



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Σχεδίαση και Ανάπτυξη της Εφαρμογής Vessel Tracking System με χρήση Spatial Database Design and Application Development of Vessel Tracking System using Spatial Database
Όνοματεπώνυμο Φοιτητή	Ιωάννης Καραγουστής
Πατρώνυμο	Νικόλαος
Αριθμός Μητρώου	ΜΠΣΠ/ 10025
Επιβλέποντες	Ιωάννης Θεοδωρίδης, Καθηγητής Μάριος Βόντας, Διδακτορικός

Πανεπιστήμιο Πειραιώς

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Ιωάννης Θεοδωρίδης
Καθηγητής

Ιωάννης Σίσκος
Καθηγητής

Νίκος Πελέκης
Λέκτορας

Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας αποτελεί η σχεδίαση και ανάπτυξη της εφαρμογής Vessel Tracking System. Η εφαρμογή αυτή σχεδιάστηκε ώστε να χρησιμοποιηθεί από ναυτιλιακές εταιρίες στις οποίες θα παρέχει τη δυνατότητα παρακολούθησης του στόλου τους σε πλοία που έχουν εισάγει.

Μέσω της εφαρμογής αυτής ο κάθε εξουσιοδοτημένος χρήστης της εταιρίας θα έχει τη δυνατότητα να δει πληροφορίες σχετικές με τις θέσεις των πλοίων, τις τροχιές των πλοίων καθώς και σχετικές πληροφορίες για κάθε σημείο τους όπως ταχύτητα, εκτίμηση χρόνου άφιξης σε προορισμό, διεύθυνση τροχιάς κ.α. Επίσης ο χρήστης θα μπορεί να λαμβάνει ειδοποιήσεις στη περίπτωση που κάποιο πλοίο εισέλθει σε οποιαδήποτε περιοχή ενδιαφέροντος ενώ θα είναι σε θέση να αναγνωρίσει και το σημείο εισόδου του πλοίου στην περιοχή αυτή.

Abstract

This dissertation focuses on the design and development of the Vessel Tracking System Desktop Application. This application was designed to be used by Shipping Companies in order to provide them the ability to be able to monitoring all the vessels of their fleets.

Through this application, each authorized user of the shipping company will be able to get information regarding vessel points in the map, vessels routes and more details for them such as speed, ETA, Destination Port, heading etc. Moreover each user will be able to get notifications in case a vessel inserts into a Zone of Interest and also to recognize the specific vessel's entry point in the aforementioned area.

Περιεχόμενα

Περίληψη	3
Abstract	3
1. Εισαγωγή	9
1.1 Αντικείμενο Διπλωματικής εργασίας.....	9
1.2 Δομή της διπλωματικής εργασίας.....	10
2. Συστήματα Γεωγραφικών Πληροφοριών (GIS) & Παγκόσμιο Σύστημα Εντοπισμού Θέσης (GPS)	11
2.1 Τι είναι το GIS.....	11
2.2 Περιγραφή GIS Συστημάτων	12
2.3 Τι είναι το GPS.....	14
3. Το Αυτόματο Σύστημα Εντοπισμού AIS (Automatic Identification System).....	15
3.1 Λειτουργία του AIS.....	15
3.2 Τύποι και συσκευές του AIS	16
3.2.1 Τύπος A (Class A)	16
3.2.2 Τύπος B (Class B)	17
3.2.3 AtoN (Aids to Navigation).....	17
3.2.4 SART (Search and Rescue Transponder)	17
3.2.5 Specialist AIS Transponders.....	18
3.2.6 AIS Base Stations	18
3.3 AIS Δεδομένα.....	19
4. Εργαλεία Ανάπτυξης και Υλοποίησης.....	21
4.1 Microsoft Visual Studio 2012	21
4.1.1 Windows Forms	21
4.1.2 .NET Framework	21
4.1.2.1 Common Language Runtime.....	22
4.1.2.2 .NET Class Libraries	23
4.2 Microsoft SQL Server	24
4.2.1 Microsoft SQL Server 2012.....	24
4.2.1.1 Γεωγραφικοί τύποι δεδομένων του SQL Server 2012	24
4.2.1.2 Διαφορές μεταξύ γεωγραφικών και γεωμετρικών τύπων δεδομένων.....	28
4.2.2 SQL Agent.....	29
4.3 JavaScript	29

4.4	JSON (JavaScript Object Notation)	29
4.5	Html (Hyper Text Markup Language)	30
4.6	Google Maps JavaScript API v3	31
5.	Αντίστοιχες Υλοποιημένες Εφαρμογές	31
5.1	Marinetraffic.....	31
5.2	VesselFinder.....	33
5.3	VT Explorer.....	34
5.4	Συγκριτικός πίνακας εφαρμογών	35
6.	Σχεδιασμός και Ανάλυση της Βάσης Δεδομένων.....	36
6.1	Το σύνολο δεδομένων (dataset) της εφαρμογής	38
6.2	Οι Πίνακες της εφαρμογής.....	40
6.3	Ανάλυση Βασικών Πινάκων της Εφαρμογής	41
6.3.1	Πίνακας tblImisPoolData.....	41
6.3.2	Πίνακας tblLiveShipData	42
6.3.3	Πίνακας tblLiveShipDataLineString	42
6.3.4	Πίνακας tblZonesOfInterest & tblZonesOfInterestCoordinates	43
6.3.5	Πίνακας tblShips.....	44
6.3.6	Πίνακας tblNotifications.....	44
6.3.7	Πίνακας tblShipType.....	45
6.4	Ανάλυση βασικών Store Procedures.....	46
6.4.1	SM_InitializeImisPoolData.....	47
6.4.2	SM_ReadPoolData.....	50
6.4.3	SM_PolylinesIntoPolygons.....	50
6.4.4	SM_GetVesselsLast2CurrentPos.....	51
6.4.5	SM_GetVesselsRoutes.....	51
7.	Αρχιτεκτονική του Συστήματος.....	52
8.	Υλοποίηση	53
8.1	Εμφάνιση Google Maps χάρτη	53
8.2	Οι κλάσεις του προγράμματος	54
8.3	Modes λειτουργίας	55
8.4	Τα .htm αρχεία και το API της Google	56
8.5	Περιεχόμενο των .htm αρχείων.....	57
8.6	Το JSON ως interface μεταξύ VB.NET και JavaScript.	61
8.7	Εμφάνιση των InfoWindows.....	65

8.8	Τα icons των Markers μέσω της εφαρμογής Dropbox.....	65
9.	Περιγραφή λειτουργικότητας	71
9.1	Οθόνη Σύνδεσης Χρήστη.....	71
9.2	Καρτέλα πληροφορίας πλοίων (Vessel Information Oriented Tab)	72
9.2.1	Λιμάνια – Ζώνες Ενδιαφέροντος (Ports – Zones Of Interest)	72
9.2.2	Εικόνα 9-4: Επιλογή εισαγωγής Λιμανιών και Ζωνών Ενδιαφέροντος	73
9.2.3	Εμφάνιση τελευταίας ληφθείσας θέσης πλοίου (Ship Latest Received Positions)	73
9.2.4	Εμφάνιση διαδρομών πλοίων (Ship Routes).....	73
9.2.5	Φόρτωση του χάρτη	74
9.2.6	Διαθέσιμα φίλτρα με βάση τα στοιχεία των πλοίων	77
9.2.7	Διαθέσιμο φίλτρο με βάση την ημερομηνία	78
9.3	Καρτέλα εμφάνισης ειδοποιήσεων (Vessel Notification Oriented Tab).....	80
9.4	Καρτέλα Διαχείρισης (Vessel Administrator Oriented Tab)	84
9.4.1	Διαχείριση χρηστών (Manage Users)	84
9.4.2	Διαχείριση Χωρών (Manage Countries)	85
9.4.3	Διαχείριση Λιμανιών (Manage Ports).....	86
9.4.4	Διαχείριση Πλοίων (Manage Vessels).....	87
9.4.5	Διάστημα χρόνου ανανέωση χάρτη (Map Refresh Interval).....	88
10.	Μελλοντικές επεκτάσεις.....	89
10.1	Αναζήτηση με βάση το Όνομα/MMSI Number του πλοίου	89
10.2	Μηχανισμός αποστολής ειδοποιήσεων	89
10.3	Διόρθωση υπαρχόντων λειτουργικών προβλημάτων	90
10.4	Βελτίωση των επιδόσεων (performance) της εφαρμογής.....	90
11.	Βιβλιογραφία	92

Κατάλογος Εικόνων

Εικόνα 2-1: Θεματικά Επίπεδα (layers) δημιουργίας ενός GIS συστήματος.....	13
Εικόνα 2-2: Αναπαράσταση του συστήματος GPS της γης με τους 24 δορυφόρους (4 ανά τροχιά) σε τροχιά.....	14
Εικόνα 3-1: Γενική εικόνα της ροής των AIS δεδομένων	16
Εικόνα 3-2: Συσκευή αναμετάδοσης δεδομένων Τύπου Α.....	17
Εικόνα 3-3: Συσκευή αναμετάδοσης δεδομένων Τύπου Β.....	17
Εικόνα 3-4: Συσκευή αναμετάδοσης SART.....	18
Εικόνα 3-5: Επισκόπηση AIS συστήματος.....	19
Εικόνα 4-1: Τα 4 βασικά σύνολα της αρχιτεκτονικής του .NET Framework.....	22
Εικόνα 4-2: Παραδείγματα έγκυρων Linestring αντικειμένων, χωρικών δεδομένων.....	25
Εικόνα 4-3: Παραδείγματα έγκυρων CircularString αντικειμένων, χωρικών δεδομένων.....	25
Εικόνα 4-4: Παραδείγματα έγκυρων CompoundCurve αντικειμένων, χωρικών δεδομένων.....	26
Εικόνα 4-5: Παραδείγματα έγκυρων Polygon αντικειμένων, χωρικών δεδομένων.....	26
Εικόνα 4-6: Παραδείγματα έγκυρων MultiLineString αντικειμένων, χωρικών δεδομένων.....	27
Εικόνα 4-7: Παραδείγματα έγκυρων MultiPolygon αντικειμένων, χωρικών δεδομένων.....	27
Εικόνα 4-8: Η γεωμετρική ιεραρχία στην οποία βασίζονται τα γεωμετρικού και γεωγραφικού τύπου χωρικά δεδομένα.....	28
Εικόνα 4-9: Σύνταξη JSON εντολής αντικειμένου (object).....	30
Εικόνα 4-10: Σύνταξη JSON εντολής πίνακα (array).....	30
Εικόνα 5-1: Η εφαρμογή MarineTraffic.....	32
Εικόνα 5-2: Η εφαρμογή VesselFinder.....	33
Εικόνα 5-3: Η εφαρμογή VT Explorer.....	34
Εικόνα 6-1: Σχισιακό μοντέλο Βάσης Δεδομένων της εφαρμογής.....	37
Εικόνα 6-2: Αρχική εικόνα AIS δεδομένων (dataset) που χρησιμοποιήθηκαν στην εφαρμογή.....	38
Εικόνα 6-3: Εικόνα των εισαγμένων AIS δεδομένων στη Βασή Δεδομένων.....	38
Εικόνα 6-4: Τελική μορφή των AIS δεδομένων που χρησιμοποιήθηκαν στην εφαρμογή.....	40
Εικόνα 6-5: Δεδομένα του πίνακα tblImisPoolData.....	42
Εικόνα 6-6: Δεδομένα του πίνακα tblLiveShipData.....	42
Εικόνα 6-7: Δεδομένα του πίνακα tblLiveShipDataLineString.....	43
Εικόνα 6-8: Δεδομένα του πίνακα tblZonesOfInterest.....	43
Εικόνα 6-9: Δεδομένα του πίνακα tblZonesOfInterestCoordinates.....	44
Εικόνα 6-10: Δεδομένα του πίνακα tblShips.....	44
Εικόνα 6-11: Δεδομένα του πίνακα tblNotifications.....	45
Εικόνα 6-12: Δεδομένα του πίνακα tblShipType.....	45
Εικόνα 6-13: Η λίστα των Store Procedures που δημιουργήθηκαν και χρησιμοποιήθηκαν στην παρούσα εφαρμογή.....	46
Εικόνα 6-14: Διάγραμμα λειτουργίας της Store Procedure SM_InitializeImisPoolData.....	49
Εικόνα 7-1: Διάγραμμα αρχιτεκτονικής της εφαρμογής VTS.....	52
Εικόνα 8-1: Καρτέλα reference αρχείων που χρησιμοποιήθηκαν στην εφαρμογή μέσα στο Visual Studio.....	64
Εικόνα 8-2: Εικόνα δημόσιων (public) φακέλων και αρχείων της εφαρμογής Dropbox που χρησιμοποιήθηκαν στην παρούσα εφαρμογή ως εικονίδια.....	66
Εικόνα 8-3: Το public link που χρησιμοποιήθηκε για τον εντοπισμό των επιθυμητών αρχείων.....	66
Εικόνα 8-4: Δείγμα εικονιδίων που χρησιμοποιήθηκε στην παρούσα εφαρμογή για την απεικόνιση των πλοίων, ανάλογα με την πορεία τους σε μοίρες.....	68
Εικόνα 9-1: Οθόνη σύνδεσης χρήστη.....	71
Εικόνα 9-2: Μήνυμα λάθους εισαγωγής χρήστη.....	71
Εικόνα 9-3: Γενική εικόνα εφαρμογής.....	72

Εικόνα 9-4: Επιλογή εισαγωγής Λιμανιών και Ζωνών Ενδιαφέροντος.....	73
Εικόνα 9-5: Επιλογή φίλτρων εμφάνισης θέσης πλοίων.....	73
Εικόνα 9-6: Εμφάνιση μηνύματος λάθους στη περίπτωση μη επιλογής πλοίων	73
Εικόνα 9-7: Εμφάνιση φίλτρων εμφάνισης θέσης πλοίων ή διαδρομών πλοίων.....	74
Εικόνα 9-8: Εμφάνιση διαδρομής επιλεγμένων πλοίων και του αντίστοιχου InfoWindow	74
Εικόνα 9-9: Εμφάνιση λιμανιών, περιοχών ενδιαφέροντος και θέσης πλοίων στον χάρτη.....	75
Εικόνα 9-10: InfoWindow με τα χαρακτηριστικά του λιμανιού του Πειραιά	75
Εικόνα 9-11: InfoWindow πλοίου και η πληροφορία συγκεκριμένου στίγματος.....	76
Εικόνα 9-12: Η συνολική εικόνα της εφαρμογής με εμφάνιση στον χάρτη όλων των επιλεγμένων από το φίλτρο πλοίων και το InfoWindow του επιλεγμένου στον χάρτη πλοίου	77
Εικόνα 9-13: Πολλαπλά φίλτρα επιλογής Ονόματος, Προορισμού, Τύπου και Εθνικότητας πλοίων.....	78
Εικόνα 9-14: Φίλτρα επιλογής ημερομηνίας	78
Εικόνα 9-15: Εφαρμογή επιλεγμένων φίλτρων στον χάρτη	79
Εικόνα 9-16: Επιλογή συγκεκριμένων πλοίων καθώς και συγκεκριμένων διαδρομών (legs).....	80
Εικόνα 9-17: Information Tab: Εμφανίζει ειδοποιήσεις για τα πλοία η διαδρομή των οποίων έχει εισέλθει μέσα από κάποια ζώνη ενδιαφέροντος.....	80
Εικόνα 9-18: Το ακριβές σημείο στο οποίο το πλοίο εισήλθε στην ζώνη ενδιαφέροντος.....	81
Εικόνα 9-19: Το InfoWindow του σημείου εισαγωγής ενός πλοίου σε μια ζώνη ενδιαφέροντος.....	81
Εικόνα 9-20: Πολλαπλή επιλογή των ειδοποιήσεων και εμφάνιση των σημείων εισόδου όλων των πλοίων	82
Εικόνα 9-21: Μήνυμα διαγραφής ειδοποίησης.....	83
Εικόνα 9-22: Διαγραφή της ειδοποίησης από το Notification Tab.....	83
Εικόνα 9-23: Administrator Tab	84
Εικόνα 9-24: Καρτέλα διαχείρισης χρηστών (Manage Users).....	85
Εικόνα 9-25: Καρτέλα διαχείρισης Χωρών (Manage Countries)	86
Εικόνα 9-26: Καρτέλα διαχείρισης λιμανιών (Manage Ports).....	87
Εικόνα 9-27: Καρτέλα διαχείρισης πλοίων (Manage Vessels)	88
Εικόνα 9-28: Διαχείριση του χρόνου ανανέωσης του χάρτη από τον administrator.....	88
Εικόνα 9-29: Εμφάνιση χρόνου που απομένει για ανανέωση του χάρτη.....	89
Εικόνα 10-1: Λειτουργικό πρόβλημα στη περίπτωση εμφάνισης πολλαπλής διαδρομής περισσότερων του ενός πλοίων	90

1. Εισαγωγή

1.1 Αντικείμενο Διπλωματικής εργασίας

Στη παρούσα διπλωματική εργασία θα ασχοληθούμε με την ανάπτυξη ενός λογισμικού μέσο του οποίου μια ναυτιλιακή εταιρία θα μπορεί να παρακολουθήσει (tracking) και να λαμβάνει διάφορες πληροφορίες σχετικά με τον στόλο της ανεξαρτήτως τύπου πλοίων (διαχείριση δεδομένων και απεικόνιση).

Στο συγκεκριμένο επιχειρηματικό κλάδο ο όρος διαχείριση στόλου είναι αρκετά πολύπλευρος. Μια ναυτιλιακή εταιρία μπορεί να διαχειρίζεται το στόλο των πλοίων της χρησιμοποιώντας κοινές αλλά πολύπλοκες εφαρμογές όπως το excel αλλά μπορεί και να χρησιμοποιεί διάφορα εξειδικευμένα και προηγμένα λογισμικά μέσο των οποίων δημιουργούνται αυτοματοποιημένες διαδικασίες. Τμήματα τέτοιων προγραμμάτων είναι το Document Management (για διαχείριση εργασιών επάνω στο πλοίο) το Purchasing (για διαχείριση παραγγελιών), το Planned Maintenance (για διαχείριση εργασιών επάνω στο πλοίο) το Crewing (διαχείριση πληρωμάτων) κ.α. Βέβαια οι μεγάλες εταιρίες χρησιμοποιούν ένα συνδυασμό των παραπάνω ώστε να έχουν ένα πλήρες αυτοματοποιημένο management του στόλου τους ενώ χρησιμοποιούν και αποκλειστικές εφαρμογές τις οποίες έχουν σχεδιάσει και δημιουργήσει προσαρμόζοντας αυτές στις απαιτήσεις τους.

Όλες οι παραπάνω εφαρμογές είναι τμήματα μιας συνολικής διαχείρισης την οποία μπορεί και πρέπει να εφαρμόζει η κάθε ναυτιλιακή εταιρία για να είναι σε θέση να ακολουθήσει τα διάφορα ναυτιλιακά πρότυπα αλλά και για να ελέγχει τον στόλο της και να αναλύει αποδοτικότερα τα δεδομένα που δημιουργούνται σε σχέση με αυτόν, χρησιμοποιώντας αναγνωρισμένες και αποδοτικότερες διεργασίες.

Το λογισμικό το οποίο θα υλοποιήσουμε στη παρούσα εργασία αποτελεί ένα τμήμα διαχείρισης το οποίο δεν αναφέρεται παραπάνω. Σχετίζεται όμως με το κομμάτι της διαχείρισης και παρακολούθησης του στόλου που ανήκει στην εκάστοτε ναυτιλιακή εταιρία, δηλαδή της οπτικοποίησης του στίγματος των καραβιών της στο παγκόσμιο χάρτη, της μελλοντικής τους θέσης σε αυτόν, την αναφορά διαφόρων στατιστικών για το κάθε πλοίο και αρκετά άλλα στοιχεία τα οποία μπορούν να χρησιμοποιηθούν για καλύτερη, οικονομικότερη και ασφαλέστερη διαχείριση του στόλου και τα οποία μια ναυτιλιακή εταιρία οφείλει να γνωρίζει. Είναι δηλαδή ένα σύστημα Γεωγραφικής Πληροφορίας το οποίο χρησιμοποιούμε κατάλληλα ώστε να εξάγουμε χρήσιμα συμπεράσματα.

Για να υλοποιηθούν τα παραπάνω θα χρησιμοποιηθεί συγκεκριμένος χάρτης για την απεικόνιση των στιγμάτων των πλοίων καθώς και χωρική βάση δεδομένων ώστε να είμαστε σε θέση να καταχωρίσουμε χωρικά δεδομένα αλλά και να αναλύσουμε αρκετά από αυτά με κατάλληλο τρόπο.

Ως αρχικός στόχος ορίζεται μέσω του λογισμικού αυτού, η εταιρία να είναι σε θέση να αναγνωρίζει τα σημεία στο χάρτη στα οποία βρίσκονται όλα τα πλοία του στόλου της κάθε χρονική στιγμή καθώς και τις διάφορες τροχιές (διαδρομές) τις οποίες έχουν καλύψει κατά τη διάρκεια των ταξιδιών που έχουν υλοποιήσει. Επομένως θα είναι σε θέση να αναγνωρίζει την κατεύθυνση των συγκεκριμένων πλοίων ανά συγκεκριμένες χρονικές στιγμές καθώς και τα ενδιαμέσα λιμάνια που τυχόν υπάρχουν κατά την διάρκεια του ταξιδιού.

Στη συνέχεια θα προστεθούν χαρακτηριστικά σχετικά με την καλύτερη διαχείριση του στόλου σε σχέση με τρέχοντες τιμές ανάλυσης σχετικών παραγόντων όπως ταχύτητα πλοίου, εκτίμηση χρόνου άφιξης σε προορισμό και άλλα που θα δούμε στη συνέχεια.

Επίσης θα οριστούν και περιοχές αυξημένου ενδιαφέροντος πάνω στο χάρτη στις οποίες όταν κάποιο πλοίο εισέρχεται θα ειδοποιείται ο χρήστης του γραφείου ή άλλοι εμπλεκόμενοι ώστε να ενεργούν αναλόγως. Ακόμα θα μπορούν να εισαχθούν και διάφορων ειδών ειδοποιήσεις προς τον τελικό χρήστη ώστε να προληφθούν/αποφευχθούν διάφορες αρνητικές καταστάσεις.

Οι Ζώνες Ενδιαφέροντος (Zones of Interest) αποτελούν τα θαλάσσια κομμάτια μέσα στα οποία όταν τα πλοία εισέρχονται θα πρέπει η εταιρία να έχει ακριβή γνώση για αυτό. Στην συγκεκριμένες περιοχές τα πλοία θα πρέπει είτε όσο γίνεται να τις αποφεύγουν είτε να μειώνουν ταχύτητα ανάλογα με το είδος της ζώνης.

Οι Ζώνες Ενδιαφέροντος πρακτικά για το χρήστη θα μπορούσε να είναι όλα τα παρακάτω:

- Ζώνες υψηλού πειρατικού κινδύνου
- Θαλάσσιες περιοχές προστασίας οικοσυστήματος.
- Θαλάσσιες περιοχές έρευνας και διάσωσης.
- Ζώνες που πρέπει τα πλοία να αποφεύγουν για τον έλεγχο της ρύπανσης
- Περιοχές υπό το καθεστώς διαχείρισης καταστροφών
- Περιοχές που λαμβάνουν χώρα υποβρύχιες εργασίες
- Περιοχές με υψηλή επικινδυνότητα λόγω πάγου
- Θαλάσσιες περιοχές στρατιωτικών επιχειρήσεων
- Περιοχές που έχει γίνει κάποια βύθιση πλοίων ή σύγκρουση πλοίων ή αναλόγων συμβάντων
- Περιοχές αναμονής για διαφορετικού τύπου πλοία (παγοθραυστικά, ρυμουλκά, πλοηγίδες)

1.2 Δομή της διπλωματικής εργασίας

Η παρούσα διπλωματική εργασία αποτελείται από το παρόν εισαγωγικό κεφάλαιο καθώς και από άλλα δέκα κεφάλαια το περιεχόμενο των οποίων περιγράφεται παρακάτω.

Στο 2^ο κεφάλαιο γίνεται αναφορά και περιγραφή των Συστημάτων Γεωγραφικών Πληροφοριών (GIS), στο Παγκόσμιο Σύστημα Εντοπισμού Θέσης (GPS) το οποίο είναι προαπαιτούμενο για ένα GIS σύστημα καθώς και στα χωρικά δεδομένα τα οποία επίσης είναι αλληλένδετα με τα GIS συστήματα.

Στο 3^ο κεφάλαιο γίνεται αναφορά στο σύστημα AIS (Automatic Identification System) και στα δεδομένα που σχετίζονται με αυτό. Επίσης επεξηγείται η λειτουργία του ,η ροή και οι τύποι των δεδομένων καθώς και τα συστήματα και συσκευές που το αποτελούν.

Στο 4^ο κεφάλαιο γίνεται αναφορά και επεξήγηση των εργαλείων ανάπτυξης και υλοποίησης που χρησιμοποιήθηκαν στη παρούσα εφαρμογή που αναπτύχθηκε ενώ παρουσιάζεται και ένα βασικό θεωρητικό υπόβαθρο των τεχνολογιών αυτών. Επίσης γίνεται ανάλυση και παρουσίαση των γεωγραφικών δεδομένων του SQL Server τα οποία χρησιμοποιούνται στην συνέχεια στην υλοποίηση της εφαρμογής.

Στο 5^ο κεφάλαιο παρουσιάζονται αντίστοιχες υλοποιημένες εφαρμογές τις οποίες συναντάμε ήδη στο εμπόριο δωρεάν ή επί πληρωμή. Αναλύεται η λειτουργικότητα των εφαρμογών, οι υπηρεσίες που προσφέρουν στον χρήστη ενώ παρουσιάζεται και σχετικός συγκριτικός πίνακας των εφαρμογών σχετικά με τα κυριότερα χαρακτηριστικά αντίστοιχων εφαρμογών.

Στο 6^ο κεφάλαιο αναλύεται ο σχεδιασμός, η ανάλυση και η δομή της βάσης δεδομένων της εφαρμογής. Επίσης παρουσιάζεται το εννοιολογικό και σχεσιακό σχήμα της βάσης δεδομένων ενώ γίνεται και εκτενής ανάλυση και περιγραφή των βασικών πινάκων και Store Procedures της υλοποιημένης εφαρμογής της παρούσας εργασίας.

Στο 7^ο κεφάλαιο παρουσιάζεται η σχεδίαση και η υλοποίηση της εφαρμογής καθώς και η συνολική αρχιτεκτονική του συστήματος. Επίσης δίνεται και το διάγραμμα του συνολικού μοντέλου της εφαρμογής με όλη τη ροή των δεδομένων και της πληροφορίας.

Στο 8^ο κεφάλαιο περιγράφονται τα κυριότερα τμήματα υλοποίησης της εφαρμογής. Επίσης γίνεται παρουσίαση και ανάλυση των κυριότερων κομματιών κώδικα που χρησιμοποιήθηκε κατά την ανάπτυξη και συνολική περιγραφή των λύσεων που δόθηκαν στα κυριότερα προβλήματα που αντιμετωπίστηκαν.

Στο 9^ο κεφάλαιο αναπτύσσεται η περιγραφή της λειτουργικότητας της εφαρμογής, η περιγραφή των οθονών της εφαρμογής με εκτενή αναφορά στις λειτουργίες που έχει στη διάθεση του ο χρήστης μέσω αυτής καθώς και η παρουσίαση της εφαρμογής μέσω αρκετών εικόνων (screenshots) της.

Στο 10^ο κεφάλαιο αναλύονται κάποιες ιδέες για μελλοντικές επεκτάσεις και βελτιώσεις που θα μπορούσαν να υλοποιηθούν στη παρούσα κατάσταση της εφαρμογής

Στο 11^ο κεφάλαιο παρουσιάζεται η βιβλιογραφία από την οποία αντλήθηκαν διάφορες πληροφορίες για την επίλυση όλων των προβλημάτων που εμφανίστηκαν καθώς και για την κατανόηση μέρους της θεωρίας πάνω στην οποία βασίστηκε η συγκεκριμένη διπλωματική εργασία.

2. Συστήματα Γεωγραφικών Πληροφοριών (GIS) & Παγκόσμιο Σύστημα Εντοπισμού Θέσης (GPS)

2.1 Τι είναι το GIS

Το Σύστημα Γεωγραφικών Πληροφοριών (ΣΓΠ), γνωστό ευρέως και ως GIS (Geographic Information systems) είναι σύστημα διαχείρισης χωρικών δεδομένων (spatial data) και συσχετισμένων ιδιοτήτων. Ο όρος αυτός αναφέρεται σε συστήματα τα οποία έχουν τη δυνατότητα να χειρίζονται γεωγραφικά δεδομένα. Ουσιαστικά τα GIS είναι πληροφοριακά συστήματα (Information systems) που παρέχουν την δυνατότητα συλλογής, διαχείρισης, αποθήκευσης, επεξεργασίας, ανάλυσης και οπτικοποίησης, σε ψηφιακό περιβάλλον, των δεδομένων που σχετίζονται με τον χώρο. Το σύστημα συνολικά δεν περιλαμβάνει μόνο το αντίστοιχο λογισμικό και υλικό που απαιτείτε σε όλα τα συστήματα αλλά επιπροσθέτως περιλαμβάνει και ειδικές συσκευές για την δημιουργία και την απεικόνιση χαρτών αλλά και συστήματα επικοινωνιών που απαιτούνται για να συνδέσουν τα διάφορα συστατικά από τα οποία αποτελούνται.

Σε σύγκριση με τους απλούς χάρτες, ένα σύστημα GIS έχει το πλεονέκτημα ότι η αποθήκευση των δεδομένων γίνεται χωριστά από την αναπαράστασή τους. Αυτό έχει σαν αποτέλεσμα τα ίδια δεδομένα να μπορούν να αναπαρασταθούν με διαφορετικούς τρόπους, π.χ. μπορούμε να μεγεθύνουμε τον (ψηφιακό πλέον) χάρτη, να εμφανίσουμε συγκεκριμένες μόνο περιοχές, να κάνουμε υπολογισμούς αποστάσεων μεταξύ τοποθεσιών, να δημιουργήσουμε πίνακες που να δείχνουν τα διάφορα χαρακτηριστικά του χάρτη καθώς και να απεικονίσουμε επιπλέον πληροφορία πάνω στο χάρτη εάν χρειάζεται.

Όλα τα δεδομένα σε ένα σύστημα GIS είναι γεω-καταχωρημένα, δηλ. συνδεδεμένα με μια συγκεκριμένη γεωγραφική τοποθεσία της επιφάνειας της γης μέσω ενός συστήματος συντεταγμένων. Τα δεδομένα αυτά συνήθως λέγονται γεωγραφικά ή χαρτογραφικά ή χωρικά (spatial) και μπορεί να συσχετίζονται με μια σειρά από περιγραφικά δεδομένα τα οποία και τα χαρακτηρίζουν μοναδικά. Τα συστήματα GIS αποτυπώνουν χωρικά δεδομένα σε γεωγραφικό ή χαρτογραφικό ή καρτεσιανό σύστημα συντεταγμένων.

Βασικό χαρακτηριστικό των GIS είναι ότι τα χωρικά δεδομένα συνδέονται και με περιγραφικά δεδομένα, π.χ. μια ομάδα σημείων που αναπαριστούν θέσεις πόλεων συνδέεται με ένα πίνακα όπου κάθε εγγραφή εκτός από τη θέση περιέχει πληροφορίες όπως ονομασία, πληθυσμός κλπ.

Ένα από τα πιο συνηθισμένα συστήματα γεωγραφικών συντεταγμένων είναι αυτό του γεωγραφικού μήκους και γεωγραφικού πλάτους. Σ' αυτό το σύστημα συντεταγμένων, κάθε τοποθεσία προσδιορίζεται σχετικά με τον ισημερινό και τη γραμμή μηδενικού γεωγραφικού μήκους που περνά από το αστεροσκοπείο Greenwich της Αγγλίας. Υπάρχουν πολλά άλλα γεωγραφικά συστήματα συντεταγμένων και κάθε GIS σύστημα θα πρέπει να μπορεί να μετατρέπει τις συντεταγμένες από το ένα σύστημα στο άλλο.

Η χωρική πληροφορία αναπαρίσταται με δυο τρόπους:

- Ως διανυσματικά δεδομένα με τη μορφή σημείων, γραμμών και πολυγώνων
- Ως δικτυωτά (raster) δεδομένα, οργανωμένα συστηματικά σε κελιά (όπως π.χ. μια ψηφιακή εικόνα).

2.2 Περιγραφή GIS Συστημάτων

Τα βασικά συστατικά μέρη ενός GIS τα οποία πρέπει να βρίσκονται σε ισορροπία και αλληλεξάρτηση είναι τα εξής :

- Το συνολικό εξοπλισμό δηλαδή το υπολογιστικό σύστημα και τα περιφερειακά του (hardware)
- Το λογισμικό - αλγόριθμοι (software)
- Τα δεδομένα (data)
- Οι χρήστες
- Οι μέθοδοι που χρησιμοποιούνται

Το υπολογιστικό σύστημα μπορεί να είναι είτε ένας προσωπικός υπολογιστής ενός χρήστη είτε ένα ισχυρό υπολογιστικό σύστημα ανάλογα την εφαρμογή. Και στις δύο περιπτώσεις το σύστημα όμως θα πρέπει να έχει αρκετά μεγάλη υπολογιστική ισχύ ώστε να ανταπεξέλθει στις απαιτήσεις οι οποίες συνήθως είναι αρκετά υψηλές λόγω του είδους της χωρικής πληροφορίας και των συνήθως μεγάλου όγκου δεδομένων.

Τα περιφερειακά διακρίνονται σε περιφερειακά εισόδου που επιτρέπουν την είσοδο των στοιχείων στο σύστημα, τα περιφερειακά εξόδου που συμμετέχουν στην παρουσίαση και απεικόνιση των στοιχείων καθώς και περιφερειακά διαχείρισης που χρησιμοποιούνται στην αποθήκευση των δεδομένων. Περιφερειακές συσκευές εισόδου μπορεί να είναι ψηφιοποιητές (digitizers), σαρωτές (scanners) και GPS ως data entry devices μέσω των οποίων επιτυγχάνεται η είσοδος χωρικών δεδομένων ενώ για μη χωρικά δεδομένα χρησιμοποιούνται κοινές συσκευές όπως πληκτρολόγια. Περιφερειακές συσκευές εξόδου θεωρούνται τα συστήματα απεικόνισης της πληροφορίας όπως σχεδιογράφοι (plotters) και οθόνες υψηλής ανάλυσης. Τέλος περιφερειακά διαχείρισης δεδομένων θεωρούνται οι σκληροί δίσκοι με μεγάλη αποθηκευτική ικανότητα.

Το λογισμικό πρέπει να έχει τη δυνατότητα να επιτελεί πολύπλοκες και αναλυτικές διαδικασίες. Θα πρέπει να επιτρέπει την ψηφιοποίηση δεδομένων, δηλαδή την εισαγωγή σημείων, γραμμών, πολυγώνων καθώς και την εισαγωγή χαρακτηριστικών ιδιοτήτων και στατιστικών

Επίσης πρέπει να παρέχει τη δυνατότητα της επεξεργασίας των δεδομένων, όπως τον εντοπισμό σφαλμάτων, τον συνδυασμό και την τακτοποίηση των δεδομένων μέσα στην αντίστοιχη βάση καθώς και την ενημέρωση της βάσης με νέα δεδομένα. Η ανάλυση των δεδομένων περιλαμβάνει την εκτέλεση εντολών Boolean δηλαδή ΚΑΙ, Ή και ΟΧΙ (AND, OR και NOT) πάνω σε διαφορετικά επίπεδα δεδομένων, μέτρηση μηκών και επιφανειών και στατιστική επεξεργασία.

Τα δεδομένα που χρησιμοποιούνται στα Γεωγραφικά Συστήματα Πληροφοριών διακρίνονται σε δύο μεγάλες κατηγορίες :

- Τα χωρικά δεδομένα, τα οποία χαρακτηρίζονται αποκλειστικά από τη θέση τους στο χώρο σε σχέση με κάποιο σύστημα συντεταγμένων και διακρίνονται σε τέσσερις βασικές κατηγορίες: Τα σημειακά δεδομένα, τα γραμμικά δεδομένα, τα επιφανειακά δεδομένα και τα δεδομένα αναγλύφου.
- Τα μη χωρικά ή περιγραφικά δεδομένα, τα οποία σχετίζονται ή περιγράφουν τα χαρακτηριστικά ή τις ιδιότητες της υπόψη χωρικής θέσης.

Οι χρήστες αποτελούν εξίσου σημαντικά συστατικά μέρη ενός Γεωγραφικού Συστήματος Πληροφοριών και χωρίζονται στις παρακάτω κατηγορίες. Τους τεχνικούς οι οποίοι σχεδιάζουν και συντηρούν το σύστημα συνολικά, τους μηχανικούς οι οποίοι μέσα από το συγκεκριμένο σύστημα πρέπει να λάβουν κάποια απόφαση σχετική με την εκάστοτε εργασία τους και του απλούς χρήστες οι οποίοι χρησιμοποιούν το σύστημα σαν μέρος της καθημερινότητάς τους συνήθως για απλή πληροφόρηση.

Τέλος οι διάφοροι μέθοδοι είναι οι διαδικασίες μέσα στις οποίες πρέπει να κινείται η εφαρμογή και λειτουργία του συστήματος σύμφωνα με της λειτουργία του οργανισμού που το χρησιμοποιεί.

Ένα σύστημα GIS επιτρέπει πράξεις πάνω σε χωρικά δεδομένα, δηλ. χρησιμοποιώντας γεωγραφικά μήκη και πλάτη. Παράδειγμα μιας τέτοιας πράξης είναι: «Ποιες πόλεις βρίσκονται λιγότερο από 1000 χιλιομέτρων η μία από την άλλη;». Επιτρέπει δηλ. τον προσδιορισμό των χωρικών σχέσεων ανάμεσα στα χαρακτηριστικά (features) του χάρτη. Επιπλέον συνδέει χωρικά δεδομένα με γεωγραφική πληροφορία για

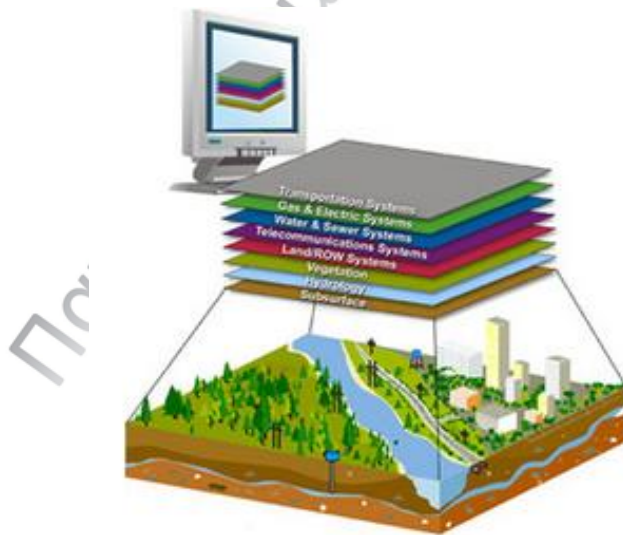
ένα συγκεκριμένο χαρακτηριστικό του χάρτη. Η πληροφορία αποθηκεύεται ως ιδιότητες (attributes) του γραφικά παρουσιαζόμενου χαρακτηριστικού σε μια Βάση Δεδομένων. Για κάθε χαρακτηριστικό αποθηκεύονται τρεις βασικές πληροφορίες στη ΒΔ: η γεωγραφική πληροφορία, η προβολή (projections) πάνω στην οποία εκφράζεται η γεωγραφική πληροφορία και οι ιδιότητές του. Για κάθε χαρακτηριστικό του χάρτη αποθηκεύονται ακόμα στη ΒΔ του GIS οι εξής πληροφορίες: τι χαρακτηριστικό είναι, που βρίσκεται και πως σχετίζεται με άλλα χαρακτηριστικά.

Επίσης υπάρχει και η έννοια του στρώματος (layer) πολλά εκ των οποίων αποτελούν ένα ψηφιακό χάρτη. Δηλαδή ο χάρτης που βλέπει ο τελικός χρήστης ουσιαστικά αποτελείται από πολλαπλά layers το καθένα από τα οποία απεικονίζει και διαφορετικά χαρακτηριστικά. Δηλαδή στη περίπτωση ενός ψηφιακού χάρτη που απεικονίζει τον κόσμο, ένα layer θα μπορούσε να είναι η απεικόνιση των συνόρων των χωρών, άλλο layer η απεικόνιση των σημείων που δηλώνουν την πρωτεύουσα της κάθε χώρας και ούτω καθεξής. Ο χρήστης ουσιαστικά ανάλογα με τα χαρακτηριστικά που τον ενδιαφέρουν μπορεί να επιλέξει να δει μόνο τα στρώματα εκείνα που τον ενδιαφέρουν έτσι ώστε να εστιάσει μόνο στην ζητούμενη κάθε φορά πληροφορία.

Η χωρική πληροφορία αποθηκεύεται με τη μορφή τριών βασικών χαρακτηριστικών: του σημείου, της γραμμής και του πολυγώνου. Η περιγραφική πληροφορία εμφανίζεται με τη μορφή συμβόλων και ετικετών πάνω στο χάρτη. Η δύναμη του συστήματος βασίζεται στη δυνατότητα που έχει να συνδυάζει αυτά τα δυο είδη πληροφορίας.

Τελικά ένα σύστημα GIS σχεδιάζεται ώστε να μπορεί να απαντήσει στους παρακάτω τύπους ερωτήσεων και οι οποίοι είναι δύσκολο να απαντηθούν σε διαφορετικού τύπου συστήματα:

- με βάση την *τοποθεσία*, δηλ. τι βρίσκεται σε μια συγκεκριμένη τοποθεσία;
- με βάση μια *συνθήκη*, δηλ. η εύρεση μιας τοποθεσίας που ικανοποιεί συγκεκριμένες συνθήκες;
- με βάση κάποιες *τάσεις*, δηλ. τι άλλαξε μετά από κάποιο χρονικό διάστημα;
- με βάση κάποια *πρότυπα* (patterns) π.χ. χρονική καθυστέρηση όταν κυριαρχούν συγκεκριμένες εδαφικές συνθήκες βάση κάποια *μοντελοποίηση*, δηλ. «τι θα συνέβαινε εάν» ερωτήσεις.



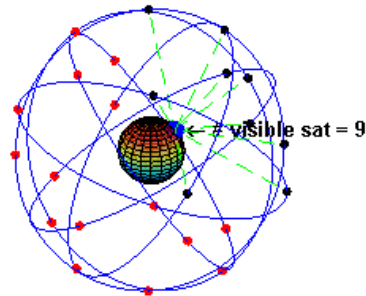
Εικόνα 2-1: Θεματικά Επίπεδα (layers) δημιουργίας ενός GIS συστήματος

2.3 Τι είναι το GPS

Το GPS (Global Positioning System) είναι ένα παγκόσμιο σύστημα εντοπισμού θέσης το οποίο αρχικά ξεκίνησε ως ένα στρατιωτικό δορυφορικό σύστημα πλοήγησης που αναπτύχθηκε από το υπουργείο άμυνας των Η.Π.Α με την ονομασία «NAVSTAR GPS» (Navigation Signal Timing and Ranging Global Positioning System). Βασίζεται σε ένα 'πλέγμα' εικοσιτεσσάρων δορυφόρων μεγάλης ακριβείας της γης που τους επιτρέπουν να εκπέμπουν ραδιοσήματα με μεγάλη ακρίβεια από ειδικές συσκευές που ονομάζονται 'δέκτες GPS'. Οι δέκτες αυτοί παρέχουν ακριβείς πληροφορίες για τη θέση ενός σημείου, το υψόμετρό του, την ταχύτητα και την κατεύθυνση της κίνησης του. Επίσης, σε συνδυασμό με ειδικό λογισμικό χαρτογράφησης μπορούν να απεικονίσουν γραφικά τις πληροφορίες αυτές.

Η θέση στην επιφάνεια της γης προσδιορίζεται από τη λήψη και σύγκριση των σημάτων τριών τέτοιων δορυφόρων η οποία μεταφράζεται στην μοναδική τομή τριών κώνων των οποίων οι κορυφές είναι οι τρεις δορυφόροι.

Η τυπική ακρίβεια μέτρησης του GPS είναι $\pm 100\text{m}$ και μπορεί να φθάσει τα $\pm 10\text{m}$ με τη βοήθεια διαφορικού GPS. Το GPS επιτρέπει τον προσδιορισμό της θέσης σημείων σ' όλο τον κόσμο 24 ώρες το εικοσιτετράωρο κάτω από οποιοσδήποτε καιρικές συνθήκες.



Εικόνα 2-2: Αναπαράσταση του συστήματος GPS της γης με τους 24 δορυφόρους (4 ανά τροχιά) σε τροχιά

Τα συστήματα GIS έχουν μια ευρεία περιοχή εφαρμογών. Τυπικές εφαρμογές τους είναι στη διαχείριση του περιβάλλοντος και των πόρων του, στο σχεδιασμό και ανάπτυξη νέας υποδομής των πόλεων, δρόμων κλπ, στις θαλάσσιες και επίγειες μεταφορές, στην ασφάλεια στη θάλασσα, στις βιομηχανίες τηλεπικοινωνιών, ύδατος, ηλεκτρικού όπου η έμφαση ρίχνεται στη συντήρηση και λειτουργία των δικτύων τους, ακόμα και για στρατιωτικές εφαρμογές.

Τα συστήματα GIS είναι μια ενεργή περιοχή της ψηφιακής τεχνολογίας αφού η ικανότητα των συστημάτων αυτών να αποθηκεύουν σχέσεις ανάμεσα στα χαρακτηριστικά, πέρα από τα ίδια τα χαρακτηριστικά και τις ιδιότητές τους, είναι ένα από τα πιο σημαντικά χαρακτηριστικά ισχύος και ευελιξίας αυτής της τεχνολογίας.

3. Το Αυτόματο Σύστημα Εντοπισμού AIS (Automatic Identification System)

3.1 Λειτουργία του AIS

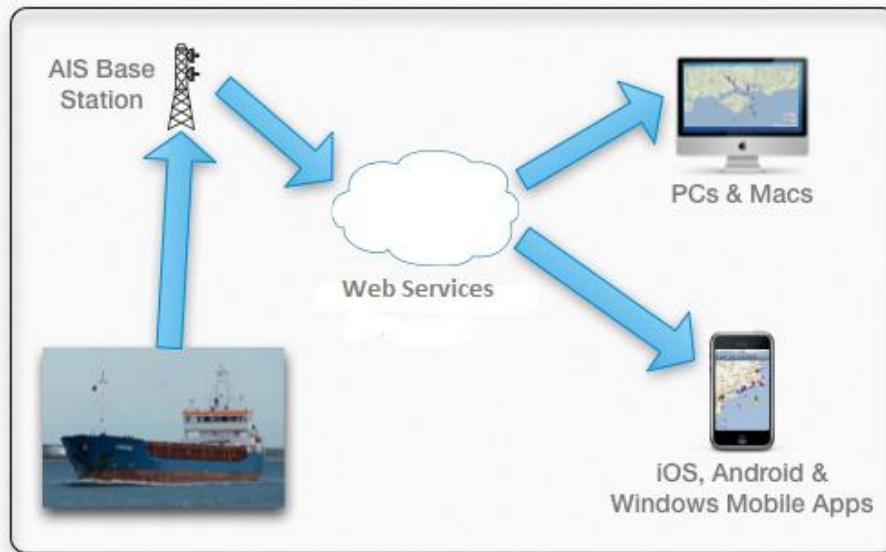
Στη ναυτιλία τα προηγούμενα χρόνια δεν υπήρχε η δυνατότητα καταγραφής της θέσης όλων των πλοίων που κινούνταν στις θάλασσες. Η όποια καταγραφή τους γινόταν από χερσαία συστήματα το οποία σε πάρα πολλά θαλάσσια σημεία ήταν εκτός εμβέλειας και επομένως τα πλοία ήταν ουσιαστικά απροστάτευτα και χαμένα από οποιοδήποτε σύστημα μέχρι τη στιγμή που θα πλησίαζαν σε κάποια στεριά.

Όμως από το Δεκέμβριο του 2004 ο Διεθνής Οργανισμός Ναυτιλίας (IMO) απαιτεί από όλα τα πλοία άνω των 300 τόνων να φέρουν συσκευή AIS (Automatic Identification System), η οποία μεταδίδει περιοδικά τη θέση τους, την ταχύτητα και την πορεία τους, καθώς και ορισμένες στατικές πληροφορίες, όπως το όνομα του σκάφους, τις διαστάσεις και λεπτομέρειες σχετικές με το τρέχον ταξίδι. Το AIS είναι ένα αυτόματο σύστημα εντοπισμού θέσης το οποίο χρησιμοποιείτε για να εντοπίζονται οι θέσεις των πλοίων. Αυτό υλοποιείται μέσω κεντρικών σταθμών συλλογής δεδομένων (AIS base stations) και δορυφόρων, ανταλλάζοντας μεταξύ κοντινών πλοίων δεδομένα. Τα δεδομένα αυτά μαζί με κάποια ναυτικά ραντάρ είναι αυτή τη στιγμή η κύρια μέθοδος αποφυγής ατυχημάτων μεταξύ των πλοίων.

Οι πληροφορίες που παρέχει το σύστημα του AIS είναι ο κωδικός ταυτοποίησης του πλοίου, η θέση του, η κατεύθυνση του και η ταχύτητά του. Περιέχει τόσο σημαντική πληροφορία διότι έχει ως στόχο να βοηθήσει τους διαχειριστές των πλοίων στο στόχο τους καθώς και να επιτρέπει στις επίσημες αρχές να παρακολουθούν και να ελέγχουν τις κινήσεις των πλοίων ανά πάσα στιγμή. Λόγω αυτού είναι πλέον σε θέση να αναγνωρίζουν ξεχωριστά το κάθε πλοίο και να προσδιορίσουν την συγκεκριμένη θέση του καθώς και το σύνολο των κινήσεων που διαγράφει. Επίσης χρησιμοποιώντας αυτά τα δεδομένα υπάρχει πλέον η δυνατότητα της αποτύπωσης της θέσης του πλοίου και των κινήσεών του συνολικά σε εικονικούς χάρτες σε πραγματικούς χρόνους. Αυτά όλα είναι πάρα πολύ σημαντικά αφού όταν ένα πλοίο βρίσκεται στα ανοιχτά της θάλασσας είναι ζωτικής σημασίας να γνωρίζουν οι διαχειριστές του ή και ο καπετάνιος τις κινήσεις και την ταυτότητα των κοντινών του πλοίων ώστε να αποφύγει τυχών συγκρούσεις ή οποιαδήποτε άλλο κίνδυνο μπορεί να προκύψει. Παλαιότερα για τον σκοπό αυτό χρησιμοποιούνταν προληπτικοί μηχανισμοί όπως παρατηρητές (με κιάλια νυχτερινής όρασης), ανταλλαγές ήχων μεταξύ των κοντινών πλοίων και τέλος τα απλά ραντάρ. Όμως όλα τα παραπάνω δεν επιτύχαιναν πάντα τον στόχο τους για διάφορους λόγους όπως καθυστερήσεων ανθρώπινου παράγοντα, ανακρίβειες και διάφορες άλλες δυσλειτουργίες που μπορεί να οδηγούσαν σε κάποια σύγκρουση. Έτσι πλέον λόγω του AIS έχει βελτιωθεί σε πάρα πολύ μεγάλο βαθμό η ασφάλεια και ο έλεγχος στις θαλάσσιες συγκοινωνίες.

Στη συνέχεια θα αναλύσουμε τη λειτουργία του συστήματος AIS και γιατί έχει καταφέρει να είναι τόσο αποδεκτό συνολικά.

Όπως ήδη αναφέρθηκε παραπάνω το AIS παρέχει πληροφορίες στοιχείων ταυτότητας, θέσης και ταχύτητας ανά τακτά χρονικά διαστήματα των πλοίων. Η μετάδοση των πληροφοριών αυτών γίνεται από κάθε ένα πλοίο μέσω ενός ενσωματωμένου VHF πομποδέκτη. Οι πληροφορίες προέρχονται μέσα από διάφορους αισθητήρες ναυσιπλοΐας του πλοίου (συνήθως μέσω του συστήματος παγκόσμιας δορυφορικής πλοήγησης GNSS). Οι πληροφορίες ταυτότητας το όνομα και το διακριτικό ταυτοποίησης του VHF πομπού προγραμματίζονται με την εγκατάσταση του εξοπλισμού να μεταδίδονται ανά τακτά χρονικά διαστήματα. Οι πληροφορίες αυτές ομαδοποιούνται στους πομποδέκτες του AIS και αποστέλλονται στους σταθμούς συγκέντρωσης σημάτων μέσω ενδιάμεσων πλοίων ή απευθείας στην περίπτωση που το πλοίο είναι κοντά σε κάποιο σταθμό. Παρακάτω εμφανίζεται μια γενική μορφή της ροής των AIS δεδομένων.



Εικόνα 3-1: Γενική εικόνα της ροής των AIS δεδομένων

3.2 Τύποι και συσκευές του AIS

Το AIS διακατέχεται από κάποιο συγκεκριμένο πρότυπο το οποίο περιλαμβάνει αρκετά υπό-πρότυπα τα οποία ονομάζονται τύποι (types). Για κάθε τύπο παρέχονται αρκετές λεπτομερείς τεχνικές προδιαγραφές οι οποίες διασφαλίζουν την συνολική ακεραιότητα του παγκοσμίου συστήματος AIS εντός της οποίας όλοι οι τύποι αυτού πρέπει να λειτουργούν. Οι κυριότεροι τύποι του AIS συστήματος είναι οι παρακάτω:

3.2.1 Τύπος A (Class A)

Εμπεριέχει ένα πομποδέκτη AIS ο οποίος είναι τοποθετημένος στο πλοίο, μπορεί να κάνει εκπομπή και λήψη και λειτουργεί χρησιμοποιώντας την μέθοδο SOTDMA (Self-Organized Time Division Multiple Access). Η συγκεκριμένη μέθοδος είναι μέθοδος πρόσβασης καναλιών επικοινωνίας που σχεδιάστηκε από τον Hakan Lans και βασίζεται στην πολυπλεξία διαίρεσης του χρόνου. Δηλαδή η μέθοδος αυτή δίνει τη δυνατότητα μετάδοσης και λήψης ανεξαρτήτων σημάτων μέσω μιας κοινής διαδρομής σημάτων με τη χρήση συγχρονισμένων διακοπών κατάλληλα σε κάθε άκρο της γραμμής μετάδοσης.

Ο τύπος αυτός απευθύνεται σε μεγάλα εμπορικά πλοία. Απαιτεί έναν πομποδέκτη με εσωτερική μνήμη στην οποία θα έχει την δυνατότητα να διατηρήσει μέρος της πληροφορίας που πρόκειται να στείλει ώστε να έχει γνώση εκ των προτέρων για αυτή. Αυτό γίνεται ώστε να μπορεί να προ-ανακοινώσει την μετάδοση της πληροφορίας ώστε να έχει προτεραιότητα στη λήψη από το σύστημα του AIS. Αυτό επιτυγχάνεται μέσω δύο δεκτών σε συνεχή λειτουργία. Τα συστήματα αυτού του τύπου λαμβάνουν όλες της κατηγορίες των AIS μηνυμάτων με ρυθμό μετάδοσης προεπιλεγμένα (by default) ανά λίγα δευτερόλεπτα ενώ το κόστος τους ξεπερνάει τα 1500 ευρώ



Εικόνα 3-2: Συσκευή αναμετάδοσης δεδομένων Τύπου A

3.2.2 Τύπος B (Class B)

Τα συστήματα αυτού του τύπου χρησιμοποιούν επίσης ένα πομποδέκτη AIS για εκπομπή και λήψη χρησιμοποιώντας είτε την μέθοδο SOTDMA όπως προηγουμένως είτε την μέθοδο CSTDMA (Carrier Sense Time Division Multiple Access). Συγκεκριμένα η λήψη των μηνυμάτων προσδιορίζεται από το χρόνο που αποστέλλονται τα μηνύματα της κατηγορίας A ή ανάλογα με το εύρος εμβέλειας των σταθμών AIS. Παρακολουθώντας συνεχώς το επίπεδο του θορύβου των AIS ράδιο καναλιών. Αυτό το επίπεδο χρησιμοποιείται ως μονάδα αναφοράς για την ισχύ του λαμβανόμενου σήματος στην αρχή κάθε slot. Κατά την μετάδοση απαιτείται ένα TDMA slot τυχαία και μετριέται η ισχύς του σήματος στην αρχή του slot. Εάν η ισχύς του σήματος είναι σημαντικά πάνω από το επίπεδο του θορύβου η υποδοχή αυτή θεωρείται ότι είναι σε χρήση και η μετάδοση αναβάλλεται. Αντίθετα αν η ισχύς του σήματος είναι κοντά στο επίπεδο background το slot αυτό θεωρείται ότι είναι ανενεργό και η μετάδοση πραγματοποιείται κανονικά.

Τα συστήματα αυτού του τύπου απευθύνονται σε ελαφρύτερα εμπορικού και ψυχαγωγικού χαρακτήρα πλοία. Ο by default χρόνος μετάδοσης είναι κάθε 30 δευτερόλεπτα. Αυτό βέβαια ποικίλει ανάλογα με την ταχύτητα του σκάφους ή τυχόν οδηγίες που έχουν ληφθεί από τους σταθμούς βάσης AIS. Τέλος τα συστήματα αυτά απαιτούν μια ενσωματωμένη συσκευή GPS και ορισμένα LEDs ενώ έχουν την δυνατότητα να λάβουν όλες τις κατηγορίες μηνυμάτων AIS ενώ το κόστος τους φτάνει τα 740 ευρώ.



Εικόνα 3-3: Συσκευή αναμετάδοσης δεδομένων Τύπου B

3.2.3 AtoN (Aids to Navigation)

Είναι πομποδέκτες για εκπομπή και λήψη σημάτων που λειτουργούν μέσω του FATDMA (Fixed-access time-division multiple-access). Σχεδιάστηκαν για να λαμβάνουν και να αποστέλλουν δεδομένα σχετικά με τις καιρικές συνθήκες καθώς και τις συνθήκες της θάλασσας. Επίσης μια άλλη σημαντική λειτουργία που επιτελούν είναι και η αναμετάδοση των μηνυμάτων του AIS δικτύου ώστε να καλύπτει αυτό περισσότερες περιοχές.

3.2.4 SART (Search and Rescue Transponder)

Η συγκεκριμένη συσκευή έχει σχεδιαστεί ειδικά για το AIS σύστημα. Χρησιμοποιεί τη μέθοδο PATDMA (re-announce time-division multiple-access) καθώς και μια τροποποιημένη PATDMA μέθοδο.

Η συσκευή επιλέγει τυχαία ένα slot για να μεταδώσει μια σειρά από οκτώ μηνυμάτων παράλληλα ανά λεπτό για να μεγιστοποιήσει την πιθανότητα επιτυχούς μετάδοσης. Επίσης αυτού του είδους οι συσκευές έχουν την δυνατότητα να μεταδίδουν μηνύματα σε απόσταση μεγαλύτερη των πέντε μιλίων. Τα μηνύματα αυτά έχουν συγκεκριμένη μορφή που αναγνωρίζεται μόνο από τις άλλες AIS συσκευές. Τέλος οι SART συσκευές σχεδιάστηκαν για να χρησιμοποιούνται περιοδικά και όχι συνεχόμενα σε καταστάσεις έκτακτης ανάγκης λόγω

της μεθόδου RATDMA που χρησιμοποιούν η οποία αυξάνει πάρα πολύ την κίνηση των μηνυμάτων λόγω της πολλαπλής αποστολής των μηνυμάτων.



Εικόνα 3-4: Συσκευή αναμετάδοσης SART

3.2.5 Specialist AIS Transponders

Παρά το γεγονός ότι υπάρχουν δημοσιευμένες συγκεκριμένες προδιαγραφές για τις AIS συσκευές πολλές ναυτιλιακές αρχές επιτρέπουν και ενθαρρύνουν παράλληλα την δημιουργία υβριδικών AIS συσκευών. Όπως συμβαίνει σε τέτοιες περιπτώσεις οι συσκευές αυτές δημιουργούνται ώστε να καλύψουν κάποια επιπλέον χαρακτηριστικά και λειτουργίες που απαιτούνται ώστε να καλυφθούν συγκεκριμένες ανάγκες. Αυτές οι συσκευές από την άλλη, θα πρέπει να ακολουθούν συγκεκριμένα πρότυπα ώστε να διατηρηθεί η ακεραιότητα και η αξιοπιστία του υπάρχοντος συστήματος καθώς και η δομή της μετάδοσης και του συνολικού σχεδιασμού του συστήματος. Μια τέτοια συσκευή θεωρείται ο πομποδέκτης 'Identifier AIS'. Ουσιαστικά είναι τύπου κλάσης B χρησιμοποιώντας την μέθοδο CSTDMA αλλά έχει σχεδιαστεί με αρκετές μετατροπές από της επίσημες συσκευές κλάσης B. Τα διαφορετικά χαρακτηριστικά που έχουν προστεθεί είναι η δυνατή μπαταρία που έχει χρησιμοποιηθεί το χαμηλότερο κόστος, η ευκολότερη διαδικασία τοποθέτησης καθώς και η μαζικότερη κατασκευή τους. Βέβαια αυτές οι συσκευές δεν μπορούν να λάβουν την διεθνή πιστοποίηση ότι ακολουθούν τα σωστά διεθνή πρότυπα αλλά θα περάσουν από τεχνικές αξιολόγησης και δοκιμές για να διασφαλιστεί ότι η λειτουργία του δεν βλάπτει τη λειτουργία του συνολικού υπάρχοντος συστήματος AIS

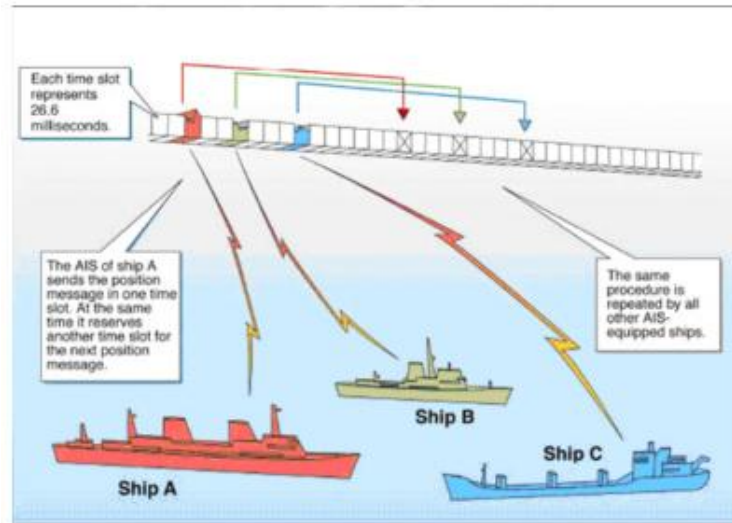
3.2.6 AIS Base Stations

Είναι σταθμοί με κατάλληλο εξοπλισμό οι οποίοι λαμβάνουν τα μηνύματα που αποστέλλουν οι παραπάνω κατηγορίες. Πραγματοποιούν εκπομπή και λήψη σημάτων και λειτουργούν χρησιμοποιώντας το πρωτόκολλο SOTDMA (Self-Organized Time Division Multiple Access) με δυνατότητα περίπου 4000 αναφορών το λεπτό. Οι σταθμοί βάσεις AIS έχουν ένα σύνθετο σύνολο χαρακτηριστικών και λειτουργιών και είναι σε θέση να ελέγξουν το AIS σύστημα καθώς και όλες τις συσκευές που λειτουργούν εκεί. Επίσης έχουν την δυνατότητα να κάνουν μεμονωμένες αναμεταδώσεις σημάτων για διάφορες αναφορές κατάστασης καθώς και για μεταδώσεις αλλαγών συχνότητας.

Ειδικότερα κάθε σταθμός βάσης είναι εξοπλισμένος με έναν δέκτη AIS, έναν ηλεκτρονικό υπολογιστή και μια σύνδεση στο Internet. Ο δέκτης AIS λαμβάνει δεδομένα, τα οποία υποβάλλονται σε επεξεργασία από ένα απλό λογισμικό στον υπολογιστή και στη συνέχεια αποστέλλονται σε μια κεντρική βάση δεδομένων μέσω ενός "web service". Αυτό το λογισμικό είναι ελεύθερο για όσους ενδιαφέρονται, με αδεία χρήσης GNU. Τα

δεδομένα που λαμβάνονται από τον δέκτη AIS είναι κωδικοποιημένα σε μηνύματα NMEA (6-bit απλό κείμενο).

Ένα παράδειγμα μηνύματος είναι το παρακάτω: !AIVDM,1,1,,B,1INS<8@P001cnWFEdSmh00bT0000,0*38



Εικόνα 3-5: Επισκόπηση AIS συστήματος

3.3 AIS Δεδομένα

Τα μηνύματα AIS περιλαμβάνουν τους παρακάτω βασικούς τύπους πληροφορίας:

- Δυναμική πληροφορία, όπως η θέση του πλοίου, η ταχύτητα, η πορεία, και η ταχύτητα στροφής.
- Στατική πληροφορία, όπως το όνομα του πλοίου, ο αριθμός IMO, ο αριθμός MMSI και οι διαστάσεις του.
- Πληροφορίες που σχετίζονται με το συγκεκριμένο ταξίδι που εκτελεί, όπως προορισμός, χρόνος εκτιμώμενης άφιξης (ETA) και βύθισμα.

Υπάρχουν είκοσι επτά (27) διαφορετικού τύπου και επιπέδου μηνύματα τα οποία μπορούν να σταλθούν από τους AIS πομποδέκτες τα οποία βλέπουμε στον παρακάτω πίνακα. Υπάρχουν συγκεκριμένα μηνύματα για την κλάση A και συγκεκριμένα μηνύματα για την κλάση B. Επίσης είναι διαφορετικά τα μηνύματα που απευθύνονται στις άλλες κατηγορίες των συσκευών (AtoN, SART, Specialist AIS Transponders) που χρησιμοποιούν το AIS σύστημα. Αυτά τα μηνύματα καθορίζονται από την NMEA (National Marine Electronics Associations)

Συγκεκριμένα τα μηνύματα με id 6,8,25 και 26 είναι ιδιωτικά (κρυφά) και παρέχουν ASM (Application Specific Messages) δεδομένα τα οποία χρησιμοποιούνται από τις αρμόδιες αρχές να καθορίσουν υπό κατηγορίες άλλων AIS μηνυμάτων. Αντίθετα τα μηνύματα που περιέχουν MMSI κωδικό προορισμού δεν είναι ιδιωτικά και μπορούν να ληφθούν και να αποκωδικοποιηθούν από οποιονδήποτε δέκτη.

Message ID	Name	Description	Priority	Access scheme	Communication state	Transmitted by mobile station (M)/base station (B)
1	Position report	Scheduled position report; (Class A shipborne mobile equipment)	1	SOTDMA, RATDMA, ITDMA(1)	SOTDMA	M
2	Position report	Assigned scheduled position report; (Class A shipborne mobile equipment)	1	SOTDMA(9)	SOTDMA	M
3	Position report	Special position report, response to interrogation; (Class A shipborne mobile equipment)	1	RATDMA(1)	ITDMA	M
4	Base station report	Position, UTC, date and current slot number of base station	1	FATDMA(3) (7), RATDMA(2)	SOTDMA	B
5	Static and voyage related data	Scheduled static and voyage related vessel data report; (Class A shipborne mobile equipment)	4(5)	RATDMA, ITDMA(2)	N/A	M
6	Binary addressed message	Binary data for addressed communication	4	RATDMA(10), FATDMA, ITDMA(2)	N/A	M/B
7	Binary acknowledgement	Acknowledgement of received addressed binary data	1	RATDMA, FATDMA, ITDMA(2)	N/A	M/B
8	Binary broadcast message	Binary data for broadcast communication	4	RATDMA(10), FATDMA, ITDMA(2)	N/A	M/B
9	Standard SAR aircraft position report	Position report for airborne stations involved in SAR operations, only	1	SOTDMA, RATDMA, ITDMA(1)	SOTDMA ITDMA	M
10	UTC/date inquiry	Request UTC and date	3	RATDMA, FATDMA, ITDMA(2)	N/A	M/B
11	UTC/date response	Current UTC and date if available	3	RATDMA, ITDMA(2)	SOTDMA	M
12	Addressed safety related message	Safety related data for addressed communication	2	RATDMA(10), FATDMA, ITDMA(2)	N/A	M/B
13	Safety related acknowledgement	Acknowledgement of received addressed safety related message	1	RATDMA, FATDMA, ITDMA(2)	N/A	M/B
14	Safety related broadcast message	Safety related data for broadcast communication	2	RATDMA(10), FATDMA, ITDMA(2)	N/A	M/B
15	Interrogation	Request for a specific message type (can result in multiple responses from one or several stations)	3	RATDMA, FATDMA, ITDMA(2)	N/A	M/B
16	Assignment mode command	Assignment of a specific report behaviour by competent authority using a Base station	1	RATDMA, FATDMA(2)	N/A	B
17	DGNSS broadcast binary message	DGNSS corrections provided by a base station	2	RATDMA(3), RATDMA(2)	N/A	B
18	Standard Class B equipment position report	Standard position report for Class B shipborne mobile equipment to be used instead of Messages 1	1	SOTDMA, ITDMA(1),	SOTDMA, ITDMA	M
				CSTDMA		
19	Extended Class B equipment position report	Extended position report for class B shipborne mobile equipment; contains additional static information(8)	1	ITDMA	N/A	M
20	Data link management message	Reserve slots for Base station(s)	1	FATDMA(3), RATDMA	N/A	B
21	Aids-to-navigation report	Position and status report for aids-to-navigation	1	FATDMA(3), RATDMA(2)	N/A	M/B
22	Channel management(6)	Management of channels and transceiver modes by a Base station	1	FATDMA(3), RATDMA(2)	N/A	B
23	Group assignment command	Assignment of a specific report behaviour by competent authority using a Base station to a specific	1	FATDMA, RATDMA	N/A	B
24	Static data report	Additional data assigned to an MMSI	4	RATDMA, ITDMA,	N/A	M/B
		Part A: Name		CSTDMA,		
		Part B: Static Data				
				FATDMA		
25	Single slot binary message	Short unscheduled binary data transmission (Broadcast or addressed)	4	RATDMA, ITDMA,	N/A	M/B
				CSTDMA,		
				FATDMA		
26	Multiple slot binary message with Communications State	Scheduled binary data transmission (Broadcast or addressed)	4	SOTDMA, RATDMA, ITDMA	SOTDMA, ITDMA	M/B
27	Position report for long range applications	Scheduled position report; (Class A shipborne mobile equipment outside base station coverage)	1	RATDMA	N/A	M
28-63	Undefined; Reserved for future use	N/A	N/A	N/A	N/A	M

Τα δεδομένα θέσης AIS είναι διαθέσιμα στο διαδίκτυο μέσω ιδιωτικής εκμετάλλευσης γεωγραφικών συστημάτων πληροφοριών. Τον Δεκέμβριο του 2004, η IMO «Επιτροπή Θαλάσσιας Ασφάλειας» καταδίκασε τη δημοσίευση στο διαδίκτυο των δεδομένων AIS, θεωρώντας ότι η ελεύθερη παρακολούθηση της ναυσιπλοΐας παρεμποδίζει το έργο των αρμόδιων αρχών. Ωστόσο, οι υποστηρικτές της ελεύθερης διακίνησης των πληροφοριών αντιτείνουν ότι δεν υφίσταται θέμα, από τη στιγμή που κάθε σκάφος εφοδιασμένο με AIS, μπορεί απλά να το απενεργοποιήσει, οποιαδήποτε ώρα και στιγμή το επιθυμεί.

4. Εργαλεία Ανάπτυξης και Υλοποίησης

Σε αυτό το κεφάλαιο θα γίνει μια σύντομη περιγραφή των εργαλείων/συστημάτων που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής

Συγκεκριμένα χρησιμοποιήθηκαν τα παρακάτω:

- Η σουίτα Microsoft Visual Studio 2012.
- Η γλώσσα προγραμματισμού Visual Basic .NET πάνω στην οποία είναι βασισμένος ο κώδικας της εφαρμογής
- Το .NET Framework το οποίο είναι ένα ολοκληρωμένο μοντέλο προγραμματισμού της Microsoft για δημιουργία εφαρμογών.
- Το Σύστημα Διαχείρισης Βάσεων Δεδομένων Microsoft SQL Server 2012
- Η γλώσσα JavaScript που χρησιμοποιήθηκε για την απεικόνιση των δεδομένων στον χάρτη της εφαρμογής.
- Η γλώσσα JSON ως διασύνδεση (interface) για την μετάβαση των δεδομένων από τη VB.NET στην JavaScript.

4.1 Microsoft Visual Studio 2012

Το Visual Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης IDE (Integrated Development Environment) που χρησιμοποιείται για την ανάπτυξη διαδικτυακών και desktop εφαρμογών καθώς και διαδικτυακών υπηρεσιών (Web Services).

Το Visual Studio εισάγει μια νέα διαδικασία ανάπτυξης προγραμμάτων, τον παραστατικό προγραμματισμό, που αλλάζει τον τρόπο εγγραφής και εκτέλεσης των προγραμμάτων, οδηγώντας σε αύξηση της παραγωγικότητας. Παρέχει προχωρημένα εργαλεία για τη διόρθωση λαθών, για την τεκμηρίωση και εγγραφή κώδικα, για την ανάπτυξη user interfaces (διεπαφών χρήστη), για τη σχεδίαση κλάσεων καθώς και για την σχεδίαση του σχήματος μιας βάσης δεδομένων (database schema). Επίσης υποστηρίζει πολλά plug-ins για προηγμένες λειτουργίες όπως testing και refactoring.

Οι ενσωματωμένες γλώσσες προγραμματισμού του Visual Studio είναι οι Visual C++, Visual C# και Visual Basic. Επίσης, παρέχεται υποστήριξη και για άλλες γλώσσες όπως τις F#, Python, Ruby οι οποίες εγκαθίστανται ξεχωριστά μέσω language services.

4.1.1 Windows Forms

Windows forms είναι το όνομα που δόθηκε στη γραφική διεπαφή (User Interface) προγραμματισμού εφαρμογών το λεγόμενο Graphic API το οποίο αποτελεί ένα κομμάτι του .NET Framework. Τα Windows forms παρέχουν πρόσβαση στα τοπικά στοιχεία διεπαφής των Windows, αφού διαμορφώνουν το υπάρχων Windows API, σε ένα κώδικα διαχείρισης.

4.1.2 .NET Framework

Σε γενικά πλαίσια το .NET Framework είναι μία βιβλιοθήκη για υπολογιστές με λειτουργικό σύστημα Windows. Δίνει τη δυνατότητα στους προγραμματιστές να χρησιμοποιήσουν διάφορες εξελιγμένες λειτουργίες αρκεί η εφαρμογή τους να είναι συμβατή με όλα τα συστήματα που υποστηρίζουν και έχουν εγκατεστημένο το .NET Framework. Οι διάφορες εκδόσεις του .NET Framework (v.1 έως 4.5) μπορούν να βρίσκονται εγκατεστημένες ταυτόχρονα σε έναν υπολογιστή. Το Visual Studio 2012 συγκεκριμένα επιτρέπει την ανάπτυξη μιας εφαρμογής με στόχευση σε οποιαδήποτε από τις εκδόσεις 1 έως 4.5.

Το .NET Framework σχεδιάστηκε ώστε να ικανοποιεί τρεις στόχους. Αρχικά έπρεπε να κάνει τις εφαρμογές των Windows πιο αξιόπιστες, βελτιώνοντας το βαθμό της ασφάλειάς τους. Δεύτερο, προοριζόταν για την απλούστευση της ανάπτυξης Web εφαρμογών και υπηρεσιών (Web Services), οι οποίες θα έτρεχαν και σε φορητές συσκευές. Τρίτο, το Framework σχεδιάστηκε για να παρέχει ένα σύνολο βιβλιοθηκών που μπορούσαν να λειτουργήσουν με πολλές γλώσσες.

Παρακάτω θα αναλύσουμε τα βασικά σύνολα που το αποτελούν πέρα από τις γλώσσες προγραμματισμού ώστε να κατανοηθεί η ακριβής λειτουργία του.

Η αρχιτεκτονική του .NET Framework χωρίζεται σε 4 τμήματα όπως φαίνεται στο παρακάτω πίνακα. Το Common Language Runtime (CLR), ένα σύνολο από βιβλιοθήκες, ένα σύνολο από γλώσσες προγραμματισμού και από την ASP.NET.

Common Language Runtime (CLR)
Βιβλιοθήκες (Class Libraries)
Γλώσσες προγραμματισμού (C#, VC++, VB.NET, Jscript.NET)
ASP.NET

Εικόνα 4-1: Τα 4 βασικά σύνολα της αρχιτεκτονικής του .NET Framework

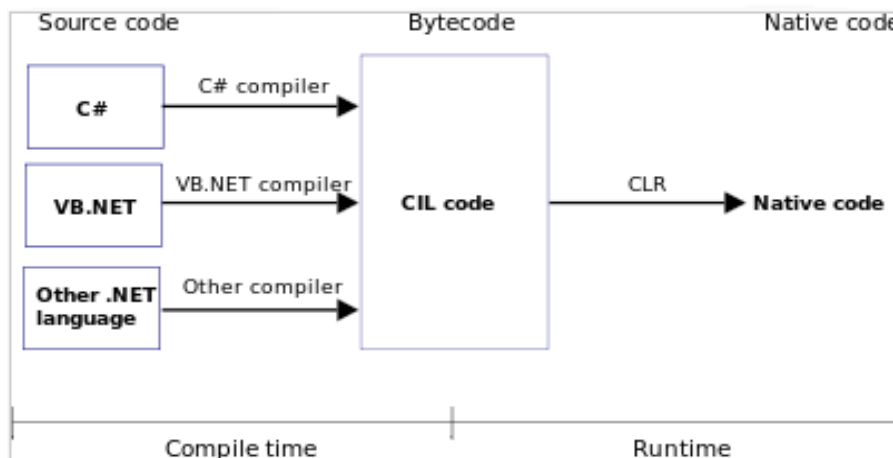
4.1.2.1 Common Language Runtime

Οι γλώσσες προγραμματισμού συνήθως αποτελούνται από έναν compiler και ένα runtime περιβάλλον. Ο compiler μεταφράζει τον κώδικα σε εκτελέσιμο αρχείο που μπορεί να εκτελεστεί από τους χρήστες. Το runtime περιβάλλον παρέχει ένα σύνολο υπηρεσιών του λειτουργικού συστήματος, στον εκτελέσιμο κώδικα. Οι υπηρεσίες αυτές είναι ενσωματωμένες σε ένα επίπεδο runtime (Runtime Layer) που επιτρέπει στον κώδικα να μην ασχολείται με λεπτομέρειες χαμηλού επιπέδου του λειτουργικού συστήματος. Τέτοιες λειτουργίες μπορεί να είναι η διαχείριση μνήμης, εγγραφή και ανάγνωση αρχείων κλπ.

Πριν το .NET Framework κάθε γλώσσα είχε και το δικό της runtime περιβάλλον. Συγκεκριμένα η Visual Basic ερχόταν με το MSVBVM60.DLL, ενώ η Visual C++ με το MSVCRT.DLL. Το περιβάλλον ενσωματωνόταν με τον εκτελέσιμο κώδικα και έπρεπε να εγκατασταθεί στο μηχάνημα του χρήστη.

Το βασικό πρόβλημα με τα περιβάλλοντα αυτά, βρίσκεται στο ότι ήταν σχεδιασμένα για χρήση με μόνο μία γλώσσα. Δεν μπορούσαν να χρησιμοποιηθούν λειτουργίες από το περιβάλλον μιας γλώσσας, σε μία άλλη. Έτσι, ένας από τους βασικούς στόχους του .NET Framework ήταν να ενοποιήσει τα runtime περιβάλλοντα ώστε οι προγραμματιστές να μπορούν να χρησιμοποιούν μόνο ένα περιβάλλον. Έτσι η λύση που δόθηκε ήταν η Common Language Runtime (CLR) που σημαίνει κοινός χρόνος εκτέλεσης γλωσσών προγραμματισμού.

Το CLR παρέχει δυνατότητες όπως διαχείριση μνήμης, ασφάλεια, διαχείριση λαθών κλπ, και όλα αυτά για κάθε γλώσσα που δουλεύει με το .NET Framework. Το CLR επίσης επιτρέπει στις γλώσσες να συνεργάζονται μεταξύ τους. Μπορεί για παράδειγμα να δεσμευτεί ένα κομμάτι μνήμης με κώδικα γραμμένο στην Visual Basic .NET και το ίδιο κομμάτι να ελευθερωθεί με κώδικα γραμμένο σε άλλη γλώσσα όπως η C#.



Εικόνα 9: Η CLR γλώσσα μετατρέπει την CIL (Common Intermediate Language) γλώσσα, σε γλώσσα του χαμηλότερου επιπέδου

4.1.2.2 .NET Class Libraries

Η βιβλιοθήκη κλάσεων (class library) του .NET Framework είναι μία συλλογή από έτοιμες κλάσεις, διεπαφές και τύπους που παρέχουν λειτουργικότητα στο σύστημα και είναι τα στοιχεία πάνω στα οποία «χτίζεται» μία εφαρμογή. Οι προγραμματιστές αρέσκονται στο να δουλεύουν με κώδικα που ήδη έχει δοκιμαστεί και φαίνεται να λειτουργεί, όπως για παράδειγμα το Win32 API και οι βιβλιοθήκες MFC. Η επαναχρησιμοποίηση κώδικα ήταν στόχος της προγραμματιστικής κοινότητας, από πολύ παλιά. Πολλές γλώσσες είχαν πρόσβαση σε κομμάτια κώδικα δοκιμασμένα, έτοιμα για εκτέλεση. Οι προγραμματιστές που χρησιμοποιούσαν την Visual C++ είχαν επωφεληθεί από βιβλιοθήκες όπως η Microsoft Foundation Classes (MFC) που τους επέτρεπαν να δημιουργήσουν εφαρμογές Windows εύκολα και γρήγορα. Ωστόσο, το ότι οι βιβλιοθήκες αυτές ήταν προορισμένες για μία μόνο γλώσσα σήμαινε ότι δεν μπορούσαν να χρησιμοποιηθούν με καμία άλλη γλώσσα. Το .NET Framework παρέχει πολλές κλάσεις για να βοηθήσει τους προγραμματιστές στην επαναχρησιμοποίηση κώδικα. Οι βιβλιοθήκες .NET Class Libraries περιέχουν κώδικα για προγραμματιστικά θέματα όπως νήματα, εγγραφή/ανάγνωση αρχείων, υποστήριξη βάσεων δεδομένων, μετατροπή σε XML, δομές δεδομένων όπως στοίβες και ουρές κλπ. Το καλύτερο σημείο βέβαια είναι το ότι η βιβλιοθήκη είναι διαθέσιμη σε κάθε γλώσσα που λειτουργεί με το .NET Framework.

Για παράδειγμα στην παρούσα εφαρμογή έχουν χρησιμοποιηθεί οι παρακάτω βιβλιοθήκες :

System.Data

Δίνει τη πρόσβαση σε classes σχετικές με την αρχιτεκτονική ADO.NET μέσω της οποίας δίνεται η δυνατότητα της διαχείρισης δεδομένων από πολλαπλές πηγές δεδομένων.

System.Data.SqlClient

Αποτελεί το Framework για την χρήση του SQL Server το οποίο παρέχει τη δυνατότητα πρόσβασης σε μια βάση δεδομένων SQL Server

System.Configuration

Περιέχει διάφορους τύπους κλάσεων οι οποίοι με τη σειρά τους παρέχουν μοντέλα για τον χειρισμό των δεδομένων διαμόρφωσης. Συγκεκριμένα ο τύπος που χρησιμοποιήθηκε ονομάζεται ApplicationSettingsBase Class και λειτουργεί ως βασική κλάση για την χρησιμοποίηση και διαχείριση των Window Forms.

System.Security.Permissions

Περιέχει διάφορες κλάσεις οι οποίες ελέγχουν την πρόσβαση σε δραστηριότητες και πόρους που έχουν σχέση με την ασφάλεια.

System.Windows.Forms

Περιέχει διάφορες κλάσεις που χρησιμοποιούνται για την δημιουργία των Windows-based εφαρμογών ώστε να αξιοποιούν πλήρως όλα τα χαρακτηριστικά διεπαφής χρήστη (user interface) που είναι διαθέσιμα στο λειτουργικό σύστημα Microsoft Windows.

4.2 Microsoft SQL Server

4.2.1 Microsoft SQL Server 2012

Ο SQL Server είναι μια σχεσιακή βάση δεδομένων, η οποία αναπτύσσεται από τη Microsoft. Οι κύριες γλώσσες που χρησιμοποιούνται είναι η T-SQL και η ANSI SQL. Ο SQL Server βγήκε για πρώτη φορά στην αγορά το 1989 σε συνεργασία με την Sybase.

Η κύρια μονάδα αποθήκευσης στοιχείων είναι μια βάση δεδομένων, η οποία αποτελείται από μια συλλογή πινάκων και κώδικα.

Η έκδοση SQL Server 2012 ανακοινώθηκε το 2011 από τη Microsoft με την κωδική ονομασία Denali. Κυκλοφόρησε τον Μάρτιο του 2012 ενώ το Νοέμβριο του ίδιου έτους κυκλοφόρησε και το Service Pack 1. Η συγκεκριμένη έκδοση είναι η τελευταία που υποστηρίζει εγγενώς το OLEDB API και πλέον θα υποστηρίζεται μόνο το ODBC API. Κάποια από τα νέα χαρακτηριστικά που εισήχθησαν στον SQL Server είναι το 'AlwaysOn SQL Server Failover Cluster', η εισαγωγή δυναμικών views και functions, εργαλεία για την απλοποίηση της μεταφοράς των βάσεων δεδομένων μεταξύ των διαφόρων instances, βελτιώσεις στα χωρικά χαρακτηριστικά κ.α.

4.2.1.1 Γεωγραφικοί τύποι δεδομένων του SQL Server 2012

Σαν DBMS χρησιμοποιήθηκε ο Microsoft SQL Server 2012 οποίος δίνει την δυνατότητα στο χρήστη να εισάγει και να χρησιμοποιήσει χωρικά (spatial) δεδομένα. Με τον όρο χωρικά εννοούμε δεδομένα που αντιπροσωπεύουν πληροφορίες για αντικείμενα με γεωγραφική θέση και γεωμετρική σχήμα. Συγκεκριμένα ο SQL Server υποστηρίζει δύο τύπους χωρικών δεδομένων.

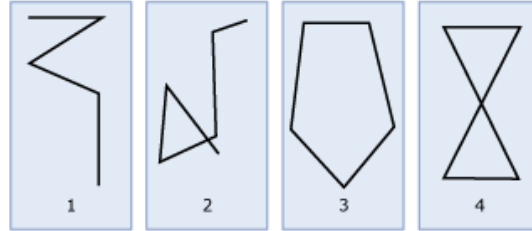
- Τον γεωμετρικό τύπο δεδομένων (geometry data) ο οποίος αντιπροσωπεύει δεδομένα στο ευκλείδειο σύστημα συντεταγμένων (στο επίπεδο δηλαδή).
- και τον γεωγραφικό τύπο δεδομένων (geography data) . ο οποίος αντιπροσωπεύει δεδομένα στο γεωγραφικό σύστημα συντεταγμένων.

Και οι δύο τύποι δεδομένων υλοποιούνται με χρήση CLR (Common Language Runtime) από τον ίδιο τον SQL Server..

Οι γεωμετρικοί και γεωγραφικοί τύποι δεδομένων υποστηρίζουν δεκάξι χωρικά αντικείμενα δεδομένων. Από αυτά μόνο τα παρακάτω έχουν υπόσταση και μπορεί ο χρήστης να χρησιμοποιήσει στον SQL Server 2012. Αυτά είναι:

- Point (Σημείο): Το point για τον γεωγραφικό τύπο δεδομένων αντιπροσωπεύει μια μοναδική τοποθεσία (σημείο) που ορίζεται από ένα συγκεκριμένο γεωγραφικό μήκος (longitude) και πλάτος (latitude). Οι τιμές των lat και Long μετριοούνται σε μοίρες. Οι τιμές για το latitude ορίζονται πάντα μεταξύ του διαστήματος [-90, 90]. Σε περίπτωση που εισαχθούν τιμές έξω από το συγκεκριμένο διάστημα ο χρήστης λαμβάνει σφάλμα. Αντίστοιχα οι τιμές για το longitude ορίζονται πάντα μεταξύ του διαστήματος [-180, 180]. Επίσης στην περίπτωση εισαγωγής τιμής εκτός του προαναφερθέντος εύρους αυτόματα προσαρμόζονται εντός του εύρους αυτού. Για παράδειγμα αν εισαχθεί η τιμή 190 τότε αυτόματα μετατρέπεται σε -170 ($190 - 360 = -170$).

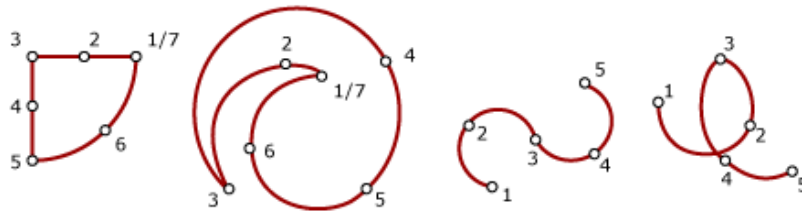
- *Linestring (Γραμμή)*: Το LineString είναι μια διάστασης αντικείμενο το οποίο αντιπροσωπεύει μια σειρά από σημεία (points) τα οποία ενώνονται μέσω ευθυγράμμων τμημάτων. Ένα απλό όχι κλειστό (non closed) LineString εμφανίζεται στην παρακάτω εικόνα.



Εικόνα 4-2: Παραδείγματα έγκυρων Linestring αντικειμένων, χωρικών δεδομένων

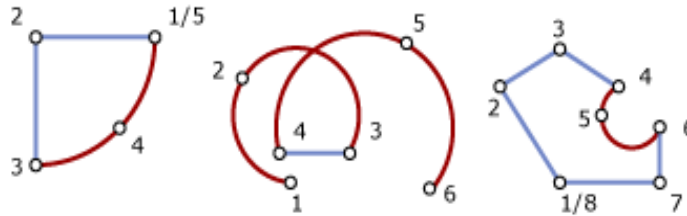
Για να είναι έγκυρος ένας τέτοιος τύπος δεδομένων θα πρέπει κατά τον ορισμό τους να ικανοποιούνται τα παρακάτω κριτήρια:

- ✓ Αν δεν είναι άδειο θα πρέπει να περιέχει τουλάχιστον δύο διαφορετικά σημεία (2 ζεύγη Lat Lon).
 - ✓ Δεν μπορεί να επικαλύψει τον εαυτό του σε ένα διάστημα δύο ή περισσότερων συνεχόμενων σημείων
- *CircularString*: Είναι μια συλλογή από συνεχή κυκλικά τμήματα τόξου. Ένα κυκλικό τμήμα τόξου είναι ένα κυρτό τμήμα που ορίζεται από τρία σημεία σε ένα διδιάστατο επίπεδο. Στον συγκεκριμένο τύπο δεδομένων το πρώτο σημείο δεν μπορεί να είναι ίδιο με το τελευταίο σημείο ενώ αν όλα τα σημεία (και τα 3) είναι συγγραμμικά τότε το συγκεκριμένο αντιμετωπίζεται ως ένα ευθύγραμμο τμήμα. Ένα τέτοιου τύπου δεδομένο είναι αποδεκτό στις παρακάτω περιπτώσεις:
 - ✓ Εάν είναι άδειο
 - ✓ Εάν αποτελείται από μονό αριθμό Lat, Lon σημείων
 - ✓ Εάν αποτελείται από περισσότερα από 1 Lat, Lon σημεία



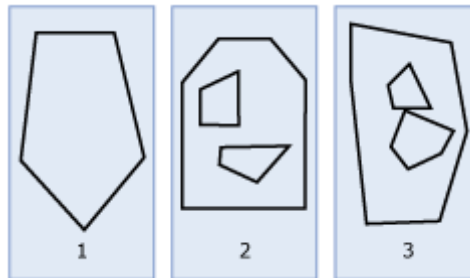
Εικόνα 4-3: Παραδείγματα έγκυρων CircularString αντικειμένων, χωρικών δεδομένων

- *CompoundCurves*: Είναι μια συλλογή (collection) από κανένα ή περισσότερα συνεχόμενα CircularString ή LineString, γεωγραφικών ή γεωμετρικών τύπων. Για να είναι έγκυρος ο τύπος αυτός θα πρέπει να ικανοποιούνται τα παρακάτω κριτήρια:
 - ✓ Θα πρέπει να περιέχει τουλάχιστον 1 CircularString ή LineString τύπο δεδομένων
 - ✓ Θα πρέπει η ακολουθία των παραπάνω τύπων να είναι συνεχόμενη
 Ενώ για να είναι αποδεκτός θα πρέπει να πληροί τις παρακάτω προϋποθέσεις:
 - ✓ Να είναι άδειο
 - ✓ Να είναι έγκυρα όλα τα κυκλικά τμήματα τόξων που τον αποτελούν
 - ✓ Να ενώνονται όλα τα κυκλικά τμήματα τόξου που τον αποτελούν



Εικόνα 4-4: Παραδείγματα έγκυρων CompoundCurve αντικειμένων, χωρικών δεδομένων

- *Polygon (Πολύγωνο)*: Το Polygon είναι μια δύο διαστάσεων επιφάνεια που αποθηκεύεται ως μια ακολουθία σημείων τα οποία καθορίζουν έναν εξωτερικό δακτύλιο και κανένα ή περισσότερους εσωτερικούς.



Εικόνα 4-5: Παραδείγματα έγκυρων Polygon αντικειμένων, χωρικών δεδομένων

Το polygon μπορεί να σχηματίζεται από έναν δακτύλιο ο οποίος έχει τουλάχιστον τρία διαφορετικά σημεία. Επίσης ένα πολύγωνο μπορεί να είναι άδειο. Ο εξωτερικός καθώς και οποιοσδήποτε εσωτερικός δακτύλιος ορίζει τα σύνορα του πολύγωνου ενώ ο χώρος εντός των δακτυλίων ορίζει το εσωτερικό του πολύγωνου.

Ένα πολύγωνο θεωρείται έγκυρο όταν ικανοποιεί τα παρακάτω κριτήρια:

- ✓ Θα πρέπει αν δεν είναι άδειο
- ✓ Να περιέχει τουλάχιστον τέσσερα σημεία (4 ζεύγη Lat Lon)
- ✓ Το πρώτο και το τελευταίο σημείο του LineString που σχηματίζει το πολύγωνο θα πρέπει να είναι το ίδιο (ώστε να κλείνει σωστά το πολύγωνο).

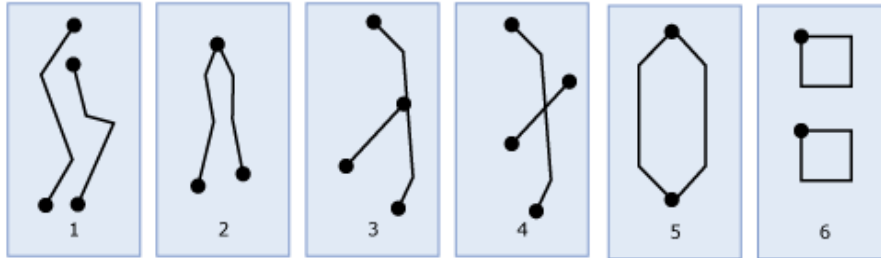
- *CurvePolygon*: Είναι μια τοπολογικά κλειστή επιφάνεια που ορίζεται από έναν εξωτερικό δακτύλιο ο οποίος περιέχει κανέναν ή περισσότερους εσωτερικούς δακτυλίους. Τα χαρακτηριστικά ενός CurvePolygon καθορίζονται από τα παρακάτω κριτήρια:

- ✓ Τα σύνορα (boundary) του συγκεκριμένου τύπου ορίζονται από τον εξωτερικό δακτύλιο και όλους τους υπόλοιπους εσωτερικούς δακτυλίους
- ✓ Το εσωτερικό του ορίζεται ως ο χώρος ανάμεσα του εξωτερικού δακτυλίου και όλων των εσωτερικών δακτυλίων

Ενώ για να είναι αποδεκτός θα πρέπει να πληροί τις παρακάτω προϋποθέσεις:

- ✓ Να είναι άδειο
- ✓ Να περιέχει μόνο κυκλικούς δακτυλίους οι οποίοι με τη σειρά τους να είναι αποδεκτοί.

- *MultiPoint*: Είναι μια συλλογή (collection) από κανένα ή περισσότερα σημεία (points). Ο συγκεκριμένος τύπος θεωρείται ότι έχει μηδενικά όρια (boundaries)
- *MultiLineString*: Είναι μια συλλογή (collection) από κανέναν ή περισσότερους γεωμετρικού ή γεωγραφικούς τύπους δεδομένων LineString.



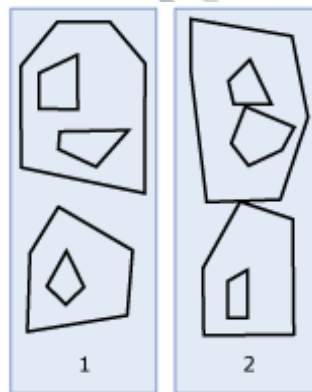
Εικόνα 4-6: Παραδείγματα έγκυρων MultiLineString αντικειμένων, χωρικών δεδομένων

Για να είναι αποδεκτά θα πρέπει ή να είναι άδεια ή να αποτελούνται από μόνο αποδεκτούς τύπους δεδομένων LineString

Ενώ για να είναι έγκυροι θα πρέπει να πληρούν τα παρακάτω κριτήρια:

- ✓ Όλα τα τμήματα που το αποτελούν θα πρέπει να αποτελούνται από έγκυρα LineStrings
- ✓ Δεν θα πρέπει παραπάνω από 2 LineStrings που αποτελούν μέρος του MultiLineString να επικαλύπτονται πάνω από ένα σημείο. Τα LineStrings μπορούν όμως να τέμνουν τους εαυτούς τους ή άλλα Linestrings σε άπειρα σημεία.

- MultiPolygon: Είναι μια συλλογή (collection) από κανένα ή περισσότερα Πολύγωνα (Polygons).



Εικόνα 4-7: Παραδείγματα έγκυρων MultiPolygon αντικειμένων, χωρικών δεδομένων

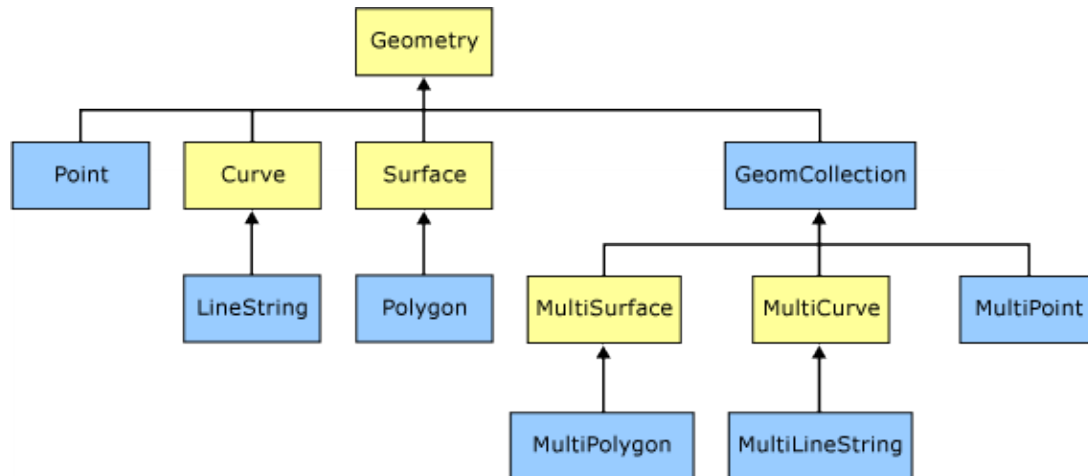
Αποδεκτά θεωρούνται τα MultiPolygons που πληρούν τις παρακάτω συνθήκες:

- ✓ Να είναι άδεια
- ✓ Όλα τα πολύγωνα που αποτελούν το MultiPolygons να είναι αποδεκτά με τη σειρά τους.

Ενώ θεωρούνται έγκυρα όταν πληρούν τα παρακάτω κριτήρια:

- ✓ Όλα τα Polygons που αποτελούν το MultiPolygon να είναι έγκυρα
- ✓ Κανένα από τα polygons να μην υπερκαλύπτατε από κανένα άλλο.

Το παρακάτω σχήμα περιγράφει την γεωμετρική ιεραρχία στην οποία βασίζονται τα γεωμετρικού και γεωγραφικού τύπου δεδομένα.



Εικόνα 4-8: Η γεωμετρική ιεραρχία στην οποία βασίζονται τα γεωμετρικού και γεωγραφικού τύπου χωρικά δεδομένα.

4.2.1.2 Διαφορές μεταξύ γεωγραφικών και γεωμετρικών τύπων δεδομένων.

Οι δύο τύποι χωρικών δεδομένων συμπεριφέρονται σχετικά όμοια αλλά υπάρχουν μερικές βασικές διαφορές στο πώς ο SQL Server αποθηκεύει και χειρίζεται τα δεδομένα τους.

Τα δεδομένα που ορίζουν ένα τύπου LineString και ένα τύπο Polygon είναι οι κορυφές τους και μόνο. Η σύνδεση μεταξύ δύο κορυφών στα γεωμετρικού τύπου δεδομένα είναι μια ευθεία γραμμή. Αντίθετα η αντίστοιχη σύνδεση δύο κορυφών στα γεωγραφικά τύπου δεδομένα είναι ένα μικρό ελλειπτικό τόξο το οποίο ενώνει τις κορυφές.

Τα κυκλικά αυτά τοξοειδή τμήματα για τα γεωγραφικά δεδομένα ορίζονται στο Καρτεσιανό επίπεδο x και y συντεταγμένων ενώ τα κυκλικά τμήματα ορίζονται από τμήματα καμπύλης αντικειμένων τύπου σφαίρας. Κάθε παράλληλος στη σφαίρα αναφοράς μπορεί να ορίζεται από 2 συμπληρωματικά τόξα όπου τα σημεία για τα τόξα αυτά έχουν μια σταθερή γωνία γεωγραφικού πλάτους.

Σχετικά με τις μετρήσεις στα χωρικού τύπου δεδομένα, στην επίπεδη γη οι μετρήσεις των αποστάσεων και των περιοχών δίνονται στις ίδιες μονάδες μέτρηση όπως οι συντεταγμένες. Για παράδειγμα χρησιμοποιώντας γεωμετρικούς τύπους δεδομένων η απόσταση μεταξύ του σημείου (2,2) και του σημείου (5,6) είναι 5 μονάδες ασχέτως από την μονάδα μέτρησης.

Στο ελλειψοειδές σύστημα όμως της γης οι συντεταγμένες δίνονται σε μοίρες γεωγραφικού πλάτους και μήκους. Επομένως οι αποστάσεις και οι περιοχές μετριοούνται σε μέτρα και τετραγωνικά μέτρα αν και η κάθε μέτρηση μπορεί να εξαρτάται από το SRID (Spatial Reference Identifier). Το SRID είναι μια μοναδική τιμή η οποία χρησιμοποιείται για την ταυτοποίηση των προβολών των χωρικών δεδομένων στο σύστημα συντεταγμένων.

Σε ένα επίπεδο σύστημα ο δακτύλιος ενός πολύγону δεν είναι σημαντικός παράγοντας. Για παράδειγμα ένα πολύγону που περιγράφεται από τα σημεία ((0, 0), (10, 0), (0, 20), (0, 0)) είναι ίδιο με ένα πολύγону που περιγράφεται από τα διαφορετικά σημεία ((0, 0), (0, 20), (10, 0), (0, 0)).

Σε ένα ελλειψοειδές σύστημα όμως, ένα πολύγону δεν έχει κανένα νόημα χωρίς να έχει ορισμένο προσανατολισμό. Στο γεωγραφικό τύπο δεδομένων για την αποθήκευση χωρικών δεδομένων πρέπει να καθορίζετε ο προσανατολισμός του δακτυλίου και να περιγράφεται με ακρίβεια η θέση του.

4.2.2 SQL Agent

Ονομάζεται επίσης και SQL Server Agent. Αποτελεί ένα εργαλείο που τρέχει στο παρασκήνιο του συστήματος διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) SQL Server.

Το εργαλείο αυτό επιτρέπει στον διαχειριστή της βάσης δεδομένων (DBA – Data Base Administrator) να προγραμματίσει αυτοματοποιημένες εργασίες (jobs) όπως και άλλες εργασίες διαχείρισης του RDBMS.

Εκτελείται ως windows service και χειρίζεται διάφορες εργασίες όπως την αυτόματη δημιουργία αντιγράφων ασφαλείας (automation backup), την εκτέλεση προγραμματισμένων εργασιών (Job scheduling), τα δικαιώματα των χρηστών (user permissions) και γενικά την παρακολούθηση της βάση δεδομένων (database monitoring). Όλα τα παραπάνω δεν απαιτούν πάντα λειτουργίες του SQL Server. Για παράδειγμα μπορεί να οριστεί τα αντίγραφα ασφαλείας των βάσεων να δημιουργούνται από ένα εξωτερικό πρόγραμμα των windows.

Στη δική μας περίπτωση μέσω του SQL Agent χρησιμοποιούνται τα SQL Agent Jobs. Στη πραγματικότητα αυτά αποτελούνται από μια σειρά από βήματα (steps) μέσα στα οποία ο διαχειριστής μπορεί να ορίσει να εκτελεστούν συγκεκριμένες εργασίες. Ο SQL Server δίνει και τη δυνατότητα της γραφικής απεικόνισης (GUI – Graphical User Interface) επομένως ανεξαρτήτου εμπειρίας του διαχειριστή μπορεί να καθορίσει τα βήματα για τις εργασίες που επιθυμεί.

Τέλος αφού ορίσει μια δουλειά (job) μπορεί να προγραμματίσει την συχνότητα εκτέλεσης της δουλειάς καθώς και τα διάστημα του κάθε πότε θα εκτελείτε (intervals). Για παράδειγμα, μπορεί να ορίσει μια δουλειά να εκτελείτε μια φορά την ημέρα, μια φορά την εβδομάδα, μία φορά το μήνα ή και συγκεκριμένες ώρες.

4.3 JavaScript

Η ανάπτυξη της γλώσσα προγραμματισμού JavaScript ξεκίνησε από την εταιρεία Netscape με την επωνυμία Mocha ενώ αργότερα μετονομάστηκε σε LiveScript και τελικά JavaScript κυρίως λόγω του ότι η ανάπτυξη της επηρεάστηκε από τη γλώσσα προγραμματισμού Java.

Η JavaScript είναι μία διερμηνευμένη (interpreted) γλώσσα προγραμματισμού με ιδιότητες αντικειμενοστραφούς γλώσσας προγραμματισμού, χωρίς όμως να μπορεί να χαρακτηριστεί ως πλήρης αντικειμενοστραφής. Η γλώσσα αυτή, κτίστηκε ουσιαστικά πάνω στο πρότυπο των γλωσσών C και C++, ενώ αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από την Java αν και γενικά οι δύο αυτές γλώσσες δεν σχετίζονται. Επίσης ένα πολύ σημαντικό χαρακτηριστικό που τη διαφοροποιεί είναι ότι διαχειρίζεται τους τύπους δεδομένων πιο χαλαρά (loosely typed) σε σχέση με τη σφικτή διαχείριση τύπων δεδομένων (strongly typed) που γίνεται στις προαναφερόμενες γλώσσες.

Η JavaScript χρησιμοποιείται κυρίως σε εφαρμογές ιστοσελίδων αλλά και σε έγγραφα PDF, σε εξειδικευμένους browsers και σε μικρές εφαρμογές επιφάνειας εργασίας (desktop widgets).

Χαρακτηριστικό της JavaScript είναι ότι οι μεταβλητές δεν είναι απαραίτητο να έχουν ένα συγκεκριμένο τύπο ενώ είναι δυνατόν ακόμα και να αλλάζουν τύπο κατά τη διάρκεια της ζωής τους.

4.4 JSON (JavaScript Object Notation)

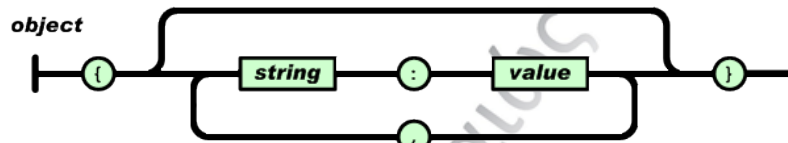
Η γλώσσα JSON (JavaScript Object Notation) είναι μια σημασιολογική γλώσσα σε μορφή κειμένου, παρόμοια με την XML, η οποία χρησιμοποιείται ως μια εναλλακτική μορφή αναπαράστασης απλών δομών δεδομένων και συστοιχιών που ονομάζονται αντικείμενα. Συνήθως χρησιμοποιείται για την ανταλλαγή δεδομένων μεταξύ διαφορετικής τεχνολογίας εφαρμογών.. Είναι βασισμένο πάνω σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript, Standard ECMA-262 Έκδοση 3η - Δεκέμβριος 1999.

Το JSON είναι ένα πρότυπο κειμένου το οποίο είναι τελείως ανεξάρτητο από γλώσσες προγραμματισμού (language-independent) αλλά χρησιμοποιεί πρακτικές (conventions) οι οποίες είναι γνωστές στους προγραμματιστές της οικογένειας προγραμματισμού C, συμπεριλαμβανομένων των C, C++, C#, Java, JavaScript, Perl, Python, και πολλών άλλων. Αυτές οι ιδιότητες κάνουν το JSON μια ιδανική γλώσσα προγραμματισμού ανταλλαγής δεδομένων.

Το JSON είναι χτισμένο σε δύο δομές:

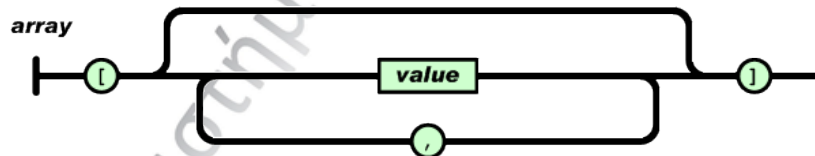
- Μια συλλογή από ζευγάρια ονομάτων/τιμών. Σε διάφορες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένα object, καταχώριση, δομή, λεξικό, πίνακα hash (hash table), λίστα κλειδιών, ή associative πίνακα.
- Μία ταξινομημένη λίστα τιμών. Στις περισσότερες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένας πίνακας (array), διάνυσμα, λίστα, ή ακολουθία.

Ένα αντικείμενο (object) είναι ένα άτακτο σύνολο από ζευγάρια ονομάτων/τιμών. Ένα αντικείμενο (object) ξεκινάει με { (αριστερό άγκιστρο) και τελειώνει με } (δεξιό άγκιστρο). Κάθε όνομα ακολουθείται από : (άνω-κατω τελεία) και τα ζευγάρια ονόματος/τιμής χωρίζονται από , (κόμμα).



Εικόνα 4-9: Σύνταξη JSON εντολής αντικειμένου (object)

Ένας πίνακας (array) είναι μια συλλογή από τιμές σε σειρά. Ένας πίνακας (array) ξεκινάει με [(αριστερή αγκύλη) και τελειώνει με] (δεξιά αγκύλη). Οι τιμές χωρίζονται με , (κόμμα).



Εικόνα 4-10: Σύνταξη JSON εντολής πίνακα (array)

4.5 Html (Hyper Text Markup Language)

HTML είναι το ακρωνύμιο των λέξεων Hyper Text Markup Language (Γλώσσα Σήμανσης Υπερκειμένου) και είναι η βασική γλώσσα δόμησης σελίδων του διαδικτύου.

Είναι μια περιγραφική γλώσσα, δηλαδή ένας ειδικός τρόπος γραφής κειμένου. Αποτέλεσε υποσύνολο της γλώσσας SGML (Standard Generalized Markup Language) που επινοήθηκε από την IBM προκειμένου να λυθεί το πρόβλημα της μη τυποποιημένης εμφάνισης κειμένων στα διάφορα υπολογιστικά συστήματα. Είναι η πρώτη και η πιο διαδεδομένη γλώσσα περιγραφής ιστοσελίδας την οποία αναγνωρίζουν πλέον όλοι οι browsers. Αρχικά είχε δημιουργηθεί με σκοπό την μορφοποίηση κειμένου, αλλά αργότερα αναπτύχθηκε και ενσωμάτωσε και σχεδιαστικές τεχνικές.

Η γλώσσα χρησιμοποιεί ένα αριθμό από ειδικές ετικέτες (tags) για την μορφοποίηση κειμένου, για την δημιουργία συνδέσμων (links) μετάβασης ανάμεσα των σελίδων, για την εισαγωγή εικόνων, ήχου κ.α.

Οι ετικέτες ουσιαστικά ελέγχουν την δομή και την μορφή του κειμένου της ιστοσελίδας. Επίσης παρέχουν πληροφορίες προς τον web browser για την σελίδα που πρόκειται να εμφανίσουν, όπως ο τίτλος της σελίδας ή ο συγγραφέας της, κ.α.

Η HTML διαθέτει ένα πεπερασμένο αριθμό ετικετών που μπορούμε να χρησιμοποιήσουμε ωστόσο ο αριθμός αυτός δεν παραμένει σταθερός. Κατά διαστήματα το W3Consortium το οποίο ανέπτυξε και διαχειρίζεται τα

πρότυπα της HTML, δημοσιεύει νέα πρότυπα στα οποία προσθέτει καινούργιες ετικέτες που καλύπτουν ή διορθώνουν μια λειτουργία στο προηγούμενο πρότυπο. Η τελευταία αναθεώρηση του HTML προτύπου είναι η HTML5.

Οι HTML ετικέτες γράφονται ανάμεσα στα σύμβολα '<' και '>' για παράδειγμα < όνομα-ετικέτας >. Οι περισσότερες HTML ετικέτες αποτελούνται από μια ετικέτα αρχής, μια ετικέτα τέλους και ανάμεσα σε αυτές υπάρχει το κείμενο που χαρακτηρίζεται από τις ετικέτες αυτές. Η ετικέτα τέλους περιέχει τον χαρακτήρα '/' πριν το όνομα της ετικέτας για να προσδιορίσει το τέλος, για παράδειγμα <όνομα ετικέτας> ... κείμενο ... </όνομα ετικέτας>

Όταν ένας web browser ανοίγει ένα αρχείο HTML τα στοιχεία (tags) μεταφράζονται σε κατάλληλα χαρακτηριστικά με αποτελέσματα την εμφάνιση και τη λειτουργικότητα της συγκεκριμένης σελίδας.

Η δημιουργία αρχείων HTML είναι πάρα πολύ απλή. Αρκεί να "τρέξουμε" έναν οποιοδήποτε αναλυτή κειμένου, όπως το notepad των Windows ή το notepad++. Να γράψουμε τον κώδικα HTML που επιθυμούμε και να το αποθηκεύσουμε (σώσουμε) σε ένα αρχείο με κατάληξη .htm ή .html.

Συνήθως αποθηκεύουμε με επέκταση .html όταν θα δημοσιεύσουμε τα αρχεία μας σε Unix περιβάλλοντα. Αντίθετα ο εξυπηρετητής Windows NT καταλαβαίνει τα αρχεία και σαν .htm και σαν .html. Τα αρχεία αυτά είναι απλά αρχεία κειμένου σε μορφή ASCII και δεν περιέχουν πληροφορίες για το περιβάλλον ή τα προγράμματα με τα οποία θα λειτουργήσουν.

4.6 Google Maps JavaScript API v3

Η Google προσφέρει μια ποικιλία από APIs (Application Programming Interfaces), ως επί το πλείστον για προγραμματιστές διαδικτυακών εφαρμογών. Τα APIs βασίζονται σε δημοφιλή προϊόντα της Google όπως το Google Maps, το Google Earth, το AdWords και το YouTube. Το Google Maps JavaScript API συγκεκριμένα επιτρέπει στο χρήστη να ενσωματώσει το Google Maps στην δική του εφαρμογή είτε είναι web σελίδα είτε είναι διαδικτυακή εφαρμογή desktop.

Η έκδοση 3 σχεδιάστηκε ειδικά στο να είναι πιο γρήγορη και καλύτερα εφαρμόσιμη στους παραδοσιακούς browsers αλλά κυρίως στις κινητές συσκευές (mobile devices) αφού αυτή είναι η τάση της τεχνολογίας πλέον. Το συγκεκριμένο API προσφέρει μια σειρά από δυνατότητες για τον χειρισμό των χαρτών και την εισαγωγή περιεχομένου μέσα από την παροχή αρκετών υπηρεσιών. Είναι μια δωρεάν υπηρεσία που διατίθεται σε οποιονδήποτε προγραμματιστή θέλει να το χρησιμοποιήσει σε μια δωρεάν σελίδα ή εφαρμογή.

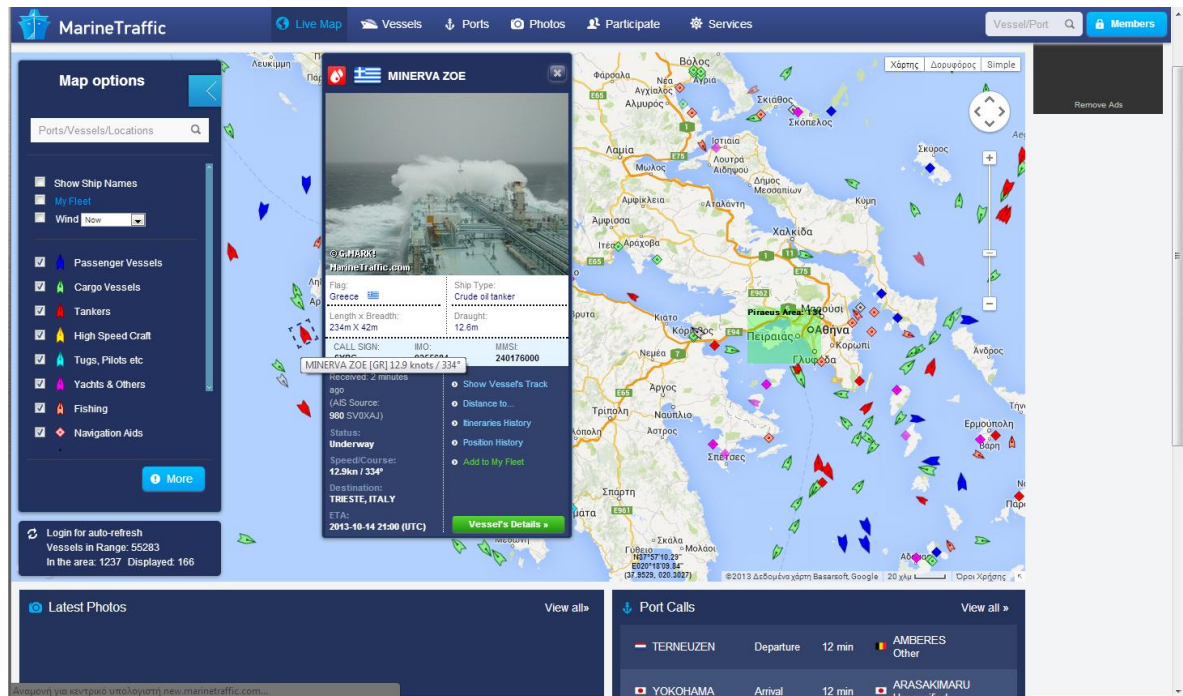
5. Αντίστοιχες Υλοποιημένες Εφαρμογές

Αντίστοιχα συστήματα και εφαρμογές με αυτή που δημιουργήσαμε στην παρούσα εργασία υπάρχουν πολλά στο εμπόριο. Επίσης μεγάλες ναυτιλιακές εταιρίες δεν μπαίνουν καν στη διαδικασία να αγοράσουν κάποιο αντίστοιχο λογισμικό αλλά υλοποιούν εκείνες κάποιο σύμφωνα με τα δικά του πρότυπα και ανάλογα με τις ανάγκες τους. Στο παρόν κεφάλαιο θα δούμε κάποια από τα ανταγωνιστικά συστήματα τα οποία ουσιαστικά προτείνουν κάποιες λύσεις σε προβλήματα που σχετίζονται με τον έλεγχο της πορείας των πλοίων και την παρακολούθησή τους.

5.1 Marinetraffic

Οι δημοφιλέστερη τέτοια εφαρμογή είναι η διαδικτυακή εφαρμογή marinetraffic την οποία μπορεί κανείς να απαντήσει ακολουθώντας το παρακάτω σύνδεσμο <http://www.marinetraffic.com/ais/>.

Σε αυτό ο κάθε χρήστης μπορεί δωρεάν να αντλήσει πληροφορίες σχετικά με τις τρέχουσες θέσεις όλων των πλοίων σε κάθε γωνία της γης. Επίσης μπορεί να ενημερωθεί για οποιαδήποτε πληροφορία σχετικά με κάθε πλοίο όπως όνομα, σημαία, τύπος, μέγεθος, φωτογραφίες κ.α. Η αρχική σελίδα της συγκεκριμένης ιστοσελίδας φαίνεται παρακάτω.



Εικόνα 5-1: Η εφαρμογή MarineTraffic

Η εφαρμογή αυτή είναι ελεύθερα προσβάσιμη από όλους αφού είναι ένα ελεύθερο έργο βασισμένο στην κοινοτική συνεισφορά. Στόχος της συγκεκριμένης εφαρμογή είναι η αξιοποίησή του σε ερευνητικές εφαρμογές, η μελέτη και σχεδίαση αλληλεπιδραστικών πληροφοριακών συστημάτων και η εξαγωγή χρήσιμων συμπερασμάτων για την κυκλοφορία των πλοίων. Μέσα στους ευρύτερους στόχους του είναι να παρέχει στο κοινό πληροφορίες πραγματικού χρόνου για τις κινήσεις των πλοίων, τα λιμάνια και τους φάρους.

Η συλλογή των δεδομένων βασίζεται στο διεθνές σύστημα AIS (Automatic Identification System). Η επέκταση της κάλυψης βασίζεται στην κοινότητα εθελοντών και στο ενδιαφέρον τοπικών Αρχών που εγκαθιστούν έναν δέκτη και στέλνουν στον κεντρικό server τα δεδομένα που συλλέγουν για την περιοχή τους. Έτσι στη συνέχεια τα δεδομένα αυτά γίνονται ελεύθερα προσβάσιμα από τον κάθε χρήστη που το επιθυμεί.

Τα δεδομένα που λαμβάνονται μέσω του AIS καταχωρούνται στη βάση δεδομένων σε πραγματικό χρόνο και συνεπώς είναι άμεσα διαθέσιμα στο χάρτη και στις υπόλοιπες σελίδες. Όμως, οι θέσεις κάποιων πλοίων είναι πιθανό να μην ανανεώνονται συνεχώς (π.χ. όταν κάποιο πλοίο είναι οριακά εντός της εμβέλειας των σταθμών λήψης). Σε αυτήν την περίπτωση τα στίγματα κάποιων πλοίων μπορεί να εμφανίζονται έως και μία ώρα μετά την τελευταία λήψη τους.

Υπηρεσίες που παρέχει η εφαρμογή

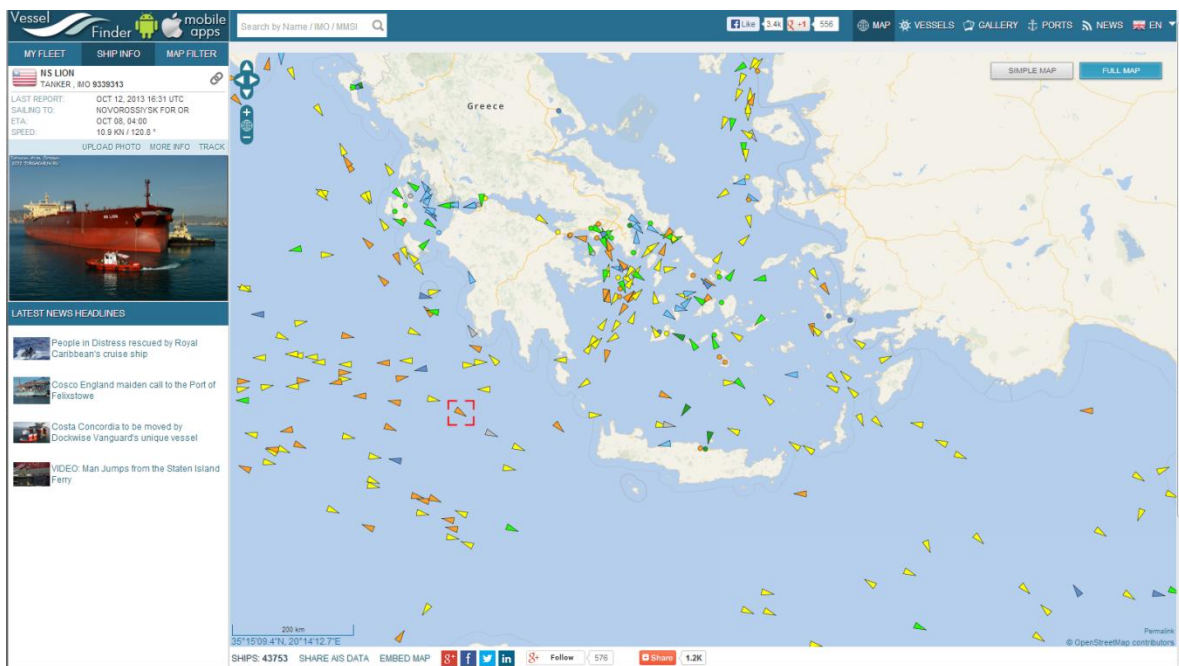
- Δίνει τη δυνατότητα στον χρήστη να παρακολουθήσει συγκεκριμένα πλοία ή συγκεκριμένο στόλο πλοίων καθώς και να λαμβάνει ειδοποιήσεις μέσω email για τις αφίξεις και αναχωρήσεις τους. Επίσης ο χρήστης μπορεί να ορίσει σημεία ενδιαφέροντος καθώς και να λάβει ειδοποιήσεις μέσω SMS αν έχει ενεργοποιήσει την αντίστοιχη υπηρεσία
- Προσφέρει τη χρήση και ενσωμάτωση του χάρτη που παρέχει η ιστοσελίδα στη σελίδα οποιουδήποτε χρήστη με την προϋπόθεση ότι είναι μη κερδοσκοπικού ενδιαφέροντος.
- Προσφέρει παγκόσμια ευρετήριο φάρων και βοηθημάτων ναυσιπλοΐας.
- Προσφέρει και ιστορικά δεδομένα χρόνων σχετικά με πλοία, λιμάνια ή περιοχών, για οποίον χρήστη ενδιαφέρεται.

- Παρέχεται και σε android ή iOS εφαρμογή

5.2 VesselFinder

Πρόκειται για μια διαδικτυακή εφαρμογή στην οποία μέσω AIS δεδομένων εμφανίζονται τα σημεία των πλοίων πάνω σε ένα χάρτη ώστε ο κάθε χρήστης να έχει τη δυνατότητα να λάβει πληροφορίες σχετικά με αυτά. Η διεύθυνση της ιστοσελίδας είναι <http://www.vesselfinder.com/>.

Η αρχική σελίδα φαίνεται παρακάτω.



Εικόνα 5-2: Η εφαρμογή VesselFinder

Το VesselFinder λειτουργεί και εμφανίζει τα πλοία χρησιμοποιώντας AIS δεδομένα τα οποία η εταιρία αποθηκεύει σε δικό της server ενώ καλύπτει έναν αριθμό πλοίων κοντά στις 300 χιλιάδες. Την πληροφορία του εκάστοτε πλοίου που επιλέγει ο χρήστης την εμφανίζει δεξιά στην ιστοσελίδα και όχι επάνω στον χάρτη ως infowindow.

Στις πληροφορίες που παρέχει στον χρήστη περιλαμβάνονται ο τύπος του πλοίου ο προορισμός του, η ταχύτητα του, η κατεύθυνση που έχει, ο εκτιμώμενος χρόνος άφιξης του πλοίου στην προορισμό του, φωτογραφία του πλοίου καθώς και η ώρα που ελήφθη το συγκεκριμένο σήμα. Επίσης ο χρήστης μπορεί να αναζητήσει πληροφορίες για συγκεκριμένα πλοία χρησιμοποιώντας το όνομα τους ή τον MMSI αριθμό.

Υπηρεσίες που παρέχει η εφαρμογή

- Παγκόσμια κάλυψη δεδομένων των πλοίων
- Δίνει τη δυνατότητα στον χρήστη να αναζητήσει μέσω ονόματος ή αριθμού IMO/MMSI ένα πλοίο και να δει πληροφορίες όπως τον τελικό του προορισμό ή τον χρόνο εκτιμώμενης άφιξης προορισμού κ.α..
- Για όλα τα πλοία που εμφανίζονται ο χρήστης μπορεί να λάβει πληροφορίες σχετικά με την τρέχουσα θέση τους, την πορεία του πλοίου καθώς και ιστορικά δεδομένα.

- Παρέχει ενημέρωση στον χρήστη για όλα τα τελευταία νέα της ναυτιλίας σχετικά με διάφορα περιστατικά πειρατείας, οικονομικά θέματα, αγοραπωλησίες νέων πλοίων και για διάφορα άλλα σχετικά θέματα
- Ο χρήστης μπορεί να βρει μεγάλο αριθμό φωτογραφιών για το κάθε πλοίο.

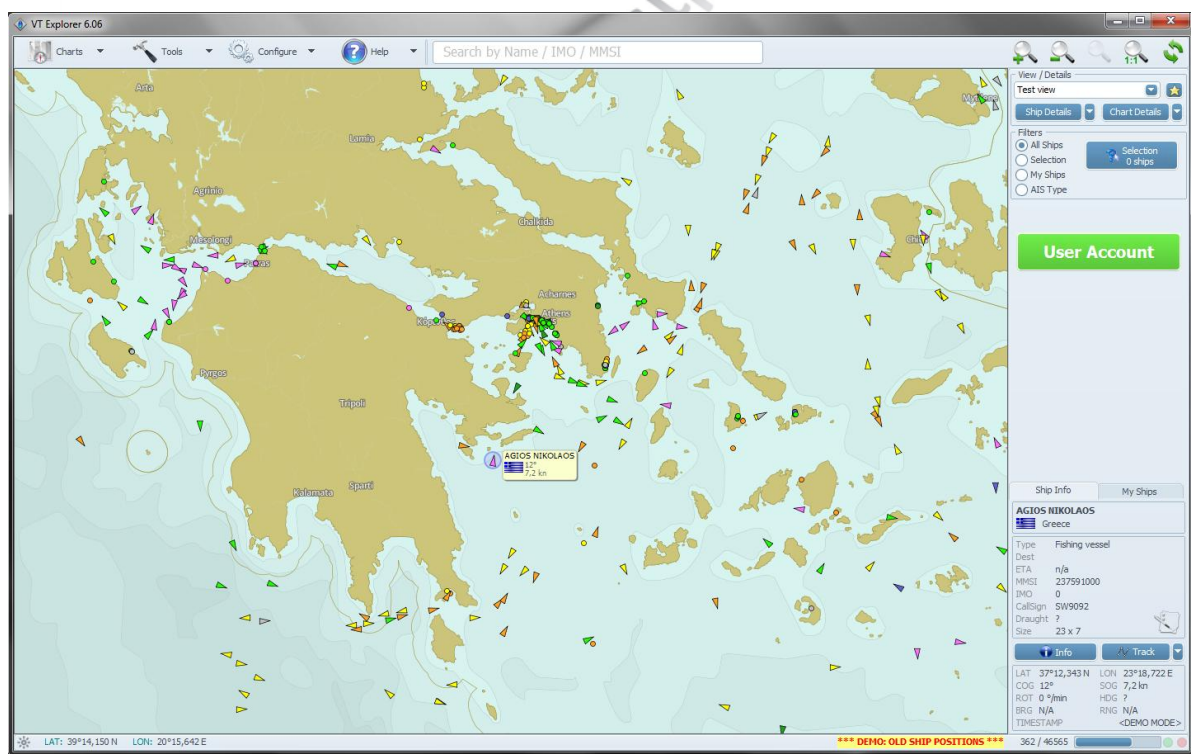
5.3 VT Explorer

Η συγκεκριμένη εφαρμογή είναι desktop διαδικτυακή εφαρμογή και όχι ιστοσελίδα. Η εφαρμογή αυτή αποτελεί ένα σύστημα για τη συλλογή την επεξεργασία και τη προβολή AIS δεδομένων σε πραγματικό χρόνο.

Ο κύριος στόχος της εφαρμογής αυτής είναι να παρέχει στο χρήστη δεδομένα κίνησης των πλοίων σε πραγματικό χρόνο. Χρήστες της εφαρμογής μπορεί να είναι οποιαδήποτε εταιρία, μεγάλοι οργανισμοί θαλάσσιου κλάδου καθώς και απλοί χρήστες που χρειάζονται αντίστοιχες πληροφορίες.

Οι συγκεκριμένοι χρήστες για να λάβουν όμως τις συνολικές πληροφορίες που παρέχει η εφαρμογή θα πρέπει να ανοίξουν λογαριασμό και να πληρώσουν συνδρομή. Μπορούν επίσης από την αντίστοιχη ιστοσελίδα (<http://www.vtexplorer.com/>) να κατεβάσουν το demo της εφαρμογής το οποίο παρέχει μια μικρή γκάμα πληροφορίας σε σχέση με την ολοκληρωμένη έκδοση της εφαρμογής.

Παρακάτω εμφανίζεται η εικόνα της εφαρμογής.



Εικόνα 5-3: Η εφαρμογή VT Explorer

Υπηρεσίες που παρέχει η εφαρμογή

- Δίνει τη δυνατότητα στον χρήστη να παρακολουθήσει συγκεκριμένα πλοία ή συγκεκριμένο στόλο πλοίων καθώς και να λαμβάνει ειδοποιήσεις μέσω email για τις αφίξεις και αναχωρήσεις τους.

- Για όλα τα πλοία που εμφανίζονται ο χρήστης μπορεί να λάβει πληροφορίες σχετικά με την τρέχουσα θέση τους, την πορεία του πλοίου καθώς και ιστορικά δεδομένα.
- Εκτός από τα παραπάνω δεδομένα ο χρήστης μπορεί να λάβει πληροφορίες σχετικά με τα τεχνικά στοιχεία του πλοίου καθώς και δεδομένα σχετικά με την εταιρία ιδιοκτησίας ή διαχείρισης του πλοίου.
- Παρέχει δυνατότητα προηγμένης αναζήτησης (advanced search) πλοίων. Ο χρήστης μπορεί να εντοπίσει κάποιο πλοίο αναζητώντας το από το όνομά του και τον τύπο του μέχρι το μέγεθός του, το βάρος του φορτίου του ή την ιδιοκτήτρια εταιρία.

5.4 Συγκριτικός πίνακας εφαρμογών

Παρακάτω εμφανίζεται ο πίνακας σύγκρισης των παραπάνω εφαρμογών καθώς και της εφαρμογής της παρούσας διπλωματικής εργασίας VTS (Vessel Tracking System).

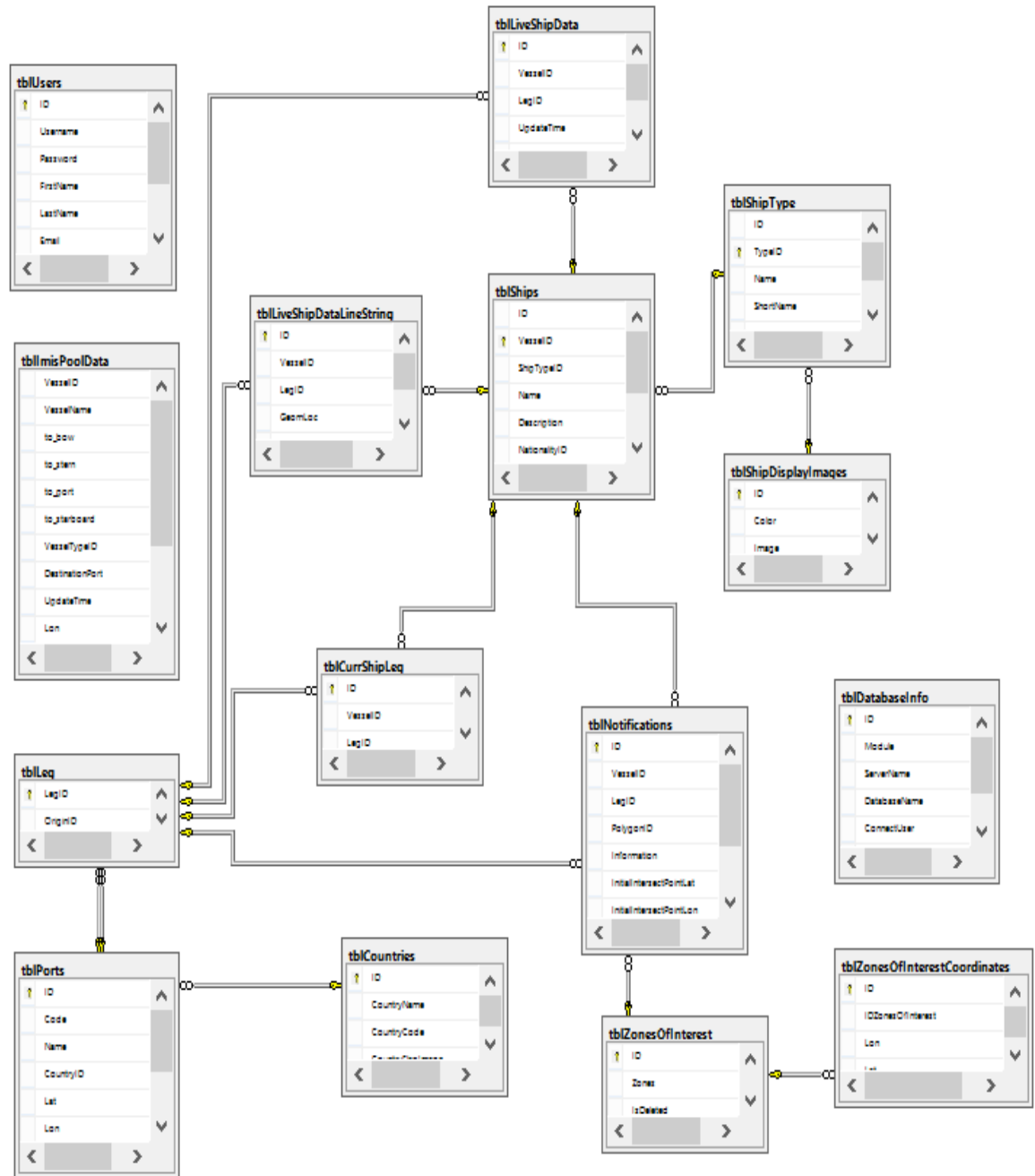
Υπηρεσίες Και Χαρακτηριστικά	Marintraffic (http://www.marinetraffic.com/ais/)	VesselFinder (http://www.vesselfinder.com/)	VT Explorer (VT Explorer 6.06)	VTS (Vessel Traffic System)
Εφαρμογή ιστοσελίδας	✓	✓	✗	✗
Desktop Διαδικτυακή Εφαρμογή	✗	✗	✓	✓
Δυνατότητα εμφάνισης θέσης πλοίων	✓	✓	✓	✓
Δυνατότητα εμφάνισης διαδρομής πλοίων	✓	✓	✓	✓
Δυνατότητα εμφάνισης πολλαπλών διαδρομών πλοίων	✗	✗	✗	✓
Δυνατότητα εμφάνισης ιστορικού της διαδρομής πλοίων	✓	✓	✓	✓
Εμφάνιση InfoWindow στον χάρτη για τις πληροφορίες πλοίων	✓	✗ (Εμφανίζει την πληροφορία δεξιά στην οθόνη)	✓ (Μέρος της πληροφορίας)	✓
Δυνατότητα εμφάνισης λιμανιών στο χάρτη	✓	✗	✗	✓
Δυνατότητα αναζήτησης (by name \ MMSI No etc.)	✓	✓	✓	✗
Πολλαπλή επιλογή φίλτρων (όνομα πλοίου, τύπος πλοίου, σημαία πλοίου, προορισμός)	✓	✗ (Φίλτρο μόνο ανά τύπο πλοίου)	✓	✓
Φίλτρο ημερομηνίας και ώρας	✗	✗	✗	✓

Επιλογή φίλτρου ανά leg (διαδρομή)	✗	✗	✗	✓
Εμφάνιση όλων των πλοίων των διαθέσιμων AIS στιγμάτων	✓	✓	✓	✗
Δυνατότητα εξατομίκευσης με χρήση προσωπικού στόλου πλοίων	✓	✓	✓	✓
Εμφάνιση περιοχών ενδιαφέροντος και ειδοποιήσεις εισόδου	✗	✗	✗	✓
Εμφάνιση σημείων ενδιαφέροντος (όπως λιμάνια) και ειδοποιήσεις εισόδου και εξόδου	✓	✗	✗	✓
Εμφάνιση πληροφοριών σχετικές με το πλοίο (φωτογραφίες, χρόνο κατασκευής κ.α.)	✓	✓	✓	✗
Εμφάνιση πληροφορίας τρέχοντος σημείου (ETA, Speed, Course/Heading)	✓	✓	✓	✓
Μπορεί ο χρήστης να καθορίσει το refresh rate του χάρτη	✗	✗	✗	✓
Δωρεάν και ολοκληρωμένη χρήση της εφαρμογής	✓	✗	✗	✓
Αριθμός πλοίων στη βάση τους	451.125	304.261	268.533	Προσθέτει ο διαχειριστής ανά πελάτη

6. Σχεδιασμός και Ανάλυση της Βάσης Δεδομένων

Σε αυτό το κομμάτι αναλύεται ο σχεδιασμός και η δομή της βάσης δεδομένων. Ο Σχεδιασμός έγινε με στόχο τον όσο το δυνατόν γρηγορότερο χρόνο εκτέλεση των εντελών μέσω της εφαρμογής. Επίσης για λόγους καλύτερης διαχείρισης και συντήρησης της εφαρμογής χρησιμοποιήθηκαν αρκετές Store Procedures για την ανάκτηση διάφορων λιστών καθώς και για τις περιπτώσεις καταχωρήσεων ή διαγραφής εγγραφών.

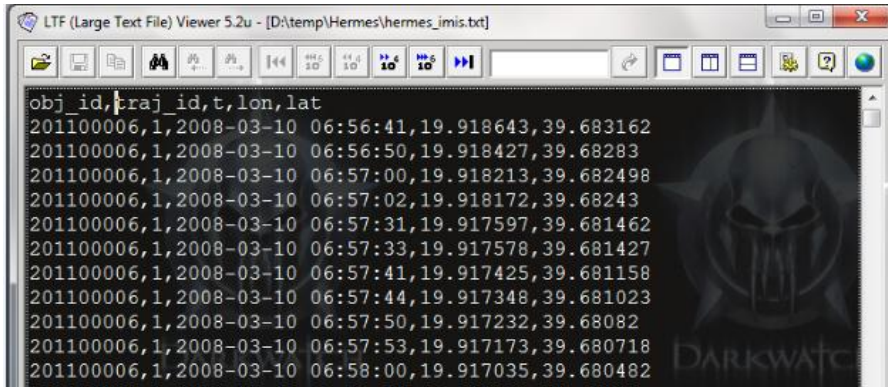
Παρακάτω εμφανίζεται το διάγραμμα της βάσης δεδομένων της εφαρμογής και οι σχέσεις μεταξύ των πινάκων.



Εικόνα 6-1: Σχεσιακό μοντέλο Βάσης Δεδομένων της εφαρμογής

6.1 Το σύνολο δεδομένων (dataset) της εφαρμογής

Στην παρούσα εργασία τα δεδομένα που χρησιμοποιήθηκαν ήταν κάποια σέτ από τα AIS δεδομένα της παρακάτω μορφής. Αυτά προήρθαν από μηνύματα τύπου 5 (Static and Voyage Related Data).



Εικόνα 6-2: Αρχική εικόνα AIS δεδομένων (dataset) που χρησιμοποιήθηκαν στην εφαρμογή

Για λόγους ευκολότερης διαχείρισης των δεδομένων αποφασίστηκε να μην ληφθούν και χρησιμοποιηθούν μεγάλοι όγκοι δεδομένα τα οποία θα εμφανίζονται στο χάρτη και τα οποία θα δυσχέραιναν τις συνολικές επιδόσεις του συστήματος. Έτσι χρησιμοποιήθηκαν δεδομένα συγκεκριμένου αριθμού τυχαίων πλοίων. Τα δεδομένα αυτά τα θεωρούμε ως το αρχικό pool για την εφαρμογή από το οποίο θα λαμβάνονται τα δεδομένα των πλοίων. Παρακάτω βλέπουμε τη μορφή των δεδομένων όπως έχουν εισαχθεί στην βάση δεδομένων την οποία θα αναλύσουμε παρακάτω.

VesselID	VesselName	to_bow	to_stem	to_port	to_starboard	VesselTypeID	DestinationPort	UpdateTime	Lon	Lat
211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:30:59.000	32.90717	34.28466
211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:03.000	32.90684	34.28433
211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:06.000	32.90662	34.2841
211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:12.000	32.90619	34.28366
211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:14.000	32.90601	34.28347
211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:24.000	32.90532	34.28277
211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:26.000	32.90521	34.28266
211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:28.000	32.90511	34.28255
211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:29.000	32.90499	34.28243
211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:30.000	32.90484	34.28228
211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:33.000	32.90454	34.28199
211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:34.000	32.90447	34.28194
211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:35.000	32.90434	34.28182
211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:39.000	32.90412	34.28161
211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:48.000	32.90349	34.28096

Εικόνα 6-3: Εικόνα των εισαγμένων AIS δεδομένων στη Βασή Δεδομένων

Τα στοιχεία (κολόνες) των δεδομένων αυτών αποτελούνται από τα παρακάτω:

Πεδίο	Περιγραφή
Vessel ID (MMSI)	Είναι η ταυτότητα του πλοίου. Στην πραγματικότητα είναι το MMSI (Maritime Mobile Service Identity) που αντιστοιχεί σε κάθε πλοίο. Είναι μια σειρά από 9 ψηφία τα οποία στέλνονται σε ψηφιακή μορφή μέσω των ραδιοσυχνοτήτων ως μοναδική ταυτότητα αναγνώρισης των πλοίων. Οι κωδικοί αυτοί έχουν δημιουργηθεί με τέτοιο τρόπο ώστε να μπορούν να χρησιμοποιηθούν και να σταλούν ακόμα και μέσω τηλεφώνου ή τέλεξ ώστε να μπορούν να καλούν τα πλοία αυτόματα.
Vessel Name	Αντιστοιχεί στο όνομα του πλοίου
To_bow	Είναι μέτρηση μήκους του μπροστινού μέρους του πλοίου σε μέτρα. Στην εφαρμογή μας δεν χρησιμοποιείται.
To_stream	Είναι μέτρηση μήκους του πίσω μέρους του πλοίου σε μέτρα. Στην εφαρμογή μας δεν χρησιμοποιείται
To_port	Είναι μέτρηση του μήκους του αριστερού μέρους του πλοίου σε μέτρα. Στην εφαρμογή μας δεν χρησιμοποιείται.
To_starboard	Είναι μέτρηση του μήκους του δεξιού μέρους του πλοίου σε μέτρα. Στην εφαρμογή μας δεν χρησιμοποιείται.
Vessel Type ID	Είναι η ταυτότητα του τύπου του πλοίου στο οποίο αντιστοιχεί το συγκεκριμένο πλοίο. Οι κωδικοί που χρησιμοποιούνται ως τύποι πλοίων είναι καθορισμένοι από συγκεκριμένα πρότυπα και αριθμούν από το 1 έως το 99. Μεγαλύτεροι κωδικού από το 99 δεν χρησιμοποιούνται αν και πολλές φορές στην πραγματική ζωή κάποιοι AIS αναμεταδότες τα χρησιμοποιούν. Στη περίπτωση αυτή (όπου vessel type id > 99) το AIS σύστημα το χειρίζεται ως 0 δηλαδή άγνωστος τύπος. Παρακάτω φαίνεται ο πίνακας με του τύπους των πλοίων και τους αντίστοιχους κωδικούς αυτών.
Destination Port	Είναι το λιμάνι το οποίο το πλοίο έχει ως προορισμό. Στα δεδομένα που έχουμε είναι σε αλφαριθμητική τιμή το οποίο όμως θα δούμε στην συνέχεια πως θα το μετασχηματίσουμε σε ακέραιο ώστε να είμαστε σε θέση να το χειριστούμε ως μοναδικό κωδικό.
Update Time	Είναι η ημερομηνία και ώρα κατά την οποία ελήφθη η συγκεκριμένη εγγραφή
Lon (longitude)	Το γεωγραφικό μήκος της προσδιοριζόμενης θέσης του πλοίου για την συγκεκριμένη εγγραφή.
Lat (Latitude)	Το γεωγραφικό πλάτος της προσδιοριζόμενης θέσης του πλοίου για την συγκεκριμένη εγγραφή

Ο συγκεκριμένος πίνακας χρησιμοποιείται ως ο αρχικός πίνακας των δεδομένων μας (pool table). Για τον λόγο αυτό έχει προστεθεί και μια επιπλέον στήλη η οποία θα ορίζει αν η κάθε εγγραφή του πίνακα έχει διαπεραστεί ή όχι.

Επίσης για λόγους καλύτερης διαχείρισης των δεδομένων μετασχηματίσαμε το DestinationPort σε DestinationPortID δηλαδή σε πεδίο με ακέραιο αριθμό. Αυτό έγινε ώστε να είμαστε διασφαλισμένοι κατά την λειτουργία της εφαρμογής όταν γίνεται αναφορά σε κάποιο DestinationPort ότι δεν θα υπάρχουν εγγραφές με ορθογραφικά λάθη ή ίδια ονόματα λιμανιών με διαφορετικές ονομασίες πχ. ANTVERP και ANTWERP.

Έτσι με τις προσθήκες αυτές έχουμε ως πίνακα συγκέντρωσης δεδομένων (pool data) έναν πίνακα με δεδομένα της παρακάτω μορφής.

VesselID	VesselName	to_bow	to_stem	to_port	to_starboard	VesselTypeID	DestinationPort	UpdateTime	Lon	Lat	IsParsed	DestinationPortID	
1	213083000	SVYATAYA ALEKSANDRA	105	40	5	15	70	TUZLA	2013-07-30 16:47:51.000	24.32918	37.46438	0	43
2	213083000	SVYATAYA ALEKSANDRA	105	40	5	15	70	TUZLA	2013-07-30 18:14:51.000	24.48794	37.66253	0	43
3	213083000	SVYATAYA ALEKSANDRA	105	40	5	15	70	TUZLA	2013-07-30 18:15:51.000	24.48914	37.6652	0	43
4	213083000	SVYATAYA ALEKSANDRA	105	40	5	15	70	TUZLA	2013-07-30 20:03:29.000	24.62835	37.94348	0	43
5	213083000	SVYATAYA ALEKSANDRA	105	40	5	15	70	TUZLA	2013-07-30 20:03:41.000	24.62869	37.94388	0	43
6	211695000	CAP ROCA	181	53	15	10	79	FELIXTOWE	2013-08-07 03:27:09.000	-3.54276	49.60953	0	12
7	211695000	CAP ROCA	181	53	15	10	79	FELIXTOWE	2013-08-07 03:27:20.000	-3.54159	49.60995	0	12
8	211695000	CAP ROCA	181	53	15	10	79	FELIXTOWE	2013-08-07 03:28:45.000	-3.5324	49.61318	0	12
9	211695000	CAP ROCA	181	53	15	10	79	FELIXTOWE	2013-08-07 03:29:20.000	-3.52867	49.61449	0	12
10	211695000	CAP ROCA	181	53	15	10	79	FELIXTOWE	2013-08-07 03:31:34.000	-3.51422	49.61958	0	12
11	211695000	CAP ROCA	181	53	15	10	79	FELIXTOWE	2013-08-07 03:31:45.000	-3.51305	49.61999	0	12
12	211695000	CAP ROCA	181	53	15	10	79	FELIXTOWE	2013-08-07 03:31:57.000	-3.51177	49.62044	0	12
13	211695000	CAP ROCA	181	53	15	10	79	FELIXTOWE	2013-08-07 03:32:09.000	-3.5106	49.62085	0	12
14	211695000	CAP ROCA	181	53	15	10	79	FELIXTOWE	2013-08-07 03:32:15.000	-3.50986	49.62111	0	12

Εικόνα 6-4: Τελική μορφή των AIS δεδομένων που χρησιμοποιήθηκαν στην εφαρμογή

6.2 Οι Πίνακες της εφαρμογής

Παρακάτω εμφανίζεται μια σύντομη περιγραφή σχετικά με τον λόγο ύπαρξης του κάθε πίνακα και των δεδομένα που περιέχει. Για τους κυριότερους πίνακες θα γίνει αναλυτικότερη ανάλυση στη συνέχεια.

Όνομα Πίνακα	Περιγραφή
tblCountries	Αποθηκεύει όλα τα ονόματα των χωρών που ο administrator του συστήματος έχει εισάγει στην βάση δεδομένων. Ο administrator μπορεί να εισάγει και να διαγράψει δεδομένα μέσω της εφαρμογής.
tblImisPoolData	Είναι ο αρχικός πίνακας που έχει αποθηκευμένες όλες τις εγγραφές τις οποίες θα διαβάσει η εφαρμογή.
tblDatabaseInfo	Αποθηκεύει την πληροφορία σχετικά με το όνομα του server και της βάσης δεδομένων καθώς και τα διαπιστευτήρια (credentials) για σύνδεση στον server.
tblPorts	Αποθηκεύει όλα τα ονόματα των λιμανιών που ο administrator του συστήματος έχει εισάγει στην βάση δεδομένων καθώς και τα σημεία του γεωγραφικού μήκους και πλάτους που καθορίζουν την θέση τους. Ο administrator μπορεί να εισάγει και να διαγράψει δεδομένα μέσω της εφαρμογής.
tblShipDisplayImages	Αποθηκεύει τη λίστα με τα ονόματα των χρώματα με το οποίο αντιπροσωπεύεται το κάθε πλοίο ανάλογα με τον τύπο του.
tblShips	Αποθηκεύει τα ονόματα των πλοίων που ο administrator του συστήματος έχει εισάγει στην βάση δεδομένων. Επίσης κρατάει πληροφορία σχετικά με την εθνικότητα του πλοίου και τον τύπο του καθώς και τον μοναδικό MMIS κωδικό που αντιστοιχεί σε κάθε πλοίο.
tblShipType	Περιέχει λίστα με τα ονόματα των τύπων των πλοίων. Η πληροφορία του συγκεκριμένου πίνακα παραμένει σταθερή κατά τη διάρκεια χρήσης της εφαρμογής.
tblUsers	Αποθηκεύει πληροφορία σχετικά με τα στοιχεία των χρηστών οι οποίοι μπορούν να εισέλθουν στην εφαρμογή και τα οποία ο administrator του συστήματος έχει εισάγει στην βάση δεδομένων. Ο administrator μπορεί να εισάγει και να διαγράψει δεδομένα μέσω της εφαρμογής.
tblZonesOfInterest	Αποθηκεύει με χρήση γεωγραφικών χωρικών δεδομένων τις συντεταγμένες των ζωνών ενδιαφέροντος οι οποίες έχουν εισαχθεί στο σύστημα. Η

	πληροφορία του συγκεκριμένου πίνακα παραμένει σταθερή κατά τη διάρκεια χρήσης της εφαρμογής.
tblZonesOfInterestCoordinates	Αποθηκεύει με χρήση σημείων γεωγραφικού μήκους και πλάτους τις συντεταγμένες των ζωνών ενδιαφέροντος οι οποίες έχουν εισαχθεί στο σύστημα. Η πληροφορία του συγκεκριμένου πίνακα παραμένει σταθερή κατά τη διάρκεια χρήσης της εφαρμογής.
tblLeg	Αποθηκεύει με νέο id τα ζεύγη εγγραφών του λιμανιού αναχώρησης και του λιμανιού προορισμού
tblCurrShipLeg	Αποθηκεύει την σχέση μεταξύ των πλοίων και των legs (του παραπάνω πίνακα)
tblLiveShipData	Αποτελεί τον βασικό πίνακα της εφαρμογής. Η εφαρμογή αποτυπώνει τις εγγραφές που εμπεριέχονται μόνο σε αυτόν τον πίνακα. Εκεί αποθηκεύονται οι εγγραφές των σημείων των πλοίων από τον tblImisPoolData πίνακα ανά καθορισμένο χρόνο.
tblLiveShipDataLineString	Αποθηκεύει σε μορφή γεωγραφικών χωρικών δεδομένων (linestring) τα σημεία που εμπεριέχονται στον πίνακα tblLiveShipData, ανά πλοίο και ανά διαδρομή (leg)
tblNotifications	Αποθηκεύει τις ειδοποιήσεις που εμφανίζονται στον χρήστη στη περίπτωση που ένα πλοίο (linestring από το tblLiveShipDataLineString) περάσει μέσα σε μία ζώνη ενδιαφέροντος (polygon από το tblZonesOfInterest).

6.3 Ανάλυση Βασικών Πινάκων της Εφαρμογής

6.3.1 Πίνακας tblImisPoolData

Όπως εμφανίζεται και στο διάγραμμα της βάσης (και όπως έχει αναφερθεί και σε προηγούμενο κεφάλαιο) ο αρχικός πίνακας της εφαρμογής είναι ο tblImisPoolData. Σε αυτόν τον πίνακα έχουν καταγραφεί όλες οι εγγραφές που θα χρησιμοποιηθούν στην πορεία της εφαρμογής. Ουσιαστικά αυτός ο πίνακας αντικαθιστά στον πραγματικό κόσμο τα πραγματικά δεδομένα τα οποία αποστέλλουν τα πλοία μέσω συστημάτων AIS και με χρήση GPS στις ναυτιλιακές εταιρίες (αφού δεν θα μπορούσε στην παρούσα εφαρμογή να λαμβάνουμε live δεδομένα από πλοία).

Τα δεδομένα αυτά περιέχουν το MMSI number (ή VesselID) το όνομα του πλοίου, το λιμάνι προορισμού και το γεωγραφικό μήκος και πλάτος της κάθε ληφθείσας θέσης του πλοίου. Για τη συγκεκριμένη εφαρμογή στα υπάρχοντα δεδομένα πραγματοποιήσαμε τις παρακάτω τροποποιήσεις όπως αναφέρθηκαν και παραπάνω:

- Είσοδος πεδίου IsParsed: Το συγκεκριμένο πεδίο λειτουργεί ως σημαία (flag) για να αναγνωρίζει το σύστημα ποιες από τις εγγραφές έχουν μεταφερθεί/διαβαστεί ώστε να μην τις ξανα-χρησιμοποιεί.
- Είσοδος πεδίου DestinationPortID. Λόγω του ότι τα αρχικά δεδομένα εμπεριείχαν πολλά λάθη στο λεκτικό της DestinationPort κολόνας, το πεδίο αυτό εισήχθη στο πίνακα ώστε να αναχθούν σε συγκεκριμένα ids, συγκεκριμένα λεκτικά από το πεδίο DestinationPort. Το πεδίο αυτό σχετίζεται με το πεδίο του πίνακα tblports με σχέση ένα προς πολλά.
- Μετατροπή του UpdateTime πεδίο στον τρέχοντα χρόνο. Λόγω του ότι οι εγγραφές αυτές αντιστοιχούν σε παλαιότερα δεδομένα πλοίων στο συγκεκριμένο πεδίο υπήρχαν εγγραφές παλαιότερου χρόνου. Έτσι για την αληθοφάνεια της εφαρμογής (ώστε να υπάρχουν παλαιότερα αλλά και νέα στοιχεία εγγραφών και να μπορούν να ανανεώνονται τα δεδομένα της εφαρμογής με νέα) προστέθηκε στον χρόνο που υπήρχε ένα διάστημα κάποιον χρόνων ώστε να υπάρχουν διαθέσιμες εγγραφές σε λογικά και βολικά στο συγκεκριμένο χρόνο ανάπτυξης της εφαρμογής, εύρη ημερομηνίας.

VesselID	VesselName	to_bow	to_stem	to_port	to_starboard	VesselTypeID	DestinationPort	UpdateTime	Lon	Lat	IsParsed	DestinationPortID	
1	211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:30:59.000	32.90717	34.28466	0	25
2	211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:03.000	32.90684	34.28433	0	25
3	211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:06.000	32.90662	34.2841	0	25
4	211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:12.000	32.90619	34.28366	0	25
5	211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:14.000	32.90601	34.28347	0	25
6	211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:24.000	32.90532	34.28277	0	25
7	211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:26.000	32.90521	34.28266	0	25
8	211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:28.000	32.90511	34.28255	0	25
9	211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:29.000	32.90499	34.28243	0	25
10	211695000	CAP ROCA	181	53	15	10	79	LIMASSOL	2013-09-21 21:31:30.000	32.90484	34.28228	0	25

Εικόνα 6-5: Δεδομένα του πίνακα tblImisPoolData

6.3.2 Πίνακας tblLiveShipData

Αποτελεί τον κυριότερο πίνακα της εφαρμογή αφού ό τι εμφανίζεται στην οθόνη της εφαρμογής ουσιαστικά έχει προέλθει από τα στοιχεία αυτού του πίνακα. Είναι ο πίνακας που κρατάει τα live δεδομένα της εφαρμογής δηλαδή τα δεδομένα εκείνα που λαμβάνονται για τα πλοία ανά συγκεκριμένη χρονική περίοδο και στη συνέχεια αποτυπώνονται στο χάρτη της εφαρμογής.

Περιέχει ουσιαστικά μόνο τα απαραίτητα δεδομένα από τον αρχικό πίνακα tblImisPoolData. Για την ακρίβεια κρατάει το MMSI number, το UpdateTime της κάθε εγγραφής και το Lat και Lon των εγγραφών. Σε αυτές της εγγραφές προστίθεται και το πεδίο του LegID το οποίο ορίζει το Leg (διαδρομή) στην οποία βρίσκεται τη δεδομένη χρονική στιγμή το πλοίο. Παρακάτω θα δούμε πως το LegID συμπληρώνεται.

ID	VesselID	LegID	UpdateTime	Lat	Lon	
1	459363	211695000	81	2013-09-21 21:30:59.000	34.2846603393555	32.907169342041
2	459364	211695000	81	2013-09-21 21:31:03.000	34.2843284606934	32.9068412780762
3	459365	211695000	81	2013-09-21 21:31:06.000	34.2840995788574	32.9066200256348
4	459366	211695000	81	2013-09-21 21:31:12.000	34.2836608886719	32.9061889648438
5	459367	211695000	81	2013-09-21 21:31:14.000	34.2834701538086	32.9060096740723
6	459368	211695000	81	2013-09-21 21:31:24.000	34.2827682495117	32.9053192138672
7	459369	211695000	81	2013-09-21 21:31:26.000	34.2826614379883	32.9052085876465
8	459370	211695000	81	2013-09-21 21:31:28.000	34.2825508117676	32.9051094055176
9	459371	211695000	81	2013-09-21 21:31:29.000	34.2824287414551	32.9049911499023
10	459372	211695000	81	2013-09-21 21:31:30.000	34.2822799682617	32.9048385620117

Εικόνα 6-6: Δεδομένα του πίνακα tblLiveShipData

Ο πίνακας αυτός γεμίζει μέσω της εκτέλεσης 2 βασικών Store Procedure της 'SM_InitializeImisPoolData' και της 'SM_ReadPoolData' η λειτουργία των οποίων αναλύεται παρακάτω.

6.3.3 Πίνακας tblLiveShipDataLineString

Αποθηκεύει τα ίδια ακριβώς δεδομένα με τον πίνακα tblLiveShipData με τη διαφορά ότι χρησιμοποιεί χωρικά δεδομένα. Έτσι λοιπόν σε πολλαπλές εγγραφές που αντιπροσωπεύουν μια διαδρομή ενός πλοίου στον πίνακα tblLiveShipData στον συγκεκριμένο πίνακα αποθηκεύονται ως μια εγγραφή που αντιστοιχεί σε συγκεκριμένο πλοίο για συγκεκριμένο Leg και έχει την μορφή χωρικών δεδομένων.

ID	VesselID	LegID	GeomLoc	GeogLoc	GeogLocText
1	106482	213083000	44	0xE61000000104A7000000DCD781734654384096218E7571BB...	0xE61000000104A700000096218E7571BB4240DCD781734654...
2	106483	211695000	45	0xE61000000104210600009D90C8E92570CC0BC74931804C...	0xE61000000104210600008C74931804CE48409D90C8E925...
3	106484	211695000	46	0xE610000001046B000000B476DB85E63AF18F2497FF907E13...	0xE610000001046B0000002497FF907E134940B476DB85E63A...
4	106485	211695000	47	0xE61000000104CF000000DF89592F8672EFBF93D7958A815...	0xE61000000104CF000000D93D7958A8154940DF89592F8672...
5	106486	212724000	48	0xE610000001046A070000041C42959A3DE5BF1D5A643BDf2...	0xE610000001046A0700001D5A643BDf2F4940041C42959A3...
6	106487	212784000	49	0xE61000000104D30300009F71E1404816E7BFEC2FB8270F33...	0xE61000000104D3030000EC2FB8270F3349409F71E1404816...
7	106488	211695000	50	0xE61000000104CA0E00002C7DE882FA96993F68226C787A3...	0xE61000000104CA0E000068226C787A3D49402C7DE882FA9...
8	106489	211444000	51	0xE61000000104CD0A0000DC4603780B24C4BF17B7D100DE4...	0xE61000000104CD0A000017B7D100DE424940DC4603780B2...
9	106490	210392000	52	0xE6100000010488000000A27B8585183A935F07CE195142...	0xE61000000104880000005F07CE1951424940FA27B8585183...
10	106491	210392000	53	0xE61000000104F501000045F0BF95ECC8C8BF05A3923A013...	0xE61000000104F501000005A3923A013D494045F0BF95ECC8...
11	106492	210392000	54	0xE61000000104310100007AFCDEA63FFB58F273108AC1C3...	0xE6100000010431010000273108AC1C3249407AFCDEA63FF...
12	106493	210392000	55	0xE61000000104E0060000E15E3FC4D28EFBF41F163CC5D2...	0xE61000000104E006000041F163CC5D2849400E15E3FC4D2...

Εικόνα 6-7: Δεδομένα του πίνακα tblLiveShipDataLineString

6.3.4 Πίνακας tblZonesOfInterest & tblZonesOfInterestCoordinates

Στους πίνακες αυτούς όπως έχει αναφερθεί παραπάνω αποθηκεύονται τα δεδομένα αναφορικά με τα γεωγραφικά χωρικά δεδομένα των περιοχών ενδιαφέροντος. Πιο συγκεκριμένα ο πίνακας tblZonesOfInterest περιέχει τα δεδομένα αυτά σε χωρική (spatial) μορφή και ο πίνακας tblZonesOfInterestCoordinates περιέχει τα ίδια δεδομένα αλλά με Lat Lon σημεία ανά εγγραφή για κάθε ζώνη ενδιαφέροντος.

Το ερώτημα που προκύπτει είναι γιατί δημιουργήθηκε η ανάγκη να κρατήσουμε τα ίδια δεδομένα και στις δύο μορφές. Η απάντηση στο ερώτημα αυτό έχει να κάνει με την αρχιτεκτονική του συστήματος και θα επεξηγηθεί αναλυτικότερα στο αντίστοιχο κεφάλαιο. Σαν μια γρήγορη αναφορά, αυτό γίνεται λόγω του ότι στη βάση δεδομένων γίνεται χρήση των spatial δεδομένων για να υπολογιστούν διάφορα στοιχεία που απαιτούνται ενώ αντίθετα τα ίδια δεδομένα για να εμφανιστούν στον χάρτη μέσω JavaScript πρέπει να είναι της μορφής Lat Lon.

Τα δεδομένα του συγκεκριμένου πίνακα αυτά στην παρούσα έκδοση της εφαρμογής δεν μπορούν να αλλάξουν ούτε από τον απλό χρήστη του συστήματος ούτε από τον administrator. Έχουν εισαχθεί χειροκίνητα και για να εισαχθούν νέα δεδομένα θα πρέπει να γίνει με χρήση T-SQL.

ID	Zones	IsDeleted	Name
1	0xE610000001040B000000569DD5027B504240EBC726F911...	0	Rhodes-Mammaris Strait
2	0xE6100000010405000000627ECE589B224240000000B0F32...	0	Chania - Milos Area
3	0xE610000001040700000015C8AB976B964240000000B0D38...	0	Myrtoon Sea
4	0xE6100000010406000000B96ED1E1F622494000000080C2B...	1	English Channel Middle
5	0xE61000000104050000002FA36F779EE5484000000000AB...	0	English Channel Door

Εικόνα 6-8: Δεδομένα του πίνακα tblZonesOfInterest

	ID	IDZonesOfInterest	Lon	Lat	IsDeleted
1	1	1	27.95609	36.45332	0
2	2	1	28.01376	36.43344	0
3	3	1	28.09753	36.43785	0
4	4	1	28.19366	36.46658	0
5	5	1	28.24585	36.48645	0
6	6	1	28.31863	36.4478	0
7	7	1	28.37494	36.47652	0
8	8	1	28.19366	36.53061	0
9	9	1	28.23074	36.62875	0
10	10	1	28.05908	36.53171	0
11	11	1	27.95609	36.45332	0
12	18	3	24.14825	36.27037	0
13	19	3	24.19769	35.87521	0
14	20	3	24.53277	35.80396	0

Εικόνα 6-9: Δεδομένα του πίνακα tblZonesOfInterestCoordinates

6.3.5 Πίνακας tblShips

Είναι ο πίνακας στον οποίο αποθηκεύονται τα πλοία. Περιέχεται το MMSI number (κάτω από τη στήλη VesselID) και το οποίο είναι το πρωτεύον κλειδί του πίνακα. Μέσω αυτού συνδέεται ο πίνακας με όλους του υπόλοιπους πίνακες που σχετίζονται με την οντότητα του πλοίου.

ID	VesselID	ShipTypeID	Name	Description	NationalityID	ShortName	isDeleted
1	201100015	70	DARIL	DARIL	5	DAR	0
2	201100019	70	M/V ELENI	M/V ELENI	3	ELE	0
3	205475000	0	UNION JADE	UNION JADE	10	UNJAD	0
4	205535000	70	LOWLANDS ORCHID	LOWLANDS ORCHID	10	LOWORC	0
5	207017000	70	GEORGI GRIGOROV	GEORGI GRIGOROV	6	GEG	0
6	209255000	31	EDT PROTEA	EDT PROTEA	1	EPR	0
7	209373000	70	PONTODAMON	PONTODAMON	1	PONT	0
8	209640000	70	MEDCASPIAN	MEDCASPIAN	1	MED	0
9	210392000	70	EWL CARIBBEAN	EWL CARIBBEAN	5	EWL	0
10	210689000	72	CMA CGM PAPAGAYO	CMA CGM PAPAGAYO	5	CCP	0
11	210737000	70	ANASTASIA	ANASTASIA	1	ANA	0

Εικόνα 6-10: Δεδομένα του πίνακα tblShips

6.3.6 Πίνακας tblNotifications

Αποθηκεύει όλη την απαραίτητη πληροφορία που χρειάζεται ώστε να παρουσιάσει στον χρήστη το σημείο κατά το οποίο ένα πλοίο εισχώρησε σε κάποιο σημείο ενδιαφέροντος. Κατά το διάστημα που ελέγχεται αν έχουμε νέα διαθέσιμα δεδομένα για να εισαχθούν στον live πίνακα (και αφού εισαχθούν αυτά) γίνεται ο έλεγχος αν υπάρχει κάποιο πλοίο κατά την διαδρομή του οποίου περνάει μέσα σε κάποιο σημείο

ενδιαφέροντος. Αν βρεθεί κάποιο τέτοιο σημείο τότε καταγράφεται στο συγκεκριμένο πίνακα. Στη συνέχεια και ανά ορισμένη χρονική διάρκεια γίνεται έλεγχος αν υπάρχει κάποιο νέο τέτοιο σημείο ώστε να εμφανιστεί στον κατάλληλο σημείο της εφαρμογής προς ενημέρωση του χρήστη.

ID	VesselID	LegID	PolygonID	Information	InitialIntersectPointLat	InitialIntersectPointLon	DateTime	isDiscard	
1	21	212570000	76	4	Please note that Vessel 'SEA LEOPARD' has entered a...	37.1951976750207	24.2267704314294	1900-01-01 00:00:00.000	0
2	22	213083000	44	4	Please note that Vessel 'SVYATAYA ALEKSANDRA' h...	37.2275996228214	24.1993757784457	1900-01-01 00:00:00.000	0
3	23	213083000	77	4	Please note that Vessel 'SVYATAYA ALEKSANDRA' h...	37.2185281832325	24.2128283270076	1900-01-01 00:00:00.000	0
4	24	211695000	67	6	Please note that Vessel 'CAP ROCA' has entered a Zo...	49.6129183737803	-3.8529216888441	1900-01-01 00:00:00.000	0

Εικόνα 6-11: Δεδομένα του πίνακα tblNotifications

6.3.7 Πίνακας tblShipType

Για λόγους καλύτερης διαχείρισης των δεδομένων δημιουργήσαμε τον παραπάνω πίνακα κάνοντας την αντιστοίχιση του ακεραίου με το αντίστοιχο αλφαριθμητικό από τον παρακάτω πίνακα. Στο πίνακα αυτό εμφανίζονται όλοι οι κωδικοί ανά τύπο πλοίου που έχουν προσδιοριστεί σύμφωνα με τα διεθνή πρότυπα και χρησιμοποιούνται από τα αντίστοιχα του AIS συστήματα.

ID	TypeID	Name	ShortName	isDeleted	ShipDisplayImagesID
1	0	Not available	Not Avail	0	1
2	2	Future use	Future Use	0	1
3	29	Wing in ground	WIG	0	4
4	31	Towing	TOW	0	3
5	33	Dredging/underwater OPS	OPS	0	6
6	34	Diving OPS	OPS	0	6
7	36	Sailing	SAI	0	5
8	52	Tug	TUG	0	3
9	70	Cargo-All Types	CARG	0	2
10	71	Cargo-Haz Cat A	CRGHA	0	2
11	72	Cargo-Haz Cat B	CRGHB	0	2
12	77	Cargo-For future use	CRGF	0	2
13	79	Cargo-No info	CRGNI	0	2
14	91	Unknown-Haz Cat A	UnHazA	0	1
15	99	Unknown-No info	OPS	0	1

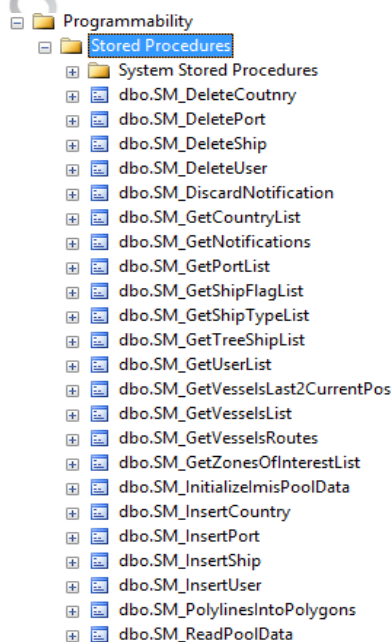
Εικόνα 6-12: Δεδομένα του πίνακα tblShipType

Επομένως κάνοντας μια απλή αντιστοιχία του τύπου με τον κωδικό έχουμε πλέον την σχέση μεταξύ του id και της περιγραφής του για να τα χρησιμοποιήσουμε στην εφαρμογή μας.

Code	Ship & Cargo Classification	Code	Ship & Cargo Classification
0	Not available (default)	61	Passenger, Hazardous category A
1-19	Reserved for future use	62	Passenger, Hazardous category B
20	Wing in ground (WIG), all ships of this type	63	Passenger, Hazardous category C
21	Wing in ground (WIG), Hazardous category A	64	Passenger, Hazardous category D
22	Wing in ground (WIG), Hazardous category B	65	Passenger, Reserved for future use
23	Wing in ground (WIG), Hazardous category C	66	Passenger, Reserved for future use
24	Wing in ground (WIG), Hazardous category D	67	Passenger, Reserved for future use
25	Wing in ground (WIG), Reserved for future use	68	Passenger, Reserved for future use
26	Wing in ground (WIG), Reserved for future use	69	Passenger, No additional information
27	Wing in ground (WIG), Reserved for future use	70	Cargo, all ships of this type
28	Wing in ground (WIG), Reserved for future use	71	Cargo, Hazardous category A
29	Wing in ground (WIG), Reserved for future use	72	Cargo, Hazardous category B
30	Fishing	73	Cargo, Hazardous category C
31	Towing	74	Cargo, Hazardous category D
32	Towing, length exceeds 200m or breadth exceeds 25m	75	Cargo, Reserved for future use
33	Dredging or underwater ops	76	Cargo, Reserved for future use
34	Diving ops	77	Cargo, Reserved for future use
35	Military ops	78	Cargo, Reserved for future use
36	Sailing	79	Cargo, No additional information
37	Pleasure Craft	80	Tanker, all ships of this type
38	Reserved	81	Tanker, Hazardous category A
39	Reserved	82	Tanker, Hazardous category B
40	High speed craft (HSC), all ships of this type	83	Tanker, Hazardous category C
41	High speed craft (HSC), Hazardous category A	84	Tanker, Hazardous category D
42	High speed craft (HSC), Hazardous category B	85	Tanker, Reserved for future use
43	High speed craft (HSC), Hazardous category C	86	Tanker, Reserved for future use
44	High speed craft (HSC), Hazardous category D	87	Tanker, Reserved for future use
45	High speed craft (HSC), Reserved for future use	88	Tanker, Reserved for future use
46	High speed craft (HSC), Reserved for future use	89	Tanker, No additional information
47	High speed craft (HSC), Reserved for future use	90	Other Type, all ships of this type
48	High speed craft (HSC), Reserved for future use	91	Other Type, Hazardous category A
49	High speed craft (HSC), No additional information	92	Other Type, Hazardous category B
50	Pilot Vessel	93	Other Type, Hazardous category C
51	Search and Rescue vessel	94	Other Type, Hazardous category D
52	Tug	95	Other Type, Reserved for future use
53	Port Tender	96	Other Type, Reserved for future use
54	Anti-pollution equipment	97	Other Type, Reserved for future use
55	Law Enforcement	98	Other Type, Reserved for future use
56	Spare - Local Vessel	99	Other Type, no additional information
57	Spare - Local Vessel		
58	Medical Transport		
59	Noncombatant ship according to PR Resolution No. 18		
60	Passenger, all ships of this type		

6.4 Ανάλυση βασικών Store Procedures

Παρακάτω βλέπουμε την λίστα με τις Store Procedures που έχουν δημιουργηθεί για την παρούσα εφαρμογή.



Εικόνα 6-13: Η λίστα των Store Procedures που δημιουργήθηκαν και χρησιμοποιήθηκαν στην παρούσα εφαρμογή

Κάποιες από τις παραπάνω Store Procedures χρησιμοποιούνται στη διαχείριση διαδικαστικών ζητημάτων της εφαρμογής όπως να κάνουν εισαγωγή ή διαγραφή διάφορων εγγραφών οντοτήτων στη βάση, ενώ κάποιες έχουν κομβικό ρόλο στην λειτουργία της εφαρμογής. Παρακάτω εξηγείται η λειτουργία μερικών από αυτές ενώ γίνεται και μια ανάλυση στα κυριότερα σημεία του T-SQL κώδικα που χρησιμοποιήθηκε..

6.4.1 SM_InitializeImisPoolData

Μέσω της παραπάνω Store Procedure πραγματοποιείται η αρχικοποίηση των δεδομένων της εφαρμογής. Χρειαζόμαστε την αρχικοποίηση ώστε να συλλέξουμε τα δεδομένα της εφαρμογής μέχρι την δεδομένη χρονική στιγμή. Δηλαδή λόγω της απουσίας ενός server που θα τραβάει δεδομένα 24 ώρες το 24ωρο χρειαζόμαστε να τραβήξουμε τα δεδομένα που συλλέχτηκαν σε χρονικές στιγμές προηγούμενες της χρονικής στιγμής χρήσης της εφαρμογής. Αυτό επιτυγχάνεται με τον παρακάτω κομμάτι της Store Procedure. Περαιτέρω ανάλυση γίνεται στο κεφάλαιο της αρχιτεκτονικής της εφαρμογής.

```
select VesselID, DestinationPortID, UpdateTime, Lon, Lat
FROM tblImisPoolData
WHERE UpdateTime <= @CURRENTDATETIME1 AND IsParsed = 0
```

Έτσι κατά την εκτέλεση της συγκεκριμένης Store Procedure συλλέγονται δεδομένα από τον πίνακα tblImisPoolData και αποθηκεύονται στον tblLiveShipData. Τα δεδομένα αυτά έχουν ημερομηνία (UpdateTime) μικρότερη ή ίση με την χρονική στιγμή της εκτέλεσης (@CURRENTDATETIME1). Επίσης δεν έχουν ξανά διαβαστεί, συνθήκη η οποία ικανοποιείται με το IsParsed = 0.

Μια άλλη διαδικασία που εκτελείται μέσω της συγκεκριμένης Store Procedure είναι η αντιστοίχιση της κάθε εγγραφής σε συγκεκριμένο LegID. Αυτό πραγματοποιείται με τον παρακάτω συλλογισμό.

Αρχικά ελέγχουμε τον προορισμό της κάθε εγγραφής για το συγκεκριμένο πλοίο. Αν για τον προορισμό αυτό δεν υπάρχει εγγραφή στον πίνακα tblLeg τότε δημιουργείται μια νέα εγγραφή και το id της σώζεται στο πεδίο LegID του tblLiveShipData. Αν υπάρχει εγγραφή τότε δεν δημιουργείται νέα στο πίνακα tblLeg και απλά σώζεται στον πίνακα tblLiveShipData. Στη συνέχεια εξετάζεται η επόμενη εγγραφή, αν έχει ίδιο λιμάνι προορισμού με την επόμενη τότε πάλι δεν δημιουργείται εγγραφή στο πίνακα tblLeg. Αν όμως το λιμάνι προορισμού είναι διαφορετικό (για το ίδιο πάντα πλοίο) τότε δημιουργείται μια διαφορετική εγγραφή στον πίνακα tblLeg με λιμάνι αναχώρησης το προηγούμενο LegID και λιμάνι προορισμού το νέο LegID. Αυτό το νέο LegID είναι αυτό που σώζεται πλέον στο πίνακα tblLiveShipData.

Όλα τα παραπάνω αποτελούν μέρος ενός κέρσορα (SQL Cursor) Παρακάτω εμφανίζεται ο T-SQL κώδικας του κέρσορα μέσω του οποίου επιτυγχάνεται αυτό καθώς και το διάγραμμα της συνολικής λογικής λειτουργίας της Store Procedure.

```

declare @IDs varchar (50)
declare @legID int
declare @PreLegID int

DECLARE db_cursor CURSOR FOR
SELECT ID
from #tmp
order by UpdateTime

OPEN db_cursor
FETCH NEXT FROM db_cursor into @IDs

WHILE @@FETCH_STATUS = 0
BEGIN

    Declare @tmpVesselID int
    Declare @tmpDestinationPortID int
    declare @OrigID int
    Set @tmpVesselID = (select vesselID from #tmp where id = @IDs)
    Set @tmpDestinationPortID = (select DestinationPortID from #tmp
    where id = @IDs)
    SET @OrigID = ISNULL((Select DestinationID from tblleg where Legid =
    (select LegID from tblcurrShipLeg where VesselID = @tmpVesselID
    and IsCurrent = 1)),45)
    SET @legID = (select LegID from tblLeg where OriginID = @OrigID and
    DestinationID = @tmpDestinationPortID)

    IF not exists (select LegID from tblleg where destinationID =
    @tmpDestinationPortID and OriginID = @OrigID)
    BEGIN

        If (@OrigID <> @tmpDestinationPortID)
        Begin
            Insert into tblleg (OriginID, DestinationID)
            Values (@OrigID , @tmpDestinationPortID )

            SET @legID = (select LegID from tblLeg where OriginID =
            @OrigID and DestinationID = @tmpDestinationPortID)
        END
    END

    IF not exists (select vesselID from tblcurrshipleg where vesselID =
    @tmpVesselID and LegID = @legID ) and (@legID IS NOT NULL)
    BEGIN

        Update tblCurrShipLeg set IsCurrent = 0
        where VesselID = @tmpVesselID

        insert into tblcurrshipleg (vesselID, LegID, IsCurrent)
        values (@tmpVesselID,@legID, 1)

    END

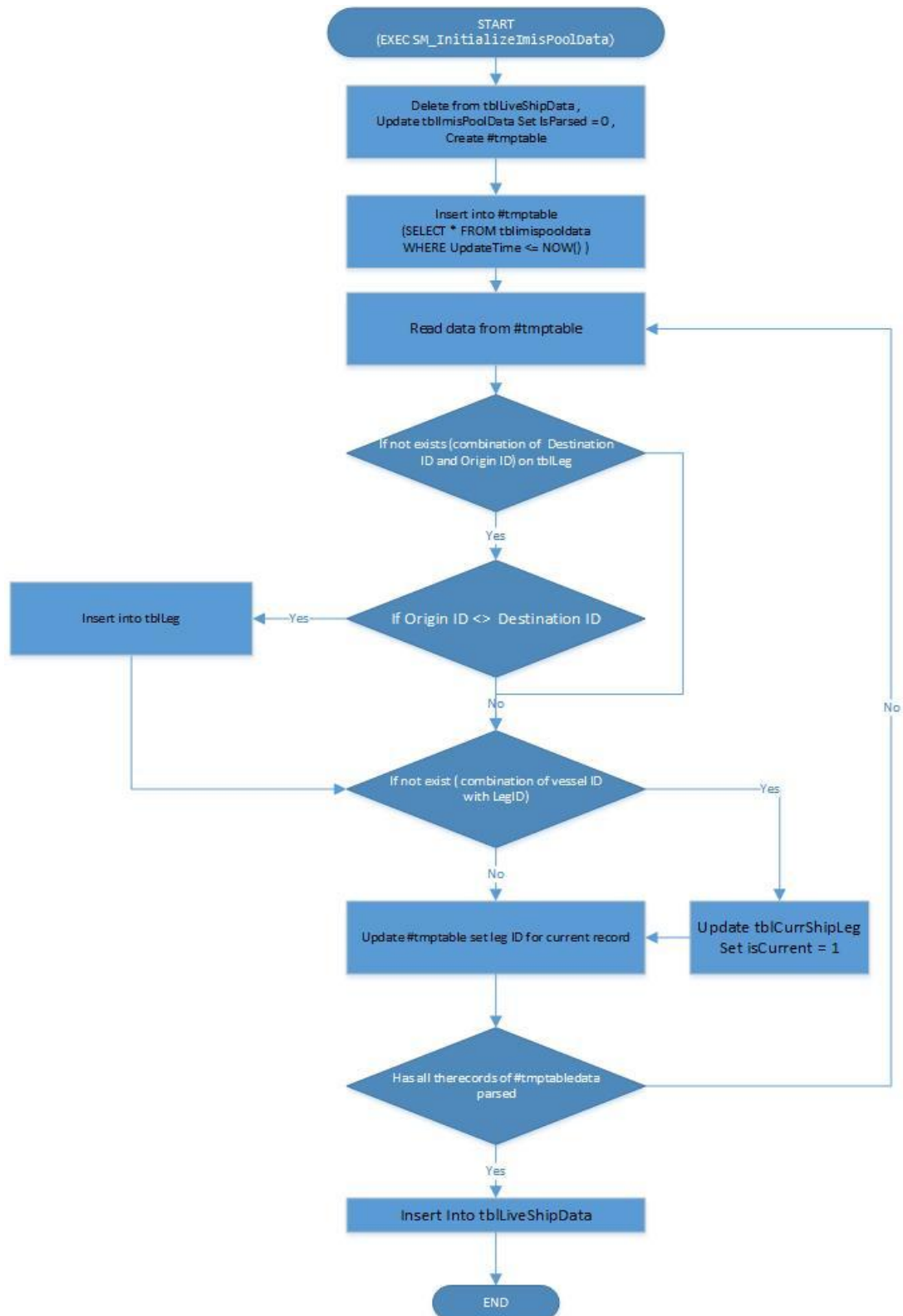
    update #tmp set LegID = (select legid from tblLeg where LegID =
    (select LegID from tblCurrShipLeg where VesselID = @tmpVesselID
    and IsCurrent = 1))
    where id = @IDs

    FETCH NEXT FROM db_cursor into @IDs
    END

CLOSE db_cursor
DEALLOCATE db_cursor;

```

Η παραπάνω διεργασία εμφανίζεται και στο παρακάτω διάγραμμα:



Εικόνα 6-14: Διάγραμμα λειτουργίας της Store Procedure SM_InitializeImisPoolData

Η συγκεκριμένη Store Procedure αποτελεί βήμα (step) ενός Agent Job ‘GISInitializePoolData’ αλλά δεν έχει οριστεί κάποια χρονική συγκεκριμένη περίοδος ανάλογα με την οποία θα εκτελείται. Αυτό γιατί χρησιμοποιείται για την αρχικοποίηση της εφαρμογής και επομένως ο χρήστης διαχειριστής την εκτελεί χειροκίνητα εκτελώντας το αντίστοιχο Job κατά την πρώτη φορά λειτουργίας του συστήματος.

6.4.2 SM_ReadPoolData.

Η συγκεκριμένη Store Procedure αποτελεί μέρος ενός βήματος (step) του Agent Job ‘GISGettingData’. Το job αυτό έχει οριστεί να εκτελείται ανά 10 λεπτά εάν ο Agent «τρέχει» (είναι ενεργοποιημένος).

Η λειτουργία της Store Procedure αυτής είναι να διαβάζει δεδομένα από τον πίνακα tblImisPoolData και να τα μεταφέρει στον live πίνακα της εφαρμογής δηλαδή τον tblLiveShipData.

Τα δεδομένα που μεταφέρει πρέπει να ικανοποιούν κάποια χαρακτηριστικά. Αυτά εξαρτώνται από την ημερομηνία και συγκεκριμένα το UpdateDate πεδίο καθώς και το εάν έχουν ξανά διαβαστεί ή όχι δηλαδή είναι 1 (true) ή 0 (false) το IsParsed πεδίο. Σε κάθε εκτέλεση λοιπόν της συγκεκριμένης SP μεταφέρονται τα δεδομένα που έχουν ημερομηνία και ώρα μέχρι πριν από 10 λεπτά από τον χρόνο εκτέλεσης και που δεν έχουν ξανά διαβαστεί.

6.4.3 SM_PolylinesIntoPolygons

Η συγκεκριμένη Store Procedure αποτελεί και αυτή μέρος ενός βήματος του Agent Job ‘GISGettingData’ αλλά και του Job που εκτελεί την αρχικοποίηση των δεδομένων δηλαδή του ‘GISInitializePoolData’.

Ο ρόλος της Store Procedure αυτής είναι να εντοπίσει εάν υπάρχουν πλοία τα οποία έχουν περάσει μέσα στα όρια οποιασδήποτε ζώνης ενδιαφέροντος. Αν βρεθεί κάποιο τέτοιο πλοίο τότε μια συγκεκριμένη εγγραφή εισάγεται στον πίνακα tblNotifications ώστε στη συνέχεια να μπορέσει να ειδοποιηθεί ο χρήστης.

Πιο συγκεκριμένα ο έλεγχος για το αν ένα πλοίο βρίσκεται εντός μια περιοχής ενδιαφέροντος γίνεται μέσω της παρακάτω εντολής.

```
ZOI.Zones.STIntersects(LSDLS.GeogLoc ) = 1
```

Η εντολή αυτή γυρίζει 1 στην περίπτωση που μια ζώνη ενδιαφέροντος (ZOI.Zones) τέμνεται με την διαδρομή που ακολουθεί το πλοίο (LSDLS.GeogLoc). Επομένως τα αποτελέσματα που γυρίζει το query αφορούν μόνο πλοία που έχουν περάσει τα όρια μιας περιοχής ενδιαφέροντος πράγμα που σημαίνει ότι κάθε εγγραφή αντιστοιχεί σε κάθε πλοίο που έχει περάσει στο εσωτερικό κάποιας ζώνης. Το συγκεκριμένο query είναι το παρακάτω.

```
SELECT count (*)
FROM tblLiveShipDataLineString LSDLS
INNER JOIN tblZonesOfInterest ZOI on ZOI.Zones.STIntersects(LSDLS.GeogLoc ) = 1
INNER JOIN tblShips S on S.VesselID = LSDLS.VesselID
WHERE NOT EXISTS (Select 1 from tblNotifications where VesselID = LSDLS.VesselID AND
PolygonID = ZOI.ID)
AND ZOI.Zones.STIntersects(LSDLS.GeogLoc ) = 1 AND ZOI.IsDeleted = 0
```

Αν λοιπόν όντως μετά τον έλεγχο εντοπιστεί κάποιο πλοίο να βρίσκεται εντός μια περιοχής ενδιαφέροντος στη συνέχεια βρίσκουμε πιο είναι το ακριβές σημείο της ζώνης από το οποίο το πλοίο πέρασε μέσα. Αυτό επιτυγχάνεται μέσω της χρήσης ενός κέρσορα (SQL Cursor). Για κάθε εγγραφή που λαμβάνεται μέσω του κέρσορα εκτελείτε η παρακάτω εντολή.

```

SET @polygone = (select Zones from tblZonesOfInterest where id = @polygonID)
SET @polyline = (select GeogLoc from tblLiveShipDataLineString where VesselID =
@vesselId and LegID = @legID)
SET @result = (SELECT @polygone.STIntersection(@polyline).STStartPoint() AS Shape)
SET @points = @result.STAsText()

```

Το αποτέλεσμα της εντολής είναι της μορφής Point (Lon Lat) δηλαδή

```
POINT (24.226770431429426 37.195197675020694)
```

Συγκεκριμένα αυτό επιτυγχάνεται με την εντολή STStartPoint μέσω της οποίας λαμβάνεται το πρώτο σημείο της τομής του πολύγону (ζώνη ενδιαφέροντος) και της γραμμής (διαδρομή πλοίου).

Στη συνέχεια μέσω T-SQL κώδικα απομονώνουμε ξεχωριστά τα Lon και Lat σημεία και ενημερώνουμε τον πίνακα tblNotifications ώστε να ενημερωθεί ο χρήστης για το σημείο αυτό που υπολογίσαμε.

6.4.4 SM_GetVesselsLast2CurrentPos

Η συγκεκριμένη store procedure ανακτά τα δεδομένα σχετικά με τις θέσεις των πλοίων. Δηλαδή στη περίπτωση που ο χρήστης επιλέξει να δει μόνο τα τελευταία ληφθέντα σημεία των πλοίων. Στην SP λαμβάνονται υπόψη ως προαιρετικές παράμετροι (Optional Parameters) οι τυχόν επιλογές φίλτρων του χρήστη μέσω της εφαρμογής. Αυτές οι επιλογές όπως παρουσιάζεται και στη συνέχεια στο κεφάλαιο που αναλύεται η εφαρμογή αφορούν συγκεκριμένη λίστα ονομάτων πλοίων, λιμάνια προορισμών, τύπο πλοίου, σημαία πλοίου καθώς και φίλτρο διαστήματος συγκεκριμένων ημερομηνιών.

Στη προκειμένη λειτουργία της εφαρμογής (Ship Latest Received Positions) δεν μας ενδιαφέρει μόνο η τελευταία θέση του πλοίου που ελήφθει αλλά και η προηγούμενη. Αυτό συμβαίνει ώστε να μπορούμε να υπολογίζουμε στοιχεία του πλοίου όπως την κατεύθυνση (υπολογίζοντας της διαφορές των σημείων) και την ταχύτητα του (υπολογίζοντας διάστημα και χρόνο που έκανε να το καλύψει). Για τον λόγο αυτό χρησιμοποιούμε Analytic Functions του SQL Server όπως φαίνεται παρακάτω.

```

select VesselID, LegID , UpdateTime , Lat, Lon,
LAG(UpdateTime , 1,0) OVER (PARTITION BY VesselID ORDER BY UpdateTime ASC) AS
PreUpdateTime,
LEAD (Lat , 1, 0) OVER (PARTITION BY VesselID ORDER BY UpdateTime DESC) AS
PrePositionLat,
LEAD (Lon , 1, 0) OVER (PARTITION BY VesselID ORDER BY UpdateTime DESC) AS
PrePositionLon,
ROW_NUMBER () OVER (Partition by VesselID Order by UpdateTime DESC ) as ROWNUM
from tblLiveShipData

```

Το παραπάνω είναι μέρος της συγκεκριμένης Store Procedure και χρησιμοποιώντας το Lag και το Lead μπορούμε να ανακτήσουμε, εκτός από τις εγγραφές για τις οποίες κάναμε αναζήτηση, και τις προηγούμενες αυτών, αναφορικά με τα Lat και Lon σημεία (LEAD – Lat για το PrePositionLat (το Lat της προηγούμενης εγγραφής) και αντίστοιχα για το Lon) αλλά και το χρόνο (Lag of UpdateTime)

6.4.5 SM_GetVesselsRoutes

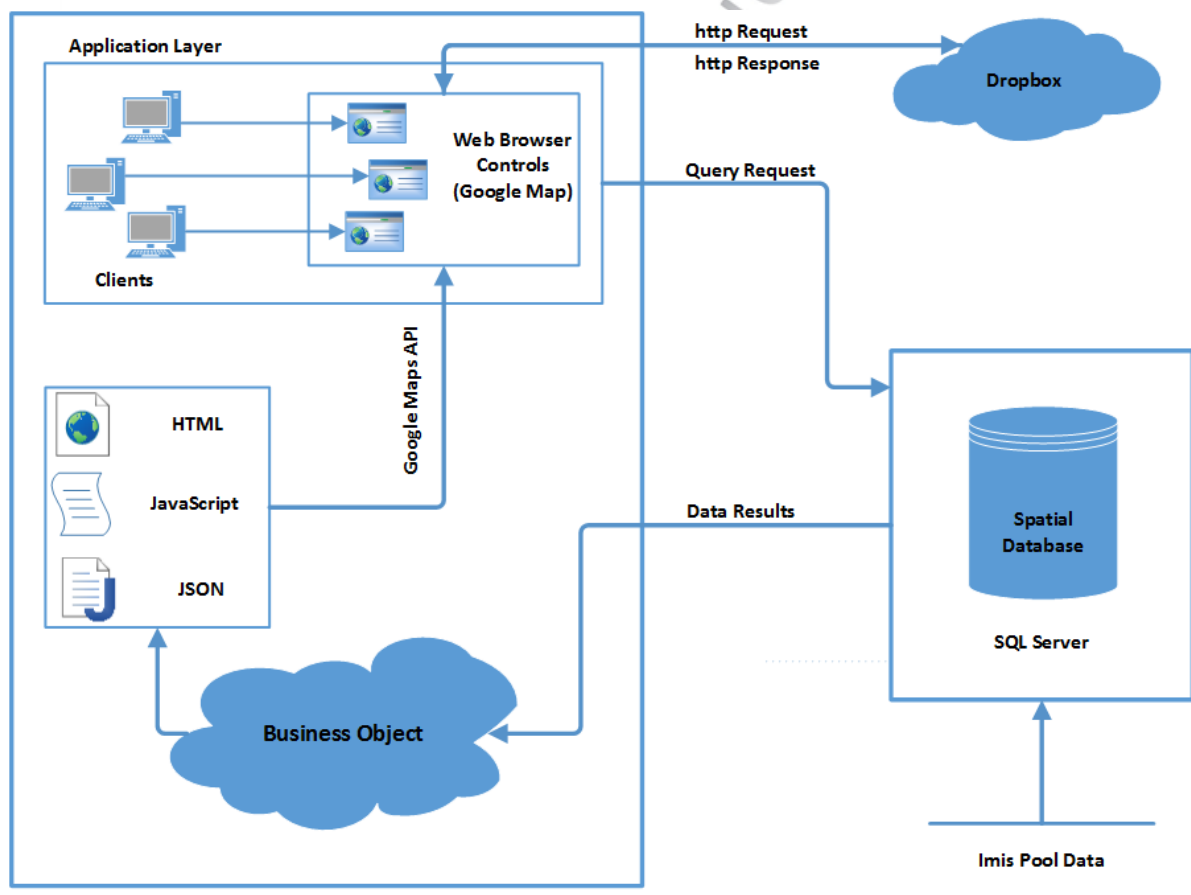
Η συγκεκριμένη Store Procedure ανακτά τα δεδομένα σχετικά με της συνολικές διαδρομές των πλοίων. Δηλαδή εκτελείται όταν ο χρήστης επιλέξει να δει διαδρομές των πλοίων και όχι συγκεκριμένες θέσεις τους. Και η συγκεκριμένη Store Procedure λαμβάνει υπόψη προαιρετικές παραμέτρους (Optional Parameters) που αντιστοιχούν στα φίλτρα επιλογής του χρήστη. Επιπρόσθετα όμως υπάρχει και μια επιπλέον παράμετρος που

σχετίζεται με το Leg (διαδρομή του πλοίου). Δηλαδή με το συγκεκριμένο Leg το οποίο επιλέγει ο χρήστης να δει ανά πλοίο.

7. Αρχιτεκτονική του Συστήματος

Στην ενότητα αυτή περιγράφεται η αρχιτεκτονική της εφαρμογής Vessel Tracking System (Σύστημα Παρακολούθησης Θέσης Πλοίων) που αποτελεί και το αντικείμενο της παρούσας εργασίας.

Η εφαρμογή στηρίζεται σε μια αρχιτεκτονική windows desktop client - server με Google maps interoperability χρησιμοποιώντας JSON & JavaScript όπως φαίνεται και στην παρακάτω εικόνα. Αυτή η αρχιτεκτονική μας επιτρέπει τη δυνατότητα εμφάνισης στον τελικό χρήστη του χάρτη μέσω του οποίου γίνεται η απεικόνιση των δεδομένων που στην συγκεκριμένη περίπτωση είναι τα πλοία μιας ναυτιλιακής εταιρίας.



Εικόνα 7-1: Διάγραμμα αρχιτεκτονικής της εφαρμογής VTS

Αρχικά το κομμάτι του Application Layer το οποίο αποτελείται από πολλαπλούς clients ανάλογα με τους χρήστες του συστήματος. Στον κάθε client υπάρχει και το αντίστοιχο Web Browser Control στο οποίο υλοποιείται και εμφανίζεται ο χάρτης με τον οποίο ο χρήστης μπορεί να αλληλεπιδρά.

Αυτό έχει δημιουργηθεί με τη χρήση του Visual Studio και συγκεκριμένα μέσω VB.NET η οποία είναι και η βασική γλώσσα προγραμματισμού υλοποίησης της εφαρμογής. Το Web Browser Control χρησιμοποιήθηκε για να μπορέσουμε να υλοποιήσουμε σε μια desktop εφαρμογή όπως η παρούσα, την εμφάνιση του χάρτη. Μέσω του control αυτού γίνεται η οποιαδήποτε εμφάνιση στο επίπεδο του χάρτη. Για την παρούσα εφαρμογή δηλαδή θα μπορούσαμε να πούμε ότι το control αυτό είναι ουσιαστικά ότι είναι ο browser σε μια διαδικτυακή εφαρμογή (web application).

Αντίστοιχα για να αποδώσουμε οτιδήποτε χρειαζόμαστε στο control αυτό, δηλαδή στο χάρτη, χρησιμοποιούμε γλώσσα προγραμματισμού JavaScript. Μέσω JavaScript δηλαδή κάνουμε αρχικοποίηση στον χάρτη και εμφανίζουμε οποιοδήποτε αντικείμενο (marker) υπάρχει σε οποιαδήποτε στιγμή πάνω στο χάρτη.

Πιο συγκεκριμένα, η εφαρμογή μέσω του χρήστη και μέσω συγκεκριμένων επιλογών που έχει τη δυνατότητα να κάνει αναφορικά με το τι θέλει να εμφανίσει στο χάρτη, εκτελεί κάποια ερωτήματα στη βάση δεδομένων του συστήματος. Στη συνέχεια τα ερωτήματα αυτά απαντώνται και τα δεδομένα τα οποία προκύπτουν σάζονται στις αντίστοιχες κλάσεις της εφαρμογής. Στη συνέχεια θα πρέπει τα δεδομένα αυτά να περάσουν στο επίπεδο του Web Browser Control για να εμφανιστούν στον χάρτη.

Αυτό επιτυγχάνεται μέσω JSON arrays. Δηλαδή μεταφορτώνονται τα δεδομένα από τις κλάσεις σε JSON αντικείμενα (JSON object) τα οποία με τη σειρά τους μέσω κατάλληλων εντελών μεταφορτώνονται στα .htm αρχεία το περιεχόμενο των οποίων είναι υπεύθυνο για το τι εμφανίζεται στον χάρτη.

Εκεί μέσω JavaScript τα δεδομένα τελικά φορτώνονται σε κατάλληλα αντικείμενα (JS objects) τα οποία χειριζόμαστε κατάλληλα ώστε να γίνει η απεικόνιση των δεδομένων της επιλογής του χρήστη, στον χάρτη της εφαρμογής του. Στο παραπάνω βασιζόμαστε και στο API της Google (Google Maps JavaScript API v3) το οποίο παρέχει τη δυνατότητα στους προγραμματιστές μέσα από κατάλληλες εντολές και συναρτήσεις να χειριστούν τους Google Maps χάρτες ώστε να εμφανίζουν οτιδήποτε χρειάζεται η κάθε εφαρμογή πάνω στον χάρτη.

Επίσης το κάθε web browser control μέσω εντολών JavaScript μπορεί να καλέσει μέσα από την εφαρμογή Dropbox τα κατάλληλα εικονίδια τα οποία απεικονίζουν τα διάφορα σημεία που βρίσκονται επάνω στον χάρτη (markers), ώστε να παρέχουν στον χρήστη ένα πιο αντιπροσωπευτικό και κατανοητό user interface ανάλογα την εφαρμογή.

8. Υλοποίηση

Στο παρόν κεφάλαιο θα γίνει μια αναφορά στα κυριότερα κομμάτια της υλοποίησης της εφαρμογής ενώ θα παρουσιαστούν και τα βασικά κομμάτια κώδικα μέσω των οποίων επιτυγχάνεται η αντίστοιχη υλοποίηση. Κάποια από αυτά αναφέρονται γιατί είναι κομβικά για την εφαρμογή και κάποια άλλα λόγω του ότι είχαν κάποια παραπάνω δυσκολία στην υλοποίηση τους.

Η υλοποίηση της εφαρμογής όπως έχει ήδη αναφερθεί έγινε με χρήση την πλατφόρμας Visual Studio 2012 και συγκεκριμένα της γλώσσας προγραμματισμού VB.NET η οποία χρησιμοποιεί Windows Forms για τις διάφορες υλοποιήσεις. Επίσης χρησιμοποιήθηκε και γλώσσα προγραμματισμού JavaScript για τα .htm αρχεία μέσω των οποίων απεικονίζονται τα διάφορα στοιχεία στον χάρτη. Τέλος χρησιμοποιήθηκε σε μεγάλο βαθμό και η T-SQL για την δημιουργία των Store Procedures της εφαρμογής όπως είδαμε σε προηγούμενο κεφάλαιο.

8.1 Εμφάνιση Google Maps χάρτη

Λόγω του ότι η εφαρμογή μας είναι desktop εφαρμογή και χρησιμοποιεί windows forms έπρεπε να βρεθεί κάποια λύση στο να αντικατασταθεί ο κλασικός browser των διαδικτυακών εφαρμογών με κάτι άλλο έτσι ώστε να μπορέσουμε να εμφανίσουμε τον απαραίτητο χάρτη. Για τον λόγο αυτό χρησιμοποιήθηκε μια κλάση

που ονομάζεται WebBrowser Class και την οποία παρέχει το .NET Framework της Microsoft ώστε να δίνει τη δυνατότητα στον χρήστη να μπορεί να πλοηγηθεί σε διαδικτυακές σελίδες μέσω των Windows forms. Με τη χρήση του παρακάτω κώδικα λοιπόν γίνεται αρχικοποίηση της κλάσης GoogleControl την οποία έχουμε δημιουργήσει ώστε μέσω αυτής να γίνεται και η αρχικοποίηση της έτοιμης κλάσης WebBrowser. Στη συνέχεια προστίθεται στο πάνελ που έχει οριστεί (pnlMap) για να εμφανιστεί ο χάρτης. Η κλάση GoogleControl έχει και άλλες υπό-ρουτίνες τις οποίες θα δούμε στη συνέχεια.

```
GoogleControl1 = New GoogleControl
GoogleControl1.Dock = DockStyle.Fill
pnlMap.Controls.Add(GoogleControl1)
```

```
Sub New ()
```

```
    MyBase.New ()
    WebBrowser1 = New WebBrowser
    WebBrowser1.Dock = DockStyle.Fill
    WebBrowser1.AllowWebBrowserDrop = False
    WebBrowser1.IsWebBrowserContextMenuEnabled = False
    WebBrowser1.WebBrowserShortcutsEnabled = False
    WebBrowser1.ObjectForScripting = Me
    WebBrowser1.ScriptErrorsSuppressed = False
    AddHandler WebBrowser1.DocumentCompleted, AddressOf WebBrowser1_DocumentCompleted
    Me.Controls.Add(WebBrowser1)
    Me.Controls.Add(StatusStrip1)
```

```
End Sub
```

Έτσι χρησιμοποιώντας τα παραπάνω είμαστε σε θέση να εμφανίσουμε τον Google Maps χάρτη στο σημείο που επιθυμούμε μίας desktop εφαρμογής.

8.2 Οι κλάσεις του προγράμματος

Στο επόμενο βήμα δημιουργήθηκαν οι κλάσεις της εφαρμογής που αντιπροσωπεύουν της οντότητες που παρουσιάστηκαν στο διάγραμμα της βάσης δεδομένων και χρησιμοποιήθηκαν ώστε να γίνεται άμεσα η ανάκτηση της ζητούμενης πληροφορίας από τη βάση και να είναι άμεσα διαθέσιμη οποτεδήποτε ζητηθεί.

clsVessel.vb

Είναι από τις βασικές κλάσεις του συστήματος αφού χρησιμοποιείται για την ανάκτηση πληροφορίας σχετικά με τα πλοία. Συγκεκριμένα καλείται όταν ο χρήστης επιλέξει να εμφανίσει στο χάρτη τις ακριβείς θέσεις των πλοίων και αποθηκεύει όλα τα δεδομένα που σχετίζονται με το κάθε πλοίο και τα χαρακτηριστικά, πολλά από τα οποία στη συνέχεια θα εμφανιστούν σαν πληροφορία στο info windows του κάθε πλοίου. Οι ιδιότητες της κλάσης αυτής ‘γεμίζουν’ (populated) μέσω της Store Procedure ‘SM_GetVesselsLast2CurrentPos’ η λειτουργία της οποίας έχει περιγραφεί σε προηγούμενο κεφάλαιο.

clsVesselRoute.vb

Είναι επίσης από τις βασικές κλάσεις του συστήματος. Περιλαμβάνει ιδιότητες σχετικές με τις διαδρομές των πλοίων, δηλαδή καλείται όταν ο χρήστης επιλέξει να εμφανίσει τις διαδρομές των πλοίων. Αρχικά γίνεται η αρχικοποίηση της κλάσης και στη συνέχεια αναλόγως τις επιλογές του χρήστη εκτελείται η Store Procedure ‘SM_GetVesselsRoutes’ μέσω τις οποίας ‘γεμίζουν’ (populated) οι ιδιότητες της κλάσης.

clsZonesOfInterest.vb

Η συγκεκριμένη κλάση περιέχει απλές ιδιότητες των περιοχών ενδιαφέροντος. Όταν ο χρήστης επιλέξει να εμφανίσει αυτές τις περιοχές η κλάση φορτώνεται και με τη χρήση της Store Procedure ‘SM_GetZonesOfInterestList’ ‘γεμίζουν’ (populated) οι ιδιότητες της για να χρησιμοποιηθούν στη πορεία της εφαρμογής και να εμφανιστούν οι περιοχές ενδιαφέροντος κατάλληλα.

clsNotifications.vb

Είναι η κλάση η οποία κρατάει την πληροφορία σχετικά με τις ειδοποιήσεις που εμφανίζονται στον χρήστη. Οι ιδιότητές της ‘γεμίζουν’ (populated) μέσω της Store Procedure ‘SM_GetNotifications’ και στις συνέχεια χρησιμοποιούνται για την εμφάνιση της αντίστοιχης πληροφορίας στον χρήστη.

clsPorts.vb

Η κλάση αυτή περιέχει όλες τις ιδιότητες σχετικά με τα λιμάνια. ‘Γεμίζουν’ (populated) με τη χρήση και εκτέλεση της Store Procedure ‘SM_GetPortList’ και χρησιμοποιείται για την εμφάνιση της αντίστοιχης πληροφορίας.

clsUser.vb

Περιέχει τις ιδιότητες που σχετίζονται με τους χρήστες του συστήματος. Ουσιαστικά στη παρούσα εφαρμογή χρησιμοποιείται για να φορτώσει όλα τα σχετικά με τους χρήστες δεδομένα στο αντίστοιχο grid της εφαρμογής.

8.3 Modes λειτουργίας

Έχοντας λοιπόν δημιουργήσει τον χάρτη μας ας δούμε πως θα αναπαραστήσουμε εκεί τα χωρικά δεδομένα που μας ενδιαφέρουν. Τα δεδομένα αυτά για την συγκεκριμένη εφαρμογή είναι τα σημεία των λιμανιών, οι ζώνες ενδιαφέροντος και τα στίγματα των πλοίων επάνω στο χάρτη. Τα σημεία των πλοίων χωρίζονται σε δύο κατηγορίες.

- Τα σημείο (points) τα οποία αναπαριστούν το τελευταίο σημείο που έχει ληφθεί για το κάθε πλοίο ή οποιοδήποτε άλλο σημείο μιας ζητούμενης χρονικής στιγμής
- Τις διαδρομές (LineStrings) τις οποίες έχει διανύσει το κάθε πλοίο από την αρχικοποίηση των δεδομένων μας μέχρι το τρέχον χρονικό σημείο ή κάποιο άλλο χρονικό σημείο που μας ενδιαφέρει.

Στην εφαρμογή μας θεωρούμε τα τέσσερα παραπάνω χαρακτηριστικά ως αυτά σύμφωνα με τα οποία θα πρέπει να γίνονται οι κατάλληλες διεργασίες ώστε να είναι σε θέση ο χρήστης να εμφανίσει χρησιμοποιώντας την εφαρμογή τα αποτελέσματα που τον ενδιαφέρουν. Τα παραπάνω έχει ο χρήστης την δυνατότητα να τα ενεργοποιήσει στον χάρτη.

Εκτός από τον χάρτη όμως η εφαρμογή έχει και αρχική λειτουργία αλλά και δυνατότητα εξαγωγής ειδοποιήσεων σε περίπτωση πλήρωσης κάποιων κριτηρίων. Λόγο των παραπάνω ορίσαμε στην εφαρμογή μας 4 καταστάσεις (modes). Αυτές είναι οι παρακάτω:

- Initial: Είναι η κατάσταση που βρίσκεται η εφαρμογή μόλις ο χρήστης κάνει login. Εμφανίζεται ο προεπιλεγμένος (default) χάρτης χωρίς καμία άλλη πληροφορία επάνω του.
- VesselPositions: Είναι η κατάσταση στην οποία ο χρήστης έχει επιλέξει να εμφανίσει συγκεκριμένες θέσεις/σημεία πλοίων και όχι τροχιές αυτών
- VesselTrajectories: Είναι η κατάσταση της εφαρμογής στην οποία ο χρήστης έχει επιλέξει να εμφανίσει τις τροχιές των πλοίων
- NotificationMode: Είναι η κατάσταση της εφαρμογής κατά την οποία γίνεται ο έλεγχος για το εάν υπάρχουν ειδοποιήσεις για τον χρήστη ώστε να τις εμφανίσει.

Έτσι χρησιμοποιώντας το παρακάτω κομμάτι του κώδικα ανάλογα με την κατάσταση που βρισκόμαστε, σύμφωνα με τις επιλογές του χρήστη, φορτώνεται η κλάση GoogleControl και με τη σειρά τους τα αρχεία .htm τα οποία ορίζουν το περιεχόμενο του χάρτη φορτώνονται στο web browser και τέλος εμφανίζεται ο χάρτης.

```

Private Sub GoogleControl_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load

    Try
        If frmVTSMMain.blnInitial Then WebBrowser1.DocumentText =
            My.Computer.FileSystem.ReadAllText("GMap_InitialMap.htm")
            _MapViewType = MapViewType.Initial
        ElseIf frmVTSMMain.blnShips Then WebBrowser1.DocumentText =
            My.Computer.FileSystem.ReadAllText("GMap_VesselsCurrentPos.htm")
            _MapViewType = MapViewType.VesselPositions
        ElseIf frmVTSMMain.blnShipTrajectories Then WebBrowser1.DocumentText =
            My.Computer.FileSystem.ReadAllText("GMap_VesselRoutePolyline_NEW.htm")
            _MapViewType = MapViewType.VesselTrajectories
        ElseIf frmVTSMMain.blnNotificationMode Then WebBrowser1.DocumentText =
            My.Computer.FileSystem.ReadAllText("GMap_NotificationMode.htm")
            _MapViewType = MapViewType.NotificationMode
        Else
            WebBrowser1.DocumentText =
            My.Computer.FileSystem.ReadAllText("GMap_InitialMap.htm")
            _MapViewType = MapViewType.Initial
        End If

    Catch ex As Exception
        Throw New Exception(ex.Message)
    End Try

End Sub

```

8.4 Τα .htm αρχεία και το API της Google

Τα αρχεία .htm είναι αυτά σύμφωνα με τα οποία ορίζεται κάθε φορά το περιεχόμενο του χάρτη. Το περιεχόμενο των αρχείων είναι σε html μορφή και η γλώσσα προγραμματισμού που χρησιμοποιείτε είναι η JavaScript. Παρακάτω βλέπουμε το κομμάτι αρχικοποίησης ενός τέτοιου αρχείου στο οποίο εισάγονται η βιβλιοθήκη για το API του GoogleMaps καθώς και βιβλιοθήκες για τη χρήση του JSON.

```

<!DOCTYPE html>
<html>
  <head>
    <!-- The label <meta> specify that the map should show in full screen
    and the user can't modify the size. -->
    <meta name="viewport" content="initial-scale=1.0, user-
      scalable=no" />
    <style type="text/css">
      html { height: 100% }
      body { height: 100%; margin: 0; padding: 0 }
      #map_canvas { height: 100% }
    </style>
    <script type="text/javascript"
      src="https://maps.googleapis.com/maps/api/js?libraries=weather&sensor
      =true"> </script>
    <script type="text/javascript" src="C:\Current
      Projects\GoogleMaps_vb12_v4\References\Json\JSON-js-master\json2.js"
      ></script>
    <script type="text/javascript" src="C:\Current
      Projects\GoogleMaps_vb12_v4\References\Json\JSON-js-
      master\json_parse.js" ></script>
    <script type="text/javascript" >

```

Μέσω του API της Google χρησιμοποιώντας συγκεκριμένες εντολές μπορούμε να εισάγουμε τον χάρτη ο οποίος φορτώνεται όταν το συγκεκριμένο .htm αρχείο καλείται από τον κώδικα της εφαρμογής.

```
var myOptions = {
  center: new google.maps.LatLng(37.753344, 24.433594),
  zoom: 8,
  mapTypeControl: true,
  mapTypeControlOptions: {style:
    google.maps.MapTypeControlStyle.DROPDOWN_MENU},
  navigationControl: true,
  navigationControlOptions: {style:
    google.maps.NavigationControlStyle.SMALL},
  //mapTypeId: google.maps.MapTypeId.ROADMAP
  mapTypeId: google.maps.MapTypeId.TERRAIN
}

map = new google.maps.Map(document.getElementById("map_canvas"), myOptions);
```

8.5 Περιεχόμενο των .htm αρχείων

Όπως έχει αναφερθεί και σε προηγούμενα κεφάλαια τα .htm αρχεία είναι υπεύθυνα για το περιεχόμενο στο οποίο θα εμφανίζεται στον Google Map χάρτη της εφαρμογής. Όπως επίσης αναφέρθηκε παραπάνω, η παρούσα εφαρμογή ουσιαστικά χωρίζεται σε 4 modes (καταστάσεις). Έτσι για το κάθε mode έχει υλοποιηθεί διαφορετικό .htm αρχείο ώστε να γίνεται το parse του αρχείου γρηγορότερα και ο χρόνος απόκρισης και δημιουργίας του χάρτη να είναι χαμηλότερος. Επίσης έχοντας περισσότερα από ένα αρχεία είναι ευκολότερο για τον προγραμματιστή να διαβάσει καθώς και να επεξεργαστεί το κάθε αρχείο βρίσκοντας ευκολότερα το σημείο το οποίο ψάχνει.

Παρακάτω θα δούμε τα βασικά κομμάτια κώδικα των αρχείων αυτών περιγράφοντας την λειτουργία τους.

GMap InitialMap.htm

Αντιστοιχεί στην κατάσταση Initial στην οποία εμφανίζεται ο προεπιλεγμένος (default) χάρτης χωρίς καμία άλλη πληροφορία επάνω στον χάρτη. Το κομμάτι του κώδικα εμφάνισης του χάρτη είναι το ίδιο για όλα τα αρχεία ανεξαρτήτου mode, αφού πρώτα εμφανίζεται ο βασικός χάρτης και πάνω σε αυτόν προστίθενται κάθε φορά τα απαραίτητα ανάλογα με τις επιλογές του χρήστη.

Στη συγκεκριμένη περίπτωση μπορεί να μιλάμε για το αρχικό mode αλλά ακόμα και σε αυτό ο χρήστης έχει τη δυνατότητα να εμφανίσει τα λιμάνια καθώς και όλες τις περιοχές ενδιαφέροντος. Οι αντίστοιχες επιλογές είναι όμοιες σε όλες τις καταστάσεις λειτουργίας.

Η ροή των δεδομένων μέχρι να φτάσουμε στο σημείο αυτό περιγράφεται στην επόμενη παράγραφο. Εν συντομία τα δεδομένα των περιοχών αυτών έχουν ληφθεί από την βάση δεδομένων, έχουν προστεθεί στην αντίστοιχη VB.NET κλάση, στη συνέχεια μετασχηματίστηκαν σε JSON Objects ώστε να περάσουν στα htm αρχεία, οπου τελικά μεταφορτώθηκαν σε JavaScript objects και τελικά είναι σε θέση να χρησιμοποιηθούν.

Παρακάτω εμφανίζεται ο κώδικας σύμφωνα με τον οποίο εμφανίζονται τα σημεία ενδιαφέροντος επάνω στον χάρτη.

Αρχικά ελέγχεται αν υπάρχουν στοιχεία σχετικά με τις περιοχές ενδιαφέροντος. Αν υπάρχουν, μετασχηματίζεται το JSON array σε JavaScript Object και στη συνέχεια ξεκινάει ένας βρόγχος (loop) στο οποίο για τη κάθε περιοχή ενδιαφέροντος διαβάζονται τα Lat και Lon του κάθε σημείου τους ώστε τελικά να σχηματιστούν από τα σημεία αυτά οι περιοχές στο χάρτη. Όλα τα παραπάνω σημεία για την κάθε περιοχή αποθηκεύονται στη μεταβλητή Border μέσω της οποίας περνάνε και εμφανίζονται στον χάρτη.

```

if (strZoneOfInterestList !== 'empty') {
  //Populates Json with Zone of Interest in a JavaScript Object
  var objJSONZoneOfInterest= JSON.parse(strZoneOfInterestList)
  var initialID = 1
  var latLng = [];
  for (var _i=0; _i <= objJSONZoneOfInterest.length -1 ; _i++){
    var latLng= [];
    var Border = []
    while (initialID == objJSONZoneOfInterest[_i].IDPiracyZones)
    {
      latLng.push(new google.maps.LatLng(objJSONZoneOfInterest[_i].Lon ,
        objJSONZoneOfInterest[_i].Lat));
      _i++;
      if (_i > objJSONZoneOfInterest.length - 1) {
        break;
      }
    }
    var Border = new google.maps.Polygon({
      paths: latLng,
      strokeColor: "#FF0000",
      strokeOpacity: 0.8,
      strokeWeight: 2,
      fillColor: "#FF0000",
      fillOpacity: 0.35
    });

    Border.setMap(map);
    initialID++;
  }
}

```

Ίδιας λογικής είναι και κώδικας μέσω του οποίου φορτώνονται τα σημεία και οι markers τους για τα λιμάνια. Η διαφορά είναι ότι στη προκειμένη περίπτωση ορίζεται και το εικονίδιο του κάθε σημείου με τον αντίστοιχο κώδικα όπως αναλύεται σε παρακάτω παράγραφο του κεφαλαίου .

```

if (strPortList !== 'empty') {

  var myJSONPortListObject = strPortList
  var objJSONPortList= JSON.parse(strPortList)
  var infoWindow = new google.maps.InfoWindow;

  for (i=0;i<objJSONPortList.length;i++){
    //Prevent to show the Unknown port point
    if (objJSONPortList[i].PortLat !== '0' && objJSONPortList[i].PortLon !== '0') {
      iconBase = 'https://dl.dropboxusercontent.com/u/10540326/';
      image = 'harbor%20(2).png';

      var location = new
      google.maps.LatLng(objJSONPortList[i].PortLat,objJSONPortList[i].PortLon);

      var marker = new google.maps.Marker({
        position: location,
        map: map,
        icon: iconBase + image,
        html: msg
      });
    }
  }
}

```


GMap_VesselsCurrentPos.htm

Το συγκεκριμένο .htm αρχείο καλείται όταν ο χρήστης επιλέγει να δει τα σημεία τα οποία βρίσκονται τα πλοία σε κάποια δεδομένη χρονική στιγμή ή την τελευταία εγγραφή της θέσης τους για μια συγκεκριμένη χρονική περίοδο. Στο συγκεκριμένο αρχείο υπάρχουν τα κομμάτια του κώδικα για τα λιμάνια και τις περιοχές ενδιαφέροντος καθώς επίσης και ένας βρόγχος μέσω του οποίου διαβάζονται τα δεδομένα που απαιτούνται για να εμφανιστούν ως πληροφορία για κάθε πλοίο ενώ γίνονται και διάφοροι υπολογισμοί όπου απαιτούνται για στοιχεία που δεν υπάρχει πληροφορία αλλά πρέπει να εμφανιστεί, όπως η ταχύτητα του πλοίου ή η εκτίμηση του χρόνου άφιξης στον προορισμό (ETA - Estimation Time Of arrival).

Ο βρόγχος για την εμφάνιση του στίγματος των πλοίων έχει την αντίστοιχη μορφή με αυτούς που εμφανίστηκαν παραπάνω. Με τη διαφορά ότι μέσα σε αυτόν πραγματοποιούνται διάφοροι υπολογισμοί μέρος των οποίων βλέπουμε παρακάτω.

Το παρακάτω κομμάτι κώδικα υπολογίζει την ευκλείδεια απόσταση σε μίλια που πρέπει να καλύψει το πλοίο από την τρέχουσα θέση του για να φτάσει στον προορισμό του. Χρησιμοποιώντας την θέση του λιμανιού προορισμού και την θέση που τώρα βρίσκεται το πλοίο (μέσω των αντίστοιχων Lat Lon σημείων) υπολογίζει την απόσταση σε μέτρα και στη συνέχεια σε Ναυτικά μίλια.

```
var nextportpos = new
    google.maps.LatLng(objJSON[i].DestinationLat,objJSON[i].DestinationLon);
var shipcurpos = new google.maps.LatLng(objJSON[i].VesselLat,objJSON[i].VesselLon);
var metretoport =
    google.maps.geometry.spherical.computeDistanceBetween(shipcurpos,nextportpos);
var milestoport = metretoport * 0.000621371192; //converts to nautical miles from
    meters
milestoport = milestoport.toFixed(2); //rounding with 2 digits after comma
```

Το παρακάτω κομμάτι κώδικα υπολογίζει την ταχύτητα με την οποία κινείται το πλοίο αφού αυτή η πληροφορία δεν είναι διαθέσιμη στα αρχικά δεδομένα. Παρατηρούμε στον κώδικα ότι αρχικά υπολογίζεται η απόσταση μεταξύ της προηγούμενης ληφθείσας θέσης και της τελευταίας ληφθείσας θέσης και αποθηκεύεται στη μεταβλητή MyGetDistance. Στη συνέχεια υπολογίζεται ο χρόνος τον οποίο χρειάστηκε το πλοίο για να διανύσει την απόσταση αυτή (MyGetDateTimeDiff) και στη συνέχεια διαιρώντας τα δύο παραπάνω και μετατρέποντας το αποτέλεσμα σε Knots από χιλιόμετρα, έχουμε το αποτέλεσμα της ταχύτητας του πλοίου για τη συγκεκριμένη διαδρομή.

```
var MyGetDistance =
    getDistanceFromLatLonInKm(objJSON[i].PreVesselLat,objJSON[i].PreVesselLon,objJSON[
    i].VesselLat,objJSON[i].VesselLon)
var MyGetDateTimeDiff =
    diff((JSONDateWithTime(objJSON[i].PreLastUpdateDate)),(JSONDateWithTime(objJSON[i]
    .LastUpdateDate)))
var MyGetSpeed = Math.round(((MyGetDistance/MyGetDateTimeDiff) * 0.539956803) * 100)
    / 100 //(Kmh/Hour) and convert to knots
```

GMap_VesselRoutePolyline_NEW.htm

Το συγκεκριμένο .htm αρχείο καλείται όταν ο χρήστης επιλέγει να δει τις διαδρομές των πλοίων δηλαδή πολλαπλά σημεία από τα οποία πέρασε το πλοίο στη διάρκεια μιας επιλεγμένης χρονικής περιόδου. Σε γενικές γραμμές η λογική του κώδικα είναι παρόμοια με προηγουμένως. Ένα σημείο που διαφέρει είναι ότι στη προκειμένη περίπτωση εκτός του ότι πρέπει να επιλέξουμε συγκεκριμένου χρώματος marker θα πρέπει και οι γραμμές οι οποίες ενώνουν τους markers αυτούς να έχουν το αντίστοιχο χρώμα.

```

//Get color of each polyline
if (objJSON.length >= 1) {
    var vesselttype = objJSON[1].VesselType
    switch (vesselttype)
    {
        case 'Cargo Ship': linecolor = '#0000FF' ; //blue
            break;
        case 'Tanker': linecolor = '#00FFFF' ; //cyan
            break;
        case 'LNG': linecolor = '#98FF98' ; //lightgreen
            break;
        case 'Passenger Vessel': linecolor = '#FF0000' ; //red
            break;
        case 'Fishing': linecolor = '#808080' ; //lightgrey
            break;
        default: linecolor = 'black' ;
    }
}
else if (objJSON.length < 1) {
    linecolor = 'black' ;
}

//Set polyline
var polylineoptns = {
    strokeColor: linecolor,
    strokeOpacity: 0.8,
    strokeWeight: 0.8,
    map: map
};

polyline = new google.maps.Polyline(polylineoptns);

```

Επίσης η ένωση των markers μέσω των παραπάνω γραμμών επιτυγχάνεται μέσω της παρακάτω εντολής η οποία αποθηκεύει το κάθε σημείο του κάθε πλοίο σε ένα array (z) και στη συνέχεια το εμφανίζει στο χάρτη.

```

polyline = new google.maps.Polyline(polylineoptns);
var z = 0;
var path = [];
path[z] = polyline.getPath();
.
.
.
path[z].push(marker.getPosition());

```

GMap_NotificationMode.htm

Το συγκεκριμένο αρχείο είναι υπεύθυνο για την εμφάνιση του σημείου τομής μεταξύ της διαδρομής ενός πλοίο και των συνόρων μιας περιοχής ενδιαφέροντος. Παρακάτω εμφανίζεται ο κώδικας σύμφωνα με τον οποίο ορίζεται ο marker ο οποίος ορίζει το ακριβές σημείο τομής των δύο, λαμβάνονται υπόψη το Lat και το Lon του σημείου το οποίο έχει βρεθεί μέσω συγκεκριμένης Store Procedure του SQL Server σε προηγούμενο στάδιο.

```

for (i=0;i<objJSON.length;i++){

    iconBase = 'https://dl.dropboxusercontent.com/u/135449836/GMapsApp/';
    ShipColorIcon = 'NotificationPoint'
    image = ShipColorIcon + '.png';

    var location = new
        google.maps.LatLng(objJSON[i].InitialIntersectPointLat,objJSON[i].InitialIntersectPointLon);

    var marker = new google.maps.Marker({
        position: location,
        map: map,
        icon: iconBase + image
    });
}

```

8.6 Το JSON ως interface μεταξύ VB.NET και JavaScript.

Στη συνέχεια το ζητούμενο ήταν να γίνει η συλλογή των δεδομένων σύμφωνα με τις επιλογές του χρήστη και να περαστούν στα αρχεία .htm ώστε να εμφανιστεί η κατάλληλη πληροφορία στον χάρτη. Στην προκειμένη περίπτωση μετά από την εκτέλεση των ερωτημάτων (queries) στη βάση δεδομένων τα δεδομένα συλλέγονται στις αντίστοιχες κλάσεις της εφαρμογής και στη συνέχεια πρέπει να περάσουν στα .htm αρχεία.

Για να επιτευχθεί αυτό χρησιμοποιήθηκε ένα είδος interface με την χρήση του JSON η λειτουργία του οποίου περιγράφηκε παραπάνω. Έτσι μέσω του JSON τα εκάστοτε δεδομένα είναι σε θέση να περάσουν από το βασικό πρόγραμμα της εφαρμογής δηλαδή την VB.NET στα htm αρχεία ώστε με χρήση JavaScript και του αντίστοιχου API της Google να πραγματοποιηθεί η εμφάνιση των απαιτούμενων στοιχείων στο χάρτη.

Συγκεκριμένα στο κομμάτι της VB.NET και μέσω της παρακάτω εντολής φορτώνονται όλα τα δεδομένα (εάν υπάρχουν) στο αρχείο htm

```

WebBrowser1.Document.InvokeScript("MyJsonFun", New Object() {strZonesOfInterestList, strPortList, strjson})

```

Οι 3 παράμετροι που εμφανίζονται στην συγκεκριμένη εντολή είναι αντικείμενα (objects) στα οποία έχει γίνει αρχικοποίηση σε μορφή JSON ώστε να μπορέσουν να αποθηκεύσουν τα δεδομένα από τις αντίστοιχες κλάσεις.

- Το strZonesOfInterestList object περιέχει τα δεδομένα σχετικά με τις ζώνες ενδιαφέροντος
- Το strPortList object περιέχει δεδομένα σχετικά με τα λιμάνια προορισμού και απόπλου.
- Το strjson object μεταφέρει όλη την πληροφορία σχετικά με τα πλοία. Το περιεχόμενο της πληροφορίας αυτής μεταβάλλεται κάθε φορά ανάλογα με τις επιλογές του χρήστη.

Μέσω της παραπάνω εντολής τα δεδομένα που βρίσκονται στα 3 αντικείμενα (objects) περνάνε μέσω της συνάρτησης (function) 'MyJsonFun' στα .htm αρχεία. Αναλυτικά ο κώδικας:

```

Private Sub WebBrowser1_DocumentCompleted(ByVal sender As Object, ByVal e As
System.Windows.Forms.WebBrowserDocumentCompletedEventArgs)
If (Not (WebBrowser1.Document Is Nothing)) Then
If e.Url.Equals(WebBrowser1.Url) Then
Dim strjson As Object = ""
Dim strZonesOfInterestList As Object = ""
Dim strPortList As Object = ""

If frmVTSMMain.chkPorts.Checked = True Then
Dim portlist As New List(Of clsPorts)
portlist = clsPorts.GetDataPortList()
strPortList = JsonConvert.SerializeObject(portlist,
Formatting.Indented)
Else
strPortList = "empty"
End If
If frmVTSMMain.chkZones.Checked = True Then
Dim ZonesOfInterestList As New List(Of clsZonesOfInterest)
ZonesOfInterestList = clsZonesOfInterest.GetZonesOfInterest()
strZonesOfInterestList =
JsonConvert.SerializeObject(ZonesOfInterestList,
Formatting.Indented)
Else
strZonesOfInterestList = "empty"
End If
If frmVTSMMain.blnInitial Then
strjson = "empty"
ElseIf frmVTSMMain.blnShips Then
Dim Shiplist As New List(Of clsVessel)
Shiplist = clsVessel.GetVesselsCurrentPos()
strjson = JsonConvert.SerializeObject(Shiplist, Formatting.Indented)
ElseIf frmVTSMMain.blnShipTrajectories Then
Dim VesselRoute As New List(Of clsVesselRoute)
VesselRoute = clsVesselRoute.GetVesselsCurrentLegRoute
strjson = JsonConvert.SerializeObject(VesselRoute,
Formatting.Indented)
ElseIf frmVTSMMain.blnNotificationMode Then
Dim ZonesOfInterestList As New List(Of clsZonesOfInterest)
ZonesOfInterestList = clsZonesOfInterest.GetZonesOfInterest()
strZonesOfInterestList =
JsonConvert.SerializeObject(ZonesOfInterestList,
Formatting.Indented)
Dim NotificationPointsList As New List(Of clsNotifications)
NotificationPointsList = clsNotifications.GetNotifications()
strjson = JsonConvert.SerializeObject(NotificationPointsList,
Formatting.Indented)
End If

frmVTSMMain.blnPorts = False
frmVTSMMain.blnShips = False
frmVTSMMain.blnShipTrajectories = False

WebBrowser1.Document.InvokeScript("MyJsonFun", New Object()
{strZonesOfInterestList, strPortList, strjson})
End If
End If
End Sub

```

Στη συνέχεια και μετά την εκτέλεση του παραπάνω κώδικα όλα τα δεδομένα μας έχουν μεταφερθεί στην πλευρά των .htm αρχείων.

Τα δεδομένα αυτά θεωρούνται ως JSON arrays και είναι της παρακάτω μορφής.

```
[
  {
    "VesselID": "205475000",
    "VesselName": "UNION JADE",
    "VesselType": "Not available",
    "DestinationPort": "Unknown",
    "DestinationCountry": "Unknown",
    "VesselColorIcon": "Lightgray",
    "Nationality": "Belgium",
    "VesselLat": 50.23883056640625,
    "VesselLon": -0.31014999747276306,
    "LastUpdateDate": "\Date(1378005078000+0300)\V",
    "PreVesselLat": 50.236549377441406,
    "PreVesselLon": -0.3303300142288208,
    "PreLastUpdateDate": "\Date(1378004797000+0300)\V",
    "DestinationLat": 0.0,
    "DestinationLon": 0.0,
    "LegID": "62"
  },
  {
    "VesselID": "205535000",
    "VesselName": "LOWLANDS ORCHID",
    "VesselType": "Cargo-All Types",
    "DestinationPort": "GUAIBA",
    "DestinationCountry": "Brazil",
    "VesselColorIcon": "LightGreen",
    "Nationality": "Belgium",
    "VesselLat": 50.302829742431641,
    "VesselLon": -1.3256000280380249,
    "LastUpdateDate": "\Date(1379798191000+0300)\V",
    "PreVesselLat": 50.303050994873047,
    "PreVesselLon": -1.3239200115203857,
    "PreLastUpdateDate": "\Date(1379798173000+0300)\V",
    "DestinationLat": -23.0068,
    "DestinationLon": -44.0352,
    "LegID": "82"
  },
  {
    "VesselID": "207017000",
    "VesselName": "GEORGI GRIGOROV",
    "VesselType": "Cargo-All Types",
    "DestinationPort": "FAWLEY",
    "DestinationCountry": "United Kingdom",
    "VesselColorIcon": "LightGreen",
    "Nationality": "Malta",
    "VesselLat": 50.052669525146484,
    "VesselLon": -2.6229500770568848,
    "LastUpdateDate": "\Date(1378791356000+0300)\V",
    "PreVesselLat": 50.0565185546875,
    "PreVesselLon": -2.6010200977325439,
    "PreLastUpdateDate": "\Date(1378791117000+0300)\V",
    "DestinationLat": 50.7525,
    "DestinationLon": -1.291,
    "LegID": "72"
  }
]
```

Εκεί έχουμε δημιουργήσει μια συνάρτηση (function) με το όνομα 'MyJsonFun' η οποία έχει ως παραμέτρους τα JSON objects τα οποία ορίσαμε στο κομμάτι του VB κώδικα. Η ονομασία της function αυτής και οι παράμετροι φαίνονται στο παρακάτω κομμάτι κώδικα.

```
function MyJsonFun (strZonesOfInterestList, strPortList, strjson) {
```

Μέσα στην function θα πρέπει ανάλογα με την κάθε περίπτωση (σε ποιες παραμέτρους υπάρχουν δεδομένα) να διαβάσουμε τα δεδομένα και να τα μεταφέρουν σε αντικείμενα της JavaScript αφού ουσιαστικά μεταφερθήκαν ως αλφαριθμητικά (string). Επομένως αντιστοιχώντας το καθένα από αυτά σε κάποιο

JavaScript object στη πορεία της εφαρμογής θα μπορεί να διαβαστεί κατάλληλα καθώς και να ανακτηθεί οποιοδήποτε από τις ιδιότητες του.

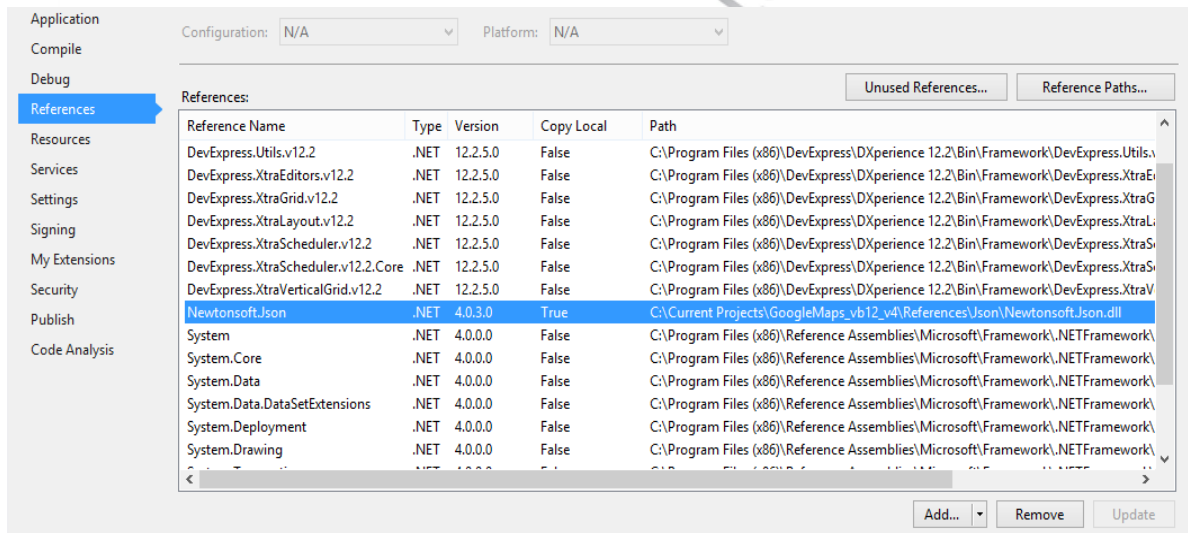
Αυτό επιτυγχάνεται με τις παρακάτω εντολές της JavaScript

```
var myJSONObject = strjson
var objJSON= JSON.parse(strjson)
```

Έτσι τελικά στο αντικείμενο (object) 'objJSON' βρίσκονται όλα τα δεδομένα τα οποία ζητήθηκαν μετά από τις επιλογές του χρήστη από την βάση δεδομένων, αποθηκεύτηκαν στις αντίστοιχες VB κλάσεις, στη συνέχεια μετατράπηκαν σε JSON objects ώστε μέσω του αντίστοιχου interface να περάσουν στα html αρχεία και τελικά να δημιουργηθούν κατάλληλα Objects όπου τα αρχικά δεδομένα πλέον έχουν την μορφή ιδιοτήτων τους.

Για να υπάρχει η δυνατότητα αυτής της χρήσης του JSON array ως interface ορίσαμε αρχικά ως reference στο Visual Studio το αντίστοιχο dll για το JSON δηλαδή το 'Newtonsoft.Json.dll'. Είναι το αντίστοιχο JSON Framework για το .NET.

Αυτό για εμάς είναι σαν ένα μαύρο κουτί. Ουσιαστικά περιέχει συγκεκριμένες κλάσεις τις οποίες εμείς τις εκμεταλλευόμαστε μέσω συγκεκριμένων εντολών αλλά χωρίς να επηρεάσουμε τίποτε στο τρόπο λειτουργίας του και στο κώδικα του.



Εικόνα 8-1: Καρτέλα reference αρχείων που χρησιμοποιήθηκαν στην εφαρμογή μέσα στο Visual Studio

Επίσης και από την πλευρά της JavaScript πρέπει να κάνουμε κάτι αντίστοιχο. Εκεί αυτό επιτυγχάνεται μέσω δύο συγκεκριμένων αρχείων που χρησιμοποιούνται ως βιβλιοθήκες. Του 'json2.js' και του 'json_parse.js'. Στη προκειμένη περίπτωση δεν εισάγουμε τα αρχεία ως reference αλλά μέσω των παρακάτω εντολών η JavaScript τα αναγνωρίζει και τα χρησιμοποιεί καταλλήλως ανάλογα με τις εντολές που της δίνουμε.

```
<script type="text/javascript" src="C:\Current
Projects\GoogleMaps_vb12_v4\References\Json\JSON-js-master\json2.js" >
</script>
<script type="text/javascript" src="C:\Current
Projects\GoogleMaps_vb12_v4\References\Json\JSON-js-master\json_parse.js" >
</script>
```


8.7 Εμφάνιση των InfoWindows

InfoWindows ονομάζονται τα παράθυρα πληροφορίας που εμφανίζονται στο χάρτη όταν ο χρήστης περάσει τον κέρσορα πάνω από κάποιον marker (mouse over). Η πληροφορία αυτή πρέπει να έχει ήδη περάσει στο .htm αρχείο ή να δημιουργηθεί μέσα σε αυτό ώστε να εμφανιστεί.

Παρακάτω φαίνεται ο κώδικας που χρησιμοποιήθηκε για την εμφάνιση του InfoWindow το οποίο μεταφέρει πληροφορία στο χρήστη όταν εκείνος κάνει mouse over σε κάποιο marker ενός πλοίου στη περίπτωση εμφάνισης των σημείων των πλοίων

```

var content = ( '<b>' + '<font size="4" color="#1E90FF">' + objJSON[i].VesselName +
'</font>' + ' (' + objJSON[i].VesselType + ')' + '</b>' +
" <br />" +
" <br />Latitude: " + objJSON[i].VesselLat +
" <br />Longitude: " + objJSON[i].VesselLon +
" <br />To : " + objJSON[i].DestinationPort + ' (' + objJSON[i].DestinationCountry
+ ')' +
" <br />Est. Speed: " + MyGetSpeed + " Knots" +
" <br />Heading: " + heading +
" <br />Mile To Port : " + milestoport +
" <br />ETA : " + ETA +
" <br />Flag: " + objJSON[i].Nationality
);

var timestamp = ('<P ALIGN="RIGHT">' + '<font size=2>' + MarkerLastUpdateDate +
'</font>' + '</P>');

var infowindow = new google.maps.InfoWindow({
  shadowStyle : 1,
  padding : 10,
  borderRadius : 8,
  width:192,
  height:30
});

infowindow.setContent(content + timestamp);

```

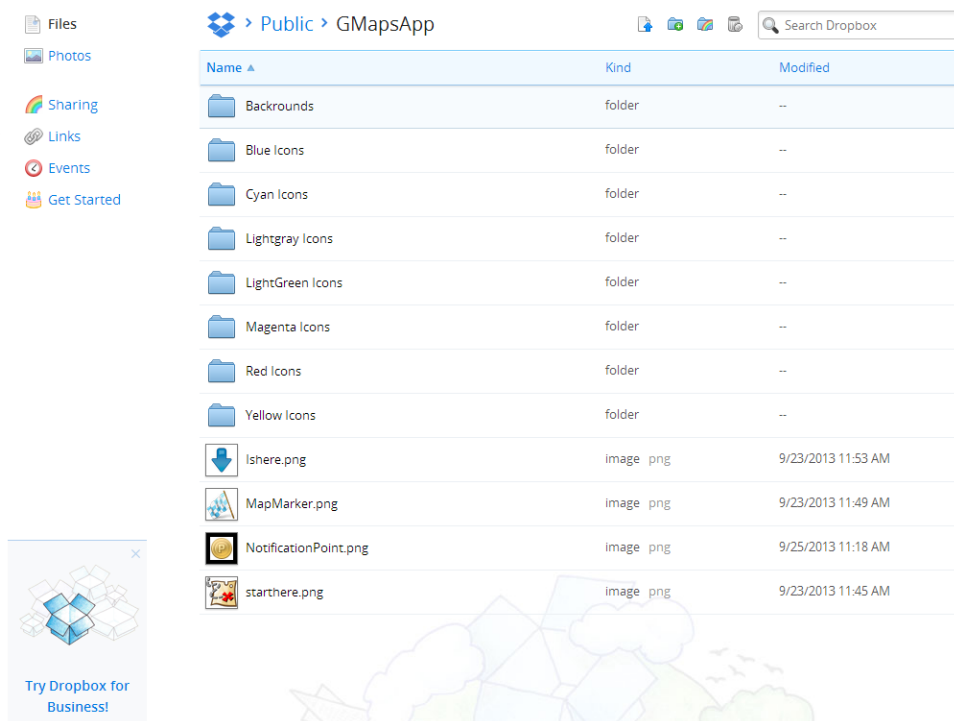
8.8 Τα icons των Markers μέσω της εφαρμογής Dropbox

Σε εφαρμογές που περιέχουν χάρτη όπως και στη παρούσα, πολλές φορές δημιουργείται η ανάγκη να χρησιμοποιηθούν εικονίδια (icons) για τα σημεία που αναπαριστούνται στο χάρτη (markers) διαφορετικά από τα προεπιλεγμένα και αυτά που παρέχει η ίδια η Google. Στις διαδικτυακές εφαρμογές τα εικονίδια αυτά βρίσκονται σε έναν Server τον οποίο ο χρήστης 'χτυπάει' ώστε να πάρει το εικονίδιο που τον ενδιαφέρει (το 'path' (διαδρομή του αρχείου) που βρίσκεται το εικονίδιο συγκεκριμένα) ώστε να το χρησιμοποιήσει στην εφαρμογή του.

Στην παρούσα εφαρμογή όμως δεν υπάρχει κάποια αντίστοιχη υλοποίηση server (ούτε καν τοπικού server) για να χρησιμοποιηθεί και να μπορέσουμε να διαβάσουμε τα paths των αντίστοιχων markers που θέλαμε να χρησιμοποιήσουμε. Έτσι θεωρήθηκε ως βέλτιστη λύση η χρησιμοποίηση της εφαρμογής 'Dropbox'.

Με την εφαρμογή αυτή ο χρήστης έχει την δυνατότητα να δημιουργήσει δημόσιους (public) φακέλους. Εκεί μπορεί να αποθηκεύσει οτιδήποτε τον ενδιαφέρει να έχει πρόσβαση σε αυτό μέσω μιας εφαρμογής ακόμα και μέσω άλλων συσκευών. Στην εφαρμογή αυτή ο χρήστης μπορεί να έχει πρόσβαση και διαδικτυακά αλλά και μέσω της δομής των φακέλων που ο ίδιος έχει δημιουργήσει στο desktop του.

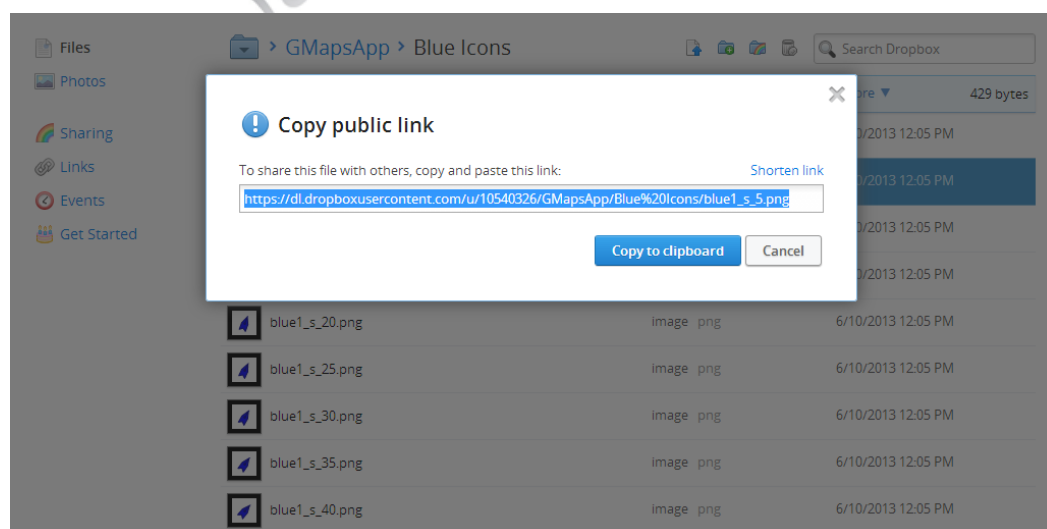
Παρακάτω φαίνεται η δομή που δημιουργήσαμε εμείς ώστε να αποθηκεύσουμε όλα τα εικονίδια των markers που θέλαμε να χρησιμοποιήσουμε.



Εικόνα 8-2: Εικόνα δημόσιων (public) φακέλων και αρχείων της εφαρμογής Dropbox που χρησιμοποιήθηκαν στην παρούσα εφαρμογή ως εικονίδια

Αντίστοιχα παρακάτω εμφανίζεται το link μέσω του οποίου μπορούμε να καλέσουμε κάποιο εικονίδιο ώστε να το εμφανίσουμε εκεί που επιθυμούμε. Είναι το 'Copy Public Link' το οποίο είναι ουσιαστικά το path του συγκεκριμένου εικονιδίου στο διαδίκτυο, κάτω από το http πρωτόκολλο.

Έτσι λοιπόν έχουμε για το κάθε εικονίδιο που μας ενδιαφέρει ένα συγκεκριμένο link με το οποίο μπορούμε να καλέσουμε το εικονίδιο αυτό και με αυτό τον τρόπο μπορέσαμε να παρακάμψουμε την χρήση οποιουδήποτε Server (τοπικού ή μη) μόνο και μόνο ως χώρου φύλαξης και παροχής εικονιδίων.



Εικόνα 8-3: Το public link που χρησιμοποιήθηκε για τον εντοπισμό των επιθυμητών αρχείων

Το συγκεκριμένο εμείς το υλοποιούμε μέσω JavaScript με τη χρήση του παραπάνω κομματιού κώδικα.

Ουσιαστικά τα εικονίδια της εφαρμογής είναι δύο κατηγοριών. Είναι τα εικονίδια των markers των λιμανιών τα οποία είναι όμοια σε κάθε περίπτωση και τα εικονίδια των markers των πλοίων τα οποία είναι διαφορετικά μεταξύ τους όπως θα δούμε παρακάτω.

Συγκεκριμένα για την εμφάνιση των εικονιδίων των λιμανιών χρησιμοποιούμε το παρακάτω κομμάτι JavaScript κώδικα:

Στην μεταβλητή iconBase ορίζεται ο φάκελος κάτω από τον οποίο είναι αποθηκευμένο το εικονίδιο και στη μεταβλητή image κρατείται το όνομα του εικονιδίου (αφού είναι το ίδιο για όλα τα λιμάνια) Έτσι στη συνέχεια που ορίζεται ο marker το εικονίδιο του (icon) ορίζεται ως η ένωση των παραπάνω μεταβλητών και επομένως έχουμε το συνολικό path του εικονιδίου στο Dropbox.

```
iconBase = 'https://dl.dropboxusercontent.com/u/13829837/';  
image = 'harbor%20(2).png';  
.  
.  
.  
var marker = new google.maps.Marker({  
    position: location,  
    map: map,  
    icon: iconBase + image,  
    html: msg  
});
```

Για την εμφάνιση των εικονιδίων των markers των πλοίων τα πράγματα είναι λίγο περισσότερο πολύπλοκα. Το δύσκολο στο συγκεκριμένο κομμάτι είναι ότι οι markers των πλοίων πρέπει να είναι διαφορετικοί ως προς το χρώμα ώστε να ξεχωρίζει ο τύπος του πλοίου κάθε φορά, ενώ επίσης θα πρέπει να περιγράφεται και η κατεύθυνση του πλοίου μέσα από το εικονίδιο. Δηλαδή να σχηματίζεται η γωνία που μας επιτρέπει να έχουμε μια σωστή αναπαράσταση της κατεύθυνσης και πορείας του πλοίου κάθε φορά.

Για να επιτευχθεί αυτό χρησιμοποιήθηκαν εικονίδια πλοίων τα οποία θα πρέπει να αναπαριστούν την οποιαδήποτε κατεύθυνση είναι δυνατόν να έχει το πλοίο. Έτσι ορίσαμε ότι σε σύνολο 360 μοιρών καλό θα είναι να έχουμε ένα εικονίδιο αν 5 μοίρες. Δηλαδή χρειαζόμαστε ακριβώς 72 εικονίδια αφού $360/72 = 5$. Τα εικονίδια που χρησιμοποιήσαμε λοιπόν είναι της παρακάτω μορφής. Δηλαδή εικονίδια με κλήση από 0 μέχρι 360 μοίρες ανά 5 μοίρες.



Εικόνα 8-4: Δείγμα εικονιδίων που χρησιμοποιήθηκε στην παρούσα εφαρμογή για την απεικόνιση των πλοίων, ανάλογα με την πορεία τους σε μοίρες

Επίσης τα συγκεκριμένα είναι χρώματος μπλε. Αντίστοιχα για διαφορετικούς τύπους πλοίων υπάρχουν διαφορετικού χρώματος εικονίδια. Έτσι λοιπόν έχοντας όλα τα παραπάνω υπόψη με τη χρήση του παρακάτω κώδικα πραγματοποιείται η αναπαράσταση του κάθε marker με το σωστό χρώματος εικονίδιο αλλά και με την σωστή κάθε φορά κατεύθυνση

Πιο συγκεκριμένα παρακάτω γίνεται έλεγχος του χρώματος με το οποίο αναπαριστάτε το κάθε πλοίο και αναλόγως ορίζεται η iconBase μεταβλητή καθώς και η ShipColorIcon.

```

if (objJSON[i].VesselColorIcon == "Blue") {
    iconBase =
'https://dl.dropboxusercontent.com/u/13829837/GMapsApp/Blue%20Icons/';
    ShipColorIcon = 'blue1_s_'
}
else if (objJSON[i].VesselColorIcon == "Cyan") {
    iconBase =
'https://dl.dropboxusercontent.com/u/13829837/GMapsApp/Cyan%20Icons/';
    ShipColorIcon = 'cyan1_s_'
}
else if (objJSON[i].VesselColorIcon == "LightGreen") {
    iconBase =
'https://dl.dropboxusercontent.com/u/13829837/GMapsApp/LightGreen%20Icons/';
    ShipColorIcon = 'lightgreen1_s_'
}
else if (objJSON[i].VesselColorIcon == "Red") {
    iconBase =
'https://dl.dropboxusercontent.com/u/13829837/GMapsApp/Red%20Icons/';
    ShipColorIcon = 'red1_s_'
}
else if (objJSON[i].VesselColorIcon == "Lightgray") {
    iconBase =
'https://dl.dropboxusercontent.com/u/13829837/GMapsApp/Lightgray%20Icons/';
    ShipColorIcon = 'lightgray1_s_'
}
else if (objJSON[i].VesselColorIcon == "Yellow") {
    iconBase =
'https://dl.dropboxusercontent.com/u/13829837/GMapsApp/Yellow%20Icons/';
    ShipColorIcon = 'yellow1_s_'
}
}

```

Με το παραπάνω έχουμε ορίσει πιο θα είναι το path ώστε να βρούμε το κατάλληλο χρώματος εικονίδιο για το κάθε πλοίο. Στη συνέχεια πρέπει να δούμε ποιας γωνίας εικονίδιο θα χρησιμοποιήσουμε ελέγχοντας τη γωνία που είναι αποθηκευμένη στις ιδιότητες του JSON object για το κάθε πλοίο.

Έτσι λοιπόν στο παρακάτω κώδικα υπολογίζουμε το heading (κατεύθυνση σε μοίρες) για το δεδομένο σημείου του κάθε πλοίου. Το Heading μπορεί να είναι οποιοσδήποτε αριθμός μεταξύ του 0 και του 360. Επειδή εμείς έχουμε εικονίδια μόνο αν 5 μοίρες θα πρέπει να βρούμε ποιο εικονίδιο θα ταιριάζει καλύτερα στο heading που βρίσκουμε κάθε φορά. Επομένως χρησιμοποιώντας μόνο το τελευταίο ψηφίο του αριθμού και με ένα αλγόριθμο που δημιουργήσαμε βρίσκουμε σε ποιο εικονίδιο αντιστοιχεί καλύτερα το κάθε heading που υπολογίζουμε. Αν θα πρέπει δηλαδή το heading που ισούται με 172 να καλέσει το εικονίδιο 170 ή το 175. Στην προκειμένη περίπτωση θα χρησιμοποιήσει το 170.

Τελικά τα σωστά εικονίδιο σώζεται στη μεταβλητή image και αντιστοιχίζεται στις ιδιότητες του αντίστοιχου marker.

```
var heading = google.maps.geometry.spherical.computeHeading(preshippos, shipcurpos);
    var heading = Math.round(heading);
    if (heading < 0) {
        heading = (heading + 360)
    }
    //Get only the last digit in order to check which icon you are going to use
    var HeadLastDigit = heading % 10;

if (HeadLastDigit == 0) {
    image = ShipColorIcon + heading + '.png';
}
else if (HeadLastDigit == 5) {
    image = ShipColorIcon + heading + '.png';
}
else {
    var GetBearingValue_5 = heading - (heading % 10) + 5;
    var GetBearingValue_0 = heading - (heading % 10) + 0;
    var rBLD_5 = (heading - heading)
    if (rBLD_5 > 0) {
        if (rBLD_5 >= 3){
            heading = heading + (5 - rBLD_5 )
        }
        else {
            heading = GetBearingValue_5
        }
    }
    else {
        if (rBLD_5 > -3) {
            heading = GetBearingValue_5
        }
        else {
            heading = GetBearingValue_0
        }
    }
}
image = ShipColorIcon + heading + '.png';

var marker = new google.maps.Marker({
    position: location,
    map: map,
    icon: iconBase + image
});
```

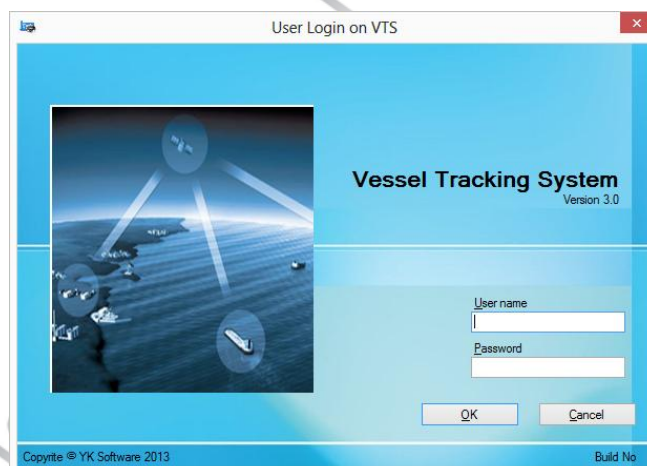

9. Περιγραφή λειτουργικότητας

Το Vessel Tracking System είναι μια desktop εφαρμογή η οποία επιτρέπει στον χρήστη μιας εταιρίας να λαμβάνει πληροφορίες σχετικά με τον στόλο των πλοίων της εταιρίας. Η πληροφορία που λαμβάνεται είναι σχετική με τα διαθέσιμα πλοία της εταιρίας και αφορούν την θέση, την τροχιά καθώς και άλλες πληροφορίες σχετικές με τα πλοία που θα δούμε στη συνέχεια. Προσπαιτούμενο για να λειτουργήσει σωστά η εφαρμογή είναι ο υπολογιστής στον οποίο θα εγκατασταθεί να έχει σύνδεση με τον διαδίκτυο ώστε να λαμβάνονται αλλά και να αποτυπώνονται όλες οι πληροφορίες που χρειάζονται.

Η βασική οθόνη της εφαρμογής αποτελείται από 2 βασικά μέρη. Το κομμάτι που εμφανίζεται ο χάρτης με όλες τις σχετικές πληροφορίες που αφορούν τα πλοία του στόλου της εταιρίας και το κομμάτι το οποίο χειρίζεται ο χρήστης κάνοντας τις επιλογές του ανάλογα με την πληροφορία που θέλει να αποκομίσει. Το κομμάτι αυτό των επιλογών του χρήστη μπορούμε να το χωρίσουμε σε τρία επιμέρους τμήματα. Το τμήμα της πληροφορίας σχετικά με τα πλοία (Vessel Information Tab), το τμήμα της πληροφορίας σχετικά με τις ειδοποιήσεις προς τον χρήστη (Notification Tab) και τέλος το τμήμα σχετικά με τη διαχείριση της εφαρμογής (Administrator Tab).

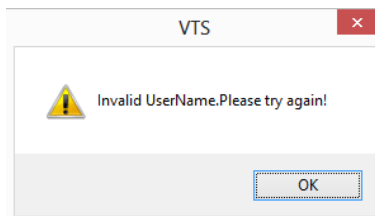
9.1 Οθόνη Σύνδεσης Χρήστη

Ανοίγοντας την εφαρμογή εμφανίζεται η οθόνη σύνδεσης του χρήστη (log in screen) που φαίνεται παρακάτω. Ο χρήστης πληκτρολογεί το προσωπικό όνομα χρήσης (User name) και συνθηματικό (Password) του και επιλέγει το 'OK' για να συνδεθεί.



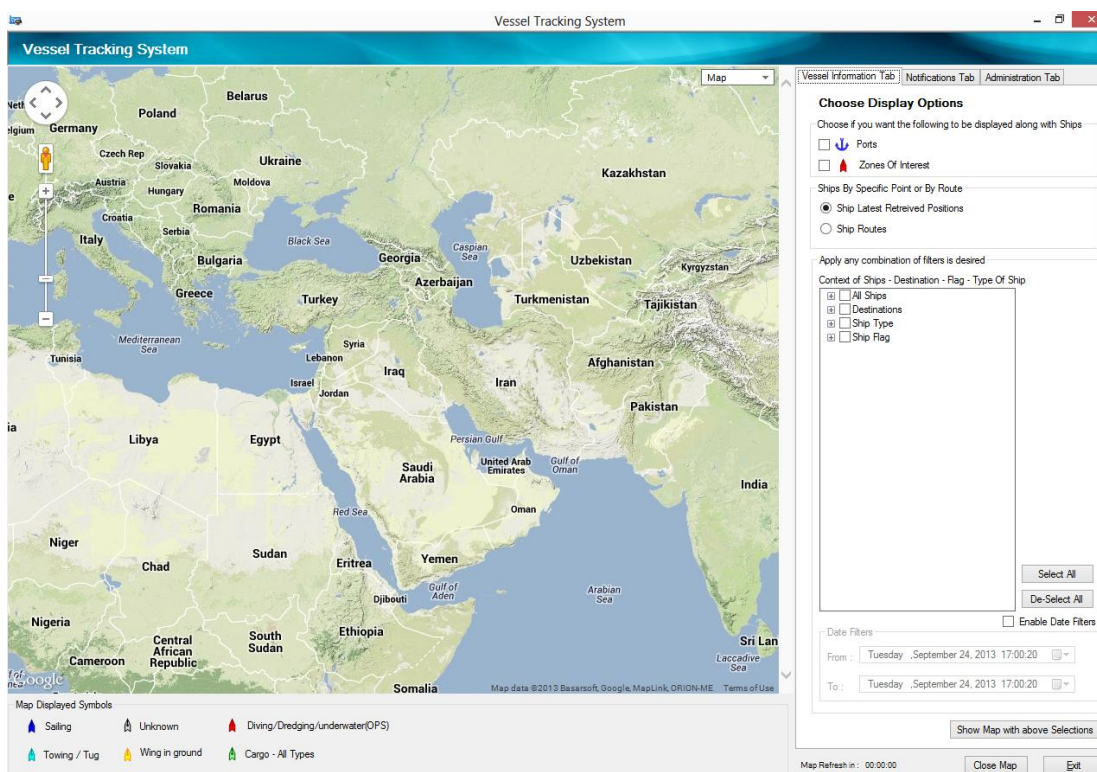
Εικόνα 9-1: Οθόνη σύνδεσης χρήστη

Σε περίπτωση λάθους ένα μήνυμα της παρακάτω μορφής ενημερώνει το χρήστη για τον λάθος του. Στην συνέχεια και επιλέγοντας 'OK' ο χρήστης ξανά-δοκιμάζει να εισέλθει στην εφαρμογή



Εικόνα 9-2: Μήνυμα λάθους εισαγωγής χρήστη

Μόλις ο χρήστης συνδεθεί επιτυχώς, η παρακάτω οθόνη εμφανίζεται στο χρήστη.



Εικόνα 9-3: Γενική εικόνα εφαρμογής

Στο μεγαλύτερο μέρος της οθόνης εμφανίζεται ο χάρτης (χωρίς κάποια πληροφορία ακόμα) και στα δεξιά της οθόνης εμφανίζονται οι καρτέλες μέσω των οποίων ο χρήστης κάνει συγκεκριμένες επιλογές ανάλογα με την επιθυμητή πληροφορία που χρειάζεται.

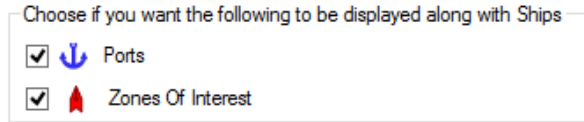
9.2 Καρτέλα πληροφορίας πλοίων (Vessel Information Oriented Tab)

Σε αυτή την καρτέλα ο χρήστης έχει τη δυνατότητα να επιλέξει αν θέλει να δει στον χάρτη μαζί με τα πλοία, τα λιμάνια και τις ζώνες ενδιαφέροντος. Επίσης μπορεί να επιλέξει αν θέλει να δει τις τελευταίες τρέχουσες θέσεις των πλοίων ή τις τροχιές αυτών ανά leg (τροχιά ανά προορισμό) καθώς και να φιλτράρει τις επιλογές του χρησιμοποιώντας φίλτρα όπως το όνομα πλοίου, ο προορισμός του, τον τύπο του πλοίου, την εθνικότητα του καθώς και φίλτρα ανάμεσα σε συγκεκριμένες ημερομηνίες. Στη συνέχεια αν ο χρήστης επιλέξει το “Show Map with above Selections” θα εμφανιστούν στο χάρτη οι πληροφορίες ανάλογα με τις παραπάνω επιλογές του.

9.2.1 Λιμάνια – Ζώνες Ενδιαφέροντος (Ports – Zones Of Interest)

Ports: Ο χρήστης μπορεί να επιλέξει το συγκεκριμένο κουτάκι (checkbox) παράλληλα με οποιαδήποτε άλλη επιλογή θέλει να κάνει. Αν το επιλέξει στη συνέχεια που θα επιλέξει το “Show Map with above Selections” θα εμφανιστούν στον χάρτη όλα τα εισηγμένα λιμάνια. Τα λιμάνια εμφανίζονται με τη χρήση markers χρώματος μπλε και σχήματος άγκυρας.

Zones of Interest: Όπως παραπάνω ο χρήστης μπορεί να επιλέξει το συγκεκριμένο κουτάκι (checkbox) παράλληλα με οποιαδήποτε άλλη επιλογή θέλει να κάνει. Αν το επιλέξει στη συνέχεια που θα επιλέξει το “Show Map with above Selections” θα εμφανιστούν στον χάρτη όλες οι εισηγμένες ζώνες ενδιαφέροντος. Οι ζώνες ενδιαφέροντος εμφανίζονται στο χάρτη με ροζ χρώμα.



9.2.2 Εικόνα 9-4: Επιλογή εισαγωγής Λιμανιών και Ζωνών Ενδιαφέροντος

9.2.3 Εμφάνιση τελευταίας ληφθείσας θέσης πλοίου (Ship Latest Received Positions)

Ship Latest Received Positions: Εμφανίζει τις τελευταίες εισηγμένες θέσεις των πλοίων. Οι θέσεις αυτές δεν θεωρούνται οι τρέχουσες θέσεις (live positions) αλλά τα σημεία των πλοίων που ελήφθησαν τελευταία στη βάση δεδομένων.

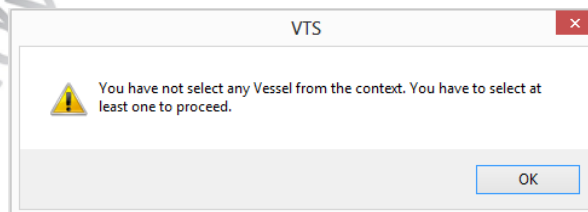


Εικόνα 9-5: Επιλογή φίλτρων εμφάνισης θέσης πλοίων

Αυτή η επιλογή μαζί με την ‘Ship Routes’ που θα δούμε παρακάτω είναι διαζευκτικές επιλογές για τον χρήστη αφού μπορεί να επιλέξει να δει ένα από τα δύο κάθε φορά και όχι και τα δύο μαζί.

Στη συνέχεια ο χρήστης θα πρέπει να επιλέξει τα φίλτρα ανάλογα με το τι θέλει να εμφανίσει στο χάρτη. Η πρώτη επιλογή των φίλτρων γίνεται ανάμεσα σε συγκεκριμένα πλοία, προορισμό πλοίων, τύπο πλοίων και εθνικότητα.

Ο χρήστης πρέπει οπωσδήποτε να επιλέξει ένα πλοίο τουλάχιστον για να μπορέσει να δει τα αποτελέσματα στο χάρτη. Αν δεν επιλέξει κάποιο πλοίο το παρακάτω μήνυμα τον ενημερώνει γι’ αυτό.



Εικόνα 9-6: Εμφάνιση μηνύματος λάθους στη περίπτωση μη επιλογής πλοίων

9.2.4 Εμφάνιση διαδρομών πλοίων (Ship Routes)

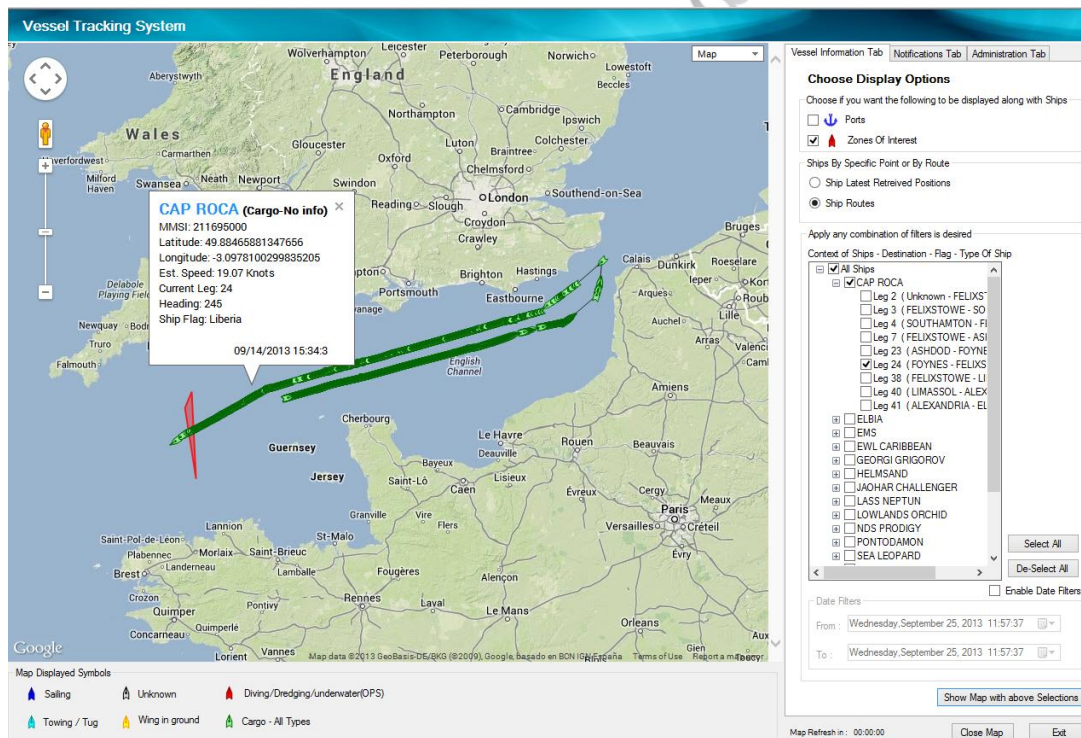
Εμφανίζει στο χάρτη τη διαδρομή του επιλεγμένου πλοίου ανά λιμάνι προορισμού.

Ships By Specific Point or By Route

- Ship Latest Retrieved Positions
 Ship Routes

Εικόνα 9-7: Εμφάνιση φίλτρων εμφάνισης θέσης πλοίων ή διαδρομών πλοίων

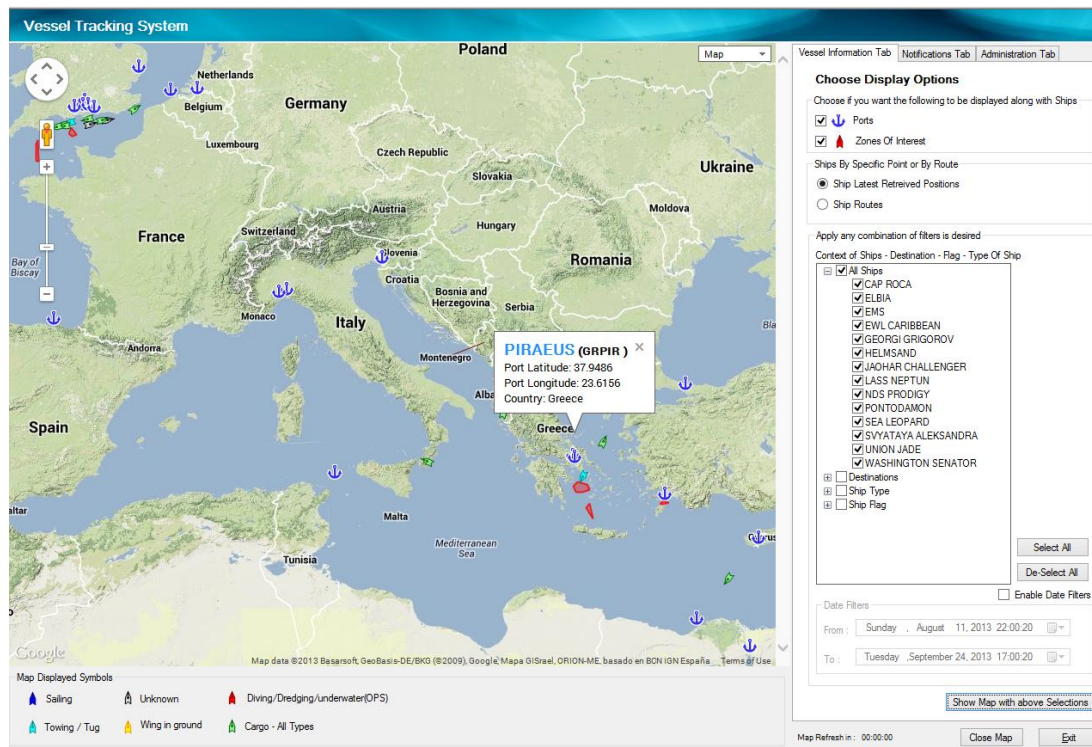
Στην περίπτωση που ο χρήστης επιλέξει το “Show Routes” όπως αναφέρθηκε παραπάνω θα είναι σε θέση να εμφανίσει ολόκληρη τη διαδρομή του πλοίου. Η διαδρομή αυτή ‘χτίζεται’ χρησιμοποιώντας όλα τα σημεία της θέσης του πλοίου ανά λιμάνι προορισμού. Στην κατηγορία αυτή των διαδρομών του πλοίου εισέρχεται και η έννοια του leg η οποία ορίζεται ως μια διαδρομή ανά λιμάνι προορισμού σε λιμάνι προορισμού. Επομένως ο χρήστης έχει τη δυνατότητα να επιλέξει και συγκεκριμένο leg για το κάθε πλοίο ώστε να εμφανίσει αυτό ακριβώς που επιθυμεί.



Εικόνα 9-8: Εμφάνιση διαδρομής επιλεγμένου πλοίου και του αντίστοιχου InfoWindow

9.2.5 Φόρτωση του χάρτη

Παρακάτω βλέπουμε τον χάρτη που εμφανίζεται όταν ο χρήστης έχει επιλέξει να εμφανίσει ‘Ports’ – ‘Zones Of Interest’ – ‘Ship Latest Received Positions’ και ‘All Ships’



Εικόνα 9-9: Εμφάνιση λιμανιών, περιοχών ενδιαφέροντος και θέσης πλοίων στον χάρτη

Ο χρήστης στη συνέχεια περνώντας τον κέρσορα πάνω από το οποιοδήποτε λιμάνι είναι σε θέση να δει σχετικές πληροφορίες όπως φαίνεται παρακάτω.

PIRAEUS (GRPIR) ✕
 Port Latitude: 37.9486
 Port Longitude: 23.6156
 Country: Greece

Εικόνα 9-10: InfoWindow με τα χαρακτηριστικά του λιμανιού του Πειραιά

Port Name: Το όνομα του λιμανιού.

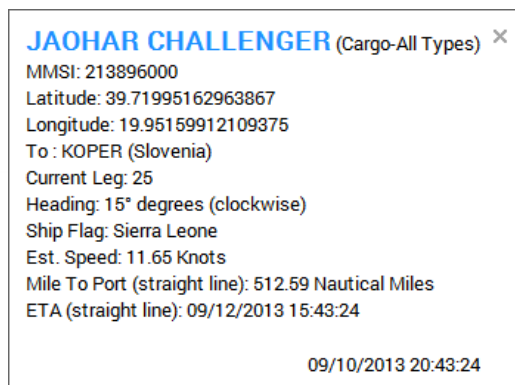
Short Name Port: Το σύντομο όνομα του λιμανιού σύμφωνα με την διεθνές ορολογία

Port Latitude: Το γεωγραφικό μήκος που αντιστοιχεί στο σημείο που βρίσκεται το λιμάνι

Port Longitude: Το γεωγραφικό πλάτος που αντιστοιχεί στο σημείο που βρίσκεται το λιμάνι

Country: Την χώρα στην οποία ανήκει το λιμάνι

Επίσης περνώντας τον κέρσορα πάνω από κάποιο πλοίο (mouse over) είναι σε θέση να πάρει όλες τις πληροφορίες που αφορούν αυτό το πλοίο μέσω του παραθύρου πληροφορίας (information window). Η πληροφορία που εμφανίζεται στο χρήστη περιλαμβάνει τα παρακάτω για το κάθε πλοίο.



Εικόνα 9-11: InfoWindow πλοίου και η πληροφορία συγκεκριμένου στίγματος

Name: Το όνομα του πλοίου

Vessel Type: Ο τύπος του πλοίου

MMSI: Ο μοναδικός αριθμός που αντιπροσωπεύει το συγκεκριμένο πλοίο

Latitude: Συντεταγμένη του γεωγραφικού πλάτους που καθορίζει τη θέση του πλοίου στην επιφάνεια της Γης.

Longitude: Συντεταγμένη του γεωγραφικού μήκους που καθορίζει τη θέση του πλοίου στην επιφάνεια της Γης.

To: Ο προκαθορισμένος προορισμός του πλοίου

Current Leg: Η ταυτότητα (id) της τρέχουσας διαδρομής του πλοίου. Αφορά την διαδρομή του από το λιμάνι αναχώρησης προς το λιμάνι προορισμού.

Heading: Οι γεωμετρικές μοίρες κατεύθυνσης της τρέχουσας πορείας του πλοίου ακολουθώντας τη φορά των δεικτών του ρολογιού.

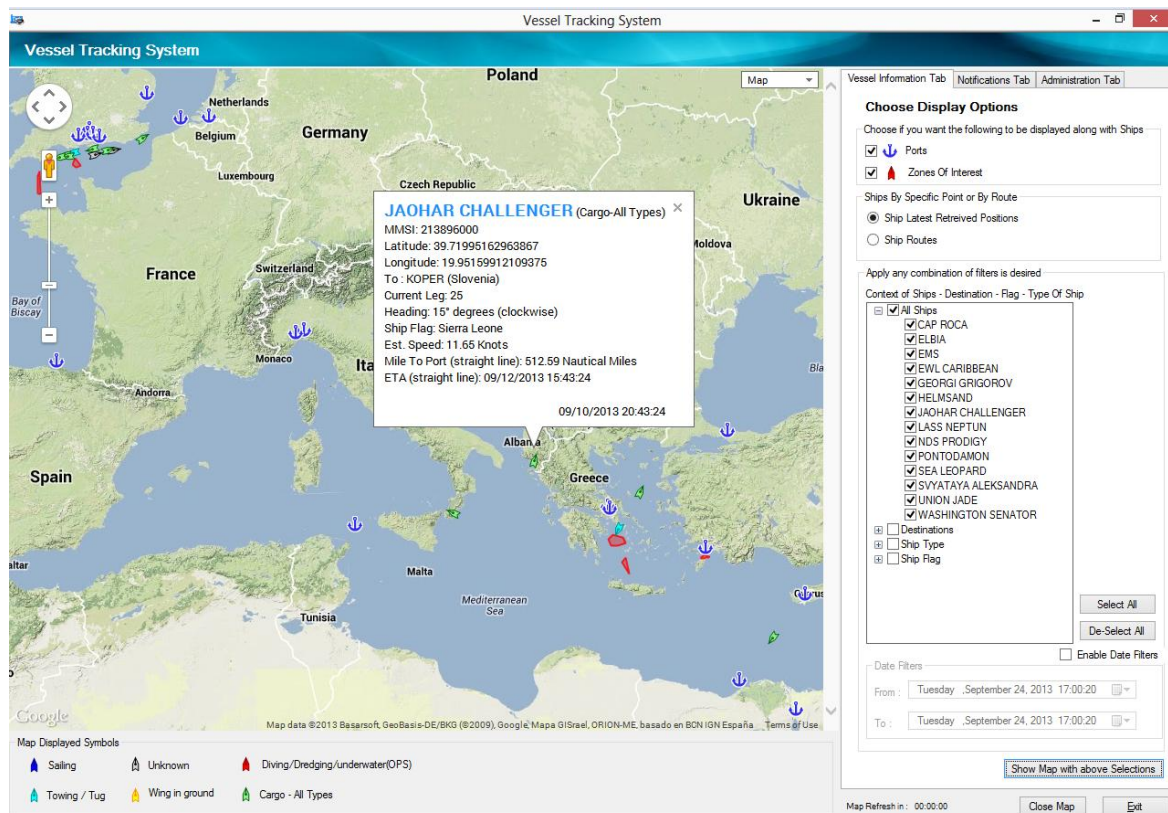
Ship Flag: Η εθνικότητα υπό τη οποία έχει καταχωρηθεί το πλοίο ώστε να καθορίζεται η Νομική δικαιοδοσία βάση της οποίας λειτουργεί.

Est. Speed (Estimation Speed): Εκτιμώμενη ταχύτητα. Υπολογίζεται λαμβάνοντας υπόψη το χρόνο που ελήφθη το προηγούμενο σημείο του πλοίου και το χρόνο που χρειάστηκε να διανύσει το πλοίο μέχρι το τρέχον σημείο.

Miles To Port: Η απόσταση από το τρέχον σημείο του πλοίου μέχρι το λιμάνι προορισμού, Η απόσταση αυτή εμφανίζεται σε μίλια και υπολογίζεται ως η ευκλείδεια απόσταση μεταξύ του τρέχοντος σημείου που βρίσκεται το πλοίο και του λιμανιού.

ETA: (Estimation Time of Arrival): Είναι ένα μέτρο που δείχνει το πότε το πλοίο θα φτάσει στο λιμάνι προορισμού του. Λαμβάνεται υπόψη η ευκλείδεια απόσταση από το λιμάνι προορισμού και υπολογίζεται ο χρόνος που απαιτείται ώστε το πλοίο να φτάσει στο λιμάνι.

Update Date: Η ημερομηνία που ελήφθη το συγκεκριμένο σημείο του πλοίου.



Εικόνα 9-12: Η συνολική εικόνα της εφαρμογής με εμφάνιση στον χάρτη όλων των επιλεγμένων από το φίλτρο πλοίων και το InfoWindow του επιλεγμένου στον χάρτη πλοίου

9.2.6 Διαθέσιμα φίλτρα με βάση τα στοιχεία των πλοίων

Επίσης μπορεί να κάνει συγκεκριμένες επιλογές χρησιμοποιώντας τα φίλτρα ονόματος πλοίου (Ships), προορισμού (Destination), τύπου πλοίου (Type of Ship) ή σημαίας πλοίου (Flag) από το δέντρο όπως εμφανίζεται παρακάτω.

Επίσης έχει τη δυνατότητα να επιλέξει το "Select All" για να επιλέξει όλα τα κουτάκια του δέντρου καθώς και το "De-Select All" για να καταργήσει όλες τις επιλογές.

Context of Ships - Destination - Flag - Type Of Ship

All Ships

- CAP ROCA
- ELBIA
- EMS
- EWL CARIBBEAN
- GEORGI GRIGOROV
- HELMSAND
- JAOHAR CHALLENGER
- LASS NEPTUN
- NDS PRODIGY
- PONTODAMON
- SEA LEOPARD
- SVYATAYA ALEKSANDRA
- UNION JADE
- WASHINGTON SENATOR

 Destinations
 Ship Type
 Ship Flag

Εικόνα 9-13: Πολλαπλά φίλτρα επιλογής Ονόματος, Προορισμού, Τύπου και Εθνικότητας πλοίων

9.2.7 Διαθέσιμο φίλτρο με βάση την ημερομηνία

Ο χρήστης επιλέγοντας το κουτάκι στο 'Enable Date Filters' ενεργοποιεί και το φίλτρο της ημερομηνίας και της ώρας. Έτσι από τη στιγμή που αυτό είναι ενεργοποιημένο θα πρέπει να επιλέξει το συγκεκριμένο διάστημα των ημερομηνιών που επιθυμεί αφού αυτό θα λαμβάνεται υπόψη στα αποτελέσματα που θα του επιστρέφει η εφαρμογή.

Αντίστοιχα μπορεί να απενεργοποιήσει το συγκεκριμένο φίλτρο καταργώντας την επιλογή 'Enable Date Filters' ώστε να μην λαμβάνεται υπόψη στα αποτελέσματα της αναζήτησής του.

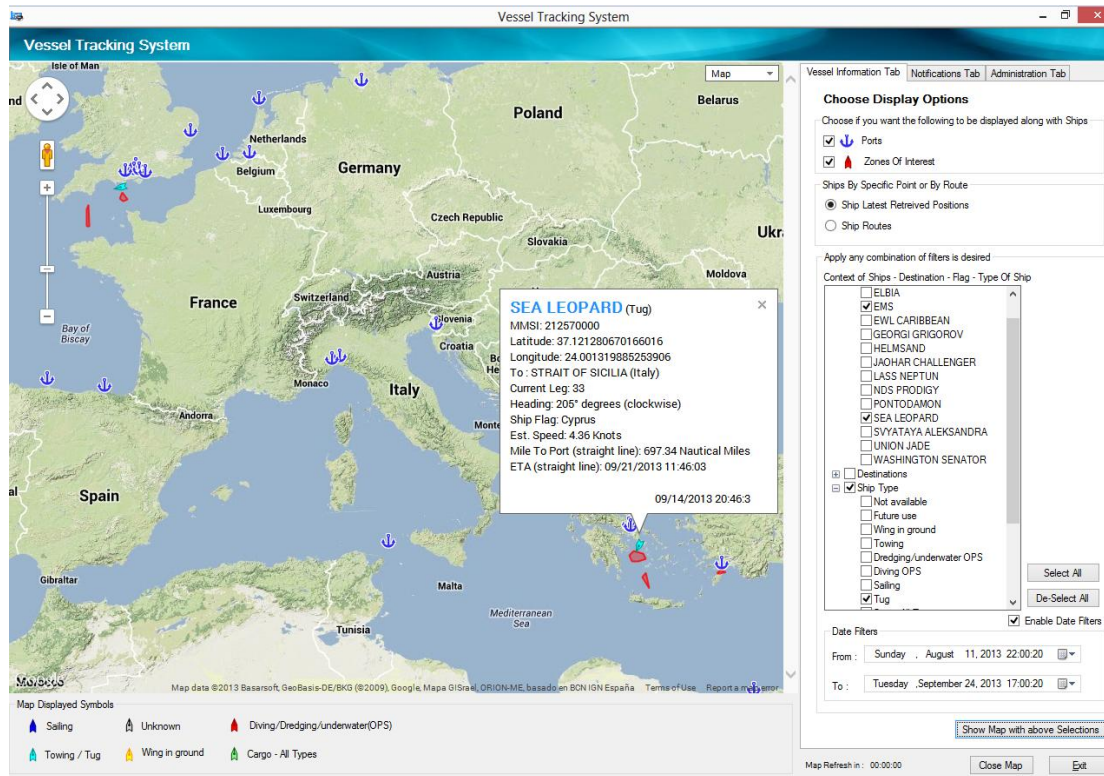
Enable Date Filters

Date Filters

From : Sunday, August 11, 2013 22:00:19

To : Wednesday, September 25, 2013 14:30:19

Εικόνα 9-14: Φίλτρα επιλογής ημερομηνίας

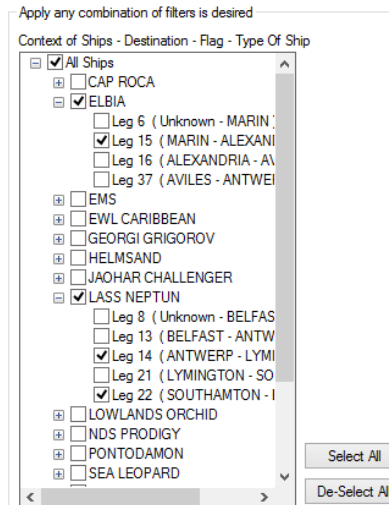


Εικόνα 9-15: Εφαρμογή επιλεγμένων φίλτρων στον χάρτη

Βλέπουμε παραπάνω ότι έχουν επιλεχθεί συγκεκριμένα πλοία συγκεκριμένου τύπου και σε κάποιο συγκεκριμένο εύρος ημερομηνιών. Παρατηρούμε ότι όντως τα φίλτρα έχουν εφαρμοστεί και στον χάρτη.

Το Information Window εμφανίζει κάποιες από τις πληροφορίες της προηγούμενης επιλογής ‘Ship Latest Received Positions’.

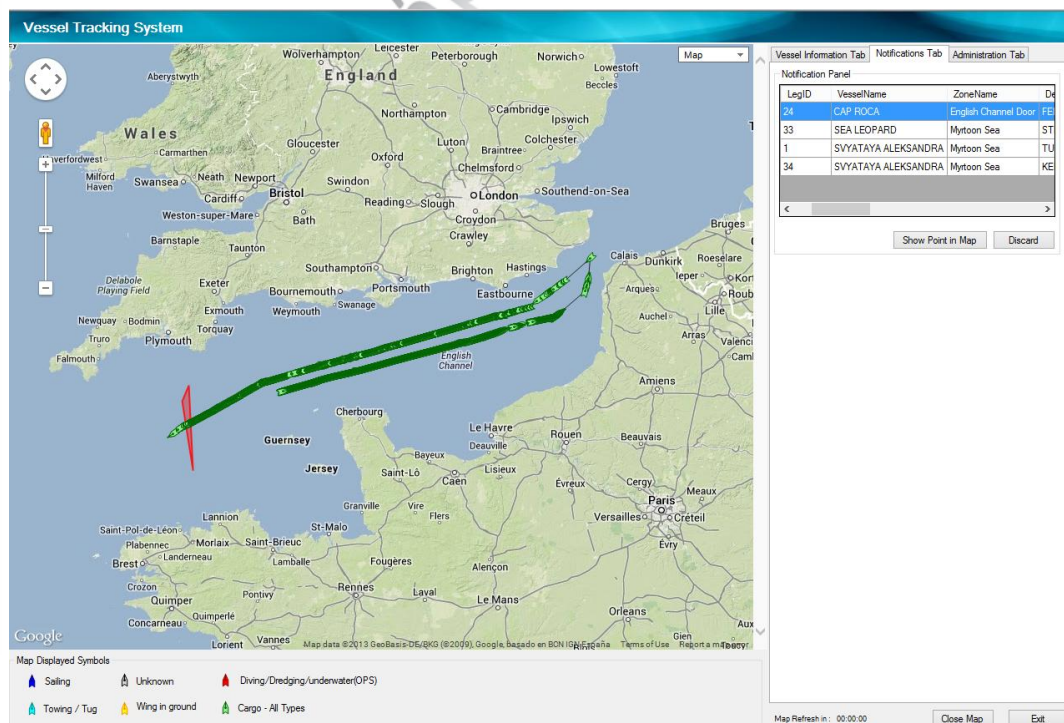
Επίσης όλα τα φίλτρα μπορούν να εφαρμοστούν με τον ίδιο τρόπο όπως και παραπάνω καθώς και το φίλτρο της ημερομηνίας και ώρας. Η μόνη διαφοροποίηση είναι ότι προστίθεται και η διαδρομή (Leg Id) ως φίλτρο αναφορικά με το εκάστοτε επιλεγμένο πλοίο. Έτσι ο χρήστης μπορεί να κάνει πολλαπλές επιλογές πλοίων και διαδρομών αυτών όπως φαίνεται παρακάτω. Για λόγους ενημέρωσης του χρήστη έχει προστεθεί δίπλα στην πληροφορία του leg id και το λιμάνι αναχώρησης και προορισμού που αποτελούν την εκάστοτε διαδρομή.



Εικόνα 9-16: Επιλογή συγκεκριμένων πλοίων καθώς και συγκεκριμένων διαδρομών (legs)

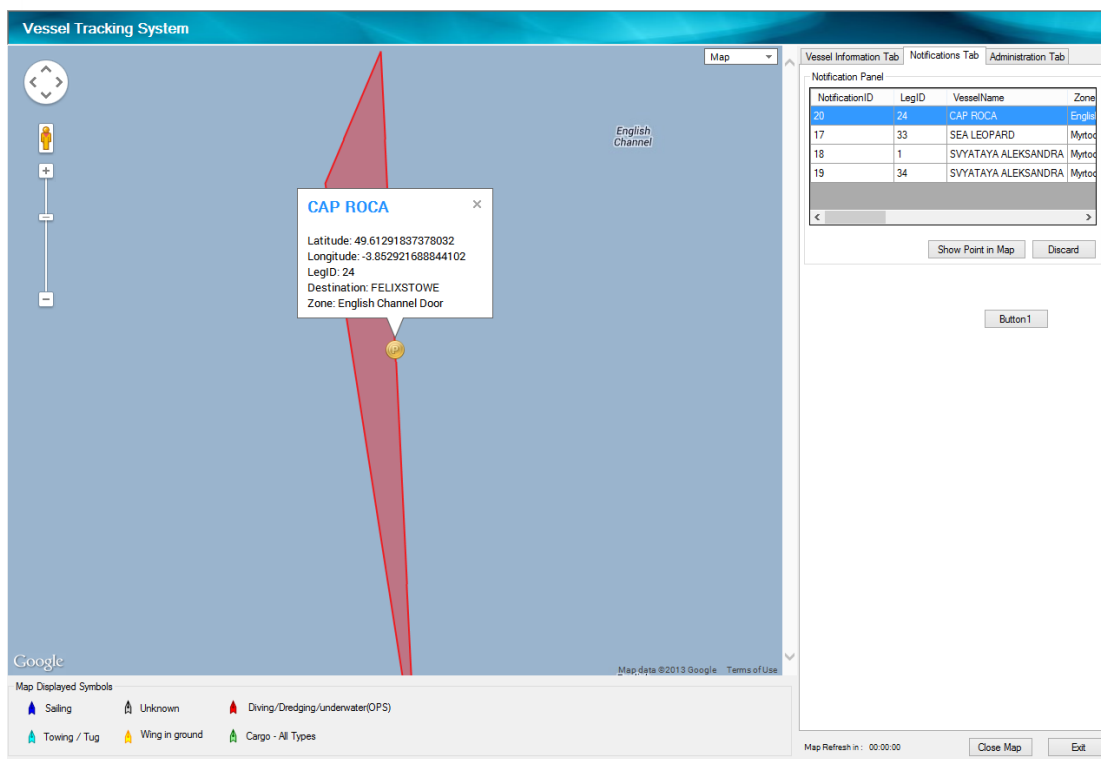
9.3 Καρτέλα εμφάνισης ειδοποιήσεων (Vessel Notification Oriented Tab)

Η καρτέλα αυτή εμφανίζει ειδοποιήσεις στον χρήστη στη περίπτωση που κάποιο πλοίο εισέλθει σε οποιαδήποτε ζώνη ενδιαφέροντος ώστε να προβεί στις ενέργειες που απαιτούνται ανάλογα με της κατευθύνσεις της εκάστοτε εταιρίας. Επίσης ο χρήστης έχει τη δυνατότητα να εμφανίσει στο χάρτη το ακριβές σημείο που το πλοίο εισήλθε στη ζώνη ενδιαφέροντος καθώς και σχετικές πληροφορίες για το πλοίο.



Εικόνα 9-17: Information Tab: Εμφανίζει ειδοποιήσεις για τα πλοία η διαδρομή των οποίων έχει εισέλθει μέσα από κάποια ζώνη ενδιαφέροντος

Βλέπουμε στις δύο παραπάνω εικόνες ότι το πλοίο που εισήλθε σε μία ζώνη ενδιαφέροντος εμφανίζεται στο Notification Panel μαζί με ορισμένες σχετικές πληροφορίες ώστε να μπορέσει ο χρήστης να ειδοποιηθεί κατάλληλα. Επίσης όπως φαίνεται παρακάτω το Notification Panel δίνει τη δυνατότητα αλληλεπίδρασης με το χάρτη αφού ο χρήστης μπορεί να επιλέξει οποιαδήποτε εγγραφή και επιλέγοντας το 'Show Point in Map' να δει το ακριβές σημείο όπου το πλοίο εισήλθε στην ζώνη ενδιαφέροντος.



Εικόνα 9-18: Το ακριβές σημείο στο οποίο το πλοίο εισήλθε στην ζώνη ενδιαφέροντος

Επίσης διαπερνώντας ο χρήστης πάνω από το σημείο έχει τη δυνατότητα να δει συγκεκριμένες πληροφορίες όπως εμφανίζονται παρακάτω



Εικόνα 9-19: Το InfoWindow του σημείου εισαγωγής ενός πλοίου σε μια ζώνη ενδιαφέροντος

Vessel Name: Το όνομα του πλοίου που εισήλθε στη ζώνη ενδιαφέροντος.

Latitude: Το γεωγραφικό μήκος του σημείου που εισήλθε το πλοίο

Longitude: Το γεωγραφικό πλάτος του σημείου που εισήλθε το πλοίο

Leg ID: Η ταυτότητα (id) της τρέχουσας διαδρομής του πλοίου. Αφορά την διαδρομή του από λιμάνι αναχώρησης προς το λιμάνι προορισμού.

Destination: Το λιμάνι προορισμού του πλοίου κατά τη διάρκεια της διαδρομής του μέσα από τη ζώνη ενδιαφέροντος.

Zone: Το όνομα της ζώνης ενδιαφέροντος.

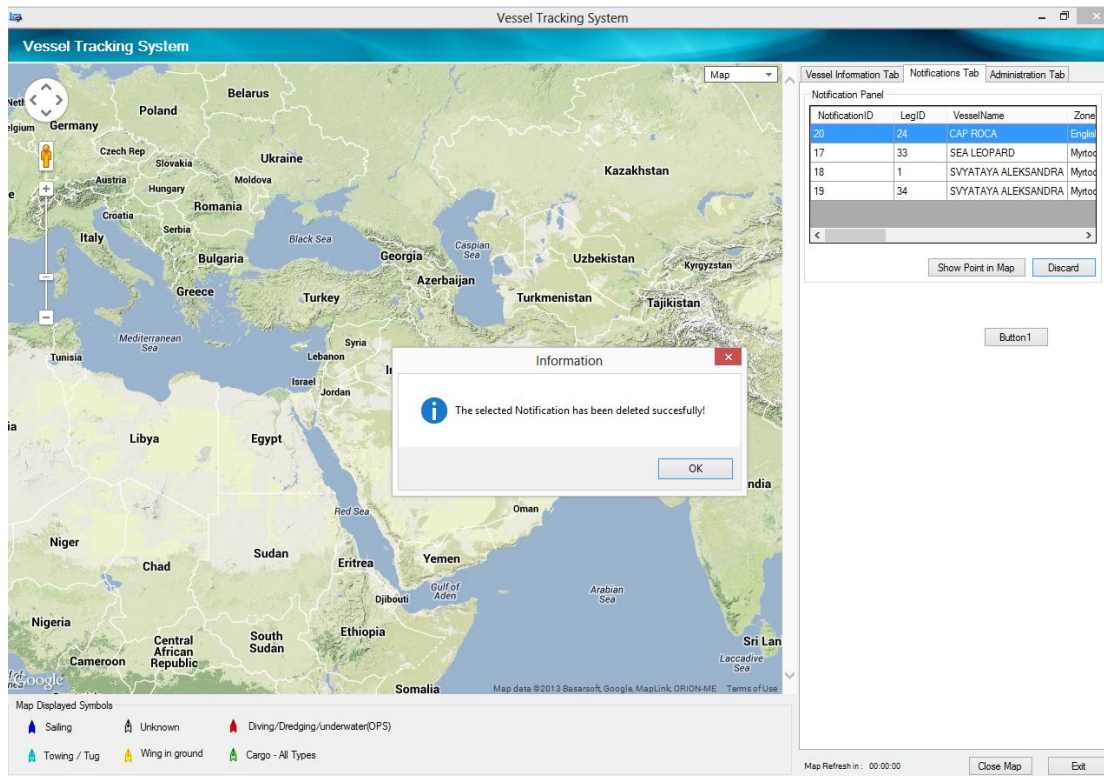
Επίσης ο χρήστης έχει τη δυνατότητα της πολλαπλής επιλογής των ειδοποιήσεων σχετικά με την είσοδο των πλοίων στις ζώνες ενδιαφέροντος όπως φαίνεται παρακάτω

The screenshot displays the Vessel Tracking System interface. The main map shows the Aegean Sea with a red-shaded zone. A tooltip for the vessel 'SVYATAYA ALEKSANDRA' is visible, providing its coordinates and destination. The notification panel on the right lists several vessels and their entry points into the zone.

NotificationID	LegID	VesselName	Zone
20	24	CAP ROCA	Englis
17	33	SEA LEOPARD	Myrtos
18	1	SVYATAYA ALEKSANDRA	Myrtos
19	34	SVYATAYA ALEKSANDRA	Myrtos

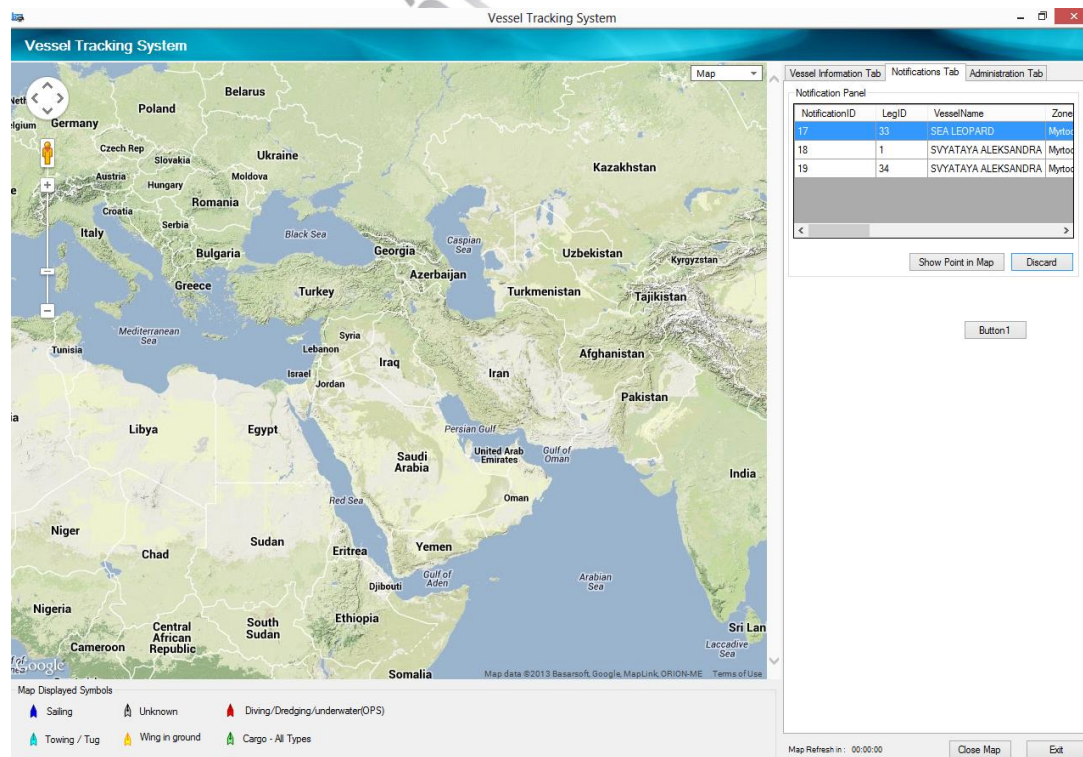
Εικόνα 9-20: Πολλαπλή επιλογή των ειδοποιήσεων και εμφάνιση των σημείων εισόδου όλων των πλοίων

Τέλος επιλέγοντας το 'Discard' ο χρήστης έχει την δυνατότητα να διαγράψει μία ή περισσότερες ειδοποιήσεις όπως φαίνεται παρακάτω.



Εικόνα 9-21: Μήνυμα διαγραφής ειδοποίησης

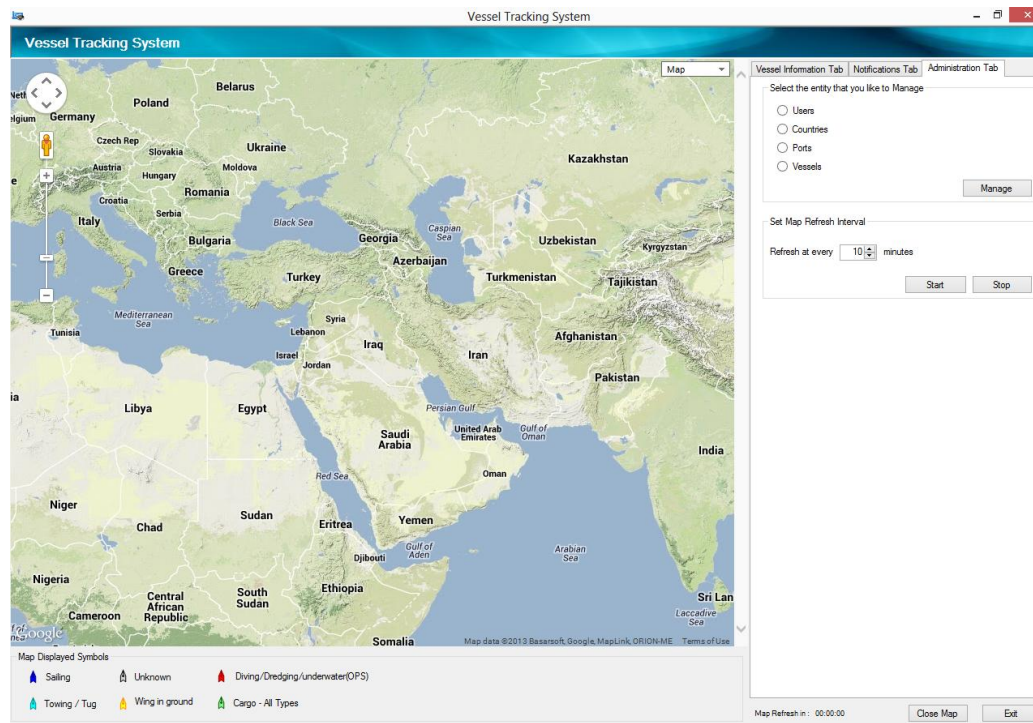
Εμφανίζεται το μήνυμα επιβεβαίωσης διαγραφής και στη συνέχεια η ειδοποίηση αφαιρείται από τη λίστα και δεν είναι πλέον εμφανής στο χρήστη.



Εικόνα 9-22: Διαγραφή της ειδοποίησης από το Notification Tab

9.4 Καρτέλα Διαχείρισης (Vessel Administrator Oriented Tab)

Αυτή η καρτέλα είναι διαθέσιμη μόνο στον διαχειριστή (administrator) του συστήματος. Μέσω αυτής ο διαχειριστής μπορεί να εισάγει στο σύστημα δεδομένα σχετικά με τους χρήστες, τις χώρες, τα λιμάνια και τα πλοία. Επίσης μπορεί να διαχειριστεί και το χρόνο ανανέωσης του χάρτη με τα νέα δεδομένα που λαμβάνονται κάθε φορά στη βάση δεδομένων.

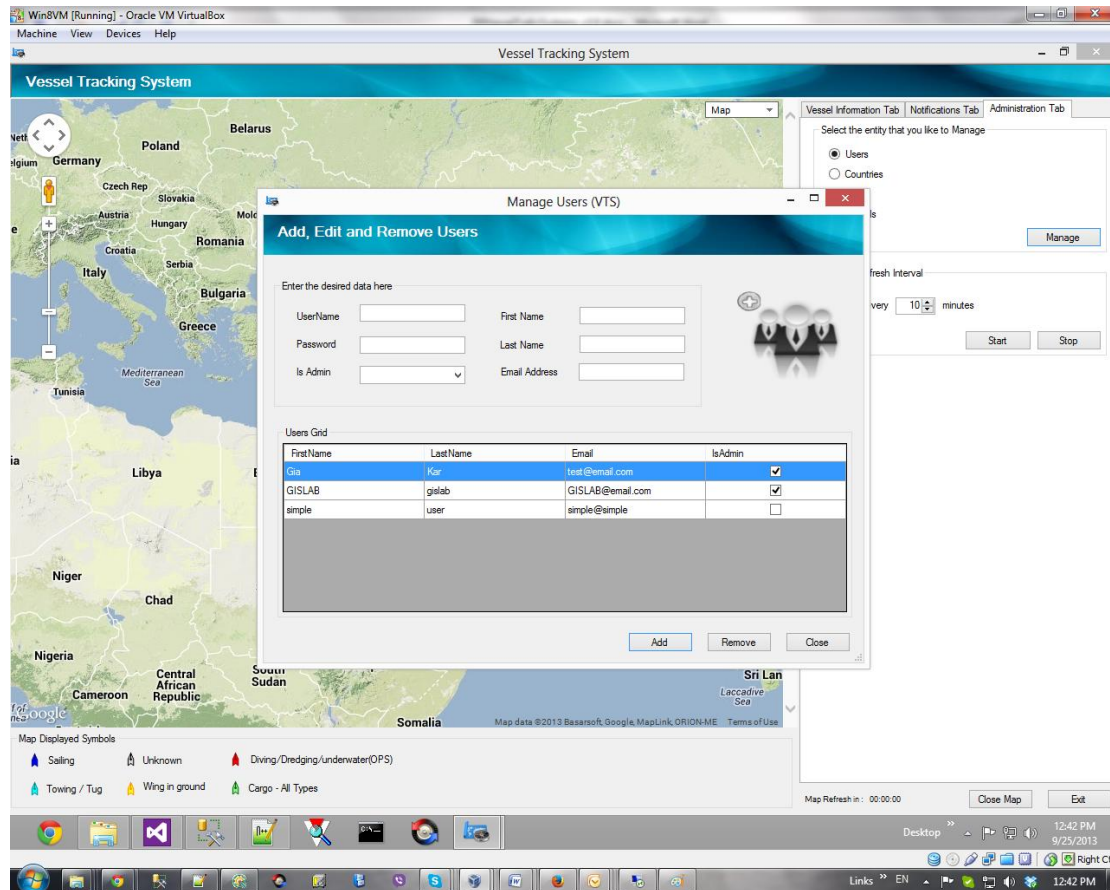


Εικόνα 9-23: Administrator Tab

Στο πλαίσιο ομάδας “Select the entity that you like to Manage” ο διαχειριστής μπορεί να επιλέξει την οντότητα (entity) την οποία θέλει να διαχειριστεί. Στη συνέχεια επιλέγοντας ‘Manage’ ανοίγει το αντίστοιχο παράθυρο για να προχωρήσει στις απαραίτητες αλλαγές.

9.4.1 Διαχείριση χρηστών (Manage Users)

Ο διαχειριστής (administrator) μπορεί να εισάγει ή και να διαγράψει δεδομένα σχετικά με τους χρήστες του συστήματος καθώς και να καθορίσει το δικαίωμα του αν κάποιος χρήστης θα είναι και διαχειριστής ή όχι.

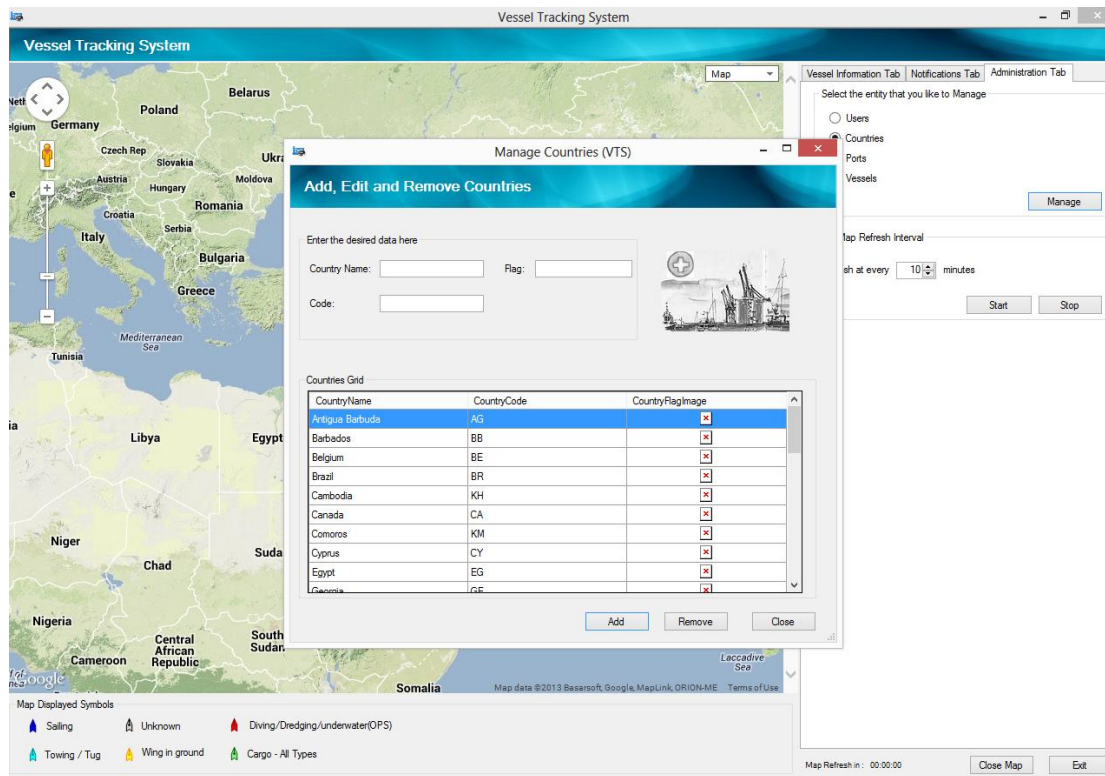


Εικόνα 9-24: Καρτέλα διαχείρισης χρηστών (Manage Users)

UserName: Το όνομα χρήστη που επιθυμεί ο χρήστης να χρησιμοποιεί για να εισέλθει στην εφαρμογή.
Password: Εισαγωγή κωδικού χρήστη που θα να χρησιμοποιεί ο χρήστης για να εισέλθει στην εφαρμογή.
Is Admin: Ορίζεται αν ο χρήστης θα είναι διαχειριστής ή όχι επιλέγοντας Yes/No.
First Name: Το μικρό όνομα του χρήστη.
Last Name: Το επίθετο του χρήστη.
Email Address: Το ηλεκτρονικό ταχυδρομείο του χρήστη. (Μελλοντική χρήση)

9.4.2 Διαχείριση Χωρών (Manage Countries)

Ο διαχειριστής (administrator) μπορεί να εισάγει ή και να διαγράψει δεδομένα σχετικά με τις χώρες που βρίσκονται στο σύστημα.



Εικόνα 9-25: Καρτέλα διαχείρισης Χωρών (Manage Countries)

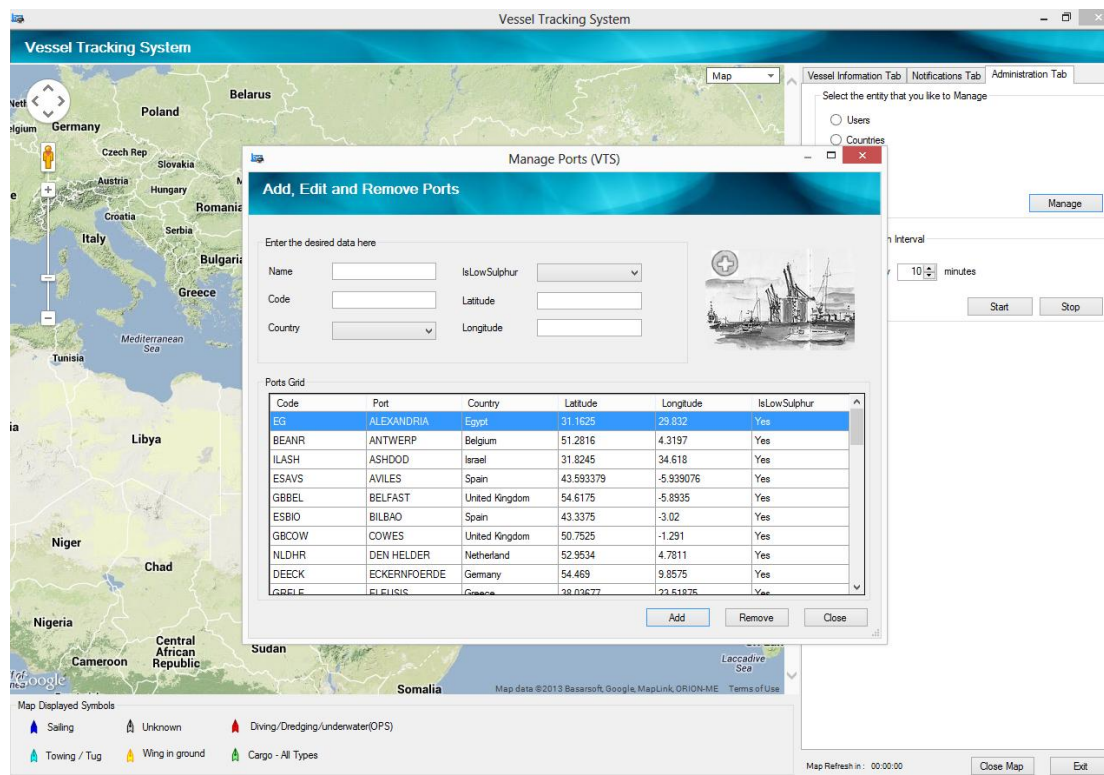
Country Name: Το όνομα της χώρας.

Code: Το κωδικό όνομα της χώρας σύμφωνα με τα διεθνή πρότυπα.

Flag: Εισαγωγή σημαίας της χώρας. (Μελλοντική χρήση)

9.4.3 Διαχείριση Λιμανιών (Manage Ports)

Ο διαχειριστής μπορεί να εισάγει ή και να διαγράψει δεδομένα σχετικά με τα λιμάνια προορισμού των πλοίων του. Επίσης μπορεί να δει και τη λίστα των λιμανιών με όλες τις πληροφορίες που έχει εισάγει για αυτά.



Εικόνα 9-26: Καρτέλα διαχείρισης λιμανιών (Manage Ports)

Name: Εισαγωγή ονόματος λιμανιού.

Code: Εισαγωγή κωδικού λιμανιού σύμφωνα με τα διεθνή πρότυπα.

Country: Επιλογή χώρας που βρίσκεται το λιμάνι από τη λίστα των χωρών που έχει εισαχθεί παραπάνω.

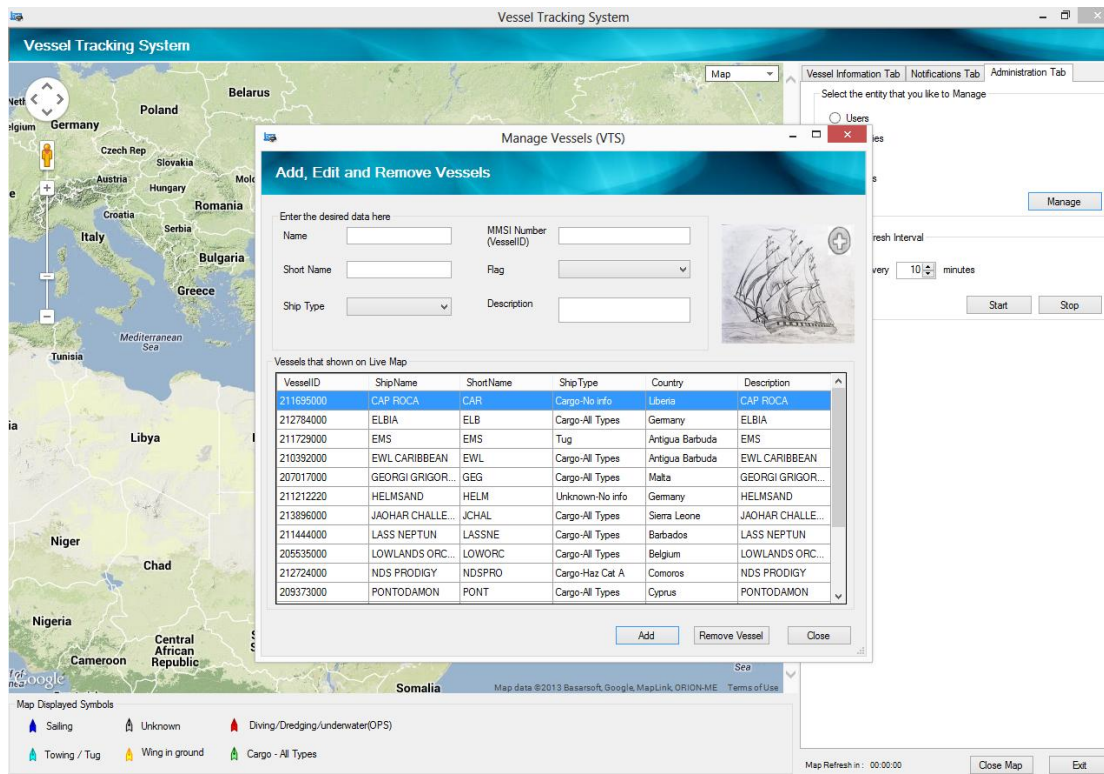
IsLowSulphur: Επιλογή του εάν το λιμάνι επιβάλλει καύσιμα χαμηλής περιεκτικότητας σε θείο (Μελλοντική χρήση).

Latitude: Εισαγωγή γεωγραφικού μήκος λιμανιού σύμφωνα με τα διεθνή πρότυπα.

Longitude: Εισαγωγή γεωγραφικού πλάτους λιμανιού σύμφωνα με τα διεθνή πρότυπα.

9.4.4 Διαχείριση Πλοίων (Manage Vessels)

Ο διαχειριστής μπορεί να εισάγει ή και να διαγράψει δεδομένα σχετικά με τα πλοία. Επίσης μπορεί να δει και τη λίστα των πλοίων με όλες τις πληροφορίες που έχει εισάγει για αυτά.



Εικόνα 9-27: Καρτέλα διαχείρισης πλοίων (Manage Vessels)

Name: Το επίσημο όνομα του πλοίου.

Short Name: Η επίσημη συντομογραφία του πλοίου.

Ship Type: Επιλογή του τύπου του πλοίου ανάλογα με την χρήση του (Tanker, LNG, Cargo etc).

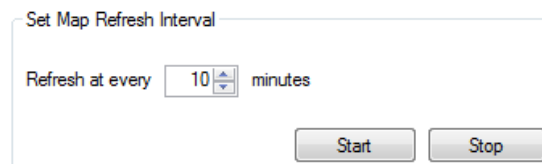
MMSI Number (Vessel ID): Ο επίσημος μοναδικός κωδικός του πλοίου σύμφωνα με τον οποίο ταυτοποιείται.

Flag: Επιλογή εθνικότητας υπό τη οποία έχει καταχωρηθεί το πλοίο ώστε να καθορίζεται η νομική δικαιοδοσία βάσης της οποίας λειτουργεί.

Description: Εισαγωγή περιγραφής.

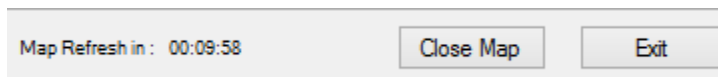
9.4.5 Διάστημα χρόνου ανανέωση χάρτη (Map Refresh Interval)

Επίσης ο διαχειριστής μπορεί να ορίσει το χρονικό διάστημα ανανέωσης του χάρτη από το 'Set Map Refresh Interval'. Ορίζοντας το διάστημα του χρόνου (σε λεπτά) που επιθυμεί να γίνεται η ανανέωση του χάρτη και στη συνέχεια επιλέγοντας 'Start' ο χρόνος για την ανανέωση μετράει αντίστροφα ενώ εμφανίζεται προς ενημέρωση του χρήστη και στο δεξιά κάτω μέρος της οθόνης.



Εικόνα 9-28: Διαχείριση του χρόνου ανανέωσης του χάρτη από τον administrator

Επίσης επιλέγοντας το 'Stop' ο διαχειριστής έχει τη δυνατότητα να απενεργοποιήσει την ανανέωση του χάρτη και να την ενεργοποιήσει όποτε εκείνος επιθυμεί ακολουθώντας την ίδια διαδικασία. Τέλος όταν ο χρόνος παρέλθει και αν η διαδικασία ανανέωσης δεν έχει απενεργοποιηθεί, ο χρόνος ξανά-ξεκινάει να μετράει αντίστροφα για την επόμενη ανανέωση.



Εικόνα 9-29: Εμφάνιση χρόνου που απομένει για ανανέωση του χάρτη

10. Μελλοντικές επεκτάσεις

Στη ενότητα αυτή θα αναφερθούμε σε διάφορες πιθανές μελλοντικές επεκτάσεις που θα μπορούσαν να υλοποιηθούν στην παρούσα εφαρμογή.

10.1 Αναζήτηση με βάση το Όνομα/MMSI Number του πλοίου

Δυνατότητα του χρήστη για αναζήτηση με βάση το όνομα ενός πλοίου ή με βάση τον MMSI αριθμό. Το συγκεκριμένο χαρακτηριστικό είναι αρκετά χρήσιμο για τον χρήστη ώστε να είναι σε θέση μόνο και μόνο με την γνώση του ονόματος του πλοίου ή του MMSI του πλοίου άμεσα να δει το στίγμα του ή τη τρέχουσα διαδρομή του. Επίσης το συγκεκριμένο και τεχνικά δεν έχει ιδιαίτερη δυσκολία στο να υλοποιηθεί. Αρκεί να γίνεται μια αναζήτηση στη βάση σύμφωνα με το όνομα του πλοίου (κατά πάσα πιθανότητα χρησιμοποιώντας την εντολή Like του SQL) ή το MMSI του ώστε να εμφανίζει ο χρήστης σε μια λίστα τα αποτελέσματα και από εκεί ο χρήστης να επιλέγει το πλοίο που τον ενδιαφέρει ώστε να δει άμεσα την σχετική πληροφορία στον χάρτη.

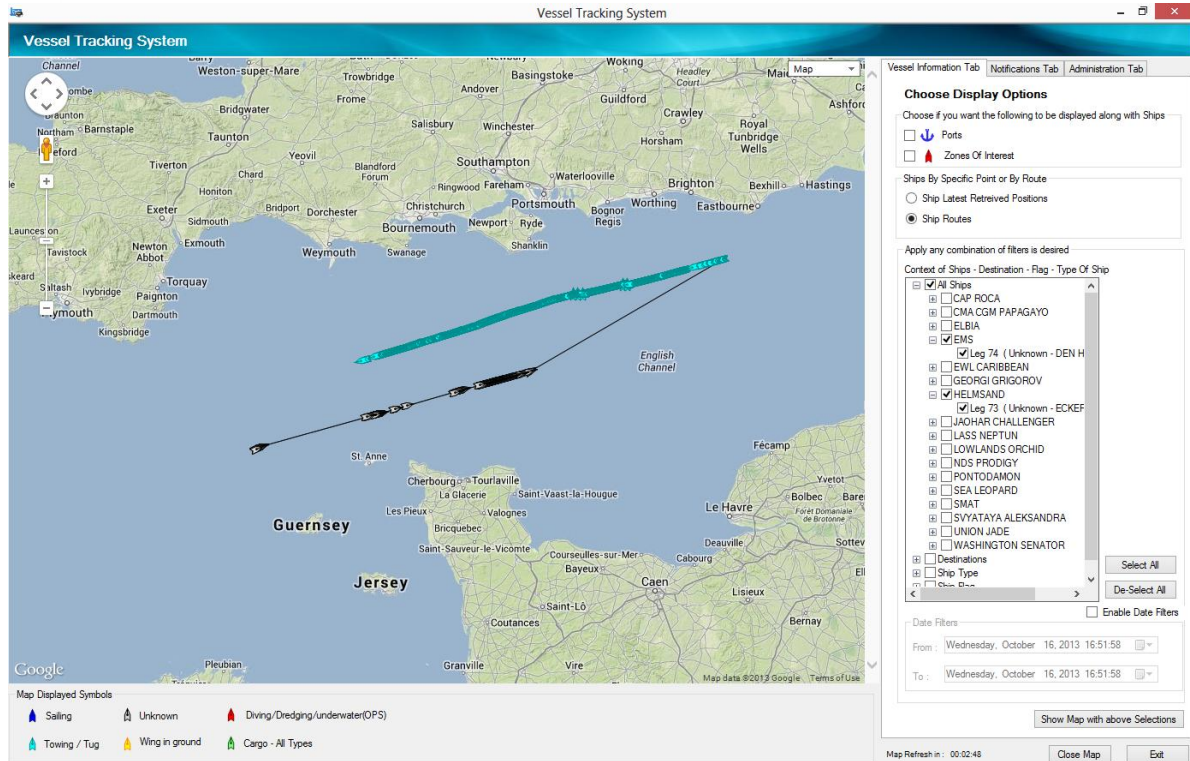
10.2 Μηχανισμός αποστολής ειδοποιήσεων

Ένα άλλο χαρακτηριστικό που θα μπορούσε να προστεθεί είναι ένας μηχανισμός αποστολής ειδοποιήσεων email ή και sms (Short Message Service) για να ενημερώνεται ο χρήστης άμεσα στη περίπτωση που ένα πλοίο εισέλθει σε μια περιοχή ενδιαφέροντος.

Υπάρχουν περιπτώσεις που κάποιος συγκεκριμένος ρόλος χρήστη μιας ναυτιλιακής εταιρίας θα πρέπει να ειδοποιηθεί άμεσα για τέτοιου είδους συμβάντα επομένως στη περίπτωση αυτή θα μπορούσε αυτή η ανάγκη να εξυπηρετηθεί. Το συγκεκριμένο χαρακτηριστικό ειδικά ως email service μπορεί να υλοποιηθεί αρκετά εύκολα αφού έχει προβλεφθεί ήδη στη βάση δεδομένων και συγκεκριμένα στον πίνακα tblusers να εισαχθεί ένα πεδίο με το email account του κάθε χρήστη που εισάγεται στο σύστημα. Επίσης η ειδοποιήσεις σαν οντότητα ήδη υπάρχουν στο σύστημα με τη διαφορά ότι εμφανίζονται στον χρήστη σε συγκεκριμένο σημείο του User Interface. Επομένως αρκεί να οριστεί ένας SMTP Client με συγκεκριμένο email host, πόρτα (port) και διαπιστευτήρια (credentials), μέσω του κώδικα της εφαρμογής ώστε η συγκεκριμένη υπηρεσία να είναι διαθέσιμη στο μέλλον.

10.3 Διόρθωση υπαρχόντων λειτουργικών προβλημάτων

Θα μπορούσε επίσης να διορθωθεί το λειτουργικό πρόβλημα που υπάρχει αναφορικά με τη δυνατότητα του χρήστη να επιλέξει πολλαπλές διαδρομές πλοίων. Στη περίπτωση αυτή όταν ο χρήστης επιλέξει να δει περισσότερα από ένα πλοία συναντάμε το σενάριο που περιγράφεται στην παρακάτω εικόνα.



Εικόνα 10-1: Λειτουργικό πρόβλημα στη περίπτωση εμφάνισης πολλαπλής διαδρομής περισσότερων του ενός πλοίων

Παρατηρούμε ότι ενώ ο χρήστης έχει επιλέξει να δει τις διαδρομές δύο πλοίων (και μάλιστα συγκεκριμένα κομμάτια διαδρομών - legs) η εφαρμογή δείχνει σωστά τις δύο διαδρομές των πλοίων αλλά το πρόβλημα που υπάρχει είναι ότι τις ενώνει κιάλας. Δηλαδή αντί η γραμμή που δηλώνει την διαδρομή του πλοίου να ενώσει τα εικονίδια της διαδρομής του πρώτου πλοίου και στη συνέχεια να σταματήσει ώστε να ενώσει μόνο τα εικονίδια του δεύτερου επιλεγμένου πλοίου, συνεχίζει και επομένως ενώνει τις δύο διαδρομές των πλοίων και στον χάρτη εμφανίζεται ως μία με το παραπάνω αποτέλεσμα.

10.4 Βελτίωση των επιδόσεων (performance) της εφαρμογής

Θα πρέπει να βρεθούν τρόποι να βελτιωθεί και το performance της εφαρμογής ώστε να μπορεί ο χρήστης να επιλέγει μεγάλο αριθμό πλοίων και να εμφανίζει τις διαδρομές τους χωρίς να υπάρχουν καθυστερήσεις στην απεικόνιση των δεδομένων.

Σε παρόμοια συστήματα, στην περίπτωση της απεικόνισης μέρους της διαδρομής του πλοίου χρησιμοποιούνται ως markers απλά σχεδιασμένα εικονίδια χαμηλής ποιότητας τα οποία εμφανίζονται κατά μήκος της γραμμής που απεικονίζει την διαδρομή του πλοίου και ανά συγκεκριμένα διαστήματα. Στη περίπτωση αυτή το μόνο εικονίδιο marker που εμφανίζεται είναι το τελευταίο στίγμα του πλοίου. Έτσι δεν χρειάζεται μεγάλη επεξεργαστική ισχύ για να εμφανίσει η εφαρμογή στον χάρτη κάποιον αριθμό χαμηλής ποιότητας εικονιδίων (που απεικονίζουν την πορεία του πλοίου) και ενός marker (που απεικονίζει το τελευταίο στίγμα του πλοίου).

Αντίθετα στη συγκεκριμένη εφαρμογή εμφανίζονται τα υψηλής ποιότητας εικονίδια των markers ανά συγκεκριμένη θέση κατά μήκος της γραμμικής απεικόνισης της διαδρομής του πλοίου κατά την οποία υπάρχει διαθέσιμο στίγμα του πλοίου. Έτσι για κάθε διαδρομή ενός πλοίου, φορτώνονται πολλαπλοί markers με υψηλής ποιότητας εικονίδια τα οποία επηρεάζουν σε πάρα πολύ μεγάλο βαθμό τις επιδόσεις της εφαρμογής.

Πανεπιστήμιο Πειραιώς

11. Βιβλιογραφία

1. Σημειώσεις μαθήματος Πανεπιστήμιο Πειραιώς, Τμήμα πληροφορικής, Γεωγραφικά πληροφοριακά συστήματα (GIS), Γιάννης Θεοδωρίδης, Νίκος Πελέκης, Ομάδα Διαχείρισης Δεδομένων, Εργαστήριο Πληροφοριακών συστημάτων.
2. Παρουσίαση μαθήματος GIS, May 2011, InfoLab mobility data collection
3. M. Vondas, N.Pelekis, Y.Theodoridis, C.Ray, V.Karkaletsis, S.Petridis, A.Miliou, 2012, Big AIS Data Proccessing for Environmentally Safe Shipping?
4. C. Claramunt, T.Devogele, S.Fourier, V.Noyon, M.Petit, C.Ray, 2007, Maritime GIS: From Monitoring to Simulation Systems
5. Alexandru Boicea, 2004, Spatial Database for Geographical Information System (GIS) Applications
6. N. Pelekis, E.Frentzos, N.Giatrakos, Y.Theodoridis, 2008, HERMES: Aggregative LBS via a Trajectory DB Engine
7. M. Perez, R. Chang, R. Billings, 2009, Automatic Identification systems (AIS) Data Use in Marine Vessel Emission Estimation
8. V. Popovich, C. Claramunt, V. Osipov, C. Ray, T. Wang, D. Berbenev, 2009, Integration of Vessel Traffic Control System and Geographical Information System
9. AA. Alesheikh, H. Helali, HA. Behroz, 2002, Web GIS: Technologies and Its Applications
10. Ιστοσελίδα τεκμηρίωσης των AIS (Automatic Identification Systems) συστημάτων, http://en.wikipedia.org/wiki/Automatic_Identification_System
11. Ιστοσελίδα τεκμηρίωσης των GIS (Geographical Information systems) συστημάτων, http://en.wikipedia.org/wiki/Geographic_information_system
12. Ιστοσελίδα τεκμηρίωσης της γλώσσας προγραμματισμού JavaScript, <http://en.wikipedia.org/wiki/JavaScript>
13. Επίσημη ιστοσελίδα της Google που παρέχει οδηγίες για το Google Maps Javascript API v3, <https://developers.google.com/maps/documentation/javascript/reference>
14. Επίσημη ιστοσελίδα της Microsoft που παρέχει οδηγίες για τις βιβλιοθήκες της γλώσσας Visual Basic .Net, <http://msdn.microsoft.com/en-us/library/be9zzz62.aspx?cs-save-lang=1&cs-lang=vb#code-snippet-3>
15. Ιστοσελίδα τεκμηρίωσης των χωρικών βάσεων δεδομένων, http://en.wikipedia.org/wiki/Spatial_database
16. Ιστοσελίδα που παρέχει πληροφορίες σχετικά με εντολές και συναρτήσεις γεώ-χωρικών δεδομένων του SQL Server, <http://weblogs.asp.net/jhallal/archive/2011/12/27/geospatial-data-in-sql-server-2008.aspx>
17. Ιστοσελίδα που παρέχει πληροφορίες σχετικά με χωρικά δεδομένα του SQL Server, <http://www.jasonfollas.com/blog/archive/2008/03/14/sql-server-2008-spatial-data-part-1.aspx>
18. Ιστοσελίδα που παρέχει πληροφορίες σχετικά με γεωγραφικά και γεωμετρικά δεδομένα του SQL Server, <http://gcfrench.wikidot.com/sql-2008>

19. Ιστοσελίδα που παρέχει πληροφορίες σχετικά με εντολές χωρικών δεδομένων του SQL Server, <http://www.west-wind.com/weblog/posts/2012/Jun/21/Basic-Spatial-Data-with-SQL-Server-and-Entity-Framework-50>
20. Ιστοσελίδα που περιέχει πληροφορίες σχετικά με analytic functions του SQL Server, <http://technet.microsoft.com/en-us/library/hh213234.aspx>
21. Ιστοσελίδα που παρέχει πληροφορίες σχετικά με την γλώσσα προγραμματισμού JSON, http://www.tutorialspoint.com/json/json_tutorial.pdf
22. Ιστοσελίδα που περιέχει πληροφορίες σχετικά με τη γλώσσα προγραμματισμού JavaScript, <http://www.javascripter.net/faq/index.htm>
23. Επίσημη ιστοσελίδα του Microsoft SQL Server σχετικά με ερωτήματα για τέμνοντα πολύγωνα, <http://social.msdn.microsoft.com/Forums/sqlserver/en-US/0082b524-218c-488b-8c9f-84f50bd548f9/does-any-points-intersect-a-polygon?forum=sqlspatial>
24. Ιστοσελίδα που παρέχει πληροφορίες σχετικά με AIS δεδομένα , http://gpsd.berlios.de/AIVDM.html#_introduction
25. Ιστοσελίδα που παρέχει πληροφορίες σχετικά με το HERMES και τα δεδομένα του, <https://hermes-mod.java.net/>
26. Ιστοσελίδα της διαδικτυακής εφαρμογής MarineTraffic, <http://www.marinetraffic.com/ais/>
27. Ιστοσελίδα της διαδικτυακής εφαρμογής VesselFinde, <http://www.vesselfinder.com/>
28. Ιστοσελίδα της εφαρμογής VT Explorer, <http://www.vtexplorer.com/vessel-tracking-software-features.html>