ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
Π.Μ.Σ Πληροφοριακά Συστήματα & Υπηρεσίες
Ειδίκευση: Προηγμένα Πληροφοριακά Συστήματα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Διαλειτουργικότητα στις ηλεκτρονικές προμήθειες: Υλοποίηση επικυρωτή European Single Procurement Document (ESPD)

Όνομα Σιακανδάρης Γεώργιος Απόλλων

Επιβλέπουσα Καθηγήτρια:
Ανδριάνα Πρέντζα, Καθηγήτρια

ΠΕΙΡΑΙΑΣ

ΦΕΒΡΟΥΑΡΙΟΣ 2022
ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Διαλειτουργικότητα στις ηλεκτρονικές προμήθειες: Υλοποίηση επικυρωτή European Single Procurement Document (ESPD)

Σιακανδάρης Γεώργιος Απόλλων
Α.Μ.: ΜΕ1913

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία έχει ως αντικείμενο τη μελέτη της διαλειτουργικής υπηρεσίας European Single Procurement Document (ESPD) και την υλοποίηση της αντίστοιχης εφαρμογής ως προς το κομμάτι του ESPD validator. Πριν από την υπηρεσία του ESPD θα έπρεπε, οι εταιρείες να υποβάλουν διάφορα έγγραφα για να αποδείξουν ότι πληρούσαν τα κριτήρια αποκλεισμού και επιλογής μιας ανταγωνιστικής διαδικασίας, για παράδειγμα ότι είχαν πληρώσει φόρους και δεν είχαν καταδικαστεί για εγκληματική δραστηριότητα. Οι επιχειρήσεις και οι δημόσιοι φορείς είναι πλέον σε θέση να ανταποκριθούν σε αυτές τις υποχρεώσεις με ένα ενιαίο έγγραφο, το ESPD. Το Ευρωπαϊκό Έγγραφο Ενιαίας Σύμβασης (ESPD) της υπηρεσίας αυτής διευκολύνει τη συμμετοχή στις δημόσιες συμβάσεις. Το έγγραφο αυτό είναι ένα ενιαίο έντυπο δήλωσης που περιγράφει την καταλληλότητα, την οικονομική κατάσταση και τις ικανότητες μιας εταιρείας και χρησιμοποιείται ως προκαταρκτικό αποδεικτικό στοιχείο σε όλες τις διαδικασίες δημοσίων συμβάσεων. Για τη δημιουργία του εγγράφου υπάρχουν εφαρμογές που καθοδηγούν τους χρήστες σε κάθε κράτος-μέλος της Ευρωπαϊκής Ένωσης. Σε όλες τις εφαρμογές αυτές είναι απαραίτητο να υπάρχει ένας τρόπος ελέγχου ώστε να εξετάζεται η ορθότητα της υλοποίησής τους. Αυτός ο έλεγχος γίνεται μέσω ενός Validator, όπως και παρουσιάζεται στα επόμενα κεφάλαια.
Έχοντας ένα έγγραφο ESPD, μια εφαρμογή ESPD Validator καλείται να αναλύσει το έγγραφο και να το ελέγξει για τυχόν λάθη. Θα πρέπει να κάνει ελέγχους ως προς την ορθότητα των δεδομένων που θα εισάγονται και να εμφανίζει τα αντίστοιχα αποτελέσματα σε περίπτωση που υπάρχουν λάθη. Ο λόγος που χρειάζεται να γίνει αυτό είναι για τον έλεγχο της σωστής λειτουργίας των υπηρεσιών ESPD που αναπτύσσονται, δηλαδή αν έχουν εφαρμοστεί σωστά οι προδιαγραφές της κάθε έκδοσης.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Ηλεκτρονικές Προμήθειες

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:
Ευρωπαϊκό Έγγραφο Ενιαίας Σύμβασης, Vaadin, Spring, Validator

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω την επιβλέπουσα καθηγήτρια μου, κα. Ανδριάνα Πρέντζα, για την άμεση και έμπρακτη βοήθειά της καθώς και τους υποψήφιους Διδάκτορες Μαρία και Κώστα.
Θα ήθελα επίσης να ευχαριστήσω την οικογένειά μου για την ηθική και οικονομική υποστήριξη που μου προσέφεραν σε αυτό το ταξίδι.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΠΡΟΛΟΓΟΣ

Οι χώρες της Ευρωπαϊκής Ένωσης μπορούν να χρησιμοποιήσουν την ψηφιακή τεχνολογία και την επανάσταση των ψηφιακών δεδομένων για να βελτιώσουν την αποτελεσματικότητα και την παροχή των δημόσιων υπηρεσιών, καθώς και τη διαφάνεια και την εμπιστοσύνη των πολιτών. Πλέον οι περισσότερες υπηρεσίες που σε παλαιότερο χρόνο εκτελούνταν με φυσική παρουσία, όπως η κατάθεση εγγράφων στους δημόσιους φορείς πλέον γίνονται από απόσταση και με μεγαλύτερη ακρίβεια, χάρις στην τεχνολογική εξέλιξη των ημερών μας. Με τη χρήση των νέων εργαλείων κατάθεσης και υποβολής του ESPD που έχει υλοποιηθεί, υπάρχει μια τεράστια ευκαιρία για τη βελτίωση της παροχής υπηρεσιών και της συνολικής ικανότητας κάθε επιχείρησης να αναλαμβάνει έργα μεγαλύτερου εύρους, καταλήγοντας σε προσβάσιμες και με επίκεντρο τον πολίτη υπηρεσίες, ενώ ταυτόχρονα μειώνεται το κόστος συναλλαγών, αξιολόγησης και ο χρόνος ανάληψης και εκτέλεσης των υπηρεσιών αυτών. Το ESPD αποτελεί κομμάτι των ηλεκτρονικών δημόσιων προμηθειών και διευκολύνει τις επιχειρήσεις και τον δημόσιο τομέα στους διαγωνισμούς. Ο Validator με τη σειρά του σαν αναπόσπαστο κομμάτι της διεργασίας αυτής βοηθάει στο να ελεγχθούν τα συστήματα ESPD τόσο για λάθη όσο και για το αν είναι διαλειτουργικά βάσει των προδιαγραφών.

1. ΕΙΣΑΓΩΓΗ

1.1 Ορισμός του προβλήματος

Σε μια εποχή που τα μέσα κοινωνικής δικτύωσης είναι στη ζωή μας, όλες οι υπηρεσίες αναδιαμορφώνονται σε ηλεκτρονική μορφή. Το Ευρωπαϊκό έγγραφο ενιαίας σύμβασης έρχεται με τη σειρά του να ανοίξει τον δρόμο στον τομέα της ανάληψης έργων σε Ευρωπαϊκό επίπεδο με μια σωστή δόμηση και αποσαφήνιση των διαφορετικών δεδομένων για κάθε χώρα ξεχωριστά, ενοποιώντας τα σε μια λύση. Στην παρούσα διπλωματική εργασία σκοπός είναι να προσεγγισθεί μια σαφή υλοποίηση του ESPD validator καθώς και να παρουσιαστεί μια λεπτομερή ανάλυση της χρήσης του ESPD.

1.2 Δομή της μεταπτυχιακής διπλωματικής εργασίας

Η παρούσα διπλωματική εργασία διαρθρώνεται σε έξι κεφάλαια με σκοπό τη διευκόλυνση της σταδιακής μελέτης που απαιτεί το πολυδιάστατο θέμα του Ευρωπαϊκού εγγράφου ενιαίας σύμβασης και την παρακολούθηση της διπλωματικής.

- Στο πρώτο κεφάλαιο ορίζεται το πρόβλημα, η δομή και η συνεισφορά της διπλωματικής.
- Στο δεύτερο κεφάλαιο γίνεται βιβλιογραφική επισκόπηση, παρουσιάζεται η δομή του ESPD καθώς και τα κύρια συστατικά που χρησιμοποιεί όπως Schema,Model και Validator τα οποία αποτελούν και υλοποιούν το πλαίσιο προγραμματιστικά στο οποίο πρέπει να κινηθούν και να συμμορφωθούν οι φορείς και οι επιχειρήσεις που θέλουν να συμμετέχουν στους εκάστοτε διαγωνισμούς. Επίσης παρουσιάζονται εφαρμογές με παρόμοια υλοποίηση που προϋπήρχαν της παρούσης καθώς αναφέρονται και υπηρεσίες ελέγχου προδιαγραφών που πρέπει να πληρούνται ώστε το λογισμικό αυτό να είναι συμβατό με συγκεκριμένες προδιαγραφές που έχουν οριστεί.
- Στο τρίτο κεφάλαιο παρουσιάζεται ο σχεδιασμός του συστήματος με διαγραμματική ανάλυση της παρούσας διπλωματικής, όπως διάγραμμα περιπτώσεων χρήσης, διάγραμμα δραστηριοτήτων καθώς και μια ανάλυση της διαδικασίας αυτής του ESPD και των εμπλεκόμενων επιχειρηματικών διαδικασιών.
- Στο τέταρτο κεφάλαιο παρουσιάζεται η υλοποίηση του συστήματος, οι λόγοι επιλογής των τεχνολογιών που χρησιμοποιήθηκαν και το πώς συνεισφέρουν σε αυτή την διπλωματική.
- Στο πέμπτο κεφάλαιο γίνεται η παρουσίαση του συστήματος που υλοποιήθηκε, παρατηρείται ένα δείγμα του κώδικα για το UI με τις ανάλογες τεχνολογίες καθώς και παραδείγματα με αποτελέσματα που προέκυψαν από διαφορετικά σενάρια.
- Τέλος, το έκτο κεφάλαιο αποτελείται από τα συμπεράσματα που εξήχθησαν από την εργασία αυτή και γίνονται κάποιες προτάσεις για περαιτέρω βελτιώσεις σχετικά με το εξεταζόμενο θέμα.

1.3 Συνεισφορά της μεταπτυχιακής διπλωματικής εργασίας

Σκοπός την υλοποίησης αυτής είναι να μπορεί να γίνει ένας ουσιαστικός έλεγχος μεταξύ των εγγράφων αυτών του ESPD ως προς την ορθότητα της συμπλήρωσης και της σωστής χρήσης με αυτοματοποιημένους τρόπους στα πλαίσια της Ευρωπαϊκής ένωσης και στους κανόνες που τη διέπουν σηματοδοτώντας μια σωστή διαδικασία validation αυτή καθαυτή.

2. ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ

2.1 Ευρωπαϊκό Έγγραφο Ενιαίας Σύμβασης (ESPD)

Κάθε χρόνο, πολλές δημόσιες αρχές στην Ευρωπαϊκή Ένωση (ΕΕ) δαπανούν αρκετό ποσοστό του ΑΕΠ τους για την απόκτηση αγαθών και υπηρεσιών. Είναι αγοραστές σε ορισμένους σημαντικούς οικονομικούς τομείς, όπως η ενέργεια, οι μεταφορές και η υγειονομική περίθαλψη.
Ο ανοιχτός ανταγωνισμός είναι απαραίτητος για να διασφαλίσει ότι οι δημόσιες υπηρεσίες παρέχονται με λογικό κόστος. Ωστόσο, για πολλές μικρές και μεσαίες επιχειρήσεις που ενδιαφέρονται να συμμετάσχουν, η ποσότητα της γραφειοκρατίας που απαιτείται για την υποβολή προσφορών για συμβάσεις μπορεί να είναι τρομακτική. Η Ευρωπαϊκή Επιτροπή έχει δεσμευτεί να προσφέρει διαδικτυακές λύσεις για εύκολες, διαφανείς και δίκαιες διαδικασίες δημοσίων συμβάσεων που προάγουν τον ανταγωνισμό, την οικονομική ανάπτυξη και τη δημιουργία θέσεων εργασίας στην ενιαία αγορά της ΕΕ.
Η υπηρεσία ESPD [1] επιτρέπει στους αγοραστές να καθορίσουν κριτήρια αποκλεισμού και επιλογής, να προετοιμάσουν το ESPD για τους οικονομικούς φορείς στο διαδίκτυο και να το μοιραστούν με τους ενδιαφερόμενους προμηθευτές. Η υπηρεσία ESPD δημιουργεί έναν συνοπτικό πίνακα της συμμόρφωσης όλων των προσφερόντων με τα κριτήρια όταν ο αγοραστής λαμβάνει ολοκληρωμένα ηλεκτρονικά έγγραφα σύμβασης ESPD από τους προμηθευτές.
Οι προμηθευτές χρησιμοποιούν την υπηρεσία ESPD συμπληρώνοντας απλώς μια ηλεκτρονική φόρμα, χωρίς να απαιτείται επισύναψη δικαιολογητικών. Δηλώνουν ότι πληρούν τα κριτήρια και υποβάλλουν ηλεκτρονικά το ESPD με το υπόλοιπο της προσφοράς.
Η υπηρεσία ESPD είναι συνδεδεμένη με το e-Certis [2], το οποίο εμφανίζει ποιες πιστοποιήσεις πρέπει να υποβάλουν οι νικητές προμηθευτές ως απόδειξη επιλογής. Αυτό αποσαφηνίζει τη διαδικασία υποβολής προσφορών για διασυνοριακά έργα. Στη συνέχεια παρουσιάζονται το αίτημα ESPD (ESPD Request), η απάντηση ESPD (ESPD Response) και η αρχιτεκτονική του ESPD-VCD συστήματος όπου διακρίνονται τα δομικά του μέρη στην Εικόνα 1.

2.1.1 ESPD Request:

Το ESPD request είναι κλάση ανώτατου επιπέδου που περιέχει όλα τα απαραίτητα δεδομένα που θα τοποθετηθούν σε ένα ESPD Request έγγραφο όπως τα Στοιχεία αναθετουσών αρχών, τα Στοιχεία Προκήρυξης Συμβάσεων καθώς και κατάλογοι των απαιτήσεων που πρέπει να πληρούνται από τους οικονομικούς φορείς. Κάθε κριτήριο έχει μια αναφορά νομοθεσίας καθώς και μια συλλογή απαιτήσεων οργανωμένη σε ομάδες απαιτήσεων. Οι ομάδες απαιτήσεων είναι συλλογές απαιτήσεων που πρέπει να πληρούνται όλες για να ικανοποιηθεί το κριτήριο.

2.1.2 ESPD Response:

Το ESPD Response είναι μια υποκατηγορία του ESPD request που περιέχει όλα τα απαραίτητα δεδομένα που θα παρέχει ένας οικονομικός φορέας σε ένα έγγραφο ESPD Response όπως:

- Λεπτομέρειες Προμήθειας όπου περιέχουν τα Στοιχεία της αναθέτουσας αρχής και τα Στοιχεία Προκήρυξης Συμβάσεων.
- Τα Στοιχεία οικονομικών φορέων που δημιουργεί το ESPD response.
- Τον κατάλογο κριτηρίων που απαιτείται να έχει αποδειχθεί ορθό από τους οικονομικούς φορείς.

2.1.3 Ροή και λειτουργίες του ESPD:

1. Ο αγοραστής δημιουργεί/χρησιμοποιεί ένα πρότυπο espd, ορίζοντας τα κριτήρια εξαίρεσης και επιλογής.
2. Ο πλειοδότης δηλώνει την επιλεξιμότητά του και υποβάλλει το espd μαζί με την προσφορά.
3. Η υπηρεσία espd δημιουργεί αυτόματα μια επισκόπηση για όλες τις προσφορές.
4. Τα πρωτότυπα έγγραφα ζητούνται μόνο από τον νικητή.

2.1.4 Βασικά χαρακτηριστικά του ESPD είναι :

- Η Δήλωση ως προκαταρκτικό αποδεικτικό στοιχείο στις δημόσιες συμβάσεις.
- Η Απόδειξη επιλεξιμότητας που παρέχεται Διαδικτυακά.
- Το Πρότυπο που ετοιμάζεται από τον αγοραστή και συμπληρώνεται από τον πλειοδότη ή αυτόματα.
- Τα Αποδεικτικά Στοιχεία που παρέχονται μόνο από τους νικητές.
- Η Επαναχρησιμοποιήση σε μελλοντικούς διαγωνισμούς και προσφορές.
- Είναι έτοιμο για ενσωμάτωση στα Εθνικά Μητρώα.
  Παρέχει απαιτήσεις πιστοποίησης επαληθεύσιμες μέσω του νέου E-Certis.

2.2 ESPD υλοποίηση αναφοράς

Η υλοποίηση αναφοράς που έχει αναπτυχθεί από το Πανεπιστήμιο Πειραιώς [3] αποτελείται από δύο βασικά υποσυστήματα:

- Το ESPD/VCD framework που αποτελείται από τα ακόλουθα συστατικά ESPD / VCD Builder, Model, Validator, EDM και Codelists.
- Το ESPD/VCD System όπου αποτελείται από τη διαδικτυακή εφαρμογή ESPD/VCD Designer που και αυτή χρησιμοποιεί το ESPD/VCD Framework.

2.2.1 ESPD/VCD Schema

Το Schema δημιουργεί τις κλάσεις της Java [4] που αντιστοιχίζουν το επίσημο σχήμα ESPD/VCD του ESPD Request και ESPD Response, το οποίο βασίζεται στο Universal Business Language (UBL 2.1) και στο Core Criterion and Core Evidence Vocabulary (CCCEV). Το σχήμα ESPD/VCD είναι ο ακρογωνιαίος λίθος της διαλειτουργικότητας ESPD και όλες οι υλοποιήσεις συμβατές με ESPD/VCD πρέπει να έχουν υλοποιήσει το σχήμα αυτό.

2.2.2 ESPD/VCD Codelists

Το μοντέλο ανταλλαγής δεδομένων ESPD περιέχει ένα σύνολο από λίστες που χρησιμοποιούνται για την παροχή προκαθορισμένων στοιχείων σε διάφορα στοιχεία δεδομένων του σχήματος. Η βιβλιοθήκη παρέχει ένα απλό API για πρόσβαση σε όλες τις λίστες και τις αντιστοιχίσεις τους μεταξύ των τιμών της λίστας κωδικών και των πραγματικών τιμών που χρησιμοποιούνται στη φόρμα και στην εφαρμογή.

2.2.3 ESPD/VCD Model

Το μοντέλο είναι ένα απλοποιημένο domain, που βασίζεται στο CEN/BII Information Requirements Model (IRM) και στο ESPD Business Interoperability Specifiation (BIS) και VCD BIS. Μπορεί πολύ εύκολα να χρησιμοποιηθεί σε ένα UI. Η δομή του μοντέλου ακολουθεί κυρίως τη δομή CEN/BII που ορίζεται στα προφίλ ESPD και VCD. Αποτελείται από τρεις κύριες σύνθετες κλάσεις: Το ESPD Request, το ESPD Response που είναι υποκλάσεις του ESPD Request και το VCD.

2.2.4 ESPD/VCD Builder

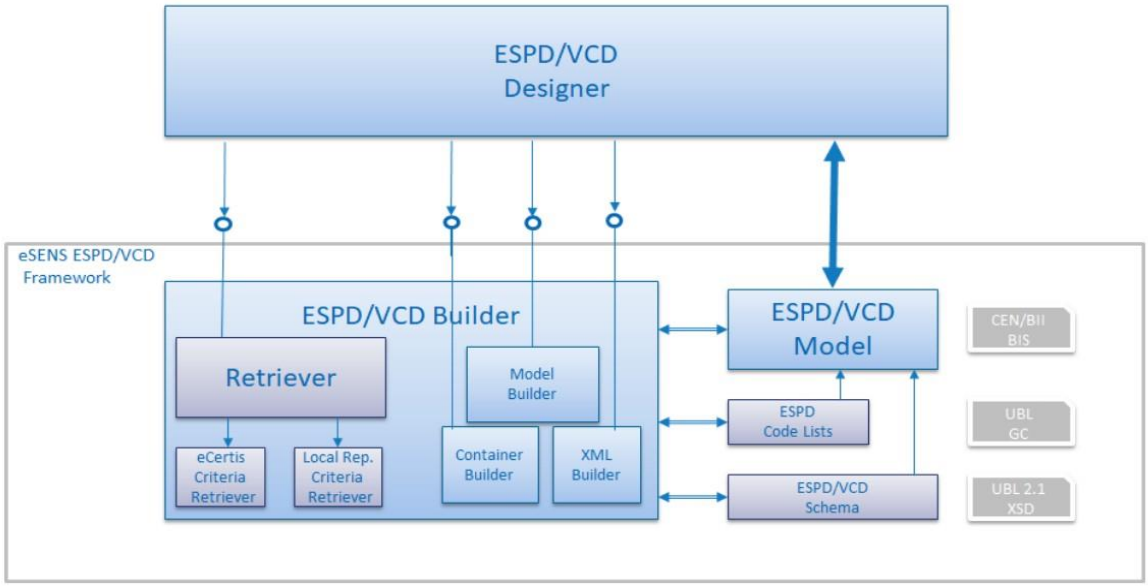
Το builder component είναι η βασική βιβλιοθήκη του ESPD/VCD Framework. Η βιβλιοθήκη περιέχει ένα σύνολο από builders που μετατρέπουν τα XML [5] ESPD/VCD Artefacts σε αντικείμενα έτοιμα για χρήση από τις εφαρμογές ενοποίησης. Οι builders αξιοποιούν επίσης το μοτίβο σχεδιασμού για τη παροχή ενός εύχρηστου API και για τη δημιουργία request και response ESPD τόσο για XML όσο και για model. Παρέχει τη λογική για τη σύνδεση της αναπαράστασης των δεδομένων του Schema component με την αναπαράσταση δεδομένων του Model component.

2.2.5 ESPD/VCD Validator

Το validator component [6] είναι μέρος του ESPD/VCD Builder και δεν εμφανίζεται ως ξεχωριστό στοιχείο. Σκοπός του είναι να ελέγχει την εγκυρότητα τόσο των εισαγόμενων όσο και των εξαγόμενων επιχειρηματικών εγγράφων XML. Η διαδικασία επικύρωσης περιλαμβάνει τις ακόλουθες πτυχές:

- Schema validation, δηλαδή ο έλεγχος εγκυρότητας έναντι των επίσημων XSD. Αυτά τα XSD πρέπει να είναι πανομοιότυπα με τα XSD που χρησιμοποιούνται για τη δημιουργία των κλάσεων Java του Schema component.
- Schematron validation, όπου εννοείται ο έλεγχος εγκυρότητας έναντι των επίσημων επιχειρηματικών κανόνων που καθορίζονται στα αντίστοιχα Schematron artefacts.

- Έλεγχος στους περαιτέρω ειδικούς κανόνες εφαρμογής.



Εικόνα 1:Μοντέλο ESPD [6]

2.3 Validation Engine και validators

Θα χρειαστεί να υλοποιηθεί μια λειτουργία ώστε να μεταφορτωθούν τα δεδομένα που δίνει το κάθε αρχείο xml που φορτώνεται στο σύστημα για να γίνουν οι αντίστοιχοι έλεγχοι ώστε να επικυρωθεί η ορθότητά τους. Για αυτό τον λόγο υλοποιείται ένας τέτοιος validator που θα καλύπτει αυτόν τον ρόλο. Σχετικά με το ESPD Validation, υπάρχουν 3 τύποι validation:

- *XML* (Schema & Schematron): γίνεται έλεγχος της δομής του XML εγγράφου, δηλαδή αν λείπει κάποιο tag ή αν υπάρχουν εσφαλμένα δεδομένα.
- *Data Acceptance*: Γίνεται έλεγχος αν έχουν περάσει σωστά κάποιες προσυμφωνημένες τιμές σε ESPD XMLs. Για παράδειγμα, όταν κάνει ένας ESPD system developer το Data Acceptance test βάζει στο όνομα του οικονομικού φορέα την τιμή "__Procurer". Στην συνέχεια, ο validator εξάγει την τιμή που έχει στο όνομα του οικονομικού φορέα το ESPD XML και ελέγχει αν είναι "__Procurer". Αυτό γίνεται για να ελεγχθεί πως ένα ESPD σύστημα έχει τη δυνατότητα να συμπληρώσει όλα τα πεδία σε ένα ESPD έγγραφο και να περάσει τις τιμές σωστά.
- *Data Persistence*: Γίνεται έλεγχος αν ένα ESPD Response XML περιέχει όλες τις πληροφορίες ενός ESPD Request XML. Γενικά το ESPD Response είναι ένα υπερσύνολο του ESPD Request και περιέχει το ίδιο μαζί με απαντήσεις σε κριτήρια. Στον έλεγχο αυτό εξετάζεται η δυνατότητα ενός ESPD συστήματος να κατασκευάσει ένα ESPD Response χωρίς να μεταβάλει το αρχικό ESPD Request.

2.4 ESPD ISA Validator

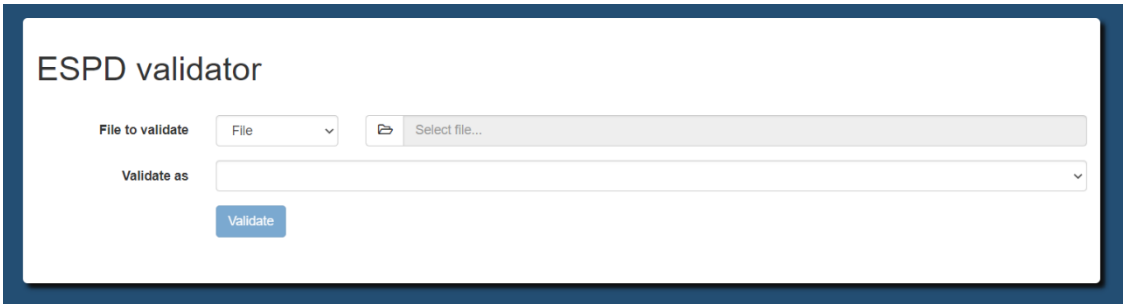
Ένα πρώτο παράδειγμα παρόμοιας λειτουργίας είναι ο ISA Validator που αναπτύχθηκε από την Ευρωπαϊκή Επιτροπή. Αυτή η υπηρεσία υλοποιείται από το Interoperability Test Bed, μια υπηρεσία που προσφέρεται από τη DG DIGIT της Ευρωπαϊκής Επιτροπής για έργα που εμπλέκονται στην παροχή διασυνοριακών δημόσιων υπηρεσιών.

2.4.1 Interoperability Test Bed

Το Interoperability Test Bed (ITB) είναι μια υπηρεσία που παρέχεται από την Ευρωπαϊκή Επιτροπή, και συγκεκριμένα από την Ομάδα Δοκιμών Διαλειτουργικότητας, για να βοηθήσει με τη διαλειτουργικότητα και τη συμμόρφωση σε δοκιμές πρωτοβουλιών που επιτρέπουν την ανάπτυξη διασυνοριακών δημόσιων υπηρεσιών. Στόχος του είναι να διευκολύνει τέτοια έργα διαλειτουργικότητας, παρέχοντας στους διευθυντές έργων τα εργαλεία που χρειάζονται για να καθιερώσουν την προσέγγιση δοκιμών που θα ακολουθήσουν οι συμμετέχοντες οργανισμοί τους. Οι υπηρεσίες δοκιμών που παρέχονται με αυτόν τον τρόπο μπορούν να παρουσιαστούν ως εργαλείο για να βοηθήσουν στην ανάπτυξη κατάλληλου λογισμικού ή ως απαιτούμενο βήμα αξιολόγησης προτού το λογισμικό μπορεί να θεωρηθεί συμβατό με μια συγκεκριμένη προδιαγραφή. Αξίζει να σημειωθεί ότι η κύρια εστίασή του είναι στα στοιχεία λογισμικού, και συγκεκριμένα:

- Η συμμόρφωση και η διαλειτουργικότητα διασφαλίζουν ότι το εν λόγω λογισμικό μπορεί να συμμετέχει σωστά και ουσιαστικά στις αναμενόμενες ανταλλαγές πληροφοριών με άλλα μέρη. Το test bed δεν προορίζεται να βοηθήσει στη δοκιμή παλινδρόμησης, απόδοσης ή διείσδυσης του λογισμικού.
- Το τεχνικό επίπεδο διασφαλίζει ότι το λογισμικό εφαρμόζεται σωστά, αλλά δεν εξετάζει τη διαλειτουργικότητα σε άλλα επίπεδα, όπως το υπάρχον νομικό πλαίσιο ή τις σχετικές οργανωτικές διαδικασίες που υποστηρίζει το λογισμικό.

Το λογισμικό του test bed GITB βρίσκεται στο κέντρο του test bed. Αυτό είναι το αποτέλεσμα του έργου GITB, το οποίο χρηματοδοτήθηκε από τη DG GROW της Ευρωπαϊκής Επιτροπής για τη δημιουργία των απαιτήσεων και του λογισμικού που απαιτούνται για την παροχή ενός test bed service διαλειτουργικότητας.



Εικόνα 2:Validator Ευρωπαϊκής Επιτροπής [7]

2.5 ESPD Validation Engine

Ένα επιπλέον παράδειγμα παρόμοιας υλοποίησης είναι αυτή που αναπτύχθηκε στα πλαίσια του έργου ESPDint από το Πανεπιστήμιο Πειραιά. Όσον αφορά το κομμάτι του user interface είναι υλοποιημένο σε JavaScript με το framework της Angular [8] και παρατηρείται πως υπάρχουν και οι τρεις τύποι validation XML Validation, Data Acceptance και Data Persistence όπως αυτοί παρουσιάστηκαν στην ενότητα 2.3 Validation Engine και validators.

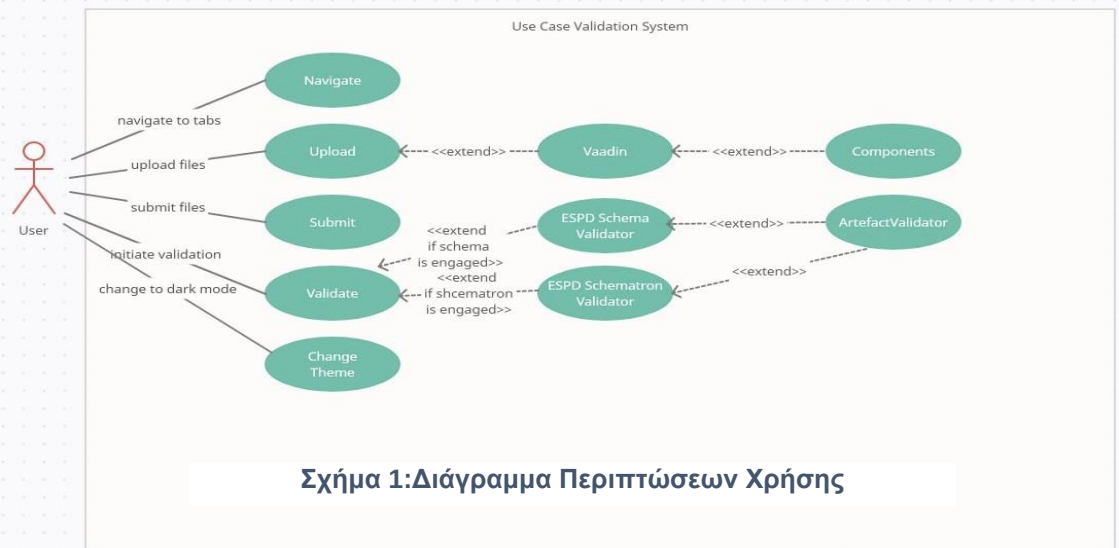
3. ΜΕΘΟΔΟΛΟΓΙΑ ΑΝΑΛΥΣΗΣ, ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΑΝΑΠΤΥΞΗΣ ΣΥΣΤΗΜΑΤΟΣ

Ο στόχος του Ευρωπαϊκού Ενιαίου Έγγραφου Προμηθειών είναι να καθιερώσει μια τυπική μορφή για την ανταλλαγή μηνυμάτων ESPD προκειμένου να καταστήσει το ESPD πιο αποτελεσματικό στην εφαρμογή και τη χρήση του. Το κομμάτι αυτό καλύπτει τις δημόσιες συμβάσεις, αν και μπορεί επίσης να χρησιμοποιηθεί σε σχέσεις επιχείρησης με επιχείρηση (B2B). Οι συναλλαγές που αναφέρονται προορίζονται να ανταλλάσσονται μεταξύ των συστημάτων υποβολής προσφορών των οικονομικών φορέων και των αναθετουσών φορέων. Αυτό σημαίνει ότι τα μέρη έχουν συνδέσει τα συστήματά τους με το Διαδίκτυο και διαθέτουν ενδιάμεσο λογισμικό που τους επιτρέπει να μεταδίδουν και να λαμβάνουν συναλλαγές με ασφαλή τρόπο και με συμφωνημένη σύνταξη. Το μοντέλο περιεχομένου συναλλαγών μπορεί να χρησιμοποιηθεί σε πλατφόρμες προμηθειών ή πύλες, διασφαλίζοντας ότι αυτές οι πλατφόρμες, καθώς και τα συστήματα προμηθειών των οικονομικών φορέων και των αναθετουσών αρχών, βασίζονται στα ίδια μοντέλα πληροφοριών και διαδικασιών, καθιστώντας τα πιο διαλειτουργικά. Στη συνέχεια παρουσιάζονται τμηματικά οι μεθοδολογίες της εργασίας αυτής καθώς και του γενικού περιεχομένου υλοποίησης με διαγραμματική αναπαράσταση.

3.1 Διάγραμμα Περιπτώσεων Χρήσης

Το Σχήμα 1 συγκεντρώνει τις ανάγκες του συστήματος απεικονίζοντας την εξωτερική όψη της υλοποίησης, αναγνωρίζει τους εσωτερικούς αλλά και τους εξωτερικούς παράγοντες που επηρεάζουν το σύστημα απεικονίζοντας τις λειτουργίες που μπορεί να χρησιμοποιήσει ο δράστης - χρήστης. Οτιδήποτε μέσα στο τετράγωνο στο Σχήμα 1 ανήκει στο πεδίο εφαρμογής του συστήματος σε αντίθεση με τον χρήστη (User) που είναι εκτός αυτού. Στις περιπτώσεις χρήσης ο χρήστης μπορεί να :

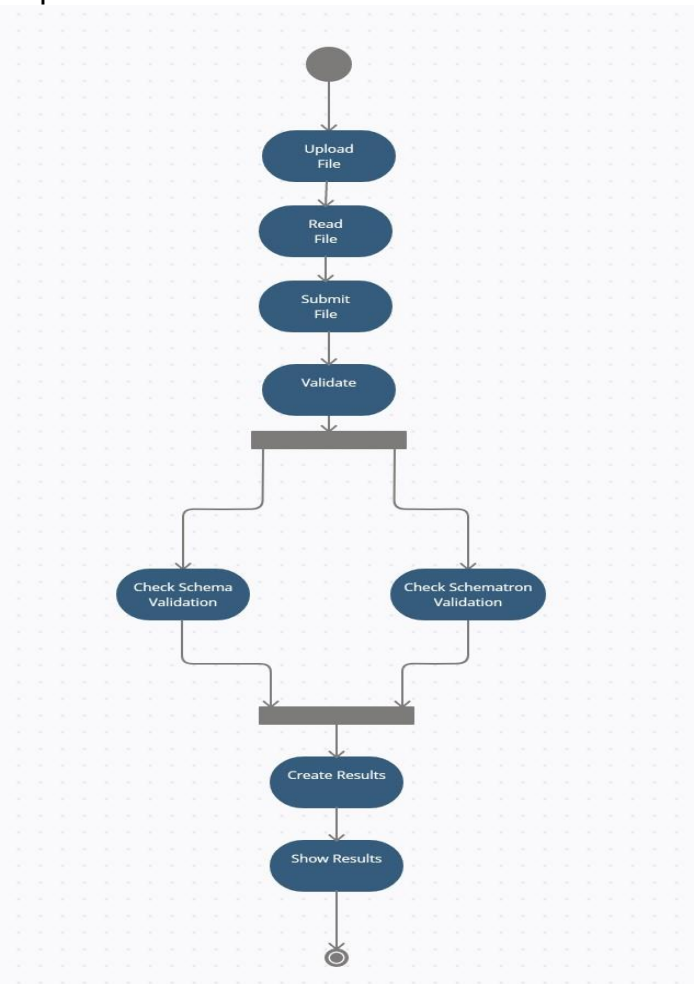
- Πλοηγηθεί σε υποσελίδες της εφαρμογής.
- Ανεβάσει αρχεία.
- Καταθέσει τα αρχεία.
- Αλλάξει θέμα χρωμάτων σε όλη την σελίδα.
- Κάνει έλεγχο λαθών στο έγγραφό του όπου θα καλείται είτε ο Schema Validator είτε ο Schematron Validator είτε και οι δύο.



Σχήμα 1:Διάγραμμα Περιπτώσεων Χρήσης

3.2 Διάγραμμα Δραστηριοτήτων

Το διάγραμμα δραστηριοτήτων (Σχήμα 2) δείχνει τη ροή του ελέγχου μέσα στο σύστημα. Μοντελοποιεί τις δραστηριότητες που πραγματοποιούνται είτε παράλληλα (Schema και Schematron Validator), είτε η μία μετά την άλλη (π.χ. Upload File και Read File). Σε περιπτώσεις μπορεί να είναι διαδοχική με την εκκίνηση της πρώτης δραστηριότητας ”Upload File” ο χρήστης μπορεί να ανεβάσει ένα αρχείο τύπου xml ,το σύστημα θα το διαβάσει ”Read File” και πατώντας το κουμπί “Submit File” θα γίνει ένας έλεγχος ”Validate” αν αυτό το αρχείο είναι τύπου xml ώστε να μπορέσει να αντλήσει τα δεδομένα που υπάρχουν σε αυτό. Σε περίπτωση που το αρχείο είναι διαφορετικού τύπου θα εμφανιστεί σαν αποτέλεσμα μόνο το όνομα του αρχείου που φορτώθηκε με μηδενικό αριθμό σφαλμάτων και ένα αντίστοιχο μήνυμα μη ορθής φόρτωσης ‘όπως ”Provided input is not an XML document”. Στη συνέχεια αν ο έλεγχος είναι ορθός θα σπάσει σε μικρότερους υπό ελέγχους για κάθε περίπτωση που χρειάζεται όπως “Check Schema Validation” και “Check Schematron Validation” όπου εκεί η κάθε περίπτωση εξετάζει τα δικά της δεδομένα βάση των περιορισμών που έχουν υλοποιηθεί από την Ευρωπαϊκή Ένωση για το πως πρέπει να είναι ένα σωστό αίτημα ως προς κατάθεση. Σε επόμενη ενέργεια τα αποτελέσματα αυτά έχουν δημιουργηθεί ”Create Results” και αποθηκευτεί και είναι έτοιμα προς αναπαράσταση “Show Results” όπου και λαμβάνει μέρος το UI του συστήματος για να τα εμφανίσει μέσω του Vaadin και των καρτελών που θα εμφανιστούν για κάθε ένα από αυτά.



Σχήμα 2: Διάγραμμα Δραστηριοτήτων

3.3 Ρόλοι

- ESPD requester: Οργανισμός που ζητά το έγγραφο ESPD ενός οικονομικού φορέα ή ενός παρόχου υπηρεσιών.
- ESPD provider: Οικονομικός φορέας ή πάροχος υπηρεσιών που παρέχει το έγγραφο ESPD.

Σχήμα 3:Διάγραμμα Περιπτώσεων Χρήσης

3.4 Εμπλεκόμενες Επιχειρηματικές Διαδικασίες

Η προμήθεια είναι ένα περίπλοκο θέμα με μια σειρά από κρίσιμες διαδικασίες, μερικές από τις οποίες απεικονίζονται στο παρακάτω Σχήμα 4.

Στο σχήμα διακρίνονται 3 βασικά στοιχεία της επιχειρησιακής λογικής αυτής :

- Business process Pre-award
- Business process Post-award
- Business process Procurement

Το ESPD υλοποιεί τη διαδικασία αυτή του σχήματος. Η διαδικασία ESPD αποτελεί μέρος της διαδικασίας πιστοποίησης στον τομέα της διαδικασίας Pre-award. Πιο αναλυτικά, όπως αναφέρθηκε και σε προηγούμενα κεφάλαια το Ευρωπαϊκό Ενιαίο Έγγραφο Προμηθειών ορίζει μια διαδικασία για την αίτηση και την υποβολή ESPD και VCD μέσω ηλεκτρονικών μηνυμάτων. Το ESPD είναι μια αυτοδήλωση του επίσημου καθεστώτος, της οικονομικής κατάστασης, των ικανοτήτων και της καταλληλότητας μιας εταιρείας. Η εφαρμογή τόσο του ESPD όσο και του VCD υποστηρίζεται κατά κύριο λόγο από τις διαδικασίες όπως διακρίνονται στο Σχήμα 4. Για συναλλαγές αιτήματος-Request και απόκρισης-Response, το ESPD και το VCD αναφέρονται στα ίδια στοιχεία και μοντέλα δεδομένων. Οι διαφορές μεταξύ του ESPD και του VCD εντοπίζονται στις διαδικασίες και όχι στις έννοιες. Το VCD είναι ένα μέσο παράδοσης πιστοποιητικών και αποδεικτικών εγγράφων που παρέχονται από δημόσιους φορείς ή τρίτους, ενώ το ESPD είναι μια αυτοδήλωση από οικονομικούς φορείς που προσφέρουν προκαταρκτικά στοιχεία. Το ESPD χρησιμοποιείται στη φάση προκαταρκτικής ανάθεσης(**Pre-awarding**) της διαδικασίας υποβολής προσφορών, ενώ το VCD χρησιμοποιείται κατά τη φάση ανάθεσης και σε διαδικασίες μετά την ανάθεση(**Post-award**), όπου η αναθέτουσα αρχή επιθυμεί από τον οικονομικό φορέα να ανανεώσει τα αποδεικτικά προεπιλογής. Οι ακόλουθες διαφορές μπορούν να συνοψιστούν ως εξής:

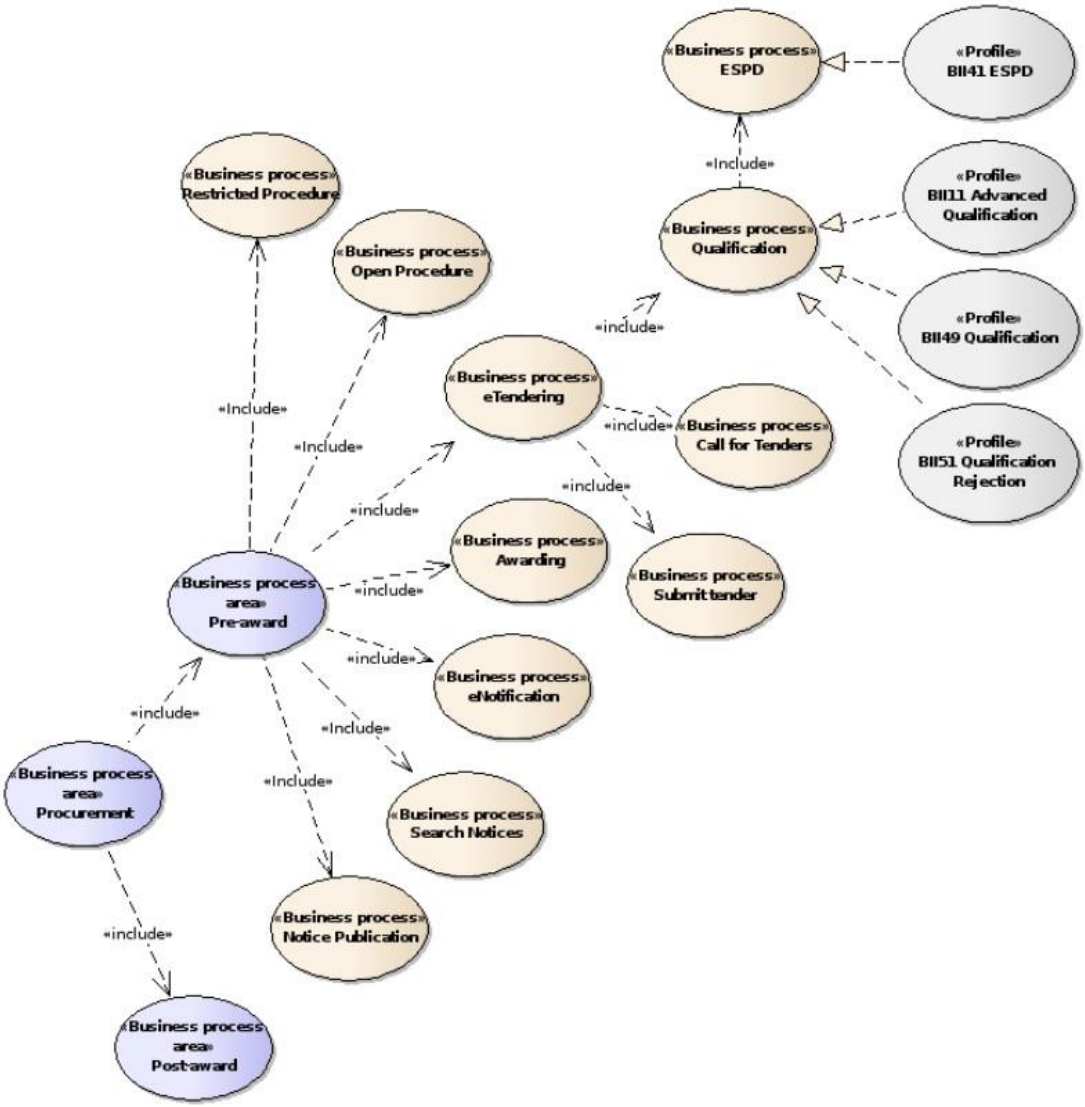
- Το ESPD χρησιμοποιείται στη φάση pre-awarding.
- Το VCD χρησιμοποιείται κατά τη φάση pre-awarding καθώς και στην post-award.
- Το ESPD χρησιμοποιεί αυτοδηλώσεις ή αναφορές σε έγγραφα σε απομακρυσμένη τοποθεσία.
- Το VCD χρησιμοποιεί αναφορές σε αντικείμενα δεδομένων που έχουν εκχωρηθεί φυσικά σε ένα φάκελο (container).

Αναθέτουσα αρχή:

- Ξεκινά να προετοιμάζει μια διαδικασία υποβολής προσφορών.
- Προετοιμάζει την προκήρυξη και την πρόσκληση υποβολής προσφορών.
- Καθορίζει τα κριτήρια προεπιλογής με τη μορφή δομημένου προτύπου ESPD (template),
- Προσθέτει τα τυποποιημένα κριτήρια προεπιλογής στην προκήρυξη και/ή στην πρόσκληση υποβολής προσφορών.

Οικονομικός φορέας:

- Συμμετέχει σε μία διαδικασία υποβολής προσφορών.
- Λαμβάνει τα έγγραφα του διαγωνισμού, συμπεριλαμβανομένων των κριτηρίων πρόκρισης.
    - Σχετικό έγγραφο που ελήφθη είναι η πρόσκληση υποβολής προσφορών, συμπεριλαμβανομένου του καταλόγου κριτηρίων προεπιλογής.
    - Ο κατάλογος των κριτηρίων προσόντων παραλαμβάνεται σε τυποποιημένη, δομημένη μορφή.
- Ανοίγει το υπόδειγμα ESPD που έλαβε.
- Δημιουργεί το τυποποιημένο έντυπο που επιβεβαιώνει ότι πληροί όλα τα κριτήρια προεπιλογής.
- Εισάγει το οριστικοποιημένο ESPD και το υποβάλλει μαζί με την προσφορά.
- Η αναθέτουσα αρχή λαμβάνει την προσφορά συμπεριλαμβανομένου του ESPD.



Σχήμα 4:Διάγραμμα Περιπτώσεων Χρήσης [9]

3.5 ESPD Διαδικασία

Το παρακάτω Σχήμα 5 δείχνει την ενδοεπικοινωνία της επιχειρηματικής διαδικασίας που υλοποιείται. Η επικοινωνία αυτή των επιχειρηματικών συνεργασιών καθορίζει τη σειρά των αλληλεπιδράσεων όταν το προφίλ εκτελείται εντός του πλαισίου του. Στο Σχήμα 5 παρατηρούνται οι κατηγορίες και οι καταστάσεις που έχουν διακριθεί:

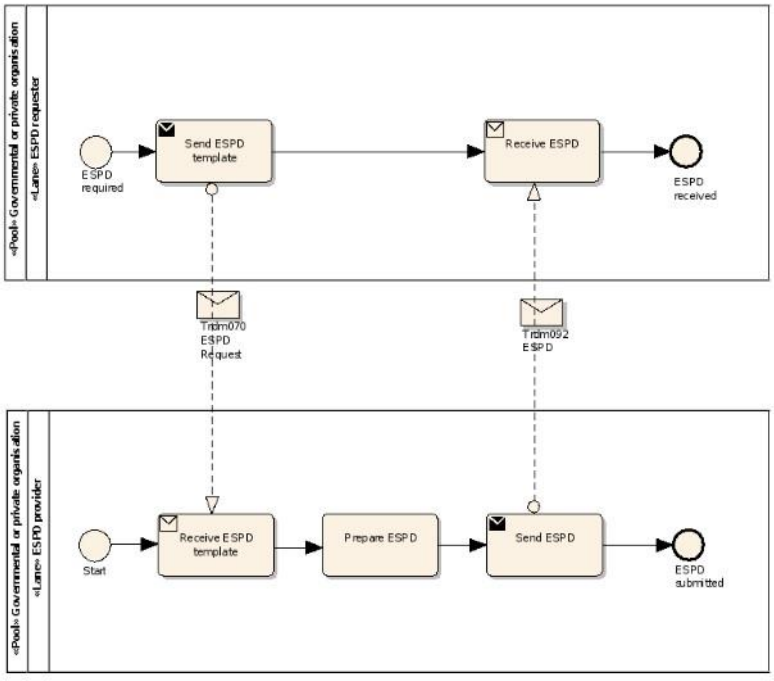
- Ο αιτών ESPD ζητά από τον πάροχο ένα ESPD. Ο πάροχος ESPD προετοιμάζει και αποστέλλει το ESPD στον αιτούντα.
- Ο αιτών χρειάζεται ESPD.
- Το ESPD έχει παρασχεθεί στον αιτούντα.
- Το Trdm092 βοηθάει στην παροχή πληροφοριών με απλοποιημένο τρόπο κατά την ανταπόκριση σε πρόσκληση υποβολής προσφορών.
- Το Trdm070 βοηθάει στη δόμηση πλαισίου ενός εγγράφου για την αναζήτηση πληροφοριών. Συγκεκριμένα, το αίτημα πρέπει να προσδιορίζεται και να έχει ημερομηνία, θα πρέπει να σημειωθεί η ημερομηνία έκδοσης, καθώς και αν το έγγραφο είναι αντίγραφο ή όχι.

Ενέργειες που γίνονται από τον αιτούντα (ESPD requester):

- Αποστολή ESPD προτύπου: Ο αιτών ζητά από τον πάροχο το έγγραφο. Το αίτημα περιλαμβάνει ένα πρότυπο που προσδιορίζει ποιες πληροφορίες πρέπει να περιλαμβάνονται στο ESPD.

Ενέργειες που γίνονται από τον πάροχο (ESPD Provider):

- Με βάση το πρότυπο, ο πάροχος δημιουργεί το έγγραφο.
- Ο πάροχος αποστέλλει το έγγραφο στον αιτούντα.



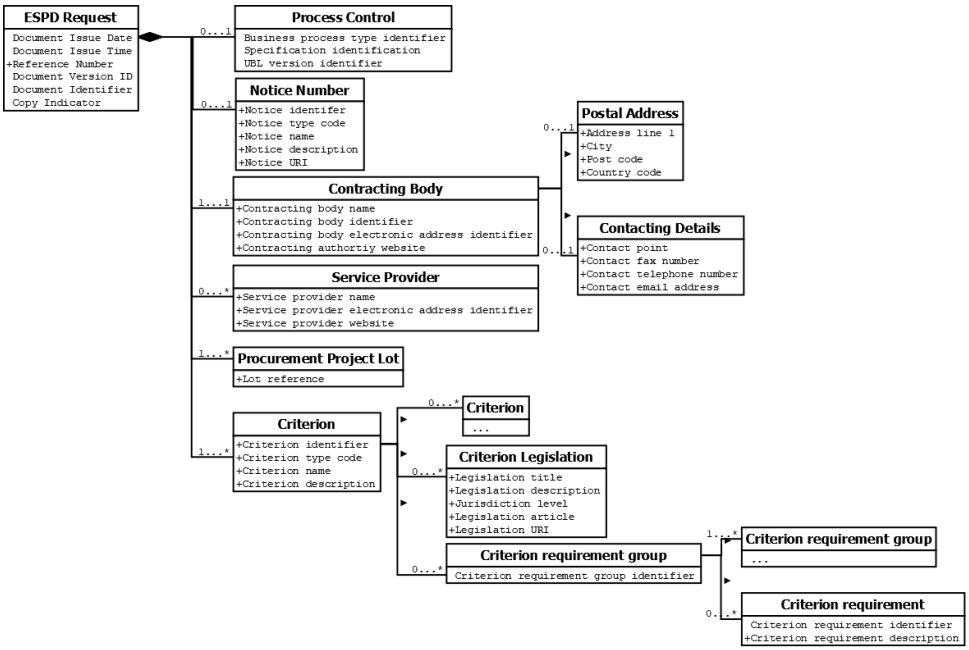
Σχήμα 5 :Ενδοεπικοινωνία Επιχειρηματικής Διαδικασίας [9]

3.6 Μοντέλο ESPD Request και ESPD Response

Οι επόμενες δύο αναπαραστάσεις Σχήμα 6 και Σχήμα 7, περιλαμβάνουν συγκεκριμένες λεπτομέρειες για κάθε κλάση και στοιχείο πληροφοριών σε ότι αφορά τις καταστάσεις του Request και του Response. Στο Σχήμα 6 παρατηρείται το μοντέλο δεδομένων για αυτόνομο ESPD: ESPD request transaction (Trdm070).

Για το ESPD Request:

- ESPD Request: Ένα δομημένο ηλεκτρονικό επιχειρηματικό έγγραφο για την αίτηση πληροφοριών προσόντων μέσω ESPD.
- Process control: Πληροφορίες σχετικά με την επιχειρηματική διαδικασία και τους κανόνες του εγγράφου.
- Notice Number: Για έργα προμηθειών που υπερβαίνουν το όριο, είναι υποχρεωτικό να προσδιορίζονται στοιχεία σχετικά με την προκήρυξη σύμβασης που δημοσιεύεται.
- Contracting body: Η αναθέτουσα αρχή ή ο αναθέτων φορέας που αγοράζει προμήθειες, υπηρεσίες ή δημόσια έργα χρησιμοποιώντας διαδικασίες διαγωνισμού.
- Postal Address: Πληροφορίες διεύθυνσης.
- Contacting details: Χρησιμοποιείται για την παροχή πληροφοριών επικοινωνίας για ένα σύνολο ή ένα άτομο.
- Service provider: Πάροχος.
- Procurement Project Lot: Εάν υπάρχει ενιαία παρτίδα έργου προμήθειας, το ESPD αναφέρεται σε έργο χωρίς παρτίδες.
- Criterion requirement: Ανάγκη εκπλήρωσης συγκεκριμένου κριτηρίου.

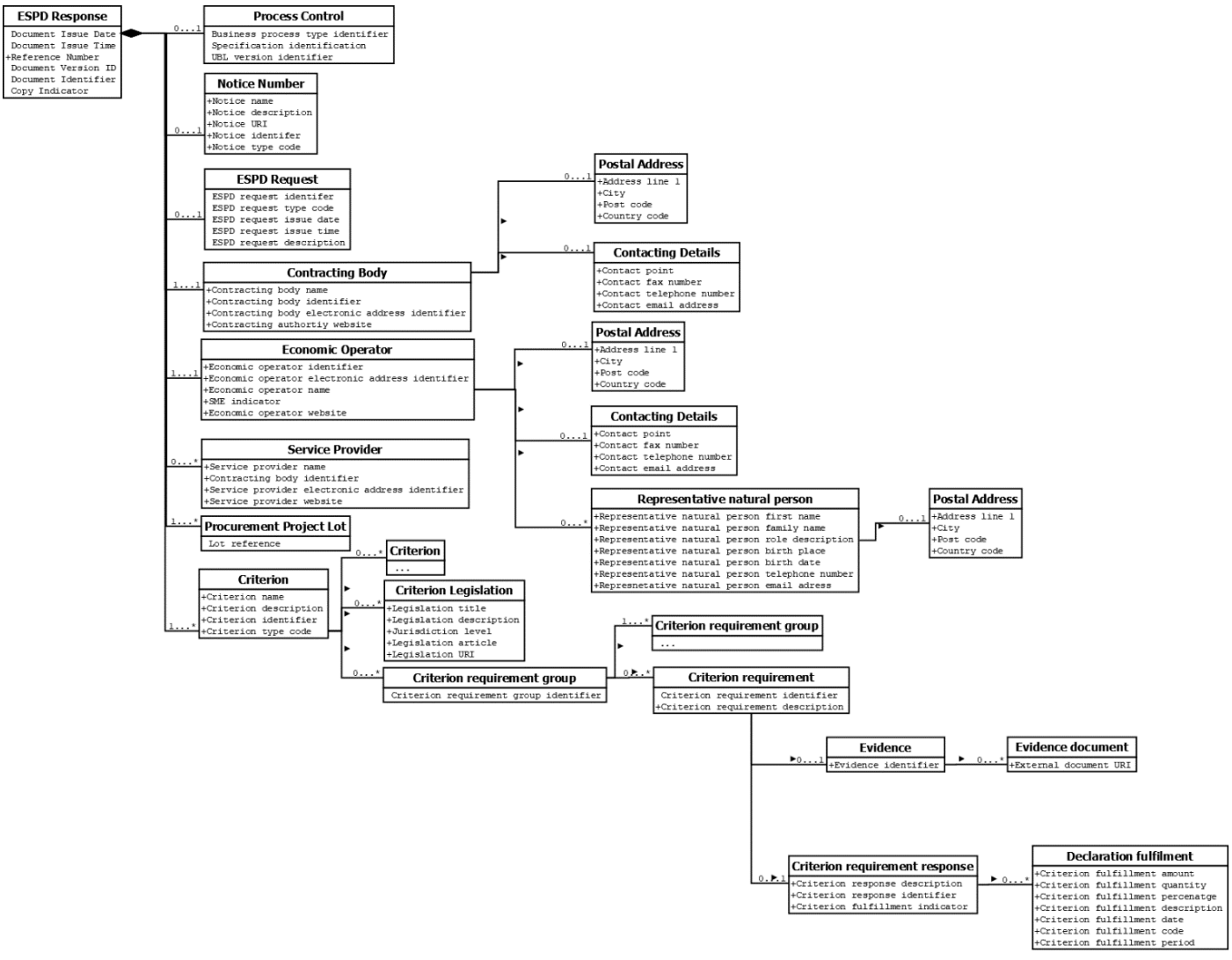


Σχήμα 6:ESPD Request [9]

Στο Σχήμα 7 παρατηρείται το μοντέλο δεδομένων: συναλλαγή απόκρισης ESPD (Trdm092)

Για το ESPD Response:

- ESPD Response: Ένα δομημένο ηλεκτρονικό επιχειρηματικό έγγραφο για την αίτηση πληροφοριών προσόντων μέσω ESPD.
- Process control: Πληροφορίες σχετικά με την επιχειρηματική διαδικασία και τους κανόνες που ισχύουν για το έγγραφο.
- Notice Number: Για έργα προμηθειών που υπερβαίνουν το όριο, είναι υποχρεωτικό να προσδιορίζονται τα στοιχεία σχετικά με την προκήρυξη σύμβασης.
- ESPD Request: Παραπομπή στο ESPD request.
- Contracting body: Η αναθέτουσα αρχή ή ο αναθέτων φορέας που αγοράζει προμήθειες, υπηρεσίες ή δημόσια έργα χρησιμοποιώντας διαδικασία.
- Postal Address: Πληροφορίες διεύθυνσης.
- Contacting details: Χρησιμοποιείται για την παροχή πληροφοριών επικοινωνίας για ένα σύνολο ή ένα άτομο.
- Service provider: Πάροχος.
- Economic operator: Κάθε φυσικό ή νομικό πρόσωπο ή δημόσια οντότητα ή ομάδα τέτοιων προσώπων ή/και οντοτήτων, συμπεριλαμβανομένης οποιασδήποτε προσωρινής ένωσης επιχειρήσεων, που προσφέρει την εκτέλεση εργασιών ή/και έργου, την προμήθεια προϊόντων ή την παροχή υπηρεσιών στην αγορά.
- Representative natural person: Πληροφορίες για άτομα που με τον ένα ή τον άλλο τρόπο εκπροσωπούν τον οικονομικό φορέα.
- Procurement Project Lot: Εάν υπάρχει ενιαία παρτίδα έργου προμήθειας, το ESPD αναφέρεται σε έργο χωρίς παρτίδες.
- Evidence: Απόδειξη.



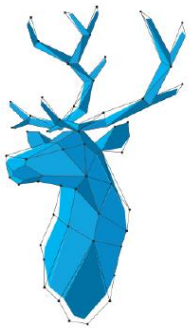
Σχήμα 7:ESPD Response [9]

4. ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ

4.1 Vaadin

Το Vaadin είναι ένα framework που βασίζεται σε Java και επιτρέπει στους προγραμματιστές να δημιουργούν user interfaces υψηλής ποιότητας. Έρχεται με μια σειρά έτοιμων προς χρήση components ενώ επιτρέπει να δημιουργηθούν νέα. Η έμφαση δίνεται στη χρηστικότητα, την επαναχρησιμοποίηση, την επέκταση και την ευελιξία που μπορεί να προσφέρει το framework ακόμα και για την αντιμετώπιση των αναγκών μεγάλων εταιρικών εφαρμογών με λειτουργίες όπως:

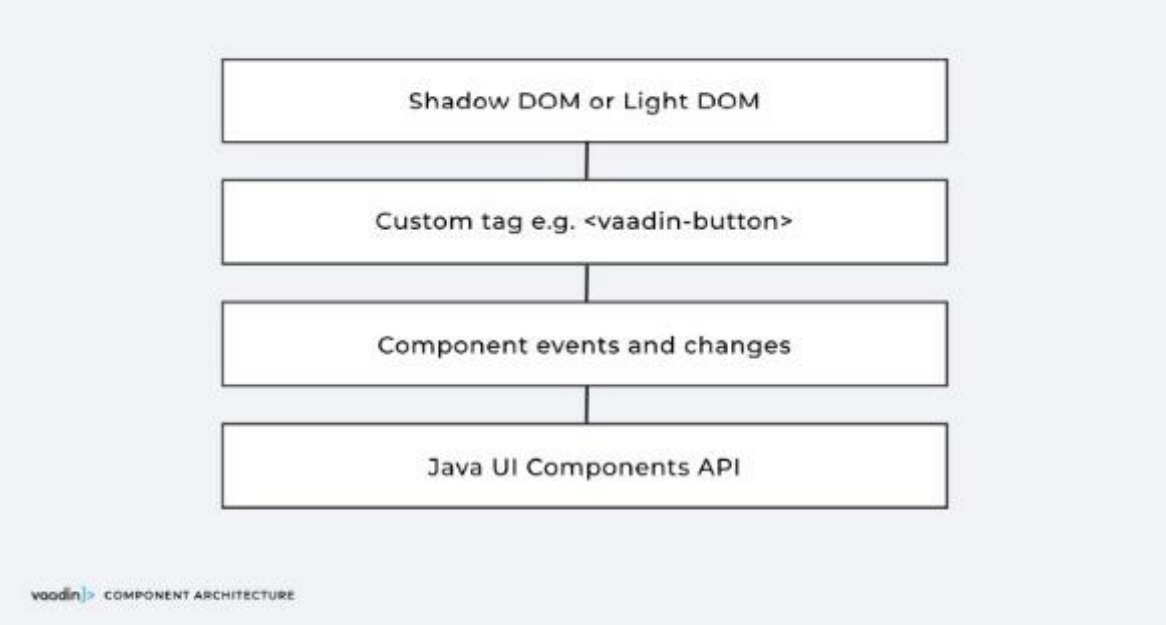
• Δημιουργία λειτουργικών μερών (components)
• Επεξεργασία DOM με το Element API
• Ενσωμάτωση web components
• Σύμπτυξη εφαρμογής για την παραγωγή
• OSGi υποστήριξη
• Λειτουργία σχεδιαστή του Vaadin
• Διαγράμματα
• Τεστ
• Ανανέωση παλιότερων εκδόσεων



Το Vaadin υποστηρίζει μια σειρά από διαφορετικά μοντέλα προγραμματισμού. Μπορεί να χρησιμοποιηθεί το Java API, τα HTML Web Components ή ένας συνδυασμός και των δύο. Μπορούν να προγραμματιστούν user interfaces κωδικοποιημένα πλήρως σε Java ή οποιαδήποτε άλλη γλώσσα συμβατή με JVM. Με τα έτοιμα components που προσφέρει το Vaadin επιτρέπεται να χρησιμοποιηθούν άλλα web framework που συμβαδίζουν με web components ή html document. Το Vaadin έχει ένα Java API από τη πλευρά του διακομιστή και ένα Java API από τη πλευρά του client για κάθε στοιχείο. Μπορούν επίσης να συνδυαστούν τα στοιχεία του client και του διακομιστή καθώς και μοντέλα από την πλευρά του client για την υλοποίηση ενός τμήματος ή του συνόλου του user interface καθώς το Vaadin είναι υπεύθυνο για την επικοινωνία.
Το Vaadin Designer είναι ένας “drag and drop” editor όπου χρησιμοποιείται για την υλοποίηση των user interfaces. Δίνει τη δυνατότητα οπτικοποίησης της εφαρμογής με έναν εύκολο τρόπο. Επιπλέον μπορεί να χρησιμοποιηθεί το πρόγραμμα επεξεργασίας για να σύρουμε στοιχεία από ένα σύνολο, όπου προσαρμόζει με ακρίβεια χαρακτηριστικά στοιχεία όπως η λεζάντα και το μέγεθος.
Το Vaadin TestBench είναι ένα πρόγραμμα που αυτοματοποιεί το testing. Πριν από κάθε παράδοση (deploy) σε περιβάλλοντα παραγωγής, γράφοντας τα τεστ σε Java, πραγματοποιούνται δοκιμές σε πραγματικό χρόνο χωρίς να χρειάζεται να ανοίξει κάποιο πρόγραμμα περιήγησης για να εκτελεστούν οι δοκιμές. Στο Vaadin τα πάντα είναι τύπου component. Για παράδειγμα αν πρέπει να υλοποιηθεί ένα κουμπί μπορεί να προγραμματιστεί σαν, new Button(). Όμοίως για text field, new TextField(). Επιπλέον μπορούν να δημιουργηθούν νέα components συνδυάζοντας components και layouts [10].

4.2 Λόγοι επιλογής

Υπάρχει ένα χαρακτηριστικό που κάνει το Vaadin μοναδικό – η υλοποίηση των Interfaces γράφοντας κώδικα μόνο σε Java καθώς δεν χρειάζεται καθόλου HTML και JavaScript. Σαν αποτέλεσμα όταν ολόκληρη η εφαρμογή είναι γραμμένη σε Java, ο κώδικας UI είναι αντικειμενοστρεφής, καθιστώντας εύκολη τη χρήση “design patterns” στον κώδικα αυτόν. Επιπλέον το μοντέλο προγραμματισμού του Vaadin αυτοματοποιεί την επικοινωνία μεταξύ του client και του διακομιστή. Φροντίζει τη διαχείριση του UI στο πρόγραμμα περιήγησης και οποιασδήποτε επικοινωνίας με τον διακομιστή. Δεδομένου ότι ο κώδικας του UI είναι σε Java, μπορεί να χρησιμοποιηθεί οποιαδήποτε από τις βιβλιοθήκες της Java. Το Vaadin περιλαμβάνει ένα σύνολο από components που λειτουργούν και φαίνονται ωραία μαζί σαν σύνολο. Όλα τα components σχεδιάστηκαν με γνώμονα τη προσβασιμότητα, είναι συμβατά με κινητές συσκευές και μπορούν να διαμορφωθούν με CSS. Το Vaadin προσφέρει επίσης τη δυνατότητα υλοποίησης του UI χρησιμοποιώντας HTML αν χρειαστεί, μαζί με αυτοματοποιημένη browser-server επικοινωνία που φροντίζει για το data binding.
Εν ολίγης το Vaadin υλοποιείται σε ένα πακέτο το οποίο περιέχει front end και back end και επιτρέπει την πιο εύκολη συντήρηση και παραμετροποίηση από προγραμματιστές που γνωρίζουν ως επί το πλείστον Java ενώ ταυτόχρονα υπάρχει μεγάλη υποστήριξη από την κοινότητα.



Εικόνα 3:Vaadin Component Architecture [11]

4.3 Αρχικοποίηση της εφαρμογής

Όταν δημιουργηθεί ένα νέο project παρατηρείται η MainLayout κλάση της Java. Η κλάση περιέχει κώδικα που υλοποιεί το annotation @Route και κάνει το view ορατό όταν πραγματοποιηθεί πλοήγηση στο http://localhost:8080, στην περίπτωση της εργασίας @Route("validator"). Το @PWA εμφανίζει μηνύματα, ενώ ενεργοποιεί αυτόματες λειτουργίες PWA, όπως μηνύματα όταν η εφαρμογή δεν είναι διαθέσιμη (εκτός σύνδεσης) ή εγκαθιστά την εφαρμογή στην αρχική οθόνη για να επιτρέπει στους χρήστες να έχουν πρόσβαση σε αυτή με ένα κλικ. Αυτός ο σχολιασμός είναι προαιρετικός. Η κλάση MainLayout επεκτείνει το AppLayout. Προκειμένου να προβληθεί το περιεχόμενο που είναι διαθέσιμο στον browser, πρέπει να γίνει επέκταση (extend) ενός υπάρχοντος UI component. Παρόμοια layout όπου μπορεί να γίνει η αρχή είναι και τα HorizontalLayout, FormLayout και SplitLayout. Όταν καλείται η εφαρμογή, το Vaadin δημιουργεί ένα νέο Instance του MainLayout το οποίο με τη σειρά του καλεί τον κατασκευαστή (constructor) της κλάσης. Στον κατασκευαστή, δημιουργείται ένα νέο κουμπί με έναν click listener που εμφανίζει μια ειδοποίηση όταν κάνουμε κλικ στο κουμπί, όπως βλέπουμε στην Εικόνα 4.

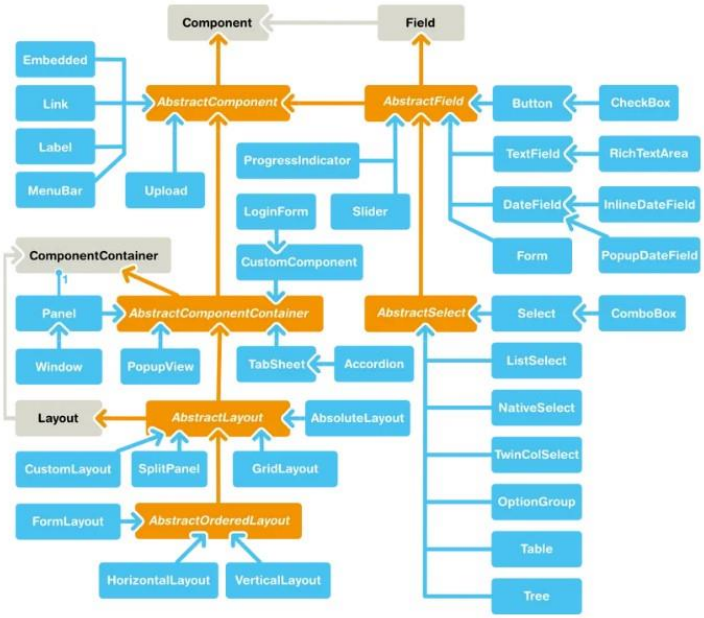
```
public class MainLayout extends AppLayout {

    public static class MenuItemInfo {

        private String text;
        private String iconClass;
        private Class<? extends Component> view;

        public MenuItemInfo(String text, String iconClass, Class<? extends Component> view) {
            this.text = text;
            this.iconClass = iconClass;
            this.view = view;
        }

        public String getText() { return text; }

        public String getIconClass() { return iconClass; }

        public Class<? extends Component> getView() { return view; }

    }

    private H1 viewTitle;

    public MainLayout() {
        setPrimarySection(Section.DRAWER);
        addToNavbar( touchOptimized: true, createHeaderContent());
        addToDrawer(createDrawerContent());

        Button toggleButton = new Button( text: "Toggle dark theme", click -> {
            ThemeList themeList = UI.getCurrent().getElement().getThemeList();

            if (themeList.contains(Lumo.DARK)) { // (2)
                themeList.remove(Lumo.DARK);
            } else {
                themeList.add(Lumo.DARK);
            }
        });
```

Εικόνα 4:Κλάση της εφαρμογής με όνομα MainLayout

4.4 Components

Το Vaadin προσφέρει μια πλούσια γκάμα έτοιμων UI components, εκτός από τα built-in, τα οποία μπορούν να χρησιμοποιηθούν για να δημιουργηθούν web applications με όλη τη λειτουργικότητα που χρειάζεται.

- Grids
- Layouts (form, split, App, Horizontal, Vertical)
- Form fields
- Upload
- Text Field
- Email Field
- Number Field
- Password Field
- Checkbox
- Radio Button Group
- List Box
- Combo Box
- Date Picker
- Time Picker
- Accordion
- Button
- Context Menu
- Dialog
- Notification
- Progress Bar
- Tabs
- Icons
- Tree Grid
- Login



Εικόνα 5:Vaadin Components [12]

4.5 Δέσμευση δεδομένων (Data Binding)

Σε πολλές εφαρμογές οι χρήστες υποβάλλουν δεδομένα συμπληρώνοντας πεδία σε φόρμες. Στις περισσότερες περιπτώσεις, αυτά τα δεδομένα αντιπροσωπεύονται σε κώδικα ως μια παρουσία ενός JavaBean. Στην περίπτωση της εργασίας που υλοποιήθηκε μπορεί να χρησιμοποιηθεί η κλάση Binder για να καθορισθεί ο τρόπος με τον οποίο οι τιμές (values) σε μια επιχειρηματική οντότητα (buisness object) συνδέονται με τα πεδία στο UI. Ο Binder διαβάζει τις τιμές στο business object και τις μετατρέπει στην αναμενόμενη μορφή του πεδίου και αντίστροφα. Μόνο τα στοιχεία που υλοποιούν το interface HasValue, όπως το TextField και το ComboBox, μπορούν να γίνουν bind από τον Binder.

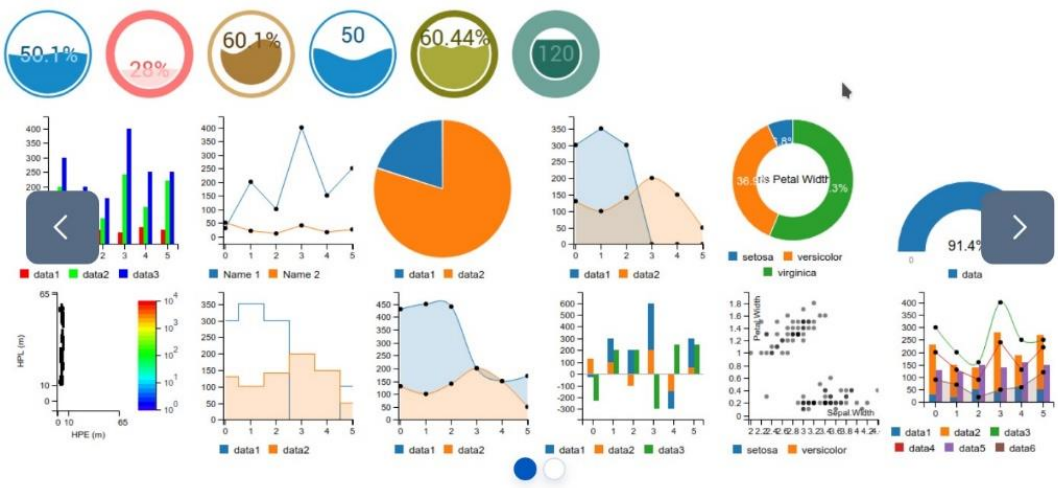
4.7 Vaadin Charts

Το Vaadin Charts είναι μια διαδραστική βιβλιοθήκη γραφημάτων με πολλές δυνατότητες. Μπορεί να απεικονίσει μονοδιάστατα ή δισδιάστατα δεδομένα σε πίνακα, καθώς και να διασκορπίσει δεδομένα με ελεύθερες τιμές X και Y, χρησιμοποιώντας μια ποικιλία μορφών γραφημάτων. Όλα τα στοιχεία του γραφήματος, καθώς και η οπτική σχεδίαση, μπορούν να προσαρμοστούν χρησιμοποιώντας ένα εξελιγμένο API καθώς και CSS. Οι ενσωματωμένες λειτουργίες επιτρέπουν στους χρήστες να αλληλοεπιδρούν με τα στοιχεία του γραφήματος με διάφορους τρόπους. Στην Εικόνα 6 διακρίνουμε τα είδη των γραφημάτων.

area	arearange	areaspline	areasplinerange
bar	boxplot	bubble	candlestick
column	columnrange	errorbar	flags
funnel	gauge	heatmap	line
ohlc	pie	polygon	pyramid
scatter	solidgauge	spline	treemap

Εικόνα 6:Vaadin Charts [13]

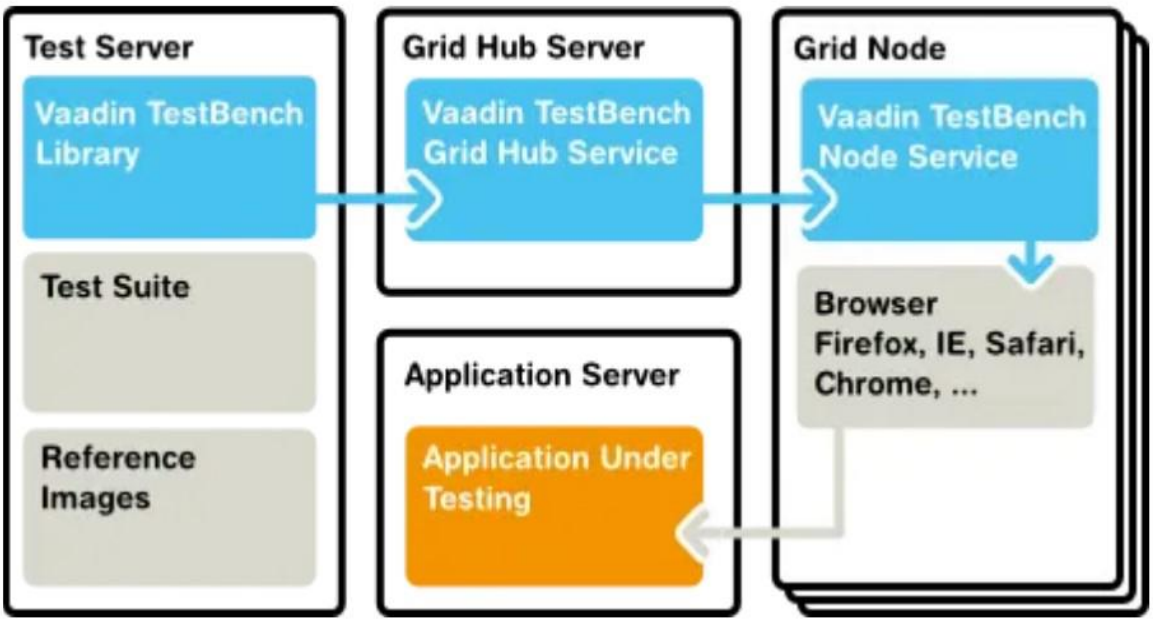
Τα γραφήματα έρχονται με πολλούς διαφορετικούς τύπους γραφημάτων όπως στην Εικόνα 7. Συνήθως καθορίζουμε τον τύπο του γραφήματος στον constructor του αντικειμένου της κλάσης της Java.



Εικόνα 7:Chart Overview [13]

4.8 Vaadin Testbench

Το Vaadin TestBench είναι ένα πρόγραμμα που επιτρέπει τη δημιουργία και την εκτέλεση τεστ στις εφαρμογές του Vaadin. Το TestBench λειτουργεί ως χρήστης της εφαρμογής και την εκτελεί, διασφαλίζοντας ότι οι ενέργειες που αναφέρονται στον κώδικα έχουν ολοκληρωθεί.
Μπορεί επίσης να επιθεωρήσει την εφαρμογή οπτικά, να κάνει μια ταυτοποίηση και επικύρωση ότι η εφαρμογή θα λειτουργεί σε όλα τα προγράμματα περιήγησης (browser). Το TestBench περιέχει υποστήριξη και σε άλλες εκδώσεις του Vaadin που κάνουν τη δοκιμή απλή και αξιόπιστη σε σύγκριση με άλλα τεστ γενικού περιεχομένου.



Εικόνα 8:Vaadin TestBench [14]

4.9 Spring

Η χρήση του Vaadin σε συνδυασμό με το Spring Boot [15] επιτρέπει να χρησιμοποιηθεί το Vaadin με τρόπο που επιταχύνει τη διαδικασία υλοποίησης και προσφέρει ένα γρήγορο και αποτελεσματικό περιβάλλον ανάπτυξης.
Το Vaadin Spring plugin επιτρέπει να χρησιμοποιηθεί το Vaadin σε συνδυασμό με το Spring Boot. Το Vaadin Start [16] είναι ο ευκολότερος τρόπος για να δημιουργήσουμε μια εφαρμογή με το Spring Boot και το Vaadin. Μπορούμε επίσης να διαμορφώσουμε και να κάνουμε λήψη ενός αρχικού project με σκοπό τη δημιουργία ενός νέου project. Σε οποιαδήποτε πλατφόρμα ανάπτυξης, το Spring Framework παρέχει μια ολοκληρωμένη αρχιτεκτονική προγραμματισμού και διαμόρφωσης για σύγχρονες επιχειρηματικές εφαρμογές που βασίζονται σε Java. Η υποστήριξη υποδομής του Spring σε επίπεδο εφαρμογών είναι ένα σημαντικό χαρακτηριστικό γιατί εστιάζει στην δομή των εφαρμογών έτσι ώστε οι χρήστες του να μπορούν να επικεντρωθούν στην επιχειρηματική λογική σε επίπεδο εφαρμογής χωρίς να χρειάζεται να ανησυχούν για συγκεκριμένες ρυθμίσεις ανάπτυξης και σύνδεσης.

Κάποια από τα χαρακτηριστικά του είναι :

- Υποστήριξη τεχνολογιών όπως: dependency injection, events, resources, i18n, validation, data binding, type conversion, SpEL, AOP.
- Για τεστ: mock objects, TestContext framework, Spring MVC Test, WebTestClient.
- Πρόσβαση δεδομένων: transactions, DAO support, JDBC, ORM, Marshalling XML.
- Spring MVC και Spring WebFlux web frameworks.
- Ενσωμάτωση: remoting, JMS, JCA, JMX, email, tasks, scheduling, cache.
- Γλώσσες προγραμματισμού: Kotlin, Groovy.

Είναι σημαντικό να κατανοήσουμε όχι μόνο τι επιτυγχάνει το framework, αλλά και ποιες αρχές τηρεί. Οι κατευθυντήριες αρχές του Spring Framework είναι οι εξής:

- Σε κάθε επίπεδο προσφέρει στους χρήστες επιλογές, επιτρέποντάς τους να αναβάλλουν τη λήψη αποφάσεων σχεδιασμού όσο το δυνατόν περισσότερο. Το ίδιο ισχύει και για πολλές άλλες ενοποιήσεις με third party API.
- Επιτρέπει μια ποικιλία απόψεων καθώς είναι ευέλικτο και δεν έχει σχεδιαστική γνώμη για το πώς πρέπει να γίνουν τα πράγματα. Καλύπτει ένα ευρύ φάσμα απαιτήσεων της εφαρμογής από διάφορες οπτικές γωνίες.
- Έχει υψηλό επίπεδο συμβατότητας (backward compatibility) γιατί η ανάπτυξη του spring έχει ενορχηστρωθεί προσεκτικά για να διασφαλιστεί ότι υπάρχουν ελάχιστες σημαντικές αλλαγές μεταξύ των εκδόσεων. Το Spring υποστηρίζει ένα προσεκτικά επιλεγμένο σύνολο εκδόσεων JDK και βιβλιοθηκών τρίτων για να διευκολύνει τη συντήρηση εφαρμογών και βιβλιοθηκών που εξαρτώνται από αυτό.

- Η ομάδα του Spring ξοδεύει πολύ χρόνο και προσπάθεια δημιουργώντας API που είναι εύκολα στη χρήση και τη συντήρηση σε πολλές εκδόσεις σε βάθος χρόνου.
- Θέτει υψηλές προσδοκίες για την ποιότητα του κώδικα, καθώς δίνει προτεραιότητα στο χρήσιμο, ενημερωμένο και ακριβές javadoc.

4.10 Maven

Το Apache Maven [17] είναι ένα εργαλείο διαχείρισης και κατανόησης που αφορά την δόμηση των projects λογισμικού. Το Maven μπορεί να διαχειριστεί την κατασκευή, την αναφορά και την τεκμηρίωση ενός έργου από μια κεντρική πληροφορία χάρη στην ιδέα του project object model (POM). Το Maven εμφανίστηκε ως μια προσπάθεια εξορθολογισμού των διαδικασιών κατασκευής. Πριν από αυτό σε πολλά έργα όπου το καθένα έχει τα δικά του αρχεία κατασκευής, ελαφρώς διαφορετικά μεταξύ τους, θα χρειαζόμασταν έναν κοινό τρόπο κατασκευής. Το Maven ήρθε να καλύψει αυτό το κενό ώστε να έχουμε έναν σαφή ορισμό του τι είναι ένα έργο, να προσφέρει έναν απλό τρόπο δημοσίευσης πληροφοριών και ένα μέσο για την ανταλλαγή JARs. Ως αποτέλεσμα υπάρχει ένα εργαλείο που μπορεί να χρησιμοποιηθεί για τη δημιουργία και τη διαχείριση οποιουδήποτε έργου που βασίζεται σε Java.
Ο κύριος σκοπός του Maven είναι να βοηθήσει τους προγραμματιστές να κατανοήσουν ολόκληρη την κατάσταση ενός έργου στο συντομότερο δυνατό χρονικό διάστημα. Το Maven αντιμετωπίζει μια σειρά ζητημάτων για να επιτύχει αυτόν τον στόχο:

- Κάνοντας τη διαδικασία κατασκευής απλή.
- Παρέχει ομαδοποιημένο κατασκευαστικό σύστημα.
- Προσφέρει βελτιωμένες πρακτικές ανάπτυξης.
- Παρέχει υψηλής ποιότητας πληροφορίες για τα έργα.

Ενώ το Maven δεν αφαιρεί την ανάγκη κατανόησης των υποκείμενων μηχανισμών, προστατεύει τους προγραμματιστές από πολυπλοκότητες.
Όπως προαναφέραμε το Maven παρέχει χρήσιμες πληροφορίες για το έργο, που προέρχονται εν μέρει από το POM και εν μέρει από τις πηγές του έργου. Μπορεί να παρέχει για παράδειγμα:

- Αλλαγή Log κατευθείαν από το source control .
- Cross reference πηγές.
- Λίστες αλληλογραφίας όπου διαχειρίζονται από το έργο.
- Dependencies, που χρησιμοποιούνται από το έργο.
- Unit test αναφορές.

4.11 Gradle

Το Gradle [18] είναι ένα δωρεάν, ανοιχτού κώδικα, εργαλείο αυτοματισμού κατασκευής εφαρμογών που μπορεί να χρησιμοποιηθεί για τη δημιουργία σχεδόν οποιασδήποτε μορφής λογισμικού. Ορισμένες από τις πιο βασικές πτυχές του είναι:

- Υψηλή απόδοση: Το Gradle εξοικονομεί χρόνο εκτελώντας μόνο εργασίες που απαιτούνται επειδή έχουν αλλάξει οι είσοδοι ή οι έξοδοί τους. Μια κρυφή μνήμη μπορεί επίσης να χρησιμοποιηθεί για την επαναχρησιμοποίηση εργασιών από προηγούμενες εκτελέσεις ή ακόμα και από ξεχωριστό υπολογιστή.
- Βάση με JVM: Το Gradle λειτουργεί στην Java Virtual Machine (JVM) και θα χρειαστεί το Java Development Kit (JDK) για να υλοποιηθεί. Επίσης δεν είναι μόνο για την κατασκευή έργων JVM αλλά υποστηρίζει και native projects.
- Συμβάσεις: Το Gradle παίρνει μια σελίδα από το βιβλίο του Maven και χρησιμοποιεί συμβάσεις για να κάνει κοινά είδη έργων απλά, όπως Java projects. Με τα σωστά plugins, μπορούν να δημιουργηθούν ελαφριά scripts για μια ποικιλία έργων. Επιπλέον το Gradle επιτρέπει να γίνει μια παράκαμψη των παραπάνω, να προστεθούν νέες εργασίες και να προσαρμοστούν στις εκδόσεις που βασίζονται σε νέες συμβάσεις με διάφορους τρόπους.
- Επεκτασιμότητα: Το Gradle μπορεί εύκολα να επεκταθεί για να συμπεριλάβει προσαρμοσμένους τύπους εργασιών ή ακόμα και ένα μοντέλο κατασκευής. Για παράδειγμα, η υποστήριξη έκδοσης Android περιλαμβάνει πολλά νέα build concepts.
- Υποστήριξη IDE: Πολλά σημαντικά IDE επιτρέπουν να εισαχθεί Gradle builds και να γίνει αλληλεπίδραση μαζί τους όπως: Android Studio, IntelliJ IDEA, Eclipse και NetBeans. Το Gradle έχει επίσης υποστήριξη για τη δημιουργία των αρχείων που απαιτούνται για τη φόρτωση ενός project στο Visual Studio.
- Διορατικότητα: Οι σαρώσεις (scans) της λειτουργίας build παρέχουν πολλές πληροφορίες σχετικά με την εφαρμογή που μπορούν να χρησιμοποιηθούν για να βρεθούν προβλήματα σε αυτήν, ενώ επιπλέον βελτιώνει την απόδοσή της. Μπορούμε επίσης να μοιραστούμε εκδόσεις με άλλους προγραμματιστές, κάτι που είναι πολύ χρήσιμο εάν χρειαζόμαστε βοήθεια για την επίλυση ενός προβλήματος στη συγκεκριμένη έκδοση.

4.12 JUnit

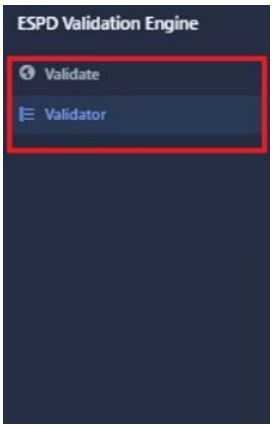
Το Junit [19] είναι ένα framework όπου βοηθά να γίνουν τεστ και διορθώσεις στο κάθε έργο. Ένα τεστ είναι ένα κομμάτι λογισμικού που εκτελείται και βεβαιώνει ότι ένα άλλο κομμάτι λογισμικού συμπεριφέρεται με έναν δεδομένο τρόπο που θέλει ο προγραμματιστής.
Αυτά διασφαλίζουν ότι ορισμένες πτυχές του λογισμικού λειτουργούν όπως προβλέπεται. Αυτές οι δοκιμές εκτελούνται συχνά αυτόματα από το σύστημα, επιτρέποντας στον προγραμματιστή να αποφύγει την παραβίαση του τρέχοντος κώδικα ενώ εργάζεται σε νέο κώδικα.
Τα αυτοματοποιημένα tests βοηθούν στον εντοπισμό προβλημάτων λογισμικού που προκαλούνται από αλλαγές στον κώδικα υλοποίησης. Μπορούν να υλοποιηθούν λειτουργίες χωρίς να χρειάζεται να γίνουν πολλές μη αυτόματες δοκιμές εάν ο κώδικάς έχει υψηλή κάλυψη από τα test που έχει δημιουργηθεί για τα κρίσιμα και εξελιγμένα σημεία του προγράμματος, καλή πρακτική είναι να δημιουργούνται τέτοιες δοκιμές. Οι getters και setters δεν είναι απαραίτητο να εξετάζονται αφού αυτοί είναι από τον πηγαίο κώδικα της Java. Εάν υλοποιούνται τεστς για μια υπάρχουσα βάση κώδικα που δεν έχει προηγουμένως, προτείνεται να ξεκινήσει η συγγραφή κώδικα σε τομείς που έχουν τα περισσότερα προβλήματα.
Υπάρχουν τρεις τύποι δοκιμών: unit, integration, και performance tests.

- Το unit τεστ είναι ένα κομμάτι κώδικα που αναπτύχθηκε από έναν προγραμματιστή, που εκτελεί μια καθορισμένη διαδικασία στον υπό δοκιμή κώδικα και επιβεβαιώνει μια συγκεκριμένη συμπεριφορά ή κατάσταση. Ένα Unit τεστ εξετάζει ένα μεμονωμένο κομμάτι κώδικα, όπως μια μέθοδο ή μια κλάση. Οι εξωτερικές εξαρτήσεις (dependencies) θα πρέπει να αντικατασταθούν σε δοκιμές από ένα (ψευδές) αντικείμενο που παράγεται στο πλαίσιο δοκιμής. Οι δοκιμές μονάδων είναι αναποτελεσματικές για την αξιολόγηση πολύπλοκων UI ή αλληλεπιδράσεων στοιχείων. Για αυτά θα πρέπει να δημιουργηθούν integration τεστς.
- Το integration test χρησιμοποιείται για την αξιολόγηση της συμπεριφοράς ενός στοιχείου ή της ενοποίησης μιας ομάδας στοιχείων. Οι δοκιμές αυτές διασφαλίζουν ότι ολόκληρο το σύστημα λειτουργεί όπως έχει προγραμματιστεί, μειώνοντας την ανάγκη για εκτεταμένες χειροκίνητες δοκιμές.
- Τα performance τεστ χρησιμοποιούνται επανειλημμένα για τη χρήση δοκιμών απόδοσης (benchmark). Στόχος τους είναι να βεβαιωθούν ότι ο υπό δοκιμή κώδικας εκτελείται αρκετά γρήγορα ακόμα και όταν έχει βαρύ φορτίο προς εκτέλεση.

5. ΠΑΡΟΥΣΙΑΣΗ ΣΥΣΤΗΜΑΤΟΣ

5.1 Υλοποίηση

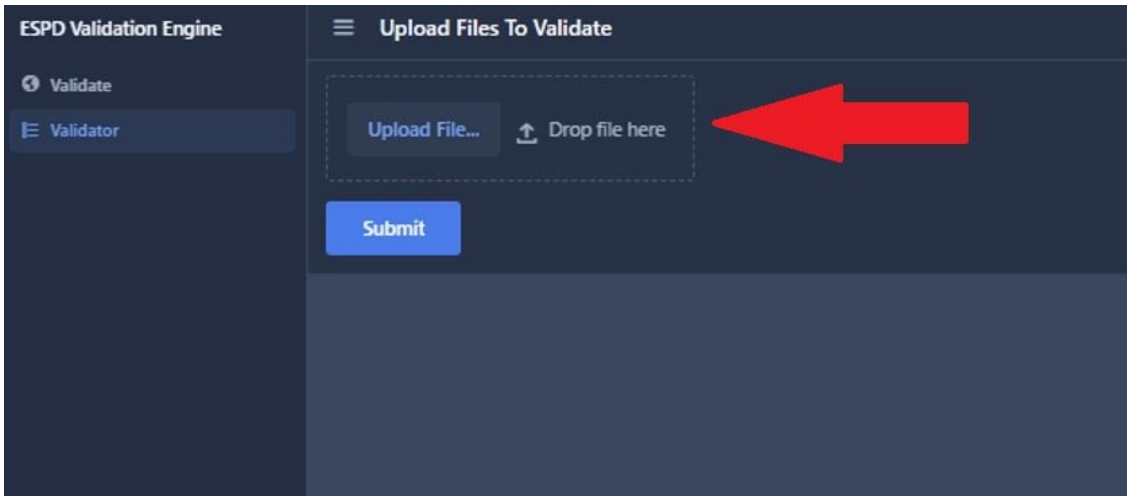
Στην υλοποίηση του ***ESPD Validation Engine*** με το framework του Vaadin, ως προς την εμφάνιση παρατηρείται πως έχει διασπαστεί η εφαρμογή σε περισσότερα από ένα components. Αριστερά υπάρχει ένα menu με διάφορα tabs όπως **Validate** και **Validator** ενώ πάνω από αυτά ο τίτλος όπως βλέπουμε στην Εικόνα 9.



Εικόνα 9:Validator

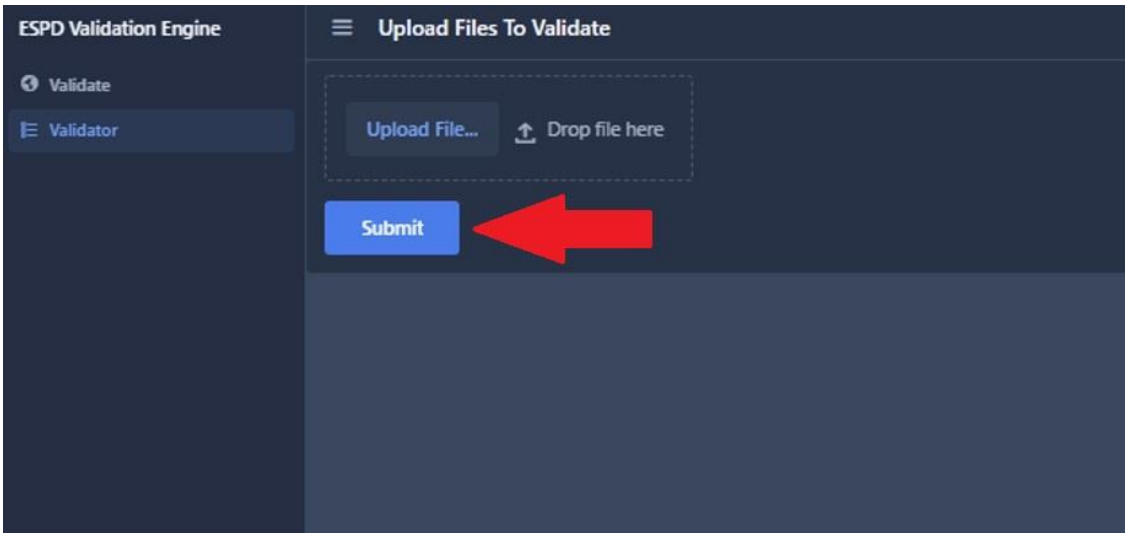
Στην δεξιά πάνω άκρη βρίσκεται το κουμπί με όνομα **Dark** όπου αλλάζει το χρώμα της εφαρμογής σε σκοτεινό ή φωτεινό πατώντας το.

Στο tab Validator έχει υλοποιηθεί η λειτουργία όπου μπορεί να προσπελαστεί το αρχείο xml με σκοπό να το διαβάσει ο validator στο back end και να βρει αν υπάρχουν λάθη στη συγκεκριμένη έκδοση. Στη συνέχεια πατώντας στο πεδίο “Drop file here” ο χρήστης μπορεί να ανεβάσει το xml αρχείο είτε επιλέγοντας το είτε κάνοντας drag and drop όπως στην Εικόνα 10. Όταν αυτό φορτωθεί θα εμφανιστεί από κάτω ένα ✓ με το όνομα του αρχείου δίπλα σε αυτό.



Εικόνα 10:Upload files

Πατώντας το κουμπί Submit όπως στην Εικόνα 11 ξεκινάει ο έλεγχος στο backend κομμάτι του κώδικα εμφανίζοντας τα Errors ξεχωριστά το κάθε ένα σε διαφορετική καρτέλα όπου το κάθε ένα έχει αναγνωριστικά πεδία όπως Location, Severity, Validation, Error καθώς και το ίδιο το πρόβλημα.



Εικόνα 11:Submit

Το πεδίο Location δείχνει τη γραμμή και τη στήλη που θα βρούμε το πρόβλημα που εντοπίστηκε από τον Validator μέσα στο αρχείο μας. Το πεδίο Severity φανερώνει τον τύπο του προβλήματος για παράδειγμα “Error” ενώ το πεδίο Validator το όνομα της κλάσης που έκανε το validation για παράδειγμα “SCHEMATRON”. Τέλος, παρατηρούμε το πρόβλημα που εντόπισε το πρόγραμμα όπως: “The element “cbc:Description” is mandatory”. Τα αποτελέσματα αυτά διακρίνονται στην Εικόνα 14.

Εικόνα 12:Σκούρα χρώματα εφαρμογής Validation

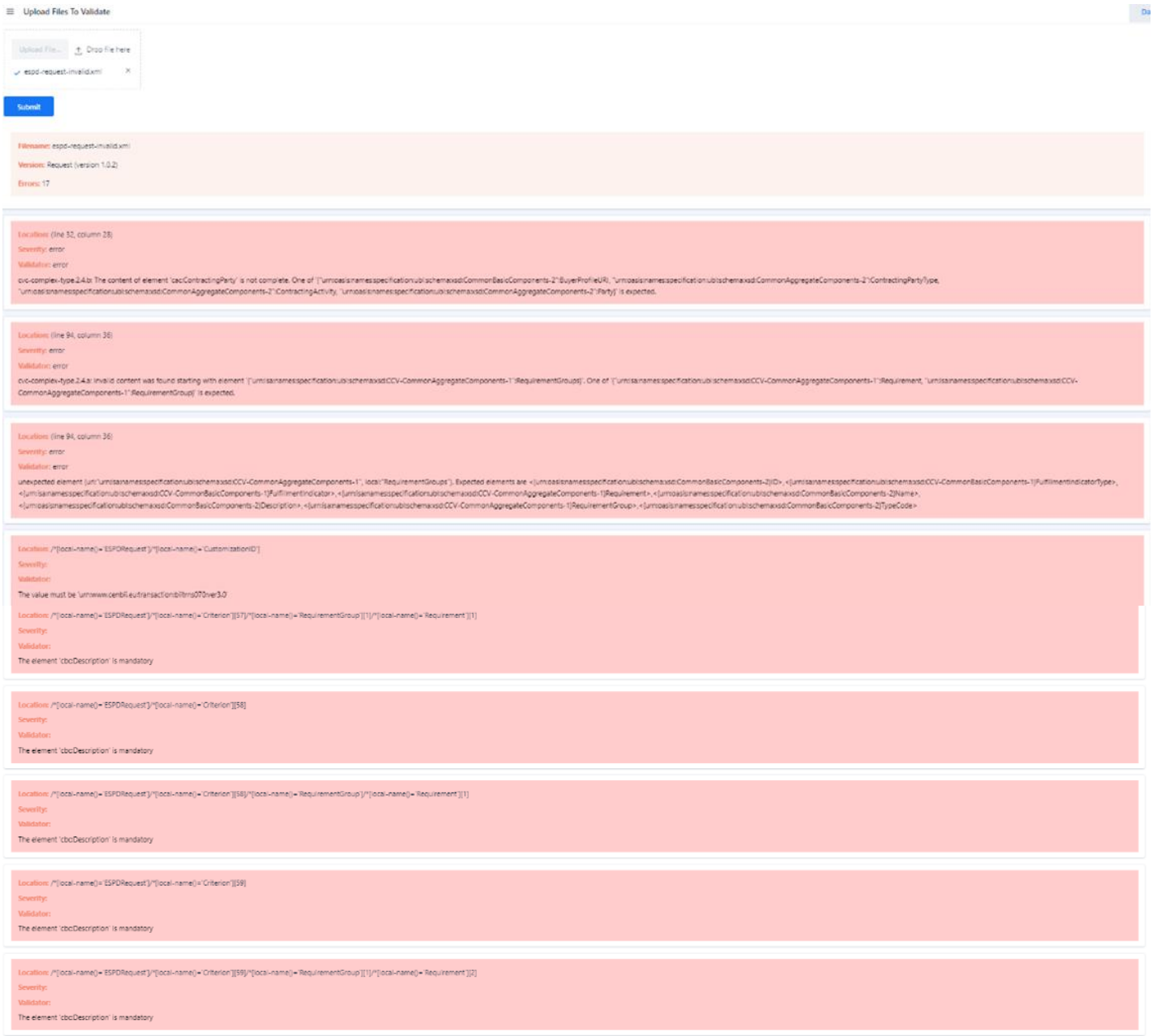
Εικόνα 13:Ανοιχτό χρώμα εφαρμογής Validation

5.2 Αποτελέσματα : Παράδειγμα Α

Όπως παρατηρείται στην Εικόνα 14 έχει δημιουργηθεί ένα πλαίσιο από layouts τύπου horizontal και vertical μέσα από το Vaadin με σκοπό να σχηματιστεί το πλαίσιο όπου θα παρουσιαστεί η υλοποίηση των αποτελεσμάτων. Μπορούν να διακριθούν τρία υπο-συστατικά: “Upload Files To Validate” με χρώμα άσπρο, “File Details” με χρώμα μπεζ και “καρτέλες” με χρώμα κόκκινο.

- Το πεδίο “Upload Files To Validate” περιέχει τη διεργασία αυτή ώστε να γίνει η φόρτωση του αρχείου xml που πρέπει να γίνει validate.
- Το πεδίο “File Details” όπου έχει εμφανιστεί έπειτα από την επιλογή “submit”, όπως περιγράφηκε προηγουμένως, περιέχει τρία τμήματα όπου χρησιμοποιούνται για να αποθηκεύσουν αντίστοιχα το όνομα του αρχείου, την έκδοση αυτού και τον αριθμό από τα συνολικά λάθη που βρέθηκαν σε όλο το αρχείο μέσω του validator.
- Για κάθε πρόβλημα που παρουσιάστηκε σαν αποτέλεσμα μέσω του validator έχει δημιουργηθεί μια καρτέλα όπου και περιέχει τέσσερα πεδία όπως:
    - Location: Γραμμή και στήλη προβλήματος.
    - Severity: Τύπος προβλήματος.
    - Validator: Τύπος ελέγχου.
    - Description: Το πρόβλημα.

Στα αποτελέσματα που παρουσιάζονται διακρίνεται το πεδίο “element cbcDescription is mandatory” όπου φανερώνει πως αυτό το πεδίο είναι υποχρεωτικό. Επιπλέον το πεδίο “SchemeAgencyId” είναι και αυτό υποχρεωτικό και δεν έχει συμπληρωθεί καθόλου κατά την οριστικοποίηση του αρχείου. Παρατηρείται μέσω των αποτελεσμάτων αυτών πως το validation έχει γίνει σωστά και έχει παρουσιάσει διαφορετικά αποτελέσματα για κάθε περίπτωση. Στην συγκεκριμένη περίπτωση εμφανίστηκαν 17 αποτελέσματα παρόμοιου ή διαφορετικού τύπου ανάλογα με την κατηγορία που ανήκουν Schema ή Schematron.



Εικόνα 14:Αποτελέσματα πρώτου παραδείγματος

5.3 Αποτελέσματα : Παράδειγμα Β

Στο παράδειγμα Β απεικονίζεται το xml αρχείο που είναι τύπου ESPD-Response σε σύγκριση με του παραδείγματος Α που ήταν Request, ενώ ο validator βρήκε πολύ λιγότερα λάθη. Στη συγκεκριμένη περίπτωση τα πεδία του Severity και του Validator γυρνάνε τιμές null και δεν εμφανίζεται κάτι στην φόρμα Εικόνα 15.

Εικόνα 15:Αποτελέσματα δεύτερου παραδείγματος

5.4 Παράδειγμα υλοποίησης του UI στο framework του Vaadin

Στην κλάση FileDetailsView υλοποιείται το πλαίσιο της Εικόνα 15 όπου περιέχονται τα πεδία FileName, Version και Errors. Αυτά γεμίζουν με την κατάλληλη πληροφορία μετά την ανάλυση των δεδομένων που κάνει η εφαρμογή. Σύμφωνα με το framework του Vaadin, όπως απεικονίζεται και στην Εικόνα 16 , η κλάση κάνει extend το VerticalLayout ώστε να βρίσκεται στο ίδιο πλαίσιο με κλάσεις που αφορούν τη συγκεκριμένη σελίδα του UI. Χρησιμοποιούνται components όπως Div και Span με σκοπό να γεμίσουν τις πληροφορίες-String που πρέπει σε μορφή Span όπως “FileName” και “Version”. Στη συνέχεια μέσα στον constructor προστίθεται στο αντίστοιχό τους Div για παράδειγμα fileNameDiv και αυτό με τη σειρά του στο detailsVertical Layout. Στο συγκεκριμένο παράδειγμα έχουμε μια υλοποίηση κώδικα όπου αναφέρεται σε css. Με αυτόν τον τρόπο μπορούμε να δώσουμε ένα όνομα στο συγκεκριμένο αυτό πεδίο ώστε να του περάσουμε ότι παραμέτρους θέλουμε μετέπειτα όσον αφορά το κομμάτι της μορφοποίησής του σε css. Επιπλέον το Vaadin μας δίνει τη δυνατότητα να γράψουμε inline css κώδικα όπου είναι στην ίδια γραμμή με την Java για τα συγκεκριμένα label, πράγμα που μας διευκολύνει σε αντίθεση με τον παραπάνω τρόπο σε ορισμένες περιπτώσεις. Όπως: submit.getStyle().set("background -color", "grey"); // inline css

```java
public class FileDetailsView extends VerticalLayout {

    VerticalLayout detailsVertical = new VerticalLayout();


    private Div fileNameDiv = new Div();
    private Span fileNameLabel = new Span( text: "Filename: ");
    private Span fileNameValue = new Span( text: "");

    private Div versionDiv = new Div();
    private Span versionLabel = new Span( text: "Version: ");
    private Span versionValue = new Span( text: "");


    private Div errorDiv = new Div();
    private Span errorLabel = new Span( text: "Errors: ");
    private Span errorValue = new Span( text: "");


    FileDetailsView() {
        fileNameDiv.add(fileNameLabel, fileNameValue);
        detailsVertical.add(fileNameDiv);

        versionDiv.add(versionLabel, versionValue);
        detailsVertical.add(versionDiv);

        errorDiv.add(errorLabel, errorValue);
        detailsVertical.add(errorDiv);
        //css for bold
        fileNameLabel.setClassName("card-error-label");
        versionLabel.setClassName("card-error-label");
        errorLabel.setClassName("card-error-label");
        detailsVertical.setClassName("file-details");
        add(detailsVertical);
    }

    public void setFileNameValue(String fileNameValue) { this.fileNameValue.setText(fileNameValue); }

    public void setVersionValue(String versionValue) {
        this.versionValue.setText(versionValue);
    }
```

Εικόνα 16:Κλάση FileDetailsView

6. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΒΕΛΤΙΩΣΕΙΣ

Ανακεφαλαιώνοντας, το Ευρωπαϊκό Έγγραφο Ενιαίας Σύμβασης, ως απόδειξη όλων των προϋποθέσεων των δημοσίων συμβάσεων της ΕΕ, δηλώνει με έναν εύκολο τρόπο την θέση και τις δυνατότητες που έχει η εκάστοτε διαγωνιζόμενη εταιρία σε έναν πανευρωπαϊκό διαγωνισμό. Μετά τη σωστή συμπλήρωση της σύμβασης αλλά και τον έλεγχο που υλοποιήθηκε από τον validator για τον εντοπισμό τυχόν λαθών, έχουμε ένα έγγραφο όπου πληρούνται τα κριτήρια που έχει θέσει η αναθέτουσα αρχή, δεν απαιτούνται περαιτέρω αποδείξεις και μπορεί να χρησιμοποιηθεί σε προσφορές που υποβάλλονται στα κράτη της Ευρωπαϊκή Ένωσης.

6.1 Μελλοντικές βελτιώσεις με την Vue.js

Με την αλματώδη πρόοδο της τεχνολογίας είναι σίγουρο πως σε μερικά χρόνια θα χρειαστεί συντήρηση του συστήματος λόγω νέων τεχνολογιών ως προς το κομμάτι του frontend αλλά και του backend, αλλά και ως προς των νέων κανονισμών που θα πρέπει να ενσωματωθούν κάθε φορά που αυτοί ανανεώνονται. Γι’ αυτό το λόγο μια διαφορετική προσέγγιση το ιδίου προβλήματος θα ήταν με μια νεότερη για τα δεδομένα τεχνολογία όπως αυτή της Vue Js [20] με σκοπό την αποσύνδεση της εφαρμογής από μια μόνο τεχνολογία όπως έχει υλοποιηθεί αυτή την στιγμή με το Vaadin. Το Vaadin ίσως να είναι ένα από τα κορυφαία framework όσον αφορά μόνο τη Java αλλά είναι σύνηθες και προτείνεται στους προγραμματιστές να μην γίνεται κατασκευή του κώδικα σε εκδόσεις που είναι μονομερείς.
Η Vue.js είναι μια γρηγορότερη, απλούστερη και πιο κομψή έκδοση των framework της React και Angular JS ενώ έχει συγκεντρώσει πολλούς θαυμαστές σε πολύ σύντομο χρονικό διάστημα για την ανάπτυξή της. Επιπλέον λειτουργεί εξ ολοκλήρου μέσω της αναπτυσσόμενης κοινότητας ανοιχτού κώδικα σε σύγκριση με την Angular που υποστηρίζεται από την Google και την React από το Facebook.

7. ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	Ελληνικός Όρος
European Single Procurement Document	Ευρωπαϊκό Έγγραφο Ενιαίας Σύμβασης
Vaadin	Το Vaadin είναι ένα framework για τη δημιουργία εφαρμογών σε Java
Spring	Το Spring είναι ένα Framework που παρέχει ολοκληρωμένη υποστήριξη υποδομής για την ανάπτυξη εφαρμογών σε Java
Validator	Είναι ένα πρόγραμμα που χρησιμοποιείται για τον έλεγχο της εγκυρότητας ή της συντακτικής ορθότητας ενός τμήματος κώδικα ή εγγράφου.
eCertis	Online ιστότοπος όπου μάς βοηθά να κατανοήσουμε τι είδους αποδεικτικά στοιχεία μπορούμε να χρησιμοποιήσουμε για να εκπληρώσουμε μια συγκεκριμένη απαίτηση σε μια ευρωπαϊκή χώρα.
Java	Γλώσσα προγραμματισμού
Components	Συστατικά μέρη
Designer	Σχεδιαστής
Charts	Διαγράμματα
TestBench	Πρόγραμμα Υλοποίησης των Τεστ
Intefaces	Το σημείο της αλληλεπίδρασης και της επικοινωνίας ανθρώπου-υπολογιστή σε μια συσκευή
JavaScript	Γλώσσα προγραμματισμού
IntelliJ IDEA	Ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών
Eclipse	Ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών
NetBeans	Ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών
Backend	Το Back end είναι το λειτουργικό μέρος μιας επιχείρησης
Javadoc	Δημιουργία τεκμηρίωσης κώδικα Java σε μορφή HTML από πηγαίο κώδικα Java
RESTful web services	Representational State Transfer
Jakarta EE	Πρώην πλατφόρμα Java
MainView	Η Αρχική κλάση
VerticalLayout	Κάθετη Διάταξη
HorizontalLayout	Οριζόντια Διάταξη
FormLayout	Διάταξη φόρμας
SplitLayout	Διαχωριστική Διάταξη
Constructor	Κατασκεύαστης

Data Binding	Η διαδικασία που δημιουργεί μια σύνδεση μεταξύ της εφαρμογής και των δεδομένων που εμφανίζει
JavaBean	Κλάσεις που ενσωματώνουν πολλά αντικείμενα σε ένα μόνο αντικείμενο
Drag and drop	Μεταφορά και απόθεση αρχείων
Web Components	Συστατικά παγκοόσμιου ιστού
OSGi Support	Υποστήριξη OSGi λειτουργίας
Vaadin TestBench	Πρόγραμμα εκτέλεσης των τεστ
Element API	
DOM	Μοντέλο xml αρχείων
Repository	Αποθήκη κώδικα
Issues	Προβλήματα προς υλοποίηση στην εφαρμογή GitLab
Merge request	Αίτημα συγχώνευσης
ESPD Request	Αίτημα εντύπου αυτοδήλωσης που χρησιμοποιείται στις διαδικασίες δημοσίων συμβάσεων.
ESPD Response	Το ESPD αποτελείται από ένα ESPD Request το οποίο δημιουργείται και εκδίδεται από τον αγοραστή στο οποίο ο προμηθευτής μπορεί να υποβάλει ESPD Response
Core Criterion and Core Evidence Vocabulary	Λεξιλόγιο αποδεικτικών στοιχείων

8. ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

BIS	Business Interoperability Specification
CCCEV	Core Criterion and Core Evidence Vocabulary
CEN/BII	CEN Workshop on Business Interoperability Interfaces for public procurement in Europe
DevOps	Development Operations
DG GROW	Directorate-General for Internal Market
DOM	Document Object Model
EE	Ευρωπαϊκή Ένωση (European Union)
HTML	Hypertext Markup Language
IDE	Integrated Development Environment
IRM	Information Requirements Model
JVM	Java Virtual Machine
POM	Project Object Model
PWA	Progressive Web Application
TOOP	The Once And Only Principle
UBL	Universal Business Language
UI	User Interface
UML	Unified Modeling Language
XML	Extensible Markup Language

9 .ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

[1] European Commision, “European Single Procurement Document (ESPD)”, 24 Aug 2016, https://ec.europa.eu/isa2/solutions/european-single-procurement-document-espd_en. [Προσπελάστηκε 8/2/2022]

[2] European Commision,“European Single Procurement Document and eCertis”, 24 Aug 2016, https://ec.europa.eu/growth/single-market/public-procurement/digital-procurement/european-single-procurement-document-and-ecertis_en. [Προσπελάστηκε 10/2/2022]

[3] A. Schmitz, “Interoperable ESPD and VCD SERVICES”, 10 nov 2017https://wiki.ds.unipi.gr/display/ESPDInt/Homepage. [Προσπελάστηκε 8/1/2021]

[4] P. Deitel,H. Deitel,“Java How to Program”, 2 April 2003, Deitel 6th edition.

[5] Mj. Young, ”Xml βήμα βήμα” , 2 Feb 2007, εκδόσεις κλειδάριθμος.

[6] A. Schmitz, “Part II – ESPD – VCD System Architecture”, 20 Nov 2017,https://wiki.ds.unipi.gr/display/ESPDInt/Part+II+-+ESPD-VCD+System+Architecture . [Προσπελάστηκε 12/1/2021]

[7] Interoperability Test Bed,”ISA Validator”, https://www.itb.ec.europa.eu/espd/upload. [Προσπελάστηκε 7/3/2022]

[8] The MIT Licence, “Angular js”, 10 Feb.2021, https://angularjs.org/. [Προσπελάστηκε 10/2/2022]

[9] Ulf Lotzmann, “BIS 41- ESPD Version (1.0.2)”, 14 May 2018, https://wiki.ds.unipi.gr/pages/viewpage.action?pageId=47219235. [Προσπελάστηκε 7/3/2022]

[10] M. Widenius“, “Vaadin - Training Videos”, 25 Jun 2018, https://vaadin.com/learn/training. [Προσπελάστηκε 5/10/2019]

[11] Vaadin,”Overview”,3/02/2021,https://vaadin.com/docs/v8/framework/architecture/architecture-overview. [Προσπελάστηκε 7/3/2022]

[12] Vaadin,“ComponentsOverview”,04/02/2021,https://vaadin.com/docs/v7/framework/components/components-overview [Προσπελάστηκε 7/3/2022]

[13] Marcelo D. Re,"Chart library for Vaadin Flow 14" 02,03,2021, https://vaadin.com/directory/component/chart-library-for-vaadin-flow-14 [Προσπελάστηκε 7/3/2022]

[14] Vaadin, "Running Tests in a Distributed Enviroment", 02/03/2021, https://vaadin.com/docs/v8/testbench/environment/testbench-grid , [Προσπελάστηκε 7/3/2022]

[15] M.Youngstrom, "SpringBoot project initializer", 10 Oct 2021, https://start.spring.io/. [Προσπελάστηκε 8/2/2022]

[16] M. Widenius, "Vaadin Start",25 Jun 2018, https://start.vaadin.com . [Προσπελάστηκε 10/2/2021]

[17] K. Ryder, "Maven,13 Jul 2004, https://maven.apache.org/. [Προσπελάστηκε 9/8/2021]

[18] H. Docter, "Gradle,2 Apr 2008, https://gradle.org/. [Προσπελάστηκε 9/8/2021]

[19] K. Beck, E. Gamma, D. Saff, K. Vasudevan, "JUnit – Framework",28 Nov 2021, https://junit.org/junit4/. [Προσπελάστηκε 10/8/2021]

[20] E. You,"Vue js" 21 Jan 2014 , https://vuejs.org/. [Προσπελάστηκε 10/2/2022]

[21] Gov," Προμηθεύς" 4 Feb 2022, https://espdint.eprocurement.gov.gr/. [Προσπελάστηκε 10/2/2022]