

Σχεδιασμός Συστήματος Δήλωσης Φόρου Μεταβίβασης Ακινήτου

Εργασία Κατανεμημένων Συστημάτων

Γιοργκέν Τσάνι, 22105

Άλκης Γεωργής, 22015

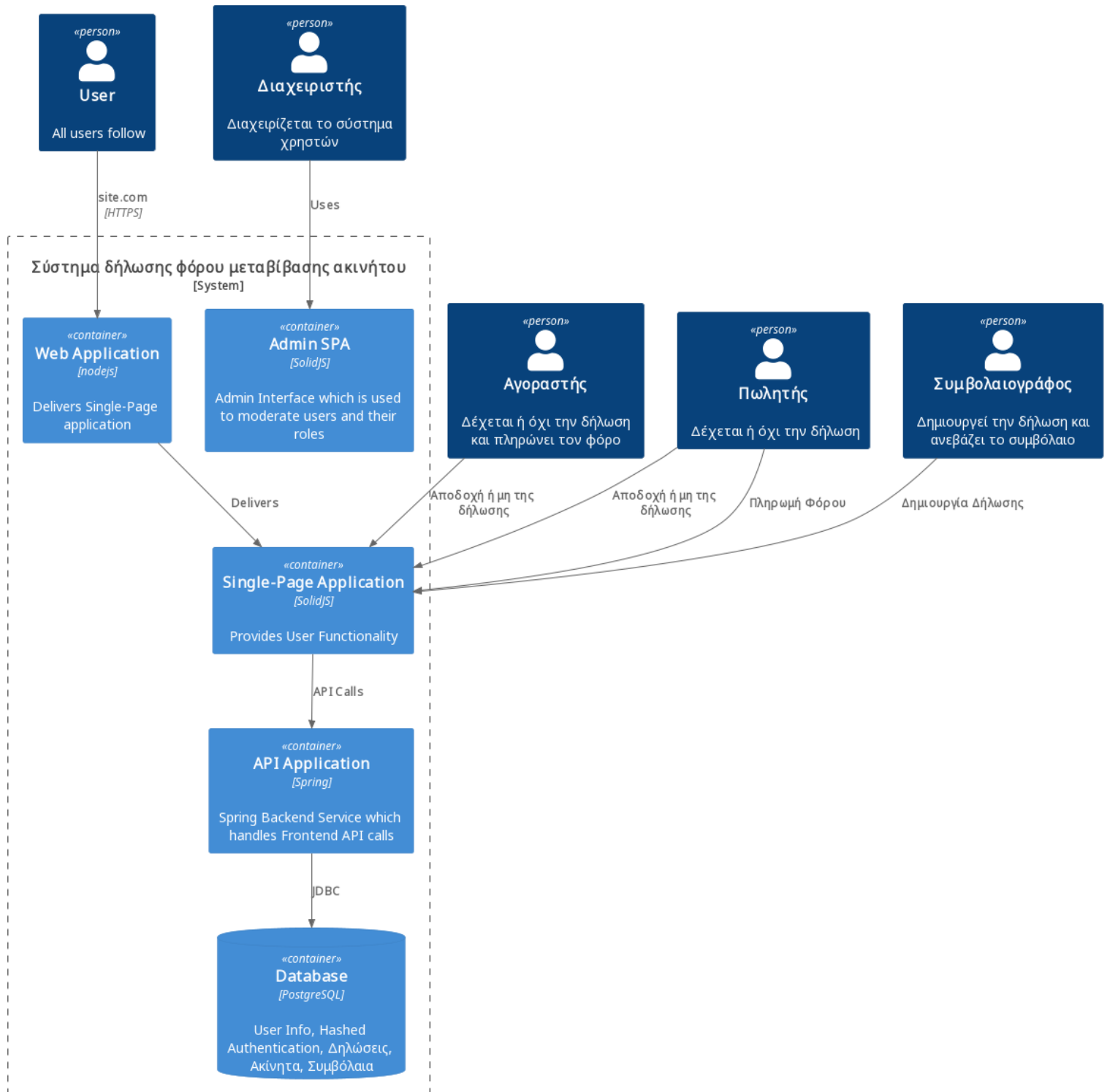
Χριστίνα Σκαρμούτσου, 22096

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](#) αλλά και του [SolidStart](#). Για το Development phase θα τρέξει ο Web Server σε nodejs locally, αλλά πιθανότατα για την παρουσίαση της εργασίας θα το ανεβάσουμε μέσω Netlify, Vercel ή κάποιον άλλον provider που υποστηρίζει το deployment ενός SolidJS project.

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.

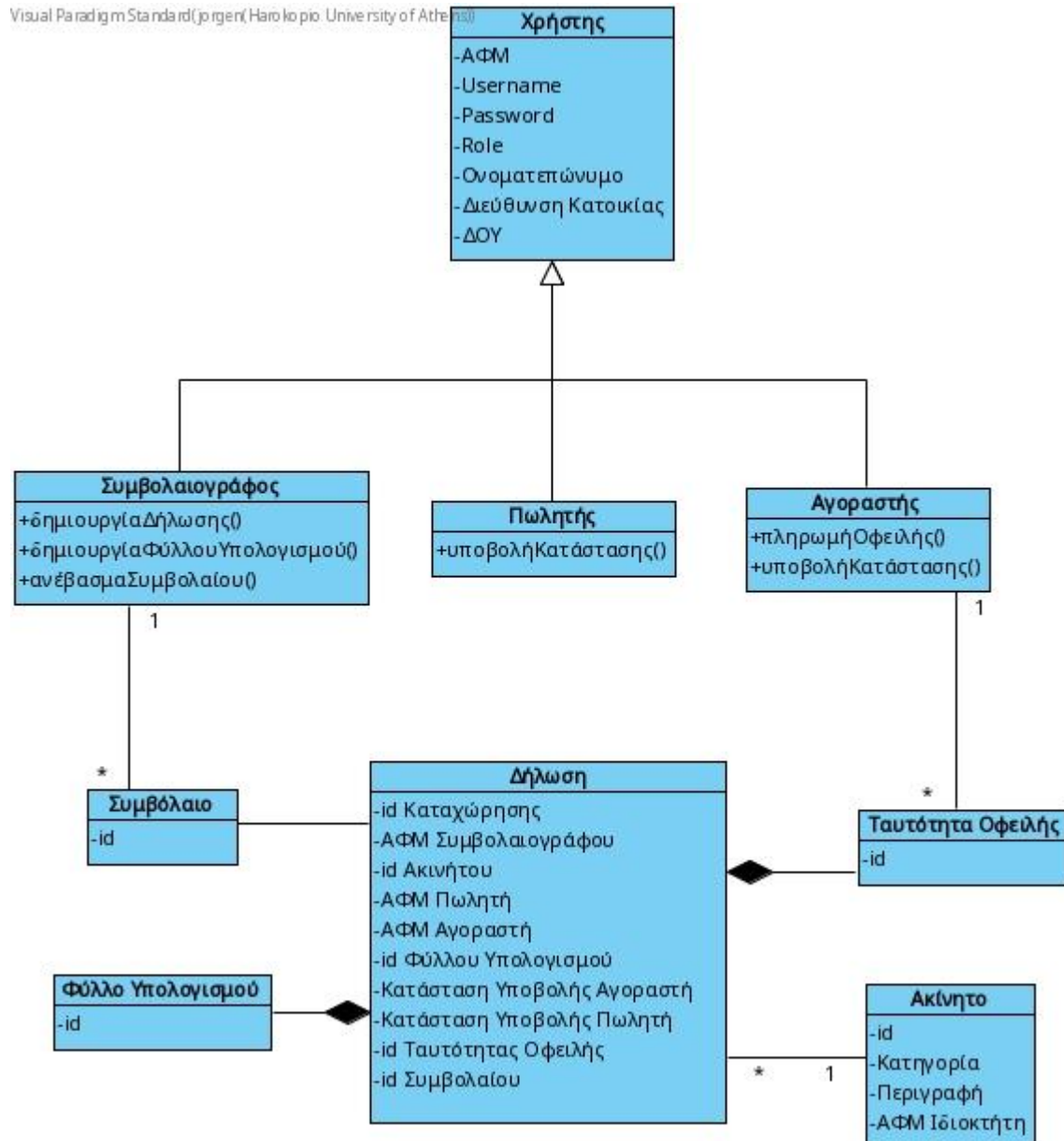
API application θα είναι το **backend** που θα υλοποιηθεί μέσω Spring, όπως και ζητείται στην εργασία.

Για την βάση δεδομένων αναφέρουμε ότι θα χρησιμοποιηθεί PostgreSQL.

Οι χρήστες θα αυθεντικοποιούνται με χρήση JWT.

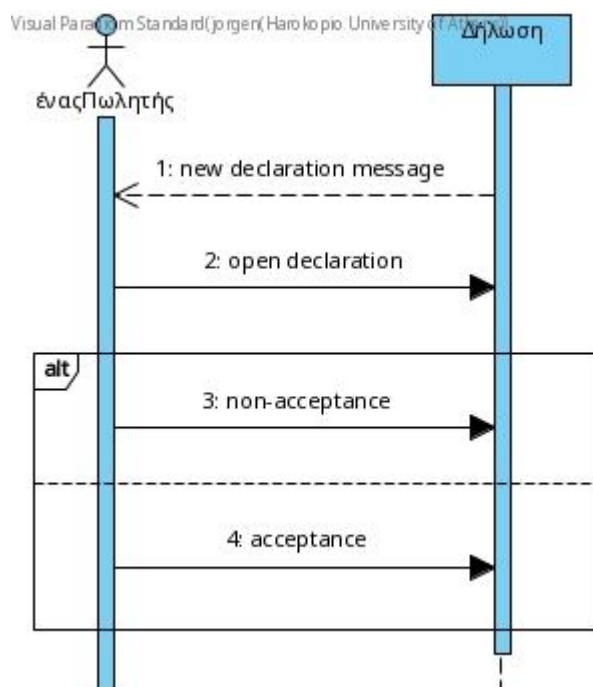
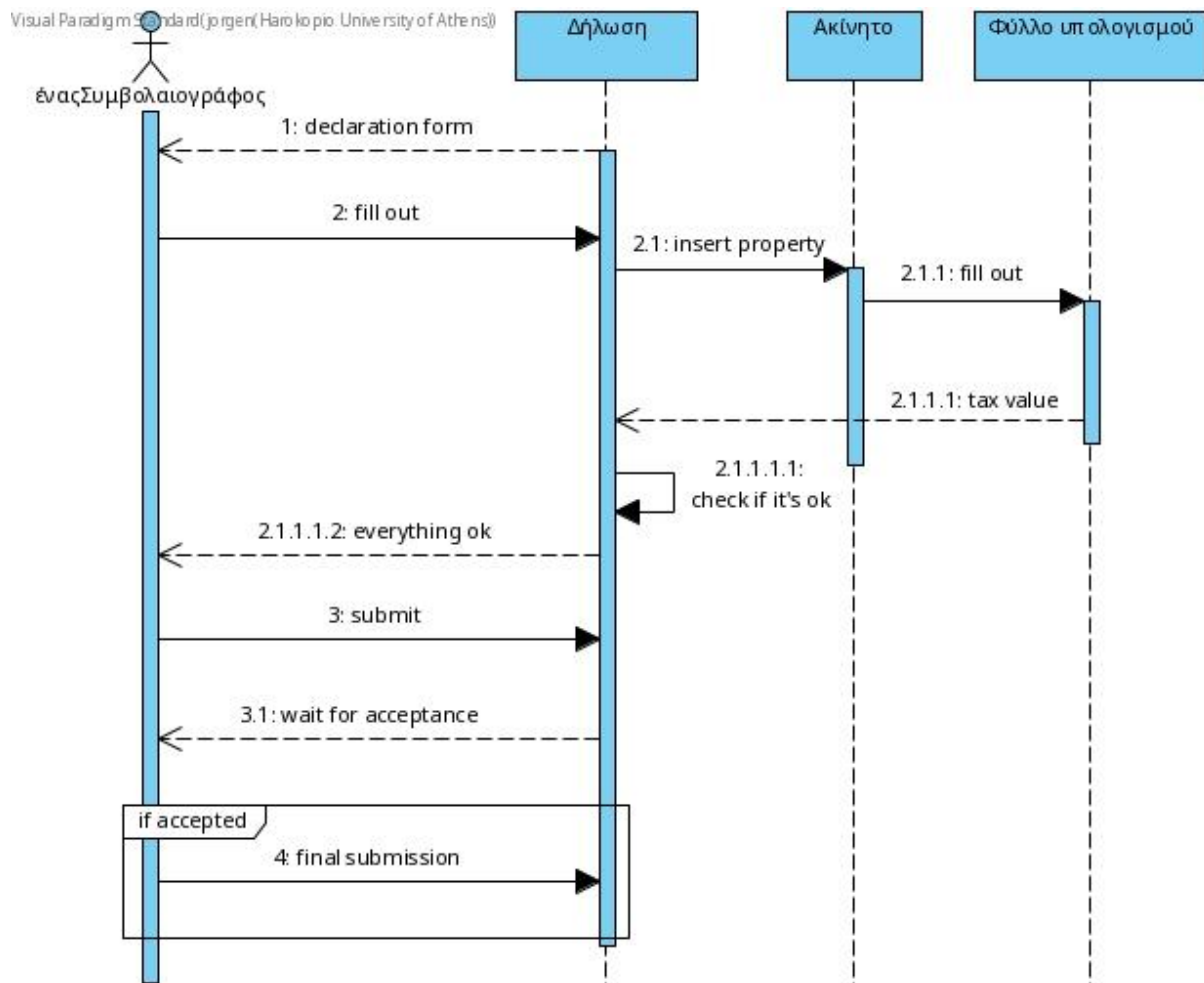
UML Διαγράμματα

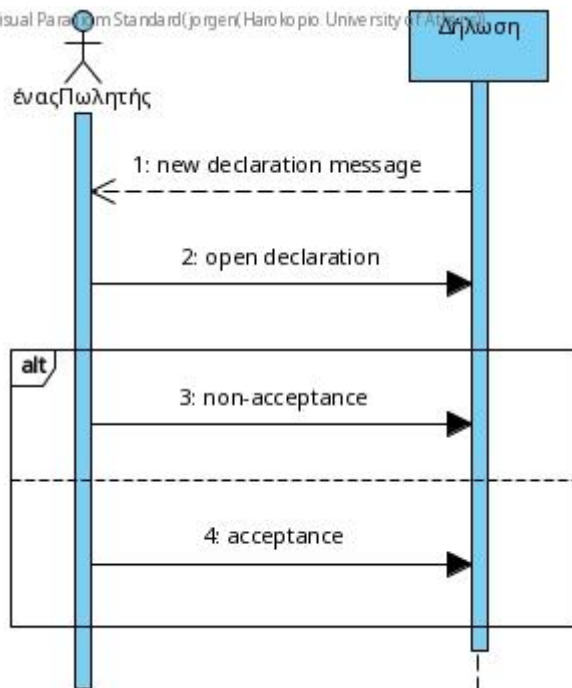
Class Diagram



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

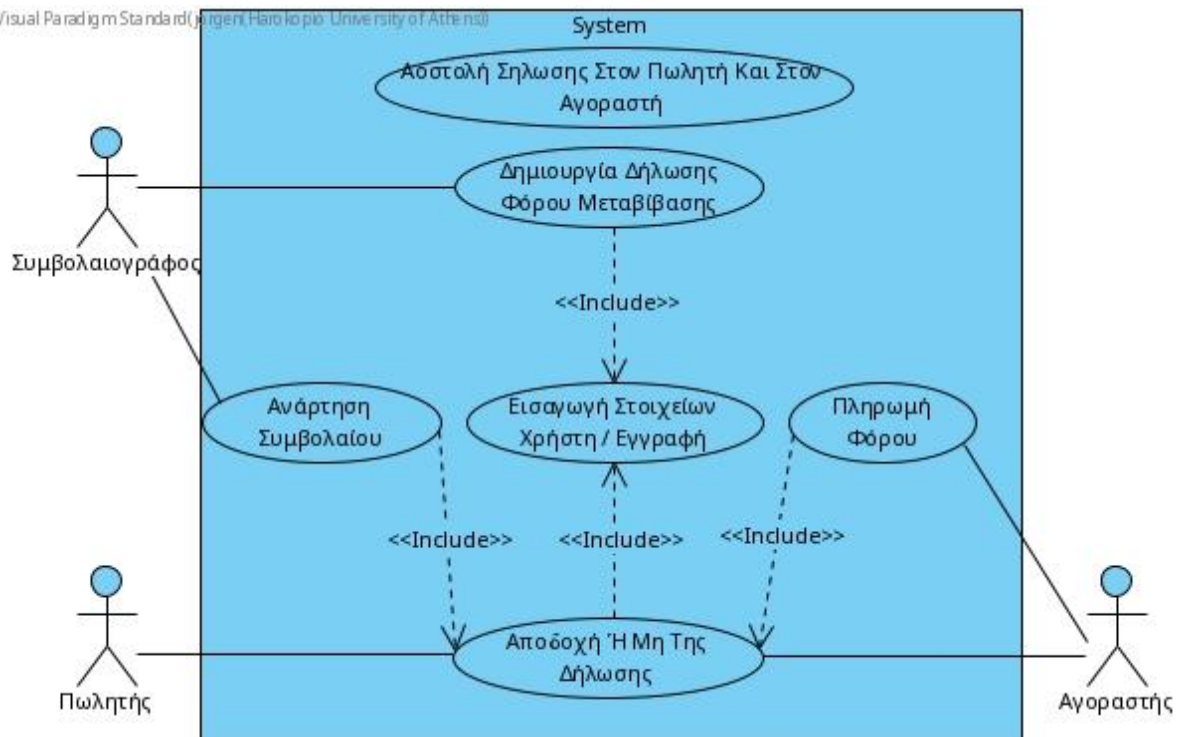
UML Sequence Diagram





UML Use Case Diagram

Visual Paradigm Standard (Katerini Haralabopoulou, University of Athens)



Παραδοχές

- A. Ο συμβολαιογράφος όταν θα συμπληρώνει την δήλωση που δημιουργεί, θα φτιάχνει δίνει τα στοιχεία του ακινήτου και αυτόματα στο σύστημα, κατά την υποβολή δηλαδή στο backend, θα δημιουργεί ένα entry σε έναν πίνακα *Ακινήτων* στην βάση, η οποία θα μπορεί να επαναχρησιμοποιηθεί.
- B. 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.
- C. Στην διαδικασία πληρωμής του φόρου από κάποιον χρήστη με τον ρόλο του *Αγοραστή* η πληρωμή θα γίνεται αυτόματα από το σύστημα, όμως θα παράγεται κανονικά η ταυτότητα οφειλής.
- D. Σχετικά με τον συμβολαιογράφο, όταν θα χρειαστεί να ανεβάσει το συμβόλαιο που έχει φτιάξει, θα ανεβάζει αρχείο (*πιθανότατα να αλλάξει αυτό και είτε θα: I) φτιάχνει εκείνη την στιγμή κάποιο πολύ απλό συμβόλαιο μέσω κάποιας φόρμας ή II) θα γίνεται και αυτό αυτόματα, όπως στην παραδοχή C.*)
- E. Εάν έστω και ένας από τους εμπλεκόμενους στη δήλωση δεν την αποδεχθεί, τότε αυτή θα γίνεται αυτόματα άκυρη και δεν θα μπορεί να συνεχιστεί.
- F. Οι χρήστες θα περνάνε από διαδικασία Sign Up και θα τους δίνεται κάποιος ΑΦΜ, ο οποίος θα παράγεται τυχαία.
- G. Εάν κάποιος *Συμβολαιογράφος* θέλει να συμμετάσχει στην διαδικασία αυτή είτε ως *Αγοραστής*, είτε ως *Πωλητής*, θα χρειαστεί να ανοίξει καινούργιο λογαριασμό.

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>

Είναι private repositories, εάν θέλετε access με ευχαρίστηση μπορώ να σας προσθέσω ως collaborator