



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ  
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Ανάπτυξη Εφαρμογής Αναφορών για το EDMS, το  
Σύστημα Διαχείρισης Μηχανικών και  
Εξοπλιστικών Δεδομένων του CERN  
(Ευρωπαϊκός Οργανισμός Πυρηνικών Ερευνών)**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Νικήτας Κ. Κοτσολάκος

**Επιβλέπων :** Γεώργιος Στασινόπουλος  
Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2017





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ  
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Ανάπτυξη Εφαρμογής Αναφορών για το EDMS, το  
Σύστημα Διαχείρισης Μηχανικών και  
Εξοπλιστικών Δεδομένων του CERN  
(Ευρωπαϊκός Οργανισμός Πυρηνικών Ερευνών)**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

Νικήτας Κ. Κοτσολάκος

**Επιβλέπων :** Γεώργιος Στασινόπουλος  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 26<sup>η</sup> Σεπτεμβρίου 2017.

.....  
Γεώργιος Στασινόπουλος  
Καθηγητής Ε.Μ.Π.

.....  
Ευστάθιος Συκάς  
Καθηγητής Ε.Μ.Π.

.....  
Ιωάννα Ρουσσάκη  
Επίκουρη Καθηγήτρια Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2017

.....  
Νικήτας Κ. Κοτσολάκος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών  
Ε.Μ.Π.

Copyright © Νικήτας Κοτσολάκος, 2017.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## **Περίληψη**

Σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάλυση και δημιουργία μίας εφαρμογής αναφορών για το EDMS, το Σύστημα Διαχείρισης Μηχανικών και Εξοπλιστικών Δεδομένων του CERN (Ευρωπαϊκό Κέντρο Πυρηνικών Ερευνών).

Στο πλαίσιο της εργασίας αναλύθηκε η αναγκαιότητα ενός τέτοιου συστήματος αναφορών και μελετήθηκαν πιθανές λύσεις. Αποφασίστηκε η ανάπτυξη μίας Εφαρμογής Μίας Σελίδας (Single Page App) με χρήση Typescript και Angular για τη δημιουργία της διεπαφής χρήστη, Java και PL/SQL για τον εξυπηρετητή και την αλληλεπίδραση με τη σχεσιακή βάση δεδομένων, και χρήση REST διαδικτυακών υπηρεσιών για την επικοινωνία της διεπαφής με τον εξυπηρετητή, όπου τα δεδομένα μεταφέρονται στη μορφή του προτύπου JSON.

Μέσα από την εφαρμογή ο χρήστης μπορεί να δημιουργήσει, δει, τροποποιήσει, αποθηκεύσει και διαγράψει αναφορές σχετικές με τα δεδομένα που είναι αποθηκευμένα στο EDMS και στα οποία έχει πρόσβαση. Οι αναφορές περιλαμβάνουν πίνακες με τα δεδομένα, τα οποία μπορούν να φιλτραριστούν, καθώς και διαγράμματα σχετικά με αυτά τα δεδομένα. Ο χρήστης μπορεί να επιλέξει ποια διαγράμματα θέλει να εμφανίζονται, καθώς και ποιες στήλες στους πίνακες δεδομένων.

Κατά την δημιουργία ή τροποποίηση των παραμέτρων μιας αναφοράς, ο χρήστης επιλέγει τις λογικές συνθήκες που πρέπει να ισχύουν στα διάφορα πεδία των δεδομένων, έχοντας έτσι τη δυνατότητα να προσαρμόσει την αναφορά όπως ακριβώς θέλει. Ταυτόχρονα, για διευκόλυνση του οι πιθανές τιμές των πεδίων προέρχονται με ασύγχρονες κλήσεις από τον εξυπηρετητή, και παρουσιάζονται σε μορφή εύκολη για επιλογή (για παράδειγμα ημερολόγιο για τις ημερομηνίες ή αναπτυσσόμενο μενού στο οποίο μπορεί να γίνει αναζήτηση για πεδία με μία λίστα πιθανών τιμών).

Τέλος, ο χρήστης μπορεί να αποθηκεύσει τις παραμέτρους μίας ή περισσότερων αναφορών, ώστε την επόμενη φορά που θα συνδεθεί να είναι άμεσα διαθέσιμες.

## **Λέξεις Κλειδιά:**

Διαδικτυακή Εφαρμογή, Εφαρμογή Μίας Σελίδας, SPA, EDMS, CERN, Ευρωπαϊκό Κέντρο Πυρηνικών Ερευνών, Αναφορά, Διαγράμματα, Typescript, Angular, Java, PL/SQL

## **Abstract:**

The purpose of this Diploma Thesis is the development of a Reporting Application for EDMS, the Engineering and Equipment Data Management Service of CERN (European Organisation For Nuclear Research).

In the scope of the thesis the need for such a reporting system was analysed and potential solutions were studied. It was decided to develop a Single Page Application, using Typescript and Angular for the front end, Java and PL/SQL for the server and the communication with the Relational Database, and the usage of RESTful Web Services for the communication between the server and the client, with data transferred in JSON format.

Through the application, a user can create, see, edit, save or delete reports regarding the data stored in EDMS, to which he has access. The reports contain tables with the relevant data, which can be filtered by columns, as well as charts about the data. The user can select which of the charts and which of the columns of the data table should be visible.

When creating or editing the parameters of a report, the user selects the logical clauses which are used on the different fields of the data, so that he can customise the report to according to his needs. At the same time, to make things easier, the possible values of the fields are retrieved with asynchronous calls from the server, which are then displayed to the user in a proper format. For example, a calendar widget is used for date selection, or a dropdown menu (which allows searching) for a field with a list of potential values.

Finally, the user may save the parameters of one or more reports for his convenience, so that they are easily available for future use.

## **Keywords:**

Web Application, Single Page Application, SPA, EDMS, CERN, European Organization for Nuclear Research, Report, Charts, Typescript, Angular, Java, PL/SQL

## **Ευχαριστίες**

Θα ήθελα να ευχαριστήσω θερμά τις αδερφές μου Αρετή Κοτσολάκου, Μαρία Κοτσολάκου και την μητέρα μου Γεωργία Γκοτσοπούλου για την ακατάπαυστη στήριξη που μου παρείχαν κατά τη διάρκεια των σπουδών μου, και την ενθάρρυνση να αντιμετωπίσω νέες προκλήσεις.

Επίσης, την Aleksandra Wardzinska για την ευκαιρία που μου έδωσε να γίνω μέλος της ομάδας του EDMS στο CERN, καθώς και τους συναδέλφους μου στο EDMS Griselda Garcia Arza, Krzysztof Pater και Tsvetelin Krastev για όλη την βοήθεια που μου παρείχαν και τις γνώσεις που μοιράστηκαν μαζί μου.





## Περιεχόμενα

Περίληψη.....	5
Λέξεις Κλειδιά:.....	5
Abstract: .....	6
Keywords: .....	6
Ευχαριστίες.....	7
Περιεχόμενα .....	9
1. Εισαγωγή .....	11
1.1 Σχετικά με το CERN .....	12
1.2 Σχετικά με τον LHC .....	13
1.3 EDMS: Το Σύστημα Διαχείρισης Μηχανικών και Εξοπλιστικών Δεδομένων του CERN .....	14
1.3.1 Βασικές λειτουργίες του EDMS.....	15
1.3.2 Κατηγορίες Χρηστών του EDMS .....	16
1.4 Βασικά Χαρακτηριστικά του EDMS .....	17
2. Προσδιορισμός του προβλήματος και Ανάλυση Απαιτήσεων.....	19
2.1 Αναγκαιότητα Αναφορών στο EDMS.....	19
2.2 Διαδικασία Συλλογής Απαιτήσεων .....	21
2.2.1 Συνάντηση τρίμηνου σχεδιασμού EDMS .....	22
2.2.2 Συνάντηση τοπικών διαχειριστών .....	22
2.2.3 Καταγραφή αιτημάτων για δεδομένα.....	22
2.2.4 Συναντήσεις με σημαντικούς χρήστες.....	23
2.2.5 Συγκέντρωση, ανάλυση και εσωτερική συζήτηση.....	24
2.3 Απαιτήσεις Εφαρμογής .....	25
2.4 Επιλογή τεχνολογικής λύσης.....	27
2.4.1 Ανάλυση πρότασης εξωτερικού συστήματος (Pentaho).....	27
2.4.2 Ανάλυση πρότασης εφαρμογής εντός του EDMS .....	28
2.4.3 Απόφαση για ανάπτυξη εφαρμογής .....	29
3. Τεχνολογίες που χρησιμοποιήθηκαν .....	31
3.1 Front End .....	31
3.1.1 Angular.....	31
3.1.2 Typescript.....	33
3.1.3 Primeng .....	34
3.1.4 Chart.js .....	34
3.2 Back End .....	35
3.2.1 Java EE.....	35
3.2.2 PL/SQL.....	35
3.2.3 Oracle SQL.....	35
3.3 Επικοινωνία Πελάτη – Εξυπηρετητή .....	36
3.3.1 JSON .....	36
RESTful Διαδικτυακές Υπηρεσίες.....	37
3.4 Αρχιτεκτονική Εφαρμογής .....	38
4. Λειτουργία Εφαρμογής .....	41
4.1 Γενική Λειτουργία Εφαρμογής .....	41
4.2 Συστατικά - Οθόνες της εφαρμογής.....	43
4.2.1 Επιλογή Παραμέτρων.....	43
4.2.2 Επιλογή Διαγραμμάτων.....	44
4.2.3 Παρουσίαση Διαγραμμάτων .....	44

4.2.4 Πίνακας Δεδομένων .....	45
4.2.5 Βασική Οθόνη Εφαρμογής.....	46
4.3 Παράδειγμα Χρήσης.....	47
5. Ασφάλεια, Απόδοση και Βελτιώσεις .....	51
5.1 Ασφάλεια .....	51
5.2 Απόδοση – Εμπειρία χρήσης.....	53
5.3 Σκέψεις για Μελλοντικές Προσθήκες .....	54
6. Υλοποίηση.....	57
6.1 Front End .....	57
6.1.1 Angular Models.....	57
6.1.2 Angular Components.....	60
6.1.3 Angular Services .....	65
6.2 Back End .....	69
6.2.1 Java Classes.....	69
6.2.2 Database Stored Procedures .....	71
7. Βιβλιογραφία .....	73
7.1 Βιβλία .....	73
7.2 Επιστημονικές Δημοσιεύσεις .....	73
7.3 Ιστοσελίδες.....	74
7.4 Άλλα .....	74

# 1. Εισαγωγή

Η παρούσα διπλωματική εργασία αφορά στη δημιουργία μιας εφαρμογής μέσω της οποίας οι χρήστες θα μπορούν να παράγουν αναφορές για δεδομένα αποθηκευμένα στο σύστημα EDMS του CERN. Συγκεκριμένα, οι χρήστες θα μπορούν να δημιουργήσουν, αποθηκεύσουν και τροποποιήσουν αναφορές πάνω στις βασικές οντότητες του EDMS, χρησιμοποιώντας πλήθος παραμέτρων για να περιορίσουν τα αποτελέσματα σε όσα τους ενδιαφέρουν.

Επίσης, θα μπορούν να επιλέξουν τι διαγράμματα και τι στοιχεία θέλουν να εμφανίζονται στις αναφορές τους, καθώς και να αποθηκεύουν τις αναφορές τους ώστε να είναι διαθέσιμες σε επόμενες συνδέσεις. Η εφαρμογή θα ενσωματωθεί ως εφαρμογή μονής σελίδας μέσα στο σύστημα EDMS, βελτιώνοντας την εμπειρία χρήσης του και προσφέροντας νέες δυνατότητες για τον έλεγχο των δεδομένων και την εξαγωγή συμπερασμάτων.

Σε αυτό το κεφάλαιο θα δοθούν ορισμένα βιβλιογραφικά στοιχεία για το CERN, τον LHC και το EDMS, τα οποία βοηθούν στην κατανόηση του γενικότερου πλαισίου του συστήματος στο οποίο δημιουργήθηκε η εφαρμογή, και αναδεικνύουν τη χρησιμότητα της. Στη συνέχεια, θα αναφερθούν οι βασικές λειτουργίες, οι ομάδες χρηστών και τα βασικά χαρακτηριστικά του EDMS, έτσι ώστε στα επόμενα κεφάλαια να είναι πιο κατανοητά τα χαρακτηριστικά και οι λειτουργίες της εφαρμογής.

## 1.1 Σχετικά με το CERN

CERN είναι το ακρωνύμιο του **Ευρωπαϊκού Οργανισμού Πυρηνικών Ερευνών** (European Organisation for Nuclear Research), προερχόμενο από το αρχικό Γαλλικό του όνομα "Conseil Européen pour la Recherche Nucléaire". Ιδρύθηκε λίγα χρόνια μετά τη λήξη του Β' παγκοσμίου πολέμου, το Σεπτέμβριο του 1954 από δώδεκα χώρες, ανάμεσα τους και η Ελλάδα, με έδρα τη Γενεύη. Σκοπός του ήταν η συνεργασία ανάμεσα στα Ευρωπαϊκά κράτη για την προώθηση της επιστημονικής θεμελιώδους έρευνας. Σήμερα έχει 22 κράτη μέλη και 7 ακόμα συνδεδεμένα κράτη, ενώ οι συνεργασίες του δεν περιορίζονται μόνο στα Ευρωπαϊκά κράτη. Ενδεικτικά, το 2017 η Ινδία εντάχθηκε ως συνδεδεμένο μέλος, ενώ οι Ιαπωνία, Ρωσία και ΗΠΑ είναι κράτη - παρατηρητές.



Έχοντας ως βασικό στόχο την συμφιλίωση και συνεννόηση των κρατών για την επιστημονική έρευνα, από την ιδρυτική συνθήκη του CERN καθορίστηκε ότι δεν θα έχει καμία ανάμειξη σε έρευνες για πολεμικούς σκοπούς, καθώς και ότι όλα τα πειραματικά και ερευνητικά αποτελέσματα θα δημοσιεύονταν και θα ήταν διαθέσιμα σε όλους. Η αποστολή του CERN με βάση την ιδρυτική του συνθήκη αποτελείται από 4 σκέλη:

- Έρευνα: Αναζήτηση απαντήσεων σχετικά με το σύμπαν
- Τεχνολογία: Προώθηση και ξεπέρασμα των τεχνολογικών ορίων
- Συνεργασία: Να φέρει τις χώρες πιο κοντά μέσω της επιστήμης
- Εκπαίδευση: Εκπαίδευση των αυριανών επιστημόνων

Βασικό ερευνητικό αντικείμενο στο CERN είναι τα στοιχειώδη σωματίδια, τα χαρακτηριστικά και οι αλληλεπιδράσεις τους, που προσφέρουν πολύτιμες πληροφορίες για το σύμπαν, την ιστορία του και τη λειτουργία του. Για το σκοπό αυτό, από την ίδρυση του το CERN κατασκεύασε μια σειρά επιταχυντών στοιχειωδών σωματιδίων, στους οποίους δέσμες σωματιδίων επιταχύνονται σε τεράστιες ταχύτητες, και στη συνέχεια συγκρούονται. Τα αποτελέσματα των συγκρούσεων αυτών καταγράφονται από ανιχνευτές, και η ανάλυση τους έχει προσφέρει χρήσιμες πληροφορίες.

Αυτή τη στιγμή διαθέτει ένα σύμπλεγμα από εννέα επιταχυντές (ευθύγραμμους και κυκλικούς), ο μεγαλύτερος εκ των οποίων είναι ο Μεγάλος Επιταχυντής Αδρονίων (Large Hadron Collider ή LHC).

## 1.2 Σχετικά με τον LHC

Ο Μεγάλος Επιταχυντής Αδρονίων (LHC) με μήκος 27 χιλιόμετρα είναι ο μεγαλύτερος επιταχυντής σωματιδίων στον κόσμο. Είναι υπόγειος, στα 100 μέτρα βάθος, και διασχίζει τα σύνορα Ελβετίας – Γαλλίας. Στον LHC εισάγονται 2 δέσμες σωματιδίων με αντίθετη φορά, που επιταχύνονται και φτάνουν σε ταχύτητες πολύ κοντά στην ταχύτητα του φωτός ( 99,999999% της ταχύτητας του φωτός). Στη συνέχεια συγκρούονται μετωπικά σε τέσσερα σημεία του δακτυλίου, όπου βρίσκονται εγκατεστημένοι οι τέσσερις ανιχνευτές σωματιδίων του LHC, ένας για κάθε πείραμα (ATLAS, CMS, Alice, LHCb).



Κατά τη λειτουργία του επιταχυντή πραγματοποιούνται περίπου ένα δισεκατομμύριο συγκρούσεις το δευτερόλεπτο, που παράγουν περίπου 1 Petabyte δεδομένων συγκρούσεων ανά δευτερόλεπτο (1 PB/s). Επειδή τα δεδομένα αυτά είναι αδύνατο να αποθηκευτούν και να αναλυθούν ακόμα και με τις σημερινές υπολογιστικές δυνατότητες, φιλτράρονται επιτόπου από τα πειράματα σε δύο επίπεδα. Έπειτα στέλνονται για αποθήκευση (και αργότερα ανάλυση) στα κέντρα δεδομένων του CERN, όπου φτάνουν περίπου «μόλις» 1 PB ανά ημέρα. Για την ανάλυση τους χρησιμοποιείται το μεγαλύτερο πλέγμα υπολογιστών στον κόσμο, το Παγκόσμιο Υπολογιστικό Πλέγμα Μεγάλου Επιταχυντή Αδρονίων (Worldwide LHC Computing Grid), που συνδέει χιλιάδες υπολογιστές σε 170 κέντρα και 41 χώρες.

Η κατασκευή του LHC εγκρίθηκε από το συμβούλιο του CERN τον Δεκέμβριο του 1994, και ο LHC τέθηκε για πρώτη φορά σε λειτουργία το Σεπτέμβριο του 2008, 14 χρόνια μετά, ενώ η λειτουργία του έχει ήδη προγραμματιστεί τουλάχιστον ως το 2035.

## 1.3 EDMS: Το Σύστημα Διαχείρισης Μηχανικών και Εξοπλιστικών Δεδομένων του CERN

Όπως φάνηκε από το παράδειγμα του LHC, τα έργα του CERN μπορεί να έχουν διάρκεια ζωής δεκαετιών, από το σχεδιασμό και την κατασκευή τους, μέχρι τη λειτουργία, τη συντήρηση και την απόσυρση τους. Ταυτόχρονα, η τεράστια πολυπλοκότητα τους, σε συνδυασμό με την συχνή εναλλαγή προσωπικού, σημαίνει ότι ένα έργο μπορεί να σχεδιαστεί, κατασκευαστεί και λειτουργήσει από διαφορετικές γενιές μηχανικών. Ταυτόχρονα με την απόφαση για την κατασκευή του LHC το 1994, αποφασίστηκε ότι είναι αναγκαίο και ένα ηλεκτρονικό σύστημα διαχείρισης δεδομένων, για την διασφάλιση της οργάνωσης και της ποιότητας των μηχανικών δεδομένων του LHC. Έτσι ξεκίνησε το EDMS.

Το EDMS είναι το Σύστημα Διαχείρισης Μηχανικών και Εξοπλιστικών Δεδομένων (Engineering and Equipment Data Management System) του CERN. Υποστηρίζει ολόκληρο τον κύκλο ζωής του LHC και των εξαρτημάτων του, από τη σύλληψη τους, τη σχεδίαση, την κατασκευή, την εγκατάσταση, τη λειτουργία, τη συντήρηση ως και την απόσυρση τους. Με το πέρασμα των χρόνων έχει γνωρίσει μεγάλες αναβαθμίσεις, και πλέον χρησιμοποιείται από πολλά τμήματα του CERN για διάφορα έργα. Βέβαια, κυριότερος και σημαντικότερος χρήστης του παραμένουν οι ομάδες υπεύθυνες για την κατασκευή, λειτουργία, αναβάθμιση και συντήρηση του LHC. Στο EDMS υπάρχουν αποθηκευμένα σήμερα περισσότερα από 1 εκατομμύρια έγγραφα με 2 εκατομμύρια αρχεία, ενώ το χρησιμοποιούν περίπου 800 μοναδικοί χρήστες την ημέρα, με περισσότερους από 2.000 σταθερούς χρήστες (μηνιαίως).

#	ID	Title	Status	Created on	Author	Document type
20	LHC-L-SF-0001 v.3.1	LHC Safety File Descriptive Part	In Work	2012-10-10	CHRISTELLE GAI	Safety File/Folder
30	LHC-L-SF-0002 v.3.1	LHC Safety File Demonstrative Part	In Work	2012-10-10	CHRISTELLE GAI	Safety File/Folder
40	LHC-L-SF-0003 v.3.3	LHC Safety File Operational Part	Cancelled	2016-04-19	CHRISTELLE GAI	Safety File/Folder
40	LHC-L-SF-0003 v.3.4	LHC Safety File Operational Part	Approval Acce	2016-06-06	CHRISTELLE GAI	Safety File/Folder
40	LHC-L-SF-0003 v.3.5	LHC Safety File Operational Part	Under Approvz	2016-07-12	CHRISTELLE GAI	Safety File/Folder
50	LHC-L-SF-0004 v.3.3	LHC Safety File REM Part	Under Approvz	2016-07-15	CHRISTELLE GAI	Safety File/Folder
50	LHC-L-SF-0004 v.3.2	LHC Safety File REM Part	Approval Acce	2016-05-02	CHRISTELLE GAI	Safety File/Folder

### 1.3.1 Βασικές λειτουργίες του EDMS

Οι βασικές λειτουργίες που προσφέρει το EDMS είναι οι εξής:

- Η δομημένη προσέγγιση στην διαχείριση των δεδομένων, με ευέλικτη οργάνωση τους σε έργα, δημιουργία και κατηγοριοποίηση εγγράφων, παραμετροποιήσιμες ιδιότητες, σύνδεση εγγράφων με αντικείμενα. Ακόμη, σύνδεση αντικειμένων μεταξύ τους για τη δημιουργία καταλόγου υλικών (bill of materials), σύνδεση των αντικειμένων με τις απαιτήσεις, τα σχέδια και την τεκμηρίωση τους.
- Κύκλο ζωής των εγγράφων και των αντικειμένων, με δυνατότητα υλοποίησης και επιβολής επίσημων διαδικασιών για τον έλεγχο, τη συζήτηση, την έγκριση και μελλοντικά την αντικατάστασή τους.
- Έλεγχος εκδόσεων, με διατήρηση όλου του ιστορικού των εγγράφων και των αντικειμένων, με καταγραφή της ημερομηνίας και του χρήστη που έκανε ή ενέκρινε τις όποιες αλλαγές, και στάδια ζωής των εγγράφων στα οποία απαγορεύεται κάθε αλλαγή αν δεν δημιουργηθεί νέα έκδοση.
- Η δυνατότητα για συνεργατική δουλειά, με ευέλικτο και παραμετροποιήσιμο σύστημα δικαιωμάτων πρόσβασης, συνδεδεμένο με την υπηρεσία CERN Single Sign On. Διευκολύνεται ακόμα από τη δυνατότητα για αυτόματο συγχρονισμό με τα e-groups του CERN, τη δυνατότητα για αλλαγή του επιπέδου πρόσβασης (απόρρητο / περιορισμένο / εσωτερικό του CERN / δημόσιο) ανάλογα και με το στάδιο στον κύκλο ζωής του εγγράφου, και άλλες δυνατότητες που επιτρέπουν την χρήση των εγγράφων όχι μόνο από την ίδια την ομάδα που τα συνέταξε, αλλά από όλους τους ενδιαφερόμενους. Συχνό φαινόμενο για παράδειγμα είναι η χρήση του από κάποια εταιρεία που ανέλαβε την κατασκευή ενός εξαρτήματος, τα σχέδια και η ανάλυση του οποίου φυλάσσονται στο EDMS.
- Δυνατότητα για αναζήτηση έργων, εγγράφων ή αντικειμένων, με βάση οποιαδήποτε ιδιότητα τους, καθώς και αναζήτηση στο εσωτερικό των αρχείων.
- Διασύνδεση με άλλα συστήματα του CERN, όπως τα προγράμματα σχεδίασης (για την απευθείας αποθήκευση και ανανέωση των σχεδίων στο EDMS) ή το σύστημα διαχείρισης περιουσιακών στοιχείων και συντήρησης του CERN (για τη σύνδεση σχεδίων και τεκμηρίωσης αντικειμένων με τα πραγματικά κατασκευασμένα και χρησιμοποιούμενα εξαρτήματα)

### 1.3.2 Κατηγορίες Χρηστών του EDMS

Οι κατηγορίες χρηστών του EDMS είναι οι ακόλουθες:

- **Επισκέπτες (Guests):** Πρόκειται για χρήστες που δεν έχουν ταυτοποιηθεί ακόμα από το σύστημα. Έχουν δικαίωμα μόνο να διαβάσουν έγγραφα τα οποία είναι δημόσια. Συνήθως είναι περιστασιακοί χρήστες που χρειάστηκε να διαβάσουν κάποιο έγγραφο, ή απλά τακτικοί χρήστες που δεν έχουν συνδεθεί στην υπηρεσία, καθώς δεν χρειαζόταν εκείνη την στιγμή για την πρόσβαση στο έγγραφο που ήθελαν.
- **Χρήστες (Users):** Πρόκειται για την μεγάλη πλειοψηφία των χρηστών, και αποτελείται από όλους όσους έχουν έναν λογαριασμό CERN. Έχουν πρόσβαση και μπορεί να δουν και να δημιουργήσουν έγγραφα και αντικείμενα σε διάφορα έργα, ανάλογα με τα δικαιώματα που έχουν καθοριστεί σε κάθε έργο. Συνήθως χρησιμοποιούν το σύστημα για να διαβάσουν έγγραφα που τους αφορούν, ή να δημιουργήσουν και να ανεβάσουν έγγραφα που απαιτούνται από την θέση εργασίας τους. Ειδική αναφορά χρειάζεται εδώ στους χρήστες που ανήκουν στα σχεδιαστικά γραφεία του CERN, καθώς παρότι δεν αποτελούν ξεχωριστή κατηγορία, πρόκειται για συστηματικούς χρήστες που καθημερινά δημιουργούν ή τροποποιούν έγγραφα και σχέδια, ενώ πολλές φορές λαμβάνουν μέρος στις διαδικασίες επίσημης έγκρισης σχεδίων.
- **Τοπικοί Διαχειριστές (Local Administrators):** Πρόκειται για χρήστες που έχουν οριστεί ως υπεύθυνοι για το EDMS σε μια συγκεκριμένη ομάδα / τμήμα / τομέα. Έχουν περάσει από εκπαίδευση πάνω στις προχωρημένες λειτουργίες του EDMS στις οποίες έχουν πρόσβαση. Είναι υπεύθυνοι για τη δημιουργία νέων έργων, τον καθορισμό διαδικασιών και των δικαιωμάτων σε όλα τα έργα για τα οποία είναι υπεύθυνοι, και αποτελούν την πρώτη γραμμή υποστήριξης στους χρήστες, καθώς μπορούν να λύσουν τα περισσότερα από τα καθημερινά προβλήματα. Είναι επίσης υπεύθυνοι για την προώθηση σωστών διαδικασιών στους χρήστες μέσα από τη χρήση του EDMS.
- **Διαχειριστές (Administrators ή Managers):** Αποτελείται από έναν μικρό αριθμό ατόμων, που είναι υπεύθυνοι για την λειτουργία του EDMS συνολικά και έχουν αυξημένα δικαιώματα (για παράδειγμα τα μέλη της υπηρεσίας υποστήριξης του EDMS ή ο υπεύθυνος της ομάδας του EDMS).



## **1.4 Βασικά Χαρακτηριστικά του EDMS**

Στο EDMS υπάρχουν τρεις βασικές οντότητες: Τα Projects (έργα), τα Documents (έγγραφα) και τα Items (αντικείμενα). Τα Projects μπορεί να εμπεριέχουν άλλα Projects, Documents ή Items. Τα Items μπορεί να εμπεριέχουν άλλα Items και Documents. Τα Documents μπορεί να εμπεριέχουν άλλα Documents και Files (αρχεία).

Κάθε Project, Document και Item έχει μερικές κοινές ιδιότητες, όπως id (αναγνωριστικό), title (τίτλο), description (περιγραφή), author (συγγραφέα), owner (ιδιοκτήτη), group (ομάδα), creation date (ημερομηνία δημιουργίας), update date (ημερομηνία ανανέωσης) και άλλα. Τα Documents διαθέτουν ακόμα version (έκδοση) και document type (τύπο εγγράφου), με κύριο το διαχωρισμό ανάμεσα σε drawings (σχέδια) και τους υπόλοιπους τύπους εγγράφων. Αυτό επειδή τα drawings διαθέτουν ορισμένες επιπλέον ιδιότητες (τύπο, μέγεθος, σχεδιαστικό γραφείο, καθώς και ημερομηνίες ελέγχων των σχεδίων στο CDD (CERN Drawing System), ένα υποσύστημα του EDMS για την έγκριση, αποθήκευση και δημοσίευση σχεδίων).

Επίσης, κάθε Document υπόκειται σε μία Release Procedure (διαδικασία έκδοσης) την οποία ακολουθεί, η οποία αποτελείται από τα διάφορα στάδια τα οποία περνάει ένα έγγραφο (για παράδειγμα In Work, Under Approval, Approved, Obsolete).

Ακόμη, κάθε οντότητα μπορεί να φέρει και έναν συγκεκριμένο Equipment Code (Κωδικό εξοπλισμού), που χρησιμοποιείται για αναφορά σε συγκεκριμένα αντικείμενα των επιταχυντών, και τα σχέδια τους, αφού αυτόματα οποιοδήποτε σχέδιο αποθηκεύεται στο EDMS παίρνει αυτόματα τον κατάλληλο Equipment Code του.

Μια πολύ σημαντική ιδιότητα όλων των οντοτήτων είναι το context (πλαίσιο) στο οποίο ανήκουν. Κάθε οντότητα ανήκει σε ένα ακριβώς Context, με βάση το οποίο καθορίζονται τα δικαιώματα πρόσβασης, τα διαθέσιμα Document Types, καθώς και τα διαθέσιμα Release Procedures για κάθε Document Type.

Επίσης καθορίζονται οι χρήστες που διαθέτουν συγκεκριμένους ρόλους για τα Release Procedures αυτού του context, κάτι που τους δίνει το δικαίωμα να πραγματοποιούν μεταβάσεις από το ένα στάδιο στο άλλο. Για παράδειγμα μόνο χρήστες με ρόλο Approval Leader στο συγκεκριμένο context μπορούν να κάνουν δεκτό ένα έγγραφο μετά από μία επίσημη Document Approval Process (Διαδικασία έγκρισης εγγράφου).

Επιπρόσθετα, στο Context καθορίζονται και το ποιοι είναι οι διαθέσιμοι Equipment Codes για τα έγγραφα και τα αντικείμενα αυτού του context, για να εξασφαλίζεται ο σωστός έλεγχος δικαιωμάτων. Τα Contexts τα διαχειρίζονται οι Local Administrators (τοπικοί διαχειριστές), που όπως είδαμε είναι βασικοί χρήστες που αναμένεται να αξιοποιήσουν αρκετά την εφαρμογή αναφορών.

## 2. Προσδιορισμός του προβλήματος και Ανάλυση Απαιτήσεων

Στο παρών κεφάλαιο πραγματοποιείται η τεκμηρίωση της αναγκαιότητας ανάπτυξης της παρούσας εφαρμογής, τα προβλήματα τα οποία καλείται να λύσει και οι διαδικασίες στη βελτίωση των οποίων θα συμβάλλει. Καταγράφεται η διαδικασία που ακολουθήθηκε για τη συλλογή και τον προσδιορισμό των απαιτήσεων της εφαρμογής, καθώς και για την επιλογή της τεχνολογικής λύσης και των εργαλείων.

### **2.1 Αναγκαιότητα Αναφορών στο EDMS**

Όπως είδαμε, το EDMS είναι ένα πολύπλοκο σύστημα που χρησιμοποιείται για να παρακολουθήσει ολόκληρη την πορεία ζωής κάποιων πολύ μεγάλων έργων, με διάρκεια ζωής πολλών δεκαετιών. Ταυτόχρονα, η χρησιμότητα των συστημάτων διαχείρισης δεδομένων είναι ευθέως ανάλογη με την ποιότητα των δεδομένων τους: Όσες δυνατότητες και να προσφέρει ένα σύστημα, αν δεν εξασφαλίζεται η ακρίβεια των δεδομένων και η συνεχής ενημέρωσή τους, χάνεται η αξιοπιστία του.

Για παράδειγμα, σε περίπτωση που προετοιμάζεται κάποια αντικατάσταση / αναβάθμιση ενός εξαρτήματος του LHC, αν υπάρχουν τα σχέδια και οι προδιαγραφές του και μπορούν να βρεθούν εύκολα, η προετοιμασία μπορεί να γίνει κατά τη διάρκεια λειτουργίας του επιταχυντή, ώστε στην τεχνική στάση να χρειάζεται να γίνει μόνο η εγκατάσταση και ο έλεγχος.

Ωστόσο, αν δεν βρεθούν τα σχέδια ή δεν φέρουν επίσημη έγκριση ή αν δεν γνωρίζουμε με βεβαιότητα ότι είναι ενημερωμένα και αφορούν το υπάρχον εξάρτημα και όχι κάποιο παλαιότερο που αντικαταστάθηκε, η χρησιμότητα τους μειώνεται, καθώς θα είναι απαραίτητος ο επιτόπιος έλεγχος πριν την κατασκευή του νέου εξαρτήματος. Στον LHC επιτόπιος έλεγχος μπορεί να λάβει χώρα μόνο αρκετές εβδομάδες μετά την παύση λειτουργίας του (λόγω ακτινοβολίας), με αποτέλεσμα καθυστερήσεις μηνών που θα μπορούσαν να αποφευχθούν.

Η διασφάλιση της ποιότητας των δεδομένων δεν είναι εύκολη, και απαιτεί συνειδητή προσπάθεια από όλα τα συνεργαζόμενα τμήματα, αλλά και αυστηρό έλεγχο. Αυτός ο έλεγχος μπορεί να είναι αρκετά δύσκολος και χρονοβόρος, και απαιτεί να είναι διαθέσιμα συγκεντρωτικά στοιχεία για έγγραφα ή αντικείμενα με συγκεκριμένα χαρακτηριστικά, ώστε να μπορεί να ελεγχθεί το σε ποιο στάδιο του κύκλου ζωής τους βρίσκονται, τότε και με ποιες διαδικασίες εγκρίθηκαν, τι απομένει να γίνει.

Ως σήμερα, σε τέτοιες περιπτώσεις οι χρήστες χρειαζόταν να έρθουν σε επαφή με την υποστήριξη του EDMS και να ζητήσουν αυτά τα στοιχεία. Στη συνέχεια, να γίνει έλεγχος ότι ο χρήστης έχει τα κατάλληλα δικαιώματα πρόσβασης και μετά κάποιος από τους προγραμματιστές να γράψει, ελέγξει και εκτελέσει το απαραίτητα ερώτημα για να τα εξαγάγει από τη βάση.

Αυτή η διαδικασία, μαζί με όλη την ενδιάμεση επικοινωνία και πιθανά νέα στοιχεία που να χρειάζονταν (αν για παράδειγμα η μελέτη των πρώτων στοιχείων απαιτεί πιο λεπτομερή ανάλυση σε ένα υποσύνολο) προκαλούσαν καθυστερήσεις στους χρήστες και απαιτούσαν χρόνο από τους προγραμματιστές του EDMS. Ταυτόχρονα, το γεγονός ότι δεν ήταν άμεσα και εύκολα διαθέσιμα μπορεί να σήμαινε ότι ο έλεγχος δεν διεξαγόταν όσο τακτικά θα έπρεπε, ιδιαίτερα σε μικρότερα έργα που χρησιμοποιούν το EDMS και δεν έχουν τις αυστηρές επίσημες διαδικασίες διασφάλισης ποιότητας που έχει ο LHC.

Επιπλέον, από το 2005 το EDMS έχει επιλεγθεί για την καταγραφή και αποθήκευση των επιθεωρήσεων ασφαλείας σε όλα τα κτήρια και τον εξοπλισμό του CERN. Για την χρήση αυτή έχουν ήδη αναπτυχθεί νέες λειτουργικότητες, όμως η δυνατότητα εξαγωγής συγκεντρωτικών στοιχείων και ανάλυσης τους είναι ζωτικής σημασίας για ένα τόσο σημαντικό θέμα. Αυτό μέχρι σήμερα γινόταν με έμμεσους τρόπους (είτε με αιτήματα προς την υπηρεσία του EDMS, είτε με χρήση στοιχείων που τηρούσαν οι υπεύθυνοι απαιτώντας όμως διπλή δουλειά για την καταγραφή τους), και πρόκειται για ένα ακόμα σημείο που αναδεικνύει την αναγκαιότητα ύπαρξης αναφορών στο EDMS.

Πέρα από το κομμάτι του ελέγχου όμως, η διαθεσιμότητα συγκεντρωτικών και αναλυτικών στοιχείων για τα έγγραφα και τα αντικείμενα στο EDMS μπορείς να προσφέρει πολύτιμες πληροφορίες για μια σειρά διαδικασίες που ακολουθούνται και πιθανούς τρόπους βελτίωσής τους, να διαπιστωθούν τάσεις των χρηστών και μέσα για την διευκόλυνση τους και πολλά άλλα. Η ομάδα που είναι υπεύθυνη για την ανάπτυξη του EDMS, εξάγει τουλάχιστον ανά κάποιους μήνες τέτοια στατιστικά στοιχεία, διαδικασία που θα επιταχυνθεί πολύ με την ανάπτυξη μιας εφαρμογής αναφορών.

## **2.2 Διαδικασία Συλλογής Απαιτήσεων**

Ιστορικά, η προηγούμενη έκδοση του EDMS (EDMS 5, που αντικαταστάθηκε το 2014) παρείχε στους τοπικούς διαχειριστές τη δυνατότητα να εξάγουν κάποια πολύ περιορισμένα συγκεντρωτικά στοιχεία για έγγραφα και αντικείμενα. Ωστόσο, οι δυνατότητες παραμετροποίησης ήταν ελάχιστες, τα στοιχεία απλά συγκεντρωτικά νούμερα και η οπτική παρουσίαση τουλάχιστον ελλιπής.

Η πρόθεση για ανάπτυξη ενός πιο σύγχρονου και λειτουργικού συστήματος αναφορών υπήρχε από την αρχή δημιουργίας της 6<sup>ης</sup> (σημερινής) έκδοσης του EDMS, αλλά καθυστέρησε λόγω άλλων προτεραιοτήτων. Ωστόσο, υπήρχαν ήδη κάποιες υποτυπώδεις σκέψεις και προτάσεις για τα χαρακτηριστικά του.

Για την ανάπτυξη του EDMS, η ομάδα ακολουθεί μια διαδικασία που βασίζεται στις ευέλικτες μεθοδολογίες (Agile Methodologies), όπως αυτές αναφέρθηκαν στο Μανιφέστο για την Ευέλικτη Ανάπτυξη Λογισμικού. Χρησιμοποιείται η μέθοδος scrum, που είναι ένα πλαίσιο επαναληπτικής και επαυξητικής ευέλικτης μεθοδολογίας ανάπτυξης λογισμικού. Στο πλαίσιο του scrum πραγματοποιούνται μηνιαία sprints, στα οποία η ομάδα καλείται να ολοκληρώσει την ανάπτυξη συγκεκριμένων χαρακτηριστικών, τα οποία γίνονται διαθέσιμα στους χρήστες μέσω της αναβάθμισης του συστήματος κάθε μήνα.

Με βάση τις μεθοδολογίες που ακολουθούνται, και λόγω και του σχετικά μικρού μεγέθους της εφαρμογής αναφορών, για την συλλογή και τον προσδιορισμό των απαιτήσεων δεν ακολουθήθηκε κάποια αυστηρή διαδικασία με επίσημη διατύπωση που μετά δεν θα μπορούσε ή θα ήταν δύσκολο να αλλάξει. Πραγματοποιήθηκε μια σειρά συναντήσεων, συζητήσεων και ερευνών για να κατανοηθούν καλύτερα τα πιο σημαντικά και χρήσιμα χαρακτηριστικά, και στη συνέχεια με βάση και την ανάπτυξη της εφαρμογής και την δοκιμή και έλεγχο της, υπήρχε η δυνατότητα για αλλαγή των προτεραιοτήτων ή και ανάπτυξη νέων χαρακτηριστικών. Οι προδιαγραφές καταγράφηκαν με τη μορφή ιστοριών χρήστη, σύμφωνα με τις αρχές της ευέλικτης μεθοδολογίας.

### **2.2.1 Συνάντηση τρίμηνου σχεδιασμού EDMS**

Στη συνάντηση για τον τρίμηνο σχεδιασμό του EDMS, τον Ιανουάριο του 2017, με βάση παλαιότερα σχέδια αλλά και την όλο και εντονότερη συνειδητοποίηση της αναγκαιότητας του συστήματος αναφορών, αποφασίστηκε να ενταχθεί η ανάπτυξη του συστήματος αναφορών στους στόχους των επόμενων μηνών, με πλάνο την ολοκλήρωση της μέσα στο καλοκαίρι. Ταυτόχρονα, συζητήθηκαν για πρώτη φορά κάποιες γενικές ιδέες και χαρακτηριστικά. Αποφασίστηκε το προπαρασκευαστικό στάδιο να περιλαμβάνει δύο θέματα, ένα για την ανάλυση των απαιτήσεων της εφαρμογής, και ένα για την μελέτη των πιθανών τεχνολογικών λύσεων. Για την ανάλυση αποφασίστηκε να υπάρξει στενή συνεργασία με τους τοπικούς διαχειριστές, που αποτελούν τους πιο ενεργούς και προχωρημένους χρήστες με πολύπλοκα καθήκοντα, που θα ωφεληθούν πιο άμεσα από τις αναφορές.

### **2.2.2 Συνάντηση τοπικών διαχειριστών**

Στο πλαίσιο της συνάντησης των τοπικών διαχειριστών του EDMS, τον Φεβρουάριο του 2017, αναφέρθηκε η απόφαση για την υλοποίηση του συστήματος αναφορών, και ζητήθηκε η συμβολή τους για τον προσδιορισμό των χαρακτηριστικών της, με πολύ θετική ανταπόκριση. Στο επόμενο διάστημα, ορισμένοι από αυτούς έστειλαν προτάσεις για το τι στοιχεία θα θεωρούσαν χρήσιμο να έχουν στη διάθεση τους.

### **2.2.3 Καταγραφή αιτημάτων για δεδομένα**

Παράλληλα, πραγματοποιήθηκε ανάλυση όλων των αιτημάτων από χρήστες για πρόσβαση σε συγκεντρωτικά δεδομένα των προηγούμενων ετών (ανατρέχοντας στα σχετικά αιτήματα) και ξεκίνησαν να καταγράφονται τα τρέχοντα. Από την ανάλυση τους, προέκυψε ότι στην πράξη αφορούσαν ορισμένες παρόμοιες περιπτώσεις, αλλά με διαφορετικούς περιορισμούς. Έπειτα, εντοπίστηκε πια από αυτά είναι πιο πιθανό να είχαν έκτακτο χαρακτήρα (λόγω κάποιου έκτακτου ζητήματος που είχε προκύψει) και ποια επαναλαμβανόμενα, χρήσιμα σε σταθερή βάση.

## 2.2.4 Συναντήσεις με σημαντικούς χρήστες

Ακολούθως πραγματοποιήθηκαν συναντήσεις με τρεις σημαντικές ομάδες χρηστών, κάθε μία υπεύθυνη για κάποιο σημαντικό έργο στο EDMS, για να συζητηθούν περαιτέρω ειδικές απαιτήσεις που μπορεί να είχαν. Αυτές οι συναντήσεις ήταν:

- Με τους υπεύθυνους του τομέα Επιθεωρήσεων Ασφαλείας (Safety Inspections). Όλες οι επιθεωρήσεις ασφαλείας για τον εξοπλισμό και τα κτήρια του CERN καταχωρούνται στο EDMS, που προσφέρει επίσης ειδοποιήσεις για την ανανέωση τους και άλλες λειτουργίες. Συζητήθηκαν ειδικότερα θέματα και παράμετροι ώστε να μπορούν να έχουν συνολική εικόνα των επιθεωρήσεων ασφαλείας και όσων θα πρέπει να πραγματοποιηθούν το επόμενο διάστημα, για τον καλύτερο προγραμματισμό τους.
- Με την ομάδα Συνεργασίας και Διαδικασιών (Cooperation and Processes). Πρόκειται για ομάδα ατόμων με μεγάλη και ποικιλόμορφη εμπειρία στο τμήμα μηχανικών, με σκοπό την προώθηση της συνεργασίας μεταξύ των ομάδων και τον καθορισμών κοινών διαδικασιών για την διασφάλιση της ποιότητας των διαδικασιών και των δεδομένων. Συζητήθηκαν ιδιαίτερα θέματα σχετικά με την δυνατότητα χρήσης του νέου συστήματος για την προτροπή των χρηστών σε ορθότερη χρήση των διαδικασιών. Για παράδειγμα, συζητήθηκε η δημιουργία μιας προεπιλεγμένης αναφοράς που θα δείχνει στους χρήστες τα έγγραφα τους που δεν έχουν περάσει επίσημη διαδικασία έγκρισης ενώ δεν έχουν ανανεωθεί για κάποιους μήνες. Ακόμη τονίστηκε η χρησιμότητα του συστήματος στην ομάδα αυτή για τον έλεγχο και την προώθηση των σωστών διαδικασιών, καθώς θα διευκολυνόταν πολύ οι έλεγχοι που πραγματοποιούν, για παράδειγμα για το αν όλα τα έγγραφα σε κάποιο έργο ακολουθούν τις κατάλληλες διαδικασίες και άλλα παρόμοια θέματα.
- Με την τομέα διαμόρφωσης και διαρρύθμισης (Configuration and Layout). Ο τομέας αυτός είναι υπεύθυνος να διασφαλίσει ότι ανά πάσα στιγμή υπάρχει γνώση και σωστή αναπαράσταση της διάταξης και των εξαρτημάτων όλων των επιταχυντών, καθώς και πλήρες ιστορικό. Αυτό είναι εξαιρετικά σημαντικό, τόσο για την γνώση του επιταχυντή στην παρούσα μορφή (και άρα τη διευκόλυνση στο σχεδιασμό των εργασιών), όσο και στο να υπάρχει η δυνατότητα να προσδιοριστούν ακριβώς το υλικό και οι συνθήκες κάτω από τις οποίες έγινε οποιοδήποτε πείραμα και παρήχθησαν δεδομένα.

Για να γίνει οποιαδήποτε εργασία στους επιταχυντές (όπως αναβάθμιση εξαρτήματος ή προσθήκη νέων αισθητήρων, καλωδίων και άλλα) θα πρέπει να περάσει από μια επίσημη διαδικασία έγκρισης μέσω του EDMS, με συζήτηση με όλους τους τομείς που πιθανόν να επηρεάζονται, υπό την εποπτεία του τομέα διαμόρφωσης και διαρρύθμισης. Για το λόγω αυτό, στις περιόδους λίγο πριν και κατά την διάρκεια των τεχνικών στάσεων υπάρχει πολύ μεγάλος φόρτος αιτημάτων και στενές προθεσμίες, καθώς οποιαδήποτε καθυστέρηση μεταφράζεται αυτόματα σε καθυστέρηση έναρξης λειτουργίας του επιταχυντή. Συζητήθηκαν χαρακτηριστικά που θα τους επιτρέπουν καλύτερο έλεγχο και οργάνωση της δουλειάς τους, με έμφαση σε αυτές τις περιόδους τεχνικών στάσεων και αναβαθμίσεων.

- Με τους υπεύθυνους του τομέα Λειτουργίας του τμήματος Δεσμών, οι οποίοι είναι υπεύθυνοι για την έναρξη και τη λειτουργία των επιταχυντών. Από το 2017 χρησιμοποιούν τις επίσημες διαδικασίες του EDMS για να επιβεβαιώσουν ότι έχουν γίνει όλοι οι απαραίτητοι έλεγχοι και να δοθεί η άδεια ώστε να ξεκινήσει η λειτουργία των επιταχυντών του CERN. Ο τομέας αυτός χρησιμοποιεί ήδη τις διαδικτυακές υπηρεσίες του EDMS για να πάρει στοιχεία σχετικά με τα έγγραφα της έγκρισης και να τα παρουσιάζει σε μία δική του εφαρμογή, και συζητήθηκαν θέματα όπως ποια από τα στοιχεία τους είναι χρήσιμα, αν θα χρειαζόντουσαν κάποια αναφορά με συνολικά ή και ιστορικά στοιχεία, καθώς και τις δυνατότητες είτε για χρήση απευθείας της τελικής αναφοράς που θα παράγεται από το EDMS, είτε μόνο των δεδομένων της, για ενσωμάτωση στο δικό τους σύστημα.

### **2.2.5 Συγκέντρωση, ανάλυση και εσωτερική συζήτηση**

Στη συνέχεια το παραπάνω υλικό συγκεντρώθηκε, αναλύθηκε και συνοψίστηκε. Με βάση αυτά προετοιμάστηκε μια αναφορά, που τέθηκε προς εσωτερική συζήτηση στο τμήμα του EDMS. Στη συνέχεια, και αφού είχαν διατυπωθεί και οι απόψεις των υπόλοιπων μελών, σε νέα συνάντηση καθορίστηκαν οι βασικές απαιτήσεις του συστήματος.



## **2.3 Απαιτήσεις Εφαρμογής**

Οι απαιτήσεις της εφαρμογής, όπως προέκυψαν μετά τα παραπάνω είναι οι ακόλουθες (παρουσιάζονται με τη μορφή ιστορίας χρηστών, σύμφωνα με τις αρχές της ευέλικτης μεθοδολογίας ):

Ο χρήστης μπορεί να δημιουργήσει μια νέα αναφορά, επιλέγοντας τις παραμέτρους που θέλει.

Ο χρήστης μπορεί να αποθηκεύσει ή να διαγράψει μια υπάρχουσα αναφορά.

Ο χρήστης μπορεί να τροποποιήσει τις παραμέτρους της αναφοράς.

Ο χρήστης μπορεί να συνδυάσει περιορισμούς σε διαφορετικά πεδία των ιδιοτήτων των δεδομένων, με τη χρήση λογικών τελεστών, ώστε να προσαρμόσει την αναφορά στα δεδομένα που θέλει.

Η παραγόμενη αναφορά αποτελείται από διαγράμματα, καθώς και από πίνακες δεδομένων που περιλαμβάνουν όλα τα Documents, Projects και Items που περιλαμβάνονται στην αναφορά.

Ο χρήστης μπορεί να επιλέξει ποια διαγράμματα θα εμφανίζονται σε μία αναφορά.

Ο χρήστης μπορεί να επιλέξει ποιες στήλες θα φαίνονται στους πίνακες δεδομένων μιας αναφοράς.

Ο χρήστης μπορεί να εξάγει σε μορφή CSV τα δεδομένα της αναφορά από τους πίνακες δεδομένων.

Ο χρήστης μπορεί να ανοίξει την σελίδα της αντίστοιχης οντότητας από τους πίνακες δεδομένων μιας αναφοράς.

Ο χρήστης μπορεί να φιλτράρει τους πίνακες δεδομένων μιας αναφοράς, είτε σε επίπεδο στήλης, είτε συνολικά.

Ο χρήστης μπορεί να ταξινομήσει τον πίνακα δεδομένων μιας αναφοράς πατώντας στην επικεφαλίδα της αντίστοιχης στήλης.

Ο χρήστης ΔΕΝ μπορεί να δει στοιχεία για ευαίσθητα έγγραφα στα οποία δεν έχει πρόσβαση.

Ο διαχειριστής μπορεί να δημιουργήσει μια ή περισσότερες αναφορές διαθέσιμες σε όλους τους χρήστες για την προώθηση σωστών διαδικασιών.

Η εφαρμογή προειδοποιεί τον χρήστη αν αυτός δοκιμάσει να φύγει χωρίς να σώσει τις αλλαγές που πραγματοποίησε στις αναφορές.

Η εφαρμογή και οι αναφορές αποτελούνται από κατά το δυνατόν ανεξάρτητα συστατικά μέρη, που μπορούν να συνδυαστούν με διάφορους τρόπους και να επαναχρησιμοποιηθούν σε διάφορα σημεία του συστήματος EDMS.

Η εφαρμογή πρέπει να διατηρεί όσο το δυνατόν πιο ασφαλή τα δεδομένα, και κατ' ελάχιστον να είναι ασφαλής από επιθέσεις SQL Injection, Cross Site Scripting, Cross Site Request Forgery.

## **2.4 Επιλογή τεχνολογικής λύσης**

Για την επιλογή της τεχνολογικής λύσης, υπήρχαν δύο υποψήφιες κατηγορίες λύσεων: Η χρήση κάποιου εξωτερικού συστήματος Business intelligence (επιχειρηματικής ευφυΐας) με ισχυρότερο υποψήφιο το Pentaho) ή η ανάπτυξη μιας εφαρμογής αναφορών κατευθείαν στο EDMS.

Πραγματοποιήθηκε έρευνα για τα πλεονεκτήματα και μειονεκτήματα κάθε μεθόδου, τόσο ψάχνοντας κοινές και αποδεκτές πρακτικές σε γνωστά προϊόντα, όσο και σε συνεργασία με άλλους τομείς του τμήματος που έχουν ήδη κάποια μορφή αναφορών, για να επωφεληθούμε από την υπάρχουσα πείρα και την άποψη τους για το αν ταιριάζει στην περίπτωση χρήσης του EDMS.

### **2.4.1 Ανάλυση πρότασης εξωτερικού συστήματος (Pentaho)**

Η ανάλυση για το εξωτερικό σύστημα, ανέδειξε ως κύριο πλεονέκτημα του την δυνατότητα των χρηστών να δημιουργούν οι ίδιοι πολύ προχωρημένες αναφορές, να επεξεργάζονται και να φιλτράρουν τα στοιχεία τους, και όλα αυτά με ελάχιστη απαιτούμενη προγραμματιστική δουλειά, από τη στιγμή που ο κύβος δεδομένων θα καθοριζόταν σωστά από τη βάση δεδομένων του EDMS. Αυτό έδινε τη δυνατότητα στους προχωρημένους χρήστες να μπορούν να αναλύουν τα δεδομένα τους με μεγάλη δύναμη και άνεση, σε ένα περιβάλλον που ήταν ήδη γνωστό σε κάποιους λόγω χρήσης του από άλλη υπηρεσία του CERN.

Τα μειονεκτήματα ενός εξωτερικού συστήματος εντοπίστηκαν στα παρακάτω:

- Στο ότι η πλήρης έκδοση του λογισμικού (που διαθέτει τις παραπάνω δυνατότητες) είναι κλειστού κώδικα και επί πληρωμή (η βασική έκδοση αποτελεί ανοιχτό λογισμικό).
- Στο ότι θα απαιτούνταν το στήσιμο, η συντήρηση και η διαχείριση ενός σχετικού εξυπηρετητή, καθώς δεν υπάρχει κεντρική υπηρεσία στο CERN που να προσφέρει αυτή την υπηρεσία). Εδώ να σημειώσουμε ότι η διαχείριση όλων των εξυπηρετητών του EDMS γίνεται από το τμήμα IT του CERN, με αποτέλεσμα να μην υπάρχει ανάγκη κατανάλωσης πόρων της ομάδας για τη συντήρηση, αναβαθμίσεις ασφαλείας, επίλυση προβλημάτων και άλλα.

- Στο γεγονός ότι η πλειοψηφία των απλών χρηστών πιθανόν δεν θα το έβλεπε ή δεν θα έμπαινε στον κόπο να το χρησιμοποιήσει ποτέ, μειώνοντας έτσι τα οφέλη, και τις όποιες δυνατότητες ενίσχυσης σωστών διαδικασιών και συμπεριφορών στους χρήστες με την παρουσίαση σε αυτούς των κατάλληλων αναφορών.

### **2.4.2 Ανάλυση πρότασης εφαρμογής εντός του EDMS**

Το βασικό μειονέκτημα της ανάπτυξης μιας εφαρμογής αναφορών εντός του EDMS ήταν ότι απαιτούνταν περισσότερος χρόνος και πόροι για την ανάπτυξη της, και ιδιαίτερα για την ανάπτυξη της διεπαφής με το χρήστη, που σε ένα εξωτερικό σύστημα θα ήταν έτοιμα. Ταυτόχρονα, μια εφαρμογή που θα αναπτυσσόταν εντός ρεαλιστικών χρονικών ορίων από έναν προγραμματιστή που ασχολούταν ταυτόχρονα και με άλλα καθήκοντα, ήταν προφανές ότι θα είχε μόνο ορισμένα βασικά χαρακτηριστικά και όχι όλες τις δυνατότητες ενός πολύπλοκου συστήματος επιχειρηματικής ευφυΐας όπως το Pentaho.

Ωστόσο, τα πλεονεκτήματα ανάπτυξης μιας εφαρμογής από το ίδιο το τμήμα φάνηκε ότι ήταν αρκετά. Αρχικά το γεγονός ότι θα ενσωματωνόταν στις σελίδες του EDMS θα το έκανε άμεσα προσβάσιμο σε όλους τους χρήστες, θα αύξανε την χρησιμότητα του λόγω της διασύνδεσης του με τα άλλα τμήματα και θα βελτίωνε την εμπειρία χρήσης. Δεύτερον, δεν απαιτούνταν η απόκτηση αδειών και άλλων συμφωνιών, που ανάμεσα σε μεγάλους οργανισμούς μπορεί να γίνει πολύπλοκη και χρονοβόρα. Τέλος, η μελλοντική υποστήριξη και βελτίωση του θα ήταν πιο εύκολη, λόγω της σχετικής γνώσης των μελών της ομάδας (αποτελείται από μηχανικούς λογισμικού και όχι διαχειριστές βάσεων δεδομένων ή συστημάτων).

### 2.4.3 Απόφαση για ανάπτυξη εφαρμογής

Με βάση τα πλεονεκτήματα και τα μειονεκτήματα που αναφέρθηκαν παραπάνω, καθώς και την ανάλυση των απαιτήσεων, από τις οποίες προέκυψε ότι το μεγαλύτερο μέρος τους θα μπορούσε να ικανοποιηθεί από μια εφαρμογή αναφορών χωρίς τρομερά αυξημένη πολυπλοκότητα, αποφασίστηκε σε πρώτη φάση η ανάπτυξη της εφαρμογής εντός του EDMS. Στη συνέχεια, και αφού χρησιμοποιηθεί από τους χρήστες, υπάρξουν σχόλια και κριτικές, θα αξιολογηθούν αυτά ώστε να μελετηθεί αν χρειάζεται να δοθούν περισσότεροι πόροι για περαιτέρω ανάπτυξη της εφαρμογής, ή ακόμα και αν έχει νόημα να μελετηθεί και η πιθανότητα χρήσης του εξωτερικού συστήματος ως επιπρόσθετο εργαλείο για τους προχωρημένους χρήστες.

Αφού επιλέχθηκε η λύση της ανάπτυξης εφαρμογής εντός του EDMS, απαιτήθηκε επόμενη ανάλυση για το τι τεχνολογίες θα χρησιμοποιηθούν.

Στο κομμάτι του εξυπηρετητή και της βάσης δεδομένων, θα χρησιμοποιούνταν τα ήδη υπάρχοντα (Java, PL/SQL, Oracle βάση δεδομένων) για διατήρηση της συνοχής με τον υπάρχον κώδικα, καθώς και δυνατότητα χρήσης ήδη έτοιμων κομματιών, κυρίως όσον αφορά στο κομμάτι της αλληλεπίδρασης με τη βάση δεδομένων με χρήση Stored Procedures.

Στο κομμάτι της διεπαφής του χρήστη όμως, υπήρχαν πάλι δύο προτάσεις: Είτε χρήση των ήδη χρησιμοποιούμενων τεχνολογιών του EDMS (Google Web Toolkit με χρήση της βιβλιοθήκης Sencha EXT for GWT), είτε χρήση κάποιου framework της Javascript με βιβλιοθήκες για τα διαγράμματα.

Τελικά προτιμήθηκε η δεύτερη λύση, για τους εξής λόγους:

- Στο EDMS χρησιμοποιείται παλιότερη έκδοση του GWT, της οποίας η υποστήριξη θα λήξει στο άμεσο μέλλον. Όμως, επειδή η αναβάθμιση του στη νεότερη έκδοση απαιτεί δουλειά ισάξια με τη συγγραφή του σε κάτι καινούριο, δεν έχει αποφασιστεί ακόμα τι θα γίνει. Επίσης, το GWT δεν δείχνει να δέχεται την προσοχή και την υποστήριξη που είχε στο παρελθόν, έχει μικρότερη κοινότητα χρηστών χωρίς να προσελκύει νέους.
- Με τη χρήση Html και Javascript είναι πολύ πιο εύκολη η ενσωμάτωση της εφαρμογής ή τμημάτων της σε διάφορα σημεία ή και σε διαφορετικές εφαρμογές. Για παράδειγμα, υπάρχουν κάποιες εφαρμογές διασυνδεδεμένες με το EDMS στο CERN, και είναι πολύ πιο εύκολο να ενσωματώσουν στοιχεία μιας αναφοράς που τους αφορά στη σελίδα τους.
- Τέλος, επειδή τα τελευταία χρόνια η Javascript γνωρίζει μεγάλη άνοδο, και με την καθιέρωση κάποιων web frameworks (πλατφόρμες ανάπτυξη εφαρμογών διαδικτύου) αλλά και package managers (διαχειριστές πακέτων) έχει εκτοξευθεί η παραγωγικότητα της, θα είναι πιο εύκολο τόσο για την τωρινή ανάπτυξη, όσο και στο μέλλον να βρεθούν προγραμματιστές με τις σχετικές γνώσεις για τη συντήρηση της εφαρμογής.

Μετά από περαιτέρω έρευνα, αποφασίστηκε η εφαρμογή να αναπτυχθεί ως Single Page Application (Εφαρμογή Μίας Σελίδας) με χρήση της Typescript, του Angular framework, και των βιβλιοθηκών primeng και chart.js για τα στοιχεία της διεπαφής και τα διαγράμματα της εφαρμογής αντίστοιχα.

## 3. Τεχνολογίες που χρησιμοποιήθηκαν

Σε αυτό το κεφάλαιο παραθέτουμε τις τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής, ορισμένα βιβλιογραφικά στοιχεία για κάθε μία από αυτές, καθώς και κάποιους βασικούς λόγους για τους οποίους επιλέχθηκαν.

### 3.1 Front End

#### 3.1.1 Angular

Πρόκειται για ένα web framework βασισμένο στην typescript για την κατασκευή διαδικτυακών εφαρμογών μονής σελίδας. Είναι ανοιχτό λογισμικό, κάτω από MIT License, που η πρώτη του έκδοση δημιουργήθηκε το 2010 από μία ομάδα εργαζομένων στην Google ως ένα framework βασισμένο στην Javascript για την γρήγορη ανάπτυξη διαδικτυακών εφαρμογών, και έγινε εξαιρετικά δημοφιλές. Αναπτύσσεται συνεχώς, κυρίως με τη συμβολή της google και της κοινότητας ανοιχτού λογισμικού που έχει αναπτυχθεί γύρω του. Το 2015 βγήκε η δεύτερη έκδοση, στην οποία το framework ξαναγράφηκε από την αρχή και πλέον βασίζεται στην Typescript. Από τότε, η αρχική έκδοση ονομάζεται AngularJS, ενώ η δεύτερη και οι επόμενες εκδόσεις (που χρησιμοποιήθηκαν εδώ) απλώς Angular, για να ξεχωρίζουν τα δύο συστήματα (καθώς κομμάτι των χρηστών δεν ακολούθησε στην νέα έκδοση, λόγω των πολλών αλλαγών που απαιτούνταν για αναβάθμιση).

Το Angular χρησιμοποιεί την HTML ως template γλώσσα, και επιτρέπει την επέκταση της με καινούρια tag, που σηματοδοτούν τα συστατικά (components) της εφαρμογής. Με αυτό τον τρόπο διευκολύνει τον διαχωρισμό της εφαρμογής σε συστατικά, με καθαρό και συνοπτικό τρόπο, και επιτρέπει να συνδυάζονται με διαφορετικούς τρόπους.

Ένα άλλο πολύ χρήσιμο χαρακτηριστικό του είναι το data binding, δηλαδή ο αυτόματος συγχρονισμός των δεδομένων ανάμεσα στο model και στο view (το μοντέλο που περιέχει τα δεδομένα και την οπτική παρουσίαση του στο χρήστη), κάτι που διευκολύνει πολύ την ανάπτυξη εφαρμογών που περιέχουν δεδομένα που αλλάζουν συχνά ή εξαρτώνται από επιλογές του χρήστη. Για παράδειγμα, αν έχουμε μια λίστα τιμών να εμφανίζεται σε ένα μενού, η οποία εξαρτάται από μία προηγούμενη επιλογή του χρήστη, με το που αλλάξει η προηγούμενη επιλογή του χρήστη, έχουμε αυτόματο συγχρονισμό και εμφάνιση των νέων δεδομένων στη λίστα μας.

Επιλέχθηκε διότι πρόκειται για ένα σύγχρονο framework, με συνεχή ανάπτυξη και μεγάλη κοινότητα χρηστών, που προσφέρει πολλές ευκολίες και βελτιώνει την παραγωγικότητα. Ταυτόχρονα, το γεγονός ότι βασίζεται σε όσο το δυνατόν ανεξάρτητα components, που χρησιμοποιούν services για να πάρουν τα δεδομένα τους ταιριάζει πολύ στην υπό ανάπτυξη εφαρμογή. Αυτό προκύπτει από το ότι θέλουμε οι αναφορές να μπορούν να προσαρμόζονται, να μπορούμε να συνδυάζουμε διαγράμματα και πίνακες δεδομένων με διάφορους τρόπους, και να παρουσιάζουμε δεδομένα στο χρήστη που αλλάζουν με τις επιλογές του.



### 3.1.2 Typescript

Πρόκειται για μια σχετικά νέα γλώσσα προγραμματισμού (οι πρώτες δοκιμαστικές εκδόσεις τις κυκλοφόρησαν το 2012, ενώ η πρώτη της έκδοση, 1.0 το 2014). Δημιουργήθηκε από τη Microsoft και είναι λογισμικό ανοιχτού κώδικα. Δείχνει να έχει αρκετά έντονη αυξητική τάση ανάμεσα στους προγραμματιστές εφαρμογών διαδικτύου.

Παρότι δεν πρόκειται για επίσημη έρευνα με αντιπροσωπευτικό δείγμα, σύμφωνα με την έρευνα της δημοφιλέστατης σε προγραμματιστές ιστοσελίδας [stackoverflow.com](https://stackoverflow.com) για το 2017 σε 64.000 προγραμματιστές, η Typescript ήταν η ένατη σε χρήση προγραμματιστική γλώσσα με 9.5%, ξεπερνώντας αρκετές άλλες γνωστές όπως Ruby, Swift, Objective-C. Βρίσκεται σε συνεχή ανάπτυξη, με σχεδόν μηνιαίες αναβαθμίσεις της γλώσσας, που διατηρούν βέβαια τη συμβατότητα με τις παλαιότερες εκδόσεις

Η Typescript είναι αυστηρό **υπερσύνολο** της Javascript, και στην ουσία πρόκειται για Javascript με την προσθήκη ενός προαιρετικού στατικού συστήματος τύπων και κάποιων άλλων βελτιώσεων. Η προσθήκη στατικού συστήματος τύπων στην Javascript γίνεται για να αντιμετωπίσει τα συχνά προβλήματα που παρουσιάζονταν σε μεγαλύτερες εφαρμογές λόγω του δυναμικού συστήματος τύπων της Javascript, καθώς όσο μεγαλώνει ο όγκος τόσο πιο εύκολο είναι να γίνουν λάθη στους τύπους των δεδομένων και στις μεταξύ τους μετατροπές. Δεδομένου ότι η Typescript είναι υπερσύνολο της Javascript, κάθε έγκυρο πρόγραμμα Javascript είναι έγκυρο πρόγραμμα Typescript. Αυτό διευκολύνει αρκετά την ανάπτυξη και διάδοση της, καθώς κάνει αρκετά εύκολη την συμβατότητα με τις χιλιάδες υπάρχουσες βιβλιοθήκες της Javascript, ενώ όποιος γνωρίζει Javascript είναι πολύ εύκολο να μάθει Typescript. Η Typescript εν τέλει μεταγλωττίζεται σε Javascript, που στη συνέχεια εκτελείται από τους browser (φυλλομετρητές).

Επιλέχθηκε επειδή το στατικό σύστημα τύπων της προσφέρει μεγαλύτερη ασφάλεια και εκφραστικότητα στον κώδικα από ότι η javascript. Αυτό είναι ιδιαίτερα χρήσιμο στην εφαρμογή που αναπτύχθηκε, καθώς οι τύποι των δεδομένων όπως λαμβάνονται από τον εξυπηρετητή και επεξεργάζονται για να παρουσιαστούν στις αναφορές είναι ένα από τα πιο πιθανά σημεία σφαλμάτων. Ακόμη, σημαντικό ρόλο στην επιλογή της έπαιξε το γεγονός ότι πρόκειται για την επίσημα προτεινόμενη γλώσσα για χρήση με το framework Angular.

### 3.1.3 Primeng

Πρόκειται για μία συλλογή γραφικών στοιχείων (rich UI components) για το Angular. Είναι λογισμικό ανοιχτού κώδικα, και προσφέρει μια σειρά γραφικών στοιχείων παρόμοιας αισθητικής που μπορούν να συνδυαστούν για την δημιουργία διεπαφής χρήστη. Τα στοιχεία αναπτύσσονται από την ίδια ομάδα και βασίζονται στο framework PrimeFaces, ένα από τα πιο δημοφιλή UI Frameworks της Java EE, και διαθέτει μεγάλη κοινότητα χρηστών. Επιλέχθηκαν γιατί έχουν όλα τα βασικά στοιχεία απαραίτητα για τη δημιουργία της εφαρμογής αναφορών, έχουν μεγάλη κοινότητα και υποστήριξη εδώ και χρόνια, ενώ είναι και αρκετά «οικεία» σε αρκετούς χρήστες λόγω της συχνής χρήσης τους σε άλλες εφαρμογές (ανάμεσα τους και εφαρμογές διασυνδεδεμένες με το EDMS).

### 3.1.4 Chart.js

Πρόκειται για μια από τις δημοφιλέστερες βιβλιοθήκες διαγραμμάτων ανοιχτού κώδικα για τη Javascript. Περιλαμβάνει 8 τύπους διαγραμμάτων, και είναι πολύ δημοφιλής για την απλότητα της, την ευκολία χρήσης και την ανταπόκριση της. Επιλέχθηκε γιατί χρησιμοποιείται ήδη για τα στοιχεία των διαγραμμάτων από την Primeng, ενώ καλύπτει όλα τα πιθανά διαγράμματα για την εφαρμογή, καθώς ως τώρα δεν έχουν προκύψει ιδιαίτερα εξεζητημένες απαιτήσεις σχετικά με τα διαγράμματα.

## **3.2 Back End**

### **3.2.1 Java EE**

Η Java EE ( Java Platform, Enterprise Edition ) αποτελεί μία από τις γνωστότερες πλατφόρμες ανάπτυξης εφαρμογών για επιχειρησιακό λογισμικό. Επεκτείνει την Java SE (Java Standard Edition) με APIs (Application Programming Interface – Διεπαφή Προγραμματισμού Εφαρμογών) χρήσιμα σε μεγάλες επιχειρησιακές εφαρμογές, για παράδειγμα σχετικά με καταναμημένα συστήματα, διαδικτυακές υπηρεσίες και άλλα. Ο εξυπηρετητής του EDMS είναι γραμμένος με αυτήν την πλατφόρμα, και γι' αυτό το λόγο χρησιμοποιείται και για το κομμάτι του εξυπηρετητή της εφαρμογής αναφορών.

### **3.2.2 PL/SQL**

Η PL/SQL (Procedural Language extension for Structured Query Language) είναι μία δομημένη γλώσσα προγραμματισμού που επεκτείνει τις δυνατότητες της SQL στις βάσεις δεδομένων της Oracle. Με τον συνδυασμό στοιχείων του δομημένου προγραμματισμού (διακλαδώσεις, επαναληπτικούς βρόγχους, συλλογές και άλλα) και της SQL, δίνει την δυνατότητα για εκτέλεση ερωτημάτων στη βάση δεδομένων και επεξεργασία των απαντήσεων. Τα τμήματα αυτά του κώδικα αποθηκεύονται στη βάση δεδομένων ως Stored Procedures (Αποθηκευμένες Διαδικασίες), και στη συνέχεια μπορούν να κληθούν με το όνομά τους. Καθώς ήδη στο EDMS η αλληλεπίδραση με τη βάση δεδομένων γίνεται με χρήση Stored Procedures οι οποίες καλούνται από τον εξυπηρετητή που τρέχει τον κώδικα Java, διατηρήθηκε η ίδια λογική για συνέπεια αλλά και επαναχρησιμοποίηση υπάρχοντων διαδικασιών.

### **3.2.3 Oracle SQL**

Πρόκειται για την SQL (Structured Query Language – Δομημένη Γλώσσα Ερωτημάτων) που χρησιμοποιείται στις βάσεις δεδομένων της Oracle. Πρόκειται για την βασική γλώσσα αλληλεπίδρασης με σχεσιακές βάσεις δεδομένων, και επιτρέπει την εύκολη και γρήγορη πρόσβαση και ανάλυση των δεδομένων της βάσης.

## 3.3 Επικοινωνία Πελάτη – Εξυπηρετητή

### 3.3.1 JSON

Το JSON (Javascript Object Notation) είναι ένα ελαφρύ πρότυπο ανταλλαγής δεδομένων, που ξεκίνησε από τη Javascript, αλλά πλέον είναι το πιο διαδεδομένο πρότυπο ανταλλαγής δεδομένων στις εφαρμογές διαδικτύου. Υποστηρίζεται από όλες τις γλώσσες, είναι εύκολο να αναλυθεί αλλά και να παραχθεί από τις μηχανές, και ταυτόχρονα είναι εύκολα κατανοητό και από τους ανθρώπους, κάτι που συνέβαλε πολύ στη διάδοσή του.

Βασίζεται σε δύο πολύ βασικές δομές δεδομένων: Μια συλλογή από ζευγάρια ονομάτων / τιμών (που στις διάφορες γλώσσες μπορεί να ερμηνεύεται ως λεξικό, πίνακας hash, λίστα κλειδιών, και άλλα), και μια ταξινομημένη λίστα τιμών (που ερμηνεύεται ως πίνακας, διάνυσμα, λίστα ή ακολουθία).

Η απλότητα του, η ελάχιστη επιβάρυνση που προκαλεί και η καθολική του υποστήριξη το κάνουν την προφανή επιλογή για την ανταλλαγή δεδομένων στην εφαρμογή μας. Ένα παράδειγμα JSON:

```
{  "documentList":
    [  {
        "id": "1051100",
        "version": "2",
        "status": "Under Approval",
        "title": "Building 112 Safety Inspection",
        "author": "Christofe Delamare",
        "releaseProcedure": "DOC-OWNER",
    }, {
        "id": "1051100",
        "version": "2",
        "status": "Under Approval",
        "title": "Building 112 Safety Inspection",
        "author": "Christofe Delamare",
        "releaseProcedure": "DOC-OWNER",
    }
  ]
}
```

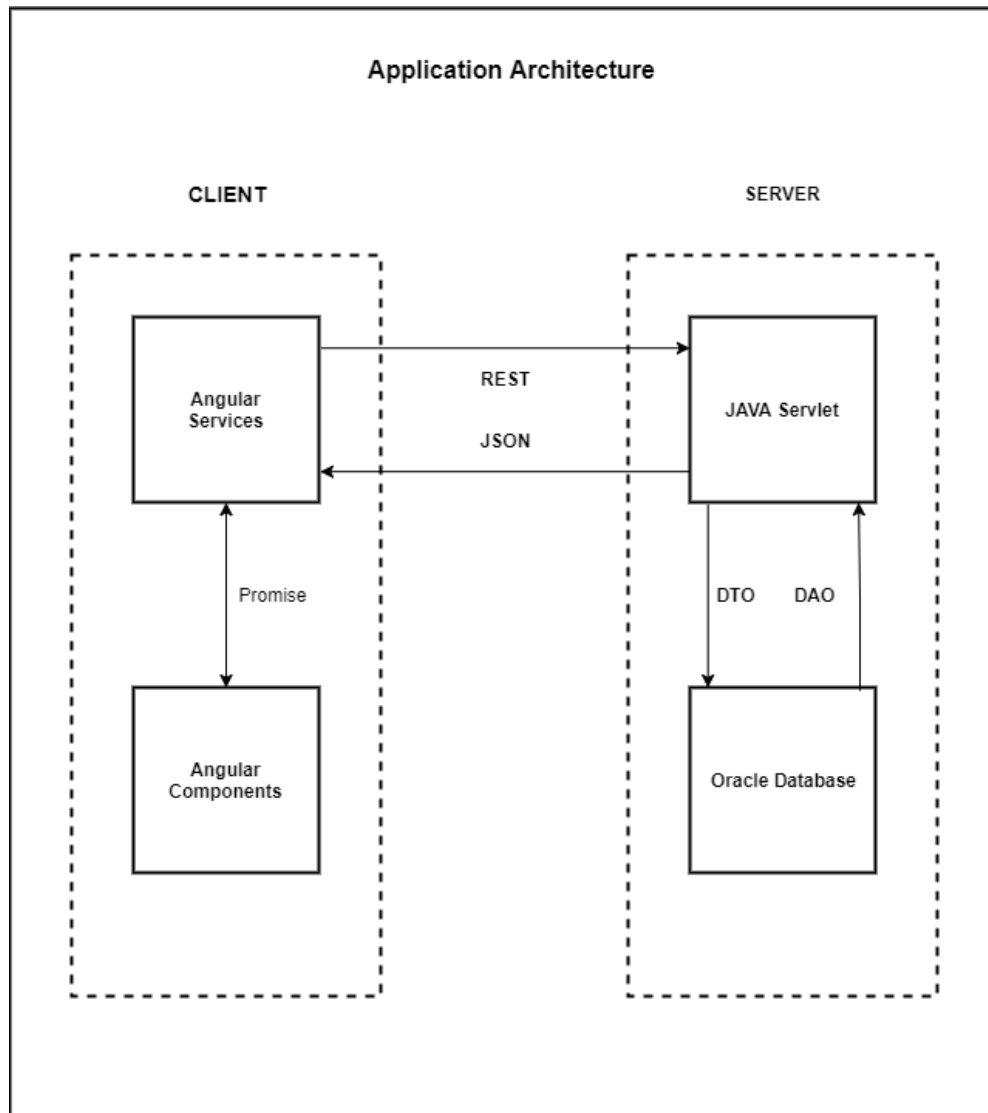
## RESTful Διαδικτυακές Υπηρεσίες

Οι διαδικτυακές υπηρεσίες REST (REpresentational State Transfer) είναι ένας τρόπος διασύνδεσης υπολογιστικών συστημάτων, που υπακούει σε κάποιους αρχιτεκτονικούς περιορισμούς. Οι βασικές αρχές του REST είναι:

- Χρήση δομημένων και εύκολα κατανοητών URI (Uniform Resource Identifier) για την πρόσβαση σε πόρους. Για παράδειγμα /user/Tom
- Χρήση JSON ή XML για την αναπαράσταση δομών δεδομένων και ιδιοτήτων
- Χρήση των μεθόδων του πρωτοκόλλου HTTP (GET, POST, PUT, DELETE) για την δήλωση μιας ενέργειας σε κάποιον πόρο.
- Statelessness (Έλλειψη κατάστασης): Κάθε κλήση είναι ξεχωριστή και ανεξάρτητη, και δεν μπορεί να αναφέρεται σε κάποια προηγούμενη.

Τα πλεονεκτήματα της REST αρχιτεκτονικής για τις διαδικτυακές υπηρεσίες είναι πολλά, καθώς επιτρέπουν την χρήση εύκολων και κατανοητών URI για την πρόσβαση σε δεδομένα, ενώ η έλλειψη κατάστασης και η προβλεψιμότητα των αιτημάτων προσφέρουν δυνατότητες απόδοσης στους εξυπηρετητές, μέσω χρήσης κατανεμημένων συστημάτων και παραπάνω από έναν εξυπηρετητές, caching των απαντήσεων και άλλα.

### 3.4 Αρχιτεκτονική Εφαρμογής



Το παραπάνω διάγραμμα μας δείχνει την γενική αρχιτεκτονική της εφαρμογής μας.

Συγκεκριμένα, στην πλευρά του πελάτη, έχουμε τα Components που δημιουργήσαμε με την Angular, και συνδυάζοντας τα μας δίνουν το γραφικό περιβάλλον της εφαρμογής. Αυτά παίρνουν τα δεδομένα που χρειάζονται από τα Services της Angular που δημιουργήσαμε, ασύγχρονα. Τα services πραγματοποιούν HTTP Requests στο RESTful API που έχουμε δημιουργήσει.

Το Servlet γραμμένο σε Java λαμβάνει τα requests, τα αναλύει και τα επεξεργάζεται, χρησιμοποιεί Data Access Objects και Data Transfer Objects για να καλέσει Stored Procedures της βάσης που θα εκτελέσουν τα queries και να πάρει τα αποτελέσματα, και στη συνέχεια τα μετατρέπει σε JSON που τα στέλνει ως απάντηση στα Services, τα οποία ειδοποιούν τα αντίστοιχα components ώστε να ανανεωθούν τα απαραίτητα γραφικά στοιχεία της εφαρμογής.

Το API το οποίο χρησιμοποιεί η εφαρμογή για την επικοινωνία μεταξύ πελάτη – εξυπηρετητή είναι το εξής:

#### **GET:**

/reportsApp/releaseProcedures

/reportsApp/documentTypes

/reportsApp/equipmentCodes

/reportsApp/contexts

/reportsApp/context/{context.id}

/reportsApp/context/{context.id}/releaseProcedures

/reportsApp/context/{context.id}/documentTypes

/reportsApp/context/{context.id}/equipmentCodes

/reportsApp/reports

/reportsApp/generateReport/{reportType}/{reportQuery}

#### **GET/PUT/DELETE:**

/reportsApp/report/{report.name}





## 4. Λειτουργία Εφαρμογής

Σε αυτό το κεφάλαιο περιγράφεται η λειτουργία της εφαρμογής που αναπτύχθηκε. Εξετάζεται η γενική λειτουργία της εφαρμογής και οι δυνατότητες που παρέχονται στους χρήστες. Στη συνέχεια παρουσιάζονται πιο αναλυτικά τα βασικά γραφικά στοιχεία και οι οθόνες της εφαρμογής καθώς και η συγκεκριμένη λειτουργία που επιτελούν. Στο τέλος παρατίθεται ένα ενδεικτικό παράδειγμα χρήσης της εφαρμογής.

### 4.1 Γενική Λειτουργία Εφαρμογής

Η βασική λειτουργία της εφαρμογής συνίσταται στη δημιουργία, παρουσίαση και τροποποίηση αναφορών σχετικών με τα δεδομένα που είναι αποθηκευμένα στο EDMS. Μια αναφορά μπορεί να αφορά μια οντότητα (Project, Item, Document), ένα Equipment Code ή ένα Context. Κάθε αναφορά μπορεί να εμπεριέχει κάποια διαγράμματα καθώς και κάποιους πίνακες δεδομένων. Ο κάθε πίνακας δεδομένων αφορά μία οντότητα (Project, item, Document). Έτσι λοιπόν, μπορούν να δημιουργηθούν οι εξής τύποι αναφορών:

- Project / Items / Documents, που αναφέρεται σε όλα τα project / items / documents με συγκεκριμένα χαρακτηριστικά, περιλαμβάνει διαγράμματα για αυτά, καθώς και έναν πίνακα δεδομένων με όλα τα project / items / documents που περιλαμβάνονται.
- Context / Equipment Code, που αναφέρεται σε ένα συγκεκριμένο context ή equipment code, περιλαμβάνει διαγράμματα είτε για αυτό είτε για τα projects / documents / items που ανήκουν σε αυτό, καθώς από έναν πίνακα δεδομένων για τα projects / documents / items που ανήκουν σε αυτό.

Την πρώτη φορά χρήσης της εφαρμογής, ο χρήστης βλέπει αυτόματα μια αναφορά που αφορά τα δικά του Documents, με ορισμένα χρήσιμα στοιχεία (όπως σε τι Status της Release Procedure βρίσκονται) που σκοπό έχουν να παρουσιάσουν μια βασική εικόνα για τους απλούς χρήστες, και ίσως να τους ωθήσουν να ολοκληρώσουν ξεχασμένες υποχρεώσεις, έγγραφα που έχουν μείνει σε λάθος στάδιο και άλλα. Το τι περιλαμβάνει αυτή η αναφορά καθορίζεται από τους διαχειριστές του EDMS.

Στη συνέχεια, ο χρήστης έχει τη δυνατότητα να τροποποιήσει τις παραμέτρους αυτής της αναφοράς ή να δημιουργήσει μια νέα. Αν δημιουργήσει νέα, αυτή θα δημιουργηθεί σε νέα καρτέλα εντός της εφαρμογής, ώστε να έχει ο χρήστης τη δυνατότητα εύκολης εναλλαγής ανάμεσα σε πολλές αναφορές. Και στις δύο περιπτώσεις (νέα ή τροποποίηση) καλείται να επιλέξει τον τύπο της αναφοράς (από αυτούς που αναφέρθηκαν παραπάνω) και τις παραμέτρους της αναφοράς, περιορίζοντας τα αποτελέσματα με βάση κάποιες από τις ιδιότητες τους.

Για παράδειγμα μπορεί να περιορίσει σε συγκεκριμένο χρονικό διάστημα δημιουργίας, υπό συγκεκριμένους συγγραφείς και άλλα. Το σύστημα επιλογής παραμέτρων αποτελεί μία λογική πρόταση, που δημιουργείται με επιλογή μια ιδιότητας και τιμών από το χρήστη, και σύνδεση τους με λογικούς τελεστές. Έτσι οι απλοί χρήστες μπορούν να χρησιμοποιήσουν απλώς τον λογικό τελεστή ΚΑΙ ώστε να δημιουργήσουν απλές αναφορές που ικανοποιούν όλες τις παραμέτρους, ενώ παρέχεται και η δυνατότητα δημιουργίας πιο περίπλοκων προτάσεων με χρήση και του τελεστή Ή και εμφώλευση έως δύο επίπεδα, ώστε οι πιο προχωρημένοι χρήστες να έχουν μεγαλύτερη ευελιξία.

Ενώ βλέπει μία αναφορά, ο χρήστης μπορεί να τροποποιήσει τη μορφή της, επιλέγοντας πια διαγράμματα από τα προσφερόμενα και ποιες στήλες των πινάκων δεδομένων θέλει να εμφανίζονται, ενώ μπορεί και να φιλτράρει οποιαδήποτε στήλη του πίνακα δεδομένων για να δει μόνο τα σχετικά έγγραφα.

Στη συνέχεια μπορεί αν θέλει να αποθηκεύσει την αναφορά, ώστε να μπορεί να την ξανατρέξει εύκολα την επόμενη φορά. Επίσης μπορεί να διαγράψει μία αναφορά, αν δεν θέλει να εμφανίζεται τις επόμενες φορές που θα συνδεθεί στο σύστημα.

## 4.2 Συστατικά - Οθόνες της εφαρμογής

### 4.2.1 Επιλογή Παραμέτρων

My Documents Report Safety Context B3 Report Documents Under Approval Rejected ECRs 2016 LS1

▼ Report Parameters

Create report on  Documents  Projects  Items  Context  Equipment Code

Select Parameters

Context IS EN-ACE-PUBLIC AND

Creation Date BEFORE 04/11/2017 + Add Parameter

Generate Report

Charts

Data tables

April 2017

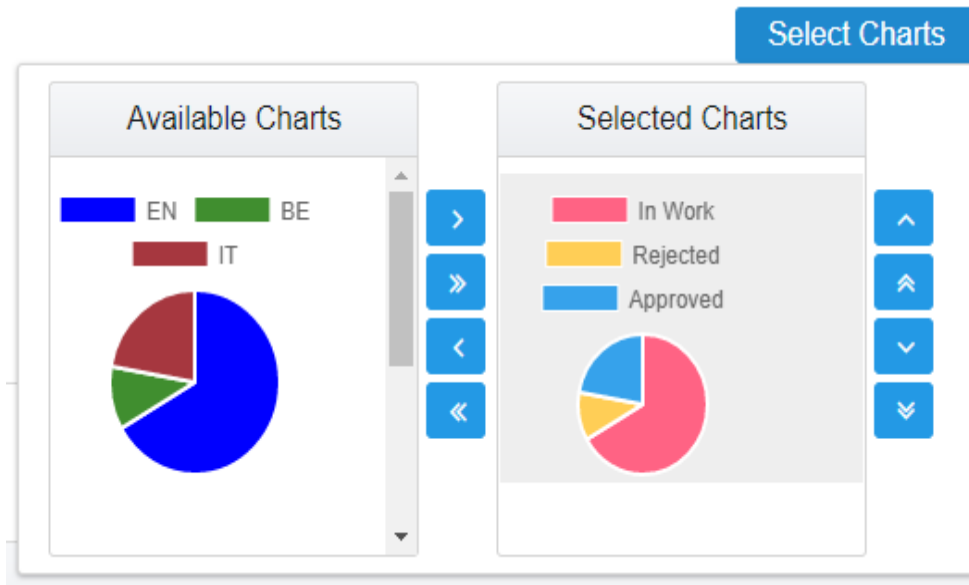
Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

Today Clear

Σε αυτό το στοιχείο, ο χρήστης καλείται να επιλέξει τις παραμέτρους της αναφοράς του. Αρχικά επιλέγει τον τύπο της αναφοράς (Documents, Projects, Items, Context, Equipment Code). Στη συνέχεια μπορεί να προσθέτει επιπλέον παραμέτρους για να φιλτραριστούν τα αποτελέσματα. Πατώντας το κουμπί “Add Parameter” προστίθεται μια νέα σειρά. Σε αυτήν ο χρήστης επιλέγει αρχικά την παράμετρο (πχ Title, Owner, Description, Context, Release Procedure, Creation Date, Update Date).

Στη συνέχεια επιλέγει ποιον τελεστή θέλει να χρησιμοποιήσει (IS, IS NOT, IN, NOT IN, Contain, Starts With, Ends With, BEFORE, AFTER, More Than, Less Than). Έπειτα, επιλέγει την τιμή. Το input που θα παρουσιαστεί στο χρήστη εξαρτάται από τον τελεστή που χρησιμοποιείται. Για παράδειγμα, αν χρησιμοποιούνται οι τελεστές Before / After, που αναφέρονται σε ημερομηνία, θα εμφανιστεί ένα ημερολόγιο. Αν επιλέξει IS / IS NOT θα εμφανιστεί dropdown menu. Αν επιλέξει IN / NOT IN θα εμφανιστεί dropdown menu με checkboxes για πολλαπλή επιλογή. Τέλος, αν επιλέξει κάποιο από τα Contain/Starts With/ Ends With / More than / Less Than θα εμφανιστεί ένα απλό text input για να γράψει την τιμή που θέλει.

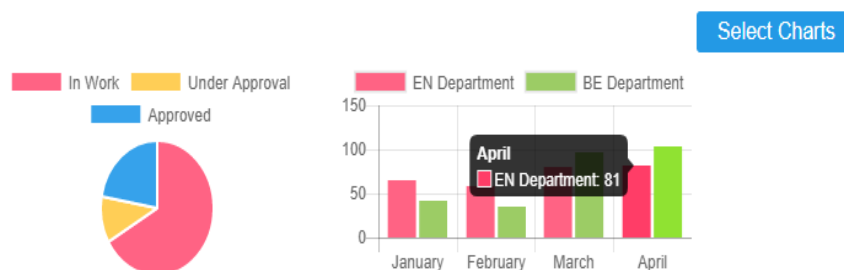
## 4.2.2 Επιλογή Διαγραμμάτων



Σε αυτό το στοιχείο ο χρήστης μπορεί να επιλέξει ποια από τα διαθέσιμα διαγράμματα θέλει να εμφανίζονται στην αναφορά του, καθώς και τη σειρά με την οποία θα εμφανίζονται. Αριστερά εμφανίζονται όλα τα διαθέσιμα διαγράμματα, και δεξιά τα επιλεγμένα. Μπορεί να επιλέξει ένα ή περισσότερα διαγράμματα (με τη χρήση των κουμπιών control και shift για μεμονωμένη ή μαζική επιλογή), και να τα μεταφέρει από τη μία λίστα στην άλλη με τα κουμπιά ή με Drag and Drop (μεταφορά και απόθεση).

## 4.2.3 Παρουσίαση Διαγραμμάτων


▼ [Charts](#)



Σε αυτό το στοιχείο ο χρήστης μπορεί να δει όσα διαγράμματα έχει επιλέξει στην αναφορά. Επίσης από εδώ μπορεί να πατήσει το κουμπί για να εμφανιστεί το στοιχείο επιλογής διαγραμμάτων αναφοράς (Select Charts).

## 4.2.4 Πίνακας Δεδομένων

Search All

 CSV

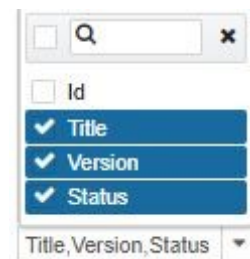
Your Document Report

Title	Version	Status
saf	Filter by Version	Under Approval
Building 112 Safety Inspection	2	Under Approval
Safety Commission Usage of CDD Presentati	A	Under Approval
Safety is our top priority	A	Under Approval

Select Columns

Σε αυτό το στοιχείο ο χρήστης βλέπει έναν πίνακα δεδομένων για μία οντότητα, ενώ μπορεί να επιλέξει ποιες στήλες θέλει να φαίνονται, και να φιλτράρει τα στοιχεία με βάση οποιαδήποτε στήλη.

Επίσης, μπορεί να εξάγει τα δεδομένα του πίνακα δεδομένων σε μορφή CSV (Comma Separated Values) για αποθήκευση και προσωπική του χρήση.



## 4.2.5 Βασική Οθόνη Εφαρμογής

The screenshot displays the application's main interface. At the top right, there are three buttons: '+ New' (blue), 'Save' (green), and 'Delete' (red). Below these are five navigation tabs: 'My Documents Report', 'Safety Context B3 Report', 'Documents Under Approval', 'Rejected ECRs', and '2016 LS1'. The left sidebar contains a menu with 'Report Parameters', 'Charts', and 'Data tables' (expanded). A search bar labeled 'Search All' is present. Below the search bar is a 'CSV' button. The main content area is titled 'Documents Included in the Report' and contains a table with columns for Title, Version, and Status. Each column has a corresponding filter input field. The table lists several documents with their respective versions and statuses.

Title	Version	Status
Building 112 Safety Inspection	2	Under Approval
Safety Commission Usage of CDD Present	A	Under Approval
Safety is our top priority	A	Under Approval
sharedFileNote	A	Approved
sharedFileNote	AB	Obsolete
sharedFileNote	AC	Obsolete
testname	1	In Work

Αυτή είναι η βασική οθόνη της εφαρμογής, η οποία αποτελείται από τα γραφικά στοιχεία που αναφέρθηκαν προηγουμένως. Σε αυτή την οθόνη υπάρχουν τα γραφικά στοιχεία Επιλογής Παραμέτρων, Παρουσίασης Διαγραμμάτων και Πίνακα Δεδομένων. Τα στοιχεία αυτά είναι συμπτυσσόμενα, ώστε ο χρήστης να μπορεί να βλέπει μόνο όποιο θέλει ανοιχτό. Επίσης, ο χρήστης βλέπει καρτέλες με τις υπόλοιπες αναφορές που έχει δημιουργήσει (αν υπάρχουν), καθώς και κουμπιά για τη δημιουργία νέας αναφοράς και την αποθήκευση ή διαγραφή της παρούσας.

## 4.3 Παράδειγμα Χρήσης

Ένας χρήστης χρησιμοποιεί για πρώτη φορά την εφαρμογή της αναφοράς. Ο χρήστης βλέπει την προεπιλεγμένη αναφορά που έχουν δημιουργήσει οι διαχειριστές, στην οποία βλέπει ορισμένα στοιχεία για τα Documents που έχει δημιουργήσει ο ίδιος.

▼ Data tables

Search All

CSV

Documents Included in the Report

Id	Title	Version	Status
1059709	wipoutbeamCONTEXT	A	In Work
1059714	specification	A	Approval Rejected
1059715	precode	A	In Work
1059717	beam4	1	BE DSO Test
1059718	beam5	1	Patrol Sectors
1059735	test1	A	In Work
1059736	vfdvbf	A	In Work

Select Columns

Στη συνέχεια αποφασίζει ότι θέλει να δημιουργήσει μια νέα αναφορά για να δει τα έγγραφα που δεν έχουν εγκριθεί ακόμα στο Context για το οποίο είναι υπεύθυνος. Πατάει το κουμπί New και εμφανίζεται το γραφικό στοιχείο επιλογής παραμέτρων αναφοράς. Αφού διαλέξει τον τύπο της αναφοράς (Report Type: Documents), θέλει να προσθέσει τις παραμέτρους της αναφοράς. Συγκεκριμένα θέλει να επιλέξει τα Documents που:

Έχουν Context “EN-ACE-PUBLIC” ΚΑΙ

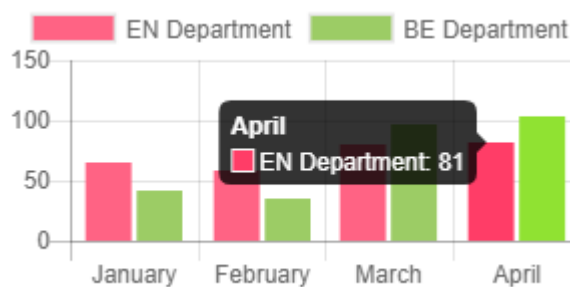
Δημιουργήθηκαν Πριν τις 04/11/2017

Πατάει λοιπόν το κουμπί Add Parameter. Δημιουργείται μια νέα σειρά από κάτω. Στο πρώτο drop down menu, από τις διαθέσιμες παραμέτρους επιλέγει Context. Στη συνέχεια, από το dropdown menu με τους τελεστές που μόλις εμφανίστηκαν επιλέγει IS, και από τη λίστα με τα contexts που εμφανίζεται επιλέγει αυτό που θέλει (EN-ACE-PUBLIC). Στη συνέχεια πατάει το κουμπί Add Parameter ξανά, και δημιουργείται νέα σειρά, ενώ παρατηρεί ότι στην πρώτη προστέθηκε ο λογικός τελεστής AND, που είναι ο επιθυμητός (αν δεν ήταν, θα μπορούσε να αλλαχθεί σε OR).

Αφού προσθέσει την παράμετρο σχετικά με την ημερομηνία δημιουργίας με τον ίδιο τρόπο, πατάει το κουμπί Generate Report, και η εφαρμογή δημιουργεί την αναφορά.

Σε αυτό το σημείο ο χρήστης σώζει την αναφορά πατώντας το κουμπί Save, γιατί θεωρεί ότι είναι μια αναφορά που θα του ξαναχρειαστεί.. Στη συνέχεια ο χρήστης επιλέγει το στοιχείο παρουσίασης διαγραμμάτων, και στη συνέχεια το κουμπί Select Charts ώστε να επιλέξει ποια διαγράμματα θέλει να εμφανίζονται.

Αφού επιλέξει το αυτό που δείχνει τα DOCUMENTS ανά τμήμα για κάθε μήνα, επιβεβαιώνει την επιλογή και βλέπει το νέο διάγραμμα.





Στη συνέχεια, πηγαίνει στο στοιχείο του πίνακα δεδομένων, και φιλτράρει ώστε να εμφανίζονται μόνο τα Documents που στον τίτλο περιλαμβάνουν τη λέξη “Safety” και βρίσκονται στο status “Under Approval”. Έτσι μπορεί να δει άμεσα αν υπάρχουν εκκρεμή έγγραφα σχετικά με την ασφάλεια που θα πρέπει να επισπευσθεί η διαδικασία έγκρισης τους. Για να εστιάσει σε όσες πληροφορίες τον ενδιαφέρουν μόνο, επιλέγει να εμφανίζονται μόνο το όνομα, η έκδοση και το status του Document.

Search All

CSV

Your Document Report

Title	Version	Status
saf	Filter by Version	Under App
Building 112 Safety Inspection	2	Under Approval
Safety Commission Usage of CDD Presentati	A	Under Approval
Safety is our top priority	A	Under Approval

Select Columns

Πατώντας στο όνομα, ανοίγει σε νέα καρτέλα το έγγραφο, για να ελέγξει κάποιες λεπτομέρειες, και μετά επιστρέφει. Τέλος, πατώντας στο κουμπί CSV, εξάγει τα δεδομένα και τα αποθηκεύει σε μορφή CSV, ώστε να τα ενσωματώσει στην εβδομαδιαία αναφορά του για την πρόοδο του έργου.



## 5. Ασφάλεια, Απόδοση και Βελτιώσεις

### 5.1 Ασφάλεια

Η αλματώδης ανάπτυξη του διαδικτύου και της σημασίας του έχει συνοδευτεί και από την αλματώδη αύξηση των επιθέσεων απέναντι σε διαδικτυακές εφαρμογές. Η ασφάλεια των διαδικτυακών εφαρμογών αποτελεί πλέον ένα πολύ σημαντικό και συνεχώς αναπτυσσόμενο πεδίο.

Το CERN, λόγω του μεγέθους, της σημασίας αλλά και του κύρους του, αποτελεί στόχο διαδικτυακών επιθέσεων. Το ζήτημα της ασφάλειας λαμβάνεται πάντα σοβαρά υπόψιν κατά την ανάπτυξη και συντήρηση εφαρμογών. Η ομάδα ασφαλείας του CERN πραγματοποιεί Penetration Testing στις διάφορες εφαρμογές του CERN, ενώ συχνά υπάρχουν παρουσιάσεις και συζητήσεις για θέματα ασφαλείας.

Στην εφαρμογή που αναπτύχθηκαν έχουν ληφθεί όλα τα μέτρα που απαιτούνται για την αποτροπή των συνηθισμένων τύπων επιθέσεων, ενώ και η αρχιτεκτονική της έχει λάβει υπόψιν ζητήματα ασφαλείας. Η ανάλυση των διαφόρων τύπων επιθέσεων και ευπαθειών είναι έξω από το σκοπό αυτής της εργασίας, ωστόσο θα αναφέρουμε ενδεικτικά μέτρα που έχουν παρθεί απέναντι στις πιο κοινές ευπάθειες :

- Η ταυτοποίηση του χρήστη (Authentication) γίνεται από την ήδη υπάρχουσα υπηρεσία του CERN Single Sign On. Όποτε ο χρήστης ζητάει κάποια αναφορά, ποτέ δεν προσδιορίζεται από το URI το ποιος χρήστης είναι, αλλά αυστηρά από το Session του.
- Κάθε είσοδος στοιχείων από το χρήστη στην εφαρμογή αντιμετωπίζεται ως ανασφαλής. Το Angular από μόνο φροντίζει ώστε ό,τι εμφανίζεται να γίνεται sanitised και escaped, αποφεύγοντας επιθέσεις τύπου Cross-Site Scripting (XSS).
- Ταυτόχρονα, στον εξυπηρετητή αντιμετωπίζουμε ολόκληρο το URI ως ανασφαλές, καθώς προέρχεται από τον χρήστη. Το query του χρήστη δεν αντιστοιχεί στους πραγματικούς πίνακες και στήλες της βάσης δεδομένων, αλλά «μεταφράζεται» με βάση λίστα αποδεκτών πινάκων και στηλών. Σε συνδυασμό με την αντιμετώπιση του ως ανασφαλές, αυτό προσφέρει προστασία απέναντι σε επιθέσεις SQL Injection.

- Το τελικό query δεν εκτελείται απευθείας στη βάση, αλλά μέσω κάποιας Stored Procedure, και τα δεδομένα μεταφέρονται στον εξυπηρετητή μέσω αυστηρά καθορισμένων Data Access Object. Αν οτιδήποτε δεν πηγαίνει σωστά στο query, αυτή η διαδικασία θα αποτύχει.
- Παρότι ο κίνδυνος από επιθέσεις τύπου Cross Site Request Forgery είναι περιορισμένος (καθώς αφορά μόνο τη δυνατότητα να αλλαχθούν οι αποθηκευμένες αναφορές ενός χρήστη), όλες οι ενέργειες της εφαρμογής που προκαλούν αλλαγές στη βάση προστατεύονται από CSRF tokens.
- Λόγω του τύπου της εφαρμογής, δεν υπάρχει κάποια πιθανότητα για επιθέσεις τύπου File Inclusion (Remote ή Local), ή για επιθέσεις τύπου Command Line Injection.
- Κατά τη δημιουργία αναφορών, ο χρήστης μπορεί να επιλέξει να μην εμφανίζονται τα ευαίσθητα έγγραφα, αν πρόκειται να μοιραστεί τα αποτελέσματα της αναφοράς με άτομα που δεν έχουν το ίδιο επίπεδο πρόσβασης.

Τέλος, επειδή προφανώς ποτέ κανείς δεν μπορεί να είναι σίγουρος για την 100% ασφάλεια μιας εφαρμογής, σημαντικό ρόλο παίζει η επίβλεψη και η καταγραφή, για να μπορεί να μελετηθεί, εντοπισθεί και διορθωθεί οποιοδήποτε κενό ασφαλείας. Για το λόγο αυτό όλα τα queries για reports καταγράφονται και κρατιούνται στα σχετικά logs (αρχεία καταγραφής) για μελλοντική ανάλυση αν χρειαστεί.

Ταυτόχρονα, τα αποτυχημένα queries κρατιούνται και σε ξεχωριστό αρχείο. Ως αποτυχημένα λογίζονται όλα τα queries που δεν μπόρεσαν να αντιστοιχηθούν σωστά στους πίνακες και τις στήλες της βάσης που έχουμε δηλώσει. Δεδομένου ότι κανονικά τα queries κατασκευάζονται από την ίδια την εφαρμογή, η ύπαρξη αποτυχημένων δείχνει πιθανότητα επιθέσεων στην εφαρμογή (ή γενικότερα προσπάθεια εύρεσης αδυναμιών), γι' αυτό και είναι σημαντικό να καταγράφονται και να παρακολουθούνται. Τέλος, υπάρχει ένα απλό script που παρακολουθεί το αρχείο καταγραφής των αποτυχημένων queries, και στέλνει άμεσα ειδοποίηση στους υπεύθυνους αν παρατηρηθεί ασυνήθιστα υψηλή δραστηριότητα.

## **5.2 Απόδοση – Εμπειρία χρήσης**

Οι σύγχρονες διαδικτυακές εφαρμογές είναι πολύ σημαντικό να προσφέρουν μια ομαλή και θετική εμπειρία χρήσης. Οι χρήστες πλέον έχουν αρκετά υψηλές απαιτήσεις για την ανταπόκριση και τη λειτουργία των εφαρμογών.

Για το λόγο αυτό στην εφαρμογή χρησιμοποιούνται παντού ασύγχρονα requests προς τον εξυπηρετητή. Έτσι, ο χρήστης μπορεί να συνεχίσει ότι κάνει όσο η εφαρμογή φορτώνει τα δεδομένα που χρειάζεται. Ταυτόχρονα, μόλις τα δεδομένα γίνουν διαθέσιμα, δεν απαιτείται ανανέωση ολόκληρης της σελίδας της εφαρμογής, αλλά μόνο του στοιχείου που χρησιμοποιεί αυτά τα δεδομένα (ή όποιας άλλης αλλαγής προκληθεί), κάνοντας τη διαδικασία πολύ πιο γρήγορη και ομαλή για τον χρήστη.

Επίσης, με τη χρήση γραφικών στοιχείων από μία αρκετά διαδεδομένη βιβλιοθήκη όπως η PrimeFaces, και ενεργοποιώντας τις αντίστοιχες δυνατότητες, προσφέρεται μια συνεπής και ομαλή εμπειρία χρήσης. Αυτό επιτυγχάνεται από τον συνδυασμό πολλών μικρών αλλά σημαντικών λεπτομερειών, στις οποίες οι χρήστες είναι συνηθισμένοι από άλλες εφαρμογές και γενικότερα επιζητούν από μία εφαρμογή. Τέτοιες είναι για παράδειγμα, η δυνατότητα ταξινόμησης ενός πίνακα με βάση μια στήλη επιλέγοντας τον τίτλο της στήλης, η δυνατότητα drag and drop ανάμεσα στις δύο λίστες επιλογής διαγραμμάτων, η δυνατότητα πολλαπλής ή μαζικής επιλογής στοιχείων με χρήση των κουμπιών control και shift αντίστοιχα, και ούτω καθεξής.

### **5.3 Σκέψεις για Μελλοντικές Προσθήκες**

Στο πλαίσιο αυτής της διπλωματικής εργασίας αναπτύχθηκε η πρώτη έκδοση της εφαρμογής αναφορών, με ορισμένα βασικά χαρακτηριστικά. Υπάρχουν ακόμα πολλά χαρακτηριστικά και βελτιώσεις που θα μπορούσαν να αναπτυχθούν, τα οποία θα πρέπει να αξιολογηθούν στο μέλλον, με βάση και τη γνώμη των χρηστών μετά τη χρήση της εφαρμογής για κάποιο διάστημα. Ενδεικτικά ορισμένα τέτοια χαρακτηριστικά μπορεί να είναι:

- Νέοι τύποι διαγραμμάτων, αναφορών ή στοιχείων διαθέσιμων σε μία αναφορά.
- Νέες παράμετροι κατά τη δημιουργία της αναφοράς, ή νέα ήδη φίλτρων
- Δυνατότητα δημιουργίας αναφορών για δεδομένα που υπάρχουν σε άλλα συστήματα, διασυνδεδεμένα με το EDMS.
- Δυνατότητα προσαρμογής της εμφάνισης των διαγραμμάτων από το χρήστη (αλλαγή μεγέθους, χρωμάτων και άλλα).
- Δυνατότητα καθορισμού από το χρήστη ενός κατώτατου ορίου πλήθους τιμών για την εμφάνιση σε ένα διάγραμμα. Για παράδειγμα, να μπορεί να ορίσει ο χρήστης ότι όλες οι επιλογές που αντιστοιχούν σε ποσοστό μικρότερο του 5% θα συγκεντρώνονται και θα εμφανίζονται ως «άλλο».
- Δυνατότητα εξαγωγής ολόκληρης της αναφοράς σαν pdf.
- Δυνατότητα αλληλεπίδρασης με τα διαγράμματα. Για παράδειγμα, όταν επιλέγεται μια στήλη σε ένα διάγραμμα, τα αποτελέσματα στον πίνακα δεδομένων που ακολουθεί να φιλτράρονται ώστε να εμφανίζονται μόνο αυτά που αντιστοιχούν σε αυτή τη στήλη.
- Προσθήκη σχετικών αναφορών σε σελίδες του EDMS που σχετίζονται. Για παράδειγμα, προσθήκη αναφοράς για ένα Context στη σελίδα διαχείρισης του Context.

Εν τέλει, υπάρχουν πολλές προσθήκες και βελτιώσεις που θα μπορούσε να σκεφτεί κανείς, που πιθανώς να βελτιώσουν την παρούσα εφαρμογή. Ωστόσο, με βάση και τις αρχές της ευέλικτης μεθοδολογίας, σε μια τέτοια εφαρμογή «η ανταπόκριση στην αλλαγή είναι πιο σημαντική από την τήρηση ενός προδιαγεγραμμένου σχεδίου».

Με την πρώτη έκδοση της εφαρμογής λοιπόν, και την ένταξη της στην δουλειά των 1000 καθημερινών χρηστών του EDMS, σίγουρα θα προέλθουν πολλές και σημαντικές προτάσεις βασισμένες σε πραγματικές περιπτώσεις χρήσης, που θα μπορούν να βελτιώσουν την παραγωγικότητα τους και να βελτιώσουν την εφαρμογή. Με τον ένα ή με τον άλλο τρόπο, ελπίζω ότι η εφαρμογή ήδη από την πρώτη της έκδοση θα συμβάλλει στη βελτιωμένη διασφάλιση της ποιότητας των δεδομένων και των διαδικασιών στο CERN.





## 6. Υλοποίηση

### 6.1 Front End

#### 6.1.1 Angular Models

Τα μοντέλα αντιπροσωπεύουν ένα αντικείμενο, και χρησιμοποιούνται ώστε να είναι πιο ευκολονόητη και συνοπτική η επεξεργασία και μεταφορά δεδομένων.

##### **DocumentModel.ts**

Μοντέλο που αναπαριστά ένα Document του EDMS. Έχει πεδία

```
cid: number;  
owner: string;  
creationDate: Date;  
updateDate: Date;  
id: string;  
version: string;  
title: string;  
description: string;  
keywords: string;  
context: string;  
type: string;  
releaseProcedure: string;  
status: string;  
equipmentCode: string;  
visibility: string;
```

## **ProjectModel.ts**

Μοντέλο που αναπαριστά ένα Project του EDMS. Έχει πεδία

```
cid: number;  
owner: string;  
creationDate: Date;  
updateDate: Date;  
id: string;  
title: string;  
description: string;  
keywords: string;  
context: string;  
equipmentCode: string;  
visibility: string;
```

## **ItemModel.ts**

Μοντέλο που αναπαριστά ένα Item του EDMS. Έχει πεδία:

```
cid: number;  
owner: string;  
creationDate: Date;  
updateDate: Date;  
id: string;  
version: string;  
title: string;  
description: string;  
keywords: string;  
context: string;  
equipmentCode: string;  
visibility: string;
```

### ContextModel.ts

Μοντέλο που αναπαριστά ένα Context του EDMS. Έχει πεδία:

```
id: string;  
name: string;  
description: string;
```

### EquipmentCodeModel.ts

Μοντέλο που αναπαριστά ένα EquipmentCode του EDMS. Έχει πεδία:

```
id: string;  
name: string;  
description: string;
```

### ReportModel.ts

Μοντέλο που αναπαριστά ένα report της εφαρμογής Έχει πεδία:

```
name: string;  
type: ReportType;  
priority: number;  
query: string;
```

Το name είναι το όνομα με το οποίο αποθηκεύει ο χρήστης την αναφορά, type ο τύπος της αναφοράς (ReportType είναι ένα ENUM με τους πιθανούς τύπους ενός report, δηλαδή document, project, item, context, equipmentCode). Priority είναι ένα νούμερο που χρησιμοποιείται για να μπορεί ο χρήστης να ταξινομήσει τις αναφορές (ποια καρτέλα εμφανίζεται πρώτη), και query είναι ένα string με όλες τις παραμέτρους της αναφοράς.

## 6.1.2 Angular Components

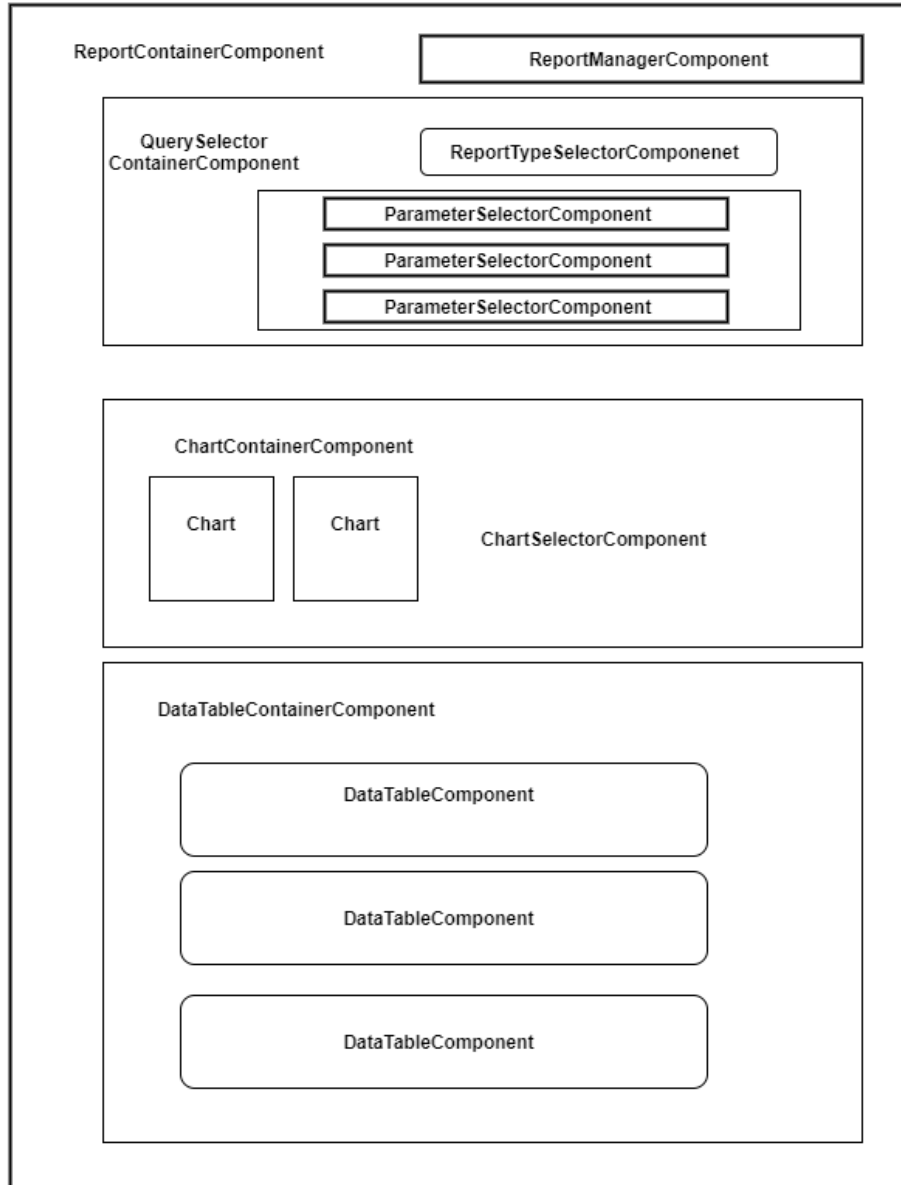
Τα Components αποτελούν τα βασικά συστατικά μιας Angular εφαρμογής. Κάθε Component ορίζει έναν selector, με το οποίο τα υπόλοιπα components μπορούν να το χρησιμοποιούν μέσα στο html template. Αν για παράδειγμα ορίσω ένα component με selector “my\_great\_component” , στην υπόλοιπη εφαρμογή μπορώ απλά να χρησιμοποιώ το html tag <my\_great\_component> για να το χρησιμοποιώ όπου και όποτε θέλω.

Κάθε component περιλαμβάνει μια κλάση typescript, με όσες ιδιότητες και μεθόδους χρειάζονται (εδώ ορίζονται και πιθανά input που χρειάζεται), ένα html template που δείχνει το περιεχόμενο του, και ένα ή περισσότερα CSS styles για τη μορφή του.

Στην επόμενη σελίδα ακολουθεί ένα απλό διάγραμμα που δείχνει πως συνδυάζονται τα components για να δημιουργηθεί η τελική εφαρμογή.

Μετά από αυτό υπάρχει μια παρουσίαση των βασικών components της εφαρμογής και της βασικής λειτουργίας κάθε component.

App Component



### **ChartSelectorComponent**

Πρόκειται για ένα component το οποίο εμφανίζει το κουμπί “Select Charts in Report”. Με το πάτημα του κουμπιού, εμφανίζει ένα overlay panel (primeng/OverlayPanelModule), στο οποίο υπάρχει ένα picklist widget (primeng/PickListModule). Αυτό εμφανίζει δυο λίστες, μία με τα διαθέσιμα διαγράμματα και μία με τα χρησιμοποιούμενα διαγράμματα, και επιτρέπει στο χρήστη να κάνει αλλαγές, να διαλέξει ποια θα χρησιμοποιούνται και ποια όχι. Υποστηρίζει drag and drop και multiple selection.

### **ChartContainerComponent**

Πρόκειται για το component υπεύθυνο για την εμφάνιση των διαγραμμάτων της αναφοράς. Περιλαμβάνει ένα chart (primeng/ChartModule) για κάθε επιλεγμένο διάγραμμα, καθώς και το ChartSelectorComponent για την επιλογή των διαγραμμάτων.

### **DataTableComponent**

Πρόκειται για component που εμφανίζει έναν πίνακα με δεδομένα (documents, projects, ή items). Υποστηρίζει φιλτράρισμα μιας στήλης ή όλων, ταξινόμηση με βάση κάποια στήλη, drop down menu με πολλαπλή επιλογή για την εμφάνιση ή απόκρυψη στηλών, καθώς και κουμπί για την εξαγωγή των δεδομένων σε CSV.

### **DataTableContainerComponent**

Πρόκειται για το component που εμφανίζει όλα τα DataTableComponent μιας αναφοράς (καθώς στην περίπτωση αναφοράς για ένα Context ή Equipment Code, θα υπάρχουν 3 DataTableComponent, ένα για documents, ένα για items και ένα για projects).

### **ReportTypeSelectorComponent**

Πρόκειται για το component, που εμφανίζει ένα radio button για την επιλογή του τύπου της αναφοράς, και πιθανές τιμές τις τιμές του ENUM DocumentType.

### **ParameterSelectorComponent**

Πρόκειται για το component που εμφανίζει μία γραμμή με βάση την οποία επιλέγεται μία παράμετρος για την αναφορά. Αποτελείται από 3 fields:

- attributeSelector: Drop down menu όπου επιλέγεται μία από τις παραμέτρους (πχ title, version, creation date).
- relationSelector: Drop down menu όπου επιλέγεται ο τελεστής που θέλουμε να χρησιμοποιηθεί (πχ StartsWith, Contain, Less Than, More Than).
- valueSelector, όπου επιλέγεται η τιμή. Το είδος του στοιχείου που εμφανίζεται εδώ (πχ dropdown menu, calendar κλπ) εξαρτάται από τον τύπο του attribute που επιλέχθηκε στο πρώτο πεδίο.

### **ParameterSelectorContainerComponent**

Έχει κουμπιά για προσθήκη ή αφαίρεση ParameterSelectorComponents, με σχέσης AND ή OR, ώστε να δημιουργείται το query της αναφοράς.

### **QuerySelectorContainerComponent**

Αποτελείται από το ReportTypeSelectorComponent και το ParameterSelectorContainerComponent, και συγκεντρώνει όλα όσα χρειάζονται για να επιλέξουμε τον τύπο και τις παραμέτρους της αναφοράς.

## **ReportManagerComponent**

Πρόκειται για το component που έχει τα κουμπιά για τη δημιουργία, αποθήκευση ή διαγραφή μίας αναφοράς.

## **ReportContainerComponent**

Αποτελείται από το ReportManagerComponent, και από ένα AccordionMenu (primeng/AccordionModule) με τρία tabs: Ένα για το QuerySelectorContainerComponent, ένα για το ChartContainerComponent και ένα για το DataTableContainerComponent. Το Accordion menu επιτρέπει στους χρήστες να αναπτύσσουν ή να συμπτύσσουν το τμήμα της αναφοράς που τους ενδιαφέρει (Παράμετροι, Διαγράμματα, Πίνακες Δεδομένων), για να μπορούν να εστιάζουν όπου χρειάζεται.

## **AppComponent**

Πρόκειται για το υψηλότερο Component της εφαρμογής. Αποτελείται από ένα TabView (primeng/TabViewModule), και κάθε tab είναι ένα ReportContainerComponent με μία αναφορά του χρήστη.



### 6.1.3 Angular Services

Τα services χρησιμοποιούνται για να φέρουμε τα δεδομένα που χρησιμοποιούν τα components με ασύγχρονο τρόπο. Στην εφαρμογή μας, κάθε service μπορεί να εκτελέσει ένα ή περισσότερα ασύγχρονα http request προς ένα URI του EDMS, και αφού λάβει τα δεδομένα σε μορφή JSON, τα μετασχηματίζει στον αντίστοιχο τύπο και τα δίνει στα components που τα ζήτησαν.

#### ContextService.ts

Πρόκειται για ένα Service που φέρνει όλα τα διαθέσιμα contexts από τον server. Χρησιμοποιείται για τα μενού στα οποία δείχνουμε τη λίστα από context.

Διαθέτει μία μέθοδο getAllContexts(), η οποία εκτελεί ένα Http Request στο URI `edms.cern.ch/reportsApp/contexts` .

Αφού πάρει τα δεδομένα σε μορφή JSON, δημιουργεί μια λίστα από αντικείμενα ContextModel και τα επιστρέφει (χρησιμοποιώντας Promise για την ασύγχρονη κλήση).

#### ReleaseProcedureService.ts

Πρόκειται για ένα Service που φέρνει όλα τα διαθέσιμα Release Procedures από τον server. Χρησιμοποιείται για τα μενού στα οποία δείχνουμε τη λίστα από release procedures.

Διαθέτει δύο μεθόδους:

- getAllReleaseProcedures(), η οποία εκτελεί ένα Http Request στο URI `edms.cern.ch/reportsApp/releaseProcedures` και διαβάζει όλες τις διαθέσιμες Release Procedures.

Αφού πάρει τα δεδομένα σε μορφή JSON, δημιουργεί μια λίστα από string και τα επιστρέφει (χρησιμοποιώντας Promise για την ασύγχρονη κλήση)

- getContextReleaseProcedures(contextName: string),  
η οποία εκτελεί ένα Http Request στο URI `edms.cern.ch/reportsApp/context/{contextId}/releaseProcedures` , και διαβάζει τις Release Procedures που είναι διαθέσιμες στο συγκεκριμένο context.

Αφού πάρει τα δεδομένα σε μορφή JSON, δημιουργεί μια λίστα από string και τα επιστρέφει (χρησιμοποιώντας Promise για την ασύγχρονη κλήση)

### **DocumentTypeService.ts**

Πρόκειται για ένα Service που φέρνει όλα τα διαθέσιμα Document Types από τον server. Χρησιμοποιείται για τα μενού στα οποία δείχνουμε τη λίστα από document types.

Διαθέτει δύο μεθόδους:

- `getAllDocumentTypes()`, η οποία εκτελεί ένα Http Request στο URI `edms.cern.ch/reportsApp/documentTypes` και διαβάζει όλες τις διαθέσιμες Release Procedures.

Αφού πάρει τα δεδομένα σε μορφή JSON, δημιουργεί μια λίστα από string και τα επιστρέφει (χρησιμοποιώντας Promise για την ασύγχρονη κλήση)

- `getContextDocumentTypes(contextName: string)`, η οποία εκτελεί ένα Http Request στο URI `edms.cern.ch/reportsApp/context/{contextId}/documentTypes` , και διαβάζει τα Document Types που είναι διαθέσιμες στο συγκεκριμένο context.

Αφού πάρει τα δεδομένα σε μορφή JSON, δημιουργεί μια λίστα από string και τα επιστρέφει (χρησιμοποιώντας Promise για την ασύγχρονη κλήση)

## EquipmentCodeService.ts

Πρόκειται για ένα Service που φέρνει όλα τα διαθέσιμα Equipment Codes από τον server. Χρησιμοποιείται για τα μενού στα οποία δείχνουμε τη λίστα από release procedures.

Διαθέτει δύο μεθόδους:

- `getAllEquipmentCodes()`, η οποία εκτελεί ένα Http Request στο URI `edms.cern.ch/reportsApp/equipmentCodes` και διαβάζει όλα τα διαθέσιμα Equipment Codes.

Αφού πάρει τα δεδομένα σε μορφή JSON, δημιουργεί μια λίστα από `EquipmentCodeModel` και τα επιστρέφει (χρησιμοποιώντας Promise για την ασύγχρονη κλήση)

- `getContextEquipmentCodes (contextName: string)`,

η οποία εκτελεί ένα Http Request στο URI `edms.cern.ch/reportsApp/context/{contextId}/equipmentCodes`, και διαβάζει τα Equipment Codes που είναι διαθέσιμα στο συγκεκριμένο context.

Αφού πάρει τα δεδομένα σε μορφή JSON, δημιουργεί μια λίστα από `EquipmentCodeModel` και τα επιστρέφει (χρησιμοποιώντας Promise για την ασύγχρονη κλήση)

## ReportDataService.ts

Πρόκειται για ένα Service που φέρνει τα δεδομένα μιας αναφοράς από τον server. Έχει πεδία:

```
report: ReportModel;  
reportDocuments: DocumentModel[];  
reportItems: ItemModel[];  
reportProjects: ProjectModel[];  
reportCharts: Chart[];
```

Στο πεδίο `report.reportQuery` αποθηκεύεται το query που θα σταλεί στον server για την αναφορά, και στα υπόλοιπα αποθηκεύονται τα δεδομένα που επιστρέφουν από τον server. Έχει τις μεθόδους:

- `generateReport()` που εκτελεί ένα `Http Request` στο `URI reportsApp/generateReport/{report.reportType}/{report.reportQuery}`

Αφού πάρει τα δεδομένα σε μορφή `JSON`, γίνεται μια βασική επεξεργασία ώστε να διαχωρισθούν και αποθηκεύονται στα αντίστοιχα πεδία.

- Μια σειρά από `getters` για τα υπόλοιπα πεδία, που επιστρέφουν δηλαδή τα `reportDocuments`, `reportItems`, `reportProjects`, `reportCharts`.

### **ReportManagementService.ts**

Πρόκειται για ένα `Service` που φέρνει τις αποθηκευμένες αναφορές του χρήστη, ή αποθηκεύει / τροποποιεί / διαγράφει αναφορές. Έχει το πεδίο

`reportList: ReportModel[];`

στο οποίο αποθηκεύονται όλες οι αναφορές του χρήστη, και έχει και τις εξής μεθόδους:

- `getReportList()` που φέρνει τη λίστα με τα `reports` του `user` από τον `server`. Εκτελεί `GET http request` στο `/reportsApp/reports`
- `saveReport(report: ReportModel)` που αποθηκεύει ένα νέο `report` στον `server` ή τροποποιεί ένα υπάρχον. Εκτελεί `PUT http request` στο `/reportsApp/report/{report.name}`
- `deleteReport(report: ReportModel)` που διαγράφει ένα `report`. Εκτελεί `DELETE http request` στο `/reportsApp/report/{report.name}`

## 6.2 Back End

### 6.2.1 Java Classes

#### **ReportServlet.java**

Πρόκειται για ένα servlet που ακούει στο api που εκθέτει η εφαρμογή. Έχει μεθόδους που αντιστοιχούν σε κάθε request που υποστηρίζεται, και καλεί τις αντίστοιχες μεθόδους. Πάντα στην αρχή κάθε request ελέγχει, επικυρώνει, κάνει escape και sanitise το input του request, και στη συνέχεια καλεί την μέθοδο που πρέπει, ανάλογα με το είδος του query, από τις παρακάτω κλάσεις:

#### **ReportGenerator.java**

Εμπεριέχει τις μεθόδους για τη δημιουργία ενός report. Συγκεκριμένα:

- Ελέγχει, επικυρώνει και κάνει escape και sanitise το report query.
- Κάνει parse το query, ανάλογα τον τύπο της εφαρμογής δημιουργεί το αντίστοιχο DAO (Data Access Object), και κάνει map τις διάφορες παραμέτρους του query στις παραμέτρους της Stored Procedure που θα κληθεί μέσω του DAO.
- Καλεί και εκτελείται η GET\_REPORT\_DATA StoredProcedure στη βάση δεδομένων.
- Διαβάζει τα αποτελέσματα από το DTO (Data Transfer Object) που επιστράφηκε, τα μετατρέπει σε μορφή JSON και απαντάει στο Request.

#### **ReportManager.java**

Είναι η κλάση που αναλαμβάνει όσα έχουν να κάνουν με την αποθήκευση των αναφορών του χρήστη. Συγκεκριμένα, διαθέτει μεθόδους, που καλώντας τις αντίστοιχες Stored Procedures της βάσης δεδομένων:

- Επιστρέφει τη λίστα με τις αναφορές του χρήστη
- Αποθηκεύει μια αναφορά (νέα ή τροποποίηση)
- Διαγράφει μια αναφορά

## **ReportParameterValuesHelper.java**

Είναι η κλάση που έχει μεθόδους για να καλέσει τις Stored Procedure και να επιστρέψει τις πιθανές τιμές των πεδίων των παραμέτρων της αναφοράς. Περιλαμβάνει μεθόδους :

- `getAllContexts();`
- `getAllReleaseProcedures();`
- `getAllEquipmentCodes();`
- `getAllDocumentTypes();`
- `getContextReleaseProcedures(String contextId);`
- `getContextQuipmentCode(String contextId)`
- `getContextDocumentTypes(String contextId)`

Κάθε μία από αυτές τις μεθόδους καλεί την αντίστοιχη Stored Procedure από τη βάση δεδομένων, και αφού λάβει τα δεδομένα, τα μετατρέπει σε JSON και τα επιστρέφει.

## 6.2.2 Database Stored Procedures

Στη βάση δεδομένων υπάρχουν αποθηκευμένες μία σειρά stored procedures, που είναι αυτές που εκτελούν όντως τα query στη βάση. Αντιστοιχούν ουσιαστικά στις μεθόδους της Java, και είναι οι εξής:

- STO\$GET\_ALL\_CONTEXTS();
- STO\$GET\_ALL\_REL\_PROCS();
- STO\$GET\_ALL\_DOC\_TYPES();
- STO\$GET\_ALL\_EQ\_CODES();
- STO\$GET\_CONTEXT\_REL\_PROCS();
- STO\$GET\_CONTEXT\_DOC\_TYPES ();
- STO\$GET\_CONTEXT\_EQ\_CODES ();
- STO\$GET\_DOC\_REPORT\_DATA();
- STO\$GET\_PROJ\_REPORT\_DATA();
- STO\$GET\_ITEM\_REPORT\_DATA();
- STO\$GET\_CONTEXT\_REPORT\_DATA();
- STO\$GET\_EQ\_CODE\_REPORT\_DATA();

Αυτές γενικά απλά εκτελούν τα query στη βάση, παίρνουν τα δεδομένα και τα επιστρέφουν στην Java Server.

Επιπλέον, αυτές που διαβάζουν documents, items και project από τη βάση, φιλτράρουν πρώτα τα αποτελέσματα με βάση τα δικαιώματα του χρήστη, ώστε να μην επιστρέψουν όσα έχουν να κάνουν με ευαίσθητα documents / items / projects στα οποία ο χρήστης δεν έχει πρόσβαση.





## 7. Βιβλιογραφία

### 7.1 Βιβλία

- Fain, Y. and Moiseev, A. (2016), **Angular 2 Development with TypeScript**, Manning Publications
- Sutherland, Jeff (2014), **Scrum The Art of Doing Twice The Work In Half the Time**, Crown Business
- Webber, J., Parastatidis, S, Robinson, I., (2010), **REST in Practice: Hypermedia and Systems Architecture** 1st Edition, O'Reilly

### 7.2 Επιστημονικές Δημοσιεύσεις

- Boyer, C. et al (2002), **The CERN EDMS – Engineering and Equipment Data Management System**, EPAC 2002
- Διακομανώλης, Β. (2016), **Ηλεκτρονική υποστήριξη εμπορικών διαδικασιών**, Πανεπιστήμιο Πατρών
- Wardzinska, A, et al (2015). **The Evolution of CERN EDMS**, CHEP 2015, Journal of Physics: Conf. Ser. 664 032032

### 7.3 Ιστοσελίδες

- **Angular** (2017) <http://www.angular.io> (Πρόσβαση στις 2 Σεπτεμβρίου 2017)
- **Oracle** (2017) <http://www.oracle.com/technetwork/database/features/plsql/index.html> (Πρόσβαση στις 21 Ιουνίου 2017)
- **OWASP** (2017) [https://www.owasp.org/index.php/Category:OWASP\\_Top\\_Ten\\_Project](https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project) (Πρόσβαση στις 19 Απριλίου 2017)
- **Stackoverflow Developers Survey Results 2017** (2017) <https://insights.stackoverflow.com/survey/2017#most-loved-dreaded-and-wanted> (Πρόσβαση στις 16 Αυγούστου 2017)
- **Stackoverflow Developers Survey Results 2016** (2016) <https://insights.stackoverflow.com/survey/2016#most-loved-dreaded-and-wanted> (Πρόσβαση στις 16 Αυγούστου 2017)
- **TypescriptLang** (2017) <https://www.typescriptlang.org/> (Πρόσβαση στις 18 Αυγούστου 2017)

### 7.4 Άλλα

- Convention for the Establishment of a European Organization for Nuclear Research (1953, όπως τροποποιήθηκε στις 17 Ιανουαρίου 1971)