



Τεχνολογικό
Πανεπιστήμιο
Κύπρου

Σχολή Μηχανικής και
Τεχνολογίας

Πτυχιακή εργασία

**Παρακολούθηση και Ανίχνευση Μετακινήσεων Πλοίων σε
Πραγματικό Χρόνο Χρησιμοποιώντας Δεδομένα AIS**

Χρίστος Χριστοδούλου

Λεμεσός, Μάιος 2022

ΤΕΧΝΟΛΟΓΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή εργασία

Παρακολούθηση και Ανίχνευση Μετακινήσεων Πλοίων σε
Πραγματικό Χρόνο Χρησιμοποιώντας Δεδομένα AIS

του

Χρίστος Χριστοδούλου

Επιβλέπων Καθηγητής

Δρ. Ηρόδοτος Ηροδότου

Λεμεσός, Μάιος 2022

Πνευματικά δικαιώματα

Copyright © Χρίστος Χριστοδούλου, 2022

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Τμήμα Ηλεκτρολογων Μηχανικων και Μηχανικων Ηλεκτρονικων Υπολογιστων και Πληροφορικης του Τεχνολογικού Πανεπιστημίου Κύπρου δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

Θα ήθελα να ευχαριστήσω ιδιαίτερα τον επιβλέποντα καθηγητή της πτυχιακής μου εργασίας, Δρ. Ηρόδοτο Ηροδότου για την καθοδήγηση και την επίβλεψη κατά τη διάρκεια του έργου. Επίσης θα ήθελα να ευχαριστήσω ιδιαίτερα όλους όσους ήταν δίπλα μου και με στηρίζανε κατά τη διάρκεια των σπουδών μου.

ΠΕΡΙΛΗΨΗ

Το Automatic Identification System(AIS) είναι ένα αυτόματο σύστημα παρακολούθησης πλοίων και υπηρεσιών κυκλοφορίας πλοίων. Οι πληροφορίες που παρέχονται από τον εξοπλισμό AIS περιλαμβάνουν, μεταξύ άλλων, τη θέση, την κατεύθυνση και την ταχύτητα των πλοίων, επιτρέποντας στις ναυτιλιακές αρχές να εντοπίζουν και να παρακολουθούν τις κινήσεις τους. Το AISafety είναι μια υπάρχουσα πλατφόρμα παρακολούθησης σκαφών που εφαρμόζεται στο Τεχνολογικό Πανεπιστήμιο Κύπρου, η οποία στέλνει ειδοποιήσεις για εισερχόμενα και εξερχόμενα πλοία από διάφορες περιοχές ενδιαφέροντος, καθώς και για ενδεχόμενες συγκρούσεις μεταξύ σκαφών. Αυτή η εργασία εισάγει τρεις βασικές βελτιώσεις στις τρέχουσες εσωτερικές λειτουργίες της πλατφόρμας. Η πρώτη περιλαμβάνει την αναβάθμιση της γραφικής διεπαφής ιστού έτσι ώστε το μέγεθος, η κατεύθυνση και η διαδρομή των πλοίων να εμφανίζονται στον χάρτη σε πραγματικό χρόνο και μέγεθος. Η δεύτερη βελτίωση είναι η βελτιστοποίηση του αλγορίθμου για τον εντοπισμό πλοίων που εισέρχονται και εξέρχονται από προκαθορισμένες περιοχές ενδιαφέροντος και να λαμβάνεται υπόψη για ελαττωματικά ή ανακριβή σήματα AIS. Η τελευταία βελτίωση σχετίζεται με τη βελτίωση της ακρίβειας του αλγόριθμου ανίχνευσης σύγκρουσης σκαφών λαμβάνοντας υπόψη τον χρόνο των σκαφών που πλησιάζουν το σημείο κλειστής συνάντησης.

Λέξεις κλειδιά: AIS δεδομένα, πλοία, αλγόριθμος σύγκρουσης, (προ)ειδοποιήσεις πλοίων

ABSTRACT

Automatic Identification System (AIS) is an automated monitoring system for ships and ship traffic services. The information provided by the AIS equipment includes, inter alia, the position, direction, and speed of ships, enabling the shipping authorities to locate and monitor their movements. AISafety is an existing vessel monitoring platform implemented at Cyprus University of Technology, which sends alerts for incoming and outgoing vessels from various areas of interest, as well as for any potential collisions between vessels. This work introduces three main enhancements to the current inner workings of the platform. The first one involves upgrading the graphical web interface so that the size, direction, and route of the ships are displayed on the map in real time and size. The second improvement is to optimize the algorithm for identifying ships entering and leaving predefined areas of interest and account for faulty or inaccurate AIS signals. The last enhancement relates to improving the accuracy of the vessel collision detection algorithm by taking into account the timing of the vessels approaching the point of closed encounter.

Keywords: AIS data, vessels, collision algorithm, vessel warnings

3.2.2	Προβολή ενός στιγμιότυπου του συστήματος στο παρελθόν.....	18
3.3	Βελτιστοποίηση αλγορίθμου εισόδου/εξόδου	19
3.3.1	Υλοποίηση	19
3.3.2	Εκτίμηση αλγορίθμου.....	20
3.3.3	Ψευδοκώδικας αλγορίθμου.....	21
4	Συζήτηση	30
5	Συμπεράσματα	31
6	Πίνακες και Διαγράμματα	32
6.1	Πίνακες.....	32
	ΕΠΙΛΟΓΟΣ	40
	ΒΙΒΛΙΟΓΡΑΦΙΑ	41
	ΠΑΡΑΡΤΗΜΑ Ι.....	42

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1 : Position Report A	32
Πίνακας 2: Position Report B	34
Πίνακας 3: Countries	35
Πίνακας 4: Navigation Status	35
Πίνακας 5: Vessel	35
Πίνακας 6: Entered /Left Area Warnings	36
Πίνακας 7: Collision Warnings.....	36
Πίνακας 8: Collision Destination.....	37
Πίνακας 9: Ship in areas	37
Πίνακας 10: Areas	37
Πίνακας 11: Area Types	38
Πίνακας 12: Ship categories	38
Πίνακας 13: Vessel Label A	38
Πίνακας 14: Vessel Label Typbe B	39

ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ

Εικόνα 1: Homepage.....	5
Εικόνα 2: Database Schema.....	6
Εικόνα 3: Vessel Type A	7
Εικόνα 4: Old Vessel Type B Old	7
Εικόνα 5: Arrow	7
Εικόνα 6: Ship details	8
Εικόνα 7: Navigation-bar - Warnings.....	10
Εικόνα 8: Entered/Left Area Warnings	10
Εικόνα 9: Collision Warnings.....	12
Εικόνα 10: Navigation bar - Map Area.....	13
Εικόνα 11: Edit Map Area Page	13
Εικόνα 12: Add Area Button	13
Εικόνα 13: Add Area Page	14
Εικόνα 14: Drawing Area Button	14
Εικόνα 15: Edit - Delete Area Buttons	14
Εικόνα 16: Area Type Buttons	14
Εικόνα 17: Edit - Delete Area Type Buttons	15
Εικόνα 18: Area Type Page	15
Εικόνα 19: New Vessel Type B.....	16
Εικόνα 20: New Vessel Type A.....	16
Εικόνα 21: Datetime Component.....	18
Εικόνα 22: Time Offset	18

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

ΤΕΠΑΚ.:	Τεχνολογικό Πανεπιστήμιο Κύπρου
GUI	Graphical User Interface
ΒΔ:	Βάση Δεδομένων

ΑΠΟΔΟΣΗ ΟΡΩΝ

Data	Δεδομένα
Metadata	Μεταδεδομένα
Vessel	Σκάφος
Components	Στοιχεία
Timestamp	Ημερομηνία ώρα
Latitude	Γεωγραφικό πλάτος

1 Εισαγωγή

Αρχικά, η μετακίνηση των πλοίων εντός και εκτός των περιοχών γύρω από την περιφέρεια της Κύπρου, καθώς επίσης και η παρουσία πολλών πιθανών συγκρούσεων μεταξύ των πλοίων, δημιούργησαν την ανάγκη για δημιουργία ενός διαδικτυακού συστήματος για την πολλαπλή ανίχνευση της κίνησης των πλοίων μεταξύ των περιοχών και των πιθανών συγκρούσεών τους.

Το σύστημα αυτόματης αναγνώρισης AISafety[1] κατασκευάστηκε με σκοπό την επίλυση των πιο πάνω προβλημάτων. Βασικές λειτουργίες του αποτελούν η προβολή των πλοίων σε πραγματικό χρόνο, καθώς επίσης και ο εντοπισμός τους, η ενημέρωση για την είσοδο/έξοδο ενός πλοίου σε μια περιοχή με τη χρήση των AIS Data [2] που λαμβάνονται από τα πλοία, όπως και η άμεση προειδοποίηση σε περίπτωση που ένα πλοίο πλησιάζει σε περιοχή υψηλού κινδύνου. Τα AIS Data είναι δεδομένα, τα οποία στέλνουν τα πλοία αποσκοπώντας στην απλή ενημέρωση για την τοποθεσία στην οποία βρίσκονται γνωρίζοντας το γεωγραφικό μήκος και πλάτος, το όνομά τους, τον τύπο τους (Εμπορικό, ψαροκάικο κτλ.), την κατεύθυνση και την ταχύτητα στην οποία κινούνται, την κατάσταση στην οποία βρίσκονται (αγκυροβολημένο, σε κίνηση), τον προορισμό, τον εκτιμώμενο χρόνο άφιξης κτλ. Με τη χρήση των παραπάνω δεδομένων και την υλοποίηση ενός κατάλληλου αλγορίθμου, έγινε εφικτή και η δυνατότητα ανίχνευσης και προβολής συγκρούσεων μεταξύ 2 πλοίων με υψηλή ακρίβεια, η οποία θεωρείται ως ακόμα μία βασική λειτουργία του συστήματος. Παράλληλα με τις δύο βασικές λειτουργίες του συστήματος, προβάλλονται στον χρήστη προειδοποιήσεις που τον ενημερώνουν πότε ένα πλοίο εισέρχεται / εξέρχεται από μία περιοχή ή όταν πρόκειται να συγκορυστούν δύο πλοία στο εγγύς μέλλον. Ακόμα μία επιπρόσθετη λειτουργία του συστήματος αποτελεί η προσθήκη, η επεξεργασία και η αφαίρεση μιας περιοχής από τον χάρτη.

Βασικό σκοπό της πτυχιακής εργασίας αποτελεί η γενική αναβάθμιση του συστήματος έτσι, ώστε να γίνει πιο εύχρηστο και φιλικό προς τον χρήστη και επίσης να προστεθούν νέες λειτουργίες προκειμένου να προσφέρει καινούριες δυνατότητες στον χρήστη, τις οποίες δεν είχε στο παρελθόν. Επιμέρους σκοποί αποτελούν η αναβάθμιση του γραφικού περιβάλλοντος για τα πλοία που παρουσιάζονται στο σύστημα, η προσθήκη νέας λειτουργίας για τη δυνατότητα μετάβασης σε ένα στιγμιότυπο μιας συγκεκριμένης

ημερομηνίας, η βελτιστοποίηση του αλγορίθμου εισόδου-εξόδου για την καλύτερη ανίχνευση εισόδου-εξόδου των πλοίων και η βελτιστοποίηση του αλγορίθμου ανίχνευσης συγκρούσεων για πιο ακριβή ανίχνευση πιθανής σύγκρουσης δύο πλοίων.

Αναφορικά με τον πρώτο επιμέρους σκοπό, αυτό που χρειαζόταν να πραγματοποιηθεί ήταν η αναβάθμιση των εικονιδίων των πλοίων σε εικονίδια που θα απεικονίζουν παράλληλα το μέγεθος και την κατεύθυνση στην οποία κινούνται έτσι ώστε να είναι πιο φιλικό προς τον χρήστη. Όμως, το κυριότερο πρόβλημα που εντοπίστηκε στο υφιστάμενο σύστημα ήταν η αδυναμία απεικόνισης του πλοίου στο πραγματικό του μέγεθος ανάλογα με την κλίμακα του χάρτη. Πιο συγκεκριμένα, το πλοίο διατηρούσε το μέγεθος του σε όλα τα επίπεδα μεγέθυνσης του χάρτη χωρίς να λαμβάνει υπόψη τις πραγματικές διαστάσεις του πλοίου που έπαιρνε από τα AIS Data του πλοίου. Έτσι με την χρήση των κατάλληλων μαθηματικών υπολογισμών έγινε εφικτή η προβολή των πλοίων ανάλογα με το πραγματικό μέγεθος και το επίπεδο μεγέθυνσης του χάρτη.

Για την υλοποίηση του δεύτερου στόχου, αρχικά παρατηρήθηκε η αδυναμία του συστήματος να προβάλλει γεγονότα στιγμιότυπα που συνέβησαν στο παρελθόν. Γι αυτό το σκοπό έχει προστεθεί στο σύστημα μία νέα λειτουργία που επιτρέπει στον χρήστη να πλοηγηθεί στο παρελθόν και να του εμφανίσει όλες τις τοποθεσίες των πλοίων πάνω στον χάρτη (μαζί με όλες τις απαραίτητες πληροφορίες) και τις προειδοποιήσεις που είχαν εμφανιστεί εκείνη την χρονική περίοδο. Επιπλέον, ο χρήστης έχει τη δυνατότητα να επαναφέρει το σύστημα σε πραγματικό χρόνο.

Αναφορικά με τον τρίτο στόχο, παρατηρήθηκε στο σύστημα ένα θέμα σχετικά με την προβολή ειδοποιήσεων σχετικά με τα πλοία που εισέρχονταν/εξέρχονταν από μία περιοχή. Πιο συγκεκριμένα, παρατηρήθηκε ότι για κάποια πλοία το σύστημα έβγαζε λανθασμένες ειδοποιήσεις για την είσοδο/έξοδο ενός πλοίου από μια περιοχή. Αυτό συνέβαινε περισσότερο λόγω του ότι καμιά φορά μπορεί το σύστημα να λάβει λανθασμένες συντεταγμένες από τα AIS Data του πλοίου. Γι αυτό τον σκοπό, ο αλγόριθμος με τον οποίο το σύστημα ανίχνευε πότε ένα πλοίο εισέρχεται/εξέρχεται από μια περιοχή βελτιστοποιήθηκε έτσι ώστε να στέλνει πιο ακριβή αποτελέσματα για τις συμπεριφορές των πλοίων. Η βασική ιδέα στο να γίνει αυτό ήταν ότι το σύστημα έπρεπε να επαληθεύει τις συμπεριφορές των πλοίων προτού να ενημερώσει τον χρήστη.

Επίσης, παράλληλα με τους πιο πάνω στόχους έγιναν και κάποιες επιπρόσθετες αλλαγές, όπως για παράδειγμα η δυνατότητα προβολής των περιοχών του συστήματος από τον

χρήστη. Αρχικά την πρόσβαση σε αυτόν τον τομέα του συστήματος είχε μόνο ο admin και ο superadmin του συστήματος, όπως επίσης είχαν και την δυνατότητα να προσθέσουν, να αφαιρέσουν και να διαγράψουν μια περιοχή.

2 Δομή Υφιστάμενου Συστήματος

2.1 Αρχιτεκτονική

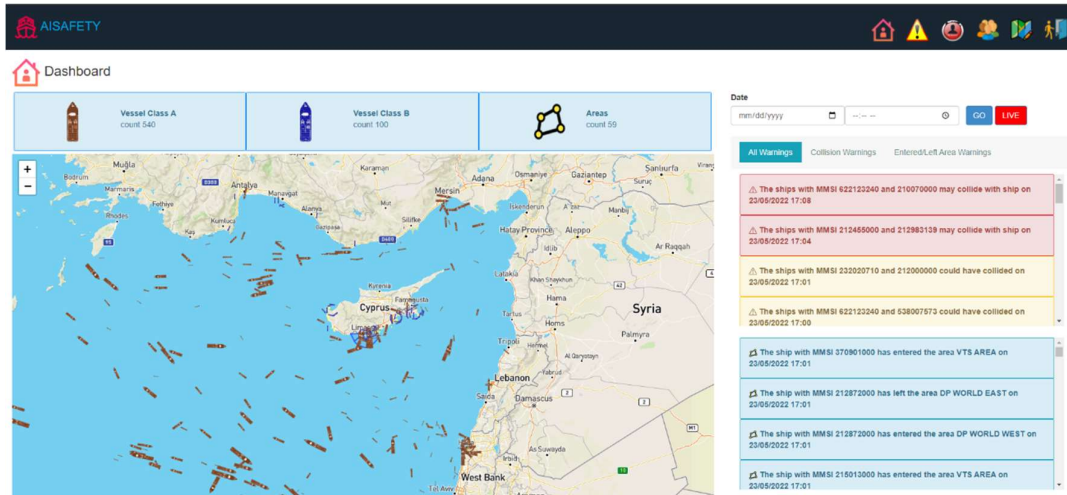
Αρχικά, η πλατφόρμα έχει υλοποιηθεί χρησιμοποιώντας το PHP framework Laravel [3].

Το σύστημα αποτελείται από τα εξής μέρη:

- Τους controllers, οι οποίοι λειτουργούν σαν τον εγκέφαλο του συστήματος και είναι υπεύθυνοι για τον συντονισμό και την καλή λειτουργία μεταξύ των υπολοίπων κομματιών της αρχιτεκτονικής.
- Τα views, τα οποία είναι οι σελίδες του συστήματος που απεικονίζουν τις διάφορες λειτουργίες του και τα διάφορα components του
- Τα routes, τα οποία είναι υπεύθυνα για την ομαλή μετάβαση από μία σελίδα στην άλλη. Όταν ο χρήστης επιλέγει να μεταβεί σε κάποιο συγκεκριμένο view, μέσω των routes μεταφέρεται στον αντίστοιχο controller και αυτός δρομολογεί το σύστημα στο κατάλληλο view.
- Τα models, τα οποία είναι υπεύθυνα για την καλή επικοινωνία με τη ΒΔ. Ανακτούν data και τα στέλνουν στο controller ο οποίος με τη σειρά του τα στέλνει σε μορφή json στο view για να τα εμφανίσει πάνω στο χάρτη ο οποίος είναι κατασκευασμένος χρησιμοποιώντας τη βιβλιοθήκη leaflet [4] της javascript[5].

Η αρχική σελίδα της πλατφόρμας περιέχει πάνω ψηλά τον συνολικό αριθμό πλοίων για κάθε τύπο πλοίου ξεχωριστά και τον συνολικό αριθμό περιοχών. Επίσης, από κάτω υπάρχει ο χάρτης όπου παρουσιάζονται όλα τα πλοία και οι περιοχές. Στο πλάι, υπάρχει το component για το νέο feature που θα εξηγήσουμε στη συνέχεια και από κάτω υπάρχουν τα warnings τα οποία είναι χωρισμένα σε 3 καρτέλες:

1. All Warnings
2. Collision Warnings
3. Entered/Left Area Warnings



Εικόνα 1: Homepage

2.2 Βάση Δεδομένων

Στο σύστημα ανταλλάζονται συνεχώς data μεταξύ των πλοίων, του συστήματος και του χρήστη. Αυτά τα data αποθηκεύονται στην ΒΔ(Εικόνα 2). Η ΒΔ του συστήματος χρησιμοποιεί το σχεσιακό μοντέλο βάσης δεδομένων και το σύστημα επικοινωνεί μαζί της χρησιμοποιώντας SQL.

Οι βασικές πληροφορίες που αποθηκεύει η βάση είναι:

- 1) AIS data για τα πλοία τύπου A και τύπου B (Πίνακας 1,2,3,4,5)
- 2) (Προ)ειδοποιήσεις για πιθανές συγκρούσεις των πλοίων ή για είσοδο / έξοδο των πλοίων σε / από μια περιοχή (Πίνακας 6,7,8)
- 3) Πληροφορίες για τις περιοχές και τα λιμάνια όπως και το ποια πλοία είναι μέσα ή είναι κοντά σε αυτά (Πίνακας 9,10,11)
- 4) Κατηγορίες των πλοίων (Πίνακας 12)
- 5) Διάφορες πληροφορίες που ωφελούν σε γενικό επίπεδο

mw_ship_conflict_destination	
123 id	int(11) NOT NULL
123 mmsi_1	int(11) NOT NULL
123 mmsi_2	int(11) NOT NULL
received_timestamp	timestamp NOT NULL

mw_ship_in_area	
123 id	int(11) NOT NULL
123 mmsi	int(11) NOT NULL
123 area_id	int(11) NOT NULL
123 type	int(11) NOT NULL

ship_type	
123 id	smallint(5) unsigned NOT NULL
abc message	varchar(60)
abc ship_category	varchar(100) NOT NULL
DateTime_Updated	timestamp NOT NULL

statistics_ship_countries	
date	date NOT NULL
abc name	varchar(100) NOT NULL
123 total	int(10)
DateTime_Updated	timestamp NOT NULL

mw_warning	
123 id	int(11) NOT NULL
123 mmsi	int(11) NOT NULL
123 area_id	int(11) NOT NULL
123 type_of_warning	int(11) NOT NULL
received_timestamp	timestamp NOT NULL

cpa_arrivals	
abc Port	varchar(5)
123 ShipIMO	int(11)
123 MMSI	int(11)
ATA	timestamp
ATD	timestamp
abc ShipType	varchar(100)
abc OriginPort	varchar(100)
abc DestinationPort	varchar(100)

destination_mapping	
abc destination_dirty	varchar(100) NOT NULL
abc ClearPortName	varchar(100)
abc CleanPortLocode	varchar(100)
abc CleanCountryName	varchar(100)
abc CleanCountryCode	varchar(100)
abc MatchingMethod	varchar(200)
123 Similarity	double
123 BatchID	int(11)
DateTime_Updated	timestamp NOT NULL

mw_users	
123 id	bigint(20) unsigned NOT NULL
123 role	int(11) NOT NULL
abc name	varchar(255) NOT NULL
abc email	varchar(255) NOT NULL
email_verified_at	timestamp
abc password	varchar(255) NOT NULL
abc remember_token	varchar(100)
created_at	timestamp
updated_at	timestamp

port_unlocode	
abc Locode	char(5) NOT NULL
abc CountryCode	char(2)
abc CountryName	varchar(100)
abc PortCode	char(3)
abc Name	varchar(100)
abc NameWoDiacritics	varchar(100)
abc Subdivision	varchar(5)
abc Latitude	varchar(8)
abc Longitude	varchar(8)
DateTime_Updated	timestamp NOT NULL

statistics_daily	
date	date NOT NULL
123 basestationreport	int(10) NOT NULL
123 positionreporta	int(10) NOT NULL
123 positionreportb	int(10) NOT NULL
123 othersignals	int(10) NOT NULL
123 staticvoyagedata	int(10) NOT NULL
123 ships	int(10) NOT NULL
123 moored	int(10) NOT NULL
123 worldeast	int(10) NOT NULL
123 worldnorth	int(10) NOT NULL
123 worldwest	int(10) NOT NULL
123 worldoro	int(10) NOT NULL
123 eurogatem	int(10) NOT NULL
123 cpawest	int(10) NOT NULL
123 cpasouth	int(10) NOT NULL
123 anchoring	int(10) NOT NULL
123 minwait	int(10) NOT NULL
123 maxwait	int(10) NOT NULL
123 avgwait	int(10) NOT NULL
123 marina	int(10) NOT NULL
DateTime_Updated	timestamp NOT NULL

fixing_device	
123 id	smallint(6) NOT NULL
abc message	varchar(30)
DateTime_Updated	timestamp NOT NULL

mw_password_resets	
abc email	varchar(255) NOT NULL
abc token	varchar(255) NOT NULL
created_at	timestamp

nav_status	
123 id	smallint(6) NOT NULL
abc message	varchar(65)
DateTime_Updated	timestamp NOT NULL

countries	
123 id	int(11) NOT NULL
123 code	int(10) unsigned NOT NULL
abc name	varchar(100)
DateTime_Updated	timestamp NOT NULL

mw_area_type	
123 id	int(11) NOT NULL
abc name	text NOT NULL
abc description	text NOT NULL
123 type_of_message	int(11) NOT NULL

ship_categories	
abc ShipType	varchar(100) NOT NULL
abc ShipCategory	varchar(100) NOT NULL
123 AvgBreadth	float
123 AvgLength	float
DateTime_Updated	timestamp NOT NULL

mw_area	
123 id	int(11) NOT NULL
abc description	mediumtext
abc email	varchar(256) NOT NULL
123 type	int(3) NOT NULL
123 active	int(11) NOT NULL
123 size	bigint(255)

known_base_stations	
123 id	int(11) NOT NULL
123 mmsi	int(11)
123 latitude	float NOT NULL
123 longitude	float NOT NULL
abc source	varchar(45) NOT NULL
abc location	varchar(100)
DateTime_Updated	timestamp NOT NULL

mw_conflict_warning	
123 id	int(11) NOT NULL
123 mmsi_1	int(11) NOT NULL
123 mmsi_2	int(11) NOT NULL
123 lat	float NOT NULL
123 lon	float NOT NULL
clock	timestamp NOT NULL
received_timestamp	timestamp

othersignals	
123 id	int(10) NOT NULL
received_timestamp	timestamp NOT NULL
123 type	smallint(6)
123 mmsi	int(11)
123 country_code	int(10) unsigned
abc message	text
123 source	smallint(6)

positionreportb	
123 id	int(11) NOT NULL
received_timestamp	timestamp NOT NULL
123 type	smallint(6)
123 repeat_indicator	tinyint(4)
123 mmsi	int(11)
123 country_code	int(10) unsigned
123 position_accuracy	tinyint(4)
123 longitude	float
123 latitude	float
123 cog	float
123 true_heading	smallint(6)
123 time_stamp	smallint(6)
123 unit_flag	tinyint(4)
123 display_flag	tinyint(4)
123 dsc_flag	tinyint(4)
123 band_flag	tinyint(4)
123 m22_flag	tinyint(4)
123 mode_flag	tinyint(4)
123 raim	tinyint(1)
123 commstate_flag	tinyint(4)
123 commstate_cs_fill	int(11)
123 sog	float
123 source	smallint(6)

vessel	
123 imo	int(11)
abc vesselId	varchar(30)
123 mmsi	int(11)
abc vesselName	varchar(256) NOT NULL
abc vesselType	varchar(32)
abc call_sign	varchar(32)
abc flag	varchar(32)
abc builtYear	varchar(8)
abc dataSource	varchar(45) NOT NULL
abc photoURL	varchar(1024)
abc buildPlace	varchar(1024)
123 breadth	float
123 length	float
123 draught	float
abc owner	varchar(256)
abc manager	varchar(256)
abc dwtSummer	varchar(10)
abc displacementSummer	varchar(10)
abc fuelConsumption	varchar(10)
abc maxSpeed	varchar(10)
abc serviceSpeed	varchar(10)
abc liquidOil	varchar(10)
DateTime_Updated	timestamp NOT NULL

staticvoyagedata	
123 id	int(11) NOT NULL
received_timestamp	timestamp NOT NULL
123 type	smallint(6)
123 repeat_indicator	tinyint(4)
123 mmsi	int(11)
123 country_code	int(10) unsigned
123 ais_version	tinyint(4)
123 imo_number	int(11)
abc call_sign	varchar(7)
abc vessel_name	varchar(20)
123 ship_type	smallint(6)
123 d_box	int(11)
123 d_stern	int(11)
123 d_port	int(11)
123 d_starboard	int(11)
123 fix_type	smallint(6)
123 eta_month	smallint(6)
123 eta_day	smallint(6)
123 eta_hour	smallint(6)
123 eta_min	smallint(6)
eta_total	datetime
123 draught	float
abc destination	varchar(50)
123 dte	tinyint(4)
123 spare	tinyint(4)
123 source	smallint(6)

basestationreport	
123 id	int(10) unsigned NOT NULL
received_timestamp	timestamp NOT NULL
123 type	tinyint(1)
123 repeat_indicator	int(11)
123 mmsi	int(11) unsigned
123 country_code	int(10) unsigned
123 utc_year	smallint(4) unsigned
123 utc_month	tinyint(2) unsigned
123 utc_day	tinyint(2) unsigned
123 utc_hour	tinyint(2) unsigned
123 utc_minute	tinyint(2) unsigned
123 utc_second	tinyint(2) unsigned
123 utc_spare	tinyint(2)
123 fix_quality	smallint(6)
123 longitude	float
123 latitude	float
123 transm_control	tinyint(1)
123 raim_flag	tinyint(1)
123 comm_state	int(11)
123 fix_type	smallint(6) unsigned
123 sync_state	tinyint(4)
123 slot_timeout	smallint(6)
123 slot_number	int(11)
123 slot_offset	int(11)
123 received_stations	smallint(6)
123 transmission_cti	tinyint(4)
123 source	smallint(6)

positionreporta	
123 id	int(11) NOT NULL
received_timestamp	timestamp NOT NULL
123 type	smallint(6)
123 repeat_indicator	tinyint(4)
123 mmsi	int(11)
123 country_code	int(10) unsigned
123 nav_status	smallint(6)
123 rot	float
123 sog	float
123 position_accuracy	tinyint(4)
123 longitude	float
123 latitude	float
123 cog	float
123 true_heading	smallint(6)
123 time_stamp	smallint(6)
123 maneuver_indicator	tinyint(4)
123 raim_flag	tinyint(1)
123 comm_state	int(11)
123 rot_over_range	tinyint(1)
123 sync_state	smallint(6)
123 slot_timeout	smallint(6)
123 slot_number	int(11)
123 slot_offset	int(11)
123 utc_hour	smallint(6)
123 utc_min	smallint(6)
123 utc_spare	smallint(6)
123 keep_flag	tinyint(1)
123 slots_to_allocate	smallint(6)
123 received_stations	smallint(6)
123 source	smallint(6)

Εικόνα 2: Database Schema

2.3 Λειτουργίες Συστήματος

2.3.1 Προβολή πλοίων πάνω στο χάρτη

Έχουμε δύο κατηγορίες πλοίων: τύπου A και τύπου B. Τα πλοία τύπου A είναι συνήθως μεγάλου μεγέθους πλοία όπως εμπορικά, ταξιδιωτικά πλοία. Τα πλοία τύπου B είναι μικρού μεγέθους καράβια όπως ψαροκάϊκα κ.α. Ο τρόπος με τον οποίο τα πλοία προβάλλονται πάνω στον χάρτη είναι ο εξής:

1. Αρχικά λαμβάνουμε AIS data ανά τακτά χρονικά διαστήματα και από τους δύο τύπους πλοίων και τα αποθηκεύουμε στη βάση δεδομένων
2. Από τον controller εκτελείται ένα ερώτημα στη βάση δεδομένων με σκοπό να πάρουμε τις πληροφορίες του κάθε πλοίου (συντεταγμένες, κατεύθυνση) τα τελευταία 20 λεπτά.
3. Στην συνέχεια τα data αυτά μεταφέρονται σε μορφή json στην σελίδα του χάρτη όπου για κάθε τύπο πλοίου απεικονίζεται το αντίστοιχο εικονίδιο (Εικόνα 3,4) όπως επίσης και η πιο πρόσφατη τοποθεσία στην οποία βρίσκεται τα τελευταία 20 λεπτά. Για την κατεύθυνση του πλοίου υπάρχει ένα ξεχωριστό εικονίδιο (Εικόνα 5) το οποίο δείχνει την κατεύθυνση του πλοίου.
4. Η διαδικασία αυτή επαναλαμβάνεται κάθε 10 δευτερόλεπτα



Εικόνα 3: Vessel Type A

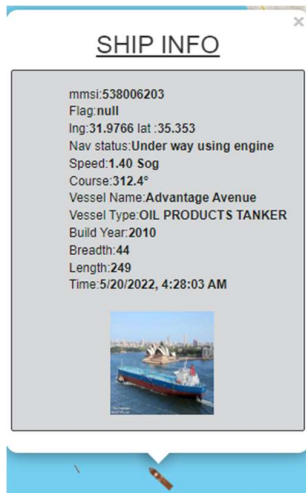


Εικόνα 4: Old Vessel Type B Old



Εικόνα 5: Arrow

Επίσης, τα πλοία που απεικονίζονται πάνω στον χάρτη έχουν την ιδιότητα όταν πατήσεις πάνω τους, εμφανίζεται μια καρτέλα με τις πληροφορίες του πλοίου (Πίνακας 13,14)(Εικόνα 6).



Εικόνα 6: Ship details

2.3.2 Ειδοποίηση του χρήστη για είσοδο/έξοδο των πλοίων σε/από μια περιοχή

Στο σύστημα υπάρχουν τέσσερα διαφορετικά είδη ειδοποιήσεων για είσοδο/έξοδο σε/από μια περιοχή:

1. Είσοδος σε μια περιοχή
2. Έξοδος από μια περιοχή
3. Προσέγγιση σε μια περιοχή υψηλού κινδύνου
4. Είσοδος σε μια περιοχή υψηλού κινδύνου

Για τη διαδικασία ανίχνευσης της κατάστασης του πλοίου γι' αυτήν την περίπτωση χρησιμοποιείται ο αλγόριθμος `vessel_in_area_algorithm`.

2.3.2.1 Αλγόριθμος *Vessel in area*

Ο αλγόριθμος αυτός ανακτά όλες τις πιο πρόσφατες πληροφορίες για την κατάσταση των πλοίων που υπάρχουν στο σύστημα τα τελευταία 20 λεπτά από τη βάση δεδομένων όπως επίσης και όλες τις περιοχές των πλοίων. Ακόμη, ανακτά και όλες τις πληροφορίες για τα πλοία που βρίσκονται μέσα ή έξω από μια συγκεκριμένη περιοχή. Στη συνέχεια ο αλγόριθμος χωρίζεται σε δύο περιπτώσεις. Η μία περίπτωση είναι όταν έχουμε γνωστά το μήκος και το πλάτος και έτσι η άλλη είναι όταν δεν τα έχουμε γνωστά. Στην πρώτη περίπτωση, ο αλγόριθμος χρησιμοποιώντας το μήκος και το πλάτος του πλοίου μπορεί να υπολογίσει με ακρίβεια το πότε ένα πλοίο βρίσκεται εντός ή εκτός από μια περιοχή χρησιμοποιώντας τα τέσσερα άκρα του πλοίου. Αυτό, έλυσε και το πρόβλημα όπου στην περίπτωση που ένα πλοίο ήταν το μισό εντός και το άλλο μισό εκτός από μια περιοχή, το σύστημα μπερδευόταν και πολλές φορές έστελνε συνεχόμενες λάθος ειδοποιήσεις. Στη δεύτερη περίπτωση, ο αλγόριθμος βασιζόταν απλά στις συντεταγμένες του πλοίου.

Άρα έχοντας εις γνώσιν κατά πόσο το πλοίο βρίσκεται εντός ή εκτός της συγκεκριμένης περιοχής, ο αλγόριθμος εξετάζει από τα data της βάσης εάν το πλοίο ήταν ήδη εντός ή εκτός της περιοχής. Εάν το πλοίο πριν ήταν εντός και τώρα υπολογίστηκε ότι το πλοίο είναι εκτός περιοχής, τότε ο αλγόριθμος εισάγει στον πίνακα `mw_warning` μια καινούργια εγγραφή που δείχνει ότι το πλοίο βγαίνει εκτός μιας περιοχής και επίσης διαγράφεται από τον πίνακα `ship_in_area` η εγγραφή που λέει για το συγκεκριμένο πλοίο ότι βρίσκεται στην συγκεκριμένη περιοχή. Αντίστοιχα, εάν το πλοίο ήταν εκτός από μια περιοχή και τώρα υπολογίστηκε ότι είναι εντός, ο αλγόριθμος εισάγει μια καινούργια εγγραφή στον πίνακα `mw_warning` που δείχνει ότι το πλοίο μπαίνει σε μια περιοχή και επίσης εισάγει μια νέα εγγραφή και στον πίνακα `ship_in_area` που δείχνει ότι το συγκεκριμένο πλοίο βρίσκεται στη συγκεκριμένη περιοχή. Η ίδια διαδικασία ισχύει και για την περίπτωση που το πλοίο πλησιάζει μια περιοχή υψηλού κινδύνου και για την περίπτωση που το πλοίο έχει μπει στην συγκεκριμένη περιοχή.

2.3.2.2 Απεικόνιση Ειδοποιήσεων

Αφότου ο αλγόριθμος εισάγει στη ΒΔ τις ειδοποιήσεις, με τη χρήση ενός `model` ανακτούμε τα data από τον πίνακα `mw_warning` για τη δημιουργία της ειδοποίησης. Το σύστημα εμφανίζει τις ειδοποιήσεις πάνω στο σύστημα με μπλε χρώμα για είσοδο/έξοδο των πλοίων σε μια περιοχή και με ανοιχτό κίτρινο την προσέγγιση/είσοδο ενός πλοίου

από μια περιοχή. Επιπλέον, εμφανίζει όλες τις ειδοποιήσεις που συνέβησαν εντός της μίας ώρας. Αξίζει να σημειωθεί ότι πατώντας το κουτάκι της ειδοποίησης, ο χάρτης εμφανίζει μία καρτέλα πάνω από το πλοίο η οποία ενημερώνει τον χρήστη για την είσοδο/έξοδο του πλοίου σε/από μια περιοχή.

Ακόμη, πατώντας στην μπάρα πλοήγησης την επιλογή των Warnings (Εικόνα 7), το σύστημα μεταφέρει τον χρήστη στην σελίδα όπου υπάρχουν μαζεμένες όλες οι ειδοποιήσεις με τις απαραίτητες πληροφορίες των πλοίων και των περιοχών(Εικόνα 8).



Εικόνα 7: Navigation-bar - Warnings

Entered/Left Warnings

ID	MMSI	WARNING TYPE	AREA NAME	RECEIVED TIMESTAMP
483922	636018874	Entered area	TRAFFIC AREA	2022-05-23 19:34:42
483921	236661000	Entered area	TRAFFIC AREA	2022-05-23 19:34:50
483920	210330000	Left area	CPA SOUTH	2022-05-23 19:33:27
483919	209711693	Left area	DP WORLD WEST	2022-05-23 19:33:26
483918	636018874	Left area	TRAFFIC AREA	2022-05-23 19:33:31
483917	212983121	Left area	ANCHORAGE	2022-05-23 19:32:17
483916	209212000	Left area	ANCHORAGE	2022-05-23 19:32:52
483915	210330000	Entered area	CPA SOUTH	2022-05-23 19:32:08
483914	210282000	Left area	PORT AREA	2022-05-23 19:32:07
483913	236661000	Left area	TRAFFIC AREA	2022-05-23 19:31:55

Εικόνα 8: Entered/Left Area Warnings

2.3.3 Προειδοποίηση Χρήστη για Πιθανή Σύγκρουση Δύο Πλοίων

2.3.3.1 Αλγόριθμος *Collision Detection*

Ο αλγόριθμος αυτός αρχικά διαγράφει όλες τις συγκρούσεις που έχουν γίνει στο παρελθόν. Στη συνέχεια πέρνει όλα τα πλοία από τη ΒΔ και τα ταξινομεί βάση το latitude. Επειτα, για κάθε πλοίο ελέγχει εάν η διαφορά του του γεωγραφικού πλάτους του από το αμέσως επόμενο είναι μικρότερη από 0,5 τότε ο αλγόριθμος χωρίζεται σε δύο περιπτώσεις:

- Να κινούνται και τα δύο πλοία
- Να κινείται μόνο το ένα πλοίο

Στην πρώτη περίπτωση, ο αλγόριθμος εκτελεί μία συνάρτηση η οποία σχηματίζει 2 γραμμές (μία για το κάθε πλοίο) και εξετάζει κατά πόσο οι 2 γραμμές αυτές συγκρούονται μεταξύ τους. Εάν συγκρούονται τότε ο αλγόριθμος υπολογίζει την εκτιμώμενη ώρα θα φτάσει το κάθε πλοίο ξεχωριστά στο σημείο που διασταυρώνονται οι γραμμές. Εάν η διαφορά των 2 αυτών χρόνων είναι $< |2|$, τότε υπολογίζει το μέσο χρόνο που υπολογίζεται τα δύο πλοία να συγκρουστούν. Αν δεν υπάρχει ήδη προειδοποίηση στον πίνακα `conflict_warnings`(πίνακας `collision`), τότε ο αλγόριθμος καταγράφει μία καινούργια προειδοποίηση ότι για τα δύο αυτά πλοία υπάρχει περίπτωση να συγκρουστούν.

Όσον αφορά την άλλη περίπτωση, ο αλγόριθμος πάλι εκτελεί τη συνάρτηση για να εξετάσει κατά πόσο οι γραμμές των πλοίων συγκρούονται μεταξύ τους. Εάν συγκρούονται ο αλγόριθμος απλά υπολογίζει τον αναμενόμενο χρόνο που θέλει να φτάσει το κινούμενο πλοίο και για σημείο σύγκρουσης κρατά την τοποθεσία του ακίνητου πλοίου. Εάν τα δύο πλοία δεν έχουν ήδη προειδοποίηση στη ΒΔ, ο αλγόριθμος καταγράφει μια καινούργια προειδοποίηση ότι για τα δύο αυτά πλοία ελλοχεύει ο κίνδυνος της σύγκρουσης

Αξίζει να σημειωθεί ότι και στις δύο περιπτώσεις η καταγραφή όλων των warnings γίνεται στο σύστημα Port Collaborative Decision Making (PortCDM).

2.3.3.2 Απεικόνιση Προειδοποιήσεων

Αφότου ο αλγόριθμος εισάγει στη ΒΔ τις προειδοποιήσεις, με τη χρήση ενός model ανακτούμε τα data από τον πίνακα `conflict warning` για τη δημιουργία της

προειδοποίησης. Το σύστημα εμφανίζει τις προειδοποιήσεις για την επερχόμενη σύγκρουση δύο πλοίων για τα επόμενα 20 λεπτά πάνω στο σύστημα με κόκκινο χρώμα και με ανοιχτό κίτρινο τις πιθανές συγκρούσεις που τελικά δεν ολοκληρώθηκαν στο παρελθόν εντός της μίας ώρας. Αξίζει να σημειωθεί ότι πατώντας το κόκκινο κουτάκι της προειδοποίησης, ο χάρτης εμφανίζει μία καρτέλα στο σημείο σύγκρουσης η οποία ενημερώνει ξανά το χρήστη για την πιθανή σύγκρουση και επίσης του εμφανίζει και τις 2 γραμμές των πλοίων που η διασταύρωση τους δείχνει στο χρήστη το πιθανό σημείο σύγκρουσης.

Ακόμη, πατώντας στην μπάρα πλοήγησης την επιλογή των Warnings (Εικόνα 7), το σύστημα μεταφέρει τον χρήστη στην σελίδα όπου υπάρχουν μαζεμένες όλες οι προειδοποιήσεις με τις απαραίτητες πληροφορίες των πλοίων και των περιοχών (Εικόνα 9).

Collision Warnings

ID	MMSI	MMSI	LATITUDE	LONGITUDE	PREDICTED TIME	RECEIVED TIMESTAMP
165664	538008915	235009410	31.3495	32.3811	2022-05-23 19:37:42	2022-05-23 19:35:42
165663	636018075	563047500	31.2862	32.3618	2022-05-23 19:33:40	2022-05-23 19:29:40
165662	431929000	403528001	31.4817	32.3862	2022-05-23 19:40:40	2022-05-23 19:26:40
165661	232020710	212000000	34.9795	33.9469	2022-05-23 19:28:39	2022-05-23 19:20:39
165660	538003101	636091833	31.7389	32.0019	2022-05-23 19:23:38	2022-05-23 19:18:38
165659	235009410	538008915	31.3082	32.369	2022-05-23 19:22:38	2022-05-23 19:16:38
165658	622110051	229007000	31.522	31.7664	2022-05-23 19:28:37	2022-05-23 19:15:37
165657	253260000	235009410	31.4608	32.4076	2022-05-23 19:18:37	2022-05-23 19:13:37
165656	212899000	212983121	34.6687	33.0627	2022-05-23 19:15:36	2022-05-23 19:11:37
165655	210595000	212899000	34.6695	33.0744	2022-05-23 19:22:36	2022-05-23 19:10:36

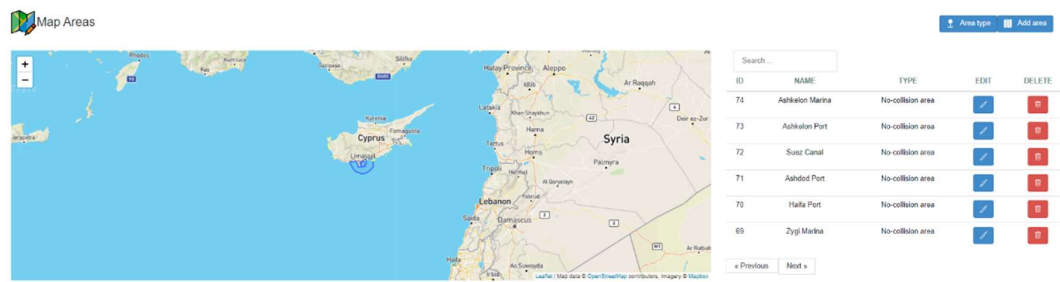
Εικόνα 9: Collision Warnings

2.3.4 Διαχείριση Περιοχών

Την ευθύνη για την διαχείριση των περιοχών την έχουν αποκλειστικά ο admin και ο superadmin. Πατώντας το εικονίδιο για επεξεργασία των περιοχών του χάρτη στη μπάρα πλοήγησης (Εικόνα 10), το σύστημα μεταφέρεται στη σελίδα όπου υπάρχουν μαζεμένες όλες οι περιοχές του χάρτη (Εικόνα 11). Εκεί, ο καθένας τους έχει τη δυνατότητα να προσθέσει μια καινούργια περιοχή πατώντας το κουμπί Add Area (Εικόνα 12).



Εικόνα 10: Navigation bar - Map Area

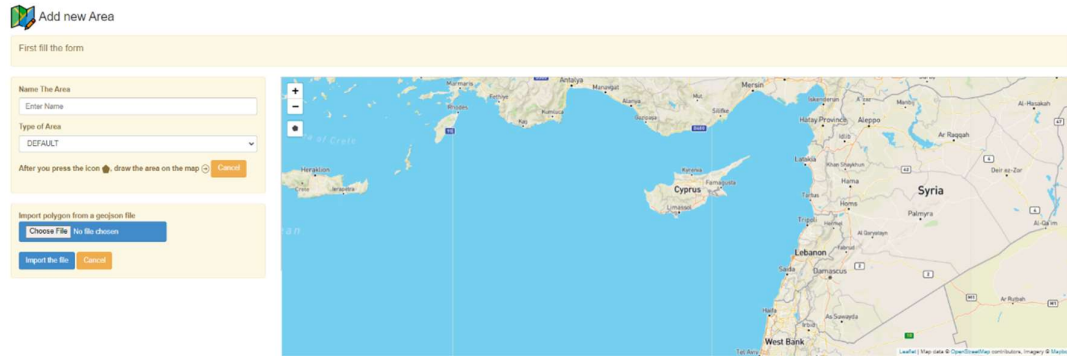


Εικόνα 11: Edit Map Area Page



Εικόνα 12: Add Area Button

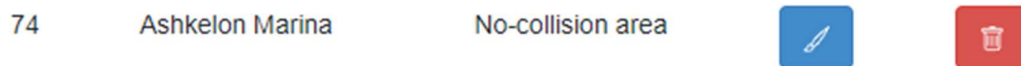
Επειτα μεταφέρεται στη σελίδα όπου έχει τη δυνατότητα να προσθέσει καινούργια περιοχή (Εικόνα 13). Ακολούθως προσθέτουν όνομα, επιλέγουν τύπο περιοχής και ακολούθως πατώντας το κουμπί πάνω στο χάρτη (Εικόνα 14) σχεδιάζουν την περιοχή που θέλουν. Ακόμη, έχουν τη δυνατότητα να ανεβάσουν ένα έτοιμο geojson αρχείο με έτοιμη μέσα την περιοχή. Μπορούν επίσης, να επεξεργαστούν και να διαγράψουν μια περιοχή πατώντας τα δύο κουμπιά δίπλα από το όνομα της περιοχής (Εικόνα 15).



Εικόνα 13: Add Area Page



Εικόνα 14: Drawing Area Button

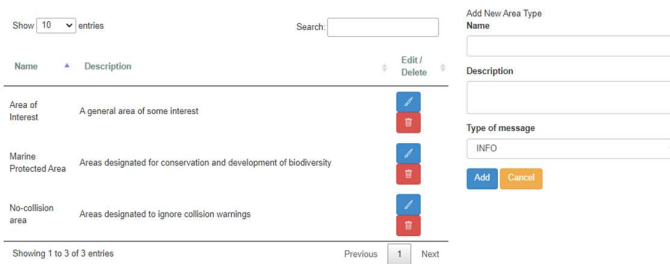


Εικόνα 15: Edit - Delete Area Buttons

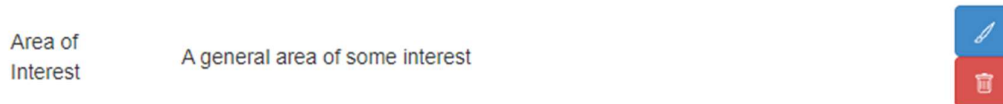
Επιπρόσθετα, έχουν τη δυνατότητα να προσθέσουν νέο τύπο περιοχής πατώντας το κουμπί Area Type (Εικόνα 16). Επομένως, το σύστημα μεταφέρεται στην αντίστοιχη σελίδα (Εικόνα 17) και έχουν τη δυνατότητα να προσθέσουν καινούργιο τύπο περιοχής δίνοντας όνομα και περιγραφή, να επεξεργαστούν και να διαγράψουν μίαν υπάρχουσα περιοχή πατώντας τα αντίστοιχα κουμπιά δίπλα από τον τύπο περιοχής (Εικόνα 18)



Εικόνα 16: Area Type Buttons



Εικόνα 18: Area Type Page



Εικόνα 17: Edit - Delete Area Type Buttons

3 Σχετική εργασία

3.1 Αναβάθμιση Γραφικού Περιβάλλοντος

3.1.1 Αλλαγή Απεικόνισης Πλοίων

Για την υλοποίηση του συγκεκριμένου στόχου, αναζητήθηκαν διαφορές επιλογές για την απεικόνιση των πλοίων. Προτιμήθηκε η επιλογή του συγκεκριμένου εικονιδίου (Εικόνα 19,20) και για τους δύο τύπους πλοίων με σκοπό την παρουσίαση του τύπου του πλοίου και της κατεύθυνσης του με ένα μόνο εικονίδιο και όχι με δύο όπως ήταν προηγουμένως και ξεχωρίζουμε τους δύο τύπους των πλοίων ανάλογα με το χρώμα τους. Συγκεκριμένα, χρησιμοποιήθηκε καφέ για πλοία τύπου Α και μπλε για πλοία τύπου Β.



Εικόνα 20: New Vessel Type A



Εικόνα 19: New Vessel Type B

3.1.2 Εφαρμογή Πραγματικού Μεγέθους κατά τη Μεγένθυση

Για την εφαρμογή του πραγματικού μεγέθους χρειάστηκαν οι εξής παραμέτροι:

- Διαστάσεις πλοίου:

Το κάθε πλοίο όπως έχουμε προαναφέρει έχει αποθηκευμένες τις διαστάσεις του μέσα στην ΒΔ από τα AIS data που στέλνει. Κάποια πλοία έχουν διαστάσεις κάποια άλλα όχι. Για την περίπτωση αυτή, έχει προστεθεί στην ΒΔ ένας καινούργιος πίνακας (πίνακας ship_categories) ο οποίος περιέχει ένα σταθερό μήκος και πλάτος ανάλογα με τον τύπο του κάθε πλοίου. Επιπλέον, στην περίπτωση που ούτε ο τύπος του πλοίου είναι γνωστός, τότε οι διαστάσεις για πλοία τύπου A και B είναι (24,180) και (10,40) αντίστοιχα, όπου η πρώτη παράμετρος αντιστοιχεί στο πλάτος και η δεύτερη στο μήκος του σκάφους.

- Κλίμακα χάρτη:

Πάνω στον χάρτη υπάρχει ενσωματωμένη η κλίμακα του χάρτη σε χιλιόμετρα και μίλια. Αξίζει να σημειωθεί ότι η κλίμακα του χάρτη αλλάζει από σημείο σε σημείο λόγω του σχήματος της γης.

- Επίπεδο μεγέθυνσης χάρτη:

Ο χάρτης έχει συνολικά 13 επίπεδα μεγέθυνσης. Αρχίζει από το επίπεδο 6 που είναι το πιο μακρινό μέχρι το επίπεδο 18 που έχει το μεγαλύτερο βαθμό μεγέθυνσης. Το προκαθορισμένο επίπεδο μεγέθυνσης του χάρτη, δηλαδή το επίπεδο στο οποίο βρίσκεται ο χάρτης όταν ο χρήστης συνδεθεί στο σύστημα, είναι το επίπεδο 7.

Για να εφαρμοστεί το πραγματικό μέγεθος του πλοίου πάνω στο χάρτη, εκτελέστηκαν οι παρακάτω υπολογισμοί. Για παράδειγμα, για να υπολογιστεί το μήκος του πλοίου θεωρούμε:

- x : πραγματικό μήκος πλοίου στον χάρτη
- πραγματικό μήκος πλοίου = 250m
- σταθερό μήκος πλοίου = 180m
- σταθερό μήκος πλοίου στον χάρτη = 21.11

Εάν για 21.11 μονάδες αντιστοιχούν σε 180m στην πραγματικότητα, πόσες μονάδες αντιστοιχούν για 250m?

$$x = (250 * 21.11) / 180$$

Το ίδιο ισχύει και για την περίπτωση που υπολογίζεται και το πλάτος του πλοίου. Αναλόγως, το ίδιο εφαρμόζεται ξεχωριστά για κάθε επίπεδο μεγένθυσης όπου αλλάζει η κλίμακα του χάρτη και παράλληλα αλλάζει και το σταθερό μήκος και πλάτος του πλοίου πλέον αλλάζει και ξαναυπολογίζεται με τη χρήση της κλίμακας.

Σημειώνεται ότι το πραγματικό μέγεθος του πλοίου ξεκινά να εφαρμόζεται από το επίπεδο μεγένθυσης 15 και πιο πάνω λόγω της δυσκολίας αναπαράστασης των πραγματικών διαστάσεων του πλοίου σε μακρινή απόσταση από την γη. Για πιο μικρά επίπεδα χρησιμοποιούνται οι διαστάσεις που έχουν υπολογισθεί για το επίπεδο 14.

3.2 Λειτουργία Χρονικής Μετάβασης

3.2.1 GUI Components

Για την υλοποίηση του δεύτερου στόχου, χρειάστηκε να προστεθεί πάνω στο σύστημα ένα σύνολο από GUI components όπου θα δίνουν συγκεκριμένα τη δυνατότητα στον χρήστη να μεταβεί σε στιγμιότυπα του συστήματος στο παρελθόν και έπειτα να μπορεί να τον επαναφέρει σε ζωντανή μετάδοση. Πιο συγκεκριμένα, προστέθηκε στο σύστημα ένα date component όπου ο χρήστης θα εισάγει την ημερομηνία στην οποία επιθυμεί να μεταβεί, ένα time component στο οποίο ο χρήστης θα εισάγει την ώρα σε ώρες και λεπτά, ένα κουμπί το οποίο θα ενημερώνει το σύστημα για την εκτέλεση της μετάβασης και ακόμα ένα κουμπί το οποίο θα είναι υπεύθυνο για την επαναφορά του συστήματος

(Εικόνα 21). Μεταγενέστερα, προστέθηκαν στο σύστημα δύο επιπλέον κουμπιά τα οποία δίνουν τη δυνατότητα στον χρήστη να μεταβεί μπρος-πίσω στο χρόνο ανά 20 λεπτά (Εικόνα 22).



Εικόνα 21: Datetime Component



Εικόνα 22: Time Offset

3.2.2 Προβολή Στιγμιότυπου του Συστήματος στο Παρελθόν

Αρα, εισάγοντας την ημερομηνία και την ώρα που ο χρήστης θέλει να μεταφερθεί και πατώντας το κουμπί GO, το σύστημα παίρνει σαν παράμετρο το συγκεκριμένο timestamp και την περνάει μέσα στον controller. Ο controller εκτελεί το ερώτημα για να πάρει τις πληροφορίες όλων των πλοίων με τη διαφορά ότι αυτή τη φορά θα ψάξει στη ΒΔ για τις πιο πρόσφατες πληροφορίες του κάθε πλοίου εντός των 20 λεπτών από το timestamp που του έδωσε ο χρήστης. Αξίζει να σημειωθεί ότι με το που πατήσει ο χρήστης το κουμπί GO, το σύστημα σταματάει την αυτόματη ανανέωση του κάθε 10 δευτερόλεπτα και πλέον δεν βρίσκεται αυτό σε ζωντανή μετάδοση.

Παράλληλα, αφού το σύστημα έχει μεταφερθεί σε μια χρονική στιγμή στο παρελθόν, εκτελεί ένα ερώτημα στη ΒΔ με σκοπό να εμφανίσει στον χρήστη όλες τις (προ)ειδοποιήσεις που έλαβαν χώρα εκείνη την χρονική στιγμή. Πιο συγκεκριμένα, εμφανίζει όλες τις ειδοποιήσεις για την είσοδο / έξοδο των πλοίων από μια περιοχή εντός της μίας ώρας και επίσης εμφανίζει όλες τις προειδοποιήσεις για πιθανή σύγκρουση δύο πλοίων εντός της μίας ώρας αλλά και τις πιθανές συγκρούσεις που επρόκειτο να γίνουν εντός των 20 λεπτών.

Επιπρόσθετα, όταν ο χρήστης πατήσει ένα από τα 2 κουμπιά για μετατόπιση +/- 20 λεπτών στο χρόνο, τότε το σύστημα εκτελεί ένα νέο ερώτημα προσθέτοντας / αφαιρώντας στην / από την παράμετρο του χρόνου 20 λεπτά και εμφανίζει όλα τα πλοία και όλες τις (προ)ειδοποιήσεις για τα πλοία εκείνης της χρονικής στιγμής. Σημειώνεται ότι, εάν ο χρήστης επιχειρήσει να εισάγει ημερομηνία η οποία είναι μεταγενέστερη της τρέχουσας ημερομηνίας, το σύστημα δεν θα εμφανίσει κάποιο πλοίο ή κάποια (προ)ειδοποίηση εκτός και αν υπάρχουν εντός της μίας ώρας.

3.3 Βελτιστοποίηση Αλγορίθμου Εισόδου/Εξόδου

Για την υλοποίηση του τρίτου στόχου, προηγήθηκε η σκέψη να δημιουργηθεί μία ενδιάμεση κατάσταση όπου θα λειτουργούσε σαν ένα ενδιάμεσο βήμα πριν την ολοκλήρωση της αποστολής της ειδοποίησης. Με αυτό τον σκοπό το σύστημα μπαίνει σε μια κατάσταση επαλήθευσης για την συμπεριφορά του κάθε πλοίου ξεχωριστά.

3.3.1 Υλοποίηση

Ετσι λοιπόν, χρησιμοποιήθηκαν τοπικές μεταβλητές ως μετρητές για την κάθε μία από τις περιπτώσεις (βλ.3.3.2) ξεχωριστά, όπου ο κάθε μετρητής για κάθε συνδυασμό πλοίου και περιοχής που έχει σχέση με το πλοίο, κρατά μια τιμή αποθηκευμένη. Αρχικά, όλοι οι μετρητές είναι αρχικοποιημένοι με 0. Κάθε φορά που ο αλγόριθμος θέλει να καταγράψει είσοδο / έξοδο ενός πλοίου σε / από μια περιοχή, προσθέτει τον αντίστοιχο μετρητή κατά ένα. Για να προχωρήσει ο αλγόριθμος στην καταγραφή της ειδοποίησης στην βάση, πρέπει ο αντίστοιχος μετρητής να ισούται με μία προκαθορισμένη τιμή που είναι αποθηκευμένη σε μία άλλη τοπική μεταβλητή. Κάθε στιγμή που ένας μετρητής αυξάνεται κατά ένα, οι υπόλοιποι μετρητές του συνδυασμού του πλοίου με την περιοχή μηδενίζονται.

Επιπρόσθετα, έχουν προστεθεί στον αλγόριθμο τοπικές μεταβλητές οι οποίες αποθηκεύουν το τελευταίο timestamp του πλοίου που έχει ανακτήσει από τη ΒΔ. Ο λόγος είναι ότι στην περίπτωση που το πλοίο δεν στείλει καινούργια AIS data στα 60 δευτερόλεπτα που χρειάζεται ο αλγόριθμος για να ανανεώσει τα data που ανακτά από τη ΒΔ, ο αλγόριθμος επαληθεύει την ίδια πρόθεση του πλοίου που είχε ήδη βάση των προηγούμενων δεδομένων. Άρα, για να αυξηθεί ο ανάλογος μετρητής για κάθε πλοίο,

πρέπει πρώτα το timestamp που είναι αποθηκευμένο στην τοπική μεταβλητή να είναι άνισο με το τελευταίο timestamp που θα ανακτήσει ο αλγόριθμος από τη βάση.

3.3.2 Εκτίμηση Αλγορίθμου

Για να εξεταστεί η αποδοτικότητα του βελτιστοποιημένου αλγορίθμου, ο αλγόριθμος εκτελέστηκε ξεχωριστά δύο φορές σε ένα timestamp, όπου υπήρχαν επαναλαμβανόμενες ειδοποιήσεις ότι ένα πλοίο μπαίνει από μια περιοχή, με τη μόνη διαφορά να είναι στην τοπική μεταβλητή που καθορίζει τις προσπάθειες που έχει ο αλγόριθμος για να αποθηκεύσει στη βάση μία ειδοποίηση. Για παράδειγμα, επιλέχθηκε η μελέτη συμπεριφοράς του πλοίου με mmsi=212872000 στις 02/05/2022 από τις 20:00 μέχρι τις 20:20. Το συγκεκριμένο πλοίο είχε την τάση να μπαίνει στην περιοχή DP WORLD RORO με id=24.

Στην μία περίπτωση ο αλγόριθμος εκτελέστηκε για $n=1$, όπου n η τιμή της τοπικής μεταβλητής, δηλαδή κάθε φορά που ο αλγόριθμος ανιχνεύει συμπεριφορές των πλοίων να εισέρχονται / εξέρχονται σε / από μια περιοχή, ο αλγόριθμος αποθηκεύει την ειδοποίηση στη βάση δεδομένων. Στην δεύτερη περίπτωση ο αλγόριθμος εκτελέστηκε για $n=2$, όπου τώρα πρέπει να επαληθευθεί η συμπεριφορά του πλοίου πριν να την αποθηκεύσει ο αλγόριθμος στη ΒΔ. Και στις δύο περιπτώσεις ο αλγόριθμος έτρεξε για 40 λεπτά. Για να υπάρξει αποκλειστικότητα στις καταγραφές του αλγορίθμου, τα αποτελέσματα είχαν αποθηκευτεί σε μία τοπική προσωρινή ΒΔ. Έχει παρατηρηθεί ότι στην δεύτερη περίπτωση έχουν καταγραφεί πιο λίγες επαναλαμβανόμενες ειδοποιήσεις σε σχέση με την πρώτη περίπτωση που αυτό δηλώνει ότι ο αλγόριθμος έχει βελτιστοποιηθεί.

Πιο συγκεκριμένα, για $n=1$ ο αλγόριθμος κατέγραψε συνολικά 15 warnings από τα 15 σήματα που έχουν σταλεί. Λογικό, αφού ο αλγόριθμος βρίσκεται σε τέτοιο βαθμό ευαισθησίας που λάθος σήματα καταγράφονται σαν ειδοποιήσεις στην ΒΔ. Για $n=2$ ο αλγόριθμος κατέγραψε συνολικά 6 warnings μέσα στη ΒΔ από τα 15 σήματα που έχουν σταλεί. Αυτό σημαίνει ότι σε αυτή την περίπτωση ο αλγόριθμος κάνει κάτι σαν φιλτράρισμα των σημάτων που στέλνονται με σκοπό να επιβεβαιωθεί η συμπεριφορά τους, γι αυτό και έχει παρατηρηθεί σημαντική μείωση στις ειδοποιήσεις που έχουν καταγραφεί.

3.3.3 Ψευδοκώδικας Αλγορίθμου

Παρακάτω ακολουθεί ο ψευδοκώδικας του βελτιστοποιημένου αλγορίθμου. Σημειώνεται ότι τα κομμάτια που έχουν προστεθεί για την βελτιστοποίηση είναι μορφοποιημένα με έντονη γραφή.

```
1 while(1)
2   query -> select area, ship_signal, ship in area
3   foreach(areas as area)
4     foreach(vessel_last_coordinate as vessel)
5       if(length=0 and breadth=0)
6         if(vessel is in ship_in_area type=0)
7           if(not is_in_polygon(vessel,area))
8             if(last_timestamps[vessel][area] != new_timestamp)
9               counter00[vessel][area]++
10              counter01[vessel][area]=0
11              counter02[vessel][area]=0
12              counter03[vessel][area]=0
13            endif
14            if(counter00==variable)
15              insert_in_warning[]->data
16              delete_ship_in_area[]->vessel,area
17              counter00[vessel][area]=0
18            endif
19          endif
20        else
21          if(is_in_polygon(vessel,area))
22            if(last_timestamps[vessel][area] != new_timestamp)
23              counter00[vessel][area]=0
24              counter01[vessel][area]++
25              counter02[vessel][area]=0
26              counter03[vessel][area]=0
27            endif
28            if(counter01==variable)
29              insert_ship_in_area[]->vessel,area
30              insert_in_warning[]->data
31              counter01[vessel][area]=0
32            endif
33
```

```

34     else
35         if(area==danger)
36             if(vessel is in ship_in_area type=1)
37                 if(not is_ship_close_of_area(vessel,area))
38                     if(last_timestamps[vessel][area] != new_timestamp)
39                         counter00[vessel][area]=0
40                         counter01[vessel][area]=0
41                         counter02[vessel][area]++
42                         counter03[vessel][area]=0
43                     endif
44                     if(counter02==variable)
45                         query->delete ship_close_of_danger_area(vessel,area)
46                         counter02[vessel][area]=0
47                     endif
48                 endif
49             else
50                 if(is_ship_close_of_area(vessel,area))
51                     if(last_timestamps[vessel][area] != new_timestamp)
52                         counter00[vessel][area]=0
53                         counter01[vessel][area]=0
54                         counter02[vessel][area]=0
55                         counter03[vessel][area]++
56                     endif
57                     if(counter03==variable)
58                         query->insert ship_close_of_danger_area(vessel,area)
59                         query->insert warning()
60                         counter03[vessel][area]=0
61                     endif
62                 endif
63             endif
64         endif
65     endif
66     endif
67     else
68         "same process but this time you check four marks and not one
69         because you have length and breadth"
70     endif
71 endforeach
72 endforeach
73 endwhile

```

3.3.3.1 Περιγραφή Ψευδοκώδικα

Γραμμή 2: Ανακτώνται από τη ΒΔ όλα τα AIS data των πλοίων, όλες οι περιοχές του συστήματος και όλα τα πλοία που ανήκουν σε συγκεκριμένες περιοχές.

Γραμμή 3-5: Εξετάζεται για κάθε περιοχή ξεχωριστά σε συνδυασμό με το σύνολο των τελευταίων συντεταγμένων του πλοίου εάν το μήκος και το πλάτος των πλοίων είναι μηδέν.

Γραμμή 6-7: Στην περίπτωση που το μήκος και το πλάτος δεν είναι μηδέν, ο αλγόριθμος ελέγχει εάν το συγκεκριμένο πλοίο για τη συγκεκριμένη περιοχή βρίσκεται εντός σύμφωνα με τα δεδομένα από τη ΒΔ. Εάν ναι, εξετάζεται κατά πόσο το συγκεκριμένο πλοίο δεν βρίσκεται ακόμη εντός της συγκεκριμένης περιοχής.

Γραμμή 8-13: Εδώ αρχικά, γίνεται έλεγχος κατά πόσον η καταγραφή που έχει λάβει τελευταία φορά το σύστημα από τη ΒΔ είναι διαφορετική από αυτή που είχε λάβει τώρα. Αυτό γίνεται ελέγχοντας τα timestamps των δύο records και εάν είναι διαφορετικά ο αλγόριθμος προχωράει κανονικά. Υπάρχουν 4 δισδιάστατοι πίνακες οι οποίοι αυξάνονται ανάλογα με την περίπτωση που υφίσταται.

- counter00[vessel][area]: Στην περίπτωση που το πλοίο βρισκόταν εντός της περιοχής και τώρα βρίσκεται εκτός
- counter01[vessel][area]: Στην περίπτωση που το πλοίο βρισκόταν κοντά στην περιοχή και τώρα βρίσκεται εντός
- counter02[vessel][area]: Στην περίπτωση που το πλοίο βρισκόταν κοντά σε μία επικίνδυνη περιοχή και τώρα δεν βρίσκεται κοντά της.
- counter03[vessel][area]: Στην περίπτωση που το πλοίο δε βρισκόταν αλλά τώρα βρίσκεται κοντά σε επικίνδυνη περιοχή

Για την προκειμένη περίπτωση, αυξάνεται ο counter00[vessel][area] κατά ένα.

Γραμμή 14-18: Εάν ο counter00[vessel][area] είναι ίσος με τη μεταβλητή variable, της οποίας η τιμή είναι καθορισμένη από πριν, ο αλγόριθμος προχωράει στην δημιουργία ειδοποίησης ότι το πλοίο βγήκε εκτός περιοχής και στη συνέχεια διαγράφει το record από τη βάση που μαρτυρούσε ότι το πλοίο αυτό βρισκόταν εντός της συγκεκριμένης περιοχής.

Απευθείας, ο counter00[vessel][area] μηδενίζεται έτσι ώστε όταν κληθεί να ο αλγόριθμος να ξαναεξετάσει στο μέλλον την ίδια περίπτωση πλοίου και περιοχής, ο αντίστοιχος μετρητής να είναι αρχικοποιημένος στο μηδέν.

Γραμμή 21-32: Από την άλλη, εάν το πλοίο βρισκόταν κοντά στην συγκεκριμένη περιοχή και τώρα ανακαλύφθηκε ότι βρίσκεται εντός, ξανα ελέγχει κατά πόσο τα δύο timestamps που προαναφέρθηκαν είναι τα ίδια. Εάν είναι διαφορετικά, αυτή την φορά ο μετρητής counter01[vessel][area] αυξάνεται κατά ένα. Στη συνέχεια, ελέγχει εάν ο μετρητής είναι ίσος με την προκαθορισμένη τιμή της μεταβλητής variable. Εάν ισούνται, δημιουργεί ειδοποίηση για είσοδο του πλοίου στην περιοχή, καταγράφει στον αντίστοιχο πίνακα το πλοίο μαζί με την περιοχή στην οποία βρίσκεται και μηδενίζει τον μετρητή counter01[vessel][area].

Γραμμή 35-37: Εάν η περιοχή που εξετάζεται θεωρείτε επικίνδυνη και το πλοίο είναι κατεγραμμένο στη βάση ότι βρίσκεται κοντά στη συγκεκριμένη περιοχή, εξετάζεται κατά πόσο πλέον βρίσκεται ακόμα κοντά στην επικίνδυνη περιοχή.

Γραμμή 38-47: Επαναλαμβάνεται παρόμοια διαδικασία με τις γραμμές 21-32. Ελέγχει τα δύο timestamps και εάν είναι διαφορετικά, αυξάνεται ο counter02[vessel][area] κατά ένα. Εξετάζει κατά πόσο ο μετρητής ισούται με την προκαθορισμένη τιμή της μεταβλητής variable και εάν ισούται διαγράφει το πλοίο από τον πίνακα με τα πλοία που βρίσκονται κοντά σε επικίνδυνες περιοχές και όπως προηγουμένως μηδενίζει τον counter02[vessel][area].

Γραμμή 50-61: Εάν τώρα, το πλοίο δε βρισκόταν κοντά σε κάποια επικίνδυνη περιοχή και τώρα ανακαλύφθηκε ότι βρίσκεται κοντά, επαναλαμβάνει την ίδια διαδικασία με τις γραμμές 38-47 με τη διαφορά ότι αυτή τη φορά αυξάνεται κατά ένα ο counter03[vessel][area]. Εάν στη συνέχεια ισούται με την τιμή της προκαθορισμένης μεταβλητής variable, προσθέτει μια καινούργια εγγραφή στον πίνακα με τα πλοία που βρίσκονται κοντά σε επικίνδυνες περιοχές, δημιουργεί μια νέα ειδοποίηση και μηδενίζει τον counter03[vessel][area].

Γραμμή 68-69: Από εδώ και στο εξής ισχύει η ίδια διαδικασία με πριν με την διαφορά ότι έχουμε το μήκος και το πλάτος των πλοίων. Αξίζει να σημειωθεί ότι έχουμε και τέσσερις νέους δισδιάστατους πίνακες οι οποίοι είναι:

- counter10[vessel][area]

- counter11[vessel][area]
- counter12[vessel][area]
- counter13[vessel][area]

Η αιτία για την δημιουργία των καινούργιων πινάκων ήταν η αποφυγή τυχόν συγκρούσεων μεταξύ των δύο περιπτώσεων.

3.4 Βελτιστοποίηση Αλγόριθμου Ανίχνευσης Συγκρούσεων

Για την υλοποίηση του τέταρτου στόχου, κρίθηκε αναγκαίο ο υπολογισμός μιας πιθανής σύγκρουσης μεταξύ δύο πλοίων να χρησιμοποιεί σαν παράμετρο την ταχύτητα των δύο πλοίων και όχι μόνο την απόσταση και τον χρόνο που θα φτάσουν στο σημείο σύγκρουσης.

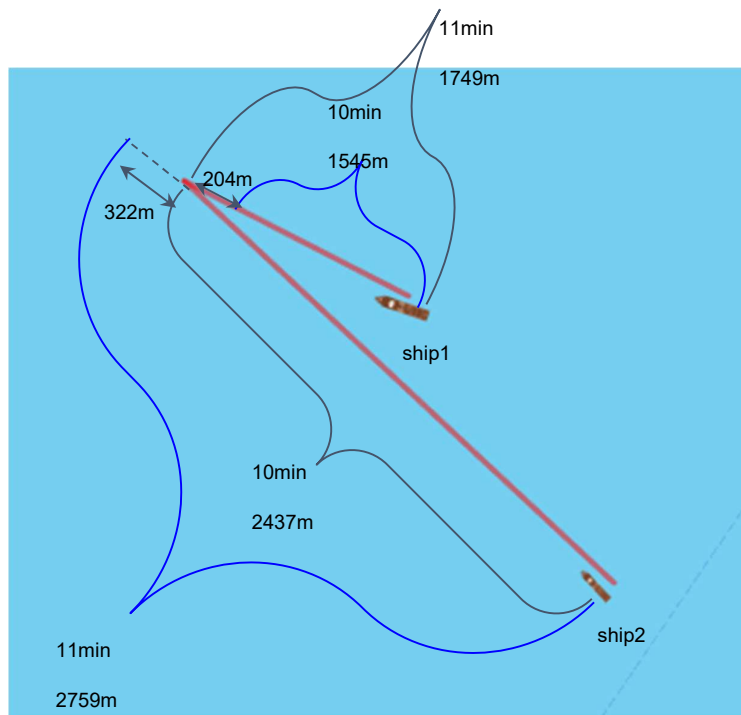
3.4.1 Υλοποίηση

Γι αυτό, χρησιμοποιήσαμε τους χρόνους που χρειάζονται τα δύο πλοία για να φτάσουν στο σημείο σύγκρουσης και τις αποστάσεις τους από αυτό. Στη συνέχεια, για το κάθε πλοίο ξεχωριστά υπολογίζεται η απόσταση που θα διανύσει στο χρόνο που χρειάζεται το άλλο πλοίο να φτάσει στο σημείο σύγκρουσης. Έτσι, υπολογίζονται δύο νέα σημεία για την τοποθεσία των πλοίων μετά από τον αντίστοιχο χρόνο. Έπειτα, υπολογίζει ξεχωριστά την απόσταση των δύο αυτών σημείων με το σημείο σύγκρουσης και κρατά τη μικρότερη από τις δύο. Τέλος, συγκρίνει την τιμή αυτή με την απόσταση ασφαλείας, η οποία είναι τιμή που έχει προκαθορισθεί από πριν. Εάν η μικρότερη απόσταση είναι μικρότερη από την απόσταση τότε υπάρχει μεγάλη πιθανότητα τα δύο αυτά πλοία να συγκρουστούν μεταξύ τους.

3.4.2 Εκτίμηση Αλγορίθμου

Για να εξεταστεί η αποδοτικότητα του αλγορίθμου, έχουν καταγραφεί στη ΒΔ για διάρκεια 24 ωρών όλα τα collision warnings με και χωρίς τη βελτιστοποίηση του αλγορίθμου. Συνολικά έχουν καταγραφεί 404 warnings χωρίς τη βελτιστοποίηση και

386 με τη βελτιστοποίηση. Επίσης, έχουν μελετηθεί δύο πραγματικές περιπτώσεις όπου έχει καταγραφεί στη ΒΔ ότι τα δύο πλοία πρόκειται να συγκρουστούν μεταξύ τους. Στην πρώτη περίπτωση, ο αλγόριθμος έκανε τους αντίστοιχους υπολογισμούς και υπολόγισε ότι όταν το ένα πλοίο φτάσει στο σημείο σύγκρουσης, η απόσταση του από το άλλο πλοίο ήταν μικρότερη από την απόσταση ασφαλείας. Αρα, σωστά πρέπει να καταγραφεί προειδοποίηση για σύγκρουση των δύο πλοίων.

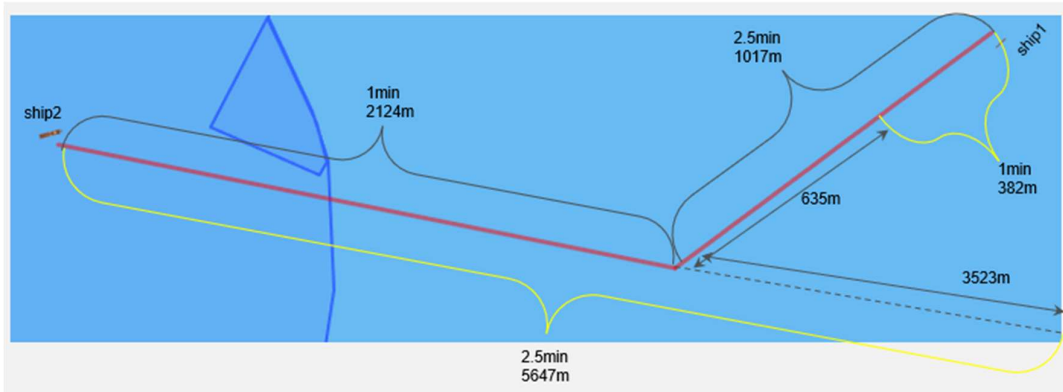


Εικόνα 23: Παράδειγμα Πιθανής Σύγκρουσης 1.1

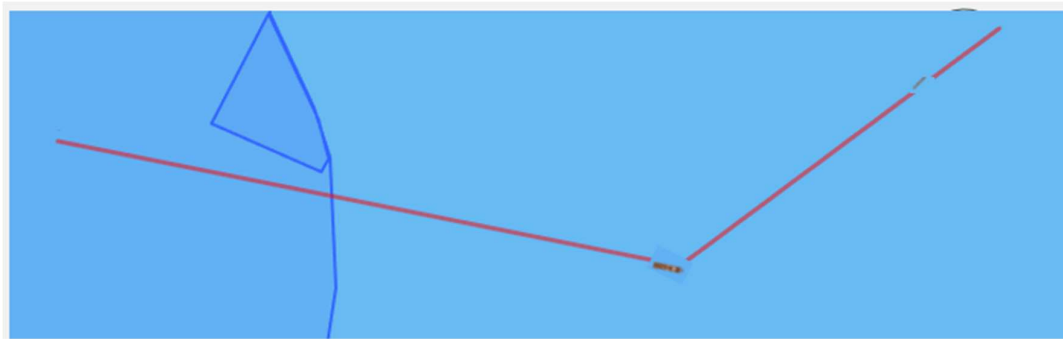


Εικόνα 24: Παράδειγμα Πιθανής Σύγκρουσης 1.2

Στη δεύτερη περίπτωση, ο αλγόριθμος έκανε τους αντίστοιχους υπολογισμούς και υπολόγισε ότι όταν το ένα πλοίο φτάσει στο σημείο σύγκρουσης, η απόσταση του από το άλλο πλοίο ήταν μεγαλύτερη από την απόσταση ασφαλείας. Αρα, σωστά δεν πρέπει να καταγραφεί προειδοποίηση για σύγκρουση των δύο πλοίων.



Εικόνα 25: Παράδειγμα Πιθανή σύγκρουσης 2.1



Εικόνα 26: Παράδειγμα Πιθανή σύγκρουσης 2.2

3.4.3 Ψευδοκώδικας Αλγορίθμου

Παρακάτω ακολουθεί ο ψευδοκώδικας του βελτιστοποιημένου αλγορίθμου. Σημειώνεται ότι τα κομμάτια που έχουν προστεθεί για την βελτιστοποίηση είναι μορφοποιημένα με έντονη γραφή.

```
1 while(1)
2 {
3   delete->all past conflict ->(conflict_table)
4   vessel=select all vessels, save zones areas
5   foreach(vessel)
6     if(ship is in save zone)
7       delete from vessel
8     endif
9   endforeach
10
11  vessel->sort_by_latitude
12
13  foreach(vessel)
14    while(next_vessel.latitude - vessel.latitude<0.5)
15      if(next_vessel and vessel move)
16        if(intersectionOfLines(next_vessel and vessel))
17          vessel_time = time_which_this_ship_go_to_mark_of_intersection
18          next_vessel_time =
time_which_this_sihp_go_to_mark_of_intersection
19          value = vessel_time - next_vessel_time
20          time_of_conflict = (vessel_time + next_vessel_time)/2
21          if(value>=-2 and value<=2)
22            d1 = distance_between_vessel_and_mark_of_intersection
23            d2 = distance_between_next_vessel_and_mark_of_intersection
24            d3 = distance_which_vessel_covers_in_ship2_time
25            d4 = distance_which_next_vessel_covers_in_ship1_time
26            d5 = |d1-d3|
27            d6 = |d2-d4|
28            if(min(d5,d6)<safety_distance)
29              if(this 2 vessel are not in conflict table)
30                insert->conflict_table, conflict_warning_table
31              endif
32            endif
33          endif
34        endif
35      elseif(next_vessel move and vessel dont move)
36        vessel.make_point_to_segment
37        if(intersectionOfLines(next_vessel and vessel))
38          next_vessel_time =
time_which_this_ship_go_to_mark_of_intersection
39          if(this 2 vessel arent in conflict_table)
```

```
40             insert->conflict_table, conflict_warning_table
41         endif
42     endif
43     endif
44     endwhile
45 endforeach
46
47 }
```

3.4.3.1 Περιγραφή Ψευδοκώδικα

Γραμμή 3-9: Διαγράφονται όλες οι προηγούμενες προειδοποιήσεις για συγκρούσεις και ανακτώνται όλα τα vessels και τα safe zones από τη ΒΔ

Γραμμή 11: Ταξινομούνται όλα τα vessels βάση του γεωγραφικού τους πλάτους.

Γραμμή 13-14: Για κάθε vessel εξετάζεται αν η απόσταση μεταξύ τους είναι μικρότερη από 50km με σκοπό να αφαιρεθούν από τη λίστα περιπτώσεις που τα vessels είναι απομακρυσμένα μεταξύ τους.

Γραμμή 15-16: Εξετάζεται η μία από τις δύο περιπτώσεις όπου και τα δύο πλοία κινούνται και επίσης ελέγχεται από μία ειδική συνάρτηση κατά πόσο οι γραμμές των δύο πλοίων συγκρούονται.

Γραμμή 17-21: Υπολογίζει τον χρόνο που χρειάζονται τα δύο vessels να φτάσουν στο σημείο σύγκρουσης. Εάν η διαφορά των δύο χρόνων είναι μικρότερος από δύο λεπτά ο αλγόριθμος προχωράει στον έλεγχο.

Γραμμή 22-28: Υπολογίζει τις αποστάσεις των δύο vessels από το σημείο σύγκρουσης και την απόσταση που θα διανύσει το κάθε ένα ξεχωριστά στο χρόνο του άλλου vessel. Στη συνέχεια υπολογίζει τη διαφορά των δύο αποστάσεων ξεχωριστά με το σημείο σύγκρουσης και ελέγχει κατά πόσο η μικρότερη απόσταση από τις δύο είναι μικρότερη από την απόσταση ασφαλείας

Γραμμή 29-31: Εάν είναι μικρότερη και δεν υπάρχει ήδη κατεγραμμένο warning στη ΒΔ, ο αλγόριθμος καταγράφει το warning στη ΒΔ.

4 Συζήτηση

Ο βασικός σκοπός της εργασίας αυτής ήταν η γενική βελτιστοποίηση του συστήματος σε πολλαπλούς τομείς, είτε στην αναπαράσταση είτε στη λειτουργικότητα είτε στην ευχρηστία του συστήματος. Βάση των αποτελεσμάτων που έφεραν οι συγκεκριμένες αλλαγές που εφαρμόσαμε στο σύστημα (βλ. 3), φαίνεται ότι οι αλλαγές αυτές στηρίζουν τις αρχικές ιδέες που υπήρξαν για την επίλυση κάποιων συγκεκριμένων προβλημάτων. Άρα, σε σχέση με πιο παλιές εκδόσεις του συστήματος παρατηρείται ότι δεν έγιναν αλλαγές οι οποίες μεταμόρφωσαν σε μεγάλο βαθμό το σύστημα, αλλά έγιναν αλλαγές σε σημεία που είτε διαγνώστηκε κάποιο πρόβλημα είτε προηγήθηκε ανάγκη για κάτι περισσότερο.

5 Συμπεράσματα

Τα συμπεράσματα που βγαίνουν από τη συγκεκριμένη εργασία είναι το πόσο σημαντικό είναι αυτός που εργάζεται πάνω σε ένα σύστημα να έχει μάθει το σύνολικό πακέτο του συστήματος σε ένα μεγάλο βαθμό. Η έλλειψη τριβής και η έλλειψη βασικών γνώσεων είναι συνήθως η αιτία για να προκληθούν διάφορα προβλήματα όπως αδυναμία και καθυστέρηση στην υλοποίηση των στόχων. Άρα, βασικός παράγοντας είναι η άρτια χρήση των εργαλείων που χρησιμοποιεί το σύστημα για αποδοτικότερη εργασία.

Για μελλοντική εργασία, υπάρχουν αρκετά σημεία στο σύστημα που μπορούν να δεχθούν βελτιστοποίηση, ίσως σε ποιο λεπτομερή σημεία. Πιο συγκεκριμένα, θα μπορούσε να δημιουργηθεί ένας καινούργιος αλγόριθμος ο οποίος θα εξετάζει τις περιπτώσεις όπου ένα πλοίο θα περνάει μέσα από μια μικρή περιοχή και θα βγαίνει απευθείας. Επίσης, οι 2 αλγόριθμοι έχουν την προοπτική να βελτιστοποιηθούν σε ακόμη μεγαλύτερο βαθμό έτσι ώστε να έχουν υψηλό βαθμό ακρίβειας και να μην είναι τόσο ευαίσθητοι.

6 Πίνακες και Διαγράμματα

6.1 Πίνακες

Πίνακας 1 : Position Report A

	Όνομα ιδιότητας	Περιγραφή
1.	Type	Αναγνωριστικό μηνύματος (1, 2, 3)
2.	Repeat Indicator	Μας δείχνει πόσες φορές ένα μήνυμα έχει επαναληφθεί
3.	mmsi	Maritime Mobile Service Identity
4.	County_code	Δηλώνει την χώρα στην οποία το πλοίο είναι εγγεγραμμένο
5.	Nav_status	Δηλώνει την κατάσταση του πλοίου (σε κίνηση, αγκυροβολημένο)
6.	ROT	Rate Of Turn, δεξιά ή αριστερά, από 0 μέχρι 720° ανά λεπτό
7.	SOG	Speed over ground, ταχύτητα με την οποία ταξιδεύει το πλοίο
8.	Position accuracy	1 = ψηλό, 0 = χαμηλό
9.	Longitude	Το γεωγραφικό μήκος του πλοίου
10.	Latitude	Το γεωγραφικό πλάτος του πλοίου
11.	COG	Course Over Ground
12.	True heading	Από 0 μέχρι 359°
13.	Timestamp	Το timestamp στο οποίο εκδόθηκε η αναφορά
14.	Maneuver indicator	Περιέχει μόνο null τιμές
15.	Raim_flag	Περιέχει μόνο null τιμές
16.	Comm_state	Περιέχει μόνο null τιμές
17.	Rot over range	Περιέχει μόνο null τιμές
18.	Sync state	Περιέχει μόνο null τιμές
19.	Slot timeout	Περιέχει μόνο null τιμές
20.	Slot number	Περιέχει μόνο null τιμές
21.	Slot offset	Περιέχει μόνο null τιμές
22.	Utc hour	Περιέχει μόνο null τιμές
23.	Utc min	Περιέχει μόνο null τιμές
24.	Keep_flag	Περιέχει μόνο null τιμές
25.	Slots to allocate	Περιέχει μόνο null τιμές

26.	Received_station	Περιέχει μόνο null τιμές
27.	Received_timestamp	Το timestamp στο οποίο έχουμε λάβει το σήμα

Πίνακας 2: Position Report B

	Όνομα ιδιότητας	Περιγραφή
1.	Type	Αναγνωριστικό μηνύματος (1, 2, 3)
2.	Repeat Indicator	Μας δείχνει πόσες φορές ένα μήνυμα έχει επαναληφθεί
3.	mmsi	Maritime Mobile Service Identity
4.	County_code	Δηλώνει την χώρα στην οποία το πλοίο είναι εγγεγραμμένο
5.	SOG	Speed over ground, ταχύτητα με την οποία ταξιδεύει το πλοίο
6.	Position accuracy	1 = ψηλό, 0 = χαμηλό
7.	Longitude	Το γεωγραφικό μήκος του πλοίου
8.	Latitude	Το γεωγραφικό πλάτος του πλοίου
9.	COG	Course Over Ground
10.	True heading	Από 0 μέχρι 359°
11.	Timestamp	Το timestamp στο οποίο εκδόθηκε η αναφορά
12.	Unit_flag	0 = Class B SOTDMA unit 1 = Class B "CS" unit
13.	Display_flag	0 = No display available 1 = Equipped with integrated display displaying Message
14.	Dsc_flag	0 = Not equipped with DSC function 1 = Equipped with DSC function (dedicated or time-shared)
15.	Band_flag	0 = Capable of operating over the upper 525 kHz band of the marine band 1 = Capable of operating over the whole marine band
16.	M22_flag	0 = No frequency management 1 = Frequency management
17.	Mode_flag	0 = Station operating in autonomous and continuous mode = default 1 = Station operating in assigned mode
18.	Raim	Receiver autonomous integrity monitoring flag of electronic position fixing device
19.	Commstate_flag	Communication state selector flag

20.	Commstate_cs_fill	SOTDMA communication state
21.	Received_timestamp	Το timestamp στο οποίο έχουμε λάβει το σήμα

Πίνακας 3: Countries

	Όνομα ιδιότητας	Περιγραφή
1.	ID	Μοναδικός αριθμός ως αναγνωριστικό
2.	Code	Κωδικός χώρας
3.	Name	Πλήρες όνομα χώρας

Πίνακας 4: Navigation Status

	Όνομα ιδιότητας	Περιγραφή
1.	ID	Μοναδικός αριθμός ως αναγνωριστικό
2.	Message	Δείχνει την κατάσταση του πλοίου π.χ. σε κίνηση, αγκυροβολημένο

Πίνακας 5: Vessel

	Όνομα ιδιότητας	Περιγραφή
1.	ID	Περιέχει ένα μοναδικό ID
2.	IMO	International Maritime Organization, περιέχει ένα μοναδικό αριθμό για κάθε vessel
3.	VesselId	Περιέχει το VesselId, το οποίο περιέχει το IMO
4.	mmsi	Περιέχει το mmsi του κάθε vessel
5.	VesselName	Περιέχει το όνομα του κάθε vessel π.χ. ANAFI WARRIOR
6.	VesselType	Περιέχει τον τύπο του κάθε vessel π.χ. TANKER

7.	Call_sign	Περιέχει την μοναδική αλφαριθμητική ταυτότητα του vessel
8.	Flag	Περιέχει τη χώρα στην οποία είναι εγγεγραμμένο το vessel π.χ. BE για το Βέλγιο
9.	BuiltYear	Περιέχει το έτος κατασκευής του vessel
10.	DataSource	Περιέχει τη πηγή από την οποία έλαβε τα δεδομένα
11.	PhotoURL	Περιέχει το URL της φωτογραφίας του vessel
12.	BuildPlace	Περιέχει την τοποθεσία κατασκευής του vessel
13.	Breadth	Περιέχει το πλάτος του vessel
14.	Length	Περιέχει το μήκος του vessel
15.	Draught	Περιέχει μόνο null τιμές
16.	Owner	Περιέχει μόνο null τιμές
17.	Manager	Περιέχει μόνο null τιμές
18.	DwtSummer	Περιέχει μόνο null τιμές
19.	DisplacementSummer	Περιέχει μόνο null τιμές
20.	FuelConsumption	Περιέχει μόνο null τιμές
21.	MaxSpeed	Περιέχει μόνο null τιμές
22.	ServiceSpeed	Περιέχει μόνο null τιμές
23.	LiquidOil	Περιέχει μόνο null τιμές

Πίνακας 6: Entered /Left Area Warnings

	Όνομα ιδιότητας	Περιγραφή
1.	ID	Μοναδικός αριθμός ως αναγνωριστικό
2.	Mmsi	Περιέχει το mmsi του κάθε vessel
3.	Area_id	Περιέχει το κώδικο της κάθε περιοχής
4.	Type_of_warning	Περιέχει τον τύπο της ειδοποίησης (1, 2, 3, 4)
5.	Received_timestamp	Το timestamp στο οποίο έχουμε λάβει την ειδοποίηση

Πίνακας 7: Collision Warnings

	Όνομα ιδιότητας	Περιγραφή
1.	ID	Μοναδικός αριθμός ως αναγνωριστικό

2.	Mmsi_1	Περιέχει το mmsi του πρώτου vessel
3.	Mmsi_2	Περιέχει το mmsi του δεύτερου vessel
4.	lat	Περιέχει το γεωγραφικό πλάτος
5.	lom	Περιέχει το γεωγραφικό μήκος
6.	Received_timestamp	Το timestamp στο οποίο έχουμε λάβει την προειδοποίηση

Πίνακας 8: Collision Destination

	Όνομα ιδιότητας	Περιγραφή
1.	ID	Μοναδικός αριθμός ως αναγνωριστικό
2.	Mmsi_1	Περιέχει το mmsi του πρώτου vessel
3.	Mmsi_2	Περιέχει το mmsi του δεύτερου vessel
4.	Received_timestamp	Το timestamp στο οποίο έχουμε λάβει την προειδοποίηση

Πίνακας 9: Ship in areas

	Όνομα ιδιότητας	Περιγραφή
1.	ID	Μοναδικός αριθμός ως αναγνωριστικό
2.	Mmsi	Περιέχει το mmsi του κάθε vessel
3.	Area_id	Περιέχει το κώδικο της κάθε περιοχής
4.	Type of warning	Περιέχει τον τύπο της ειδοποίησης (1, 2, 3, 4)

Πίνακας 10: Areas

	Όνομα ιδιότητας	Περιγραφή
1.	ID	Μοναδικός αριθμός ως αναγνωριστικό
2.	Description	Περιέχει την περιγραφή της περιοχής
3.	Name	Περιέχει το όνομα της περιοχής
4.	Type	Περιέχει τον τύπο της περιοχής
5.	Active	Μας δείχνει κατά πόσο είναι ενεργή η περιοχή

Πίνακας 11: Area Types

	Όνομα ιδιότητας	Περιγραφή
1.	ID	Μοναδικός αριθμός ως αναγνωριστικό
2.	Name	Περιέχει το όνομα της περιοχής
3.	Description	Περιέχει την περιγραφή της περιοχής
4.	Type_of_message	Περιέχει τον τύπο του μηνύματος για την συγκεκριμένη περιοχή

Πίνακας 12: Ship categories

	Όνομα ιδιότητας	Περιγραφή
1.	ShipType	Περιέχει τον τύπο του πλοίου
2.	ShipCategory	Περιέχει την κατηγορία του πλοίου
3.	AvgBreadth	Περιέχει το πλάτος του πλοίου ανά μέσο όρο
4.	AvgLength	Περιέχει το μήκος του πλοίου ανά μέσο όρο
5.	DateTime_Updated	Περιέχει το timestamp που ενημερώθηκε η συγκεκριμένη ιδιότητα

Πίνακας 13: Vessel Label A

	Όνομα ιδιότητας	Περιγραφή
1.	mmsi	Περιέχει το mmsi του κάθε vessel
2.	Flag	Περιέχει τη χώρα στην οποία είναι εγγεγραμμένο το vessel π.χ. BE για το Βέλγιο
3.	Lng	Δείχνει το γεωγραφικό μήκος
4.	Lat	Δείχνει το VesselId, το οποίο περιέχει το IMO
5.	Nav status	Δείχνει την κατάσταση του πλοίου π.χ. σε κίνηση, ακυροβλημένο
6.	Speed	Δείχνει την ταχύτητα με την οποία ταξιδεύει το πλοίο
7.	Course	Δείχνει την κατεύθυνση του σε μοίρες
8.	VesselName	Περιέχει το όνομα του κάθε vessel π.χ. ANAFI WARRIOR
9.	VesselType	Περιέχει τον τύπο του κάθε vessel π.χ. TANKER
10.	BuildYear	Περιέχει το έτος κατασκευής του vessel

11.	Breadth	Περιέχει το πλάτος του vessel
12.	Length	Περιέχει το μήκος του vessel
13.	Received_timestamp	Το timestamp στο οποίο έχουμε λάβει την προειδοποίηση
14.	PhotoURL	Περιέχει το URL της φωτογραφίας του vessel

Πίνακας 14: Vessel Label Type B

	Όνομα ιδιότητας	Περιγραφή
1.	mmsi	Περιέχει το mmsi του κάθε vessel
2.	Flag	Περιέχει τη χώρα στην οποία είναι εγγεγραμμένο το vessel π.χ. BE για το Βέλγιο
3.	Lng	Δείχνει το γεωγραφικό μήκος
4.	Lat	Δείχνει το VesselId, το οποίο περιέχει το IMO
5.	Speed	Δείχνει την ταχύτητα με την οποία ταξιδεύει το πλοίο
6.	Course	Δείχνει την κατεύθυνση του σε μοίρες
7.	Breadth	Περιέχει το πλάτος του vessel
8.	Length	Περιέχει το μήκος του vessel
9.	Received_timestamp	Το timestamp στο οποίο έχουμε λάβει την προειδοποίηση
10.	PhotoURL	Περιέχει το URL της φωτογραφίας του vessel

ΕΠΙΛΟΓΟΣ

Το AISafety είναι ένα σύστημα το οποίο καθιστάται απαραίτητο για την ασφάλεια και τον έλεγχο στη θάλασσα. Ως εκ τούτου είναι σημαντικό το σύστημα να προσφέρει όσο το δυνατόν περισσότερες υπηρεσίες στους ανθρώπους που εργάζονται στη θάλασσα και για σκοπούς ασφάλειας αλλά και για καλύτερη επικοινωνία.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] *Aisafety.cut.ac.cy*. [Online]. Available: <https://aisafety.cut.ac.cy/>. [Accessed: 21-May-2022].
- [2] “The use and usefulness of AIS Data,” *SSPA*. [Online]. Available: <https://www.sspa.se/the-use-and-usefulness-of-AIS-data>. [Accessed: 21-May-2022].
- [3] “The PHP framework for web artisans,” *Laravel*. [Online]. Available: <https://laravel.com/docs/9.x>. [Accessed: 21-May-2022].
- [4] “An open-source JavaScript library for interactive maps,” *Leaflet*. [Online]. Available: <https://leafletjs.com/>. [Accessed: 21-May-2022].
- [5] *JavaScript.com*. [Online]. Available: <https://www.javascript.com/>. [Accessed: 21-May-2022].

ΠΑΡΑΡΤΗΜΑ Ι

Εργαλεία

Τα εργαλεία που χρησιμοποιήθηκαν στην εργασία είναι τα εξής:

1. Visual Studio Code για την συγγραφή του κώδικα
2. PHPMysqlAdmin για την διαχείριση της ΒΔ
3. XAMPP για την παροχή ενός LAMP environment