



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP
Όνοματεπώνυμο Φοιτητή	Σιάτης Ευάγγελος
Πατρώνυμο	Στυλιανός
Αριθμός Μητρώου	ΜΠΠΛ-07030
Επιβλέπων	Καθηγητής Δημήτριος Δεσπότης

Ημερομηνία Παράδοσης **Οκτώβριος 2012**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Δ. Δεσπότης
Καθηγητής

Ε. Φούντας
Καθηγητής

Δ. Αποστόλου
Επικ. Καθηγητής

Περίληψη

Στα πλαίσια της παρούσας διατριβής αναπτύχθηκε ένα πληροφοριακό σύστημα διαχείρισης εργασιών συντήρησης. Το σύστημα αποτελείται από μία βάση δεδομένων, στην οποία αποθηκεύονται οι επιθυμητές πληροφορίες, και μία εφαρμογή γραφικού περιβάλλοντος η οποία παρέχει πρόσβαση στις πληροφορίες. Για την οργάνωση της κατασκευής του συστήματος χρησιμοποιήθηκε η διαδικασία ανάπτυξης έργων λογισμικού Rational Unified Process και για την υλοποίησή του χρησιμοποιήθηκε η γλώσσα προγραμματισμού C# σε συνδυασμό με την πλατφόρμα ανάπτυξης .NET. Για τη φιλοξενία της βάσης δεδομένων επιλέχθηκε το σύστημα διαχείρισης βάσεων δεδομένων MySQL. Κατά τις φάσεις ανάπτυξης του έργου κατασκευάστηκαν UML διαγράμματα που περιγράφουν τις απαιτήσεις και τον τρόπο υλοποίησης του έργου. Μετά το πέρας της ανάπτυξης πραγματοποιήθηκε έλεγχος επάρκειας της λειτουργικότητας του συστήματος.

Λέξεις κλειδιά: Πληροφοριακά Συστήματα, Σύστημα Διαχείρισης Εργασιών Συντήρησης, Βάσεις Δεδομένων, Rational Unified Process, C# , .NET

Abstract

Under the context of this essay, a Computerized Maintenance Management Systems is designed and developed. The system comprises of a database which holds the respective data and an application which provides access to these data. The system is designed and developed with use of Rational Unified Process. Moreover, C#, .NET and MySQL technologies are used for the system production. Various UML diagrams were produced according to the directives of the phases of Rational Unified Process. After completeness of the project, the system was tested in order to clarify its proper functionality, according to the specifications initially set.

Keywords: Information Systems, Database Management Systems, Databases, Rational Unified Process, C#, .NET

Στους γονείς μου και την οικογένεια μου

Πίνακας Περιεχομένων

Περιεχόμενα	8
Πίνακας Εικόνων	11
1.1 Αντικείμενο της εργασίας	14
1.2 Δομή της εργασίας	14
2 Κεφάλαιο 2: Βιβλιογραφική Ανασκόπηση	16
2.1 Βιβλιογραφική Έρευνα	16
2.1.1 Συστήματα Σχεδιασμού Επιχειρησιακών Πόρων	16
2.1.2 Συστήματα διαχείρισης εργασιών συντήρησης	17
2.2 Τεχνολογίες και Εργαλεία	18
2.2.1 Το σύστημα διαχείρισης βάσεων δεδομένων (ΣΔΒΔ) mysql..	18
2.2.2 Η γλώσσα προγραμματισμού C# και το περιβάλλον .NET	19
2.2.3 Το περιβάλλον ανάπτυξης Microsoft Visual Studio.....	20
2.2.4 Η διαδικασία ανάπτυξης RUP (Rational Unified Process)	20
3 Κεφάλαιο 3: Ανάπτυξη Εφαρμογής.....	22
3.1 Ορισμός απαιτήσεων σε φυσική γλώσσα	22
3.2 Προδιαγραφές.....	22
3.2.1 Εννοιολογικές οντότητες της εφαρμογής.....	22
3.2.2 Χρήστες και επίπεδα πρόσβασης	23
3.2.3 Λειτουργίες της εφαρμογής.....	23
3.3 Φάση σύλληψης (ανάλυση).....	26
3.3.1 Διαγράμματα περιπτώσεων χρήσης (Use Case Diagrams).....	26
3.3.2 Διαγράμματα τάξεων (Class Diagrams)	26
3.4 Φάση κλιμάκωσης (εκπόνηση μελέτης)	28

3.4.1	Διαγράμματα περιπτώσεων χρήσης (Use Case Diagrams).....	28
3.4.2	Διαγράμματα τάξεων (Class Diagrams)	28
3.4.3	Διαγράμματα δραστηριοτήτων (Activity Diagrams).....	29
3.4.4	Διαγράμματα συνεργασίας (Collaboration Diagrams)	44
3.4.5	Διαγράμματα σειράς (Sequence Diagrams)	45
3.4.6	Διαγράμματα διανομής(Deployment Diagrams).....	46
3.5	Φάση κατασκευής	46
3.5.1	Διαγράμματα περιπτώσεων χρήσης (Use Case Diagrams).....	47
3.5.2	Διαγράμματα τάξεων (Class Diagrams)	47
3.5.3	Διαγράμματα δραστηριοτήτων(Activity Diagrams).....	48
3.5.4	Διαγράμματα συνεργασίας(Collaboration Diagrams)	59
3.5.5	Διαγράμματα σειράς (Sequence Diagrams)	59
3.5.6	Διαγράμματα διανομής(Deployment Diagrams).....	60
3.5.7	Διάγραμμα Οντοτήτων Συσχετίσεων της Βάσης Δεδομένων .	60
3.5.8	Διάγραμμα Σχεσιακού Μοντέλου της Βάσης Δεδομένων.....	62
4	Κεφάλαιο 4: Περιγραφή της Εφαρμογής.....	64
4.1	Συνοπτική Περιγραφή της Εφαρμογής	64
4.1.1	Η οθόνη εισόδου	64
4.1.2	Η κεντρική οθόνη εισαγωγής στοιχείων της εφαρμογής	65
4.1.3	Η οθόνη αλλαγής στοιχείων χρήστη.....	66
4.1.4	Η οθόνη προβολής κινήσεων των χρηστών.....	67
4.1.5	Η οθόνη αντιστοίχισης ανταλλακτικού με μηχανήμα	68
4.1.6	Η οθόνη αντιστοίχιση ανταλλακτικού με ΣΠΣ	69
4.1.7	Η οθόνη πληροφοριών	69
4.1.8	Η οθόνη καταχώρησης ολοκλήρωσης ΣΠΣ	71

4.2	Σενάρια χρήσης της εφαρμογής.....	72
4.2.1	Εισαγωγή, αλλαγή στοιχείων και διαγραφή μηχανήματος....	72
4.2.2	Εισαγωγή, αλλαγή στοιχείων και διαγραφή ΣΠΣ.	75
4.2.3	Εισαγωγή, αλλαγή στοιχείων και διαγραφή ανταλλακτικού .	77
4.2.4	Εισαγωγή, αλλαγή στοιχείων και διαγραφή ιστορικού.....	81
4.2.5	Αλλαγή και διαγραφή αποθέματος ανταλλακτικού.....	84
4.2.6	Προβολή πληροφοριών και εξαγωγή δεδομένων σε αρχεία .	85
4.2.7	Καταχώρηση Ολοκλήρωσης ΣΠΣ.	86
4.2.8	Διαχείριση εφαρμογής.	87
5	Κεφάλαιο 5: Συμπεράσματα	89
5.1	Συμπεράσματα	89
5.2	Προτάσεις για περαιτέρω ανάπτυξη	90
6	Βιβλιογραφία.....	91
7	Παράρτημα.....	92

Πίνακας Εικόνων

Εικόνα 3-1: Αρχικό διάγραμμα περιπτώσεων χρήσης της εφαρμογής.....	26
Εικόνα 3-2: Αρχικό διάγραμμα τάξεων της εφαρμογής.	27
Εικόνα 3-3: Διάγραμμα περιπτώσεων χρήσης κατά τη φάση κλιμάκωσης.....	28
Εικόνα 3-4: Διάγραμμα τάξεων κατά τη φάση κλιμάκωσης.	29
Εικόνα 3-5: Διάγραμμα δραστηριοτήτων της λειτουργίας εισαγωγής νέου μηχανήματος.....	30
Εικόνα 3-6: Διάγραμμα δραστηριοτήτων της λειτουργίας αλλαγής στοιχείων μηχανήματος.	31
Εικόνα 3-7: Διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής μηχανήματος.....	32
Εικόνα 3-8: Διάγραμμα δραστηριοτήτων της λειτουργίας εισαγωγής νέου ΣΠΣ.....	33
Εικόνα 3-9: Διάγραμμα δραστηριοτήτων της λειτουργίας αλλαγής στοιχείων ΣΠΣ.	34
Εικόνα 3-10: Διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής ΣΠΣ.....	35
Εικόνα 3-11: Διάγραμμα δραστηριοτήτων της λειτουργίας εισαγωγής νέου ανταλλακτικού.....	36
Εικόνα 3-12: Διάγραμμα δραστηριοτήτων της λειτουργίας αλλαγής στοιχείων ανταλλακτικού.....	37
Εικόνα 3-13: Διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής ανταλλακτικού.	38
Εικόνα 3-14: Διάγραμμα δραστηριοτήτων της λειτουργίας εισαγωγής ιστορικού συντήρησης.....	39
Εικόνα 3-15: Διάγραμμα δραστηριοτήτων της λειτουργίας αλλαγής στοιχείων ιστορικού συντήρησης.	40
Εικόνα 3-16: Διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής ιστορικού συντήρησης.....	41
Εικόνα 3-17: Διάγραμμα δραστηριοτήτων της λειτουργίας αλλαγής ποσότητας αποθέματος.	42
Εικόνα 3-18: Διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής αποθέματος.....	43
Εικόνα 3-19: Διάγραμμα συνεργασίας κατά τη φάση κλιμάκωσης του έργου.	44
Εικόνα 3-20: Διάγραμμα σειράς για επιτυχημένη λειτουργία καταχώρησης.....	45
Εικόνα 3-21: Το διάγραμμα διανομής του έργου.....	46
Εικόνα 3-22: Το ανανεωμένο διάγραμμα περιπτώσεων χρήσης στη φάση κατασκευής του έργου.	47
Εικόνα 3-23: Το ανανεωμένο διάγραμμα τάξεων κατά τη φάση ανάπτυξης του έργου.....	48
Εικόνα 3-24: Διάγραμμα δραστηριοτήτων λειτουργίας αλλαγής στοιχείων μηχανήματος.	49
Εικόνα 3-25: Ανανεωμένο διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής μηχανήματος.	50
Εικόνα 3-26: Ανανεωμένο διάγραμμα δραστηριοτήτων της λειτουργίας αλλαγής στοιχείων ΣΠΣ.....	51
Εικόνα 3-27: Ανανεωμένο διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής ΣΠΣ.	52

Εικόνα 3-28: Ανανεωμένο διάγραμμα δραστηριοτήτων της λειτουργίας αλλαγής στοιχείων ανταλλακτικού.	53
Εικόνα 3-29: Ανανεωμένο διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής ανταλλακτικού.....	54
Εικόνα 3-30: Ανανεωμένο διάγραμμα δραστηριοτήτων της λειτουργίας αλλαγής στοιχείων ιστορικού συντήρησης.....	55
Εικόνα 3-31: Το Ανανεωμένο διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής ιστορικού συντήρησης.....	56
Εικόνα 3-32: Ανανεωμένο διάγραμμα δραστηριοτήτων της λειτουργίας αλλαγής ποσότητας αποθέματος.....	57
Εικόνα 3-33: Ανανεωμένο διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής αποθέματος.....	58
Εικόνα 3-34: Το διάγραμμα συνεργασίας κατά τη φάση κατασκευής του έργου.....	59
Εικόνα 3-35: Ανανεωμένο διάγραμμα σειράς κατά τη φάση της κατασκευής του έργου.....	60
Εικόνα 3-36: Το μοντέλο Οντοτήτων-Συσχετίσεων της βάσης δεδομένων του συστήματος.....	61
Εικόνα 3-37: Το Σχεσιακό μοντέλο της βάσης δεδομένων σε κλασική μορφή.....	62
Εικόνα 3-38: Το Σχεσιακό μοντέλο της βάσης δεδομένων από την εφαρμογή MySQL Workbench.....	63
Εικόνα 4-1: Η οθόνη εισόδου της εφαρμογής.....	65
Εικόνα 4-2: Η κεντρική οθόνη της εφαρμογής.....	66
Εικόνα 4-3: Η οθόνη αλλαγής στοιχείων χρήστη για το διαχειριστή της εφαρμογής.....	67
Εικόνα 4-4: Η οθόνη αλλαγής στοιχείων χρήστη για έναν απλό χρήστη της εφαρμογής.....	67
Εικόνα 4-5: Η σελίδα προβολής των κινήσεων των χρηστών της εφαρμογής.....	68
Εικόνα 4-6: Η οθόνη αντιστοίχισης ανταλλακτικού με μηχανήμα.....	68
Εικόνα 4-7: Η οθόνη αντιστοίχισης ανταλλακτικού με ΣΠΣ.....	69
Εικόνα 4-8: Η οθόνη πληροφοριών.....	70
Εικόνα 4-9: Η οθόνη καταχώρησης ολοκλήρωσης ΣΠΣ.....	72
Εικόνα 4-10: Λειτουργία εισαγωγής μηχανήματος.....	73
Εικόνα 4-11: Λειτουργία αλλαγής στοιχείων μηχανήματος.....	74
Εικόνα 4-12: Λειτουργία διαγραφής μηχανήματος.....	75
Εικόνα 4-13: Λειτουργία εισαγωγής ΣΠΣ.....	76
Εικόνα 4-14: Λειτουργία αλλαγής στοιχείων ΣΠΣ.....	76
Εικόνα 4-15: Λειτουργία διαγραφής ΣΠΣ.....	77
Εικόνα 4-16: Λειτουργία εισαγωγής ανταλλακτικού.....	78
Εικόνα 4-17: Λειτουργία αντιστοίχισης ανταλλακτικού με Μηχάνημα και ΣΠΣ.....	79
Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP	12

Εικόνα 4-18: Λειτουργία αλλαγής στοιχείων ανταλλακτικού.	80
Εικόνα 4-19: Λειτουργία διαγραφής ανταλλακτικού.....	81
Εικόνα 4-20: Λειτουργία εισαγωγής ιστορικού.	82
Εικόνα 4-21: Λειτουργία αλλαγής ιστορικού.	83
Εικόνα 4-22: Λειτουργία διαγραφής ιστορικού.	84
Εικόνα 4-23: Λειτουργία αλλαγής ποσότητας αποθέματος ανταλλακτικού.....	85
Εικόνα 4-24: Πίνακες (πλέγματα – grids) πληροφοριών από την οθόνη προβολής λεπτομερειών.	85
Εικόνα 4-25: Μήνυμα επιτυχούς δημιουργίας αρχείου.	86
Εικόνα 4-26: Λειτουργία καταχώρησης ολοκλήρωσης ΣΠΣ.....	86
Εικόνα 4-27: Επεξήγηση επιπέδων πρόσβασης χρηστών.	88

Κεφάλαιο 1: Εισαγωγή

Η παρούσα εργασία πραγματεύεται την ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP (Rational Unified Process). Το πληροφοριακό σύστημα έχει τη μορφή εφαρμογής που μπορεί να εκτελεστεί στο λειτουργικό σύστημα Windows της εταιρείας Microsoft και η αποθήκευση των, προς διαχείριση, πληροφοριών γίνεται σε βάση δεδομένων. Για την ανάπτυξη του συστήματος χρησιμοποιήθηκαν τεχνικές και μέθοδοι που απαντώνται στην ανάπτυξη σύγχρονων έργων λογισμικού μεγάλης κλίμακας. Χρησιμοποιήθηκαν τεχνολογίες και εργαλεία ευρέως διαδεδομένα και με μεγάλη αποδοχή από κοινότητες ανάπτυξης λογισμικού. Στις παραγράφους αυτού του κεφαλαίου περιλαμβάνεται συνοπτικά το αντικείμενο του πληροφοριακού συστήματος που αναπτύχθηκε και η δομή της εργασίας.

1.1 Αντικείμενο της εργασίας

Το πληροφοριακό σύστημα που αναπτύχθηκε στα πλαίσια της παρούσας εργασίας σχεδιάστηκε για τη διαχείριση εργασιών συντήρησης που πραγματοποιούνται σε μηχανήματα υποβρυχίων. Το σύστημα διατηρεί πληροφορίες για το σύνολο των μηχανημάτων, ανταλλακτικών και αναλώσιμων καθώς επίσης και των προγραμματισμένων, ή έκτακτων, εργασιών συντήρησης που μπορούν να πραγματοποιηθούν. Οι χρήστες καταχωρούν ιστορικά εργασιών συντήρησης και ανανεώνουν τα τρέχοντα αποθέματα ανταλλακτικών ώστε να διευκολύνεται η διαδικασία παραγγελίας τους. Ορισμένες κατηγορίες χρηστών μπορούν να εισάγουν, να διαγράφουν και να αλλάζουν πληροφορίες σχετικά με τις κύριες οντότητες του συστήματος (μηχανήματα, ανταλλακτικά, εργασίες συντήρησης). Αξίζει να σημειωθεί ότι το σύστημα σχεδιάστηκε με βάση συγκεκριμένες ανάγκες του περιβάλλοντος στο οποίο πρόκειται να χρησιμοποιηθεί, αλλά ταυτόχρονα έγινε προσπάθεια ώστε να διατηρηθεί όσο πιο γενικό γίνεται ώστε να μπορεί να εγκατασταθεί και σε διαφορετικά περιβάλλοντα εργασιών συντήρησης.

Το σύστημα περιλαμβάνει μία βάση δεδομένων όπου διατηρούνται οι πληροφορίες του περιβάλλοντος συντήρησης και μία εφαρμογή η οποία λειτουργεί ως γραφικό περιβάλλον πρόσβασης (front-end) της βάσης δεδομένων. Η εφαρμογή αναπτύχθηκε με τη χρήση της γλώσσας προγραμματισμού C# στο περιβάλλον ανάπτυξης Visual Studio της εταιρείας Microsoft. Η βάση δεδομένων διατηρείται στο σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων MySQL. Η επικοινωνία της εφαρμογής με το σύστημα διαχείρισης βάσεων δεδομένων γίνεται με τη χρήσιμη έτοιμη βιβλιοθήκη λογισμικού πρόσβασης στη μορφή dll αρχείου. Έγινε προσπάθεια η εφαρμογή να είναι εύκολη στη χρήση και τέθηκαν περιορισμοί στην εισαγωγή των δεδομένων ώστε να αποτραπούν λάθη από τη μεριά των χρηστών.

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε η επαναληπτική μέθοδος RUP (Rational Unified Process) η οποία επιβάλει ρητό ορισμό των προδιαγραφών του συστήματος και σχεδιασμού των οντοτήτων που το απαρτίζουν. Κατά τις φάσεις ανάπτυξης του έργου οι προδιαγραφές και ο σχεδιασμός του ανανεώνονται με βάση ενδιάμεσα αποτελέσματα ή νέες παρατηρήσεις σχετικά με τη χρήση του. Η μέθοδος RUP προβλέπει επίσης την κατάρτιση διαγραμμάτων συμβατών με το μοντέλο UML (Unified Modeling Language) τα οποία περιγράφουν τη δομή και τις αλληλεπιδράσεις μεταξύ των λειτουργικών οντοτήτων του συστήματος με έναν ενιαία κατανοητό και αποδεκτό τρόπο. Διαγράμματα τέτοιου τύπου κατασκευάστηκαν σε όλες τις φάσεις ανάπτυξης του συστήματος και ανανεώθηκαν με βάση αποφάσεις σχεδιασμού που λήφθηκαν ενδιάμεσα. Όπως προβλέπει η μέθοδος RUP, μετά το πέρας της ανάπτυξης του έργου έγινε έλεγχος της λειτουργικότητας του παραδοτέου προς τους χρήστες ώστε να εξασφαλιστεί ότι παρέχεται η λειτουργικότητα που αρχικά αποφασίστηκε.

1.2 Δομή της εργασίας

Στην παρούσα αναφορά περιγράφονται τα στάδια ανάπτυξης και το τελικό αποτέλεσμα της εργασίας. Πιο αναλυτικά:

Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP

- Στο 2ο κεφάλαιο γίνεται μία βιβλιογραφική επισκόπηση που καλύπτει τον τομέα των συστημάτων διαχείρισης επιχειρησιακών πόρων (Enterprise Resource Planning Systems) και συγκεκριμένα τον τομέα των συστημάτων διαχείρισης εργασιών συντήρησης. Επιπλέον παρουσιάζονται συνοπτικά τα εργαλεία και οι τεχνολογίες που χρησιμοποιήθηκαν. Σημαντικό κομμάτι του κεφαλαίου καταλαμβάνει η περιγραφή της μεθόδου RUP.
- Στο 3ο κεφάλαιο παρουσιάζεται η διαδικασία ανάπτυξης του συστήματος ακολουθώντας το μοντέλο RUP. Αρχικά αναφέρονται οι προδιαγραφές του έργου και στη συνέχεια περιγράφονται αναλυτικά οι φάσεις σύλληψης, κλιμάκωσης και κατασκευής. Παρατίθενται όλα τα UML διαγράμματα που σχεδιάστηκαν στα πλαίσια της διαδικασίας ανάπτυξης καθώς επίσης και όλες οι αλλαγές που έγιναν στις προδιαγραφές ή το σχεδιασμό του συστήματος.
- Στο 4ο κεφάλαιο περιγράφεται η τελική λειτουργικότητα του συστήματος. Πριν το σύστημα παραδοθεί στους χρήστες κρίθηκε σκόπιμο να αναλυθεί η τελική μορφή της εφαρμογής ώστε να φανεί αν εκπληρώνει τους σκοπούς για τους οποίους σχεδιάστηκε. Το κεφάλαιο περιλαμβάνει στιγμιότυπα από τις οθόνες εκτέλεσης καθώς επίσης και αναλυτική περιγραφή όλων των λειτουργιών που βρίσκονται στη διάθεση των χρηστών.
- Στο 5ο και τελευταίο κεφάλαιο της αναφοράς αποτιμάται η επιτυχία εκτέλεσης του έργου και αναλύονται χρήσιμα συμπεράσματα που προέκυψαν κατά τη διαδικασία ανάπτυξης του. Στο δεύτερο μέρος του κεφαλαίου παρουσιάζονται προτάσεις επέκτασης του συστήματος και ζητήματα προς μελλοντική μελέτη.

Κεφάλαιο 2: Βιβλιογραφική Ανασκόπηση

Σε αυτό το κεφάλαιο γίνεται μία σύντομη βιβλιογραφική επισκόπηση. Αρχικά περιγράφεται συνοπτικά το πεδίο των συστημάτων διαχείρισης πόρων και πιο συγκεκριμένα το πεδίο των συστημάτων διαχείρισης συντήρησης. Σε αυτό το πλαίσιο παρουσιάζονται σχετικές εμπορικές, και μη, εφαρμογές. Στο δεύτερο μέρος του κεφαλαίου περιγράφονται οι τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν στα πλαίσια της παρούσας εργασίας.

2.1 Βιβλιογραφική Έρευνα

Το σύστημα που αναπτύχθηκε στα πλαίσια της παρούσας εργασίας ανήκει στην κατηγορία των εφαρμογών διαχείρισης εργασιών συντήρησης (CMMS - Computerized Maintenance Management Systems). Ένας ευρύτερος τομέας εφαρμογών που περιλαμβάνει τα CMMS είναι οι εφαρμογές σχεδιασμού επιχειρησιακών πόρων (ERP – Enterprise Resource Planning) που θα αναλυθούν στην επόμενη παράγραφο.

2.1.1 Συστήματα Σχεδιασμού Επιχειρησιακών Πόρων

Τα Συστήματα Σχεδιασμού Επιχειρησιακών πόρων (ERP – Enterprise Resource Planning) [1] απαντώνται σε ένα μεγάλο εύρος βιομηχανιών με διαφορετικές ανάγκες και μεγέθη και σκοπό έχουν να βοηθήσουν στην καλύτερη οργάνωση των διαδικασιών της εταιρείας ώστε να επιτυγχάνεται μεγαλύτερη παραγωγικότητα και λιγότερα λάθη. Η ανάγκη καταγραφής των πόρων μιας εταιρείας, είτε αυτοί είναι υλικοί (πρώτες ύλες, μηχανήματα), είτε είναι άυλοι (κεφάλαια, τεχνογνωσία) είτε έμψυχοι (εργατικό δυναμικό, συνεργάτες) εμφανίστηκε από τις πρώτες μέρες της σύγχρονης επιχειρηματικότητας. Τα τελευταία χρόνια η διαδικασία της οργάνωσης των διαθέσιμων πόρων συστηματοποιήθηκε και με τη βοήθεια των τεχνολογιών πληροφορικής και επικοινωνιών έγινε διαθέσιμη με τη μορφή υπολογιστικών συστημάτων.

Σε μεγάλες επιχειρήσεις ή οργανισμούς συχνά απαντώνται ενοποιημένα ERP συστήματα που παρέχουν λειτουργικότητα σε διαφορετικά τμήματα της επιχείρησης (ή του οργανισμού). Με αυτό τον τρόπο η πληροφορία από κάθε ένα τμήμα διαχέεται και είναι ιεραρχικά διαθέσιμη. Επιπλέον, διαθέτοντας ένα ενοποιημένο ERP σύστημα, μία εταιρεία εξοικονομεί χρήματα καθώς απευθύνεται σε έναν μόνο προμηθευτή ο οποίος αναλαμβάνει την εγκατάσταση και παραμετροποίηση του συστήματος. Υπάρχουν, βέβαια και περιπτώσεις όπου ένα ενοποιημένο, και γενικό ERP σύστημα δεν ικανοποιεί τις εξειδικευμένες ανάγκες που απαιτεί μία εταιρεία ή ένα τμήμα της εταιρείας. Υπάρχουν επίσης μικρότεροι οργανισμοί και επιχειρήσεις που δε θα μπορούσαν να εκμεταλλευτούν όλες τις δυνατότητες που μπορεί να προσφέρει ένα γενικό ERP σύστημα. Για την επίλυση εξειδικευμένων αναγκών με απλούστερο και οικονομικότερο τρόπο έχουν αναπτυχθεί υποκατηγορίες ERP συστημάτων με πιο περιορισμένο πεδίο δράσης και πιο εξειδικευμένες λειτουργίες. Μερικές από αυτές τις κατηγορίες είναι:

- Συστήματα διαχείρισης πελατειακών σχέσεων (CRM – Customer Relationship Management Systems): Τέτοιες εφαρμογές οργανώνουν τις αλληλεπιδράσεις της εταιρείας με εξωτερικές, κυρίως, οντότητες, όπως π.χ. προμηθευτές, πελάτες, εξωτερικούς συνεργάτες κλπ.
- Συστήματα διαχείρισης εφοδιαστικής αλυσίδας (SCM – Supply Chain Management Systems): Οι εφαρμογές αυτές χρησιμοποιούνται για την καλύτερη οργάνωση και την επιτάχυνση των διαδικασιών παραγωγής και τροφοδοσίας προϊόντων. Οι εφαρμογές αυτές μοντελοποιούν εισερχόμενους πόρους και διαδικασίες με τελικό στόχο την ελαχιστοποίηση του χρόνου παράδοσης του τελικού προϊόντος.

- Συστήματα διαχείρισης αποθεμάτων (Inventory Management Systems): Οι εφαρμογές αυτές χρησιμοποιούνται για τη διαχείριση των πρώτων υλών που παράγει ή διακινεί μία εταιρεία με σκοπό την ταχύτερη αντιμετώπιση ελλείψεων και την πιο αποτελεσματική διαδικασία παραγγελίας.
- Συστήματα Διαχείρισης Αποθήκης (Warehouse Management Systems): Οι εφαρμογές αυτές απευθύνονται σε μεταφορικές εταιρείες ή εταιρείες logistics στις οποίες γίνεται μόνο διακίνηση προϊόντων μεταξύ διαφορετικών σταθμών και σημαντικές ανάγκες είναι η γρήγορη μεταφορά και ο γρήγορος εντοπισμός των φορτίων.
- Συστήματα Διαχείρισης Εργασιών Συντήρησης (Computerized Maintenance Management Systems): Οι εφαρμογές αυτές χρησιμοποιούνται σε τμήματα συντήρησης επιχειρήσεων (ή οργανισμών) ή σε εταιρείες που έχουν ως μόνο αντικείμενο τη συντήρηση μηχανημάτων (συνεργεία, εργαστήρια κλπ).

Τα πλεονεκτήματα από τη χρήση συστημάτων ERP γίνονται εύκολα αντιληπτά. Υπάρχει περισσότερη διαθέσιμη πληροφορία, η οποία αξιολογείται πιο εύκολα από τη διοίκηση του οργανισμού. Ως αποτέλεσμα εντοπίζονται και αντιμετωπίζονται ευκολότερα καθυστερήσεις ή λάθη και αυξάνεται η παραγωγικότητα. Οι χρόνοι παραγωγής ή οι χρόνοι απόκρισης προς αιτήματα πελατών μειώνονται με αποτέλεσμα να αυξάνει η κερδοφορία. Αυτή τη στιγμή διατίθενται μία πληθώρα έτοιμων λύσεων που παρέχουν ERP δυνατότητες. Υπάρχουν δωρεάν λύσεις ανοιχτού κώδικα (VTiger, SugarCRM, dolibarr κ.α.) ενώ στον τομέα δραστηριοποιούνται μεγάλες εταιρείες όπως η SAP, η Microsoft, η Oracle, και άλλες. Μικρότερες εταιρείες, ανάμεσά τους και αρκετές ελληνικές, δραστηριοποιούνται στον τομέα αναπτύσσοντας επεκτάσεις που συνεργάζονται με περίπλοκα ERP συστήματα ή εγκαθιστώντας και παραμετροποιώντας τέτοιες λύσεις.

Έχοντας αναφέρει τα παραπάνω, αξίζει να υπογραμμιστούν ορισμένες προκλήσεις που ανακύπτουν στο σχεδιασμό και την εγκατάσταση ERP συστημάτων. Η 1^η σημαντική πρόκληση αφορά την ευχρηστία αυτών των συστημάτων καθώς συχνά απευθύνονται σε χρήστες χωρίς υπόβαθρο στην Πληροφορική. Οι επαφές με τους χρήστες πρέπει να είναι όσο το δυνατό απλούστερες ώστε να αποφεύγεται η λανθασμένη εισαγωγή στοιχείων. Η 2^η μεγάλη πρόκληση είναι η ορθή επιλογή και παραμετροποίηση ενός ERP με βάση τις ανάγκες του οργανισμού στον οποίο θα εγκατασταθεί. Πολλές φορές μπορεί να είναι προτιμότερη η αγορά ή κατασκευή ενός συστήματος με λιγότερες αλλά πιο εξειδικευμένες δυνατότητες από την προμήθεια και παραμετροποίηση ενός πιο περίπλοκου συστήματος. Τέλος, σημαντικός παράγον που πρέπει πάντα να λαμβάνεται υπόψη είναι η ασφάλεια των δεδομένων που διαχειρίζεται ένα ERP σύστημα καθώς αντικατοπτρίζουν μεγάλο μέρος από την κατάσταση και τις προσδοκίες μιας επιχείρησης ή ενός οργανισμού.

2.1.2 Συστήματα διαχείρισης εργασιών συντήρησης

Τα συστήματα διαχείρισης εργασιών συντήρησης (CMMS - (Computerized Maintenance Management Systems) [2] απευθύνονται σε επιχειρήσεις και οργανισμούς (ή τμήματα αυτών) που ασχολούνται με το αντικείμενο της συντήρησης. Για το σκοπό αυτό συνήθως διατηρούν συγκεκριμένες πληροφορίες σχετικά με τον οργανισμό που εξυπηρετούν. Οι πιο σημαντικές από αυτές είναι:

- Σύνολο εξοπλισμού και διαθέσιμο ανταλλακτικών: Η εφαρμογή τηρεί αρχείο για όλα τα μηχανήματα και εργαλεία που εμπλέκονται στη συντήρηση καθώς επίσης και για τα ανταλλακτικά και αναλώσιμα που χρησιμοποιούνται σε εργασίες συντήρησης ή επισκευής.
- Διαδικασίες προγραμματισμένης συντήρησης: Η εφαρμογή διατηρεί πληροφορία για τις διαδικασίες συντήρησης τόσο σε επίπεδο λειτουργιών όσο και σε επίπεδο αναγκών σε ανταλλακτικά και αναλώσιμα.
- Καταγραφή κατάστασης εξοπλισμού: Η εφαρμογή κρατά πληροφορία σχετικά με την κατάσταση στην οποία βρίσκονται τα μηχανήματα και εργαλεία που διαχειρίζονται από αυτή.

- Καταγραφή ιστορικού εργασιών: Η εφαρμογή τηρεί πληροφορία για τις εργασίες , προγραμματισμένες ή μη, που έχουν γίνει στα μηχανήματα που διαχειρίζονται από αυτήν.
- Καταγραφή αποθεμάτων ανταλλακτικών: Εφαρμογές διαχείρισης συντήρησης συχνά αποθηκεύουν αναλυτικές πληροφορίες σχετικά με το απόθεμα που διατηρούν σε ανταλλακτικά και αναλώσιμα.
- Οικονομικά στοιχεία: Πολύ συχνά εφαρμογές διατήρησης συντήρησης προσφέρουν δυνατότητες αυτόματης τιμολόγησης εργασιών με βάση τις εργατοώρες και τα υλικά που δαπανήθηκαν.
- Αρχείο προσωπικού: Καταγράφονται τα πρόσωπα ή οι ρόλοι που συμμετέχουν στις διαδικασίες συντήρησης.

Υπάρχουν αρκετά εξειδικευμένα προγράμματα διαχείρισης εργασιών συντήρησης στην αγορά λογισμικού. Από την ελληνική αγορά ξεχωρίζουν οι λύσεις AIMMS της εταιρείας Atlantis Engineering [3], iMaint της εταιρείας Ολοκληρωμένα Πληροφοριακά Συστήματα [4], IMIS της εταιρείας IT Open Solutions [5], και το CMMS της εταιρείας ANCO [6]. Όλες οι παραπάνω εφαρμογές διαθέτουν τις κυριότερες λειτουργίες ενός συστήματος διαχείρισης εργασιών συντήρησης και μπορούν να παραμετροποιηθούν ανάλογα με τις ανάγκες της επιχείρησης στην οποία θα εγκατασταθούν. Μπορούν να δράσουν αυτόνομα ή σε συνεργασία με μεγαλύτερα ERP συστήματα. Υπάρχουν ακόμη πιο εξειδικευμένα προγράμματα όπως η εφαρμογή ANEΛΚΥΣΤΗΡΑΣ της εταιρείας ΜΗΧΑΝΟΓΡΑΦΗΣΗ [7] που στοχεύει στη διαχείριση περιοδικών συντηρήσεων ανελκυστήρων και η εφαρμογή Fleetmaster της εταιρείας LOGIFER [8] που αφορά τη συντήρηση οχημάτων. Στην ξένη αγορά λογισμικού υπάρχουν πληθώρα παρόμοιων προγραμμάτων για διαφορετικές ανάγκες χρηστών και οργανισμών.

2.2 Τεχνολογίες και Εργαλεία

Στα πλαίσια της παρούσας εργασίας, για την κατασκευή του συστήματος διαχείρισης εργασιών συντήρησης, χρησιμοποιήθηκαν σύγχρονα εργαλεία και τεχνολογίες που γνωρίζουν ευρεία αποδοχή και αποτελούν συνήθεις επιλογές σε διαδικασίες υλοποίησης παρόμοιων υπολογιστικών συστημάτων. Τα σημαντικότερα από τα εργαλεία αυτά παρουσιάζονται στις επόμενες παραγράφους.

2.2.1 Το σύστημα διαχείρισης βάσεων δεδομένων (ΣΔΒΔ) mysql

Τα περισσότερα σύγχρονα υπολογιστικά συστήματα χρησιμοποιούν ειδικό λογισμικό διαχείρισης βάσεων δεδομένων για την αποθήκευση των πληροφοριών που χειρίζονται. Ως Βάση Δεδομένων ορίζεται μια σαφώς δομημένη συλλογή πληροφοριών ή εγγράφων που βρίσκονται αποθηκευμένα σε υπολογιστικό σύστημα. Τα τελευταία χρόνια έχει επικρατήσει η χρήση σχεσιακών βάσεων δεδομένων στις οποίες οι πληροφορίες αποθηκεύονται με τη μορφή πινάκων που σχετίζονται μεταξύ τους μέσω ρητά ορισμένων συνδέσμων. Μια ηλεκτρονική βάση δεδομένων χρησιμοποιεί ειδικό λογισμικό για την οργάνωση της αποθήκευσης των δεδομένων της. Το λογισμικό αυτό είναι γνωστό ως Σύστημα Διαχείρισης Βάσης Δεδομένων (DBMS - Database Management System) και αναλαμβάνει τη διαφανή διαχείριση των αποθηκευμένων, σε αυτό, πληροφοριών[9].

Στα πλαίσια της παρούσας εργασίας χρησιμοποιήθηκε το σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων MySQL [10]. Το ΣΔΒΔ MySQL δημιουργήθηκε το 1994 από τους Michael Widenius και David Axmark και παρέχει δυνατότητες δημιουργίας, διαχείρισης και ανάκλησης εγγράφων από σχεσιακές βάσεις δεδομένων. Το συγκεκριμένο ΣΔΒΔ επιλέχθηκε λόγω της ευρείας αποδοχής του, της εύκολης και πλήρως τεκμηριωμένης παραμετροποίησής του και του μηδενικού κόστους απόκτησης. Το MySQL ξεκίνησε ως λογισμικό ανοιχτού κώδικα και επεκτάθηκε σημαντικά λόγω της πολύ δραστήριας κοινότητας προγραμματιστών που ενεπλάκη με αυτό. Το 2008, μετά από σχετική εξαγορά [11], τα δικαιώματα ανάπτυξης και διάθεσης του MySQL πέρασαν στην εταιρεία Sun Microsystems. Το 2009 μετά την εξαγορά της Sun από την εταιρεία Oracle[12], βασικού κατασκευαστή ΣΔΒΔ, ξεκίνησαν

αρκετές συζητήσεις σχετικά με την άδεια χρήσης του MySQL [13]. Αυτή τη στιγμή διατίθενται τόσο εκδόσεις που απαιτούν πληρωμή και απευθύνονται σε επιχειρήσεις όσο και δωρεάν εκδόσεις που διατίθενται για εκπαιδευτικούς σκοπούς ή μη κερδοσκοπικές δραστηριότητες.

Η κυριότερη λειτουργική οντότητα του συστήματος MySQL είναι ο εξυπηρετητής MySQL (MySQL Server) ο οποίος δέχεται ερωτήματα από τοπικούς MySQL Clients αλλά και από απομακρυσμένα συστήματα (π.χ. μέσω άλλων εφαρμογών όπως αυτή που αναπτύχθηκε). Τα ερωτήματα που δέχεται ο MySQL Server χωρίζονται σε εντολές διαχείρισης και εντολές SQL. Οι εντολές διαχείρισης χρησιμοποιούνται για την έναρξη συνδέσεων, για την αναγνώριση και αυθεντικοποίηση χρηστών, για την ανάθεση ξεχωριστών δικαιωμάτων σε διαφορετικές κατηγορίες χρηστών, για εξειδικευμένες ρυθμίσεις ασφάλειας και κρυπτογράφησης και γενικά για τη ρύθμιση των παραμέτρων του εξυπηρετητή.

Οι SQL εντολές χρησιμοποιούνται για το σχεδιασμό της Βάσης Δεδομένων που θα χρησιμοποιηθεί στα πλαίσια της εφαρμογής (δηλαδή τη δημιουργία πινάκων) και τη διαχείριση των δεδομένων της (εισαγωγή, διαγραφή και αναζήτηση δεδομένων). Κάποιες από τις πιο σημαντικές από αυτές τις εντολές είναι οι ακόλουθες :

- CREATE DATABASE (δημιουργία μιας Βάσης Δεδομένων)
- CREATE TABLE (ορισμός των πεδίων ενός πίνακα της Βάσης Δεδομένων)
- INSERT (εισαγωγή εγγραφών σε κάποιον πίνακα της Βάσης Δεδομένων)
- DELETE (διαγραφή εγγραφών από κάποιον πίνακα της Βάσης Δεδομένων)
- UPDATE (αλλαγή εγγραφών σε κάποιον πίνακα της Βάσης Δεδομένων)
- SELECT (αναζήτηση δεδομένων από έναν ή περισσότερους πίνακες της Βάσης Δεδομένων)

Εκτός από τις παραπάνω υπάρχουν αρκετές άλλες εντολές SQL οι οποίες αφορούν πιο περίπλοκες εφαρμογές και είναι εκτός των στόχων της παρούσας εργασίας.

Εκτός από τον εξυπηρετητή MySQL χρησιμοποιήθηκε η εφαρμογή MySQL workbench για τη διαχείριση της Βάσης Δεδομένων καθώς επίσης και η αντίστοιχη βιβλιοθήκη για τη γλώσσα C# η οποία επιτρέπει την πρόσβαση της εφαρμογής που αναπτύχθηκε στη βάση δεδομένων.

2.2.2 Η γλώσσα προγραμματισμού C# και το περιβάλλον .NET

Η C#(C Sharp)[14] είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού η οποία δημιουργήθηκε από την εταιρεία λογισμικού Microsoft στα πλαίσια της προώθησης του περιβάλλοντος .NET. Η γλώσσα C# έχει αρκετά κοινά στοιχεία και παρόμοια σύνταξη με τις γλώσσες C++ και Java, γεγονός που βοήθησε στη γρήγορη υιοθέτησή της από μεγάλο μέρος της κοινότητας προγραμματιστών εφαρμογών για το λειτουργικό σύστημα Windows. Η C# αναπτύχθηκε ώστε να υποστηρίζει το περιβάλλον .NET. Κύριο χαρακτηριστικό της γλώσσας είναι ότι ο μεταγλωττιστής της δε δημιουργεί απευθείας κώδικα μηχανής, όπως για παράδειγμα οι μεταγλωττιστές για της C++, άλλα ενδιάμεσο MSIL (Microsoft Intermediate Language) κώδικα που θα εκτελεστεί από το περιβάλλον .NET. Η παράγωγή ενδιάμεσου κώδικα ομοιάζει πολύ με την αντίστοιχη διαδικασία στη μεταγλώττιση και εκτέλεση προγραμμάτων Java.

Το .NET [15] είναι η πλέον διαδεδομένη πλατφόρμα ανάπτυξης εφαρμογών για το λειτουργικό σύστημα Windows. Κύριος σκοπός ανάπτυξης του περιβάλλοντος .NET από τη Microsoft, ήταν η απλοποίηση της υλοποίησης εφαρμογών αποκρύπτοντας τις λεπτομέρειες υλοποίησης λειτουργιών όπως η διαχείριση μνήμης, η δικτυακή επικοινωνία και η επικοινωνία με άλλες συσκευές και παρέχοντας κατάλληλες διεπαφές για τη χρήση τους. Η πλατφόρμα .NET δημιουργεί ένα ελεγχόμενο και ασφαλές περιβάλλον μέσα στο οποίο εκτελούνται οι εφαρμογές(sandbox). Εκτός από τη γλώσσα C#, το .NET υποστηρίζει και άλλες γλώσσες προγραμματισμού όπως οι Visual Basic.NET, J++ και Managed C++ οι οποίες διαθέτουν κατάλληλους .NET μεταγλωττιστές . Για την ακρίβεια, το περιβάλλον .NET “καταλαβαίνει” μόνο τη γλώσσα MSIL, οπότε, οποιαδήποτε γλώσσα προγραμματισμού μπορεί να

μεταγλωττιστεί σε αυτή μπορεί να εκτελεστεί στην πλατφόρμα .NET. Κύρια οντότητα του περιβάλλοντος .NET είναι το κοινό περιβάλλον εκτέλεσης (CLR – Common Language Runtime). Η οντότητα αυτή ορίζει το κλειστό περιβάλλον μέσα στο οποίο εκτελούνται οι εφαρμογές. Το CLR θα μπορούσε να αντιστοιχηθεί με την εικονική μηχανή της Java (JVM – Java Virtual Machine) και παρέχει αυτοματοποιημένη διαχείριση μνήμης, ασφάλεια εκτέλεσης και διεπαφές για επικοινωνία με το λειτουργικό σύστημα.

Η χρήση της γλώσσας προγραμματισμού C# επελέγη κυρίως λόγω της εύκολης εκμάθησής της και της ευρείας χρήσης της σε εφαρμογές για το λειτουργικό σύστημα Windows. Η χρήση του περιβάλλοντος .NET απλοποίησε αρκετές από τις διαδικασίες ανάπτυξης της εφαρμογής και βοήθησε στην γρήγορη κατασκευή ενός γραφικού περιβάλλοντος πλήρως εναρμονισμένου με το λειτουργικό σύστημα.

2.2.3 Το περιβάλλον ανάπτυξης Microsoft Visual Studio

Το Microsoft Visual Studio είναι ένα ενοποιημένο περιβάλλον ανάπτυξης (IDE – Integrated Development Environment) που κατασκευάζει και διαθέτει η εταιρεία Microsoft [16]. Το περιβάλλον χρησιμοποιείται κυρίως για την ανάπτυξη εφαρμογών στο λειτουργικό σύστημα Windows και παρέχει μεταγλωττιστές για τις περισσότερες γλώσσες προγραμματισμού που υποστηρίζει το .NET. Οι κυριότερες από τις λειτουργίες του Microsoft Visual Studio που χρησιμοποιήθηκαν στα πλαίσια αυτής της εργασίας ήταν:

- Η σχεδίαση φορμών και η παραγωγή κώδικα ο οποίος επεκτείνεται για να υποστηρίξει την επιθυμητή λειτουργικότητα.
- Οι δυνατότητες αυτόματης συμπλήρωσης κώδικα, επισήμανσης λαθών και αυτόματης μορφοποίησης που παρέχει οθόνη συγγραφής κώδικα (code editor).
- Οι δυνατότητες αποσφαλμάτωσης που προσφέρει ο ενσωματωμένος αποσφαλματωτής (debugger), οι οποίες βοηθούν στον εντοπισμό και τη διόρθωση λαθών του κώδικα.
- Τα γραφικά εργαλεία διαχείρισης των κλάσεων και των μεθόδων της εφαρμογής.

Το Microsoft Visual Studio είναι διαθέσιμο τόσο σε δωρεάν εκδόσεις (Express Editions) όσο και σε εκδόσεις επί πληρωμή (Professional ή Ultimate Editions). Υπάρχουν επίσης διαφορετικές εκδόσεις ανάλογα με τη γλώσσα προγραμματισμού η οποία θα χρησιμοποιηθεί. Το περιβάλλον Visual Studio μπορεί να συνδυαστεί με άλλα προϊόντα της Microsoft σχετικά με τη διατήρηση διαφορετικών εκδόσεων του κώδικα και τη συνεργασία ομάδας προγραμματιστών.

2.2.4 Η διαδικασία ανάπτυξης RUP (Rational Unified Process)

Η διαδικασία ανάπτυξης λογισμικού RUP (Rational Unified Process) είναι μια ρητά ορισμένη μέθοδος διαχείρισης, σχεδιασμού και ανάπτυξης έργων λογισμικού μεγάλης κλίμακας [17]. Είναι, για την ακρίβεια, ένα ολοκληρωμένο πλαίσιο ελέγχου της το οποίο περιλαμβάνει τα ακόλουθα χαρακτηριστικά:

- Σχεδιασμό με βάση τις οντότητες: Το σύνολο του έργου χωρίζεται σε ανεξάρτητες δομικές μονάδες λογισμικού.
- Δομημένη οργάνωση των προδιαγραφών: οι προδιαγραφές του έργου δομούνται κατάλληλα ώστε να προκύψουν οντότητες σαφώς ορισμένες διεπαφές αλληλεπίδρασης.
- Επαναληπτική ανάπτυξη κώδικα: τα τμήματα του κώδικα που αναπτύσσονται ελέγχονται, επεκτείνονται και συμπληρώνονται όσο μεγαλώνει το μέγεθος του συνολικού έργου.
- Έλεγχος των αλλαγών στον κώδικα: Τηρούνται στοιχεία για τις διαφορετικές εκδόσεις κομματιών του κώδικα και τις αλλαγές που έγιναν σε αυτά.

- Έλεγχο ποιότητας: Στα ενδιάμεσα στάδια ανάπτυξης ελέγχεται η ποιότητα των αποτελεσμάτων.
- Σαφώς ορισμένο τρόπο διαχείρισης του έργου: Οι διαδικασίες και τα πρόσωπα που συμμετέχουν καθορίζονται ρητά σε σχετικά έγγραφα.
- Γραφική αναπαράσταση των αυτόνομων οντοτήτων λογισμικού: Οι οντότητες του έργου λογισμικού και οι αλληλεπιδράσεις τους αναπαρίστανται γραφικά με χρήση του προτύπου UML (Unified Markup Language).

Η RUP ορίζει σαφώς τις φάσεις ανάπτυξης λογισμικού ως εξής:

- **Φάση σύλληψης (Inception Phase):** Σε αυτή τη φάση επιχειρείται η αρχική συνεργασία των εμπλεκόμενων ομάδων και ατόμων. Επίσης ορίζονται τα άτομα-διεπαφές(contact persons) κάθε ομάδας. Γίνεται μία αρχική καταγραφή των απαιτήσεων του έργου λογισμικού με τη συμμετοχή των μελλοντικών χρηστών του (αν αυτό είναι εφικτό). Οι αναλυτές του έργου ανατρέχουν σε παλαιότερες παρόμοιες περιπτώσεις έργου ώστε να υποβάλουν καίριες ερωτήσεις προς τους χρήστες του έργου. Ορίζεται χρονοδιάγραμμα συναντήσεων. Στο τέλος καθεμιάς από αυτές συντάσσεται έγγραφο με τα αποτελέσματα της συνάντησης. Στο στάδιο αυτό ξεκινά και η καταγραφή του προϋπολογισμού του έργου με βασική μονάδα μέτρησης τον ανθρωπομήνα (PM – Person Month). Στον προϋπολογισμό του έργου περιλαμβάνονται εκτός από τα προφανή κόστη εργασίας, επιπλέον έξοδα, όπως έξοδα εκπαίδευσης, μετακίνησης, μεταφοράς τεχνογνωσίας, αγοράς υλικού ή/και λογισμικού. Σε αυτό το στάδιο γίνεται μία αρχική ανάλυση κινδύνου ώστε να εντοπιστούν ενδεχόμενα καθυστερήσεις στην ανάλυση ή την ανάπτυξη του λογισμικού. Καθορίζονται εκτός από τον ακριβή σκοπό του έργου, οι τεχνολογίες και τα εργαλεία ανάπτυξης. Στη φάση της σύλληψης περιλαμβάνεται συχνά η καταγραφή περιπτώσεων χρήσης(Use Cases) ή και η κατασκευή πρωτοτύπου σε ένα ποσοστό 10-20% της τελικής λειτουργικότητας του έργου.
- **Φάση Κλιμάκωσης(Elaboration Phase):** Σε αυτή τη φάση ορίζεται το μεγαλύτερο μέρος (~80%) των περιπτώσεων χρήσης του έργου (Use Cases). Ως περιπτώσεις χρήσης ορίζονται όλα τα σενάρια αλληλεπίδρασης μεταξύ των ρόλων (χρηστών ή εξωτερικών συστημάτων) που συμμετέχουν στο έργο. Για κάθε περίπτωση χρήσης καταγράφονται οι αρχικές συνθήκες (προαπαιτούμενα), οι ενέργειες που μπορεί να εκτελέσει ο χρήστης και τα πιθανά μονοπάτια ανάλογα με τις επιλογές ή την κατάσταση του συστήματος. Αφού οι περιπτώσεις χρήσης καταγραφούν, αξιολογούνται με βάση την πολυπλοκότητα ώστε να εντοπιστούν κρίσιμοι χρόνοι παράδοσης. Εκτός από την καταγραφή των περιπτώσεων χρήσης, στη φάση της κλιμάκωσης σχεδιάζεται η διεπαφή του λογισμικού με το χρήστη (χωρίς ακόμη να παρέχεται η αντίστοιχη λειτουργικότητα).
- **Φάση κατασκευής(Construction Phase):** Σε αυτή τη φάση ξεκινά η πραγματική κατασκευή του έργου με τρόπο σταδιακό και αυξητικό. Σε κάθε στάδιο ανάπτυξης ελέγχεται η ποιότητα του αποτελέσματος. Τυχόν λεπτομέρειες της υλοποίησης ή περιπτώσεων χρήσης καθορίζονται επίσης σε αυτή τη φάση. Το τελικό προϊόν αυτής της φάσης είναι λογισμικό έτοιμο να παραδοθεί στους χρήστες. Πριν την παράδοση πρέπει να ελεγχθεί αν το έργο μπορεί να παραδοθεί (είναι λειτουργικό κατά το μεγαλύτερο μέρος) και σε τι ποσοστό υλοποιεί τη λειτουργικότητα που συμφωνήθηκε.
- **Φάση Μετάβασης (Transition Phase):** Στη φάση αυτή γίνεται η εφαρμογή του λογισμικού στο περιβάλλον χρηστών για το οποίο σχεδιάστηκε. Σε αυτή τη φάση το λογισμικό τυποποιείται, αποστέλλεται και εγκαθίσταται. Γίνεται επίσης συγγραφή των σχετικών εγχειριδίων χρήσης(manuals) και η εκπαίδευση των χρηστών. Ορίζεται επιπλέον ομάδα τεχνικής υποστήριξης η οποία αναλαμβάνει τον έλεγχο του έργου. Αφού το λογισμικό εγκατασταθεί αξιολογείται ως προς την παρεχόμενη λειτουργικότητα. Τυχόν προβλήματα αντιμετωπίζονται από την τεχνική υποστήριξη ή προωθούνται στην ομάδα ανάπτυξης.

Κεφάλαιο 3: Ανάπτυξη Εφαρμογής

Στο κεφάλαιο αυτό θα παρουσιαστεί αναλυτικά η διαδικασία ανάλυσης και κατασκευής του συστήματος διαχείρισης εργασιών συντήρησης. Όπως αναφέρθηκε και στην εισαγωγή, για την κατασκευή της εφαρμογής χρησιμοποιήθηκε το μοντέλο RUP (Rational Unified Process). Ως αποτέλεσμα, μεγάλο μέρος αυτού του κεφαλαίου αποτελείται από την παράθεση UML διαγραμμάτων και σχεδίων που χρησιμοποιούνται στο μοντέλο RUP. Οι απαιτήσεις της εφαρμογής περιγράφονται, αρχικά, τόσο σε φυσική γλώσσα, όσο και στην πιο αναλυτική μορφή των περιπτώσεων χρήσης (Use Cases). Καλύπτονται με τη σειρά οι φάσεις της σύλληψης (ανάλυσης), κλιμάκωσης (εκπόνησης μελέτης) και κατασκευής. Για την πρώτη φάση παρατίθενται διαγράμματα περιπτώσεων χρήσης και κλάσεων. Για τη δεύτερη φάση ορίζονται επιπλέον διαγράμματα δραστηριοτήτων, συνεργασίας, σειράς και διανομής. Στα πλαίσια της φάσης κατασκευής τα διαγράμματα της προηγούμενης φάσης συμπληρώνονται και ανανεώνονται με βάση νέες σχεδιαστικές αποφάσεις που λήφθηκαν κατά τη φάση αυτή. Επιπλέον παρουσιάζεται το μοντέλο Οντοτήτων Συσχετίσεων (E-R Model) και το Σχεσιακό Μοντέλο (Relational Model) της βάσης δεδομένων που αναπτύχθηκε για να υποστηρίξει την εφαρμογή στα πλαίσια του συστήματος.

3.1 Ορισμός απαιτήσεων σε φυσική γλώσσα

Η εφαρμογή που αναπτύχθηκε στα πλαίσια της παρούσας εργασίας ένα γραφικό περιβάλλον διαχείρισης εργασιών συντήρησης σε μηχανήματα υποβρυχίων. Η χρήση της μπορεί να επεκταθεί για διαχείριση εργασιών συντήρησης σε άλλα περιβάλλοντα με πιο γενικές ανάγκες. Η εφαρμογή οφείλει να διατηρεί πληροφορία για τα μηχανήματα, τα ανταλλακτικά τα συστήματα προγραμματισμένης συντήρησης (ΣΠΣ), το ιστορικό εργασιών συντήρησης και των τρεχόντων αποθεμάτων ανταλλακτικών. Πρέπει να υποστηρίζεται η διαβάθμιση των χρηστών της εφαρμογής ορίζοντας διαφορετικά επίπεδα πρόσβασης που διαφοροποιούν τις διαθέσιμες λειτουργίες. Οι χρήστες της, ανάλογα με το επίπεδο πρόσβασής τους, έχουν τη δυνατότητα δημιουργίας, διαγραφής ή αλλαγής μηχανημάτων, ανταλλακτικών, ΣΠΣ, εγγραφών ιστορικού ή αποθεμάτων ανταλλακτικών. Το ανώτερο επίπεδο πρόσβασης χρήστη περιλαμβάνει το διαχειριστή της εφαρμογής ο οποίος μπορεί να δημιουργήσει, να διαγράψει και να αλλάξει τις πληροφορίες για τους χρήστες της εφαρμογής. Η πρόσβαση προστατεύεται με σχετικό κωδικό, ο οποίος μπορεί να αλλάξει όταν κάθε χρήστης το επιθυμεί.

3.2 Προδιαγραφές

Στο παρόν υποκεφάλαιο θα περιγραφούν με πιο αναλυτικό τρόπο οι κύριες έννοιες της εφαρμογής, τα διαφορετικά επίπεδα πρόσβασης των χρηστών και οι λειτουργίες που μπορούν να εκτελέσουν.

3.2.1 Εννοιολογικές οντότητες της εφαρμογής

Αρχικά περιγράφονται οι οντότητες της εφαρμογής:

- **Μηχάνημα:** Είναι η πιο σημαντική οντότητα της εφαρμογής και λειτουργεί ως αναφορά για τις υπόλοιπες οντότητες. Περιγράφει ένα συγκεκριμένο μηχανήμα για το οποίο η εφαρμογή πρέπει να διατηρεί πληροφορίες. Διατηρούνται οι μέγιστες και τρέχουσες ώρες λειτουργίας του καθώς επίσης και οι διασυνδέσεις του με άλλες οντότητες της εφαρμογής.
- **Ανταλλακτικό:** Η οντότητα αυτή περιγράφει ανταλλακτικά (ή αναλώσιμα) που μπορούν να χρησιμοποιηθούν για την επισκευή ή συντήρηση μηχανημάτων. Κάθε ανταλλακτικό χρησιμοποιείται για ένα μηχανήμα ενώ κάθε μηχανήμα μπορεί να χρησιμοποιεί πολλά ανταλλακτικά. Κατά την εισαγωγή νέου ανταλλακτικού ορίζεται το μηχανήμα και το

σύστημα προγραμματισμένης συντήρησης (ΣΠΣ) για το οποίο προορίζεται. Κάθε ανταλλακτικό περιγράφεται από μοναδικό κωδικό NSN.

- **Σύστημα Προγραμματισμένης Συντήρησης (ΣΠΣ):** Η οντότητα αυτή περιγράφει ένα σύνολο ενεργειών που μπορεί να πραγματοποιηθούν για μία συγκεκριμένη εργασία συντήρησης ενός μηχανήματος. Κάθε ΣΠΣ ορίζεται για ένα μηχάνημα ενώ κάθε μηχάνημα μπορεί να διαθέτει πολλά ΣΠΣ. Κατά την εισαγωγή νέου ΣΠΣ ορίζεται το μηχάνημα για το οποίο προορίζεται. Διατηρείται επίσης πληροφορία για τα ανταλλακτικά τα οποία πρέπει να χρησιμοποιηθούν για την ολοκλήρωσή του. Επιπλέον ορίζεται η περιοδικότητά του (ημερήσιο, εβδομαδιαίο, μηνιαίο, εξαμηνιαίο ή ετήσιο).
- **Απόθεμα Ανταλλακτικού:** Η οντότητα αυτή περιγράφει το διαθέσιμο απόθεμα που υπάρχει για κάθε ανταλλακτικό. Η πληροφορία σχετικά με το τρέχον απόθεμα παρουσιάζεται και αποθηκεύεται ξεχωριστά καθώς σε μελλοντική επέκταση της εφαρμογής μπορεί να ανανεώνεται από διαφορετική πηγή δεδομένων (π.χ. κάποια άλλη βάση δεδομένων, ένα σύστημα διαχείρισης αποθήκης κ.ο.κ.). Κλειδί για την αναζήτηση και ανανέωση αποθέματος είναι ο NSN κωδικός ο οποίος αντιστοιχεί στον NSN κωδικό κάποιου ανταλλακτικού του συστήματος.
- **Ιστορικό:** Η οντότητα αυτή περιγράφει ένα ιστορικό εργασιών συντήρησης οι οποίες μπορεί να είναι είτε προγραμματισμένες (προερχόμενες από κάποιο Σ.Π.Σ. είτε έκτακτες). Για κάθε ιστορικό εργασίας συντήρησης αποθηκεύεται η περιγραφή της, το μηχάνημα στο οποίο πραγματοποιείται, η ημερομηνία εκτέλεσης και το ΣΠΣ από το οποίο προκύπτει (αν είναι προγραμματισμένη εργασία συντήρησης και όχι έκτακτη επισκευή).

3.2.2 Χρήστες και επίπεδα πρόσβασης

Οι χρήστες της εφαρμογής χωρίζονται σε 3 κατηγορίες.:

- Η πρώτη κατηγορία (απλοί χρήστες) έχει τα μικρότερα δικαιώματα πρόσβασης και αφορά χρήστες οι οποίοι μπορούν μόνο να διαβάσουν πληροφορίες σχετικά με τα διαθέσιμα μηχανήματα, ανταλλακτικά και συστήματα προγραμματισμένης συντήρησης (Σ.Π.Σ.) και μπορούν να καταχωρήσουν πληροφορίες αποθεματικών και ιστορικού εκτέλεσης κάποιας προγραμματισμένης – ή μη – εργασίας συντήρησης. Οι χρήστες αυτής της κατηγορίας δε μπορούν να δημιουργήσουν νέα μηχανήματα, ανταλλακτικά ή συστήματα προγραμματισμένης συντήρησης.
- Η δεύτερη κατηγορία (αναβαθμισμένοι χρήστες) έχει αυξημένα δικαιώματα πρόσβασης επιτρέποντας την εισαγωγή, διαγραφή και μετατροπή μηχανημάτων, ανταλλακτικών και συστημάτων προγραμματισμένης συντήρησης.
- Τέλος, η τρίτη κατηγορία αφορά το διαχειριστή της εφαρμογής ο οποίος εκτός από τα παραπάνω δικαιώματα πρόσβασης, μπορεί να δημιουργήσει και να διαγράψει χρήστες της εφαρμογής. Η ανάθεση του επιπέδου πρόσβασης κάθε χρήστη γίνεται από το διαχειριστή της εφαρμογής.

3.2.3 Λειτουργίες της εφαρμογής

Η εφαρμογή παρέχει διαφορετικές λειτουργίες ανάλογα με το επίπεδο πρόσβασης κάθε χρήστη. Στην παρούσα παράγραφο παρουσιάζονται συνοπτικά όλες οι λειτουργίες της εφαρμογής:

- **Εισαγωγή μηχανήματος:** Με τη λειτουργία αυτή μία νέα εγγραφή μηχανήματος εισάγεται στο σύστημα. Πρέπει να συμπληρωθεί το όνομα του μηχανήματος, οι μέγιστες και τρέχουσες ώρες λειτουργίας. Όλα τα πεδία πρέπει να συμπληρωθούν και δεν πρέπει να υπάρχει εγγραφή μηχανήματος με το ίδιο όνομα. Η λειτουργία είναι διαθέσιμη μόνο σε αναβαθμισμένους χρήστες και το διαχειριστή της εφαρμογής.

- **Αλλαγή μηχανήματος:** Με τη λειτουργία αυτή αλλάζουν τα πεδία μίας υπάρχουσας εγγραφής μηχανήματος. Η ανανέωση προϋποθέτει την εισαγωγή του κωδικού της εγγραφής. Γίνεται έλεγχος αν όντως υπάρχει εγγραφή με αυτό τον κωδικό. Όλα τα πεδία πρέπει να έχουν τιμές και δεν πρέπει να υπάρχει εγγραφή μηχανήματος με το ίδιο όνομα. Η λειτουργία είναι διαθέσιμη μόνο σε αναβαθμισμένους χρήστες και το διαχειριστή της εφαρμογής.
- **Διαγραφή μηχανήματος:** Με τη λειτουργία αυτή διαγράφεται μία εγγραφή μηχανήματος. Η διαγραφή προϋποθέτει την εισαγωγή του κωδικού της εγγραφής. Γίνεται έλεγχος αν όντως υπάρχει εγγραφή με αυτό τον κωδικό. Πριν την οριστική διαγραφή ζητείται επιβεβαίωση από το χρήστη. Η λειτουργία είναι διαθέσιμη μόνο σε αναβαθμισμένους χρήστες και το διαχειριστή της εφαρμογής.
- **Εισαγωγή ΣΠΣ** Με τη λειτουργία αυτή μία νέα εγγραφή ΣΠΣ εισάγεται στο σύστημα. Πρέπει να συμπληρωθεί το όνομα του ΣΠΣ, το μηχάνημα στο οποίο εφαρμόζεται και η περιοδικότητά του. Όλα τα πεδία πρέπει να συμπληρωθούν και δεν πρέπει να υπάρχει εγγραφή ΣΠΣ με το ίδιο όνομα. Η λειτουργία είναι διαθέσιμη μόνο σε αναβαθμισμένους χρήστες και το διαχειριστή της εφαρμογής.
- **Αλλαγή ΣΠΣ:** Με τη λειτουργία αυτή αλλάζουν τα πεδία μίας υπάρχουσας εγγραφής ΣΠΣ. Η ανανέωση προϋποθέτει την εισαγωγή του κωδικού της εγγραφής. Γίνεται έλεγχος αν όντως υπάρχει εγγραφή με αυτό τον κωδικό. Όλα τα πεδία πρέπει να έχουν τιμές και δεν πρέπει να υπάρχει εγγραφή ΣΠΣ με το ίδιο όνομα. Η λειτουργία είναι διαθέσιμη μόνο σε αναβαθμισμένους χρήστες και το διαχειριστή της εφαρμογής.
- **Διαγραφή ΣΠΣ:** Με τη λειτουργία αυτή διαγράφεται μία εγγραφή ΣΠΣ. Η διαγραφή προϋποθέτει την εισαγωγή του κωδικού της εγγραφής. Γίνεται έλεγχος αν όντως υπάρχει εγγραφή με αυτό τον κωδικό. Πριν την οριστική διαγραφή ζητείται επιβεβαίωση από το χρήστη. Η λειτουργία είναι διαθέσιμη μόνο σε αναβαθμισμένους χρήστες και το διαχειριστή της εφαρμογής.
- **Εισαγωγή ανταλλακτικού:** Με τη λειτουργία αυτή μία νέα εγγραφή ανταλλακτικού εισάγεται στο σύστημα. Πρέπει να συμπληρωθεί το όνομα του ανταλλακτικού, ο κωδικός NSN του, και η ποσότητα που απαιτείται να υπάρχει ως απόθεμα στο σύστημα. Επιπλέον ορίζονται τα μηχανήματα και τα ΣΠΣ στα οποία μπορεί να χρησιμοποιηθεί. Το όνομα ο κωδικός NSN και η ποσότητα πρέπει απαραίτητα να συμπληρωθούν και δεν πρέπει να υπάρχει εγγραφή ανταλλακτικού με το ίδιο όνομα. Συσχετίσεις με ΣΠΣ και μηχανήματα μπορούν να προστεθούν και σε επόμενο στάδιο. Η λειτουργία είναι διαθέσιμη μόνο σε αναβαθμισμένους χρήστες και το διαχειριστή της εφαρμογής.
- **Αλλαγή ανταλλακτικού:** Με τη λειτουργία αυτή αλλάζουν τα πεδία μίας υπάρχουσας εγγραφής ανταλλακτικού. Η ανανέωση προϋποθέτει την εισαγωγή του κωδικού της εγγραφής. Γίνεται έλεγχος αν όντως υπάρχει εγγραφή με αυτό τον κωδικό. Όνομα ο κωδικός NSN και η ποσότητα πρέπει απαραίτητα να έχουν τιμές και δεν πρέπει να υπάρχει εγγραφή ανταλλακτικού με το ίδιο όνομα. Με αυτή τη λειτουργία μπορούν να προστεθούν ή να διαγραφούν συσχετίσεις με μηχανήματα και ΣΠΣ. Η λειτουργία είναι διαθέσιμη μόνο σε αναβαθμισμένους χρήστες και το διαχειριστή της εφαρμογής.
- **Διαγραφή ανταλλακτικού:** Με τη λειτουργία αυτή διαγράφεται μία εγγραφή ανταλλακτικού. Η διαγραφή προϋποθέτει την εισαγωγή του κωδικού της εγγραφής. Γίνεται έλεγχος αν όντως υπάρχει εγγραφή με αυτό τον κωδικό. Πριν την οριστική διαγραφή ζητείται επιβεβαίωση από το χρήστη. Η λειτουργία είναι διαθέσιμη μόνο σε αναβαθμισμένους χρήστες και το διαχειριστή της εφαρμογής.
- **Εισαγωγή ιστορικού:** Με τη λειτουργία αυτή μία νέα εγγραφή ιστορικού εισάγεται στο σύστημα. Πρέπει να συμπληρωθεί η περιγραφή του ιστορικού, η ημερομηνία εκτέλεσης, το μηχάνημα στο οποίο εκτελέστηκε και το σχετικό ΣΠΣ, αν αφορούσε προγραμματισμένη

συντήρηση. Όλα τα πεδία, εκτός του ΣΠΣ, πρέπει να συμπληρωθούν. Η λειτουργία είναι διαθέσιμη σε όλους τους χρήστες της εφαρμογής.

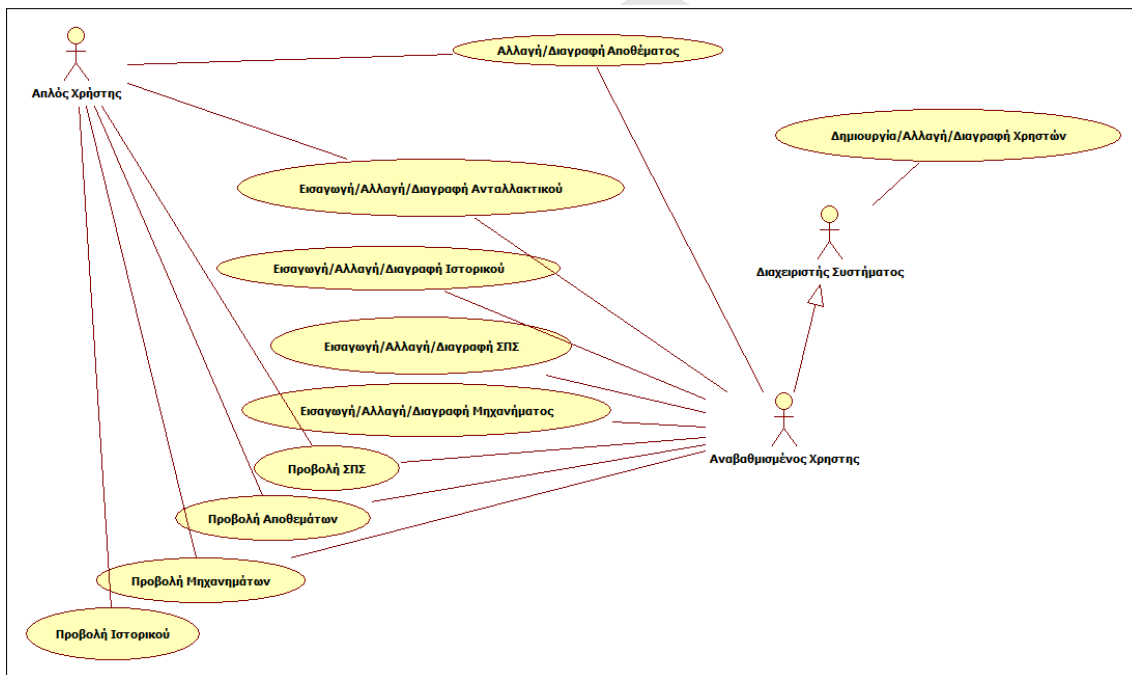
- **Αλλαγή ιστορικού:** Με τη λειτουργία αυτή αλλάζουν τα πεδία μίας υπάρχουσας εγγραφής ιστορικού. Η ανανέωση προϋποθέτει την εισαγωγή του κωδικού της εγγραφής. Γίνεται έλεγχος αν όντως υπάρχει εγγραφή με αυτό τον κωδικό. Όλα τα πεδία, εκτός από το ΣΠΣ που είναι προαιρετικό, πρέπει να έχουν τιμές. Η λειτουργία είναι διαθέσιμη σε όλους τους χρήστες της εφαρμογής.
- **Διαγραφή ιστορικού:** Με τη λειτουργία αυτή διαγράφεται μία εγγραφή ιστορικού. Η διαγραφή προϋποθέτει την εισαγωγή του κωδικού της εγγραφής. Γίνεται έλεγχος αν όντως υπάρχει εγγραφή με αυτό τον κωδικό. Πριν την οριστική διαγραφή ζητείται επιβεβαίωση από το χρήστη. Η λειτουργία είναι διαθέσιμη σε όλους τους χρήστες της εφαρμογής.
- **Αλλαγή αποθέματος ανταλλακτικού:** Με τη λειτουργία αυτή αλλάζει το τρέχον απόθεμα ενός ανταλλακτικού. Η ανανέωση προϋποθέτει την εισαγωγή του NSN κωδικού του αποθέματος. Γίνεται έλεγχος αν όντως υπάρχει απόθεμα με αυτό το NSN. Η λειτουργία είναι διαθέσιμη σε όλους τους χρήστες της εφαρμογής.
- **Διαγραφή αποθέματος ανταλλακτικού:** Με τη λειτουργία αυτή διαγράφεται μία εγγραφή αποθέματος ανταλλακτικού προϋποθέτει την εισαγωγή του NSN κωδικού του αποθέματος. Γίνεται έλεγχος αν όντως υπάρχει απόθεμα με αυτό το NSN. Πριν την οριστική διαγραφή ζητείται επιβεβαίωση από το χρήστη. Η λειτουργία είναι διαθέσιμη σε όλους τους χρήστες της εφαρμογής.
- **Προβολή Μηχανημάτων:** Με τη λειτουργία αυτή προβάλλονται όλες οι εγγραφές μηχανημάτων σε πίνακα (grid). Η λειτουργία είναι διαθέσιμη σε όλους τους χρήστες της εφαρμογής.
- **Προβολή ΣΠΣ:** Με τη λειτουργία αυτή προβάλλονται όλες οι εγγραφές ΣΠΣ σε πίνακα (grid). Η λειτουργία είναι διαθέσιμη σε όλους τους χρήστες της εφαρμογής.
- **Προβολή ανταλλακτικών:** Με τη λειτουργία αυτή προβάλλονται όλες οι εγγραφές ανταλλακτικών σε πίνακα (grid). Η λειτουργία είναι διαθέσιμη σε όλους τους χρήστες της εφαρμογής.
- **Προβολή εγγραφών ιστορικού:** Με τη λειτουργία αυτή προβάλλονται όλες οι εγγραφές ιστορικού σε πίνακα (grid). Η λειτουργία είναι διαθέσιμη σε όλους τους χρήστες της εφαρμογής.
- **Προβολή αποθεμάτων ανταλλακτικών:** Με τη λειτουργία αυτή προβάλλονται όλες οι εγγραφές αποθεμάτων ανταλλακτικών σε πίνακα (grid). Η λειτουργία είναι διαθέσιμη σε όλους τους χρήστες της εφαρμογής.
- **Δημιουργία χρήστη:** Με τη λειτουργία αυτή ο διαχειριστής της εφαρμογής μπορεί να προσθέσει έναν νέο χρήστη. Ορίζονται το όνομα, το επώνυμο, το όνομα χρήστη(username), ένας κωδικός(password), η διεύθυνση ηλεκτρονικού ταχυδρομείου(email) και το επίπεδο πρόσβασης. Η λειτουργία είναι διαθέσιμη μόνο στο διαχειριστή της εφαρμογής.
- **Αλλαγή πληροφοριών χρήστη:** Με τη λειτουργία αυτή οι απλοί και αναβαθμισμένοι χρήστες μπορούν να αλλάξουν τα προσωπικά τους στοιχεία (όνομα, επώνυμο, κωδικός, διεύθυνση ηλεκτρονικού ταχυδρομείου). Ο διαχειριστής της εφαρμογής μπορεί να αλλάξει τα προσωπικά στοιχεία και το επίπεδο πρόσβασης όλων των χρηστών της εφαρμογής.
- **Διαγραφή χρήστη:** Με τη λειτουργία αυτή ο διαχειριστής της εφαρμογής μπορεί να διαγράψει έναν από τους χρήστες της εφαρμογής. Η λειτουργία είναι διαθέσιμη μόνο στο διαχειριστή της εφαρμογής.

3.3 Φάση σύλληψης (ανάλυση)

Κατά τη φάση αυτή έγινε μία αρχική καταγραφή των απαιτήσεων της εφαρμογής. Οι απαιτήσεις καταγράφηκαν και πήραν τη μορφή λειτουργιών – περιπτώσεων χρήσης (Use Cases) όπως παρουσιάστηκαν στις προηγούμενες παραγράφους. Κατά τη διαδικασία πρώτης ανάλυσης χρησιμοποιήθηκαν κάποια πρώτα διαγράμματα περιπτώσεων χρήσης και κλάσεων ώστε να δημιουργηθεί ένα πρώτο σκαρίφημα του έργου. Στο παρόν υποκεφάλαιο θα παρουσιαστούν και θα αναλυθούν συνοπτικά αυτά τα διαγράμματα.

3.3.1 Διαγράμματα περιπτώσεων χρήσης (Use Case Diagrams)

Τα διαγράμματα περιπτώσεων χρήσης (Use Case Diagrams) χρησιμοποιούνται για τη μοντελοποίηση του πλαισίου λειτουργίας της εφαρμογής και την καταγραφή των προδιαγραφών της. Περιλαμβάνουν τις λειτουργίες/περιπτώσεις χρήσης, τους χρήστες του συστήματος καθώς επίσης και πιθανές σχέσεις γενίκευσης. Το διάγραμμα περιπτώσεων χρήσης που παρουσιάζεται στην επόμενη εικόνα, και αφορά την εφαρμογή που αναπτύχθηκε στα πλαίσια της παρούσας εργασίας, περιλαμβάνει όλους τους διαθέσιμους ρόλους χρηστών. Επιπλέον περιλαμβάνει τις διαθέσιμες περιπτώσεις χρήσης ανά τύπο χρήστη. Αξίζει να σημειωθεί πως ο ρόλος του διαχειριστή συστήματος περιγράφεται ως επέκταση του ρόλου του αναβαθμισμένου χρήστη έχοντας επιπλέον πρόσβαση στη λειτουργία διαχείρισης χρηστών. Ο διαχειριστής του συστήματος έχει πρόσβαση σε όλες τις λειτουργίες των αναβαθμισμένων χρηστών επιπλέον της διαχείρισής τους.

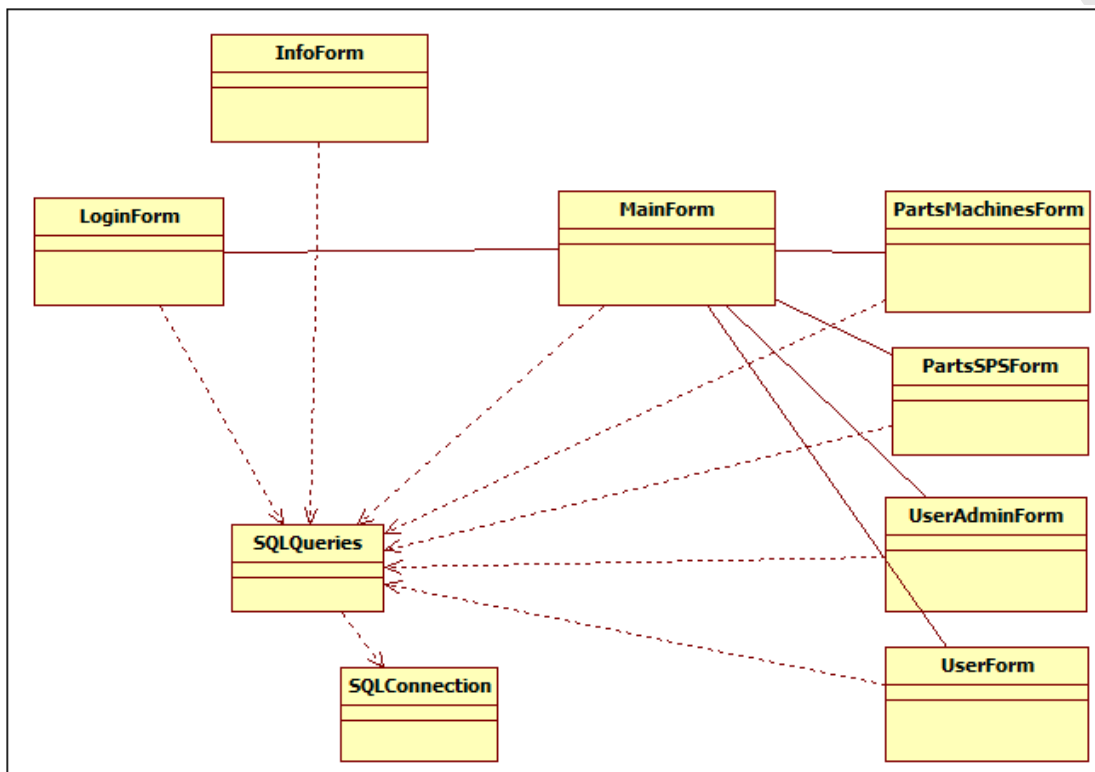


Εικόνα 3-1: Αρχικό διάγραμμα περιπτώσεων χρήσης της εφαρμογής.

3.3.2 Διαγράμματα τάξεων (Class Diagrams)

Στην πρώτη φάση υλοποίησης του έργου σχεδιάστηκε ένα διάγραμμα τάξεων, το οποίο παρουσιάζει ένα πρώτο σχέδιο των κλάσεων που θα χρησιμοποιηθούν κατά την ανάπτυξη της εφαρμογής και των συσχετίσεων που υπάρχουν μεταξύ τους. Καθώς η εφαρμογή λειτουργεί ως ένα front-end (παραθυρικό περιβάλλον πρόσβασης) βάσης δεδομένων, οι τάξεις που περιγράφονται είναι κυρίως τάξεις που

περιγράφουν οθόνες εκτέλεσης (φόρμες – forms) της εφαρμογής. Επιπλέον υπάρχουν βοηθητικές τάξεις για πρόσβαση στη βάση δεδομένων.



Εικόνα 3-2: Αρχικό διάγραμμα τάξεων της εφαρμογής.

Πιο συγκεκριμένα:

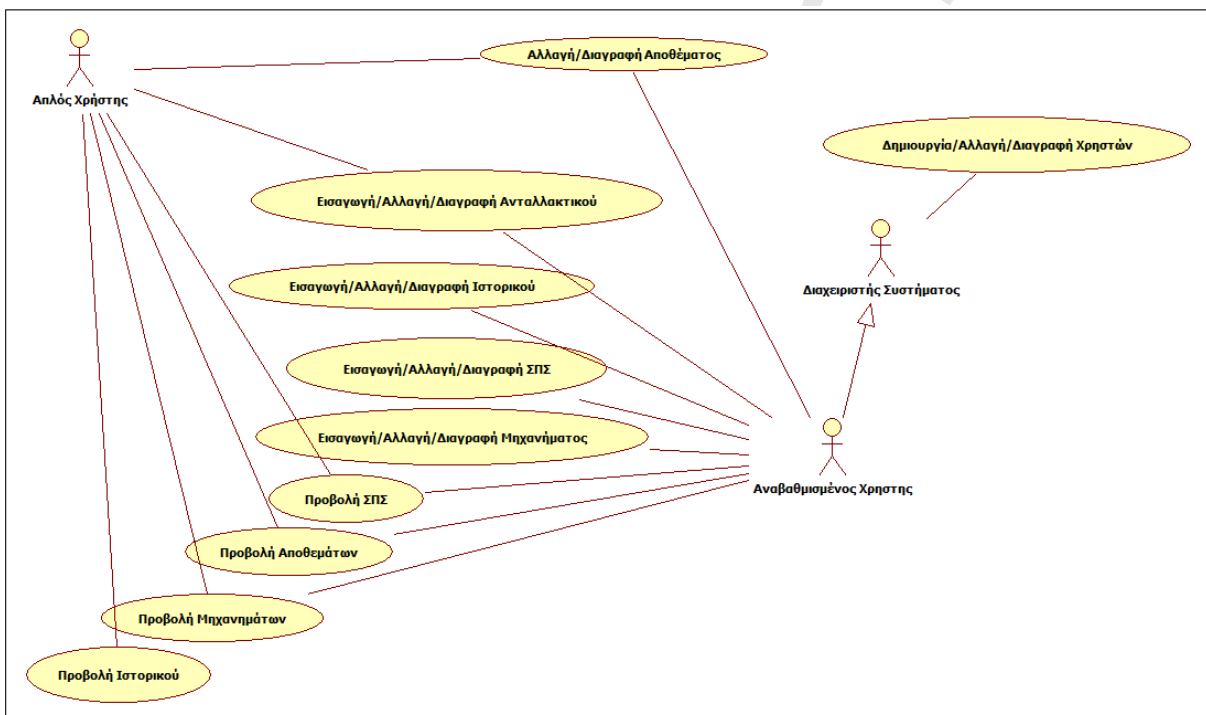
- η τάξη LoginForm περιγράφει την οθόνη εισόδου της εφαρμογής
- η τάξη MainForm περιγράφει την κεντρική σελίδα της εφαρμογής και δημιουργείται από την LoginForm
- η τάξη PartsSPSForm περιγράφει τη φόρμα αντιστοίχισης ανταλλακτικού με ΣΠΣ και δημιουργείται από την MainForm
- η τάξη PartsMachinesForm περιγράφει τη φόρμα αντιστοίχισης ανταλλακτικού με μηχάνημα και δημιουργείται από την MainForm
- η τάξη UserForm περιγράφει την οθόνη αλλαγής στοιχείων κάθε χρήστη και δημιουργείται από την MainForm
- η τάξη UserAdminForm περιγράφει την οθόνη διαχείρισης χρηστών του διαχειριστή της εφαρμογής και δημιουργείται από την MainForm
- η τάξη InfoForm περιγράφει την οθόνη πληροφοριών της εφαρμογής και δημιουργείται από την MainForm
- η τάξη SQLQueries περιγράφει τη βοηθητική τάξη που υλοποιεί ερωτήματα προς τη βάση. Χρησιμοποιείται από όλες τις προαναφερθείσες κλάσεις
- η τάξη SQLConnection περιγράφει τη βοηθητική τάξη που διατηρεί τη σύνδεση με τη βάση και χρησιμοποιείται από την τάξη SQLQueries

3.4 Φάση κλιμάκωσης (εκπόνηση μελέτης)

Στη φάση κλιμάκωσης (εκπόνησης μελέτης) σχεδιάστηκε το μεγαλύτερο ποσοστό των περιπτώσεων χρήσης και καταστρώθηκε το σχέδιο ανάπτυξης της εφαρμογής. Στα πλαίσια της ανάλυσης των απαιτήσεων και του σχεδιασμού χρησιμοποιήθηκαν UML διαγράμματα που περιγράφουν παραμέτρους του συστήματος. Στο παρόν υποκεφάλαιο παρουσιάζονται αυτά τα διαγράμματα και περιγράφεται η σημασία τους.

3.4.1 Διαγράμματα περιπτώσεων χρήσης (Use Case Diagrams)

Κατά τη φάση της ανάλυσης της εφαρμογής δεν ορίστηκαν νέες λειτουργίες ή περιπτώσεις χρήσης. Ως αποτέλεσμα, δεν υπήρξαν αλλαγές στο διάγραμμα περιπτώσεων χρήσης που σχεδιάστηκε κατά τη φάση σύλληψης.

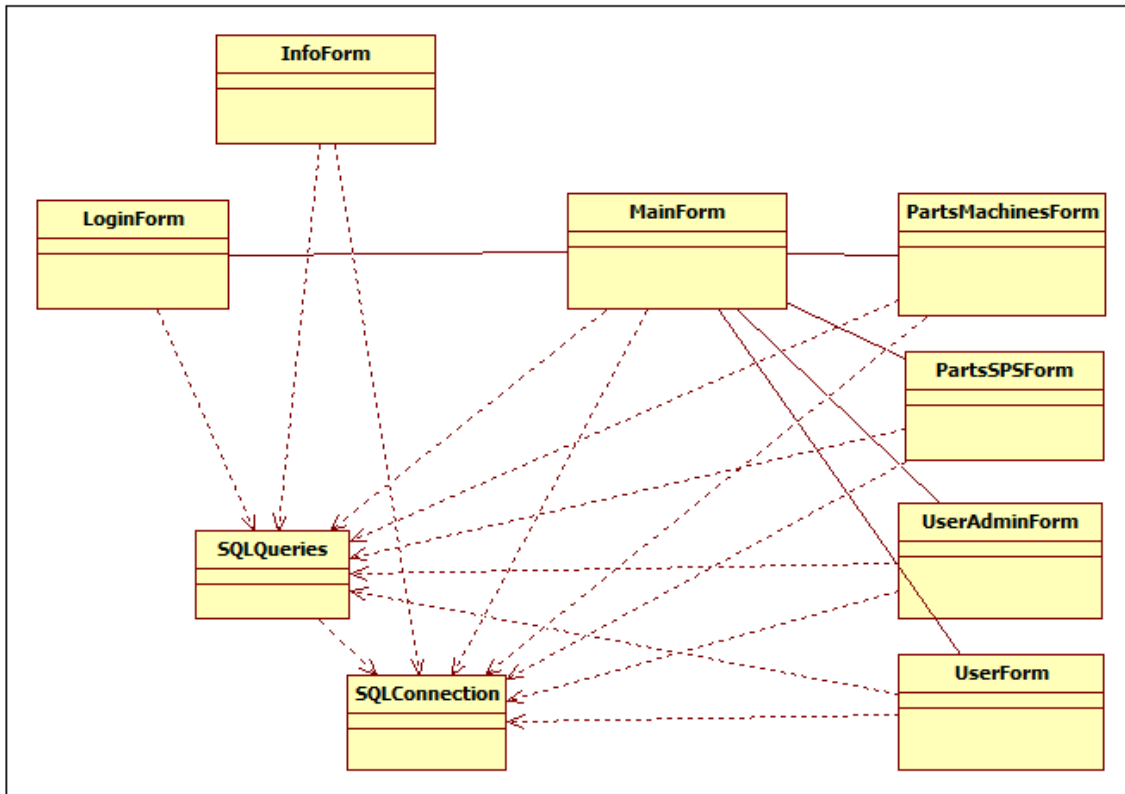


Εικόνα 3-3: Διάγραμμα περιπτώσεων χρήσης κατά τη φάση κλιμάκωσης.

3.4.2 Διαγράμματα τάξεων (Class Diagrams)

Κατά τη φάση κλιμάκωσης διαπιστώθηκε ότι είναι πιθανό οι φόρμες της εφαρμογής να χρησιμοποιήσουν αυτόνομα τη σύνδεση με τη βάση δεδομένων, ώστε να γεμίσουν με δεδομένα κάποιους από τους πίνακές τους, χωρίς μεσολάβηση της τάξης SQLQueries. Ως αποτέλεσμα στο αναθεωρημένο διάγραμμα τάξεων προστέθηκαν σχέσεις εξάρτησης από όλες τις κλάσεις (πλην της LoginForm που θα χρησιμοποιεί έτοιμο ερώτημα της κλάσης SQLQueries) προς την κλάση SQLConnection. Οι σχέσεις εξάρτησης με την κλάση SQLQueries διατηρήθηκαν καθώς έτοιμα ερωτήματα θα χρησιμοποιηθούν σε όλες τις οθόνες της εφαρμογής. Αξίζει να σημειωθεί πως λόγω της φύσης της εφαρμογής, δεν κρίθηκε σκόπιμο να κατασκευαστεί διάγραμμα αντικειμένων. Η εφαρμογή λειτουργεί ως παραθυρικό περιβάλλον πρόσβασης στη βάση δεδομένων οπότε δεν διατηρούνται διαφορετικές καταστάσεις αντικειμένων με ενδιαφέρον καταγραφής. Τα αντικείμενα της εφαρμογής,

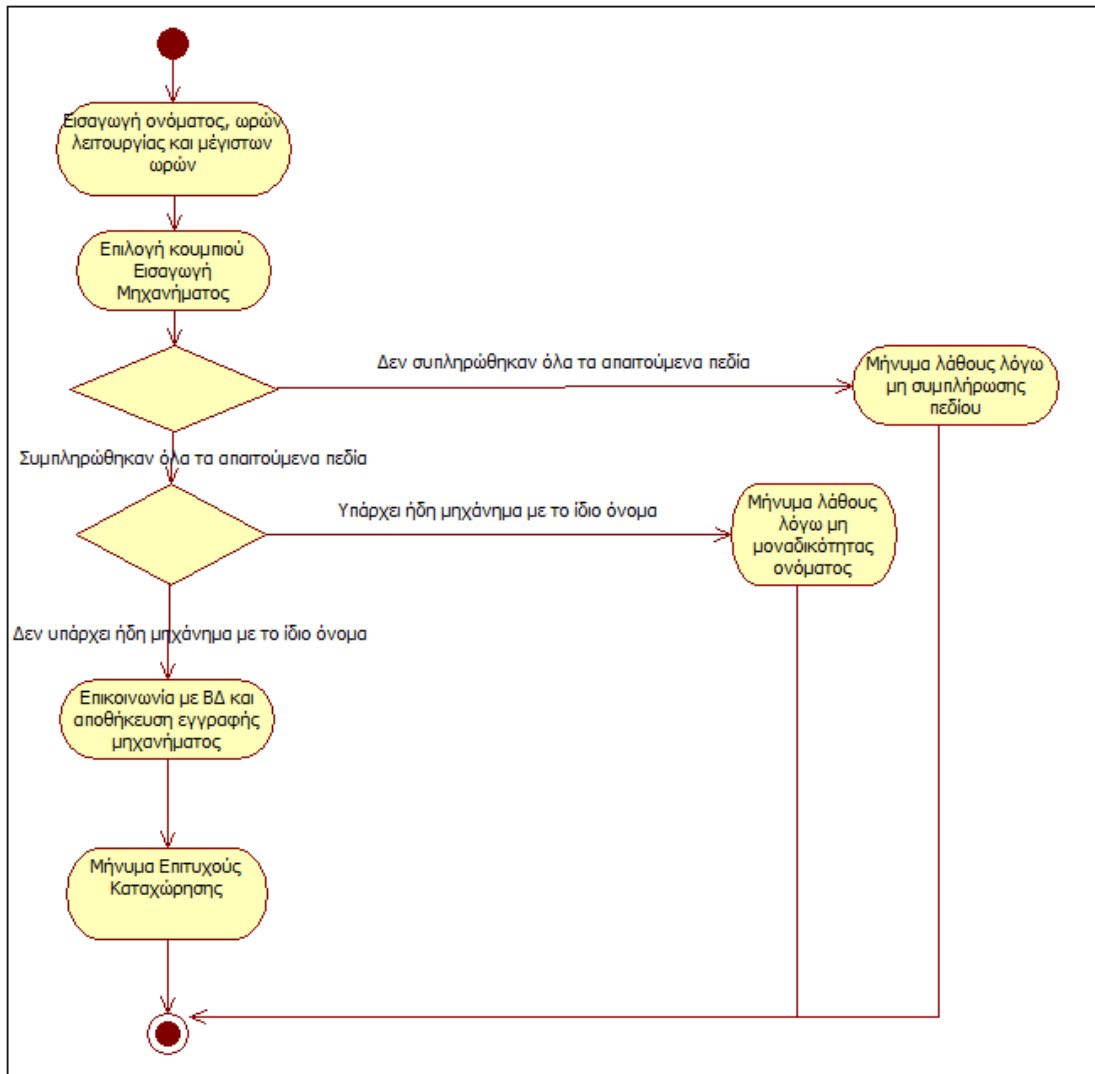
ως εκφάνσεις των αντίστοιχων κλάσεων, είναι στην ουσία οι οθόνες που περιγράφουν κάθε μία από αυτές.



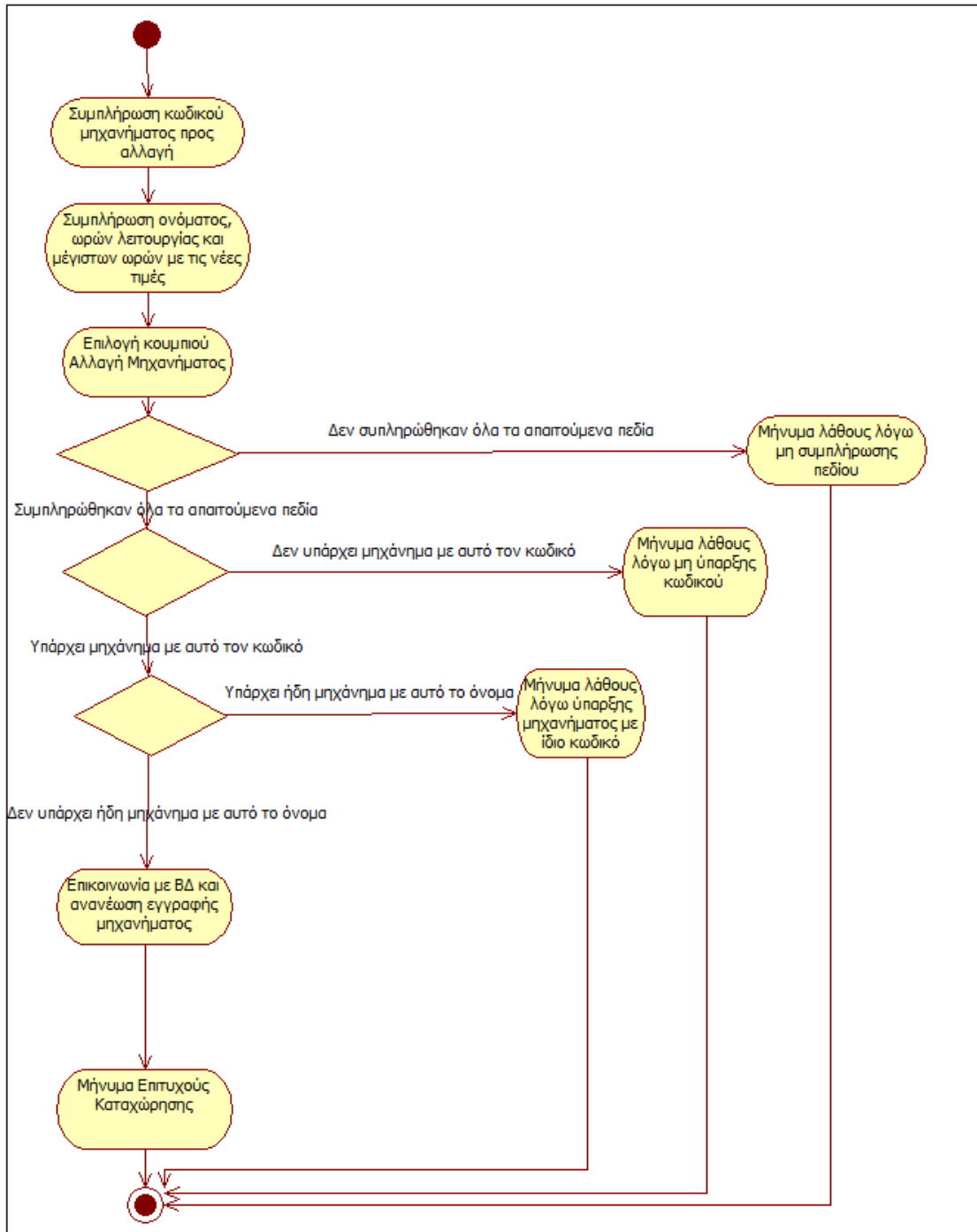
Εικόνα 3-4: Διάγραμμα τάξεων κατά τη φάση κλιμάκωσης.

3.4.3 Διαγράμματα δραστηριοτήτων (Activity Diagrams)

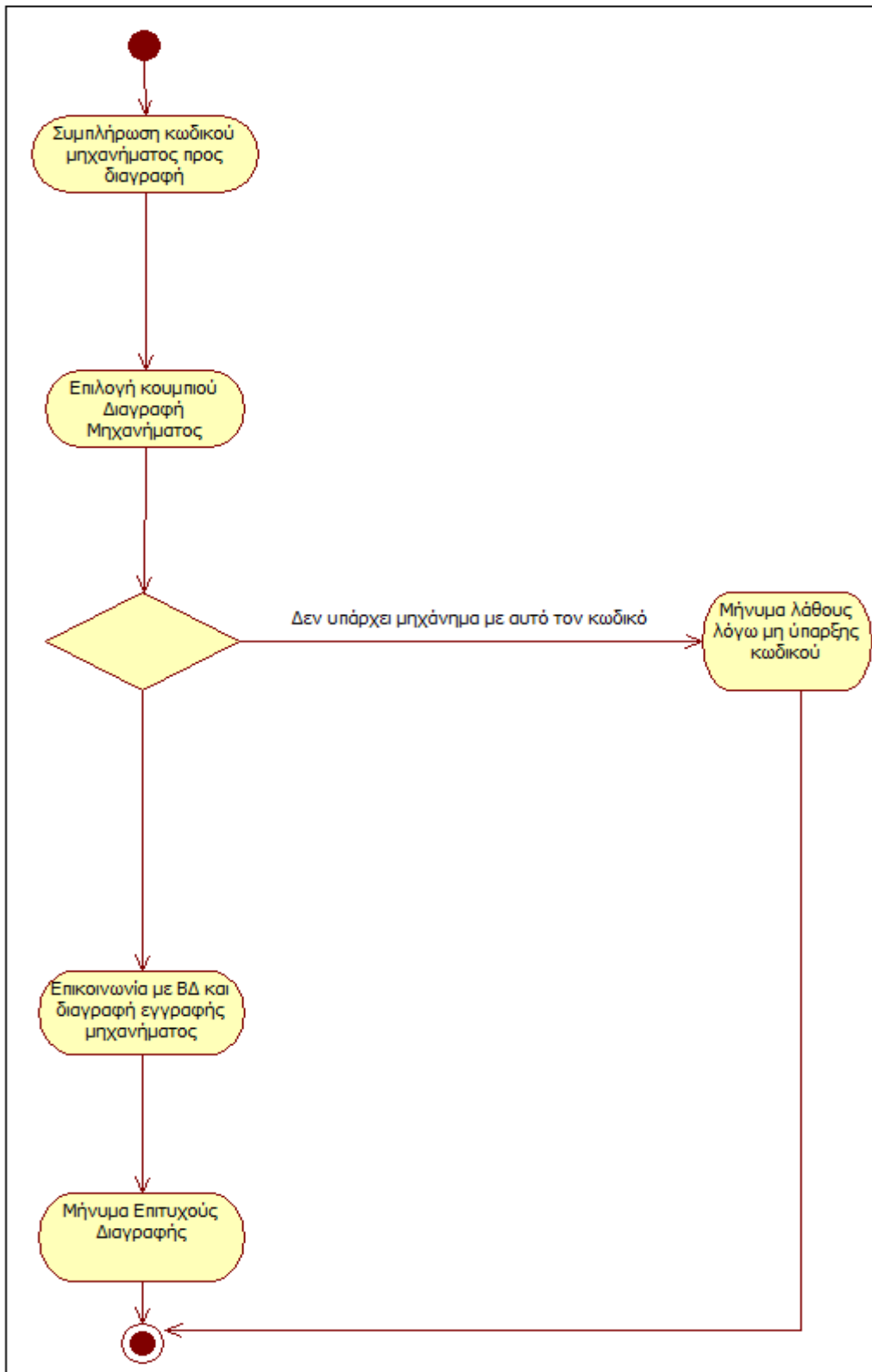
Στη φάση κλιμάκωσης κρίθηκε σκόπιμο να αναλυθεί με μεγαλύτερη λεπτομέρεια η ακολουθία κινήσεων που απαιτείται για την εκτέλεση καθεμιάς από τις λειτουργίες. Για το σκοπό αυτό σχεδιάστηκαν διαγράμματα δραστηριοτήτων (Activity Diagrams), τα οποία σύμφωνα με το μοντέλο UML μπορούν να χρησιμοποιηθούν για την αναπαράσταση της ροής του ελέγχου (workflow) σε ένα υπολογιστικό σύστημα. Λόγω των πολλών λειτουργιών που έχουν οριστεί για το σύστημα (βλ. 3.2.3) κρίθηκε σκόπιμο να σχεδιαστεί ξεχωριστό διάγραμμα δραστηριότητας για κάθε μία από τις πιο σημαντικές. Οι λειτουργίες της προβολής δεδομένων, καθώς απαιτούν την επιλογή ενός μόνο κουμπιού για την εμφάνιση των δεδομένων, δεν αναπαράστηκαν με διαγράμματα δραστηριοτήτων.



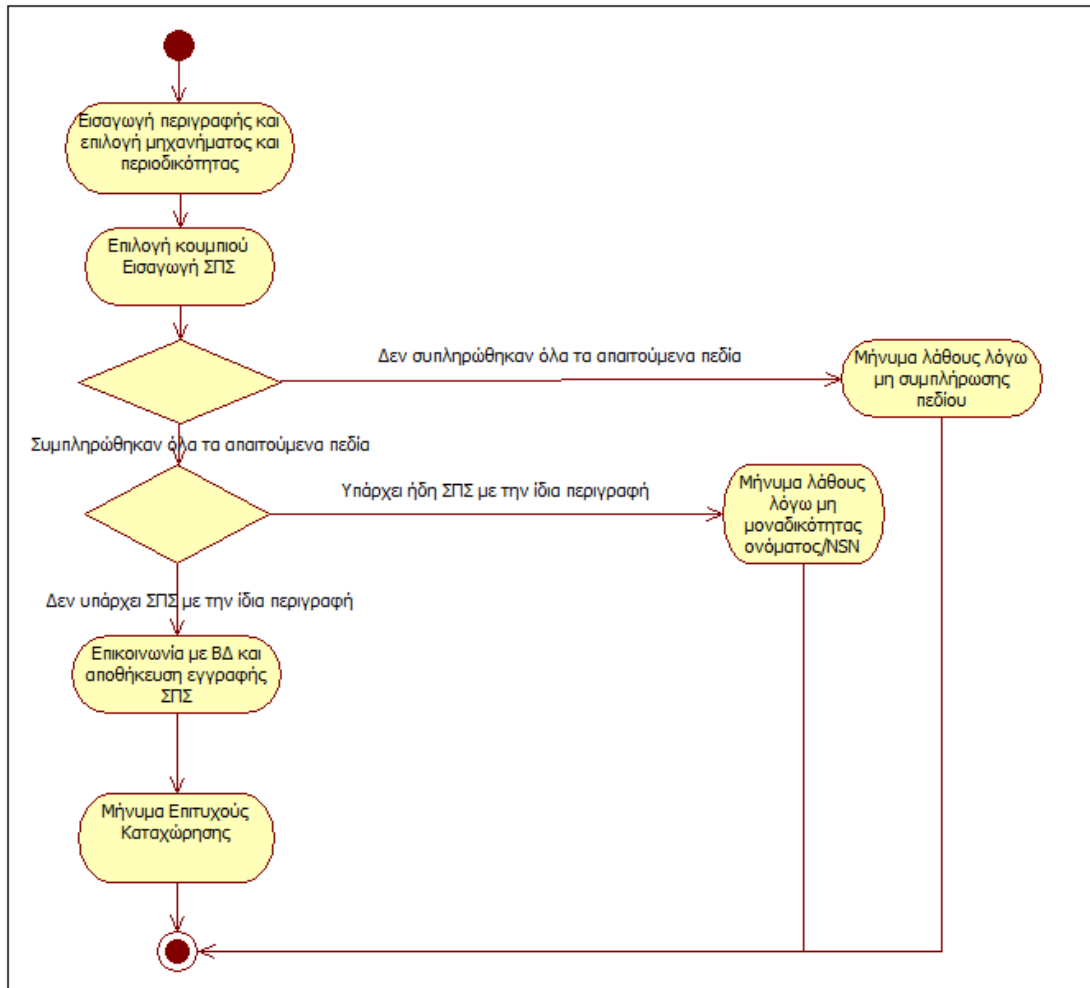
Εικόνα 3-5: Διάγραμμα δραστηριοτήτων της λειτουργίας εισαγωγής νέου μηχανήματος.



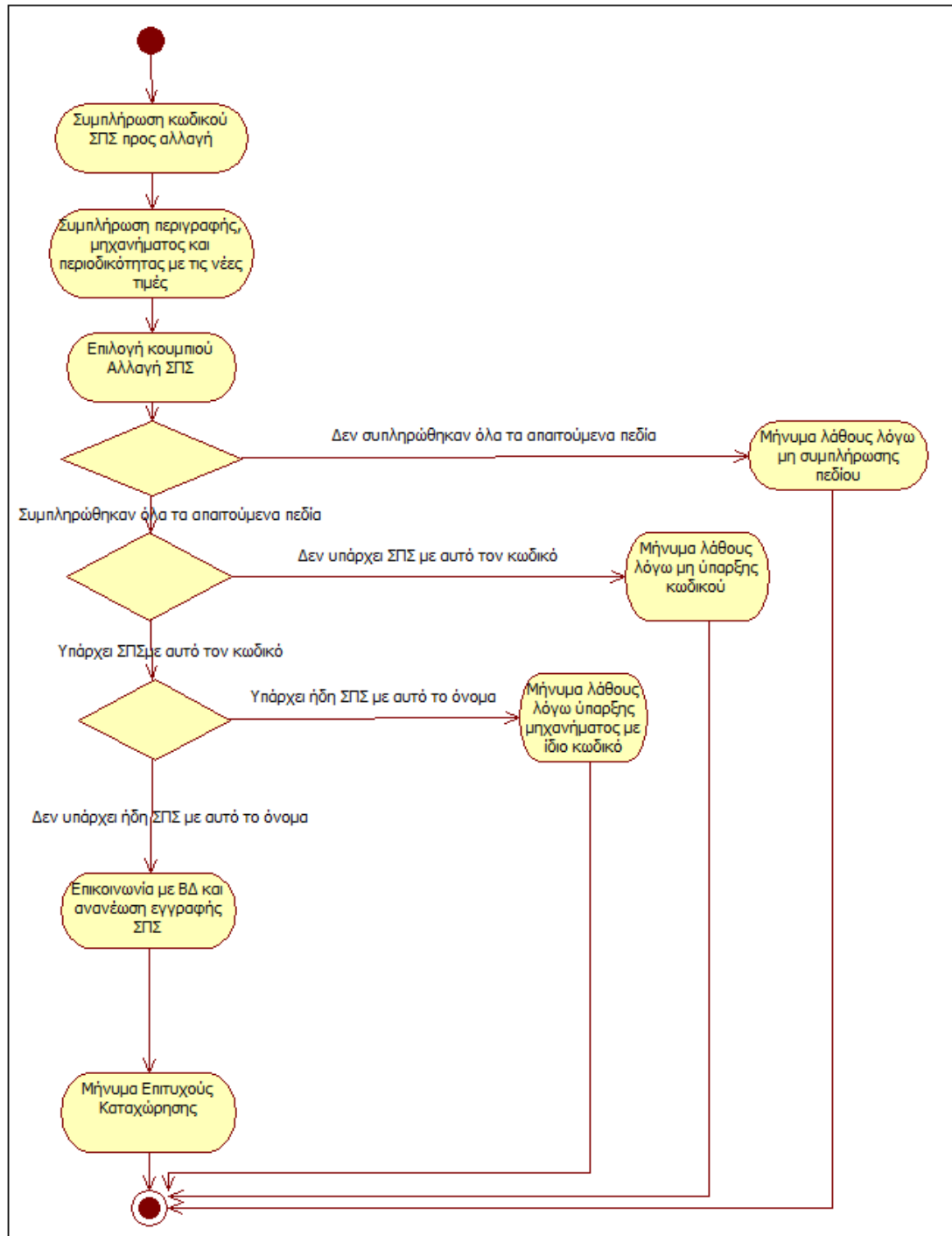
Εικόνα 3-6: Διάγραμμα δραστηριοτήτων της λειτουργίας αλλαγής στοιχείων μηχανήματος.



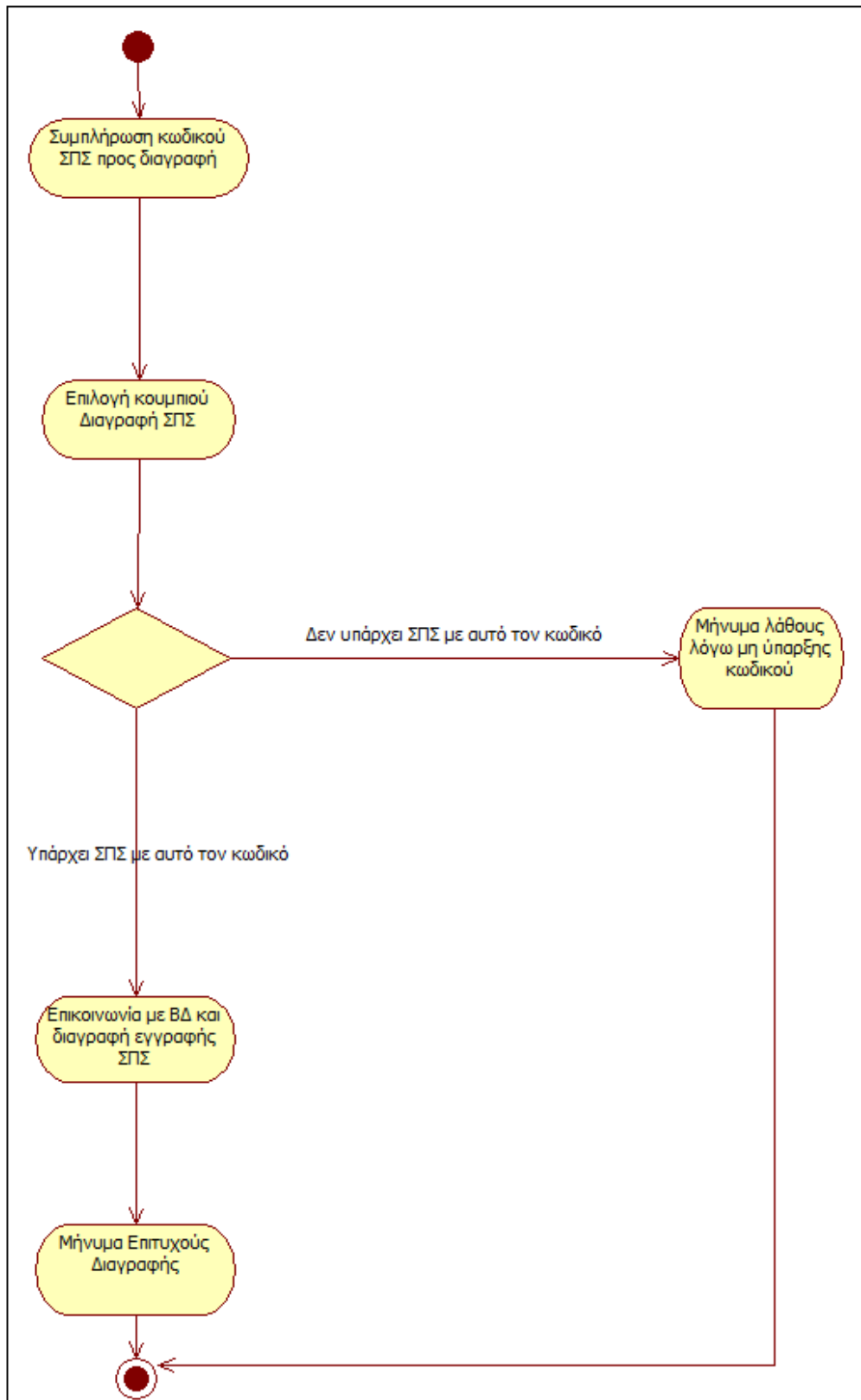
Εικόνα 3-7: Διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής μηχανήματος.



Εικόνα 3-8: Διάγραμμα δραστηριοτήτων της λειτουργίας εισαγωγής νέου ΣΠΣ.

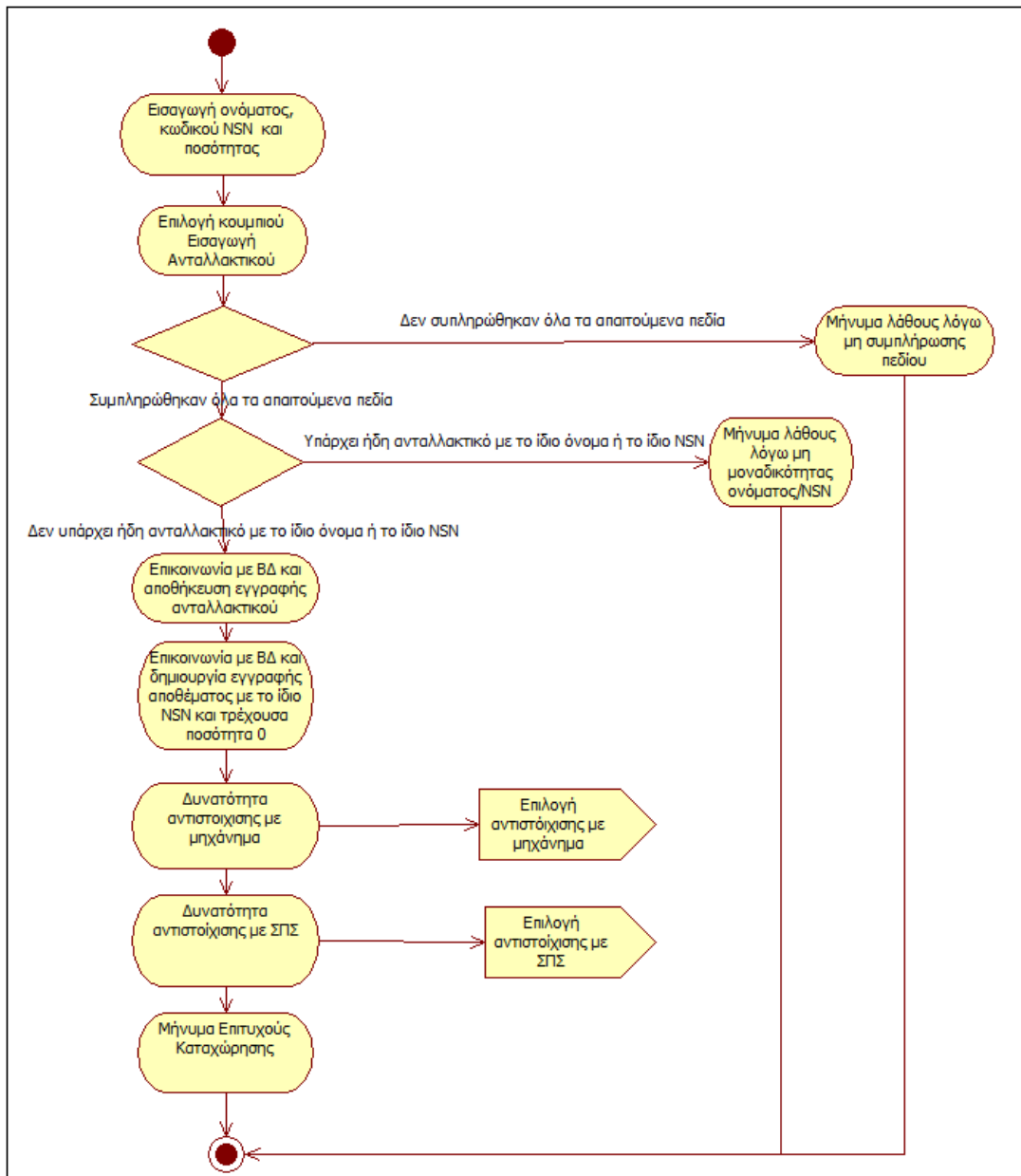


Εικόνα 3-9: Διάγραμμα δραστηριοτήτων της λειτουργίας αλλαγής στοιχείων ΣΠΣ.

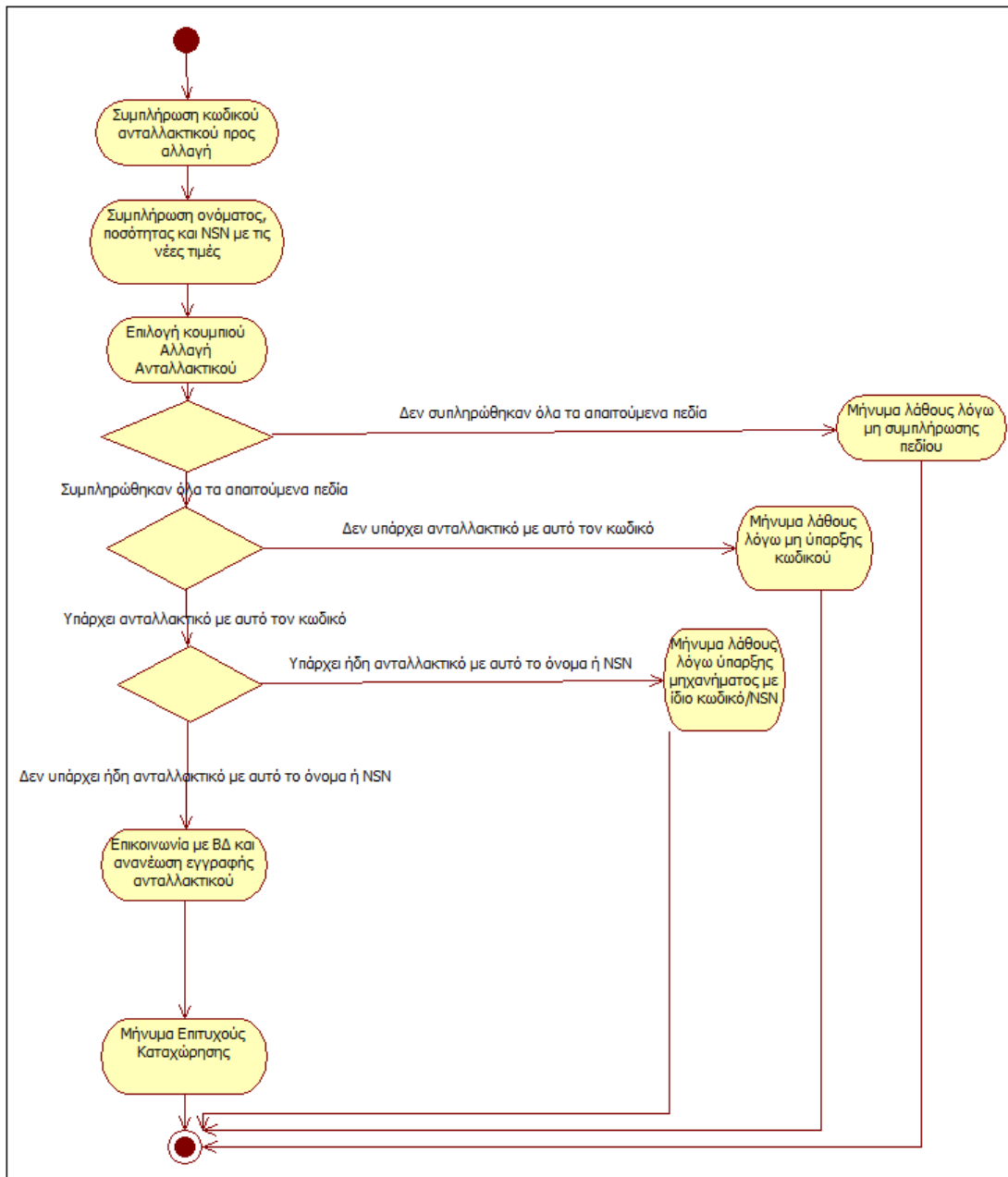


Εικόνα 3-10: Διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής ΣΠΣ.

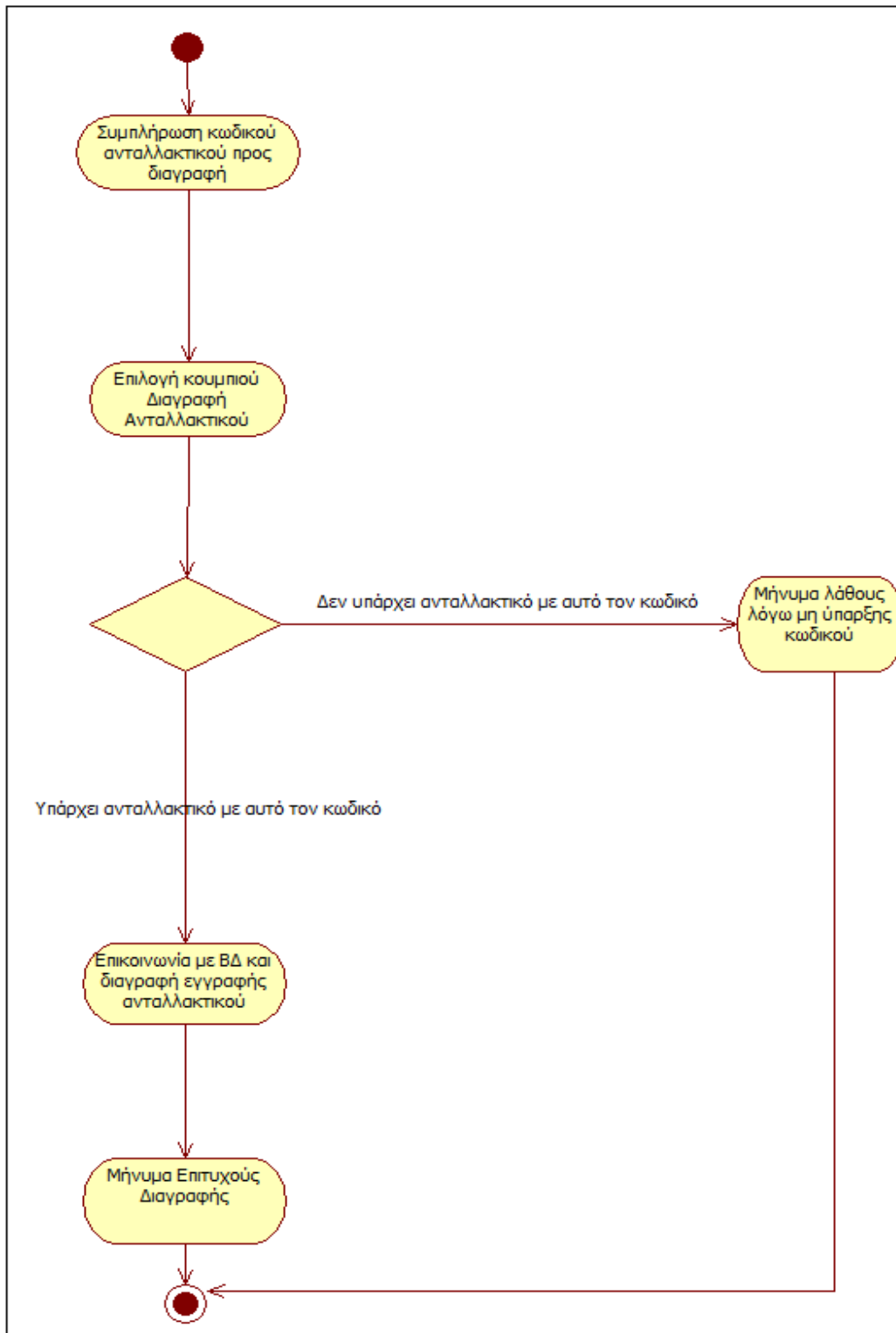
Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP



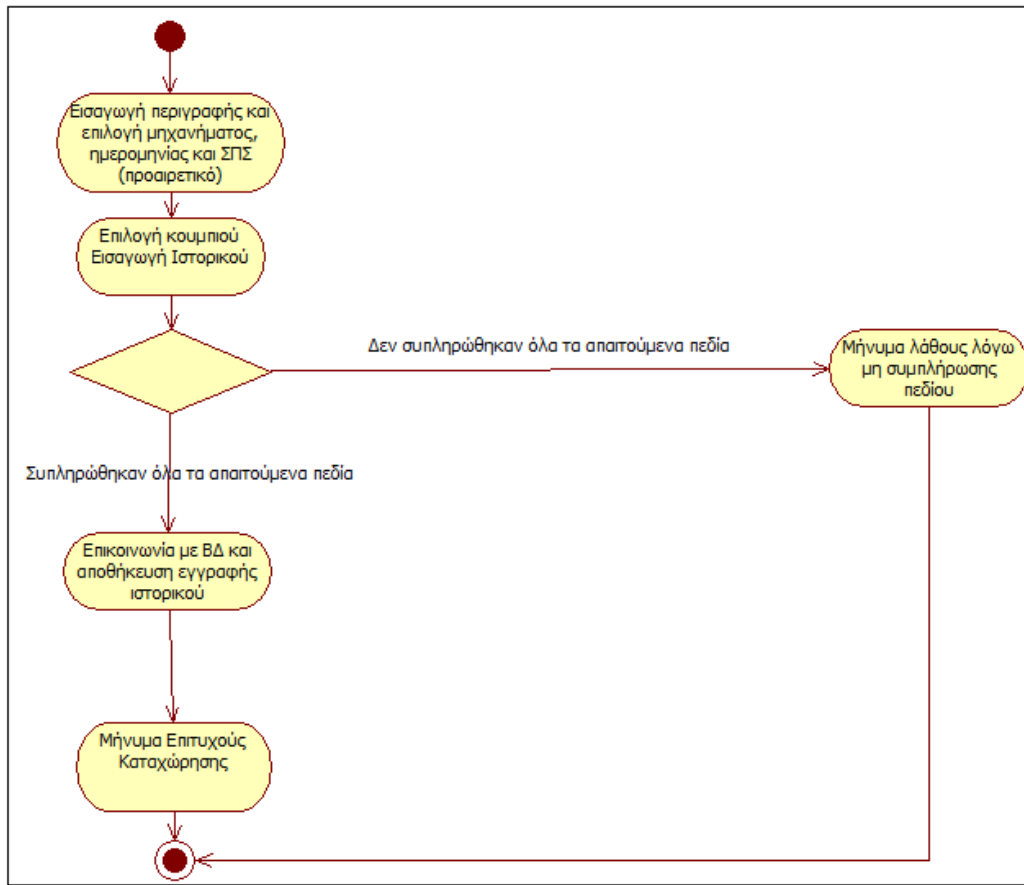
Εικόνα 3-11: Διάγραμμα δραστηριοτήτων της λειτουργίας εισαγωγής νέου ανταλλακτικού.



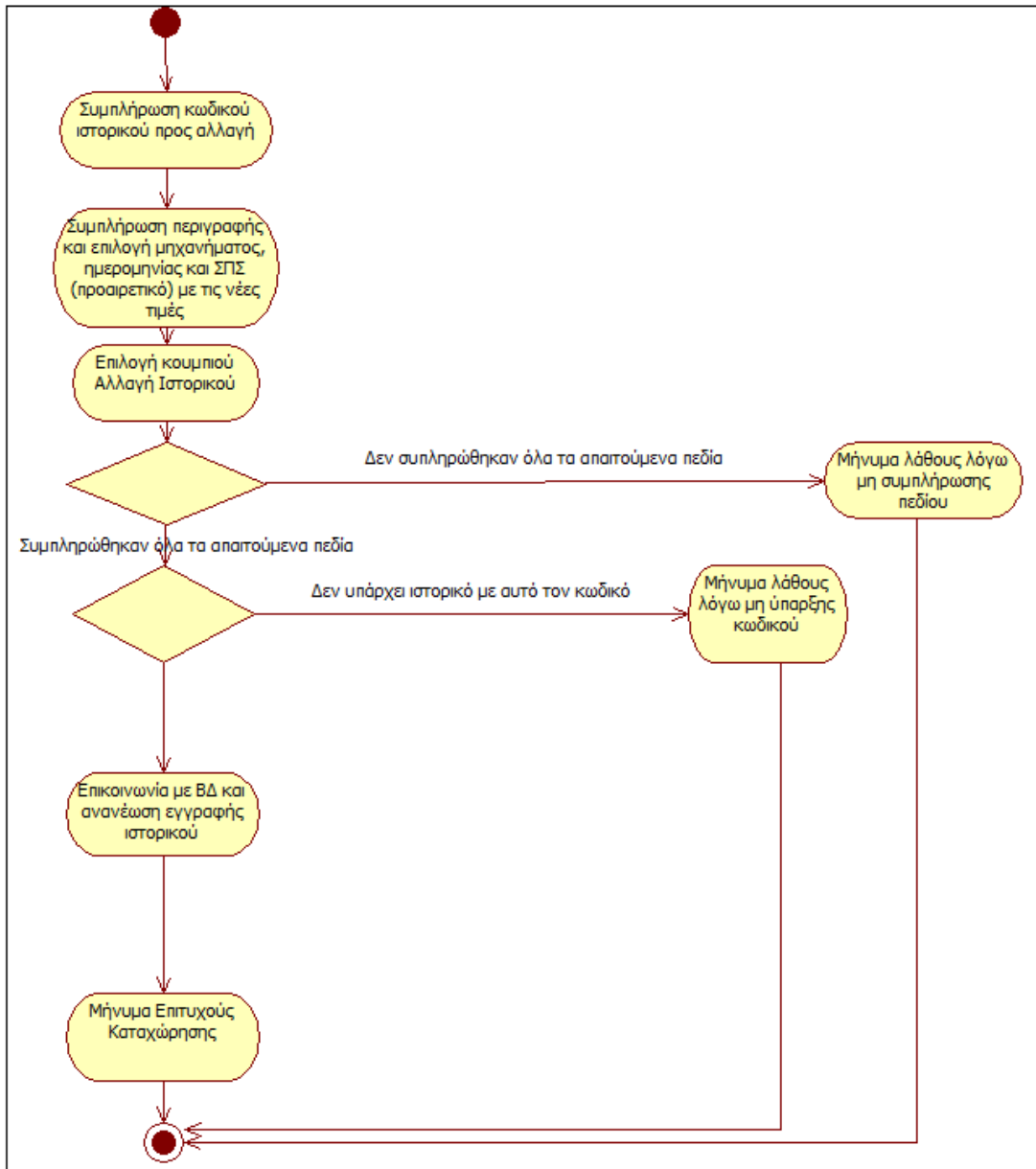
Εικόνα 3-12: Διάγραμμα δραστηριοτήτων της λειτουργίας αλλαγής στοιχείων ανταλλακτικού.



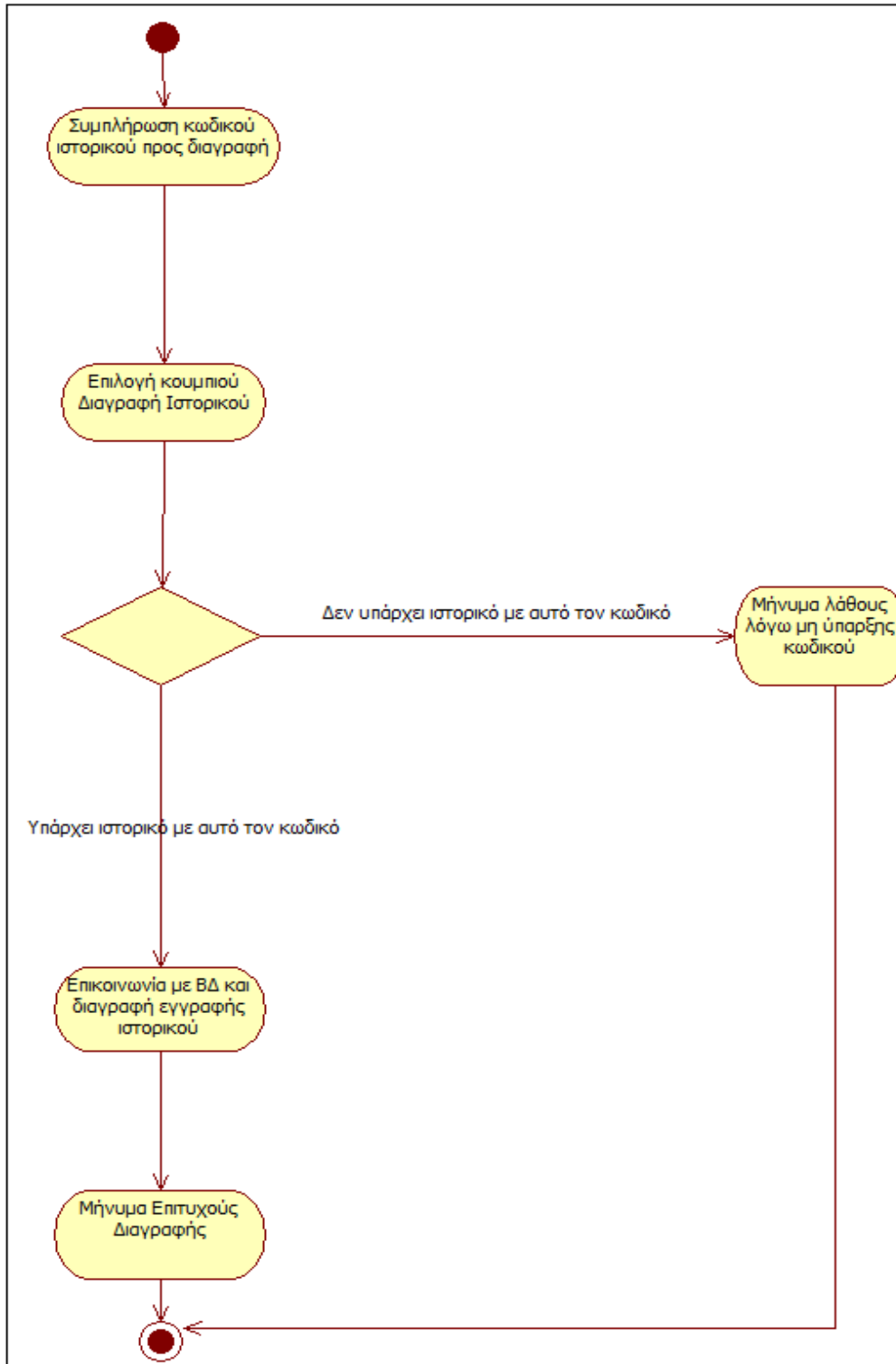
Εικόνα 3-13: Διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής ανταλλακτικού.



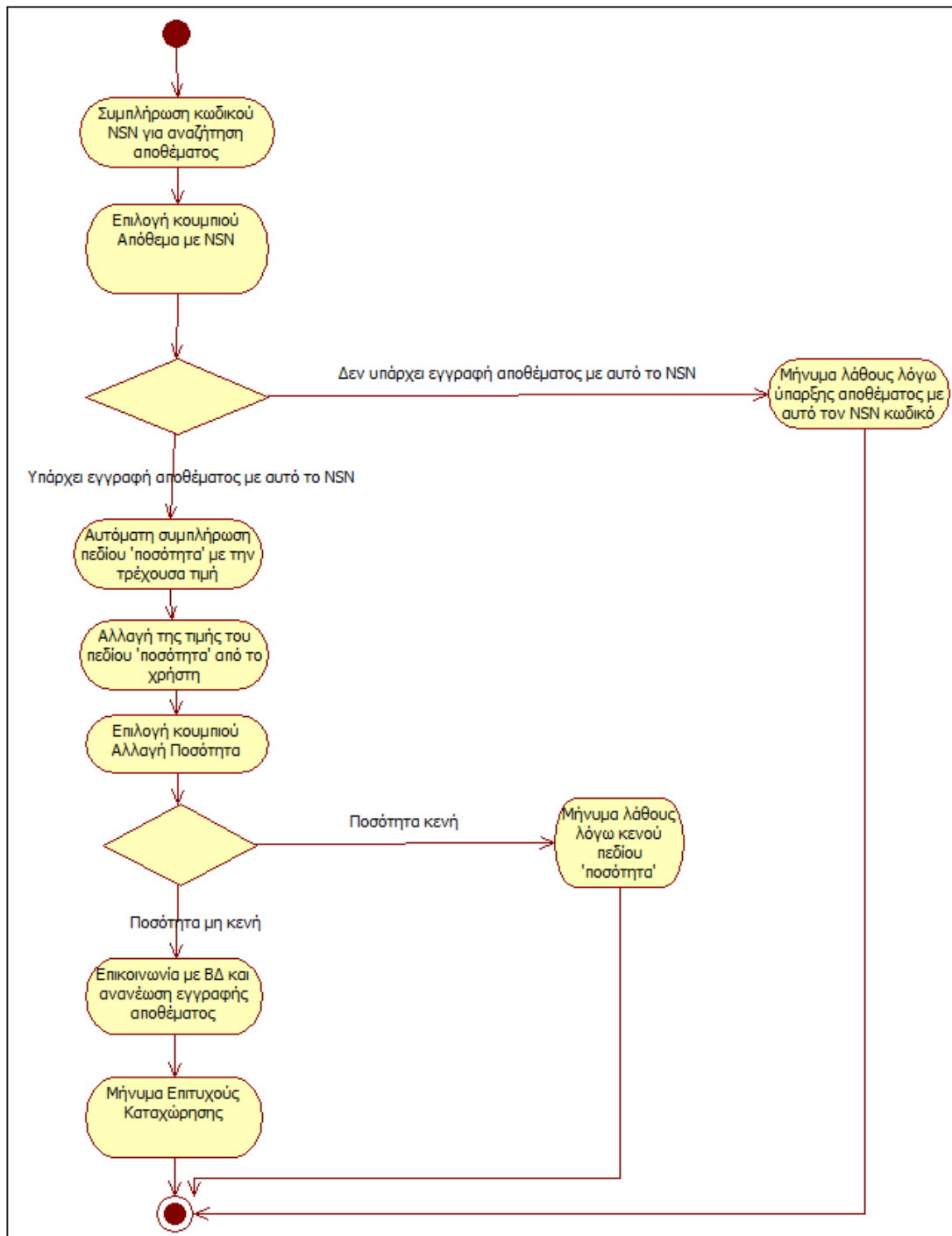
Εικόνα 3-14: Διάγραμμα δραστηριοτήτων της λειτουργίας εισαγωγής ιστορικού συντήρησης.



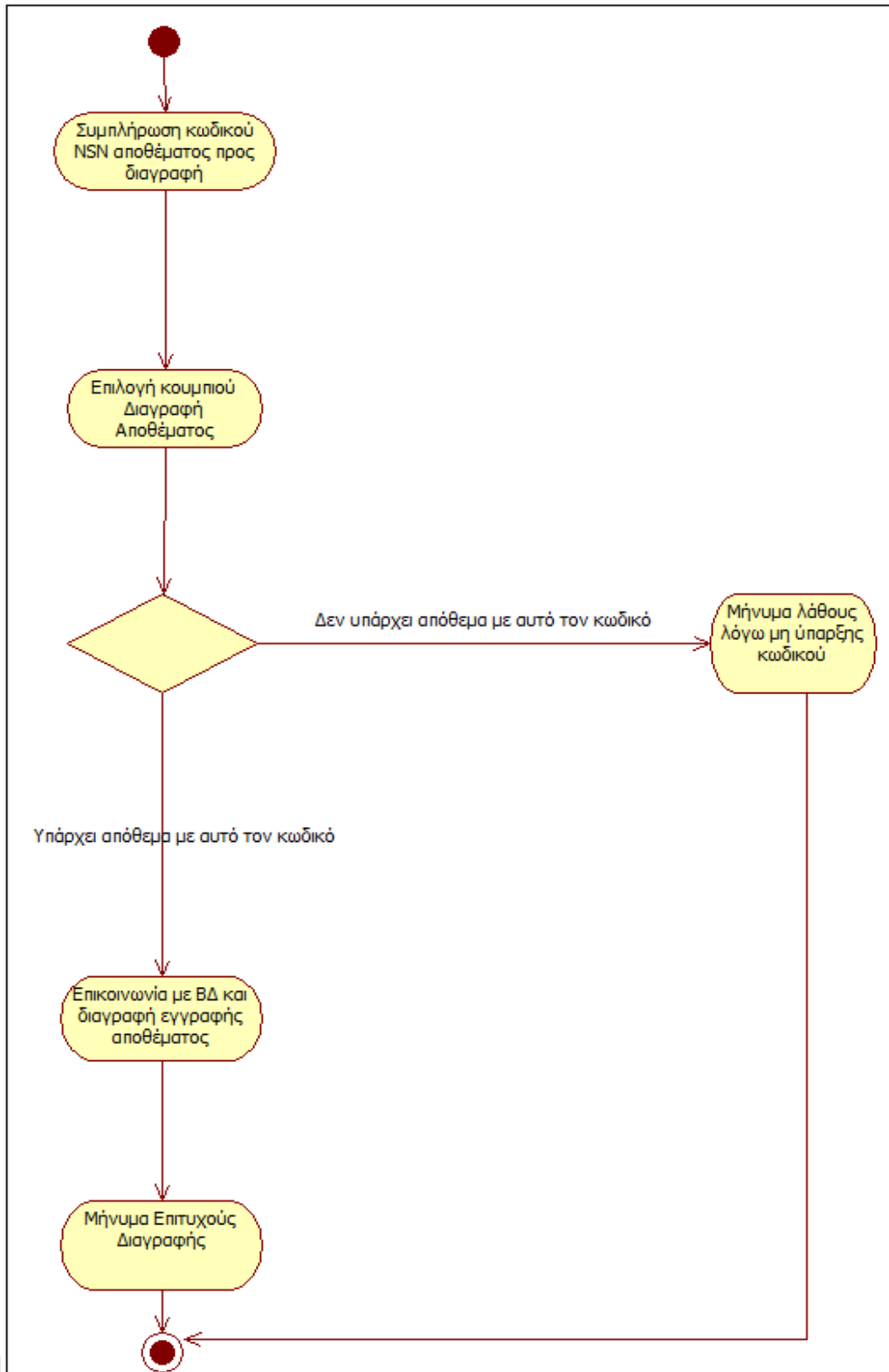
Εικόνα 3-15: Διάγραμμα δραστηριοτήτων της λειτουργίας αλλαγής στοιχείων ιστορικού συντήρησης.



Εικόνα 3-16: Διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής ιστορικού συντήρησης.



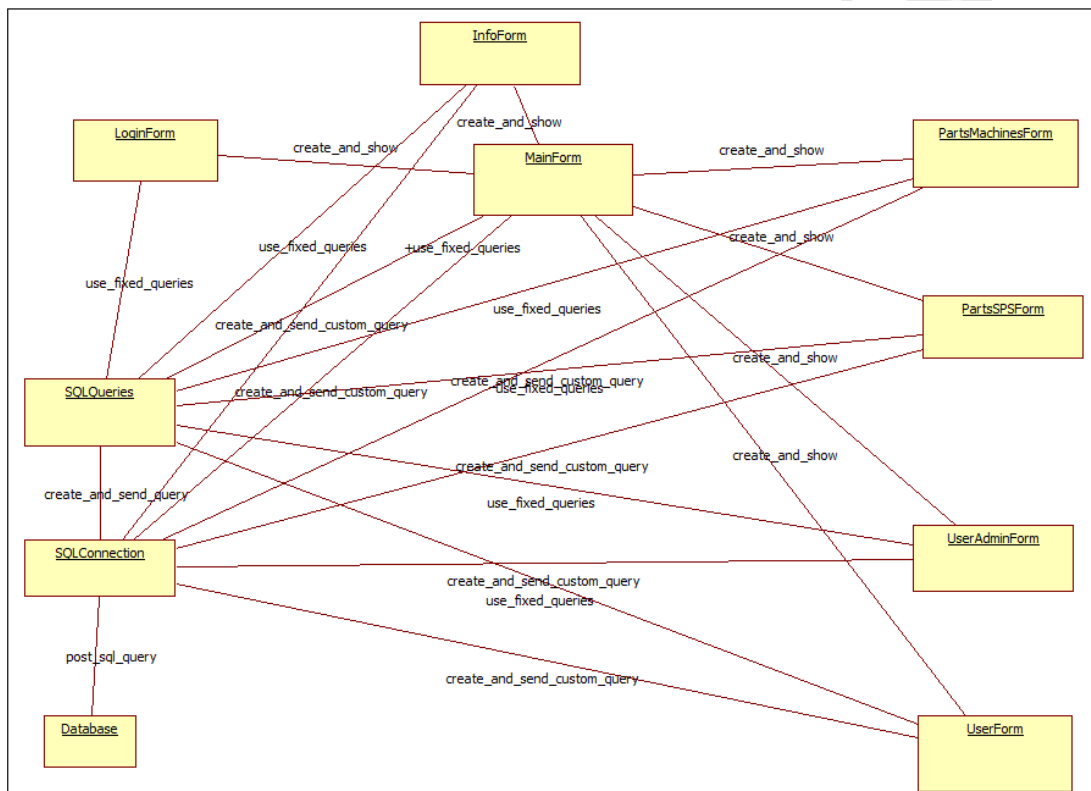
Εικόνα 3-17: Διάγραμμα δραστηριοτήτων της λειτουργίας αλλαγής ποσότητας αποθέματος.



Εικόνα 3-18: Διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής αποθέματος.

3.4.4 Διαγράμματα συνεργασίας (Collaboration Diagrams)

Στα πλαίσια της φάσης κλιμάκωσης του έργου σχηματίστηκε και ένα διάγραμμα συνεργασίας το οποίο αναπαριστά τις αλληλεπιδράσεις μεταξύ των αντικειμένων (φορμών) που απαρτίζουν το λογισμικό που αναπτύχθηκε. Στο διάγραμμα αυτό αναπαριστούνται οι επιμέρους οθόνες και η βάση δεδομένων που διατηρεί τις πληροφορίες που εισάγονται στο σύστημα. Παρατηρείται ότι οι τύποι αλληλεπιδράσεων είναι σχετικά λίγοι καθώς αφορούν είτε τη δημιουργία και εμφάνιση φόρμας από μία άλλη, είτε την υποβολή έτοιμων ή τροποποιημένων ερωτημάτων προς τη βάση δεδομένων. Όπως αναφέρθηκε και προηγουμένως αυτό είναι αναμενόμενο καθώς η εφαρμογή λειτουργεί ως ένα γραφικό περιβάλλον πρόσβασης (frontend) μίας Βάσης Δεδομένων και δεν απαιτεί περίπλοκες αλληλεπιδράσεις από τις κλάσεις που την απαρτίζουν.



Εικόνα 3-19: Διάγραμμα συνεργασίας κατά τη φάση κλιμάκωσης του έργου.

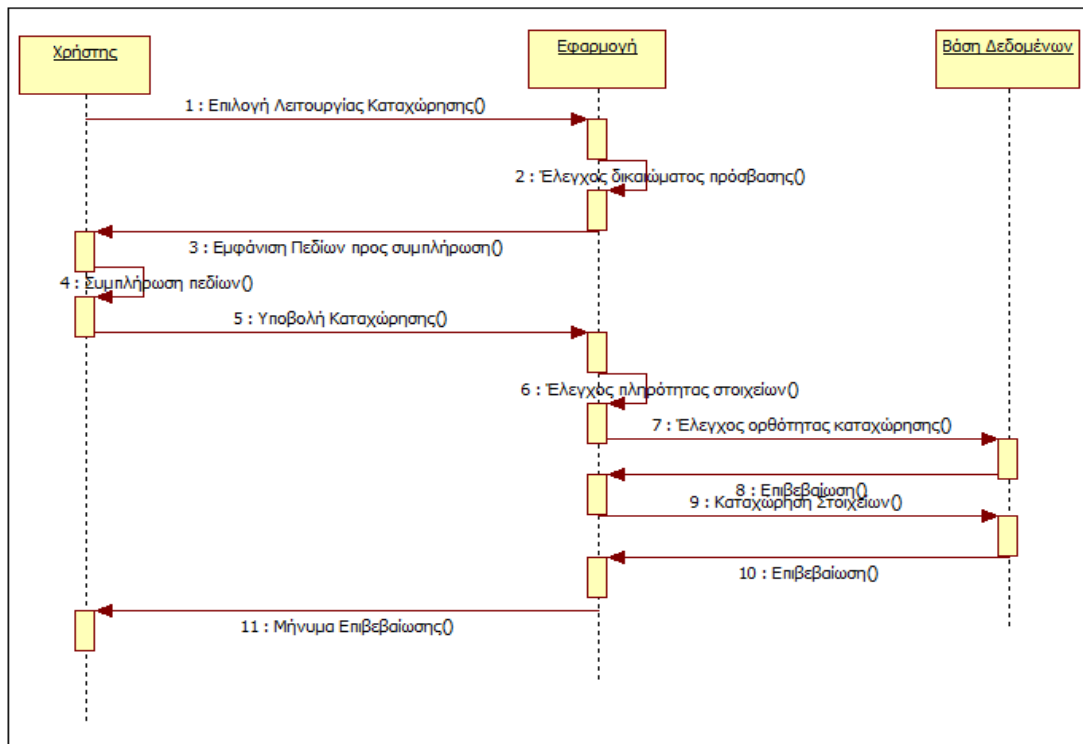
Πιο συγκεκριμένα, για το παραπάνω διάγραμμα:

- οι συνδέσεις του τύπου `create_and_show` αναπαριστούν τη σχέση που υπάρχει μεταξύ μίας οθόνης (κλάσης) η οποία δημιουργεί και εμφανίζει ένα αντικείμενο μίας άλλης κλάσης (οθόνη). Για παράδειγμα η σύνδεση `create_and_show` μεταξύ της `MainForm` και της `LoginForm` υποδεικνύει ότι η οθόνη πληροφοριών κατασκευάζεται και εμφανίζεται μέσω επιλογής που υπάρχει στην κύρια οθόνη της εφαρμογής.
- οι συνδέσεις του τύπου `use_fixed_queries` υποδηλώνουν τη χρήση έτοιμων ερωτημάτων SQL προς τη βάση από οθόνες της εφαρμογής με τη μορφή συναρτήσεων της κλάσης `SQLQueries`. Για παράδειγμα η σύνδεση `use_fixed_queries` μεταξύ της `LoginForm` και της κλάσης `SQLQueries` υποδεικνύει ότι η οθόνη εισόδου χρησιμοποιεί μία έτοιμη συνάρτηση η οποία στέλνει ερώτημα SQL προς τη βάση ώστε να ελέγξει αν οι πληροφορίες εισόδου (credentials) που έδωσε ο χρήστης είναι σωστές. Ως αποτέλεσμα επιτρέπεται μία θετική ή αρνητική απάντηση ως προς το αν ο χρήστης πρέπει να εισέλθει στο σύστημα.

- οι συνδέσεις του τύπου `create_and_send_custom_query` αναπαριστούν τη σχέση που υπάρχει μεταξύ μίας οθόνης (κλάσης) η οποία διαμορφώνει ένα κατά περίπτωση (custom) SQL ερώτημα και το υποβάλει μέσω συνάρτησης της κλάσης `SQLConnection`. Ως αποτέλεσμα επιστρέφεται ένα σύνολο δεδομένων, αν πρόκειται για ερώτημα αναζήτησης ή το αποτέλεσμα επιτυχίας αν πρόκειται για ερώτημα καταχώρησης.
- η σύνδεση του τύπου `post_sql_query` μεταξύ της τάξης `SQLConnection` και του συστήματος διαχείρισης βάσεων δεδομένων, υποδηλώνει την τελική υποβολή του ερωτήματος μέσω της ενδιάμεσης βιβλιοθήκης λογισμικού (dll) με πρωτόκολλο που ορίζεται από αυτή. Μετά την υποβολή του ερωτήματος, το αποτέλεσμα επιστρέφεται στην κλάση που το ζήτησε μέσω της κλάσης `SQLConnection`.

3.4.5 Διαγράμματα σειράς (Sequence Diagrams)

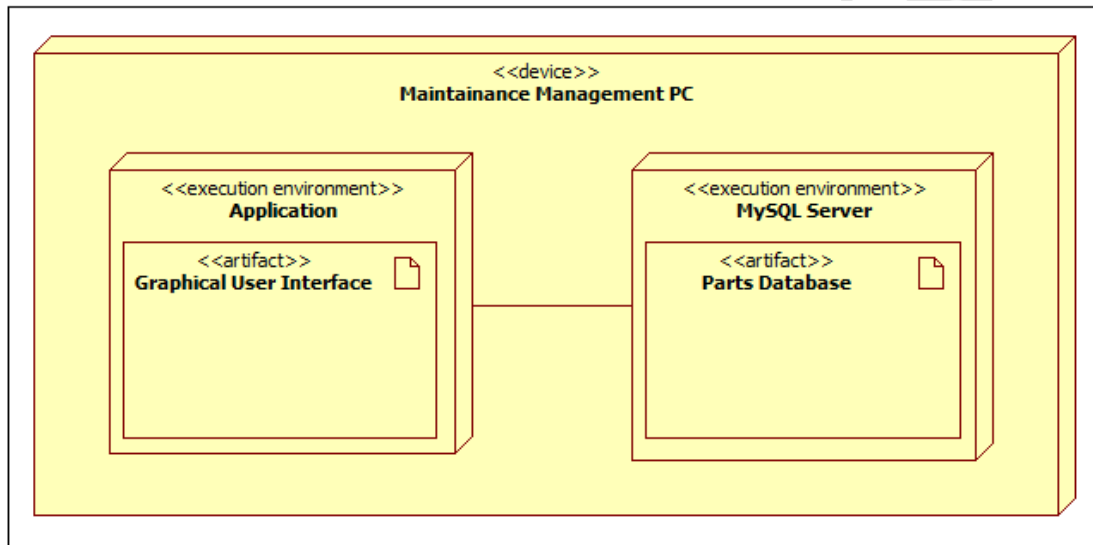
Κατά την πρώτη φάση σχεδίασης του έργου δόθηκε μεγάλη σημασία στον τρόπο επικοινωνίας της εφαρμογής με τη βάση δεδομένων ανάλογα με τις λειτουργίες που εκτελεί ο χρήστης. Θεωρήθηκε σωστό να εκτελούνται προκαταρκτικοί έλεγχοι από την εφαρμογή (για τη συμπλήρωση όλων των απαραίτητων πεδίων), στη συνέχεια επικοινωνία της εφαρμογής με τη βάση για έλεγχο της επιχειρούμενης αλλαγής στα δεδομένα και τελικά η επικοινωνία για την καταχώρηση της αλλαγής. Το μοντέλο αυτό χρησιμοποιήθηκε σε όλες τις λειτουργίες καταχώρησης της εφαρμογής και παρουσιάζεται στο διάγραμμα σειράς (sequence diagram) που ακολουθεί. Το διάγραμμα απεικονίζει μία επιτυχή διαδικασία καταχώρησης. Αν κάποιος από τους ελέγχους δεν επιστρέψει ορθό αποτέλεσμα, η διαδικασία διακόπτεται και επιστρέφεται επεξηγηματικό μήνυμα λάθους στο χρήστη.



Εικόνα 3-20: Διάγραμμα σειράς για επιτυχημένη λειτουργία καταχώρησης.

3.4.6 Διαγράμματα διανομής(Deployment Diagrams)

Όπως αναφέρθηκε και προηγουμένως η εφαρμογή λειτουργεί ως ένα γραφικό περιβάλλον πρόσβασης σε μία βάση δεδομένων. Έχοντας λάβει υπόψη τις συγκεκριμένες προδιαγραφές του έργου σχεδιάστηκε το ακόλουθο διάγραμμα διανομής (Deployment diagram) το οποίο δηλώνει με ποιο τρόπο είναι εγκατεστημένα τα διάφορα μέρη του συστήματος. Όπως φαίνεται στο ίδιο σύστημα υπάρχει η εφαρμογή (Application) και το σύστημα διαχείρισης βάσης δεδομένων (ΣΔΒΔ - MySQL Server). Η εφαρμογή και το ΣΔΒΔ θεωρούνται περιβάλλοντα εκτέλεσης (με βάση το μοντέλο UML) και περιλαμβάνουν τα UML artifacts του γραφικού περιβάλλοντος(Graphical User Interface) και της βάσης δεδομένων(Parts Database) αντίστοιχα. Τα 2 περιβάλλοντα εκτέλεσης επικοινωνούν μαζί τους με τη χρήση ενδιάμεσης βιβλιοθήκης λογισμικού.



Εικόνα 3-21: Το διάγραμμα διανομής του έργου.

3.5 Φάση κατασκευής

Κατά την φάση κατασκευής του έργου, οριστικοποιήθηκαν οι λειτουργίες που θα προσφέρει η εφαρμογή στους χρήστες. Επιπλέον καθορίστηκε με μεγαλύτερη ακρίβεια ο τρόπος υλοποίησης ορισμένων κομματιών του συστήματος. Πολύ σημαντικές προσθήκες ήταν η δυνατότητα αυτόματης συμπλήρωσης τιμών με επιλογή, η ολοκλήρωση ΣΠΣ με αυτόματη καταχώριση ιστορικού και η προβολή των κινήσεων των χρηστών από το διαχειριστή της εφαρμογής.

Κατά τον αρχικό σχεδιασμό της εφαρμογής, ο χρήστης όφειλε να εισάγει μόνος του τους κωδικούς οντοτήτων που ήθελε να αλλάξει, να χειριστεί ή να διαγράψει. Θεωρήθηκε ότι θα ήταν χρήσιμο να συμπληρώνονται αυτόματα αυτοί οι κωδικοί και τα σχετικά πεδία, με επιλογή μίας από τις γραμμές του πίνακα αποτελεσμάτων. Για παράδειγμα, αν ο χρήστης θέλει να αλλάξει ή να διαγράψει κάποιο μηχάνημα, μπορεί να επιλέξει την προβολή όλων των μηχανημάτων, να επιλέξει τη γραμμή του μηχανήματος που τον ενδιαφέρει από τον πίνακα αποτελεσμάτων, να δει να συμπληρώνονται αυτόματα οι κωδικοί και τα πεδία του εν λόγω μηχανήματος και τέλος να επιλέξει αν θέλει να το αλλάξει ή να το διαγράψει. Η λειτουργία αυτή επεκτάθηκε σε όλες τις λειτουργίες καταχώρισης της εφαρμογής και επηρέασε ορισμένα από τα UML διαγράμματα.

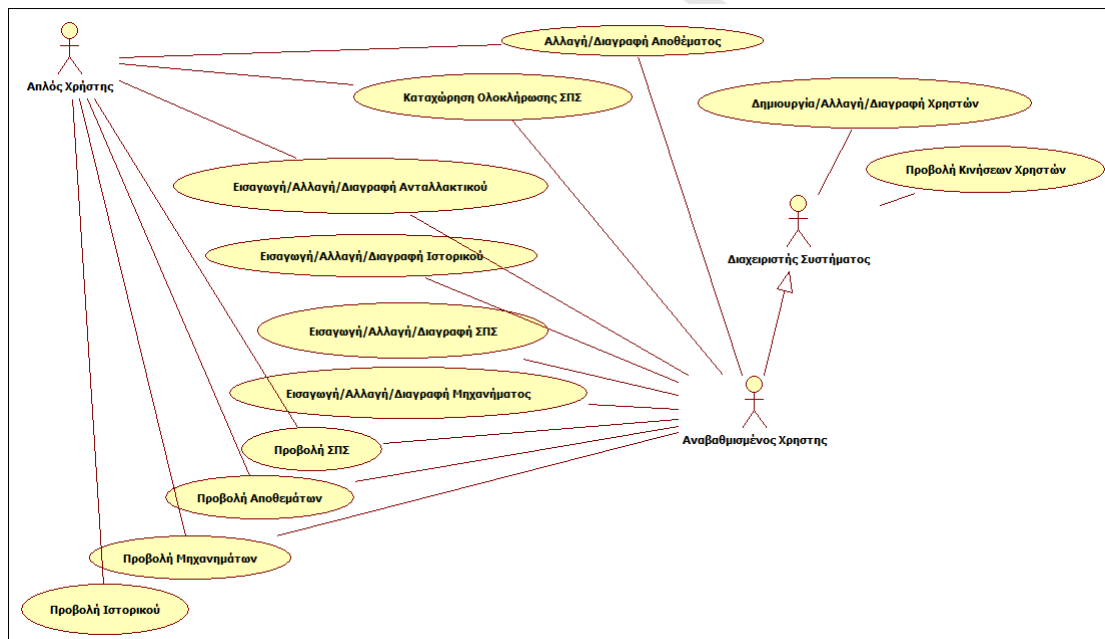
Όσον αφορά την καταχώριση ολοκλήρωσης ΣΠΣ με αυτόματη ενημέρωση ιστορικού, κρίθηκε σκόπιμο να δοθεί η δυνατότητα στους χρήστες της εφαρμογής να επιλέξουν μία από τις εργασίες(ΣΠΣ) ημέρας στην οθόνη πληροφοριών και να καταχωρίσουν την ολοκλήρωσή της. Για το σκοπό αυτό προστέθηκε φόρμα εισαγωγής η οποία περιλαμβάνει τα ανταλλακτικά που χρησιμοποιήθηκαν και την

ημερομηνία εκτέλεσης. Αφού ο χρήστης εισάγει τα στοιχεία της εργασίας συντήρησης που πραγματοποιήθηκε, εισάγεται νέα εγγραφή ιστορικού συντήρησης. Επιπλέον ορίζεται η επόμενη ημερομηνία εκτέλεσης του ΣΠΣ και ανανεώνεται το απόθεμα των ανταλλακτικών μειούμενο κατά αυτά που χρησιμοποιήθηκαν. Τέλος, υπολογίζεται αυτόματα η απαιτούμενη ποσότητα κάθε ανταλλακτικού ώστε να θεωρείται επαρκές. Για το σκοπό αυτό υπολογίζεται σε πόσα ΣΠΣ μέχρι το τέλος του χρόνου συμμετέχει το εν λόγω ανταλλακτικό.

Για το διαχειριστή της εφαρμογής προστέθηκε η δυνατότητα προβολής των κινήσεων καταχώρισης νέου μηχανήματος και νέου ιστορικού από τους χρήστες της εφαρμογής. Ο διαχειριστής επιλέγοντας το σχετικό κουμπί από την κεντρική οθόνη έχει τη δυνατότητα να δει πότε και από ποιόν χρήστη καταχωρήθηκαν νέες εγγραφές μηχανημάτων ή ιστορικού. Αυτό κρίθηκε αναγκαίο για καλύτερο έλεγχο των καταχωρίσεων.

3.5.1 Διαγράμματα περιπτώσεων χρήσης (Use Case Diagrams)

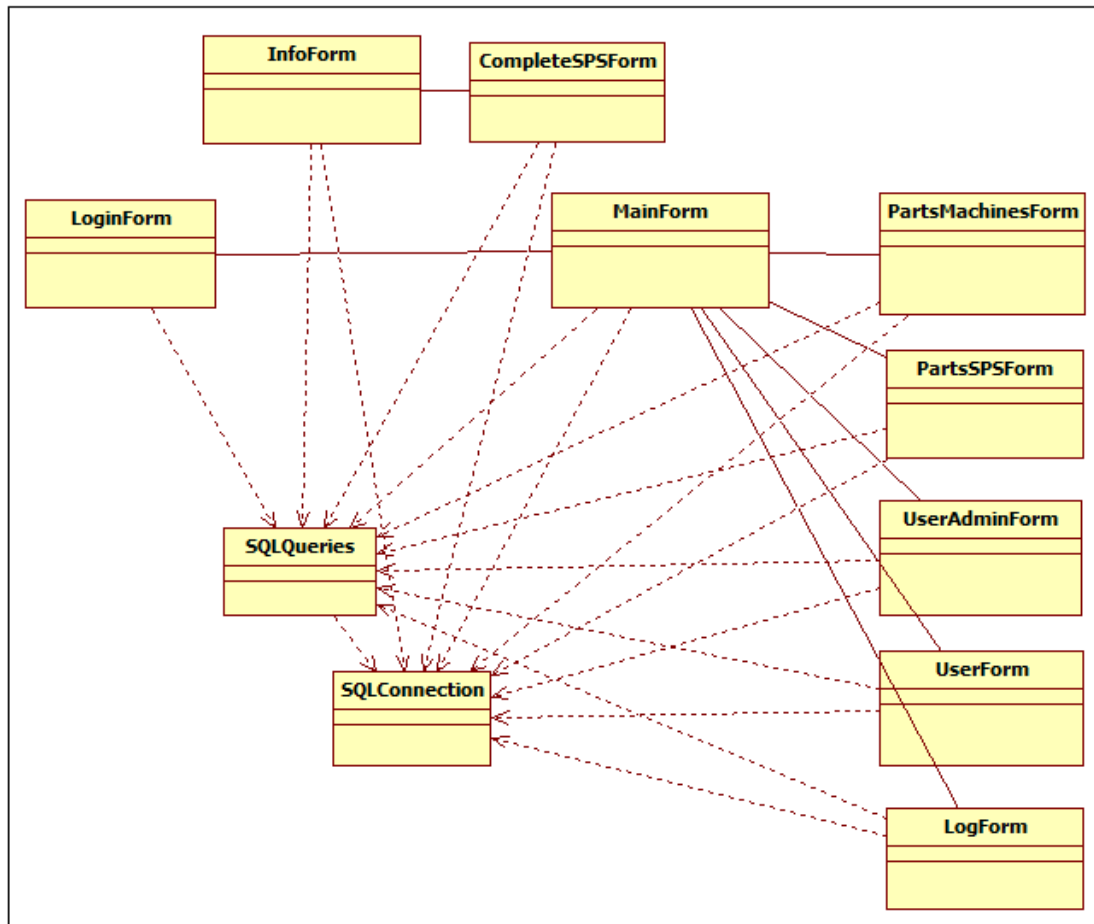
Το διάγραμμα περιπτώσεων χρήσης τροποποιήθηκε κατάλληλα κατά την φάση ανάπτυξης ώστε να περιλάβει τις λειτουργίες που προστέθηκαν.



Εικόνα 3-22: Το ανανεωμένο διάγραμμα περιπτώσεων χρήσης στη φάση κατασκευής του έργου.

3.5.2 Διαγράμματα τάξεων (Class Diagrams)

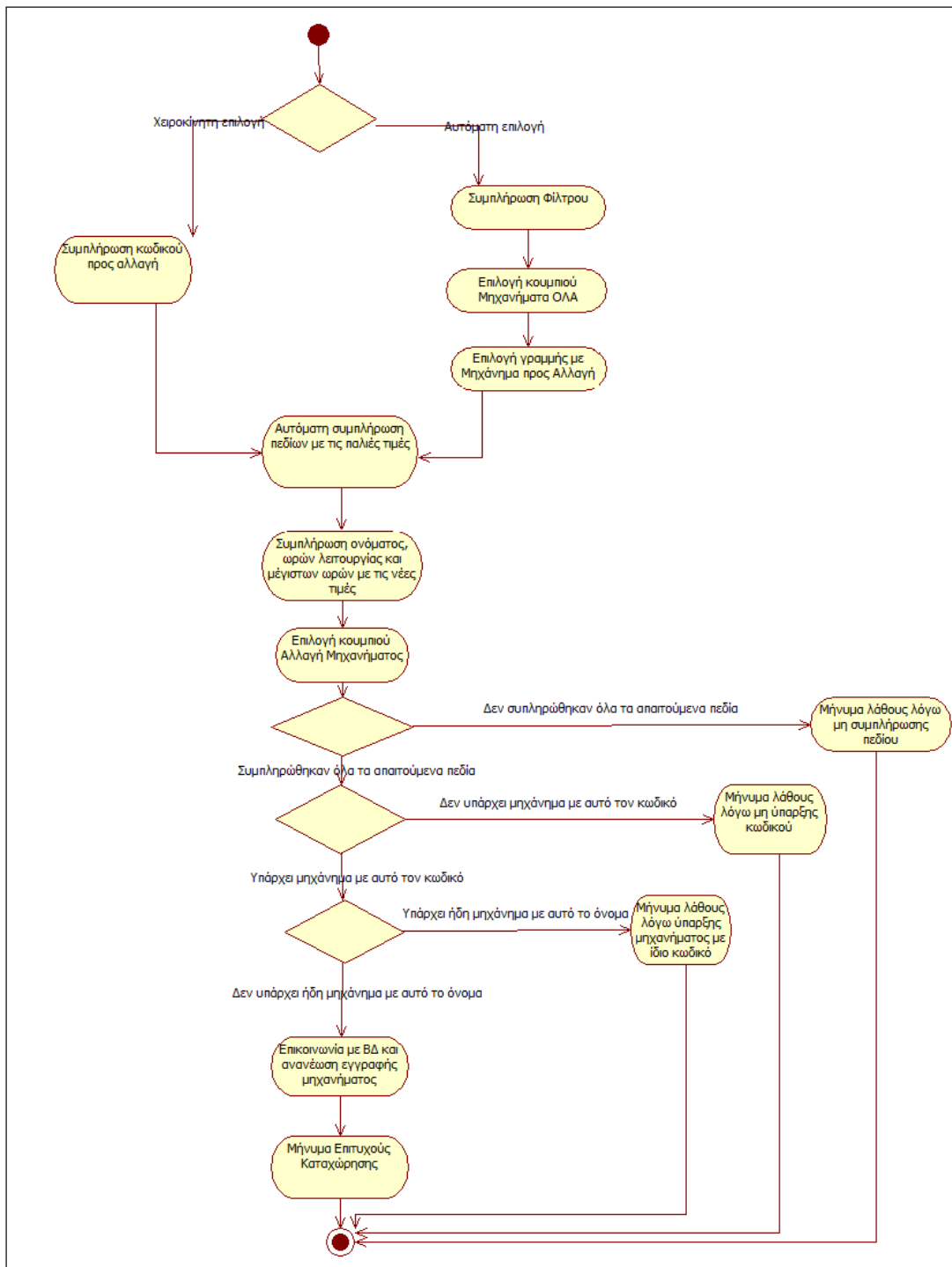
Το διάγραμμα τάξεων τροποποιήθηκε κατάλληλα κατά την φάση ανάπτυξης ώστε να περιλάβει τις οθόνες εισαγωγής στοιχείων που προστέθηκαν. Στο διάγραμμα προστέθηκε η κλάση CompleteSPSForm που περιγράφει την οθόνη καταχώρισης ολοκλήρωσης ΣΠΣ η οποία μπορεί να ενεργοποιηθεί από την οθόνη πληροφοριών. Επιπλέον προστέθηκε η κλάση LogForm η οποία περιγράφει την οθόνη προβολής κινήσεων χρηστών που είναι διαθέσιμη μόνο στο διαχειριστή του συστήματος μέσω επιλογής στην κύρια οθόνη της εφαρμογής.



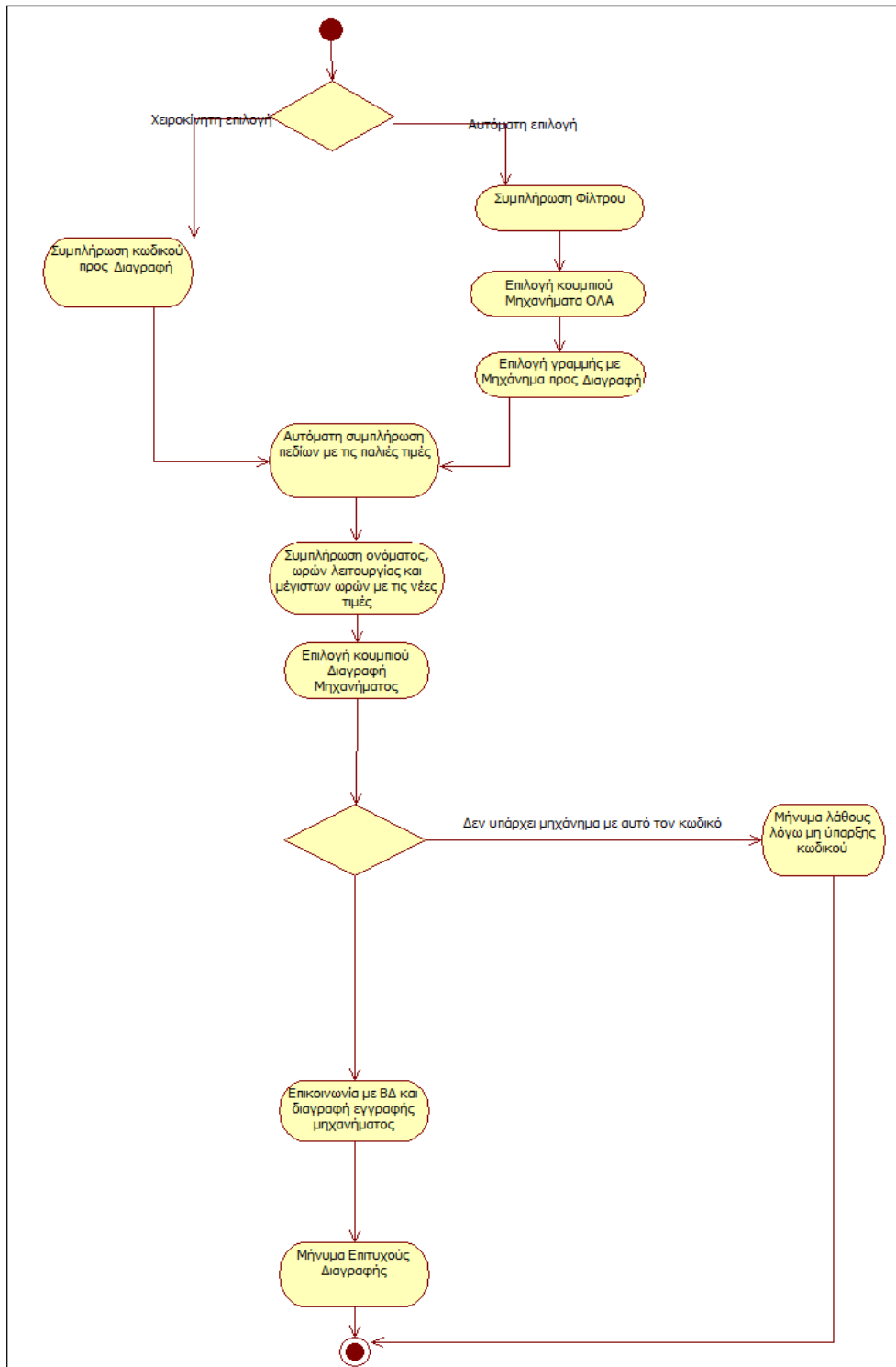
Εικόνα 3-23: Το ανανεωμένο διάγραμμα τάξεων κατά τη φάση ανάπτυξης του έργου.

3.5.3 Διαγράμματα δραστηριοτήτων(Activity Diagrams)

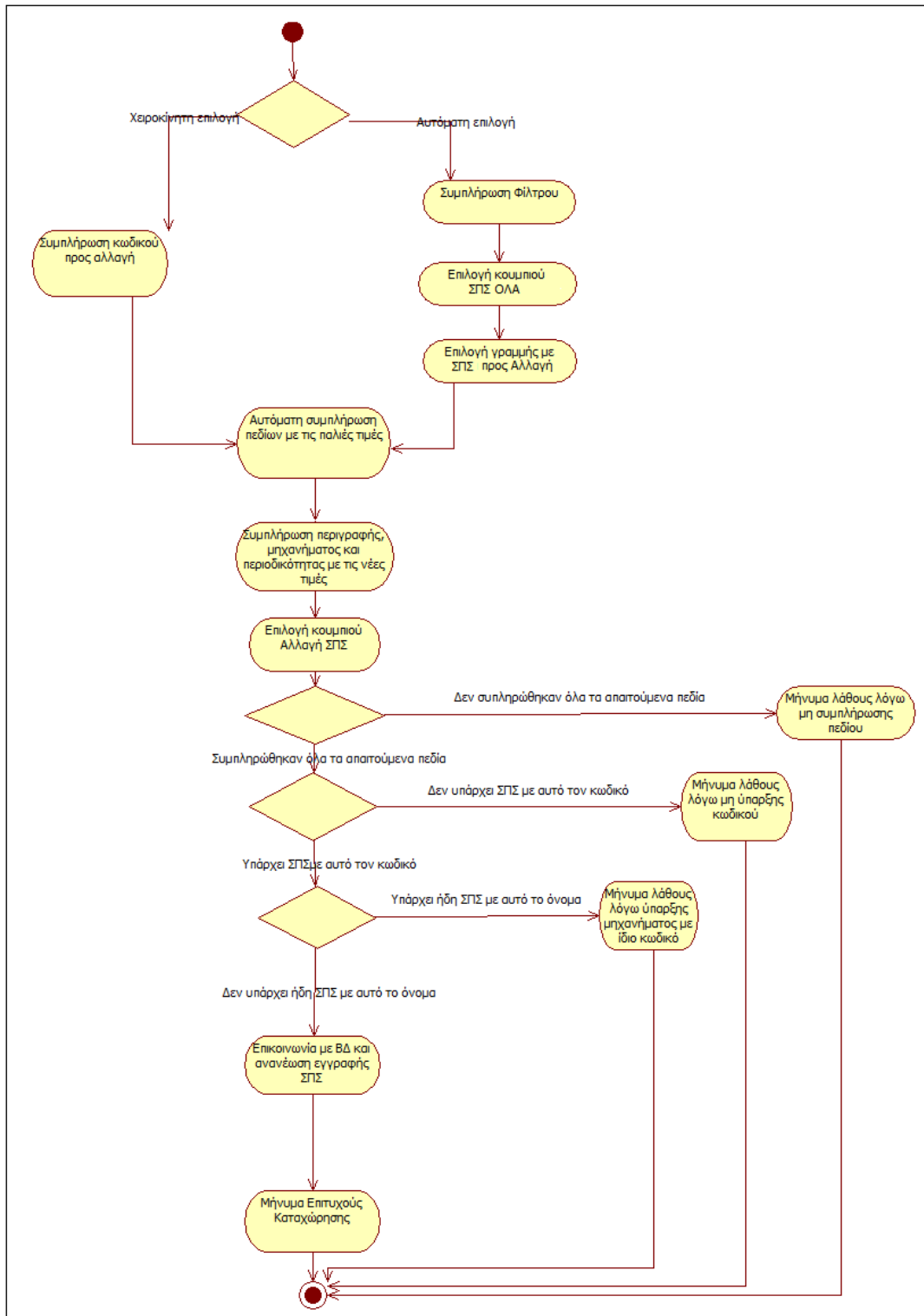
Λόγω της νέας δυνατότητας αυτόματης συμπλήρωσης κωδικών και πεδίων που προστέθηκε άλλαξαν τα διαγράμματα δραστηριοτήτων των λειτουργιών αλλαγής και διαγραφής οντοτήτων. Τα ανανεωμένα διαγράμματα παρατίθενται παρακάτω.



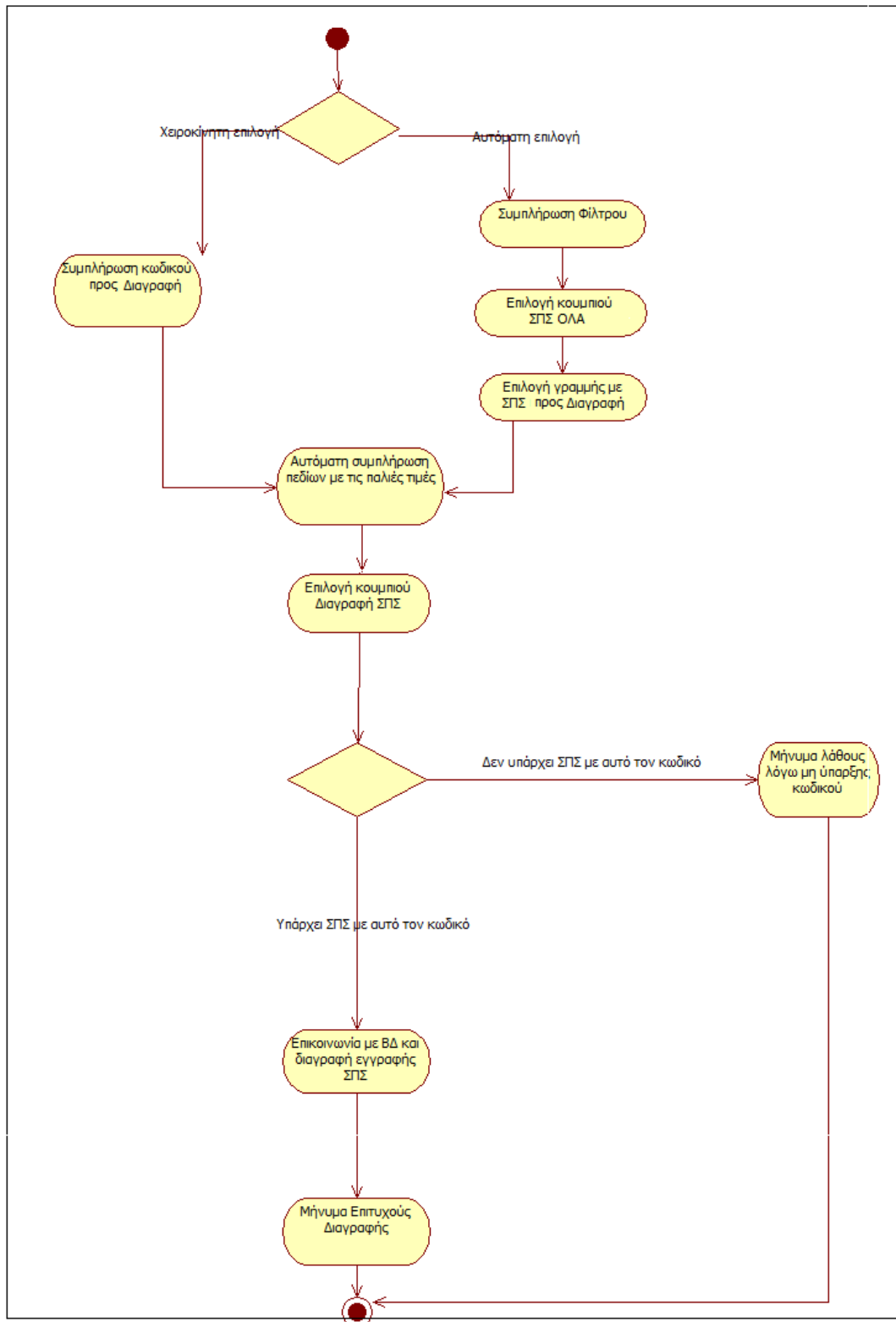
Εικόνα 3-24: Διάγραμμα δραστηριοτήτων λειτουργίας αλλαγής στοιχείων μηχανήματος.



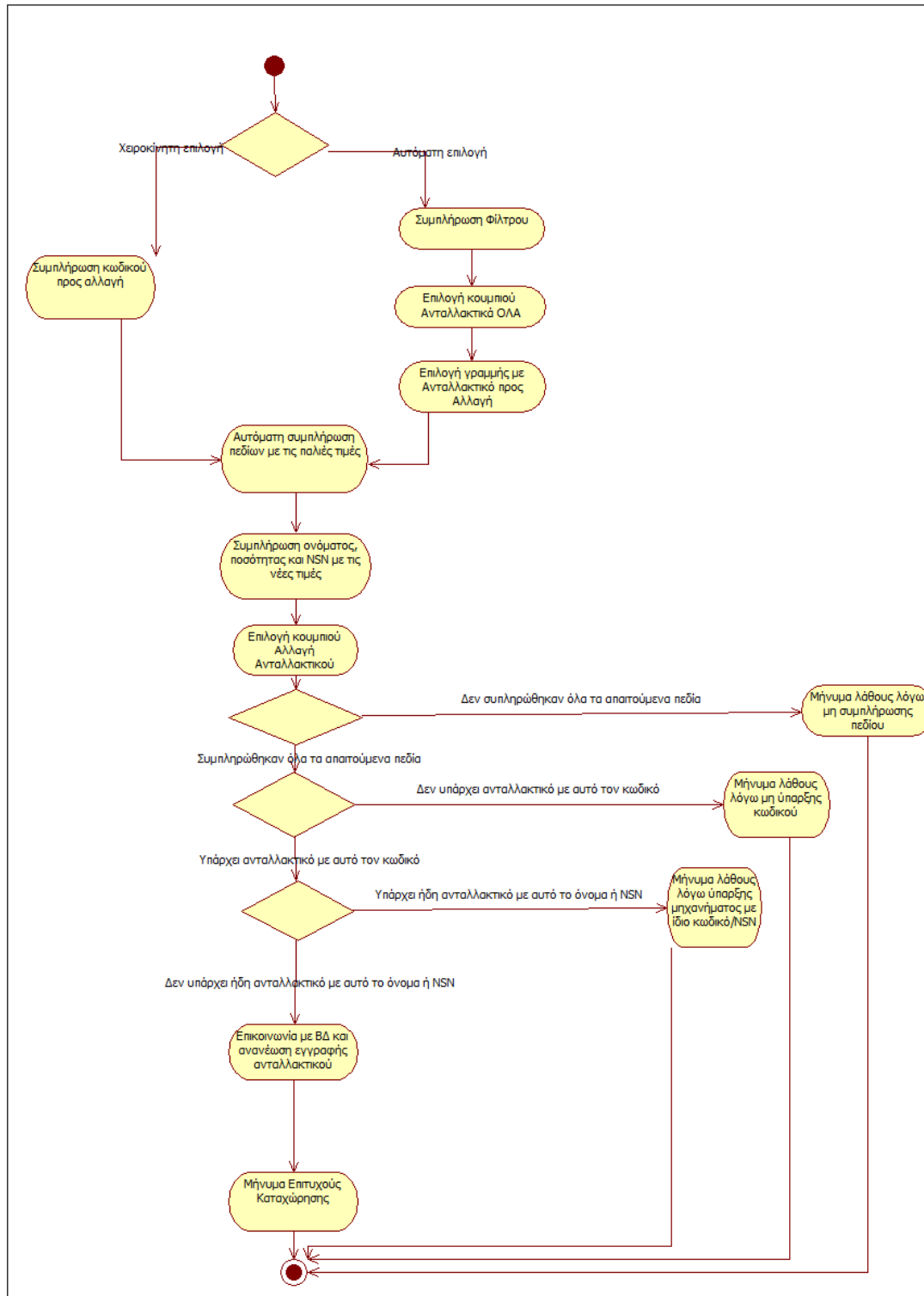
Εικόνα 3-25: Ανανεωμένο διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής μηχανήματος.



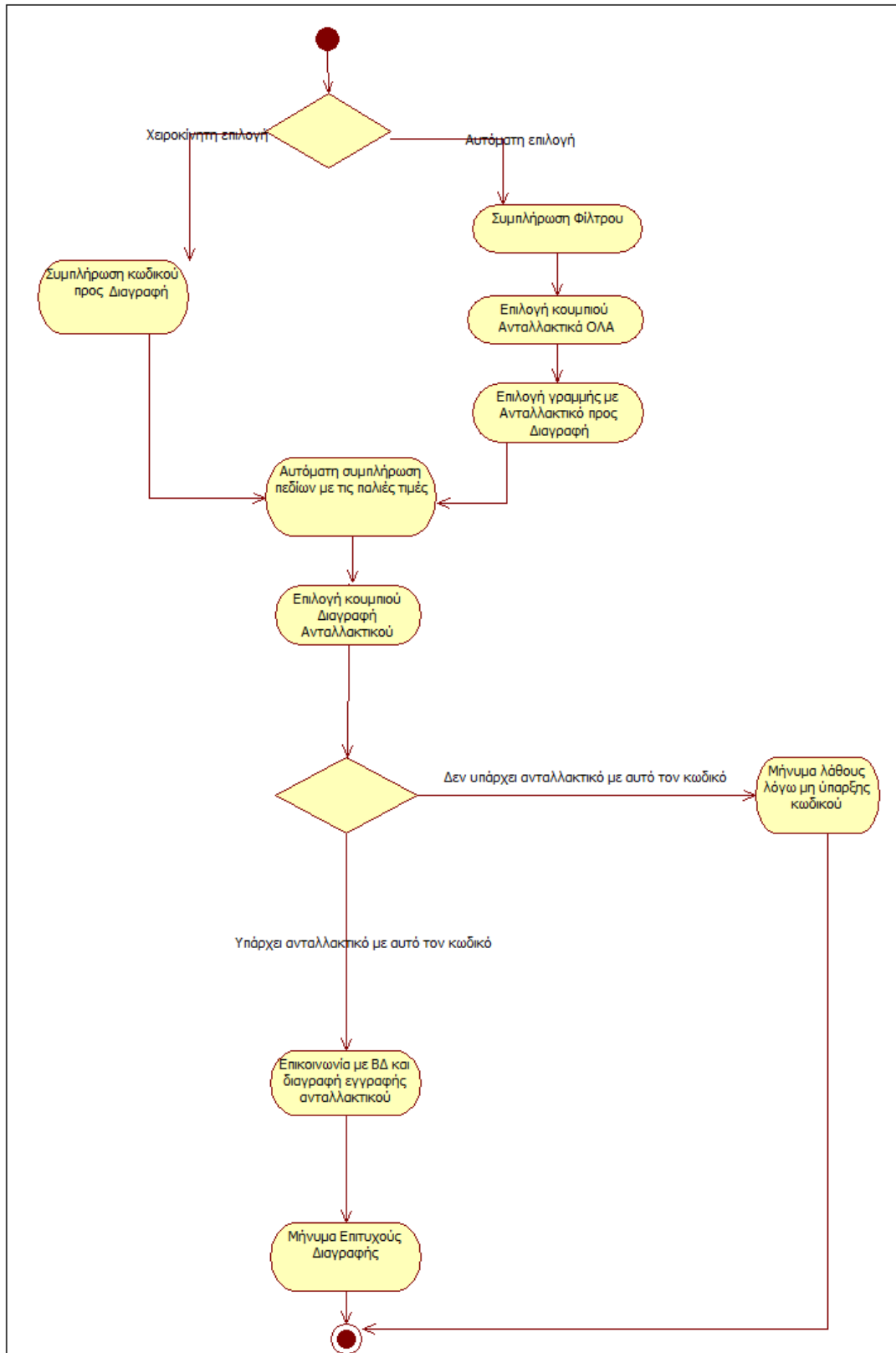
Εικόνα 3-26: Ανανεωμένο διάγραμμα δραστηριοτήτων της λειτουργίας αλλαγής στοιχείων ΣΠΣ.



Εικόνα 3-27: Ανανεωμένο διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής ΣΠΣ.

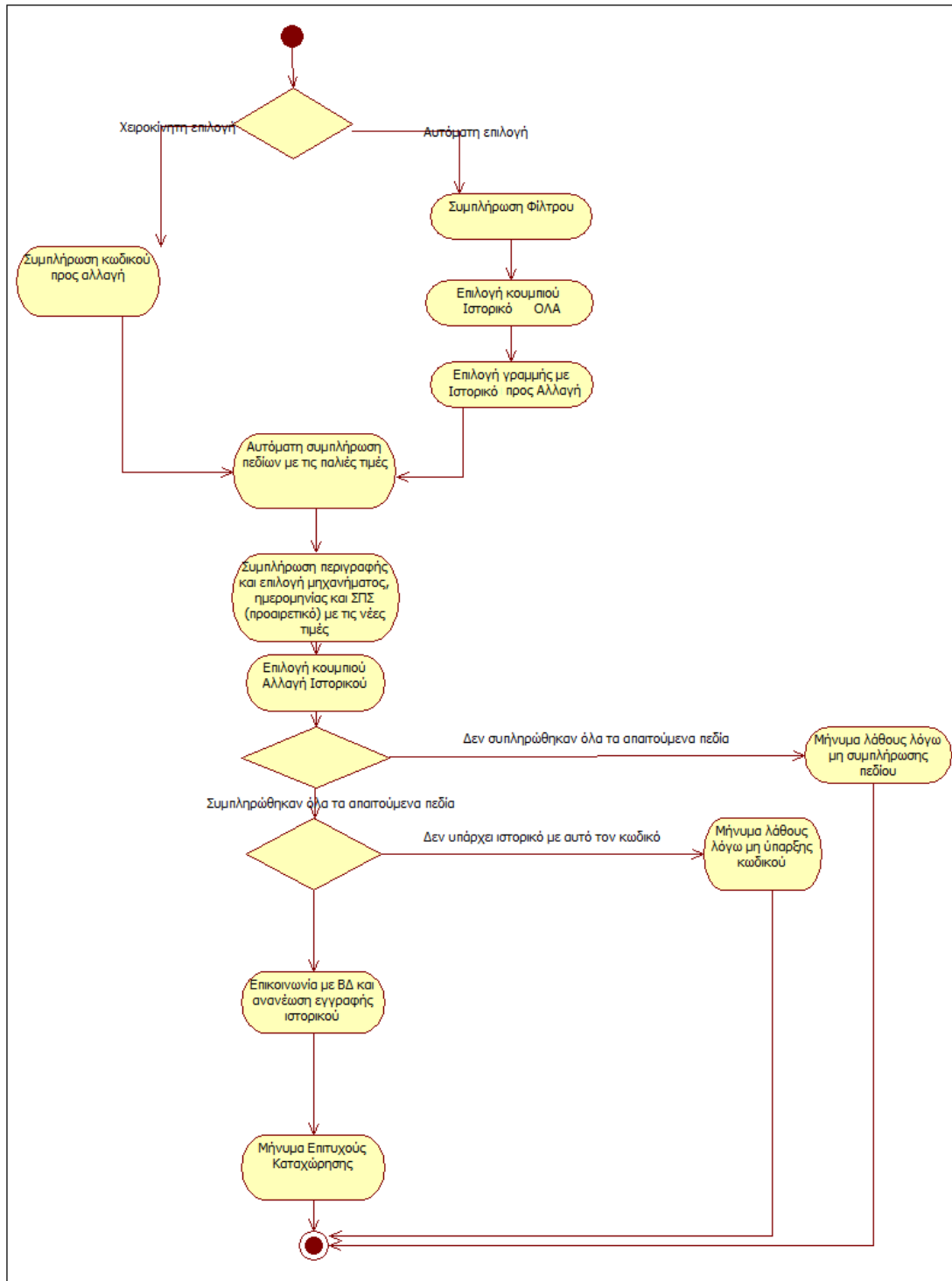


Εικόνα 3-28: Ανανεωμένο διάγραμμα δραστηριοτήτων της λειτουργίας αλλαγής στοιχείων ανταλλακτικού.

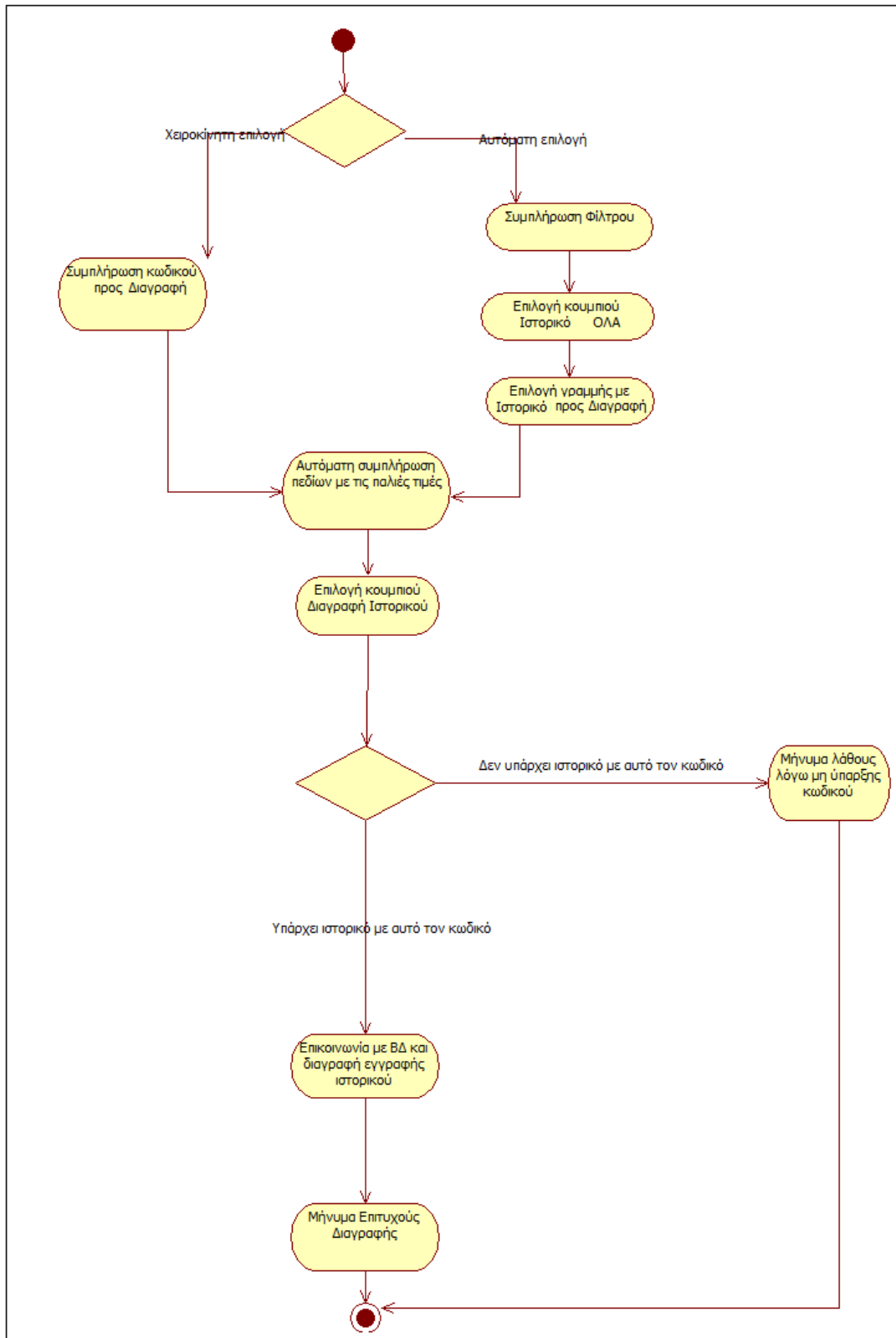


Εικόνα 3-29: Ανανεωμένο διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής ανταλλακτικού.

Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP

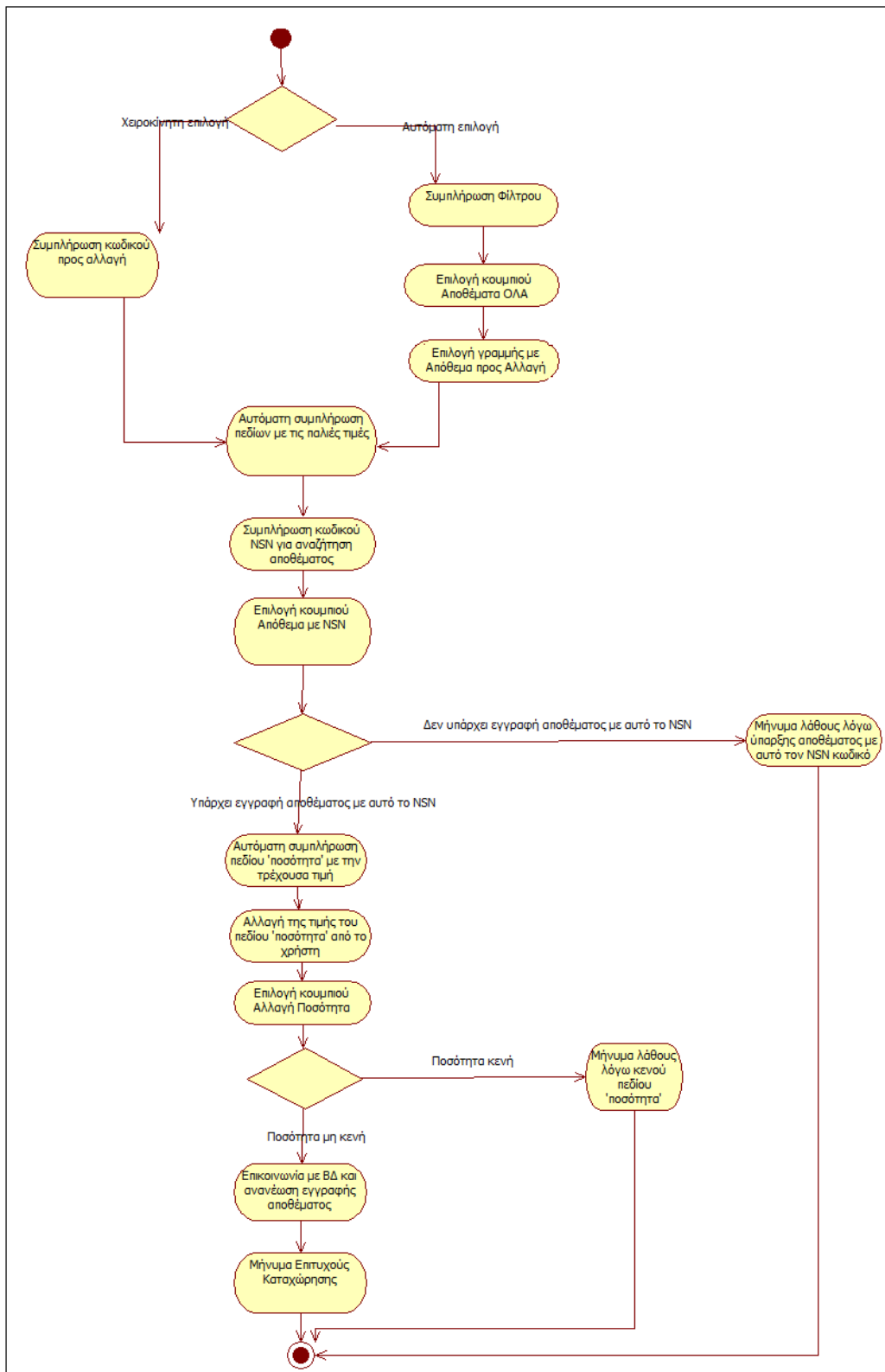


Εικόνα 3-30: Ανανεωμένο διάγραμμα δραστηριοτήτων της λειτουργίας αλλαγής στοιχείων ιστορικού συντήρησης.



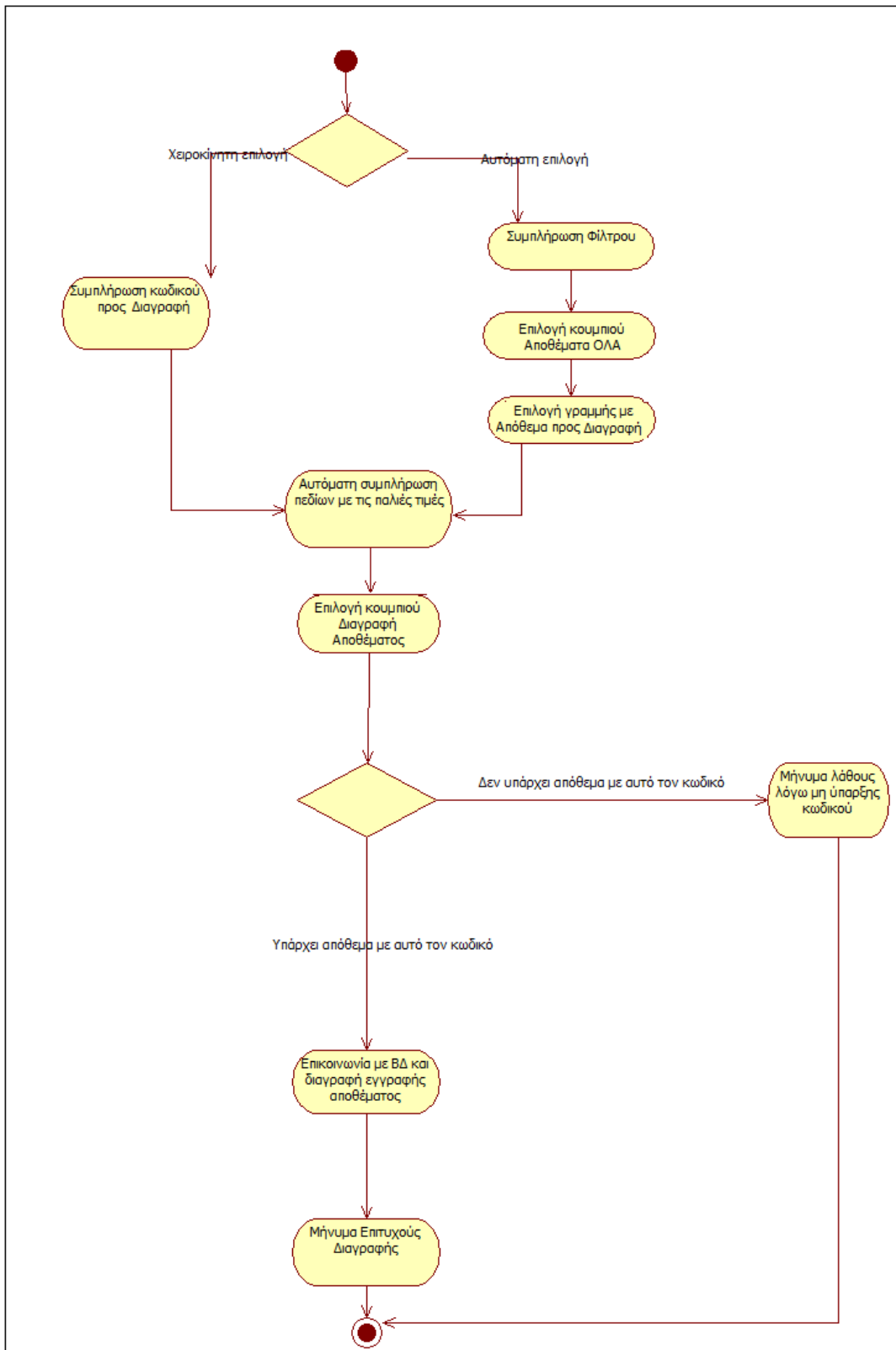
Εικόνα 3-31: Το Ανανεωμένο διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής ιστορικού συντήρησης.

Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP



Εικόνα 3-32: Ανανεωμένο διάγραμμα δραστηριοτήτων της λειτουργίας αλλαγής ποσότητας αποθέματος.

Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP

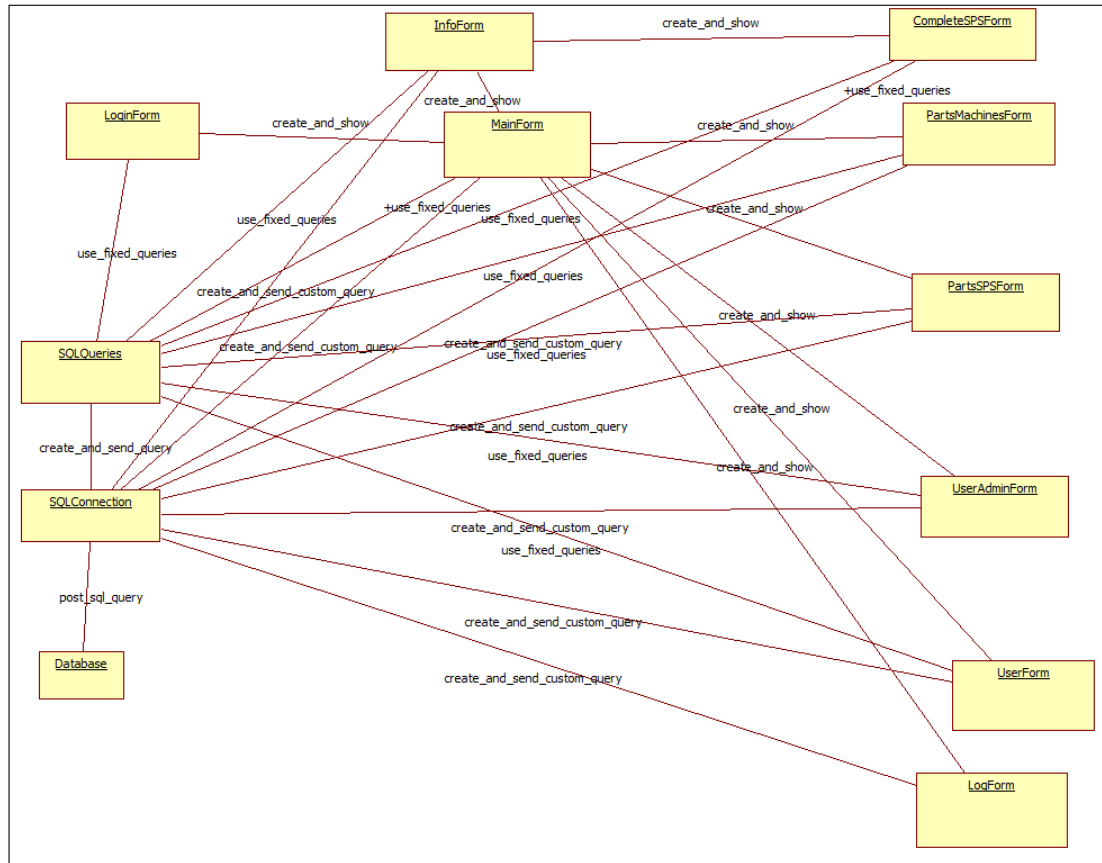


Εικόνα 3-33: Ανανεωμένο διάγραμμα δραστηριοτήτων της λειτουργίας διαγραφής αποθέματος.

Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP

3.5.4 Διαγράμματα συνεργασίας(Collaboration Diagrams)

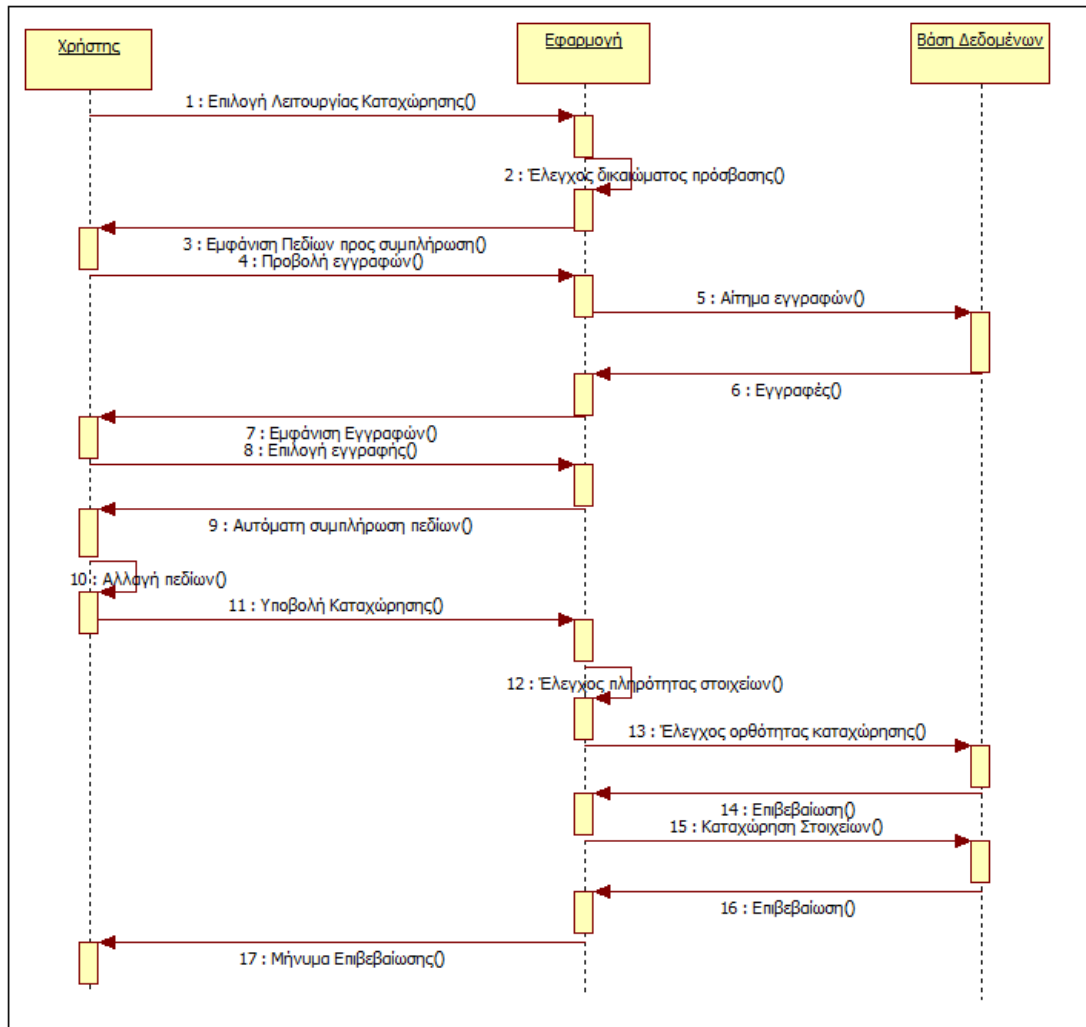
Το διάγραμμα συνεργασίας του έργου τροποποιήθηκε ελαφρά κατά τη φάση κατασκευής καθώς προστέθηκε η λειτουργία ολοκλήρωσης ΣΠΣ και αυτόματης καταχώρησης ιστορικού συντήρησης. Για την υλοποίηση της λειτουργίας υλοποιήθηκε καινούρια οθόνη (CompleteSPSForm) η οποία δημιουργείται από αντίστοιχο κουμπί στην οθόνη πληροφοριών. Επιπλέον απεικονίζεται το αντικείμενο της οθόνης προβολής κινήσεων χρηστών (LogForm) η οποία εμφανίζει τις κινήσεις των χρηστών μετά από επιλογή της σχετικής λειτουργίας στην κεντρική οθόνη από το διαχειριστή του συστήματος.



Εικόνα 3-34: Το διάγραμμα συνεργασίας κατά τη φάση κατασκευής του έργου.

3.5.5 Διαγράμματα σειράς (Sequence Diagrams)

Η προσθήκη της νέας δυνατότητας αυτόματης συμπλήρωσης κωδικών και πεδίων απεικονίζεται και στο διάγραμμα σειράς που περιγράφει μία επιτυχημένη λειτουργία καταχώρησης.



Εικόνα 3-35: Ανανεωμένο διάγραμμα σειράς κατά τη φάση της κατασκευής του έργου.

3.5.6 Διαγράμματα διανομής (Deployment Diagrams)

Το διάγραμμα διανομής της εφαρμογής δεν άλλαξε κατά τη φάση κατασκευής.

3.5.7 Διάγραμμα Οντοτήτων Συσχετίσεων της Βάσης Δεδομένων

Η βάση δεδομένων που κρατά τις πληροφορίες σχεδιάστηκε στα πρώτα στάδια κατασκευής του έργου με βάση τις προδιαγραφές του. Στο παρακάτω σχήμα παρουσιάζεται το μοντέλο οντοτήτων συσχετίσεων της βάσης δεδομένων.

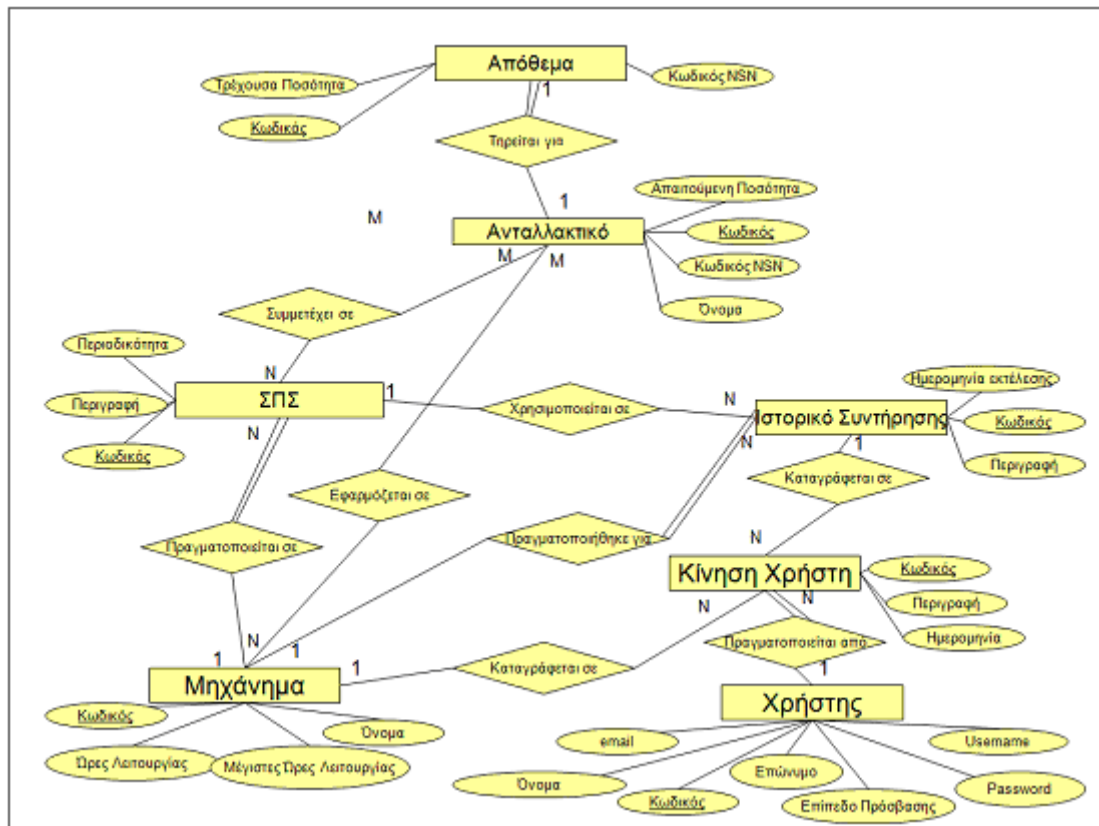
Οι διπλές γραμμές μεταξύ οντοτήτων και συσχετίσεων φανερώνουν ότι η εν λόγω οντότητα πρέπει να συμμετέχει τουλάχιστον μία φορά στην εν λόγω συσχέτιση για να έχει νόημα. Αναλυτικά:

- ένα Ιστορικό Συντήρησης πρέπει να αφορά τουλάχιστον ένα (και ακριβώς ένα) Μηχάνημα, γεγονός το οποίο συμβολίζεται από τη διπλή γραμμή μεταξύ της συσχέτισης «Πραγματοποιήθηκε για» και της οντότητας Ιστορικό Συντήρησης

Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP

- ένα Απόθεμα πρέπει να διατηρείται για τουλάχιστον ένα (και ακριβώς ένα) Ανταλλακτικό, γεγονός το οποίο συμβολίζεται από τη διπλή γραμμή μεταξύ της συσχέτισης «Τηρείται για» και της οντότητας Απόθεμα
- ένα ΣΠΣ πρέπει να πραγματοποιείται σε τουλάχιστον ένα (και ακριβώς ένα) Μηχάνημα, γεγονός το οποίο συμβολίζεται από τη διπλή γραμμή μεταξύ της συσχέτισης «Πραγματοποιείται σε» και της οντότητας ΣΠΣ
- μία Κίνηση Χρήστη πρέπει να πραγματοποιείται από τουλάχιστον έναν (και ακριβώς έναν) Χρήστη, γεγονός το οποίο συμβολίζεται από τη διπλή γραμμή μεταξύ της συσχέτισης «Πραγματοποιείται από» και της οντότητας Κίνηση Χρήστη

Σε αυτό το σημείο αξίζει να σημειωθεί ότι η διπλή γραμμή συμβολίζει συσχέτιση με σημασία «τουλάχιστον ένα». Η σημασία «ακριβώς ένα» είναι ανεξάρτητη και προκύπτει αν και ο λόγος πληθικότητας μιας συσχέτισης προς μία οντότητα είναι 1.



Εικόνα 3-36: Το μοντέλο Οντοτήτων-Συσχετίσεων της βάσης δεδομένων του συστήματος.

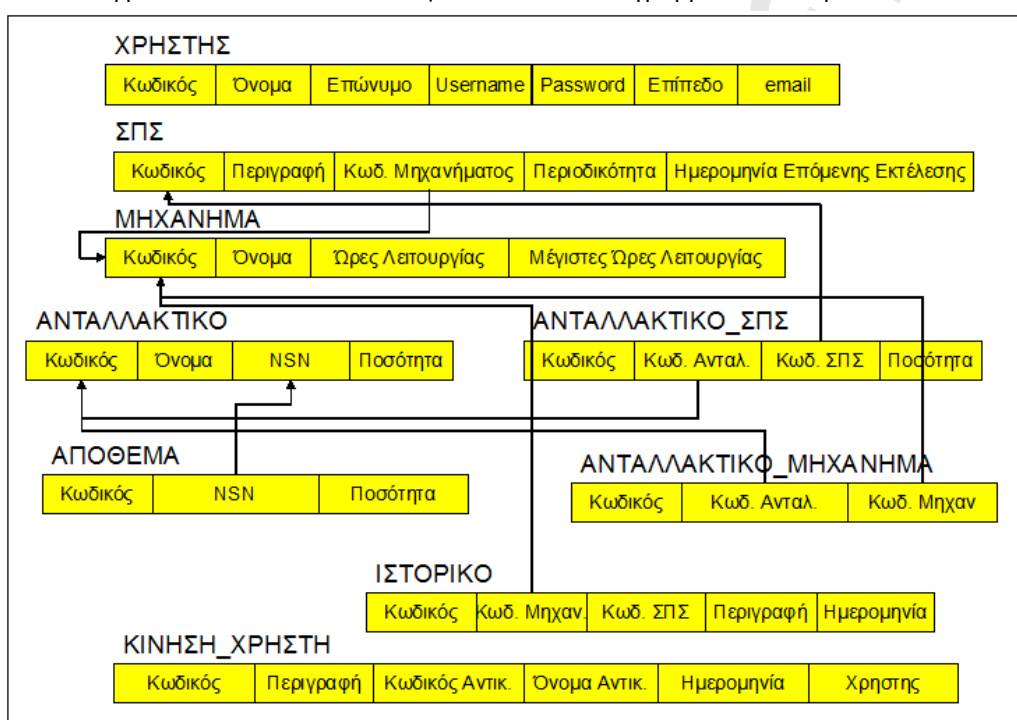
Στο διάγραμμα αναπαρίστανται και οι λόγοι πληθικότητας με τους ακόλουθους τύπους:

- 1-1(ένα-προς-ένα) : Αντιστοιχίζεται μια οντότητα ενός τύπου με το πολύ ή ακριβώς μια οντότητα ενός άλλου τύπου. Για παράδειγμα το κάθε Απόθεμα αντιστοιχεί το πολύ σε ένα Ανταλλακτικό και αντίστροφα.
- N (ένα-προς-πολλά): Αντιστοιχίζεται μια οντότητα ενός τύπου με κανένα, ένα ή πολλά στιγμιότυπα ενός άλλου τύπου. Κάθε ΣΠΣ μπορεί να χρησιμοποιηθεί σε πολλά Ιστορικά Συντήρησης και κάθε Ιστορικό συντήρησης μπορεί να αφορά το πολύ ένα ΣΠΣ.
- M-N (πολλά-προς-πολλά):Αντιστοιχίζεται κάθε στιγμιότυπο του ενός τύπου με ένα, κανένα ή πολλά στιγμιότυπα του άλλου τύπου. Για παράδειγμα ένα Ανταλλακτικό μπορεί να

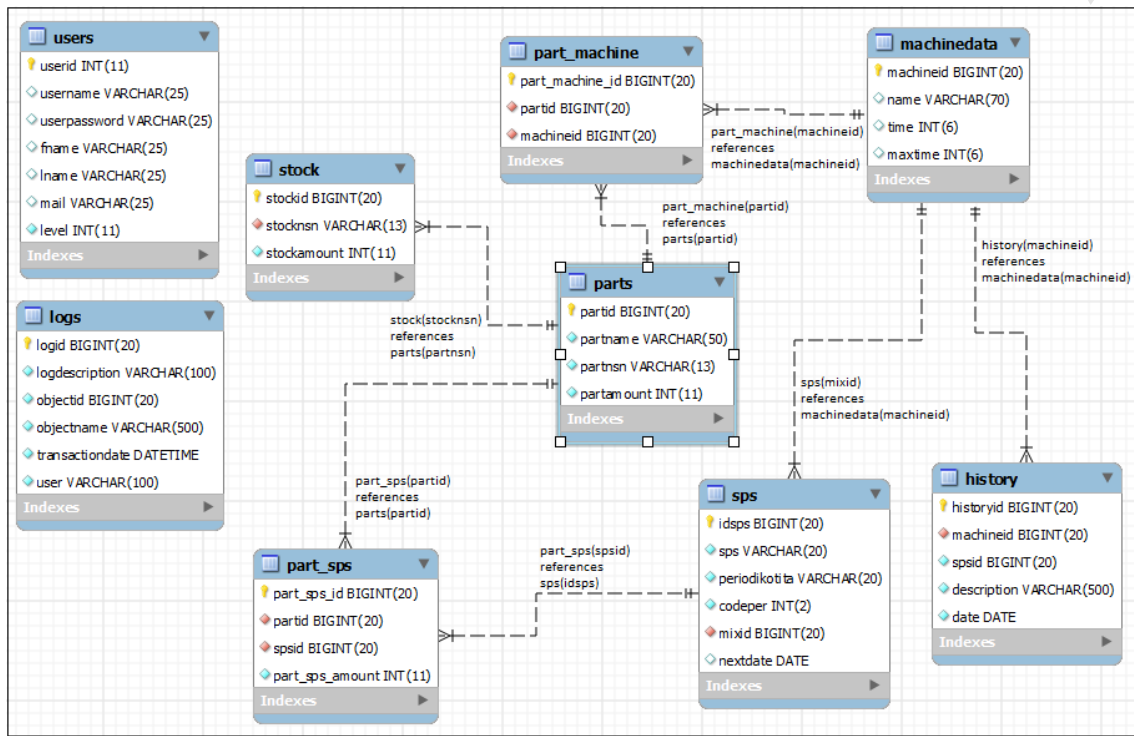
εφαρμόζεται σε πολλά Μηχανήματα και για κάθε Μηχάνημα υπάρχουν πολλά Ανταλλακτικά που μπορούν να προσαρμοστούν σε αυτό.

3.5.8 Διάγραμμα Σχεσιακού Μοντέλου της Βάσης Δεδομένων

Από το μοντέλο Οντοτήτων-Συσχετίσεων, μετά από κατάλληλους μετασχηματισμούς, δημιουργήθηκε το σχεσιακό μοντέλο της Βάσης Δεδομένων. Το σχεσιακό μοντέλο περιλαμβάνει τους πίνακες που απαρτίζουν τη βάση, τα πεδία τους και σχέσεις ξένων κλειδιών που αναπτύσσονται μεταξύ τους. Στις επόμενες 2 εικόνες παρουσιάζεται το σχεσιακό μοντέλο της βάσης. Στη μεν πρώτη το σχεσιακό μοντέλο έχει την κλασική μορφή από τη θεωρία Βάσεων Δεδομένων ενώ στη δεύτερη έχει παραχθεί με τη χρήση της εφαρμογής MySQL Workbench και απεικονίζει την τρέχουσα κατάσταση του μοντέλου της Βάσης Δεδομένων του συστήματος. Η δεύτερη εικόνα περιλαμβάνει τα ονόματα των πεδίων και των πινάκων στα αγγλικά. Γίνεται εύκολα αντιληπτό ότι τα δύο διαγράμματα είναι όμοια.



Εικόνα 3-37: Το Σχεσιακό μοντέλο της βάσης δεδομένων σε κλασική μορφή.



Εικόνα 3-38: Το Σχεσιακό μοντέλο της βάσης δεδομένων από την εφαρμογή MySQL Workbench.

Κεφάλαιο 4: Περιγραφή της Εφαρμογής

Στο παρόν κεφάλαιο περιγράφεται αναλυτικά η τελική έκδοση της εφαρμογής και παρουσιάζονται αναλυτικά παραδείγματα χρήσης της που καταδεικνύουν την επιτυχημένη υλοποίηση της λειτουργικότητας που ορίστηκε στις προδιαγραφές του 3^{ου} κεφαλαίου.

4.1 Συνοπτική Περιγραφή της Εφαρμογής

Η εφαρμογή διαχείρισης προγραμμάτων συντήρησης που υλοποιήθηκε στα πλαίσια της παρούσας εργασίας μπορεί να χρησιμοποιηθεί για τη διαχείριση ενός συνόλου μηχανημάτων και ανταλλακτικών, υπό το πλαίσιο προγραμματισμένων, ή μη, διαδικασιών συντήρησης που καταγράφονται και αποθηκεύονται σε αυτή. Όπως αναφέρθηκε αναλυτικά στο κομμάτι των προδιαγραφών, αποθηκεύονται πληροφορίες για τα μηχανήματα, τα ανταλλακτικά, τα αποθέματα ανταλλακτικών, τα συστήματα προγραμματισμένης συντήρησης και για τις εργασίες συντήρησης που έχουν πραγματοποιηθεί. Υποστηρίζονται διαφορετικοί ρόλοι χρηστών στο σύστημα με διαφορετικά δικαιώματα πρόσβασης. Υπάρχει τέλος σύστημα καταγραφής των κινήσεων των χρηστών.

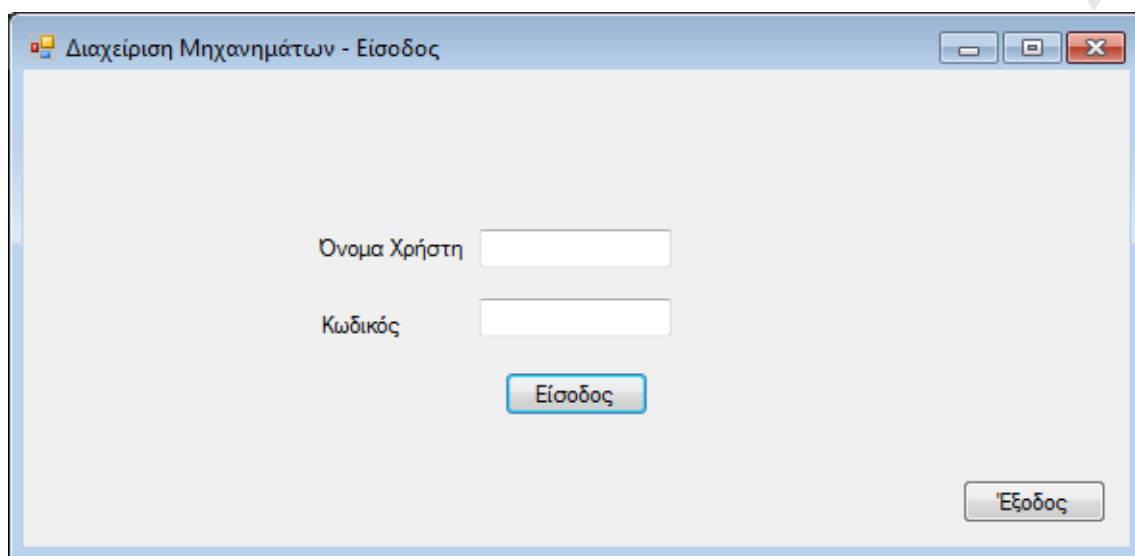
Η εφαρμογή μπορεί να εκτελεστεί σε ηλεκτρονικό υπολογιστή με λειτουργικό σύστημα Windows. Η εκτέλεση της εφαρμογής έχει δοκιμαστεί ορθά σε συστήματα με εκδόσεις των Windows 7 και Windows XP. Η αποθήκευση των δεδομένων που διαχειρίζεται η εφαρμογή γίνεται στο σύστημα διαχείρισης βάσεων δεδομένων MySQL. Είναι απαραίτητο να εγκατασταθεί ο MySQL Server στο σύστημα που θα εγκατασταθεί η εφαρμογή. Εναλλακτικά ο MySQL server μπορεί να εγκατασταθεί σε απομακρυσμένο σύστημα και η επικοινωνία της εφαρμογής με αυτό να γίνεται μέσω δικτύου. Λεπτομέρειες σχετικά με την εγκατάσταση της εφαρμογής MySQL server και την απαραίτητη παραμετροποίηση υπάρχουν στο παράρτημα της εργασίας. Πριν την πρώτη εκτέλεση της εφαρμογής θα πρέπει να δημιουργηθεί το μοντέλο της βάσης και να αποθηκευτεί στο ΣΔΒΔ MySQL ώστε η εφαρμογή να μπορεί να επικοινωνήσει με αυτό. Λεπτομέρειες σχετικά με τη δημιουργία του μοντέλου της βάσης δεδομένων και την εισαγωγή του στον MySQL Server υπάρχουν στο 2^ο μέρος του παραρτήματος αυτής της εργασίας. Η εφαρμογή δεν απαιτεί εγκατάσταση. Για τη χρήση της απαιτείται η αποσυμπίεση ενός .zip αρχείου και η εκτέλεση του εκτελέσιμου MIKP.exe που βρίσκεται στο φάκελο που προκύπτει από αυτό.

Έγινε προσπάθεια η συνολική λειτουργικότητα της εφαρμογής να χωριστεί σε σαφώς διαχωρισμένες οθόνες εκτέλεσης οι οποίες ομαδοποιούν κοινές λειτουργίες. Για το σκοπό αυτό χρησιμοποιήθηκαν διαφορετικές φόρμες εισαγωγής στοιχείων. Ακολουθούν σύντομες περιγραφές των επιμέρους οθονών εκτέλεσης της εφαρμογής.

4.1.1 Η οθόνη εισόδου

Η οθόνη αυτή είναι η αρχική οθόνη της εφαρμογής και εμφανίζεται αμέσως μόλις ο χρήστης εκκινήσει την εφαρμογή. Σε αυτή την οθόνη ο χρήστης εισάγει το Όνομα Χρήστη (username) και τον Κωδικό (password) που αντιστοιχεί σε αυτόν. Αν αλλάξει γνώμη και δεν επιθυμεί να εκκινήσει την εφαρμογή μπορεί να επιλέξει να την τερματίσει επιλέγοντας το κουμπί έξοδος. Όπως αναφέρθηκε και στις προδιαγραφές, νέοι χρήστες δημιουργούνται μόνο από το διαχειριστή της εφαρμογής. Ο μόνος χρήστης που υπάρχει αμέσως μετά την πρώτη εκκίνηση της εφαρμογής είναι ο διαχειριστής της με Όνομα Χρήστη admin και Κωδικό 1234.

Αφού ο χρήστης εισάγει τα στοιχεία του η εφαρμογή επικοινωνεί με τη βάση δεδομένων ώστε να ελέγξει αν ο χρήστης υπάρχει και έχει τα συγκεκριμένα στοιχεία. Αν ο χρήστης εισάγει λανθασμένα στοιχεία ή δεν εισάγει στοιχεία εμφανίζεται σχετικό μήνυμα λάθους. Αν τα στοιχεία είναι σωστά τυπώνεται μήνυμα υποδοχής και εμφανίζεται η κεντρική οθόνη της εφαρμογής που περιγράφεται στην επόμενη παράγραφο.



Εικόνα 4-1: Η οθόνη εισόδου της εφαρμογής.

4.1.2 Η κεντρική οθόνη εισαγωγής στοιχείων της εφαρμογής

Η οθόνη αυτή είναι η κεντρική οθόνη της εφαρμογής και εμφανίζεται μεγιστοποιημένη αμέσως μετά την επιτυχή είσοδο του χρήστη.

Στο κάτω μέρος της εφαρμογής υπάρχει πλέγμα πληροφοριών (grid) στο οποίο εμφανίζονται πληροφορίες για τις οντότητες της εφαρμογής συνολικά. Δεξιά από το πλέγμα υπάρχει πεδίο κειμένου στο οποίο εμφανίζονται πληροφορίες για κάθε ξεχωριστή οντότητα όποτε επιλέγεται μία γραμμή από το πλέγμα. Στο πάνω μέρος της εφαρμογής υπάρχουν κουμπιά για την εισαγωγή και προβολή δεδομένων.

Τα κουμπιά «Μηχανήματα ΟΛΑ», «Ανταλλακτικά ΟΛΑ», «ΣΠΣ ΟΛΑ», «Ιστορικό ΟΛΑ» «Αποθέματα ΟΛΑ» συμπληρώνουν το πλέγμα (grid) στο κάτω μέρος της εφαρμογής με πληροφορίες για τα μηχανήματα, τα ανταλλακτικά, τα ΣΠΣ, τις εγγραφές Ιστορικού και τα αποθέματα ανταλλακτικών του συστήματος αντίστοιχα. Για κάθε ένα από τα κουμπιά από αυτά μπορεί να εφαρμοστεί φίλτρο ώστε να περιοριστούν οι εγγραφές που εμφανίζονται.

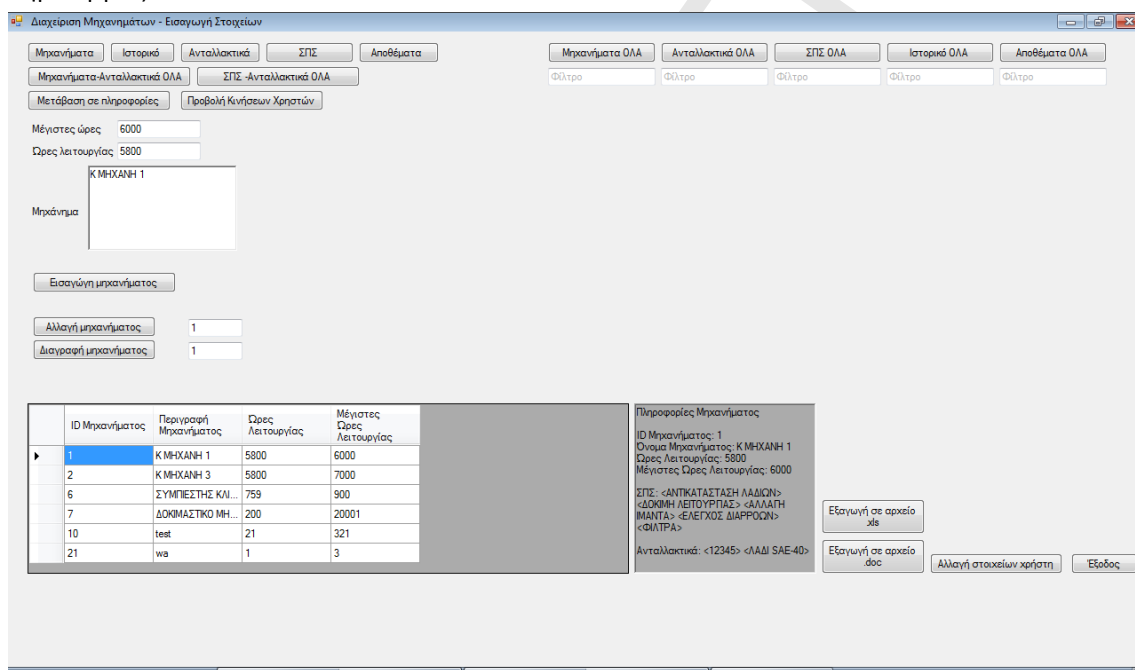
Τα κουμπιά «Μηχανήματα», «Ανταλλακτικά», «ΣΠΣ», «Ιστορικό», «Αποθέματα» ενεργοποιούν πεδία για την εισαγωγή, διαγραφή και αλλαγή στοιχείων για τα μηχανήματα, τα ανταλλακτικά, τα ΣΠΣ, τις εγγραφές ιστορικού και τα αποθέματα που διατηρεί η εφαρμογής αντίστοιχα. Αν επιλεγεί κάποιο από αυτά τα κουμπιά εμφανίζονται προς συμπλήρωση μόνο τα πεδία της συγκεκριμένης οντότητας. Για την ακρίβεια

- αν επιλεγεί το κουμπί «Μηχανήματα» εμφανίζονται τα κουμπιά «Εισαγωγή Μηχανήματος», «Αλλαγή Μηχανήματος», «Διαγραφή Μηχανήματος».
- αν επιλεγεί το κουμπί «Ιστορικό» εμφανίζονται τα κουμπιά «Εισαγωγή Ιστορικού», «Αλλαγή Ιστορικού» και «Διαγραφή Ιστορικού».
- αν επιλεγεί το κουμπί «Ανταλλακτικά» εμφανίζονται τα κουμπιά «Εισαγωγή Ανταλλακτικού», «Αλλαγή Ανταλλακτικού», «Διαγραφή Ανταλλακτικού», «Αντιστοίχιση Ανταλλακτικού με Μηχάνημα» και «Αντιστοίχιση Ανταλλακτικού με ΣΠΣ»
- αν επιλεγεί το κουμπί «ΣΠΣ» εμφανίζονται τα κουμπιά «Εισαγωγή ΣΠΣ», «Αλλαγή ΣΠΣ» και «Διαγραφή ΣΠΣ».
- αν επιλεγεί το κουμπί «Αποθέματα» εμφανίζονται τα κουμπιά «Απόθεμα με NSN», «Αλλαγή Ποσότητα», και «Διαγρ. αποθέματος».

Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP

Αν έχει επιλεγεί κάποιο από τα κουμπιά αυτά και έχει επιλεγεί το αντίστοιχο κουμπί προβολής, επιλέγοντας μία γραμμή από το πλέγμα συμπληρώνονται αυτόματα τα αντίστοιχα πεδία εισαγωγής ώστε να διευκολύνονται οι λειτουργίες ανανέωσης στοιχείων και διαγραφής. Για παράδειγμα (όπως φαίνεται και στην οθόνη) αν έχει επιλεγεί το κουμπί «Μηχανήματα» και στη συνέχεια το κουμπί «Μηχανήματα ΟΛΑ» τότε επιλέγοντας μία γραμμή από το πλέγμα συμπληρώνονται αυτόματα τα πεδία με τις τιμές της συγκεκριμένης εγγραφής. Ο χρήστης μπορεί να αλλάξει τις τιμές τους επιτόπου και να επιλέξει «Αλλαγή Μηχανήματος» ή να επιλέξει «Διαγραφή Μηχανήματος» για να διαγράψει τη συγκεκριμένη εγγραφή. Επιπλέον δεξιά του πλέγματος εμφανίζονται πληροφορίες για το συγκεκριμένο Μηχάνημα.

Τα κουμπιά «Μηχανήματα-Ανταλλακτικά ΟΛΑ» και «ΣΠΣ-Ανταλλακτικά ΟΛΑ» εμφανίζουν όλες τις αντιστοιχίσεις ανταλλακτικών με Μηχανήματα και ΣΠΣ αντίστοιχα. Τα κουμπιά «Μετάβαση σε Πληροφορίες», «Προβολή Κινήσεων Χρηστών» και «Αλλαγή στοιχείων χρήστη» ενεργοποιούν τις αντίστοιχες οθόνες. Τα κουμπιά «Εξαγωγή σε αρχείο .doc» και «Εξαγωγή σε αρχείο .xls» δημιουργούν αρχεία που περιέχουν τα τρέχοντα περιεχόμενα του πλέγματος. Τα αρχεία αποθηκεύονται αυτόματα στην Επιφάνεια Εργασίας του συστήματος και περιλαμβάνουν στο όνομά τους την ημερομηνία και ώρα δημιουργίας.



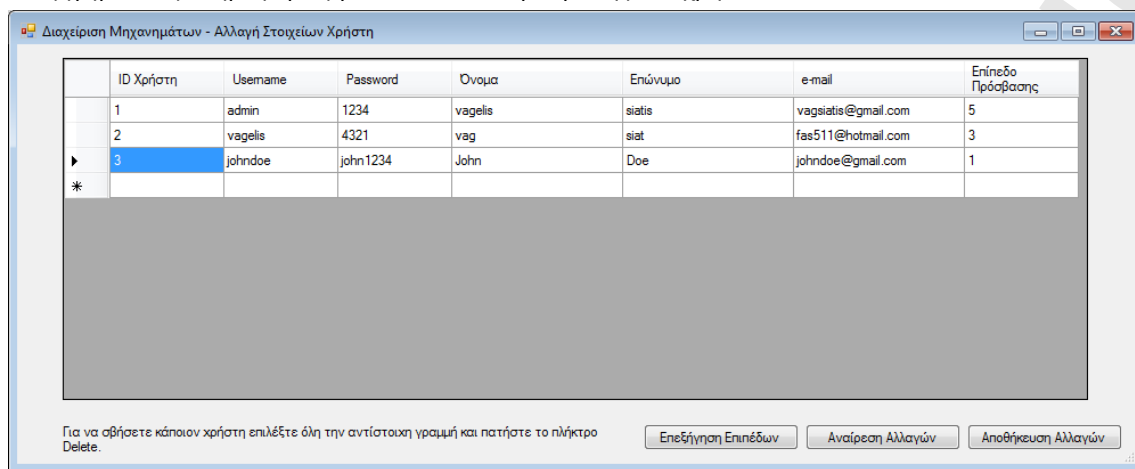
Εικόνα 4-2: Η κεντρική οθόνη της εφαρμογής.

4.1.3 Η οθόνη αλλαγής στοιχείων χρήστη

Η οθόνη αυτή ενεργοποιείται επιλέγοντας το κουμπί «Αλλαγή στοιχείων χρήστη» από την κεντρική οθόνη της εφαρμογής και εμφανίζεται με διαφορετικούς τρόπους ανάλογα με το επίπεδο πρόσβασης του χρήστη.

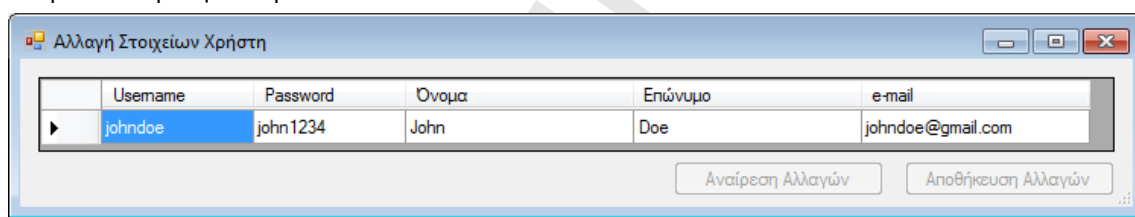
Ο διαχειριστής της εφαρμογής (admin – επίπεδο 5) μπορεί να χρησιμοποιήσει την οθόνη με τη μορφή που έχει στην εικόνα 4.3 και έχει τη δυνατότητα να δημιουργήσει, και να διαγράψει χρήστες καθώς επίσης και να αλλάξει κάποια από τα στοιχεία τους. Για κάθε χρήστη αποθηκεύεται το Όνομα Χρήστη (Username), ο Κωδικός (Password), το Όνομα, το Επώνυμο, μία διεύθυνση e-mail και το επίπεδο πρόσβασης (από 1 έως 5). Ο διαχειριστής μπορεί να επέμβει σε όλα τα πεδία που αφορούν ένα χρήστη. Για να προστεθεί ένας χρήστης συμπληρώνεται η τελευταία κενή γραμμή του πίνακα, ενώ για να διαγραφεί επιλέγεται η αντίστοιχη γραμμή και σβήνεται με το πλήκτρο Delete του Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP

πληκτρολογίου. Οι αλλαγές αποθηκεύονται επιλέγοντας το κουμπί «Αποθήκευση αλλαγών» και αναιρούνται επιλέγοντας το κουμπί «Αναίρεση Αλλαγών». Το κουμπί «Επεξήγηση Επιπέδων» παρέχει επεξηγηματικές πληροφορίες για τα επίπεδα πρόσβασης των χρηστών.



Εικόνα 4-3: Η οθόνη αλλαγής στοιχείων χρήστη για το διαχειριστή της εφαρμογής.

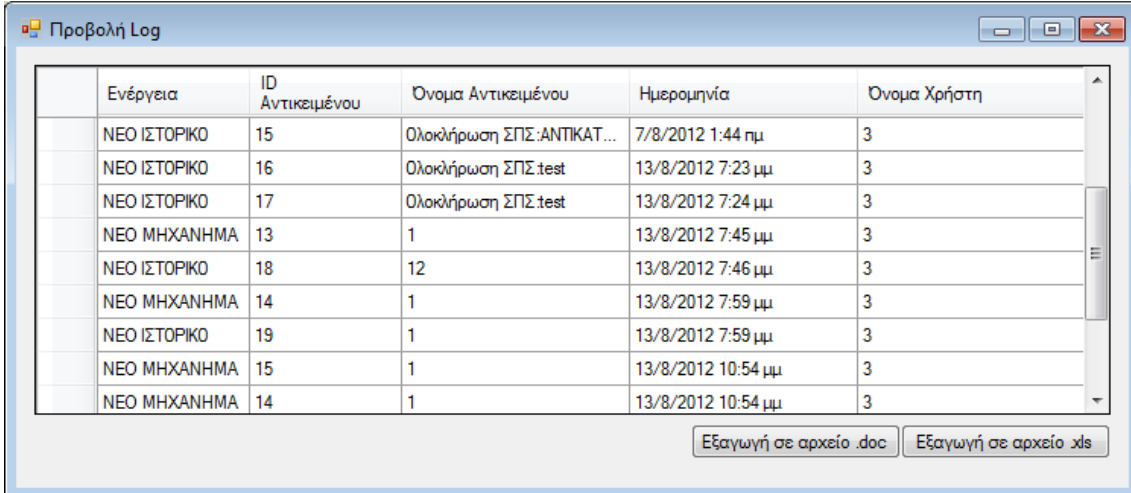
Ένας απλός χρήστης της εφαρμογής (επίπεδο 1 έως 4) μπορεί να χρησιμοποιήσει την οθόνη με τη μορφή που έχει στην εικόνα 4.4 και έχει τη δυνατότητα να αλλάξει κάποια από τα στοιχεία που αφορούν τον ίδιο. Δε μπορεί να αλλάξει το επίπεδο πρόσβασής του και το Όνομα Χρήστη. Επιπλέον δε μπορεί να διαγράψει ή να προσθέσει κάποιο χρήστη. Οι αλλαγές που κάνει ο χρήστης στα στοιχεία του αποθηκεύονται επιλέγοντας το κουμπί «Αποθήκευση αλλαγών» και αναιρούνται επιλέγοντας το κουμπί «Αναίρεση Αλλαγών».



Εικόνα 4-4: Η οθόνη αλλαγής στοιχείων χρήστη για έναν απλό χρήστη της εφαρμογής.

4.1.4 Η οθόνη προβολής κινήσεων των χρηστών

Η οθόνη αυτή είναι διαθέσιμη μόνο στο διαχειριστή της εφαρμογής (επίπεδο πρόσβασης 5) και ενεργοποιείται από το κουμπί «Προβολή κινήσεων χρηστών» της κεντρικής οθόνης της εφαρμογής. Η οθόνη προβάλλει ορισμένες από τις καταχωρήσεις που πραγματοποιούν οι χρήστες της εφαρμογής. Η εφαρμογή τηρεί στατιστικά σχετικά με τις καταχωρήσεις νέων μηχανημάτων και νέων ιστορικών συντήρησης στο σύστημα. Για κάθε καταχώρηση τηρείται ο τύπος της (NEO ΙΣΤΟΡΙΚΟ ή NEO ΜΗΧΑΝΗΜΑ), η ημερομηνία εκτέλεσής της, το Όνομα Χρήστη του χρήστη που την πραγματοποίησε και στοιχεία για το αντικείμενο (Μηχάνημα ή Ιστορικό) που δημιουργήθηκε. Η οθόνη παρέχει δυνατότητες ταξινόμησης ανά πεδίο του πίνακα και επιπλέον εξαγωγή των δεδομένων σε αρχεία τύπου .xls ή .doc μέσω των αντίστοιχων κουμπιών. Στην τρέχουσα έκδοση της εφαρμογής δεν τηρείται πληροφορία για τις υπόλοιπες συναλλαγές που μπορούν να γίνουν με την εφαρμογή. Σε επόμενο στάδιο θα μπορούσε να προστεθεί η αντίστοιχη λειτουργικότητα.



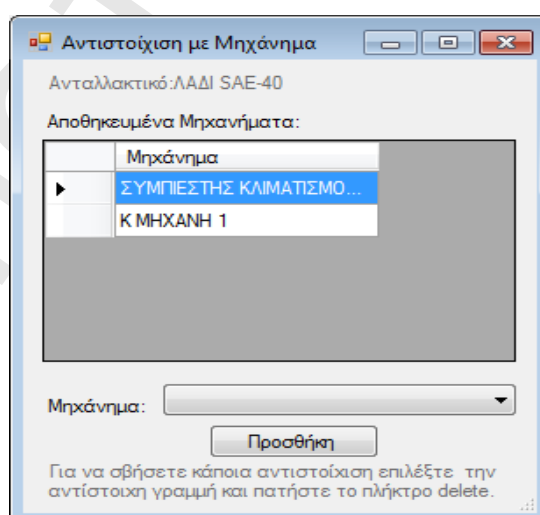
Ενέργεια	ID Αντικειμένου	Όνομα Αντικειμένου	Ημερομηνία	Όνομα Χρήστη
NEO ΙΣΤΟΡΙΚΟ	15	Ολοκλήρωση ΣΠΣ:ΑΝΤΙΚΑΤ...	7/8/2012 1:44 ημ	3
NEO ΙΣΤΟΡΙΚΟ	16	Ολοκλήρωση ΣΠΣ.test	13/8/2012 7:23 μμ	3
NEO ΙΣΤΟΡΙΚΟ	17	Ολοκλήρωση ΣΠΣ.test	13/8/2012 7:24 μμ	3
NEO ΜΗΧΑΝΗΜΑ	13	1	13/8/2012 7:45 μμ	3
NEO ΙΣΤΟΡΙΚΟ	18	12	13/8/2012 7:46 μμ	3
NEO ΜΗΧΑΝΗΜΑ	14	1	13/8/2012 7:59 μμ	3
NEO ΙΣΤΟΡΙΚΟ	19	1	13/8/2012 7:59 μμ	3
NEO ΜΗΧΑΝΗΜΑ	15	1	13/8/2012 10:54 μμ	3
NEO ΜΗΧΑΝΗΜΑ	14	1	13/8/2012 10:54 μμ	3

Εικόνα 4-5: Η σελίδα προβολής των κινήσεων των χρηστών της εφαρμογής.

4.1.5 Η οθόνη αντιστοίχισης ανταλλακτικού με μηχανήμα

Η οθόνη αυτή χρησιμοποιείται για να αντιστοιχίσει ένα ανταλλακτικό με ένα μηχανήμα στα πλαίσια της εφαρμογής. Η αντιστοίχιση υποδηλώνει ότι το εν λόγω ανταλλακτικό μπορεί να χρησιμοποιηθεί σε κάποια εργασία συντήρησης (προγραμματισμένες ή έκτακτης) αυτού του μηχανήματος. Η οθόνη αυτή εμφανίζεται είτε αμέσως μετά τη δημιουργία ενός νέου ανταλλακτικού είτε αν επιλεγεί το κουμπί «Αντιστοίχιση Ανταλλακτικού με Μηχάνημα» για κάποιο ανταλλακτικό. Στη 2^η περίπτωση θα πρέπει στο πεδίο δίπλα από το σχετικό κουμπί να έχει συμπληρωθεί (χειροκίνητα ή αυτόματα) ο κωδικός ενός ανταλλακτικού του συστήματος.

Στο πάνω μέρος της εφαρμογής εμφανίζεται το όνομα του ανταλλακτικού. Στη συνέχεια εμφανίζονται τα μηχανήματα με τα οποία έχει ήδη αντιστοιχηθεί. Ο χρήστης μπορεί να προσθέσει νέα αντιστοίχιση επιλέγοντας ένα μηχανήμα από το σχετικό ComboBox και στη συνέχεια πατώντας το κουμπί «Προσθήκη». Για να διαγράψει μία αντιστοίχιση θα πρέπει να επιλέξει την αντίστοιχη γραμμή και να πιάσει το πλήκτρο Delete από το πληκτρολόγιο. Οι αλλαγές αποθηκεύονται αυτόματα στο σύστημα.

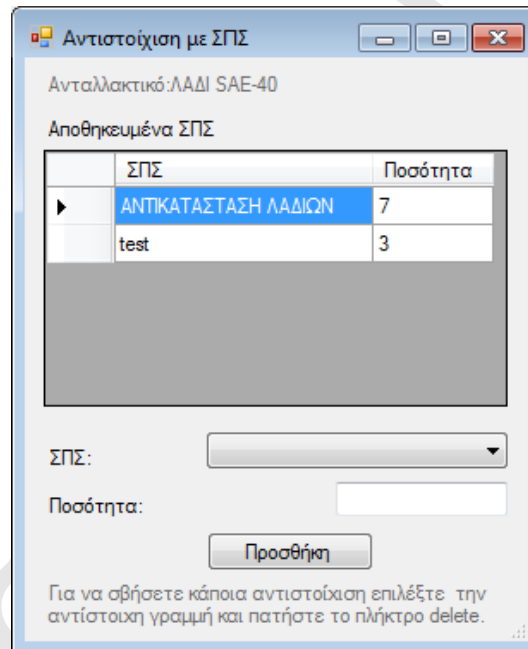


Εικόνα 4-6: Η οθόνη αντιστοίχισης ανταλλακτικού με μηχανήμα.

4.1.6 Η οθόνη αντιστοίχιση ανταλλακτικού με ΣΠΣ

Η οθόνη αυτή χρησιμοποιείται για να αντιστοιχίσει ένα ανταλλακτικό με ένα Σύστημα Προγραμματισμένης Συντήρησης (ΣΠΣ). Η αντιστοίχιση υποδηλώνει ότι το εν λόγω ανταλλακτικό μπορεί να χρησιμοποιηθεί στο συγκεκριμένο ΣΠΣ. Η οθόνη αυτή εμφανίζεται είτε αμέσως μετά τη δημιουργία ενός νέου ανταλλακτικού είτε αν επιλεγεί το κουμπί «Αντιστοίχιση Ανταλλακτικού με ΣΠΣ» για κάποιο ανταλλακτικό. Στη 2^η περίπτωση θα πρέπει στο πεδίο δίπλα από το σχετικό κουμπί να έχει συμπληρωθεί (χειροκίνητα ή αυτόματα) ο κωδικός ενός ανταλλακτικού του συστήματος. Σε κάθε αντιστοίχιση ανταλλακτικού με ΣΠΣ αποθηκεύεται και η απαιτούμενη ποσότητα του ανταλλακτικού για μία ολοκλήρωση του ΣΠΣ.

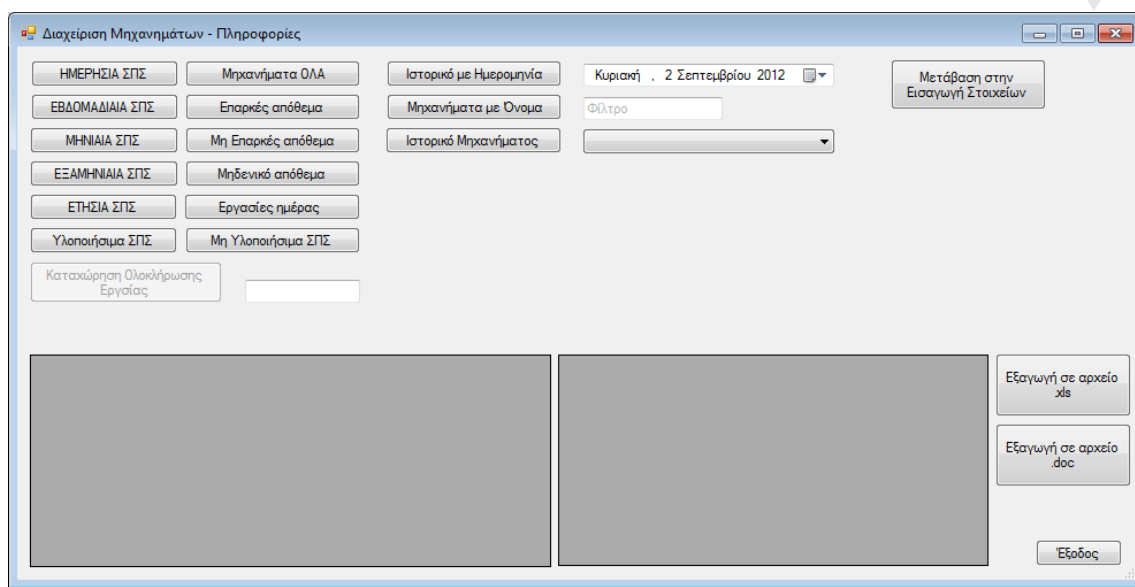
Στο πάνω μέρος της εφαρμογής εμφανίζεται το όνομα του ανταλλακτικού. Στη συνέχεια εμφανίζονται τα μηχανήματα με τα οποία έχει ήδη αντιστοιχηθεί και οι ποσότητες συμμετοχής. Ο χρήστης μπορεί να προσθέσει αντιστοίχιση επιλέγοντας ένα μηχανήμα από το σχετικό ComboBox, ορίζοντας την ποσότητα στο σχετικό πεδίο και πατώντας το κουμπί «Προσθήκη». Για να διαγράψει μία αντιστοίχιση θα πρέπει να επιλέξει την αντίστοιχη γραμμή και να πιάσει το πλήκτρο Delete από το πληκτρολόγιο. Οι αλλαγές αποθηκεύονται αυτόματα στο σύστημα. Δεν υποστηρίζεται αλλαγή της ποσότητας μιας αντιστοίχισης. Αν αυτό καταστεί αναγκαίο θα πρέπει η αντιστοίχιση να διαγραφεί και να δημιουργηθεί νέα με την επιθυμητή ποσότητα.



Εικόνα 4-7: Η οθόνη αντιστοίχισης ανταλλακτικού με ΣΠΣ.

4.1.7 Η οθόνη πληροφοριών

Η οθόνη προβάλλει πληροφορίες που βρίσκονται ήδη αποθηκευμένες στο σύστημα και παρέχει τη δυνατότητα γρήγορης καταχώρησης ολοκλήρωσης μίας προγραμματισμένης εργασίας συντήρησης (ΣΠΣ). Η οθόνη διαθέτει εκτός από κουμπιά εμφάνισης των ζητούμενων πληροφοριών, 2 πλέγματα πληροφοριών (grids). Στο αριστερό πλέγμα εμφανίζονται οι πληροφορίες που ζήτησε ο χρήστης ενώ στο δεξιό εμφανίζονται πληροφορίες σχετικά με κάθε μία από της γραμμές του αριστερού πλέγματος.



Εικόνα 4-8: Η οθόνη πληροφοριών.

Με τα κουμπιά «ΗΜΕΡΗΣΙΑ ΣΠΣ», «ΕΒΔΟΜΑΔΙΑΙΑ ΣΠΣ», «ΜΗΝΙΑΙΑ ΣΠΣ», «ΕΤΗΣΙΑ ΣΠΣ», ο χρήστης μπορεί να δει πληροφορίες σχετικά με τα ΣΠΣ που έχουν καταχωρηθεί στο σύστημα. Οι πληροφορίες εμφανίζονται στο αριστερό πλέγμα, και περιλαμβάνουν τον κωδικό του ΣΠΣ, την περιγραφή του, την περιοδικότητά του και τον κωδικό του μηχανήματος για το οποίο προορίζεται. Αν επιλεγεί κάποια γραμμή του αριστερού πλέγματος πραγματοποιείται έλεγχος αν το συγκεκριμένο ΣΠΣ μπορεί να εκτελεστεί με βάση τα απαιτούμενα ανταλλακτικά και το απόθεμα που υπάρχει σε αυτά την τρέχουσα στιγμή. Αν κάποια ανταλλακτικά αυτού του ΣΠΣ δεν επαρκούν εμφανίζονται στο δεξί πλέγμα μαζί με τον NSN κωδικό και την απαιτούμενη ποσότητα. Επιπλέον, με την επιλογή κάποιας γραμμής του αριστερού πλέγματος συμπληρώνεται αυτόματα το πεδίο για την Καταχώρηση Ολοκλήρωσης ΣΠΣ.

Τα κουμπιά «Υλοποίηση ΣΠΣ» και «Μη Υλοποίηση ΣΠΣ» εμφανίζουν στο αριστερό πλέγμα ΣΠΣ που μπορούν ή δε μπορούν (αντίστοιχα) να εκτελεστούν τη δεδομένη χρονική στιγμή με βάση το απόθεμα ανταλλακτικών. Για τα μη υλοποιημένα ΣΠΣ, αν επιλεγεί κάποια γραμμή του αριστερού πλέγματος εμφανίζονται στο δεξί πλέγμα τα ανταλλακτικά αυτού του ΣΠΣ που δεν επαρκούν μαζί με τον NSN κωδικό και την απαιτούμενη ποσότητα προς παραγγελία. Επιπλέον, με την επιλογή κάποιας γραμμής του αριστερού πλέγματος συμπληρώνεται αυτόματα το πεδίο για την Καταχώρηση Ολοκλήρωσης ΣΠΣ.

Το κουμπί «Εργασίες Ημέρας» εμφανίζει στο αριστερό πλέγμα πληροφορίες για τα ΣΠΣ που πρέπει να ολοκληρωθούν τη συγκεκριμένη μέρα ή έπρεπε να ολοκληρωθούν σε κάποια προηγούμενη μέρα και δεν ολοκληρώθηκαν. Οι πληροφορίες εμφανίζονται στο αριστερό πλέγμα, και περιλαμβάνουν τον κωδικό του ΣΠΣ, την περιγραφή του, την περιοδικότητά του και τον κωδικό του μηχανήματος για το οποίο προορίζεται. Αν επιλεγεί κάποια γραμμή του αριστερού πλέγματος πραγματοποιείται έλεγχος αν το συγκεκριμένο ΣΠΣ μπορεί να εκτελεστεί με βάση τα απαιτούμενα ανταλλακτικά και το απόθεμα που υπάρχει σε αυτά την τρέχουσα στιγμή. Αν κάποια ανταλλακτικά αυτού του ΣΠΣ δεν επαρκούν εμφανίζονται στο δεξί πλέγμα μαζί με τον NSN κωδικό και την απαιτούμενη ποσότητα. Επιπλέον, με την επιλογή κάποιας γραμμής του αριστερού πλέγματος συμπληρώνεται αυτόματα το πεδίο για την Καταχώρηση Ολοκλήρωσης ΣΠΣ.

Τα κουμπιά «Επαρκές απόθεμα», «Μη επαρκές απόθεμα», «Μηδενικό απόθεμα» εμφανίζουν πληροφορίες για ανταλλακτικά ανάλογα με την κατάσταση στην οποία βρίσκεται το απόθεμά τους. Όπως αναφέρθηκε και προηγουμένως για κάθε ανταλλακτικό ορίζεται η απαιτούμενη ποσότητα, η ποσότητα δηλαδή που πρέπει να υπάρχει διαθέσιμη για την ολοκλήρωση των προγραμματισμένων ΣΠΣ όλου του έτους. Η ποσότητα αυτή ανανεώνεται ανάλογα με τα ΣΠΣ στα οποία αντιστοιχεί το εν λόγω

ανταλλακτικό. Επιπλέον τηρείται το τρέχον πραγματικό απόθεμα κάθε ανταλλακτικού. Το κουμπί «Επαρκές Απόθεμα» εμφανίζει στο αριστερό πλέγμα ανταλλακτικά που το απόθεμά τους είναι μεγαλύτερο ή ίσο από την απαιτούμενη ποσότητα. Το κουμπί «Μη Επαρκές Απόθεμα» εμφανίζει στο αριστερό πλέγμα ανταλλακτικά που το απόθεμά τους είναι μικρότερο από την απαιτούμενη ποσότητα. Το κουμπί «Μηδενικό Απόθεμα» εμφανίζει στο αριστερό πλέγμα ανταλλακτικά που το απόθεμά τους είναι μηδενικό. Στο αριστερό πλέγμα εμφανίζονται πληροφορίες το Όνομα, το NSN, η απαιτούμενη και η διαθέσιμη ποσότητα κάθε ανταλλακτικού. Για τα ανταλλακτικά με μη επαρκές ή μηδενικό απόθεμα το δεξί πλέγμα συμπληρώνεται με τις ποσότητες που πρέπει να παραγγελθούν.

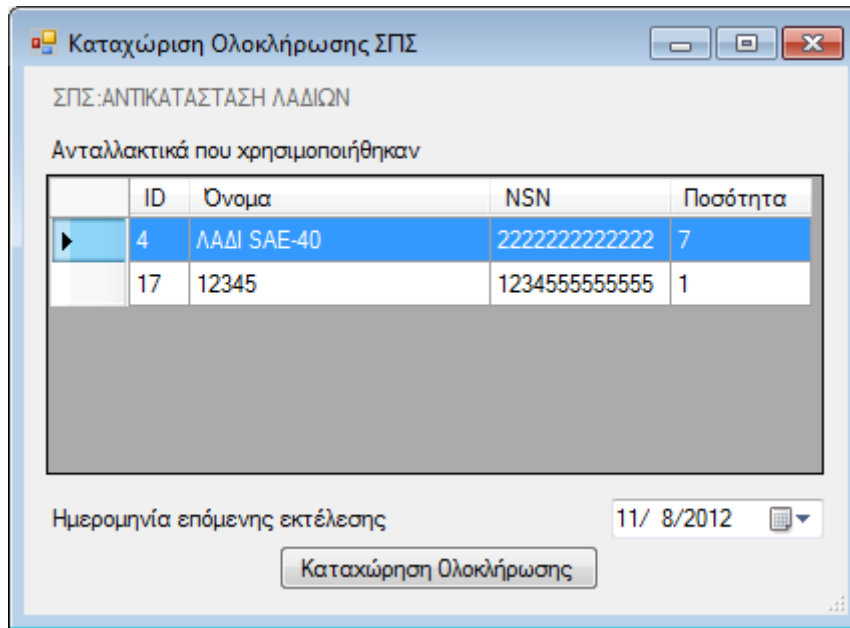
Τα κουμπιά «Μηχανήματα» και «Μηχανήματα με Όνομα» εμφανίζουν πληροφορίες για τα μηχανήματα (Κωδικό, Προγραφή, Ώρες Λειτουργίας, Μέγιστες Ώρες Λειτουργίας) στο αριστερό πλέγμα.

Τα κουμπιά «Ιστορικό με Ημερομηνία» και «Ιστορικό Μηχανήματος» συμπληρώνουν το αριστερό πλέγμα με πληροφορίες σχετικά με Ιστορικό Συντηρήσεων σε συγκεκριμένη ημερομηνία ή για συγκεκριμένο μηχάνημα αντίστοιχα. Για κάθε ιστορικό εμφανίζεται ο κωδικός του Μηχανήματος, ο κωδικός Ιστορικού, ο κωδικός ΣΠΣ (αν έγινε στα πλαίσια κάποιου ΣΠΣ και δεν ήταν έκτακτο), η περιγραφή του και η ημερομηνία εκτέλεσης. Αν επιλεγεί κάποια γραμμή από το αριστερό πλέγμα εμφανίζονται πληροφορίες για το συγκεκριμένο Ιστορικό και για άλλα ΣΠΣ που αφορούν αυτό το Μηχάνημα.

Τα κουμπιά «Εξαγωγή σε αρχείο .xls» και «Εξαγωγή σε αρχείο .doc» αποθηκεύουν τα περιεχόμενα του αριστερού και του δεξιού πλέγματος σε αρχεία αντίστοιχου τύπου. Τα αρχεία αποθηκεύονται αυτόματα στην Επιφάνεια Εργασίας του συστήματος και περιλαμβάνουν στο όνομά τους την ημερομηνία και ώρα δημιουργίας.

4.1.8 Η οθόνη καταχώρησης ολοκλήρωσης ΣΠΣ

Η οθόνη αυτή ενεργοποιείται από το κουμπί «Καταχώρησης Ολοκλήρωσης ΣΠΣ» της οθόνης πληροφοριών και αφού έχει συμπληρωθεί ο κωδικός ενός ΣΠΣ. Στο πάνω μέρος της οθόνης εμφανίζεται το όνομα του ΣΠΣ. Στη συνέχεια εμφανίζονται τα ανταλλακτικά που συμμετείχαν στο ΣΠΣ και οι απαιτούμενες ποσότητες. Αν χρησιμοποιήθηκαν περισσότερα ή λιγότερα ανταλλακτικά ο χρήστης μπορεί να αλλάξει την ποσότητα. Η ημερομηνία επόμενης εκτέλεσης ορίζεται αυτόματα ανάλογα με την περιοδικότητα του ΣΠΣ ώστε να είναι η αμέσως επόμενη από την τελευταία ημερομηνία εκτέλεσης. Ο χρήστης μπορεί αν θέλει να αλλάξει αυτή την ημερομηνία σε μεταγενέστερο χρόνο. Επιλέγοντας το κουμπί «Καταχώρηση Ολοκλήρωσης» καταχωρείται η ολοκλήρωση του ΣΠΣ με τα συμπληρωμένα χαρακτηριστικά. (Λεπτομέρειες για τους ελέγχους και το αποτέλεσμα της Ολοκλήρωσης ΣΠΣ υπάρχουν στο επόμενο Υποκεφάλαιο της αναφοράς).



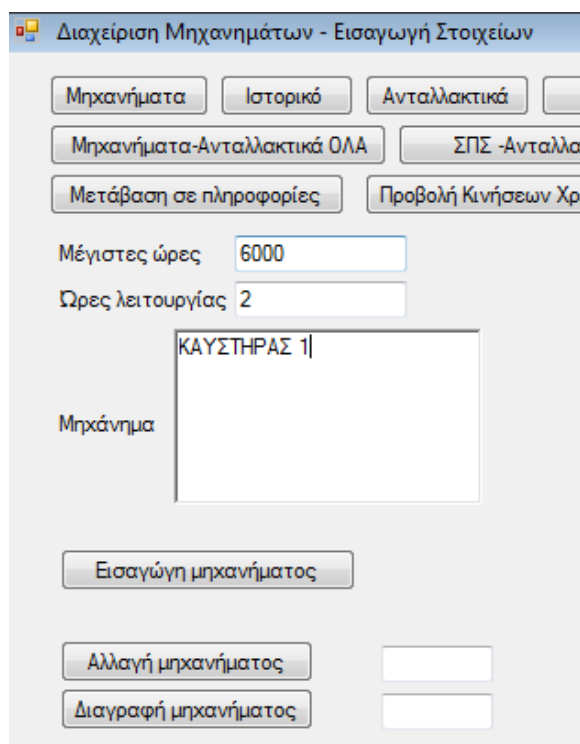
Εικόνα 4-9: Η οθόνη καταχώρισης ολοκλήρωσης ΣΠΣ.

4.2 Σενάρια χρήσης της εφαρμογής

Στην προηγούμενη παράγραφο περιγράφηκαν οι οθόνες της εφαρμογής και οι λειτουργίες που μπορούν να προσφέρουν. Στις επόμενες παραγράφους θα παρουσιαστούν πιο αναλυτικά ορισμένες από τις πιο σημαντικές λειτουργίες της εφαρμογής με σκοπό να καταδειχτεί ότι εκπληρώνονται κατά το μεγαλύτερο μέρος οι προδιαγραφές που τέθηκαν στη φάση σχεδίασης του έργου.

4.2.1 Εισαγωγή, αλλαγή στοιχείων και διαγραφή μηχανήματος.

Η πιο σημαντική οντότητα που διατηρεί η εφαρμογή είναι το μηχάνημα. Για κάθε μηχάνημα αποθηκεύονται στο σύστημα το όνομά του, ένας αυτόματα ανατιθέμενος κωδικός και οι μέγιστες και τρέχουσες ώρες λειτουργίας του. Επιπλέον αποθηκεύονται συσχετίσεις του μηχανήματος με ανταλλακτικά, ΣΠΣ, και εγγραφές ιστορικού.



Εικόνα 4-10: Λειτουργία εισαγωγής μηχανήματος.

Για να εισάγει ο χρήστης μία νέα εγγραφή μηχανήματος επιλέγει το κουμπί «Μηχανήματα» στην κεντρική οθόνη, συμπληρώνει τα πεδία «Μέγιστες Ώρες», «Ώρες Λειτουργίας», «Μηχάνημα» και στη συνέχεια επιλέγει το κουμπί «Δημιουργία Μηχανήματος». Το σύστημα ελέγχει αν έχουν συμπληρωθεί όλα τα απαραίτητα πεδία, αν οι μέγιστες ώρες λειτουργίας είναι μεγαλύτερες από τις τρέχουσες και ότι δεν υπάρχει ήδη εγγραφή μηχανήματος με το ίδιο όνομα. Αν κάποιο λάθος υπάρχει στην εισαγωγή στοιχείων τυπώνεται σχετικό πληροφοριακό μήνυμα σε νέο παράθυρο. Αν η εισαγωγή είναι σωστή δημιουργείται νέα εγγραφή, ανατίθεται μοναδικός κωδικός μηχανήματος (ID) και αποθηκεύεται στη βάση. Εμφανίζεται μήνυμα επιτυχούς καταχώρησης και επιπλέον στο πλέγμα στο κάτω μέρος της οθόνης εμφανίζονται όλες οι εγγραφές μηχανημάτων που πλέον περιλαμβάνουν και το νέο μηχάνημα.

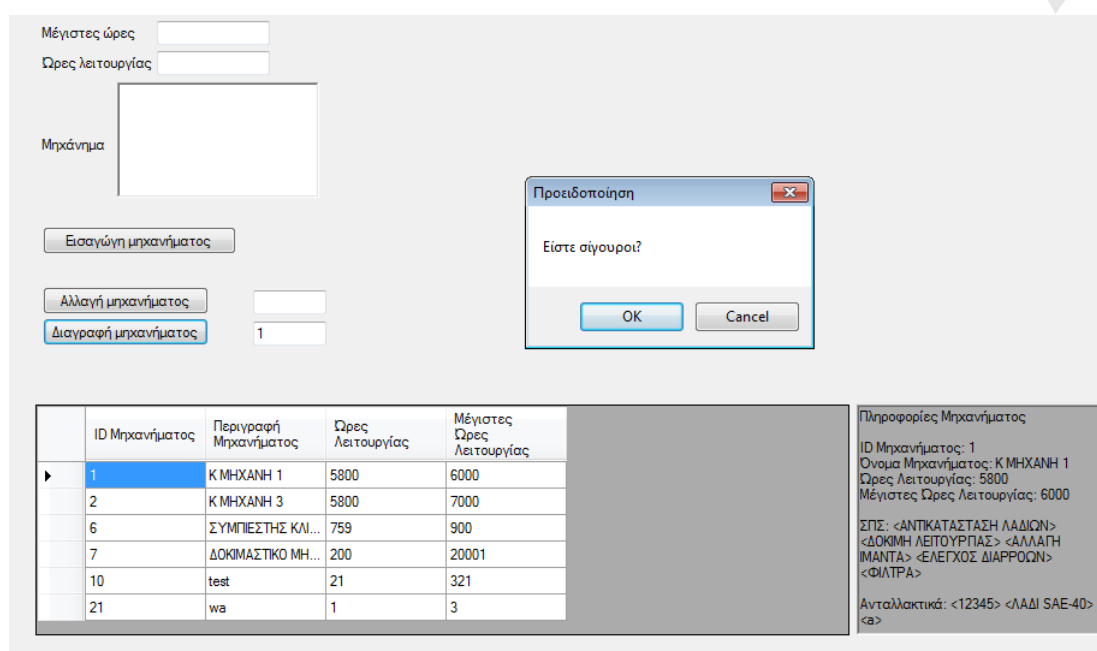
The screenshot shows a software interface for engine management. At the top, there are buttons for 'Μηχανήματα-Ανταλλακτικά ΟΛΑ' and 'ΣΠΣ -Ανταλλακτικά ΟΛΑ'. Below these are buttons for 'Μετάβαση σε πληροφορίες' and 'Προβολή Κινήσεων Χρηστών'. There are three filter boxes labeled 'Φίλτρο'. The main area contains input fields for 'Μέγιστες ώρες' (6000), 'Ωρες λειτουργίας' (5820), and a text box for 'Μηχάνημα' containing 'Κ ΜΗΧΑΝΗ 1'. Below these are buttons for 'Εισαγωγή μηχανήματος', 'Αλλαγή μηχανήματος' (with a value of 1), and 'Διαγραφή μηχανήματος' (with a value of 1). A table at the bottom lists engine data, and a sidebar on the right shows 'Πληροφορίες Μηχανήματος'.

ID Μηχανήματος	Περιγραφή Μηχανήματος	Ωρες Λειτουργίας	Μέγιστες Ωρες Λειτουργίας
1	Κ ΜΗΧΑΝΗ 1	5800	6000
2	Κ ΜΗΧΑΝΗ 3	5800	7000
6	ΣΥΜΠΙΕΣΤΗΣ ΚΛΙ...	759	900
7	ΔΟΚΙΜΑΣΤΙΚΟ ΜΗ...	200	20001
10	test	21	321
21	wa	1	3

Πληροφορίες Μηχανήματος
 ID Μηχανήματος: 1
 Όνομα Μηχανήματος: Κ ΜΗΧΑΝΗ 1
 Ωρες Λειτουργίας: 5800
 Μέγιστες Ωρες Λειτουργίας: 6000
 ΣΠΣ: <ΑΝΤΙΚΑΤΑΣΤΑΣΗ ΛΑΔΙΩΝ>
 <ΔΟΚΙΜΗ ΛΕΙΤΟΥΡΓΙΑΣ> <ΑΛΛΑΓΗ
 ΙΜΑΝΤΑ> <ΕΛΕΓΧΟΣ ΔΙΑΡΡΟΩΝ>
 <ΦΙΛΤΡΑ>
 Ανταλλακτικά: <12345> <ΛΑΔΙ SAE-40>
 <α>

Εικόνα 4-11: Λειτουργία αλλαγής στοιχείων μηχανήματος.

Για να αλλάξει ο χρήστης κάποιες από τις εγγραφές ενός μηχανήματος πρέπει πρώτα να επιλέξει το κουμπί «Μηχανήματα». Στη συνέχεια πρέπει να συμπληρώσει χειροκίνητα ή αυτόματα το πεδίο δίπλα από το κουμπί «Αλλαγή Μηχανήματος». Το πεδίο συμπληρώνεται αυτόματα αν επιλεγεί το κουμπί «Μηχανήματα ΟΛΑ» και στη συνέχεια επιλεγεί μία από τις εγγραφές από το πλέγμα πληροφοριών. Αν ο χρήστης θυμάται τον κωδικό ενός μηχανήματος μπορεί να συμπληρώσει το πεδίο και χειροκίνητα. Αν στο πεδίο συμπληρωθεί έγκυρος κωδικός μηχανήματος τα πεδία «Μέγιστες Ώρες», «Ωρες Λειτουργίας», και «Μηχάνημα» συμπληρώνονται με τις πληροφορίες του συγκεκριμένου μηχανήματος. Ο χρήστης μπορεί να αλλάξει όσες από αυτές θέλει και αφού ολοκληρώσει να επιλέξει «Αλλαγή Μηχανήματος». Το σύστημα εκτελεί ελέγχους σχετικά με τα περιεχόμενα των πεδίων όπως στην εισαγωγή μηχανήματος και εμφανίζει αντίστοιχα μηνύματα. Μετά την αλλαγή, στο πλέγμα στο κάτω μέρος της εφαρμογής εμφανίζονται όλες οι εγγραφές μηχανημάτων που πλέον περιλαμβάνουν τις νέες πληροφορίες.



Εικόνα 4-12: Λειτουργία διαγραφής μηχανήματος.

Με παρόμοιο τρόπο μπορεί να γίνει η διαγραφή ενός μηχανήματος. Ο χρήστης επιλέγει το κουμπί «Μηχάνηματα» από την κεντρική οθόνη και στη συνέχεια συμπληρώνει (αυτόματα ή χειροκίνητα) το πεδίο δίπλα από το κουμπί «Διαγραφή Μηχανήματος». Μόλις ένας έγκυρος κωδικός εισαχθεί, τα πεδία «Μέγιστες Ώρες», «Ώρες Λειτουργίας», και «Μηχάνημα» συμπληρώνονται με τις πληροφορίες του συγκεκριμένου μηχανήματος ώστε ο χρήστης να ελέγξει ότι αυτό είναι το μηχάνημα που επιθυμεί να διαγράψει. Όταν επιλεγεί το κουμπί «Διαγραφή Μηχανήματος» εμφανίζεται μήνυμα επιβεβαίωσης της διαγραφής. Αν ο χρήστης απαντήσει καταφατικά η εγγραφή μηχανήματος διαγράφεται και στο κάτω μέρος της εφαρμογής εμφανίζονται όλες οι εγγραφές μηχανημάτων που πλέον δεν περιλαμβάνουν το διαγραφέν μηχανήμα.

Οι συσχετίσεις του μηχανήματος με ανταλλακτικά, ΣΠΣ, και εγγραφές ιστορικού δημιουργούνται από άλλες λειτουργίες του συστήματος.

4.2.2 Εισαγωγή, αλλαγή στοιχείων και διαγραφή ΣΠΣ.

Πολύ σημαντική έννοια για την εφαρμογή είναι τα Συστήματα Προγραμματισμένης Συντήρησης (ΣΠΣ). Για κάθε ΣΠΣ αποθηκεύονται στο σύστημα το μηχάνημα στο οποίο εκτελείται, η περιγραφή του ΣΠΣ και η περιοδικότητά του. Η περιοδικότητα μπορεί να είναι ημερήσια, εβδομαδιαία, μηνιαία, εξαμηνιαία ή ετήσια. Το μηχάνημα στο οποίο εκτελείται το ΣΠΣ πρέπει να έχει ήδη καταχωρηθεί στο σύστημα. Επίσης στο σύστημα αποθηκεύονται αντιστοιχίσεις του ΣΠΣ με ανταλλακτικά και ποσότητες καθώς επίσης και η προβλεπόμενη ημερομηνία επόμενης εκτέλεσής του.

Για να εισάγει ο χρήστης μία νέα εγγραφή ΣΠΣ επιλέγει το κουμπί «ΣΠΣ» στην κεντρική οθόνη, επιλέγει το μηχάνημα και την περιοδικότητα από τα αντίστοιχα κουμπιά επιλογών, γράφει την περιγραφή του ΣΠΣ και στη συνέχεια επιλέγει το κουμπί «Δημιουργία ΣΠΣ». Το σύστημα ελέγχει αν έχουν συμπληρωθεί όλα τα απαραίτητα πεδία και ότι δεν υπάρχει ήδη εγγραφή ΣΠΣ με το ίδιο όνομα. Αν κάποιο λάθος υπάρχει στην εισαγωγή στοιχείων τυπώνεται σχετικό πληροφοριακό μήνυμα σε νέο παράθυρο. Αν η εισαγωγή είναι σωστή δημιουργείται νέα εγγραφή, ανατίθεται μοναδικός κωδικός ΣΠΣ (ID) και αποθηκεύεται στη βάση. Εμφανίζεται μήνυμα επιτυχούς καταχώρησης και επιπλέον στο πλέγμα στο κάτω μέρος της οθόνης εμφανίζονται όλες οι εγγραφές ΣΠΣ που πλέον περιλαμβάνουν και το νέο ΣΠΣ.

Εικόνα 4-13: Λειτουργία εισαγωγής ΣΠΣ.

Για να αλλάξει ο χρήστης κάποια από τα στοιχεία ενός ΣΠΣ πρέπει πρώτα να επιλέξει το κουμπί «ΣΠΣ». Στη συνέχεια πρέπει να συμπληρώσει χειροκίνητα ή αυτόματα το πεδίο δίπλα από το κουμπί «Αλλαγή ΣΠΣ». Το πεδίο συμπληρώνεται αυτόματα αν επιλεγεί το κουμπί «ΣΠΣ ΟΛΑ» και στη συνέχεια επιλεγεί μία από τις εγγραφές από το πλέγμα πληροφοριών. Αν ο χρήστης θυμάται τον κωδικό ενός ΣΠΣ μπορεί να συμπληρώσει το πεδίο και χειροκίνητα. Αν στο πεδίο συμπληρωθεί έγκυρος κωδικός ΣΠΣ τα πεδία «Μηχάνημα», «ΣΠΣ» και «Περιοδικότητα» συμπληρώνονται με τις πληροφορίες του συγκεκριμένου ΣΠΣ. Ο χρήστης μπορεί να αλλάξει όσες από αυτές θέλει και αφού ολοκληρώσει να επιλέξει «Αλλαγή ΣΠΣ». Το σύστημα εκτελεί ελέγχους σχετικά με τα περιεχόμενα των πεδίων όπως στην εισαγωγή ΣΠΣ και εμφανίζει αντίστοιχα μηνύματα. Μετά την αλλαγή, στο πλέγμα στο κάτω μέρος της εφαρμογής εμφανίζονται όλες οι εγγραφές ΣΠΣ που πλέον περιλαμβάνουν τις νέες πληροφορίες.

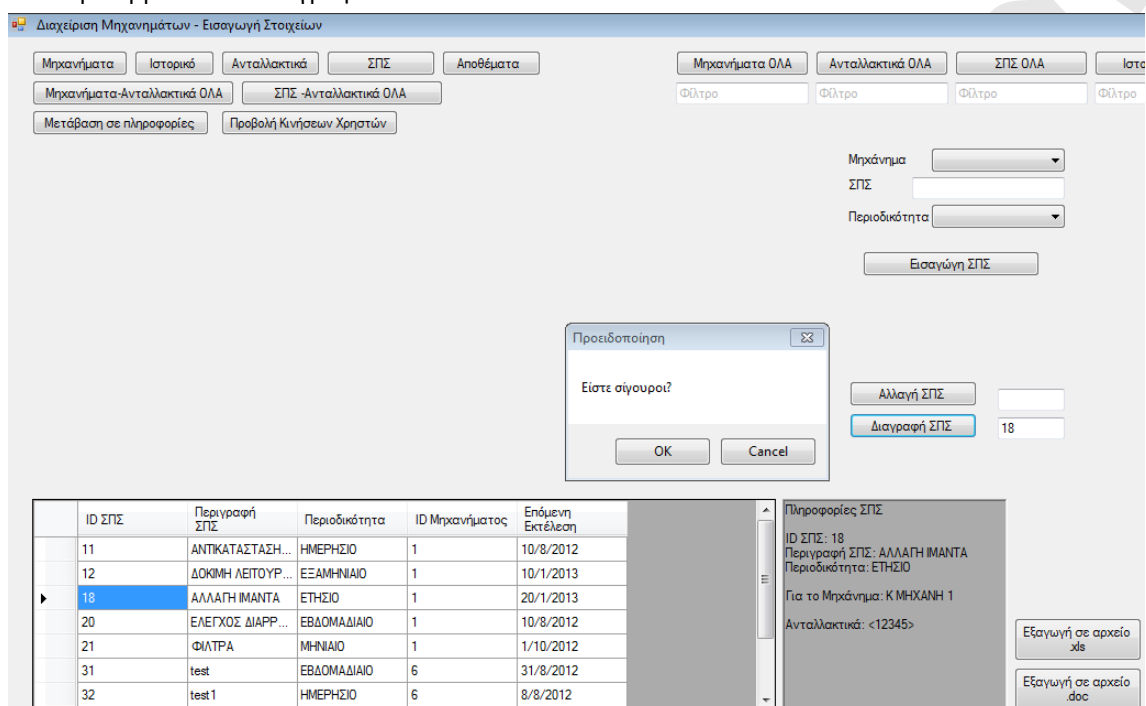
ID ΣΠΣ	Περιγραφή ΣΠΣ	Περιοδικότητα	ID Μηχανήματος	Επόμενη Εκτέλεση
11	ΑΝΤΙΚΑΤΑΣΤΑΣΗ...	ΗΜΕΡΗΣΙΟ	1	10/8/2012
12	ΔΟΚΙΜΗ ΛΕΙΤΟΥΡ...	ΕΞΑΜΗΝΙΑΙΟ	1	10/1/2013
18	ΑΛΛΑΓΗ ΙΜΑΝΤΑ	ΕΤΗΣΙΟ	1	20/1/2013
20	ΕΛΕΓΧΟΣ ΔΙΑΡΡ...	ΕΒΔΟΜΑΔΙΑΙΟ	1	10/8/2012
21	ΦΙΛΤΡΑ	ΜΗΝΙΑΙΟ	1	1/10/2012
31	test	ΕΒΔΟΜΑΔΙΑΙΟ	6	31/8/2012
32	test1	ΗΜΕΡΗΣΙΟ	6	8/8/2012

Πληροφορίες ΣΠΣ
 ID ΣΠΣ: 18
 Περιγραφή ΣΠΣ: ΑΛΛΑΓΗ ΙΜΑΝΤΑ
 Περιοδικότητα: ΕΤΗΣΙΟ
 Για το Μηχάνημα: Κ ΜΗΧΑΝΗ 1
 Ανταλλακτικά: <12345>

Εικόνα 4-14: Λειτουργία αλλαγής στοιχείων ΣΠΣ.

Με παρόμοιο τρόπο μπορεί να γίνει η διαγραφή ενός ΣΠΣ. Ο χρήστης επιλέγει το κουμπί «ΣΠΣ» από την κεντρική οθόνη και στη συνέχεια συμπληρώνει (αυτόματα ή χειροκίνητα) το πεδίο δίπλα από το κουμπί «Διαγραφή ΣΠΣ». Μόλις ένας έγκυρος κωδικός εισαχθεί, τα πεδία Μηχάνημα», «ΣΠΣ» και Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP

«Περιοδικότητα» συμπληρώνονται με τις πληροφορίες του συγκεκριμένου ΣΠΣ ώστε ο χρήστης να ελέγξει ότι αυτό είναι το ΣΠΣ που επιθυμεί να διαγράψει. Όταν επιλεγεί το κουμπί «Διαγραφή ΣΠΣ» εμφανίζεται μήνυμα επιβεβαίωσης της διαγραφής. Αν ο χρήστης απαντήσει καταφατικά η εγγραφή ΣΠΣ διαγράφεται και στο κάτω μέρος της εφαρμογής εμφανίζονται όλες οι εγγραφές ΣΠΣ που πλέον δεν περιλαμβάνουν το διαγραφέν ΣΠΣ.



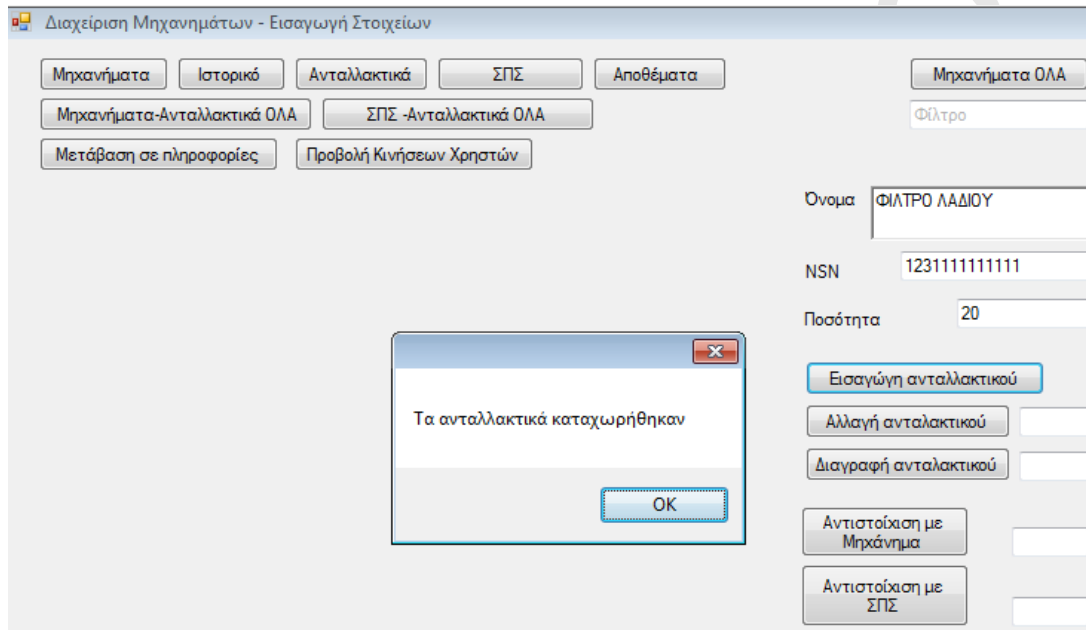
Εικόνα 4-15: Λειτουργία διαγραφής ΣΠΣ.

Η συσχέτιση κάθε ΣΠΣ με τα ανταλλακτικά που μπορούν να χρησιμοποιηθούν για αυτό γίνεται από το αντίστοιχο μενού για την εισαγωγή και αλλαγή στοιχείων των ανταλλακτικών(βλ 4.2.3). Η ημερομηνία επόμενης εκτέλεσης του ΣΠΣ δημιουργείται κατά την εισαγωγή του με βάση την περιοδικότητά του και ανανεώνεται κάθε φορά που καταχωρείται μία ολοκλήρωση του ΣΠΣ. Τα εξαμηνιαία ΣΠΣ ορίζονται προς εκτέλεση στις 10/7 και 10/1 κάθε έτους μηνιαία την 1η κάθε μήνα, α εβδομαδιαία κάθε Παρασκευή και τα ετήσια στις 20/1 κάθε έτους.

4.2.3 Εισαγωγή, αλλαγή στοιχείων και διαγραφή ανταλλακτικού

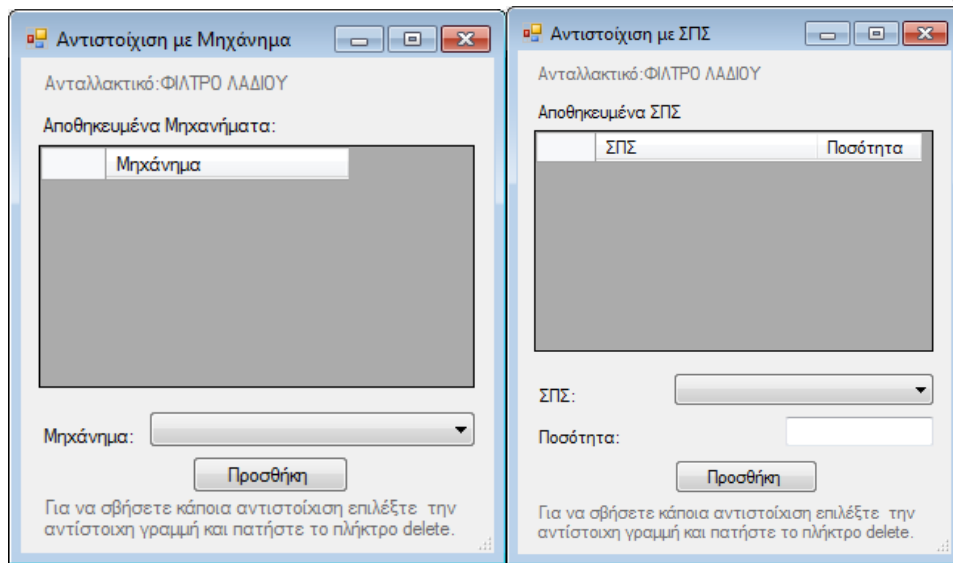
Σημαντικό ρόλο στην εφαρμογή παίζουν οι εγγραφές ανταλλακτικών. Για κάθε ανταλλακτικό αποθηκεύονται στο σύστημα το όνομά του, ένας αυτόματα ανατιθέμενος κωδικός, ένας χειροκίνητα ανατιθέμενος αλφαριθμητικός κωδικός (NSN) και η απαιτούμενη ποσότητά του. Επιπλέον αποθηκεύονται συσχετίσεις του ανταλλακτικού με αποθέματα, μηχανήματα και ΣΠΣ. Η αντιστοίχιση του ανταλλακτικού με μηχανήματα και ΣΠΣ αποθηκεύεται σε ξεχωριστούς πίνακες της βάσης και πραγματοποιείται μέσω των αντίστοιχων οθονών αντιστοίχισης.(βλ. 4.1.5 και 4.1.6). Το τρέχον απόθεμα των ανταλλακτικών που διατηρούνται στο σύστημα διατηρείται σε διαφορετικό πίνακα της βάσης δεδομένων. Αυτή η επιλογή έγινε για να υπάρχει μεγαλύτερη ευελιξία και η δυνατότητα διατήρησης αυτής της πληροφορίας σε διαφορετική βάση στο μέλλον. Για κάθε εγγραφή αποθέματος διατηρείται ένας αυτόματα ανατιθέμενος κωδικός, η ποσότητα του αποθέματος και ο κωδικός NSN. Κάθε εγγραφή αποθέματος αντιστοιχεί σε ένα συγκεκριμένο ανταλλακτικό και η αντιστοίχιση γίνεται μέσω του κωδικού NSN.

Για να εισάγει ο χρήστης μία νέα εγγραφή ανταλλακτικού επιλέγει το κουμπί «Ανταλλακτικά» στην κεντρική οθόνη, συμπληρώνει τα πεδία «Όνομα», «NSN», και «Ποσότητα». και στη συνέχεια επιλέγει το κουμπί «Δημιουργία Ανταλλακτικού». Το σύστημα ελέγχει αν έχουν συμπληρωθεί όλα τα απαραίτητα πεδία και ότι δεν υπάρχει ήδη εγγραφή ανταλλακτικού με το ίδιο όνομα. Αν κάποιο λάθος υπάρχει στην εισαγωγή στοιχείων τυπώνεται σχετικό πληροφοριακό μήνυμα σε νέο παράθυρο. Αν η εισαγωγή είναι σωστή δημιουργείται νέα εγγραφή ανταλλακτικού, ανατίθεται μοναδικός κωδικός (ID) και αποθηκεύεται στη βάση. Στη συνέχεια εμφανίζεται μήνυμα επιτυχούς καταχώρησης και ο χρήστης ενημερώνεται ότι δημιουργήθηκε νέα εγγραφή αποθέματος για αυτό το ανταλλακτικό με αρχική ποσότητα 0.



Εικόνα 4-16: Λειτουργία εισαγωγής ανταλλακτικού.

Κατόπιν ενεργοποιούνται με τη σειρά οι οθόνες αντιστοίχισης του ανταλλακτικού με μηχανήμα και ΣΠΣ. Ο χρήστης κάνει όσες αντιστοιχίσεις επιθυμεί. Αν το ανταλλακτικό αντιστοιχηθεί με ΣΠΣ η απαιτούμενη ποσότητά του ανανεώνεται ώστε να καλύπτεται η εκτέλεση όλων των ΣΠΣ που συμμετέχει μέχρι το τέλος του έτους. Μετά την ολοκλήρωση της εισαγωγής εμφανίζεται μήνυμα επιτυχούς καταχώρησης και επιπλέον στο πλέγμα στο κάτω μέρος της οθόνης εμφανίζονται όλες οι εγγραφές ανταλλακτικών που πλέον περιλαμβάνουν και το νέο ανταλλακτικό.



Εικόνα 4-17: Λειτουργία αντιστοίχισης ανταλλακτικού με Μηχάνημα και ΣΠΣ.

Για να αλλάξει ο χρήστης κάποιες από τις εγγραφές ενός ανταλλακτικού πρέπει πρώτα να επιλέξει το κουμπί «Ανταλλακτικά». Στη συνέχεια πρέπει να συμπληρώσει χειροκίνητα ή αυτόματα το πεδίο δίπλα από το κουμπί «Αλλαγή Ανταλλακτικού». Το πεδίο συμπληρώνεται αυτόματα αν επιλεγεί το κουμπί «Ανταλλακτικά ΟΛΑ» και στη συνέχεια επιλεγεί μία από τις εγγραφές από το πλέγμα πληροφοριών. Αν ο χρήστης θυμάται τον κωδικό ενός ανταλλακτικού μπορεί να συμπληρώσει το πεδίο και χειροκίνητα. Αν στο πεδίο συμπληρωθεί έγκυρος κωδικός ανταλλακτικού τα πεδία «Όνομα», «NSN», και «Ποσότητα» συμπληρώνονται με τις πληροφορίες του συγκεκριμένου ανταλλακτικού. Ο χρήστης μπορεί να αλλάξει όσες από αυτές θέλει και αφού ολοκληρώσει να επιλέξει «Αλλαγή Ανταλλακτικού». Το σύστημα εκτελεί ελέγχους σχετικά με τα περιεχόμενα των πεδίων όπως στην εισαγωγή ανταλλακτικού και εμφανίζει αντίστοιχα μηνύματα. Μετά την αλλαγή, στο πλέγμα στο κάτω μέρος της εφαρμογής εμφανίζονται όλες οι εγγραφές ανταλλακτικών που πλέον περιλαμβάνουν τις νέες πληροφορίες. Ο χρήστης μπορεί να αλλάξει τις αντιστοιχίσεις ανταλλακτικών με μηχανήματα ή ΣΠΣ επιλέγοντας τα αντίστοιχα κουμπιά και συμπληρώνοντας τις θρόνες που ενεργοποιούν.

Διαχείριση Μηχανημάτων - Εισαγωγή Στοιχείων

Μηχανήματα Ιστορικό Ανταλλακτικά ΣΠΣ Αποθέματα

Μηχανήματα-Ανταλλακτικά ΟΛΑ ΣΠΣ -Ανταλλακτικά ΟΛΑ

Μετάβαση σε πληροφορίες Προβολή Κινήσεων Χρηστών

Μηχανήματα ΟΛΑ Ανταλλακτικά ΟΛΑ ΣΠΣ

Φίλτρο Φίλτρο Φίλτρο

Όνομα

NSN

Ποσότητα

Εισαγωγή ανταλλακτικού

Αλλαγή ανταλλακτικού

Διαγραφή ανταλλακτικού

Αντιστοίχιση με Μηχάνημα

Αντιστοίχιση με ΣΠΣ

ID Ανταλλακτικού	Όνομα Ανταλλακτικού	NSN	Ποσότητα
1	ΚΥΛΙΝΔΡΟΚΕΦΑ...	5555555555555	3
2	ΚΑΥΣΤΗΡΑΣ	3333333333333	10
3	ΦΛΑΝΤΖΑ	1111111111111	12
4	ΛΑΔΙ SAE-40	2222222222222	984
6	test part 3107	7777777777777	20
19	ΦΙΛΤΡΟ ΛΑΔΙΟΥ	1231111111111	20

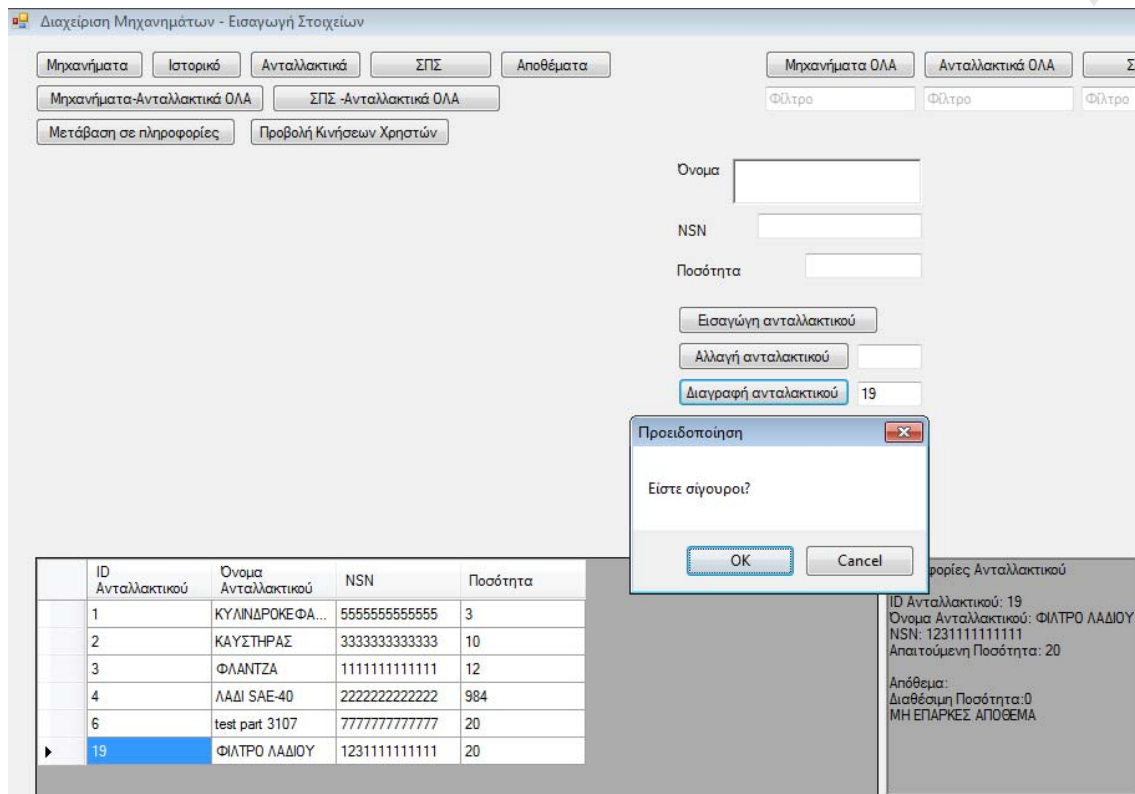
Πληροφορίες Ανταλλακτικού

ID Ανταλλακτικού: 19
Όνομα Ανταλλακτικού: ΦΙΛΤΡΟ ΛΑΔΙΟΥ
NSN: 1231111111111
Απαιτούμενη Ποσότητα: 20

Απόθεμα:
Διαθέσιμη Ποσότητα: 0
ΜΗ ΕΠΑΡΚΕΣ ΑΠΟΘΕΜΑ

Εικόνα 4-18: Λειτουργία αλλαγής στοιχείων ανταλλακτικού.

Με παρόμοιο τρόπο μπορεί να γίνει η διαγραφή ενός ανταλλακτικού. Ο χρήστης επιλέγει το κουμπί «Ανταλλακτικά» από την κεντρική οθόνη και στη συνέχεια συμπληρώνει (αυτόματα ή χειροκίνητα) το πεδίο δίπλα από το κουμπί «Διαγραφή Ανταλλακτικού». Μόλις ένας έγκυρος κωδικός εισαχθεί, τα πεδία «Όνομα», «NSN», και «Ποσότητα» συμπληρώνονται με τις πληροφορίες του συγκεκριμένου ανταλλακτικού ώστε ο χρήστης να ελέγξει ότι αυτό είναι το ανταλλακτικό που επιθυμεί να διαγράψει. Όταν επιλεγεί το κουμπί «Διαγραφή Ανταλλακτικού» εμφανίζεται μήνυμα επιβεβαίωσης της διαγραφής. Αν ο χρήστης απαντήσει καταφατικά η εγγραφή ανταλλακτικού διαγράφεται και στο κάτω μέρος της εφαρμογής εμφανίζονται όλες οι εγγραφές ανταλλακτικών που πλέον δεν περιλαμβάνουν το διαγραφέν ανταλλακτικό.



Εικόνα 4-19: Λειτουργία διαγραφής ανταλλακτικού.

4.2.4 Εισαγωγή, αλλαγή στοιχείων και διαγραφή ιστορικού

Κάθε ολοκλήρωση εργασίας συντήρησης, προγραμματισμένης (που αντιστοιχεί σε ΣΠΣ) ή έκτακτης, καταχωρείται ως ιστορικό στο σύστημα. Η καταχώρηση ιστορικού για ολοκλήρωση εργασίας που αντιστοιχεί με ΣΠΣ μπορεί να γίνει με την επιλογή Καταχώρησης Ολοκλήρωσης ΣΠΣ (βλ. 4.2.7). Υπάρχει όμως η δυνατότητα, ο χρήστης να εισάγει νέο ιστορικό, που αντιστοιχεί σε ΣΠΣ ή όχι, από την κεντρική οθόνη της εφαρμογής. Επιπλέον παρέχεται η δυνατότητα αλλαγής των στοιχείων μιας εγγραφής ιστορικού ή διαγραφής της. Για κάθε εγγραφή ιστορικού αποθηκεύονται στο σύστημα η περιγραφή της, ένας αυτόματα ανατιθέμενος κωδικός, το μηχάνημα στο οποίο έγινε, το ΣΠΣ στο οποίο αντιστοιχεί (αν είναι προγραμματισμένη και όχι έκτακτη) και η ημερομηνία εκτέλεσής της.

Για να εισάγει ο χρήστης μία νέα εγγραφή ιστορικού επιλέγει το κουμπί «Ιστορικό» στην κεντρική οθόνη, συμπληρώνει τα πεδία «Μηχάνημα», «ΣΠΣ», «Ημερομηνία», «Ιστορικό και στη συνέχεια επιλέγει το κουμπί «Δημιουργία Ιστορικού». Η ημερομηνία τίθεται στην τρέχουσα ημερομηνία αλλά μπορεί να αλλάξει από το χρήστη. Το πεδίο ΣΠΣ μπορεί να μείνει κενό υποδεικνύοντας πως η εν λόγω εργασία δεν έγινε στα πλαίσια κάποιου ΣΠΣ. Το σύστημα ελέγχει αν έχουν συμπληρωθεί όλα τα απαραίτητα πεδία. Αν κάποιο λάθος υπάρχει στην εισαγωγή στοιχείων τυπώνεται σχετικό πληροφοριακό μήνυμα σε νέο παράθυρο. Αν η εισαγωγή είναι σωστή δημιουργείται νέα εγγραφή, ανατίθεται μοναδικός κωδικός ιστορικού (ID) και αποθηκεύεται στη βάση. Εμφανίζεται μήνυμα επιτυχούς καταχώρησης και επιπλέον στο πλέγμα στο κάτω μέρος της οθόνης εμφανίζονται όλες οι εγγραφές ιστορικού που πλέον περιλαμβάνουν και το νέο ιστορικό.

Διαχείριση Μηχανημάτων - Εισαγωγή Στοιχείων

Μηχανήματα Ιστορικό Ανταλλακτικά ΣΠΣ Αποθέματα

Μηχανήματα-Ανταλλακτικά ΟΛΑ ΣΠΣ -Ανταλλακτικά ΟΛΑ

Μετάβαση σε πληροφορίες Προβολή Κινήσεων Χρηστών

Μηχάνημα Κ ΜΗΧΑΝΗ 1

ΣΠΣ

Ημερομηνία 9/2012

Ιστορικό ΕΚΤΑΚΤΗ ΕΠΙΣΚΕΥΗ Κ. ΜΗΧΑΝΗΣ

Εισαγωγή ιστορικού

Αλλαγή ιστορικού

Διαγραφή ιστορικού

Εικόνα 4-20: Λειτουργία εισαγωγής ιστορικού.

Για να αλλάξει ο χρήστης κάποιες από τα στοιχεία ενός ιστορικού πρέπει πρώτα να επιλέξει το κουμπί «Ιστορικό». Στη συνέχεια πρέπει να συμπληρώσει χειροκίνητα ή αυτόματα το πεδίο δίπλα από το κουμπί «Αλλαγή Ιστορικού». Το πεδίο συμπληρώνεται αυτόματα αν επιλεγεί το κουμπί «Ιστορικό ΟΛΑ» και στη συνέχεια επιλεγεί μία από τις εγγραφές από το πλέγμα πληροφοριών. Αν ο χρήστης θυμάται τον κωδικό μιας εγγραφής ιστορικού μπορεί να συμπληρώσει το πεδίο και χειροκίνητα. Αν στο πεδίο συμπληρωθεί έγκυρος κωδικός ιστορικού τα πεδία «Μηχάνημα», «ΣΠΣ», «Ημερομηνία» και «Ιστορικό» συμπληρώνονται με τις πληροφορίες του συγκεκριμένου ιστορικού. Ο χρήστης μπορεί να αλλάξει όσες από αυτές θέλει και αφού ολοκληρώσει να επιλέξει «Αλλαγή Ιστορικού». Το σύστημα εκτελεί ελέγχους σχετικά με τα περιεχόμενα των πεδίων όπως στην εισαγωγή ιστορικού και εμφανίζει αντίστοιχα μηνύματα. Μετά την αλλαγή, στο πλέγμα στο κάτω μέρος της εφαρμογής εμφανίζονται όλες οι εγγραφές ιστορικού που πλέον περιλαμβάνουν τις νέες πληροφορίες.

Διαχείριση Μηχανημάτων - Εισαγωγή Στοιχείων

Μηχανήματα Ιστορικό Ανταλλακτικά ΣΠΣ Αποθέματα

Μηχανήματα-Ανταλλακτικά ΟΛΑ ΣΠΣ-Ανταλλακτικά ΟΛΑ

Μηχανήματα ΟΛΑ Ανταλλακτικά ΟΛΑ ΣΠΣ

Μετάβαση σε πληροφορίες Προβολή Κινήσεων Χρηστών

Μηχάνημα Κ ΜΗΧΑΝΗ 1

ΣΠΣ

Ημερομηνία 7/ 9/2012

ΕΚΤΑΚΤΗ

Δευ	Τρι	Τετ	Πεμ	Παρ	Σαβ	Κυρ
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Today: 9/9/2012

Εισαγωγή ιστορικού

Αλλαγή ιστορικού

Διαγραφή ιστορικού 23

ID Ιστορικού	ID Μηχανήματος	ID ΣΠΣ	Περιγραφή	Ημερομηνία
14	1	12	sfd	31/7/2012
15	1	11	Ολοκλήρωση ΣΠ...	7/8/2012
16	6	31	Ολοκλήρωση ΣΠ...	13/8/2012
17	6	31	Ολοκλήρωση ΣΠ...	13/8/2012
21	6	31	Ολοκλήρωση ΣΠ...	14/8/2012
22	1	21	Ολοκλήρωση ΣΠ...	2/9/2012
23	1	-1	ΕΚΤΑΚΤΗ ΕΠΙΣΚΕ...	7/9/2012

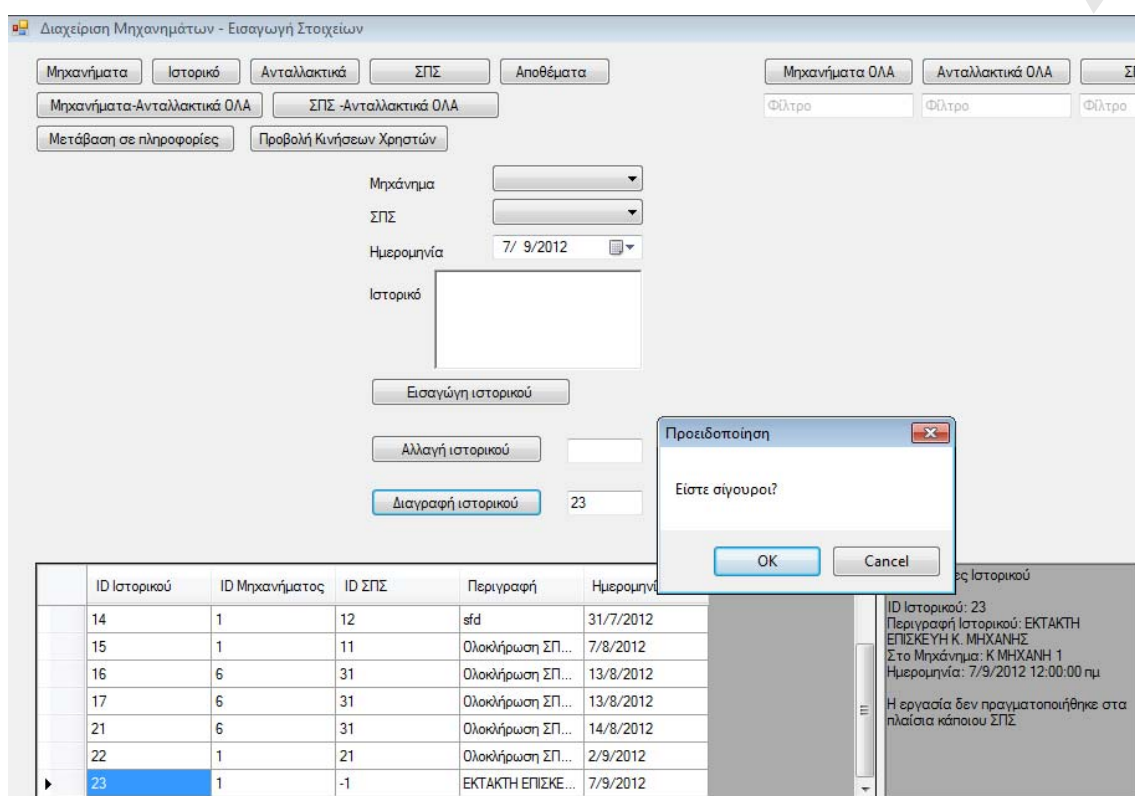
Πληροφορίες Ιστορικού

ID Ιστορικού: 23
 Περιγραφή Ιστορικού: ΕΚΤΑΚΤΗ ΕΠΙΣΚΕΥΗ Κ. ΜΗΧΑΝΗΣ
 Στο Μηχάνημα: Κ ΜΗΧΑΝΗ 1
 Ημερομηνία: 7/9/2012 12:00:00 ημ

Η εργασία δεν πραγματοποιήθηκε στα πλαίσια κάποιου ΣΠΣ

Εικόνα 4-21: Λειτουργία αλλαγής ιστορικού.

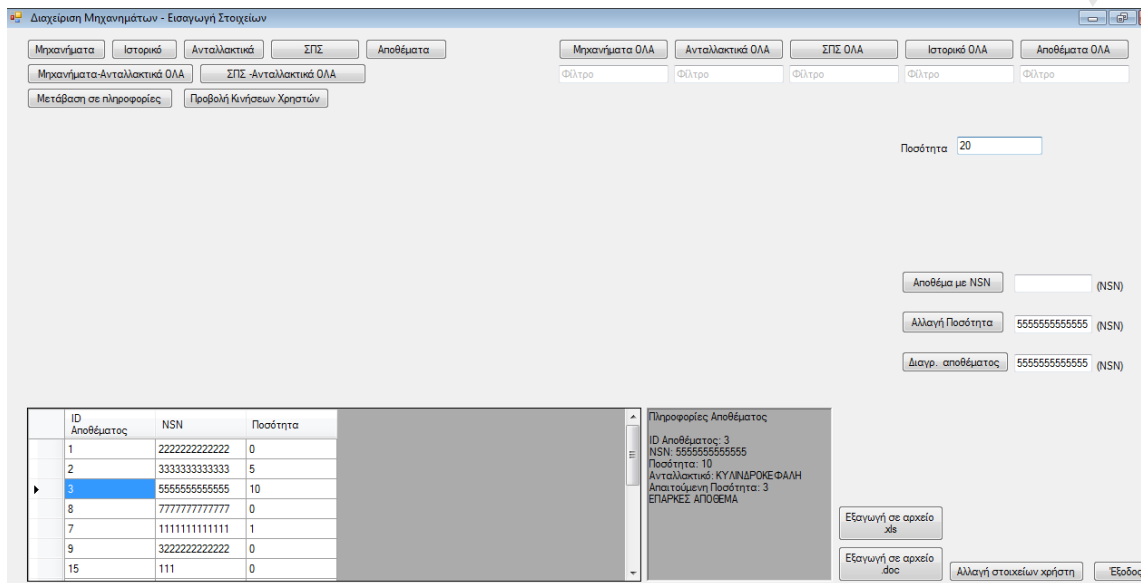
Με παρόμοιο τρόπο μπορεί να γίνει η διαγραφή ενός ιστορικού. Ο χρήστης επιλέγει το κουμπί «Ιστορικό» από την κεντρική οθόνη και στη συνέχεια συμπληρώνει (αυτόματα ή χειροκίνητα) το πεδίο δίπλα από το κουμπί «Διαγραφή Ιστορικού». Μόλις ένας έγκυρος κωδικός εισαχθεί, τα πεδία «Μηχάνημα», «ΣΠΣ», «Ημερομηνία» και «Ιστορικό» συμπληρώνονται με τις πληροφορίες του συγκεκριμένου ιστορικού ώστε ο χρήστης να ελέγξει ότι αυτό είναι το ιστορικό που επιθυμεί να διαγράψει. Όταν επιλεγεί το κουμπί «Διαγραφή Ιστορικού» εμφανίζεται μήνυμα επιβεβαίωσης της διαγραφής. Αν ο χρήστης απαντήσει καταφατικά η εγγραφή μηχανήματος διαγράφεται και στο κάτω μέρος της εφαρμογής εμφανίζονται όλες οι εγγραφές ιστορικού που πλέον δεν περιλαμβάνουν το διαγραφέν ιστορικό.



Εικόνα 4-22: Λειτουργία διαγραφής ιστορικού.

4.2.5 Αλλαγή και διαγραφή αποθέματος ανταλλακτικού.

Το τρέχον απόθεμα των ανταλλακτικών που διατηρούνται στο σύστημα διατηρείται σε διαφορετικό πίνακα της βάσης δεδομένων. Αυτή η επιλογή έγινε για να υπάρχει μεγαλύτερη ευελιξία και η δυνατότητα διατήρησης αυτής της πληροφορίας σε διαφορετική βάση στο μέλλον. Για κάθε εγγραφή αποθέματος διατηρείται ένας αυτόματα ανατιθέμενος κωδικός, η ποσότητα του αποθέματος και ο κωδικός NSN. Κάθε εγγραφή αποθέματος αντιστοιχεί σε ένα συγκεκριμένο ανταλλακτικό και η αντιστοίχιση γίνεται μέσω του κωδικού NSN. Όπως προαναφέρθηκε, κάθε φορά που εισάγεται νέο ανταλλακτικό δημιουργείται μία νέα εγγραφή αποθέματος με το ίδιο NSN. Με τη δημιουργία νέας εγγραφής αποθέματος η ποσότητά του τίθεται σε 0. Ο χρήστης μπορεί να αλλάξει την ποσότητα ενός αποθέματος ή να διαγράψει μία εγγραφή αποθέματος επιλέγοντας το κουμπί «Απόθεμα» στην κεντρική οθόνη της εφαρμογής. Ο χρήστης μπορεί να αναζητήσει απόθεμα με βάση το NSN. Μπορεί να χρησιμοποιήσει την επιλογή «Αποθέματα ΟΛΑ» ώστε να επιλέξει από εκεί το απόθεμα που τον ενδιαφέρει. Αν βρεθεί απόθεμα με έγκυρο NSN συμπληρώνεται το πεδίο Ποσότητα με την τρέχουσα τιμή του αποθέματος. Ο χρήστης μπορεί να αλλάξει την τιμή επιλέγοντας το κουμπί «Αλλαγή Ποσότητα».



Εικόνα 4-23: Λειτουργία αλλαγής ποσότητας αποθέματος ανταλλακτικού.

Αν ο χρήστης το επιθυμεί μπορεί να διαγράψει μία ολόκληρη εγγραφή αποθέματος, αν π.χ. το συγκεκριμένο ανταλλακτικό βγήκε από την παραγωγή ή καταργήθηκε.

4.2.6 Προβολή πληροφοριών και εξαγωγή δεδομένων σε αρχεία

Σε διάφορα σημεία της εφαρμογής ο χρήστης έχει τη δυνατότητα να δει πληροφορίες σχετικά με τις αποθηκευμένες οντότητες στο σύστημα. Τα σημεία αυτά περιγράφηκαν αναλυτικά στο προηγούμενο υποκεφάλαιο ανά οθόνη εκτέλεσης. Αφού ο χρήστης επιλέξει το κατάλληλο κουμπί η εφαρμογή συμπληρώνει σχετικά πλέγματα πληροφοριών (grids). Στη συνέχεια ο χρήστης μπορεί να επιλέξει κάποια από τις εγγραφές του πλέγματος ώστε να συμπληρωθούν αυτόματα ορισμένα πεδία της εφαρμογής.

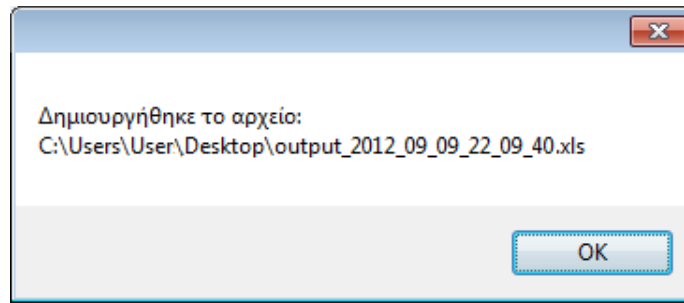
Ποσότητες προς Παραγγελία:				
	Όνομα Ανταλλακτικού	NSN Ανταλλακτικού	Απαιτούμενη Ποσότητα	Διαθέσιμη Ποσότητα
▶	ΚΑΥΣΤΗΡΑΣ	333333333333	10	5
	ΦΛΑΝΤΖΑ	111111111111	12	1
	ΛΑΔΙ SAE-40	222222222222	984	0
	test part 3107	777777777777	20	0
	ΦΙΛΤΡΟ ΛΑΔΙΟΥ	123111111111	20	0

	Όνομα Ανταλλακτικού	NSN Ανταλλακτικού	Ποσότητα για Παραγγελία
▶	ΚΑΥΣΤΗΡΑΣ	333333333333	5
	ΦΛΑΝΤΖΑ	111111111111	11
	ΛΑΔΙ SAE-40	222222222222	984
	test part 3107	777777777777	20
	ΦΙΛΤΡΟ ΛΑΔΙΟΥ	123111111111	20

Εικόνα 4-24: Πίνακες (πλέγματα – grids) πληροφοριών από την οθόνη προβολής λεπτομερειών.

Σε ορισμένες οθόνες της εφαρμογής (κεντρική οθόνη, οθόνη πληροφοριών και προβολή κινήσεων χρηστών) παρέχεται η δυνατότητα αποθήκευσης των δεδομένων σε αρχεία της μορφής .doc ή .xls ώστε να χρησιμοποιηθούν για αρχειοθέτηση, για έκδοση αναφοράς (reporting) ή για κάποια φόρμα παραγγελίας επιπλέον ανταλλακτικών. Τα αρχεία αποθηκεύονται στην επιφάνεια εργασίας του συστήματος και έχουν σαν όνομα την ακριβή ημερομηνία και ώρα που δημιουργήθηκαν. Εκτός από τα δεδομένα, τα αρχεία περιλαμβάνουν επεξηγηματικές πληροφορίες ώστε να μπορούν να είναι αυτόνομες οντότητες πληροφορίας.

Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP

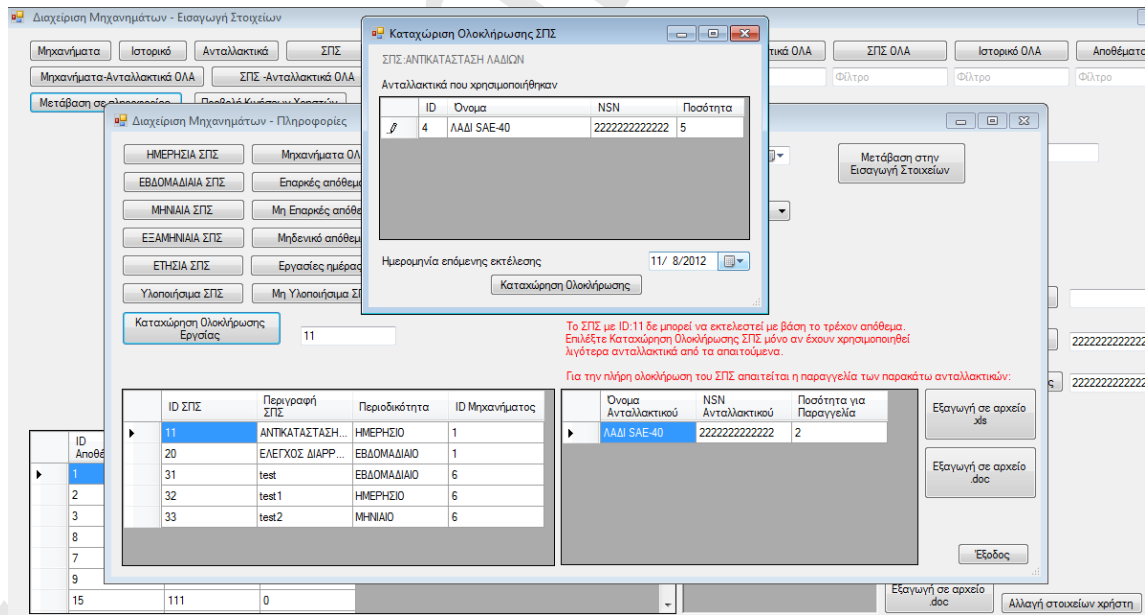


Εικόνα 4-25: Μήνυμα επιτυχούς δημιουργίας αρχείου.

4.2.7 Καταχώρηση Ολοκλήρωσης ΣΠΣ.

Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο, ο χρήστης μπορεί να καταχωρήσει την ολοκλήρωση μιας προγραμματισμένης εργασίας συντήρησης από την οθόνη πληροφοριών της εφαρμογής. Για να ενεργοποιηθεί η επιλογή θα πρέπει να έχει συμπληρωθεί στο αντίστοιχο πεδίο ένας έγκυρος κωδικός ΣΠΣ. Ο κωδικός ΣΠΣ μπορεί να συμπληρωθεί είτε χειροκίνητα είτε επιλέγοντας μία από τις εγγραφές του πλέγματος πληροφοριών της οθόνης.

Αφού εμφανιστεί η οθόνη καταχώρησης ολοκλήρωσης ΣΠΣ, αν περιλαμβάνονται ανταλλακτικά, στο ΣΠΣ η εξ' ορισμού ποσότητα είναι αυτή που ορίστηκε όταν το κάθε ένα από αυτά συνδυάστηκε με το εν λόγω ΣΠΣ. Ο χρήστης μπορεί να αλλάξει αυτή την ποσότητα αν χρησιμοποιήθηκαν λιγότερα ή περισσότερα κομμάτια του ανταλλακτικού από το απόθεμα κατά την εκτέλεση της εργασίας. Με αυτό τον τρόπο δίνεται η δυνατότητα καταχώρησης ακόμη και μη υλοποιήσιμων ΣΠΣ με βάση το απόθεμα (αρκεί να χρησιμοποιήθηκαν λιγότερα από τα απαιτούμενα ανταλλακτικά). Αν ο χρήστης εισάγει τιμή μεγαλύτερη από το απόθεμα για κάποιο ανταλλακτικό, κατά την ολοκλήρωση της εργασίας θα πάρει μήνυμα λάθους. Στην καταχώρηση ολοκλήρωσης ΣΠΣ η επόμενη ημερομηνία εκτέλεσης ορίζεται με βάση αυτή που είχε το ΣΠΣ πριν. Ο χρήστης, αν το επιθυμεί, μπορεί να αλλάξει αυτή την ημερομηνία πριν την καταχώρηση.



Εικόνα 4-26: Λειτουργία καταχώρησης ολοκλήρωσης ΣΠΣ.

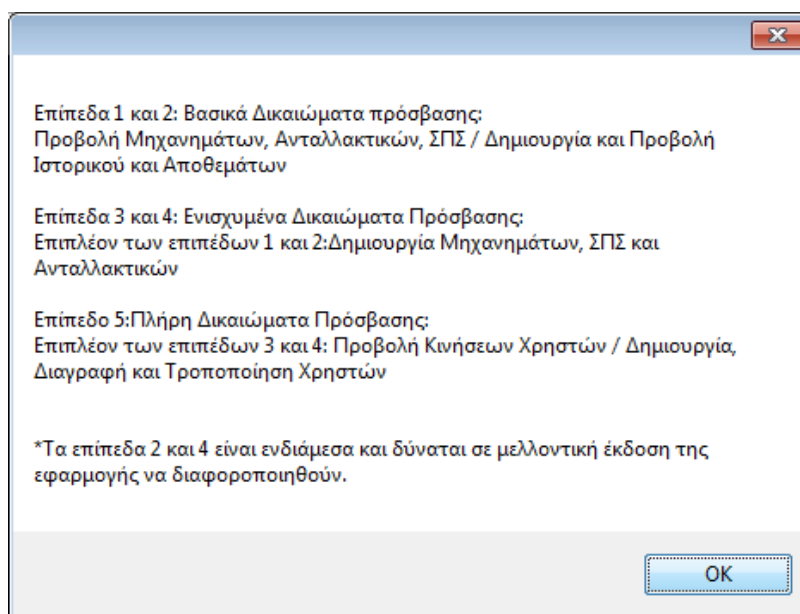
Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP

Με την επιλογή του κουμπιού «Ολοκλήρωση ΣΠΣ» της οθόνης καταχώρησης ολοκλήρωσης ΣΠΣ, το σύστημα ελέγχει ότι υπάρχει επαρκές απόθεμα για όλα τα ανταλλακτικά. Αν υπάρχει, για κάθε ένα από αυτά μειώνει το απόθεμά του. Επιπλέον καταχωρείται νέο ιστορικό αντιστοιχισμένο με το εν λόγω ΣΠΣ. Η επόμενη ημερομηνία εκτέλεσης του ΣΠΣ τίθεται στην τιμή που έχει το αντίστοιχο πεδίο. Τέλος, ανανεώνεται η απαιτούμενη ποσότητα του ανταλλακτικού. Όπως προαναφέρθηκε, ο υπολογισμός απαιτούμενης ποσότητας ενός ανταλλακτικού γίνεται όταν το ανταλλακτικό συνδέεται με ή αποσυνδέεται από ένα ΣΠΣ. Είναι αναγκαίο να γίνει και όταν χρησιμοποιείται στην ολοκλήρωση κάποιου ΣΠΣ ώστε η ποσότητα να είναι όσο γίνεται πιο ακριβής χρονικά. Η ποσότητα υπολογίζεται με βάση τις επίσημες ημερομηνίες ολοκλήρωσης ΣΠΣ. Οριακές συνθήκες όπως δίσεκτα έτη ή ημερομηνίες κοντά στην εκτέλεση ΣΠΣ δε λαμβάνονται υπόψη. Η ποσότητα που υπολογίζεται είναι μία καλή προσέγγιση των ανταλλακτικών που θα χρειαστούν για έναν χρόνο και όχι ακριβής με βάση π.χ. ΣΠΣ που έχουν ήδη ολοκληρωθεί πριν την ώρα τους.

4.2.8 Διαχείριση εφαρμογής.

Οι χρήστες της εφαρμογής χωρίζονται σε 3 κατηγορίες: Η πρώτη κατηγορία (απλοί χρήστες – επίπεδο 1 και 2) έχει τα μικρότερα δικαιώματα πρόσβασης και αφορά χρήστες οι οποίοι μπορούν μόνο να διαβάσουν πληροφορίες σχετικά με τα διαθέσιμα μηχανήματα, ανταλλακτικά και συστήματα προγραμματισμένης συντήρησης (Σ.Π.Σ.) και μπορούν να καταχωρήσουν πληροφορίες αποθεματικών και ιστορικού εκτέλεσης κάποιας προγραμματισμένης – ή μη – εργασίας συντήρησης. Οι χρήστες αυτής της κατηγορίας δε μπορούν να δημιουργήσουν νέα μηχανήματα, ανταλλακτικά ή συστήματα προγραμματισμένοι συζήτησης. Η δεύτερη κατηγορία (αναβαθμισμένοι χρήστες – επίπεδο 3 και 4) έχει αυξημένα δικαιώματα πρόσβασης επιτρέποντας την εισαγωγή, διαγραφή και μετατροπή μηχανημάτων, ανταλλακτικών και συστημάτων προγραμματισμένης συντήρησης. Τέλος, η τρίτη κατηγορία (επίπεδο 5), αφορά το διαχειριστή της εφαρμογής ο οποίος διαθέτει δύο επιπλέον επιλογές σε σχέση με τους απλούς χρήστες.

- Η 1^η επιλογή είναι η δημιουργία, αλλαγή και διαγραφή χρηστών. Μέσα από την οθόνη αλλαγής στοιχείων χρήστη ο διαχειριστής της εφαρμογής μπορεί να διαχειριστεί τους χρήστες του συστήματος. Ο διαχειριστής είναι αυτός που ρυθμίζει τα επίπεδα πρόσβασης κάθε χρήστη.
- Η 2^η επιλογή είναι η προβολή των κινήσεων των χρηστών. Ο διαχειριστής μπορεί να δει πότε και από ποιόν χρήστη έγιναν καταχωρήσεις νέου μηχανήματος ή νέου ιστορικού εκτέλεσης. Με αυτό τον τρόπο μπορούν να εντοπιστούν λάθη ή κακόβουλες ενέργειες προς το σύστημα, αφού δύο από τις σημαντικότερες δραστηριότητες του συστήματος καταγράφονται αναλυτικά. Αν σε επόμενο στάδιο ανάπτυξης απαιτηθεί, μπορεί να ενεργοποιηθεί η καταγραφή και άλλων δραστηριοτήτων.



Εικόνα 4-27: Επεξήγηση επιπέδων πρόσβασης χρηστών.

Κεφάλαιο 5: Συμπεράσματα

Στο παρόν κεφάλαιο παρατίθενται τα συμπεράσματα που προέκυψαν από την ανάπτυξη του συστήματος και ορισμένες προτάσεις για επέκταση των λειτουργιών του σε μελλοντικές νεότερες εκδόσεις.

5.1 Συμπεράσματα

Η κατασκευή του συστήματος διαχείρισης εργασιών συντήρησης έφτασε μέχρι και τη φάση κατασκευής με βάση το μοντέλο RUP. Η φάση μετάβασης δεν υλοποιήθηκε καθώς το σύστημα δεν εγκαταστάθηκε προς χρήση από τους τελικούς χρήστες. Παρ' όλα αυτά, όπως προβλέπει το μοντέλο RUP (βλ. 2.2.4), στο τέλος της φάσης ανάπτυξης πρέπει να ελεγχθεί το κατά πόσο παρέχεται από την τρέχουσα έκδοση του έργου η λειτουργικότητα που απαιτήθηκε. Όπως παρουσιάστηκε στο κεφάλαιο 4, η τελική λειτουργικότητα της εφαρμογής διαχείρισης συντήρησης εκπληρώνει τις προδιαγραφές που παρουσιάστηκαν στο υποκεφάλαιο 3.2 και τις επιπλέον λειτουργίες που εισήχθησαν κατά τη φάση της κατασκευής (ολοκλήρωση ΣΠΣ, αυτόματη συμπλήρωση πεδίων, προβολή κινήσεων χρηστών). Όπως φαίνεται από τα στιγμιότυπα και την περιγραφή των περιπτώσεων χρήσης της εφαρμογής, οι λειτουργίες είναι διαθέσιμες στους χρήστες με τρόπο σαφή και διαφανή περιορίζοντας περιπτώσεις λαθών στην εισαγωγή δεδομένων. Τυχόν λάθη που εντοπίστηκαν από τον έλεγχο της εφαρμογής διορθώθηκαν ώστε να εξασφαλιστεί η επιστροφή σωστών αποτελεσμάτων και η καταχώρηση δεδομένων με ορθό τρόπο.

Η χρήση της επαναληπτικής διαδικασίας ανάπτυξης RUP, βοήθησε στον καλύτερο σχεδιασμό των λειτουργιών του συστήματος και στην ανακάλυψη επιπλέον αναγκών που δεν είχαν διαφανεί στις αρχικές προδιαγραφές. Η αυτόματη συμπλήρωση πεδίων με επιλογή γραμμών, διευκολύνει σημαντικά το χρήστη καθώς ημι-αυτοματοποιεί τις διαδικασίες αλλαγής και διαγραφής δεδομένων και χειρισμού των οντοτήτων. Η ολοκλήρωση ΣΠΣ με αυτόματη ανανέωση ημερομηνιών και αποθεμάτων παρέχει ένα πολύ ισχυρό, και παραμετροποιήσιμο, τρόπο καταχώρησης ιστορικού στους χρήστες. Τέλος, η δυνατότητα προβολής πληροφοριών για τις κινήσεις των χρηστών είναι ένα χρήσιμο εργαλείο για το διαχειριστή του συστήματος ώστε να εντοπίσει λάθη ή κακόβουλες ενέργειες. Οι πρόσθετες αυτές λειτουργίες αναδείχθηκαν χάρη στη διαδικασία RUP και αποτέλεσαν σημαντικές προσθήκες στο συνολικό έργο.

Παρά τα πλεονεκτήματα που επέφερε η χρήση της διαδικασίας RUP έγινε φανερό πως είναι κατάλληλη για έργα λογισμικού μεγαλύτερης κλίμακας. Δεν υπήρχε ομάδα προγραμματιστών ή αναλυτών οπότε δεν ορίστηκαν έγγραφα ρόλων. Επιπλέον ορισμένα διαγράμματα UML δεν ήταν απαραίτητα για την περιγραφή απλών όψεων του συστήματος. Καθώς η εφαρμογή διαχείρισης δρα ως παραθυρικό περιβάλλον πρόσβασης στη βάση, δεν υπήρχαν πολλά περιθώρια για χρήση τεχνικών αντικειμενοστραφούς προγραμματισμού πέρα από τη δημιουργία των φορμών του γραφικού περιβάλλοντος. Τα αντικείμενα της εφαρμογής ήταν κυρίως οι διαφορετικές οθόνες και σαν αποτέλεσμα δεν υπήρχαν πολλές αλληλεπιδράσεις οι οποίες θα μπορούσαν να αναπαρασταθούν. Σε κάθε περίπτωση βέβαια, η διαδικασία RUP βοήθησε στην ορθή ανάπτυξη του έργου χωρίς μεγάλες καθυστερήσεις και με καλό αποτέλεσμα.

Σε αυτό το σημείο αξίζει μία αναφορά στα εργαλεία που χρησιμοποιήθηκαν. Η χρήση του περιβάλλοντος Visual Studio διευκόλυνε την ανάπτυξη αρκετών κομματιών της εφαρμογής, Το παραθυρικό περιβάλλον σχεδιάστηκε και τροποποιήθηκε με πολύ εύκολο τρόπο ενώ, λόγω πολύ των διαδεδομένων τεχνικών ανάπτυξης, η σύνδεση του γραφικού περιβάλλοντος με την πηγή των δεδομένων, δηλαδή τον εξυπηρετητή MySQL, έγινε επίσης εύκολη υπόθεση. Η χρήση, τέλος, της πλατφόρμας .NET προσέφερε καλύτερη εικόνα και εκτέλεση της εφαρμογής καθώς συνεργάζεται αρμονικά με το λειτουργικό σύστημα Windows.

5.2 Προτάσεις για περαιτέρω ανάπτυξη

Όπως αναφέρθηκε και σε προηγούμενα σημεία το σύστημα που κατασκευάστηκε θα μπορούσε να επεκταθεί με αρκετούς τρόπους ώστε να επιτελεί περισσότερες λειτουργίες ή να βρει εφαρμογή σε περισσότερα περιβάλλοντα. Παρακάτω παρατίθενται ορισμένες από αυτές τις πιθανές επεκτάσεις:

- Εισαγωγή περισσότερων ρόλων χρηστών: Τα επίπεδα πρόσβασης 2 και 4 θα μπορούσαν να χρησιμοποιηθούν ώστε να δημιουργηθούν νέα επίπεδα πρόσβασης για πιο σαφή διαχωρισμό στις διαθέσιμες λειτουργίες κάθε επιπέδου.
- Διατήρηση στοιχείων συντηρητών: Το σύστημα θα μπορούσε να διατηρεί πιο λεπτομερή στοιχεία για όσους συμμετέχουν στις διαδικασίες συντήρησης.
- Οικονομική διαχείριση: Θα ήταν χρήσιμο το σύστημα να προσφέρει καταγραφή των δαπανούμενων εργατοωρών ώστε να υπάρχει η δυνατότητα αυτόματης τιμολόγησης εργασιών ή πληρωμής υπερωριών.
- Αυτόματες παραγγελίες: Το σύστημα θα μπορούσε να επεκταθεί ώστε να υποβάλει αυτόματες παραγγελίες σε προμηθευτές ανταλλακτικών ή αναλώσιμων διατηρώντας τη σχετική πληροφορία.
- Βελτίωση γραφικού περιβάλλοντος: Το γραφικό περιβάλλον θα μπορούσε να βελτιωθεί ώστε να είναι ακόμα πιο εύχρηστο. Για το σκοπό αυτό θα ήταν απαραίτητη μία αξιολόγηση από τη μεριά των χρηστών.
- Πολλαπλά σημεία πρόσβασης: Το σύστημα θα μπορούσε να λειτουργήσει με το μοντέλο πελάτη εξυπηρετητή έχοντας μία κεντρική βάση δεδομένων και εγκαταστάσεις της εφαρμογής διαχείρισης σε πολλαπλά σημεία με ταυτόχρονη πρόσβαση. Για το σκοπό αυτό θα έπρεπε πιθανώς να γίνουν αλλαγές στον τρόπο πρόσβασης στη βάση ώστε να εξασφαλιστεί η ατομικότητα ορισμένων περίπλοκων πράξεων.

Βιβλιογραφία

- [1] M. G. Shields. "E-Business and ERP: Rapid Implementation and Project Planning. (2001)" John Wiley and Sons, 2001(pages 7-9).
- [2] W. W. Cato, R. K. Mobley. "Computer-Managed Maintenance Systems, Second Edition: A Step-by-Step Guide to Effective Management of Maintenance, Labor, and Inventory". Elsevier Science & Technology Books, 2001.
- [3] Η ιστοσελίδα προώθησης του εργαλείου AIMMS της εταιρείας Atlantis Engineering: http://www.abe.gr/aimms_gr.html .
- [4] Η ιστοσελίδα προώθησης του εργαλείου iMaint της εταιρείας Ολοκληρωμένα Πληροφοριακά Συστήματα: <http://www.imaint.com/el/proionta/imaint> .
- [5] Η ιστοσελίδα προώθησης του εργαλείου IMIS της εταιρείας IT Open Solutions: http://www.itoss.gr/vertical_solutions/industrial_systems/ .
- [6] Η ιστοσελίδα προώθησης του εργαλείου CMMS της εταιρείας ANCO: <http://www.anco.gr/default.asp?pid=101&la=1> .
- [7] Η ιστοσελίδα προώθησης της εξειδικευμένης εφαρμογής «ΑΝΕΛΚΥΣΤΗΡΑΣ» της εταιρείας ΜΗΧΑΝΟΓΡΑΦΗΣΗ: <http://mhxanografhsh.gr/programsoftware/epixeirhseis-mhxanografhsh/texnika-grafeia-synthrhshs-anelkysthrwn.html> .
- [8] Η ιστοσελίδα προώθησης του εργαλείου Fleetmaster της εταιρείας LOGIFER: <http://www.fleetmaster.gr/> .
- [9] A. Silberschatz, H. F. Korth, S. Sudarshan. "Συστήματα Βάσεων Δεδομένων" (Μεταφρασμένο). Εκδόσεις Γκιούρδα, 2002
- [10] Η ιστοσελίδα του ΣΔΒΔ MySQL: <http://www.mysql.com/>.
- [11] Δελτίο τύπου εξαγοράς του MySQL από την εταιρεία Sun : <http://www.mysql.com/news-and-events/sun-to-acquire-mysql.html> .
- [12] Δελτίο τύπου εξαγοράς της εταιρείας Sun από την εταιρεία Oracle: <http://www.oracle.com/us/sun/index.htm> .
- [13] Σύντομη περιγραφή της κίνησης "Help saving MySQL": <http://monty-says.blogspot.gr/2009/12/help-saving-mysql.html> .
- [14] "C# Language Specification" (4^η έκδοση). Ecma International. 2006.
- [15] Η ιστοσελίδα προώθησης του περιβάλλοντος .NET: <http://www.microsoft.com/net> .
- [16] Η ιστοσελίδα προώθησης του περιβάλλοντος ανάπτυξης Visual Studio της εταιρείας Microsoft: <http://www.microsoft.com/visualstudio/en-us> .
- [17] S.W. Ambler. "A Manager's Introduction to The Rational Unified Process". AmbySoft, 2005.

Παράρτημα

1 Κώδικας SQL της Βάσης Δεδομένων

Η Βάση Δεδομένων που χρησιμοποιείται για το υπολογιστικό σύστημα μπορεί να εγκατασταθεί σε ένα ΣΔΒΔ χρησιμοποιώντας κώδικα που παρατίθεται στο παρόν παράδειγμα.

ΚΕΝΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Η χρήση του ακόλουθου κώδικα SQL δημιουργεί μία κενή βάση δεδομένων. Δεν υπάρχουν καθόλου τιμές και ο μόνος χρήστης είναι ο χρήστης admin με password 1234.

```
use ptychiaki;

SET FOREIGN_KEY_CHECKS=0;
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
--
-- Database: `ptychiaki`
--
-----
--
-- Table structure for table `history`
--
DROP TABLE IF EXISTS `history`;
CREATE TABLE IF NOT EXISTS `history` (
  `historyid` bigint(20) NOT NULL AUTO_INCREMENT,
  `machineid` bigint(20) NOT NULL,
  `spsid` bigint(20) NOT NULL,
  `description` varchar(500) CHARACTER SET utf8 NOT NULL,
  `date` date NOT NULL,
  PRIMARY KEY (`historyid`),
  KEY `machineid` (`machineid`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

Ο πίνακας history περιλαμβάνει τις εγγραφές ιστορικού συντήρησης.

- Το πεδίο historyid είναι ο αύξων αριθμός της εγγραφής και αυξάνεται αυτόματα με την εισαγωγή νέας εγγραφής.

Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP

- Το πεδίο machineid περιγράφει το id του μηχανήματος για το οποίο πραγματοποιήθηκε η εν λόγω συντήρηση. Είναι ξένο κλειδί προς το πεδίο machineid του πίνακα machinedata. Αν το αντίστοιχο machinedata entry διαγραφεί οι history entries που αναφέρονται σε αυτό δε σβήνονται.
- Το πεδίο spsid περιγράφει το id του ΣΠΣ στα πλαίσια του οποίου πραγματοποιήθηκε η εργασία συντήρησης. Αν η εργασία δεν πραγματοποιήθηκε στα πλαίσια κάποιου ΣΠΣ παίρνει την τιμή -1. Δεν ορίστηκε ως ξένο κλειδί.
- Το πεδίο description περιλαμβάνει την περιγραφή της εργασίας συντήρησης.
- Το πεδίο date περιλαμβάνει την ημερομηνία εκτέλεσης της εργασίας συντήρησης.

```
--
-- Dumping data for table `history`
--
-----
--
-- Table structure for table `logs`
--
DROP TABLE IF EXISTS `logs`;
CREATE TABLE IF NOT EXISTS `logs` (
  `logid` bigint(20) NOT NULL AUTO_INCREMENT,
  `logdescription` varchar(100) CHARACTER SET utf8 NOT NULL,
  `objectid` bigint(20) NOT NULL,
  `objectname` varchar(500) CHARACTER SET utf8 NOT NULL,
  `transactiondate` datetime NOT NULL,
  `user` varchar(100) NOT NULL,
  PRIMARY KEY (`logid`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

Ο πίνακας logs περιλαμβάνει τις εγγραφές κινήσεων χρηστών.

- Το πεδίο logid είναι ο αύξων αριθμός της εγγραφής και αυξάνεται αυτόματα με την εισαγωγή νέας εγγραφής.
- Το πεδίο logdescription περιλαμβάνει την περιγραφή της κίνησης χρήστη.
- Το πεδίο objectid περιγράφει το id του μηχανήματος ή του ιστορικού του οποίου η δημιουργία καταγράφεται με αυτή την εγγραφή. Δεν ορίστηκε ως ξένο κλειδί.
- Το πεδίο objectname περιγράφει το όνομα του μηχανήματος ή του ιστορικού του οποίου η δημιουργία καταγράφεται με αυτή την εγγραφή. Δεν ορίστηκε ως ξένο κλειδί.
- Το πεδίο transactiondate περιλαμβάνει την ημερομηνία καταγραφής του log.
- Το πεδίο user περιλαμβάνει το όνομα του χρήστη που εκτέλεσε την εν λόγω ενέργεια.

```
--
-- Dumping data for table `logs`
--
-----
--
```

```
-- Table structure for table `machinedata`  
--  
DROP TABLE IF EXISTS `machinedata`;  
CREATE TABLE IF NOT EXISTS `machinedata` (  
  `machineid` bigint(20) NOT NULL AUTO_INCREMENT,  
  `name` varchar(70) CHARACTER SET utf8 DEFAULT NULL,  
  `time` int(6) DEFAULT NULL,  
  `maxtime` int(6) DEFAULT NULL,  
  PRIMARY KEY (`machineid`),  
  UNIQUE KEY `name` (`name`),  
  UNIQUE KEY `name_2` (`name`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

Ο πίνακας machinedata περιλαμβάνει τις εγγραφές μηχανημάτων.

- Το πεδίο machineid είναι ο αύξων αριθμός της εγγραφής και αυξάνεται αυτόματα με την εισαγωγή νέας εγγραφής.
- Το πεδίο name περιγράφει το όνομα του μηχανήματος. Το όνομα πρέπει να είναι μοναδικό.
- Το πεδίο time περιλαμβάνει τις τρέχουσες ώρες λειτουργίας του μηχανήματος.
- Το πεδίο maxtime περιλαμβάνει τις μέγιστες ώρες λειτουργίας του μηχανήματος.

```
-- Dumping data for table `machinedata`  
--  
-----  
--  
-- Table structure for table `parts`  
--  
DROP TABLE IF EXISTS `parts`;  
CREATE TABLE IF NOT EXISTS `parts` (  
  `partid` bigint(20) NOT NULL AUTO_INCREMENT,  
  `partname` varchar(50) CHARACTER SET utf8 NOT NULL,  
  `partnsn` varchar(13) NOT NULL,  
  `partamount` int(11) NOT NULL DEFAULT '0',  
  PRIMARY KEY (`partid`),  
  UNIQUE KEY `partname` (`partname`,`partnsn`),  
  UNIQUE KEY `partname_2` (`partname`),  
  UNIQUE KEY `partid` (`partid`),  
  KEY `partid_2` (`partid`),  
  KEY `partname_3` (`partname`),  
  KEY `partnsn` (`partnsn`),  
  KEY `partamount` (`partamount`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

Ο πίνακας parts περιλαμβάνει τις εγγραφές ανταλλακτικών.

Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP

- Το πεδίο `partid` είναι ο αύξων αριθμός της εγγραφής και αυξάνεται αυτόματα με την εισαγωγή νέας εγγραφής.
- Το πεδίο `partname` περιγράφει το όνομα του ανταλλακτικού. Το όνομα ανταλλακτικού είναι μοναδικό.
- Το πεδίο `partnsn` περιλαμβάνει τον μοναδικό κωδικό . Ο κωδικός `nsn` είναι μοναδικός.
- Το πεδίο `partamount` περιλαμβάνει την απαιτούμενη ποσότητα ανταλλακτικού.

```
-- Dumping data for table `parts`
```

```
--
```

```
-----
```

```
--
```

```
-- Table structure for table `part_machine`
```

```
--
```

```
DROP TABLE IF EXISTS `part_machine`;
```

```
CREATE TABLE IF NOT EXISTS `part_machine` (
```

```
  `part_machine_id` bigint(20) NOT NULL AUTO_INCREMENT,
```

```
  `partid` bigint(20) NOT NULL,
```

```
  `machineid` bigint(20) NOT NULL,
```

```
  PRIMARY KEY (`part_machine_id`),
```

```
  KEY `partid` (`partid`),
```

```
  KEY `machineid` (`machineid`),
```

```
  KEY `partid_2` (`partid`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

Ο πίνακας `part_machine` περιλαμβάνει τις αντιστοιχίσεις ανταλλακτικών με μηχανήματα.

- Το πεδίο `part_machine_id` είναι ο αύξων αριθμός της εγγραφής και αυξάνεται αυτόματα με την εισαγωγή νέας εγγραφής.
- Το πεδίο `partid` περιγράφει το id του ανταλλακτικού της αντιστοίχισης και είναι ξένο κλειδί προς το πεδίο `partid` του πίνακα `parts`.
- Το πεδίο `machineid` περιγράφει το id του μηχανήματος της αντιστοίχισης και είναι ξένο κλειδί προς το πεδίο `machineid` του πίνακα `machinedata`.
- Αν ένα ανταλλακτικό ή μηχανήμα διαγραφεί ή αλλάξει το ID του, διαγράφονται ή ανανεώνονται και οι αντίστοιχες εγγραφές από τον πίνακα `part_machine`. (ON DELETE CASCADE ON UPDATE CASCADE).

```
--
```

```
-- Dumping data for table `part_machine`
```

```
--
```

```
-----
```

```
--
```

```
-- Table structure for table `part_sps`
```

```
--
```

```
DROP TABLE IF EXISTS `part_sps`;
```

```
CREATE TABLE IF NOT EXISTS `part_sps` (
```

```
  `part_sps_id` bigint(20) NOT NULL AUTO_INCREMENT,
```

Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP


```

`partid` bigint(20) NOT NULL,
`spsid` bigint(20) NOT NULL,
`part_sps_amount` int(11) NOT NULL,
PRIMARY KEY (`part_sps_id`),
KEY `partid` (`partid`),
KEY `spsid` (`spsid`),
KEY `part_sps_amount` (`part_sps_amount`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

Ο πίνακας part_sps περιλαμβάνει τις αντιστοιχίσεις ανταλλακτικών με μηχανήματα.

- Το πεδίο part_sps_id είναι ο αύξων αριθμός της εγγραφής και αυξάνεται αυτόματα με την εισαγωγή νέας εγγραφής.
- Το πεδίο partid περιγράφει το id του ανταλλακτικού της αντιστοίχισης και είναι ξένο κλειδί προς το πεδίο partid του πίνακα parts.
- Το πεδίο spsid περιγράφει το id του ΣΠΣ της αντιστοίχισης και είναι ξένο κλειδί προς το πεδίο idsps του πίνακα sps.
- Το πεδίο part_sps_amount περιλαμβάνει την απαιτούμενη ποσότητα ανταλλακτικού στην εν λόγω αντιστοίχιση με ΣΠΣ.
- Αν ένα ανταλλακτικό ή ΣΠΣ διαγραφεί ή αλλάξει το ID του, διαγράφονται ή ανανεώνονται και οι αντίστοιχες εγγραφές από τον πίνακα part_sps. (ON DELETE CASCADE ON UPDATE CASCADE).

```

--
-- Dumping data for table `part_sps`
--
-----
--
-- Table structure for table `sps`
--
DROP TABLE IF EXISTS `sps`;
CREATE TABLE IF NOT EXISTS `sps` (
  `idsps` bigint(20) NOT NULL AUTO_INCREMENT,
  `sps` varchar(20) CHARACTER SET utf8 NOT NULL,
  `periodikotita` varchar(20) CHARACTER SET utf8 NOT NULL,
  `codeper` int(2) NOT NULL,
  `mixid` bigint(20) NOT NULL,
  `nextdate` date DEFAULT NULL,
  PRIMARY KEY (`idsps`),
  UNIQUE KEY `sps` (`sps`),
  KEY `mixid` (`mixid`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

Ο πίνακας sps περιλαμβάνει τις εγγραφές ΣΠΣ.

- Το πεδίο `idsps` είναι ο αύξων αριθμός της εγγραφής και αυξάνεται αυτόματα με την εισαγωγή νέας εγγραφής.
- Το πεδίο `sps` περιλαμβάνει την περιγραφή του ΣΠΣ. Η περιγραφή του ΣΠΣ είναι μοναδική.
- Το πεδίο `periodikotita` περιγράφει την περιοδικότητα του ΣΠΣ με λεκτική μορφή.
- Το πεδίο `coderep` περιγράφει την περιοδικότητα του ΣΠΣ με λεκτική μορφή (Δε χρησιμοποιείται πλέον από την εφαρμογή. Παρέμεινε για μελλοντικές επεκτάσεις).
- Το πεδίο `mixid` περιγράφει το `id` του μηχανήματος στο οποίο μπορεί να εκτελεστεί το εν λόγω ΣΠΣ. Είναι ξένο κλειδί προς το πεδίο `mchineid` του πίνακα `machinedata`.
- Το πεδίο `nextdate` περιλαμβάνει την ημερομηνία επόμενης εκτέλεσης του ΣΠΣ.

```
--
-- Dumping data for table `sps`
--
-----
--
-- Table structure for table `stock`
--
DROP TABLE IF EXISTS `stock`;
CREATE TABLE IF NOT EXISTS `stock` (
  `stockid` bigint(20) NOT NULL AUTO_INCREMENT,
  `stocknsn` varchar(13) NOT NULL,
  `stockamount` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`stockid`),
  UNIQUE KEY `stocknsn` (`stocknsn`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

Ο πίνακας `stock` περιλαμβάνει τα τρέχοντα αποθέματα ανταλλακτικών.

- Το πεδίο `stockid` είναι ο αύξων αριθμός της εγγραφής και αυξάνεται αυτόματα με την εισαγωγή νέας εγγραφής.
- Το πεδίο `stocknsn` περιγράφει τον κωδικό `nsn` του αποθέματος και είναι ξένο κλειδί προς το πεδίο `partnsn` του πίνακα `parts`.
- Το πεδίο `stockamount` περιλαμβάνει την τρέχουσα ποσότητα ανταλλακτικού.
- Αν ένα ανταλλακτικό ή αλλάξει το ID του, διαγράφονται ή ανανεώνονται και οι αντίστοιχες εγγραφές από τον πίνακα `stock`. (ON DELETE CASCADE ON UPDATE CASCADE).

```
--
-- Dumping data for table `stock`
--
-----
--
-- Table structure for table `users`
--
DROP TABLE IF EXISTS `users`;
CREATE TABLE IF NOT EXISTS `users` (
```

```
`userid` int(11) NOT NULL AUTO_INCREMENT,  
`username` varchar(25) DEFAULT NULL,  
`userpassword` varchar(25) DEFAULT NULL,  
`fname` varchar(25) CHARACTER SET utf8 DEFAULT NULL,  
`lname` varchar(25) CHARACTER SET utf8 DEFAULT NULL,  
`mail` varchar(25) DEFAULT NULL,  
`level` int(11) NOT NULL DEFAULT '1',  
PRIMARY KEY (`userid`),  
UNIQUE KEY `username` (`username`),  
UNIQUE KEY `username_2` (`username`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;
```

Ο πίνακας users περιλαμβάνει τις πληροφορίες για τους χρήστες της εφαρμογής.

- Το πεδίο userid είναι ο αύξων αριθμός της εγγραφής και αυξάνεται αυτόματα με την εισαγωγή νέας εγγραφής.
- Το πεδίο username περιγράφει τον κωδικό χρήστη. Ο κωδικός χρήστη είναι μοναδικός.
- Το πεδίο userpassword περιλαμβάνει τον κωδικό πρόσβασης του χρήστη.
- Το πεδίο fname περιλαμβάνει το όνομα του χρήστη.
- Το πεδίο lname περιλαμβάνει το επίθετο του χρήστη.
- Το πεδίο mail περιλαμβάνει τον email του χρήστη.
- Το πεδίο level περιγράφει το επίπεδο πρόσβασης του χρήστη.

```
--
```

```
-- Dumping data for table `users`
```

```
--
```

```
INSERT INTO `users` (`userid`, `username`, `userpassword`, `fname`, `lname`, `mail`, `level`) VALUES  
(1, 'admin', '1234', 'admin', 'admin', 'admin@parts.com', 5);
```

```
--
```

```
-- Constraints for dumped tables
```

```
--
```

```
--
```

```
-- Constraints for table `history`
```

```
--
```

```
ALTER TABLE `history`
```

```
ADD CONSTRAINT `history_ibfk_1` FOREIGN KEY (`machineid`) REFERENCES `machinedata`  
(`machineid`);
```

Το πεδίο machineid του πίνακα history είναι ξένο κλειδί προς το πεδίο machineid του πίνακα machinedata. Το όνομα του περιορισμού ξένου κλειδιού είναι history_ibfk_1.

```
--
```

Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP

```
-- Constraints for table `part_machine`
```

```
--
```

```
ALTER TABLE `part_machine`
```

```
  ADD CONSTRAINT `part_machine_ibfk_1` FOREIGN KEY (`machineid`) REFERENCES `machinedata`  
  (`machineid`) ON DELETE CASCADE ON UPDATE CASCADE,
```

```
  ADD CONSTRAINT `part_machine_ibfk_2` FOREIGN KEY (`partid`) REFERENCES `parts` (`partid`) ON  
  DELETE CASCADE ON UPDATE CASCADE;
```

- Το πεδίο machineid του πίνακα part_machine είναι ξένο κλειδί προς το πεδίο machineid του πίνακα machinedata. Το όνομα του περιορισμού ξένου κλειδιού είναι part_machine_ibfk_1. Αν η αναφερόμενη εγγραφή του πίνακα machinedata διαγραφεί, θα διαγραφούν όλες εγγραφές του πίνακα part_machine αναφέρονται σε αυτή(ON DELETE CASCADE). Αν το id της αναφερόμενης εγγραφής του πίνακα machinedata αλλάξει θα αλλάξουν αντίστοιχα τα id των εγγραφών του πίνακα part_machine(ON UPDATE CASCADE).
- Το πεδίο partid του πίνακα part_machine είναι ξένο κλειδί προς το πεδίο partid του πίνακα parts. Το όνομα του περιορισμού ξένου κλειδιού είναι part_machine_ibfk_2. Αν η αναφερόμενη εγγραφή του πίνακα parts διαγραφεί, θα διαγραφούν όλες εγγραφές του πίνακα part_machine αναφέρονται σε αυτή(ON DELETE CASCADE). Αν το id της αναφερόμενης εγγραφής του πίνακα parts αλλάξει θα αλλάξουν αντίστοιχα τα id των εγγραφών του πίνακα part_machine(ON UPDATE CASCADE).

```
--
```

```
-- Constraints for table `part_sps`
```

```
--
```

```
ALTER TABLE `part_sps`
```

```
  ADD CONSTRAINT `part_sps_ibfk_2` FOREIGN KEY (`partid`) REFERENCES `parts` (`partid`) ON DELETE  
  CASCADE ON UPDATE CASCADE,
```

```
  ADD CONSTRAINT `part_sps_ibfk_3` FOREIGN KEY (`spsid`) REFERENCES `sps` (`idsps`) ON DELETE  
  CASCADE ON UPDATE CASCADE;
```

- Το πεδίο partid του πίνακα part_sps είναι ξένο κλειδί προς το πεδίο partid του πίνακα parts. Το όνομα του περιορισμού ξένου κλειδιού είναι part_sps_ibfk_2. Αν η αναφερόμενη εγγραφή του πίνακα parts διαγραφεί, θα διαγραφούν όλες εγγραφές του πίνακα part_sps αναφέρονται σε αυτή(ON DELETE CASCADE). Αν το id της αναφερόμενης εγγραφής του πίνακα parts αλλάξει θα αλλάξουν αντίστοιχα τα id των εγγραφών του πίνακα part_sps(ON UPDATE CASCADE).
- Το πεδίο spsid του πίνακα part_sps είναι ξένο κλειδί προς το πεδίο idsps του πίνακα sps. Το όνομα του περιορισμού ξένου κλειδιού είναι part_sps_ibfk_3. Αν η αναφερόμενη εγγραφή του πίνακα sps διαγραφεί, θα διαγραφούν όλες εγγραφές του πίνακα part_sps αναφέρονται σε αυτή(ON DELETE CASCADE). Αν το id της αναφερόμενης εγγραφής του πίνακα sps αλλάξει θα αλλάξουν αντίστοιχα τα id των εγγραφών του πίνακα part_sps(ON UPDATE CASCADE).

```
--
```

```
-- Constraints for table `sps`
```

```
--
```

```
ALTER TABLE `sps`
```

```
  ADD CONSTRAINT `sps_ibfk_1` FOREIGN KEY (`mixid`) REFERENCES `machinedata` (`machineid`) ON  
  DELETE CASCADE ON UPDATE CASCADE;
```

- Το πεδίο mixed του πίνακα sps είναι ξένο κλειδί προς το πεδίο machineid του πίνακα machinedata. Το όνομα του περιορισμού ξένου κλειδιού είναι sps_ibfk_1. Αν η αναφερόμενη εγγραφή του πίνακα machinedata διαγραφεί, θα διαγραφούν όλες εγγραφές του πίνακα sps αναφέρονται σε αυτή (ON DELETE CASCADE). Αν το id της αναφερόμενης εγγραφής του πίνακα machinedata αλλάξει θα αλλάξουν αντίστοιχα τα id των εγγραφών του πίνακα sps (ON UPDATE CASCADE).

```
--
-- Constraints for table `stock`
--
ALTER TABLE `stock`
  ADD CONSTRAINT `stock_ibfk_1` FOREIGN KEY (`stocknsn`) REFERENCES `parts` (`partnsn`) ON DELETE
  CASCADE ON UPDATE CASCADE;
```

- Το πεδίο stocknsn του πίνακα stock είναι ξένο κλειδί προς το πεδίο partnsn του πίνακα parts. Το όνομα του περιορισμού ξένου κλειδιού είναι stock_ibfk_1. Αν η αναφερόμενη εγγραφή του πίνακα parts διαγραφεί, θα διαγραφούν όλες εγγραφές του πίνακα stock αναφέρονται σε αυτή (ON DELETE CASCADE). Αν το id της αναφερόμενης εγγραφής του πίνακα parts αλλάξει θα αλλάξουν αντίστοιχα τα id των εγγραφών του πίνακα stock (ON UPDATE CASCADE).

```
SET FOREIGN_KEY_CHECKS=1;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΜΕ ΔΟΚΙΜΑΣΤΙΚΕΣ ΤΙΜΕΣ

Η χρήση του ακόλουθου κώδικα SQL δημιουργεί μία βάση δεδομένων η οποία περιέχει κάποιες φανταστικές, δοκιμαστικές τιμές.

```
use ptychiaki;
SET FOREIGN_KEY_CHECKS=0;
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
--
-- Database: `ptychiaki`
--
-----
--
-- Table structure for table `history`
--
DROP TABLE IF EXISTS `history`;
```

Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP

```
CREATE TABLE IF NOT EXISTS `history` (  
  `historyid` bigint(20) NOT NULL AUTO_INCREMENT,  
  `machineid` bigint(20) NOT NULL,  
  `spsid` bigint(20) NOT NULL,  
  `description` varchar(500) CHARACTER SET utf8 NOT NULL,  
  `date` date NOT NULL,  
  PRIMARY KEY (`historyid`),  
  KEY `machineid` (`machineid`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=24 ;  
  
--  
-- Dumping data for table `history`  
--  
INSERT INTO `history` (`historyid`, `machineid`, `spsid`, `description`, `date`) VALUES  
(1, 1, -1, 'ΤΕΣΤ ΙΣΤΟΡΙΚΟ', '2012-07-24'),  
(2, 1, -1, 'ΚΑΤΑΓΡΑΦΗ', '2012-07-10'),  
(10, 7, -1, 'jhfhgn', '2012-07-30'),  
(14, 1, 12, 'sfd', '2012-07-31'),  
(15, 1, 11, 'Ολοκλήρωση ΣΠΣ:ΑΝΤΙΚΑΤΑΣΤΑΣΗ ΛΑΔΙΩΝ', '2012-08-07'),  
(16, 6, 31, 'Ολοκλήρωση ΣΠΣ:test', '2012-08-13'),  
(17, 6, 31, 'Ολοκλήρωση ΣΠΣ:test', '2012-08-13'),  
(21, 6, 31, 'Ολοκλήρωση ΣΠΣ:test', '2012-08-14'),  
(22, 1, 21, 'Ολοκλήρωση ΣΠΣ:ΦΙΛΤΡΑ', '2012-09-02'),  
(23, 1, -1, 'ΕΚΤΑΚΤΗ ΕΠΙΣΚΕΥΗ Κ. ΜΗΧΑΝΗΣ', '2012-09-07');  
  
-----  
--  
-- Table structure for table `logs`  
--  
DROP TABLE IF EXISTS `logs`;  
CREATE TABLE IF NOT EXISTS `logs` (  
  `logid` bigint(20) NOT NULL AUTO_INCREMENT,  
  `logdescription` varchar(100) CHARACTER SET utf8 NOT NULL,  
  `objectid` bigint(20) NOT NULL,  
  `objectname` varchar(500) CHARACTER SET utf8 NOT NULL,  
  `transactiondate` datetime NOT NULL,  
  `user` varchar(100) NOT NULL,  
  PRIMARY KEY (`logid`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=22 ;  
  
--  
-- Dumping data for table `logs`  
--  
INSERT INTO `logs` (`logid`, `logdescription`, `objectid`, `objectname`, `transactiondate`, `user`) VALUES  
(1, 'NEO ΜΗΧΑΝΗΜΑ', 12, '1213', '2012-07-29 17:49:25', 'admin'),  
Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη  
χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP
```

```
(2, 'NEO ΙΣΤΟΡΙΚΟ', 12, 'test3007', '2012-07-30 02:27:48', '3'),
(3, 'NEO ΜΗΧΑΝΗΜΑ', 13, 'test machine 3007', '2012-07-30 02:28:54', 'admin'),
(4, 'NEO ΜΗΧΑΝΗΜΑ', 14, 'δεδε', '2012-07-30 22:30:14', '3'),
(5, 'NEO ΙΣΤΟΡΙΚΟ', 13, 'xzcxcz', '2012-07-30 22:33:59', '3'),
(6, 'NEO ΙΣΤΟΡΙΚΟ', 14, 'sfd', '2012-07-31 21:13:48', '3'),
(7, 'NEO ΙΣΤΟΡΙΚΟ', 15, 'Ολοκλήρωση ΣΠΣ:ΑΝΤΙΚΑΤΑΣΤΑΣΗ ΛΑΔΙΩΝ', '2012-08-07 01:44:13', '3'),
(8, 'NEO ΙΣΤΟΡΙΚΟ', 16, 'Ολοκλήρωση ΣΠΣ:test', '2012-08-13 19:23:41', '3'),
(9, 'NEO ΙΣΤΟΡΙΚΟ', 17, 'Ολοκλήρωση ΣΠΣ:test', '2012-08-13 19:24:02', '3'),
(10, 'NEO ΜΗΧΑΝΗΜΑ', 13, '1', '2012-08-13 19:45:51', '3'),
(11, 'NEO ΙΣΤΟΡΙΚΟ', 18, '12', '2012-08-13 19:46:50', '3'),
(12, 'NEO ΜΗΧΑΝΗΜΑ', 14, '1', '2012-08-13 19:59:10', '3'),
(13, 'NEO ΙΣΤΟΡΙΚΟ', 19, '1', '2012-08-13 19:59:28', '3'),
(14, 'NEO ΜΗΧΑΝΗΜΑ', 15, '1', '2012-08-13 22:54:07', '3'),
(15, 'NEO ΜΗΧΑΝΗΜΑ', 14, '1', '2012-08-13 22:54:10', '3'),
(16, 'NEO ΜΗΧΑΝΗΜΑ', 15, '1', '2012-08-13 22:54:53', '3'),
(17, 'NEO ΜΗΧΑΝΗΜΑ', 21, 'wa', '2012-08-14 00:02:56', 'admin'),
(18, 'NEO ΙΣΤΟΡΙΚΟ', 20, 'was', '2012-08-14 00:03:15', 'admin'),
(19, 'NEO ΙΣΤΟΡΙΚΟ', 21, 'Ολοκλήρωση ΣΠΣ:test', '2012-08-14 00:04:14', 'admin'),
(20, 'NEO ΙΣΤΟΡΙΚΟ', 22, 'Ολοκλήρωση ΣΠΣ:ΦΙΛΤΡΑ', '2012-09-02 02:50:47', 'admin'),
(21, 'NEO ΙΣΤΟΡΙΚΟ', 23, 'ΕΚΤΑΚΤΗ ΕΠΙΣΚΕΥΗ Κ. ΜΗΧΑΝΗΣ', '2012-09-09 22:04:27', 'admin');
```

```
-----
--
-- Table structure for table `machinedata`
--
DROP TABLE IF EXISTS `machinedata`;
CREATE TABLE IF NOT EXISTS `machinedata` (
  `machineid` bigint(20) NOT NULL AUTO_INCREMENT,
  `name` varchar(70) CHARACTER SET utf8 DEFAULT NULL,
  `time` int(6) DEFAULT NULL,
  `maxtime` int(6) DEFAULT NULL,
  PRIMARY KEY (`machineid`),
  UNIQUE KEY `name` (`name`),
  UNIQUE KEY `name_2` (`name`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;
--
-- Dumping data for table `machinedata`
--
INSERT INTO `machinedata` (`machineid`, `name`, `time`, `maxtime`) VALUES
(1, 'Κ ΜΗΧΑΝΗ 1', 5800, 6000),
(2, 'Κ ΜΗΧΑΝΗ 3', 5800, 7000),
(6, 'ΣΥΜΠΙΕΣΤΗΣ ΚΛΙΜΑΤΙΣΜΟΥ 1', 759, 900),
(7, 'ΔΟΚΙΜΑΣΤΙΚΟ ΜΗΧΑΝΗΜΑ 1', 200, 20001);
```

Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP

```
-----  
--  
-- Table structure for table `parts`  
--  
DROP TABLE IF EXISTS `parts`;  
CREATE TABLE IF NOT EXISTS `parts` (  
  `partid` bigint(20) NOT NULL AUTO_INCREMENT,  
  `partname` varchar(50) CHARACTER SET utf8 NOT NULL,  
  `partnsn` varchar(13) NOT NULL,  
  `partamount` int(11) NOT NULL DEFAULT '0',  
  PRIMARY KEY (`partid`),  
  UNIQUE KEY `partname` (`partname`,`partnsn`),  
  UNIQUE KEY `partname_2` (`partname`),  
  UNIQUE KEY `partid` (`partid`),  
  KEY `partid_2` (`partid`),  
  KEY `partname_3` (`partname`),  
  KEY `partnsn` (`partnsn`),  
  KEY `partamount` (`partamount`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=22 ;  
  
--  
-- Dumping data for table `parts`  
--  
INSERT INTO `parts` (`partid`, `partname`, `partnsn`, `partamount`) VALUES  
(1, 'ΚΥΛΙΝΔΡΟΚΕΦΑΛΗ', '55555555555555', 3),  
(2, 'ΚΑΥΣΤΗΡΑΣ', '33333333333333', 10),  
(3, 'ΦΛΑΝΤΖΑ', '11111111111111', 12),  
(4, 'ΛΑΔΙ SAE-40', '22222222222222', 984),  
(6, 'test part 3107', '77777777777777', 20),  
(21, 'ΦΙΑΤΡΟ ΛΑΔΙΟΥ', '123111111111', 48);  
-----
```

```
--  
-- Table structure for table `part_machine`  
--  
DROP TABLE IF EXISTS `part_machine`;  
CREATE TABLE IF NOT EXISTS `part_machine` (  
  `part_machine_id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `partid` bigint(20) NOT NULL,  
  `machineid` bigint(20) NOT NULL,  
  PRIMARY KEY (`part_machine_id`),  
  KEY `partid` (`partid`),  
  KEY `machineid` (`machineid`),
```

Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP


```
KEY `partid_2` (`partid`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=29 ;
--
-- Dumping data for table `part_machine`
--
INSERT INTO `part_machine` (`part_machine_id`, `partid`, `machineid`) VALUES
(18, 6, 2),
(22, 4, 6),
(28, 4, 1);
-----
--
-- Table structure for table `part_sps`
--
DROP TABLE IF EXISTS `part_sps`;
CREATE TABLE IF NOT EXISTS `part_sps` (
  `part_sps_id` bigint(20) NOT NULL AUTO_INCREMENT,
  `partid` bigint(20) NOT NULL,
  `spsid` bigint(20) NOT NULL,
  `part_sps_amount` int(11) NOT NULL,
  PRIMARY KEY (`part_sps_id`),
  KEY `partid` (`partid`),
  KEY `spsid` (`spsid`),
  KEY `part_sps_amount` (`part_sps_amount`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=23 ;
--
-- Dumping data for table `part_sps`
--
INSERT INTO `part_sps` (`part_sps_id`, `partid`, `spsid`, `part_sps_amount`) VALUES
(17, 4, 31, 3),
(22, 4, 11, 7);
-----
--
-- Table structure for table `sps`
--
DROP TABLE IF EXISTS `sps`;
CREATE TABLE IF NOT EXISTS `sps` (
  `idsps` bigint(20) NOT NULL AUTO_INCREMENT,
  `sps` varchar(20) CHARACTER SET utf8 NOT NULL,
  `periodikotita` varchar(20) CHARACTER SET utf8 NOT NULL,
  `codeper` int(2) NOT NULL,
  Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη
  χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP
```

```

`mixid` bigint(20) NOT NULL,
`nextdate` date DEFAULT NULL,
PRIMARY KEY (`idsps`),
UNIQUE KEY `sps` (`sps`),
KEY `mixid` (`mixid`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=36 ;
--
-- Dumping data for table `sps`
--
INSERT INTO `sps` (`idsps`, `sps`, `periodikotita`, `codeper`, `mixid`, `nextdate`) VALUES
(11, 'ΑΝΤΙΚΑΤΑΣΤΑΣΗ ΛΑΔΙΩΝ', 'ΗΜΕΡΗΣΙΟ', 1, 1, '2012-08-10'),
(12, 'ΔΟΚΙΜΗ ΛΕΙΤΟΥΡΓΙΑΣ', 'ΕΞΑΜΗΝΙΑΙΟ', 4, 1, '2013-01-10'),
(18, 'ΑΛΛΑΓΗ ΙΜΑΝΤΑ', 'ΕΤΗΣΙΟ', 5, 1, '2013-01-20'),
(20, 'ΕΛΕΓΧΟΣ ΔΙΑΡΡΟΩΝ', 'ΕΒΔΟΜΑΔΙΑΙΟ', 2, 1, '2012-08-10'),
(21, 'ΦΙΛΤΡΑ', 'ΜΗΝΙΑΙΟ', 3, 1, '2012-10-01'),
(31, 'test', 'ΕΒΔΟΜΑΔΙΑΙΟ', 2, 6, '2012-08-31'),
(32, 'test1', 'ΗΜΕΡΗΣΙΟ', 1, 6, '2012-08-08'),
(33, 'test2', 'ΜΗΝΙΑΙΟ', 3, 6, '2012-09-01'),
(34, 'test3', 'ΕΞΑΜΗΝΙΑΙΟ', 4, 6, '2013-01-10'),
(35, 'test4', 'ΕΤΗΣΙΟ', 5, 6, '2013-01-20');
-----
--
-- Table structure for table `stock`
--
DROP TABLE IF EXISTS `stock`;
CREATE TABLE IF NOT EXISTS `stock` (
  `stockid` bigint(20) NOT NULL AUTO_INCREMENT,
  `stocknsn` varchar(13) NOT NULL,
  `stockamount` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`stockid`),
  UNIQUE KEY `stocknsn` (`stocknsn`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=22 ;
--
-- Dumping data for table `stock`
--
INSERT INTO `stock` (`stockid`, `stocknsn`, `stockamount`) VALUES
(1, '22222222222222', 7),
(2, '33333333333333', 5),
(3, '55555555555555', 10),
(7, '11111111111111', 1),
(8, '77777777777777', 0),
(21, '12311111111111', 0);

```

Ανάλυση και κατασκευή ενός πληροφοριακού συστήματος διαχείρισης εργασιών συντήρησης μηχανημάτων με τη χρήση της αντικειμενοστραφούς μεθόδου ανάπτυξης RUP

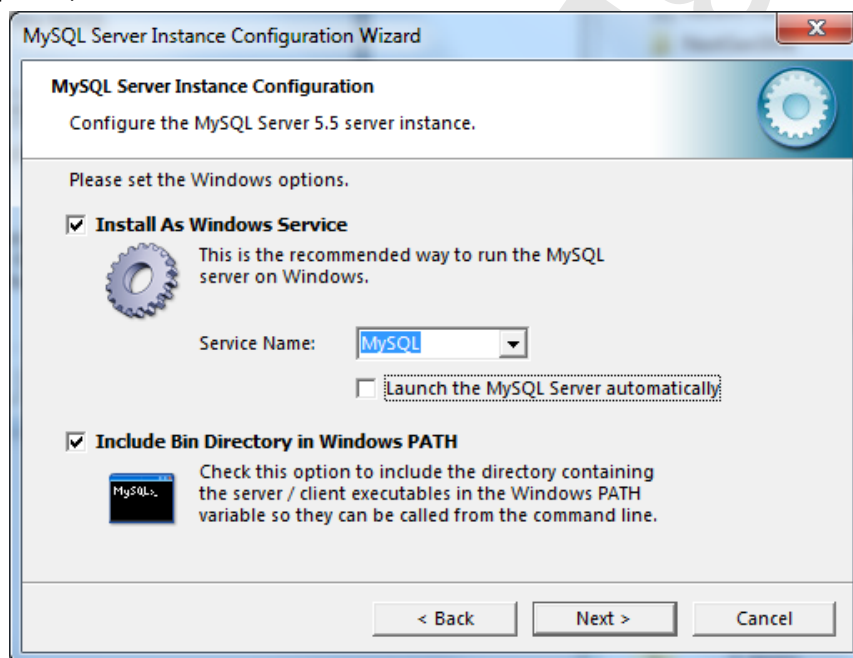
```
-----
--
-- Table structure for table `users`
--
DROP TABLE IF EXISTS `users`;
CREATE TABLE IF NOT EXISTS `users` (
  `userid` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(25) DEFAULT NULL,
  `userpassword` varchar(25) DEFAULT NULL,
  `fname` varchar(25) CHARACTER SET utf8 DEFAULT NULL,
  `lname` varchar(25) CHARACTER SET utf8 DEFAULT NULL,
  `mail` varchar(25) DEFAULT NULL,
  `level` int(11) NOT NULL DEFAULT '1',
  PRIMARY KEY (`userid`),
  UNIQUE KEY `username` (`username`),
  UNIQUE KEY `username_2` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=35 ;
--
-- Dumping data for table `users`
--
INSERT INTO `users` (`userid`, `username`, `userpassword`, `fname`, `lname`, `mail`, `level`) VALUES
(1, 'admin', '1234', 'vagelis', 'siatis', 'vagsiatis@gmail.com', 5),
(2, 'vagelis', '4321', 'vag', 'siat', 'fas511@hotmail.com', 3),
(3, 'johndoe', 'john1234', 'John', 'Doe', 'johndoe@gmail.com', 1),
(32, '1', '1', '1', '1', '1', 1),
(33, '3', '3', '3', '3', '3', 3),
(34, '5', '5', '5', '5', '5', 5);
--
-- Constraints for dumped tables
--
--
-- Constraints for table `history`
--
ALTER TABLE `history`
  ADD CONSTRAINT `history_ibfk_1` FOREIGN KEY (`machineid`) REFERENCES `machinedata`
  (`machineid`);
--
-- Constraints for table `part_machine`
--
ALTER TABLE `part_machine`
  ADD CONSTRAINT `part_machine_ibfk_1` FOREIGN KEY (`machineid`) REFERENCES `machinedata`
  (`machineid`) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ADD CONSTRAINT `part_machine_ibfk_2` FOREIGN KEY (`partid`) REFERENCES `parts` (`partid`) ON
DELETE CASCADE ON UPDATE CASCADE;
--
-- Constraints for table `part_sps`
--
ALTER TABLE `part_sps`
  ADD CONSTRAINT `part_sps_ibfk_2` FOREIGN KEY (`partid`) REFERENCES `parts` (`partid`) ON DELETE
  CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `part_sps_ibfk_3` FOREIGN KEY (`spsid`) REFERENCES `sps` (`idsps`) ON DELETE
  CASCADE ON UPDATE CASCADE;
--
-- Constraints for table `sps`
--
ALTER TABLE `sps`
  ADD CONSTRAINT `sps_ibfk_1` FOREIGN KEY (`mixid`) REFERENCES `machinedata` (`machineid`) ON
  DELETE CASCADE ON UPDATE CASCADE;
--
-- Constraints for table `stock`
--
ALTER TABLE `stock`
  ADD CONSTRAINT `stock_ibfk_1` FOREIGN KEY (`stocknsn`) REFERENCES `parts` (`partnsn`) ON DELETE
  CASCADE ON UPDATE CASCADE;
SET FOREIGN_KEY_CHECKS=1;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

2 Εγκατάσταση Εργαλείων και σύνδεση εφαρμογής με τη βάση

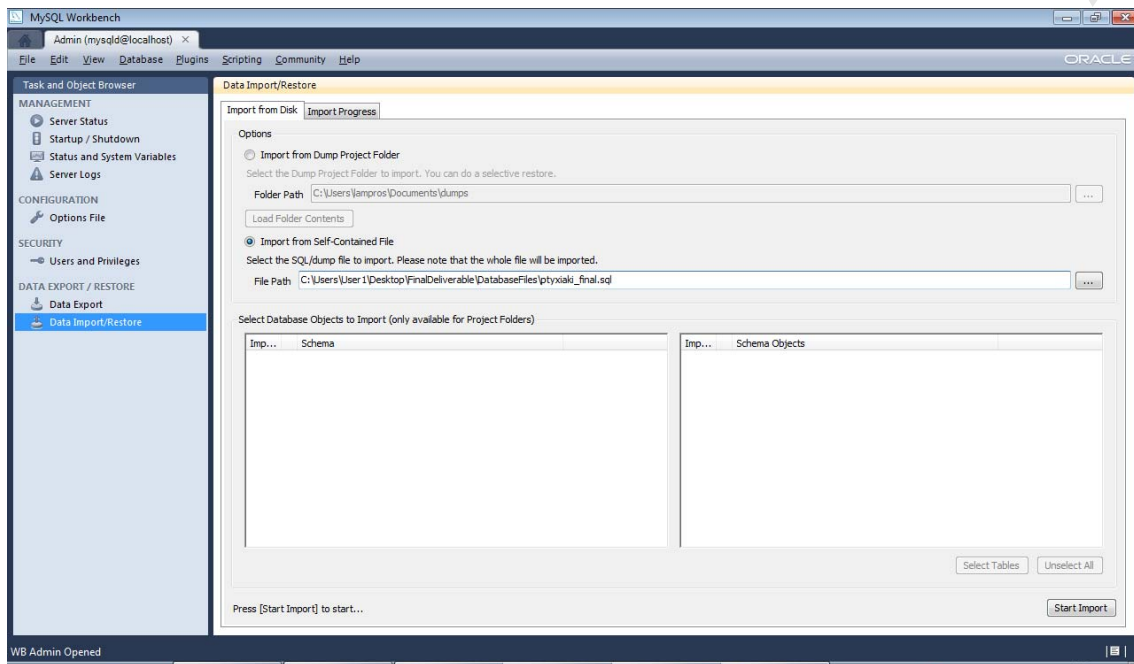
Για να χρησιμοποιηθεί το υπολογιστικό σύστημα απαιτείται η εγκατάσταση του εξυπηρετητή MySQL. Στα πλαίσια της ανάπτυξης του συστήματος χρησιμοποιήθηκε η δωρεάν έκδοση του εξυπηρετητή (Community Edition). Για την εγκατάσταση της βάσης δεδομένων απαιτείται κάποιο πρόγραμμα πρόσβασης στη βάση. Κατά την ανάπτυξη του έργου χρησιμοποιήθηκε το MySQL Workbench.

Το πρόγραμμα MySQL Server Community Edition μπορούμε να το μεταφορτώσουμε από το δικτυακό τόπο <http://dev.mysql.com/downloads/mysql/>. Το αποθηκεύουμε στον Η/Υ που θα χρησιμοποιηθεί και το εγκαθιστούμε ακολουθώντας τα βήματα εγκατάστασης και επιλέγοντας τις εξ'ορισμού (default) επιλογές. Κατά την εγκατάσταση μας ζητείται να ορίσουμε κωδικό για τον root χρήστη. Στο τέλος της εγκατάστασης (βλ. παρακάτω εικόνα) επιλέγουμε ο εξυπηρετητής να εγκατασταθεί ως υπηρεσία (Service) των Windows για να εκκινεί αυτόματα μετά από κάθε επανεκκίνηση του Η/Υ.

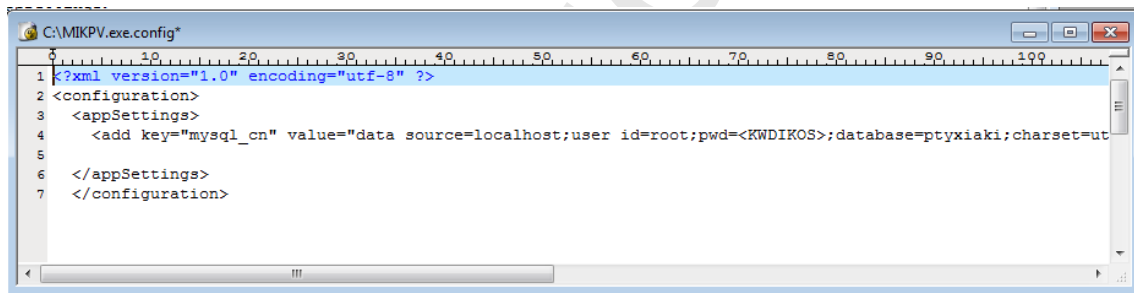


Το πρόγραμμα MySQL Workbench μπορούμε να το μεταφορτώσουμε από το δικτυακό τόπο <http://dev.mysql.com/downloads/workbench/>. Το αποθηκεύουμε στον Η/Υ που θα χρησιμοποιηθεί και το εγκαθιστούμε ακολουθώντας τα βήματα εγκατάστασης και επιλέγοντας τις εξ'ορισμού (default) επιλογές.

Αφού τα 2 προγράμματα εγκατασταθούν ενεργοποιούμε μία νέα σύνδεση από το MySQL Workbench προς τον εξυπηρετητή, δημιουργούμε μία νέα βάση με το όνομα «ptychiaki» και στη συνέχεια με την επιλογή import εγκαθιστούμε κάποια από τις εκδόσεις του SQL κώδικα που παρατίθενται στο προηγούμενο παράρτημα.



Αφού εγκατασταθεί η βάση στο configuration file της εφαρμογής θα πρέπει να συμπληρωθεί ο κωδικός του root χρήστη ώστε η εφαρμογή να μπορεί να συνδεθεί στη βάση δεδομένων.



3 Αποσπάσματα κώδικα της εφαρμογής

Το παρακάτω κομμάτι κώδικα χρησιμοποιείται για την αναγνώριση και τη διατήρηση πληροφοριών για τον τρέχοντα χρήστη.

```
public bool Login(string username, string password)
{
    bool result = false;
    try
    {
        string qry = string.Format("select userid,level,username from users where username='{0}' and
userpassword='{1}' ", username, password);
        DataSet ds = qsql.FreeQueryDataSet(qry);
        if ((ds.Tables.Count > 0) && (ds.Tables[0].Rows.Count > 0))
        {
            result = true;

            Program.userid = Convert.ToInt32(ds.Tables[0].Rows[0].ItemArray[0].ToString());
            Program.level = Convert.ToInt32(ds.Tables[0].Rows[0].ItemArray[1].ToString());
            Program.username = ds.Tables[0].Rows[0].ItemArray[2].ToString();
        }
    }
    catch
    {
        result = false;
    }
    return result;
}
```

Η πληροφορία σχετικά με το επίπεδο χρήστη αποθηκεύεται στη μεταβλητή level της κλάσης program ώστε να χρησιμοποιηθεί σε έλεγχοι πρόσβασης όπως ο παρακάτω:

```
private bool levelCheck(){
    if (Program.level < 3)
    {
        MessageBox.Show("Η λειτουργία αυτή είναι διαθέσιμη για χρήστες με επίπεδο >=3\ηΕπίπεδο
τρέχοντος χρήστη: " + Program.level);
        return false;
    }
    return true;
}
```

Το παρακάτω κομμάτι κώδικα εισάγει μία νέα αντιστοίχιση ΣΠΣ με ανταλλακτικό όταν ο χρήστης επιλέξει την εν λόγω λειτουργία. Η αποθήκευση στη βάση γίνεται με χρήση κατάλληλου sql query (insert).

```
private void button1_Click(object sender, EventArgs e)
{
    int amount=0;
    try{
        amount=Int32.Parse(textBox1.Text);
    }
    catch{
        MessageBox.Show("Εισάγετε έγκυρη ποσότητα");
        return;
    }

    if (comboBox1.SelectedIndex > -1)
    {
        SQLCRUD queryMachine = new SQLCRUD();
        string qry = string.Format("insert into part_sps (partid,spsid,part_sps_amount) values ('{0}','{1}','{2}");
        queryMachine.FreeQuery(qry);
        FillTable();
        populateSPS();
        updateAmountOfPart(objectID);
    }
    else
    {
        MessageBox.Show("Επιλέξτε ΣΠΣ");
    }
}
```

Το παρακάτω απόσπασμα κώδικα εκτελείται όταν ο χρήστης επιλέξει την ολοκλήρωση ενός ΣΠΣ. Πρώτα καλείται η συνάρτηση checkStock για να ελεγχθεί αν υπάρχει επαρκές απόθεμα για τα ανταλλακτικά που χρησιμοποιήθηκαν. Στη συνέχεια καλούνται με τη σειρά

- η συνάρτηση updateStock για να ανανεωθεί το τρέχον απόθεμα όλων των ανταλλακτικών που συμμετείχαν στο ΣΠΣ
- η συνάρτηση updateAmount για να γίνει εκ νέου υπολογισμός της προβλεπόμενης απαιτούμενης ποσότητας κάθε ανταλλακτικού μέχρι το τέλος του χρόνου
- η συνάρτηση updateDate για να οριστεί η επόμενη ημερομηνία εκτέλεσης του ΣΠΣ με βάση την επιλογή του χρήστη
- η συνάρτηση createHistory ώστε να δημιουργηθεί μία νέα εγγραφή ιστορικού

Τέλος τυπώνεται ενημερωτικό μήνυμα.


```
private void button1_Click(object sender, EventArgs e)
{
    if (checkStock())
    {
        updateStock();
        updateAmount();
        updateDate(objectID, dateTimePicker1.Value);
        createHistory();
        MessageBox.Show("Η Ολοκλήρωση του ΣΠΣ καταχωρήθηκε επιτυχώς. Τα αποθέματα και οι απαιτούμενες ποσότητες των συμμετεχόντων ανταλλακτικών ανανεώθηκαν κατάλληλα.");
        this.Close();
    }
}
```