D F M

ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΔΗΛΩΣΕΩΝ ΦΟΡΟΥ ΜΕΤΑΒΙΒΑΣΗΣ

Συνεργατική Υλοποίηση

Αθανάσιος Τάχος

Κώστας Κώστου

Πέτρος Πέτρου

Αθήνα, 20/01/2022

# **Εισαγωγή**

Το DFM είναι ένα πληροφοριακό σύστημα διαχείρισης δηλώσεων φόρου μεταβίβασης. Δημιουργήθηκε με γνώμονα την απλότητα και την ευχρηστία χωρίς να γίνουν «εκπτώσεις» σε θέματα ασφάλειας και αποτελεσματικότητας.

# **Περιεχόμενα**

Table of Contents

[**Εισαγωγή** 1](#_Toc125145875)

[**Περιεχόμενα** 1](#_Toc125145876)

[**Χρήστες και Ρόλοι** 2](#_Toc125145877)

[**Σενάριο Εργασίας** 3](#_Toc125145878)

[**Υλοποίηση** 5](#_Toc125145879)

# **Χρήστες και Ρόλοι**

Το σύστημα είναι προσβάσιμο μόνο σε πιστοποιημένους χρήστες καθώς οι πληροφορίες που διαχειρίζεται είναι ευαίσθητες. Για το λόγο αυτό προσδιορίστηκαν οι παρακάτω ρόλοι:

* Διαχειριστής (Administrator)
* Συμβολαιογράφος (Notary)
* Πωλητής (Seller)
* Αγοραστής (Buyer)

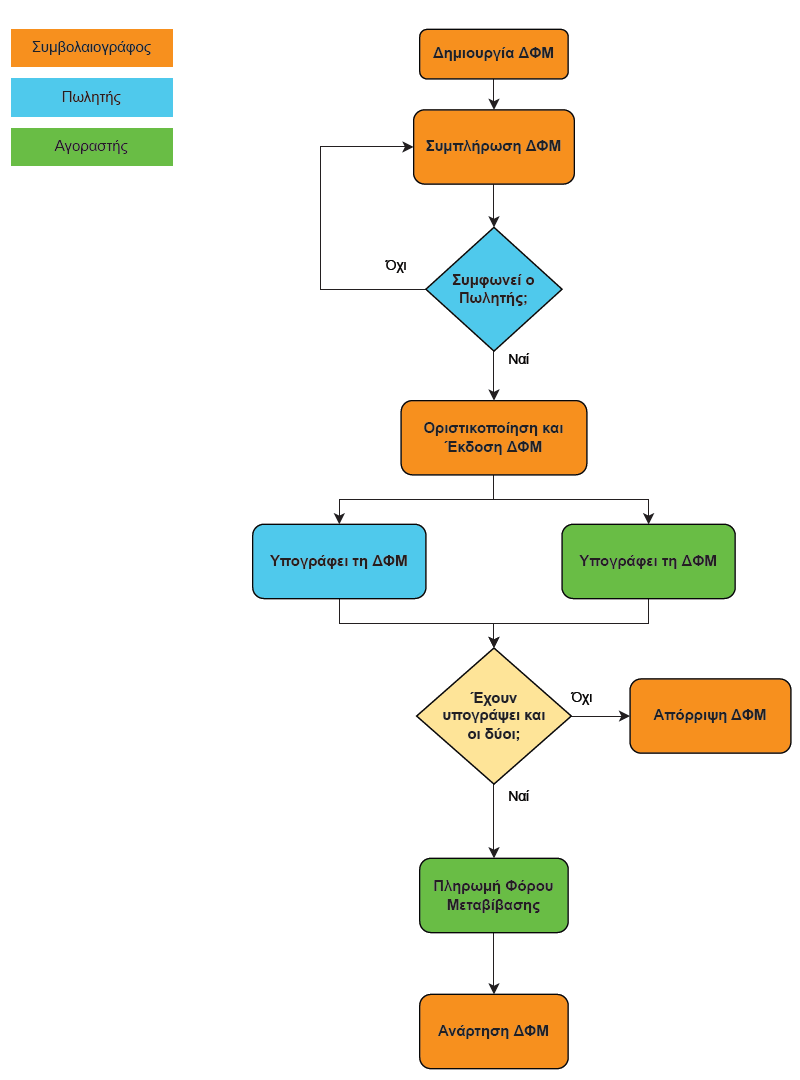
Κάθε ρόλος φέρει και διαφορετικά δικαιώματα που βάση αυτών δημιουργούνται λειτουργικές περιοχές πρόσβασης. Συγκεκριμένα τα δικαιώματα ανά ρόλο:

* Διαχειριστής
  + Μπορεί να δει και να επεξεργαστεί οποιαδήποτε πληροφορία του συστήματος.
* Συμβολαιογράφος
  + Μπορεί να δει τις πληροφορίες των πωλητών, των αγοραστών και των ακινήτων χωρίς να μπορεί να τις επεξεργαστεί.
  + Μπορεί να δει και να επεξεργαστεί τις πληροφορίες των δηλώσεων
* Πωλητής
  + Μπορεί να δει τις πληροφορίες των συμβολαιογράφων, των αγοραστών και των δηλώσεων που συσχετίζονται με αυτόν.
  + Μπορεί να δει και να επεξεργαστεί τις πληροφορίες των ακινήτων.
* Αγοραστής
  + Μπορεί να δει τις πληροφορίες των συμβολαιογράφων, των πωλητών, των ακινήτων και των δηλώσεων που συσχετίζονται με αυτόν.

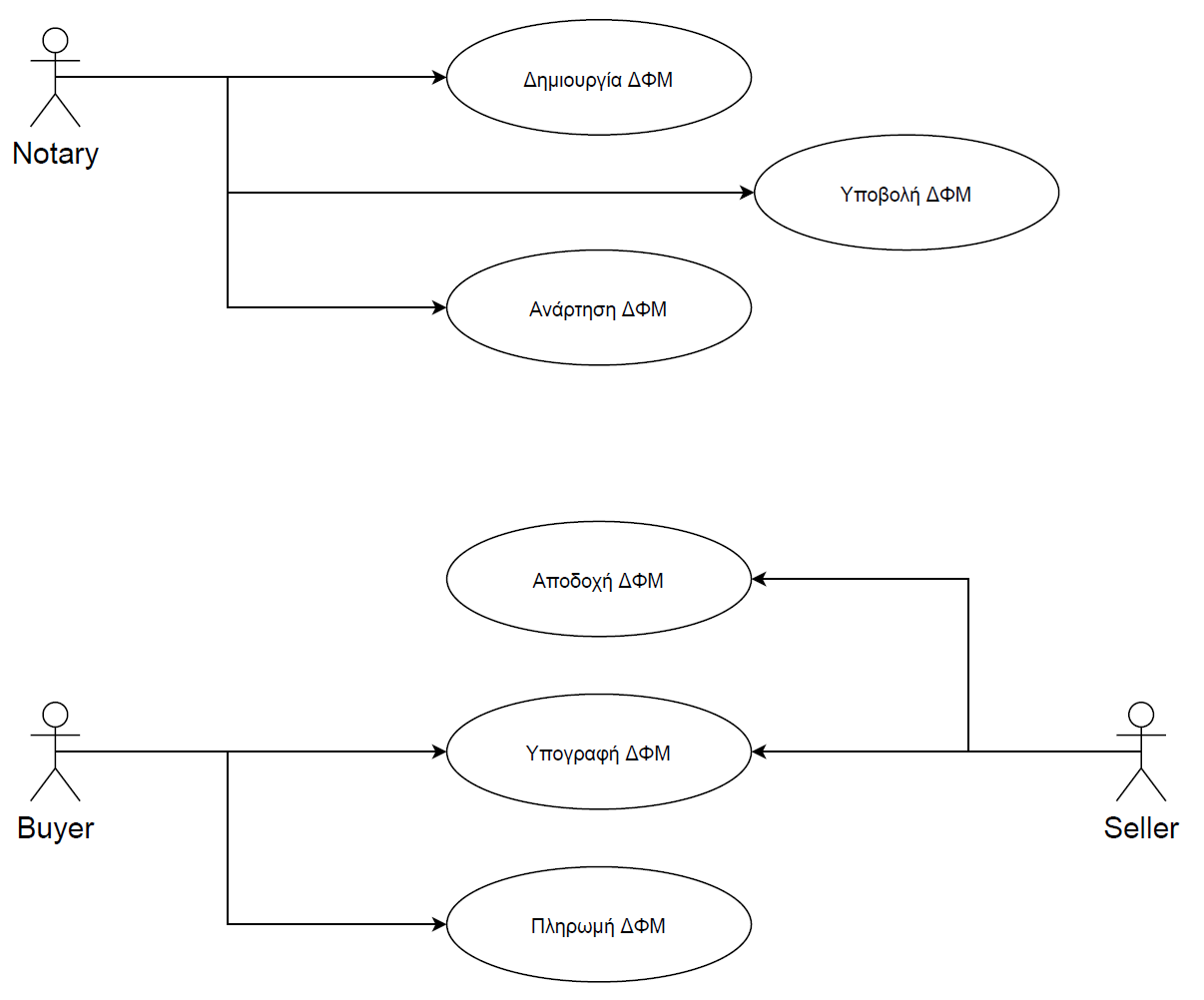
Κατά την πρώτη είσοδο του χρήστη στο σύστημα προσδιορίζεται και ο ρόλος που επιθυμεί να έχει. Δεν μπορεί να γίνει επιλογή ρόλου Διαχειριστή.

# **Σενάριο Εργασίας**

Το βασικό σενάριο εργασίας είναι η σύναψη της δήλωσης φόρου μεταβίβασης. Για την διεκπεραίωση του εν λόγο σεναρίου αναπτύχθηκε η παρακάτω ροή εργασίας:



Οι ενέργειες που πρέπει να ακολουθήσει ο κάθε χρήστης για την διεκπεραίωση του πρωθύστερου σεναρίου αναλύονται στο παρακάτω διάγραμμα:



# **Υλοποίηση**

Για την ανάπτυξη του DFM χρησιμοποιήθηκε Java έκδοση 19.0.1 κάνοντας χρήση του Spring Boot Framework με τα παρακάτω εργαλείων:

* mysql-connector-java

Βιβλιοθήκη πρόσβασης στη Βάση Δεδομένων

* spring-boot-starter-data-jpa

Μέσο επικοινωνίας με τη Βάση Δεδομένων χρήση του Hibernate ORM

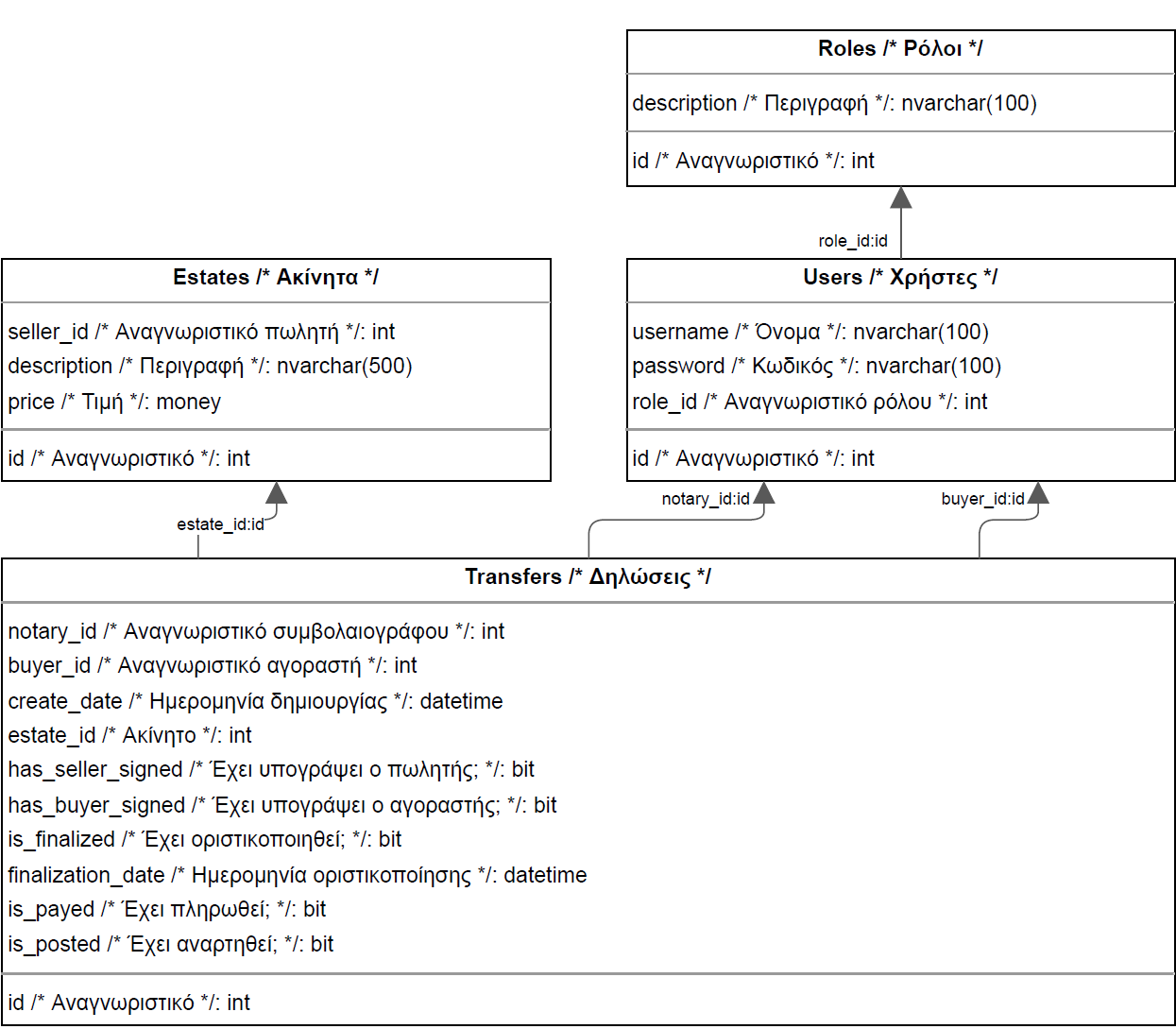
* spring-boot-starter-security

Μηχανισμός διαχείρισης χρηστών και ρόλων

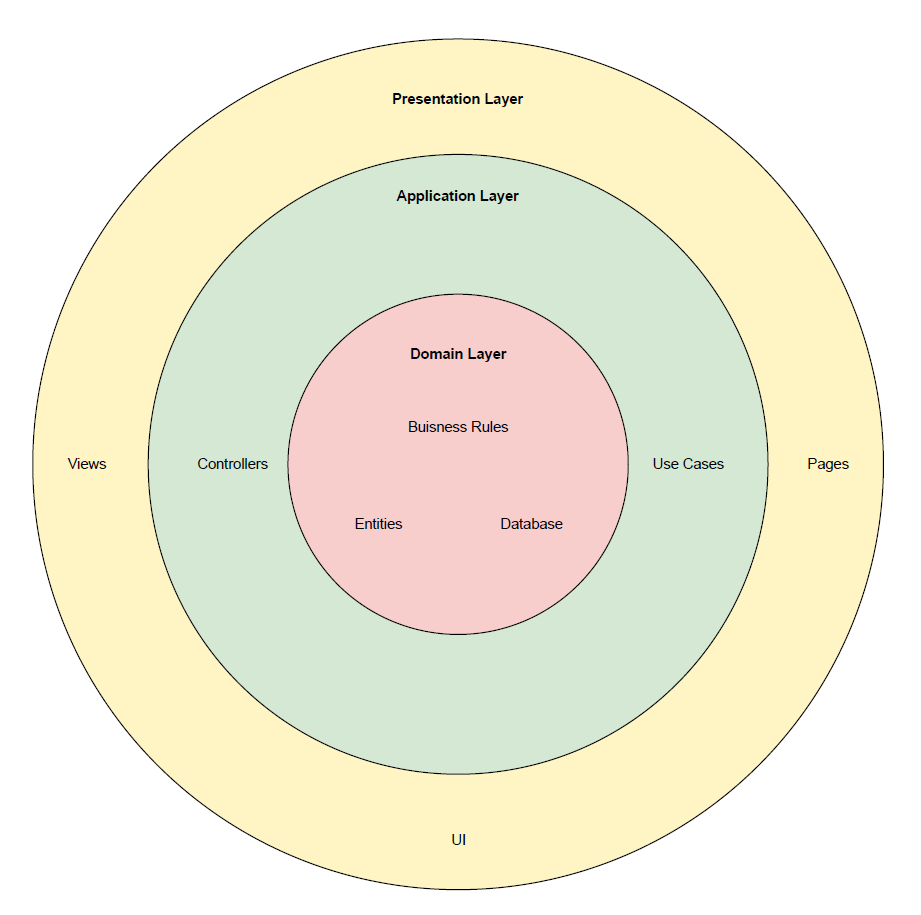
* spring-boot-starter-web

Τρόπος δημιουργίας διαδικτυακής εφαρμογής

Η Βάση Δεδομένων που επιλέχθηκε είναι η MySQL. Η υλοποίηση έγινε κάνοντας χρήση ενός σχήματος (scheme) με τους απαραίτητους πίνακες για τη διατήρηση των δεδομένων. Ενδεικτικά παρατίθεται το σχήμα:



Η αρχιτεκτονική που στήθηκε η εφαρμογή είναι βασισμένη σε επίπεδα (layers) και λέγεται Clean Architecture. Το κάθε επίπεδο έχει διακριτές δυνατότητες και πρόσβαση σε λειτουργίες.



Πέρα από την Clean Architecture χρησιμοποιήθηκε και το Model View Controller Design Pattern.

Με βάση τα παραπάνω ο κώδικας που αφορά το κάθε επίπεδο διαχωρίζεται σε:

* Presentation Layer
  + Διαθέτει τον κώδικα που τρέχει στον πελάτη (client) δηλαδή στο πρόγραμμα περιήγησης (browser). Πιο συγκεκριμένα είναι το User Interface που παρέχεται μέσα από τις Pages που φορτώνουν Views για την εμφάνιση αυτού.
* Application Layer
  + Έρχεται σε επαφή με το Presentation Layer μέσω των Endpoints που ορίζονται στους Controllers. Οι Controllers «ακούν» σε request που γίνονται από τον client και επιστρέφουν πίσω τα απαραίτητα δεδομένα που χρειάζονται τα Views για να «ζωγραφίσουν» το Interface.
  + Επίσης, έρχεται σε επαφή και με το Domain Layer για την άντληση των δεδομένων μέσω των Repositories που καλούν τα Services. Τα Services έχουν τα Use Cases.
* Domain Layer
  + Έχει την σύνδεση με τη Βάση Δεδομένων μέσω των Repositories, τα οποία έχουν όλες τις απαραίτητες μεθόδους για την άντληση και επεξεργασία των δεδομένων. Επίσης, φέρουν και όλα τα Business Rules που πρέπει το σύστημα και τα δεδομένα να «υπακούν» προστατεύοντας με αυτόν τον τρόπο την ακεραιότητα και την κανονικότητα των δεδομένων.
  + Για την επίτευξη των παραπάνω υπάρχουν τα μοντέλα (models) των πινάκων της Βάσης Δεδομένων που τα Repositories φορτώνουν μέσα τους τα δεδομένα.