



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΤΗΛΕΜΑΤΙΚΗΣ

ΚΑΤΕΥΘΥΝΣΗ ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΙ ΕΦΑΡΜΟΓΕΣ ΙΣΤΟΥ

Δημιουργία πλατφόρμας για αυτοματοποίηση ενεργειών διαχείρισης χρηστών

Διπλωματική Εργασία

Βελισσάρη Παρασκευά

Αθήνα, 2020



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΤΗΛΕΜΑΤΙΚΗΣ**

ΚΑΤΕΥΘΥΝΣΗ ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΙ ΕΦΑΡΜΟΓΕΣ ΙΣΤΟΥ

Τριμελής Εξεταστική Επιτροπή

Ανάργυρος Τσαδήμας

Ε.Τ.Ε.Π., Πληροφορική και Τηλεματική, Χαροκόπειο Πανεπιστήμιο

Θωμάς Καμαλάκης

**Αναπληρωτής καθηγητής, Πληροφορική και Τηλεματική, Χαροκόπειο
Πανεπιστήμιο**

Βασίλειος Δαλάκας

Ε.ΔΙ.Π., Πληροφορική και Τηλεματική, Χαροκόπειο Πανεπιστήμιο

Ο Βελισσάρης Παρασκευάς

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 προσβάλλει 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.

ΓΝΩΜΙΚΟ

«Τα αγαθά κόποις κτώνται»

Αριστοτέλης,

384 π.Χ. – 322 π.Χ.,

Αρχαίος Έλληνας φιλόσοφος και επιστήμονας

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία εκπονείται στα πλαίσια του Π.Μ.Σ. Πληροφορικής και Τηλεματικής του τμήματος Πληροφορικής και Τηλεματικής, του Χαροκοπείου Πανεπιστημίου.

Με την περάτωση της παρούσας εργασίας, θα ήθελα να ευχαριστήσω τον Ε.Τ.Ε.Π. του τμήματος Πληροφορικής και Τηλεματικής κ. Ανάργυρο Τσαδήμα, καθώς και τον κ. Γεώργιο Συμιακάκη, για την πολύτιμη βοήθειά τους και το καλό κλίμα συνεργασίας που αναπτύξαμε.

Επίσης, θα ήθελα να ευχαριστήσω τον κ. Δαλάκα Βασίλειο, Ε.ΔΙ.Π. του τμήματος Πληροφορικής και Τηλεματικής και τον κ. Καμαλάκη Θωμά, Αναπληρωτή Καθηγητή του τμήματος Πληροφορικής και Τηλεματικής, για τη συμμετοχή τους στην τριμελή εξεταστική επιτροπή της παρούσας εργασίας.

Τέλος, θέλω να ευχαριστήσω την οικογένειά μου για την πνευματική τους στήριξη κατά τη διάρκεια των σπουδών μου.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1.	ΕΙΣΑΓΩΓΗ	11
1.1	Τρέχουσα διαδικασία.....	11
1.2	Ανάγκη δημιουργίας αυτής της εφαρμογής	12
1.3	Γιατί Angular 2+ framework	13
1.4	Γιατί Django framework	13
2.	ΤΕΧΝΟΛΟΓΙΕΣ	14
2.1	JavaScript.....	14
2.2	Angular	15
2.3	TypeScript.....	15
2.4	Bootstrap.....	16
2.5	Angular Material.....	17
2.6	JWT (JSON Web Token)	17
2.7	Python	18
2.8	Django	19
2.9	PostgreSQL	21
2.10	IDEs.....	21
3.	ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ	22
3.1	Use Case diagram	23
3.2	Activity diagram.....	24
3.3	Class diagram.....	26
3.4	Deployment diagram.....	26
4.	ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ	27
4.1	Προαπαιτούμενα.....	28
4.1.1	Git	28
4.1.2	npm και Node.js	28
4.1.3	Python	29
4.1.4	PIP.....	29
4.2	Εγκατάσταση PostgreSQL.....	30
4.2.1	PostgreSQL και dependencies	30
4.3	Εγκατάσταση Angular framework.....	30
4.3.1	Angular	30
4.3.2	Styling	31
4.3.3	ngx-captcha	31
4.3.4	ngx-toastr	32
4.4	Εγκατάσταση Virtualenv	32

4.5	Εγκατάσταση Django framework	32
4.5.1	Ενεργοποίηση Virtualenv	32
4.5.6	python-ldap	38
4.5.7	django-auth-ldap	39
4.5.8	PyJWT	40
4.5.9	psycopg2.....	40
5.	ΟΔΗΓΟΣ ΧΡΗΣΗΣ.....	41
5.1	Landing page	41
5.2	Οθόνη ακύρωσης αιτήματος από το χρήστη	44
5.3	Οθόνη Γραμματείας	46
6.	ΣΥΜΠΕΡΑΣΜΑΤΑ	48
6.1	Επεκτάσεις.....	49
7.	ΒΙΒΛΙΟΓΡΑΦΙΑ - ΑΝΑΦΟΡΕΣ.....	49

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικ 1 Τρέχουσα διαδικασία	12
Εικ 2 Use Case diagram	24
Εικ 3 Activity diagram	25
Εικ 4 Class diagram	26
Εικ 5 Deployment diagram	27
Εικ 6 Landing page	42
Εικ 7 Πληροφορίες συνέχισης διαδικασίας	43
Εικ 8 Υπάρχον ενεργό αίτημα	43
Εικ 9 Νέο αίτημα εντός 24 ωρών από την έκδοση κωδικού	44
Εικ 10 Επιτυχής ακύρωση	44
Εικ 11 Χρήση του ίδιου συνδέσμου 2η φορά	45
Εικ 12 Ακύρωση έπειτα από έκδοση νέου κωδικού	45
Εικ 13 Οθόνη νέου κωδικού	46
Εικ 14 Ο χρήστης έχει πρώτα ακυρώσει το αίτημα	47
Εικ 15 Δεύτερη χρήση ιδίου συνδέσμου	47

Περίληψη

Ο σκοπός της εργασίας ήταν ο σχεδιασμός και η υλοποίηση μιας εύχρηστης web εφαρμογής με τη χρήση του Django framework (Python) και του Angular framework, που θα επιτρέπει σε κάποιον υπάρχοντα χρήστη του Χαροκοπείου Πανεπιστημίου (φοιτητής, μέλος ΔΕΠ, οποιοσδήποτε άλλος χρήστης του Δικτύου) να προβεί σε αίτημα έκδοσης νέου ιδρυματικού κωδικού αυτοματοποιημένα. Αυτή τη στιγμή δεν υφίσταται κάποια παρόμοια εφαρμογή στο Πανεπιστήμιο και η τρέχουσα διαδικασία που ακολουθείται είναι χρονοβόρα.

Το αντικείμενο της εργασίας είναι μια web εφαρμογή, μέσω της οποίας η Γραμματεία ενός τμήματος, θα μπορεί να εκδίδει άμεσα ένα νέο ιδρυματικό κωδικό για κάποιον χρήστη του Δικτύου του Πανεπιστημίου, ο οποίος πρώτα έχει κάνει αίτηση έκδοσης νέου κωδικού, μέσω της ίδιας εφαρμογής.

Η μεθοδολογία που ακολουθήθηκε ήταν η μελέτη των λογισμικών που θα χρησιμοποιούσαμε στην πορεία (Angular 7.2, Python 3.6.1, Django 2.1.5, PostgreSQL 10) είτε διαβάζοντας το επίσημο documentation είτε δοκιμάζοντας λειτουργίες των λογισμικών, δημιουργώντας μικρά projects. Στη συνέχεια προσπαθήσαμε να αφομοιώσουμε τις ανάγκες της εφαρμογής και σε αυτό το σκοπό βοήθησαν πολύ ο κ. Τσαδήμας και ο κ. Συμιακάκης, με τη συμβολή των οποίων έγινε η σχεδίαση του συστήματος. Όταν είχαμε την πρώτη εικόνα για το πώς πρέπει να υλοποιηθεί η εφαρμογή, ξεκίνησε η υλοποίηση.

Το αποτέλεσμα είναι μια σύγχρονη εφαρμογή, που θα διευκολύνει τους χρήστες που χρησιμοποιούν το Δίκτυο του Χαροκοπείου Πανεπιστημίου καθώς και τις Γραμματείες των τμημάτων, όσον αφορά στο αίτημα έκδοσης νέου ακαδημαϊκού κωδικού.

Στα επόμενα κεφάλαια περιγράφονται οι τεχνολογίες που χρησιμοποιήθηκαν, αναλύονται τα βήματα υλοποίησης της εφαρμογής και γίνεται μια παρουσίαση του περιβάλλοντος.

Λέξεις κλειδιά: εφαρμογές ιστού, έκδοση κωδικού, django, angular

Abstract

The purpose of this thesis was to design and implement a user-friendly web application using the Django framework (Python) and the Angular framework, which would allow an existing user of the Harokopio University (student, faculty member, any other Network user) to request a new academic password through an automated process. There is currently no such application at Harokopio University and the current process is time consuming.

The subject of the thesis is a web application through which the Secretariat of a Department will be able to immediately issue a new academic password to a user of the University Network, who has already applied for a new password through the same application.

The methodology followed was the study of the software that would be used in this thesis (Angular 7.2, Python 3.6.1, Django 2.1.5, PostgreSQL 10) either by reading the official documentation or by creating small applications. Then we tried to assimilate the needs of the application and with the help of Mr. Tsadimas and Mr. Simiakakis, the system was designed. When we had the designs ready, the implementation started.

The result is a modern web app that will make the process of issuing a new academic password easier, for users of the Harokopeio University Network, as well as the Secretariats of the Departments.

The following chapters describe the technologies used, the implementation steps and the user interface of the application.

Keywords: web application, new password, django, angular

1. ΕΙΣΑΓΩΓΗ

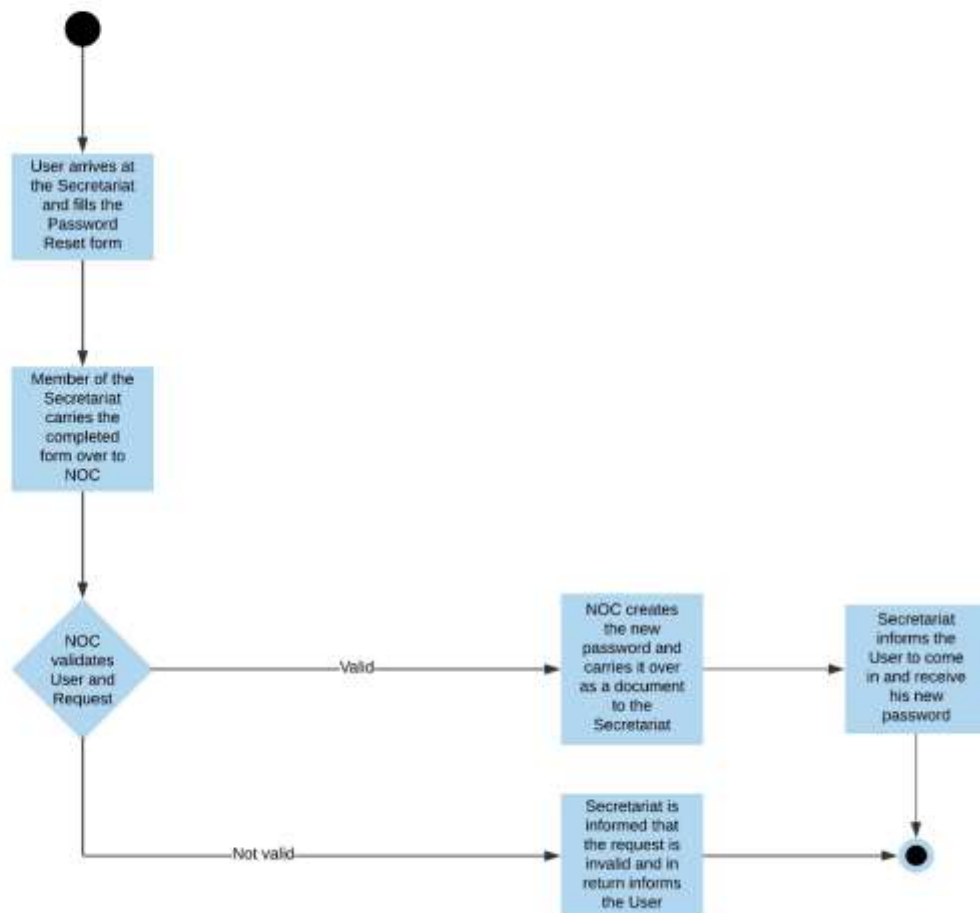
Το να χάσει ή ξεχάσει κάποιος χρήστης τον κωδικό του υπολογιστή του ή του e-mail του ή οποιασδήποτε άλλης υπηρεσίας, είναι ένα φαινόμενο που εμφανίζεται και στην εποχή μας, καθώς όλο και περισσότερες online υπηρεσίες απαιτούν κάποιο είδος εγγραφής πριν τις χρησιμοποιήσουμε. Παρόλα αυτά, δε χρησιμοποιούμε όλοι κάποιον password manager προκειμένου να αποθηκεύονται οι κωδικοί μας, ή ίσως δεν εμπιστευόμαστε έναν για να τον χρησιμοποιήσουμε, παρότι οι σύγχρονοι browsers αποθηκεύουν τους κωδικούς αυτόματα. Κατά αυτόν τον τρόπο, όταν «χαθεί» ο κωδικός μας, ξεκινάμε τη διαδικασία ανάκτησης.

1.1 Τρέχουσα διαδικασία

Στην περίπτωση του Χαροκοπείου Πανεπιστημίου, αν ένας χρήστης χάσει/ξεχάσει τον ιδρυματικό του κωδικό, για το e-mail ή το PC, θα πρέπει να μεταβεί στη Γραμματεία του τμήματός του, να κάνει έγγραφο αίτημα για έκδοση νέου κωδικού το οποίο θα προωθηθεί στο Κέντρο Πληροφορικής και Δικτύων. Εκεί θα γίνει η πιστοποίηση της εγκυρότητας του αιτήματος και έλεγχος για το αν ο χρήστης που πραγματοποίησε το αίτημα είναι ενεργός στο Δίκτυο. Εν συνεχεία θα εκδοθεί ο νέος κωδικός, θα μεταφερθεί εγγράφως στην αρμόδια Γραμματεία και θα κληθεί ο χρήστης που έκανε το αίτημα, ώστε να του κοινοποιηθεί ο νέος κωδικός. Αυτή η διαδικασία χρειάζεται όπως είναι φυσικό, κάποιες ημέρες για να ολοκληρωθεί.

Στο παρακάτω διάγραμμα εμφανίζεται σχηματικά η διαδικασία.

Current process activity diagram



Εικ 1 Τρέχουσα διαδικασία

1.2 Ανάγκη δημιουργίας αυτής της εφαρμογής

Η διπλωματική αυτή εργασία αποτελεί μια web εφαρμογή που αυτοματοποιεί κατά το δυνατόν την τρέχουσα διαδικασία αίτησης και έκδοσης ενός νέου ιδρυματικού κωδικού, για οποιονδήποτε υπάρχοντα χρήστη του Δικτύου του Χαροκοπέιου Πανεπιστημίου.

Όπως αναφέραμε και παραπάνω, η τρέχουσα διαδικασία που ακολουθείται είναι χρονοβόρα, καθώς για την όλη διαδικασία χρειάζεται να απασχοληθούν δύο (2) τμήματα και τουλάχιστον τρία (3) άτομα από το Προσωπικό του Πανεπιστημίου. Σε όλο αυτό θα πρέπει να συνυπολογίσουμε και την «ταλαιπωρία» του χρήστη, που θα πρέπει να μεταβεί στη Γραμματεία δύο (2) φορές, ενώ θα περιμένει την έκδοση του νέου κωδικού σε αόριστο χρόνο.

Αξίζει να τονιστεί πως στην εφαρμογή αυτή πραγματοποιούνται έλεγχοι για «sram» αιτήματα, ώστε να διασφαλιστεί η ασφάλεια του κωδικού του κάθε χρήστη καθώς και να μην αυξάνεται ο φόρτος εργασίας της εκάστοτε Γραμματείας με ψευδή αιτήματα.

1.3 Γιατί Angular 2+ framework

Ένα SPA (Single-page Application)[[1](#)] παίρνει μόνο μία HTML σελίδα από το server, όταν ο χρήστης επισκέπτεται αυτήν την εφαρμογή. Μαζί με αυτή τη σελίδα, ο server στέλνει και μια «μηχανή» για την εφαρμογή στον browser. Η «μηχανή» αυτή ελέγχει την επεξεργασία, τα εισερχόμενα κι εξερχόμενα δεδομένα μέσω microservices και τη φόρτωση των υπολοίπων σελίδων της εφαρμογής. Αυτό έχει σαν αποτέλεσμα η πλειοψηφία του κώδικα να τρέχει στον browser και λιγότερος κώδικας στο server. Επειδή μειώνεται η εξάρτηση από το server, όσοι χρήστες κι αν χρησιμοποιούν την εφαρμογή, η απόδοση δεν επηρεάζεται.

Το Angular framework κατασκευάζει SPAs.

1.4 Γιατί Django framework

Το Django είναι γραμμένο στη γλώσσα Python, η οποία είναι εύκολη στην εκμάθηση και γρήγορη στη συγγραφή. Το framework έχει σχεδιαστεί με τρόπο που βοηθάει τους προγραμματιστές να αναπτύξουν ταχύτατα εφαρμογές. Από την σύλληψη μια ιδέας μέχρι την ανάπτυξη και την έκδοση μιας εφαρμογής, το Django είναι ταυτόχρονα οικονομικό και αποδοτικό. Παλαιότερα διαφημιζόταν ως «το framework για προγραμματιστές με (αυστηρές) προθεσμίες». Περιέχει λειτουργίες και βοηθήματα που θα χρειαστεί ο προγραμματιστής κατά την ανάπτυξη μιας εφαρμογής, συν πληθώρα βιβλιοθηκών. Είναι ασφαλές, ευέλικτο και κάνει scale αρκετά γρήγορα κάτω από φόρτο, με τη βοήθεια ενός load balancer. Τέλος, υπάρχει πολύ μεγάλη κοινότητα για το Django, οπότε τυχόν προβλήματα κατά την ανάπτυξη μιας εφαρμογής από έναν προγραμματιστή, βρίσκουν γρήγορα λύση.

2. ΤΕΧΝΟΛΟΓΙΕΣ

2.1 JavaScript

Η JavaScript (αναφέρεται και ως «JS»)[2] είναι διερμηνευμένη γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές. Αρχικά αποτέλεσε μέρος της υλοποίησης των browsers, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο της σελίδας που εμφανίζεται.

Η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C. Η JavaScript αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων, για παράδειγμα έγγραφα PDF, site-specific browsers, desktop widgets. Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως το Node.js) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη web εφαρμογών στην πλευρά του διακομιστή (server-side). Το πρότυπο της γλώσσας κατά τον οργανισμό τυποποίησης ECMA ονομάζεται ECMAScript[3].

Η JavaScript δεν θα πρέπει να συγχέεται με τη Java, που είναι διαφορετική γλώσσα προγραμματισμού και με διαφορετικές εφαρμογές. Η χρήση της λέξης «Java» στο όνομα της γλώσσας έχει περισσότερη σχέση με το προφίλ του προϊόντος που έπρεπε να έχει και λιγότερο με κάποια πιθανή συμβατότητα ή άλλη στενή σχέση με τη Java. Ρόλο σε αυτήν τη σύγχυση έπαιξε και ότι η Java και η JavaScript έχουν δεχτεί σημαντικές επιρροές από τη γλώσσα C, ειδικά στο συντακτικό, ενώ είναι και οι δύο αντικειμενοστρεφείς γλώσσες. Τονίζεται ότι ο σωστός τρόπος γραφής της είναι «JavaScript» και όχι «Java script» σαν δύο λέξεις, όπως λανθασμένα γράφεται ορισμένες φορές.

2.2 Angular

Το Angular (κοινώς αναφερόμενο ως «Angular 2+» ή «Angular v2 και άνω»)[4] είναι ένα web application framework ανοικτού κώδικα βασισμένο σε TypeScript, που συντηρείται από την ομάδα Angular της Google και από μια κοινότητα ατόμων και εταιρειών. Το Angular είναι από την ίδια ομάδα που δημιούργησε την AngularJS.

Το Angular δεν είναι μία αναβαθμισμένη έκδοση της AngularJS. Είναι ένα framework γραμμένο πλήρως από την αρχή με αρκετές βελτιώσεις από την Google και είναι διαθέσιμο από το Σεπτέμβριο του 2016. Μία από τις κυριότερες διαφορές έχει να κάνει με τον τρόπο διαχείρισης των δρομολογήσεων, δηλαδή της έννοιας του routing. Ο τρόπος προσέγγισης όμως για τον εκάστοτε προγραμματιστή παραμένει ο ίδιος, καθώς η κυριότερη έννοια στο πλαίσιο λογισμικού και αυτή που έκανε τη διαφορά όταν κυκλοφόρησε η AngularJS, είναι η έννοια του component. Η AngularJS δημιουργήθηκε χωρίς να έχει υπόψη την υποστήριξη για κινητά και το πώς θα εμφανίζονται σε αυτά οι αντίστοιχες ιστοσελίδες. Η υποστήριξη δόθηκε μέσω πρόσθετων βιβλιοθηκών, ενώ το Angular γράφτηκε με προτεραιότητα την αρχιτεκτονική των εφαρμογών Διαδικτύου για κινητά.

Το Angular μπορεί να αναπτυχθεί κυρίως χρησιμοποιώντας (αλλά όχι αποκλειστικά) TypeScript, η οποία είναι ένα υπερσύνολο της JavaScript. Η TypeScript είναι η γλώσσα στην οποία δημιουργήθηκε το πλαίσιο λογισμικού Angular. Είναι μια γλώσσα που παράγει πιο οργανωμένο κώδικα JavaScript και είναι εύκολο να ξεκινήσει κανείς να δουλεύει σε Angular μέσω αυτής. Η Google έχει χρησιμοποιήσει σε έργα της DART, η οποία είναι μία ακόμα γλώσσα προγραμματισμού που μπορεί να χρησιμοποιηθεί. Συνολικά, στο Angular μπορούν να χρησιμοποιηθούν οι εξής γλώσσες: JavaScript (ES5,ESNext), TypeScript και DART.

Εφόσον το Angular είναι client-side τεχνολογία, χρησιμοποιήθηκε για να χτιστεί το user interface της εφαρμογής.

2.3 TypeScript

Η TypeScript[5] είναι μία γλώσσα ανοικτού κώδικα που δημιουργήθηκε και συντηρείται από τη Microsoft. Η πρώτη σταθερή έκδοση δόθηκε τον Οκτώβριο του 2012 και όπως γράφεται και στο specification της γλώσσας, δημιουργήθηκε για να καλύψει τις ανάγκες ομάδων JavaScript προγραμματιστών οι οποίες αναπτύσσουν και διατηρούν πολύ μεγάλες εφαρμογές. Πρόκειται

ουσιαστικά για ένα υπερσύνολο της JavaScript με αυστηρή και στατική σύνταξη το οποίο μεταγλωττίζεται σε JavaScript κώδικα. Ένας προγραμματιστής που έχει γνώση από αντικειμενοστρεφή προγραμματισμό και JavaScript μπορεί εύκολα και γρήγορα να αρχίσει την εκμάθηση της γλώσσας, ενώ το στατικό συντακτικό (που συναντάμε σε πολλές γλώσσες όπως η Java) βοηθάει στην αποφυγή λαθών πριν τρέξουμε την εφαρμογή. Ένα ακόμα πλεονέκτημα της γλώσσας είναι πως αν πάρουμε κώδικα JavaScript και αλλάξουμε την κατάληξη του αρχείου σε TypeScript (δηλαδή από .js σε .ts) αυτός πάλι θα μπορέσει να μεταγλωττιστεί σε JavaScript. Η μεταγλώττιση σε JavaScript είναι ουσιαστικά μια μετατροπή η οποία χρειάζεται ένα εργαλείο που ονομάζεται transpiler, το οποίο μας παρέχεται από το Angular CLI. Στην TypeScript έχουν ενσωματωθεί χαρακτηριστικά που παρουσιάστηκαν και στις πρόσφατες εκδόσεις της JavaScript όπως οι κλάσεις (classes) και οι λειτουργικές μονάδες (modules). Ταυτόχρονα η γλώσσα έχει και δικές της προσθήκες όπως τους Τύπους (types) καθώς και την εισαγωγή/εξαγωγή ενοτήτων (module import/export). Για να τρέξει ο TypeScript κώδικας, θα χρειαστεί να εγκαταστήσουμε στον υπολογιστή μας το Node.js.

2.4 Bootstrap

Το Bootstrap[6] είναι ένα από τα πιο δημοφιλή και ισχυρά frameworks για την κατασκευή ιστοσελίδων, που υπάρχουν σήμερα. Δημιουργήθηκε από τους Mark Otto, Jacob Thornton και μία μικρή ομάδα προγραμματιστών. Κυκλοφόρησε πρώτη φορά στις 19 Αυγούστου του 2011. Η πρωταρχική του ονομασία ήταν Twitter Blueprint, καθώς βρισκόταν υπό ανάπτυξη για την υποστήριξη κάποιων εργαλείων της πλατφόρμας του Twitter. Στη συνέχεια, μετονομάστηκε σε Bootstrap και φέρει το όνομα αυτό μέχρι και σήμερα. Πρόκειται για ένα framework ανοιχτού κώδικα, με το οποίο μπορεί κανείς να αναπτύξει web εφαρμογές, καθώς και ιστοσελίδες, χρησιμοποιώντας τεχνολογίες όπως HTML, CSS και JavaScript. Διαθέτει έτοιμα templates με πολύ όμορφα αισθητικά κουμπιά, φόρμες, τυπογραφία, όπως επίσης και μοντέρνα μενού πλοήγησης. Υποστηρίζεται από όλους τους browsers, σύγχρονους και παλαιότερους.

2.5 Angular Material

Το Material design[7] είναι ένα σύνολο σχεδιαστικών κανόνων που δημιουργήθηκε από τη Google και ανακοινώθηκε για πρώτη φορά στο συνέδριο Google I/O στις 25 Ιουνίου, 2014. Επεκτείνει το μοντέλο των «καρτών» (πρωτοεμφανίστηκε στο Google Now) που είναι σχεδιασμός, βασισμένος σε διάταξη πίνακα (grid layout), με κινούμενα σχέδια και μεταβάσεις που ανταποκρίνονται στις κινήσεις των χρηστών, χρήση padding και εφέ βάθους με κατάλληλο φωτισμό και σκίαση των αντικειμένων. Ο σχεδιαστής Matías Duarte εξήγησε ότι «σε αντίθεση με το πραγματικό χαρτί, το ψηφιακό υλικό μπορεί να επεκταθεί και να μορφοποιηθεί έξυπνα. Το material έχει φυσικές επιφάνειες και άκρα. Οι σκιές δίνουν νόημα σε αυτό που αγγίζεις». Η Google υποστηρίζει ότι η καινούρια σχεδιαστική τους γλώσσα βασίζεται στο χαρτί και το μολύβι.

2.6 JWT (JSON Web Token)

Το JWT (JSON Web Token)[8] είναι ένα Internet standard για τη δημιουργία JSON-based access tokens, που σιγουρεύουν κάποιον αριθμό αξιώσεων. Για παράδειγμα, ένας server μπορεί να δημιουργήσει ένα token που έχει την αξίωση «συνδεδεμένος ως διαχειριστής» και να τα προωθήσει σε ένα χρήστη/πελάτη. Ο τελευταίος μπορεί στη συνέχεια να χρησιμοποιεί αυτό το token όταν επικοινωνεί με το server, για να αποδεικνύει ότι είναι όντως συνδεδεμένος ως διαχειριστής. Τα tokens υπογράφονται ψηφιακά από το ένα μέρος με το ιδιωτικό κλειδί κρυπτογράφησης, ούτως ώστε και τα δύο μέρη να πιστοποιούν την αυθεντικότητα του token, χρησιμοποιώντας το δημόσιο κλειδί κρυπτογράφησης.

Τα tokens είναι σχεδιασμένα να είναι συμπαγή, URL-safe και χρησιμοποιήσιμα, ειδικά σε μια περίπτωση SSO (Single Sign On). Οι αξιώσεις JWT συνήθως χρησιμοποιούνται για να μεταβιβάσουν την ταυτότητα ενός πιστοποιημένου χρήστη, όπως στο παράδειγμα που περιγράφηκε παραπάνω.

Το JWT βασίζεται σε άλλα JSON-based standards, το JSON Web Signature και το JSON Web Encryption.

2.7 Python

Τη λίστα των γλωσσών προγραμματισμού υψηλού επιπέδου, όπως είναι η Java, C++, C# και άλλες, έρχεται να συμπληρώσει η Python[9]. Η δημιουργία της ξεκίνησε το 1980 από τον Ολλανδό Guido van Rossum και η κυκλοφορία της πραγματοποιήθηκε το 1991. Πρόκειται για μία γλώσσα προγραμματισμού γενικού σκοπού, η οποία είναι εύκολη σε εκμάθηση. Για αυτό το λόγο καθίσταται ιδανική, για να κάνει κάποιος τα πρώτα του βήματα στο χώρο του προγραμματισμού. Ο σκοπός του δημιουργού της είναι να την καταστήσει ευχάριστη στους χρήστες, που θα την επιλέξουν και γι' αυτό πήρε το όνομά της από τους γνωστούς Βρετανούς κωμικούς Monty Python. Οι τρόποι που απαιτούνται, ώστε ένα πρόγραμμα να μετατραπεί σε γλώσσα μηχανής υψηλού επιπέδου με σκοπό να γίνει κατανοητή και εκτελέσιμη από έναν ηλεκτρονικό υπολογιστή, είναι δύο. Πρώτο τρόπο αποτελεί η διαδικασία της μεταγλώττισης (compiling), κατά την οποία ο compiler αναλαμβάνει τον πηγαίο κώδικα (source code) και παράγει ένα πρόγραμμα (executable), το οποίο μπορούμε να εκτελέσουμε. Ως δεύτερος τρόπος ορίζεται η διαδικασία της διερμηνείας (interpreted), κατά την οποία ο interpreter μεταφράζει και εκτελεί το πρόγραμμα απευθείας, χωρίς να χρειαστεί πρώτα να δημιουργήσει κάποιο εκτελέσιμο αρχείο. Η Python επομένως, χρησιμοποιεί τη δεύτερη διαδικασία, για να μετατρέψει ένα πρόγραμμα σε γλώσσα μηχανής. Είναι, δηλαδή, μία interpreted γλώσσα προγραμματισμού. Αυτό αποτελεί ένα από τα βασικά χαρακτηριστικά της, το οποίο την καθιστά γρήγορη, αλλά και ευέλικτη σε σχέση με άλλες γλώσσες προγραμματισμού. Ταυτόχρονα, είναι μία αντικειμενοστρεφής γλώσσα προγραμματισμού, είναι ανοικτού κώδικα, πράγμα το οποίο σημαίνει πως μπορεί κανείς να την κατεβάσει ελεύθερα στον υπολογιστή του και να την εγκαταστήσει σε οποιοδήποτε λειτουργικό σύστημα επιθυμεί (Linux, MacOS, Windows). Η Python επίσης είναι ανεξάρτητη πλατφόρμας. Αυτό σημαίνει πως ο κώδικας κάποιου προγράμματός της μπορεί να εκτελεστεί σε ποικίλα συστήματα, όπως για παράδειγμα στη γνωστή παιχνιδομηχανή Playstation. Χαρακτηρίζεται μάλιστα από τον δυναμικό τύπο του συστήματός της, καθώς και από την αυτόματη διαχείριση μνήμης την οποία διαθέτει.

Τον τελευταίο χρόνο, παρατηρείται παγκοσμίως πως η γλώσσα Python χρησιμοποιείται αρκετά σε τομείς όπως Data Mining, AI, Analytics και Telecommunications.

2.8 Django

Το Django[10] είναι ένα ελεύθερο και ανοιχτού κώδικα web framework βασισμένο σε Python. Η ονομασία του, την οποία έδωσαν οι δημιουργοί του Adrian Holovaty και Simon Willison το έτος 2005, προέρχεται από τον Βέλγο - Γάλλο κιθαρίστα Django Reinhardt. Αξίζει να αναφερθεί πως το Django ξεκίνησε αρχικά σαν project, που δημοσιεύθηκε στην εφημερίδα Lawrence Journal-World στο Kansas των Ηνωμένων Πολιτειών. Συντηρείται από το Ίδρυμα Λογισμικού Django(DSF), ανεξάρτητο οργανισμό που ιδρύθηκε ως μη κερδοσκοπικός οργανισμός.

Ο πρωταρχικός στόχος του Django είναι να διευκολύνει τη δημιουργία πολύπλοκων δικτυακών τόπων που βασίζονται σε Βάσεις Δεδομένων. Το Django υπογραμμίζει την επαναχρησιμοποίηση και την «δυνατότητα σύνδεσης» των components, τον λιγότερο κώδικα, το «low coupling», την ταχεία ανάπτυξη και την αρχή του «DRY»(Do not Repeat Yourself)[11]. Η Python χρησιμοποιείται σε ολόκληρο το περιβάλλον, ακόμα και για αρχεία ρυθμίσεων και μοντέλα δεδομένων.

Τα σύγχρονα frameworks ακολουθούν το αρχιτεκτονικό πρότυπο Model-View-Controller (MVC)[12]. Το Django βασίζεται στο συγκεκριμένο πρότυπο, ή Model-Template-View (MTV), όπως το περιγράφει το ίδιο. Παρακάτω δίνεται η περιγραφή των τριών μερών από τα οποία αποτελείται το συγκεκριμένο framework:

- *Model*: Πρόκειται για την αναπαράσταση, αλλά και τη συσχέτιση των δεδομένων, που έχουν δηλωθεί ως κλάσεις στη γλώσσα Python και βρίσκονται στη Βάση Δεδομένων. Το Django κάνει χρήση της τεχνικής ORM (Object-relational Mapping)[13], που έχει την ιδιότητα να μετατρέπει τα δεδομένα μεταξύ δύο μη συμβατών συστημάτων, χρησιμοποιώντας αντικειμενοστρεφείς γλώσσες προγραμματισμού
- *View*: Η αναπαράσταση των δεδομένων γίνεται σε μία HTML σελίδα, η οποία κάνει χρήση ενός ορισμένου, στη γλώσσα Python, κώδικα. Στην ουσία πρόκειται για το μέρος που αφορά στο χρήστη, αλλά και στη δυνατότητα που εκείνος αποκτά, ώστε να αλληλοεπιδρά με τον browser
- *Controller*: Αποτελεί το πιο βασικό κομμάτι του συστήματος, καθώς είναι υπεύθυνος για τη διαχείριση της πληροφορίας, η οποία ανακτάται από τη Βάση Δεδομένων μέσω του *Model*, αλλά και για την πληροφορία που εμφανίζεται στην οθόνη μέσω του *View*. Έχει

την ικανότητα να διαχειρίζεται απαντήσεις κι ερωτήματα, καθώς και να δημιουργεί τη σύνδεση με τη Βάση Δεδομένων

Ένα από τα κυριότερα χαρακτηριστικά που διαθέτει το συγκεκριμένο framework είναι η ιστοσελίδα του Διαχειριστή (Admin Interface). Μέσω αυτής της ιστοσελίδας, παρέχεται η δυνατότητα στον χρήστη να τροποποιήσει έναν ήδη υπάρχοντα λογαριασμό, όπως και το να έρθει σε επαφή με τα μοντέλα, τα οποία έχουν δημιουργηθεί στη Βάση Δεδομένων. Αυτό επιτυγχάνεται όταν το Django χρησιμοποιεί τις κλάσεις, τις οποίες έχουμε δηλώσει μέσα στο αρχείο `admin.py`. Παράλληλα, το Django περιλαμβάνει έναν πολύ ελαφρύ αλλά ταχύ web server για την ανάπτυξη μιας εφαρμογής, καθώς και για το testing αυτής.

Ακόμα ένα χαρακτηριστικό το οποίο θα μπορούσαμε να αναφέρουμε, είναι πως το Django διαθέτει κρυφή μνήμη (cache). Κάθε φορά που κάποιος χρήστης ζητάει μία ιστοσελίδα, ο web server από τον οποίο ο χρήστης τη ζητάει, πραγματοποιεί κάποιους υπολογισμούς, για παράδειγμα ερωτήματα στη Βάση Δεδομένων. Με τον τρόπο αυτό αποφεύγεται η επανεκτέλεση ενός τέτοιου είδους υπολογισμού και το αποτέλεσμα αποθηκεύεται στη μνήμη cache, επιστρέφοντας την απάντηση στο χρήστη σε μικρότερο χρονικό διάστημα.

Βάση αυτών των χαρακτηριστικών, θα μπορούσαμε να πούμε πως το Django είναι:

- *Γρήγορο*: Ο τρόπος με τον οποίο έχει δημιουργηθεί, εξυπηρετεί τις ανάγκες ακόμα και του πιο απαιτητικού προγραμματιστή, εύκολα και σε σύντομο χρονικό διάστημα.
- *Ασφαλές*: Δίνει τη δυνατότητα στον προγραμματιστή να κάνει deploy την εφαρμογή του με τη «σιγουριά» που προσφέρει το HTTPS. Τα templates του Django έρχονται ενσωματωμένα με τους λεγόμενους χαρακτήρες διαφυγής (escape specific characters), προστατεύοντας έτσι το χρήστη από XSS attacks (Cross Site Scripting). Ακόμα, το framework έχει την ιδιότητα να προστατεύει τους χρήστες του από την επίθεση SQL Injection.
- *Επεκτάσιμο*: Οποιαδήποτε προσθήκη ή αλλαγή αν κάνει κάποιος, όσον αφορά το κομμάτι του hardware στον web ή application server, αυτή δεν επηρεάζει το συγκεκριμένο framework.

Στα πλαίσια αυτής της διπλωματικής, το Django χρησιμοποιήθηκε ως server-side τεχνολογία, μέσω της οποίας οργανώθηκε το Business Logic και η διαχείριση της Βάσης Δεδομένων.

2.9 PostgreSQL

Η PostgreSQL[14] είναι μια σχεσιακή Βάση Δεδομένων ανοικτού κώδικα με πολλές δυνατότητες. Η ανάπτυξη της διαρκεί ήδη πάνω από δύο δεκαετίες και βασίζεται σε μια αποδεδειγμένα καλή αρχιτεκτονική η οποία έχει δημιουργήσει μια ισχυρή αντίληψη των χρηστών της γύρω από την αξιοπιστία, την ακεραιότητα δεδομένων και την ορθή λειτουργία.

Η PostgreSQL τρέχει σε όλα τα βασικά λειτουργικά συστήματα, στα οποία περιλαμβάνονται το Linux, το macOS και τα Windows. Είναι συμβατή με ACID[15] και συμπεριλαμβάνει τους περισσότερους SQL92 και SQL99 τύπους δεδομένων συμπεριλαμβανομένων INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL και TIMESTAMP. Επίσης υποστηρίζει αποθήκευση μεγάλων δυαδικών αντικειμένων (binary), όπως εικόνες, ήχοι ή βίντεο. Διαθέτει επίσης περιβάλλοντα προγραμματισμού για τις γλώσσες C, C++, Java, Perl, Python, Ruby, Tcl, καθώς και υποστήριξη για την πλατφόρμα .NET και το πρότυπο ODBC, ενώ περιλαμβάνει και εξαιρετικό εγχειρίδιο χρήσης.

2.10 IDEs

Για την ανάπτυξη της εφαρμογής, χρησιμοποιήθηκαν δύο (2) IDEs, δηλαδή λογισμικά που προσφέρουν ένα ολοκληρωμένο περιβάλλον ανάπτυξης για μια γλώσσα προγραμματισμού. Τα δύο (2) αυτά λογισμικά είναι της τσέχικης εταιρείας JetBrains.

Η JetBrains s.r.o. (πρώην IntelliJ Software s.r.o.)([16]) είναι μια εταιρεία ανάπτυξης λογισμικού, τα εργαλεία της οποίας απευθύνονται σε προγραμματιστές λογισμικού και διαχειριστές έργων. Από το 2019, η εταιρεία διαθέτει πάνω από 990 υπαλλήλους στα γραφεία της στην Πράγα, Αγία Πετρούπολη, Μόσχα, Μόναχο, Βοστώνη, Νοβοσιμπίρσκ, Άμστερνταμ, Foster City και Marlton, New Jersey. Η εταιρεία προσφέρει μια εκτεταμένη οικογένεια ολοκληρωμένων αναπτυξιακών περιβαλλόντων (IDE) για τις γλώσσες προγραμματισμού Java, Kotlin, Ruby, Python, PHP, C, Objective-C, C ++, C #, Go, JavaScript και SQL. Το περιοδικό InfoWorld της απένειμε το βραβείο "Technology of the Year Award" το 2011 και το 2015.

Αυτά τα IDEs παρέχουν ανάλυση κώδικα, γραφικό εργαλείο εντοπισμού σφαλμάτων, ενσωματωμένο unit tester, καθώς και σύνδεση με συστήματα ελέγχου εκδόσεων (VCSes). Είναι πολλαπλής πλατφόρμας, με εκδόσεις Windows, macOS και Linux. Το ένα ονομάζεται PyCharm[17] και χρησιμοποιείται κυρίως για τη γλώσσα Python, ενώ το άλλο ονομάζεται WebStorm[18] και χρησιμοποιείται κυρίως για τη γλώσσα JavaScript. Ένα ακόμα χαρακτηριστικό, το οποίο πρέπει να τονίσουμε, είναι πως το τερματικό (terminal), το οποίο διαθέτει το WebStorm καθώς και το PyCharm, δίνει την δυνατότητα στον χρήστη να πληκτρολογεί τις απαραίτητες εντολές μέσα σε αυτό κρατώντας επαφή με τον κώδικα, χωρίς να χρειάζεται να εναλλάσσει παράθυρα στο λειτουργικό σύστημα όπου εργάζεται.

Η Κοινοτική έκδοση κυκλοφορεί βάσει της Άδειας Apache. Υπάρχει επίσης Επαγγελματική έκδοση με επιπλέον χαρακτηριστικά, η οποία απαιτεί άδεια επί πληρωμή.

3. ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ

Βασικό κομμάτι της ανάπτυξης μιας εφαρμογής, είναι η ανάλυση των προδιαγραφών και η σχεδίαση των λειτουργιών. Αποφασίσαμε πως η εφαρμογή θα επικοινωνεί με το web server μέσω REST[19] API κλήσεων.

Ξεκινήσαμε με τις εξής παραδοχές:

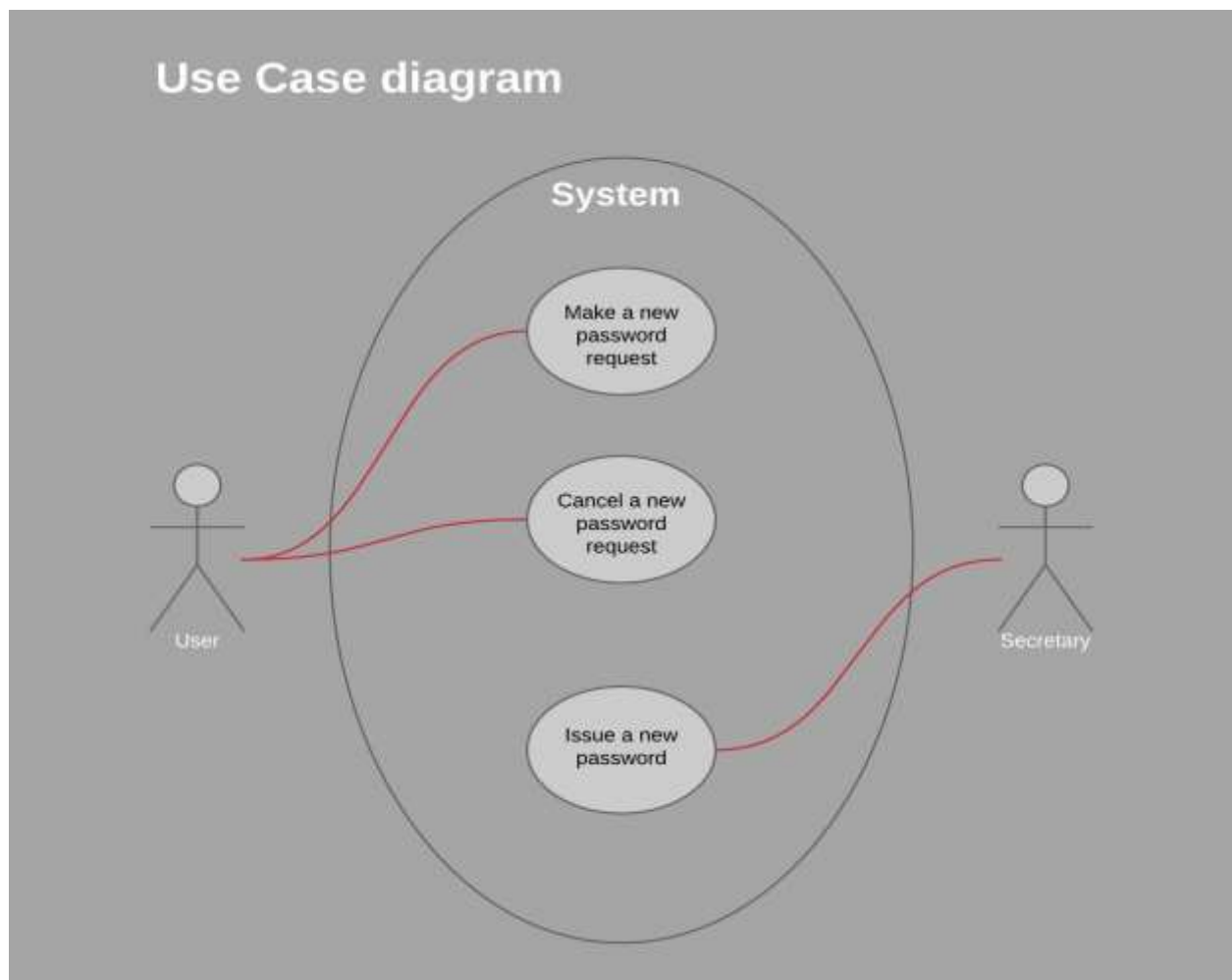
- Οι χρήστες της εφαρμογής δε θα χρειάζεται να κάνουν εγγραφή στην εφαρμογή για να τη χρησιμοποιήσουν. Αποφασίστηκε αυτό διότι θα υπάρχει μόνο μία (1) οθόνη, αυτή που θα περιέχει τη φόρμα του αιτήματος νέου κωδικού. Θα γίνεται όμως αυθεντικοποίηση του χρήστη που κάνει το αίτημα, μέσω του LDAP server του Πανεπιστημίου. Αν δεν γίνει επιτυχώς η αυθεντικοποίηση, τότε ο χρήστης δε θα μπορεί να κάνει το αίτημα
- Η ακύρωση ενός αιτήματος έκδοσης νέου κωδικού, θα γίνεται υπό κάποιες προϋποθέσεις:
 - Εάν δεν έχει εκδοθεί ήδη νέος κωδικός για το συγκεκριμένο αίτημα
 - Εάν δεν έχει ακυρωθεί ήδη το συγκεκριμένο αίτημα
 - Εάν δεν έχει λήξει η περίοδος ισχύος του συγκεκριμένου αιτήματος

- Η έκδοση νέου κωδικού από μια Γραμματεία θα γίνεται εάν:
 - Η περίοδος ισχύος του συγκεκριμένου αιτήματος δεν έχει λήξει
 - Δεν έχει εκδοθεί ήδη νέος κωδικός για το συγκεκριμένο αίτημα
 - Ο χρήστης δεν έχει ακυρώσει το αίτημα

3.1 Use Case diagram

Το Use Case diagram[[20](#)] περιλαμβάνει όλες τις βασικές λειτουργίες που διαθέτει ένα σύστημα.

Στο παρακάτω διάγραμμα βλέπουμε τις επιλογές που έχουν διαθέσιμες οι δύο (2) κατηγορίες χρηστών της εφαρμογής. Ο οποιοσδήποτε χρήστης μπορεί να κάνει αίτημα για έκδοση νέου κωδικού και ακύρωση αυτού του αιτήματος. Ο χειριστής της Γραμματείας μπορεί να εκδώσει ένα νέο κωδικό για κάποιον χρήστη που έχει κάνει αίτημα.



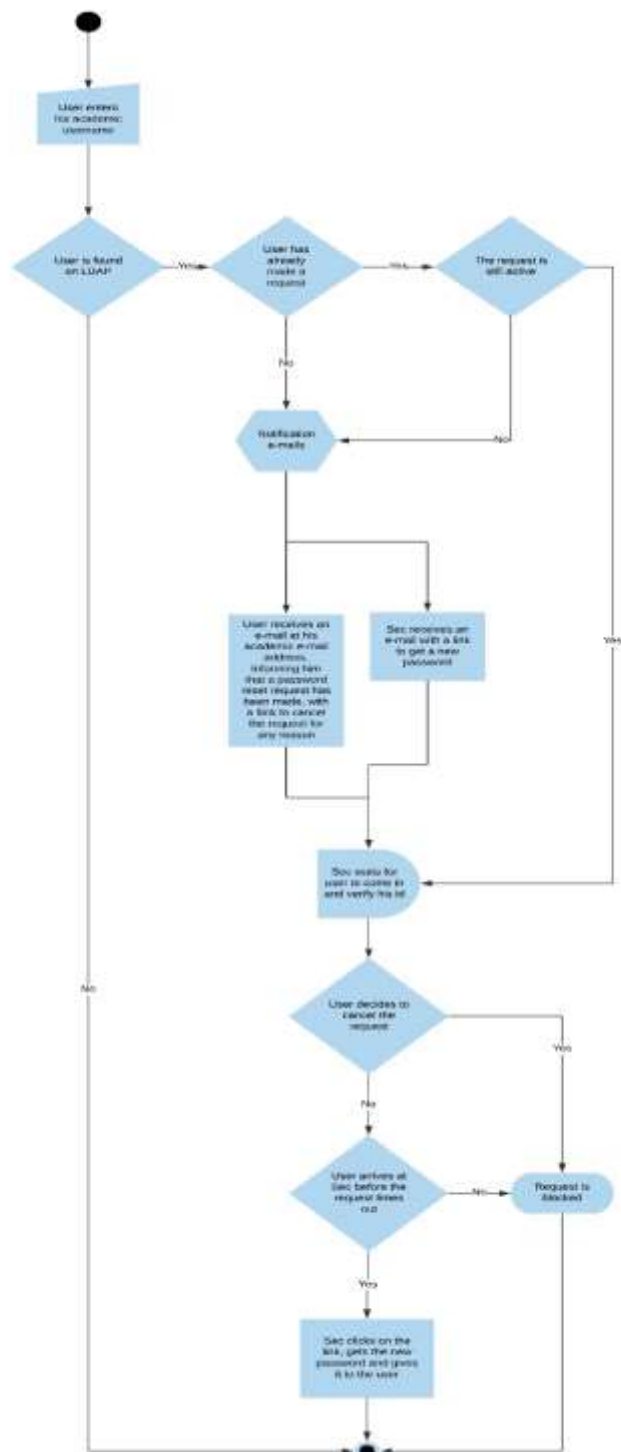
Εικ 2 Use Case diagram

3.2 Activity diagram

Το Activity diagram[21] δείχνει τη ροή των ενεργειών μιας λειτουργίας, προκειμένου να ολοκληρωθεί επιτυχώς ή να εμφανίσει κάποιο διαγνωστικό μήνυμα σφάλματος στο χρήστη.

Στο παρακάτω διάγραμμα βλέπουμε τη ροή από το αίτημα νέου κωδικού έως και την έκδοση αυτού.

Activity diagram

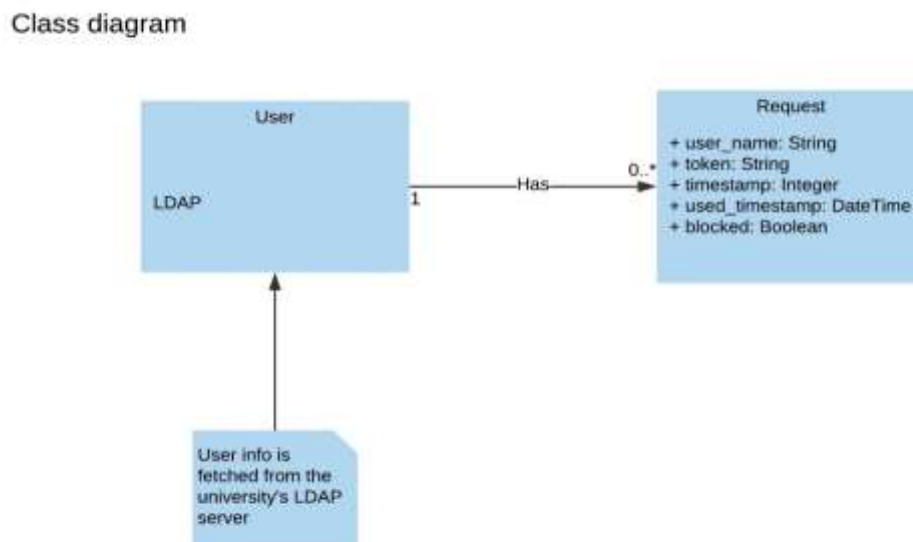


Euk 3 Activity diagram

3.3 Class diagram

Το Class diagram[22] περιλαμβάνει τις οντότητες της εφαρμογής και τις σχέσεις μεταξύ των.

Στο παρακάτω διάγραμμα βλέπουμε τη σχέση ενός χρήστη της εφαρμογής με το αίτημα νέου κωδικού, καθώς και τα πεδία του τελευταίου.



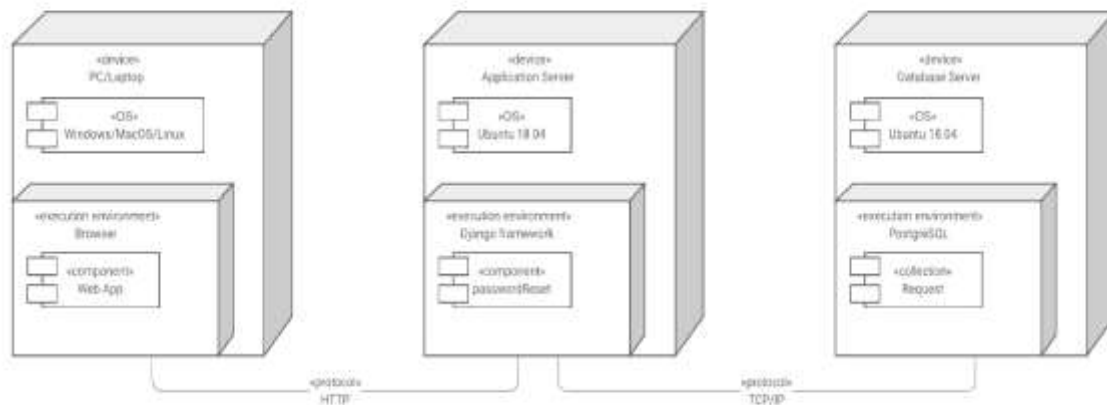
Εικ 4 Class diagram

3.4 Deployment diagram

Το Deployment diagram[23] δείχνει τη διαμόρφωση των κόμβων και τα στοιχεία που ζουν σε αυτούς. Είναι ένα είδος δομικού διαγράμματος που χρησιμοποιείται για τη μοντελοποίηση των φυσικών πτυχών ενός αντικειμενοστρεφούς συστήματος. Επίσης χρησιμοποιείται συχνά για να μοντελοποιήσει τη στατική θέαση ανάπτυξης ενός συστήματος (τοπολογία του υλικού).

Στο παρακάτω διάγραμμα παρατηρούμε τη διαμόρφωση του συστήματος, στο οποίο θα τρέχει η εφαρμογή που αναπτύξαμε.

Deployment diagram



Εικ 5 Deployment diagram

4. ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ

Η υλοποίηση της εφαρμογής έγινε σε λειτουργικό σύστημα Ubuntu 18.04[24]. Κατά συνέπεια, οι εντολές που παρατίθενται παρακάτω, τρέχουν σε ένα τέτοιο περιβάλλον.

Το λειτουργικό φιλοξενήθηκε σε ηλεκτρονικό υπολογιστή με τα εξής χαρακτηριστικά:

- CPU 8-CORE 4 GHz
- RAM 16 GB DDR3 1600 MHz
- SSD 250 GB

Η εφαρμογή δεν έχει ιδιαίτερες απαιτήσεις από τον web/application server στον οποίο θα φιλοξενηθεί. Απαραίτητο είναι να «τρέχει» λειτουργικό σύστημα Ubuntu 18.04, στο οποίο έγινε και η ανάπτυξη. Άλλη έκδοση λειτουργικού ίσως προκαλέσει ασυμβατότητες με τις βιβλιοθήκες/πακέτα που απαιτεί η εφαρμογή για να εκτελεστεί σε περιβάλλον παραγωγής. Θα πρέπει όμως να δώσουμε δύναμη hardware, ώστε να η λειτουργία της εφαρμογής να γίνεται απρόσκοπτα ακόμα και σε περίπτωση που γίνουν μαζικά αιτήματα από τους χρήστες. Ένας

server με τα παρακάτω χαρακτηριστικά hardware αρκεί για να λειτουργήσει σωστά το λειτουργικό σύστημα, η εφαρμογή και η Βάση Δεδομένων:

- CPU 2-CORE 2,5 GHz
- RAM 8 GB DDR3 1333 MHz
- SSD 40 GB

4.1 Προαπαιτούμενα

Για την εγκατάσταση του λογισμικού που χρησιμοποιεί η εφαρμογή, απαραίτητο είναι να υπάρχουν ήδη εγκατεστημένα τα παρακάτω προγράμματα/βιβλιοθήκες.

4.1.1 Git

Το Git[25] είναι ένα σύστημα ελέγχου εκδόσεων (λέγεται και σύστημα ελέγχου αναθεωρήσεων ή σύστημα ελέγχου πηγαίου κώδικα) με έμφαση στην ταχύτητα, στην ακεραιότητα των δεδομένων και στην υποστήριξη για κατανεμημένες μη γραμμικές ροές εργασίας. Το Git σχεδιάστηκε και αναπτύχθηκε αρχικά από τον Λίνους Τόρβαλντς για την ανάπτυξη του πυρήνα Linux το 2005 και έχει γίνει από τότε το πιο διαδεδομένο σύστημα ελέγχου εκδόσεων για ανάπτυξη λογισμικού.

Το χρησιμοποιήσαμε για να υπάρχει αποθηκευμένος ο πηγαίος κώδικας της εφαρμογής σε ένα ασφαλές αποθετήριο στο GitHub, ώστε να μπορούμε να εργαζόμαστε σε αυτόν από οποιοδήποτε μηχάνημα έχει πρόσβαση στο Διαδίκτυο. Επίσης όποιος έχει το σύνδεσμο, μπορεί να δει τον κώδικα καθώς το αποθετήριο είναι δημόσιο.

Αυτός είναι ο σύνδεσμος που μας μεταφέρει στο αποθετήριο του GitHub, όπου και βρίσκεται ο πηγαίος κώδικας της εν λόγω εφαρμογής: <https://github.com/veliside/thesis>.

4.1.2 npm και Node.js

Για να εγκατασταθεί και να τρέξει το Angular framework σε περιβάλλον ανάπτυξης, χρειάζεται να έχουμε στον υπολογιστή μας εγκατεστημένα το npm και το Node.js.

Το `npm` (Node Package Manager)[26] είναι ένας διαχειριστής πακέτων για τη γλώσσα προγραμματισμού JavaScript. Είναι ο προεπιλεγμένος διαχειριστής πακέτων για το περιβάλλον εκτέλεσης JavaScript, `Node.js`. Αποτελείται από ένα τερματικό, που ονομάζεται επίσης `npm`, και μια ηλεκτρονική βάση δεδομένων δημόσιων και «πληρωμένων για ιδιωτικά» πακέτων, που ονομάζεται μητρώο `npm`. Τα διαθέσιμα πακέτα μπορούν να αναζητηθούν μέσω της ιστοσελίδας του `npm`. Το διαχειριστή πακέτων και το μητρώο, τα διαχειρίζεται το `npm, Inc`.

Το `Node.js`[27] είναι μια πλατφόρμα ανάπτυξης λογισμικού (κυρίως διακομιστών), χτισμένη σε περιβάλλον JavaScript. Στόχος του `Node.js` είναι να παρέχει έναν εύκολο τρόπο δημιουργίας κλιμακωτών διαδικτυακών εφαρμογών. Σε αντίθεση με τα περισσότερα σύγχρονα περιβάλλοντα ανάπτυξης εφαρμογών δικτύων, μία διεργασία `node` δεν στηρίζεται στην πολυνηματικότητα αλλά σε ένα μοντέλο ασύγχρονης επικοινωνίας εισόδου/εξόδου.

Η εγκατάσταση αυτών των λογισμικών γίνεται με ένα βήμα, αρκεί να επισκεφθούμε την ιστοσελίδα <https://nodejs.org/en/download/> και να κατεβάσουμε-εκτελέσουμε την LTS έκδοση για το λειτουργικό μας σύστημα.

4.1.3 Python

Το Django framework απαιτεί να υπάρχει προεγκατεστημένη η γλώσσα Python στον υπολογιστή μας. Αρκεί να επισκεφθούμε τη διεύθυνση <https://www.python.org/downloads/> και να κατεβάσουμε-εκτελέσουμε την έκδοση που μας ενδιαφέρει, για το λειτουργικό μας σύστημα, συγκεκριμένα στην εφαρμογή μας η έκδοση 3.6.1..

4.1.4 PIP

Για την ευκολότερη διαχείριση των πακέτων μας, θα χρησιμοποιήσουμε το PIP (Pip Installs Packages)[28], το οποίο έχει την ιδιότητα να εγκαθιστά και να απεγκαθιστά πακέτα της Python. Το PIP εγκαθίσταται πλέον αυτόματα μαζί με τις νέες εκδόσεις της γλώσσας Python. Αν δεν το έχουμε στο σύστημά μας μετά την εγκατάσταση της Python, μπορούμε να τρέξουμε την εντολή **`sudo apt-get install python3-pip`**.

4.2 Εγκατάσταση PostgreSQL

Για τις ανάγκες αυτής της διπλωματικής εργασίας, δημιουργήθηκε μια νέα Βάση Δεδομένων με όνομα *thesis*, η οποία περιέχει έναν (1) πίνακα (table) με όνομα *REQUEST*.

4.2.1 PostgreSQL και dependencies

Για να λειτουργήσει η PostgreSQL με την Python, χρειάζεται να εγκατασταθούν επιπλέον βιβλιοθήκες. Αυτές οι βιβλιοθήκες εγκαθίστανται με τη χρήση της εντολής **sudo apt-get install python3-dev libpq-dev**. Αφότου ολοκληρωθεί το βήμα αυτό, τρέχουμε την εντολή **sudo apt-get install postgresql-10**, ώστε να εγκατασταθεί η PostgreSQL έκδοση 10, στο σύστημά μας.

4.3 Εγκατάσταση Angular framework

4.3.1 Angular

Σε ένα τερματικό, αρκεί να τρέξουμε την εντολή **npm install -g @angular/cli@7.3.1**. Αυτή η εντολή θα εγκαταστήσει την έκδοση 7.3.1 του framework, που είναι η έκδοση που χρησιμοποιήθηκε στην εφαρμογή αυτή. Η παράμετρος **-g** σημαίνει πως η εγκατάσταση γίνεται globally, δηλαδή θα έχουμε πρόσβαση στο framework σε όλο το σύστημά μας και όχι μόνο για το συγκεκριμένο project.

Κατά τη δημιουργία ενός νέου project, το Angular εγκαθιστά και τα υπόλοιπα πακέτα/dependencies που απαιτούνται για να δημιουργηθεί-τρέξει ένα Angular project.

Για να δημιουργήσουμε μια νέα εφαρμογή με το όνομα «myapp», τρέχουμε την εντολή **ng new myapp**, οπότε και θα δημιουργηθεί ένας φάκελος με το όνομα «myapp», ο οποίος περιέχει τα αρχεία της νέας εφαρμογής.

Για να τρέξει η εφαρμογή μας σε περιβάλλον ανάπτυξης, τρέχουμε την εντολή **ng serve** και πλέον η εφαρμογή μας θα «ζει» στη διεύθυνση <http://localhost:4200>. Αξίζει να σημειώσουμε πως με αυτόν τον τρόπο, με κάθε αλλαγή που κάνουμε στον πηγαίο κώδικα της εφαρμογής, αυτόματα θα ανανεώνεται και η εφαρμογή μας.

Για να δημιουργηθούν τα στατικά αρχεία που απαιτούνται για να τρέξει το framework σε ένα server, χρησιμοποιείται η εντολή **ng build**, οπότε θα γίνει εξαγωγή αυτών των αρχείων στο φάκελο «myapp». Για στατικά αρχεία για περιβάλλον παραγωγής, χρειάζεται να βάλουμε την παράμετρο **--prod**, δηλαδή **ng build --prod**.

4.3.2 Styling

Για την εμφάνιση της εφαρμογής μας, χρησιμοποιήθηκαν το Angular Material και το Bootstrap framework.

Αυτά τα δύο (2) τα εγκαθιστούμε τρέχοντας διαδοχικά τις εντολές:

- **npm install --save @angular/material@7.2.0 @angular/cdk@7.2.0 @angular/animations@7.2.0**
- **npm install --save @ng-bootstrap/ng-bootstrap@4.0.2.**

4.3.3 ngx-captcha

Το reCAPTCHA^[29] είναι ένα σύστημα το οποίο αναπτύχθηκε αρχικά από τους Luis von Ahn, Ben Maurer, Colin McMillen, David Abraham και Manuel Blum στο Πανεπιστήμιο Carnegie Mellon. Εξαγοράστηκε από την Google το Σεπτέμβριο του 2009. Είναι ένα text-based CAPTCHA αλλά με μια σημαντική καινοτομία. Οι λέξεις που εμφανίζονται για την λύση αυτού του CAPTCHA είναι λέξεις από βιβλία τα οποία είναι στο στάδιο της ψηφιοποίησης. Το reCAPTCHA ζητά από τους χρήστες να εισάγουν λέξεις που εμφανίζονται σε παραμορφωμένες εικόνες κειμένου στην οθόνη. Με την υποβολή των δύο λέξεων που απαιτούνται από το reCAPTCHA προστατεύονται και οι ιστοσελίδες από bots, που προσπαθούν να έχουν πρόσβαση σε υπηρεσίες περιορισμένης πρόσβασης και βοηθά στην ψηφιοποίηση βιβλίων.

Το συγκεκριμένο πακέτο είναι μια υλοποίηση για Angular του reCAPTCHA της Google, κάνοντας πιο εύκολη τη χρήση του. Για την εγκατάσταση, τρέχουμε **npm install --save ngx-captcha@6.3.0**.

4.3.4 ngx-toastr

Οι όροι «pop-up notification», «toast», «passive pop-up notification», «desktop notification», «notification bubble» ή απλά «notification»[\[30\]](#), αναφέρονται σε ένα γραφικό στοιχείο ελέγχου που μεταδίδει συγκεκριμένα συμβάντα στον χρήστη, χωρίς να τους αναγκάζει να αντιδρούν αμέσως σε αυτήν την ειδοποίηση, σε αντίθεση με τα συμβατικά πλαίσια αναδυόμενων μηνυμάτων. Τα desktop notifications συνήθως εξαφανίζονται αυτόματα μετά από σύντομο χρονικό διάστημα. Συχνά το περιεχόμενό τους αποθηκεύεται στη συνέχεια σε κάποιο γραφικό στοιχείο, το οποίο επιτρέπει στους χρήστες να έχουν πρόσβαση σε προηγούμενες ειδοποιήσεις σε ύστερο χρόνο.

Το συγκεκριμένο πακέτο παρέχει έτοιμα αναδυόμενα πλαίσια (toasts), με όμορφο σχεδιασμό και animations, για μηνύματα που αφορούν σε συμβάντα της εφαρμογής, προς τον χρήστη. Για την εγκατάσταση, τρέχουμε **npm install --save ngx-toastr@9.1.2**.

4.4 Εγκατάσταση Virtualenv

Το Virtualenv[\[31\]](#) είναι ένα εργαλείο που μας δίνει τη δυνατότητα να δημιουργούμε Virtual environments, καθένα από τα οποία έχει τα δικά του πακέτα Python χωρίς να επεμβαίνει στο global installation κάποιου πακέτου. Είναι δηλαδή isolated. Για την εγκατάστασή του, τρέχουμε την εντολή **sudo pip3 install virtualenv**.

4.5 Εγκατάσταση Django framework

Η εγκατάσταση του Django και των λοιπών πακέτων που απαιτούνται για την εφαρμογή αυτή, θα γίνει εντός ενός εικονικού περιβάλλοντος (Virtualenv). Η χρήση ενός τέτοιου δεν είναι αναγκαστική, αλλά επιλέχθηκε για τους λόγους που εξηγήσαμε παραπάνω.

4.5.1 Ενεργοποίηση Virtualenv

Εφόσον έχουμε εγκαταστήσει το Virtualenv, τρέχουμε την εντολή **virtualenv --python=python3 --always-copy venv**. Αυτή δημιουργεί ένα νέο φάκελο με το όνομα *venv* στο φάκελο που

βρισκόμαστε (το project μας) και σε αυτόν εγκαθιστά ένα νέο εικονικό περιβάλλον με βιβλιοθήκες Python. Με το **--python=python3** δηλώνουμε ότι για το νέο εικονικό περιβάλλον, θέλουμε να χρησιμοποιήσουμε την έκδοση 3+ της Python και όχι τη 2.7 που είναι by default. Με το **--always-copy**, επισημαίνουμε στο Virtualenv να αντιγράψει εκ νέου στο φάκελο venv τα core files/libraries της Python και να μη χρησιμοποιήσει το global installation. Εν συνεχεία, τρέχουμε την εντολή **source venv/bin/activate** και το νέο εικονικό περιβάλλον ενεργοποιείται.

Ενημερωτικά, για να απενεργοποιήσουμε το εικονικό περιβάλλον, τρέχουμε την εντολή **deactivate**.

4.5.2 Django

Η εγκατάσταση γίνεται με την εντολή **pip install Django==2.1.5**, οπότε και θα εγκατασταθεί η έκδοση 2.1.5 του framework.

Για να δημιουργήσουμε ένα νέο project με το όνομα «myproject», τρέχουμε την εντολή **django-admin startproject myproject**. Θα δημιουργηθεί ένα νέος φάκελος με το όνομα «myproject», που θα περιέχει τα κύρια αρχεία του project μας.

Για τη δημιουργία μιας νέας εφαρμογής με το όνομα «myapp» σε αυτό το project, τρέχουμε **python manage.py startapp myapp**. Σαν αποτέλεσμα δημιουργείται νέος φάκελος με το όνομα «myapp», μέσα στο φάκελο «myproject», που περιέχει τα αρχεία της νέας εφαρμογής.

Για να τρέξει το project, πρέπει να ενεργοποιήσουμε τον development server του Django, με την ακόλουθη εντολή: **python manage.py runserver**. Ο server «ζει» στη διεύθυνση *http://localhost:8000*.

Σε προηγούμενη ενότητα αναφέραμε το προαιρετικό admin interface του Django. Για να έχουμε πρόσβαση σε αυτό, το οποίο βρίσκεται στη διεύθυνση *http://localhost:8000/admin*, χρειάζεται να ορίσουμε πρώτα ένα διαχειριστή, ή όπως τον αναφέρει το framework, superuser. Αυτό γίνεται με την εντολή **python manage.py createsuperuser**. Αφού εισάγουμε τα στοιχεία που μας ζητά, ο superuser μας έχει «γεννηθεί» και μπορούμε να έχουμε πρόσβαση στο εν λόγω διαχειριστικό.

4.5.3 Constants

Δηλώθηκαν στο αρχείο *settings.py* οι παρακάτω μεταβλητές, οι οποίες χρησιμοποιούνται σε διάφορα σημεία της εφαρμογής, για να είναι εύκολα προσβάσιμες και εύκολα τροποποιήσιμες:

DAY_SECONDS = 24 * 60 * 60, που είναι τα δευτερόλεπτα που έχει ένα 24ωρο

BASE_URL = 'http://0.0.0.0:8000/', η URL που «ζει» η εφαρμογή

BLOCK_URL = 'block-request/', το endpoint μέσω του οποίου ο χρήστης μπορεί να ακυρώσει το αίτημα που έκανε

NEW_PASS_URL = 'new-password/', το endpoint που χρησιμοποιεί η Γραμματεία ενός τμήματος για να εκδώσει το νέο κωδικό.

Επίσης δημιουργήθηκε ένα αρχείο με όνομα *response_messages.py*, το οποίο περιέχει τα λεκτικά που χρησιμοποιούνται στις απαντήσεις του back-end, χωρισμένα ανά κατηγορία κλήσης (request) και κατάστασης που θα επιστραφεί από το server (HTTP status). Αυτό είναι σημαντικό όπως και στην παραπάνω περίπτωση, δηλαδή τα μηνύματα αυτά να είναι προσβάσιμα από κάθε σημείο της εφαρμογής και εύκολα τροποποιήσιμα.

Συγκεκριμένα:

Generic application error

generic_error = ("Η δημιουργία του αιτήματος απέτυχε. Παρακαλώ δοκιμάστε εκ νέου αργότερα. "

"Αν το πρόβλημα παραμένει, επικοινωνήστε με το Κ.Π.Δ."

)

Block request

block_error_db = ("Η ακύρωση του αιτήματος απέτυχε. Παρακαλώ δοκιμάστε εκ νέου αργότερα. "

"Αν το πρόβλημα παραμένει, επικοινωνήστε με το Κ.Π.Δ."

)

block_error_new_pass = "Ο σύνδεσμος δεν μπορεί να χρησιμοποιηθεί, έχει εκδοθεί νέος κωδικός στις {}"

block_error_used_link = "Ο σύνδεσμος έχει χρησιμοποιηθεί και το αίτημα έχει ακυρωθεί"

block_success_no_email = ("Η ακύρωση του αιτήματος έγινε επιτυχώς. Λόγω συστημικού σφάλματος, δε θα λάβετε επιβεβαιωτικό e-mail στον ιδρυματικό λογαριασμό σας")

block_success = "Η ακύρωση του αιτήματος έγινε επιτυχώς"

New request

new_request_success = "Το αίτημα δημιουργήθηκε επιτυχώς"

invalid_user_name = "Παρακαλώ εισάγετε ένα έγκυρο όνομα χρήστη"

user_not_found = "Ο χρήστης δεν βρέθηκε στον LDAP"

New password

new_password_user_canceled = "Το αίτημα έχει ακυρωθεί από το χρήστη"

new_password_used_link = "Ο σύνδεσμος έχει χρησιμοποιηθεί και έχει εκδοθεί νέος κωδικός, στις {}"

new_password_system_failure = "Η δημιουργία του νέου κωδικού για το χρήστη {} απέτυχε"

new_password_success = "Ο νέος κωδικός δημιουργήθηκε επιτυχώς"

4.5.4 Logging

Για το logging της εφαρμογής χρησιμοποιήθηκε ο ενσωματωμένος logger[[32](#)] του Django, με δική μας παραμετροποίηση, η οποία έγινε στο αρχείο settings.py και φαίνεται παρακάτω:

```
LOGGING = {  
    'version': 1,  
    'disable_existing_loggers': False,  
    'handlers': {  
        'file': {  
            'level': 'DEBUG',  
            'class': 'logging.FileHandler',
```

```

        'filename': '../logs/error.log'
    }
},
'loggers': {
    'django': {
        'handlers': ['file'],
        'level': 'DEBUG',
        'propagate': True
    }
}
}.

```

4.5.5 Αποστολή e-mails

Όταν γίνεται αίτημα νέου κωδικού για έναν χρήστη μέσω της εφαρμογής, αποστέλλεται στο ιδρυματικό e-mail αυτού του χρήστη, e-mail που αφορά στο αίτημα, καθώς και στη Γραμματεία του τμήματος που ανήκει. Για αυτή τη λειτουργικότητα επιλέχθηκε ο ενσωματωμένος e-mail sender[33] του Django. Στο αρχείο *settings.py* χρειάστηκε να γίνουν ορισμένες προσθήκες, όπως φαίνεται παρακάτω.

Προκειμένου να γίνει σωστά η σύνδεση με τον e-mail server του Πανεπιστημίου:

```

EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = 465
EMAIL_HOST_USER = '*****@hua.gr'
EMAIL_HOST_PASSWORD = '*****'
EMAIL_USE_SSL = True.

```

Ορισμός του e-mail για κάθε γραμματεία/τμήμα που αφορά τον εκάστοτε χρήστη που κάνει το αίτημα, αναφέρονται ενδεικτικά:

```

EMAIL_GEO = 'geosec@hua.gr'
EMAIL_IT = 'itsec@hua.gr'
EMAIL_ITP = 'itpsec@hua.gr'
EMAIL_DIET = 'dsec@hua.gr'

```

```
EMAIL_HS = 'hsec@hua.gr'
```

```
EMAIL_NOC = 'noc@hua.gr'.
```

Το σώμα αυτού του e-mail βρίσκεται σε ένα αρχείο *email.html*, στο φάκελο *templates* του Django project μας:

```
<html>
```

```
<head>
```

```
<style>
```

```
.block {
```

```
    display: block;
```

```
}
```

```
.margin-bottom {
```

```
    margin-bottom: 2rem;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```

```

```
<p style="font-size: 0.875rem;" class="margin-bottom">
```

```
    {{ body }}
```

```
</p>
```

```
<p style="font-size: small;">
```

```
<span class="block" style="padding-bottom: 1rem;">
```

```
<small class="block">
```

```
    Κέντρο Πληροφορικής και Δικτύων
```

```
</small>
```

```

<small class="block">

    Χαροκόπειο Πανεπιστήμιο

</small>

</span>

<span class="block">

    <small class="block">

        Network Operation Center

    </small>

    <small class="block">

        Harokopio University

    </small>

</span>

</p>

</body>

</html>

```

Έχουμε προσθέσει custom CSS για τη μορφοποίηση, καθώς και το λογότυπο του Χαροκοπείου Πανεπιστημίου στην κορυφή. Το πεδίο `{{ body }}` είναι δυναμικό, δηλαδή τα περιεχόμενα του κυρίως κειμένου του e-mail αυτού, συμπληρώνονται αυτόματα ανάλογα με την περίπτωση και τον παραλήπτη.

4.5.6 python-ldap

Οι υπηρεσίες καταλόγου είναι βάσεις δεδομένων, οι οποίες οργανώνουν εγγραφές και είναι βελτιστοποιημένες για διαδικασίες ανάγνωσης και αναζήτησης.

Το LDAP (Lightweight Directory Access Protocol)[[34](#)] είναι ένα πρωτόκολλο ανοικτού προτύπου για την πρόσβαση σε υπηρεσίες καταλόγου X.500. Το πρωτόκολλο τρέχει πάνω από το επίπεδο μεταφοράς ενός δικτύου. Στην περίπτωση του Διαδικτύου αυτό είναι το TCP. Χρησιμοποιεί τη

δικτυακή διαστρωμάτωση TCP/IP για τα επίπεδα δικτύου και μεταφοράς, σε αντίθεση με την περίπλοκη διαστρωμάτωση του μοντέλου OSI. Επίσης υιοθετεί και άλλες απλουστεύσεις, όπως η αναπαράσταση τιμών γνωρισμάτων και δομές πληροφορίας του πρωτοκόλλου ως αλφαριθμητικά κειμένου (strings), τα οποία σχεδιάστηκαν ώστε να γίνεται η υλοποίηση περισσότερο απλή και εύκολη.

Το πακέτο **python-ldap** παρέχει ένα αντικειμενοστρεφές API για επικοινωνία με έναν LDAP server. Είναι χτισμένο πάνω στην open-source βιβλιοθήκη OpenLDAP 2.x. Είναι προαπαιτούμενο για τη λειτουργία του πακέτου **django-auth-ldap**. Για την εγκατάσταση, τρέχουμε την εντολή **pip install python-ldap==3.1.0**.

4.5.7 django-auth-ldap

Αυτό το πακέτο χρησιμοποιείται για την αυθεντικοποίηση ενός χρήστη μιας εφαρμογής, με έναν LDAP server. Η αρχικοποίησή του είναι απλή, αλλά παρέχει ένα πλούσιο περιβάλλον για ρυθμίσεις που αφορούν χρήστες, ομάδες και δικαιώματα. Στην εφαρμογή που υλοποιήσαμε χρησιμοποιείται για τον έλεγχο εάν ένας χρήστης είναι ενεργός στον LDAP του Πανεπιστημίου. Για την εγκατάσταση, τρέχουμε την εντολή **pip install django-auth-ldap==1.7.0**.

Εν συνεχεία έπρεπε να τροποποιηθεί το αρχείο settings.py του Django, συγκεκριμένα:

```
AUTHENTICATION_BACKENDS = [  
    'django_auth_ldap.backend.LDAPBackend',  
    'django.contrib.auth.backends.ModelBackend',  
],
```

ώστε να «ξέρει» το Django ότι η αυθεντικοποίηση των χρηστών της εφαρμογής δε θα γίνεται με τον τρόπο που έχει by default το framework, αλλά μέσω ενός LDAP server.

Επιπλέον σε αυτό το αρχείο, προστέθηκαν οι ρυθμίσεις για τον LDAP server που χρησιμοποιεί η εφαρμογή, οι οποίες είναι:

```
# LDAP  
AUTH_LDAP_SERVER_URI = "*****"  
MAIN_DN = "dc=hua,dc=gr"  
USERS_DN = "ou=USERS,ou=DIT," + MAIN_DN  
AUTH_LDAP_BIND_DN = "*****@hua.gr"
```

```
AUTH_LDAP_BIND_PASSWORD = "*****"  
AUTH_LDAP_PERMIT_EMPTY_PASSWORD = False.
```

4.5.8 PyJWT

Αυτό το πακέτο είναι μια Python υλοποίηση του JWT (JSON Web Token). Για την εγκατάσταση, τρέχουμε την εντολή **pip install PyJWT==1.7.1**.

Απαραίτητο ήταν να γίνουν οι παρακάτω προσθήκες στο αρχείο settings.py του Django, συγκεκριμένα η αρχικοποίηση κάποιων σταθερών που χρησιμοποιούνται για τη δημιουργία ενός token της εφαρμογής:

```
# Token  
TOKEN_ISS = "HUA"  
TOKEN_ALGORITHM = "HS256".
```

4.5.9 psycopg2

Το Django, by default, έχει ενσωματωμένη μια SQLite Βάση Δεδομένων. Στην περίπτωση αυτής της εφαρμογής που χρησιμοποιήσαμε την PostgreSQL, εγκαταστάθηκε επιπλέον η βιβλιοθήκη **psycopg2** για τη σύνδεση της τελευταίας με το Django.

Αυτή η βιβλιοθήκη είναι η πιο δημοφιλής υλοποίηση για σύνδεση μιας Βάση Δεδομένων PostgreSQL με τη γλώσσα Python. Βασίζεται στο Python DB API 2.0 specification. Έχει σχεδιαστεί για μεγάλες multi-threaded εφαρμογές, που εκτελούν ταυτόχρονα μεγάλο αριθμό SQL εντολών INSERT και UPDATE. Επίσης είναι UNICODE friendly. Για την εγκατάσταση, τρέχουμε την εντολή **pip install psycopg2==2.8.3**.

Απαραίτητο ήταν να γίνουν οι παρακάτω αλλαγές στο αρχείο settings.py του Django:

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql_psycopg2',  
        'NAME': 'thesis',  
        'USER': 'thesisuser',
```



```
'PASSWORD': '*****',  
  
'HOST': 'localhost',  
  
'PORT': '5432'  
  
}  
  
}.
```

Το κλειδί *ENGINE* δείχνει στο Django ποιος driver έχει χρησιμοποιηθεί για σύνδεση με μια Βάση Δεδομένων. *NAME* είναι το όνομα της Βάση Δεδομένων που χρησιμοποιεί η εφαρμογή. *USER* είναι το όνομα του χρήστη, που κατέχει τα απαραίτητα δικαιώματα, ώστε να διαχειρίζεται αυτή τη Βάση Δεδομένων, αφού έχει εισάγει το σωστό *PASSWORD*. *HOST* είναι ο server στον οποίο βρίσκεται η συγκεκριμένη Βάση Δεδομένων. Στο πεδίο *PORT* εισάγεται ο αριθμός της πόρτας που «ακούει» ο παραπάνω server.

Τα δύο τελευταία βήματα είναι να τρέξουμε τις εντολές **python3 manage.py makemigrations** και **python3 manage.py migrate**. Η πρώτη εντολή μας επιστρέφει ποιες αλλαγές έχουμε κάνει στα μοντέλα της εφαρμογής στο αρχείο `models.py`. Η δεύτερη εφαρμόζει αυτές τις αλλαγές, δηλαδή δημιουργεί τους κατάλληλους πίνακες και σχέσεις στη Βάση Δεδομένων που έχουμε συνδέσει με το Django.

5. ΟΔΗΓΟΣ ΧΡΗΣΗΣ

Η εφαρμογή θα φιλοξενείται σε server του Χαροκοπέιου Πανεπιστημίου και θα έχει δημόσια URL. Στη συνέχεια θα παρατεθούν εικόνες από τις οθόνες της εφαρμογής και επεξηγήσεις για κάθε μία, οι οποίες είναι σε περιβάλλον παραγωγής αλλά τρέχουν σε τοπικό server.

5.1 Landing page

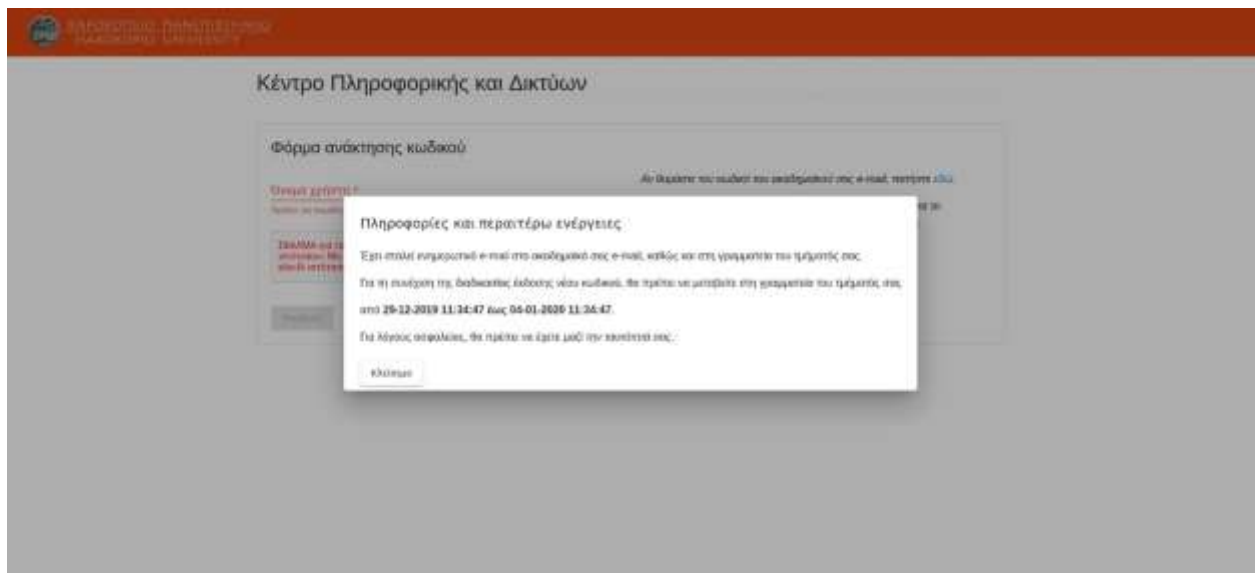
Η landing page χρησιμοποιείται μόνον από κάποιον χρήστη του Πανεπιστημίου που θέλει να κάνει αίτημα έκδοσης νέου κωδικού και όχι από τη Γραμματεία κάποιου τμήματος. Περιέχει μια φόρμα με μόνο ένα (1) πεδίο, το κουμπί υποβολής καθώς και έναν μηχανισμό anti-spam, συγκεκριμένα το reCAPTCHA της Google. Στο πεδίο αυτό ο χρήστης θα πρέπει να εισάγει το

username του, π.χ. itr12345, χωρίς το @hua.gr. Αν για κάποιο λόγο εισάγει ολόκληρο το ιδρυματικό του e-mail, η εφαρμογή κρατάει μόνο το κομμάτι με το username. Στη συνέχεια θα πρέπει να κάνει κλικ στο κουτάκι «Δεν Είμαι Ρομπότ» και θα ανοίξει ένα παράθυρο διαλόγου, όπου ο χρήστης θα πρέπει να κάνει «κλικ» μόνο στις εικόνες με το περιεχόμενο που του αναφέρει αυτό το παράθυρο. Αφού γίνει αυτό επιτυχώς, ενεργοποιείται το κουμπί της υποβολής. Μόλις το πατήσει ο χρήστης, εμφανίζεται ένα νέο παράθυρο διαλόγου με πληροφορίες για τη συνέχιση της διαδικασίας έκδοσης νέου κωδικού. Παράλληλα αποστέλλονται δύο (2) e-mails, ένα στη Γραμματεία του τμήματος του χρήστη με το αίτημά του και άλλο ένα στο ιδρυματικό e-mail του ίδιου με ένα σύνδεσμο ακύρωσης του αιτήματος, για την περίπτωση που άλλαξε γνώμη ή δεν πραγματοποίησε εκείνος το αίτημα.



Εικ 6 Landing page

Σε αντίθετη περίπτωση που ο χρήστης γνωρίζει/θυμάται τον κωδικό του ιδρυματικού του e-mail, έχει τη δυνατότητα μέσω άλλης ιστοσελίδας, συγκεκριμένα <https://ssaml.hua.gr/>, να αλλάξει τον κωδικό του e-mail και αυτομάτως αυτός ο νέος κωδικός να ισχύει και για τον ιδρυματικό του λογαριασμό. Για τη διαδικασία αυτή παρέχεται σύνδεσμος, στο δεξί μέρος της landing page.



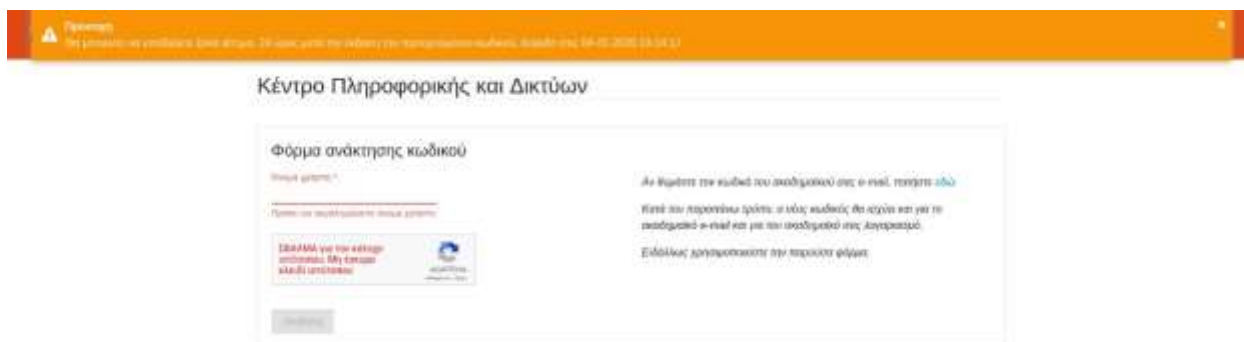
Εικ 7 Πληροφορίες συνέχισης διαδικασίας

Στην περίπτωση που ένας χρήστης έχει ενεργό αίτημα και προσπαθήσει να κάνει ένα εκ νέου, η διαδικασία δε συνεχίζεται και εμφανίζεται σχετικό μήνυμα που του αναφέρει πως υπάρχει ήδη ενεργό αίτημα και μέχρι ποια ημερομηνία θα είναι έγκυρο.



Εικ 8 Υπάρχον ενεργό αίτημα

Εάν ένας χρήστης προσπαθήσει να κάνει νέο αίτημα πριν περάσουν 24 ώρες από την έκδοση του τελευταίου κωδικού που έλαβε, η διαδικασία σταματά και ενημερώνεται σχετικά.



Εικ 9 Νέο αίτημα εντός 24 ωρών από την έκδοση κωδικού

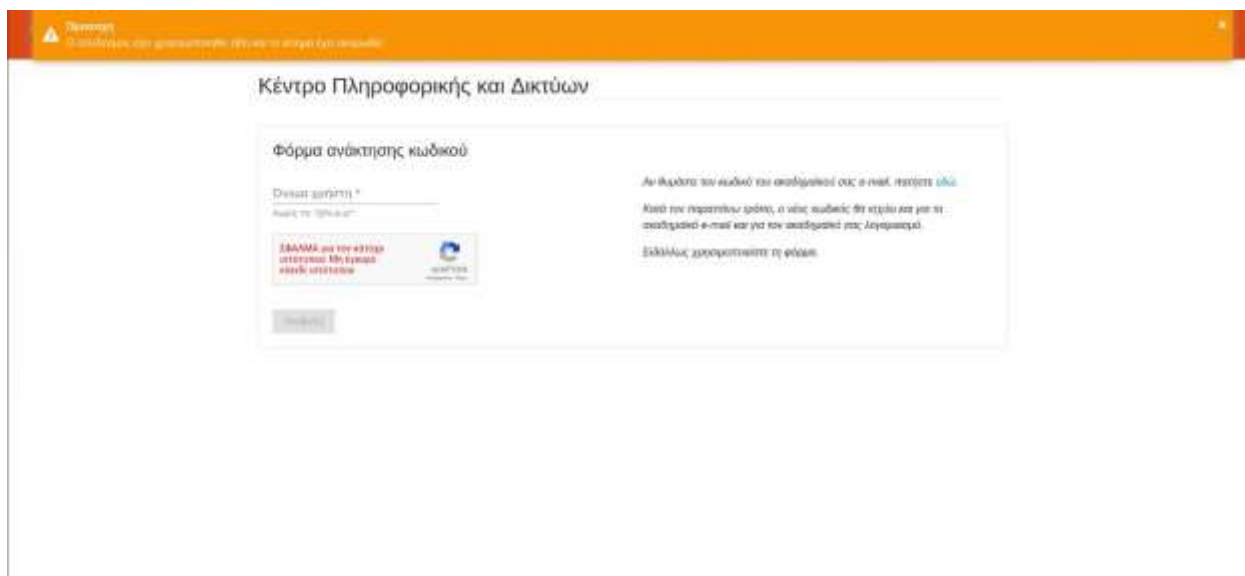
5.2 Οθόνη ακύρωσης αιτήματος από το χρήστη

Αν ένας χρήστης βρει/θυμηθεί τον κωδικό του ή δεν έχει πραγματοποιήσει εκείνος το αίτημα, μπορεί να κάνει ακύρωση αυτού μέσω του συνδέσμου που έλαβε στο ιδρυματικό του e-mail κατά τη δημιουργία του αιτήματος. Μετά τη χρήση του συνδέσμου, μεταφέρεται στην παρακάτω οθόνη.



Εικ 10 Επιτυχής ακύρωση

Αν ο χρήστης ακυρώσει το αίτημα και σε επόμενο χρόνο χρησιμοποιήσει πάλι το σύνδεσμο ακύρωσης που είχε λάβει, εμφανίζεται σχετικό μήνυμα πως ο σύνδεσμος έχει ήδη χρησιμοποιηθεί και η διαδικασία σταματά.



Εικ 11 Χρήση του ίδιου συνδέσμου 2η φορά

Στην περίπτωση που η διαδικασία ολοκληρωθεί και ο χρήστης λάβει το νέο κωδικό, αλλά στη συνέχεια χρησιμοποιήσει το σύνδεσμο ακύρωσης του αιτήματος, κάτι που δεν είναι εφικτό, ενημερώνεται πως έχει εκδοθεί ήδη νέος κωδικός για το αίτημα αυτό, καθώς και την ημερομηνία που έγινε αυτό.

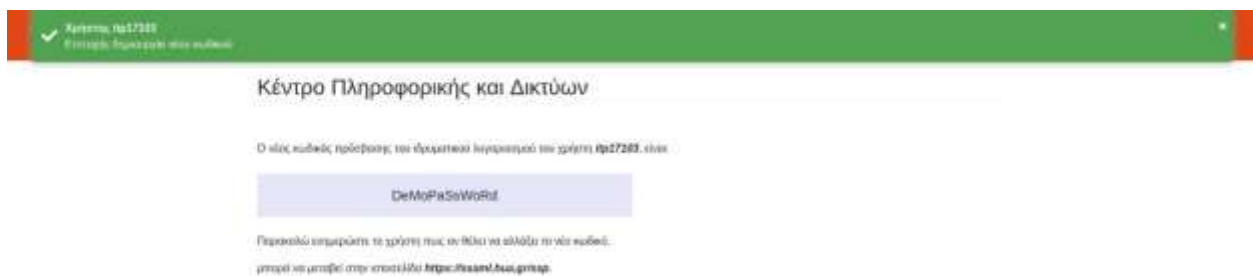


Εικ 12 Ακύρωση έπειτα από έκδοση νέου κωδικού

5.3 Οθόνη Γραμματείας

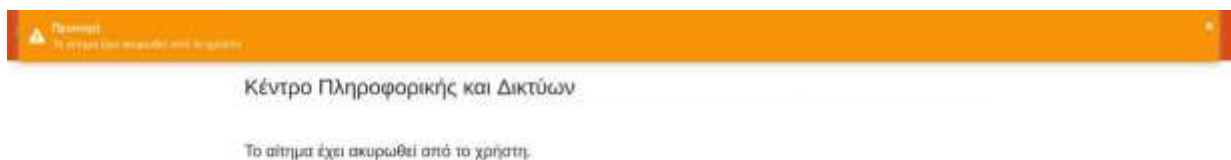
Αυτή είναι η μοναδική οθόνη της εφαρμογής, στην οποία έχουν πρόσβαση οι Γραμματείες των τμημάτων.

Όταν μεταβεί ένας χρήστης που έχει κάνει αίτημα έκδοσης νέου κωδικού στη Γραμματεία του τμήματός του, ο χειριστής του PC της Γραμματείας χρησιμοποιεί το σύνδεσμο που έχει λάβει με e-mail στο λογαριασμό της Γραμματείας, για το συγκεκριμένο χρήστη. Εμφανίζεται η παρακάτω οθόνη με το νέο κωδικό που θα κοινοποιήσει στο χρήστη.



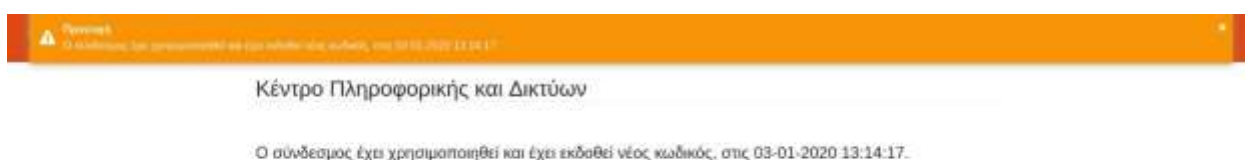
Εικ 13 Οθόνη νέου κωδικού

Αν ο χρήστης έχει ακυρώσει το αίτημα αλλά έπειτα μεταβεί στη γραμματεία για να παραλάβει νέο κωδικό, εμφανίζεται σχετικό μήνυμα, όπως φαίνεται στην παρακάτω εικόνα.



Εικ 14 Ο χρήστης έχει πρώτα ακυρώσει το αίτημα

Εάν ο χειριστής της Γραμματείας προσπαθήσει να χρησιμοποιήσει για δεύτερη φορά έναν σύνδεσμο, κάτι που δεν επιτρέπεται, εμφανίζεται προειδοποιητικό μήνυμα πως ο σύνδεσμος έχει ήδη χρησιμοποιηθεί.



Εικ 15 Δεύτερη χρήση ιδίου συνδέσμου

6. ΣΥΜΠΕΡΑΣΜΑΤΑ

Έχοντας πλέον μια ολοκληρωμένη και production-ready υλοποίηση, μπορούμε να συμπεράνουμε πως η δημιουργία μιας εφαρμογής τέτοιου τύπου δεν είναι πάντα εύκολη, γιατί θα πρέπει κανείς να λάβει υπόψη του πολλές παραμέτρους, αλλά και να βρει μια ισορροπία ανάμεσα στις ανάγκες του χρήστη και το τι πραγματικά θα είναι πρακτικό και εύχρηστο. Επιπρόσθετα, το να δημιουργεί κανείς μια εφαρμογή που θα χρησιμοποιηθεί άμεσα και από αρκετούς χρήστες, δημιουργεί την ανάγκη αυτή η εφαρμογή να λειτουργεί όσο το δυνατόν καλύτερα.

Τα οφέλη αυτής της νέας εφαρμογής, είναι ένα user friendly περιβάλλον με τα κατάλληλα μηνύματα προς το χρήστη, όποτε χρειάζεται, μέσω του οποίου θα πραγματοποιούνται στο εξής τα αιτήματα για έκδοση νέου ιδρυματικού κωδικού, από το σύνολο των χρηστών του Χαροκοπέιου Πανεπιστημίου. Οι λειτουργίες που αξίζει να αναφερθούν είναι:

- η δυνατότητα σε έναν χρήστη να πραγματοποιεί το αίτημα ηλεκτρονικά και απομακρυσμένα, χωρίς σύνδεση μέσω VPN
- η προστασία από «spam» αιτήματα
- η δυνατότητα ακύρωσης του αιτήματος από το χρήστη που το πραγματοποίησε
- η αυτόματη αναζήτηση στον LDAP του Πανεπιστημίου του χρήστη που κάνει το αίτημα αν είναι ενεργός, ώστε να μη χάνει χρόνο η Γραμματεία για την ίδια εργασία
- η εξάλειψη της ανάγκης φυσικής επικοινωνίας της Γραμματείας με το Κέντρο Πληροφορικής και Δικτύων

Το Πανεπιστήμιο θα επωφεληθεί από αυτήν την εφαρμογή, καθώς η διαδικασία έκδοσης νέου κωδικού, από το αίτημα έως την έκδοση, θα γίνεται αυτοματοποιημένα, με μόνη εξαίρεση τη φυσική παρουσία του χρήστη που έχει κάνει το αίτημα, στη Γραμματεία. Το τελευταίο θα γίνεται για λόγους ασφαλείας-ταυτοποίησης του χρήστη.

6.1 Επεκτάσεις

Επόμενο βήμα θα ήταν η αναβάθμιση του Django framework στην τελευταία σταθερή έκδοση, η οποία έχει επιλύσει ορισμένα ζητήματα ασφάλειας. Τη στιγμή της εκπόνησης αυτής της διπλωματικής, η τελευταία έκδοση ήταν και αυτή που χρησιμοποιήθηκε, η 2.1.5.

Στην έκδοση της εφαρμογής που περιγράψαμε, κάθε Γραμματεία λαμβάνει e-mails με αιτήματα και θα πρέπει να ψάξει να βρει το e-mail με το αίτημα του χρήστη, όταν ο τελευταίος παρουσιαστεί για ταυτοποίηση. Μελλοντικά, θα μπορούσε να δημιουργηθεί μια οθόνη ειδικά για τις Γραμματείες, όπου θα μπορούσαν να βλέπουν τα αιτήματα συνολικά των χρηστών, να μπορούν να τα φιλτράρουν βάσει status και να μπορούν να κάνουν αναζήτηση με βάση το username ενός χρήστη. Σε αυτό το σενάριο θα χρειαστεί να δημιουργηθεί ένας νέος ρόλος χρήστη στην εφαρμογή, προκειμένου να εξασφαλισθεί πως μόνο ο χειριστής μιας Γραμματείας θα μπορεί να εισέλθει σε αυτήν τη νέα οθόνη, με τα credentials του.

7. ΒΙΒΛΙΟΓΡΑΦΙΑ - ΑΝΑΦΟΡΕΣ

[1] SPA (Single-page Application):

https://en.wikipedia.org/wiki/Single-page_application

[2] JavaScript:

<https://en.wikipedia.org/wiki/JavaScript>

[3] ECMA Script:

<https://en.wikipedia.org/wiki/ECMAScript>

[4] Angular framework:

[https://en.wikipedia.org/wiki/Angular_\(web_framework\)](https://en.wikipedia.org/wiki/Angular_(web_framework))

[5] TypeScript:

<https://en.wikipedia.org/wiki/TypeScript>

[6] Bootstrap:

[*https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)*](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))

[7] Material Design:

[*https://en.wikipedia.org/wiki/Material_Design*](https://en.wikipedia.org/wiki/Material_Design)

[8] JWT (JSON Web Token):

[*https://en.wikipedia.org/wiki/JSON_Web_Token*](https://en.wikipedia.org/wiki/JSON_Web_Token)

[9] Python:

[*https://en.wikipedia.org/wiki/Python_\(programming_language\)*](https://en.wikipedia.org/wiki/Python_(programming_language))

[10] Django framework:

[*https://en.wikipedia.org/wiki/Django_\(web_framework\)*](https://en.wikipedia.org/wiki/Django_(web_framework))

[11] D.R.Y.:

[*https://en.wikipedia.org/wiki/Don%27t_repeat_yourself*](https://en.wikipedia.org/wiki/Don%27t_repeat_yourself)

[12] The Model-View-Controller Design Pattern:

[*https://djangobook.com/model-view-controller-design-pattern*](https://djangobook.com/model-view-controller-design-pattern)

[13] ORM (Object-relational Mapping):

[*https://en.wikipedia.org/wiki/Object-relational_mapping*](https://en.wikipedia.org/wiki/Object-relational_mapping)

[14] PostgreSQL:

[*https://en.wikipedia.org/wiki/PostgreSQL*](https://en.wikipedia.org/wiki/PostgreSQL)

[15] ACID:

[*https://en.wikipedia.org/wiki/ACID*](https://en.wikipedia.org/wiki/ACID)

[16] JetBrains:

[*https://en.wikipedia.org/wiki/JetBrains*](https://en.wikipedia.org/wiki/JetBrains)

[17] PyCharm:

[*https://en.wikipedia.org/wiki/JetBrains#PyCharm*](https://en.wikipedia.org/wiki/JetBrains#PyCharm)

[18] WebStorm:

[*https://en.wikipedia.org/wiki/JetBrains#WebStorm*](https://en.wikipedia.org/wiki/JetBrains#WebStorm)

[19] REST (Representational State Transfer):

[*https://en.wikipedia.org/wiki/Representational_state_transfer*](https://en.wikipedia.org/wiki/Representational_state_transfer)

[20] Use case diagram:

[*https://en.wikipedia.org/wiki/Use_case_diagram*](https://en.wikipedia.org/wiki/Use_case_diagram)

[21] Activity diagram:

[*https://en.wikipedia.org/wiki/Activity_diagram*](https://en.wikipedia.org/wiki/Activity_diagram)

[22] Class diagram:

[*https://en.wikipedia.org/wiki/Class_diagram*](https://en.wikipedia.org/wiki/Class_diagram)

[23] Deployment diagram:

[*https://en.wikipedia.org/wiki/Deployment_diagram*](https://en.wikipedia.org/wiki/Deployment_diagram)

[24] Ubuntu:

[*https://en.wikipedia.org/wiki/Ubuntu*](https://en.wikipedia.org/wiki/Ubuntu)

[25] Git:

[*https://en.wikipedia.org/wiki/Git*](https://en.wikipedia.org/wiki/Git)

[26] npm:

[*https://en.wikipedia.org/wiki/Npm_\(software\)*](https://en.wikipedia.org/wiki/Npm_(software))

[27] Node.js:

[*https://en.wikipedia.org/wiki/Node.js*](https://en.wikipedia.org/wiki/Node.js)

[28] PIP (Pip Installs Packages):

[*https://en.wikipedia.org/wiki/Pip_\(package_manager\)*](https://en.wikipedia.org/wiki/Pip_(package_manager))

[29] reCAPTCHA:

[*https://en.wikipedia.org/wiki/ReCAPTCHA*](https://en.wikipedia.org/wiki/ReCAPTCHA)

[30] Toast:

[*https://en.wikipedia.org/wiki/Pop-up_notification*](https://en.wikipedia.org/wiki/Pop-up_notification)

[31] Virtualenv:

[*https://virtualenv.pypa.io/en/latest/*](https://virtualenv.pypa.io/en/latest/)

[32] Django logger:

[*https://docs.djangoproject.com/en/2.1/topics/logging/*](https://docs.djangoproject.com/en/2.1/topics/logging/)

[33] Django e-mail sender:

[*https://docs.djangoproject.com/en/2.1/topics/email/*](https://docs.djangoproject.com/en/2.1/topics/email/)

[34] LDAP:

[*https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol*](https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol)