

Σχεδιασμός Συστήματος Δήλωσης Φόρου Μεταβίβασης Ακινήτου

Εργασία Κατανεμημένων Συστημάτων

Γιοργκέν Τσάνι, 22105

Άλκης Γεωργής, 22015

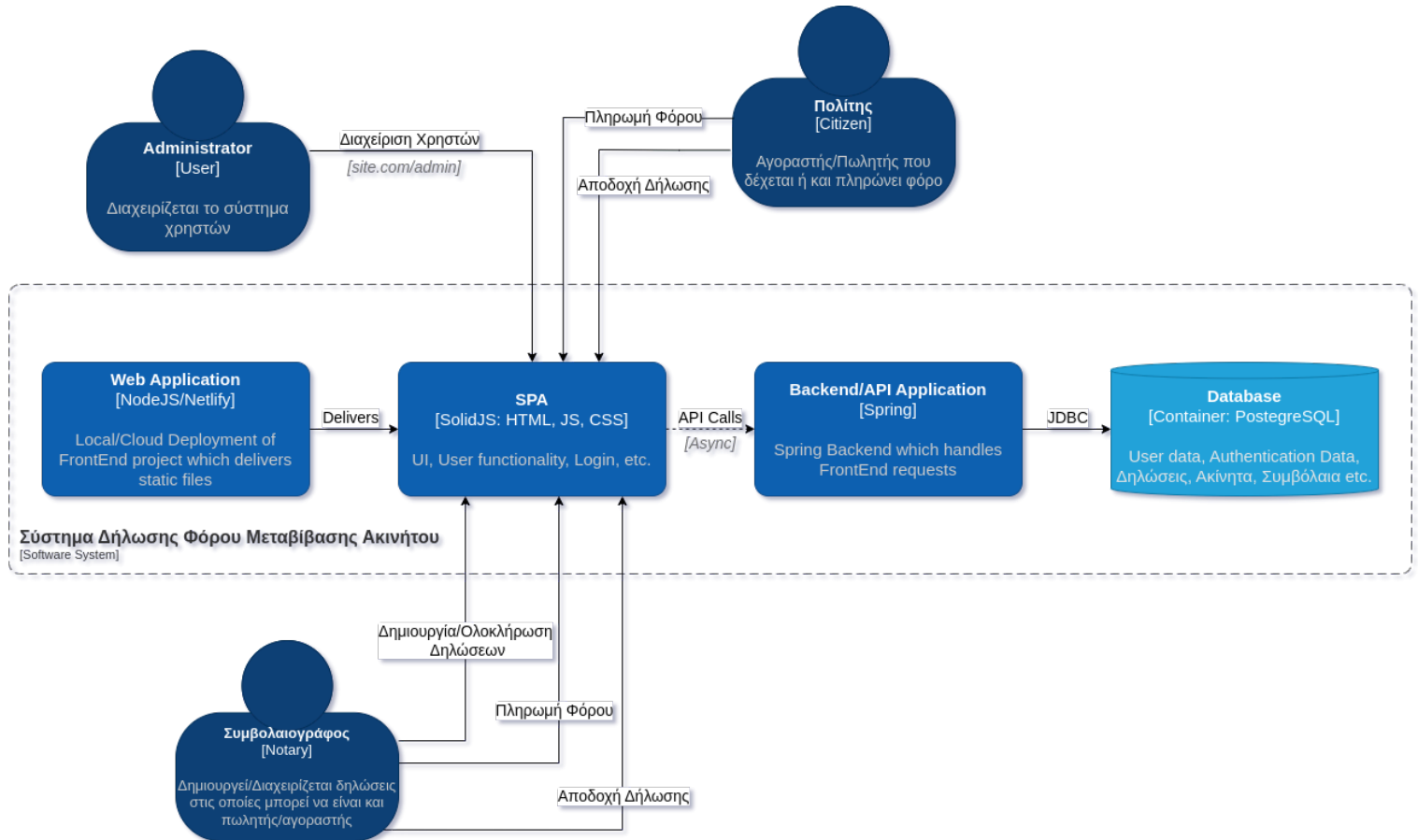
Χριστίνα Σκαρμούτσου, 22096

Ομάδα 335 στο e-class

| | |
|-----------------------------|-----------|
| Γενική Αρχιτεκτονική | 3 |
| UML Διαγράμματα | 5 |
| Class Diagram | 5 |
| UML Sequence Diagram | 6 |
| UML Use Case Diagram | 8 |
| Παραδοχές | 9 |
| Repositories | 10 |

Γενική Αρχιτεκτονική

Με βάση το [C4 model](#) παρουσιάζουμε την αρχιτεκτονική του συστήματος στο C1 επίπεδο.



Για την αποφυγή των πολλαπλών ομοίων βελών από κάθε χρήστη προς το Web Application, δείχνουμε μέσω του General User ότι κάθε χρήστης περνάει από έναν server ο οποίος επιστρέφει ένα Single Page Application που θα υλοποιηθεί με την βοήθεια του [SolidJS](#). Για το Development phase θα τρέξει ο Web Server σε nodejs locally, αλλά για την παρουσίαση της εργασίας έχει γίνει deployment στο Netlify.

Θα υπάρχει και ένα ειδικό route για τους administrators του συστήματος το οποίο θα υλοποιηθεί και αυτό μέσω του SolidJS για να επικρατεί ομοιογένεια στο interface.

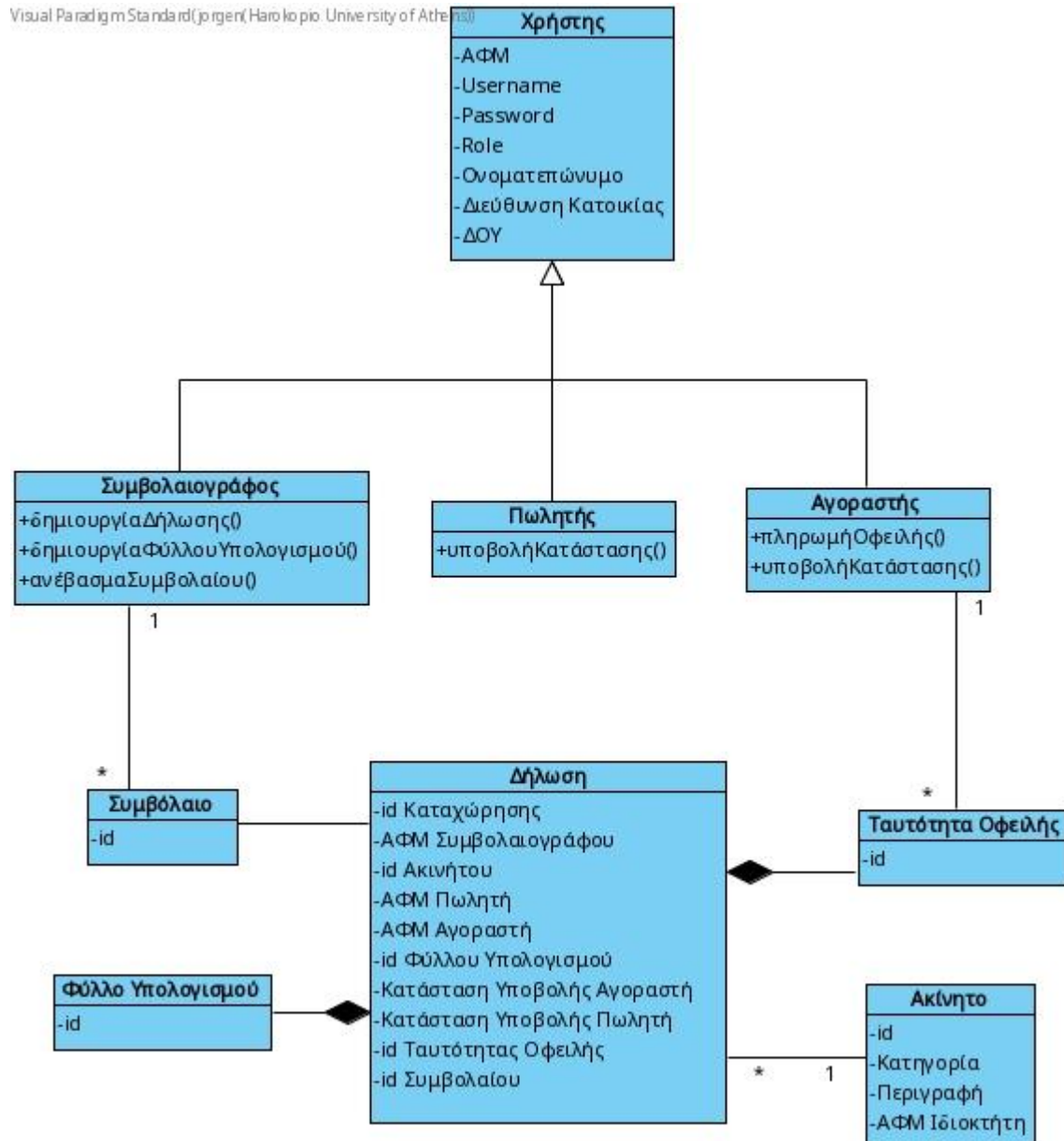
API application θα είναι το **backend** που θα υλοποιηθεί μέσω Spring, όπως και ζητείται στην εργασία.

Για την βάση δεδομένων αναφέρουμε ότι θα χρησιμοποιηθεί PostgreSQL.

Οι χρήστες θα αυθεντικοποιούνται με χρήση JWT.

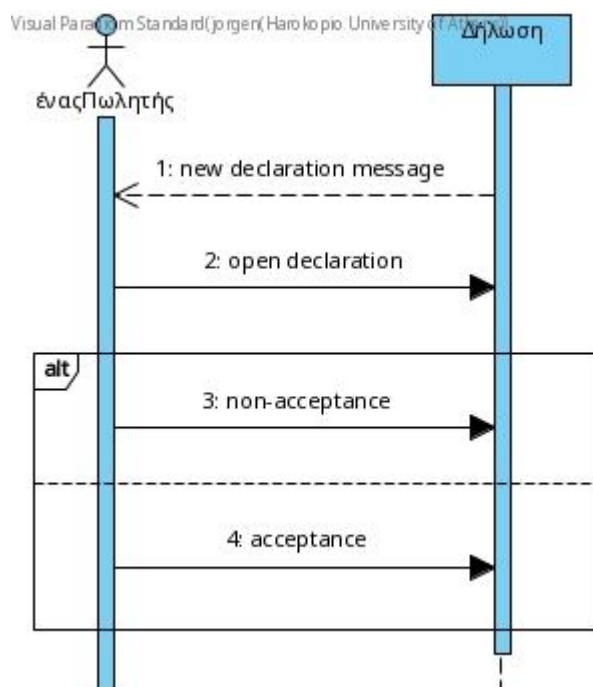
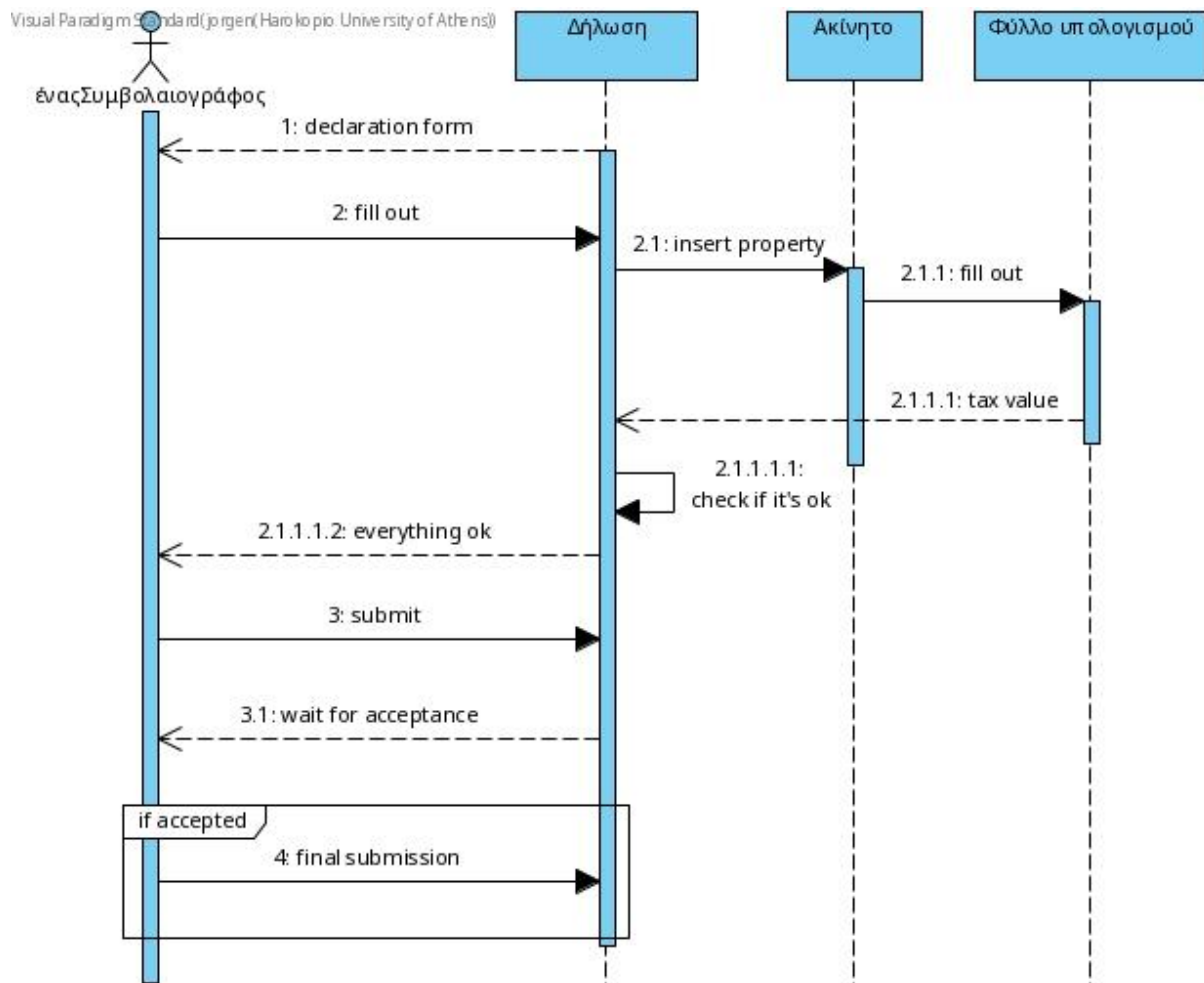
UML Διαγράμματα

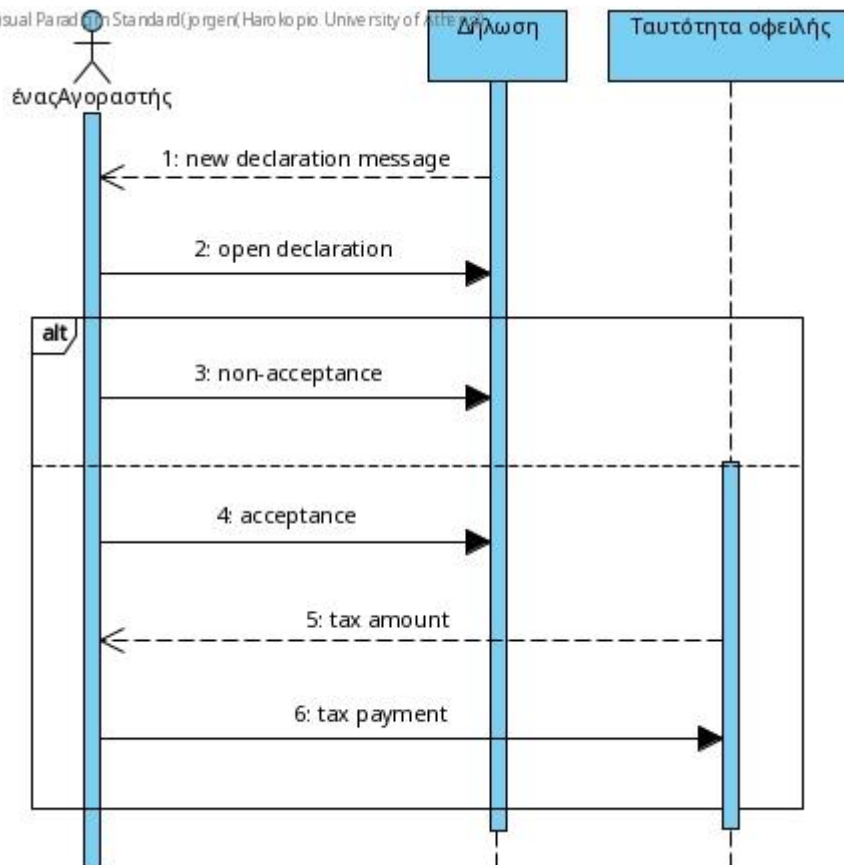
Class Diagram



Πιθανότητα να υπάρξουν αλλαγές όσο θα συνεχιστεί το development της εργασίας, για κάθε τυχόν αλλαγές θα υπάρχει changelog.

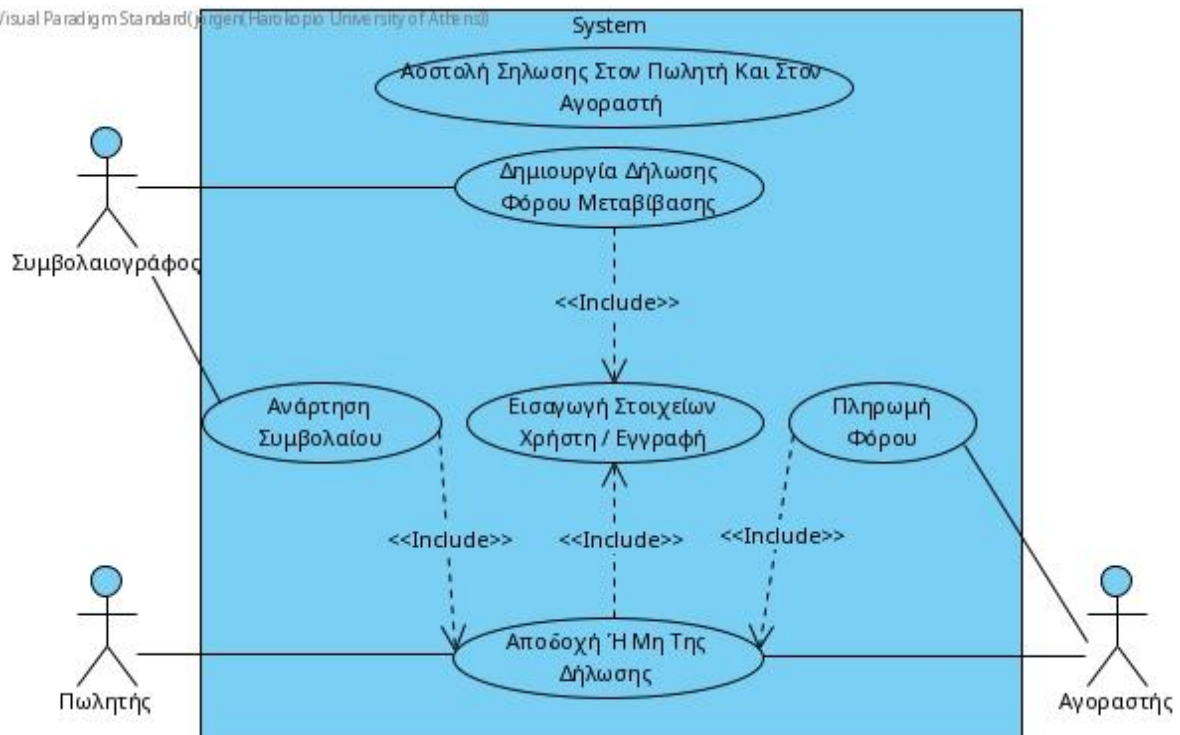
UML Sequence Diagram





UML Use Case Diagram

Visual Paradigm Standard (Katerini Haralabopoulou, University of Athens)



Παραδοχές

1. Ο συμβολαιογράφος όταν θα συμπληρώνει την δήλωση που δημιουργεί, θα δίνει τα στοιχεία του ακινήτου και αυτόματα στο σύστημα, κατά την υποβολή στο backend, θα δημιουργεί ένα entry σε έναν πίνακα Ακινήτων στην βάση, η οποία θα μπορεί να επαναχρησιμοποιηθεί.
2. Θα υπάρχει σύστημα υπολογισμού του φόρου το οποίο πιθανότατα να μην συμπίπτει με την πραγματικότητα, καθώς κανείς στην ομάδα δεν είναι οικείος με την διαδικασία αυτή.
3. Θεωρούμε ότι η πληρωμή “γίνεται” εκτός από το σύστημα μας. Εμφανίζεται μόνο το ποσό του φόρου σε κάθε δήλωση και όχι αν έγινε πληρωμή του.
4. Ο συμβολαιογράφος, όταν θα χρειαστεί να ανεβάσει το συμβόλαιο που έχει φτιάξει, θα ανεβάζει τον κωδικό του συμβολαίου που “υπάρχει” σε κάποιο άλλο σύστημα.
5. Οι χρήστες θα περνάνε από διαδικασία Account Sign Up για να συμπληρώσουν τα στοιχεία τους.
6. Οι συμβολαιογράφοι μπορούν να συμμετάσχουν **και** ως αγοραστές ή πωλητές κάποιου ακινήτου.
7. Το backend και η βάση είναι deployed στο Google Cloud, μέσω των App Engine και SQL υπηρεσιών αντίστοιχα. Το frontend είναι deployed στο Netlify.
8. Σελίδες/Routing: Εάν κάποιος χρήστης προσπαθήσει να κάνει access τις σελίδες /admin, /account/new, /declarations χωρίς να έχει κάνει login, θα γίνεται αυτόματα redirected στο homepage. Τα routes κάνουνε simulate ένα traditional page, όμως είναι ακόμη ένα SPA
9. Session Storage και JWT: Ελέγχουμε εάν οι χρήστες έχουν τα δικαιώματα που χρειάζονται για να δουν κάποια σελίδα μέσω JWT που δίνεται από το Backend και αποθηκεύεται στο sessionStorage του browser
10. Η επιλογή του sessionStorage ήταν για να διευκολύνει το testing κατά το development και δεν το αλλάξαμε, αλλά μπορεί εύκολα να γίνει αυτή η αλλαγή, καθώς δεν θα ήταν σωστό να λειτουργήσει η εφαρμογή με δεδομένα μόνο στο sessionStorage.

Repositories

Specs: <https://github.com/jgenc/hua-distributed-project-specs>

Backend: <https://github.com/jgenc/hua-distributed-project-backend>

Frontend: <https://github.com/jgenc/hua-distributed-project-frontend>