1/11/2023

Ομαδική Εργασία Μαθήματος

Καθηγητής: Κυριάκος Κρητικός

**321-4002 – Τεχνολογία Λογισμικού**

Μέλη:

* icsd14092 Κοτσαράπογλου Ιάσων - Ειρηναίος
* icsd17212 Χαμακιώτη Ελένη

Περιεχόμενα

[Σύνοψη 2](#_Toc124069699)

[Οργάνωση 2](#_Toc124069700)

[Α. Βασικοί ρόλοι 2](#_Toc124069701)

[Β. Βασικά στάδια ανάπτυξης 2](#_Toc124069702)

[Απαιτήσεις 3](#_Toc124069703)

[Α. Λειτουργικές 3](#_Toc124069704)

[Β. Μη Λειτουργικές 5](#_Toc124069705)

[Σχεδίαση 6](#_Toc124069706)

[Α. Αρχιτεκτονική & Περιβάλλον Συστήματος 6](#_Toc124069707)

[Component diagram 6](#_Toc124069708)

[Β. Επιχειρησιακές Διαδικασίες Συστήματος 7](#_Toc124069709)

[Process diagrams 7](#_Toc124069710)

[Class diagram 15](#_Toc124069711)

[Δ. Δεδομένα 16](#_Toc124069712)

[ERD 16](#_Toc124069713)

[Υλοποίηση 17](#_Toc124069714)

[Α. URL αποθετηρίου GitHub 17](#_Toc124069715)

[Β. Τεκμηρίωση Υλοποίησης 17](#_Toc124069716)

[Συμπεράσματα 19](#_Toc124069717)

# Σύνοψη

Για την εργασία αυτή αναπτύχθηκε μια RESTful εφαρμογή όπου κάθε ενέργεια η οποία ικανοποιεί μια λειτουργική απαίτηση πραγματοποιείται μέσω ενός endpoint. Για όλες τις οντότητες του συστήματος παρέχονται οι βασικές C(create) R(read) U(update) D(delete) λειτουργίες οι οποίες ανάλογα με το ρόλο του χρήστη μετά την αυθεντικοποίησή του η και χωρίς αυτή (επισκέπτης) παράγουν/ φιλτράρουν τα αναμενόμενα αποτελέσματα.

# Οργάνωση

## Α. Βασικοί ρόλοι

Δεν μπορέσαμε να βρούμε δίκαιο τρόπο για να χωρίσουμε τους ρόλους στα δύο γιατί κάποιος από τους δύο θα φορτωνόταν πολύ, ενώ ο άλλος όχι. Επομένως αποφασίσαμε να μην έχουμε συγκεκριμένους ρόλους κατά την διάρκεια της ανάπτυξής αλλά να δουλεύουμε παράλληλα και την ίδια στιγμή έτσι ώστε να έχουμε την ίδια εικόνα της εργασίας και οι δύο.

## Β. Βασικά στάδια ανάπτυξης

Ο συνολικός χρόνος της εργασίας είναι περίπου 1,5 μήνας. Ξεκινήσαμε να ασχολούμαστε με την εργασία στις 25 Νοέμβριου 2022 και την ολοκληρώσαμε 09 Ιανουαρίου 2022

* 1ο Μέρος Εργασίας – Απαιτήσεις: 25 Νοέμβριου 2022 - 28 Νοέμβριου 2022
* 2ο Μέρος Εργασίας – Σχεδίαση: 30 Νοέμβριου 2022 -5 Δεκεμβρίου 2022
* 3ο Μέρος Εργασίας – Υλοποίηση: 6 Δεκεμβρίου – 30 Δεκεμβρίου 2022
* Καταγραφή αναφοράς: 03 Ιανουαρίου 2023 - 09 Ιανουαρίου 2022

# Απαιτήσεις

## Α. Λειτουργικές

Μια λειτουργική απαίτηση είναι μια δήλωση του πώς πρέπει να συμπεριφέρεται ένα σύστημα. Καθορίζει τι πρέπει να κάνει το σύστημα προκειμένου να ανταποκριθεί στις ανάγκες ή τις προσδοκίες του χρήστη. Οι λειτουργικές απαιτήσεις μπορούν να θεωρηθούν ως χαρακτηριστικά που εντοπίζει ο χρήστης. Διαφέρουν από τις μη λειτουργικές απαιτήσεις, οι οποίες καθορίζουν τον τρόπο λειτουργίας του συστήματος εσωτερικά (π.χ. απόδοση, ασφάλεια κ.λπ.).

Οι λειτουργικές απαιτήσεις αποτελούνται από δύο μέρη: τη λειτουργία και τη συμπεριφορά. Η λειτουργία είναι αυτό που κάνει το σύστημα. Η συμπεριφορά είναι ο τρόπος με τον οποίο το κάνει το σύστημα. Στην παρούσα εργασία οι λειτουργικές απαιτήσεις είναι εμφανής από την εκφώνηση. Εμείς καταγράψαμε τις παρακάτω:

* Ένας επιμελητής πρέπει να έχει την δυνατότητα να δημιουργεί ειδήσεις.
* Ένας επιμελητής πρέπει να έχει την δυνατότητα να τροποποιεί ειδήσεις.
* Ένας επιμελητής πρέπει να έχει την δυνατότητα να υποβάλλει ειδήσεις.
* Ένας επιμελητής πρέπει να έχει την δυνατότητα να εγκρίνει ειδήσεις.
* Ένας επιμελητής πρέπει να έχει την δυνατότητα να απορρίπτει ειδήσεις.
* Ένας επιμελητής πρέπει να έχει την δυνατότητα να δημοσιεύει ειδήσεις.
* Ένας επιμελητής πρέπει να έχει την δυνατότητα να αναζήτα ειδήσεις.
* Ένας επιμελητής μπορεί να δει τις ειδήσεις όλων των καταστάσεων.
* Ένας επιμελητής πρέπει να έχει την δυνατότητα να προσθέτει σχόλια.
* Ένας επιμελητής πρέπει να έχει την δυνατότητα να τροποποιεί σχόλια
* Ένας επιμελητής πρέπει να έχει την δυνατότητα να εγκρίνει σχόλια.
* Ένας επιμελητής πρέπει να έχει την δυνατότητα να απορρίπτει σχόλια.
* Ένας επιμελητής μπορεί να δει τα σχόλια όλων των καταστάσεων.
* Ένας επιμελητής πρέπει να έχει την δυνατότητα να δημιουργεί θέματα.
* Ένας επιμελητής πρέπει να έχει την δυνατότητα να τροποποιεί θέματα.
* Ένας επιμελητής πρέπει να έχει την δυνατότητα να εγκρίνει θέματα.
* Ένας επιμελητής πρέπει να έχει την δυνατότητα να απορρίπτει θέματα.
* Ένας επιμελητής μπορεί να δει τα θέματα όλων των καταστάσεων.
* Ένας δημοσιογράφος πρέπει να έχει την δυνατότητα να δημιουργεί ειδήσεις.
* Ένας δημοσιογράφος πρέπει να έχει την δυνατότητα να τροποποιεί ειδήσεις.
* Ένας δημοσιογράφος πρέπει να έχει την δυνατότητα να υποβάλλει ειδήσεις.
* Ένας δημοσιογράφος πρέπει να έχει την δυνατότητα να αναζήτα δημοσιευμένες ειδήσεις ή ειδήσεις που του ανήκουν και βρίσκονται σε άλλη κατάσταση (π.χ. δημιουργημένη).
* Ένας δημοσιογράφος πρέπει να έχει την δυνατότητα να δημιουργεί σχόλια
* Ένας δημοσιογράφος μπορεί να δει μόνο σχόλια που έχουν εγκριθεί από τον επιμελητή.
* Ένας δημοσιογράφος πρέπει να έχει την δυνατότητα να δημιουργεί θέματα.
* Ένας δημοσιογράφος πρέπει να έχει την δυνατότητα να τροποποιεί θέματα.
* Ένας δημοσιογράφος πρέπει να έχει την δυνατότητα να αναζήτα θέματα που έχουν εγκριθεί από έναν επιμελητή ή θέματα που του ανήκουν και βρίσκονται σε άλλη κατάσταση (π.χ. δημιουργημένο).
* Ένας επισκέπτης πρέπει να έχει την δυνατότητα να προσθέτει σχόλια.
* Ένας επισκέπτης πρέπει να έχει την δυνατότητα να αναζητά εγκεκριμένα σχόλια .
* Ένας επισκέπτης πρέπει να έχει την δυνατότητα να αναζητά εγκεκριμένα θέματα.
* Ένας επισκέπτης πρέπει να έχει την δυνατότητα να αναζητά δημοσιευμένες ειδήσεις.
* Κατά την δημιουργία μιας είδησης θα πρέπει να δίνονται τίτλος περιεχόμενο και τουλάχιστον ένα σχετικό θέμα.
* Κατά την δημιουργία μιας είδησης το σύστημα θα πρέπει να αποθηκεύει αυτόματα το αναγνωριστικό της είδησης, την ημερομηνία δημιουργίας της καθώς και την κατάσταση της που πρέπει να είναι δημιουργημένη.
* Για να γίνει η τροποποίηση μιας είδησης από έναν δημοσιογράφο ή έναν επιμελητή πρέπει να γίνεται έλεγχος από το σύστημα αν είναι σε κατάσταση δημιουργημένη.
* Κατά την υποβολή μιας είδησης το σύστημα θα πρέπει να αποθηκεύει την κατάσταση της που πρέπει να είναι υποβεβλημένη.
* Κατά την αποδοχή μιας είδησης το σύστημα θα πρέπει να αποθηκεύει την κατάσταση της που πρέπει να είναι εγκεκριμένη.
* Κατά την απόρριψη μιας είδησης το σύστημα θα πρέπει να αποθηκεύει την κατάσταση της που πρέπει να είναι δημιουργημένη.
* Ο επιμελητής κατά την απόρριψη της είδησης θα πρέπει να αναφέρει και τον λόγο απόρριψης.
* Ο λόγος απόρριψης θα πρέπει να αποθηκεύεται στην βάση δεδομένων
* Κατά την δημοσίευση μιας είδησης το σύστημα θα πρέπει να αποθηκεύει αυτόματα την ημερομηνία δημοσίευσης της καθώς και την κατάσταση της που πρέπει να είναι δημοσιευμένη.
* Κατά την δημιουργία ενός σχολίου το σύστημα θα πρέπει να αποθηκεύει αυτόματα το αναγνωριστικό του σχολίου, καθώς και την κατάσταση του που πρέπει να είναι δημιουργημένο.
* Κατά την δημιουργία ενός σχολίου θα πρέπει να δίνεται το περιεχόμενο και αν πρόκειται για επισκέπτη σε περίπτωση που το θέλει δίνει και το όνομά του. Αν πρόκειται για επιμελητή ή δημοσιογράφο το όνομα πρέπει να συμπληρώνεται αυτόματα από το σύστημα.
* Κατά την τροποποίηση ενός σχολίου πρέπει να γίνεται έλεγχος αν είναι σε κατάσταση δημιουργημένο.
* Κατά την αποδοχή ενός σχολίου το σύστημα θα πρέπει να αποθηκεύει την κατάσταση του που πρέπει να είναι εγκεκριμένο.
* Κατά την απόρριψη ενός σχολίου το σύστημα θα πρέπει να το διαγράφει από την βάση δεδομένων.
* Κατά την δημιουργία ενός θέματος το σύστημα θα πρέπει να αποθηκεύει αυτόματα το αναγνωριστικό του θέματος, καθώς και την κατάσταση του που πρέπει να είναι δημιουργημένο.
* Κατά την δημιουργία ενός θέματος θα πρέπει να δίνεται το όνομα του καθώς και αν υπάρχει το πατρικό του θέμα.
* Κατά την τροποποίηση ενός θέματος πρέπει να γίνεται έλεγχος αν είναι σε κατάσταση δημιουργημένο.
* Κατά την αποδοχή ενός θέματος το σύστημα θα πρέπει να αποθηκεύει την κατάσταση του που πρέπει να είναι εγκεκριμένο.
* Κατά την απόρριψη ενός θέματος το σύστημα θα πρέπει να το διαγράφει από την βάση δεδομένων.

## Β. Μη Λειτουργικές

Μη Λειτουργικές Απαιτήσεις είναι οι περιορισμοί ή οι απαιτήσεις που επιβάλλονται στο σύστημα. Καθορίζουν το χαρακτηριστικό ποιότητας του λογισμικού. Οι Μη Λειτουργικές Απαιτήσεις ασχολούνται με ζητήματα όπως η επεκτασιμότητα, η δυνατότητα συντήρησης, η απόδοση, η φορητότητα, η ασφάλεια, η αξιοπιστία και πολλά άλλα. Στην παρούσα εργασία και οι μη λειτουργικές απαιτήσεις είναι εμφανής από την εκφώνηση. Εμείς καταγράψαμε τις παρακάτω:

* Το σύστημα θα πρέπει να είναι ασφαλές από μη εξουσιοδοτημένη πρόσβαση.
* Το σύστημα θα πρέπει να ανταποκρίνεται γρήγορα.
* Το σύστημα θα πρέπει να είναι αξιόπιστο.
* Το σύστημα δεν θα πρέπει να έχει bugs.
* Το σύστημα θα πρέπει να παράγει κατάλληλα μηνύματα λάθους σε περιπτώσεις λανθασμένης χρήσης του από τους χρήστες του.
* Το σύστημα δεν θα πρέπει να επιτρέπει να διπλότυπα
* Το σύστημα θα πρέπει να επιτρέπει την πρόσβαση στις λειτουργίες του με κατάλληλους κανόνες πρόσβασης
* Το σύστημα θα πρέπει να ενημερώνει την βάση δεδομένων μόνο εφόσον η αντίστοιχη λειτουργία εκτελεστεί πλήρως και με σωστό τρόπο.

# Σχεδίαση

## Α. Αρχιτεκτονική & Περιβάλλον Συστήματος

### Component diagram

To Component diagram αναλύει το πραγματικό σύστημα υπό ανάπτυξη σε διάφορα υψηλά επίπεδα λειτουργικότητας. Κάθε στοιχείο είναι υπεύθυνο για έναν σαφή στόχο σε ολόκληρο το σύστημα και αλληλοεπιδρά μόνο με άλλα βασικά στοιχεία με βάση την ανάγκη γνώσης. Επομένως για το δικό μας σύστημα αυτά τα μέρη είναι η βάση δεδομένων, ο Tomcat server, το υποσύστημα για την διαχείριση των ειδήσεων και των σχολίων, καθώς και το υποσύστημα για την διαχείριση των θεμάτων.

Diagram

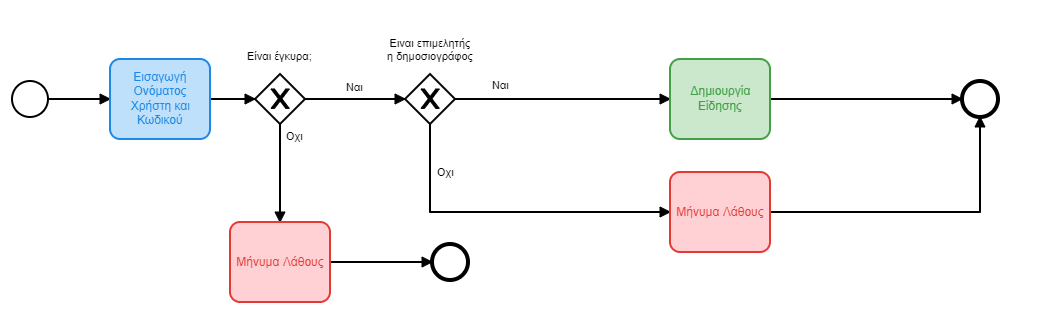
Description automatically generated

## Β. Επιχειρησιακές Διαδικασίες Συστήματος

### Process diagrams

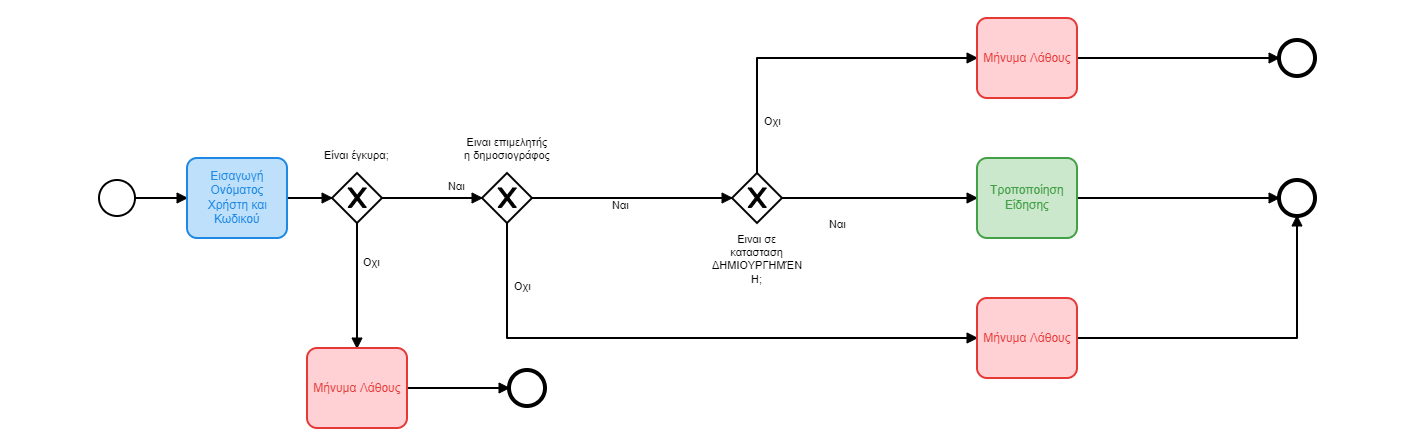
Ένα process diagram αποτελείται από δραστηριότητες, συμβάντα και πύλες, τις οποίες μια ροή ακολουθίας τοποθετεί σε μια ακολουθία ροής. Παρακάτω φαίνονται τα process diagrams ανάλογα με την ενέργεια που προσπαθεί να κάνει ένας χρήστης.

**Δημιουργία Είδησης**

****

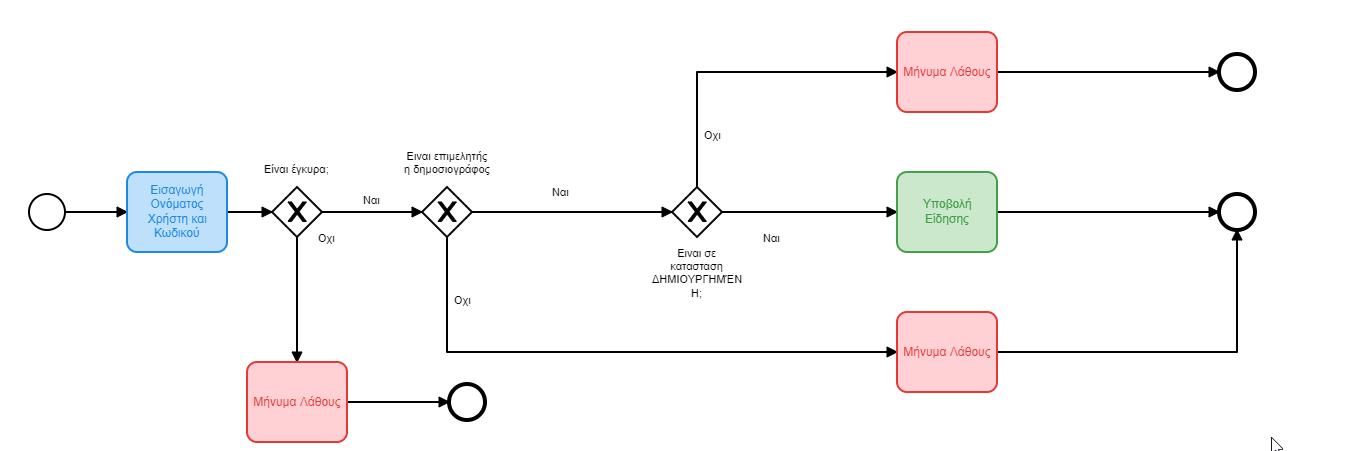
Όπως φαίνεται και παραπάνω για να γίνει η δημιουργία μιας είδησης που γίνεται μόνο από ταυτοποιημένους χρήστες (επιμελητή ή δημοσιογράφο) σύμφωνα με την εκφώνηση, το σύστημα ελέγχει την εγκυρότητα του ονόματος χρήστη και του κωδικού πρόσβασης και σε περίπτωση που κάτι από τα δύο είναι λάθος παράγει το κατάλληλο μήνυμα λάθους. Σε περίπτωση που είναι και τα δύο έγκυρα τότε το σύστημα κάνει έλεγχο στο ρόλο του χρήστη και αν πρόκειται για επιμελητή ή δημοσιογράφο προχωρά στην αποθήκευση μιας είδησης με δημιουργημένη κατάσταση, διαφορετικά παράγει το κατάλληλο μήνυμα λάθους.

**Τροποποίηση Είδησης**

****

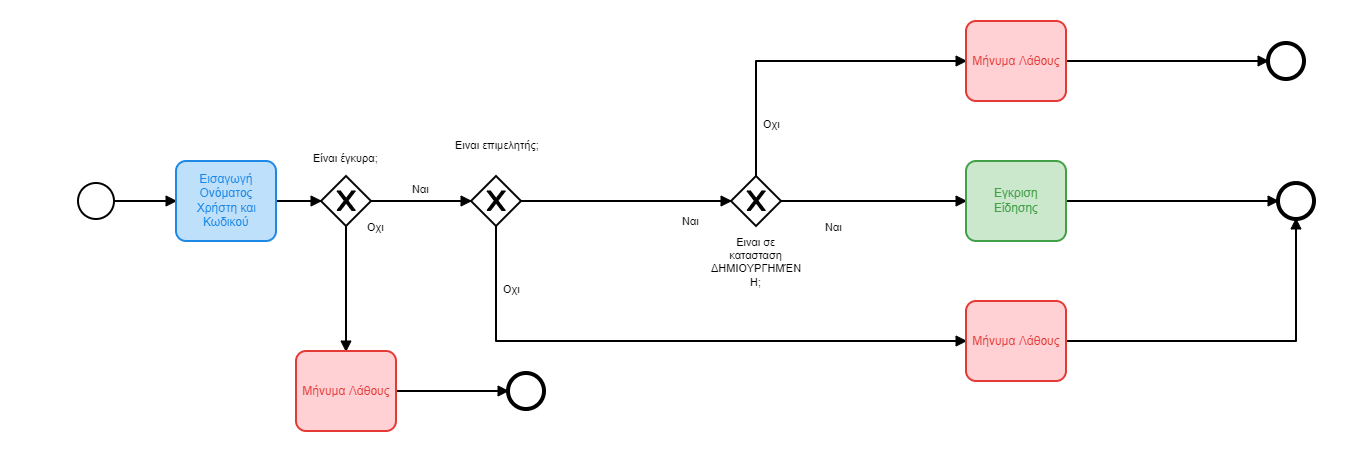
Όπως φαίνεται και παραπάνω για να γίνει η τροποποίηση μιας είδησης που γίνεται μόνο από ταυτοποιημένους χρήστες (επιμελητή ή δημοσιογράφο) σύμφωνα με την εκφώνηση, το σύστημα ελέγχει την εγκυρότητα του ονόματος χρήστη και του κωδικού πρόσβασης και σε περίπτωση που κάτι από τα δύο είναι λάθος παράγει το κατάλληλο μήνυμα λάθους. Σε περίπτωση που είναι και τα δύο έγκυρα τότε το σύστημα κάνει έλεγχο στο ρόλο του χρήστη και αν πρόκειται για επιμελητή ή δημοσιογράφο κάνει έλεγχο αν και η κατάσταση της είδησης είναι δημιουργημένη και αν είναι προχωράει στην τροποποίηση της είδησης, διαφορετικά και στις δυο περιπτώσεις παράγει το κατάλληλο μήνυμα λάθους.

**Υποβολή Είδησης**

****

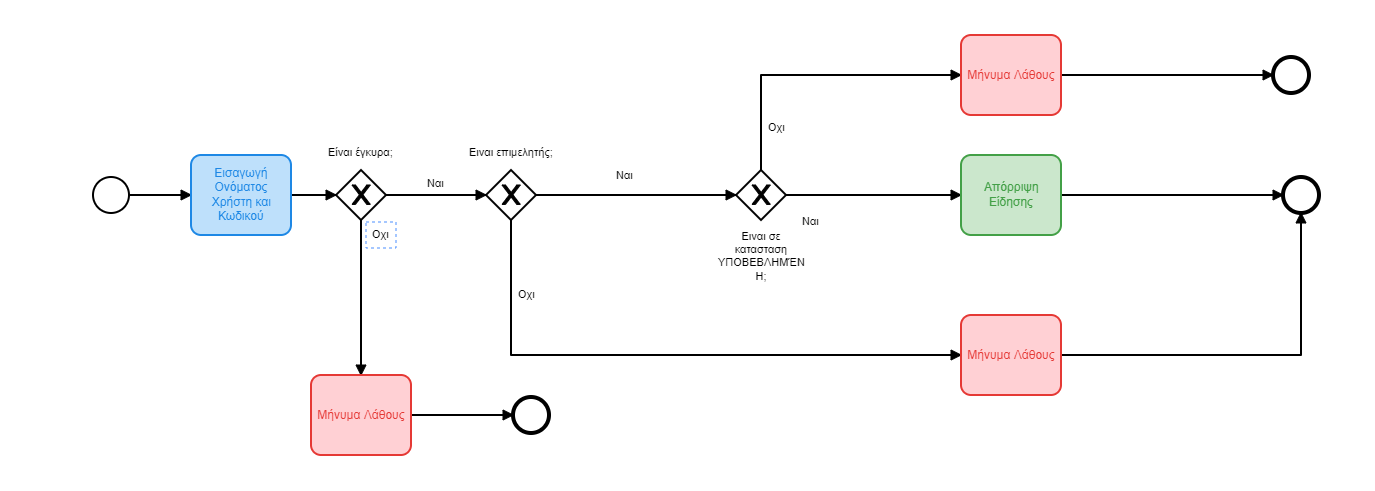
Όπως φαίνεται και παραπάνω για να γίνει η υποβολή μιας είδησης που γίνεται μόνο από ταυτοποιημένους χρήστες (επιμελητή ή δημοσιογράφο) σύμφωνα με την εκφώνηση, το σύστημα ελέγχει την εγκυρότητα του ονόματος χρήστη και του κωδικού πρόσβασης και σε περίπτωση που κάτι από τα δύο είναι λάθος παράγει το κατάλληλο μήνυμα λάθους. Σε περίπτωση που είναι και τα δύο έγκυρα τότε το σύστημα κάνει έλεγχο στο ρόλο του χρήστη και αν πρόκειται για επιμελητή ή δημοσιογράφο κάνει έλεγχο αν και η κατάσταση της είδησης είναι δημιουργημένη και αν είναι προχωράει στην υποβολή της είδησης, διαφορετικά και στις δυο περιπτώσεις παράγει το κατάλληλο μήνυμα λάθους.

**Έγκριση Είδησης**

****

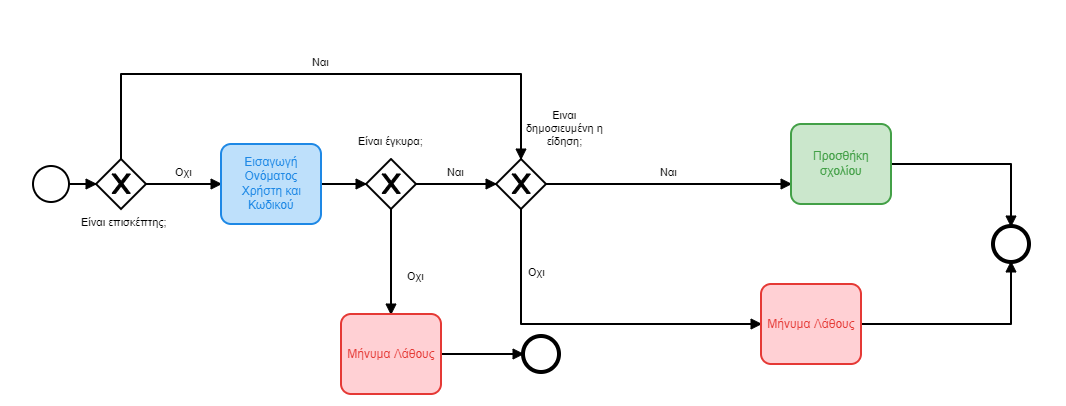
Όπως φαίνεται και παραπάνω για να γίνει η αποδοχή μιας είδησης που γίνεται μόνο από ταυτοποιημένους χρήστες (επιμελητή ή δημοσιογράφο) σύμφωνα με την εκφώνηση, το σύστημα ελέγχει την εγκυρότητα του ονόματος χρήστη και του κωδικού πρόσβασης και σε περίπτωση που κάτι από τα δύο είναι λάθος παράγει το κατάλληλο μήνυμα λάθους. Σε περίπτωση που είναι και τα δύο έγκυρα τότε το σύστημα κάνει έλεγχο στο ρόλο του χρήστη και αν πρόκειται για επιμελητή ή δημοσιογράφο κάνει έλεγχο αν και η κατάσταση της είδησης είναι δημιουργημένη και αν είναι προχωράει στην αποδοχή της είδησης, διαφορετικά και στις δυο περιπτώσεις παράγει το κατάλληλο μήνυμα λάθους.

**Απόρριψη Είδησης**

****

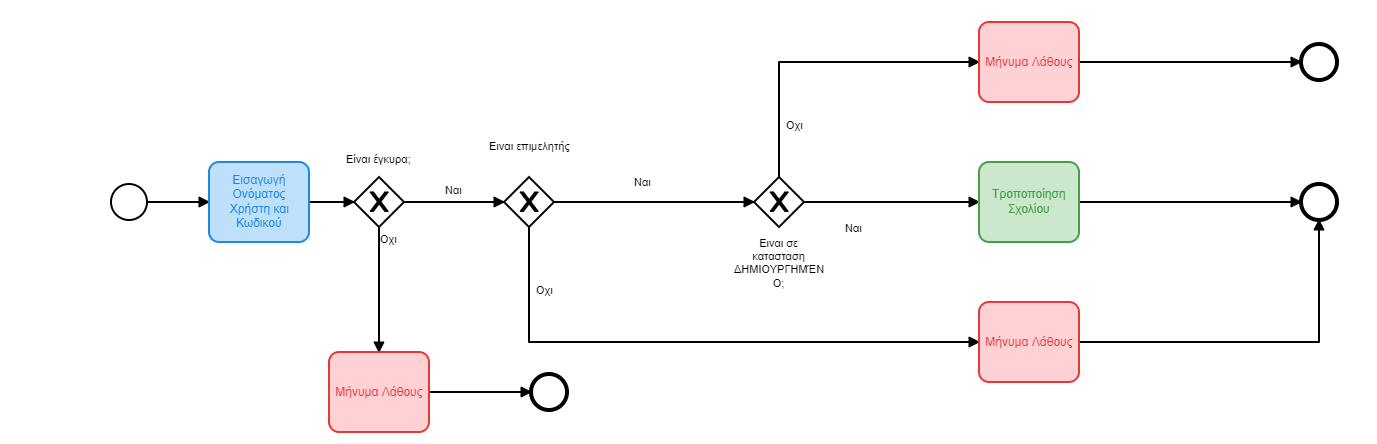
Όπως φαίνεται και παραπάνω για να γίνει η απόρριψη μιας είδησης που γίνεται μόνο από ταυτοποιημένους χρήστες (επιμελητή ή δημοσιογράφο) σύμφωνα με την εκφώνηση, το σύστημα ελέγχει την εγκυρότητα του ονόματος χρήστη και του κωδικού πρόσβασης και σε περίπτωση που κάτι από τα δύο είναι λάθος παράγει το κατάλληλο μήνυμα λάθους. Σε περίπτωση που είναι και τα δύο έγκυρα τότε το σύστημα κάνει έλεγχο στο ρόλο του χρήστη και αν πρόκειται για επιμελητή ή δημοσιογράφο κάνει έλεγχο αν και η κατάσταση της είδησης είναι δημιουργημένη και αν είναι προχωράει στην απόρριψη της είδησης, διαφορετικά και στις δυο περιπτώσεις παράγει το κατάλληλο μήνυμα λάθους.

**Προσθήκη σχολίου**

****

Όπως φαίνεται και παραπάνω για να γίνει η προσθήκη ενός σχολίου που μπορεί να γίνει είτε από επισκέπτη είτε από δημοσιογράφο είτε από επιμελητή, και για αυτό τον λόγο δίνεται η δυνατότητα να ξεκινήσει η διαδικασία και χωρίς την εισαγωγή ονόματος χρήστη και κωδικού πρόσβασης. Επομένως αν πρόκειται για επισκέπτη η ταυτοποίηση δεν είναι απαραίτητη, αλλιώς το σύστημα ελέγχει την εγκυρότητα του ονόματος χρήστη και του κωδικού πρόσβασης και σε περίπτωση που κάτι από τα δύο είναι λάθος παράγει το κατάλληλο μήνυμα λάθους. Σε περίπτωση που είναι και τα δύο έγκυρα τότε το σύστημα ελέγχει αν η σχετική είδηση είναι δημοσιευμένη και αν είναι αποθηκεύει ένα σχόλιο με κατάσταση δημιουργημένο, αλλιώς παράγει το κατάλληλο μήνυμα λάθους.

**Τροποποίηση Σχολίου**

****

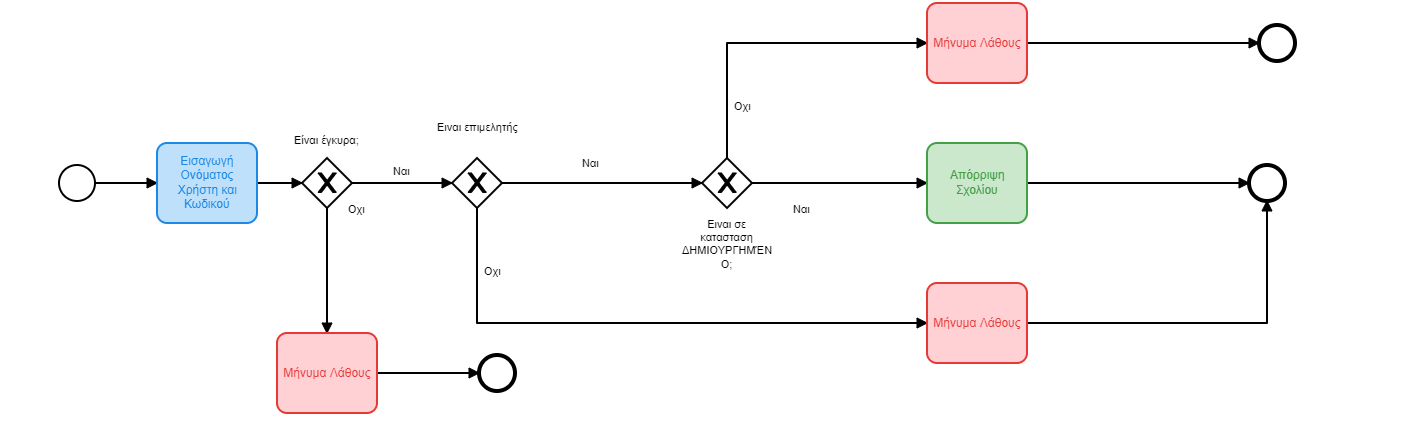
Όπως φαίνεται και παραπάνω για να γίνει η τροποποίηση ενός σχολίου που γίνεται μόνο από ταυτοποιημένους χρήστες (επιμελητή) σύμφωνα με την εκφώνηση, το σύστημα ελέγχει την εγκυρότητα του ονόματος χρήστη και του κωδικού πρόσβασης και σε περίπτωση που κάτι από τα δύο είναι λάθος παράγει το κατάλληλο μήνυμα λάθους. Σε περίπτωση που είναι και τα δύο έγκυρα τότε το σύστημα κάνει έλεγχο στο ρόλο του χρήστη και αν πρόκειται για επιμελητή κάνει έλεγχο αν και η κατάσταση του σχολίου είναι δημιουργημένη και αν είναι προχωράει στην τροποποίηση του σχολίου, διαφορετικά και στις δυο περιπτώσεις παράγει το κατάλληλο μήνυμα λάθους.

**Έγκριση Σχολίου**

**Diagram

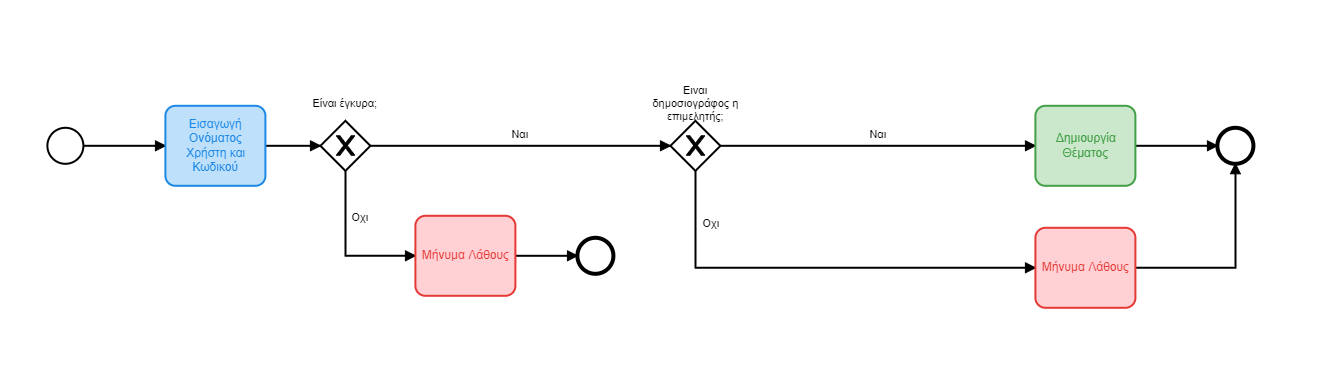
Description automatically generated**Όπως φαίνεται και παραπάνω για να γίνει η έγκριση ενός σχολίου που γίνεται μόνο από ταυτοποιημένους χρήστες (επιμελητή) σύμφωνα με την εκφώνηση, το σύστημα ελέγχει την εγκυρότητα του ονόματος χρήστη και του κωδικού πρόσβασης και σε περίπτωση που κάτι από τα δύο είναι λάθος παράγει το κατάλληλο μήνυμα λάθους. Σε περίπτωση που είναι και τα δύο έγκυρα τότε το σύστημα κάνει έλεγχο στο ρόλο του χρήστη και αν πρόκειται για επιμελητή κάνει έλεγχο αν και η κατάσταση του σχολίου είναι δημιουργημένη και αν είναι προχωράει στην έγκριση του σχολίου, διαφορετικά και στις δυο περιπτώσεις παράγει το κατάλληλο μήνυμα λάθους.

**Απόρριψη Σχολίου**

****

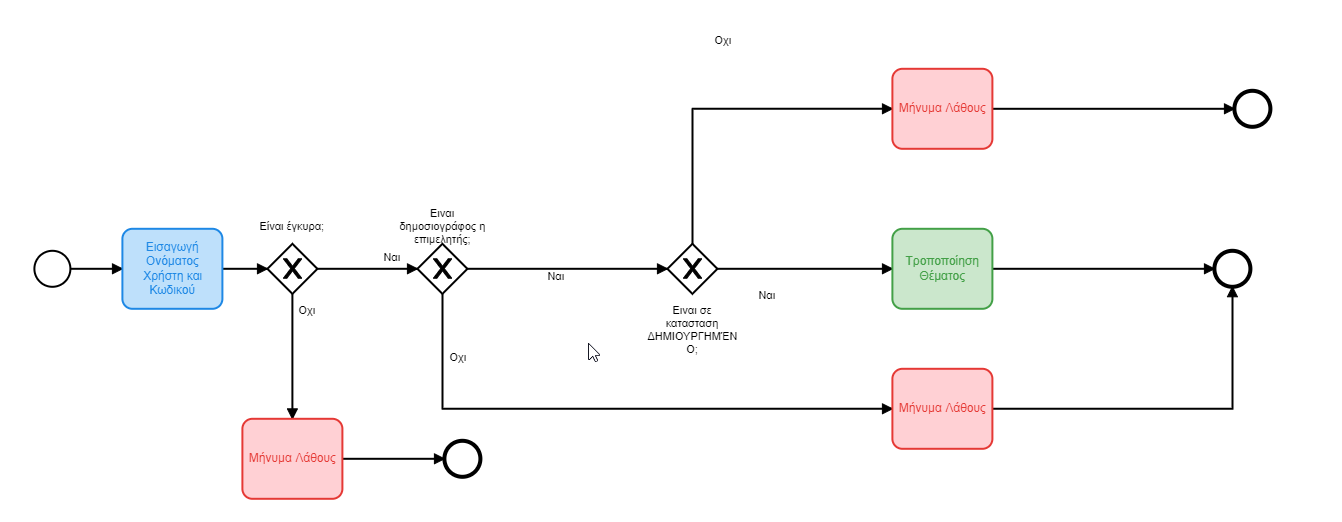
Όπως φαίνεται και παραπάνω για να γίνει η απόρριψη ενός σχολίου που γίνεται μόνο από ταυτοποιημένους χρήστες (επιμελητή) σύμφωνα με την εκφώνηση, το σύστημα ελέγχει την εγκυρότητα του ονόματος χρήστη και του κωδικού πρόσβασης και σε περίπτωση που κάτι από τα δύο είναι λάθος παράγει το κατάλληλο μήνυμα λάθους. Σε περίπτωση που είναι και τα δύο έγκυρα τότε το σύστημα κάνει έλεγχο στο ρόλο του χρήστη και αν πρόκειται για επιμελητή κάνει έλεγχο αν και η κατάσταση του σχολίου είναι δημιουργημένη και αν είναι προχωράει στην απόρριψη του σχολίου, διαφορετικά και στις δυο περιπτώσεις παράγει το κατάλληλο μήνυμα λάθους.

**Δημιουργία Θέματος**



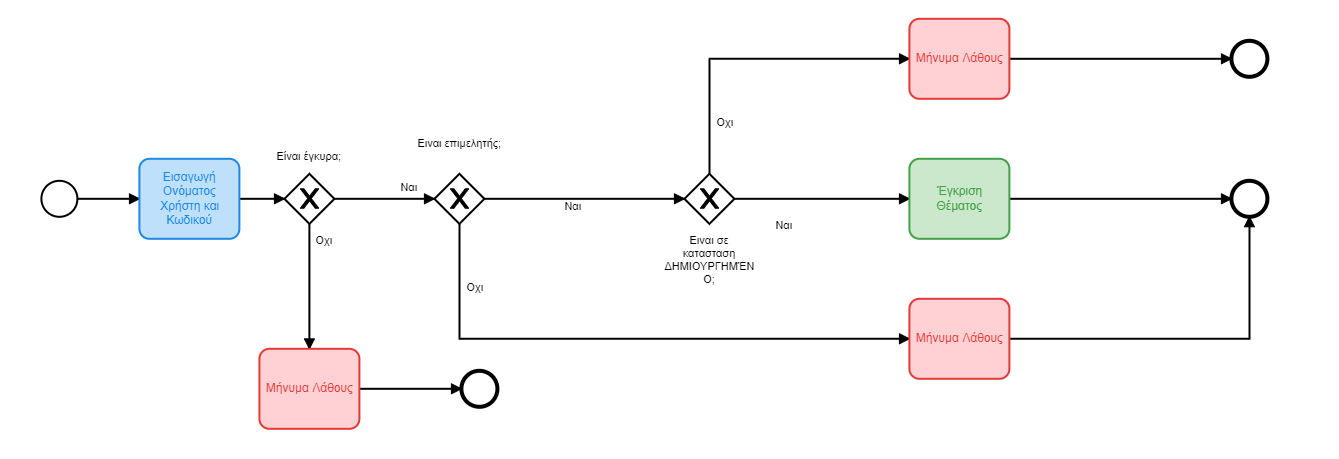
Όπως φαίνεται και παραπάνω για να γίνει η δημιουργία ενός θέματος που γίνεται μόνο από ταυτοποιημένους χρήστες (επιμελητή ή δημοσιογράφο) σύμφωνα με την εκφώνηση, το σύστημα ελέγχει την εγκυρότητα του ονόματος χρήστη και του κωδικού πρόσβασης και σε περίπτωση που κάτι από τα δύο είναι λάθος παράγει το κατάλληλο μήνυμα λάθους. Σε περίπτωση που είναι και τα δύο έγκυρα τότε το σύστημα κάνει έλεγχο στο ρόλο του χρήστη και αν πρόκειται για επιμελητή ή δημοσιογράφο προχωρά στην αποθήκευση ενός θέματος με δημιουργημένη κατάσταση, διαφορετικά παράγει το κατάλληλο μήνυμα λάθους.

**Τροποποίηση Θέματος**



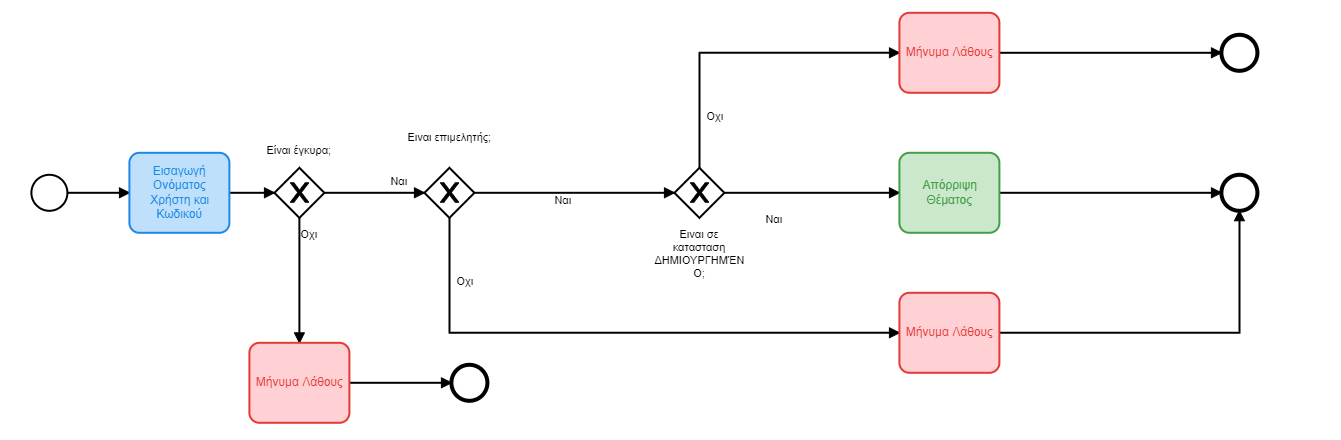
Όπως φαίνεται και παραπάνω για να γίνει η τροποποίηση ενός θέματος που γίνεται μόνο από ταυτοποιημένους χρήστες (επιμελητή ή δημοσιογράφο) σύμφωνα με την εκφώνηση, το σύστημα ελέγχει την εγκυρότητα του ονόματος χρήστη και του κωδικού πρόσβασης και σε περίπτωση που κάτι από τα δύο είναι λάθος παράγει το κατάλληλο μήνυμα λάθους. Σε περίπτωση που είναι και τα δύο έγκυρα τότε το σύστημα κάνει έλεγχο στο ρόλο του χρήστη και αν πρόκειται για επιμελητή ή δημοσιογράφο κάνει έλεγχο αν και η κατάσταση του θέματος είναι δημιουργημένη και αν είναι προχωράει στην τροποποίηση του θέματος, διαφορετικά και στις δυο περιπτώσεις παράγει το κατάλληλο μήνυμα λάθους.

**Έγκριση Θέματος**

****

Όπως φαίνεται και παραπάνω για να γίνει η έγκριση ενός θέματος που γίνεται μόνο από ταυτοποιημένους χρήστες (επιμελητή ή δημοσιογράφο) σύμφωνα με την εκφώνηση, το σύστημα ελέγχει την εγκυρότητα του ονόματος χρήστη και του κωδικού πρόσβασης και σε περίπτωση που κάτι από τα δύο είναι λάθος παράγει το κατάλληλο μήνυμα λάθους. Σε περίπτωση που είναι και τα δύο έγκυρα τότε το σύστημα κάνει έλεγχο στο ρόλο του χρήστη και αν πρόκειται για επιμελητή ή δημοσιογράφο κάνει έλεγχο αν και η κατάσταση του θέματος είναι δημιουργημένη και αν είναι προχωράει στην έγκριση του θέματος, διαφορετικά και στις δυο περιπτώσεις παράγει το κατάλληλο μήνυμα λάθους.

**Απόρριψη Θέματος**

****

Όπως φαίνεται και παραπάνω για να γίνει η απόρριψη ενός θέματος που γίνεται μόνο από ταυτοποιημένους χρήστες (επιμελητή ή δημοσιογράφο) σύμφωνα με την εκφώνηση, το σύστημα ελέγχει την εγκυρότητα του ονόματος χρήστη και του κωδικού πρόσβασης και σε περίπτωση που κάτι από τα δύο είναι λάθος παράγει το κατάλληλο μήνυμα λάθους. Σε περίπτωση που είναι και τα δύο έγκυρα τότε το σύστημα κάνει έλεγχο στο ρόλο του χρήστη και αν πρόκειται για επιμελητή ή δημοσιογράφο κάνει έλεγχο αν και η κατάσταση του θέματος είναι δημιουργημένη και αν είναι προχωράει στην απόρριψη του θέματος, διαφορετικά και στις δυο περιπτώσεις παράγει το κατάλληλο μήνυμα λάθους.

Γ. Οντότητες Συστήματος

### Class diagram

Το Class Diagram είναι βασικά μια γραφική αναπαράσταση της στατικής όψης του συστήματος και αντιπροσωπεύει διαφορετικές πτυχές της εφαρμογής. Στην ουσία δηλαδή είναι η μοντελοποίηση των βασικών οντοτήτων του συστήματος. Για το δικό μας σύστημα με βάση την εκφώνηση οι βασικές οντότητες όπως φαίνεται και παρακάτω είναι ο Χρήστης (User\_Entity) ,οι ειδήσεις ( News\_Entity) , τα σχόλια ( Comment\_Entity) και τα θέματα (Subject\_Entity)

Graphical user interface

Description automatically generated with low confidence

## Δ. Δεδομένα

### ERD

Το διάγραμμα οντοτήτων συσχετίσεων, εμφανίζει τη σχέση των συνόλων οντοτήτων που είναι αποθηκευμένα σε μια βάση δεδομένων. Με άλλα λόγια, τα διαγράμματα ER βοηθούν στην εξήγηση της λογικής δομής των βάσεων δεδομένων. Τα διαγράμματα ER δημιουργούνται με βάση τρεις βασικές έννοιες: οντότητες, χαρακτηριστικά και σχέσεις. Παρακάτω φαίνονται οι συσχετίσεις για το δικό μας σύστημα. Φαίνεται δηλαδή ότι μια είδηση σχετίζεται με τουλάχιστον ένα θέμα, και μπορεί να έχει από 0 έως n σχόλια. Επίσης φαίνεται ότι ένας χρήστης μπορεί να σχετίζεται με 0 έως n ειδήσεις ή θέματα αντίστοιχα.

A screenshot of a computer

Description automatically generated with medium confidence

# Υλοποίηση

## Α. URL αποθετηρίου GitHub

https://github.com/KotsarapoglouIasonas/icsd14092\_icsd17212.git

## Β. Τεκμηρίωση Υλοποίησης

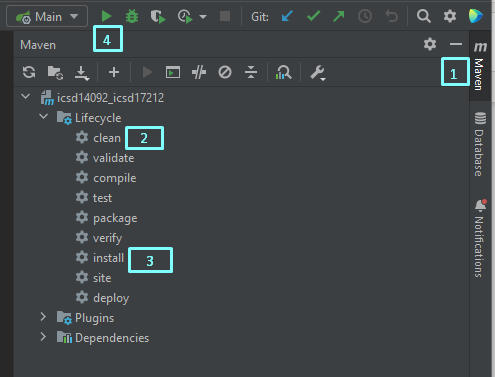
Το πρώτο βήμα είναι να έχουμε την εφαρμογή σε ένα IDE (π.χ. Intellij). Στην συνέχεια, ανοίγουμε το project από το version control κάνοντάς το clone από το github. Για να γίνει αυτό θα πρέπει να ακολουθηθούν τα παρακάτω βήματα:

Graphical user interface, text

Description automatically generated

Και εδώ βάζουμε το URL αποθετηρίου GitHub που βρίσκεται στην από πάνω ενότητα.

Το επόμενο βήμα είναι να τρέξουμε το παρακάτω:



Και τέλος να τρεξουμε με από την κλάση Main την main μεθοδό της επιλέγοντας το σχηματακι με το Play .

Για να δούμε τα διαθέσιμες λειτουργίες τις εφαρμογής πηγαίνουμε στο <http://localhost:8080/texnologia_logismikou/swagger-ui.html>.

Για την υποκείμενη βάση δεδομένων χρησιμοποιήσαμε μια open source την postgres συγκεκριμένα την 13,9.Για την διαχείριση της βάσης δεδομένων χρησιμοποιήσαμε το pgadmin , το οποίο το εγκαταστήσαμε από εδώ <https://www.pgadmin.org/download/pgadmin-4-windows/>. Αφού έχει γίνει πρώτα η εγκατάσταση της postgres το pgadmin θα την βρει αυτόματα. Για να συνδεθούμε βάλαμε απλά τα credentials που ορίσαμε στην postgres και στην συνέχεια δημιουργήσαμε μια βάση με όνομα texnologia\_logismikou.

# Συμπεράσματα

Με την υλοποίηση αυτής της εργασίας αναπτύχθηκε μια RESTful εφαρμογή με διάφορες λειτουργίες με ένα framework που όλο ακούγαμε αλλά κανένας από τους δύο μας δεν είχε ασχοληθεί το Spring boot. Επίσης ήρθαμε σε επαφή με ένα maven project και μάθαμε τις βασικές του λειτουργίες. Τα προβλήματα που αντιμετωπίσαμε είχαν να κάνουν και με την λειτουργικότητα του καινούργιου framework αλλά και με το να κατανοήσουμε πλήρως τις λειτουργίες της εφαρμογής. Δεν ήταν λίγες οι φορές που κατά την διάρκεια του manual testing βρίσκαμε bugs ή σκεφτόμασταν και κάτι ακόμα που δεν το πληρούσε η εφαρμογή. Τέλος όσο αναφορά τα best practices δώσαμε όσο πιο σαφή ονόματα μπορούσαμε στις μεθόδους στις κλασεις κ.λπ. Αυτό που θα αλλάζαμε σε επόμενη εργασία είναι να διαχωρίσουμε τα apis ανάλογα με τις λειτουργίες που έχουν και όχι να έχουμε ένα γενικό Controller με όλα τα endpoints εκεί.