

Ατομική Διπλωματική Εργασία

**ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΙΣΗΣ ΠΛΑΤΦΟΡΜΑΣ
ΓΙΑ ΤΟ ΜΑΘΗΜΑ ΤΕΧΝΟΛΟΓΙΑΣ ΛΟΓΙΣΜΙΚΟΥ**

Άντρια Παναγιώτου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2021

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αανάπτυξη συστήματος διαχείρισης πλατφόρμας για το μάθημα
τεχνολογίας λογισμικού
Άντρια Παναγιώτου

Επιβλέπουσα Καθηγήτρια
Γεωργία Καπιτσάκη

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2021

Ευχαριστίες

Η παρούσα ατομική διπλωματική εργασία εκπονήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης πτυχίου στον κλάδο της Πληροφορικής του Πανεπιστημίου Κύπρου ωστόσο η πραγματοποίησή της δεν θα ήταν εφικτή χωρίς την βοήθεια κάποιων σημαντικών ανθρώπων.

Πρωτίστως θα ήθελα να εκφράσω τις ευχαριστίες μου στην υπεύθυνη επιβλέπουσα καθηγήτρια Δρ. Γεωργία Καπιτσάκη για την εμπιστοσύνη και την ευκαιρία που μου έδωσε να ασχοληθώ με το συγκεκριμένο θέμα, αλλά και για την ανατροφοδότηση και τις συμβουλές που μου παρείχε καθ' όλη την διάρκεια. Στοιχεία τα οποία συνέβαλαν στο μέγιστο, κατά την άποψη μου, για την επιτυχή ολοκλήρωση της Διπλωματική Εργασίας

Επίσης, θα ήθελα να ευχαριστήσω τους συμφοιτητές μου και γενικά τον κάθε ένα που αφιέρωσε λίγο χρόνο για να συμπληρώσει τα ερωτηματολόγια τα οποία δημιούργησα με σκοπό τον καθορισμό απαιτήσεων και την αξιολόγηση της πλατφόρμας.

Κλείνοντας, θέλω να εκφράσω ένα μεγάλο ευχαριστώ στην οικογένεια μου για την αγάπη, την υποστήριξη και το κουράγιο που μου παρείχε για όλα αυτά τα χρόνια των σπουδών μου και όχι μόνο. Τέλος, ευχαριστώ τους κοντινούς μου φίλους οι οποίοι με υπομονή και κουράγιο είναι πάντα δίπλα μου και με στηρίζουν σε κάθε μου βήμα.

Περίληψη

Με την πάροδο του χρόνου η τεχνολογία και ειδικά η επιστήμη της πληροφορικής εισέρχεται όλο και περισσότερο στην ζωή μας αυτοματοποιώντας λειτουργίες που στο παρελθόν αποτελούσαν λειτουργίες ρουτίνας κυρίως χειροκίνητες. Ένας από τους τομείς που δέχτηκαν αλλαγές και βελτιωτικές κινήσεις είναι ο τομέας της εκπαίδευσης όπου με την εισαγωγή της νέας τεχνολογίας πληροφοριών και επικοινωνιών (ΤΠΕ) έχει αλλάξει σημαντικά τη μορφή και τη λειτουργία της. Στις εφαρμογές αυτές κατηγοριοποιείται και η πλατφόρμα διαχείρισης του μαθήματος ΕΠΑ343 – Software Engineering η οποία υλοποιήθηκε στα πλαίσια της διπλωματικής μου εργασίας.

Η διπλωματική μου εργασίας έχει ως θέμα ανάπτυξης μίας διαδικτυακής εφαρμογής η οποία θα χρησιμοποιείται για σκοπό διαχείρισης του μαθήματος ΕΠΑ343 από τους υπεύθυνους καθηγητές, εγγεγραμμένους φοιτητές και οργανισμούς οι οποίοι είναι πρόθυμοι να αναθέσουν ιδέες έργων τα οποία θα τους ενδιέφεραν να υλοποιηθούν. Αυτή η ιδέα προέκυψε από την έλλειψη λογισμικού το οποίο θα μπορούσε να ανταπεξέλθει σε όλες τις απαιτήσεις του μαθήματος και υποχρεώσεων του φοιτητή χωρίς να αναγκάζεται ο καθηγητής να χρησιμοποιεί πληθώρα λογισμικών για να ικανοποιήσει τις ανάγκες του μαθήματος.

Σκοπός λοιπόν της υλοποίησης της συγκεκριμένης πλατφόρμας είναι η συσπείρωση όλων των απαραίτητων λειτουργιών οι οποίες είναι απαραίτητες για την διαχείριση του μαθήματος σε μία συγκεκριμένη εφαρμογή. Πολλές από τις λειτουργίες ίσως υπήρχαν ήδη υλοποιημένες σε αλλά διάφορα course management συστήματα αναγκάζοντας τον καθηγητή και τους φοιτητές να επισκέπτονται καθημερινά διάφορες εφαρμογές με τον κίνδυνο να συγχύσουν πληροφορίες ή από την πλευρά των φοιτητών να ξεχάσουν κάποια υποχρέωση τους. Συμπερασματικά, η δημιουργία μίας τέτοια πλατφόρμας θα πρόσφερε την ευκολία τόσο σε μαθητές όσο και σε καθηγητές να έχουν εύκολη πρόσβαση στις λειτουργίες, θα αποθήκευε όλες τις πληροφορίες σε μία βάση δεδομένων για το μάθημα και θα αυτοματοποιούσε λειτουργίες.

Στα πλαίσια της διπλωματικής εργασίας μέχρι το παρόν στάδιο καταφέραμε να υλοποιήσουμε την πλατφόρμα αυτή με τις κυριότερες ανάγκες του μαθήματος να είναι διαθέσιμες και να εκτελούνται πλήρως με τα αναμενόμενα αποτελέσματα. Ίσως σε κάποιο μεταγενέστερο στάδιο γίνει επέκταση των λειτουργιών με κάποιες προσθήκες οι οποίες θα διευκόλυναν ακόμη περισσότερο το μάθημα και ιδιαίτερα τον καθηγητή.

Περιεχόμενα

Εισαγωγή	1
1.1 Εισαγωγή.....	1
1.2 Σκοπός Διπλωματικής Εργασίας	1
1.3 Δομή Διπλωματικής Εργασίας	3
1.4 Έρευνα για παρόμοια συστήματα	4
1.5 Ακρόνυμα και Συντομογραφίες.....	7
Ανάλυση Απαιτήσεων και Προδιαγραφών	10
2.1. Εισαγωγή.....	10
2.1.1. Μοντέλο Κύκλου Ζωής	10
2.1.2. Προσδιορισμός Τυπικών Χρηστών	13
2.1.3. Περιβάλλον Εργασίας.....	16
2.1.4. Περιορισμοί και Παραδοχές συστήματος.....	17
2.2. Προσδιορισμός Αναγκών	18
2.2.1. Ανάλυση Ερωτηματολογίου	18
2.2.2. Γενικά Συμπεράσματα	18
2.3. Ανάλυση Αναγκών.....	19
2.3.1. Personas	19
2.3.2. Σενάρια Χρήσης.....	21
2.4. Καθορισμός Κανόνων Σχεδιασμού και Ευχρηστίας	25
2.5. Καθορισμός Απαιτήσεων.....	25
2.5.1. Εισαγωγή	25
2.5.2. Λειτουργικές Απαιτήσεις	26
2.5.3. Μη λειτουργικές Απαιτήσεις	30
2.6. Διάγραμμα Ροής Εμπειρίας Χρήστη	33
2.6.1. Διαγράμματα ροής υποσελίδα καθηγητή	34
2.6.2. Διαγράμματα ροής υποσελίδα οργανισμού.....	35
2.6.3. Διαγράμματα ροής υποσελίδα φοιτητή	35
2.7. Σχεδίαση Βάσης Δεδομένων	35
2.7.1. Διάγραμμα Οντοτήτων Συσχετίσεων.....	36
2.7.2. Πίνακες Βάσης Δεδομένων	39
2.7.3. Χωρητικότητα Βάσης Δεδομένων.....	42
Περιγραφή Συστήματος	44
3.1. Γενική Περιγραφή Συστήματος.....	44
3.2. Περιορισμοί.....	45
3.3. Αρχιτεκτονική Συστήματος.....	46
3.4. Επιλογή Κατάλληλων Εργαλείων και Τεχνολογιών	47
3.5. Εργαλεία ανάπτυξης.....	50
3.5.1. Λογισμικό Ανάπτυξης.....	50
3.5.1.1. NPM	50
3.5.1.2. Visual Studio Code.....	51
3.5.2. Γλώσσες Προγραμματισμού και Τεχνολογίες	51
3.5.2.1. HTML	51
3.5.2.2. CSS	52
3.5.2.3. JavaScript	52
3.5.2.4. Material UI	53
3.5.2.5. Node JS	53
3.5.2.6. React JS.....	54
3.5.2.7. Express JS.....	54
3.5.2.8. React Redux.....	54

3.5.2.9. Axios	55
Λειτουργίες Συστήματος.....	56
4.1. Κοινές λειτουργίες συστήματος	58
4.1.1. Σύνδεση	58
4.1.2. Λειτουργία επανάκτησης κωδικού πρόσβασης.....	59
4.1.3. Λειτουργία Αποσύνδεσης.....	60
4.1.4. Λειτουργία αλλαγής γλώσσας εμφάνισης περιεχομένου σελίδας.....	60
4.1.5. Λειτουργία διαχείρισης ειδοποιήσεων λογαριασμών	61
4.1.6. Λειτουργία αλλαγής κωδικού πρόσβασης.....	61
4.1.7. Λειτουργία προβολής και αποθήκευσης στοιχείων GDPR.....	62
4.1.8. Λειτουργία αλλαγής στοιχεία προφίλ	63
4.2. Διεπαφή Οργανισμού	63
4.2.1. Δημιουργία Λογαριασμού	63
4.2.2. Έργα Ανάθεσης.....	64
4.2.2.1. Περιορισμένη εμφάνιση αποτελεσμάτων.....	64
4.2.2.2. Δημιουργία καινούριου Ε.Α.	65
4.2.2.3. Επεξεργασία στοιχείων Ε.Α.	65
4.2.2.4. Προβολή ενδιαφερόμενων ομάδων για συγκεκριμένο Ε.Α.	66
4.2.2.5. Διαγραφή Ε.Α.	66
4.2.3. Προφίλ	67
4.2.3.1. Επεξεργασία στοιχείων προφίλ.....	67
4.2.3.2. Μεταφόρτωση εικόνας προφίλ - λογότυπο	68
4.3. Διεπαφή Φοιτητή	69
4.3.1. Κεντρική Σελίδα	69
4.3.2. Παραδοτέα.....	70
4.3.3. Ερωτηματολόγια.....	71
4.3.4. Έργα Ανάθεσης.....	71
4.3.4.1. Λειτουργία ένδειξης ενδιαφέροντος σε Ε.Α.....	72
4.3.5. Βαθμοί.....	73
4.3.6. Ομάδες Φοιτητών.....	73
4.3.6.1. Προβολή ομάδας και ενέργειες ομάδας.....	74
4.3.6.2. Αποστολή αιτήματος σε ομάδες για συνένωση	74
4.3.6.3. Αποστολή αιτήματος σε συγκεκριμένο άτομο για συνένωση.....	74
4.3.6.4. Αιτήματα που έχουν ήδη ληφθεί.....	75
4.3.7. Προφίλ	76
4.3.7.1. Προσθήκη και αφαίρεση γνώσεων.....	77
4.4. Διεπαφή Καθηγητή	77
4.4.1. Κεντρική Σελίδα	77
4.4.2. Παραδοτέα.....	78
4.4.2.1. Δημιουργία καινούριου παραδοτέου	79
4.4.2.2. Ενέργειες διαγραφής και περιορισμού πρόσβασης.....	79
4.4.3. Ερωτηματολόγια.....	80
4.4.3.1. Δημιουργία καινούριου ερωτηματολόγιου	81
4.4.3.2. Ενέργειες διαγραφής και περιορισμού πρόσβασης.....	81
4.4.4. Έργα Ανάθεσης.....	82
4.4.4.1. Περιορισμένη εμφάνιση αποτελεσμάτων.....	82
4.4.4.2. Δημιουργία καινούριου Ε.Α.	82
4.4.4.3. Επεξεργασία στοιχείων Ε.Α.	83
4.4.4.4. Προβολή ενδιαφερόμενων ομάδων για συγκεκριμένο Ε.Α.	83
4.4.4.5. Διαγραφή Ε.Α.	84
4.4.5. Βαθμοί.....	85
4.4.5.1. Ανάρτηση βαθμών για συγκεκριμένο φοιτητή.....	86

4.4.5.2.	Ανάρτηση βαθμών για ομάδα ατόμων	86
4.4.5.3.	Προβολή βαθμών τρέχοντος ακαδημαϊκού έτους	86
4.4.6.	Ομάδες Φοιτητών	88
4.4.6.1.	Δημιουργία ομάδων φοιτητών	89
4.4.6.2.	Επεξεργασία ομάδων φοιτητών	89
4.4.6.3.	Εμφάνιση φοιτητών χωρίς ομάδα	90
4.4.6.4.	Διαγραφή / Επικύρωση ομάδας φοιτητών	90
4.4.7.	Δημιουργία λογαριασμών τύπου φοιτητή	90
4.4.7.1.	Μεταφόρτωση αρχείου CSV	91
4.4.7.2.	Προβολή φοιτητών	91
4.5.	Ενδεικτικά σημεία υλοποίησης	92
4.5.1.	Εισαγωγή συστατικών (components), εικονιδίων και λειτουργιών	92
4.5.2.	Εισαγωγή στοιχείων και τιμών από το state Redux	93
4.5.2.1.	Περιγραφή υλοποίησης ‘selectedlan’	93
4.5.2.2.	Περιγραφή υλοποίησης ‘loginsession’	93
4.5.3.	Μετάφραση σελίδας	94
4.5.4.	Εμφάνισης ενημερωτικών μηνυμάτων	94
4.5.5.	Δομή σελίδας προφίλ	95
4.5.6.	Εμφάνιση κατάλληλου πίνακα αλλαγής στοιχείων χρήστη	96
4.5.7.	Εμφάνιση δευτέρας λειτουργίας για οργανισμό και φοιτητή	96
4.5.8.	Αρχειοποίηση μεταβλητών σελίδας	96
4.5.9.	Εισαγωγή στοιχείων από Β.Δ.	97
4.5.10.	Ράβδος ανάδρασης	97
4.5.11.	Έλεγχος φόρμας επεξεργασίας στοιχείων προφίλ	98
4.5.12.	Ανανέωση γνώσεων στην Β.Δ.	98
4.5.13.	Περιγραφή προσθήκης και αφαίρεσης γνώσεων	99
4.5.14.	Μεταφόρτωση εικόνας προφίλ	100
4.5.15.	Διαγραφή ανεβασμένου αρχείου – εικόνας προφίλ	101
	Έλεγχος και Αξιολόγηση Συστήματος	102
5.1.	Εισαγωγή	102
5.2.	Δοκιμή φυλλομετρητών ιστών	102
5.3.	Έλεγχος Λειτουργιών	103
5.3.1.	Χειροκίνητες δοκιμές	103
5.3.2.	Χρήση εργαλείου αυτοματοποιημένων δοκιμών	107
5.3.3.	Δοκιμές από άτομα διαφόρων ηλικιών, γνώσεων και επαγγελματιών	108
5.4.	Αποτελέσματα Ερωτηματολόγιου Αξιολόγησης Πλατφόρμας	108
5.5.	Καθορισμός Κανόνων Σχεδιασμού και Ευχρηστίας	108
5.6.	Αξιολόγηση Συστήματος	112
	Συμπεράσματα	114
6.1.	Επεκτασιμότητα Συστήματος	114
6.2.	Ανασκόπηση Διπλωματικής Εργασίας	115
6.3.	Γενικά Συμπεράσματα	116
A.1.	Ερωτηματολόγιο	A-1
A.2.	Γραφικές Παραστάσεις	A-3
B.1.	Ερωτηματολόγιο	B-1
B.2.	Απαντήσεις	B-6
Γ.1.	Ερωτηματολόγιο	Γ-1

Κεφάλαιο 1

Εισαγωγή

1.1 Γενική Εισαγωγή

1.2 Σκοπός Διπλωματικής Εργασίας

1.3 Δομή Διπλωματικής Εργασίας

1.4 Έρευνα για παρόμοια συστήματα

1.6 Ορισμοί, Ακρώνυμα και Συντομογραφίες

1.1 Εισαγωγή

Στο πλαίσιο της διπλωματικής μου εργασίας έχει υλοποιηθεί μία διαδικτυακή εφαρμογή για την διαχείριση του μαθήματος ΕΠΛ343 – Τεχνολογίες διαδικτύου. Τα τελευταία χρόνια ειδικά με την εμφάνιση της πανδημίας COVID-19 στις ζωές μας η εκπαίδευση άλλαξε τρόπο και όλα πλέον διαχειρίζονται από το διαδίκτυο. Το γεγονός αυτό έρχεται να προσθέσει σημαντικότητα στο λόγο ύπαρξης της συγκεκριμένης πλατφόρμας καθώς θα είναι πολύ πιο εύκολο ο καθηγητής να αναθέτει τα πάντα σε μία σελίδα και να προτρέπει τους φοιτητές να την επισκεφτούν έτσι ώστε να καλύψουν τις υποχρεώσεις τους και να βρουν τις απαντήσεις σε αρκετές από τις απορίες τους.

Είναι μία πολύ πρωτότυπη ιδέα ιστοσελίδας καθώς πολύ λίγα μαθήματα στις μέρες μας έχουν ιστοσελίδα με δυναμικό περιεχόμενο το οποίο εναλλάσσεται για κάθε είδος χρήστη και είναι ακριβώς στις απαιτήσεις του μαθήματος. Τα περισσότερα μαθήματα χρησιμοποιούν Course Management Systems τα οποία είναι φτιαγμένα για συγκεκριμένες λειτουργίες οι οποίες δεν μπορούν να αλλάξουν και είναι γενικά για όλα τα μαθήματα. Αντιθέτως, η συγκεκριμένη πλατφόρμα είναι εξειδικευμένη και ανταποκρίνεται πλήρως στις ανάγκες του μαθήματος και παρόλο που είναι υλοποιημένες κύριες λειτουργίες του μαθήματος σε μελλοντικά στάδια μπορεί με προσθήκη να περισσότερων λειτουργιών να ανταποκριθεί πλήρως σε όλα τις ανάγκες που χρειάζεται ένας καθηγητής.

1.2 Σκοπός Διπλωματικής Εργασίας

Ο κύριος σκοπός αυτής της διπλωματικής εργασίας ήταν αρχικά η εύκολη διαχείριση του πανεπιστημιακού μαθήματος ΕΠΛ343. Στο συγκεκριμένο μάθημα υπάρχει η πραγματική

ανάγκη υλοποίησης εφαρμογής είτε από κάποιο οργανισμό είτε από άτομα μεμονωμένα σε αντίθεση με άλλα όπου οι εργασίες βασίζονται σε υποθετικές περιπτώσεις υλοποίησης ή σκέψης κυρίως του καθηγητή. Κατά συνέπεια, το γεγονός αυτό δημιουργεί την ανάγκη της επικοινωνίας μαζί με το άτομο το οποίο είναι υπεύθυνο για τον οργανισμό κυρίως για τον καθορισμό απαιτήσεων αλλά και για την τελική αξιολόγηση του Ε.Α.. Η δημιουργία της συγκεκριμένης πλατφόρμας θα διευκολύνει την επικοινωνία αυτή και θα υπάρχει πρόσβαση στο σύστημα τόσο του φοιτητή και του καθηγητή (όπως συμβαίνει στα περισσότερα συστήματα τα οποία είναι παρόμοια) αλλά και του ατόμου που είναι υπεύθυνο για την επικοινωνία με τους φοιτητές για το έργο το επιθυμούν να αναπτυχθεί.

Προηγούμενες χρονιές που δεν υπήρχε η ευκαιρία χρησιμοποίησης παρόμοιου συστήματος οι φοιτητές χρησιμοποιούσαν διάφορες πλατφόρμες που υπηρετούσαν διαφορετικό σκοπό και λόγο ύπαρξης έτσι ώστε να έχουν πρόσβαση στα απαραίτητα και να ενημερώνονται για πληροφορίες του μαθήματος όπως βαθμοί, εργασίες κτλ.. Μερικά από τα συστήματα τα οποία χρησιμοποιούνταν είναι e-learning συστήματα τα οποία όμως δεν περιέχουν όλες τις απαραίτητες λειτουργίες που χρειάζεται το μάθημα, καθώς δεν είναι τόσο εξειδικευμένα, έτσι δημιουργείται η ανάγκη δημιουργίας της πλατφόρμας για να αποφευχθεί η χρήση πολλαπλών συστημάτων και η αποφυγή οποιασδήποτε σύγχυσης ίσως υπάρχει.

Υπάρχουν διάφορα σενάρια τα οποία η δημιουργία μίας πλατφόρμας εξειδικευμένης για το μάθημα αυτό θα αποτελούσε πολύ χρήσιμο εργαλείο και θα διευκόλυνε φοιτητές, καθηγητές αλλά και τα άτομα που ήταν υπεύθυνα για τους οργανισμούς. Για παράδειγμα για την ανάρτηση βαθμών ο καθηγητής χρησιμοποιούσε τόσο το Moodle όσο και το ηλεκτρονικό ταχυδρομείο για ενημέρωση των φοιτητών και αποστολή προσωπικών μηνυμάτων σε κάθε ομάδα μαζί με το σχετικό αρχείο ανατροφοδότησης για το παραδοτέο. Άρα οι βαθμοί υπήρχαν αναρτημένοι σε 2 διαφορετικές εφαρμογές και επίσης το πιο δύσκολο ήταν η ρουτίνα που εκτελούσε ο καθηγητής σε κάθε παραδοτέο, δηλαδή να ελέγχει τα άτομα τα οποία αποτελούν την συγκεκριμένη ομάδα για να τους αποστείλει μήνυμα τόσο για το παραδοτέο αλλά και η ανάρτηση του βαθμού τους. Με την υλοποίηση της πλατφόρμας η ανάγκη αυτή εξαφανίζεται καθώς ο καθηγητής μπορεί πολύ απλά μέσα από την λειτουργία ανάρτησης βαθμού να επιλέξει το αναγνωριστικό της ομάδας και να μεταφορτώσει το σχετικό αρχείο ανατροφοδότησης μαζί με τον βαθμό και την στιγμή που γίνει η ανάρτηση η πλατφόρμα θα είναι υπεύθυνη για αποστολή μηνύματος στο ηλεκτρονικό ταχυδρομείο των φοιτητών για ενημέρωση. Επιπλέον, σε αντίθεση με το παρελθόν οι φοιτητές θα μπορούν να βλέπουν συμμαζεμένες όλες τις υποχρεώσεις τους έτσι ώστε να μειώνεται η πιθανότητα να ξεχάσουν κάποια, όπως για παράδειγμα τα ερωτηματολόγια τα οποία στο παρελθόν ο καθηγητής υπενθυμίζει τους

φοιτητές μέσω μηνύματος στο ηλεκτρονικό ταχυδρομείο που περιείχε το συγκεκριμένο υπερσύνδεσμο.

Συμπερασματικά, από τα πιο πάνω μπορεί εύκολα να διαφανεί ο σκοπός της συγκεκριμένης διπλωματικής όπως και η ανάγκη να υπάρξει ένα τέτοιο σύστημα. Αποτελεί μία πρωτότυπη ιδέα που πιθανόν στο μέλλον να ακολουθήσουν και άλλα μαθήματα τόσο από το Πανεπιστήμιο Κύπρου αλλά και γενικά από άλλα. Στην εποχή την οποία βρισκόμαστε η αυτοματοποίηση των πάντων είναι γεγονός σε διάφορους τομείς την ζωής μας και η δημιουργία μίας τέτοιας πλατφόρμας υποστηρίζει τους γρήγορους ρυθμούς ζωής μας.

1.3 Δομή Διπλωματικής Εργασίας

Η διπλωματική εργασία περιλαμβάνει 6 κεφάλαια τα οποία περιγράφονται πιο κάτω:

Το πρώτο κεφάλαιο με το οποίο ξεκινάει το κείμενο είναι εισαγωγική ενότητα η οποία γίνεται μία βασική περιγραφή του συστήματος. Πιο συγκεκριμένα αναλύεται γενική περιγραφή και ο σκοπός ο οποίος δημιουργήθηκε ένα τέτοιο σύστημα μαζί με την κεντρική ιδέα ανάπτυξης. Επίσης, αναλύονται και κάποια παρόμοια συστήματα για να διαφανεί κατά πόσο το σύστημα προσφέρει κάτι διαφορετικό από τα ήδη υπάρχοντα. Τέλος, υπάρχει ένας πίνακας μαζί με ορισμούς, ακρώνυμα και συντομογραφίες οι οποίες υπάρχουν στο κείμενο και θα βοηθήσουν την κατανόηση του.

Το δεύτερο κεφάλαιο εμβαθύνει περισσότερο στο τρόπο με τον οποίο υλοποιήθηκε η εφαρμογή δηλαδή το θεωρητικό και τεχνικό υπόβαθρο. Αρχικά αναλύονται οι περιορισμοί οι οποίοι θα περιόριζαν τον τρόπο υλοποίησης της πλατφόρμας μας σε θέματα κυρίως ασφάλειας, απόδοσης και επεκτασιμότητας. Στην συνέχεια περιγράφεται η αρχιτεκτονική του συστήματος και όλα τα εργαλεία και τεχνολογίες οι οποίες χρησιμοποιήθηκαν για την υλοποίηση του συστήματος. Τέλος, καταγράφεται και η άδειες λογισμικού οι οποίες η εφαρμογή μας περιέχει και κατά πόσο μπορούν να συνυπάρξουν.

Στο τρίτο κεφάλαιο γίνεται κυρίως ανάλυση απαιτήσεων και προδιαγραφών. Αρχικά με την χρήση του κύκλου ανάπτυξης LUCID περιγράφονται τα στάδια ανάπτυξης λογισμικού. Στην συνέχεια προσδιορίζονται οι τυπικοί χρήστες, το περιβάλλον εργασίας τους και παραδοχές που ίσως υπάρχουν για την δημιουργία του συστήματος. Το κεφάλαιο αυτό πραγματεύεται τον τρόπο που έγινε η εξαγωγή των απαιτήσεων και των προδιαγραφών με βάση τα χαρακτηριστικά των χρηστών του συστήματος, δηλαδή το ερωτηματολόγιο, τις personas και τα σενάρια χρήσης. Επιπρόσθετα, γίνεται εξαγωγή των λειτουργικών και μην λειτουργικών απαιτήσεων που θα υποστηρίζει η πλατφόρμα και παρουσιάζεται η ροή υποσελίδας για κάθε ένα διαφορετικό τύπο χρήστη της ιστοσελίδας.

Με την ανάλυση η οποία έγινε στο προηγούμενο κεφάλαιο δημιουργήθηκαν όλες οι λειτουργίες τις οποίες θα υποστηρίζει η πλατφόρμα και περιγράφονται πλήρως στο κεφάλαιο τέσσερα. Υπάρχει ένα μικρός οδηγός για κάθε τύπο χρήστη ο οποίο περιγράφει τις λειτουργίες τις οποίες έχει δυνατότητα να εκτελέσει μαζί με συμπληρωματικά στιγμιότυπα και κάποιες πληροφορίες οι οποίες είναι σημαντικό να αναφερθούν ειδικά για τους νέους χρήστες.

Στο πέμπτο κεφάλαιο παρουσιάζεται ο έλεγχος του τελικού συστήματος. Αναλύεται το ερωτηματολόγιο από τους χρήστες αλλά και σχόλια τα οποία προέκυψαν από την αξιολόγηση του συστήματος από πιθανούς χρήστες. Σε αυτό το σημείο ελέγχεται κατά πόσο έχουν τηρηθεί οι οδηγίες σχεδιασμού και οι στόχοι ευχρηστίας καθώς και η υποκειμενική ικανοποίηση του χρήστη.

Το έκτο και τελευταίο κεφάλαιο παρουσιάζει τα συμπεράσματα και τον επίλογο αυτής της διπλωματικής εργασίας. Συγκεκριμένα παρουσιάζονται ιδέες σχετικά με την επεκτασιμότητα του συστήματος που λόγω χρόνου δεν υλοποιήθηκαν αλλά σε κάποιο μελλοντικό στάδιο θα ήταν πολύ ωραίες ιδέες για επέκταση. Επίσης, γίνεται η τελική ανασκόπηση της εφαρμογής και της όλης εμπειρίας η οποία αποκτήθηκε κατά την εκπόνηση της διπλωματικής εργασίας.

1.4 Έρευνα για παρόμοια συστήματα

Διάφορα πανεπιστήμια αλλά και καθηγητές χρησιμοποιούν πληθώρα επιλογών όσο αφορά εφαρμογές οι οποίες θα αποτελέσουν τον τρόπο επικοινωνίας τους μαζί με φοιτητές. Η ανάγκη αυτή υπάρχει κυρίως για την ανάρτηση βαθμών, εκπαιδευτικού υλικού αλλά και επικοινωνία μέσω φόρουμ. Αυτού του είδους συστήματα περιέχουν παρόμοιες λειτουργίες με την πλατφόρμα την οποία υλοποιήσαμε στα πλαίσια της διπλωματικής εργασίας αλλά κανένα από αυτά δεν μπορεί να υποστηρίξει όλες τις ανάγκες διαχείρισης που υπάρχουν στο συγκεκριμένο μάθημα. Πιο κάτω υπάρχουν μερικά παραδείγματα εκπαιδευτικού περιεχομένου τα οποία περιγράφουν της εφαρμογές αυτές μαζί με τις δυνατότητες τους:

- Moodle:



Εικόνα 1.4. 1: Λογότυπο Moodle

Αποτελεί ένα σύστημα διαχείρισης μαθημάτων (Course Management System) το οποίο ανήκει στην κατηγορία Learning Management System (LMS) και παρέχεται ως ελεύθερο λογισμικό ανοικτού κώδικα GNU Public License. Δημιουργήθηκε από τον Martin Dougiamas στην εκπόνηση του PhD του το 1999 και σύμφωνα με αυτόν, έχει δημιουργηθεί πάνω στη φιλοσοφία του κοινωνικού δομητισμού. Οποιοδήποτε σύστημα υποστηρίζει PHP είναι σε θέση να το τρέξει ενώ έχει την δυνατότητα να συνδυάζει πολλούς και διαφορετικούς τύπους Β.Δ. με ιδιαίτερη προτίμηση στην MySQL.

Γενικά διαθέτει μία πολύ μεγάλη πικουιλία από εργαλεία και λειτουργίες για την υποστήριξη του μαθήματος γεγονός το οποίο δεν είναι τυχαίο αφού πάνω από 90 εκατομμύρια χρήστες σε όλο τον κόσμο το χρησιμοποιούν τόσο σε ακαδημαϊκό επίπεδο όσο και σε επιχειρησιακό. Μερικές από τις πιο σημαντικές λειτουργίες του είναι ^[25]:

- διαχείριση πόρων (έγγραφα, μαθήματα, γλωσσάριο)
- υποστήριξη της επικοινωνίας [ομάδες συζητήσεων (forums), συζήτηση (chat), ιστολόγιο (blog)]
- προαγωγή ομαδικής εργασίας (wiki, βάση δεδομένων, ομάδες συζητήσεων, γλωσσάριο)
- υποστήριξη αξιολόγησης (κουίζ, εργασίες, βαθμολόγιο)
- έλεγχος διαχείρισης (ομάδες, ημερολόγιο, αναφορές χρήσης, βαθμολόγιο, ερωτηματολόγιο)

Σημαντικό είναι να αναφερθεί ότι έχει δεχθεί μεγάλη παγκόσμια αποδοχή αφού μπορεί να υποστηρίξει περισσότερες από 75 διαφορετικές γλώσσες.

- Blackboard:

Το Blackboard Learn αποτελεί ένα εικονικό μαθησιακό περιβάλλον και σύστημα μάθησης που αναπτύχθηκε από τη Blackboard Inc. Είναι ένα λογισμικό διακομιστή που βασίζεται στον Ιστό με διαχείριση μαθημάτων, προσαρμόσιμη ανοιχτή αρχιτεκτονική και επεκτάσιμο σχεδιασμό ο οποίος μπορεί να ενσωματωθεί σε συστήματα πληροφοριών μαθητών και πρωτόκολλα επαλήθευσης ταυτότητας (login system verify). Μπορεί να εγκατασταθεί σε έναν τοπικό διακομιστή που φιλοξενείται από το Blackboard ASP Solutions ή να φιλοξενείται στο Amazon Web Services ως λογισμικό ως υπηρεσία. Ο κύριος στόχος του είναι να προσθέσει διαδικτυακά στοιχεία σε μαθήματα που προσφέρονται παραδοσιακά πρόσωπο με πρόσωπο και να αναπτύξει εντελώς διαδικτυακά μαθήματα με λίγες ή καθόλου συναντήσεις πρόσωπο με πρόσωπο [27].

Όσο αφορά επικοινωνία φοιτητών καθηγητών παρέχονται οι πιο κάτω λειτουργίες:

- Ανακοίνωση: Οι εκπαιδευτικοί μπορούν να αναρτήσουν ανακοινώσεις για να ενημερώσουν τους μαθητές.
- Συνομιλία (Online Chat): Αυτή η λειτουργία επιτρέπει στους συνδεδεμένους μαθητές να συνομιλούν με άλλους μαθητές στην τάξη τους σε πραγματικό χρόνο.

- Συζήτηση/Φόρουμ: Αυτή η δυνατότητα επιτρέπει στους μαθητές και τους δασκάλους να δημιουργήσουν νήματα συζήτησης και να ανταποκριθούν στο δημιουργημένο περιεχόμενο.
- Mail: Το Blackboard Mail επιτρέπει στους μαθητές και τους δασκάλους να στέλνουν αλληλογραφία μεταξύ τους.
- Αποστολή μηνύματος (μόνο με την επεκτάσιμη λειτουργία Uper): Οι μαθητές και οι δάσκαλοι μπορούν να επικοινωνούν μεταξύ τους στο μάθημα, αν και όχι σε πραγματικό χρόνο.

Όσο αφορά την ενημέρωση περιεχομένου του μαθήματος παρέχονται οι πιο κάτω λειτουργίες:

- Προσθήκη περιεχομένου: Αυτή η δυνατότητα επιτρέπει στους καθηγητές να δημοσιεύουν άρθρα, εργασίες, βίντεο κ.λπ. Το περιεχόμενο μπορεί να περιοριστεί ώστε να είναι διαθέσιμο μόνο υπό ορισμένες προϋποθέσεις, όπως μετά την υποβολή μιας εργασίας ή σε μια συγκεκριμένη στιγμή.
- Ημερολόγιο: Οι καθηγητές μπορούν να χρησιμοποιήσουν αυτήν τη λειτουργία για να δημοσιεύσουν σημαντικές ημερομηνίες (π.χ. λήξη για εργασίες και εξετάσεις).
- Αξιολόγηση: Αυτή η καρτέλα επιτρέπει στους καθηγητές να δημοσιεύουν κουίζ και εξετάσεις και επιτρέπει στους μαθητές να έχουν πρόσβαση σε αυτά διαδικτυακά.
- Μεταφόρτωση εργασιών: Αυτές οι δυνατότητες επιτρέπουν στην εργασία και στους μαθητές να υποβάλλουν εργασίες στο διαδίκτυο.
- Ανάρτηση βαθμολογίας: Οι καθηγητές μπορούν να δημοσιεύσουν βαθμούς στον πίνακα κάθε μαθητή έτσι ώστε να μπορούν εύκολα να δουν τον βαθμό τους για τις εργασίες τις οποίες ανάρτησαν.



Εικόνα 1.4. 2: Λογότυπο Blackboard

- Piazza:



Εικόνα 1.4. 3: Λογότυπο Piazza

Το Piazza είναι ένα σύστημα διαχείρισης μάθησης που επιτρέπει στους μαθητές να κάνουν ερωτήσεις με τη μορφή φόρουμ. Ο καθηγητής μπορεί να συντονίσει τη συζήτηση και να εγκρίνει τη σωστή απάντηση. Το λογισμικό εφευρέθηκε από την Pooja Nath το 2009. Χρησιμοποιείται κυρίως να μειώσει τον χρόνο απόκρισης ερωτήσεων μέσω μηνύματος ηλεκτρονικού ταχυδρομείου και να προσφέρει ένα κοινό μέρος για τους μαθητές να συζητήσουν έξω από την τάξη για θέματα που αφορούν το μάθημα. Έχει τη δυνατότητα να επιτρέψετε στους μαθητές να δημοσιεύουν ανώνυμα, ενθαρρύνοντας έτσι πιο εμπειριστατωμένες συζητήσεις. [26]

Κάθε ερώτηση θα δημιουργήσει μια συλλογική απάντηση, οποιοσδήποτε χρήστης (καθηγητής ή μαθητής) μπορεί να συνεισφέρει σε αυτό (απαντώντας ή θέτοντας κάποιο άλλο σχετικό ερώτημα). Επιτρέπει στους χρήστες να επισυνάψουν εξωτερικά αρχεία στην ανάρτηση, να χρησιμοποιήσουν τη μορφή LaTeX, να δουν το ιστορικό δημοσιεύσεων της ανάρτησης, και προσθέστε ερωτήσεις παρακολούθησης και λάβετε ειδοποιήσεις μέσω email όταν προστίθεται νέο περιεχόμενο. Η διεπαφή περιλαμβάνει μια λίστα δυναμικών αναρτήσεων στην αριστερή πλευρά της οθόνης, ένα κεντρικό πλαίσιο για προβολή και συνεισφορά μεμονωμένων αναρτήσεων, καθώς και μια κορυφαία γραμμή για έλεγχο λογαριασμού.

Τα πιο πάνω παραδείγματα αφορούσαν τον εκπαιδευτικό τομέα όπως δηλαδή και η κατηγορία στην οποία ανήκει η πλατφόρμα μας αλλά δεν υπάρχουν πολλές ομοιότητες με τον τρόπο λειτουργίας των 2 εφαρμογών. Από την άλλη υπάρχουν τα συστήματα εύρεσης εργασίας (παραδείγματα Εικόνα 1.4.4) τα οποία παρόλο που δεν ανήκουν στις πλατφόρμες διαχείρισης μαθημάτων έχουν αρκετές παρόμοιες λειτουργίες. Η λογική η οποία χρησιμοποιούν είναι παρόμοια με την δική μας αφού κάθε χρήστης ανεβάζει το βιογραφικό του (CV) και μπορεί να δει κενές θέσεις εργασίας ανάλογα με τις δυνατότητες του αλλά και γενικά όλες τις δυνατές επιλογές. Παρόμοια λογική ακολουθεί και η δική μας σελίδα όπου οι φοιτητές συμπληρώνουν τις γνώσεις τους και μπορούν να δουν στην συνέχεια τα έργα ανάθεσης τα οποία ανταποκρίνονται στις δεξιότητες τους αλλά και γενικά όλα τα αναρτημένα. Επιπρόσθετα, οργανισμοί συνδέονται στις σελίδες εύρεσης εργασίας, αναρτούν κενές θέσεις εργασίας για τους χρήστες, όπως ακριβώς και οι οργανισμοί στην δική μας πλατφόρμα, και στην συνέχεια μπορούν να δουν τους ενδιαφερόμενους χρήστες για να προχωρήσουν τις διαδικασίες είτε στην μία πλατφόρμα είτε στην άλλη. Συμπερασματικά, παρόλο που σελίδες όπως το Moodle, Blackboard, Piazza οι οποίες ανήκουν στο τομέα εκπαίδευσης είναι πιο αφηρημένα και γενικά υλοποιημένες σε σχέση με την πλατφόρμα μας ενώ τα συστήματα εύρεσης εργασίας έχουν αρκετά παρόμοιες λειτουργίες.



Εικόνα 1.4. 4: Λογότυπα συστημάτων εύρεσης εργασίας

1.5 Ακρόνυμα και Συντομογραφίες

Στο πιο κάτω πίνακα φαίνονται διάφοροι ορισμοί και ακρωνύμια τα οποία εμφανίζονται στο κείμενο.

Ακρόνυμα, συντομογραφίες	Επεξήγηση
B.Δ.	Βάση Δεδομένων
E.A.	Έργο Ανάθεσης

O.A.E.	Ομάδα Ανάθεσης Έργου
Π.Κ	Πανεπιστήμιο Κύπρου
JWT	JSON Web Token
REST	Representational State Transfer
RWS	RESTful Web Service
GUI	Graphical User Interface
URL	Uniform Resource Locator
API	Application programming interface
CLI	Command Line Interface
DDL	Data Definition Language
DOM	Document Object Model
ERD	Enterprise Resource Planning
OTK	One Time Key
GDPR	General Data Protection Regulation
ISO	International Organization for Standardization
Χρήστης	Οποιοδήποτε άτομα συνδέεται στην πλατφόρμα
Ιστοσελίδα	Πλατφόρμα, Διαδικτυακή εφαρμογή
Καθηγητής	Χρήστης με λογαριασμό ο οποίος έχει δυνατότητες τύπου καθηγητή
Οργανισμός	Χρήστης με λογαριασμό ο οποίος έχει δυνατότητες τύπου οργανισμού
Φοιτητής	Χρήστης με λογαριασμό ο οποίος έχει δυνατότητες τύπου φοιτητή
Πιστοποιητικά σύνδεσης	Στοιχεία με τα οποία συνδέεται ο χρήστης στην πλατφόρμα όνομα χρήστη και κωδικός πρόσβασης
Διεπαφή	Το μέσο επικοινωνίας μιας οντότητας με το περιβάλλον της.

Domain	http://10.16.33.114:3000/
--------	---

Κεφάλαιο 2

Ανάλυση Απαιτήσεων και Προδιαγραφών

2.1.Εισαγωγή

- 2.1.1. Μοντέλο Κύκλου Ζωής
- 2.1.2. Προσδιορισμός Τυπικών Χρηστών
- 2.1.3. Περιβάλλον Εργασίας
- 2.1.4. Περιορισμοί και Παραδοχές συστήματος

2.2.Προσδιορισμός Αναγκών

- 2.2.1. Ανάλυση Ερωτηματολογίου
- 2.2.2. Γενικά Συμπεράσματα

2.3.Ανάλυση Αναγκών

- 2.3.1. Personas
- 2.3.2. Σενάρια Χρήσης

2.4.Καθορισμός Κανόνων Σχεδιασμού και Ευχρηστίας

2.5.Καθορισμός Απαιτήσεων

- 2.5.1. Εισαγωγή
- 2.5.2. Λειτουργικές Απαιτήσεις
- 2.5.3. Μη λειτουργικές Απαιτήσεις

2.6.Διάγραμμα Ροής Εμπειρίας Χρήστη

- 2.6.1. Διάγραμμα Ροής Υποσελίδα Καθηγητή
- 2.6.2. Διάγραμμα Ροής Υποσελίδα Οργανισμού
- 2.6.3. Διάγραμμα Ροής Υποσελίδα Φοιτητή

2.7.Σχεδίαση Βάσης Δεδομένων

- 2.7.1. Διάγραμμα Οντοτήτων Συσχετίσεων
- 2.7.2. Πίνακες Βάσης Δεδομένων
- 2.7.3. Χωρητικότητα Βάσης Δεδομένων

2.1.Εισαγωγή

2.1.1. Μοντέλο Κύκλου Ζωής

Τα περισσότερα πληροφορικά συστήματα ακολουθούν ένα μοντέλο κύκλου ζωής με συγκεκριμένη μεθοδολογία για την διαδικασία παραγωγής περιεχομένου και συγγραφής λογισμικού από την αρχή της ανάπτυξης (καθορισμός ιδέα) μέχρι την ολοκλήρωση (σύστημα

σε λειτουργία). Η υλοποίηση της πλατφόρμας μας αποτελεί σύστημα με αντικειμενοστραφείς και ανθρωποκεντρικής ανάπτυξης διαδραστικού συστήματος η οποία έχει τρεις βασικές αρχές οι οποίες διέπουν τις βασικές δραστηριότητες της διαδραστικής σχεδίασης. Συγκεκριμένα εστιάζουμε στους χρήστες του συστήματος και τις εργασίες που επιτελούν με αυτό, από τις αρχικές φάσεις σχεδιασμού, σε όλες τις φάσεις σχεδιασμού μετράμε την αντίδραση των χρηστών με χρήση πρότυπων διεπιφανειών, εγχειριδίων, προσομοιωτών κλπ. και ακολουθούμε επαναληπτική διαδικασία σχεδίασης. Η σχεδίαση η οποία ακολουθούμε είναι η Λογική Χρηστοκεντρική Διαδραστική Σχεδίαση (Logical User – Centered Interactive Design - LUCID) η οποία διακρίνεται από έξι φάσεις ανάπτυξης και έχει πολλά κοινά στοιχεία με το εξελικτικό μοντέλο. Επικεντρώνεται στο βέλτιστο σχεδιασμό όλων των λειτουργιών κυρίως από την μεριά του χρήστη αποκτώντας την μέγιστη δυνατή απόδοση, αποδοτικότητα, εξυπηρέτηση, ανοχή σφαλμάτων και ευκολία εκμάθησης (οι 5 αυτές αρχές είναι γνωστές και ως ‘Five-E’). Στα πρώτα βήματα της διαδικασίας αυτής γίνεται η ανάπτυξη της αρχικής ιδέας του συστήματος και η ανάλυση αναγκών και απαιτήσεων. Στην συνέχεια γίνεται ο σχεδιασμός του προϊόντος (πλατφόρμα - διαδικτυακή εφαρμογή) με πρότυπη βασική οθόνη και γίνεται επαναληπτικός σχεδιασμός και βελτίωση πρωτοτύπου μέχρι να καταλήξουμε στο τελικό σχέδιο. Στις τελευταίες δύο φάσεις γίνεται η ανάπτυξη του συστήματος και η υποστήριξη αρχικής λειτουργίας [18]. Μελετώντας την Εικόνα 3.1.1.1 παρατηρούμε τις φάσεις αυτές ξεκάθαρα και φαίνεται ο εξελικτικός χαρακτήρας της διαδικασίας αφού σε κάθε καινούρια φάση καινούριες λειτουργίες δημιουργούνται και υπάρχει διαδοχική βελτίωση στο σύστημα. Πιο κάτω περιγράφονται πιο αναλυτικά οι φάσεις σχεδίασης:

– Φάση 1: Ανάπτυξη αρχικής ιδέας συστήματος:

Σκοπός της πρώτης φάσης είναι να δημιουργηθεί πρώτα η ιδέα του προϊόντος (διαδικτυακή εφαρμογή) και στη συνέχεια να καθορίσει τους επιχειρηματικούς του στόχους. Μόλις ολοκληρωθούν, οι ομάδες θα πρέπει να συγκροτηθούν για να συμβάλουν στην ανάπτυξη, το πρόγραμμα, τον προϋπολογισμό του έργου και το σχέδιο διαθεσιμότητας. Με αυτόν τον τρόπο, προσδιορίζονται τυπικοί χρήστες, καθώς και τεχνικές και περιβαλλοντικές παράμετροι. Τέλος γίνεται ορισμός πλάνου ανάπτυξης, προσωπικού που πρόκειται να εμπλακεί στην ανάπτυξη στο μέλλον.

– Φάσης 2: Ανάλυση αναγκών και απαιτήσεων:

Σκοπός της δεύτερης φάσης είναι να προσδιορίσει διαφορετικές ομάδες χρηστών (τρία είδη τυπικών χρηστών – καθηγητής, οργανισμός, φοιτητής) με βάση τα χαρακτηριστικά τους και να αναλύσει τις εργασίες σε μεμονωμένες και ξεχωριστές οντότητες. Γίνεται επίσης ανάλυση αναγκών μέσω δημιουργίας σεναρίων με συμμετοχή των χρηστών γεγονός που συμβάλει στην

ανθρωποκεντρική σχεδίαση. Αυτό θα διευκολύνει την ανάλυση των απαιτήσεων, επειδή η αφοσίωση των χρηστών και οι περιγραφές θέσεων εργασίας θα δημιουργηθούν μέσω της βασικής ροής εργασίας (data flow) . Επιπλέον, εντοπίζονται τα κύρια αντικείμενα και δομές (περιγράφονται στο Κεφάλαιο 2) οι οποίες θα χρησιμοποιηθούν στην διεπιφάνεια χρήστη και επιλύονται τεχνικά προβλήματα και περιορισμοί που ίσως υπήρχαν τεχνικού χαρακτήρα.

– Φάση 3: Σχεδιασμός προϊόντος με πρότυπη βασική οθόνη:

Σκοπός της τρίτης φάσης είναι να καθορίσει συγκεκριμένους στόχους χρηστικότητας και ευχρηστίας με βάση τις ανάγκες των χρηστών (κυρίως οι οποίες περιγράφονται στην φάση 2) και να δημιουργήσει οδηγίες σχεδίασης και διαδραστικούς οδηγούς στυλ. Η ευχρηστία είναι μία από τις βασικές παραμέτρους της ποιότητας ενός διαδραστικού συστήματος. Σκοπός επίσης είναι να υπάρχει ευκολία στην εκμάθηση από τους χρήστες του συστήματος, υψηλή απόδοση, χαμηλή συχνότητα σφαλμάτων χρήστη, εύκολη ανάκτηση από σφάλματα, εύκολο να διατηρήσει τις γνώσεις του σχετικά με τη χρήση και τελικά να κάνει οι χρήστες αισθάνονται υποκειμενικά ικανοποιημένοι, τεχνικές οι οποίες σχετίζονται με την αλληλεπίδραση ανθρώπου υπολογιστή και θα είναι βασικό στοιχείο της αξιολόγησης μας. Επιπλέον, σε αυτό το στάδιο, καθορίζεται το μοντέλο ομάδας και πλοήγησης της κύριας οθόνης, ώστε να μπορούν να χρησιμοποιηθούν εργαλεία ταχείας δημιουργίας πρωτοτύπων για τη δημιουργία ενός πρωτοτύπου της κύριας οθόνης και στη συνέχεια να ζητείται η γνώμη του χρήστη για το πρωτότυπο αυτό. Το πρωτότυπο θα περιλαμβάνει και τον ορισμό βασικών οθονών όπως για παράδειγμα η εισαγωγική οθόνη, η κεντρική οθόνη (home screen) και βασικές οθόνες κύριων διεργασιών για την πλατφόρμα. Θα ληφθούν μετρήσεις ευχρηστίας για το πρωτότυπο το οποίο κατασκευάστηκε για να υπάρχει η δυνατότητα σύγκρισης στα επόμενα στάδια.

– Φάση 4: Επαναληπτικός σχεδιασμός και βελτίωση πρωτοτύπου:

Στην τέταρτη φάση υπάρχει μία επαναληπτική διαδικασία όπου επεκτείνεται και βελτιώνεται συνεχώς η διαδικτυακή εφαρμογή πλατφόρμας. Γίνεται επέκταση, υλοποίηση και συνένωση των βασικών οθονών σε πλήρες σύστημα με συνεχής εμπειρική αξιολόγηση των διεπιφανειών από ειδικούς διαδραστικών συστημάτων. Επιπρόσθετα από την συνεχή αξιολόγηση θα γίνεται και διενέργεια μετρήσεων ευχρηστίας ευρείας κλίμακας για όλο το σύστημα αλλά και σε κάθε τέλος επαναληπτικής σχεδίασης θα γίνεται παραγωγή ενός πλήρους πρωτοτύπου με λεπτομερείς περιγραφή.

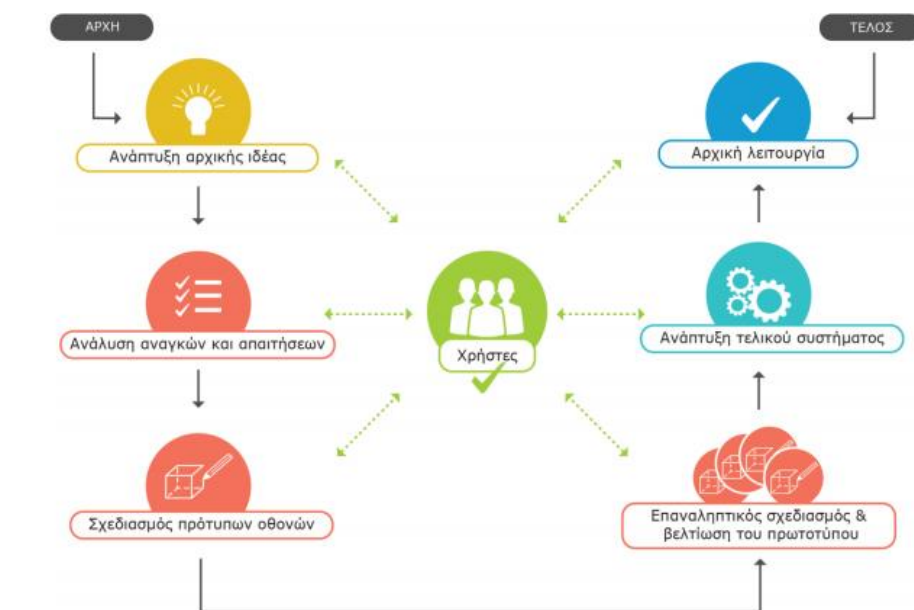
– Φάση 5: Ανάπτυξη συστήματος :

Με το τέλος της τέταρτης φάσης καταλήξαμε στις προδιαγραφές και το πρωτότυπο της διαδικτυακής πλατφόρμας όπου στην πέμπτη φάση θα αποτελέσουν τις κατευθυντήριες γραμμές για την υλοποίηση του συστήματος. Με την ανάπτυξη του συστήματος μπορεί να

προκύψουν νέα εργαλεία και πρακτικές συμπληρωματικά από αυτά που αποφασίσαμε αρχικά αλλά και τεχνικά προβλήματα μαζί με προκλήσεις τα οποία δεν υπολογίσαμε. Αναμφισβήτητα θα χρειαστεί αντιμετώπιση των τροποποιήσεων αυτών που ενδεχομένως να προκύψουν. Επιπλέον αναπτύσσονται και ηλεκτρονικά βοηθήματα (on-line help), εγχειρίδια χρήσης και εκπαιδευτικού υλικού για να είναι πιο εύκολη η εκμάθηση του συστήματος και η χρήση του από τους χρήστες.

– Φάση 6: Υποστήριξη αρχικής λειτουργίας :

Τελευταίο βήμα αποτελεί η υποστήριξη και εκπαίδευση των χρηστών με οποιοδήποτε τρόπο και οποιαδήποτε στιγμή το χρειαστούν. Σημαντικό στην διαδικασία να υπάρχει καταγραφή και αξιολόγηση των συμβάντων σε περίπτωση ανανέωσης και βελτίωσης του συστήματος αλλά και να τα διαχωρίσουν ανάλογα με την σημαντικότητα και την ανάγκη διόρθωσης τους. Αυτό προϋποθέτει να υπάρχει συντήρηση του συστήματος οποιαδήποτε στιγμή χρειαστεί είτε από θέμα λειτουργιών, αλλαγών, ασφάλειας πληροφοριών τόσο των χρηστών όσο και τις ιστοσελίδας.



Εικόνα 3.1.1. 1: Το μοντέλο ανθρωποκεντρικού σχεδιασμού LUCID

2.1.2. Προσδιορισμός Τυπικών Χρηστών

Στην διαδικτυακή πλατφόρμα υπάρχουν τρεις κύριες οντότητες τυπικών χρηστών όσο αφορά την χρήση και τις δυνατότητες τους στο υπάρχον σύστημα. Συμπληρωματικά όμως μπορούν να χωριστούν και ανάλογα με την εμπειρία χρήσης τους σε Η/Υ και με βάση τη χρήση τους με την συγκεκριμένη εφαρμογή όπως φαίνεται και πιο κάτω:

Πιο αναλυτικά:

- Με βάση την εμπειρία χρήσης

- Χρήστες οι οποίοι χρησιμοποίησαν παρόμοια συστήματα (περιγράφονται στην ενότητα 1.4) και αλληλοεπιδρούν με το διαδίκτυο συχνά. Οι χρήστες αυτοί έχουν ήδη αποκτήσει όλες τις απαραίτητες γνώσεις για το διαδίκτυο, συνεπώς και για την πλατφόρμα μας, αφού υιοθετούμε πρακτικές παρόμοιες με διάφορες γνωστές ιστοσελίδες και πλατφόρμες. Συμπληρωματικά μαζί με οποιαδήποτε προηγούμενη εμπειρία έχουν αποκτήσει σε παρόμοιες διαδικτυακές εφαρμογές πιστεύουμε ότι δεν θα αντιμετωπίσουν δυσκολίες χρήσης του συστήματός μας.
- Χρήστες που δεν αλληλοεπιδρούν με το διαδίκτυο συχνά:
Οι χρήστες αυτοί δεν έχουν ιδιαίτερη εμπειρία χρήσης διαδικτύου. Μπορεί να αντιμετωπίσουν μερικές δυσκολίες τις πρώτες φορές που θα έρθουν σε επαφή με το σύστημα καθώς και με την χρήση λειτουργιών. Λόγω του ότι όμως δημιουργήσαμε το σύστημα μας όσο πιο εύχρηστο και απλό γίνεται η δυσκολία δεν θα είναι μεγάλη και ίσως να αφορά εξειδικευμένες περιπτώσεις αλλά γενικά ίσως να εκτελούν λειτουργίες επιτυχημένα απλά με πιο αργό τρόπο από τους έμπειρους χρήστες.
- Με βάση συχνότητα χρήσης του συστήματος
 - Συχνοί χρήστες συστήματος:
Υπάρχουν χρήστες οι οποίοι θα χρησιμοποιούν το σύστημα μας σε καθημερινή βάση, για παράδειγμα φοιτητές και καθηγητές. Αυτή η ομάδα θα μπορεί πολύ εύκολα να μάθει το σύστημα, ακόμα και αν δεν είχε κάποια προηγούμενη γνώση με διαδικτυακά συστήματα και αν εξοικειωθεί πλήρως με τις λειτουργίες. Με τη πάροδο του χρόνου και μέσα από την καθημερινή χρήση θα ολοκληρώνουν τις διεργασίες/λειτουργίες τους σε πολύ πιο μικρό χρονικό διάστημα σε σχέση με τις πρώτες μέρες αλληλεπίδρασης με το σύστημα.
 - Ευκαιριακοί χρήστες συστήματος:
Υπάρχουν χρήστες οι οποίοι δεν θα χρησιμοποιούν το σύστημα συχνά, για παράδειγμα χρήστες με λογαριασμό τύπου οργανισμού, αυτοί θα αλληλοεπιδρούν μαζί με το συγκεκριμένο σύστημα ανά κάποιο χρονικό διάστημα και ίσως θυμούνται κάποια στοιχεία από προηγούμενη τους χρήση για να βελτιώσουν την εμπειρία χρήσης τους. Παράλληλα όμως θα υπάρχουν και περιοχές άγνωστες γι' αυτούς μπορούν όμως εύκολα να μάθουν το σύστημα αφού είναι πολύ εύχρηστο και κατανοητό από τους χρήστες.

- Οντότητες τυπικών χρηστών με βάση την χρήση της ιστοσελίδας:

- ο Οργανισμός:

Χρήστης με λογαριασμό τύπου οργανισμού αρχικά θα δημιουργεί από μόνος του λογαριασμό με τα στοιχεία του αφού έχει αποδεχτεί του όρους και τις προϋποθέσεις της ιστοσελίδας. Κανόνες, συγκατάθεση, πληροφορίες και δικαιώματα συμπεριλαμβάνονται στη σελίδα που πρέπει να διαβάσει ο χρήστης πριν την εγγραφή του. Στην συνέχεια θα μπορεί να συνδεθεί και να αποκτήσει πρόσβαση σε όλες τις λειτουργίες που του αντιστοιχούν αλλά και να επεξεργαστεί τα στοιχεία προφίλ του αν επιθυμεί. Συγκεκριμένα η κύρια δυνατότητα του είναι να δημιουργεί Ε.Α. που επιθυμεί να υλοποιηθούν είτε πλήρως είτε σε στάδιο σχεδίασης δοκιμαστικά από φοιτητές του μαθήματος. Επιπλέον, έχει την δυνατότητα να βλέπει όλες τις προηγούμενες αναρτήσεις του, να τις επεξεργάζεται και να ορίζει Ο.Α.Ε για την υλοποίηση. Μπορεί επίσης να αναρτά αρχεία μέσω του συστήματος σε περίπτωση που χρειάζονται και να κάνει έξοδο από το σύστημα.

- ο Φοιτητής:

Οι λογαριασμοί φοιτητών δημιουργούνται με την εισαγωγή αρχείου csv από κάποιο χρήστη με τύπο λογαριασμό καθηγητή το οποίο περιέχει πληροφορίες σχετικά με αυτόν φοιτητή οι οποίες εξάγονται από την βοηθητική ιστοσελίδα του Banner¹. Στην συνέχεια λαμβάνουν ένα ενημερωτικό μήνυμα στο ηλεκτρονικό τους ταχυδρομείο και μπορούν πλέον να συνδεθούν στην ιστοσελίδα με τα στοιχεία εισόδου τα οποία έλαβαν. Μία από τις αρχικές τους δυνατότητες είναι να δημιουργήσουν ομάδες στα πλαίσια του μαθήματος για να έχουν την δυνατότητα να δείξουν το ενδιαφέρον τους σε συγκεκριμένα Ε.Α. Επιπρόσθετα, έχουν την δυνατότητα να δουν τις τελικές ημερομηνίες για τα παραδοτέα τους, να επισκεφτούν σελίδες με ερωτηματολόγια του μαθήματος και να επεξεργαστούν τα στοιχεία τους τόσο από θέμα πληροφοριών στην Β.Δ. όσο και τις ρυθμίσεις του λογαριασμού τους. Τέλος μπορεί να δει κάποια στατιστικά και στοιχεία για το μάθημα μέσω της κεντρικής σελίδας, να περιηγηθεί με την βοήθεια του menu και τέλος να αποσυνδεθεί.

1

<https://baneis.ucy.ac.cy:8443/authenticationendpoint/login.do?Name=PreLoginRequestProcessor&TARGET=https%3A%2F%2Fbanners.ucy.ac.cy%3A8082%2FStudentSelfService%2Flogin%2Fcas&commonAuthCallerPath=%252Fcas%252Flogin&forceAuth=true&passiveAuth=false&tenantDomain=carbon.super&sessionDataKey=b390994c-72c2-4c3a-a9aa-dcf32c0905ca&relyingParty=StudentSelfService&type=cas&sp=StudentSelfService&isSaaSApp=false&authenticators=BasicAuthenticator:LOCAL>

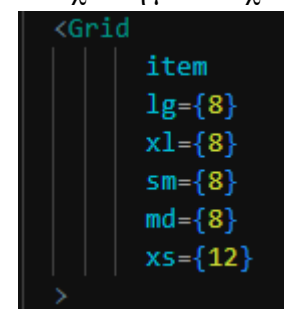
ο Καθηγητής:

Κάθε χρήστης με λογαριασμό τύπου καθηγητή έχει σχεδόν πλήρη πρόσβαση σε όλες τις λειτουργίες της πλατφόρμας καθώς εκείνος χειρίζεται και βαθμολογεί το μάθημα. Αρχικά συνδέεται με τα στοιχεία εισόδου του όπως όλοι οι χρήστες και στην συνέχεια έχει την δυνατότητα να δημιουργήσει καινούρια παραδοτέα, ερωτηματολόγια και Ε.Α. Σε κάθε μία από τις δημιουργίες που μόλις ανάφερα μπορεί επίσης να αναρτήσει και αρχεία μέσω του συστήματος έτσι ώστε να υπάρχουν συμπληρωματικές πληροφορίες, να τα επεξεργαστεί και να τα διαγράψει οποιαδήποτε στιγμή αυτός επιθυμεί. Κάθε αρχή νέου ακαδημαϊκού εξαμήνου θα έχει την αρμοδιότητα να ανεβάσει ένα αρχείο csv με συγκεκριμένη μορφή το οποίο θα περιέχει τα στοιχεία για δημιουργία λογαριασμών φοιτητών και να δει πληροφορίες σχετικά με αυτούς. Επιπρόσθετα, υπάρχει σχετική λειτουργία βαθμολόγησης φοιτητών με την οποία υπάρχει η δυνατότητα μαζικής βαθμολόγησης (ατόμων της ίδια ομάδας) ή ατομική βαθμολόγησης με την πλατφόρμα να ενημερώνει τον χρήστη μέσω ηλεκτρονικού ταχυδρομείου για ευκολία. Τέλος, ανα πάσα στιγμή μπορεί να δημιουργήσει, να επεξεργαστεί ή να διαγράψει οποιαδήποτε ομάδα στα πλαίσια του μαθήματος έχει δημιουργηθεί έτσι ώστε όλοι οι φοιτητές να έχουν ομάδα για να μπορούν να αναλάβουν κάποιο Ε.Α.

2.1.3. Περιβάλλον Εργασίας

Η πλατφόρμα θα μπορεί να προσαρμόζεται τόσο σε οθόνη κινητών συσκευών όσο και υπολογιστών, ανεξαρτήτως λειτουργικού συστήματος, φυλλομετρητή, διαστάσεων και ανάλυσης οθόνης της συσκευής, που χρησιμοποιεί ο χρήστης. Δηλαδή θα υπάρχει δυνατότητα πρόσβασης στην πλατφόρμα από κάθε είδος ηλεκτρονική με σύνδεση στο διαδίκτυο γι' αυτό τον λόγο είναι απαραίτητο η σελίδα να αλλάζει δομή ανάλογα με τα pixels της οθόνης έτσι ώστε να αποφεύγεται αποτέλεσμα με ελλείπει στοιχεία ή με στοιχεία τα οποία αποκρύπτονται. Γι' αυτό τον λόγο με την χρήση της ετικέτας Grid, όπως φαίνεται και στην Εικόνα 4.5.21, η δομή εναλλάσσεται ανάλογα με το μέγεθος της συσκευής σε 5 διαφορετικά μεγέθη (xs,sm,md,lg,xl) τα οποία αντιστοιχούν στα ποιο κάτω όρια (breakpoints) [17]:

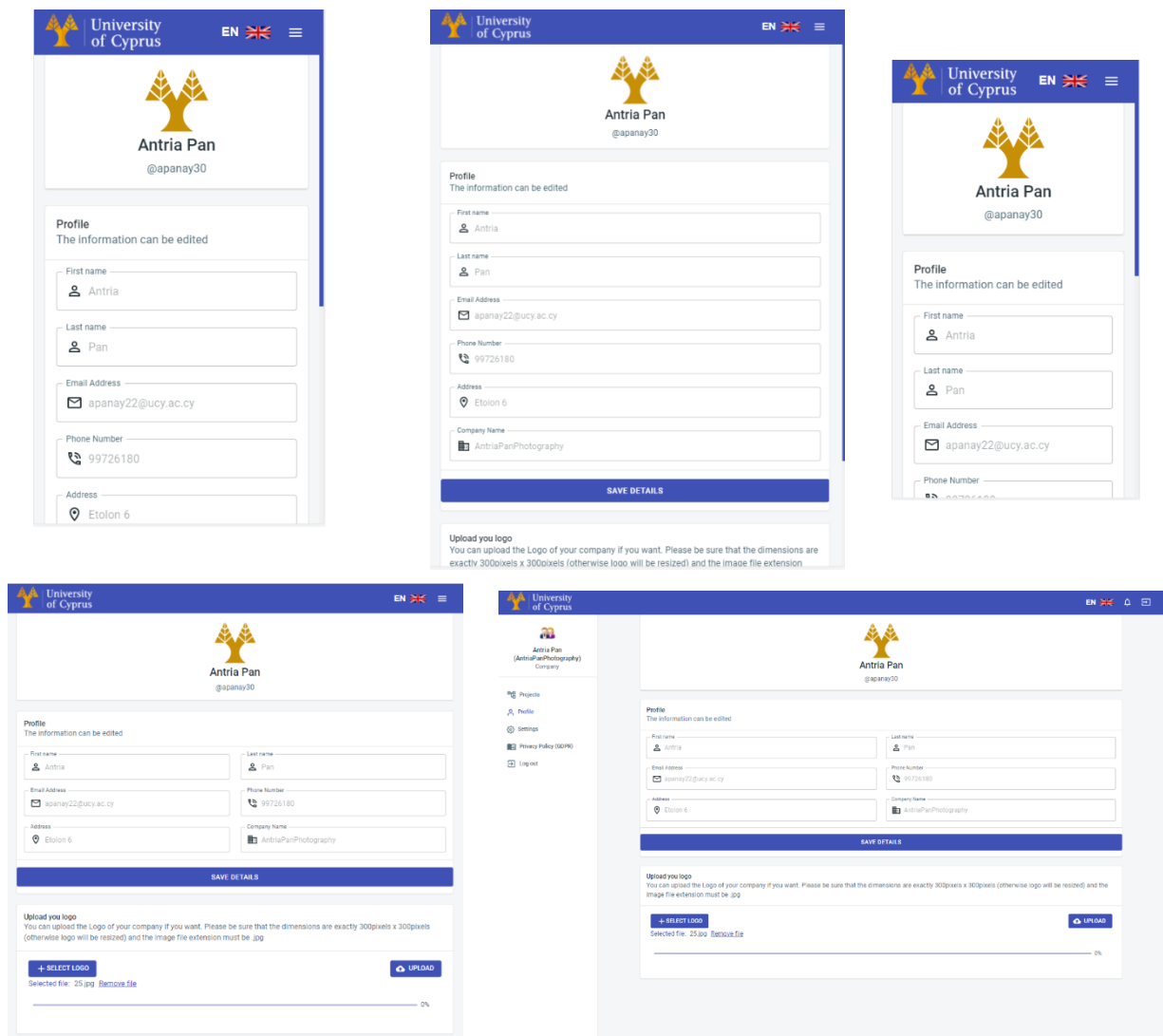
- 320px—480px: Mobile devices
- 481px—768px: iPads, Tablets
- 769px—1024px: Small screens, laptops
- 1025px—1200px: Desktops, large screens



Εικόνα 4.5. 1: Αλλαγή δομή σελίδας ανάλογα pixels

- 1201px and more— Extra large screens, TV

Η εναλλαγή αυτή φαίνεται και στη δομή της σελίδας προφίλ σε διαφορετικά μεγέθη οθονών και διαστάσεων pixels όπως φαίνεται και στην Εικόνα 4.5.22



Εικόνα 4.5. 2: Προβολή πλατφόρμας σε διαφορετικά μεγέθη συσκευών

2.1.4. Περιορισμοί και Παραδοχές συστήματος

Για την υλοποίηση της πλατφόρμας θα χρειαστεί ένας διακομιστής, βάση δεδομένων και υπηρεσίες που θα εκτελούνται σε ένα Virtual Machine με λειτουργικό σύστημα Ubuntu έτσι ώστε να υπάρχει εύκολη απομακρυσμένη σύνδεση με ssh. Η Β.Δ. θα πρέπει να είναι σε θέση να υποστηρίξει περίπου 32000 χιλιάδες χρήστες (υπολογισμός γίνεται στην ενότητα 3.7.3) όπου οι 100 από αυτούς τουλάχιστο θα μπορούν να συνδέονται και να χρησιμοποιούν την ιστοσελίδα ταυτόχρονα. Επιπρόσθετα η εφαρμογή θα πρέπει να είναι σε θέση να εκτελείται σε διάφορα λειτουργικά συστήματα iOS, Android, Windows και Linux με τουλάχιστον 1GB

μνήμης και διπύρηνους επεξεργαστές. Επιπλέον, η συσκευή με την οποία συνδέεται ο χρήστης θα πρέπει να μπορεί να έχει σύνδεση στο διαδίκτυο, να υπάρχει εγκατεστημένος οποιοσδήποτε φυλλομετρητής (π.χ. Internet Explorer, Google Chrome κτλ.) και να προσαρμόζεται δυναμικά σε κάθε διαφορετική συσκευή με διαφορετικές αναλύσεις οθονών. Όσο αφορά τον διακομιστή, ο οποίος θα βρίσκεται σε VM πρέπει να έχει την δυνατότητα να είναι συνεχώς ενεργός και να είναι πάντα προσβάσιμη τόσο η ιστοσελίδα όσο και τα αρχεία που περιέχει. Συμπληρωματικά σε περίπτωση που δημιουργηθούν καινούριες ανάγκες για το μάθημα και πιο εξειδικευμένες πληροφορίες θα χρειαστεί μια πιο εξειδικευμένη βάση δεδομένων και ανανέωση της πλατφόρμας. Τέλος, υπάρχει εγκατεστημένη βιβλιοθήκη η οποία λειτουργεί ως διακομιστής ηλεκτρονικού ταχυδρομείου και αποστέλλει email στους φοιτητές π.χ. ανάρτηση βαθμών, εγγραφή στην ιστοσελίδα, επανάκτηση κωδικού πρόσβασης κτλ. ο οποίος συνεργάζεται με την πλατφόρμα αποστολής μηνυμάτων της Google (το Gmail) για να στέλνονται τα email άρα τόσο το Gmail όσο και το nodemailer πρέπει να λειτουργούν με τον αναμενόμενο τρόπο.

2.2.Προσδιορισμός Αναγκών

2.2.1. Ανάλυση Ερωτηματολογίου

Για να προσδιοριστούν μερικές από τις ανάγκες της ιστοσελίδας δημιουργήθηκε ερωτηματολόγιο το οποίο απαντήσαν φοιτητές του τμήματος πληροφορικής του Πανεπιστημίου Κύπρου οι οποίοι έχουν ήδη παρακολουθήσει το μάθημα. Η γνώμη τους είναι σημαντική καθώς θα μπορούσαν αντικειμενικά να συγκρίνουν καταστάσεις και να επιλέξουν τι πραγματικά θεωρούν χρήσιμο να υπάρχει γεγονός το οποίο έχει υψηλή βαρύτητα καθώς γνωρίζουν ήδη τις δυσκολίες τις οποίες αντιμετώπισαν. Το ερωτηματολόγιο έχει απαντηθεί από περίπου 40 άτομα (είναι προσβάσιμο από το ΠΑΡΑΡΤΗΜΑ Α – Β.1) με διαφορετικές απόψεις και γνώσεις για το μάθημα καθώς είναι διαφορετικών ηλικιών, φύλου και ακαδημαϊκού εξαμήνου. Με το τέλος του ερωτηματολογίου δημιουργήθηκαν γραφικές παραστάσεις με τις ανάγκες των φοιτητών οι οποίες αντικατοπτρίζουν τα αποτελέσματα του ερωτηματολογίου (είναι προσβάσιμο από το ΠΑΡΑΡΤΗΜΑ Α – Β.2)

2.2.2. Γενικά Συμπεράσματα

Τα γενικά συμπεράσματα του ερωτηματολογίου εμφανίζονται πιο κάτω:

- Οι περισσότερες φοιτητές οι οποίοι απάντησα το ερωτηματολόγιο είναι κυρίως άντρες (λογικό αφού αποτελούν την πλειοψηφία του τμήματος Πληροφορικής) - 32 – άντρες , 7 – γυναίκες.
- Η συντριπτική πλειοψηφία πιστεύει ότι η υλοποίηση μίας τέτοιας πλατφόρμας θα ήταν χρήσιμη για την διαχείριση (38 Ναι – 1 Όχι).

- Απόψεις οι οποίες θεωρήθηκαν λιγότερο χρήσιμες από τους φοιτητές και δεν υλοποιήθηκαν στην συγκεκριμένη έκδοση της εφαρμογής είναι η λειτουργία επιθεώρησης και βαθμολόγησης ομάδων από άλλες ομάδες, η συμπλήρωση των ερωτηματολογίων όλων στην πλατφόρμα παρά στο Google Forms και η δυνατότητα βαθμολόγησης από τους φοιτητές στον καθηγητή με το τέλος κάθε διάλεξης για βελτίωση.
- Υπάρχουν και κάποιες ιδέες οι οποίες συγκέντρωσαν πολλές ψήφους είτε υλοποιήθηκαν ακριβώς με την περιγραφή στο ερωτηματολόγιο, είτε με κάποιες μικρές διαφορές είτε θα υλοποιηθούν σε κάποιο μεταγενέστερο στάδιο. Συγκεκριμένα τις πιο πολλές ψήφους πήρε η δυνατότητα πρόσβασης σε προηγούμενες ιδέες παραδοτέων (υλοποιήθηκε) και η διαδικτυακή συμπλήρωση των παραδοτέων (σε μεταγενέστερο στάδιο υλοποίησης της πλατφόρμας).
- Η πλειοψηφία των φοιτητών επέλεξαν ότι θέλει να ενημερώνεται και με τους δύο τρόπους (μέσω μηνύματος στο ηλεκτρονικό ταχυδρομείο, στην πλατφόρμα) – 24 από τους 39 και ακριβώς οι μισοί από τους 24 επιθυμούν να στέλνεται ένα μήνυμα στην ομάδα ενώ οι άλλοι μισοί θέλουν να παίρνουν μήνυμα όλα τα μέλη της ομάδας (υλοποιήθηκε).
- Σχετικά με την μία συμπληρωματική εισήγηση την οποία πήραμε γενικά συνοψίζει τους στόχους της σελίδας αφού επιθυμεί να μπορεί να βρίσκει όλα όσα ψάχνει και ο καθηγητής να διορθώνει εύκολα, να βαθμολογεί και να ανεβάζει ανατροφοδότηση.

2.3.Ανάλυση Αναγκών

2.3.1. Personas

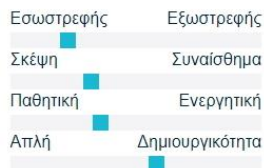
Με τον όρο persona αναφερόμαστε σε ένα συγκεκριμένο είδος νοητού προφίλ χρήστη το οποίο θεωρείτε ιδανικό να χρησιμοποιήσει το προϊόν μας (διαδικτυακή εφαρμογή). Χαρακτηρίζει το target group των ατόμων που επισκέπτονται την πλατφόρμα μας, έχουν κάποια συγκεκριμένα χαρακτηριστικά και λαμβάνουμε υπόψη την πιθανή αντίδραση των ατόμων αυτών σε κάθε αλλαγή η οποία γίνεται. Στόχος είναι να κατανοηθούν πλήρως τα συναισθήματα τα οποία δημιουργούνται στους χρήστες της πλατφόρμας, να αναλυθεί ο τρόπος με τον οποίο θα αντιμετώπιζαν ένα τέτοιο σύστημα αλλά και να καταγραφούν πιθανά προβλήματα. Χρειάζεται επίσης να σημειωθεί ότι τα personas καλό θα ήταν να είναι όσο πιο ρεαλιστικά και ακριβή γίνεται έτσι ώστε να προσαρμοστεί η συμπεριφορά των χρηστών και οι ενέργειες τους μέσα στην εφαρμογή [19].

Φοιτήτρια - Ελευθερία Κωνσταντίνου



"You only live once and you need to enjoy life to go out and achieve whatever you want to."

Προσωπικότητα



Ηλικία: 22

Επάγγελμα: Φοιτήτρια

Οικογενιακή κατάσταση:

Άγαμος

Τοποθεσία: Λεμεσός, Κύπρος

Στόχοι

- Να αποφοιτήσει από το πανεπιστήμιο στα 4 χρόνια.
- Να βελτιώσει τους βαθμούς της
- Να σταματήσει να αναβάλει καταστάσεις
- Να ξεκινήσει γυμναστική

Ενδιαφέροντα

- Ταξίδια και εκδρομές
- Επιτραπέζια με φίλους
- Φωτογραφικές εξορμήσεις

Περιγραφή:

Η Ελευθερία είναι μία νέα, εργατική και δραστήρια φοιτήτρια του Π.Κ. η οποία είναι πολυάσχολη και της αρέσει να περνάει πολλές ώρες με τους φίλους της δοκιμάζοντας καινούριες εμπειρίες. Ταυτόχρονα αφιερώνει κάποιες ώρες την ημέρα για καθημερινό διάβασμα και να κάνει τις πολλές εργασίες της οποίες έχει από το πανεπιστήμιο. Γι' τον λόγο αυτό επισκέπτεται καθημερινά διάφορες βοηθητικές ιστοσελίδες για τα μαθήματα της (Blackboard, Moodle, ιστοσελίδες μαθημάτων κτλ.) και διαβάζει τα emails της για να ενημερωθεί.

Συχνές ιστοσελίδες που επισκέπτεται:

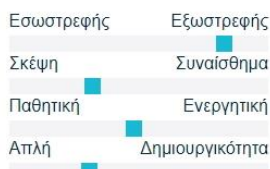


Καθηγητής - Γιάννης Αριστοδήμου



"Teachers encourage, minds to think hands to create & hearts to love."

Προσωπικότητα



Ηλικία: 34

Επάγγελμα: Καθηγητής

Οικογενιακή

κατάσταση: Έγγαμος

Τοποθεσία: Λευκωσία, Κύπρος

Στόχοι

- Να διαβάσει περισσότερα βιβλία
- Να αφιερώνει περισσότερο χρόνο στον εαυτό του
- Να αποκτήσει καλύτερη σχέση με τους φοιτητές (μαθητές του)

Ενδιαφέροντα

- Βόλτες με την οικογένεια
- Ταινίες και σειρές
- Διαβάζει και γράφει βιβλία

Περιγραφή:

Ο Γιάννης θεωρείται επιτυχημένος αφού σε μία σχετικά μικρή ηλικία κατάφερε να είναι καθηγητής σε πανεπιστήμιο μετά από πολλά χρόνια σπουδών στο εξωτερικό (κυρίως Αγγλία και Ολλανδία). Έχει κερδίσει διάφορα βραβεία για τα άρθρα τα οποία έγραψε σε γνωστά περιοδικά όπως το IEEE και συνεχώς προσπαθεί να βελτιώνεται και να μαθαίνει καινούρια πράγματα. Χρησιμοποιεί διάφορες σελίδες για να ανεβάζει υλικό για το μάθημα το οποίο διδάσκει στο πανεπιστήμιο και πρέπει να κοιτάζει καθημερινά διάφορες ιστοσελίδες (Blackboard, Moodle, ιστοσελίδες μαθημάτων, outlook κτλ.) και να στέλνει αρκετά email έτσι ώστε να ενημερώνει τους μαθητές του έγκαιρα. Του αρέσει να διευρίνει τον κύκλο του συνεχώς γνωρίζοντας καινούριους ανθρώπους αφού είναι ιδιαίτερα εξωστρεφής κάτι το οποίο τον βοηθάει και στην δουλειά του γνωρίζοντας άτομα σε διάφορες εταιρίες τα οποία μπορούν να προσφέρουν έργα για ολοκλήρωση στους φοιτητές του. Στον ελεύθερο του χρόνο λατρεύει να περνά χρόνο με την οικογένειά του.

Συχνές ιστοσελίδες που επισκέπτεται:



Υπάλληλος/Διευθυντής Εταιρίας - Γιώργος Γεωργίου



"In order to succeed, your desire for success should be greater than your fear of failure"

Προσωπικότητα

Εσωστρεφής	Εξωστρεφής
Σκέψη	Συναίσθημα
Παθητική	Ενεργητική
Απλή	Δημιουργικότητα

Ηλικία: 48
Επάγγελμα: Υπάλληλος/
 Διευθυντής Εταιρίας
Οικογενιακή
κατάσταση: Έγγαμος
Τοποθεσία: Λάρνακα, Κύπρος

Στόχοι

- Να ξεκινήσει κάποιο καινούριο σπορτ
- Να αυξηθούν τα κέρδη της επιχείρησής του
- Να μάθει καινούρια πράγματα και τεχνολογίες

Ενδιαφέροντα

- Κολύμπι
- Διαβάζει επιστημονικά και οικονομικά περιοδικά
- Σκάκι, sudoku

Περιγραφή:

Ο Γιώργος Γεωργίου ξεκίνησε από μικρή σχετικά ηλικία να εργάζεται στην οικογενιακή επιχείρηση η οποία ασχολείται με διοίκηση εταιριών και προσφέρει βοήθεια και συμβουλές να αναπτυχθούν. Του αρέσει να μελετάει συνεχώς τα νέα τις επικαιρότητας, να βλέπει εμπνευστικά βιντεάκια του εξωτερικού και να μαθαίνει καινούρια πράγματα. Η εταιρία του αποτελείται από μεγάλο προσωπικό και συνεχώς αναλαμβάνει πολλές εταιρίες και projects καθώς είναι αρκετά γνωστή στην αγορά.

Στον ελεύθερο του χρόνο του αρέσει να 'ακονίζει' το μυαλό του καθώς συχνά παίζει σκάκι με τους φίλους του και λύνει sudoku. Επίσης, τα σαββατοκυριακάπηγαί για κολύμπι στην θάλασσα μαζί με την οικογένειά του.

Συχνές ιστοσελίδες που επισκέπτεται:








2.3.2. Σενάρια Χρήσης

Τα σενάρια χρήσης αποτελούνται από νοητές ιστορίες με πρωταγωνιστές τα Personas τα οποία αναλύσαμε στην ενότητα 3.3.1. Συγκεκριμένα, ένα σενάριο χρήσης δείχνει τον τρόπο με τον οποίο θα αντιδρούσε ο χρήστης (persona) στην εκτέλεση συγκεκριμένης λειτουργίας του συστήματος, πως θα ήταν η εμπειρία χρήσης του, τις δυσκολίες που θα αντιμετώπιζε και γενικά την άποψη του για το σύστημα. Ο στόχος δημιουργίας σεναρίων είναι να βοηθήσουν τους σχεδιαστές και προγραμματιστές του συστήματος να μπουν στο μυαλό και την ιδεολογία του χρήστη έτσι ώστε να επιλέξουν την καλύτερη δυνατή εναλλακτική στα πρότυπα οθονών τα οποία δημιουργήθηκαν. Θα αποτελούσε σοβαρή παράλειψη να μην τονίσουμε ότι τα σενάρια χρήσης αντιπροσωπεύουν μία συγκεκριμένη ομάδα χρηστών και όχι όλους τους πιθανούς χρήστες καθώς υπάρχουν και εξειδικευμένες περιπτώσεις οι οποίες δεν εμπίπτουν στο σύνολο. Σημαντικό θεωρείτε επίσης το γεγονός ότι τα σενάρια πρέπει να αναφέρουν το περιβάλλον εργασίας των χρηστών, υπό ποιες συνθήκες εκτελούν κάθε εργασία μαζί με συμπληρωματικές λεπτομέρειες σχετικά με τον τρόπο διεξαγωγής. Αυτό θα βοηθήσει έτσι ώστε τα αποτελέσματα να είναι όσο πιο κοντά γίνεται στην πραγματικότητα, να περιγράφουν μία ρεαλιστική κατάσταση ως προς το πραγματικό σύστημα και να είναι όσο το δυνατό πιο ακριβής.

Σενάριο 1 – Personas (Φοιτήτρια – Ελευθερία Κωνσταντίνου) :

Η Ελευθερία είναι μία 22χρονη φοιτήτρια Πληροφορικής η οποία κατάγεται από την Λεμεσό αλλά λόγω της σπουδής της στο Πανεπιστήμιο Κύπρου διαμένει πλέον Λευκωσία. Ως φοιτήτρια στον ελεύθερο της χρόνο της αρέσει να περνάει χρόνο με τους φίλους και συμφοιτητές της με χαλαρές εξορμήσεις και παίζοντας επιτραπέζια. Ως νεαρή φοιτήτρια με ψηλό φόρτο εργασίας με πολλές παράλληλες εργασίες και υποχρεώσεις χρειάζεται να προγραμματίζεται συχνά έτσι ώστε να υπάρχει αρκετά ελεύθερος χρόνος για διασκέδαση χωρίς αυτό να επηρεάσει τους βαθμούς της. Μερικές φορές όμως είναι λίγο δύσκολο καθώς πολλοί καθηγητές χρησιμοποιούν πολλές διαφορετικές ιστοσελίδες για να ανεβάζουν υλικό και να κάνουν της ενημερώσεις που χρειάζονται για το μάθημα τόσο από φόρουμ όσο και από μηνύματα στο ηλεκτρονικό ταχυδρομείο. Η Ελευθερία στην αρχή της ακαδημαϊκής χρονιάς στο 3^ο έτος γράφτηκε στο μάθημα ΕΠΛ343 και η καθηγήτρια του μαθήματος τους πρότεινε να χρησιμοποιήσουν την πλατφόρμα για να δημιουργήσουν ομάδες και να δουν τις υποχρεώσεις τους. Το απόγευμα στο καθημερινό διάβασμα λαμβάνει μήνυμα εγγραφής από την καθηγήτρια με τα στοιχεία εισόδου της πλατφόρμας και αποφασίζει να δοκιμάσει. Μόλις έχει κάνει σύνδεση στην σελίδα και αντικρίζει ένα πολύ φιλικό περιβάλλον με απλή σχεδίαση, παρόμοια με άλλα συστήματα που είχε χρησιμοποιήσει στο παρελθόν και κατανοητές οδηγίες. Στην αρχή πλοηγείτε στην σελίδα με χρήση του μενού και σε πολύ λίγα λεπτά επισκέφτηκε όλες τις σελίδες ώστε να γνωρίζει της λειτουργίες που υπάρχουν. Βρήκε ιδιαίτερα χρήσιμη την σελίδα των παραδοτέων η οποία περιείχε τις σημαντικές ημερομηνίες για το μάθημα ώστε να μην ξεχάσει κάποια. Στην συνέχεια άλλαξε τον κωδικό πρόσβασης της, αφού ήταν ένας τυχαίος συνδυασμός σχεδιασμένος από την πλατφόρμα που δημιουργήθηκε όταν η καθηγήτρια έφτιαχνε τους λογαριασμούς και τέλος επισκέφτηκε την σελίδα των ομάδων όπου εκεί βρήκε ειδοποίηση για να γίνει μέλος στην ομάδα που έφτιαξαν οι φίλοι της για το μάθημα. Ευχαριστημένη και χαρούμενη που μάζεψε όλες τις χρήσιμες πληροφορίες τόσο γρήγορα αποσυνδέθηκε από την πλατφόρμα και συνέχισε το απογευματινό της διάβασμα.

Συμπεράσματα σεναρίου:

Διαβάζοντας το πιο πάνω σενάριο συμπεραίνουμε ότι είναι ένα πρωτότυπο σύστημα για μάθημα πανεπιστημίου όπου οι περισσότεροι φοιτητές χρησιμοποιούν πολλά μεμονωμένα συστήματα και η χρήση του συγκεκριμένου δημιουργεί ικανοποίηση. Εκτός από αυτό παρατηρούμε ότι αυξάνει την παραγωγικότητα και κάνει τους φοιτητές πιο υπεύθυνους στις υποχρεώσεις τους καθώς του υπενθυμίζει στην αρχική σελίδα τις σημαντικές πληροφορίες του μαθήματος. Επιπλέον, λόγω των τεχνολογικών γνώσεων που έχουν άτομα στην συγκεκριμένα ηλικία, ιδιαίτερα φοιτητές πληροφορικής, η διαδικτυακή εφαρμογή αυτή φαίνεται πολύ απλή για αυτούς και δεν αντιμετωπίζουν σχεδόν κανένα πρόβλημα κατανόησης και χρήσης.

Σενάριο 2 – Personas (Καθηγητής – Γιάννης Αριστοδήμου) :

Ο Γιάννης Αριστοδήμου είναι ένας 34χρονος νέος ο οποίος μετά της πολύχρονες σπουδές του στο εξωτερικό σε ευρωπαϊκές χώρες την Αγγλία και την Ολλανδία αποφάσισε να επιστρέψει πίσω στην Κύπρο και να συνεχίσει να διδάσκει σε πανεπιστήμιο. Στην πρώτη του χρονιά επιστρέφοντας στην Κύπρο ανέλαβε θέση ως αναπληρωτής καθηγητής στο Πανεπιστήμιο Κύπρου για να διδάξει το μάθημα Πληροφορικής ΕΠΛ343 – Software Engineering και αποφάσισε να χρησιμοποιήσει την πλατφόρμα διαχείρισης του μαθήματος την οποία του είχε προτείνει ο καθηγητής εργαστηρίου για να τον εντάξει στο κλίμα και να του μάθει πως διδασκόταν τα προηγούμενα χρόνια το μάθημα. Παίρνει τα στοιχεία εισόδου και πηγαίνοντας στο γραφείο του αποφασίζει να δοκιμάσει την πλατφόρμα και να δημιουργήσει τους λογαριασμούς των φοιτητών όπως του είχε προτείνει ο καθηγητής εργαστηρίου. Η λειτουργία δημιουργίας μαζικών λογαριασμών χρειάζεται ιδική μορφή έτσι για να σιγουρευτεί ο καθηγητής ότι θα ανεβάσει την σωστή μορφή αρχείου πήρε την γνώμη από τον βοηθό εργαστηρίου και επί της ευκαιρίας του περιέγραψε και λίγο τις λειτουργίες της πλατφόρμας. Στην συνέχεια ο καθηγητής δημιούργησε τους λογαριασμούς και ο κάθε φοιτητής έλαβε μήνυμα στο ηλεκτρονικό του ταχυδρομείο με τα στοιχεία εισόδου του για την ιστοσελίδα. Συνέχισε την περιήγηση και ανάρτησε της ημερομηνίες για τα παραδοτέα για όλη την χρονιά ,με την επιλογή όμως να μην εμφανίζονται όλα στους φοιτητές, το ίδιο έπραξε και για τα ερωτηματολόγια. Τέλος, διάβασε κάποια παλιά Ε.Α. τα οποία ήταν αναρτημένα στην πλατφόρμα και επισημασμένα ως κλειστά, λόγο του ότι υλοποιήθηκαν και αποσυνδέθηκε από την πλατφόρμα αφού ήταν πλέον ώρα για να κάνει μάθημα στους φοιτητές.

Συμπεράσματα σεναρίου:

Διαβάζοντας το πιο πάνω σενάριο συμπεραίνουμε ότι παρόλο που είχε κάνει σπουδές στο εξωτερικό και γνώρισε διάφορους τεχνολογικά ανεπτυγμένους τρόπους εκμάθησής στην αρχή χρειάστηκε μία μικρή βοήθεια από τον βοηθό εργαστηρίου ο οποίος είχε προηγούμενη πείρα ώστε να μάθει τις βασικές λεπτομέρειες και ιδιαιτερότητες της πλατφόρμας. Στην συνέχεια παρατηρούμε ότι δεν αντιμετώπισε οποιαδήποτε δυσκολία και εκτέλεσε με επιτυχία αρκετές από τις δυνατότητες της σελίδας χωρίς κανένα πρόβλημα.

Σενάριο 3 – Personas (Υπάλληλος/Διευθυντής – Γιώργος Γεωργίου) :

Ο Γιώργος Γεωργίου είναι ο 48χρονος Διευθυντής του οργανισμού ‘ABC’ η οποία ασχολείται με μεγάλα έργα (projects) οργανισμών για να βελτιώσει τα οικονομικά των οργανισμών και να προτείνει καινούριες καμπάνιες διαφήμισης ώστε να αυξήσουν την κερδοφορία τους. Λόγω του ότι είναι μία οικογενειακή οργανισμός με πολλά χρόνια εμπειρίας στην Κυπριακή αγορά χρειαζόταν κάποια ανανέωση στον τρόπο διαχείρισης του οργανισμού καθώς ειδικά τα

τελευταία χρόνια που είχε αποκτήσει μεγαλύτερη φήμη στην αγορά και ήταν υπεύθυνη για πολλά έργα. Μία μέρα ο ανιψιός του ο οποίος ήταν φοιτητής πληροφορικής στο Πανεπιστήμιο Κύπρου και παρακολουθούσε το μάθημα ΕΠΛ343 – Software Engineering ανέφερε στο θείο του την ευκαιρία αυτή να δημιουργία λογαριασμού στην πλατφόρμα του μαθήματος με σκοπό μία ομάδα φοιτητών να υλοποιήσει κάποια εφαρμογή η οποία θα βοηθούσε την ανανέωση του οργανισμού. Μετά από μία σύσκεψη στην οργανισμός αποφασίστηκε να προχωρήσουν με την δημιουργία λογαριασμού καθώς ήταν μία καλή ευκαιρία και για τις δύο πλευρές να επωφεληθούν. Ο Γιώργος επισκέπτεται για πρώτη φορά την ιστοσελίδα και αφού δεν είχε στοιχεία σύνδεσης δημιουργεί λογαριασμό στην ιστοσελίδα για να μπορεί να έχει πρόσβαση στις λειτουργίες. Μεταφέρεται στην σελίδα δημιουργίας λογαριασμού τύπου εταιρείας συμπληρώνει όλα τα απαραίτητα στοιχεία μαζί και αποδέχεται τους όρους της ιστοσελίδας. Στην συνέχεια συνδέεται στην ιστοσελίδα και παρατηρεί αρχικά να βρίσκεται στην σελίδα προφίλ με στοιχεία του οργανισμού και αποφασίζει να μεταφορτώσει το λογότυπο του οργανισμού του ως εικόνα προφίλ έτσι ώστε να ξεχωρίζει. Στην συνέχεια εξερευνώντας την σελίδας παρατηρεί στην γραμμή μενού την δυνατότητα να μεταφράσει την σελίδα στα ελληνικά και την επιλέγει αφού οι ορολογίες για εκείνον είναι πιο κατανοητές στα ελληνικά. Εξερεύνησε την σελίδα για μερικά λεπτά μέχρι να μεταφερθεί στην σελίδα δημιουργίας Ε.Α. Παρατηρεί ότι δεν υπάρχει κάποιο αναρτημένο αφού μπορεί να δει μόνο τα δικά του και ξεκινάει την διαδικασία δημιουργίας. Αρχικά συμπληρώνει όνομα έργου, περιγραφή έργου, επιλέγει είδος εφαρμογής (desktop λόγω του ότι το ήθελε να εγκατασταθεί στους υπολογιστές των υπαλλήλων στην οργανισμός να χειρίζονται τα έργα τα οποία είχαν αναλάβει) και τέλος ανεβάζει συμπληρωματικό αρχείο με πληροφορίες από την ανακοίνωση του οργανισμού και επιλέγει την δημιουργία. Η βοηθητική ράβδος ανάδρασης τον βοήθησε να καταλάβει την στιγμή την οποία έγινε η ανάρτηση και το ενημερωτικό μήνυμα το οποίο εμφανίστηκε το επιβεβαίωσε. Σε διάστημα μίας ώρας παρατήρησε ότι ήδη μία ομάδα έδειξε ενδιαφέρον στο έργο του και επέλεξε να δει περισσότερες πληροφορίες για τα συγκεκριμένα άτομα. Χαρούμενος πλέον που όπως όλα δείχνουν θα υλοποιηθεί η εφαρμογή του οργανισμού του αποσυνδέεται για να συνεχίσει της δραστηριότητες της ημέρας του.

Συμπεράσματα σεναρίου:

Διαβάζοντας το πιο πάνω σενάριο συμπεραίνουμε ότι ένα άτομο με μειωμένες γνώσεις από τεχνολογία, αφού ο οργανισμός του τα είχε όλα χειρόγραφα, κατάφερε να αναρτήσει Ε.Α. στους φοιτητές παρόλη την αργοπορία που υπήρχε για πρώτη φορά. Επιπλέον, οι λειτουργίες ανατροφοδότησης κατάστασης (ράβδος ανάδρασης και ενημερωτικά μηνύματα) λειτούργησαν πολύ βοηθητικά σε αυτό το είδος χρήστη. Συμπληρωματικά, η δυνατότητα μετάφρασης της

σελίδας τόσο σε αγγλικά όσο και σε ελληνικά φαίνεται ότι θα διευκολύνει αρκετά άτομα που δεν είναι καλοί χειριστές και των δύο γλωσσών.

2.4.Καθορισμός Κανόνων Σχεδιασμού και Ευχρηστίας

Ένας από τους αρχικούς στόχους οι οποίοι ορίστηκαν τόσο από πλευράς καθηγητή όσο και από πλευράς φοιτητή ήταν η πλατφόρμα και συγκεκριμένα η διεπιφάνεια μαζί με την λειτουργικότητα να είναι απλή και κατανοητή από όλα τα είδη χρηστών αφού θα υπάρχουν χρήστες διάφορων ηλικιών. Αυτό μπορούμε να το επιβεβαιώσουμε αν λάβουμε υπόψη τις απαντήσεις στα ερωτηματολόγια και την ανάλυση σεναρίων που έχουν δημιουργηθεί με την χρήση personas. Για την δημιουργία της πλατφόρμας ακολουθήθηκαν κανόνες σχεδιασμού και ευχρηστίας από τους 10 κανόνες του Nielsen αλλά και από τους κανόνες σχεδιασμού NASA. Αποτελούν μέρος της Ευρετικής Αξιολόγησης η οποία αναπτύχθηκε την δεκαετία του 90 από τον Jacob Nielsen και το 2014 το Human Integration Design Processes (HIDP) από το “National Aeronautics and Space Administration International Space Station Program Johnson Space Center Houston, Texas”. Πιο κάτω φαίνονται αναλυτικά οι κανόνες σχεδίασης και ευχρηστίας οι οποίοι καθορίστηκαν και υλοποιήθηκαν στην πλατφόρμα:

2.5.Καθορισμός Απαιτήσεων

2.5.1. Εισαγωγή

Σκοπός της δημιουργίας της πλατφόρμας ήταν η διευκόλυνση διαχείρισης του μαθήματος πληροφορικής ΕΠΛ343 – Τεχνολογία Λογισμικού (Software Engineering) και η αυτοματοποίηση ορισμένων λειτουργιών. Αρχικά έγιναν συναντήσεις με την υπεύθυνη καθηγήτρια πληροφορικής του μαθήματος κ. Γεωργία Καπιτσάκη έτσι ώστε να καταγραφούν απαιτήσεις και λειτουργίες οι οποίες είναι απαραίτητες να υπάρχουν και κάποιες άλλες προαιρετικές ανάλογα με τα αποτελέσματα του ερωτηματολογίου από φοιτητές οι οποίοι θα είναι βασικοί μας χρήστες. Αξίζει, επιπλέον ν’ αναφερθούμε σε συζητήσεις (μικρές συνεντεύξεις) τις οποίες έκανα με φίλους και γνωστούς μου από το τμήμα της πληροφορικής πριν την δημιουργία του αποτελεσμάτων για να καθοριστούν μερικές ανάγκες που θεωρούν οι φοιτητές σημαντικές ώστε να υπάρχουν ως παραδείγματα στο ερωτηματολόγιο. Ωστόσο, για τον τελικό προσδιορισμό των απαιτήσεων προσθέσαμε και δικές μας επιλογές οι οποίες έχουν ληφθεί υπόψη από την έρευνα η οποία έγινε στον ανταγωνισμό παρόμοιων συστημάτων αλλά και από άλλα επιτυχημένα παρόμοια συστήματα.

Με τον όρο απαιτήσεις θεωρούμε ότι αναφερόμαστε στο “τι” πρέπει να κάνει το σύστημα, δηλαδή ποιο θα είναι το αποτέλεσμα και τι θα είναι η ενδιάμεση λειτουργία και όχι το “πώς» δηλαδή τον τρόπο με τον οποίο θα γίνεται η διαδικασία αυτή. Όταν καταγραφούν όλες οι

απαιτήσεις, επόμενο βήμα είναι να καταγραφούν οι τεχνικές απαιτήσεις του συστήματος οι οποίες αποτελούν ένα από τα πιο σημαντικά σημεία που πρέπει να συμφωνηθούν ανάμεσα στον πελάτη και τον προγραμματιστή/κατασκευαστή, έτσι αποφασίζονται και οι ανάγκες που θα έχουν οι χρήστες.

Οι απαιτήσεις συνολικά χωρίζονται σε δύο κύριες κατηγορίες τις λειτουργικές και τις μη λειτουργικές απαιτήσεις. Οι λειτουργικές είναι αυτές που είναι υπεύθυνες να ορίσουν τις υπηρεσίες, λειτουργίες και δυνατότητες τις οποίες θα πρέπει να παρέχει το σύστημα, τις εξόδους τις οποίες θα έχει με συγκεκριμένες εισόδους και πως θα πρέπει να συμπεριφέρεται σε συγκεκριμένες καταστάσεις. Δηλαδή πρόκειται για συναρτήσεις οι οποίες θα εκτελούνται με κάποιος συγκεκριμένο σκοπό με μία συγκεκριμένη είσοδο και παράγουν κάποια συγκεκριμένη έξοδο, παραδείγματα από αυτές φαίνονται στο “ΠΑΡΑΡΤΗΜΑ Γ” το οποίο περιέχει τους ορισμούς των συναρτήσεων ένωσης backend (Node JS) με frontend (React JS). Στην αντίθετη περίπτωση βρίσκονται οι μη λειτουργικές απαιτήσεις οι οποίες αποτελούν περιορισμούς που υπάρχουν στις υπηρεσίες, στις λειτουργίες ή στις συναρτήσεις γενικά στην πλατφόρμα οι οποίες περιορίζουν την χρήση του συστήματος. Για να γίνει πιο σαφές περιγράφουν δηλαδή ιδιότητες του συστήματος που εκφράζονται βάση χρηστικότητα, αποδοτικότητα και ασφάλειας. Συνεπώς κρίνονται ως κρίσιμες καθώς σε περίπτωση που υπάρχει λάθος στις μη λειτουργικές απαιτήσεις και στην κανονική ροή της εκτέλεσης τους τότε το σύστημα δεν θα λειτουργεί με τον αναμενόμενο τρόπο και ίσως είναι πλέον ανούσια η χρήση του.

2.5.2. Λειτουργικές Απαιτήσεις

ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ - ΔΙΑΔΙΚΤΥΑΚΗΣ ΠΛΑΤΦΟΡΜΑΣ		
A/A	ΑΠΑΙΤΗΣΗ (ΤΥΠΟΣ ΛΟΓΑΡΙΑΣΜΟΥ)	ΠΕΡΙΓΡΑΦΗ
1	Εγγραφή στην ιστοσελίδα (οργανισμός)	Ο χρήστης έχει δικαίωμα να εγγραφεί στην ιστοσελίδα χρησιμοποιώντας έγκυρα στοιχεία και αποδεχτεί τους όρους της πλατφόρμας.
2	Σύνδεση στην ιστοσελίδα (οργανισμός, καθηγητής, φοιτητής)	Ο χρήστης έχει τη δυνατότητα να συνδεθεί στην ιστοσελίδα χρησιμοποιώντας ένα έγκυρο email και κωδικό τα οποία ταιριάζουν.
3	Αλλαγή προσωπικών στοιχείων (οργανισμός, καθηγητής, φοιτητής)	Κάθε διαφορετικός τύπος χρήστη έχει την δυνατότητα να αλλάξει τα στοιχεία που αντιστοιχούν στο προφίλ του.

4	Προσθήκη/ Αλλαγή/ Διαγραφή εικόνας προφίλ (οργανισμός)	Ο χρήστης με λογαριασμό οργανισμού μπορεί ανά πάσα στιγμή να ανεβάσει, να αλλάξει ή να διαγράψει την εικόνα προφίλ του (λογότυπο).
5	Προσθήκη/Αλλαγή/Διαγραφή γνώσεων (φοιτητής)	Ο χρήστης με λογαριασμό φοιτητή μπορεί ανά πάσα στιγμή να προσθέσει καινούριες γλώσσες προγραμματισμού και frameworks επιθυμεί, να τα αλλάξει ή να τα διαγράψει.
6	Αλλαγή Κωδικού (οργανισμός, καθηγητής, φοιτητής)	Κάθε χρήστης έχει την δυνατότητα να αλλάξει τον κωδικό πρόσβασης του.
7	Ανάκτηση Κωδικού (οργανισμός, καθηγητής, φοιτητής)	Κάθε χρήστης έχει την δυνατότητα να ανακτήσει τον κωδικό πρόσβασης του σε περίπτωση που τον έχει ξεχάσει ακολουθώντας μίας διαδικασία από βήματα.
8	Αλλαγή ρυθμίσεων για ειδοποιήσεις (οργανισμός, καθηγητής, φοιτητής)	Κάθε χρήστης έχει την δυνατότητα να αλλάξει τις ρυθμίσεις λογαριασμού ανάλογα με την προτίμηση του στην ενημέρωση ειδοποιήσεων.
9	Προβολή κανόνων GDPR (οργανισμός, καθηγητής, φοιτητής)	Κάθε χρήστης ανά πάσα στιγμή μπορεί να διαβάσει τους όρους και κανονισμούς της σελίδας σχετικά με το GDPR.
10	Δημιουργία και αποθήκευση στοιχείων GDPR για τον λογαριασμό του χρήστη (οργανισμός, καθηγητής, φοιτητής)	Κάθε χρήστης μπορεί να ζητήσει να δημιουργηθεί ένα αρχείο το οποίο περιέχει όλες τις πληροφορίες οι οποίες είναι αποθηκευμένες για τον λογαριασμό του και να τις αποθηκεύσει.
11	Αίτηση για διαγραφή λογαριασμού (οργανισμός, καθηγητής, φοιτητής)	Κάθε χρήστης την στιγμή που το επιθυμεί μπορεί να ζητήσει να διαγραφεί ο λογαριασμός του μαζί με τα στοιχεία που είναι αποθηκευμένα για αυτόν.
12	Περιορισμός χρήσης στοιχείων GDPR λογαριασμού (οργανισμός)	Κάθε χρήστης με λογαριασμό τύπου οργανισμού μπορεί να ζητήσει για ένα συγκεκριμένο χρονικό διάστημα να

		περιοριστεί η χρήση των στοιχείων του στην πλατφόρμα.
13	Δημιουργία/Επεξεργασία/Διαγραφή Ε.Α (οργανισμός, καθηγητής)	Χρήστες με λογαριασμό οργανισμού και καθηγητή μπορούν ανά πάσα στιγμή να δημιουργήσουν άπειρα Ε.Α, να τα επεξεργαστούν και να τα διαγράψουν.
14	Επιλογή Ο.Α.Ε (οργανισμός, καθηγητής)	Ο δημιουργός του συγκεκριμένου Ε.Α. δηλαδή χρήστης με λογαριασμό οργανισμού ή καθηγητή μπορεί να επιλέξει την ομάδα που θα το αναλάβει και να δει και επιπλέον πληροφορίες για τα άτομα τις ομάδας (γνώσεις, email κτλ.)
15	Προβολή και περιορισμός αποτελεσμάτων Ε.Α. (οργανισμός, καθηγητής, φοιτητής)	Χρήστες με λογαριασμό τύπου οργανισμού μπορούν να δουν Ε.Α. τα οποία δημιούργησαν οι ίδιοι ενώ αντίθετα καθηγητές και φοιτητές μπορούν να τα δουν όλα. Επιπλέον, ανάλογα με τα Ε.Α. μπορούν να περιορίσουν τα αποτελέσματα με φίλτρο ανάλογα με την κατάσταση έργου ή το είδος υλοποίησης.
16	Δημιουργία/Επεξεργασία/Διαγραφή παραδοτέων (καθηγητής)	Χρήστες με λογαριασμό καθηγητή μπορούν ανά πάσα στιγμή να δημιουργήσουν άπειρα παραδοτέα, να τα επεξεργαστούν και να τα διαγράψουν.
17	Δημιουργία/Επεξεργασία/Διαγραφή ερωτηματολόγια (καθηγητής)	Χρήστες με λογαριασμό καθηγητή μπορούν ανά πάσα στιγμή να δημιουργήσουν άπειρα ερωτηματολόγια, να τα επεξεργαστούν και να τα διαγράψουν.
18	Προβολή παραδοτέων (καθηγητής, φοιτητής)	Χρήστες με λογαριασμό καθηγητή, φοιτητή μπορούν ανά πάσα στιγμή να δουν τα παραδοτέα.
19	Προβολή ερωτηματολογίων (καθηγητής, φοιτητής)	Χρήστες με λογαριασμό καθηγητή, φοιτητή μπορούν ανά πάσα στιγμή να δουν τα ερωτηματολόγια.

20	Βαθμολόγηση παραδοτέων ανά ομάδα και ανά φοιτητή (καθηγητής)	Χρήστες με λογαριασμό καθηγητή έχουν την δυνατότητα να βαθμολογούν οποιοδήποτε παραδοτέο είτε ανά ομάδα είτε ανά φοιτητή.
21	Προβολή βαθμών (καθηγητής, φοιτητής)	Κάθε χρήστης με λογαριασμό φοιτητή μπορεί να δει πληροφορίες για τους βαθμούς του ενώ οι καθηγητές πληροφορίες για όλους τους φοιτητές.
22	Αποστολή και προβολή αιτημάτων σε φοιτητές (φοιτητής)	Κάθε χρήστη με λογαριασμό φοιτητή μπορεί να στείλει αίτημα σε όλες τις ομάδες ότι ψάχνει ομάδα ή σε συγκεκριμένο χρήστη να συνδεθεί στην ομάδα του και εννοείται να δει τα αιτήματα τα οποία πήρε από άλλους φοιτητές.
23	Προβολή πληροφοριών για άτομα της ομάδας του (φοιτητής)	Μπορεί ανά πάσα στιγμή ο χρήστης με λογαριασμό φοιτητή να δει κάποιες πληροφορίες σχετικά με τα άλλα μέλη της ομάδας του.
24	Προβολή φοιτητών χωρίς ομάδας (καθηγητής)	Χρήστες με λογαριασμό καθηγητή μπορούν να δουν φοιτητές οι οποίοι δεν έχουν ομάδα έτσι ώστε να κάνουν ποιο σωστό διαμοιρασμό ομάδων.
25	Δημιουργία/Διαγραφή/Επικύρωση ομάδας (καθηγητής)	Χρήστες με λογαριασμό καθηγητή μπορούν να δημιουργήσουν, να διαγράψουν αλλά και να επικυρώσουν οποιαδήποτε ομάδα με φοιτητές.
26	Προσθήκη φοιτητή σε συγκεκριμένη ομάδα (καθηγητής)	Κάθε χρήστης με λογαριασμό τύπου καθηγητή έχει την δυνατότητα να προσθέσει οποιοδήποτε άτομο σε οποιαδήποτε ομάδα αν αυτός το επιθυμεί.
27	Ένδειξη ενδιαφέροντος σε Ε.Α. (φοιτητής)	Κάθε χρήστης φοιτητής έχει την δυνατότητα να δείξει ενδιαφέρον σε οποιοδήποτε Ε.Α. υπάρχει έτσι ώστε να το αναλάβει με την ομάδα του.
28	Δημιουργία μαζικών λογαριασμών φοιτητών (καθηγητής)	Κάθε καθηγητής έχει την δυνατότητα να μεταφορτώσει αρχείο με συγκεκριμένη μορφή

		έτσι ώστε να δημιουργηθούν οι λογαριασμοί φοιτητών που περιγράφονται στο αρχείο.
29	Αποσύνδεση (οργανισμός, καθηγητής, φοιτητής)	Κάθε χρήστης έχει την δυνατότητα να αποσυνδεθεί όποτε αυτός επιθυμεί.
30	Αλλαγή γλώσσας (οργανισμός, καθηγητής, φοιτητής)	Κάθε χρήστης έχει την δυνατότητα να αλλάζει την γλώσσα προβολής της σελίδας ανάλογα με την προτίμησή του.

2.5.3. Μη λειτουργικές Απαιτήσεις

ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ - ΔΙΑΔΙΚΤΥΑΚΗΣ ΠΛΑΤΦΟΡΜΑΣ		
A/A	ΑΠΑΙΤΗΣΗ	ΠΕΡΙΓΡΑΦΗ
Ευχρηστία		
1	Εύκολο στην εκμάθηση (Learnability)	Η εκμάθηση είναι ιδιαίτερα χρήσιμη ιδικά σε μη έμπειρους χρήστες οι οποίοι δεν έχουν τις απαραίτητες γνώσεις από προηγούμενα συστήματα. Υπάρχουν όμως τεχνικές οι οποίες βοηθούν τους χρήστες να αποφεύγουν τα λάθη και να μαθαίνουν το σύστημα πιο εύκολα, για παράδειγμα εικονίδια τα οποία αντικατοπτρίζουν την λειτουργία τους και δίνουν την δυνατότητα να προβλέψει ο χρήστης την συνέχεια.
Απόδοση		

2	Ταχύτητα απόκρισης	Η ταχύτητα απόκρισης του συστήματος την στιγμή που πλοηγείτε στο σύστημα και εκτελεί λειτουργίες πρέπει να είναι μικρή σχετικά σε χρόνο (κλάσματα δευτερολέπτου) έτσι ώστε να μην δημιουργεί αίσθηση καθυστέρησης στον χρήστη.
3	Ταυτόχρονοι χρήστες πλατφόρμας	Λόγω του ότι πρόκειται για μάθημα πανεπιστημίου με ακροατήριο περίπου 80 άτομα η ιστοσελίδα πρέπει να είναι σε θέση να ανταποκριθεί χωρίς καμία δυσκολία σε τουλάχιστο 100 άτομα την ίδια χρονική στιγμή χωρίς να αντιμετωπίζει κάποιο πρόβλημα.
Ασφάλεια		
4	Προστασία προσωπικών δεδομένων χρήστη	Τα δεδομένα όλων των χρηστών είναι αποθηκευμένα στην Β.Δ. και άρα υπάρχει πρόσβαση σε αυτά από τους διαχειριστές ανά πάσα στιγμή. Ο κωδικός πρόσβασης ο οποίος δίνει πρόσβαση στις λειτουργίες αποθηκεύεται στην βάση κρυπτογραφημένος σε 64bit συμβολοσειρά. Με αυτό τον τρόπο ο διαχειριστής δεν μπορεί να αποκρυπτογραφήσει τον κωδικό.
5	Προστασία ιδιωτικών πόρων και δεδομένων που βρίσκονται στο διακομιστή	Οι λειτουργίες και τα ερωτήματα προς την Β.Δ. να μπορούν να γίνουν με επιτυχία μόνο από εξουσιοδοτημένους χρήστες που έχουν συνδεθεί στην πλατφόρμα.
6	JavaScript Injection	Η τεχνολογία JavaScript χρησιμοποιείται ευρέως σε εφαρμογές που βασίζονται στο Web τόσο για καλό όσο και για κακόβουλο σκοπό. Σε κακόβουλη περίπτωση τα άτομα μπορούν να τροποποιήσουν τις υπάρχουσες πληροφορίες στην ιστοσελίδα όπως για παράδειγμα τα πεδία και ετικέτες σε μία

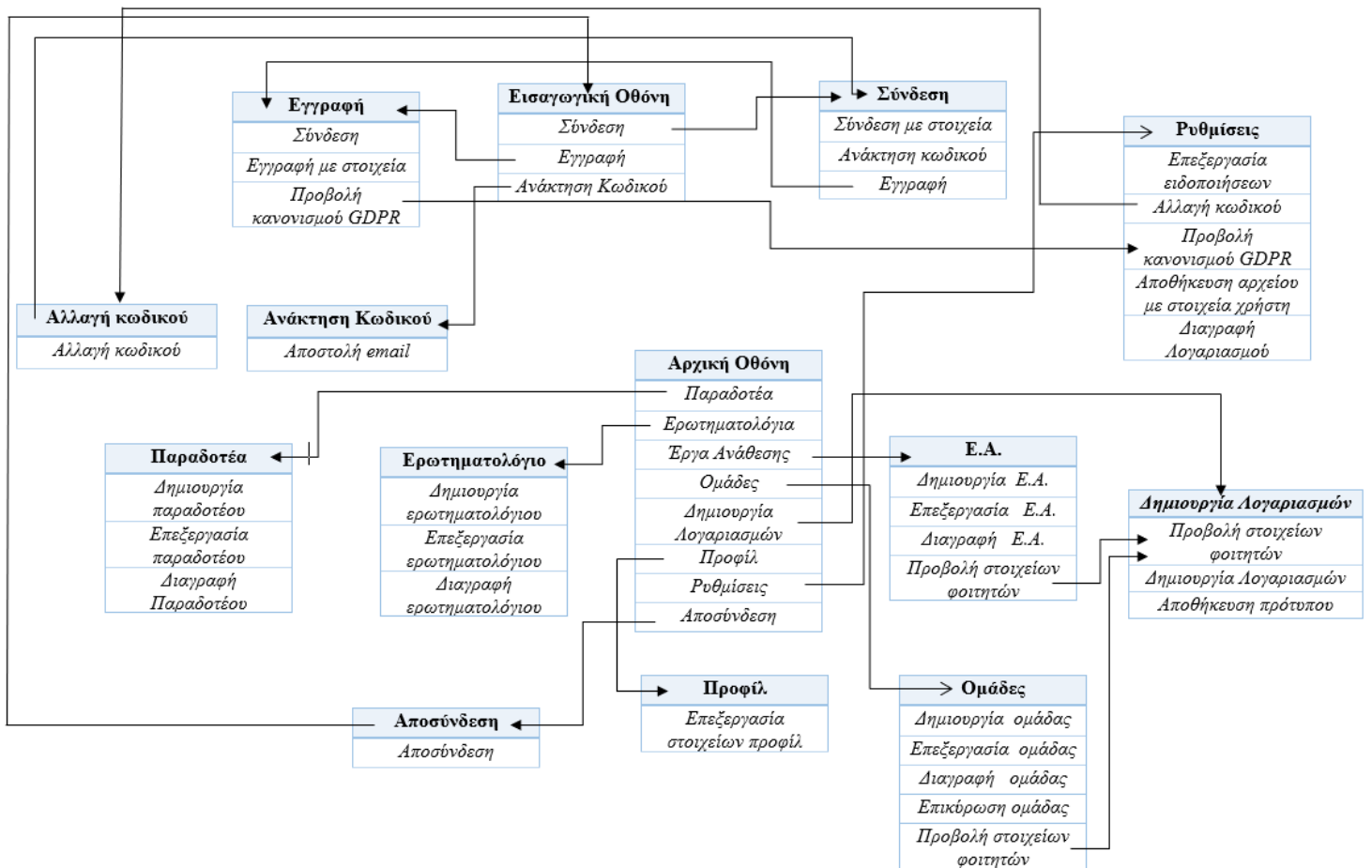
	φόρμα, αποθήκευση των cookies χωρίς κρυπτογράφηση και γενικά οποιωνδήποτε άλλων τιμών οι οποίοι υπάρχουν στον ιστότοπο ή την εφαρμογή ιστού. Χρησιμοποιώντας το JavaScript Injection υπάρχουν στοιχεία ελέγχου για όλες τις φόρμες όπου οι χρήστες ίσως θέλουν κακόβουλα να εισάγουν δεδομένα στη βάση δεδομένων, μπορούμε να αποτρέψουμε τις επιθέσεις με την τεχνική αυτή στον μεγαλύτερο βαθμό [20] .
7 SQL Injection	Οι επιθέσεις SQL περιλαμβάνουν εισαγωγή ή "έγχυση" ερωτημάτων SQL σε εφαρμογές μέσω δεδομένων εισόδου πελάτη με σκοπό να κλέψουν ευαίσθητα δεδομένα από τη Β.Δ., να τροποποιήσει δεδομένα Β.Δ. (εισαγωγή / επεξεργασία / διαγραφή), να εκτελέσει επικίνδυνες λειτουργίες Β.Δ. (π.χ. διαγραφή στοιχείων) και ανάκτηση αρχείων Β.Δ.. Η επίθεση έγχυσης SQL είναι μια επίθεση έγχυσης στην οποία οι δηλώσεις SQL (π.χ. Antria ‘ SELECT * FROM users’’) εισάγονται στην είσοδο του επιπέδου δεδομένων (εισαγωγή δεδομένων σε μια φόρμα) για την εκτέλεση προκαθορισμένων δηλώσεων SQL [21]. Υπάρχουν διάφοροι τρόποι να αποκρύψουμε τέτοιες επιθέσεις αρχικά ελέγχοντας τις φόρμες στις οποίες οι χρήστες μπορούν να εισάγουν πληροφορίες και επίσης εισαγάγουμε τα δεδομένα με μορφή μεταβλητών και χρήση ‘?’ ώστε να αποτρέπει απευθείας τέτοιες επιθέσεις SQL Injection [22].
Ιδιωτικότητα	

8	Απόκρυψη προσωπικών πληροφοριών των φοιτητών από χρήστες με λογαριασμό οργανισμού	Οι χρήστες με λογαριασμό τύπο οργανισμού αναθέτουν τα έργα στις διάφορες ομάδες και μπορούν να δουν επιπλέον πληροφορίες για τις ομάδες που έδειξαν ενδιαφέρον χωρίς όμως την ταυτοποίηση τους με όνομα και επώνυμο μέχρι την στιγμή της ανάθεσης έργου.
Αξιοπιστία		
9	Σωστή μεταφόρτωση αρχείων χωρίς να καταστραφούν	Η ιστοσελίδα πρέπει να μπορεί να εγγυηθεί την επιτυχημένη μεταφόρτωση αρχείων τόσο φωτογραφιών για τις εικόνες προφίλ όσο και αρχεία που περιέχουν συμπληρωματικές πληροφορίες.
10	Ενημέρωση χρηστών σε σημαντικές στιγμές π.χ. ανάρτηση βαθμών	Η ιστοσελίδα πρέπει να είναι σε θέση να αποστέλλει email ακριβώς την στιγμή την οποία ένας καθηγητής βαθμολογεί ένα άτομο έτσι ώστε να μην υπάρχει καθυστέρηση και ο παραλήπτης να είναι πάντοτε ο σωστός για να μην πληροφορήσουμε λανθασμένα άλλο άτομο και να δημιουργηθεί σύγχυση.
Διαθεσιμότητα		
11	24ωρη διαθεσιμότητα πλατφόρμας, διακομιστή και Β.Δ	Η πλατφόρμα πρέπει να είναι ενεργή και να προσφέρει της υπηρεσίες της 24 ώρες την ημέρα καθώς δεν υπάρχουν συγκεκριμένα ωράρια χρήσης αλλά αντιθέτως Β.Δ. και διακομιστής πρέπει να είναι και αυτοί ενεργοί μαζί με την πλατφόρμα για να υπάρχει πρόσβαση στα δεδομένα.

2.6.Διάγραμμα Ροής Εμπειρίας Χρήστη

Λαμβάνοντας υπόψη τις απαιτήσεις οι οποίες περιγράφονται στην ενότητα 3.5 δημιουργήθηκαν τρία διαγράμματα χρήσης (data flow) τα οποία δείχνουν τις λειτουργίες και τις δυνατότητες κάθε διαφορετικής ομάδας χρηστών όπως περιγράφονται και πιο κάτω. Αυτού του είδους τα διαγράμματα σχεδιάζονται κυρίως για να υπάρχει ο τρόπος πλοήγησης μέσα

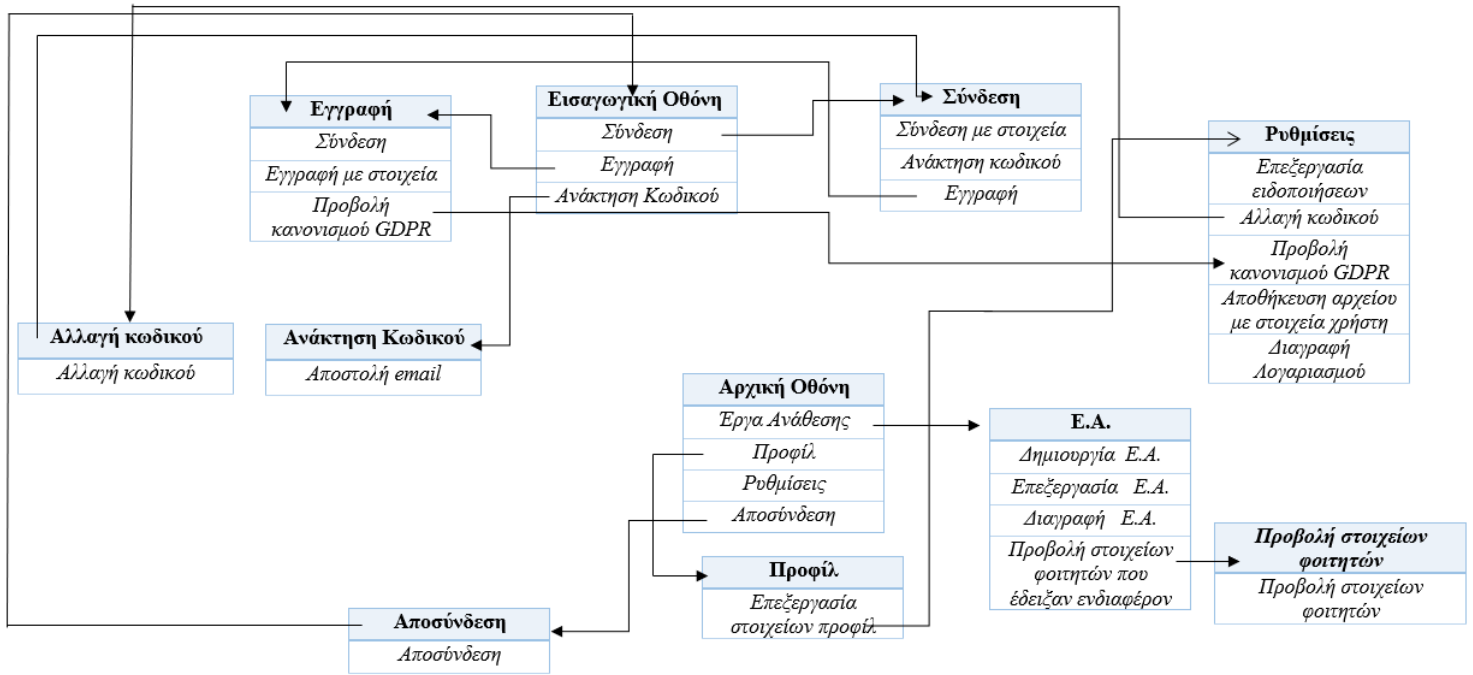
στην σελίδα και η σύνδεση των διάφορων πρότυπων οθονών μεταξύ τους έτσι ώστε να περιγράφεται η εμπειρία χρήσης καθορίζοντας όλες τις οθόνες που βλέπει κάθε διαφορετική ομάδα χρηστών μαζί με τις απαιτήσεις οι οποίες περιεγράφηκαν στην προηγούμενη ενότητα.



Εικόνα 3.6. 1: Data Flow εμπειρίας καθηγητή

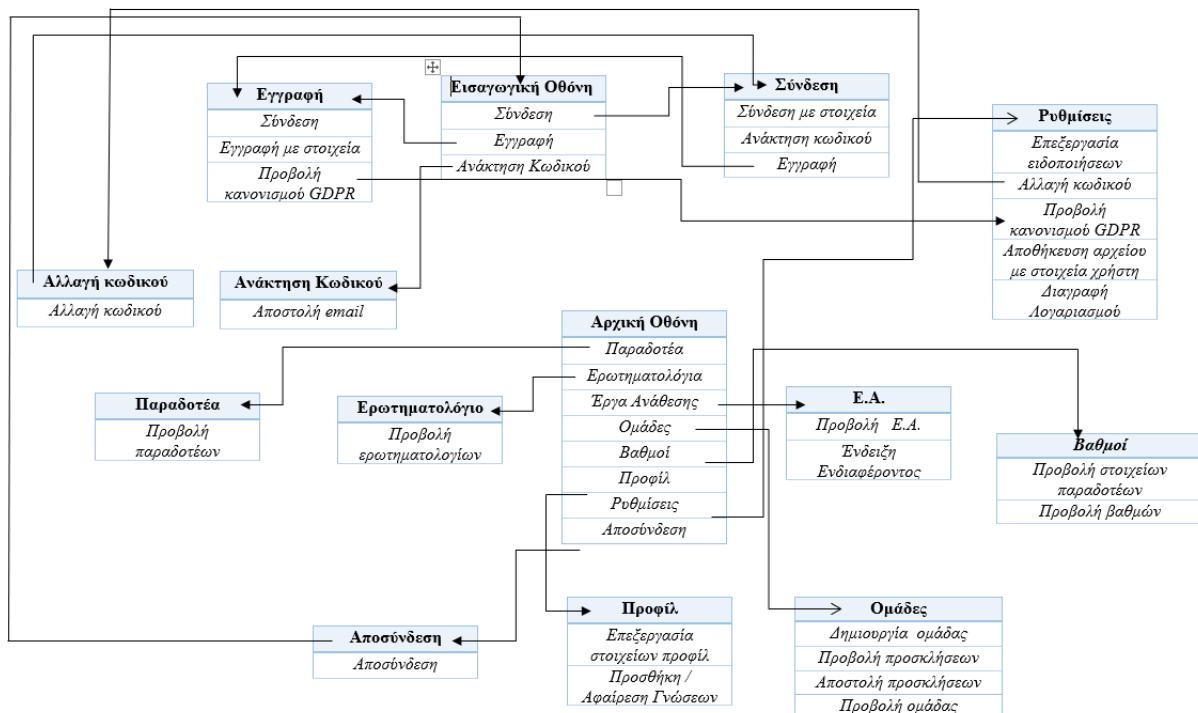
2.6.1. Διαγράμματα ροής υποσελίδα καθηγητή

2.6.2. Διαγράμματα ροής υποσελίδα οργανισμού



Εικόνα 3.6. 2: Data Flow εμπειρίας οργανισμού

2.6.3. Διαγράμματα ροής υποσελίδα φοιτητή



Εικόνα 3.6. 3: Data Flow εμπειρίας φοιτητή

2.7. Σχεδίαση Βάσης Δεδομένων

Για την αποθήκευση των πληροφοριών που χειρίζεται η πλατφόρμα μας θεωρήθηκε πιο κατάλληλη, μεταξύ άλλων, η χρήση της MySQL. Η MySQL είναι ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων με περισσότερες από 11 εκατομμύρια εγκαταστάσεις παγκοσμίως. Ο όρος "σχεσιακή βάση δεδομένων" αναφέρεται σε μια συλλογή δεδομένων που οργανώνονται σε σχετικούς πίνακες, παρέχοντας παράλληλα έναν μηχανισμό ανάγνωσης, γραφής, τροποποίησης ή εκτέλεσης ακόμη πιο πολύπλοκων διαδικασιών στα δεδομένα. Ο σκοπός της είναι να οργανώσει την αποθήκευση πληροφοριών σύμφωνα με τις ερωτήσεις (queries) που τίθενται στη σχεσιακή βάση δεδομένων και να έχει τη δυνατότητα εξαγωγής αυτών των πληροφοριών σε μια ιδιαίτερα οργανωμένη μορφή. Η σχεσιακή βάση δεδομένων σχεδιάστηκε από τον Edgar Codd το 1970 [1]. Σύμφωνα με τους όρους της άδειας GNU General Public License, το MySQL είναι ένα δωρεάν λογισμικό ανοιχτού κώδικα και μπορεί επίσης να χρησιμοποιηθεί με διάφορες άλλες άδειες χωρίς να υπάρξει οποιοδήποτε πρόβλημα.



Η MySQL προτιμήθηκε ανάμεσα σε άλλες επιλογές αφού αποτελεί μία σταθερή, αξιόπιστη και ισχυρή λύση με κάποια πολύ σημαντικά χαρακτηριστικά – κλειδιά. Συγκεκριμένα θεωρείται παγκοσμίως ως το πιο ασφαλές και αξιόπιστο σύστημα διαχείρισης βάσεων δεδομένων μαζί με την απaráμιλλη επεκτασιμότητα που προσφέρει, στην διαχείριση των ενσωματωμένων εφαρμογών της, με μικρό σχετικά αντίκτυπο ακόμη και σε μαζικές αποθήκες τάξεως Terabyte δεδομένων. Επιπλέον, δίνει την δυνατότητα στους διαχειριστές του συστήματος να διαμορφώσουν τον διακομιστή (server) με σκοπό την άψογη απόδοση και μαζί με ευκολία της διαχείρισης που την συνοδεύει εξοικονομεί χρόνο αντιμετώπισης προβλημάτων την καθιστά την πλέον καλύτερη επιλογή για να αποθηκεύουμε πληροφορίες της πλατφόρμας.

Η γλώσσα προγραμματισμού η οποία χρησιμοποιείται για δημιουργία, συντήρηση και ανάκτηση της σχεσιακής βάσης δεδομένων είναι η SQL (Structured Query Language). Παρόλο που υπάρχει συγκεκριμένο ISO πρότυπο (9075-1)² που σχετίζεται με την ορισμό της SQL, οι περισσότερες εφαρμογές και η σύνταξη είναι ελαφρώς διαφορετικές.

Πιο κάτω φαίνονται κάποιες πληροφορίες σχετικά με την έκδοση που χρησιμοποιείται στην πλατφόρμα:

- Έκδοση: 8.0.18
- Documentation: <http://dev.mysql.com/doc/>

2.7.1. Διάγραμμα Οντοτήτων Συσχετίσεων

² ISO/IEC 9075-1:2008

Πιο κάτω υπάρχει ένα διάγραμμα σχέσεων οντοτήτων (ERD) – Εικόνα 3.7.1.1 το οποίο δείχνει τις σχέσεις των πινάκων που είναι αποθηκευμένες στην βάση δεδομένων μας, τα χαρακτηριστικά τους μαζί με τον τύπο δεδομένων τους αλλά και τις ιδιότητες τους. Η κατασκευή αυτού του σχήματος λαμβάνει υπόψη τις αρχές του σχεδιασμού της βάσης δεδομένων. Πληροί τις λειτουργικές απαιτήσεις του χρήστη, επιταχύνει την αποθήκευση και ανάκτηση δεδομένων και αποφεύγετε την περιττή χρήση της μνήμης του συστήματος. Το γράφημα συσχέτισης υιοθετεί τη δεύτερη κοινή του μορφή (2NF) [25], αποτρέποντας έτσι τον πλεονασμό πληροφοριών. Υπάρχουν διάφορα είδη σχέσης των διαφόρων οντοτήτων και περιγράφεται η σημασία τους στην Εικόνα 3.7.1.2. Για παράδειγμα από το πιο πάνω σχήμα μπορούμε να δούμε ότι ο πίνακας “teams” μαζί με τον πίνακα ”grades” έχουν σχέση “one to many” δηλαδή κάθε ομάδα έχει πολλούς βαθμούς αλλά μια εγγραφή βαθμού πρέπει να είναι συσχετισμένη με μόνο μία ομάδα. Ενώ για παράδειγμα οι πίνακες “teams”-“accounts” έχουν την σχέση “one to one (mandatory)” όπου ένα account μπορεί να έχει μία ομάδα αλλά μπορεί και να μην έχει (στην αρχικοποίηση, όταν δεν έχει ομάδα, είναι αρχικοποιημένο με null). Επιπλέον ακόμα μία πληροφορία που μπορούμε να δούμε από την βάση δεδομένων είναι τα πεδία κάθε πίνακα μαζί με τον τύπο δεδομένων τους, επιπλέον χαρακτηριστικά (π.χ. το πρωτεύον κλειδί, ξένο κλειδί) και κατά πόσο έχουν την δυνατότητα να είναι κενά (NULL) ή όχι. Οι πληροφορίες αυτές εμφανίζονται σε εμάς μέσω των εικονιδίων δίπλα από κάθε πεδίο

2.7.2. Πίνακες Βάσης Δεδομένων

Πιο κάτω παρουσιάζονται οι πίνακες της Βάσης Δεδομένων μαζί με μία περιγραφή για τον τρόπο που χρησιμοποιούνται στην πλατφόρμα μας και το πρωτεύον κλειδί τους το οποίο μετατρέπει τις εγγραφές τους σε μοναδικές.

- Πίνακας 1 - “projecttypes”

Πρωτεύον κλειδί: IDType (αύξων μοναδικός αριθμός)

Στον συγκεκριμένο πίνακα αποθηκεύεται η αντιστοιχία των αριθμών με την λεκτική τους σημασία για κάθε ένα από τα διαφορετικά είδη projects που υπάρχουν. Μέχρι στιγμής υπάρχουν 5 διαφορετικά “Web application”, “Desktop application”, “Mobile application”, “Not sure yet” και “Any type”.

- Πίνακας 2 - “teamrequests”

Πρωτεύον κλειδί: IDRequest (αύξων μοναδικός αριθμός)

Στον συγκεκριμένο πίνακα αποθηκεύονται τα αιτήματα που κάνουν οι φοιτητές με σκοπό να βρουν ομάδα για την ανάθεση έργου. Συγκεκριμένα υπάρχουν 2 περιπτώσεις, η πρώτη είναι ένας φοιτητής να καλέσει ένα άλλο φοιτητή για να γίνει μέλος στην ομάδα του τότε θα αποθηκευτεί η ομάδα του φοιτητή και ο προσκεκλημένος φοιτητής. Η δεύτερη περίπτωση είναι ένας φοιτητής να ψάχνει ομάδα τότε στέλνεται πρόσκληση σε όλες τις ομάδες σε περίπτωση που έχουν κενή θέση και επιθυμούν άλλο μέλος.

- Πίνακας 3 - “projects”

Πρωτεύον κλειδί: IDProjects (αύξων μοναδικός αριθμός)

Στον συγκεκριμένο πίνακα αποθηκεύονται πληροφορίες σχετικά με κάθε έργο ανάθεσης που υπάρχει αναρτημένο στην σελίδα. Οι πληροφορίες αυτές είναι το αναγνωριστικό του λογαριασμού που δημιούργησε το έργο, το όνομα του έργου, ο αριθμός της ομάδας που στο τέλος θα είναι υπεύθυνη για ολοκλήρωση, μία μικρή επιπλέον περιγραφή για το έργο, κατά πόσο είναι ακόμη ενεργό ή όχι, ο τύπος του (Πίνακας 1) και η ημερομηνία ανάθεσης.

- Πίνακας 4 - “deliverables”

Πρωτεύον κλειδί: IDDeliverable (αύξων μοναδικός αριθμός)

Στον συγκεκριμένο πίνακα αποθηκεύονται πληροφορίες σχετικά με κάθε παραδοτέο που καλείται να παραδώσει ο φοιτητής και που υπεύθυνος για δημιουργία του είναι ο καθηγητής. Οι πληροφορίες αυτές είναι το όνομα του παραδοτέου, η βαρύτητα στον τελικό του βαθμό, η

μέγιστη βαθμολογία, κατά πόσο θα είναι ορατό ή όχι στους φοιτητές και η ημερομηνίες δημιουργίας και παράδοσης.

- Πίνακας 5 - “questionaries”

Πρωτεύον κλειδί: IDQuestionarie (αύξων μοναδικός αριθμός)

Στον συγκεκριμένο πίνακα αποθηκεύονται πληροφορίες σχετικά με κάθε ερωτηματολόγιο που καλείται να συμπληρώσει ο φοιτητής και ανεβάζει ο καθηγητής. Οι πληροφορίες αυτές είναι το όνομα του ερωτηματολογίου, ο υπερσύνδεσμος που βρίσκεται αναρτημένο, κατά πόσο θα είναι ορατό ή όχι στους φοιτητές και η τελική ημερομηνία παράδοσης του.

- Πίνακας 6 - “instructors”

Πρωτεύον κλειδί + ξένο κλειδί: IDInstructor (από πίνακα 14)

Στον συγκεκριμένο πίνακα αποθηκεύονται επιπλέον στοιχεία για λογαριασμό των χρηστών της πλατφόρμας που θα έχουν ειδικότητα καθηγητή. Συγκεκριμένα εκτός από τα ήδη αποθηκευμένα στον “πίνακα 14” θα αποθηκευτεί επίσης και το ονοματεπώνυμο του καθηγητή.

- Πίνακας 7 - “typeaccounts”

Πρωτεύον κλειδί: IDType (αύξων μοναδικός αριθμός)

Στον συγκεκριμένο πίνακα αποθηκεύεται η αντιστοιχία των αριθμών με την λεκτική τους σημασία για κάθε ένα από τα διαφορετικά είδη λογαριασμού που υπάρχουν. Μέχρι στιγμής υπάρχουν 3 διαφορετικοί τύποι λογαριασμού “Καθηγητή”, ”Οργανισμού” και ”Φοιτητή”.

- Πίνακας 8 - “companies”

Πρωτεύον κλειδί + ξένο κλειδί: IDCompany (από πίνακα 14)

Στον συγκεκριμένο πίνακα αποθηκεύονται επιπλέον στοιχεία για λογαριασμό των χρηστών της πλατφόρμας που θα έχουν ειδικότητα οργανισμού. Συγκεκριμένα εκτός από τα ήδη αποθηκευμένα στον “πίνακα 14” θα αποθηκευτεί επίσης και το όνομα του οργανισμού, η διεύθυνση του οργανισμού, το τηλέφωνο επικοινωνίας, το ονοματεπώνυμο του ατόμου που είναι υπεύθυνο για επικοινωνία, ένα λογότυπο του οργανισμού και κατά πόσο ο λογαριασμός έχει περιορισμούς ή όχι.

- Πίνακας 9 - “activatecodes”

Πρωτεύον κλειδί + ξένο κλειδί: AccountID (από πίνακα 14)

Στον συγκεκριμένο πίνακα αποθηκεύονται οι μίας χρήσης κωδικοί όσο αφορά λειτουργίες για επαναφορά κωδικού και ενεργοποίηση λογαριασμού. Οι πληροφορίες που είναι

αποθηκευμένες είναι ο μοναδικός κωδικός και αν χρησιμοποιήθηκε ή όχι για να περιορίσουν την χρήση στην μία φορά.

- Πίνακας 10 - “teams”

Πρωτεύον κλειδί: TeamNumber (αύξων μοναδικός αριθμός)

Στον συγκεκριμένο πίνακα αποθηκεύονται πληροφορίες σχετικά με τις ΟΑΕ. Συγκεκριμένα αν έχουν γίνει αποδεκτές από κάποιο λογαριασμό καθηγητή και αν είναι ακόμη ενεργές για το τρέχον έτος.

- Πίνακας 11 - “interestedteamsinprojects”

Πρωτεύον κλειδί: IDInterest (αύξων μοναδικός αριθμός)

Στον συγκεκριμένο πίνακα αποθηκεύονται πληροφορίες σχετικά με το ενδιαφέρον που δείχνουν οι χρήστες για να αναλάβουν ένα ΕΑ. Συγκεκριμένα είναι αποθηκευμένη η ομάδα που έδειξε ενδιαφέρον, το αναγνωριστικό του ΕΑ και κατά πόσο έγινε η ανάθεση από τον λογαριασμό που το δημιούργησε.

- Πίνακας 12 - “students”

Πρωτεύον κλειδί + ξένο κλειδί: AccountID (από πίνακα 14)

Στον συγκεκριμένο πίνακα αποθηκεύονται επιπλέον στοιχεία για λογαριασμό των χρηστών της πλατφόρμας που θα έχουν ειδικότητα φοιτητή. Συγκεκριμένα εκτός από τα ήδη αποθηκευμένα στον “πίνακα 14” θα αποθηκευτεί επίσης και το ονοματεπώνυμο του φοιτητή, η ομάδα που είναι ορισμένος, κάποιες γνώσεις από γλώσσες προγραμματισμού και frameworks (αν δήλωσε), αν ανήκει σε κάποια κατηγορία “Web”, “Desktop”, “Mobile” (σύμφωνα με τις γνώσεις του) και αν είναι ενεργός ο λογαριασμός του.

- Πίνακας 13 - “grades”

Πρωτεύον κλειδί: IDGrade (αύξων μοναδικός αριθμός)

Στον συγκεκριμένο πίνακα αποθηκεύονται πληροφορίες σχετικά με τους βαθμούς των φοιτητών που καλείται να ανεβάσει ο καθηγητής για κάθε φοιτητή. Οι πληροφορίες αυτές είναι το αναγνωριστικό για το παραδοτέο που ορίστηκε βαθμός, το αναγνωριστικό του φοιτητή που ορίστηκε ο βαθμός, ο βαθμός, κατά πόσο θα είναι ορατό ή όχι στον φοιτητή και κάποιες επιπλέον πληροφορίες ανατροφοδότησης (αν υπάρχουν).

- Πίνακας 14 - “accounts”

Πρωτεύον κλειδί: IDAccount (αύξων μοναδικός αριθμός)

Στον συγκεκριμένο πίνακα αποθηκεύονται όλες οι βασικές πληροφορίες για ένα λογαριασμό στην πλατφόρμα. Συγκεκριμένα περιέχει τον τύπο του λογαριασμού “πίνακας 7”, το ηλεκτρονικό ταχυδρομείο επικοινωνίας, το όνομα χρήστη, ο κωδικός πρόσβασης (κρυπτογραφημένος), η ημερομηνία δημιουργίας λογαριασμού, παλιές χρονολογίες που ο χρήστης μπορεί να είχε ξανά λογαριασμό (κυρίως για φοιτητές που επαναλαμβάνουν το μάθημα), αν ο λογαριασμός είναι ενεργός και πληροφορίες που αφορούν την ενημέρωση του (π.χ. αν προτιμάει να του αποστέλλονται μηνύματα στο ηλεκτρονικό ταχυδρομείο).

2.7.3. Χωρητικότητα Βάσης Δεδομένων

Στην συγκεκριμένη ενότητα θα υπολογιστεί η μέγιστη χωρητικότητας δεδομένων που μπορεί να υποστηρίξει η βάση δεδομένων μας. Λαμβάνοντας υπόψη τις οδηγίες και τις πληροφορίες που παρουσιάζει η ίδια η MySQL στο εγχειρίδιο αναφοράς³, για την συγκεκριμένη έκδοση που χρησιμοποιούμε, μπορούμε να υπολογίσουμε το χώρο που θα χρειαστεί αν οι πίνακες μας είναι πλήρως συμπληρωμένοι. Στον πιο κάτω πίνακα παρουσιάζεται η μέγιστη ανάγκη χώρου για κάθε πίνακα, ανάλογα με τα πεδία που περιέχει, όπως και συνολικά για ολόκληρη την βάση δεδομένων.

Όνομα Πίνακα	Μέγιστος αριθμός εγγραφών	Χώρος για μία εγγραφή	Συνολικός χώρος πίνακα
<i>Projecttypes</i>	16777215	25 bytes	420 MB
<i>Teamrequests</i>	16777215	11 bytes	185 MB
<i>Projects</i>	32767	250 bytes	9 MB
<i>Deliverables</i>	16777215	19 bytes	319 MB
<i>Questionaries</i>	16777215	130 bytes	2,2 GB
<i>Companies</i>	32767	30 bytes	1 MB
<i>Instructors</i>	32767	22 bytes	1 MB
<i>Typeaccounts</i>	127	15 bytes	2 KB
<i>Activatecodes</i>	32767	7 bytes	230 KB
<i>Teams</i>	32767	3 bytes	100 KB
<i>Interestedteamsinprojects</i>	16777215	5 bytes	84 MB
<i>Students</i>	32767	80 bytes	3 MB
<i>Accounts</i>	32767	132 bytes	5 MB
<i>Grades</i>	32767	115 bytes	4 MB
Συνολικός χώρος αποθήκευσης:			3,3 GB

³ <https://dev.mysql.com/doc/refman/8.0/en/storage-requirements.html>

Καταλήγουμε λοιπόν στο συμπέρασμα ότι η βάση δεδομένων μας θα χρειαστεί 3,3 GB αποθηκευτικό χώρο στην περίπτωση που ο αριθμός των εγγραφών φτάσει τον μέγιστο. Αυτή την πιθανότητα την θεωρούμε σπάνια να συμβεί σε σύντομο χρονικό διάστημα καθώς αν λάβουμε υπόψη ότι κάθε εξάμηνο βρίσκονται στο μάθημα περίπου 100 φοιτητές και ο μέγιστος αριθμός λογαριασμών είναι 32767 (μέγιστος αριθμός που μπορεί να πάρει ο τύπος δεδομένων Integer) αυτό συνεπάγεται με το ότι υπάρχει αρκετός χώρος για περίπου 327 εξάμηνα.

Κεφάλαιο 3

Περιγραφή Συστήματος

3.1.Γενική Περιγραφή Συστήματος

3.2.Περιορισμοί

3.3.Αρχιτεκτονική Συστήματος

3.4.Εργαλεία ανάπτυξης

3.4.1. Λογισμικό Ανάπτυξης

3.4.1.1. NPM

3.4.1.2. Visual Studio Code

3.4.2. Γλώσσες Προγραμματισμού και Τεχνολογίες

3.4.2.1. HTML

3.4.2.2. CSS

3.4.2.3. JavaScript

3.4.2.4. Material UI

3.4.2.5. Node JS

3.4.2.6. React JS

3.4.2.7. Express JS

3.4.2.8. React Redux

3.4.2.9. Axios

3.1.Γενική Περιγραφή Συστήματος

Η ανάγκη δημιουργίας του συγκεκριμένου συστήματος ήταν η διαχείριση του υποχρεωτικού μαθήματος Τεχνολογίας Λογισμικού - ΕΠΑ343 του Π.Κ. με μία διαδικτυακή εφαρμογή προσαρμοσμένη στις ανάγκες διάλεξης και εργαστηρίου. Είναι σχεδιασμένο για να αυτοματοποιεί λειτουργίες οι οποίες προσφέρονται στους φοιτητές, στις εταιρίες αλλά και στους καθηγητές εργαστηρίου και διάλεξης. Οι χρήστες με λογαριασμό τύπου φοιτητή θα μπορούν να δημιουργούν ομάδες για την ομαδική εργασία υλοποίησης έργου, θα έχουν

πρόσβαση στις ημερομηνίες ερωτηματολογίων και παραδοτέων και θα μπορούν να βλέπουν τα διαθέσιμα έργα για το τρέχον ακαδημαϊκό με την δυνατότητα να δείξουν ενδιαφέρον για υλοποίηση. Όσο αφορά χρήστες με τύπο λογαριασμού καθηγητή οι κύριες λειτουργίες εκτέλεσης είναι η δημιουργία υπερσυνδέσμων για παραδοτέα και ερωτηματολόγια, η ανάρτηση βαθμών φοιτητών ατομικά αλλά και σε ομάδα, δημιουργία έργων ανάθεσης και η δημιουργία μαζικών λογαριασμών φοιτητή για κάθε έτος. Τέλος, άτομα τα οποία είναι υπεύθυνα για την διαχείριση λογαριασμού τύπου εταιρίας θα έχουν την δυνατότητα δημιουργίας έργων ανάθεσης και στην συνέχεια θα μπορούν να ορίζουν την Ο.Α.Ε. Γενικά όλοι οι χρήστες θα έχουν πρόσβαση και σε κάποιες γενικές λειτουργίες διαχείρισης του λογαριασμού όπως για παράδειγμα αλλαγή στοιχείων πρόσβασης και πληροφοριών λογαριασμού, αλλαγή προτιμήσεων ενημέρωσης αλλά και δυνατότητες οι οποίες αφορούν την ασφάλεια τους σχετικά με τον Γενικό Κανονισμό για την Προστασία Δεδομένων GDPR. Πιο αναλυτικά περιγράφονται οι λειτουργίες της σελίδας στο κεφάλαιο 4.

3.2.Περιορισμοί

Απαραίτητο για οποιαδήποτε διαδικτυακή εφαρμογή είναι η ηλεκτρονική συσκευή του χρήστη να υποστηρίζει σύνδεση στο διαδίκτυο με εγκατεστημένη εφαρμογή περιήγησης ιστού (π.χ. Google Chrome, Internet Explorer, Microsoft Edge κ.λπ.). Η εμφάνιση επιπλέον σε κάθε διαφορετική διαδικτυακή εφαρμογή φυλλομετρητή ιστού καλείτε να είναι ομοιόμορφη και όπως μπορούμε να δούμε και στο κεφάλαιο 5.2 είναι κάτι το οποίο υπάρχει σε κάθε υποσελίδα.

Ίσως ο πιο σημαντικός περιορισμός ο οποίος το σύστημα μας υποστηρίζει είναι ο περιορισμός χρήσης της σελίδας από μη εξουσιοδοτημένα άτομα. Συγκεκριμένα, για την πρόσβαση στην εφαρμογή απαραίτητη είναι η χρήση αναγνωριστικού του χρήστη μαζί με τον ανάλογο κωδικό πρόσβασης, έτσι ώστε η πρόσβαση στο σύστημα να είναι ελεγχόμενη και να εμποδίζεται η χρήση της εφαρμογής από μη εξουσιοδοτημένους χρήστες. Με αυτό τον τρόπο περιορίζεται η ανεξέλεγκτη δημιουργία λογαριασμών στο σύστημα, η πρόσβαση τόσο στο σύστημα όσο και στις λειτουργίες τις οποίες προσφέρει σε πιθανόν κακόβουλους χρήστες με τυχόν εχθρικές ενέργειες για το σύστημα. Επίσης, με την ταυτοποίηση του χρήστη υποδεικνύεται το είδος του χρήστη και υπάρχει διαφοροποίηση στις λειτουργίες τις οποίες ο χρήστης είναι σε δυνατότητα να εκτελέσει δημιουργώντας ένα ευέλικτο και προσαρμοσμένο σύστημα στο κάθε ένα ξεχωριστά.

Επιπρόσθετα, η πλοήγηση στην εφαρμογή, πρέπει να είναι κατανοητή και εύκολη μεταξύ των διαφόρων λειτουργιών της, ώστε οι ενέργειες να διαδέχονται η μία την άλλη χωρίς ο χρήστης να προσπίπτει σε ανεπιθύμητες καταστάσεις του συστήματος. Στόχος είναι ο χρήστης να επιτυγχάνει το σκοπό του γρήγορα, αποτελεσματικά και χωρίς ιδιαίτερη δυσκολία.

Ακόμη, σημαντικό είναι το γεγονός, οι εντολές να εκτελούνται σε σύντομο χρονικό διάστημα έτσι ώστε να μην προκαλείται δυσφορία στο χρήστη λόγω του χρόνου καθυστέρησης. Ένας επιτρεπτός χρόνος διαχωρίζεται σε κλάσματα δευτερολέπτων για την πλοήγηση στην σελίδα ενώ δίνεται ευελιξία καθυστέρησης στην μεταφόρτωση αρχείων με τον μέγιστο επιθυμητό χρόνο να βρίσκεται στο 1 λεπτό, ανάλογα με το μέγεθος του αρχείου.

Όσο αφορά την μνήμη δεν έχουμε κάποιο μεγάλο περιορισμό, απλά η συσκευή που χρησιμοποιεί ο χρήστης να έχει εγκατεστημένο ένα φυλλομετρητή ιστού όπου απαιτεί χώρο κάποιων δεκάδων Mbytes. Επιπλέον, χρειάζεται μερικά Mbytes για να αποθηκεύει τα δεδομένα του φυλλομετρητή ιστού (π.χ. cookies, cache). Γενικά, αποτελεί ένα πολύ μικρό αριθμό απαραίτητης μνήμης και χώρου αποθήκευσης σε σχέση με παρόμοια συστήματα γεγονός το οποίο δεν προκαλεί δυσκολίες στον χρήστη.

3.3.Αρχιτεκτονική Συστήματος

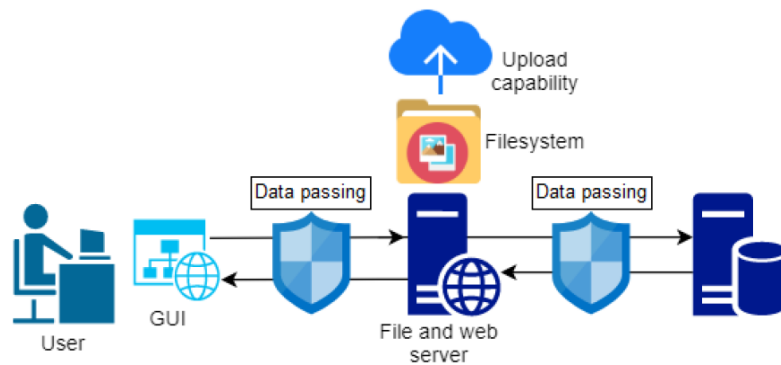
Η αρχιτεκτονική η οποία ακολουθήσαμε στην υλοποίηση του συστήματος μας είναι το μοντέλο τριών επιπέδων πελάτη και διακομιστή το οποίο φαίνεται και στην Εικόνα 2.3.1 και 2.3.2 με frontend και backend υλοποίηση. Το σύστημα στηρίζεται στην ύπαρξη γραφικού περιβάλλοντος (GUI) με το οποίο οι χρήστες της σελίδας μας θα αλληλοεπιδρούν με αυτό εκτελώντας τις λειτουργίες που βρίσκονται στις δυνατότητες κάθε τύπου λογαριασμού. Ο διακομιστής εκτελεί τα αιτήματα τα οποία λαμβάνει κάτω από την τεχνολογία RESTful Web Service.

Το Representational State Transfer είναι ένα σύνολο σχεδιαστικών αρχών οποιασδήποτε διαδικτυακής υπηρεσίας το οποίο επικεντρώνεται στην ασφάλεια και διαμοιρασμό των πόρων του συστήματος (π.χ. δεδομένα). Μέσω του πρωτόκολλου HTTP μεταβάλλεται η κατάσταση των πόρων του συστήματος η οποία περιγράφεται κάθε φορά στα αιτήματα από διάφορους clients [24]. Συγκεκριμένα, υπάρχει η δυνατότητα λόγω των αρκετών end-points στο σύστημα να διατίθενται αρκετά αιτήματα ανάλογα με τις υπηρεσίες και ανάγκες του πελάτη. Στην συγκεκριμένη περίπτωση ο διακομιστής στην ενδιάμεση φάση της αρχιτεκτονικής εκτελεί τα αιτήματα στο RWS, στη συνέχεια αυτό με την σειρά του επικοινωνεί μαζί με άλλα στοιχεία της Β.Δ. και τέλος λαμβάνει απάντηση σε μορφή JSON.

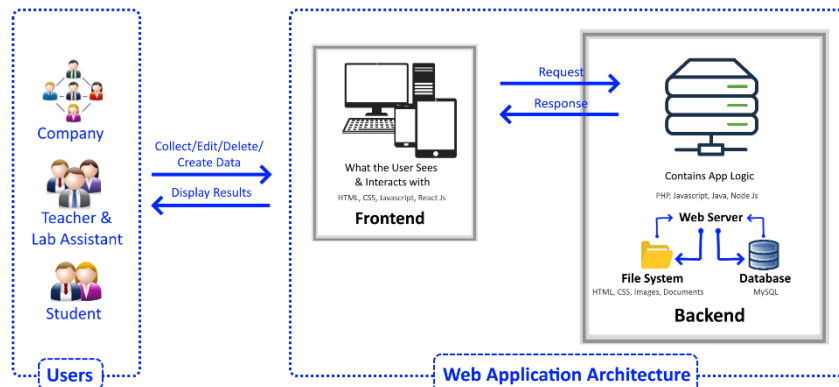
Αρχικά για τα πρώτα στάδια της υλοποίησης η σύνδεση πελάτη – εξυπηρετητή ήταν τοπική (LAN) με την χρήση ενός οποιουδήποτε φυλλομετρητή και της εφαρμογής WAMP. Στην συνέχεια η υλοποίηση μεταφέρθηκε σε ένα Virtual Machine για να δίνεται η δυνατότητα σε μεγαλύτερη μερίδα ατόμων να επισκέπτονται την πλατφόρμα (WAN), με την προϋπόθεση ότι υπάρχει σύνδεση με VPN στο τμήμα πληροφορικής του Π.Κ.. Ο εξυπηρετητής ο οποίος

χρησιμοποιείται και στις 2 περιπτώσεις είναι ο Apache μαζί με την PHP και την Β.Δ. MySQL η οποία εμφανίζεται με την εφαρμογή της phpMyAdmin.

Για την δυνατότητα της πλατφόρμας να προσφέρει δυνατότητες επικοινωνίας χρήστη με το σύστημα (frontend), μεταφόρτωση αρχείων στον διακομιστή (backend) αλλά και σύνδεση με την Β.Δ. χρειαστήκαμε την ύπαρξη 3 ports. Συγκεκριμένα το port 3000 αντιστοιχεί στην επικοινωνία του GUI, το port 5000 χρειάζεται για σύνδεση με τον διακομιστή και το port 3306 για την δημιουργία Β.Δ.. Η δυνατότητα να συνυπάρχουν τα 3 αυτά port είναι η ρύθμιση localhost η οποία υπάρχει στα αρχεία configuration των 3^{ων} αυτών port στο VM καθιστώντας εύκολη την επικοινωνία μεταξύ τους στην διεύθυνση 10.16.33.144:3000/.



Εικόνα 2.3. 1: Γενική αρχιτεκτονική συστήματος για επικοινωνία σύνδεσης HTTP



Εικόνα 2.3. 2: Αρχιτεκτονική συστήματος frontend και backend

3.4.Επιλογή Κατάλληλων Εργαλείων και Τεχνολογιών

Αρχικά λαμβάνοντας υπόψη τις απαιτήσεις της διαδικτυακής εφαρμογής αλλά και το γεγονός ότι η πλατφόρμα θα λειτουργούσε σε διάφορα λειτουργικά συστήματα (ανάλογα με την συσκευή πρόσβασης του χρήστη) έγινε μία μικρή έρευνα στις τεχνολογίες που υπάρχουν και θα μπορούσαν να καλύψουν τις ανάγκες αυτές. Το σύστημα θα είναι προσβάσιμο από οποιαδήποτε ηλεκτρονική συσκευή με πρόσβαση στο διαδίκτυο όπως υπολογιστές και κινητές συσκευές. Σημαντικό είναι το γεγονός ότι δεν υπάρχει περιορισμός στο λειτουργικό σύστημα χρήσης (Windows, Linux, macOS, Android) αλλά ούτε και στον φυλλομετρητή ιστού (Google

Chrome, Internet Explorer, Mozilla Firefox κτλ.) που θα χρησιμοποιεί ο χρήστης άρα απαραίτητη προϋπόθεση είναι η τεχνολογίες ανάπτυξης να μπορούν να προσαρμοστούν ανάλογα.

Με βάση τα παραπάνω και μετά από έρευνα, καταλήξαμε σε τρεις διακριτές κατηγορίες ανάπτυξης εφαρμογών σε κινητά τηλέφωνα και φορητές συσκευές. Στις πιο κάτω κατηγορίες χρησιμοποιούνται διαφορετικές γλώσσες προγραμματισμού και εργαλεία το κάθε ένα με τα δικά του πλεονεκτήματα και μειονεκτήματα όπως αναλύονται και πιο κάτω [2].

1) Native app development:

Είναι η ανάπτυξη λογισμικού το οποίο σχεδιάζεται για να λειτουργεί σε συγκεκριμένες συνθήκες και σε μία συγκεκριμένη πλατφόρμα και λειτουργικό σύστημα. Αυτός ο τρόπος ανάπτυξης απαιτεί η ανάπτυξη του λογισμικού να γίνεται στη γλώσσα που χρησιμοποιεί το λογισμικό και η συσκευή. Εκμεταλλεύεται την ευκολία που υπάρχει να χειριστεί άμεσα οποιοδήποτε υλικό διαθέτει η συσκευή όπως για παράδειγμα GPS, κάμερα, μικρόφωνο, offline access κτλ. Ως αποτέλεσμα επιτρέπει στους χρήστες να έχουν μία καλύτερη εμπειρία χρήσης στην μέγιστη δυνατή απόδοση.

2) Hybrid app development:

Είναι η ανάπτυξη λογισμικού το οποίο περιέχει τόσο native όσο και διαδικτυακή υλοποίηση ταυτόχρονα. Η εφαρμογή περιέχει συνήθως κώδικα σε τεχνολογίες διαδικτύου (HTML, CSS, JavaScript) που συνήθως χρησιμοποιεί την γλώσσα προγραμματισμού που χρησιμοποιείται και στο συγκεκριμένο χώρο που φιλοξενεί το σύστημα (π.χ. apache). Με αυτό τον τρόπο υπάρχει η δυνατότητα πρόσβασης σε όλες τις λειτουργίες που δίνει το λειτουργικό σύστημα όπως την κάμερα ή το Bluetooth της συσκευής, μέσω της εγκατάστασης πρόσθετων βιβλιοθηκών. Προσφέρει πολύ καλή εμπειρία χρήσης και υψηλή απόδοση χρηστών όπως και οι native εφαρμογές. Ωστόσο, θεωρείται πρόκληση να υπάρχει ταυτόχρονα και πολύ καλή εμπειρία χρήσης αλλά και εύκολη πλοήγηση από οπτικής άποψης.

3) Cross Platform app development:

Είναι η ανάπτυξη λογισμικού η οποία γίνεται με μία συγκεκριμένη γλώσσα προγραμματισμού και στην συνέχεια ο κώδικας μεταφράζεται ανάλογα με την υποστηριζόμενη γλώσσα της συσκευής. Αυτό δίνει την δυνατότητα στο λογισμικό, χωρίς επιπλέον κόπο και συγγραφή κώδικα από τον προγραμματιστή, να μπορεί να ανταποκριθεί με την ίδια εφαρμογή σε κάθε διαφορετική συσκευή και οποιοδήποτε λειτουργικό σύστημα. που οποίο περιέχει τόσο native όσο και διαδικτυακή υλοποίηση ταυτόχρονα. Προσφέρει πολύ καλή εμπειρία χρήσης όπως τα άλλα δύο με την διαφορά ότι η απόδοση εξαρτάται από το framework το οποίο θα χρησιμοποιηθεί.

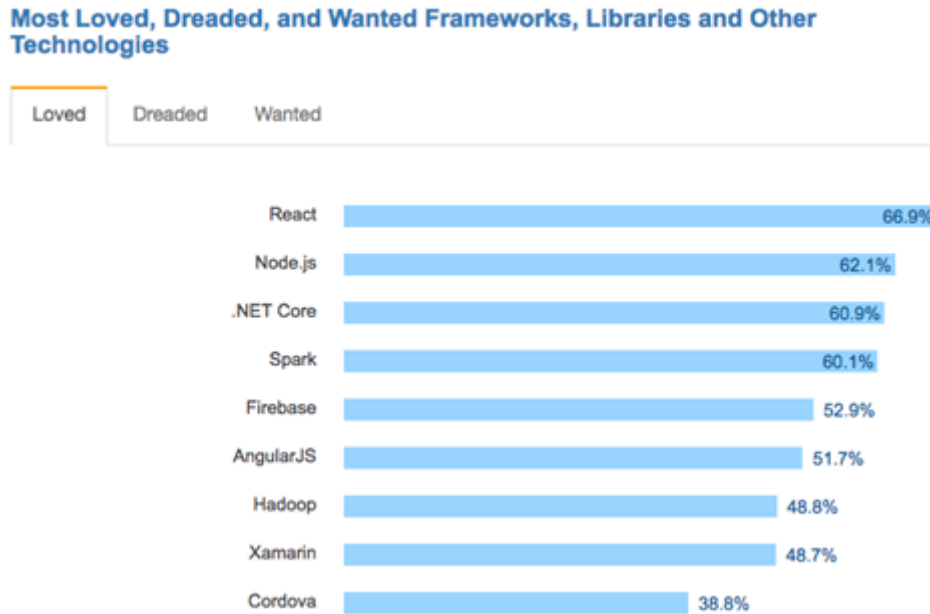
	Native	Hybrid	Cross-platform
<i>Debugging</i>	Χρησιμοποιεί native debugging tools	Χρησιμοποιεί native και web debugging tools	Ανάλογα με το framework
<i>Codebase</i>	Ξεχωριστός κώδικας για κάθε διαφορετική πλατφόρμα	Ένας κώδικας που μεταφράζεται αυτόματα ανάλογα με την συσκευή	Ένας κώδικας που μεταφράζεται αυτόματα ανάλογα με την συσκευή
<i>Επαναχρησιμοποίηση κώδικα</i>	Ο κώδικας είναι συγκεκριμένος για την κάθε πλατφόρμα	Κοινός κώδικας για όλες τις πλατφόρμες, ωστόσο απαιτείται ενθυλάκωση σε εφαρμογή που είναι γραμμένη στην γλώσσα του κάθε συστήματος	Κοινός κώδικας σε όλες τις πλατφόρμες
<i>Επίδοση</i>	Υψηλή	Χαμηλή	Μέτρια
<i>Γλώσσα ανάπτυξης</i>	Swift, Objective C για iOS και Kotlin, Java για Android	HTML, CSS, JavaScript	JavaScript, C#, Dart
<i>Κόστος</i>	Ψηλό	Χαμηλό	Χαμηλό
<i>Χρόνος</i>	Ψηλό	Χαμηλό	Χαμηλό
<i>Προκλήσεις</i>	Απαιτείται αρκετός χρόνος η ίδια εφαρμογή αναπτύσσεται σε διαφορετικές γλώσσες	Γρήγορος ρυθμός ανάπτυξης	Γρήγορος ρυθμός ανάπτυξης εφόσον υπάρχουν προγραμματιστές με κάποια εμπειρία

Πίνακας 2.4. 1: Σύγκριση Native vs Hybrid vs Cross-Platform

Λαμβάνοντας υπόψη τον Πίνακα 2.4.1 αλλά και βάση των απαιτήσεων της αρχικής ιδέας καταλήγουμε στο συμπέρασμα ότι πλέον καταλληλότερη θα είναι η χρήση cross-platform υλοποίησης. Συγκεκριμένα η υλοποίηση θα γίνει με την χρήση του framework React JS καθώς είναι βασισμένο στην JavaScript, μία γλώσσα η οποία υπάρχει ήδη προηγούμενη γνώση και κυρίως θεωρείται μία από τις πλέον ανεπτυγμένες και πολυχρησιμοποιούμενες προγραμματιστικές γλώσσες της εποχής.

Η React JS θεωρείται μία πολύ αποτελεσματική και ευέλικτη βιβλιοθήκη της JavaScript για την δημιουργία διεπαφών χρήστη και μας βοηθάει πολύ γρήγορα και απλά να με την χρήση

επεκτάσεων (components) να φτιάξουμε διαδικτυακές εφαρμογές. Είναι μία σχετικά καινούρια τεχνολογία που αρχικά χρησιμοποιήθηκε ως εργαλείο για κάποιες εφαρμογές του Facebook,



Εικόνα 2.4. 1: Αποτελέσματα έρευνας Stackoverflow 2020

στην συνέχεια αναπτύχθηκε περαιτέρω και σήμερα βρίσκεται μεταξύ των πιο αγαπημένων γλωσσών προγραμματισμού μέσα από μία έρευνα [5] του Stack Overflow Developer to 2020 με ποσοστό 66.9%.

Δεν θεωρείται τυχαίο το γεγονός ότι μέχρι σήμερα την χρησιμοποιούν περίπου 100,000 ιστοσελίδες καθώς προσφέρει αρκετά και σημαντικά πλεονεκτήματα σε σχέση με αντίστοιχα frameworks. Αξίζει, επιπλέον να αναφερθεί ότι προσφέρει υψηλή ποιότητα κώδικα με λιγότερες προγραμματιστικές ώρες λόγω του ότι η επαναχρησιμοποιεί κομμάτια έτοιμου κώδικα και λειτουργιών σε διάφορα σημεία της εφαρμογής έτσι ώστε να αυξάνεται η παραγωγικότητα. Επιπρόσθετα, υπάρχει αρκετή ευελιξία στην δομή των προγραμμάτων που δημιουργεί παράλληλα και ένα πολύ εύκολο τρόπο συντήρησης της εφαρμογής για μελλοντικά στάδια. Γίνεται, επομένως εύκολα αντιληπτή η κλίμακα απόδοσης της εφαρμογής να κυμαίνεται σε πολύ ψηλά ποσοστά. Χρειάζεται επίσης να σημειωθεί ότι διαθέτει ένα τεράστιο αριθμό από βιβλιοθήκες ανοικτού κώδικα (open-source) για backend (Node JS) αλλά και εργαλεία (Redux) που είναι απαραίτητα για να διευκολύνουν τη ζωή ενός προγραμματιστή. Συμπερασματικά λαμβάνοντας υπόψη όλα τα πιο πάνω η πλέον κατάλληλη προγραμματιστική λύση είναι η React JS ώστε να μπορούμε να εκμεταλλευτούμε τις λειτουργίες της.

3.5.Εργαλεία ανάπτυξης

3.5.1. Λογισμικό Ανάπτυξης

3.5.1.1. NPM

Το NPM (Node Package Manager) είναι ένας διαχειριστής πρόσθετων για τη γλώσσα προγραμματισμού JavaScript. Υποστηρίζεται κυρίως από δημιουργία προγραμματιστών βασισμένοι σε ανοικτού κώδικα και συνηθίζουν να κοινοποιούν αλλά και να χρησιμοποιούν βιβλιοθήκες ή κοινά προγράμματα ανοιχτού κώδικα που παρέχουν πρόσθετες λειτουργίες για το υπό ανάπτυξη έργο. Αποτελείται από τρία ξεχωριστά συστατικά την ιστοσελίδα, από έναν πελάτη στη γραμμή εντολών που αλληλεπιδρά με το απομακρυσμένο ευρετήριο (CLI) ή από το αρχείο δεδομένων. Όταν χρησιμοποιείτε αυτήν τη βιβλιοθήκη, θα δημιουργηθεί ένα αρχείο package.json που περιέχει τα ονόματα και τις εκδόσεις όλων των εγκατεστημένων πακέτων. Επομένως, μπορούν να αναβαθμιστούν αυτόματα χωρίς αυτά να χρειάζονται ειδική μεταχείριση από τον προγραμματιστή. Επιπλέον, χρησιμοποιώντας το συγκεκριμένο αρχείο πολύ εύκολα με την εκτέλεση μίας μόνο εντολής να εγκατασταθούν όλα τα απαραίτητα αρχεία και βιβλιοθήκες για ένα πρόγραμμα.



Εικόνα 2.5.1. 1: Λογότυπο NPM

3.5.1.2. Visual Studio Code

Το Visual Studio Code είναι ένας δωρεάν επεξεργαστής κώδικα ο οποίος αναπτύχθηκε από την Microsoft και υποστηρίζει μία μεγάλη ποικιλία από προγραμματιστικές γλώσσες τόσο για δημιουργία του προγραμματιστικού υπόβαθρου αλλά και για δημιουργία της γραφικής διαπροσωπίας. Μπορεί να εκτελεστεί σε Windows, Linux, macOS και είναι ένα εργαλείο για την ανάπτυξη εφαρμογών ιστού και cloud. Είναι κατάλληλο για τη σύνταξη κώδικα σε πολλές γλώσσες προγραμματισμού, όπως: Python, JavaScript, Php, C κ.λπ. Οι δυνατότητες του είναι πολύ βοηθητικές καθώς υποστηρίζει εντοπισμό σφαλμάτων, ενσωματωμένο έλεγχο Git και Github, εύκολη αναδιαμόρφωση κώδικα αλλά και έξυπνη ολοκλήρωση κώδικα για πιο γρήγορο και εύκολο προγραμματισμό. Υποστηρίζει επίσης τη δημιουργία Web Applications, class designer και επιπλέον παρέχει ένα Database Schema Designer. Υπάρχει η δυνατότητα εγκατάστασης συγκεκριμένων Plug Ins τα οποία μπορούν να αυξήσουν την αποδοτικότητα του προγραμματιστή σε κάθε επίπεδο. Υπάρχει μία πλήρης εργαλειοθήκη η οποία περιέχει cobmo boxes, radio buttons, buttons και text boxes κτλ. στην οποία μπορούμε να βρούμε ποικίλα εργαλεία για να χρησιμοποιήσουμε και να δημιουργήσουμε την διαπροσωπία του συστήματος πολύ πιο εύκολα. Το συγκεκριμένο εργαλείο επιτρέπει επίσης τη χρήση πολλών λειτουργιών οι οποίες είναι παρόμοιες με εκείνες του Λειτουργικού Συστήματος (στην συγκεκριμένη περίπτωση Windows) που είναι πολύ οικίες στους χρήστες του συστήματος [7].



Εικόνα 2.5.1. 2: Λογότυπο Visual Studio Code

3.5.2. Γλώσσες Προγραμματισμού και Τεχνολογίες

3.5.2.1. HTML

Η γλώσσα σήμανσης υπερκειμένου HTML όπως φαίνεται και από τα αρχικά της Hyper Text Markup Language είναι η κύρια γλώσσα σήμανσης για πολλές ιστοσελίδες και αποτελείται συνήθως από βασικά δομικά στοιχεία σε μορφή ετικετών (tags) για να ορίσουν την δομή της. Τα στοιχεία αυτά περικλείονται ανάμεσα στο σύμβολο του μεγαλύτερου ‘<’ και του μικρότερου ‘>’ και περιέχουν το στοιχείο μέσα όπως για παράδειγμα
. Συγκεκριμένα ο φυλλομετρητής ιστού είναι υπεύθυνος να διαβάσει τα έγγραφα HTML, να δημιουργήσει την δομή κάθε σελίδας και να παρουσιάσει το περιεχόμενο της στην θέση των ετικετών. Επιτρέπει την ενσωμάτωση εικόνων, βίντεο και άλλων αντικειμένων στη σελίδα που μπορούν να χρησιμοποιηθούν για την εμφάνιση διαδραστικών φορμών. Γενικά η μορφή μίας σελίδας που είναι βασισμένη μόνο σε HTML θεωρείται στατική αλλά με την προσθήκη τεχνολογιών που περιγράφονται πιο κάτω (π.χ. CSS, JavaScript) υπάρχει η δυνατότητα να μετατραπεί σε δυναμική [6].



Εικόνα 2.5.2. 1:
Λογότυπο HTML 5

3.5.2.2. CSS

Το CSS είναι το ακρωνύμιο της φράσης “Cascading Style Sheets” και με την τεχνολογία αυτή καθορίζετε η εμφάνιση κάθε σελίδας στον επισκέπτη. Αυτό επιτυγχάνεται με την μορφοποίηση των διαφόρων στοιχείων της HTML μειώνοντας τους σχεδιαστές/προγραμματιστές από κόπο και χρόνο καθώς μειώνεται σημαντικά ο όγκος εργασίας λόγω της δυνατότητας επαναχρησιμοποίησης μορφοποιήσεων σε διάφορα σημεία. Ετικέτες στην HTML που προσφέρουν παρόμοια αποτελέσματα με το CSS όπως η και τα χρωματικά στοιχεία προστέθηκαν στην έκδοση HTML 3.2. Το CSS είναι ιδιαίτερος χρήσιμο λόγο του ότι μία σελίδα πρέπει να ακολουθεί κοινή γραμματοσειρά, χρωματισμούς και αναλογία στο μέγεθος κειμένου και περιεχομένου σε κάθε υποσελίδα. Από τα πιο πάνω φαίνεται είναι φανερό ότι το χωριστό CSS αρχείο που υπάρχει από την έκδοση HTML 4.0 μπορούν να αλλάξουν την μορφοποίηση ολόκληρης της διαδικτυακής εφαρμογής ταυτόχρονα και όχι μόνο σε ένα αρχείο [8].



Εικόνα 2.5.2. 2:
Λογότυπο CSS

3.5.2.3. JavaScript

Η JavaScript (JS) αρχικά αποτελούσε μέρος της υλοποίησης των φυλλομετρητών ιστού για να χειρίζεται σενάρια που εκτελούνταν από την πλευρά του πελάτη προς τον διακομιστή (client-side scripts), να μπορούν να ανταλλάσσουν δεδομένα ασύγχρονα και το περιεχόμενο από στατικό (με την ύπαρξη μόνο HTML) να μετατραπεί σε δυναμικό ώστε να αλλάζει το περιεχόμενο ανάλογα με την κατάσταση. Θεωρείται μία διερμηνευμένη γλώσσα προγραμματισμού και σεναρίων που βασίζεται κυρίως σε πρωτότυπα (prototype-based). Η σύνταξη της είναι σαφώς επηρεασμένη από αυτή της C αλλά όσο αφορά σε θέμα ονομάτων και συμβάσεων ονοματοδοσίας μοιάζει πολύ στην Java και αυτό είναι το μόνο κοινό τους αφού αντιθέτως με την κοινή γνώμη που επηρεάζεται από την ομοιότητα των ονομάτων των δύο

γλωσσών δεν σχετίζονται καθόλου σημασιολογικά. Υποστηρίζει αντικειμενοστραφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού με για βασικές αρχές σχεδιασμού να είναι παρόμοιες από τις προγραμματιστικές γλώσσες Self και Scheme. Η JavaScript δεν περιορίζει την χρήση της μόνο σε διαδικτυακές εφαρμογές αλλά και σε έγγραφα PDF, εξειδικευμένους φυλλομετρητές (site-specific browsers), μικρές εφαρμογές επιφάνειας εργασίας (desktop widgets) αλλά και νεότερες εικονικές μηχανές (VM) και πλαίσια ανάπτυξης όπως το Node.js (χρησιμοποιείται και περιγράφεται στην ενότητα 2.5.2.5) [9].



Εικόνα 2.5.2. 3:
Λογότυπο JavaScript

3.5.2.4. Material UI

Το Material UI (User Interface) είναι ένα σύνολο από σχεδιαστικούς κανόνες το οποίο παρουσιάστηκε στις 25 Ιουνίου το 2014 από την Google σε ένα συνέδριο της. Είναι μια βιβλιοθήκη συστατικών στοιχείων για το React JS με ισχυρά στοιχεία προσφέροντας μία όμορφη εφαρμογή με τα έτοιμα και προσαρμοσμένα συστατικά στοιχεία τα οποία πολύ εύκολα μπορούν να δεχτούν επεξεργασία. Γενικά είναι μία ανοικτού κώδικα βιβλιοθήκη με πολλά αυτοματοποιημένα συστατικά που προσφέρουν μορφοποίηση τόσο στον χρωματικό αλλά και στην δομή της σελίδας με ευπαρουσίαστο τρόπο. Δεν είναι τυχαίο το γεγονός ότι χρησιμοποιείται από εκατοντάδες χιλιάδες προγραμματιστές και κάποιες πολύ γνωστές οργανισμοί, όπως NETFLIX, Amazon, Unity, Spotify κτλ. , στηρίζουν την εφαρμογή τους στον σχεδιασμό με Material UI.



Εικόνα 2.5.2. 4:
Λογότυπο Material UI

3.5.2.5. Node JS

Το Node JS είναι μία πλατφόρμα ανάπτυξης λογισμικού, τις περισσότερες φορές για τους διακομιστές – γι' αυτό είναι κυρίως back-end, βασισμένο στο περιβάλλον JavaScript. Στόχος του Node JS είναι να βοηθάει τον προγραμματιστή να δημιουργήσει πιο εύκολα κλιμακωτές διαδικτυακές εφαρμογές, στηρίζεται στο μοντέλο ασύγχρονης επικοινωνίας εισόδου/εξόδου, κάτι που μέχρι σήμερα αποτελεί πρόκληση για τα περισσότερα σύγχρονα περιβάλλοντα ανάπτυξης εφαρμογών δικτύων. Η πρώτη έκδοση του παρουσιάστηκε το Μάιο του 2009 από τον Ryan Lienhart Dahl, χρησιμοποιεί άδεια λογισμικού MIT και είναι γραμμένη σε C++ και JavaScript. Όπως φαίνεται και στην Εικόνα 2.4.1 το Node JS είναι μία από τις αγαπημένες και πολυχρησιμοποιημένες τεχνολογίες από διάφορους προγραμματιστές αφού υπάρχουν πολλά και σημαντικά κίνητρα να την ακολουθήσουν. Αρχικά, είναι πολύ εύκολη στην εκμάθηση καθώς στηρίζεται στην JavaScript και με απλές προγραμματιστικές γνώσεις απλοποιεί τα πράγματα πολύ. Επιπλέον, το γεγονός ότι είναι ανοικτού κώδικα ενθαρρύνει την υποστήριξη και τη συμβολή διαφόρων προγραμματιστών με στόχο τη βελτίωση του αλλά και την δημιουργία περισσότερων συστατικών και πόρων για επαναχρησιμοποίηση (υπάρχουν περισσότερα από 650.000

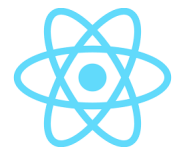


Εικόνα 2.5.2. 5:
Λογότυπο Node JS

δωρεάν πακέτα κώδικα). Σημαντικό είναι επίσης να αναφερθεί ότι διασφαλίζεται η υψηλή ταχύτητα και απόδοση της εφαρμογής ειδικά για εφαρμογές πραγματικού χρόνου (real-time), όπως η πλατφόρμα μας.

3.5.2.6. React JS

Η React JS είναι ανοικτού κώδικα front-end JavaScript βιβλιοθήκη η οποία χρησιμοποιείται κυρίως για δημιουργία UI και UI συστατικά (components). Παρουσιάστηκε πρώτη φορά τον Μάιο του 2013 από τον Jordan Walke, είναι γραμμένη σε JavaScript και έχει άδεια λογισμικού MIT. Κύριος στόχος της React είναι η ανάπτυξη μίας σελίδας ή εφαρμογής κινητού τηλεφώνου και η διαχείριση της κατάστασης/απόδοσης της στο DOM. Φυσικά υπάρχει η δυνατότητα με την προσθήκη επιπλέον βιβλιοθηκών και δρομολόγηση (routing) να δημιουργηθεί μια εφαρμογή μεγαλύτερου όγκου και πολυπλοκότητας. Η React καθιστά ανώδυνη τη δημιουργία διαδραστικών διεπαφών χρήστη αφού μπορεί εύκολα να ενημερώσει αποτελεσματικά τα σωστά στοιχεία όταν αλλάζουν τα δεδομένα στο πρόγραμμα. Επιπρόσθετα, ως δηλωτική γλώσσα προσφέρει στον χρήστη την δυνατότητα να εντοπίζει ευκολότερα τα σφάλματα και κάνει τον κώδικα πιο προβλέψιμο. Επιπλέον, είναι βασισμένη σε συστατικά (component-based) δεδομένου ότι η λογική των στοιχείων είναι βασισμένη σε JavaScript, υπάρχει η δυνατότητα πολύ εύκολα να μεταφερθούν και να επαναχρησιμοποιηθούν ενθυλακωμένα στοιχεία εντός και εκτός DOM [12].



Εικόνα 2.5.2. 6:
Λογότυπο React JS

3.5.2.7. Express JS

Η Express JS είναι μία back-end εφαρμογή διαδικτύου που χρησιμοποιείται ειδικά για το Node.js, κυκλοφορεί σαν δωρεάν και ανοικτού κώδικα λογισμικό κάτω από την MIT άδεια λογισμικού. Σχεδιάστηκε για εφαρμογές διαδικτύου και API's από τον TJ Holowaychuk, τον StrongLoop και άλλους τον Νοέμβριο το 2010 και είναι γραμμένη στην JavaScript. Δεδομένου ότι το Express.js απαιτεί μόνο javascript, αλλά και το ότι είναι ένα πλαίσιο του Node.js (το μεγαλύτερο μέρος του κώδικα είναι ήδη γραμμένο) γίνεται ευκολότερο για προγραμματιστές να δημιουργούν εφαρμογές ιστού και API χωρίς πολύ κόπο. Το Express.js προσφέρει απλότητα, ευελιξία, αποδοτικότητα, μινιμαλισμό και επεκτασιμότητα στους προγραμματιστές. Έχει επίσης το πλεονέκτημα της ισχυρής απόδοσης καθώς είναι ένα πλαίσιο του Node.js. Το Node.js μεταφέρει όλες τις εκτελέσεις πολύ γρήγορα με τη βοήθεια του Event Loop που αποφεύγει κάθε είδους αναποτελεσματικότητα. Η ισχυρή απόδοση του Node.js αλλά και η ευκολία κωδικοποίησης χρησιμοποιώντας το Express.js είναι τα πιο δημοφιλή χαρακτηριστικά που αγαπούν οι προγραμματιστές εφαρμογών ιστού.



Εικόνα 2.5.2. 7:
Λογότυπο Express JS

3.5.2.8. React Redux

Η React Redux είναι ανοικτού κώδικα front-end JavaScript βιβλιοθήκη η οποία χρησιμοποιείται κυρίως για να διαχειριστή και να αποθηκεύσει την “κατάσταση” (state) των συστατικών (components) του GUI. Για παράδειγμα να γνωρίζει σε κάθε υποσελίδα της εφαρμογής την γλώσσα που ο χρήστης επιθυμεί να εμφανίζεται το περιεχόμενο. Παρουσιάστηκε πρώτη φορά τον Δεκέμβριο του 2019 από τον Dan Abramov και τον Andrew Clark, είναι γραμμένη σε JavaScript για cross-platform εφαρμογές και έχει άδεια λογισμικού MIT. Ενώ χρησιμοποιείται ως εργαλείο διαχείρισης κατάστασης με το React, μπορεί να το χρησιμοποιηθεί με οποιοδήποτε άλλο πλαίσιο JavaScript ή βιβλιοθήκη. Είναι ελαφρύ στα 2KB (συμπεριλαμβανομένων των εξαρτήσεων), οπότε δεν δημιουργεί κάποια επιπλέον καθυστέρηση στην εκτέλεση του προγράμματος. Με την React Redux, η κατάσταση της εφαρμογής διατηρείται σε μία μεταβλητή γνωστή και ως “store” και κάθε στοιχείο μπορεί να έχει πρόσβαση σε οποιαδήποτε κατάσταση αντικειμένου με αυτή την μεταβλητή.



Εικόνα 2.5.2. 8: Λογότυπο Redux

3.5.2.9. Axios

Το Axios είναι με JavaScript βιβλιοθήκη η οποία μπορεί να χρησιμοποιηθεί για την εκτέλεση αιτημάτων HTTP έτσι ώστε να λειτουργεί τόσο στις πλατφόρμες φυλλομετρητών όσο και στις πλατφόρμες Node.js. Υποστηρίζεται από όλα τα σύγχρονα προγράμματα περιήγησης, συμπεριλαμβανομένης της υποστήριξης για IE8 και νεότερη έκδοση. Λειτουργεί σαν promised-based και γι’ αυτό υπάρχει η δυνατότητα να γράφονται τόσο ασύγχρονα όσο και σύγχρονα αιτήματα μαζί με XHR. Κάποια από τα πιο σημαντικά του πλεονεκτήματα είναι ότι υποστηρίζει παλαιότερα προγράμματα περιήγησης, μπορεί να ακυρώσει οποιοδήποτε αίτημα ανά πάσα στιγμή, μπορεί να ορίσει χρονικό περιθώριο για να αποδεκτή μια απόκριση από ένα διακομιστή, υπάρχει η ενσωματωμένη προστασία CSRF και τέλος μετασχηματίζει τα δεδομένα αυτόματα σε μορφή JSON [14].



Εικόνα 2.5.2. 9: Λογότυπο Axios

Κεφάλαιο 4

Λειτουργίες Συστήματος

4.1. Κοινές λειτουργίες συστήματος

- 4.1.1. Σύνδεση
- 4.1.2. Λειτουργία επανάκτησης κωδικού πρόσβασης
- 4.1.3. Λειτουργία Αποσύνδεσης
- 4.1.4. Λειτουργία αλλαγής γλώσσας εμφάνισης
- 4.1.5. Λειτουργία διαχείρισης ειδοποιήσεων λογαριασμών
- 4.1.6. Λειτουργία αλλαγής κωδικού πρόσβασης
- 4.1.7. Λειτουργία προβολής και αποθήκευσης στοιχείων GDPR
- 4.1.8. Λειτουργία αλλαγής στοιχεία προφίλ

4.2. Διεπαφή Οργανισμού

- 4.2.1. Δημιουργία Λογαριασμού
- 4.2.2. Έργα Ανάθεσης
 - 4.2.2.1. Περιορισμένη εμφάνιση αποτελεσμάτων
 - 4.2.2.2. Δημιουργία καινούριου Ε.Α.
 - 4.2.2.3. Επεξεργασία στοιχείων Ε.Α.
 - 4.2.2.4. Προβολή ενδιαφερόμενων ομάδων για συγκεκριμένο Ε.Α.
 - 4.2.2.5. Διαγραφή Ε.Α.
- 4.2.3. Προφίλ
 - 4.2.3.1. Επεξεργασία στοιχείων προφίλ
 - 4.2.3.2. Μεταφόρτωση εικόνας προφίλ - λογοτύπου

4.3. Διεπαφή Φοιτητή

- 4.3.1. Κεντρική Σελίδα
- 4.3.2. Παραδοτέα
- 4.3.3. Ερωτηματολόγια
- 4.3.4. Έργα ανάθεσης
 - 4.3.4.1. Λειτουργία ένδειξης ενδιαφέροντος σε Ε.Α
- 4.3.5. Βαθμοί
- 4.3.6. Ομάδες Φοιτητών
 - 4.3.6.1. Προβολή ομάδας και ενέργειες ομάδας
 - 4.3.6.2. Αποστολή αιτήματος σε ομάδες για συνένωση
 - 4.3.6.3. Αποστολή αιτήματος σε συγκεκριμένο άτομο για συνένωση

- 4.3.6.4. Αιτήματα που έχουν ήδη ληφθεί Ομάδες Φοιτητών
- 4.3.7. Προφίλ
 - 4.3.7.1. Προσθήκη και αφαίρεση γνώσεων
- 4.4. Διεπαφή Καθηγητή
 - 4.4.1. Κεντρική Σελίδα
 - 4.4.2. Παραδοτέα
 - 4.4.2.1. Δημιουργία καινούριου παραδοτέου
 - 4.4.2.2. Ενέργειες διαγραφής και περιορισμού πρόσβασης
 - 4.4.3. Ερωτηματολόγια
 - 4.4.3.1. Δημιουργία καινούριου παραδοτέου
 - 4.4.3.2. Ενέργειες διαγραφής και περιορισμού πρόσβασης
 - 4.4.4. Έργα Ανάθεσης
 - 4.4.4.1. Περιορισμένη εμφάνιση αποτελεσμάτων
 - 4.4.4.2. Δημιουργία καινούριου Ε.Α.
 - 4.4.4.3. Επεξεργασία στοιχείων Ε.Α.
 - 4.4.4.4. Προβολή ενδιαφερόμενων ομάδων για συγκεκριμένο Ε.Α.
 - 4.4.4.5. Διαγραφή Ε.Α.
 - 4.4.5. Βαθμοί
 - 4.4.5.1. Ανάρτηση βαθμών για συγκεκριμένο φοιτητή
 - 4.4.5.2. Ανάρτηση βαθμών για ομάδα ατόμων
 - 4.4.5.3. Προβολή βαθμών τρέχοντος ακαδημαϊκού έτους
 - 4.4.6. Ομάδες Φοιτητών
 - 4.4.6.1. Δημιουργία ομάδων φοιτητών
 - 4.4.6.2. Επεξεργασία ομάδων φοιτητών
 - 4.4.6.3. Εμφάνιση φοιτητών χωρίς ομάδα
 - 4.4.6.4. Διαγραφή / Επικύρωση ομάδας φοιτητών
 - 4.4.7. Δημιουργία λογαριασμών τύπου φοιτητή
 - 4.4.7.1. Μεταφόρτωση αρχείου CSV
 - 4.4.7.2. Προβολή φοιτητών
- 4.5. Ενδεικτικά σημεία υλοποίησης
 - 4.5.1. Εισαγωγή συστατικών (components), εικονιδίων και λειτουργιών
 - 4.5.2. Εισαγωγή στοιχείων και τιμών από το state Redux
 - 4.5.2.1. Περιγραφή υλοποίησης 'selectedlan'
 - 4.5.2.2. Περιγραφή υλοποίησης 'loginsession'
 - 4.5.3. Μετάφραση σελίδας
 - 4.5.4. Εμφάνιση ενημερωτικών μηνυμάτων

- 4.5.5. Δομή σελίδας προφίλ
- 4.5.6. Εμφάνιση κατάλληλου πίνακα αλλαγής στοιχείων χρήστη
- 4.5.7. Εμφάνιση δεύτερης λειτουργίας για οργανισμό και φοιτητή
- 4.5.8. Αρχικοποίηση μεταβλητών σελίδας
- 4.5.9. Εισαγωγή στοιχείων από Β.Δ.
- 4.5.10. Ράβδος ανάδρασης
- 4.5.11. Έλεγχοι φόρμας επεξεργασίας στοιχείων προφίλ
- 4.5.12. Ανανέωση γνώσεων στην Β.Δ.
- 4.5.13. Περιγραφή προσθήκης και αφαίρεσης γνώσεων
- 4.5.14. Μεταφόρτωση εικόνας προφίλ
- 4.5.15. Διαγραφή ανεβασμένου αρχείου – εικόνας προφίλ

4.1. Κοινές λειτουργίες συστήματος

4.1.1. Σύνδεση

Η σελίδα αυτή αποτελεί τη πρώτη επαφή του χρήστη με το σύστημα καθώς είναι απαραίτητη να συμπληρωθεί από τον χρήστη με σωστά στοιχεία η οποία βρίσκεται στην διεύθυνση 'domain'/login. Αποτελεί την αρχική σελίδα που παραπέμπει τον επισκέπτη μόλις επισκεφτεί τον υπερσύνδεσμο της πλατφόρμας για να βεβαιωθεί ότι αρχικά έχει λογαριασμό και στην συνέχεια να του δώσει την δυνατότητα να έχει πρόσβαση στις λειτουργίες τις σελίδας. Τα απαραίτητα στοιχεία για συμπλήρωση είναι το όνομα χρήστη "username" και ο κωδικός πρόσβασης. Σε περίπτωση που ο χρήστης συμπληρώσει σωστά τα στοιχεία (ο έλεγχος θα γίνει στην Β.Δ.

κατά πόσο υπάρχουν τα συγκεκριμένα στοιχεία και είναι συνδεδεμένα) θα τον παραπέμψει στην αρχική σελίδα για άτομα με λογαριασμό οργανισμού, δηλαδή την σελίδα με της ρυθμίσεις του λογαριασμού ενώ σε αντίθετη περίπτωση θα εμφανίσει ενημερωτικό μήνυμα ότι τα στοιχεία εισόδου είναι εσφαλμένα. Αν δεν έχει λογαριασμό τότε επιλέγοντας τον υπερσύνδεσμο "Sign Up" μπορεί να δημιουργήσει ένα ενώ σε περίπτωση που δεν θυμάται τον κωδικό πρόσβασης του μπορεί να επιλέξει τον υπερσύνδεσμο "Forgot password" και να τον παραπέμψει στην ανάλογη σελίδα για να ακολουθήσει τα ανάλογα βήματα. [Εικόνα 4.1.1]

Εικόνα 4.1. 1: Σύνδεση

4.1.2. Λειτουργία επανάκτησης κωδικού πρόσβασης

Αυτή η σελίδα [Εικόνα 4.1.2] θεωρείται υποσελίδα της σελίδας της “Σύνδεσης” και χρησιμοποιείται για βοηθητικό σκοπό σε περίπτωση που κάποιος χρήστης ξεχάσει τον κωδικό πρόσβασης του, η οποία βρίσκεται στην διεύθυνση ‘domain’/forget. Ο χρήστης χρειάζεται να συμπληρώσει το όνομα χρήστη του “username”, το ηλεκτρονικό ταχυδρομείο που είναι συνδεδεμένο με τον λογαριασμό του και τέλος να επιλέξει τον υπερσύνδεσμο “SEND INSTRUCTIONS”. Αν τα στοιχεία εισόδου είναι έγκυρα θα βγάλει ειδοποίηση ότι προχωράει επιτυχημένα στο επόμενο βήμα (και σελίδα) σε αντίθετη περίπτωση θα παραμείνει στην ίδια σελίδα με ένα αποτυχημένο μήνυμα που εξηγεί το πρόβλημα. Η διαδικασία αυτή επαναλαμβάνεται μέχρι ο χρήστης να εισάγει επιτυχημένο ζεύγος στοιχείων. Την στιγμή που ο χρήστης θα εισάγει έγκυρα στοιχεία θα του αποσταλεί μήνυμα στο ηλεκτρονικό του ταχυδρομείο που θα περιέχει ένα μίας χρήσης μοναδικό αριθμό για να επιβεβαιώσει αφενός ότι το ηλεκτρονικό ταχυδρομείο ανήκει στο συγκεκριμένο χρήστη και αφετέρου ότι την ταυτότητα του χρήστη. [Εικόνα 4.1.3]

Εικόνα 4.1. 2: Λειτουργία επανάκτησης κωδικού πρόσβασης

Εικόνα 4.1. 3: Λειτουργία επανάκτησης κωδικού πρόσβασης – Βήμα 2

Password reset for platform www.platform343.ucy.ac.cy



epl343cs.ucy.ac.cy@gmail.com <epl343cs.ucy.ac.cy@gmail.com>
4/20/2021 2:12 AM

To: apanay22@ucy.ac.cy

Password Reset for Platform 343!

You, or someone acting as you has requested a password reset.

To continue with your password reset, please click [here](#) to continue and enter the following One Time Key (OTK) **539629** to continue.

If you do not wish to change your password at this time, you do not need to take any action.

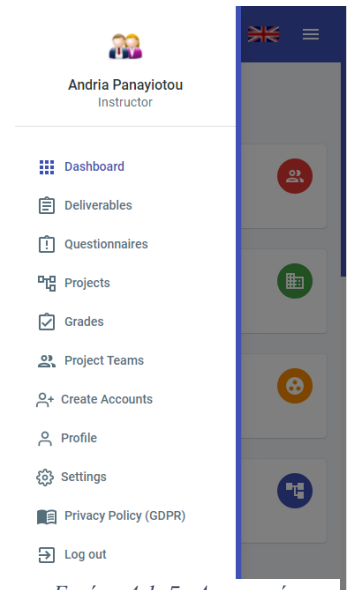
Εικόνα 4.1. 4: Στιγμιότυπο μηνύματος για λειτουργία επανάκτησης κωδικού πρόσβασης

Στη συνέχεια αφού κατευθύνθηκε στο βήμα 2 (και στην διεύθυνση ‘domain’/forgetcode) καλείτε να συμπληρώσει αυτή την φορά το όνομα χρήστη “username”, τον μίας χρήσης μοναδικό αριθμό (OTK) και τον καινούριο κωδικό πρόσβασης 2 φορές για επιβεβαίωση. Γίνονται οι κατάλληλοι έλεγχοι ότι για παράδειγμα ο OTK δεν ξαναχρησιμοποιήθηκε και αντιστοιχεί στο συγκεκριμένο χρήστη αλλά και ότι οι 2 κωδικοί πρόσβασης είναι όμοιοι. Στην συνέχεια ο χρήστης επιλέγει τον υπερσύνδεσμο “CHANGE PASSWORD ACCOUNT”

εμφανίζεται ξανά επιτυχημένο (ή αποτυχημένο μήνυμα ανάλογα με την κατάσταση) και ο χρήστης κατευθύνεται ξανά στην σε σελίδα της σύνδεσης [Εικόνα 4.1.1].

4.1.3. Λειτουργία Αποσύνδεσης

Η λειτουργία αποσύνδεσης είναι διαθέσιμη στον χρήστη υπό την προϋπόθεση ότι αρχικά έχει γίνει επιτυχημένη η λειτουργία της σύνδεσης (4.1.1). Δεν υπάρχει ξεχωριστή σελίδα αλλά ο χρήστης έχει πρόσβαση σε αυτή από το κεντρικό μενού της εφαρμογής [Εικόνα 4.1.5] αλλά και από την κεφαλίδα της εφαρμογής [Εικόνα 4.1.6]. Συγκεκριμένα μόλις ο χρήστης εκτελέσει αυτή την λειτουργία διαγράφεται η κατάσταση που είναι αποθηκευμένη στον συγκεκριμένο υπολογιστή (session) με στοιχεία του χρήστη (π.χ. αναγνωριστικό αριθμό χρήστη, τύπο λογαριασμού, ονοματεπώνυμο κτλ.) και παραπέμπει τον χρήστη ξανά στην σελίδα της σύνδεσης (4.1.1).



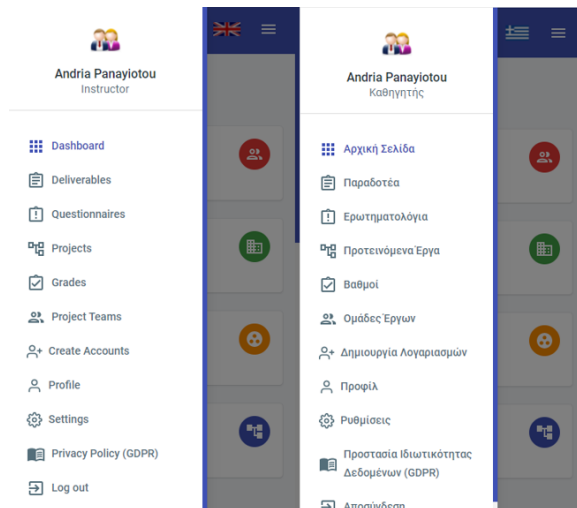
Εικόνα 4.1. 5: Λειτουργία Αποσύνδεσης μενού



Εικόνα 4.1. 6: Λειτουργία Αποσύνδεσης κεφαλίδα

4.1.4. Λειτουργία αλλαγής γλώσσας εμφάνισης περιεχομένου σελίδας

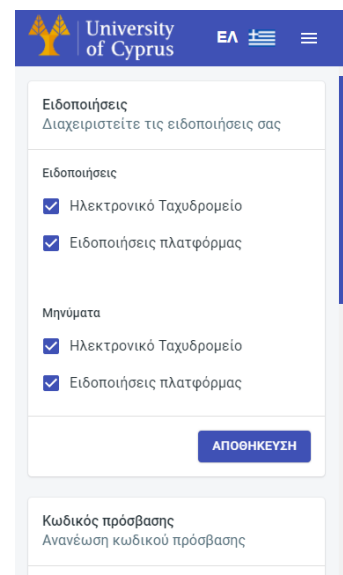
Η διαδικτυακή είναι διαθέσιμη σε 2 γλώσσες (Ελληνικά, Αγγλικά) για να μπορεί ο κάθε χρήστης να επιλέγει αυτή που τεριάζει περισσότερο σε εκείνον από την κεφαλίδα σε κάθε σελίδα [Εικόνα 4.1.6]. Ο τρόπος με τον οποίο γίνεται η μετάφραση επεξηγείτε στο κεφάλαιο 4.5.3. Κάθε περιεχόμενο της σελίδα (εξαιρουμένου τα δεδομένα που ανεβάζουν οι χρήστες π.χ. πληροφορίες έργων κτλ.) γίνεται μετάφραση αμέσως όπως φαίνεται και από την Εικόνα 4.1.7 όπου παρουσιάζεται η μετάφραση για την γραμμή μενού στα αριστερά.



Εικόνα 4.1. 7: Λειτουργία αλλαγής γλώσσας εμφάνισης περιεχομένου σελίδας

4.1.5. Λειτουργία διαχείρισης ειδοποιήσεων λογαριασμών

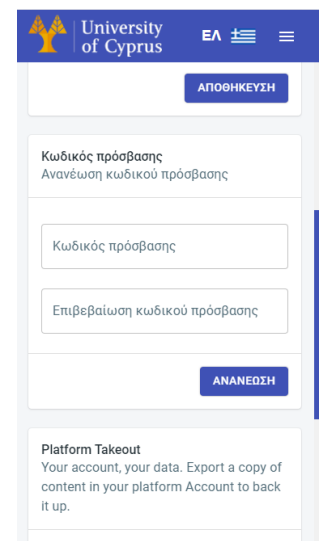
Κάθε χρήστης έχει την δυνατότητα να επιλέξει τον τρόπο με τον οποίο θα ενημερώνεται από την πλατφόρμα για διάφορες λειτουργίες και δραστηριότητες των χρηστών, για παράδειγμα την στιγμή που ο καθηγητής ανεβάζει βαθμό για κάποιο παραδοτέο θα πρέπει να ενημερωθεί ο φοιτητής. Η λειτουργία αυτή είναι διαθέσιμη στην υποσελίδα ‘domain’/app/settings. Από προεπιλογή είναι ορισμένο να παίρνει μηνύματα στο ηλεκτρονικό του ταχυδρομείο ο φοιτητής όμως από την σελίδα των ρυθμίσεων υπάρχει η δυνατότητα να τερματίσει ή να επαναλειτουργήσει την λειτουργία αυτή ανάλογα με την εξατομίκευση κάθε ατόμου και πως προτιμάει να ενημερώνεται. [Εικόνα 4.1.8]



Εικόνα 4.1. 8: Λειτουργία διαχείρισης ειδοποιήσεων λογαριασμών

4.1.6. Λειτουργία αλλαγής κωδικού πρόσβασης

Κάθε χρήστης έχει την δυνατότητα ανά πάσα στιγμή (μετά την σύνδεση του στην σελίδα – για να γίνει η ταυτοποίηση) να αλλάξει τον κωδικό πρόσβασης του αφού επισκεφτεί την υποσελίδα ‘domain’/app/settings. Το μόνο που έχει να κάνει είναι να συμπληρώσει τον νέο κωδικό πρόσβασης μαζί με τον επιβεβαιωτικό κωδικό πρόσβασης και τότε θα γίνει ο συνηθισμένος έλεγχος για ομοιότητα και στην συνέχεια θα γίνει η αλλαγή στην Β.Δ.. Κατά συνέπεια από την επόμενη φορά που θα θέλει να συνδεθεί στην πλατφόρμα θα πρέπει να συμπληρώσει τον καινούριο κωδικό πρόσβασης έτσι ώστε να έχει πρόσβαση.

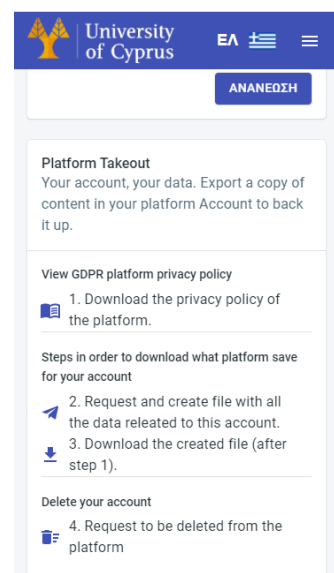


Εικόνα 4.1. 9: Λειτουργία αλλαγής κωδικού πρόσβασης

4.1.7. Λειτουργία προβολής και αποθήκευσης στοιχείων GDPR

Λαμβάνοντας υπόψη τον Γενικό Κανονισμό για την Προστασία Δεδομένων⁴ κάθε χρήστης έχει την δυνατότητα να γνωρίζει τα στοιχεία τα οποία αποθηκεύονται για το πρόσωπο του και εννοείται αυτό θα ξεκινήσει να γίνεται αφού πρώτα γίνει την συγκατάθεση του (αυτό ο χρήστης στην πλατφόρμα μας το κάνει την στιγμή που δημιουργεί λογαριασμό). Κάθε χρήστης έχει την δυνατότητα να διαβάσει της οδηγίες και κανόνες που ακολουθεί η συγκεκριμένη πλατφόρμα και ποια στοιχεία κρατάει για τον κάθε χρήστη ανάλογα με τον τύπο του λογαριασμού και πως τα επεξεργάζεται. Αυτό μπορεί να το διαβάσει αν επιλέξει τον υπερσύνδεσμο, ο οποίος βρίσκεται στην υποσελίδα ('domain'/app/settings) στο Βήμα 1 – “Download the privacy policy of the platform” και θα ανοίξει σε καινούριο παράθυρο στο φυλλομετρητή

ένα αρχείο το οποίο τα περιγράφει αναλυτικά. Επιπλέον κάθε χρήστης έχει την δυνατότητα να δει αποθηκεύσει στον υπολογιστή του όλα τα στοιχεία που υπάρχουν στην πλατφόρμα σχετικά με αυτόν ακολουθώντας απλά 2 βήματα. Αρχικά, θα ζητήσει να δημιουργηθεί το αρχείο το οποίο περιέχει τις πληροφορίες αυτές επιλέγοντας τον υπερσύνδεσμο “Request and create file with all the data related to this account” και στην συνέχεια μόλις ετοιμαστεί θα εμφανιστεί ένα ενημερωτικό μήνυμα για να συνεχίσει στο επόμενο βήμα στο οποίο απλά θα επιλέξει τον υπερσύνδεσμο “Download the created file” και τότε θα ξεκινήσει η αποθήκευση του αρχείου. Μία ακόμη δυνατότητα που δίνεται στους χρήστες είναι η οριστική διαγραφή τους από την πλατφόρμα μαζί με τα στοιχεία τους. Για χρήστη με λογαριασμό οργανισμού μόλις ζητήσει την οριστική του διαγραφή ενημερώνει τον ηλεκτρονικό ταχυδρομείο της πλατφόρμας με μήνυμα και γίνεται αμέσως η διαγραφή ενώ για φοιτητή στέλνεται απλά μήνυμα όπως φαίνεται στην Εικόνα 4.1.11 και χρειάζεται έγκριση από χρήστη με τύπο καθηγητή.



Εικόνα 4.1. 10: Λειτουργία προβολής και αποθήκευσης στοιχείων GDPR

⁴ http://www.dataprotection.gov.cy/dataprotection/dataprotection.nsf/page3a_gr/page3a_gr?opendocument



epl343cs.ucy.ac.cy@gmail.com <epl343cs.ucy.ac.cy@gmail.com>

4/19/2021 12:03 AM



To: antriaapoel@gmail.com

Student request deletion!

The below student requested to be deleted from our platform and also all his/het data.

Account credentials are:

Username: **gpapav02**

Name: **Giorgos**

Surname: **Papavarnavas**

ID: **90**

Εικόνα 4.1. 11: Στιγμιότυπο μηνύματος για λειτουργία προβολής και αποθήκευσης στοιχείων GDPR

4.1.8. Λειτουργία αλλαγής στοιχεία προφίλ

Κάθε χρήστης έχει το δικαίωμα αν επισκεφτεί την σελίδα προφίλ, η οποία βρίσκεται στην υποσελίδα ('domain'/app/account), να επεξεργαστεί και να αλλάξει τα στοιχεία που αφορούν

Εικόνα 4.1. 12: Λειτουργία αλλαγής στοιχεία προφίλ

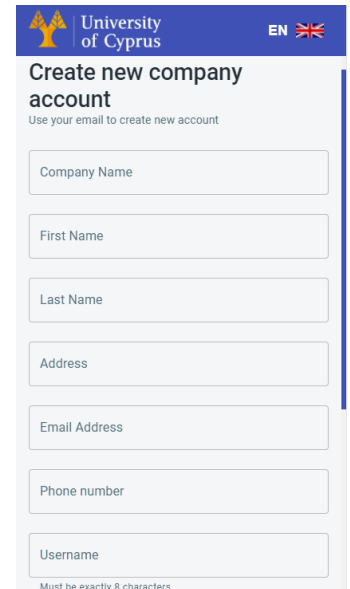
το όνομα, το επώνυμο του και το ηλεκτρονικό του ταχυδρομείο. Για κάθε τύπο λογαριασμού οργανισμός/φοιτητή/καθηγητή υπάρχουν επιπλέον ξεχωριστές ρυθμίσεις οι οποίες θα αναλυθούν στις λειτουργίες κάθε τύπου λογαριασμού ξεχωριστά. Όσο αφορά τις γενικές αλλαγές στο πλαίσιο κειμένου (textbox) εμφανίζεται η τιμή που ήδη υπάρχει στην Β.Δ. αποθηκευμένη και αν ο χρήστης επιλέξει να αλλάξει στο τέλος επιλέγει τον υπερσύνδεσμο “ΑΠΟΘΗΚΕΥΣΗ ΠΛΗΡΟΦΟΡΙΩΝ”, γίνεται η σχετική ενημέρωση στην Β.Δ. και τέλος εμφανίζεται και το ανάλογο μήνυμα επιτυχημένης αλλαγής στον χρήστη.

4.2. Διεπαφή Οργανισμού

4.2.1. Δημιουργία Λογαριασμού

Οι χρήστες με τύπο λογαριασμό οργανισμού δημιουργούν τον λογαριασμό τους μόνοι τους επιλέγοντας αυτοί τα στοιχεία που επιθυμούν για τον εαυτό τους. Η δυνατότητα αυτή βρίσκεται στην υποσελίδα ('domain'/register) και ο χρήστης μπορεί να κατευθυνθεί σε αυτή από την σελίδα της 'Σύνδεσης' η οποία λειτουργεί σαν αρχική σελίδα πριν να συνδεθεί κάποιος χρήστης με τα στοιχεία του. Ο νέος χρήστης καλείται να συμπληρώσει υποχρεωτικά το όνομα και το επώνυμο του ατόμου επικοινωνίας για την οργανισμός, την διεύθυνση ηλεκτρονικού ταχυδρομείου και ένα τηλέφωνο επικοινωνίας.

Επιπλέον, θα πρέπει να επιλέξει όνομα χρήστη "username" (το οποίο πρέπει να είναι ακριβώς 8 χαρακτήρες) και κωδικό πρόσβασης τα οποία θα χρησιμοποιεί για να συνδέεται στην σελίδα. Προαιρετικά μπορεί να συμπληρώσει και επιπλέον στοιχεία όπως για παράδειγμα το όνομα του οργανισμού και την διεύθυνση της. Στην συνέχεια γίνεται ο έλεγχος από την πλατφόρμα για την ορθότητα των στοιχείων εισόδου (π.χ. αν το ηλεκτρονικό ταχυδρομείο βρίσκεται στην σωστή μορφή, αν συμπληρώθηκαν όλα τα υποχρεωτικά πεδία κτλ.), αν υπάρχει διαθέσιμος λογαριασμός με το συγκεκριμένο όνομα χρήστη (λειτουργεί σαν μοναδικό κλειδί πεδίο / απαγορεύονται τα διπλότυπα) και κατά πόσο ο χρήστης αποδέχτηκε τους κανόνες της πλατφόρμας σχετικά με τον Γενικό Κανονισμό της Προστασίας Δεδομένων (GDPR). Σε περίπτωση που δεν ισχύει κάτι από τα πιο πάνω εμφανίζεται ενημερωτικό μήνυμα στο χρήστη και η δημιουργία του λογαριασμού αναστέλλεται μέχρι να συμπληρωθούν κατάλληλα. Τέλος, τα δεδομένα αυτά μεταφέρονται στην Β.Δ. όπου δημιουργείται ένας λογαριασμός τύπου οργανισμού και πλέον μπορεί να συνδεθεί με τα πιστοποιητικά σύνδεσης του (όνομα χρήστη, κωδικό πρόσβασης).



University of Cyprus EN

Create new company account

Use your email to create new account

Company Name

First Name

Last Name

Address

Email Address

Phone number

Username

Must be exactly 8 characters

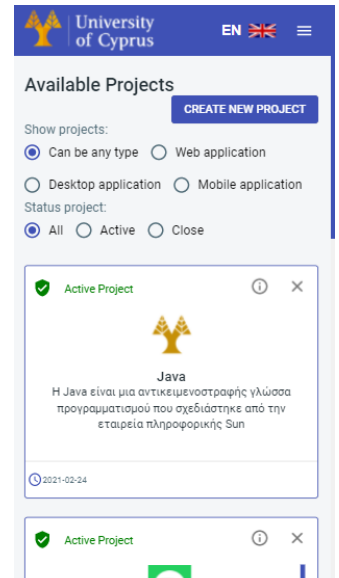
Εικόνα 4.2.1. 1: Λειτουργία δημιουργίας λογαριασμού

4.2.2. Έργα Ανάθεσης

Μία από της πιο σημαντικές λειτουργίες που μπορεί να εκτελέσει ένας χρήστης με λογαριασμό τύπου οργανισμού είναι να επισκεφτεί την σελίδα με τα έργα ανάθεσης ('domain'/projects). Εκεί μπορεί να δει όλα τα προηγούμενα έργα ανάθεσης (Ε.Α.) που δημιούργησε και να ανεβάσει καινούρια.

4.2.2.1. Περιορισμένη εμφάνιση αποτελεσμάτων

Αρχικά όπως βλέπουμε και στην Εικόνα 4.2.2.1 μπορεί να επιλέξει ποια έργα να εμφανίζονται και να περιορίσει τα περιεχόμενα επιλέγοντας φίλτρο σχετικά με το είδος της εφαρμογής ή την κατάσταση του έργου. Όσο αφορά το φίλτρο με το είδος της εφαρμογής υπάρχουν 4 διαθέσιμες επιλογές, οι 3 είναι ανάλογα με το είδος που όρισε ο δημιουργός του έργου ότι προτιμάει την υλοποίηση (διαδικτυακή εφαρμογή, desktop εφαρμογή, κινητή εφαρμογή) και η άλλη εμφανίζει όλα τα είδη έργου ανάθεσης ανεξαιρέτως κατηγορίας. Σχετικά με το φίλτρο σχετικά με την κατάσταση έργου υπάρχουν 3 επιλογές, η πρώτη εμφανίζει όλα τα έργα και των 2 κατηγοριών, η δεύτερη εμφανίζει μόνο τα ενεργά έργα (δηλαδή αυτά που έχει την δυνατότητα ο φοιτητής να τα επιλέξει για υλοποίηση) και η τρίτη επιλογή εμφανίζει τα έργα που έχουν ήδη λήξει (και ο φοιτητής δεν έχει την δυνατότητα να τα επιλέξει).



Εικόνα 4.2.2. 1: Προβολή έργων ανάθεσης

4.2.2.2. Δημιουργία καινούριου Ε.Α.

Χρήστες με λογαριασμό οργανισμού ή καθηγητή έχουν την δυνατότητα να δημιουργήσουν άπειρα Ε.Α. αρκεί να επιλέξουν τον υπερσύνδεσμο “CREATE NEW PROJECT” για να εμφανιστεί καινούριο αναδυόμενο παράθυρο όπως φαίνεται στην Εικόνα 4.2.2.3. Ο χρήστης έτσι ώστε να δημιουργήσει επιτυχημένα ένα καινούριο Ε.Α. καλείτε να συμπληρώσει ένα αντιπροσωπευτικό όνομα μαζί με μία περιγραφή, να επιλέξει τον τύπο της εφαρμογής (διαδικτυακή εφαρμογή, desktop εφαρμογή, κινητή εφαρμογή, οποιουδήποτε τύπου, αν δεν έχει αποφασιστεί ακόμη) και προαιρετικά αν θέλει να επιλέξει ένα αρχείο PDF με επιπλέον πληροφορίες. Στην συνέχεια επιλέγει τον υπερσύνδεσμο “CREATE PROJECT”, γίνεται ο έλεγχος για την συμπλήρωση όλων των πεδίων, μεταφορτώνεται το αρχείο περιγραφής στον διακομιστή, γίνεται η εγγραφή στην Β.Δ. και τέλος εμφανίζεται ενημερωτικό μήνυμα για την κατάσταση δημιουργίας του Ε.Α.

4.2.2.3. Επεξεργασία στοιχείων Ε.Α.

Οποιαδήποτε στιγμή επιθυμεί ο δημιουργός ενός συγκεκριμένου Ε.Α. έχει την δυνατότητα να επεξεργαστεί της πληροφορίες του. Συγκεκριμένα μπορεί να ενημερώσει ίσως τον τίτλο ή την περιγραφή του Ε.Α., να προσθέσει ή να αφαιρέσει υλικό (για παράδειγμα αρχείο με επιπλέον πληροφορίες PDF) ή να αλλάξει την κατάσταση από ενεργό έργο σε κλειστό. Αυτό μπορεί να το πετύχει επιλέγοντας το εικονίδιο με το μολυβί που αντιστοιχεί στην επεξεργασία όπως φαίνεται και στην Εικόνα 4.2.2.1. Εκτελούνται οι κατάλληλοι έλεγχοι για ορθή ενημέρωση

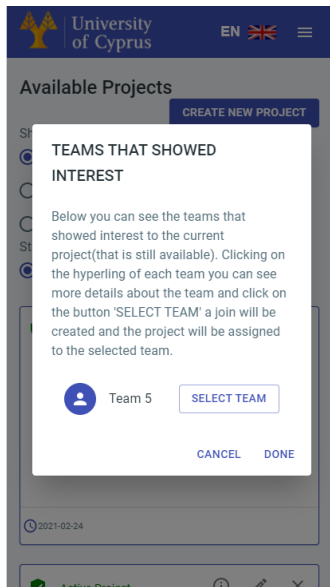
των στοιχείων και γίνεται η αλλαγή τόσο στον διακομιστή για πιθανή ενημέρωση του PDF αρχείου με τις περισσότερες πληροφορίες όσο και στην Β.Δ.

4.2.2.4. Προβολή ενδιαφερόμενων ομάδων για συγκεκριμένο Ε.Α.

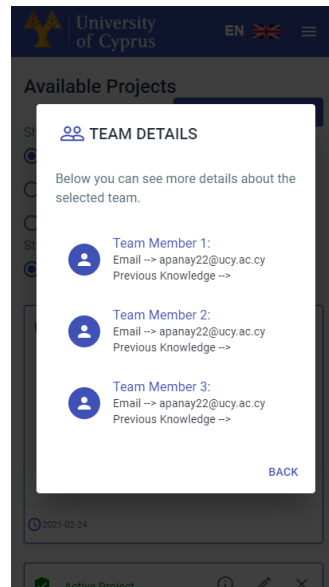
Όλοι οι φοιτητές που βρίσκονται σε ομάδα που έχει εγκριθεί από κάποιο καθηγητή έχουν την δυνατότητα να δείξουν ενδιαφέρον για να αναλάβουν το συγκεκριμένο έργο και να το υλοποιήσουν. Ο χρήστης ο οποίος δημιούργησε το συγκεκριμένο Ε.Α. μπορεί να δει όλες τις ομάδες οι οποίες έδειξαν ενδιαφέρον (επιλέγοντας το εικονίδιο της πληροφορίας όπως φαίνεται στην Εικόνα 4.2.2.1) και κάποιες επιπλέον πληροφορίες γι' αυτούς, για παράδειγμα από τον αριθμό των μελών που απαρτίζουν την ομάδα, ηλεκτρονικό ταχυδρομείο επικοινωνίας και τις προηγούμενες γνώσεις που έχουν τα άτομα αυτά όπως φαίνεται στην Εικόνα 4.2.2.5. Επιπλέον, προσωπικά στοιχεία όπως το όνομα και το επίθετο των ατόμων που απαρτίζουν την ομάδα δεν είναι διαθέσιμα σε αρχικό στάδιο έτσι ώστε να προστατευτούν και να γίνει πιο αντικειμενική ανάθεση Ε.Α. σε ομάδες. Στην συνέχεια, μόλις ο οργανισμός αποφασίσει σε ποια ομάδα θέλει να αναθέσει το έργο επιλέγει το κουμπί με τον υπερσύνδεσμο “SELECT TEAM” (όπως φαίνεται στην ομάδα Εικόνα 4.2.2.2) και πλέον όρισε ομάδα ανάθεσης. Από αυτή την στιγμή και μετά μπορεί να δει και πιο προσωπικές πληροφορίες όπως για παράδειγμα το όνομα και το επώνυμο κάθε ατόμου στην ομάδα για να μπορεί να γίνει η ταυτοποίηση ευκολότερα, όπως φαίνεται και στην Εικόνα 4.2.2.4.

4.2.2.5. Διαγραφή Ε.Α.

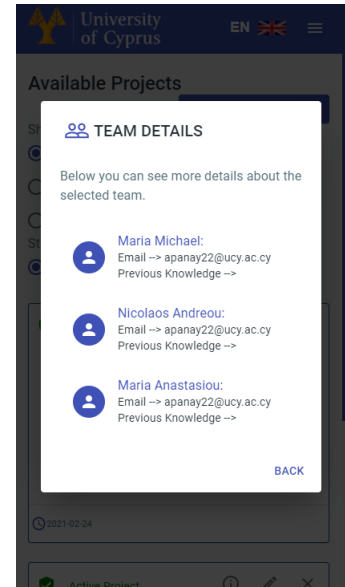
Ο δημιουργός κάθε Ε.Α. έχει την δυνατότητα ανά πάσα στιγμή να το διαγράψει επιλέγοντας το τρίτο εικονίδιο του Χ (στο συγκεκριμένο Ε.Α.) όπως φαίνεται στη εικόνα 4.2.2.1. Εμφανίζεται αναδυόμενο παράθυρο για να επιβεβαιωθεί ξανά η απόφαση του χρήστη να διαγράψει το συγκεκριμένο Ε.Α. όπως φαίνεται και στην Εικόνα 4.2.2.7. και να αποφευχθούν ατυχήματα από λανθασμένη επιλογή του υπερσυνδέσμου διαγραφής. Η διαγραφή συμβαίνει τόσο στην Β.Δ. για τα στοιχεία που είναι αποθηκευμένα όσο και στον διακομιστή ο οποίος περιέχει αρχείο PDF με περισσότερες πληροφορίες.



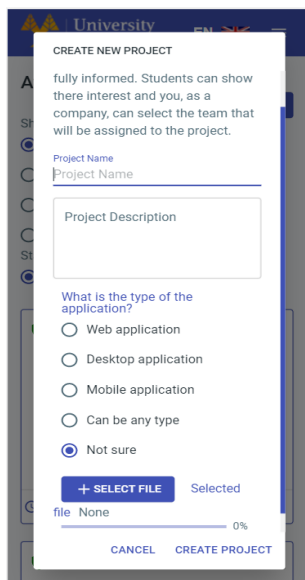
Εικόνα 4.2.2. 7: Προβολή ενδιαφερόμενων ομάδων



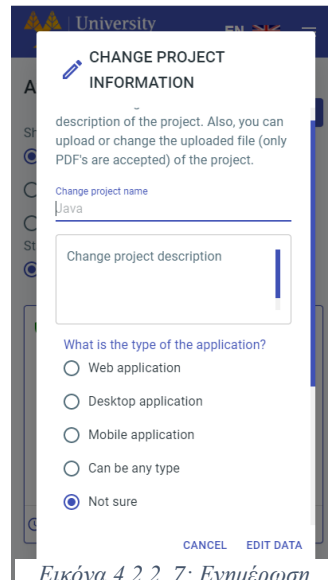
Εικόνα 4.2.2. 7: Προβολή στοιχείων ομάδας - ανώνυμα



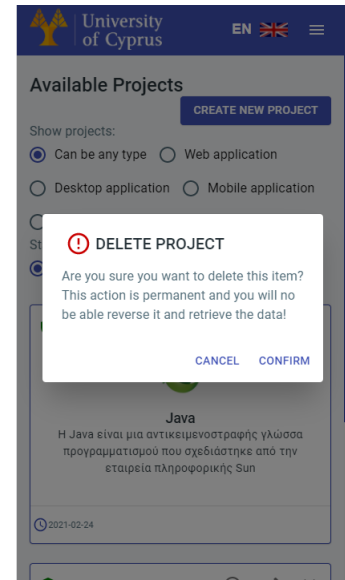
Εικόνα 4.2.2. 7: Προβολή στοιχείων ομάδας - ταυτοποιημένα



Εικόνα 4.2.2. 7: Δημιουργία έργου ανάθεσης



Εικόνα 4.2.2. 7: Ενημέρωση πληροφοριών Ε.Α.



Εικόνα 4.2.2. 7: Διαγραφή Ε.Α.

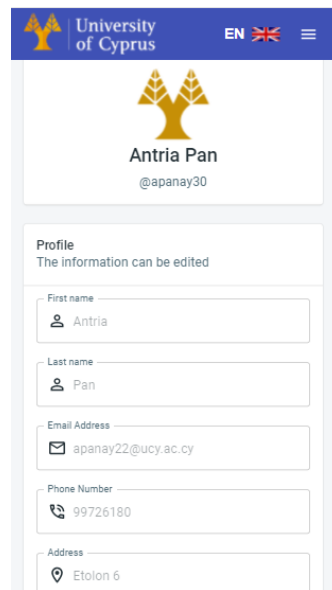
4.2.3. Προφίλ

4.2.3.1. Επεξεργασία στοιχείων προφίλ

Κάθε χρήστης έχει το δικαίωμα αν επισκεφτεί την σελίδα προφίλ, η οποία βρίσκεται στην υποσελίδα ('domain'/app/account), να επεξεργαστεί και να αλλάξει τα στοιχεία που αφορούν το όνομα το επώνυμο του και το ηλεκτρονικό του ταχυδρομείο. Επιπλέον από τα κοινά επεξεργαζόμενα στοιχεία τα οποία περιγράφονται στην ενότητα 4.1.8 επεξεργάζεται επίσης ο αριθμός κινητού τηλεφώνου, η διεύθυνση και το όνομα του οργανισμού.

Για κάθε τύπο λογαριασμού οργανισμός/φοιτητή/καθηγητή υπάρχουν επιπλέον ξεχωριστές ρυθμίσεις οι οποίες θα αναλυθούν στις λειτουργίες κάθε τύπου λογαριασμού ξεχωριστά. Όσο

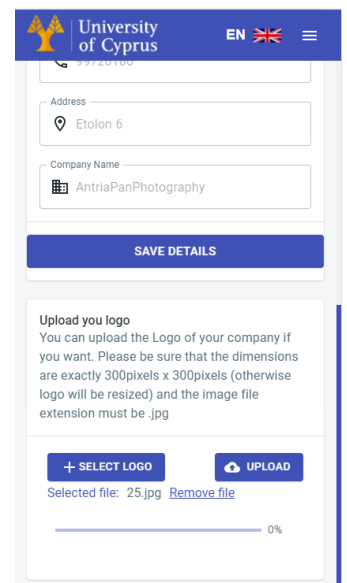
αφορά τις γενικές αλλαγές στο πλαίσιο κειμένου (textbox) εμφανίζεται η τιμή που ήδη υπάρχει στην Β.Δ. αποθηκευμένη και αν ο χρήστης επιλέξει να αλλάξει στο τέλος επιλέγει τον υπερσύνδεσμο “ΑΠΟΘΗΚΕΥΣΗ ΠΛΗΡΟΦΟΡΙΩΝ”, γίνεται η σχετική ενημέρωση στην Β.Δ. και τέλος εμφανίζεται και το ανάλογο μήνυμα επιτυχημένης αλλαγής στον χρήστη.



Εικόνα 4.2.3. 1: Επεξεργασία στοιχείων προφίλ

4.2.3.2. Μεταφόρτωση εικόνας προφίλ - λογότυπο

Χρήστες με λογαριασμό οργανισμού έχουν την δυνατότητα να ανεβάζουν εικόνες προφίλ που στις περισσότερες περιπτώσεις είναι το λογότυπο του οργανισμού τους. Αυτό βοηθάει στην γρήγορη ταυτοποίηση τόσο από φοιτητές αλλά και καθηγητές κυρίως στα αναρτημένα Ε.Α όπως φαίνεται και στην Εικόνα 4.2.2.1. Η διαδικασία είναι πολύ απλή καθώς αν ο χρήστης επισκεφτεί την σελίδα προφίλ, η οποία βρίσκεται στην υποσελίδα ('domain'/app/account) επιλέγοντας τον υπερσύνδεσμο “SELECT LOGO” μπορεί να επιλέξει οποιαδήποτε εικόνα μορφής “.jpg” και κατά προτίμηση με διαστάσεις 300 pixels X 300 pixels. Η πλατφόρμα δέχεται εικόνες όλων των διαστάσεων απλά σε αντίθετη περίπτωση γίνεται αυτόματη αλλαγή του μεγέθους και πολύ πιθανόν να μειωθεί/αλλοιωθεί η ανάλυση της εικόνας. Στην συνέχεια μετά την επιτυχής επιλογή του αρχείου εικόνας ο χρήστης θα επιλέξει τον υπερσύνδεσμο “UPLOAD” για να γίνει η μεταφόρτωση στον διακομιστή αλλά και η σύνδεση στην Β.Δ. ότι υπάρχει συνδεδεμένη εικόνα. Όσο εκτελείτε η διαδικασία που μόλις αναφέρθηκε υπάρχει μία οριζόντια ράβδος αρχικοποιημένη με το ποσοστό 0% η οποία δείχνει την ανάδραση της εκτέλεσης και αλλάζει ανάλογα μέχρι να φτάσει στο μέγιστο 100% το οποίο ισούται τόσο με το επιτυχημένο τέλος της εκτέλεσης της διαδικασίας όσο και με τις



Εικόνα 4.2.3. 2: Μεταφόρτωση εικόνας προφίλ

αλλαγές σε διακομιστή και Β.Δ. Μία ακόμη δυνατότητα είναι οποιαδήποτε στιγμή ο χρήστης να αφαιρέσει την εικόνα προφίλ που έχει ήδη μεταφορτώσει με την επιλογή του υπερσυνδέσμου “Remove file” και τότε γίνεται η ενημέρωση τόσο στην Β.Δ. όσο και στον διακομιστή. Ανά πάσα στιγμή στο “Selected File” αν υπάρχει κάποια ονομασία αρχείου τότε ισούται με το γεγονός ότι ο χρήστης μεταφόρτωσε αρχείο αλλιώς θα αναφέρετε ως “None” και



Εικόνα 4.2.3. 3: Προκαθορισμένες εικόνες προφίλ για όλα τα τύποι λογαριασμού

θα υπάρχει η προκαθορισμένη εικόνα προφίλ όπως φαίνεται και στην Εικόνα 4.1.3.3.

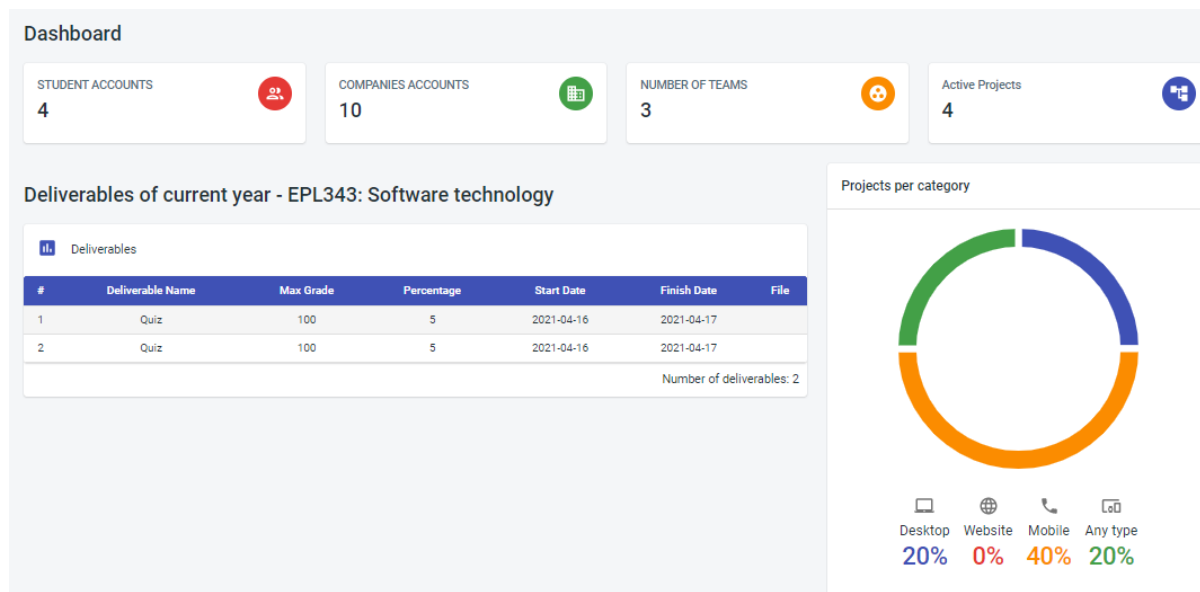
Ο λόγος για τον οποίο υπάρχουν φωτογραφίες προφίλ και για τους άλλους δύο τύπους λογαριασμών (Καθηγητή, Φοιτητή), παρόλο που δεν εμφανίζονται κάπου για τους υπόλοιπους χρήστες, είναι καθαρά θέμα εμφάνισης για τον ίδιο τον χρήστη στην περιοχή του μενού όπως φαίνεται και στην Εικόνα 4.1.5. Η μόνη διαφορά είναι ότι η εικόνα προφίλ στην περίπτωση αυτή δεν μπορεί να γίνει επεξεργασία και εμφανίζεται πάντα η προκαθορισμένη.

4.3.Διεπαφή Φοιτητή

4.3.1. Κεντρική Σελίδα

Η κεντρική σελίδα η οποία κατευθύνεται ο χρήστης με τύπο λογαριασμού Φοιτητή φαίνεται στην Εικόνα 4.1.3.4. Αρχικά μπορεί να δει κάποιες πληροφορίες σχετικά για το τρέχον ακαδημαϊκό έτος όπως για παράδειγμα ο αριθμός φοιτητών που είναι εγγεγραμμένοι για την φετινή ακαδημαϊκή χρονιά, τον συνολικό πλήθος ενεργών χρηστών με τύπο λογαριασμό ‘οργανισμού’ που μπορούν να αναρτήσουν Ε.Α. , τον αριθμό των ομάδων που δημιουργήθηκαν και τα ενεργά Ε.Α. που μπορεί κάποια ΟΑΕ να δείξει ενδιαφέρον για υλοποίηση. Στον πίνακα (κάτω αριστερά της σελίδας) φαίνονται όλα τα παραδοτέα που υπάρχουν για το τρέχον ακαδημαϊκό έτος μαζί με κάποιες επιπλέον πληροφορίες για παράδειγμα τον μέγιστο βαθμό, το αντίκτυπο που υπάρχει στον τελικό βαθμό, ένα προαιρετικό αρχείο PDF με επιπρόσθετες πληροφορίες και το εύρος ημερομηνιών που θα είναι ενεργό. Συμπληρωματικά από τον αριθμό των ενεργών Ε.Α. το οποίο αναφέρεται απλά στο πλήθος τους (στο πάνω μέρος της σελίδας) υπάρχει και κυκλική γραφική παράσταση η οποία δείχνει παραστατικά την αναλογία και την προτίμηση που έχουν οι οργανισμοί στο είδος της εφαρμογής για τα έργα τους. Για παράδειγμα στο στιγμιότυπο της Εικόνας 4.1.3.4 υπάρχουν 20% desktop εφαρμογές, 0% διαδικτυακές εφαρμογές, 40% κινητές εφαρμογές και 20%

οποιοδήποτε τύπου δηλαδή από τις τέσσερις συνολικά που υπάρχουν οι δύο είναι κινητές εφαρμογές, μία είναι desktop εφαρμογή, μία είναι οποιοδήποτε τύπου και δεν υπάρχει καμία διαδικτυακή εφαρμογή.



Εικόνα 4.1.3. 4: Κεντρική σελίδα λογαριασμού φοιτητή

4.3.2. Παραδοτέα

Στην σελίδα των παραδοτέων η οποία βρίσκεται στην διεύθυνση 'domain'/deliverables εμφανίζεται ένα πίνακα όπως φαίνεται και στην Εικόνα 4.3.2.1 στον οποίο παρουσιάζονται όλα τα παραδοτέα μαζί με συμπληρωματικές πληροφορίες όπως μπορείτε να αντικρίσετε. Αρχικά από τα αριστερά στα δεξιά εμφανίζεται ο αύξων αριθμός, το όνομα του παραδοτέου, ο μέγιστος βαθμός για το συγκεκριμένο παραδοτέο, το ποσοστό που λειτουργεί ως αντίκτυπο στο τελικό βαθμό του φοιτητή, το εύρος των ημερομηνιών που είναι ισχύει το συγκεκριμένο παραδοτέο (αρχική και τελική ημερομηνία) και τέλος ένα αρχείο με επιπλέον πληροφορίες που μπορεί να υπάρχει προαιρετικά. Το συγκεκριμένο αρχείο ο φοιτητής επιλέγοντας τον υπερσύνδεσμο με το 'Download file' έχει την δυνατότητα να το διαβάσει και να το αποθηκεύσει στην ηλεκτρονική του συσκευή. Στην συγκεκριμένη σελίδα εμφανίζονται μόνο

Deliverables of current year - EPL343: Software technology

#	Deliverable Name	Max Grade	Percentage	Start Date	Finish Date	File
1	Quiz	100	5	2021-04-16	2021-04-17	
2	Deliverable Phase A	100	10	2021-05-02	2021-05-02	Download File

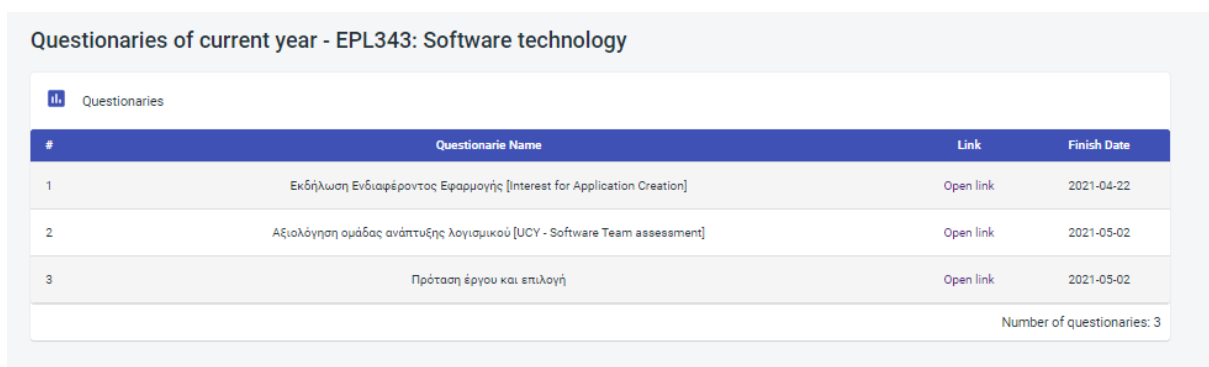
Number of deliverables: 2

Εικόνα 4.3.2. 1: Στιγμιότυπο οθόνης σελίδας παραδοτέων

τα παραδοτέα τα οποία ο καθηγητής έχει επιλέξει ως προσβάσιμα στους φοιτητές, συμπερασματικά μπορεί ο καθηγητής να έχει ανεβάσει από την αρχή της χρονιάς όλα τα παραδοτέα και να τα εμφανίζει σε συγκεκριμένη ημερομηνία (η διαδικασία περιγράφεται στην ενότητα 4.4.2.2). Τέλος στο υποσέλιδο του πίνακα υπάρχει συνολικά το πλήθος των παραδοτέων σε περίπτωση που είναι μεγάλος αριθμός να ξεχωρίζει αμέσως ο αριθμός.

4.3.3. Ερωτηματολόγια

Στην σελίδα των ερωτηματολογίων η οποία βρίσκεται στην διεύθυνση ‘domain’/ questionnaires εμφανίζει ένα πίνακα όπως φαίνεται και στην Εικόνα 4.3.3.1 στον οποίο παρουσιάζονται όλα τα ερωτηματολόγια μαζί με συμπληρωματικές πληροφορίες όπως μπορείτε να αντικρίσετε. Αρχικά από τα αριστερά στα δεξιά εμφανίζεται ο αύξων αριθμός, το όνομα του ερωτηματολογίου, τον σύνδεσμο που μας παραπέμπει στην φόρμα του ερωτηματολογίου και την ημερομηνία της τελικής προθεσμίας που πρέπει να συμπληρωθεί από τους φοιτητές. Όταν ο φοιτητής επιλέξει τον υπερσύνδεσμο “Open Link” τον παραπέμπει στην ηλεκτρονική φόρμα συμπλήρωσης του ερωτηματολογίου (συνήθως στο Google Forms). Στην συγκεκριμένη σελίδα εμφανίζονται μόνο τα ερωτηματολόγια τα οποία ο καθηγητής έχει επιλέξει ως προσβάσιμα στους φοιτητές, συμπερασματικά μπορεί ο καθηγητής να έχει ανεβάσει από την αρχή της χρονιάς όλους τους υπερσυνδέσμους για τα ερωτηματολόγια και να τα εμφανίζει σε συγκεκριμένη ημερομηνία (η διαδικασία περιγράφεται στην ενότητα 4.4.3.2). Τέλος στο υποσέλιδο του πίνακα υπάρχει συνολικά το πλήθος των ερωτηματολογίων σε περίπτωση που είναι μεγάλος αριθμός να ξεχωρίζει αμέσως ο αριθμός.



#	Questionnaire Name	Link	Finish Date
1	Εκδήλωση Ενδιαφέροντος Εφαρμογής [Interest for Application Creation]	Open link	2021-04-22
2	Αξιολόγηση ομάδας ανάπτυξης λογισμικού [UCV - Software Team assessment]	Open link	2021-05-02
3	Πρόταση έργου και επιλογή	Open link	2021-05-02

Number of questionnaires: 3

Εικόνα 4.3.3. 1: Στιγμιότυπο οθόνης ερωτηματολογίων

4.3.4. Έργα Ανάθεσης

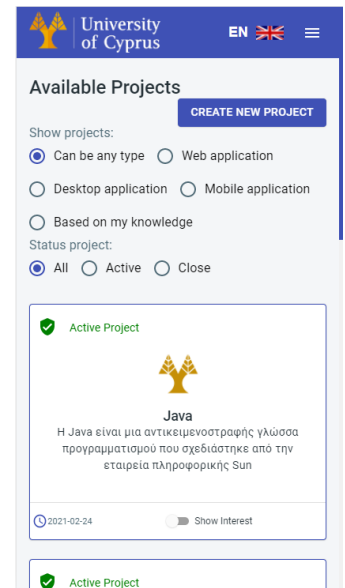
Μία από της πιο σημαντικές λειτουργίες που μπορεί να εκτελέσει ένας χρήστης με λογαριασμό τύπου ‘φοιτητή’ είναι να επισκεφτεί την σελίδα με τα έργα ανάθεσης (‘domain’/projects). Εκεί μπορεί να δει όλα τα προηγούμενα έργα ανάθεσης (Ε.Α.) που έχουν δημιουργηθεί και δείξει το ενδιαφέρον του σε κάποια από αυτά ώστε να επιλεγεί η ομάδα του για υλοποίηση.

4.3.4.1. Λειτουργία ένδειξης ενδιαφέροντος σε Ε.Α.

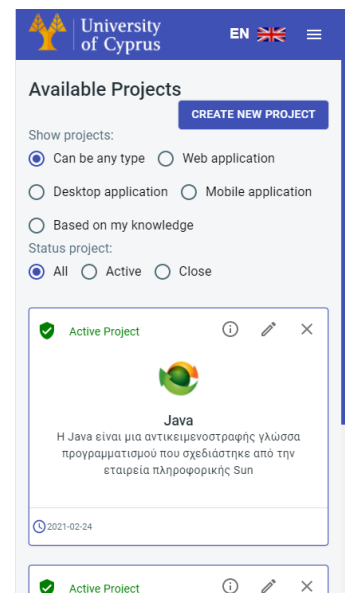
Ένας από τους λόγους που δημιουργήθηκε η πλατφόρμα ήταν ο διαμοιρασμός και η ανάθεση Ε.Α. και με την λειτουργία ένδειξης ενδιαφέροντος μπορούν οι Ο.Α.Ε. να προσθέσουν τον εαυτό τους στην λίστα που παρουσιάζεται στην Εικόνα 4.2.2.2. Απαραίτητη προϋπόθεση για να μπορεί ο φοιτητής να εκτελέσει την συγκεκριμένη λειτουργία είναι ο φοιτητής να βρίσκεται ήδη σε ομάδα που την έχει ήδη εγκρίνει χρήστης με λογαριασμό ‘φοιτητή’ και το Ε.Α. να είναι ενεργό (active). Ο χρήστης επιλέγοντας το διακόπτη κουμπί με την επιγραφή “Show interest” προσθέτει την ομάδα του στην λίστα με τις ενδιαφερόμενες ομάδες για το συγκεκριμένο Ε.Α. και από την συγκεκριμένη στιγμή και μετά κάθε μέλος της συγκεκριμένης ομάδας θα βλέπει επιλεγμένο το συγκεκριμένο διακόπτη. Αν η ομάδα αλλάξει γνώμη οποιαδήποτε στιγμή επιλέγοντας τον διακόπτη για δεύτερη φορά μπορεί να αναιρέσει το ενδιαφέρον που έδειξε η ομάδα.

4.3.4.2. Περιορισμένη εμφάνιση αποτελεσμάτων

Αρχικά όπως βλέπουμε και στην Εικόνα 4.3.4.2.1 μπορεί να επιλέξει ποια έργα να εμφανίζονται και να περιορίσει τα περιεχόμενα επιλέγοντας φίλτρο σχετικά με το είδος της εφαρμογής ή την κατάσταση του έργου. Όσο αφορά το φίλτρο με το είδος της εφαρμογής υπάρχουν 5 διαθέσιμες επιλογές, οι 3 είναι ανάλογα με το είδος που όρισε ο δημιουργός του έργου ότι προτιμάει την υλοποίηση (διαδικτυακή εφαρμογή, desktop εφαρμογή, κινητή εφαρμογή), η άλλη είναι ανάλογα με τις γνώσεις σε γλώσσες προγραμματισμού και frameworks που έχει ορίσει στα στοιχεία του προφίλ του (επεξηγείται στην ενότητα 4.3.7.1) και η τελευταία εμφανίζει όλα τα είδη έργου ανάθεσης ανεξαιρέτως κατηγορίας. Σχετικά με το φίλτρο σχετικά με την κατάσταση έργου υπάρχουν 3 επιλογές, η πρώτη εμφανίζει όλα τα έργα και των 2 κατηγοριών, η δεύτερη εμφανίζει μόνο τα ενεργά έργα (δηλαδή αυτά που έχει την δυνατότητα ο φοιτητής να τα επιλέξει για υλοποίηση) και η τρίτη επιλογή εμφανίζει τα έργα που έχουν ήδη λήξη (και ο φοιτητής δεν έχει την δυνατότητα να τα επιλέξει).



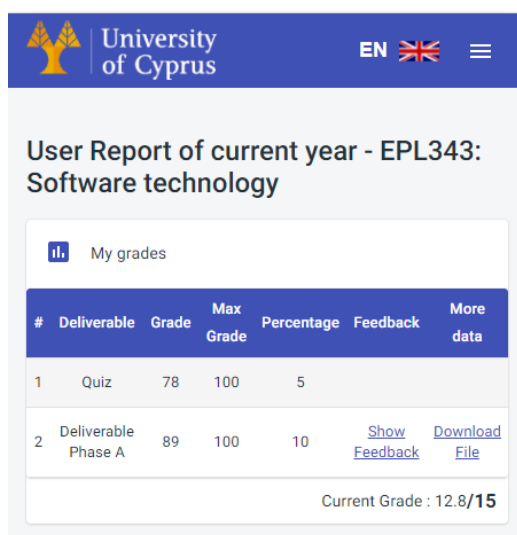
Εικόνα 4.3.3. 2: Λειτουργία ένδειξης ενδιαφέροντος Ε.Α.



Εικόνα 4.3.4.2. 8: Προβολή έργων ανάθεσης

4.3.5. Βαθμοί

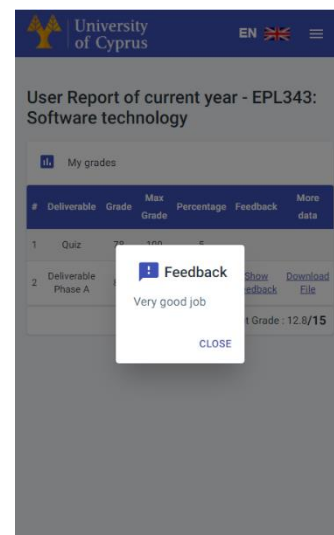
Ένας φοιτητής μπορεί να επισκεφτεί την σελίδα με διεύθυνση ‘domain’/grades στην πλατφόρμα μας για να δει ανά πάσα στιγμή τους βαθμούς του για το τρέχον ακαδημαϊκό έτος για το μάθημα EPL343 – Software Engineering μαζί με συμπληρωματικές πληροφορίες όπως μπορείτε να αντικρίσετε και στην Εικόνα 4.3.5.1. Αρχικά από τα αριστερά στα δεξιά εμφανίζεται ο αύξων αριθμός, το όνομα του παραδοτέου, ο βαθμός που πήρε ο φοιτητής, ο μέγιστος βαθμός για το συγκεκριμένο παραδοτέο, το ποσοστό που λειτουργεί ως αντίκτυπο στο τελικό βαθμό του φοιτητή, προαιρετική ανατροφοδότηση από τον καθηγητή για το συγκεκριμένο παραδοτέο και τέλος ένα αρχείο με επιπλέον πληροφορίες που μπορεί να υπάρχει προαιρετικά. Το συγκεκριμένο αρχείο ο φοιτητής, επιλέγοντας τον υπερσύνδεσμο με το ‘Download file’, έχει την δυνατότητα να το διαβάσει και να το αποθηκεύσει στην ηλεκτρονική του συσκευή ενώ για την προαιρετική ανατροφοδότηση επιλέγοντας τον υπερσύνδεσμο ‘Show Feedback’ εμφανίζεται ένα αναδυόμενο παράθυρο όπως φαίνεται και στην Εικόνα 4.3.5.2. το οποίο το περιέχει. Τέλος στο υποσέλιδο του πίνακα υπάρχει ο συνολικός βαθμός του φοιτητή μέχρι την συγκεκριμένη χρονική στιγμή ανάλογα με τον αριθμό των παραδοτέων που έχουν βαθμολογηθεί μέχρι στιγμής, δηλαδή στα δεξιά της αναλογίας είναι ο μέγιστος αριθμός των μονάδων που μπορούσε ο φοιτητής να είχε αποκτήσει ενώ αριστερά αυτές που έχει αποκτήσει ανάλογα με την βαθμολογία του.



#	Deliverable	Grade	Max Grade	Percentage	Feedback	More data
1	Quiz	78	100	5		
2	Deliverable Phase A	89	100	10	Show Feedback	Download File

Current Grade : 12.8/15

Εικόνα 4.3.5.1: Στιγμιότυπο οθόνης εμφάνισης βαθμών φοιτητή



Εικόνα 4.3.5. 2: Εμφάνιση ανατροφοδότησης

4.3.6. Ομάδες Φοιτητών

Οι φοιτητές καλούνται να δημιουργήσουν ομάδες μέχρι 7 – 8 άτομα ανάλογα με τον αριθμό των φοιτητών που θα ορίσει ο καθηγητής για την συγκεκριμένη ακαδημαϊκή χρονιά από την σελίδα της πλατφόρμας που βρίσκεται στην διεύθυνση ‘domain’/app/team. Κάθε φοιτητής ο

ο οποίος βρίσκεται σε κάποια ομάδα. Οι ομάδες για να είναι έγκυρες πρέπει ένας καθηγητής με τις επιβεβαιώσει λειτουργία που περιγράφεται στην ενότητα 4.4.6.4.

4.3.6.1. Προβολή ομάδας και ενέργειες ομάδας

Αρχικά όταν ο φοιτητής δεν βρίσκεται σε κάποια ομάδα η σελίδα εμφανίζεται όπως στην Εικόνα 4.3.6.5, δηλαδή χωρίς τον πίνακα (στο κάτω μέρος της σελίδας) ο οποίος φαίνεται στην Εικόνα 4.3.6.1. Ο πίνακας αυτός αρχικά στο πάνω μέρος εμφανίζει το αναγνωριστικό της ομάδας, δηλαδή τον μοναδικό αριθμό (για παράδειγμα στην συγκεκριμένη περίπτωση είναι ο αριθμός 5), στην συνέχεια εμφανίζει τα μέλη οποία είναι εγγεγραμμένα στην ομάδα μαζί με κάποιες επιπλέον πληροφορίες (όνομα χρήστη, ονοματεπώνυμο και γνώσεις που ίσως έχει και ανάρτησε στο προφίλ του στην ιστοσελίδα) και τέλος το υποσέλιδο της ομάδας με κάποιες ενέργειες. Στα αριστερά του υποσέλιδου υπάρχει το πλήθος των ατόμων που είναι ήδη εγγεγραμμένα στην ομάδα σε αναλογία με τον μέγιστο αριθμό ατόμων που έχει ορίσει ο καθηγητής σε σχέση με το τρέχον ακαδημαϊκό έτος. Στα δεξιά του υποσέλιδου υπάρχει η δυνατότητα ανά πάσα στιγμή (πριν από την έγκριση του καθηγητή η οποία περιγράφεται στην ενότητα 4.4.6.4) να αποχωρήσει ανά πάσα χρονική στιγμή επιλέγοντας τον υπερσύνδεσμο “LEAVE THE TEAM”, τότε θα εμφανιστεί ένα αναδυόμενο παράθυρο όπως αυτό φαίνεται στην Εικόνα 4.3.6.4 για να επιβεβαιώσει την απόφαση του φοιτητή. Το αναδυόμενο αυτό παράθυρο υπάρχει στην περίπτωση που έγινε αυτή η κίνηση κατά λάθος ο συγκεκριμένος υπερσύνδεσμος και να αποτρέψει λανθασμένες κινήσεις.

4.3.6.2. Αποστολή αιτήματος σε ομάδες για συνένωση

Φοιτητές οι οποίοι δεν έχουν ήδη δημιουργήσει ομάδα στο μάθημα μπορούν να στείλουν αίτημα σε όλες τις άλλες ομάδες για να γνωστοποιήσουν την θέση τους σχετικά με την διαθεσιμότητα τους. Οι ομάδες αυτές αν έχουν ελεύθερη θέση στην ομάδα και χρειάζονται επιπλέον μέλος θα είναι σε θέση να αποδεχτούν την πρόταση από τον συγκεκριμένο φοιτητή. Το αίτημα αυτό αποστέλνεται όταν ο φοιτητής επιλέξει τον υπερσύνδεσμο “Send request” όπως βλέπουμε και στην Εικόνα 4.3.6.2, έχει επίσης την δυνατότητα να ακυρώσει το αίτημα του αυτό επιλέγοντας ξανά το συγκεκριμένο υπερσύνδεσμο ο οποίος πλέον θα γράφει “Cancel request” και κατά συνέπεια ακυρώνεται το προηγούμενο αίτημα.

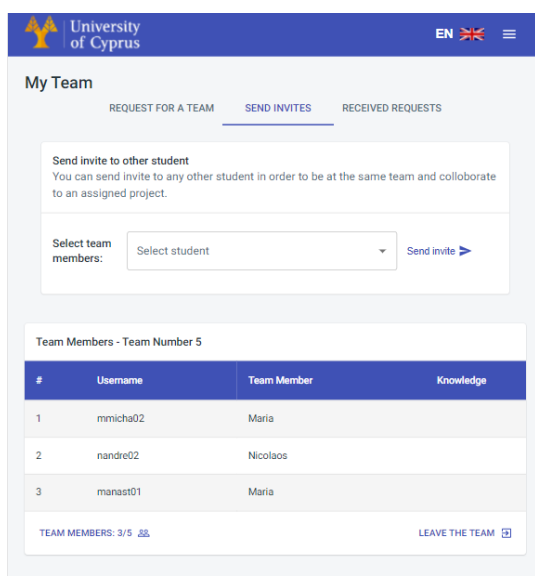
4.3.6.3. Αποστολή αιτήματος σε συγκεκριμένο άτομο για συνένωση

Μία ακόμη δυνατότητα η οποία δίνεται στους φοιτητές είναι να προσκαλέσουν τους συμμαθητές τους να έλθουν στην ίδια ομάδα με αυτούς και να είναι συνεργάτες στο Ε.Α. το οποίο θα πρέπει να υλοποιήσουν. Σε περίπτωση που ο φοιτητής δεν βρίσκεται σε κάποια ομάδα προηγουμένως και προσκαλέσει κάποιο άλλο φοιτητή τότε αυτόματα δημιουργείται μία καινούρια ομάδα και ο συγκεκριμένος χρήστης ο οποίος έστειλε το αίτημα είναι πλέον μέλος

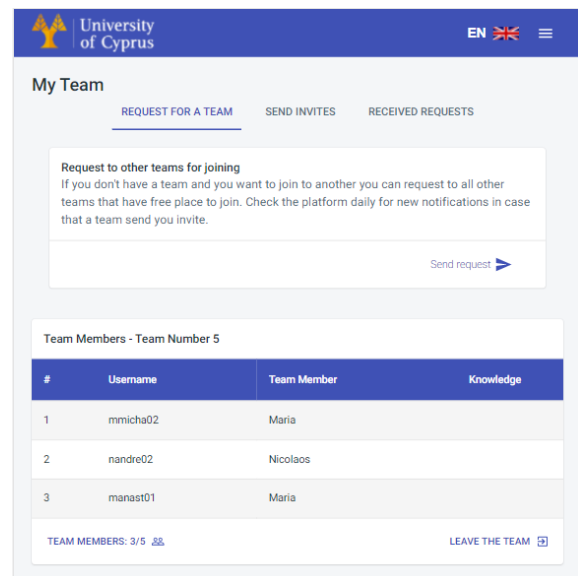
της για να μπορεί και ο παραλήπτης του αιτήματος να εισέλθει στην ομάδα. Σε αντίθετη περίπτωση αν ο φοιτητής είναι ήδη σε κάποια ομάδα το αίτημα θα σταλεί κανονικά, μαζί με τον αριθμό της ομάδας που ο αποστολέας του αιτήματος είναι ήδη μέλος, στο άτομο που ο χρήστης έχει επιλέξει από το dropdown μενού (υπάρχουν αναρτημένα τα username όλων των συμφοιτητών του συγκεκριμένου χρήστη). Τέλος επιλέγει τον υπερσύνδεσμο “Send invite” και το αίτημα αποστέλλεται όπως φαίνεται και στην Εικόνα 4.3.6.1.

4.3.6.4. Αιτήματα που έχουν ήδη ληφθεί

Υπάρχουν δύο πιθανά ήδη αιτημάτων που μπορεί κάποιος φοιτητής να έχει παραλάβει από κάποιο άλλο φοιτητή όπως φαίνονται και στην Εικόνα 4.3.6.3. Το πρώτο είδος είναι αυτό που περιγράφεται στην ενότητα 4.3.6.3. - δηλαδή αποστολή αιτημάτων σε συγκεκριμένο άτομο και ο παραλήπτης έχει δύο πιθανές επιλογές την αποδοχή ή την απόρριψη της πρότασης. Αν η πρόταση απορριφτεί δεν θα εμφανίζεται πλέον στα αιτήματα που έχει δεχτεί ο συγκεκριμένος χρήστης αντιθέτως αν ο παραλήπτης επιλέξει αποδοχή τότε μία σειρά από ελέγχων θα εκτελεστεί πριν την μεταφορά του χρήστη στην συγκεκριμένη ομάδα. Αρχικά, η πλατφόρμα θα ελέγξει αν υπάρχει κενός θέση στην ομάδα που έγινε το αίτημα, δηλαδή δεν έχει ξεπεραστεί ο μέγιστος αριθμός που όρισε ο καθηγητής για το τρέχον ακαδημαϊκό έτος, στην συνέχεια θα γίνει έλεγχος αν ο φοιτητής δεν είναι ήδη μέλος σε κάποια ομάδα και θα εμφανίσει τα ανάλογα μηνύματα επιτυχημένης ή αποτυχημένης προσθήκης. Το δεύτερο είδος μηνύματος το οποίο περιγράφεται στην ενότητα 4.3.6.2. - δηλαδή αποστολή αιτημάτων σε όλες τις ομάδες λειτουργεί μόνο σαν ενημερωτικού χαρακτήρα μήνυμα. Συγκεκριμένα, τα άτομα τα οποία βρίσκονται σε ομάδες θα βλέπουν το συγκεκριμένο μήνυμα και αν επιθυμούν να εντάξουν το συγκεκριμένο άτομο στην ομάδα τους χρησιμοποιώντας την λειτουργία που περιγράφεται στην ενότητα 4.3.6.3. θα στείλει αίτημα στο όνομα χρήστη ‘username’ το οποίο αναφέρεται στο μήνυμα. Υπάρχει και η δυνατότητα ο παραλήπτης να επιλέξει την επιλογή “Close” και το

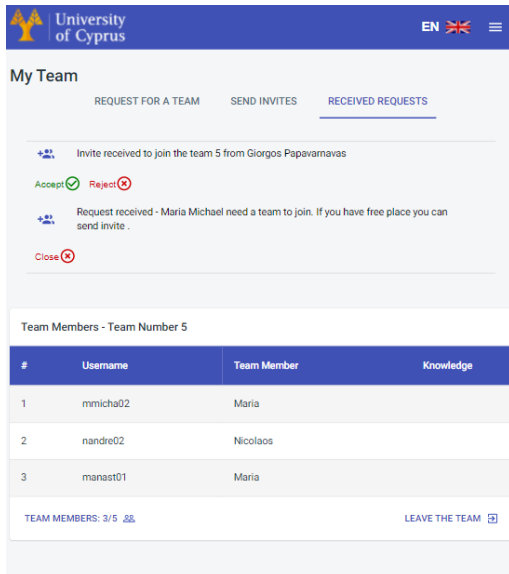


Εικόνα 4.3.6. 1: Αποστολή αιτήματος σε άτομα για συνένωση

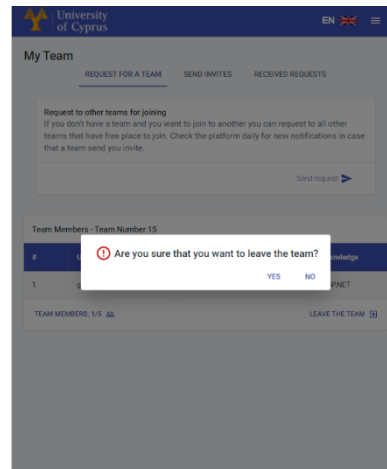


Εικόνα 4.3.6. 2: Αποστολή αιτήματος σε ομάδα για συνένωση

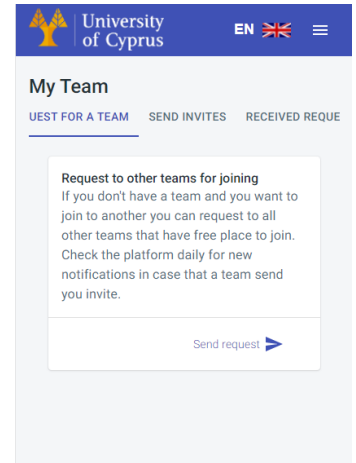
ενημερωτικό μήνυμα να μην εμφανίζεται πλέον είτε επειδή η ομάδα είναι πλήρης είτε επειδή δεν θέλουν να εντάξουν το συγκεκριμένο άτομο στο δυναμικό της ομάδας.



Εικόνα 4.3.6. 3: Προβολή αιτημάτων



Εικόνα 4.3.6. 4: Αναχώρηση από την ομάδα

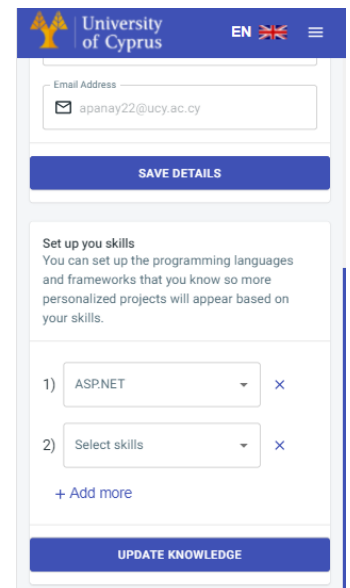


Εικόνα 4.3.6. 5: Προβολή σελίδας χωρίς ομάδα

4.3.7. Προφίλ

Κάθε χρήστης έχει το δικαίωμα αν επισκεφτεί την σελίδα προφίλ, η οποία βρίσκεται στην υποσελίδα ('domain'/app/account), να επεξεργαστεί και να αλλάξει τα στοιχεία που αφορούν το όνομα το επώνυμο του και το ηλεκτρονικό του ταχυδρομείο όπως περιγράφονται και στην ενότητα 4.1.8 για όλους τους χρήστες.

Επιπλέον, οι χρήστες με τύπο λογαριασμού 'φοιτητή' έχουν την δυνατότητα να προσθέσουν γνώσεις σε προγραμματιστικές γλώσσες και frameworks που πιθανόν να έχουν έτσι ώστε η ιστοσελίδα να αποκτήσει εξειδίκευση στο συγκεκριμένο χρήστη. Μία χρήση της πληροφορίας αυτής είναι για παράδειγμα στην σελίδα των Ε.Α. που περιγράφεται στην ενότητα 4.3.4. και συγκεκριμένα στην εμφάνιση περιορισμένων αποτελεσμάτων ανάλογα με τις γνώσεις του χρήστη. Υπάρχει μια αρκετά μεγάλη λίστα (Εικόνα 4.3.7.2) με προγραμματιστικές γλώσσες και frameworks και όλα είναι συνδεδεμένα με τις τρεις πιθανές κατηγορίες (διαδικτυακή εφαρμογή, desktop εφαρμογή και κινητή εφαρμογή) έτσι ανάλογα με



Εικόνα 4.3.7. 1: Προσθήκη γνώσεων και framework

```
const options = ["ASP.NET", "ActionScript", "Adobe PhoneGap", "Angular", "Apache Groovy", "Appcelerator Titanium", "C Windows API", "C#", "C#(.NET)", "C++", "CSS", "Clojure", "CodeIgniter", "CoffeeScript", "Corona SDK", "Corona", "Dart", "Delphi", "Django", "Electron", "Elixir", "Embarcadero Delphi", "Ember", "Express", "F#", "Flutter", "Framework 7", "Go", "GoLang/Go", "HTL", "HTML5", "Haskell", "Intel XDK", "Ionic", "jQuery Mobile", "Java", "JavaFX", "JavaScript", "Kotlin", "Laravel", "Lua", "METEOR", "Mobile Angular UI", "Native Script", "NativeScript", "ObjectPascal", "Objective-C", "Ocaml", "PHP", "PLAY", "Perl", "PhoneGap", "Processing", "Python", "QML", "Qt", "R", "React Native", "React", "Ruby on Rails", "Ruby", "Rust", "SQL", "Scala", "Scheme", "Scratch", "Sencha Touch", "Spring", "Svelte", "Swift", "Swiftic", "Swing", "Symfony", "TypeScript", "UWP", "VB.Net", "Vue", "WPF", "WinForms", "Xamarin", "jQuery"];
```

Εικόνα 4.3.7. 2: Επιλογές προγραμματιστικών γλωσσών και framework

τις γνώσεις του φοιτητή να αναγνωρίζει ανάλογα την προτίμηση του σε μία ή περισσότερες κατηγορίες εφαρμογής.

4.3.7.1. Προσθήκη και αφαίρεση γνώσεων

Οι φοιτητές έχουν την δυνατότητα να προσθέσουν άπειρα πεδία γνώσεων στο προφίλ τους επιλέγοντας τον υπερσύνδεσμο “Add more” όπως φαίνεται και στην εικόνα 4.3.7.1. για να εμφανιστεί κενό πεδίο το οποίο στην συνέχεια αφού ο χρήστης επιλέξει από την dropdown λίστα επιλογών θα δείχνει την επιλογή του. Οποιαδήποτε στιγμή επιθυμεί σε περίπτωση λανθασμένης επιλογής μπορεί ο φοιτητής να επεξεργαστεί την επιλογή του επιλέγοντας κάτι διαφορετικό από την λίστα. Υπάρχει επίσης η δυνατότητα επιλέγοντας το εικονίδιο ‘X’ στα δεξιά κάθε επιλογής να διαγράψει την συγκεκριμένη επιλογή και να μην υπάρχει καθόλου. Τέλος για να αποθηκευτούν οι αλλαγές ο χρήστης πρέπει να επιλέξει τον υπερσύνδεσμο “UPDATE KNOWLEDGE” και τότε γίνεται η ανανέωση στην Β.Δ. και στην ιστοσελίδα με ένα μήνυμα επιβεβαίωσης για την αλλαγή.

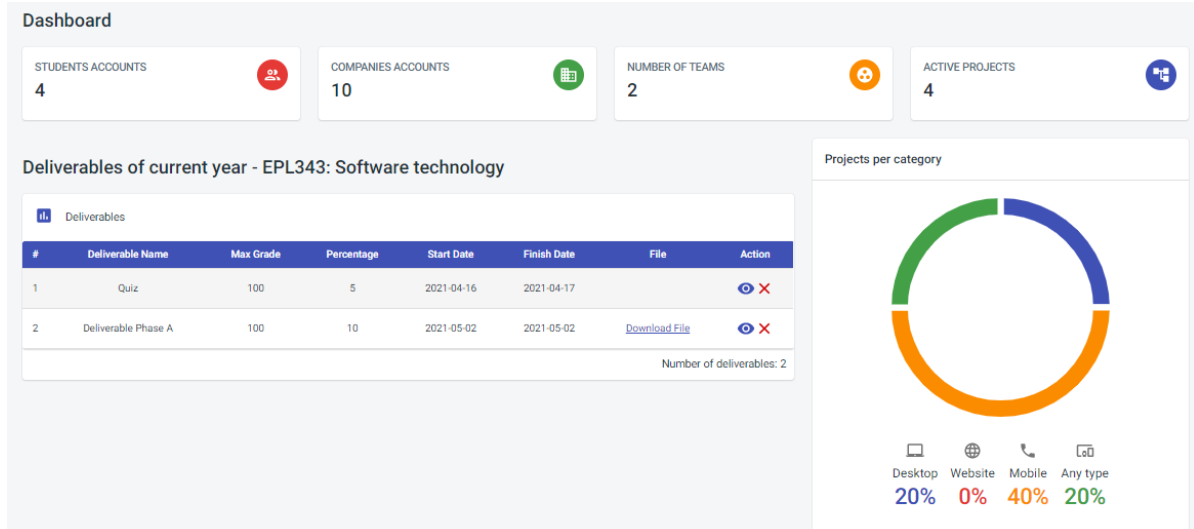
4.4.Διεπαφή Καθηγητή

4.4.1. Κεντρική Σελίδα

Η κεντρική σελίδα η οποία κατευθύνεται ο χρήστης με τύπο λογαριασμού Καθηγητή φαίνεται στην Εικόνα 4.4.1.1. Αρχικά μπορεί να δει κάποιες πληροφορίες σχετικά για το τρέχον ακαδημαϊκό έτος όπως για παράδειγμα ο αριθμός φοιτητών που είναι εγγεγραμμένοι για την φετινή ακαδημαϊκή χρονιά, τον συνολικό πλήθος ενεργών χρηστών με τύπο λογαριασμό ‘οργανισμού’ που μπορούν να αναρτήσουν Ε.Α. , τον αριθμό των ομάδων που δημιουργήθηκαν και τα ενεργά Ε.Α. που μπορεί κάποια ΟΑΕ να δείξει ενδιαφέρον για υλοποίηση. Αρχικά από τα αριστερά στα δεξιά εμφανίζεται ο αύξων αριθμός, το όνομα του παραδοτέου, ο μέγιστος βαθμός για το συγκεκριμένο παραδοτέο, το ποσοστό που λειτουργεί ως αντίκτυπο στο τελικό βαθμό του φοιτητή, το εύρος των ημερομηνιών που είναι ισχύει το συγκεκριμένο παραδοτέο (αρχική και τελική ημερομηνία), ένα αρχείο με επιπλέον πληροφορίες που μπορεί να υπάρχει προαιρετικά και τέλος κάποιες ενέργειες που μπορεί να εκτελέσει ο καθηγητής. Το συγκεκριμένο αρχείο ο χρήστης επιλέγοντας τον υπερσύνδεσμο με το ‘Download file’ έχει την δυνατότητα να το διαβάσει και να το αποθηκεύσει στην ηλεκτρονική του συσκευή. Οι ενέργειες που έχει την δυνατότητα να εκτελέσει ένας καθηγητής είναι να ορίσει αν το παραδοτέο θα είναι προσβάσιμο ή όχι στους χρήστες με το εικονίδιο ‘ματάκι’ στα αριστερά και να διαγράψει το συγκεκριμένο παραδοτέο με το εικονίδιο ‘X’ στα δεξιά, οι λειτουργίες περιγράφονται καλύτερα στην Ενότητα 4.4.2. Στην συγκεκριμένη σελίδα εμφανίζονται όλα τα παραδοτέα τα οποία κάθε χρήστης με τύπο λογαριασμού καθηγητή έχει

επιλέξει δημιουργήσει. Τέλος στο υποσέλιδο του πίνακα υπάρχει συνολικά το πλήθος των παραδοτέων σε περίπτωση που είναι μεγάλος αριθμός να ξεχωρίζει αμέσως ο αριθμός. Υπάρχει στην αρχική σελίδα γιατί είναι μία αρκετά σημαντική πληροφορία και δίνει άμεση πρόσβαση στην σελίδα των παραδοτέων.

Συμπληρωματικά από τον αριθμό των ενεργών Ε.Α. το οποίο αναφέρεται απλά στο πλήθος τους (στο πάνω μέρος της σελίδας) υπάρχει και κυκλική γραφική παράσταση η οποία δείχνει



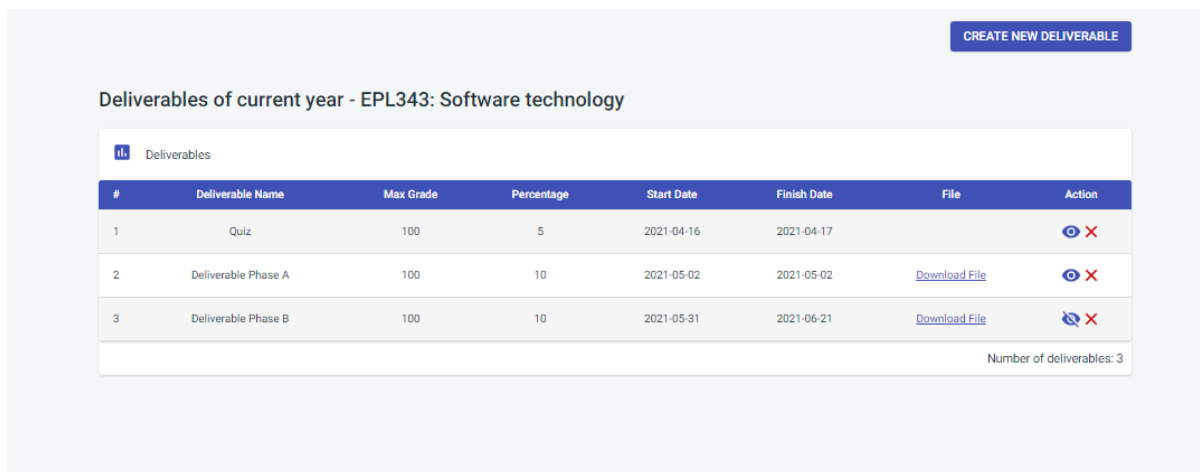
Εικόνα 4.4.1. 1: Κεντρική σελίδα λογαριασμού καθηγητή

παραστατικά την αναλογία και την προτίμηση που έχουν οι οργανισμοί στο είδος της εφαρμογής για τα έργα τους. Για παράδειγμα στο στιγμιότυπο της Εικόνας 4.1.3.4 υπάρχουν 20% desktop εφαρμογές, 0% διαδικτυακές εφαρμογές, 40% κινητές εφαρμογές και 20% οποιουδήποτε τύπου δηλαδή από τις τέσσερις συνολικά που υπάρχουν οι δύο είναι κινητές εφαρμογές, μία είναι desktop εφαρμογή, μία είναι οποιουδήποτε τύπου και δεν υπάρχει καμία διαδικτυακή εφαρμογή.

4.4.2. Παραδοτέα

Στην σελίδα των παραδοτέων η οποία βρίσκεται στην διεύθυνση 'domain'/deliverables εμφανίζει ένα πίνακα όπως φαίνεται και στην Εικόνα 4.4.2.1 στον οποίο παρουσιάζονται όλα τα παραδοτέα μαζί με συμπληρωματικές πληροφορίες όπως μπορείτε να αντικρίσετε. Αρχικά από τα αριστερά στα δεξιά εμφανίζεται ο αύξων αριθμός, το όνομα του παραδοτέου, ο μέγιστος βαθμός για το συγκεκριμένο παραδοτέο, το ποσοστό που λειτουργεί ως αντίκτυπο στο τελικό βαθμό του φοιτητή, το εύρος των ημερομηνιών που είναι ισχύει το συγκεκριμένο παραδοτέο (αρχική και τελική ημερομηνία), ένα αρχείο με επιπλέον πληροφορίες που μπορεί να υπάρχει προαιρετικά και τέλος κάποιες ενέργειες που μπορεί να εκτελέσει ο καθηγητής. Το συγκεκριμένο αρχείο ο χρήστης επιλέγοντας τον υπερσύνδεσμο με το 'Download file' έχει την

δυνατότητα να το διαβάσει και να το αποθηκεύσει στην ηλεκτρονική του συσκευή. Στην συγκεκριμένη σελίδα εμφανίζονται όλα τα παραδοτέα τα οποία κάθε χρήστης με τύπο λογαριασμού καθηγητή έχει δημιουργήσει. Τέλος στο υποσέλιδο του πίνακα υπάρχει συνολικά το πλήθος των παραδοτέων σε περίπτωση που είναι μεγάλος αριθμός να ξεχωρίζει αμέσως ο αριθμός.



#	Deliverable Name	Max Grade	Percentage	Start Date	Finish Date	File	Action
1	Quiz	100	5	2021-04-16	2021-04-17		
2	Deliverable Phase A	100	10	2021-05-02	2021-05-02	Download File	
3	Deliverable Phase B	100	10	2021-05-31	2021-06-21	Download File	

Number of deliverables: 3

Εικόνα 4.4.2. 1: Στιγμιότυπο οθόνης σελίδας παραδοτέων

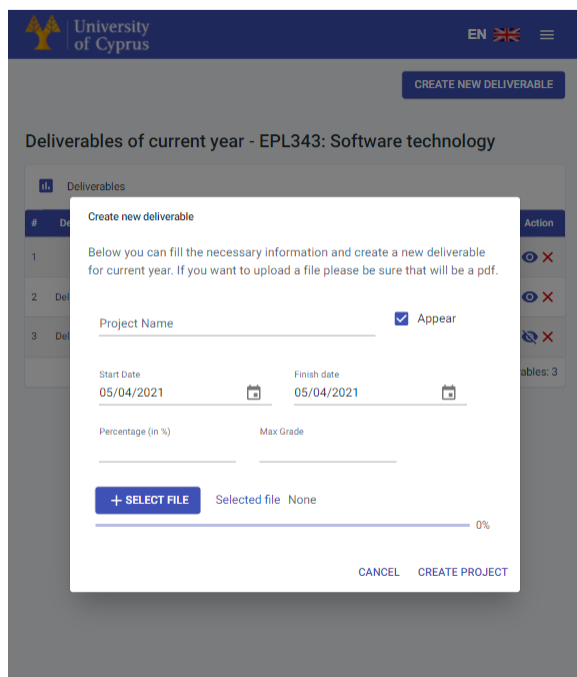
4.4.2.1. Δημιουργία καινούριου παραδοτέου

Κάθε χρήστης με λογαριασμό τύπου καθηγητή μπορεί ανά πάσα στιγμή να δημιουργήσει καινούριο παραδοτέο επιλέγοντας τον υπερσύνδεσμο “CREATE NEW DELIVERABLE” και την συμπλήρωση των απαραίτητων στοιχείων που εμφανίζονται στο αναδυόμενο παράθυρο το οποίο φαίνεται στην Εικόνα 4.4.2.2. Απαραίτητα στοιχεία για συμπλήρωση είναι το όνομα του παραδοτέου, η ρύθμιση κατά πόσο θα είναι εμφανές το παραδοτέο στους φοιτητές ή όχι, το εύρος ημερομηνιών το οποίο ο φοιτητής θα πρέπει να παραδώσει δηλαδή η ημερομηνία ανάθεσης και η ημερομηνία ανάθεσης του παραδοτέου, το ποσοστό αντίκτυπο που θα έχει η βαθμολογία του παραδοτέου στον τελικό βαθμό και ο μέγιστος βαθμός που μπορεί να βαθμολογηθεί. Προαιρετικό στοιχείο θεωρείται η μεταφόρτωση ενός αρχείου με συμπληρωματικές πληροφορίες συνήθως PDF ή WORD document. Συμπληρώνοντας τα στοιχεία αυτά ο καθηγητής θα επιλέξει τον υπερσύνδεσμο “CREATE DELIVERABLE ” για να ενημερώσει κατάλληλα την Β.Δ. και τον διακομιστή σε περίπτωση μεταφόρτωσης αρχείου. Τέλος, η σελίδα ανανεώνεται και εμφανίζεται ενημερωτικό μήνυμα στο χρήστη για επιτυχημένη ή αποτυχημένη δημιουργία παραδοτέου.

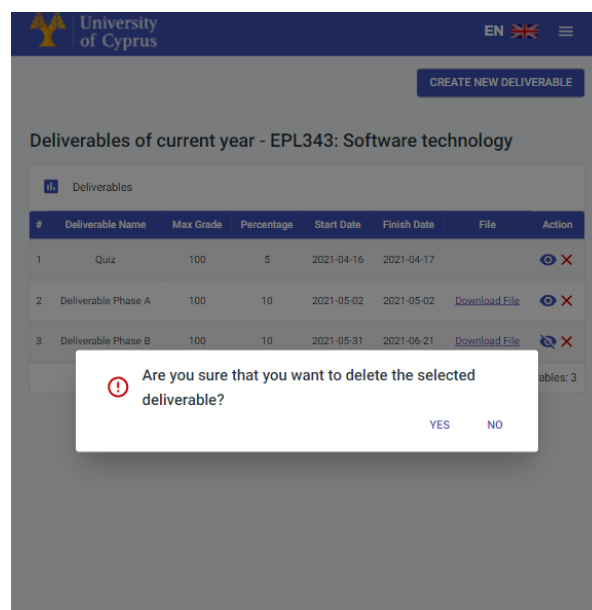
4.4.2.2. Ενέργειες διαγραφής και περιορισμού πρόσβασης

Οι ενέργειες που έχει την δυνατότητα να εκτελέσει ένας καθηγητής είναι να ορίσει αν το παραδοτέο θα είναι προσβάσιμο ή όχι στους χρήστες και να διαγράψει κάποιο από αυτά. Επιλέγοντας το εικονίδιο ‘ματάκι’ στα αριστερά τότε υπάρχουν 2 καταστάσεις τις οποίες

μπορεί να βρεθεί, η πρώτη είναι η προκαθορισμένη η οποία εμφανίζει τα παραδοτέα σε όλους και η δεύτερη με το εικονίδιο που μοιάζει σαν ‘διαγραμμένο ματάκι’ αποκρύπτει το συγκεκριμένο παραδοτέο από τους φοιτητές. Συγκρίνοντας την Εικόνα 4.4.2.1 μαζί με την Εικόνα 4.3.2.1 παρατηρούμε ότι την ίδια χρονική στιγμή υπάρχουν 2 εμφανές παραδοτέα για



Εικόνα 4.4.2. 2: Δημιουργία καινούριου παραδοτέου

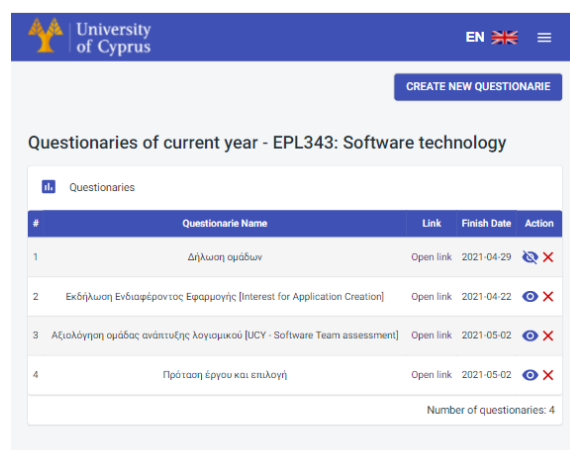


Εικόνα 4.4.2. 3: Ενημερωτικό μήνυμα διαγραφής παραδοτέου

τον χρήστη με λογαριασμό φοιτητή ενώ για τον χρήστη με λογαριασμό καθηγητή εμφανίζονται 3 γεγονός που δικαιολογείται λόγω του ότι το παραδοτέο ‘Deliverable Phase B’ έχει απενεργοποιημένη την εμφάνιση του. Όσο αφορά την λειτουργία διαγραφής ο χρήστης επιλέγει το εικονίδιο ‘X’ στα δεξιά του παραδοτέου που επιθυμεί, και εμφανίζεται αναδυόμενο μήνυμα με επιβεβαιωτικό μήνυμα, όπως φαίνεται στην Εικόνα 4.4.2.3 για να αποφευχθεί οποιαδήποτε λανθασμένη και μη ανατρέψιμη κατάσταση.

4.4.3. Ερωτηματολόγια

Στην σελίδα των ερωτηματολογίων η οποία βρίσκεται στην διεύθυνση ‘domain’/ questionnaires εμφανίζει ένα πίνακα όπως φαίνεται και στην Εικόνα 4.4.3.1 στον οποίο παρουσιάζονται όλα τα ερωτηματολόγια μαζί με συμπληρωματικές πληροφορίες όπως μπορείτε να αντικρίσετε. Αρχικά από τα αριστερά στα δεξιά εμφανίζεται ο αύξων αριθμός, το όνομα του ερωτηματολογίου, τον σύνδεσμο που μας παραπέμπει στη φόρμα του ερωτηματολογίου, την ημερομηνία της τελικής προθεσμίας που πρέπει να συμπληρωθεί από τους φοιτητές και τέλος κάποιες ενέργειες που μπορεί να



Εικόνα 4.4.3. 1: Στιγμιότυπο σελίδας ερωτηματολογίων

εκτελέσει ο καθηγητής.. Όταν ο χρήστης επιλέξει τον υπερσύνδεσμο “Open Link” τον παραπέμπει στην ηλεκτρονική φόρμα συμπλήρωσης του ερωτηματολογίου (συνήθως στο Google Forms). Στην συγκεκριμένη σελίδα εμφανίζονται όλα τα ερωτηματολόγια τα οποία κάθε χρήστης με τύπο λογαριασμού καθηγητή έχει δημιουργήσει. Τέλος στο υποσέλιδο του πίνακα υπάρχει συνολικά το πλήθος των ερωτηματολογίων σε περίπτωση που είναι μεγάλος αριθμός να ξεχωρίζει αμέσως ο αριθμός.

4.4.3.1. Δημιουργία καινούριου ερωτηματολογίου

Κάθε χρήστης με λογαριασμό τύπου καθηγητή μπορεί ανά πάσα στιγμή να δημιουργήσει καινούριο παραδοτέο επιλέγοντας τον υπερσύνδεσμο “CREATE NEW QUESTIONARIE” και την συμπλήρωση των απαραίτητων στοιχείων που εμφανίζονται στο αναδυόμενο παράθυρο το οποίο φαίνεται στην Εικόνα 4.4.3.3. Απαραίτητα στοιχεία για συμπλήρωση είναι το όνομα του ερωτηματολογίου, η ρύθμιση κατά πόσο θα είναι εμφανές το ερωτηματολόγιο στους φοιτητές ή όχι, η τελική ημερομηνία συμπλήρωσης του και ο σύνδεσμος ο οποίος θα παραπέμπει στο ερωτηματολόγιο (τις περισσότερες φορές θα είναι σύνδεσμος που παραπέμπει στο google forms). Συμπληρώνοντας τα στοιχεία αυτά ο καθηγητής θα επιλέξει τον υπερσύνδεσμο “CREATE QUESTIONARIE ” για να ενημερώσει κατάλληλα την Β.Δ. και τέλος, η σελίδα ανανεώνεται και εμφανίζεται ενημερωτικό μήνυμα στο χρήστη για επιτυχημένη ή αποτυχημένη δημιουργία συνδέσμου για ερωτηματολόγιο.

4.4.3.2. Ενέργειες διαγραφής και περιορισμού πρόσβασης

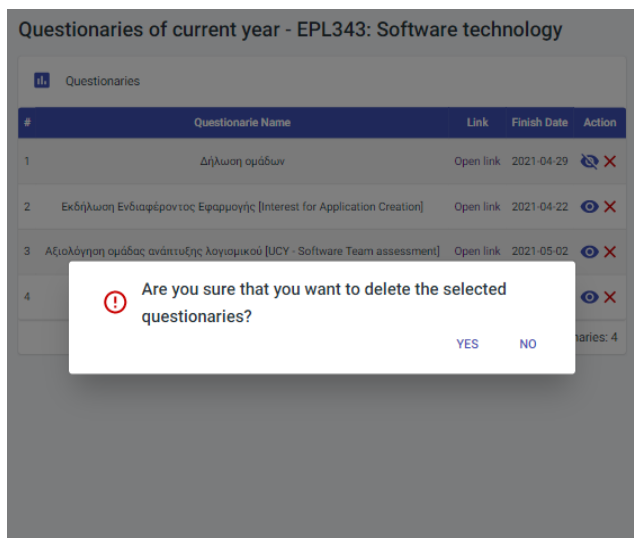
Οι ενέργειες που έχει την δυνατότητα να εκτελέσει ένας καθηγητής είναι να ορίσει αν το ερωτηματολόγιο θα είναι προσβάσιμο ή όχι στους χρήστες και να διαγράψει κάποιο από αυτά. Επιλέγοντας το εικονίδιο ‘ματάκι’ στα αριστερά τότε υπάρχουν 2 καταστάσεις τις οποίες μπορεί να βρεθεί, η πρώτη είναι η προκαθορισμένη η οποία εμφανίζει τα παραδοτέα σε όλους και η δεύτερη με το εικονίδιο που μοιάζει σαν ‘διαγραμμένο ματάκι’ αποκρύπτει το συγκεκριμένο ερωτηματολόγιο από τους φοιτητές. Συγκρίνοντας την Εικόνα 4.4.3.1 μαζί με την Εικόνα 4.3.3.1 παρατηρούμε ότι την ίδια χρονική στιγμή υπάρχουν 3 εμφανές ερωτηματολόγια για τον χρήστη με λογαριασμό φοιτητή ενώ για τον χρήστη με λογαριασμό καθηγητή εμφανίζονται 4 γεγονός που δικαιολογείται λόγο του ότι το ερωτηματολόγιο ‘Δήλωση ομάδων’ έχει απενεργοποιημένη την εμφάνιση του. Όσο αφορά την λειτουργία διαγραφής ο χρήστης επιλέγει το εικονίδιο ‘X’ στα δεξιά του ερωτηματολογίου που επιθυμεί,

και εμφανίζεται αναδυόμενο μήνυμα με επιβεβαιωτικό μήνυμα, όπως φαίνεται στην Εικόνα 4.4.3.2 για να αποφευχθεί οποιαδήποτε λανθασμένη και μη ανατρέψιμη κατάσταση.

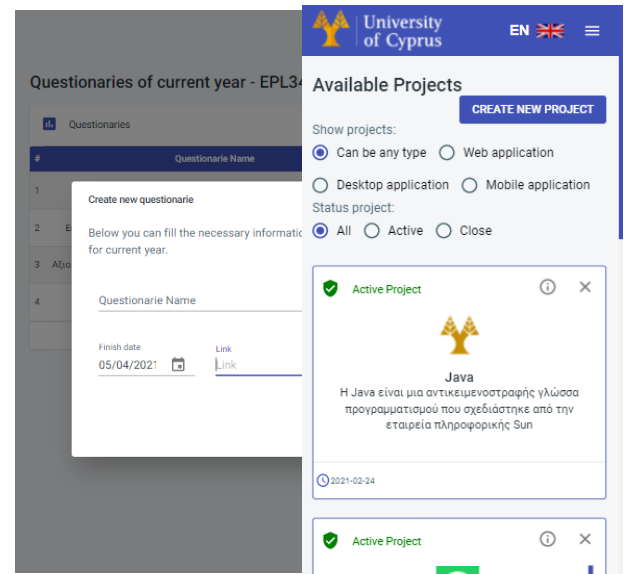
4.4.4. Έργα Ανάθεσης

Μία από της πιο σημαντικές λειτουργίες που μπορεί να εκτελέσει ένας χρήστης με λογαριασμό τύπου καθηγητή είναι να επισκεφτεί την σελίδα με τα έργα ανάθεσης (‘domain’/projects). Εκεί μπορεί να δει όλα τα προηγούμενα έργα ανάθεσης (Ε.Α.) που δημιούργησαν όλοι οι χρήστες (καθηγητές και οργανισμοί) και να ανεβάσει καινούρια.

4.4.4.1. Περιορισμένη εμφάνιση αποτελεσμάτων



Εικόνα 4.4.3. 2: Ενημερωτικό μήνυμα για διαγραφή ερωτηματολογίου



Εικόνα 4.4.3. 3: Δημιουργία υπερσυνδέσμου για Εικόνα 4.4.4. 9: Προβολή έργων ανάθεσης

Αρχικά όπως βλέπουμε και στην Εικόνα 4.4.4.1 μπορεί να επιλέξει ποια έργα να εμφανίζονται και να περιορίσει τα περιεχόμενα επιλέγοντας φίλτρο σχετικά με το είδος της εφαρμογής ή την κατάσταση του έργου. Όσο αφορά το φίλτρο με το είδος της εφαρμογής υπάρχουν 4 διαθέσιμες επιλογές, οι 3 είναι ανάλογα με το είδος που όρισε ο δημιουργός του έργου ότι προτιμάει την υλοποίηση (διαδικτυακή εφαρμογή, desktop εφαρμογή, κινητή εφαρμογή) και η άλλη εμφανίζει όλα τα είδη έργου ανάθεσης ανεξαιρέτως κατηγορίας. Σχετικά με το φίλτρο σχετικά με την κατάσταση έργου υπάρχουν 3 επιλογές, η πρώτη εμφανίζει όλα τα έργα και των 2 κατηγοριών, η δεύτερη εμφανίζει μόνο τα ενεργά έργα (δηλαδή αυτά που έχει την δυνατότητα ο φοιτητής να τα επιλέξει για υλοποίηση) και η τρίτη επιλογή εμφανίζει τα έργα που έχουν ήδη λήξει (και ο φοιτητής δεν έχει την δυνατότητα να τα επιλέξει).

4.4.4.2. Δημιουργία καινούριου Ε.Α.

Χρήστες με λογαριασμό καθηγητή έχουν την δυνατότητα να δημιουργήσουν άπειρα Ε.Α. αρκεί να επιλέξουν τον υπερσύνδεσμο “CREATE NEW PROJECT” για να εμφανιστεί καινούριο αναδυόμενο παράθυρο όπως φαίνεται στην Εικόνα 4.4.4.3. Ο χρήστης έτσι ώστε να δημιουργήσει επιτυχημένα ένα καινούριο Ε.Α. καλείτε να συμπληρώσει ένα αντιπροσωπευτικό όνομα μαζί με μία περιγραφή, να επιλέξει τον τύπο της εφαρμογής (διαδικτυακή εφαρμογή, desktop εφαρμογή, κινητή εφαρμογή, οποιουδήποτε τύπου, αν δεν έχει αποφασιστεί ακόμη) και προαιρετικά αν θέλει να επιλέξει ένα αρχείο PDF με επιπλέον πληροφορίες. Στην συνέχεια επιλέγει τον υπερσύνδεσμο “CREATE PROJECT”, γίνεται ο έλεγχος για την συμπλήρωση όλων των πεδίων, μεταφορτώνεται το αρχείο περιγραφής στον διακομιστή, γίνεται η εγγραφή στην Β.Δ. και τέλος εμφανίζεται ενημερωτικό μήνυμα για την κατάσταση δημιουργίας του Ε.Α.

4.4.4.3. Επεξεργασία στοιχείων Ε.Α.

Οποιαδήποτε στιγμή επιθυμεί ο δημιουργός ενός συγκεκριμένου Ε.Α. έχει την δυνατότητα να επεξεργαστεί της πληροφορίες του. Συγκεκριμένα μπορεί να ενημερώσει ίσως τον τίτλο ή την περιγραφή του Ε.Α., να προσθέσει ή να αφαιρέσει υλικό (για παράδειγμα αρχείο με επιπλέον πληροφορίες PDF) ή να αλλάξει την κατάσταση από ενεργό έργο σε κλειστό. Αυτό μπορεί να το πετύχει επιλέγοντας το εικονίδιο με το μολυβί που αντιστοιχεί στην επεξεργασία όπως φαίνεται και στην Εικόνα 4.4.4.1. Εκτελούνται οι κατάλληλοι έλεγχοι για ορθή ενημέρωση των στοιχείων και γίνεται η αλλαγή τόσο στον διακομιστή για πιθανή ενημέρωση του PDF αρχείου με τις περισσότερες πληροφορίες όσο και στην Β.Δ.

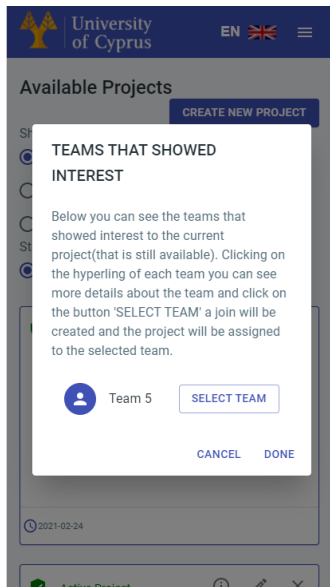
4.4.4.4. Προβολή ενδιαφερόμενων ομάδων για συγκεκριμένο Ε.Α.

Όλοι οι φοιτητές που βρίσκονται σε ομάδα που έχει εγκριθεί από κάποιο καθηγητή έχουν την δυνατότητα να δείξουν ενδιαφέρον για να αναλάβουν το συγκεκριμένο έργο και να το υλοποιήσουν. Ο χρήστης ο οποίος δημιούργησε το συγκεκριμένο Ε.Α. αλλά και κάθε χρήστης με λογαριασμό καθηγητή μπορεί να δει όλες τις ομάδες οι οποίες έδειξαν ενδιαφέρον (επιλέγοντας το εικονίδιο της πληροφορίας όπως φαίνεται στην Εικόνα 4.4.4.1) και κάποιες επιπλέον πληροφορίες γι’ αυτούς, για παράδειγμα από τον αριθμό των μελών που απαρτίζουν την ομάδα, ηλεκτρονικό ταχυδρομείο επικοινωνίας και τις προηγούμενες γνώσεις που έχουν τα άτομα αυτά όπως φαίνεται στην Εικόνα 4.4.4.5. Επιπλέον, προσωπικά στοιχεία όπως το όνομα και το επίθετο των ατόμων που απαρτίζουν την ομάδα δεν είναι διαθέσιμα σε αρχικό στάδιο έτσι ώστε να προστατευτούν και να γίνει πιο αντικειμενική ανάθεση Ε.Α. σε ομάδες. Στην συνέχεια, μόλις ο οργανισμός αποφασίσει σε ποια ομάδα θέλει να αναθέσει το έργο επιλέγει το κουμπί με τον υπερσύνδεσμο “SELECT TEAM” (όπως φαίνεται στην ομάδα Εικόνα 4.4.4.2) και πλέον όρισε ομάδα ανάθεσης. Από αυτή την στιγμή και μετά μπορεί να δει

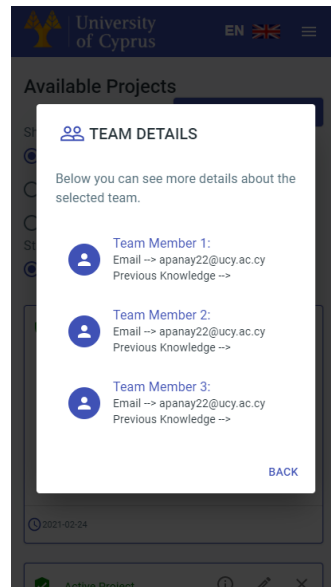
και πιο προσωπικές πληροφορίες όπως για παράδειγμα το όνομα και το επώνυμο κάθε ατόμου στην ομάδα για να μπορεί να γίνει η ταυτοποίηση ευκολότερα, όπως φαίνεται και στην Εικόνα 4.4.4.4.

4.4.4.5. Διαγραφή Ε.Α.

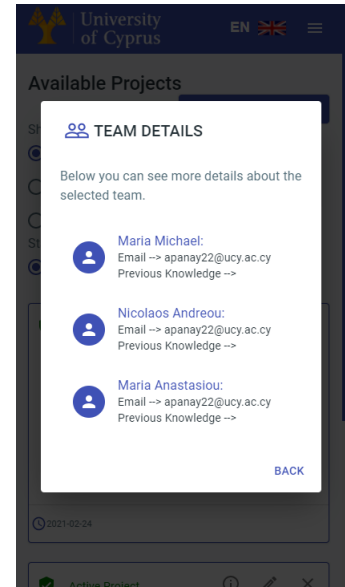
Ο δημιουργός κάθε Ε.Α. αλλά και κάθε χρήστης με τύπο λογαριασμό καθηγητή έχει την δυνατότητα ανά πάσα στιγμή να το διαγράψει επιλέγοντας το τρίτο εικονίδιο του X (στο συγκεκριμένο Ε.Α.) όπως φαίνεται στη εικόνα 4.4.4.1. Εμφανίζεται αναδυόμενο παράθυρο για να επιβεβαιωθεί ξανά η απόφαση του χρήστη να διαγράψει το συγκεκριμένο Ε.Α. όπως φαίνεται και στην Εικόνα 4.4.4.7. και να αποφευχθούν ατυχήματα από λανθασμένη επιλογή του υπερσυνδέσμου διαγραφής. Η διαγραφή συμβαίνει τόσο στην Β.Δ. για τα στοιχεία που είναι αποθηκευμένα όσο και στον διακομιστή ο οποίος περιέχει αρχείο PDF με περισσότερες πληροφορίες. Ο λόγος που κάθε χρήστης με τύπο λογαριασμό καθηγητή έχει την δυνατότητα να διαγράψει οποιοδήποτε Ε.Α. είναι η προστασία και η ανάρτηση κατάλληλων Ε.Α. για τους φοιτητές, ώστε μη κατάλληλα έργα να διαγράφονται και κάθε καθηγητής να έχει περισσότερα προνόμια από τους άλλους δύο τύπους λογαριασμών.



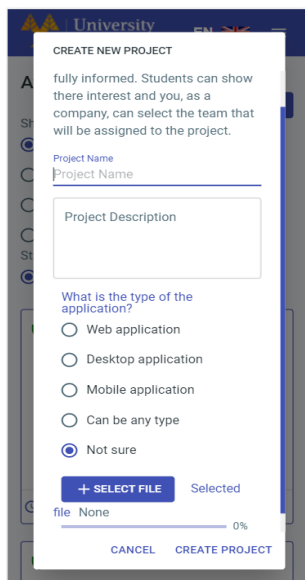
Εικόνα 4.4.4. 15: Προβολή ενδιαφερόμενων ομάδων



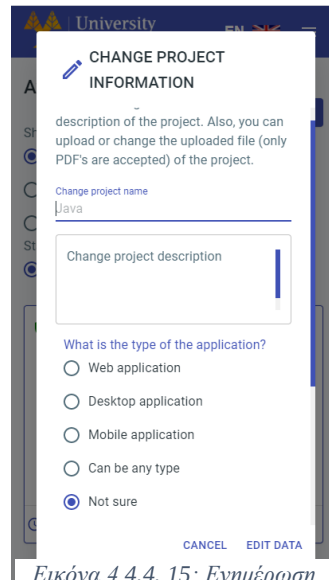
Εικόνα 4.4.4. 15: Προβολή στοιχείων ομάδας - ανώνυμα



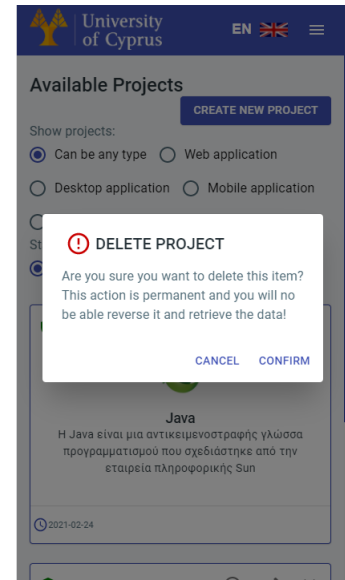
Εικόνα 4.4.4. 15: Προβολή στοιχείων ομάδας - ταυτοποιημένα



Εικόνα 4.4.4. 15: Δημιουργία έργου ανάθεσης



Εικόνα 4.4.4. 15: Ενημέρωση πληροφοριών Ε.Α.



Εικόνα 4.4.4. 15: Διαγραφή Ε.Α.

4.4.5. Βαθμοί

Ένας καθηγητής μπορεί να επισκεφτεί την σελίδα με διεύθυνση 'domain'/grades στην πλατφόρμα μας για να ανεβάσει βαθμούς για τους φοιτητές του τρέχοντος ακαδημαϊκού έτους για το μάθημα ΕΠΛ343 – Software Engineering μαζί με συμπληρωματικές πληροφορίες όπως μπορείτε να αντικρίσετε και στην Εικόνα 4.4.5.1. Οι φοιτητές έχουν παραδοτέα τα οποία παίρνουν βαθμό συνολικά από ομαδική δουλειά και τις περισσότερες φορές παίρνουν όλοι οι φοιτητές της ίδιας ομάδας τον ίδιο βαθμό ενώ κάποια άλλα παραδοτέα ή εξετάσεις ο καθένας παίρνει ξεχωριστό βαθμό, γι' αυτό τον λόγο υπάρχουν και δύο ξεχωριστές λειτουργίες ανάρτησης βαθμών όπως περιγράφονται στις ενότητες 4.4.5.1. Μόλις ο καθηγητής αναρτήσει

κάποιο βαθμό για ένα παραδοτέο θα αποσταλεί και ενημερωτικό μήνυμα στο ηλεκτρονικό ταχυδρομείο του βαθμολογημένου φοιτητή (αν έχει ενεργοποιημένη την ρύθμιση που περιγράφεται στην ενότητα 4.1.5) όπως φαίνεται και στην Εικόνα 4.4.5.4.

4.4.5.1. Ανάρτηση βαθμών για συγκεκριμένο φοιτητή

Σε περίπτωση που ο καθηγητής επιθυμεί να αναρτήσει βαθμό για ένα συγκεκριμένο φοιτητή χρησιμοποιεί την πρώτη λειτουργία η οποία φαίνεται στην Εικόνα 4.4.5.2 στο πάνω μέρος της σελίδας. Για την συγκεκριμένη λειτουργία επιλέγει τον φοιτητή είτε από την dropdown λίστα είτε μέσω αναζήτησης όνομα χρήστη ή με ονοματεπώνυμο και στην συνέχεια επιλέγει το παραδοτέο το οποίο βαθμολογήθηκε από την dropdown λίστα και τέλος πληκτρολογεί τον βαθμό του φοιτητή. Στην συνέχεια μπορεί να το αναρτήσει χωρίς επιπλέον σχόλια επιλέγοντας τον υπερσύνδεσμο “Submit grade” ή αν επιθυμεί περισσότερα σχόλια επιλέγει τον υπερσύνδεσμο “Upload more data” και εμφανίζεται ένα αναδυόμενο παράθυρο όπως φαίνεται την Εικόνα 4.4.5.1. Στο συγκεκριμένο παράθυρο έχει την δυνατότητα να συμπληρώσει ανατροφοδότηση ή να επιλέξει να ανεβάσει και κάποιο συμπληρωματικό αρχείο με σχόλια και τέλος θα επιλέξει τον υπερσύνδεσμο “UPLOAD DATA” για να γίνει η σχετική ενημέρωση τόσο στην Β.Δ. όσο και στον διακομιστή.

4.4.5.2. Ανάρτηση βαθμών για ομάδα ατόμων

Σε περίπτωση που ο καθηγητής επιθυμεί να αναρτήσει βαθμό για πολλά άτομα ταυτόχρονα δηλαδή μία ένα συγκεκριμένη Ο.Α.Ε χρησιμοποιεί την δεύτερη λειτουργία η οποία φαίνεται στην Εικόνα 4.4.5.2 στο κάτω μέρος της σελίδας. Για την συγκεκριμένη λειτουργία επιλέγει τον αναγνωριστικό αριθμό Ο.Α.Ε είτε από την dropdown λίστα είτε μέσω με πληκτρολόγηση και στην συνέχεια επιλέγει το παραδοτέο το οποίο βαθμολογήθηκε από την dropdown λίστα και τέλος πληκτρολογεί τον βαθμό της ομάδας. Στην συνέχεια μπορεί να το αναρτήσει χωρίς επιπλέον σχόλια επιλέγοντας τον υπερσύνδεσμο “Submit grade” ή αν επιθυμεί περισσότερα σχόλια επιλέγει τον υπερσύνδεσμο “Upload more data” και εμφανίζεται ένα αναδυόμενο παράθυρο όπως φαίνεται την Εικόνα 4.4.5.1. Στο συγκεκριμένο παράθυρο έχει την δυνατότητα να συμπληρώσει ανατροφοδότηση ή να επιλέξει να ανεβάσει και κάποιο συμπληρωματικό αρχείο με σχόλια και τέλος θα επιλέξει τον υπερσύνδεσμο “UPLOAD DATA” για να γίνει η σχετική ενημέρωση τόσο στην Β.Δ. όσο και στον διακομιστή.

4.4.5.3. Προβολή βαθμών τρέχοντος ακαδημαϊκού έτους

Η πλατφόρμα κατά την διάρκεια λειτουργίας της θα περιέχει ένα αρκετά μεγάλο αριθμό λογαριασμών τύπου φοιτητών με πολλές αναρτήσεις βαθμών για τον κάθε ένα και οι καθηγητές θα χρειαστεί πολλές φορές να περιορίσουν τα αποτελέσματα τους στο τρέχον ακαδημαϊκό έτος ή ακόμη και σε πιο εξειδικευμένες περιπτώσεις μία ομάδα ή ένα άτομο. Οι

πληροφορίες σχετικά με τους βαθμούς των φοιτητών εμφανίζονται στον πίνακα στο κάτω μέρος τις σελίδας όπως φαίνεται και στην Εικόνα 4.4.5.3. Στον πίνακα αυτό υπάρχουν λειτουργίες οι οποίες φαίνονται στην Εικόνα 4.4.5.5. οι οποίες διευκολύνουν αρκετά αυτή την κατάσταση με τα πολλά αποτελέσματα. Αρχικά, από τα αριστερά στα δεξιά, υπάρχει η δυνατότητα για αναζήτηση (search) η οποία ψάχνει ανάμεσα σε όλους τους φοιτητές τόσο με ονοματεπώνυμο όσο και με όνομα χρήστη 'username'. Δεύτερη λειτουργία είναι η αποθήκευση αρχείου η οποία αποθηκεύει όλα τα αποτελέσματα που εμφανίζονται στην οθόνη σε ένα αρχείο CSV ενώ με την τρίτη λειτουργία εκτυπώνονται τα συγκεκριμένα αποτελέσματα. Η προτελευταία λειτουργία χρησιμοποιείται για να επιλέξει ο καθηγητής τις στήλες τις οποίες επιθυμεί να εμφανίζονται ώστε να δώσει περισσότερη έμφαση σε κάποιες και λιγότερη σε άλλες που ίσως να μην είναι τόσο σημαντικές. Τέλος η τελευταία λειτουργία είναι σαν φίλτρο (filter) λόγω του ότι μπορεί να εμφανίσει περιορισμένα αποτελέσματα σχετικά με κάποιο περιορισμό για παράδειγμα το όνομα, το παραδοτέο, η χρονιά εγγραφής, ο αναγνωριστικό αριθμός ομάδας κτλ.



Εικόνα 4.4.5. 5: Λειτουργίες προβολής λογαριασμών φοιτητών

[SUBMIT GRADES](#)
[SEE GRADES](#)

Submit grade for student
Below you are able to add a grade for a specific student for a specific assignment. You can select the assignment and the student and the grade.

/100

Upload more data
[Submit Grade](#)

Submit grade for team
Below you are able to add a grade for a team assignment and this grade will be the same to all team members. You can select the assignment and the team and the grade.

/100

Upload more data
[Submit Grade](#)

Εικόνα 4.4.5. 2: Στιγμιότυπο σελίδας βαθμών

[SUBMIT GRADES](#)
[SEE GRADES](#)

Academic grades 2021									
	Username	Name	Surname	Deliverable	Team Number	Grade	Max Grade	Percentage	Year
☐	nandre02	Nicolaos	Andreou	Quiz	5	34	100	5	2021
☐	manast01	Maria	Anastasiou	Quiz	5	23	100	5	2021
☐	gpapav02	Giorgos	Papavamavas	Quiz		78	100	5	2021
☐	gpapav02	Giorgos	Papavamavas	Deliverable Phase A		89	100	10	2021
☐	nandre02	Nicolaos	Andreou	Deliverable Phase A	5	78	100	10	2021

Rows per page: 1-5 of 5

Εικόνα 4.4.5. 1: Εμφάνιση βαθμών για το τρέχον ακαδημαϊκό έτος

Feedback on your assignment submission on www.platform343.ucy.ac.cy



epl343cs.ucy.ac.cy@gmail.com <epl343cs.ucy.ac.cy@gmail.com>

5/4/2021 12:19 AM



To: apanay22@ucy.ac.cy

Hello student,

An instructor has posted some feedback on your assignment submission for "Deliverable Phase A"
You can see it appended to your [assignment submission](#) .

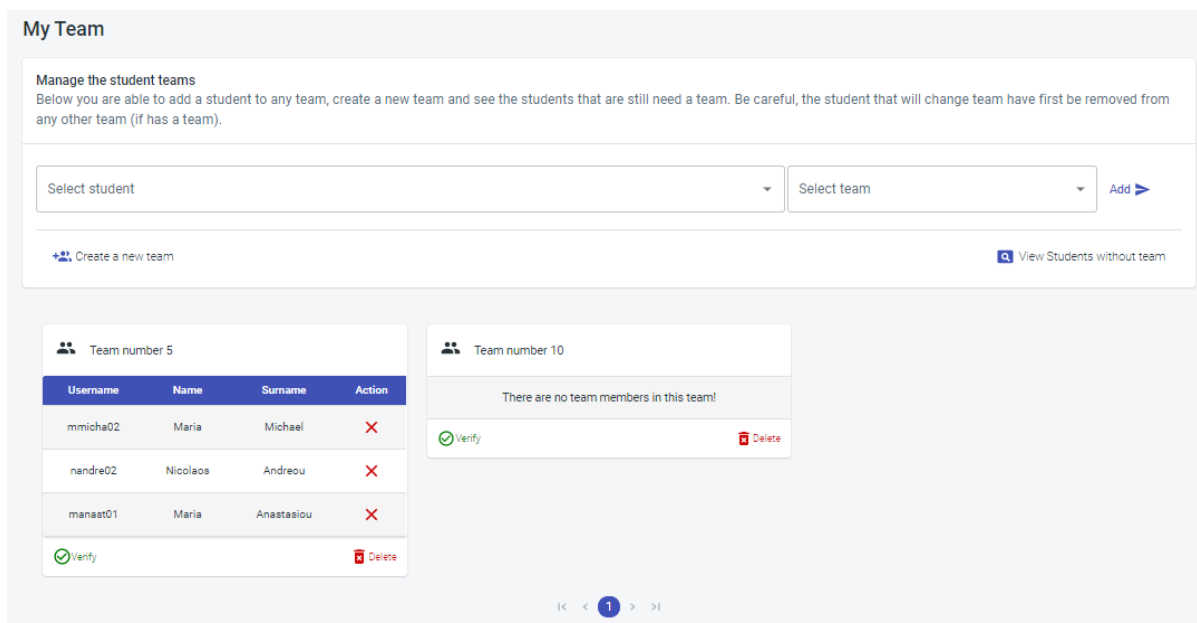
If you have any question you can send an email to an instructor and see the uploaded feedback (if it's available).

Εικόνα 4.4.5. 4: Ενημερωτικό μήνυμα ηλεκτρονικού ταχυδρομείου για ανάρτηση βαθμών

Εικόνα 4.4.5. 3: Επιπλέον πληροφορίες για βαθμό παραδοτέου

4.4.6. Ομάδες Φοιτητών

Οι φοιτητές αρχικά καλούνται να δημιουργήσουν ομάδες με τους συμφοιτητές τους έτσι ώστε να έχουν την δυνατότητα να δείξουν το ενδιαφέρον του σε Ε.Α. όπως περιγράφεται στην ενότητα 4.3.4.1. Οι ομάδες αυτές όμως για να είναι έγκυρες πρέπει να γίνουν επικύρωση (verify) από τουλάχιστο ένας χρήστη με λογαριασμό καθηγητή, λειτουργία η οποία γίνεται στην διεύθυνση 'domain'/app/team και φαίνεται στην Εικόνα 4.4.6.1. Κάποιες φορές ίσως ο καθηγητής να πρέπει να επεξεργαστεί τις ομάδες είτε προσθέτοντας άτομα είτε αφαιρώντας είτε δημιουργώντας τελείως καινούριες ομάδες.



Εικόνα 4.4.6. 1: Στιγμιότυπο σελίδας ομάδων

4.4.6.1. Δημιουργία ομάδων φοιτητών

Αρχικά ο καθηγητής έχει την δυνατότητα να δημιουργήσει καινούρια ομάδα από την αρχή επιλέγοντας τον υπερσύνδεσμο “Create a new team” η οποία φαίνεται στην εικόνα 4.4.6.1. και θα δημιουργηθεί μία νέα ομάδα όπως αυτή φαίνεται στην ομάδα με αναγνωριστικό αριθμό 10. Με την επιλογή του συγκεκριμένου υπερσυνδέσμου η σελίδα ανανεώνεται και γίνεται η ενημέρωση και στην Β.Δ.

4.4.6.2. Επεξεργασία ομάδων φοιτητών

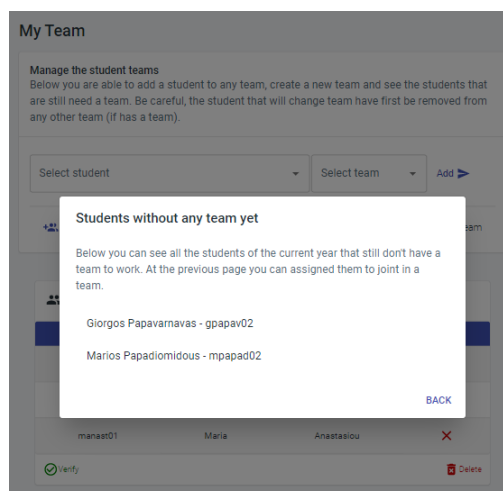
Αρχικά, ο καθηγητής ανά πάσα στιγμή μπορεί να διαγράψει κάθε φοιτητή από οποιαδήποτε ομάδα επιλέγοντας το εικονίδιο του ‘X’ δεξιά από το όνομα του φοιτητή και στην συνέχεια γίνεται η ενημέρωση στην Β.Δ., η σελίδα ανανεώνεται και ένα ενημερωτικό μήνυμα επιτυχημένης ή αποτυχημένης εκτέλεσης εμφανίζεται στο χρήστη. Επιπλέον με το διαδραστικό πίνακα στο πάνω μέρος της σελίδας όπως φαίνεται και στην Εικόνα 4.4.6.1. μπορεί πολύ εύκολα ο καθηγητής να επιλέξει ένα άτομο από την dropdown λίστα με φοιτητές, η οποία έχει την δυνατότητα αναζήτησης είτε με ονοματεπώνυμο ή με όνομα χρήστη για ευκολία και στο επόμενο dropdown το αναγνωριστικό της ομάδας στην οποία επιθυμεί να προσθέσει τον συγκεκριμένο χρήστη. Τέλος επιλέγοντας τον υπερσύνδεσμο “Add” γίνεται η ενημέρωση στην Β.Δ. και ανανεώνεται η σελίδα με τις αλλαγές. Σημαντική προϋπόθεση είναι αρχικά η ομάδα να έχει κενές θέσεις και επίσης ο επιλεγμένος φοιτητής να μην βρίσκεται σε κάποια άλλη ομάδα, σε αυτή την περίπτωση πρέπει ο καθηγητής να τον διαγράψει από την προηγούμενη του ομάδα και να τον προσθέσει στην καινούρια.

4.4.6.3. Εμφάνιση φοιτητών χωρίς ομάδα

Μία συμπληρωματική λειτουργία είναι αυτή που εμφανίζει μία λίστα με φοιτητές οι οποίοι δεν είναι μέλη σε κάποια ομάδα στα πλαίσια του μαθήματος και αναλαμβάνει ο καθηγητής να τους προσθέσει σε κάποια υπάρχουσα. Επιλέγοντας τον υπερσύνδεσμο “View students without team” από την σελίδα όπως φαίνεται και στον διαδραστικό πίνακα της Εικόνας 4.4.6.1. εμφανίζεται αναδυόμενο παράθυρο όπως φαίνεται και στην Εικόνα 4.4.6.2. με τους φοιτητές οι οποίοι δεν βρίσκονται σε κάποια ομάδα.

4.4.6.4. Διαγραφή / Επικύρωση ομάδας φοιτητών

Όταν πλέον οι φοιτητές έχουν δημιουργήσει τις ομάδες τους για να έχουν την δυνατότητα να δείχνουν ενδιαφέρον σε Ε.Α. αλλά και να εκτελούν διάφορες λειτουργίες πρέπει ένας χρήστης με λογαριασμό τύπου καθηγητή να την εγκρίνει, συγκεκριμένα ο χρήστης επιλέγει τον υπερσύνδεσμο “Verify” στην ομάδα που επιθυμεί. Αντιθέτως υπάρχει και η επιλογή “Delete” με την οποία η ομάδα διαγράφεται εντελώς και οι φοιτητές οι οποίοι ήταν μέλη της δεν έχουν πλέον ομάδα, οι αλλαγές αυτές ενημερώνουν την Β.Δ. και η σελίδα ανανεώνεται.



Εικόνα 4.4.6. 2: Λίστα φοιτητών χωρίς ομάδα

4.4.7. Δημιουργία λογαριασμών τύπου φοιτητή

Κάθε νέο ακαδημαϊκό έτος ένας μεγάλος αριθμός από λογαριασμούς τύπου φοιτητή δημιουργείται καθώς το μάθημα είναι υποχρεωτικό είναι μία λειτουργία ρουτίνας. Γι’ αυτό τον λόγο η λειτουργία η οποία βρίσκεται στην διεύθυνση ‘domain’/app/students. προσφέρει ευκολία δημιουργίας πολλαπλών λογαριασμών ταυτόχρονα με την μεταφόρτωση ενός αρχείου τύπου CSV με προκαθορισμένη μορφή που περιέχει τις απαραίτητες πληροφορίες για τον κάθε φοιτητή. Οι πληροφορίες αυτές μπορούν πολύ εύκολα να γίνουν εξαγωγή από την ιστοσελίδα

Banner με την οποία οι φοιτητές κάνουν εγγραφή στα μαθήματα για κάθε ακαδημαϊκό έτος και να αποθηκευτούν σε ένα αρχείο τύπου CSV.

Import

+ UPLOAD CSV FILE

DOWNLOAD SAMPLE

UPLOAD

0%

Students List - Year2021

☐	Username	Name	Surname	Email	Team Number	Working Project
☐	mmicha02	Maria	Michael	apanay22@ucy.ac.cy		
☐	ndandre02	Nicolaos	Andreou	apanay22@ucy.ac.cy		
☐	manast01	Maria	Anastasiou	apanay22@ucy.ac.cy		
☐	gpapav02	Giorgos	Papavarnavas	apanay22@ucy.ac.cy		
☐	mpapad02	Marios	Papadiomidous	apanay22@ucy.ac.cy		

Rows per page: 10 1-5 of 5

Εικόνα 4.4.7. 1: Στιγμιότυπο σελίδας δημιουργίας λογαριασμών φοιτητών

4.4.7.1. Μεταφόρτωση αρχείου CSV

Όταν ο χρήστης εκτελεί την λειτουργία αυτή για πρώτη φορά και δεν γνωρίζει την μορφή μπορεί να επιλέξει τον υπερσύνδεσμο “DOWNLOAD SAMPLE” και να αποθηκεύσει ένα δείγμα όπως αυτό στην Εικόνα 4.4.7.3. για να κατανοήσει την μορφή στην οποία πρέπει να δημιουργηθεί το CSV αρχείο. Ο καθηγητής τότε δημιουργεί ένα αρχείο CSV, έστω όπως το παράδειγμα στην Εικόνα 4.4.7.3, επιλέγει τον υπερσύνδεσμο “UPLOAD CSV FILE” και στην συνέχεια το αρχείο από την ηλεκτρονική συσκευή με την οποία είναι συνδεδεμένος στην πλατφόρμα. Τέλος επιλέγει τον υπερσύνδεσμο “UPLOAD” και όσο εκτελείτε η διαδικασία δημιουργίας λογαριασμών η οριζόντια ράβδος δείχνει την ανάδραση της εκτέλεσης και αλλάζει ανάλογα μέχρι να φτάσει στο μέγιστο 100% το οποίο ισούται τόσο με το επιτυχημένο τέλος της εκτέλεσης της διαδικασίας όσο και με τις αλλαγές στην Β.Δ.

	A	B	C	D	E
1	Name	Surname	Email	Username	
2					
3					
4					
5					
6					
7					
8					
9					
10					

Εικόνα 4.4.7. 3: Δείγμα αρχείου

	A	B	C	D
1	Name	Surname	Email	Username
2	Antria	Panayiotou	apanay22@ucy.ac.cy	apanay22
3	Maria	Anastasiou	manast01@ucy.ac.cy	manast01
4	Nicolaos	Andreou	nanadr02@ucy.ac.cy	nanadr02
5	Chryso	Makri	cmakri02@ucy.ac.cy	cmakri02
6	Giorgos	Papavarnavas	gpapav02@ucy.ac.cy	gpapav02
7	Marios	Papadiomidous	mpapad02@ucy.ac.cy	mpapad02
8	Maria	Michael	mmicha02@ucy.ac.cy	mmicha02
9				
10				

Εικόνα 4.4.7. 3: Παράδειγμα μεταφορτωμένου αρχείου

4.4.7.2. Προβολή φοιτητών

Η πλατφόρμα κατά την διάρκεια λειτουργίας της θα περιέχει ένα αρκετά μεγάλο αριθμό λογαριασμών τύπου φοιτητών και οι καθηγητές θα χρειαστεί πολλές φορές να περιορίσουν τα αποτελέσματα τους στο τρέχον ακαδημαϊκό έτος ή ακόμη και σε πιο εξειδικευμένες περιπτώσεις μία ομάδα ή ένα άτομο. Οι πληροφορίες σχετικά με τους λογαριασμούς

εμφανίζονται στον πίνακα στο κάτω μέρος τις σελίδας όπως φαίνεται και στην Εικόνα 4.4.7.1. Στον πίνακα αυτό υπάρχουν λειτουργίες οι οποίες φαίνονται στην Εικόνα 4.4.7.4. οι οποίες διευκολύνουν αρκετά αυτή την κατάσταση με τα πολλά αποτελέσματα. Αρχικά, από τα αριστερά στα δεξιά, υπάρχει η δυνατότητα για αναζήτηση (search) η οποία ψάχνει ανάμεσα σε όλους τους φοιτητές τόσο με ονοματεπώνυμο όσο και με όνομα χρήστη 'username'. Δεύτερη λειτουργία είναι η αποθήκευση αρχείου η οποία αποθηκεύει όλα τα αποτελέσματα που εμφανίζονται στην οθόνη σε ένα αρχείο CSV ενώ με την τρίτη λειτουργία εκτυπώνονται τα συγκεκριμένα αποτελέσματα. Η προτελευταία λειτουργία χρησιμοποιείται για να επιλέξει ο χρήστης τις στήλες τις οποίες επιθυμεί να εμφανίζονται ώστε να δώσει περισσότερη έμφαση σε κάποιες και λιγότερη σε άλλες που ίσως να μην είναι τόσο σημαντικές. Τέλος η τελευταία λειτουργία είναι σαν φίλτρο (filter) λόγο του ότι μπορεί να εμφανίσει περιορισμένα αποτελέσματα σχετικά με κάποιο περιορισμό για παράδειγμα το όνομα, η χρονιά εγγραφής, ο αναγνωριστικός αριθμός ομάδας, το Ε.Α. το οποίο είναι υπεύθυνη η ομάδα κτλ.



Εικόνα 4.4.7. 4: Λειτουργίες προβολής λογαριασμών φοιτητών

4.5.Ενδεικτικά σημεία υλοποίησης

Πιο κάτω φαίνεται η ενδεικτική υλοποίηση για την σελίδα προφίλ του χρήστη η οποία βρίσκεται στην διεύθυνση 'domain'/app/account. Η ιδιαιτερότητα η οποία έχει η συγκεκριμένη σελίδα είναι ότι αλλάζει ανάλογα με τις λειτουργίες και δυνατότητες κάθε τύπο λογαριασμού και είναι προσβάσιμη και από κάθε χρήστη. Οι λειτουργίες περιγράφονται στις ενότητες 4.3.7, 4.1.8 και 4.2.3 αντίστοιχα και πιο κάτω υπάρχει η περιγραφή για την υλοποίηση μέσω κώδικα JavaScript.

4.5.1. Εισαγωγή συστατικών (components), εικονιδίων και λειτουργιών

Αρχικά γίνεται η εισαγωγή συγκεκριμένων μεταβλητών π.χ. γραμμή 9 – LAN_EN , βιβλιοθηκών που περιέχουν σημαντικές λειτουργίες π.χ. γραμμή 1 – React και γραμμή 6 – connect από την Redux για να εισαγάγουν τα στοιχεία που είναι ήδη αποθηκευμένα στο state ή να τα ενημερώσουν και τέλος οποιαδήποτε άλλα στοιχεία π.χ. εικονίδια (γραμμή 10,11,12) / αρχεία κειμένου για μετάφραση (γραμμή 7,8).

```
1 import React from 'react';
2 import PropTypes from 'prop-types';
3 import * as Yup from 'yup';
4 import { Formik } from 'formik';
5 import { Box, Button, Card, CardContent, CardHeader, Divider, Grid, TextField } from '@material-ui/core';
6 import { connect } from 'react-redux';
7 import jsonGR from "src/assets/data/greek.json";
8 import jsonEN from "src/assets/data/english.json";
9 import { LAN_EN } from 'src/actions/types';
10 import InputAdornment from '@material-ui/core/InputAdornment';
11 import PersonOutlineOutlinedIcon from '@material-ui/icons/PersonOutlineOutlined';
12 import EmailOutlinedIcon from '@material-ui/icons/EmailOutlined';
```

Εικόνα 4.5.1: Εισαγωγή συστατικών (components), εικονιδίων και λειτουργιών

4.5.2. Εισαγωγή στοιχείων και τιμών από το state Redux

Η συγκεκριμένη συνάρτηση χρησιμοποιείται συνήθως για να καλέσει την κατάσταση τιμών της πλατφόρμας η οποία είναι ήδη αποθηκευμένη με την βοήθεια της βιβλιοθήκης Redux. Στην Εικόνα 4.5.2. φαίνεται ότι παίρνει στοιχεία που αφορούν τον χρήστη που είναι συνδεδεμένος στην σελίδα με όνομα μεταβλητής “loginsession” όσο και στοιχεία για την γλώσσα προτιμήσεις “selectedlan” και αυτά τα αποθηκεύει στο συγκεκριμένο state της σελίδας για να το χρησιμοποιήσει.

```
99   const mapStateToProps = state => {
100     return { loginsession: state.loginsession,
101             selectedlan: state.selectedlan };
102   };
```

Εικόνα 4.5. 2: Τιμές από την κατάσταση σελίδας

4.5.2.1. Περιγραφή υλοποίησης ‘selectedlan’

Κάθε χρήστης την στιγμή που επιλέγει την γλώσσα με την οποία θα εμφανίζεται η πλατφόρμα μία μεταβλητή ενημερώνει την κατάσταση του συστήματος στις δύο πιθανές τιμές ‘LAN_GR’ – ελληνικά και ‘LAN_EN’ – αγγλικά. Ανάλογα με την τιμή εισόδου στον τύπο της μεταβλητής εκτελείτε το κατάλληλο μονοπάτι διακλάδωσης και αλλάζει η μεταβλητή σε όλο το πρόγραμμα.

```
1   import { LAN_GR, LAN_EN } from '../actions/types';
2
3   const INITIAL_STATE = {
4     selectedlan: LAN_EN
5   };
6
7   export default (state = INITIAL_STATE, action) => {
8     switch (action.type) {
9       case LAN_GR:
10        return { selectedlan: LAN_GR };
11       case LAN_EN:
12        return { selectedlan: LAN_EN };
13       default:
14        return state;
15     }
16   };
```

Εικόνα 4.5. 33 Reducer επιλεγμένης γλώσσας

4.5.2.2. Περιγραφή υλοποίησης ‘loginsession’

Κάθε χρήστης την στιγμή την οποία συνδέεται στην ιστοσελίδα μία σειρά από μεταβλητές ενημερώνονται, συγκεκριμένα ο τύπος σύνδεσης (LOGOUT, STUDENT, COMPANY, TEACHER), το όνομα χρήστη, το αναγνωριστό χρήστη, το αναγνωριστικό τύπου λογαριασμού (2 – καθηγητής, 3 – φοιτητής, 4- οργανισμός), αριθμός ομάδας φοιτητή, γνώσεις σε διαδικτυακές / κινητές / desktop εφαρμογές και ονοματεπώνυμο. Αυτά τα στοιχεία στην αρχή είναι ίσα με κενό (‘null’) όπως φαίνεται και στην Εικόνα 4.5.4. , στην συνέχεια ανάλογα με τον τύπο σύνδεσης οι μεταβλητές συμπληρώνονται αναλόγως και όσες δεν υπάρχουν για το συγκεκριμένο τύπο λογαριασμού π.χ. ομάδα για τύπο λογαριασμού οργανισμού μένουν κενά, όπως φαίνεται και στην Εικόνα 4.5.5. Σε περίπτωση εκτέλεσης λειτουργίας αποσύνδεσης όλες οι μεταβλητές επιστρέφουν στην αρχική τους τιμή το κενό ‘null’. Ανάλογα με την τιμή

εισόδου στον τύπο της μεταβλητής εκτελείτε το κατάλληλο μονοπάτι διακλάδωσης και αλλάζει η μεταβλητή σε όλο το πρόγραμμα

```
3  const INITIAL_STATE = {
4    loginSession: null,
5    Username: null,
6    ID: null,
7    typeAccount: null,
8    team: null,
9    web: null,
10   desktop: null,
11   mobile: null,
12   name: null,
13   surname: null
14  };
```

Εικόνα 4.5. 4: Αρχικοποίηση κατάστασης (State)

```
16  export default (state = INITIAL_STATE, action) => {
17    switch (action.type) {
18      case USER_TEACHER:
19        return {
20          loginSession: USER_TEACHER,
21          Username: action.username,
22          ID: action.id,
23          typeAccount: 2,
24          team: null,
25          name: action.name,
26          surname: action.surname
27        };
28      case USER_STUDENT:
29        return {
30          loginSession: USER_STUDENT,
31          Username: action.username,
32          ID: action.id,
33          typeAccount: 3,
34          team: action.team,
35          web: action.web,
36          desktop: action.desktop,
37          mobile: action.mobile,
38          name: action.name,
39          surname: action.surname
40        };
41    }
42  };
```

Εικόνα 4.5.5: Στοιχεία κατάστασης (State)

```
41  case USER_COMPANY:
42    return {
43      loginSession: USER_COMPANY,
44      Username: action.username,
45      ID: action.id,
46      typeAccount: 4,
47      team: null,
48      name: action.name,
49      surname: action.surname
50    };
51  case 'LOGOUT':
52    return {
53      loginSession: NO_USER,
54      Username: null,
55      ID: null,
56      typeAccount: null,
57      team: null,
58      name: null,
59      surname: null
60    };
61  default:
62    return state;
63  }
64  };
```

4.5.3. Μετάφραση σελίδας

Χρησιμοποιώντας την πληροφορία της μεταβλητής η οποία είναι αποθηκευμένη στο πεδίο 'selectedLan' η οποία περιγράφεται στην ενότητα 4.5.2.1. βλέπουμε την γλώσσα προτίμησης του χρήστη και κατά συνέπεια ενώ αρχικά στην Εικόνα 4.5.1. στις γραμμές 7,8 εισάγουμε δύο κείμενα JSON στο τέλος χρησιμοποιούμε μόνο το ένα από τα δύο, αυτό της προτίμησης του χρήστη. Σε κάθε σημείο το οποίο περιέχει κείμενο θα εισάγεται μία μεταβλητή για παράδειγμα {data.value1} η οποία αντλεί την πληροφορία από το κατάλληλο JSON αρχείο έτσι ώστε να αλλάζει αυτόματα η γλώσσα.

```
296  var language=this.props.selectedlan.selectedlan;
297  var data = language === LAN_EN?jsonEN.profilePage:jsonGR.profilePage;
```

Εικόνα 4.5. 6: Εισαγωγή μεταφρασμένου κειμένου

4.5.4. Εμφάνισης ενημερωτικών μηνυμάτων

Για να εμφανίζουμε ενημερωτικά μηνύματα στον χρήστη χρησιμοποιούμε την τεχνική Query String η οποία επιπλέον από την διεύθυνση της σελίδα προσθέτει στοιχεία με τιμές οι οποίες έχουν την ακόλουθη μορφή http://URL_DOMAIN.com.cy?title=Query_string&action=edit [15]. Υπάρχει δηλαδή μία επιπλέον πληροφορία μέσω URL όπως φαίνεται και στην συγκεκριμένη περίπτωση στην Εικόνα 4.5.7. αναγνωρίζεται μέσω της λέξης 'update' η οποία

αντίστοιχη στην ενημέρωση των στοιχείων του χρήστη στην Β.Δ.. Σε περίπτωση που η τιμή είναι ίση με 1 τότε έγινε επιτυχημένη ενημέρωση στην Β.Δ. σε αντίθετη περίπτωση οποιαδήποτε άλλη τιμή και συγκεκριμένα ίση με 0 αντιστοιχεί σε αποτυχημένη ενημέρωση. Ανάλογα με το αποτέλεσμα της τιμής αυτής εμφανίζεται επιτυχημένο ή αποτυχημένο μήνυμα ανάλογα για διάρκεια ίση με 10000 χρονικές στιγμές και μήνυμα το οποίο βρίσκεται στο αρχείο JSON.

```
40 const update = (new URLSearchParams(search)).get('update');
41 if (update === '1') {
42   toast.success(data.success, {
43     duration: 10000,
44   });
45 } else if (update === '0') {
46   toast.error(data.wrong, {
47     duration: 10000,
48   });
49 }
```

Εικόνα 4.5. 7: Εμφάνιση ενημερωτικών μηνυμάτων

4.5.5. Δομή σελίδας προφίλ

Στην Εικόνα 4.5.8 φαίνεται η κεντρική δομή της ιστοσελίδας προφίλ η οποία υπάρχει σε όλα τα είδη λογαριασμών πίνακα ανανέωσης στοιχείων με την διαφορά ότι η χρήστες με λογαριασμό οργανισμού έχουν επιπλέον δυνατότητες αλλαγής στοιχείων. Επίσης κάτω από τον πίνακα αυτό για τους λογαριασμούς φοιτητή και οργανισμού υπάρχει και ένας δεύτερος πίνακας ο οποίος περιέχει προσθήκη γνώσεων και προσθήκη εικόνας προφίλ αντίστοιχα. Ενώ η δομή της σελίδας είναι σταθερή εναλλάσσεται ανάλογα με τον τύπο λογαριασμού του συνδεδεμένου χρήστη και αυτό συμβαίνει λόγο του ότι στις γραμμές 81 και 91 υπάρχουν μεταβλητές οι οποίες αλλάζουν και θα περιγράφονται στην ενότητα 4.5.6 και 4.5.7.

```
56 return (
57   <Page
58     title="Account"
59   >
60     <Toaster position="top-center" reverseOrder={true}/>
61     <Container maxWidth="lg">
62       <Grid container spacing={3} >
63         <Grid...
64         <Grid
65           item...
66           xs={12}
67         >
68           {ProfileDetails(props.loginSession.typeAccount)}
69         </Grid>
70         <Grid
71           item...
72           xs={12}
73         >
74           {logo}
75         </Grid>
76       </Grid>
77     </Container>
78   </Page>
79 );
```

Εικόνα 4.5. 8: Δομή ιστοσελίδας προφίλ

4.5.6. Εμφάνιση κατάλληλου πίνακα αλλαγής στοιχείων χρήστη

Η μεταβλητή η οποία αντιστοιχεί στην εμφάνιση του κατάλληλου πίνακα επεξεργασίας στοιχείων λογαριασμού χρήστη ονομάζεται 'ProfileDetails' και παίρνει ως είσοδο τον τύπο του λογαριασμού από την μεταβλητή που αποθηκεύεται μέσω Redux. Μέσω εντολής διακλάδωσης για λογαριασμούς φοιτητή και καθηγητή οι πληροφορίες αλλαγής προφίλ είναι ίδιες γι' αυτό και επισυνάπτεται η ίδια ετικέτα 'ProfileDetailsST' ενώ στην διακλάδωση με τιμή 4 είναι διαφορετική ετικέτα 'ProfileDetailsCompany' οι οποίες περιγράφονται πιο κάτω.

```
16 const ProfileDetails = (typeAccount) => {
17   if(typeAccount === 2 ){
18     return <ProfileDetailsST></ProfileDetailsST>;
19   }
20   else if(typeAccount === 3 ){
21     return <ProfileDetailsST></ProfileDetailsST>;
22   }
23   else if(typeAccount === 4 ){
24     return <ProfileDetailsCompany></ProfileDetailsCompany>;
25   }
26 }
```

Εικόνα 4.5.9: Διαφοροποίηση πίνακα επεξεργασίας στοιχείων

4.5.7. Εμφάνιση δεύτερης λειτουργίας για οργανισμό και φοιτητή

Η μεταβλητή η οποία αντιστοιχεί στην εμφάνιση της δεύτερης λειτουργίας για τύπο λογαριασμού χρήστη οργανισμού ή φοιτητή ονομάζεται 'logo' και παίρνει ως είσοδο τον τύπο του λογαριασμού από την μεταβλητή που αποθηκεύεται μέσω Redux. Μέσω εντολής διακλάδωσης για λογαριασμό οργανισμού επισυνάπτεται η ετικέτα 'ProfileLogo', για λογαριασμό φοιτητή η ετικέτα 'Knowledge' ενώ για λογαριασμό καθηγητή ο χώρος αυτός είναι κενός λόγος του ότι δεν υπάρχει κάποια επιπλέον δυνατότητα όσο αφορά την επεξεργασία στοιχείων προφίλ για αυτόν.

```
51 var logo = ((props.loginSession.typeAccount === 4)
52   ? <ProfileLogo />
53   :(props.loginSession.typeAccount === 3)?<Knowledge />:null);
```

Εικόνα 4.5. 40: Επιπλέον λειτουργία επεξεργασίας προφίλ

4.5.8. Αρχικοποίηση μεταβλητών σελίδας

Στην Εικόνα 4.5.11 φαίνονται οι αρχικοποιήσεις οι οποίες έγιναν στις σελίδες ProfileDetailsCompany, 'ProfileLogo' και 'Knowledge', τα περισσότερα πεδία με κενό string

```
19 constructor(props) {
20   super(props);
21   this.state = {
22     firstName: '',
23     lastName: '',
24     email: '',
25     phone: '',
26     address: '',
27     companyName: '',
28     type:0
29   };
30 }

48 constructor(props) {
49   super(props);
50   this.state = {
51     firstName: '',
52     lastName: '',
53     email: '',
54     phone: '',
55     address: '',
56     companyName: '',
57     message: '',
58     selectedFile: null,
59     data: this.props.loginSession,
60     loaded: null,
61     DocumentInfo: null
62   };
63 }

116 constructor(props) {
117   super(props);
118   this.state = {
119     skills: [""].map(item => ({
120       id: uuidV4(),
121       item,
122     })))
123   };
124 }
```

Εικόνα 4.5. 51: Αρχικοποιήσεις μεταβλητών κατάστασης (State)

λόγω του ότι είναι φόρμα με πεδία τα οποία θα συμπληρωθούν στην συνέχεια από την εισαγωγή στοιχείων από την Β.Δ. διαδικασία η οποία περιγράφεται στην ενότητα 4.1.3.

4.5.9. Εισαγωγή στοιχείων από Β.Δ.

Μόλις ο χρήστης επισκεφτεί την σελίδα προφίλ του πρέπει να γίνει εισαγωγή στοιχείων από την Β.Δ. γι' αυτό και χρησιμοποιείται η λειτουργία `ComponentDidMount` η οποία ανήκει στην βιβλιοθήκη της `React` που κάνουμε εισαγωγή στην πρώτη γραμμή του κώδικα μας. Η λειτουργία αυτή εκτελείται με κάθε ανανέωση της σελίδας και αυτή με την σειρά της καλεί την λειτουργία `getUserInfo` και σε κάθε περίπτωση λαμβάνει διαφορετικά στοιχεία ανάλογα με την σελίδα χρήσης της. Για παράδειγμα στην Εικόνα 4.5.12 φαίνεται η λήψη που γίνεται για να παρθούν τα υπάρχοντα στοιχεία χρήστη με λογαριασμό εταιρείας ενώ στην εικόνα 4.5.13 γίνεται η λήψη των γνώσεων (skills) για κάποιο χρήστη με λογαριασμό φοιτητή και στις δύο περιπτώσεις είναι ο τρόπος με τον οποίο αρχικοποιείται ο πίνακας που εμφανίζεται. Παρόμοια λογική ακολουθείται και στην εικόνα 4.5.14 όπου λαμβάνει την πληροφορία κατά πόσο ο οργανισμός έχει ανεβάσει εικόνα προφίλ για να την εμφανίσει.

```
32 componentDidMount() {
33   | this.getUserInfo();
34 }
35
36 getUserInfo() {
37   requests.getProfileData(this.props.loginSession.Username, (response)
38     this.setState({firstName: response.queryResults.Name,
39       lastName: response.queryResults.Surname,
40       address: response.queryResults.Address,
41       companyName: response.queryResults.CompanyName,
42       phone: response.queryResults.Telephone,
43       email: response.queryResults.Email,
44       type: response.queryResults.TypeAccount
45     });
46   });
47 }
```

Εικόνα 4.5. 12 Στοιχεία λογαριασμού οργανισμού

```
277 componentDidMount() {
278   | this.getUserInfo();
279 }
280
281 getUserInfo() {
282   requests.getProfileData(this.props.loginSession.Username, (response)
283     var skillsJson= JSON.parse(response.queryResults.Knowledge);
284     this.setState({skills: skillsJson.all.map(item => ({
285       id: uuidV4(),
286       item,
287     })),
288   });
289   });
290   });
291 }
```

Εικόνα 4.5. 13 Εισαγωγή γνώσεων χρήστη

```
64 componentDidMount() {
65   | this.getUserInfo();
66 }
67
68 getUserInfo() {
69   requests.getProfileData(this.props.loginSession.Username, (response) => {
70     | this.setState({DocumentInfo: response.queryResults.Logo });
71   });
72 }
```

Εικόνα 4.5. 64 Εισαγωγή εικόνας προφίλ .

4.5.10. Ράβδος ανάδρασης

Υπάρχει η συγκεκριμένη ράβδος ανάδρασης η οποία φαίνεται στην Εικόνα 4.5.15 και χρησιμοποιείται σε διάφορα σημεία στην ιστοσελίδα, κυρίως στις λειτουργίες που χρειάζεται να ανεβάσει κάποιο αρχείο για παράδειγμα η συγκεκριμένη είναι για την μεταφόρτωση της εικόνας προφίλ στο λογαριασμό τύπο οργανισμού. Είναι μία οριζόντια ράβδος η οποία στα δεξιά δείχνει το ποσοστό προόδου το οποίο είναι αποθηκευμένο στο state της σελίδας και αλλάζει ανάλογα με την μεταφόρτωση που περιγράφεται στην ενότητα 4.2.3.2.

```

31 function LinearProgressWithLabel(props) {
32   return (
33     <Box style={{margin: "10px 20px"}} display="flex" alignItems="center">
34       <Box width="100%" mr={1}>
35         <LinearProgress variant="determinate" value={props.value} />
36       </Box>
37       <Box minWidth={35}>
38         <Typography variant="body2" color="textSecondary">{`${Math.round(
39           props.value,
40         )}%`}</Typography>
41       </Box>
42     </Box>
43   );
44 }

```

Εικόνα 4.5. 15: Ράβδος ανάδρασης με ποσοστό μεταφόρτωσης

4.5.11. Έλεγχοι φόρμας επεξεργασίας στοιχείων προφίλ

Για την δημιουργία της φόρμας η οποία εμφανίζεται για την επεξεργασία στοιχείων προφίλ χρησιμοποιείται η ετικέτα `Formik` η οποία έχει λειτουργίες αρχικοποίησης στοιχείων και αυτόματους ελέγχους για τα στοιχεία. Η αρχικοποίηση γίνεται στην γραμμή 58 της Εικόνας 4.5.16 με τις προκαθορισμένες τιμές μέχρι να γίνει η εισαγωγή από την Β.Δ., λειτουργία η οποία περιγράφεται στην ενότητα 4.1.3. Όσο αφορά τους ελέγχους πεδίων γίνεται έλεγχος για τον τύπο δεδομένων για παράδειγμα για τον αριθμό τηλεφώνου, στην γραμμή 64, υπάρχει περιορισμός να χρησιμοποιηθούν μόνο αριθμοί (όχι γράμματα) και ο αριθμός ψηφίων να είναι ίσως με 8, σε αντίθετη περίπτωση θα εμφανίζονται μηνύματα λάθους για να ενημερώνουν τον χρήστη και να διακόπτεται η διαδικασία ανανέωσης με σκοπό να περιοριστούν λάθος ενημερώσεις.

```

57 <Formik
58   initialValues={{firstName: '',lastName: '',email: '',phone: '',address: '', companyname: ''}}
59   validationSchema={ Yup.object().shape({
60     firstName: Yup.string().max(255),
61     email: Yup.string().email(data.messages.value5a).max(255),
62     address: Yup.string().max(255),
63     lastName: Yup.string().max(255),
64     phone: Yup.string()
65       .matches(/^\d{1,4}[\d-]*([\d-]{2,3})?[\d-]*([\d-]{2,4})?[\d-]*?[\d-]{3,4}?$/),
66     data.messages.value6).max(8, data.messages.value6).min(8, data.messages.value6),
67     companyName: Yup.string().max(255),
68   })
69 }
70   onSubmit={ (values, {setSubmitting, setErrors, setStatus, resetForm}) => {
71     requests.updateProfile(this.props.loginSession.Username,values.firstName, values.lastName, values.email,
72       values.phone, values.address, values.companyname , function (response) {
73       setSubmitting(false);
74       window.location.replace('/app/account?update=1');
75     });
76   }}
77 >

```

Εικόνα 4.5. 76: Προκαθορισμένες τιμές φόρμας και έλεγχοι

4.5.12. Ανανέωση γνώσεων στην Β.Δ.

Με τη λειτουργία `'handleUpload'` ανανεώνονται οι αλλαγές οι οποίες έκανε ο φοιτητής στις γλώσσες προγραμματισμού και frameworks και ενημερώνεται η Β.Δ. Υπάρχει μια μεταβλητή πίνακα ο οποίος περιέχει όλες τις γλώσσες προγραμματισμού και frameworks και ακόμα 3 μεταβλητές πινάκων οι οποίες περιέχουν τα δεδομένα αυτά διαμοιρασμένα το καθένα στην ειδικότητα του. Για παράδειγμα όπως φαίνεται στις γραμμές 142, 159 και 202 της Εικόνας 4.5.17 υπάρχουν οι 3 μεταβλητές οι οποίες αναφέρονται στις γλώσσες προγραμματισμού για desktop εφαρμογές, κινητές εφαρμογές και διαδικτυακές εφαρμογές. Συγκεκριμένα όπως φαίνεται και στην ροή ελέγχου διακλάδωσης των γραμμών 250 με 264 ανάλογα με τις επιλογές

του φοιτητή ελέγχει αν η επιλογή του από την dropdown λίστα (αποθηκεύεται στο state της σελίδας και αρχικοποιείται από την εισαγωγή η οποία γίνεται από την Β.Δ και περιγράφεται στην λειτουργία 4.1.3) βρίσκεται στο συγκεκριμένο πίνακα επιλογών και ενημερώνεται η ανάλογη μεταβλητή. Στην συνέχεια εκτελείτε ένα ερώτημα (query) μέσω της Express JS (Node js) και όταν το αποτέλεσμα είναι επιτυχημένο, δηλαδή επιστραφεί status 200, ανανεώνεται η σελίδα με την χρήση της τεχνικής Query String η οποία περιγράφεται στην ενότητα 4.5.4.

```
140 handleUpload = (event) => {
141   event.preventDefault();
142   > const desktop = [ ...
158   ]
159   > const mobile = [ ...
201   ];
202   > const web = [ ...
244   ];
245   let webValue=0;
246   let desktopValue=0;
247   let appValue=0;
248
249   this.state.skills.map((item) => {
250     desktop.map((item2)=> {
251       if(item2 === item.item){
252         desktopValue=1;
253       }
254     })
255     web.map((item2)=> {
256       if(item2 === item.item){
257         webValue=1;
258       }
259     })
260     mobile.map((item2)=> {
261       if(item2 === item.item){
262         appValue=1;
263       }
264     })
265   });
266   requests.updateKnowledge(this.state.skills, webValue, appValue, desktopValue,this.props.loginSession.ID,
267     function (response) {
268       if(response.queryResults === 200){
269         window.location.replace('/app/account?update=1');
270       }
271       else{
272         window.location.replace('/app/account?update=0');
273       }
274     });
275   };
```

Εικόνα 4.5. 87: Ανανέωση γνώσεων στην Β.Δ.

4.5.13. Περιγραφή προσθήκης και αφαίρεσης γνώσεων

Η λειτουργία handleAddKnowledge εκτελείται την στιγμή που ο χρήστη επιλέξει τον υπερσύνδεσμο “AddMore” και τότε μία επιπλέον επιλογή προστίθεται στην λίστα με ένα τυχαίο ξεχωριστό αναγνωριστικό id για να ξεχωρίζουν και την τιμή επιλογής από την dropdown λίστα.

Η λειτουργία handleDeleteKnowledge εκτελείται την στιγμή που ο χρήστη επιλέξει τον υπερσύνδεσμο “X” στα δεξιά κάθε επιλογής για να διαγραφτεί. Ανάλογα με την επιλεγμένη τιμή της dropdown λίστα αναζητάει την αποθηκευμένη κατάσταση σελίδας για να την

αφαιρέσει από την λίστα με την χρήση της μεθόδου filter η οποία απορρίπτει (άρα αφαιρεί) τις τιμές που είναι ίσες με την επιλεγμένη .

```
126   handleAddKnowledge = () => {
127     const skills2 = [...this.state.skills, {id: uuidV4() ,item:''}];
128     this.setState({
129       skills: skills2
130     });
131   }
132 }
133
134   handleDeleteKnowledge = (value) => {
135     this.setState(prevState => ({
136       skills: prevState.skills.filter((x) => x.id !== value )
137     }));
138   }
139 }
```

Εικόνα 4.5. 98: Ανανέωση γνώσεων (Προσθήκη και αφαίρεση)

4.5.14. Μεταφόρτωση εικόνας προφίλ

Με την λειτουργία 'handleUpload' μεταφορτώνεται η εικόνα προφίλ του χρήστη αφού επιλέχτηκε αρχικά το αρχείο από την συσκευή που ο χρήστης είναι συνδεδεμένος στην πλατφόρμα αλλιώς εμφανίζει ενημερωτικό μήνυμα (γραμμή 77). Δημιουργείται μεταβλητή τύπου FormData (γραμμή 80), έχει την δυνατότητα να περιέχει κωδικοποιημένα αντικείμενα (object) τύπου multipart/form-data [16] , στην οποία αποθηκεύουμε την εικόνα – αντικείμενο που θα στείλουμε στο Node JS για να μεταφορτωθεί στον διακομιστή και να ενημερωθεί η Β.Δ.. Γίνεται ασύγχρονο κάλεσμα με την βιβλιοθήκη axios η οποία περιγράφεται στην ενότητα 2.4.2.9 και εκτός από την μεταβλητή FormData αποστέλλεται επίσης το αναγνωριστικό του χρήστη (ID), το όνομα του αρχείου και ο φάκελος στον κατάλογο που αποθηκεύονται τα αρχεία στον διακομιστή (path directory). Την στιγμή της μεταφόρτωσης ανανεώνεται και η ράβδος ανανέωσης η οποία περιγράφεται στην ενότητα 4.1.4 (γραμμές 87-92) ενημερώνοντας την μεταβλητή loaded στην κατάσταση της σελίδας. Με το τέλος της εκτέλεσης εμφανίζεται ενημερωτικό μήνυμα επιτυχημένες εκτέλεσης (γραμμή 96) και ανανεώνεται η σελίδα με την χρήση της τεχνικής Query String η οποία περιγράφεται στην ενότητα 4.5.4.

```
74   handleUpload = (event) => {
75     event.preventDefault();
76     if (!this.state.selectedFile) {
77       this.setState({message: 'Please select a file first!' });
78       return;
79     }
80     const dataFile = new FormData();
81
82     var path = '/platforma343/public/uploads/CompanyLogos/';
83     dataFile.append('file', this.state.selectedFile, this.state.selectedFile.name);
84     axios
85       .post('http://localhost:9000/uploadLogo?id='+this.state.data.ID+'&name='+this.state.selectedFile.name
86         + '&path=' + path, dataFile, {
87         onUploadProgress: (ProgressEvent) => {
88           this.setState({
89             loaded: Math.round(
90               (ProgressEvent.loaded / ProgressEvent.total) * 100
91             )
92           });
93         }
94       })
95       .then((res) => {
96         this.setState({selectedFile: '',message: 'Uploaded successfully!' });
97       }).then(setTimeout(function(){ window.location.replace('/app/account?changes=1'); }, 1000)
98     )
99   };
```

Εικόνα 4.5. 1910: Μεταφόρτωση εικόνας προφίλ

4.5.15. Διαγραφή ανεβασμένου αρχείου – εικόνας προφίλ

Με την λειτουργία 'RemoveFile' διαγράφεται ένα ήδη μεταφορτωμένο αρχείο το οποίο είναι συνδεδεμένο με τον λογαριασμό του χρήστη και γι' αυτό στην ασύγχρονη σύνδεση μέσω της βιβλιοθήκη axios, η οποία περιγράφεται στην ενότητα 2.4.2.9, εκτός από τον φάκελος στον κατάλογο ο οποίος περιέχει το αποθηκευμένο αρχείο στον διακομιστή (path directory) αποστέλλεται και ο αναγνωριστικός αριθμός χρήστη (ID) . Οι αλλαγές οι οποίες εκτελούνται συμβαίνουν τόσο στην Β.Δ. όσο και στον διακομιστή με τα μεταφορτωμένα αρχεία. Με το τέλος της εκτέλεσης εμφανίζεται ενημερωτικό μήνυμα επιτυχημένες εκτέλεσης (γραμμή 108) και ανανεώνεται η σελίδα με την χρήση της τεχνικής Query String η οποία περιγράφεται στην ενότητα 4.5.4.

```
101 RemoveFile = (event) => {  
102   event.preventDefault();  
103   var filename = ''; //this.state.data.DocumentInfo;  
104   var id = this.state.data.ID;  
105   var path = '/platforma343/public/uploads/CompanyLogos/';  
106   fetch('http://localhost:9000/deleteLogo?path='+path+'&name='+filename+'&id='+id)  
107     .then(  
108       this.setState({message: 'Removed successfully!' })  
109     ).then(setTimeout(function(){ window.location.replace('/app/account?changes=1'); }, 1000))  
110 }
```

Εικόνα 4.5. 20: Διαγραφή εικόνας προφίλ

Κεφάλαιο 5

Έλεγχος και Αξιολόγηση Συστήματος

5.1.Εισαγωγή

5.2.Έλεγχος Λειτουργιών

5.2.1. Χειροκίνητες δοκιμές

5.2.2. Χρήση εργαλείου αυτοματοποιημένων δοκιμών

5.2.3. Δοκιμές από άτομα διαφόρων ηλικιών, γνώσεων και επαγγελματών

5.3.Αποτελέσματα Ερωτηματολογίου Αξιολόγησης Πλατφόρμας

5.4.Καθορισμός Κανόνων Σχεδιασμού και Ευχρηστίας

5.5.Αξιολόγηση Συστήματος

5.1.Εισαγωγή

Με το τέλος της υλοποίησης όλων των λειτουργιών του συστήματος και γενικά της πλατφόρμας πραγματοποιήθηκε έλεγχος και δοκιμές σχετικά με την ορθότητα και την ακρίβεια των αποτελεσμάτων. Η διαδικασία η οποία ακολουθήσαμε περιείχε έλεγχο των διαδικασιών σε όλα τα στάδια υλοποίησης μαζί με ανατροφοδότηση από την υπεύθυνη καθηγήτρια κ. Καπιτσάκη με τις αλλαγές αυτές να γίνονται κατά την διάρκεια προγραμματισμού. Επιπλέον, μαζικές δοκιμές έγιναν και τον τελευταίο μήνα ο οποίος χρησιμοποιήθηκε περισσότερο για βελτιστοποίηση του συστήματος από πλευράς αποδοτικότητας και ακρίβειας τόσο με μεμονωμένη εκτέλεση λειτουργιών όσο και σαν σύνολο.

5.2.Δοκιμή φυλλομετρητών ιστών

Στην πιο κάτω εικόνες 5.2.1 , 5.2.2 και 5.2.3 φαίνονται ενδεικτικές εικόνες για την εμφάνιση της ιστοσελίδας μας σε 3 διαφορετικούς φυλλομετρητές ιστούς (Brave, Google Chrome, Microsoft Edge). Γενικά η εμφάνιση είναι ομοιόμορφη στις περισσότερες σελίδες σε όλους τους φυλλομετρητές αλλά εμφανίζονται κάποιες πολύ μικρές διαφορές. Για παράδειγμα στο Microsoft Edge, εμφανίζεται η επιλογή να εμφανιστεί ο κωδικός πρόσβασης αυτόματα με την επιλογή του εικονιδίου του οποίου εμφανίζεται, ενώ στο Google Chrome και Brave όπως μπορούμε να δούμε και από τις εικόνες.

Sign in

Username

Password

SIGN IN NOW

Don't have an account? [Create an account](#)
[Forgot password?](#)

Εικόνα 5.2. 3: Google Chrome v91.0.4472.77

Sign in

Username

Password

SIGN IN NOW

Don't have an account? [Create an account](#)
[Forgot password?](#)

Εικόνα 5.2. 3: v1.25.68 Chromium: 91.0.4472.77

Sign in

Username

Password

SIGN IN NOW

Don't have an account? [Create an account](#)
[Forgot password?](#)

Εικόνα 5.2. 3: Microsoft Edge v91.0.864.37

5.3. Έλεγχος Λειτουργιών

5.3.1. Χειροκίνητες δοκιμές

Στο πιο κάτω πίνακα φαίνονται οι δοκιμές οι οποίες έγιναν για κάθε λειτουργία του συστήματος και για τις 3 ομάδες χρηστών της πλατφόρμας μας. Αξίζει να σημειωθεί ότι δεν ήταν σε όλη την διάρκεια των δοκιμών όλες οι δοκιμές αποδεκτές και χρειάστηκε επαναληπτικός έλεγχος με την κατάσταση αυτών να μεταβάλλεται είτε σε κατάσταση αποδοχής είτε απόρριψης.

ΔΟΚΙΜΕΣ ΛΕΙΤΟΥΡΓΙΩΝ - ΔΙΑΔΙΚΤΥΑΚΗΣ ΠΛΑΤΦΟΡΜΑΣ		
A/A	ΛΕΙΤΟΥΡΓΙΕΣ (ΤΥΠΟΣ ΛΟΓΑΡΙΑΣΜΟΥ)	ΠΕΡΙΓΡΑΦΗ ΔΟΚΙΜΩΝ
1	Εγγραφή στην ιστοσελίδα (οργανισμός)	Συμπλήρωση όλων των απαραίτητων πεδίων στην κατάλληλη μορφή Αποδοχή όρων & προϋποθέσεων πλατφόρμας

		Εμφάνισης ενημερωτικών μηνυμάτων ανάδρασης για κατάσταση εγγραφής
2	Σύνδεση στην ιστοσελίδα (οργανισμός, καθηγητής, φοιτητής)	Έλεγχος αναμενόμενης και ορθής απόκρισης ανάλογα με τα δεδομένα της Β.Δ. Εμφάνισης ενημερωτικών μηνυμάτων
3	Αλλαγή προσωπικών στοιχείων (οργανισμός, καθηγητής, φοιτητής)	Έλεγχος αναμενόμενης και ορθής απόκρισης ανάλογα με τα δεδομένα της Β.Δ. Εμφάνιση πεδίων που αντιστοιχούν στο τύπο λογαριασμού Εμφάνισης ενημερωτικών μηνυμάτων
4	Προσθήκη/ Αλλαγή/ Διαγραφή εικόνας προφίλ (οργανισμός)	Έλεγχος αναμενόμενης και ορθής απόκρισης αλλαγών στην Β.Δ. και διακομιστή Αλλαγή μεγέθους εικόνας σε μεταφόρτωση διαφορετικών διαστάσεων από 300X300 pixels από τον χρήστη
5	Προσθήκη/Αλλαγή/Διαγραφή γνώσεων (φοιτητής)	Έλεγχος αναμενόμενης και ορθής απόκρισης αλλαγών στην Β.Δ. Προσθήκη μεγάλου αριθμού γνώσεων Ταυτόχρονες αλλαγές (προσθήκη & αφαίρεση) Εμφάνισης ενημερωτικών μηνυμάτων
6	Αλλαγή Κωδικού (οργανισμός, καθηγητής, φοιτητής)	Έλεγχος αναμενόμενης και ορθής απόκρισης αλλαγών στην Β.Δ. Εμφάνισης ενημερωτικών μηνυμάτων
7	Ανάκτηση Κωδικού (οργανισμός, καθηγητής, φοιτητής)	Έλεγχος αναμενόμενης και ορθής απόκρισης πληροφοριών από την Β.Δ. Αποστολή μηνύματος ενημέρωσης στο ηλεκτρονικό ταχυδρομείο Εμφάνισης ενημερωτικών μηνυμάτων
8	Αλλαγή ρυθμίσεων για ειδοποιήσεις (οργανισμός, καθηγητής, φοιτητής)	- Έλεγχος αναμενόμενης και ορθής απόκρισης αλλαγών στην Β.Δ. - Εμφάνισης ενημερωτικών μηνυμάτων

9	Προβολή κανόνων GDPR (οργανισμός, καθηγητής, φοιτητής)	Προβολή αναμενόμενου αποτελέσματος
10	Δημιουργία και αποθήκευση στοιχείων GDPR για τον λογαριασμό του χρήστη (οργανισμός, καθηγητής, φοιτητής)	Έλεγχος αναμενόμενης και ορθής απόκρισης πληροφοριών της Β.Δ και του διακομιστή Εμφάνισης ενημερωτικών μηνυμάτων
11	Αίτηση για διαγραφή λογαριασμού (οργανισμός, καθηγητής, φοιτητής)	Αποστολή μηνύματος ενημέρωσης στο ηλεκτρονικό ταχυδρομείο της πλατφόρμας Έλεγχος αναμενόμενης και ορθής απόκρισης αλλαγών στην Β.Δ.
12	Περιορισμός χρήσης στοιχείων GDPR λογαριασμού (οργανισμός)	Έλεγχος αναμενόμενης και ορθής απόκρισης αλλαγών στην Β.Δ Αλλαγή των στοιχείων χρήσης της πλατφόρμας στους άλλους χρήστες
13	Δημιουργία/Επεξεργασία/Διαγραφή Ε.Α (οργανισμός, καθηγητής)	Έλεγχος αναμενόμενης και ορθής απόκρισης αλλαγών στην Β.Δ Δημιουργία όλων των τύπων Ε.Α. Μεταφόρτωση και διαγραφή αρχείων Ταξινόμηση Ε.Α. με κάθε τρόπο Έλεγχος εμφάνισης κατάλληλων δυνατοτήτων ανάλογα με τύπο χρήστη (π.χ. ο καθηγητής να μπορεί να τα διαγράψει τα Ε.Α.) Εμφάνισης ενημερωτικών μηνυμάτων
14	Επιλογή Ο.Α.Ε (οργανισμός, καθηγητής)	Έλεγχος αναμενόμενης και ορθής απόκρισης αλλαγών στην Β.Δ Προβολή στοιχείων/πληροφοριών ομάδων που έδειξαν ενδιαφέρον με προσωπικά στοιχεία και χωρίς Δυνατότητα αναίρεσης ανάθεσης
16	Δημιουργία/Επεξεργασία/Διαγραφή/ Προβολή παραδοτέων (καθηγητής)	Έλεγχος αναμενόμενης και ορθής απόκρισης αλλαγών στην Β.Δ

		Δημιουργία / Διαγραφή παραδοτέων Μεταφόρτωση και διαγραφή αρχείων Έλεγχος εμφάνισης κατάλληλων δυνατοτήτων ανάλογα με τύπο χρήστη (π.χ. ο καθηγητής να μπορεί να τα διαγράψει) Δυνατότητα απόκρυψης από φοιτητές Εμφάνισης ενημερωτικών μηνυμάτων
17	Δημιουργία/Επεξεργασία/Διαγραφή / Προβολή ερωτηματολόγια (καθηγητής)	Έλεγχος αναμενόμενης και ορθής απόκρισης αλλαγών στην Β.Δ Δημιουργία / Επεξεργασία / Διαγραφή ερωτηματολογίων Έλεγχος εμφάνισης κατάλληλων δυνατοτήτων ανάλογα με τύπο χρήστη (π.χ. ο καθηγητής να μπορεί να τα διαγράψει) Δυνατότητα απόκρυψης από φοιτητές Εμφάνισης ενημερωτικών μηνυμάτων
18	Βαθμολόγηση παραδοτέων ανά ομάδα και ανά φοιτητή (καθηγητής)	Έλεγχος αναμενόμενης και ορθής απόκρισης αλλαγών στην Β.Δ Συμπλήρωση όλων των απαραίτητων πεδίων στην κατάλληλη μορφή Αποστολή ενημερωτικού μηνύματος στο ηλεκτρονικό ταχυδρομείο για 1 άτομο και για κάθε άτομο της ομάδας Ανάρτηση βαθμών για ένα άτομο και για ομάδα ατόμων Μεταφόρτωση και διαγραφή αρχείων Εμφάνισης ενημερωτικών μηνυμάτων
19	Αποστολή και προβολή αιτημάτων σε φοιτητές (φοιτητής)	Έλεγχος αναμενόμενης και ορθής απόκρισης αλλαγών στην Β.Δ Συμπλήρωση όλων των απαραίτητων πεδίων στην κατάλληλη μορφή

		Αποστολή ενημερωτικού μηνύματος στην πλατφόρμα στο παραλήπτη Δυνατότητα αποδοχής/απόρριψης και απόκρυψης αιτημάτων Δυνατότητα αναίρεσης αιτήματος Εμφάνιση ενημερωτικών μηνυμάτων
20	Δημιουργία/Διαγραφή/Επικύρωση / Επεξεργασία ομάδας (καθηγητής)	Έλεγχος αναμενόμενης και ορθής απόκρισης αλλαγών στην Β.Δ Συμπλήρωση όλων των απαραίτητων πεδίων στην κατάλληλη μορφή Έλεγχος αριθμού ατόμων σε ομάδες Προβολή ατόμων χωρίς ομάδα Επικύρωση / Διαγραφή / Δημιουργία / Επεξεργασία ομάδων Εμφάνιση ενημερωτικών μηνυμάτων
21	Δημιουργία μαζικών λογαριασμών φοιτητών (καθηγητής)	Έλεγχος αναμενόμενης και ορθής απόκρισης αλλαγών στην Β.Δ Φοιτητές οι οποίοι γράφονται στο μάθημα δεύτερη φορά να σημειώνεται στην Β.Δ. Αποστολή στοιχείων εισόδου στην πλατφόρμα στο ηλεκτρονικό τους ταχυδρομείο Έλεγχος μορφής CSV αρχείου Δυνατότητα μεταφόρτωσης αρκετά μεγάλου αρχείου

5.3.2. Χρήση εργαλείου αυτοματοποιημένων δοκιμών

Το εργαλείο το οποίο χρησιμοποιήθηκε για αυτοματοποιημένες δοκιμές είναι το Selenium IDE το οποίο έχει σχεδιαστεί για να καταγράφει την αλληλεπίδραση του χρήστη σε οποιοδήποτε ιστότοπο για να καταργηθεί η ανάγκη των χειροκίνητων επαναληπτικών λήψεων.⁵ Είναι ένα ανοικτού κώδικα λογισμικό με αυτοματοποιημένες δοκιμές με σκοπό τον εύκολο εντοπισμό

⁵ <https://chrome.google.com/webstore/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd>

σφαλμάτων με την χρήση ελέγχου ροής επαναληπτικών διαδικασιών (if, while κτλ.) και σημεία διακοπής μόλις εντοπιστεί ένα λάθος για να ενημερώνει τον χρήστη σχετικά με τον λόγο αποτυχίας έτσι να είναι πιο εύκολη και οι διόρθωση (debugging). Οι αυτοματοποιημένες δοκιμές επικεντρώθηκαν στην ορθή συμπλήρωση όλων των φορμών που περιέχει η σελίδα μας, δημιουργία και επεξεργασία παραδοτέων / ερωτηματολογίων / Ε.Α. σε μεγάλους αριθμούς αλλά και δοκιμές ασφάλειας της σελίδα όσο αφορά την προσπέλαση των ελέγχων που εκτελούνται από προγραμματιστικής πλευράς από το σύστημα μας [23].

5.3.3. Δοκιμές από άτομα διαφορετικών ηλικιών, γνώσεων και επαγγελματιών

Αναθέσαμε σε άτομα διαφορετικών ηλικιών με διαφορετικό μορφωτικό επίπεδο και γνώσεις να δοκιμάσουν την διαδικτυακή εφαρμογή με σκοπό να δούμε πόσο εύκολη στην χρήση είναι σε τυχαίο δείγμα χρηστών. Οι χρήστες δοκίμασαν τις λειτουργίες της ιστοσελίδας και μας εξέφρασαν την άποψη και τα σχόλια τους σχετικά με αυτή. Οι περισσότερες δοκιμές έγιναν από φοιτητές του Πανεπιστημίου Κύπρου, για τύπο λογαριασμού φοιτητή, από διάφορα ακαδημαϊκά έτη, οι περισσότεροι 3^ο και 4^ο. οι οποίοι είχαν παρακολουθήσει ήδη το μάθημα ΕΠΛ343 και γνώριζαν τις απαιτήσεις που υπήρχαν. Επιπλέον, δοκιμές έγιναν και από άτομα μεγαλύτερων ηλικιών οι οποίοι θα χρησιμοποιούσαν την πλατφόρμα με τύπο λογαριασμό οργανισμού και θα προσπαθήσουν να αλληλοεπιδράσουν με αυτή χωρίς κάποια προηγούμενη γνώση. Τέλος κάποιες δοκιμές έγιναν και από την υπεύθυνη καθηγήτρια για τύπο λογαριασμού καθηγητή σχετικά με την μελλοντική χρήση του συστήματος. Η ανατροφοδότηση η οποία έγινε με την χρήση ερωτηματολογίου, περιγράφεται στην ενότητα 5.3, είχε αρκετά θετικά σχόλια τόσο για την λειτουργικότητα της ιστοσελίδας όσο και για την σχεδίαση με ελάχιστες μικρές αλλαγές τις οποίες λάβαμε υπόψη και βελτιώσαμε στο σύστημα.

5.4.Αποτελέσματα Ερωτηματολογίου Αξιολόγησης Πλατφόρμας

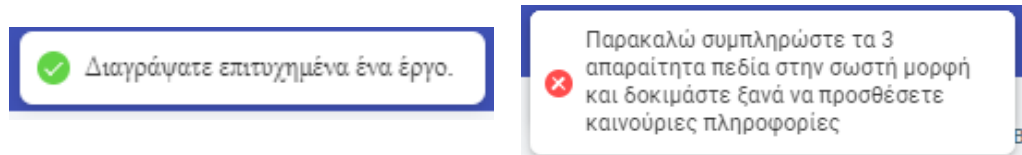
5.5.Καθορισμός Κανόνων Σχεδιασμού και Ευχρηστίας

Ένας από τους αρχικούς στόχους οι οποίοι ορίστηκαν τόσο από πλευράς καθηγητή όσο και από πλευράς φοιτητή ήταν η πλατφόρμα και συγκεκριμένα η διεπιφάνεια μαζί με την λειτουργικότητα να είναι απλή και κατανοητή από όλα τα είδη χρηστών αφού θα υπάρχουν χρήστες διάφορων ηλικιών. Αυτό μπορούμε να το επιβεβαιώσουμε αν λάβουμε υπόψη τις απαντήσεις στα ερωτηματολόγια και την ανάλυση σεναρίων που έχουν δημιουργηθεί με την χρήστη personas. Για την δημιουργία της πλατφόρμας ακολουθήθηκαν κανόνες σχεδιασμού και ευχρηστίας από τους 10 κανόνες του Nielsen αλλά και από τους κανόνες σχεδιασμού NASA. Αποτελούν μέρος της Ευρετικής Αξιολόγησης η οποία αναπτύχθηκε την δεκαετία του 90 από τον Jacob Nielsen και το 2014 το Human Integration Design Processes (HIDP) από το “National Aeronautics and Space Administration International Space Station Program Johnson

Space Center Houston, Texas”. Πιο κάτω φαίνονται αναλυτικά οι κανόνες σχεδίασης και ευχρηστίας οι οποίοι καθορίστηκαν και υλοποιήθηκαν στην πλατφόρμα:

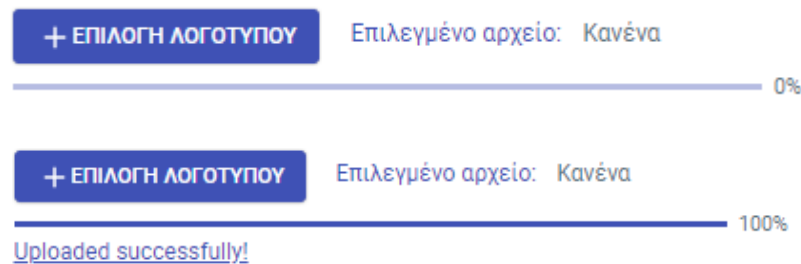
1. Ορατότητα της κατάστασης του συστήματος – Visibility of system status:

Οι χρήστες ενημερώνονται από το σύστημα για το τι συμβαίνει και την ανάδραση του συστήματος σε λειτουργίες τις οποίες εκτελούν σε εύλογο χρόνο. Με το τέλος της εκτέλεσης των περισσότερων λειτουργιών εμφανίζεται ενημερωτικό μήνυμα ανάδρασης είτε για επιτυχημένη είτε για αποτυχημένο αποτέλεσμα. Κάποια παραδείγματα φαίνονται στην Εικόνα 5.4.1 με το επιτυχημένο αποτέλεσμα στα αριστερά και το αποτυχημένο στα δεξιά.



Εικόνα 5.4. 1: Ενημερωτικά μηνύματα ανάδρασης λειτουργίας

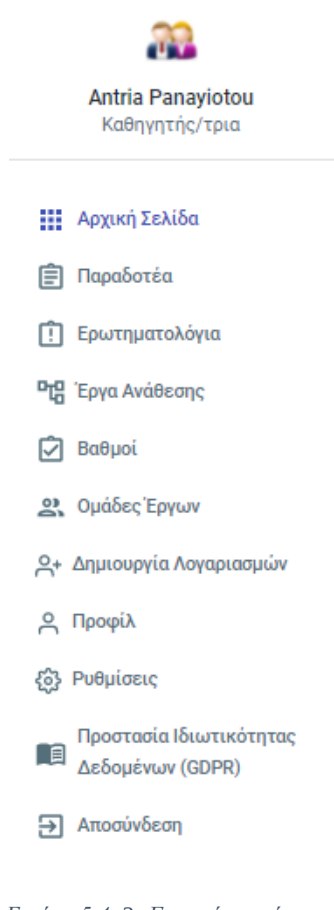
Επιπλέον, υπάρχει οριζόντια ράβδος ανάδρασης αρχικοποιημένη με το ποσοστό 0% και την χρησιμοποιούμε για τις λειτουργίες σχετικά με την μεταφόρτωση αρχείων στον διακομιστή. Το ποσοστό που εμφανίζεται αλλάζει ανάλογα με το ποσοστό μεταφόρτωσης στον διακομιστή μέχρι να φτάσει στο μέγιστο 100% το οποίο ισούται και με το επιτυχημένο τέλος της εκτέλεσης της διαδικασίας σε διακομιστή και Β.Δ. Στην Εικόνα 5.4.2. φαίνονται τα δύο στάδια της ράβδου ανάδρασης στην αρχή και στο τέλος μαζί με ένα επιπλέον ενημερωτικό μήνυμα σχετικά με την επιτυχημένη μεταφόρτωση.



Εικόνα 5.4. 2: Ράβδος ανάδρασης

2. Συσχέτιση συστήματος και πραγματικού κόσμου – Match between system and real world

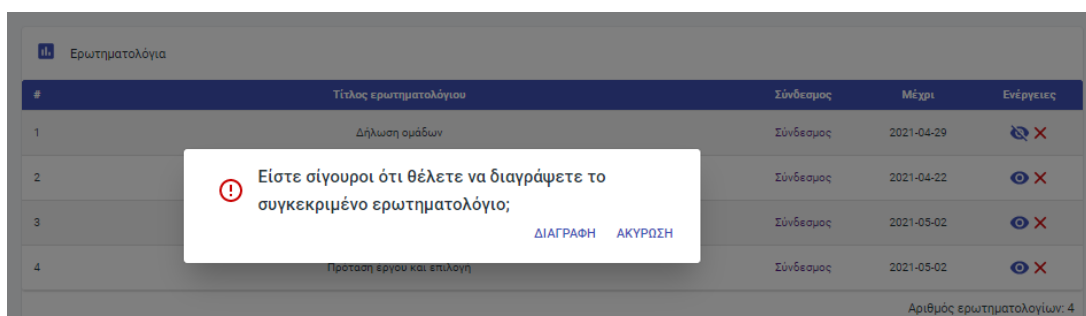
Περιέχει λέξεις, φράσεις και έννοιες οι οποίες είναι οικείες στον χρήστη και δεν είναι εξειδικευμένες έτσι ώστε να μην τις καταλαβαίνουν μόνο άτομα με γνώσεις πληροφορικής αλλά γενικά κάθε χρήστης της σελίδας. Για παράδειγμα στην γραμμή μενού δίπλα από κάθε κατηγορία υπάρχει εικονίδιο σχετικό με την σημασία της λέξης (π.χ. γρανάζι για ρυθμίσεις) όπως φαίνεται και στην Εικόνα 5.4.3.



Εικόνα 5.4. 3: Γραμμή μενού

3. Έλεγχος του συστήματος από τον χρήστη – User control and freedom

Παρέχονται σαφής και εύκολες έξοδοι διαφυγής από σημεία στα οποία οι χρήστες βρέθηκαν χωρίς να το περιμένουν μαζί με την επιλογή αναίρεσης ή διαγραφής οποιασδήποτε ενέργειας τους, όπως φαίνεται και στην Εικόνα 5.4.4. Επιπλέον για να μεταφερθούν σε κάποιο σημείο της πλατφόρμας υπάρχουν διάφοροι τρόποι π.χ. πληκτρολογώντας το URL, από την γραμμή μενού και από την χρήση υπερσυνδέσμου από κάποια άλλη σελίδα.



Εικόνα 5.4. 4: Μήνυμα επιβεβαίωσης

4. Συνέπεια και τήρηση προτύπων – Consistency and standards

Παρόμοιες ή ίδιες δράσεις εκτελούνται με τον ίδιο τρόπο σε κάθε μέρος του συστήματος για παράδειγμα τα κουμπιά έχουν ετικέτες (αποθήκευση, ακύρωση κ.λπ.) βοηθώντας έτσι

το χρήστη να καταλάβει ακριβώς τη λειτουργία τους και είναι σχεδιασμένα με τον ίδιο τρόπο σε όλες τις σελίδες. Επιπρόσθετα, σε κάθε υποσελίδα υπάρχει η ίδια γραμματοσειρά, χρώμα γραμμάτων για παρόμοιες δομές στοιχείων και μέγεθος γραμμάτων.

5. Σχεδιασμός για αποτροπή σφαλμάτων χρήστη – Error prevention

Υπάρχουν μηνύματα λάθους τα οποία είναι υποβοηθητικά και η χρησιμοποιείται απλή γλώσσα για την περιγραφή των σφαλμάτων και δίνονται τρόποι επίλυσης τους. Επιπλέον, για παράδειγμα όπως φαίνεται και στην Εικόνα 5.4.5 φαίνεται το αποτέλεσμα αν ο χρήστης δεν συμπληρώσει τα υποχρεωτικά πεδία, αν συμπληρώσει με λάθος μορφή δεδομένων (τηλέφωνο) ή με διαφορετικό τρόπο από τον αναμενόμενο (όνομα χρήστη).

Ηλεκτρονικό Ταχυδρομείο

Το πεδίο είναι υποχρεωτικό

Τηλέφωνο

123

Το τηλέφωνο δεν είναι στη σωστή μορφή

Όνομα χρήστη (username)

apanay

Ακριβώς 6 χαρακτήρες

Εικόνα 5.4. 5: Μηνύματα σφάλματος

6. Ελαχιστοποίηση του μνημονικού φορτίου - Recognition rather than recall

Τα αντικείμενα, οι δυνατές δράσεις και οι διάφορες επιλογές διαρκώς ορατές στο χρήστη όπως για παράδειγμα η γραμμή μενού για να μπορεί να προηγηθεί σε κάθε σελίδα όπως και μία μικρή περιγραφή κάτω από κάθε λειτουργία όπως φαίνεται και στην Εικόνα 5.4.6 για να υπενθυμίζει στο χρήστη της δυνατότητες του.

Ανάρτηση βαθμών για φοιτητή

Πιο κάτω μπορείτε να ανεβάσετε βαθμό για ένα συγκεκριμένο φοιτητή για ένα συγκεκριμένο παραδοτέο. Υπάρχει επιπλέον η δυνατότητα να ανεβάσετε και επιπρόσθετες πληροφορίες μέσω αρχείου PDF και σχόλιου ανατροφοδότησης επιλέγοντας τον υπερσύνδεσμο "Μεταφόρτωση περισσότερων πληροφοριών".

Επιλογή φοιτητή

Επιλογή παραδοτέου

Βαθμός /100

Μεταφόρτωση περισσότερων πληροφοριών

Ανάρτηση βαθμών ➤

Εικόνα 5.4. 6: Παράδειγμα περιγραφής λειτουργίας

7. Αποφυγή περιττών στοιχείων – Aesthetic and minimalist design

Δεν παρέχεται αχρείαστη και μη απαραίτητη πληροφορία στο χρήστη καθώς είναι εμφανές μόνο οι σημαντικές πληροφορίες οι οποίες πρέπει να γνωρίζει κάθε χρήστης.

8. Προσαρμοστικότητα - Responsibility

Η πλατφόρμα μπορεί να προσαρμοστεί σε οθόνες κινητών και υπολογιστών, ανεξάρτητα από το λειτουργικό σύστημα, το πρόγραμμα περιήγησης, το μέγεθος και την ανάλυση οθόνης της συσκευής του χρήστη. Με άλλα λόγια, η πλατφόρμα είναι προσβάσιμη μέσω οποιουδήποτε είδους ηλεκτρονικής συσκευής με σύνδεση στο Διαδίκτυο.

9. Προφανές και αναμενόμενες ιδιότητες διεπιφάνειας - Obvious and expected interface properties

Οι φόρμες που δημιούργησα είναι παρόμοιες με όλες τις φόρμες που χρησιμοποιούνται στο διαδίκτυο. Εάν κάποιος χρήστης συμπλήρωσε έστω και μια φορά φόρμα διαδικτυακά στο παρελθόν, τότε θα του είναι πολύ εύκολο να συμπληρώσει φόρμες που υπάρχουν στο σύστημα.

5.6.Αξιολόγηση Συστήματος

Με το τέλος της υλοποίησης τα άτομα τα οποία δοκίμασαν την σελίδα απάντησαν ένα ερωτηματολόγιο για να καταγραφούν τα συμπεράσματα τους. Οι περισσότεροι χρήστες ήταν φοιτητές του τμήματος πληροφορικής του Πανεπιστημίου Κύπρου οι οποίοι έχουν ήδη παρακολουθήσει το μάθημα και γνώριζαν τις απαιτήσεις. Η γνώμη τους είναι σημαντική καθώς θα μπορούσαν αντικειμενικά να συγκρίνουν καταστάσεις και να αξιολογήσουν την πλατφόρμα με αντικειμενικό τρόπο. Το ερωτηματολόγιο έχει απαντηθεί από περίπου 30 άτομα (είναι προσβάσιμο από το ΠΑΡΑΡΤΗΜΑ Γ – Γ.1) με διαφορετικές απόψεις και γνώσεις για το μάθημα καθώς είναι διαφορετικών ηλικιών, φύλου και ακαδημαϊκού εξαμήνου. Με το τέλος του ερωτηματολογίου δημιουργήθηκαν γραφικές παραστάσεις με τις ανάγκες των φοιτητών οι οποίες αντικατοπτρίζουν τα αποτελέσματα του ερωτηματολογίου (είναι προσβάσιμο από το ΠΑΡΑΡΤΗΜΑ Γ – Γ.2)

Τα γενικά συμπεράσματα του ερωτηματολογίου εμφανίζονται πιο κάτω:

- Οι περισσότερες φοιτητές οι οποίοι απάντησα το ερωτηματολόγιο είναι κυρίως άντρες (λογικό αφού αποτελούν την πλειοψηφία του τμήματος Πληροφορικής) - 12 – άντρες , 7 – γυναίκες.
- Οι περισσότεροι από τους δοκιμαστές της πλατφόρμας δοκίμασαν την σελίδα και από τους τρεις διαφορετικούς τύπους λογαριασμού γι' αυτό έχουν μία σφαιρική άποψη. Συγκεκριμένα 16 δοκιμές ως φοιτητής, 16 ως καθηγητής και 14 ως οργανισμός.
- Όλα τα άτομα βρήκαν την ύπαρξη μίας τέτοιας ιστοσελίδας χρήσιμη.
- Η πλειοψηφία των ατόμων θεώρησε πάρα πολύ εύκολη την εκμάθηση της ιστοσελίδας ενώ οι υπόλοιποι δήλωσαν ότι είναι πολύ εύκολο γεγονός που διαχωρίζει και κατά πόσο έχουν δημιουργήσει ήδη το νοητικό μοντέλο στο μυαλό τους για μία τέτοια ιστοσελίδα.

- Όλα τα άτομα δήλωσαν από πολύ ως πάρα πολύ ότι η πλατφόρμα είναι χρήσιμη για την δημιουργία και ανάθεση έργων.
- Σχεδόν όλοι δήλωσαν ότι η οργάνωση της σελίδας είναι πολύ κατανοητή.
- Στα γενικά σχόλια όπου οι χρήστες μπορούσαν να δηλώσουν την ελεύθερη τους άποψη πολλά είναι τα σημεία τα οποία δήλωσαν ότι τους άρεσαν, μερικά από αυτά είναι η μοντέρνα και φιλική διεπαφή, η ευκολία χρήσης αλλά και η δυνατότητα προβολής και επιλογής έργων ανάθεσης. Επίσης πρότειναν και μερικές βελτιώσεις και λειτουργίες οι οποίες δεν βρίσκονται στην σελίδα και θα τις θεωρούσαν χρήσιμες γεγονός το οποίο συμφωνούμε και περιγράφεται καλύτερα στην ενότητα 6.1 με την επεκτασιμότητά του συστήματος.

Κεφάλαιο 6

Συμπεράσματα

6.1.Επεκτασιμότητα Συστήματος

6.2.Ανασκόπηση Διπλωματικής Εργασίας

6.3.Γενικά Συμπεράσματα

6.1.Επεκτασιμότητα Συστήματος

Το σύστημα το οποίο υλοποιήσαμε περιέχει μία μερίδα από τις πιο σημαντικές και απαραίτητες λειτουργίες για την διαχείριση του μαθήματος. Είναι πλήρως υλοποιημένες και το σύστημα μπορεί να χρησιμοποιηθεί άμεσα στα πλαίσια του μαθήματος. Λόγω του χρονικού περιθωρίου για την εκπόνηση της διπλωματικής εργασίας δεν ήταν δυνατό να υλοποιηθούν πλήρως όλες οι λειτουργίες οι οποίες χρειάζονται και θα βοηθούσαν ακόμη περισσότερο στην διαχείριση του μαθήματος. Επιπλέον, αφού λάβαμε υπόψη μας το ερωτηματολόγιο αξιολόγησης από τους χρήστες τις σελίδας μας το οποίο αναλύσαμε στο κεφάλαιο 5.3 συμπεράναμε ότι υπάρχουν λειτουργίες οι οποίες οι χρήστες θεωρούσαν χρήσιμες να υπάρχουν στην σελίδα μας. Ίσως σε κάποιο μεταγενέστερο στάδιο να γίνει επέκταση του υπάρχον συστήματος με επιπρόσθετες λειτουργίες οι οποίες θα πρόσφεραν περισσότερες δυνατότητες και ευελιξία στους χρήστες. Κάποιες από τις πιθανές βελτιωτικές κινήσεις οι οποίες θα μπορούσαν να γίνουν είναι οι ακόλουθες:

- Λειτουργία συζήτησης / Φόρουμ: Με την συγκεκριμένη λειτουργία θα δίνεται η δυνατότητα στους φοιτητές να κάνουν ερωτήσεις και στην συνέχεια άλλοι φοιτητές ή καθηγητές να απαντούν ανάλογα. Επίσης, θα μπορούν οι καθηγητές να τοποθετούν και αυτοί ανακοινώσεις προς τους φοιτητές με την δυνατότητα να τους ενημερώνει και στο ηλεκτρονικό τους ταχυδρομείο.
- Εμφάνιση ειδοποιήσεων: Οι χρήστες να μπορούν να δουν εύκολα με κάποιο είδος ειδοποίησης στην πλατφόρμα για αλλαγές, οι ενημερώσεις τις οποίες τους ενδιαφέρουν π.χ. αίτημα για εισδοχή σε ομάδα κτλ..
- Ανάρτηση παραδοτέων: Οι χρήστες να έχουν την δυνατότητα να ανεβάσουν τα παραδοτέα τους στην σελίδα όπου εμφανίζονται οι ημερομηνίες παράδοσης.

- Μεταφόρτωση εκπαιδευτικού υλικού: Οι καθηγητές να έχουν την δυνατότητα ανάρτησης υλικού όπως διαλέξεις, αρχεία κτλ. έτσι ώστε οι φοιτητές να έχουν πρόσβαση σε αυτά εύκολα.
- Δυνατότητα προβολής στοιχείων επικοινωνίας των εταιριών: Θα ήταν ωραίο αν οι καθηγητές και οι φοιτητές θα μπορούσαν να δουν τα πιο σημαντικά στοιχεία επικοινωνίας για εταιρίες έτσι ώστε να μπορούν να επικοινωνήσουν εύκολα με αυτούς.
- Λειτουργία αποδοχής ανάθεσης έργου: Υπάρχει πιθανότητα κάποιες ομάδες να δεχτούν περισσότερο από μία προτάσεις για να υλοποιήσουν κάποιο έργο. Θα ήταν ωραίο να υπάρχει η δυνατότητα να επιλέξει η ομάδα πιο έργο από όλα επιθυμεί περισσότερο να υλοποιήσει.
- Δυνατότητα επεξεργασίας των στοιχείων της Β.Δ.: Να δίνεται η δυνατότητα στους καθηγητές να επεξεργαστούν όλα τα στοιχεία που υπάρχουν στην Β.Δ. από την σελίδα χωρίς να χρειάζεται σε κάποιες στιγμές γνώση και πρόσβαση από τον καθηγητή στην Β.Δ.

Πιο πάνω φαίνονται κάποιες ιδέες οι οποίες θα ήταν ωραίες να υπάρχουν και θα πρόσθεταν μοναδική εμπειρία χρήσης τόσο σε κάθε χρήστη της σελίδας ασχέτως από τον τύπο λογαριασμό του.

6.2.Ανασκόπηση Διπλωματικής Εργασίας

Εξαρχής σκοπός της διπλωματικής εργασίας ήταν η βελτίωση του τρόπου με τον οποίο οι καθηγητές επικοινωνούσαν με τους φοιτητές και τους οργανισμούς, οι οποίοι ανάθεταν Ε.Α. στο μάθημα ΕΠΛ343. Με το τέλος της διπλωματικής εργασίας ο στόχος αυτός θεωρούμε ότι επιτυγχάνεται σε μεγάλο βαθμό καθώς κύριες λειτουργίες υλοποιήθηκαν πλήρως και η πλατφόρμα μπορεί να χρησιμοποιηθεί χωρίς να υπάρχει οποιοδήποτε πρόβλημα από προγραμματιστική άποψη. Η πλατφόρμα η οποία δημιουργήθηκε είναι εύχρηστη και εκσυγχρονισμένη με σύγχρονες τεχνολογίες οι οποίες πρωταγωνιστούν σε παρόμοιες εφαρμογές και υπάρχει σχετικά γρήγορη ανταπόκριση στα αιτήματα των χρηστών.

Κατά τη διάρκεια της υλοποίησης του συστήματος ακολούθησα καλές πρακτικές προγραμματισμού τις οποίες διδάχτηκα στα μαθήματα ΕΠΛ343 - Τεχνολογία Λογισμικού και ΕΠΛ449 - Επαγγελματική Πρακτική Τεχνολογίας Λογισμικού τα οποία συνέβαλαν στο μέγιστο για να ακολουθήσω την αναμενόμενη πορεία ανάπτυξης συστήματος. Επιπλέον σημαντικό ήταν και το μάθημα ΕΠΛ435 - Αλληλεπίδραση Ανθρώπου Υπολογιστή το οποίο βοήθησε για την υλοποίηση ενός εύχρηστου συστήματος, με καλές αρχές σχεδίασης και σωστή διάταξη οθονών.

Οι υπάρχουσες γνώσεις δεν ήταν αρκετές όμως για την υλοποίηση ιστοσελίδας μεγάλης κλίμακας ιδιαίτερα με την χρήση framework τόσο για backend όσο και για frontend. Σημαντικό ρόλο στις γνώσεις τις οποίες απέκτησα ήταν η παρακολούθηση διαδικτυακών εκπαιδευτικών βίντεο (online tutorials) από την ιστοσελίδα Udemy σχετικά με τις τεχνολογίες Node JS, React JS, Redux. Με το τέλος των βίντεο εκβάθυνσης αποκτήθηκαν οι βασικές γνώσεις έτσι ώστε να η αρχή της υλοποίησης να είναι πιο βατή.

Ωστόσο όπως είναι λογικό, υπήρχαν και κάποιες προκλήσεις τις οποίες στο τέλος αντιμετωπίστηκαν με επιτυχία και πλέον το σύστημα λειτουργεί χωρίς κανένα πρόβλημα. Συγκεκριμένα, πρόκληση αποτέλεσε το γεγονός ότι πληροφορίες στέλνονται από το frontend (React) στο backend (Express – Node JS) και στοιχεία επιστρέφουν πίσω ως ανατροφοδότηση της κατάστασης. Όλες οι τεχνολογίες τρέχουν στο ίδιο σύστημα σε ένα Virtual Machine σε διαφορετικά ports άρα η σύνδεση θα ήταν localhost και μαζί με την Β.Δ. το πρόβλημα αυτό αντιμετωπίστηκε με το πέρας κάποιου χρονικού διαστήματος και την εξοικείωση του τρόπου σκέψης.

Ακόμη μία δυσκολία η οποία στην αρχή αποτελούσε πρόβλημα ήταν η δημιουργία ιστοσελίδας με αρχιτεκτονική RESTfull API και άρα η εναλλαγή της δομής και του περιεχομένου ανάλογα με τον τύπο λογαριασμού ο οποίος θα έκανες σύνδεση.

Τέλος, η κατανόηση της μεθοδολογίας Redux χρειάστηκε αρκετό χρόνο εξοικείωσης ώστε το σύστημα να αντιλαμβάνεται τα στοιχεία του χρήστη όταν κάνει σύνδεση και να αναγνωρίζει της λειτουργίες τις οποίες έχει την δυνατότητα να εκτελέσει στην πλατφόρμα μας. Αξίζει να σημειωθεί ότι ήταν αρκετά σημαντικό να ακολουθηθεί προσεκτική πορεία υλοποίησης ώστε να μην παρουσιάζονται ευαίσθητες πληροφορίες στους χρήστες που δεν είναι ασφαλές και σωστό να γνωρίζουν.

6.3.Γενικά Συμπεράσματα

Το σύστημα είναι σε θέση να υποστηρίξει μεγάλη μερίδα χρηστών χωρίς προβλήματα ασφάλειας και λειτουργικότητας. Υπάρχουν διαφορετικές επιλογές ανάλογα με τον τύπο σύνδεσης και το περιβάλλον είναι φιλικό και εύχρηστο. Η ιστοσελίδα είναι έτοιμη για χρήση και υποστηρίζει τις απαιτήσεις της μεγαλύτερης ομάδας χρηστών καθώς οι απαιτήσεις καθορίστηκαν από άτομα τα οποία γνώριζαν τις απαιτήσεις και τις υποχρεώσεις που έχει κάποιος για το σύστημα.

Γενικά, στον κόσμο της πληροφορικής και των τεχνολογιών μεγάλα ποσά ξοδεύονται για ανάπτυξη συστημάτων τα οποία κάνουν την ζωή των ανθρώπων πιο εύκολη. Με την υλοποίηση της διπλωματικής αυτής πιστεύω ότι κέρδισα πάρα πολλές καινούριες γνώσεις κάτι που με κάνει πολύ χαρούμενη καθώς είναι ακριβώς ο τομέας που θα ήθελα να ασχοληθώ και

στο μέλλον. Η δυνατότητα που μου δόθηκε να αναλάβω ένα τόσο μεγάλο έργο από την αρχή μέχρι το τέλος θεωρώ είναι μία μοναδική ευκαιρία που με βοήθησε να αποκτήσω γνώσεις και λύσεις προβλημάτων για κάθε ένα από τα στάδια της διαδικασίας ανάπτυξης. Κάθε στιγμή της διπλωματικής εργασίας από την αρχή στο καθορισμό απαιτήσεων μέχρι την αξιολόγηση του τελικού έργου με δοκιμές οι οποίες είναι ζωτικής σημασίας, μου πρόσφεραν γνώσεις οι οποίες ακράδαντα πιστεύω θα μου είναι απαραίτητες για το μέλλον.

Με την ολοκλήρωση αυτής της διπλωματικής εργασίας είμαι πιο ολοκληρωμένη ως προγραμματίστρια και φυσικά πιο έμπειρη μετά την υλοποίηση ενός αρκετά μεγάλου έργου το οποίο ανάλαβα εξ' ολοκλήρου. Επιπρόσθετα, είμαι πλέον πιο έτοιμη και σίγουρη για τον εαυτό μου αλλά και για την διαχείριση που χρειάζονται έργα με τόσο μεγάλο φόρτο εργασίας.

Βιβλιογραφία

- [1] BISCOBING, JACQUELINE, “What is a Relational Database?”, SearchDataManagement, 2021 Ανακτήθηκε από: <https://searchdatamanagement.techtarget.com/definition/relational-database>
- [2] KIDECHA, SANJAY, “Native vs. Hybrid vs. Cross-Platform: How and What to Choose?” - DZone Web Dev. *dzone.com* , 2020 Ανακτήθηκε από: <https://dzone.com/articles/native-vs-hybrid-vs-cross-platform-how-and-what-to>
- [3] NWAR, RAMI, “Why Choose ReactJS for your Next Project” | Eastern Peak. *Eastern Peak - Technology Consulting & Development Company* , 2017 Ανακτήθηκε από: <https://easternpeak.com/blog/why-choose-reactjs-for-your-next-project/>
- [4] PISUWALA, UBAID, The benefits of ReactJS and reasons to choose it for your project. *Peerbits* Ανακτήθηκε από: <https://www.peerbits.com/blog/reasons-to-choose-reactjs-for-your-web-development-project.html>
- [5] Stack Overflow Developer Survey 2020. *Stack Overflow* Ανακτήθηκε από: <https://insights.stackoverflow.com/survey/2020>
- [6] HTML - Βικιπαίδεια, 2021. *El.wikipedia.org* Ανακτήθηκε από: <https://el.wikipedia.org/wiki/HTML>
- [7] Visual Studio Code - Wikipedia, 2021. *En.wikipedia.org* Ανακτήθηκε από: https://en.wikipedia.org/wiki/Visual_Studio_Code
- [8] CSS - Βικιπαίδεια, 2021. *El.wikipedia.org* Ανακτήθηκε από: <https://el.wikipedia.org/wiki/CSS>
- [9] JavaScript - Βικιπαίδεια, 2021. *El.wikipedia.org* Ανακτήθηκε από: <https://el.wikipedia.org/wiki/JavaScript>
- [10] Node JS - Βικιπαίδεια, 2020. *El.wikipedia.org* Ανακτήθηκε από: <https://el.wikipedia.org/wiki/Nodejs>
- [11] CAPAN, TOMISLAV, 2021, Why The Hell Would I Use Node.js? A Case-by-Case Tutorial. *Toptal Engineering Blog* Ανακτήθηκε από: <https://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js>
- [12] SURVE, SURAJ, 2021, Why You Should Use React.js For Web Development. *freeCodeCamp.org* Ανακτήθηκε από: <https://www.freecodecamp.org/news/why-use-react-for-web-development/>

- [13] Redux - Βικιπαίδεια, 2020. El.wikipedia.org Ανακτήθηκε από:
[https://en.wikipedia.org/wiki/Redux_\(JavaScript_library\)](https://en.wikipedia.org/wiki/Redux_(JavaScript_library))
- [14] COPEs, FLAVIO, 2018, HTTP requests using Axios. Flaviocopes.com
Ανακτήθηκε από: <https://flaviocopes.com/axios/>
- [15] Query string - Βικιπαίδεια, 2020. El.wikipedia.org Ανακτήθηκε από:
https://en.wikipedia.org/wiki/Query_string
- [16] <https://javascript.info/formdata>
- [17] Cem Eygi, Media Query CSS Tutorial – Standard Resolutions, CSS Breakpoints, and Target Phone Sizes, Απρίλη 20220 Αναρτήθηκε από :
<https://www.freecodecamp.org/news/css-media-queries-breakpoints-media-types-standard-resolutions-and-more/#:~:text=320px%20%E2%80%94%20480px%3A%20Mobile%20devices,%E2%80%94%201200px%3A%20Desktops%2C%20large%20screens>
- [18] Jennifer Preece, Yvonne Rogers, Helen Sharp, Interaction Design: Beyond Human-Computer Interaction, John Wiley & Sons, Second Edition, 2007
- [19] John Haydon (2017, December 06), How to create user personas from
https://www.huffpost.com/entry/user-personas-websites_b_1793594?guccounter=1
- [20] “JavaScript Injection” Διαθέσιμο από: <http://www.testingsecurity.com/how-to-test/injection-vulnerabilities/JavaScript-Injection>
- [21] “SQL injection”. [Ηλεκτρονικό]. Διαθέσιμο από:
https://www.owasp.org/index.php/SQL_Injection [ημερομηνία πρόσβασης στη πηγή: 10/05/2018]
- [22] “Κώδικες για αποτροπή Injections”, [Ηλεκτρονικό]. Διαθέσιμο από:
<https://docs.apigee.com/api-platform/reference/policies/regular-expression-protection>
- [23] <https://www.selenium.dev/selenium-ide/>
- [24] “REST”, [Ηλεκτρονικό]. Διαθέσιμο από:
<http://www.users.dpem.tuc.gr/gougousis/rest/>
- [25] Moodle - Βικιπαίδεια, 2021. El.wikipedia.org Ανακτήθηκε από:
<https://el.wikipedia.org/wiki/Moodle>
- [26] Piazza - Βικιπαίδεια, 2021. El.wikipedia.org Ανακτήθηκε από:
[https://en.wikipedia.org/wiki/Piazza_\(web_service\)](https://en.wikipedia.org/wiki/Piazza_(web_service))
- [27] Blackboard - Βικιπαίδεια, 2021. El.wikipedia.org Ανακτήθηκε από:
https://en.wikipedia.org/wiki/Blackboard_Learn

Παράρτημα

Α Ερωτηματολόγιο δημιουργίας πλατφόρμας

A.1. Ερωτηματολόγιο

A.2. Γραφικές Παραστάσεις

A.1. Ερωτηματολόγιο

Υλοποίηση Πλατφόρμας για ΕΠΛ343

Στα πλαίσια της ατομικής διπλωματικής μου εργασίας θα δημιουργήσουμε μία διαδικτυακή πλατφόρμα με σκοπό τη διαχείριση του υποχρεωτικού μαθήματος ΕΠΛ343. Θα δίνεται στους φοιτητές αλλά και στον διδάσκοντα η δυνατότητα ευκολότερης ανάθεσης έργων λογισμικού για την ομαδική εργασία του μαθήματος, παράδοσης εργασιών αλλά και πρόσβασης στο εκπαιδευτικό υλικό του μαθήματος. Κάθε φοιτητής θα έχει τον δικό του λογαριασμό έτσι ώστε να μπορεί να έχει πρόσβαση στις δυνατότητες τις σελίδας, ομοίως και οι εταιρίες που θα δίνουν έργα για υλοποίηση θα μπορούν να δημιουργούν λογαριασμό και στη συνέχεια να τις δημοσιεύουν.

Με την εμπειρία που αποκτήσατε από το μάθημα ΕΠΛ343 θα θέλαμε την γνώμη σας σχετικά με τις ανάγκες που θα θεωρούσατε χρήσιμες να υπάρχουν στην ιστοσελίδα έτσι ώστε να υπάρχει ευκολία διαχείρισης του μαθήματος και από την δική σας πλευρά αλλά και από την πλευρά του διδάσκοντα.

Τα δεδομένα θα χρησιμοποιηθούν ανώνυμα για τη διπλωματική μου εργασία και για σκοπούς υλοποίησης της πλατφόρμας. Με τη συμπλήρωση του ερωτηματολογίου, συμφωνείτε με το παραπάνω.

*** Απαιτείται**

Φύλο: *

☐ Άντρας

☐ Γυναίκα

Έτος σπουδών: *

- ☐ 3^ο
- ☐ 4^ο
- ☐ 5^ο
- ☐ 6^ο
- ☐ Απόφοιτος/η

Θα θεωρούσες χρήσιμη την ύπαρξη μίας τέτοιας πλατφόρμας διαχείρισης; *

- ☐ Ναι
- ☐ Όχι

Ποιό/ά από τα παρακάτω θα θεωρούσες χρήσιμα να υπάρχουν: *

- ☐ Η διαδικτυακή (online) συμπλήρωση των παραδοτέων
- ☐ Να δίνεται η δυνατότητα από τον project manager να μοιράσει εργασία στα άλλα μέλη της ομάδας
- ☐ Μία μικρή περιοχή συζήτησης (forum) κάτω από κάθε διάλεξη με ερωτήσεις και απαντήσεις σχετικά με το κάθε κεφάλαιο
- ☐ Δυνατότητα βαθμολόγησης κάθε διάλεξης από τους φοιτητές για ανατροφοδότηση του διδάσκοντα σε περίπτωση που κάποια διάλεξη δεν είναι αρκετά επεξηγηματική
- ☐ Σελίδα όπου περιέχει χρήσιμους συνδέσμους και ιστοσελίδες ανάλογα με το κάθε έργο υλοποίησης (π.χ. app, ιστοσελίδα κτλ.)
- ☐ Πρόσβαση σε παραδείγματα παλαιότερων έργων από φοιτητές
- ☐ Μία σελίδα με μαζεμένες όλες τις φόρμες ανατροφοδότησης και η συμπλήρωση τους μέσα στην πλατφόρμα
- ☐ Να δίνεται η δυνατότητα να εκφράσει 1 ομάδα προτίμηση για κάποιο από τα υφιστάμενα έργα στην αρχή του εξαμήνου
- ☐ Να μπορεί να γίνεται review το παραδοτέο μίας ομάδας από μία άλλη με την δυνατότητα να προσθέσει και κάποιο σχόλιο από κάτω
- ☐ Άλλο: _____

Ποιό θεωρείς πιο χρήσιμο τρόπο ανατροφοδότησης από τον διδάσκοντα *

- ☐ Μέσω email (σε κάθε μέλος της ομάδας)
- ☐ Μέσω email (στην ομάδα)
- ☐ Ακριβώς κάτω από το κάθε παραδοτέο στην πλατφόρμα
- ☐ Μέσω email (σε κάθε μέλος της ομάδας) ΚΑΙ ακριβώς κάτω από το κάθε παραδοτέο στην πλατφόρμα
- ☐ Μέσω email (στην ομάδα) ΚΑΙ ακριβώς κάτω από το κάθε παραδοτέο στην πλατφόρμα

Θα ήθελες να προτείνεις μήπως κάποια εισήγηση σχετικά με την δημιουργία της πλατφόρμας;

Η απάντησή σας

Υποβολή

Μην υποβάλετε ποτέ κωδικούς πρόσβασης μέσω των Φορμών Google.

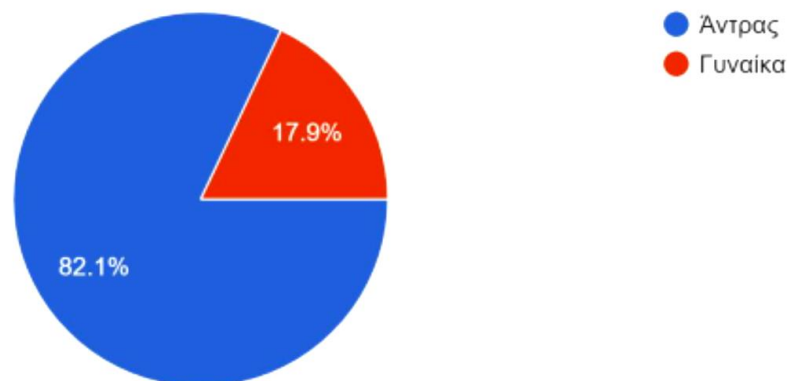
Αυτό το περιεχόμενο δεν έχει δημιουργηθεί και δεν έχει εγκριθεί από την Google. [Αναφορά κακής χρήσης - Όροι Παροχής Υπηρεσιών - Πολιτική απορρήτου](#)

Google Φόρμες

A.2. Γραφικές Παραστάσεις

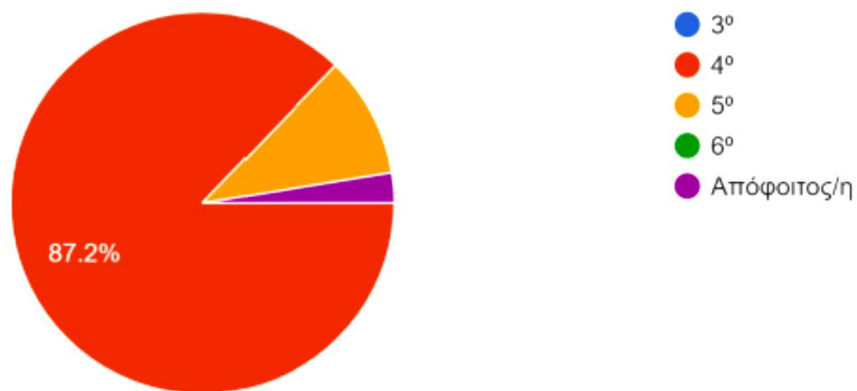
Φύλο:

39 responses



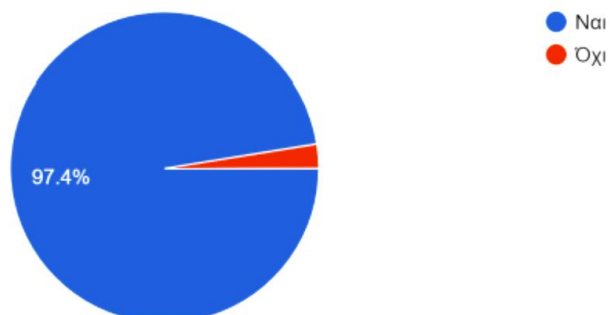
Έτος σπουδών:

39 responses



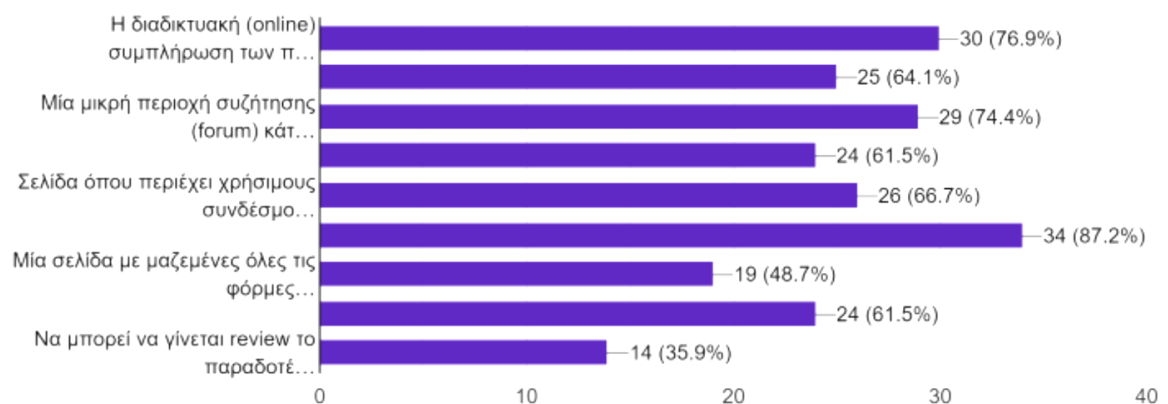
3) Θα θεωρούσες χρήσιμη την ύπαρξη μίας τέτοιας πλατφόρμας διαχείρισης;

39 responses



4) Ποιό/ά από τα παρακάτω θα θεωρούσες χρήσιμα να υπάρχουν:

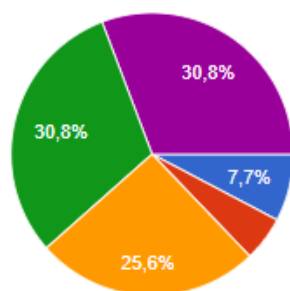
39 responses



Ποιό θεωρείς πιο χρήσιμο τρόπο ανατροφοδότησης από τον διδάσκοντα



39 απαντήσεις



- Μέσω email (σε κάθε μέλος της ομάδας)
- Μέσω email (στην ομάδα)
- Ακριβώς κάτω από το κάθε παραδοτέο στην πλατφόρμα
- Μέσω email (σε κάθε μέλος της ομάδας) ΚΑΙ ακριβώς κάτω από το κάθε παραδοτέο στην πλατφόρμα
- Μέσω email (στην ομάδα) ΚΑΙ ακριβώς κάτω από το κάθε παραδοτέο στην πλατφόρμα

Θα ήθελες να προτείνεις μήπως κάποια εισήγηση σχετικά με την δημιουργία της πλατφόρμας;

1 απάντηση

Να μπορεί ευκολα ο φοιτητης να βρει ολα οσα χρειαζεται και να δει παραδείγματα για το πως να συμπληρωσει τα παραδοτεα. Επισης ο καθηγητης να μπορεί ευκολα να τα διορθωσει και να δώσει σχολια για το καθε παραδοτεο

Παράρτημα

Β Ερωτηματολόγιο αξιολόγησης πλατφόρμας

B.1. Ερωτηματολόγιο

B.2. Απαντήσεις

B.1. Ερωτηματολόγιο

Αξιολόγηση Πλατφόρμας ΕΠΛ343

Στα πλαίσια της ατομικής διπλωματικής μου εργασίας δημιουργήσαμε μία διαδικτυακή πλατφόρμα με σκοπό τη διαχείριση του υποχρεωτικού μαθήματος ΕΠΛ343. Δίνεται στους φοιτητές αλλά και στον διδάσκοντα η δυνατότητα ευκολότερης ανάθεσης έργων λογισμικού για την ομαδική εργασία του μαθήματος, δημιουργία ομάδων ανάθεσης και πρόσβασης στα ερωτηματολόγια του μαθήματος. Κάθε φοιτητής θα έχει τον δικό του λογαριασμό έτσι ώστε να μπορεί να έχει πρόσβαση στις δυνατότητες τις σελίδας, ομοίως και οι εταιρίες που θα δίνουν έργα για υλοποίηση θα μπορούν να δημιουργούν λογαριασμό και στη συνέχεια να διαλέγουν ομάδα υλοποίησης για τα έργα τους.

Μετά την δοκιμή σας με την καινούρια πλατφόρμα και την εμπειρία που αποκτήσατε από το μάθημα ΕΠΛ343 θα θέλαμε την γνώμη σας σχετικά με την αξιολόγηση της ιστοσελίδας. Συγκεκριμένα υπάρχουν ερωτήσεις που αφορούν την ευχρηστία της ιστοσελίδας, κατά πόσο είναι χρήσιμη για το μάθημα και την δυσκολία που θα χρειαστεί ένας καινούριος χρήστης να εγκλιματιστεί με την ιστοσελίδα.

Τα δεδομένα θα χρησιμοποιηθούν ανώνυμα για τη διπλωματική μου εργασία και για σκοπούς αξιολόγησης της πλατφόρμας. Με τη συμπλήρωση του ερωτηματολογίου, συμφωνείτε με το παραπάνω.

*** Απαιτείται**

Φύλο: *

- ☐ Άντρας
- ☐ Γυναίκα
- ☐ Δεν απαντώ

Δοκιμή ως λογαριασμός: *

- ☐ Φοιτητή
- ☐ Καθηγητή
- ☐ Εταιρία/οργανισμός

Θεωρείς χρήσιμη την ύπαρξη μίας τέτοιας ιστοσελίδας; *

- ☐ Ναι
- ☐ Όχι
- ☐ Ίσως, αλλά χρειάζεται κάποιες βελτιώσεις

Πώς θα αξιολογούσατε την ιστοσελίδα μας σε σχέση με την κλίμακα αξιολόγησης *

	Ανεπαρκής	Αποδεκτή	Ικανοποιητική	Πολύ Καλή	Άριστη
Ευκολία στην πλοήγηση	☐	☐	☐	☐	☐
Δημιουργία / Προβολή έργων ανάθεσης	☐	☐	☐	☐	☐
Ποιότητα περιεχομένου	☐	☐	☐	☐	☐
Ακρίβεια στις πληροφορίες	☐	☐	☐	☐	☐
Αισθητική πλατφόρμας	☐	☐	☐	☐	☐
Χρησιμότητα περιεχομένου	☐	☐	☐	☐	☐
Ταχύτητα ιστοσελίδας (loading)	☐	☐	☐	☐	☐
Γλώσσα περιεχομένου	☐	☐	☐	☐	☐
Δομή περιεχομένων (πεδία-υποκατηγορίες)	☐	☐	☐	☐	☐

Πόσο εύκολη θεωρείτε την εκμάθηση της πλατφόρμας; *

- ☐ Πάρα πολύ
- ☐ Πολύ
- ☐ Μέτρια
- ☐ Λίγο
- ☐ Καθόλου

Πόσο χρήσιμη θεωρείτε την πλατφόρμα για την δημιουργία/ανάθεση έργων; *

- ☐ Πάρα πολύ
- ☐ Πολύ
- ☐ Μέτρια
- ☐ Λίγο
- ☐ Καθόλου

Πόσο κατανοητή σας φάνηκε η οργάνωση της ιστοσελίδας; *

- ☐ Πολύ Κατανοητή
- ☐ Κατανοητή
- ☐ Μέτρια
- ☐ Καθόλου Κατανοητή

Παρακαλώ συμπληρώστε την άποψη σας με τις πιο κάτω αναφορές: *

	Διαφωνώ απόλυτα	Διαφωνώ	Ούτε συμφωνώ, ούτε διαφωνώ	Συμφωνώ	Συμφωνώ απόλυτα
Θεωρώ ότι θα χρησιμοποιηούσα το σύστημα συχνά	☐	☐	☐	☐	☐
Βρήκα το σύστημα αχρείαστα περίπλοκο	☐	☐	☐	☐	☐
Θεωρώ ότι το σύστημα είναι εύκολο στην χρήση	☐	☐	☐	☐	☐
Θεωρώ ότι οι λειτουργίες του συστήματος είναι καλά ενοποιημένες	☐	☐	☐	☐	☐
Θεωρώ ότι το σύστημα έχει πολλές ασυνέπειες	☐	☐	☐	☐	☐
Θεωρώ πως θα είναι εύκολο για τον περισσότερο κόσμο να μάθει να χρησιμοποιεί το σύστημα	☐	☐	☐	☐	☐
Θεωρώ πως θα χρειάζονται την βοήθεια ατόμου με τεχνικές γνώσεις για να χρησιμοποιήσω το σύστημα	☐	☐	☐	☐	☐

Θεωρώ ότι το
σύστημα δεν
είναι καθόλου
ευέλικτο

☐☐☐☐☐

Θεωρώ ότι
χρειάζεται να
μάθω πολλές
πληροφορίες
πριν να
χρησιμοποιήσω
το σύστημα

☐☐☐☐☐

Νιώθω άνετα με
την χρήση του
συστήματος

☐☐☐☐☐

Πόσο ξεκάθαρες ήταν οι λειτουργίες της πλατφόρμας για τα δικαιώματα των
δεδομένων σας (π.χ. ανάκτηση, διαγραφή); *

1

2

3

4

5

Καθόλου ξεκάθαρες

☐☐☐☐☐

Πολύ ξεκάθαρες

Αξιολογείτε συνολικά την ιστοσελίδα μας. *

- ☐ Άριστα
- ☐ Πολύ Ικανοποιητική
- ☐ Ικανοποιητική
- ☐ Αποδεκτή
- ☐ Ανεπαρκής

Τι σας άρεσε περισσότερο στην ιστοσελίδα;

Η απάντησή σας

Τι σας άρεσε λιγότερο στην ιστοσελίδα;

Η απάντησή σας

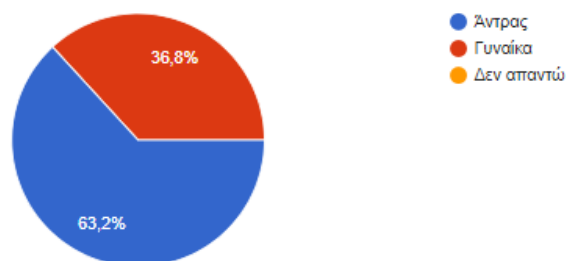
Τι επιπλέον λειτουργίες θα επιθυμούσατε να διαθέτει η πλατφόρμα;

Η απάντησή σας

B.2. Απαντήσεις

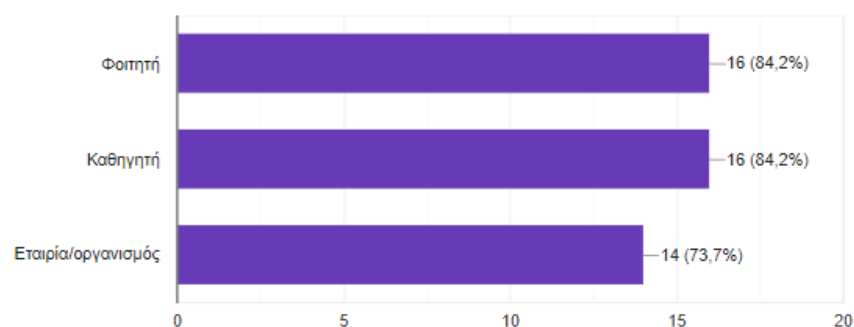
Φύλο:

19 απαντήσεις



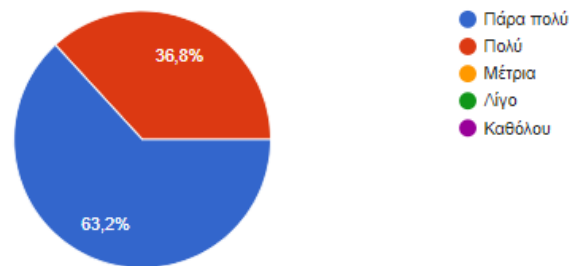
Δοκιμή ως λογαριασμός:

19 απαντήσεις



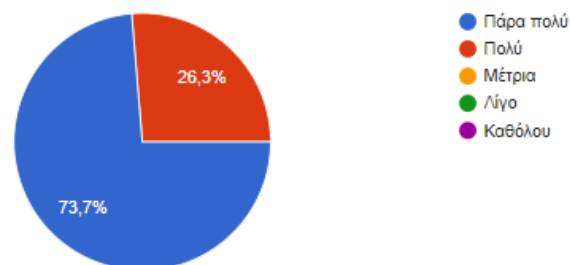
Πόσο εύκολη θεωρείτε την εκμάθηση της πλατφόρμας;

19 απαντήσεις



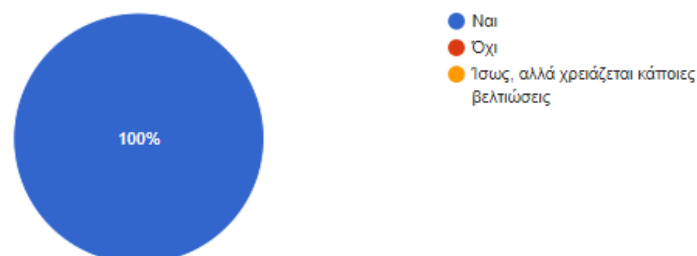
Πόσο χρήσιμη θεωρείτε την πλατφόρμα για την δημιουργία/ανάθεση έργων;

19 απαντήσεις

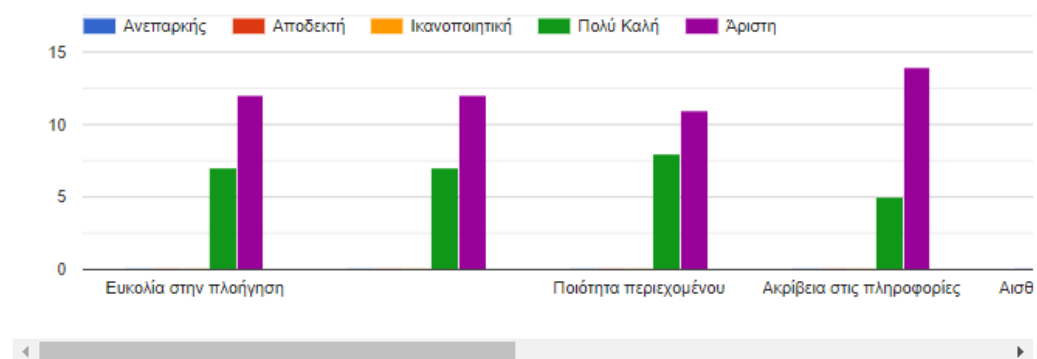


Θεωρείς χρήσιμη την ύπαρξη μίας τέτοιας ιστοσελίδας;

19 απαντήσεις

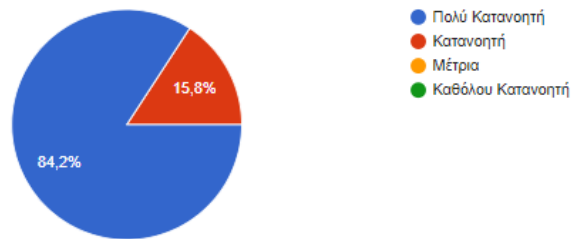


Πώς θα αξιολογούσατε την ιστοσελίδα μας σε σχέση με την κλίμακα αξιολόγησης



Πόσο κατανοητή σας φάνηκε η οργάνωση της ιστοσελίδας;

19 απαντήσεις

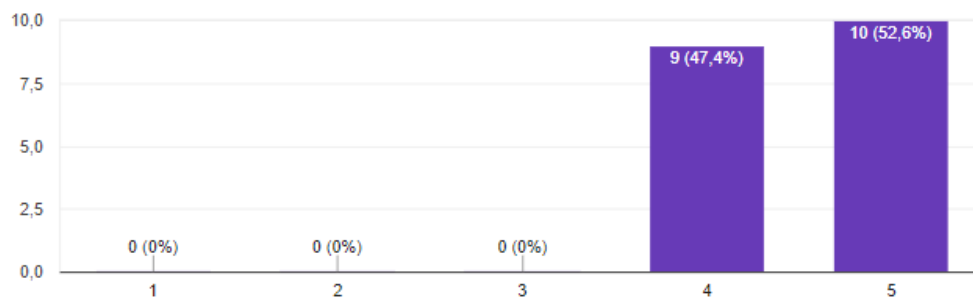


Παρακαλώ συμπληρώστε την άποψή σας με τις πιο κάτω αναφορές:



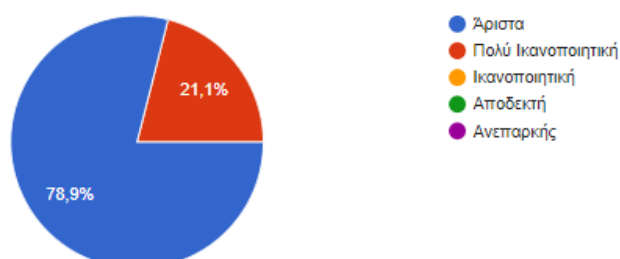
Πόσο ξεκάθαρες ήταν οι λειτουργίες της πλατφόρμας για τα δικαιώματα των δεδομένων σας (π.χ. ανάκτηση, διαγραφή);

19 απαντήσεις



Αξιολογείστε συνολικά την ιστοσελίδα μας.

19 απαντήσεις



Τι σας άρεσε περισσότερο στην ιστοσελίδα;

7 απαντήσεις

η δομη

Ευκολία στο να δώ όλα τα προσφερόμενα έργα

Η διεπαφή. Πολύ μοντέρνα και φιλική.

Το ότι δίνεται η δυνατότητα για επιλογή ομάδας στους φοιτητές και να δούμε και να επιλέξουμε project.

Η αισθητική της πλατφόρμας

Το μενού

Ευκολία στη χρήση, Ταχύτητα, Δυνατότητα δημιουργίας ομάδας μέσω αιτημάτων

Τι σας άρεσε λιγότερο στην ιστοσελίδα;

6 απαντήσεις

τιποτα

Στο κεφάλαιο του GDPR δυσκολεύτηκε να καταλάβω ότι ήταν υπερσυνδέσμοι.

Η ονομασία

Δεν μου έρχεται κάτι στο νου που δεν μου άρεσε.

Η ταχύτητα της ιστοσελίδας (μάλλον οφείλεται σε εξωτερικούς παράγοντες)

Το username και το email

Τι επιπλέον λειτουργίες θα επιθυμούσατε να διαθέτει η πλατφόρμα;

7 απαντήσεις

νομιζω εχει οτι χρειαζεται

Απευθείας συνομιλία της ομάδας με την εταιρεία με την οποία θα δουλέψουν

Η ιστοσελίδα είναι πλήρης!

Τίποτα περισσότερο.

Εισαγωγή βαθμολογίας εξέτασης κάθε μαθήματος και τελικού βαθμού μαθήματος (στο τέλος του έτους)

Να υπήρχαν φορουμας

Δυνατότητα ανταλλαγής μηνυμάτων

Παράρτημα

Γ Λειτουργίες για σύνδεση frontend με backend

Γ.1. Ερωτηματολόγιο

```
/**
 * A function exporting the function for logging in,
 * defined in user Requests.
 * @function login
 *
 * @param {string} username - The username of the user.
 * @param {string} password - The password of the user.
 * @param {boolean} createSession - True if we want a new session to be created.
 * @param {function} f - The callback function, which contains an error value (null if correct) and a boolean showing success.
 */
exports.login = function(username, password, createSession, f) {
  userReq.login(username, password, createSession, (response) => f(response));
}

/**
 * A function exporting the function for adding a new user
 * to the database defined in user Requests.
 * @function insertUser
 *
 * @param {string} username - The username of the new user.
 * @param {string} firstname - The first name of the new user.
 * @param {string} lastname - The last name of the new user.
 * @param {string} address - The address of the new user.
 * @param {string} emailAddress - The email address of the new user.
 * @param {string} password - The password of the new user.
 * @param {integer} phoneNumber - The phone number of the new user.
 * @param {string} companyName - The company name of the new user.
 * @param {function} f - The callback function, which contains an error value (null if correct) and a boolean which describes whether the function succeeded.
 */
exports.insertUser = function(username, firstName, lastName, address, emailAddress, password, phoneNumber, companyName, f) {
  userReq.insertUser(username, firstName, lastName, address, emailAddress, password, phoneNumber, companyName, (result) => f(result));
}

/**
 * A function exporting the function for activate a new company account.
```

```

* @function checkActivationCode
*
* @param {string} username - The username of the new user.
* @param {integer} activationCode - The activationCode of the user.
* @param {function} f - The callback function, which contains an error value
(null if correct) and a number which describes whether the function succeeded.
*/
exports.checkActivationCode = function(username, activationCode, f) {
  userReq.checkActivationCode(username, activationCode, (result) => f(result))
;
}

/**
* A function exporting the function for sending forget code for change password.
* @function sendForgetPasswordCode
*
* @param {string} username - The username of the user.
* @param {string} email - The email of the new user.
* @param {function} f - The callback function, which contains an error value
(null if correct) and a number which describes whether the function succeeded.
*/
exports.sendForgetPasswordCode = function(username, email, f) {
  userReq.sendForgetPasswordCode(username, email, (result) => f(result));
}

/**
* A function exporting the function for sending check the forgot code.
* and change the user password.
* @function checkForgetPasswordCode
*
* @param {string} password - The new password of user.
* @param {string} username - The username of the user.
* @param {integer} otk - The otk of the user.
* @param {function} f - The callback function, which contains an error value
(null if correct) and a number which describes whether the function succeeded.
*/
exports.checkForgetPasswordCode = function(username, otk, password, f) {
  userReq.checkForgetPasswordCode(username, otk, password, (result) => f(result));
}

/**
* A function exporting the data of the user.
* @function selectProfileData
*
* @param {string} username - The username of the user.
* @param {function} f - The callback function, which contains an error value
(null if correct) and a number which describes whether the function succeeded.
*/
exports.selectProfileData = function(username, f) {
  userReq.selectProfileData(username, (result) => f(result));
}

```

```

}

/**
 * A function exporting the function for retrieving the currently
 * logged in user from the database, defined in user Requests.
 * @function getProfileData
 *
 * @param {function} f - The callback function, which contains an error value
 (null if correct) and an object with the user's info.
 */
exports.getProfileData = function(username, f) {
  userReq.getProfileData(username, (user) => f(user));
}

/**
 * A function exporting the function for retrieving the currently
 * logged in user from the database, defined in user Requests.
 * @function getProfileDataID
 *
 * @param {function} f - The callback function, which contains an error value
 (null if correct) and an object with the user's info.
 */
exports.getProfileDataID = function(id, f) {
  userReq.getProfileDataID(id, (user) => f(user));
}

/**
 * A function exporting the function for retrieving the currently
 * interested teams for a current project from the database, defined in user R
equests.
 * @function getInterestProject
 *
 * @param {function} f - The callback function, which contains an error value
 (null if correct) and an object with the user's info.
 */
exports.getInterestProject = function(id, f) {
  userReq.getInterestProject(id, (user) => f(user));
}

/**
 * A function that update the profile data of the user.
 *
 * @function updateProfile
 *
 * @param {string} username - The username of the user.
 * @param {string} firstname - The new first name of the user.
 * @param {string} lastname - The new last name of the user.
 * @param {string} address - The new address of the user.
 * @param {string} emailAddress - The new email addresse of the user.
 * @param {integer} phoneNumber - The new phone numebr of the user.
 * @param {string} companyName - The new company name of the user.
 * @param {function} f - The callback function, which contains an error value
 (null if correct) and a boolean which describes whether the function succeeded
.

```

```

*/
exports.updateProfile = function(username, firstName, lastName, email, phone,
address, companyName, f) {
  userReq.updateProfile(username, firstName, lastName, email, phone, address, c
ompanyName, (result) => f(result));
}

/**
 * A function exporting the student details.
 * @function getStudents
 *
 * @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
 */
exports.getStudents = function( f) {
  userReq.getStudents((user) => f(user));
}

/**
 * A function exporting the student details.
 * @function getStudents2
 *
 * @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
 */
exports.getStudents2 = function( f) {
  userReq.getStudents2((user) => f(user));
}

/**
 * A function exporting the function for retrieving the currently
 * active or not projects.
 * @function getProjectsData
 *
 * @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
 */
exports.getProjectsData = function(f) {
  userReq.getProjectsData((user) => f(user));
}

/**
 * A function that create interest on Project with a specific student.
 * @function setInterestProject
 * @param {string} team - The team number of the user.
 * @param {string} idProject - The project Id of the project that the user is
interested.
 * @param {boolean} interest - The interest that we want to change in the data
base of the user.
 *
 * @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
 */
exports.setInterestProject = function(team, idProject, interest, f) {

```

```

    userReq.setInterestProject(team, idProject, interest, (user) => f(user));
}

/**
 * A function that create or update the Knowledge of a user with account student.
 * @function updateKnowledge
 * @param {string} web - An integer to show interest of a student to web applications.
 * @param {string} all - An array to show all program knowledge of students.
 * @param {string} app - An integer to show interest of a student to mobile applications.
 * @param {string} desktop - An integer to show interest of a student to desktop applications.
 * @param {integer} id - The id of the student user.
 *
 * @param {function} f - The callback function, which contains an error value (null if correct) and an object with the user's info.
 */
exports.updateKnowledge = function(all, web,app,desktop,id, f) {
    userReq.updateKnowledge(all, web,app,desktop,id, (user) => f(user));
}

/**
 * A function that create or update the name and description of a project.
 * @function updateProject
 * @param {string} name - The new name of the project that will be updated in the database.
 * @param {string} description - The new description of the project that will be updated in the database.
 * @param {string} radio - The new radio type of the project that will be updated in the database.
 * @param {integer} id - The id of the Project that will be updated in the database.
 * @param {integer} status - The status of the Project that will be updated in the database.
 *
 * @param {function} f - The callback function, which contains an error value (null if correct) and an object with the user's info.
 */
exports.updateProject = function(description, name, radio,id,status, f) {
    userReq.updateProject(description, name, radio,id,status, (user) => f(user))
;
}

/**
 * A function that update the interest of the teams in the database.
 * @function assignTeamProject
 * @param {string} team - The team of the project that will be updated about the project in the database.
 * @param {string} value - The new value type of the project that will be updated in the database.
 * @param {integer} id - The id of the Project that will be updated in the database.

```

```

*
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.assignTeamProject = function(id, team, value, f) {
  userReq.assignTeamProject(id, team, value, (user) => f(user));
}
/**
* A function exporting the function for retrieving teams details.
* @function getTeamMembers
*
* @param {string} team - The team number that will export details.
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.getTeamMembers = function(team , f) {
  userReq.getTeamMembers(team, (user) => f(user));
}
/**
* A function exporting the function for retrieving settings for a specific us
er.
* @function getSettings
*
* @param {string} id - The id of the user that data will be exported.
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.getSettings = function(id , f) {
  userReq.getSettings(id, (user) => f(user));
}
/**
* A function that update the notification and messages settings for users.
* @function saveSettings
* @param {integer} id - The id of the user that data will be updates.
* @param {integer} ne - The notifications for email for user settings.
* @param {integer} nn - The notifications for platform for user settings.
* @param {integer} me - The messages for email for user settings.
* @param {integer} mn - The messages for platform for user settings.
*
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.saveSettings = function(id, ne, nn,me,mn, f) {
  userReq.saveSettings(id, ne, nn,me,mn, (user) => f(user));
}
/**
* A function that update the update of a user in the database.
* @function updatePass
*
* @param {string} password - The new password of the user.
* @param {integer} id - The id of the user that will change account password.

```

```

* @param {integer} status - The status of the Project that will be updated in
the database.
*
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.updatePass = function(id, password, f) {
  userReq.updatePass(id, password, (user) => f(user));
}
/**
* A function exporting the function for retrieving team details.
* @function getTeams
*
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.getTeams = function(f) {
  userReq.getTeams( (user) => f(user));
}
/**
* A function exporting the function for retrieving team member details.
* @function getTeamMembers
*
* @param {integer} team - The team that will members export the invites.
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.getTeamMembers = function(team, f) {
  userReq.getTeamMembers( team, (user) => f(user));
}
/**
* A function exporting the function for retrieving team member details.
* @function deleteteam
*
* @param {integer} team - The team that will be delete and members will be re
lease .
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.deleteteam = function(team, f) {
  userReq.deleteteam( team, (user) => f(user));
}
/**
* A function that create a new team.
* @function CreateTeam
*
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.CreateTeam = function(f) {
  userReq.CreateTeam( (user) => f(user));
}

```

```

/**
 * A function that verify a team,
 * @function verifyTeam
 *
 * @param {integer} team - The team that will be verified.
 * @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
 */
exports.verifyTeam = function(team, f) {
  userReq.verifyTeam(team, (user) => f(user));
}
/**
 * A function that will delete a student.
 * @function deleteStudent
 *
 * @param {integer} username - The username of the student that will be delete
d.
 * @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
 */
exports.deleteStudent = function(username, f) {
  userReq.deleteStudent( username, (user) => f(user));
}
/**
 * A function that send request to other teams for joining.
 * @function sendRequestTeam
 *
 * @param {integer} id - The id of the student that want request.
 * @param {integer} action - The action will be request or cancel request.
 * @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
 */
exports.sendRequestTeam = function(id, action, f) {
  userReq.sendRequestTeam( id,action, (user) => f(user));
}
/**
 * A function exporting the function for retrieving specific team details.
 * @function getTeam
 *
 * @param {function} number - The team number that data will be exposed.
 * @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
 */
exports.getTeam = function(number, f) {
  userReq.getTeam(number, (user) => f(user));
}

/**
 * A function exporting the data for retrieving specific request.
 * @function getRequestsAll
 *
 * @param {integer} id - The id of the person that data will be exposed.

```

```

* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.getRequestsAll = function(id, f) {
  userReq.getRequestsAll(id, (user) => f(user));
}

/**
* A function that a user leave his/her current team.
* @function leaveTeam
*
* @param {integer} id - The id of the student that want request.
* @param {integer} team - The previous team number that want to leave in orde
r to check if the team still have members.
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.leaveTeam = function(id,team, f) {
  userReq.leaveTeam( id, team, (user) => f(user));
}

/**
* A function that is used by teacher account and rearrange the team members.
* @function addStudentToTeam
*
* @param {integer} username - The username of the student that will change te
am.
* @param {integer} team - The new team number of the student.
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.addStudentToTeam = function(username,team, f) {
  userReq.addStudentToTeam( username, team, (user) => f(user));
}

/**
* A function that find the students without a team yet.
* @function getStudentsWithoutTeam
*
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.getStudentsWithoutTeam = function(f) {
  userReq.getStudentsWithoutTeam((user) => f(user));
}

/**
* A function that send invite (status = 1 ) to user to join the team.
* @function handleSendInvite
*
* @param {integer} receiver - The username of the student that will be invite
d to join the team.
* @param {integer} team - The team number of the sender.
* @param {integer} id - The id number of the sender.

```

```

* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.handleSendInvite = function(team, receiver,id, f) {
  userReq.handleSendInvite(team, receiver,id, (user) => f(user));
}
/**
* A function that receive all the invites for joining to other teams with a s
pecific account id or team number.
* @function getInvites
*
* @param {integer} team - The team number of the team to check if has request
s.
* @param {integer} id - The id of the user to check if has requests.
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.getInvites = function(id, team, f) {
  userReq.getInvites(id, team, (user) => f(user));
}
/**
* A function that 'close' the request and make it to disappear from the scree
n.
* @function dontShowRequest
*
* @param {integer} id - The id of the request that we want to disappear.
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.dontShowRequest = function(id, f) {
  userReq.dontShowRequest(id, (user) => f(user));
}

/**
* A function that restricted data from a profile.
* @function handleRestrict
*
* @param {integer} id - The id of the request that we want to restrict their
data.
* @param {integer} type - The type of the restricted data.
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.handleRestrict = function(id,type, f) {
  userReq.handleRestrict(id,type, (user) => f(user));
}
/**
* A function that is called when a user reject a request .
* @function rejectRequest
*
* @param {integer} id - The id of the request that we want to reject by a use
r.

```

```

* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.rejectRequest = function(id, f) {
  userReq.rejectRequest(id, (user) => f(user));
}
/**
* A function that is called when a user accept a request .
* @function acceptRequest
*
* @param {integer} IDRequest - The id of the request - invite.
* @param {integer} IDUser - The id of the user that will accept the request.
* @param {integer} team - The team number that a user will join.
* @param {string} usernameSender - The username of the user that send the request.
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.acceptRequest = function(IDRequest, IDUser, team, usernameSender, f)
{
  userReq.acceptRequest(IDRequest, IDUser, team, usernameSender, (user) => f
(user));
}
/**
* A function that return all the deliverables.
* @function getDeliverables
*
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.getDeliverables = function(f) {
  userReq.getDeliverables((user) => f(user));
}
/**
* A function that return all the grades.
* @function getGrades
*
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.getGrades = function(f) {
  userReq.getGrades((user) => f(user));
}
/**
* A function that return all the grades.
* @function getGrades3
*
* @param {function} f - The callback function, which contains an error value
(null if correct) and an object with the user's info.
*/
exports.getGrades3 = function(f) {
  userReq.getGrades3((user) => f(user));
}

```

```

    }
    /**
     * A function that return the grades for a specific user.
     * @function getGrades
     *
     * @param {integer} id - The id of user that the grades will be exported.
     * @param {function} f - The callback function, which contains an error value (null if correct) and an object with the user's info.
     */
    exports.getGrades2 = function(id, f) {
        userReq.getGrades2(id, (user) => f(user));
    }
    /**
     * A function that delete a selected deliverable.
     * @function deleteDeliverable
     *
     * @param {integer} id - The id of deliverable that will be deleted.
     * @param {function} f - The callback function, which contains an error value (null if correct) and an object with the user's info.
     */
    exports.deleteDeliverable = function(id, f) {
        userReq.deleteDeliverable(id, (user) => f(user));
    }
    /**
     * A function that will change deliverable settings.
     * @function appearDel
     *
     * @param {integer} id - The id of deliverable that will be change appear change.
     * @param {integer} timi - The new value of apper setting.
     * @param {function} f - The callback function, which contains an error value (null if correct) and an object with the user's info.
     */
    exports.appearDel = function(timi,id, f) {
        userReq.appearDel(timi,id, (user) => f(user));
    }
    /**
     * A function that return all the questionnaires.
     * @function getQuestionarie
     *
     * @param {function} f - The callback function, which contains an error value (null if correct) and an object with the user's info.
     */
    exports.getQuestionarie = function(f) {
        userReq.getQuestionarie ((user) => f(user));
    }
    /**
     * A function that will change questionarie settings.
     * @function appearQues
     *
     * @param {integer} id - The id of questionarie that will be change appear change.

```

```

    * @param {integer} timi - The new value of apper setting.
    * @param {function} f - The callback function, which contains an error value (null if correct) and an object with the user's info.
    */
    exports.appearQues = function(timi,id, f) {
        userReq.appearQues(timi,id, (user) => f(user));
    }
    /**
     * A function that delete a selected questionnaire.
     * @function deleteQuestionarie
     *
     * @param {integer} id - The id of questionnaire that will be deleted.
     * @param {function} f - The callback function, which contains an error value (null if correct) and an object with the user's info.
     */
    exports.deleteQuestionarie = function(id, f) {
        userReq.deleteQuestionarie(id, (user) => f(user));
    }
    /**
     * A function that download all the data for a specific user.
     * @function downloadData
     *
     * @param {integer} id - The id of user that their data will be downloaded.
     * @param {integer} type - The account type of the user.
     * @param {function} f - The callback function, which contains an error value (null if correct) and an object with the user's info.
     */
    exports.downloadData = function(id,type, f) {
        userReq.downloadData(id,type, (user) => f(user));
    }
    /**
     * A function that will delete a user account.
     * @function deleteAccount
     *
     * @param {integer} id - The id of the user that will be deleted.
     * @param {function} f - The callback function, which contains an error value (null if correct) and an object with the user's info.
     */
    exports.deleteAccount = function(id, f) {
        userReq.deleteAccount( id, (user) => f(user));
    }
    /**
     * A function that will delete a user account.
     * @function deleteAccountStudent
     *
     * @param {integer} id - The id of the user that will be deleted.
     * @param {function} f - The callback function, which contains an error value (null if correct) and an object with the user's info.
     */
    exports.deleteAccountStudent = function(id, f) {
        userReq.deleteAccountStudent( id, (user) => f(user));
    }
}

```

```
/**
 * A function that will export data of companies.
 * @function getCompanies
 *
 * @param {function} f - The callback function, which contains an error value
 * (null if correct) and an object with the user's info.
 */
exports.getCompanies = function( f) {
  userReq.getCompanies( (user) => f(user));
}
```