

ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ



“Εγκατάσταση και λειτουργία εφαρμογής
διαχείρισης πτυχιακών εργασιών στο τμήμα
Πληροφορικής και Τηλεματικής”

Πτυχιακή εργασία

Αλέξανδρος Καρβουνιάρης

Αθήνα 2020

ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ



Επιβλέπων:

Ηρακλής Βαρλάμης, Επίκουρος καθηγητής του τμήματος Πληροφορικής και Τηλεματικής του Χαροκοπείου Πανεπιστημίου

Μέλη:

Θωμάς Καμαλάκης, Επίκουρος καθηγητής του τμήματος Πληροφορικής και Τηλεματικής του Χαροκοπείου Πανεπιστημίου

Ανάργυρος Τσαδήμας, Εργαστηριακό Διδακτικό Προσωπικό του τμήματος Πληροφορικής και Τηλεματικής του Χαροκοπείου Πανεπιστημίου

Ο Αλέξανδρος Καρβουνιάρης

δηλώνω υπεύθυνα ότι:

- 1. Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλει τα πνευματικά δικαιώματα τρίτων.**

- 2. Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.**

Περίληψη

Αντικείμενο της παρούσας εργασίας αποτελεί η λειτουργία μιας εφαρμογής διαχείρισης πτυχιακών εργασιών στο τμήμα Πληροφορικής και Τηλεματικής του Χαροκοπείου πανεπιστημίου. Ως πρώτο βήμα, αποφασίστηκε να γίνει συνεισφορά σε ένα ήδη υπάρχον λογισμικό διαχείρισης πτυχιακών εργασιών ανοικτού κώδικα, έτσι ώστε να φτάσει στο επίπεδο να πληροί τις προϋποθέσεις που απαιτούνται για την ενσωμάτωσή του ως υπηρεσία του Χαροκοπείου πανεπιστημίου.

Τα κίνητρα και τα οφέλη πίσω από τη συγκεκριμένη επιλογή θα αναφερθούν στην αρχή της **Ενότητας 1**, ενώ η κεντρική ιδέα είναι η μείωση του χρόνου σχεδίασης και ανάπτυξης της εφαρμογής, που αργότερα θα χρειαστεί να αφιερωθεί στη διαδικασία εγκατάστασης και λειτουργίας.

Η εφαρμογή ανοικτού κώδικα που χρησιμοποιήθηκε ως βάση, επιτρέπει την αλληλεπίδραση μεταξύ των παρακάτω ατόμων (actors): **Student** (Φοιτητής), **Professor** (Καθηγητής), **Secretariat** (Γραμματεία), **External** (Εξωτερικός), **Admin** (Διαχειριστής) και **Guest** (Φιλοξενούμενος). Περαιτέρω πληροφορίες για την εφαρμογή παρατίθενται στην **Ενότητα 1**. Για να επιτευχθεί η πλήρης ενσωμάτωση του παραπάνω συστήματος στις υπηρεσίες του πανεπιστημίου, απαιτείται η επέκταση της πρώτης έκδοσης της πλατφόρμας, έτσι ώστε **(α)** να μπορεί να αλληλεπιδρά με τον κατάλογο χρηστών του οργανισμού, **(β)** να μπορεί να δέχεται εξωτερικούς συνεργάτες ως μέλη μιας εργασίας, **(γ)** να μπορεί να συνθέτει αυτόματα έγγραφο με όλες τις αναθέσεις που έχουν γίνει ανά επιθυμητή περίοδο και **(δ)** να λειτουργεί σε απομακρυσμένο server (εξυπηρετητή) του πανεπιστημίου.

Αναφορικά με το (α), έγινε ενσωμάτωση βιβλιοθήκης για την πρόσβαση στον κατάλογο χρηστών του οργανισμού ο οποίος είναι αποθηκευμένος σε LDAP (Lightweight Directory Access Protocol) server, με αποτέλεσμα να μπορούν να συνδεθούν στην εφαρμογή όλα τα μέλη του οργανισμού με τα ίδια διαπιστευτήρια που χρησιμοποιούν για να συνδεθούν σε οποιαδήποτε άλλη υπηρεσία του πανεπιστημίου. Ανάλογα με τον ρόλο του χρήστη στον οργανισμό, ανακατευθύνεται σε αντίστοιχη σελίδα της πλατφόρμας. Διαφορετικά, σε περίπτωση που ανατεθεί στο χρήστη ο ρόλος του φιλοξενούμενου από το σύστημα, έχει υλοποιηθεί η δυνατότητα αλλαγής ρόλου από τον admin (διαχειριστή).

Στην **Ενότητα 2** περιγράφεται αναλυτικά η διαχείριση ρόλων στο σύστημα.

Σχετικά με το (β), όπως θα αναλυθεί και στην **Ενότητα 2**, σαν λειτουργική απαίτηση προέκυψε η ανάπτυξη ενός επιπλέον ρόλου στο σύστημα, ο οποίος θα δίνει την δυνατότητα σε εξωτερικούς συνεργάτες να δέχονται να γίνουν μέλη σε κάποια εργασία, καθώς και να παρακολουθούν όλα τα προσχέδια που ανεβάζει ο φοιτητής για την εργασία αυτή. Ταυτόχρονα, οι καθηγητές του πανεπιστημίου θα μπορούν να κάνουν αιτήματα σε

εξωτερικούς συνεργάτες να πάρουν μέρος στην παρακολούθηση μιας συγκεκριμένης εργασίας. Σημειώνεται, πως ο εξωτερικός συνεργάτης θα πρέπει να κάνει εγγραφή στο σύστημα συμπληρώνοντας τα στοιχεία του και σε επόμενο χρόνο ο διαχειριστής (admin), θα δύναται να του αναθέτει το ρόλο του External-Professor και να ενεργοποιεί τον λογαριασμό του. (Προσοχή: Οποιαδήποτε εσωτερική εγγραφή στο σύστημα πρέπει πρώτα να εγκριθεί από τον διαχειριστή, ενώ τα μέλη του πανεπιστημίου δεν χρειάζονται να εγγραφούν)

Για το (γ), στα πλαίσια της αυτοματοποίησης των διαδικασιών που εκτελούνται από την γραμματεία, προσαρτήθηκε παράδειγμα εγγράφου το οποίο περιέχει αναλυτικά τις αναθέσεις των εργασιών σε φοιτητές, με σκοπό να γίνεται η αυτόματη δημιουργία του από το σύστημα. Πλέον, το διοικητικό προσωπικό του οργανισμού επιπρόσθετα από την πλήρη εικόνα των αναθέσεων για φοιτητές και καθηγητές, έχει την δυνατότητα να κατεβάσει στον υπολογιστή του αρχείο που περιέχει αυτές τις πληροφορίες για οποιαδήποτε ημερολογιακή περίοδο επιθυμεί.

Όσον αφορά το (δ), η αρχική έκδοση της εφαρμογής επεκτάθηκε τοπικά σε μορφή container, κάνοντας χρήση της πλατφόρμας docker.

Στην **Ενότητα 3** θα γίνει εκτενής ανάλυση αυτής της διαδικασίας. Επιγραμματικά, με χρήση του πρωτοκόλλου ssh (secure shell), έγινε η πρόσβαση σε απομακρυσμένο server του πανεπιστημίου, όπου κι έγινε εγκατάσταση του docker σε αυτόν. Έπειτα, με την εντολή scp (secure copy) ο source code της εφαρμογής μεταφέρθηκε στον remote server. Κατά τη διαδικασία ανάπτυξης και ελέγχου της εφαρμογής, η μεταφορά του τοπικού κώδικα στον server καθιστά επαναλαμβανόμενη διαδικασία, και για τον λόγο αυτό προτείνεται εναλλακτική μέθοδος.

Επίσης, στην **Ενότητα 4** παρουσιάζεται εγχειρίδιο χρήσης της πλατφόρμας, το οποίο καλύπτει τις βασικότερες λειτουργίες της, όπως είναι η δήλωση και ανάθεση εργασιών, ο ορισμός τριμελούς και η εκτύπωση αρχείου αναθέσεων από την γραμματεία.

Συμπληρωματικά, έγιναν ποικίλες αλλαγές/διορθώσεις τόσο στο backend όσο και στο frontend της πρώτης έκδοσης της εφαρμογής, με σκοπό την εξασφάλιση της ορθής λειτουργίας και συνοχή της.

Λέξεις Κλειδιά: Εφαρμογές Ιστού, Ανοικτό Λογισμικό, Διαδικτυακές πλατφόρμες, διαχείριση θεμάτων πτυχιακών εργασιών

Abstract

The object of the present thesis is the operation of a thesis management application in the Department of Informatics and Telematics of Harokopio University. As a first step, it was decided to contribute to an already existing open source dissertation management software, so that it meets all the high standard requirements of Harokopio University in order to be integrated in its services.

The open source application used as a base project, allows the interaction between the following individuals (actors): Student, Professor, Secretariat, External, Admin and Guest. In order to achieve the full integration of the above system in the services of the university, the extension of the first version of the platform is necessary, so that **(a)** it can interact with the user list of the organization, **(b)** it can accept external partners as members in thesis **(c)** it is able to automatically compose a document with all assignments made during a desired period and **(d)** it can operate on a remote server of the university.

Regarding **(a)**, a library was integrated to access the organization's user list stored on a LDAP (Lightweight Directory Access Protocol) server, so that all members of the organization can connect to the application using the same credentials as they use to connect to any other university service.

For **(b)**, a new functional requirement emerged that indicates the development of an additional role in the system, which will enable external collaborators to accept to become members of a thesis, as well as to monitor all the drafts submitted by the student for this assignment. This new role (**External-Professor**) can be assigned to an external user by the administrator of the system at any given time after the registration of the respective user.

As for **(c)**, in the context of the process automation performed by the secretariat, in the new version of the application the administrative staff of the organization, in addition to the complete overview of the assignments for students and teachers, they have the ability to download a file that contains all the assignment information, during any calendar period.

Lastly for **(d)**, the first version of the application was containerized by leveraging the docker platform. This action, had a tremendous positive impact in terms of developer operations (DevOps). In more detail, it made the development and testing processes easier as well as the deployment of the entire application in the remote university server.

Keywords: open source, web application, web platform, open source software, thesis management

Περιεχόμενα

1 Εισαγωγή	8
1.1 Αναγνώριση Διαδικασίας	8
1.2 Έργα ανοιχτού κώδικα	9
1.3 Σύστημα διαχείρισης πτυχιακών εργασιών ανοικτού κώδικα	10
1.4 Τεχνολογίες που χρησιμοποιήθηκαν	14
2 Επέκταση αρχικής έκδοσης της εφαρμογής	15
2.1 Σύνδεση από μέλη του Χαροκοπείου πανεπιστημίου	16
2.2 Εξωτερικός συνεργάτης ως μέλος τριμελούς	18
2.3 Διαχείριση ρόλων	20
2.4 Αρχείο αναθέσεων	22
3 Λειτουργίες προγραμματιστή και Εγκατάσταση λογισμικού σε server	25
3.1 docker-compose	25
3.2 Εγκατάσταση και λειτουργία εφαρμογής σε απομακρυσμένο server	26
4 Εγχειρίδιο βασικών σεναρίων χρήσης	28
4.1 Δήλωση θέματος εργασίας (Professor)	29
4.2 Ανάθεση θέματος εργασίας	30
4.3 Ορισμός Τριμελούς (Professor & External-Professor)	31
4.4 Κατέβασμα εγγράφου αναθέσεων (Secretariat)	33
5 Συμπεράσματα	34
5.1 Μελλοντικές εργασίες	34
Αναφορές	35

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικ.1. Πρόταση Θέματος	Σελ 14
Εικ.2. Φοιτητής-Καθηγητής-Θέμα	Σελ 14
Εικ.3. Τεχνολογίες (MEAN Stack).....	Σελ 16
Εικ.4. Διάγραμμα ροής σύνδεσης μελών	Σελ 19
Εικ.5. Ρόλοι μελών πανεπιστημίου	Σελ 21
Εικ.6. Διάγραμμα ροής διαχείρισης ρόλων	Σελ 23
Εικ.7. Διάγραμμα ροής εγγράφου αναθέσεων	Σελ 25

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πιν.1. Σενάριο χρήσης σύνδεσης μελών.....	Σελ 18
Πιν.2. Σενάριο χρήσης εξωτερικών μελών ως τριμελής.....	Σελ 21
Πιν.3. Σενάριο χρήσης διαχείρισης ρόλων.....	Σελ 22
Πιν.4. Σενάριο χρήσης εγγράφου αναθέσεων.....	Σελ 24

1 Εισαγωγή

Στις μέρες μας, πολλές υπηρεσίες που παλαιότερα απαιτούσαν τη φυσική παρουσία των ενδιαφερόμενων έχουν μεταφερθεί στο διαδίκτυο. Ως φυσική συνέχεια, όλο και περισσότεροι δημόσιοι φορείς λαμβάνουν δράσεις για την ψηφιοποίηση των υπηρεσιών τους. Στα πλαίσια λοιπόν της ψηφιοποίησης του Χαροκοπείου πανεπιστημίου, κρίθηκε αναγκαία η ανάπτυξη λογισμικού έτσι ώστε να αντικαταστήσει την υπάρχουσα διαδικασία διαχείρισης πτυχιακών εργασιών του οργανισμού.

1.1 Αναγνώριση Διαδικασίας

Έως και σήμερα, στη μέση της διαδικασίας διαχείρισης πτυχιακών εργασιών βρίσκεται η γραμματεία, η οποία δρα ως διαμεσολαβητής ανάμεσα στον καθηγητή και τον φοιτητή. Πιο συγκεκριμένα, σε υψηλό επίπεδο η διαδικασία έχει ως εξής: Ο **καθηγητής** μοιράζεται με τη **γραμματεία** τα θέματα που θα επιβλέπει, η **γραμματεία** αφού συλλέξει τα θέματα από όλους τους **καθηγητές** τα μοιράζεται με τους **φοιτητές** και εκείνοι δηλώνουν το θέμα που τους ενδιαφέρει να ασχοληθούν πάλι στην **γραμματεία**. Έπειτα, η **γραμματεία** ζητά την επιβεβαίωση του επιβλέποντα **καθηγητή** για να ολοκληρώσει επιτυχώς την ανάθεση και προχωράει σε σύνθεση εγγράφου με όλες τις αναθέσεις. Στη παραπάνω διαδικασία υπάρχουν διάφορες εναλλακτικές ροές οι οποίες δεν θα αναλυθούν περαιτέρω. Φαίνεται παρόλα αυτά, ο αριθμός των βημάτων που χρειάζονται για να ολοκληρωθεί.

Η ιδέα μιας κεντρικής τοποθεσίας όπου ο φοιτητής θα έχει άμεση επικοινωνία με τον καθηγητή και αντίστροφα, αλλά και η γραμματεία θα μπορεί να παρακολουθεί τη διαδικασία απομακρυσμένα δεν είναι νέα. Παρόλα αυτά, δεν υπάρχει ολοκληρωμένη υλοποίηση τέτοιας πλατφόρμας που να δρα ως υπηρεσία κάποιου πανεπιστημίου.

Ένας πιθανός λόγος που συμβαίνει αυτό, είναι ότι η ανάπτυξη και η λειτουργική εγκατάσταση μιας τέτοιας εφαρμογής στο διαδίκτυο είναι αρκετά κοστοβόρα. Εντούτοις, δεν δύναται να διεκπαιρευθεί εξ ολοκλήρου από 1 άτομο μέσα σε εύλογο χρονικό διάστημα (πχ. ως πτυχιακή εργασία), με αποτέλεσμα να ξεκινά η ανάπτυξη της και να μένει στη μέση.

Είναι κατανοητό, πως δεν χρειάζεται η ανάπτυξη μιας ακόμη ημιτελούς εφαρμογής, αλλά ένας λειτουργικός πιλότος ο οποίος αρχικά θα δοκιμαστεί και σε επόμενο βήμα θα είναι σε θέση να ενσωματωθεί στις υπηρεσίες του πανεπιστημίου.

Ως πιθανή λύση στο παραπάνω πρόβλημα προτάθηκε η συνεισφορά σε ήδη υπάρχον έργο ανοικτού κώδικα, όπου θα προστεθούν οι απαραίτητες επιπρόσθετες λειτουργίες που χρειάζονται, για να ολοκληρωθεί η ανάπτυξη της πρώτης έκδοσης της εφαρμογής και να λειτουργεί μαζί με τις υπάρχουσες διαδικασίες του Χαροκοπείου πανεπιστημίου.

Τι είναι όμως ένα έργο ανοικτού κώδικα, ποιά είναι τα κίνητρα στην συνεισφορά του και γιατί αποτελεί λύση στο πρόβλημα όπως περιγράφηκε;

1.2 Έργα ανοιχτού κώδικα

Γενικά, υπάρχουν δύο (2) τύποι ανάπτυξης λογισμικών: τα λογισμικά κλειστού κώδικα (proprietary) και τα λογισμικά ανοιχτού κώδικα (open source). Η σημαντική διαφορά ανάμεσα στους δύο αυτούς τύπους είναι, πως ενώ στο πρώτο ο πηγαίος κώδικας (source code) της εφαρμογής είναι διαθέσιμος μόνο στο φυσικό πρόσωπο που τον έχει γράψει (άτομο ή εταιρία), στο δεύτερο είναι διαθέσιμος σε ειδικά ψηφιακά αποθετήρια που μπορεί να τον δει ο οποιοσδήποτε. Επιπλέον, τα λογισμικά ανοιχτού κώδικα επιτρέπουν την ελεύθερη πρόσβαση, εγκατάσταση ακόμη και την επέκταση τους από τρίτους.

Στην πραγματικότητα, τα λογισμικά ανοιχτού κώδικα αντιπροσωπεύουν πολλά περισσότερα από τη διάθεση του πηγαίου κώδικα τους στον κόσμο. Οι ίδιοι οι χρήστες των λογισμικών αυτών, συνεισφέρουν συστηματικά σε αυτά με ποικίλους τρόπους κάνοντας τα καλύτερα. Πιο συχνά συμπληρώνουν λειτουργίες σε αυτά αλλά και διορθώνουν ελαττώματα που τυχόν προκύπτουν. Δεν είναι τυχαίο γεγονός ότι κάποια από αυτά είναι τόσο πετυχημένα που ανταγωνίζονται κλειστά λογισμικά που υποστηρίζονται από κεφαλαιούχες εταιρείες με τρανό παράδειγμα τον Apache web server , ο οποίος νικάει σε δημοτικότητα τον αντίστοιχο Netscape της Microsoft. Μέχρι σήμερα, πολλά πετυχημένα λογισμικά ανοιχτού κώδικα συνεχίζουν να αναπτύσσονται, διανέμονται και υποστηρίζονται από τους χρήστες και για τους χρήστες χωρίς την επέμβαση κατασκευαστών/προμηθευτών. [1]

Έχοντας εδραιωθεί στη σημερινή πραγματικότητα, στα λογισμικά ανοιχτού κώδικα επιλέγουν να επενδύσουν πλέον και εμπορικές εταιρίες συνεισφέροντας στην ανάπτυξή τους. Μάλιστα

υπάρχουν και πολλές περιπτώσεις που εταιρείες έχουν κάνει διαθέσιμο στον κόσμο το κλειστό λογισμικό τους ή και ξεκινούν να αναπτύσουν νέα έργα ανοιχτού λογισμικού οι ίδιες. Όσο οξύμωρο κι αν ακούγεται αυτό, υπάρχουν επαρκή κίνητρα για τις εμπορικές εταιρείες να επενδύουν σε λογισμικά ανοιχτού κώδικα. Ενδεικτικά, η πώληση συμπληρωματικών υπηρεσιών και υπηρεσιών υποστήριξης αποτελεί έναν από τους κυριότερους τρόπους απόκτησης χρηματικής αμοιβής και επομένως αξιόπιστο κίνητρο για τις εταιρίες. Ταυτόχρονα, το κόστος για τις εταιρείες όταν αναθέτουν την ανάπτυξη των λογισμικών του σε εξωτερικές κοινότητες είναι μειωμένο. [2]

Περισσότερο όμως ενδιαφέρον έχουν τα κίνητρα για τους χρήστες που συνεισφέρουν στην ανάπτυξη λογισμικών ανοιχτού κώδικα, τις περισσότερες φορές χωρίς χρηματική ανταμοιβή.

1. Σύμφωνα με συνεντεύξεις στους ίδιους τους συνεισφέροντες, φαίνεται να επικαλούνται ως τον κυριότερο τους κίνητρο, την προσωπική εξέλιξη και ταυτόχρονα την φήμη που τους αποδίδεται από τις συνεισφορές τους [3]. Είναι αλήθεια, πως μια ποιοτική συνεισφορά σε ένα έργο ανοιχτού λογισμικού φέρει το όνομα αυτού που την έκανε και κατ' επέκταση ενισχύει τη φήμη του από την κοινότητα και οποιοδήποτε άτομο/εταιρία παρακολουθεί το έργο.
2. Επιπλέον κίνητρο, αποτελεί και η άμεση ανάγκη του ατόμου για βελτιώσεις του λογισμικού στο οποίο συνεισφέρει. [4]. Είναι λογικό για κάποιον που είναι χρήστης ενός λογισμικού και χρειάζεται μία βελτίωση ή επιπλέον λειτουργία του λογισμικού να την προσθέσει ο ίδιος εφόσον έχει την δυνατότητα. Πολλές φορές, τα λογισμικά κλειστού τύπου δεν μπορούν να καλύψουν τις ανάγκες όλων των χρηστών αφού είτε αδυνατούν να τις γνωρίζουν στο σύνολό τους, είτε διότι δεν τους συμφέρει οικονομικά να το κάνουν. Στην ίδια κατηγορία τοποθετείται και η προσέγγιση της παρούσας εργασίας στα πλαίσια της ανάπτυξης λειτουργικού λογισμικού για τη διαχείριση πτυχιικών εργασιών με την έννοια πως η χρήση του λογισμικού ανοιχτού κώδικα που επιλέχθηκε μειώνει σημαντικά το χρόνο ανάπτυξης, αλλά χρειάζεται να προστεθούν παραπάνω λειτουργίες.

1.3 Σύστημα διαχείρισης πτυχιικών εργασιών ανοιχτού κώδικα

Το έργο ανοιχτού κώδικα που επιλέχθηκε να αποτελέσει τη βάση για το λογισμικό διαχείρισης πτυχιικών εργασιών του Χαροκοπείου πανεπιστημίου αναπτύχθηκε στα πλαίσια του Google Summer of Code 2019 από τον [Μιχάλη Χατζηαναστάση](#) και διατίθεται με ειδικές άδειες, οι οποίες επιτρέπουν στους χρήστες να μελετήσουν, να τροποποιήσουν και να βελτιώσουν το λογισμικό. Ολοκληρωμένο το έργο βρίσκεται στην εξής διεύθυνση:

<https://github.com/eellak/gsoc2019-tms>

Αφορά την ανάπτυξη μιας διαδικτυακής εφαρμογής με τίτλο Thesis Management System ή αλλιώς TMS, η οποία υποστηρίζει τη διαδικασία της διαχείρισης πτυχιακών εργασιών σε όλο το εύρος της. Η εφαρμογή σε βασικό επίπεδο, αποσκοπεί στον περιορισμό της χρονοβόρας διαδικασίας της διαχείρισης πτυχιακών εργασιών και της γραφειοκρατίας που απαιτεί, κρίνοντας την κατάλληλη για να αποτελέσει το σημείο αναφοράς της παρούσας εργασίας.

Οι ρόλοι που υποστηρίζει το σύστημα σε αυτή την **πρώτη έκδοση** είναι οι εξής:

- **Guest**: Φιλοξενούμενος χρήστης με περιορισμένη πρόσβαση μόνο στην αρχική σελίδα. Δεν απαιτείται η είσοδος του φιλοξενούμενου χρήστη στο σύστημα.
- **Student**: Πανεπιστημιακός φοιτητής. Ο ρόλος του φοιτητή, υποστηρίζει λειτουργίες όπως:
 - Η πλοήγηση διαθέσιμων εργασιών στις οποίες μπορεί να κάνει αίτημα με σχόλιο, το οποίο πηγαίνει κατευθείαν στον επιβλέποντα καθηγητή.
 - Ο έλεγχος όλων των αιτημάτων του που εκκρεμούν.
 - Η επισκόπηση του θέματος της εργασίας που του έχει ανατεθεί, αλλά και η δυνατότητα να αναρτά προσχέδια τα οποία μοιράζονται και στα μέλη της της εργασίας.
 - Η δυνατότητα να προτείνει ο ίδιος νέο θέμα εργασίας το οποίο πρέπει να εγκριθεί από κάποιον καθηγητή του ίδιου πανεπιστημίου, προκειμένου να προστεθεί στην πλίσινά των διαθέσιμων θεμάτων.

Απαιτείται η είσοδος του στο σύστημα με τα διαπιστευτήρια που υπάρχουν στον κατάλογο χρηστών του πανεπιστημίου, η οποία υλοποιείται στα πλαίσια της παρούσας εργασίας.

- **Professor**: Καθηγητής που είναι ενταγμένος σε πανεπιστήμιο. Ένας πανεπιστημιακός καθηγητής μπορεί να:
 - Δημιουργεί νέα θέματα εργασιών, τα οποία σε επόμενη φάση προσθέτονται στη πλίσινά των διαθέσιμων θεμάτων επιτρέποντας στους φοιτητές να κάνουν αίτηση σε αυτά.
 - Να επιβλέπει εργασίες, είτε ως κύριος επιβλέπων ή ως μέλος τριμελούς επιτροπής. Ως κύριος επιβλέπων ή δημιουργός εργασίας, μπορεί να ζητήσει τη συμμετοχή άλλων καθηγητών ως μέλη τριμελούς στην εργασία..
 - Να διαχειρίζεται τυχόν αιτήματα για συμμετοχή σε εργασίες από άλλους καθηγητές ή/και αιτήματα φοιτητών για ανάθεση τους σε εργασίες.
 - Να βλέπει και να διαχειρίζεται εκκρεμή νέα θέματα εργασιών που έχουν προταθεί από φοιτητές ή εξωτερικούς συνεργάτες.

Όπως ισχύει και για τον φοιτητή, απαιτείται η είσοδος του στο σύστημα.

- **External**: Εξωτερικός συνεργάτης. Η λειτουργία που υποστηρίζεται από το σύστημα για τον ρόλο αυτόν στην **πρώτη έκδοση της εφαρμογής**, περιορίζεται στο:
 - να μπορεί να προτείνει νέα θέματα εργασιών, τα οποία μπορούν να εγκρίνουν ή όχι καθηγητές ώστε να προστεθούν στην πλίσινα των διαθέσιμων θεμάτων.

Εξωτερικοί συνεργάτες μπορούν σε πρώτη φάση να εγγραφούν στο σύστημα συμπληρώνοντας τα στοιχεία τους. Εντούτοις, απαιτείται να εγκριθεί ο λογαριασμός τους από τον διαχειριστή του συστήματος ώστε να μπορούν να συνδεθούν στο σύστημα.

- **Secretariat**: Ο ρόλος της γραμματείας. Η γραμματεία έχει τη δυνατότητα να έχει εικόνα:
 - των φοιτητών και των αναθέσεων τους σε εργασίες
 - και των καθηγητών του πανεπιστημίου που είναι ενταγμένη

Όπως και οι άλλοι ρόλοι που είναι πανεπιστημιακά μέλη πρέπει να συνδεθούν στο σύστημα με τα διαπιστευτήρια που χρησιμοποιούν για να συνδεθούν και στις υπόλοιπες υπηρεσίες του οργανισμού. Αυτό θα υλοποιηθεί ξεχωριστά σαν επιπρόσθετη λειτουργία και αναφέρεται στην επόμενη ενότητα.

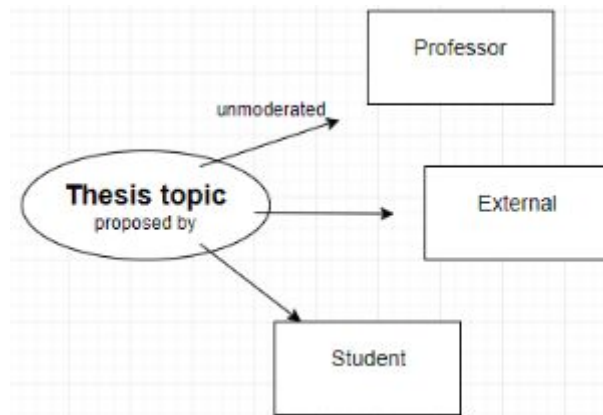
- **Admin**: Τέλος ο admin αντιστοιχεί στον ρόλο του διαχειριστή στο σύστημα. Μπορεί να:
 - Προσθέσει νέα πανεπιστήμια στο σύστημα.
 - Να αφαιρέσει υπάρχοντα πανεπιστήμια.
 - Να επιβλέπει και να ενεργοποιεί λογαριασμούς εξωτερικών συνεργατών.
 - Να επιβλέπει τους πανεπιστημιακούς χρήστες.
 - Να αφαιρέσει οποιοδήποτε χρήστη από το σύστημα.

Για όλους τους παραπάνω ρόλους, η εφαρμογή παρέχει αντίστοιχα εργαλεία και γραφικές διεπαφές επιτρέποντας την αλληλεπίδραση τους σε ένα κοινό περιβάλλον.

Για την καλύτερη κατανόηση του κύκλου ζωής της, η εφαρμογή μπορεί να χωριστεί σε τέσσερις (4) περιόδους.

1. **Περίοδος πρότασης θέματος εργασίας**

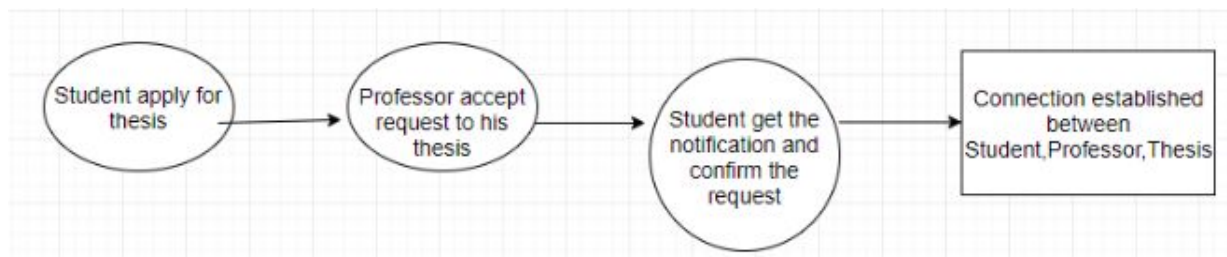
Σε αυτή τη περίοδο, οι εξωτερικοί συνεργάτες και οι πανεπιστημιακοί φοιτητές μπορούν να προτείνουν ένα θέμα εργασίας και να περιμένουν την έγκρισή του από τους καθηγητές (ελεγχόμενη πρόσβαση). Αν κάποιος καθηγητής εγκρίνει μια πρόταση και οι εξωτερικοί ή φοιτητές επιβεβαιώσουν την ενέργεια, το θέμα της εργασίας ενεργοποιείται. Μόνο οι καθηγητές μπορούν να δημιουργήσουν θέματα εργασίας χωρίς ελεγχόμενη πρόσβαση.



Εικ.1

2. Περίοδος υποβολής αιτήσεων πτυχιακών εργασιών

Στα πλαίσια αυτής της περιόδου, οι φοιτητές κάνουν αίτηση σε θέματα εργασιών του πανεπιστημίου τους που τους ενδιαφέρουν. Έπειτα, ο καθηγητής που έχει δημιουργήσει το εκάστοτε θέμα, μπορεί να απαντήσει στο αίτημα του φοιτητή ενώ ο τελευταίος ενημερώνεται για την ενδεχόμενη αποδοχή του. Ύστερα από την τελική επιβεβαίωση του φοιτητή εφαρμόζεται η σύνδεση στο τρίπτυχο **φοιτητή, καθηγητή και εργασίας** (Student, Professor, Thesis). Αυτό σημαίνει πως η **εργασία** (Thesis), της οποίας δημιουργός είναι ο **καθηγητής** (Professor), έχει ανατεθεί στον **φοιτητή** (Student). Επιπρόσθετα, ο καθηγητής μπορεί να προσκαλέσει δύο (2) επιπλέον καθηγητές οι οποίοι θα μπορούν να επιβλέπουν την εργασία. Κατά την επέκταση της εφαρμογής θα δύναται να προσκαλέσει και εξωτερικούς.



Εικ. 2

3. Περίοδος σύνταξης της εργασίας

Σε αυτή τη περίοδο, ο φοιτητής (Student) συντάσσει την εργασία που του έχει ανατεθεί σύμφωνα με την προηγούμενη διαδικασία. Σε συνεννόηση με τον δημιουργό καθηγητή (Professor) της εργασίας του μπορεί να ανεβάζει προσχέδια της δουλειάς του ανά τακτά χρονικά διαστήματα. Τα προσχέδια αυτά, μοιράζονται με όλους τους επιβλέποντες κι έτσι παρακολουθείται η πρόοδος του φοιτητή από όλους τους εμπλεκόμενους καθηγητές (τριμελής επιτροπή).

4. Περίοδος εκπόνησης της εργασίας

Τελικά, όταν ο φοιτητής με τη ταυτόχρονη συνεννόηση έχει την τελική έκδοση της εργασίας που του έχει ανατεθεί, την καταχωρεί στο σύστημα. Όπως και με τα προσχέδια, η τελική έκδοση μοιράζεται στην τριμελή επιτροπή και ύστερα μεταφορτώνεται στη ψηφιακή αποθήκη μαζί με όλες τις υπόλοιπες εργασίες που έχουν υλοποιηθεί.

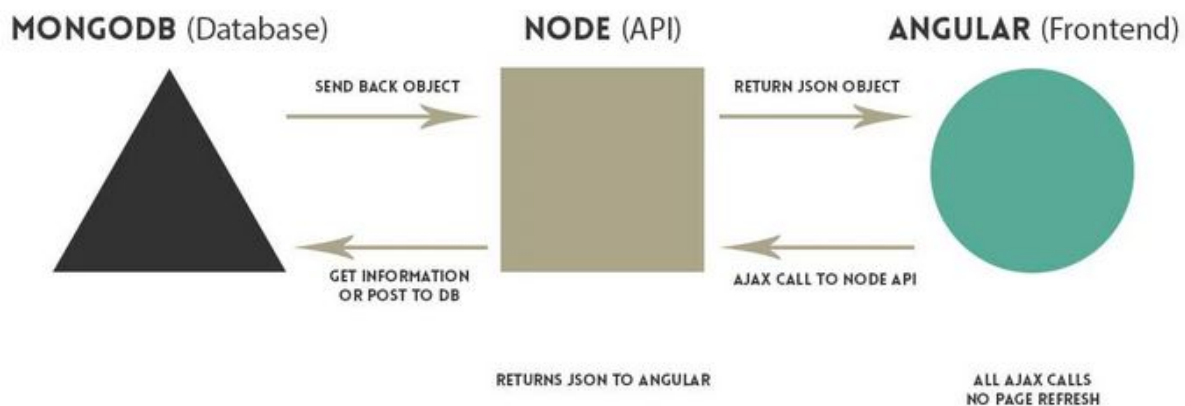
1.4 Τεχνολογίες που χρησιμοποιήθηκαν

Το Thesis Management System (TMS), εκμεταλλεύεται το MEAN stack. Το MEAN stack είναι μια δωρεάν στοίβα λογισμικού JavaScript ανοιχτού κώδικα για τη δημιουργία δυναμικών ιστοτόπων και εφαρμογών διαδικτύου, ή αλλιώς είναι μια συλλογή τεχνολογιών βασισμένων σε JavaScript που χρησιμοποιούνται για την ανάπτυξη διαδικτυακών εφαρμογών. Από τον client (χρήστη) ως τον server και την βάση δεδομένων το MEAN είναι full-stack Javascript [5]. Το MEAN είναι ακρωνύμιο για:

- **M** = MongoDB. Δημοφιλής κατακευματισμένη βάση δεδομένων που χρησιμοποιεί NoSql δομή. Στο έργο έγινε χρήση του *mongoose framework* της MongoDB για το NodeJS.

- **E** = Express.js, ένα framework που υποστηρίζει NodeJS έργα.
- **A** = Angular 8, framework για την ανάπτυξη διαδικτυακών εφαρμογών.
- **N** = Node.js. Το πιο σημαντικό συστατικό στοιχείο του MEAN stack. Το Node.js είναι μια πλατφόρμα ανάπτυξης λογισμικού χτισμένη σε περιβάλλον Javascript. Στόχος του Node είναι να παρέχει ένα εύκολο τρόπο δημιουργίας κλιμακωτών διαδικτυακών εφαρμογών.

Το MEAN stack υποστηρίζει την MVC (Model View Controller) αρχιτεκτονική. Επίσης όλα τα συστατικά στοιχεία του είναι ανοικτού κώδικα.



Εικ. 3

Η παραπάνω εικόνα περιγράφει την ροή πληροφορίας ανάμεσα στις τεχνολογίες που χρησιμοποιούνται.

Τόσο στο backend της εφαρμογής όσο και στο frontend, χρησιμοποιήθηκαν πλειάδες βιβλιοθηκών για την εξασφάλιση ορισμένων λειτουργιών. Οι ίδιες αρχιτεκτονικές χρησιμοποιήθηκαν και για την διαδικασία της επέκτασης της εφαρμογής ενώ έγινε χρήση μιας ακόμη βιβλιοθήκης.

2 Επέκταση αρχικής έκδοσης της εφαρμογής

Ενώ η αρχική έκδοση της εφαρμογής έτσι όπως έχει αναπτυχθεί, υποστηρίζει δυνητικά σχεδόν ολόκληρη τη διαδικασία διαχείρισης πτυχιακών εργασιών, δεν είναι πλήρως λειτουργική.

Σε πρώτη φάση, κατά τη χρήση της πλατφόρμας παρατηρήθηκαν διάφορα μικρά ελαττώματα, τόσο σε λειτουργικό επίπεδο όσο και σε επίπεδο εμπειρίας χρήστη. Τα πρώτα, διορθώθηκαν προσθέτοντας μερικές γραμμές κώδικα χωρίς όμως να λύνουν καίρια προβλήματα στην περαιτέρω ανάπτυξη του λογισμικού. Όσον αφορά τα δεύτερα, αποφασίστηκε να χρησιμοποιηθεί αποκλειστικά μια βιβλιοθήκη για την υλοποίηση των γραφικών διεπαφών του χρήστη έτσι ώστε να εξασφαλιστεί η συνοχή στην πλατφόρμα και κατ' επέκταση να βελτιωθεί η συνολική εμπειρία του χρήστη (Angular Material).

Παράλληλα, κατά την επικοινωνία με το Χαροκόπειο πανεπιστήμιο (κλειστή συνέντευξη) αυτή τη φορά με την ιδιότητα του πελάτη, προέκυψαν τέσσερις (4) επιπλέον λειτουργικές απαιτήσεις του συστήματος που δεν είχαν υλοποιηθεί στην πρώτη του έκδοση. Επιγραμματικά, θα πρέπει:

- Τα μέλη του πανεπιστημίου να συνδέονται στο σύστημα με τα ίδια διαπιστευτήρια που χρησιμοποιούν για να συνδεθούν και σε οποιαδήποτε άλλη υπηρεσία του Χαροκοπέιου πανεπιστημίου.
- Να μπορεί ένας εξωτερικός συνεργάτης να είναι μέρος της τριμελούς επιτροπής μιας εργασίας και να επιβλέπει την πρόοδο του φοιτητή σε αυτή.
- Να έχει την δυνατότητα ο διαχειριστής του συστήματος να διαχειρίζεται/αλλάζει τους ρόλους των χρηστών (πχ. να μπορεί να κάνει έναν φιλοξενούμενο χρήστη καθηγητή ή φοιτητή).
- Η γραμματεία να κατεβάζει ένα αρχείο τύπου .csv που να περιέχει τα ονοματεπώνυμα των φοιτητών και τις εργασίες που τους έχουν ανατεθεί, την ημερομηνία ανάθεσης και τους επιβλέποντες καθηγητές στις εργασίες αυτές. Επίσης, χρειάζεται να έχει την δυνατότητα να επιλέξει ανάμεσα σε ποιά περίοδο επιθυμεί να εμφανιστούν οι αντίστοιχες αναθέσεις.

Όλες οι παραπάνω λειτουργικές απαιτήσεις, χρειάστηκαν επεμβάσεις τόσο σε επίπεδο ανάπτυξης όσο και σχεδίασης του υπάρχοντος λογισμικού για να έρθουν εις πέρας.

2.1 Σύνδεση από μέλη του Χαροκοπείου πανεπιστημίου

Αρχικά, η διαδικασία της ανάλυσης προηγείται της διαδικασίας υλοποίησης μιας λειτουργικής απαίτησης. Σαν πρώτο βήμα, θα ορίσουμε περιγραφικά το σενάριο χρήσης της σύνδεσης από μέλη του Χαροκοπείου πανεπιστημίου.

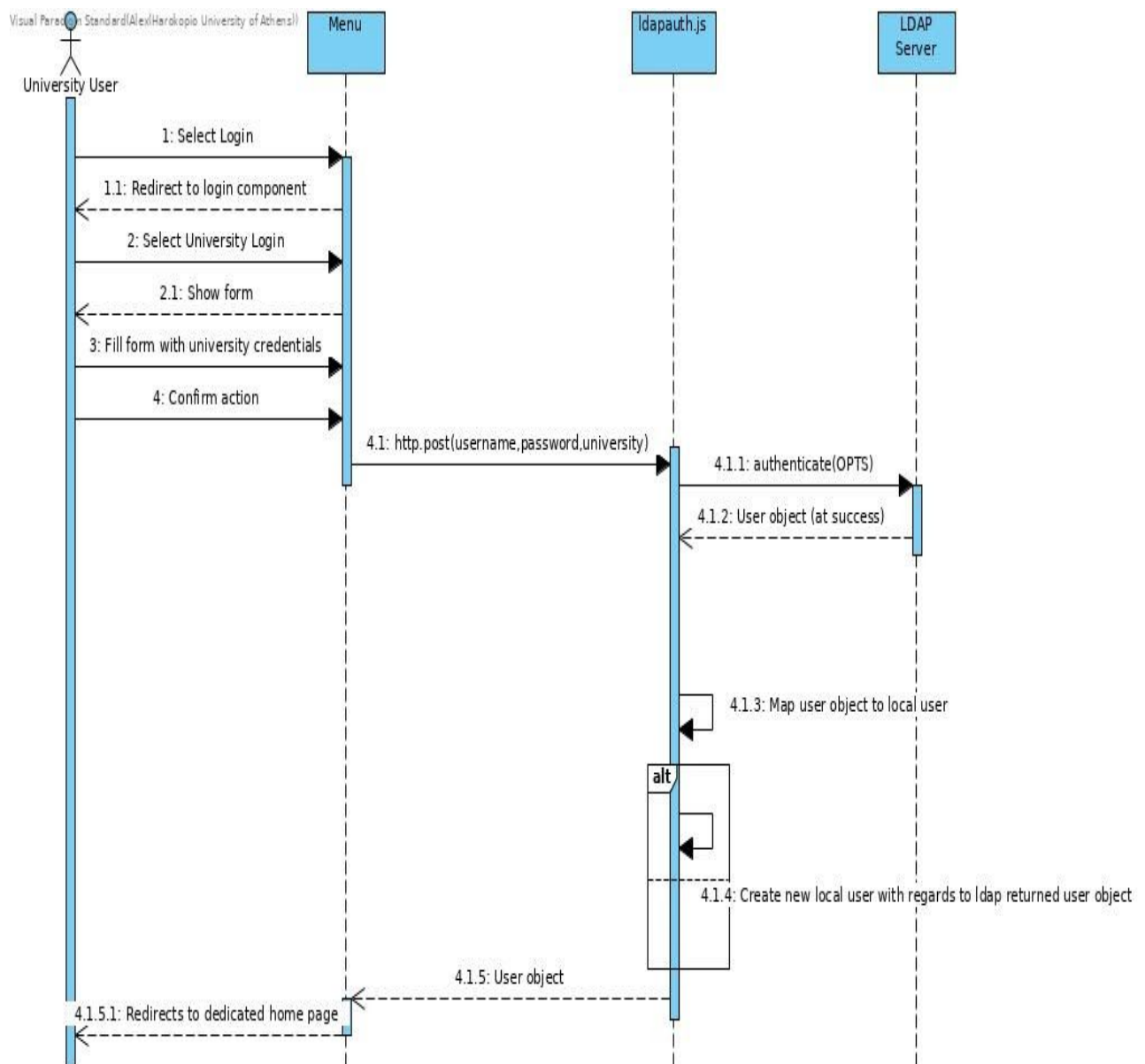
Use Case	Σύνδεση από μέλη του Χαροκοπείου πανεπιστημίου	
Brief Description	Οι χρήστες που ανήκουν στο Χαροκόπειο πανεπιστήμιο μπορούν να συνδεθούν στο σύστημα χρησιμοποιώντας τα στοιχεία του πανεπιστημιακού λογαριασμού τους.	
Actors	Φοιτητής, Καθηγητής, Γραμματεία	
Normal Flows	Tasks	Data
	Ο χρήστης επιλέγει Login	selection
	Ο χρήστης επιλέγει University Login	selection
	Ο χρήστης συμπληρώνει τα στοιχεία του	username, email
	Το σύστημα επικοινωνεί με τον LDAP server του πανεπιστημίου	bindDN, password
	Το σύστημα αυθεντικοποιεί τον χρήστη	username, email
	Ο LDAP server επιστρέφει το αντικείμενο του χρήστη	mail', 'givenName', 'sn', 'title', 'company'
	Το σύστημα αποθηκεύει τον χρήστη τοπικά στην βάση	_id, email, role, name, lastname, university
	Ο χρήστης ανακατευθύνεται στην αντίστοιχη σελίδα	token

Πιν. 1

Παραπάνω, περιγράφεται αναλυτικά το σενάριο χρήσης της σύνδεσης μελών του πανεπιστημίου στο σύστημα.

Όσον αφορά την τεχνική υλοποίηση, έγινε χρήση της βιβλιοθήκης [passport-ldapauth](#). Η βιβλιοθήκη αυτή, επιτρέπει ουσιαστικά την αυθεντικοποίηση χρησιμοποιώντας το LDAP πρωτόκολλο, ενσωματώνοντας κάποιο middleware που κάνει ακριβώς αυτό. Σημαντικό ρόλο εδώ, έχει ο καθορισμός ενός αντικειμένου διαμόρφωσης (configuration object), το οποίο περιέχει οδηγίες για τη σύνδεση του συστήματος με τον απομακρυσμένο LDAP server.

Για την καλύτερη κατανόηση των δράσεων που θα συμβαίνουν στο σύστημα όταν ένας χρήστης προσπαθεί να συνδεθεί με τα στοιχεία του πανεπιστημίου, μπορούμε να στραφούμε στο αντίστοιχο διάγραμμα ροής:



2.2 Εξωτερικός συνεργάτης ως μέλος τριμελούς

Στην πρώτη έκδοση της εφαρμογής, ο ρόλος του εξωτερικού συνεργάτη είναι περιορισμένος στην πρόταση θέματος εργασίας και μόνο. Δεν είναι λίγες οι φορές όμως, που καθηγητές πανεπιστημίου επιλέγουν να συνεργάζονται με εξωτερικούς συναδέλφους στην επίβλεψη εργασιών. Επίσης, θα έλεγε κανείς ότι είναι λογικό, στην περίπτωση που κάποιος εξωτερικός συνεργάτης προτείνει ένα θέμα εργασίας το οποίο εγκριθεί από έναν καθηγητή, να μπορεί να επιβλέπει και την πρόοδο του. Παρόλα αυτά, δεν πρέπει να δίνεται σε όλους τους χρήστες η δυνατότητα να επιβλέπουν εργασίες.

Λαμβάνοντας υπόψη τα παραπάνω, αποφασίστηκε να προστεθεί ένας επιπλέον ρόλος στον ήδη υπάρχον μοντέλο του εξωτερικού συνεργάτη. Ο νέος ρόλος έχει τίτλο “External-Professor” ή αλλιώς εξωτερικός καθηγητής και περιέχει όλες τις λειτουργίες ενός εξωτερικού συνεργάτη συν τη δυνατότητα να μπορεί να δέχεται αιτήματα και να επιβλέπει εργασίες όπως οι καθηγητές (Professors).

Όπως έχει αναφερθεί στη προηγούμενη ενότητα, όλοι οι εξωτερικοί συνεργάτες πρέπει να εγγραφούν στο σύστημα και μετά να εγκριθεί ο λογαριασμός τους από τον διαχειριστή του συστήματος (admin). Παρομοίως και σε αυτή τη περίπτωση, οι εξωτερικοί καθηγητές θα κάνουν εγγραφή στο σύστημα ως εξωτερικοί συνεργάτες και ο διαχειριστής (admin) θα μπορεί να αλλάξει τον ρόλο τους σε αυτόν του εξωτερικού καθηγητή (External-Professor). Σημειώνεται, πως αν ο διαχειριστής προσπαθήσει να αλλάξει τον ρόλο του ίδιου ή κάποιου άλλου διαχειριστή θα εμφανιστεί μήνυμα λάθους ώστε να αποτραπεί η απουσία ενός διαχειριστή από το σύστημα.

Για να γίνουν όλα αυτά, έγιναν αλλαγές στο σχέδιο της βάσης δεδομένων του συστήματος αλλά και σε διάφορα άλλα σημεία που έπρεπε να επιτρέπεται η πρόσβαση εξωτερικών συνεργατών σε λειτουργίες καθηγητών. Επίσης, προέκυψαν δύο (2) νέα σενάρια χρήσης. Πρώτον, μέχρι πρότινος οι καθηγητές μπορούσαν να προτείνουν μόνο άλλους καθηγητές για να συμμετέχουν στην τριμελή επιτροπή μιας εργασίας, ενώ τώρα θα πρέπει να μπορούν να προσκαλέσουν και εξωτερικούς καθηγητές αντίστοιχα. Δεύτερον, πρέπει ο διαχειριστής του συστήματος να μπορεί να αλλάξει το ρόλο του εξωτερικού συνεργάτη σε εξωτερικό καθηγητή έτσι ώστε να μπορεί ο δεύτερος να έχει πρόσβαση στις επιπλέον λειτουργίες.

Το πρώτο, περιγράφεται πιο απλά ως εξής:

Use Case	Πρόταση εξωτερικών καθηγητών για συμμετοχή στην τριμελή επιτροπή	
Brief Description	Οι καθηγητές θα μπορούν να προσκαλέσουν εξωτερικούς καθηγητές να συμμετάσχουν στην τριμελή επιτροπή και να επιβλέπουν την πρόοδο της εργασίας	
Actors	Καθηγητής	
Normal Flows	Tasks	Data
	Ο χρήστης επιλέγει Assigned Theses	selection
	Ο χρήστης επιλέγει εξωτερικό καθηγητή που θέλει να προσκαλέσει από drop-down menu	selection
	Ο χρήστης επιβεβαιώνει το αίτημα	Assigned_thesis object, professor_id

Πιν. 2

Για το παραπάνω, φτιάχτηκαν όσες γραφικές διεπαφές χρειαζόνταν και ένας επιπλέον GET controller, που χρειαζόταν για να εμφανίζει το σύνολο των εξωτερικών καθηγητών.

Το δεύτερο σενάριο χρήσης θα αναλυθεί περαιτέρω αμέσως μετά.

2.3 Διαχείριση ρόλων

Κατά την διαδικασία ανάπτυξης της εφαρμογής, παρατηρήθηκε πως οι ρόλοι των μελών του πανεπιστημίου όπως είναι καταγεγραμμένοι στον LDAP server έχουν πληθώρα διαφορετικών τιμών. Για παράδειγμα, ένας καθηγητής πανεπιστημίου μπορεί να έχει τον τίτλο του Αναπληρωτή καθηγητή, Ομότιμου καθηγητή κα. Μπορεί ακόμη και να αλλάξει ο τίτλος του στην πάροδο του χρόνου. Παράλληλα, ένας συνεργάτης του πανεπιστημίου δύναται να έχει

τελείως διαφορετικό τίτλο στον κατάλογο χρηστών αλλά να εκτελεί χρέη καθηγητή. Εντούτοις, κρίνεται αναγκαία η δυνατότητα διαχείρισης ρόλων στο σύστημα. Η λειτουργικότητα αυτή θα προστεθεί στα σενάρια χρήσης του διαχειριστή του συστήματος (admin).

Σημειώνεται ότι το σύστημα μπορεί να αναγνωρίζει και να αντιστοιχεί αυτόματα με τοπικούς ρόλους, τους παρακάτω τίτλους μελών του πανεπιστημίου:

LDAP Title	TMS Role
Προπτυχιακός Φοιτητής	Student
Επίκουρος Καθηγητής	Professor
Καθηγητής	Professor
Λέκτορας	Professor
Ομότιμος Καθηγητής	Professor
Αναπληρωτής Καθηγητής	Professor
Διοικητικό Προσωπικό	Secretariat

Εικ. 5

Το σενάριο χρήσης της ενημέρωσης ρόλου θα προστεθεί στις λειτουργίες του διαχειριστή για τους πανεπιστημιακούς χρήστες αλλά και για τους εξωτερικούς συνεργάτες. Ο διαχειριστής μπορεί να αλλάξει τους φοιτητές, καθηγητές, γραμματείς μεταξύ τους και τους εξωτερικούς συνεργάτες σε εξωτερικούς καθηγητές αντίστοιχα.

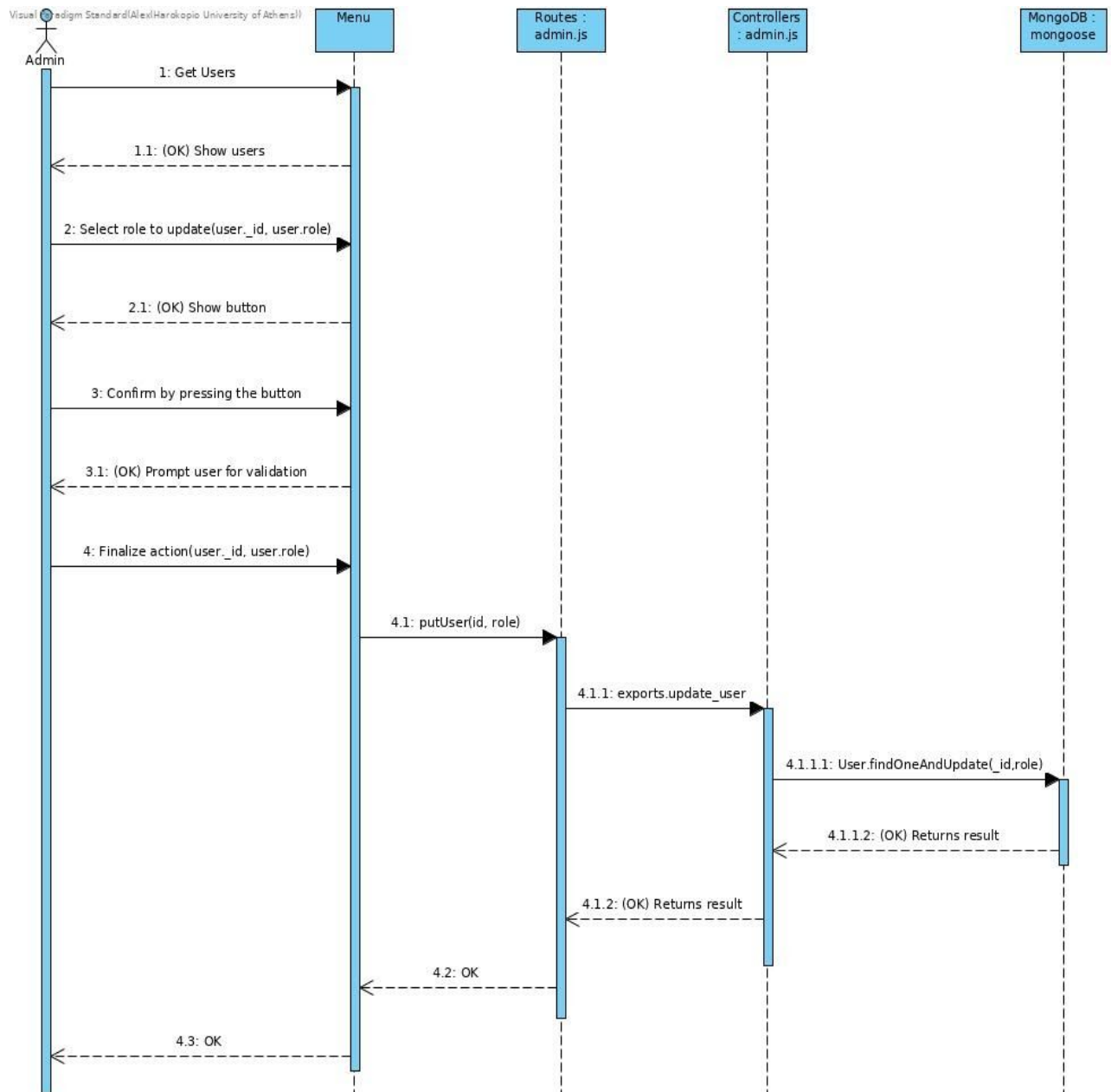
Use Case	Ενημέρωση ρόλου χρήστη συστήματος	
Brief Description	Ο διαχειριστής του συστήματος (admin) θα μπορεί να αλλάξει τον ρόλο των πανεπιστημιακών χρηστών και των εξωτερικών χρηστών κατά βούληση.	
Actors	Διαχειριστής συστήματος (Admin)	
Normal Flows	Tasks	Data
	Ο χρήστης κάνει είσοδο στο σύστημα ως διαχειριστής	
	Ο χρήστης επιλέγει Get Users ή Get Externals	selection
	Στην γραμμή του ζητούμενου χρήστη στην αντίστοιχη στήλη "Role" επιλέγει τον επιθυμητό ρόλο για ανανέωση από αφιερωμένο drop-down menu	user._id, user.role
	Ο χρήστης οριστικοποιεί την ενέργεια με το πάτημα του κουμπιού "Update Role"	selection
	Το σύστημα ρωτάει τον χρήστη για οριστικοποίηση του αιτήματος	selection
	Το σύστημα μέσω Rest API (put request) ενημερώνει την αλλαγή στην βάση. (Express + mongoose framework)	userId, role

	Ο χρήστης ανακατευθύνεται στην κεντρική σελίδα.	
--	---	--

Πιν.3

Παραπάνω περιγράφεται το σενάριο χρήσης της ενημέρωσης ρόλου των χρηστών από το διαχειριστή.

Το αντίστοιχο διάγραμμα ροής είναι το εξής:



Εικ.6

2.4 Αρχείο αναθέσεων

Στα πλαίσια της διαδικασίας διαχείρισης των πτυχιακών εργασιών στο Χαροκόπειο πανεπιστήμιο, όπως έγινε σαφές, η γραμματεία αποτελεί τον κύριο μεσολαβητή σε όλα τα βήματα. Παράλληλα, στα καθήκοντα τους περιλαμβάνεται η σύνταξη αρχείων, αρχικά με τα διαθέσιμα θέματα και τελικά με το σύνολο των αναθέσεων της εκάστοτε περιόδου. Η πρώτη εργασία πλέον δεν χρήζει ύπαρξης, αφού οι καθηγητές μπορούν απευθείας να ανεβάσουν τα θέματα τους στην πλατφόρμα. Από την άλλη, κρίνεται σκόπιμο να αναπτυχθεί εκ νέου μια λειτουργία για την γραμματεία που να αυτοματοποιεί και την δεύτερη εργασία.

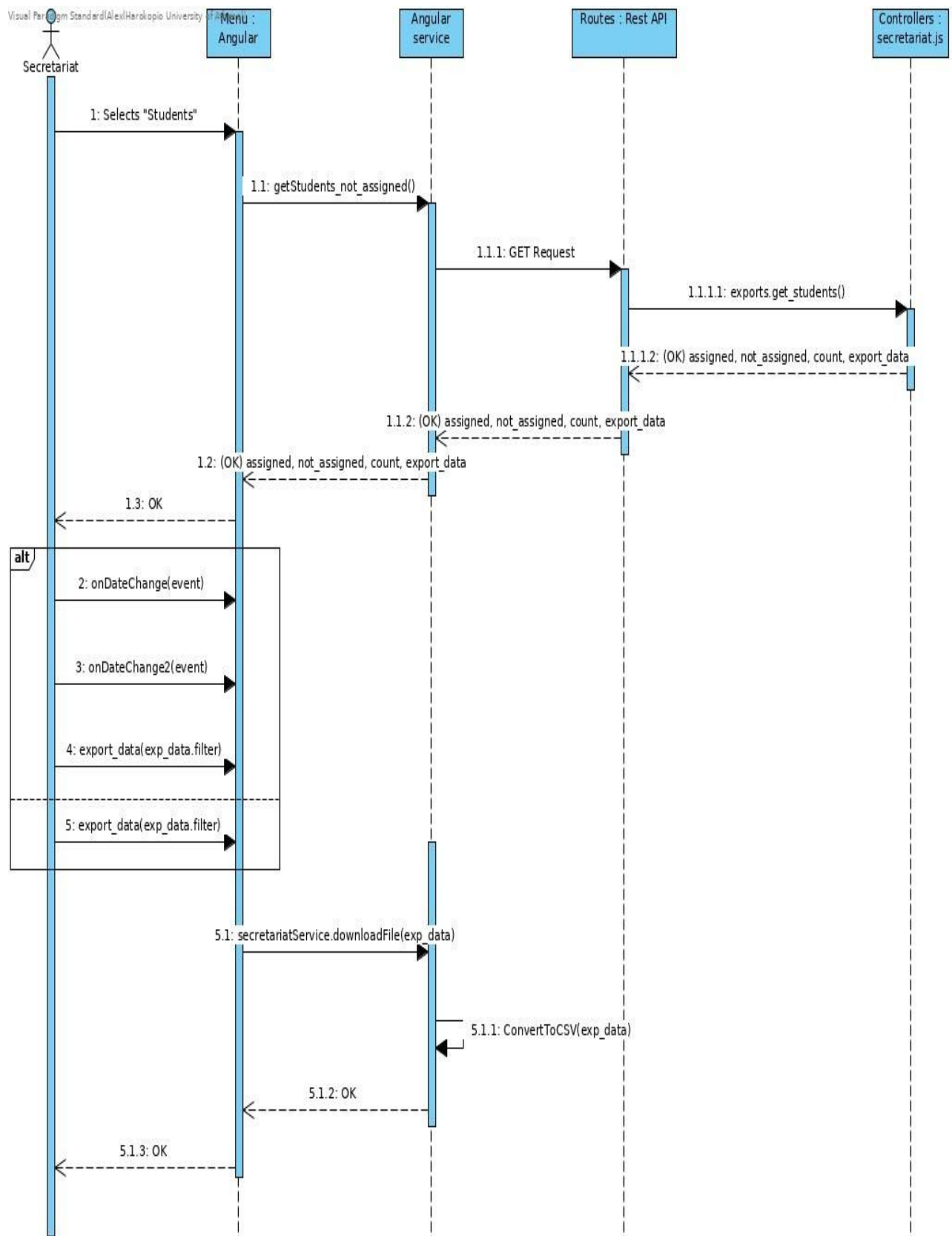
Το σενάριο χρήσης που προκύπτει λοιπόν, είναι το εξής:

Use Case	Κατέβασμα εγγράφου αναθέσεων	
Brief Description	Η γραμματεία θα έχει την δυνατότητα να κατεβάζει το έγγραφο που περιέχει τις αναθέσεις σε μορφή .csv καθώς και να επιλέγει επιθυμητή περίοδο των αναθέσεων αυτών	
Actors	Γραμματεία	
Normal Flows	Tasks	Data
	Ο χρήστης κάνει είσοδο στο σύστημα ως γραμματεία	
	Ο χρήστης επιλέγει το μενού "Students"	selection
	Ο χρήστης επιλέγει ημερομηνία ανάθεσης "από" και "μέχρι" από τα αντίστοιχα inputs.	from_date, to_date
	Ο χρήστης οριστικοποιεί την ενέργεια πατώντας το κουμπί "Export"	exp_data

Πιν. 4

Ενώ η περιγραφή του σεναρίου χρήσης είναι απλή, η τεχνική υλοποίηση είναι περισσότερο πολύπλοκη. Για τη βελτιστοποίηση όσον αφορά τον χρόνο απόκρισης, αποφασίστηκε να δημιουργούνται τα δεδομένα του αρχείου σε πρώτη φάση στο backend (nodejs), σε μορφή json. Έπειτα, το frontend (Angular) θα είναι υπεύθυνο για το φιλτράρισμα (κατά ημερολογιακή περίοδο), τη μετατροπή του τύπου του αρχείου, από json σε csv και τέλος για τη εντολή του download στο σύστημα αρχείων του χρήστη.

Πιο ξεκάθαρα μπορούμε να το δούμε εδώ:



Εικ. 7

3 Λειτουργίες προγραμματιστή και Εγκατάσταση λογισμικού σε server

Για το σύνολο των λειτουργιών του προγραμματιστή, επιλέχθηκε καθολικά η χρήση της πλατφόρμας Docker. Καθώς η πολυπλοκότητα στην αλληλεπίδραση μεταξύ των τεχνολογιών που χρησιμοποιούνται στην εφαρμογή είναι αυξημένη, το Docker απλοποιεί και επιταχύνει την ροή εργασίας, απομονώνοντας όλες τις επιμέρους διεργασίες της εφαρμογής σε containers (δοχεία). <https://www.docker.com/why-docker>

Τα containers, είναι τυπικές μονάδες λογισμικού που συσκευάζουν κώδικα, έτσι ώστε μία εφαρμογή να εκτελείται γρήγορα και αξιόπιστα από το ένα υπολογιστικό περιβάλλον στο άλλο. Μια Docker εικόνα (image) είναι ένα ελαφρύ, αυτόνομο, εκτελέσιμο πακέτο λογισμικού που περιλαμβάνει όλα όσα χρειάζονται για την εκτέλεση μιας εφαρμογής: κώδικα, runtime, εργαλεία συστήματος, βιβλιοθήκες συστήματος και ρυθμίσεις. <https://www.docker.com/resources/what-container>

3.1 docker-compose

Σε πρώτη φάση, χρησιμοποιήθηκε το εργαλείο Compose για τη διαδικασία του containerization της εφαρμογής. Η αναφερόμενη διαδικασία έχει ουσιαστικά τρία (3) βήματα δράσεων:

1. Ο καθορισμός του περιβάλλοντος της εφαρμογής με Dockerfiles, έτσι ώστε να μπορεί να επαναχρησιμοποιηθεί οπουδήποτε αλλού.

Στο TMS, προστέθηκε ένα Dockerfile στο backend για τη δημιουργία του NodeJs container (“backend”) και ένα στο front για το containerization της Angular πλευράς (“frontend”). Υπάρχουν επιπλέον σαφείς οδηγίες σε μορφή σχολίων στα αρχεία αυτά.

2. Ο καθορισμός των services (υπηρεσιών) οι οποίες συντελούν την εφαρμογή, σε ένα αρχείο με τίτλο docker-compose.yml, έτσι ώστε να μπορούν να τρέξουν μαζί σε απομονωμένο περιβάλλον.

Στην εφαρμογή προστέθηκε το αρχείο αυτό στο πρώτο επίπεδο του συστήματος αρχείων. Μέσα καθορίζονται τρεις υπηρεσίες: αυτή του node-app που περιγράφει το backend service, της angular-app που περιγράφει το frontend service και τέλος το mongo service που δίνει εντολή στην πλατφόρμα του

Docker να αναζητήσει την επίσημη εικόνα της mongodb από το docker hub χωρίς να χρειάζεται να καθοριστεί ξεχωριστά. Επιπλέον, υπάρχει η δυνατότητα χρήσης των volumes, τα οποία χρησιμοποιούνται για την διατήρηση των δεδομένων των containers.

3. Τέλος, αφού έχουν ολοκληρωθεί τα προηγούμενα βήματα με την βοήθεια της γραμμής εντολών, το Compose ξεκινά και τρέχει ολόκληρη την εφαρμογή.

Πιο συγκεκριμένα, οι εντολές που χρειάζονται κατά σειρά προτεραιότητας είναι:

- **\$ docker-compose build**

Την πρώτη φορά που τρέχουμε την εντολή αυτή σε ένα νέο περιβάλλον χρειάζεται κάμποση ώρα αφού κατεβάζει τα απαραίτητα στοιχεία για να λειτουργήσει η εφαρμογή. Για τις επόμενες φορές κατατρέπει στην μνήμη cache και μόνο όταν εντοπίσει νέα εισαγωγή κατεβάζει εκ νέου κάποιο πακέτο.

- **\$ docker-compose up**

Με την εντολή αυτή, το Compose εργαλείο τρέχει την εφαρμογή. Σημειώνεται, ότι με την παραπάνω εντολή η εφαρμογή θα τρέξει στην ίδια διεργασία, ενώ αν θέλουμε να τρέξει σε ξεχωριστή μπορούμε να προσθέσουμε την παράμετρο -d.

(docker-compose up -d)

- **\$ docker-compose down**

Αν θέλουμε να κατεβάσουμε την εφαρμογή για να την επανεκκινήσουμε, αυτή είναι η εντολή.

Όλα τα παραπάνω, έχουν υλοποιηθεί και προστεθεί στα πλαίσια της νέας έκδοσης της εφαρμογής και βρίσκονται εδώ: <https://github.com/eellak/gsoc2019-tms>.

3.2 Εγκατάσταση και λειτουργία εφαρμογής σε απομακρυσμένο server

Έπειτα από τη διαδικασία της ανάπτυξης της εφαρμογής στο περιβάλλον του προγραμματιστή, ακολουθεί η διαδικασία εγκατάστασης της σε άλλο περιβάλλον με σκοπό να δοκιμαστεί σε παραπάνω από ένα μηχάνημα.

Αρχικά, βασική προϋπόθεση για την λειτουργία της εφαρμογής σε απομακρυσμένο μηχάνημα εκτός του τοπικού δικτύου, είναι το μηχάνημα αυτό να έχει ήδη εγκατεστημένο το Docker και βεβαίως να έχει ανοικτά/διαθέσιμα τα ports που έχουμε ήδη καταγράψει στις οδηγίες που βρίσκονται στο αρχείο docker-compose.yml.

Παρακάτω, προτείνονται δύο τρόποι που εκμεταλλεύονται το εργαλείο Compose για την εγκατάσταση εφαρμογών σε απομακρυσμένους server, που τρέχουν βεβαίως το Docker engine στο εσωτερικό τους.

Σημειώνεται, ότι και στις δύο περιπτώσεις πρέπει να χρησιμοποιείται το SSH Protocol (Secure Shell) για την ασφαλή πρόσβαση στον απομακρυσμένο server. Το SSH, αποτελεί μια από τις ασφαλέστερες μεθόδους προσπέλασης υπολογιστή από υπολογιστή, αφού η κίνηση (traffic) μεταξύ των δύο υπολογιστών κρυπτογραφείται (σύμφωνα με το AES), αλλά και το ίδιο το πρωτόκολλο περιλαμβάνει έναν μηχανισμό που διασφαλίζει την ακεραιότητα των δεδομένων, χρησιμοποιώντας αλγόριθμους κατακερματισμού, όπως ο SHA-12.

1. Χειροκίνητα, αντιγράφοντας όλα τα αρχεία που συντελούν την εφαρμογή και χρησιμοποιώντας το docker-compose για την λειτουργία της.

Πιο συγκεκριμένα, σαν πρότυπο της ροής των εργασιών:

- **\$ scp -r /complete/local/path/to/src user@remotehost:/path/to/src**

Εδώ, ουσιαστικά αντιγράφουμε τον κώδικα της εφαρμογής στο απομακρυσμένο μηχάνημα.

- **\$ ssh user@remotehost (for default ssh port 22)**

Συνδεόμαστε μέσω SSH στον remote server.

- **\$ pip install docker-compose**

Εγκαθιστούμε στο μηχάνημα το εργαλείο Compose.

- **\$ cd /path/to/src/app-name**

Μεταφερόμαστε στη τοποθεσία της εφαρμογής στο μηχάνημα.

- **\$ docker-compose build**

Χτίζουμε την εφαρμογή.

- **\$ docker-compose up -d**

Και την τρέχουμε σε ξεχωριστή διεργασία.

Σημειώνεται ότι για κάθε αλλαγή στον κώδικα της εφαρμογής πρέπει να επαναληφθεί η παραπάνω διαδικασία.

2. Χρησιμοποιώντας τα docker contexts

Τα docker contexts είναι ένας μηχανισμός που παρέχει πληροφορίες σύνδεσης στα endpoints του Docker API και τις αποθηκεύει για μελλοντική χρήση.

- **\$ docker context create hua_remote - - docker "host=ssh://user@remotemachine"**

Με την παραπάνω εντολή δημιουργούμε ένα νέο context με όνομα hua_remote που θα κρατάει το μονοπάτι σύνδεσης με το απομακρυσμένο μηχάνημα.

- **\$ docker-compose - - context hua_remote up -d**

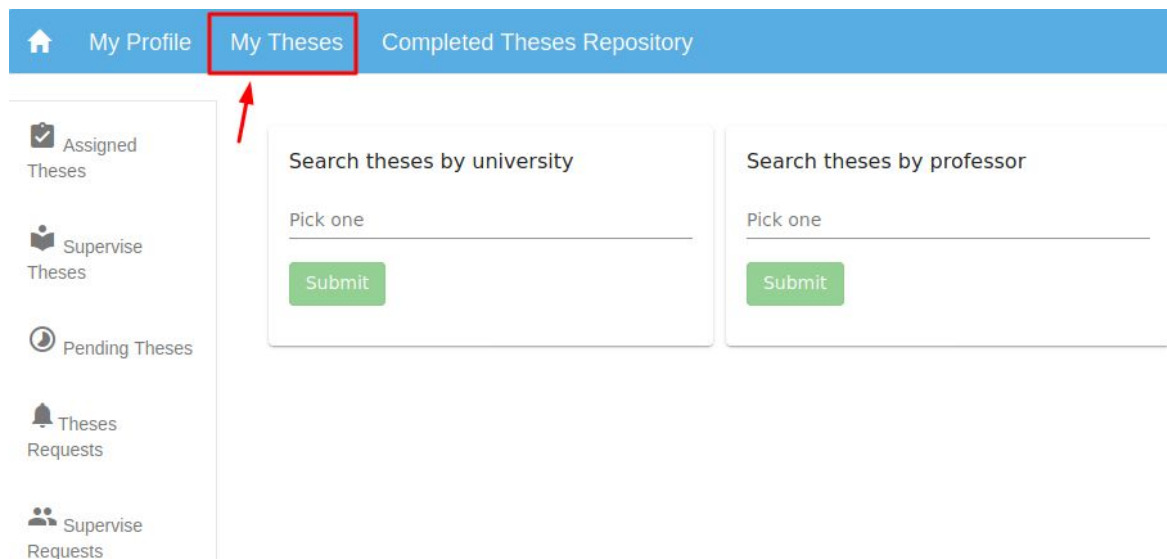
Έτσι, μπορούμε να χρησιμοποιήσουμε το Compose, υποδεικνύοντας το context που μόλις δημιουργήσαμε για να τρέξει η εφαρμογή στο απομακρυσμένο μηχάνημα.

4 Εγχειρίδιο βασικών σεναρίων χρήσης

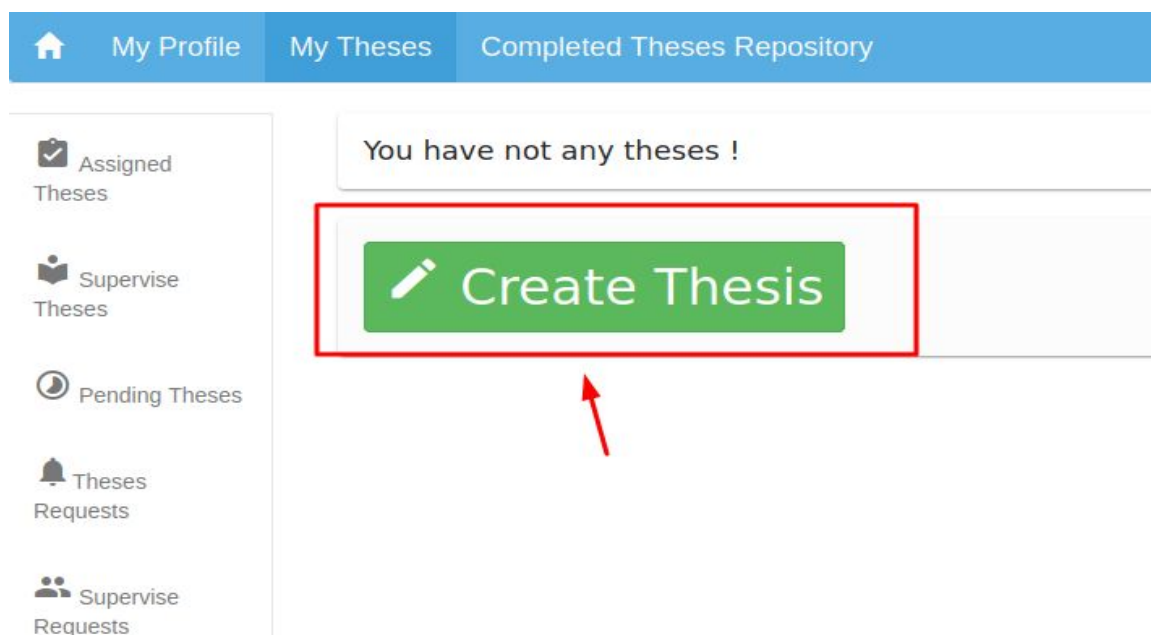
Στην παρούσα ενότητα, θα γίνει επίδειξη μερικών σεναρίων χρήσης της εφαρμογής με εικόνες, ώστε να γίνουν κατανοητές ορισμένες λειτουργίες της πλατφόρμας από την πλευρά του χρήστη. Μπορούν να χρησιμοποιηθούν ως *εγχειρίδιο χρήσης* των βασικότερων λειτουργιών της πλατφόρμας. Κάθε σενάριο χρήσης θα έχει αντίστοιχη λεζάντα που θα εξηγεί την περιγραφή του σεναρίου και τον ρόλο του δρώντα.

4.1 Δήλωση θέματος εργασίας (Professor)

Για να δηλώσουμε νέο Θέμα εργασίας ως καθηγητής (Professor), από το οριζόντιο μενού επιλέγουμε **“My Theses”**, έπειτα **“Create Thesis”**, συμπληρώνουμε τα αντίστοιχα πεδία με τις πληροφορίες της εργασίας και οριστικοποιούμε με το κουμπί **“Create”**.



Εικ. 8



Εικ. 9

Create a new Thesis
Please complete the list below in order to create a new thesis. You can also upload a pdf file of your thesis.

Title
Εγκατάσταση και λειτουργία εφαρμογής διαχείρισης πτυχιακών εργασιών στο τμήμα Πληροφορικής και Τηλεματικής

Prerequisites
Software engineering, DevOps

Description
Εγκατάσταση και λειτουργία εφαρμογής διαχείρισης πτυχιακών εργασιών στο τμήμα Πληροφορικής και Τηλεματικής

Tags
Software engineering, DevOps, TMS

Choose Pdf File
Browse... No file selected.

Create **Cancel**

Εικ. 10

4.2 Ανάθεση θέματος εργασίας

Student(Φοιτητής)

Ο φοιτητής από το **κάθετο μενού** μπορεί να επιλέξει **“Available Theses”**, για να του εμφανιστούν οι διαθέσιμες εργασίες και αφού επιλέξει αυτή που τον ενδιαφέρει μπορεί να κάνει **“Apply”** γράφοντας ένα σύντομο κείμενο που θα δει ο καθηγητής.

Available Theses Results

Created Time	T Title	University	Professor	Prerequisites	Apply
Jul 1, 2020	Εγκατάσταση και λειτουργία εφαρμογής διαχείρισης πτυχιακών εργασιών στο τμήμα Πληροφορικής και Τηλεματικής	Χαροκόπειο Πανεπιστήμιο	Αλέξανδρος Καρβουνιάρης	Software engineering, DevOps	+

Εικ. 11

Professor(Καθηγητής)

Έπειτα, ο **καθηγητής** από το **κάθετο μενού** μπορεί να επιλέξει **“Theses Requests”** για να δει τυχόν αιτήματα από φοιτητές και να τα δεχτεί πατώντας το **“Apply”**.

My Profile My Theses Completed Theses Repository

Assigned Theses
Supervise Theses
Pending Theses
Theses Requests
Supervise Requests

Requests for Theses Results

Requests from students to your theses

Created Time	T Title	Applied Student	Prerequisites	Message from student	Accept	Reject
Jul 1, 2020	Εγκατάσταση και λειτουργία εφαρμογής διαχείρισης πτυχιακών εργασιών στο τμήμα Πληροφορικής και Τηλεματικής	Νικόλαος Νταντής	Software engineering, DevOps	Ενδιαφέρομαι για αυτό το θέμα.	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Εικ. 12
Student(Φοιτητής)

Ο φοιτητής τώρα, μπορεί από το **κάθετο μενού**, να επιλέξει **“My Requests for Thesis”** και να επιβεβαιώσει την ανάθεση του σε αυτό το θέμα, πατώντας **“Confirm Thesis”**.

My Profile My Thesis Completed Theses Repository

Available Theses
My Requests for Thesis
Propose Thesis

You have applied to these theses:

Created Time	T Title	Professor	Text	Accepted from Professor
Jul 1, 2020	Εγκατάσταση και λειτουργία εφαρμογής διαχείρισης πτυχιακών εργασιών στο τμήμα Πληροφορικής και Τηλεματικής	Αλέξανδρος Καρβουνιάρης	Ενδιαφέρομαι για αυτό το θέμα.	<input checked="" type="button" value="Confirm Thesis"/>

Εικ. 13

Έτσι ολοκληρώνεται η διαδικασία της ανάθεσης.

4.3 Ορισμός Τριμελούς (Professor & External-Professor)

Professor

Για να οριστεί η τριμελής επιτροπή σε μια εργασία, ο **καθηγητής** από το **κάθετο μενού** να επιλέξει **“Assigned Theses”** και να επιλέξει από τα αντίστοιχα μενού τους συνεργάτες που επιθυμεί και να τους προσκαλέσει, πατώντας **“Propose Supervisor”**.

[My Profile](#)
[My Theses](#)
[Completed Theses Repository](#)

Assigned Theses

Supervise Theses

Pending Theses

Theses Requests

Supervise Requests

Assigned Theses Results

Assigned Time	Title	Student Assigned	Prerequisites	Drafts	Supervisors
Jul 1, 2020	Εγκατάσταση και λειτουργία εφαρμογής διαχείρισης πτυχιακών εργασιών στο τμήμα Πληροφορικής και Τηλεματικής	Νικόλαος Νταντής	Software engineering, DevOps		<div>University Supervisors</div> <div>University Professor</div> <div>External Supervisors</div> <div>External Professor</div> <div>Αγγελική Σπανοπ...</div> <div>Propose Supervisor</div>

Εικ. 14

External-Professor

Τότε, ο External-Professor από το κάθετο μενού επιλέγει “Supervise Requests” αποδέχεται την πρόσκληση και πλέον μπορεί να επιβλέπει την εργασία από την επιλογή “Supervise Theses” στο κάθετο μενού.

[My Profile](#)
[Completed Theses Repository](#)

My Theses

Create Thesis

Supervise Theses

Supervise Requests

Requests for Supervisor Results

Created Time	Title	Professor	Student	Message from professor	Accept	Reject
Jul 1, 2020	Εγκατάσταση και λειτουργία εφαρμογής διαχείρισης πτυχιακών εργασιών στο τμήμα Πληροφορικής και Τηλεματικής	Αλέξανδρος Καρβουνιάρης	Νικόλαος Νταντής	Πρόσκληση για εργασ		

Εικ. 15

[My Profile](#)
[Completed Theses Repository](#)

My Theses

Create Thesis

Supervise Theses

Supervise Requests

Supervise Theses Results

You are supervisor in these theses:

Assigned Time	Title	Professor	Student Assigned	Prerequisites	Drafts
Jul 1, 2020	Εγκατάσταση και λειτουργία εφαρμογής διαχείρισης πτυχιακών εργασιών στο τμήμα Πληροφορικής και Τηλεματικής	Αλέξανδρος Καρβουνιάρης	Νικόλαος Νταντής	Software engineering, DevOps	

Εικ. 16

Η ίδια διαδικασία μπορεί να ακολουθηθεί κι από οποιοδήποτε καθηγητή του πανεπιστημίου από τα ανάλογα μενού.

Professor

Τέλος, όταν δεχτούν την πρόσκληση τα δύο μέλη, ο καθηγητής έχει την παρακάτω εικόνα για την εργασία και δεν μπορεί να προσκαλέσει παραπάνω συνεργάτες.

Assigned Theses Results

Assigned Time	Title	Student Assigned	Prerequisites	Drafts	Supervisors
Jul 1, 2020	Εγκατάσταση και λειτουργία εφαρμογής διαχείρισης πτυχιακών εργασιών στο τμήμα Πληροφορικής και Τηλεματικής	Νικόλαος Νταντής	Software engineering, DevOps		<ul style="list-style-type: none">Αγγελική ΣπανοπούλουΚωνσταντίνος Παπαδόπουλος

Εικ. 17

4.4 Κατέβασμα εγγράφου αναθέσεων (Secretariat)

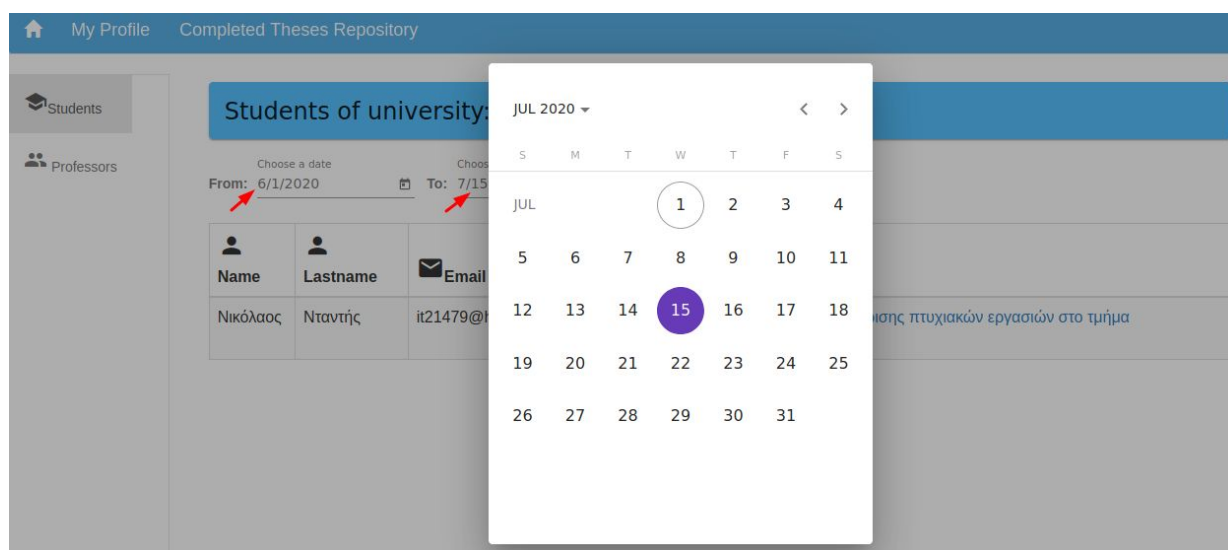
Από την πλευρά της, η **γραμματεία** μπορεί πλέον να επιλέξει από το **κάθετο μενού “Students”**, να επιλέξει την ημερολογιακή περίοδο που επιθυμεί συμπληρώνοντας τις ημερομηνίες με ταμπέλες **From**, **To** και να ολοκληρώσει τη διαδικασία πατώντας το κουμπί **“Export”**.

Students of university:

From: Choose a date To: Choose a date Export

Name	Lastname	Email	Thesis Assigned
Νικόλαος	Νταντής	it21479@hua.gr	Εγκατάσταση και λειτουργία εφαρμογής διαχείρισης πτυχιακών εργασιών στο τμήμα Πληροφορικής και Τηλεματικής

Εικ. 18



Εικ. 19

Το έγγραφο αποθηκεύεται σε επιλεγμένη τοποθεσία στο σύστημα αρχείων του χρήστη, με όνομα “**Theses_Assignments.csv**” και μπορεί να εκτυπωθεί από την γραμματαία.

	A	B	C	D	E	F	G	H
1	A/A	ΟΝΟΜΑΤΕΠΩΝΥΜΟ	ΑΜ	ΤΙΤΛΟΣ	ΗΜΕΡΟΜΗΝΙΑ	ΕΠΙΒΛΕΠΩΝ	ΜΕΛΟΣ1	ΜΕΛΟΣ2
2	1	Νικόλαος Νταντής	it21479	Εγκατάσταση και λειτουργία εφαρμογής	Wed Jul 01 2020	Αλέξανδρος Καρβουνιάρης	Αγγελική Σπανοπούλου	Κωνσταντίνος Παπαδόπουλος

Εικ. 20

5 Συμπεράσματα

Στο περιεχόμενο του παρόντος εγγράφου, περιγράφονται αναλυτικά τα βήματα και οι εργασίες που χρειάστηκαν για την εγκατάσταση και λειτουργία πιλοτικής εφαρμογής, για την διαχείριση πτυχιακών εργασιών στο Χαροκόπειο πανεπιστήμιο. Πλέον, μπορεί οποιοδήποτε μέλος του πανεπιστημίου, είτε είναι καθηγητής, φοιτητής ή διοικητικό προσωπικό να συνδεθεί με τα στοιχεία του στην αφοσιωμένη πλατφόρμα. Παράλληλα, ανάλογα με τον ρόλο που διατηρεί στο πανεπιστήμιο, μπορεί να: αναζητεί, προτείνει, αναθέτει, επιβλέπει, εκπονεί, ανεβάζει/κατεβάζει υλικό για εργασίες όλα αυτά σε μια κεντρική τοποθεσία για όλους. Επιπλέον, εξωτερικοί συνεργάτες μπορούν να λάβουν μέρος στη διαδικασία, ενώ ο διαχειριστής του συστήματος υπάρχει για να χειρίζεται σε υψηλό επίπεδο το σύστημα. Τέλος, έχει παρατεθεί εγχειρίδιο χρήσης για τις βασικότερες λειτουργίες της εφαρμογής, βοηθώντας τον χρήστη να νιώθει περισσότερη άνεση κατά τη χρήση της. Μένει μόνο να γίνει πλήρης δοκιμή από πραγματικούς χρήστες, σε πραγματικές συνθήκες για να φανεί η αξία της πλατφόρμας.

5.1 Μελλοντικές εργασίες

Ενώ η πλατφόρμα είναι πλήρως λειτουργική, σειρά έχει η εισαγωγή περισσότερων πανεπιστημιακών οργανισμών στο σύστημα. Για τον σκοπό αυτό, πρέπει να δημιουργηθεί ένα νέο αρχείο το οποίο θα περιέχει τα αντικείμενα διαμόρφωσης (configuration objects) για όλα τα εμπλεκόμενα πανεπιστήμια. Έτσι, αντί να καλείται ένα στατικό αντικείμενο με το configuration του Χαροκοπέιου πανεπιστημίου, θα παίρνει σαν όρισμα μια συνάρτηση που θα γυρνάει το επιθυμητό configuration. Όλα αυτά, αν το ζητούμενο πανεπιστήμιο έχει τον κατάλογο χρηστών του σε LDAP server, σε άλλη περίπτωση θα χρειαστεί χρήση άλλης βιβλιοθήκης.

Ακόμη, έχει νόημα η υλοποίηση ενός συστήματος ενημερώσεων, που ο σκοπός του θα είναι να ενημερώνει τον εκάστοτε χρήστη για μία αλλαγή που τον ενδιαφέρει απευθείας στο email του. Εδώ, μπορούμε να χρησιμοποιήσουμε το [Nodemailer Module](#). Είναι εύκολο στη χρήση του και αποτελεί την πλέον διαδεδομένη λύση για αποστολή emails στο NodeJs.

Τέλος, αποτελεί προσωπική δέσμευση του συγγραφέα, να υποστηρίξει μελλοντικές προσπάθειες επέκτασης της εφαρμογής και να διορθώσει ελαττώματα που ίσως προκύψουν, δεδομένου ότι έχει διαθέσιμο χρόνο να το κάνει.

Αναφορές

[1] Von Hippel, Eric. "Learning from open-source software." *MIT Sloan management review* 42.4 (2001): 82-86.

[2] Andersen-Gott, Morten, Gheorghita Ghinea, and Bendik Bygstad. "Why do commercial companies contribute to open source software?." *International journal of information management* 32.2 (2012): 106-117.

[3] Oreg, Shaul, and Oded Nov. "Exploring motivations for contributing to open source initiatives: The roles of contribution context and personal values." *Computers in human behavior* 24.5 (2008): 2055-2073.

[4] Lakhani, Karim R., and Eric Von Hippel. "How open source software works: "free" user-to-user assistance." *Produktentwicklung mit virtuellen Communities*. Gabler Verlag, 2004. 303-339.

[5] Maulik Pandya. "MEAN Stack Architecture: MongoDB, ExpressJS, AngularJS, and NodeJS" <https://evincedev.com/blog/mean-stack-architecture/>