



Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Αυτόνομη διαδικτυακή πλατφόρμα για
την περιήγηση και εξερεύνηση
σημασιολογικών (RDF) δεδομένων

Αλέξανδρος Βασιλάκης
CSD 3612

Επίβλεψη: Παύλος Φαφαλιός

Επόπτης καθηγητής: Γιάννης Τζίτζικας

Μέλος της Επιτροπής Παρακολούθησης/Αξιολόγησης: Δημήτρης Πλεξουσάκης

Ηράκλειο, Ιούλιος 2024

Table of Contents

<i>Λίστα εικόνων</i>	3
<i>1. Εισαγωγή</i>	4
<i>2. Υπόβαθρο και Πλαίσιο Εργασίας</i>	5
<i>2.1. Το μοντέλο δεδομένων RDF</i>	5
<i>2.1.1. RDF</i>	5
<i>2.1.2. URI</i>	5
<i>2.1.3. SPARQL</i>	6
<i>2.2 Το έργο SeaLiT και τα δεδομένα του</i>	6
<i>2.2.1 Οντολογία των δεδομένων SeaLit</i>	7
<i>3. Εφαρμογή Πλοιήγησης σε RDF δεδομένα</i>	9
<i>3.1. Αρχιτεκτονική και Βασικές λειτουργίες συστήματος</i>	9
<i>3.1.1. Ανάγνωση RDF αρχείων και Φόρτωση γράφου στη μνήμη</i>	9
<i>3.1.2. Επισκόπηση έργου και βασικών οντοτήτων</i>	10
<i>3.1.3. Σελιδοποίηση</i>	10
<i>3.1.4. Ταξινόμηση</i>	11
<i>3.1.5. Φίλτρα</i>	14
<i>3.1.6. Διαγράμματα(Charts)</i>	17
<i>3.1.7. Προβολή λεπτομερειών μιας καταχώρησης</i>	17
<i>3.2. Διεπαφή χρήστη (User Interface)</i>	18
<i>3.3. Τεχνολογίες</i>	25
<i>3.3.1. Γλώσσα Προγραμματισμού</i>	25
<i>3.3.2. Βιβλιοθήκες που χρησιμοποιήθηκαν:</i>	25
<i>4. Μοντέλο Διαμόρφωσης(<i>Configuration model</i>)</i>	27
<i>4.1. Ιδιότητες αρχείου διαμόρφωσης</i>	27
<i>4.2 Σχεδιασμός αρχείου διαμόρφωσης</i>	27
<i>4.2.1. Παράδειγμα αρχείου διαμόρφωσης</i>	27
<i>5. Εγκατάσταση Συστήματος</i>	33
<i>6. Απαιτήσεις Συστήματος και Απόδοση</i>	33
<i>7. Επίλογος</i>	34
<i>7.1. Βασικά συμπεράσματα</i>	34
<i>7.2. Αδυναμίες</i>	34
<i>7.3. Μελλοντική εργασία (Βελτιστοποίηση εφαρμογής – Βελτιώσεις)</i>	34
<i>8. Αναφορές / Πηγές</i>	35

Λίστα εικόνων

Εικόνα 1: Υψηλού επιπέδου ιδιότητες και κλάσεις του προτύπου CIDOC-CRM.....	7
Εικόνα 2: Παράδειγμα μοντελοποίησης πληροφοριών ενός πλοίου.....	8
Εικόνα 3: Ταξινόμηση Z-A	11
Εικόνα 4: Ταξινόμηση A-Z	11
Εικόνα 5: Τυχαία ταξινόμηση ενός column	12
Εικόνα 6: Αύξουσα Ταξινόμηση	13
Εικόνα 7: Φθίνουσα Ταξινόμηση	13
Εικόνα 8: Αναζήτηση αποτελεσμάτων που "ξεκινούν με"	14
Εικόνα 9: Εμφάνιση αποτελεσμάτων με κενή τιμή καταχώρησης.....	14
Εικόνα 10: Αναζήτηση αριθμών ενός εύρους τιμών.....	15
Εικόνα 11: Αναζήτηση αριθμούς "μεγαλύτερων ή ίσων από"	15
Εικόνα 12: Αναζήτηση ημερομηνιών "μετά από την"	16
Εικόνα 13: Αναζήτηση ακριβής ημερομηνίας	16
Εικόνα 14: Ταξινόμηση που πρέπει να ισχύουν " και τα δύο " φίλτρα	17
Εικόνα 15: Ταξινόμηση που αρκεί να ισχύει " ένα από τα δύο " φίλτρα.....	17
Εικόνα 16: Αρχική σελίδα εφαρμογής	18
Εικόνα 17: Σελίδα εμφάνισης μιας κατηγορίας	19
Εικόνα 18: Εμφάνιση διαγραμμάτων columns μιας κατηγορίας	20
Εικόνα 19: Δυνατότητα εμφάνισης πολλαπλών διαγραμμάτων την ίδια στιγμή	21
Εικόνα 20: Σελίδα με περισσότερες πληροφορίες μιας καταχώρησης	22
Εικόνα 21: Ένα URI, μπορεί να κληθεί από ένα πεδίο πολλαπλές φορές	23
Εικόνα 22: Σελίδα "About" που αφορά την εφαρμογή	24
Εικόνα 23: Αποκλειστικά βασικά στοιχεία της εφαρμογής	28
Εικόνα 24: Πεδίο με αριθμό των κατηγοριών των οντοτήτων.....	28
Εικόνα 25: Στοιχεία που αφορούν κάθε κατηγορία	28
Εικόνα 26: Παράδειγμα με στοιχεία κάθε κατηγορίας.....	29
Εικόνα 27: Σελίδα χωρίς την παράμετρο OPTIONAL στο query που δίνει τα αποτελέσματα	30
Εικόνα 28: Σελίδα με την παράμετρο OPTIONAL στο query που δίνει τα αποτελέσματα ...	31
Εικόνα 29: Σελίδα με περίπτωση που δεν έχουν όλες οι μεταβλητές τιμή	32

1. Εισαγωγή

Ο σκοπός της παρούσας διπλωματικής εργασίας ήταν η δημιουργία μιας αυτόνομης διαδικτυακής πλατφόρμας για την εύκολη και γρήγορη περιήγηση και εξερεύνηση σημασιολογικών δεδομένων. Η χρησιμοποιούμενη μέθοδος αποθήκευσης και μεταφοράς δεδομένων είναι μια τεχνολογία του Σημασιολογικού Ιστού, η RDF. Με τον όρο «αυτόνομη» διαδικτυακή πλατφόρμα, σημαίνει ότι δεν απαιτεί την σύνδεση με κάποιο σημασιολογικό αποθετήριο (triplestore), αλλά φορτώνει στη μνήμη τον γράφο γνώσης για την εκτέλεση των (SPARQL) ερωτημάτων. Σύμφωνα με την RDF(Resource Description Framework), τα δεδομένα αποθηκεύονται με την μορφή τριπλετών (triples) της μορφής subject-predicate-object (υποκείμενο-κατηγόρημα-αντικείμενο)).

Η SPARQL αποτελεί την βασική γλώσσα για την υποβολή ερωτημάτων σε μια RDF βάση δεδομένων.

Προκειμένου να γίνει η φόρτωση και η διαχείριση των RDF δεδομένων ήταν απαραίτητο να χρησιμοποιηθεί ένας εξυπηρετητής δικτύου (web server). Στην συγκεκριμένη εφαρμογή χρησιμοποιήθηκε ένας open source web server, o Apache Tomcat. Για την υποστήριξη της JAVA, επιλέχτηκε ένα open source Semantic Web Framework(πλαίσιο ανοικτού Κώδικα Σημασιολογικού Ιστού), το Apache Jena. Αυτό, προκειμένου να κάνει την εξαγωγή δεδομένων και εγγραφή σε γραφήματα RDF, παρέχει ένα API, το οποίο τρέχει όταν έχουν οριστεί οι κατάλληλες προδιαγραφές. Για να συμβεί αυτό, δημιουργήθηκε ένα configuration file (config.properties) στο οποίο έχουμε ορίσει διάφορες παραμέτρους του συστήματος, καθώς και λεπτομέρειες που αφορούν τα δεδομένα, όπως τις κατηγορίες οντοτήτων που μας ενδιαφέρει να εμφανίζονται στην εφαρμογή.

Επιπλέον, στα πλαίσια της εργασίας χρησιμοποιήθηκαν βιβλιοθήκες JavaScript, όπως AG Grid και Chart JS, για την δημιουργία πινάκων και διαγραμμάτων που αφορούσαν τα δεδομένα που χρειάζονταν να διαχειριστούν.

Για την λειτουργία της εφαρμογής, χρησιμοποιήθηκαν δεδομένα του έργου SeaLit(στον τομέα της Ναυτικής Ιστορίας), τα οποία αφορούν την διαδικασία μετάβασης από τα ιστιοπλοϊκά στα ατμόπλοια, κατά τον 19^ο αιώνα.

Οι βασικές λειτουργίες που υποστηρίζει η εφαρμογή είναι:

- Επισκόπηση κατηγοριών οντοτήτων ενδιαφέροντος
- Προβολή όλων των οντοτήτων μιας κατηγορίας σε μορφή πίνακα
- Ταξινόμηση δεδομένων πίνακα ανά στήλη
- Φίλτραρισμα οντοτήτων πίνακα ανά στήλη
- Σελιδοποίηση πίνακα
- Προβολή διαγράμματος με βάση ένα χαρακτηριστικό των οντοτήτων
- Σελίδα οντότητας (με περισσότερες πληροφορίες για την οντότητα)

Τα κεφάλαια που θα παρουσιαστούν παρακάτω είναι τα εξής:

- Κεφάλαιο 2: Υπόβαθρο και Πλαίσιο Εργασίας
- Κεφάλαιο 3: Εφαρμογή πλοήγησης σε RDF δεδομένα
- Κεφάλαιο 4: Μοντέλο διαμόρφωσης
- Κεφάλαιο 5: Εγκατάσταση συστήματος
- Κεφάλαιο 6: Απαιτήσεις συστήματος και Απόδοση
- Κεφάλαιο 7: Τελικά Συμπεράσματα

2. Υπόβαθρο και Πλαίσιο Εργασίας

2.1. Το μοντέλο δεδομένων RDF

Για το μοντέλο δεδομένων RDF, μας ενδιαφέρει επιπλέον να γνωρίζουμε τις έννοιες των URI και SPARQL.¹

2.1.1 RDF

Το RDF (Resource Description Framework) αποτελεί ένα μοντέλο δεδομένων για το διαδίκτυο. Χρησιμοποιείται για την αναπαράσταση πληροφοριών που αφορούν πηγές του διαδικτύου, χωρίς να είναι απαραίτητο ότι τις παίρνουμε απευθείας από αυτό (Μπορεί να υπάρχουν σε ένα βιβλίο, ή να αναφέρονται από κάποιον άνθρωπο).

Το RDF προορίζεται για καταστάσεις όπου οι πληροφορίες υποβάλλονται σε επεξεργασία από εφαρμογές. Αυτές οι εφαρμογές θα δώσουν το τελικό αποτέλεσμα που θα εμφανιστεί στον χρήστη.

Υπάρχουν διάφορες μορφές RDF δεδομένων:

- TRIG
- RDF/XML
- N-Triples
- N-Quads
- Turtle

Αυτή η μέθοδος χρησιμοποιείται για την καλύτερη σύνδεση, εύρεση και κοινοποίηση δεδομένων. Ουσιαστικά υποδηλώνει τον Σημασιολογικό Ιστό (Semantic Web/Network).

Ο βασικός τρόπος αναπαράστασης των δεδομένων είναι triples(τριπλέτες) (subject-predicate-object)(Υποκείμενο-Κατηγόρημα-Αντικείμενο), όπου:

- Υποκείμενο: αποτελεί το «στοιχείο» που ενεργεί
- Κατηγόρημα: η ιδιότητα του υποκειμένου που μας ενδιαφέρει
- Αντικείμενο: Η τιμή ή η παραπομπή σε άλλο υποκείμενο(για να βρεθεί η τιμή) στην οποία γίνεται η ενέργεια του κατηγορήματος

2.1.2. URI

Το URI (Uniform Resource Identifiers) αποτελεί ένα σύνολο χαρακτήρων που ορίζουν ένα όνομα ή μια αναφορά του διαδικτύου.

Τύποι URI είναι:

- URL (Uniform Recource Locator): Υποδηλώνει την τοποθεσία μιας αναφοράς στο διαδίκτυο. (Συνήθως ξεκινάει με "https://")
- URN (Uniform Resource Name): Αναφέρεται σε ένα όνομα, χωρίς την απαραίτητη χρήση μιας τοποθεσίας (όπως του URL)

¹ Οι έννοιες πάρθηκαν από το μάθημα «HY-561 Διαχείριση Δεδομένων στον Παγκόσμιο Ιστό», elearn-uoc, <https://elearn.uoc.gr/user/view.php?id=13674&course=4510>

2.1.3. SPARQL

Η SPARQL (Structured Protocol and RDF Query Language) αποτελεί γλώσσα ερωτημάτων για την αναζήτηση και χρήση δεδομένων που είναι αποθηκευμένα σε RDF μορφή.

Η κύρια χρήση της είναι η αναζήτηση δεδομένων RDF, εξαγωγή πληροφοριών και τρόπο σύνδεσης μεταξύ τους.

Η δομή που φτιάχνεται ένα SPARQL ερώτημα είναι:

- SELECT: επιλέγει τα στοιχεία που θέλουμε να εμφανίσει μέσα από το RDF αρχείο
- WHERE: Καθορίζει τις συνθήκες που θα βρεθεί η ιδιότητα του στοιχείου που θέλουμε

Όπως αναφέρθηκε, τα RDF αρχεία αναπαρίστανται σαν τριπλέτες.

Έτσι και το πρότυπο αναζήτησης τους από το query, θα είναι αντίστοιχο.

Επιτρέπεται να χρησιμοποιηθούν συνθήκες (REPLACE) , φίλτρα (FILTER) , περιορισμοί(MAX), συγκεντρωτικές λειτουργίες (COUNT, AVG) για την βελτίωση των αποτελεσμάτων που θα επιστρέψει το ερώτημα.

2.2 Το έργο SeaLit και τα δεδομένα του

Το έργο SeaLit² αποτελεί ένα project συνεργασίας ερευνητών της Μεσογείου (ICS-FORTH-Hellas, University of Barcelona, University of Genoa, University of Aix-Marseille and the Centre for Cultural Informatics/ICS/FORTH) χρηματοδοτούμενο από το ERC Starting Grant 2016, που χρησιμοποιείται στον τομέα της Ναυτικής Ιστορίας. Το πλήρες όνομα του έργου είναι Seafaring Lives in Transition, Mediterranean Maritime Labour and Shipping, 1850s-1920s, αφού η μελέτη αφορά την διαδικασία μετάβασης από τα ιστιοπλοϊκά στα ατμόπλοια και τον τρόπο με τον οποίο οι ναυτικοί αντέδρασαν σε αυτήν την αλλαγή στις περιοχές της Μεσογείου και της Μαύρης Θάλασσας, κατά το χρονικό διάστημα 1850 και 1920(19^{ος} αιώνας).

Αυτή η εποχή ήταν μια σημαντική αλλαγή για την ναυτική κοινότητα, αφού έκτοτε τα ατμόπλοια αποτέλεσαν τον κυρίαρχο τρόπο μεταφοράς στην θάλασσα.

Τα δεδομένα του έργου έχουν αντληθεί από διάφορες πηγές, 16 διαφορετικών τύπων αρχείων, όπως ημερολόγια των πλοίων, βιβλία με διαδρομές, σημειώσεις, αφίξεις και αναχωρήσεις των πλοίων από κάθε λιμάνι, λίστες πληρωμάτων με ατομικές πληροφορίες μεταξύ αυτών όπως αναγνωριστικά, ονοματεπώνυμα, πληρωμές-μισθοδοσίες.

Κάποια από τα σημαντικότερα στοιχεία που χρησιμοποιήθηκαν στο παρόν project είναι:

- Πρόσωπα (Persons)
- Τοποθεσίες (Locations)
- Ταξίδια(Voyages)
- Πλοία (Ships)
- Ημερομηνίες (Dates)
- Χρονικά διαστήματα (Time-Spans)
- Αναγνωριστικά (Identifiers)

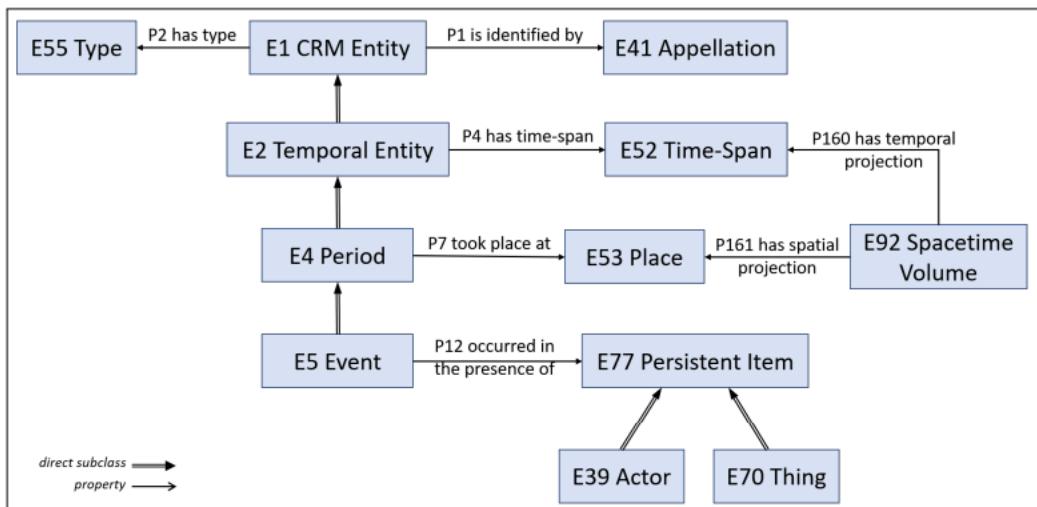
² Επίσημη σελίδα δεδομένων SeaLit, <https://www.ics.forth.gr/isl/fast-cat#usecase>

2.2.1 Οντολογία των δεδομένων SeaLit

Όπως έχει ήδη αναφερθεί τα δεδομένα SeaLit, αποτελούν ένα μοντέλο δεδομένων RDF. Η οντολογία λοιπόν που τα περιγράφει έχει στόχο τη διευκόλυνση της ανταλλαγής, ενσωμάτωσης και διαμεσολάβησης ανόμοιων πληροφοριών που σχετίζονται με τη ναυτική ιστορία. Έτσι προσφέρει σημασιολογικούς ορισμούς για τον συνδυασμό διαφορετικών πηγών των δεδομένων. Με αυτό τον τρόπο, δημιουργείται μια κοινή γλώσσα για όσους ειδικεύονται στον τομέα, επιτυγχάνοντας μια συνεργασία μεταξύ τους.

Η οντολογία χρησιμοποιεί μια επέκταση του προτύπου CIDOC Conceptual Reference Model (ISO 21127:2014) version 7.2.1, ως μια γενική οντολογία της ανθρώπινης δραστηριότητας, πραγμάτων και γεγονότων που συμβαίνουν στο χώρο και στο χρόνο. Η επιλογή του συγκεκριμένου προτύπου έγινε καθώς αποτελεί μια τυπική οντολογία για την τεκμηρίωση της πολιτιστικής κληρονομιάς, που χρησιμοποιείται ευρέως στους τομείς της πολιτιστικής κληρονομιάς, της ιστορίας και της αρχαιολογίας. Ένας ακόμα λόγος χρήσης του είναι η διευκόλυνση που παρέχει για την ενοποίηση δεδομένων με ήδη παρόμοια σύνολα, αλλά και με μελλοντικά. Η ανάπτυξη της ακολούθησε μια από κάτω προς τα πάνω διαδικασία από πρωτογενή δεδομένα που συλλέχθηκαν σε αυτό το έργο.

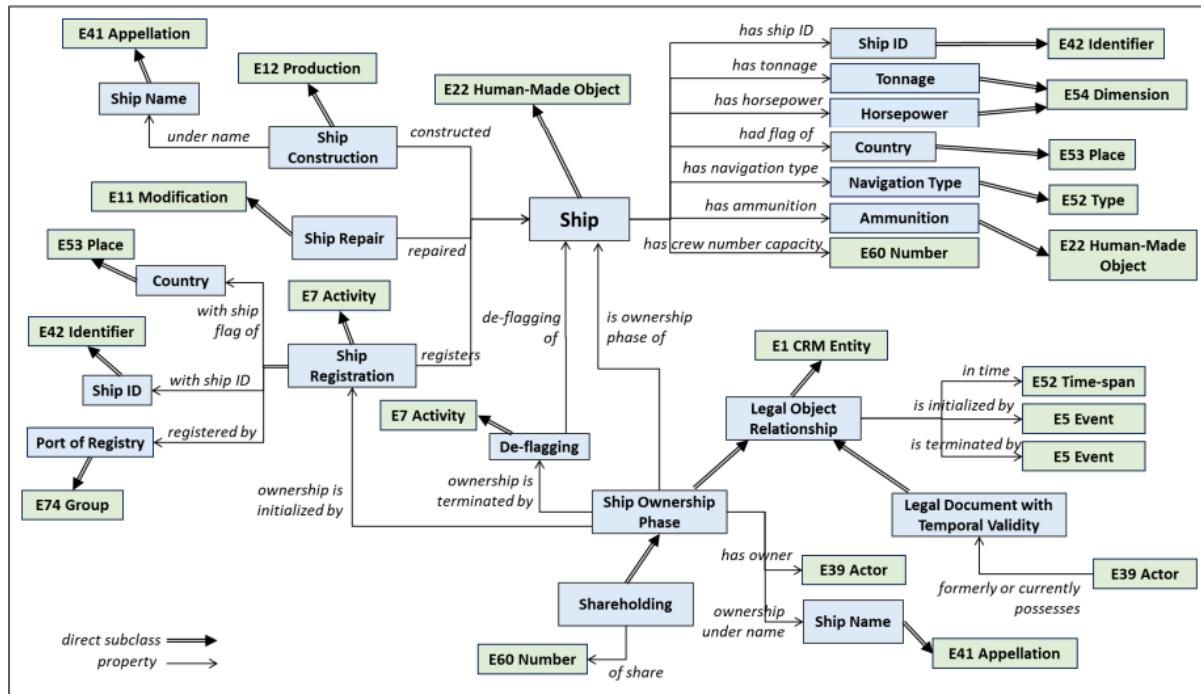
Βασικές έννοιες που χρησιμοποιεί το CIDOC CRM, συνδέονται όπως φαίνονται στην παρακάτω εικόνα³.



Εικόνα 1: Υψηλού επιπέδου ιδιότητες και κλάσεις του προτύπου CIDOC-CRM

³ Πηγή εικόνας: P. Fafalios, A. Kritsotaki, M. Doerr, "The SeaLit Ontology - An Extension of CIDOC-CRM for the Modeling and Integration of Maritime History Information", ACM Journal on Computing and Cultural Heritage, 2023. <https://doi.org/10.1145/3586080>

Η παρακάτω εικόνα δείχνει ένα μέρος της οντολογίας SeaLiT το οποίο αφορά τον τρόπο που μοντελοποιούνται πληροφορίες για ένα πλοίο, όπως η κατασκευή του, τα χαρακτηριστικά του, κτλ.⁴



Εικόνα 2: Παράδειγμα μοντελοποίησης πληροφοριών ενός πλοίου

Στην περίπτωση μας, η μελέτη του μοντέλου που χρησιμοποιούν τα δεδομένα είναι σημαντική για να μπορούμε να ορίσουμε τα ερωτήματα στο configuration file, τα οποία θα μας δώσουν τα αποτελέσματα που χρειαζόμαστε.

Η χρήση του CIDOC-CRM για το SeaLit project, παρέχει ένα δομημένο και ξεκάθαρο τρόπο ολοκλήρωσης, περιγραφής και αναζήτησης των δεδομένων του έργου. Εφόσον, γίνει κατανοητή η οντολογία και ο τρόπος που περιγράφονται οι οντότητες και οι σχέσεις τους, επιτυγχάνεται η βέλτιστη ενοποίηση και ανάκτηση των δεδομένων, επιτρέποντας την ανάλυση τους.

⁴ Πηγή εικόνας: P. Fafalios, A. Kritsotaki, M. Doerr, "The SeaLiT Ontology - An Extension of CIDOC-CRM for the Modeling and Integration of Maritime History Information", ACM Journal on Computing and Cultural Heritage, 2023. <https://doi.org/10.1145/3586080>

3. Εφαρμογή Πλοήγησης σε RDF δεδομένα

3.1. Αρχιτεκτονική και Βασικές λειτουργίες συστήματος

3.1.1. Ανάγνωση RDF αρχείων και Φόρτωση γράφου στη μνήμη

Η ανάγνωση RDF αρχείων και φόρτωση γράφου στην μνήμη, επιτρέπει την αποτελεσματική επεξεργασία και αναζήτηση δεδομένων.

Με την χρήση του framework Apache Jena, γίνεται η αποθήκευση και διαχείριση των RDF δεδομένων. Αυτό που γίνεται από το σύστημα είναι η ανάλυση των αρχείων RDF για εξαγωγή των δεδομένων και ύστερα η φόρτωση τους στην μνήμη για γρήγορη πρόσβαση.

Τα δεδομένα που χρησιμοποιήθηκαν στην παρούσα εφαρμογή αφορούσαν πληροφορίες από την Ναυτική Ιστορία και έχουν εξαχθεί από λίστες πληρωμάτων. Συγκεκριμένα υπάρχει ένα έγγραφο τεσσάρων σελίδων που παραδίδεται από τις λιμενικές αρχές που βρίσκονται στο λιμάνι νηολόγησης ενός πλοίου και αφορούν καταλόγους πληρωμάτων που φυλάσσονται από το κράτος. Αυτά τα έγγραφα, εκδόθηκαν κατά την αναχώρηση του πλοίου από το λιμάνι μιας χώρας, τη στιγμή της εγγραφής του πληρώματος, και κάλυπταν περιόδους συνεχούς ναυσιπλοΐας (μέγιστης περιόδου δύο χρόνων).

Στο εσωτερικό, οι λίστες πληρωμάτων περιέχουν πληροφορίες:

- για το πλοίο,
- τους πλοιοκτήτες και
- τα μέλη του πληρώματος,
- συμπεριλαμβανομένων προσωπικών και επαγγελματικών στοιχείων.

Στην τελευταία σελίδα υπήρχε συνήθως ένας κατάλογος των λιμανιών, αποτελούμενος από προξενικές σφραγίδες και δηλώσεις που χρειαζόταν κάθε καπετάνιος πριν φύγει από λιμάνι, ανεξάρτητα από το αν σταματούσε εκεί από ανάγκη (για υγειονομικούς και τελωνειακούς ελέγχους) ή για εμπορικές εργασίες. Παρέχουν στοιχεία ταυτότητας πλοίου:

- όνομα,
- χωρητικότητα,
- τύπος,
- τόπος και ημερομηνία κατασκευής.

Παρέχουν επίσης πολύτιμες πληροφορίες για τους ναυτικούς:

- ημερομηνία επιβίβασης,
- ημερομηνία απόρριψης,
- επώνυμο,
- όνομα,
- ημερομηνία και τόπος γέννησης,
- κατάταξη στο πλοίο,
- αύξων αριθμός και
- μισθοί.

Αυτά τα στοιχεία αποτελούσαν τα κύρια δεδομένα, από τα οποία συλλέχτηκαν πληροφορίες, για την διαμόρφωση της διαδικτυακής πλατφόρμας περιήγησης και εξερεύνησης.

3.1.2. Επισκόπηση έργου και βασικών οντοτήτων

Κατά την είσοδο στην διαδικυακή εφαρμογή, υπάρχει:

- Μια περιγραφή του έργου, για το οποίο παρουσιάζονται τα δεδομένα
- Μια επισκόπηση των κατηγοριών και του αριθμού οντοτήτων τους, για τα δεδομένα του έργου

ΛΕΙΤΟΥΡΓΙΕΣ ΠΟΥ ΥΠΟΣΤΗΡΙΖΕΙ Ο ΠΙΝΑΚΑΣ

Για την παρουσίαση των δεδομένων κάθε κατηγορίας, έχει δημιουργηθεί ένας πίνακας που εμφανίζει, ανάλογα την περίπτωση, στήλες με βασικά δεδομένα που ενδιαφέρουν τον χρήστη με την πρώτη ματιά.

Ο πίνακας υποστηρίζει τις εξής λειτουργίες:

3.1.3. Σελιδοποίηση

Επειδή ο όγκος των δεδομένων είναι μεγάλος, γίνεται σελιδοποίηση των αποτελεσμάτων, πετυχαίνοντας ταυτόχρονα καλύτερη διαχείριση τους(ανάγνωση) και ευκολότερη πλοήγηση στα αποτελέσματα.

Επιπλέον υπάρχει η δυνατότητα τροποποίησης του αριθμού των απεικονιζόμενων γραμμών του πίνακα.

(Γίνεται κατανοητό παρακάτω στην ενότητα 3.2 Διεπαφή χρήστη)

3.1.4. Ταξινόμηση

Για κάθε στήλη του πίνακα, έχει βρεθεί ο τύπος των περιεχομένων που περιλαμβάνει. Σε εκείνες που περιλαμβάνουν:

- **Strings(χαρακτήρες):**

- Default αναπαράσταση: τυχαία(συγκεκριμένα είναι με την σειρά που βρέθηκαν στα αρχεία που αναζητήθηκαν)
- Γίνεται αλφαριθμητική ταξινόμηση ($A \rightarrow Z$ or $Z \rightarrow A$)

Destination	Count
Azof	1
Azoff	2
Azoff	1
Azov	2
Bari	1
Civitavecchia	1
Danubio	3

Εικόνα 4: Ταξινόμηση A-Z

Destination	Count
Tiro (city)/ Libano (coun...	1
Smirne	1
Salonicco	1
Pozzallo	1
Odessa	1
Nord Europa	1
Nizza	1
Napoli	1
Mola di Gaeta	1
Messina e Odessa	1

Εικόνα 3: Ταξινόμηση Z-A

- **Αριθμούς:**

- Default αναπαράσταση: τυχαία (συγκεκριμένα είναι με την σειρά που βρέθηκαν στα αρχεία που αναζητήθηκαν)
- Γίνεται Αύξουσα ($1 \rightarrow 9$) ή Φθίνουσα ($9 \rightarrow 1$) ταξινόμηση

The screenshot shows a web-based catalog interface. At the top, there's a blue header bar with 'Main' and 'About' links. Below it, a banner says 'Welcome to FastCat Catalogues' over a background of a 3D hexagonal grid. On the left, a sidebar lists categories: Persons (1910), Locations (442), Voyages (50), Ships (98), Span (1727), and Identifier (985). A green arrow points from the text 'Τυχαία Ταξινόμηση' to the 'Identifier' button. The main area has a dark blue header 'Identifier'. It contains a table with columns: Label, URI, and More. The 'Label' column is highlighted with an orange border. The table lists 10 entries, each with a unique ID and its corresponding URI starting with <https://rs.sealitproject.e...>. At the bottom of the table, there are pagination controls: 'Page Size: 100', '1 to 100 of 985', and navigation arrows.

Label	URI	More
1045330	https://rs.sealitproject.e...	More info
3710	https://rs.sealitproject.e...	More info
12385	https://rs.sealitproject.e...	More info
9882	https://rs.sealitproject.e...	More info
12106	https://rs.sealitproject.e...	More info
93	https://rs.sealitproject.e...	More info
3519	https://rs.sealitproject.e...	More info
1076	https://rs.sealitproject.e...	More info
9112	https://rs.sealitproject.e...	More info
4174	https://rs.sealitproject.e...	More info

Εικόνα 5: Τυχαία ταξινόμηση ενός column

- **Ημερομηνίες:**
 - Default αναπαράσταση: τυχαία(συγκεκριμένα είναι με την σειρά που βρέθηκαν στα αρχεία που αναζητήθηκαν)
 - Γίνεται Αύξουσα ή Φθίνουσα ημερολογιακή ταξινόμηση

The screenshot shows a digital catalog interface titled "Catalogues". A search bar at the top contains the text "Date ↑". Below the search bar is a table with two columns: "Date" and "More info". The "Date" column lists dates from 1864-01-17 to 1864-07-18. The "More info" column contains links for each date entry. At the bottom of the table, there is a page navigation bar with the text "801 to 900 of 922" and arrows for navigating through the pages.

Date ↑	More
1864-01-17	More info
1864-01-23	More info
1864-01-23	More info
1864-02-03	More info
1864-02-10	More info
1864-02-21	More info
1864-03-18	More info
1864-05-30	More info
1864-06-08	More info
1864-07-18	More info
801 to 900 of 922	K < Page 9 of 10 > K

Εικόνα 6: Αύξουσα Ταξινόμηση

The screenshot shows a digital catalog interface titled "Catalogues". A search bar at the top contains the text "Date ↓". Below the search bar is a table with two columns: "Date" and "More info". The "Date" column lists dates from 1861-07-18 to 1861-06-22. The "More info" column contains links for each date entry. At the bottom of the table, there is a page navigation bar with the text "401 to 500 of 922" and arrows for navigating through the pages.

Date ↓	More
1861-07-18	More info
1861-07-17	More info
1861-07-15	More info
1861-07-11	More info
1861-07-05	More info
1861-07-03	More info
1861-07-01	More info
1861-06-28	More info
1861-06-23	More info
1861-06-22	More info
401 to 500 of 922	K < Page 5 of 10 > K

Εικόνα 7: Φθίνουσα Ταξινόμηση

Με κλικ πάνω στο column που μας ενδιαφέρει γίνεται μια από τις τρεις ταξινομήσεις κάθε φορά.

(*Για τις απαιτήσεις των παραδειγμάτων, προστέθηκαν στην εφαρμογή δυο επιπλέον κατηγορίες που περιλαμβάνουν ημερομηνίες και αριθμητικά δεδομένα.)

3.1.5. Φίλτρα

Στον πίνακα όπου εμφανίζονται τα δεδομένα κάθε κατηγορίας, για κάθε στήλη(ετικέτα-label) υπάρχουν φίλτρα.

Αυτά τα φίλτρα είναι από τις πιο χρήσιμες λειτουργίες της εφαρμογής, καθώς οι χρήστες μπορούν με άμεσο και γρήγορο τρόπο να βρουν οποιαδήποτε πληροφορία χρειάζονται.

Το φίλτρο που θα έχει κάθε στήλη, προκύπτει από το περιεχόμενο τους.

Για κάθε περίπτωση έχει δημιουργηθεί ένα συγκεκριμένο template, όπου:

- Αν περιλαμβάνουν κείμενο, τότε ο χρήστης μπορεί με αλφαριθμητικούς χαρακτήρες να δει:
 - Contains: Αν υπάρχει ή όχι, κάτι που να ταιριάζει με αυτό που πληκτρολογεί
 - Equals: Αν υπάρχει ή όχι, μια ολόκληρη η λέξη που αναζητεί
 - Begins/Ends with: Αν ξεκινάει ή τελειώνει με τους χαρακτήρες που έβαλε
 - Blank: Τις καταχωρήσεις της στήλης που είναι ή δεν είναι κενές

Main About

Welcome to FastCat C

Persons

First Name Last Name

Begins with ▾

AND OR

Contains ▾

Filter...

First Name	Last Name
Carolina	
Carlo	
Costanzo	
Clemente Luigi	
Cesare	Figari
Cesare	
Carlo	Antola
Cosimo Domenico	Faleo
Cesare	Figari
Capurro	Fortunato

Page Size: 100

Εικόνα 8: Αναζήτηση αποτελεσμάτων που "ξεκινούν με"

Main About

Welcome to FastCat C

Persons

First Name Last Name

Blank ▾

AND OR

Contains ▾

Filter...

First Name	Last Name
	Denegri

Page Size: 100

Εικόνα 9: Εμφάνιση αποτελεσμάτων με κενή τιμή καταχώρησης

(*Δεν υπάρχει case-sensitivity στο search bar και ο χρήστης θα πάρει αποτέλεσμα είτε γράψει με κεφαλαία, είτε με μικρά γράμματα)

- Αν περιλαμβάνουν ψηφία, τότε ο χρήστης μπορεί να ψάξει με βάση κάποιο αριθμητικό χαρακτήρα να δει:
 - Equals: Αν κάτι είναι ίσο ή όχι, με τον αριθμό που πληκτρολόγησε
 - Greater/Less than: Αριθμούς μεγαλύτερους ή μικρότερους από αυτόν που καταχώρησε
 - Greater/Less than or equal to: Αριθμούς μεγαλύτερους/μικρότερους ή ίσους από εκείνο που καταχώρησε
 - Between: Αριθμούς μεταξύ των αριθμών που έβαλε(εδώ εμφανίζονται 2 πεδία που θα εισάγει τιμή ο χρήστης)
 - Blank: Τις καταχωρήσεις της στήλης που είναι ή δεν είναι κενές

Main About

Welcome to FastCat C

Persons (1910)
Locations (442)
Voyages (50)
Ships (98)
Span (1727)
Identifier (985)

Identifier

Label = URI
1045330 Greater than or equal... ▾
12385
9882
12106
9112
12608
18189
11467
6295
12660

Equals ▾
Filter...

https://rs.sealitproject.e...

Page Size: 100

Όλοι οι αριθμοί "μεγαλύτεροι ή ίσοι"

Εικόνα 11: Αναζήτηση αριθμούς "μεγαλύτερων ή ίσων από"

Main About

Welcome to FastCat C

Persons (1910)
Locations (442)
Voyages (50)
Ships (98)
Span (1727)
Identifier (985)

Identifier

Label = URI
3897 Between ▾
3899
3889
3894
3893
3826

Equals ▾
Filter...

https://rs.sealitproject.e...

Όλοι οι αριθμοί που βρίσκονται σε ένα εύρος τιμών

Τα αποτελέσματα προκύπτουν από δύο πεδία

Page Size: 100

Εικόνα 10: Αναζήτηση αριθμών ενός εύρους τιμών

- Αν περιλαμβάνουν ημερομηνίες, τότε ο χρήστης μπορεί να δει:
 - Equals: Αν περιλαμβάνετε ή όχι, μια ακριβής ημερομηνία
 - Before/After: Τις ημερομηνίες πριν ή μετά από εκείνη που καταχώρησε
 - Between: Τις ημερομηνίες μεταξύ ενός διαστήματος(εδώ εμφανίζονται 2 πιεδία που θα εισάγει τιμή ο χρήστης)
 - Blank: Τις καταχωρήσεις της στήλης που είναι ή δεν είναι κενές

Eikόνα 13: Αναζήτηση ακριβής ημερομηνίας

Eikόνα 12: Αναζήτηση ημερομηνιών "μετά από την"

Μεταξύ των διαφόρων φίλτρων, είναι δυνατόν το φιλτράρισμα να γίνεται ταυτοχρόνως με 2 παραμέτρους, τις οποίες μπορεί ο χρήστης να διαλέξει αν το αποτέλεσμα θα το πάρει με την χρήση:

- AND, όπου πρέπει και τα δυο φίλτρα να είναι αληθή
- OR, όπου αρκεί το ένα από τα δυο φίλτρα να είναι αληθή.

Main About

Welcome to FastCat C

Persons (1910)
Locations (442)
Voyages (50)
Ships (98)
Span (1727)
Identifier (985)

Voyages

Destination Label

Mar Nero AND OR Contains

Nord Europa
Danubio
Messina e Odessa
Mar Nero
Danubio
Mar Nero
Danubio (region) / Roma...
Mar Nero

Voyage of ship Dittatore...
Voyage of ship Industria
Voyage of ship Ider
Voyage of ship Antonietta
Voyage of ship Dante

Page Size: 100

Εικόνα 15: Ταξινόμηση που αρκεί να ισχύει "ένα από τα δύο" φίλτρα

Catalogues

Date More

After AND OR Before

1863-02-09
1862-11-19
1862-11-19
1862-11-19
1863-01-16
1863-01-16
1863-02-18
1863-02-18
1862-11-20

More info
More info

1 to 38 of 38 | Page 1 of 1

Εικόνα 14: Ταξινόμηση που πρέπει να ισχύουν "και τα δύο" φίλτρα

3.1.6. Διαγράμματα(Charts)

Τα διαγράμματα προκύπτουν από τα δεδομένα κάθε στήλης και σχετίζονται με τον αριθμό εμφάνισης των δεδομένων που περιέχουν.

Ανάλογα την περίπτωση, μπορούν να περιέχουν σημαντική πληροφορία στον χρήστη, επιλέγοντας να εμφανίσουν το διάγραμμα μιας ή περισσότερων στήλων κάποιας οντότητας στην οποία βρίσκεται εκείνη την στιγμή.

(Γίνεται κατανοητό παρακάτω στην ενότητα 3.2. Διεπαφή Χρήστη).

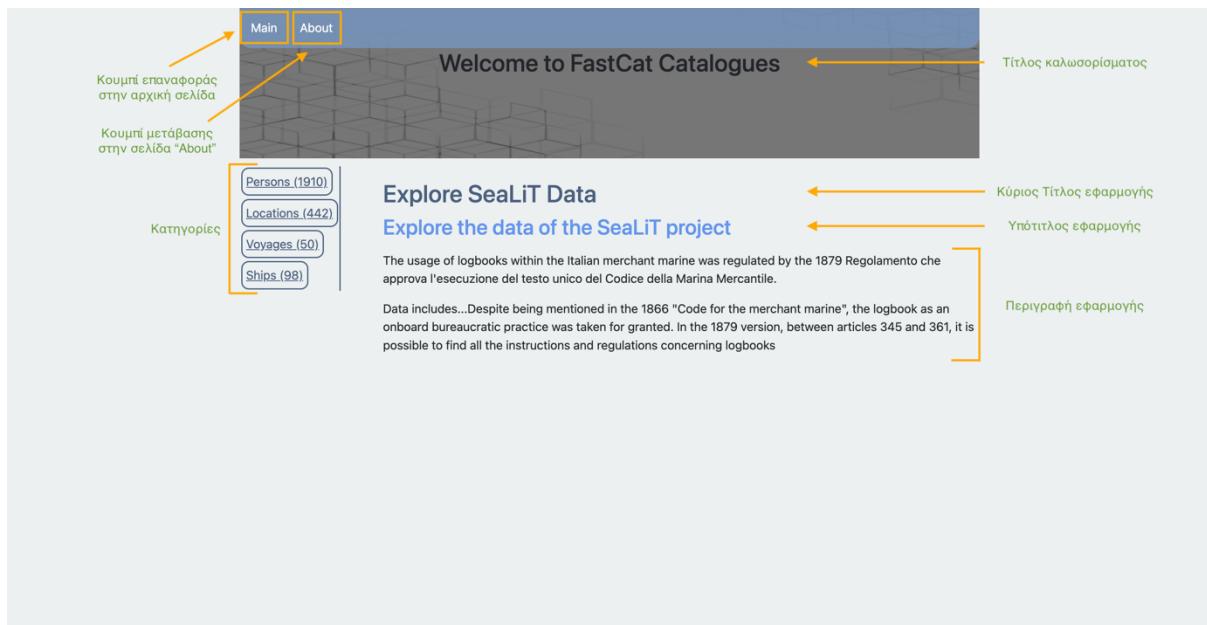
3.1.7. Προβολή λεπτομερειών μιας καταχώρησης

Όταν ο χρήστης ενδιαφέρεται να μάθει περαιτέρω πληροφορίες για μια συγκεκριμένη καταχώρηση, μπορεί να πατήσει το link που βρίσκεται στην τελευταία στήλη δεξιά του πίνακα.

Το σύστημα εκεί, του παρουσιάζει μια σελίδα με περισσότερες λεπτομέρειες για εκείνη την καταχώρηση, οι οποίες εξαρτώνται ανάλογα με την κατηγορία.
(Γίνεται κατανοητό παρακάτω στην ενότητα 3.2. Διεπαφή Χρήστη).

3.2. Διεπαφή χρήστη (User Interface)

Κατά την είσοδο στην εφαρμογή βλέπουμε την αρχική σελίδα⁵:



Εικόνα 16: Αρχική σελίδα εφαρμογής

Υπάρχουν:

- Κουμπί επαναφορά στην αρχική σελίδα
- Κουμπί μετάβασης στην σελίδα "About"
- Κατηγορίες εφαρμογής
- Τίτλος Καλωσορίσματος
- Κύριος Τίτλος Εφαρμογής
- Υπότιτλος Εφαρμογής
- Περιγραφή Εφαρμογής

⁵ Use of Bootstrap: Open source icon library for design of the page, <https://getbootstrap.com/docs/5.3/getting-started/introduction/>

Αφού επιλεχθεί μια κατηγορία εμφανίζεται η παρακάτω σελίδα:

Nέο page για την εμφάνιση κατηγοριών

Παράμετρος που εμφανίζει την κατηγορία που έχει επιλεχθεί

Όνομα επιλεγμένης κατηγορίας

Επισήμανση με διαφορετικό χρώμα ενεργής κατηγορίας

Columns με βασικές πληροφορίες της κατηγορίας που επιλέχθηκε

Κουμπί τροποποίησης του αριθμού των rows που φαίνονται στον πίνακα

Κουμπί εμφάνισης διαγραμμάτων των columns

Charts: □ Type □ Label □ URI

Main About

Welcome to FastCat Catalogues

Ships

Type	Label	URI	More
master	ship Aurora	More info	
brigantino	ship Aurora	More info	
master	ship Antonio	More info	
brigantino	ship Antonio	More info	
brigantino	ship Emilia	More info	
master	ship La Concezione	More info	
brigantino	ship La Concezione	More info	
master	ship Emilia	More info	
brigantino	ship Emilia	More info	
brigantino	ship Andrea	More info	

Page Size: 100 ▾

20
50
100

1 to 100 of 146

< < Page 1 of 2 > >

Καταχωρήσεις(rows) του πίνακα

Πλοήγηση στις σελίδες του πίνακα

Κουμπί μετάβασης σε περισσότερες λεπτομέρειες μιας καταχώρησης

Εικόνα 17: Σελίδα εμφάνισης μιας κατηγορίας

Υπάρχουν:

- Στο URL:
 - grid.jsp: Το νέο page που δημιουργήθηκε για την αναπαράσταση των δεδομένων κάθε κατηγορίας
 - obj: Αποτελεί την παράμετρο που αναδεικνύει το όνομα της κατηγορίας που έχει επιλεχθεί.
- Εδώ θα είναι μια από τις:
 - Persons
 - Locations
 - Voyages
 - Ships
- Δεξιά από τις κατηγορίες υπάρχει:
 - Όνομα επιλεγμένης κατηγορίας
 - Πίνακας δεδομένων επιλεγμένης κατηγορίας:
 - Στην πρώτη γραμμή εμφανίζονται τα ονόματα των βασικών columns
 - (Τα ονόματα προκύπτουν από αυτά που έχουν οριστεί μέσω των queries, στο configuration file)
 - Στο κάτω μέρος υπάρχουν οι λειτουργίες του:
 - Δυνατότητα αλλαγής του μεγέθους των γραμμών της σελίδας

2. Αριθμός γραμμών στην παρούσα σελίδα και των υπολειπόμενων
 3. Σελιδοποίηση
- Στην τελευταία στήλη του πίνακα υπάρχει κουμπί μετάβασης σε περισσότερες λεπτομέρειες για την συγκεκριμένη καταχώρηση

Στην κάτω μεριά της προηγούμενης εικόνας υπάρχουν τα διαγράμματα(Charts):



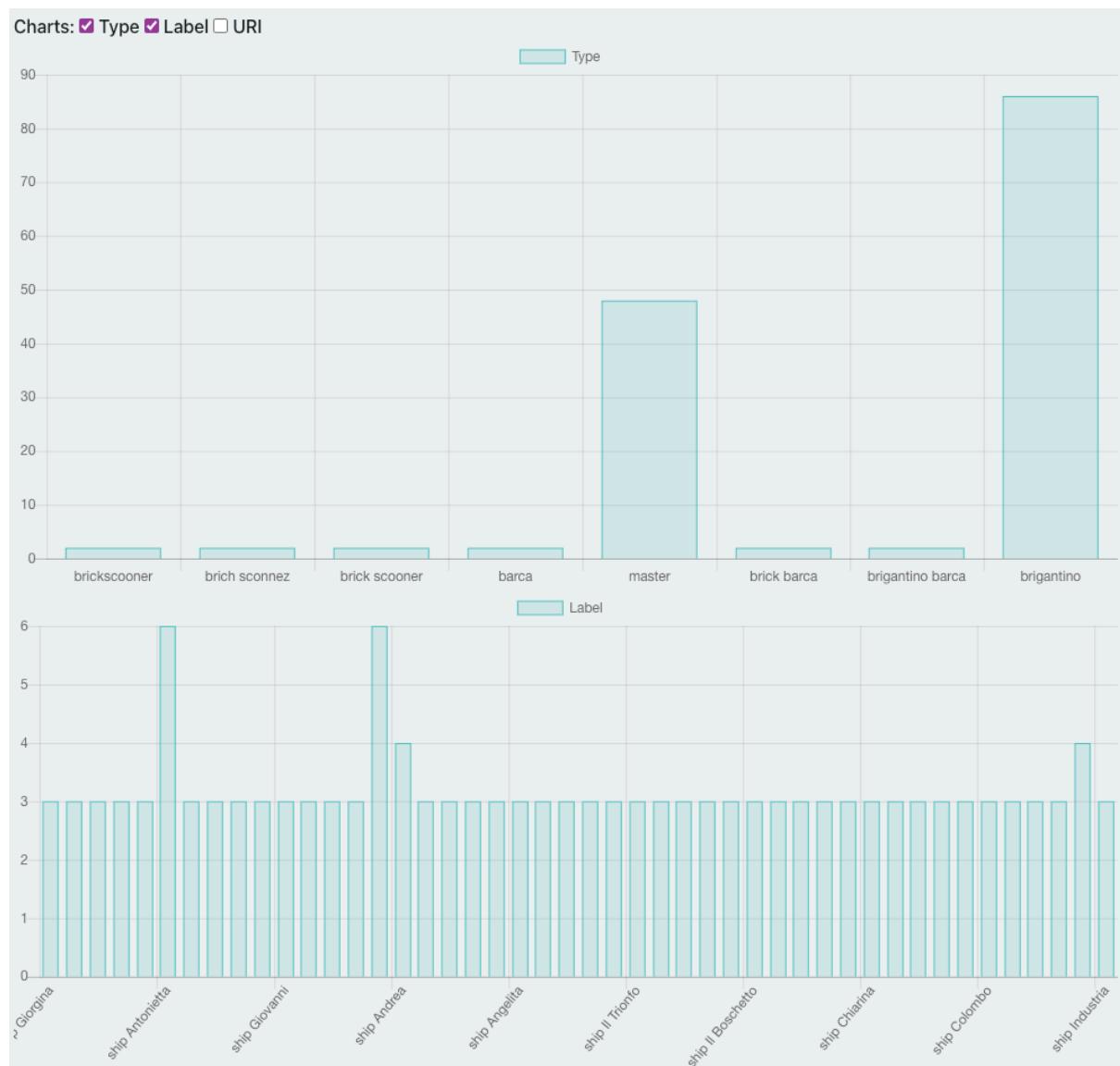
Σε αυτά ο χρήστης μπορεί να επιλέξει να δει το διάγραμμα που επιθυμεί.

Σε καθένα απ' αυτά υπάρχουν οι εξής πληροφορίες:

- Στον κάθετο άξονα(στα αριστερά), υπάρχει ο αριθμός εμφάνισης των στοιχείων της επιλεγμένης στήλης του πίνακα.
(Το εύρος των τιμών προκύπτει από τα δεδομένα)
- Ο οριζόντιος άξονας (στην κάτω μεριά), δείχνει τα στοιχεία του επιλεγμένου column
- Όταν ο κέρσορας πηγαίνει σε μια ομάδα στοιχείων του διαγράμματος, εμφανίζονται:
 - Το όνομα της ομάδας
 - Η ποσότητα εμφάνισης τους στον πίνακα

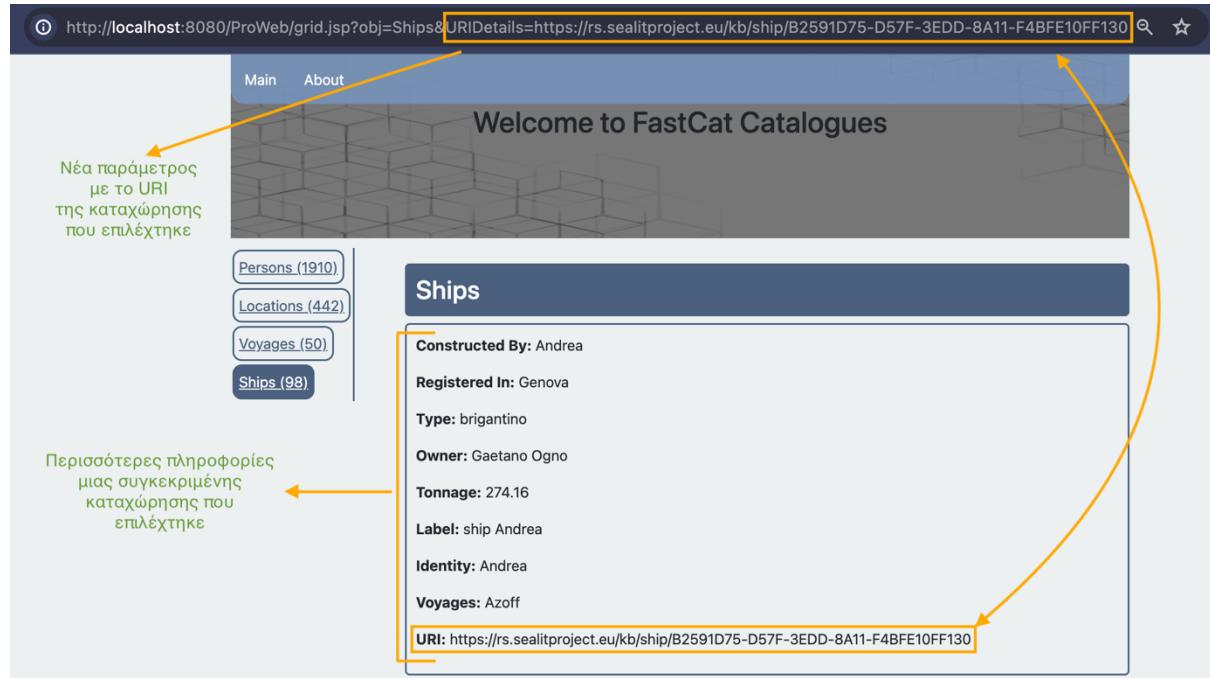
Η επιλογή του παραπάνω διαγράμματος έγινε επειδή αποτελεί ένα πολύ καλό παράδειγμα, μιας και δίνει χρήσιμα αποτελέσματα.

Επιπλέον μπορούν να εμφανιστούν και πολλαπλά διαγράμματα την ίδια στιγμή:



Εικόνα 19: Δυνατότητα εμφάνισης πολλαπλών διαγραμμάτων την ίδια στιγμή

Όταν πατήσουμε το “More info” link ενός record(row) στο δεξιό column του πίνακα κάθε κατηγορίας, μεταφερόμαστε σε νέα σελίδα του ίδιου jsp page, χρησιμοποιώντας την παράμετρο “URLDetails”, με (μοναδική) τιμή το URI της καταχώρησης που επιλέχτηκε.



Εικόνα 20: Σελίδα με περισσότερες πληροφορίες μιας καταχώρησης

Αναφέρθηκε προηγουμένως ότι το URI κάθε καταχώρησης είναι μοναδικό. Ωστόσο, μπορεί το ίδιο URI να καλείται πολλαπλές φορές σε ένα άλλο πεδίο. Όπως φαίνεται στο παρακάτω παράδειγμα, το Voyage του Ship: Indipendenza Peruviana με URI: urn:uuid:d15bd72d-f160-4d8c-9463-fc28059140b4, ήταν από διαφορετικά λιμάνια και πολλά άτομα είχαν κάνει αυτό το ταξίδι.

http://localhost:8080/ProWeb/grid.jsp?obj=Voyages&URI=urn:uuid:d15bd72d-f160-4d8c-9463-fc28059140b4

Voyages

Destination: Danubio

Carried By:

- Domenico Schiaffino
- Antonio Schiappacasse
- Benedetto Avegno
- Eugenio Marchese
- Lorenzo Olivari
- Giuseppe Schiappacasse
- Fortunato Dellacasa
- total crew number of Indipendenza Peruviana
- Michele Capurro
- Larzachi Giorgio Nicolati
- Bartolomeo Pellegrini
- Simone Figari
- Nicolo Schiappacasse
- Angelo Mortola

Leaving:

- Departure of ship Indipendenza Peruviana ,from port Sulina ,on date 1861-03-23
- Departure of ship Indipendenza Peruviana ,from port Braila ,on date 1861-05-03
- Departure of ship Indipendenza Peruviana ,from port Costantinopoli ,on date 1861-03-20
- Departure of ship Indipendenza Peruviana ,from port Cefalonia ,on date 1862-01-08
- Departure of ship Indipendenza Peruviana ,from port Costantinopoli ,on date 1861-10-26
- Departure of ship Indipendenza Peruviana ,from port Livorno ,on date 1862-02-07
- Departure of ship Indipendenza Peruviana ,from port Costantinopoli ,on date 1861-05-13
- Departure of ship Indipendenza Peruviana ,from port Gloucester ,on date 1861-08-05
- Departure of ship Indipendenza Peruviana ,from port Cardiff ,on date 1861-08-19
- Departure of ship Indipendenza Peruviana ,from port Sulina ,on date 1861-05-08
- Departure of ship Indipendenza Peruviana ,from port Longone ,on date 1861-02-21
- Departure of ship Indipendenza Peruviana ,from port Malta ,on date 1861-09-27
- Departure of ship Indipendenza Peruviana ,from port Odessa ,on date 1861-10-12
- Departure of ship Indipendenza Peruviana ,from port Costantinopoli ,on date 1861-10-09

Label: Voyage of ship Indipendenza Peruviana

Period: 1861-02-15 1862-02-13

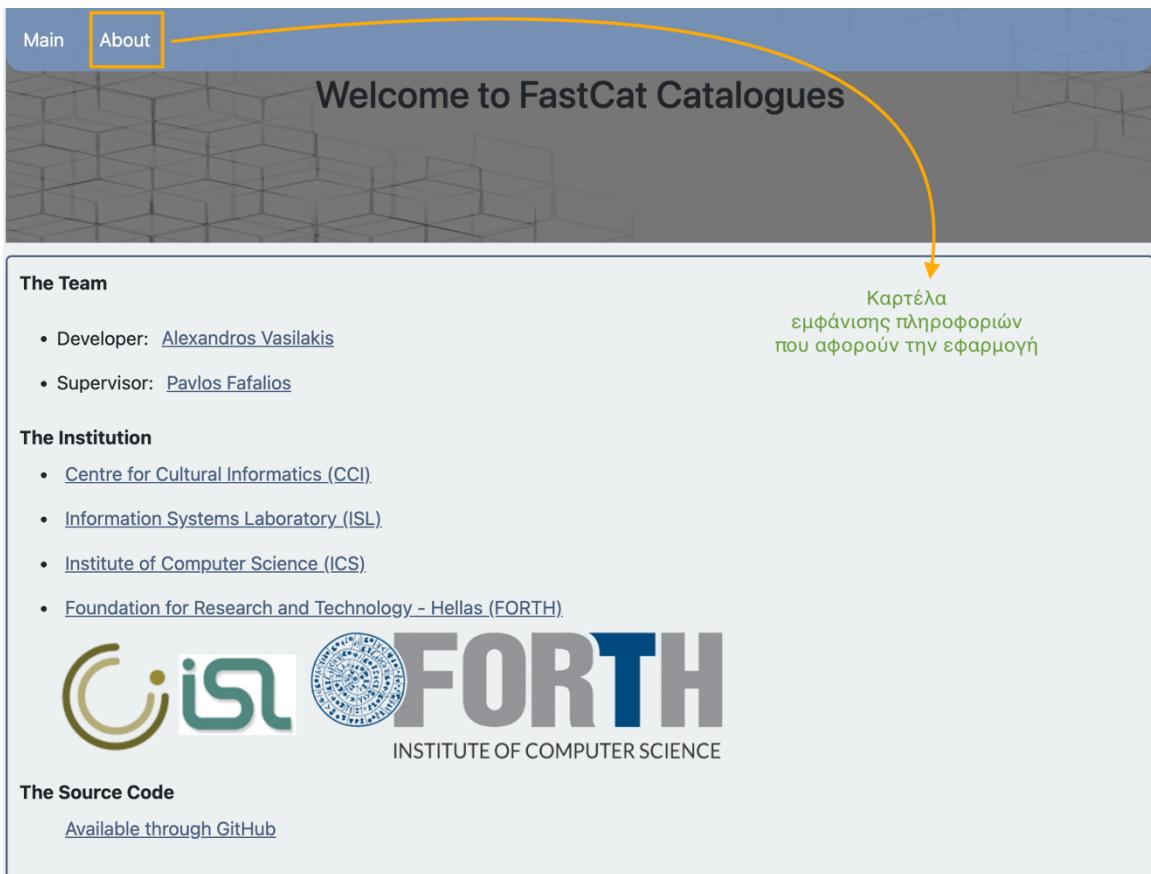
URI: urn:uuid:d15bd72d-f160-4d8c-9463-fc28059140b4

Το ίδιο URI, καλείται σε κάποιο πεδίο πολλαπλές φορές

Εικόνα 21: Ένα URI, μπορεί να κληθεί από ένα πεδίο πολλαπλές φορές

Τέλος, υπάρχει η σελίδα “About”, όπου παρουσιάζονται

- Ο προγραμματιστής που ανέπτυξε την εφαρμογή
- Ο επόπτης που παρακολουθούσε την πορεία ανάπτυξης
- Το ίδρυμα για το οποίο γίνεται η εργασία
- Παραπομπή στον πηγαίο κώδικα



Εικόνα 22: Σελίδα "About" που αφορά την εφαρμογή

3.3. Τεχνολογίες

3.3.1. Γλώσσα Προγραμματισμού

Η γλώσσα προγραμματισμού με την οποία έγινε η ανάπτυξη της εφαρμογής είναι η Java.

Λόγω του ότι χρειάστηκαν αρκετά κομμάτια html, ήταν καλύτερο να χρησιμοποιηθούν αρχεία jsp(Java server page), αντί για servlets.

Το jsp είναι μια server side programming τεχνολογία, που επιτρέπει τη δημιουργία δυναμικής, ανεξάρτητης πλατφόρμας μεθόδου για τη δημιουργία Web apps. Με την χρήση του JSP υπάρχει δυνατότητα πρόσβασης σε όλο το Java API, με πρόσβαση στην βάση δεδομένων

3.3.2. Βιβλιοθήκες που χρησιμοποιήθηκαν:

Για την υλοποίηση κάποιων λειτουργιών της εφαρμογής, χρησιμοποιήθηκαν τόσο βιβλιοθήκες Java, όσο βιβλιοθήκες JavaScript

Java Libraries

Apache Jena:

Το Apache Jena⁶ αποτελεί ένα open source framework της Java, για την δημιουργία Web εφαρμογών. Παρέχει ένα ολοκληρωμένο API με χρήση δεδομένων RDF και υποστηρίζει ένα ευρύ φάσμα λειτουργιών για την διαχείριση τους.

JavaScript Libraries

AG Grid:

Για την δημιουργία του πίνακα, χρησιμοποιήθηκε η βιβλιοθήκη AG Grid⁷, μιας και οι λειτουργίες που παρέχει είναι πολυάριθμες όπως φιλτράρισμα, ταξινόμηση, σελιδοποίηση.

Ο πίνακας οριοθετεί με κατάλληλο τρόπο τα columns κάθε κατηγορίας. (Τα columns προκύπτουν με βάση τα queries που υπάρχουν στο configuration file)

Με τα δεδομένα που χρησιμοποιήθηκαν, αφού γίνει η επιλογή μιας οντότητας θα εμφανιστούν:

- Οι βασικές πληροφορίες για κάθε καταχώρηση, το οποίο προκύπτει από ένα SPARQL Query
- Επιπλέον στήλη που παραπέμπει σε σελίδα με περισσότερες πληροφορίες για κάθε καταχώρηση

⁶ Apache Jena framework, https://jena.apache.org/tutorials/rdf_api.html

⁷ AG Grid, JavaScript framework, <https://www.ag-grid.com/javascript-data-grid/reference/>

Στο παρακάτω παράδειγμα εμφανίζεται αυτό στο οποίο αναφερόμαστε:

Έστω ότι επιλέγουμε την κατηγορία Ships. Στον πίνακα που παρουσιάζεται, εμφανίζονται όλες οι καταχωρήσεις που έχουμε για αυτή την κατηγορία με κάποιες βασικές πληροφορίες(columns). Στην προκειμένη περίπτωση είναι:

- URI
- Label
- ShipType

Πατώντας το σύνδεσμο “More Info”, που βρίσκεται στην τελευταία στήλη μιας καταχώρησης(record)(row), μεταφερόμαστε σε νέα σελίδα, όπου εμφανίζονται (αν υπάρχουν) περισσότερες πληροφορίες (εδώ έχει προστεθεί η δυνατότητα OPTIONAL, σε περιπτώσεις που για κάποιο πεδίο δεν είναι γνωστή αυτή η πληροφορία) με επιπλέον πεδία. Στην περίπτωση του παραδείγματος είναι:

- Identity
- Registered In
- Owner
- Tonnage
- Constructed By

Chart JS

Για την δημιουργία των charts, επιλέχτηκε η βιβλιοθήκη Chart JS⁸. Με την χρήση αυτής της βιβλιοθήκης, δίνεται η δυνατότητα με εύκολο και κατανοητό τρόπο προς τον χρήστη, η οππικοποίηση των δεδομένων για μια στήλη της κατηγορίας που επέλεξε να εμφανίσει.

⁸ Chart JS, Open Source HTML Charts, <https://www.chartjs.org/docs/latest/getting-started/>

4. Μοντέλο Διαμόρφωσης(Configuration model)

Στα αρχεία που έχουν δημιουργηθεί στο παρόν project, υπάρχει εκείνο που αφορά την διαμόρφωση των ρυθμίσεων (configuration properties file : config.properties). Ο τρόπος με τον οποίο διαμορφώθηκε είναι δυναμικός, αφού τα ονόματα και οι παράμετροι που ορίζονται σε αυτό το αρχείο, είναι εκείνα που θα εμφανιστούν στην πλατφόρμα.

4.1. Ιδιότητες αρχείου διαμόρφωσης

Σε αυτό μπορούν να οριστούν (σε ένα preprocessing βήμα) διάφορες παραμέτροι του συστήματος, καθώς και λεπτομέρειες που αφορούν τα δεδομένα, όπως τις κατηγορίες οντοτήτων που μας ενδιαφέρει να εμφανίζονται στην εφαρμογή και τις σχέσεις που μπορεί να έχουν οι οντότητες μεταξύ τους.

Αποτελεί το βασικό αρχείο, η βάση δεδομένων, που χρησιμοποιεί η εφαρμογή όπου:

- από την μια, με τα τωρινά δεδομένα, υπάρχει το ενδεχόμενο-δυνατότητα να δεχτεί αλλαγές με την πάροδο του χρόνου
- από την άλλη, αν αλλάξουν τα δεδομένα, τότε αλλάζουν κι αυτά που παρουσιάζονται στην πλατφόρμα.

Σε κάθε περίπτωση, εφόσον ακολουθήσουμε την ίδια δομή του αρχείου και η αλλαγή γίνει προσεκτικά και στοχευμένα, η πλατφόρμα θα παρουσιάσει οτιδήποτε περιέχει εκείνο το αρχείο ρυθμίσεων, με επιτυχία, χωρίς την παραμικρή τροποποίηση του κώδικα του project.

4.2 Σχεδιασμός αρχείου διαμόρφωσης

Το configuration file είναι απαραίτητο για να τρέξει η εφαρμογή και χρησιμοποιείται για δοκιμή, επιβεβαίωση, διόρθωση και βελτιστοποίηση των λειτουργιών της εφαρμογής που δημιουργήθηκε.

4.2.1. Παράδειγμα αρχείου διαμόρφωσης

Το αρχείο, με τα δεδομένα που χρησιμοποιήθηκαν, έχει δομηθεί με τον εξής τρόπο:

Η πρώτη πληροφορία που είναι απαραίτητο να οριστεί, είναι το όνομα του φακέλου που περιλαμβάνει τα δεδομένα, τον οποίο εισάγουμε στο path:
«ProWeb/src/main/resources/ » του project.

application.folder	= SeaLit
--------------------	----------

Εικόνα 23: Όνομα φακέλου με δεδομένα της εφαρμογής

Όπου “SeaLit”, το όνομα του φακέλου με τα δεδομένα.

Στην συνέχεια περιλαμβάνει τα υπόλοιπα βασικά στοιχεία, αποκλειστικά για την εφαρμογή, τα οποία είναι:

1. Τίτλος Καλωσορίσματος
2. Κύριος Τίτλος Εφαρμογής
3. Υπότιτλος Εφαρμογής
4. Περιγραφή Εφαρμογής

(Με τον τρόπο που φαίνονται στο πρώτο μέρος της ενότητας 3.2. Διεπαφή Χρήστη)

```
application.welcomeTitle      =
application.title              =
application.subtitle           =
application.description        =
```

Εικόνα 24: Αποκλειστικά βασικά στοιχεία της εφαρμογής

Έχετε υπάρχει :

- το πεδίο που περιλαμβάνει τον αριθμό των κατηγοριών των οντοτήτων που μας ενδιαφέρουν

```
categories                  = 1,2,3,4
```

Εικόνα 25: Πεδίο με αριθμό των κατηγοριών των οντοτήτων

και στην συνέχεια,

- στοιχεία που αφορούν κάθε μια κατηγορία(π.χ. categories.X.name (όπου “X”, ο αριθμός για κάθε κατηγορία)

```
categories.1.name            =
categories.1.count_query     =
categories.1.info            =
categories.1.details         =
```

Εικόνα 26: Στοιχεία που αφορούν κάθε κατηγορία

- categories.X.name = Το όνομα της κατηγορίας οντοτήτων
- categories.X.count_query = Ο αριθμός εμφανίσεων (Αφορά τον αριθμό που ένα query βρίσκει μια κατηγορία με το όνομα που μας ενδιαφέρει)
- categories.X.info = Οι βασικές πληροφορίες μιας κατηγορίας (Μέσω ενός query, παίρνουμε τα βασικά columns που θέλουμε να εμφανίζονται σαν πρώτη εικόνα στον πίνακα κάθε κατηγορίας)
- categories.X.details = Οι περαιτέρω λεπτομέρειες μιας κατηγορίας (Μέσω ενός query, παρέχονται περισσότερες λεπτομέρειες για το row, που ο χρήστης επέλεξε να δει)

```

categories.4.name      = Ships
categories.4.count_query = select (count(distinct ?uri) as ?num) where { ?uri a <http://www.sealit.gr/ontology/sealit#Ship> }
categories.4.info       = select ?URI ?Label ?Type where { ?URI a <http://www.sealit.gr/ontology/sealit#Ship>; \
                                                               <http://www.w3.org/2000/01/rdf-schema#label> ?Label ; \
                                                               <http://www.cidoc-crm.org/cidoc-crm/P2\_has\_type> ?ShipType . \
                                                               <http://www.w3.org/2000/01/rdf-schema#label> ?Type}
categories.4.details    = select ?URI ?Label ?Type ?Identity ?Registered_In ?Constructed_By ?Owner ?Tonnage ?Voyages where { \
                                                               ?URI a <http://www.sealit.gr/ontology/sealit#Ship>; \
                                                               <http://www.w3.org/2000/01/rdf-schema#label> ?Label . \
                                                               OPTIONAL{ ?URI ?ShipType <http://www.w3.org/2000/01/rdf-schema#label> ?Type } \
                                                               OPTIONAL{ ?URI ?ShipIdentity <http://www.w3.org/2000/01/rdf-schema#label> ?Identity } \
                                                               OPTIONAL{ ?URI ?ShipRegistered <http://www.cidoc-crm.org/cidoc-crm/P7\_took\_place\_at> ?TookPlace } \
                                                               OPTIONAL{ ?URI ?ShipOwnership <http://www.sealit.gr/ontology/sealit#has\_owner> ?OwnerShip } \
                                                               OPTIONAL{ ?URI ?ShipTonnage <http://www.cidoc-crm.org/cidoc-crm/P90\_has\_value> ?Tonnage } \
                                                               OPTIONAL{ ?URI ?ShipVoyages <http://www.sealit.gr/ontology/sealit#voyages> ?Voyages } \
                                                               OPTIONAL{ ?URI ?Destination <http://www.w3.org/2000/01/rdf-schema#label> ?Destination } \
                                                               OPTIONAL{ ?URI ?ShipConstructed <http://www.sealit.gr/ontology/sealit#was\_constructed\_by> ?UnderName } \
                                                               <http://www.w3.org/2000/01/rdf-schema#label> ?Constructed_By }

Triple, που μας δίνει απευθείας την τιμή
Triple με μετατροπή, που δίνει μόνο την σημαντική πληροφορία
Triple, που η πληροφορία δίνεται μέσω ενός path

```

Εικόνα 27: Παράδειγμα με στοιχεία κάθε κατηγορίας

Όπως φαίνεται στην παραπάνω εικόνα, υπάρχουν περιπτώσεις που η πληροφορία που χρειαζόμαστε, δεν δίνεται αμέσως, αλλά χρειάζεται να ακολουθήσουμε κάποιο path για να την πάρουμε

Προβλήματα που προέκυψαν:

Υπάρχει η πιθανότητα κάποιο column του query, λόγω έλλειψης δεδομένων, να μην έχει τιμή.

Επίλυση:

Σε αυτή την περίπτωση, έχει μπει σε κάθε τέτοιο column η επιπλέον παράμετρος OPTIONAL, προκειμένου να μην δημιουργείται πρόβλημα στην εμφάνιση των αποτελεσμάτων.

- Χωρίς την παράμετρο OPTIONAL:

Όπως φαίνεται στην παρακάτω εικόνα, παίρνουμε την ίδια σελίδα με εκείνη του πίνακα, μαζί με την παράμετρο στο URL που περιλαμβάνει το URI του record(row) που επιλέχτηκε.

http://localhost:8080/ProWeb/grid.jsp?obj=Persons&URIDetails=https://rs.sealitproject.eu/kb/person/5A9B421F-A48F-3A36-8ECD-09B7443B7EE2

Welcome to FastCat Catalogues

Main About

Persons (1910) Locations (442) Voyages (50) Ships (98)

Persons

First Name	Last Name	URI	More
Emanuele	Gimelli	https://rs.sealitproject.e...	More info
Agostino	Graffione	https://rs.sealitproject.e...	More info
Carolina	Gaggero	https://rs.sealitproject.e...	More info
Giacomo	Calcagno	https://rs.sealitproject.e...	More info
Gerolamo	Solimano	https://rs.sealitproject.e...	More info
Franco	Schiattino	https://rs.sealitproject.e...	More info
Domenico Biagio	Ogno	https://rs.sealitproject.e...	More info
Luigi	Pagano	https://rs.sealitproject.e...	More info
Giobatta	Ansaldo	https://rs.sealitproject.e...	More info
Giuseppe	Molfino	https://rs.sealitproject.e...	More info

Page Size: 100 1 to 100 of 1,911 < Page 1 of 20 > >>

Εικόνα 28: Σελίδα χωρίς την παράμετρο OPTIONAL στο query που δίνει τα αποτελέσματα

- Με την χρήση της παραμέτρου OPTIONAL:
Επιλύεται το πρόβλημα και έχουμε τα επιθυμητά αποτελέσματα.

1^η περίπτωση:

Main About

Welcome to FastCat Catalogues

Persons

First Name: Emanuele

Birth Date:

- Date 1843
- 1843

Profession: Mozzo

Employment: provided service

Last Name: Gimelli

Receiver:

- 20
- 4.64

Residence: Portofino

Identity: 7082

URI: <https://rs.sealitproject.eu/kb/person/A11EE0D9-64DA-31C8-94CB-C978D94C243F>

Όλες οι μεταβλητές αυτού του record, έχουν τιμή

Εικόνα 29: Σελίδα με την παράμετρο OPTIONAL στο query που δίνει τα αποτελέσματα

Οπως φαίνεται στην εικόνα παραπάνω, υπάρχουν αρκετές πληροφορίες για την συγκεκριμένη καταχώρηση και όλες οι μεταβλητές έχουν τιμή.

2η περίπτωση:

The screenshot shows the homepage of the FastCat Catalogues. At the top, there are links for 'Main' and 'About'. The main title is 'Welcome to FastCat Catalogues'. On the left, there is a sidebar with four categories: 'Persons (1910)', 'Locations (442)', 'Voyages (50)', and 'Ships (98)'. A green annotation on the left side of the sidebar reads: 'Κάποιες από τις μεταβλητές αυτού του record, έχουν τιμή'. An orange arrow points from this annotation to the 'Identity' section of the main content area. The main content area has a dark blue header 'Persons'. It displays the following information:

- First Name:** Luigi
- Type:** master
- Last Name:** Pagano
- Identity:**
 - Pagano
 - 2639
- URI:** <https://rs.sealitproject.eu/kb/person/5A9B421F-A48F-3A36-8ECD-09B7443B7EE2>

Εικόνα 30: Σελίδα με περίπτωση που δεν έχουν όλες οι μεταβλητές τιμή

Από την άλλη, σε αυτή την εικόνα, πεδία όπως Birth Date, Employment, Receiver, δεν έχουν τιμή, ωστόσο δεν δημιουργείται το παραμικρό πρόβλημα, αφού όλα τα υπόλοιπα (πεδία) εμφανίζονται κανονικά.

5. Εγκατάσταση Συστήματος

Η εφαρμογή απαιτεί την εγκατάσταση των:

- Java Runtime Environment (η δημιουργία του project έγινε με την έκδοση 17.0.8.1)(Java Platform)
- Java EE 6 Web (Java EE Version)

Επίσης είναι απαραίτητο να οριστεί στον φάκελο “resources”, ο οποίος βρίσκεται στον main φάκελο του source folder (src/main/resources):

- Ο φάκελος με τα δεδομένα που θα τρέξει η εφαρμογή(π.χ. SeaLitData)
- Το αρχείο config.properties που ορίζει συγκεκριμένα τα αντικείμενα που θα πάρω από τον από πάνω φάκελο

Αφού γίνει “Run” το project(index.jsp), εκτελείται ο Apache Tomcat (server).

Αυτό θα ανοίξει μια καρτέλα στον browser του μηχανήματος που τρέχει η εφαρμογή και θα γίνει μετάβαση στην σελίδα:

<http://localhost:8080/ProWeb>, όπου

- localhost: η IP address του δικτύου που τρέχει η εφαρμογή,
- 8080: το port που χρησιμοποιείται συνήθως από web servers,
- ProWeb: Τίτλος της εφαρμογής που ορίστηκε

όπου εμφανίζεται το main page της εφαρμογής.

(Ο πλήρης κώδικας, υπάρχει στο [GitHub⁹](#))

6. Απαιτήσεις Συστήματος και Απόδοση

Το μηχάνημα στο οποίο έτρεχε η εφαρμογή, είχε τα παρακάτω χαρακτηριστικά:

Λειτουργικό Σύστημα	MacOS Sonoma 14.2.1
Επεξεργαστής	Apple M1 3.2GHz
Μνήμη	16 GB RAM

Η απόδοση της εφαρμογής, φυσικά έχει να κάνει με τον όγκο των δεδομένων, αφού όλα φορτώνονται στην μνήμη. Σε περίπτωση με πολύ μεγάλο όγκο δεδομένων, είναι απαραίτητη η χρήση ενός ισχυρότερου μηχανήματος.

Τα δεδομένα SeaLit που έχουμε στην διάθεση μας και χρησιμοποιούνται στην εφαρμογή έχουν μέγεθος 100MB και ο αριθμός των triples(subject-predicate-object) είναι 104936.

Για αυτό το μέγεθος του dataset, το πρόγραμμα, στο μηχάνημα που έγινε η υλοποίηση, είναι απόλυτα αποδοτικό.

Ο μέσος χρόνος που χρειάζεται το σύστημα, για να:

- Τρέξει η εφαρμογή, είναι ~12s
- Γίνει η πλοήγηση των διαφορετικών σελίδων της εφαρμογής, είναι ~5s.

⁹ Source Code of “An Autonomus Web Platform for browsing and exploring semantic(RDF) data” A.Vasilakis, 2024, <https://github.com/AleVas1453/Thesis>

7. Επίλογος

7.1. Βασικά συμπεράσματα

Σε αυτή την διπλωματική αναπτύχθηκε μια εφαρμογή που είχε στόχο την παρουσίαση RDF δεδομένων. Η δυνατότητα εμφάνισης τους με κατανοητό και απλό τρόπο, είναι ζωτικής σημασίας, μιας και διευκολύνει την ανάγνωση και αναζήτηση μιας συγκεκριμένης πληροφορίας άμεσα. Η εφαρμογή υποστηρίζει μια πληθώρα λειτουργιών, κάνοντας την περιήγηση της, φιλική προς κάθε χρήστη. Επιπλέον αποδείχτηκε αποτελεσματική, τόσο για τον χειρισμό του όγκου δεδομένων που είχε στην διάθεση της, όσο και για τους χρόνους που χρειάζεται για την εναλλαγή μεταξύ των σελίδων της. Είναι λογικό ότι έχουν δημιουργηθεί προκλήσεις και περιορισμοί, ωστόσο είναι βέβαιο πως μπορούν να ξεπεραστούν σε μελλοντικό χρόνο.

7.2. Αδυναμίες

- Περιορισμένο scalability(επεκτασιμότητα). Κατά την εκκίνηση του προγράμματος, όλα τα δεδομένα φορτώνονται στην μνήμη. Κάτι που σε τεράστιο όγκο δεδομένων (στην προκειμένη περίπτωση αφορά των αριθμό των τριπλετών που φορτώνονται), είναι βέβαιο ότι είτε δεν θα φορτώσουν καθόλου τα δεδομένα, είτε θα πάρει πολύ χρόνο.
- Περιπτώσεις ονομάτων με ειδικούς χαρακτήρες(τόνους, σύμβολα) δεν δίνουν αποτέλεσμα. Χρειάζεται decode-encode που θα λύσει το πρόβλημα

7.3. Μελλοντική εργασία (Βελτιστοποίηση εφαρμογής – Βελτιώσεις)

Οι δυνατότητες που μπορούν να γίνουν στον μέλλον είναι πολυάριθμες με κυρίαρχες τις εξής:

- Κουμπί λήψης, τόσο των δεδομένων του πίνακα και κάθε καταχώρησης ξεχωριστά, όσο και των διαγραμμάτων
- Διασύνδεση των κατηγοριών, με σκοπό την μετάβαση από μια κατηγορία σε άλλη, όταν βρισκόμαστε στην σελίδα που εμφανίζονται περισσότερες πληροφορίες μιας καταχώρησης του πίνακα
- Επέκταση του navbar, με περισσότερες επιλογές
- Βελτίωση της ταχύτητας της εφαρμογής, για περιπτώσεις μεγάλου όγκου δεδομένων
- Δημιουργία επιπλέον διαγραμμάτων, όπως διαγράμματος πίτας(pie chart), που θα μπορούν να δώσουν πιο κατανοητή πληροφορία για κάποια στήλη μιας κατηγορίας

8. Αναφορές / Πηγές

1. Πληροφορίες για τις έννοιες των δεδομένων RDF, URI, SPARQL από τις διαλέξεις του μαθήματος «HY-561 Διαχείριση Δεδομένων στον Παγκόσμιο Ιστό» μέσω της πλατφόρμας elearn.uoc
<https://elearn.uoc.gr/user/view.php?id=13674&course=4510>
2. Επίσημη σελίδα Apache Jena για την διαμόρφωση του API
https://jena.apache.org/tutorials/rdf_api.html
3. Δεδομένα SeaLit
<https://www.ics.forth.gr/isl/fast-cat#usecase>
4. P. Fafalios, A. Kritsotaki, and M. Doerr, "The SeaLiT Ontology – An Extension of CIDOC-CRM for the Modeling and Integration of Maritime History Information". ACM Journal on Computing and Cultural Heritage, 2023.
<https://arxiv.org/pdf/2301.04493>