



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

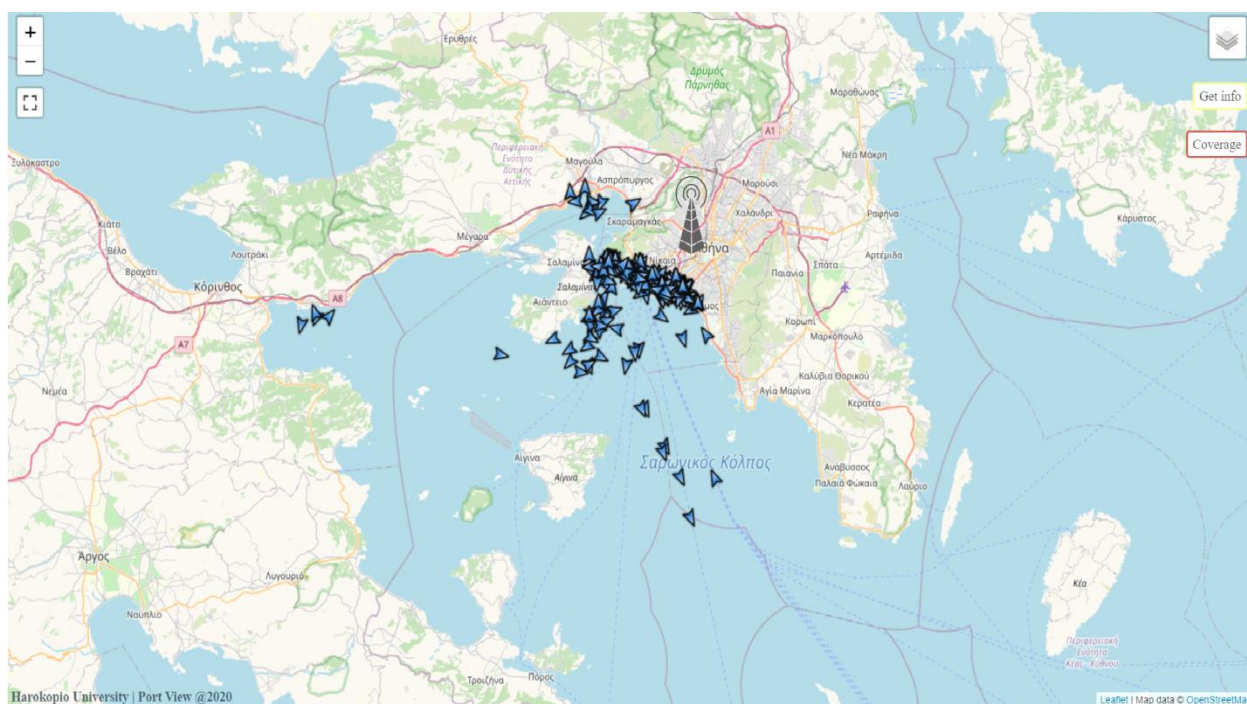
ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ και ΤΗΛΕΜΑΤΙΚΗΣ

**Τίτλος Εργασίας: Σύστημα ζωντανής παρακολούθησης ναυτιλιακής κίνησης
από επίγειο δέκτη**

Πτυχιακή εργασία

Όνομα φοιτήτριας: Ελισάβετ Δάφνη Πανταζή



Αθήνα 2020



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ και ΤΗΛΕΜΑΤΙΚΗΣ

Τριμελής Εξεταστική Επιτροπή

Κωνσταντίνος Τσερπές (Επιβλέπων καθηγητής)

**Επίκουρος Καθηγητής, Τμήμα Πληροφορικής και Τηλεματικής,
Χαροκόπειο Πανεπιστήμιο**

Ηρακλής Βαρλάμης

**Επίκουρος Καθηγητής, Τμήμα Πληροφορικής και Τηλεματικής,
Χαροκόπειο Πανεπιστήμιο**

Χρήστος Μιχαλακέλης

**Επίκουρος Καθηγητής, Τμήμα Πληροφορικής και Τηλεματικής,
Χαροκόπειο Πανεπιστήμιο**

Η Ελισάβετ Δάφνη Πανταζή,

δηλώνω υπεύθυνα ότι:

- 1) Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλει τα πνευματικά δικαιώματα τρίτων.
- 2) Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω πολύ όλους όσους συνεργάστηκαν μαζί μου για να πραγματοποιηθεί αυτή η εργασία, όσους βρισκόντουσαν δίπλα μου σε όλο αυτό το απαιτητικό διάστημα για εμένα και όσους με βοήθησαν να ολοκληρώσω και να πετύχω τον στόχο μου ακόμα και χωρίς να το γνωρίζουν.

Πίνακας περιεχομένων

Περίληψη	7
Abstract	8
Κατάλογος Εικόνων	9
Κατάλογος Πινάκων	10
Κατάλογος Εικόνων με κώδικα	10
Συντομογραφίες	11
Εισαγωγή	13
1. Αυτόματο Σύστημα Αναγνώρισεως (AIS)	15
1.1. Γενική εικόνα	15
1.2. Εφαρμογές του AIS	15
1.3. Λειτουργία του AIS	17
1.4. Περιεχόμενα ενός AIS μηνύματος	19
2. State of the Art	21
2.1. Παρόμοιες εφαρμογές στις μέρες μας	21
2.2. Web mapping	25
2.2.1. Διαδικτυακές πλατφόρμες	25
2.2.2. Εργαλεία και βιβλιοθήκες για τους χάρτες	26
2.3 Εργαλεία που χρησιμοποιήθηκαν στην εργασία	28
2.3.1. Κύρια γλώσσα προγραμματισμού	28
2.3.2. Βιβλιοθήκες	28
2.3.3. Συστατικά στοιχεία της εφαρμογής	29
3. Διατύπωση και Σχηματισμός εφαρμογής	30
3.1 Θέμα πτυχιακής	30
3.2 Συλλογή δεδομένων	31
3.2.1. Ορισμός API	31
3.2.2. Μορφή δεδομένων	31
4. Ανάλυση Εφαρμογής	32
4.1. Περιγραφή και Λειτουργία Dataset	32
4.1.1 Χρήση API στην εφαρμογή	32

4.1.2 Συνδέσεις με το API	32
4.1.3. Δείγμα Θαλάσσιων δεδομένων	33
4.2. Βάση εφαρμογής	34
4.2.1. SVG	35
4.3. Λειτουργίες εφαρμογής	36
4.3.1. Προβολή θέσης καραβιών σε πραγματικό χρόνο	36
4.3.2. Προβολή εξατομικευμένων πληροφοριών καραβιού	39
4.3.3. Προβολή θαλάσσιας διαδρομής επιλεγμένου καραβιού	43
4.3.4. Προβολή κάλυψης της εμβέλειας της κεραίας	45
5. Συμπεράσματα	49
6. Github	50
Βιβλιογραφία	50

Περίληψη

Κατά την διάρκεια της δεκαετίας του 1990 αναπτύχθηκε το Αυτόματο Σύστημα Αναγνώρισεως (Automatic Identification System - AIS) ένα σύστημα παρακολούθησης σκαφών που μεταδίδει τις ακριβείς θέσεις τους. Ακολούθησε συνεχής εξέλιξη, βελτιστοποίηση, διάδοση και η υποχρεωτική, επί το πλείστον, εφαρμογή του λόγω ναυτικών ρυθμίσεων. Σήμερα πια σχεδόν όλα τα πλοία είναι μέρος αυτού του συστήματος. Αυτό σημαίνει ότι υπάρχει θαλάσσια επικοινωνία μεταξύ των πλοίων αλλά και της ακτής, μέσω ανταλλαγής μηνυμάτων. Είναι εφικτό λοιπόν, με τα κατάλληλα μέσα και την πρόσβαση σε αυτά τα μηνύματα, κάποιος να αντλήσει δεδομένα πραγματικού χρόνου για την τοποθεσία, την ταχύτητα, το αναγνωριστικό του σκάφους και όχι μόνο. Κάτι τέτοιο αποτέλεσε και θα αποτελεί την βάση για ένα εύρος εφαρμογών το οποίο μπορεί να αφορά είτε τον πιο απλό χρήστη μιας σχετικής εφαρμογής είτε εταιρείες, οργανισμούς, ακόμα και κράτη. Γενικότερα, η θαλάσσια επιτήρηση για λόγους ασφάλειας, πρόληψης, αποτελεσματικότητας, επικοινωνίας ίσως ακόμα και περιβαλλοντικούς καθίσταται καίριας σημασίας για την ναυτιλία.

Στην παρούσα εργασία αναλύεται η δημιουργία μιας διαδικτυακής εφαρμογής, πιο συγκεκριμένα ενός λεπτομερούς διαδραστικού χάρτη. Η υλοποίηση έγινε σε γλώσσα προγραμματισμού JavaScript και χρησιμοποιήθηκε συνδυασμός της βιβλιοθήκης Leaflet.js και της D3.js, σε διαφορετικά επίπεδα του χάρτη. Τα δεδομένα της θαλάσσιας κυκλοφορίας που χρησιμοποιούνται προκύπτουν από την σύνδεση της εφαρμογής με μία βάση δεδομένων η οποία αποτελείται από παραγόμενα - σε τακτά χρονικά διαστήματα - AIS μηνύματα των σκαφών που βρίσκονται εντός της εμβέλειας μιας κεραίας-δέκτη AIS. Η σύνδεση αυτή για την άντληση πληροφοριών γίνεται μέσω μιας Διεπαφής Προγραμματισμού Εφαρμογών (API), δηλαδή ενός ενδιάμεσου λογισμικού που επιτρέπει την επικοινωνία μεταξύ δύο εφαρμογών παραδίδοντας αιτήματα και μεταφέροντας τις απαντήσεις τους. Ο χάρτης προσφέρει απεικόνιση των θέσεων, παρακολούθηση σε πραγματικό χρόνο, αλλά και εξαγωγή πληροφοριών για τα χαρακτηριστικά και τον προορισμό των σκαφών που βρίσκονται μέσα σε συγκεκριμένη περιοχή εμβέλειας γύρω από τον νομό Αττικής. Η εμβέλεια αυτή μπορεί να υπολογιστεί και να προβληθεί όπως και τα δεδομένα των σκαφών πάνω στον χάρτη. Στην πλοήγηση σε αυτό το χάρτη δίνεται ακόμα η δυνατότητα - εκτός των παραπάνω - να απεικονιστεί και η μέχρι τώρα πορεία της διαδρομής κάθε πλοίου καθώς και η απόκτηση πρόσβασης σε μερικά από τα AIS μηνύματά του.

Λέξεις κλειδιά: [χάρτης, παρακολούθηση, ΑΣΑ, D3.js, πραγματικός χρόνος]

Abstract

During the 90s an Automatic Identification System (AIS) was developed whose primary task was to follow the course of sailing vessels and register their exact position. Since then, there has been a continuous evolutionary progress and improvement of the system as well as, for the most part, its mandatory enforcement due to certain naval adjustments. Nowadays, almost all ships are registered with this system. This reality provides a maritime communication between ships and the coast through exchanges of messages. It is feasible now, using the appropriate means, to draw sufficient data, in real time, about the location, the speed, the identification of a certain ship and much more than that. Such an outcome has constituted the basis for a wide range of applications, whether it concerns a simple user of a relative application or large firms, organizations and most important whole countries. In general, the naval observation for safety reasons, prevention, effectiveness, communication and even environmental causes, has now become vital for the marine world.

In this project, the process of creating a web application and particularly an interactive web map is analyzed. The implementation of the programming was carried out in JavaScript and a conflation was used of the libraries Leaflet and D3.js, on different levels of the map. The marine data used in the project emerges from the connection of the application to a database which consists of the output -in regular time spans- of AIS messages coming from the ships within range of an antenna-receiver AIS. The connection of the data extraction is achieved through an Application Programming Interface (API) that is an intermediate software that allows communication between two applications by delivering requests and transferring their responses. The map offers a depiction of the position of the vessels, monitoring, as well as access to the characteristics and destination of a ship that sail within range of a particular area round the prefecture of Attiki. The coverage of the area can be estimated and displayed along with the ships data in real time. While navigating on this map, another possibility emerges -apart from the ones mentioned above- namely to have the depiction of the ship's preceding route up to that moment, as well as access to some of its AIS messages.

Keywords: [map, monitoring, AIS, D3.js, real-time]

Κατάλογος Εικόνων

Εικ.1. Απεικόνιση εφαρμογής του AIS ως προς την αποφυγή συγκρούσεων.....σελ.16	
Εικ.2. Συνδέσεις με το AIS.....σελ.17	
Εικ.3. Εικονική αναπαράσταση λειτουργίας AIS.....σελ.19	
Εικ.4. Δείγμα της εφαρμογής Marine-Traffic.....σελ.22	
Εικ.5. Δείγμα της εφαρμογής FleetMon.....σελ.23	
Εικ.6. Δείγμα της εφαρμογής VesselFinder.....σελ.24	
Εικ.7. Δείγμα της εφαρμογής Vesseltracker.....σελ.24	
Εικ.8. Δείγμα ενός από τα JSON data.....σελ.34	
Εικ.9. Παράδειγμα βασικού χάρτη Leaflet.....σελ.35	
Εικ.10. Λειτουργία προβολής θέσης καραβιών.....σελ.39	
Εικ.11. Προβολή πληροφοριών καραβιού στον χρήστη.....σελ.42	
Εικ.12. Προβολή διαδρομής επιλεγμένου καραβιού.....σελ.45	
Εικ.13. Προβολή κάλυψης της εμβέλειας της κεραίας.....σελ.48	

Κατάλογος Πινάκων

Πίνακας 1. Συντομογραφίες.....σελ.11

Κατάλογος Εικόνων με κώδικα

Εικ. Α. URL σύνδεσης με το API.....σελ.33
Εικ. Β. URL για τα στίγματα τεσσάρων ωρών ενός πλοίου.....σελ.33
Εικ. Γ. URL για στίγματα των τεσσάρων τελευταίων ωρών.....σελ.33
Εικ. Δ. Δημιουργία χάρτη.....σελ.35
Εικ. Ε. Συνάρτηση για την παραλαβή των δεδομένων.....σελ.36
Εικ. ΣΤ. Συνάρτηση δημιουργίας εικονιδίων για κάθε καράβι.....σελ.37
Εικ. Ζ. Συνάρτηση ανανέωσης εικονιδίων στις μετακινήσεις του χάρτη.....σελ.38
Εικ. Η. Συνάρτηση ανανέωσης δεδομένων χάρτη.....σελ.39
Εικ. Θ. Συνάρτηση για mouse interaction.....σελ.40
Εικ. Ι. Δημιουργία D3 button.....σελ.41
Εικ. Κ. Δημιουργία D3 tooltip.....σελ.41
Εικ. Λ. CSS tooltip styling.....σελ.42
Εικ. Μ. Συνάρτηση για συλλογή δεδομένων ενός πλοίου.....σελ.43
Εικ. Ν. Δημιουργία εικονιδίων για όλα τα στίγματα της διαδρομής.....σελ.44
Εικ. Ξ. Δημιουργία γραμμής της διαδρομής ενός πλοίου.....σελ.44
Εικ. Ο. Δείγμα ψευδοκώδικα για την λειτουργία του convex hull.....σελ.46
Εικ. Π. Χρήση convex hull και δημιουργία D3 path.....σελ.47
Εικ. Ρ. Συνάρτηση ανανέωσης path και ορισμός προβολής σημείων.....σελ.48

Συντομογραφίες

AIS	Automatic Identification System
D3	Data-Driven-Documents
GPS	Global Positioning System
IMO	International Maritime Organisation
ETA	Estimated Time of Arrival
MMSI	Maritime Mobile Service Identity number
UTC	Coordinated Universal Time
SOG	Speed Over Ground
COG	Course Over Ground
VHF	Very High Frequency
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
SVG	Scalable Vector Graphics
VTs	Vessel Traffic Services
SOLAS	International Convention for the Safety of Life at Sea
ECDIS	Electronic Chart and Information Systems
GT	Gross Tonnage
CPA	Closest Point of Approach
MSC	Maritime Safety Committee
VDR	Voyage Data Recorder
VDL	VHF Data Link
VDES	VHF Data Exchange System

TDMA	Time Division Multiple Access
API	Application Programming Interface
KML	Keyhole Markup Language
GML	Geography Markup Language
SDK	Software Development Kit
GIS	Geographic Information System
JSON	JavaScript Object Notation
CSV	Comma-Separated-Values
DOM	Document Object Model
IP	Internet Protocol

Πίνακας 1

Εισαγωγή

Κάπου μέσα σε αυτήν την πληθώρα θαλάσσιων δεδομένων σχετικά με την θέση και την πορεία των σκαφών, ήρθε η στιγμή που η προϋπάρχουσα επιθυμία και ανάγκη για την αξιοποίηση τους έγινε πλέον δυνατότητα. Μία σημαντική εφαρμογή που προκύπτει από την αξιοποίηση της πληθώρας των δεδομένων σχετικά με τα σκάφη στη θάλασσα, είναι η θαλάσσια επιτήρηση. Κάτι που πραγματικά έγινε μέσω του AIS, το οποίο σχεδιάστηκε για να είναι ικανό να παρέχει αυτόματα πληροφορίες από το πλοίο σε άλλα κοντινά πλοία και παράκτιες αρχές. Υπάρχουν, βέβαια, περισσότερα από ένα συστήματα για θαλάσσια επιτήρηση. Τα βασικά χαρακτηριστικά του που ίσως το έκαναν να ξεχωρίσει είναι ότι είναι ανοιχτό και σε μερικές περιπτώσεις απαιτεί πιο φθηνό και απλό εξοπλισμό.

Οι πληροφορίες αυτές οργανώνονται στα δυναμικά (Dynamic) και στατικά (Static) μηνύματα. Τα Static μηνύματα σχετίζονται με τα χαρακτηριστικά του πλοίου και αποτελούνται από: International Maritime Organisation number (IMO), Call Sign, Name, Type, Dimensions, Type of positioning system, Draught, Destination και ETA (Estimated Time of Arrival). Τα Dynamic μηνύματα μεταφέρουν πληροφορίες σχετικά με την κίνηση του πλοίου και αποτελούνται από τα: Maritime Mobile Service Identity number (MMSI), AIS Navigational Status, Rate of Turn, Speed Over Ground, Position Coordinates, Course Over Ground, Heading, Bearing at own position και UTC seconds [3]. Όλα αυτά τα AIS δεδομένα καταγράφονται από πολυάριθμους επίγειους σταθμούς στα λιμάνια και από πολλά πρακτορεία παγκοσμίως και είναι γνωστά ως T-AIS data (terrestrial). Παράλληλα, ειδικά εξοπλισμένοι δορυφόροι μπορούν να ανιχνεύσουν και να καταγράψουν και αυτοί AIS μηνύματα και σε αυτήν την περίπτωση χρησιμοποιείται ο όρος S-AIS (satellite) για τα παραπάνω δεδομένα.

Τα πλοία λοιπόν, φέρουν τους κατάλληλους πομποδέκτες AIS, οι οποίοι αποτελούνται από έναν δέκτη εντοπισμού θέσης GPS και έναν πομποδέκτη VHF. Οι πομποδέκτες αυτοί μεταδίδουν τα προαναφερθέντα δεδομένα με μία περιοδικότητα η οποία εξαρτάται από το ρυθμό μεταβολής της κατάστασης του σκάφους. Οι σταθμοί βάσης, τα υπόλοιπα πλοία ή οι δορυφόροι έχουν έναν δέκτη AIS και λαμβάνουν αυτές τις πληροφορίες και κάπως έτσι εγκαθιδρύθηκε μια μορφή επικοινωνίας, η οποία είναι ανιχνεύσιμη μέχρι το διάστημα και έγινε η βάση διαφόρων και πολύπλευρων εφαρμογών και ιδεών.

Μια εφαρμογή που θα αξιοποιεί αυτήν την επικοινωνία και τα δεδομένα της, ήταν άμεσο επακόλουθο των γοργών ρυθμών ανάπτυξης της τεχνολογίας αλλά και του γεγονότος ότι μπορεί αρκετά εύκολα να τα συλλέξει κάποιος. Ένα από τα σημαντικά σημεία σε αυτό το κομμάτι είναι η σωστή και λειτουργική αναπαράσταση όλων αυτών των σημάτων σε ένα σύστημα ή στην προκειμένη περίπτωση σε έναν χάρτη, μέσα ενός υπολογιστή. Αυτό χρειάζεται να γίνει σε μια διαδικτυακή εφαρμογή, εφόσον θέλουμε να ανανεώνεται η ροή των δεδομένων και όχι να πειραματιστούμε σε κάποιο συγκεκριμένο dataset.

Η χρήση της απεικόνισης για την παρουσίαση πληροφοριών σε χάρτες, επιστημονικά σχεδιαγράμματα και όχι μόνο, χρησιμοποιείται εδώ και χιλιάδες χρόνια. Στις μέρες μας όταν μιλάμε για data visualization ο νους μας περιστρέφεται γύρω από την επιστήμη των υπολογιστών, γλώσσες προγραμματισμού, βιβλιοθήκες, εργαλεία. Η εκάστοτε συνδυαστική επιλογή όλων των παραπάνω είναι και αυτή που θα δώσει το καλύτερο δυνατό εικονικό αποτέλεσμα προς τους χρήστες. Όσον αφορά την υλοποίηση στην συγκεκριμένη περίπτωση, η εφαρμογή που αναλύεται παρακάτω γράφτηκε με την ανοιχτού κώδικα βιβλιοθήκη της JavaScript, Leaflet.js και με την επίσης ανοιχτού κώδικα βιβλιοθήκη της JavaScript D3.js (από το Data-Driven Documents). Η τελευταία ξεχωρίζει προσφέροντας δυναμικά γραφικά, αναπαραστάσεις δεδομένων, αλληλεπιδράσεις και κίνηση σχεδίων. Ενώ μαζί με όλα τα παραπάνω χρησιμοποιήθηκαν και HTML, SVG και CSS για την υλοποίηση της διαδικτυακής εφαρμογής.

Έχοντας λοιπόν τα εφόδια και τελικά μια υλοποιημένη εφαρμογή προς παρακολούθηση της κίνησης των θαλάσσιων περιοχών μένει μόνο να αναφερθούμε στην αξία της. Η σημασία της εφαρμογής ποικίλλει, ξεκινάει από τον πιο απλό χρήστη και την επιθυμία του να γνωρίζει την θέση του πλοίου με το οποίο μπορεί να ταξιδεύει εκείνος - στο παρόν ή το προσεχές μέλλον - ή κάποιο οικείο πρόσωπο. Ακόμα, μπορεί το ταξίδι να αφορά κάποιο εμπόρευμα που έχει αποστείλει ή περιμένει να λάβει. Σε μεγαλύτερη κλίμακα είναι αρκετά σημαντικό για μια ναυτιλιακή εταιρεία να έχει υπό επιτήρηση τον στόλο της, να έχει τον έλεγχο για την ασφάλεια του αλλά και τα δεδομένα για να προβεί σε τυχόν βελτιώσεις ή άμεσες επεμβάσεις όπου αυτό χρειάζεται - καθώς υπάρχουν τα ακριβή στίγματα της πορείας ενός σκάφους. Παράλληλα έχει αξία για οποιαδήποτε εταιρεία μπορεί να εκμεταλλευτεί τα παρεχόμενα δεδομένα προς κάποιο ηθικό σκοπό. Τελευταία, αλλά εξίσου σημαντική, είναι η περιβαλλοντική οπτική μιας τέτοιας εφαρμογής. Είναι αναμφισβήτητα θετικό και ελπιδοφόρο να υπάρχει η δυνατότητα από την παρακολούθηση των πλοίων να εξάγονται συμπεράσματα τυχόν αυξημένης κίνησης, ποσοστά ρύπανσης και να διαμορφώνονται και να λαμβάνονται τα κατάλληλα μέτρα για την μείωση τους. Το ίδιο και για τις θαλάσσιες προστατευόμενες περιοχές με σκοπό την διασφάλιση της προστασίας τους. Μάλιστα η Διεθνής Σύμβαση του Διεθνούς Ναυτιλιακού Οργανισμού για την Ασφάλεια της Ζωής στη Θάλασσα (SOLAS - International Convention for the Safety of Life at Sea) απαιτεί την τοποθέτηση του AIS σε διεθνή πλοία και σκάφη που ταξιδεύουν με 300 ή μεγαλύτερη ολική χωρητικότητα (GT) και σε όλα τα επιβατηγά πλοία ανεξαρτήτως μεγέθους [4].

Συνοψίζοντας το σημαντικότερο από όλα αυτά, μαζί με την προστασία του περιβάλλοντος, είναι η ασφάλεια στις θαλάσσιες διαδρομές. Σύμφωνα με το Διεθνή Ναυτιλιακό Οργανισμό, αυτός ήταν και ο αντικειμενικός σκοπός της ανάπτυξης του συστήματος AIS. Η βελτίωση του επιπέδου ασφαλείας κατά τον πλου, η δυνατότητα εκτελέσεως ασφαλέστερης και αποτελεσματικότερης ναυτιλίας, η αναγνώριση των στόχων, η υποβοήθηση της παρακολούθησης των στόχων, η απλούστευση της επικοινωνίας/ανταλλαγής πληροφοριών μεταξύ πλοίων και η παροχή επιπρόσθετης πληροφορίας για ορθή εκτίμηση του ναυτιλιακού περιβάλλοντος [1][2].

1. Αυτόματο Σύστημα Αναγνώρισεως (AIS)

1.1. Γενική εικόνα

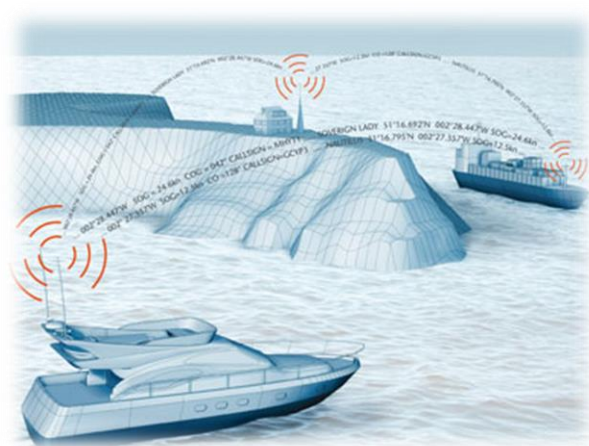
Σε μόλις περίπου δέκα χρόνια, η θαλάσσια τεχνολογία διένυσε πολύ δρόμο, συμπεριλαμβανομένων τεχνολογικών εξελίξεων, διεθνών εντολών και διεθνών προτύπων. Ιστορικά, υπήρξαν κι άλλες μορφές θαλάσσιας πλοήγησης που ήταν αποτελεσματικές στην πλοήγηση από το ένα μέρος στο άλλο. Έχοντας όμως κατά νου την ασφάλεια, χρειαζόντουσαν περαιτέρω βελτιώσεις. Η έλλειψη ταυτοποίησης του σκάφους, σε συνδυασμό με τις καθυστερήσεις στο χρόνο και την έλλειψη πληροφοριών προορισμού και ταχύτητας οδήγησαν στην ανάπτυξη του AIS και στην αύξηση της ασφάλειας, ιδίως σε πολυσύχναστες πλωτές οδούς. Το AIS, σε σύγκριση με άλλες τεχνολογίες θαλάσσιας πλοήγησης, είναι μια σχετικά νέα καινοτομία. Το πλεονέκτημα του έναντι άλλων τεχνολογιών είναι ότι μεταδίδει δεδομένα με οργανωμένο και διαλειτουργικό τρόπο. Συγχωνεύει στατικά και δυναμικά δεδομένα μαζί.

Το Αυτόματο Σύστημα Ταυτοποίησης (AIS) είναι ένα ναυτικό τεχνικό πρότυπο που αναπτύχθηκε από τον Διεθνή Ναυτιλιακό Οργανισμό (International Maritime Organization - IMO). Λειτουργεί ως ένα σύστημα αυτόματης παρακολούθησης και ανταλλαγής πληροφοριών που χρησιμοποιεί αναμεταδότες σε πλοία και χρησιμοποιείται από υπηρεσίες κυκλοφορίας πλοίων (VTS-Vessel Traffic Services). Αυτό το επιτυγχάνει μέσω ενός τυποποιημένου πομποδέκτη VHF και ενός ενσωματωμένου GPS μαζί με άλλους ηλεκτρονικούς αισθητήρες πλοήγησης (π.χ. μια γυροσυσκευή ή ένα δείκτη ταχύτητας στροφής). Ενώ, οι πληροφορίες που παρέχονται από τον εξοπλισμό αυτό (μοναδική ταυτοποίηση, θέση, ταχύτητα, πορεία, κατεύθυνση κλπ) μπορούν να εμφανιστούν σε μια οθόνη (π.χ. ραντάρ) ή σε ηλεκτρονική οθόνη γραφήματος και σύστημα πληροφοριών (ECDIS). Κάπως έτσι μπορεί να υπάρχει ένα σύμβολο για κάθε πλοίο εντός της εκάστοτε εμβέλειας, το καθένα με ένα διάνυσμα ταχύτητας (υποδεικνύοντας ταχύτητα και κατεύθυνση), με την θέση του σε GPS ή διαφορική ακρίβεια GPS. Επιπλέον κάνοντας κλικ στο σύμβολο του πλοίου, μπορεί να μαθαίνεις το όνομα του πλοίου, την πορεία, την ταχύτητα, το MMSI και άλλες πληροφορίες, κάτι το οποίο εν τέλει παρουσιάζεται σε αυτήν την εργασία.

1.2. Εφαρμογές του AIS

Ο αρχικός σκοπός του AIS ήταν αποκλειστικά η αποφυγή σύγκρουσης στις θαλάσσιες μεταφορές - ιδίως σε σημεία που δεν είναι εντός της εμβέλειας των συστημάτων που βρίσκονται στην ακτή - και ταυτόχρονα είναι και η κύρια μέθοδος που χρησιμοποιείται για αυτόν τον σκοπό. Προσδιορίζει κάθε σκάφος ξεχωριστά, μαζί με τη συγκεκριμένη θέση και την πορεία του, επιτρέποντας τη δημιουργία μιας εικονικής ένδειξης σε πραγματικό χρόνο. Τα AIS περιλαμβάνουν μια ποικιλία αυτόματων υπολογισμών που βασίζονται σε αυτές τις αναφορές θέσης, όπως το πλησιέστερο σημείο προσέγγισης (CPA) και συναγερμούς σύγκρουσης. Καθώς όμως το AIS δεν χρησιμοποιείται από όλα τα σκάφη, συνήθως χρησιμοποιείται σε συνδυασμό με ραντάρ.

Όσον αφορά την θαλάσσια ασφάλεια, το AIS επιτρέπει στις αρχές να εντοπίζουν συγκεκριμένα σκάφη και τη δραστηριότητά τους εντός ή κοντά στην Αποκλειστική Οικονομική Ζώνη ενός έθνους (ΑΟΖ). Όταν τα δεδομένα AIS συγχωνεύονται με τα υπάρχοντα συστήματα ραντάρ, οι αρχές είναι σε θέση να κάνουν πιο εύκολη διάκριση μεταξύ των σκαφών. Αυτά τα δεδομένα μπορούν να υποβληθούν σε αυτόματη επεξεργασία για τη δημιουργία κανονικοποιημένων μοτίβων δραστηριότητας για μεμονωμένα σκάφη, τα οποία όταν παραβιαστούν, δημιουργούν μια ειδοποίηση επισημαίνοντας έτσι τις πιθανές απειλές για αποτελεσματικότερη χρήση των στοιχείων ασφαλείας. Το AIS βελτιώνει την ευαισθητοποίηση στον τομέα της ναυτιλίας και επιτρέπει αυξημένη ασφάλεια και έλεγχο.

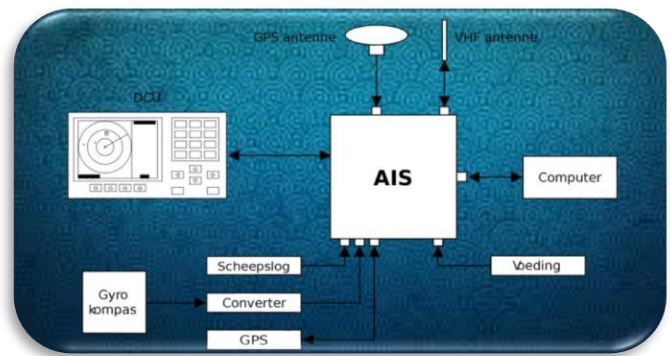


Εικ.1. Απεικόνιση εφαρμογής του AIS ως προς την αποφυγή συγκρούσεων

Παράλληλα, κατά την 79η σύνοδό του Δεκέμβριο του 2004, η Επιτροπή Ασφάλειας της Ναυσιπλοΐας (MSC) συμφώνησε ότι, σχετικά με το ζήτημα των ελεύθερων διαθέσιμων δεδομένων που μεταδίδονται από πλοία δημιουργημένα από το αυτόματο σύστημα πληροφοριών (AIS), η δημοσίευση τους στον παγκόσμιο ιστό ή αλλού θα μπορούσε να είναι επιζήμια για την ασφάλεια των πλοίων και των λιμενικών εγκαταστάσεων και υπονομεύει τις προσπάθειες του Οργανισμού και των κρατών μελών του για ενίσχυση της ασφάλειας της ναυσιπλοΐας και της ασφάλειας στον διεθνή τομέα των θαλάσσιων μεταφορών. Έτσι, η Επιτροπή καταδίκασε τέτοιου είδους δημοσιεύσεις και κάλεσε τις κυβερνήσεις των κρατών μελών, με την επιφύλαξη των διατάξεων της εθνικής τους νομοθεσίας, να αποθαρρύνουν όσους μοιράζουν δεδομένα AIS σε άλλους για δημοσίευση στο Παγκόσμιο Ιστό ή αλλού από αυτό. Επιπλέον, καταδίκασε και όσους δημοσιεύουν ανεύθυνα δεδομένα AIS, ιδίως εάν προσφέρουν υπηρεσίες στη ναυτιλία και στον λιμενικό κλάδο.

Σε περιπτώσεις έρευνας ατυχημάτων, για το συντονισμό των επιτόπιων πόρων μιας θαλάσσιας έρευνας και διάσωσης (SAR), οι πληροφορίες AIS που λαμβάνονται από VTS είναι ιδιαίτερα σημαντικές έως και επιτακτικής ανάγκης για τη διερεύνηση τους. Δεδομένου ότι παρέχουν ακριβή ιστορικά δεδομένα σχετικά με την ώρα, την ταυτότητα, τη θέση με βάση το GPS, την επικεφαλίδα πυξίδας, την πορεία πάνω από το έδαφος, την ταχύτητα και τους ρυθμούς στροφής (πληροφορίες που παρέχονται από το ραντάρ είναι λιγότερο ακριβείς), ακόμη και αν το εύρος AIS περιορίζεται στο εύρος ραδιοσυχνοτήτων VHF. Μια πληρέστερη εικόνα των γεγονότων θα μπορούσε να ληφθεί από τα δεδομένα του Voyage Data Recorder (VDR) εάν είναι διαθέσιμα και διατηρημένα επί του σκάφους για λεπτομέρειες σχετικά με την κίνηση του πλοίου, τη φωνητική επικοινωνία και τις εικόνες ραντάρ κατά τη διάρκεια των ατυχημάτων. Ωστόσο, τα δεδομένα VDR δεν διατηρούνται λόγω της περιορισμένης αποθήκευσης δώδεκα ωρών από την απαίτηση του IMO [5].

Ακόμα, σχετικά με τον έλεγχο των εθνικών αλιευτικών στόλων, το AIS χρησιμοποιείται ευρέως από τις εθνικές αρχές για τον εντοπισμό και την παρακολούθηση των δραστηριοτήτων τους. Καθώς επιτρέπει στις αρχές να παρακολουθούν αξιόπιστα και αποτελεσματικά τις δραστηριότητες των αλιευτικών σκαφών κατά μήκος της ακτογραμμής τους, συνήθως σε απόσταση 100 km, ανάλογα με την τοποθεσία και την ποιότητα των παραλιακών δεκτών/σταθμών βάσης με συμπληρωματικά δεδομένα από δορυφορικά δίκτυα. Φυσικά το ίδιο ισχύει και για στόλους εταιρειών, παρακολούθηση φορτίων και της παγκόσμιας τοποθεσίας τους ή και της προόδου τους.



Εικ.2. Συνδέσεις με το AIS

Ολοκληρώνοντας, αυτό το σύστημα μπορεί να εφαρμοστεί προς όφελος του περιβάλλοντος μέσω εκτιμήσεως της τρέχουσας κατάστασης των ωκεανών και της επιφάνειάς τους μετά από ανάλυση δεδομένων AIS. Διαφορετικά σε μια επιθυμία προστασίας υποδομών θαλάσσιου βυθού (καλώδια ή αγωγούς), με τους ιδιοκτήτες να χρησιμοποιούν πληροφορίες AIS για την παρακολούθηση των δραστηριοτήτων των σκαφών κοντά στα περιουσιακά τους στοιχεία σε σχεδόν πραγματικό χρόνο.

1.3. Λειτουργία του AIS

Το AIS είναι ένα σύστημα εκπομπής πλοίου που λειτουργεί σαν αναμεταδότης, στη θαλάσσια ζώνη VHF, το οποίο είναι ικανό να χειρίζεται πάνω από 4.500 αναφορές ανά λεπτό και να ενημερώνεται κάθε δύο δευτερόλεπτα. Χρησιμοποιεί την τεχνολογία Self-Organizing Time Division Multiple Access (SOTDMA) για να καλύψει αυτόν τον υψηλό ρυθμό μετάδοσης και να εξασφαλίσει αξιόπιστη λειτουργία αποστολής σε πλοίο.

Το εύρος κάλυψης VHF του AIS είναι παρόμοιο με άλλες εφαρμογές VHF και ουσιαστικά θα εξαρτηθεί από το ύψος που είναι τοποθετημένη η κεραία VHF και από την ισχύ μετάδοσης. Η διάδοσή του είναι ελαφρώς καλύτερη από αυτή του ραντάρ, λόγω του μεγαλύτερου μήκους κύματος, οπότε είναι πιθανό να "δεις" γύρω από τις στροφές και πίσω από τα νησιά εάν οι μάζες της γης δεν είναι πολύ υψηλές. Το AIS χρησιμοποιεί τα παγκόσμια κατανεμημένα κανάλια Marine Band 87 και 88 και συγκεκριμένα την υψηλή πλευρά του duplex από τα δύο ραδιοφωνικά κανάλια. MHz. Η χρήση δύο διαφορετικών συχνοτήτων υπερβραχέων κυμάτων γίνεται για την αποφυγή προβλημάτων παρεμβολών, καθώς επίσης και για λόγους που εξυπηρετούν την απρόσκοπτη συμμετοχή του μέγιστου δυνατού αριθμού πλοίων στο δίκτυο. Οι περισσότερες μεταδόσεις AIS αποτελούνται από ριπές πολλών μηνυμάτων. Σε αυτές τις περιπτώσεις, μεταξύ των μηνυμάτων, ο πομπός AIS πρέπει να αλλάξει κανάλι.

Κανάλι A 161.975 MHz (87B)

Κανάλι B 162.025 MHz (88B)

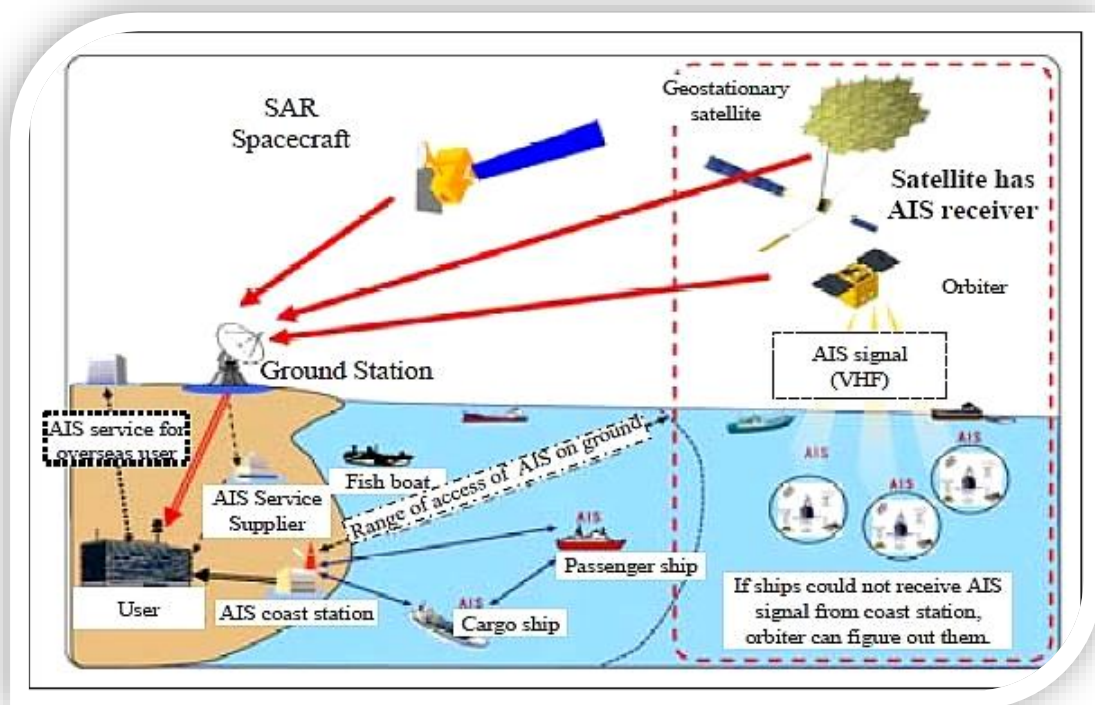
Τα απλά κανάλια 87Α και 88Α χρησιμοποιούν χαμηλότερη συχνότητα, ώστε να μην επηρεάζονται από αυτήν την κατανομή και μπορούν ακόμα να χρησιμοποιηθούν όπως έχουν καθοριστεί για το σχέδιο θαλάσσιων κινητών συχνοτήτων.

Το AIS είναι μια πολύπλοκη τεχνολογία που διέπεται από ένα αυστηρό διεθνές «πρότυπο» τεχνολογίας που ελέγχεται από το IEC. Αυτό διασφαλίζει ότι η ακεραιότητα και η λειτουργική απόδοση του συστήματος AIS και η διαλειτουργικότητα μεταξύ των συσκευών AIS είναι εγγυημένη. Στο κέντρο ενός δικτύου AIS βρίσκεται το σχήμα TDMA τεχνολογίας (πολλαπλή πρόσβαση διαίρεσης χρόνου), για την κατανομή και την κοινή χρήση των διαθέσιμων κυμάτων αέρα στη συχνότητα AIS, χρησιμοποιώντας τον λεγόμενο slot map. Τα 60 δευτερόλεπτα (= 1 καρτέ) τεμαχίζονται σε 2250 χρονικά διαστήματα (time slots) για να χωρέσουν μια αναφορά VHF από έναν σταθμό AIS. Καθώς χρησιμοποιούνται 2 κανάλια, ο συνολικός αριθμός χρονικών χρόνων είναι 4500 σε 60 δευτερόλεπτα όπως αναφέρθηκε στην αρχή. Σε αντίθεση με το όνομα δεν υπάρχει ένας χάρτης slot και τίποτα σαν κεντρικός χάρτης slot, αλλά ουσιαστικά όλοι οι συμμετέχοντες σταθμοί AIS θα αποθηκεύουν και θα ενημερώνουν τον δικό τους slot χάρτη με βάση τα δεδομένα AIS που λαμβάνονται και τους υπολογισμούς τους. Για τον βέλτιστο συγχρονισμό με τον "slot map", είναι επίσης σημαντικό το AIS να διαθέτει αξιόπιστες πληροφορίες GPS καθώς η ακριβής ώρα προέρχεται από εκεί.

Κάθε time slot επιτρέπει ένα συγκεκριμένο χώρο για το μέγεθος της μετάδοσης AIS, ο οποίος με τη σειρά του περιορίζεται από την ποσότητα πληροφοριών που μεταφέρει ένα μόνο μήνυμα AIS. Το σύστημα AIS έχει σχεδιαστεί για να προσφέρει σε τοπικά, περιφερειακά και εθνικά συστήματα, ως εκ τούτου το σύστημα AIS προσαρμόζεται έξυπνα στον αριθμό των σκαφών εντός μιας δεδομένης περιοχής. Εάν ο "slot map" πρόκειται να υπερφορτωθεί, δίνεται προτεραιότητα στα slot των πλησιέστερων σκαφών, διασφαλίζοντας ότι θα παρατηρηθούν πρώτα. Στην πράξη αυτό σημαίνει ότι ανεξάρτητα από το πόσα σκάφη παρακολουθούνται, ένα σωστά διαμορφωμένο σύστημα AIS θα διασφαλίσει ότι όλα τα σκάφη είναι ορατά.

Η χωρητικότητα του συστήματος θεωρήθηκε σχεδόν απεριόριστη, επιτρέποντας μεγάλο αριθμό πλοίων να εξυπηρετηθούν την ίδια ώρα. Ωστόσο, τα τελευταία χρόνια με αυξημένη κίνηση, ολοένα και περισσότερες συσκευές και εύρος ζώνης με ένα σημαντικό φορτίο στη χωρητικότητα του VHF Data Link (VDL) - σχεδόν εξαντλημένο από νέες εφαρμογές - τα κανάλια AIS έχουν υπερφορτωθεί σε πολυσύχναστα λιμάνια και διαδρομές.

Έτσι, το σύστημα επόμενης γενιάς AIS που ονομάζεται διεθνώς σύστημα ανταλλαγής δεδομένων VHF (VDES - VHF Data Exchange System) σχεδιάζεται για τη βελτίωση αυτών των καταστάσεων.



Εικ.3. Εικονική αναπαράσταση λειτουργίας AIS

1.4. Περιεχόμενα ενός AIS μηνύματος

Το AIS, όπως έχει προαναφερθεί, επιτρέπει την ανταλλαγή μηνυμάτων μεταξύ των πλοίων και παράκτιων σταθμών. Ένας ενεργοποιημένος αναμεταδότης AIS συνεχίζει να μεταδίδει πληροφορίες ακόμη και όταν το σκάφος του είναι αγκυροβολημένο. Οι πληροφορίες εκπομπής των μηνυμάτων που περιέχονται σε κάθε πακέτο δεδομένων AIS μπορούν να κατηγοριοποιηθούν στις δύο κύριες κατηγορίες που ακολουθούν:

✚ Δυναμικά Δεδομένα (Dynamic Data)

Μεταδίδονται αυτόματα κάθε 2 έως 10 δευτερόλεπτα ανάλογα με την ταχύτητα και την πορεία του σκάφους ενώ βρίσκονται σε κίνηση και κάθε 3 λεπτά από αγκυροβολημένα σκάφη.

Αφορούν πληροφορίες σχετικά με την κίνηση του πλοίου.

- Maritime Mobile Service Identity number (MMSI) - ένας μοναδικός 9ψήφιος αριθμός αναγνώρισης
- AIS Navigational Status - τρέχουσα ναυτιλιακή κατάσταση (εν πλω, αγκυροβολημένο, κλπ)
- Rate of Turn - ρυθμός στροφής (δεξιά ή αριστερά | 0 με 720 μοίρες το λεπτό)

- Speed Over Ground - ταχύτητα σε σχέση με τον βυθό (0 με 102 κόμβους*)
- Course Over Ground - πορεία σε σχέση με τον βυθό (0 με 359 μοίρες)
- Position Coordinates - γεωγραφικές συντεταγμένες (μήκος/πλάτος)
- Heading - κατεύθυνση (0 με 359 μοίρες)
- Bearing at own position – η κατά συγκεκριμένη κατεύθυνση παρατήρηση του αντικειμένου (0 με 359 μοίρες)
- UTC seconds - συγχρονισμένος παγκόσμιος χρόνος

* κόμβος: μονάδα ταχύτητας ίση με ένα ναυτικό μίλι την ώρα (1.852km / h)

Στατικά Δεδομένα (Static Data)

και σχετικά με το ταξίδι (voyage related information), τα οποία παρέχονται από το πλήρωμα του πλοίου και μεταδίδονται κάθε 6 λεπτά ανεξάρτητα από την κατάσταση κίνησης του σκάφους.

Μεταφέρουν πληροφορίες σχετικά με την περιγραφή του πλοίου.

- International Maritime Organisation number (IMO) - αριθμός αναγνώρισης του πλοίου, ο οποίος παραμένει ίδιος και κατά τη μεταφορά της εγγραφής του πλοίου σε άλλη χώρα
- Call Sign - διακριτικό κλήσης, διεθνές ραδιοφωνικό σήμα που έχει εκχωρηθεί στο σκάφος από τη χώρα νηολόγησης
- Name - όνομα πλοίου, μέχρι 20 χαρακτήρες
- Type - τύπος πλοίου και φορτίου
- Dimensions - διαστάσεις πλοίου, κατά προσέγγιση στον πλησιέστερο μετρητή (με βάση τη θέση του σταθμού AIS στο πλοίο)
- Θέση της κεραίας του συστήματος εντοπισμού πάνω στο σκάφος
- Type of positioning system - τύπος συστήματος εντοπισμού θέσης (GPS, DGPS)
- Destination - προορισμός
- Draught - Βύθισμα πλοίου (0,1 με 25,5 μέτρα)
- ETA – εκτιμώμενη ώρα άφιξης

Είναι πολύ σημαντικό, το πλήρωμα ή ο υπεύθυνος σκάφους να επιβεβαιώνει ότι παρέχονται στο σύστημα σωστές πληροφορίες για το πλοίο σχετικά με όλα τα πεδία που σχετίζονται με τα στατικά δεδομένα και τις πληροφορίες ταξιδιού.

2. State of the Art

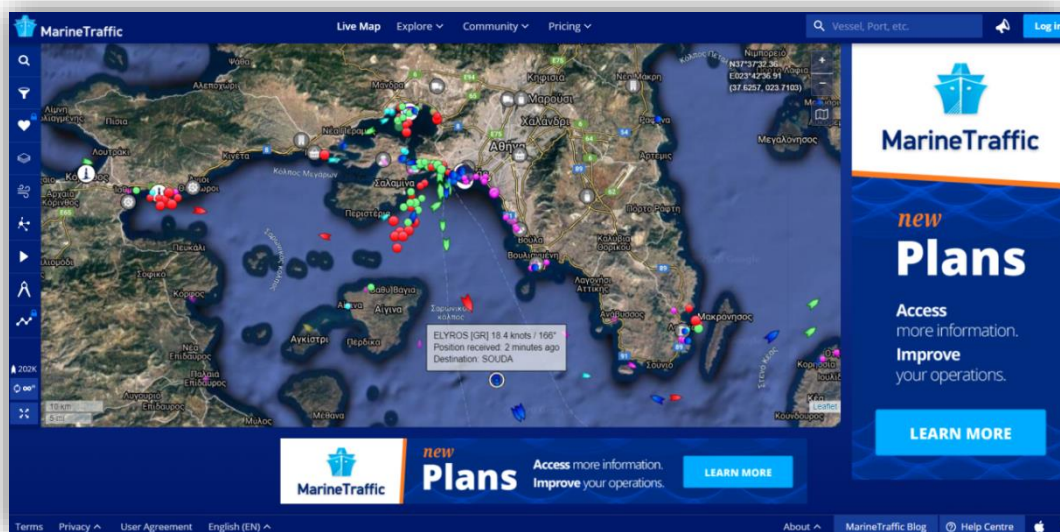
Η ταχεία πρόοδος της τεχνολογίας βελτίωσε σημαντικά τις μεθόδους monitoring και tracking των πλοίων που υπήρχαν στο πρόσφατο παρελθόν. Ενώ οι ταχέως αναπτυσσόμενες δορυφορικές υπηρεσίες επέτρεψαν τώρα την παρακολούθηση των πλοίων σε όλο τον κόσμο, κάνοντας την ευκολότερη και διασφαλίζοντας έτσι την ασφάλεια των πλοίων από διάφορες απειλές. Σε ένα τόσο ενδιαφέρον πεδίο, δεν είναι λίγοι αυτοί που θέλησαν και θέλουν να ασχοληθούν με την θαλάσσια επιτήρηση προσθέτοντας ο καθένας το δικό τους "στίγμα". Με τη βοήθεια των αναμεταδοτών με τους οποίους είναι εξοπλισμένα τα σκάφη, οι δορυφόροι συλλέγουν σήματα AIS από τα σκάφη για να παρέχουν τη θέση, τη διαδρομή, την ταχύτητα και τον τύπο του πλοίου, μεταξύ άλλων, μέσω διαφόρων διαδικτυακών πλατφορμών. Κάπου στο ενδιάμεσο όλων αυτών των διαδικασιών, μεταξύ της συλλογής των μηνυμάτων-δεδομένων και της προβολής τους, κάθε εφαρμογή θα χρειαστεί να επεξεργαστεί τα δεδομένα της. Συνήθως εφαρμόζονται περαιτέρω αλγόριθμοι αυτόματου καθαρισμού και αυτόματης διόρθωσης, καθιστώντας δυνατή την σωστή και ακριβή παρακολούθηση σκαφών οποιαδήποτε δεδομένη στιγμή, από οπουδήποτε στον κόσμο.

2.1. Παρόμοιες εφαρμογές στις μέρες μας

Επί του παρόντος, αρκετά websites παρέχουν παρακολούθηση πλοίων χρησιμοποιώντας το AIS για την εμφάνιση της θέσης σε πραγματικό χρόνο των πλοίων. Ωστόσο, δεδομένου ότι υπάρχουν διάφορες διαδικτυακές υπηρεσίες για την παροχή τέτοιων πληροφοριών, ο σκοπός του παρεχόμενου περιεχομένου συνήθως ποικίλλει. Επίσης, αν και οι περισσότεροι από αυτούς τους παρόχους υπηρεσιών υπόσχονται θέση "σε πραγματικό χρόνο" των πλοίων, τα δεδομένα που εμφανίζονται είναι πάντα με καθυστέρηση λίγων λεπτών ή ακόμα και αρκετών ωρών. Επομένως, αυτές οι πληροφορίες δεν πρέπει σε καμία περίπτωση να χρησιμοποιούνται για σκοπούς πλοήγησης.

Παρακάτω αναφέρονται μερικοί από τους εξέχοντες ιστότοπους που χρησιμοποιούνται ευρέως για σχεδόν ακριβή παρακολούθηση πλοίων στο διαδίκτυο. Αυτοί οι ιχνηλάτες σκαφών παρέχουν όχι μόνο την τοποθεσία του πλοίου, αλλά και τεχνικές και μη τεχνικές λεπτομέρειες, καθορισμένες διαδρομές, ακόμη και φωτογραφίες.

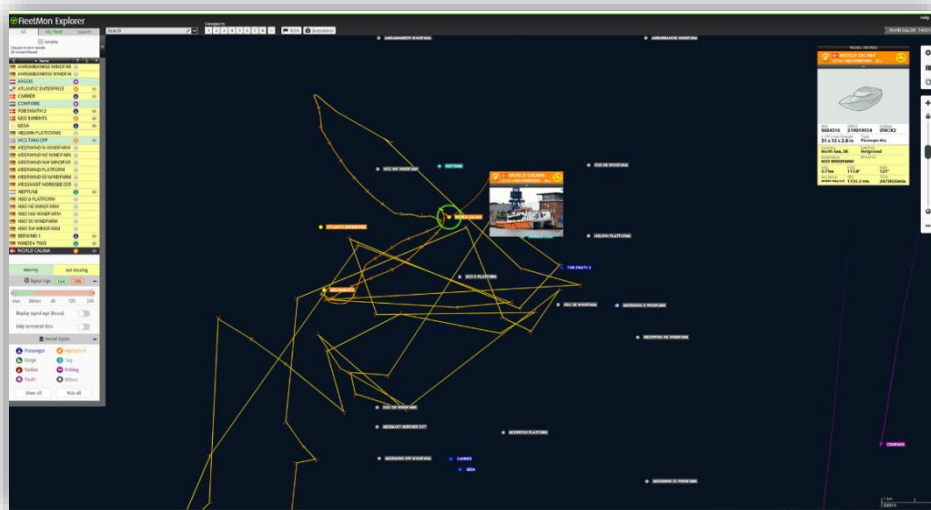
- ◆ Marine-Traffic (από το 2007). Η Marine-Traffic, ένα από τα πιο διάσημα και ευρέως χρησιμοποιούμενα διαδικτυακά συστήματα παρακολούθησης πλοίων, προσφέρει δεδομένα σε πραγματικό χρόνο των πλοίων μαζί με προηγμένες δυνατότητες αναζήτησης. Προσφέρει στους ιδιοκτήτες σκαφών πολλές υπηρεσίες, όπως δορυφορική κάλυψη AIS, βελτιωμένη παρακολούθηση δορυφόρων, χάρτες προηγμένης πυκνότητας, ναυτικούς χάρτες κ.λπ. Εκτός από την παρακολούθηση πλοίων, η Marine-Traffic επιτρέπει επίσης την παρακολούθηση διαφόρων λιμένων σε όλο τον κόσμο. Το σύστημα AIS της ιστοσελίδας διαθέτει μια τεράστια βάση δεδομένων πλοίων κάθε τύπου. Σύμφωνα με την Marine-Traffic, καταγράφει τουλάχιστον 800 εκατομμύρια θέσεις σκαφών καθώς και 18 εκατομμύρια συμβάντων που σχετίζονται με πλοία και λιμάνια κάθε μήνα και προσφέρει πληροφορίες για 650000 θαλάσσια περιουσιακά στοιχεία, όπως πλοία, λιμάνια και φώτα κ.λπ. Αυτό το ανοιχτό, κοινοτικό έργο προσφέρει μια σειρά υπηρεσιών δωρεάν, ενώ οι πιο προηγμένες λειτουργίες παρέχονται με χρέωση. Σύμφωνα με τον ιστότοπο, λειτουργεί περισσότερους από 2.000 σταθμούς AIS που βρίσκονται σε περισσότερες από 165 χώρες και έχει 2 εκατομμύρια εγγεγραμμένους χρήστες και περίπου 13 εκατομμύρια επισκέψεις στον ιστότοπό του κάθε μήνα.



Εικ.4. Δείγμα της εφαρμογής Marine-Traffic

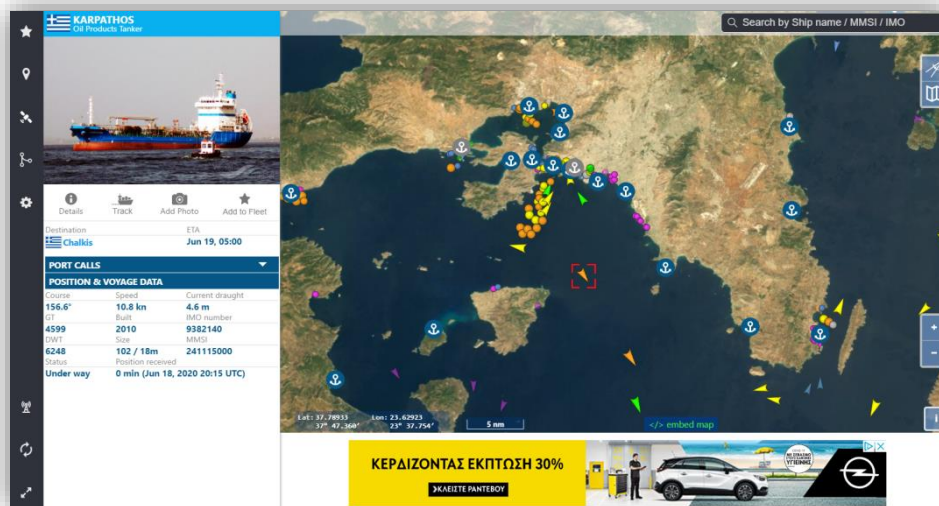
- ◆ FleetMon (από το 2007). Η FleetMon δημιούργησε την πρώτη δημόσια συλλογή δεδομένων πλοίων και λιμανιών στον κόσμο, η οποία βοηθά τους οργανισμούς να παρέχουν καλύτερες αναφορές υλικοτεχνικής υποστήριξης (logistics). Προσφέρει δεδομένα θέσης AIS σε πραγματικό χρόνο των πλοίων, καθώς και τεχνικές πληροφορίες για περισσότερα από 500.000 πλοία. Ακόμα, παρέχει τα στοιχεία του πλοίου, τα δρομολόγια και τις αφίξεις στο λιμάνι, τα μοτίβα των συναλλαγών του και τις φωτογραφίες σύμφωνα με την αναζήτηση του πελάτη. Όλα αυτά τα

διαθέτει για σημαντικές επιχειρήσεις στον κόσμο, ώστε να μπορούν να αντιμετωπίσουν τις θαλάσσιες προκλήσεις τους. Επίσης, χρησιμοποιώντας το διαδραστικό εργαλείο που ονομάζεται FleetMon Explorer, μπορείς να δεις σε πραγματικό χρόνο μια προβολή της θαλάσσιας κυκλοφορίας και να έχεις πρόσβαση σε πληροφορίες σχετικά με την παρακολούθηση των λειτουργιών, το χρονοδιάγραμμα των logistics, την παρακολούθηση των στόλων και την ανάλυση κυκλοφορίας κ.λπ. Το FleetMon έχει επί του παρόντος πελάτες σε 164 χώρες και περίπου μισό εκατομμύριο χρήστες. Παράλληλα είναι μια από τις καλύτερες υπηρεσίες παρακολούθησης πλοίων που προσφέρει επίσης Android και iOS εφαρμογές.



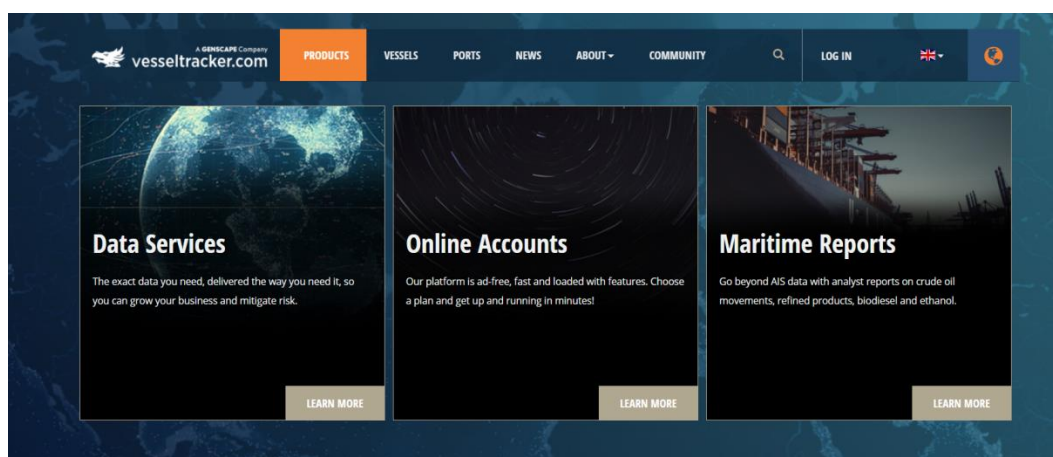
Εικ.5. Δείγμα της εφαρμογής FleetMon

- ◆ VesselFinder (από το 2011). Το VesselFinder προσφέρει real-time δεδομένα σε περισσότερα από 100.000 πλοία σε καθημερινή βάση με τη βοήθεια ενός μεγάλου δικτύου επίγειων δεκτών και δορυφόρων AIS. Παρέχοντας υπηρεσίες σε ψηφιακές, κοινωνικές και κινητές πλατφόρμες, το VesselFinder προσφέρει αποτελεσματικές υπηρεσίες παρακολούθησης, κλήσεις λιμένων, master data και ένα λεπτομερές ιστορικό ταξιδιού κάθε πλοίου. Επιπλέον, παρέχει ιστορικού περιεχομένου υπηρεσίες δεδομένων AIS που παρέχουν εξατομικευμένες αναφορές μετακινήσεων πλοίων, ανάλυση της πυκνότητας κίνησης και προσομοίωση βίντεο των κινήσεων των πλοίων, μεταξύ άλλων. Προσφέρεται επίσης σε Android και iOS εφαρμογές και είναι εν μέρει δωρεάν.



Εικ.6. Δείγμα της εφαρμογής VesselFinder

- ◆ Vesseltracker (από το 2006). Το Vesseltracker με βάση δεδομένων για περισσότερα από 1.000.000 πλοία, παρέχει προσαρμοσμένα δεδομένα και σε πραγματικό χρόνο αλλά και ιστορικά για πλοία που ταξιδεύουν σε όλο τον κόσμο. Επεξεργάζεται μισό δισεκατομμύριο μηνύματα που λαμβάνονται καθημερινά από ιδιόκτητο δίκτυο δεκτών AIS και έναν τεχνολογικά προηγμένο και αποκλειστικά AIS δορυφορικό αστερισμό και διαθέτει πάνω από 250.000 χρήστες από όλο τον κόσμο. Σε αυτούς δίνει την δυνατότητα να παρακολουθούν τα σκάφη με τη βοήθεια των προβολών διαφόρων επιπέδων στο χάρτη, συμπεριλαμβανομένων των Google δορυφόρων, του Google Earth, ναυτικών χαρτών, πληροφορίες για τον καιρό κ.λπ. Ο ιστότοπος διαθέτει μια τεράστια βάση δεδομένων που αποτελείται από ενημερωμένες προδιαγραφές πλοίων, χαρακτηριστικά, πληροφορίες ιδιοκτησίας και διαχείρισης κ.λπ. Επιπλέον, το Vesseltracker προσφέρει θαλάσσιες ειδήσεις, αναλυτικά στοιχεία και αναφορές για κινήσεις σκαφών και για συμβάντα λιμένων.



Εικ.7. Δείγμα της εφαρμογής Vesseltracker

2.2. Web mapping

Η δημιουργία ενός καλού web χάρτη δεν είναι εύκολη υπόθεση, αλλά με την πάροδο των ετών έχουμε δει πολλά εκπληκτικά παραδείγματα «γεωγραφικής οπτικοποίησης» (geovisualization). Οι περισσότερες εφαρμογές και ιστότοποι που χρησιμοποιούμε σήμερα είναι βέβαιο ότι θα εμφανίζουν τοποθεσία και διάφορα δεδομένα σε ένα χάρτη. Χρησιμοποιούνται παντού, από ιστότοπους παράδοσης φαγητού έως διαδικτυακά καταστήματα και σύνθετες εφαρμογές για εταιρείες logistics. Ευρέως χρησιμοποιούνται φυσικά και στον κόσμο του web design και ειδικά όταν αφορά την παρουσίαση των δεδομένων αναμένεται να είναι διαδραστικοί.

Ακολουθούν παραδείγματα των μέσων υλοποίησης ενός διαδραστικού χάρτη που χρησιμοποιούνται στην σύγχρονη εποχή. Αυτό όσον αναφορά την JavaScript καθώς είναι η γλώσσα προγραμματισμού με την οποία αναπτύχθηκε η εφαρμογή που παρουσιάζεται σε αυτήν την εργασία. Στην συνέχεια, παρουσιάζονται και αναλύονται οι γλώσσες, οι βιβλιοθήκες και γενικότερα όλα τα εργαλεία που επιλέχθηκαν για την εκπόνηση του προγραμματιστικού μέρους αυτής της πτυχιακής.

2.2.1. Διαδικτυακές πλατφόρμες

Πλατφόρμα ονομάζεται μια ομάδα τεχνολογιών που χρησιμοποιούνται ως βάση πάνω στην οποία αναπτύσσονται άλλες εφαρμογές, διαδικασίες ή τεχνολογίες. Με άλλα λόγια ένα framework στο οποίο μπορούν να τρέχουν εφαρμογές. Στην συγκεκριμένη εργασία δηλαδή αυτό που θα μπορούσε να χρησιμοποιηθεί ως βάση για την απεικόνιση και όχι μόνο, ο χάρτης της εφαρμογής. Παρακάτω αναφέρονται μερικά από τα πιο γνωστά mapping platforms.

- Mapbox. Το Mapbox είναι μια ανοιχτού κώδικα πλατφόρμα δεδομένων τοποθεσίας για κινητές και διαδικτυακές εφαρμογές. Παρέχει δομικά στοιχεία για την προσθήκη χαρακτηριστικών τοποθεσίας όπως χάρτες, αναζήτηση και πλοήγηση σε οποιαδήποτε διαδρομή δημιουργηθεί. Είναι μια από τις πιο προηγμένες λύσεις χαρτογράφησης στον ιστό αυτήν τη στιγμή, κατασκευασμένο για προγραμματιστές από έμπειρους προγραμματιστές. Παρέχει ένα σύνολο εργαλείων για τη δημιουργία προσαρμοσμένων χαρτών ιστού και κινητών και ένα Mapbox Editor. Προσφέρει στους χρήστες έλεγχο του στυλ (συμπεριλαμβανομένου του χάρτη φόντου) και έχει πρόσθετες υπηρεσίες, όπως δορυφορικές εικόνες και γεωκωδικοποίηση (geocoding) ή οδηγίες. Το Mapbox προσφέρει πέντε προγράμματα, που κυμαίνονται από ένα δωρεάν πρόγραμμα εκκίνησης έως ένα επιχειρηματικό πρόγραμμα μεγάλου όγκου.

- Google Maps. Οι Χάρτες Google προσφέρουν ένα σύνολο από API για διαφορετικούς σκοπούς χαρτογράφησης. Προσφέρει πρόσβαση στα δεδομένα χαρτογράφησης της Google που περιλαμβάνουν μεταξύ άλλων και το StreetView. Υπάρχουν εδώ και αρκετό καιρό, ώστε πια να

θεωρούνται ηγέτες στη βιομηχανία χαρτογράφησης και συχνά μια ασφαλής επιλογή. Επιπλέον, το JavaScript API διευκολύνει την ενσωμάτωση των Χαρτών Google σε οποιαδήποτε εφαρμογή, υπηρεσία ή ιστότοπο πια. Οι υπηρεσίες που προσφέρει δεν είναι δωρεάν και η τιμολόγηση διαφέρει ανάλογα με κάθε project.

- Bing Maps. Οι χάρτες Microsoft Bing είναι μια επίσης δημοφιλής πλατφόρμα χαρτογράφησης ιστού ιστού που παρέχεται ως μέρος της Microsoft's Bing suite των μηχανών αναζήτησης και τροφοδοτείται από Bing Maps for Enterprise framework. Από την δημιουργία τους το 2005, υπάρχει σταθερή προσθήκη νέων δυνατοτήτων και βελτιώσεων στην πλατφόρμα χαρτών Bing. Στην μία από τις δύο version τους υπάρχει ένα AJAX Interactive SDK που παρέχει δείγματα χαρακτηριστικών και αποσπάσματα κώδικα για προγραμματιστές που δημιουργούν εφαρμογές JavaScript. Η πλατφόρμα γενικά παρέχει πολλές επιλογές API για εφαρμογές, όπως Web Control, έλεγχος εφαρμογών Windows Store, Υπηρεσίες REST και Υπηρεσίες Spatial Data. Είναι εν μέρει δωρεάν και κατόπιν αιτήματος με προσαρμοσμένη τιμολόγηση.
- ArcGIS. Το ArcGIS API για JavaScript είναι ένας ελαφρύς τρόπος ενσωμάτωσης χαρτών και εργασιών σε εφαρμογές ιστού. Διατίθεται με πολλά χαρακτηριστικά, δείγματα, ομαδοποιημένα widgets και πρότυπα. Είναι δημιουργημένο από την Esri (Environmental Systems Research Institute), έναν διεθνή προμηθευτή λογισμικού συστήματος γεωγραφικών πληροφοριών (GIS), web GIS και εφαρμογών διαχείρισης γεωγραφικών βάσεων δεδομένων. Η Esri διαθέτει ένα καλά σχεδιασμένο portal για προγραμματιστές που περιλαμβάνει πρόσβαση σε ArcGIS APIs και SDK, καλή τεκμηρίωση, live demos, δείγματα κώδικα και άλλα. Γενικότερα η τεχνολογία Esri φαίνεται να χρησιμοποιείται από περισσότερους από 350.000 οργανισμούς παγκοσμίως, και πάνω από τα δύο τρίτα είναι εταιρείες Fortune 500.

2.2.2. Εργαλεία και βιβλιοθήκες για τους χάρτες

Βιβλιοθήκη (library) ονομάζουμε μια συλλογή από έτοιμα υποπρογράμματα που χρησιμοποιείται για την ανάπτυξη λογισμικού. Οι βιβλιοθήκες περιέχουν υποβοηθητικό κώδικα και δεδομένα, κλάσεις και λειτουργίες, παρέχοντας με αυτόν τον τρόπο υπηρεσίες σε προγράμματα και επιτρέποντας τη δημιουργία εφαρμογών σε υψηλότερο επίπεδο προγραμματισμού.

Προγραμματιστικό εργαλείο θεωρείται ένα πρόγραμμα υπολογιστή ή μια εφαρμογή που χρησιμοποιούν οι προγραμματιστές για να δημιουργήσουν, να αποσφαλματώσουν (debugging) ή να συντηρήσουν άλλα προγράμματα και εφαρμογές. Ο όρος αναφέρεται συνήθως σε σχετικά απλά προγράμματα που μπορούν να συνδυαστούν μαζί για την επίτευξη κάποιου αποτελέσματος.

Τα εργαλεία χαρτών μπορούν πραγματικά να μειώσουν την προσπάθεια που απαιτείται για την παραγωγή ενός διαδραστικού χάρτη, καθώς επιτρέπουν στον προγραμματιστή να εφαρμόσει κάθε είδους δυναμικούς χάρτες με μόνο λίγες γραμμές κώδικα. Οι βιβλιοθήκες μπορούν να χρησιμοποιηθούν για την τοποθέτηση ειδικών σημαδιών στον χάρτη, την σχεδίαση γραμμών με ειδικά προσαρμοσμένα γνωρίσματα, την εμφάνιση κειμένων, την αλληλεπίδραση με τον κέρσορα ή άλλα σημεία του χάρτη και πολλά άλλα ακόμα. Επιπλέον, οι περισσότεροι χάρτες μπορούν να δεχτούν περαιτέρω εξατομίκευση με CSS. Παρακάτω αναφέρονται μερικές από τις καλύτερες βιβλιοθήκες JavaScript για τη δημιουργία διαδραστικών χαρτών για τις εφαρμογές ιστού. Παρέχουν πληθώρα λειτουργιών και χαρακτηριστικών που βοηθούν τους προγραμματιστές να δημιουργούν πολύπλοκους χάρτες εύκολα και γρήγορα.

- ❖ OpenLayers. Το OpenLayers είναι μια βιβλιοθήκη JavaScript ανοιχτού κώδικα υψηλής απόδοσης για την εμφάνιση δεδομένων χάρτη σε προγράμματα περιήγησης ιστού και για τη δημιουργία διαδραστικών χαρτών χρησιμοποιώντας διάφορες υπηρεσίες χαρτογράφησης. Παρέχει ένα API για τη δημιουργία πλούσιων διαδικτυακών γεωγραφικών εφαρμογών παρόμοιων με τους Χάρτες Google και τους Χάρτες Bing. Χρησιμοποιεί ως πηγή δεδομένων για το χάρτη το OpenStreetMap. Το προϊόν είναι κατάλληλο για κατασκευή χαρτών σε όλες τις συσκευές και τα προγράμματα περιήγησης και μπορεί να χρησιμοποιηθεί με το CSS για εξατομίκευση. Επίσης, επίσης ικανό να παρέχει διανυσματικά δεδομένα από GeoJSON, TopoJSON, KML, GML και άλλες μορφές γεωγραφικών δεδομένων.
- ❖ Chart.js. Το Chart.js είναι μια δωρεάν βιβλιοθήκη JavaScript ανοιχτού κώδικα για οπτικοποίηση δεδομένων, η οποία υποστηρίζει 8 τύπους γραφημάτων: bar, line, area, pie, bubble, radar, polar, και scatter. Είναι η δεύτερη πιο δημοφιλής βιβλιοθήκη χαρτογράφησης JS στο GitHub (από τον αριθμό των αστεριών) μετά την D3.js, που θεωρείται σημαντικά πιο εύκολο στη χρήση, αλλά λιγότερο προσαρμόσιμο από την τελευταία. Αποδίδεται σε HTML5 canvas και θεωρείται ως μία από τις καλύτερες βιβλιοθήκες οπτικοποίησης δεδομένων.
- ❖ Polymaps. Το Polymaps είναι μια δωρεάν βιβλιοθήκη JavaScript για τη δημιουργία δυναμικών, διαδραστικών και προσαρμοσμένων χαρτών. Εκτός από τη συνηθισμένη χαρτογραφία από OpenStreetMap, CloudMade, Bing, κ.λπ., λειτουργεί και με image-based και vector-tiled χάρτες παρέχοντας γρήγορη απεικόνιση από multi-zoom datasets πάνω στους χάρτες και υποστηρίζοντας ποικιλία οπτικών παρουσιάσεων. Το κύριο πλεονέκτημά του είναι ότι καθιστά πολύ εύκολη τη σύνθεση δεδομένων raster και vector από πολλές διαφορετικές πηγές. Επίσης, καθώς τα Polymaps μπορούν να φορτώσουν δεδομένα σε ένα πλήρες εύρος κλιμάκων, είναι κατάλληλο για εμφάνιση πληροφοριών από επίπεδο χώρας μέχρι και πολιτείας, πόλης, γειτονιάς

και μεμονωμένους δρόμους. Την ίδια στιγμή είναι αρκετά αποτελεσματικά, γρήγορα, με καλή διαχείριση της φόρτωσης πλακιδίων φόντου και είναι μόνο 30k Javascript.

Στα παραπάνω σύγχρονα μέσα ανάπτυξης εφαρμογών συμπεριλαμβάνονται και όσα ακολουθούν στην επόμενη παράγραφο, τα οποία είναι και αυτά που χρησιμοποιήθηκαν για την παρούσα εργασία και θα αναλυθούν στο επόμενο κεφάλαιο.

2.3 Εργαλεία που χρησιμοποιήθηκαν στην εργασία

2.3.1. Κύρια γλώσσα προγραμματισμού

Η ανάπτυξη κώδικα σε JavaScript είναι συνήθως ελαφριά και μια επιτυχημένη διαδικασία, όπως συνέβη και με την επιλογή της για την παρούσα εφαρμογή. Η JavaScript είναι μια ισχυρή αντικειμενοστραφής γλώσσα που θεωρείται να είναι ιδανική για την ανάπτυξη εφαρμογών στο διαδίκτυο αλλά και εκτός σύνδεσης, με λειτουργίες πρώτης κατηγορίας. Κάτι που δεν μπορεί να ειπωθεί για τις περισσότερες γλώσσες εκεί έξω. Ένα από τα καλύτερα πράγματα για το JavaScript είναι ότι υπάρχουν πάρα πολλά προϋπάρχοντα εργαλεία που μπορούν να χρησιμοποιηθούν. Επίσης, το JavaScript συνδυάστηκε με CSS για τη δημιουργία δυναμικών ιστοσελίδων, οι οποίες έχουν γίνει δημοφιλείς ως μια πιο αποτελεσματική εναλλακτική λύση σε σχέση με ιστότοπους που βασίζονται σε Flash. Αρχικά αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται. Ωστόσο, πολλά περιβάλλοντα εκτός προγράμματος περιήγησης όπως Node.js, Apache, CouchDB και Adobe Acrobat, τη χρησιμοποιούν επίσης. Η Javascript δεν θα πρέπει να συγχέεται με τη Java, που είναι διαφορετική γλώσσα προγραμματισμού και με διαφορετικές εφαρμογές. Η χρήση της λέξης "Java" στο όνομα της γλώσσας έχει περισσότερη σχέση με το προφίλ του προϊόντος που έπρεπε να έχει και λιγότερο με κάποια πιθανή συμβατότητα ή άλλη στενή σχέση με τη Java. Ρόλο σε αυτήν τη σύγχυση έπαιξε και ότι η Java και η Javascript έχουν δεχτεί σημαντικές επιρροές από τη γλώσσα C, ειδικά στο συντακτικό, ενώ είναι και οι δύο αντικειμενοστραφείς γλώσσες [6].

2.3.2. Βιβλιοθήκες

- *Leaflet*

Αρχική έκδοση: Μάιος 2011.

Γλώσσα προγραμματισμού: JavaScript

Τύπος: βιβλιοθήκη JavaScript

Μια ευρέως χρησιμοποιούμενη βιβλιοθήκη ανοιχτού κώδικα JavaScript που χρησιμοποιείται για τη δημιουργία διαδραστικών χαρτών. Λειτουργεί αποτελεσματικά σε όλες τις μεγάλες πλατφόρμες επιτραπέζιων και κινητών συσκευών, υποστηρίζοντας επίσης HTML5 και CSS3. Έχει σχεδιαστεί με γνώμονα την απλότητα, την απόδοση και τη χρηστικότητα, μπορεί να επεκταθεί με πολλά πρόσθετα (για το στυλ, για διαφορετικά διαδραστικά επίπεδα στο χάρτη, κλπ), έχει ένα εύχρηστο API και έναν απλό, αναγνώσιμο πηγαίο κώδικα. Μαζί με το OpenLayers και το API Χαρτών Google, είναι μια από τις πιο δημοφιλείς βιβλιοθήκες χαρτογράφησης JavaScript και χρησιμοποιείται από μεγάλους ιστότοπους όπως το FourSquare, το Pinterest και το Flickr. Τέλος, λαμβάνοντας υπόψιν και το μέγεθος της, περίπου 38 KB JS, έχει όλες τις δυνατότητες για τη δημιουργία εφαρμογών χαρτογράφησης ιστού που χρειάζονται οι περισσότεροι προγραμματιστές.

- *D3.js*

Αρχική έκδοση: Φεβρουάριος 2011.

Γλώσσα προγραμματισμού: JavaScript

Τύπος: Data Visualization, βιβλιοθήκη JavaScript

Μια JavaScript βιβλιοθήκη για την παραγωγή δυναμικών και διαδραστικών οπτικοποιήσεων των δεδομένων σε προγράμματα περιήγησης ιστού. Χρησιμοποιεί πρότυπα SVG, HTML5 και CSS και «ζωντανεύει» τα εκάστοτε δεδομένα. Η βιβλιοθήκη D3.js χρησιμοποιεί προεγκατεστημένες συναρτήσεις για την επιλογή στοιχείων, τη δημιουργία αντικειμένων SVG, την τροποποίηση του στυλ τους ή την προσθήκη μεταβάσεων, δυναμικών εφέ ή συμβουλευτικών πληροφοριών σε αυτά. Αυτά τα αντικείμενα μπορούν έπειτα να μορφοποιηθούν χρησιμοποιώντας CSS. Τα μεγάλα σύνολα δεδομένων μπορούν να συνδεθούν με αντικείμενα SVG χρησιμοποιώντας συναρτήσεις D3.js για τη δημιουργία γραφημάτων και διαγραμμάτων. Ενώ αυτά τα δεδομένα μπορούν να είναι σε διάφορες μορφές όπως JSON, τιμές διαχωρισμένες με κόμμα (CSV) ή GeoJSON. Ωστόσο, εάν απαιτείται, οι συναρτήσεις JavaScript μπορούν να γραφτούν για να διαβάσουν άλλες μορφές δεδομένων. Η έμφαση της D3 στα πρότυπα ιστού προσφέρει όλες τις δυνατότητες των σύγχρονων προγραμμάτων περιήγησης χωρίς να πρέπει να περιοριστείς με ένα ιδιόκτητο framework, συνδυάζοντας ισχυρά στοιχεία οπτικοποίησης και μια προσέγγιση που βάσει δεδομένων για τον χειρισμό του DOM.

2.3.3. Συστατικά στοιχεία της εφαρμογής

Παρακάτω αναφέρονται συνολικά τα στοιχεία που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής:

- ✓ Leaflet.js Η κορυφαία βιβλιοθήκη JavaScript ανοιχτού κώδικα για διαδραστικούς χάρτες φιλικούς προς κινητά.
- ✓ Javascript. Διερμηνευμένη γλώσσα προγραμματισμού που δημιουργήθηκε το 1995 από την εταιρεία Netscape.
- ✓ D3.js Μια βιβλιοθήκη JavaScript για την οπτικοποίηση δεδομένων χρησιμοποιώντας πρότυπα ιστού.
- ✓ HTML. Η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, με τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων.
- ✓ CSS. Γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων ύφους που χρησιμοποιείται για τον έλεγχο της εμφάνισης (style) ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης.
- ✓ SVG. Επεκτάσιμη γλώσσα σήμανσης (XML) - βασισμένη σε διανυσματική μορφή εικόνας για δισδιάστατα γραφικά με υποστήριξη αλληλεπίδρασης και κινούμενης εικόνας.
- ✓ API. Η διεπαφή των προγραμματιστικών διαδικασιών που παρέχει ένα λειτουργικό σύστημα, βιβλιοθήκη ή εφαρμογή προκειμένου να επιτρέπει να γίνονται προς αυτά αιτήσεις από άλλα προγράμματα ή/και ανταλλαγή δεδομένων.

3. Διατύπωση και Σχηματισμός εφαρμογής

3.1 Θέμα πτυχιακής

Το θέμα και τελικά ο στόχος που τέθηκε και επιτεύχθηκε σε αυτήν την πτυχιακή εργασία είναι η παρακολούθηση σε πραγματικό χρόνο των θέσεων διαφόρων πλοίων. Επιπλέον, η εμφάνιση πληροφοριών για τα στοιχεία των πλοίων και της διαδρομής την οποία έχουν διανύσει μέχρι και την πιο πρόσφατη τους θέση. Η βάση για όλα τα παραπάνω αποφασίστηκε να είναι ένας λεπτομερής, διαδραστικός και φιλικός προς τους χρήστες χάρτης.

Για να πραγματοποιηθεί αυτός ο στόχος αρχικά χρειαζόντουσαν τα κατάλληλα δεδομένα, τα οποία δεν χρησιμοποιήθηκαν από κάποιο ενδεικτικό σύνολο θαλάσσιων δεδομένων αλλά από αληθή βάση δεδομένων. Στην περίπτωση μας αυτό έγινε μέσω ενός API που μας παρείχε μια ροή δεδομένων από σταθμό βάσης της υπηρεσίας AIS.

3.2 Συλλογή δεδομένων

Η συλλογή των δεδομένων έγινε εφικτή λόγω ενός πομποδέκτη AIS στο λεκανοπέδιο Αττικής, σε υψόμετρο 60 μέτρων και με ενδεικτική έκταση κάλυψης περίπου 615 km² (παλιότερα στοιχεία έως και 1048.9 km²). Η έκταση αναφέρεται ως ενδεικτική και συνήθως αλλάζει καθώς σε αυτό παίζουν ρόλο και οι τρέχουσες καιρικές συνθήκες. Όσον αφορά την ροή δεδομένων, η συχνότητα με την οποία λαμβάνονται είναι περίπου κάθε ένα λεπτό. Για αυτό το λόγο και στον κώδικα της εφαρμογής έχει προγραμματιστεί ανανέωση των δεδομένων ανά ένα λεπτό. Η σύνδεση όλων αυτών των δεδομένων και της εφαρμογής όπως έχει αναφερθεί ξανά έγινε μέσω API.

3.2.1. Ορισμός API

Μια διεπαφή προγραμματισμού εφαρμογών (API) είναι μια διεπαφή υπολογιστών που καθορίζει τις αλληλεπιδράσεις μεταξύ πολλαπλών διαμεσολαβητών λογισμικού. Καθορίζει τα είδη των κλήσεων ή των αιτημάτων που μπορούν να γίνουν, τον τρόπο πραγματοποίησής τους, τις μορφές δεδομένων που πρέπει να χρησιμοποιηθούν, τις συμβάσεις που πρέπει να ακολουθηθούν κ.λπ. Επίσης, μπορεί να παρέχει μηχανισμούς επέκτασης, έτσι ώστε οι χρήστες να μπορούν να επεκτείνουν την υπάρχουσα λειτουργικότητα με διάφορους τρόπους και σε διαφορετικούς βαθμούς [7]. Ακόμα, ένα API μπορεί να είναι εξ ολοκλήρου προσαρμοσμένο, ειδικό ως προς ένα στοιχείο ή μπορεί να σχεδιαστεί με βάση ένα βιομηχανικό πρότυπο για να διασφαλιστεί η διαλειτουργικότητα του. Πιο απλά, το API είναι ένα ενδιάμεσο λογισμικό που επιτρέπει την επικοινωνία μεταξύ δύο εφαρμογών. Είναι ο φορέας που παραδίδει το αίτημά και στη συνέχεια επιστέφει την απάντηση. Το γεγονός ότι καθορίζει τις λειτουργίες που είναι ανεξάρτητες από τις αντίστοιχες υλοποιήσεις τους, επιτρέπει στις εν λόγω υλοποιήσεις και ορισμούς να διαφέρουν χωρίς να εκθέτει ο ένας τον άλλον. Ως εκ τούτου, μέσα από ένα καλό API είναι ευκολότερο να αναπτυχθεί ένα πρόγραμμα με την παροχή των δομικών στοιχείων, καθώς είναι σύνηθες όταν οι προγραμματιστές ξεκινάνε την δημιουργία ενός κώδικα να μην ξεκινάνε από το απόλυτο μηδέν.

3.2.2. Μορφή δεδομένων

Τα θαλάσσια δεδομένα που επεξεργάζονται καθ' όλη την διάρκεια της εφαρμογής είναι σε μορφή JSON. Το JSON είναι μια ανοιχτή τυπική μορφή αρχείου για ανταλλαγή δεδομένων, η οποία χρησιμοποιεί κείμενο αναγνώσιμο από τον άνθρωπο (είναι σε μορφή κειμένου) για την αποθήκευση και τη μετάδοση αντικειμένων-δεδομένων. Τα αντικείμενα αυτά αποτελούνται από μια συλλογή από ζεύγη χαρακτηριστικών-τιμών και οργανώνονται σε ταξινομημένες λίστες τιμών (πίνακας, διάνυσμα, λίστα ή ακολουθία). Ταυτόχρονα είναι εύκολο και για τα μηχανήματα να την αναλύσουν και να δημιουργήσουν. Προήλθε από JavaScript, αλλά πολλές σύγχρονες γλώσσες προγραμματισμού περιλαμβάνουν κώδικα για τη δημιουργία και ανάλυση δεδομένων μορφής JSON. Είναι μια μορφή δεδομένων ανεξάρτητη από τη

γλώσσα και με ένα ευρύ φάσμα εφαρμογών. Όλες αυτές οι ιδιότητες καθιστούν το JSON μια ιδανική γλώσσα ανταλλαγής δεδομένων.

4. Ανάλυση Εφαρμογής

4.1. Περιγραφή και Λειτουργία Dataset

4.1.1 Χρήση API στην εφαρμογή

Όπως αναφέρθηκε και στην παραπάνω παράγραφο 3.2. η συλλογή των δεδομένων γίνεται ανά ένα λεπτό από έναν πομποδέκτη AIS μέσω API. Στην συγκεκριμένη εργασία, το API που καλείται από την εφαρμογή βρίσκεται στην IP address <http://83.212.240.62/>. Συγκεκριμένα, μεταβαίνοντας στην <http://83.212.240.62:3000/maritime/api-now/> έχουμε πρόσβαση στα δεδομένα σε πραγματικό χρόνο κάθε φορά.

Όσον αφορά τις επιλογές που υπάρχουν για να φιλτράρουμε ελαφρώς τα δεδομένα, υπάρχουν ακόμα τρεις μορφές που μπορεί να έχει το URL.

- Για ανάκτηση δεδομένων για ένα συγκεκριμένο σκάφος τις τελευταίες δώδεκα ώρες: [http://83.212.240.62:3000/maritime/api/"mmsi"](http://83.212.240.62:3000/maritime/api/)
- Για ανάκτηση δεδομένων για ένα συγκεκριμένο σκάφος κατά τη διάρκεια μιας περιόδου (Epoch σε UTC): [http://83.212.240.62:3000/maritime/api/"from"/"to"/"mmsi"](http://83.212.240.62:3000/maritime/api/)
- Για ανάκτηση δεδομένων για όλα τα σκάφη κατά τη διάρκεια μιας περιόδου (Epoch σε UTC): [http://83.212.240.62:3000/maritime/api/"from"/"to"](http://83.212.240.62:3000/maritime/api/)

Συμπληρώνοντας το mmsi χρειάζεται να γράψουμε τον μοναδικό 9ψήφιο αριθμό αναγνώριση του πλοίου που μας ενδιαφέρει.

Συμπληρώνοντας τα from/to χρειάζεται ο χρόνος να είναι σε Unix Time (ο αριθμός των δευτερολέπτων που έχουν περάσει από τις 00:00:00 UTC Πέμπτη, 1 Ιανουαρίου 1970)

4.1.2 Συνδέσεις με το API

Στον κώδικα της εφαρμογής κλήσεις προς το API συμβαίνουν σε τέσσερις διαφορετικές περιπτώσεις, με διαφορετικά -χρονικά- δεδομένα επιστροφής, ανάλογα με το σκοπό που θα εξυπηρετήσουν. Το αρχικό URL είναι δηλωμένο ως global variable για την διευκόλυνση μετέπειτα λειτουργιών.

```

7 // Global variables
8 const api_url = "http://83.212.240.62:3000/maritime/api-now/ ";

```

Εικ. Α. URL σύνδεσης με το API

Παρακάτω αναφέρονται τα σημεία στα οποία η εφαρμογή «τραβάει» δεδομένα μέσω συναρτήσεων, οι οποίες θα αναλυθούν σε επόμενη παράγραφο σχετικά με τις λειτουργίες της εφαρμογής. Η κλήση γίνεται μέσω της μεθόδου `fetch()`.

-Η πρώτη συνάρτηση που καλείται στον κώδικα είναι και αυτή που κάνει την πρώτη ανάκτηση δεδομένων. Αφορά το `api_url` και γίνεται μέσα από την `async function getDataOfShips()`. Με αυτόν τον τρόπο είναι διαθέσιμα όλα τα σήματα των πλοίων που λαμβάνονται από την κεραία, την στιγμή που έγινε το αίτημα.

-Το ίδιο URL χρησιμοποιείται και στην `async function updateData()`, η οποία αφορά την ανανέωση των δεδομένων για να υπάρχει το αποτέλεσμα του πραγματικού χρόνου. Αυτή η συνάρτηση -άρα και τα δεδομένα- είναι προγραμματισμένη να καλείται κάθε ένα λεπτό.

-Το επόμενο αίτημα για δεδομένα γίνεται στην `async function getRouteData(mmsi)`, την συνάρτηση που χρησιμοποιείται για να συλλέξει τα δεδομένα μιας διαδρομής ενός πλοίου. Σε αυτήν την περίπτωση παίρνοντας την παράμετρο `mmsi` για το πλοίο που μας ενδιαφέρει σχηματίζεται ένα νέο προσαρμοσμένο URL. Με αυτό συλλέγονται όλα τα σήματα του συγκεκριμένου πλοίου τις τελευταίες τέσσερις ώρες.

```

261 let url = "http://83.212.240.62:3000/maritime/api/" + minus4H + "/" + currentDate + '/' + mmsi;

```

Εικ. Β. URL για στίγματα τεσσάρων ωρών ενός πλοίου

-Η τελευταία περίπτωση ανάκτησης δεδομένων από το API είναι στην `async function showCoverage()` όπου σκοπός της συνάρτησης είναι να συλλέξει όλα τα στίγματα από όλα τα πλοία που κατέγραψε τις τελευταίες τέσσερις ώρες. Οπότε, δημιουργείται και πάλι ένα URL.

```

419 const urlA = "http://83.212.240.62:3000/maritime/api/" + minus4H + "/" + currentDate;

```

Εικ. Γ. URL για στίγματα των τεσσάρων τελευταίων ωρών

4.1.3. Δείγμα Θαλάσσιων δεδομένων

Τα θαλάσσια αυτά δεδομένα παραλαμβάνονται σε μορφή JSON, περιέχουν δηλαδή ζεύγη χαρακτηριστικών - τιμών για όλα τα πλοία από τα οποία δέχονται σήμα. Τα χαρακτηριστικά είναι ίδια για όλα τα πλοία. Σε ιδανική περίπτωση δεν περιέχουν ποτέ λάθη, ανακρίβειες ή άγνωστες τιμές. Στην πραγματικότητα, κάθε εταιρεία ή ομάδα επεξεργασίας φροντίζει για αυτόν τον «καθαρισμό» πριν τα χρησιμοποιήσει σε κάποια υπηρεσία.

Οι πληροφορίες λοιπόν που λαμβάνονται, έχουν αναλυθεί στην παράγραφο 1.4. και είναι οι εξής:

"timestamp", "mmsi", "imo", "navigational_status", "longitude", "latitude", "heading", "cog", "sog", "ship_name", "call_sign", "ship_type" (τύπος πλοίου), "draught" (βύθισμα), "size_bow" (πλώρη), "size_stern" (πρύμνη), "size_port" (λιμένας), "size_starboard" (δεξιά πλευρά πλοίου) και "destination". Ως δομή δεδομένων βρίσκονται μέσα σε μία λίστα.

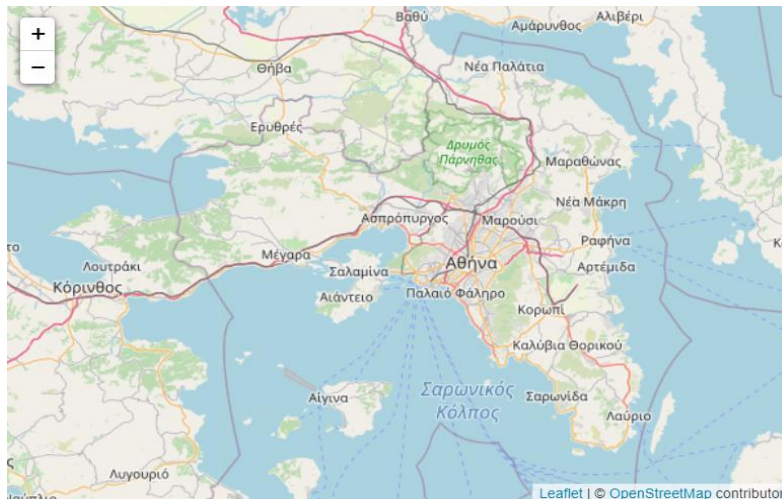
```
{
  "timestamp": "2020-06-23T19:25:22.000Z",
  "mmsi": "210205000",
  "imo": "9204324",
  "navigational_status": "9",
  "longitude": 23.640403333333333,
  "latitude": 37.9407,
  "heading": 56,
  "cog": 314.7,
  "sog": 0,
  "ship_name": "SIFNOS JET",
  "call_sign": "5BLV4",
  "ship_type": "40",
  "draught": "30",
  "size_bow": "23",
  "size_stern": "22",
  "size_port": "10",
  "size_starboard": "3",
  "destination": "SE-SI-MI-SI-SE-PIR"
}
```

Εικ.8. Δείγμα ενός από τα JSON data

4.2. Βάση εφαρμογής

Πριν ακόμα αναλύσουμε στις λειτουργίες της εφαρμογής, θα πρέπει να αναφερθούμε στην βάση όλων αυτών, στο που θα παρουσιαστούν και θα κάνουν όλες αυτές τις δυνατότητες εφικτές, στον χάρτη. Ο χάρτης που επιλέχθηκε και είναι στο κατώτερο επίπεδο παρουσίασης της εφαρμογής, εισήχθη από την JavaScript βιβλιοθήκη Leaflet. Μαζί με αυτόν κρατήθηκαν δύο από τις πολλές ιδιότητες που προσφέρει, αυτή της μεγέθυνσης - σμίκρυνσης και αυτή της αλλαγής της μορφής του χάρτη (βασικό, ανθρωπιστικό κλπ.). Ακόμα προστέθηκε το plug-in για προβολή της εφαρμογής σε εικόνα πλήρους οθόνης. Είναι ένας αρκετά λεπτομερής χάρτης και επιδέχεται πολλές προσθήκες και τροποποιήσεις, κάτι που είναι αρκετά σημαντικό. Ειδικά στην συγκεκριμένη περίπτωση που όλη η υπόλοιπη ανάπτυξη της εφαρμογής θα γινόταν με την χρήση της D3.js, όπως και έγινε τελικά εφικτό και συνδυάστηκαν με επιτυχία.

Όσο για το τελικό αποτέλεσμα της εμφάνισης της εφαρμογής στον χρήστη, τα περισσότερα στοιχεία του χάρτη έχουν τροποποιηθεί με χρήση CSS για εξατομίκευση της εμφάνισης τους για αυτό και ο συγκεκριμένος κώδικας είναι εξωτερικός και αποθηκευμένος σε ένα ξεχωριστό αρχείο map.css, το οποίο είναι συνδεδεμένο με το HTML αρχείο της εφαρμογής.



Εικ.9. Παράδειγμα βασικού χάρτη Leaflet

Το πρώτο κομμάτι που προσδιορίζεται στον κώδικα είναι ο χάρτης. Δημιουργήθηκε μια μεταβλητή `map` στην οποία ανατέθηκε ο χάρτης, ενώ ταυτόχρονα έπρεπε να προσδιοριστεί ποιο κομμάτι του χάρτη θα είναι αυτό που θα πρωτοεμφανίζεται στο χρήστη και πόσο μεγεθυμένο θα είναι. Στην δική μας περίπτωση, η ευρύτερη περιοχή γύρω από το λεκανοπέδιο της Αττικής. Σε αυτό το αρχικό σημείο ορίστηκε και ακόμα ένα επίπεδο του χάρτη, ένα `svgLayer`, το οποίο προορίζεται για την προβολή των καραβιών και συνδέθηκε με το αρχικό layer. Η σύνδεση γίνεται μέσα από συνάρτηση που παρέχει η D3, `d3.select()`. Το παρακάτω κομμάτι κώδικα προγραμματίζει ακριβώς αυτά.

```
1 let map = L.map("map").setView([37.86, 23.6], 10).setMinZoom(3);
2 let svgLayer = L.svg().addTo(map);
3 let svg = d3.select("#map").select("svg");
4 // Append empty placeholder g element to the SVG that'll contain geometry elements
5 let g = svg.select("g");
6
7 mapLink = L.tileLayer("http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png").addTo(map);
```

Εικ. 1. Δημιουργία βασικού χάρτη

4.2.1. SVG

Κάπου εδώ ας αναφέρουμε για το Scalable Vector Graphics (SVG), ότι είναι μια μορφή διανυσματικής εικόνας με βάση την επεκτάσιμη γλώσσα σήμανσης (XML) για δισδιάστατα γραφικά με υποστήριξη για διαδραστικότητα και κινούμενα σχέδια. Είναι ένα ανοιχτό πρότυπο που έχει αναπτυχθεί από το 1999 και χρησιμοποιεί CSS για τροποποιήσεις του στυλ και JavaScript για scripting. Όλα τα μεγάλα σύγχρονα προγράμματα περιήγησης ιστού - συμπεριλαμβανομένων των Mozilla Firefox, Internet Explorer, Google Chrome, Opera, Safari και Microsoft Edge - έχουν υποστήριξη απόδοσης SVG. Το ωραίο με τη χρήση του SVG για τη δημιουργία χαρτών είναι ότι σημαίνει πως πολλές από τις ίδιες επιλογές στυλ που

μπορούν να εφαρμοστούν σε έγγραφα ιστού με CSS θα λειτουργήσουν επίσης και για τη σχεδίαση των επιθυμητών χαρτών.

4.3. Λειτουργίες εφαρμογής

Οι περισσότερες από τις λειτουργίες του κώδικα της εφαρμογής που σχετίζονται με τα δεδομένα, την παραλαβή και την επεξεργασία τους, είναι προγραμματισμένες να λειτουργούν ασύγχρονα. Αυτή η επιλογή έγινε λόγω του μεγάλου όγκου των δεδομένων και τυχόν καθυστερήσεων που μπορεί να υπάρξουν. Σκοπός είναι η ομαλότητα και λειτουργικότητα του προγράμματος. Έτσι, σε οποιαδήποτε τυχόν καθυστέρηση υπάρξει, δεν θα τερματιστεί απότομα - λόγω σφάλματος - η λειτουργία του προγράμματος. Επίσης, στο χρονικό διάστημα που θα υπάρχει αυτή η αναμονή, οι πόροι θα μπορούν να χρησιμοποιηθούν από άλλα κομμάτια και μόλις υπάρξει ανταπόκριση - απάντηση από το εκκρεμών αίτημα η λειτουργία θα μπορεί να συνεχιστεί ξανά από το σημείο που βρισκόταν.

4.3.1. Προβολή θέσης καραβιών σε πραγματικό χρόνο

Η κυριότερη, ουσιαστικά, λειτουργία της εφαρμογής είναι η προβολή της θέσης των καραβιών και η συνεχόμενη ανανέωση της θέσης τους, ώστε να παρέχεται παρακολούθηση σε πραγματικό χρόνο. Ταυτόχρονα με αυτό προβάλλεται και η πορεία κατεύθυνσης τους. Για τον σκοπό αυτόν έχουν δημιουργηθεί τέσσερις κύριες συναρτήσεις.

Αρχικά τα δεδομένα καλούνται ασύγχρονα μέσα από την `async function getDataOfShips()`, στέλνοντας αίτημα μέσω του API και αποθηκεύοντας την απάντηση που θα πάρει σε μια μεταβλητή (`response`). Έπειτα χρησιμοποιώντας την μέθοδο `json()`, όλη η ροή απόκρισης που λάβαμε επεξεργάζεται και αποθηκεύεται ως JSON σε μια άλλη μεταβλητή (`Data`). Τέλος για κάθε ένα από τα δεδομένα που παραλάβαμε, δημιουργείται μέσα από την μέθοδο `L.latLng` της Leaflet ένα αντικείμενο που αντιπροσωπεύει ένα γεωγραφικό σημείο με το δεδομένο γεωγραφικό πλάτος και μήκος που έχει ήδη το καθένα. Ακολουθεί το αντίστοιχο κομμάτι κώδικα.

```
112 // Get the ship data from the api
113 async function getDataOfShips() {
114     const response = await fetch(api_url);
115     Data = await response.json();
116
117     Data.forEach(function (d) {
118         d.latLong = new L.LatLng(d.latitude, d.longitude);
119     });
120 }
```

Εικ. Ε. Συνάρτηση για την παραλαβή των δεδομένων

Μετά από αυτό ακολουθεί η στιγμή που τα πλοία χρειάζεται να απεικονιστούν στο χάρτη, κάτι που γίνεται μέσα από την συνάρτηση `async function initializeShips()`. Εφόσον πάρει με την σειρά της τα δεδομένα από την προηγούμενη συνάρτηση, δημιουργεί τα εικονίδια για όλα τα πλοία και τα αποθηκεύει σε μια μεταβλητή (`ships`). Η δημιουργία ακολουθεί τις διαδικασίες της D3 όπου καθορίζονται, τι θα δημιουργηθεί (`path`, `text`, `image`, κλπ), το dataset το οποίο θα εισάγει και όλα τα χαρακτηριστικά σχετικά με το εικονίδιο. Όπως για παράδειγμα η προέλευση της εικόνας, το μέγεθος, το `index` για το επίπεδο του χάρτη στο οποίο θα βρεθεί, κρατείται το `mmsi` για μετέπειτα χρήση και «επισυνάπτονται» δύο συναρτήσεις οι οποίες θα κληθούν αντίστοιχα μόλις ο χρήστης πατήσει πάνω στο εικονίδιο (`click`) ή περάσει τον κέρσορα από πάνω (`mouseover`).

Τέλος, ένα πολύ σημαντικό σημείο της συνάρτησης είναι εκείνο το οποίο προστίθεται ένα χαρακτηριστικό μεταμόρφωσης (`transform`) το οποίο είναι αυτό που ευθύνεται για την σωστή προβολή και περιστροφή του εικονιδίου. Χρησιμοποιώντας την μέθοδο προβολής `map.latLngToLayerPoint()` - με παράμετρο τα αντικείμενα που φτιάξαμε παραπάνω (`latLong`) - οι συντεταγμένες μετατρέπονται σε σημεία σε επίπεδο χάρτη και ταυτόχρονα ορίζουμε τον αριθμό - σε μοίρες - της επιθυμητής περιστροφής, το οποίο το έχουμε από τα AIS δεδομένα (`cog`). Ακολουθεί η συνάρτηση.

```
122 // Function to display the ships
123 async function initializeShips() {
124     await getDataOfShips();
125
126     // Create the images for each ship on the map
127     ships = g
128         .selectAll("image")
129         .data(Data)
130         .enter()
131         .append("svg:image")
132         .attr("x", "-11.5px")
133         .attr("y", "-19.5px")
134         .attr("id", function (d) {
135             return d.mmsi;
136         })
137         .attr("class", "image")
138         .attr("xlink:href", "./ship-icon.png")
139         .style("z-index", 100000)
140         .attr("width", 25 + "px")
141         .attr("height", 25 + "px")
142         .attr("transform", function (d) {
143             let layerPoint = map.latLngToLayerPoint(d.latLong);
144             return (
145                 "translate(" +
146                 layerPoint.x +
147                 "," +
148                 layerPoint.y +
149                 ") rotate(" +
150                 d.cog +
151                 ")"
152             );
153         })
154         .on("mouseover", function (d) {
155             d3.select(this).style("cursor", "help");
156         })
157         .on("click", click);
```

Εικ. ΣΤ. Συνάρτηση δημιουργίας εικονιδίων για κάθε καράβι

Το επόμενο βήμα είναι, εφόσον από το χάρτη παρέχουμε στον χρήστη μεγέθυνση και σμίκρυνση, να φροντίσουμε κάθε φορά που θα αλλάζει το zoom του χάρτη, τα πλοία να επανατοποθετούνται στην σωστή τους θέση. Για το σκοπό αυτό της ακρίβειας, δημιουργήθηκε η συνάρτηση `drawAndUpdateShips()`. Κάθε φορά που υπάρχει μετακίνηση στο χάρτη επιλέγει, ένα ένα, όλα τα πλοία που υπάρχουν στο `svg` και ουσιαστικά είναι σαν να τα ξανασχεδιάζει κάνοντας αυτά που έγιναν και στην προηγούμενη συνάρτηση. Αυτή την φορά όμως το μόνο που τροποποιεί είναι τα χαρακτηριστικά του `transform` (θέση και περιστροφή). Παράλληλα δίνει και ένα χαρακτηριστικό `id` στο κάθε πλοίο, το οποίο το ορίζει με αριθμό το `mmsi` του, για μετέπειτα χρήση. Η συγκεκριμένη συνάρτηση δεν είναι ασύγχρονη καθώς δεν «περιμένει» τα δεδομένα αλλά τροποποιεί τα ήδη υπάρχοντα.

```
160 // Corrections on map movement
161 map.on("moveend", drawAndUpdateShips);

169 // Set id for the ship and positioning correctly on map
170 function drawAndUpdateShips() {
171   Data.forEach((d) => {
172     let ship = svg.select("g").selectAll("[id='" + d.mmsi + "']");
173     d.latLong = new L.LatLng(d.latitude, d.longitude);
174     ship
175       .attr("transform", function () {
176         let layerPoint = map.latLngToLayerPoint(d.latLong);
177         return (
178           "translate(" +
179             layerPoint.x +
180             "," +
181             layerPoint.y +
182           ") rotate(" +
183             d.cog +
184           ")"
185         );
186       });
187   });
188 }
```

Εικ. Ζ. Συνάρτηση ανανέωσης εικονιδίων στις μετακινήσεις του χάρτη

Τέλος, το μόνο που έμεινε σχετικά με αυτήν την λειτουργία είναι η ανανέωση των θέσεων των καραβιών ώστε να υπάρχει το γνώρισμα του πραγματικού χρόνου. Η ανανέωση έχει προγραμματιστεί να γίνεται κάθε 30" καλώντας την `updateData()`. Σε αυτήν την συνάρτηση γίνεται ένας συνδυασμός των παραπάνω, τροποποιώντας και πάλι μόνο το χαρακτηριστικό (`attribute`) που αφορά την θέση και την κατεύθυνση των πλοίων. Κάθε 30 δευτερόλεπτα «τραβάει» νέα δεδομένα για τα πλοία, τα αποθηκεύει, ξαναφτιάχνει αντικείμενα για το γεωγραφικό τους σημείο και «μεταμορφώνει» για κάθε καράβι το σημείο του σε επίπεδο χάρτη και την κατεύθυνση ανανεώνοντας το.

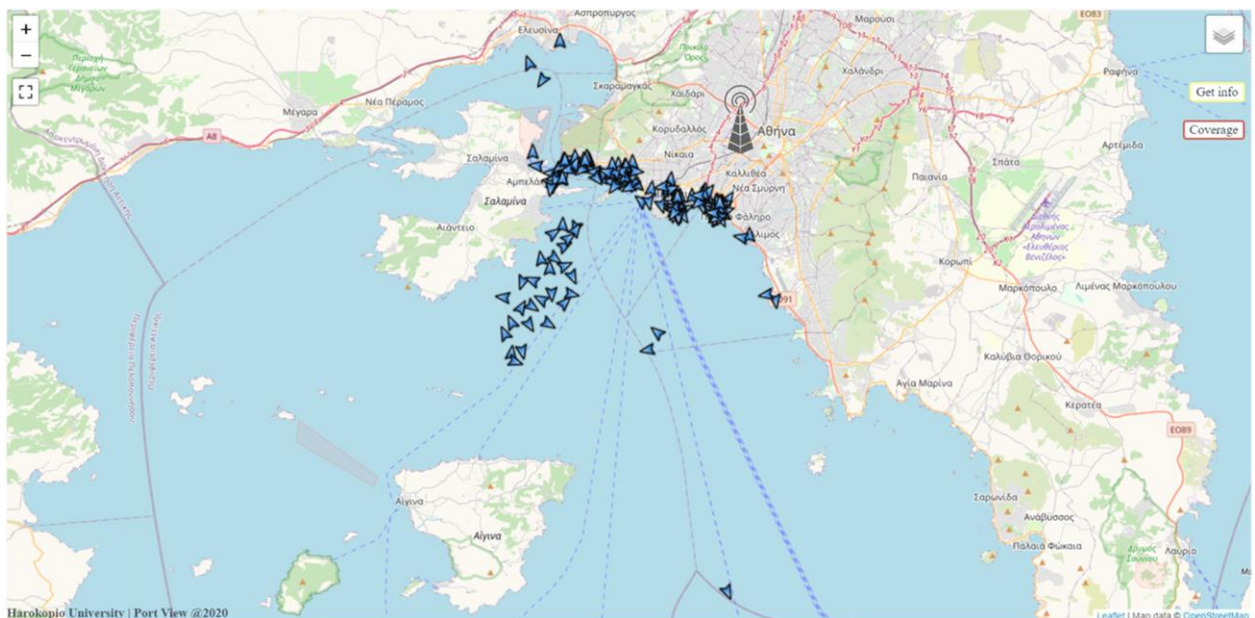
```

190 // Get current data and transform-update with the new latLong
191 async function updateData() {
192   console.log('updated!');
193   const response = await fetch(api_url);
194   Data = await response.json();
195
196   Data.forEach((d) => {
197     let ship = svg.select("g").selectAll("[id='" + d.mmsi + "'");
198     d.latLong = new L.LatLng(d.latitude, d.longitude);
199     ship
200       .attr("transform", function () {
201         let layerPoint = map.latLngToLayerPoint(d.latLong);
202         return (
203           "translate(" +
204             layerPoint.x +
205             "," +
206             layerPoint.y +
207             ") rotate(" +
208               d.cog +
209             ")"
210         );
211       });
212   });
213 }

```

Εικ. Η. Συνάρτηση ανανέωσης δεδομένων χάρτη

Το τελικό αποτέλεσμα που εμφανίζεται στον χρήστη μέσα από την εφαρμογή, σχετικά με όλα τα παραπάνω, απεικονίζεται στο παρακάτω στιγμιότυπο οθόνης.



Εικ.10. Λειτουργία προβολής θέσης καραβιών

4.3.2. Προβολή εξατομικευμένων πληροφοριών καραβιού

Όταν ο χρήστης πατήσει πάνω σε ένα εικονίδιο καραβιού, καλεί έμμεσα την μία από τις δύο συναρτήσεις που έχουν οριστεί για αλληλεπίδραση με το ποντίκι. Αυτή είναι η function `click(d, i)`, η οποία εμφανίζει

ένα μικρό παραθυράκι με πληροφορίες (**tooltip**) για το συγκεκριμένο καράβι και δύο κουμπιά για περαιτέρω χρήσεις. Οι πληροφορίες αφορούν την μέρα και ώρα που λήφθηκε το σήμα (**timestamp**), το γεωγραφικό μήκος και πλάτος, την ταχύτητα και τον προορισμό του πλοίου. Αυτό γίνεται δυνατό περνώντας αυτές τις πληροφορίες ως παραμέτρους από συνάρτηση σε συνάρτηση μέσα από τις αλυσιδωτές κλήσεις και εμφανίζοντας τα με την μέθοδο **html()**. Παράλληλα, από τα κουμπιά παρέχεται η επιλογή της εμφάνισης της διαδρομής που έχει διανύσει το πλοίο μέχρι εκείνη την στιγμή ή να κλείσει το παράθυρο με τις πληροφορίες. Το καθένα θα καλέσει την αντίστοιχη κατάλληλη συνάρτηση με το πάτημα του, έχοντας χρησιμοποιήσει την **d3.select()** για αυτήν την σύνδεση.

Επίσης, χρησιμοποιούνται δύο **transition()** που προσφέρει η D3 για το κομμάτι της οπτικοποίησης. Έτσι όταν το καράβι επιλέγεται γίνεται κατά το ήμισυ διαφανές, αλλά και το παραθυράκι του πλοίου καθυστερεί ελαφρά να εμφανιστεί χάριν ομαλότητας (**easy-in**). Η μέθοδος **duration()** που πραγματοποιεί την καθυστέρηση παίρνει ως παράμετρο **milliseconds**. Τέλος, με την μέθοδο **style()** προσδιορίζεται σε πιο σημείο ακριβώς θα εμφανίζεται το παραθυράκι σε σχέση με το εικονίδιο του πλοίου. Ακολουθεί το αντίστοιχο κομμάτι κώδικα.

```
214 // Functions for mouse interaction
215 function click(d, i) {
216     // "Focus" on the selected ship
217     d3.select(this).transition().duration(50).attr("opacity", "0.5");
218     // Makes the div appear on click
219     tooltip.transition().duration(750).style("opacity", 1);
220
221     tooltip
222         .html(
223             "<button id='but1' onclick='getRouteData()'>Route</button>" +
224             "<button id='but2'>Close</button>" +
225             "<br>" +
226             d.timestamp +
227             "<br>" +
228             d.ship_name +
229             "<br>" +
230             "Longitude: " +
231             d.longitude.toFixed(6) +
232             "°<br>" +
233             "Latitude: " +
234             d.latitude.toFixed(6) +
235             "°<br>" +
236             "SOG: " +
237             d.sog +
238             "kn<br>" +
239             "Destination: " +
240             d.destination
241         )
242         .style("left", d3.event.pageX + 12 + "px")
243         .style("top", d3.event.pageY - 28 + "px");
244
245     d3.select("#but1").on("click", function () {
246         getRouteData(d.mmsi);
247     });
248     d3.select("#but2").on("click", mouseout);
249 }
```

Εικ. Θ. Συνάρτηση για mouse interaction

Για να πραγματοποιηθεί ωστόσο όλο αυτό, έχει οριστεί από την αρχή του κώδικα η μεταβλητή `tooltip` και μέσω της D3 έχει δημιουργηθεί και εκχωρηθεί στην μεταβλητή ένα μικρό τμήμα (`div`). Αυτό είναι αρχικά ορισμένο να μην είναι ορατό και γενικά δεν έχει καμία επίδραση στο περιεχόμενο μέχρι να διαμορφωθεί με χρήση CSS.

Με παρόμοιο τρόπο έχουν δημιουργηθεί και τροποποιηθεί στιλιστικά και τα κουμπιά του χάρτη, όπως το κουμπί με το οποίο προβάλλεται η κάλυψη της κεραίας. Σε αυτό, επιλέγεται ο χάρτης ως σημείο με το οποίο θα «συνδεθεί» και θα εμφανιστεί, εκχωρείται ένα `button`, επιλέγεται το κείμενο που θα υπάρχει πάνω σε αυτό και χαρακτηριστικά όπως το επίπεδο του χάρτη στο οποίο θα βρεθεί ή η ομάδα στυλ στην οποία θα ανήκει. Τέλος, προσδίδεται η λειτουργία η οποία θα ξεκινήσει με το που πατηθεί το κουμπί από τον χρήστη και σε αυτήν την περίπτωση είναι η συνάρτηση `showCoverage`. Ακολουθεί δείγμα από τον κώδικα για την δημιουργία του κουμπιού στον χάρτη.

```
85 // Map button to display the antenna's coverage
86 var buttonA = d3
87   .select("#map")
88   .append("button")
89   .text("Coverage")
90   .attr("class", "buttonA")
91   .style("z-index", "10003")
92   .on("click", showCoverage);
93
```

Εικ. Ι. Δημιουργία D3 button

Παρακάτω ακολουθεί ο κώδικας της δημιουργίας για το tooltip που αναφέρθηκε παραπάνω, το απόσπασμα από τις τροποποιήσεις του στυλ του με CSS που έγιναν σε αυτήν την περίπτωση σχετικά με την θέση του, το σχήμα, την γραμματοσειρά, τα χρώματα, την ομαλή μετάβαση του όταν εμφανίζεται κλπ. Έπειτα, υπάρχει η εικόνα από το στιγμιότυπο του τελικού αποτελέσματος, έχοντας πατήσει ενδεικτικά ένα καράβι από τα διαθέσιμα στον χάρτη.

```
94 // On-click ship info
95 var tooltip = d3
96   .select("#map")
97   .append("div")
98   .style("opacity", 0)
99   .attr("class", "tooltip")
100  .style("z-index", "10001");
```

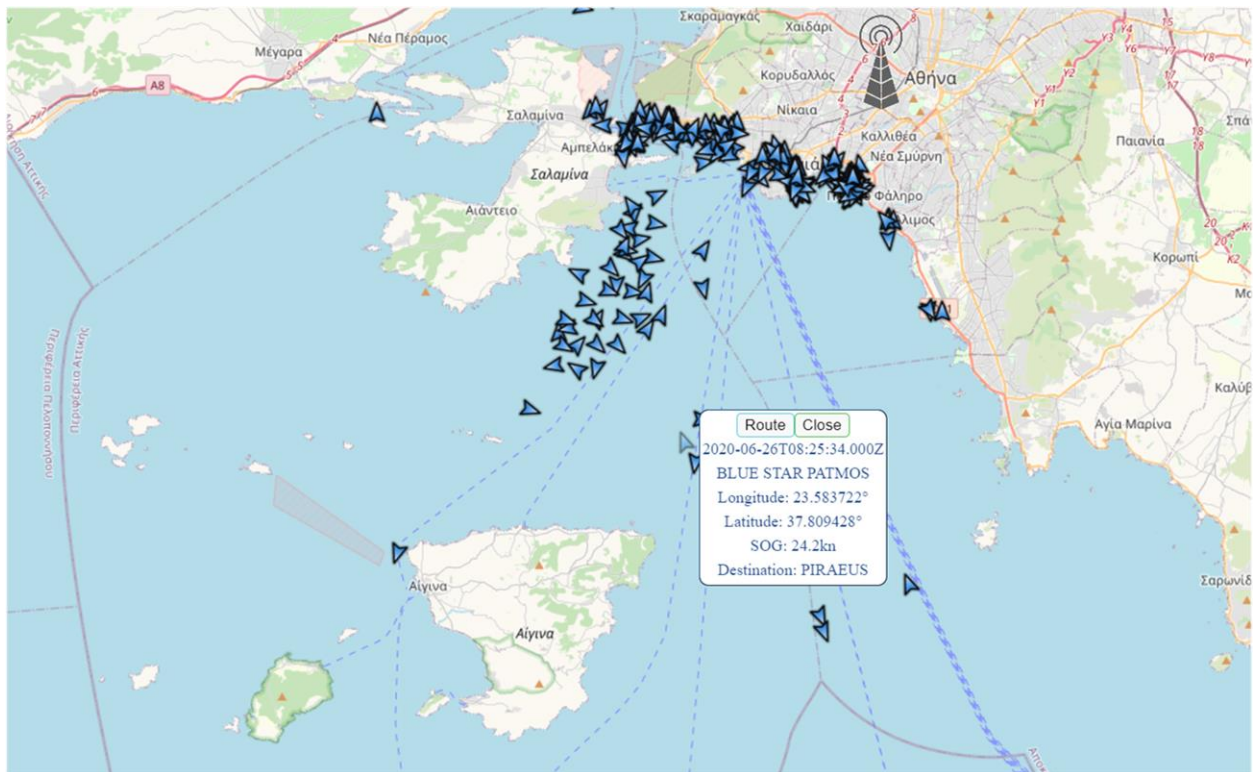
Εικ. Κ. Δημιουργία D3 tooltip

```

99  .tooltip {
100    position: absolute;
101    display: inline-block;
102    text-align: center;
103    width: auto;
104    height: auto;
105    padding: 2px;
106    font-size: 14px;
107    color: #194182;
108    background: whitesmoke;
109    border: #194182;
110    border-radius: 8px;
111    border-style: groove;
112    border-width: 1px;
113    transition: ease-in-out;
114    font-family: 'Times New Roman';
115    z-index: 10000;
116  }
117  .tooltip > svg {
118    margin: auto;
119  }

```

Εικ. Α. CSS tooltip styling



Εικ.11. Προβολή πληροφοριών καραβιού στον χρήστη

4.3.3. Προβολή θαλάσσιας διαδρομής επιλεγμένου καραβιού

Όταν ο χρήστης πατήσει το κουμπί με την επιλογή Route για να δει την διαδρομή που έχει διανύσει μέχρι τώρα το πλοίο καλείται η συνάρτηση `async function getRouteData(mmsi)`. Στην αρχή της συνάρτησης απομακρύνονται τυχόν πληροφορίες που υπάρχουν από άλλα καράβια. Έπειτα, έχοντας ως παράμετρο το `mmsi` δημιουργεί ένα url όπως έχει προαναφερθεί για να αποκτήσει πρόσβαση στα δεδομένα του συγκεκριμένου πλοίου. Για τους χρονολογικούς υπολογισμούς χρησιμοποιείται η βοήθεια της `Date.now()` -επιστρέφει την τωρινή ώρα σε UTC- και ως χρονικό περιθώριο επιλέγονται τα δεδομένα των τελευταίων τεσσάρων ωρών. Στο τέλος, με τα δεδομένα που έχει συλλέξει ως παράμετρο καλεί την `drawRoute(newData)` για να σχεδιάσει την γραμμή της διαδρομής.

```
251 // Create the url to access route information of the selected ship
252 async function getRouteData(mmsi) {
253     // To remove current displayed routes
254     d3.selectAll('.images').remove();
255     d3.select('.line').remove();
256
257     console.log(mmsi);
258     currentDate = 0;
259     minus4H = 0;
260     currentDate = Math.floor(Date.now() / 1000);
261     minus4H = currentDate - 14400;
262     let url = "http://83.212.240.62:3000/maritime/api/" + minus4H + "/" + currentDate + "/" + mmsi;
263     console.log(url);
264
265     const response = await fetch(url);
266     const newData = await response.json();
267     console.log(newData);
268     drawRoute(newData);
269 }
```

Εικ. Μ. Συνάρτηση για συλλογή δεδομένων ενός πλοίου

Με την εκτέλεση της `function drawRoute(newData)` το παράθυρο πληροφοριών που είναι ήδη ορατό μεταβαίνει αριστερά της οθόνης ώστε να μην εμποδίζει στην προβολή της διαδρομής. Σε περίπτωση που για κάποιο λόγο δεν υπάρχουν δεδομένα για τα σήματα ή υπάρχει μόνο ένα γιατί το πλοίο μόλις εισήλθε στην εμβέλεια της κεραίας, εμφανίζονται τα κατάλληλα μηνύματα στον χρήστη. Στην συνέχεια δημιουργούνται με παρόμοιο τρόπο όπως και για τα αρχικά καράβια, εικονίδια για κάθε ένα από τα σήματα του πλοίου που έχουμε λάβει. Όλα αυτά τα εικονίδια-σήματα με την βοήθεια της D3 μεθόδου `d3.svg.line()` και της συνάρτησης `displayLine(newData)` συνδέονται με μια γραμμή. Δημιουργείται η μεταβλητή `lineGenerator` η οποία υπολογίζει τα σημεία της x και y με βάση το γεωγραφικό μήκος και πλάτος του κάθε σήματος. Έπειτα με την `function displayLine(newData)` ορίζεται και πάλι μια μεταβλητή `line` η οποία με την δομή της `lineGenerator`, τα σήματα που συλλέχθηκαν -`datum(newData)`- και μερικές μορφοποίησης της γραμμής για την εμφάνιση, όπως το εφέ σχεδιάσής της, προβάλλει την διαδρομή. Ακολουθούν σημαντικά αποσπάσματα της συνάρτησης.

```

290 images = d3.select("#map")
291   .select("svg")
292   .selectAll("images")
293   .data(newData)
294   .enter()
295   .append("svg:image")
296   .attr("x", "-11.5px")
297   .attr("y", "-19.5px")
298   .attr("class", "images")
299   .attr("xlink:href", "./highlight.png")
300   .attr("width", 24 + "px")
301   .attr("height", 24 + "px")
302   .attr("transform", function (d) {
303     let layerPoint = map.latLngToLayerPoint(d.latLng);
304     return (
305       "translate(" +
306         layerPoint.x +
307         "," +
308         layerPoint.y +
309         ") rotate(" +
310         d.cog +
311         ")"
312     );
313   })
314   .attr("fill-opacity", 0.5);
315
316 let lineGenerator = d3.svg.line()
317   .x(function (d) { return map.latLngToLayerPoint([d.latitude, d.longitude]).x })
318   .y(function (d) { return map.latLngToLayerPoint([d.latitude, d.longitude]).y })
319   .interpolate("monotone");

```

Εικ. Ν. Δημιουργία εικονιδίων για όλα τα στήγματα της διαδρομής

```

321 function displayLine(newData) {
322   line = d3.select("#map")
323     .select("svg")
324     .append("path")
325     .datum(newData)
326     .attr("class", "line")
327     .attr({
328       d: lineGenerator(newData),
329       fill: 'none',
330       stroke: "#fffd7a",
331       "stroke-width": "3px",
332       "shape-rendering": "geometricPrecision"
333     });
334
335   let totalLength = line.node().getTotalLength();
336
337   line.attr("stroke-dasharray", totalLength + " " + totalLength)
338     .attr("stroke-dashoffset", totalLength)
339     .transition()
340     .duration(4000)
341     .ease("linear")
342     .attr("stroke-dashoffset", 0);
343 }
344 displayLine(newData);

```

Εικ. Ξ. Δημιουργία γραμμής της διαδρομής ενός πλοίου

Φυσικά στο τέλος όλων αυτών υπάρχουν κατάλληλες συναρτήσεις για την ανανέωση όλων αυτών στις διάφορες μετακινήσεις του χάρτη και την σωστή επαναπροβολή τους. Καθώς και η συνάρτηση με την οποία όταν ο χρήστης πατήσει το κουμπί **Close** όλα τα στοιχεία και οι πληροφορίες για το καράβι που είχε επιλεγεί απομακρύνονται από τον χάρτη για να συνεχιστεί η πλοήγηση του χρήστη. Ακολουθεί το αποτέλεσμα που εμφανίζεται στον χρήστη.



Εικ.12. Προβολή διαδρομής επιλεγμένου καραβιού

4.3.4. Προβολή κάλυψης της εμβέλειας της κεραίας

Για την επίτευξη αυτής της λειτουργίας χρησιμοποιήθηκε ο αλγόριθμος **convex hull**. Οι αλγόριθμοι που κατασκευάζουν κυρτά κύττα διαφόρων αντικειμένων έχουν ένα ευρύ φάσμα εφαρμογών στα μαθηματικά και την επιστήμη των υπολογιστών. Το **convex hull** ενός συνόλου σημείων ορίζεται ως το μικρότερο κυρτό πολύγωνο, το οποίο περικλείει όλα τα σημεία του συνόλου. Ενώ, κυρτό σημαίνει ότι το πολύγωνο δεν έχει γωνία που κάμπτεται προς τα μέσα. Ο αλγόριθμος κατασκευάζει το **convex hull** ενός συνόλου 2-διαστάσεων σημείων σε χρόνο $O(n \log n)$. Αυτό το κάνει ταξινομώντας τα σημεία λεξικογραφικά (πρώτα με συντεταγμένη το x , και σε περίπτωση ισοπαλίας, με συντεταγμένη το y) και στη συνέχεια, κατασκευάζοντας άνω και κάτω κύτος (**hull**) των σημείων στο $O(n)$ χρόνο. Τα σημεία που θα ταξινομηθούν θα πρέπει να είναι από τρία και πάνω.

Ακολουθεί παράδειγμα ψευδοκώδικα για το πώς λειτουργεί ο αλγόριθμος **convex hull**.

Input: a list P of points in the plane.

Precondition: There must be at least 3 points.

```
Sort the points of P by x-coordinate (in case of a tie, sort by y-coordinate).
```

```
Initialize U and L as empty lists.
```

```
The lists will hold the vertices of upper and lower hulls respectively.
```

```
for i = 1, 2, ..., n:
```

```
    while L contains at least two points and the sequence of last two points  
        of L and the point P[i] does not make a counter-clockwise turn:  
        remove the last point from L  
    append P[i] to L
```

```
for i = n, n-1, ..., 1:
```

```
    while U contains at least two points and the sequence of last two points  
        of U and the point P[i] does not make a counter-clockwise turn:  
        remove the last point from U  
    append P[i] to U
```

```
Remove the last point of each list (it's the same as the first point of the  
other list).
```

```
Concatenate L and U to obtain the convex hull of P.
```

```
Points in the result will be listed in counter-clockwise order.
```

Εικ. 0. Δείγμα ψευδοκώδικα για την λειτουργία του convex hull

Η βασική συνάρτηση είναι η `async function showCoverage()` στην οποία και πάλι δημιουργείται ένα συγκεκριμένο `urlA` καθώς θέλουμε να συγκεντρώσουμε τα σήματα από όλα τα καράβια που λήφθηκαν τις τελευταίες τέσσερις ώρες. Σε όλη την επεξεργασία των δεδομένων μέχρι την κατάληξη στα τελικά που χρησιμοποιήθηκαν για να σχηματίσουν το πολύγωνο χρησιμοποιήθηκαν και βοηθητικές μαθηματικές συναρτήσεις όπως `function comparison(a, b)` και `function cross(a, b, o)`. Επίσης, ήταν επιθυμητό το φιλτράρισμα των δεδομένων που είχαν απόσταση μεγαλύτερη από 250 km και η απόρριψη αυτών. Η λογική πίσω από αυτό είναι ότι υπάρχει περίπτωση να λάβει η κεραία σήματα από κάπου μακριά, είτε λόγω λανθασμένης μετάδοσης, είτε γιατί καμιά φορά υπό καλές συνθήκες μπορεί όντως η κεραία να λάβει στίγματα από τόσο μακριά. Σε κάθε περίπτωση όμως αυτές οι εξαιρέσεις δεν είναι αντιπροσωπευτικές της κάλυψης της κεραίας, οπότε και αφαιρούνται από τους υπόλοιπους υπολογισμούς. Σε αυτό βοήθησε η επίσης γνωστή φόρμουλα `haversine`, μέσα από την συνάρτηση `function getHaversineDistance(Ulong, Ulat, longitude, latitude)`. Ο τύπος `haversine` καθορίζει την απόσταση του μεγάλου κύκλου μεταξύ δύο σημείων σε μια σφαίρα, δεδομένων των γεωγραφικών μήκους και γεωγραφικών πλάτους τους και είναι πάρα πολύ σημαντικός στην πλοήγηση.

Με αυτόν τον τρόπο καταλήξαμε στο τελικό dataset των δεδομένων που θα χρησιμοποιηθούν τα οποία αποθηκεύονται σε μια μεταβλητή `hull` και έπειτα με την μέθοδο `map()` φτιάχνονται arrays με τα δεδομένα για κάθε στοιχείο. Στην συνέχεια, καθώς η επιθυμητή μορφή για την συνέχεια των δεδομένων ήταν GeoJSON, γίνεται η μετατροπή τους σύμφωνα ανάλογη σύνταξη. Με αυτά τα δεδομένα πια, χρησιμοποιούνται οι συναρτήσεις `d3.geo.transform()` και `d3.geo.path()` για το πώς θα λειτουργήσει το `path` το οποίο θα «μεταφράζει» τα GeoJSON κατάλληλα ώστε να προβληθούν στο svg. Φυσικά υπάρχουν και πάλι μέσω της D3 η ανάθεση χαρακτηριστικών στο path για την δημιουργία του σε οπτικό Σύστημα ζωντανής παρακολούθησης ναυτιλιακής κίνησης από επίγειο δέκτη

επίπεδο (`d3_features`) αλλά και η κατάλληλη συνάρτηση για την σωστή επανατοποθέτηση του πολυγώνου κάθε φορά που υπάρχει μετακίνηση στον χάρτη. Αυτό, στο συγκεκριμένο σημείο της εφαρμογής, γίνεται μέσω της συνάρτησης `reset()`. Τέλος στην `projectPoint(x,y)` ορίζεται, για να μπορεί να χρησιμοποιηθεί στον ορισμό του `transform`, ο τρόπος προβολής ενός σημείου με δύο συντεταγμένες για να επιτευχθεί η σύνδεση της D3 με την Leaflet. Ακολουθούν κομμάτια από την συνάρτηση και όσα αναφέρθηκαν σε αυτήν την παράγραφο και το τελικό αποτέλεσμα που θα εμφανιστεί στον χρήστη.

```
466     const hull = convexHull(coords);
467
468     let data = [];
469     hull.map(function (d) {
470         data.push([d[1], d[0]]
471     });
472
473     console.log(data);
474
475     let geoData = {
476         type: "FeatureCollection", features: [{
477             type: "Feature",
478             geometry: {
479                 type: "Polygon",
480                 coordinates: [data]
481             }
482         }]
483     };
484     console.log(geoData);
485
486     // Create a d3.geo.path to convert GeoJSON to SVG
487     let transform = d3.geo.transform({ point: projectPoint }),
488         path = d3.geo.path().projection(transform);
489
490
491     // Create path elements for each of the features
492     d3_features = g.selectAll("path")
493         .data(geoData.features)
494         .enter()
495         .append("path")
496         .attr("class", "polygon")
497         .attr("z-index", 500);
498
499     map.on("moveend", function () {
500         reset()
501     });
```

Εικ. Π. Χρήση convex hull και δημιουργία D3 path

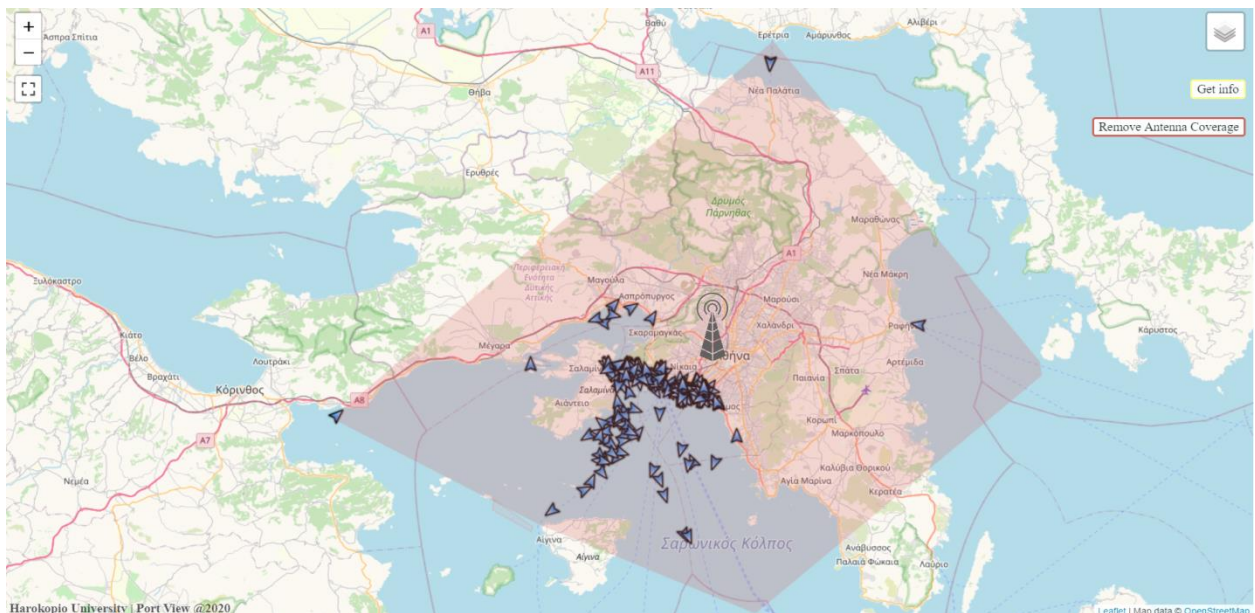
```

503     reset();
504     // Fit the SVG element to leaflet's map layer
505     function reset() {
506
507         d3_features.attr("d", path)
508             .style("fill-opacity", 0.23)
509             .attr('fill', '#d15858')
510             .attr("z-index", 500);
511     }
512     // Use Leaflet to implement a D3 geometric transformation.
513     function projectPoint(x, y) {
514         let point = map.latLngToLayerPoint(new L.LatLng(y, x));
515         this.stream.point(point.x, point.y);
516     }

```

Εικ. Ρ. Συνάρτηση ανανέωσης path και ορισμός προβολής σημείων

Πατώντας το κουμπί Remove Antenna Coverage ο χρήστης μπορεί να κλείσει την προβολή της κάλυψης. Γενικότερα όλες οι λειτουργίες της εφαρμογής μπορούν να χρησιμοποιηθούν παράλληλα, όπως π.χ. η προβολή της εμβέλειας της κεραίας με την προβολή πληροφοριών για οποιοδήποτε καράβι είναι επιθυμητό.



Εικ.13. Προβολή κάλυψης της εμβέλειας της κεραίας

Στο τέλος του κώδικα καλείται η συνάρτηση με την οποία αρχίζει να λειτουργεί όλη η εφαρμογή.

```

527     initializeShips();

```

5. Συμπεράσματα

Ολοκληρώνοντας αυτήν την εργασία και κάνοντας μια μικρή ανακεφαλαίωση, υπάρχει μια αίσθηση χαράς και πληρότητας. Το εγχείρημα της δημιουργίας μιας εφαρμογής για την παρακολούθηση σκαφών στέφθηκε με επιτυχία. Αυτή η εφαρμογή είναι λειτουργική και μπορεί να εκπληρώσει τον σκοπό της: να βοηθήσει τον χρήστη της να έχει απαραίτητες πληροφορίες για το καράβι που τον ενδιαφέρει, όπως να δει την θέση του, την διαδρομή του, τα χαρακτηριστικά του, είτε δυναμικά είτε στατιστικά, την εμβέλεια της κάλυψης της περιοχής και άρα μέχρι ποιο σημείο μπορεί να εξυπηρετηθεί από την εφαρμογή. Επιπλέον, μπορεί να χρησιμοποιηθεί και σε περιπτώσεις έρευνας, διάσωσης και διερεύνησης ατυχημάτων, στην προστασία υποδομών αλλά και θαλάσσιων προστατευόμενων περιοχών και, σίγουρα, στην παρατήρηση και στον έλεγχο της κυκλοφορίας των σκαφών. Όλα αυτά με απλότητα και αποτελεσματικότητα.

Η είσοδος στον θαλάσσιο κόσμο της τεχνολογίας, έχει μεγάλο ενδιαφέρον. Φυσικά, χρειάζεται αρκετή μελέτη, διαρκή ενημέρωση, επιμονή και αρκετή υπομονή, ειδικά στην περίπτωση που αναπτύσσεις μια εφαρμογή σε αυτόν τον τομέα, η οποία υπόσχεται να παρέχει κάποιες υπηρεσίες, στις οποίες θα βασιστεί ένας τρίτος. Παράλληλα με αυτό, η επιθυμία της υλοποίησης της εφαρμογής με δύο διαφορετικές βιβλιοθήκες JavaScript έφερε στο προσκήνιο διαφορετικά τεχνικά θέματα, π.χ. ποιες εκδόσεις κάθε βιβλιοθήκης λειτουργούν βέλτιστα μαζί ή πώς να καταναμεηθούν σωστά τα layers του χάρτη. Το σημαντικότερο από όλα αυτά όμως, όπως και με κάθε δυσκολία – πρόβλημα, είναι ότι, μόλις το λύσεις, αυτόματα έχεις εμπλουτίσει τις γνώσεις σου και την εμπειρία σου.

Στην δική μου περίπτωση, επιπρόσθετα σε όλα αυτά θεώρησα μεγάλο όφελος την ενασχόληση μου με την D3, το οποίο, αν και χαρακτηρίζεται δύσκολο να μπεις στον «κόσμο» του, μπορεί να προσφέρει πολύ δυναμικά στοιχεία στο κομμάτι του data visualization. Ακόμα, λόγω του μεγάλου χρονικού διαστήματος ενασχόλησης μου και πολλών ωρών μελέτης, κατάφερα να εξασκήσω περαιτέρω τις ικανότητες μου για συνδυαστική αλλά και κριτική σκέψη. Τέλος, είδα ένα κομμάτι του προγραμματισμού που μου κίνησε το ενδιαφέρον και σίγουρα απόλαυσα την προσωπική ικανοποίηση της κατάκτησης του στόχου μου.

Το μόνο που μένει να σκεφτείς μετά από όλα αυτά είναι ίσως τι θα μπορούσες να κάνεις ακόμη σε περίπτωση που είχες άπειρο χρόνο. Το λεγόμενο future work. Κάποιες ενδεικτικές λειτουργίες που θα μπορούσαν να προστεθούν είναι οι ακόλουθες: προσθήκη διαφορετικών εικονιδίων ή διαφορετικού χρώματος για κάθε πλοίο ανάλογα με τον τύπο (επιβατηγό, φορτηγό, δεξαμενόπλοιο κλπ), αναζήτηση συγκεκριμένου πλοίου με το όνομά του ή τον προορισμό του πληκτρολογώντας το σε ένα πλαίσιο αναζήτησης, προσθήκη μπάρας με χρονολογική σήμανση συνδεδεμένη με τα AIS δεδομένα. Συγκεκριμένα σε αυτήν την εκδοχή, ο χρήστης θα μπορούσε μετακινώντας την μπάρα του χρόνου προς τα πίσω να δει τον χάρτη και τις θέσεις των πλοίων σε προηγούμενο χρόνο, δίνοντας πρόσβαση σε στοιχεία μέχρι ένα συγκεκριμένο παρελθοντικό όριο.

6. Github

Project: <https://github.com/EDpanta/marine-surveillance>

Βιβλιογραφία

1. https://www.nauticast.com/at/cms/about_ais, Information on AIS.
2. http://www.eugenfound.edu.gr/appdata/documents/books_pdf/e_j00086.pdf
3. <https://help.marinetraffic.com/hc/en-us/>
4. <http://www.imo.org/en/OurWork/safety/navigation/pages/ais.aspx>, Regulations for carriage of AIS.
5. <https://www.nautinst.org>, The Nautical Institute."Automatic Identification System (AIS): A Human Factors Approach" (PDF)
6. https://el.wikipedia.org/wiki/JavaScript#Javascript_%CE%BA%CE%B1%CE%B9_Java
7. Fisher, Sharon (1989). ["OS/2 EE to Get 3270 Interface Early"](#). *Google Books*.
8. <https://leafletjs.com/>
9. <https://d3js.org/>
10. <https://www.navcen.uscg.gov/?pageName=aismain>, Automatic Identification System.
11. <http://www.allaboutais.com/index.php/en/>
12. <https://www.fleetmon.com/company/our-story/>
13. <https://www.vesselfinder.com/about>
14. https://www.vesseltracker.com/en/static/about_vesseltracker.html
15. <https://openlayers.org/>
16. <https://www.mapbox.com>
17. <https://developers.google.com/maps/documentation/>
18. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
19. <https://opensource.ellak.gr/2017/01/31/9-ergalia-web-mapping-anichtou-logismikou/>
20. <https://www.microsoft.com/en-us/maps/choose-your-bing-maps-api>

21. <http://polymaps.org/>
22. <https://developers.arcgis.com/>
23. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
24. <https://medium.com/@pascal.sommer.ch/a-gentle-introduction-to-the-convex-hull-problem-62dfcabee90c>
25. https://en.wikipedia.org/wiki/Application_programming_interface
26. <https://www.json.org/json-en.html>
27. <https://leanpub.com/leaflet-tips-and-tricks/read>
28. https://en.wikipedia.org/wiki/Haversine_formula
29. <https://developer.mozilla.org/el/>
30. Montewka, J., Kujala, P., Ylitalo, J., 2009. The quantitative assessment of marine traffic safety in the Gulf of Finland, on the basis of AIS data. ZeszytyNaukowe/AkademiaMorska w Szczecinie, pp. 105–115