



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΑΓΡΟΝΟΜΩΝ – ΤΟΠΟΓΡΑΦΩΝ ΜΗΧΑΝΙΚΩΝ
ΔΠΜΣ «ΓΕΩΠΛΗΡΟΦΟΡΙΚΗ»

**Σύστημα Υποστήριξης Αποφάσεων
για την Διαχείριση Στόλου Οχημάτων**

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

του

ΝΙΚΟΛΑΟΥ ΧΡΥΣΟΠΟΥΛΟΥ

Επιβλέπων : Τίμος Σελλής
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2009

.....
ΝΙΚΟΛΑΟΣ ΧΡΥΣΟΠΟΥΛΟΣ

Λγός (ΠΒ)

MsC in Computer Science

© 2009 – All rights reserved

Περίληψη

Ο σκοπός της μεταπτυχιακής εργασίας ήταν η ανάπτυξη μεθοδολογίας για την υποστήριξη αποφάσεων στην διαχείριση στόλου οχημάτων. Η Διαχείριση ενός στόλου οχημάτων, αποτελεί ένα από τα πλέον σημαντικά θέματα, που αφορούν στην υπηρεσίες θέσης (Location Based Services – LBS) και στις Βάσεις Δεδομένων Κινουμένων Αντικειμένων (Moving Objects Databases – MOD). Με τη μεθοδολογία αυτή μια επιχείρηση επιτυγχάνει τη διαχείριση των οχημάτων της (δρομολόγια – προτεραιότητα εξυπηρέτησης – δυνατότητα παράλληλης εξυπηρέτησης – συντήρηση) με τέτοιο τρόπο που να αυξήσει το κέρδος και την αποτελεσματικότητά της.

Από την άλλη ένα Σύστημα Υποστήριξης Αποφάσεων (Decision Support System – DSS) μας δίνει επιλογές και λύσεις για την ορθολογική και σωστή χρησιμοποίηση των πόρων μας. Μας δίνει τη δυνατότητα να λαμβάνουμε αποφάσεις γρήγορα και έγκαιρα. Οι πληροφορίες διαμοιράζονται στους ενδιαφερόμενους και παρέχουν όλα τα απαραίτητα δεδομένα για την διαχείριση των οχημάτων.

Ο συνδυασμός λοιπόν των δύο αυτών θεμάτων, αποτελεί βασικό εργαλείο για επιχειρήσεις, που εκτελούν μεταφορές συνδράμοντας με τα μέσα τους, άλλες επιχειρήσεις ή εκτελούν με τον στόλο τους τις μεταφορές της ίδιας της επιχείρησης.

Λέξεις Κλειδιά: <<Σύστημα Υποστήριξης Αποφάσεων, Διαχείριση Στόλου Οχημάτων, Βάσεις Δεδομένων Κινουμένων Αντικειμένων, Χωρικές Βάσεις Δεδομένων, μεταφορές>>

Abstract

The scope of this thesis was the development of a methodology for the decision support of a fleet management. Fleet Management is an important issue for the Location Based Services and the Moving Object Databases. With the proposed methodology, a company may manage its vehicles (routing – priority of service – parallel service – maintenance) in order to increase its profit and its efficiency.

On the other hand, a Decision Support System gives choices and solutions for the right and rational use of the resources. It gives the opportunity of a fast and right-on-time decision making. All the information is distributed to the users that need them and gives them all the needed data for the fleet management.

Both these issues combined, are an important tool for companies, who execute transports, either for themselves or by giving their sources to other companies.

Keywords: <<Decision Support System, Fleet Management, Moving Object Databases, Spatial Databases, transports>>

Πίνακας περιεχομένων

1	Εισαγωγή.....	1
1.1	Λήψη Αποφάσεων στη Διαχείριση Στόλου Οχημάτων	1
1.2	Αντικείμενο Μεταπτυχιακής Εργασίας.....	3
1.2.1	Συνεισφορά	4
1.3	Οργάνωση κειμένου.....	4
2	Σχετικές εργασίες.....	7
2.1	Συστήματα Υποστήριξης Αποφάσεων.....	7
2.2	Διαχείριση Κινουμένων Αντικειμένων με χρήση εργαλείων GIS	8
2.3	Εμπορικές Λύσεις Διαχείρισης Στόλου Οχημάτων	10
3	Θεωρητικό υπόβαθρο.....	13
3.1	Συστήματα Υποστήριξης Αποφάσεων.....	13
3.2	Διαχείριση Στόλου Οχημάτων	16
3.3	Γεωγραφικά Συστήματα Πληροφοριών.....	20
3.4	Χωρικά Συστήματα Υποστήριξης Αποφάσεων	23
3.5	Αντικειμενοστραφής Μεθοδολογία και UML.....	26
3.6	Χωρικές Βάσεις Δεδομένων	30
4	Ανάλυση Απαιτήσεων Συστήματος.....	33
4.1	Αρχιτεκτονική.....	33
4.2	Περιγραφή Λειτουργιών	35
4.2.1	Υποσύστημα Καταχώρησης - Ταξινόμησης.....	35
4.2.2	Υποσύστημα Κινήσεων.....	36
4.2.3	Υποσύστημα Συντήρησης.....	38
4.2.4	Υποσύστημα Απόσυρσης.....	39
4.2.5	Υποσύστημα Τερματικών.....	40
4.3	Μοντέλο Οντοτήτων Συσχετίσεων	41
4.4	Βάση Δεδομένων	43
5	Υλοποίηση	49
5.1	Πλατφόρμες και προγραμματιστικά εργαλεία	49

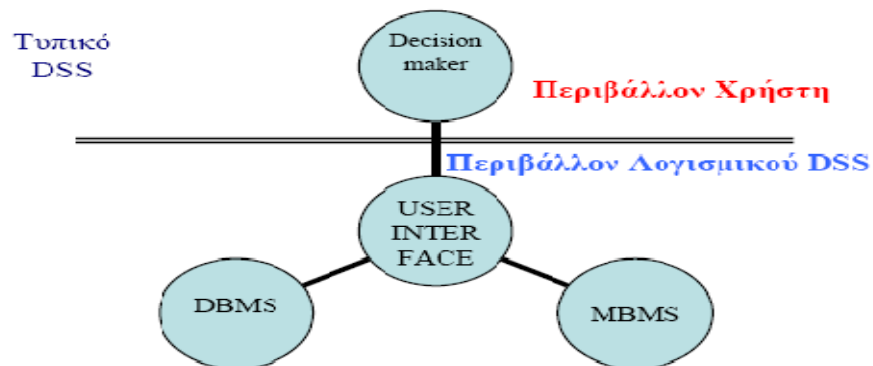
5.1.1	<i>Προσαρμογή πλατφόρμας ανάπτυξης</i>	49
5.1.2	<i>Χρήση των Google Maps σε συνδυασμό με MySQL/PHP</i>	52
5.1.3	<i>Υλοποίηση Βάσης Δεδομένων</i>	59
6	Έλεγχος	67
6.1	Μεθοδολογία ελέγχου	67
6.2	Αναλυτική παρουσίαση ελέγχου	69
7	Επίλογος	83
7.1	Σύνοψη και συμπεράσματα	83
7.2	Μελλοντικές επεκτάσεις	84
8	Βιβλιογραφία	85

1

Εισαγωγή

1.1 Λήψη Αποφάσεων στη Διαχείριση Στόλου Οχημάτων

Στις αρχές του 1970 ξεκίνησε η ακαδημαϊκή έρευνα πάνω στα Συστήματα Υποστήριξης Αποφάσεων (ΣΥΑ). Πολλοί ήταν οι ορισμοί που δόθηκαν για αυτά, όλοι όμως είχαν κοινό τόπο: Δεν εστιάζουν σε μια περιοχή λήψης αποφάσεων συγκεκριμένου εύρους και , κυρίως, δεν υποκαθιστούν αλλά υποστηρίζουν τον χρήστη στη διαδικασία λήψης αποφάσεων. Τα ΣΥΑ υποστηρίζουν αποφάσεις σε προβλήματα δομημένα, ημιδομημένα ή και κακώς δομημένα, και παρέχουν διεπαφή με τον χρήστη καθώς και σύνδεση με μια Βάση Δεδομένων, που αποθηκεύει και επεξεργάζεται τα δεδομένα, σε συνδυασμό με την βάση μοντέλων (Εικόνα 1).



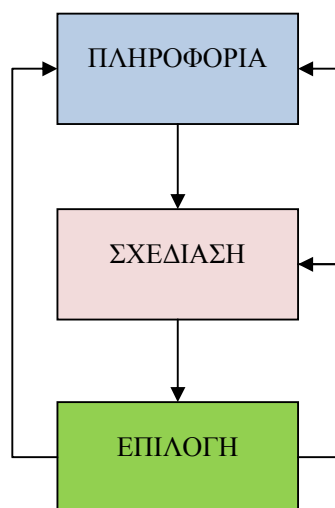
Εικόνα 1: Τυπική διάρθρωση ενός ΣΥΑ

Η Διαχείριση Στόλου (Fleet Management) παρέχει σε μια επιχείρηση -ανεξάρτητα από το μέγεθός της- τη δυνατότητα να ελέγχει το σύνολο των οχημάτων και μεταφορικών μέσων της και, κατά συνέπεια, να κατανέμει καλύτερα τους πόρους της και να μειώνει το λειτουργικό κόστος. Διαχείριση Στόλου επομένως είναι το σύνολο των τεχνολογιών και των συστημάτων το οποίο επιτρέπει σε μία επιχείρηση να έχει πλήρη έλεγχο των οχημάτων της. Ο τρόπος λειτουργίας ενός τέτοιου συστήματος αποσκοπεί στη βελτιστοποίηση αρκετών επιμέρους επιχειρηματικών διαδικασιών, στο διαχειριστικό έλεγχο, που οδηγούν τελικά, στο επιθυμητό αποτέλεσμα, δηλαδή στη μείωση του κόστους και την καλύτερη κατανομή των πόρων της επιχείρησης.

Βασικό στοιχείο της Διαχείρισης Στόλου, είναι η ύπαρξη χωρικών δεδομένων, που επηρεάζουν την όλη διαδικασία. Τα αντικείμενα διαχείρισης, δηλαδή τα οχήματα, δεν είναι σταθερά, αλλά κινούνται, με αποτέλεσμα κάθε στιγμή η θέση τους να μεταβάλλεται. Επομένως, ένα τυπικό ΣΥΑ δεν μπορεί να ανταπεξέλθει πλήρως στις ανάγκες μιας τέτοιας διεργασίας. Χρειάζεται να επεκταθεί ώστε να συμπεριλάβει και την χωρική πληροφορία.

Έτσι οδηγούμαστε στα Χωρικά Συστήματα Υποστήριξης Αποφάσεων (ΧΣΥΑ). Ένας πρώτος ορισμός τους μας λέει ότι είναι τα υπολογιστικά συστήματα τα οποία είναι σχεδιασμένα ώστε να υποστηρίζουν έναν χρήστη ή μια ομάδα χρηστών στην δημιουργία λύσεων σε ημιδομημένα ή κακώς δομημένα χωρικά προβλήματα με την χρήση χωρικών δεδομένων, επιχειρηματικών μοντέλων, εργαλείων λογισμικού και έμπειρων συστημάτων. Βασικό σκοπό έχουν κυρίως την αύξηση της αποτελεσματικότητας του λήπτη αποφάσεων, παρά την αύξηση της αποδοτικότητας της λήψης αποφάσεων, πράγμα που επιτυγχάνεται με την ενσωμάτωση και αλληλεπίδραση των αποφάσεων του χρήστη με τις εφαρμογές και τα μοντέλα του συστήματος καθώς και με την ευκολία χρήσης του.

Τα Γεωγραφικά Συστήματα Πληροφοριών (ΓΣΠ) χρησιμοποιούνται όλο και περισσότερο τα τελευταία χρόνια, και η Διαχείριση Στόλου αποτελεί μια από τις πλέον διαδεδομένες εφαρμογές τους. Πως όμως θα διευκολυνθεί η λήψη αποφάσεων, και πιο συγκεκριμένα η εργασία ενός λήπτη αποφάσεων από την χρησιμοποίηση ενός συστήματος με δυνατότητες ΓΣΠ; Η διαδικασία λήψης αποφάσεων, διακρίνεται σε 3 φάσεις (πληροφορία, σχεδίαση, επιλογή). Τα ΓΣΠ έχουν τη δυνατότητα να αλληλεπιδρούν ποικιλοτρόπως με τις φάσεις αυτές, και έτσι να οδηγούμαστε σταδιακά στα ΧΣΥΑ.



Εικόνα 2: Η διαδικασία λήψης αποφάσεων

1.2 Αντικείμενο Μεταπτυχιακής Εργασίας

Η μεταπτυχιακή εργασία είναι μια web εφαρμογή, για την υποστήριξη αποφάσεων στη διαχείριση στόλου οχημάτων. Είναι μια εφαρμογή που περιέχει χωρικό κομμάτι αλλά και μια καθαρή εφαρμογή Βάσεων Δεδομένων. Βοηθά μια επιχείρηση στο να διαχειρίζεται με σωστό τρόπο τα οχήματά της, να έχει μια συνολική και εποπτική εικόνα αυτών με πολλές δυνατότητες επεκτάσεων. Δίνει τη δυνατότητα να υπάρχουν μέσω web όλες οι διαδικασίες που απαιτούνται για να λάβουν χώρα οι διαδικασίες για την εκτέλεση των διάφορων λειτουργιών, με αποτέλεσμα να υπάρχει κέρδος χρόνου και κατά συνέπεια και εξοικονόμηση πόρων (υλικών αλλά και ανθρώπινου δυναμικού).

Βασικό κίνητρο για την υλοποίησή της είναι η προσπάθεια μοντελοποίησης του τρόπου που λειτουργούν τα γραφεία κίνησης οχημάτων τόσο στον ιδιωτικό τομέα, όσο και στον δημόσιο και κυρίως στον Στρατιωτικό. Υπάρχει μια συγκεκριμένη μεθοδολογία, η οποία μοντελοποιούμενη μέσω της εφαρμογής και αποκτώντας και νέες επιλογές και διευκολύνσεις, δίνει μια νέα διάσταση στον τρόπο λειτουργίας των Γραφείων Κινήσεων.

Μέσω της μοντελοποίησης αυτής, αναγνωρίστηκαν τα υποσυστήματα της εφαρμογής και οι απαιτήσεις ενός εκάστου. Με χρήση κατάλληλων διαγραμμάτων σχεδιάστηκε ο τρόπος λειτουργίας σε διάφορες ενδεχόμενες καταστάσεις και ο τρόπος αντιμετώπισης αυτών.

Αφού, λοιπόν, αναγνωρίστηκαν οι απαιτήσεις του συστήματος, σχεδιάστηκε η ΒΔ, υλοποιήθηκε, και στη συνέχεια σχεδιάστηκε το web περιβάλλον το οποίο είναι και αυτό στο

οποίο έχει πρόσβαση ο τελικός χρήστης. Έγινε χρήση λογισμικού ανοιχτού κώδικα, μιας και πλέον, λογισμικό τέτοιας μορφής είναι αυτό που προτιμάται και με ένα φανταστικό σενάριο, έγινε έλεγχος της εφαρμογής και του τρόπου λειτουργίας της. Αναγνωρίστηκαν ελλείψεις και βρέθηκαν μελλοντικές επεκτάσεις της, ώστε να αποτελεί ένα πλήρες εργαλείο διαχείρισης στόλου οχημάτων.

1.2.1 Συνεισφορά

Η συνεισφορά της μεταπτυχιακής εργασίας συνοψίζεται ως εξής:

1. Αρχικά έγινε μελέτη όλων των διαθέσιμων τεχνολογιών που αφορούν το τομέα της εργασίας. Πιο συγκεκριμένα μελετήθηκαν τα αφορούντα τις χωρικές βάσεις δεδομένων, τα συστήματα υποστήριξης αποφάσεων, τη διαχείριση στόλων οχημάτων για να φτάσουμε τελικά στο πλαίσιο λειτουργίας των χωρικών συστημάτων υποστήριξης αποφάσεων.

2. Έγινε χρήση του συνδυασμού Tomcat-MySQL-PHP και Google Maps, με κάποιες ιδιαιτερότητες στην εγκατάσταση των πρώτων τριών με τρόπο ώστε να συνεργάζονται χωρίς προβλήματα.

3. Η χρήση των Google Maps απαιτεί τα χωρικά δεδομένα σε συγκεκριμένη μορφή. Έτσι υλοποιήθηκε τρόπος μετατροπής των συντεταγμένων από μορφή Well Known Text (WKT) σε μορφή lat και long για να μπορούν να χρησιμοποιηθούν από τους χάρτες της Google.

4. Εφαρμόστηκε εναλλακτική υλοποίηση των χωρικών συναρτήσεων που ακόμα δεν υποστηρίζονται από το ΣΔΒΔ της MySQL. Πιο συγκεκριμένα η συνάρτηση εύρεσης της απόστασης δυο σημείων (distance) καθώς και της εύρεσης του πλησιέστερου γείτονα, υλοποιήθηκε με τον εναλλακτικό αυτό τρόπο.

1.3 Οργάνωση κειμένου

Η εργασία έχει την ακόλουθη δομή: Στο επόμενο κεφάλαιο παρουσιάζονται τόσο εργασίες όσο και εμπορικές λύσεις με περιεχόμενο σχετικό με την υλοποιούμενη εφαρμογή. Στο τρίτο κεφάλαιο της εργασίας γίνεται βιβλιογραφική μελέτη και ανάλυση του θεωρητικού υπόβαθρου των τεχνολογιών που χρησιμοποιήθηκαν. Το 4^ο Κεφάλαιο, είναι εκείνο στο οποίο αναλύονται οι απαιτήσεις του συστήματος και υλοποιείται η Βάση Δεδομένων. Βασιζόμενοι στην αρχιτεκτονική, διακρίνουμε τα υποσυστήματα της εφαρμογής, με τις ιδιαιτερότητές του το καθένα, τα αναλύουμε και καταλήγουμε στο Μοντέλο Οντοτήτων Συσχετίσεων της Βάσης Δεδομένων. Στη συνέχεια λαμβάνει χώρα ο φυσικός σχεδιασμός της και τελικά η δημιουργία της. Στο 5^ο Κεφάλαιο ασχολούμαστε με την περιγραφή των εργαλείων που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής καθώς και των πλατφόρμων ανάπτυξης,

ως επί το πλείστον λογισμικό ανοιχτού κώδικα. Πλήρης ανάλυση της εφαρμογής γίνεται στο 6^ο Κεφάλαιο, όπου αναπτύσσεται ένα σενάριο και μέσω αυτού γίνεται έλεγχος των αποτελεσμάτων της εφαρμογής. Συμπεράσματα, ελλείψεις και μελλοντικές επεκτάσεις της εφαρμογής αναλύονται στο 7^ο Κεφάλαιο, και η βιβλιογραφία στην οποία στηριχθήκαμε στο 8^ο.

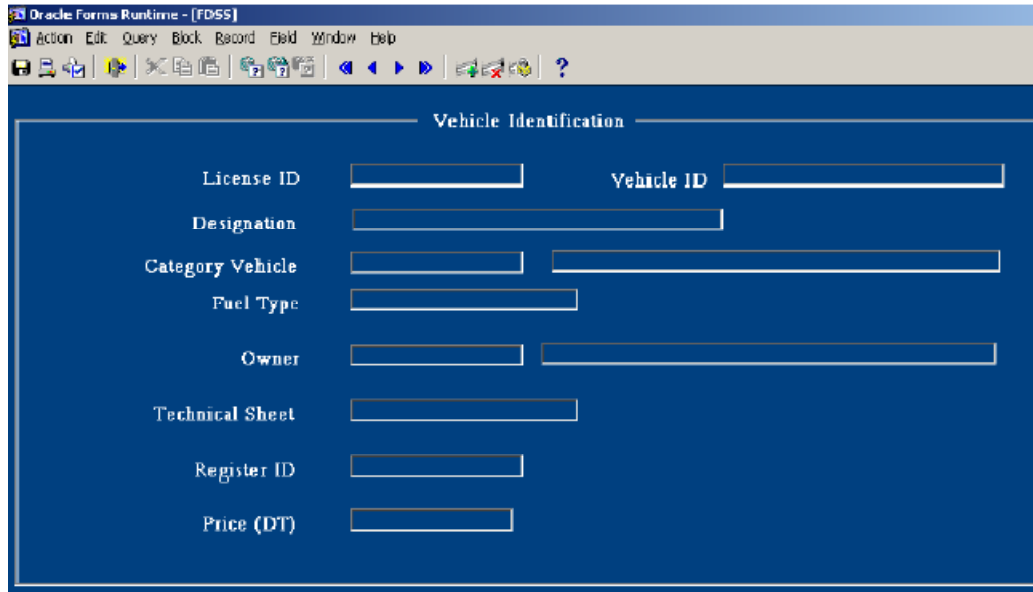
2

Σχετικές εργασίες

Εύκολα γίνονται κατανοητοί και ευδιάκριτοι οι θεματικοί χώροι πάνω στους οποίους θα κινηθεί η εργασία: Ασχολούμαστε με ένα Σύστημα Υποστήριξης Αποφάσεων που αφορά τη Διαχείριση Στόλου Οχημάτων, συνεπώς εμπεριέχεται και το χωρικό κομμάτι μιας και η χωρική πληροφορία αποτελεί σημαντικό κομμάτι στη διαχείριση. Σαφώς και υπάρχει αρκετό υλικό, τόσο σε επίπεδο έρευνας, όσο και σε εμπορικό επίπεδο: λύσεις που απευθύνονται σε εταιρίες για τη διαχείριση του στόλου οχημάτων τους. στο χώρο αυτό, που ανήκει σε μια ή και περισσότερες από τις θεματικές ενότητες του χώρου μέσα στον οποίο κινούμαστε.

2.1 Σύστημα Υποστήριξης Αποφάσεων

Ο Chiheb Saidane στην εργασία του για το Naval Postgraduate School του Monterrey, ανέπτυξε μια εφαρμογή για τη διαχείριση στόλου στρατιωτικών οχημάτων. Μια εφαρμογή πολύ σχετική με την υλοποιούμενη. Δίνεται η δυνατότητα διαχείρισης των στρατιωτικών οχημάτων, με τρόπο παρεμφερή με το δικό μας. Η εφαρμογή έχει 4 υποσυστήματα όπως και η δική μας, αλλά δεν ασχολείται καθόλου με το χωρικό κομμάτι. Ασχολείται με τα διαδικαστικά, αλλά όχι με την υλοποίησή τους. Είναι μια client-server εφαρμογή (όχι web εφαρμογή) και υλοποιείται με τη βοήθεια εργαλείων ανάπτυξης (oracle forms) (Εικόνα 3). Στην εργασία παρουσιάζεται περισσότερο η μεθοδολογία και λιγότερο η εφαρμογή καθαυτή. Είναι δε πλήρως προσαρμοσμένη στις στρατιωτικές απαιτήσεις.

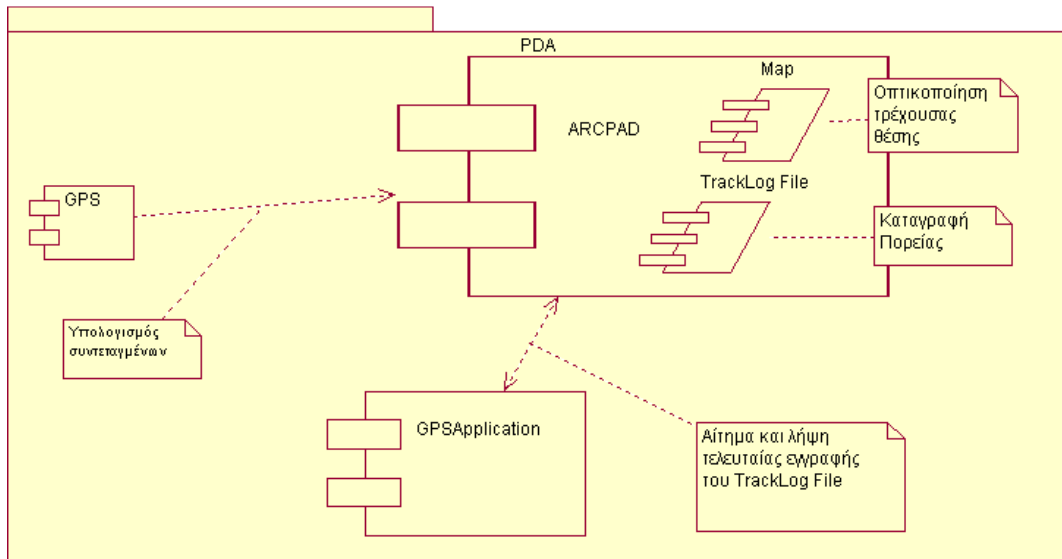


Εικόνα 3: Παράδειγμα φόρμας

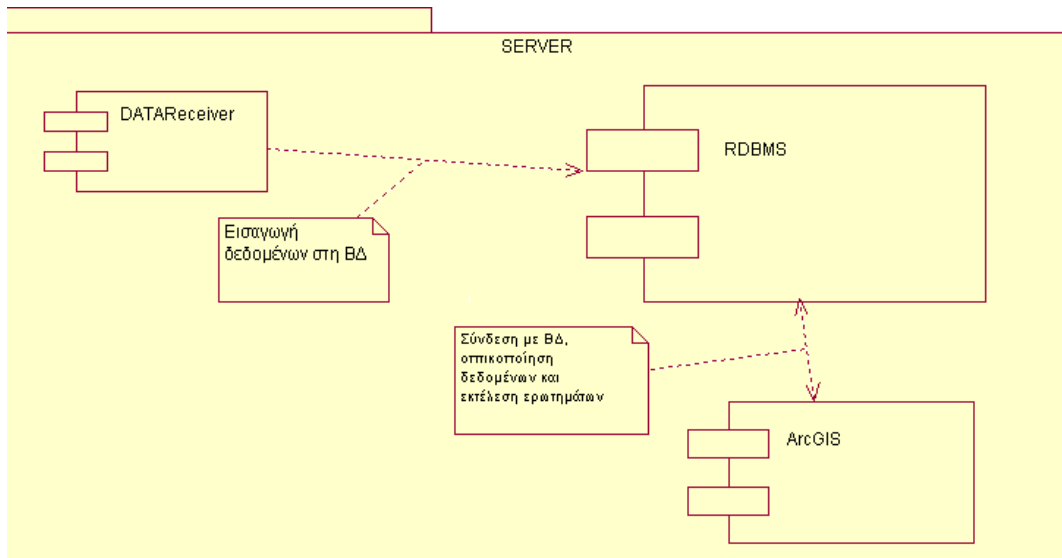
2.2 Διαχείριση Κινομένων Αντικειμένων με χρήση εργαλείων

GIS

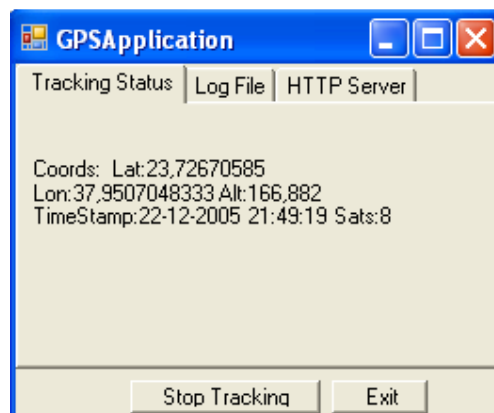
Οι Θεοδωρίδης-Χρυσόπουλος (2006) ανέπτυξαν μια εφαρμογή παρακολούθησης της κίνησης οχημάτων ή ανθρώπων με χρήση υπάρχουσας τεχνολογίας από την ESRI (ArcPAD στο τερματικό, ArcGIS στο server) (Εικόνα 4 και Εικόνα 5). Το κινούμενο αντικείμενο έστελνε ανά τακτά χρονικά διαστήματα το στίγμα του στον server (Εικόνα 6) και η θέση του εμφανιζόταν μέσω του ArcGIS στο χάρτη (Εικόνα 7). Χωρική πληροφορία με τη στενή έννοια του όρου δεν υπήρχε, μιας και οι συντεταγμένες αποθηκεύονταν όχι ως χωρικό δεδομένο, αλλά σαν αριθμός, που προβαλλόταν σε μια γεωαναφερόμενη εικόνα. Τα δε ερωτήματα προς τη βάση ήταν αποκλειστικά και μόνο τα απλά ερωτήματα που δίνει το ArcGIS τη δυνατότητα να εκτελέσουμε (Εικόνα 8). Δεν υπήρχε δυνατότητα ο κινητός χρήστης να έχει πρόσβαση στη ΒΔ μέσω web, αλλά η μόνη λειτουργία που είχε στη διάθεσή του, ήταν να αποστέλλει το στίγμα του. Επίσης, μιας και υπήρχε η απαίτηση ταυτόχρονης αποστολής στίγματος με την εμφάνιση της θέσης του κινητού σταθμού στο χάρτη, υπήρξε μια σύγκρουση του ArcPDA με την εφαρμογή αποστολής στιγμάτων. Το πρόβλημα λύθηκε με το να υπάρχει μικρή καθυστέρηση στην αποστολή και να στέλνεται στίγμα το οποίο δεν λαμβανόταν από το GPS κατευθείαν, αλλά από το log file που αποθήκευε την όλη διαδρομή. Καθυστέρηση η οποία όμως δεν ήταν σημαντική και δεν άλλαζε την ρεαλιστικότητα της εφαρμογής.



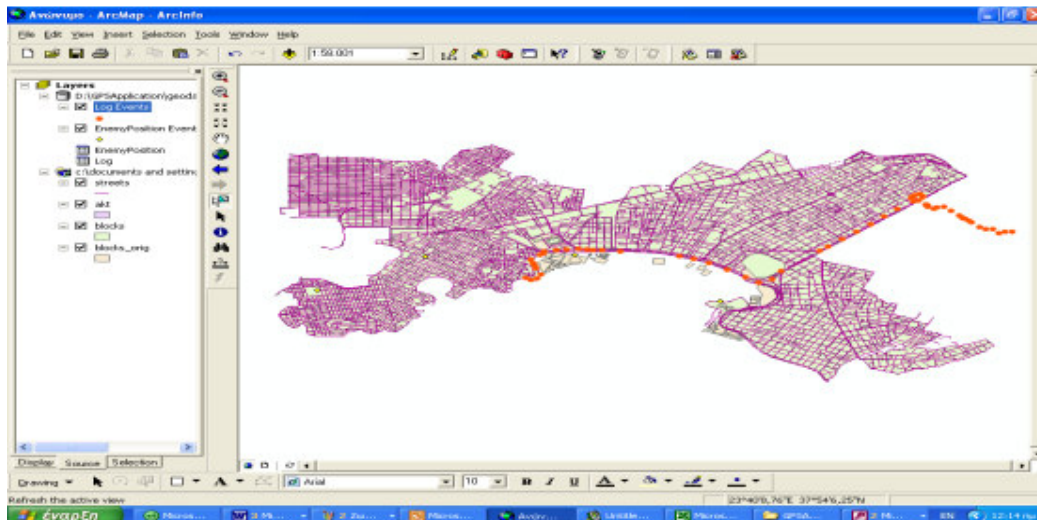
Εικόνα 4: Διαδικασίες στο τερματικό



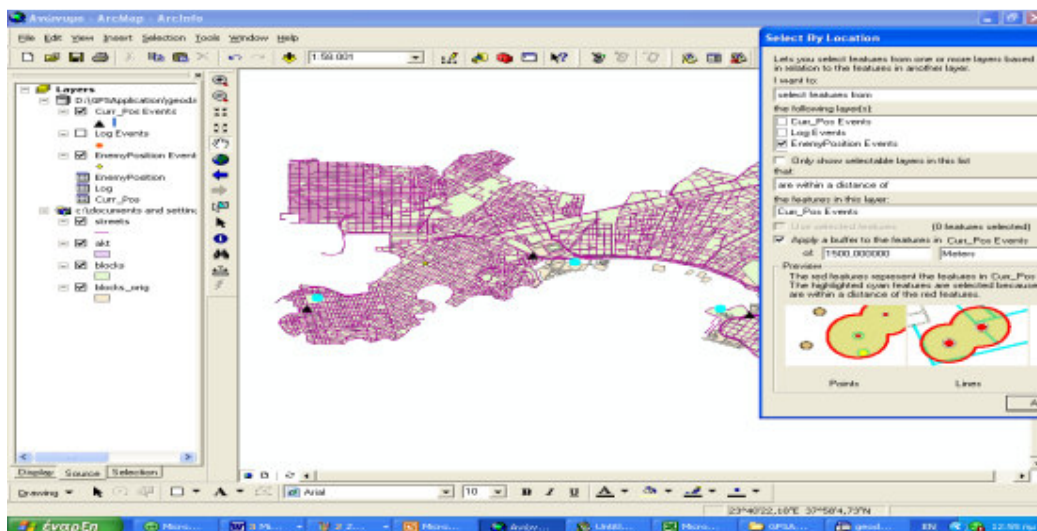
Εικόνα 5: Διαδικασίες στον εξυπηρετητή



Εικόνα 6: Αποστολή στίγματος



Εικόνα 7: Εμφάνιση πορείας



Εικόνα 8: Εκτέλεση ερωτημάτων

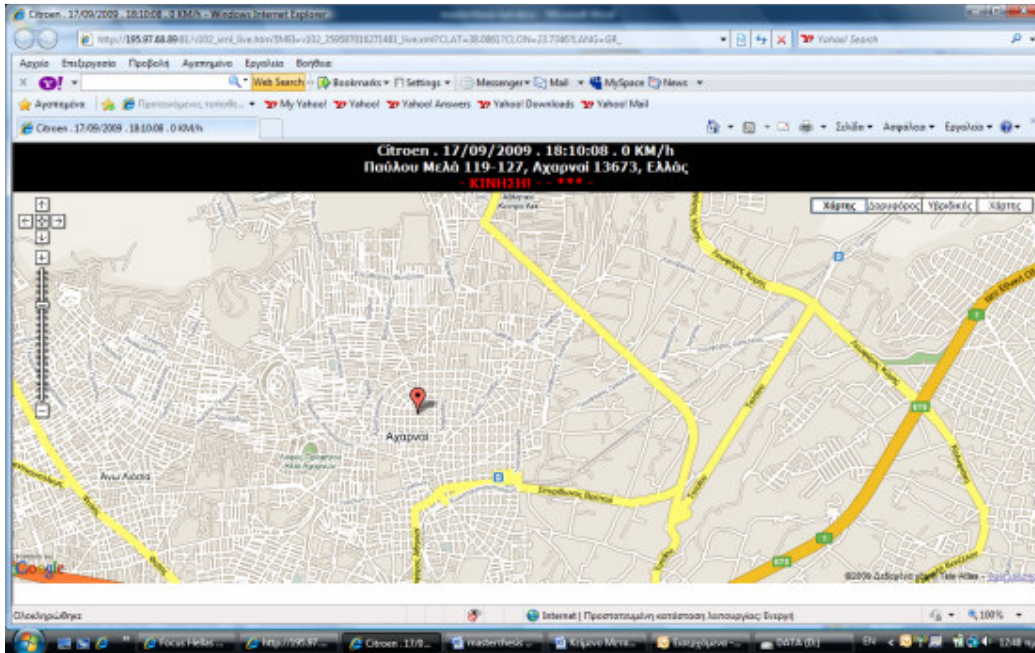
2.3 Εμπορικές Λύσεις Διαχείρισης Στόλου Οχημάτων

Στο εμπόριο υπάρχει πληθώρα λύσεων παρακολούθησης κίνησης οχημάτων. Η παρακολούθηση γίνεται με διάφορες από τις υπάρχουσες τεχνολογίες, που αναλύονται στο επόμενο κεφάλαιο, και οι υπηρεσίες που παρέχονται ποικίλουν ανάλογα με τη συνδρομή και το κόστος.

Μια πρώτη λύση είναι το GPSTracker (Εικόνα 9) της εταιρίας Focus Hellas (<http://gpstracker.gr/gps/index.jsp>). Οι παρεχόμενες υπηρεσίες συνοψίζονται στα παρακάτω:

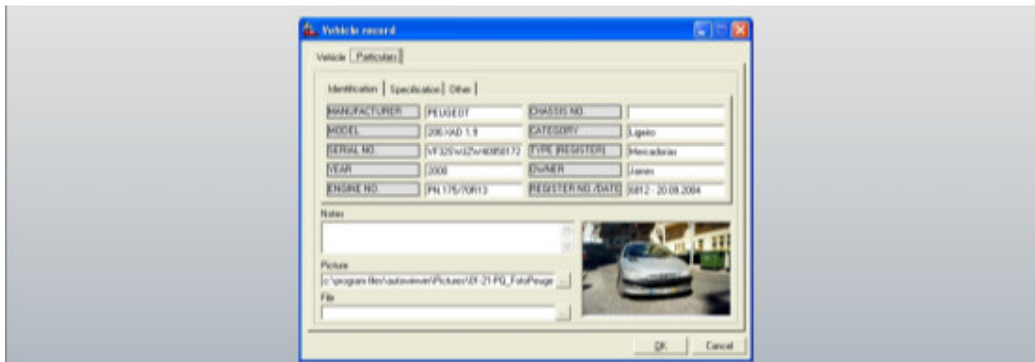
- i. Λειτουργεί με SMS ή GPRS
- ii. Ανανεώνει ανά 30 sec.

- iii. Εκτελεί έλεγχο ταχύτητας ή μετακίνησης χωρίς άδεια.
- iv. Εκτελεί έλεγχο κίνησης εντός προκαθορισμένης περιοχής.
- v. Δυνατότητα έκτακτης ειδοποίησης
- vi. Ιστορικό Μετακινήσεων
- vii. Επιπλέον ελέγχους (θερμοκρασία, καύσιμα κλπ).

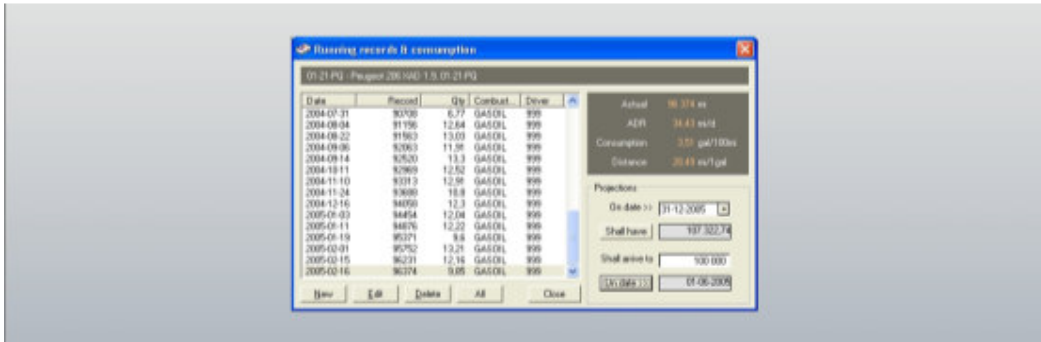


Εικόνα 9: GPS Tracker

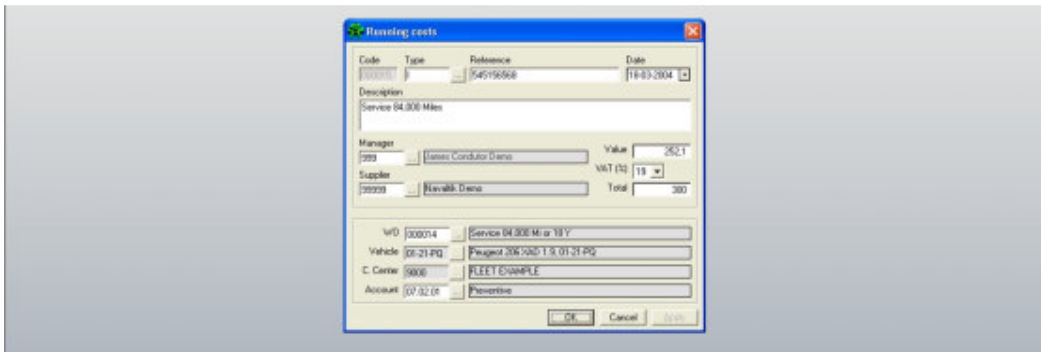
Μια ενδεικτική δεύτερη λύση που βρέθηκε είναι το λογισμικό της εταιρίας autowinwin (<http://www.autowinwin.com>). Αποτελεί κλασσική περίπτωση λογισμικού για τη διαχείριση στόλου οχημάτων, χωρίς να υπάρχει δυνατότητα οπτικοποίησης σε χάρτη. Λειτουργεί καθαρά διαχειριστικά και είναι μια εύκολη λύση, που παρέχει και δικαίωμα δωρεάν δοκιμής για ένα μήνα. Ορισμένα screenshots της εφαρμογής ακολουθούν (Εικόνα 10,Εικόνα 11,Εικόνα 12,Εικόνα 13).



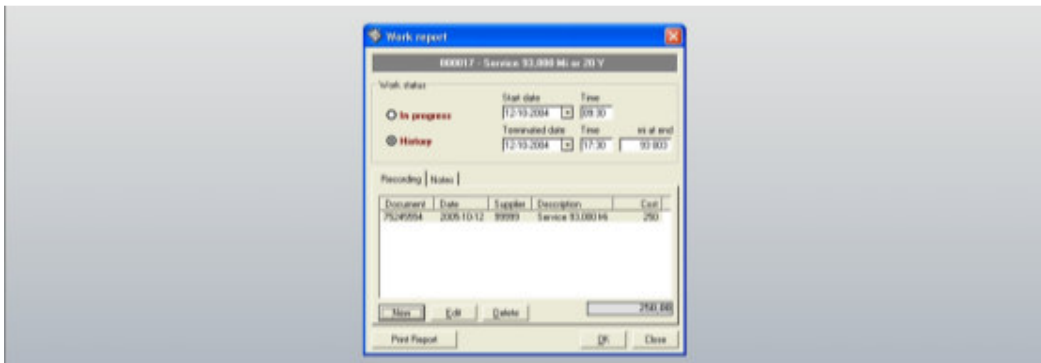
Εικόνα 10: Καρτέλα Οχήματος



Εικόνα 11: Κινήσεις και Καταναλώσεις



Εικόνα 12: Κόστη Συντήρησης



Εικόνα 13: Καρτέλα Συντήρησης

3

Θεωρητικό υπόβαθρο

Στην πορεία για την υλοποίηση ενός ΣΥΑ για την Διαχείριση Στόλου Οχημάτων, είναι αναγκαίο να κατανοηθούν πλήρως οι τεχνολογίες και μέθοδοι , που αποτελούν τα συστατικά κομμάτια του. Οι τεχνολογίες αυτές, έχουν ήδη αναφερθεί, και στη συνέχεια θα γίνει μια λεπτομερής παρουσίασή τους. Ξεκινώντας από τα παραδοσιακά ΣΥΑ και την Διαχείριση Στόλου Οχημάτων, θα συνεχίσουμε με τα ΓΣΠ και θα ολοκληρώσουμε με τα ΧΣΥΑ. Επίσης θα γίνει αναφορά στη Αντικειμενοστραφή μεθοδολογία και στη γλώσσα μοντελοποίησης UML που στοιχεία της θα χρησιμοποιηθούν, καθώς και στις Χωρικές Βάσεις Δεδομένων.

3.1 Συστήματα Υποστήριξης Αποφάσεων

Η λήψη των αποφάσεων είναι αποτέλεσμα σύνθετων διαδικασιών, που ξεφεύγουν από την ευθύνη ενός ατόμου. Η τελική απόφαση είναι μια διαδικασία συνεχούς και κυκλικής μελέτης των δεδομένων και των εναλλακτικών λύσεων.

Στην όλη διαδικασία επιλογής της τελικής απόφασης, αλληλεπιδραστικά συστήματα λογισμικού, τα οποία καλούνται ΣΥΑ, αποσκοπούν στη βελτίωση της διαδικασίας λήψης αποφάσεων, κυρίως σε χώρους προβλημάτων που χαρακτηρίζονται από χαμηλό βαθμό δόμησης.

Η χρήση των ΣΥΑ γίνεται για πολλού λόγους, σημαντικότεροι από τους οποίους είναι:

- i. Ποιότητα απόφασης
- ii. Βελτίωση επικοινωνίας

- iii. Μείωση κόστους
- iv. Αύξηση παραγωγικότητας
- v. Κέρδος χρόνου
- vi. Ικανοποίηση πελάτη και υπαλλήλων.

Ο αποφασίζων υποστηρίζεται από αναλυτικές μεθόδους και μοντέλα, ώστε να θέτει στόχους και να ορίζει τις εναλλακτικές λύσεις. Να αναλύει τις επιπτώσεις τους, να τις αξιολογεί και να επιλέγει την πλέον κατάλληλη. Τα ΣΥΑ, επομένως, υποστηρίζουν, δεν αντικαθιστούν. Παρέχουν πρόσβαση σε κατάλληλες Βάσεις Δεδομένων και σε αναλυτικά – ποσοτικά μοντέλα. Με την χρήση τους, διευρύνεται το πεδίο αντίληψης των χρηστών και προοδευτικά βελτιώνονται οι αποφάσεις.

Όπως αναφέρθηκε προηγουμένως (Εικόνα 2), η διαδικασία λήψης αποφάσεων, αποτελείται από τρία στάδια, τα οποία στη συνέχεια θα αναλυθούν:

i. Στο πρώτο στάδιο, το στάδιο της πληροφορίας, περιλαμβάνονται όλες οι διαδικασίες ελέγχου για την πλήρη ανάλυση της κατάστασης που επικρατεί, και για την οποία ο αποφασίζων, πρέπει να λάβει απόφαση. Δηλαδή εξετάζεται η πραγματικότητα, αναγνωρίζεται και ορίζεται το πρόβλημα.

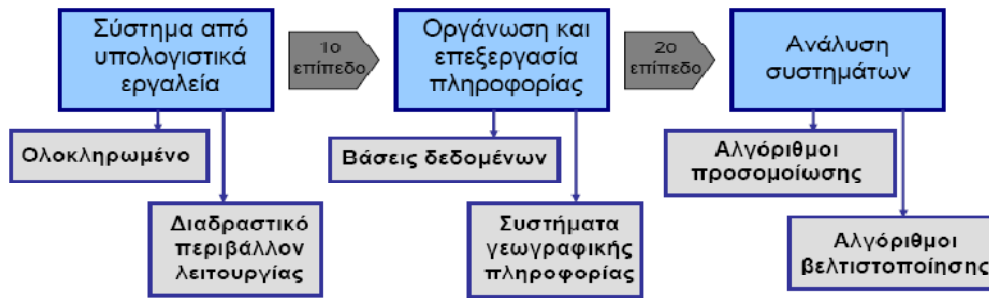
ii. Στο δεύτερο στάδιο, το στάδιο της σχεδίασης, ανακαλύπτονται, αναλύονται και αναπτύσσονται εναλλακτικές λύσεις, για το πρόβλημα που απαιτεί λύση. Χρησιμοποιείται κατάλληλο μοντέλο δεδομένων που περιλαμβάνεται στο ΣΥΑ.

iii. Στην τρίτη φάση, την φάση της επιλογής, επιλέγεται η καλύτερη λύση, που έχει αναλυθεί και αξιολογηθεί, με βάση κανόνες και κριτήρια.

Στον παραπάνω ,κατά Simon, ορισμό των 3 φάσεων, πρέπει να προστεθεί και μια ακόμη φάση, η φάση της ανάπτυξης, η οποία δίνει και την τελική λύση στο αρχικό πρόβλημα.

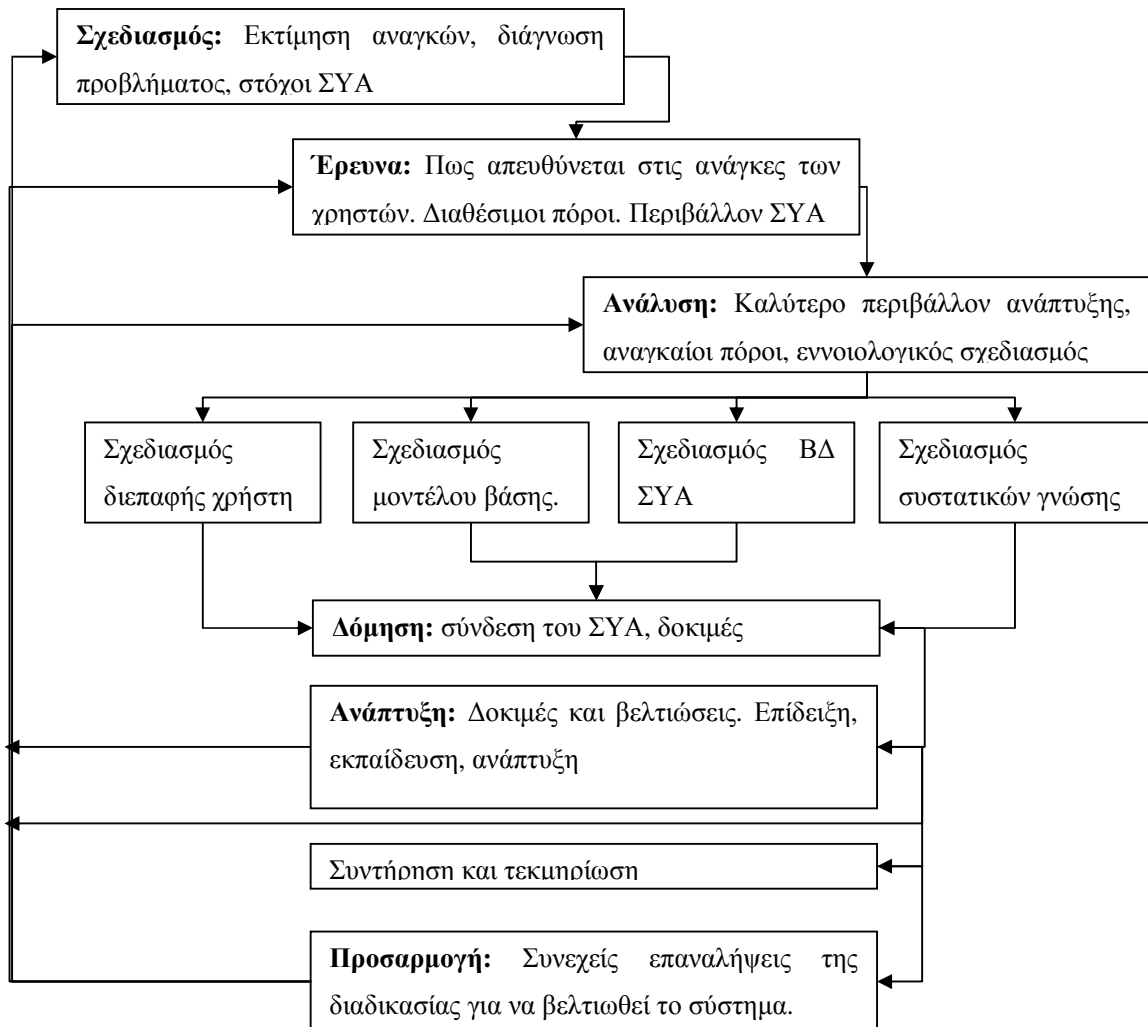
Με λίγα λόγια, ΣΥΑ είναι ένα σύστημα λογισμικού, που υποστηρίζει τη διαδικασία λήψης αποφάσεων, βοηθώντας στην κατανόηση των επιπτώσεων των αποφάσεων. Είναι επομένως ένα ολοκληρωμένο σύστημα από υπολογιστικά εργαλεία, με διαδραστικό περιβάλλον λειτουργίας, παρέχει δυνατότητες οργάνωσης και επεξεργασίας της πληροφορίας, περιέχει υπολογιστικά εργαλεία ανάλυσης συστημάτων και σχεδιάζεται με στόχο την υποβοήθηση στη λήψη αποφάσεων, μέσω διατύπωσης και λεπτομερούς μελέτης σειράς εναλλακτικών επιλογών (Εικόνα 14).

Για την μοντελοποίηση ενός σεναρίου, που εξετάζεται από ένα ΣΥΑ, χρησιμοποιούνται διάφορες μέθοδοι , δυναμικές ή στατικές. Πιο διαδεδομένη μέθοδος, είναι η χρήση διαγραμμάτων επιρροής, τα οποία παρέχουν γραφική αναπαράσταση του μοντέλου, αποτελούν μοντέλο για το ίδιο το μοντέλο και δίνουν μια εποπτική εικόνα της υπάρχουσας κατάστασης.



Εικόνα 14: Δομικός ορισμός ΣΥΑ

Η γενική διαδικασία ανάπτυξης ενός ΣΥΑ, φαίνεται στο επόμενο σχήμα. Υπάρχει η φάση πριν την σχεδίαση, η φάση της σχεδίασης και τέλος οι υπόλοιπες φάσεις της ανάπτυξης. Η κάθε επιμέρους φάση συνδέεται με τις προηγούμενες και τις επόμενες της, έτσι ώστε ανά πάσα στιγμή να υπάρχει δυνατότητα να επιστρέψουμε σε προηγούμενο στάδιο και να γίνουν οι απαραίτητες διορθώσεις.



Εικόνα 15: Η γενική διαδικασία ανάπτυξης ΣΥΑ

3.2 Διαχείριση Στόλου Οχημάτων

Τα συστήματα Διαχείρισης Στόλου δίνουν απάντηση σε ένα από τα βασικά ερωτήματα που απασχολούν τις επιχειρήσεις που διαθέτουν οχήματα: πού βρίσκονται τα οχήματα της εταιρείας. Η απάντηση ίσως γενικά να μην είναι δύσκολη. Όταν όμως συζητάμε για την θέση των οχημάτων σε πραγματικό χρόνο, με σκοπό την αξιοποίηση των στοιχείων για να αυξηθεί η παραγωγικότητα της επιχείρησης και των εργαζομένων με ταυτόχρονη μείωση των λειτουργικών δαπανών, τότε συνειδητοποιούμε τα πρώτα οφέλη και την ανάγκη για ένα σύστημα διαχείρισης στόλου.

Γενικά, Διαχείριση Στόλου καλείται το σύνολο των τεχνολογιών και συστημάτων που επιτρέπουν σε μία επιχείρηση να έχει πλήρη έλεγχο των μεταφορικών της (επιβατικά αυτοκίνητα, φορτηγά, λεωφορεία, δίκυκλα, πλοία και αεροπλάνα). Ένα τέτοιο σύστημα λειτουργεί με τέτοιο τρόπο ώστε να βελτιστοποιούνται αρκετές επιμέρους επιχειρηματικές διαδικασίες, να υπάρχει διαχειριστικός έλεγχος και, κατά συνέπεια, να μειώνεται το κόστος και να κατανέμονται καλύτερα οι πόροι της επιχείρησης.

Με ένα σύστημα διαχείρισης στόλου, υπάρχει δυνατότητα άμεσης πληροφόρησης σχετικά με αρκετά από τα θέματα που αφορούν τα οχήματα, ανάλογα με τις ανάγκες. Εκτός του βασικού και πλέον διαδεδομένου στοιχείου: της ακριβούς γεωγραφικής θέσης κάθε οχήματος, υπάρχει πληροφόρηση για πλήθος άλλων στοιχείων:

- i. Ταχύτητα,
- ii. Διαδρομή που έχει ακολουθηθεί,
- iii. Θερμοκρασία,
- iv. Κατανάλωση καυσίμου
- v. Πλήθος άλλων στοιχείων, ανάλογα με τις απαιτήσεις.

Πρακτικά είναι ένα σύστημα παρακολούθησης της θέσης κάθε οχήματος της εταιρείας, μαζί με τα λοιπά στοιχεία που απαιτούνται, το οποίο επιπλέον περιλαμβάνει ένα υποσύστημα που επιτρέπει την επικοινωνία μεταξύ του οχήματος και ενός κεντρικού σταθμού.

Συνήθως στα οχήματα εγκαθίσταται ένα GPS, που συλλέγει τα στίγματα που μαζί με τις άλλες απαιτούμενες πληροφορίες στέλνονται στο κέντρο ελέγχου. Εκτός του GPS και ανάλογα με το τις πληροφορίες που πρέπει να συλλεχθούν, τοποθετούνται συμπληρωματικές συσκευές που καταγράφουν τα στοιχεία.

Η πιο απλή μορφή ενός τέτοιου συστήματος, με περιορισμένες όμως δυνατότητες είναι με την χρήση SMS. Η πλέον συνηθισμένη περίπτωση όμως είναι με την χρήση GPS (για τον εντοπισμό της θέσης), PDA (για την αναπαράσταση της θέσης του οχήματος) και ένα σύστημα ασύρματης επικοινωνίας, μέσω κατάλληλου δικτύου (GPRS,3G,HSDPA,TETRA), ώστε να υπάρχει μία αμφίδρομη επαφή μεταξύ του οχήματος και του κέντρου ελέγχου της

επιχείρησης. Μια τέτοια υλοποίηση επιτρέπει μία σειρά από σημαντικές διευκολύνσεις, οι κυριότερες από τις οποίες είναι η αποστολή οδηγιών προς τον οδηγό του οχήματος και η επαναδρομολόγηση του οχήματος, προκειμένου να καλυφθεί μία επείγουσα ανάγκη που δημιουργήθηκε απρόσμενα ή να γίνει καλύτερη εκμετάλλευση του διαθέσιμου χρόνου του οχήματος. Όλα αυτά μπορεί να υπάρχουν ακόμα και σε μια μόνο συσκευή, διαφορετικά θα πρέπει να επικοινωνούν μεταξύ τους.

Το GPS ως τεχνολογία για τον εντοπισμό της θέσης, φαίνεται να καλύπτει τις ανάγκες. Υπάρχουν βέβαια και εναλλακτικές σε περίπτωση που δεν υπάρχει κάλυψη από δορυφόρο (πυκνοκατοικημένες περιοχές). Πιο διαδεδομένη είναι η χρήση των δικτύων κινητής τηλεφωνίας, που προσφέρουν ακρίβεια στον εντοπισμό της θέσης μέχρι 30 μέτρα. Το πρόβλημα που υπάρχει εδώ είναι τα προσωπικά δεδομένα του χρήστη, αλλά αυτό δεν αφορά την παρούσα μελέτη.

Ως προς την επικοινωνία του τερματικού με τον κεντρικό σταθμό γίνεται χρήση δυο τεχνολογιών. Η πρώτη είναι τα δίκτυα κινητής τηλεφωνίας, και πιο συγκεκριμένα αυτά που χρησιμοποιούν την τεχνολογία GPRS, που επιτρέπουν τη μετάδοση δεδομένων σε ταχύτητα της τάξεως των 56 Kbps. Η ταχύτητα δεν είναι μεγάλη, αλλά επειδή ο όγκος των δεδομένων που αποστέλλονται είναι μικρός, είναι ικανοποιητική. Εναλλακτικά με 3G ή HSDPA, επιτυγχάνονται μεγαλύτερες ταχύτητες. Βασικό πλεονέκτημα της τεχνολογίας αυτής, είναι η πανελλαδική κάλυψη που παρέχουν τα δίκτυα κινητής τηλεφωνίας.

Δεύτερη τεχνολογία το TETRA, που όμως δεν παρέχει πανελλαδική κάλυψη, αλλά έχει 3 βασικά πλεονεκτήματα: μεγαλύτερη ασφάλεια, καλύτερη ποιότητα φωνής και η δυνατότητα για πραγματοποίηση ομαδικών κλήσεων.

Σκοπός της διαχείρισης στόλου είναι να βελτιστοποιηθεί ο τρόπος λειτουργίας της επιχείρησης. Αυτό επιτυγχάνεται με την παροχή αποδοτικότερων τρόπων υπολογισμού κρίσιμων μεγεθών για την ίδια την επιχείρηση, ενώ ταυτόχρονα αξιοποιεί σύγχρονα αποτελεσματικά και ακριβή εργαλεία.

Οι λύσεις διαχείρισης στόλου, βρίσκουν εφαρμογή σε πλήθος επιχειρήσεων, μικρού, μεσαίου και μεγάλου μεγέθους. Σε μεταφορικές και τουριστικές επιχειρήσεις, μέσα μεταφοράς (λεωφορεία, ταξί), ασθενοφόρα, σώματα ασφαλείας, ένοπλες δυνάμεις ή ακόμα και ιδιωτικές εταιρείες ασφαλείας. Σε εταιρείες ενοικίασης αυτοκινήτων και σε εταιρείες ταχυμεταφορών.

Το παράδοξο που εμφανίζεται αρκετές φορές, είναι μικρές ή μεσαίες επιχειρήσεις να καταφεύγουν σε λύσεις διαχείρισης στόλου, ενώ άλλες με μεγάλο αριθμό οχημάτων όχι. Αυτό είναι καθαρά θέμα πολιτικής της κάθε εταιρείας και των διαθέσιμων πόρων, σε συνδυασμό με το πόσο κρίσιμη είναι η ποιότητα αλλά και η ποσότητα των πληροφοριών που συλλέγουν με την λύση αυτή.

Τα πλεονεκτήματα που πηγάζουν από την χρήση μιας εφαρμογής διαχείρισης στόλου είναι αρκετά. Το πρώτο και σημαντικότερο ίσως, είναι ότι πλέον με γρήγορο και εύκολο τρόπο, μπορούν να συλλεχθούν πληροφορίες που μέχρι τώρα ήταν δύσκολο να συλλεχθούν. Έτσι μπορούν να παραχθούν κατάλληλες αναφορές για τον τρόπο που κινούνται τα οχήματα, τα δρομολόγια τους, τη συχνότητα κινήσεων κάθε ενός κλπ. και έτσι να βρεθούν τρόποι βελτίωσης της υπάρχουσας υποδομής. Αυτές με τη σειρά τους οδηγούν στη σωστή λήψη αποφάσεων, ως προς τον τρόπο διάθεσης των πόρων, τις ανάγκες σε προσωπικό και μέσα, επιλογή εναλλακτικών διαδρομών.

Ειδικά το τελευταίο είναι μια από τις νέες τάσεις στα συστήματα διαχείρισης στόλου. Με δεδομένο ότι όλο και περισσότερες εταιρείες πραγματοποιούν πωλήσεις και παραδόσεις σε σημεία, τα οποία δεν είναι από πριν καθορισμένα, μία λύση που θα επιτρέπει την προσαρμογή του δρομολογίου, ανάλογα με τις συνθήκες που διαμορφώνονται κάθε στιγμή, κρίνεται αναγκαία. Μια τέτοια λειτουργία,, γνωστή ως επαναδρομολόγηση, χαρακτηρίζεται από τα παρακάτω:

- i. Δυναμική αναπροσαρμογή του προγράμματος διανομών για την αντιμετώπιση σε πραγματικό χρόνο εκτάκτων απαιτήσεων είτε από τους πελάτες είτε λόγω εξωτερικών παρεμβάσεων όπως τα κυκλοφοριακά προβλήματα, έργα σε δρόμους, πορείες κλπ.
- ii. Αμφίδρομη επικοινωνία δεδομένων μέσω packet-based κινητών δικτύων (GPRS).
- iii. Βελτιωμένη οργάνωση στόλου μέσω της συνεχούς παρακολούθησης συγκεκριμένων λειτουργιών και αποθήκευσή τους σε βάση δεδομένων καθώς επίσης και μέσω της επαναδρομολόγησης του στόλου σε πραγματικό χρόνο.
- iv. Συλλογή μετρήσεων απόδοσης σε πραγματικό χρόνο (ώρες εργασίας, υπολογισμός ποσοστού % των παραδόσεων που γίνονται εγκαίρως κλπ), οι οποίες μπορούν να οδηγήσουν στην αποδοτικότερη διαχείριση του ανθρώπινου δυναμικού.
- v. Ανακοίνωση της κατάστασης της παραγγελίας στους πελάτες σε πραγματικό χρόνο.
- vi. Βέλτιστη αξιοποίηση του στόλου με τη διαχείριση οχημάτων που διεκπεραιώνουν διανομές και παραλαβές προϊόντων ταυτόχρονα.
- vii. Επιτυχής επίτευξη αυστηρών ή ελαστικών χρονικών παραθύρων παράδοσης.
- viii. Αυτοματοποίηση της αλληλεπίδρασης μεταξύ του λογισμικού, που είναι εγκατεστημένο στα συστήματα του στόλου, και του συστήματος ERP της επιχείρησης.
- ix. Ελαχιστοποίηση της αναστάτωσης της αρχικής ισορροπίας του συστήματος μέσω της χρήσης του μικρότερου κατά το δυνατόν αριθμού οχημάτων για την ικανοποίηση των δυναμικών γεγονότων.

Σε μια τέτοια περίπτωση, αν και οι απαιτήσεις προκειμένου να υλοποιηθεί είναι αρκετές, τα οφέλη είναι μεγαλύτερα:

- i. Μείωση των συνολικών δαπανών του στόλου οχημάτων με τη μείωση του αριθμού φορτηγών που χρησιμοποιούνται.
- ii. Υποστήριξη προηγμένων τεχνολογιών διαχείρισης σχέσεων με τους πελάτες (CRM), μέσω της χρήσης περισσότερων φιλικών προς τον χρήστη διεπαφών (interfaces) για τον εντοπισμό της παράδοσης μιας παραγγελίας.
- iii. Βελτιωμένη λήψη αποφάσεων για την επίλυση των προβλημάτων που ανακύπτουν.
- iv. Υπηρεσίες προστιθεμένης αξίας. Λαμβάνοντας υπόψη όλα τα παραπάνω μπορεί να ειπωθεί ότι ένα τέτοιο σύστημα προσθέτει αξία στην επιχείρηση με τη βελτίωση των δραστηριοτήτων αλυσίδων ανεφοδιασμού και, επομένως, μπορεί να οδηγήσει σε καλύτερη εξυπηρέτηση πελατών, χαμηλότερο κόστος και, βεβαίως, σε βελτιωμένη χρήση των πάγιων ενεργητικών και του στόλου.
- v. Βελτιωμένος χειρισμός απρόβλεπτων γεγονότων. Χρησιμοποιώντας έξυπνους αλγορίθμους επαναδρομολόγησης και αντίστοιχου λογισμικού, κάθε εταιρεία μεταφορών είναι σε θέση να αποκριθεί με τον βέλτιστο τρόπο σε καταστάσεις κρίσης.
- vi. Περισσότερος χρόνος λειτουργίας. Μέσω της χρήσης των τηλεματικών αισθητήρων, ο ωφέλιμος χρόνος λειτουργίας των φορτηγών μεγιστοποιείται, καθώς ελαχιστοποιείται ο χαμένος χρόνος από τις απρόβλεπτες διακοπές.
- vii. Έλεγχος απόδοσης. Η απόδοση οδηγών μπορεί να ελεγχθεί σε πραγματικό χρόνο μέσω των μετρήσεων που συλλέγονται συνεχώς.
- viii. Εντοπισμός σε πραγματικό χρόνο. Μέσω μιας σελίδας στο διαδίκτυο η επιχείρηση, καθώς επίσης και οι πελάτες, μπορούν να λάβουν τις πληροφορίες σε πραγματικό χρόνο για κάθε παράδοση.
- ix. Βελτιωμένη διαχείριση.

Κλείνοντας με τα πλεονεκτήματα μιας λύσης διαχείρισης στόλου, αξίζει να αναφερθεί, ότι μακροπρόθεσμα και με συλλογή πληροφοριών σε βάθος χρόνου, καταγράφονται πολλά στοιχεία που χρησιμοποιούνται, ομαδοποιημένα κατάλληλα, σε ανάλυση των υπάρχουσών καταστάσεων και εύρεση καλύτερων μεθόδων για μελλοντικές περιπτώσεις. Υπάρχει επίσης και οικονομία ως προς το κόστος των επικοινωνιών, μιας και πλέον δεν απαιτείται συνεχής επικοινωνία των κινουμένων με το κέντρο, μιας και μεγάλο μέρος των πληροφοριών, στέλνονται συνεχώς και με μικρότερο κόστος.

Συνοψίζοντας με τα πλεονεκτήματα που προκύπτουν από την χρήση του:

- i. Αυτόματος εντοπισμός των οχημάτων και καλύτερος έλεγχός τους και του τρόπου με τον οποίο κινούνται σε πραγματικό χρόνο.
- ii. Οπτικοποίηση κάθε πληροφορίας σε χάρτες.
- iii. Αποδοτικότερη δρομολόγηση των οχημάτων με παράλληλη μείωση του χρόνου απασχόλησης των εργαζομένων σε αυτά.

- iv. Βελτίωση των χρόνων παράδοσης και διεκπεραίωσης εργασιών.
- v. Βελτίωση της ποιότητας των παρεχόμενων υπηρεσιών.
- vi. Αύξηση πελατολογίου μέσω της παροχής βελτιωμένων υπηρεσιών.
- vii. Εξασφάλιση των απαιτούμενων συνθηκών διακίνησης και παράδοσης (θερμοκρασία κλπ), παράμετρος ιδιαίτερα σημαντική για ευπαθή προϊόντα που διακινούνται με οχήματα-ψυγεία.
- viii. Εντοπισμός των πελατών ή γενικότερα των σημείων που προκαλούν καθυστέρηση στα δρομολόγια των οχημάτων.
- ix. Μείωση της ανάγκης επικοινωνίας με τους οδηγούς μέσω παραδοσιακών μεθόδων.
- x. Αύξηση της απόδοσης του στόλου των οχημάτων.
- xi. Δυνατότητα αποτροπής κλοπής ή κίνησης εκτός προκαθορισμένων ορίων.
- xii. Παροχή αισθήματος ασφάλειας στους οδηγούς έναντι απρόβλεπτων γεγονότων.
- xiii. Αποτελεσματική αντιμετώπιση εκτάκτων καταστάσεων.
- xiv. Δυνατότητα σύνδεσης με τηλεφωνικό κέντρο και σύστημα βέλτιστης δρομολόγησης.
- xv. Γεωγραφική απεικόνιση και κατηγοριοποίηση όλου του πελατολογίου.
- xvi. Εξαγωγή στατιστικών δεδομένων σχετικά με τις κινήσεις των οχημάτων.
- xvii. Μείωση του κόστους λειτουργίας.
- xviii. Δυνατότητα λήψης σε πραγματικό χρόνο δεδομένων που μπορούν να βοηθήσουν στη λήψη σωστών αποφάσεων.
- xix. Βελτιστοποίηση οικονομικών μεγεθών μέσω, μεταξύ άλλων, και της μείωσης του διαχειριστικού κόστους.
- xx. Καλύτερη διαχείριση ενός από τα σημαντικότερα περιουσιακά στοιχεία μίας επιχείρησης, των οχημάτων της.

Τα προβλήματα που υπάρχουν αυτή τη στιγμή στο χώρο, είναι από τη μια το κόστος εγκατάστασης και χρήσης, που αρχικά μοιάζει μεγάλο, και από την άλλη η αντιμετώπιση που έχει ένα τέτοιο σύστημα από τους οδηγούς των οχημάτων, που πολλές φορές νοιώθουν ότι παρακολουθούνται. Απαιτείται υπομονή και εκπαίδευση, ώστε να συνειδητοποιηθεί από όλους τους εμπλεκόμενους, ότι τα οφέλη είναι σημαντικά και διόλου ευκαταφρόνητα.

3.3 Γεωγραφικά Συστήματα Πληροφοριών

Με μια πρώτη προσέγγιση τα ΓΣΠ ορίζονται συνοπτικά ως ένα σύνολο εργαλείων συλλογής, αποθήκευσης, ανάκτησης, ανάλυσης και εμφάνισης χωρικών δεδομένων. Σύμφωνα με τον F.Hanigan, ΓΣΠ είναι ένα πληροφοριακό σύστημα, που μπορεί να:

- i. Συλλέγει , αποθηκεύει, και ανακτά πληροφορίες βασιζόμενο στην χωρική τους διάσταση.

ii. Αναγνωρίζει τοποθεσίες εντός ενός συγκεκριμένου περιβάλλοντος οι οποίες απαντούν σε συγκεκριμένα κριτήρια.

iii. Ανακαλύπτει σχέσεις ανάμεσα σε σετ δεδομένων του περιβάλλοντος.

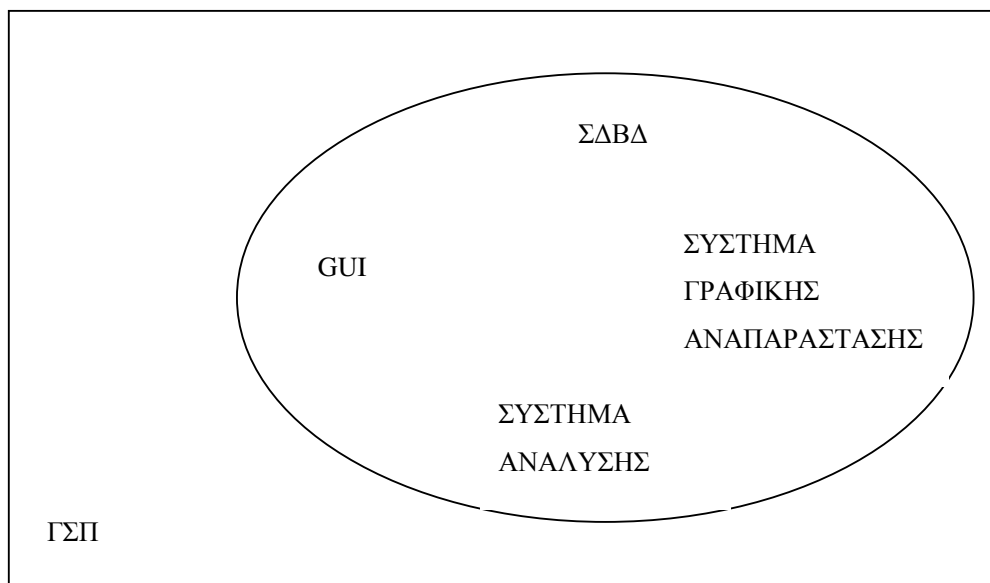
iv. Αναλύει χωρικά τα δεδομένα βοηθώντας στην λήψη αποφάσεων σχετικά με το υπό εξέταση περιβάλλον.

v. Διευκολύνει την επιλογή και την παρουσίαση δεδομένων σε εφαρμογές ικανές να εκτιμούν με την χρήση αναλυτικών μοντέλων την επίδραση της μεταβολής των δεδομένων στο περιβάλλον.

vi. Παρουσιάζει το επιλεγμένο περιβάλλον τόσο γραφικά όσο και αριθμητικά πριν ή μετά από ανάλυση.

Από τον ορισμό αυτό προκύπτει ότι τα γεωγραφικά συστήματα πληροφοριών δεν είναι απλώς ένας ακόμα εναλλακτικός τρόπος προβολής δεδομένων με γραφικά. Είναι μια πλήρης συλλογή εργαλείων για την συλλογή , αποθήκευση , ανάκτηση και απεικόνιση χωρικών πληροφοριών.

Το λογισμικό ΓΣΠ έχει συγκεκριμένη αρχιτεκτονική, που στην τυπική περίπτωση αποτελείται από το γραφικό περιβάλλον (GUI), τη βάση δεδομένων , το σύστημα ανάλυσης και απεικόνισης με γραφικά των δεδομένων (Εικόνα 16)



Εικόνα 16: Περιβάλλον ΓΣΠ

Από τα παραπάνω, προκύπτει και το βασικό πλεονέκτημα των ΓΣΠ: η αποθήκευση των δεδομένων γίνεται χωριστά από την αναπαράστασή τους, κάτι που έχει ως αποτέλεσμα τα ίδια δεδομένα να μπορούν να αναπαρασταθούν με διαφορετικούς τρόπους, ενώ παράλληλα η χρήση της υπολογιστικής ισχύος των σύγχρονων υπολογιστών, επιτρέπει την διαχείριση μεγάλων ποσοτήτων δεδομένων εύκολα και γρήγορα. Τα ΓΣΠ δέχονται δεδομένα από

πίνακες, χάρτες, αισθητήρες και παράγουν προϊόντα όπως αναφορές, χάρτες, στατιστικά δεδομένα. Για να επιτευχθεί αυτό, διαλειτουργούν με ένα σύστημα διαχείρισης βάσεων δεδομένων (ΣΔΒΔ), με προγράμματα στατιστικής και γενικά προγράμματα ανάλυσης, με προγράμματα CAD και προγράμματα επεξεργασίας εικόνας.

Τα GIS μπορούν να χρησιμοποιηθούν τόσο με ανυσματικά (vector) (σημεία, γραμμές, πολύγωνα) όσο και με ψηφιοποιημένα (raster) δεδομένα., ενώ οι λειτουργίες της βάσης δεδομένων τους μπορούν να εφαρμοστούν σε χωρικά και μη δεδομένα.

Στις βασικές λειτουργίες ενός GIS περιλαμβάνονται τα παρακάτω:

- i. Επιλογή δεδομένων και επερωτήσεις
- ii. Δημιουργία Buffer zones
- iii. Γεωμετρικές μετατροπές
- iv. Γεωαναφορά
- v. Λειτουργίες επικάλυψης
- vi. Τοπογραφικές λειτουργίες
- vii. Χωρικοί υπολογισμοί.

Μια από τις πλέον βασικές ιδιότητες των ΓΣΠ, είναι η δυνατότητα που έχουν να αλληλεπιδρούν με υπολογιστικά συστήματα, όπως είναι τα συστήματα μοντελοποίησης, προσομοίωσης και λήψης αποφάσεων.

Η σχέση τους με τα ΣΥΑ έγκειται στο πως ένα ΓΣΠ, μπορεί να εμπλακεί με τις 3 βασικές φάσεις της λήψης μιας απόφασης. Έτσι, κατά φάση, έχουμε:

i. Στη φάση της πληροφορίας, ένα ΓΣΠ μπορεί να παίζει καθοριστικό ρόλο, διότι έχει την ικανότητα να διαχειρίζεται διαφορετικά είδη δεδομένων με αποτέλεσμα τη συμβολή του στην κατανόηση του προβλήματος για το οποίο πρέπει να ληφθεί απόφαση, και επίσης διότι με τη γραφική αναπαράσταση δίνει στον αποφασίζων, τη δυνατότητα να εντοπίσει διαστάσεις του προβλήματος, που δεν θα μπορούσε διαφορετικά.

ii. Στη δεύτερη φάση, τη φάση της σχεδίασης, συνεισφέρει με την παροχή εναλλακτικών λύσεων που βασίζονται στις χωρικές αρχές της συνδετικότητας, συνέχειας-γεινίας, εγγύτητας και των μεθόδων επικάλυψης.

iii. Στη φάση της επιλογής, τα ΓΣΠ συνεισφέρουν με την ικανότητα που έχουν να συγχωνεύουν την απόφαση του λήπτη αποφάσεων με την όλη διαδικασία λήψης αποφάσεων.

Οι κατηγορίες προβλημάτων στα οποία καλούμαστε να δώσουμε λύση και να πάρουμε απόφαση, με τη συνεισφορά των ΓΣΠ, χωρίζονται σε 3 κατηγορίες:

i. Παραδοσιακά πεδία δράσης των ΓΣΠ (δασολογία, χωροταξία κλπ). Με την αυτοματοποίηση της παραγωγής αυξάνει η παραγωγικότητα,

ii. Δρομολόγηση και ανάλυση περιοχής. Ο συνδυασμός χρήσης επιχειρηματικών μοντέλων ΓΣΠ δίνει την δυνατότητα μοντελοποίησης της δρομολόγησης οχημάτων, που διαφορετικά ήταν πολύ δύσκολο.

iii. Marketing όπου αυξημένη διαθεσιμότητα χωρικών δεδομένων, μπορεί να οδηγήσει σε μεγαλύτερη ανάλυση της αγοράς.

Ένα καλά οργανωμένο δίκτυο διανομών, με τη βοήθεια των ΓΣΠ, παρέχει τη δυνατότητα σε κάθε επιχείρηση που διαθέτει στόλο οχημάτων να ελαχιστοποιήσει τόσο το συνολικό χρόνο μεταφοράς των προϊόντων της όσο και το αντίστοιχο κόστος. Οι διάφορες τεχνικές υπολογισμού ελάχιστης διαδρομής, μέσω των ΓΣΠ, παρέχουν αυτή τη δυνατότητα άμεσα και εύκολα.

3.4 Χωρικά Συστήματα Υποστήριξης Αποφάσεων

Βασίζόμενοι στον ορισμό των ΣΥΑ και τροποποιώντας τον κατάλληλα, ώστε να μπορεί να συμπεριλάβει και τα Χωρικά Συστήματα Υποστήριξης Αποφάσεων, μπορούμε να πούμε ότι ονομάζονται τα υπολογιστικά συστήματα τα οποία είναι σχεδιασμένα ώστε να υποστηρίζουν έναν χρήστη ή μια ομάδα χρηστών στην δημιουργία λύσεων σε ημιδομημένα ή κακώς δομημένα χωρικά προβλήματα με την χρήση χωρικών δεδομένων, επιχειρηματικών μοντέλων, εργαλείων λογισμικού και έμπειρων συστημάτων με βασικό σκοπό την αύξηση της αποτελεσματικότητας των αποφασίζοντων, πράγμα που επιτυγχάνεται με την ενσωμάτωση και αλληλεπίδραση των αποφάσεων του χρήστη με τις εφαρμογές και τα μοντέλα του συστήματος καθώς και με την ευκολία χρήσης του. Συνεπώς, τα ΧΣΥΑ είναι ΣΥΑ, που εστιάζουν σε προβλήματα αποφάσεων τα οποία σχετίζονται με το γεωγραφικό περιβάλλον.

Η προσθήκη του όρου «Χωρικά» στο «Σύστημα Υποστήριξης Αποφάσεων», μπορεί να οδηγήσει στο εσφαλμένο συμπέρασμα ότι τα ΧΣΥΑ είναι μιας μορφής ΓΣΠ. Κάτι τέτοιο δεν ισχύει αφού μεταξύ των δύο υπάρχουν βασικές διαφορές. Τα ΓΣΠ εστιάζουν στην πληροφορία, ενώ τα ΧΣΥΑ στην λήψη απόφασης. Εντελώς διαφορετική είναι και η συμμετοχή του χρήστη στα συστήματα αυτά, ενώ και τα παραγόμενα αποτελέσματα είναι διαφορετικά. Οι διαφορές μεταξύ των ΓΣΠ και των ΧΣΥΑ, φαίνονται συνοπτικά στον Πίνακα 1.

Αντικείμενο	ΓΣΠ	ΧΣΥΑ
Βασικό Αντικείμενο	Πληροφορία	Λήψη απόφασης
Προβλήματα	Πλήρως δομημένα	Ημιδομημένα ή κακώς δομημένα
Λειτουργία	Πραγματικός κόσμος και προβλήματα σε αυτών	Υποστήριξη της διαδικασίας λήψης αποφάσεων σε προβλήματα με χωρικά δεδομένα
Αποτελέσματα	Χάρτες	Αποφάσεις – Λύσεις
Χρήστης	Παθητικός, καθοδηγείται από το σύστημα	Ενεργητικός, καθοδηγεί το σύστημα
Αντικειμενικός σκοπός	Αύξηση της ταχύτητας επεξεργασίας των δεδομένων	Βελτίωση της αποτελεσματικότητας λήψης αποφάσεων.

Πίνακας 1: Διαφορές ΓΣΠ – ΧΣΥΑ

Από την άλλη πλευρά, τα ΧΣΥΑ έχουν πολλά κοινά στοιχεία με τα ΣΥΑ (επίλυση ημιδομημένων προβλημάτων, εύκολο περιβάλλον χρήστη, εύρεση εναλλακτικών λύσεων, αλληλεπιδραστική και αναδρομική διαδικασία λύσης προβλημάτων). Έχουν όμως και επιπλέον χαρακτηριστικά:

- i. Παροχή μηχανισμών για την εισαγωγή χωρικών δεδομένων.
- ii. Απεικόνιση σχέσεων και δομών χωρικών δεδομένων.
- iii. Παροχή μηχανισμών χωρικής – γεωγραφικής ανάλυσης.
- iv. Εξαγωγή αποτελεσμάτων σε διάφορες χωρικές μορφές.

Τα ΧΣΥΑ επομένως, παίζουν σημαντικό ρόλο στη διαδικασία λήψης αποφάσεων. Κάθε πρόβλημα το οποίο απαιτεί λύση, σημείο εκκίνησης είναι ο «πραγματικός κόσμος». Στον «πραγματικό κόσμο» βρίσκεται το πρόβλημα, και σε αυτόν επιδρούν (θετικά ή αρνητικά) οι διάφοροι παράγοντες, επηρεάζοντας ανάλογα και τη διαδικασία εύρεσης λύσης και τη λήψη αποφάσεων .

Στα ΧΣΥΑ υπάρχει μεγάλος αριθμός χωρικών δεδομένων, τα οποία περιγράφουν τα αντικείμενα του πραγματικού κόσμου. Υπάρχουν μαθηματικά μοντέλα, που περιγράφουν τους νόμους και τους κανόνες που τον διέπουν. Υπάρχουν όμως και ευέλικτοι μηχανισμοί, τόσο όσο προς την αρχιτεκτονική των αντικειμένων, όσο και ως προς την εκτελεστική

δυνατότητα. Τα παραπάνω, συνεργάζονται αρμονικά και δένουν, ώστε να βοηθήσουν τον αποφασίζοντα, στη διαδικασία λήψης απόφασης, και εύρεσης λύσης.

Η κατηγοριοποίηση των ΧΣΥΑ, γίνεται ανάλογα με το είδος των προβλημάτων, που καλούνται να λύσουν. Έτσι προκύπτουν οι παρακάτω κατηγορίες:

i. Συστήματα Επεξεργασίας Χωρικών Δεδομένων: Περιέχουν πολύ καλά ορισμένα τα δεδομένα, τις στρατηγικές, τα κριτήρια εκτίμησης και τους περιορισμούς. Το ίδιο το σύστημα παράγει την απόφαση. Προορίζονται για συγκεκριμένα και σαφώς ορισμένα προβλήματα.

ii. Χωρικά Συστήματα Λήψης Αποφάσεων: Εδώ τα προβλήματα είναι ημιδομημένα, και αποτελούν την πλέον βασική κατηγορία. Σκοπό έχουν την αύξηση της αποτελεσματικότητας της διαδικασίας λήψης της απόφασης. Επιτρέπουν και καθοδηγούν τους χρήστες στο να λαμβάνουν καλύτερες αποφάσεις, αφού με δυναμική και αναδρομική διαδικασία γίνεται αρχικά ανίχνευση και στη συνέχεια μοντελοποίηση του προβλήματος.

iii. Χωρικά Έμπειρα Συστήματα: Γίνεται κωδικοποίηση της γνώσης των ειδικών σε κάποιο θέμα και στη συνέχεια χρησιμοποιείται στη λήψη απόφασης, ακόμα και από απλούς χρήστες. Τα προβλήματα επιλύονται όπως αν τα επίλυε κάποιος ειδικός, ή ακόμα καλύτερα. Με βάση τη γνώση δημιουργούνται κανόνες και τελικά υπάρχει αλληλεπίδραση συστήματος και αποφασίζοντα.

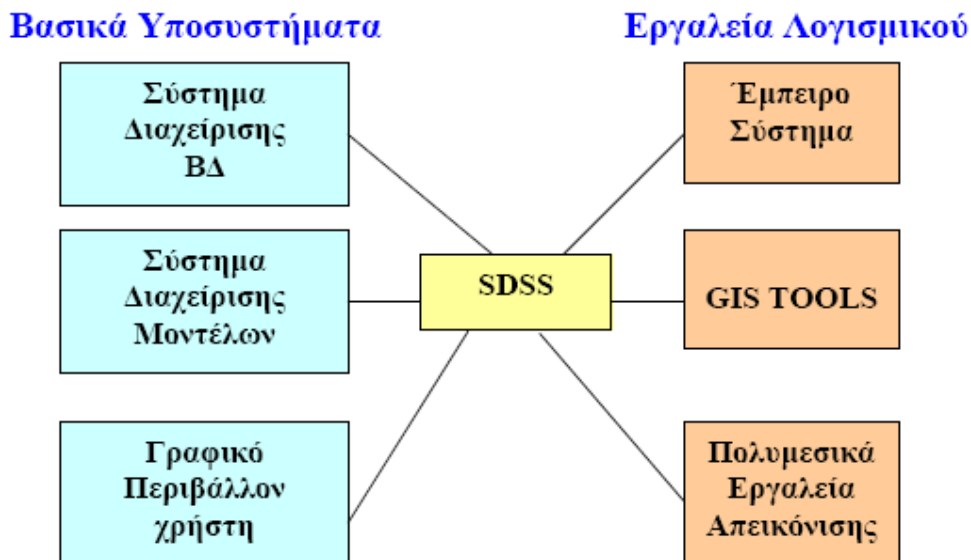
Μια δεύτερη ταξινόμηση των ΧΣΥΑ είναι ανάλογα με την αρχιτεκτονική του συστήματος στο οποίο είναι εγκατεστημένα. Χωρίζονται σε δικτυακά και stand-alone. Σε αντίθεση με τα stand-alone, τα δικτυακά έχουν τα στοιχεία τους εγκατεστημένα σε περισσότερους τους ενός υπολογιστές. Πολλοί χρήστες συμμετέχουν στη διαδικασία λήψης απόφασης, ακόμα και αν αυτοί βρίσκονται σε διαφορετικό ,γεωγραφικά, χώρο.

Σε κάθε περίπτωση όμως, ένα ΧΣΥΑ αποτελείται από 3 βασικά υποσυστήματα καθώς και εργαλεία λογισμικού (Εικόνα 17), τα οποία θα δούμε αναλυτικά:

i. Σύστημα Διαχείρισης Βάσης Δεδομένων: Περιλαμβάνει όλες τις ΒΔ (χωρικές, σχεσιακές) που χρησιμοποιεί το σύστημα. Επιτρέπει στους χρήστες να ανακτούν , ενημερώνουν και να συντηρούν τις ΒΔ.

ii. Σύστημα Διαχείρισης Μοντέλων: Περιέχει την ευφυΐα του συστήματος στις βάσεις που διαθέτει. Πιο συγκεκριμένα στη Βάση Μοντέλων έχει τα επιχειρηματικά μοντέλα που χρησιμοποιούνται από τον χρήστη. Στη Βάση Μεθόδων υπάρχουν οι αλγόριθμοι δημιουργίας των επιχειρηματικών μοντέλων. Στη Βάση Εργαλείων υπάρχουν τα εργαλεία λογισμικού, που μετατρέπουν τα δεδομένα ώστε να έχουν την κατάλληλη μορφή για να αποθηκευτούν στις ΒΔ. Τέλος, υπάρχει και η Βάση Γνώσης, που έχει όλη την γνώση, που παράγεται από ένα έμπειρο σύστημα και με βάση την οποία αξιολογείται όλο το σύστημα και τα παραγόμενα από αυτό.

iii. Γραφικό Περιβάλλον Χρήστη: Είναι το μέσο με το οποίο ο χρήστης χρησιμοποιεί τους πόρους και παράγει λύσεις. Βασικές του λειτουργίες είναι η λήψη απόφασης, η ανάκτηση πληροφοριών, η εμφάνιση και εξαγωγή πληροφοριών και η γενικότερη διαχείριση του συστήματος.



Εικόνα 17: Δομικά Στοιχεία ΧΣΥΑ

- iv. Εργαλεία GIS: Για την χρήση γεωγραφικών πληροφοριών
- v. Έμπειρά Συστήματα: για την παραγωγή γνώσης σχετικά με το πρόβλημα.
- vi. Πολυμεσικά εργαλεία απεικόνισης: Για να παρουσιάζουν γραφικά τη διαδικασία λήψης καθώς συμβαίνει.

Κλείνοντας με τα ΧΣΥΑ, τονίζεται ότι αποτελούν μια από τις πιο σημαντικές κατηγορίες των ΣΥΑ, λόγω της υπάρχουσας σχέσης μεταξύ χωρικής πληροφορίας και τν πληροφοριών που απαιτούνται για την λήψη απόφασης. Τα ΧΣΥΑ προσθέτουν τη χωρική διάσταση σε παραδοσιακούς τομείς των ΣΥΑ, όπως η δρομολόγηση και επιτρέπουν στους αποφασίζοντες να συνδυάσουν χωρικά δεδομένα με δεδομένα άλλων μορφών.

3.5 Αντικειμενοστραφής Μεθοδολογία και UML

Η Αντικειμενοστραφής μεθοδολογία είναι μια προσέγγιση για την ανάπτυξη συστημάτων και εφαρμογών, που ενθαρρύνει την επαναχρησιμοποίηση των συστατικών του λογισμικού. Με την μεθοδολογία αυτή, ένα υπολογιστικό σύστημα μπορεί να αναπτυχθεί με τέτοιο τρόπο ώστε να είναι δυνατή η επαναχρησιμοποίηση των υπάρχόντων συστατικών και διευκολύνει το διαμοιρασμό τους σε άλλα συστήματα. Επιτυγχάνεται με τον τρόπο αυτό υψηλή παραγωγικότητα, λιγότερη συντήρηση και βελτίωση της ποιότητας.

Βασικός σκοπός της είναι η παραγωγή λογισμικού με τη μέθοδο της «συναρμολόγησης». Γίνεται χρήση υπαρχόντων συστατικών, τα οποία συνδυάζονται με διάφορους τρόπους, προκειμένου να ικανοποιήσουν τις ανάγκες των χρηστών, και μόνο ό,τι αποτελεί καινούρια λειτουργικότητα δημιουργείται.

Εφαρμόζεται σε απλά μοντέλα αντικειμένων που αναπτύσσονται από την ανάλυση και τη φάση της σχεδίασης μέχρι το επίπεδο του προγραμματισμού. Τα αντικείμενα περιέχουν τόσο τα δεδομένα όσο και τις λειτουργίες που επιδρούν σε αυτά. Ένα αντικείμενο μπορεί να προσπελαστεί μόνο με κάποια από τις λειτουργίες του και έτσι λεπτομέρειες υλοποίησης αποκρύπτονται. Έτσι δημιουργείται η βάση για βελτιώσεις στο πώς γίνονται αντιληπτά, στην ποιότητα και τη συντήρησή τους, και τέλος στην επεκτασιμότητά τους, ώστε να γίνουν τα απαραίτητα στοιχεία ορθά σχεδιασμένων Αντικειμενοστραφών συστημάτων.

Εστιάζοντας στα πλεονεκτήματα της Αντικειμενοστραφούς μεθοδολογίας, προκύπτουν τα παρακάτω:

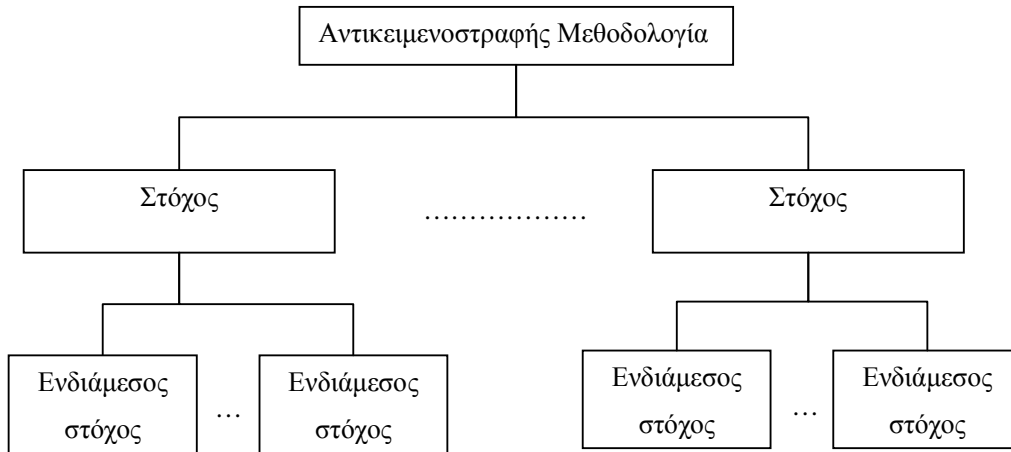
- i. Βελτιώνεται η παραγωγικότητα, μιας και η ανάπτυξη εφαρμογών διευκολύνεται από την επαναχρησιμοποίηση των υπαρχόντων στοιχείων με αποτέλεσμα η όλη διαδικασία τελικά να επιταχύνεται.
- ii. Παράγονται υψηλής ποιότητας συστήματα, μιας και έχει γίνει ήδη έλεγχος μεγάλου μέρους των στοιχείων.
- iii. Το κόστος συντήρησης ενός τέτοιου συστήματος είναι μικρότερο, αφού κάθε αλλαγή μπορεί να εστιαστεί και να γίνει εύκολα αντιληπτή η περιοχή του προβλήματος.
- iv. Η επαναχρησιμοποίηση είναι συνέπεια του γεγονότος ότι γίνεται συγκέντρωση και ορθή διαχείριση των στοιχείων, τα οποία είτε αναπτύσσονται εσωτερικά του συστήματος, είτε αποκτώνται εξωτερικά από άλλα συστήματα.
- v. Διευκολύνεται η διαχείριση, μιας και ελαττώνεται η πολυπλοκότητα. Διασπάζοντας μια πολύπλοκη λύση στα συστατικά της και ενθυλακώνοντας κάθε στοιχείο σε άλλα, μπορεί εύκολα να γίνει η διαχείρισή τους.

Η αντικειμενοστραφής μεθοδολογία χωρίζεται σε στάδια. Κάθε στάδιο αποτελείται από στόχους, οι οποίοι αποτελούνται από αριθμό ενδιάμεσων στόχων (Εικόνα 18).

Τόσο οι στόχοι όσο και οι ενδιάμεσοι στόχοι, μπορούν να ομαδοποιηθούν σε μια σειρά ενεργειών, που τελικά οδηγεί στη ολοκλήρωση της εφαρμογής.

Πρώτος στόχος μπορεί να θεωρηθεί το γενικό πλάνο της εφαρμογής. Σκοπό έχει:

- i. Να αναγνωριστεί ο βασικός σκοπός,
- ii. Να οριστούν οι απαιτήσεις,
- iii. Να γίνει εκτίμηση των συστατικών, που θα επαναχρησιμοποιηθούν,
- iv. Να εκτιμηθεί το κατά πόσο είναι εφικτό να επιτευχθεί το εγχείρημα.



Εικόνα 18: Δομή Αντικειμενοστραφούς Μεθοδολογίας

Στη συνέχεια ακολουθεί ο ορισμός της αρχιτεκτονικής του εγχειρήματος. Πιο συγκεκριμένα:

- i. Ορίζονται τα στοιχεία και συστατικά της αρχιτεκτονικής που θα ακολουθηθεί.
- ii. Προσδιορίζονται οι προδιαγραφές.
- iii. Προσδιορίζονται σχέσεις μεταξύ των δομικών στοιχείων.
- iv. Γίνεται κατανόηση του πως θα πρέπει να ικανοποιηθούν οι απαιτήσεις.

Ακολουθεί η τεχνική αρχιτεκτονική, όπου:

- i. Ορίζονται τα τεχνικά στοιχεία και χαρακτηριστικά.
- ii. Προσδιορίζονται οι προδιαγραφές τους.
- iii. Προσδιορίζονται οι σχέσεις και εξαρτήσεις μεταξύ αυτών καθώς και των στοιχείων

του προηγούμενου σταδίου

- iv. Ορίζονται , μοντελοποιούνται και ελέγχονται τα στοιχεία που ορίστηκαν.

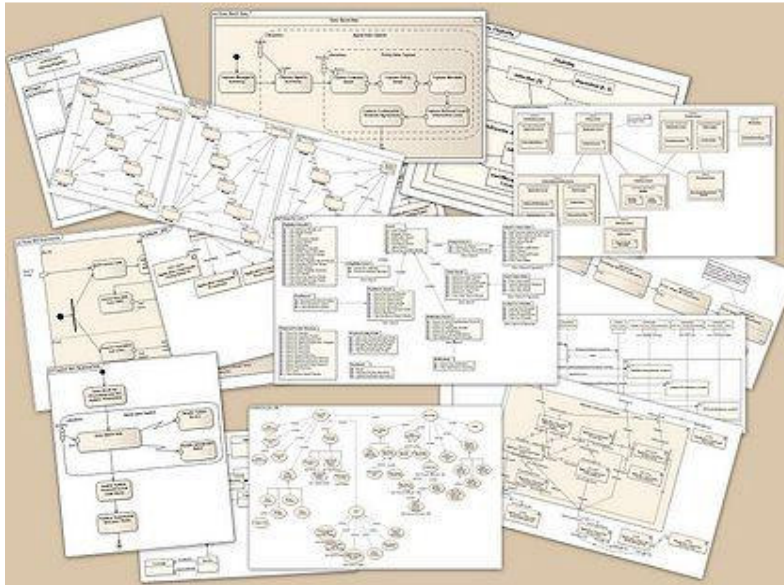
Στη συνέχεια λαμβάνει χώρα επαυξητική σχεδίαση και δόμηση, όπου αναπτύσσονται λύσεις με τέτοιο τρόπο, ώστε η παράδοση να γίνει όσο το δυνατόν ταχύτερα και γίνεται έλεγχος ώστε να επιβεβαιωθεί ο ορθός τρόπος λειτουργίας του κώδικα που αναπτύχθηκε.

Η διαδικασία ολοκληρώνεται με την ανάπτυξη, όπου το τελικό σύστημα , ελεγμένο και δοκιμασμένο, παραδίδεται, γίνεται μετατροπή των δεδομένων σε μορφή συμβατή του συστήματος και εκπαιδεύονται οι τελικοί χρήστες.

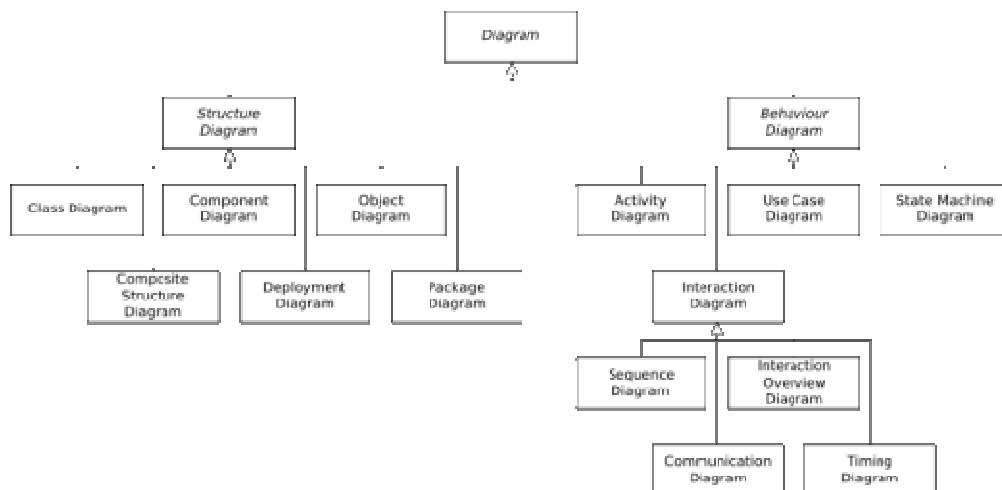
Σε όλη τη διαδικασία γίνεται ευρεία χρήση των Use Cases. Το συγκεκριμένο μοντέλο περιγράφει όλες τις λειτουργικές απαιτήσεις που θα αναπτυχθούν. Επίσης ιδιαίτερα χρήσιμο είναι το μοντέλο των κλάσεων, που βοηθά τη δόμηση από το μοντέλο Οντοτήτων – Συσχετίσεων σε δομημένες μεθόδους και έτσι προκύπτει ένα ευρύτερο σύνολο σχέσεων, που μπορούν να μοντελοποιηθούν με τέτοιο τρόπο ώστε να διευκολύνεται η επαναχρησιμοποίηση των υπάρχοντων κλάσεων.

Άλλες τεχνικές που χρησιμοποιούνται είναι τα ιεραρχικά διαγράμματα και οι περιγραφές των λειτουργιών καθώς και τα διαγράμματα ροής.

Απαραίτητο εργαλείο για όλα τα παραπάνω , είναι η γλώσσα UML. Η συγκεκριμένη γλώσσα μοντελοποίησης, επιτρέπει τον προσδιορισμό, οπτικοποίηση ,τροποποίηση, δόμηση και τεκμηρίωση των στοιχείων ενός αντικειμενοστραφούς συστήματος. Δίνει ένα τυποποιημένο τρόπο να γραφεί το σχέδιο ενός συστήματος, χρησιμοποιώντας εννοιολογικά στοιχεία, όπως τους χρήστες (με την εννοιολογική έννοια), τις διαδικασίες και τις δραστηριότητες του συστήματος, αλλά και συγκεκριμένα πράγματα, όπως δηλώσεις γλωσσών προγραμματισμού, σχήματα ΒΔ και επαναχρησιμοποιήσιμα στοιχεία λογισμικού. Μπορεί να χρησιμοποιηθεί με κάθε είδους διαδικασία, μέσω του κύκλου ζωής του λογισμικού και παράλληλα με διαφορετικές τεχνολογίες ανάπτυξης. Αποτελεί πλέον ένα στάνταρ γλώσσας μοντελοποίησης. Η UML κάνει ευρεία χρήση των διαγραμμάτων (Εικόνα 19), τα οποία χωρίζονται σε δύο κατηγορίες: Στατικά ή δομικά (διαγράμματα κλάσεων, σύνθετα διαγράμματα δόμησης κλπ), που περιγράφουν τα στοιχεία που πρέπει να υπάρχουν στο σύστημα και δυναμικά ή συμπεριφορικά που εστιάζουν στο τι πρέπει να συμβαίνει στο υπό δημιουργία σύστημα (διαγράμματα ακολουθίας, δραστηριοτήτων, κατάστασης κλπ)(Εικόνα 20).



Εικόνα 19: Διαγράμματα UML (πηγή: Wikipedia)



Εικόνα 20: Κατηγοριοποίηση Διαγραμμάτων UML (Πηγή: wikipedia)

3.6 Χωρικές Βάσεις Δεδομένων

Οι χωρικές βάσεις δεδομένων αποτελούν την αιχμή της έρευνας στον τομέα των βάσεων δεδομένων, με βασική εφαρμογή την υποστήριξη των Γεωγραφικών Συστημάτων Πληροφοριών, τα οποία είναι ιδιαίτερα χρήσιμα στην χαρτογράφηση περιοχών αλλά και δικτύων (οδικών, τηλεφωνικών κλπ.).

Μια χωρική βάση δεδομένων παρέχει όλα όσα και μια απλή βάση δεδομένων αλλά επιπλέον και τη δυνατότητα να παρασταθούν και να αποθηκευτούν τύποι χωρικών δεδομένων (σημεία στο χώρο, ευθείες ή ακόμα και πολύπλοκα γεωμετρικά σχήματα). Επίσης υποστηρίζει τις σχέσεις μεταξύ τους, μεταξύ των ιδιοτήτων τους και των διαφόρων πράξεων τους. Τέλος δίνουν τη δυνατότητα αποδοτικού και απλού τρόπου αναζήτησης και προσπέλασης των τύπων χωρικών δεδομένων.

Γενικά, μια χωρική βάση δεδομένων αποθηκεύει αντικείμενα που περιγράφονται με χωρικά χαρακτηριστικά. Οι χωρικές βάσεις δεδομένων υποστηρίζουν έννοιες για βάσεις δεδομένων που καταγράφουν αντικείμενα σ' έναν πολυδιάστατο χώρο. Έτσι, τα περισσότερα αν όχι όλα ΣΔΒΔ, έχουν και τις κατάλληλες επεκτάσεις ώστε να συμπεριληφθούν γεωμετρικές έννοιες δύο διαστάσεων όπως σημεία, γραμμές και τμήματα γραμμών, κύκλοι, πολύγωνα και τόξα και να προσδιορίζονται τα χωρικά χαρακτηριστικά των αντικειμένων. Επιπλέον περιλαμβάνονται και χωρικές πράξεις για να εφαρμοσθούν στα χωρικά χαρακτηριστικά των αντικειμένων -για παράδειγμα, να υπολογισθεί η απόσταση μεταξύ δύο αντικειμένων.

Αντίστοιχα, υπάρχουν και σε αναλογία με τις κλασσικές επερωτήσεις, χωρικές επερωτήσεις, οι οποίες χωρίζονται σε 3 βασικές κατηγορίες:

i. Επερώτηση διαστήματος: Βρίσκει τα αντικείμενα ενός συγκεκριμένου τύπου που βρίσκονται μέσα σε δεδομένη χωρική περιοχή ή σε συγκεκριμένη απόσταση από δοθείσα θέση.

ii. Επερώτηση για τον κοντινότερο γείτονα: Βρίσκει ένα αντικείμενο συγκεκριμένου τύπου που βρίσκεται πιο κοντά σε δεδομένη θέση.

iii. Χωρικές συνενώσεις ή επικαλύψεις: Τυπικά συνενώνει αντικείμενα δύο τύπων με βάση κάποια χωρική συνθήκη, όπως αντικείμενα που τέμνονται ή επικαλύπτονται χωρικά ή που βρίσκονται σε κάποια απόσταση το ένα από το άλλο.

Ειδικά ευρετήρια χρησιμοποιούνται για την απάντηση των χωρικών ερωτημάτων, με πιο τυπική περίπτωση, τα R-δέντρα και τις παραλλαγές τους.

Τα R-δένδρα ομαδοποιούν στους ίδιους τερματικούς κόμβους (φύλλα) ενός δομημένου δενδροειδούς ευρετηρίου τα αντικείμενα που βρίσκονται φυσικά κοντά με κάποια χωρική προσέγγιση. Αφού ένα φύλο μπορεί να δείχνει μόνο σε κάποιο πλήθος αντικειμένων, χρειάζονται αλγόριθμοι για τον διαχωρισμό του χώρου σε ορθογώνιους υποχώρους που περιέχουν τα αντικείμενα.

Τυπικά κριτήρια για τον διαχωρισμό του χώρου περιλαμβάνουν την ελαχιστοποίηση των ορθογώνιων περιοχών, αφού αυτό οδηγεί σε γρήγορη συντόμευση του χώρου αναζήτησης, τα γνωστά ελάχιστα περιβάλλοντα παραλληλόγραμμα.

Προβλήματα όπως η ύπαρξη αντικειμένων με επικαλυπτόμενες χωρικές περιοχές αντιμετωπίζονται με διαφορετικούς τρόπους από τις πολλές διαφορετικές παραλλαγές των R-δένδρων. Οι εσωτερικοί κόμβοι των R-δένδρων αντιστοιχούν σε ορθογώνια που η περιοχή τους καλύπτει όλα τα ορθογώνια του υποδένδρου τους. Επομένως τα R-δένδρα μπορούν να απαντήσουν εύκολα επερωτήσεις όπως: βρες όλα τα αντικείμενα σε δοθείσα περιοχή, περιορίζοντας τη δενδρική αναζήτηση σε εκείνα τα υποδένδρα των οποίων τα ορθογώνια τέμνονται με την περιοχή που δίδεται στην επερώτηση.

4

Ανάλυση Απαιτήσεων Συστήματος

Στο κεφάλαιο αυτό παρουσιάζονται η ανάλυση απαιτήσεων του συστήματος. Ειδικότερα αναφέρονται οι απαιτήσεις και οι προδιαγραφές, περιγράφεται η αρχιτεκτονική και τέλος περιγράφονται τα επιμέρους υποσυστήματα και οι λειτουργίες. Κατάληξη όλων των παραπάνω, ο εννοιολογικός και φυσικός σχεδιασμός της Βάσης Δεδομένων και τελικά η δημιουργία της.

4.1 Αρχιτεκτονική

Το σύστημα που υλοποιείται, αποτελεί μια εφαρμογή client-server. Υπάρχει δηλαδή το κεντρικό τμήμα της (ο server), στον οποίο βρίσκεται στην ουσία η ΒΔ και η εφαρμογή γενικότερα, και οι clients, που ανάλογα με τα δικαιώματα πρόσβασης που έχουν, προσπελαίνουν, σε web περιβάλλον, τη ΒΔ και χρησιμοποιούν ανάλογα την εφαρμογή. Επίσης υπάρχει και ο διαχωρισμός της εφαρμογής στο χωρικό και το μη χωρικό κομμάτι της. Τα καθαρά διαδικαστικά τμήματα (απόφαση κίνησης, συντήρηση οχήματος κλπ) ανήκουν στο μη χωρικό κομμάτι, χωρίς αυτό να σημαίνει ότι δεν ασχολείται καθόλου με αυτό (πχ. σε ένα κινούμενο όχημα του δίνουμε πληροφορίες για τον πιο κοντινό σταθμό συντήρησης). Στο χωρικό κομμάτι έχουμε όλες τις πληροφορίες που αφορούν στα χωρικά δεδομένα.

Το σύστημα ,λοιπόν, αποτελείται από 5 υποσυστήματα (Εικόνα 21), τα οποία διαχειρίζονται ειδικές λειτουργίες το καθένα:

i. Υποσύστημα Ταξινόμησης και Καταχώρησης οχημάτων: Είναι το υποσύστημα που είναι υπεύθυνο για την καταχώρηση, ταξινόμηση και αναγνώριση των οχημάτων.

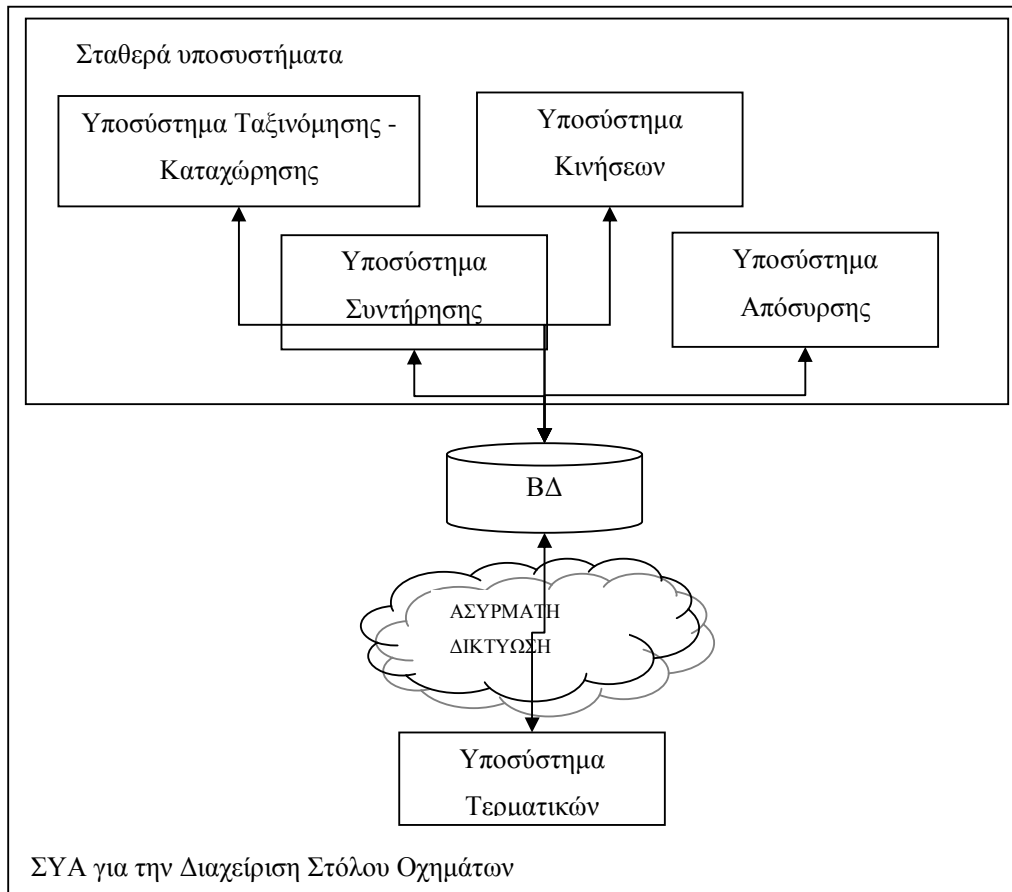
ii. Υποσύστημα Κινήσεων Οχημάτων: Είναι το υποσύστημα που συλλέγει τις αιτήσεις μεταφοράς και λαμβάνει την απόφαση αν μια κίνηση θα υλοποιηθεί ή όχι. Επίσης μέσω του υποσυστήματος αυτού, εκδίδονται τα δελτία κίνησης και ενημερώνεται η λοιπή διαθεσιμότητα οχημάτων.

iii. Υποσύστημα Συντήρησης Οχημάτων: Είναι υπεύθυνο για την συντήρηση των οχημάτων, τόσο την προληπτική όσο και την επισκευαστική.

iv. Υποσύστημα Απόσυρσης Οχημάτων: Λαμβάνει τις αιτήσεις απόσυρσης οχημάτων και αποφασίζει αν κάποιο όχημα θα αποσυρθεί, οπότε και εκδίδει δελτίο απόσυρσης.

v. Υποσύστημα κινουμένων οχημάτων: Κάθε όχημα στέλνει ενημερώσεις για την τρέχουσα θέση του, και ενημερώνεται από τα άλλα συστήματα, μέσω της ΒΔ.

Κάθε ένα από τα παραπάνω υποσυστήματα χειρίζεται δεδομένα, χωρικά ή μη. Τα 4 πρώτα υποσυστήματα δημιουργούν ένα μεγάλο υποσύστημα, που είναι σε κάποιο σταθερό σημείο – κέντρο ελέγχου, ενώ το 5^ο είναι κινητό και αντιστοιχεί στα οχήματα, τα οποία κινούνται.



Εικόνα 21: Αρχιτεκτονική Συστήματος

4.2 Περιγραφή Λειτουργιών

Το υλοποιούμενο σύστημα αποτελείται από 5 υποσυστήματα. Κάθε υποσύστημα εκτελεί συγκεκριμένες λειτουργίες, για τις οποίες είναι σχεδιασμένο και συνεργάζεται με τα υπόλοιπα προκειμένου να επιτευχθεί ο συνολικός σκοπός. Στη συνέχεια θα περιγραφούν για κάθε υποσύστημα οι επιμέρους λειτουργίες, ώστε να γίνει πλήρως κατανοητός ο ρόλος του.

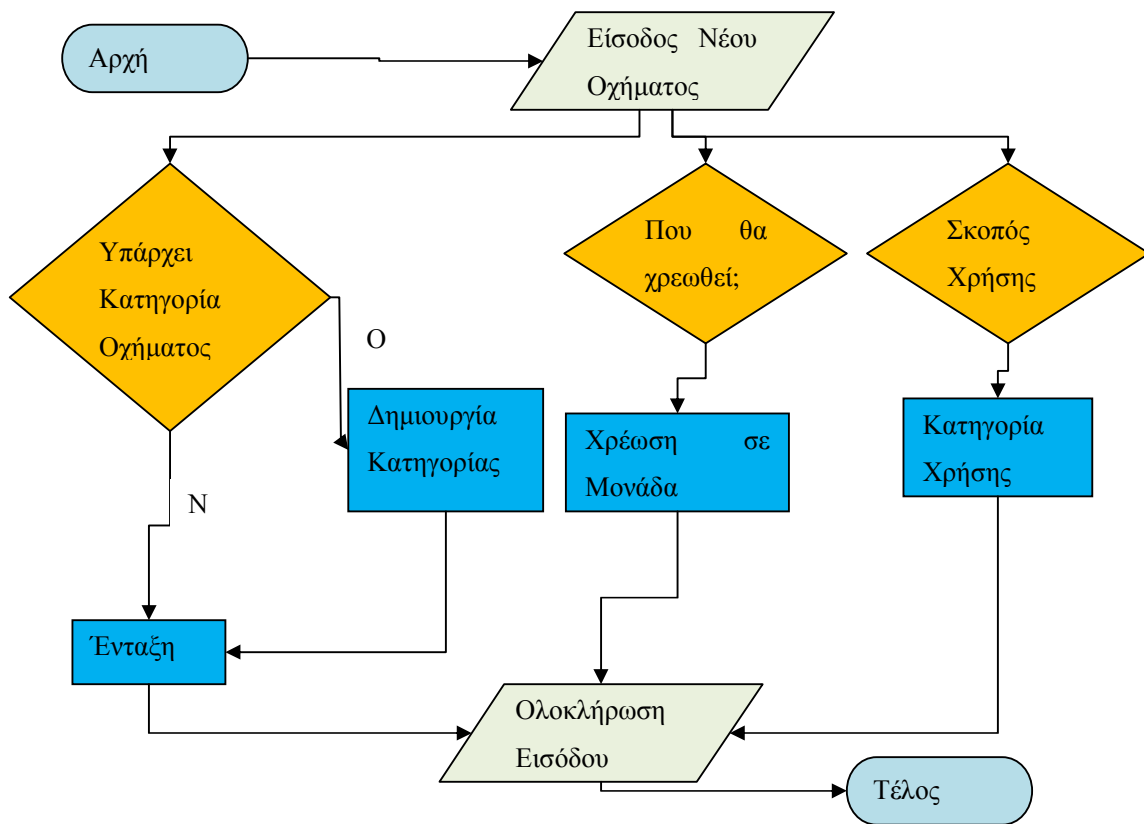
Η περιγραφή λειτουργιών προέρχεται μετά από συζητήσεις με τους υπονήφιους χρήστες της εφαρμογής, έτσι ώστε να κατανοηθούν οι απαιτήσεις τους τόσο σε λειτουργίες όσο και σε δεδομένα. Γίνεται χρήση διαγραμμάτων ροής δεδομένων, ακολουθιακών διαγραμμάτων και σεναρίων, ώστε να προσδιοριστούν οι λειτουργικές απαιτήσεις.

4.2.1 Υποσύστημα Καταχώρησης - Ταξινόμησης

Είναι το υποσύστημα εκείνο το οποίο είναι υπεύθυνο για την καταχώρηση ενός νέου οχήματος, την ταξινόμησή του σε κατηγορίες ανάλογα με τον τύπο του και την τήρηση διαχειριστικών στοιχείων για αυτό. Συγκεκριμένα:

- i. Τηρεί στοιχεία για τις κατηγορίες οχημάτων (οχήματα προσωπικού, μεταφοράς υλικών κλπ) και έχει την δυνατότητα να εισάγει νέες κατηγορίες ανάλογα με τις απαιτήσεις.
- ii. Για κάθε όχημα τηρεί στοιχεία ταυτοποίησης (αριθμό πλαισίου, πινακίδας). Επίσης τηρεί στοιχεία για το σε ποιο τμήμα της επιχείρησης είναι χρεωμένο.
- iii. Τηρεί στοιχεία χρήσης του οχήματος, αν δηλαδή χρησιμοποιείται για εκπαίδευση, για αποστολή ή είναι ανενεργό.

Για την καλύτερη κατανόηση των διεργασιών που λαμβάνουν χώρα στο υποσύστημα αυτό, όπως και σε όλα τα υπόλοιπα, θα γίνει χρήση διαγραμμάτων ροής δεδομένων. Τα διαγράμματα αυτά, με την χρήση τυποποιημένων σχημάτων (παραλληλόγραμμο αντιστοιχεί σε διεργασία, ρόμβος σε απόφαση, πλάγιο παραλληλόγραμμο σε είσοδο – έξοδο κλπ), οπτικοποιούν τις διεργασίες και διευκολύνουν στη συνέχεια την ανάπτυξη του λογισμικού. Βασίζόμενοι, λοιπόν, στις αρμοδιότητες του πρώτου υποσυστήματος, όπως αυτές έχουν καταγραφεί από την ανάλυση απαιτήσεων, δημιουργούμε το ακόλουθο διάγραμμα ροής δεδομένων (Εικόνα 22).



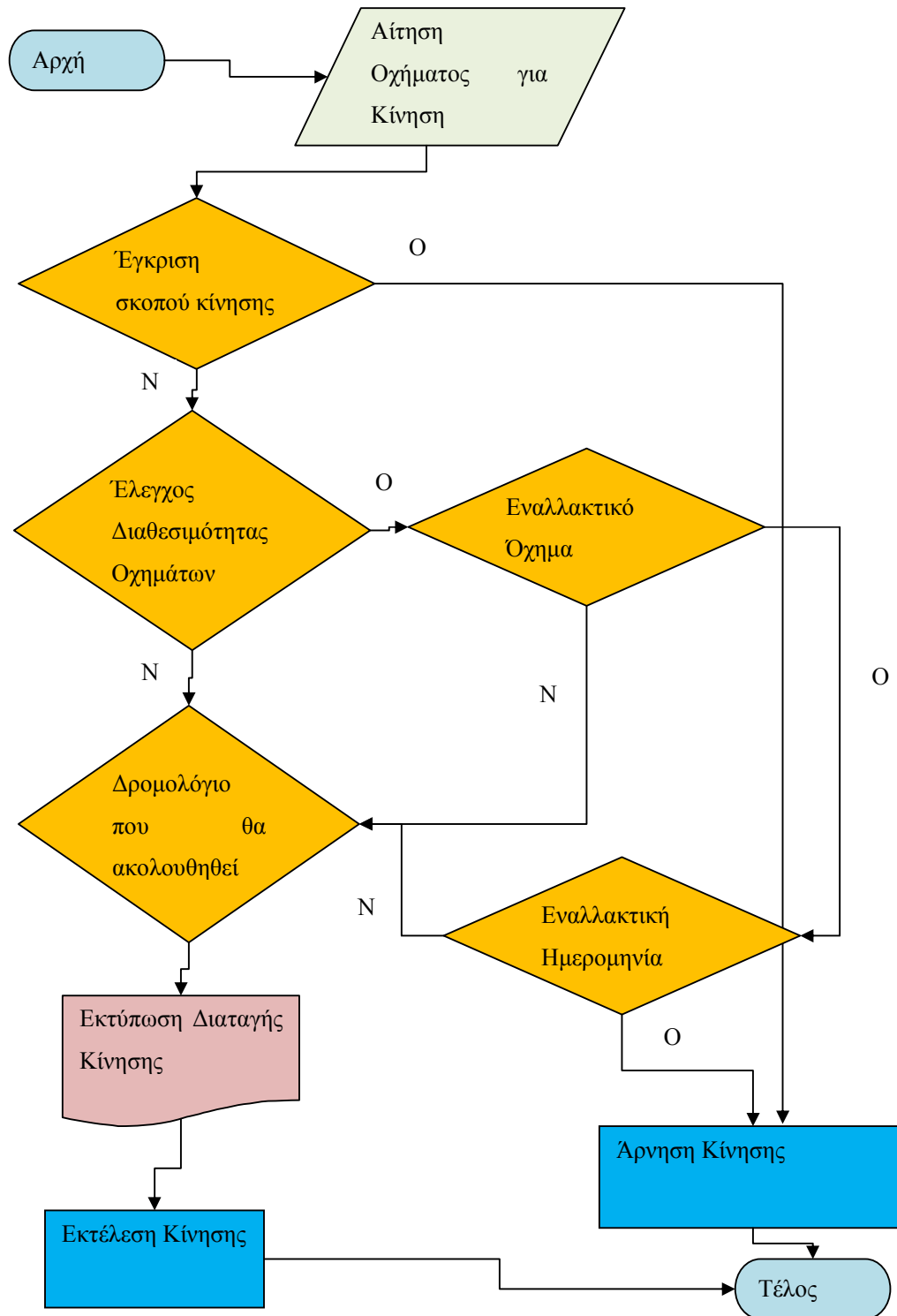
Εικόνα 22: Διάγραμμα Ροής Υποσυστήματος Καταχώρησης - Ταξινόμησης

4.2.2 Υποσύστημα Κινήσεων

Το υποσύστημα αυτό είναι υπεύθυνο για κάθε κίνηση η οποία γίνεται και ελέγχεται από το σύστημα. Συγκεκριμένα:

- i. Μέσω αυτού του συστήματος γίνονται οι αιτήσεις μεταφορών και τηρούνται στοιχεία για αυτές (τι όχημα ζητάτε, πόσα οχήματα χρειάζεται μια συγκεκριμένη κίνηση, ποιος θα πάει και που, ποιος ζητά την κίνηση και για πότε κλπ).
- ii. Λαμβάνεται απόφαση επί του αιτήματος κίνησης ώστε να λάβουν γνώση οι ενδιαφερόμενοι. Τα στοιχεία που συλλέγονται και τηρούνται είναι τι κατηγορίας οχήματα τελικά θα κινηθούν, σε τι ποσότητα, που θα πάνε και πότε, ποιος είναι ο αιτών την αίτηση, που τελικά θα έχει το κέρδος από την όλη διαδικασία.
- iii. Το υποσύστημα αυτό τηρεί όλα τα απαιτούμενα στοιχεία για τις θέσεις ενδιαφέροντος, ανάλογα με την περίπτωση, καθώς και τα στοιχεία των πόλεων, του οδικού δικτύου κλπ.
- iv. Εκδίδεται το δελτίο κίνησης με βάση το οποίο επιτρέπεται η κίνηση και περιέχει όλες τις πληροφορίες που απαιτούνται για να εκτελεστεί αυτή.

Το διάγραμμα ροής δεδομένων για το υποσύστημα αυτό παρουσιάζεται στο ακόλουθο σχήμα (Εικόνα 23).



Εικόνα 23: Διάγραμμα Ροής Υποσυστήματος Κινήσεων

4.2.3 Υποσύστημα Συντήρησης

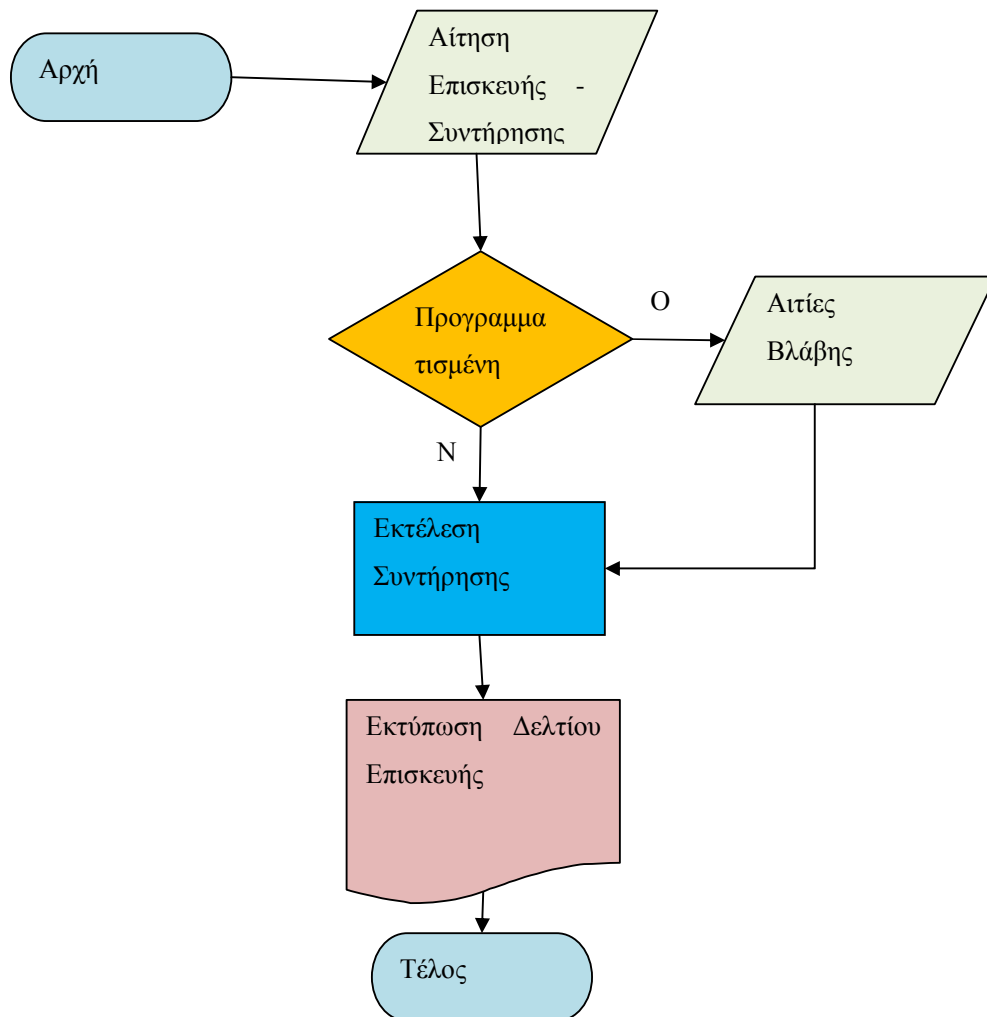
Το τρίτο κατά σειρά υποσύστημα είναι αυτό της συντήρησης, η οποία χωρίζεται σε δύο σκέλη: Την επισκευαστική και την προληπτική.

i. Η προληπτική συντήρηση λαμβάνει χώρα ανά τακτά χρονικά διαστήματα, με σκοπό να διαπιστωθεί η κατάσταση του οχήματος και να αποτραπούν βλάβες. Τηρούνται ,λοιπόν στοιχεία, για το πότε έλαβε χώρα η τελευταία, τι αποτελέσματα είχε και πότε είναι προγραμματισμένη η επόμενη, καθώς και το κόστος που είχε.

ii. Η επισκευαστική συντήρηση λαμβάνει χώρα όποτε απαιτηθεί προκειμένου να αντιμετωπισθούν έκτακτες βλάβες. Τηρούνται πληροφορίες κόστους επισκευής, ημερομηνίας αιτήματος, τρέχουσας θέσης οχήματος.

iii. Εκδίδεται δελτίο επισκευής, ώστε να τηρούνται πληροφορίες για όλες τις εργασίες συντήρησης που έχουν εκτελεστεί στο όχημα, μαζί με τις πληροφορίες που τις αφορούν.

Οι διαδικασίες εμφανίζονται στο διάγραμμα ροής που ακολουθεί (Εικόνα 24).



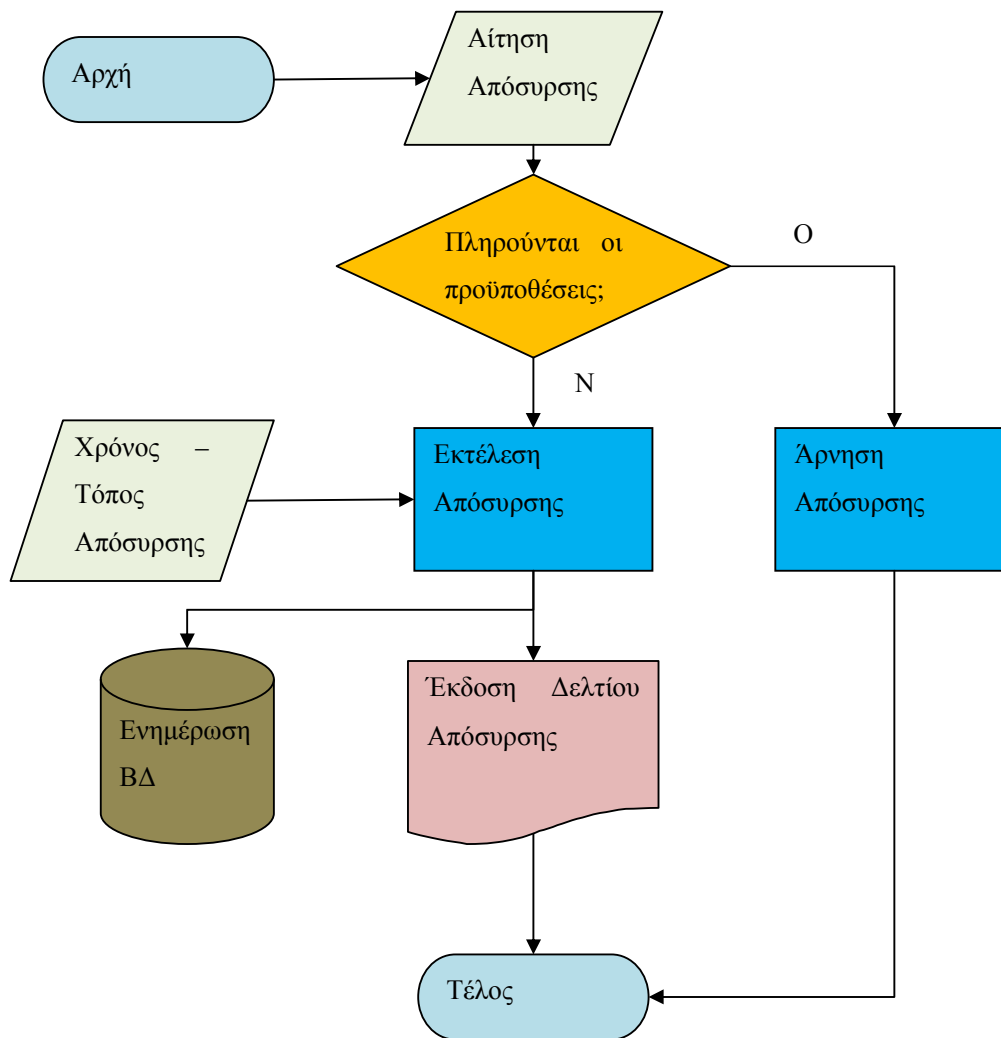
Εικόνα 24: Διάγραμμα Ροής Υποσυστήματος Συντηρήσεως

4.2.4 Υποσύστημα Απόσυρσης

Το υποσύστημα απόσυρσης οχημάτων είναι το υποσύστημα εκείνο που ασχολείται με όλες τις διαδικασίες απόσυρσης ενός οχήματος.

- i. Συγκεντρώνει τις αιτήσεις απόσυρσης ενός οχήματος που αποστέλλονται από τους ενδιαφερόμενους, μαζί με τις απαραίτητες πληροφορίες.
- ii. Λαμβάνει απόφαση για το αν ένα όχημα είναι πλέον αναγκαίο να αποσυρθεί, τότε θα γίνει αυτό και που θα είναι ο προορισμός του.
- iii. Εκδίδει το δελτίο απόσυρσης και ενημερώνει για την διαφοροποίηση στον αριθμό των οχημάτων που είναι διαθέσιμα.

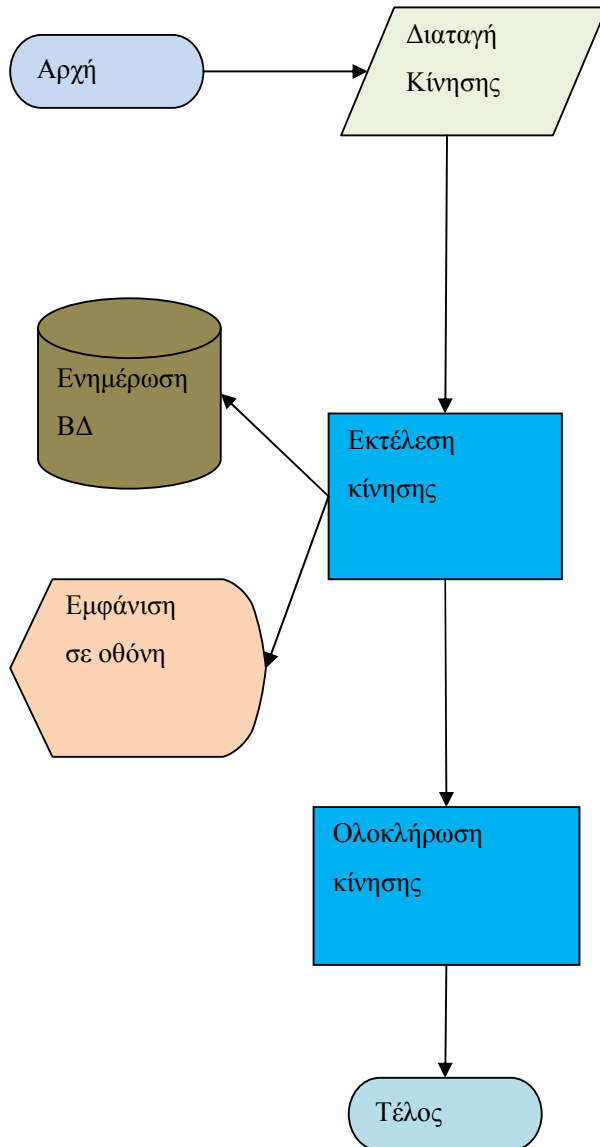
Οι διαδικασίες του Υποσυστήματος Απόσυρσης, φαίνονται στο διάγραμμα που ακολουθεί (Εικόνα 25).



Εικόνα 25: Διάγραμμα Ροής Υποσυστήματος Απόσυρσης

4.2.5 Υποσύστημα Τερματικών

Είναι το κινητό υποσύστημα της εφαρμογής. Τα κινούμενα οχήματα, αποστέλλουν τη θέση τους στον κεντρικό σταθμό ώστε να γνωρίζουμε που βρίσκονται ανά πάσα στιγμή, και λαμβάνουν πληροφορίες σχετικά με τη διαδρομή που πρέπει να εκτελέσουν καθώς και χρήσιμα στοιχεία για αυτή (συμβεβλημένοι σταθμοί ανεφοδιασμού και συνεργεία, γειτονικές εγκαταστάσεις, ποια άλλα οχήματα βρίσκονται πλησίον κλπ). Οι διαδικασίες φαίνονται στο διάγραμμα ροής που ακολουθεί (Εικόνα 26).



Εικόνα 26: Διάγραμμα Ροής Υποσυστήματος Τερματικών

4.3 Μοντέλο Οντοτήτων Συσχετίσεων

Ο εννοιολογικός σχεδιασμός της ΒΔ που θα χρησιμοποιηθεί για τις ανάγκες της εφαρμογής, είναι το επόμενο στάδιο στη διαδικασία σχεδιασμού της. Χρησιμοποιώντας ένα εννοιολογικό μοντέλο υψηλού επιπέδου δημιουργούμε το εννοιολογικό σχήμα, δηλαδή μια περιεκτική περιγραφή των απαιτήσεων των χρηστών σχετικά με τα δεδομένα, με λεπτομερείς περιγραφές των τύπων δεδομένων, των συσχετίσεων και των περιορισμών. Με το εννοιολογικό σχήμα εξασφαλίζεται ότι όλες οι απαιτήσεις έχουν συμπεριληφθεί και δεν περιέχουν αντιθέσεις. Το πλέον διαδεδομένο μοντέλο δεδομένων υψηλού επιπέδου που χρησιμοποιείται για τον εννοιολογικό σχεδιασμό, είναι το μοντέλο Οντοτήτων – Συσχετίσεων (ΟΣ), το οποίο και θα χρησιμοποιήσουμε για τον σχεδιασμό μας. Στην περίπτωση που εξετάζεται, η ύπαρξη χωρικής πληροφορίας στη βάση, έχει ως αποτέλεσμα, το μοντέλο ΟΣ να επεκταθεί (ΕΟΣ) ώστε να περιλάβει και την χωρική πληροφορία.

Μια πολύ γρήγορη περιγραφή της σχεδιαζόμενης ΒΔ και με βάση όλες τις απαιτήσεις που έχουν καταγραφεί, έχει ως αποτέλεσμα τα παρακάτω:

i. Ένα όχημα όταν αγοράζεται, εισέρχεται στο σύστημα, κατηγοριοποιείται, χρεώνεται σε μια μονάδα και χρησιμοποιείται για κάποιο λόγο, ο οποίος μπορεί να αλλάζει ανά διαστήματα (πχ ένα όχημα όταν παλιώνει χρησιμοποιείται για εκπαίδευση).

ii. Ένα όχημα συμμετέχει σε κινήσεις οι οποίες ξεκινούν από κάποια πόλη και τελειώνουν σε κάποια άλλη. Ιδανικά θα υπάρχει και δρομολόγηση, δηλαδή ποιοι δρόμοι θα ακολουθηθούν και ποια είναι η βέλτιστη διαδρομή.

iii. Οι κινήσεις που εκτελούνται είναι αποτέλεσμα αιτήσεων που γίνονται και για τις οποίες (τις αιτήσεις) λαμβάνεται κάποια απόφαση (θετική ή αρνητική).

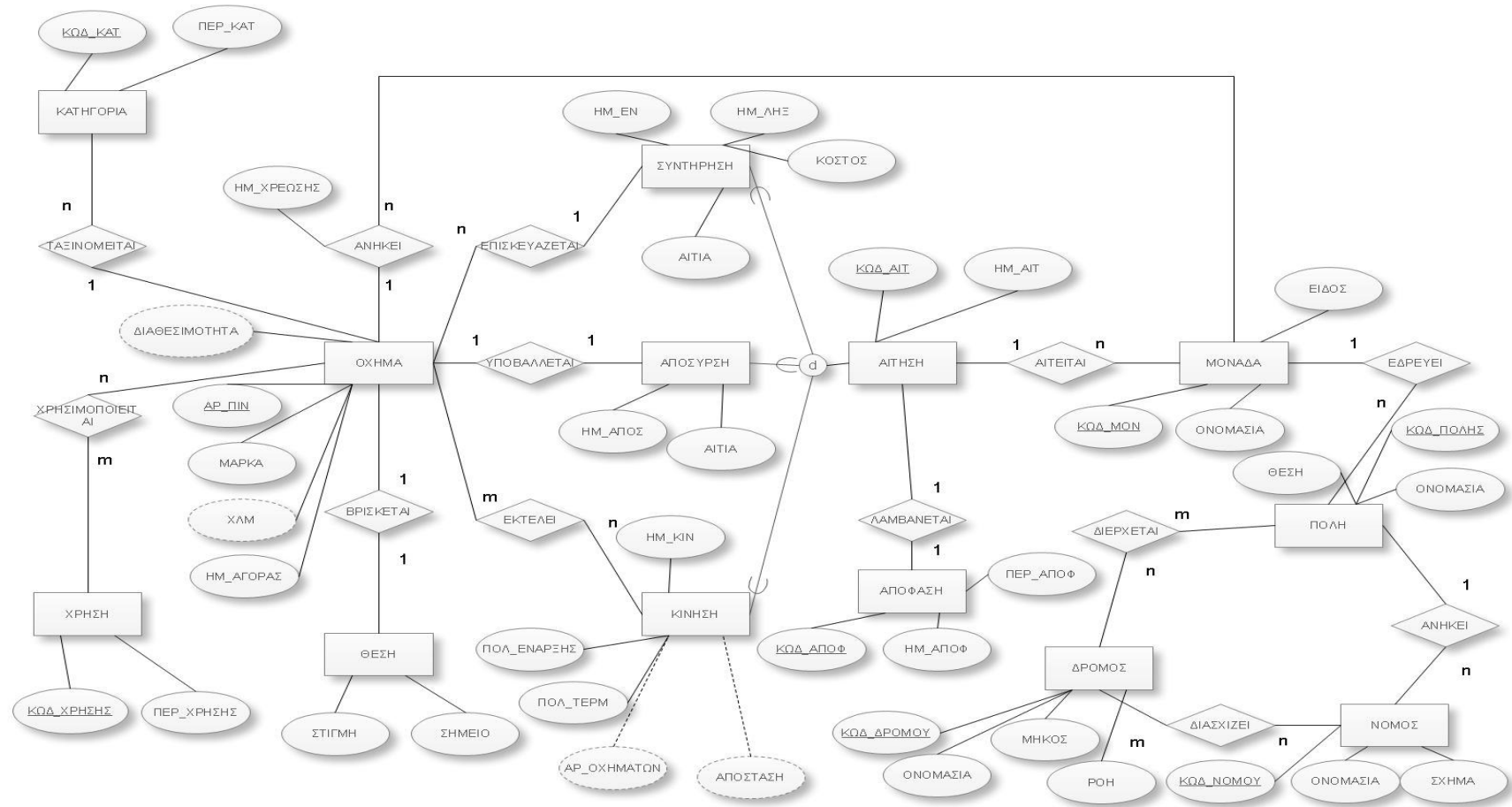
iv. Ένα όχημα, προγραμματισμένα ή έκτακτα, υποβάλλεται σε κάποια συντήρηση, η οποία είναι αποτέλεσμα αντίστοιχου αιτήματος της μονάδας στην οποία είναι χρεωμένο το όχημα.

v. Το όχημα, κατά την κίνηση του, στέλνει μηνύματα με την τρέχουσα θέση του ώστε να είναι δυνατή η παρακολούθησή του.

vi. Το όχημα, μετά από αίτηση της Μονάδας και απόφαση του αρμόδιου τμήματος, αποσύρεται από την κυκλοφορία.

vii. Υπάρχουν καταχωρημένα το οδικό δίκτυο, οι πόλεις και οι νομοί της περιοχής ενδιαφέροντος.

Προκύπτει, επομένως, το διάγραμμα Ο-Σ, της επόμενης σελίδας (Εικόνα 27).



Εικόνα 27: Διάγραμμα Οντοτήτων - Συσχετίσεων ΒΔ

Το παραπάνω διάγραμμα Ο-Σ, έχει τα παρακάτω χαρακτηριστικά:

i. Οντότητες που αντιπροσωπεύουν τα βασικά στοιχεία της εφαρμογής (όχημα, μονάδα, κίνηση, συντήρηση κλπ). Κάποιες από τις οντότητες είναι της κατηγορίας ISA. Οι οντότητες αυτές είναι οντότητες που ανήκουν σε μια μεγαλύτερη κατηγορία οντοτήτων (στην περίπτωσή μας, γίνεται αίτηση που μπορεί να είναι αίτηση για κίνηση, συντήρηση ή απόσυρση). Οι οντότητες της μορφής ISA, έχουν τα γνωρίσματα της υπερκλάσης οντοτήτων στην οποία ανήκουν και τα επιπλέον δικά τους γνωρίσματα.

ii. Συσχετίσεις, που συνδέουν τις οντότητες μεταξύ τους, ώστε να μπορούν να αντληθούν στοιχεία από διαφορετικές οντότητες μεταξύ τους και να εξάγονται τα διάφορα συμπεράσματα.

iii. Γνωρίσματα, που δίνουν σε κάθε οντότητα και συσχέτιση τις απαραίτητες πληροφορίες, ώστε να μπορούν να αναγνωρίζονται και να δίνουν μετά από τα κατάλληλα ερωτήματα τα δεδομένα με συνέπεια.

iv. Γνωρίσματα στην οντότητα «ΚΙΝΗΣΗ» εκτός των άλλων και τα παραγόμενα γνωρίσματα «AP_ΟΧΗΜΑΤΩΝ» και «ΑΠΟΣΤΑΣΗ», που υπολογίζονται από τον συνολικό αριθμό οχημάτων που μετέχουν στη συγκεκριμένη κίνηση, και το συνολικό μήκος των δρόμων που μετέχουν σε αυτή. Όμοια, παραγόμενο γνώρισμα είναι και το «ΧΛΜ» της σχέσης «ΟΧΗΜΑ», που δείχνει τη συνολική απόσταση που έχει διανύσει το όχημα.

v. Η σχέση «ΘΕΣΗ», θα έχει μια εγγραφή για κάθε όχημα του στόλου, που θα δείχνει την τρέχουσα θέση του.

vi. Κατά την μετάβαση από το διάγραμμα Ο-Σ στο σχεσιακό σχήμα της ΒΔ, και την εκλέπτυνση του σχεδιασμού, κάποιες συσχετίσεις, όπως θα αναλυθεί και στη συνέχεια, θα μετατραπούν σε γνωρίσματα (συσχετίσεις 1-1, 1-n), ενώ κάποιες άλλες θα δημιουργήσουν νέες σχέσεις.

4.4 Βάση Δεδομένων

Αμέσως μετά τον εννοιολογικό σχεδιασμό, λαμβάνει χώρα ο λογικός σχεδιασμός. Στο στάδιο αυτό, με τη βοήθεια ενός ΣΔΒΔ, το εννοιολογικό σχήμα, που έχει δημιουργηθεί με τη βοήθεια ενός υψηλού επιπέδου μοντέλο δεδομένων, μετασχηματίζεται στο μοντέλο δεδομένων υλοποίησης. Έτσι προκύπτει ένα σχήμα της Βάσης Δεδομένων, εκφρασμένο στο μοντέλο δεδομένων υλοποίησης του ΣΔΒΔ.

Στην περίπτωσή μας, χρησιμοποιούμε το σχεσιακό μοντέλο δεδομένων οπότε το διάγραμμα Ο-Σ του εννοιολογικού σχεδιασμού θα μετατραπεί στο σχεσιακό σχήμα.

Κατά τη μετάβαση από τον εννοιολογικό σχεδιασμό στο λογικό, γίνεται και η εκλέπτυνση του σχήματος, όπου κάποιες συσχετίσεις μετατρέπονται σε γνώρισμα. Πιο αναλυτικά, αν μια συσχέτιση είναι της μορφής 1 – n ή 1-1 μετατρέπεται σε γνώρισμα μιας από τις οντότητες που συμμετέχουν στη συσχέτιση (η απόφαση ανήκει στον σχεδιαστή της βάσης). Αν είναι m-n, μετατρέπεται σε νέα σχέση. Για να επιτευχθεί αυτό, λαμβάνονται υπόψη οι κανόνες κανονικοποίησης της ΒΔ, ώστε να καταλήξουμε στο σχεσιακό σχήμα. Το πώς μετατρέπεται η κάθε συσχέτιση φαίνεται στον Πίνακα 2.

Ειδικά για την τελευταία συσχέτιση του πίνακα, («ΛΑΜΒΑΝΕΤΑΙ») πρέπει να αναφερθεί, ότι ως συσχέτιση 1-1, ολόκληρη η οντότητα «ΑΠΟΦΑΣΗ» γίνεται γνώρισμα στην οντότητα «ΑΙΤΗΣΗ» μεταφέροντας εκεί τα γνώρισμά της. Αντίθετα για την συσχέτιση «ΕΔΡΕΥΕΙ», αν και είναι 1-1 δεν θα μεταφερθεί σαν γνώρισμα σε κάποια από τις συσχετιζόμενες οντότητες, και αυτό διότι θέλουμε να υπάρχει μια σχέση που να κρατά τις τρέχουσες θέσεις των οχημάτων μας, χωρίς να γίνονται συνεχείς ενημερώσεις στη σχέση «ΟΧΗΜΑ». Επίσης για την πρώτη συσχέτιση, η οποία κανονικά δημιουργεί νέα σχέση, θα ακολουθηθεί διαφορετική προσέγγιση μιας και οι χρήσεις ενός οχήματος είναι συγκεκριμένες. Αντί να δημιουργήσουμε νέα σχέση, θα μεταφέρουμε την οντότητα «ΧΡΗΣΙΜΟΠΟΙΕΙΤΑΙ» σαν γνώρισμα στο «ΟΧΗΜΑ», όπου θα έχει σαν τύπο δεδομένων, τον τύπο δεδομένων σύνολο, από όπου θα μπορεί το όχημα να πάρει μια ή περισσότερες τιμές.

Οι συσχετίσεις στις χωρικές οντότητες (πόλη, δρόμος, νομός) είναι περιττές (πχ μια πόλη σε ποιο νομό ανήκει) αφού και μόνο από τα χωρικά γνώρισμα προσδιορίζονται οι συσχετίσεις αυτές. Παρόλα αυτά, είναι απαραίτητη η ύπαρξή τους, ώστε να ελαχιστοποιηθεί το σφάλμα που προκύπτει από τα Ελάχιστα Περιβάλλοντα Παραλληλόγραμμα, που πολλές φορές δίνει κάποια πόλη να ανήκει σε δύο νομούς, μιας και βρίσκεται σε σημείο που επικαλύπτεται από περισσότερα του ενός παραλληλόγραμμα, και εξαρτάται από την υλοποίηση που έχει ακολουθηθεί στο ΣΔΒΔ που θα χρησιμοποιηθεί.

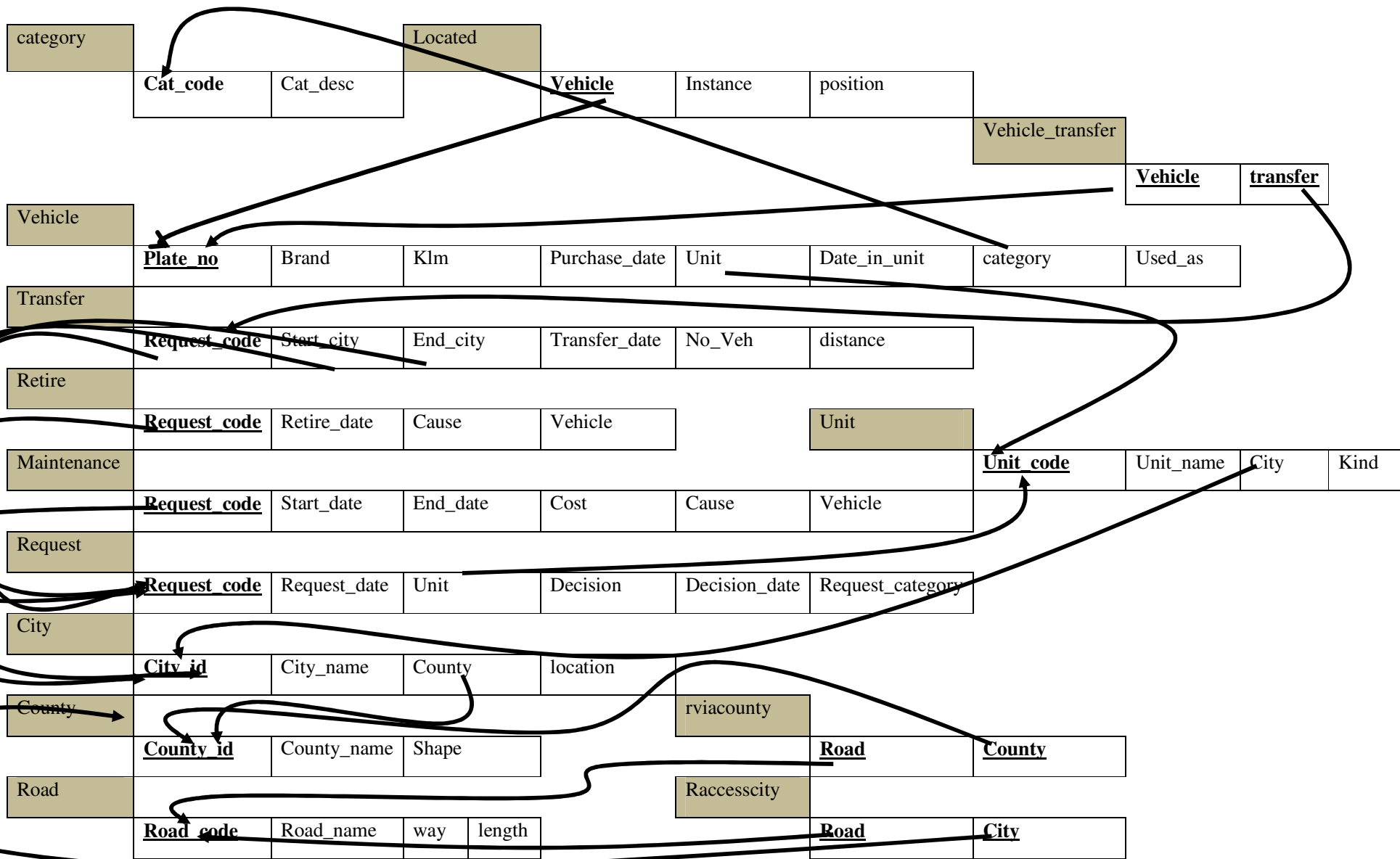
ΣΥΣΧΕΤΙΣΗ	ΟΝΤΟΤΗΤΑ 1	ΟΝΤΟΤΗΤΑ 2	ΤΥΠΟΣ ΣΥΣΧΕΤΙΣΗΣ	ΑΠΟΤΕΛΕΣΜΑ
ΧΡΗΣΙΜΟΠΟΙΕΙΤΑΙ	ΧΡΗΣΗ	ΟΧΗΜΑ	m-n	Νέα Σχέση
ΤΑΞΙΝΟΜΕΙΤΑΙ	ΚΑΤΗΓΟΡΙΑ	ΟΧΗΜΑ	n-1	Γνώρισμα στην «ΟΧΗΜΑ»
ΑΝΗΚΕΙ	ΟΧΗΜΑ	ΜΟΝΑΔΑ	1-n	Γνώρισμα στην «ΟΧΗΜΑ»
ΒΡΙΣΚΕΤΑΙ	ΟΧΗΜΑ	ΘΕΣΗ	1-n	Γνώρισμα στην «ΟΧΗΜΑ»
ΕΠΙΣΚΕΥΑΖΕΤΑΙ	ΟΧΗΜΑ	ΣΥΝΤΗΡΗΣΗ	n-1	Γνώρισμα στην «ΣΥΝΤΗΡΗΣΗ»
ΕΚΤΕΛΕΙ	ΟΧΗΜΑ	ΚΙΝΗΣΗ	m-n	Νέα Σχέση
ΥΠΟΒΑΛΛΕΤΑΙ	ΟΧΗΜΑ	ΑΠΟΣΥΡΣΗ	1-1	Γνώρισμα στην «ΑΠΟΣΥΡΣΗ»
ΔΙΑΣΧΙΖΕΙ	ΔΡΟΜΟΣ	ΝΟΜΟΣ	m-n	Νέα Σχέση
ΔΙΕΡΧΕΤΑΙ	ΔΡΟΜΟΣ	ΠΟΛΗ	m-n	Νέα Σχέση
ΑΝΗΚΕΙ	ΠΟΛΗ	ΝΟΜΟΣ	1-n	Γνώρισμα στην «ΠΟΛΗ»
ΕΔΡΕΥΕΙ	ΜΟΝΑΔΑ	ΠΟΛΗ	1-n	Γνώρισμα στην «ΜΟΝΑΔΑ»
ΒΡΙΣΚΕΤΑΙ	ΟΧΗΜΑ	ΘΕΣΗ	1-1	Γνώρισμα στην «ΘΕΣΗ»
ΑΙΤΕΙΤΑΙ	ΜΟΝΑΔΑ	ΑΙΤΗΣΗ	n-1	Γνώρισμα στην «ΑΙΤΗΣΗ»
ΛΑΜΒΑΝΕΤΑΙ	ΑΙΤΗΣΗ	ΑΠΟΦΑΣΗ	1-1	Γνώρισμα στην «ΑΙΤΗΣΗ»

Πίνακας 2: Μετατροπή Συσχετίσεων σε Σχέσεις ή Γνωρίσματα

Με βάση τα παραπάνω, δημιουργείται το σχεσιακό σχήμα της (Εικόνα 28), όπου τα κλειδιά κάθε σχέσης είναι έντονα γραμμένα και υπογραμμισμένα, ενώ τα ξένα κλειδιά ξεκινούν από την σχέση που αναφέρεται και καταλήγουν στην αναφερόμενη. Ειδικά για τα ξένα κλειδιά, στον Πίνακα 3 φαίνονται αναλυτικά πως έχουν χρησιμοποιηθεί. Πρέπει επίσης να διευκρινιστεί, ότι σε αντίθεση με το διάγραμμα Ο-Σ, στο οποίο χρησιμοποιήθηκαν ελληνικοί όροι στις οντότητες, τις συσχετίσεις και τα γνωρίσματα, στο σχεσιακό σχήμα της ΒΔ θα χρησιμοποιηθούν οι αντίστοιχοι αγγλικοί.

Σχέση που αναφέρεται	Γνώρισμα (Ξένο κλειδί)	Αναφερόμενη σχέση	Γνώρισμα Αναφερόμενης Σχέσης
Vehicle	Category	Category	Cat_code
Vehicle	Unit	Unit	Unit_code
Vehicle_transfer	Vehicle	Vehicle	Plate_no
Vehicle_transfer	Transfer	Transfer	Request_code
Transfer	Request_code	Request	Request_code
Transfer	Start_city	City	City_id
Transfer	end_city	City	City_id
Retire	Request_code	Request	Request_code
Maintenance	Request_code	Request	Request_code
Request	Unit	Unit	Unit_code
Unit	City	City	City_id
City	County	County	County_id
Rviacounty	Road	Road	Road_code
Rviacounty	County	County	County_id
Raccesscity	Road	Road	Road_code
Raccesscity	city	City	City_code
Located	Vehicle	Vehicle	Plate_no

Πίνακας 3: Ξένα Κλειδιά



Εικόνα 28:Σχεσιακό Σχήμα ΒΔ

5

Υλοποίηση

Μετά και την σχεδίαση της ΒΔ και το σχεσιακό σχήμα που προέκυψε μετά και τον εννοιολογικό σχεδιασμό, ακολουθεί η υλοποίηση της εφαρμογής. Το κεφάλαιο αυτό ασχολείται με ότι έχει να κάνει με την υλοποίηση και τις λεπτομέρειες αυτής.

5.1 Πλατφόρμες και προγραμματιστικά εργαλεία

Για την υλοποίηση της εφαρμογής, χρησιμοποιήθηκε, κυρίως λογισμικό ανοιχτού κώδικα, λόγω των πλεονεκτημάτων και της ευρείας χρήσης που έχει. Πιο συγκεκριμένα, πάνω σε λειτουργικό σύστημα Windows Vista, εγκαταστάθηκε ο web εξυπηρετητής Apache Tomcat και το ΣΔΒΔ MySQL. Επίσης εγκαταστάθηκε η γλώσσα PHP για να μπορεί η εφαρμογή που θα αναπτυχθεί να συναλλάσσεται με την ΒΔ. Τέλος η οπτικοποίηση των χωρικών δεδομένων πάνω σε χάρτη, έγινε με τη βοήθεια των google maps api.

Υλοποιήθηκε μια web εφαρμογή, στην οποία έχουν πρόσβαση οι χρήστες, ο καθένας ανάλογα με τα δικαιώματά του. Μέσω του ιστοτόπου γίνεται προσπέλαση της ΒΔ και ανακτώνται δεδομένα ή γίνεται ενημέρωση των υπαρχόντων, ανάλογα με την περίπτωση.

5.1.1 Προσαρμογή πλατφόρμας ανάπτυξης

Η πλατφόρμα πάνω στην οποία αναπτύχθηκε η εφαρμογή είναι το περιβάλλον WAMP (Windows – Apache – MySQL – PHP) με την διαφοροποίηση ότι αντί για τον Apache HTTP server, εγκαταστάθηκε ο Apache Tomcat, ώστε να είναι δυνατή η διαχείριση java servlets αν

απαιτηθεί παράλληλα με την php και τη MySQL. Η εγκατάσταση των συστατικών αυτών, παρουσιάζει ιδιαίτερο ενδιαφέρον και αρκετές ιδιαιτερότητες και για το λόγο αυτό αναλύεται διεξοδικά (επισημαίνεται ότι οι διαδρομές αρχείων που αναφέρονται, είναι ενδεικτικές και αφορούν στην εγκατάσταση που έγινε σε συγκεκριμένο υπολογιστή):

i. Αρχικά εγκαθιστούμε το Java Development Kit (JDK) και προσθέτουμε την διαδρομή στην μεταβλητή περιβάλλοντος PATH. Το JDK που εγκαταστάθηκε είναι το 1.6.0_06 και προστέθηκε στην μεταβλητή περιβάλλοντος PATH η διαδρομή: C:\Program Files\Java\jdk1.6.0_06\bin ή την αντίστοιχη του υπολογιστή που εργαζόμαστε.

ii. Στη συνέχεια εγκαθίσταται ο Tomcat. Η έκδοση που εγκαταστάθηκε είναι η 5.5.28. Μετά την εγκατάσταση, δημιουργούμε την μεταβλητή περιβάλλοντος JAVA_HOME, με διαδρομή την C:\Program Files\Java\jdk1.6.0_06. Αν η εγκατάσταση έχει γίνει σωστά όταν εκκινήσουμε την υπηρεσία και μεταβούμε μέσω ενός Internet Explorer στη διεύθυνση <http://localhost:8080>, θα δούμε την σελίδα καλωσορίσματος του Tomcat.

iii. Ελέγχουμε αν στο directory του Tomcat υπάρχει ο φάκελος catalina και μέσα σε αυτόν ο υποφάκελος localhost. Αν δεν υπάρχουν τους δημιουργούμε. Στην συνέχεια μέσα στο φάκελο localhost, δημιουργούμε ένα xml αρχείο, με περιεχόμενο: <Context debug="0" reloadable="true" privileged="true"/> και με όνομα το **ίδιο** όνομα με το οποίο στη συνέχεια θα δημιουργήσουμε φάκελο για την εφαρμογή μας. Στην περίπτωση μας, το αρχείο ονομάστηκε fmdss.xml.

iv. Με το **ίδιο** όνομα όπως με το xml που μόλις δημιουργήσαμε, δημιουργούμε έναν φάκελο στο φάκελο webapps του directory του Tomcat, δηλαδή fmdss, και μέσα σε αυτόν τον υποφάκελο WEB-INF. Στον τελευταίο, θα δημιουργηθούν οι φάκελοι classes (όπου θα αποθηκεύονται και θα μεταγλωττίζονται τα Java κομμάτια της εφαρμογής) και lib, όπου θα εισαχθούν κάποια αρχεία απαραίτητα για την διαλειτουργικότητα Tomcat με την PHP.

v. Για να δουλεύει ο Tomcat με jsp, προσθέτουμε στη μεταβλητή περιβάλλοντος CLASSPATH τη διαδρομή του αρχείου jsp-api.jar (C:\Tomcat 5.5\common\lib\jsp-api.jar).

vi. Όμοια, για να μπορεί να δουλεύει με servlet, προσθέτουμε το servlet-api.jar (C:\Tomcat 5.5\common\lib\servlet-api.jar).

vii. Τα servlet τρέχουν είτε από το ROOT directory, είτε από το φάκελο της εφαρμογής με τη χρήση του invoker, ο οποίος όμως δεν είναι εξ ορισμού ενεργοποιημένος. Για το λόγο αυτό, στο φάκελο C:\Tomcat 5.5\conf τροποποιούμε το αρχείο web.xml ώστε να χρησιμοποιεί servlet, αφαιρώντας τα σχόλια από τα tag servlet και servlet-mappings που αφορούν τον invoker, και στη συνέχεια το αρχείο context.xml, προσθέτοντας στο context tag: reloadable="true" privileged="true".

viii. Στο υποφάκελο WEB-INF του φακέλου της εφαρμογής, δημιουργούμε ένα xml αρχείο, με όνομα web.xml, στο οποίο για κάθε servlet που χρησιμοποιούμε προσθέτουμε και τα αντίστοιχα tags.

ix. Εγκαθιστούμε την PHP μαζί με τις αντίστοιχες επεκτάσεις PECL. Στην εφαρμογή χρησιμοποιήθηκε η έκδοση 5.2.5. Μετά την εγκατάσταση προσθέτουμε στην μεταβλητή περιβάλλοντος PATH, τις διαδρομές τις PHP: C:\Program Files\php-5.2.5-Win32 και C:\Program Files\php-5.2.5-Win32\ext.

x. Στο φάκελο της PHP, μετονομάζουμε το αρχείο php.ini-dist σε php.ini και στη συνέχεια το τροποποιούμε αφαιρώντας τα σχόλια από την ενότητα των επεκτάσεων για το αρχείο php_java.dll. (αν δεν υπάρχει αυτή επέκταση, απλά προσθέτουμε extension=php_java.dll). Αυτό γίνεται για να μπορεί ο Tomcat, που έχει σχεδιαστεί για servlet, να χειρίζεται αιτήματα php. Τέλος προσθέτουμε στον φάκελο C:\Program Files\php-5.2.5-Win32 το αρχείο php5servlet.dll από τα αντίστοιχα PECL.

xi. Τροποποιούμε το web.xml που δημιουργήσαμε στο βήμα vii, προσθέτοντας στα αντίστοιχα σημεία τα παρακάτω servlet tags:

```
<servlet>
  <servlet-name>php</servlet-name>
  <servlet-class>net.php.servlet</servlet-class>
</servlet>
<servlet>
  <servlet-name>php-formatter</servlet-name>
  <servlet-class>net.php.formatter</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>php</servlet-name>
  <url-pattern>*.php</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>php-formatter</servlet-name>
  <url-pattern>*.phps</url-pattern>
</servlet-mapping>
```

xii. Από τα PECL, βρίσκουμε το αρχείο phpsrvlt.jar και το αποσυμπιέζουμε μέσω της γραμμής εντολών, με την εντολή: jar -xf phpsrvlt.jar. Δημιουργούνται δύο φάκελοι: META-INF και net. Στη διαδρομή net\php\ τροποποιούμε τα αρχεία reflect.properties και servlet.properties, ώστε να γράφουν php5servlet και στη συνέχεια δημιουργούμε ένα αρχείο jar (μέσω της εντολής: jar cvf php5servlet.jar net\php*.*) και με όνομα php5servlet.jar το οποίο τοποθετούμε στο φάκελο lib του φακέλου της εφαρμογής μας.

xiii. Εγκαθιστούμε την MySQL (έκδοση 5.1) και προσθέτουμε τη διαδρομή της, στην μεταβλητή PATH (C:\Program Files\MySQL\MySQL Server 5.1\bin). Επίσης εγκαθιστούμε το MySQL Query Browser, για μια πιο φιλική διεπαφή του ΣΔΒΔ με τον χρήστη.

xiv. Δημιουργούμε την μεταβλητή περιβάλλοντος PHPRC με τιμή C:\Program Files\php-5.2.5-Win32 και όμοια με το βήμα x από το αρχείο php.ini, αφαιρούμε τα σχόλια από τις

extensions php_mysql.dll και php_mysql_i.dll. Με τις ενέργειες αυτές, η MySQL μπορεί να συνεργάζεται και με την Php και με τον Tomcat.

xv. Τέλος, για να συνεργάζεται η MySQL με servlets και Tomcat, κάνουμε χρήση του JDBC driver, (mysql-connector-java-5.1.5) και τοποθετούμε το αρχείο mysql-connector-java-5.1.5-bin.jar στο φάκελο lib του φακέλου της εφαρμογής μας. Τροποποιούμε το αρχείο web.xml που δημιουργήσαμε προσθέτοντας τα tags:

```
<servlet>
  <servlet-name>JavaWmysql</servlet-name>
  <servlet-class>JavaWmysql</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>JavaWmysql</servlet-name>
  <url-pattern>/JavaWmysql</url-pattern>
</servlet-mapping>
```

Και η εγκατάσταση ολοκληρώθηκε. Από το σημείο αυτό και μετά, τοποθετούμε τις ιστοσελίδες μέσα στον φάκελο της εφαρμογής και μέσω ενός φυλλομετρητή ιστού, έχουμε πρόσβαση σε αυτές.

Η γλώσσα PHP μιλά με τη ΒΔ μέσω της διαδικασίας υποβολής ερωτημάτων. Η όλη επικοινωνία στηρίζεται στα παρακάτω βήματα:

- i. Συνδεόμαστε με τον εξυπηρετητή χωρίς την ανάγκη χρήσης username και password. Στην πραγματική χρήση της εφαρμογής, αντί του localhost, θα πρέπει να εισαχθεί η διεύθυνση του server. Επίσης θα χρειάζεται και χρήση username και password.
- ii. Συνδεόμαστε με τη ΒΔ, χρησιμοποιώντας το όνομα του σχήματος το οποίο θα προσπελάσουμε.
- iii. Υποβάλλουμε το ερώτημά μας (ερώτημα, ενημέρωση, εισαγωγή, διαγραφή).
- iv. Παίρνουμε το αποτέλεσμα το οποίο αν επιθυμούμε το εκτυπώνουμε αφού όμως πρώτα το αναθέσουμε σε μια μεταβλητή.
- v. Αποσυνδεόμαστε από τη ΒΔ.

Οι ιστοσελίδες μπορούν να δημιουργηθούν με διάφορα εργαλεία σχεδιασμού ιστοσελίδων. Στην εφαρμογή χρησιμοποιήθηκε το freeware εργαλείο ACEHtml.

5.1.2 Χρήση των Google Maps σε συνδυασμό με MySQL/PHP.

Απαραίτητη προϋπόθεση για να μπορέσει να γίνει χρήση των Google Maps, είναι να διαθέτουμε το google maps api key, το οποίο διατίθεται δωρεάν. Αρκεί να διαθέτουμε ένα λογαριασμό google, δηλαδή λογαριασμό ηλεκτρονικού ταχυδρομείου gmail. Στην ιστοσελίδα <http://code.google.com/intl/el/apis/maps/signup.html>, με αποδοχή των όρων χρήσης και δίνοντας το όνομα της ιστοσελίδας στην οποία θα χρησιμοποιηθούν οι χάρτες, χορηγείται από την υπηρεσία ένα μοναδικό κλειδί με τη χρήση του οποίου και εφόσον υπάρχει πρόσβαση στο διαδίκτυο, οι χάρτες μπορούν να χρησιμοποιηθούν στην ιστοσελίδα για την

οποία έχει δηλωθεί. Στην περίπτωσή μας, τέθηκε σαν ιστοτόπος χρήσης των google maps, το <http://localhost:8080>, οπότε ότι τοποθετείται κάτω από αυτό, μπορεί να χρησιμοποιεί google maps.

Οι χάρτες αποτελούνται από 3 βασικά στοιχεία:

- i. Τη διαδρομή ενός javascript της google
- ii. Κώδικα javascript που περιέχει τις διάφορες παραμέτρους του χάρτη μας,
- iii. Τον ορισμό του σημείου στη σελίδα που θα εμφανίζεται ο χάρτης.

Από τα 3 αυτά στοιχεία, τα δύο πρώτα τοποθετούνται στο header της ιστοσελίδας, ενώ το τρίτο στο κυρίως μέρος της (body).

```
<html>
  <script charset="UTF-8"
src="http://maps.google.com/maps?file=api&v=2&hl=el&oe=utf-
&key=ABQIAAAABggkywTW10zDMHK9C7c-nRTwM0brOpm-
All5BF6PoaKBxRWWERQKcZBf19AwaudyvafHjrTDBylz0Q&sensor=false"
type="text/javascript"></script>
  <script type="text/javascript">
    function initialize() {
      if (GBrowserIsCompatible()) {
        var map = new GMap2(document.getElementById("map_canvas"));
        map.setCenter(new GLatLng(37.5119, 22.3773), 7);
        map.setUIToDefault(); } }
    </script>
  <body onload="initialize()" onunload="GUnload()">
    <div id="map_canvas" style="width: 500px; height: 300px"></div>
</body>
</html>
```

Αν παρατηρήσουμε στο παραπάνω κομμάτι κώδικα, καθορίζουμε και το κέντρο της περιοχής του χάρτη, τον οποίο θα εμφανίσουμε στη σελίδα μας, καθώς επίσης και το επίπεδο ζουμ (`map.setCenter(new GLatLng(37.5119, 22.3773), 7);`). Εδώ έχουν περαστεί οι συντεταγμένες της πόλης της Τριπόλεως με επίπεδο ζουμ 7.

Δίνονται επίσης επιπλέον δυνατότητες οπτικοποίησης κάποιων στοιχείων στο χάρτη, με πολύ απλό τρόπο. Αν θέλουμε να δείξουμε ένα σημείο στο χάρτη μας, αρκεί να προσθέσουμε ένα marker με τις συντεταγμένες του σημείου αυτού ή ακόμα και να μας βγάζει και κάποιο μήνυμα (μόνιμα ή με κλικ πάνω του). Για παράδειγμα αν στο παραπάνω κομμάτι κώδικα προσθέσουμε τις παρακάτω γραμμές αμέσως μετά τον ορισμό του κέντρου του χάρτη, προκύπτει το αποτέλεσμα της Εικόνας 29, όπου έχουμε τον χάρτη κεντραρισμένο στη Τρίπολη, και με σημάδι στο Ναύπλιο με την πληροφορία: Πρώτη Πρωτεύουσα της Ελλάδας.

```

var baseIcon = new GIcon(G_DEFAULT_ICON);
baseIcon.shadow = "http://www.google.com/mapfiles/shadow50.png";
baseIcon.iconSize = new GSize(20, 34);
baseIcon.shadowSize = new GSize(37, 34);
baseIcon.iconAnchor = new GPoint(9, 34);
baseIcon.infoWindowAnchor = new GPoint(9, 2);
function createMarker(point, index) {
    var redIcon = new GIcon(baseIcon);
    redIcon.image = "http://www.google.com/mapfiles/marker.png";
    markerOptions = { icon:redIcon };
    var marker = new GMarker(point, markerOptions);
    GEvent.addListener(marker, "click", function() {
        marker.openInfoWindowHtml("First Capital Of Greece");
    });
    return marker; }
    var latlng = new GLatLng(37.569, 22.810);
    map.addOverlay(createMarker(latlng));

```



Εικόνα 29: Στοιχειώδης χάρτης Google Map

Με το παραπάνω παράδειγμα, δείξαμε στο χάρτη μας, ένα σημείο με κάποιες πληροφορίες. Το σημείο αυτό και τα στοιχεία του, τα περνάμε εμείς ρητά, μέσα στον κώδικα της ιστοσελίδας. Αν θέλουμε περισσότερα του ενός σημεία, πρέπει να χρησιμοποιήσουμε ένα αρχείο xml, το οποίο να περιλαμβάνει τα σημεία αυτά. Το αρχείο αυτό περιλαμβάνει μέσα σε tags με όνομα marker τα στοιχεία που θέλουμε να απεικονιστούν το χάρτη. Οι πληροφορίες που θέλουμε να υπάρχουν είναι οπωσδήποτε τα lat, long και ό,τι άλλο επιθυμούμε να εμφανίζεται στο μπαλόνι πληροφοριών. Παράδειγμα τέτοιου tag φαίνεται παρακάτω, όπου έχουμε τρεις τοποθεσίες, με τις συντεταγμένες τους και πρόσθετες πληροφορίες παρόμοιας μορφής με το Ναύπλιο, που δείξαμε πιο πάνω:

```

<markers>
  <marker desc="First Capital of Greece" lat="37.569" lng="22.810"/>
  <marker desc="Capital of Lakonia" lat="37.077" lng="22.431"/>
  <marker desc="Capital of Peloponese" lat="37.511" lng="22.377"/>
</markers>

```

Κάνοντας τις κατάλληλες τροποποιήσεις ο κώδικας της ιστοσελίδας μετατρέπεται ανάλογα και τελικά προκύπτει το αποτέλεσμα της Εικόνα 30.

Στην περίπτωση αυτή, το απαιτούμενο αρχείο xml καλείται αλλά έχει δημιουργηθεί στατικά, τα δεδομένα του δεν αποτελούν αποτέλεσμα ερώτησης προ μια ΒΔ.



Εικόνα 30: Χάρτης με περισσότερα του ενός σημεία και κλικ σε ένα από αυτά

Σκοπός μας είναι να προσαρμοστούν έτσι οι ιστοσελίδες μας, ώστε τα σημεία να προέρχονται δυναμικά, μετά από κλήσεις της Βάσης Δεδομένων.

Για να μπορέσουμε να χρησιμοποιήσουμε τη MySQL μαζί με τους Google Maps, θα πρέπει οι πίνακες να έχουν γνωρίσματα που να αντιστοιχούν στις γεωγραφικές συντεταγμένες. (lat – lng). Στη ΒΔ που έχει σχεδιαστεί όλα τα γεωγραφικά σημεία είναι της μορφής: POINT(22.377342825086 37.511920392079), οπότε πρέπει να βρεθεί ένας τρόπος να βγάλουμε από την παραπάνω συμβολοσειρά, τους χαρακτήρες εκείνους που θα μας δώσουν τις συντεταγμένες στη μορφή που θέλουμε. Για να το γεωγραφικό μήκος (22.377342825086), τα πράγματα είναι απλά: Ζητάμε μέσω της συνάρτησης της PHP substr να μας δώσει από τον έβδομο χαρακτήρα μέχρι και τον δέκατο τρίτο (substr (6,6)), οπότε προκύπτει το αποτέλεσμα που θέλουμε(22.377), το οποίο και μπορούμε να απεικονίσουμε. Για το γεωγραφικό πλάτος (37.511920392079), επειδή πρέπει να βρούμε τον ακριβές χαρακτήρα από τον οποίο ξεκινά, ακολουθούμε την παρακάτω διαδικασία: Το γεωγραφικό πλάτος ξεκινά αμέσως μετά τον κενό χαρακτήρα που ακολουθεί τις πλήρεις συντεταγμένες του γεωγραφικού μήκους. Την

θέση του τη βρίσκουμε μέσω της συνάρτησης `strpos('συμβολοσειρά αναζήτησης', ' ')`, και στη συνέχεια ζητάμε από τον επόμενο χαρακτήρα μέχρι και για έξι ακόμα. Οπότε προκύπτει το αποτέλεσμα που θέλουμε. Παράδειγμα κώδικα για τα παραπάνω, ακολουθεί:

```
$lng=substr ($cn, 6, 6);  
$lp=strpos ($cn, ' ');  
$lat=substr ($cn, $lp+1, 6);
```

Όπου `$cn` η συμβολοσειρά που αποτελεί τις συντεταγμένες του σημείου με τη μορφή που αναφέρθηκε πιο πάνω.

Για να εμφανίσουμε στοιχεία στο χάρτη, που προέρχονται από τη ΒΔ, αρκεί κάθε φορά να δημιουργείται δυναμικά xml αρχείο όπως και παραπάνω, με τα στοιχεία που θέλουμε κάθε φορά, και στη συνέχεια να κληθεί το αρχείο αυτό, ώστε να απεικονιστεί πάνω στο χάρτη. Σα διαδικασία το παραπάνω έχει κάποιες δυσκολίες, μιας και τα δεδομένα που θέλουμε να απεικονίζονται κάθε φορά αλλάζουν και δεν είναι σταθερά.

Στην περίπτωσή μας ακολουθήθηκε εναλλακτική μέθοδος, η οποία στην ουσία δημιουργεί και χρησιμοποιεί άμεσα το xml αρχείο, χωρίς να χρειαστεί να το καλέσει.

Ορίζεται ο χάρτης και ορίζεται συνάρτηση τόσο για την τοποθέτηση των δεικτών όσο και για την εμφάνιση του παραθύρου με τις διάφορες πληροφορίες, που εμείς επιθυμούμε. Στη συνέχεια έχουμε το block της PHP, που αφού πάρει τις κατάλληλες παραμέτρους από προηγούμενη σελίδα και συνδεθεί με τη ΒΔ, τοποθετεί το αποτελέσματα σε array και μέσω της πιο πάνω συνάρτησης δημιουργίας δεικτών, τα απεικονίζει στο χάρτη.

Στο παράδειγμα που ακολουθεί, ζητάμε να μας δοθεί η θέση όλων των οχημάτων της μονάδας με κωδικό 7. Αρχικά παρατίθεται ο κώδικας παραγωγής της ιστοσελίδας, στη συνέχεια το αποτέλεσμα όπως φαίνεται στην οθόνη (Εικόνα 31) και στο τέλος μέσω της προβολής του αρχείου προέλευσης της ιστοσελίδας, τι έχει παραχθεί για να δούμε το αποτέλεσμα που βλέπουμε στην οθόνη (θα παρατεθούν μόνο τα αποτελέσματα της επερώτησης και όχι ο υπόλοιπος κώδικας).

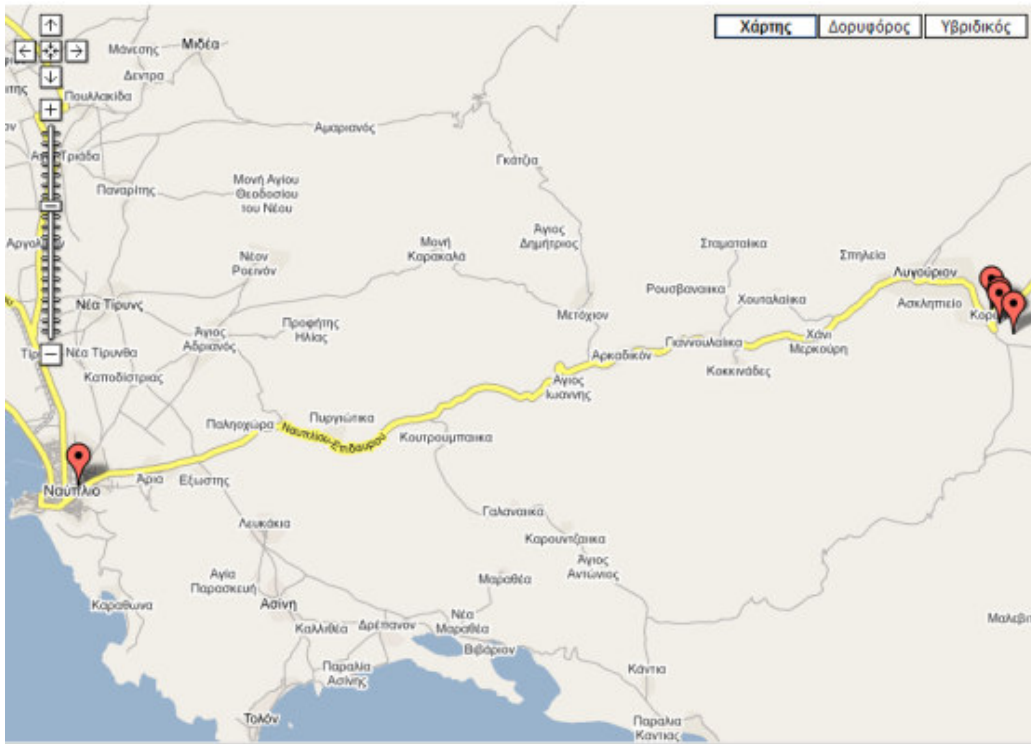
```
<html><head>  
  
<script  
src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAABgqkywTW10zDMHK9C7c-nRTwM0brOpm-A115BF6PoaKBxRWWERQKcZBf19AwauydvafHjrTDBylz0Q&sensor=false"  
type="text/javascript"></script>  
</head><body>  
  
<div id="map" style="width: 800px; height: 600px"></div>  
<script type="text/javascript">  
  
//<![CDATA[  
  
var map = new GMap2 (document.getElementById ("map"));  
map.addControl (new GLargeMapControl ());  
map.addControl (new GMapTypeControl ());
```



```

map.addControl(new GScaleControl());
map.setCenter(new GLatLng(37.5119, 22.3773),7);
// Creates a marker whose info window displays the given number
function createMarker(point, number)
{var marker = new GMarker(point);
// Show this markers index in the info window when it is clicked
var html =number;
GEvent.addListener(marker, "click", function()
{marker.openInfoWindowHtml(html)});
return marker;
};
<?php
$link = mysql_connect("localhost", "root", "") or die("Could not connect: "
. mysql_error());
mysql_selectdb("fmdss",$link) or die ("Can\'t use dbmapserver : " .
mysql_error());
$result = mysql_query("select located.vehicle,lat,lng
from vehicle,located,unit
where vehicle.plate_no=located.vehicle
and vehicle.unit=unit.unit_code
and unit.unit_code = '7';",$link);
if (!$result)
{
    echo "no results ";}
while($row = mysql_fetch_array($result))
{
    $lat=$row['lat'];
    $lng=$row['lng'];
echo "var point = new GLatLng($lat,$lng);\n";
echo "var marker = createMarker(point, '" . addslashes($row['vehicle']) .
"';)\n";
echo "map.addOverlay(marker);\n";
echo "\n";}
mysql_close($link);
?>//]]></script></body></html>

```



Εικόνα 31: Εμφάνιση οχημάτων συγκεκριμένης μονάδας και μεγέθυνση σε περιοχή

```

var point = new GLatLng(37.6030,23.0550);
var marker = createMarker(point, 'YMZ4777');
map.addOverlay(marker);

var point = new GLatLng(37.6060,23.0530);
var marker = createMarker(point, 'YMZ4778');
map.addOverlay(marker);

var point = new GLatLng(37.9150,22.8780);
var marker = createMarker(point, 'YMZ4779');
map.addOverlay(marker);

var point = new GLatLng(37.9180,22.8860);
var marker = createMarker(point, 'YMZ4780');
map.addOverlay(marker);

var point = new GLatLng(37.6040,23.0550);
var marker = createMarker(point, 'YMZ4782');
map.addOverlay(marker);

var point = new GLatLng(37.5690,22.8100);
var marker = createMarker(point, 'YMZ4783');
map.addOverlay(marker);

var point = new GLatLng(38.2520,21.7350);
var marker = createMarker(point, 'YMZ4784');
map.addOverlay(marker);

var point = new GLatLng(37.6010,23.0590);

```

```
var marker = createMarker(point, 'YMZ4785');
map.addOverlay(marker);
```

Με την μεθοδολογία αυτή, λοιπόν, θα μπορούμε να εμφανίζουμε τα όποια αποτελέσματα επιθυμούμε στην οθόνη μας.

5.1.3 Υλοποίηση Βάσης Δεδομένων

Για τη δημιουργία της ΒΔ, δημιουργούμε αρχικά το σχήμα της βάσης και στη συνέχεια δημιουργούμε απλούς σχεσιακούς πίνακες. Οι πίνακες τροποποιούνται στη συνέχεια με την προσθήκη των γεωγραφικών πεδίων, όπου χρειάζεται καθώς και με την προσθήκη πρωτεύοντων κλειδιών. Τα πρωτεύοντα κλειδιά (αν αυτά απαρτίζονται από ένα γνώρισμα), θα μπορούσαν να οριστούν μαζί με τη σχέση. Προτιμάται όμως να ορίζονται με εντολή τροποποίησης της σχέσης, ώστε να πάρουν όνομα, ώστε να υπάρχει η δυνατότητα μελλοντικής διαγραφής των. Επόμενο βήμα είναι να φορτωθούν με τα αρχικά δεδομένα (για να μην έχει σημασία η σειρά με την οποία δημιουργούνται και γεμίζουν οι πίνακες) και τέλος ορίζουμε τους περιορισμούς αναφορικής ακεραιότητας.

Στη συνέχεια παρατίθενται οι εντολές create, με τον κατάλληλο σχολιασμό, όπου χρειάζεται, σχετικά με τον τύπο δεδομένων που χρησιμοποιήθηκε για κάθε γνώρισμα. Οι εντολές αυτές, υπάρχουν και στο συνοδευτικό αρχείο fmdss.sql, που θα χρησιμοποιηθεί για την δημιουργία της βάσης:

i. Δημιουργία Σχήματος

- Create schema fmdss

ii. Χρήση σχήματος

- use fmdss;

iii. Δημιουργία σχέσης category

- create table category (
 cat_code int,
 cat_desc varchar(20) not null);

• alter table category add constraint category_PK primary key (cat_code);

iv. Δημιουργία σχέσης vehicle

- create table vehicle (
 plate_no char(7),
 brand varchar(20) not null,
 klm double(19,4) not null default '0',
 purchase_date date not null,
 unit int not null,

```

date_in_unit    date not null,
category       int not null,
availability    boolean,
used_as        set ("Training","Transports", "Administrative"));

```

Η χρήση του τύπου δεδομένων set προτιμάται διότι υποχρεώνει το χρήστη να χρησιμοποιήσει ότι υπάρχει μέσα στο σύνολο. Έτσι αποφεύγονται ορθογραφικά λάθη που θα δημιουργούσαν διαφορετικά αποτελέσματα (πχ Trainning αντί Training δημιουργεί μια ακόμα περίπτωση χρήσης οχήματος) . Ο χρήστης έχει τη δυνατότητα να ορίζει 2 ή και περισσότερα αντικείμενα από το σύνολο (πχ Training και Administrative) κάτι που επιδιώκουμε και μας εξοικονομεί μια παραπάνω σχέση. Επίσης επιλέγεται μιας και η MySQL δεν υποστηρίζει τη δημιουργία πεδίου ορισμού (domain).

Το γνώρισμα klm, δηλαδή τα συνολικά χιλιόμετρα που έχει διανύσει το όχημα, είναι ένα παραγόμενο γνώρισμα, που θα προκύπτει από την άθροιση όλων των αποστάσεων των κινήσεων, που έχει μετάσχει το όχημα και θα υπολογιστεί στη συνέχεια, μέσω αποθηκευμένης διαδικασίας (stored procedure). Όμοια και το γνώρισμα availability, που μέσω διαδικασίας θα ελέγχει αν το όχημα είναι διαθέσιμο. (Διαθέσιμο είναι ένα όχημα αν δεν βρίσκεται σε κίνηση, αν δεν υποβάλλεται σε συντήρηση και αν δεν έχει αποσυρθεί).

v. Δημιουργία σχέσης located

- create table located (

```

vehicle        char(7),
instance       timestamp
lat            double(4,4)
long           double(4,4));

```

- alter table located add constraint located_PK primary key (vehicle);

Στη συνέχεια προσθέτουμε το γεωγραφικό πεδίο, που μας προσδιορίζει την τρέχουσα θέση του οχήματος.

- alter table located add position point;

vi. Δημιουργία σχέσης vehicle_transfer

- create table vehicle_transfer(

```

vehicle        char(7) not null,
transfer       int not null);

```

Στη σχέση αυτή, πρωτεύων κλειδί είναι ο συνδυασμός των γνωρισμάτων, επομένως ως σύνθετο πρωτεύων κλειδί, θα οριστεί με εντολή τροποποίησης της σχέσης:

- alter table vehicle_transfer add constraint v_t_PK primary key (vehicle,transfer);

vii. Δημιουργία σχέσης transfer

- create table transfer (
 request_code int,
 transfer_date date not null,
 no_veh int,
 start_city int,
 end_city int,
 distance double(19,4),
 t_condition set("In Progress","Accomplished"));
- alter table transfer add constraint transfer_PK primary key (request_code);

Όμοια με τη σχέση vehicle, έτσι και εδώ το γνώρισμα που αφορά στον συνολικό αριθμό οχημάτων που μετέχουν στην κίνηση (no_veh), και στη συνολική απόσταση της κίνησης (distance) θα υπολογίζονται και θα αποθηκεύονται μέσω διαδικασιών ή όψεων, που θα δημιουργηθούν μετά την υλοποίηση της βάσης και τη φόρτωσή της με δεδομένα. Επίσης τα γνωρίσματα πόλης έναρξης και λήξης κίνησης, θα χρησιμοποιούνται, εκτός των άλλων, και για τον υπολογισμό της απόστασης που διανύει το όχημα, με το μειονέκτημα ότι η απόσταση αυτή είναι η απόσταση σε ευθεία γραμμή.

viii. Δημιουργία σχέσης retire

- create table retire (
 request_code int,
 retire_date date,
 cause set ("Severe Damages","Old Vehicle"),
 vehicle char(7) not null);
- alter table retire add constraint retire_PK primary key (request_code);

ix. Δημιουργία σχέσης maintenance

- create table maintenance (
 request_code int,
 start_date date,
 end_date date,
 cost int,
 cause set ("Programmed", "Sudden", "Accident"),
 vehicle char(7));

- alter table maintenance add constraint maintenance_PK primary key (request_code);

x. Δημιουργία σχέσης request

- create table request (


```

          request_code int auto inc,
          request_date date not null,
          unit int not null,
          decision boolean,
          decision_date date,
          request_category set("Retire","Maintenance","Transfer"));
      
```
- alter table request add constraint request_PK primary key (request_code);

xi. Δημιουργία σχέσης city

- create table city (


```

          city_id int not null,
          city_name varchar(30),
          county int not null);
      
```
- alter table city add constraint city_PK primary key (city_id);
- alter table city add location point;

Στη σχέση αυτή, το γνώρισμα county που είναι ξένο κλειδί στη σχέση county, είναι πραγματικά περιττό. Μια πόλη ανήκει στο νομό στην επιφάνεια του οποίου βρίσκεται. Λόγω όμως της περίπτωσης των ΕΠΠ, μπορεί μια πόλη να επιστρέφεται ότι ανήκει σε δύο ή και σε περισσότερους νομούς. Για το λόγο αυτό, προσθέτουμε και το ξένο κλειδί αυτό, με αποτέλεσμα να έχουμε σίγουρα ορθά και συνεπή αποτελέσματα.

xii. Δημιουργία σχέσης county

- create table county (


```

          county_id int not null,
          county_name varchar(15));
      
```
- alter table county add constraint county_PK primary key (county_id);
- alter table county add shape polygon;

xiii. Δημιουργία σχέσης road

- Create table road (


```

          road_code int,
          road_name varchar(60) ,
      
```

```
length double(19,4);
```

- alter table road add constraint road_PK primary key (road_code);
- alter table road add way linestring;

xiv. Δημιουργία σχέσης unit

- create table unit (

```
unit_code int,  
unit_name varchar(40),  
city int not null,  
category set("Factory","HQ","Sales Place","Service"));
```
- alter table unit add constraint unit_PK primary key (unit_code);

xv. Δημιουργία σχέσης rviacounty

- create table rviacounty (

```
road int,  
county int);
```
- alter table rviacounty add constraint rvc_PK primary key (road,county);

xvi. Δημιουργία σχέσης raccesscity

- create table raccesscity (

```
road int,  
city int);
```
- alter table raccesscity add constraint rac_PK primary key (road,city);

Αφού υλοποιήθηκε η ΒΔ μέσω των αρχικών εντολών create και alter, επόμενο στάδιο είναι να γεμίσει με τα αρχικά δεδομένα, σύμφωνα με το σενάριο που θα αναλυθεί στο επόμενο κεφάλαιο. Ο μόνος λόγος για τον οποίο γίνεται αυτό πριν τον ορισμό των περιορισμών αναφορικής ακεραιότητας, είναι για να μην έχει σημασία η σειρά με την οποία γεμίζουν οι πίνακες δεδομένα. Στη συνέχεια παρατίθενται οι εντολές ορισμού περιορισμών αναφορικής ακεραιότητας, χωρίς ιδιαίτερο σχολιασμό, μιας και έχουν αναφερθεί στον Πίνακα 3(Οι εντολές βρίσκονται και στο αρχείο integrity constraints.sql):

- alter table vehicle add constraint veh_cat_FK foreign key (category) references category (cat_code);
- alter table vehicle add constraint veh_unit_FK foreign key (unit) references unit (unit_code);
- alter table located add constraint loc_veh_FK foreign key (vehicle) references vehicle (plate_no);

- alter table vehicle_transfer add constraint veh_trans_veh_FK foreign key (vehicle) references vehicle (plate_no);
- alter table vehicle_transfer add constraint veh_trans_trans_FK foreign key (transfer) references transfer(request_code);
- alter table transfer add constraint trans_req_FK foreign key (request_code) references request (request_code);
- alter table transfer add constraint trans_citys_FK foreign key (start_city) references city (city_id);
- alter table transfer add constraint trans_citye_FK foreign key (end_city) references city (city_id);
- alter table retire add constraint ret_req_FK foreign key (request_code) references request (request_code);
- alter table maintenance add constraint main_req_FK foreign key (request_code) references request (request_code);
- alter table request add constraint req_unit_FK foreign key (unit) references unit (unit_code);
- alter table unit add constraint unit_city_FK foreign key (city) references city (city_id);
- alter table city add constraint cit_county_FK foreign key (county) references county (county_id);
- alter table rviacounty add constraint rvc_county_FK foreign key (county) references county (county_id);
- alter table rviacounty add constraint rvc_road_FK foreign key (road) references road (road_code);
- alter table raccesscity add constraint rac_city_FK foreign key (city) references city (city_id);
- alter table raccesscity add constraint rac_road_FK foreign key (road) references road (road_code);

Η εισαγωγή των στοιχείων γίνεται αμέσως μετά την δημιουργία της βάσης. Όταν δημιουργείται η βάση, η κατάστασή της είναι η κενή, η κατάσταση χωρίς δεδομένα. Με την πρώτη εισαγωγή δεδομένων μεταφερόμαστε στην αρχική κατάσταση. Από εκεί και μετά κάθε πράξη εισαγωγής ή τροποποίησης μας οδηγεί σε μια νέα κατάσταση της βάσης δεδομένων, η οποία πρέπει να είναι έγκυρη, δηλαδή να είναι μια κατάσταση που να υπακούει τους περιορισμούς που έχουν τεθεί. Είναι ευθύνη του ΣΔΒΔ να εξασφαλίσει ότι η νέα κατάσταση θα είναι έγκυρη με τους μηχανισμούς που διαθέτει. Έτσι σε περίπτωση που μια εισαγωγή ή ενημέρωση παραβιάζει κάποιον περιορισμό, το ΣΔΒΔ μπορεί είτε να προσπαθήσει μόνο του

να διορθώσει το σφάλμα, είτε να αρνηθεί την τροποποίηση ενημερώνοντας και εξηγώντας το λόγο για τον οποίο συνέβη αυτό.

Η αρχική κατάσταση της ΒΔ προκύπτει με την εισαγωγή των δεδομένων. Τα στοιχεία που έχουν εισαχθεί είναι ενδεικτικά για να δουλέψει η εφαρμογή και σε καμία περίπτωση δεν ανταποκρίνονται στο πλήθος των στοιχείων που θα είχε η εφαρμογή σε πραγματική χρήση.

Στην εφαρμογή έχουν γίνει κάποιες παραδοχές κατά την εισαγωγή των αρχικών δεδομένων. Αναλυτικά:

- i. Όλα τα οχήματα βρίσκονται σε τυχαίες θέσεις μέσα στο οδικό δίκτυο.
- ii. Τα οχήματα όταν κινούνται επιστρέφουν στη βάση τους. Δηλαδή για μια κίνηση Πάτρα-Αίγιο, θα πρέπει να υπάρχει και η επιστροφή Αίγιο-Πάτρα. Στα αρχικά δεδομένα αυτό δεν έχει εφαρμοστεί, αλλά θα γίνει σε κάθε επόμενη εισαγωγή κίνησης.
- iii. Όταν ένα όχημα πηγαίνει για συντήρηση στην Αρεόπολη, και μπορεί να κινηθεί (δηλαδή δεν το μεταφέρει γερανός), θα πρέπει να υπάρχει κίνηση από το σημείο μετάβασης στην Αρεόπολη και επιστροφή στο αρχικό σημείο. Ούτε τέτοιες κινήσεις έχουν καταγραφεί στην αρχική κατάσταση της ΒΔ, αλλά καταγράφονται σε μετέπειτα καταστάσεις που προκύπτουν.

Τέλος, πρέπει να σημειωθεί, ότι όπου χρειάζεται, γίνεται χρήση των όψεων, για την εκτέλεση των διαφόρων λειτουργιών. Αυτό διότι οι όψεις ,ως εικονικοί πίνακες, είναι πάντα ενημερωμένες μιας και δημιουργούνται από τους πίνακες τους οποίους απαρτίζεται κάθε φορά που θέλουμε να αντλήσουμε δεδομένα από αυτή.

6

Έλεγχος

Το κεφάλαιο που ακολουθεί, πραγματεύεται τον έλεγχο που έγινε για την λειτουργία και αξιοπιστία της εφαρμογής. Με τη χρήση ενός σεναρίου, ελέγχουμε τις λειτουργίες σε διάφορες περιπτώσεις που μπορούν να εμφανιστούν.

6.1 Μεθοδολογία ελέγχου

Στη συνέχεια θα αναλυθεί το σενάριο λειτουργίας το οποίο ακολουθήθηκε. Τα χωρικά δεδομένα για το σενάριο αυτό, αφορούν την περιοχή της Πελοποννήσου.

Στην Πελοπόννησο δραστηριοποιείται μια εταιρία παραγωγής φρέσκου γάλακτος με την επωνυμία «ΕΨΙΛΟΝ». Η εταιρία αυτή έχει τα κεντρικά της γραφεία στην πόλη της Τρίπολης, ενώ διαθέτει και 3 εργοστάσια παραγωγής (Κυπαρισσία, Ξυλόκαστρο, Νάυπλιο). Γραφεία, εκτός από τα Κεντρικά στην Τρίπολη, υπάρχουν και στο Αίγιο, το Άργος και την Σπάρτη. Τέλος υπάρχουν και τα κεντρικά σημεία πώλησης, στην Ολυμπία, το Γύθειο, τη Μεγαλόπολη, την Καλαμάτα και το Κρανίδι. Επίσης στην Αρέοπολη, υπάρχει ο συμβεβλημένος σταθμός συντήρησης, όπου εκτελούνται όλες οι εργασίες συντήρησης των οχημάτων.

Η πολιτική της εταιρίας είναι να μεταφέρει το γάλα, από τους σταθμούς παραγωγής, στους σταθμούς διανομής. Από το σημείο αυτό και μετά έρχονται οι ενδιάμεσοι διανομείς και το παραλαμβάνουν για να το μεταφέρουν στους τελικούς πωλητές. Το τι γίνεται από τα κεντρικά σημεία πώλησης και μετά, δεν θα απασχολήσει την εφαρμογή.

Τέλος, υπάρχουν και οι πωλητές, οι οποίοι μεταβαίνουν στις διάφορες πόλεις για τις οποίες είναι υπεύθυνοι και έρχονται σε επαφή με καταστήματα λιανικής, ώστε να προωθήσουν το προϊόν τους.

Με βάση τα παραπάνω, κάθε μια από τις εγκαταστάσεις της εταιρίας, έχει αριθμό οχημάτων σύμφωνα με τον πίνακα που ακολουθεί (Πίνακας 4).

ΕΓΚΑΤΑΣΤΑΣΗ	ΟΧΗΜΑ				
	ΔΙΟΙΚΗΤΙΚΑ	ΦΟΡΤΗΓΟ 2,5 τν	ΦΟΡΤΗΓΟ 3,5 τν	ΦΟΡΤΗΓΟ 5 τν	ΒΥΤΙΟΦΟΡΑ
Κεντρικά Γραφεία Τρίπολης	3	1	1	-	-
Γραφεία Αιγίου	2	-	-	-	-
Γραφεία Σπάρτης	2	-	-	-	-
Γραφεία Αργους	2	-	-	-	-
Εργοστάσιο Παραγωγής Κυπαρισσίας	1	4	5	2	2
Εργοστάσιο Παραγωγής Ξυλοκάστρου	1	3	6	3	3
Εργοστάσιο Παραγωγής Ναυπλίου	1	2	4	1	1
Σημείο Πώλησης Ολυμπίας	5	1	1	1	-
Σημείο Πώλησης Γυθείου	4	2	1	1	-
Σημείο Πώλησης Μεγαλόπολης	5	3	1	2	-
Σημείο Πώλησης Καλαμάτας	4	2	2	1	-
Σημείο Πώλησης Κρανιδίου	2	1	1	1	-
Σταθμός Συντήρησης Αρεόπολης	1	-	-	-	-

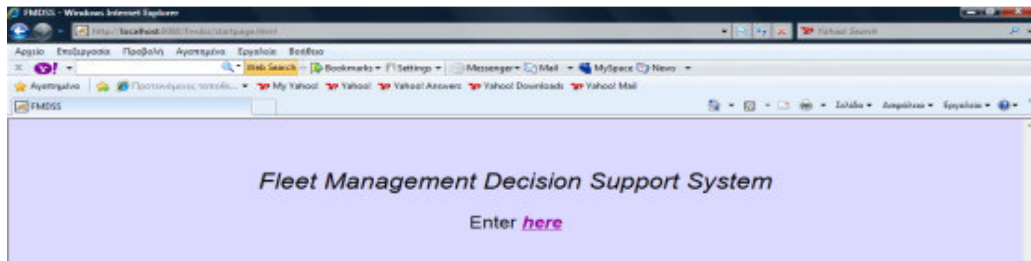
Πίνακας 4: Οχήματα ανά εγκατάσταση

Στη συνέχεια θα παρουσιαστούν πλήρως οι λειτουργίες της εφαρμογής, και πιο συγκεκριμένα:

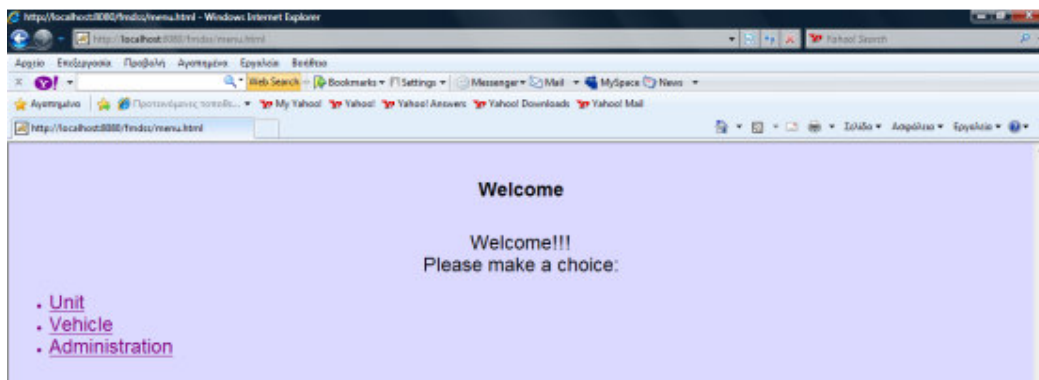
- i. Αγορά νέων οχημάτων και χρέωσή τους στις μονάδες,
- ii. Αίτηση από μονάδα για κίνηση, συντήρηση, απόσυρση με θετικές ή αρνητικές απαντήσεις με έλεγχο διαθεσιμότητας οχημάτων,
- iii. Υλοποίηση κίνησης οχημάτων και ενημέρωση των νέων στοιχείων για το όχημα (πχ ενημέρωση της συνολικής απόστασης).
- iv. Εμφάνιση οχημάτων σε χάρτη και εκτέλεση χωρικών ερωτημάτων για τα οχήματα (πχ πλησιέστερος σταθμός πώλησης από την τρέχουσα θέση).

6.2 Αναλυτική παρουσίαση ελέγχου

Για να έχουμε πρόσβαση στην εφαρμογή, αρκεί να εισάγουμε σε ένα φυλλομετρητή ιστού, τη διεύθυνση <http://localhost:8080/fmdss/startpage.html> και βλέπουμε την αρχική εικόνα της εφαρμογής (Εικόνα 32) από όπου συνδεόμαστε με το βασικό μενού επιλογών (Εικόνα 33).

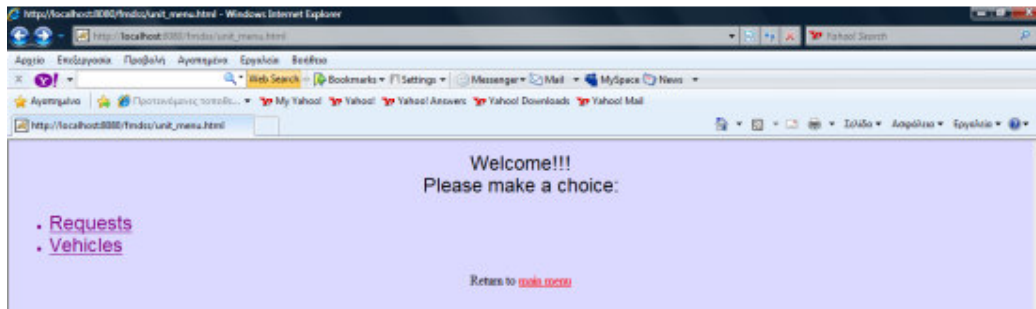


Εικόνα 32: Αρχική Σελίδα εφαρμογής



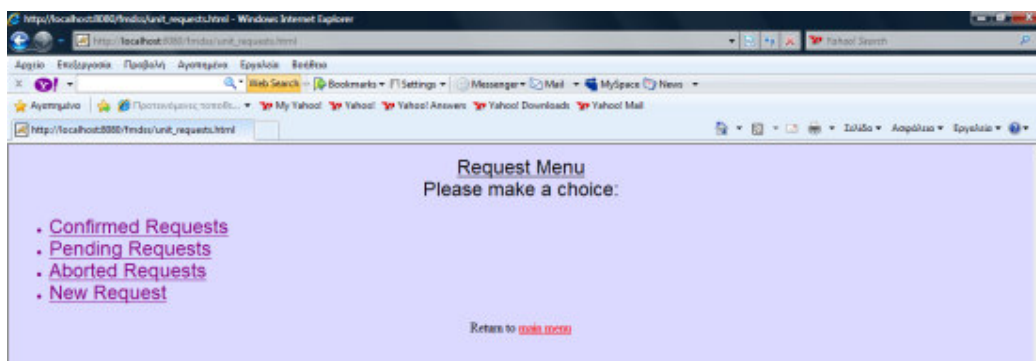
Εικόνα 33: Βασικό Μενού Επιλογών

Πρώτη επιλογή είναι το μενού των μονάδων, όπου δίνεται η δυνατότητα σε μια μονάδα να επεξεργαστεί τις αιτήσεις που έχει κάνει ή να ελέγξει την κατάσταση των οχημάτων της (Εικόνα 34).



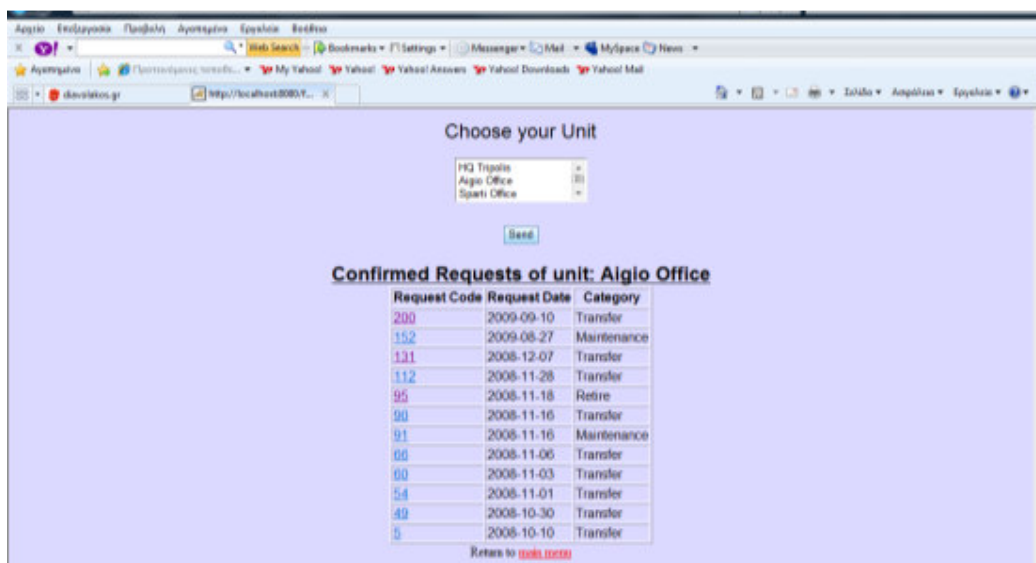
Εικόνα 34: Μενού Μονάδων

Πιο συγκεκριμένα αν γίνει επιλογή του μενού των αιτήσεων, δίνεται σε κάθε μονάδα να δει τις αιτήσεις που έχει κάνει (ποιες έχουν εγκριθεί, ποιες έχουν απορριφθεί και ποιες εκκρεμούν) αλλά και να υποβάλει μια νέα (Εικόνα 35).



Εικόνα 35: Μενού Αιτήσεων

Οι πρώτες τρεις περιπτώσεις, λειτουργούν με παρόμοιο τρόπο. Γίνεται επιλογή της μονάδας που μας ενδιαφέρει, και επιστρέφονται τα αντίστοιχα αιτήματα, από τα οποία μπορούμε μέσω υπερσυνδέσεων να πάρουμε επιπρόσθετες πληροφορίες (Εικόνα 36).



Εικόνα 36: Εγκεκριμένες Αιτήσεις της "Aigio Office"

Έτσι, η υπερσύνδεση μιας μεταφοράς, μας δίνει κατά σειρά πληροφορίες για την κίνηση (πότε ζητήθηκε, πότε λήφθηκε απόφαση, και από πού προς τα πού είναι) (Εικόνα 37), και στη συνέχεια ποιο όχημα την εκτέλεσε και πότε (Εικόνα 38), την απόσταση που διανύθηκε και την κατάσταση στην οποία βρίσκεται τώρα η κίνηση. Τέλος, η υπερσύνδεση στον αριθμό οχήματος, μας δίνει την συνολική εικόνα του αυτοκινήτου (Εικόνα 40), ενώ η υπερσύνδεση προς τον δεσμό “current position” μας δείχνει τη θέση του οχήματος που εκτέλεσε την κίνηση στο χάρτη (Εικόνα 39).

Request Code	Request Date	Decision Date	Request Category	From	To
200	2009-09-10	2009-09-10	Transfer	PYLOS	AIGIO

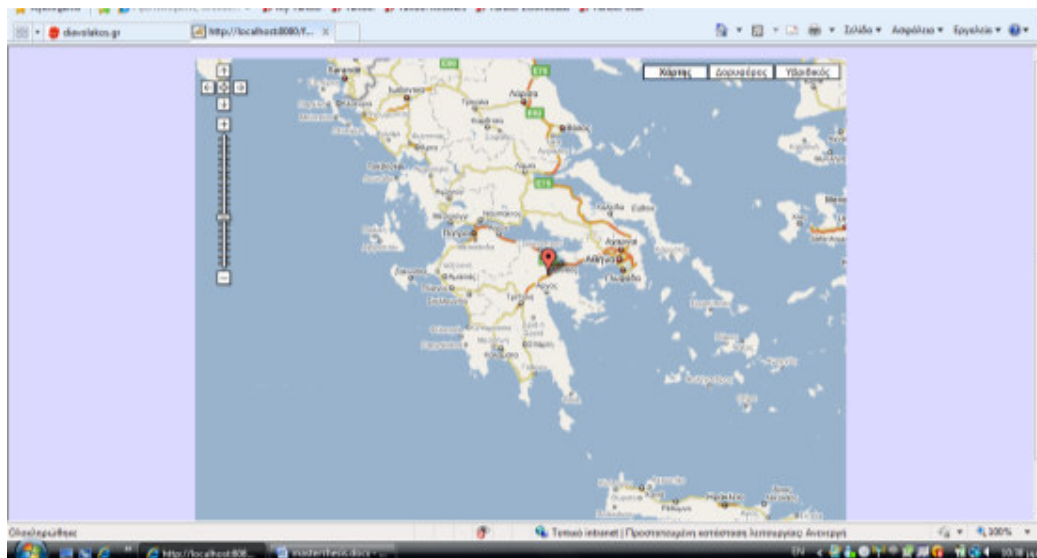
Returns to [main menu](#)

Εικόνα 37: Λεπτομέρειες Αιτήματος

Request Code	Transfer Date	From	To	Vehicle	Condition	Distance
200	2009-09-10	PYLOS AIGIO	YMG24742	In Progress	140	

[Current Position](#)
[Add Transfers](#)
Returns to [main menu](#)

Εικόνα 38: Λεπτομέρειες Κίνησης



Εικόνα 39: Τρέχουσα θέση οχήματος στο χάρτη

Vehicle Status: YMZ4742

Brand	Total Kilometres	Purchase Date	Usage	Category
Honda	120000.00	2002-06-18	Administrative	Car

Transfers

Request Code	Transfer Date	From	To	Condition	Distance
5	2008-10-12	AIGIO	KILINI	Accomplished	100
49	2008-11-01	AIGIO	KILINI	Accomplished	100
54	2008-11-03	AIGIO	LOYTRA KAIAFAS	Accomplished	89
60	2008-11-05	AIGIO	HORA	Accomplished	125
66	2008-11-08	AIGIO	AREOPOLI	Accomplished	161
90	2008-11-18	AIGIO	ANDRITSENA	Accomplished	79
112	2008-11-30	AIGIO	MILOI	Accomplished	94
131	2008-12-09	AIGIO	LOYTRA ELENIS	Accomplished	98
200	2009-09-10	PYLOS	AIGIO	In Progress	140

Maintenances

Request Code	Start date	End Date	Cause	Cost
91	2008-11-18	2008-11-23	Accident	1120
152	2009-08-29	2009-09-10	Programmed	1500

[Current Position](#)
[Return to main menu](#)

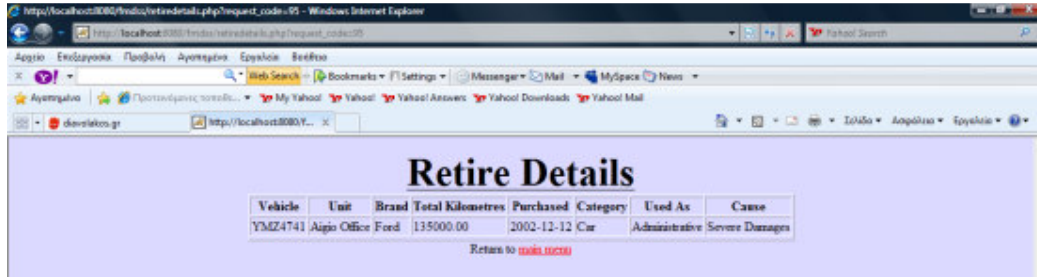
Εικόνα 40: Κατάσταση οχήματος που εκτέλεσε την κίνηση

Με παρόμοιο τρόπο λειτουργεί και η επιλογή μιας συντήρησης. Μας δίνει αρχικά τις βασικές πληροφορίες συντήρησης (Εικόνα 41) , η υπερσύνδεση του οχήματος, μας οδηγεί όπως και πριν στην συνολική κατάσταση του οχήματος (Εικόνα 40), ενώ η υπερσύνδεση της κατηγορίας, μας δίνει όλες τις πληροφορίες της συντήρησης (Εικόνα 42).

Εικόνα 41: Λεπτομέρειες Αιτήματος Συντήρησης

Εικόνα 42: Λεπτομέρειες Συντήρησης

Η περίπτωση τέλος μιας απόσυρσης οχήματος, μας δίνει τις αρχικές πληροφορίες αιτήματος όπως και στην περίπτωση της συντήρησης (Εικόνα 41) και στη συνέχεια τις λεπτομέρειες της απόσυρσης (Εικόνα 43). Οι υπόλοιπες υπερσυνδέσεις λειτουργούν όπως και στις αντίστοιχες άλλες περιπτώσεις.



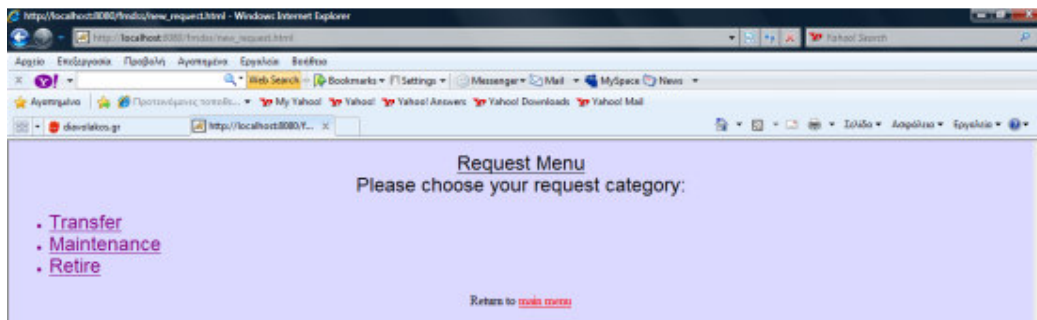
The screenshot shows a web browser window with the URL http://localhost:8080/htdocs/retireDetails.php?request_code=05. The page title is "Retire Details". Below the title is a table with the following data:

Vehicle	Unit	Brand	Total Kilometres	Purchased	Category	Used As	Cause
ΥΜΖ4741	Aigio	Office Ford	135000.00	2002-12-12	Car	Administrative	Severe Damages

Below the table, there is a link: [Returns to main menu](#).

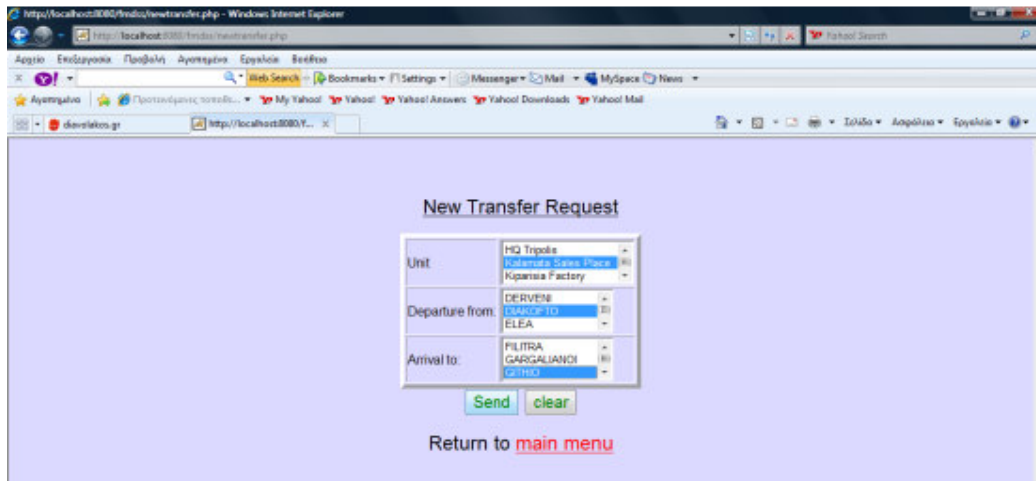
Εικόνα 43: Λεπτομέρειες Απόσυρσης

Η επιλογή υποβολής μιας νέας αίτησης (Εικόνα 35), μας οδηγεί σε νέα οθόνη, όπου γίνεται επιλογή του είδους της αίτησης (Εικόνα 44), όπου ανάλογα με το τι επιλέξουμε εκτελούνται και αντίστοιχες διαδικασίες.

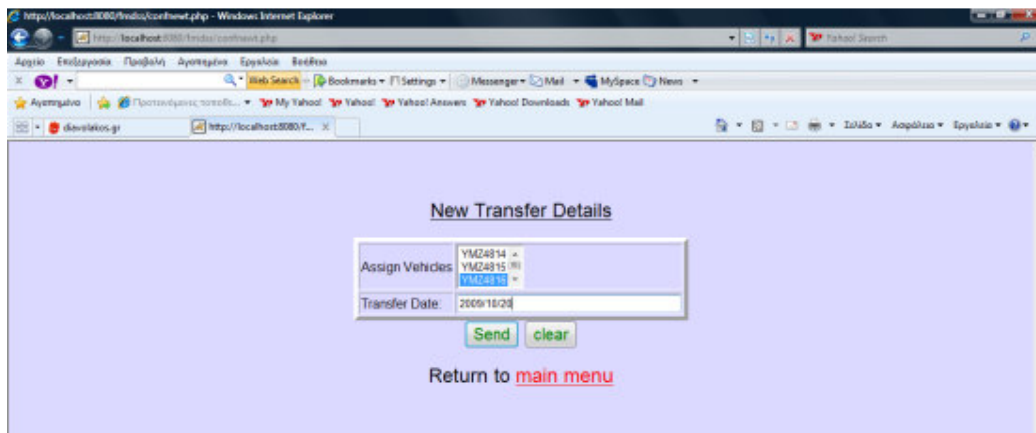


Εικόνα 44: Επιλογή Κατηγορίας Αίτησης

Η περίπτωση αίτησης για κίνηση έχει ως πρώτο στάδιο την επιλογή της μονάδας που αιτείται καθώς επίσης και από ποια πόλη προς ποια θα γίνει η κίνηση (Εικόνα 45). Οι επιλογές αυτές, μαζί με το κωδικό της αίτησης και την ημερομηνία της, καταχωρούνται στους πίνακες request και transfer. Στο στάδιο αυτό γίνεται και ο υπολογισμός της απόστασης των δύο πόλεων. Στην εφαρμογή, η απόσταση είναι η ευθεία απόσταση μεταξύ των δύο πόλεων, και συνεπώς διαφέρει από την πραγματική. Επίσης γίνεται επιλογή του οχήματος της μονάδας που θα εκτελέσει την κίνηση (η επιλογή γίνεται από τα διαθέσιμα οχήματα) και η επιλογή της ημερομηνίας μεταφοράς (Εικόνα 46). Αφού γίνουν όλα αυτά, η κίνηση καταχωρείται και αναμένει την απόφαση που θα ληφθεί.

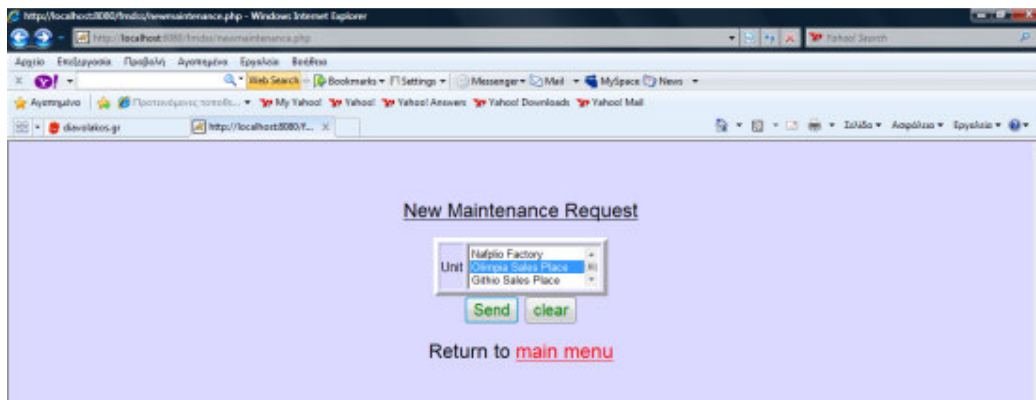


Εικόνα 45: Καταχώρηση Μονάδας - Αφετηρίας - Τερματισμού

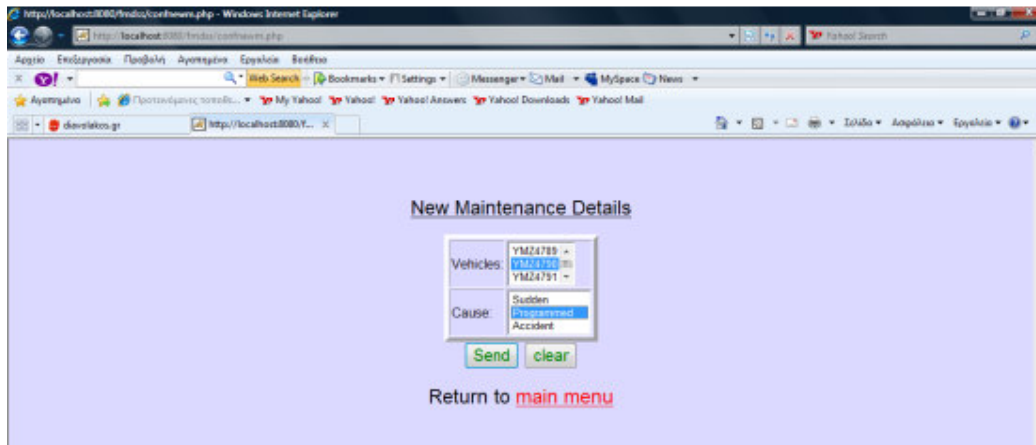


Εικόνα 46: Καταχώρηση Οχήματος - Ημερομηνίας Μεταφοράς

Ανάλογες διαδικασίες λαμβάνουν χώρα και στην περίπτωση συντήρησης. Αρχικά επιλέγεται η μονάδα (Εικόνα 47), στη συνέχεια το όχημα και ο λόγος για τον οποίο γίνεται η αίτηση συντήρησης (Εικόνα 48). Γίνονται οι κατάλληλες ενημερώσεις των πινάκων maintenance – request, και η συντήρηση περνά σε κατάσταση αναμονής για την έγκρισή της.



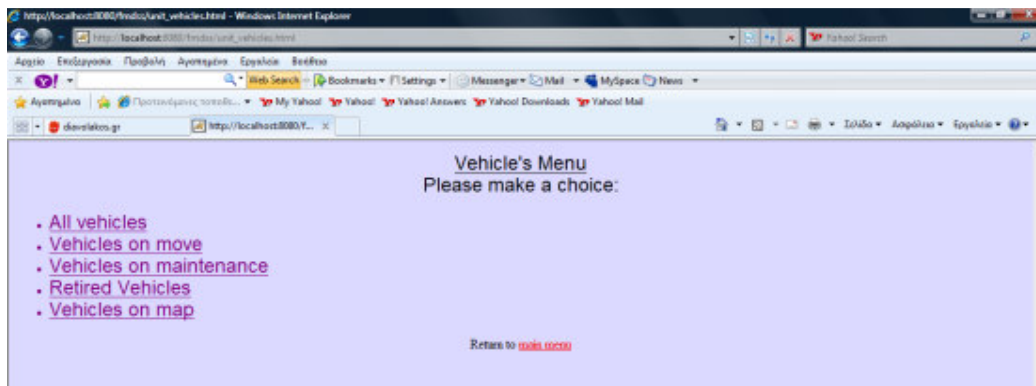
Εικόνα 47: Καταχώρηση Μονάδας



Εικόνα 48: Καταχώρηση Οχήματος – Αιτίας

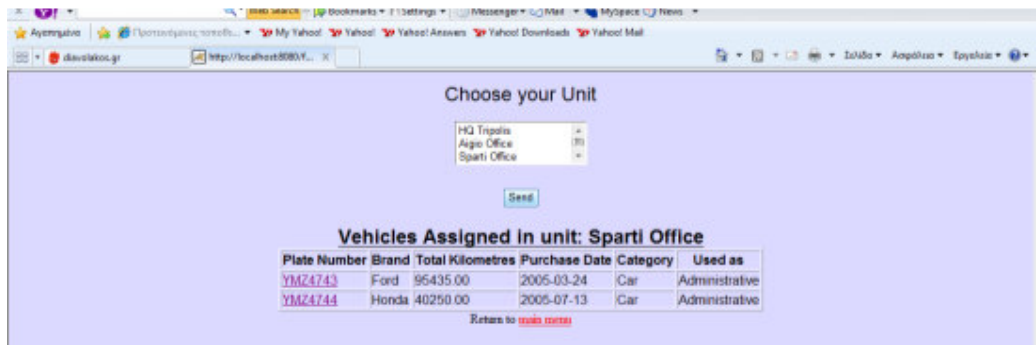
Νέα αίτηση για απόσυρση οχήματος, είναι πανομοιότυπη διαδικασία με τη συντήρηση, μόνο που εδώ γίνονται εισαγωγές και ενημερώσεις στον πίνακα retire.

Δεύτερη μεγάλη κατηγορία επιλογών για μια μονάδα, είναι η επιλογή των οχημάτων τη, όπου μας δίνεται η δυνατότητα να δούμε όλα τα οχήματα μιας μονάδας, ποια είναι σε κίνηση, ποια σε συντήρηση και ποια έχουν αποσυρθεί. Επίσης υπάρχει και η επιλογή να δούμε τα οχήματα στο χάρτη (Εικόνα 49).



Εικόνα 49: Επιλογές για τα οχήματα μιας Μονάδας

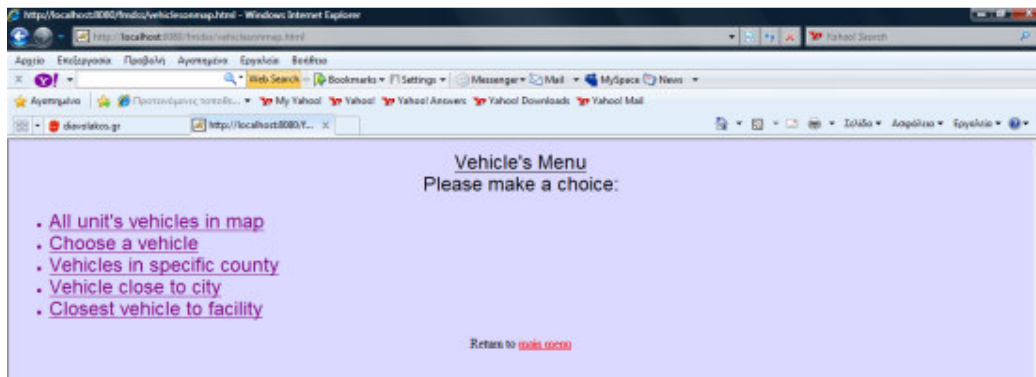
Οι πρώτες επιλογές έχουν σαν αποτέλεσμα να δούμε τα οχήματα, ανάλογα με την κατάσταση στην οποία βρίσκονται. Τα λοιπά links στις σελίδες, λειτουργούν όπως αναλύθηκε πιο πάνω. Έτσι αν επιλέξουμε να δούμε όλα τα οχήματα μιας μονάδας, προκύπτει το ακόλουθο αποτέλεσμα:



Εικόνα 50: Όλα τα οχήματα μιας Μονάδας

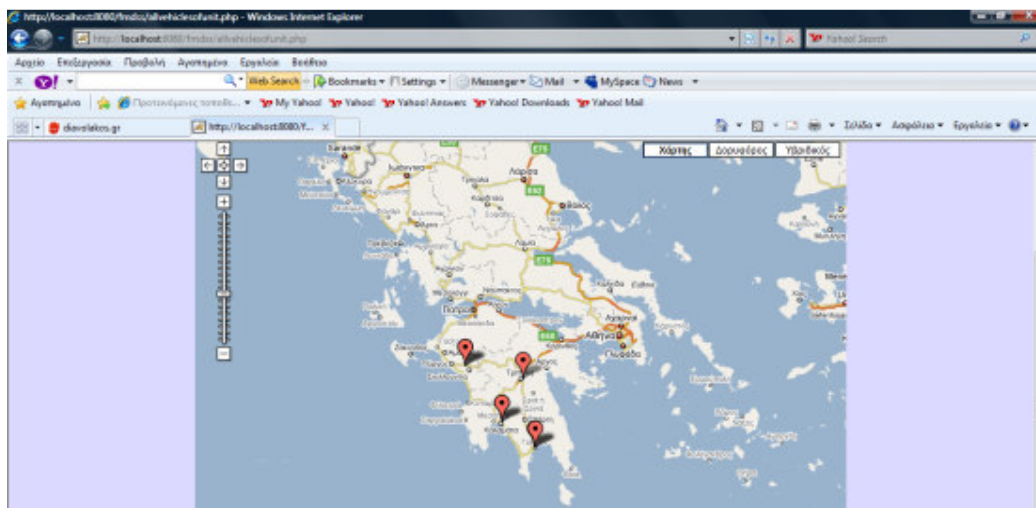
Αντίστοιχα λειτουργούν και οι άλλες περιπτώσεις.

Ιδιαίτερη υπάρχει στην επιλογή Vehicles On Map, που μας οδηγεί σε μια σειρά άλλων επιλογών (όλα τα οχήματα, συγκεκριμένο όχημα, όχημα σε νομό, όχημα κοντά σε πόλη, όχημα κοντά σε εγκατάσταση) (Εικόνα 51).

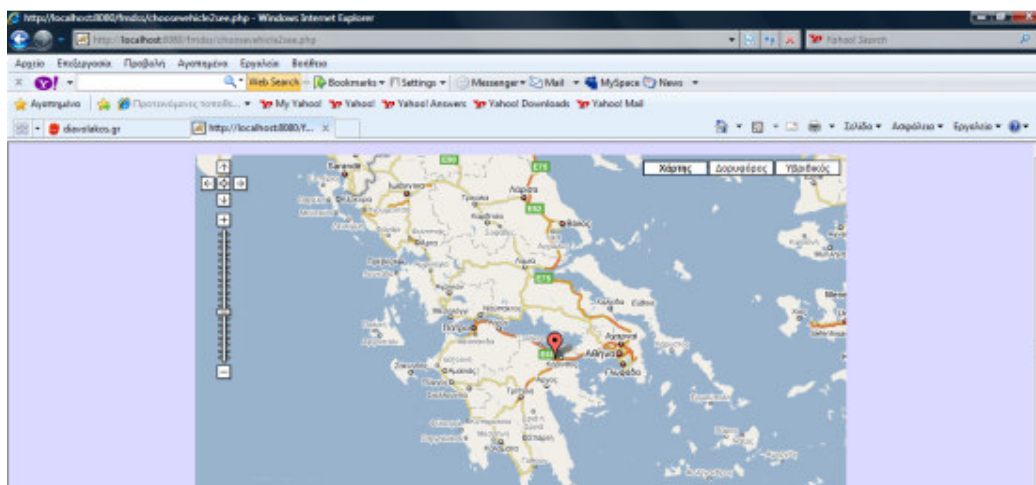


Εικόνα 51: Επιλογές Οχημάτων στον Χάρτη

Με την πρώτη επιλογή, προκύπτουν όλα τα οχήματα συγκεκριμένης μονάδας (Εικόνα 52), με την δεύτερη η θέση συγκεκριμένου οχήματος (Εικόνα 53).



Εικόνα 52: Θέσεις οχημάτων μιας μονάδας



Εικόνα 53: Θέση Οχήματος

Οι άλλες τρεις περιπτώσεις, έχουν να κάνουν και με χωρικό κομμάτι. Θέλουμε να δούμε ποια οχήματα κινούνται εντός συγκεκριμένου νομού, ποιο βρίσκεται πιο κοντά σε συγκεκριμένη πόλη, και ποιο πλησίον άλλου οχήματος, οπότε και προκύπτουν ανάλογες εικόνες όπως και πιο πάνω. Στην περίπτωση του νομού, κάνουμε επιλογή νομού, οπότε εκτελείται ερώτημα εγκλεισμού, όπου ζητούνται όλα τα οχήματα που βρίσκονται εντός της περιφέρειας του νομού (`where within(position,(select shape from county where county_name = '...'))`). Εδώ εμφανίζεται και το πρόβλημα των ΕΠΠ, που έχουν ως αποτέλεσμα, ένα όχημα να φαίνεται πως βρίσκεται εντός δυο νομών.

Οι άλλες δύο περιπτώσεις, είναι διαδικασία εύρεσης πλησιέστερου γείτονα, που όμως επειδή δεν υλοποιείται ακόμα στη MySQL, το υπολογίζουμε με εναλλακτική τεχνική. Βρίσκουμε όλες τις αποστάσεις των οχημάτων μιας μονάδας από μια πόλη ή μια εγκατάσταση αντίστοιχα (πχ για την πόλη με τη χρήση της «`ROUND(GLength(LineStringFromWKB(LineString(AsBinary(c1.location),AsBinary(l1.position))))*100`» και ζητάμε να επιστραφεί η μικρότερη τιμή, βάζοντας ένα `min()` σε όλο το παραπάνω. Έτσι λύνεται το πρόβλημα του πλησιέστερου γείτονα, όχι όμως των `k`-πλησιέστερων γειτόνων.

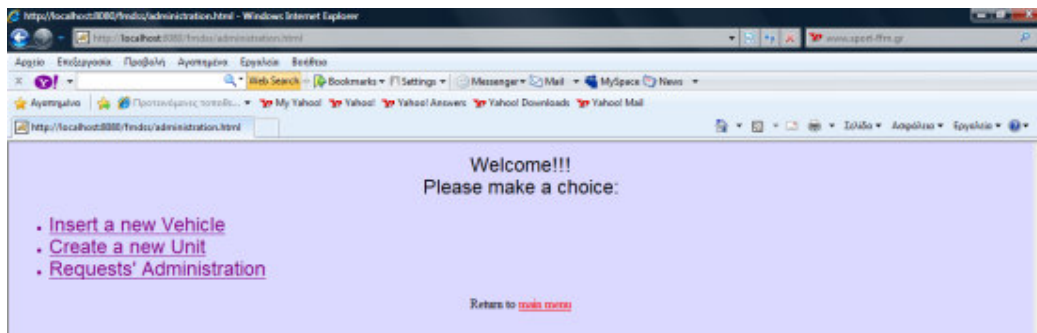
Δεύτερη βασική επιλογή είναι η επιλογή Vehicle, στην οποία δίνουμε ως είσοδο τον αριθμό κυκλοφορίας του οχήματος, και λαμβάνουμε τη θέση του στο χάρτη. Υπάρχουν δύο επιλογές μετά. Είτε να δούμε την το ιστορικό του όπως προηγουμένως, είτε να δούμε την κατάσταση κίνησης του: Σε ποιο νομό βρίσκεται, σε ποιο δρόμο κινείται, τι βρίσκεται πλησίον του (πόλη, όχημα, εγκατάσταση) και σε τι απόσταση (Εικόνα 54).



Εικόνα 54: Κατάσταση Κίνησης Οχήματος

Η πίνακας που μας την κατάσταση κίνησης του οχήματος, προκύπτει μετά από εκτέλεση μιας σειράς χωρικών ερωτημάτων. Πιο συγκεκριμένα για τον υπολογισμό μέσα σε ποιο νομό κινείται, γίνεται εκτέλεση ερωτήματος εγκλεισμού όπως και πριν `within(GeomFromText('point(συντεταγμένες)'),shape))`. Για τον υπολογισμό του δρόμου κίνησης, ερώτημα τομής (`within(GeomFromText('point(συντεταγμένες)'),way))`). Τέλος, για τον υπολογισμό πλησιέστερης πόλης, εγκατάστασης, οχήματος, εργαζόμαστε όπως και προηγουμένως, με την ιδιαιτερότητα ότι για το όχημα, πρέπει να προσθέσουμε ένα ακόμα κριτήριο, που θα υπολογίζει και την απόσταση ενός οχήματος από τον εαυτό του, θα την βρίσκει μηδενική και θα την επιστρέφει ως ελάχιστη.

Τελευταία ομάδα επιλογών είναι οι επιλογές διαχείρισης (Εικόνα 55), όπου έχουμε τη δυνατότητα να δημιουργήσουμε μια νέα μονάδα (Εικόνα 56), να εισάγουμε ένα νέο όχημα (Εικόνα 57) και, κυρίως, να λάβουμε αποφάσεις σχετικά με τα αιτήματα που έχουν υποβληθεί (Εικόνα 58).



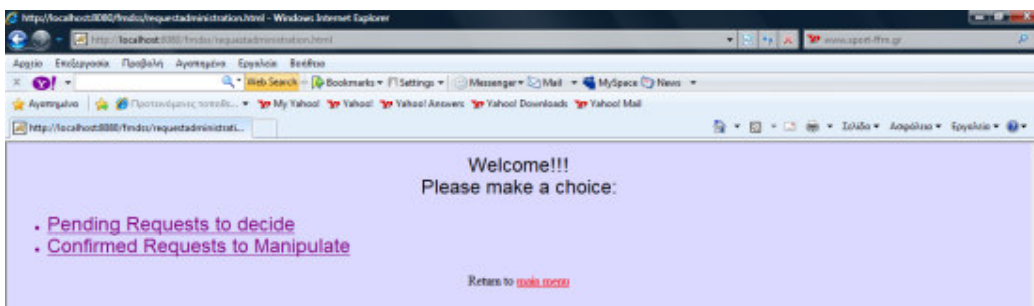
Εικόνα 55: Επιλογές Διαχείρισης



Εικόνα 56: Εισαγωγή Μονάδας

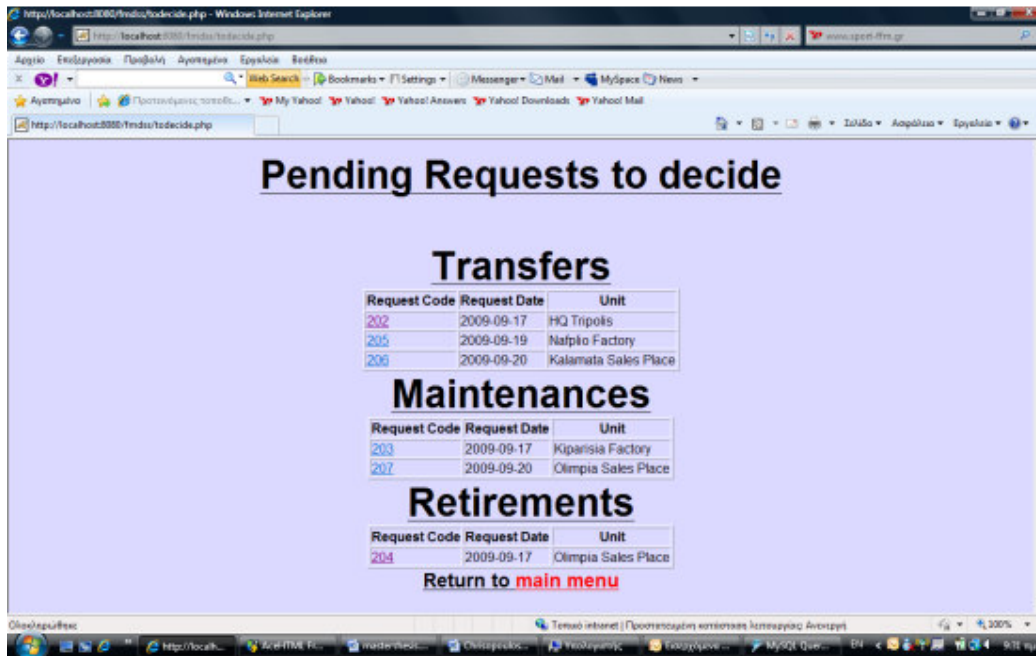


Εικόνα 57: Εισαγωγή Οχήματος



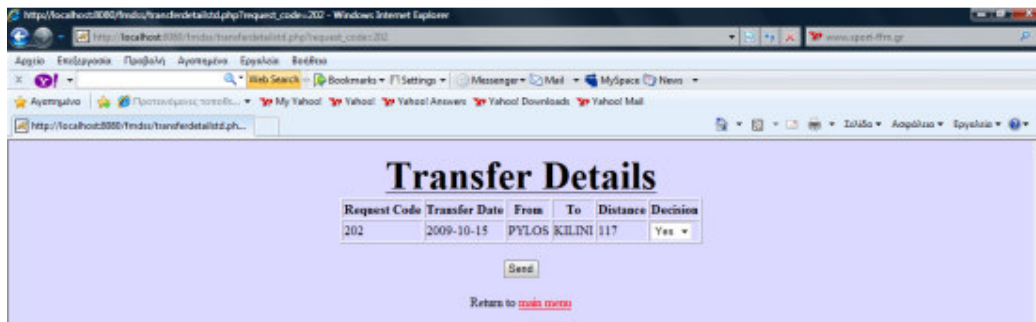
Εικόνα 58: Διαχείριση Αιτημάτων

Η επιλογή διαχείρισης αιτημάτων έχει δυο εναλλακτικές: Εκκρεμείς αιτήσεις για να ληφθεί απόφαση (Εικόνα 59), αιτήματα για τα οποία έχει ληφθεί απόφαση, για χειρισμό (). Σε κάθε περίπτωση τα αιτήματα εμφανίζονται ανά κατηγορία.



Εικόνα 59: Εκκρεμείς Αιτήσεις για λήψη απόφασης

Αν επιλέξουμε να αποφασίσουμε για μια κίνηση, βλέπουμε τα χαρακτηριστικά της (αφετηρία, προορισμός, ημερομηνία κίνησης, απόσταση), και επιλογή απόφασης (Yes/No). (Εικόνα 60)



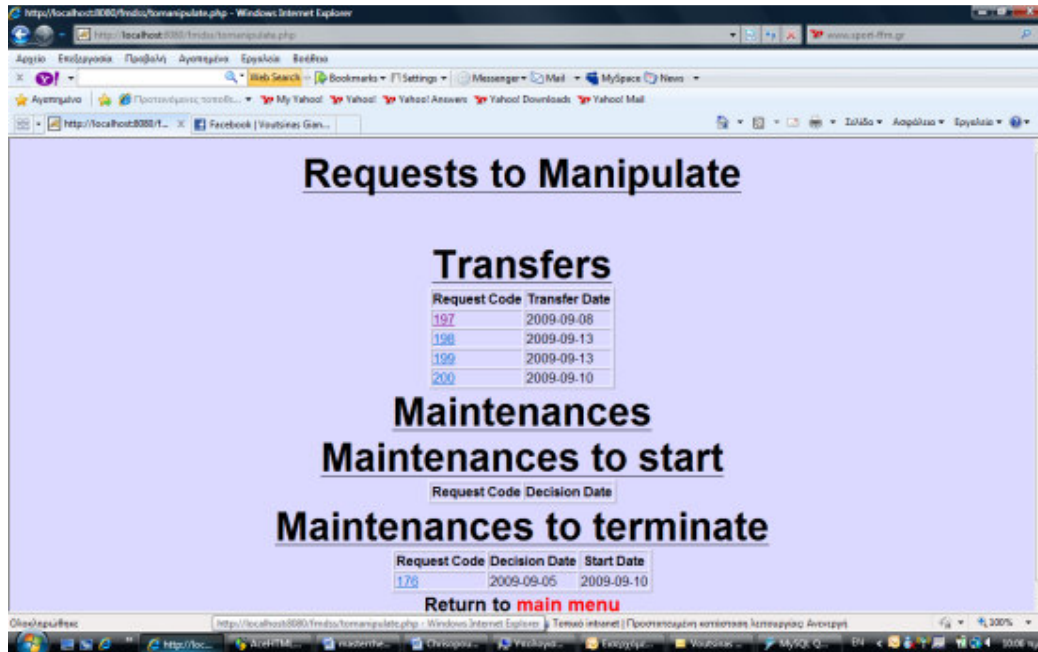
Εικόνα 60: Κίνηση για Απόφαση

Μόλις ληφθεί απόφαση, γίνεται ενημέρωση των πεδίων απόφασης και ημερομηνίας απόφασης στον πίνακα request. Αν η απόφαση είναι αρνητική, διαγράφεται η κίνηση από τον πίνακα vehicle_transfer, και ενημερώνεται η κατάσταση της κίνησης (Cancelled). Αν είναι θετική, αλλάζει η κατάσταση σε scheduled και δημιουργείται μια νέα κίνηση για επιστροφή στην αφετηρία, έχοντας ως παραδοχή ότι δεν γίνονται κινήσεις με πολλά σκέλη.

Παρόμοια λειτουργεί και η απόφαση συντήρησης. Αν είναι αρνητική, αλλάζει η διαθεσιμότητα του οχήματος, ώστε να είναι και πάλι διαθέσιμο, ενώ αν είναι θετική προγραμματίζεται και κίνηση προς τον χώρο συντήρησης.

Τέλος στην απόσυρση, ανάλογα με την απόφαση, ενημερώνονται οι πίνακες και η διαθεσιμότητα του οχήματος.

Τελευταίο μενού επιλογών η διαχείριση των αιτημάτων για τα οποία έχει ληφθεί απόφαση (Εικόνα 61).



Εικόνα 61: Διαχείριση Αιτημάτων για τα οποία έχει ληφθεί απόφαση

Η επιλογή αυτή έχει νόημα μόνο για κινήσεις και συντηρήσεις. Πιο συγκεκριμένα έχουμε τη δυνατότητα να αλλάξουμε την κατάσταση μιας κίνησης, ακόμα και να την ακυρώσουμε. Όταν την κάνουμε In Progress αλλάζει η διαθεσιμότητα του οχήματος σε μη διαθέσιμο, ενώ όταν την κάνουμε Accomplished, γίνεται και πάλι διαθέσιμο, ενώ αλλάζει και η τιμή στον δείκτη χιλιομέτρων.

Αντίστοιχα στις συντηρήσεις, είτε τις ξεκινάμε, είτε τις τερματίζουμε εισάγοντας τα αντίστοιχα στοιχεία (Ημερομηνίες, κόστος κλπ). Αν τερματιστεί μια συντήρηση, υλοποιείται και μια κίνηση προς την έδρα του οχήματος.

7

Επίλογος

Στο κεφάλαιο αυτό γίνεται σύνοψη της μεταπτυχιακής διατριβής και των αποτελεσμάτων. Αναφέρονται κατευθύνσεις για την συνέχεια και επέκταση των αποτελεσμάτων.

7.1 Σύνοψη και συμπεράσματα

Αφού έγινε εκτενής βιβλιογραφική μελέτη των διαφόρων τεχνολογιών και μεθοδολογιών, υλοποιήθηκε μια web εφαρμογή για τη διαχείριση στόλου οχημάτων. Χρησιμοποιήθηκε λογισμικό ανοικτού κώδικα και τα αποτελέσματα ήταν ικανοποιητικά. Με τη χρήση της MySQL ως ΣΔΒΔ, αντιμετωπίστηκαν προβλήματα ως απόρροια της μη ολοκληρωμένης υλοποίησης χωρικών συστατικών, αλλά εν μέρει επιλύθηκαν με εναλλακτικές μεθόδους. Η εμφάνιση των χωρικών δεδομένων σε χάρτη, έγινε μέσω των google maps, ένα ισχυρό εργαλείο, με πολλές δυνατότητες. Το διαχειριστικό κομμάτι έγινε προσπάθεια να είναι όσο το δυνατόν πιο κοντά στην πραγματικότητα, με βάση την υπάρχουσα εμπειρία από την διεκπεραίωση κινήσεων στο εργασιακό περιβάλλον.

Μικρά προβλήματα υπήρξαν και κυρίως είχαν να κάνουν με τις συνεχείς ενημερώσεις που έπρεπε να γίνονται όταν άλλαζε η κατάσταση ενός οχήματος, ή μιας αίτησης. (πχ. όταν ένα όχημα εκτελεί μια κίνηση, δεν εμφανίζεται διαθέσιμο ώστε να γίνει αίτηση κίνησης για αυτό σε άλλη ημερομηνία. Για να εμφανιστεί πρέπει να αλλάξει η διαθεσιμότητά του, άρα να ολοκληρωθεί η κίνηση).

Επίσης όλες οι αποστάσεις είναι ευθείες που ενώνουν τα δυο σημεία και δεν ανταποκρίνονται στις πραγματικές αποστάσεις που εκτελούνται σε μια κίνηση.

Τέλος, η περίπτωση του πλησιέστερου γείτονα λύθηκε με εναλλακτικό τρόπο, ενώ των $k - \text{πλησιέστερων}$ όχι.

7.2 Μελλοντικές επεκτάσεις

Μελλοντικές επεκτάσεις της εφαρμογής υπάρχουν αρκετές. Άλλες μπορούν να υλοποιηθούν άμεσα, ενώ για άλλες χρειάζεται χρόνος ενασχόλησης. Πιο συγκεκριμένα:

i. Δρομολόγηση: Θα πρέπει όταν γίνεται αίτηση μιας κίνησης, να προκύπτει και η βέλτιστη διαδρομή (είτε χιλιομετρικά, είτε χρονικά). Υπάρχουν λύσεις για αυτό με πιο συνηθισμένη τον αλγόριθμο Dijkstra.

ii. Αποστολή στίγματος από τα τερματικά ώστε να ενημερώνουν για την τρέχουσα θέση τους: Αυτό μπορεί να υλοποιηθεί είτε σε java είτε σε C#. Αποτελεί μια από τις άμεσες προτεραιότητες, ώστε να μπορεί η εφαρμογή να σταθεί σε πραγματική χρήση.

iii. Δυνατότητα εκτέλεσης διαδρομών με πολλά σκέλη: Στην παρούσα φάση, μια κίνηση γίνεται από ένα σημείο σε ένα άλλο και επιστροφή στο αρχικό. Θέλουμε να έχουμε κινήσεις σε πολλά σκέλη με ή όχι επιστροφή.

iv. Εξουσιοδοτημένη χρήση: Στην παρούσα φάση δεν υπάρχει έλεγχος. Κάθε χρήστης μπορεί να μπει στην εφαρμογή και να δράσει όπως θέλει χωρίς περιορισμούς. Έλεγχος χρήστη και ανάλογα με τα δικαιώματα του χρήση ορισμένων μόνο λειτουργιών.

v. Εξωραϊσμός της web εφαρμογής με εικόνες ώστε να είναι πιο φιλική στο χρήστη, αλλά να δίνει και περισσότερες πληροφορίες.

8

Βιβλιογραφία

Ν.Χρυσόπουλος, Γ.Θεοδορίδης. Διαχείριση Κίνησης Αντικειμένων σε Πραγματικό Χρόνο με εργαλεία GIS,Μεταπτυχιακή Διατριβή, Πανεπιστήμιο Πειραια,2006

E. Frentzos, K. Gratsias, N. Pelekis, Y. Theodoridis. Nearest Neighbor Search on Moving Object Trajectories. Proc. 9th International Symposium on Spatial and Temporal Databases (SSTD'05), Angra dos Reis, Brazil, August 2005

Tao,Y. ,Papadias ,D., and Shen,Q., Continuous Nearest Neighbor Search,Proceedings of VLBD,2002.

MySQL 5.0 Reference Manual, (Online Help)

R.Elmasri – S.BNavathe. Θεμελιώδεις Αρχές Συστημάτων Βάσεων Δεδομένων, τόμοι Α΄ και Β΄ (μτφ. Μ.Χατζόπουλου) , 2000

T.Connolly – C.Beggs- A.Strachan, Συστήματα Βάσεων Δεδομένων τόμος Α΄ (1999)