

**ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ**  
**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

Πληροφοριακό σύστημα συλλογής, διαχείρισης και απεικόνισης σε πραγματικό χρόνο περιβαλλοντικών δεδομένων κατά την πορεία μετακινούμενων οχημάτων



**Κωνσταντίνος Αρετάκης**

Εξεταστική επιτροπή:

Επιβλέπων: Κωνσταντίνος Καλαϊτζάκης

Καθηγητής: Γεώργιος Σταυρακάκης

Επίκουρος Καθηγητής: Σπύρος Παπαευθυμίου

Χανιά, 2017

## Περίληψη

Η εξέλιξη του πολιτισμού ανέκαθεν συνοδεύονταν από αυξημένες ανάγκες μεταφοράς. Σήμερα παρά ποτέ οι ανάγκες αυτές έχουν εκτοξευθεί με αποτέλεσμα την διαρκή αύξηση και των μέσων μεταφοράς.

Η ραγδαία αυτή αύξηση του αριθμού των μέσων που καθημερινά μετακινούνται στην υφήλιο έχει άμεσες συνέπειες στην κλιματική αλλαγή και συνεπώς είναι απαραίτητη η άμεση εποπτεία τους για τον αποτελεσματικότερο έλεγχο των επιδράσεων τους.

Στα πλαίσια αυτής της διπλωματικής σχεδιάστηκε και υλοποιήθηκε μια διαδικτυακή εφαρμογή για την απεικόνιση σε διαδραστικό γεωγραφικό χάρτη περιβαλλοντικών δεδομένων που συλλέγονται από ροές δεδομένων δικτύων αισθητήρων κατά την πορεία μετακινούμενων οχημάτων.

Η ανάπτυξη της εφαρμογής έγινε με γνώμονα την δυνατότητα προσθήκης αυθαίρετου αριθμού ροών δεδομένων, περιβαλλοντικών συνιστωσών και ειδών μετακινούμενων οχημάτων για την αποτελεσματικότερη ανάλυση των περιβαλλοντικών επιπτώσεων κατά την πορεία των τελευταίων.

## Ευχαριστίες

Αρχικά θα ήθελα να εκφράσω τις ευχαριστίες μου στους καθηγητές μου για την βοήθεια τους στην ολοκλήρωση της εργασίας μου.

Κατόπιν θα ήθελα να ευχαριστήσω τον Επίκουρο καθηγητή κ. Σπύρο Παπαευθυμίου και τον υποψήφιο διδάκτορα Γιώργο Νικολουδάκη για τη βοήθεια και την υποστήριξη τους κατά την εκπόνηση της εργασίας μου.

Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου για την κατανόησή τους.

# Περιεχόμενα

Περίληψη.....	2
Ευχαριστίες.....	3
Λίστα Εικόνων .....	6
Εισαγωγή .....	7
Συνεισφορά εργασίας .....	9
Δομή της Διατριβής .....	9
Κεφάλαιο 1 .....	11
Σχεδίαση και ανάπτυξη εφαρμογής.....	11
Λειτουργικές απαιτήσεις εφαρμογής .....	11
Προδιαγραφές ανάπτυξης της εφαρμογής.....	12
Απόδοση .....	13
Κλιμάκωση.....	13
Συντηρησιμότητα .....	13
Επεκτασιμότητα .....	13
Πρότυπα & τεχνικές ανάπτυξης εφαρμογών.....	14
Τεχνική σχεδίασης αρθρωτού (modular) προγραμματισμού.....	14
Πρότυπο σχεδίασης MVC.....	14
Τεχνική ORM.....	16
Τεχνική Caching βάσης δεδομένων.....	17
Τεχνική διαχείρισης εξαρτήσεων .....	18
Εργαλεία ανάπτυξης.....	18
CakePHP.....	19
Python.....	20
Google Maps API .....	20
Κεφάλαιο 2 .....	22
Τα κέντρα δεδομένων .....	22
Αναλυτική επεξήγηση του συστήματος AIS .....	23
Αναλυτική επεξήγηση του συστήματος ADS-B. ....	25
Κεφάλαιο 3 .....	28
Η εφαρμογή.....	28
Υπηρεσία συλλογής δεδομένων σε πραγματικό χρόνο .....	28
Web Crawler .....	32

Συλλογή περιβαλλοντικών δεδομένων .....	34
Εφαρμογή WEB για την απεικόνιση σε πραγματικό χρόνο των περιβαλλοντικών δεδομένων.....	36
Κεφάλαιο 4 .....	42
Συμπεράσματα χρήσης και λειτουργική επιβεβαίωση της εφαρμογής .....	42
Αρχεία Καταγραφής .....	42
Γεωγραφική Κάλυψη .....	43
Εξαγωγή Δεδομένων .....	45
Στατιστικά Χρήσης .....	45
Πρόσβαση από Κινητές Συσκευές Responsive Design .....	46
Χρήση βάσεως Δεδομένων και εκτεταμένο μοντέλο οντοτήτων-συσχετισμών (EE-R μοντέλο).....	47
Κεφάλαιο 5 .....	48
Μελλοντικές επεκτάσεις .....	48
Βιβλιογραφία.....	50

## Λίστα Εικόνων

Εικόνα 1: Διαχρονική εξέλιξη του μεταφερομένου φορτίου ανά κατηγορία πλοίου.....	8
Εικόνα 2: Κατηγοριοποίηση εκπομπών αέριων θερμοκηπίου στην ΕΕ για το έτος 2014.....	8
Εικόνα 3: Σχεδιάγραμμα Συστήματος.....	12
Εικόνα 4: Αρχιτεκτονική MVC.....	15
Εικόνα 5: ORM Mapping.....	16
Εικόνα 6: Database Caching.....	17
Εικόνα 7: Ροή δεδομένων προς το πληροφοριακό σύστημα.....	22
Εικόνα 8: Τυπικό Πακέτο δεδομένων AIS.....	25
Εικόνα 9: Δεδομένα Εισόδου μετά την Αποκωδικοποίηση.....	25
Εικόνα 10: Αποκωδικοποίηση δεδομένων συστήματος Ads-b.....	27
Εικόνα 11: Δεδομένα εισόδου πληροφοριακού συστήματος.....	27
Εικόνα 12: Τμήμα log αρχείου της υπηρεσίας συλλογής δεδομένων.....	29
Εικόνα 13 Config.ini.....	30
Εικόνα 14 Ενημέρωση κατάστασης εκτέλεσης της εφαρμογής.....	30
Εικόνα 15 Αναπαράσταση Μοντέλου για την Αποθήκευση Δεδομένων Κίνησης Πλοίων.....	31
Εικόνα 16 Αποτύπωση του Μοντέλου Κίνησης στην Βάση Δεδομένων.....	31
Εικόνα 17: Δείγμα πεδίων δεδομένων σάρωσης ιστοσελίδας.....	33
Εικόνα 18: Αναπαράσταση Μοντέλου για την Αποθήκευση Μεταδεδομένων.....	34
Εικόνα 19: Αποτύπωση του Μοντέλου Μεταδεδομένων στην Βάση Δεδομένων.....	34
Εικόνα 20: Απάντηση Openweather σε JSON μορφή.....	35
Εικόνα 21: Απάντηση κέντρου δεδομένων σε JSON μορφή.....	35
Εικόνα 22: Αναπαράσταση Μοντέλου για την Αποθήκευση Περιβαλλοντικών Δεδομένων.....	35
Εικόνα 23: Αποτύπωση του Μοντέλου Περιβαλλοντικών Δεδομένων στην Βάση Δεδομένων.....	36
Εικόνα 24: Γενική Εικόνα Εφαρμογής.....	37
Εικόνα 25 : Λειτουργία αναζήτησης πλοίου με αυτόματη συμπλήρωση.....	38
Εικόνα 26: Λειτουργία εστίασης εφαρμογής στο επιλεγμένο πλοίο.....	39
Εικόνα 27: Λειτουργία εξαγωγής δεδομένων.....	39
Εικόνα 28: Λειτουργία αναδυόμενου παραθύρου με πληροφορίες.....	40
Εικόνα 29: Λειτουργία εμφάνισης πορείας πλοίου.....	40
Εικόνα 30: Αρχείο Καταγραφής Υπηρεσίας Υπηρεσία εφαρμογής Web.....	42
Εικόνα 31: Αρχείο Καταγραφής Υπηρεσίας συλλογής Μεταδεδομένων Κινουμένων Μέσων.....	43
Εικόνα 32: Αρχείο Καταγραφής Υπηρεσίας Περιβαλλοντολογικών Δεδομένων.....	43
Εικόνα 33: Αρχείο Καταγραφής Υπηρεσίας συλλογής δεδομένων.....	43
Εικόνα 34: Χάρτης Κάλυψης συστήματος AIS.....	44
Εικόνα 35: Χάρτης Κάλυψης συστήματος Ads-b.....	44
Εικόνα 36: Δείγμα αρχείου εξαγωγής δεδομένων csv.....	45
Εικόνα 37: Responsive Design.....	46
Εικόνα 38: Τάξη Μεγέθους Δεδομένων σε σχεσιακή βάση.....	47
Εικόνα 39: E-R σχεσιακής βάσης Δεδομένων.....	47

# Εισαγωγή

Η κλιματική αλλαγή είναι από τα πλέον σημαντικά προβλήματα παγκοσμίως του σύγχρονου ανθρώπου με άμεσες συνέπειες τόσο στην υγεία του όσο και στην ποιότητα ζωής του γενικότερα. Ως κλιματική αλλαγή εννοούμε οποιαδήποτε μεταβολή που παρατηρείται στο κλίμα με το πέρασμα του χρόνου, είτε οφείλεται σε φυσικές επιδράσεις, είτε προκαλείται από την ανθρώπινη δραστηριότητα.

Όλα τα επιστημονικά δεδομένα σχετικά με την κλιματική αλλαγή δείχνουν ότι υπάρχει άμεση ανάγκη για μια συλλογική προσπάθεια των κυβερνήσεων, των επιχειρήσεων και των ιδιωτών για την αντιμετώπιση της κλιματικής αλλαγής. Αυτό απαιτεί μεγάλες και συνεχείς μειώσεις των εκπομπών αερίων του θερμοκηπίου σε παγκόσμιο επίπεδο και την ισχυρή διεθνή δέσμευση.

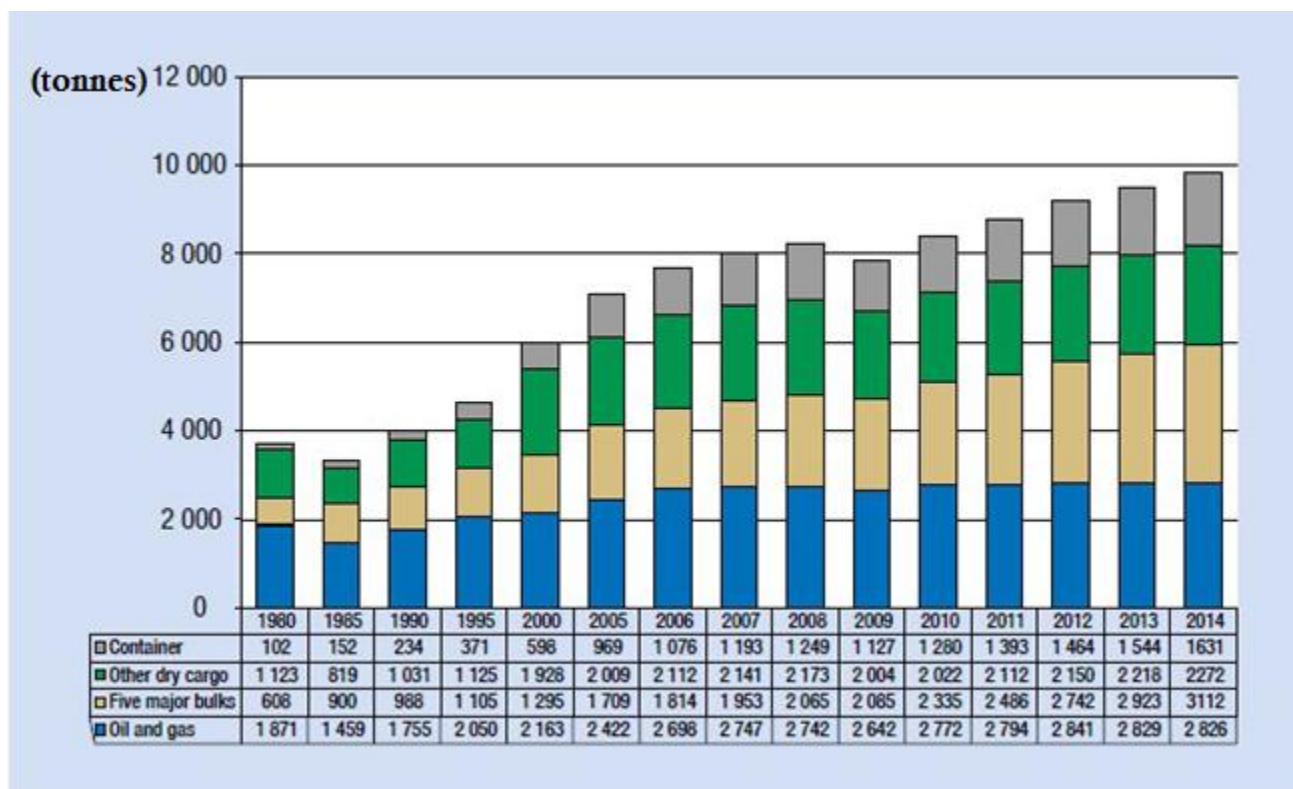
Η ΕΕ έχει προ πολλού ταχθεί υπέρ της ανάγκης περιορισμού της υπερθέρμανσης του πλανήτη στο ανώτατο όριο των 2°C. Η επιτακτική αυτή ανάγκη αναγνωρίζεται τώρα από τη διεθνή κοινότητα. Η Ευρωπαϊκή Ένωση βασίζει τις πολιτικές της σε ισχυρά επιστημονικά επιχειρήματα. Πιστεύει ότι πρέπει να αποτελεί παράδειγμα όσον αφορά τη δράση για την αντιμετώπιση της κλιματικής αλλαγής, μέσω δεσμευτικών στόχων για τα κράτη μέλη και πρωτοβουλιών.

Μεταξύ των άλλων επιπτώσεων του φαινομένου της υπερθέρμανσης του πλανήτη είναι και η συχνότερη εμφάνιση ακραίων καιρικών φαινομένων. Για πρώτη φορά μάλιστα, σε αυτή την έκθεση αξιολόγησης, γίνεται λόγος για την αύξηση της έντασης των τροπικών καταιγίδων καθώς και άλλα ακραία φαινόμενα όπως οι καύσωνες και οι ξηρασίες.

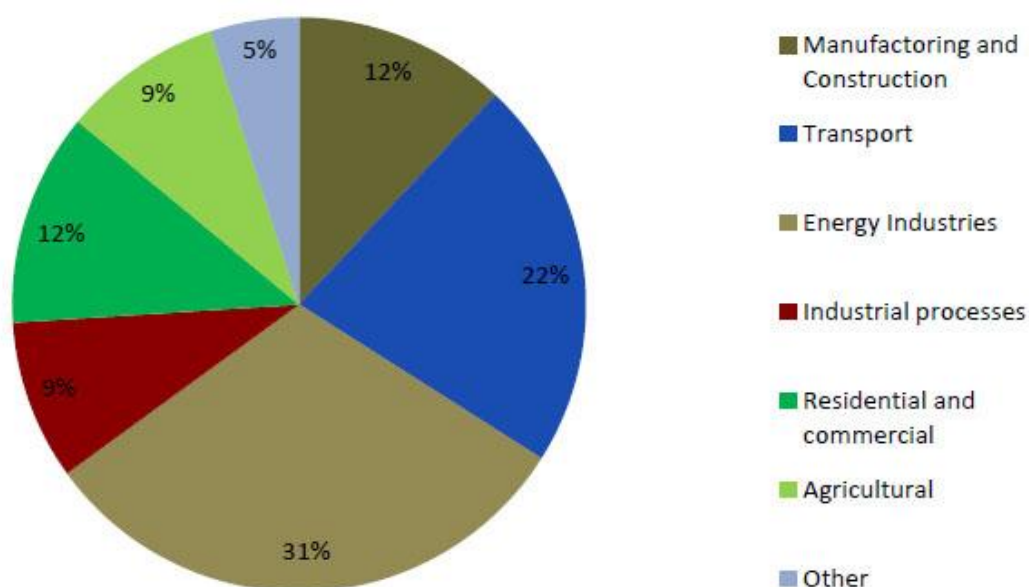
Όλα αυτά τα ακραία καιρικά φαινόμενα θα επιφέρουν και ακραίες αλλαγές σε διαφορετικά σημεία του πλανήτη. Σύμφωνα με τις προβλέψεις που παρατίθενται στη μελέτη του Εθνικού Αστεροσκοπείου Αθηνών για τις επιπτώσεις της κλιματικής αλλαγής στον Ελλαδικό χώρο, ο ρυθμός αύξησης της μέσης στάθμης της θάλασσας στη Μεσόγειο εκτιμάται σε 5 εκ./δεκαετία μέχρι το 2100. Αν αυτό το σενάριο επαληθευτεί, τότε μερικές παραθαλάσσιες περιοχές μπορεί να εξαφανιστούν λόγω εισχώρησης της θάλασσας στην ξηρά ή λόγω διάβρωσης του εδάφους. Οι περιοχές της Μεσογείου που αναμένεται να εμφανίσουν το μεγαλύτερο πρόβλημα είναι το Δέλτα του Νείλου, η Βενετία και η Θεσσαλονίκη.

Τα παραπάνω καθιστούν σαφή την άμεση ανάγκη για έλεγχο και μείωση των εκπομπών αερίων παγκοσμίως. Ένα σημαντικό ποσοστό των ρύπων οφείλεται στην αύξηση των μέσων μεταφοράς που διασχίζουν τον παγκόσμιο χάρτη καλύπτοντας τις αυξημένες ανάγκες για αγαθά του σύγχρονου ανθρώπου.

Όπως φαίνεται και στην Εικόνα 1 τα τελευταία χρόνια υπάρχει συνεχής αύξηση των θαλάσσιων μεταφορών όπως αποτυπώνεται από την αύξηση του φορτίου (σε εκατομμύρια τόνους) που μεταφέρθηκε διεθνώς [1].



Εικόνα 1: Διαχρονική εξέλιξη του μεταφερομένου φορτίου ανά κατηγορία πλοίου



Εικόνα 2: Κατηγοριοποίηση εκπομπών αέριων θερμοκηπίου στην ΕΕ για το έτος 2014



Η αύξηση αυτή του αριθμού των μέσων μεταφοράς έχει άμεσες επιδράσεις στο περιβάλλον εξαιτίας των ρύπων που εκπέμπουν κατά την μετακίνησή τους.

Ο καταμερισμός των εκπομπών αερίων του θερμοκηπίου μόνο για την ευρωπαϊκή ένωση κατά το έτος 2014 φαίνεται στην Εικόνα 2.

Όπως φαίνεται οι μεταφορές είναι η δεύτερη σημαντικότερη πηγή εκπομπών. Τα παραπάνω στοιχεία φανερώνουν την ανάγκη ανάπτυξης εργαλείων για την αποτελεσματική εποπτεία και τον έλεγχο των περιβαλλοντικών επιπτώσεων κατά την μετακίνηση των μέσων μεταφοράς.

## **Συνεισφορά εργασίας**

Η εργασία αυτή έχει ως στόχο την συλλογή και την απεικόνιση σε πραγματικό χρόνο δεδομένων κίνησης καθώς και περιβαλλοντικών δεδομένων κατά την πορεία μετακινούμενων οχημάτων μέσω διαδικτυακής εφαρμογής.

Η εφαρμογή αντλεί τα δεδομένα της από κέντρα δεδομένων (data centers) τα οποία αναμεταδίδουν σε πραγματικό χρόνο δεδομένα κίνησης πλοίων και περιβαλλοντικών μεταβλητών όπως συλλέγονται από δίκτυα αισθητήρων τοποθετημένα στο λιμάνι της Σούδας και του Ηρακλείου.

Η απεικόνιση των απαραίτητων πληροφοριών γίνεται μέσω μιας φιλικής προς τον χρήστη ιστοσελίδας με την βοήθεια διαδραστικού γεωγραφικού χάρτη και ένα σύνολο λειτουργικών στοιχείων για την αποτελεσματικότερη χρήση της πληροφορίας που παρέχεται.

Επιπλέον η εφαρμογή παρέχει την δυνατότητα επέκτασής της για την απεικόνιση διαφορετικών τύπων μετακινούμενων οχημάτων και περιβαλλοντικών μεταβλητών από απεριόριστο αριθμό data centers και για διαφορετικά είδη κινούμενων μέσων.

Απώτερος στόχος είναι η παροχή στον ενδιαφερόμενο της απαραίτητης υποδομής και εργαλείων για την άμεση και αποτελεσματική εποπτεία, ανάλυση και έλεγχο της πορείας μετακινούμενων οχημάτων σε συνδυασμό με περιβαλλοντικές συνιστώσες της περιοχής στην οποία αυτά κινούνται.

## **Δομή της Διατριβής**

Η παρούσα διατριβή έχει δομηθεί κατά τέτοιο τρόπο ώστε ο αναγνώστης να λάβει μια σαφή εικόνα όλων των επί μέρους σταδίων της ανάπτυξης του πληροφοριακού συστήματος από την σχεδίαση μέχρι την υλοποίηση και την μελλοντική εξέλιξή του.

Στο πρώτο κεφάλαιο αναλύονται οι λειτουργικές απαιτήσεις της εφαρμογής. Κατόπιν

αναλύονται οι προδιαγραφές με την οποία σχεδιάστηκε και υλοποιήθηκε η εφαρμογή. Στην συνέχεια περιγράφονται τα σύγχρονα σχεδιαστικά πρότυπα και τεχνικές που υιοθετήθηκαν για να επιτευχθούν οι παραπάνω προδιαγραφές. Τέλος παρουσιάζονται τα εργαλεία ανάπτυξης στα οποία βασίστηκε η υλοποίηση της εφαρμογής στα πλαίσια που προαναφέρθηκαν.

Στο δεύτερο κεφάλαιο περιγράφεται η υποδομή των κέντρων δεδομένων από τα οποία λαμβάνονται τα δεδομένα που χειρίζεται η εφαρμογή. Περιγράφονται τα σχετικά πρωτόκολλα επικοινωνίας αναλυτικά και ο τρόπος ροής της πληροφορίας από τα κινούμενα μέσα προς το πληροφοριακό μας σύστημα.

Στο τρίτο κεφαλαίο αναλύονται οι συνιστώσες της εφαρμογής που υλοποιήθηκαν και από κοινού αποτελούν το πληροφοριακό μας σύστημα. Αρχικά παρουσιάζεται η υπηρεσία συλλογής των δεδομένων από τα κέντρα δεδομένων σε πραγματικό χρόνο. Περιγράφονται επίσης οι συνιστώσες της εφαρμογής για την συλλογή μεταδεδομένων από τρίτες πηγές για τον εμπλουτισμό των πληροφοριών που παρέχει η εφαρμογή μας. Αφού αναλυθεί ο τρόπος συλλογής των περιβαλλοντολογικών δεδομένων από την εφαρμογή μας εν συνεχεία παρουσιάζεται η διεπαφή του χρήστη στο πρόγραμμα περιήγησης (browser) και των δυνατοτήτων που παρέχει η εφαρμογή στον τελικό χρήστη.

Στο τέταρτο κεφάλαιο αναλύονται τα συμπεράσματα και τα αποτελέσματα από την λειτουργία της εφαρμογής και δίνονται προτάσεις σχετικά με πιθανές επεκτάσεις της εφαρμογής για την μελλοντική ανάπτυξη και εξέλιξή του πληροφοριακού συστήματος.

# Κεφάλαιο 1

## Σχεδίαση και ανάπτυξη εφαρμογής

Στο κεφάλαιο αυτό αναλύονται οι παράμετροι που λήφθηκαν υπόψη κατά την σχεδίαση και ανάπτυξη της εφαρμογής ώστε να δημιουργηθεί ένα ολοκληρωμένο πληροφοριακό σύστημα απεικόνισης περιβαλλοντικών δεδομένων.

Πιο συγκεκριμένα αναλύονται οι λειτουργικές απαιτήσεις της εφαρμογής, οι προδιαγραφές που οφείλει να τηρεί στα πλαίσια ενός σύγχρονου πληροφοριακού συστήματος καθώς και οι σύγχρονες αρχές και τεχνικές προγραμματισμού που ακολουθηθήκαν.

### Λειτουργικές απαιτήσεις εφαρμογής

Στην ενότητα αυτή περιγράφονται οι λειτουργικές απαιτήσεις της εφαρμογής οι οποίες και ορίζουν τις βασικές λειτουργικές μονάδες της και οριοθετούν την περαιτέρω σχεδίαση και ανάπτυξη της. Συνοπτικά οι λειτουργικές απαιτήσεις της εφαρμογής είναι οι εξής:

- συλλογή και επεξεργασία της ροής δεδομένων (data feed) που εκπέμπουν οι εξυπηρετητές των κέντρων δεδομένων
- αποθήκευση των επεξεργασθέντων δεδομένων σε βάση δεδομένων
- εφαρμογή διαδικτύου για την παρουσίαση των δεδομένων σε πραγματικό χρόνο μέσω browser
- απεικόνιση δεδομένων σε διαδραστικό γεωγραφικό χάρτη
- εξαγωγή ιστορικών δεδομένων σε φιλική μορφή προς τον τελικό χρήστη.

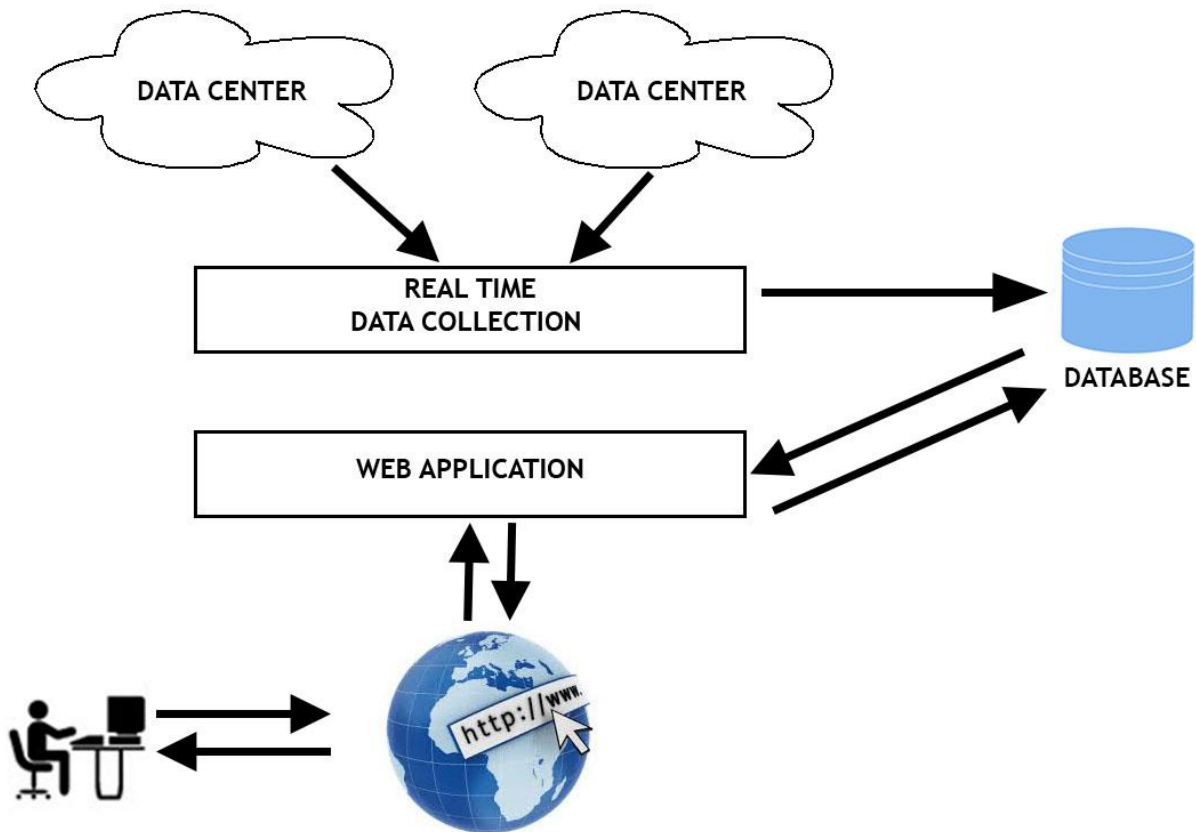
Στην Εικόνα 3 παρουσιάζεται η δομή της εφαρμογής με τις επιμέρους λειτουργικές μονάδες όπως αυτές ορίζονται από τις παραπάνω λειτουργικές απαιτήσεις.

Όπως φαίνεται και από την παραπάνω εικόνα για την συλλογή και επεξεργασία της ροής δεδομένων από κάθε server απαιτείται η ανάπτυξη μιας υπηρεσίας (service) η οποία επικοινωνεί μαζί του και διαβάζει συνεχώς τα δεδομένα που στέλνει. Η υπηρεσία είναι επίσης υπεύθυνη για την επεξεργασία και την αποθήκευση των δεδομένων που στέλνει ο server.

Η αποθήκευση των δεδομένων γίνεται σε σχεσιακή βάση δεδομένων στην οποία κατόπιν μπορεί να συνδεθεί η εφαρμογή διαδικτύου για την ανάκτηση και παρουσίαση τους. Μέσω της εφαρμογής διαδικτύου ο τελικός χρήστης μπορεί να δει τα δεδομένα σε πραγματικό χρόνο μέσω ενός απλού προγράμματος περιήγησης (browser).

Τα δεδομένα που λαμβάνει ο χρήστης μέσω του browser πρέπει να απεικονίζονται σε διαδραστικό γεωγραφικό χάρτη ώστε να μπορεί να δει εύκολα την θέση του μετακινούμενου οχήματος, το ίχνος της πορείας του, τα περιβαλλοντικά δεδομένα που έχουν ληφθεί κατά την πορεία του και σχετικές πληροφορίες όπως για παράδειγμα το όνομα του, τον τύπο του κτλ.

Τέλος ο χρήστης έχει την δυνατότητα εξαγωγής των δεδομένων με μορφοποίηση οριοθετημένου κειμένου για προσωπική χρήση και ενδεχόμενη περαιτέρω ανάλυση τους με άλλα προγράμματα που μπορούν να διαβάσουν την μορφοποίηση αυτή.



Εικόνα 3: Σχεδιάγραμμα - δομή προτεινόμενου συστήματος

## Προδιαγραφές ανάπτυξης της εφαρμογής

Η υλοποίηση κάθε ολοκληρωμένου πληροφοριακού συστήματος βασίζεται πάνω σε θεμελιώδεις αρχές ανάπτυξης εφαρμογών οι οποίες χαρακτηρίζουν τους ποιοτικούς της στόχους. Παρακάτω παρουσιάζονται οι αρχές ανάπτυξης που ακολουθηθήκαν κατά την υλοποίηση της παρούσας εφαρμογής και οι οποίες υπαγορεύουν την χρήση συγκεκριμένων προτύπων, τεχνικών και εργαλείων ανάπτυξης που παρουσιάζονται παρακάτω.

## **Απόδοση**

Η απόδοση μιας εφαρμογής αναφέρεται στην ταχύτητα με την οποία διεκπεραιώνει τα αιτήματα εργασιών που δέχεται από τους χρήστες της. Καθώς η απόδοση είναι το βασικό κριτήριο το οποίο απαιτεί κάποιος να πληροί μια εφαρμογή είναι απαραίτητο κατά την ανάπτυξη της να ληφθούν υπόψη όλες οι σύγχρονες τεχνικές που καθορίζουν την απόδοση της. Επιπρόσθετα, οι τεχνικές αυτές θα κρίνουν και την επιτυχία του επόμενου ποιοτικού στόχου της εφαρμογής όπως αυτός ορίζεται από την αρχή της κλιμάκωσης.

## **Κλιμάκωση**

Ο όρος κλιμάκωση στον τομέα ανάπτυξης πληροφοριακών συστημάτων αναφέρεται στην δυνατότητα απόκρισης του συστήματος όσο μεγαλώνει ο αριθμός των διεργασιών που καλείται να διεκπεραιώσει με την ελάχιστη δυνατή αύξηση των διαθέσιμων πόρων της. Έτσι, για παράδειγμα, μια εφαρμογή διαδικτύου με μεριμνά στην κλιμάκωση οφείλει να μπορεί να αποκριθεί χωρίς δραματική μείωση της απόδοσης της και χωρίς την χρήση επιπλέον πόρων είτε δέχεται αιτήσεις εξυπηρέτησης από 10 χρήστες είτε από 1000. Αντίθετα η μη τήρηση αυτής της αρχής θα οδηγούσε στην δραματική μείωση της απόδοσης της η θα απαιτούσε την αύξηση των διαθέσιμων πόρων της και κατά συνέπεια το κόστος συντήρησης της.

## **Συντηρησιμότητα**

Η συντηρησιμότητα αναφέρεται στο σύνολο των ενεργειών που γίνονται μετά την έναρξη λειτουργίας της εφαρμογής και αποσκοπούν στην βελτίωση της, στην διόρθωση σφαλμάτων και στην αφαίρεση παρωχημένων τεχνικών και τεχνολογιών. Οι ενέργειες αυτές οφείλουν να γίνονται το ταχύτερο δυνατόν και με το μικρότερο κόστος ώστε η εφαρμογή να διατηρείται στην ακμή της καθ' όλη την διάρκεια λειτουργίας της. Κατά συνέπεια απαιτείται μέριμνα κατά την ανάπτυξη της εφαρμογής ώστε να ενσωματωθούν όλες οι απαραίτητες τεχνικές και εργαλεία που θα επιτρέπουν την ταχύτερη και οικονομικότερη συντήρηση της.

## **Επεκτασιμότητα**

Η αρχή αυτή αναφέρεται στην δυνατότητα επέκτασης του πληροφοριακού συστήματος ώστε να μπορεί να ενσωματώσει νέες λειτουργίες και τεχνολογίες με όσο το δυνατόν λιγότερο κόστος. Η τήρηση ή μη της αρχής αυτής καθορίζει το κατά πόσο το σύστημα θα μπορέσει να ενσωματώσει γρήγορα και εύκολα νέες λειτουργίες ανταποκρινόμενο σε μελλοντικές απαιτήσεις από το σύστημα, καθώς και να εναρμονιστεί με νέες τεχνολογίες που θα το αποτρέψουν από το να καταστεί παρωχημένο και εν τέλει αδρανές.

## Πρότυπα & τεχνικές ανάπτυξης εφαρμογών

Στην προηγούμενη ενότητα ορίστηκαν οι ποιοτικοί στόχοι της εφαρμογής μέσω ενός συνόλου αρχών ανάπτυξης εφαρμογών. Σε αυτή την ενότητα αναλύονται τα πρότυπα και οι τεχνικές που χρησιμοποιήθηκαν για την επίτευξη αυτών των στόχων.

Η επιλογή του συνόλου των προτύπων και των τεχνικών που αναλύονται παρακάτω έγινε με γνώμονα τον κανόνα «Follow the best practices» (ακολουθώ τις καλύτερες τεχνικές) βάσει του οποίου για κάθε επιμέρους συνιστώσα ενός πληροφοριακού συστήματος επιλέγονται οι καλύτερες τεχνικές όπως αυτές έχουν αποκρυσταλλωθεί μέχρι σήμερα στον τομέα της ανάπτυξης πληροφοριακών συστημάτων.

### Τεχνική σχεδίασης αρθρωτού (modular) προγραμματισμού

Ο αρθρωτός (modular) προγραμματισμός είναι μια τεχνική σχεδίασης εφαρμογών με έμφαση στον διαχωρισμό της εφαρμογής σε ανεξάρτητα τμήματα (modules) τα οποία παρέχουν μια διακριτή λειτουργικότητα στην εφαρμογή και τα οποία αλληλεπιδρούν μεταξύ τους μέσω καλά ορισμένων κανόνων [2].

Τα βασικά πλεονεκτήματα του αρθρωτού προγραμματισμού είναι η επαναχρησιμοποίηση ενός module σε διάφορα σημεία της εφαρμογής, η μείωση του συνολικού μεγέθους της και κατ' επέκταση η μεγαλύτερη ευκολία στην συντήρηση του συστήματος.

Η ανάπτυξη μιας εφαρμογής με χρήση της τεχνικής του αρθρωτού προγραμματισμού είναι από τις πλέον βασικές αρχές στην ανάπτυξη εφαρμογών και η επιτυχία της εξέλιξης ενός συστήματος βασίζεται σε μεγάλο βαθμό στον σαφή και καλό ορισμό των modules του και της διεπαφής τους.

Για τους παραπάνω λόγους δόθηκε ιδιαίτερη έμφαση στην χρήση αυτής της τεχνικής για την ανάπτυξή της παρούσας εφαρμογής κάνοντας τις εξής επιλογές:

- χρήση προτύπου σχεδίασης MVC και των framework CakePHP και Django.
- χρήση τεχνική διαχείρισης εξαρτήσεων και το εργαλείο Composer.

Όπως θα περιγράψουμε και στις αντίστοιχες ενότητες τα παραπάνω πρότυπα και εργαλεία ευνοούν την ανάπτυξη της εφαρμογής σε modules, δίνοντας στην εφαρμογή μας και τα αντίστοιχα πλεονεκτήματα.

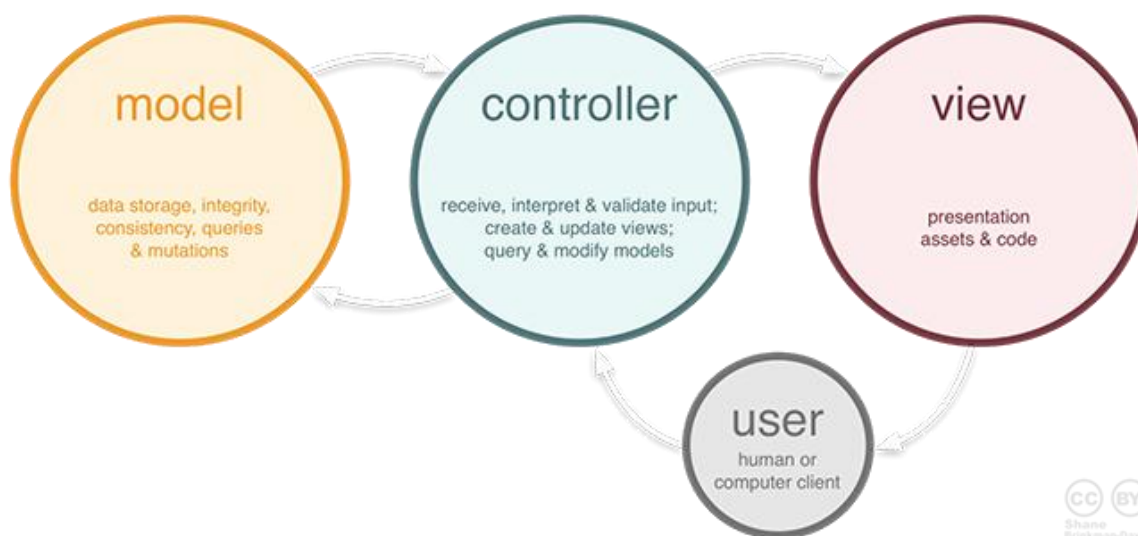
### Πρότυπο σχεδίασης MVC

Το πρότυπο Model - View - Controller (MVC) είναι ένα σχεδιαστικό πρότυπο για την δημιουργία πληροφοριακών συστημάτων το οποίο βασίζεται στον διαχωρισμό της

εφαρμογής σε 3 διακριτά μέρη (modules) τα οποία αλληλοσυνδέονται μεταξύ τους για να αποκριθούν στην αλληλεπίδραση του χρήστη με το πληροφοριακό σύστημα [3].

Τα 3 διακριτά μέρη, στα οποία οφείλει και το όνομα του το πρότυπο, είναι το Model (Μοντέλο), View (Παρουσίαση), Controller (Χειριστής) όπως φαίνεται και στην Εικόνα 4.

Το Model module είναι υπεύθυνο για την λογική της εφαρμογής που αφορά την επεξεργασία, την αποθήκευση και την ανάκτηση των πληροφοριών που είναι απαραίτητα για την ολοκλήρωση της ενέργειας του χρήστη. Το τμήμα αυτό δεν γνωρίζει το View τμήμα της εφαρμογής και η πληροφορία ρέει αμφίδρομα από το Model στον Controller αποσυνδέοντας έτσι εντελώς την παρουσίαση της εφαρμογής από το λειτουργικό της κομμάτι.



**Εικόνα 4:** Αρχιτεκτονική MVC

Το Controller module είναι υπεύθυνο για την επεξεργασία της ενέργειας του χρήστη και την ροή της πληροφορίας από το Model προς το View. Μέσω του Controller επιτυγχάνεται η διασύνδεση και των τριών module του προτύπου.

Το View module είναι υπεύθυνο για την παρουσίαση των δεδομένων στον τελικό χρήστη. Αντλεί την απαραίτητη πληροφορία από τον Controller όπως ήδη αναφέραμε και δεν γνωρίζει το Model τμήμα της εφαρμογής.

Ο παραπάνω εννοιολογικός διαχωρισμός της λογικής της εφαρμογής σε ανεξάρτητα modules με καλά ορισμένες αρμοδιότητες προσφέρει το σύνολο των πλεονεκτημάτων που αναφέραμε στο κεφάλαιο του αρθρωτού προγραμματισμού. Πιο συγκεκριμένα η απόζευξη της παρουσίασης από την λογική της εφαρμογής δίνει την δυνατότητα ανεξάρτητης ανάπτυξης μιας συνιστώσας της χωρίς να επηρεασθεί η άλλη.

Έτσι θα μπορούσαμε σε μια εφαρμογή να προχωρήσουμε σε επανασχεδιασμό της παρουσίασης προς τον χρήστη διατηρώντας ανέπαφη την λογική της εφαρμογής. Επίσης θα μπορούσαμε να εμπλουτίσουμε ή να βελτιώσουμε την λογική της εφαρμογής χωρίς να επηρεασθεί η παρουσίαση της.

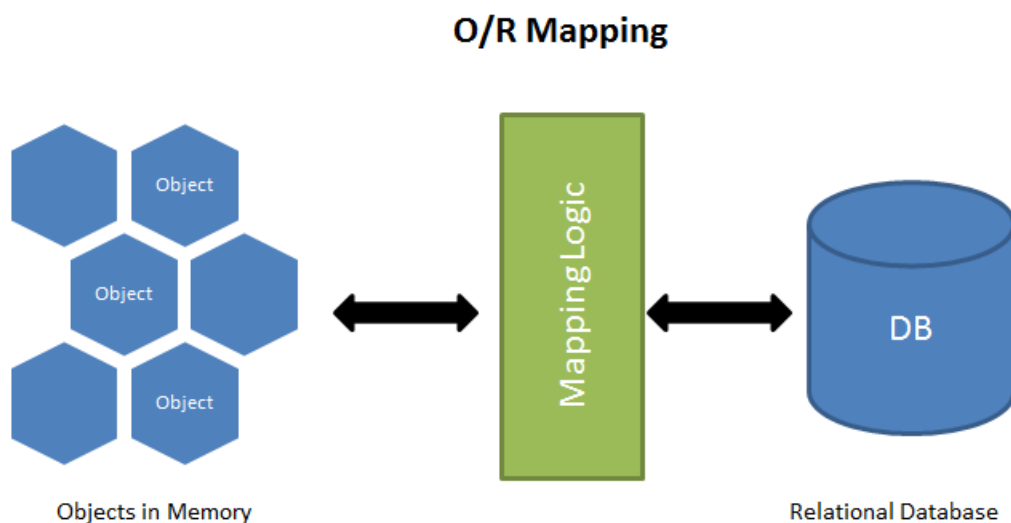
Η επιτυχία του προτύπου MVC φαίνεται και από το πλήθος δημοφιλών frameworks που το έχουν υιοθετήσει όπως Ruby on Rails, Django, Angular, CakePHP και πολλά άλλα. Στην παρούσα εφαρμογή το MVC πρότυπο χρησιμοποιήθηκε τόσο στην Web εφαρμογή μέσω του framework CakePHP όσο και στην υπηρεσία συλλογής δεδομένων από τους σταθμούς μέσω του Django framework.

Δομώντας την εφαρμογή μας στο πρότυπο αυτό εντάσσουμε το σύνολο των πλεονεκτημάτων που αναφέραμε και θέτουμε ταυτόχρονα στέρεα θεμέλια για την περαιτέρω εξέλιξη της βάσει των προδιαγραφών που έχουμε ορίσει.

### Τεχνική ORM

Το ORM (object relational mapping) είναι μια τεχνική προγραμματισμού για την μετατροπή δεδομένων μεταξύ ασύμβατων συστημάτων σε αντικειμενοστραφής γλώσσες προγραμματισμού [4].

Πιο συγκεκριμένα με αυτή την τεχνική ένα σύνολο μεταβλητών - τιμών τα οποία προέρχονται από κάποιο σύστημα, συνήθως μια βάση δεδομένων, απεικονίζονται σε ένα αντικείμενο (object) της αντικειμενοστραφούς γλώσσας που τα εισάγει.



Εικόνα 5: ORM Mapping

Η μετατροπή αυτή ουσιαστικά εισάγει ένα επίπεδο αφαίρεσης στην εφαρμογή και επιτρέπει την αναπαράσταση και αλληλεπίδραση των λειτουργικών μονάδων της με όρους χειρισμού αντικειμένων, αποσυνδέοντας την ταυτόχρονα από τις λεπτομέρειες ανάκτησης των δεδομένων από το εκάστοτε σύστημα διαχείρισης βάσεων δεδομένων όπως παρουσιάζεται και στην Εικόνα 5.

Η τεχνική ORM χρησιμοποιείται κατά κόρον στα MVC frameworks συμβάλλοντας στο modularization του προτύπου. Η σύζευξη της τεχνικής με την παρούσα εφαρμογή έγινε



στο πλαίσιο των MVC frameworks που χρησιμοποιήθηκαν καθώς χρησιμοποιείται τόσο από το CakePHP όσο και από το Django framework.

Η χρήση της τεχνικής αυτής βοήθησε στην υλοποίηση της λογικής της εφαρμογής μέσω του επιπέδου αφαίρεσης στην περιγραφή των λειτουργικών μονάδων που εισάγει.

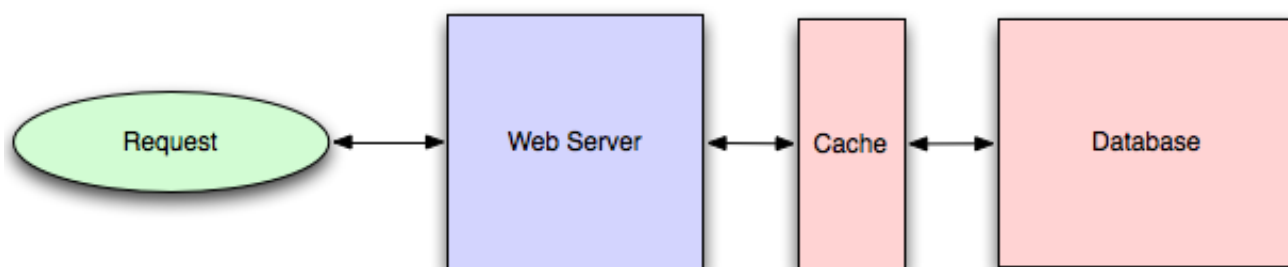
Έτσι για παράδειγμα ήταν εφικτό (με αυτή την τεχνική) να υλοποιηθεί η εφαρμογή δηλώνοντας ένα αντικείμενο τύπου Vessel με συγκεκριμένα πεδία όπως όνομα, αναγνωριστικό κτλ και να προχωρήσουμε στην λογική της χρησιμοποιώντας αυτά τα αντικείμενα αδιαφορώντας πως είναι αποθηκευμένα στην βάση δεδομένων.

Η τεχνική ORM είναι υπεύθυνη να μετατρέψει μια εγγραφή από την βάση δεδομένων σε ένα αντικείμενο τύπου Vessel. Επιπρόσθετα αναπτύσσοντας με αυτή την τεχνική επιτυγχάνουμε ευκολότερη επεκτασιμότητα και κλιμάκωση της εφαρμογής έχοντας αποσυνδέσει την λογική από το σχήμα και τους μηχανισμούς της βάσης δεδομένων και κατά συνέπεια μπορούμε να προχωρήσουμε σε αναβάθμιση τους αφήνοντας την λογική άθικτη και στην τεχνική ORM την μετατροπή των δεδομένων στα αντικείμενα της εφαρμογής.

### Τεχνική Caching Βάσης δεδομένων

Η τεχνική caching βάσης δεδομένων (database caching) χρησιμοποιείται κατά την ανάπτυξη εφαρμογών διαδικτύου για βελτίωσης την ταχύτητας ανάκτησης των δεδομένων από την βάση δεδομένων και συνεπώς την απόδοση της εφαρμογής, παίζοντας σημαντικό ρόλο στην σταθερότητα της σε ενδεχομένη κλιμάκωση της. Η τεχνική αυτή βασίζεται στην κατανομή του βάρους των ερωτημάτων σε πολλαπλά χαμηλού κόστους caching συστήματα όπως για παράδειγμα το memcached ή file cache.

Πιο συγκεκριμένα έστω ερώτημα προς την βάση το οποίο προέρχεται από μια ενέργεια του χρήστη και το οποίο επιστρέφει ένα σύνολο αποτελεσμάτων. Η τεχνική caching αποθηκεύει τα αποτελέσματα σε προσωρινή μνήμη (cache) και επιστρέφει το αποτέλεσμα από εκεί όταν ξαναζητηθεί παρακάμπτοντας την ερώτηση προς την βάση η οποία θα ήταν χρονοβόρα. Στην Εικόνα 6 φαίνεται η λογική της παραπάνω τεχνικής.



Εικόνα 6: Database Caching

Επιπλέον υπάρχει η δυνατότητα να ορισθεί χρόνος ζωής για τον οποίο το ερώτημα στην προσωρινή μνήμη είναι έγκυρο μετά το πέρας του οποίου το σύστημα αναζητά τα δεδομένα από την βάση. Στην παρούσα εφαρμογή η τεχνική του database caching αποτελεί μέρος του CakePHP framework υποστηρίζοντας διάφορα συστήματα caching από τα οποία έχουμε επιλέξει το file caching για ερωτήματα που θέλουμε να κρατήσουμε σε προσωρινή μνήμη.

Έχοντας στην διάθεση μας αυτή την τεχνική μπορούμε να επιταχύνουμε χρονοβόρα ερωτήματα όπως για παράδειγμα η εύρεση της πορείας ενός πλοίου αποθηκεύοντας στην cache την πρώτη φορά που θα ζητηθεί η συγκεκριμένη πορεία.

Μελλοντικές αναζήτησης της ίδια πορείας και επιστρέφουν τα δεδομένα από την cache μειώνοντας τον μέσο όρο απόκρισης του συστήματος και κατ επέκταση την απόδοση και τον βαθμό κλιμάκωσης του.

### **Τεχνική διαχείρισης εξαρτήσεων**

Οι διαχειριστές εξαρτήσεων χρησιμοποιούνται σε πολλά σύγχρονα frameworks και γλώσσες προγραμματισμού και παρέχουν ένα κοινό πλαίσιο για την διαχείριση των εξαρτήσεων και των βιβλιοθηκών που απαιτούνται για την υλοποίηση μιας εφαρμογής ή την προσθήκη επιπλέον λειτουργικότητάς.

Η χρήση ενός διαχειριστή εξαρτήσεων στην ανάπτυξη μιας εφαρμογής προσθέτει πλεονεκτήματα επεκτασιμότητας της εφαρμογής καθώς γίνεται ευκολότερη η εγκατάσταση και ενσωμάτωση στο σύστημα νέων βιβλιοθηκών που παρέχουν επιπλέον λειτουργικότητα στην εφαρμογή. Επιπλέον ευνοεί την ανάπτυξη με την τεχνική του αρθρωτού προγραμματισμού όπως έχουμε αναφέρει στην αντίστοιχη ενότητα.

Κατά την ανάπτυξη της παρούσας εφαρμογής χρησιμοποιήθηκε το εργαλείο διαχείρισης εξαρτήσεων Composer. Ο Composer είναι ο κύριος διαχειριστής εξαρτήσεων για PHP και χρησιμοποιείται από την πλειοψηφία των PHP frameworks μεταξύ των οποίων και το CakePHP που χρησιμοποιούμε, το οποίο είναι και αυτό με την σειρά του σένα πακέτο του Composer.

### **Εργαλεία ανάπτυξης**

Η επιλογή των εργαλείων ανάπτυξης έγινε με κριτήριο το σύνολο των προδιαγραφών, των προτύπων και των τεχνικών ανάπτυξης που αναλύθηκαν στις προηγούμενες ενότητες.

Όπως φαίνεται και στην Εικόνα 3 η εφαρμογή αποτελείται από δυο βασικές συνιστώσες. Για την συνιστώσα της εφαρμογής που αφορά τον τελικό χρήστη και την απεικόνιση των δεδομένων στο πρόγραμμα περιήγησης (browser) επιλέχτηκε η γλώσσα δημιουργίας δυναμικών εφαρμογών διαδικτύου PHP και το framework CakePHP.

Για την υλοποίηση της υπηρεσίας συλλογής δεδομένων από τα κέντρα δεδομένων επιλέχτηκε η γλώσσα προγραμματισμού Python και το framework Django. Για την αποθήκευση των δεδομένων και την ανάκτηση τους από την διαδικτυακή εφαρμογή επιλέχτηκε το σύστημα διαχείρισης βάσης δεδομένων MySQL.

Παρακάτω παρουσιάζονται συνοπτικά τα κύρια χαρακτηριστικά του CakePHP και της Python ως κριτήρια επιλογής.

## CakePHP

Το CakePHP είναι ένα PHP framework ανοιχτού λογισμικού για την ταχυτάτη ανάπτυξη δυναμικών εφαρμογών διαδικτύου που ακολουθεί το MVC πρότυπο. Το CakePHP είναι δομημένο με τις παρακάτω τεχνικές ανάπτυξης:

- παρέχεται ως πακέτο στον διαχειριστή πακέτων Composer
- χρησιμοποιεί την τεχνική ORM κατά την σύνδεση του με την βάση δεδομένων
- παρέχει μηχανισμούς caching βάσης δεδομένων
- είναι λογισμικό ανοιχτού κώδικα

Πέραν των παραπάνω τα οποία έχουμε ήδη αναλύσει στην ενότητα «Πρότυπα & τεχνικές ανάπτυξης εφαρμογών» η επιλογή του συγκεκριμένου framework έγινε και για τους εξής λόγους:

1. Εύκολη εκμάθηση του. Η εύκολη εκμάθηση του συγκεκριμένου framework προσφέρει πλεονέκτημα ταχύτερης ένταξης στην διαδικασία ανάπτυξης για όσους θελήσουν να συνεισφέρουν μελλοντικά στην εξέλιξη του συστήματος.
2. Τεχνική σύμβασης αντί ρύθμισης  
Η τεχνική αυτή την οποία έχει υιοθετήσει το CakePHP αναφέρεται στον ορισμό ενός συνόλου συμβάσεων αντί παραμέτρων ρύθμισης όπως συνηθίζεται σε άλλα συστήματα για την ρύθμιση του.  
Το πλεονέκτημα αυτής της τεχνικής είναι ότι επιταχύνεται η ρύθμιση του συστήματος για όσους ακολουθούν το σύνολο των συμβάσεων.
3. Αυτόματη δημιουργία κώδικα  
Το CakePHP παρέχει ενσωματωμένο εργαλείο για την αυτόματη δημιουργία κώδικα το οποίο επιταχύνει δραστικά την διαδικασία εκκίνησης ανάπτυξης μιας εφαρμογής.
4. Ασφάλεια  
Στο CakePHP υπάρχουν ενσωματωμένα εργαλεία για την προστασία της εφαρμογής από γνωστές τεχνικές παραβίασης της ασφάλειας του συστήματος.

Βάσει των παραπάνω το CakePHP είναι ιδανικό εργαλείο για την ανάπτυξη της συνιστώσας της διαδικτυακής εφαρμογής για την απεικόνιση των δεδομένων στο πρόγραμμα περιήγησης του χρήστη.

## Python

Η Python είναι μια γλώσσα προγραμματισμού με δυνατότητες γρήγορης και αποδοτικής υλοποίησης ενός συστήματος. Τα βασικά πλεονεκτήματά της είναι:

- ευκολία στην εκμάθηση της
- υψηλού επιπέδου τύποι δεδομένων
- μικρής έκτασης και ευκολότερα στην συντήρηση προγράμματα
- ευανάγνωστος κώδικας
- πληθώρα βιβλιοθηκών και frameworks
- λογισμικό ανοιχτού κώδικα

Η επιλογή της Python έναντι της PHP για την υλοποίηση της συνιστώσας της εφαρμογής που αφορά την υπηρεσία συλλογής των δεδομένων έγκειται στην απόφαση να αναπτύξουμε την υπηρεσία αυτή στην μικρότερη δυνατόν έκταση ώστε να πετύχουμε ευκολότερη επέκταση και συντήρηση της αλλά ταυτόχρονα να μην χρεωθούμε με την επιβάρυνση επιπλέον βιβλιοθηκών πέραν των απαραίτητων για την διασύνδεση με την βάση δεδομένων.

Η PHP ως γλωσσά με έμφαση στην δημιουργία δυναμικών σελίδων δεν παρέχει εγγενώς και με την ίδια ευελιξία που παρέχουν άλλες γλώσσες προγραμματισμού το απαραίτητο πλαίσιο για ανάπτυξη μιας τέτοιας υπηρεσίας όπως η συλλογή δεδομένων από τα κέντρα δεδομένων.

Αντίθετα η Python προσφέρει το σύνολο των κλάσεων και των τύπων δεδομένων που απαιτούνται για την ανάπτυξη της ενώ ταυτόχρονα παρέχει και ORM τεχνικές μέσω του Django framework για την διασύνδεση με την βάση δεδομένων.

Επιπλέον η Python παρέχει συγκριτικά πλεονέκτημα και σε σχέση με άλλες γλώσσες προγραμματισμού που θα μπορούσαν να χρησιμοποιηθούν για την υλοποίηση της υπηρεσίας συλλογής δεδομένων όπως έμφαση σε κοινές προγραμματιστικές τεχνικές, μικρότερης έκτασης κώδικας, αντικειμενοστραφής λογική, κ.α. οριστικοποιώντας την απόφασή να χρησιμοποιηθεί για την συγκεκριμένη συνιστώσα της εφαρμογής [5].

## Google Maps API

Το Google Maps API είναι μια δωρεάν υπηρεσία χαρτογράφησης που παρέχει η Google για την δημιουργία διαδραστικών χαρτών για εφαρμογές διαδικτύου. Μέσω του API παρέχεται μια πολύ πλούσια συλλογή χαρακτηριστικών για την αναπαράσταση πληροφορίας σε γεωγραφικό χάρτη. Ενδεικτικά αναφέρουμε:

- απεικόνιση σημείου (marker) με συγκεκριμένες συντεταγμένες
- δυνατότητα χρήσης ξεχωριστού εικονιδίου για κάθε marker στο χάρτη
- δυνατότητα ανάθεσης επιπλέον αυθαίρετων χαρακτηριστικών στον κάθε marker

- άνοιγμα αναδυομένου παραθύρου (pop up) με πληροφορίες για κάθε marker
- εμφάνιση ίχνους πορείας (path) στον χάρτη
- zoom in, zoom out στον χάρτη
- πλοήγηση πάνω στο χάρτη
- απεικόνιση επιπέδων (layers) πληροφορίας

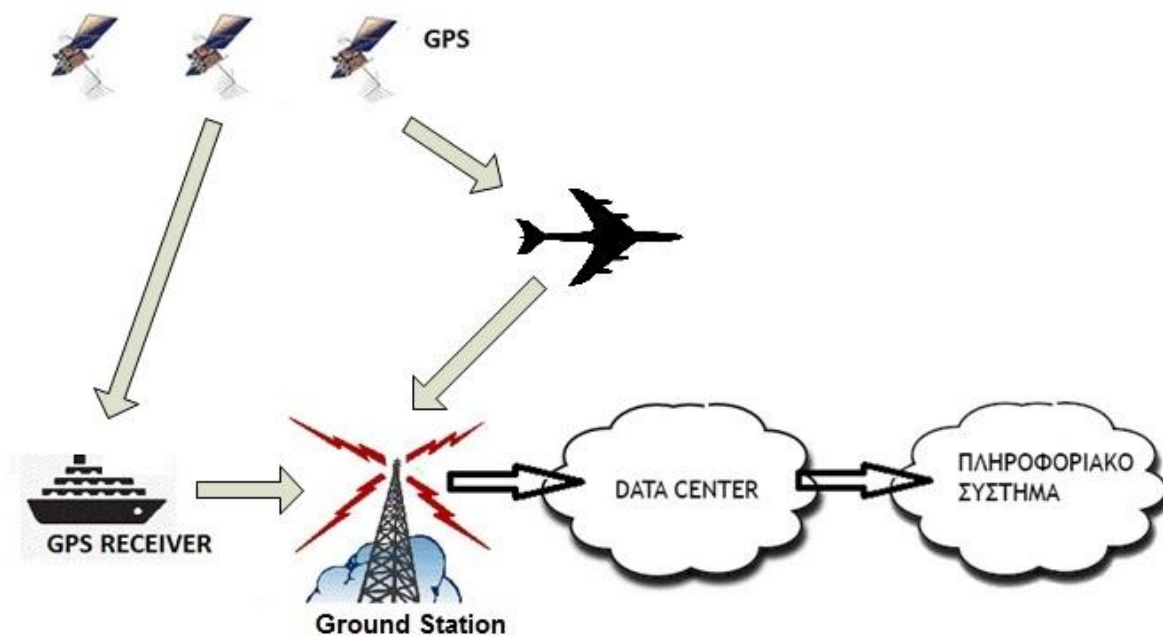
Η επιλογή του Google Maps API για την ανάπτυξη του διαδραστικού χάρτη απεικόνισης των δεδομένων, παρέχει την δυνατότητα χρήσης των παραπάνω χαρακτηριστικών, καθώς και την δυνατότητα χρήσης μελλοντικά επιπλέον χαρακτηριστικών του API για την αναπαράσταση πρόσθετης πληροφορίας.

# Κεφάλαιο 2

## Τα κέντρα δεδομένων

Στο κεφάλαιο αυτό θα παρουσιάσουμε λεπτομέρειες σχετικά με την δομή των κέντρων δεδομένων από τα οποία αντλούμε τα δεδομένα της εφαρμογής. Τα δεδομένα που διαχειριζόμαστε αρχικά για την ανάπτυξη της εφαρμογής προέρχονται από δύο κέντρα δεδομένων τα οποία αναμεταδίδουν δεδομένα που λαμβάνουν από δίκτυα αισθητήρων σχετικά με την πορεία πλοίων στο λιμάνι της Σούδας και του Ηρακλείου αντίστοιχα. Τα δύο κέντρα δεδομένων χρησιμοποιήθηκαν για την παροχή δεδομένων αλλά και για τις ανάγκες λειτουργικού ελέγχου της εφαρμογής. Στην τελική φάση ανάπτυξης της εφαρμογής έγινε σύνδεση της εφαρμογής και με κέντρα δεδομένων κίνησης αεροσκαφών ώστε να επιβεβαιωθεί η επεκτασιμότητα του πληροφοριακού συστήματος.

Το κάθε δίκτυο αισθητήρων αποτελείται από μια κεραία η οποία διαθέτει ενσωματωμένους αισθητήρες για συλλογή δεδομένων θέσης πλοίων καθώς και περιβαλλοντικών δεδομένων. Τα δεδομένα σχετικά με την θέση του κάθε πλοίου προέρχονται από το σύστημα AIS (Automatic Identification System). Το σύστημα AIS σχεδιάστηκε αρχικά για να βοηθήσει την αποφυγή συγκρούσεων πλοίων, καθώς και να υποστηρίξει τις λιμενικές αρχές στην επίτευξη του καλύτερου ελέγχου της θαλάσσιας κυκλοφορίας.



Εικόνα 7: Ροή δεδομένων προς το πληροφοριακό σύστημα.

Οι πομποί AIS που είναι εγκατεστημένοι στα πλοία περιλαμβάνουν έναν δέκτη εντοπισμού θέσης GPS που υπολογίζει τις συντεταγμένες της θέσης του πλοίου, την ταχύτητά του και την πορεία του και του αναμεταδίδει. Στη συνέχεια, ο αποδέκτης του σήματος μπορεί με

χρήση ειδικού λογισμικού που επεξεργάζεται τα δεδομένα να αποθηκεύσει ή να χρησιμοποιήσει την πληροφορία αυτή [6].

Μόλις λάβει τα δεδομένα το δίκτυο αισθητήρων τα αναμεταδίδει στο κέντρο δεδομένων κωδικοποιημένα. Στην συνέχεια το κέντρο δεδομένων τα αποκωδικοποιεί, τα φιλτράρει και τα αναμεταδίδει με την σειρά του προς όλους τους συνδεδεμένους με το κέντρο αποδέκτες. Η αντίστοιχη διαδικασία λαμβάνει χώρα και στην περίπτωση ανάλυσης κίνησης και άλλων μεσών. Στην Εικόνα 7 παρουσιάζεται η ροή δεδομένων από τα κινούμενα μέσα προς το πληροφοριακό σύστημα.

Η αναμετάδοση των δεδομένων από το κέντρο δεδομένων γίνεται μέσω δικτύου με την χρήση TCP ή UDP πρωτόκολλου. Ο αποδέκτης αφού συνδεθεί με χρήση των παραπάνω διαδικτυακών πρωτοκόλλων στο κέντρο δεδομένων μπορεί να διαβάσει την ροή δεδομένων που αναμεταδίδει το κέντρο και να την αποθηκεύσει ή να την επεξεργαστεί. Τα δεδομένα λαμβάνονται ως οριοθετημένο κείμενο όπου κάθε γραμμή περιέχει με προκαθορισμένη σειρά πληροφορίες σχετικά με το μέσο, την θέση του, την ώρα μετάδοσης του σήματος καθώς και άλλες μεταβλητές που ορίζουν το πρωτόκολλα μετάδοσης και περιγράφονται αναλυτικά στις αμέσως επόμενες ενότητες.

## Αναλυτική επεξήγηση του συστήματος AIS

Το σύστημα AIS είναι ένα σύστημα αυτόματης ανταλλαγής ψηφιακών σημάτων μεταξύ πλοίων, αλλά και παράκτιων συστημάτων κυκλοφορίας πλοίων, στη συχνότητα των υπερβραχέων κυμάτων (VHF). Μέσω του συστήματος αυτού επιτυγχάνεται η αμοιβαία ενημέρωση όλων των πλοίων, της ταυτότητάς τους, του φορτίου τους, του λιμένα απόπλου και κατάπλου, καθώς και άλλων χρήσιμων πληροφοριών. Το σύστημα AIS σχεδιάστηκε αρχικά για να βοηθήσει την αποφυγή συγκρούσεων πλοίων, καθώς και να υποστηρίξει τις λιμενικές αρχές στην επίτευξη του καλύτερου ελέγχου της θαλάσσιας κυκλοφορίας. Οι αναμεταδότες που βρίσκονται στα πλοία μπορούν ανάλογα την κατηγορία τους να μεταδώσουν :

- Στατικές παραμέτρους, δηλαδή:
  1. τη ναυτιλιακή κινητή δορυφορική ταυτότητα (MMSI)
  2. τον αριθμό αναγνώρισης IMO
  3. το όνομα του πλοίου (έως 20 χαρακτήρες) και το διακριτικό κλήσεως
  4. τις διαστάσεις του πλοίου στρογγυλοποιημένες σε ακέραιο αριθμό μέτρων
  5. ο τύπος του πλοίου (δεξαμενόπλοιο, κρουαζιερόπλοιο κλπ)
  6. η θέση επί του πλοίου, που αναφέρεται το στίγμα
  7. ο τύπος ηλεκτρονικής συσκευής προσδιορισμού στίγματος (απλό ή διαφορικό GPS)
- Δυναμικές παραμέτρους, δηλαδή:
  1. η θέση του πλοίου
  2. ο συγχρονισμένος παγκόσμιος χρόνος
  3. η αληθής πορεία από 0° έως 359°, όπως αυτή εισάγεται από τη γυροπυξίδα

4. η πορεία ως προς το βυθό
5. η ταχύτητα ως προς το βυθό
6. η ναυτιλιακή κατάσταση του πλοίου (εν πλω, αγκυροβολημένο, ακυβέρνητο κλπ)
7. ο ρυθμός στροφής, δεξιά (+) ή αριστερά (-)
8. ο ρυθμός ανανέωσης αναφοράς

• Παραμέτρους ταξιδιού, δηλαδή:

1. το βύθισμα του πλοίου
2. ο τύπος του φορτίου
3. ο προορισμός
4. ο εκτιμώμενος χρόνος κατάπλου (μήνας, ημέρα, ώρα και λεπτό σε συγχρονισμένο παγκόσμιο χρόνο)

Τα πλοία τα οποία ενσωματώνουν πομποδέκτη κατηγορίας Α εκπέμπουν όλες τις παραπάνω πληροφορίες ανά 2 -10 sec ενώ πλοία εφοδιασμένα με πομποδέκτη κατηγορίας Β εκπέμπουν ένα υποσύνολο των παραπάνω πληροφοριών και σε μεγαλύτερα χρονικά διαστήματα.

Οι αναμεταδότες των πλοίων αφού συγκεντρώσουν τις απαραίτητες πληροφορίες από συνεργαζόμενα υποσυστήματα (GPS) και λοιπούς αισθητήρες που υπάρχουν πάνω στο πλοίο τα αναμεταδίδουν κωδικοποιημένα, σε συγκεκριμένη συχνότητα και χρόνο.

Ένα τυπικό μήνυμα AIS έχει την παρακάτω μορφή

`!AIVDM,1,1,,A,14eG;o@034o8sd<L9i:a;WF>062D,0*7D`

όπου:

!AIVDM: Τύπος του μηνύματος

1: Αριθμός των προτάσεων.

1: Αριθμός της τρέχουσας πρότασης.

A: Το κανάλι που χρησιμοποιήθηκε για την εκπομπή.

14eG;... Οι κωδικοποιημένες πληροφορίες χρησιμοποιώντας AIS-ASCII16

0\* Τέλος δεδομένων (End of Data)

7D NMEA checksum (NMEA 0183 Standard CRC16)

Ένα δίκτυο αισθητήρων (δέκτες) φροντίζει για την λήψη των ανωτέρω σημάτων και την αναμετάδοση τους σε κωδικοποιημένη μορφή στα κέντρα δεδομένων. Ένα τυπικό πακέτο δεδομένων που προωθείται για αποκωδικοποίηση στους servers του συστήματος παρουσιάζεται στην παρακάτω Εικόνα 8.



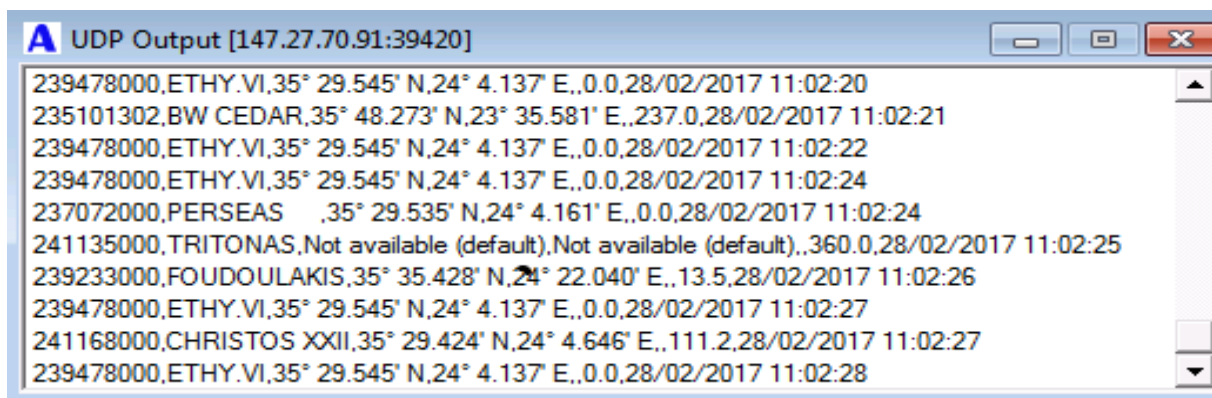
```

!AIVDM,1,1,,A,13aEOK?P00PD2wVMdLDRhgvL289?,0*26
!AIVDM,1,1,,B,16S`2cPP00a3UF6EKT@2: ?v0r0S2,0*00
!AIVDM,2,1,9,B,53nFBv01SJ<thHp6220H4heHTf222222222221?50:454o<`9QS1UDp,0*09
!AIVDM,2,2,9,B,8888888888888880,2*2E
!AIVDM,2,1,9,B,53nFBv01SJ<thHp6220H4heHTf222222222221?50:454o<`9QS1UDp,0*09
!AIVDM,2,1,9,B,53nFBv01SJ<thHp6220H4heHTf222222222221?50:454o<`9QS1UDp,0*09

```

**Εικόνα 8:** Τυπικό Πακέτο δεδομένων AIS

Ενώ στην Εικόνα 9 παρουσιάζονται τα δεδομένα εισόδου του πληροφοριακού συστήματος μετά την αποκωδικοποίηση.



**Εικόνα 9:** Δεδομένα Εισόδου μετά την Αποκωδικοποίηση

Το πληροφοριακό σύστημα θα μπορεί να υποστηρίξει και δεδομένα κίνησης άλλων μεταφορικών μεσών δεδομένου ότι η φιλοσοφία μετάδοσης χαρακτηριστικών κίνησης παραμένει η ίδια. Για παράδειγμα μπορούν άμεσα να ενσωματωθούν στο σύστημα κέντρα δεδομένων παροχής πληροφορίας θέσης αεροσκαφών ADS-B (Automatic Dependent Surveillance Broadcast).

## Αναλυτική επεξήγηση του συστήματος ADS-B

Το σύστημα ads-b (Automatic dependent surveillance - broadcast ) πρόκειται για ένα αυτόματο σύστημα μετάδοσης πληροφορίας θέσεως και επιμέρους πληροφοριών πτήσης από ένα αεροσκάφος προς τους σταθμούς εδάφους αλλά και προς τα άλλα αεροσκάφη που ίπτανται στην περιοχή. Το σύστημα ονομάζεται αυτόματο (Automatic) καθότι δεν απαιτείται η παρέμβαση κάποιου χρήστη ή χειριστή για την αποστολή των πληροφοριών και εξαρτώμενο ( dependent ) καθώς η εκπεμπόμενη πληροφορία εξαρτάται από τα ναυτιλιακά συστήματα του αεροσκάφους. Η εκπομπή των πληροφοριών από το αεροσκάφος γίνεται στην συχνότητα των 1090 MHz με συνεχόμενο ρυθμό. Οι πληροφορίες αυτές περιλαμβάνουν.

1. Αριθμός ICAO του αεροσκάφους.
2. Χαρακτηριστικό κλήσης (CALL SIGN).
3. Υψόμετρο πτήσης
4. Ταχύτητα σε σχέση με το έδαφος
5. Πληροφορία θέσης ( Γεωγραφικό πλάτος, Γεωγραφικό μήκος)
6. Πορεία

και μπορούν να ληφθούν εύκολα από ένα σταθμό βάσης. Ένα τυπικό μήνυμα Ads-b αποτελείται από μια 112 bit η ανάλυση των οποίων παρουσιάζεται στον Πίνακα 1.

Ads-b message : | DF 5 | \*\* 3 | ICAO 24 | DATA 56 | PI 24

nBits	Bits	Χαρακτηριστικό	Περιγραφή
5	1 - 5	DF	Downlink Format “10001” binary
3	6 - 8	CA	Capability (additional identifier)
24	9- 32	ICAO	ICAO aircraft address
56	33 - 88	DATA + TC	Data + Type Code
24	89 - 112	PI	Parity

Πίνακας 1: Δομή μηνύματος Ads-b

Κάθε μήνυμα ξεκινάει με την ακολουθία bit 10001 το οποίο ονομάζεται Downlink format. Στην συνέχεια ακολουθούν 3 bit επιπλέον χαρακτηριστικού CA που σχετίζονται με τον τύπο του μηνύματος. Ακολουθεί το χαρακτηριστικό νούμερο ICAO του αεροσκάφους και εν συνεχεία τα δεδομένα μαζί με τον τύπο των δεδομένων που αποστέλλονται. Το μήνυμα ολοκληρώνεται με πληροφορία 24bit checksum/parity για τον έλεγχο σφαλμάτων κατά την μετάδοση.

Οι τύποι της πληροφορίας που μεταδίδεται στο πεδίο Data καθορίζονται από την παράμετρο TC (Type Code) και μπορεί να είναι ανάλογα την τιμή του πεδίου όπως φαίνεται και στον Πίνακα 2.

TC	Content
1 - 4	Χαρακτηριστικό Αεροσκάφους - Aircraft identification
5 - 8	Πληροφορία θέσης σε σχέση με το έδαφος - Surface position
9 - 18	Πληροφορία θέσης Αέρος - Airborne position (w/ Baro Altitude)
19	Ταχύτητα Αεροσκάφους - Airborne velocities
20 - 22	Πληροφορία θέσης -Airborne position (w/ GNSS Height)
23 - 31	Δεσμευμένα για άλλες χρήσεις - Reserved for other uses

Πίνακας 2: Τύποι πληροφορίας περιεχομένου δεδομένων μηνύματος ads-b

Ένα παράδειγμα μηνύματος του συστήματος ads-b είναι :

8D4840D6202CC371C32CE0576098 (hex)

Παρατηρούμε ότι το μήνυμα ξεκινάει με την χαρακτηριστική ακολουθία του Downlink Format “10001” (Binary). Ο κωδικός TC ισούται με “00100” (Binary) δηλαδή με τέσσερα σε δεκαδικό σύστημα αρίθμησης άρα πρόκειται για μήνυμα αεροσκάφους και πεδίου δεδομένων δίνει “001011 001100 001101 110001 110000 110010 110011 100000” (Binary) που αντιστοιχεί στους χαρακτήρες ASCII : KLM1023\_.

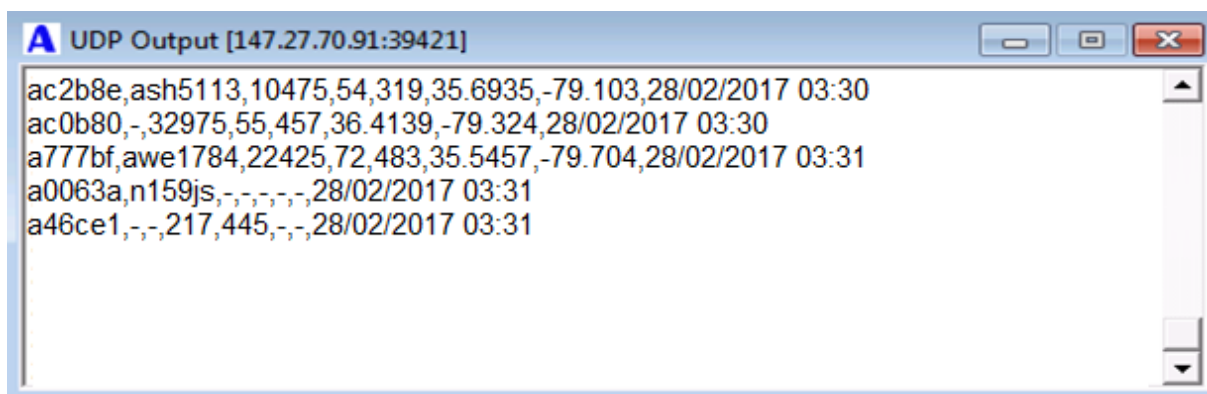
Ένα δίκτυο αισθητήρων (δέκτες) φροντίζει για την λήψη των ανωτέρω σημάτων και την αναμετάδοση τους σε κωδικοποιημένη μορφή στα κέντρα δεδομένων.

Ένα τυπικό πακέτο δεδομένων που προωθείται για αποκωδικοποίηση στους servers του συστήματος η έξοδος του οποίου έπειτα την αποκωδικοποίηση παρουσιάζεται στην παρακάτω Εικόνα 10.

ICAO	Flight	Alt (ft)	Heading	Knots	Lat	Lon
ac2b8e	ASH5113	10475	54	319	35.6935	-79.103
ac0b80	-	32975	55	457	36.4139	-79.324
a777bf	AWE1784	22425	72	483	35.5457	-79.704
a0063a	N159JS	-	-	-	-	-
a46ce1	-	-	217	445	-	-

**Εικόνα 10:** Αποκωδικοποίηση δεδομένων συστήματος Ads-b.

Ενώ στην Εικόνα 11 παρουσιάζονται τα δεδομένα εισόδου στο πληροφοριακό σύστημα μετά την αποκωδικοποίηση



**Εικόνα 11:** Δεδομένα εισόδου πληροφοριακού συστήματος

# Κεφάλαιο 3

## Η εφαρμογή

Σε αυτό το κεφάλαιο θα περιγράψουμε την δομή του πληροφοριακού συστήματος που υλοποιήσαμε, αναλύοντας τα δομικά του συστατικά και τις επιμέρους συνιστώσες του και παρουσιάζοντας προβλήματα που έπρεπε να επιλυθούν καθώς και τα δεδομένα που συλλέγονται σύμφωνα με αντίστοιχες μελέτες σε αντίστοιχα προβλήματα [7] [8]. Όπως έχουμε ήδη αναφέρει το σύστημα αποτελείται από δύο βασικές συνιστώσες: (Εικόνα 3)

- a) την υπηρεσία συλλογής δεδομένων σε πραγματικό χρόνο
- b) την εφαρμογή διαδικτύου

### Υπηρεσία συλλογής δεδομένων σε πραγματικό χρόνο

Όπως είδαμε στο κεφάλαιο 2 όπου έγινε περιγραφή της λειτουργίας των κέντρων δεδομένων για την λήψη δεδομένων απαιτείται η σύνδεση στο κέντρο για την παραλαβή των αποκωδικοποιημένων δεδομένων.

Η σύνδεση αυτή μπορεί να επιτευχθεί είτε TCP είτε UDP πρωτοκόλλου ώστε στην συνέχεια να γίνει η επεξεργασία της ροής δεδομένων ως οριοθετημένο κείμενο. Για την σύνδεση με το κέντρο δεδομένων επιλέχτηκε το TCP πρωτόκολλο ως πιο αξιόπιστο<sup>1</sup> για την επικοινωνία μεταξύ του κέντρου και της εφαρμογής δεδομένου και του όγκο της διακινούμενης πληροφορίας μεταξύ κέντρου και εφαρμογής.

Η ανάπτυξη αυτής της συνιστώσας της εφαρμογής έγινε χρησιμοποιώντας την γλώσσα Python. Δεδομένου του γεγονότος ότι η ροή δεδομένων είναι συνεχής και σε πραγματικό χρόνο η συνιστώσα της εφαρμογής για την συλλογή των δεδομένων έπρεπε να υλοποιηθεί ως υπηρεσία που εκτελείται διαρκώς στο παρασκήνιο του λειτουργικού συστήματος του εξυπηρετητή που φιλοξενεί την εφαρμογή. Τέτοιες υπηρεσίες παρασκήνιου στο λειτουργικό σύστημα Linux (το οποίο φιλοξενεί και την παρούσα εφαρμογή) ονομάζονται daemon services. Οι daemon services έχουν το πλεονέκτημα ότι μπορούν να «τρέξουν» χωρίς τον έλεγχο κάποιου χρήστη, στο παρασκήνιο, και να έχουν πρόσβαση στους απαιτούμενους πόρους για την ορθή λειτουργία τους. Επίσης ο χειρισμός τους μπορεί να γίνει χρησιμοποιώντας την ήδη υλοποιημένη υποδομή του λειτουργικού συστήματος. Έτσι ο χρήστης καθώς και το λειτουργικό σύστημα μπορεί ανά πάσα στιγμή να εκκινήσει, διακόψει, επανεκκινήσει την εφαρμογή ή να δει την κατάσταση της.

Για να πετύχουμε την αδιάκοπη λήψη των δεδομένων ορίσαμε την υπηρεσία ως υπηρεσία

---

<sup>1</sup> [http://www.diffen.com/difference/TCP\\_vs\\_UDP](http://www.diffen.com/difference/TCP_vs_UDP)

εκκίνησης (startup process) ώστε να εξασφαλίσουμε την εκκίνησή της από το λειτουργικό σύστημα σε περίπτωση επανεκκίνησης του τελευταίου. Για την ολοκλήρωση της μετατροπής της εφαρμογής σε υπηρεσία αλλά και της αυτόματης εκκίνησης της ένα αρχείο που περιγράφει τις διαδικασίες χειρισμού της υπηρεσίας τοποθετήθηκε μέσα στο φάκελο /etc/init.d του λειτουργικού συστήματος.

Αφού λύσαμε το ζήτημα της απρόσκοπτης εκτέλεσης της εφαρμογής ως διεργασία συστήματος καθώς και την αυτόματη εκκίνηση της, έπρεπε εν συνεχεία να χειριστούμε εξωτερικές αστοχίες όπως η περίπτωση απώλειας ή αδυναμίας δημιουργίας TCP σύνδεσης με το κέντρο δεδομένων. Τέτοιες αστοχίες μπορούν να συμβούν είτε λόγω προβλήματος δικτύου είτε λόγω μη λειτουργίας των δεκτών η του κέντρου αποκωδικοποίησης. Στις περιπτώσεις αυτές δεν επιτρέπεται στην υπηρεσία να τερματίσει την λειτουργία της αλλά αντ' αυτού να περιμένει ανενεργή για κάποιο χρονικό διάστημα και στην συνέχεια να επιχειρεί να συνδεθεί πάλι με το κέντρο. Η διαδικασία αυτή εκτελείται συνεχώς μέχρι την επιτυχή επανασύνδεση της υπηρεσίας με τα κέντρα δεδομένων. Ταυτόχρονα η υπηρεσία ενημερώνει τα αρχεία καταγραφής της (log) για τις διαδικασίες επιτρέποντας έτσι τόσο την αποσφαλμάτωση όσο και τον έλεγχο της ορθής λειτουργίας του συστήματος όπως φαίνεται και στην Εικόνα 12.

```
2017-03-19 20:43:40,340 - INFO - Connecting to STATION 1 (147.27.70.91:39422) socket.
2017-03-19 20:43:40,343 - INFO - Connecting to STATION 2 (147.27.70.103:39422) socket.
2017-03-19 20:43:40,348 - INFO - Connection established: STATION 1 (147.27.70.91:39422).
2017-03-19 20:43:40,350 - INFO - Closing connection to STATION 1 socket
2017-03-19 20:43:40,350 - INFO - Connection established: STATION 2 (147.27.70.103:39422).
2017-03-19 20:43:40,350 - INFO - Closing connection to STATION 2 socket
2017-03-19 20:43:50,526 - INFO - Connecting to STATION 1 (147.27.70.91:39422) socket.
2017-03-19 20:43:50,529 - INFO - Connecting to STATION 2 (147.27.70.103:39422) socket.
2017-03-19 20:43:50,535 - INFO - Connection established: STATION 1 (147.27.70.91:39422).
2017-03-19 20:43:50,535 - INFO - Connection established: STATION 2 (147.27.70.103:39422)
```

**Εικόνα 12:** Τμήμα log αρχείου της υπηρεσίας συλλογής δεδομένων

Μετά την επιτυχή υλοποίηση σύνδεσης με ένα κέντρο δεδομένων υλοποιήθηκε η δυνατότητα της υπηρεσίας να συνδέεται σε περισσότερα του ενός κέντρου δεδομένων. Όπως έχουμε ήδη αναφέρει στις προδιαγραφές που ορίσαμε για το πληροφοριακό σύστημα συμπεριλαμβάνεται και η δυνατότητα επέκτασης της και επομένως θα θέλαμε η υπηρεσία να μπορεί να λειτουργεί για την περίπτωση λήψης δεδομένων από N κέντρα δεδομένων. Βάσει αυτού η υπηρεσία υλοποιήθηκε ως πολυνηματική διεργασία αρχικοποιώντας ένα νήμα για κάθε κέντρο δεδομένων στο οποίο συνδεόμαστε. Για την υλοποίηση της πολυνηματικότητας χρησιμοποιήσαμε την βιβλιοθήκη “multiprocessing” της Python. Δόθηκε έμφαση ώστε οι απαραίτητες ρυθμίσεις για την σύνδεση σε ένα επιπλέον σταθμό να είναι οι ελάχιστες δυνατές καταλήγοντας στο παρακάτω απλό αρχείο ρυθμίσεων Config.ini το οποίο παρουσιάζεται στην Εικόνα 13.

```
[node1]
title: STATION 1
ip: 147.27.70.91
port: 39422
lat: 35.490193
lon: 24.0666989

[node2]
title: STATION 2
ip: 147.27.70.103
port: 39422
lat: 35.3439686
lon: 25.1468797
```

Εικόνα 13: Config.ini

Για κάθε κέντρο ορίζουμε ένα συμβολικό όνομα για τον σταθμό, την IP, την θύρα σύνδεσης και τέλος τις γεωγραφικές του συντεταγμένες. Η εφαρμογή κατά την εκκίνηση της (Κύριο Νήμα) δημιουργεί ένα καινούριο νήμα για κάθε σταθμό που υπάρχει στο αρχείο «Config.ini» το οποίο συνδέεται για την λήψη των δεδομένων από το αντίστοιχο κέντρο. Η εκτέλεση της εφαρμογής ως υπηρεσία η κατάσταση της καθώς και τα επιμέρους νήματα για την σύνδεση σε δύο κέντρα δεδομένων παρουσιάζεται στην παρακάτω Εικόνα 14 όπου φαίνονται τα αποτελέσματα εκτέλεσης έλεγχου κατάστασης της υπηρεσίας της εφαρμογής.

```
root@ais: ~
root@ais:~# service mari-feed status
● mari-feed.service
   Loaded: loaded (/etc/init.d/mari-feed; bad; vendor preset: enabled)
   Active: active (running) since Kup 2017-01-22 10:01:21 EET; 2 months 29 days ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 6
   Memory: 47.6M
      CPU: 16h 28min 52.275s
   CGroup: /system.slice/mari-feed.service
           └─1223 python /home/mari-feed/main.py
             └─1279 python /home/mari-feed/main.py
               └─1280 python /home/mari-feed/main.py

Warning: Journal has been rotated since unit was started. Log output is incomplete or unavailable.
root@ais:~# █
```

Εικόνα 14: Ενημέρωση κατάστασης εκτέλεσης της εφαρμογής

Έχοντας εξασφαλίσει διαρκή σύνδεση με τα κέντρα δεδομένων υλοποιήσαμε την επεξεργασία και την αποθήκευση των δεδομένων που λαμβάνουμε. Η επεξεργασία των δεδομένων γίνεται διαβάζοντας γραμμή-γραμμή την ροή δεδομένων και διαχωρίζοντας το οριοθετημένο κείμενο στα αντίστοιχα πεδία. Επιπλέον δημιουργούμε και νέα πεδία με μορφή κατάλληλη για επεξεργασία όπως για παράδειγμα η μετατροπή της ημερομηνίας σε αριθμό timestamp για ευκολότερη ταξινόμηση και μετατροπή σε άλλη μορφή κατά την παρουσίαση των δεδομένων στην web εφαρμογή. Η εφαρμογή φροντίζει για τον έλεγχο της ορθότητας της πληροφορίας για την απαλοιφή επιπλέον χαρακτήρων καθώς και για την απόρριψη εσφαλμένων μηνυμάτων. Τέτοια μηνύματα δύναται να προκύψουν από



εσφαλμένες λήψης μηνυμάτων στου δέκτες τα όποια οδηγούν σε λάθος αποκωδικοποιημένα μηνύματα. Η εφαρμογή φροντίζει για την απόρριψη τέτοιου είδους μηνυμάτων.

Τα έγκυρα πλέον πεδία αποθηκεύονται σε πίνακα της βάσης δεδομένων της εφαρμογής με την τεχνική ORM. Όπως έχουμε αναφέρει στην αντίστοιχη ενότητα (δες Τεχνική ORM) στόχος είναι να μην εμπλακούμε με τις λεπτομέρειες σχετικά με τις λειτουργίες εγγραφής και ανάγνωσης για το συγκεκριμένο σύστημα διαχείρισης βάσεων δεδομένων. Αντίθετα θέλουμε να αντιμετωπίζουμε τις εγγραφές ως αντικείμενα της γλώσσας προγραμματισμού που χρησιμοποιούμε τα οποία μπορούμε να αποθηκεύσουμε και να ανακτήσουμε από κάποια βάση δεδομένων. Η λογική αυτή θα φανεί ιδιαίτερα χρήσιμη σε μελλοντικές επεκτάσεις της εφαρμογής και σε ενδεχόμενη αλλαγή της αρχιτεκτονικής του συστήματος διαχείρισης βάσεων δεδομένων. Η αναπαράσταση μιας τέτοιας οντότητας ORM για την αποθήκευση δεδομένων της κίνησης των πλοίων καθώς και η αντιστοίχιση της σε βάση δεδομένων Mysql που χρησιμοποιούμε φαίνονται στην Εικόνα 15 και Εικόνα 16 αντίστοιχα.

```
class Vessel(models.Model):
    id = models.AutoField(primary_key=True)
    mmsi = models.IntegerField()
    vessel_name = models.CharField(max_length=255)
    mid = models.CharField(max_length=64)
    latitude = models.CharField(max_length=32)
    longitude = models.CharField(max_length=32)
    sog = models.FloatField()
    gog = models.FloatField()
    received_time = models.CharField(max_length=32)
    received_timestamp = models.IntegerField()
    lat = models.CharField(max_length=16)
    lon = models.CharField(max_length=16)
    station = models.CharField(max_length=16)
    class Meta:
        db_table = "vessels"
```

Εικόνα 15: Αναπαράσταση Μοντέλου για την Αποθήκευση Δεδομένων Κίνησης Πλοίων

#	Name	Datatype	Length/Set	Unsigned	Allow N...	Zerofill	Default	Comment	Collation
1	id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT		
2	latitude	CHAR	32	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		ascii_general_ci
3	longitude	CHAR	32	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		ascii_general_ci
4	sog	FLOAT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
5	gog	FLOAT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
6	received_time	CHAR	24	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		ascii_general_ci
7	received_times...	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
8	lat	CHAR	16	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		ascii_general_ci
9	lon	CHAR	16	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		ascii_general_ci
10	mmsi_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
11	station	CHAR	16	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		ascii_general_ci

Εικόνα 16: Αποτύπωση του Μοντέλου Κίνησης στην Βάση Δεδομένων

Ένα από τα προβλήματα που αντιμετωπίσαμε μετά την υλοποίηση του παραπάνω σχεδιασμού ήταν ότι λόγω των πολλαπλών και συνεχών δεδομένων ο πίνακας αποθήκευσης των θέσεων των πλοίων γινόταν αρκετά μεγάλος με αποτέλεσμα η γρήγορη αναζήτηση της τελευταίας θέσης ενός πλοίου να γίνεται ιδιαίτερα δαπανηρή. Αντιμετωπίσαμε αυτό το πρόβλημα δημιουργώντας ένα νέο πίνακα στην βάση δεδομένων και κρατώντας στον πίνακα αυτό μόνο την τελευταία θέση του κάθε πλοίου. Η εφαρμογή

φροντίζει στην συνεχή ενημέρωση του πίνακα αυτού με την τελευταία θέση του κάθε πλοίου καθώς και την απαλοιφή πλοίων που δεν έχουν αναφέρει θέση για ένα συγκεκριμένο χρονικό διάστημα. Συνεπώς είναι ιδιαίτερα γρήγορη η εμφάνιση της παρούσας κατάστασης ανά δεδομένη χρονική στιγμή αναζητώντας την πληροφορία μόνο στον πίνακα αυτόν του οποίου το μέγεθος είναι σχετικά μικρό και η ανάκτηση της πληροφορίας πολύ γρήγορη. Το μοντέλο για τον νέο πίνακα καθώς και η αποτύπωση του σε βάση δεδομένων είναι ακριβώς το ίδιο με αυτό που παρουσιάστηκε στην εικόνα και εικόνα αντίστοιχα απλώς διαφέρουν το όνομα του πίνακα καθώς και τα αποθηκευμένα δεδομένα.

Κλείνοντας την περιγραφή της τρέχουσας συνιστώσας της εφαρμογής υπενθυμίζουμε ότι η υλοποίηση έγινε με χρήση της γλώσσας προγραμματισμού Python στην οποία βρήκαμε δομές και κλάσεις υψηλού επιπέδου για την υλοποίηση της σύνδεσης μέσω TCP στα κέντρα δεδομένων, την πολυνηματική επεξεργασία, τον χειρισμό οριοθετημένου κειμένου και την αποθήκευση μέσω τεχνικής ORM κρατώντας την έκταση της υπηρεσίας στο ελάχιστο δυνατόν.

## Web Crawler

Στην προηγούμενη ενότητα περιγράφηκε ο τρόπος λήψης διαχείρισης και αποθήκευσης των δεδομένων που αποστέλλουν τα κινούμενα αντικείμενα. Επιπλέον όμως αυτών των δεδομένων που μεταφέρονται στην εφαρμογή απευθείας από το κέντρο δεδομένων ήταν επιθυμητός ο εμπλουτισμός τους με χρήσιμα μεταδεδομένα τα οποία απαιτούνται για την διενέργεια ερευνών για τις περιβαλλοντολογικές επιπτώσεις από την κίνηση των μέσων. Για παράδειγμα για την περίπτωση των πλοίων δεδομένα όπως ο τύπος του πλοίου (επιβατηγό πλοίο, φορτηγό κτλ), η σημαία του, το βάρος του, η ισχύς των κύριων μηχανών, είτε δεν παρέχονται επειδή δεν υποστηρίζονται από το πρότυπο τους AIS είτε τα πλοία δεν υποχρεώνονται να εκπέμπουν τις πληροφορίες αυτές.

Συνεπώς για την συλλογή των μεταδεδομένων αυτών αποφασίστηκε η υλοποίηση μιας βοηθητικής συνιστώσας της εφαρμογής η οποία συλλέγει τα απαιτούμενα δεδομένα από άλλες ιστοσελίδες που είναι διαθέσιμα ελεύθερα με αυτόματο τρόπο χωρίς να είναι απαραίτητη η παρέμβαση του χρήστη. Μια τέτοια υπηρεσία η οποία συλλέγει δεδομένα από άλλες ιστοσελίδες ονομάζεται web crawler.

Στην παρούσα εφαρμογή ο web crawler σαρώνει την ιστοσελίδα [marinetraffic.gr](http://marinetraffic.gr) η οποία εκτός από πληροφορίες θέσης των πλοίων παρέχει και χαρακτηριστικά των πλοίων σε παγκόσμια κλίμακα. Χρησιμοποιώντας σαν κοινό μοναδικό χαρακτηριστικό για κάθε πλοίο το αριθμό mmsi μπορούμε να εξάγουμε τα δεδομένα από την σελίδα αυτή και να τα αποθηκεύσουμε στην βάση. Το βασικό πρόβλημα που συναντήσαμε ήταν ότι οι πληροφορίες αυτές δεν είναι ορατές πριν την είσοδο χρήστη. Οι κλασσικοί web crawlers δεν εκτελούν ενέργειες χρήστη όπως σύνδεση σε λογαριασμό μέσω κωδικών πρόσβασης και απλά σαρώνουν το δημόσια ορατό περιεχόμενο της σελίδας ακολουθώντας τους



συνδέσμους που εντοπίζουν για να σαρώσουν τις υπόλοιπες σελίδες του ιστότοπου.

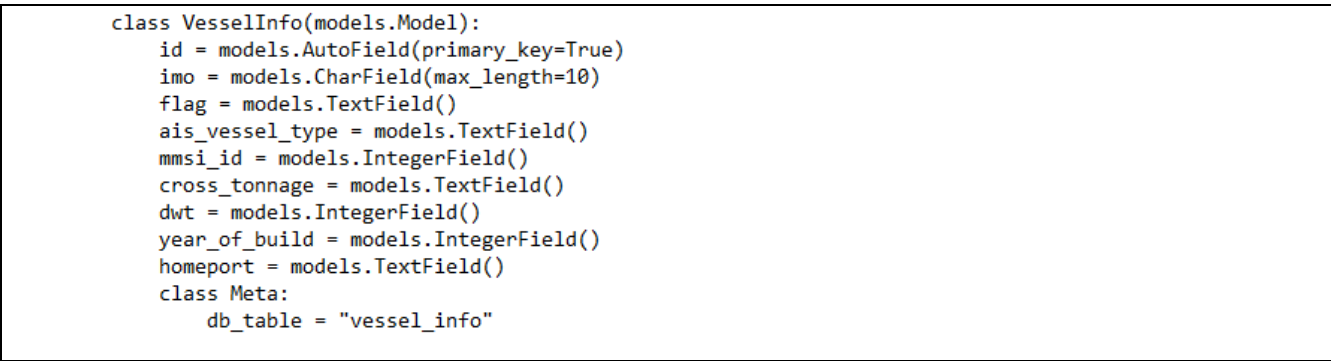
Αντιμετωπίσαμε αυτό το πρόβλημα χρησιμοποιώντας εργαλεία τα οποία έχουν την δυνατότητα εκτέλεσης ενεργειών χρήστη στην σελίδα που σαρώνουν. Πιο συγκεκριμένα χρησιμοποιήθηκε ο headless browser<sup>2</sup> phantomjs ο οποίος παρέχει την δυνατότητα εκτέλεσης ενεργειών χρήστη μέσω javascript<sup>3</sup>. Μέσω του phantomjs μπορέσαμε να συνδεθούμε στον λογαριασμό μας στην συγκεκριμένη ιστοσελίδα και να ανακτήσουμε τα μεταδεδομένα μέσω σάρωσης. Κάποια από τα πεδία που σαρώνει η εφαρμογή από την ιστοσελίδα που αναφέραμε παρουσιάζονται στην Εικόνα 17. Η εξαγωγή των δεδομένων βασίζεται στην προκαθορισμένη δομή της συγκεκριμένης ιστοσελίδας και στην ανάγνωση των δεδομένων που υπάρχουν σε συγκεκριμένα html attributes με συγκεκριμένη ονοματοδοσία.



```
> IMO: 9178599
> Name: ELYROS
> MMSI: 240685000
> Vessel Type: RO-RO/PASSENGER SHIP
> Gross Tonnage: 33635
> Summer DWT: 5590 t
> Build: 1998
> Flag: GREECE
> Home port: CHANIA
```

Εικόνα 17: Δείγμα πεδίων δεδομένων σάρωσης ιστοσελίδας

Ο crawler που υλοποιήσαμε παρακολουθεί περιοδικά την βάση δεδομένων για την εύρεση νέων πλοίων χωρίς μεταδεδομένα και στην συνεχεία σαρώνει τις σελίδες που ορίζουμε ως πηγή προέλευσης των μεταδεδομένων για την ανάκτηση τους. Η διαδικασία γίνεται αυτόματα και χωρίς την παρέμβαση του χρήστη. Πάλι ακλουθώντας πιστά την τεχνική ORM δημιουργήσαμε ένα μοντέλο για την αποθήκευση των μεταδεδομένων που παρουσιάζεται στην Εικόνα 18 ενώ η αποτύπωση του μοντέλου σε πίνακα της βάσης δεδομένων mysql που χρησιμοποιούμε παρουσιάζεται στην Εικόνα 18.



```
class VesselInfo(models.Model):
    id = models.AutoField(primary_key=True)
    imo = models.CharField(max_length=10)
    flag = models.TextField()
    ais_vessel_type = models.TextField()
    mmsi_id = models.IntegerField()
    cross_tonnage = models.TextField()
    dwt = models.IntegerField()
    year_of_build = models.IntegerField()
    homeport = models.TextField()
    class Meta:
        db_table = "vessel_info"
```

<sup>2</sup> [https://en.wikipedia.org/wiki/Headless\\_browser](https://en.wikipedia.org/wiki/Headless_browser)

<sup>3</sup> <https://en.wikipedia.org/wiki/JavaScript>

**Εικόνα 18:** Αναπαράσταση Μοντέλου για την Αποθήκευση Μεταδεδομένων

Η εκτέλεση του Crawler γίνεται περιοδικά ώστε να μην επιβαρύνεται η σελίδα στόχος και μόνο για τα κινούμενα μέσα τα οποία δεν διαθέτουν μεταδεδομένα. Σε κάθε ενέργεια ενημερώνεται το αρχείο καταγραφής της εφαρμογής. Ο πυρήνας του Crawler γράφτηκε σε γλώσσα Python εκτός του headless browser ο οποίος είναι γραμμένος σε javascript.

#	Name	Datatype	Length/Set	Unsigned	Allow N...	Zerofill	Default	Comment	Collation
1	id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT		
2	mmsi_id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
3	imo	CHAR	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		ascii_general_
4	flag	TEXT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	No default		utf8_general_
5	ais_vessel_type	TEXT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	No default		utf8_general_
6	dwt	INT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	No default		
7	year_of_build	INT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	No default		
8	homeport	INT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	No default		
9	cross_tonnage	INT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	No default		

**Εικόνα 19:** Αποτύπωση του Μοντέλου Μεταδεδομένων στην Βάση Δεδομένων

## Συλλογή περιβαλλοντικών δεδομένων

Στην προηγούμενες ενότητες έγινε περιγραφή του τρόπου λήψης διαχείρισης και αποθήκευσης των δεδομένων που αποστέλλουν τα κινούμενα αντικείμενα καθώς και ο τρόπος συλλογής διαχείρισης και αποθήκευσης των μεταδεδομένων αυτών. Για την μελέτη των περιβαλλοντολογικών επιπτώσεων από την κίνηση των μέσων που διαχειρίζεται η εφαρμογή είναι απαραίτητα στου ερευνητές και κάποια περιβαλλοντολογικά δεδομένα. Μια συνιστώσα της εφαρμογής φροντίζει για την συλλογή των δεδομένων αυτών με αυτόματο τρόπο έτσι ώστε να συνοδεύουν τα υπόλοιπα δεδομένα και μεταδεδομένα κίνησης και να διευκολύνουν τα χρήστη.

Η συλλογή των δεδομένων γίνεται από δυο διαφορετικές πηγές αλλά με τον ίδιο τρόπο. Η μία πηγή που χρησιμοποιείται είναι από αισθητήρες περιβαλλοντικών δεδομένων που είναι εγκαταστημένοι πλησίον των κεραιών συλλογής δεδομένων και η δεύτερη πηγή είναι χρησιμοποιώντας μια υπηρεσία περιβαλλοντικών δεδομένων όπως αυτή του <https://openweathermap.org>.

Η εφαρμογή είναι γραμμένη στην γλώσσα προγραμματισμού python και εκτελείται ανά τακτά χρονικά διαστήματα, συλλέγοντας και αποθηκεύοντας τα επιθυμητά περιβαλλοντολογικά δεδομένα. Για κάθε σταθμό που έχει εισαχθεί στο σύστημα του αρχείο config.ini που περιγράφηκε σε προηγούμενη ενότητα η εφαρμογή στέλνει ένα ερώτημα προς το API της εφαρμογής που περιέχει της γεωγραφικές συντεταγμένες του σημείου για το οποίο τα περιβαλλοντικά δεδομένα μας ενδιαφέρουν. Το ερώτημα είναι της μορφής <http://samples.openweathermap.org/data/2.5/weather?lat=X&lon=Y> όπου X το γεωγραφικό μήκος και Y το γεωγραφικό πλάτος του σημείου ενδιαφέροντος. Η απάντηση στο παραπάνω ερώτημα είναι ένα αντικείμενο JSON όπως αυτό παρουσιάζεται στην Εικόνα 20.

```
{
  "coord": {
    "lon": 139.01,
    "lat": 35.02
  },
  "weather": [
    {
      "id": 800,
      "main": "Clear",
      "description": "clear sky",
      "icon": "01n"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 285.514,
    "pressure": 1013.75,
    "humidity": 100,
    "temp_min": 285.514,
    "temp_max": 285.514
  },
  "sea_level": 1023.22,
  "grnd_level": 1013.75,
  "wind": {
    "speed": 5.52,
    "deg": 311
  },
  "clouds": {
    "all": 0
  },
  "dt": 1485792967,
  "sys": {
    "message": 0.0025,
    "country": "JP",
    "sunrise": 1485726240,
    "sunset": 1485763863,
    "id": 1907296,
    "name": "Tawarano",
    "cod": 200
  }
}
```

**Εικόνα 20:** Απάντηση Openweather σε JSON μορφή

Στην περίπτωση που για την άντληση των δεδομένων χρησιμοποιούνται αισθητήρες που βρίσκονται στον χώρο των κεραιών η απάντηση που λαμβάνει η εφαρμογή παρουσιάζεται στην Εικόνα 21.

```
{
  "coord": {
    "lon": 34.12,
    "lat": 35.02
  },
  "weather": [
    {
      "main": {
        "temp": 285.514,
        "humidity": 98
      },
      "wind": {
        "speed": 5.52,
        "deg": 391
      }
    }
  ]
}
```

**Εικόνα 21:** Απάντηση κέντρου δεδομένων σε JSON μορφή

Η εφαρμογή αφού λάβει τα παραπάνω δεδομένα φροντίζει για την εξαγωγή των επιθυμητών δεδομένων και την αποθήκευση τους. Πάλι ακολουθώντας πιστά την τεχνική ORM δημιουργήσαμε ένα μοντέλο για την αποθήκευση των περιβαλλοντολογικών δεδομένων που παρουσιάζεται στην Εικόνα 22 ενώ η αποτύπωση του μοντέλου σε πίνακα της βάσης δεδομένων mysql που χρησιμοποιούμε παρουσιάζεται στην Εικόνα 23.

```
class Clima(models.Model):
    id = models.AutoField(primary_key=True)
    lat = models.CharField(max_length=16)
    lon = models.CharField(max_length=16)
    received_timestamp = models.IntegerField()
    temperature = models.FloatField()
    wind = models.FloatField()
    wind_deg = models.IntegerField()
    humidity = models.FloatField()
    station = models.CharField(max_length=16)
    station_name = models.CharField(max_length=255)
    class Meta:
        db_table = "climatic_info"
```

**Εικόνα 22:** Αναπαράσταση Μοντέλου για την Αποθήκευση Περιβαλλοντικών Δεδομένων.

#	Name	Datatype	Length/Set	Unsigned	Allow N...	Zerofill	Default	Comment	Collation
1	id	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT		
2	lat	CHAR	16	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		ascii_general_ci
3	lon	CHAR	16	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		ascii_general_ci
4	received_times...	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
5	temperature	FLOAT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
6	wind	FLOAT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
7	humidity	FLOAT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
8	wind_deg	INT	4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
9	station	CHAR	16	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		ascii_general_ci
10	station_name	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		

## Εφαρμογή WEB για την απεικόνιση σε πραγματικό χρόνο των περιβαλλοντικών δεδομένων

Στις προηγούμενες ενότητες παρουσιάστηκε το back-end του πληροφοριακού συστήματος εστιάζοντας στον τρόπο συλλογής διαχείρισης και αποθήκευσης των δεδομένων καθώς και στις πηγές από τις οποίες προκύπτουν τα δεδομένα αυτά. Στην παρούσα ενότητα θα παρουσιαστεί το front-end κομμάτι του πληροφοριακού συστήματος δηλαδή η διεπαφή με τον χρήστη και το πώς μπορούν να παρουσιαστούν και να χρησιμοποιηθούν τα δεδομένα τα οποία η εφαρμογή έχει ήδη συλλέξει.

Για την απεικόνιση των δεδομένων που συλλέγονται από τους σταθμούς υλοποιήθηκε μια web εφαρμογή η οποία συνδέεται και ανακτά τα δεδομένα από την βάση στην οποία η υπηρεσία συλλογής δεδομένων τα αποθηκεύει και κατόπιν τα παρουσιάζει σε γεωγραφικό χάρτη.

Καθώς τα δεδομένα όπως αναφέραμε στην προηγούμενη ενότητα αποθηκεύονται σε πραγματικό χρόνο η απεικόνισή τους είναι και αυτή σε πραγματικό χρόνο. Για την υλοποίηση της web εφαρμογής επιλέχθηκε το PHP framework CakePHP για το σύνολο των προδιαγραφών που προσφέρει όπως αναλύθηκαν σε προηγούμενη ενότητας (δες CakePHP).

Η δομή της εφαρμογής ακολουθεί το πρότυπο MVC οι οποίες στην παρούσα περίπτωση είναι:

### 1. Model

Αποτελείται κυρίως από το μοντέλο Vessel και VesselHistory τα οποία είναι υπεύθυνα για την ανάκτηση και την προετοιμασία των δεδομένων για απεικόνιση των πληροφοριών για την θέση ενός πλοίου και του ίχνους του. Το μοντέλο Vessel όπως έχουμε ήδη αναφέρει αποθηκεύει την τελευταία θέση των κινούμενων μέσων, περιέχοντας μόνο τα μέσα που έχουν πρόσφατα δηλώσει την θέση τους. Το μοντέλο VesselHistory περιέχει όλη εκείνη την πληροφορία κίνησης η οποία έχει συλλεχθεί από την εφαρμογή.

### 2. Controller

Παρέχει μεθόδους για τον χειρισμό των ενεργειών (actions) του χρήστη.

Για παράδειγμα όταν ο χρήστης ζητήσει το ίχνος ενός πλοίου με τότε η εφαρμογή μέσω κατάλληλης κλήσης καλεί την μέθοδο getTrack(mmsi) του VesselHistory controller η οποία με την σειρά της ζητάει από το VesselHistory model τα δεδομένα που απαιτούνται για την απεικόνιση του ίχνους του πλοίου.

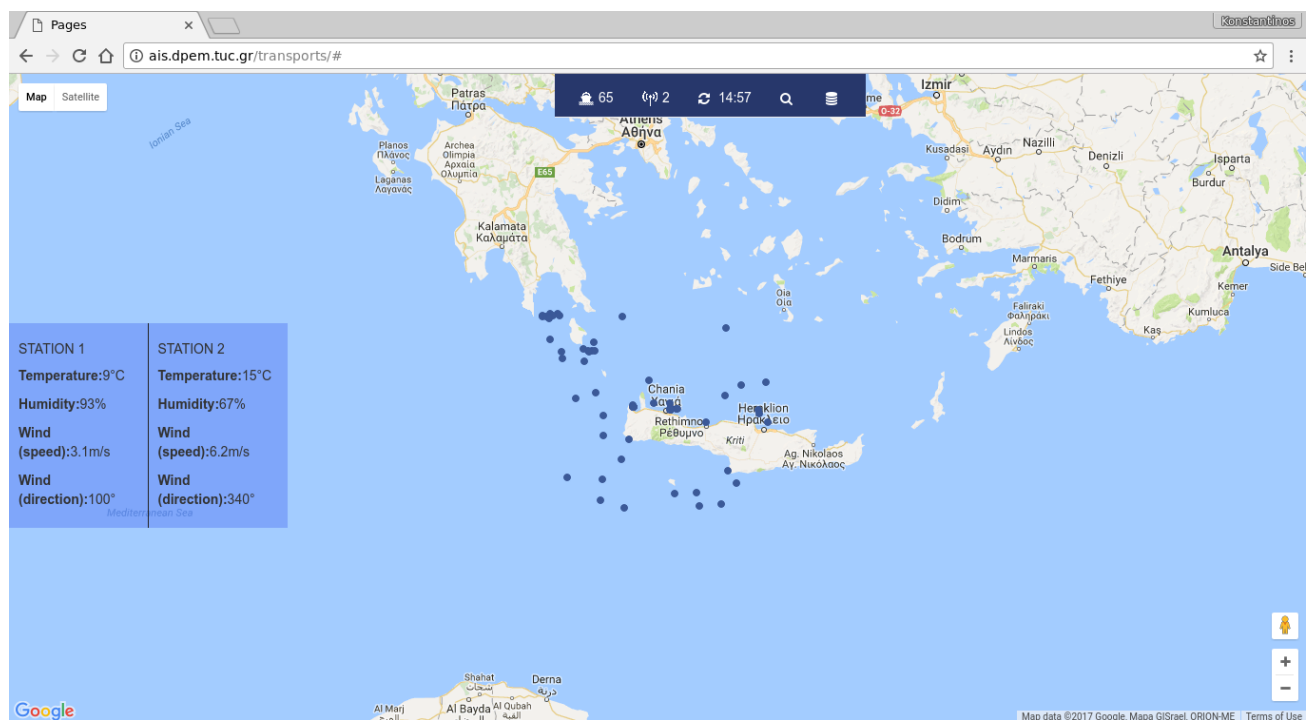
Πρέπει να σημειωθεί ότι παρόλο ότι το όνομα του controller είναι το ίδιο με το όνομα του model από τα οποία ζητάει τα δεδομένα αυτό είναι καθαρά θέμα σύμβασης και δεν υπάρχει κανένας περιορισμός στο όνομα του controller και στο σύνολο των models που μπορεί να χρησιμοποιήσει.

### 3. View

Απεικονίζει τα δεδομένα σε διαδραστικό γεωγραφικό χάρτη τον οποίο φτιάχνουμε με την υπηρεσία Google Maps.

Επιπλέον ενσωματώνει στην σελίδα του browser όλα τα πλαίσια ελέγχου που αντιστοιχούν στις ενέργειες που θέλουμε να παρέχουμε στον χρήστη. Και τέλος παρουσιάζει χρήσιμες πληροφορίες στον χρήστη.

Στην παρακάτω Εικόνα 24 φαίνεται η απεικόνιση στον χάρτη των θέσεων όλων των πλοίων, τα πλαίσια ελέγχου καθώς και χρήσιμες πληροφορίες όπως πόσα πλοία απεικονίζονται την τρέχουσα στιγμή και από πόσες κεραίες γίνεται λήψη των δεδομένων.



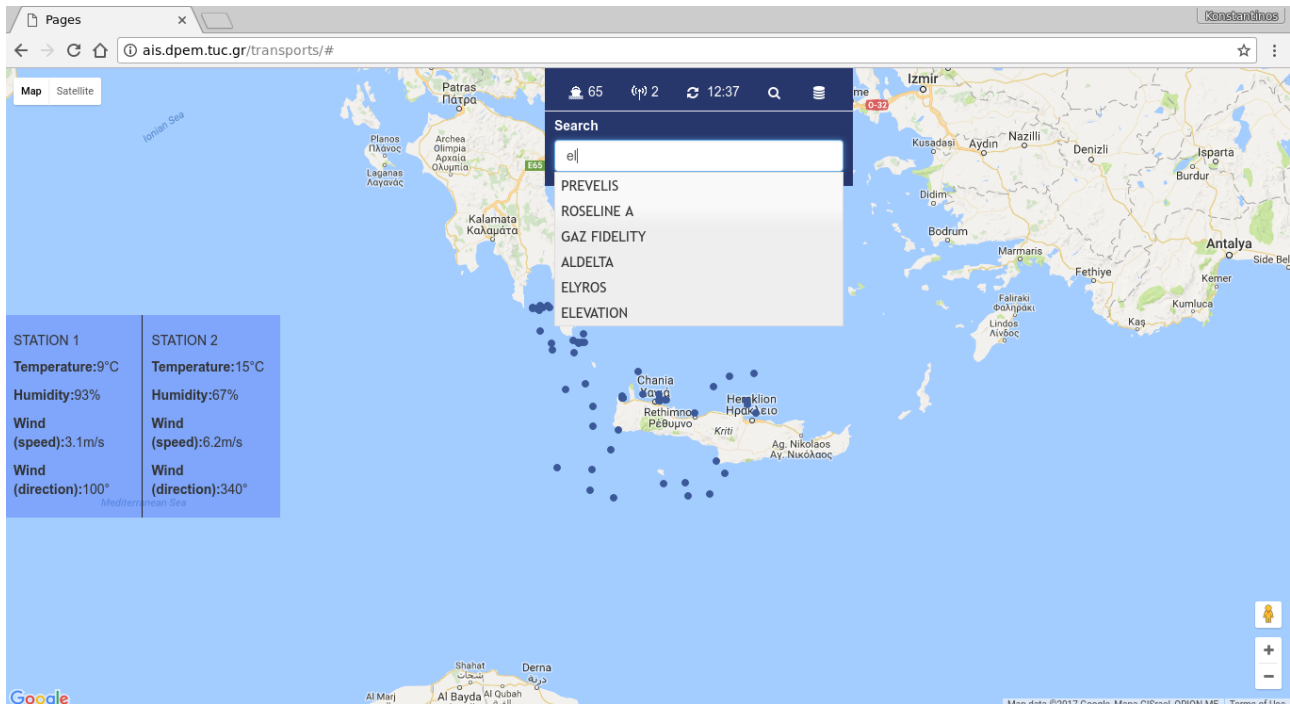
Εικόνα 24: Γενική Εικόνα Εφαρμογής

Όπως φαίνεται και στην παραπάνω εικόνα η σελίδα που βλέπει ο χρήστης χωρίζεται σε δυο κύρια μέρη:

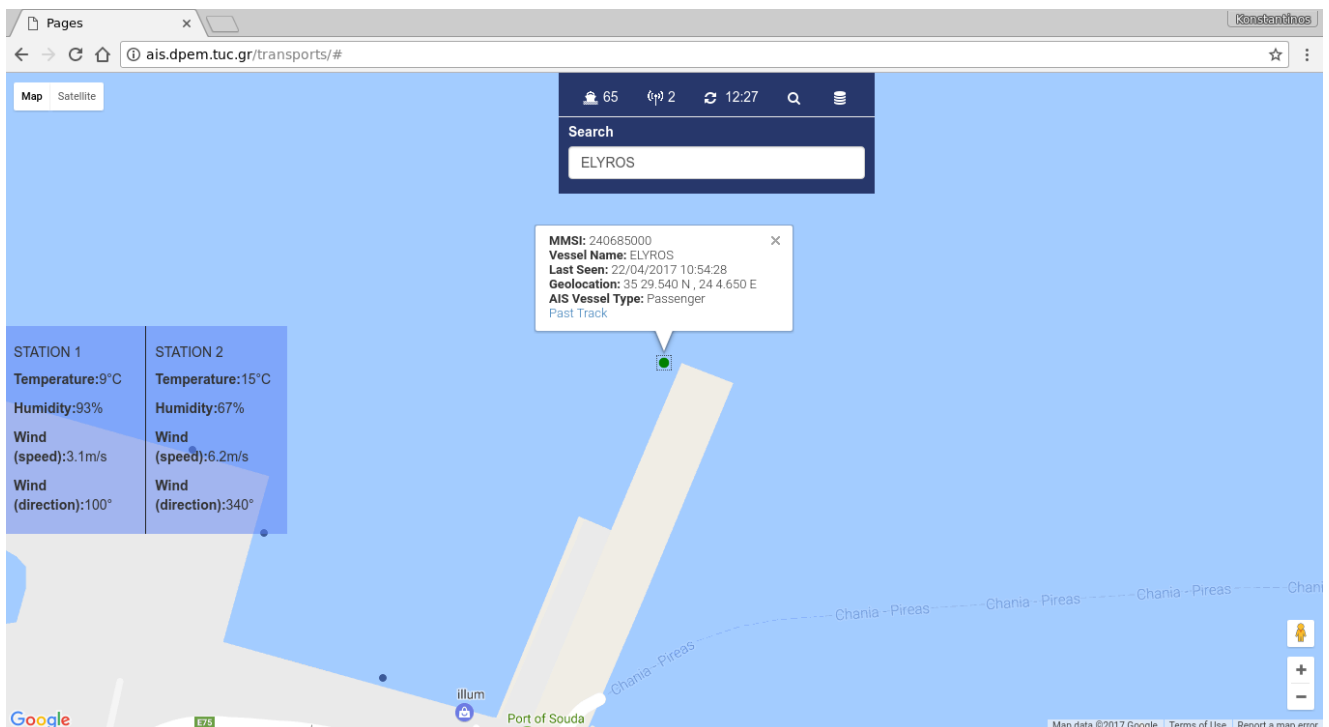
1. Το κεντρικό πάνελ ενεργειών και πληροφοριών που βρίσκεται στο κέντρο του πάνω μέρους της σελίδας
2. Τον διαδραστικό χάρτη που καταλαμβάνει όλη την οθόνη

Μέσω του κεντρικού πάνελ παρέχεται στον χρήστη ένα σύνολο ενεργειών καθώς και πληροφοριών για την αποτελεσματικότερη και ταχύτερη χρήση της εφαρμογής. Στα αριστερά εμφανίζεται ο αριθμός των κινούμενων μέσων που έχουν δηλώσει πρόσφατα την θέση τους και άρα εμφανίζονται στον διαδραστικό χάρτη. Δεξιά εμφανίζεται ο αριθμός των ξεχωριστών κέντρων δεδομένων που έχει συνδεθεί η εφαρμογή. Ακριβώς δίπλα εμφανίζεται σε μορφή αντίστροφης μέτρησης ο χρόνος για την ανανέωση της πληροφορίας των κινούμενων μέσων που απεικονίζονται. Πατώντας πάνω στο μεγεθυντικό φακό δίνεται η δυνατότητα αναζήτησης της θέσης ενός συγκεκριμένου πλοίου

βάσει του ονόματος του ή του αναγνωριστικού του mmsi. Όπως φαίνεται και στην Εικόνα 25 ο χρήστης μπορεί να πληκτρολογήσει το πλαίσιο αναζήτησης όλο ή μέρος του ονόματος του πλοίου ή το αναγνωριστικό του και από την λίστα που εμφανίζεται μπορεί να επιλέξει σε ποιο πλοίο θέλει να εστιάσει. Μόλις ο χρήστης επιλέξει ένα πλοίο από την λίστα ο διαδραστικό χάρτης εστιάζει στο συγκεκριμένο πλοίο και το παρουσιάζει με διαφορετικό εικονίδιο (marker) όπως παρουσιάζεται στην Εικόνα 26.

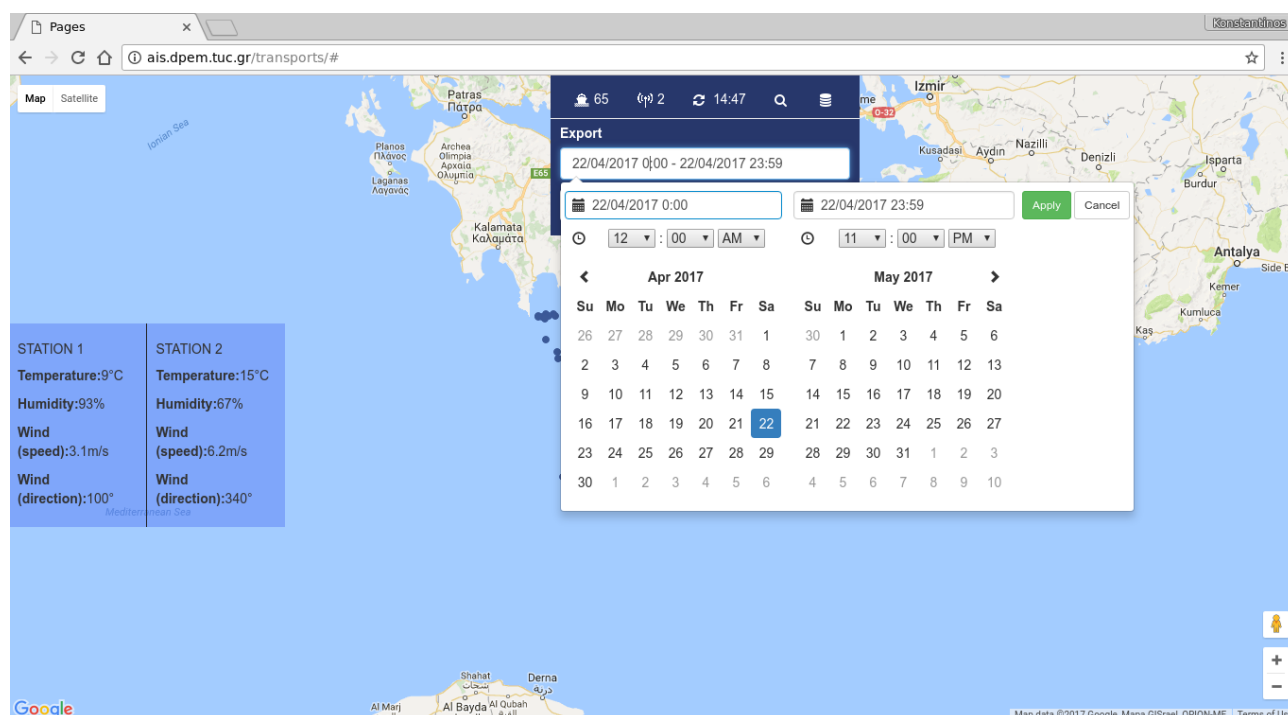


Εικόνα 25: Λειτουργία αναζήτησης πλοίου με αυτόματη συμπλήρωση



**Εικόνα 26:** Λειτουργία εστίασης εφαρμογής στο επιλεγμένο πλοίο

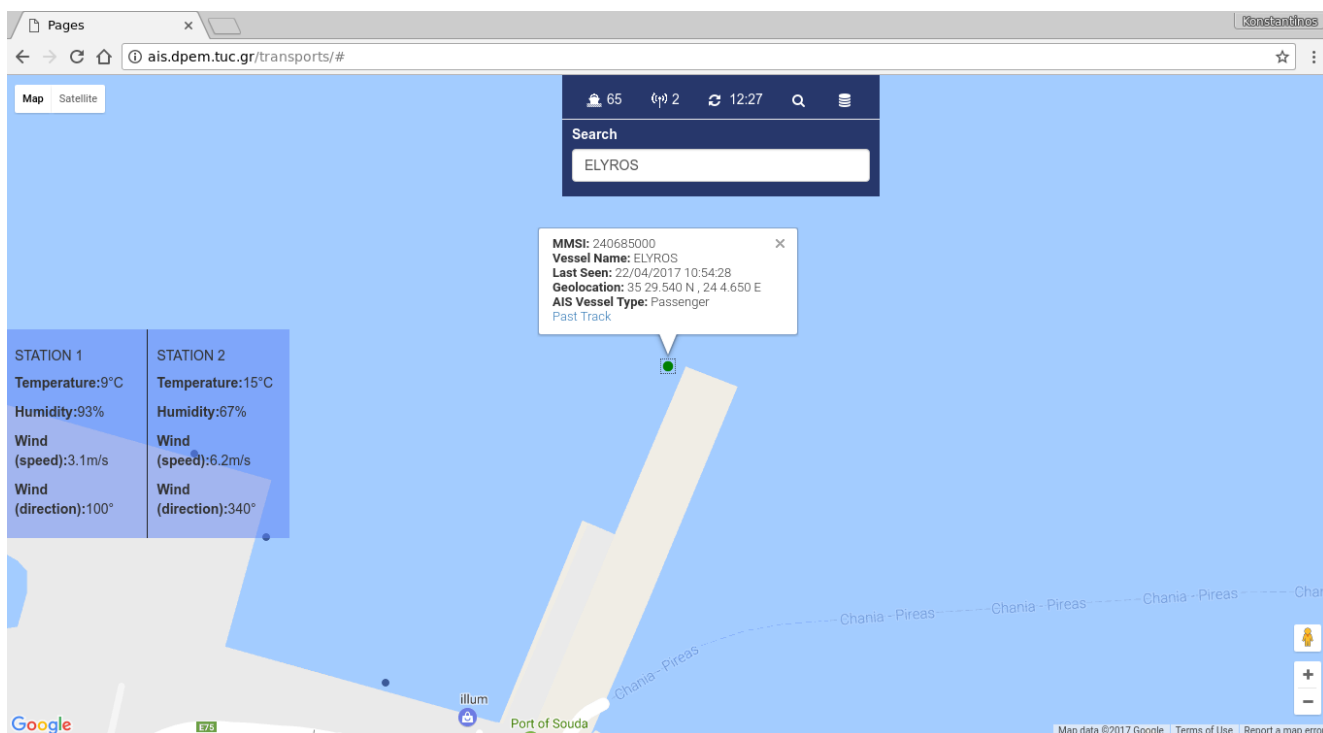
Μια άλλη δυνατότητα που παρέχει το κεντρικό panel είναι η εξαγωγή των δεδομένων σε μορφή οριοθετημένου κειμένου τα οποία μπορεί στην συνέχεια να χρησιμοποιήσει για περαιτέρω ανάλυση μέσω τρίτων εφαρμογών. Όπως φαίνεται και στην Εικόνα 27 ο χρήστης μπορεί να επιλέξει από το ημερολόγιο που εμφανίζεται όταν πατάει στο αντίστοιχο πλαίσιο ελέγχου το χρονικό διάστημα για το οποίο θέλει να εξαγάγει τα δεδομένα σε αρχείο. Το είδος η μορφή του αρχείου καθώς και τα περιεχόμενα του παρουσιάζονται εκτενέστερα στο επόμενο κεφάλαιο.



**Εικόνα 27:** Λειτουργία εξαγωγής δεδομένων

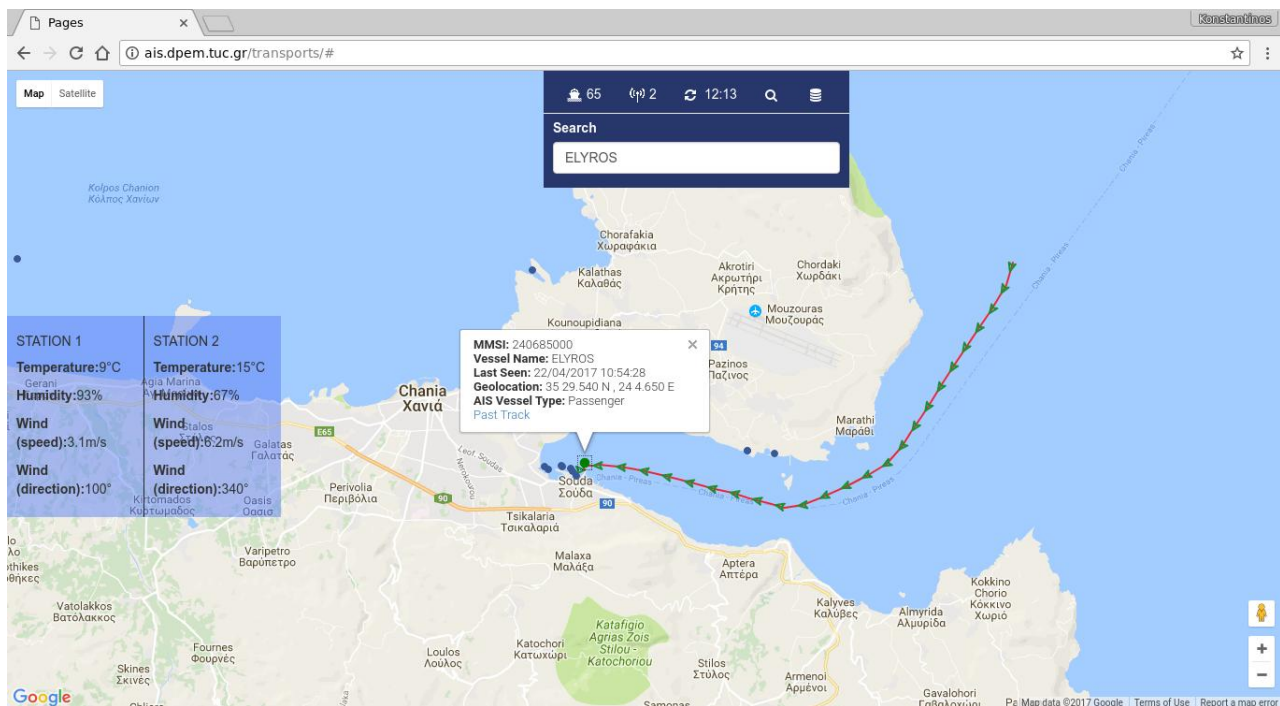
Πέρα από το κεντρικό πάνελ ενέργειες χρήστη επιτρέπονται και από τον διαδραστικό χάρτη. Πιο συγκεκριμένα ο χρήστης έχει την δυνατότητα πλοήγησης στον χάρτη και δυνατότητα zoom in zoom out δυνατότητες που είναι ενσωματωμένες από προεπιλογή σε κάθε διαδραστικό χάρτη Google Maps. Ωστόσο μέσω του Google Maps API που χρησιμοποιήσαμε έχουμε την δυνατότητα να παρέχουμε στον χρήστη επιπλέον ενέργειες που σχετίζονται με την συγκεκριμένη εφαρμογή. Μια από τις ενέργειες αυτές είναι το άνοιγμα αναδυόμενου παραθύρου με πληροφορίες (popup) όταν ο χρήστης πατήσει πάνω σε κάποιον marker. Όπως φαίνεται και στην Εικόνα 28 μόλις ο χρήστης πατήσει σε κάποιον marker ανοίγει ένα αναδυόμενο παράθυρο με πληροφορίες σχετικά με το πλοίο. Κάτω από τις πληροφορίες του πλοίου υπάρχει στοιχείο ελέγχου για την ανάκτηση του ίχνους του πλοίου.





Εικόνα 28: Λειτουργία αναδυόμενου παραθύρου με πληροφορίες

Με το πάτημα του στοιχείου ελέγχου για το ίχνος του πλοίου και με χρήση των δυνατοτήτων που δίνει το Google Maps API παρέχουμε στον χρήστη την δυνατότητα να δει πάνω στον χάρτη την πορεία του πλοίου (Εικόνα 29).



Εικόνα 29: Λειτουργία εμφάνισης πορείας πλοίου



Όπως και με τις άλλες συνιστώσες της εφαρμογής έτσι και με την απεικόνιση των δεδομένων η δομή της έχει στηθεί ώστε να παρέχει την μεγαλύτερη δυνατή ευελιξία επέκτασης της. Η χρήση του Google Maps API δίνει την παραπάνω δυνατότητα μέσω μιας πολύ πλούσιας βιβλιοθήκης όπως έχουμε περιγράψει σε προηγούμενη ενότητα.

Μελλοντικά λοιπόν στο κεντρικό πάνελ μπορούμε να προσθέσουμε στοιχεία ελέγχου για φιλτράρισμα βάσει επίπεδων (layers), επιπλέον φίλτρα βάσει των διάφορων χαρακτηριστικών των μετακινούμενων μέσων, εμπλουτισμός των αναδυόμενων παραθύρων με πληροφορίες και φιλικότερο σχεδιασμό κ.α.

Έχοντας δώσει έμφαση στην επεκτασιμότητα της παρουσίασης όλα τα παραπάνω χαρακτηριστικά μπορούν να ενσωματωθούν με ελάχιστο ή μηδενικό κόστος δομικών ή τεχνικών αλλαγών.

# Κεφάλαιο 4

## Συμπεράσματα χρήσης και λειτουργική επιβεβαίωση της εφαρμογής

Στο παρών κεφάλαιο παρουσιάζονται αρχικά τα αποτελέσματα χρήσης της εφαρμογής καθώς και οι λειτουργίες που έχουν αναπτυχθεί για τον έλεγχο της ορθής λειτουργίας της εφαρμογής.

### Αρχεία Καταγραφής

Ο έλεγχος λειτουργίας της εφαρμογής καθώς και η ανίχνευση σφαλμάτων μπορεί να επιτευχθεί από ένα σύστημα καταγραφής (logging) το οποίο έχει αναπτυχθεί για όλες τις συνιστώσες της εφαρμογής. Ήδη στις προηγούμενες ενότητες έχουν παρουσιαστεί τμηματικά κάποιες λειτουργίες του συστήματος καταγραφής. Ξεχωριστά αρχεία καταγραφής δημιουργούνται για της παρακάτω συνιστώσες της εφαρμογής και παρουσιάζονται στις ακόλουθες εικόνες.

1. Υπηρεσία εφαρμογής Web (Εικόνα 30)
2. Υπηρεσία συλλογής Μεταδεδομένων Κινουμένων Μέσων (Εικόνα 31)
3. Υπηρεσία συλλογής Περιβαλλοντολογικών Δεδομένων (Εικόνα 32)
4. Υπηρεσία συλλογής δεδομένων από τα κέντρα δεδομένων (Εικόνα 34)

Για την δημιουργία του αρχείου καταγραφής της Υπηρεσίας εφαρμογής WEB χρησιμοποιήθηκε ο ενσωματωμένος μηχανισμός καταγραφής του CAKEPHP.

```
2017-02-24 19:05:06 Error: [Error] Call to a member function toArray() on null
Request URL: /marittraffic/export
Referer URL: http://ais.dpem.tuc.gr/marittraffic/
Client IP: 147.27.10.186
Stack Trace:
#0 /var/www/html/marittraffic/src/Controller/VesselsController.php(91): App\Model\Table\ClimaTable->getClimaAtTime(Array)
#1 [internal function]: App\Controller\VesselsController->export()
#2 /var/www/html/marittraffic/vendor/cakephp/cakephp/src/Controller/Controller.php(429): call_user_func_array(Array, Array)
#3 /var/www/html/marittraffic/vendor/cakephp/cakephp/src/Routing/Dispatcher.php(114): Cake\Controller\Controller->invokeAction()
#4 /var/www/html/marittraffic/vendor/cakephp/cakephp/src/Routing/Dispatcher.php(87): Cake\Routing\Dispatcher->invoke(Object(App\Controller\VesselsController)
#5 /var/www/html/marittraffic/webroot/index.php(36): Cake\Routing\Dispatcher->dispatch(Object(Cake\Network\Request), Object(Cake\Network\Response))
#6 {main}

2017-02-24 19:05:06 Error: [Cake\Routing\Exception\MissingControllerException] Controller class Css could not be found.
Request URL: /marittraffic/css/cake.css
Referer URL: http://ais.dpem.tuc.gr/marittraffic/export
Client IP: 147.27.10.186
Stack Trace:
#0 /var/www/html/marittraffic/webroot/index.php(36): Cake\Routing\Dispatcher->dispatch(Object(Cake\Network\Request), Object(Cake\Network\Response))
#1 {main}

2017-02-24 19:05:06 Error: [Cake\Routing\Exception\MissingControllerException] Controller class Img could not be found.
Request URL: /marittraffic/img/cake.power.gif
Referer URL: http://ais.dpem.tuc.gr/marittraffic/export
Client IP: 147.27.10.186
Stack Trace:
#0 /var/www/html/marittraffic/webroot/index.php(36): Cake\Routing\Dispatcher->dispatch(Object(Cake\Network\Request), Object(Cake\Network\Response))
#1 {main}
```

Εικόνα 30: Αρχείο Καταγραφής Υπηρεσίας Υπηρεσία εφαρμογής Web

```

2017-03-24 17:46:06,398 - INFO - crawling finished
2017-03-24 17:48:06,559 - INFO - crawling finished
2017-03-24 17:50:06,736 - INFO - crawling finished
2017-03-24 17:52:07,034 - INFO - crawling finished
2017-03-24 17:54:06,989 - INFO - crawling finished
2017-03-24 17:56:06,305 - INFO - crawling finished
2017-03-24 17:58:06,466 - INFO - crawling finished
2017-03-24 18:00:05,869 - INFO - crawling finished
2017-03-24 18:02:06,416 - INFO - crawling finished
2017-03-24 18:04:05,807 - INFO - crawling finished
2017-03-24 18:06:06,198 - INFO - crawling finished
2017-03-24 18:08:06,424 - INFO - crawling finished

```

**Εικόνα 31: Αρχείο Καταγραφής Υπηρεσίας συλλογής Μεταδεδομένων Κινουμένων Μέσων**

```

2017-02-09 20:21:17,934 - DEBUG - "GET /data/2.5/weather?units=metric&lat=35.490193&lon=24.0666989&appid=3b9d05593854ad7f55ff6827974be13d HTTP/1.1" 200 434
2017-02-09 20:21:18,078 - INFO - Starting new HTTP connection (1): api.openweathermap.org
2017-02-09 20:21:18,205 - DEBUG - "GET /data/2.5/weather?units=metric&lat=35.3439686&lon=25.1468797&appid=3b9d05593854ad7f55ff6827974be13d HTTP/1.1" 200 443
2017-02-09 20:21:24,419 - INFO - Starting new HTTP connection (1): api.openweathermap.org
2017-02-09 20:21:24,554 - DEBUG - "GET /data/2.5/weather?units=metric&lat=35.490193&lon=24.0666989&appid=3b9d05593854ad7f55ff6827974be13d HTTP/1.1" 200 434
2017-02-09 20:21:24,661 - INFO - Starting new HTTP connection (1): api.openweathermap.org
2017-02-09 20:21:24,786 - DEBUG - "GET /data/2.5/weather?units=metric&lat=35.3439686&lon=25.1468797&appid=3b9d05593854ad7f55ff6827974be13d HTTP/1.1" 200 429
2017-02-09 20:21:44,247 - INFO - Starting new HTTP connection (1): api.openweathermap.org
2017-02-09 20:21:44,378 - DEBUG - "GET /data/2.5/weather?units=metric&lat=35.490193&lon=24.0666989&appid=3b9d05593854ad7f55ff6827974be13d HTTP/1.1" 200 434
2017-02-09 20:21:44,512 - INFO - Starting new HTTP connection (1): api.openweathermap.org
2017-02-09 20:21:44,644 - DEBUG - "GET /data/2.5/weather?units=metric&lat=35.3439686&lon=25.1468797&appid=3b9d05593854ad7f55ff6827974be13d HTTP/1.1" 200 443

```

**Εικόνα 32: Αρχείο Καταγραφής Υπηρεσίας Περιβαλλοντολογικών Δεδομένων**

```

2016-09-19 20:43:40,340 - INFO - Connecting to STATION 1 (147.27.70.91:39422) socket.
2016-09-19 20:43:40,343 - INFO - Connecting to STATION 2 (147.27.70.103:39422) socket.
2016-09-19 20:43:40,348 - INFO - Connection established: STATION 1 (147.27.70.91:39422).
2016-09-19 20:43:40,350 - INFO - Closing connection to STATION 1 socket
2016-09-19 20:43:40,350 - INFO - Connection established: STATION 2 (147.27.70.103:39422).
2016-09-19 20:43:40,350 - INFO - Closing connection to STATION 2 socket
2016-09-19 20:43:50,526 - INFO - Connecting to STATION 1 (147.27.70.91:39422) socket.
2016-09-19 20:43:50,529 - INFO - Connecting to STATION 2 (147.27.70.103:39422) socket.
2016-09-19 20:43:50,535 - INFO - Connection established: STATION 1 (147.27.70.91:39422).
2016-09-19 20:43:50,535 - INFO - Connection established: STATION 2 (147.27.70.103:39422).
2016-09-19 20:43:50,537 - INFO - Closing connection to STATION 2 socket
2016-09-19 20:43:50,542 - INFO - Closing connection to STATION 1 socket
2016-09-19 20:44:00,544 - INFO - Connecting to STATION 2 (147.27.70.103:39422) socket.
2016-09-19 20:44:00,547 - INFO - Connection established: STATION 2 (147.27.70.103:39422).
2016-09-19 20:44:00,550 - INFO - Closing connection to STATION 2 socket
2016-09-19 20:44:00,553 - INFO - Connecting to STATION 1 (147.27.70.91:39422) socket.
2016-09-19 20:44:00,557 - INFO - Connection established: STATION 1 (147.27.70.91:39422).

```

**Εικόνα 33: Αρχείο Καταγραφής Υπηρεσίας συλλογής δεδομένων**

## Γεωγραφική Κάλυψη

Για την λειτουργία επιβεβαίωσης της εφαρμογής έγινε σύνδεση της εφαρμογής με κέντρα δεδομένων κίνησης πλοίων και αεροσκαφών. Η θέση των κεραιών καθώς και η γεωγραφική κάλυψη όπως αυτή αποτυπώθηκε από τα λαμβανόμενα δεδομένα παρουσιάζεται στην Εικόνα 34 και Εικόνα 35, για την κίνηση των πλοίων και το αεροσκαφών αντίστοιχα.



## Εξαγωγή Δεδομένων

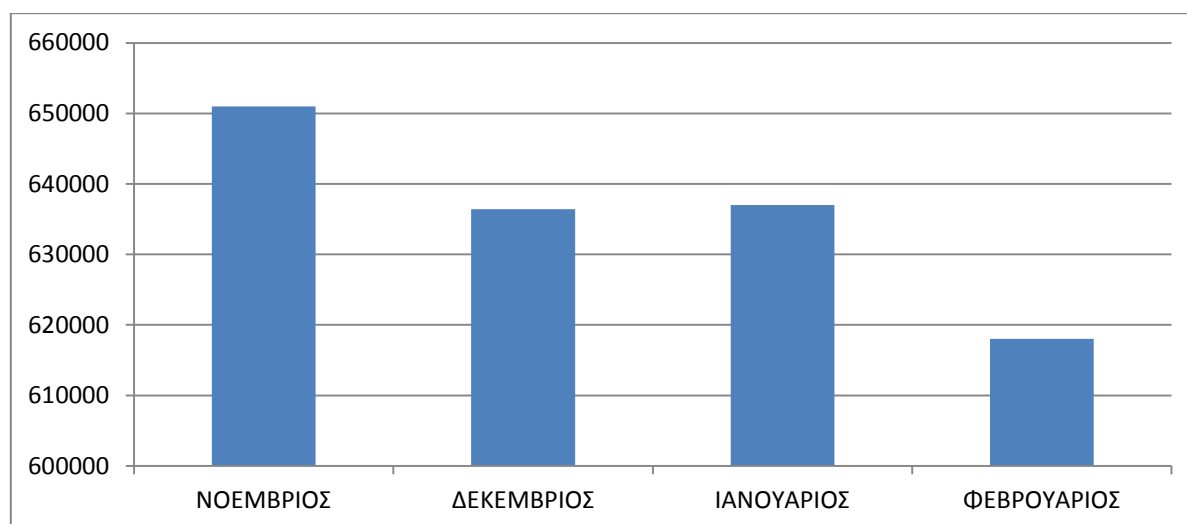
Είδαμε στην προηγούμενη ενότητα ότι είναι δυνατή η εξαγωγή συγκεντρωτικών δεδομένων από τον χρήστη για μια συγκεκριμένη χρονική περίοδο. Τα εξαγόμενα δεδομένα συνδυάζουν την πληροφορία κίνησης που έχει συλλεχθεί από τα κέντρα δεδομένων, μαζί με τα δεδομένα που έχουν ληφθεί από τους αντίστοιχους σταθμούς και υπηρεσίες. Τα αρχεία αυτά είναι σε μορφή οριοθετημένου κειμένου (.csv) και παρέχουν στον χρήστη όλες εκείνες της απαραίτητες πληροφορίες (σύμφωνα και με την σχετική βιβλιογραφία) που απαιτούνται για την διεξαγωγή περιβαλλοντολογικών μελετών και ανάλυση της επιβάρυνσης του περιβάλλοντος από την κίνηση των μεσών αυτών. Ένα δείγμα του αρχείου αυτού παρουσιάζεται στην Εικόνα 36.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
id	received_time	lat	lon	mmsi_id	imo	gt	Year_build	flag	Type	station	temperature	wind	humidity	wind_deg
1	24/03/2017 19:57:36	36.6359	23.43872	538005899	9331921	31236	2007	Marshall Is [MH]	Cargo	STATION 1	12	2.6	93	170
2	24/03/2017 19:57:35	35.51717	23.63747	352962000	9712474	21801	2015	Panama [PA]	Cargo	STATION 1	12	1.17	93	13
3	24/03/2017 19:57:35	36.41858	22.96693	255740000	9271573	78896	2004	Portugal [PT]	Tanker	STATION 1	12	2.6	93	170
4	24/03/2017 19:57:32	36.32685	22.59533	220567000	9288930	23246	2004	Denmark [DK]	Tanker	STATION 1	12	1.17	93	13
5	24/03/2017 19:57:31	36.40567	23.2635	249852000	9708772	29705	2017	Malta [MT]	Tanker	STATION 1	12	2.6	93	170
6	24/03/2017 19:57:27	36.39178	23.52643	538001857	8401133	15375	1986	Marshall Is [MH]	Cargo - Hazard A (Major)	STATION 1	12	1.17	93	13
7	24/03/2017 19:57:26	36.6903	23.43687	511370000	9074729	15899	1995	Palau [PW]	Cargo	STATION 1	12	2.6	93	170
8	24/03/2017 19:57:15	36.35175	22.58453	620091000	7526924	7886	1977	Comoros [KM]	Tanker	STATION 1	12	1.17	93	13
9	24/03/2017 19:57:13	37.06827	23.47905	357409000	9111357	16418	1995	Panama [PA]	Cargo	STATION 1	12	2.6	93	170
10	24/03/2017 19:57:12	36.57248	22.01292	636015129	9007817	23953	1991	Liberia [LR]	Cargo - Hazard A (Major)	STATION 1	12	1.17	93	13

Εικόνα 36: Δείγμα αρχείου εξαγωγής δεδομένων csv

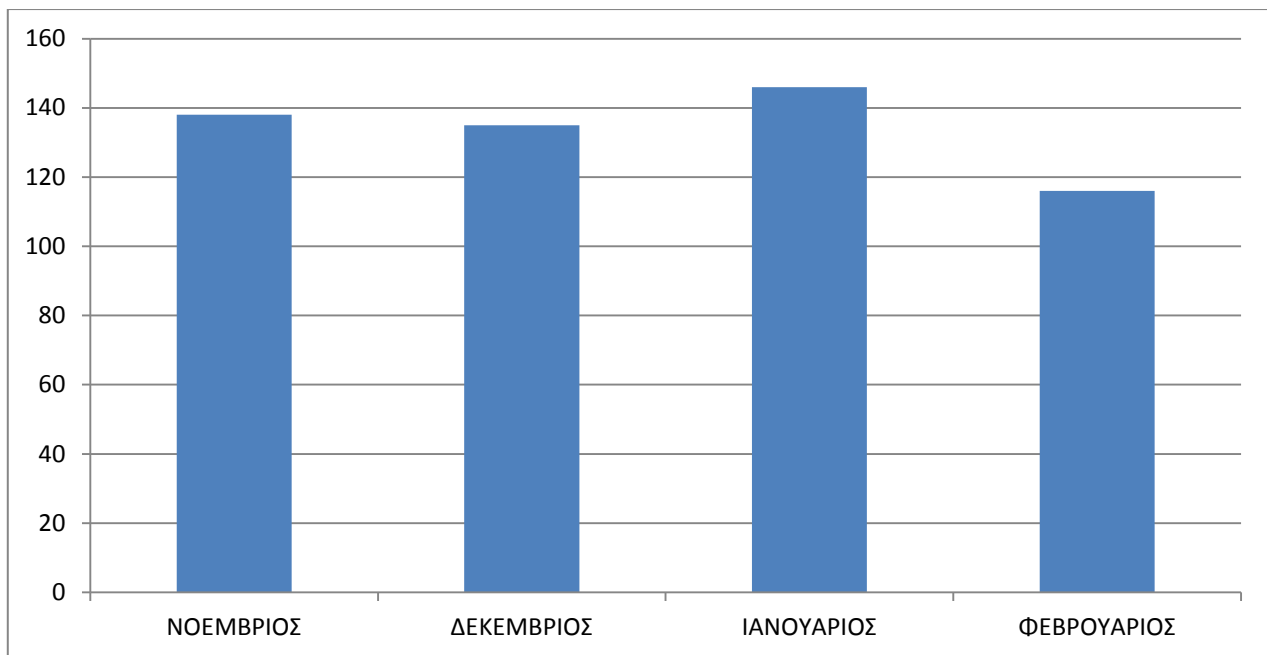
## Στατιστικά Χρήσης

Στην συγκεκριμένη ενότητα παρουσιάζονται κάποια στατιστικά δεδομένα από την λειτουργία της εφαρμογής που χρησιμοποιήθηκαν για την ποιοτική επιβεβαίωση της. Στο Γράφημα 1 παρουσιάζεται ο αριθμός των αναφορών θέσεων ανά μήνα που έλαβε η εφαρμογή από τον σταθμό συλλογής δεδομένων πλοίων της περιοχής των χανίων. Στο Γράφημα 2 παρουσιάζεται ο αριθμός των διαφορετικών πλοίων ανά μήνα για τον ίδιο σταθμό.



Γράφημα 1: Αναφορές θέσης Πλοίων ανα μήνα από σταθμό AIS Χανίων

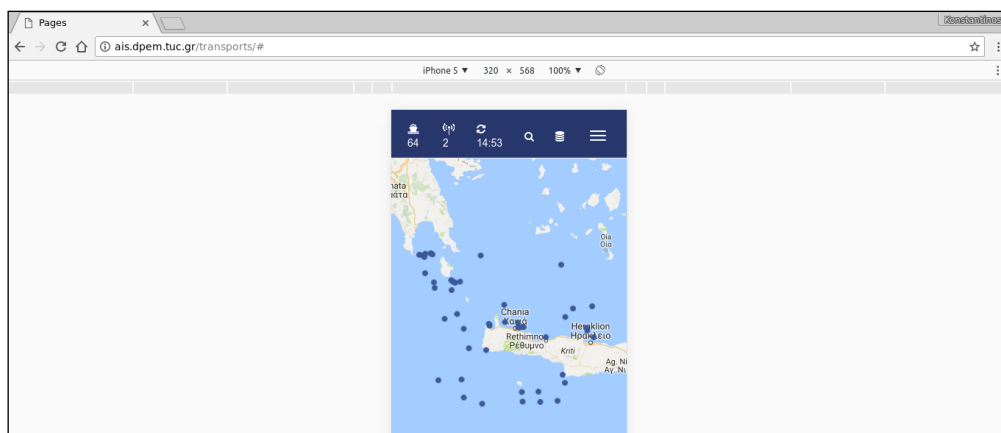




Γράφημα 2: Αριθμός ξεχωριστών πλοίων ανα μήνα από σταθμό AIS Χανίων

## Πρόσβαση από Κινητές Συσκευές Responsive Design

Είναι γεγονός ότι στις μέρες μας η πρόσβαση στις ιστοσελίδες από κινητές συσκευές τείνει να ξεπεράσει την πρόσβαση από συμβατικές συσκευές. Ήταν αρχική σχεδιαστική απαίτηση ο σχεδιασμός μιας εφαρμογής προσβάσιμης και λειτουργικής και από κινητές συσκευές. Η εφαρμογή αναπτύχθηκε σύμφωνα με τις αρχές του responsive web design χρησιμοποιώντας την βιβλιοθήκη ανοικτού κώδικα “Bootstrap” μέσω του οποίου η εφαρμογή ανιχνεύει τις διάφορες μεταβλητές από το εξωτερικό και εσωτερικό περιβάλλον και ανταποκρίνεται ανάλογα στον επισκέπτη. Το τελικό αποτέλεσμα παρουσιάζεται στην Εικόνα 37 όπου γίνεται και εμφανή το γεγονός ότι η εφαρμογή προσαρμόζει το μέγεθος και τα βασικά χαρακτηριστικά της (μενού, χάρτης, κείμενο) ανάλογα με τις διαστάσεις της οθόνης της συσκευής του χρήστη.



Εικόνα 37: Responsive Design

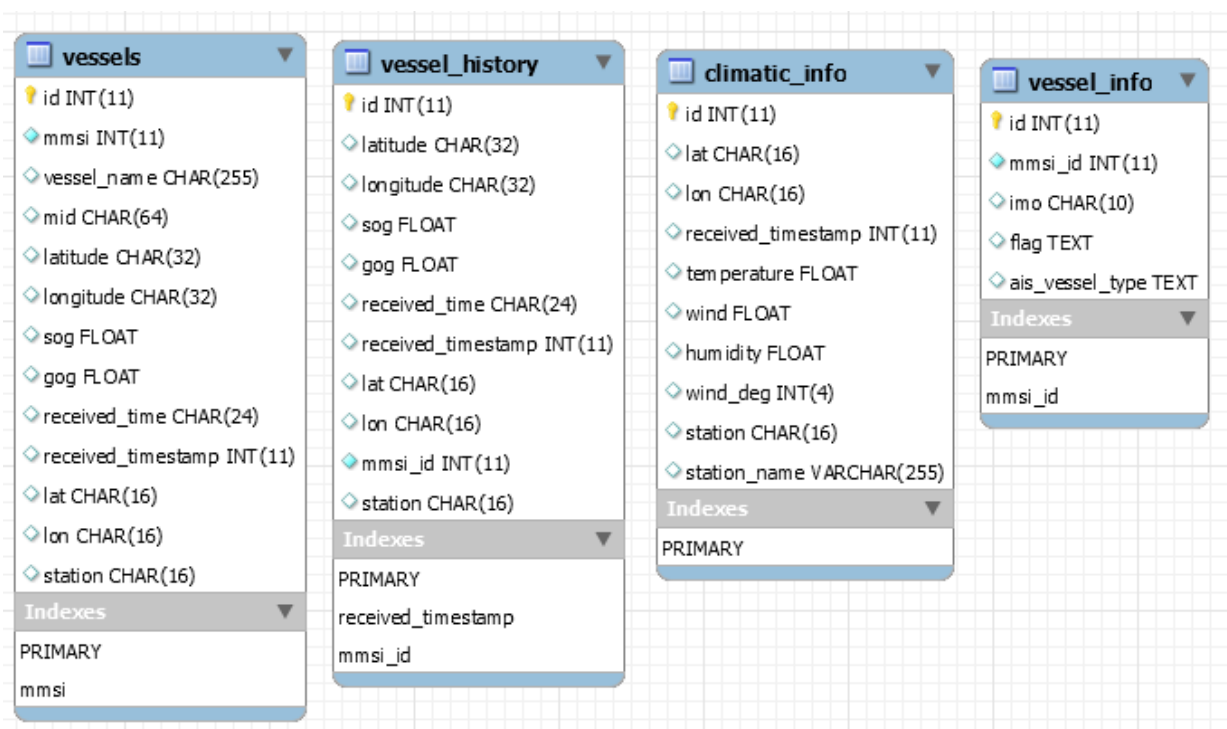
## Χρήση Βάσεως Δεδομένων και εκτεταμένο μοντέλο οντοτήτων-συσχετισμών (ΕΕ-R μοντέλο)

Η χρήση της βάσης δεδομένων για δεδομένα ενός ολόκληρου μήνα για ένα σταθμό συλλογής δεδομένων παρουσιάζεται στην Εικόνα 38. Παρατηρούμε την δραματική αύξηση του μεγέθους του πίνακα που συλλέγει τα δεδομένα κίνησης. Από το γεγονός αυτό συμπεραίνουμε ότι η δημιουργία του πίνακα προσωρινής αποθήκευσης των μέσων που δήλωσαν πρόσφατα την θέση τους ήταν μονόδρομος δεδομένου ότι θέλαμε η εφαρμογή να ανταποκρίνεται άμεσα και δεύτερον σε περίπτωση που εφαρμογή συνδεθεί με περισσότερα κέντρα δεδομένων κρίνεται αναγκαία η χρησιμοποίηση ειδικευμένων καταναμημένων βάσεων για την αποθήκευση της πληροφορίας. Κάτι τέτοιο φυσικά λόγω του τρόπου σχεδίασης και ανάπτυξης της εφαρμογής, όπως έχει ήδη περιγραφεί, μπορεί να υλοποιηθεί με ελάχιστο προγραμματιστικό κόστος.

Name ^	Rows	Size	Created	Updated	Engine	Comment	Type
climatic_info	119,437	12.5 MiB	2017-03-14 02:42:36	2017-03-24 18:03:03	InnoDB		Table
vessels	8,082	8.8 MiB	2016-08-02 09:06:07	2017-03-15 10:09:45	MyISAM		Table
vessel_history	7,062,177	1.3 GiB	2016-08-02 09:06:35	2017-03-15 10:09:45	MyISAM		Table
vessel_info	8,046	720.0 KiB	2017-03-14 02:42:36	2017-03-24 18:02:06	InnoDB		Table

Εικόνα 38: Τάξη Μεγέθους Δεδομένων σε σχεσιακή βάση

Το μοντέλο οντοτήτων-συσχετισμών E-R όπως αυτό αποτυπώνεται από τον ορισμό των μοντέλων ORM φαίνεται στην Εικόνα 39.



Εικόνα 39: E-R σχεσιακής βάσης Δεδομένων.



# Κεφάλαιο 5

## Μελλοντικές επεκτάσεις

Στο κεφάλαιο αυτό θα δοθούν κάποιες προτάσεις για περαιτέρω επέκταση της εφαρμογής μελλοντικά με στόχο να εξελιχθεί σε ένα ολοκληρωμένο και ενιαίο σύστημα ανάλυσης των περιβαλλοντικών επιπτώσεων κατά την πορεία μετακινούμενων οχημάτων.

Όπως έχει ήδη αναφερθεί, παρότι η εφαρμογή στην παρούσα φάση εστιάζει στην απεικόνιση περιβαλλοντικών δεδομένων κατά την κίνηση πλοίων δεν συνδέεται αποκλειστικά με τις συγκεκριμένες πληροφορίες.

Η ανάπτυξη της εφαρμογής έγινε με γνώμονα την επέκταση της για απεικόνιση οποιουδήποτε μετακινούμενου μέσου και απεριόριστου αριθμού περιβαλλοντικών μεταβλητών, μέσω εύκολα παραμετροποιήσιμης σύνδεσης με κέντρα δεδομένων από τα οποία θα λαμβάνονται πληροφορίες σχετικά με τον εκάστοτε τύπο μετακινούμενου μέσου.

Σε πρώτη φάση, λοιπόν η εφαρμογή μπορεί να επεκταθεί άμεσα ώστε να απεικονίζει δεδομένα από περισσότερους σταθμούς - λιμάνια καθώς και περισσότερες περιβαλλοντικές συνιστώσες.

Επίσης μπορεί να γίνει χρήση της δυνατότητας του διαδραστικού χάρτη για απεικόνιση των περιβαλλοντικών δεδομένων σε επίπεδα (layers). Στα επιπλέον layers μπορεί να γίνεται απεικόνιση της περιβαλλοντικής επιβάρυνσης που προκύπτει από την κίνηση των μεσών δεδομένου ότι όλα τα δεδομένα που είναι απαραίτητα για τον υπολογισμό της ατμοσφαιρικής ρύπανσης έχουν συλλεχθεί και αποθηκευτεί και είναι διαθέσιμα στο πληροφοριακό σύστημα. Θα μπορούσε να αναπτυχθεί δηλαδή ένα πραγματικού χρόνου module το οποίο θα υπολογίζει την περιβαλλοντική επιβάρυνση. Εν συνεχεία ο χρήστης θα μπορούσε να επιλέξει ποια περιβαλλοντική συνιστώσα θέλει να παρατηρήσει και να εμφανίζεται το αντίστοιχο layer.

Μια τρίτη μελλοντική επέκταση της εφαρμογής είναι η σύνδεση της με περισσότερα κέντρα δεδομένων που στέλνουν δεδομένα κατά την κίνηση εναέριων μετακινούμενων οχημάτων και υλοποίηση του module που αναφέρθηκε στην προηγούμενη παράγραφο για τα μέσα αυτά.

Καθώς οι παραπάνω επεκτάσεις θα προκαλέσουν δραματική αύξηση του όγκου δεδομένων που καλείται να διαχειριστεί η εφαρμογή οδηγούμαστε σε μια ακόμα πρόταση επέκτασης της που είναι η αποθήκευση των δεδομένων σε κατακευματισμένες βάσεις δεδομένων (distributed databases) και κατά συνέπεια την γενικότερη επέκταση της για διαχείριση «μεγάλων» δεδομένων (big data).

Μια ακόμα χρήσιμη επέκταση της εφαρμογής θα ήταν η έκδοση της ως υπηρεσία μέσω

Restful API [9]. Οι χρήστες της υπηρεσίας θα μπορούν μέσω κλήσεων στο Restful API να λάβουν κατάλληλα μορφοποιημένα δεδομένα σε πραγματικό χρόνο τα οποία στην συνέχεια μπορούν να χρησιμοποιήσουν για περαιτέρω ανάλυση μέσω τρίτων εφαρμογών. Μια από αυτές τις τρίτες εφαρμογές αποτελεί και μια τελευταία πρόταση για επέκταση της εφαρμογής με την ανάπτυξη εφαρμογής για κινητά που θα κάνει χρήση της παραπάνω υπηρεσίας για μικρότερη επιβάρυνση στον όγκο δεδομένων που μεταφέρονται.

Κλείνοντας να αναφέρουμε ότι στην παρούσα εργασία σκοπός ήταν να θέσουμε στέρεες βάσεις χρησιμοποιώντας την ακμή της τεχνολογίας και τις καλύτερες πρακτικές για την ανάπτυξη ενός ολοκληρωμένου συστήματος συλλογής, διαχείρισης και απεικόνισης σε πραγματικό χρόνο περιβαλλοντικών δεδομένων κατά την πορεία μετακινούμενων οχημάτων.

Υλοποιήσαμε μια εφαρμογή η οποία πληροί τις παραπάνω προδιαγραφές κατά την πορεία πλοίων στο λιμάνι της Σούδας και του Ηρακλείου και μπορεί να ενσωματώσει πληροφορία κίνησης αεροσκαφών από αντίστοιχους σταθμούς.

Μέσω και των παραπάνω μελλοντικών επεκτάσεων της, ελπίζουμε στην δημιουργία ενός εργαλείου εποπτείας και αντιμετώπισης των περιβαλλοντικών επιπτώσεων που επιφέρουν παγκοσμίως οι αυξανόμενες μετακινήσεις οχημάτων.

# Βιβλιογραφία

- [1] U. Environmental Protection Agency. [Online]. Available: <https://www.epa.gov/international-cooperation/reducing-air-pollution-international-transportation>.
- [2] Wikipedia. [Online]. Available: [https://en.wikipedia.org/wiki/Modular\\_programming](https://en.wikipedia.org/wiki/Modular_programming).
- [3] Wikipedia. [Online]. Available: <https://en.wikipedia.org/wiki/Model-view-controller>.
- [4] OsinLab. [Ηλεκτρονικό]. Available: [www.osinlab.com/what-is-an-orm/](http://www.osinlab.com/what-is-an-orm/).
- [5] Python. [Online]. Available: <https://www.python.org/doc/essays/comparisons/>.
- [6] Γ. Νικολουδάκης, ΑΝΑΠΤΥΞΗ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΓΙΑ ΤΗΝ ΔΙΑΧΕΙΡΙΣΗ ΚΑΙ ΤΟΝ ΥΠΟΛΟΓΙΣΜΟ ΤΟΥ ΕΝΕΡΓΕΙΑΚΟΥ ΑΠΟΤΥΠΩΜΑΤΟΣ ΜΕΣΩΝ ΜΑΖΙΚΗΣ ΜΕΤΑΦΟΡΑΣ., 2015.
- [7] S. Papaefthimiou, I. Sitzimis, K. Andriosopoulos, "A methodological approach for environmental characterization of ports," *Maritime Policy & Management*, vol. 44, pp. 81-93, 2017.
- [8] S. Papaefthimiou, A. Maragkogianni, K. Andriosopoulos, "Evaluation of cruise ships emissions in the Mediterranean basin: the case of Greek ports," *International Journal of Sustainable Transportation*, vol. 10, p. 985, 2016.
- [9] Wikipedia. [Online]. Available: [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer).
- [10] Κ. Κουφός, ΥΠΟΛΟΓΙΣΜΟΣ ΕΝΕΡΓΕΙΑΚΟΥ ΑΠΟΤΥΠΩΜΑΤΟΣ ΔΙΕΘΝΩΝ ΑΕΡΟΛΙΜΕΝΩΝ, 2015.
- [11] Προστασία Περιβάλλοντος, Δ.Α. Αυλωνίτης.
- [12] Wikipedia, "Automatic Dependent Surveillance," [Online]. Available: [https://en.wikipedia.org/wiki/Automatic\\_dependent\\_surveillance\\_%E2%80%93\\_broadcast](https://en.wikipedia.org/wiki/Automatic_dependent_surveillance_%E2%80%93_broadcast).
- [13] Wikipedia, "Automatic identification system," [Online]. Available: [https://en.wikipedia.org/wiki/Automatic\\_identification\\_system](https://en.wikipedia.org/wiki/Automatic_identification_system).