγεθος:	29,83mm x 21mm
Μέγιστο μέγεθος:	74,58mm x 52mm
Ονομαστικό μέγεθος:	37,29mm x 26mm
Διάσταση - Χ:	0,330mm

Σημείωση: το ύψος του barcode έχει στρογγυλοποιηθεί.

Σύμβολο UPC - E



Σχήμα 40. Σύμβολο UPC - E

Ελάχιστο μέγεθος:	17,69mm x 21mm
Μέγιστο μέγεθος:	44,22mm x 52mm
Ονομαστικό μέγεθος:	21,11mm x 26mm
Διάσταση - Χ:	0,330mm

Σημείωση: το ύψος του barcode έχει στρογγυλοποιηθεί.

Τα σύμβολα EAN/UPC μπορούν να εκτυπωθούν με Συντελεστή Μεγέθυνσης από 80% έως 200% του ονομαστικού μεγέθους. Για να εξασφαλίζεται η αποτελεσματική ανάγνωσή τους σε οποιοδήποτε περιβάλλον, συμπεριλαμβανομένων των scanner που είναι προσαρμοσμένοι σε ιμάντα μεταφοράς, θα πρέπει να χρησιμοποιείται Συντελεστής Μεγέθυνσης με τιμή τουλάχιστον 150%.

Τα σύμβολα αυτά έχουν σχεδιαστεί έτσι ώστε να μπορούν να διαβαστούν κάτω από οποιαδήποτε κατεύθυνση.

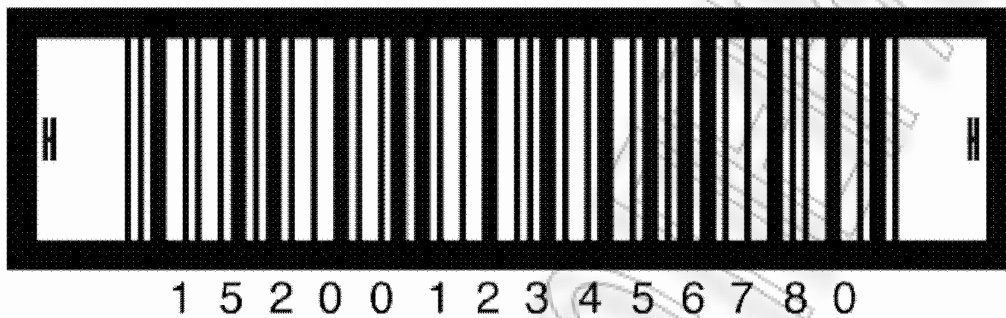
Η μείωση του ύψους του συμβόλου, αφαιρεί τη δυνατότητα ανάγνωσης από οποιαδήποτε κατεύθυνση. Η μείωση αυτή θα πρέπει να αποτελεί την έσχατη λύση, όταν ο υπάρχον χώρος επαρκεί μόνο για κάποιο σύμβολο μειωμένου ύψους.

Συνίσταται η χρήση ενός δείκτη για τον καθορισμό των ανοιχτόχρωμων περιθωρίων (Ελεύθερες Ζώνες), κατά προτίμηση του χαρακτήρα > ή του < σε κατάλληλη θέση ώστε η αιχμή να συμπίπτει με το όριο του περιθωρίου.

1.4.2.1.8.2. Σύμβολο ITF – 14

Για τις εταιρίες που επιθυμούν να εκτυπώνουν το barcode άμεσα πάνω στο χαρτόνι των καρτοκιβωτίων και ειδικά πάνω σε χαρτόνι συσκευασίας, καταλληλότερο είναι το σύμβολο ITF - 14. Οι ανοχές κατά την εκτύπωση του είναι λιγότερο απαιτητικές.

Για το σύμβολο αυτό, μπορεί να είναι εφικτή η προεκτύπωση ή η άμεση εκτύπωση, με εκτυπωτή θερμικής μεταφοράς ή ink - jet ανάλογα με την περίπτωση.



Σχήμα 41. 1.4.2.1.8.2. Σύμβολο ITF – 14

Ελάχιστο μέγεθος (50%): 71,40mm x 12,70mm

Μέγιστο μέγεθος (100%): 142,75mm x 32mm

Ονομαστικό μέγεθος: 142,75mm x 32mm

Διάσταση - X: 1,016mm

Οι παραπάνω διαστάσεις δεν περιλαμβάνουν την περιβάλλουσα γραμμή.

Για να εξασφαλίζεται η αποτελεσματική ανάγνωση σε οποιοδήποτε περιβάλλον συμπεριλαμβανομένων και των scanners που είναι προσαρμοσμένοι σε ιμάντα μεταφοράς, θα πρέπει να χρησιμοποιείται συντελεστής μεγέθυνσης με τιμή τουλάχιστον 50%.

Κάποια από τα βασικά χαρακτηριστικά της συμβολογίας ITF - 14 είναι τα ακόλουθα:

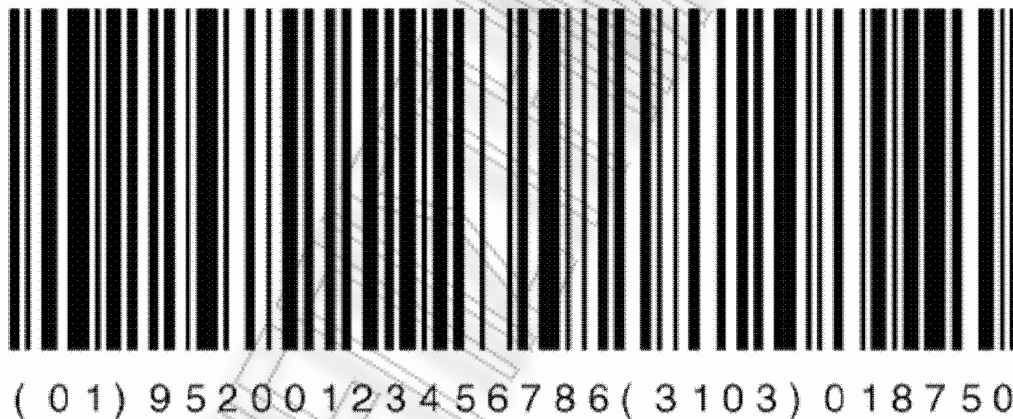
1. Αριθμητική συμβολογία
2. Σταθερό μήκος
3. Μονοδιάστατη και γραμμική συμβολογία
4. Απαιτεί ζυγό αριθμό ψηφίων
5. Υπάρχουν μόνο δυο πλάτη μπαρών (φαρδιές και στενές)

6. Διαβάζονται αμφίδρομα (bidirectionally)
7. Το σύμβολο περιβάλλεται από ένα πλαίσιο (bearer bar)
8. Η ονομαστική διάσταση του συμβόλου είναι το μέγεθος 100%.
9. Τα επιτρεπόμενα μεγέθη κυμαίνονται από 25% έως 100% του ονομαστικού

1.4.2.1.8.3. Σύμβολο GS1 – 128

Η συμβολογία GS1 - 128 είναι μια παραλλαγή της συμβολογίας Code 128. Η άδεια για τη χρήση της έχει δοθεί αποκλειστικά στον GS1 και δεν περιορίζεται για ανάγνωση σε προϊόντα που περνούν από σημεία λιανικής πώλησης. Είναι η μόνη συμβολογία που υποστηρίζεται από τον οργανισμό GS1 και επιτρέπει την αναγραφή επιπλέον πληροφοριών, πέραν του κωδικού αναγνώρισης.

Το σύμβολο GS1 - 128 έχει μεταβλητό μήκος, ανάλογα με το πλήθος των απεικονιζόμενων χαρακτήρων, τον τύπο των κωδικοποιούμενων χαρακτήρων και την Διάσταση - X που μπορεί να επιτευχθεί.



Σχήμα 42. 1.4.2.1.8.3. Σύμβολο GS1 – 128

Για ένα δεδομένο μήκος πληροφοριών, το μέγεθος του συμβόλου μεταβάλλεται μεταξύ ακραίων τιμών, ώστε να μπορεί να ανταποκριθεί στις κλίμακες της ποιότητας που επιτυγχάνεται μέσω των διαφόρων μεθόδων εκτύπωσης. Το σύμβολο έχει σχεδιαστεί έτσι ώστε να μπορεί να διαβαστεί από δυο κατευθύνσεις μέσω σταθερών ή φορητών scanners. Για το λόγο αυτό, δεν μπορούν να προσδιοριστούν ελάχιστα και μέγιστα μεγέθη.

Βάση των ονομαστικών διαστάσεων (Διάσταση - X: 1mm) τα σύμβολα GS1 - 128 μπορούν να εκτυπώνονται με Συντελεστή Μεγέθυνσης από 25% έως 100%.

Για να εξασφαλίζεται η αποτελεσματική ανάγνωση σε οποιοδήποτε περιβάλλον συμπεριλαμβανομένων των scanners που είναι προσαρμοσμένα σε μάντα μεταφοράς, θα πρέπει να χρησιμοποιείται Συντελεστής Μεγέθυνσης με τιμή τουλάχιστον 50%.

Κάποια από τα βασικά χαρακτηριστικά της συμβολογίας GS1 - 128 είναι τα ακόλουθα:

1. Αλφαριθμητική συμβολογία
2. Μεταβλητού μήκους (48 αλφαριθμητικοί ή 96 αριθμητικοί χαρακτήρες)
3. Περιέχει έναν ειδικό χαρακτήρα, FNC1
4. Αποτελεί υποσύνολο της συμβολογίας CODE 128
5. Μονοδιάστατη και γραμμική συμβολογία
6. Υψηλή πυκνότητα
7. Διαβάζεται αμφίδρομα (bidirectionally)
8. Η ονομαστική διάσταση του συμβόλου είναι το μέγεθος 100%
9. Τα επιτρεπόμενα μεγέθη κυμαίνονται από 25% έως 100% του ονομαστικού
10. Συναρμογή δεδομένων

1.4.2.1.8.4. Reduced Space Symbology (RSS)

Πρόκειται για μια νέα συμβολογία η οποία έχει πρόσφατα ενταχθεί στο σύστημα GS1. Η συμβολογία RSS είναι μονοδιάστατη, υψηλής πυκνότητας και αναγνώσιμη υπό οποιαδήποτε διεύθυνση.



Σχήμα 43. Reduced Space Symbology (RSS)

1.4.2.1.8.5. RSS & COMPOSITE

Μια συμβολογία που συνδυάζει ένα γραμμικό σύμβολο με ένα άλλο δισδιάστατο. Το δισδιάστατο σύμβολο δεν μπορεί να υπάρξει μόνο του. Η συγκεκριμένη συμβολογία διαβάζεται κάτω από οποιαδήποτε διεύθυνση.



Σχήμα 44. 1.4.2.1.8.5. RSS & COMPOSITE

1.4.2.1.9. Στοιχεία που πρέπει να λαμβάνονται υπ' όψιν για τη χρήση των συμβολογιών

Η συμβολογία GS1 - 128 είναι η μόνη συμβολογία που δίνει τη δυνατότητα απεικόνισης επιπλέον πληροφοριών, πέραν του κωδικού αναγνώρισης GTIN.

Αν, για οποιοδήποτε λόγο, υπάρχει ανάγκη εκτύπωσης επιπλέον πληροφοριών πέραν των στοιχείων αναγνώρισης (σειριακός αριθμός, κλπ.) και η Μονάδα Εμπορίας φέρει ήδη κάποιο σύμβολο EAN - 8, ITF - 14 ή GS1 - 128 με Δείκτη Εφαρμογής (01) τότε μπορεί να χρησιμοποιηθεί μια από τις ακόλουθες λύσεις:

- ✓ Να επικολληθεί μια ετικέτα με τις επιπλέον πληροφορίες σε μορφή συμβόλου GS1 - 128, πέραν του υπάρχοντος συμβόλου ITF - 14. Σε αυτή την περίπτωση, όλα τα σύμβολα θα πρέπει να είναι οριζόντια ευθυγραμμισμένα.
- ✓ Να επικολληθεί μια ετικέτα η οποία θα καλύπτει το υπάρχον σύμβολο. Ο κωδικός αναγνώρισης ο οποίος απεικονίζεται μέσω του αρχικού συμβόλου θα πρέπει να εκτυπώνεται πάνω στην ετικέτα, μαζί με τα υπόλοιπα επιλεγμένα δεδομένα χαρακτηριστικών, κατά προτίμηση σε μορφή συμβόλου GS1 - 128 με συναρμογή των πληροφοριών.

1.4.2.1.10. Επιλογή μεταξύ των barcode

Η κωδικοποίηση των ειδών και η φυσική εφαρμογή / επικόλληση των barcode είναι δυο ξεχωριστές διαδικασίες. Είναι πολύ πιθανό, σε κάποιες εταιρίες, η διεξαγωγή τους να γίνεται σε ξεχωριστές τοποθεσίες. Η «πηγή» - ο κάτοχος της εμπορικής επωνυμίας - αποδίδει τον κωδικό στο είδος και ο κατασκευαστής τον επικολλά στην συσκευασία.

Επίσης, είναι δυνατόν να κωδικοποιείται κάποιο είδος χωρίς να αποτυπώνεται πάνω του κάποιο barcode. Αυτό μπορεί να συμβαίνει όταν είναι κυριολεκτικά αδύνατον να επικολληθεί ή να αποτυπωθεί ένα barcode, για παράδειγμα πάνω σε κάποιο πολύ μικρό καλλυντικό ή σε ένα ηλεκτρονικό εξάρτημα, σε ένα φορτίου άμμου, κλπ. Στη συνέχεια, ο κωδικός μπορεί να χρησιμοποιηθεί σε ηλεκτρονικά μηνύματα, π.χ. στην Ηλεκτρονική Ανταλλαγή Δεδομένων (EDI).

Οι χρήστες θα πρέπει να λαμβάνουν υπ' όψιν τα παρακάτω όταν επιλέγουν μεταξύ των διαφόρων συμβολογιών:

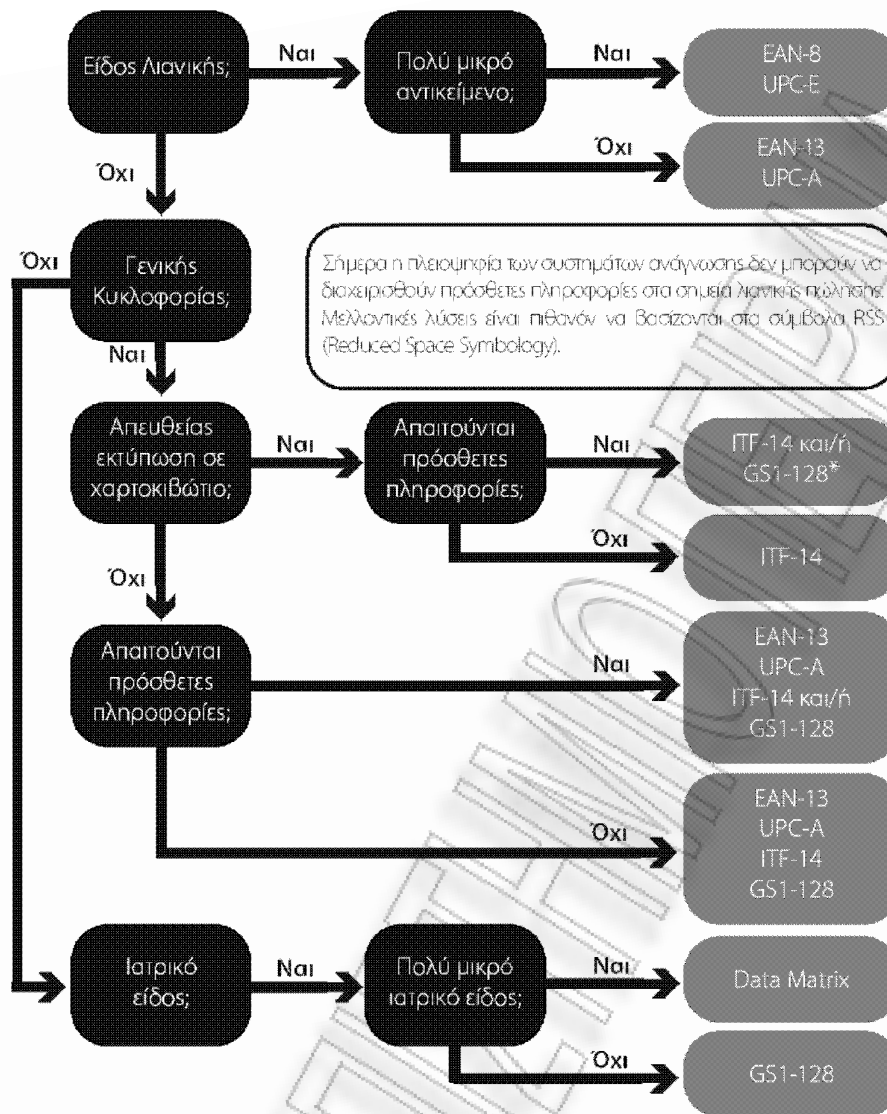
1. Τον διαθέσιμο χώρο πάνω στο αντικείμενο που θα εκτυπωθεί το barcode.
2. Τον τύπο των πληροφοριών που θα κωδικοποιηθούν μέσω του συμβόλου barcode (κωδικός αναγνώρισης μόνο ή κωδικός αναγνώρισης και επιπλέον πληροφορίες).
3. Το λειτουργικό περιβάλλον στο οποίο θα γίνει η αναγνώριση του συμβόλου (σημείο λιανικής πώλησης, αποθήκη, κλπ.).

Οι διάφοροι κωδικοί GS1 μπορούν να απεικονισθούν με συγκεκριμένα σύμβολα barcode:

Κωδικός	Μπορεί να απεικονισθεί με barcode
GTIN - 8	EAN - 8, ITF - 14, GS1 - 128* *Αφού προστεθεί ο Δείκτης και συμπληρωθεί ο κωδικός από αριστερά με 5 μηδενικά.
GTIN - 12	UPC - E*, UPC - A, EAN - 13, ITF - 14 *Μόνο για συγκεκριμένους κωδικούς
GTIN - 13	EAN - 13, ITF - 14, GS1 - 128
GTIN - 14	ITF - 14, GS1 - 128

Συγκεκριμένα σύμβολα barcode χρησιμοποιούνται για τη σήμανση συγκεκριμένων τύπων αντικειμένων:

Μικρά αντικείμενα λιανικής πώλησης	Αντικείμενα λιανικής πώλησης	Αντικείμενα χονδρικής πώλησης
EAN - 8	EAN - 13	ITF - 14
UPC - E	UPC - A	GS1 - 128



Σχήμα 45. 1.4.2.1.10. Επιλογή μεταξύ των barcode

1.4.2.2. Radio Frequency Tags (RFID)

Η τεχνολογία Ραδιοσυχνικής Αναγνώρισης (RFID, *Radio Frequency Identification*), στην οποία θα αναφερόμαστε από εδώ και στο εξής με το όνομα RFID, είναι η τεχνολογία που χρησιμοποιεί τα ραδιοκύματα (*radio waves*) με σκοπό αυτόματα να αναγνωρίζει (*identify*), να εντοπίζει (*track*), να συλλέγει και να αποθηκεύει πληροφορίες (*data capture*) έμψυχων και άψυχων αντικειμένων. Οι συχνότητες των ραδιοκυμάτων και τα αντικείμενα στα οποία εφαρμόζεται, η τεχνολογία RFID, ποικίλουν ανάλογα με την εφαρμογή και τους σκοπούς της.

Η τεχνολογία RFID είναι μέλος της οικογένειας τεχνολογιών Αυτόματης Αναγνώρισης και Συλλογής Δεδομένων (*AIDC, Automatic Identification and Data Capture*) και αποτελεί την τεχνολογική εξέλιξη των γραμμωτών κωδικών (*barcodes*). Οι δυνατότητες που δίνει η τεχνολογία RFID είναι πολύ μεγάλες και αυτό θα προσπαθήσουμε να αναδείξουμε στο κεφάλαιο αυτό παράλληλα με την περιγραφή του τρόπου λειτουργίας της τεχνολογίας RFID και των ιδιαίτερων χαρακτηριστικών της.

Τα RFID αποτελούν μια οικονομική, καινοτόμα και αποτελεσματική λύση στην ικνηλασιμότητα των μονάδων διακίνησης όπως containers, παλέτες ακόμη και ατομικά προϊόντα. Το δίκτυο RFID readers αναγνωρίζει την παλέτα, καταγράφει τις κινήσεις της μέσα στην αποθήκη, τη μεταφορά της στα οχήματα και αν υπάρχει το αντίστοιχο δίκτυο στους πελάτες, την άφιξή της στον πελάτη. Ακόμη, σε συνδυασμό με δίκτυο GPRS πάνω στο όχημα καταγράφεται η ακριβής τοποθεσία, ο χρόνος παραμονής και οι συνθήκες διακίνησης. Ακόμη οι ετικέτες R.F.I.D. δεν χρησιμοποιούνται μόνο για ογκώδεις μεταφορές αλλά και για ατομικές συσκευασίες, όπως τρόφιμα, ρούχα ή εξαρτήματα. Παρακάτω ακολουθεί λεπτομερή εξέταση της τεχνολογίας RFID.

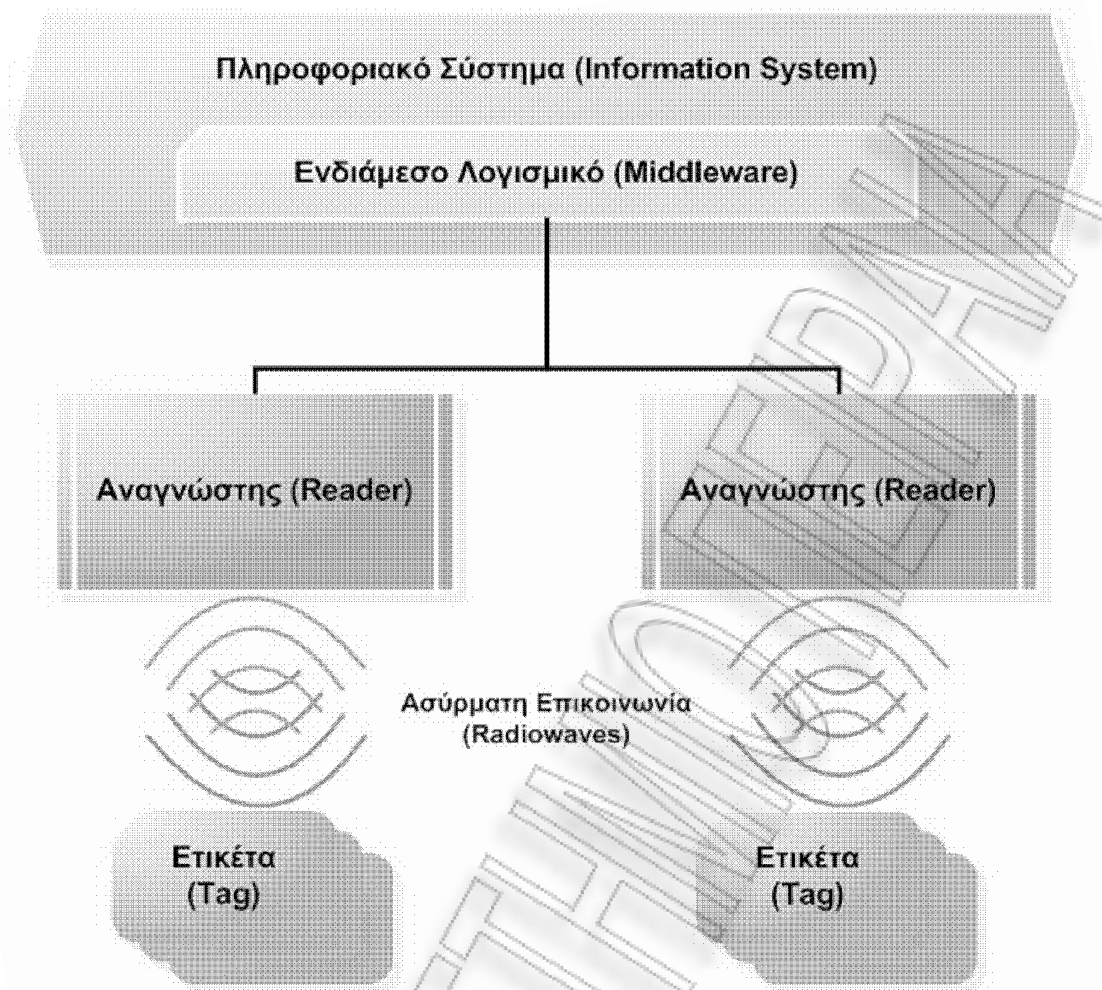
1.4.2.2.1. Αρχιτεκτονική

Η τεχνολογία RFID βρίσκεται στα άκρα ενός πληροφοριακού συστήματος. Είναι στην ουσία ένας διαφορετικός τρόπος διασύνδεσης με αντικείμενα που επιθυμούμε να αναγνωρίζουμε, να εντοπίζουμε και να συλλέγουμε πληροφορίες για αυτά. Η διασύνδεση είναι ασύρματη και βασίζεται στα ραδιοκύματα τα οποία μεταδίδονται στον αέρα. Παράλληλα η αναγνώριση αντικειμένων δεν απαιτεί οπτική επαφή (σε αντίθεση με τον γραμμωτό κώδικα που έχει μέσο διασύνδεσης τις υπέρυθρες και απαιτεί οπτική επαφή).

Ένα σύστημα RFID περιλαμβάνει τρία βασικά στοιχεία:

- την Ετικέτα (*tag*), η οποία αναφέρεται στην βιβλιογραφία και ως πομποδέκτης (*transponder*)
- τον Αναγνώστη (*reader*), ο οποίος αποτελείται από την κεραία (*antenna*) και την μονάδα ελέγχου (*control unit*)
- και το Ενδιάμεσο Λογισμικό (*Middleware*), το οποίο λειτουργεί ως «γέφυρα» επικοινωνίας μεταξύ του αναγνώστη και του πληροφοριακού συστήματος

Η αρχιτεκτονική του συστήματος RFID απεικονίζεται στην που ακολουθεί και αφορά τις τρεις οντότητες που αναφέραμε δηλαδή τις ετικέτες, τους αναγνώστες και το ενδιάμεσο λογισμικό. Τα υπόλοιπα μέρη του πληροφοριακού συστήματος (εξυπηρετητές, δίκτυα, τερματικά κ.α.) παραλείπονται καθώς είναι πέρα από την οπτική αυτού του κεφαλαίου.



Σχήμα 46. Αρχιτεκτονική του συστήματος RFID

1.4.2.2.2. Λειτουργία συστήματος RFID

Η λειτουργία ενός RFID συστήματος βασίζεται στην δυναμική και αμφίδρομη επικοινωνία μεταξύ των μερών που απαρτίζουν το σύστημα, τα οποία περιγράψαμε παραπάνω. Ας δούμε όμως μέσω ενός παραδείγματος ένα τρόπο χρήσης ενός RFID συστήματος σε μια ξενοδοχειακή μονάδα.

Η RFID ετικέτα βρίσκεται προσκολλημένη πάνω σε κάποιο αντικείμενο (π.χ. μια κάρτα εισόδου σε δωμάτιο ξενοδοχείου) και περιέχει συγκεκριμένες πληροφορίες οι οποίες ποικίλουν ανάλογα με τον σκοπό της χρήσης του συστήματος RFID (π.χ. ένα μοναδικό κωδικό για τον προσδιορισμό του δωματίου και ένα μοναδικό κωδικό για τον προσδιορισμό του πελάτη). Ο πελάτης του ξενοδοχείου κρατώντας την κάρτα πλησιάζει στην πόρτα του δωματίου του όπου είναι εγκατεστημένος ένας RFID αναγνώστης. Όταν η κάρτα βρεθεί εντός της εμβέλειας της κεραίας του αναγνώστη αυτόματα η μονάδα ελέγχου επικοινωνεί,

με ραδιοκύματα, με την ετικέτα και παίρνει τις πληροφορίες που χρειάζεται. Εδώ διευκρινίζεται ότι η ετικέτα έχει και αυτή ενσωματωμένη μια κεραία. Στην συνέχεια το ενδιαμέσο λογισμικό, που κατανοεί τα δεδομένα που στέλνει η μονάδα ελέγχου του αναγνώστη, περνάει τις πληροφορίες στη σωστή μορφή στο πληροφοριακό σύστημα του ξενοδοχείου και ελέγχεται αν ο πελάτης μένει στο δωμάτιο με τον συγκεκριμένο αναγνώστη. Τελικά και εφόσον διαπιστωθεί ότι ο συγκεκριμένος πελάτης μένει στο συγκεκριμένο δωμάτιο η πόρτα του δωματίου ξεκλειδώνεται.

Όπως θα διαπιστώσατε η χρήση του RFID αφορά την επικοινωνία αναγνώστη - ετικέτας και στην συνέχεια την μεταφορά των δεδομένων από το ενδιαμέσο λογισμικό στο πληροφοριακό σύστημα. Το παράδειγμά μας είναι αρκετά απλό καθώς σε πραγματικές εφαρμογές επιτελούνται εργασίες εκατέρωθεν μεταξύ πληροφοριακού συστήματος και αναγνώστη - ετικέτας. Για παράδειγμα θα μπορούσε να γίνει μια εγγραφή στην ετικέτα με την χρέωση του πελάτη. Στην περίπτωση αυτή το πληροφοριακό σύστημα δίνει την εντολή στο ενδιαμέσο λογισμικό να γίνει η εγγραφή της ετικέτας, το ενδιαμέσο λογισμικό μεταφέρει σε κατάλληλη μορφή την εντολή αυτή στην μονάδα ελέγχου του αναγνώστη ο οποίος επικοινωνεί με την ετικέτα και γράφει τα δεδομένα που του ζητήθηκαν στην ετικέτα ανανεώνοντας έτσι τα δεδομένα της.

Η αρχιτεκτονική του συστήματος RFID είναι σταθερή ως προς την ροή των δεδομένων (ετικέτα □ αναγνώστης □ ενδιαμέσο λογισμικό □ πληροφοριακό σύστημα) αλλά όχι και ως προς την διακριτότητα των επιμέρους στοιχείων. Συγκεκριμένα παρατηρείται μια τάση για ολοκλήρωση της κεραίας, της μονάδας ελέγχου και του ενδιαμέσου λογισμικού σε μια συσκευή που ονομάζεται αναγνώστης. Σε κάθε περίπτωση η ετικέτα είναι αυτόνομη οντότητα.

1.4.2.2.3. Βασικά στοιχεία συστήματος RFID

Όπως προαναφέρθηκε τα βασικά στοιχεία του συστήματος RFID είναι τρία: η ετικέτα, ο αναγνώστης και το ενδιαμέσο λογισμικό. Τα στοιχεία αυτά ανάλογα με τις ιδιότητες τους καθορίζουν για ποια εφαρμογή είναι κατάλληλα και ποιες είναι οι δυνατότητες της εφαρμογής.

1.4.2.2.3.1. Ετικέτα (Tag)

Η ετικέτα RFID περιλαμβάνει μια κεραία (*antenna*) και ένα ολοκληρωμένο κύκλωμα (*IC*). Η κεραία χρησιμοποιείται για την αμφίδρομη αποστολή σημάτων μέσω των ραδιοκυμάτων με τον αναγνώστη. Το ολοκληρωμένο κύκλωμα είναι αυτό που καθορίζει κάθε φορά αν θα γίνει εκπομπή ή λήψη δεδομένων και έχει την δυνατότητα να τα αποθηκεύει στην μνήμη του. Η μνήμη κυμαίνεται από 4 μέχρι 128KB.

Οι ετικέτες κατηγοριοποιούνται σε:

- παθητικές (*passive*), ημιπαθητικές - ημιενεργητικές (*semi passive or semi active*) και ενεργητικές (*active*).
- αναγνώσιμες (*Read only*), μίας εγγραφής-πολλών αναγνώσεων (*Write Once Read Many*) και επανεγγράψιμες (*Read - Write*).

Εν γένει θα μπορούσαν να κατηγοριοποιηθούν ακόμα ως προς τις φυσικές τους διαστάσεις, την κατασκευή τους και ως προς την εφαρμογή τους. Στην συνέχεια του κεφαλαίου θα μιλήσουμε και για αυτές τις κατηγοριοποιήσεις.

Παθητικές, ημιπαθητικές-ημιενεργητικές και ενεργητικές ετικέτες

Οι ετικέτες κατηγοριοποιούνται κυρίως σε παθητικές και ενεργητικές ανάλογα με την πηγή ενέργειας τους. Οι ενεργητικές ετικέτες διαθέτουν μπαταρία, η οποία είναι ενσωματωμένη στην ετικέτα ενώ οι παθητικές ετικέτες αντλούν την ενέργεια τους από το σήμα που στέλνει ο αναγνώστης. Στον πίνακα που ακολουθεί παρουσιάζονται συνολικά οι διαφορές μεταξύ παθητικών και ενεργητικών ετικετών.

Ετικέτες	Παθητικές (<i>Passive</i>)	Ενεργητικές (<i>Active</i>)
Πηγή Ενέργειας	Λειτουργούν χωρίς μπαταρία. Κατά την είσοδο τους στο πεδίο εκπομπής του αναγνώστη ενεργοποιούνται λαμβάνοντας ενέργεια από τα σήματα του αναγνώστη	Απαιτούν μπαταρία για την λειτουργία τους. Όταν εισέρχονται στο πεδίο του αναγνώστη αυτοενεργοποιούνται
Χρόνος Ζωής	Απεριόριστος	Περιορισμένος (<i>battery-dependent</i>)
Μέγεθος	Μικρό (προσαρμοζόμενο)	Μεγάλο (απουσία ευελιξίας)
Κόστος	Χαμηλό (20 λεπτά - 3€)	Υψηλό (20€ και άνω)
Ισχύς Εκπομπής Αναγνώστών	Ισχυρή εκπομπή	Όχι ιδιαίτερες απαιτήσεις εκπομπής
Απόσταση Ανάγνωσης	Μικρή (20cm – 6m)	Μεγάλη (30m – 40m)
Φωτογραφία		

Σχήμα 47. Διαφορές μεταξύ παθητικών και ενεργητικών ετικετών

Επίσης υπάρχει και μια τρίτη υποκατηγορία ετικετών που ονομάζονται ημιπαθητικές ή ημιενεργητικές ετικέτες. Οι ετικέτες αυτές περιέχουν μπαταρία η οποία όμως δεν χρησιμοποιείται για τη μετάδοση ραδιοκυμάτων στον αναγνώστη παρά μόνο για τη λειτουργία του ολοκληρωμένου κυκλώματος τους (π.χ. μπορούν να έχουν ενσωματωμένο ένα αισθητήρα θερμοκρασίας μετρώντας την θερμοκρασία του περιβάλλοντος ανά τακτά χρονικά διαστήματα και όταν εισέλθουν στο πεδίο εκπομπής του αναγνώστη μεταδίδουν τα δεδομένα που έχουν αποθηκεύσει). Στην βιβλιογραφία αναφέρονται συνήθως ως ενεργητικές ετικέτες καθώς περιέχουν μπαταρία. Όμως λόγω των μικρότερων απαιτήσεων τους σε ισχύ η μπαταρία άρα και το μέγεθος τους είναι σημαντικά μικρότερο γεγονός που τις κάνει και φθηνότερες από τις ενεργητικές ετικέτες. Η απαιτούμενη απόσταση ανάγνωσης των ημιπαθητικών ή ημιενεργητικών ετικετών είναι μεγαλύτερη από αυτή των παθητικών και μικρότερη από αυτή των ενεργητικών ετικετών. Τέλος οι ετικέτες αυτές είναι συνήθως μιας χρήσης, δηλαδή όταν αποφορτιστεί η μπαταρία τους αχρηστεύονται.

Αναγνώσιμες, μίας εγγραφής-πολλών αναγνώσεων και επανεγγράψιμες ετικέτες

Όπως προαναφέρθηκε οι ετικέτες έχουν μνήμη, λόγω του ολοκληρωμένου κυκλώματος που περιέχουν. Επομένως οι ετικέτες κατηγοριοποιούνται ανάλογα με την δυνατότητα επανεγγραφής τους σε αναγνώσιμες, μίας εγγραφής-πολλών αναγνώσεων και επανεγγράψιμες.

Οι αναγνώσιμες ετικέτες εγγράφονται μία φορά με τα κατάλληλα δεδομένα κατά την κατασκευή τους (συνήθως ένα σειριακό αριθμό και ένα ψηφίο ελέγχου) και οι αναγνώστες μπορούν μόνο να διαβάσουν τα δεδομένα και όχι να τα τροποποιήσουν.

Οι ετικέτες μίας εγγραφής-πολλών αναγνώσεων εγγράφονται κατά την κατασκευή τους, μπορούν όμως να εγγραφούν και από τον χρήστη μόνο μια φορά ακόμα. Έπειτα μετατρέπονται σε αναγνώσιμες ετικέτες.

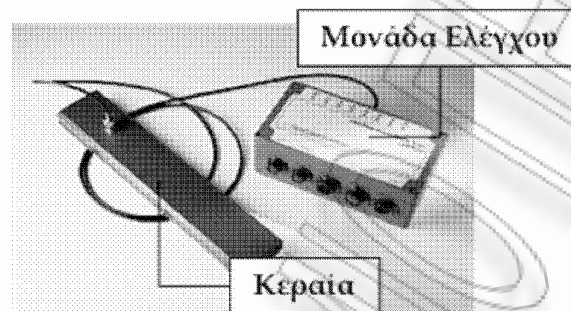
Οι επανεγγράψιμες ετικέτες εγγράφονται κατά την κατασκευή τους, όμως οι αναγνώστες έχουν την δυνατότητα εκτός από το να διαβάζουν τα δεδομένα τους, να τα τροποποιούν (εισαγωγή, διαγραφή) απεριόριστα. Στον πίνακα που ακολουθεί παρουσιάζονται συνολικά οι διαφορές μεταξύ αναγνώσιμων και επανεγγράψιμων ετικετών.

Ετικέτες	Αναγνώσιμες (Read Only)	Μίας εγγραφής - Πολλών Αναγνώσεων (WORM)	Επανεγγράψιμες (Read - Write)
Ανάγνωση	Απεριόριστα	Απεριόριστα	Απεριόριστα
Εγγραφή κατά την κατασκευή	Ναι	Ναι	Ναι
Εγγραφή κατά την χρήση	Όχι	Μία φορά μόνο	Απεριόριστα
Ευελξία	Μικρή	↔	Μεγάλη
Ασφάλεια	Μεγάλη	↔	Μικρή
Κόστος	Μικρό	↔	Μεγάλο
Εφαρμογές	Έλεγχος πρόσβασης	Διαχείριση εφοδιαστικής αλυσίδας	Αυτόματη συλλογή διοδίων, έλεγχος βιομηχανικής παραγωγής

Σχήμα 48. Διαφορές μεταξύ αναγνώσιμων και επανεγγράψιμων ετικετών

1.4.2.3.2. Αναγνώστες

Ο αναγνώστης είναι μια συσκευή που αναλαμβάνει να επικοινωνήσει με την ετικέτα μέσω των ραδιοκυμάτων και για το λόγο αυτό ενσωματώνει κεραία. Επίσης περιέχει μια μονάδα ελέγχου που καθορίζει τις ενέργειες που κάνει ο αναγνώστης (αποστολή/λήψη σημάτων, ανάγνωση/ εγγραφή ετικετών κ.α.), ενέργειες που καθορίζονται από το ενδιάμεσο λογισμικό. Επίσης η μονάδα ελέγχου αναλαμβάνει την επικοινωνία με το πληροφορικό σύστημα μέσω του ενδιάμεσου λογισμικού που παίζει το ρόλο μεταφραστή και για τις δύο πλευρές.



Σχήμα 49. Barcode αναγνώστης

Ανάλογα με την εφαρμογή, τις τεχνικές ιδιότητες και τις φυσικές διαστάσεις τους, οι αναγνώστες κατηγοριοποιούνται σε:

1. Σταθερούς Αναγνώστες
2. Ολοκληρωμένους Αναγνώστες
3. Αναγνώστες Χειρός
4. Ενσωματωμένους Αναγνώστες

Στους πίνακες που ακολουθούν περιγράφονται οι ιδιότητες για κάθε μία από τις κατηγορίες των αναγνώστών.

Αναγνώστες	Σταθεροί	Ολοκληρωμένοι
Γενικά Χαρακτηριστικά	Περιέχουν 2 – 8 κεραίες	Περιέχουν 1 κεραία
Εφαρμογές	Χρησιμοποιούνται κυρίως στην εφοδιαστική αλυσίδα (σε εισόδους αποβάθρων φόρτωσης/ εκφόρτωσης, σε ταινίες μεταφοράς προϊόντων)	Χρησιμοποιούνται κυρίως σε εφαρμογές ελέγχου πρόσβασης (σε εισόδους/ εξόδους κρίσιμων υποδομών)
Τεχνικά Χαρακτηριστικά	16-bit/ 32-bit επεξεργαστές, περιέχουν λειτουργικό σύστημα, δυνατότητα επεξεργασίας σήματος	16-bit επεξεργαστές, περιέχουν λειτουργικό σύστημα, αυξημένες δυνατότητες ανάγνωσης εγγραφής
Δικτύωση	TCP/ IP ανεξάρτητοι κόμβοι, κατέχουν δικό τους API, χρησιμοποιούν μία σειρά από πρωτόκολλα (DHCP, HTTP, Telnet or SSH, NTP, SNMP)	Σπάνια TCP/ IP ανεξάρτητοι κόμβοι, συνήθως χρησιμοποιούν σύνδεση σειριακή (RS-232) ή USB
Φωτογραφία		

Σχήμα 50. Ιδιότητες για κάθε μία από τις κατηγορίες των αναγνωστών

Αναγνώστες	Χειρός	Ενσωματωμένοι
Γενικά Χαρακτηριστικά	Περιέχουν 1 κεραία	Περιέχουν 1 κεραία
Εφαρμογές	Χρησιμοποιούνται κυρίως στην εφοδιαστική αλυσίδα για έλεγχο αποθέματος	Χρησιμοποιούνται κυρίως για ενσωμάτωση σε συσκευές όπως οι εκτυπωτές ετικετών RFID, ταξινόμητες κιβωτίων, τερματικά POS
Τεχνικά Χαρακτηριστικά	16-bit/ 32-bit επεξεργαστές, περιέχουν λειτουργικό σύστημα, δυνατότητα επεξεργασίας σήματος	Δεν περιέχουν επεξεργαστή, δεν περιέχουν λειτουργικό
Δικτύωση	Ασύρματοι TCP/ IP κόμβοι, συνδέονται απευθείας με εξυπηρετητές (συνήθως περιοδικά) χρησιμοποιώντας εφαρμογές μεταφοράς δεδομένων	Δεν έχουν ικανότητες δικτύωσης, χρησιμοποιούν σύνδεση USB, Σειριακή (RS-232) or PCMCIA
Φωτογραφία		

Σχήμα 51. Ιδιότητες για κάθε μία από τις κατηγορίες των αναγνώστων 2

1.4.2.2.3.3. Ενδιάμεσο Λογισμικό (Middleware)

Το ενδιάμεσο λογισμικό είναι ο «αντιπρόσωπος» του RFID αναγνώστη στο πληροφοριακό σύστημα της εκάστοτε εταιρίας. Αναλαμβάνει να προωθεί τόσο προς τον αναγνώστη τα δεδομένα και τις εντολές που δέχεται από το πληροφοριακό σύστημα όσο και τα δεδομένα και τις εντολές που δέχεται από τον αναγνώστη προς το πληροφοριακό σύστημα. Οι εντολές προς τον αναγνώστη αφορούν κυρίως πράξεις που πρέπει να γίνουν πάνω σε μια ετικέτα (εύρεση ετικέτας, ανάγνωση κωδικού ετικέτας, ανάγνωση δεδομένων ετικέτας, εγγραφή δεδομένων στην ετικέτα, καταστροφή ετικέτας κ.α.) αλλά και πράξεις που αφορούν τον ίδιο τον αναγνώστη (ανάγνωση κατάστασης αναγνώστη, αλλαγή ρυθμίσεων αναγνώστη, ανάγνωση κωδικού αναγνώστη κ.α.) και ονομάζονται ως εντολές αναγνώστη. Τα δεδομένα που μεταφέρονται εκατέρωθεν μεταξύ αναγνώστη και πληροφοριακού συστήματος αφορούν είτε τα δεδομένα που αποθηκεύονται σε μια ετικέτα είτε δεδομένα που απαιτούνται για την επικοινωνία μεταξύ Π.Σ. και αναγνώστη.

1.4.2.2.4. Πρότυπα

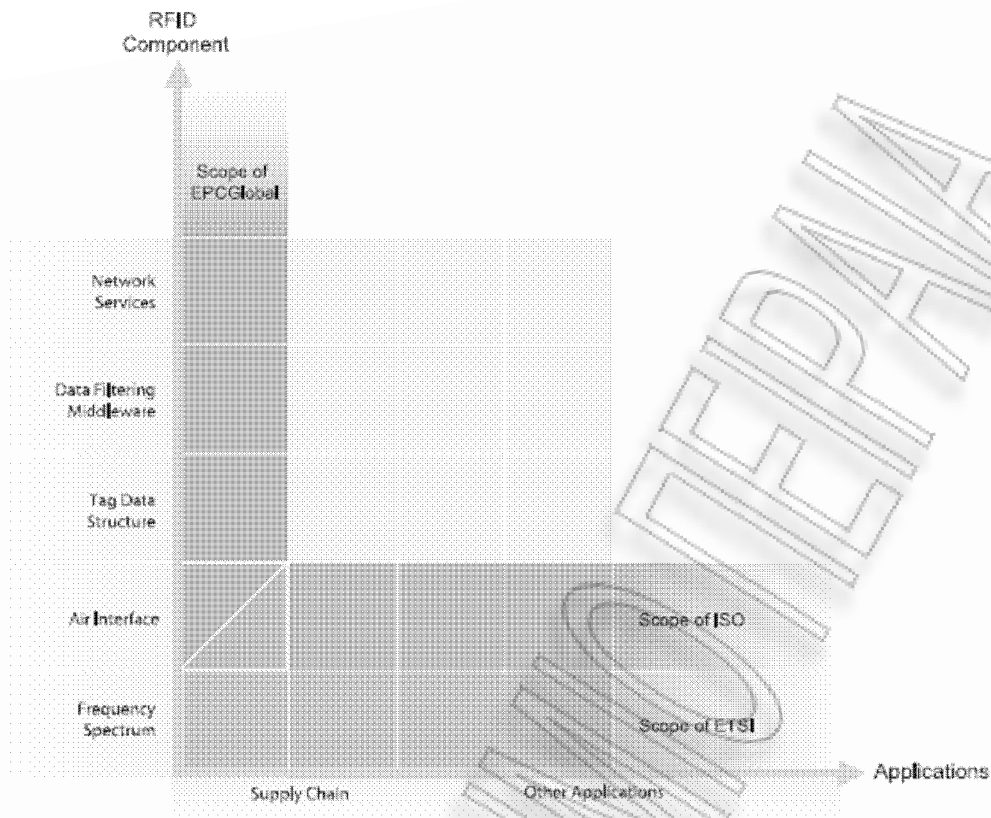
Η τεχνολογία RFID χρησιμοποιεί τις ραδιοσυχνότητες και για το λόγο αυτό απαιτούνται πρότυπα που θα καθορίζουν ποιο κομμάτι του φάσματος συχνοτήτων θα δεσμεύει, τα επίπεδα εκπομπής και θέματα παρεμβολών με άλλες ράδιο-υπηρεσίες.

Επίσης η ύπαρξη πολλών κατασκευαστών - προμηθευτών τεχνολογίας RFID δημιουργεί πρόβλημα στον καταναλωτή (στην συγκεκριμένη περίπτωση ο καταναλωτής είναι η εταιρία που θα εγκαταστήσει ένα σύστημα RFID) που καλείται να επικοινωνήσει με διαφορετικά συστήματα RFID (π.χ. πως θα γνωρίζει μια εταιρία ποιο είναι το κατάλληλο σύστημα RFID για μια εφαρμογή ελέγχου πρόσβασης).

Παράλληλα το όραμα της αγοράς για ένα ανοικτό και παγκόσμιο σύστημα διαχείρισης της εφοδιαστικής αλυσίδας, με χρήση της τεχνολογίας RFID, απαιτεί πρότυπα προκειμένου αυτό να γίνει πραγματικότητα. Για τους παραπάνω λόγους έχουν αναπτυχθεί μια σειρά από πρότυπα από συγκεκριμένους οργανισμούς που είναι οι:

- ❖ Παγκόσμιος Οργανισμός Προτυποποίησης (*ISO, International Organization for Standardization*).
- ❖ Παγκόσμιο Ηλεκτροτεχνικό Συμβούλιο (*IEC, International Electrotechnical Council*).
- ❖ Ευρωπαϊκό Ινστιτούτο Προτύπων Τηλεπικοινωνιών (*ETSI, European Telecommunications Standards Institute*).
- ❖ EPC global.

Ο κάθε οργανισμός στοχεύει σε μια διαφορετική πτυχή της τεχνολογίας RFID και αναπτύσσει πρότυπα για αυτή. Στο Διάγραμμα που ακολουθεί φαίνονται οι σχέσεις μεταξύ τεχνολογίας RFID και οργανισμών.



Σχήμα 52. RFID και πρότυπα

1.4.2.2.5. Πλεονεκτήματα Τεχνολογίας RFID

Μερικά από τα οφέλη που μπορούν να αποκομίσουν οι επιχειρήσεις με τη χρήση της τεχνολογίας RFID είναι:

1. Μείωση κόστους
2. Αύξηση παραγωγικότητας
3. Μείωση σε λάθη, κλοπές και πλαστογραφίες
4. Ενημέρωση του προσωπικού σε πραγματικό χρόνο
5. Αύξηση αποδοτικότητας και ποιότητας υπηρεσιών
6. Ακρίβεια και αποδοτικότητα στις παραλαβές
7. Διαφάνεια στη διαχείριση
8. Μείωση αποθεμάτων

9. Αποδοτικότητα και ακρίβεια στην αποστολή
10. Βοήθεια στην ανάκληση προϊόντων
11. Μείωση προϊόντων που δεν διακινούνται
12. Μείωση των περιπτώσεων έλλειψης αποθέματος (out-of-stock)
13. Δεν απαιτείται πλέον ούτε το άνοιγμα ενός κιβωτίου ούτε πολύ περισσότερο η καταμέτρηση του περιεχομένου

1.4.2.2.6. Μειονεκτήματα Τεχνολογίας RFID

Φυσικά, υπάρχουν και ορισμένα μειονεκτήματα που χρειάζεται να ξεπεραστούν, δεδομένου ότι η τεχνολογία δεν είναι ακόμη αρκετά ώριμη. Για παράδειγμα, το κόστος των ετικετών κατά κύριο λόγο, αλλά και γενικότερα του εξοπλισμού, παραμένει σχετικά υψηλό. Επιπλέον, σε αρκετές περιπτώσεις σημειώνεται ασυμφωνία σχετικά με τις χρησιμοποιούμενες συχνότητες, με αποτέλεσμα οι κατασκευαστές να προωθούν προϊόντα για διαφορετικές συχνότητες και κατά συνέπεια να υπάρχει πρόβλημα διαλειτουργικότητας. Ακόμη, καταγράφονται επιμέρους προβλήματα σε ιδιαίτερες εφαρμογές, όπως για παράδειγμα, όταν η ετικέτα - πομποδέκτης (tag) βρίσκεται κοντά σε υγρά ή μέταλλα. Όλα αυτά πάντως αναμένεται να ξεπεραστούν στο προσεχές μέλλον.

1.4.2.2.7. Σύγκριση RFID & Barcode

Κάποιοι εκτιμούν ότι το RFID (Radio Frequency Identification) θα είναι απλώς η ασύρματη εκδοχή του barcode για τα επόμενα χρόνια, όμως όλα δείχνουν ότι η αξία αυτής της τεχνολογίας δεν σταματά εκεί. Οι λόγοι που συναινούν σε αυτό είναι πολλοί, όπως:

- ✓ Μια ετικέτα RFID μπορεί να μεταφέρει αρκετές πιο χρήσιμες πληροφορίες από ένα barcode, όπως για παράδειγμα την ημερομηνία λήξεως, στοιχείο ιδιαίτερα χρήσιμο για πολλά ευπαθή προϊόντα όπως π.χ. το γάλα.
- ✓ Τα barcodes είναι μια "line-of-sight" τεχνολογία, κάτι που σημαίνει ότι ο scanner θα πρέπει να "βλέπει" το γραμμωτό κώδικα για να τον διαβάσει. Αντίθετα, οι ετικέτες RFID δεν απαιτούν από τον αναγνώστη κάτι τέτοιο και μπορούν να διαβαστούν όσο βρίσκονται μέσα στην ακτίνα ανάγνωσής του.

Παρόλα αυτά, και για το άμεσο τουλάχιστον μέλλον, δεν διαφαίνεται αντικατάσταση των barcodes, τα οποία είναι σαφώς φθηνότερα από τις ετικέτες RFID, αλλά και αποτελεσματικά σε συγκεκριμένους τομείς. Έτσι, το πιο πιθανό είναι τα barcodes και το RFID να συνυπάρχουν για αρκετά χρόνια.

1.4.2.2.8. Θέματα Ασφαλείας Τεχνολογίας RFID

Οι ετικέτες RFID που οι εταιρίες χρησιμοποιούν ή σκοπεύουν να χρησιμοποιήσουν στην εφοδιαστική αλυσίδα στο εγγύς μέλλον και στη συσκευασία των προϊόντων σε μακροπρόθεσμο στάδιο, θα περιλαμβάνουν μόνο έναν Ηλεκτρονικό Κωδικό Προϊόντος (EPC). Ο κωδικός αυτός θα συνδέεται με δεδομένα που βρίσκονται σε online βάσεις δεδομένων. Μερικές πληροφορίες σχετικές με το προϊόν μπορεί να είναι προσβάσιμες από τον καθένα (π.χ. η ταυτότητα του προϊόντος), αλλά κάποιες άλλες (όπως ο τόπος και ο χρόνος κατασκευής) θα είναι προσβάσιμες μόνο από αυτούς στους οποίους ο κατασκευαστής θέλει να διαθέσει την πληροφορία. Έτσι π.χ. μια αλυσίδα τροφίμων δεν θα μπορεί να έχει πρόσβαση σε πληροφορίες προϊόντων που πωλεί μια ανταγωνιστική της εταιρία.

Επίσης, ένα άλλο θέμα που ενδέχεται να προκύψει αφορά στη δυνατότητα που μπορεί να αποκτήσουν οι Αρχές (π.χ. κυβερνήσεις, αρχές ασφάλειας) να παρακολουθούν τους καταναλωτές εκείνους που έχουν αγοράσει και φορούν ή μεταφέρουν προϊόντα με ετικέτες RFID, όπως πορτοφόλι, ρούχα, κ.λ.π. Βεβαίως, σε αυτή την περίπτωση ο καταναλωτής μπορεί εύκολα να προστατευθεί αφαιρώντας την ετικέτα από το αντικείμενο.

Γενικά οι απειλές που μπορεί να δεχθεί ένα σύστημα RFID είναι:

- ❖ Κακόβουλη τροποποίηση δεδομένων
- ❖ Πλαστή ταυτότητα ετικέτας
- ❖ Απενεργοποίηση
- ❖ Αποκόλληση
- ❖ Παρακολούθηση
- ❖ Μπλοκάρισμα
- ❖ Παρεμβολή
- ❖ Πλαστή ταυτότητα αναγνώστη

Τέλος, και επειδή έχουν προκύψει ερωτήματα σχετικά με πιθανούς κινδύνους για την υγεία, θα πρέπει να σημειώσουμε ότι η τεχνολογία RFID χρησιμοποιεί τις χαμηλές συχνότητες του ηλεκτρομαγνητικού φάσματος, και για το λόγο αυτό τα κύματα που εκπέμπουν οι αναγνώστες δεν είναι επικίνδυνα για τη δημόσια υγεία.

1.4.3. Ηλεκτρονική Επεξεργασία Δεδομένων (Electronic Data Processing - EDP)

Η ηλεκτρονική επεξεργασία των δεδομένων πραγματοποιείται μέσω ειδικών λογισμικών διαχείρισης παρτίδων - ικνηλασιμότητας. Τα λογισμικά διαχείρισης παρτίδων - ικνηλασιμότητας, σε συνδυασμό με την υψηλή τεχνολογία, μάς δίνουν πλέον την δυνατότητα πέραν της πληροφορίας που χρειάζεται για να εκδοθεί το σχετικό παραστατικό πώλησης να συλλέξουμε και να συσχετίσουμε τα είδη και τις παρτίδες τους με το παραστατικό και τον πελάτη. Στην συνέχεια η πληροφορία μπορεί να μεταφερθεί ηλεκτρονικά στο κεντρικό σύστημα του διανομέα και πιθανόν να φτάσει μέχρι τον κατασκευαστή. Έτσι έχουμε την δυνατότητα ικνηλάτησης των προϊόντων μέχρι το σημείο λιανικής. Ως χθες αυτό φάνταζε αδύνατον ή τουλάχιστον πολύ δύσκολο να υλοποιηθεί. Όμως με τα κατάλληλα εργαλεία η καταγραφή της πληροφορίας των παρτίδων για κάθε προϊόν ξεχωριστά και η άμεση ενημέρωση του κεντρικού συστήματος είναι πλέον εφικτή.

Τα κύρια δεδομένα που διαχειρίζονται τα συγκεκριμένα λογισμικά είναι οι αριθμοί παρτίδας των πρώτων υλών, οι ημερομηνίες λήξης των πρώτων υλών, οι αριθμοί παρτίδας των ενδιάμεσων προϊόντων, οι αριθμοί παρτίδας των τελικών προϊόντων, οι ημερομηνίες λήξης των τελικών προϊόντων, στοιχεία για τις ανακληθέντες παρτίδες, κ.α.

Η βασικότερη επεξεργασία των παραπάνω δεδομένων αφορά την συσχέτιση των πρώτων υλών, των ενδιάμεσων προϊόντων, των τελικών προϊόντων και των πελατών. Η συσχέτιση αυτή μας παραθέτει στοιχεία απαραίτητα για την επιχείρηση καθώς και για την λειτουργία της ανάκλησης.

Ακολουθεί μια σύντομη παρουσίαση ορισμένων λογισμικών διαχείρισης παρτίδων - ικνηλασιμότητας:

- *GL1 Trace* από την εταιρία G - Logic.
- «*Ικνηλάτης*» από την εταιρία Sinodos.com.
- *TracePro* από την εταιρία Acmon Systems A.E.
- *ItemTracker* από την εταιρία Agrosoft.

Τέλος, αξίζει να αναφέρουμε ότι τα περισσότερα εμπορικά προγράμματα ERP / ERM νέας γενιάς υποστηρίζουν την ύπαρξη παρτίδων, όπως άλλωστε γίνεται και με το λογισμικό διαχείρισης της αποθήκης WHMS. Σε περιπτώσεις απουσίας ειδικού λογισμικού διαχείρισης των παρτίδων, μπορεί η διαχείριση των παρτίδων ή της ημερομηνίας λήξεων να γίνεται από την βάση δεδομένων του προγράμματος διαχείρισης των Πωλήσεων επί Αυτοκινήτου. Τα συστήματα Πώλησης επί Αυτοκινήτου νέας γενιάς χρησιμοποιούν πλέον συσκευές EDA's (Enterprise Digital Assistant) με ενσωματωμένο αναγνώστη γραμμωτού κώδικα, που έχει την δυνατότητα ανάγνωση των ετικετών με κώδικα EAN 128 που ενσωματώνει αρκετές

πληροφορίες σχετικά με το προϊόν όπως είναι ο κωδικός είδους, η παρτίδα (LOT number), το βάρος ή η ημερομηνία λήξεως.

1.4.4. Ηλεκτρονική Ανταλλαγή Δεδομένων (Electronic Data Interchange - EDI)

Κάθε μέρα, οι επιχειρήσεις παράγουν και επεξεργάζονται έναν τεράστιο όγκο εγγράφων. Τα έγγραφα αυτά, που ξεκινούν από δελτία παραγγελιών και τιμολόγια και φτάνουν έως τους καταλόγους προϊόντων και τις αναφορές πωλήσεων παρέχουν ζωτικής σημασίας πληροφορίες οι οποίες προηγούνται, συνοδεύουν ή ακολουθούν τα φυσικά αγαθά σε εμπορική συναλλαγή.

Οι περισσότερες διαδικασίες δεν μπορούν να διεξαχθούν σωστά αν δεν υπάρχουν διαθέσιμες οι κατάλληλες πληροφορίες για το είδος, όπως π.χ. στην περίπτωση που ο ταμίας περνά το είδος από τον scanner, αλλά στην οθόνη του ταμείου εμφανίζεται το μήνυμα «άγνωστο είδος».

Συνεπώς, εκτός από τη φυσική ροή των αγαθών απαιτείται να υπάρχει και μια παράλληλη ροή πληροφοριών μεταξύ των εμπορικών εταιρών. Οι παρεχόμενες πληροφορίες θα πρέπει να είναι πλήρεις και σαφείς. Συγκεκριμένα:

1. Το όνομα του προμηθευτή και ο GLN (Διεθνής Κωδικός Θέσης) της εταιρίας.
2. Η ημερομηνία της υλοποίησης (ημερομηνία από τη οποία οι εμπορικοί εταίροι μπορούν να χρησιμοποιήσουν τις πληροφορίες).
3. Ο GTIN του προϊόντος.
4. Η πλήρης περιγραφή για τα μηνύματα EDI ή για τα έγγραφα της συναλλαγής και η σύντομη περιγραφή για την απόδειξη του ταμείου.
5. Τα φυσικά χαρακτηριστικά του προϊόντος (διαστάσεις, καθαρό βάρος).
6. Η περιγραφή των διαφόρων τυποποιημένων ομαδοποιήσεων αυτής της Μονάδας Εμπορίας, συμπεριλαμβανομένου και του πλήθους των μεμονωμένων Μονάδων Εμπορίας που περιέχονται σε μεγαλύτερες Μονάδες.
7. Η περιγραφή της παλέτας ή του τρόπου τοποθέτησης στην παλέτα.
8. Για κάθε Μονάδα Εμπορίας, ο GTIN και η ποσότητα των μονάδων του αμέσως προηγούμενου επίπεδου συσκευασίας (αν υπάρχει φυσικά).

Η ανταλλαγή των παραπάνω πληροφοριών, μέσω μηνυμάτων, από ένα σύστημα Η/Υ σε κάποιο άλλο γίνεται χρησιμοποιώντας ένα αποτελεσματικό εργαλείο το οποίο το παρέχει το EDI (Ηλεκτρονική Ανταλλαγή Δεδομένων). Ακολουθεί λεπτομερή ανάλυση του συγκεκριμένου εργαλείου.

Το EDI (Ηλεκτρονική Ανταλλαγή Δεδομένων), όπως προαναφέραμε, παρέχει στους εμπορικούς εταίρους ένα αποτελεσματικό εργαλείο για την αυτόματη αποστολή εμπορικών δεδομένων από ένα σύστημα Η/Υ σε κάποιο άλλο. Στο EDI αντικαθίσταται η ανταλλαγή των έντυπων εγγράφων μεταξύ επιχειρήσεων με μηνύματα, κατάλληλα να αποσταλούν ηλεκτρονικά μεταξύ Η/Υ. Αυτή η ανταλλαγή αφορά στις εμπορικές συναλλαγές αλλά και στις σχετικές διαδικασίες του εμπορικού τομέα, του τομέα Logistics και του οικονομικού τομέα.

Για κάθε επιχείρηση, η επιτυχής υλοποίηση της εν λόγω τεχνολογίας απαιτεί υψηλό βαθμό αφοσίωσης, όχι μόνο από τα ανώτερα διοικητικά στελέχη αλλά και από μια ευρεία κλίμακα στελεχών, που είναι υπεύθυνα για διάφορους τομείς δραστηριοτήτων μέσα στην επιχείρηση. Θα πρέπει να εξεταστούν οι υπάρχουσες εταιρικές πολιτικές και διαδικασίες, πρέπει να αναθεωρηθούν ίσως οι τρέχουσες λειτουργικές διαδικασίες, ενώ θα δημιουργηθούν νέες επιχειρηματικές σχέσεις, οι οποίες απαιτούν σωστή διαχείριση.

Στο κέντρο του συστήματος, βρίσκεται η καλύτερη χρήση και ανταλλαγή των πληροφοριών εσωτερικά, αλλά και μεταξύ των εμπορικών εταιρών, έτσι ώστε τα εξαρτώμενα μέρη να είναι καλύτερα πληροφορημένα και πιο αξιόπιστα.

Στο σύστημα GS1, το EDI υλοποιείται μέσω τυποποιημένων μηνυμάτων EANCOM® και XML.

Το πρότυπο EANCOM® είναι ένας λεπτομερής οδηγός για την υλοποίηση των τυποποιημένων μηνυμάτων UN/EDIFACT. Περιλαμβάνει μηνύματα με σαφείς ορισμούς και επεξηγήσεις του τρόπου χρήσης όλων των πεδίων δεδομένων, επιτρέποντας στους εμπορικούς εταίρους να ανταλλάσσουν εμπορικά έγγραφα και στοιχεία που αφορούν τα προϊόντα με απλό, ακριβή και οικονομικό τρόπο.

Υπάρχουν διάφοροι τύποι μηνυμάτων, οι οποίοι ανταποκρίνονται σε όλες τις επιχειρηματικές απαιτήσεις στα διάφορα στάδια μιας εμπορικής σχέσης:

1. Τα κύρια μηνύματα δεδομένων περιγράφουν τα σχετικά συναλλασσόμενα μέρη και τα προϊόντα.
2. Η εμπορική συναλλαγή ξεκινά με το μήνυμα παραγγελίας και τελειώνει με το μήνυμα ειδοποίησης πολλαπλής χρέωσης ή πίστωσής, ακολουθώντας την λογική σειρά ενός κύκλου εμπορικών συναλλαγών.
3. Τα μηνύματα αναφορών και σχεδιασμού χρησιμοποιούνται για την ενημέρωση του εμπορικού εταίρου σχετικά με την εμπορική δραστηριότητα ή για τον σχεδιασμό των μελλοντικών βελτιώσεων, επιτρέποντας έτσι την εξομάλυνση των διαδικασιών της εφοδιαστικής αλυσίδας.
4. Γενικά μηνύματα, τα οποία χρησιμοποιούνται για την αποστολή γενικών πληροφοριών σε μια ή περισσότερες διευθύνσεις.

Το EANCOM® δεν είναι μόνο ένα σύνολο τυποποιημένων μηνυμάτων, αλλά βασίζεται και στη χρήση των διεθνών κωδικών GS1 αντί των κωδικών που συμφωνούνται μεταξύ δυο μόνο εμπορικών εταιρών. Η χρήση των κωδικών GS1, θα απλοποιήσει με φυσιολογικό τρόπο την υιοθέτηση του EDI από μελλοντικούς εμπορικούς εταίρους.

Ο GTIN, ο οποίος περιγράφεται παραπάνω και χρησιμοποιείται για την αναγνώριση Μονάδων Εμπορίας, αποτελεί το μόνο διεθνές και πολυτομεακό σύστημα κωδικοποίησης που παρέχει έναν μοναδικό και σαφή κωδικό αναγνώρισης για κάθε είδος και τις παραλλαγές του, άσχετα με τον τόπο προέλευσης του και τον προορισμό του. Η χρήση του στα μηνύματα EANCOM® είναι ιδιαίτερα σημαντική για τα ανοικτά περιβάλλοντα. Οι εταιρίες δεν θα χρειάζεται να τηρούν περίπλοκους καταλόγους για την διασταύρωση των στοιχείων με βάση τους εσωτερικούς κωδικούς κάθε εμπορικού εταίρου.

Ο GLN αντίστοιχα, παρέχει τον πιο αποτελεσματικό τρόπο αποστολής πληροφοριών σχετικά με την «θέση» ή τα στοιχεία αναγνώρισης της εταιρίας. Πέραν της χρήσης του στα μηνύματα EANCOM®, μπορεί να χρησιμοποιείται στα δίκτυα για την δρομολόγηση μηνυμάτων EDI στο καθορισμένο «γραμματοκιβώτιο», στον καθορισμένο σταθμό εργασίας ή στην καθορισμένη εφαρμογή.

Τα μηνύματα EANCOM® έχουν σχεδιαστεί κατάλληλα ώστε να εκμεταλλεύονται πλήρως τα σχετικά πρότυπα που αφορούν την αναγνώριση προϊόντων και «θέσεων» και τη χρήση συμβόλων barcode, παρέχοντας μέγιστη αποδοτικότητα και οφέλη για τον χρήστη. Η χρήση τέτοιων μηνυμάτων και προτύπων αυξάνει καθημερινά σε παγκόσμια κλίμακα.

Η Ηλεκτρονική Ανταλλαγή Δεδομένων που επιτυγχάνεται με τη χρήση μηνυμάτων EANCOM®, απαιτεί τη χρησιμοποίηση ειδικών συνδέσεων - τα Value Added Networks. Τα VANs αν και ακριβά ακόμα στη χρήση τους, είναι πολύ αξιόπιστα και απαιτούν ειδική παροχή υπηρεσιών. Αυτός είναι και ο κύριος λόγος που οι εταιρίες οι οποίες ασχολήθηκαν με αυτό το είδος δικτύων, είναι οι μεγαλύτερες στο χώρο. Οι μικρομεσαίες επιχειρήσεις συνέχισαν τη χρήση του ξεπερασμένου και επιρρεπή σε λάθη παραδοσιακού τρόπου ανταλλαγής των εμπορικών τους εγγράφων.

Ταυτόχρονα με την αλματώδη ανάπτυξη του internet, οι επιχειρήσεις εξέφρασαν την ανάγκη τους να χρησιμοποιήσουν αυτό το μέσο (internet), για την ανταλλαγή των εμπορικών τους εγγράφων. Η απάντηση στην ανάγκη της αυτόματης ανταλλαγής των εμπορικών εγγράφων από Η/Υ σε Η/Υ δια μέσω του internet, δόθηκε από τη γλώσσα XML - eXtensible Markup Language.

1.5. ΛΕΙΤΟΥΡΓΙΕΣ ΣΥΣΤΗΜΑΤΟΣ ΙΧΝΗΛΑΣΙΜΟΤΗΤΑΣ

1.5.1. Απόσυρση – Ανάκληση Προϊόντων

Σύμφωνα με την ισχύουσα νομοθεσία υπάρχει σαφής τρόπος αντιμετώπισης του θέματος για την περίπτωση των μη ασφαλών τροφίμων και ζωοτροφών, δηλαδή αυτών που είναι επιβλαβή για την υγεία και αυτών που είναι ακατάλληλα για ανθρώπινη και ζωική κατανάλωση, ενώ χρειάζεται ρύθμιση για τις περιπτώσεις των μη κανονικών τροφίμων και ζωοτροφών. Βασική και ουσιαστική είναι η ευθύνη των επιχειρήσεων.

Στο σημείο αυτό θεωρούμε σκόπιμο να αναφέρουμε τους ορισμούς των εννοιών απόσυρση και ανάκληση.

Απόσυρση: Αποτελεί κάθε μέτρο με στόχο να εμποδιστεί η διανομή, η έκθεση και η προσφορά προϊόντος στους καταναλωτές (Οδηγία 2001/95/EK).

Ανάκληση: Αποτελεί κάθε μέτρο που αποβλέπει στην επιστροφή ενός επικίνδунου προϊόντος, το οποίο ο παραγωγός, ή ο διανομέας του έχει ήδη προμηθεύσει ή διαθέσει στους καταναλωτές (Οδηγία 2001/95).

Σημαντικό ρόλο στην διαδικασία της απόσυρσης - ανάκλησης ενός προϊόντος κατέχει ο **υπεύθυνος της επιχείρησης τροφίμων - ζωοτροφών**^{**}. Ο όρος υπεύθυνος επιχείρησης τροφίμων - ζωοτροφών αφορά όλα τα φυσικά ή νομικά πρόσωπα που έχουν την ευθύνη να εξασφαλίσουν ότι πληρούνται οι απαιτήσεις της νομοθεσίας για τα τρόφιμα - ζωοτροφές μέσα στην επιχείρηση που έχουν υπό τον έλεγχό τους (Κανονισμός 178/2002/EK).

Παρακάτω παραθέτουμε τις ενέργειες στις οποίες πρέπει να προβούν οι υπεύθυνοι μιας επιχείρησης τροφίμων - ζωοτροφών καθώς και οι αρμόδιες αρχές στις περιπτώσεις «μη ασφαλών» τροφίμων.

^{**} Ως επιχείρηση τροφίμων - ζωοτροφών θεωρείται κάθε επιχείρηση κερδοσκοπική και μη, δημόσια ή ιδιωτική, η οποία ασκεί οποιαδήποτε από τις δραστηριότητες που συνδέονται με οποιοδήποτε στάδιο της παραγωγής, μεταποίησης και διανομής τροφίμων - ζωοτροφών (Κανονισμός 178/2002/EK).

1.5.1.1. Υποχρεώσεις επιχειρήσεων περί μη ασφαλών τροφίμων – ζωοτροφών

1.5.1.1.1. Κοινοτική Νομοθεσία (Κανονισμός 178/2002/ΕΚ)

1.5.1.1.1.1. Επιχειρήσεις τροφίμων – ζωοτροφών

Επιχειρήσεις τροφίμων - ζωοτροφών που εισαγάγουν, παράγουν, μεταποιούν, παρασκευάζουν ή διανέμουν τρόφιμα - ζωοτροφές.

Ο υπεύθυνος μιας επιχείρησης τροφίμων - ζωοτροφών εφόσον κρίνει ή έχει λόγους να πιστεύει ότι ένα προϊόν που έχει εισαγάγει, παραγάγει, μεταποιήσει, παρασκευάσει ή διανείμει δε συμμορφώνεται με τις απαιτήσεις για την ασφάλεια των τροφίμων - ζωοτροφών έχει τις εξής υποχρεώσεις:

- ✓ Ξεκινά αμέσως διαδικασίες απόσυρσης / ανάκλησης, εφόσον το τρόφιμο - ζωοτροφή απομακρύνθηκε από τον άμεσο έλεγχο του υπευθύνου της επιχείρησης τροφίμων.
- ✓ Ενημερώνει και συνεργάζεται με τις αρχές.
- ✓ Ενημερώνει τους καταναλωτές με ακρίβεια και αποτελεσματικότητα για τους λόγους ανάκλησης, εφόσον το προϊόν ενδέχεται να έχει φθάσει στους καταναλωτές.

Στις περιπτώσεις που το προϊόν κρίνεται, ή υπάρχουν υπόνοιες ότι ενδέχεται να είναι επιβλαβές ο υπεύθυνος της επιχείρησης τροφίμων - ζωοτροφών οφείλει να:

- ✓ Ενημερώνει αμέσως τις αρμόδιες αρχές για:
 - ✗ Το γεγονός.
 - ✗ Τα μέτρα που λαμβάνει, προκειμένου να αποτρέψει τους κινδύνους για τον τελικό καταναλωτή.
- ✓ Μην εμποδίσει ούτε αποθαρρύνει οποιοδήποτε πρόσωπο να συνεργαστεί, σύμφωνα με την εθνική νομοθεσία και τη νομική πρακτική, με τις αρμόδιες αρχές όταν τούτο μπορεί να έχει ως αποτέλεσμα την αποφυγή, τη μείωση ή την εξάλειψη κινδύνου προερχόμενου από τρόφιμο - ζωοτροφή.
- ✓ Συνεργάζεται με τις αρμόδιες αρχές σχετικά με τα μέτρα που λαμβάνονται για την αποφυγή ή τη μείωση των κινδύνων που προκαλεί το προϊόν.

Επιχειρήσεις με δραστηριότητες λιανικού εμπορίου ή διανομής με τις οποίες δεν επηρεάζεται η συσκευασία, η επισήμανση, η ασφάλεια ή η ακεραιότητα των τροφίμων - ζωοτροφών.

Ο υπεύθυνος μιας επιχείρησης τροφίμων - ζωοτροφών, ο οποίος έχει την ευθύνη για δραστηριότητες λιανικού εμπορίου ή διανομής με τις οποίες δεν επηρεάζεται η συσκευασία, η επισήμανση, η ασφάλεια ή η ακεραιότητα των προϊόντων:

- ✓ Ξεκινά, εντός των ορίων των δραστηριοτήτων του, διαδικασίες για την απόσυρση από την αγορά του προϊόντος.
- ✓ Μεταδίδει τις σχετικές πληροφορίες που είναι αναγκαίες για την ανίχνευση ενός τροφίμου - ζωοτροφής.
- ✓ Συνεργάζεται με τους παραγωγούς, μεταποιητές, παρασκευαστές, ή / και τις αρμόδιες αρχές όσον αφορά τα μέτρα που αυτοί λαμβάνουν.

Οι παραπάνω ενέργειες από των υπεύθυνο των τροφίμων - ζωοτροφών έχουν ως στόχο:

- ✓ Την ακρίβεια των διαδικασιών απόσυρσης / ανάκλησης.
- ✓ Την παροχή κατάλληλων πληροφοριών στους καταναλωτές.
- ✓ Την αξιολόγηση του κινδύνου από τις αρμόδιες αρχές.

Γενικές υποχρεώσεις υπευθύνων επιχειρήσεων τροφίμων - ζωοτροφών

Επιβάλλεται οι υπεύθυνοι των επιχειρήσεων τροφίμων - ζωοτροφών να:

- ✓ Είναι σε θέση να προσδιορίζουν από ποιον και σε ποιον έγινε η προμήθεια ενός προϊόντος.
- ✓ Διαθέτουν συστήματα και διαδικασίες για την παροχή των σχετικών πληροφοριών στις αρμόδιες αρχές κατόπιν αίτησής τους (Άρθρο 18 σχετικά με την Ικνηλασιμότητα).

Για την εκπλήρωση των στόχων του Άρθρου 18 σχετικά με την Ικνηλασιμότητα κρίνεται απαραίτητη η καταχώρηση των ακολούθων στοιχείων. Τα στοιχεία αυτά μπορούν να ομαδοποιηθούν σε δυο κατηγορίες ανάλογα με το βαθμό προτεραιότητας τους:

- ✓ Α Κατηγορία. Περιλαμβάνει όλα τα στοιχεία που διαβιβάζονται αμέσως στις αρμόδιες αρχές, ανεξαρτήτου περίπτωσης.
 - ✗ Επωνυμία και διεύθυνση του προμηθευτή, είδος των προϊόντων που προμηθεύτηκε^{††}.

^{††} Όταν μια επιχείρηση τροφίμων προμηθεύεται επανειλημμένα τον ίδιο τύπο προϊόντων, η απαίτηση της Ικνηλασιμότητας δεν εκπληρώνεται με απλώς και μόνον την καταχώρηση της επωνυμίας του προμηθευτή και της φύσης των προϊόντων.

- ✗ Επωνυμία και διεύθυνση του πελάτη, είδος των προϊόντων που παραδόθηκαν στον πελάτη[‡].
- ✗ Ημερομηνία συναλλαγής / παράδοσης.
- ✓ Β Κατηγορία. Περιλαμβάνει στοιχεία, των οποίων η καταγραφή συνίσταται θερμά και διατίθενται μόλις αυτό είναι ευλόγως δυνατό, εντός δέουσας προθεσμίας ανάλογα με τις περιπτώσεις.
 - ✗ Όγκος ή ποσότητα προϊόντων.
 - ✗ Αριθμός παρτίδας, αν υπάρχει.
 - ✗ Λεπτομερέστερη περιγραφή του προϊόντος (προϊόντα προσυσκευασμένα ή χύδη, ακατέργαστα ή μεταποιημένα προϊόντα, κ.α.).

Ο υπεύθυνος μιας επιχείρησης μπορεί να λάβει γνώση ότι ένα προϊόν δεν πληροί τις προϋποθέσεις της ασφάλειας των τροφίμων - ζωοτροφών από:

- ✓ Αρμόδιες αρχές ελέγχου
 - ✗ Επιθεώρηση.
 - ✗ Δειγματοληψία & Ανάλυση^{§§}.
 - ✗ Alert.
 - ✗ Επιστημονικά Δεδομένα.
- ✓ Εσωτερική πληροφόρηση

1.5.1.1.1.2. Αρμόδιες αρχές ελέγχου

Τα κράτη μέλη:

- ✓ Εκτελούν τη νομοθεσία για τα προϊόντα.
- ✓ Παρακολουθούν και επαληθεύουν εάν τηρούνται οι σχετικές απαιτήσεις της νομοθεσίας αυτής από τους υπευθύνους των επιχειρήσεων τροφίμων - ζωοτροφών σε όλα τα στάδια παραγωγής, μεταποίησης και διανομής.

[‡] Στην περίπτωση των πελατών η απαιτήσεως της Ικνηλασιμότητας είναι ίδιες με αυτές που ισχύουν και στους προμηθευτές.

^{§§} Στις περιπτώσεις που έχει δικαίωμα άσκησης έφεσης εφαρμόζει τις οριζόμενες διαδικασίες από το αποτέλεσμα.

- ✓ Καθορίζουν το σύστημα των κυρώσεων που επιβάλλονται στις παραβιάσεις της νομοθεσίας για τα προϊόντα. Οι εν λόγω κυρώσεις πρέπει να είναι αποτελεσματικές, ανάλογες και αποτρεπτικές.

1.5.1.1.2. Εθνική Νομοθεσία (ΚΥΑ 121/2003 ΦΕΚ 685 Β)

1.5.1.1.2.1. Επιχειρήσεις τροφίμων – ζωοτροφών

Ο ιδιοκτήτης ή ο νόμιμος εκπρόσωπος της επιχείρησης τροφίμων - ζωοτροφών σε περιπτώσεις που έχει εισαγάγει, παράγει, μεταποιήσει, παρασκευάσει, διανείμει, διαθέσει ή διαθέτει προς πώληση προϊόν που δεν συμμορφώνεται με τις απαιτήσεις για την ασφάλεια των τροφίμων - ζωοτροφών υποχρεούται άμεσα:

- ✓ Σε απόσυρση ή ανάκληση από την αγορά των τροφίμων - ζωοτροφών. Τα αποσυρόμενα προϊόντα μεταφέρονται σε ανεξάρτητο από τα ασφαλή χώρο στον οποίο τοποθετείται πινακίδα με την ένδειξη «μη ασφαλή προϊόντα». Ο ιδιοκτήτης υποχρεούται να τηρεί βιβλίο στο οποίο καταχωρεί ημερολογιακά το είδος, την ποσότητα και στοιχεία που αφορούν τα προϊόντα που έχουν αποσυρθεί - ανακληθεί, έτσι ώστε να είναι σε θέση να ενημερώνει τις αρμόδιες αρχές ελέγχου.
- ✓ Σε έγγραφη ενημέρωση της αρμόδιας υπηρεσίας του ΕΦΕΤ, γνωστοποιώντας:
 - ✗ Το είδος, την ποσότητα, τα στοιχεία συσκευασίας και επισήμανσης των προϊόντων αυτών, καθώς και την παρτίδα των προϊόντων.
 - ✗ Τα λεπτομερή στοιχεία των επιχειρήσεων στις οποίες ενδεχομένως διατέθηκαν τα εν λόγω προϊόντα.
- ✓ Σε καταστροφή των προϊόντων που δεν έχουν κριθεί δεκτά βιομηχανοποίησης, παρουσία των αρμόδιων αρχών ελέγχου.

1.5.1.1.2.2. Αρμόδιες αρχές ελέγχου

Οι αρμόδιες αρχές ελέγχου στις περιπτώσεις που κρίνεται κατά τον τακτικό ή έκτακτο έλεγχο ή μετά από καταγγελίες πολιτών ή μετά από σήμα του συστήματος έγκαιρης προειδοποίησης για τα τρόφιμα - ζωοτροφές:

1. Κατάσχουν τα εν λόγω προϊόντα.

2. Διασφαλίζουν με την παρουσία τους την καταστροφή τους εφόσον δεν κριθούν δεκτά βιομηχανοποίησης.
3. Αποστέλλουν την υπόθεση στην αρμόδια εισαγγελία.

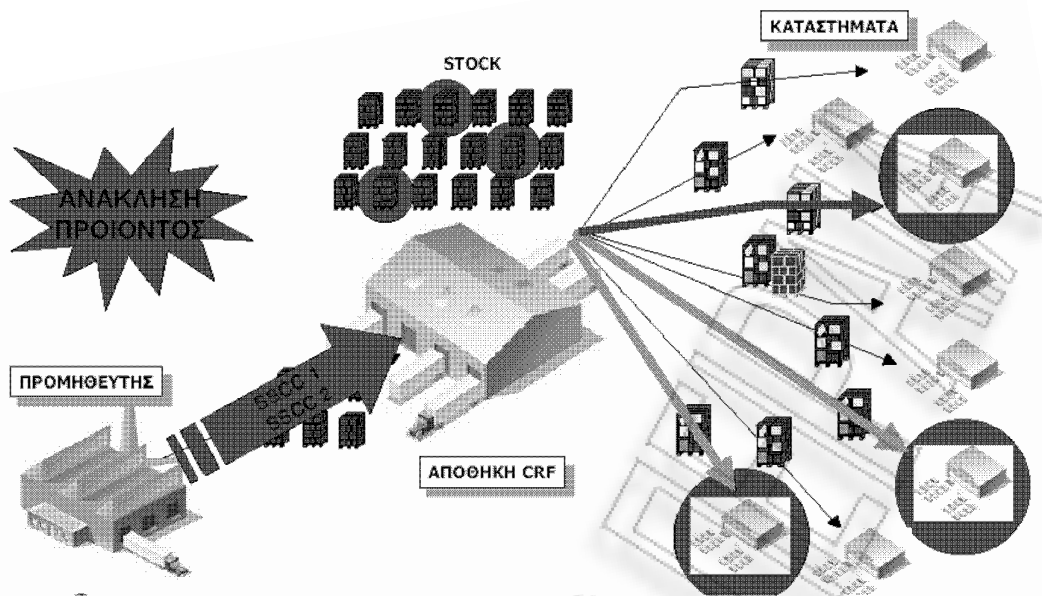
Γενικά έχει ιδιαίτερη σημασία το σύστημα Ικνηλασιμότητας της κάθε επιχείρησης τροφίμων - ζωοτροφών, να σχεδιάζεται έτσι ώστε να παρακολουθείται η φυσική ροή των προϊόντων από τη διαδικασία της παραγωγής μέχρι και την διανομή τους στους πελάτες. Γεγονός το οποίο καθιστά τη διαδικασία της ανάκλησης πιο έγκαιρη, ορθή και αποτελεσματική.

Στο σημείο αυτό ας εξετάσουμε την έννοια της παρτίδας, η οποία ορίζεται ως το σύνολο μιας ποσότητας ενός «υλικού», η οποία έχει παραχθεί ή συσκευαστεί κάτω από τις ίδιες συνθήκες. Σαν υλικό ορίζεται κάθε μορφή τελικού ή ημιτελούς προϊόντος, πρώτης ύλης ή υλικού συσκευασίας.

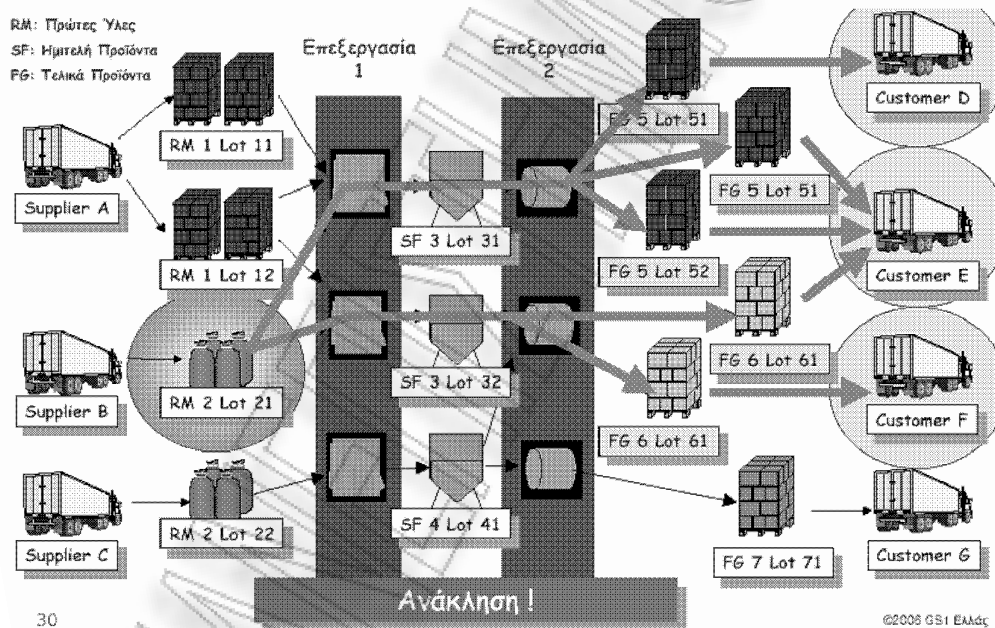
Η ανάκληση των προϊόντων βασίζεται εξολοκλήρου στους κωδικούς παρτίδας. Ο κωδικός παρτίδας, lot ή batch, είναι ένα σύνολο χαρακτήρων τα οποία αποτελούν την κωδικοποίηση της συγκεκριμένης παρτίδας και περιέχουν τις παρακάτω πληροφορίες: ημερομηνία παραγωγής - επεξεργασίας, τόπο παραγωγής - επεξεργασίας (χώρα ή και εργοστάσιο) και άλλες πληροφορίες, όπως π.χ. βάρδια.

Πρώτες ύλες □ Προϊόντα

Στις εικόνες που ακολουθούν παρουσιάζεται η διαδικασία της ανάκλησης προϊόντων με βάση τις πρώτες ύλες που έχουν χρησιμοποιηθεί για την παραγωγή τους. Η πρώτη εικόνα περιγράφει την διαδικασία ανάκλησης προϊόντων τα οποία η επιχείρηση τα προμηθεύεται, ως τελικά προϊόντα, και στη συνέχεια τα διανέμει στους πελάτες τις. Ενώ η δεύτερη εικόνα παρουσιάζει τη διαδικασία της ανάκλησης προϊόντων όπου η επιχείρηση προμηθεύεται πρώτες ύλες, παράγει ημιτελή προϊόντα και στη συνέχεια τελικά προϊόντα.



Σχήμα 53. Ανάκληση προϊόντων 1



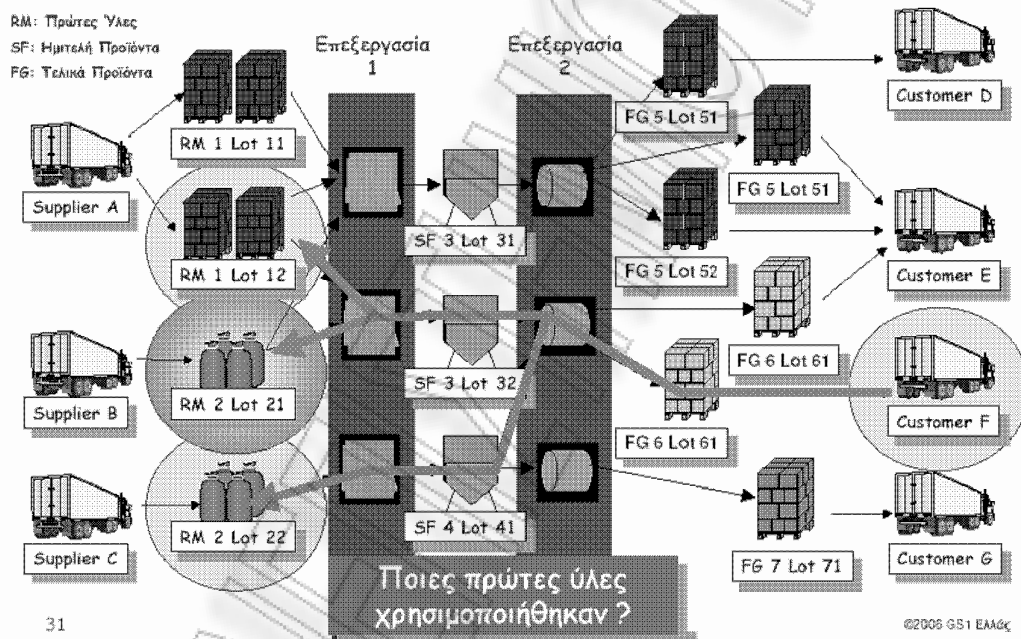
Σχήμα 54. Ανάκληση προϊόντων 2

Σχετικά με την παραπάνω εικόνα παρατηρούμε ότι κάθε πρώτη ύλη RM, ημιτελές προϊόν SF και τελικό προϊόν FG έχει ένα μοναδικό κωδικό παρτίδας lot. Με βάση τους κωδικούς αυτούς μπορούμε να εντοπίζουμε από κάθε πρώτη ύλη, ποια ημιτελή και τελικά προϊόντα έχουν παραχθεί. Γεγονός το οποίο επιτρέπει στον υπεύθυνο κάθε επιχείρησης τροφίμων - ζωοτροφών, όταν ενημερωθεί από κάποιον προμηθευτή ότι υπάρχει πρόβλημα με τις πρώτες ύλες που του προμήθευσε, να μπορεί να ανακαλεί όλα τα τελικά προϊόντα που

έχουν παραχθεί από την συγκεκριμένη πρώτη ύλη. Δηλαδή, όπως διαφαίνεται και στην παραπάνω εικόνα, όταν ο προμηθευτής B μας ενημερώσει για κάποιο πρόβλημα στην πρώτη ύλη RM 2 γνωρίζουμε ότι η συγκεκριμένη πρώτη ύλη έχει χρησιμοποιηθεί για την παραγωγή των ημιτελών προϊόντων SF 3 lot 31 και SF 3 lot 32, τα οποία χρησιμοποιήθηκαν στην παραγωγή των τελικών προϊόντων FG 5 lot 51, FG 5 lot 52 και FG 6 lot 61. Έτσι μπορούμε άμεσα και ορθά να ενημερώσουμε τους πελάτες D, E και F για την πρόθεση μας να ανακαλέσουμε τα συγκεκριμένα τελικά προϊόντα.

Προϊόντα ▫ Πρώτες ύλες

Μια εξίσου σημαντική ικανότητα που μας παρέχει ένα σύστημα διαχείρισης παρτίδων - ικνηλασιμότητας είναι η ανάκληση προϊόντων όταν ενημερωθούμε για ελαττωματικά προϊόντα από κάποιον πελάτη μας. Η παραπάνω ικανότητα περιγράφεται αναλυτικά στην εικόνα που ακολουθεί.



Σχήμα 55. Ανάκληση με διαχείριση παρτίδων

Στην συγκεκριμένη εικόνα παρατηρούμε ότι ο πελάτης F μας ενημερώνει για κάποιο πρόβλημα στα τελικά προϊόντα FG 6 Lot 61. Έχοντας καταγεγραμμένους τους κωδικούς των ημιτελών προϊόντων τα οποία χρησιμοποιήθηκαν για την παραγωγή των προϊόντων FG 6 Lot 61, μπορούμε γρήγορα και ορθά να αντλήσουμε στοιχεία για το ποιες πρώτες ύλες χρησιμοποιήθηκαν (περιγραφή, κωδικός παρτίδας). Οπότε στην περίπτωση που το πρόβλημα οφείλετε σε κάποια από αυτές τις πρώτες ύλες έχουμε τη δυνατότητα να ενημερώσουμε όλους τους άλλους πελάτες μας οι οποίοι έχουν προμηθευτεί τελικά προϊόντα κατασκευασμένα από την ελαττωματική πρώτη ύλη. Δηλαδή στην προκειμένη

περίπτωση, αν τελικά διαπιστώναμε πρόβλημα στην πρώτη ύλη RM 2 Lot 22 θα έπρεπε να ενημερώσουμε και τον πελάτη F για πιθανό πρόβλημα στα τελικά προϊόντα FG 7 Lot 71.

1.5.2. Διαχείριση αποθηκών

Ικνηλασιμότητα των προϊόντων σημαίνει έγκυρη και έγκαιρη πληροφόρηση για την πορεία των προϊόντων στην εφοδιαστική αλυσίδα. Συνεπώς σημαίνει πιο στοχευμένες, ως προς τις πραγματικές ανάγκες της αγοράς, παραγγελίες και συνεπώς μικρότερα αποθέματα που αποφέρουν μείωση κόστους αποθήκευσης, δεσμευμένου κεφαλαίου και μεταφορών. Επίσης τα συστήματα διαχείρισης παρτίδων - Ικνηλασιμότητας, χρησιμοποιώντας την τεχνολογία RFID, συμβάλλουν στην αποδοτικότερη διαχείριση αποθηκών καθώς προσφέρουν την δυνατότητα χωρικού και ποσοτικού εντοπισμού των προϊόντων μέσα στην αποθήκη με ένα απλό σκανάρισμα με την χρήση κατάλληλου RFID αναγνώστη.

1.5.3. Ποιοτικός έλεγχος προϊόντων και πληροφόρηση καταναλωτή

Σε ένα barcode ή σε μια ετικέτα RFID που φέρει ένα προϊόν καταγράφονται - αποθηκεύονται πληροφορίες για το ίδιο το προϊόν. Για παράδειγμα είναι δυνατόν να γνωρίζει ο καταναλωτής την όλη πορεία ενός πουλερικού από την μέρα γέννησης του μέχρι και την σφαγή του (που γεννήθηκε, που και πως έγινε η εκτροφή του που και πότε έγινε η σφαγή του). Επίσης δίνεται η δυνατότητα για περαιτέρω παρακολούθηση των προϊόντων ως προς τις συνθήκες μεταφοράς και αποθήκευσης (αισθητήρες θερμοκρασίας ενσωματωμένους στις ετικέτες RFID).

1.6. ΕΦΑΡΜΟΓΗ ΣΥΣΤΗΜΑΤΟΣ ΙΧΝΗΛΑΣΙΜΟΤΗΤΑΣ

Για την επιλογή και εφαρμογή του κατάλληλου συστήματος ιχνηλασιμότητας απαιτείται η εξέταση ορισμένων παραμέτρων οι οποίοι αφορούν τόσο την ίδια την εταιρία όσο και τους υπόλοιπους εμπορικούς εταίρους της εφοδιαστικής αλυσίδας. Παρακάτω παρουσιάζουμε την όλη διαδικασία εφαρμογής ενός συστήματος Ιχνηλασιμότητας ομαδοποιημένη σε έξι βήματα.

Βήμα 1^ο: Προετοιμασία

1. Ποιος είναι υπεύθυνος για τη λειτουργία του συστήματος.
2. Ποια τμήματα της εταιρίας εμπλέκονται άμεσα.
3. Ποια είναι η προϋπολογιζόμενη επένδυση για αυτό το σύστημα.
4. Ποιος είναι ο χρόνος εφαρμογής ενός τέτοιου project.
5. Εάν υπάρχουν ήδη εγκατεστημένα συστήματα ιχνηλασιμότητας με τους εμπορικούς εταίρους της εφοδιαστικής αλυσίδας.

Βήμα 2^ο: Ανάλυση απαιτήσεων

1. Ποιος είναι ο σκοπός του συστήματος.
2. Κάλυψη της ασφάλειας των τροφίμων ή απλά των απαιτήσεων των πελατών.
3. Κάλυψη όλων των κατηγοριών προϊόντων ή μέρος αυτών.

Βήμα 3^ο: Τεχνική ανάλυση

1. Υπάρχουν ήδη συστήματα καταχώρησης ή ελέγχου ποιότητας τα οποία θα πρέπει να ληφθούν υπόψη.
2. Τι επίδραση θα έχουν τα νέα συστήματα στα τρέχοντα συστήματα ή στις εταιρικές διαδικασίες.
3. Θα είναι το νέο σύστημα συμβατό με αυτά των υπολοίπων εμπορικών εταίρων.

Βήμα 4^ο: Επιλογή

1. Έρευνα αγοράς.
2. Ποιο είναι το καλύτερο σύστημα που ικανοποιεί τις ανάγκες της εταιρίας.

Βήμα 5^ο: Εφαρμογή

1. Κωδικοποίηση και ανάγνωση κωδικών Μονάδων Εμπορίας-Logistics-Θέσεων.
2. Ηλεκτρονική επεξεργασία και ανταλλαγή δεδομένων.
3. Καταμερισμός εργασιών και καθηκόντων.

Βήμα 6^ο: Εσωτερικός έλεγχος και δοκιμή του συστήματος - Δοκιμή της διαδικασίας ανάκλησης

1. Επαλήθευση των συμφωνημένων διαδικασιών και αξιολόγησή τους.
2. Δοκιμή του συστήματος σε σχέση με τις διαδικασίες ανάκλησης.
3. Δοκιμή όχι μόνο στα πλαίσια της εταιρίας αλλά και εξωτερικά (μεταξύ των εμπορικών εταίρων).

1.7. ΕΠΟΜΕΝΑ ΒΗΜΑΤΑ ΣΤΗΝ ΙΧΝΗΛΑΣΙΜΟΤΗΤΑ

1.7.1. EPC και EPCglobal Network

Η EPCglobal είναι μια ένωση που διοικείται από αντιπρόσωπους από διάφορους χώρους (αναφέρονται παρακάτω) και αναπτύσσει πρότυπα που στοχεύουν στην παροχή κατάλληλης τεχνολογίας για την αύξηση της αποτελεσματικότητας και την μείωση των λαθών στην λειτουργία της εφοδιαστικής αλυσίδας. Ενδεικτικά οι μετέχοντες στην EPCglobal είναι:

- Οργανισμοί Εμπορίου: UCC, EAN
- Προμηθευτές προϊόντων: Gillette, Johnson & Johnson, Procter & Gamble
- Λιανέμποροι: Wal-Mart, Metro AG
- Κυβέρνηση: Υπουργείο Αμύνης ΗΠΑ (*US Department of Defence*)
- Τεχνολογία: Hewlett-Packard, Cisco Systems
- Ακαδημαϊκός χώρος: Ινστιτούτο Τεχνολογίας Μασαχουσέτης (*MIT*)

Η EPCglobal πιστεύει ότι θα επιτύχει τους στόχους της με την αυτοματοποίηση του εντοπισμού προϊόντων μέσω της τεχνολογίας RFID και συγκεκριμένα με την χρήση φθηνών RFID ετικετών και τον ορισμό ενός παγκόσμιο πλαισίου ανταλλαγής πληροφοριών. Για το λόγο αυτό έχει αναπτύξει το EPCglobal Network, ένα καταναμημένο δίκτυο υπηρεσιών, και έχει ορίσει έξι κλάσεις RFID ετικετών με αύξουσα λειτουργικότητα.

Το EPCglobal Network είναι ένα δίκτυο που καθιστά δυνατή την άμεση, μονοσήμαντη και αυτόματη αναγνώριση τεμαχίων στην εφοδιαστική αλυσίδα και τον διαμοιρασμό των δεδομένων τους. Στόχος του είναι η «πραγματική» ορατότητα (visibility) της εφοδιαστικής αλυσίδας, με την παροχή αναγνώρισης οποιουδήποτε τεμαχίου (κωδικός και Serial Number), οποιασδήποτε εταιρίας, οποιασδήποτε βιομηχανίας, οπουδήποτε στον κόσμο με σκοπό να κάνει τις εταιρίες περισσότερο αποτελεσματικές. Το EPCglobal Network αποτελείται από πέντε βασικά στοιχεία:

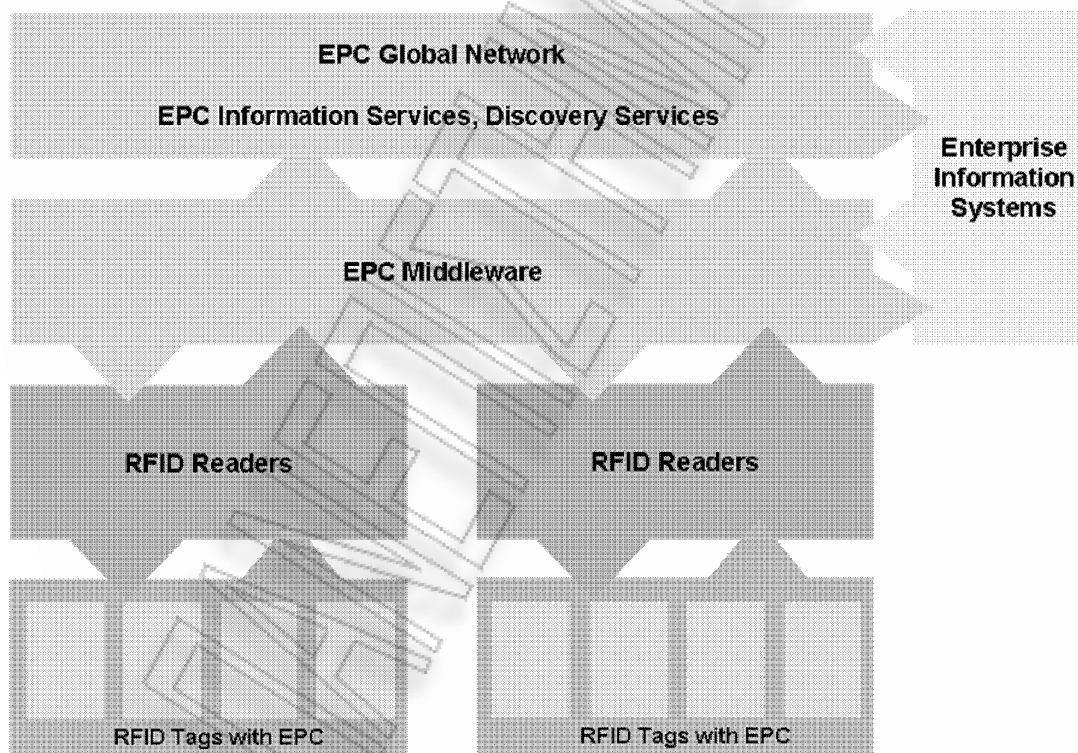
1. Ηλεκτρονικός Κωδικός Προϊόντος (*EPC, Electronic Product Code*): Ο EPC είναι ένας μοναδικός αριθμός ταυτοποίησης προϊόντος σε επίπεδο τεμαχίου που αποτελείται από 64 - 256 bits.
2. Σύστημα Αναγνώρισης (*ID System*): Το Σύστημα Αναγνώρισης (*ID System*) αποτελείται από RFID αναγνώστες και ετικέτες. Οι RFID ετικέτες είναι παθητικές και περιέχουν μόνο τον κωδικό EPC του αντικειμένου στο οποίο επικολλούνται. Οι RFID αναγνώστες διαβάζουν το

EPC και το στέλνουν στα τοπικά πληροφοριακά συστήματα της επιχείρησης μέσω του EPC λογισμικού (*EPC Middleware*).

3. Λογισμικό EPC (*EPC Middleware*): Το λογισμικό EPC (*EPC Middleware*) διαχειρίζεται γεγονότα ανάγνωσης πραγματικού χρόνου και αναλαμβάνει να επικοινωνήσει τις πληροφορίες που δέχεται στις Υπηρεσίες Πληροφοριών EPC και στα τοπικά πληροφοριακά συστήματα της επιχείρησης. Η EPCglobal αναπτύσσει μια πρότυπη διεπαφή εφαρμογής για υπηρεσίες, επιτρέποντας την ανταλλαγή δεδομένων μεταξύ αναγνώστων EPC και πληροφοριακών συστημάτων.

4. Υπηρεσίες Πληροφοριών EPC (*EPCIS*, *EPC Information Services*): Οι Υπηρεσίες Πληροφοριών EPC (*EPCIS*, *EPC Information Services*) επιτρέπουν σε χρήστες την ανταλλαγή EPC δεδομένων με εμπορικούς συνεργάτες μέσω του EPCglobal Network.

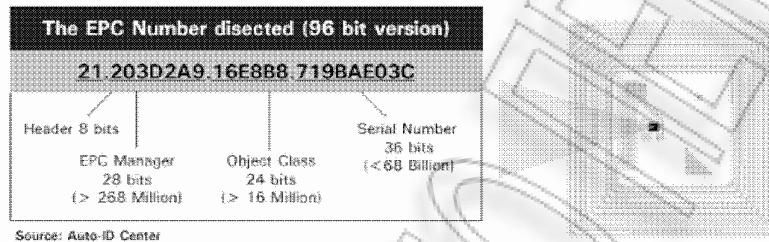
5. Υπηρεσίες Ανακάλυψης (*Discovery Services*): Οι υπηρεσίες Ανακάλυψης (*Discovery Services*) είναι ένα σετ υπηρεσιών που επιτρέπουν στους χρήστες να αναζητήσουν παγκοσμίως, δεδομένα σχετικά με ένα συγκεκριμένο κωδικό EPC και αποκτήσουν πρόσβαση σε αυτά. Μία από τις υπηρεσίες ανακάλυψης είναι η Υπηρεσία Ονοματοδοσίας Αντικειμένων (*ONS*, *Object Naming Service*).



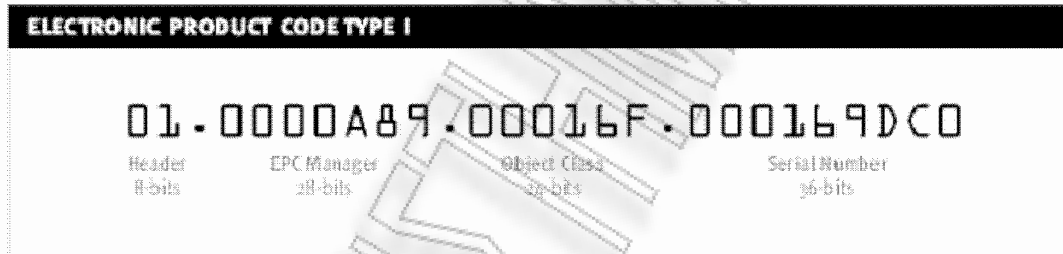
Σχήμα 56. RFID and EPC

1.7.1.1. Ηλεκτρονικός Κωδικός Προϊόντος

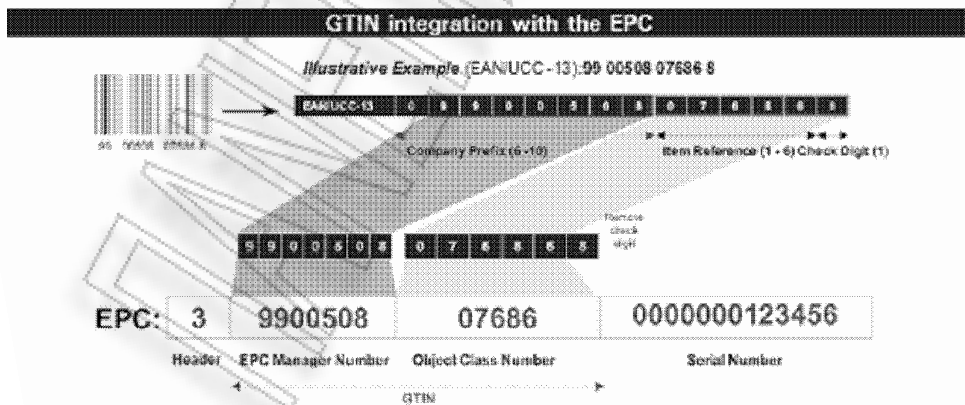
Ο Ηλεκτρονικός Κώδικας Προϊόντος (*EPC, Electronic Product Code*) χρησιμοποιείται σε συνδυασμό με την τεχνολογία RFID προκειμένου να βελτιώσει κυρίως την αποτελεσματική διαχείριση της εφοδιαστικής αλυσίδας και να μειώσει τα λειτουργικά κόστη. Ο EPC είναι αποτέλεσμα ενός παγκόσμιου εγχειρήματος προκειμένου να επιτευχθεί καλύτερη συνεννόηση μεταξύ των μελών της εφοδιαστικής αλυσίδας. Αυτός ο κώδικας παρέχει γρήγορες και λεπτομερείς πληροφορίες για ένα προϊόν σε οποιοδήποτε σημείο της εφοδιαστικής αλυσίδας. Ο EPC είναι παρόμοιος του Παγκόσμιου Κώδικα Προϊόντος (*UPC, Universal Product Code*), ο οποίος χρησιμοποιείται στους γραμμωτούς κωδικούς.



Σχήμα 57. EPC Number



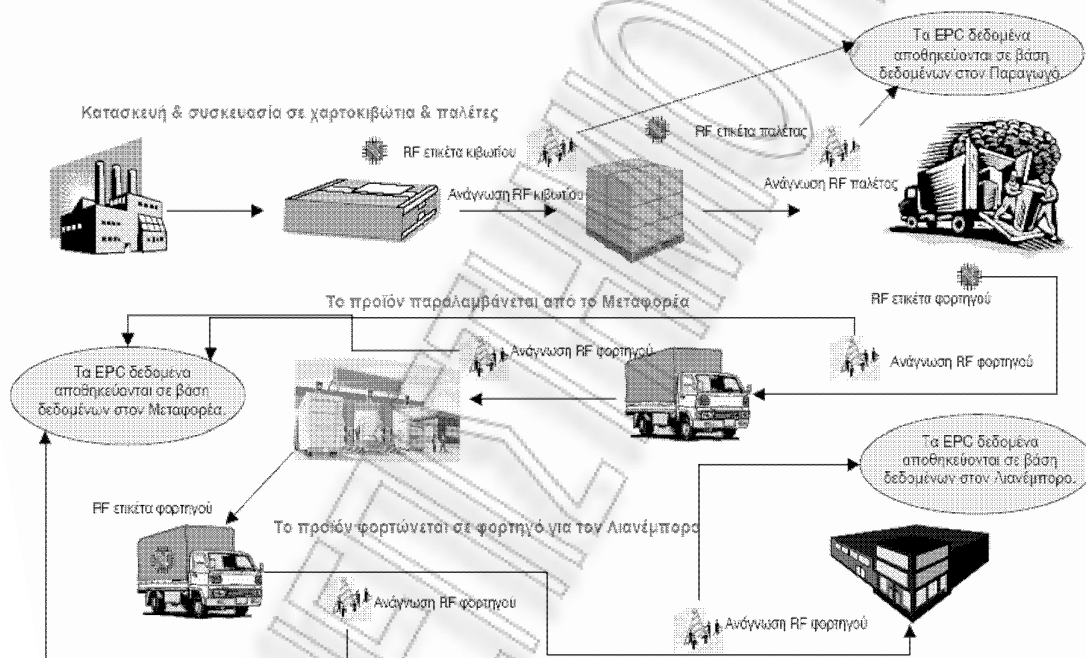
Σχήμα 58. EPC Number 2



Σχήμα 59. GTIN integration with the EPC

Ο EPC είναι ένας μοναδικός αριθμός αποτελούμενος από 64 - 256 bits και περιλαμβάνει τέσσερα διακριτά πεδία:

- ✓ Επικεφαλίδα (*Header*): Η επικεφαλίδα αποτελείται από 8-bits και προσδιορίζει το μήκος του Ηλεκτρονικού Κωδικού Προϊόντος
- ✓ Διαχειριστής Ηλεκτρονικού Κωδικού Προϊόντος (*EPC manager*): Προσδιορίζει τον κατασκευαστή του προϊόντος
- ✓ Κλάση του αντικειμένου (*Object Class*): Αναφέρεται στον ακριβή τύπο του αντικειμένου, με τον ίδιο τρόπο όπως η Μονάδα Διατήρησης Αποθέματος SKU (Stock Keeping Unit)
- ✓ Σειριακός Αριθμός (*Serial Number*): Πρόκειται για το συγκεκριμένο σειριακό αριθμό που προσδιορίζει το αντικείμενο



Σχήμα 60. Λειτουργία εφοδιαστικής αλυσίδας με RF και EPC

1.8. ΣΥΜΠΕΡΑΣΜΑΤΑ

Μετά από αυτή την αναλυτική περιγραφή όλων των θεμάτων που αφορούν την ικνηλασιμότητα, αλλά και των διαδικασιών και τεχνολογιών που την περιστοιχίζουν μπορούμε να καταλήξουμε σε κάποια συμπεράσματα.

Πρώτα από όλα είναι εμφανές πως η ικνηλασιμότητα των προϊόντων είναι μια απαραίτητη διαδικασία των επιχειρήσεων των τροφίμων και ποτών. Αυτό συμβαίνει τόσο από την ανάγκη της παρακολούθησης της πορείας των προϊόντων μέσα στην εφοδιαστική αλυσίδα από την πλευρά της επιχείρησης, όσο και από τις απαιτήσεις των νόμων και ευρωπαϊκών κανονισμών.

Από την άλλη πλευρά η έλλειψη ολοκληρωμένης και συγκεντρωτικής βιβλιογραφίας καθιστά δύσκολη την μελέτη και την εμβάθυνση στο αντικείμενο, για αυτό και το πρώτο μέρος της εργασίας αφορά μόνο την συγκέντρωση όλων των πληροφοριών που αφορούν την θεωρητική προσέγγιση της ικνηλασιμότητας. Θεωρούμε πως είναι αναγκαία η σύνταξη ενός εγγράφου που θα περιλαμβάνει όλα τα θέματα που εμπλέκονται με το αντικείμενο της ικνηλασιμότητας.

Μπορούμε να επισημάνουμε πως από την ανάλυση και εμβάθυνση στο αντικείμενο της ικνηλασιμότητας προκύπτει πως υπάρχουν συγκεκριμένες αρχές που την διέπουν, οι οποίες πρέπει να λαμβάνονται υπόψη στην δημιουργία διαδικασιών και να τηρούνται στην εκτέλεση αυτών. Ακόμη, έχουν αναπτυχθεί τεχνολογίες για την υποστήριξη της διαδικασίας της ικνηλασιμότητας - barcodes κ.α. - , ενώ ταυτόχρονα αναπτύσσονται και νεότερες - rfid, EPC κ.α.. Τέλος, δημιουργούνται ολοκληρωμένα συστήματα που παρακολουθούν την διαδικασία της ικνηλασιμότητας στις επιχειρήσεις.

Η ανάγκη για σύστημα που παρακολουθεί την ικνηλασιμότητα των προϊόντων, πρώτων υλών κ.α. μπορεί να θεωρηθεί βέβαιη αφού όλες οι διαδικασίες μιας επιχείρησης σήμερα παρακολουθούνται μέσω συστημάτων υπολογιστών. Ένα τέτοιο σύστημα θα προσπαθήσουμε να περιγράψουμε, να σχεδιάσουμε και υλοποιήσουμε στο επόμενο τμήμα της εργασίας εξετάζοντας το με βάση τις απαιτήσεις του πελάτη - επιχείρησης.

Μέρος 2ο: ***

**Πρακτική εφαρμογή στα πλαίσια ανάλυσης,
σχεδίασης και μερικής υλοποίησης ενός λογισμικού
ιχνηλασιμότητας**

*** Η δομή του δεύτερου μέρους της εργασίας ακολουθεί το πρότυπο IEEE (830-1993) που χρησιμοποιείται για τεκμηρίωση λογισμικού. Για το λόγο αυτό, συμπεράσματα, σκέψεις ή λοιπά σχόλια περιλαμβάνονται στην υπάρχουσα δομή.

2.1. ΟΡΙΣΜΟΣ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

2.1.1. Παρουσίαση του προβλήματος

☞ Γενικά

Ως πρόβλημα που οδηγεί στην σχεδίαση του συγκεκριμένου λογισμικού μπορούμε να ορίσουμε την ανάγκη για παρακολούθηση, μέσα από όλες τις φάσεις της παραγωγής, της μεταποίησης και της διανομής, τη διαδρομή ενός τροφίμου, μιας ζωοτροφής ή ενός ζώο από το οποίο εξάγονται βρώσιμες ουσίες ή ουσίες που πρόκειται ή αναμένεται να ενσωματωθούν σε τρόφιμο που προορίζεται για άνθρωπο ή ζώο. Η παραπάνω αυτή διαδρομή ενός τροφίμου μπορεί να είναι αμφίδρομη, δηλαδή αφορά τόσο την παραγωγή και διανομή του όσο και την ανάκληση του.

☞ Τι ζητάει ο «πελάτης»

Το λογισμικό θα παρέχει τη δυνατότητα καταγραφής όλων των απαιτούμενων στοιχείων πρώτων υλών, προϊόντων και πελατών καθώς και τη σύνδεση των παραπάνω με βάση τη δημιουργία μοναδικής ταυτότητας (lot number) του καθενός εξ αυτών. Επιπλέον, θα πρέπει ο χρήστης να έχει την δυνατότητα να αναζητά στοιχεία ή καταχωρίσεις οποιονδήποτε σχέσεων - για παράδειγμα [προϊόντα ανά πελάτη], [πρώτες ύλες ανά προϊόν], [πρώτες ύλες ανά πελάτη] - ώστε να μπορεί να προβεί στις ανάλογες ενέργειες - π.χ. ανάκληση προϊόντων. Επίσης, το λογισμικό θα πρέπει να περιέχει όλα τα συστατικά ενός σύγχρονου λογισμικού που αφορούν θέματα ασφάλειας, διαχείρισης χρηστών, επιλογής γλώσσας, διασύνδεσης με άλλα συστήματα κ.λ.π.

☞ Σπουδαιότητα

Οφείλουμε να επισημάνουμε την σπουδαιότητα του λογισμικού διότι πρόκειται για λογισμικό που αφορά την ασφάλεια των τροφίμων και κατ' επέκταση των καταναλωτών τους. Το λογισμικό πρέπει να είναι σε θέση να δώσει γρήγορα και με ακρίβεια όλες τις απαραίτητες πληροφορίες που χρειάζεται ο χρήστης για να αντιμετωπίσει έγκαιρα και σωστά καταστάσεις που θέτουν σε κίνδυνο την ασφάλεια των καταναλωτών. Η ανάγκη αυτή διατυπώνεται καθαρά και από τους κανονισμούς και οδηγίες της Ευρωπαϊκής Ένωσης (οδηγία 2001/CE, κανονισμός 178/2002) που αφορούν την ασφάλεια των τροφίμων και των ζωοτροφών, οι οποίες άρχισαν να ισχύουν από την 1^η Ιανουαρίου 2005.

2.1.2. Περιληπτική σκιαγράφηση της λύσης

☞ Μοντέλο Κύκλου Ζωής

Για την παραγωγή του λογισμικού θα ακολουθήσουμε το μοντέλο της πρωτοτυποποίησης διότι δίνει την δυνατότητα να δοκιμάζονται τα τμήματα που έχουν

υλοποιηθεί και επομένως να συνεχίζεται η υλοποίηση στα επόμενα, αν συμφωνεί ο «πελάτης», ή να διορθώνονται, αν χρειάζεται. Ιδιαίτερα σημαντικό σε μία τέτοια εφαρμογή που δεν επιτρέπει λάθη. Επιπλέον, το μοντέλο αυτό ενδείκνυται σε εφαρμογές σχετικά μεγάλου μεγέθους όπως αυτή.

▣ Συλλογή Απαιτήσεων

Η συλλογή των απαιτήσεων βασίζεται πρώτα στους κανονισμούς και τις οδηγίες της Ευρωπαϊκής Ένωσης και κατόπιν σε συνεντεύξεις με εταιρίες που χρησιμοποιούν ή θα χρησιμοποιήσουν λογισμικό Ικνηλασιμότητας. Τέλος, απαιτήσεις θα συλλεχθούν και κατά την δημιουργία του μοντέλου παράστασης του λογισμικού.

▣ Επιβεβαίωση Απαιτήσεων

Αφού πραγματοποιηθούν τα παραπάνω θα γίνει η επιβεβαίωση των απαιτήσεων παρουσιάζοντας στον «πελάτη» έγγραφα με την αναλυτική παρουσίαση του λογισμικού. Αν ο «πελάτης» ικανοποιηθεί, και εφόσον ικανοποιούνται όλοι οι κανονισμοί και οι οδηγίες της Ε.Ε., θα ξεκινήσει η κατασκευή της εφαρμογής, αν όχι θα ξαναγίνει μελέτη του προβλήματος.

Το λογισμικό θεωρούμε ότι θα ικανοποιεί όλες τις απαιτήσεις του «πελάτη» και θα λειτουργεί σε περιβάλλον Microsoft Windows, ενώ παράλληλα θα συνεργάζεται με όλα τα δημοφιλή ERP συστήματα και εμπορικά πακέτα.

2.2. ΕΓΓΡΑΦΟ ΠΡΟΔΙΑΓΡΑΦΩΝ ΤΩΝ ΑΠΑΙΤΗΣΕΩΝ ΑΠΟ ΤΟ ΛΟΓΙΣΜΙΚΟ

2.2.1. Εισαγωγή

2.2.1.1. Ταυτότητα του εγγράφου

Πρόκειται για ένα από τα έγγραφα τεκμηρίωσης του λογισμικού, στο οποίο καταγράφονται οι απαιτήσεις του κατασκευαστή και του «πελάτη» σύμφωνα με το διεθνές πρότυπο του IEEE (830-1993).

2.2.1.2. Σκοπός

Το παρών έγγραφο απευθύνεται στον «πελάτη» για την λεπτομερή περιγραφή, ταξινόμηση και ιεράρχηση των απαιτήσεων τόσο του «πελάτη» όσο και του κατασκευαστή. Καθοριστικό στην πορεία του έργου είναι το έγγραφο αυτό διότι ελλείψεις και αστοχίες όσων αναφέρονται σε αυτό θα μεταφερθούν σε όλη την υπόλοιπη διαδικασία κατασκευής του λογισμικού και ασφαλώς στο τελικό προϊόν.

2.2.1.3. Εμβέλεια

Το έγγραφο αναφέρεται σε όλες τις γενικές λειτουργίες, απαιτήσεις περιορισμών και συντήρησης, καθώς και συμπεριφορές του λογισμικού σε ανεπιθύμητες καταστάσεις. Δεν περιγράφει τις λειτουργίες που εκτελούνται στο παρασκήνιο, αλλά περιέχει όλα τα στοιχεία που το συνδέουν με το εξωτερικό περιβάλλον και τη χρήση αυτού.

2.2.1.4. Ορισμοί, ακρωνύμια, συντομογραφίες

E.E. : Ευρωπαϊκή Ένωση.

«πελάτης»: Με τον ορισμό «πελάτης» αναφερόμαστε σε κάποιον υποθετικό πελάτη του λογισμικού και ταυτόχρονα στους συγγραφείς του παρόντος κειμένου.

2.2.1.5. Πηγές αναφορών

Κύρια πηγή πληροφοριών αποτελούν οι κανονισμοί και οι οδηγίες της E.E. καθώς και η ομάδα του «πελάτη» με την οποία συνεργαζόμαστε. Μας παρέχουν πληροφορίες τόσο για τον τρόπο παρουσίασης των αποτελεσμάτων από την εφαρμογή, της δομής οργάνωσης των στοιχείων (χαρακτηριστικά προϊόντων - πελατών, φόρμες και πίνακες εμφάνισης κ.λ.π.), όσο και τις μεθόδους με τις οποίες θα παράγονται τα αποτελέσματα (βέλτιστες ημέρες παραγγελίας, υπολογισμοί ημερομηνιών λήξης

κ.α.). Αυτά επιτυγχάνονται μέσω εκθέσεων αναφορών από τον «πελάτη», καθώς και από συζητήσεις - συνεντεύξεις με ομάδα του «πελάτη».

Επιπλέον χρησιμοποιήθηκε και η εξής βιβλιογραφία:

- ✓ Ian Smith, Anthony Furness, 2006, “Improving traceability in food processing and distribution”, CRC publisher.
- ✓ L. A. Rabade, J. A. Alfaro, 2005, “Journal of Purchasing and Supply Management”, Elsevier publisher.
- ✓ Michele Lees, 2003, “Food Authenticity and Traceability”, 1 edition CRC publisher.
- ✓ Mike Dillon, 2007, “Traceability in Food”, Blackwell Publishers.

2.2.1.6. Περίληψη

Το έγγραφο αυτό περιγράφει κυρίως τις γενικές λειτουργίες που θα είναι σε θέση να πραγματοποιεί η εφαρμογή. Επιπλέον, καθορίζει τις εφαρμογές που θα πρέπει να συνεργάζονται με αυτό και συγκεκριμενοποιεί τις απαιτήσεις σε υπολογιστική ισχύ, αξιοπιστία και επιδόσεις. Αναφέρεται στη συμπεριφορά του λογισμικού προς το εξωτερικό του περιβάλλον - χρήστες κ.λ.π. - και όχι στα εσωτερικά του στοιχεία. Τέλος, προετοιμάζει την κατασκευή των μοντέλων παράστασης του λογισμικού και του εγγράφου σχεδίασης του λογισμικού.

2.2.2. Γενική περιγραφή του λογισμικού

2.2.2.1. Στίγμα

Το λογισμικό αυτό απευθύνεται κυρίως σε παραγωγικές επιχειρήσεις τροφίμων, ποτών και ζωοτροφών, οι οποίες οφείλουν να παρακολουθούν τις παραγωγικές τους διαδικασίες καθώς και την πορεία των προϊόντων τους μέσα στην τροφική αλυσίδα. Όσον αφορά τα ανταγωνιστικά προϊόντα, αξίζει να αναφέρουμε ότι υπάρχουν αρκετά, όπως:

- *GL1 Trace* από την εταιρία G - Logic.
- «*Ιχνηλάτης*» από την εταιρία Sinodos.com.
- *TracePro* από την εταιρία Acmon Systems A.E.
- *ItemTracker* από την εταιρία Agrosoft.

Από την άλλη πλευρά, έχει δοθεί ιδιαίτερη έμφαση ώστε το λογισμικό αυτό να καλύπτει τα αδύνατα σημεία των υπολοίπων.

2.2.2.2. Προοπτική

Υπάρχει άμεσα εμπορική προοπτική. Το λογισμικό κατασκευάζεται για την πώληση του σε εταιρίες παραγωγής νωπών και κατεψυγμένων τροφίμων, μονάδες εκτροφής ζώων, σφαγεία, φυραματοποιίες, εργαστήρια τυποποίησης τροφίμων, αλλαντοποιίες, συσκευαστήρια λαχανικών και σε όλους τους εμπλεκόμενους φορείς στο χώρο των τροφίμων, ποτών και ζωοτροφών. Επίσης, τη συγκεκριμένη έκδοση του λογισμικού πρόκειται να τη διαδεχθούν νέες, αναβαθμισμένες και πληρέστερες εκδόσεις.

2.2.2.3. Γενικές Λειτουργίες του λογισμικού

- ✓ Καταγράφει τις πρώτες ύλες που χρησιμοποιεί η επιχείρηση με τα χαρακτηριστικά τους.
- ✓ Καταγράφει τα προϊόντα που παράγει η επιχείρηση με τα χαρακτηριστικά τους.
- ✓ Καταγράφει τους πελάτες της επιχείρησης με όλα τα στοιχεία τους.
- ✓ Παρακολουθεί και ενημερώνει τα αποθέματα των πρώτων υλών.
- ✓ Παρακολουθεί και ενημερώνει τα αποθέματα των προϊόντων.
- ✓ Καταγράφει και παρακολουθεί την παραγωγική διαδικασία.
- ✓ Διαχειρίζεται τις παραγγελίες των πελατών.
- ✓ Παρέχει τη δυνατότητα ανάκλησης παρτίδων πρώτων υλών ή προϊόντων.
- ✓ Παρέχει τη δυνατότητα παρακολούθησης της πορείας μιας παρτίδας πρώτης ύλης ή προϊόντος από την παραγωγή μέχρι και την πώληση.

2.2.2.4. Χαρακτηριστικά χρηστών

Οι χρήστες είναι οι υπάλληλοι των επιχειρήσεων που εγκαθίσταται το λογισμικό, από απλούς εργάτες έως υψηλόβαθμα στελέχη.

2.2.2.5. Περιορισμοί

- Η εφαρμογή θα τρέχει σε λειτουργικό σύστημα Microsoft Windows Server 2003.
- Απαιτείται η ύπαρξη Microsoft SQL Server 2005 Express.
- Απαιτείται η ύπαρξη ενσύρματου ή ασύρματου δικτύου.
- Θα υπάρχει κάποιο νομικό πλαίσιο σύμφωνα με το οποίο θα πρέπει να συμμορφώνετε τόσο η εφαρμογή όσο και οι χρήστες της. Αυτό θα συνταχθεί από αντίστοιχη επιστημονική ομάδα νομικών συμβούλων.

2.2.2.6. Παραδοχές και εξαρτήσεις

2.2.2.6.1. Παραδοχές και εξαρτήσεις κατά την ανάπτυξη

Η ανάπτυξη του λογισμικού θα μεταβληθεί εάν:

- ❑ Αλλάξουν οι κανονισμοί και οι οδηγίες της Ε.Ε.
- ❑ Αλλάξουν οι απαιτήσεις του «πελάτη».
- ❑ Δεν έχουν γίνει κατανοητές οι απαιτήσεις του «πελάτη».

2.2.2.6.2. Παραδοχές και εξαρτήσεις κατά την χρήση

Η χρήση του λογισμικού θα διαταραχθεί εάν:

- ❑ Αλλάξουν τα δεδομένα λειτουργίας του λογισμικού (π.χ. αλλαγή στις παραγωγικές διαδικασίες).
- ❑ Υπάρξουν τεχνικά προβλήματα στο δίκτυο ή σε άλλες εφαρμογές που συνεργάζεται το λογισμικό (π.χ. ERP σύστημα).

2.2.3. Ειδικές απαιτήσεις

2.2.3.1. Απαιτήσεις εξωτερικών διαπροσωπειών

2.2.3.1.1. Διαπροσωπείες χρήστη

- ☞ Το λογισμικό θα μπορεί να ελέγχεται με τη χρήση του mouse και του πληκτρολογίου.
- ☞ Το λογισμικό θα μπορεί να δέχεται δεδομένα και από Barcode Scanner.
- ☞ Το λογισμικό θα συνοδεύεται από αναλυτικό εγχειρίδιο χρήστη.
- ☞ Οθόνες επικοινωνίας.*

2.2.3.1.2. Διαπροσωπείες υλικού

- ☞ Το λογισμικό θα επικοινωνεί με εκτυπωτή για την εκτύπωση πληροφοριών σχετικά με αποθέματα, παραγωγή και παραγγελίες πελατών.
- ☞ Θα είναι συνδεδεμένο με δίκτυο, ενσύρματο ή ασύρματο, για να υπάρχει πρόσβαση από όλους τους χρήστες.

* Οι οθόνες επικοινωνίας χρήστη – λογισμικού θα περιγραφούν αναλυτικότερα στο έγγραφο περιγραφής του σχεδίου του λογισμικού.

- ☞ Το λογισμικό παρέχει τη δυνατότητα σύνδεσης με Barcode Scanner.

2.2.3.1.3. Διαπροσωπείες λογισμικού

- ☞ Το λογισμικό θα συνεργάζεται με το πρόγραμμα διαχείρισης βάσεων δεδομένων Microsoft SQL Server 2005, για την αποθήκευση και διάθεση στους χρήστες πληροφοριών σχετικά με τις πρώτες ύλες, τα προϊόντα, τους πελάτες, τις παραγγελίες, κ.α.
- ☞ Στην περίπτωση που η εταιρία υποστηρίζεται από ERP, το λογισμικό αυτό θα μπορεί να συνεργάζεται με το λογισμικό του ERP της εταιρίας.

2.2.3.1.4. Διαπροσωπείες επικοινωνιών

- ☞ Θα επικοινωνεί με υπολογιστές που είναι συνδεδεμένοι στο δίκτυο, ενσύρματο ή ασύρματο, ώστε να υπάρχει πρόσβαση από όλους τους χρήστες, με βάση το πρωτόκολλο TCP/IP.
- ☞ Θα επικοινωνεί με το σύστημα διαχείρισης βάσεως δεδομένων Microsoft SQL Server 2005.

2.2.3.2. Λειτουργικές απαιτήσεις

2.2.3.2.1. Τρόπος λειτουργίας 1 (Διαχείριση Πρώτων Υλών)

2.2.3.2.1.1. Λειτουργική απαίτηση 1.1

- Περιγραφή: Εισαγωγή νέων πρώτων υλών.
- Είσοδος: Δεδομένα από χρήστη.
- Επεξεργασία: Προετοιμασία και αποθήκευση δεδομένων στον πίνακα Πρώτων Υλών.
- Έξοδος: Ενημερωμένη βάση δεδομένων Πρώτων Υλών.

2.2.3.2.1.2. Λειτουργική απαίτηση 1.2

- Περιγραφή: Τροποποίηση πρώτων υλών.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Πρώτων Υλών.
- Επεξεργασία: Τροποποίηση και αποθήκευση δεδομένων στον πίνακα Πρώτων Υλών.
- Έξοδος: Ενημερωμένη βάση δεδομένων Πρώτων Υλών.

2.2.3.2.1.3. Λειτουργική απαίτηση 1.3

- Περιγραφή: Διαγραφή πρώτων υλών.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Πρώτων Υλών, βάση δεδομένων Προϊόντων, βάση δεδομένων Αποθεμάτων Πρώτων Υλών.
- Επεξεργασία: Έλεγχος σύνδεσης πρώτων υλών με προϊόντα και διαγραφή.

- Έξοδος: Ενημερωμένη βάση δεδομένων Πρώτων Υλών, μήνυμα λάθους.

2.2.3.2.2. Τρόπος λειτουργίας 2 (Διαχείριση Προϊόντων)

2.2.3.2.2.1. Λειτουργική απαίτηση 2.1

- Περιγραφή: Εισαγωγή νέων προϊόντων.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Πρώτων Υλών.
- Επεξεργασία: Προετοιμασία και αποθήκευση δεδομένων στον πίνακα Προϊόντων.
- Έξοδος: Ενημερωμένη βάση δεδομένων Προϊόντων.

2.2.3.2.2.2. Λειτουργική απαίτηση 2.2

- Περιγραφή: Τροποποίηση προϊόντων.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Προϊόντων, βάση δεδομένων Πρώτων Υλών.
- Επεξεργασία: Τροποποίηση και αποθήκευση δεδομένων στον πίνακα Προϊόντων.
- Έξοδος: Ενημερωμένη βάση δεδομένων Προϊόντων.

2.2.3.2.2.3. Λειτουργική απαίτηση 2.3

- Περιγραφή: Διαγραφή προϊόντων.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Προϊόντων, βάση δεδομένων Παραγόμενων Προϊόντων, βάση δεδομένων Παραγγελιών, βάση δεδομένων Αποθεμάτων Προϊόντων.
- Επεξεργασία: Έλεγχος σύνδεσης προϊόντων με πρώτες ύλες - πελάτες και διαγραφή.
- Έξοδος: Ενημερωμένη βάση δεδομένων Προϊόντων, μήνυμα λάθους.

2.2.3.2.3. Τρόπος λειτουργίας 3 (Διαχείριση Πελατών)

2.2.3.2.3.1. Λειτουργική απαίτηση 3.1

- Περιγραφή: Εισαγωγή νέων πελατών.
- Είσοδος: Δεδομένα από χρήστη.
- Επεξεργασία: Προετοιμασία και αποθήκευση δεδομένων στον πίνακα Πελατών.
- Έξοδος: Ενημερωμένη βάση δεδομένων Πελατών.

2.2.3.2.3.2. Λειτουργική απαίτηση 3.2

- Περιγραφή: Τροποποίηση πελατών.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Πελατών.
- Επεξεργασία: Τροποποίηση και αποθήκευση δεδομένων στον πίνακα Πελατών.
- Έξοδος: Ενημερωμένη βάση δεδομένων Πελατών.

2.2.3.2.3.3. Λειτουργική απαίτηση 3.3

- Περιγραφή: Διαγραφή πελατών.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Πελατών, βάση δεδομένων Παραγγελιών.
- Επεξεργασία: Έλεγχος σύνδεσης πελατών με προϊόντα και διαγραφή.
- Έξοδος: Ενημερωμένη βάση δεδομένων Πελατών, μήνυμα λάθους.

2.2.3.2.4. Τρόπος λειτουργίας 4 (Διαχείριση Αποθεμάτων Πρώτων Υλών)

2.2.3.2.4.1. Λειτουργική απαίτηση 4.1

- Περιγραφή: Εισαγωγή αποθεμάτων πρώτων υλών.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Πρώτων Υλών.
- Επεξεργασία: Προετοιμασία και αποθήκευση δεδομένων στον πίνακα Αποθεμάτων Πρώτων Υλών.
- Έξοδος: Ενημερωμένη βάση δεδομένων Αποθέματος Πρώτων Υλών.

2.2.3.2.4.2. Λειτουργική απαίτηση 4.2

- Περιγραφή: Τροποποίηση αποθεμάτων πρώτων υλών.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Αποθέματος Πρώτων Υλών.
- Επεξεργασία: Τροποποίηση και αποθήκευση δεδομένων στον πίνακα Αποθεμάτων Πρώτων Υλών.
- Έξοδος: Ενημερωμένη βάση δεδομένων Αποθέματος Πρώτων Υλών.

2.2.3.2.4.3. Λειτουργική απαίτηση 4.3

- Περιγραφή: Διαγραφή αποθεμάτων πρώτων υλών.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Αποθέματος Πρώτων Υλών, βάση δεδομένων Παραγόμενων Προϊόντων.
- Επεξεργασία: Έλεγχος σύνδεσης αποθέματος πρώτων υλών με προϊόντα και διαγραφή.
- Έξοδος: Ενημερωμένη βάση δεδομένων Αποθέματος Πρώτων Υλών, μήνυμα λάθους.

2.2.3.2.5. Τρόπος λειτουργίας 5 (Διαχείριση Αποθεμάτων Προϊόντων)

2.2.3.2.5.1. Λειτουργική απαίτηση 5.1

- Περιγραφή: Εισαγωγή αποθεμάτων προϊόντων.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Προϊόντων.
- Επεξεργασία: Προετοιμασία και αποθήκευση δεδομένων στον πίνακα Αποθεμάτων Προϊόντων.
- Έξοδος: Ενημερωμένη βάση δεδομένων Αποθέματος Προϊόντων.

2.2.3.2.5.2. Λειτουργική απαίτηση 5.2

- Περιγραφή: Τροποποίηση αποθεμάτων προϊόντων.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Αποθέματος Προϊόντων.
- Επεξεργασία: Τροποποίηση και αποθήκευση δεδομένων στον πίνακα Αποθεμάτων Προϊόντων.
- Έξοδος: Ενημερωμένη βάση δεδομένων Αποθέματος Προϊόντων.

2.2.3.2.5.3. Λειτουργική απαίτηση 5.3

- Περιγραφή: Διαγραφή αποθεμάτων προϊόντων.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Αποθέματος Προϊόντων, βάση δεδομένων Παραγγελιών.
- Επεξεργασία: Έλεγχος σύνδεσης αποθέματος προϊόντων με παραγγελίες και διαγραφή.
- Έξοδος: Ενημερωμένη βάση δεδομένων Αποθέματος Πρώτων Υλών, μήνυμα λάθους.

2.2.3.2.6. Τρόπος λειτουργίας 6 (Παραγωγή Προϊόντων)

2.2.3.2.6.1. Λειτουργική απαίτηση 6.1

- Περιγραφή: Προετοιμασία παραγωγής προϊόντων.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Προϊόντων, βάση δεδομένων Αποθεμάτων Πρώτων Υλών.
- Επεξεργασία: Αλγόριθμος παραγωγής.
- Έξοδος: Φόρμα παραγωγής (οθόνη), μήνυμα λάθους.

2.2.3.2.6.2. Λειτουργική απαίτηση 6.2

- Περιγραφή: Υλοποίηση παραγωγής προϊόντων.
- Είσοδος: Δεδομένα από χρήστη, φόρμα παραγωγής (οθόνη), βάση δεδομένων Αποθεμάτων Πρώτων Υλών, βάση δεδομένων Αποθεμάτων Προϊόντων.
- Επεξεργασία: Αύξηση αποθέματος προϊόντων, μείωση αποθέματος πρώτων υλών.
- Έξοδος: Ενημερωμένη βάση δεδομένων Αποθέματος Προϊόντων, ενημερωμένη βάση δεδομένων Αποθέματος Πρώτων Υλών, ενημερωμένη βάση δεδομένων Παραγόμενων Προϊόντων, μήνυμα λάθους.

2.2.3.2.7. Τρόπος λειτουργίας 7 (Παραγγελία Προϊόντων)

2.2.3.2.7.1. Λειτουργική απαίτηση 7.1

- Περιγραφή: Δημιουργία νέας παραγγελίας.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Προϊόντων, βάση δεδομένων Πελατών.

- Επεξεργασία: Προετοιμασία και αποθήκευση δεδομένων στον πίνακα Παραγγελιών.
- Έξοδος: Ενημερωμένη βάση δεδομένων Παραγγελιών, ενημερωμένη κατάσταση παραγγελιών (προς εκτέλεση).

2.2.3.2.7.2. Λειτουργική απαίτηση 7.2

- Περιγραφή: Τροποποίηση παραγγελιών.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Παραγγελιών.
- Επεξεργασία: Τροποποίηση και αποθήκευση δεδομένων στον πίνακα Παραγγελιών.
- Έξοδος: Ενημερωμένη βάση δεδομένων Παραγγελιών.

2.2.3.2.7.3. Λειτουργική απαίτηση 7.3

- Περιγραφή: Διαγραφή παραγγελιών.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Παραγγελιών
- Επεξεργασία: Έλεγχος σύνδεσης παραγγελιών με πελάτες, εάν η κατάσταση της προς διαγραφή παραγγελίας είναι ολοκληρωμένη τότε δεν επιτρέπεται η διαγραφή, αύξηση αντίστοιχων αποθεμάτων προϊόντων και διαγραφή παραγγελίας.
- Έξοδος: Ενημερωμένη βάση δεδομένων Παραγγελιών, ενημερωμένη βάση δεδομένων Αποθεμάτων Προϊόντων, μήνυμα λάθους.

2.2.3.2.7.4. Λειτουργική απαίτηση 7.4

- Περιγραφή: Εκτέλεση παραγγελιών.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Παραγγελιών, βάση δεδομένων Αποθέματος Προϊόντων.
- Επεξεργασία: Έλεγχος αποθέματος προϊόντων, ταξινόμηση βάση των lot number του αποθέματος των προϊόντων.
- Έξοδος: Φόρμα εκτέλεσης παραγγελιών (οθόνη), ενημερωμένη βάση δεδομένων Εκτελεσμένων Παραγγελιών, ενημερωμένη κατάσταση παραγγελίας (υπό εκτέλεση), μήνυμα λάθους.

2.2.3.2.7.5. Λειτουργική απαίτηση 7.5

- Περιγραφή: Ολοκλήρωση παραγγελιών.
- Είσοδος: Δεδομένα από χρήστη, φόρμα Εκτελεσμένων Παραγγελιών, βάση δεδομένων Παραγγελιών.
- Επεξεργασία: Μείωση αποθέματος προϊόντων και ολοκλήρωση παραγγελιών.
- Έξοδος: Ενημερωμένη κατάσταση παραγγελίας (ολοκληρωμένη), μήνυμα λάθους.

2.2.3.2.8. Τρόπος λειτουργίας 8 (Ιχνηλασιμότητα)

2.2.3.2.8.1. Λειτουργική απαίτηση 8.1

- Περιγραφή: Αναζήτηση πρώτων υλών.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Παραγόμενων Προϊόντων, βάση δεδομένων Εκτελεσμένων Παραγγελιών, βάση δεδομένων Παραγγελιών.
- Επεξεργασία: Εύρεση και παρουσίαση των αναζητούμενων πρώτων υλών βάσει προϊόντων ή / και πελατών.
- Έξοδος: Φόρμα αναζήτησης πρώτων υλών (οθόνη) , μήνυμα λάθους.

2.2.3.2.8.2. Λειτουργική απαίτηση 8.2

- Περιγραφή: Αναζήτηση προϊόντων.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Παραγόμενων Προϊόντων, βάση δεδομένων Εκτελεσμένων Παραγγελιών, βάση δεδομένων Παραγγελιών.
- Επεξεργασία: Εύρεση και παρουσίαση των αναζητούμενων προϊόντων βάσει πρώτων υλών ή / και πελατών.
- Έξοδος: Φόρμα αναζήτησης προϊόντων (οθόνη), μήνυμα λάθους.

2.2.3.2.8.3. Λειτουργική απαίτηση 8.3

- Περιγραφή: Αναζήτηση πελατών.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Παραγόμενων Προϊόντων, βάση δεδομένων Εκτελεσμένων Παραγγελιών, βάση δεδομένων Παραγγελιών.
- Επεξεργασία: Εύρεση και παρουσίαση των αναζητούμενων πελατών βάσει προϊόντων ή / και πρώτων υλών.
- Έξοδος: Φόρμα αναζήτησης πελατών (οθόνη), μήνυμα λάθους.

2.2.3.2.8.4. Λειτουργική απαίτηση 8.4

- Περιγραφή: Ανάκληση προϊόντων.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Παραγόμενων Προϊόντων, βάση δεδομένων Εκτελεσμένων Παραγγελιών, βάση δεδομένων Παραγγελιών.
- Επεξεργασία: Εύρεση συγκεκριμένων προϊόντων ή προϊόντων τα οποία έχουν παραχθεί από συγκεκριμένες πρώτες ύλες και ανάκληση αυτών.
- Έξοδος: Ενημερωμένη βάση δεδομένων Ανάκλησης, μήνυμα λάθους.

2.2.3.2.9. Τρόπος λειτουργίας 9 (Κεντρική Διαχείριση Εφαρμογής)

2.2.3.2.9.1. Λειτουργική απαίτηση 9.1

- Περιγραφή: Εισαγωγή στην εφαρμογή.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων χρηστών.

- Επεξεργασία: Έλεγχος των στοιχείων εισαγωγής από το χρήστη (username - password), κάθε χρήστης έχει και συγκεκριμένες δυνατότητες.
- Έξοδος: Είσοδος στην εφαρμογή, μήνυμα λάθους.

2.2.3.2.9.2. Λειτουργική απαίτηση 9.2

- Περιγραφή: Επιλογή λειτουργίας.
- Είσοδος: Δεδομένα από χρήστη.
- Επεξεργασία: Καταγραφή επιλογής στη βάση δεδομένων Χρηστών με σκοπό να γνωρίζουμε τις ενέργειες κάθε χρήστη.
- Έξοδος: Άνοιγμα επιλεγμένης λειτουργίας, ενημερωμένη βάση δεδομένων Χρηστών.

2.2.3.2.9.3. Λειτουργική απαίτηση 9.3

- Περιγραφή: Κλείσιμο εφαρμογής.
- Είσοδος: Δεδομένα από χρήστη.
- Επεξεργασία: Προετοιμασία και έλεγχο μη ολοκληρωμένων λειτουργιών.
- Έξοδος: Κλείσιμο εφαρμογής, μήνυμα εκκρεμοτήτων.

2.2.3.2.10. Τρόπος λειτουργίας 10 (Διαχείριση Χρηστών)

2.2.3.2.10.1. Λειτουργική απαίτηση 10.1

- Περιγραφή: Εισαγωγή νέων χρηστών.
- Είσοδος: Δεδομένα από χρήστη.
- Επεξεργασία: Προετοιμασία και αποθήκευση δεδομένων στον πίνακα Χρηστών.
- Έξοδος: Ενημερωμένη βάση δεδομένων Χρηστών.

2.2.3.2.10.2. Λειτουργική απαίτηση 10.2

- Περιγραφή: Τροποποίηση χρηστών.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Χρηστών.
- Επεξεργασία: Τροποποίηση και αποθήκευση δεδομένων στον πίνακα Χρηστών.
- Έξοδος: Ενημερωμένη βάση δεδομένων Χρηστών.

2.2.3.2.10.3. Λειτουργική απαίτηση 10.3

- Περιγραφή: Διαγραφή χρηστών.
- Είσοδος: Δεδομένα από χρήστη, βάση δεδομένων Χρηστών.
- Επεξεργασία: Έλεγχος σύνδεσης χρηστών με παραγωγή, παραγγελία και ανάκληση προϊόντων, διαγραφή αυτών.
- Έξοδος: Ενημερωμένη βάση δεδομένων Χρηστών, μήνυμα λάθους.

2.2.3.2.11. Τρόπος λειτουργίας 11 (Συντήρηση)

2.2.3.2.11.1. Λειτουργική απαίτηση 11.1

- Περιγραφή: Συντήρηση Συστήματος
- Είσοδος: Παράμετροι / Πεδία συντήρησης
- Επεξεργασία: Αλλαγή τιμών στα πεδία συντήρησης
- Έξοδος: Ενημερωμένη βάση δεδομένων συντήρησης προγράμματος

2.2.3.2.11.2. Λειτουργική απαίτηση 11.2

- Περιγραφή:
- Είσοδος:
- Επεξεργασία:
- Έξοδος:

2.2.3.2.11.3. Λειτουργική απαίτηση 11.3

- Περιγραφή:
- Είσοδος:
- Επεξεργασία:
- Έξοδος:

2.2.3.3. Απαιτήσεις επιδόσεων

- Ο χρόνος ενημέρωσης της βάσης δεδομένων να μην ξεπερνά τα 80msec.
- Η χρόνος αναμονής για την περιήγηση μεταξύ των σελίδων του λογισμικού να μην ξεπερνά τα 25msec.
- Ο μέγιστος χρόνος απόκρισης εφαρμογής για οποιαδήποτε ενέργεια του χρήστη δεν πρέπει να ξεπερνά τα 6sec. Σε αντίθετη περίπτωση να εμφανίζει μήνυμα που να ενημερώνει τον χρήστη και να αναφέρει τους λόγους της καθυστέρησης.

2.2.3.4. Περιορισμοί σχεδίασης

2.2.3.4.1. Περιορισμοί από το υλικό

Δεν υπάρχουν περιορισμοί από το υλικό εάν θεωρήσουμε ότι αυτό είναι συμβατό με το λειτουργικό σύστημα που θα χρησιμοποιηθεί και καλύπτει προδιαγραφές υπολογιστή που θα τρέχει η εφαρμογή (Βλ. Άλλες απαιτήσεις).

2.2.3.4.2. Συμμόρφωση με πρότυπα

- Η δομή του εγγράφου συμμορφώνετε με το διεθνές πρότυπο του IEEE(830-1993).
- Για την κατασκευή του λογισμικού ακολουθούμε το μοντέλο πρωτοτυποποίησης.

2.2.3.5. Χαρακτηριστικά του λογισμικού

2.2.3.5.1. Αξιοπιστία

- Σε περίπτωση απρόβλεπτου τερματισμού της λειτουργίας του λογισμικού θα πρέπει με την επανεκκίνηση το λογισμικό θα πρέπει να επανέλθει στην τελευταία κατάσταση που βρισκόταν.
- Σε περίπτωση απρόβλεπτου τερματισμού τη στιγμή που ενημερώνεται η βάση δεδομένων θα πρέπει να κρατούνται τα στοιχεία της βάσης πριν τις πιθανές αλλαγές και να πληροφορείται ο χρήστης για την αποτυχία ενημέρωσης.
- Σε περίπτωση που δύο ή περισσότεροι χρήστες προσπαθούν να προσπελάσουν τα ίδια δεδομένα και να τα μεταβάλλουν το λογισμικό θα πρέπει να τα κλειδώνει και να δίνει την δυνατότητα στον πρώτο κατά χρονική προτεραιότητα, ενώ ταυτόχρονα να εμφανίζει κατάλληλο μήνυμα στον δεύτερο.

2.2.3.5.2. Διαθεσιμότητα

Η εφαρμογή πρέπει να είναι διαθέσιμη κατά το 99,8% του χρόνου λειτουργίας της. Στον υπόλοιπο χρόνο μπορεί να υπάρξει αδυναμία ανταπόκρισης από την εφαρμογή.

2.2.3.5.3. Ασφάλεια

- Δεν επιτρέπεται η διαγραφή στοιχείων της βάσης δεδομένων χωρίς την είσοδο χρήστη με την μορφή Administrator.
- Η διαγραφή των βάσεων δεδομένων επιτρέπεται μόνο με την διαγραφή ολόκληρης της εφαρμογής.
- Η διαγραφή της εφαρμογής επιτρέπεται μόνο με την χρήση κατάλληλου κωδικού.
- Δεν επιτρέπεται παρέμβαση χρήστη στις βάσεις δεδομένων.

2.2.3.5.4. Χαρακτηριστικά συντήρησης

Η συντήρηση πραγματοποιείται μόνο με την χρήση του CD εγκατάστασης και κατάλληλου κωδικού. Οι δυνατότητες είναι οι εξής:

- Επιδιόρθωση του λογισμικού (εντοπίζει και επιδιορθώνει σφάλματα στην εγκατάσταση ή έλλειψη αρχείων κ.λ.π.).
- Προσθαφαίρεση / επιδιόρθωση βάσης δεδομένων
- Κατάργηση της εγκατάστασης της εφαρμογής.

2.2.3.5.5. Μεταφερσιμότητα

Η εφαρμογή θα λειτουργεί μόνο σε περιβάλλον Microsoft Windows.

2.2.3.6. Άλλες απαιτήσεις

2.2.3.6.1. Φυσικές απαιτήσεις

- Η εφαρμογή θα μπορεί να λειτουργεί σε οποιαδήποτε έκδοση Microsoft Windows, αλλά κατά προτίμηση στο Microsoft Windows Server 2003.
- Προδιαγραφές υπολογιστή:
 - Pentium III / 733MHZ
 - 1024MB RAM
 - 800X600 Ανάλυση οθόνης
 - CD-ROM 4X
 - Σύνδεση σε δίκτυο.

2.2.3.6.2. Απαιτήσεις Βάσεων Δεδομένων

- Η εφαρμογή θα πρέπει να διατηρεί αρχείο πρώτων υλών με τα εξής στοιχεία:
 - Κωδικός
 - Περιγραφή
 - ΜΜ Εισαγωγής
 - Χρόνος ανάλωσης
 - Μονάδα χρόνου
 - Κωδικοποίηση παρτίδας
 - Κατάσταση προϊόντος
 - Κωδικός αποθήκης
 - Κωδικός συσκευασίας
 - Ελάχιστο βάρος
 - Εκτιμώμενο βάρος
 - Μέγιστο βάρος
 - ΜΜ Βάρους

- Η εφαρμογή θα πρέπει να διατηρεί αρχείο προϊόντων με τα εξής πεδία:
 - Κωδικός
 - Περιγραφή
 - ΜΜ Εισαγωγής
 - ΜΜ Πώλησης
 - Χρόνος ανάλωσης
 - Μονάδα χρόνου
 - Μέθοδος διάθεσης
 - Κωδικοποίηση παρτίδας
 - Κατάσταση προϊόντος
 - Κωδικός αποθήκης
 - Κωδικός συσκευασίας
 - Τρόπος πώλησης
 - Ελάχιστο βάρος
 - Εκτιμώμενο βάρος
 - Μέγιστο βάρος
 - ΜΜ Βάρους
 - Πρώτη ύλη 1
 - Πρώτη ύλη 2
 - Πρώτη ύλη ...
- Η εφαρμογή θα πρέπει να διατηρεί αρχείο πελατών με τα εξής πεδία:
 - Κωδικός
 - Επωνυμία
 - ΑΦΜ
 - Διεύθυνση
 - Προορισμός
 - Τηλέφωνο
 - Επαφή
 - Τηλ επαφής
 - Ηλ. Διεύθυνση επαφής
 - Αριθμός αυτοκινήτου
 - Κατάσταση πελάτη
 - Ημερομηνία εισαγωγής
- Η εφαρμογή θα πρέπει να διατηρεί αρχείο αποθέματος πρώτων υλών με τα εξής στοιχεία:

- Κωδικός
 - Περιγραφή
 - Παρτίδα
 - ΜΜ Εισαγωγής
 - Βάρος
 - Υπόλοιπο βάρους
 - Τεμάχια
 - Υπόλοιπο τεμαχίων
 - Ανάκληση αποθέματος
 - Ημερομηνία λήξης
 - Ημερομηνία εισαγωγής
 - Ημερομηνία επεξεργασίας
- Η εφαρμογή θα πρέπει να διατηρεί αρχείο αποθέματος προϊόντων με τα εξής πεδία:
- Κωδικός
 - Περιγραφή
 - Παρτίδα
 - ΜΜ Εισαγωγής
 - Βάρος
 - Υπόλοιπο βάρους
 - Τεμάχια
 - Υπόλοιπο τεμαχίων
 - Ανάκληση αποθέματος
 - Ημερομηνία λήξης
 - Ημερομηνία εισαγωγής
 - Ημερομηνία επεξεργασίας
- Η εφαρμογή θα πρέπει να διατηρεί αρχείο παραγόμενων προϊόντων με τα εξής πεδία:
- Κωδικός προϊόντος
 - LOT προϊόντος
 - Ποσότητα προϊόντος
 - Κωδικός πρώτης ύλης 1
 - LOT πρώτης ύλης 1
 - Ποσότητα πρώτης ύλης 1
 - Κωδικός πρώτης ύλης ...

- LOT πρώτης ύλης ...
- Ποσότητα πρώτης ύλης ...
- Η εφαρμογή θα πρέπει να διατηρεί αρχείο παραγγελιών με τα εξής στοιχεία:
 - Κωδικός παραγγελίας
 - Κωδικός πελάτη
 - Ημερομηνία ολοκλήρωσης
 - Ημερομηνία επεξεργασίας
 - Προορισμός
 - Σχόλια
- Η εφαρμογή θα πρέπει να διατηρεί αρχείο γραμμών παραγγελιών με τα εξής πεδία:
 - Κωδικός παραγγελίας
 - Κωδικός προϊόντος
 - Lot προϊόντος
 - Ποσότητα
 - ΜΜ Παραγγελίας
 - Εκτελεσμένα τεμάχια
 - Εκτελεσμένα κιλά
- Η εφαρμογή θα πρέπει να διατηρεί αρχείο ανακληθέντων προϊόντων με τα εξής πεδία:
 - Κωδικός προϊόντος
 - Lot προϊόντος
 - Ποσότητα προϊόντος
 - Κωδικός παραγγελίας
 - Επωνυμία πελάτη
 - ΜΜ πώλησης
 - Εκτελεσμένα τεμάχια
 - Σχόλια
 - Κωδικός πρώτης ύλης 1
 - Lot πρώτης ύλης 1
 - Ποσότητα πρώτης ύλης 1
 - Κωδικός πρώτης ύλης ...
 - Lot πρώτης ύλης ...
 - Ποσότητα πρώτης ύλης ...
- Η εφαρμογή θα πρέπει να διατηρεί αρχείο χρηστών με τα εξής στοιχεία:

- Κωδικός
- Όνομα πρόσβασης
- Κωδικός πρόσβασης
- Όνομα
- Επίθετο
- Ομάδα

2.2.3.6.3. Απαιτήσεις Υλοποίησης

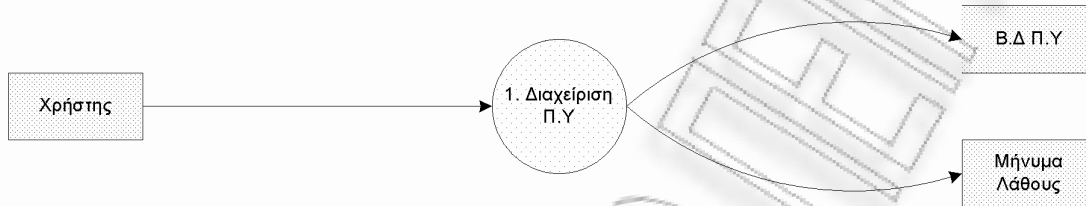
Για την συγγραφή του πυγαίου κώδικα της εφαρμογής θα χρησιμοποιήσουμε την γλώσσα προγραμματισμού VB.Net

2.3. ΜΟΝΤΕΛΑ ΠΑΡΑΣΤΑΣΗΣ ΛΟΓΙΣΜΙΚΟΥ

2.3.1. Διαγράμματα ροής δεδομένων

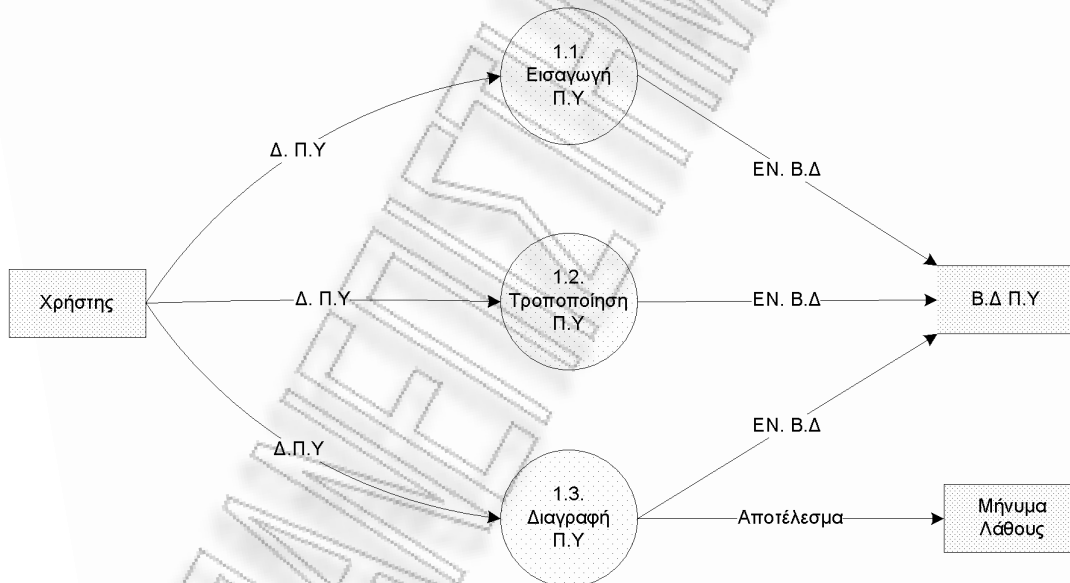
2.3.1.1. 1ος Τρόπος λειτουργίας (Διαχείριση πρώτων υλών)

2.3.1.1.1. Επίπεδο 1



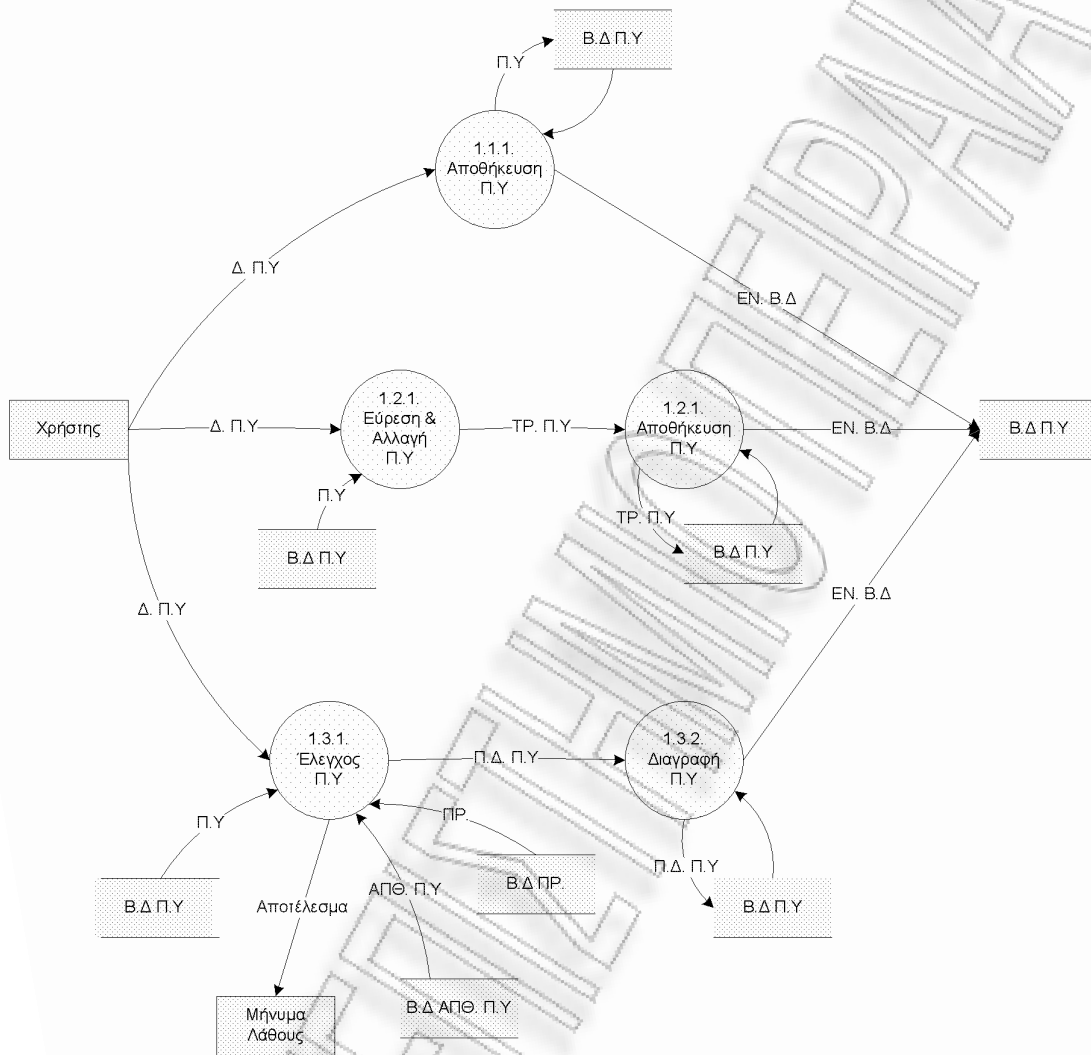
Διάγραμμα 1. 1ος Τρόπος Λειτουργίας - Επίπεδο 1

2.3.1.1.2. Επίπεδο 2



Διάγραμμα 2. 1ος Τρόπος Λειτουργίας - Επίπεδο 2

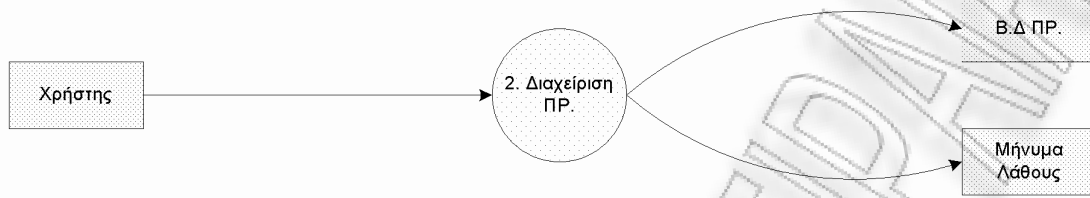
2.3.1.1.3. Επίπεδο 3



Διάγραμμα 3. 1ος Τρόπος Λειτουργίας - Επίπεδο 3

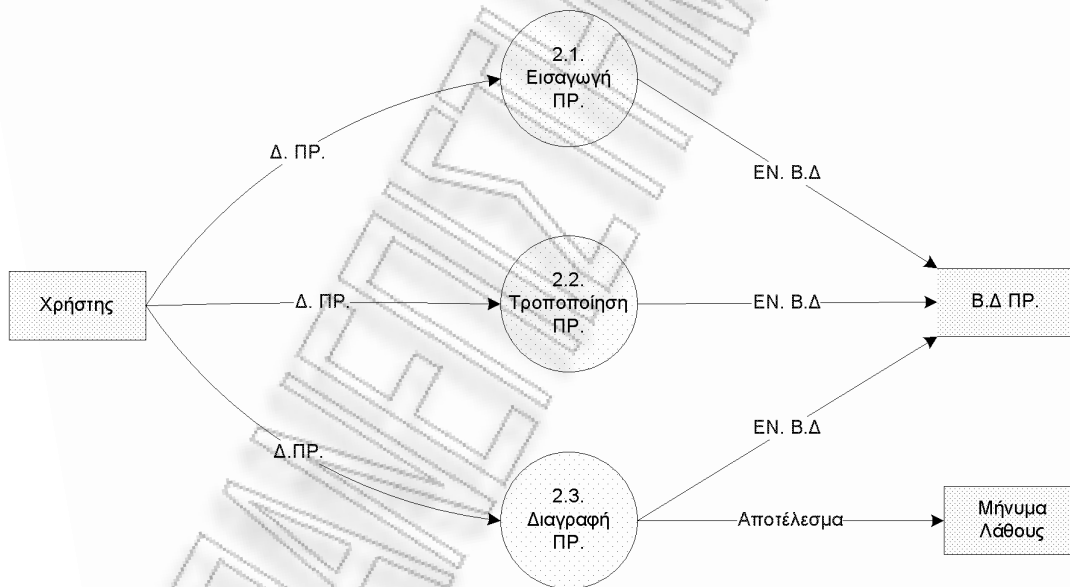
2.3.1.2. 2ος Τρόπος λειτουργίας (Διαχείριση προϊόντων)

2.3.1.2.1. Επίπεδο 1



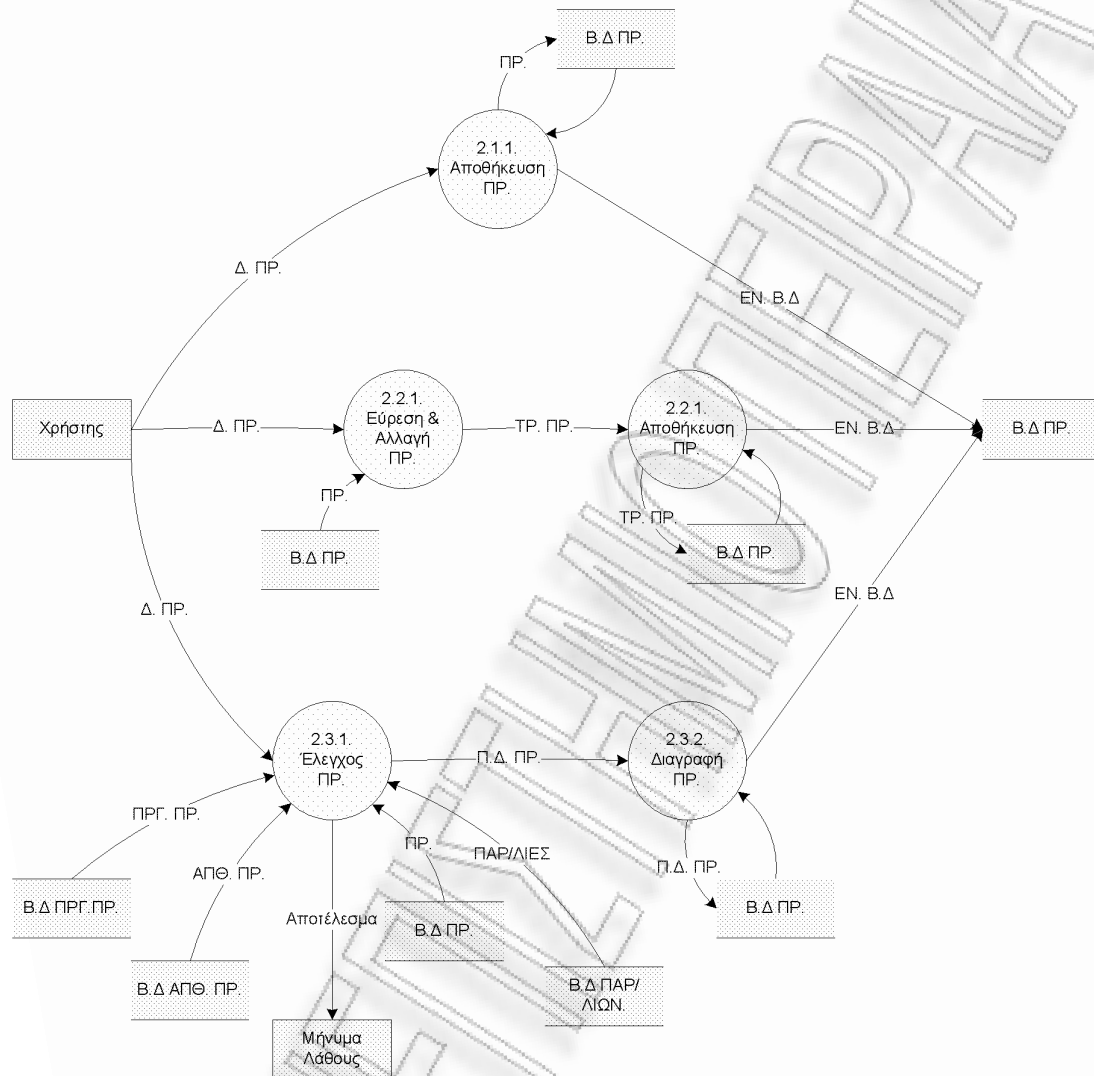
Διάγραμμα 4. 2ος Τρόπος λειτουργίας - Επίπεδο 1

2.3.1.2.2. Επίπεδο 2



Διάγραμμα 5. 2ος Τρόπος λειτουργίας - Επίπεδο 2

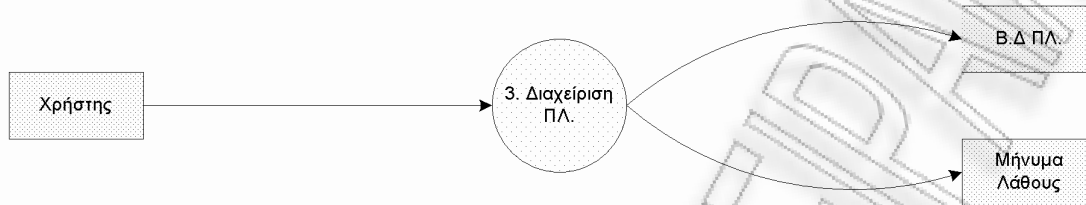
2.3.1.2.3. Επίπεδο 3



Διάγραμμα 6. 2ος Τρόπος Λειτουργίας - Επίπεδο 3

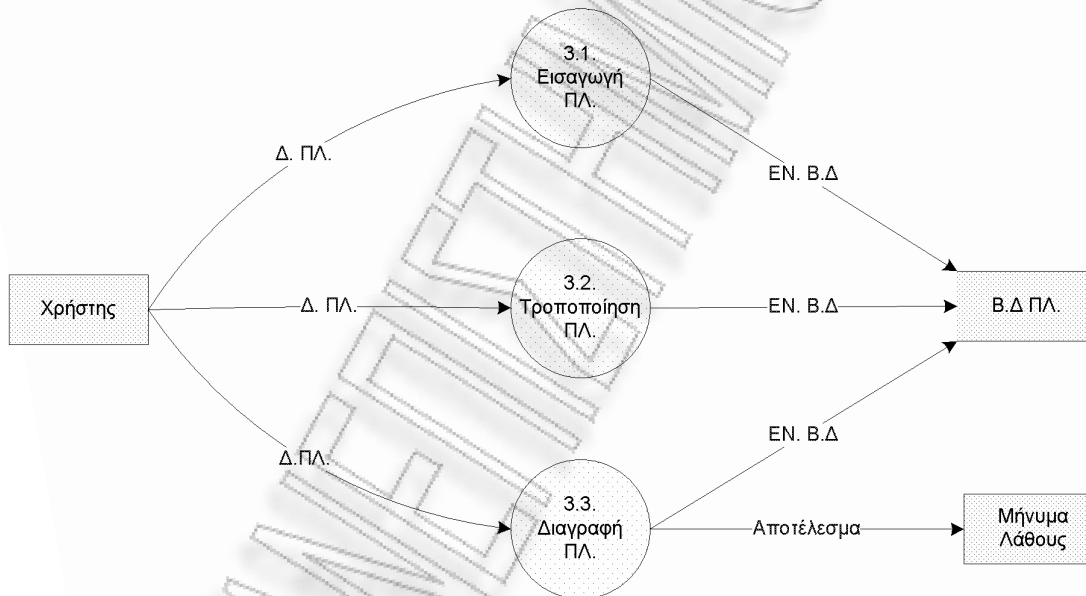
2.3.1.3. 3ος Τρόπος λειτουργίας (Διαχείριση πελατών)

2.3.1.3.1. Επίπεδο 1



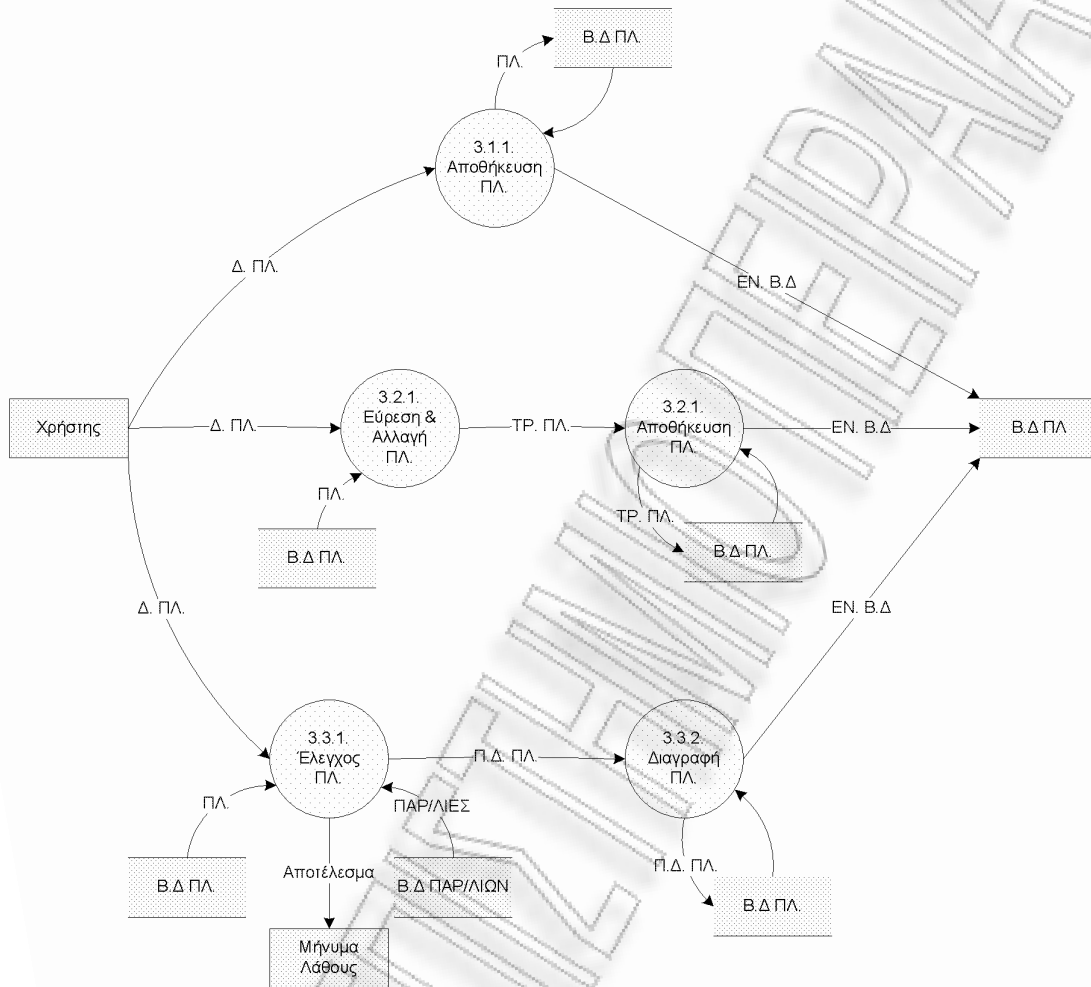
Διάγραμμα 7. 3ος Τρόπος λειτουργίας - Επίπεδο 1

2.3.1.3.2. Επίπεδο 2



Διάγραμμα 8. 3ος Τρόπος λειτουργίας - Επίπεδο 2

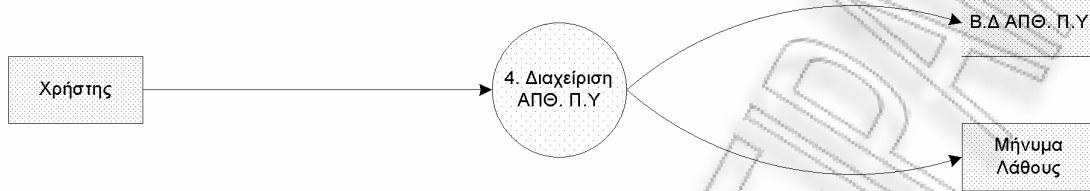
2.3.1.3.3. Επίπεδο 3



Διάγραμμα 9. 3ος Τρόπος Λειτουργίας - Επίπεδο 3

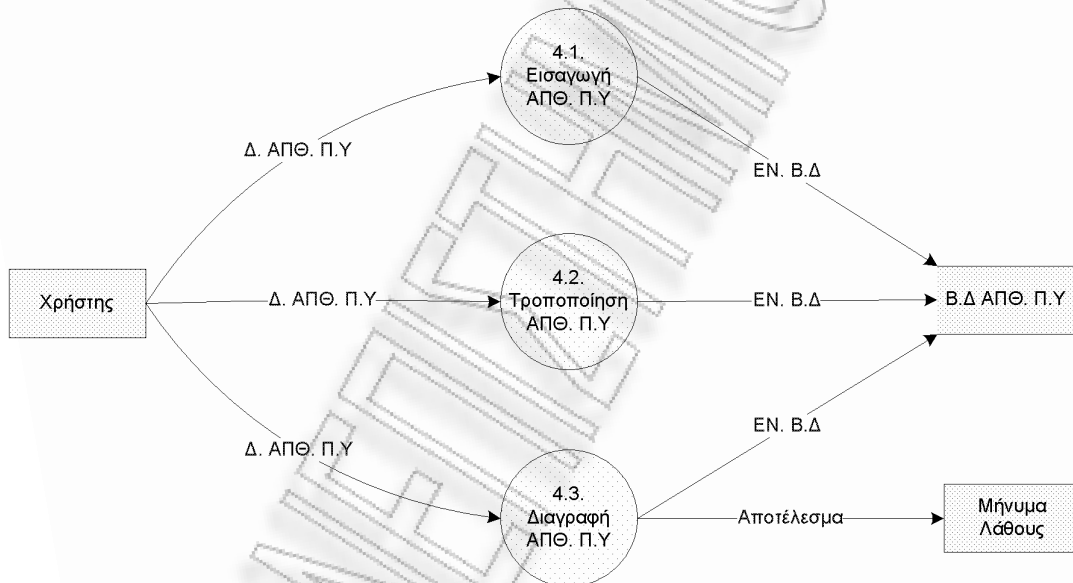
2.3.1.4. 4ος Τρόπος λειτουργίας (Διαχείριση αποθεμάτων πρώτων υλών)

2.3.1.4.1. Επίπεδο 1



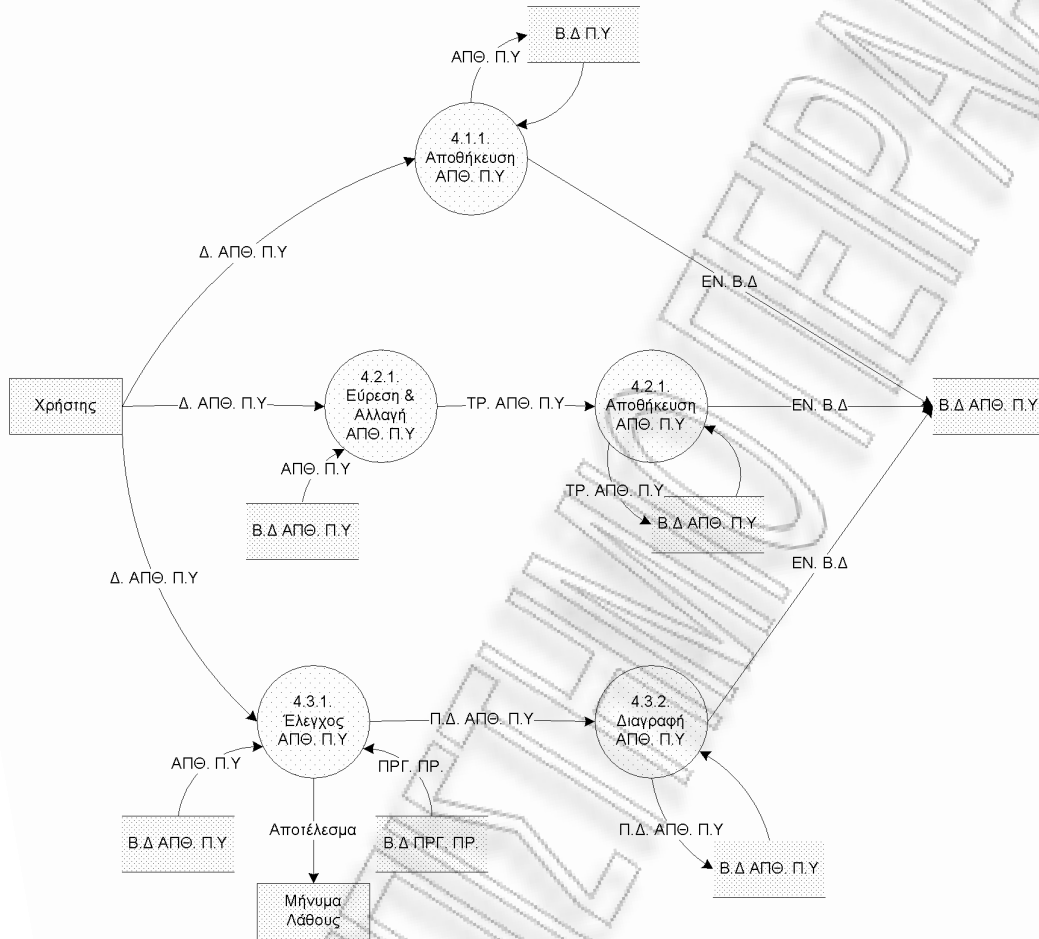
Διάγραμμα 10. 4ος Τρόπος Λειτουργίας - Επίπεδο 1

2.3.1.4.2. Επίπεδο 2



Διάγραμμα 11. 4ος Τρόπος Λειτουργίας - Επίπεδο 2

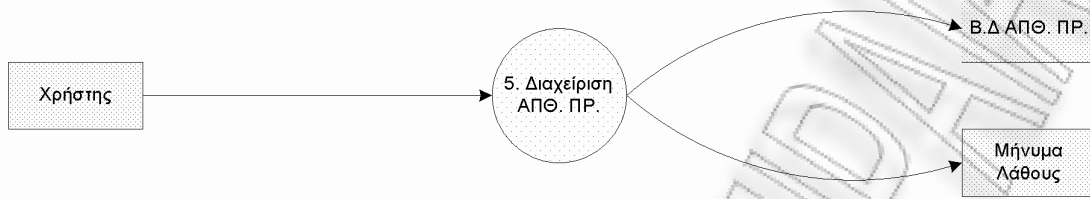
2.3.1.4.3. Επίπεδο 3



Διάγραμμα 12. 4ος Τρόπος Λειτουργίας - Επίπεδο 3

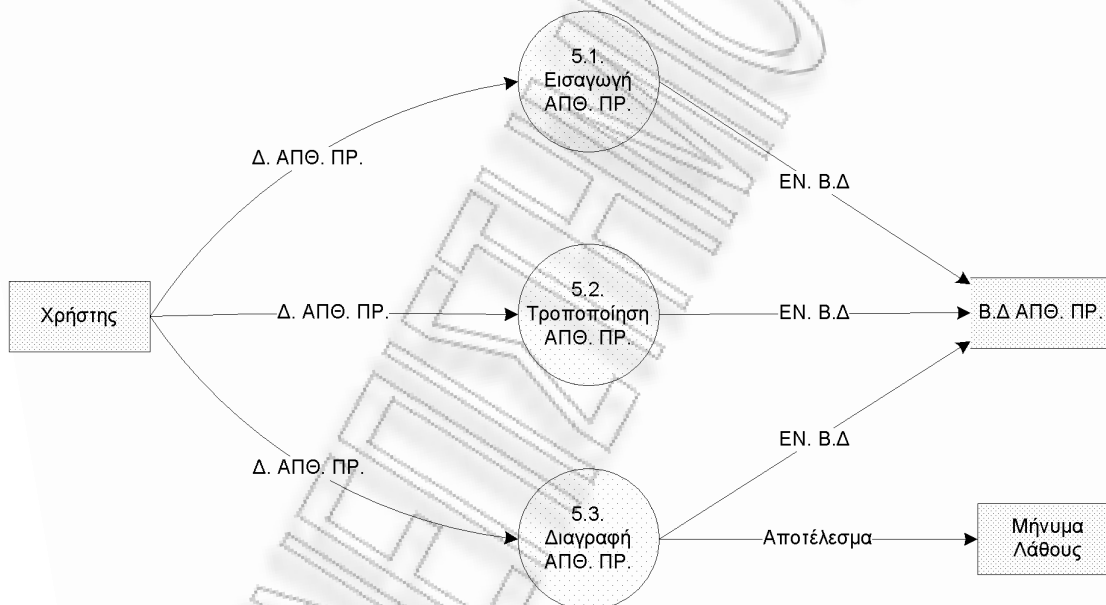
2.3.1.5. 5ος Τρόπος Λειτουργίας (Διαχείριση αποθεμάτων προϊόντων)

2.3.1.5.1. Επίπεδο 1



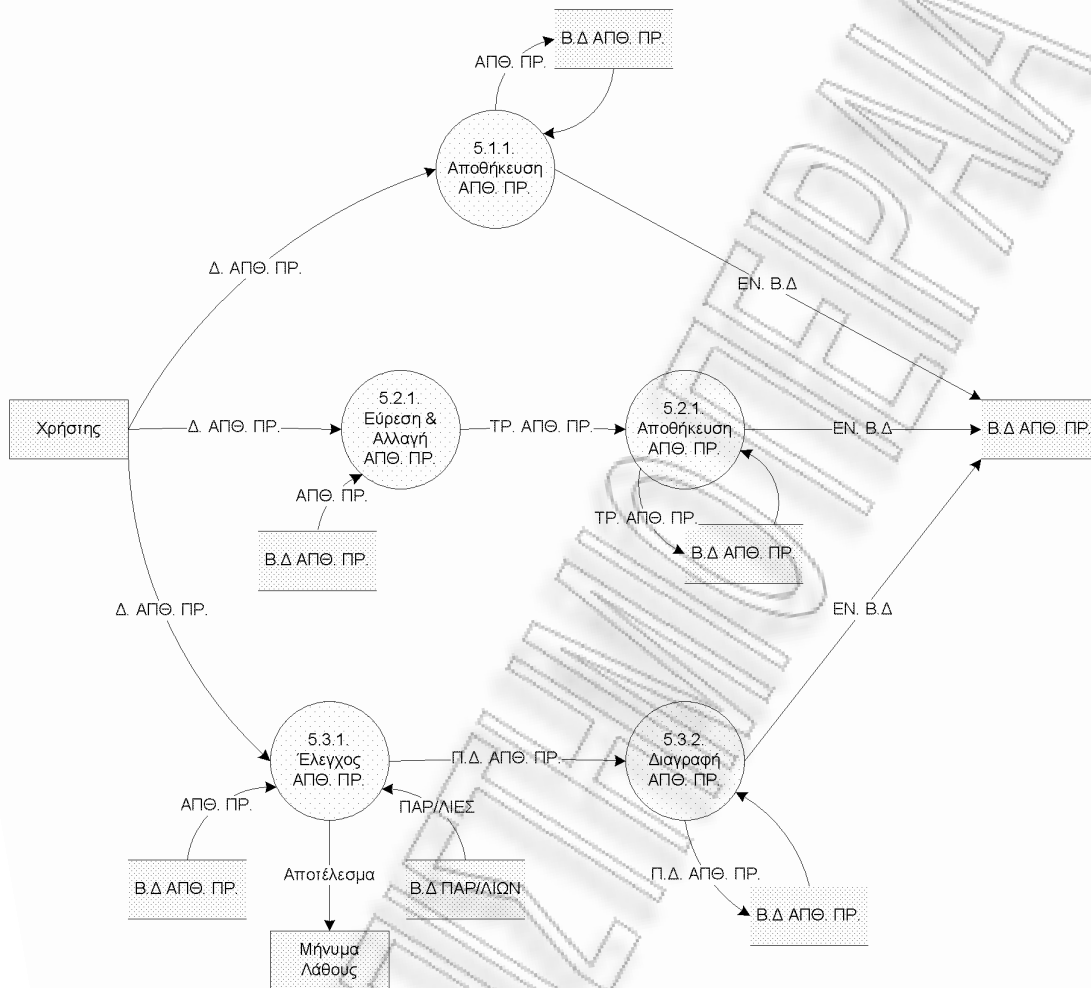
Διάγραμμα 13. 5ος Τρόπος Λειτουργίας - Επίπεδο 1

2.3.1.5.2. Επίπεδο 2



Διάγραμμα 14. 5ος Τρόπος Λειτουργίας - Επίπεδο 2

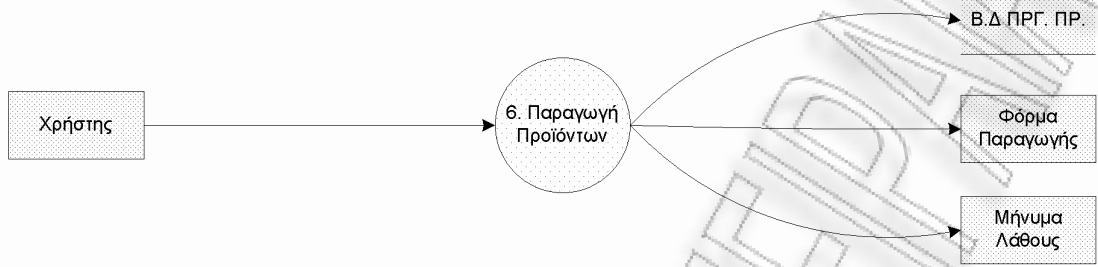
2.3.1.5.3. Επίπεδο 3



Διάγραμμα 15. 5ος Τρόπος Λειτουργίας - Επίπεδο 3

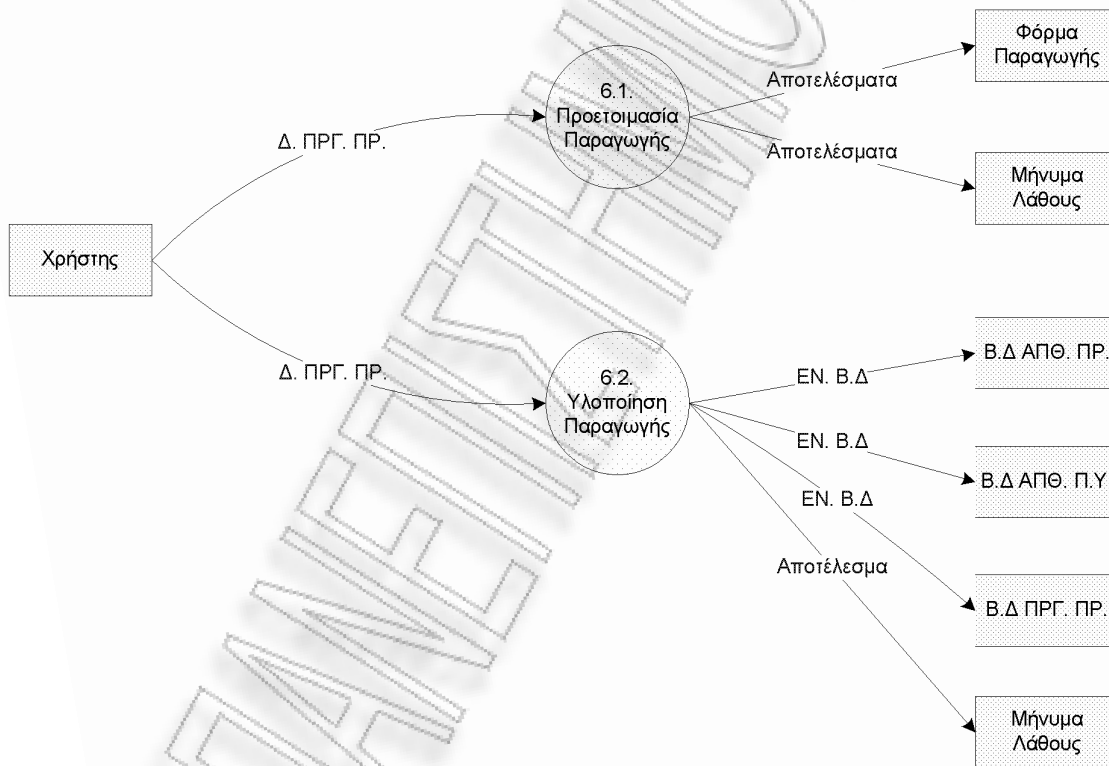
2.3.1.6. 6ος Τρόπος Λειτουργίας (Παραγωγή προϊόντων)

2.3.1.6.1. Επίπεδο 1



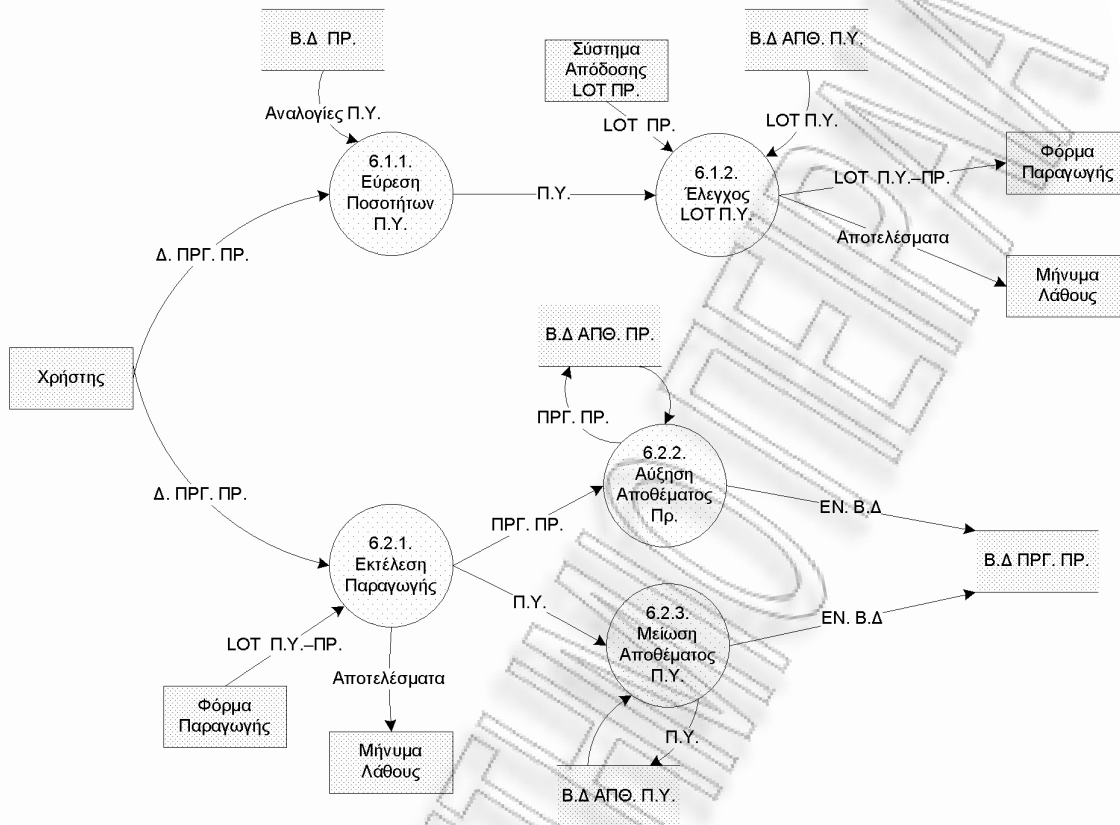
Διάγραμμα 16. 6ος Τρόπος Λειτουργίας - Επίπεδο 1

2.3.1.6.2. Επίπεδο 2



Διάγραμμα 17. 6ος Τρόπος Λειτουργίας - Επίπεδο 2

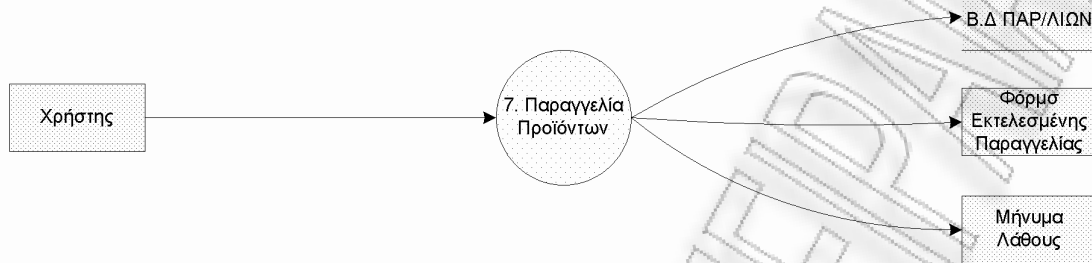
2.3.1.6.3. Επίπεδο 3



Διάγραμμα 18. 6ος Τρόπος Λειτουργίας - Επίπεδο 3

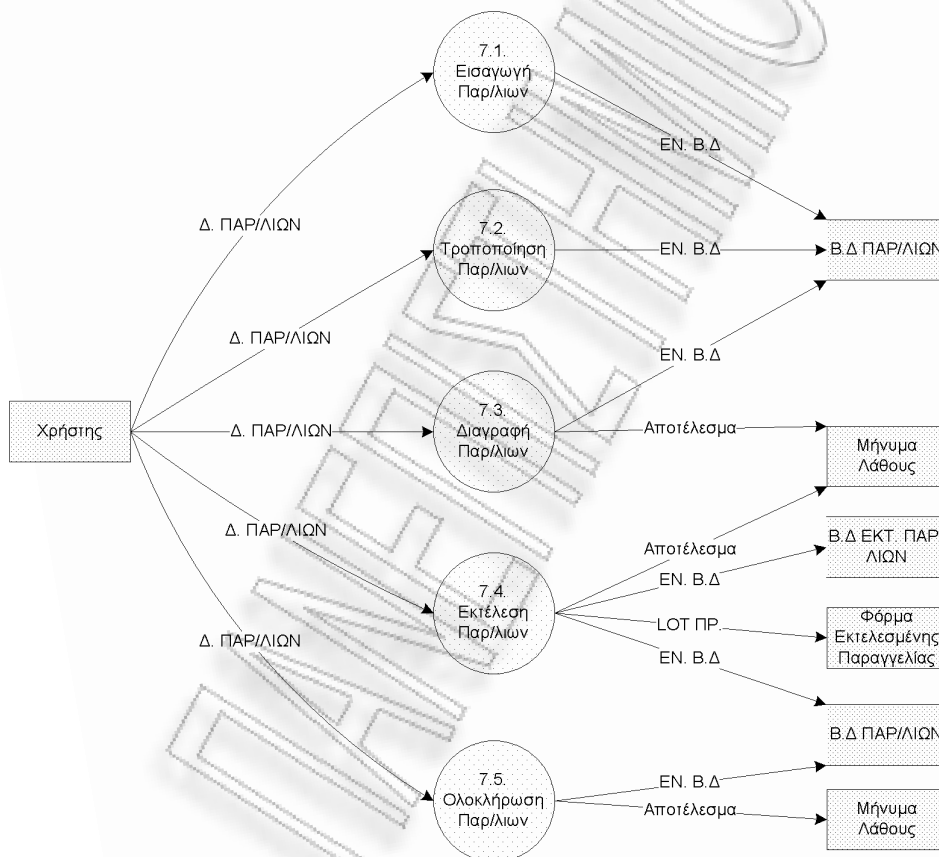
2.3.1.7. 7ος Τρόπος Λειτουργίας (Παραγγελίες προϊόντων)

2.3.1.7.1. Επίπεδο 1



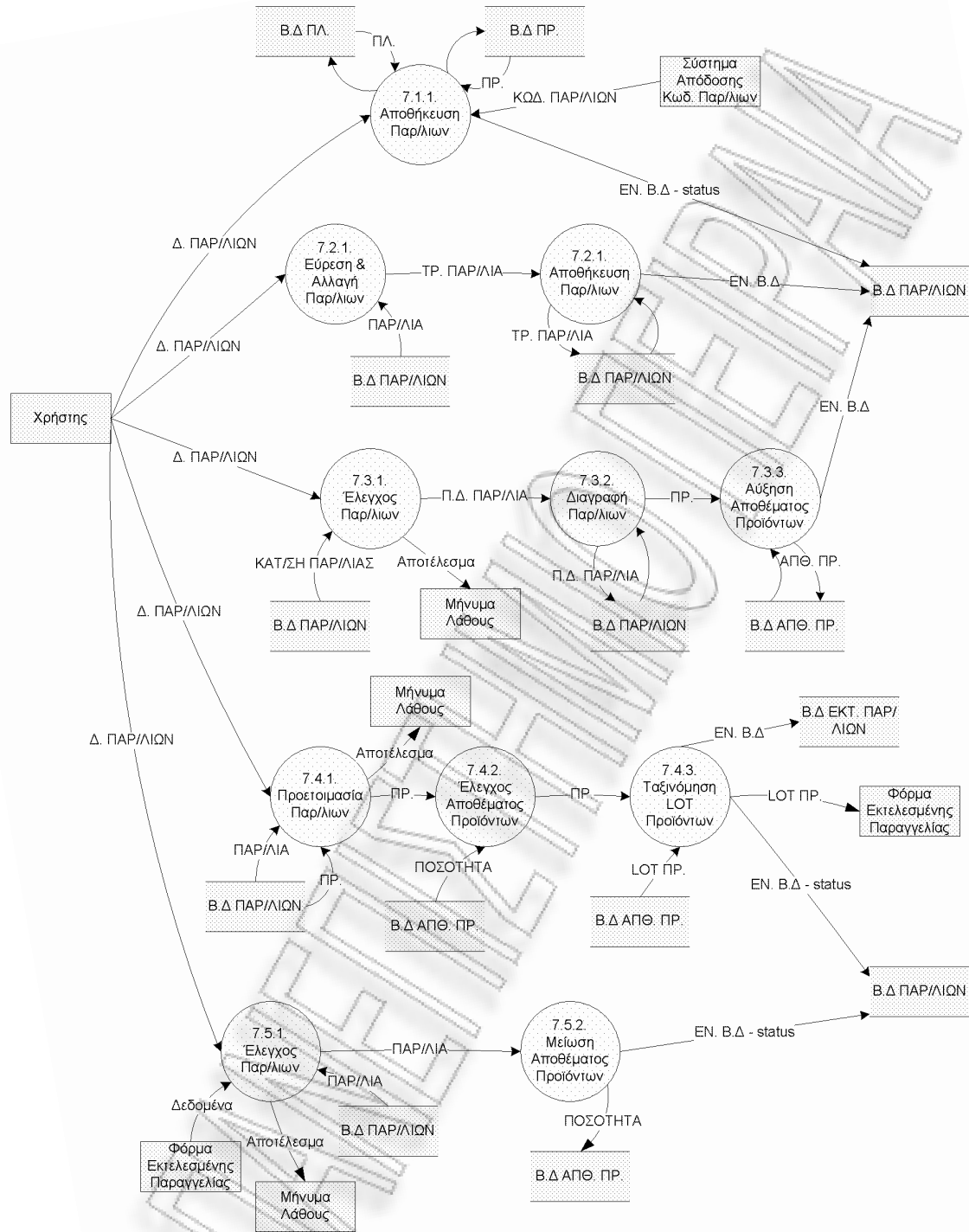
Διάγραμμα 19. 7ος Τρόπος Λειτουργίας - Επίπεδο 1

2.3.1.7.2. Επίπεδο 2



Διάγραμμα 20. 7ος Τρόπος Λειτουργίας - Επίπεδο 2

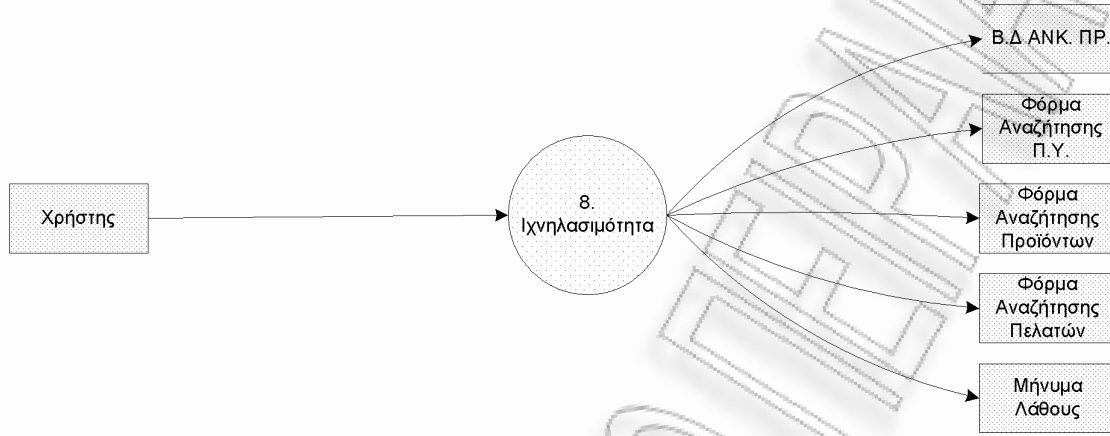
2.3.1.7.3. Επίπεδο 3



Διάγραμμα 21. 7ος Τρόπος Λειτουργίας - Επίπεδο 3

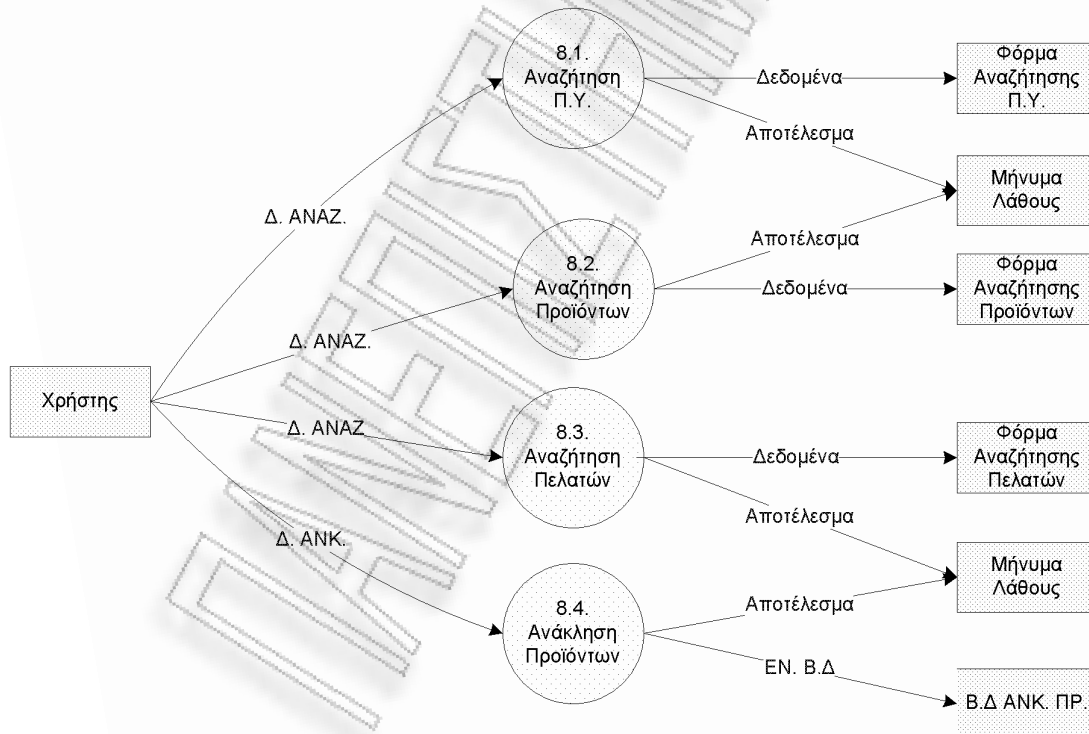
2.3.1.8. 8ος Τρόπος Λειτουργίας (Ιχνηλασιμότητα)

2.3.1.8.1. Επίπεδο 1



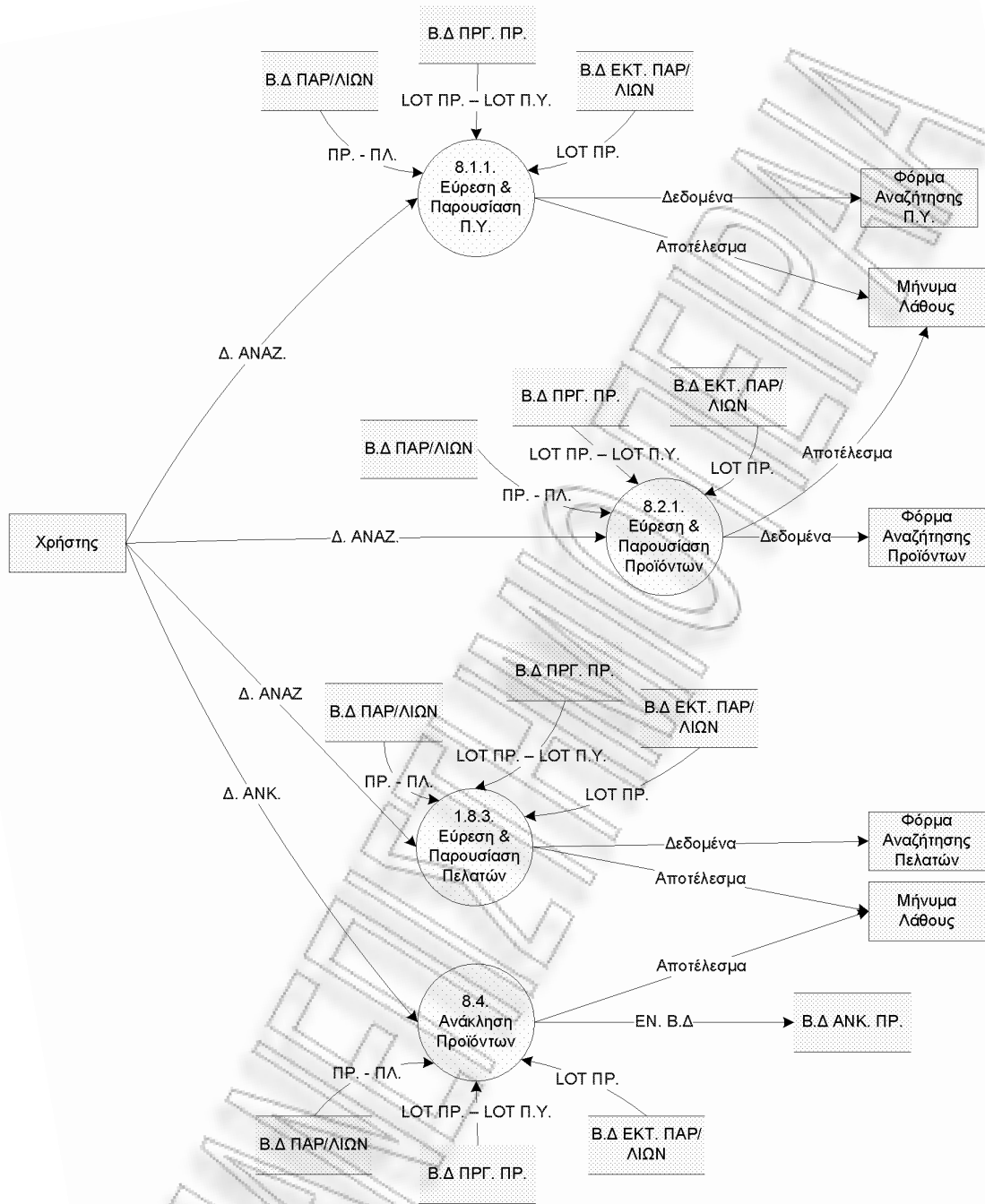
Διάγραμμα 22. 8ος Τρόπος Λειτουργίας - Επίπεδο 1

2.3.1.8.2. Επίπεδο 2



Διάγραμμα 23. 8ος Τρόπος Λειτουργίας - Επίπεδο 2

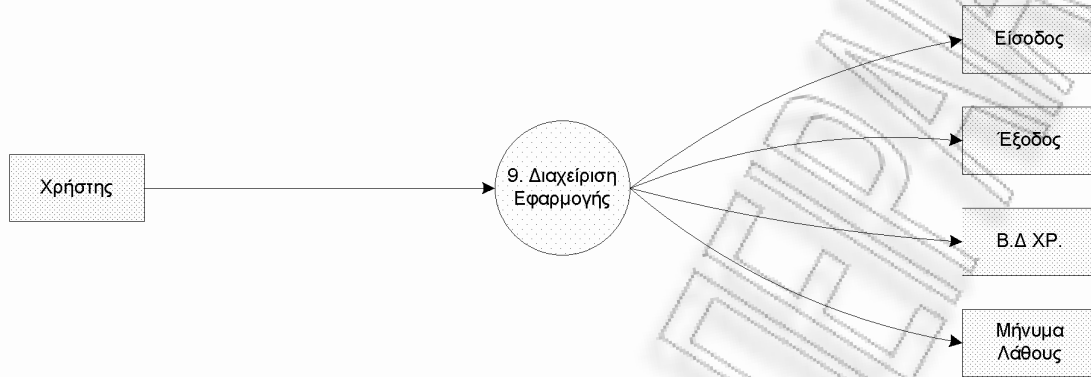
2.3.1.8.3. Επίπεδο 3



Διάγραμμα 24. 8ος Τρόπος Λειτουργίας - Επίπεδο 3

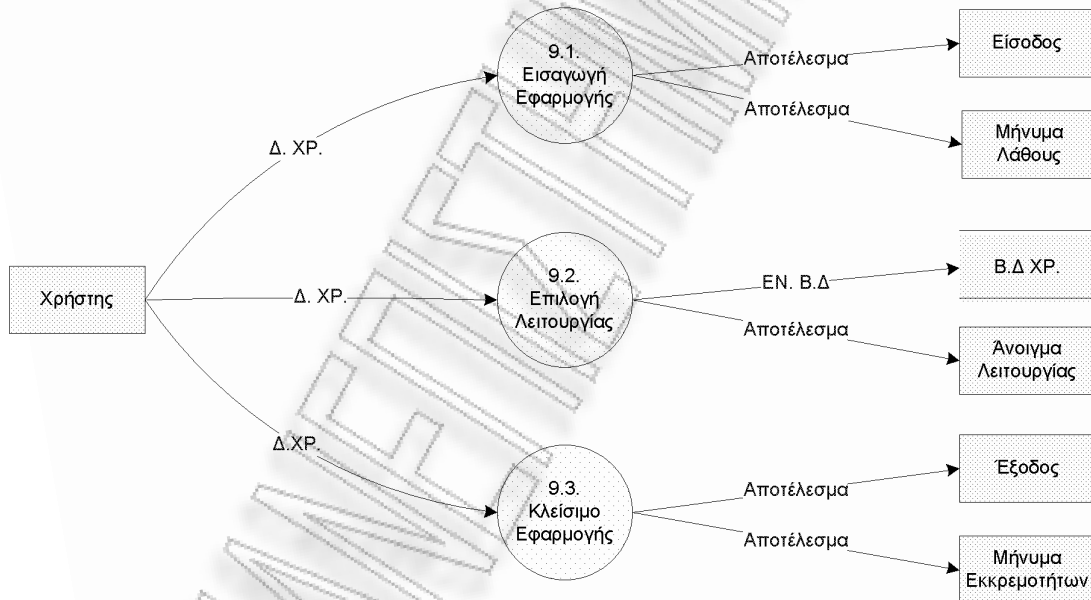
2.3.1.9. 9ος Τρόπος Λειτουργίας (Κεντρική διαχείριση εφαρμογής)

2.3.1.9.1. Επίπεδο 1



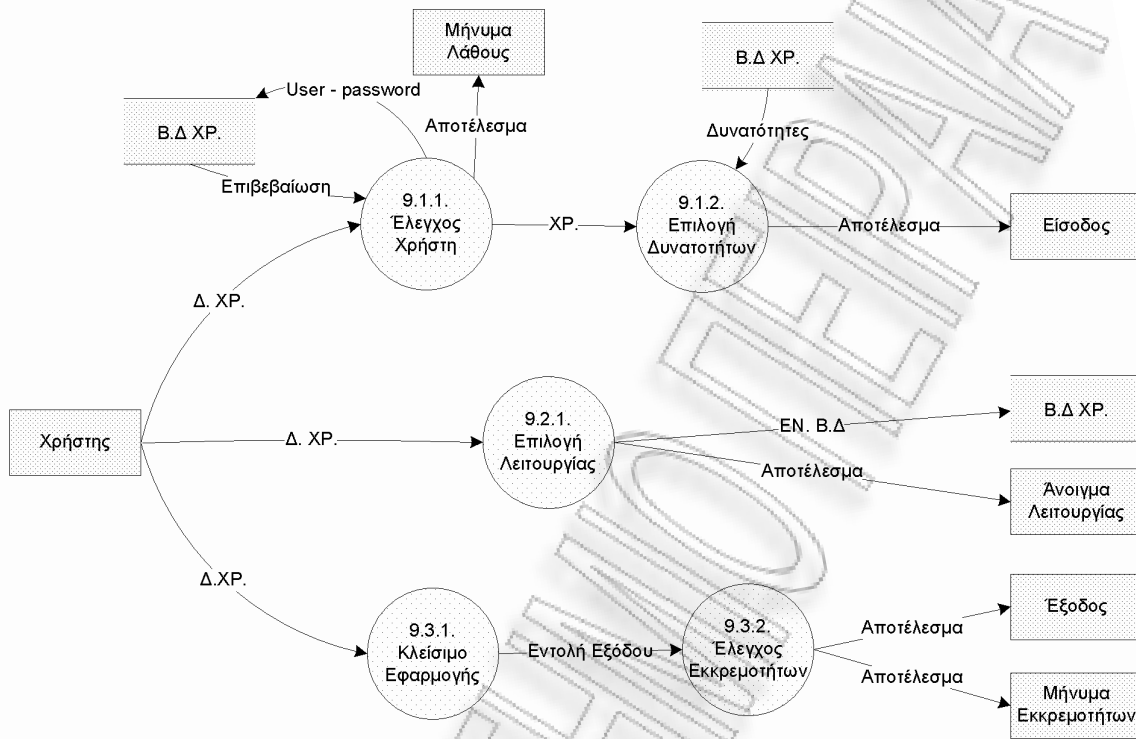
Διάγραμμα 25. 9ος Τρόπος Λειτουργίας - Επίπεδο 1

2.3.1.9.2. Επίπεδο 2



Διάγραμμα 26. 9ος Τρόπος Λειτουργίας - Επίπεδο 2

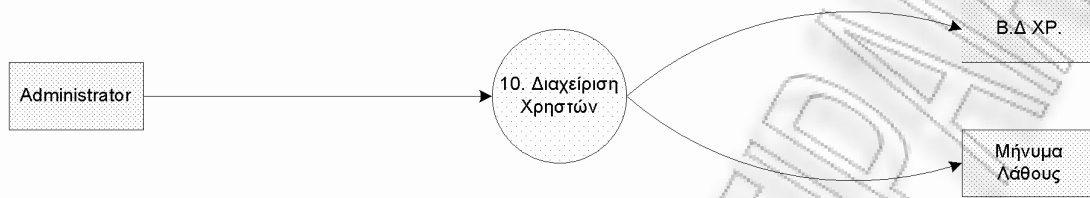
2.3.1.9.3. Επίπεδο 3



Διάγραμμα 27. 9ος Τρόπος Λειτουργίας - Επίπεδο 3

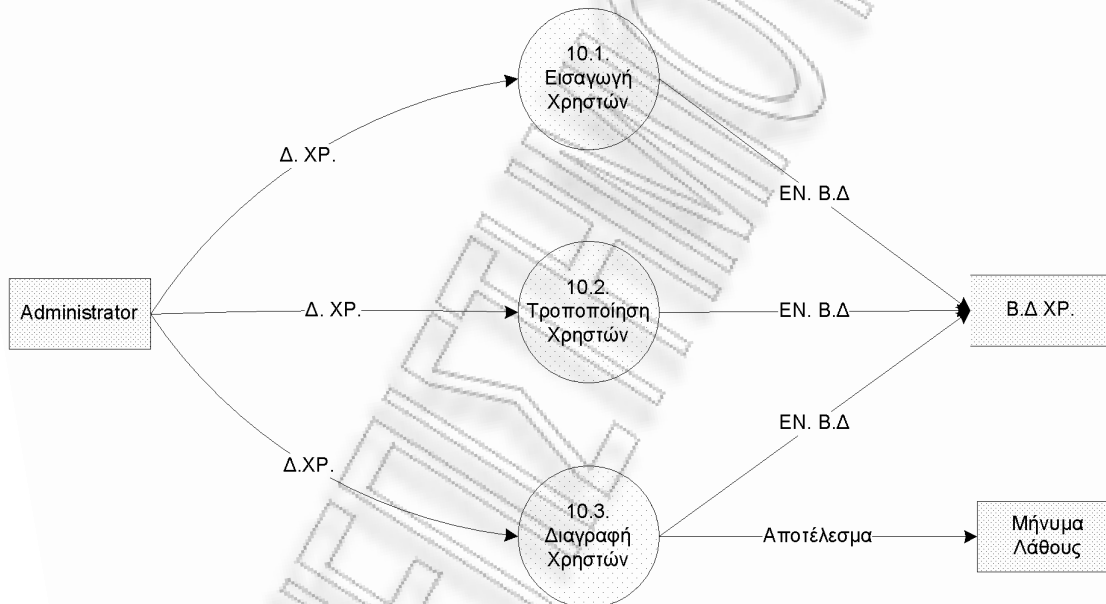
2.3.1.10. 10^{ος} Τρόπος Λειτουργίας (Διαχείριση χρηστών)

2.3.1.10.1. Επίπεδο 1



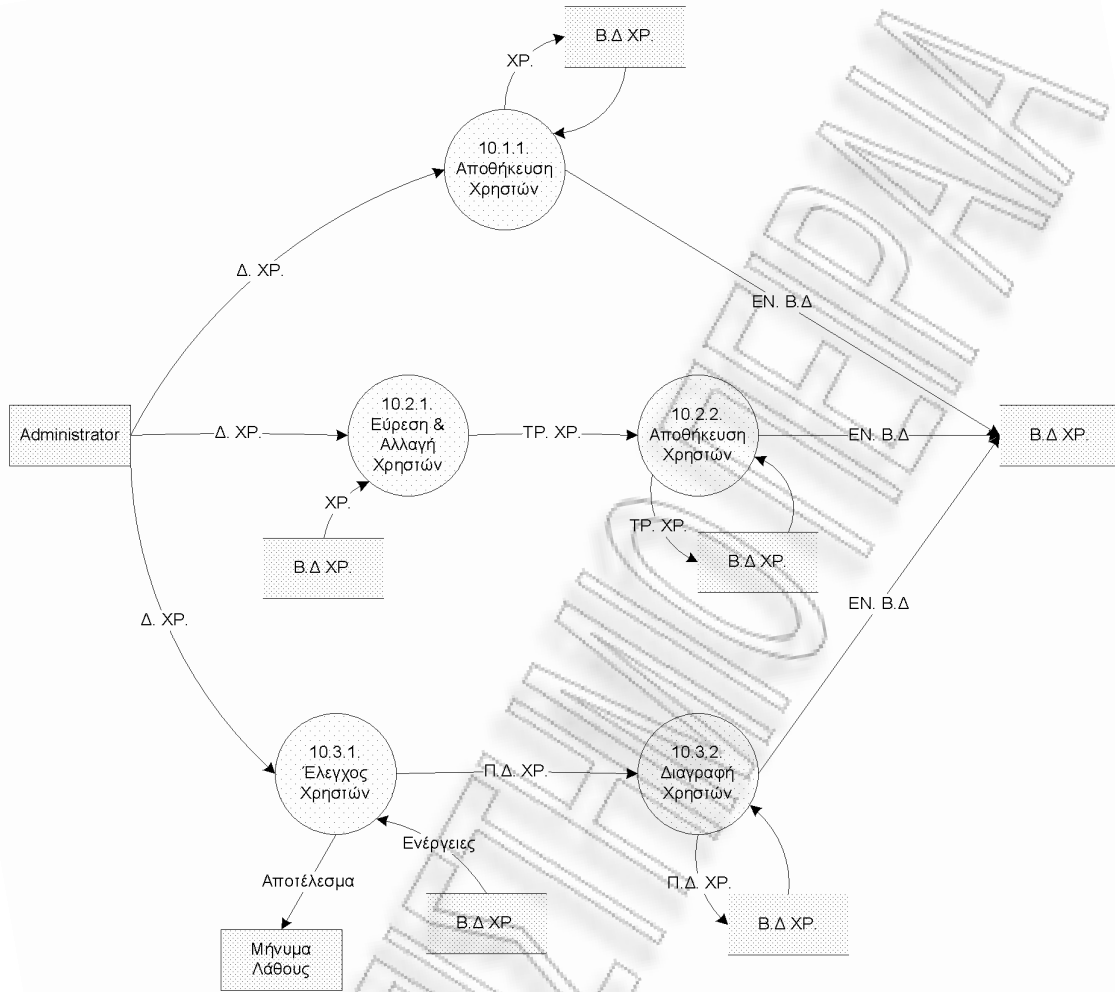
Διάγραμμα 28. 10ος Τρόπος Λειτουργίας - Επίπεδο 1

2.3.1.10.2. Επίπεδο 2



Διάγραμμα 29. 10ος Τρόπος Λειτουργίας - Επίπεδο 2

2.3.1.10.3. Επίπεδο 3

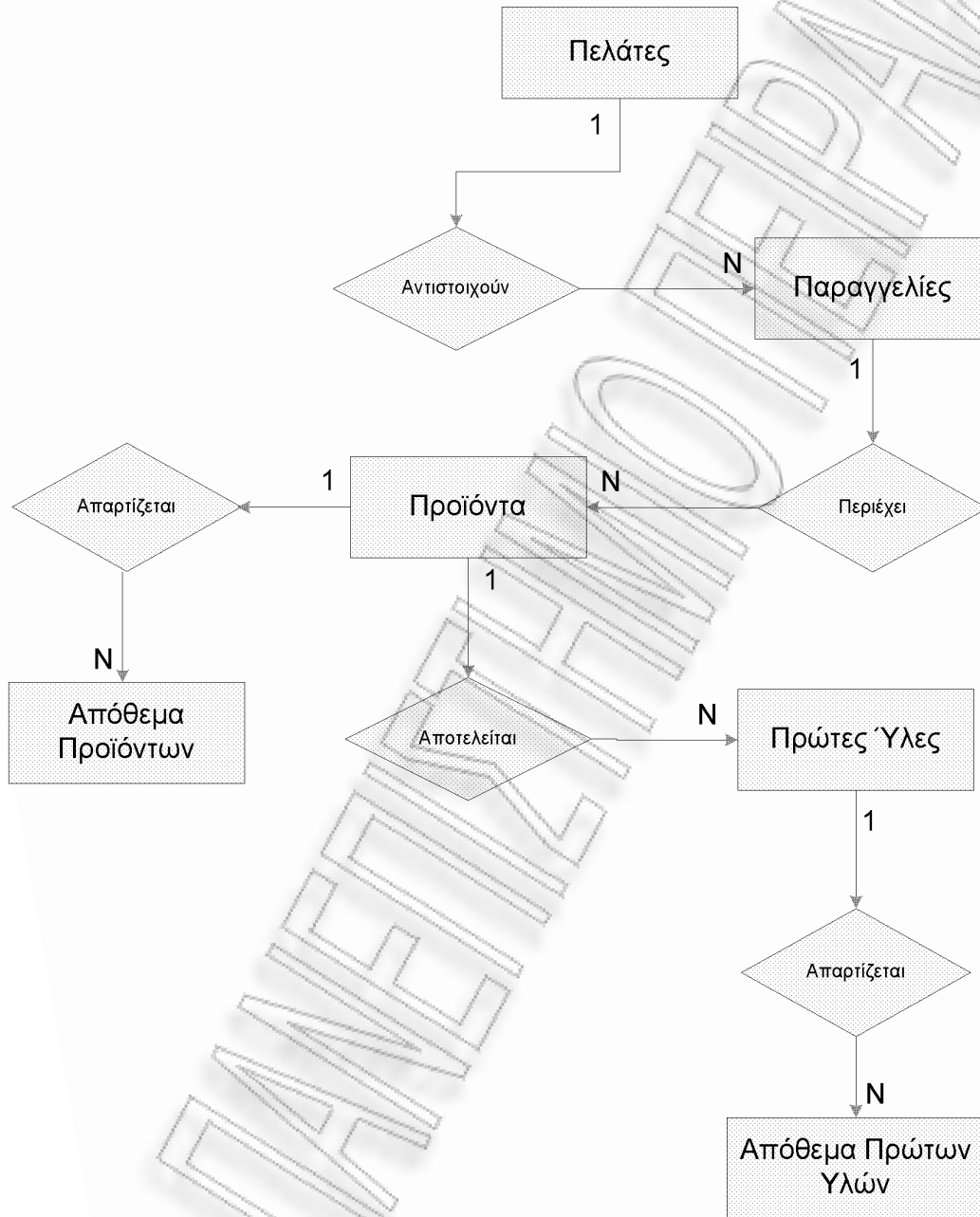


Διάγραμμα 30. 10ος Τρόπος Λειτουργίας - Επίπεδο 3

2.3.1.11. 11ος Τρόπος λειτουργίας (Συντήρηση)

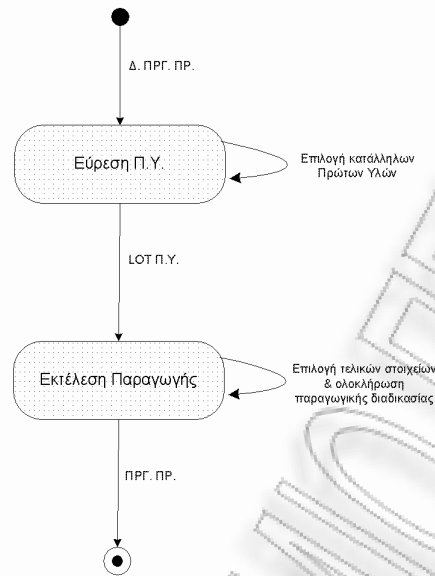
Ο 11ος τρόπος λειτουργίας δεν περιγράφεται στο παρών έγγραφο.

2.3.2. Διάγραμμα οντοτήτων – συσχετίσεων

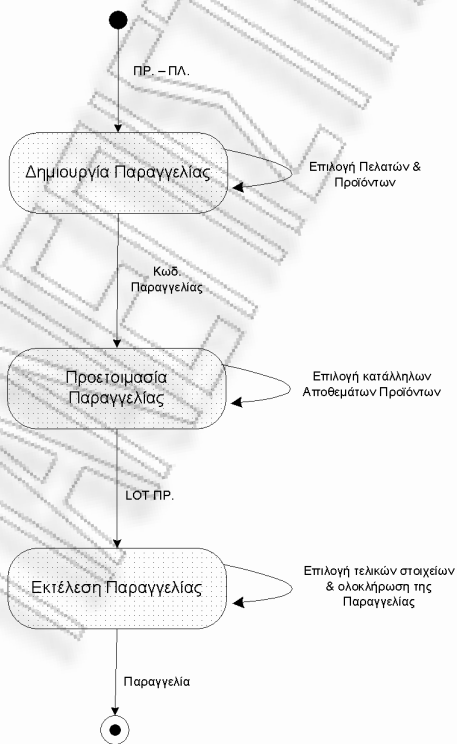


Διάγραμμα 31. Οντοτήτων - Συσχετίσεων

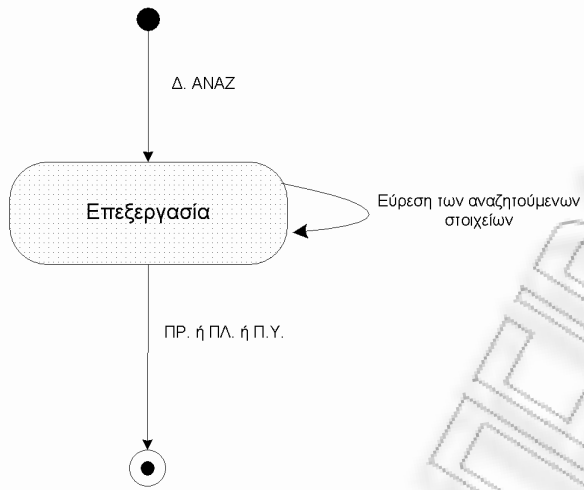
2.3.3. Διαγράμματα μετάβασης καταστάσεων



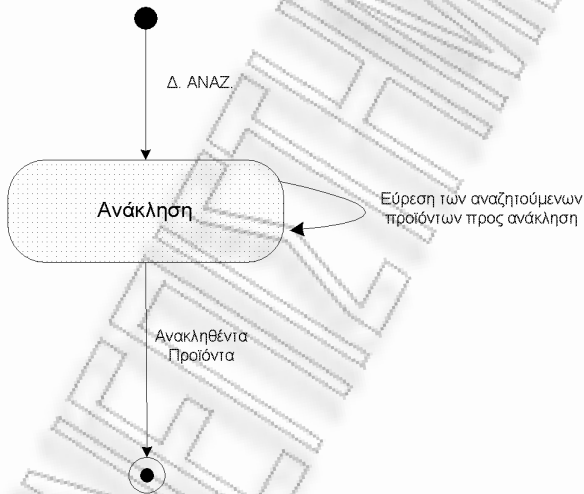
Διάγραμμα 32. Μετάβαση καταστάσεων (Παραγωγή)



Διάγραμμα 33. Μετάβαση καταστάσεων (Παραγγελία)



Διάγραμμα 34. Μετάβαση καταστάσεων (Εύρεση ΠΡ., ΠΛ., Π.Υ.)



Διάγραμμα 35. Μετάβαση καταστάσεων (Ανάκληση)

2.3.4. Λεξικό δεδομένων

Όνομασία	Άλλες ονομασίες	Που	Πως	Περιεχόμενα	Όρια
Βάση Δεδομένων	Β.Δ	Γενικός χαρακτηρισμός για όλες τις Βάσεις Δεδομένων	Ως στοιχείο εισόδου και εξόδου	Προϊόντα, Πρώτες Ύλες, Πελάτες, Αποθέματα, Παραγγελίες, κ.α.	
Πρώτες Ύλες	Π.Υ.	Αποθήκευση Π.Υ., Εύρεση & Αλλαγή Π.Υ., Έλεγχος Π.Υ., Αποθήκευση ΑΠΘ.Π.Υ., Εύρεση ποσοτήτων Π.Υ., Μείωση ΑΠΘ. Π.Υ.	Ως στοιχείο εισόδου και εξόδου	Στοιχεία Πρώτων Υλών	
LOT Πρώτων Υλών	LOT Π.Υ.	Έλεγχος LOT Π.Υ., Εκτέλεση Παραγωγής, Φόρμα Παραγωγής, Εύρεση & Παρουσίαση Π.Υ., Εύρεση & Παρουσίαση Προϊόντων, Εύρεση & Παρουσίαση Πελατών, Ανάκληση Προϊόντων	Ως στοιχείο εισόδου & εξόδου	Αριθμητικά Ψηφία	
Βάση Δεδομένων Πρώτων Υλών	Β.Δ. Π.Υ.	Β.Δ. Π.Υ.	Ως πίνακας	Πίνακας Πρώτων Υλών	
Ενημερωμένη Βάση Δεδομένων	ΕΝ. Β.Δ.	Σε όλους τους μετασχηματισμούς που προκύπτουν αλλαγές στις Β.Δ	Ως στοιχείο εξόδου	Πίνακας εγγραφών πεδίων Β.Δ	
Τροποποιημένες Πρώτες Ύλες	ΤΡ. Π.Υ.	Αποθήκευση Π.Υ.	Ως στοιχείο εισόδου και εξόδου	Νέα στοιχεία Πρώτων Υλών	
Απόθεμα Προϊόντων	ΑΠΘ. ΠΡ.	Έλεγχος ΠΡ., Αποθήκευση ΑΠΘ. ΠΡ., Εύρεση & Αλλαγή ΑΠΘ. ΠΡ., Έλεγχος ΑΠΘ. ΠΡ., Αύξηση ΑΠΘ. ΠΡ.	Ως στοιχείο εισόδου και εξόδου	Στοιχεία Αποθέματος Προϊόντων	
Προϊόντα	ΠΡ.	Έλεγχος Π.Υ., Αποθήκευση ΠΡ., Εύρεση & Αλλαγή ΠΡ., Έλεγχος ΠΡ., Αποθήκευση Παρ/λιων, Προετοιμασία Παρ/λιων, Έλεγχος ΑΠΘ. ΠΡ., Ταξινόμηση LOT ΠΡ., Εύρεση & Παρουσίαση Π.Υ., Εύρεση & Παρουσίαση ΠΡ., Εύρεση & Παρουσίαση ΠΛ., Ανάκληση Προϊόντων	Ως στοιχείο εισόδου και εξόδου	Στοιχεία Προϊόντων	

LOT Προϊόντων	LOT ΠΡ.	Έλεγχος LOT Π.Υ., Εκτέλεση Παραγωγής, Φόρμα Παραγωγής, Εύρεση & Παρουσίαση Π.Υ., Εύρεση & Παρουσίαση Προϊόντων, Εύρεση & Παρουσίαση Πελατών, Ανάκληση Προϊόντων, Ταξινόμηση LOT Προϊόντων	Ως στοιχείο εισόδου και εξόδου	Αριθμητικά Ψηφία	
Προς Διαγραφή Πρώτες Ύλες	Π.Δ. Π.Υ.	Διαγραφή Π.Υ.	Ως στοιχείο εισόδου και εξόδου	Στοιχεία διαγραφόμενων Πρώτων Υλών	
Βάση Δεδομένων Προϊόντων	Β.Δ. ΠΡ.	Β.Δ. ΠΡ.	Ως πίνακας	Πίνακας Προϊόντων	
Δεδομένα Προϊόντων	Δ.ΠΡ.	Αποθήκευση ΠΡ., Εύρεση & Αλλαγή ΠΡ., Έλεγχος ΠΡ.	Ως στοιχείο εισόδου	Στοιχεία Προϊόντων εισαγόμενα από το χρήστη	
Τροποποιημένα Προϊόντα	ΤΡ. ΠΡ.	Αποθήκευση ΠΡ.	Ως στοιχείο εισόδου και εξόδου	Τροποποιημένα στοιχεία Προϊόντων	
Παραγόμενα Προϊόντα	ΠΡΓ. ΠΡ.	Έλεγχος ΠΡ., Έλεγχος ΑΠΘ. Π.Υ., Αύξηση ΑΠΘ. ΠΡ.	Ως στοιχείο εισόδου και εξόδου	Στοιχεία Παραγόμενων Προϊόντων	
Βάση Δεδομένων Παραγόμενων Προϊόντων	Β.Δ. ΠΡΓ. ΠΡ.	Β.Δ. ΠΡΓ. ΠΡ.	Ως πίνακας	Πίνακας Παραγόμενων Προϊόντων	
Προς Διαγραφή Προϊόντα	Π.Δ. ΠΡ.	Διαγραφή ΠΡ.	Ως στοιχείο εισόδου και εξόδου	Στοιχεία διαγραφόμενων Προϊόντων	
Βάση Δεδομένων Αποθέματος Προϊόντων	Β.Δ. ΑΠΘ. ΠΡ.	Β.Δ. ΑΠΘ. ΠΡ.	Ως πίνακας	Πίνακας Αποθέματος Προϊόντων	
Βάση Δεδομένων Παραγγελιών	Β.Δ. ΠΑΡ/ΛΙΩΝ	Β.Δ. ΠΑΡ/ΛΙΩΝ	Ως πίνακας	Πίνακας Παραγγελιών	
Πελάτες	ΠΛ.	Αποθήκευση ΠΛ., Εύρεση & Αλλαγή ΠΛ., Έλεγχος ΠΛ., Αποθήκευση Παρ/λιων, Εύρεση & Παρουσίαση Π.Υ., Εύρεση & Παρουσίαση ΠΡ., Εύρεση & Παρουσίαση ΠΛ.	Ως στοιχείο εισόδου και εξόδου	Στοιχεία Πελατών	

Βάση Δεδομένων Πελατών	Β.Δ. ΠΛ.	Β.Δ. ΠΛ.	Ως πίνακας	Πίνακας Πελατών	
Δεδομένα Πελατών	Δ. ΠΛ.	Αποθήκευση ΠΛ., Εύρεση & Αλλαγή ΠΛ., Έλεγχος ΠΛ.	Ως στοιχείο εισόδου	Στοιχεία Πελατών εισαγόμενα από το χρήστη	
Τροποποιημένοι Πελάτες	ΤΡ. ΠΛ.	Αποθήκευση ΠΛ.	Ως στοιχείο εισόδου και εξόδου	Τροποποιημένα στοιχεία πελατών	
Προς Διαγραφή Πελάτες	Π.Δ. ΠΛ.	Διαγραφή ΠΛ.	Ως στοιχείο εισόδου και εξόδου	Στοιχεία διαγραφόμενων Πελατών	
Απόθεμα Πρώτων Υλών	ΑΠΘ. Π.Υ.	Έλεγχος Π.Υ., Αποθήκευση ΑΠΘ. Π.Υ., Εύρεση & Αλλαγή ΑΠΘ. Π.Υ., Έλεγχος ΑΠΘ. Π.Υ.	Ως στοιχείο εισόδου και εξόδου	Στοιχεία Αποθέματος Πρώτων Υλών	
Δεδομένα Αποθέματος Πρώτων Υλών	Δ. ΑΠΘ. Π.Υ.	Αποθήκευση ΑΠΘ. Π.Υ., Εύρεση & Αλλαγή ΑΠΘ. Π.Υ., Έλεγχος ΑΠΘ. Π.Υ.	Ως στοιχείο εισόδου	Στοιχεία Αποθέματος Πρώτων Υλών εισαγόμενα από το χρήστη	
Βάση Δεδομένων Αποθέματος Πρώτων Υλών	Β.Δ. ΑΠΘ. Π.Υ.	Β.Δ. ΑΠΘ. Π.Υ.	Ως πίνακας	Πίνακας Αποθέματος Πρώτων Υλών	
Τροποποιημένο Απόθεμα Πρώτων Υλών	ΤΡ. ΑΠΘ. Π.Υ.	Αποθήκευση ΑΠΘ. Π.Υ.	Ως στοιχείο εισόδου και εξόδου	Τροποποιημένα στοιχεία Αποθέματος Πρώτων Υλών	
Προς Διαγραφή Απόθεμα Πρώτων Υλών	Π.Δ. ΑΠΘ. Π.Υ.	Διαγραφή ΑΠΘ. Π.Υ.	Ως στοιχείο εισόδου και εξόδου	Στοιχεία διαγραφόμενων Αποθεμάτων Πρώτων Υλών	
Δεδομένα Αποθέματος Προϊόντων	Δ. ΑΠΘ. ΠΡ.	Αποθήκευση ΑΠΘ. ΠΡ., Εύρεση & Αλλαγή ΑΠΘ. ΠΡ., Έλεγχος ΑΠΘ. ΠΡ.	Ως στοιχείο εισόδου	Στοιχεία Αποθέματος Προϊόντων εισαγόμενα από το χρήστη	

Τροποποιημένο Απόθεμα Προϊόντων	ΤΡ. ΑΠΘ. ΠΡ.	Αποθήκευση ΑΠΘ. ΠΡ.	Ως στοιχείο εισόδου και εξόδου	Τροποποιημένα στοιχεία Αποθέματος Προϊόντων	
Προς Διαγραφή Απόθεμα Προϊόντων	Π.Δ. ΑΠΘ. ΠΡ.	Διαγραφή ΑΠΘ. ΠΡ.	Ως στοιχείο εισόδου και εξόδου	Στοιχεία διαγραφόμενων Αποθεμάτων Προϊόντων	
Δεδομένα Παραγόμενων Προϊόντων	Δ. ΠΡΓ.ΠΡ.	Εύρεση ποσοτήτων Π.Υ., Εκτέλεση Παραγωγής	Ως στοιχείο εισόδου	Στοιχεία Παραγόμενων Προϊόντων εισαγόμενα από το χρήστη	
Δεδομένα Παραγγελιών	Δ. ΠΑΡ/ΛΙΩΝ	Αποθήκευση Παρ/λιων, Εύρεση & Αλλαγή Παρ/λιων, Έλεγχος Παρ/λιων, Προετοιμασία Παρ/λιων, Έλεγχος Παρ/λιων	Ως στοιχείο εισόδου και εξόδου	Στοιχεία Παραγγελιών εισαγόμενα από το χρήστη	
Τροποποιημένες Παραγγελίες	ΤΡ. ΠΑΡ/ΛΙΩΝ	Αποθήκευση Παρ/λιων	Ως στοιχείο εισόδου και εξόδου	Τροποποιημένα στοιχεία Παραγγελιών	
Προς Διαγραφή Παραγγελίες	Π.Δ. ΠΑΡ/ΛΙΕΣ	Διαγραφή Παρ/λιων	Ως στοιχείο εισόδου και εξόδου	Στοιχεία διαγραφόμενων Παραγγελιών	
Βάση Δεδομένων Εκτελεσμένων Παραγγελιών	Β.Δ. ΕΚΤ. ΠΑΡ/ΛΙΩΝ	Β.Δ. ΕΚΤ. ΠΑΡ/ΛΙΩΝ	Ως πίνακας	Πίνακας Εκτελεσμένων Παραγγελιών	
Βάση Δεδομένων Ανακληθέντων Παρτίδων	Β.Δ. ΑΝΚ. ΠΡ.	Β.Δ. ΑΝΚ. ΠΡ.	Ως πίνακας	Πίνακας Ανακληθέντων Παρτίδων	
Δεδομένα Αναζήτησης	Δ. ΑΝΑΖ.	Εύρεση & Παρουσίαση Π.Υ., Εύρεση & Παρουσίαση ΠΡ., Εύρεση & Παρουσίαση ΠΛ., Ανάκληση Προϊόντων	Ως στοιχείο εισόδου	Στοιχεία Αναζήτησης εισαγόμενα από το χρήστη	
Δεδομένα Χρήστη	Δ. ΧΡ.	Αποθήκευση ΧΡ., Εύρεση & Αλλαγή ΧΡ., Έλεγχος ΧΡ., Επιλογή λειτουργίας, Κλείσιμο εφαρμογής	Ως στοιχείο εισόδου	Στοιχεία Χρηστών εισαγόμενα από	

				τον Administrator - Χρήστη	
Τροποποιημένοι Χρήστες	ΤΡ. ΧΡ.	Αποθήκευση ΧΡ.	Ως στοιχείο εισόδου και εξόδου	Τροποποιημένα στοιχεία Χρηστών	
Προς Διαγραφή Χρήστες	Π.Δ. ΧΡ.	Διαγραφή ΧΡ.	Ως στοιχείο εισόδου και εξόδου	Στοιχεία διαγραφόμενων Χρηστών	
Βάση Δεδομένων Χρηστών	Β.Δ. ΧΡ.	Β.Δ. ΧΡ.	Ως πίνακας	Πίνακας χρηστών	

2.4. ΕΓΓΡΑΦΟ ΠΕΡΙΓΡΑΦΗΣ ΤΟΥ ΣΧΕΔΙΟΥ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ

2.4.1. Εισαγωγή

2.4.1.1. Γενική περιγραφή

Το έγγραφο αυτό αποτελεί ένα ενιαίο δομικό κέλυφος στο οποίο ενσωματώνονται όλα τα επιμέρους προϊόντα της σχεδίασης με τη μορφή διαγραμμάτων αλλά και με τη μορφή κειμένων.

2.4.1.2. Εμβέλεια

Το έγγραφο αυτό καλύπτει τα σχέδια διάταξης, την αρχιτεκτονική σχεδίαση, τη σχεδίαση διαπροσωπειών, τη λεπτομερή σχεδίαση των μονάδων του λογισμικού, καθώς και τη σχεδίαση σχεσιακών βάσεων δεδομένων.

2.4.1.3. Ορισμοί και ακρωνύμια

Δεν χρησιμοποιούνται ορισμοί και έννοιες ειδικού σκοπού. Οι συντομεύσεις που υπάρχουν στα διαγράμματα περιέχονται στο λεξικό δεδομένων.

2.4.1.4. Αναφορές

Δεν χρειάστηκαν ιδιαίτερες αναφορές για τη παράσταση του εγγράφου.

2.4.2. Περιγραφή αποσύνθεσης του λογισμικού

2.4.2.1. Αποσύνθεση σε μονάδες

Οι μονάδες του λογισμικού είναι οι παρακάτω:

1. Διαχείριση Π.Υ.

1.1. Εισαγωγή Π.Υ.

1.1.1. Αποθήκευση Π.Υ.

1.2. Τροποποίηση Π.Υ.

1.2.1. Εύρεση και αλλαγή Π.Υ.

1.2.2. Αποθήκευση Π.Υ.

1.3. Διαγραφή Π.Υ.

1.3.1. Έλεγχος Π.Υ.

1.3.2. Διαγραφή Π.Υ.

2. Διαχείριση ΠΡ.

2.1. Εισαγωγή ΠΡ.

2.1.1. Αποθήκευση ΠΡ.

2.2. Τροποποίηση ΠΡ.

2.2.1. Εύρεση και αλλαγή ΠΡ.

2.2.2. Αποθήκευση ΠΡ.

2.3. Διαγραφή ΠΡ.

2.3.1. Έλεγχος ΠΡ.

2.3.2. Διαγραφή ΠΡ.

3. Διαχείριση ΠΛ.

3.1. Εισαγωγή ΠΛ.

3.1.1. Αποθήκευση ΠΛ.

3.2. Τροποποίηση ΠΛ.

3.2.1. Εύρεση και αλλαγή ΠΛ.

3.2.2 Αποθήκευση ΠΛ.

3.3. Διαγραφή ΠΛ.

3.3.1. Έλεγχος ΠΛ.

3.3.2. Διαγραφή ΠΛ.

4. Διαχείριση ΑΠΘ.Π.Υ.

4.1. Εισαγωγή ΑΠΘ.Π.Υ.

4.1.1. Αποθήκευση ΑΠΘ.Π.Υ.

4.2. Τροποποίηση ΑΠΘ.Π.Υ.

4.2.1. Εύρεση & Αλλαγή ΑΠΘ.Π.Υ.

4.2.2. Αποθήκευση ΑΠΘ.Π.Υ.

4.3. Διαγραφή ΑΠΘ.Π.Υ.

4.3.1. Έλεγχος ΑΠΘ.Π.Υ.

4.3.2. Διαγραφή ΑΠΘ.Π.Υ.

5. Διαχείριση ΑΠΘ.ΠΡ.

5.1. Εισαγωγή ΑΠΘ. ΠΡ.

5.1.1 Αποθήκευση ΑΠΘ. ΠΡ.

5.2. Τροποποίηση ΑΠΘ. ΠΡ.

5.2.1. Εύρεση & Αλλαγή ΑΠΘ. ΠΡ.

5.2.2. Αποθήκευση ΑΠΘ. ΠΡ.

5.3. Διαγραφή ΑΠΘ. ΠΡ.

5.3.1 Έλεγχος ΑΠΘ. ΠΡ.

5.3.2 Διαγραφή ΑΠΘ. ΠΡ.

6. Παραγωγή προϊόντων

6.1. Προετοιμασία παραγωγής

6.1.1. Εύρεση ποσοτήτων Π.Υ.

6.1.2. Έλεγχος LOT Π.Υ.

6.2. Υλοποίηση παραγωγής

6.2.1. Εκτέλεση παραγωγής

6.2.2. Αύξηση αποθέματος ΠΡ.

6.2.3. Μείωση αποθέματος Π.Υ.

7. Παραγγελία ΠΡ.

7.1 Εισαγωγή παραγγελιών

7.1.1. Αποθήκευση παραγγελιών

7.2. Τροποποίηση παραγγελιών

7.2.1. Εύρεση & Αλλαγή παραγγελιών

7.2.2. Αποθήκευση παραγγελιών

7.3. Διαγραφή παραγγελιών

7.3.1. Έλεγχος παραγγελιών

7.3.2. Διαγραφή παραγγελιών

7.3.3. Αύξηση αποθέματος ΠΡ.

7.4. Εκτέλεση παραγγελιών

7.4.1. Προετοιμασία παραγγελιών

7.4.2. Έλεγχος αποθέματος ΠΡ.

7.4.3. Ταξινόμηση LOT ΠΡ.

7.5. Ολοκλήρωση παραγγελιών

7.5.1. Έλεγχος Παραγγελιών

7.5.2. Μείωση αποθέματος ΠΡ.

8. Ικνηλασιμότητα

8.1. Αναζήτηση Π.Υ.

8.1.1. Εύρεση και παρουσίαση Π.Υ.

8.2. Αναζήτηση ΠΡ.

8.2.1. Εύρεση και παρουσίαση ΠΡ.

8.3. Αναζήτηση πελατών

8.3.1. Εύρεση και παρουσίαση πελατών

8.4. Ανάκληση ΠΡ.

8.4.1. Ανάκληση ΠΡ.

9. Διαχείριση εφαρμογής

9.1. Έναρξη Εφαρμογής

9.1.1. Έλεγχος χρήστη

9.1.2. Επιλογή δυνατοτήτων

9.2. Επιλογή λειτουργίας

9.2.1. Επιλογή λειτουργίας

9.3. Κλείσιμο εφαρμογής

9.3.1. Κλείσιμο εφαρμογής

9.3.2. Έλεγχος εκκρεμοτήτων

10. Διαχείριση χρηστών

10.1. Εισαγωγή χρηστών

10.1.1. Αποθήκευση χρηστών

10.2. Τροποποίηση χρηστών

10.2.1. Εύρεση και αλλαγή χρηστών

10.2.2. Αποθήκευση χρηστών

10.3. Διαγραφή χρηστών

10.3.1. Έλεγχος χρηστών

10.3.2. Διαγραφή χρηστών

2.4.2.1.1. Περιγραφή μονάδας 1

1.1.1. Δέχεται τα δεδομένα των πρώτων υλών από τον χρήστη και τα αποθηκεύει στην αντίστοιχη βάση δεδομένων.

2.4.2.1.2. Περιγραφή μονάδας 2

1.2.1. Δέχεται τα δεδομένα των πρώτων υλών για αλλαγή από τον χρήστη και από την αντίστοιχη βάση δεδομένων και στέλνει τα τροποποιημένα δεδομένα προς αποθήκευση.

2.4.2.1.3. Περιγραφή μονάδας 3

1.2.2. Δέχεται τα τροποποιημένα δεδομένα των πρώτων υλών και τα αποθηκεύει στην αντίστοιχη βάση δεδομένων.

2.4.2.1.4. Περιγραφή μονάδας 4

1.3.1. Δέχεται δεδομένα πρώτων υλών, ελέγχει εάν υπάρχουν προϊόντα που σχετίζονται με αυτές τις πρώτες ύλες ή υπάρχουν αποθέματα αυτών των υλών και στέλνει δεδομένα προς διαγραφή.

2.4.2.1.5. Περιγραφή μονάδας 5

1.3.2. Δέχεται δεδομένα πρώτων υλών προς διαγραφή και τα στέλνει στην αντίστοιχη βάση.

2.4.2.1.6. Περιγραφή μονάδας 6.

2.1.1. Δέχεται τα δεδομένα των προϊόντων από τον χρήστη και τα αποθηκεύει στην αντίστοιχη βάση δεδομένων.

2.4.2.1.7. Περιγραφή μονάδας 7.

2.1.2. Δέχεται τα δεδομένα των προϊόντων για αλλαγή από τον χρήστη και από την αντίστοιχη βάση δεδομένων και στέλνει τα τροποποιημένα δεδομένα προς αποθήκευση.

2.4.2.1.8. Περιγραφή μονάδας 8.

2.2.2. Δέχεται τα τροποποιημένα δεδομένα των προϊόντων και τα αποθηκεύει στην αντίστοιχη βάση δεδομένων.

2.4.2.1.9. Περιγραφή μονάδας 9.

2.3.1. Δέχεται δεδομένα προϊόντων, ελέγχει εάν υπάρχουν παραγγελίες που σχετίζονται με τα προϊόντα ή υπάρχουν αποθέματα αυτών των προϊόντων και στέλνει δεδομένα προς διαγραφή.

2.4.2.1.10. Περιγραφή μονάδας 10.

2.3.2. Δέχεται δεδομένα προϊόντων προς διαγραφή και τα στέλνει στην αντίστοιχη βάση.

2.4.2.1.11. Περιγραφή μονάδας 11.

3.1.1. Δέχεται τα δεδομένα των πελατών από τον χρήστη και τα αποθηκεύει στην αντίστοιχη βάση δεδομένων.

2.4.2.1.12. Περιγραφή μονάδας 12.

3.2.1. Δέχεται τα δεδομένα των πελατών για αλλαγή από τον χρήστη και από την αντίστοιχη βάση δεδομένων και στέλνει τα τροποποιημένα δεδομένα προς αποθήκευση.

2.4.2.1.13. Περιγραφή μονάδας 13.

3.2.2. Δέχεται τα τροποποιημένα δεδομένα των πελατών από τον χρήστη και τα αποθηκεύει στην αντίστοιχη βάση δεδομένων.

2.4.2.1.14. Περιγραφή μονάδας 14.

3.3.1. Δέχεται δεδομένα πελατών, ελέγχει εάν υπάρχουν παραγγελίες που σχετίζονται με αυτούς τους πελάτες και στέλνει δεδομένα προς διαγραφή.

2.4.2.1.15. Περιγραφή μονάδας 15.

3.3.2. Δέχεται δεδομένα πελατών προς διαγραφή και τα στέλνει στην αντίστοιχη βάση.

2.4.2.1.16. Περιγραφή μονάδας 16.

4.1.1. Δέχεται τα δεδομένα αποθεμάτων πρώτων υλών και τα αποθηκεύει στην αντίστοιχη βάση δεδομένων.

2.4.2.1.17. Περιγραφή μονάδας 17

4.2.1. Δέχεται τα δεδομένα αποθεμάτων πρώτων υλών για αλλαγή από τον χρήστη και από την αντίστοιχη βάση δεδομένων και στέλνει τα τροποποιημένα δεδομένα προς αποθήκευση.

2.4.2.1.18. Περιγραφή μονάδας 18.

4.2.2. Δέχεται τα τροποποιημένα δεδομένα αποθεμάτων πρώτων υλών και τα αποθηκεύει στην αντίστοιχη βάση δεδομένων.

2.4.2.1.19. Περιγραφή μονάδας 19.

4.3.1. Δέχεται δεδομένα αποθεμάτων πρώτων υλών, ελέγχει εάν υπάρχει αναμονή παραγωγής που σχετίζεται με αυτά τα αποθέματα πρώτων υλών και στέλνει δεδομένα προς διαγραφή.

2.4.2.1.20. Περιγραφή μονάδας 20.

4.3.2. Δέχεται δεδομένα αποθεμάτων πρώτων υλών προς διαγραφή και τα στέλνει στην αντίστοιχη βάση.

2.4.2.1.21. Περιγραφή μονάδας 21.

5.1.1. Δέχεται τα δεδομένα αποθεμάτων προϊόντων και τα αποθηκεύει στην αντίστοιχη βάση δεδομένων.

2.4.2.1.22. Περιγραφή μονάδας 22.

5.2.1. Δέχεται τα δεδομένα αποθεμάτων προϊόντων για αλλαγή από τον χρήστη και από την αντίστοιχη βάση δεδομένων και στέλνει τα τροποποιημένα δεδομένα προς αποθήκευση.

2.4.2.1.23. Περιγραφή μονάδας 23.

5.2.2. Δέχεται τα τροποποιημένα δεδομένα αποθεμάτων προϊόντων και τα αποθηκεύει στην αντίστοιχη βάση δεδομένων.

2.4.2.1.24. Περιγραφή μονάδας 24.

5.3.1. Δέχεται δεδομένα αποθεμάτων προϊόντων, ελέγχει εάν υπάρχει εκκρεμείς παραγγελίες που σχετίζονται με αυτά τα αποθέματα προϊόντων και στέλνει δεδομένα προς διαγραφή.

2.4.2.1.25. Περιγραφή μονάδας 25.

5.3.2. Δέχεται δεδομένα αποθεμάτων προϊόντων προς διαγραφή και τα στέλνει στην αντίστοιχη βάση.

2.4.2.1.26. Περιγραφή μονάδας 26.

6.1.1. Δέχεται δεδομένα παραγωγής από τον χρήστη και δεδομένα από την βάση δεδομένων προϊόντων και στέλνει δεδομένα ποσοτήτων πρώτων υλών.

2.4.2.1.27. Περιγραφή μονάδας 27.

6.1.2. Δέχεται δεδομένα ποσοτήτων πρώτων υλών και εξάγει δεδομένα πρώτων υλών προς τον χρήστη.

2.4.2.1.28. Περιγραφή μονάδας 28.

6.2.1. Δέχεται δεδομένα παραγωγής και δεδομένα πρώτων υλών (ποσότητες και LOT) και στέλνει δεδομένα πρώτων υλών και παραγόμενων προϊόντων.

2.4.2.1.29. Περιγραφή μονάδας 29.

6.2.2. Δέχεται δεδομένα παραγόμενων προϊόντων και στέλνει δεδομένα αποθεμάτων προϊόντων στην αντίστοιχη βάση.

2.4.2.1.30. Περιγραφή μονάδας 30.

6.2.3. Δέχεται δεδομένα αποθεμάτων πρώτων υλών και στέλνει δεδομένα αποθεμάτων πρώτων υλών στην αντίστοιχη βάση.

2.4.2.1.31. Περιγραφή μονάδας 31

7.1.1. Δέχεται τα δεδομένα παραγγελιών από τον χρήστη, δεδομένα από το σύστημα απόδοσης κωδ. παραγγελιών και τα αποθηκεύει στην αντίστοιχη βάση δεδομένων.

2.4.2.1.32. Περιγραφή μονάδας 32

7.2.1. Δέχεται τα δεδομένα των παραγγελιών για αλλαγή από τον χρήστη και από την αντίστοιχη βάση δεδομένων και στέλνει τα τροποποιημένα δεδομένα προς αποθήκευση.

2.4.2.1.33. Περιγραφή μονάδας 33

7.2.2. Δέχεται τα τροποποιημένα δεδομένα των παραγγελιών και τα αποθηκεύει στην αντίστοιχη βάση δεδομένων.

2.4.2.1.34. Περιγραφή μονάδας 34

7.3.1. Δέχεται δεδομένα παραγγελιών, ελέγχει την κατάσταση της παραγγελίας και στέλνει δεδομένα προς διαγραφή.

2.4.2.1.35. Περιγραφή μονάδας 35

7.3.2. Δέχεται δεδομένα παραγγελιών προς διαγραφή και τα στέλνει στην αντίστοιχη βάση.

2.4.2.1.36. Περιγραφή μονάδας 36.

7.3.3. Δέχεται δεδομένα παραγγελιών προς διαγραφή και δεδομένα αποθεμάτων προϊόντων και στέλνει τροποποιημένα δεδομένα αποθεμάτων προϊόντων στην αντίστοιχη βάση.

2.4.2.1.37. Περιγραφή μονάδας 37.

7.4.1. Δέχεται δεδομένα για εύρεση παραγγελίας και στέλνει όλα τα αντίστοιχα δεδομένα παραγγελιών.

2.4.2.1.38. Περιγραφή μονάδας 38.

7.4.2. Δέχεται δεδομένα παραγγελιών και αποθεμάτων προϊόντων (ποσότητες) και στέλνει δεδομένα παραγγελιών.

2.4.2.1.39. Περιγραφή μονάδας 39.

7.4.3. Δέχεται δεδομένα παραγγελιών και αποθεμάτων προϊόντων (LOT numbers) και στέλνει αντίστοιχα δεδομένα σε χρήστη και βάση δεδομένων.

2.4.2.1.40. Περιγραφή μονάδας 40.

7.5.1. Δέχεται δεδομένα παραγγελιών και στέλνει τροποποιημένα δεδομένα παραγγελιών και δεδομένα αποθεμάτων προϊόντων.

2.4.2.1.41. Περιγραφή μονάδας 41.

7.5.2. Δέχεται δεδομένα παραγγελιών και στέλνει δεδομένα αποθεμάτων προϊόντων για την αντίστοιχη μείωση τους.

2.4.2.1.42. Περιγραφή μονάδας 42.

8.1.1. Δέχεται δεδομένα αναζήτησης από τον χρήστη, από τις βάσεις πρώτων υλών, προϊόντων και παραγγελιών και εξάγει δεδομένα πρώτων υλών και λοιπών αποτελεσμάτων.

2.4.2.1.43. Περιγραφή μονάδας 43.

8.2.1. Δέχεται δεδομένα αναζήτησης από τον χρήστη, από τις βάσεις πρώτων υλών, προϊόντων και παραγγελιών και εξάγει δεδομένα προϊόντων και λοιπών αποτελεσμάτων.

2.4.2.1.44. Περιγραφή μονάδας 44.

8.3.1. Δέχεται δεδομένα αναζήτησης από τον χρήστη, από τις βάσεις πρώτων υλών, προϊόντων και παραγγελιών και εξάγει δεδομένα πελατών και λοιπών αποτελεσμάτων.

2.4.2.1.45. Περιγραφή μονάδας 45.

8.4.1. Δέχεται δεδομένα αναζήτησης από τον χρήστη από τις βάσεις πρώτων υλών, προϊόντων και παραγγελιών και στέλνει τα αντίστοιχα αποτελέσματα στον χρήστη, ενώ ταυτόχρονα ενημερώνει την βάση δεδομένων ανακληθέντων προϊόντων.

2.4.2.1.46. Περιγραφή μονάδας 46.

9.1.1. Δέχεται δεδομένα από τον χρήστη και από την βάση δεδομένων χρηστών και στέλνει μήνυμα αποδοχής και στοιχείων χρήστη.

2.4.2.1.47. Περιγραφή μονάδας 47.

9.1.2. Δέχεται στοιχεία από τον χρήστη και στέλνει τροποποιημένα δεδομένα στη βάση δεδομένων χρηστών και στις κατάλληλες διεπαφές.

2.4.2.1.48. Περιγραφή μονάδας 48.

9.2.1. Δέχεται δεδομένα από τον χρήστη και στέλνει κατάλληλο μήνυμα στην κεντρική εφαρμογή.

2.4.2.1.49. Περιγραφή μονάδας 49.

9.3.1. Δέχεται δεδομένα από τον χρήστη και στέλνει κατάλληλο μήνυμα στην κεντρική εφαρμογή.

2.4.2.1.50. Περιγραφή μονάδας 50.

9.3.2. Δέχεται μήνυμα από την κεντρική εφαρμογή και εμφανίζει μήνυμα ή την τερματίζει.

2.4.2.1.51. Περιγραφή μονάδας 51.

10.1.1. Δέχεται τα δεδομένα χρηστών και τα αποθηκεύει στην αντίστοιχη βάση δεδομένων.

2.4.2.1.52. Περιγραφή μονάδας 52.

10.2.1. Δέχεται τα δεδομένα των χρηστών για αλλαγή και από την αντίστοιχη βάση δεδομένων και στέλνει τα τροποποιημένα δεδομένα προς αποθήκευση.

2.4.2.1.53. Περιγραφή μονάδας 53.

10.2.2. Δέχεται τα τροποποιημένα δεδομένα των χρηστών και τα αποθηκεύει στην αντίστοιχη βάση δεδομένων.

2.4.2.1.54. Περιγραφή μονάδας 54.

10.3.1. Δέχεται δεδομένα χρηστών, ελέγχει εάν υπάρχουν σχετιζόμενες εκκρεμείς ενέργειες με τους αντίστοιχους χρήστες και στέλνει δεδομένα προς διαγραφή.

2.4.2.1.55. Περιγραφή μονάδας 55.

10.3.2. Δέχεται δεδομένα χρηστών προς διαγραφή και τα στέλνει στην αντίστοιχη βάση.

2.4.2.2. Αποσύνθεση σε ταυτόχρονες διεργασίες

Η περιγραφή των διεργασιών διαφαίνεται στο διάγραμμα ροής δεδομένων.

2.4.2.3. Αποσύνθεση δεδομένων

Η περιγραφή των οντοτήτων των δεδομένων διαφαίνεται στο διάγραμμα ροής δεδομένων.

2.4.3. Περιγραφή εξαρτήσεων

Η περιγραφή των εξαρτήσεων μεταξύ μονάδων, διεργασιών και δεδομένων φαίνεται στο διάγραμμα ροής δεδομένων.

2.4.3.1. Εξαρτήσεις μεταξύ μονάδων

Παραπομπή στο διάγραμμα ροής δεδομένων

2.4.3.2. Εξαρτήσεις μεταξύ διεργασιών

Παραπομπή στο διάγραμμα ροής δεδομένων

2.4.3.3. Εξαρτήσεις μεταξύ δεδομένων

(παραπομπή στο διάγραμμα οντοτήτων - συσχετίσεων)

2.4.4. Περιγραφή διαπροσωπειών

2.4.4.1. Διαπροσωπείες μονάδων

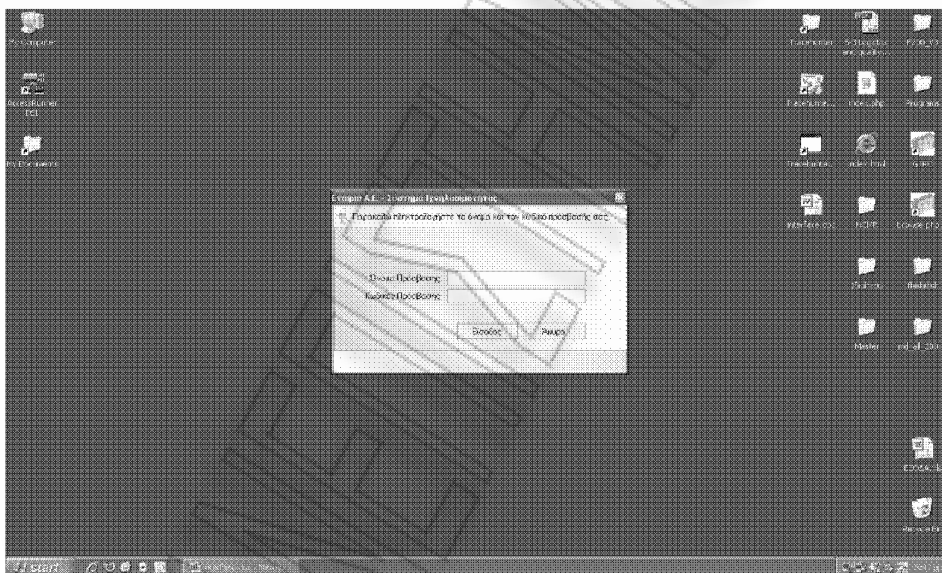
Παραπομπή στο διάγραμμα ροής δεδομένων

2.4.4.2. Διαπροσωπείες διεργασιών

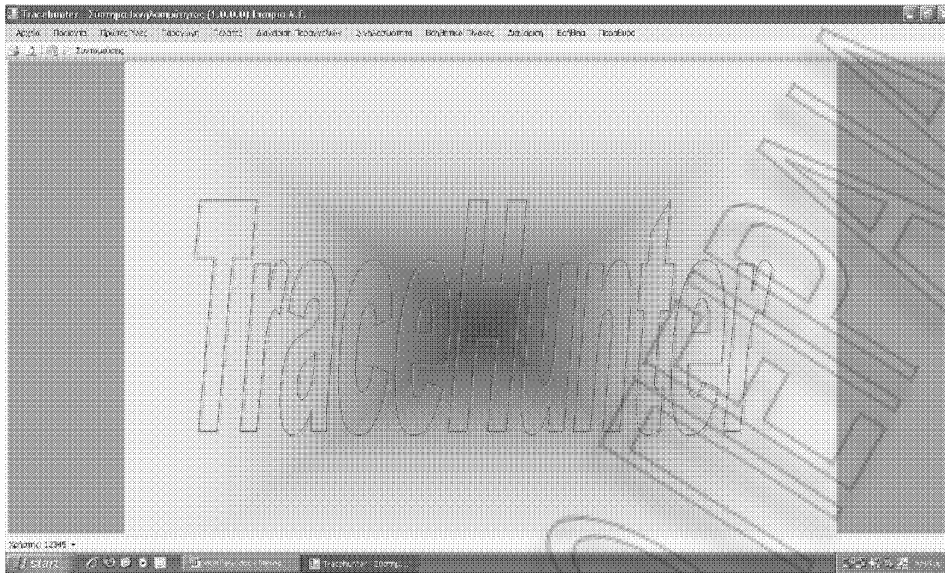
Παραπομπή στο διάγραμμα ροής δεδομένων

2.4.4.3. Διαπροσωπείες χρήστη

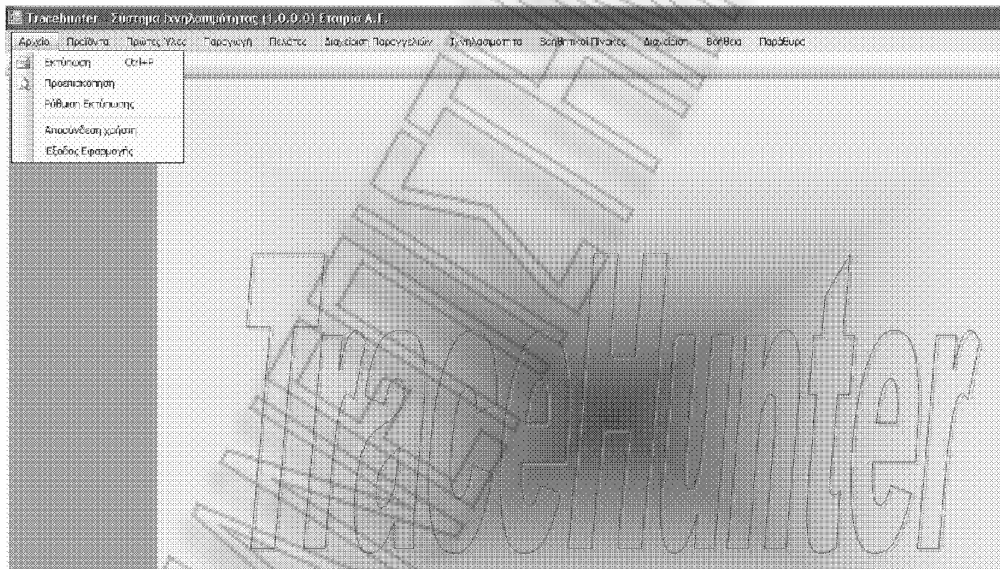
Είσοδος στο Σύστημα Ιχνηλασιμότητας



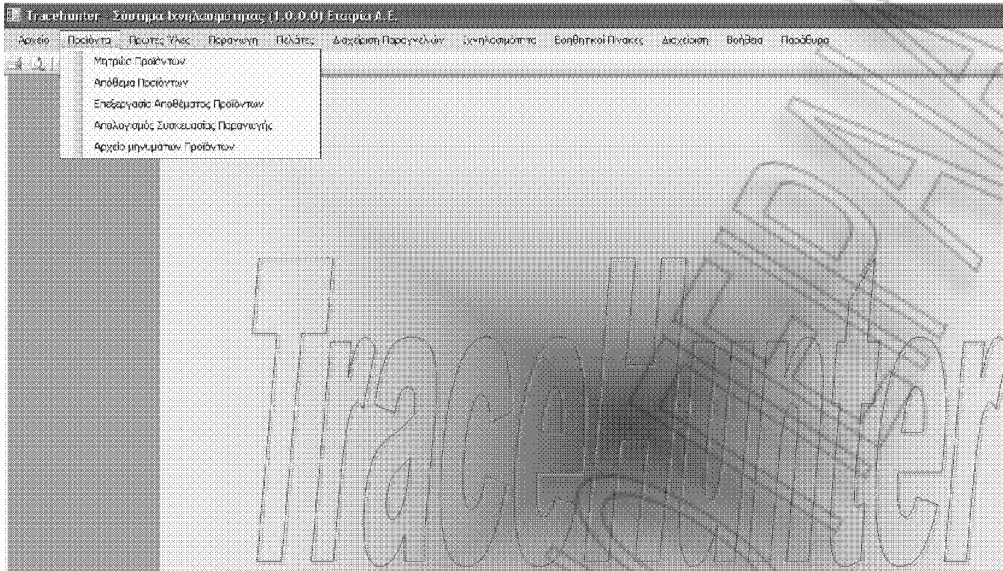
Κεντρικό μενού εφαρμογής



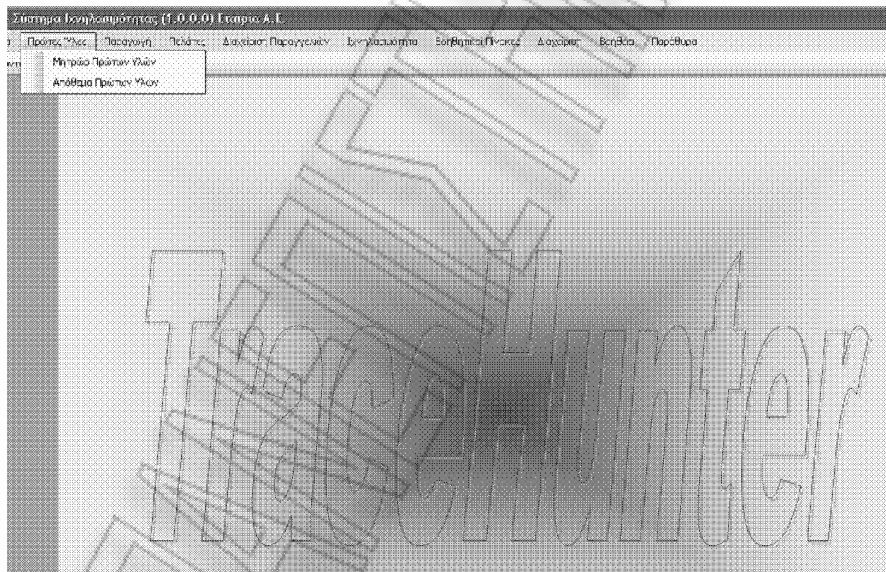
Μενού «Αρχείο»



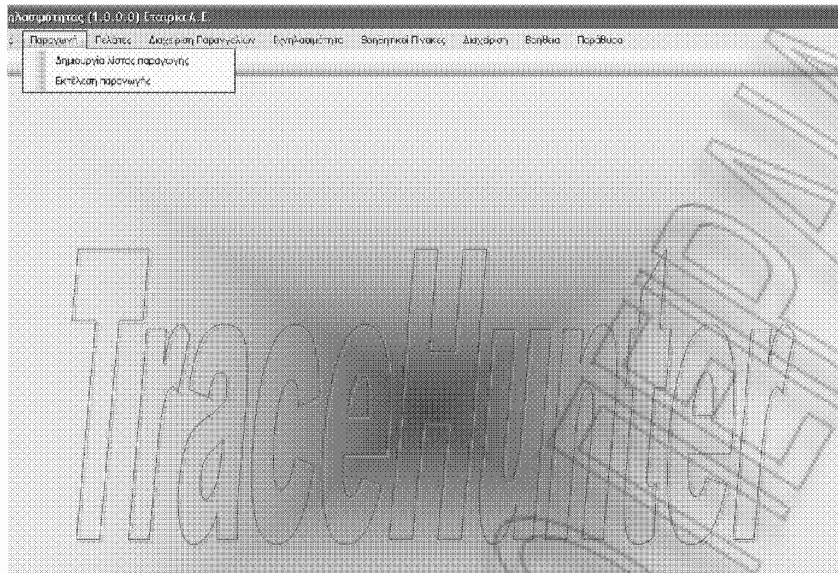
Μενού «Προϊόντα»



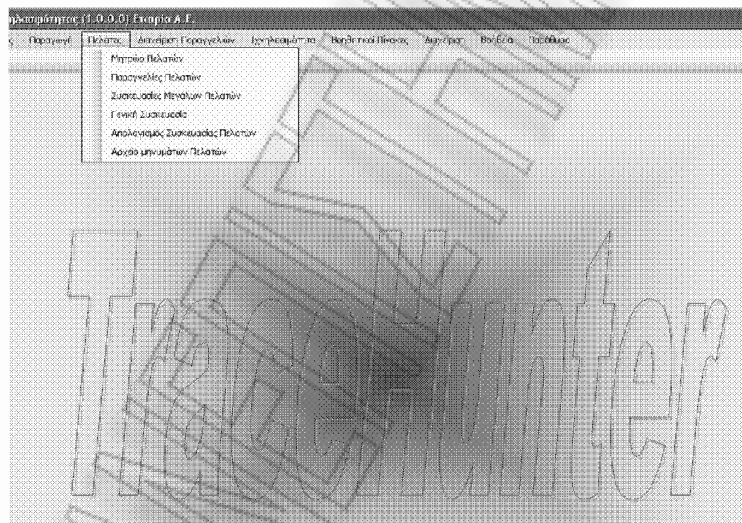
Μενού «Πρώτες ύλες»



Μενού «Παραγωγή»



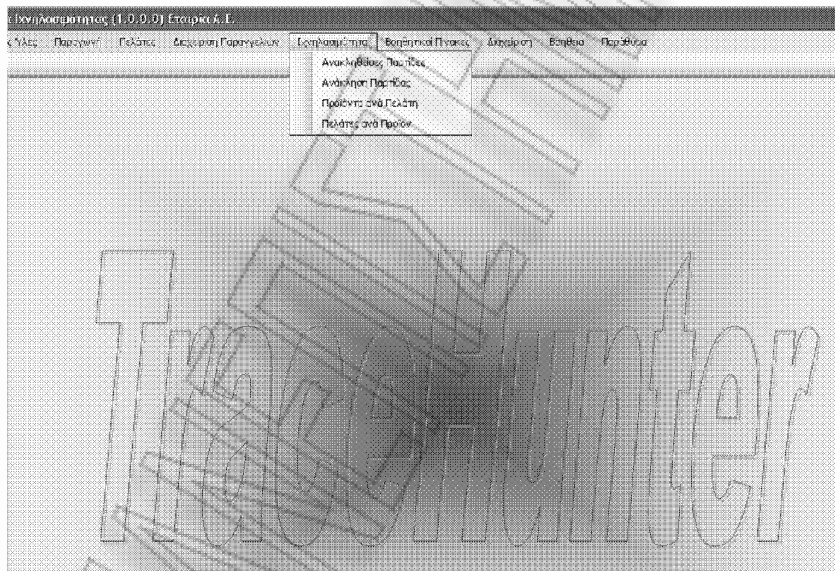
Μενού «Πελάτες»



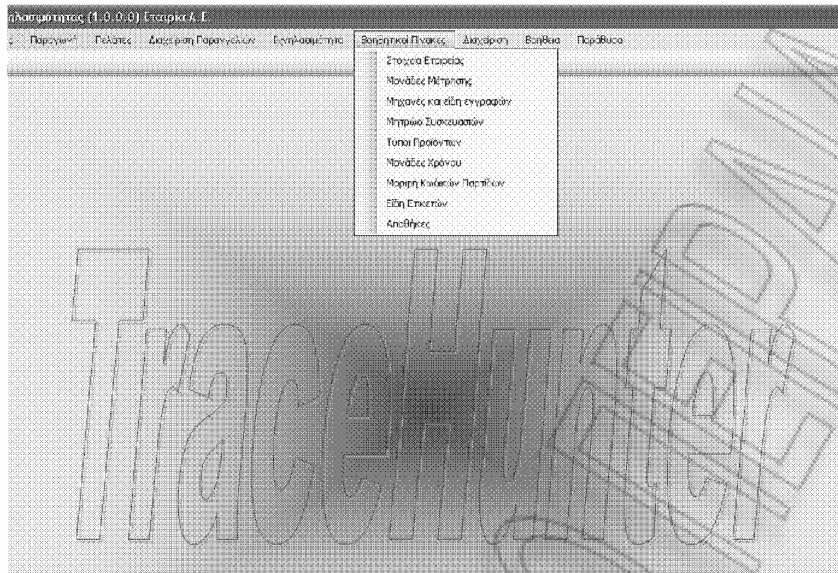
Μενού «Διαχείριση παραγγελιών»



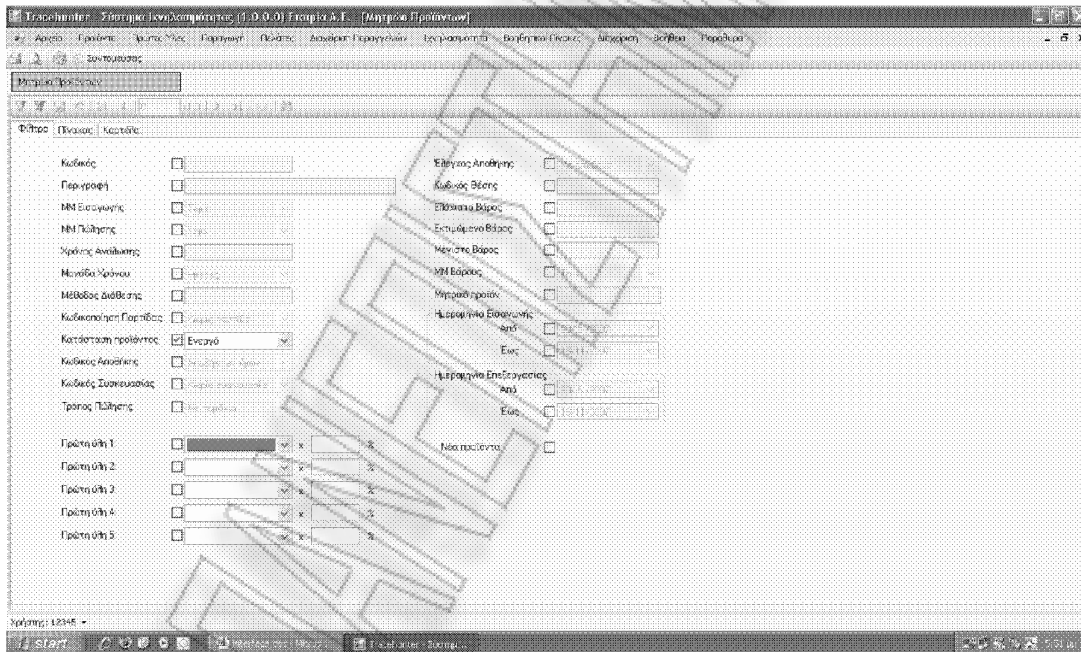
Μενού «Ιχνηλασιμότητα»



Μενού «Βοηθητικοί πίνακες»



Προϊόντα / Φόρμα «Μητρώο Προϊόντων» (φίλτρο)



Προϊόντα / Φόρμα «Απόθεμα προϊόντων» (πίνακας)

Κωδικός Προϊόντος	Περιγραφή Προϊόντος	ΜΜ Στοιχείου	Ποσότητα	Συνολικό Βάρος	Υπόλοιπο Βάρος	Επιπλέον Τεμάχια	Υπόλοιπο Τεμαχίων	Ημέρα Λήξης
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	150207	232.50	232.55	0	0	15/5/21
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	160207	317.14	317.14	0	0	17/5/21
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	161205	456.03	456.03	0	0	15/5/21
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	151205	714.87	714.37	0	0	15/5/21
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	211205	435.15	435.15	0	0	22/7/21
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	220107	736.42	736.42	0	0	22/7/21
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	220307	254.53	254.53	0	0	20/5/21
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	221205	65.87	65.87	0	0	20/5/21
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	230207	488.84	488.84	0	0	24/5/21
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	230307	513.35	513.35	0	0	21/5/21
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	230407	577.63	577.58	0	0	22/7/21
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	250107	52.27	52.27	0	0	25/4/21
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	260107	132.42	132.42	0	0	26/4/21
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	271205	206.12	206.12	0	0	27/3/21
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	281205	916.05	916.05	0	0	20/5/21
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	250107	436.51	436.51	0	0	23/4/21
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	250307	570.03	570.03	0	0	27/5/21
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	251105	655.32	655.32	0	0	25/5/21
00114	ΜΟΡΤΑΔΑΛΛΑ ΜΠΟΛΟΝΑ ΨΗΤΗ ΜΚΡΗ	Καδέ	300107	889.39	889.39	0	0	30/4/21
00115	ΠΑΡΙΖΑΣ PLUS 2 ΞΗΜ (1.5 ΕΥΡΟΣ)	Τεμάχ	210007	0.00	0.00	8	8	23/5/21
00122	ΑΔΥΝΑΜΑ ΚΟΚΤΑΪΡΑ ΒΡΑΣΤΑ	Καδέ	010007	746.94	746.94	0	0	19/5/21
00122	ΑΔΥΝΑΜΑ ΚΟΚΤΑΪΡΑ ΒΡΑΣΤΑ	Καδέ	020007	207.01	207.01	0	0	19/5/21
00122	ΑΔΥΝΑΜΑ ΚΟΚΤΑΪΡΑ ΒΡΑΣΤΑ	Καδέ	030007	696.97	696.97	0	0	19/5/21

Προϊόντα / Φόρμα «Μητρώο Προϊόντων» (καρτέλα)

Κωδικός:

Περιγραφή:

ΜΜ Στοιχείου:

ΜΜ Πόλης:

Χρόνος ανώτατης:

Μεγάλο Χρόνος:

Μέγιστος διαθέσιμος:

Κωδικός/ημερ. Γαρίδας:

Κατάσταση προϊόντος:

Κωδικός/επιπέδους:

Κωδικός Συναρμολογίας:

Τρόπος Πώλησης:

Επίπεδος Ανάθροιστος:

Κωδικός θέσης:

Επιπλέον Βάρος:

Μέγιστο Βάρος:

ΜΜ Εύρους:

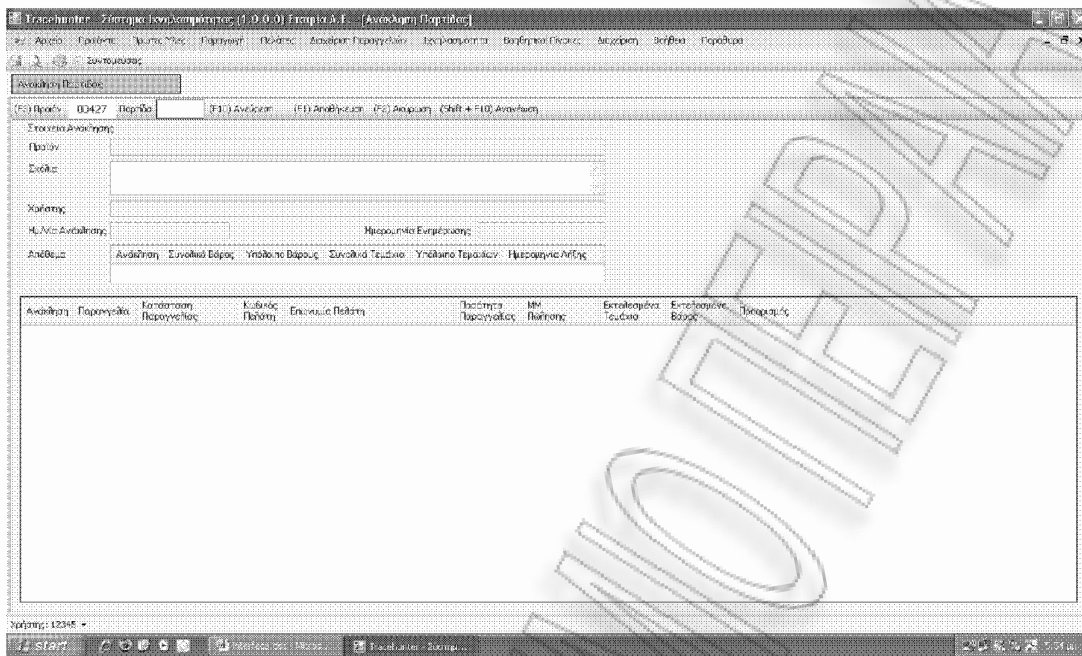
Μητρώο γαρίδας:

Περιγραφή Προϊόντος:

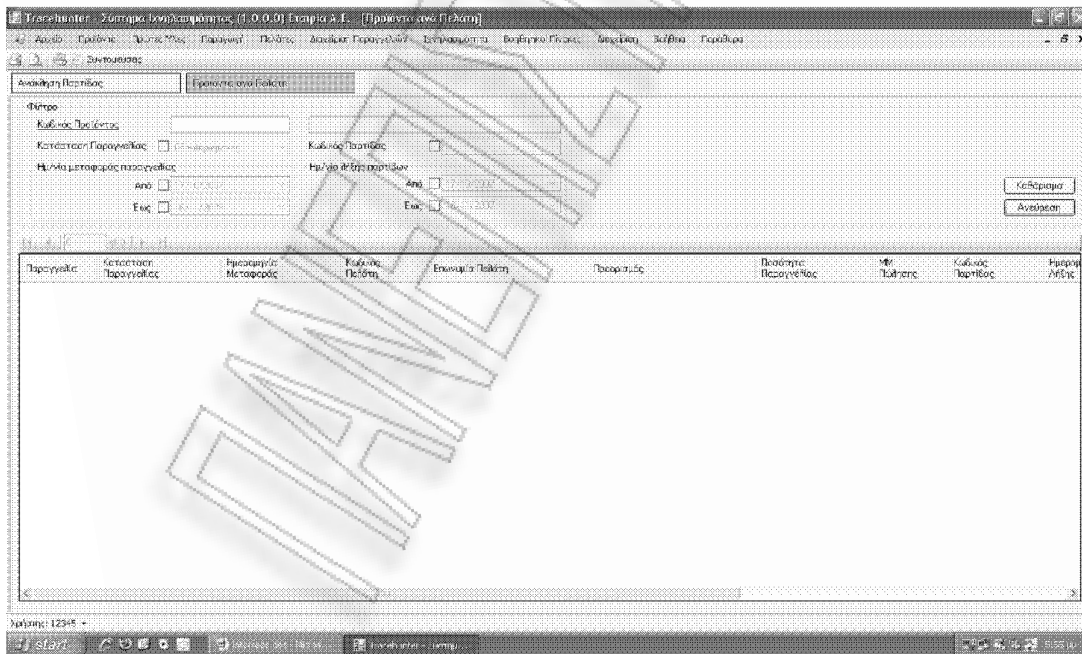
Ημερομηνία Εισαγωγής:

Ημερομηνία Επεξεργασίας:

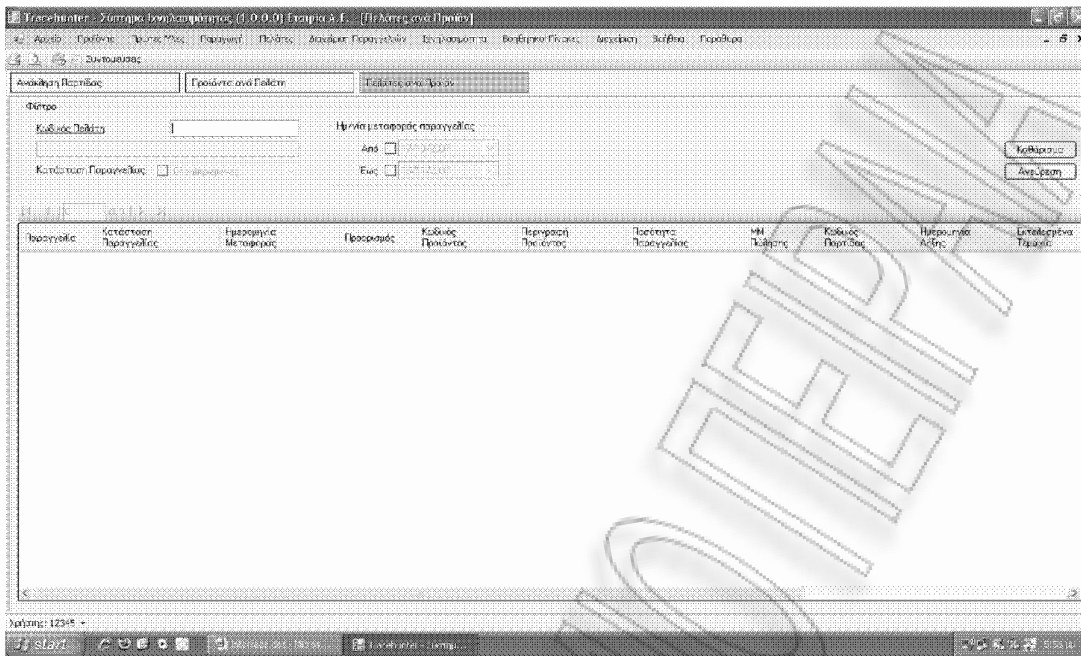
Ιχνηλασιμότητα / Φόρμα «Ανάκληση Παρτίδας»



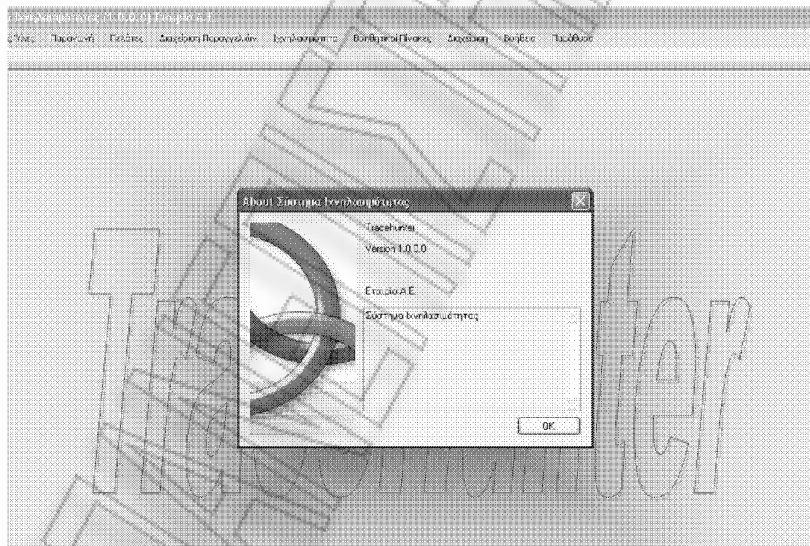
Ιχνηλασιμότητα / Φόρμα «Προϊόντα ανά πελάτη»



Ιχνηλασιμότητα / Φόρμα «Πελάτες ανά προϊόν»



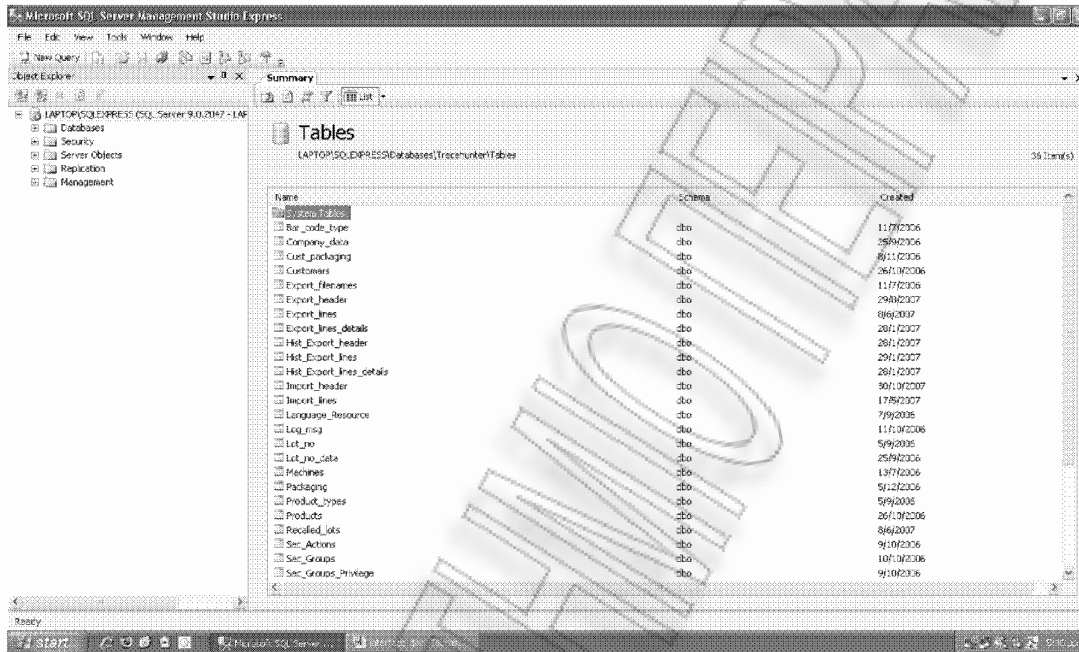
Οθόνη πληροφοριών συστήματος



2.4.4.4. Εξωτερικές διαπρωσιπέεις

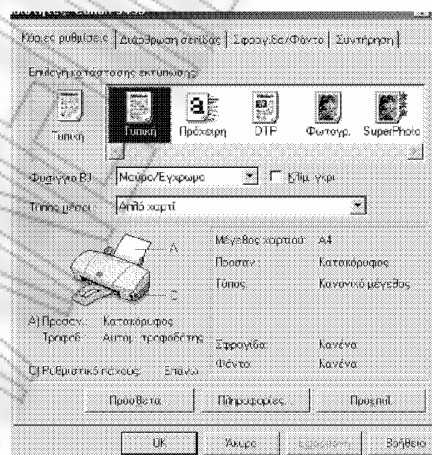
2.4.4.4.1. Συστήματα λογισμικού

➤ Θα συνεργάζεται με διαχείρισης βάσεων δεδομένων SQL.

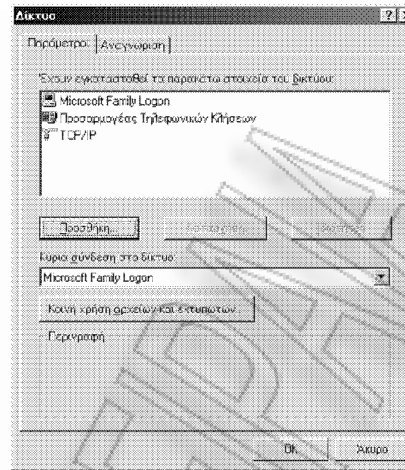
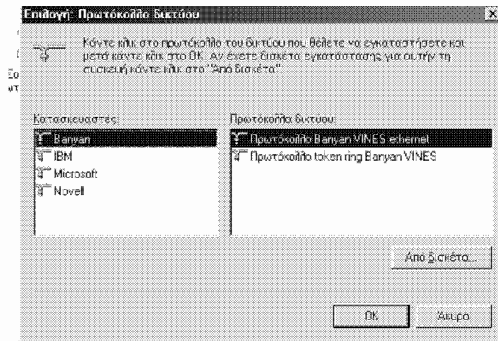


2.4.4.4.2. Συσκευές

➤ Το λογισμικό θα επικοινωνεί με εκτυπωτή.



➤ Θα είναι συνδεδεμένο κάρτα δικτύου για να λαμβάνει δεδομένα από άλλα συστήματα (π.χ. ERP).



- Θα είναι συνδεδεμένο με barcode reader / scanner για την εισαγωγή στο σύστημα κωδικών προϊόντων, πρώτων υλών, LOT numbers κ.α..

2.4.5. Λεπτομερές σχέδιο μονάδων

Το λεπτομερές σχέδιο μονάδων περιέχει μέρη από τον πηγαίο κώδικα του συστήματος. Στο παρών κείμενο είναι χωρισμένο σε πιο συγκεντρωτικά τμήματα από μονάδες ώστε να είναι όσο το δυνατόν πιο κατανοητό.

Login form

```
Imports System.Data.SqlClient

Public Class frmLogin

    Private Sub Form_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Me.Text = IT.A.Configuration("Global\Company\Info\Name") & " -
" & IT.A.Configuration("Tracehunter\General\Info\Description")
        Me.lblStatus.Text = IT.A.Language.Phrase(84, "Σύνδεση") &
"..."
        Me.lblLoginName.Text = IT.A.Language.Phrase(89, "Χρήστης")
        Me.lblPassword.Text = IT.A.Language.Phrase(88, "Κωδικός
Πρόσβασης")
        Me.cmdOK.Text = IT.A.Language.Phrase(29, "Εφαρμογή")
        Me.cmdCancel.Text = IT.A.Language.Phrase(28, "Άκυρο")
        Me.lblInfo.Text = IT.A.Language.Phrase(91, "Παρακαλώ
πληκτρολογήστε το όνομα και τον κωδικό πρόσβασης σας.")
    End Sub
```

```
Private Sub cmdOK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdOK.Click
    If Me.txtLoginName.Text.Length = 0 Then
        Me.txtLoginName.Focus()
        Exit Sub
    End If

    Me.lblStatus.Visible = True
    Me.Refresh()
    If IT.A.User.Login(Me.txtLoginName.Text, Me.txtPassword.Text)
Then
        frmMDI.Show()
        Me.Close()
        Me.Dispose()
    Else
        Me.lblStatus.Visible = False
        IT.Functions.RaiseError(IT.A.Language.Phrase(90, "Λάθος
στοιχεία σύνδεσης"), Me, , 1000)
        Me.txtLoginName.Focus()
        Me.txtLoginName.SelectAll()
    End If
End Sub

Private Sub cmdCancel_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmdCancel.Click
    End
End Sub

End Class
```


Menus form

```
Imports System.Data.SqlClient

Public Class frmMenus

    Private Const NEW_MENU_ITEM_NAME As String = "New Menu Item"
    Private Const DISABLE_ACTION_COMBO_TAG As String = "DO_NOT_SELECT_INDEX"
    Private Const SEPARATOR_CHAR As String = "-"

    Private dt As DataTable
    Private da As New SqlDataAdapter

    Public Sub New()

        ' This call is required by the Windows Form Designer.
        InitializeComponent()

        ' Bind fields to TextBox controls
        Me.txtID.DataBindings.Add(New Binding("Text",
        Me.BindingSource1, "ID", True))
        Me.txtPID.DataBindings.Add(New Binding("Text",
        Me.BindingSource1, "PID", True))
        Me.txtCaption.DataBindings.Add(New Binding("Text",
        Me.BindingSource1, "Caption", True))
        Me.picImage.DataBindings.Add(New Binding("Image",
        Me.BindingSource1, "Image", True))
        Me.txtLanguage_Resource_ID.DataBindings.Add(New
        Binding("Text", Me.BindingSource1, "Language_Resource_ID", True))
        Me.txtActionID.DataBindings.Add(New Binding("Text",
        Me.BindingSource1, "ActionID", True))
        Me.udOrdinal_Position.DataBindings.Add(New Binding("Text",
        Me.BindingSource1, "Ordinal_Position", True))

        Dim con As New SqlConnection(IT.A.ConnectionString)
        Dim com As New SqlCommand("SELECT * FROM Sec_Menus", con)

        dt = New DataTable("Menu")
        da.SelectCommand = com
        da.Fill(dt)
        Me.BindingSource1.DataSource = dt

        LoadActions()
    End Sub
End Class
```

```
End Sub

Private Sub LoadActions()
    Dim con As New SqlConnection(IT.A.ConnectionString)
    Dim com As New SqlCommand("SELECT ID, Name FROM Sec_Actions",
con)

    Dim dtActions As New DataTable("Actions")
    Dim daActions As New SqlDataAdapter

    daActions.SelectCommand = com
    daActions.Fill(dtActions)
    Me.cmbActionID.DataSource = dtActions
    Me.cmbActionID.DisplayMember = "Name"
    Me.cmbActionID.ValueMember = "ID"
End Sub

Private Sub Form_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Me.Text = IT.A.Language.Phrase(129, "Διαχείριση Μενού")
    LoadParentMenus(Me.tvMenus.Nodes("Key0"))
    Me.tvMenus.Nodes("Key0").Expand()
End Sub

Private Sub LoadParentMenus(ByVal tvNode As TreeNode)
    Dim SQL As String = "SELECT * FROM Sec_Menus WHERE PID=" & 0
    Dim con As New SqlClient.SqlConnection(IT.A.ConnectionString)
    Dim com As New SqlClient.SqlCommand(SQL, con)
    con.Open()
    Dim reader As SqlClient.SqlDataReader = com.ExecuteReader
    If reader.HasRows Then
        Do While reader.Read
            Dim strKey As String = "Key" & CStr(reader("ID"))
            Dim strText As String = CStr(reader("Caption"))
            Dim image As Image = Nothing
            If Not IsDBNull(reader("Image")) Then
                Dim bytImage() As Byte = CType(reader("Image"),
Byte())
                Dim stream As New System.IO.MemoryStream(bytImage)
                image = Drawing.Image.FromStream(stream)
                lstImage.Images.Add(strKey, image)
            End If
            Dim node As TreeNode = tvNode.Nodes.Add(strKey,
strText)
```

```

        If Not image Is Nothing Then
            node.ImageKey = strKey
            node.SelectedImageKey = strKey
        End If
        LoadMenuItems (node, CInt(reader("ID")))
    Loop
End If
reader.Close()
con.Close()
End Sub

Private Sub LoadMenuItems(ByVal tvNode As TreeNode, ByVal intID As
Integer)
    Dim SQL As String = "SELECT * FROM Sec_Menus WHERE PID=" &
intID

    Dim con As New SqlClient.SqlConnection(IT.A.ConnectionString)
    Dim com As New SqlClient.SqlCommand(SQL, con)
    con.Open()
    Dim reader As SqlClient.SqlDataReader = com.ExecuteReader
    If reader.HasRows Then
        Do While reader.Read
            Dim strKey As String = "Key" & CStr(reader("ID"))
            Dim strText As String = CStr(reader("Caption"))
            Dim image As Image = Nothing
            If Not IsDBNull(reader("Image")) Then
                Dim bytImage() As Byte = CType(reader("Image"),
Byte())

                Dim stream As New System.IO.MemoryStream(bytImage)
                image = Drawing.Image.FromStream(stream)
                lstImage.Images.Add(strKey, image)
            End If
            Dim node As TreeNode = tvNode.Nodes.Add(strKey,
strText)

            If Not image Is Nothing Then
                node.ImageKey = strKey
                node.SelectedImageKey = strKey
            End If

            LoadMenuItems (node, CInt(reader("ID")))
        Loop
    End If
    reader.Close()
    con.Close()
End Sub

```

```

End Sub

Private Sub tvMenus_AfterSelect(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.TreeViewEventArgs) Handles
tvMenus.AfterSelect
    SaveChanges (False)
    lblPath.Text = Me.tvMenus.SelectedNode.FullPath
    PanelEditMenu.Enabled = Not Me.tvMenus.SelectedNode.Name =
"Key0"
    Me.ToolBar_cmdDelete.Enabled = Not
Me.tvMenus.SelectedNode.Name = "Key0"
    Me.ToolBar_cmdSave.Enabled = Not Me.tvMenus.SelectedNode.Name
= "Key0"
    Me.ToolBar_cmdUndo.Enabled = Not Me.tvMenus.SelectedNode.Name
= "Key0"
    Me.ToolBar_cmdAddSeparator.Enabled = Not
Me.tvMenus.SelectedNode.Name = "Key0"
    LoadFields (Me.tvMenus.SelectedNode)

    Me.txtCaption.Enabled = Not (Me.txtCaption.Text =
SEPARATOR_CHAR)
    Me.cmdSelectImage.Enabled = Not (Me.txtCaption.Text =
SEPARATOR_CHAR)
    Me.cmdRemoveImage.Enabled = Not (Me.txtCaption.Text =
SEPARATOR_CHAR)
    Me.txtLanguage_Resource_ID.Enabled = Not (Me.txtCaption.Text =
SEPARATOR_CHAR)
    Me.cmbActionID.Enabled = Not (Me.txtCaption.Text =
SEPARATOR_CHAR)
    Me.lblIsSeparator.Visible = Me.txtCaption.Text =
SEPARATOR_CHAR

End Sub

Private Sub LoadFields (ByVal CurNode As TreeNode)

    Dim intID As Integer = CInt (CurNode.Name.Substring(3))
    Dim strSQL As String = "SELECT * FROM Sec_Menus WHERE ID=" &
intID

    da.SelectCommand = New SqlCommand(strSQL)
    da.SelectCommand.Connection = New
SqlConnection(IT.A.ConnectionString)

```

```
Dim comBuilder As SqlCommandBuilder = New
SqlCommandBuilder(da)
dt.Clear()
da.Fill(dt)

SetActionComboIndex()

Me.BindingSource1.DataSource = dt

End Sub

Private Sub cmdSelectImage_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cmdSelectImage.Click
Dim result As DialogResult = dlgOpenFile.ShowDialog()
If result = Windows.Forms.DialogResult.OK Then
Me.picImage.Image =
Drawing.Image.FromFile(dlgOpenFile.FileName)
End If
End Sub

Private Sub cmdRemoveImage_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cmdRemoveImage.Click
Me.picImage.Image = Nothing
End Sub

Private Sub SaveChanges(ByVal blnMode As Boolean)
Me.BindingSource1.CurrencyManager.EndCurrentEdit()
If dt Is Nothing Then Exit Sub
Dim dtChanges As DataTable = dt.GetChanges
If dtChanges Is Nothing Then Exit Sub
If blnMode Then
da.Update(dtChanges)
dtChanges.AcceptChanges()
Dim CurNode As TreeNode = Me.tvMenus.SelectedNode
Dim strKey As String = "Key" & Me.txtID.Text
CurNode.Text = Me.txtCaption.Text
If Me.picImage.Image Is Nothing Then
CurNode.ImageIndex = 0
CurNode.SelectedImageIndex = 0
lstImage.Images.RemoveByKey(strKey)
Else
lstImage.Images.RemoveByKey(strKey)
lstImage.Images.Add(strKey, Me.picImage.Image)
CurNode.ImageKey = strKey
```

```
        CurNode.SelectedImageKey = strKey
    End If
Else
    dt.RejectChanges()
End If
SetActionComboIndex()
End Sub

Private Sub ToolBarAdd_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ToolBar_cmdAdd.Click
    Add()
End Sub

Private Sub ToolBarAddSeparator_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ToolBar_cmdAddSeparator.Click
    Add(True)
End Sub

Private Sub ToolBarDelete_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ToolBar_cmdDelete.Click
    Dim strKey As String = Me.tvMenus.SelectedNode.Name
    Dim intID As Integer = CInt(strKey.Substring(3))
    Dim con As New SqlClient.SqlConnection(IT.A.ConnectionString)
    Dim com As New SqlClient.SqlCommand
    con.Open()
    com.Connection = con
    com.CommandText = "DELETE FROM Sec_Menus WHERE PID=" & intID
    com.ExecuteNonQuery()
    com.CommandText = "DELETE FROM Sec_Menus WHERE ID=" & intID
    com.ExecuteNonQuery()
    con.Close()
    Dim nodeToDelete As TreeNode = Me.tvMenus.SelectedNode
    Dim parentNode As TreeNode = nodeToDelete.Parent
    Me.tvMenus.Nodes.Remove(nodeToDelete)
End Sub

Private Sub ToolBarSave_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ToolBar_cmdSave.Click
    SaveChanges(True)
End Sub

Private Sub ToolBarUndo_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ToolBar_cmdUndo.Click
```

```
        SaveChanges (False)
    End Sub

    Private Sub cmbActionID_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmbActionID.SelectedIndexChanged
        If Not Me.cmbActionID.Tag = DISABLE_Action_COMBO_TAG Then
            Me.txtActionID.Text =
Me.cmbActionID.SelectedValue.ToString
        End If
    End Sub

    Private Sub SetActionComboIndex()
        If Me.dt.Rows.Count > 0 Then
            Me.cmbActionID.Tag = DISABLE_Action_COMBO_TAG

            Dim intNewIndex As Integer =
CInt(Me.dt.Rows(0).Item("ActionID").ToString())

            Me.cmbActionID.SelectedValue = intNewIndex

            Me.cmbActionID.Tag = ""
        End If
    End Sub

    Private Sub Add(Optional ByVal blnIsSeparator As Boolean = False)

        Dim con As New SqlConnection(IT.A.ConnectionString)
        Dim com As New SqlCommand
        con.Open()
        com.Connection = con

        com.CommandText = "SELECT Max(ID) FROM Sec_Menus"

        Dim objNewID As Object = com.ExecuteScalar
        Dim strNewID As String = ""
        Dim strNEW_MENU_ITEM_NAME As String = IIf(blnIsSeparator,
SEPARATOR_CHAR, NEW_MENU_ITEM_NAME)

        If IsDBNull(objNewID) Then
            strNewID = "1"
        Else
            strNewID = CStr(CInt(objNewID) + 1)
        End If
    End Sub
```

```
        Dim          strPID          As          String          =
Me.tvMenus.SelectedNode.Name.Substring(3)

        com.CommandText = "INSERT INTO Sec_Menus (ID, PID, Caption,
ActionID) VALUES (@ID, @PID, @Caption, @ActionID)"

        com.Parameters.Add("@ID", SqlDbType.Int)
        com.Parameters.Add("@PID", SqlDbType.Int)
        com.Parameters.Add("@Caption", SqlDbType.NVarChar)
        com.Parameters.Add("@ActionID", SqlDbType.TinyInt)

        com.Parameters(0).Value = CType(strNewID, Integer)
        com.Parameters(1).Value = CType(strPID, Integer)
        com.Parameters(2).Value = CType(strNEW_MENU_ITEM_NAME, String)

        com.Parameters(3).Value = 0

        com.ExecuteNonQuery()

        con.Close()

        Dim          NewNode          As          TreeNode          =
Me.tvMenus.SelectedNode.Nodes.Add("Key" & strNewID,
strNEW_MENU_ITEM_NAME)

        Me.tvMenus.SelectedNode = NewNode
    End Sub

    Private Sub ToolBar_ReloadApplication_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ToolBar_ReloadApplication.Click
        Application.Restart()
    End Sub

End Class
```


General module

```
Imports system.xml
```

```
Namespace IT.A
```

```
    <Global.Microsoft.VisualBasic.HideModuleNameAttribute()> _
    Friend Module Shortcuts

        Friend ReadOnly Property ConnectionString() As String
            Get
                Return
                    Global.TracehunterClasses.Singletons.ConnectionString
            End Get
        End Property

        Friend      ReadOnly      Property      Configuration()      As
Global.TracehunterClasses.Configuration.Configuration
            Get
                Return
                    Global.TracehunterClasses.Singletons.Configuration
            End Get
        End Property

        Friend      ReadOnly      Property      User()      As
Global.TracehunterClasses.Session.User
            Get
                Return Global.TracehunterClasses.Singletons.User
            End Get
        End Property

        Friend      ReadOnly      Property      Language()      As
Global.TracehunterClasses.Language.Language
            Get
                Return
                    Global.TracehunterClasses.Singletons.Language(Global.TracehunterClasses
                    .Singletons.Configuration("Tracehunter\General\Info\Language"))
            End Get
        End Property

    End Module
```

```
End Namespace
```

```
Public Enum it_OrderExecutionModeEnum
    Unset
    DI80
    Bizerba
End Enum

Public Enum it_OpenMode
    ProductAdd
    ProductDelete
    ProductEdit
    DetailAddNew
    DetailEdit
    Normal
End Enum

Public Enum it_PrepareFormEnum
    SearchOrder
    Scan
    EditHeader
    EditLines
    EditLineComments
    EditDetail
    EditOrder
End Enum

Public Enum it_ProductPackEnum
    Items = 1
    Weight = 2
End Enum

Public Enum it_CalculatingActionEnum
    Add
    Subtract
End Enum

Public Enum it_OrderStatus
    Unset = -1      'Καμία κατάσταση
    Registered = 0 'Καταχωρημένη
    InHand = 1     'Υπό εκτέλεση
    Completed = 2  'Εκτελεσμένη
    TransferToERP = 3 'Μεταφορά σε ERP
End Enum
```

```
Public Enum it_CustomerFields
    ID = 0
    Code
    Name
    Address
    Tel
    Contact_person
    Contact_person_tel
    Contact_person_email
    Recall_products
    Status
End Enum

Public Enum it_OrderFields
    Order_no = 0
    Customer_code
    Destination
    Export_date
    Comments
    Boxes
    Itinerary
    Status
End Enum

Public Class it_ExportLine
    Private plngOrderCode As Long
    Private plngProductCode As Long
    Private pstrProductDescription As String
    Private pdblWeight As Double
    Private plngQuantity As Long
    Private pstrPack As String
    Private pstrComment As String
End Class

Public Enum it_MachineType
    UNKNOWN = 0
    FCW = 1
End Enum

Public Enum it_MessageType
    UNKNOWN = 0
    HEADER = 1
    DETAIL = 2
    ENDING = 3
```

```
End Enum
```

```
Public Enum it_PortStatusEnum
```

```
    DISABLED = 0
```

```
    NEEDS_ADJUSTMENT = 1
```

```
    DISCONNECTED = 2
```

```
    CONNECTED = 3
```

```
End Enum
```

```
Public Class it_OrderHeader
```

```
    Private pDataAdapter As SqlClient.SqlDataAdapter
```

```
    Private pDataTable As System.Data.DataTable
```

```
    Public Sub New(ByVal strOrderNo As String)
```

```
        If strOrderNo.Length = 0 Then Throw New Exception("Invalid  
paramater")
```

```
        Dim con As New SqlClient.SqlConnection(IT.A.ConnectionString)
```

```
        Me.pDataTable = New System.Data.DataTable
```

```
        Me.pDataAdapter = New SqlClient.SqlDataAdapter
```

```
        Me.pDataAdapter.SelectCommand = New  
SqlClient.SqlCommand("SELECT * FROM Export_Header WHERE Order_no='" &  
strOrderNo & "'", con)
```

```
        Me.pDataAdapter.AcceptChangesDuringUpdate = True
```

```
        Me.pDataAdapter.Fill(Me.pDataTable)
```

```
        If Me.pDataTable.Rows.Count = 0 Then Throw New  
Exception("Requested order not found.")
```

```
        Dim pCommandBuilder As New  
SqlClient.SqlCommandBuilder(Me.pDataAdapter)
```

```
    End Sub
```

```
    Default Public Property Field(ByVal index As it_OrderFields) As  
Object
```

```
        Get
```

```
            Return Me.pDataTable.Rows(0).Item(index)
```

```
        End Get
```

```
        Set(ByVal value As Object)
```

```
        Me.pDataTable.Rows(0).Item(index) = value
    End Set

End Property

Public Sub Update()
    Me(it_OrderFields.Status) = it_OrderStatus.InHand
    Me.pDataAdapter.Update(Me.pDataTable)
End Sub

Public Sub Dispose()
    Me.pDataTable.Dispose()
    Me.pDataAdapter.Dispose()
End Sub

End Class

Public Class it_Customer

    Private pDataAdapter As SqlClient.SqlDataAdapter
    Private pDataTable As System.Data.DataTable

    Public Sub New(ByVal intID As Integer)

        If intID = 0 Then Throw New Exception("Invalide paramater")

        Dim con As New SqlClient.SqlConnection(IT.A.ConnectionString)

        Me.pDataTable = New System.Data.DataTable

        Me.pDataAdapter = New SqlClient.SqlDataAdapter
        Me.pDataAdapter.SelectCommand = New
        SqlClient.SqlCommand("SELECT * FROM Customers WHERE ID='" & intID &
        "'", con)
        Me.pDataAdapter.AcceptChangesDuringUpdate = True
        Me.pDataAdapter.Fill(Me.pDataTable)

        If Me.pDataTable.Rows.Count = 0 Then Throw New
        Exception("Requested customer not found.")

        Dim pCommandBuilder As New
        SqlClient.SqlCommandBuilder(Me.pDataAdapter)

    End Sub
```

```
Public Sub New(ByVal strCode As String)

    If strCode.Length = 0 Then Throw New Exception("Invalid
parameter")

    Dim con As New SqlConnection(IT.A.ConnectionString)

    Me.pDataTable = New System.Data.DataTable

    Me.pDataAdapter = New SqlConnection.SqlDataAdapter
    Me.pDataAdapter.SelectCommand = New
SqlConnection.SqlCommand("SELECT * FROM Customers WHERE Code='" & strCode
& "'", con)
    Me.pDataAdapter.AcceptChangesDuringUpdate = True
    Me.pDataAdapter.Fill(Me.pDataTable)

    If Me.pDataTable.Rows.Count = 0 Then Throw New
Exception("Requested customer not found.")

    Dim pCommandBuilder As New
SqlConnection.SqlCommandBuilder(Me.pDataAdapter)

End Sub

Default Public Property Field(ByVal index As it_CustomerFields) As
Object
    Get
        Return Me.pDataTable.Rows(0).Item(index)
    End Get
    Set(ByVal value As Object)
        Me.pDataTable.Rows(0).Item(index) = value
    End Set
End Property

Public Sub Update()
    Me.pDataAdapter.Update(Me.pDataTable)
End Sub

Public Sub Dispose()
    Me.pDataTable.Dispose()
    Me.pDataAdapter.Dispose()
End Sub
```

```
End Class
```

```
Public Class it_Factories
```

```
    Public Shared Function MachineMessageFactory(ByVal strRecord As  
String) As it IMessageFactory
```

```
        If strRecord.Length = 0 Then Throw New Exception("You must  
specify a record being descended from a Com Port")
```

```
        Dim mt As it_MachineType = it_MachineType.FCW
```

```
        Dim msgf As it IMessageFactory
```

```
        Select Case mt
```

```
            Case it_MachineType.FCW
```

```
                msgf = New it_FCWMessageFactory
```

```
            Case Else
```

```
                Throw New Exception("Unknown machine type.")
```

```
        End Select
```

```
        Return msgf
```

```
    End Function
```

```
End Class
```

```
Public Interface it IMessageFactory
```

```
    Function BuildMessage(ByVal strRecord As String) As it_Message
```

```
End Interface
```

```
Public Class it_FCWMessageFactory
```

```
    Implements it IMessageFactory
```

```
    Public Function BuildMessage(ByVal strRecord As String) As  
it_Message Implements it IMessageFactory.BuildMessage
```

```
        Dim msg As New it_Message
```

```
        With msg
```

```
            .MachineType = it_MachineType.FCW
```

```
            .MachineCode = ""
```

```
.MessageType = it_MessageType.HEADER
.MessageText = strRecord

Select Case .MessageType

    Case it_MessageType.HEADER
        .UserCode = ""
        .ProductCode = ""
        .BarcodeCode = ""
        .LotCode = ""

    Case it_MessageType.DETAIL
        .Weight = 0
        .Weight2 = 0

    Case it_MessageType.ENDING
        .EndFlag = ""

    Case Else

End Select

End With

Return msg

End Function

End Class

Public Class it_Message

    Dim pMessageType As it_MessageType
    Dim pMessageText As String
    Dim pMachineType As it_MachineType
    Dim pMachineCode As String
    Dim pUserCode As String
    Dim pProductCode As String
    Dim pBarcodeCode As String
    Dim pLotCode As String
    Dim pWeight As Double
    Dim pWeight2 As Double
    Dim pEndFlag As String
```



```
Public Sub New()  
    Me.MessageType = it_MessageType.UNKNOWN  
    Me.MessageText = ""  
    Me.MachineType = it_MachineType.UNKNOWN  
    Me.MachineCode = ""  
    Me.UserCode = ""  
    Me.ProductCode = ""  
    Me.BarcodeCode = ""  
    Me.LotCode = ""  
    Me.Weight = 0  
    Me.Weight2 = 0  
    Me.EndFlag = ""  
End Sub  
  
Public Property MessageType() As it_MessageType  
    Get  
        Return Me.pMessageType  
    End Get  
    Set(ByVal myMessageType As it_MessageType)  
        Me.pMessageType = myMessageType  
    End Set  
End Property  
  
Public Property MessageText() As String  
    Get  
        Return Me.pMessageText  
    End Get  
    Set(ByVal myMessageText As String)  
        Me.pMessageText = myMessageText  
    End Set  
End Property  
  
Public Property MachineType() As it_MachineType  
    Get  
        Return Me.pMachineType  
    End Get  
    Set(ByVal myMachineType As it_MachineType)  
        Me.pMachineType = myMachineType  
    End Set  
End Property  
  
Public Property MachineCode() As String  
    Get  
        Return Me.pMachineCode
```

```
End Get
Set(ByVal myMachineCode As String)
    Me.pMachineCode = myMachineCode
End Set
End Property
```

```
Public Property UserCode() As String
Get
    Return Me.pUserCode
End Get
Set(ByVal myUserCode As String)
    Me.pUserCode = myUserCode
End Set
End Property
```

```
Public Property ProductCode() As String
Get
    Return Me.pProductCode
End Get
Set(ByVal myProductCode As String)
    Me.pProductCode = myProductCode
End Set
End Property
```

```
Public Property BarcodeCode() As String
Get
    Return Me.pBarcodeCode
End Get
Set(ByVal myBarcodeCode As String)
    Me.pBarcodeCode = myBarcodeCode
End Set
End Property
```

```
Public Property LotCode() As String
Get
    Return Me.pLotCode
End Get
Set(ByVal myLotCode As String)
    Me.pLotCode = myLotCode
End Set
End Property
```

```
Public Property Weight() As Double
Get
```

```
        Return Me.pWeight
    End Get
    Set(ByVal myWeight As Double)
        Me.pWeight = myWeight
    End Set
End Property

Public Property Weight2() As Double
    Get
        Return Me.pWeight2
    End Get
    Set(ByVal myWeight2 As Double)
        Me.pWeight2 = myWeight2
    End Set
End Property

Public Property EndFlag() As String
    Get
        Return Me.pEndFlag
    End Get
    Set(ByVal myEndFlag As String)
        Me.pEndFlag = myEndFlag
    End Set
End Property

End Class

Public Class WarehouseBuilder

    Private pstrWorkingDirectory As String
    Private pstrHistoryDirectory As String

    Public Sub New()
        Me.WorkingDirectory = New String("")
        Me.HistoryDirectory = New String("")
    End Sub

    Public Sub New(ByVal strWorkingDirectory As String, ByVal
strHistoryDirectory As String)
        Me.WorkingDirectory = strWorkingDirectory
        Me.HistoryDirectory = strHistoryDirectory
    End Sub

    Property WorkingDirectory()
```

```
Get
    Return Me.pstrWorkingDirectory
End Get

Set(ByVal strDirectory)
    Me.pstrWorkingDirectory = strDirectory
End Set

End Property

Property HistoryDirectory()

    Get
        Return Me.pstrHistoryDirectory
    End Get

    Set(ByVal strDirectory)
        Me.pstrHistoryDirectory = strDirectory
    End Set

End Property

Public Sub UpDateDatabase()
    MessageBox.Show("UpDateDatabase")
End Sub

Public Sub MoveFileToHistory()
    MessageBox.Show("MoveFileToHistory")
End Sub

Public Sub ExecuteMessage(ByVal myMessage As it_Message)

    Select Case myMessage.MessageType

        Case it_MessageType.HEADER

            Me.CreateHeader(myMessage)

        Case it_MessageType.DETAIL

            Me.WriteDetail(myMessage)

        Case it_MessageType.ENDING
```

```
        Me.CloseHeader(myMessage)
        Me.UpDateDatabase()
        Me.MoveFileToHistory()

    Case Else
        Throw New Exception("Unknown or invalid message
type.")

    End Select

End Sub

Private Sub CreateHeader(ByVal myMessage As it_Message)

    Dim myXmlSettings As New XmlWriterSettings
    myXmlSettings.Indent = True
    myXmlSettings.OmitXmlDeclaration = True

    Using myXmlWriter As XmlWriter = XmlWriter.Create("test.xml",
myXmlSettings)

        With myXmlWriter
            .WriteStartElement("HEADER")
            .WriteAttributeString("MachineCode",
myMessage.MachineCode)
            .WriteAttributeString("UserCode", myMessage.UserCode)
            .WriteAttributeString("ProductCode",
myMessage.ProductCode)
            .WriteAttributeString("BarcodeCode",
myMessage.BarcodeCode)
            .WriteAttributeString("LotCode", myMessage.LotCode)
        End With

    End Using

End Sub

Private Sub WriteDetail(ByVal myMessage As it_Message)
    Dim doc As New XmlDocument

    doc.Load("test.xml")

    Dim Weight As XmlAttribute = doc.CreateAttribute("Weight")
```

```
Dim Weight2 As XmlAttribute = doc.CreateAttribute("Weight2")

Weight.InnerText = myMessage.Weight
Weight2.InnerText = myMessage.Weight2

Dim detail As XmlElement = doc.CreateElement("Detail")

detail.Attributes.Append(Weight)
detail.Attributes.Append(Weight2)

doc.DocumentElement.AppendChild(detail)

doc.Save("test.xml")

End Sub

Private Sub CloseHeader(ByVal myMessage As it_Message)
    MessageBox.Show("CloseFile")
End Sub

End Class
```

Table search

```
Imports System.Data.SqlClient

Public Class frmTableSearch_Full

    Private pstrFieldValue As String = ""
    Private pintFieldIndex As Integer = 0
    Private pblnFieldIndexVisible As Boolean = True
    Private pstrFormCaption As String = ""

    Private pblnMustCancel As Boolean
    Private pdtSearch As DataTable
    Private phashSQLs As Hashtable
    Private pBookMarkBGColor As Color = Color.Yellow
    Private pBookMarkBGColor_Selection As Color = Color.Aqua
    Private pBookMarkForeColor_Selection As Color = Color.Red

    Public Sub New()

        ' This call is required by the Windows Form Designer.
        InitializeComponent()

        ' Add any initialization after the InitializeComponent() call.
        InitControls()

    End Sub

    Public Sub New(ByVal dtSearch As DataTable, Optional ByVal strFormCaption As String = "", Optional ByVal intFieldIndex As Integer = 0, Optional ByVal blnFieldIndexVisible As Boolean = True)

        ' This call is required by the Windows Form Designer.
        InitializeComponent()

        pintFieldIndex = intFieldIndex
        pblnFieldIndexVisible = blnFieldIndexVisible
        pstrFormCaption = strFormCaption

        ' Add any initialization after the InitializeComponent() call.
        InitControls(dtSearch)

        Me.TopMost = True
    End Sub
End Class
```

```
End Sub

Public ReadOnly Property FormCaption() As String
    Get
        Return Me.pstrFormCaption
    End Get
End Property

Public ReadOnly Property FieldValue() As String
    Get
        Return Me.pstrFieldValue
    End Get
End Property

Public ReadOnly Property FieldIndex() As Integer
    Get
        Return Me.pintFieldIndex
    End Get
End Property

Public ReadOnly Property FieldIndexVisible() As Boolean
    Get
        Return Me.pblnFieldIndexVisible
    End Get
End Property

Public Property DataTable() As DataTable
    Get
        Return Me.pdtSearch
    End Get
    Set(ByVal value As DataTable)
        Me.pdtSearch = value
        FormatControls()
    End Set
End Property

Private Sub InitControls(Optional ByVal dtSearch As DataTable =
Nothing)

    pblnMustCancel = False

    If Me.FormCaption.Length > 0 Then Me.Text = Me.FormCaption
```



```
Me.cmbSearchTable.Items.AddRange(New String() {"<None>",
"<Αναζήτηση φιλτραρισμένου πίνακα>", "<Αναζήτηση σε όλους τους
πίνακες>", "Πελάτες", "Προϊόντα", "Παραγγελίες"})

phashSQLs = New Hashtable
phashSQLs.Add("Customers", "SELECT * FROM Customers")
phashSQLs.Add("Products", "SELECT * FROM Products")
phashSQLs.Add("Export Header", "SELECT * FROM Export_header")

'!!! DANGER !!! next line calls FormatControls, so it must be
called last
Me.DataTable = dtSearch

If Me.FieldIndex > Me.gridSearch.Columns.Count - 1 Then
pintFieldIndex = 0

End Sub

Private Sub gridSearch_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles gridSearch.KeyDown
If e.KeyCode = Keys.Return Then e.Handled = True
End Sub

Private Sub gridSearch_KeyUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles gridSearch.KeyUp
Select Case e.KeyCode
Case Keys.Return : FinishSelection()
Case Keys.Escape : FinishSelection(False)
End Select
End Sub

Private Sub gridSearch_MouseDoubleClick(ByVal sender As Object,
ByVal e As System.Windows.Forms.MouseEventArgs) Handles
gridSearch.MouseDoubleClick
FinishSelection()
End Sub

Private Sub FinishSelection(Optional ByVal blnSelect As Boolean =
True)
If blnSelect Then
pstrFieldValue =
Me.gridSearch.SelectedRows(0).Cells(Me.FieldIndex).Value
Me.DialogResult = Windows.Forms.DialogResult.OK
Else
```

```
        Me.DialogResult = Windows.Forms.DialogResult.None
    End If
    Me.Close()
End Sub

Private Sub FormatControls()

    If Me.DataTable Is Nothing Then

        Me.cmbSearchTable.SelectedIndex = 0
        Me.cmbSearchTable.Enabled = True

    Else

        Me.cmbSearchTable.SelectedIndex = 1
        Me.cmbSearchTable.Enabled = False

    End If

    FormatColumnHeaders()

End Sub

Private Sub cmbSearchTable_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs)

    Me.cmbSearchTable.SuspendLayout()

    If Me.DataTable Is Nothing And Me.cmbSearchTable.SelectedIndex
= 1 Then
        MsgBox("Δεν βρέθηκε φιλτραρισμένος πίνακας. Δεν θα γίνει
αναζήτηση")
        Me.cmdSearch.Enabled = False
        Me.gridSearch.Columns.Clear()
        Exit Sub
    End If

    Select Case Me.cmbSearchTable.SelectedIndex
        Case 0
            Me.cmdSearch.Enabled = False
        Case 1
            Me.cmdSearch.Enabled = True
        Case 2
```

```
        Me.cmdSearch.Enabled = False      'προσωρινά δεν θέλω να  
ψάχνει σε όλους τους πίνακες. Ίσως τελικά και να μην υπάρξει ποτέ αυτή  
η δυνατότητα
```

```
    Case Else
```

```
        Me.cmdSearch.Enabled = True
```

```
    End Select
```

```
    FormatColumnHeaders()
```

```
    Me.cmbSearchTable.ResumeLayout(False)
```

```
End Sub
```

```
Private Sub FormatColumnHeaders()
```

```
    Me.gridSearch.Columns.Clear()
```

```
    Select Case Me.cmbSearchTable.SelectedIndex
```

```
        Case 0      '<None>
```

```
            'already cleared
```

```
        Case 1      '<Search filtered table>
```

```
            For Each column As DataColumn In Me.DataTable.Columns
```

```
                Dim colIndex As Integer =
```

```
Me.gridSearch.Columns.Add(column.ColumnName, column.Caption)
```

```
                If colIndex = Me.FieldIndex And Not  
Me.FieldIndexVisible Then
```

```
                    Me.gridSearch.Columns(colIndex).Visible =  
False
```

```
                End If
```

```
            Next
```

```
        Case 2      '<Search all tables>
```

```
            Me.gridSearch.Columns.Add("Table", "Table")
```

```
            Me.gridSearch.Columns.Add("Field", "Field")
```

```
            Me.gridSearch.Columns.Add("Value", "Value")
```

```
        Case Else  'Specific table's column schema
```

```

        'take sql statement and concatenate a 'TOP 0'
statement in order to return no rows
        'because i only need to take the table's schema

        Dim sqlSelect As String =
CStr(phashSQLs(Me.cmbSearchTable.Text)).Replace("SELECT ", "SELECT TOP
0 ")

        Dim con As New SqlConnection(IT.A.ConnectionString)
        Dim com As New SqlCommand(sqlSelect, con)
        con.Open()

        Dim readerColumns As SqlDataReader = com.ExecuteReader
        Dim tableColumns As DataTable =
readerColumns.GetSchemaTable
        For i As Integer = 0 To tableColumns.Rows.Count - 1
            Dim colIndex As Integer =
Me.gridSearch.Columns.Add(tableColumns.Rows(i).Item("ColumnName"),
tableColumns.Rows(i).Item("ColumnName"))
            If colIndex = Me.FieldIndex And Not
Me.FieldIndexVisible Then
                Me.gridSearch.Columns(i).Visible = False
            End If
        Next
        readerColumns.Close()
        con.Close()

    End Select
End Sub

Private Sub cmbSearchWhat_TextChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles cmbSearchWhat.TextChanged
    Me.cmdSearch.Enabled = Me.cmbSearchWhat.Text.Length > 0 And
Me.cmbSearchTable.SelectedIndex <> 0 And
Me.cmbSearchTable.SelectedIndex <> 2

    'προσωρινά
End Sub

Private Sub cmdSearch_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmdSearch.Click

    Dim blnMustSearch As Boolean = False

```

```

blnMustSearch = Me.cmdSearch.Text = "Αναζήτηση"

Me.cmdSearch.Text = IIf(blnMustSearch, "Άκυρο", "Αναζήτηση")
Me.cmbSearchWhat.Enabled = Not blnMustSearch
Me.cmbSearchWhat.Visible = Not blnMustSearch
Me.pbarSearchProgress.Visible = blnMustSearch
Me.lblSearchProgressCounter.Visible = blnMustSearch

'Next statement is enough for the searching thread to
recognize user cancel request
'No other action for cancel is required here.
pblnMustCancel = Not blnMustSearch

If blnMustSearch Then

    Me.gridSearch.Rows.Clear()
    Me.gridSearch.Refresh()

    If
Me.cmbSearchWhat.Items.Contains(Me.cmbSearchWhat.Text)
Me.cmbSearchWhat.Items.Add(Me.cmbSearchWhat.Text)
Not
Then

        Dim strSearchWhat As String = Me.cmbSearchWhat.Text
        Dim blnSearchFilteredTable As Boolean =
Me.cmbSearchTable.SelectedIndex = 1
        Dim blnSearchAllTables As Boolean =
Me.cmbSearchTable.SelectedIndex = 2
        Dim blnMatchCase As Boolean = Me.chkMatchCase.Checked
        Dim blnMatchWholeWord As Boolean =
Me.chkMatchWholeWord.Checked

        Dim dtSearch As DataTable = pdtSearch

        If Not blnSearchFilteredTable And Not blnSearchAllTables
Then
            Dim daSearch As New SqlDataAdapter
            Dim sqlSelect As String =
CStr(phashSQLs(Me.cmbSearchTable.Text))
            daSearch.SelectCommand = New SqlCommand(sqlSelect)
            daSearch.SelectCommand.Connection =
New
SqlConnection(IT.A.ConnectionString)
            dtSearch = New DataTable(Me.cmbSearchTable.Text)
            daSearch.Fill(dtSearch)
        End If

```

```
Me.lblSearchProgressCounter.Visible = True

    Dim worker As New SearchTableDelegate(AddressOf
SearchTable)
    worker.BeginInvoke(strSearchWhat, dtSearch, blnMatchCase,
blnMatchWholeWord, AddressOf SearchTableCompleted, Nothing)

    End If

End Sub

Private Delegate Sub SearchTableDelegate(ByVal strSearchWhat As
String, ByVal LookInDataTable As DataTable, ByVal blnMatchCase As
Boolean, ByVal blnMatchWholeWord As Boolean)
Private Delegate Sub UpdateProgressDelegate(ByVal MaxValue As
Long)
Private Delegate Sub ReturnToMainThreadDelegate()
Private Delegate Sub AddRowDelegate(ByVal LookInDataTable As
DataTable, ByVal intRowIndex As Integer)
Private Delegate Sub BookMarkDelegate(ByVal colRowIndeces As
Collection)

Private Sub SearchTable(ByVal strSearchWhat As String, ByVal
LookInDataTable As DataTable, ByVal blnMatchCase As Boolean, ByVal
blnMatchWholeWord As Boolean)

    System.Threading.Thread.CurrentThread.Priority =
Threading.ThreadPriority.Lowest

    Dim blnMustBookmark As Boolean = False
    Dim intRowIndex As Integer = -1
    Dim intColumnIndex As Integer = -1
    Dim colRowIndeces As New Collection
    Dim colColumnIndeces As Collection

    For intRowIndex = 0 To LookInDataTable.Rows.Count - 1

        If pblnMustCancel Then Exit For

        Dim updateprogr As New UpdateProgressDelegate(AddressOf
UpdateProgress)
        Me.BeginInvoke(updateprogr, LookInDataTable.Rows.Count)
```

```

colColumnIndeces = New Collection
For intColumnIndex = 0 To LookInDataTable.Columns.Count -
1
    blnMustBookmark = False
    Dim strCurValue As String =
LookInDataTable.Rows(intRowIndex).Item(intColumnIndex).ToString
    Select Case True
        Case blnMatchCase And blnMatchWholeWord :
blnMustBookmark = strCurValue = strSearchWhat
        Case blnMatchCase And Not blnMatchWholeWord :
blnMustBookmark = strCurValue.Contains(strSearchWhat)
        Case Not blnMatchCase And blnMatchWholeWord :
blnMustBookmark = strCurValue.ToUpper = strSearchWhat.ToUpper
        Case Not blnMatchCase And Not blnMatchWholeWord :
blnMustBookmark = strCurValue.ToUpper.Contains(strSearchWhat.ToUpper)
    End Select
    If blnMustBookmark Then
colColumnIndeces.Add(intColumnIndex)
    Next
    If colColumnIndeces.Count > 0 Then
        Dim addrowdel As New AddRowDelegate(AddressOf AddRow)
        Me.BeginInvoke(addrowdel, LookInDataTable,
intRowIndex)

        colRowIndeces.Add(colColumnIndeces)
        If colRowIndeces.Count Mod 3 = 0 Then
System.Threading.Thread.Sleep(0)
        End If
    Next

    If colRowIndeces.Count > 0 Then
        Dim mark As New BookMarkDelegate(AddressOf BookMark)
        Me.BeginInvoke(mark, colRowIndeces)
    End If

End Sub

Private Sub AddRow(ByVal LookInDataTable As DataTable, ByVal
intRowIndex As Integer)
    System.Threading.Thread.CurrentThread.Priority =
Threading.ThreadPriority.Highest
    With Me.gridSearch
        Dim intNewRowIndex As Integer = .Rows.Add
        For intNewColumnIndex As Integer = 0 To .Columns.Count - 1

```

```

        .Rows (intNewRowIndex).Cells (intNewColumnIndex).Value =
LookInDataTable.Rows (intRowIndex).Item (intNewColumnIndex).ToString
    Next
End With
End Sub

Private Sub BookMark (ByVal colRowIndeces As Collection)

    Dim colColumnIndeces As Collection

    With Me.gridSearch.Rows
        For i As Integer = 1 To colRowIndeces.Count
            colColumnIndeces = CType (colRowIndeces (i), Collection)
            With .Item (i - 1).Cells
                For c As Integer = 1 To colColumnIndeces.Count
                    .Item (colColumnIndeces (c)).Style.BackColor =
pBookMarkBGColor

                    .Item (colColumnIndeces (c)).Style.SelectionBackColor =
pBookMarkBGColor_Selection

                    .Item (colColumnIndeces (c)).Style.Selectionforecolor =
pBookMarkForeColor_Selection
                Next
            End With
        Next
    End With

End Sub

Private Sub UpdateProgress (ByVal MaxValue As Long)
    If MaxValue <> Me.pbarSearchProgress.Maximum Then
        Me.lblSearchProgressCounter.Text = "0 of " & MaxValue
        Me.pbarSearchProgress.Value = 0
        Me.pbarSearchProgress.Minimum = 0
        Me.pbarSearchProgress.Maximum = MaxValue
        Me.pbarSearchProgress.Visible = True
    Else
        Me.pbarSearchProgress.Value += 1
        Me.lblSearchProgressCounter.Text =
Me.pbarSearchProgress.Value & " of " & Me.pbarSearchProgress.Maximum
    End If
End Sub

```



```
Private Sub SearchTableCompleted(ByVal ar As IAsyncResult)
    Dim ReturnToMain As New ReturnToMainThreadDelegate(AddressOf
ReturnToMainThread)
    Me.BeginInvoke(ReturnToMain)
End Sub

Private Sub ReturnToMainThread()

    Me.cmdSearch.Text = "Αναζήτηση"
    Me.cmdSearch.Enabled = True
    Me.cmbSearchWhat.Visible = True
    Me.cmbSearchWhat.Enabled = True
    Me.lblSearchProgressCounter.Visible = False
    Me.pbarSearchProgress.Visible = False
    Me.pbarSearchProgress.Value = 0
    pblnMustCancel = False
    Me.gridSearch.Focus()
End Sub

End Class
```

Order Form

```
Imports System.Data.SqlClient
```

```
Public Class frmOrderEdit
```

```
    Private isAdding As Boolean = False
```

```
    Private Const Default_Product_Description As String = "Επιλογή  
Προϊόντος"
```

```
    Private pblnReadOnlyOrder As Boolean = False
```

```
    Private pblnIsLoaded As Boolean = False
```

```
    Private pblnAllowCellValueChanged As Boolean = False
```

```
    Private intCountUOMs As Integer = 0
```

```
    Private sqlSELECT_Headers As String = _
        "SELECT  Export_header.Order_no,  Export_header.Customer_code,
Customers.Name      AS      Customer_name,      Customers.Address      AS
Customer_address,
Export_header.Destination,
Export_header.Export_date,      Export_header.Update_date,
Export_header.Comments,      Export_header.Next_Box_no,
Export_header.Itinerary,      Export_header.Status,      CASE      WHEN
Export_header.Status      = 0      THEN      '(0)      ΝΕΑ      ΠΑΡΑΓΓΕΛΙΑ'      WHEN
Export_header.Status      = 1      THEN      '(1)      ΥΠΟ      ΕΚΤΕΛΕΣΗ'      WHEN
Export_header.Status= 2 THEN '(2) ΟΛΟΚΛΗΡΩΜΕΝΗ' ELSE '<Unknown>' END +
' (Εισήχθη: ' + CONVERT(nvarchar, Export_header.Transfer_date, 103) +
')' AS Status2 " & _
        "FROM      Export_header      INNER      JOIN      Customers      ON
Export_header.Customer_code      =      Customers.Code      WHERE
(Export_header.Order_no = @OrderNo)"
```

```
    Private sqlSELECT_Lines As String = _
        "SELECT
Export_lines.Additional_product,
Export_lines.Product_code,      Products.Descr      AS      Product_descr,
Export_lines.Order_no_qty,      Export_lines.UOM_O,      CASE      WHEN
Products.Sale_per = 1 THEN Export_lines.Export_qty ELSE 0 END AS
Export_qty_Items,      CASE      WHEN      Products.Sale_per      = 2      THEN
Export_lines.Export_qty      ELSE      0      END      AS      Export_qty_Weight,
Export_lines.Comments,      Export_lines.ID,      Export_lines.Order_no,
Products.Sale_per, Products.Status " & _
        "FROM      Export_lines      INNER      JOIN      Products      ON
Export_lines.Product_code = Products.Code LEFT OUTER JOIN (SELECT
Cust_packaging.ID,      Cust_packaging.Customer_code,
Cust_packaging.Product_code,      Cust_packaging.Packaging_code,
```

```

Cust_packaging.Pieces,          Cust_packaging.Est_weight,
Cust_packaging.UOM_W,          Cust_packaging.Min_life,
Cust_packaging.Time_periods_ID FROM Export_header INNER JOIN
Cust_packaging ON Export_header.Customer_code =
Cust_packaging.Customer_code WHERE (Export_header.Order_no =
@OrderNo)) AS Q_Cust_packaging ON Export_lines.Product_code =
Q_Cust_packaging.Product_code WHERE Export_lines.Order_no = @OrderNo"

```

```
Private dsOrderEdit As DataSet
```

```
Private dtHeaders As DataTable
```

```
Private dtLines As DataTable
```

```
Private daHeaders As SqlDataAdapter
```

```
Private daLines As SqlDataAdapter
```

```
Public Sub New()
```

```
' This call is required by the Windows Form Designer.
```

```
InitializeComponent()
```

```
' Add any initialization after the InitializeComponent() call.
```

```
Me.BindHeader()
```

```
Me.Fill()
```

```
End Sub
```

```
Public Sub New(ByVal strOrderNo As String)
```

```
' This call is required by the Windows Form Designer.
```

```
InitializeComponent()
```

```
' Add any initialization after the InitializeComponent() call.
```

```
Me.BindHeader()
```

```
Me.Fill()
```

```
Me.ts_txtOrderNo.Text = strOrderNo
```

```
End Sub
```

```
Private Sub BindHeader()
```

```
Me.Header_txtOrderNo.DataBindings.Add(New Binding("Text",
Me.bsHeaders, "Order_no", True))
```

```
Me.Header_txtOrderStatus.DataBindings.Add(New Binding("Text",  
Me.bsHeaders, "Status2", True))  
Me.Header_txtCustomerCode.DataBindings.Add(New Binding("Text",  
Me.bsHeaders, "Customer_code", True))  
Me.Header_txtCustomerName.DataBindings.Add(New Binding("Text",  
Me.bsHeaders, "Customer_name", True))  
Me.Header_txtCustomerAddress.DataBindings.Add(New  
Binding("Text", Me.bsHeaders, "Customer_address", True))  
Me.Header_txtComments.DataBindings.Add(New Binding("Text",  
Me.bsHeaders, "Comments", True))  
Me.Header_txtExportDate.DataBindings.Add(New Binding("Text",  
Me.bsHeaders, "Export_date", True))  
Me.Header_txtUpdateDate.DataBindings.Add(New Binding("Text",  
Me.bsHeaders, "Update_date", True))  
Me.Header_txtDestination.DataBindings.Add(New Binding("Text",  
Me.bsHeaders, "Destination", True))  
Me.Header_txtItinerary.DataBindings.Add(New Binding("Text",  
Me.bsHeaders, "Itinerary", True))  
Me.Header_txtNextBoxNo.DataBindings.Add(New Binding("Text",  
Me.bsHeaders, "Next_Box_no", True))
```

```
End Sub
```

```
Private Sub Fill(Optional ByVal strOrderNo As String = "")  
  
pblnIsLoaded = False  
  
dsOrderEdit = New DataSet("dsOrderEdit")  
dtHeaders = Nothing  
dtLines = Nothing  
daHeaders = Nothing  
daLines = Nothing  
  
LoadTable(strOrderNo, "Export_headers", sqlSELECT_Headers, "",  
"", "", dtHeaders, daHeaders)  
LoadTable(strOrderNo, "Export_lines", sqlSELECT_Lines,  
"Order_no", "Export_headers", "Order_no", dtLines, daLines)  
  
Me.bsHeaders.DataSource = dsOrderEdit  
Me.bsHeaders.DataMember = "Export_headers"  
  
Me.bsLines.DataSource = dsOrderEdit  
Me.bsLines.DataMember = "Export_lines"
```

```
Lines_Grid_LoadUOM()  
  
Me.Lines_Grid.DataSource = Me.bsLines  
  
pblnIsLoaded = Me.Lines_Grid.Rows.Count > 1  
  
End Sub  
  
Private Sub LoadTable(ByVal strOrderNo As String, ByVal table_name  
As String, ByVal select_statement As String, _  
    ByVal child_column_name As String, ByVal parent_table_name As  
String, ByVal parent_column_name As String, _  
    ByRef data_table As DataTable, ByRef data_adapter As  
SqlDataAdapter)  
  
    Dim con As New SqlConnection(IT.A.ConnectionString)  
    Dim com As New SqlCommand(select_statement, con)  
  
    com.Parameters.Add("@OrderNo", SqlDbType.NVarChar)  
    com.Parameters(0).Value = strOrderNo  
  
    data_adapter = New SqlDataAdapter(com)  
  
    data_adapter.TableMappings.Add("Table", table_name)  
    data_adapter.Fill(dsOrderEdit)  
  
    data_table = dsOrderEdit.Tables(table_name)  
  
    If child_column_name.Length > 0 Then  
        Dim parent_table As DataTable =  
dsOrderEdit.Tables(parent_table_name)  
        Dim foreign_key As New  
ForeignKeyConstraint(parent_table.Columns(parent_column_name),  
data_table.Columns(child_column_name))  
        data_table.Constraints.Add(foreign_key)  
    End If  
  
End Sub  
  
Private Sub Lines_Grid_LoadUOM()  
  
    Dim daUOMLookUp As New SqlDataAdapter  
    Dim dtUOMLookUp As New DataTable("dtUOMLookUp")  
    Dim con As New SqlConnection(IT.A.ConnectionString)
```

```
Dim com As New SqlCommand("SELECT Count(*) AS CountUOMs FROM
UOM", con)

con.Open()
intCountUOMs = CInt(com.ExecuteScalar)

daUOMLookUp.SelectCommand = New SqlCommand("SELECT Cast(Code
AS tinyint) AS UOM_CODE, Descr FROM UOM", con)
daUOMLookUp.Fill(dtUOMLookUp)

Me.Lines_Grid_UOM_O.DataSource = dtUOMLookUp
Me.Lines_Grid_UOM_O.ValueMember = "UOM_CODE"
Me.Lines_Grid_UOM_O.DisplayMember = "Descr"

con.Close()

End Sub

Private Sub Form_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

Me.Show()
If Me.ts_txtOrderNo.Text.Length > 0 Then
Me.ts_cmdOrderNo.PerformClick()
If pblnIsLoaded Then
Me.PrepareForm(it_PrepareFormEnum.EditOrder)
Else
Me.PrepareForm(it_PrepareFormEnum.SearchOrder)
End If
End Sub

Private Sub Form_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
If Not RestoreDirty(True, True) Then e.Cancel = True
End Sub

Private Sub Form_KeyUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles Me.KeyUp
Select Case e.KeyCode
Case Keys.Escape : If Me.ts_txtOrderNo.Selected Then
Me.ts_txtOrderNo.SelectAll()
Case Keys.F1 And Not e.Shift :
Me.ts_cmdSave.PerformClick()
```

```

        Case Keys.F2 And Not e.Shift :
Me.ts_cmdCancel.PerformClick()
        Case Keys.F3 And Not e.Shift :
Me.ts_cmdDeleteProduct.PerformClick()
        Case Keys.F4 And Not e.Shift :
Me.ts_cmdAddProduct.PerformClick()
        Case Keys.F5 And Not e.Shift :
Me.ts_cmdSearchProduct.PerformClick()
        Case Keys.F6 And Not e.Shift
        Case Keys.F7 And Not e.Shift
        Case Keys.F8 And Not e.Shift
        Case Keys.F9 And Not e.Shift
        Case Keys.F10 And Not e.Shift :
Me.ts_cmdOrderNo.PerformClick() : e.SuppressKeyPress = True
        Case Keys.F10 And e.Shift :
Me.ts_cmdRefresh.PerformClick() : e.SuppressKeyPress = True
    End Select
End Sub

```

```

Private Sub PrepairForm(ByVal MyFormStatus As it_PreparFormEnum)
    Select Case MyFormStatus
        Case it_PreparFormEnum.SearchOrder

            Me.ts_txtOrderNo.Text = ""
            Me.ts_txtOrderNo.Enabled = True
            Me.ts_cmdSave.Visible = False
            Me.ts_cmdCancel.Visible = False
            Me.ts_cmdDeleteProduct.Visible = False
            Me.ts_cmdAddProduct.Visible = False
            Me.ts_cmdSearchProduct.Visible = False
            Me.ts_cmdRefresh.Visible = False
            Me.Header_panel.Enabled = False
            Me.Lines_Grid.Enabled = False

            Me.ts_txtOrderNo.Focus()

        Case it_PreparFormEnum.EditOrder

            Me.ts_txtOrderNo.Enabled = False
            Me.ts_cmdSave.Visible = Not pblnReadOnlyOrder
            Me.ts_cmdCancel.Visible = Not pblnReadOnlyOrder
            Me.ts_cmdDeleteProduct.Visible = Not pblnReadOnlyOrder

```

```
Me.ts_cmdAddProduct.Visible = Not pblnReadOnlyOrder
Me.ts_cmdSearchProduct.Visible = Not pblnReadOnlyOrder
Me.ts_cmdRefresh.Visible = Not pblnReadOnlyOrder
Me.Header_panel.Enabled = True

Me.Header_txtComments.ReadOnly = pblnReadOnlyOrder
Me.Header_txtDestination.ReadOnly = pblnReadOnlyOrder
Me.Header_txtItinerary.ReadOnly = pblnReadOnlyOrder

Me.Header_txtComments.TabStop = Not pblnReadOnlyOrder
Me.Header_txtDestination.TabStop = Not
pblnReadOnlyOrder
Me.Header_txtItinerary.TabStop = Not pblnReadOnlyOrder

Me.Lines_Grid.Enabled = True
Me.Lines_Grid.ReadOnly = pblnReadOnlyOrder
Me.Lines_Grid.AllowUserToAddRows = Not
pblnReadOnlyOrder

Me.Lines_Grid.Focus()

Case Else

IT.Functions.RaiseError("Invalid parameter passed in
PrepairForm", Me)

End Select
End Sub

Private Sub ts_cmdOrderNo_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ts_cmdOrderNo.Click

If Me.ts_cmdOrderNo.Text = "(F10) Παραγγελία" Then
If Me.ts_txtOrderNo.Text.Length = 0 Then
Dim sqlSearchOrder As String = _
"SELECT Export_header.Order_no AS Παραγγελία,
Export_header.Customer_code AS [Κωδικός Πελάτη], Customers.Name AS
[Επωνυμία Πελάτη], CASE WHEN Export_header.Status = 0 THEN '(0) ΝΕΑ
ΠΑΡΑΓΓΕΛΙΑ' WHEN Export_header.Status = 1 THEN '(1) ΥΠΟ ΕΚΤΕΛΕΣΗ' WHEN
Export_header.Status = 2 THEN '(2) ΟΛΟΚΛΗΡΩΜΕΝΗ' ELSE '<Unknown>' END
```



```

AS [Κατάσταση Παραγγελίας], Export_header.Transfer_date AS [Ημερομηνία
Εισαγωγής] " & _
        "FROM Export_header INNER JOIN Customers ON
Export_header.Customer_code = Customers.Code " & _
        "ORDER BY [Κατάσταση Παραγγελίας], [Ημερομηνία
Εισαγωγής] DESC, Παραγγελία"
        Dim strResult As String =
IT.Functions.SearchTable_Light(sqlSearchOrder, "Αναζήτηση
Παραγγελίας", , 750)
        If strResult.Length > 0 Then Me.ts_txtOrderNo.Text =
strResult
        End If

        If Me.ts_txtOrderNo.Text.Length = 0 Then Exit Sub

        If IT.DB.OrderHeaders.Exists(Me.ts_txtOrderNo.Text) Then

            Dim order As IT.DB.OrderHeader =
IT.DB.OrderHeaders.GetOrder(Me.ts_txtOrderNo.Text)

            If order.Status.HasValue Then
                If order.Status.Value >= 2 Then 'η παραγγελία έχει
κλείσει
                    pblnReadOnlyOrder = True
                Else
                    pblnReadOnlyOrder = False
                End If
            Else
                pblnReadOnlyOrder = True
            End If

            Me.Fill(Me.ts_txtOrderNo.Text)
            Me.PrepareForm(it_PrepareFormEnum.EditOrder)
            PaintLines()
            If Not pblnReadOnlyOrder Then
IT.Functions.DisplayProductsThatNeedEdit(Me.ts_txtOrderNo.Text)
                If pblnReadOnlyOrder Then IT.Functions.RaiseError("Η
παραγγελία [" & Me.ts_txtOrderNo.Text & "] έχει κλείσει και είναι μόνο
για ανάγνωση.", Me)
            End If
            If order.HasRecalledLots Then
IT.Functions.DisplayProductsWithRecalledLots(Me.ts_txtOrderNo.Text)

            Me.ts_cmdOrderNo.Text = "(F10) Καθάρισμα"

```

```
Else
    IT.Functions.RaiseError("Η παραγγελία [" &
Me.ts_txtOrderNo.Text & "] δεν βρέθηκε.", Me)
    Me.PrepareForm(it_PrepareFormEnum.SearchOrder)
End If

Else

If RestoreDirty(True, True) Then
    Me.Fill()
    PrepareForm(it_PrepareFormEnum.SearchOrder)
    Me.ts_cmdOrderNo.Text = "(F10) Παραγγελία"
End If

End If

End Sub

Private Sub ts_cmdSave_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ts_cmdSave.Click
    RestoreDirty(True, False)
End Sub

Private Sub ts_cmdCancel_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ts_cmdCancel.Click
    RestoreDirty(False)
End Sub

Private Sub ts_cmdDeleteProduct_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ts_cmdDeleteProduct.Click
    If CanDeleteProduct(Me.Lines_Grid.CurrentRow) Then
Me.Lines_Grid.Rows.Remove(Me.Lines_Grid.CurrentRow)
    End Sub

Private Sub ts_cmdAddProduct_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ts_cmdAddProduct.Click
    Me.Lines_Grid.CurrentRow =
Me.Lines_Grid.Rows(Me.Lines_Grid.NewRowIndex).Cells(1)
    Me.Lines_Grid.CurrentRow.Selected = True
End Sub
```

```
Private Sub ts_cmdSearchProduct_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ts_cmdSearchProduct.Click
    Dim strSQLStatement As String = "SELECT Products.Code AS
Κωδικός, Products.Descr AS Περιγραφή, UOM.Descr AS [ΜΜ Πώλησης],
UOM_1.Descr AS [Τρόπος Πώλησης] FROM Products INNER JOIN UOM ON
Products.UOM_S = UOM.Code INNER JOIN UOM AS UOM_1 ON Products.Sale_per
= UOM_1.Code WHERE Products.Status = 'True' OR Products.Status IS NULL
ORDER BY Κωδικός"
    Dim strResult As String =
IT.Functions.SearchTable_Light(strSQLStatement, "Αναζήτηση Προϊόντος")
    If strResult.Length > 0 Then
        If Me.Lines_Grid.CurrentRow.IsNewRow Then
            isAdding = True
            Me.bsLines.CancelEdit()
            Me.bsLines.AddNew()
            Me.bsLines.MoveLast()
            isAdding = False
            Dim rowAdded As DataGridViewRow =
Me.Lines_Grid.Rows(Me.Lines_Grid.Rows.Count - 2)
            Lines_Grid_SetDefaultValues(rowAdded)
        End If
        Me.Lines_Grid.EndEdit()

Me.Lines_Grid.CurrentRow.Cells("Lines_Grid_Product_code").Value =
strResult

Me.Lines_Grid.CurrentRow.Cells("Lines_Grid_Order_no_qty").Selected =
True

        Me.ts_cmdAddProduct.Enabled = False
        Me.ts_cmdDeleteProduct.Enabled = False
        Me.ts_cmdSearchProduct.Enabled = False
        Me.ts_cmdRefresh.Enabled = False
    End If
End Sub

Private Sub ts_cmdRefresh_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ts_cmdRefresh.Click
    If Me.Header_txtOrderNo.Text.Length > 0 Then If
RestoreDirty(True, True) Then Me.Fill(Me.Header_txtOrderNo.Text)
    End Sub
```

```

Private Sub Lines_Grid_DataError(ByVal sender As Object, ByVal e
As System.Windows.Forms.DataGridViewDataErrorEventArgs) Handles
Lines_Grid.DataError
    Dim strMSG As String = "Invalid data in column " +
e.ColumnIndex.ToString + " of row " + e.RowIndex.ToString + ". Press
Esc to cancel then edit or enter an appropriate value."
    MsgBox(strMSG, MsgBoxStyle.Exclamation, "Data entry error")
End Sub

Private Sub Lines_Grid_SelectionChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles Lines_Grid.SelectionChanged
    If isAdding Then
        'dummy block
    ElseIf pblnIsLoaded And Not isAdding Then
        With Me.Lines_Grid.CurrentRow.Cells
            Dim blnProductDeleted As Boolean = False
            If Not IsDBNull(.Item("Lines_Grid_Status").Value) Then
                blnProductDeleted = Not
CBool(.Item("Lines_Grid_Status").Value) And Not
Me.Lines_Grid.CurrentRow.IsNewRow
            End If
            .Item("Lines_Grid_Product_code").ReadOnly = Not
(.Item("Lines_Grid_Product_descr").Value =
Default_Product_Description) Or blnProductDeleted
            .Item("Lines_Grid_Order_no_qty").ReadOnly =
blnProductDeleted
            .Item("Lines_Grid_UOM_O").ReadOnly = blnProductDeleted
            .Item("Lines_Grid_Comments").ReadOnly =
blnProductDeleted
            .Item("Lines_Grid_Product_descr").ReadOnly = True
            .Item("Lines_Grid_Export_qty_Items").ReadOnly = True
            .Item("Lines_Grid_Export_qty_Weight").ReadOnly = True
            Me.ts_cmdAddProduct.Enabled = Not
Me.Lines_Grid.CurrentRow.IsNewRow
            Me.ts_cmdDeleteProduct.Enabled = Not
Me.Lines_Grid.CurrentRow.IsNewRow And
.Item("Lines_Grid_Additional_product").Value = True
            Me.ts_cmdSearchProduct.Enabled = Not
.Item("Lines_Grid_Product_code").ReadOnly
            Me.ts_cmdRefresh.Enabled = True
        End With
    Else
        Me.ts_cmdAddProduct.Enabled = True
        Me.ts_cmdDeleteProduct.Enabled = False
    End If
End Sub

```

```

        Me.ts_cmdSearchProduct.Enabled = False
        Me.ts_cmdRefresh.Enabled = True
    End If
End Sub

Private Sub Lines_Grid_DefaultValuesNeeded(ByVal sender As Object,
ByVal e As System.Windows.Forms.DataGridViewRowEventArgs) Handles
Lines_Grid.DefaultValuesNeeded
    Lines_Grid_SetDefaultValues(e.Row)
End Sub

Private Sub Lines_Grid_SetDefaultValues(ByVal Row As
DataGridViewRow)
    With Row
        pblnAllowCellValueChanged = False
        .Cells("Lines_Grid_Additional_Product").Value = True
        .Cells("Lines_Grid_Product_code").Value = "0"
        .Cells("Lines_Grid_Product_descr").Value =
Default_Product_Description
        .Cells("Lines_Grid_Order_no_qty").Value = 0
        .Cells("Lines_Grid_UOM_0").Value = 1
        .Cells("Lines_Grid_Export_qty_Items").Value = 0
        .Cells("Lines_Grid_Export_qty_Weight").Value = 0
        .Cells("Lines_Grid_Comments").Value = ""
        .Cells("Lines_Grid_Sale_per").Value = 1
        .Cells("Lines_Grid_Order_no").Value =
Me.ts_txtOrderNo.Text
        pblnAllowCellValueChanged = True
    End With
End Sub

Private Sub Lines_Grid_CellValueChanged(ByVal sender As Object,
ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles
Lines_Grid.CellValueChanged

    If Not pblnIsLoaded Or Not pblnAllowCellValueChanged Then Exit
Sub

    If e.ColumnIndex =
Me.Lines_Grid.Rows(e.RowIndex).Cells("Lines_Grid_Product_code").Column
Index Then

        Dim strNewProductCode As String =
Me.Lines_Grid.Rows(e.RowIndex).Cells("Lines_Grid_Product_code").Value

```

```

        Dim prod As IT.DB.Product =
IT.DB.Products.GetProduct(strNewProductCode)

Me.Lines_Grid.Rows(e.RowIndex).Cells("Lines_Grid_Product_descr").Value
= prod.Description
    If prod.SalePer.HasValue Then

Me.Lines_Grid.Rows(e.RowIndex).Cells("Lines_Grid_Sale_per").Value =
prod.SalePer
        If intCountUOMs >= prod.SalePer.Value Then

Me.Lines_Grid.Rows(e.RowIndex).Cells("Lines_Grid_UOM_O").Value =
prod.SalePer
            End If
        End If
    End If

End Sub

Private Sub Lines_Grid_UserDeletingRow(ByVal sender As Object,
ByVal e As System.Windows.Forms.DataGridViewRowCancelEventArgs)
Handles Lines_Grid.UserDeletingRow
    e.Cancel = Not CanDeleteProduct(e.Row)
End Sub

Private Sub Lines_Grid_CellValidating(ByVal sender As Object,
ByVal e As System.Windows.Forms.DataGridViewCellValidatingEventArgs)
Handles Lines_Grid.CellValidating
    Try
        With Me.Lines_Grid.Rows(e.RowIndex)
            If e.ColumnIndex = 1 And Not
Me.Lines_Grid.CurrentRow.IsNewRow Then
                Dim strNewProductCode As String =
.Cells("Lines_Grid_Product_code").EditedFormattedValue
                If Not IT.DB.Products.Exists(strNewProductCode)
Then
                    .ErrorText = "Το προϊόν [" & strNewProductCode
& "] δεν υπάρχει"
                    IT.Functions.RaiseError(.ErrorText, Me)
                    e.Cancel = True
                ElseIf
IT.DB.Products.GetProduct(strNewProductCode).Status.HasValue Then

```

```

        If
CBool (IT.DB.Products.GetProduct(strNewProductCode).Status.Value) =
False Then
            .ErrorText = "Το προϊόν [" &
strNewProductCode & "] είναι απενεργοποιημένο."
            IT.Functions.RaiseError(.ErrorText, Me)
            e.Cancel = True
        End If
    End If
    ElseIf e.ColumnIndex = 3 Then 'Lines_Grid_Order_no_qty
        If Not IsNumeric(e.FormattedValue) Then
            .ErrorText = "Το πεδίο [Ποσότητα] δέχεται μόνο
αριθμούς."
            IT.Functions.RaiseError(.ErrorText, Me)
            e.Cancel = True
        ElseIf e.FormattedValue > 99999.99 Then
            .ErrorText = "Το πεδίο [Ποσότητα] δέχεται
αριθμό ΟΧΙ μεγαλύτερο από [99.999,99]."
            IT.Functions.RaiseError(.ErrorText, Me)
            e.Cancel = True
        ElseIf e.FormattedValue < 0 Then
            .ErrorText = "Το πεδίο [Ποσότητα] δέχεται μόνο
θετικούς αριθμούς."
            IT.Functions.RaiseError(.ErrorText, Me)
            e.Cancel = True
        End If
    End If
End With
Catch ex As Exception
    MsgBox(ex.Message)
End Try
End Sub

Private Sub Lines_Grid_CellEndEdit(ByVal sender As Object, ByVal e
As System.Windows.Forms.DataGridViewCellEventArgs) Handles
Lines_Grid.CellEndEdit
    Me.Lines_Grid.Rows(e.RowIndex).ErrorText = ""
End Sub

Private Sub Lines_Grid_RowValidating(ByVal sender As Object, ByVal
e As System.Windows.Forms.DataGridViewCellCancelEventArgs) Handles
Lines_Grid.RowValidating
    If Me.Lines_Grid.CurrentRow Is Nothing Then Exit Sub
    If Me.Lines_Grid.IsCurrentRowDirty Then

```

```

With Me.Lines_Grid.CurrentRow

        Dim      strCustomerCode      As      String      =
Me.Header_txtCustomerCode.Text
        Dim      strProductCode      As      String      =
CStr(.Cells("Lines_Grid_Product_code").Value)
        Dim      decOrder_no_qty      As      Decimal      =
CDec(.Cells("Lines_Grid_Order_no_qty").Value)
        Dim      intUOM_O      As      Integer      =
CInt(.Cells("Lines_Grid_UOM_O").Value)
        Dim      intSalesPer      As      Integer      =
CInt(.Cells("Lines_Grid_Sale_per").Value)
        Dim      intExport_qty_Items      As      Integer      =
CInt(.Cells("Lines_Grid_Export_qty_Items").Value)
        Dim      decExport_qty_Weight      As      Decimal      =
CDec(.Cells("Lines_Grid_Export_qty_Weight").Value)

        If Not IT.DB.Products.Exists(strProductCode) Then
            .ErrorText = "Το προϊόν [" & strProductCode & "]
δεν υπάρχει"
            .Cells("Lines_Grid_Product_descr").Value =
Default_Product_Description
            e.Cancel = True
        Else
            If ProductExists(strProductCode) Then
                .ErrorText = "Το προϊόν [" & strProductCode &
"] υπάρχει ήδη στην παραγγελία [" & Me.ts_txtOrderNo.Text & "]"
                .Cells("Lines_Grid_Product_descr").Value =
Default_Product_Description
                e.Cancel = True
            End If
        End If

        If Not e.Cancel Then

            If decOrder_no_qty < 0 Then
                .ErrorText = "Το πεδίο [Ποσότητα] πρέπει να
είναι μεγαλύτερο του μηδενός."
                e.Cancel = True
            ElseIf intUOM_O = intSalesPer And intUOM_O <> 3
Then 'αν υπάρχει αντιστοίχιση των MM, οπότε και δεν έχουμε να κάνουμε
με μεγαλομονάδα

                Select Case intSalesPer
                    Case 1

```



```

                                If                decOrder_no_qty                <
intExport_qty_Items Then
                                .ErrorText = "Το πεδίο [Ποσότητα]
πρέπει να είναι μεγαλύτερο από τα εκτελεσμένα τεμάχια."
                                e.Cancel = True
                                End If
                                Case 2
                                If                decOrder_no_qty                <
decExport_qty_Weight Then
                                .ErrorText = "Το πεδίο [Ποσότητα]
πρέπει να είναι μεγαλύτερο από το εκτελεσμένο βάρος."
                                e.Cancel = True
                                End If
                                End Select
                                ElseIf intUOM_0 = 3 Then
                                Dim    packagingQTY    As    Decimal    =
GetPackagingQTY(strCustomerCode, strProductCode, intSalesPer)
                                Select Case intSalesPer
                                Case 1
                                If packagingQTY * decOrder_no_qty <
intExport_qty_Items Then
                                .ErrorText = "Το πεδίο [Ποσότητα]
πρέπει να είναι μεγαλύτερο από τα εκτελεσμένα τεμάχια."
                                e.Cancel = True
                                End If
                                Case 2
                                If packagingQTY * decOrder_no_qty <
decExport_qty_Weight Then
                                .ErrorText = "Το πεδίο [Ποσότητα]
πρέπει να είναι μεγαλύτερο από το εκτελεσμένο βάρος."
                                e.Cancel = True
                                End If
                                End Select
                                End If
                                End If
                                If e.Cancel Then IT.Functions.RaiseError(.ErrorText,
Me)
                                End With
                                End If
                                End Sub

```

```
Private Sub Lines_Grid_RowValidated(ByVal sender As Object, ByVal  
e As System.Windows.Forms.DataGridViewCellEventArgs) Handles  
Lines_Grid.RowValidated  
    If Not Me.Lines_Grid.CurrentRow Is Nothing Then  
Me.Lines_Grid.CurrentRow.ErrorText = ""  
    End Sub
```

```
Private Function RestoreDirty(ByVal blnSaveChanges As Boolean,  
Optional ByVal AskBeforeSaving As Boolean = False) As Boolean
```

```
    If Not pblnIsLoaded Then Return True  
  
    If blnSaveChanges Then  
        Try  
            Me.Lines_Grid.CurrentRow.Cells(0) =  
Me.Lines_Grid.Rows(0).Cells(0)  
            Me.Lines_Grid.EndEdit()  
        Catch ex As Exception  
            Return False  
        End Try  
    Else  
        Me.Lines_Grid.CancelEdit()  
        Me.Lines_Grid.CurrentRow.ErrorText = ""  
    End If
```

```
    Me.bsHeaders.CurrencyManager.EndCurrentEdit()  
    Me.bsLines.CurrencyManager.EndCurrentEdit()
```

```
    If Not dsOrderEdit.HasChanges Then Return True
```

```
    Dim blnAcceptChanges As Boolean = False
```

```
    If blnSaveChanges Then  
        If AskBeforeSaving Then  
            Dim result As Microsoft.VisualBasic.MsgBoxResult =  
MsgBox(IT.A.Language.Phrase(64, "Υπάρχουν αλλαγές στα δεδομένα. Θέλετε  
να τις αποθηκεύσετε;"), MsgBoxStyle.YesNoCancel)  
            Select Case result  
                Case MsgBoxResult.Yes  
                    blnAcceptChanges = True  
                Case MsgBoxResult.No  
                    blnAcceptChanges = False
```

```
        Case MsgBoxResult.Cancel
            Return False
        End Select
    Else
        blnAcceptChanges = True
    End If
Else
    blnAcceptChanges = False
End If

UpdateChanges (blnAcceptChanges)

Return True

End Function

Private Sub UpdateChanges (ByVal blnAcceptChanges As Boolean)

    If Not blnAcceptChanges Then
        dsOrderEdit.RejectChanges()
    Else
        Me.Header_txtUpdateDate.Text = IT.Functions.GetServerDate

Me.Header_txtUpdateDate.DataBindings.Item("Text").WriteValue()
        Me.bsHeaders.CurrencyManager.EndCurrentEdit()

        Dim dtChanges_Deleted As DataTable =
Me.dsOrderEdit.Tables("Export_lines").GetChanges(DataRowState.Deleted)
        If Not dtChanges_Deleted Is Nothing Then
            Dim comDelete As New SqlCommand("DELETE FROM
Export_lines WHERE (Order_no = @Order_no) AND (Product_code =
@Product_code)", New SqlConnection(IT.A.ConnectionString))

comDelete.Parameters.Add(IT.Functions.GetParameter("@Order_no",
SqlDbType.NVarChar, "Order_no"))

comDelete.Parameters.Add(IT.Functions.GetParameter("@Product_code",
SqlDbType.NVarChar, "Product_code"))
            daLines.DeleteCommand = comDelete
        End If

        Dim dtChanges_Added As DataTable =
Me.dsOrderEdit.Tables("Export_lines").GetChanges(DataRowState.Added)
        If Not dtChanges_Added Is Nothing Then
```

```
Dim comInsert As New SqlCommand("INSERT INTO
Export_lines (Order_no, Product_code, Order_no_qty, Export_qty,
Additional_product, Comments, UOM_O) VALUES
(@Order_no,@Product_code,@Order_no_qty,0,@Additional_product,@Comments
,@UOM_O)", New SqlConnection(IT.A.ConnectionString))

comInsert.Parameters.Add(IT.Functions.GetParameter("@Order_no",
SqlDbType.NVarChar, "Order_no"))

comInsert.Parameters.Add(IT.Functions.GetParameter("@Product_code",
SqlDbType.NVarChar, "Product_code"))

comInsert.Parameters.Add(IT.Functions.GetParameter("@Order_no_qty",
SqlDbType.Decimal, "Order_no_qty"))

comInsert.Parameters.Add(IT.Functions.GetParameter("@Additional_product",
SqlDbType.Bit, "Additional_product"))

comInsert.Parameters.Add(IT.Functions.GetParameter("@Comments",
SqlDbType.NVarChar, "Comments", True))

comInsert.Parameters.Add(IT.Functions.GetParameter("@UOM_O",
SqlDbType.TinyInt, "UOM_O"))
daLines.InsertCommand = comInsert
End If

Dim dtChanges_Modified As DataTable =
Me.dsOrderEdit.Tables("Export_lines").GetChanges(DataRowState.Modified
)

If Not dtChanges_Modified Is Nothing Then
Dim comUpdate As New SqlCommand("UPDATE Export_lines
SET Order_no = @Order_no, Product_code = @Product_code, Order_no_qty =
@Order_no_qty, Additional_product = @Additional_product, Comments =
@Comments, UOM_O = @UOM_O WHERE Order_no = @Order_no AND Product_code
= @Product_code", New SqlConnection(IT.A.ConnectionString))

comUpdate.Parameters.Add(IT.Functions.GetParameter("@Order_no",
SqlDbType.NVarChar, "Order_no"))

comUpdate.Parameters.Add(IT.Functions.GetParameter("@Product_code",
SqlDbType.NVarChar, "Product_code"))

comUpdate.Parameters.Add(IT.Functions.GetParameter("@Order_no_qty",
SqlDbType.Decimal, "Order_no_qty"))
```

```
comUpdate.Parameters.Add(IT.Functions.GetParameter("@Additional_produc
t", SqlDbType.Bit, "Additional_product"))

comUpdate.Parameters.Add(IT.Functions.GetParameter("@Comments",
SqlDbType.NVarChar, "Comments", True))

comUpdate.Parameters.Add(IT.Functions.GetParameter("@UOM_O",
SqlDbType.TinyInt, "UOM_O"))
        daLines.UpdateCommand = comUpdate
    End If

    Dim comUpdateHeader As New SqlCommand("Update
Export_header SET Destination = @Destination, Comments = @Comments,
Itinerary = @Itinerary, Update_date = @Update_date WHERE Order_no =
@Order_no", New SqlConnection(IT.A.ConnectionString))

comUpdateHeader.Parameters.Add(IT.Functions.GetParameter("@Order_no",
SqlDbType.NVarChar, "Order_no"))

comUpdateHeader.Parameters.Add(IT.Functions.GetParameter("@Destination
", SqlDbType.NVarChar, "Destination"))

comUpdateHeader.Parameters.Add(IT.Functions.GetParameter("@Comments",
SqlDbType.NVarChar, "Comments", True))

comUpdateHeader.Parameters.Add(IT.Functions.GetParameter("@Itinerary",
SqlDbType.NVarChar, "Itinerary"))

comUpdateHeader.Parameters.Add(IT.Functions.GetParameter("@Update_date
", SqlDbType.DateTime, "Update_date"))
        daHeaders.UpdateCommand = comUpdateHeader

        daLines.Update(dsOrderEdit, "Export_lines")
        daHeaders.Update(dsOrderEdit, "Export_headers")

        dsOrderEdit.AcceptChanges()

    End If

End Sub
```

```

Private Function CanDeleteProduct(ByVal row As
System.Windows.Forms.DataGridViewRow) As Boolean

    If row.IsNewRow Then Return False

    Dim blnAdditionalProduct As Boolean =
row.Cells("Lines_Grid_Additional_Product").Value
    Dim strOrderNo As String =
row.Cells("Lines_Grid_Order_no").Value
    Dim strProductCode As String =
row.Cells("Lines_Grid_Product_code").Value
    Dim strProductDescr As String =
row.Cells("Lines_Grid_Product_descr").Value

    If Not blnAdditionalProduct Then
        IT.Functions.RaiseError("Το προϊόν δεν μπορεί να διαγραφεί
από την παραγγελία.", Me)
        Return False
    End If

    Dim prod As IT.DB.OrderLine =
IT.DB.OrderHeaders.GetOrder(strOrderNo).OrderLines.GetLine(strProductC
ode)

    If Not prod Is Nothing Then
        If prod.OrderDetails.Count > 0 Then
            IT.Functions.RaiseError("Το προϊόν έχει αναλυτικές
εγγραφές και δεν μπορώ να το διαγράψω.", Me)
            Return False
        End If
    End If

    If MessageBox.Show("Πρόκειται να διαγράψω το προϊόν [" &
strProductCode & " - " & strProductDescr & "] από την παραγγελία [" &
strOrderNo & "].", vbCrLf & "Να συνεχίσω;", "Διαγραφή προϊόντος από
παραγγελία.", MessageBoxButtons.YesNo, MessageBoxIcon.Information,
MessageBoxDefaultButton.Button2) = Windows.Forms.DialogResult.No Then
        Return False
    End If

    Return True

End Function

```

```
Private Function ProductExists(ByVal strProductCode As String) As Boolean
    Dim intCounter As Integer = 0
    For Each row As DataGridViewRow In Me.Lines_Grid.Rows
        If row.Cells("Lines_Grid_Product_code").Value = strProductCode Then intCounter += 1
    Next
    Return intCounter > 1
End Function
```

```
Private Sub BookMarkProductLine(ByVal strProductCode As String, Optional ByVal lngDetailID As Long = 0, Optional ByVal intDetailColumn As Integer = -1)
    For Each row As DataGridViewRow In Me.Lines_Grid.Rows
        If row.Cells("Lines_Grid_Product_code").Value = strProductCode Then
            row.Selected = True
            row.Cells("Lines_Grid_Product_code").Selected = True
            Exit For
        End If
    Next
End Sub
```

```
Private Function GetPackagingQTY(ByVal strCustomerCode As String, ByVal strProductCode As String, ByVal intSalesPer As Integer) As Decimal
    Dim decResult As Decimal = 0
    Dim sql As String = "SELECT " & IIf(intSalesPer = 1, "Pieces", "Est_weight") & " FROM Cust_packaging WHERE Customer_code = '" & strCustomerCode & "' AND Product_code = '" & strProductCode & "'"
    Dim com As New SqlCommand(sql, New SqlConnection(IT.A.ConnectionString))
    com.Connection.Open()
    decResult = CDec(com.ExecuteScalar)
    com.Connection.Close()
    Return IIf(decResult = 0, 1, decResult)
End Function
```

```
Private Sub PaintLines()
    Dim blnProductDeleted As Boolean
    For Each row As DataGridViewRow In Me.Lines_Grid.Rows
        If Not IsDBNull(row.Cells("Lines_Grid_Status").Value) Then
            blnProductDeleted = Not CBool(row.Cells("Lines_Grid_Status").Value) And Not row.IsNewRow
        End If
    Next
End Sub
```

```
                If                blnProductDeleted                Then
row.DefaultCellStyle.BackColor = Color.LightPink
                End If
            Next
        End Sub

End Class
```

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΔΙΑ

Ιχνηλασιμότητα / Customers by product

```
Imports System.Data.SqlClient
```

```
Public Class frmTrack_CustomersByProduct
```

```

    Private dt As New DataTable
    Private Const MUST_REPLACE_WITH_WHERE_CLAUSE_OR_EMPTY_STRING As
String = "@MUST_REPLACE_WITH_WHERE_CLAUSE_OR_EMPTY_STRING@"
    Private Const SQL As String = _
        "SELECT Export_header.Order_no AS Παραγγελία, CASE WHEN
Export_header.Status = 0 THEN '(0) ΝΕΑ ΠΑΡΑΓΓΕΛΙΑ' WHEN
Export_header.Status = 1 THEN '(1) ΥΠΟ ΕΚΤΕΛΕΣΗ' WHEN
Export_header.Status = 2 THEN '(2) ΟΛΟΚΛΗΡΩΜΕΝΗ' ELSE '<Unknown>' END
AS [Κατάσταση Παραγγελίας], Export_header.Transfer_date AS [Ημερομηνία
Μεταφοράς], " & _
        "Export_header.Destination AS Προορισμός,
Export_lines.Product_code AS [Κωδικός Προϊόντος], Products.Descr AS
[Περιγραφή Προϊόντος], Export_lines.Order_no_qty AS [Ποσότητα
Παραγγελίας], UOM.Descr AS [ΜΜ Πώλησης], Export_lines_details.Lot_no
AS [Κωδικός Παρτίδας], CONVERT(datetime, Import_header.Exp_date, 103)
AS [Ημερομηνία Λήξης], SUM(Export_lines_details.Items) AS [Εκτελεσμένα
Τεμάχια], " & _
        "SUM(Export_lines_details.Weight) AS [Εκτελεσμένο Βάρος],
Export_lines_details.Box_no AS [Αριθμός Κιβωτίου],
Export_header.Export_date AS [Ημερομηνία Ολοκλήρωσης] " & _
        "FROM Export_lines_details INNER JOIN Import_header ON
Export_lines_details.Lot_no = Import_header.Lot_no RIGHT OUTER JOIN
Export_header INNER JOIN Export_lines ON Export_header.Order_no =
Export_lines.Order_no INNER JOIN Products ON Export_lines.Product_code
= Products.Code INNER JOIN UOM ON Products.UOM_S = UOM.Code ON
Import_header.Product_code = Products.Code AND
Export_lines_details.Export_Lines_ID = Export_lines.ID " & _
        MUST_REPLACE_WITH_WHERE_CLAUSE_OR_EMPTY_STRING & _
        "GROUP BY Export_header.Order_no, Export_header.Transfer_date,
Export_header.Destination, Export_lines.Product_code, Products.Descr,
Export_lines.Order_no_qty, UOM.Descr, Export_lines_details.Lot_no,
Export_lines_details.Box_no, Export_header.Export_date, CASE WHEN
Export_header.Status = 0 THEN '(0) ΝΕΑ ΠΑΡΑΓΓΕΛΙΑ' WHEN
Export_header.Status = 1 THEN '(1) ΥΠΟ ΕΚΤΕΛΕΣΗ' WHEN
Export_header.Status = 2 THEN '(2) ΟΛΟΚΛΗΡΩΜΕΝΗ' ELSE '<Unknown>' END,
CONVERT(datetime, Import_header.Exp_date, 103)"

```

```

Private Sub Form_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    Clear()
    Me.Show()
    Filter(True)
End Sub

Private Sub Clear()

    If Not dt Is Nothing Then dt.Rows.Clear()

    Me.grpFilter_txtCustomerCode.Text = ""

    Me.grpFilter_grpTransferDates_From.MaxDate = Date.Today
    Me.grpFilter_grpTransferDates_From.Value =
Date.Today.Subtract(New TimeSpan(30, 0, 0, 0))
    Me.grpFilter_grpTransferDates_To.MaxDate = Date.Today
    Me.grpFilter_grpTransferDates_To.Value = Date.Today

    Me.grpFilter_cmbStatus.Items.Clear()
    Me.grpFilter_cmbStatus.Items.AddRange(New String() {"Νέες
Παραγγελίες", "Υπό Εκτέλεση", "Ολοκληρωμένες"})
    Me.grpFilter_cmbStatus.SelectedIndex = 2

    Me.grpFilter_chkStatus.Checked = False
    Me.grpFilter_grpTransferDates_chkFrom.Checked = False
    Me.grpFilter_grpTransferDates_chkTo.Checked = False

    Me.grpFilter_cmbStatus.Enabled =
Me.grpFilter_chkStatus.Checked
    Me.grpFilter_grpTransferDates_From.Enabled =
Me.grpFilter_grpTransferDates_chkFrom.Checked
    Me.grpFilter_grpTransferDates_To.Enabled =
Me.grpFilter_grpTransferDates_chkTo.Checked

End Sub

Private Sub Filter(Optional ByVal TopZero As Boolean = False)

    IT.Functions.WaitMSG()

    dt = New DataTable
    Dim SQL_Filtered As String = ""
    Dim da As New SqlDataAdapter

```

```

Dim com As New SqlCommand
com.Connection = New SqlConnection(IT.A.ConnectionString)

If TopZero Then
    SQL_Filtered =
SQL.Replace(MUST_REPLACE_WITH_WHERE_CLAUSE_OR_EMPTY_STRING,
String.Empty)
    SQL_Filtered = SQL_Filtered.Substring(0, "SELECT ".Length)
    & "TOP 0 " & SQL_Filtered.Substring("SELECT ".Length)
Else
    Dim SQL_WHERE As String = "WHERE "
    SQL_WHERE += "Export_header.Customer_code LIKE '" &
Me.grpFilter_txtCustomerCode.Text.Replace("*", "%") & "'"
    If Me.grpFilter_cmbStatus.Enabled Then SQL_WHERE += " AND
Export_header.Status = " &
Me.grpFilter_cmbStatus.SelectedIndex.ToString
    If Me.grpFilter_grpTransferDates_From.Enabled Then
SQL_WHERE += " AND CONVERT(datetime, Export_header.Transfer_date, 103)
>= CONVERT(datetime, '" & Me.grpFilter_grpTransferDates_From.Value &
"', 103)"
    If Me.grpFilter_grpTransferDates_To.Enabled Then SQL_WHERE
+= " AND CONVERT(datetime, Export_header.Transfer_date, 103) <
CONVERT(datetime, '" &
Me.grpFilter_grpTransferDates_To.Value.AddDays(1) & "', 103)"
    SQL_WHERE += " "
    SQL_Filtered =
SQL.Replace(MUST_REPLACE_WITH_WHERE_CLAUSE_OR_EMPTY_STRING, SQL_WHERE)
    If Me.grpFilter_txtCustomerCode.Text.Replace("*",
"%").Contains("%") Then
        SQL_Filtered = SQL_Filtered.Substring(0, "SELECT
".Length) & "Export_header.Customer_code AS [Κωδικός Πελάτη], " &
SQL_Filtered.Substring("SELECT ".Length)
        Dim intGroupByPos As Integer =
SQL_Filtered.IndexOf("GROUP BY ")
        SQL_Filtered = SQL_Filtered.Substring(0, intGroupByPos
+ "GROUP BY ".Length) & "Export_header.Customer_code, " &
SQL_Filtered.Substring(intGroupByPos + "GROUP BY ".Length)
    End If
End If

com.CommandText = SQL_Filtered
da.SelectCommand = com
da.Fill(dt)

```

```
Me.Panel_Results_Grid.Columns.Clear()
Me.Panel_Results_Grid.DataSource = Me.BindingSource1
Me.BindingSource1.DataSource = dt

IT.Functions.WaitMSG(True)

If Me.Panel_Results_Grid.Rows.Count > 0 Then
    Me.Panel_Results_Grid.Focus()
Else
    If Not TopZero Then IT.Functions.RaiseError("Δεν βρέθηκαν
εγγραφές για τα κριτήρια που δώσατε", Me)
    Me.grpFilter_txtCustomerCode.Focus()
End If

End Sub

Private Sub groupFilter_cmdClear_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
grpFilter_cmdClear.Click
    Clear()
    Me.grpFilter_txtCustomerCode.Focus()
End Sub

Private Sub groupFilter_cmdFilter_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
grpFilter_cmdFilter.Click
    If Me.grpFilter_txtCustomerCode.Text.Length = 0 Then
        IT.Functions.RaiseError("Πρέπει να συμπληρώσετε το πεδίο
[Κωδικός Πελάτη]", Me)
        Me.grpFilter_txtCustomerCode.Focus()
    Else
        Filter()
    End If
End Sub

Private Sub groupFilter_lblCustomerCode_LinkClicked(ByVal sender
As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
grpFilter_lblCustomerCode.LinkClicked
    Dim SQL As String = "SELECT Code AS [Κωδικός Πελάτη], Name AS
[Επωνυμία Πελάτη] FROM Customers"
    Dim strResult As String = IT.Functions.SearchTable_Light(SQL,
"Αναζήτηση Πελάτη")
    If strResult.Length > 0 Then
```

```

        Me.grpFilter_txtCustomerCode.Text = strResult
        Me.grpFilter_txtCustomerCode.SelectAll()
    End If
    Me.grpFilter_txtCustomerCode.Focus()
End Sub

Private Sub groupFilter_txtCustomerCode_TextChanged(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
grpFilter_txtCustomerCode.TextChanged
    Dim strCustomerName As String =
IT.DB.Customers.GetCustomer(Me.grpFilter_txtCustomerCode.Text).Name
    If strCustomerName Is Nothing Then
        Me.grpFilter_txtCustomerName.Text = ""
    Else
        Me.grpFilter_txtCustomerName.Text = strCustomerName
    End If
End Sub

Private Sub groupFilter_TextBox_GotFocus(ByVal sender As Object,
ByVal e As System.EventArgs) _
    Handles grpFilter_txtCustomerCode.GotFocus

    CType(sender, TextBox).SelectAll()

End Sub

Private Sub groupFilter_Fields_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) _
    Handles grpFilter_grpTransferDates_chkFrom.CheckedChanged, _
        grpFilter_grpTransferDates_chkTo.CheckedChanged, _
        grpFilter_chkStatus.CheckedChanged

    Dim check As CheckBox = CType(sender, CheckBox)

    Select Case check.Name
        Case Me.grpFilter_chkStatus.Name
            Me.grpFilter_cmbStatus.Enabled = check.Checked
        Case Me.grpFilter_grpTransferDates_chkFrom.Name
            Me.grpFilter_grpTransferDates_From.Enabled =
check.Checked
        Case Me.grpFilter_grpTransferDates_chkTo.Name
            Me.grpFilter_grpTransferDates_To.Enabled =
check.Checked
    End Select

```

End Sub

End Class

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΔΑΙΑ

Ιχνηλασιμότητα / Ανάκληση

```
Imports System.Data.SqlClient

Public Class frmRecallLots

    Private dsRecallLots As DataSet
    Private dtRecall As DataTable
    Private dtLot As DataTable
    Private dtOrders As DataTable
    Private daRecall As New SqlDataAdapter
    Private daLot As New SqlDataAdapter
    Private daOrders As New SqlDataAdapter

    Private SQL_Recall As String = _
        "SELECT  Recalled_lots.Product_code + ' ( Παρτίδα: ' +
Recalled_lots.Lot_no + ' ) ' + Products.Descr AS Προϊόν,
Recalled_lots.LoginName,  Recalled_lots.Recall_reason AS Σχόλια,
Recalled_lots.Recall_date AS [Ημερομηνία Ανάκλησης],
Recalled_lots.Update_date AS [Ημερομηνία Ενημέρωσης],
Sec_Users.FirstName + CASE WHEN Sec_Users.FirstName <> '' AND
Sec_Users.LastName <> '' THEN ' ' END + Sec_Users.LastName AS Χρήστης
" & _
        "FROM  Recalled_lots INNER JOIN Products ON
Recalled_lots.Product_code = Products.Code INNER JOIN Sec_Users ON
Recalled_lots.LoginName = Sec_Users.LoginName " & _
        "WHERE (Recalled_lots.Product_code = @Product_code) AND
(Recalled_lots.Lot_no = @Lot_no)"

    Private SQL_Lots As String = "SELECT TOP 1 Recall_status AS
Ανάκληση, Lot_no_qty AS [Συνολικό Βάρος], Lot_no_qty_left AS [Υπόλοιπο
Βάρους], Lot_no_pieces AS [Συνολικά Τεμάχια], Lot_no_pieces_left AS
[Υπόλοιπο Τεμαχίων], Exp_date AS [Ημερομηνία Λήξης], ID AS
Import_header_ID FROM Import_header WHERE (Product_code =
@Product_code) AND (Lot_no = @Lot_no)"

    Private SQL_Orders As String = _
        "SELECT  Export_lines_details.Recall_status AS Ανάκληση,
Export_header.Order_no AS Παραγγελία, Export_header.Customer_code AS
[Κωδικός Πελάτη], Customers.Name AS [Επωνυμία Πελάτη],
Export_lines.Order_no_qty AS [Ποσότητα Παραγγελίας], UOM.Descr AS [ΜΜ
Πώλησης],
SUM(Export_lines_details.Items *
Export_lines_details.Crates) AS [Εκτελεσμένα Τεμάχια], " & _
```

```

        "SUM(CASE WHEN Export_lines_details.Weight > 0 THEN
Export_lines_details.Weight - Export_lines_details.Palette_Weight -
(Export_lines_details.Crates * @Crate_Weight) ELSE 0 END) AS
[Εκτελεσμένο Βάρος], Export_header.Destination AS Προορισμός, CASE
WHEN Export_header.Status = 0 THEN '(0) ΝΕΑ ΠΑΡΑΓΓΕΛΙΑ' WHEN
Export_header.Status = 1 THEN '(1) ΥΠΟ ΕΚΤΕΛΕΣΗ' WHEN
Export_header.Status = 2 THEN '(2) ΟΛΟΚΛΗΡΩΜΕΝΗ' ELSE '<Unknown>' END
AS [Κατάσταση Παραγγελίας] " & _
        "FROM Export_header INNER JOIN Export_lines ON
Export_header.Order_no = Export_lines.Order_no INNER JOIN
Export_lines_details ON Export_lines.ID =
Export_lines_details.Export_Lines_ID INNER JOIN Customers ON
Export_header.Customer_code = Customers.Code INNER JOIN Products ON
Export_lines.Product_code = Products.Code INNER JOIN UOM ON
Products.Sale_per = UOM.Code " & _
        "WHERE (Export_lines_details.Lot_no = @Lot_no) AND
(Export_lines.Product_code = @Product_code) " & _
        "GROUP BY Export_header.Order_no, Export_header.Customer_code,
Export_header.Destination, UOM.Descr, Customers.Name,
Export_lines.Order_no_qty, Export_lines_details.Recall_status,
Export_header.Status " & _
        "HAVING (Export_header.Status = 1) OR (Export_header.Status =
2) "

```

```

Private Sub Form_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    Me.Show()
    Clear()
End Sub

```

```

Private Sub Form_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
    If Not RestoreDirty(True, True) Then e.Cancel = True
End Sub

```

```

Private Sub Form_KeyUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles Me.KeyUp
    Select Case e.KeyCode
        Case Keys.Escape : If Me.ts_txtProductCode.Selected Then
Me.ts_txtProductCode.SelectAll()
        Case Keys.F1 And Not e.Shift :
Me.ts_cmdSave.PerformClick()

```



```

        Case Keys.F2 And Not e.Shift :
Me.ts_cmdCancel.PerformClick()
        Case Keys.F3 And Not e.Shift :
Me.ts_cmdProductCode.PerformClick()
        Case Keys.F10 And Not e.Shift :
Me.ts_cmdFilter.PerformClick() : e.SuppressKeyPress = True
        Case Keys.F10 And e.Shift :
Me.ts_cmdRefresh.PerformClick() : e.SuppressKeyPress = True
    End Select
End Sub

```

```

Private Sub ts_cmdFilter_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ts_cmdFilter.Click
    If Me.ts_cmdFilter.Text = "(F10) Ανεύρεση" Then
        If Filter() Then
            Me.ts_txtProductCode.ReadOnly = True
            Me.ts_txtLot_no.ReadOnly = True
            Me.ts_cmdFilter.Text = "(F10) Καθάρισμα"
        End If
    Else
        If RestoreDirty(True, True) Then
            Clear()
            Me.ts_txtProductCode.ReadOnly = False
            Me.ts_txtLot_no.ReadOnly = False
            Me.ts_cmdFilter.Text = "(F10) Ανεύρεση"
        End If
    End If
End Sub

```

```

Private Sub ts_cmdProductCode_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ts_cmdProductCode.Click
    Dim SQL As String = "SELECT Code AS [Κωδικός Προϊόντος], Descr
AS [Περιγραφή Προϊόντος] FROM Products"
    Dim strResult As String = IT.Functions.SearchTable_Light(SQL,
"Αναζήτηση Προϊόντος")
    If strResult.Length > 0 Then
        Me.ts_txtProductCode.Text = strResult
        Me.ts_txtProductCode.SelectAll()
    End If
    Me.ts_txtProductCode.Focus()
End Sub

```

```
Private Sub ts_cmdSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ts_cmdSave.Click
    RestoreDirty(True)
End Sub

Private Sub ts_cmdCancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ts_cmdCancel.Click
    RestoreDirty(False)
End Sub

Private Sub ts_cmdRefresh_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ts_cmdRefresh.Click
    If RestoreDirty(True, True) Then Filter()
End Sub

Private Function Filter() As Boolean

    Dim strProductCode As String = Me.ts_txtProductCode.Text
    Dim strLotNo As String = Me.ts_txtLot_no.Text

    If strProductCode.Length = 0 Then
        IT.Functions.RaiseError("Πρέπει να συμπληρώσετε τον κωδικό προϊόντος", Me)
        Me.ts_txtProductCode.Focus()
        Return False
    End If
    If Not IT.DB.Products.Exists(strProductCode) Then
        IT.Functions.RaiseError("Το προϊόν [" & strProductCode & "] δεν υπάρχει.", Me)
        Me.ts_txtProductCode.SelectAll()
        Me.ts_txtProductCode.Focus()
        Return False
    End If
    If strLotNo.Length = 0 Then
        IT.Functions.RaiseError("Πρέπει να συμπληρώσετε τον κωδικό παρτίδας", Me)
        Me.ts_txtLot_no.Focus()
        Return False
    End If
    If Not LotExists(strLotNo) Then
        IT.Functions.RaiseError("Η παρτίδα [" & strLotNo & "] δεν υπάρχει.", Me)
    End If
End Function
```

```

        Me.ts_txtLot_no.SelectAll()
        Me.ts_txtLot_no.Focus()
        Return False
    End If

    Me.lblRecalledLotFoundStatus.Text = ""
    Me.lblRecalledOrdersFoundStatus.Text = ""

    dsRecallLots = New DataSet("dsRecallLots")
    dtRecall = Nothing
    dtLot = Nothing
    dtOrders = Nothing
    daRecall = Nothing
    daLot = Nothing
    daOrders = Nothing

    LoadTable(strProductCode, strLotNo, "Recall", SQL_Recall, "",
    "", "", dtRecall, daRecall)
    LoadTable(strProductCode, strLotNo, "Lots", SQL_Lots, "", "",
    "", dtLot, daLot)
    LoadTable(strProductCode, strLotNo, "Orders", SQL_Orders, "",
    "", "", dtOrders, daOrders)

    If Me.gridLot.Rows.Count = 0 Then
        Me.lblRecalledLotFoundStatus.Text = "Δεν βρέθηκε απόθεμα."
        If Me.gridOrders.Rows.Count = 0 Then
            Me.lblRecalledOrdersFoundStatus.Text = "Δεν βρέθηκαν παραγγελίες."
            Me.lblRecalledLotFoundStatus.Visible =
            Me.lblRecalledLotFoundStatus.Text.Length > 0
            Me.lblRecalledOrdersFoundStatus.Visible =
            Me.lblRecalledOrdersFoundStatus.Text.Length > 0

            Me.gridLot.Focus()

            Return True
        End Function

        Private Sub LoadTable(ByVal strProductCode As String, ByVal
        strLotNo As String, ByVal table_name As String, ByVal select_statement
        As String, _
            ByVal child_column_name As String, ByVal parent_table_name As
        String, ByVal parent_column_name As String, _

```

```

ByRef data_table As DataTable, ByRef data_adapter As
SqlDataAdapter)

```

```

Dim con As New SqlConnection(IT.A.ConnectionString)
Dim com As New SqlCommand(select_statement, con)

com.Parameters.Add("@Product_code", SqlDbType.NVarChar)
com.Parameters.Add("@Lot_no", SqlDbType.NVarChar)
com.Parameters("@Product_code").Value = strProductCode
com.Parameters("@Lot_no").Value = strLotNo
If table_name = "Orders" Then
    com.Parameters.Add("@Crate_Weight", SqlDbType.Decimal)
    com.Parameters("@Crate_Weight").Value = GetCrateWeight()
End If

```

```

data_adapter = New SqlDataAdapter(com)

```

```

data_adapter.TableMappings.Add("Table", table_name)
Dim comBuilder = New SqlCommandBuilder(data_adapter)
Dim rows As Integer = data_adapter.Fill(dsRecallLots)

```

```

Select Case table_name
    Case "Recall"
        If rows > 0 Then
            Me.BindingSource1.DataSource = dsRecallLots
            Me.BindingSource1.DataMember = "Recall"
            BindFields()
        Else
            BindFields(True)
            SetDefaultValues()
        End If
    Case "Lots"
        Me.gridLot.DataSource = dsRecallLots
        Me.gridLot.DataMember = "Lots"
    Case "Orders"
        Me.gridOrders.DataSource = dsRecallLots
        Me.gridOrders.DataMember = "Orders"
End Select

```

```

data_table = dsRecallLots.Tables(table_name)

```

```

If child_column_name.Length > 0 Then
    Dim parent_table As DataTable =
dsRecallLots.Tables(parent_table_name)

```

```
                Dim                foreign_key                As                New
ForeignKeyConstraint(parent_table.Columns(parent_column_name),
data_table.Columns(child_column_name))
                data_table.Constraints.Add(foreign_key)
            End If

        End Sub

    Private Sub Clear()
        Me.ts_txtProductCode.Text = ""
        Me.ts_txtLot_no.Text = ""
        Me.txtProduct.Text = ""
        Me.txtUserName.Text = ""
        Me.txtComments.Text = ""
        Me.txtRecallDate.Text = ""
        Me.txtUpdateDate.Text = ""
        If Not dtLot Is Nothing Then dtLot.Clear()
        If Not dtOrders Is Nothing Then dtOrders.Clear()
        Me.lblRecalledLotFoundStatus.Visible = False
        Me.lblRecalledOrdersFoundStatus.Visible = False
        Me.ts_txtProductCode.Focus()
    End Sub

    Private Sub SetDefaultValues()
        Dim ServerDate As Date = IT.Functions.GetServerDate
        Dim                strProductDescr                As                String                =
IT.DB.Products.GetProduct(Me.ts_txtProductCode.Text).Description
        If strProductDescr Is Nothing Then strProductDescr = ""
        Me.txtProduct.Text = Me.ts_txtProductCode.Text & " ( Παρτίδα:
" & Me.ts_txtLot_no.Text & " ) " & strProductDescr
        Me.txtComments.Text = ""
        Me.txtUserName.Text = IT.A.User.ToString(True)
        Me.txtRecallDate.Text = ServerDate
        Me.txtUpdateDate.Text = ServerDate
    End Sub

    Private Sub BindFields(Optional ByVal blnClear As Boolean = False)
        If blnClear Then
            Me.txtProduct.DataBindings.Clear()
            Me.txtUserName.DataBindings.Clear()
            Me.txtComments.DataBindings.Clear()
            Me.txtRecallDate.DataBindings.Clear()
            Me.txtUpdateDate.DataBindings.Clear()
        Else
```

```

        If Me.txtProduct.DataBindings.Count = 0 Then
Me.txtProduct.DataBindings.Add(New Binding("Text", Me.BindingSource1,
"Προϊόν", True))
        If Me.txtUserName.DataBindings.Count = 0 Then
Me.txtUserName.DataBindings.Add(New Binding("Text", Me.BindingSource1,
"Χρήστης", True))
        If Me.txtComments.DataBindings.Count = 0 Then
Me.txtComments.DataBindings.Add(New Binding("Text", Me.BindingSource1,
"Σχόλια", True))
        If Me.txtRecallDate.DataBindings.Count = 0 Then
Me.txtRecallDate.DataBindings.Add(New Binding("Text",
Me.BindingSource1, "Ημερομηνία Ανάκλησης", True))
        If Me.txtUpdateDate.DataBindings.Count = 0 Then
Me.txtUpdateDate.DataBindings.Add(New Binding("Text",
Me.BindingSource1, "Ημερομηνία Ενημέρωσης", True))
        End If
    End Sub

```

```

Private Function RestoreDirty(ByVal blnSaveChanges As Boolean,
Optional ByVal AskBeforeSaving As Boolean = False) As Boolean

```

```

    If dsRecallLots Is Nothing Then Return True

    If Me.gridLot.Rows.Count = 0 And Me.gridOrders.Rows.Count = 0
Then Return True

    If blnSaveChanges Then
        Me.BindingSource1.EndEdit()
        Me.gridLot.EndEdit()
        Me.gridOrders.EndEdit()
        If Me.gridLot.Focused Then
            Me.gridOrders.Focus()
            Me.gridLot.Focus()
            If Not Me.gridLot.CurrentRow Is Nothing Then
                Me.gridLot.CurrentCell =
Me.gridLot.CurrentRow.Cells(1)
                Me.gridLot.CurrentCell =
Me.gridLot.CurrentRow.Cells(0)
            End If
        End If
        If Me.gridOrders.Focused Then
            Me.gridLot.Focus()

```

```
        Me.gridOrders.Focus()
        If Not Me.gridOrders.CurrentRow Is Nothing Then
            Me.gridOrders.CurrentRow.Cells(1) =
Me.gridOrders.CurrentRow.Cells(0)
            Me.gridOrders.CurrentRow.Cells(0) =
Me.gridOrders.CurrentRow.Cells(1)
        End If
    End If
Else
    Me.BindingSource1.CancelEdit()
    Me.gridLot.CancelEdit()
    Me.gridOrders.CancelEdit()
End If

If Not dsRecallLots.HasChanges Then Return True

Dim blnAcceptChanges As Boolean = False

If blnSaveChanges Then
    If AskBeforeSaving Then
        Dim result As Microsoft.VisualBasic.MsgBoxResult =
MsgBox(IT.A.Language.Phrase(64, "Υπάρχουν αλλαγές στα δεδομένα. Θέλετε
να τις αποθηκεύσετε;"), MsgBoxStyle.YesNoCancel)
        Select Case result
            Case MsgBoxResult.Yes
                blnAcceptChanges = True
            Case MsgBoxResult.No
                blnAcceptChanges = False
            Case MsgBoxResult.Cancel
                Return False
        End Select
    Else
        blnAcceptChanges = True
    End If
Else
    blnAcceptChanges = False
End If

UpdateChanges(blnAcceptChanges)

Return True

End Function
```

```

Private Sub UpdateChanges(ByVal blnAcceptChanges As Boolean)
    If Not blnAcceptChanges Then
        dsRecallLots.RejectChanges()
    Else
        Dim blnGridLot_RecalledFound As Boolean =
gridLot_RecalledFound()
        Dim blnGridOrders_RecalledFound As Boolean =
gridOrders_RecalledFound()
        Dim strProductCode As String = Me.ts_txtProductCode.Text
        Dim strLotNo As String = Me.ts_txtLot_no.Text
        UpdateRecall(strProductCode, strLotNo, Not
(blnGridLot_RecalledFound Or blnGridOrders_RecalledFound),
dtRecall.Rows.Count = 0)
        UpdateLot()
        UpdateOrders(strProductCode, strLotNo)
        UpdateCustomers(strProductCode, strLotNo)
        dsRecallLots.AcceptChanges()
        Filter()
    End If
End Sub

```

```

Private Sub UpdateRecall(ByVal strProductCode As String, ByVal
strLotNo As String, ByVal blnDelete As Boolean, ByVal blnAdd As
Boolean)

```

```

    Dim com As New SqlCommand

    com.Parameters.Add("@Product_code", SqlDbType.NVarChar)
    com.Parameters.Add("@Lot_no", SqlDbType.NVarChar)
    com.Parameters("@Product_code").Value = strProductCode
    com.Parameters("@Lot_no").Value = strLotNo

    If blnDelete Then
        com.CommandText = "DELETE FROM Recalled_lots WHERE
(Product_code = @Product_code) AND (Lot_no = @Lot_no)"
    Else
        If blnAdd Then
            com.CommandText = "INSERT INTO Recalled_lots
(Product_code, Lot_no, LoginName, Recall_reason, Recall_date,
Update_date)
VALUES
(@Product_code,@Lot_no,@LoginName,@Recall_reason,@Recall_date,@Update_
date)"
        Else

```



```

        com.CommandText = "UPDATE Recalled_lots SET
Recall_reason = @Recall_reason, Recall_date = @Recall_date,
Update_date = @Update_date WHERE (Product_code = @Product_code) AND
(Lot_no = @Lot_no)"
    End If
    com.Parameters.Add("@Recall_reason", SqlDbType.NVarChar)
    com.Parameters.Add("@Recall_date", SqlDbType.DateTime)
    com.Parameters.Add("@Update_date", SqlDbType.DateTime)
    If blnAdd Then com.Parameters.Add("@LoginName",
SqlDbType.NVarChar)

    Dim dateserver As Date = IT.Functions.GetServerDate
    com.Parameters("@Recall_reason").Value =
Me.txtComments.Text
    com.Parameters("@Recall_date").Value = dateserver
    com.Parameters("@Update_date").Value = dateserver
    If blnAdd Then com.Parameters("@LoginName").Value =
IT.A.User.LoginName
    End If

    com.Connection = New SqlConnection(IT.A.ConnectionString)
    com.Connection.Open()
    com.ExecuteNonQuery()
    com.Connection.Close()
    com = Nothing

End Sub

Private Sub UpdateLot()
    daLot.Update(dtLot)
    dsRecallLots.Tables("Lots").AcceptChanges()
End Sub

Private Sub UpdateOrders(ByVal strProductCode As String, ByVal
strLotNo As String)

    Dim dtChanges_Modified As DataTable =
Me.dtOrders.GetChanges(DataRowState.Modified)
    If Not dtChanges_Modified Is Nothing Then
        Dim comUpdate As New SqlCommand("UPDATE
Export_lines_details SET Recall_status = @Recall_status FROM
Export_header INNER JOIN Export_lines ON Export_header.Order_no =
Export_lines.Order_no INNER JOIN Export_lines_details ON
Export_lines.ID = Export_lines_details.Export_Lines_ID WHERE

```

```

(Export_lines.Product_code = @Product_code) AND
(Export_lines_details.Lot_no = @Lot_no) AND (Export_header.Order_no =
@Order_no)", New SqlConnection(IT.A.ConnectionString))

comUpdate.Parameters.Add(IT.Functions.GetParameter("@Recall_status",
SqlDbType.Bit, "Ανάκληση", True))

comUpdate.Parameters.Add(IT.Functions.GetParameter("@Order_no",
SqlDbType.NVarChar, "Παραγγελία"))
    comUpdate.Parameters.Add("@Product_code",
SqlDbType.NVarChar)
    comUpdate.Parameters.Add("@Lot_no", SqlDbType.NVarChar)
    comUpdate.Parameters("@Product_code").Value =
strProductCode
    comUpdate.Parameters("@Lot_no").Value = strLotNo
daOrders.UpdateCommand = comUpdate
daOrders.Update(dsRecallLots, "Orders")
End If
dsRecallLots.Tables("Orders").AcceptChanges()

End Sub

Private Sub UpdateCustomers(ByVal strProductCode As String, ByVal
strLotNo As String)

    Dim comCustomers As New SqlCommand("SELECT
Export_header.Customer_code FROM Export_header INNER JOIN Export_lines
ON Export_header.Order_no = Export_lines.Order_no INNER JOIN
Export_lines_details ON Export_lines.ID =
Export_lines_details.Export_Lines_ID WHERE (Export_lines.Product_code
= '" & strProductCode & "') AND (Export_lines_details.Lot_no = '" &
strLotNo & "') GROUP BY Export_header.Customer_code", New
SqlConnection(IT.A.ConnectionString))
    comCustomers.Connection.Open()
    Dim reaCustomers As SqlDataReader = comCustomers.ExecuteReader

    If reaCustomers.HasRows Then
        Dim strCustomerCode As String = ""
        Dim rows As Integer = 0
        Dim com As New SqlCommand("SELECT Export_header.Order_no
FROM Export_header INNER JOIN Export_lines ON Export_header.Order_no =
Export_lines.Order_no INNER JOIN Export_lines_details ON
Export_lines.ID = Export_lines_details.Export_Lines_ID WHERE
(Export_lines.Product_code = '" & strProductCode & "') AND

```

```

(Export_lines_details.Lot_no = '' & strLotNo & '') AND
(Export_header.Customer_code = @Customer_code) AND
(Export_lines_details.Recall_status = 1) GROUP BY
Export_header.Order_no", New SqlConnection(IT.A.ConnectionString))
    com.Parameters.Add("@Customer_code", SqlDbType.NVarChar)
    com.Connection.Open()

    Do While reaCustomers.Read
        strCustomerCode = CStr(reaCustomers("Customer_code"))
        com.Parameters("@Customer_code").Value =
strCustomerCode

        Dim table As New DataTable
        Dim adapter As New SqlDataAdapter(com)
        rows = adapter.Fill(table)
        If rows > -1 Then
            Dim comUpdate As New SqlCommand("Update Customers
SET Recall_products = " & rows & " WHERE Code = '" & strCustomerCode &
'", New SqlConnection(IT.A.ConnectionString))
            comUpdate.Connection.Open()
            comUpdate.ExecuteNonQuery()
            comUpdate.Connection.Close()

        End If
    Loop

    com.Connection.Close()

End If

reaCustomers.Close()
comCustomers.Connection.Close()

End Sub

Private Function LotExists(ByVal strLotNo As String) As Boolean

    Dim blnReturn As Boolean
    Dim com As New SqlClient.SqlCommand("SELECT COUNT(ID) FROM
Import_header WHERE Lot_no = '" & strLotNo & "'", New
SqlClient.SqlConnection(IT.A.ConnectionString))

    Try
        com.Connection.Open()

```

```
        blnReturn = CBool(CInt(com.ExecuteScalar) > 0)
    Catch exc As Exception
        MsgBox(exc.Message)
    Finally
        com.Connection.Close()
        com = Nothing
    End Try

    Return blnReturn

End Function

Private Sub gridLot_CellContentClick(ByVal sender As Object, ByVal
e As System.Windows.Forms.DataGridViewCellEventArgs) Handles
gridLot.CellContentClick
    If Me.gridLot.Columns(e.ColumnIndex).Name =
"gridLot_Recall_status" Then
        Dim msgResult As MsgBoxResult = MsgBox("Θέλετε να ορίσω με
την ίδια τιμή ανάκλησης όλες τις παρτίδες των παραγγελιών;",
MsgBoxStyle.Question Or MsgBoxStyle.YesNo Or
MsgBoxStyle.DefaultButton1)
        If msgResult = MsgBoxResult.No Then Exit Sub
        Me.gridLot.EndEdit()
        For Each row As DataGridViewRow In Me.gridOrders.Rows
            Me.dtOrders.Rows(row.Index).Item(0) =
CBool(Me.gridLot.Rows(e.RowIndex).Cells(e.ColumnIndex).FormattedValue)
        Next
    End If
End Sub

Private Function gridLot_RecalledFound() As Boolean
    For Each row As DataGridViewRow In Me.gridLot.Rows
        If Not IsDBNull(row.Cells("gridLot_Recall_status").Value)
Then
            If CBool(row.Cells("gridLot_Recall_status").Value) =
True Then Return True
        End If
    Next
    Return False
End Function

Private Function gridOrders_RecalledFound() As Boolean
    For Each row As DataGridViewRow In Me.gridOrders.Rows
```

```
        If                                     Not
IsDBNull(row.Cells("gridOrders_Recall_status").Value) Then
        If CBool(row.Cells("gridOrders_Recall_status").Value)
= True Then Return True
        End If
    Next
    Return False
End Function

Private Function GetCrateWeight() As Decimal
    Dim decWeight As Decimal = 0
    Dim SQL As String = "SELECT Tare FROM Packaging WHERE Code=10"
    Dim con As New SqlConnection(IT.A.ConnectionString)
    Dim com As New SqlCommand(SQL, con)
    con.Open()
    decWeight = CDec(com.ExecuteScalar)
    con.Close()
    Return decWeight
End Function

End Class
```

Ιχνηλασιμότητα / Products by customer

```
Imports System.Data.SqlClient
```

```
Public Class frmTrack_ProductsByCustomer
```

```

    Private dt As New DataTable
    Private Const MUST_REPLACE_WITH_WHERE_CLAUSE_OR_EMPTY_STRING As
String = "@MUST_REPLACE_WITH_WHERE_CLAUSE_OR_EMPTY_STRING@"
    Private Const SQL As String = _
        "SELECT Export_header.Order_no AS Παραγγελία, CASE WHEN
Export_header.Status = 0 THEN '(0) ΝΕΑ ΠΑΡΑΓΓΕΛΙΑ' WHEN
Export_header.Status = 1 THEN '(1) ΥΠΟ ΕΚΤΕΛΕΣΗ' WHEN
Export_header.Status = 2 THEN '(2) ΟΛΟΚΛΗΡΩΜΕΝΗ' ELSE '<Unknown>' END
AS [Κατάσταση Παραγγελίας], Export_header.Transfer_date AS [Ημερομηνία
Μεταφοράς], " & _
        "Export_header.Customer_code AS [Κωδικός Πελάτη],
Customers.Name AS [Επωνυμία Πελάτη], Export_header.Destination AS
Προορισμός, Export_lines.Order_no_qty AS [Ποσότητα Παραγγελίας],
UOM.Descr AS [ΜΜ Πώλησης], Export_lines_details.Lot_no AS [Κωδικός
Παρτίδας], CONVERT(datetime, Import_header.Exp_date, 103) AS
[Ημερομηνία Λήξης], SUM(Export_lines_details.Items) AS [Εκτελεσμένα
Τεμάχια], " & _
        "SUM(Export_lines_details.Weight) AS [Εκτελεσμένο Βάρος],
Export_lines_details.Box_no AS [Αριθμός Κιβωτίου],
Export_header.Export_date AS [Ημερομηνία Ολοκλήρωσης] " & _
        "FROM Import_header INNER JOIN Export_lines_details ON
Import_header.Lot_no = Export_lines_details.Lot_no RIGHT OUTER JOIN
Export_header INNER JOIN Export_lines ON Export_header.Order_no =
Export_lines.Order_no INNER JOIN Products ON Export_lines.Product_code
= Products.Code INNER JOIN UOM ON Products.UOM_S = UOM.Code INNER JOIN
Customers ON Export_header.Customer_code = Customers.Code ON
Import_header.Product_code = Products.Code AND
Export_lines_details.Export_Lines_ID = Export_lines.ID " & _
        MUST_REPLACE_WITH_WHERE_CLAUSE_OR_EMPTY_STRING & _
        "GROUP BY Export_header.Order_no, Export_header.Transfer_date,
Export_header.Customer_code, Customers.Name,
Export_header.Destination, Export_lines.Order_no_qty, UOM.Descr,
Export_lines_details.Lot_no, Export_lines_details.Box_no,
Export_header.Export_date, CASE WHEN Export_header.Status = 0 THEN
'(0) ΝΕΑ ΠΑΡΑΓΓΕΛΙΑ' WHEN Export_header.Status = 1 THEN '(1) ΥΠΟ
ΕΚΤΕΛΕΣΗ' WHEN Export_header.Status = 2 THEN '(2) ΟΛΟΚΛΗΡΩΜΕΝΗ' ELSE
'<Unknown>' END, CONVERT(datetime, Import_header.Exp_date, 103)"

```

```
Private Sub Form_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
    Clear()
    Me.Show()
    Filter(True)
End Sub

Private Sub Clear()

    If Not dt Is Nothing Then dt.Rows.Clear()

    Me.grpFilter_txtProductCode.Text = ""
    Me.grpFilter_txtLotNo.Text = ""

    Me.grpFilter_grpTransferDates_From.MaxDate = Date.Today
    Me.grpFilter_grpTransferDates_From.Value =
Date.Today.Subtract(New TimeSpan(30, 0, 0, 0))
    Me.grpFilter_grpTransferDates_To.MaxDate = Date.Today
    Me.grpFilter_grpTransferDates_To.Value = Date.Today

    Me.grpFilter_grpExpDates_From.Value = Date.Today.Subtract(New
TimeSpan(30, 0, 0, 0))
    Me.grpFilter_grpExpDates_To.Value = Date.Today

    Me.grpFilter_cmbStatus.Items.Clear()
    Me.grpFilter_cmbStatus.Items.AddRange(New String() {"Νέες
Παραγγελίες", "Υπό Εκτέλεση", "Ολοκληρωμένες"})
    Me.grpFilter_cmbStatus.SelectedIndex = 2

    Me.grpFilter_chkLotNo.Checked = False
    Me.grpFilter_chkStatus.Checked = False
    Me.grpFilter_grpTransferDates_chkFrom.Checked = False
    Me.grpFilter_grpTransferDates_chkTo.Checked = False
    Me.grpFilter_grpExpDates_chkFrom.Checked = False
    Me.grpFilter_grpExpDates_chkTo.Checked = False

    Me.grpFilter_txtLotNo.Enabled = Me.grpFilter_chkLotNo.Checked
    Me.grpFilter_cmbStatus.Enabled =
Me.grpFilter_chkStatus.Checked
    Me.grpFilter_grpTransferDates_From.Enabled =
Me.grpFilter_grpTransferDates_chkFrom.Checked
    Me.grpFilter_grpTransferDates_To.Enabled =
Me.grpFilter_grpTransferDates_chkTo.Checked
```

```

        Me.grpFilter_grpExpDates_From.Enabled =
Me.grpFilter_grpExpDates_chkFrom.Checked
        Me.grpFilter_grpExpDates_To.Enabled =
Me.grpFilter_grpExpDates_chkTo.Checked

End Sub

Private Sub Filter(Optional ByVal TopZero As Boolean = False)

    IT.Functions.WaitMSG()

    dt = New DataTable
    Dim SQL_Filtered As String = ""
    Dim da As New SqlDataAdapter
    Dim com As New SqlCommand
    com.Connection = New SqlConnection(IT.A.ConnectionString)

    If TopZero Then
        SQL_Filtered =
SQL.Replace(MUST_REPLACE_WITH_WHERE_CLAUSE_OR_EMPTY_STRING,
String.Empty)
        SQL_Filtered = SQL_Filtered.Substring(0, "SELECT ".Length)
& "TOP 0 " & SQL_Filtered.Substring("SELECT ".Length)
    Else
        Dim SQL_WHERE As String = "WHERE "
        SQL_WHERE += "Export_lines.Product_code LIKE '" &
Me.grpFilter_txtProductCode.Text.Replace("*", "%") & "'"
        If Me.grpFilter_cmbStatus.Enabled Then SQL_WHERE += " AND
Export_header.Status = " &
Me.grpFilter_cmbStatus.SelectedIndex.ToString
        If Me.grpFilter_txtLotNo.Enabled Then SQL_WHERE += " AND
Export_lines_details.Lot_no LIKE '" &
Me.grpFilter_txtLotNo.Text.Replace("*", "%") & "'"
        If Me.grpFilter_grpTransferDates_From.Enabled Then
SQL_WHERE += " AND CONVERT(datetime, Export_header.Transfer_date, 103)
>= CONVERT(datetime, '" & Me.grpFilter_grpTransferDates_From.Value &
"', 103)"
        If Me.grpFilter_grpTransferDates_To.Enabled Then SQL_WHERE
+= " AND CONVERT(datetime, Export_header.Transfer_date, 103) <
CONVERT(datetime, '" &
Me.grpFilter_grpTransferDates_To.Value.AddDays(1) & "', 103)"
        If Me.grpFilter_grpExpDates_From.Enabled Then SQL_WHERE +=
" AND CONVERT(datetime, Import_header.Exp_date, 103) >=
CONVERT(datetime, '" & Me.grpFilter_grpExpDates_From.Value & "', 103)"
    
```



```

        If Me.grpFilter_grpExpDates_To.Enabled Then SQL_WHERE += "
AND CONVERT(datetime, Import_header.Exp_date, 103) < CONVERT(datetime,
'" & Me.grpFilter_grpExpDates_To.Value.AddDays(1) & "', 103)"
        SQL_WHERE += " "
        SQL_Filtered =
SQL.Replace(MUST_REPLACE_WITH_WHERE_CLAUSE_OR_EMPTY_STRING, SQL_WHERE)
        If Me.grpFilter_txtProductCode.Text.Replace(" ",
"%").Contains("%") Then
            SQL_Filtered = SQL_Filtered.Substring(0, "SELECT
".Length) & "Export_lines.Product_code AS [Κωδικός Προϊόντος], " &
SQL_Filtered.Substring("SELECT ".Length)
            Dim intGroupByPos As Integer =
SQL_Filtered.IndexOf("GROUP BY ")
            SQL_Filtered = SQL_Filtered.Substring(0, intGroupByPos
+ "GROUP BY ".Length) & "Export_lines.Product_code, " &
SQL_Filtered.Substring(intGroupByPos + "GROUP BY ".Length)
        End If
    End If

    com.CommandText = SQL_Filtered
    da.SelectCommand = com
    da.Fill(dt)

    Me.Panel_Results_Grid.Columns.Clear()
    Me.Panel_Results_Grid.DataSource = Me.BindingSource1
    Me.BindingSource1.DataSource = dt

    IT.Functions.WaitMSG(True)

    If Me.Panel_Results_Grid.Rows.Count > 0 Then
        Me.Panel_Results_Grid.Focus()
    Else
        If Not TopZero Then IT.Functions.RaiseError("Δεν βρέθηκαν
εγγραφές για τα κριτήρια που δώσατε", Me)
        Me.grpFilter_txtProductCode.Focus()
    End If

End Sub

Private Sub groupFilter_cmdClear_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
grpFilter_cmdClear.Click
    Clear()
    Me.grpFilter_txtProductCode.Focus()

```

```
End Sub

Private Sub groupFilter_cmdFilter_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles grpFilter_cmdFilter.Click
    If Me.grpFilter_txtProductCode.Text.Length = 0 Then
        IT.Functions.RaiseError("Πρέπει να συμπληρώσετε το πεδίο [Κωδικός Προϊόντος]", Me)
        Me.grpFilter_txtProductCode.Focus()
    Else
        Filter()
    End If
End Sub

Private Sub groupFilter_lblProductCode_LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles grpFilter_lblProductCode.LinkClicked
    Dim SQL As String = "SELECT Code AS [Κωδικός Προϊόντος], Descr AS [Περιγραφή Προϊόντος] FROM Products"
    Dim strResult As String = IT.Functions.SearchTable_Light(SQL, "Αναζήτηση Προϊόντος")
    If strResult.Length > 0 Then
        Me.grpFilter_txtProductCode.Text = strResult
        Me.grpFilter_txtProductCode.SelectAll()
    End If
    Me.grpFilter_txtProductCode.Focus()
End Sub

Private Sub groupFilter_txtProductCode_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles grpFilter_txtProductCode.TextChanged
    Dim strProductDescr As String = IT.DB.Products.GetProduct(Me.grpFilter_txtProductCode.Text).Description
    If strProductDescr Is Nothing Then
        Me.grpFilter_txtProductDescr.Text = ""
    Else
        Me.grpFilter_txtProductDescr.Text = strProductDescr
    End If
End Sub

Private Sub groupFilter_TextBox_GotFocus(ByVal sender As Object, ByVal e As System.EventArgs) _
```

```
Handles grpFilter_txtProductCode.GotFocus,
grpFilter_txtLotNo.GotFocus

CType(sender, TextBox).SelectAll()

End Sub

Private Sub groupFilter_Fields_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) _
Handles grpFilter_grpTransferDates_chkFrom.CheckedChanged, _
grpFilter_grpTransferDates_chkTo.CheckedChanged, _
grpFilter_grpExpDates_chkFrom.CheckedChanged, _
grpFilter_grpExpDates_chkTo.CheckedChanged, _
grpFilter_chkLotNo.CheckedChanged, _
grpFilter_chkStatus.CheckedChanged

Dim check As CheckBox = CType(sender, CheckBox)

Select Case check.Name
Case Me.grpFilter_chkLotNo.Name
Me.grpFilter_txtLotNo.Enabled = check.Checked
Case Me.grpFilter_chkStatus.Name
Me.grpFilter_cmbStatus.Enabled = check.Checked
Case Me.grpFilter_grpTransferDates_chkFrom.Name
Me.grpFilter_grpTransferDates_From.Enabled =
check.Checked
Case Me.grpFilter_grpTransferDates_chkTo.Name
Me.grpFilter_grpTransferDates_To.Enabled =
check.Checked

Case Me.grpFilter_grpExpDates_chkFrom.Name
Me.grpFilter_grpExpDates_From.Enabled = check.Checked
Case Me.grpFilter_grpExpDates_chkTo.Name
Me.grpFilter_grpExpDates_To.Enabled = check.Checked
End Select

End Sub

End Class
```

Φόρμα Πληροφοριών Συστήματος Ικνηλασιμότητας

```
Public NotInheritable Class frmAbout

    Private Sub frmAbout_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
        ' Set the title of the form.
        Dim ApplicationTitle As String
        If My.Application.Info.Title <> "" Then
            ApplicationTitle = My.Application.Info.Title
        Else
            ApplicationTitle =
System.IO.Path.GetFileNameWithoutExtension(My.Application.Info.Assembl
yName)
        End If
        Me.Text = String.Format("About {0}", ApplicationTitle)
        ' Initialize all of the text displayed on the About Box.
        ' TODO: Customize the application's assembly information in
the "Application" pane of the project
        ' properties dialog (under the "Project" menu).
        Me.LabelProductName.Text = My.Application.Info.ProductName
        Me.LabelVersion.Text = String.Format("Version {0}",
My.Application.Info.Version.ToString)
        Me.LabelCopyright.Text = My.Application.Info.Copyright
        Me.LabelCompanyName.Text = My.Application.Info.CompanyName
        Me.TextBoxDescription.Text = My.Application.Info.Description
    End Sub

    Private Sub OKButton_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles OKButton.Click
        Me.Close()
    End Sub

End Class
```

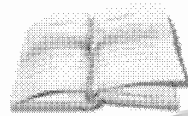
2.4.6. Λεπτομερές σχέδιο δεδομένων

Παραπομπή στο διάγραμμα οντοτήτων συσχετίσεων.

Βιβλιογραφία

Βιβλία

- 📖 Ian Smith, Anthony Furness, 2006, “Improving traceability in food processing and distribution”, CRC publisher.
- 📖 L. A. Rabade, J. A. Alfaro, 2005, “Journal of Purchasing and Supply Management”, Elsevier publisher.



- 📖 Michele Lees, 2003, “Food Authenticity and Traceability”, 1 edition CRC publisher.
- 📖 Mike Dillon, 2007, “Traceability in Food”, Blackwell Publishers
- 📖 Pfleeger, Shari Lawrence, 2003, «Τεχνολογία λογισμικού Α΄ ΤΟΜΟΣ», Εκδόσεις Κλειδάριθμος
- 📖 Pfleeger, Shari Lawrence, 2004, «Τεχνολογία λογισμικού Β΄ ΤΟΜΟΣ», Εκδόσεις Κλειδάριθμος
- 📖 Γιακουμάκης, Εμμανουήλ Α., 1993, «Τεχνολογία λογισμικού», Εκδόσεις Σταμούλη Α.Ε.

Άρθρα

- 📄 Brian Schiavo, 2007, “Getting smart with traceability: track and trace solution does more than meet bioterrorism regulations”, an article from: Food Logistics, Thomson Gale publisher.
- 📄 Food Authenticity and Traceability, 2005, an article from Food Trade Review, publisher Food Trade Press Ltd.
- 📄 John Nelson, 2005, Product and component traceability, an article from Circuits Assembly, UP Media Group publisher.
- 📄 Robin Barbero, 2006, Tracing a path to success: reducing production costs, increasing product safety, and ensuring customer satisfaction with traceability systems, an article from: Tooling & Production, Thomson Gale publisher.

Links

- 📄 www.gs1.gr
- 📄 www.plant-management.gr
- 📄 www.logistics-management.gr
- 📄 www.foodanddrinks.gr
- 📄 www.rfidtechnology.gr
- 📄 Του ελληνικού e-Business Forum: <http://www.ebusinessforum.gr>
- 📄 www.civ-viande.org/gr



- 📄 Της Ελληνικής Εταιρίας Logistics (EEL): <http://www.eel.gr>
- 📄 Της Διεθνούς Κοινότητας Logistics (SOLE): <http://www.sole.org>
- 📄 Του Ινστιτούτου Logistics πόλεων (Institute of City Logistics): <http://icl.kiban.kuciv.kyoto-u.ac.jp/>
- 📄 Του ελληνικού περιοδικού A & M : <http://www.forumltd.gr>
- 📄 Του ελληνικού περιοδικού L&M: <http://www.logistics-management.gr/index.htm>

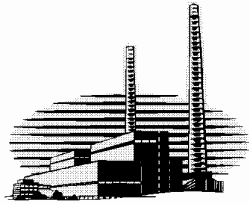
Περιοδικά

- 📄 “Plant Management”, Ετήσια έκδοση 2002 - 2003
- 📄 “Plant Management”, Ετήσια έκδοση 2005 - 2006
- 📄 “Plant Management & Logistics”, Ετήσια έκδοση 2006
- 📄 “New Economy Observer (ne.o)”, Έκδοση Δεκεμβρίου 2003
- 📄 “Αποθήκη Logistics Μεταφορές (A&M)”, Έκδοση Νοεμβρίου - Δεκεμβρίου 2003
- 📄 “Logistics & Management”, Έκδοση Ιανουαρίου - Φεβρουαρίου 2004
- 📄 “Plant Management”, Έκδοση Οκτωβρίου - Νοεμβρίου 2002

- 📖 “Plant Management”, Έκδοση Δεκεμβρίου 2001 - Ιανουαρίου 2007
- 📖 “European Purchasing & Materials Management” Ετήσια έκδοση 2006
- 📖 Retail Technology
- 📖 Plant Logistics

Εταιρίες παροχής λογισμικών ικνηλασιμότητας

- 🌐 www.g-logic.gr
- 🌐 www.agrosoft.gr



- 🌐 www.sinodos.com
- 🌐 www.supply-chain.gr
- 🌐 www.optisoft.gr
- 🌐 www.theodorou.gr

Παράρτημα