ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΑΣ, ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Εργαστήριο Γνώσης και Αβεβαιότητας

Μεταπτυχιακή Εργασία

Ηλεκτρονικό πρωτόκολλο

Τριανταφυλλάκης Ιωάννης
202220200025

Επιβλέποντες:

Εμμανουήλ Γουάλλες
Επίκουρος Καθηγητής

Βασίλης Πουλόπουλος
Επίκουρος Καθηγητής

Τρίπολη, Μάρτιος, 2022

Συμβουλευτική Επιτροπή:

1. Εμμανουήλ Γουάλλες,Επίκουρος Καθηγητής
2. Βασίλης Πουλόπουλος,Επίκουρος Καθηγητής

.

Εγκρίθηκε από την εξεταστική επιτροπή την . . . <συμπληρώνετε ημερομηνία παρουσίασης της εργασίας - βρίσκεται στο αρχείο dit-thesis.cls>.

1.Εμμανουήλ Γουάλλες, Επίκουρος Καθηγητής

2.Βασίλης Πουλόπουλος,Επίκουρος Καθηγητής

.....

Τριανταφυλλάκης Ιωάννης
Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Πανεπιστήμιο Πελοποννήσου

Η παρούσα εργασία εκπονήθηκε σε συνεργασία με το Εργαστήριο Γνώσης και Αβεβαιότητας (ΓΑΒ LAB).

Το Ηλεκτρονικό Πρωτόκολλο είναι ένα ολοκληρωμένο μηχανογραφικό περιβάλλον που αναλαμβάνει τη διαχείριση της αλληλογραφίας αντικαθιστώντας το χειρόγραφο σύστημα πρωτοκόλλου με ηλεκτρονικό. Απευθύνεται σε όλους τους οργανισμούς που διακινούν έγγραφα με οποιοδήποτε τρόπο (συμβατικό ταχυδρομείο, FAX, e-mail κλπ).Ενσωματώνει τις πλέον σύγχρονες τεχνολογίες πληροφορικής εξασφαλίζοντας αξιόπιστη διαχείριση της αλληλογραφίας. Το Ηλεκτρονικό Πρωτόκολλο προσφέρει δυνατότητες που οδηγούν στην ολοκληρωμένη και σύννομη μηχανογραφική λειτουργία στον τομέα της διαχείρισης των εγγράφων.

Abstract

In this project, we tried to make the communication between users and sectors more flexible. The Electronic Protocol is a complete computer environment that undertakes the management of mail, replacing the handwritten protocol system with an electronic one. It is addressed to all organizations that circulate documents in any way (conventional mail, FAX, e-mail, etc.). It integrates the most modern information technologies ensuring reliable mail management. The Electronic Protocol offers possibilities that lead to the complete and legal computer operation in the field of document management.

Πίνακας περιεχομένων

Εισαγωγή

Στην παρούσα διπλωματική εργασία προσπαθήσαμε να κάνουμε την επικοινωνίαμεταξύ χρη στών και φορέων πιο ευέλικτη απο τους είδη σύνηθες τρόπους που έχουν επιλεχθεί. Καταφέραμε να βελτιστοποιήσουμε ένα είδη υπάρχων μοντέλο επικοινωνίας που έχει αποδείξει σε βάθος χρό νου την χρησιμότητα του καθώς και την ευελιξία του.

Το πρόβλημα που καλεστήκαμε να λύσουμε εμπνεύστηκε από τους επιβλέποντες καθηγητές μου έχοντας εντοπίσει τις περισσότερες αδυναμίες καθώς και βελτιώσεις που θα χρειαζόντουσαν ώστε η επικοινωνία φορέων και χρηστών να γίνει πιο άμεση καθώς και πιο επιδέξια. Επέλεξα να συμμετάσχω σε αυτό τον προβληματισμό διότι από το προπτυχιακό επίπεδο είχα τη θέληση να συμμετάσχω σε ένα έργο το οποίο δημιουργούσε μεγαλύτερη ευχέρειά στις ζωές των ανθρώπων. Η μεθοδολογία ανάπτυξης που ακολουθήθηκε θα μπορούσαμε να την συγκρίνουμε με κάποιο μοντέλο περισσότερο Aggile διότι χωρίστηκε στο κατασκευαστικό κομμάτι καθώς και στο κομ μάτι ανάλυσης. Τα προβλήματα και οι δυσκολίες που συναντήσαμε κατά την διάρκεια της ήταν αρκετά διότι έπρεπε σε πρώτο στάδιο να μελετηθούν είδη υπάρχουσες υλοποιήσεις καθώς και να εναρμονιστούν με τους ρυθμούς και τον όγκο που παράγεται σε καθημερινό επίπεδο απο τους φορείς καθώς και τους χρήστες. Εάν καλούμασταν να ξεκινήσουμε εκ νέου την υλοποίηση της παρούσας πτυχιακής εργασίας, θα μπορούσαμε με σιγουριά να ανακαλύψουμε περισσότερες αδυναμίες και ελλείψεις επειδή οι ανάγκες των οργανισμών καθώς και των χρηστών θα ήταν διαφορετικές αφού αλλάζουνε σε πραγ ματικό επίπεδο χρόνου.

Οι μελλοντικές λύσεις που σημειώθηκαν καθώς και παρατηρήσεις από τις δύο πλευρές(ερευνητικό -παραγωγή έργου) ήταν η εναλλαγή της αρχιτεκτονικής απο Monolithic σε Gateaway Microservice όπως μπορείτε να δείτε παρακάτω.

Κεφάλαιο 1

Μηχανογράφιση και Ηλεκτρονικό Πρωτόκολλο

Παρατηρώντας την φωτογραφία εάν καλούμασταν να πούμε τι βλέπουμε, θα μπορούσαμε να διακρίνουμε δεδομένα τα οποία είναι απαραίτητα να φυλαχθούν ώστε να χρησιμοποιηθούν σε μελλοντικό χρόνο .

Τα δεδομένα μπορεί να είναι σημεία πληροφοριών και να περιλαμβάνουν λέξεις - έννοιες, αριθμούς, σύμβολα, διαγράμματα, σχέδια, φωτογραφίες, μαγνητοταινίες κλπ που περιγράφουν ή αντιπροσωπεύουν ποσότητες, έννοιες, ιδέες, αντικείμενα, γεγονότα, καταστάσεις και λειτουργίες. Ενδεχομένως, κάποιοι από τους τύπους δεδομένων που παρατίθενται να εμπεριέχουν ήδη εμφανείς πληροφορίες. Ωστόσο, οι πληροφοριες παραμένουν ανεπαρκείς.

Τα δεδομένα αναγράφουν καποίο μέρος ενός συμβάντος, δεν περιλαμβάνουν καμία ανάλυση, κριτική ή αξιόπιστη βάση για περαιτέρω ενέργεια. Αντιθέτως, δεν αναγράφουν με εμφανή τρόπο τη σημαντικότητά τους. Τέλος, δεν εμφανίζουν πληθώρα πληροφοριών καθώς είναι οργανωμένα για ένα συγκεκριμένο σκοπό και υπόκεινται σε ανάλυση και αξιολόγηση.

1.1 Μηχανογράφιση πριν το 1996

Η μεγαλύτερη σύγχυση που μπορεί να αναφερθεί στα δεδομένα είναι η χαρτογράφηση τους. Τα δεδομένα σε παλαιότερα έτη αποτυπωνόντουσαν σε χαρτί και εάν θέλαμε να μεταφέρουμε την πληροφορία θα έπρεπε να δώσουμε αυτό το χαρτί ώστε να την διαβάσει ο αναγνώστης, πολύ γρήγορα σημειώθηκαν προβλήματα στην μεταφορά της πληροφορίας. Ο συντάκτης που αποτυπώνει την πληροφορία χρησιμοποιούσε ένα μέσω το οποίο παράγεται από δέντρα, καταλαβαίνουμε ευθείς αμέσως ότι ο κατακερματισμός του περιβάλλοντος αυξάνεται σε επικίνδυνό βαθμό αφού

όλοι μας έχουμε την ανάγκη να συντάξουμε πληροφορίες καθώς και να επαναλάβουμε αυτή την ενέργεια πολλαπλές φορές μέσα στην μέρα. Υστέρα η πληροφορία αυτή έπρεπε να δοθεί και εκεί αναπτυσσόταν το πρόβλημα της ακεραιότητας του αρχείο διότι κατά την μεταφορά του θα μπορούσε να τροποποιηθεί, αλλοιωθεί , διαγραφεί. Έστω ότι η πληροφορία έφθανε ακέραια ως προς τον αναγνώστη και ήθελε να την μεταβιβάσει σε ένα τρίτο άτομο θα έπρεπε να επαναληφθεί όλη η διαδικασία από την αρχή , αθροίζοντας όμως τις πιθανότητες λάθους.

1.2 Μηχανογράφιση μετά το 1996

Διαβάζοντας την προηγούμενη παράγραφο ένας άνθρωπος γεννηθείς από το 1996 και ύστερα θα αναρωτιέται γιατί δεν διαμοιραζόταν το αρχείο του μέσω κάποιου καναλιού μέσω παγκόσμιου ιστού;

Ο παγκόσμιος ιστός ανέπτυξε την επικοινωνία σε επίπεδο σε ακέραιας μεταφοράς της πληροφορίας από την μία άκρη του κόσμου σε μία άλλη μέσα σε κλάσματα του δευτερόλεπτου. Παρόλα αυτά όμως η μηχανογράφηση των αρχείων δεν κατάφερε να ενσωματωθεί τόσο γρήγορά. Οι προγραμματιστές σε καθημερινό επίπεδο προσπαθούν να βελτιώνουν πολλαπλά κομμάτια τους μέσω καναλιών επικοινωνίας τα οποία έρχονται με μια διεπαφή ευέλικτη για τον καθημερινό χρήστη.

Η μηχανογράφηση εν έτη 2021 έχει προσαρμοστεί σε ένα μεγάλο επίπεδο παρόλο αυτά δεν έχει ακουμπήσει τις βέλτιστες τιμές του αφού παρατηρούμε ότι υπάρχει η ανάγκη για ανακάλυψη νέων εργαλείων. Μέρα με την μέρα παρατηρούμε ότι δημιουργούνται καινούργιες βάσεις δεδομένων ώστε να αποθηκευτούν αυτά τα αρχεία καθώς και κανάλια επικοινωνίας ώστε τα δεδομένα μας να μεταφέρονται με ασφάλεια .

1.3 Ηλεκτρονικό πρωτόκολλο

Το ηλεκτρονικό πρωτόκολλο είναι μια ακόμα προσθήκη των εργαλείων όπως αναφέραμε σε προηγούμενες ενότητες, προσαρμόζει τις μεθοδολογίες και του στόχους του κλασσικού πρωτοκόλλου που όλοι γνωρίζουμε αλλά έρχεται να επιλύσει και επιμέρους προβλήματα βάση αναγκών .

Πιο αναλυτικά το ηλεκτρονικό πρωτόκολλο καλύπτει όλες τις ανάγκες διαχείρισης σε μια και μόνο οθόνη, με σκοπό την αποτελεσματική διαχείριση εγγράφων και είναι πλήρως παραμετροποιήσιμη για τις ανάγκες του κάθε οργανισμού. Τα πλεονεκτήματα που επιλύει είναι αρκετά ώστε να το κάνουν αρκετά ευέλικτο ως προς την χρήση του. Τα βασικά που θα μπορούσαμε να σημειώσουμε είναι

- Δυνατότητα άμεσης σάρωσης εγγράφων μονοσέλιδη και πολυσέλιδη
- Δυνατότητα εκτύπωσης αποδεικτικού παραλαβής
- Δυνατότητα καταγραφής και παρακολούθησης ιστορικού ενεργειών διεκπεραίωσης
- Μηχανισμός ειδοποιήσεων (email, sms κτλ.) σε οποιοδήποτε αποστολέα
- Ενσωμάτωση του Υπηρεσιακού Οργανογράμματος στη διαδικασία της ηλεκτρονικής διακίνησης

Ακούγοντας την λέξη ηλεκτρονικό πρωτόκολλο σε πολλούς από εμάς θα μας φαίνεται γνωστή. Δεν είμαστε οι οποίοι δημιουργούμε την μεθοδολογία του ηλεκτρονικού πρωτοκόλλου αφού σε παρελθοντικούς χρόνους αρκετοί προγραμματιστές έλυσαν με επιτυχία αρκετά προβλήματα. Μια ολοκληρωμένη διαχείριση εγγράφων θα μπορούσε να βρούμε και στον κόσμο του λογισμικού ανοικτού κώδικα. Ένα χαρακτηριστικό παράδειγμα το οποίο θα μπορούσαμε να συγκρίνουμε την υλοποίηση μας είναι το e-ΛΕΚΤΡΟΝΙΚΟ ΠΡΩΤΟΚΟΛΛΟ το οποίο έχει αναπτυχθεί . Η σύγκριση

μεταξύ των δύο αυτών υλοποιήσεων είναι ότι στην δική μας περίπτωση έχουμε καταφέρει να μπορούμε να το χρησιμοποιήσουμε από οποιαδήποτε συσκευή έχει πρόσβαση στον παγκόσμιο ιστό και όχι μόνο από υπολογιστή.

Ακόμα η αποθήκευση των δεδομένων μας όπως θα δούμε παρακάτω γίνεται σε πραγματικό χρόνο και όχι σε μία συγκεκριμένη ώρα μέσα στην ημέρα, όπως μπορούμε να καταλάβουμε αυτό φέρνει ακόμα πιο κοντά την επικοινωνία και την γρηγορότερη επίλυση των προβλημάτων . Επιπλέον, αξίζει να σημειωθεί ότι οι απαιτήσεις συστήματος είναι αρκετά μικρότερες και αυτό μας βοηθά στην μετεξέλιξη του. Παράλληλα με αυτό έχει χρησιμοποιηθεί και μια διεπαφή όπως θα δούμε παρακάτω ώστε να μπορούμε να το συνδυάσουμε με είδη υπάρχων πλατφόρμες το οποίο λείπει από την υλοποίηση που έχει γίνει στο παρελθόν.

Επιπρόσθετο χαρακτηριστικό θα μπορούσε να σημειωθεί ότι μπορούν ταυτόχρονα πολλαπλοί φορείς να το χρησιμοποιήσουν μεταξύ τους χωρίς να πρέπει να το στεγάσουν ξεχωριστά.

Κεφάλαιο 2

Σύγχρονη Υλοποίηση Μηχανογράφησης

Η μηχανογράφηση σήμερα δεν χρησιμοποιείται όπως είδαμε αποκλειστικά για την συλλογή πολυμέσων σε ψηφιακή μορφή αλλά έχει πολλαπλές καθώς και πολύπλοκες χρήσεις.Την παρούσα στιγμή υπάρχουν open source λύσεις καθώς και closed source υλοποιήσεις.

Λέγοντα open source αναφερόμαστε σε υλοποιήσεις όπου ο/οι δημιουργοί τους διέμεναν δωρεάν στο ευρύ κοινό για οποιαδήποτε χρήση είτε για συγκεκριμένες λύσεις ανάλογα όπως αυτοί ορίζουν. Διάφορες εταιρείες έχουν διανέμει κατόπιν τιμής διάφορες υλοποιήσεις όπου προσπαθούν να λύσουν όπως θα δούμε παρακάτω είτε συγκεκριμένα προβλήματα είτε να να βελτιστοποιήσουν τα είδη υπάρχοντα.

2.1 Open Source

Ξεκινώντας απο τις open source λύσεις δεν θα μπορούσαμε να μην συμπεριλάβουμε την μεγαλύτερη Ελληνική προσπάθεια που έχει γίνει απο τον κ.Χρήστο Μπερέτα όπου όπως αναφέρει στο site του https://christosberetas.com κατεβάζοντας τον πηγαίο κώδικα της εφαρμογής μας δίνεται η δυνατότητα για διάφορες ενέργειες :

1) Δύο ειδών πρωτοκολλήσεις, στην πρώτη περίπτωση δίνει τη δυνατότητα στον χρήστη να πραγματοποιεί συνεχείς πρωτοκολλήσεις, δηλαδή να μην μηδενίζει (ο αύξοντας αριθμός πρωτοκόλλου) μετά το τέλος του έτους και να γίνεται συνεχής πρωτοκόλληση (σε αυτή την περίπτωση προτείνω να δημιουργηθεί ένα έτος 0000 και να διαγράψετε τα υπόλοιπα έτη που πιθανών δεν χρειάζονται). Στη δεύτερη περίπτωση, μετά το τέλος του έτους ο χρήστης δημιουργεί ένα νέο έτος και αυτό θα γίνεται κάθε χρόνο.ώστε τα πρωτόκολλα θα ταξινομούνται ανά έτος και κάθε νέα χρονιά η πρωτοκόλληση θα ξεκινά από την αρχή.

2) Αναζήτηση εισερχομένου πρωτοκόλλου με βάση την ημερομηνία καταχώρησης.

3) Αναζήτηση εξερχόμενου πρωτοκόλλου με βάση την ημερομηνία.

4) Τροποποίηση υπάρχοντος πρωτοκόλλου.

5) Διόρθωση κάποιον προβλημάτων στον κώδικα που εντοπίστηκαν που δεν αφορούν όμως την ακεραιότητα τον πληροφοριών που έχετε πρωτοκολλήσει μέχρι τώρα.

6) Επιπλέον σταθερότητα της εφαρμογής σε νεότερης γενιάς λειτουργικά συστήματα.

7) Δυνατότητα συμπίεσης της βάσης δεδομένων για εξοικονόμηση χώρου.

8) Δυνατότητα εξαγωγής όλων τον πρωτοκόλλων σε TXT αρχείο ανά έτος.

9) Δυνατότητα επισύναψης τον εξής τύπων αρχείων (PDF, DOC, DOCX, TXT, XLS, XLSX, PPT, BMP, GIF, JPG)

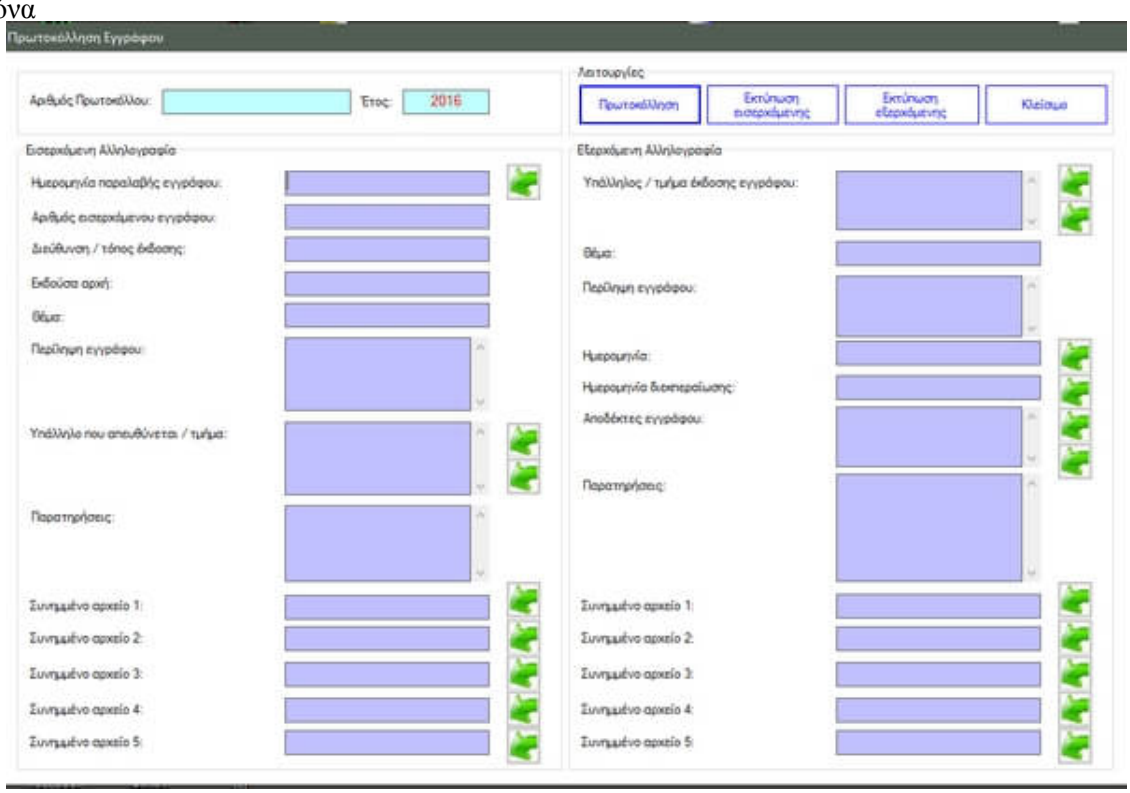
10) Αντίγραφα ασφαλείας και όχι όλη την βάση δεδομένων.

11) Πρωτοκόλληση από απόσταση μέσω Email.

12)Βάση δεδομένων τοπικά σε Server ή σε Cloud

Η εφαρμογή από ότι μπορούμε να δούμε στο ηλεκτρονική σελίδα του είναι σε native περιβάλλοντα δηλαδή για εφαρμογές τις όποιες ο χρήστης πρέπει να κάνει εγκατάσταση και όχι να τις επισκεφτεί μέσω κάποιου φυλλομετρητή(browser).

Οπτικά βλέπουμε μια γραφική διασύνδεση όμοια της αρχικής όπως βλέπουμε στην παρακάτων εικόνα



2.2 Closed source

Αναζητώντας υλοποιήσεις ηλεκτρονικού πρωτοκόλλου παρατηρήσαμε τις περισσότερες από αυτές να ειναι Closed Source. Closed Source ονομάζονται οι υλοποιήσεις τις οποίες οι δημιουργοί τους δεν επιτρέπουν καμία αλλαγή στον πηγαίο κώδικα καθώς και η άδεια χρήσης τους είναι ρητά συνδεμένη με την εταιρεία που το ανέπτυξε.

2.2.1 Ελληνικές Υλοποίησης

Αρκετές εταιρείες στον χώρο της πληροφορικής έχουν προσπαθήσει να υλοποιήσουν λύσεις σε συγκεκριμένα προβλήματα. Δυστυχώς κάποιες από αυτές δεν έχουν συντηρηθεί για αρκετά χρόνια. Μια αρκετά ανανεωμένη υλοποίηση για Desktop εφαρμογές είναι από την εταιρεία Έπαφος Έπαφος

Η εταιρεία Έπαφος έχεις αναπτύξει ένα ηλεκτρονικό πρωτόκολλο σε Desktop περιβάλλοντα για χρήστες Windows. Βλέποντας τον οδηγό χρήσης που παρέχουν μπορούμε να καταλάβουμε ότι μια απο τις τεχνολογιές που έχουν χρησιμοποιηθεί για την αποθήκευση των δεδομένων μας ειναι αντικειμενοστρεφής βάσης δεδομένων . καθώς και οι δυνατότητες που δίνουν στον τελικό χρήστη

1.Διαχείριση χρειστών
2.Εκτύπωση αρχείων
3.Προσθήκη-Διαγραφή αρχείων
4.Διόρθωση αρχείων
5.Ταξινόμηση αρχείων
6.Προεπισκόπηση αρχείων

2.2.2 Διεθνείς Υλοποίησης με δυνατότητα πολλαπλών γλωσσών

Ελκυστική υλοποιήσης στο πρόβλημα της μηχανογράφησης καθώς και του ηλεκτρονικού πρωτοκόλλου έχει δωθεί απο την εταιρεία Key Solution της οποίας η έδρα της ειναι στην Αμερική. Η Key solution παρέχει διάφορες υλοποιήσεις στους αγοραστές τις καθώς και ένα από τα μοναδικά χαρακτηριστικά της είναι η δυνατότητα πολλαπλών γλωσσών.

Η μείζων διαφορά της έναντι άλλων υλοποιήσεων βρίσκεται στην παροχή της των συγκεκριμένων χαρακτηριστικών:

1.Έξυπνη διαχείριση φορμών
2.Online υποβολή πρωτοκόλλου
3.Υποστηρίζει κριτικές, τροποποιήσεις, ανανεώσεις, ανεπιθύμητες ενέργειες και αποκλίσεις
4.Κριτική χωρίς χαρτί(Paperless review)
5.Δημιουργία ημερήσιας διάταξης συνεδρίασης και καταγραφή πρακτικών συνεδρίασης
6.Τυπικές και ad hoc αναφορές
7.Διαχείριση πρωτοκόλλου και έλεγχος έκδοσης
8.Ολοκληρωμένες διαδρομές ελέγχου

Κεφάλαιο 3

Αρχιτεκτονική και Τεχνολογίες

3.1 Εισαγωγή

Στην παρούσα υλοποίηση του ηλεκτρονικού πρωτοκόλλου προσπαθήσαμε να δημιουργήσουμε έναν τρόπο ώστε να είναι ευέλικτο καθώς και παραμετροποιήσιμο σε μελλοντικό στάδιο από κάποιον προγραμματιστή, εκτενέστερα βασίστηκε στις εξής τεχνολογίες:

1.Java
2. Spring Boot
3.Hibernate
4.HTML 5 + CSS 3
5.JUnit
6.Thymeleaf
7.MySQL
8. MVC + RestAPI
9.Spring Security

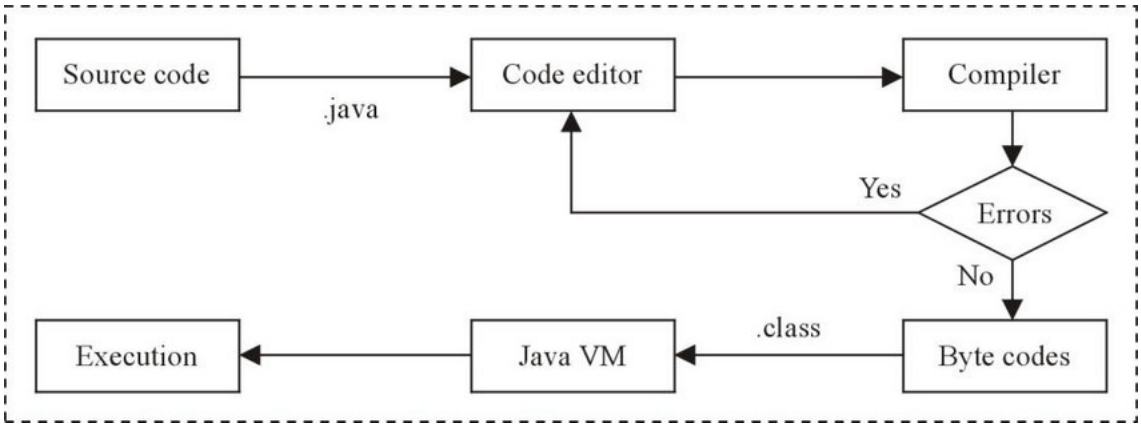
Για την αποθήκευση των δεδομένων μας χρησιμοποιήθηκε σχεσιακή βάση δεδομένων , προτιμήθηκε έναντι άλλων βάσεων δεδομένων διότι θέλαμε να παράγεται το ερώτημα το οποίο δημιουργεί ο χρήστης του συστήματος και να μην υπάρχουν αστάθειες ως προς την εξέλιξη τους. Οι σχεσιακές βάσεις δεδομένων μας αποδεικνύουν καθημερινά την λειτουργικότητας τους σε διάφορους τομείς παρόλο την αισθητή καθυστέρηση που μπορεί να δημιουργηθεί κάποιες φορές. Ένα χαρακτηριστικό παράδειγμα που μπορούμε να καταλάβουμε την λειτουργικότητα και την ακρίβεια που δίνουν είναι στα τραπεζικά συστήματα. Αρκετές φορές μπορεί να έχει παρατηρηθεί καθυστέρηση σε κάποια ATM παρόλο αυτά ο χρήστης θα καταφέρει πάντα να αποδεσμεύσει το μηχάνημα έχοντας λάβει το ποσό το οποίο θέλει να δεσμεύσει σε ρευστοποίηση . Οι σχεσιακές βάσεις δεδομένων έλαβαν τον τίτλο του σχεσιακού διότι μπορούν να συνδυάσουν δεδομένα καθώς και οντότητες.

Συγκρίνοντας πάντα τις απαιτήσεις καθώς και τις συνήθειες που είχαν μέχρι στιγμή οι χρήστες προσπαθήσαμε να μοντελοποιήσουμε τα δεδομένα όπως ένα πραγματικό πρωτόκολλο και για αυτό τον λόγο χρειάστηκε η συσχέτιση μεταξύ των δεδομένων. Η συσχέτιση των δεδομένων κατά παρελθοντικούς χρόνους συνηθιζόταν να γίνεται σαν σχήμα . Ένα σχήμα το οποίο θα δημιουργούσε γραμμές και στήλες συνδεδεμένες πάντα μεταξύ τους ώστε να περιμένουν τον χρήστη να προσθέσει δεδομένα.

Όπως πολύ γρήγορα μπορούμε να καταλάβουμε αυτό μπορεί να δημιουργήσει αρκετά προβλήματα σε μελλοντικές αλλαγές διότι θα έπρεπε να τροποποιήσουμε το σχήμα από την αρχή. Ορίζοντας όμως από την αρχή του ηλεκτρονικού πρωτοκόλλου την χρήση καθώς και την εξέλιξη που μπορεί να πάρει συνειδητοποιήθηκε η ανάγκη για έναν μηχανισμό που θα μπορούσε να αλλάξει το σχήμα γρήγορα.

Σε αυτό το σημείο το πρόβλημα λύθηκε μέσω του Hibernate, όλες οι δομές δεδομένων καθώς και οι λειτουργίες ενσωματώθηκαν μέσω της Java και πιο συγκεκριμένα της εντέκατης έκδοσης που μας δίνει.

3.2 Java



Η Java είναι μια δημοφιλής αντικειμενοστραφής γλώσσα προγραμματισμού και πλατφόρμα λογισμικού που τρέχει σε δισεκατομμύρια συσκευές, συμπεριλαμβανομένων φορητών υπολογιστών, φορητών συσκευών, κονσολών παιχνιδιών, ιατρικών συσκευών και άλλων. Οι κανόνες και η σύνταξη Java βασίζονται στις γλώσσες C και C++. Ένα από τα κύρια πλεονεκτήματα της ανάπτυξης λογισμικού σε Java είναι η φορητότητα. Αφού γράψετε κώδικα για ένα πρόγραμμα Java στον φορητό υπολογιστή σας, είναι πολύ εύκολο να μεταφέρετε τον κωδικό σας σε φορητές συσκευές.

Όταν ο James Gosling της Sun-Microsystems εφηύρε τη γλώσσα το 1991, ο κύριος στόχος της ήταν «να γράψει μια φορά, να τρέξει οπουδήποτε». Είναι επίσης σημαντικό να κατανοήσουμε ότι η Java είναι πολύ διαφορετική από την JavaScript. Η JavaScript δεν χρειάζεται να μεταγλωττιστεί και ο κώδικας Java δεν χρειάζεται να μεταγλωττιστεί. Επίσης, το JavaScript λειτουργεί μόνο σε προγράμματα περιήγησης ιστού, η Java μπορεί να εκτελεστεί οπουδήποτε. Νέα και βελτιωμένα εργαλεία ανάπτυξης λογισμικού εισέρχονται στην αγορά με ανησυχητικό ρυθμό, αντικαθιστώντας παλαιότερα προϊόντα που κάποτε θεωρούνταν απαραίτητα. Υπό το πρίσμα αυτής της συνεχιζόμενης μετάβασης, η διάρκεια ζωής της Java είναι εντυπωσιακή. Αν και η Java υπάρχει εδώ και πάνω από 20 χρόνια, εξακολουθεί να είναι η πιο ευρέως χρησιμοποιούμενη γλώσσα για την ανάπτυξη λογισμικού εφαρμογών. Οι προγραμματιστές εξακολουθούν να προτιμούν την Java από γλώσσες όπως Python, Ruby, PHP, Swift, C κ.λ.π. Ως αποτέλεσμα, η Java παραμένει μια σημαντική ανταγωνιστική απαίτηση στην αγορά εργασίας.

Πριν εξετάσουμε τους λόγους για τη συνεχιζόμενη δημοτικότητα της Java, ας ρίξουμε μια πιο προσεκτική ματιά στο τι είναι η Java και τη σημασία της για την ανάπτυξη εταιρικών εφαρμογών. Η Java είναι μια τεχνολογία που αποτελείται από μια γλώσσα προγραμματισμού και μια πλατφόρμα λογισμικού. Για να δημιουργήσετε εφαρμογές χρησιμοποιώντας Java, πρέπει να κάνετε λήψη του Java Development Kit (JDK), το οποίο είναι διαθέσιμο για Windows, macOS και Linux. Όταν γράφετε ένα πρόγραμμα στη γλώσσα προγραμματισμού Java, ο μεταγλωττιστής μετατρέπει το πρόγραμμα σε bytecode Java (το σύνολο εντολών για την Java Virtual Machine (JVM) που αποτελεί μέρος του Java Runtime Environment (JRE)). Το Java Bytecode λειτουργεί αμετάβλητο σε συστήματα με δυνατότητα JVM, ώστε να μπορείτε να εκτελείτε κώδικα Java οπουδήποτε. Η πλατφόρμα λογισμικού Java αποτελείται από ένα JVM, ένα Java API και ένα πλήρες περιβάλλον ανάπτυξης. Το JVM αναλύει και εκτελεί (ερμηνεύει) τον bytecode Java. Το Java API αποτελείται από ένα εκτεταμένο σύνολο βιβλιοθηκών, συμπεριλαμβανομένων βασικών αντικειμένων, δικτύωσης και λειτουργιών ασφαλείας. Δημιουργία XML (Extensible Markup Language). και υπηρεσίες

web.

Η γλώσσα Java και η πλατφόρμα προγραμματισμού Java δημιουργούν μαζί μια ισχυρή και δοκιμασμένη τεχνολογία για την ανάπτυξη εταιρικού λογισμικού. Ενώ τα τεχνικά επιχειρήματα για την Java είναι επιτακτικά, οι επιχειρηματικοί λόγοι για την επιλογή της Java είναι εξίσου ισχυροί. Δηλαδή, μια μεγάλη δεξαμενή ταλέντων, μια σύντομη καμπύλη μάθησης και ένα εκτεταμένο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE). Η ζήτηση για εξειδικευμένους προγραμματιστές συνεχίζει να αυξάνεται καθώς περισσότερες εταιρείες χρησιμοποιούν συνδεδεμένες συσκευές, αλγόριθμους μηχανικής εκμάθησης και λύσεις cloud. Πολλοί αναλυτές προβλέπουν ότι στο εγγύς μέλλον θα υπάρξει έλλειψη ανώτερων προγραμματιστών, γεγονός που καθιστά δύσκολη τη στελέχωση νέων προγραμμάτων λογισμικού. Η ζήτηση για προγραμματιστές κινητής τηλεφωνίας θα μπορούσε σύντομα να ξεπεράσει τη διαθέσιμη προσφορά. Η μεγάλη ομάδα προγραμματιστών Java είναι ένας καλός λόγος για να βασιστούν σημαντικές πρωτοβουλίες λογισμικού στην Java. Όταν οι διαχειριστές ανθρώπινου δυναμικού δημοσιεύουν θέσεις εργασίας για προγραμματιστές Java, μπορούν να περιμένουν να λάβουν πολλά πιστοποιημένα βιογραφικά και να καλύψουν αυτές τις θέσεις σχετικά γρήγορα. Οι διευθυντές μπορούν επίσης να αξιοποιήσουν τους συμβατικούς πόρους για να προσλάβουν προσωπικό για συγκεκριμένες εργασίες χωρίς να αυξήσουν τον αριθμό των εργαζομένων.

Εκτός από τους ανώτερους προγραμματιστές, οι μεγάλες πρωτοβουλίες λογισμικού απαιτούν μεγάλο αριθμό κατώτερων συνεργατών. Η Java εξακολουθεί να είναι μια δημοφιλής εισαγωγική γλώσσα προγραμματισμού στα κολεγιακά μαθήματα επιστήμης υπολογιστών, αλλά πολλοί απόφοιτοι δεν έχουν τις δεξιότητες για να είναι παραγωγικοί την πρώτη μέρα. Η Java είναι πιο εύκολη στην εκμάθηση και την εκμάθηση από πολλές άλλες γλώσσες προγραμματισμού που μπορούν να συντομεύσουν τους χρόνους εκμάθησης και να επιταχύνουν τα κέρδη παραγωγικότητας. Μια τεράστια διαδικτυακή κοινότητα Java από φόρουμ προγραμματιστών, σεμινάρια και ομάδες χρηστών βοηθά τους αρχάριους να μαθαίνουν γρήγορα και παρέχει στους έμπειρους προγραμματιστές αποτελεσματικά, αποδεδειγμένα εργαλεία αντιμετώπισης προβλημάτων. Στον τομέα των εργαλείων προγραμματισμού, η Java προσφέρει πολλά IDE. Οι έμπειροι προγραμματιστές Java μπορούν γρήγορα να εξοικειωθούν με το νέο περιβάλλον, δίνοντας στους διαχειριστές ανάπτυξης την ελευθερία να επιλέξουν το IDE που ταιριάζει καλύτερα στον τύπο του έργου, τον προϋπολογισμό, τη μέθοδο ανάπτυξης και το επίπεδο δεξιοτήτων προγραμματιστή τους.

3.3 Spring Framework and Microservices

Το Spring Framework παρέχει μια δυνατότητα έγχυσης εξάρτησης που επιτρέπει στα αντικείμενα να ορίζουν τις δικές τους εξαρτήσεις που εισάγει αργότερα το κοντέινερ Spring. Αυτό επιτρέπει στους προγραμματιστές να δημιουργούν αρθρωτές εφαρμογές που αποτελούνται από χαλαρά συζευγμένα στοιχεία που είναι ιδανικά για μικροϋπηρεσίες και εφαρμογές κατανεμημένου δικτύου. Το Spring Framework έχει επίσης ενσωματωμένη υποστήριξη για κοινές εργασίες που πρέπει να εκτελούν οι εφαρμογές, όπως: Σχεσιακές βάσεις δεδομένων, μετατροπή τύπου, επικύρωση, διαχείριση εξαιρέσεων, διαχείριση πόρων και συμβάντων, διεθνοποίηση κ.λ.π. Ενσωματώνεται με διάφορες τεχνολογίες Java EE όπως Remote Method Invocation (RMI), Advanced Message Quueing Protocol (AMQP) και Java Web Services.

Συνοπτικά, το Spring Framework παρέχει όλα τα εργαλεία και τις δυνατότητες που χρειάζονται οι προγραμματιστές για να δημιουργήσουν χαλαρά συνδεδεμένες εφαρμογές Java EE μεταξύ πλατφορμών που μπορούν να εκτελούν σε οποιοδήποτε περιβάλλον. Τόσο ισχυρό και ολοκληρωμένο όσο το Spring Framework, χρειάζεται πολύς χρόνος και γνώση για τη διαμόρφωση, τη ρύθμιση και την ανάπτυξη εφαρμογών Spring. Το Spring Boot διευκολύνει αυτήν την προσπάθεια με τρία σημαντικά χαρακτηριστικά. Αυτόματη διαμόρφωση-Αυτόματη ρύθμιση παραμέτρων σημαίνει ότι η εφαρμογή σας έχει προετοιμαστεί με προρυθμισμένες εξαρτήσεις που δεν χρειάζεται να ρυθμιστούν με μη αυτόματο τρόπο. Το Java Spring Boot έχει μια ενσωματωμένη δυνατότητα

αυτόματης διαμόρφωσης που διαμορφώνει αυτόματα τόσο το υποκείμενο πλαίσιο Spring όσο και τα πακέτα τρίτων με βάση τις ρυθμίσεις σας (και τις βέλτιστες πρακτικές για την αποφυγή σφαλμάτων).

Μπορείτε να παρακάμψετε αυτές τις προεπιλογές μετά την ολοκλήρωση της προετοιμασίας, αλλά μπορείτε να χρησιμοποιήσετε τη δυνατότητα αυτόματης ρύθμισης παραμέτρων του Java Spring Boot για να ξεκινήσετε γρήγορα την ανάπτυξη εφαρμογών που βασίζονται στο Spring και να μειώσετε την πιθανότητα ανθρώπινου λάθους. Το Spring Boot ακολουθεί μια ιδιόμορφη προσέγγιση για την προσθήκη και τη διαμόρφωση εξαρτήσεων εκκίνησης με βάση τις ανάγκες του έργου. Αντί να λαμβάνει όλες αυτές τις αποφάσεις και να τις ορίζει όλες με μη αυτόματο τρόπο, το Spring Boot επιλέγει ποια πακέτα θα εγκαταστήσει και ποια πακέτα θα χρησιμοποιήσει από προεπιλογή κατά την κρίση του. Για να καθορίσετε τις απαιτήσεις του έργου κατά τη διαδικασία αρχικοποίησης, επιλέξτε από πολλές εξαρτήσεις εκκίνησης (τα λεγόμενα ελατήρια εκκίνησης*initializer*) που καλύπτουν περιπτώσεις κοινής χρήσης.

Για να εκτελέσετε το Spring Boot Initializer, συμπληρώστε μια απλή φόρμα web χωρίς καμία κωδικοποίηση. Για παράδειγμα, μπορείτε να χρησιμοποιήσετε το Spring Web Starter Dependencies για να δημιουργήσετε μια εφαρμογή web που βασίζεται στο Spring με ελάχιστες ρυθμίσεις παραμέτρων, προσθέτοντας όλες τις απαιτούμενες εξαρτήσεις στο έργο σας, όπως ο διακομιστής web Apache Tomcat. Η ”Spring Security” είναι μια άλλη κοινή εξάρτηση εκκίνησης που προσθέτει αυτόματα δυνατότητες ελέγχου ταυτότητας και ελέγχου πρόσβασης στην εφαρμογή σας. Το SpringBoot περιλαμβάνει πάνω από 50 SpringStarters και είναι διαθέσιμα περισσότερα starters τρίτων. Η αυτόνομη εφαρμογή Spring Boot βοηθά τους προγραμματιστές να δημιουργήσουν εφαρμογές που είναι εύκολες στην εκτέλεση. Συγκεκριμένα, με την ενσωμάτωση ενός διακομιστή web όπως ο Tomcat ή ο Netty στην εφαρμογή σας κατά τη διαδικασία προετοιμασίας, μπορείτε να δημιουργήσετε μια αυτόνομη εφαρμογή που να εκτελείται ανεξάρτητα και ανεξάρτητα από έναν εξωτερικό διακομιστή ιστού.

Επομένως, μπορείτε να εκκινήσετε την εφαρμογή σας σε οποιαδήποτε πλατφόρμα κάνοντας απλά κλικ στην εντολή Εκτέλεση.(Η απενεργοποίηση αυτής της λειτουργίας σάς επιτρέπει να δημιουργείτε εφαρμογές που δεν διαθέτουν ενσωματωμένο διακομιστή web.) Και πάλι, τα μεγαλύτερα πλεονεκτήματα της χρήσης του Spring Boot καθώς και του Spring Framework είναι η ευκολία χρήσης και η ταχύτερη ανάπτυξη.

Θεωρητικά, αυτό γίνεται σε βάρος της μεγαλύτερης ευελιξίας που αποκτάται από την απευθείας εργασία με το Spring Framework. Ωστόσο, στην πράξη, αξίζει να χρησιμοποιήσετε το Spring Booth, εκτός αν χρειαστεί να εφαρμόσετε ή να εφαρμόσετε τη δική σας διαμόρφωση. Μπορείτε να συνεχίσετε να χρησιμοποιείτε το πολύ δημοφιλές σύστημα σχολιασμών του Spring Framework. Αυτό διευκολύνει την προσθήκη εξαρτήσεων (που δεν καλύπτονται από Spring Starters) στην εφαρμογή σας. Επιπλέον, θα συνεχίσετε να έχετε πρόσβαση σε όλες τις δυνατότητες του Spring Framework, όπως ο απλός χειρισμός συμβάντων, η επικύρωση, η δέσμευση δεδομένων, η μετατροπή τύπου, η ενσωματωμένη ασφάλεια και οι δυνατότητες δοκιμής. Το Spring Boot μπορεί να βελτιώσει σημαντικά την ανάπτυξη εάν το εύρος του έργου σας καλύπτεται από ένα μόνο Spring Starter.

3.4 Hibernate

Το Hibernate είναι μια λύση αντικειμενικής σχέσης αντιστοίχισης ανοιχτού κώδικα σε Java. Είναι ελαφρύ και αφαιρεί όλες τις ελλείψεις που αντιμετωπίζετε όταν εργάζεστε με το JDBC. Το Hibernate είναι ένα πλαίσιο Java που έχει ένα επίπεδο αφαίρεσης και χειρίζεται την υλοποίηση εσωτερικά. Η υλοποίηση περιλαμβάνει εργασίες όπως δημιουργία ερωτημάτων για λειτουργίες CRUD ή δημιουργία συνδέσεων βάσης δεδομένων. Ένα πλαίσιο είναι ουσιαστικά λογισμικό που παρέχει αφαιρέσεις για πολλές τεχνολογίες όπως JDBC, servlets κ.λπ. Το Hibernate αναπτύσσει λογική εμμονής για αποθήκευση και επεξεργασία δεδομένων για μεγαλύτερη χρήση. Είναι ένα

ελαφρύ εργαλείο ORM, το πιο σημαντικό, ανοιχτού κώδικα, δίνοντάς του ένα πλεονέκτημα σε σχέση με άλλα πλαίσια, πώς να εμφανίσετε αντικείμενα που είναι αποθηκευμένα σε μια βάση δεδομένων.

Τα εργαλεία ORM διευκολύνουν τη δημιουργία, τον χειρισμό και την πρόσβαση σε δεδομένα. Χρησιμοποιεί εσωτερικά Java API για να αλληλεπιδρά με τη βάση δεδομένων. Το Hibernate εξαλείφει τις αδυναμίες άλλων τεχνολογιών όπως το JDBC. Το Hibernate ξεπερνά την εξάρτηση της βάσης δεδομένων που αντιμετωπίζει το JDBC. Η αλλαγή της βάσης δεδομένων είναι ακριβή όταν εργάζεστε με το JDBC και το Hibernate, είναι πολύ αποτελεσματική για να γίνει αυτό. Η φορητότητα κώδικα δεν είναι δυνατή με το JDBC, όπου Το Hibernate είναι εύκολο να χειριστεί.Το Hibernate ενισχύει τις σχέσεις σε επίπεδο αντικειμένου,ξεπερνά μέρος του χειρισμού εξαιρέσεων που απαιτείται όταν εργάζεστε με το JDBC. Το Hibernate υπερνικά τις σχέσεις σε επίπεδο αντικειμένου. Το Hibernate ξεπερνά όλες τις αδυναμίες του JDBC για να παρέχει μια βέλτιστη και αποτελεσματική λύση για κάθε εργασία.

Ως πλαίσιο ανοιχτού κώδικα, είναι δωρεάν για όλους. Ο πηγαίος κώδικας για την Hibernate βρίσκεται στο διαδίκτυο και μπορεί επίσης να αλλάξει μέσω αυτού. Τα οφέλη της ευκολίας χρήσης του πλαισίου φαίνονται σε ένα πολύ μικρότερο πακέτο εγκατάστασης. Η μη χρήση κοντέινερ για εκτέλεση αυξάνει την απόδοση. Το Hibernate μπορεί να λειτουργήσει με πολλές τεχνολογίες ταυτόχρονα, αλλά αυτό δεν σημαίνει ότι το Hibernate δεν μπορεί να λειτουργήσει μόνο του.

Το Hibernate έχει την ειδική ιδιότητα ότι δεν χρειάζεται να εφαρμοστεί το API Hibernate ή να επεκταθεί οι κλάσεις Hibernate API, επειδή οι κλάσεις ανάπτυξης εφαρμογών Hibernate είναι χαλαρά συνδεδεμένες. Οι δυνατότητες που υποστηρίζονται από το Hibernate χρησιμοποιούν τη γλώσσα ερωτημάτων Hibernate με βάσεις δεδομένων. Υποστηρίζει αυτόματη λειτουργία DDL. Το Hibernate υποστηρίζει την αυτόματη δημιουργία πρωτεύοντος κλειδιού. Αρκετές φορές παρομοιάζεται με εργαλεία όπως το Reverse Engineering διότι μπορεί με ελάχιστες γραμμές κώδικα να πραγματοποιήση ερωτήματα στην βάση δεδομένων μας.

3.5 Web Resolver

Χρησιμοποιώντας τις υπηρεσίες διαδικτύου καθημερινά μπορούμε να έχουμε συναντήσει διάφορα αρχεία με κατάληξη html καθώς και css . Αρκετοί παρομοιάζουν την HTML με γλώσσα προγραμματισμού αλλά σίγουρα μπορούμε να το διαψεύσουμε διότι όπως αναφέρει και η συντομογραφία της ειναι μια Hypertext Markup Language. Μπορούμε μέσω αυτής να αποτυπώσουμε πληροφορία είτε στατική είτε δυναμική. Συχνά ακούμε τις συντομογραφίες αυτών των δύο τεχνολογιών μαζί διότι η HTML θα μας δώσει την εμφάνιση αυτών των πληροφοριών σε μορφή κειμένου αλλά δεν μπορεί να μας δώσει την απαιτούμενη μορφοποίηση ώστε να είναι πιο φιλική στον χρήστη . Η μορφοποίηση γίνεται μέσω του CSS, μπορούμε μέσω του Cascading Style Sheets να δώσουμε μια ομορφότερη απεικόνιση στο κείμενο μας απο την άποψη χρωμάτων, στοίχισης. Το κυριότερο στοιχείο της είναι η απεικόνιση της πληροφορίας σε μεγαλύτερης οθόνες καθώς και μικρότερες. Μας δίνει την λειτουργικότητα την οποία ζητούμε βάση το μέγεθος της οθόνη.

Όπως αναφέραμε η html μπορεί να έχει περιεχόμενο είτε στατικό , είτε δυναμικό. Την δυναμικότητα την χρειαζόμαστε ώστε βάση τις επιλογές του χρήστη να του εμφανίζεται διαφορετικό περιεχόμενο. Για την φόρτωσή της πληροφορίας χρειαζόμαστε ένα ενδιάμεσο κανάλι επικοινωνίας ώστε να έχουμε πρόσβαση με την βάση δεδομένων μας . Η πρόσβαση της βάσης δεδομένων συνήθως δίνεται μέσω κάποιας τρίτης γλώσσας πραγματισμού λχ PHP . Στην παρούσα υλοποίηση εκμεταλλευτήκαμε τις μέγιστες δυνατότητες του Spring Framework και πιο συγκεκριμένα του Thymeleaf .

Το Thymeleaf είναι μια σύγχρονη μηχανή προτύπων Java από την πλευρά του διακομιστή για περιβάλλοντα web και για αυτόνομο περιβάλλον. Ο κύριος στόχος του Thymeleaf είναι να φέρει κομψά φυσικά πρότυπα στη ροή εργασιών ανάπτυξής σας — HTML που μπορεί να εμφανιστεί σωστά σε προγράμματα περιήγησης και επίσης να λειτουργήσει ως στατικά πρωτότυπα, επιτρέπο-

ντας ισχυρότερη συνεργασία σε ομάδες ανάπτυξης. Με ενότητες για το Spring Framework, πλήθος ενσωματώσεων με τα αγαπημένα σας εργαλεία και τη δυνατότητα να συνδέσετε τη δική σας λειτουργικότητα, το Thymeleaf είναι ιδανικό για τη σύγχρονη ανάπτυξη ιστού HTML5 JVM — αν και υπάρχουν πολλά περισσότερα που μπορεί να κάνει. Το μεγαλύτερο πλεονέκτημα της είναι ότι μπορεί να εναρμονιστεί με το design pattern MVC.

3.6 MVC

Τα στοιχεία αρχιτεκτονικής του προτύπου MVC έχουν σχεδιαστεί για να χειρίζονται διαφορετικές πτυχές μιας εφαρμογής υπό ανάπτυξη. Το μοτίβο σχεδιασμού MVC χρησιμεύει για να διαχωρίσει το επίπεδο παρουσίασης από την επιχειρηματική λογική.

Το MVC είναι δημοφιλές στην ανάπτυξη εφαρμογών και ιστού και είναι ένα από τα πιο ευρέως χρησιμοποιούμενα μοτίβα σχεδιασμού λογισμικού για ανάπτυξη εφαρμογών. Το μοτίβο σχεδίασης του ελεγκτή προβολής μοντέλου χωρίζει τις ανησυχίες σε έναν από τους 3 κάδους: Μοντέλο-MODEL, Θέα-VIEW, Ελεγκτής-CONTROLLER

Το αρχιτεκτονικό μοτίβο του Ελεγκτή Μοντέλου Προβολής διαχωρίζει τις ανησυχίες σε έναν από τους 3 κάδους:

1.Μοντέλο: αποθηκεύει και διαχειρίζεται δεδομένα. Συχνά μια βάση δεδομένων, στο σύντομο παράδειγμά μας θα χρησιμοποιήσουμε τοπικό χώρο αποθήκευσης ιστού σε ένα πρόγραμμα περιήγησης για να επεξηγήσουμε την έννοια.

2.Προβολή: Γραφική διεπαφή χρήστη Η προβολή είναι μια οπτική αναπαράσταση των δεδομένων - όπως γράφημα, διάγραμμα, πίνακας, φόρμα. Η προβολή περιέχει όλες τις λειτουργίες που αλληλεπιδρούν άμεσα με τον χρήστη - όπως το κλικ σε ένα κουμπί ή ένα συμβάν εισαγωγής.

3.Ελεγκτής: Εγκέφαλοι της εφαρμογής. Ο ελεγκτής συνδέει το μοντέλο και την προβολή. Ο ελεγκτής μετατρέπει τις εισόδους από την προβολή σε απαιτήσεις για ανάκτηση/ενημέρωση δεδομένων στο μοντέλο. Ο ελεγκτής λαμβάνει δεδομένα από την προβολή, χρησιμοποιεί τη λογική για να μεταφράσει την είσοδο σε μια ζήτηση για το μοντέλο, το μοντέλο αρπάζει τα δεδομένα, ο ελεγκτής μεταφέρει δεδομένα από το μοντέλο πίσω στην προβολή για να τα δει ο χρήστης σε μια ωραία οθόνη.

Χρησιμοποιείται παραδοσιακά για γραφικές διεπαφές χρήστη (GUI), οι αρμοδιότητες MVC κατανέμονται μεταξύ του πελάτη και του διακομιστή, συμβατό με την αρχιτεκτονική εφαρμογών web το MVC είναι χρήσιμο μοτίβο σχεδιασμού κατά τον προγραμματισμό ανάπτυξης. Αυτός ο κώδικας χωρίζεται με βάση τη συνάρτηση είτε στο μοντέλο, την προβολή ή τον κάδο ελεγκτή Λειτουργεί καλά με το Ruby on Rails, χαλαρά συζευγμένο,αφαιρεί τις περιττές εξαρτήσεις , επαναχρησιμοποιήσιμο χωρίς τροποποίηση

Το MVC κάνει τις κατηγορίες μοντέλων επαναχρησιμοποιήσιμες χωρίς τροποποίηση:

1.Επαναχρησιμοποίηση κώδικα

2.Επεκτάσιμος κώδικας

3.Υψηλή συνοχή

4.Πιο εύκολο στη συντήρηση ή τροποποίηση

5.Υποστηρίζει πολλαπλές προβολές

6.Κάθε εξάρτημα μπορεί να ελεγχθεί ανεξάρτητα (μοντέλο, προβολή, ελεγκτής)

Το MVC είναι δημοφιλές στις εφαρμογές web, διότι οι ευθύνες κατανέμονται μεταξύ του πελάτη και του διακομιστή.

Ο σχεδιασμός MVC επιτρέπει τον διαχωρισμό των ανησυχιών - διαιρώντας τη λογική μεταξύ των 3 κάδου, έτσι ώστε κάθε κάδος να μπορεί να ενεργεί ανεξάρτητα. Το μοντέλο, η προβολή και ο ελεγκτής δεν εξαρτώνται το ένα από το άλλο. Γενικά, το λογισμικό εργάζεται από ομάδες - μια ομάδα μπορεί να έχει σχεδιαστή, μηχανικό και αρχιτέκτονα βάσης δεδομένων. Ο διαχωρισμός των ανησυχιών σημαίνει ότι κάθε μέλος της ομάδας μπορεί να εργαστεί στο κομμάτι του ταυτόχρονα, επειδή η λογική έχει χωριστεί σε κουβάδες. Ο διαχωρισμός των ανησυχιών είναι επίσης εξαιρετικός

για τη συντήρηση - οι προγραμματιστές μπορούν να διορθώσουν ένα σφάλμα σε ένα κομμάτι κώδικα, χωρίς να χρειάζεται να ελέγξουν τα άλλα κομμάτια κώδικα.

Χαλαρά συζευγμένο σημαίνει ότι το μοντέλο, η προβολή και ο ελεγκτής, ενεργούν ανεξάρτητα το ένα από το άλλο. Οι προγραμματιστές μπορούν να τροποποιήσουν ένα από τα κομμάτια και τα άλλα 2 κομμάτια θα πρέπει να συνεχίσουν να λειτουργούν και να μην απαιτούν τροποποιήσεις. Όταν σχεδιάζετε λογισμικό MVC – η λογική σε κάθε έναν από τους τρεις κάδους είναι ανεξάρτητη. Τα πάντα στο View λειτουργούν ανεξάρτητα από το μοντέλο – και αντίστροφα, η προβολή δεν θα έχει καμία λογική που να εξαρτάται από το μοντέλο. Η δημιουργία ανεξάρτητων μοντέλων και προβολών καθιστά την οργάνωση κώδικα απλή και κατανοητή και διευκολύνει τη συντήρηση. Οι προγραμματιστές μπορούν να διορθώσουν ένα σφάλμα στην προβολή χωρίς να αλλάξουν τον κωδικό μοντέλου.

3.7 REST API

Το REST API (γνωστό και ως RESTful API) είναι μια διεπαφή προγραμματισμού εφαρμογών (API ή web API) που συμμορφώνεται με τους περιορισμούς του αρχιτεκτονικού στυλ REST και επιτρέπει την αλληλεπίδραση με τις υπηρεσίες ιστού RESTful. Το REST σημαίνει μεταφορά κατάστασης αναπαράστασης και δημιουργήθηκε από τον επιστήμονα υπολογιστών Roy Fielding. Ένα API είναι ένα σύνολο ορισμών και πρωτοκόλλων για τη δημιουργία και την ενοποίηση λογισμικού εφαρμογών. Μερικές φορές αναφέρεται ως σύμβαση μεταξύ ενός παρόχου πληροφοριών και ενός χρήστη πληροφοριών—καθορίζοντας το περιεχόμενο που απαιτείται από τον καταναλωτή (η κλήση) και το περιεχόμενο που απαιτείται από τον παραγωγό (η απάντηση). Για παράδειγμα, ο σχεδιασμός API για μια υπηρεσία καιρού θα μπορούσε να προσδιορίσει ότι ο χρήστης παρέχει έναν ταχυδρομικό κώδικα και ότι ο παραγωγός απαντά με μια απάντηση 2 μερών, το πρώτο είναι η υψηλή θερμοκρασία και το δεύτερο η χαμηλή.

Με άλλα λόγια, εάν θέλετε να αλληλεπιδράσετε με έναν υπολογιστή ή ένα σύστημα για να ανακτήσετε πληροφορίες ή να εκτελέσετε μια λειτουργία, ένα API σας βοηθά να επικοινωνήσετε αυτό που θέλετε σε αυτό το σύστημα, ώστε να μπορεί να κατανοήσει και να εκπληρώσει το αίτημα.

Μπορείτε να σκεφτείτε ένα API ως μεσολαβητή μεταξύ των χρηστών ή των πελατών και των πόρων ή των υπηρεσιών Ιστού που θέλουν να αποκτήσουν. Είναι επίσης ένας τρόπος για έναν οργανισμό να μοιράζεται πόρους και πληροφορίες διατηρώντας παράλληλα την ασφάλεια, τον έλεγχο και τον έλεγχο ταυτότητας—καθορίζοντας ποιος έχει πρόσβαση σε τι.

Το REST είναι ένα σύνολο αρχιτεκτονικών περιορισμών, όχι ένα πρωτόκολλο ή ένα πρότυπο. Οι προγραμματιστές API μπορούν να εφαρμόσουν το REST με διάφορους τρόπους.

Όταν ένα αίτημα πελάτη γίνεται μέσω ενός RESTful API, μεταφέρει μια αναπαράσταση της κατάστασης του πόρου στον αιτούντα ή στο τελικό σημείο. Αυτές οι πληροφορίες, ή αναπαράσταση, παραδίδονται σε μία από τις διάφορες μορφές μέσω HTTP: JSON (Javascript Object Notation), HTML, XLT, Python, PHP ή απλό κείμενο. Το JSON είναι η πιο δημοφιλής μορφή αρχείου που χρησιμοποιείται γενικά, επειδή, παρά το όνομά του, είναι αγνωστική ως προς τη γλώσσα, καθώς και ευανάγνωστη τόσο από ανθρώπους όσο και από μηχανήματα.

Οι κεφαλίδες και οι παράμετροι είναι επίσης σημαντικές στις μεθόδους HTTP ενός αιτήματος RESTful API HTTP, καθώς περιέχουν σημαντικές πληροφορίες αναγνωριστικού ως προς τα μεταδεδομένα του αιτήματος, την εξουσιοδότηση, το ενιαίο αναγνωριστικό πόρων (URI), την προσωρινή αποθήκευση, τα cookie και άλλα. Υπάρχουν κεφαλίδες αιτημάτων και κεφαλίδες απόκρισης, το καθένα με τις δικές του πληροφορίες σύνδεσης HTTP και κωδικούς κατάστασης.

3.8 Spring Security

Το Spring Security είναι ένα πλαίσιο που επιτρέπει σε έναν προγραμματιστή να επιβάλλει περιορισμούς ασφαλείας σε εφαρμογές Ιστού που βασίζονται σε Spring Framework μέσω στοιχείων JEE. Εν ολίγοις, είναι μια βιβλιοθήκη που μπορεί να χρησιμοποιηθεί, να επεκταθεί για προσαρμογή σύμφωνα με τις ανάγκες του προγραμματιστή. Επειδή είναι μέλος της ίδιας οικογένειας Spring, ταιριάζει ομαλά με το Spring Web MVC.

Ο κύριος τομέας λειτουργίας του είναι ο χειρισμός του ελέγχου ταυτότητας και της εξουσιοδότησης σε επίπεδο αιτήματος Ιστού καθώς και στο επίπεδο επίκλησης μεθόδου. Ισως. Το μεγαλύτερο πλεονέκτημα αυτού του πλαισίου είναι ότι είναι ισχυρό αλλά εξαιρετικά προσαρμόσιμο στην εφαρμογή του. Αν και ακολουθεί τη σύμβαση του Spring για τη διαμόρφωση, οι προγραμματιστές μπορούν να επιλέξουν μεταξύ προεπιλεγμένων διατάξεων ή να τις προσαρμόσουν ανάλογα με τις ανάγκες τους.

Το Spring Security λειτουργεί σε δύο βασικούς τομείς: τον έλεγχο ταυτότητας και την εξουσιοδότηση. Ο έλεγχος ταυτότητας σημαίνει ότι, ενώ έχει πρόσβαση σε ορισμένους περιορισμένους πόρους, ο χρήστης είναι στην πραγματικότητα το κατάλληλο άτομο για να το κάνει. Υπάρχουν δύο διαδικασίες για να βεβαιωθείτε ότι ο χρήστης είναι αυθεντικός: αναγνώριση και επαλήθευση. Για παράδειγμα, ένας χρήστης ελέγχεται μέσω του ονόματος χρήστη και του κωδικού πρόσβασής του, τα οποία συνήθως αποθηκεύονται σε μια βάση δεδομένων. LDAP (Lightweight Directory Access Protocol, ένα ελαφρύ πρωτόκολλο για πρόσβαση σε υπηρεσίες καταλόγου). ή CAS (Central Authentication Service, ένα πρωτόκολλο ενιαίας σύνδεσης για τον Ιστό). Η Spring Security απαιτεί επίσης μια διεπαφή για την κωδικοποίηση του κωδικού πρόσβασης για να γίνει πιο ασφαλής.

Η εξουσιοδότηση καθορίζει την έκταση του δικαιώματος ενός χρήστη για πρόσβαση σε περιορισμένους πόρους. Διασφαλίζει ότι ένας χρήστης επιτρέπεται να έχει πρόσβαση μόνο σε εκείνα τα μέρη του πόρου που έχει εξουσιοδοτηθεί να χρησιμοποιήσει. Για παράδειγμα, ένας χρήστης ADMIN έχει απεριόριστη πρόσβαση στις ιδιότητες της εφαρμογής και μπορεί να τις αλλάξει ή να τις χειριστεί — για καλό ή για κακό. Ένας κανονικός χρήστης ή ένας χρήστης GUEST, από την άλλη πλευρά, έχει πιο ελεγχόμενη πρόσβαση και δεν απολαμβάνει τα ίδια δικαιώματα με τον χρήστη ADMIN. Αυτό ονομάζεται εξουσιοδότηση ρόλου χρήστη. Σε οποιαδήποτε εφαρμογή Ιστού, αυτό γίνεται μέσω ασφάλειας που βασίζεται σε URL.

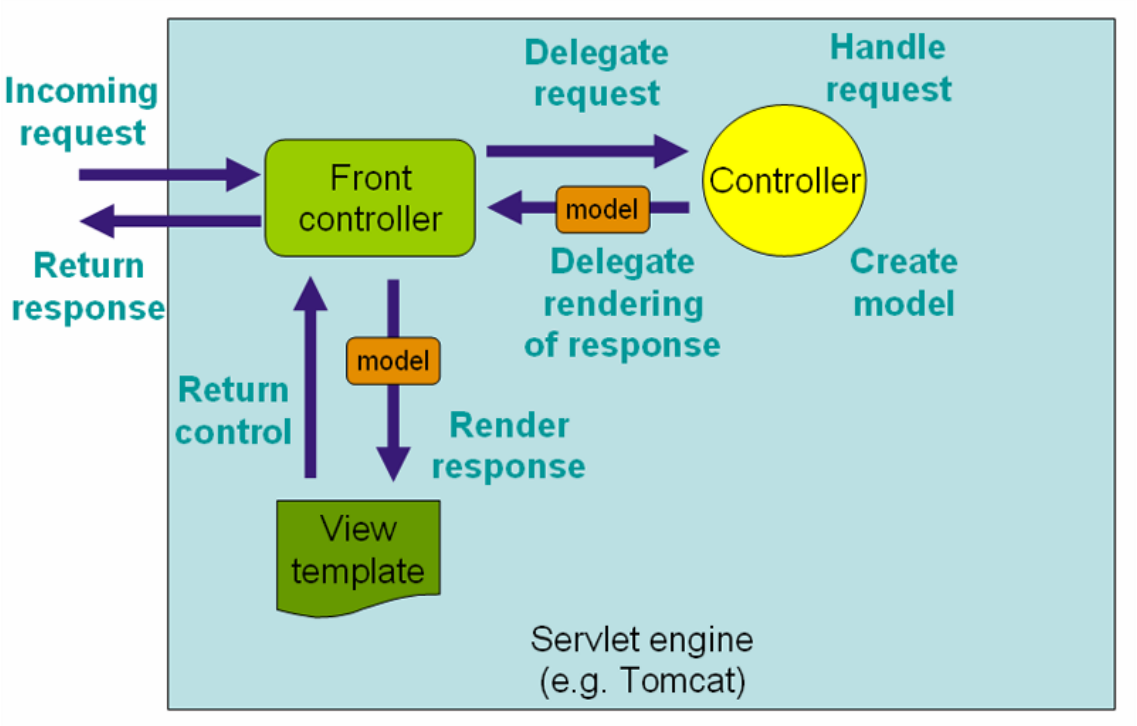
Το Spring παρέχει φίλτρα για τη διασφάλιση του ρόλου της ασφάλισης μιας εφαρμογής. Όμως, η ασφάλεια που βασίζεται σε URL δεν είναι ένας πολύ έξυπνος μηχανισμός και συχνά μπορεί να γίνει κατάχρηση. Οι κακόβουλοι χρήστες μπορούν να χειριστούν τη διεύθυνση URL και να αποκτήσουν πρόσβαση σε μια μέθοδο που προορίζεται στην πραγματικότητα για έναν διαχειριστή χρήστη. Επειδή, σε ένα σύστημα που βασίζεται σε URL, οι κλήσεις πρόσβασης με περιορισμένη μέθοδο αποστέλλονται μέσω υπερσυνδέσμων, είναι πολύ εύκολο να δημιουργήσετε ξανά την ίδια επίκληση μεθόδου από τη διεύθυνση URL και να την στείλετε στον διακομιστή. Ο διακομιστής μπορεί να εκτελέσει αφελώς τις περιορισμένες λειτουργίες χωρίς να επαληθεύσει τον ρόλο του χρήστη που επικαλέστηκε το αίτημα.

Επομένως, για την αντιμετώπιση αυτού του προβλήματος, το Spring προσφέρει ασφάλεια σε επίπεδο μεθόδου. Αυτό σημαίνει απλώς ότι μόνο ορισμένοι εξουσιοδοτημένοι χρήστες μπορούν να επικαλούνται περιορισμένες μεθόδους και απλώς η εκ νέου δημιουργία της διεύθυνσης URL και η αποστολή της στον διακομιστή δεν θα τους εκτελέσει.

3.9 Αρχιτεκτονική

Εάν μπορούσαμε να περιγράψουμε εν συντομία τι μπορεί να μας προσφέρει το framework που δημιούργησε η Red Hat όπου το ονόμασε Hibernate θα ήταν η άμεση μοντελοποίηση της βάσης δεδομένων μας σε σχήμα δίνοντας όμως μέσω ελαχίστων γραμμών κώδικα την λειτουργικότητα για προσθήκη καινούργιων γραμμών καθώς και στηλών μέσω του αντικειμενοστραφούς προγραμ-

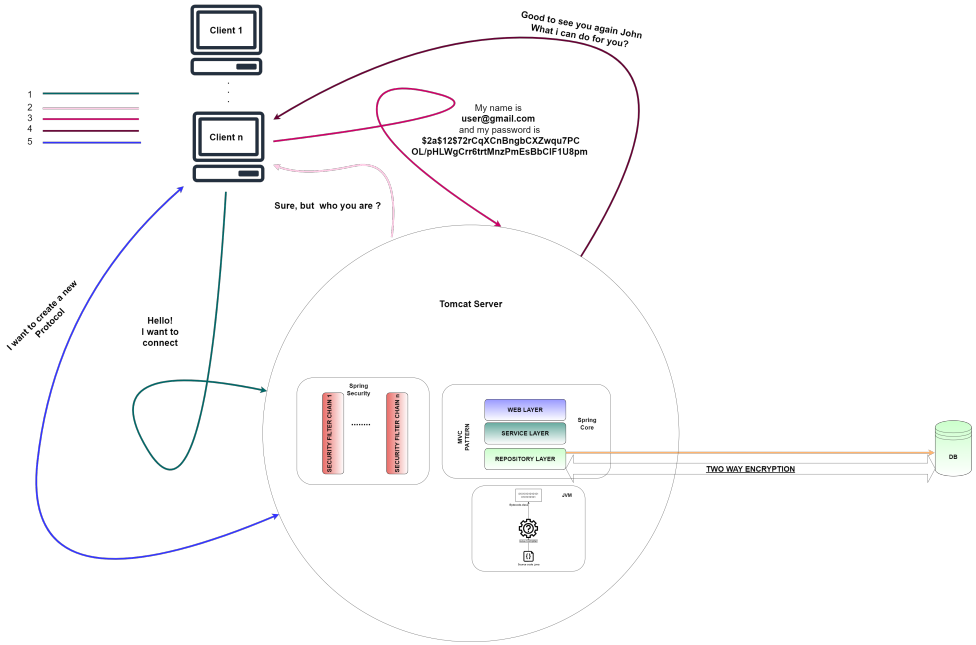
ματισμού. Η διεπαφή χρήστη με το σύστημα έγινε μέσω φυλλομετρητή. Ο χρήστης συνδέεται στον αγαπημένο του browser μέσω οποιασδήποτε συσκευής θελήσει και δημιουργεί το ανάλογο ερώτημα το οποίο θέλει να θέσει , η διεπαφή αυτή δημιουργήθηκε μέσω HTML 5, CSS 3 καθώς και Thymeleaf. Ο χρήστης χρησιμοποιώντας το πρωτόκολλο HTTP κάνει μια κλήση στον Web Server που ορίζει η Spring Boot και μέσω της Java καθώς και του Spring Security λαμβάνει την απάντηση σύμφωνα με την ερώτηση που έχει κάνει . Η δοκιμή της πλατφόρμας μας έγινε μέσω του JUnit 5 καθώς και Mockito. Ας μελετήσουμε λίγο εκτενέστερα παρόλα αυτα τις τεχνολογίες που χρησιμοποιήθηκαν .



Κεφάλαιο 4

Τεχνική Ανάλυση

Μελετώντας την προηγούμενη ενότητα θα μπορούσαμε να αναρωτηθούμε στο πώς μπορούμε να ενσωματώσουμε όλες τις τεχνολογίες που αναφέρθηκαν καθώς και εάν είναι εφικτό το integration σε ένα project Java.



Στον βασικό της πυρήνας χρησιμοποιήσαμε την γλώσσα προγραμματισμού Java. Ο τρόπος λειτουργίας της Java έχοντας φτάσει στην έκδοση που μιλάμε σήμερα βρίσκεται στην 17.01 LTS(long-term support) παραμένει ο ίδιος καθώς υπάρχουν μόνο προσθήκες σε επίπεδο λειτουργικότητας.

Δημιουργούμαι .Java αρχεία τα οποία περιέχουν τον πηγαία κώδικας μας καλούμε τον Java Compiler ώστε να αναλύσει για τυχόν σφάλματα καθώς και ασυμβατότητες. Υστέρα ο compiler παράγει απο τα .java αρχεία μας .class αρχεία τα οποία είναι της μορφής binary ώστε να μπορούν να τρέξουν σε ένα Server. Αρκετές φορές μπορούμε να το ακούσουμε ως JVM(Java Virtual Machine) ώστε να μπορεί να τρέξει τον πηγαίο κώδικα μας.

Η παρούσα υλοποίηση ηλεκτρονικού πρωτοκόλλου θα μπορούσαμε με σιγουριά να την υλοποιήσουμε με την χρήση αποκλειστικά της Java αλλά αυτό σε καμία περίπτωση δεν θα ήταν maintainable σε μελλοντικό επίπεδο. Η Java μας δίνει την δυνατότητα όμως να χρησιμοποιήσουμε βοηθητικά frameworks με την μορφή layers στα module όπως είδαμε στην προηγούμενη ενότητα. Εμείς επιλέξαμε το Spring Frameworks το οποίο έχει αποδείξει την αξία του καθώς και την λειτουργικότητα του σε επίπεδο χρόνου. Είναι σημαντικό να αναφέρουμε ότι το Spring Framework έχει χρησιμοποιηθεί από διάφορες εταιρείες . Η πιο γνωστή contributor είναι η εταιρεία Netflix όπου έχουν βοηθήσει σε μεγάλο βαθμό το Spring Framework.

Το Spring Framework κατάφερε να ενσωματόσει το γνωστό MVC pattern το οποίο με την σειρά μας το χρησιμοποιήσαμε ώστε να μπορούμε να οργανώσουμε το κωδικά μας.

Απο το Spring Framework χρησιμοποιήσαμε συνολικά τέσσερα layer.

1.Web Layer

2.Service Layer

3.Repository Layer

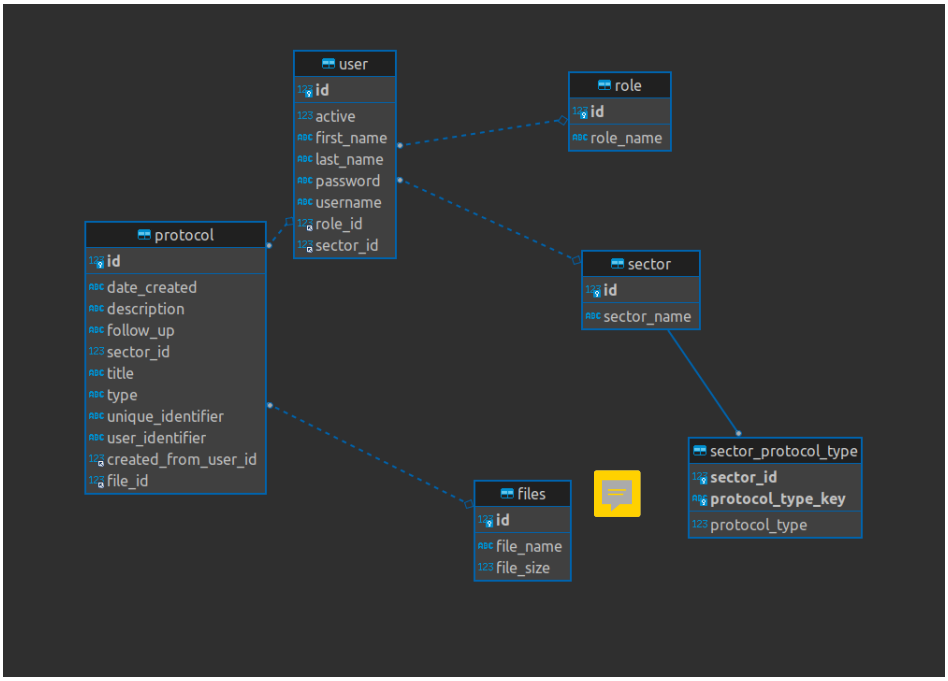
4.Security Layer

Για να καταφέρουμε όμως να εναρμονιστούν και να παραχθεί η πληροφορία πρέπει να τα στεγάσουμε έναν Server όπου στην παρούσα υλοποίηση χρησιμοποιήσαμε τον Tomcat Server.

Όπως είδαμε στην παραπάνω εικόνα ο χρήστης προσπαθεί να κάνει ένα HTTP request επισκέπτοντας την σελίδας μας. Ουσιαστικά καλέι τον Tomcat Server ο οποίος στεγάζει τα συνολικά Layers . Αρχικά ενεργοποιείτε το Security Layer και δημιουργεί μια αλυσίδα ερωτήσεων ώστε να καταλάβει εάν είναι διαπιστευμένος χρήστης καθώς και τι ενέργειες μπορεί να πράξει(Authorization - Authentication). Ο χρήστης δίνει τα διαπιστευτήρια στο Security Layer. Το Security Layer με την σειρά του καλέι το Service Layer. Το Service Layer είναι το Layer το οποίο είναι υπεύθυνο για την λειτουργικότητα. Πραγματοποιεί τις ενέργεις τις οποίες ο προγραμματιστής έχει ορίσει ώστε να γίνει η ταυτοποίησή . Στην δική μας περίπτωση λαμβάνουμε τα διαπιστευτήρια τα αποκρυπτογραφούμε και καλούμε το Repository Layer. Στο Repository Layer επικοινωνούμε με την βάση δεδομένων μας και εκτελούμε μια ερώτηση εάν ο χρήστης είναι όντως αυτός που λέει και τι ενέργειες μπορεί να παράξει .

Εάν το αποτέλεσμα της ερώτησης είναι σωστό ο Tomcat Server ενεργοποιεί το επόμενο Layer όπου κάνει resolve την οπτικοποίηση της ιστοσελίδας μας σε μορφή HTML μέσω του Thymeleaf. Το Thymeleaf όπως μελετήσαμε στην προηγούμενη ενότητα είναι ένας Web Resolver . Εάν θα έπρεπε να το περιγράψουμε σε λίγες γραμμες εκ νέου τι είναι το Thymeleaf θα μπορούσαμε να πούμε ότι είναι το View του MVC pattern το οποίο πιστά ακολουθούμε κατόπιν οι κατασκευαστές του Framework συνιστούν. Μέσω τα διαπιστευτήρια του χρήστη του δίνουμε την δυνατότητά να πράξει διάφορες λειτουργίες ανάλογα με τον ρόλο του όπως θα δούμε παρακάτω. Ένα πολύ σημαντικό σημείο υλοποίησης το οποίο έχουμε δημιουργήσει είναι το RestApi. Με το RestApi χρησιμοποιούμε την λειτουργικότητα να συνεργαστούν όχι μόνο χρήστες με πρόγραμμα αλλά πρόγραμμα με πρόγραμμα. Όπως αναφερθήκαμε στην εισαγωγή, είναι πολύ σημαντικό το επίπεδο μετεξέλιξης ώστε να χρησιμοποιηθεί και με διαφορετικές πλατφόρμες . Το RestApi έχει την ίδια λειτουργικότητα με το κανονικό HttpRequest άλλα η κεντρική διαφορά του είναι ο τρόπος με τον οποίο ανταλλάσουμε δεδομένα αφού είναι σε μορφή JSON.

Εκτενέστερα πρέπει να γνωρίζουμε ότι στον Tomcat Server ενεργοποιούμε μέσω του Spring Framework ένα signature σε κοινόχρηστη χρήση. Λέγοντας signature αναφερόμαστε σε ένα url στο οποίο ρυθμίζουμε οι ίδιοι την λειτουργικότητα που θέλουμε να χρησιμοποιήσουμε. Στην δική μας περίπτωση χρησιμοποιήσαμε ένα συγκεκριμένο signature ώστε να δώσουμε την ευκαιρία σε άλλα προγράμματα καθώς και χρήστες να δημιουργήσουν ένα συγκεκριμένο πρωτόκολλο δίνοντας τα κατάλληλα διαπιστευτήρια .


user
id
active
first_name
last_name
password
username
role_id
sector_id
role
id
role_name
protocol
id
date_created
description
follow_up
sector_id
title
type
unique_identifier
user_identifier
created_from_user_id
file_id
sector
id
sector_name
sector_protocol_type
sector_id
protocol_type_key
protocol_type
files
id
file_name
file_size

Κεφάλαιο 5

Λειτουργικότητα & Γραφική διασύνδεση

5.1 Λειτουργικότητα

Η λειτουργικότητα της εφαρμογή χωρίζεται σε τρεις βασικούς πυλώνες με τους οποίους καλούμαστε ανάλογα με τον χρήστη που είμαστε συνδεδεμένοι να εκτελέσουμε προκαθορισμένες ενέργειες. Ακριβέστερα η πλατφόρμα μας εξυπηρετεί φορείς. Ο κάθε φορέας είναι υπεύθυνος ώστε να στεγάσει μια ομάδα χρηστών δίνοντας τους την ευχέρεια ανάλογα με τα δικαιώματα τους να πραγματοποιήσουν συγκεκριμένες ενέργειες . Οι ενέργειες που μπορούν να παρθούν ανάλογα με τους χρήστες χωρίζονται σε τρεις κατηγορίες όπως θα δούμε παρακάτω εκτενέστερα.

5.1.1 Φορέας

Ο φορέας είναι η συλλογή όπου ανήκουν όλοι οι χρήστες. Μέσα στον φορέα μπορούμε να προσθέσουμε χρήστες, να διαγράψουμε χρήστες καθώς και να τον μετονομάσουμε τον ίδιο τον φορέα. Όπως είδαμε στην προηγούμενη εικόνα οι λειτουργικότητες του φορέα ανήκουν στους χρήστες GODUSER.

5.1.2 Χρήστες

Η κατηγορία χρήστες δίνει τη δυνατότητα επεξεργασία των χρηστών ως προς το πλήθος τους καθώς και των ιδιοτήτων τους. Οι ιδιότητες που φέρει ο κάθε χρήστης είναι :

1.Αναγνωριστικό αριθμός

2.Όνομα

3.Επίθετο

4.Username

5.Τελευταία σύνδεση

6.Ρόλος

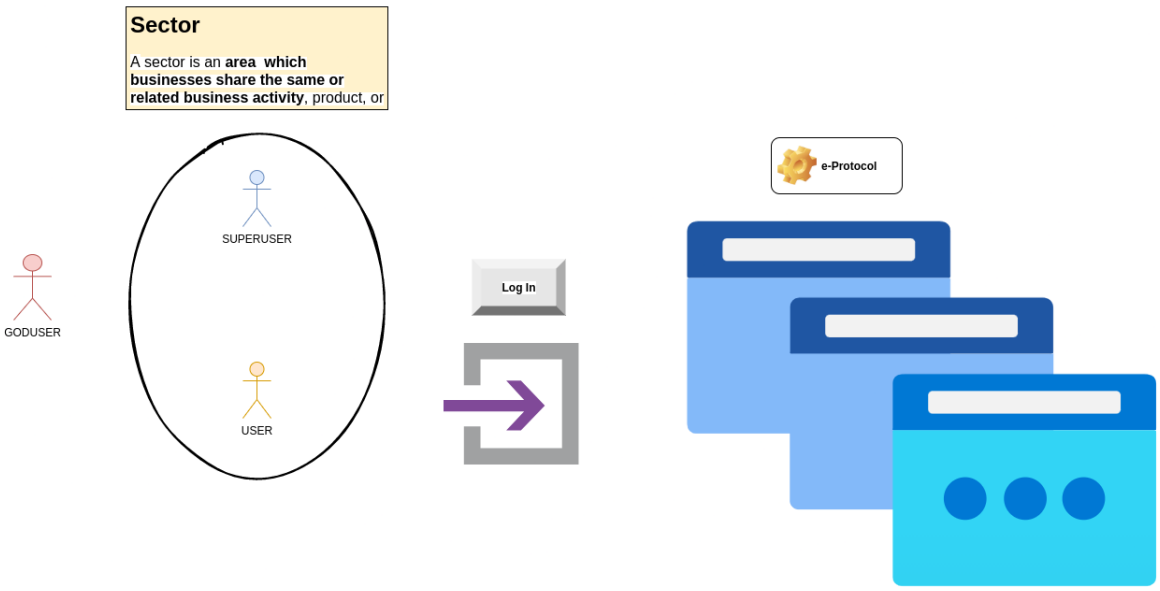
7.Φορέας

	Δημιουργία / Διαγραφή / Επεξεργασία Φορέα	Δημιουργία / Διαγραφή / Επεξεργασία Χρηστών	Δημιουργία / Διαγραφή / Επεξεργασία Πρωτοκόλλου
God_User	☑	☑	☑
Super_User	X	☑ Για τον εκάστοτε φορέα	☑ Για τον εκάστοτε φορέα
User	X	X	☑ Για τον εκάστοτε χρήστη

5.1.3 Πρωτόκολλο

Η τελευταία καθώς και μεγαλύτερη κατηγορία στην πλατφόρμα μας είναι τα Πρωτόκολλα. Στην κατηγορία αυτή μπορούμε να δημιουργήσουμε ένα πρωτόκολλο. Το κάθε πρωτόκολλο ανήκει σε έναν φορέα . Μέσα στο εκάστοτε πρωτόκολλο μας δίνεται η δυνατότητα να δημιουργούμε Εισερχόμενα-Εξερχόμενα, να επισυνάπτουμε αρχεία καθώς και να συμπληρώνουμε τον τίτλο, αποστολέα ,περιγραφή ,αρχεία . Όπως θα δούμε στην επόμενη ενότητα μέσω της γραφικής διασύνδεσης δίνεται η δυνατότητα να επεξεργαστούμε όχι μόνο το περιεχόμενο του πρωτοκόλλου άλλα και την ύπαρξη του .

5.2 Γραφική διασύνδεση



Οι συνολικοί χρήστες που μπορούν να χρησιμοποιήσουν το σύστημα μας είναι τρείς.

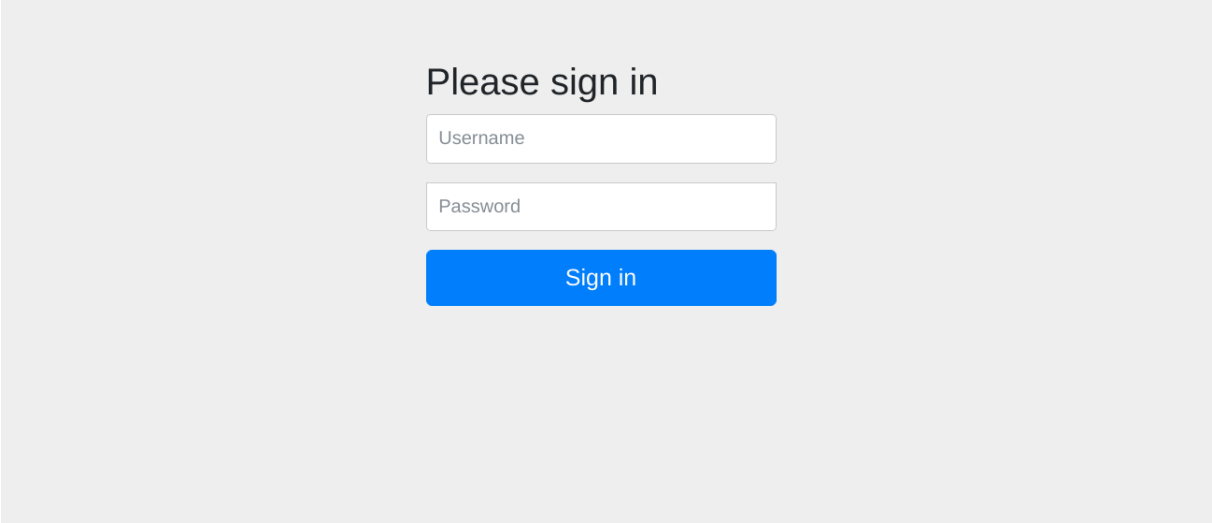
1. GODUSER
2. SUPERUSER
3. USER

Οι χρήστες αυτοί διέπονται απο κανόνες τους οποίους μας δίνεται η δυνατότητα για μια λειτουργικότητα κάθε αυτή στον καθένα. Όλοι οι χρήστες του συστήματος ανήκουν σε έναν οργανισμό/Sector. Οι οργανισμόι/Sectors ορίζονται απο τους GODUSER της πλατφόρμας, οποιαδήποτε

ενέργεια επιτυγχάνεται μέσα σε ένα οργανισμό , το γνωρίζει μόνο ο οργανισμός και δεν έχουν πρόσβαση κανένας άλλος παρα μόνο τα μελοι που τον διέπουν.

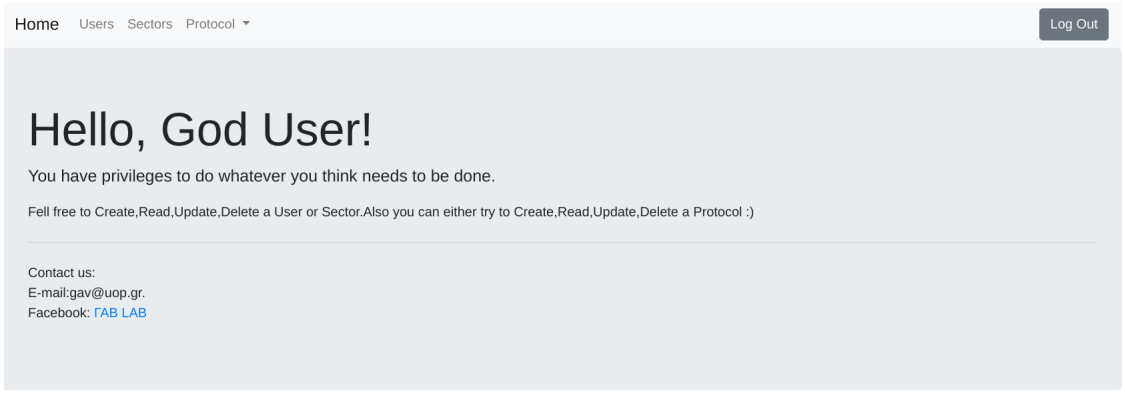
Εάν θελλήσουν οι οργανισμοί να επικοινωνήσουν μεταξύ τους μπορούν μόνο μέσω των GODUSER. Ο κάθε χρήστης όπως θα δούμε και παρακάτω μπορεί να πραγματοποιήσει συγκεκριμένες ενέργειες σύμφωνα με τον τίτλο που φέρει .

Κατα την είσοδο του στον οργανισμό του ζητείται να προσθέσει τα αναγνωριστικά του username,password. Η σελίδα που συναντάμε και οι τρείς χρήστες κατα την είσοδο τους απεικονίζεται ώς εξείς



Εάν έχει πραγματοποιηθεί επαλύθευση απο το σύστημα ότι ο χρήστης είναι εγκεκριμένος να εισέλθει τότε υπάρχει η κεντρική οθόνη όπου μπορεί να επιλέξει διάφορες λειτουργίες .

5.2.1 GODUSER

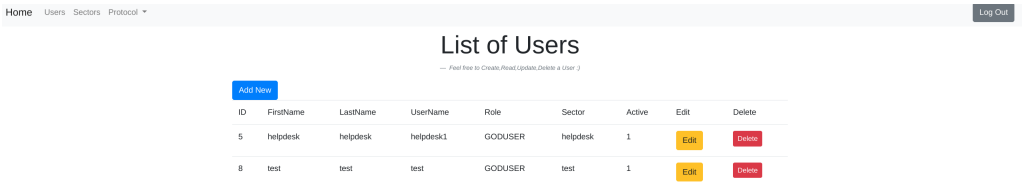


Όπως μπορούμε να δούμε πάνω στο NavBar ο χρήστης μπορεί να επιλέξει ανάμεσα σε τρείς κατηγορίες

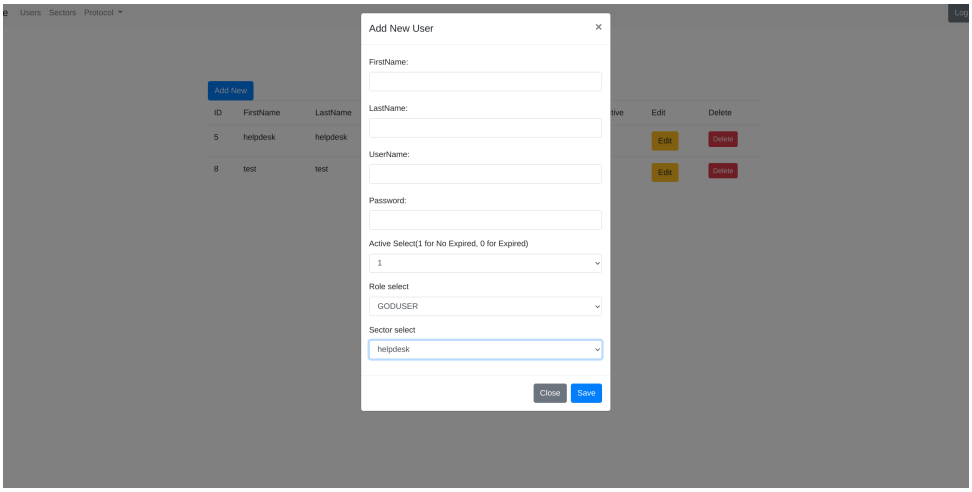
1. Users
2. Sectors

3. Protocol

Στην κατηγορία Users μπορεί να επεξεργαστει χρήστες, να τροποποιήσει τα στοιχεία τους καθώς και να διαγραψεί αν χρειστεί, ακόμα μπορεί και να δημιουργήσει και καινούργιους .

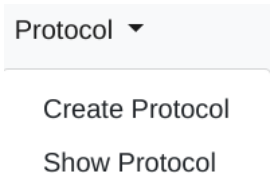


Πατόντας το κουμπί Add New ένα παράθυρο εμφανίζεται στην σελίδα του χρήστη

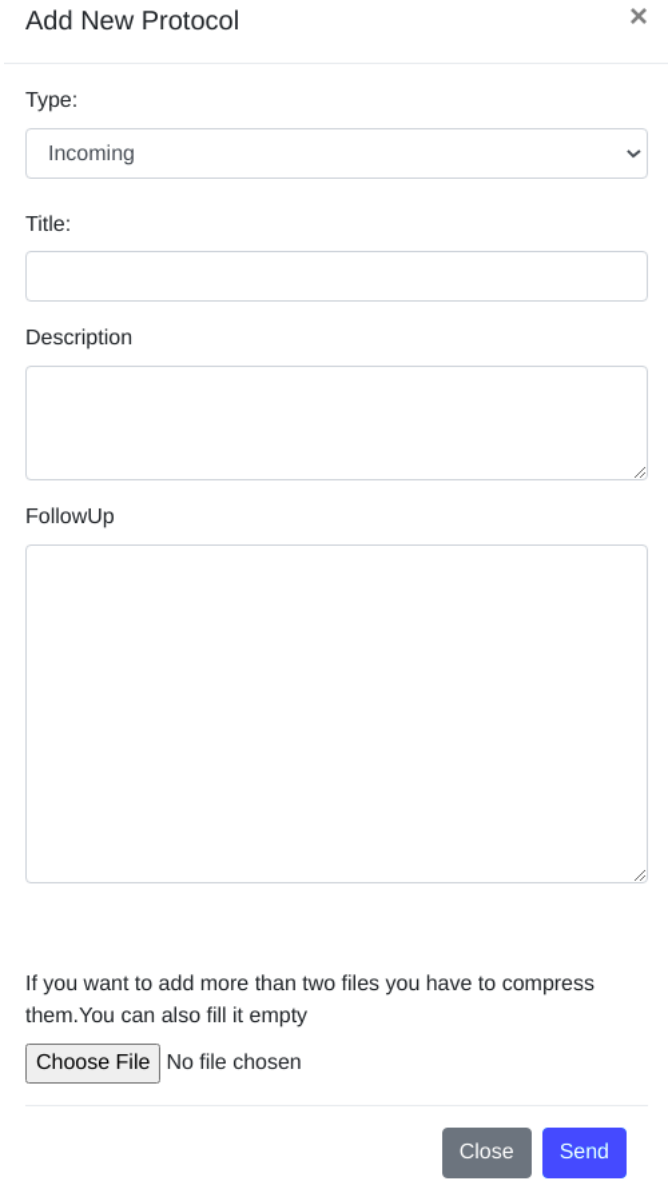


Προσθέτοντας έναν χρήστη μπορούμε να δούμε ότι πατόντας το κουμπί EDIT μπορούμε να τροποποιήσουμε οποιαδήποτε στοιχείο που συμπληρώσαμε πρίν .

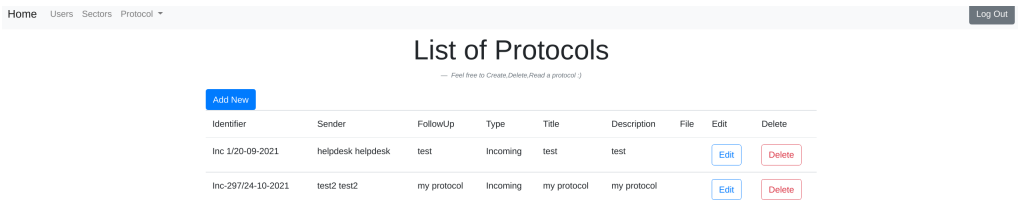
Η έκδοση ενος πρωτοκόλλου είναι από τις σημαντικότερες καθώς και πιο συχνές λειτουργίες που θα εκτελεί ο χρήστης μέσα στην πλατφόρμα , προσπαθήσαμε να ειναι απο ένα εύχρηστο UI ώστε να μπορεί ο κάθε χρήστης να έχει αμεσότητα . Πατόντας απο το NavBar στην κατηγορία πρωτόκολλο εμφανίζεται οι παρών επιλογές



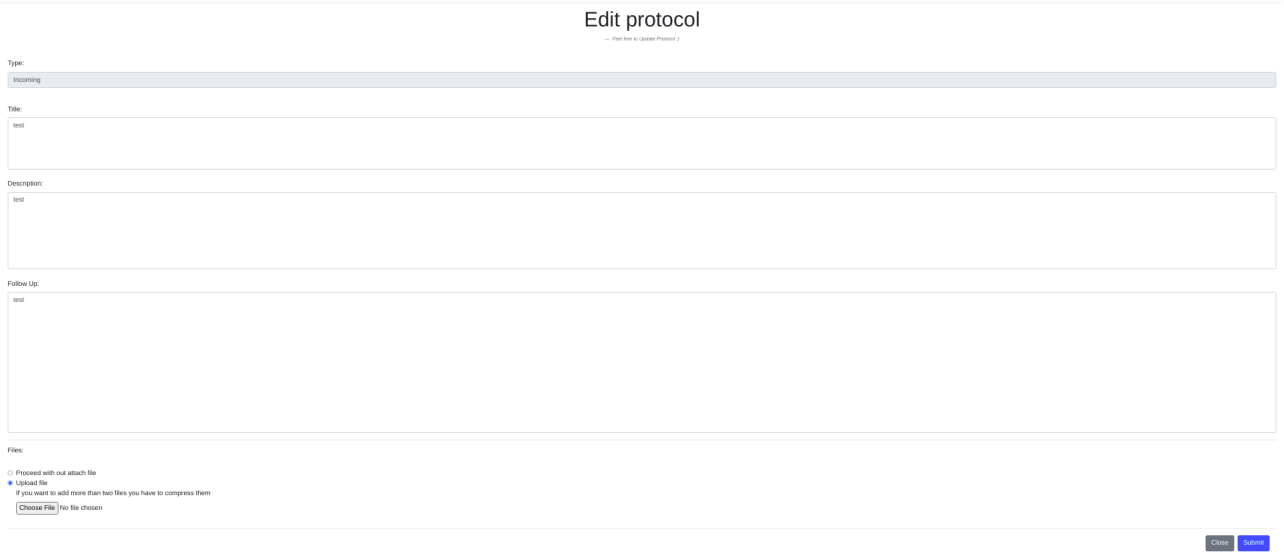
Πατόντας ο χρήστης στην επιλογή Create Protocol εμφανίζεται ενα αναδιπλούμενο παράθυρο με τις εξείς επιλογές .



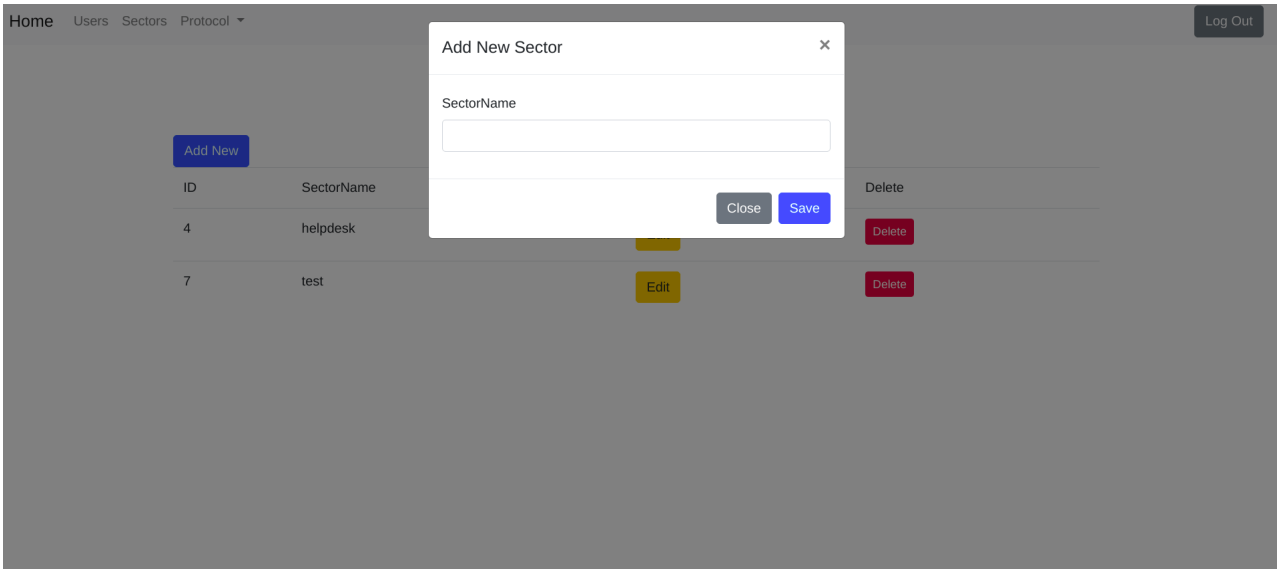
Έχοντας συμπληρώσει όλα τα απαραίτητα πεδία και πατόντας υποβολή πλοηγείται στην δεύτερη επιλογή.



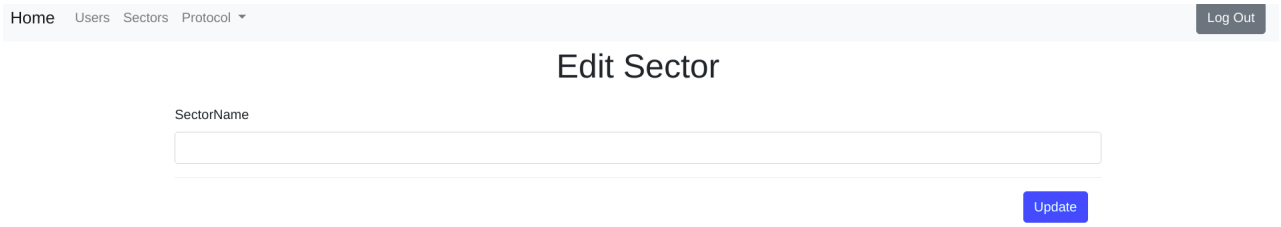
Στην επιλογή Add New θα εμφανιστεί εκ νέου το αναδυπλούμενο παράθυρο το οποίο ειναι για την προσθήκη ενος καινούργιου πρωτοκόλλου. Όπως προαναφερθήκαμε σε προηγούμενες ενότητες ο GODUSER έχει τα μεγαλύτερα δικαιώματα μέσα στην πλατφόρμα, συνεπώς μπορει να επεξεργαστεί πλήρως όλες τις πληροφορίες της σουίτας καθώς και να διαγράψει όποια θεωρεί μη αναγακαία . Πατόντας την επιλογή Edit εμφανίζεται το παράθυρο επεξεργασίας



Μέσα στις διαθέσιμες επιλογές δίνεται και η ευκαιρία να δημιουργήσει έναν καινούργιο οργανισμό πατώντας την κατηγορία Sectors.

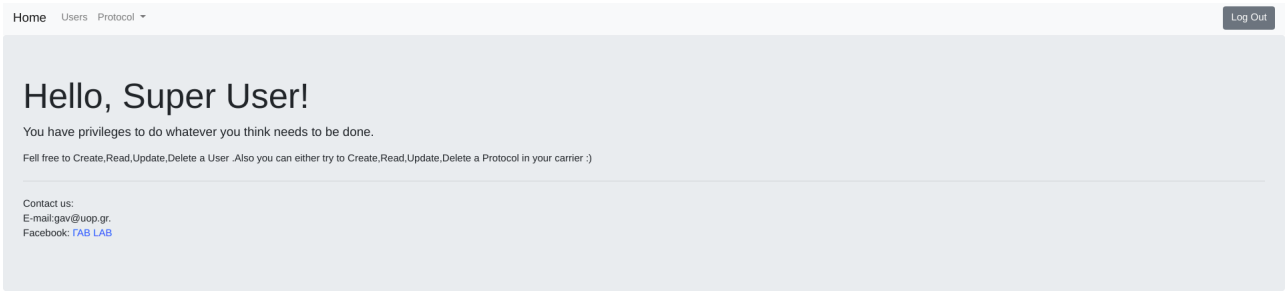


Ακόμα είναι πολύ χρήσιμο να αναφερθεί ότι εκτός από την διαγραφή του μπορούμε και να επεξεργαστούμε τα στοιχεία του και πιο συγκεκριμένα το όνομα του .



5.2.2 SUPERUSER

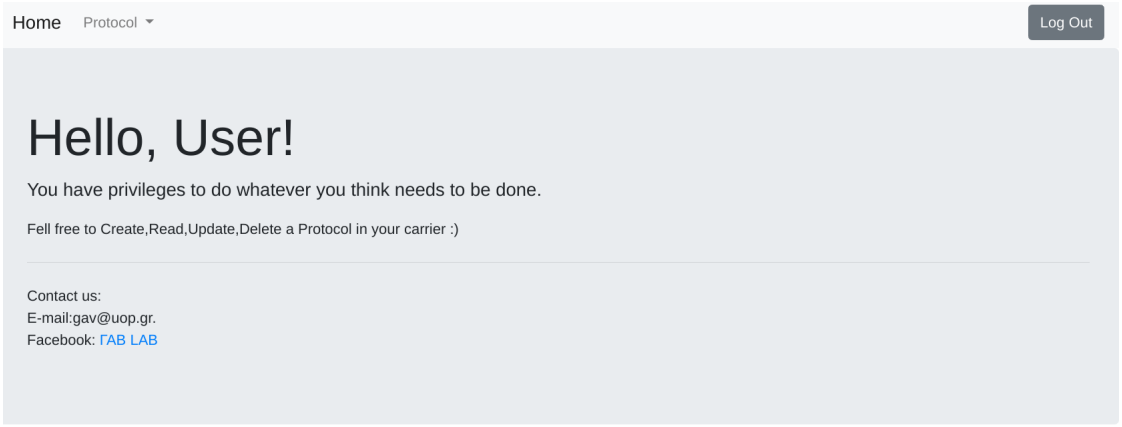
Οι SUPERUSER είναι εξίσου μια ομάδα που έχει αρκετά δικαιώματα , μπορεί με την σειρά της να εισάγει / διαγράψει / επεξεργαστεί αρκετά δεδομένα σε επίπεδο χρηστών καθώς και πρωτοκόλλων , καθώς και να δημιουργήσει ένα ερώτημα στους GODUSERS που ανήκουν στον φορέα τους. Οι SUPERUSER επισκέπτοντας την σελίδα συναντάνε την είσοδο χρήστη όπως οι GODUSER , αλλα έχουν διαφορετικές επιλογές .



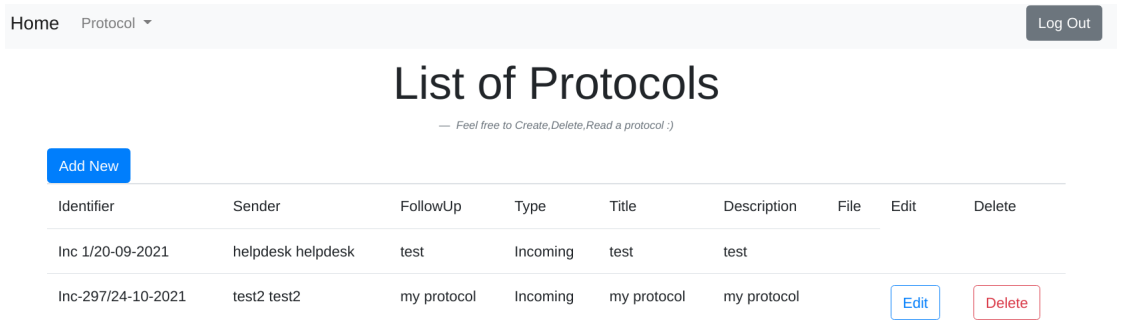
Απο το NavBar μπορούν να επεξεργαστούν τους USERS ως προς τα στοιχεία τους καθώς και να δημιουργήσουν κάποιον καινούργιο . Πολυ συμαντικό ειναι να αναφέρουμε ότι δεν μπορούν να τροποποιήσουν κανένα στοιχείο των GODUSER παρα μόνο να δούν ποιοι ειναι . Επιλέγοντας την καρτέλα Protocol μπορούν να δημιουργήσουν εκ νέου ένα πρωτόκολλο καθώς και να επεξεργαστούν όλο τα πρωτόκολλα των User και των δικών τους . Η γραφική διασύνδεση που έχει χρησιμοποιηθεί είναι ίδια με των GODUSER, με λιγότερες λεπτομέρειες ως προς την πληροφορία.

5.2.3 USER

Οι χρήστες με τον τίτλο USERS έχουν αρκετούς περιορισμούς ώς προς τα δικαιώματα τα οποία μπορούν να πράξουν μέσα στην πλατφόρμα .



Παρατηρώντας τις επιλογές μπορούμε να δούμε ότι έχουν μόνο την κατηγορία Protocol διότι τους επιτρέπεται μόνο να δημιουργήσουν ένα καινούργιο πρωτόκολλο καθώς και να επεξεργαστούν αποκλειστικά δικά τους πρωτόκολλα .



5.3 Σύνδεση μέσω τερματικού

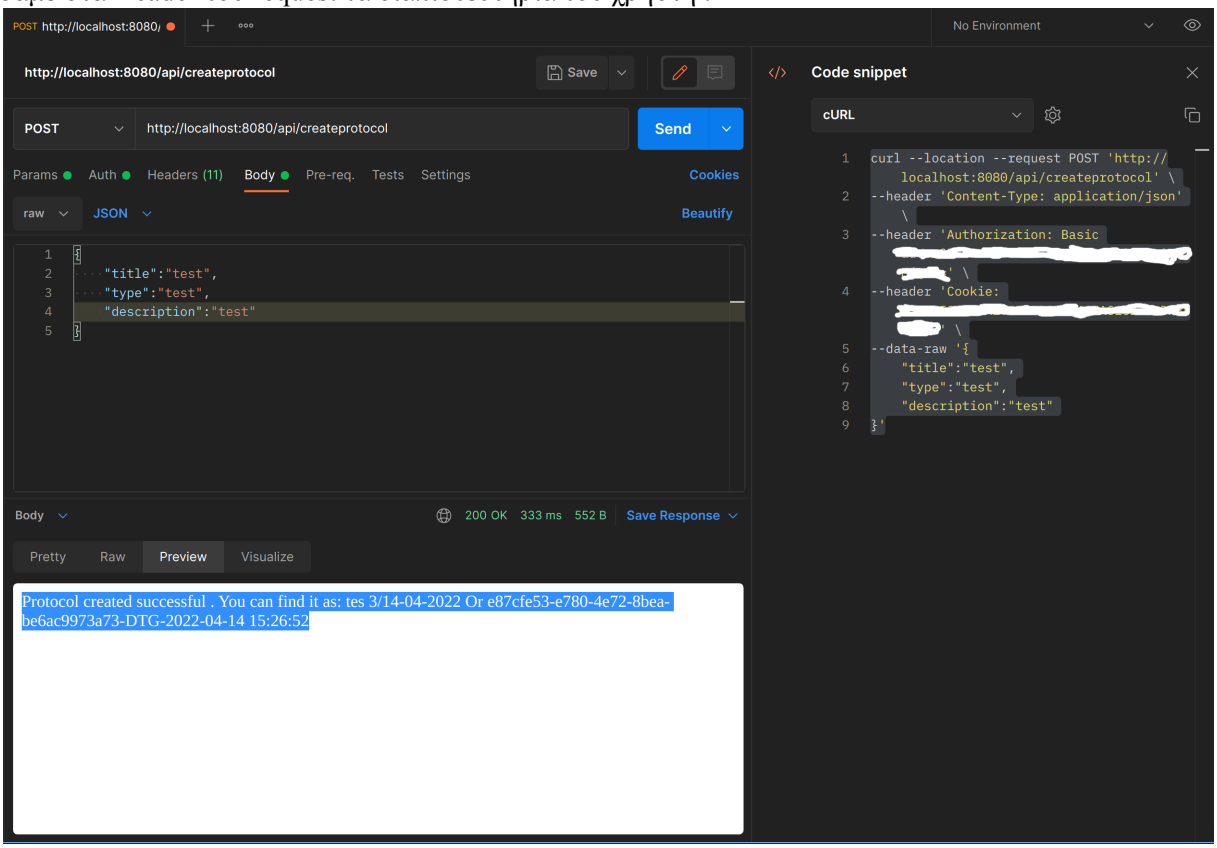
Το integration είναι από τα σημαντικότερα ώστε να μπορέσουμε να ενοποιήσουμε πολλαπλές πλατφόρμες στην επικοινωνία του. Η ενοποίηση προκύπτει από την ανάγκη των φορέων διότι σε καθημερινό επίπεδο μεγαλώνουν και δημιουργούνται καινοτόμες πλατφόρμες όπου λύνουν ιδιαίτερα προβλήματα.

Για να καταφέρουμε την επικοινωνία της κάθε πλατφόρμας με την δική μας επιλέξαμε να δημιουργήσουμε RestAPI. Με την δυνατότητα το RestAPI δημιουργήσαμε ένα endpoint όπου ο κάθε χρήστης μπορεί να δημιουργήσει ένα ηλεκτρονικό πρωτόκολλο δίχως να χρειαστεί να συνδεθεί στην γραφική διασύνδεση.

Όπως μπορούμε να καταλάβουμε προκύπτει από το γεγονός ότι ο καθημερινός χρήστης δεν θα του ήταν ιδιαίτερα εύκολο να εκδώσει μέσω του τερματικού του. Για αυτό ενδείκνυται για την επικοινωνία μεταξύ εφαρμογών.

Η επίτευξη ώστε να δημιουργηθεί η έκδοση ενός πρωτοκόλλου χρειάζεται 1.Τίτλο, 2.Τύπο, 3.Περιγραφή

Παρόλα αυτά είναι σημαντικό να διατηρήσουμε την ασφάλεια της εφαρμογής για αυτό προσθέσαμε στα Header του request τα διαπιστευτήρια του χρήστη .



Κεφάλαιο 6

Προοπτικές εξέλιξης-Συμπεράσματα

Αναλύοντας τα προβλήματα των είδη υπάρχων υλοποιήσεων, ανακαλύψαμε διάφορες μεταβλητές οι οποίες είτε δεν είχαν υλοποιηθεί με τον καλύτερο τρόπο, είτε δημιουργούσαν μεγάλη μετεξέλιξη σε πιθανούς ενδιαφερόμενους όπου θα ήθελαν να προσθέσουν σε μελλοντικό στάδιο μια περαιτέρω προσθήκη βασίζοντας πάντα στην είδη υπάρχουσα ανάλυση .

6.1 Θετικα

Η πλατφόρμα έχει αρκετά θετικά στοιχεία, η διαφοροποίηση όμως απο υπάρχουσες υλοποιήσεις εξαρτάται απο κάποια μοναδικά στοιχεία.

1.Rest API

2.Διαμόρφωση αρχείων βάση αναγκών συστήματος

3.Κρυπτογραφία σε ευαίσθητα δεδομένα

4.Χρήση τεχνολογιών και αυτόματης ενημέρωσης σε επίπεδο ασφάλειας

5.Δυναμική εκχώρηση δεδομένων

6.2 Αρνητικά

1.Έλλειψη διπλής χειραψίας κατα το Authentication

2.Απουσία βέλτιστης γραφικής διασύνδεσης

3.Μη υποστήριξη απο άτομα με ειδικές ανάγκες

4.Απουσία Microservices

Συνοψίζοντας παρατηρήσαμε την ανάγκη όλων των φορέων ώστε να δημιουργηθεί ένα πρωτόκολλο εναλλακτικού τύπου απο τις καθιερωμένες λύσεις που υπήρχαν μέχρι τώρα στην επικοινωνία φορέων και χρηστών με σκοπό την αμεσότερη επικοινωνία που μπορεί να επιφέρει βέλτιστες λύσεις στην λειτουργικότητα και την παραγωγικότητα των μέλλων που την διέπουν .

6.3 Συμπεράσματα

Κατα την σταδιακή ανάπτυξη σε επίπεδο κώδικα παρατηρήσαμε πολλαπλές προσθήκες που θα χρειαστούν σε μελλοντικό χρόνο διότι καθημερινά οι οργανισμοί καθώς και τα στελέχη τους αναπτύσσονται με αυξητικούς ρυθμούς. Στα βασικά μειονεκτήματα θα μπορούσαμε να προσθέσουμε ότι υπάρχει έλλειψη στην συνεργασία της πλατφόρμας με άτομα ειδικών αναγκών, αφού σε παρόμοιες υλοποιήσεις υπάρχει ειδική συντομογραφία ώστε να γίνεται ευκολότερη η χρήση της πλατφόρμας . Ένα ακόμα όφελος που σε μελλοντικό στάδιο θα μπορεί να δώσει μεγάλο προβάδισμα έναντι των άλλων υλοποιήσεων είναι η κλήση με Rest Api . Με την χρήση του μηχανισμού

Rest Api εκπληρώσαμε την ανάγκη των φορέων να μπορούν σε οποιαδήποτε πλατφόρμα να ενσωματώσουν την δημιουργία/εγγραφή/ενημέρωση ενός ηλεκτρονικού πρωτοκόλλου .

Η χρήση των ηλεκτρονικών πρωτοκόλλων προσφέρει βέλτιστες λύσεις στην επικοινωνία καθώς και στην αρχειοθέτηση τεράστιων όγκων πληροφορίας σε μία μικρή οθόνη κινητού. Καταλαβαίνουμε συνεπώς ότι η ποιότητα των υπηρεσιών μετεξελίσσεται και παράγει αποτελέσματα που δίνουν στα μέλη τους ευκινησία στον διαδικτυακό χώρο. Αυτό αντικατοπτρίζεται στην αποδοτικότητα των εργαζομένων και των οργανισμών αφού μπορούν μέσα σε τόσο λίγο χρόνο να παράγουν ή να αναζητήσουν τεράστιο όγκο πληροφορίας .

Βιβλιογραφία

[] *Baeldung Guides*. URL: https://www.baeldung.com/rest-with-spring-series.

[] *MySQL Documentations*. URL: https://dev.mysql.com/doc/.

[] *Spring Boot in Action Book by Craig Walls*. URL: https://books.google.gr/books/about/Spring_Boot_in_Action.html?id=IzkzEAAAQBAJ&printsec=frontcover&source=kp_read_button&hl=en&redir_esc=y#v=onepage&q&f=false.

[] *Spring Documentation*. URL: https://docs.spring.io/spring-framework/docs/current/reference/html/.

[] *Spring Security Best Practices*. URL: https://github.com/spring-projects/spring-security#readme.

[] *Thymeleaf Documentations*. URL: https://www.thymeleaf.org/documentation.html.