



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ - WEBINTELLIGENCE

**ΙΕΕAlumni: Διαδικτυακή Υπηρεσία για τους Απόφοιτους
του Τμήματος ΜΠΗΣ**

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΤΑΟΥΚΤΣΗ ΒΑΣΙΛΕΙΟΥ

Επιβλέπων : Δημήτριος Δέρβος
Καθηγητής, ΔΙ.ΠΑ.Ε.

Θεσσαλονίκη, Φεβρουάριος 2023

Η σελίδα αυτή είναι σκόπιμα λευκή.



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ – WEB
INTELLIGENCE

ΙΕΕAlumni: Διαδικτυακή Υπηρεσία για τους Απόφοιτους του Τμήματος ΜΠΗΣ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΤΑΟΥΚΤΣΗ ΒΑΣΙΛΕΙΟΥ

Επιβλέπων : Δημήτριος Δέρβος
ΔΙ.ΠΑ.Ε.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στις .

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....
Όνομα Επώνυμο

ΔΙ.ΠΑ.Ε.

.....
Όνομα Επώνυμο

ΔΙ.ΠΑ.Ε.

.....
Όνομα Επώνυμο

ΔΙ.ΠΑ.Ε.

Θεσσαλονίκη,

(Υπογραφή)

.....

Click here to enter text.

Click here to enter text.

© – Allrightsreserved

Περίληψη

Αντικείμενα της διπλωματικής εργασίας συνιστούν (α) ο σχεδιασμός και η ανάπτυξη διαδικτυακής εφαρμογής (υπηρεσίας) για τους πτυχιούχους του προπτυχιακού και των μεταπτυχιακών προγραμμάτων σπουδών του τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων (ΜΠΗΣ) του Διεθνούς Πανεπιστημίου της Ελλάδος (ΔΙΠΑΕ), (β) η λειτουργική υποστήριξη της υπηρεσίας κατά το πρώτο στάδιο της λειτουργίας της, και (γ) ο σχεδιασμός νέας έκδοσης της υπηρεσίας με ενισχυμένη λειτουργικότητα και ευελιξία.

Η υπηρεσία ονομάζεται Alumni_IEE. Ο σχεδιασμός και η ανάπτυξή της έγιναν υπό την καθοδήγηση και την επίβλεψη της ΟΜάδας Εσωτερικής Αξιολόγησης (ΟΜΕΑ) του τμήματος ΜΠΗΣ. Στην πρώτη (v.1) έκδοσή της, η υπηρεσία υποστηρίζει την από μέρους του/ης απόφοιτου καταχώρηση δεδομένων που αφορούν σπουδές και επαγγελματικό έργο που έχουν εκπονηθεί είτε πριν, είτε μετά την αποφοίτησή του/ης από το Τμήμα. Φιλοξενείται σε εξυπηρετητή του τμήματος στη διεύθυνση <https://alumni.iee.ihu.gr/omea/index.php>. Χωρίς να δημοσιοποιεί προσωπικά δεδομένα (τηλέφωνο, διεύθυνση email, κλπ.), η υπηρεσία παρέχει τη δυνατότητα επικοινωνίας μεταξύ των αποφοίτων σε τρόπο ώστε να ενισχύεται η αίσθηση συναδελφικότητας και συνεργασίας μεταξύ τους. Ταυτόχρονα, τα δεδομένα που καταχωρούν οι απόφοιτοι συνιστούν πηγή χρήσιμης πληροφορίας για το Τμήμα που μέσω αυτών παρακολουθεί της εξέλιξη της απορρόφησης των αποφοίτων του από την αγορά εργασίας, την εξέλιξη των αποφοίτων που συνεχίζουν τις ακαδημαϊκές τους σπουδές, κλπ.

Η διαδικτυακή εφαρμογή αναπτύχθηκε με χρήση της γλώσσας προγραμματισμού PHP και του συστήματος διαχείρισης βάσεων δεδομένων MariaDB. Για την υλοποίηση των επιμέρους υποσυστημάτων της έγινε χρήση των JavaScript, jQuery, PHPMailer και της βιβλιοθήκης Datatables. Η επόμενη έκδοση (Alumni_IEE v.2) έκδοση της υπηρεσίας σχεδιάζεται με την αρχιτεκτονική Model-View-Controller, ώστε να χαρακτηρίζεται από αυξημένη ευελιξία μεταξύ των BackEnd και FrontEnd υποσυστημάτων της. Το αντίστοιχο, κατάλληλο REST API πρόκειται να αναπτυχθεί με τη χρήση του Laravel Framework.

Λέξεις Κλειδιά: MVC, REST API, PHP, jQuery, MariaDB, Laravel

Η σελίδα αυτή είναι σκόπιμα λευκή.

Abstract

The MSc thesis herewith reported had a three-fold objective: (a) to design and develop a web application (service) for the alumni of the Department of Information and Electronic Engineering (IEE) at the International Hellenic University (IHU). Graduates of the undergraduate and the graduate (MSc) programs of study alike, (b) to support the service, during the initial phase of its operation, and (c) to design a new version of the service, one with enhanced functionality and increased flexibility.

The web service developed is code named Alumni_IEE v.1. The IEE Internal Evaluation Group (IEG) has had a guiding and supervisory role throughout the design and development process. Alumni_IEE v.1 enables the graduates to register personal data pertaining to their studies, training, and professional activities undertaken either before or after their graduation from the IEE department. The service operates at <https://alumni.iee.ihu.gr/omea/index.php>. Without allowing personal data like telephone numbers and email addresses to go public within the Alumni_IEE v.1 user community, the service enables each registered graduate to get in touch with other fellow graduates, thus promoting togetherness, and co-operation among its user (alumni) community. At the same time, the data collected and organized by the service comprises a source of useful information. Alumni_IEE v.1 data undergo analytical processing by the IEE department for the purpose of monitoring the value each one program of study represents to the job market, the professional and/or academic progress its graduates make following the completion of their study, etc.

Alumni_IEE v.1 has been developed in PHP, utilizing the MariaDB RDBMS. Specific operational components were developed utilizing JavaScript, jQuery, PHPMailer, and the Datatables library. Alumni_IEE v.2 is to utilize the Model-View-Controller design architecture, for additional flexibility between its BackEnd and FrontEnd sub-systems. The corresponding REST API is to be developed in the Laravel Framework.

Keywords: MVC, REST API, PHP, jQuery, MariaDB, Laravel

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας περιεχομένων

ΙΕΕAlumni: Διαδικτυακή Υπηρεσία για τους Απόφοιτους του Τμήματος ΜΠΗΣ.....	i
1 Εισαγωγή.....	1
1.1 Εκθέσεις εσωτερικής αξιολόγησης/πιστοποίησης.....	1
1.2 Αντικείμενο διπλωματικής.....	2
1.3 Οργάνωση κειμένου.....	2
2 Θεωρητικό υπόβαθρο.....	4
2.1 Μοντέλο Οντοτήτων Συσχετίσεων.....	4
2.1.1 Σημειογραφία Crow's Foot Notation.....	5
2.2 Αρχιτεκτονική MVC.....	8
2.3 API (Application programming interface).....	9
2.4 REST (Representational State Transfer).....	9
3 Ανάλυση αναγκών.....	10
3.1 Γραμματεία.....	10
3.2 Μέλος ΟΜΕΑ.....	11
3.3 Απόφοιτος.....	11
3.4 Γενικές απαιτήσεις.....	12
4 ΙΕΕAlumni v.1.....	13
4.1 Σχεσιακό μοντέλο.....	13
4.2 Διάγραμμα ER.....	14
4.3 Ανάλυση σχέσεων μεταξύ οντοτήτων.....	14
4.3.1 Συσχέτιση users – graduate.....	14
4.3.2 Συσχέτιση users – other_studies.....	15
4.3.3 Συσχέτιση users – working_experience.....	15
4.3.4 Συσχέτιση users – uncommitted_users.....	16
4.3.5 Συσχέτιση users – personal_data_exposure.....	16
4.4 Πίνακες.....	17
4.4.1 Πίνακας users.....	17
4.4.2 Πίνακας graduate.....	17

4.4.3 Πίνακας <i>other_studies</i>	17
4.4.4 Πίνακας <i>working_experience</i>	17
4.4.5 Πίνακας <i>user_personal_data_exposure</i>	17
4.4.6 Πίνακας <i>uncommitted_users</i>	17
4.5 Λειτουργικότητα Alumni v.1.....	17
4.5.1 Σύνδεση στην Εφαρμογή.....	18
4.5.2 Αλλαγή Κωδικού.....	18
4.5.3 Καταχώρηση νέου Χρήστη από τη Γραμματεία.....	19
4.5.4 Ενημέρωση Στοιχείων Υφιστάμενου Χρήστη από τη Γραμματεία.....	20
4.5.5 Αποστολή Μηνύματος Ενεργοποίησης Λογαριασμού.....	22
4.5.6 Κατάλογος Όλων των Αποφοίτων.....	22
4.5.7 Ενημέρωση Ατομικών Στοιχείων από Απόφοιτο.....	23
4.5.8 Προσθήκη/Διαγραφή Άλλων Σπουδών από Απόφοιτο.....	23
4.5.9 Προσθήκη/Διαγραφή Επαγγελματικής Εμπειρίας από Απόφοιτο.....	24
4.5.10 Επιλογή Δημοσιοποίησης Προσωπικών Δεδομένων από Απόφοιτο.....	24
4.5.11 Αναζήτηση Άλλων Αποφοίτων.....	24
4.5.12 Αποστολή Μηνύματος από Απόφοιτο σε Απόφοιτο.....	24
4.5.13 Μαζικού Τύπου Προσθήκη Νέων Αποφοίτων.....	25
5 ΙΕΕAlumni v.2.....	26
5.1 Σχεσιακό μοντέλο.....	26
5.2 Διάγραμμα ER.....	28
5.3 Ανάλυση σχέσεων μεταξύ οντοτήτων.....	28
5.3.1 Συσχέτιση <i>users – roles – degrees</i>	29
5.3.2 Συσχέτιση <i>users – graduations</i>	30
5.3.3 Συσχετίσεις <i>users – other_studies/ working_history/countries</i>	31
5.3.4 Συσχέτιση <i>users – emails</i>	32
5.3.5 Συσχέτιση <i>users – notifications/password_resets</i>	33
5.4 Πίνακες.....	33
5.4.1 Πίνακας <i>users</i>	34
5.4.2 Πίνακας <i>roles</i>	34
5.4.3 Πίνακας <i>user_roles</i>	34
5.4.4 Πίνακας <i>graduations</i>	34

5.4.5 Πίνακας <i>degrees</i>	35
5.4.6 Πίνακας <i>other_studies</i>	35
5.4.7 Πίνακας <i>working_histories</i>	35
5.4.8 Πίνακας <i>education_origins</i>	35
5.4.9 Πίνακας <i>emails</i>	35
5.4.10 Πίνακας <i>notifications</i>	35
5.4.11 Πίνακας <i>countries</i>	35
5.4.12 Πίνακας <i>scopus_codes</i>	35
5.4.13 Πίνακας <i>password_resets</i>	36
5.4.14 Πίνακας <i>email_account_reset</i>	36
5.5 Alumni Api.....	36
5.5.1 Κλήσεις API.....	37
5.5.2 Παράδειγμα: Κλήση στον πίνακα <i>Degrees</i>	39
5.5.3 Παράδειγμα: Κλήση στον πίνακα <i>Graduations</i>	40
5.5.4 Κλήσεις με παραμέτρους στο URL.....	41
6 Τεχνολογίες που χρησιμοποιήθηκαν.....	43
6.1 Λογισμικό και υπολογιστική υποδομή.....	43
6.1.1 Εγκατάσταση Εφαρμογής.....	43
6.1.2 PHP.....	43
6.1.3 Composer.....	44
6.1.4 PHPMailer.....	46
6.1.5 HTML.....	47
6.1.6 CSS.....	48
6.1.7 Javascript.....	48
6.1.8 JQuery.....	48
6.1.9 Datatables.....	48
6.1.10 Bootstrap.....	51
6.1.11 PhpSpreadsheet.....	52
6.1.12 MariaDB.....	53
6.1.13 Laravel.....	53
7 Επίλογος.....	56

7.1 Σύνοψη και συμπεράσματα.....	56
8 Βιβλιογραφία.....	57
9 Παράρτημα.....	60
9.1 Οδηγίες χρήσης εφαρμογής Alumni v.1.....	60

Κατάλογος Εικόνων

Εικόνα 1: Crow's Foot Notation: Πολλαπλότητα ενός.....	6
Εικόνα 2: Crow's Foot Notation: Πολλαπλότητα πολλών.....	6
Εικόνα 3: Crow's Foot Notation: Προαιρετική σχέση.....	6
Εικόνα 4: Σχέση προς "Πολλά" με προαιρετική συμμετοχή.....	6
Εικόνα 5: Σχέση προς "Πολλά" με υποχρεωτική συμμετοχή.....	7
Εικόνα 6: Σχέση προς "Ένα" με υποχρεωτική συμμετοχή.....	7
Εικόνα 7: Σχέση προς "Ένα" με προαιρετική συμμετοχή.....	7
Εικόνα 8: MVC Model[6].....	8
Εικόνα 9: Παράδειγμα κλήσης σε MVC Model[7].....	8
Εικόνα 10: IEEAlumni v.1 ER.....	14
Εικόνα 11: IEEAlumni v.1 Συσχέτιση users – graduate.....	14
Εικόνα 12: IEEAlumni v.1 Συσχέτιση users – other_studies.....	15
Εικόνα 13: IEEAlumni v.1 Συσχέτιση users – working_experience.....	15
Εικόνα 14: IEEAlumni v.1 Συσχέτιση users – uncommitted_users.....	16
Εικόνα 15: IEEAlumni v.1 Συσχέτιση users – personal_data_exposure.....	16
Εικόνα 16: Σύνδεση στην Εφαρμογή.....	18
Figure 17: Αλλαγή Κωδικού.....	18
Εικόνα 18: Καρτέλα ατομικών στοιχείων.....	19
Εικόνα 19: Καρτέλα καταχώρισης στοιχείων αποφοίτησης.....	19
Εικόνα 20: Καταχώρηση χρήστη τύπου «Γραμματεία» ή «Μέλος OMEA».....	20
Εικόνα 21: Κατάλογος χρηστών με παρόμοιο επίθετο.....	20
Εικόνα 22: Ενημέρωση ατομικών στοιχείων.....	21
Εικόνα 23: Τροποποίηση στοιχείων αποφοίτησης.....	21
Εικόνα 24: Προσθήκη αποφοίτησης.....	22
Εικόνα 25: Κατάλογος χρηστών που εκκερεμεί η επιβεβαίωση.....	22
Εικόνα 26: Κατάλογος όλων των αποφοίτων.....	23
Εικόνα 27: Ενημέρωση ατομικών στοιχείων από απόφοιτο.....	23
Εικόνα 28: Προσθήκη/Διαγραφή άλλων σπουδών από απόφοιτο.....	23
Εικόνα 29: Προσθήκη/Διαγραφή επαγγελματικής εμπειρίας από απόφοιτο.....	24
Εικόνα 30: Επιλογή δημοσιοποίησης προσωπικών δεδομένων από απόφοιτο.....	24

Εικόνα 31: Αποστολή Μηνύματος από Απόφοιτο σε Απόφοιτο.....	25
Εικόνα 32: IEEAlumni v.2 - Διάγραμμα ER.....	28
Εικόνα 33: IEEAlumni v.2 Συσχέτιση users – roles – degrees (Λεπτομέρειες).....	29
Εικόνα 34: IEEAlumni v.2 Συσχέτιση users – graduations (Λεπτομέρειες).....	30
Εικόνα 35: IEEAlumni v.2 Συσχετίσεις users – other_studies/ working_history/countries (Λεπτομέρειες).....	31
Εικόνα 36: IEEAlumni v.2 Συσχέτιση users – emails (Λεπτομέρειες).....	32
Εικόνα 37: IEEAlumni v.2 Συσχέτιση users – notifications/password_resets (Λεπτομέρειες).....	33
Εικόνα 38: PHP Execution[13].....	44
Εικόνα 39: Σύσταση για δήλωση 2ης διεύθυνσης email.....	60
Εικόνα 40: Αλλαγή κωδικού και ενεργοποίηση του λογαριασμού.....	61
Εικόνα 41: Δήλωση προσωπικών στοιχείων.....	61
Εικόνα 42: Σπουδές και επαγγελματική σταδιοδρομία.....	62
Εικόνα 43: Επιλογή δεδομένων προς δημοσιοποίηση.....	62
Εικόνα 44: Αναζήτηση Αποφοίτων.....	62
Εικόνα 45: Απόφοιτοι/χρήστες της υπηρεσίας.....	63
Figure 46: Εικονίδια-φάκελοι για αποστολή email σε άλλον/η απόφοιτο.....	63

1

Εισαγωγή

1.1 Εκθέσεις εσωτερικής αξιολόγησης/πιστοποίησης

Το Διεθνές Πανεπιστήμιο της Ελλάδος προκειμένου να αποτελέσει πυλώνα αριστείας στην εκπαίδευση και την έρευνα, έχει αναπτύξει και εφαρμόζει ως μέρος της στρατηγικής του Πολιτική Διασφάλισης Ποιότητας. Το κεντρικό συντονιστικό όργανο όλων των διαδικασιών διασφάλισης ποιότητας και αξιολόγησης του Ιδρύματος είναι η Μονάδα Διασφάλισης Ποιότητας (ΜΟ.ΔΙ.Π.)[1]. Η ΜΟ.ΔΙ.Π είναι υπεύθυνη για τη διασφάλιση και συνεχή βελτίωση της ποιότητας του εκπαιδευτικού και ερευνητικού έργου, καθώς και για τη διασφάλιση της ποιότητας που σχετίζεται με την αποτελεσματική λειτουργία και απόδοση των υπηρεσιών του. Η ΜΟ.ΔΙ.Π ακολουθεί τις σχετικές διεθνείς πρακτικές, ιδίως εκείνες του Ευρωπαϊκού Χώρου Ανώτατης Εκπαίδευσης, και τις αρχές και κατευθύνσεις της ΕΘΑΑΕ.

Η Ομάδα Εσωτερικής Αξιολόγησης (ΟΜ.Ε.Α) κάθε τμήματος είναι το αρμόδιο όργανο που σε συνεργασία με την οικεία ΜΟΔΙΠ έχει την ευθύνη της διεξαγωγής της διαδικασίας εσωτερικής αξιολόγησης, παρακολουθεί τη συμπλήρωση των απογραφικών δελτίων και των ερωτηματολογίων, ενημερώνει τα όργανα και τα μέλη της ακαδημαϊκής μονάδας και συντάσσει την Έκθεση εσωτερικής αξιολόγησης/πιστοποίησης της ακαδημαϊκής μονάδας[2]. Η Έκθεση Εσωτερικής Αξιολόγησης συνεκτιμάται κατά τη λήψη αποφάσεων σε όλα τα επίπεδα λειτουργίας της Ακαδημαϊκής Μονάδας. Η έκθεση αυτή διαβιβάζεται στη Γενική Συνέλευση του τμήματος, τη ΜΟΔΙΠ και μέσω αυτής στην ΕΘΑΑΕ.

Το τμήμα «Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων» ΔΙ.ΠΑ.Ε προήλθε από την σύνεωση των τμημάτων «Μηχανικών Πληροφορικής» και «Ηλεκτρονικών

Μηχανικών», δύο τμήματα από όπου έχουν αποφοιτήσει χιλιάδες πτυχιούχοι. Υπάρχει ανάγκη για την καταχώριση, οργάνωση και αξιοποίηση δεδομένων που αφορούν στους αποφοίτους των προγραμμάτων σπουδών του (προπτυχιακό, μεταπτυχιακό, και διδακτορικό επίπεδο).

Η συγκεκριμένη διπλωματική εργασία εξυπηρετεί τις άμεσες και τις μακροπρόθεσμες αυτές ανάγκες συμβάλλοντας στη συγκέντρωση στοιχείων σχετικά με την επαγγελματική και ακαδημαϊκή εξέλιξη των αποφοίτων του τμήματος, βοηθώντας το έργο της Ομάδας Εσωτερικής Αξιολόγησης. Αναπτύσσεται υπό την επίβλεψη και καθοδήγηση της Ομάδας Εσωτερικής Αξιολόγησης (ΟΜΕΑ) του τμήματος και η αξιοποίηση των δεδομένων IEEAlumni θα συμβάλλει τα μέγιστα στη διαμόρφωση και εφαρμογή πολιτικών παρακολούθησης και βελτίωσης δεικτών ποιότητας που αφορούν στο εκπαιδευτικό έργο και στην εξειδίκευση που παρέχει το τμήμα και ο αντίκτυπος τους στην απορρόφηση των αποφοίτων του τμήματος από την αγορά εργασίας.

1.2 Αντικείμενο διπλωματικής

Η εργασία χωρίζεται σε δύο μέρη. Το πρώτο μέρος αφορά στο σχεδιασμό, ανάπτυξη και τεχνική υποστήριξη της εφαρμογής IEEAlumni v.1, με σκοπό να είναι λειτουργική ως προς τις βασικές απαιτήσεις και παραδοτέα πριν την έναρξη του χειμερινού εξαμήνου του ακαδημαϊκού έτους 2022-23. Η εφαρμογή θα επιτρέψει στη γραμματεία του τμήματος να καταχωρίσει όσους αποφοιτήσουν πριν το χειμερινό εξάμηνο του 2022-2023 και στους αποφοίτους αυτούς να καταχωρίσουν πληροφορίες χρήσιμες για την στατιστική ανάλυση που κάνει η ΟΜΕΑ.

Στο δεύτερο μέρος της εργασίας θα γίνει η προετοιμασία της έκδοσης IEEAlumni v.2 η οποία προβλέπεται να λειτουργήσει το ακαδημαϊκό έτος 2024-25. Στην έκδοση IEEAlumni v.2 γίνεται εκ νέου σχεδιασμός με την λογική της ανεξαρτητοποίησης του BackEnd από το FrontEnd. Υλοποιείται API για τις κλήσεις στο BackEnd, ακολουθώντας τις κατευθυντήριες γραμμές της αρχιτεκτονικής προγραμματισμού Model–View–Controller (MVC). Η συμπλήρωση της νέας έκδοσης με FrontEnd θα γίνει σε επόμενη διπλωματική.

1.3 Οργάνωση κειμένου

Η εργασία μας αποτελείται από 9 κεφάλαια. Στο 1ο κεφάλαιο αναφερόμαστε στο αντικείμενο της εργασίας μας, στο 2ο κεφάλαιο προσεγγίζουμε το θεωρητικό υπόβαθρο στο οποίο βασίζεται η υλοποίηση του API, στο 3ο κεφάλαιο παρουσιάζουμε την ανάλυση των αναγκών που καλύπτει η εφαρμογή, στο 4ο και 5ο κεφάλαιο περιγράφουμε το σχεσιακό μοντέλο τις σχέσεις μεταξύ των οντοτήτων κάθε έκδοσης της εφαρμογής και το API που υλοποιήθηκε,

στο 6ο κεφάλαιο αναφερόμαστε στο λογισμικό και την υπολογιστική υποδομή που βασιστήκαμε, το 7ο κεφάλαιο αφορά στον επίλογο, το 8ο τη βιβλιογραφία και στο 9ο κεφάλαιο παραθέτουμε τις οδηγίες χρήσης της εφαρμογής.

2

Θεωρητικό υπόβαθρο

2.1 Μοντέλο Οντοτήτων Συσχετίσεων

Σε μια σχεσιακή βάση δεδομένων η αποθήκευση των δεδομένων γίνεται με τη μορφή πινάκων. Κάποια από τα δεδομένα των πινάκων υποδεικνύουν δεδομένα σε άλλους πίνακες και με αυτόν τον τρόπο οι πίνακες συσχετίζονται. Το Μοντέλο Οντοτήτων Συσχετίσεων (Entity Relationship Model - ERM) είναι μια αφηρημένη και εννοιολογική αναπαράσταση αυτών των δεδομένων και αποδίδεται με το διάγραμμα οντοτήτων - συσχετίσεων[3]. Τα στοιχεία που περιλαμβάνει ένα διάγραμμα οντοτήτων - συσχετίσεων είναι οι οντότητες, οι ιδιότητες των οντοτήτων και οι συσχετίσεις μεταξύ των οντοτήτων.

Η οντότητα (entity) αφορά την αναπαράσταση μιας κλάσης ή ενός αντικειμένου και στιγμιότυπο (instance) μιας οντότητας είναι μια συγκεκριμένη περίπτωση ενός τύπου οντότητας.

Η σύνδεση μεταξύ δύο ή περισσότερων τύπων οντοτήτων ονομάζεται συσχέτιση (relationship) και καθορίζει με ποιο τρόπο σχετίζονται οι οντότητες μιας βάσης δεδομένων. Οι συσχετίσεις συνήθως αποδίδονται λεκτικά ως ρήματα που συνδέουν δύο ή περισσότερα ουσιαστικά.

Μια συσχέτιση χαρακτηρίζεται από την πολλαπλότητα (Cardinality) και την υποχρεωτικότητα συμμετοχής. Η πολλαπλότητα καθορίζει το πλήθος των στιγμιότυπων αυτού του τύπου συσχέτισης, στον οποίο μια οντότητα μπορεί να συμμετάσχει και μπορεί να είναι :

- 1:N (ένα προς πολλά)
- 1:1 (ένα προς ένα) ή
- M:N (πολλά προς πολλά)

Η υποχρεωτικότητα συμμετοχής αφορά το αν η ύπαρξη στιγμιοτύπων μιας οντότητας εξαρτάται από τη σύνδεση της με στιγμιότυπα της άλλης οντότητας. Δηλαδή αν τα στιγμιότυπα μιας οντότητα μπορούν να υφίσταται χωρίς την ύπαρξη στιγμιοτύπων της άλλης οντότητας με τα οποία θα συσχετίζονται τότε έχουμε προαιρετική ή μερική συμμετοχή. Αλλιώς η υποχρεωτικότητα συμμετοχή χαρακτηρίζεται ως μερική/προαιρετική.

2.1.1 Σημειογραφία Crow's Foot Notation

Η σήμανση Crow's Foot Notation παρουσιάστηκε με την ονομασία inverted arrow το 1976 από τον Gordon C. Everest[4]. Τα βασικά στοιχεία της σήμανσης είναι[5]:

2.1.1.1 Η οντότητα (entity)

Πρόκειται για την αναπαράσταση μιας κλάσης ή ενός αντικειμένου. Μια οντότητα έχει ιδιότητες (attributes) που τη χαρακτηρίζουν. Οι οντότητες αναπαρίστανται με ορθογώνιο παραλληλόγραμμο με τον όνομα της οντότητας στην κορυφή.

2.1.1.2 Η ιδιότητα μιας οντότητας (attribute)

Μια ιδιότητα χαρακτηρίζει μια συγκεκριμένη οντότητα. Οι ιδιότητες που καθορίζουν τη μοναδικότητα ενός στιγμιοτύπου (πχ κύριο κλειδί) αποτυπώνονται με αστερίσκο (*)

2.1.1.3 Οι σχέσεις (Relationships) μεταξύ οντοτήτων

Οι σχέσεις απεικονίζουν τη συσχέτιση μεταξύ δύο οντοτήτων. Αναπαρίστανται με ευθείες γραμμές. Κάθε σχέση μπορεί να έχει όνομα – χρησιμοποιούμε ρήμα - που εκφράζει το είδος της σχέσης που συνδέει τα αντικείμενα και τοποθετείται πάνω στη γραμμή που απεικονίζει τη σχέση.

2.1.1.3.1 Πολλαπλότητα (Cardinality)

Οι σχέσεις έχουν δύο δείκτες, ένας σε κάθε άκρο της γραμμής:

- Ο πρώτος λέγεται πολλαπλότητα (multiplicity) και εκφράζει τον μέγιστο αριθμό συσχετίσεων που το στιγμιότυπο μιας οντότητας μπορεί έχει με στιγμιότυπα της άλλης οντότητας και παίρνει τιμές:
 - ένα → ένα: Όταν το στιγμιότυπο της πρώτης οντότητας συσχετίζεται με ένα και μόνο ένα στιγμιότυπο της δεύτερης οντότητας

- ένα → πολλά: Όταν το στιγμιότυπο της πρώτης οντότητας μπορεί να συσχετιστεί με περισσότερα από ένα στιγμιότυπα της δεύτερης οντότητας
- Ο δεύτερος δείκτης εκφράζει τον ελάχιστο αριθμό συσχετίσεων που το στιγμιότυπο μιας οντότητας μπορεί έχει με στιγμιότυπα της άλλης οντότητας, εκφράζει δηλαδή αν η σχέση είναι προαιρετική ή υποχρεωτική. Παίρνει τιμές μηδέν ή ένα.

Οι δείκτες αυτοί τοποθετούνται με συγκεκριμένη σειρά. Στην εξωτερική άκρη της σχέσης τοποθετείται πρώτο το σύμβολο της πολλαπλότητας και το σύμβολο που υποδεικνύει εάν η σχέση είναι υποχρεωτική ή προαιρετική εμφανίζεται μετά από αυτό.

- Μια πολλαπλότητα ενός και μια υποχρεωτική σχέση αντιπροσωπεύεται από μια ευθεία γραμμή κάθετη στη γραμμή σχέσης



Εικόνα 1: Crow's Foot Notation: Πολλαπλότητα ενός

- Ένα πλήθος πολλών αντιπροσωπεύεται από το τρίπτυχο σύμβολο που μοιάζει με πόδι κορακιού



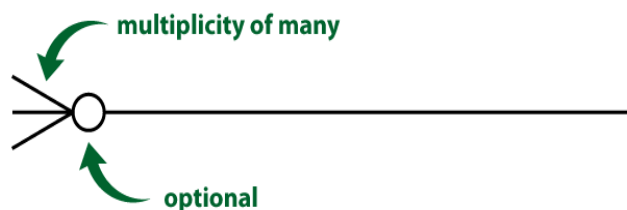
Εικόνα 2: Crow's Foot Notation: Πολλαπλότητα πολλών

- Μια προαιρετική σχέση αντιπροσωπεύεται από έναν κενό κύκλο

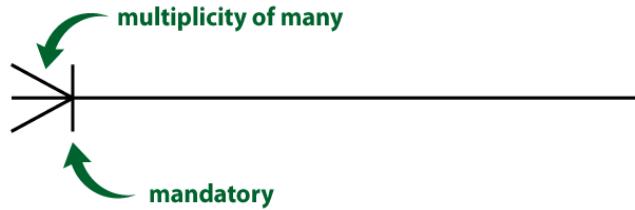


Εικόνα 3: Crow's Foot Notation: Προαιρετική σχέση

Με βάση τα παραπάνω παραθέτουμε συνδυασμούς πολλαπλότητας και προαιρετικής/υποχρεωτικής συμμετοχής (για το ένα άκρο της σχέσης)[5]:



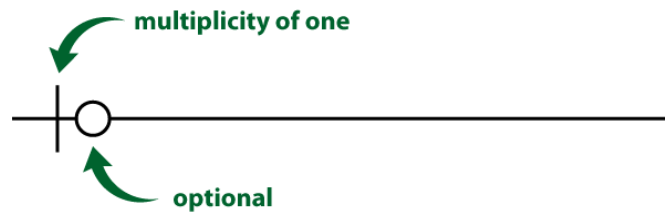
Εικόνα 4: Σχέση προς "Πολλά" με προαιρετική συμμετοχή



Εικόνα 5: Σχέση προς "Πολλά" με υποχρεωτική συμμετοχή



Εικόνα 6: Σχέση προς "Ένα" με υποχρεωτική συμμετοχή



Εικόνα 7: Σχέση προς "Ένα" με προαιρετική συμμετοχή

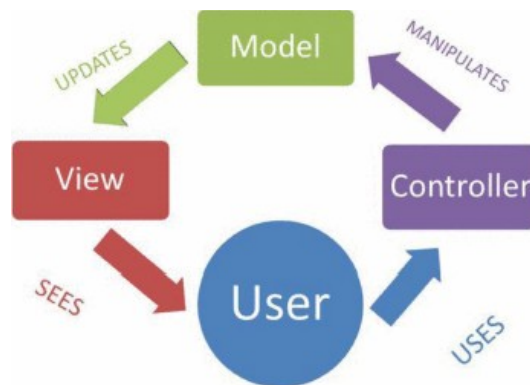
2.1.1.3.2 MySQL Workbench

Για την παραγωγή των διαγραμμάτων ER με τη σημειογραφία Crow's Foot θα χρησιμοποιηθεί το λογισμικό MySQL Workbench. Σε αυτό το λογισμικό διακρίνουμε δύο περιπτώσεις για τη σήμανση μια σχέσης ξένου κλειδιού:

- Προσδιορίσιμη συσχέτιση (identifying relationship): Πρόκειται για μια συσχέτιση όπου το στιγμότυπο της μιας οντότητας δεν μπορεί να προσδιοριστεί επαρκώς χωρίς το στιγμότυπο της άλλης οντότητας, δηλαδή όταν μέρος του κυρίου κλειδιού της είναι ξένο κλειδί στην άλλη οντότητα. Η προσδιορίσιμη συσχέτιση απεικονίζεται με συμπαγή γραμμή.
- Μη προσδιορίσιμη συσχέτιση (non-identifying relationship): Όταν το κύριο κλειδί της οντότητας δεν είναι ξένο κλειδί στην άλλη. Συμβολίζεται με διακεκομμένη γραμμή

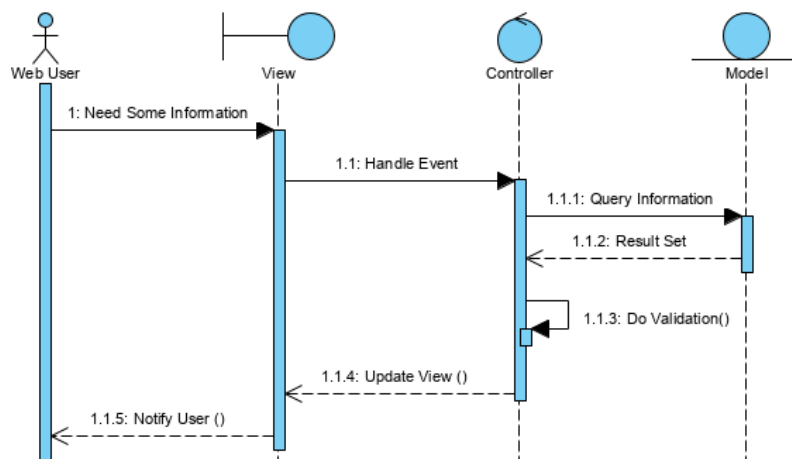
2.2 Αρχιτεκτονική MVC

Το μοντέλο Model-View-Controller (MVC) είναι ένα σχεδιαστικό πρότυπο στο οποίο η εφαρμογή σε διαχωρίζεται σε τρεις περιοχές, το μοντέλο (Model), την εμφάνιση (View) και την αλληλεπίδραση των μερών (Controller). Στην ουσία απομονώνει το χειρισμό των δεδομένων (Model) από την εμφάνιση (View) και τα η επικοινωνία αυτών των δύο μερών ανατίθεται στον ελεγκτή (Controller).



Εικόνα 8: MVC Model[6]

- Το Model διαχειρίζεται τα δεδομένα της εφαρμογής, περιγράφει τη λογική της και τις διαδικασίες της εφαρμογής. Δέχεται τα δεδομένα εισόδου μέσω του Controller.
- Το View διαχειρίζεται την εμφάνιση της εφαρμογής και την παρουσίαση των δεδομένων στον χρήστη.
- Ο Controller είναι ο σύνδεσμος μεταξύ του χρήστη και του συστήματος. Είναι ο χειριστής του Model και του View. Ελέγχει τη ροή των δεδομένων και τα παρέχει στο View προκειμένου να εμφανιστούν στον χρήστη.



Εικόνα 9: Παράδειγμα κλήσης σε MVC Model[7]

Το πλεονέκτημα αυτής της αρχιτεκτονικής είναι η ανεξαρτησία του BackEnd από το FrontEnd που καθιστά την εφαρμογή επεκτάσιμη και ευκολότερα συντηρήσιμη.

2.3 API (Application programming interface)

Ως API (Application Programming Interface) ή αλλιώς Διεπαφή Προγραμματισμού Εφαρμογών ορίζεται ένα σύνολο από μεθόδους και πρωτόκολλα που επιτρέπουν την επικοινωνία και ανταλλαγή δεδομένων μεταξύ διαφορετικών λογισμικών. Έτσι με τη χρήση του API, ξένα λογισμικά έχουν πρόσβαση στη λειτουργία της εφαρμογής μας, χωρίς να γνωρίζουν τον κώδικα ο οποίος εκτελείται. Ένα API μπορεί να σχεδιαστεί για web-based εφαρμογές, για ένα λειτουργικό σύστημα, για μια βάση δεδομένων κλπ.

Πολλές υπηρεσίες προσφέρουν δημόσια Web API που επιτρέπουν σε οποιονδήποτε να στέλνει και να λαμβάνει περιεχόμενο από την υπηρεσία χρησιμοποιώντας το πρωτόκολλο HTTP. Ένα πολύ γνωστό Web API στον Ελληνικό χώρο είναι το API της Διαύγειας[8].

2.4 REST (Representational State Transfer)

Το REST (Representational State Transfer) ή RESTful είναι μια αρχιτεκτονική για τη δημιουργία API που χρησιμοποιούν το πρωτόκολλο HTTP[9]. Με την RESTful API προσέγγιση μια συναλλαγή (transaction) προσεγγίζεται ως ένα σύνολο στοιχειωδών ενεργειών που εκτελούνται διαδοχικά προσδίδοντας ευελιξία στην ανάπτυξη της εφαρμογής.

Με ένα REST API τα στοιχεία δικτύου αντιμετωπίζονται σαν πόροι, στους οποίους ο χρήστης ζητά πρόσβαση μέσω των μεθόδων GET, PUT, POST, DELETE του HTTP πρωτοκόλλου, υλοποιώντας με αυτές τις λειτουργίες CRUD (Create, Read, Update, Delete).

- GET: Η μέθοδος αυτή χρησιμοποιείται για την ανάγνωση της τιμής ενός πόρου
- PUT: Η μέθοδος αυτή χρησιμοποιείται για την ενημέρωση υφιστάμενου πόρου
- POST: Η μέθοδος αυτή χρησιμοποιείται για την καταχώριση ενός νέου πόρου
- DELETE: Η μέθοδος αυτή χρησιμοποιείται για τη διαγραφή ενός πόρου

Κάθε REST αίτημα περιλαμβάνει τέσσερα βασικά μέρη:

- Τη μέθοδο HTTP
- Το endpoint, δηλαδή ένα Uniform Resource Identifier (URI) που υποδεικνύει τη διαδρομή προς τον πόρο.
- Τα headers του αιτήματος που περιλαμβάνουν πληροφορίες σχετικές με τον πελάτη του αιτήματος και τον εξυπηρετητή που το διαχειρίζεται, όπως πχ δεδομένα αυθεντικοποίησης, οδηγίες μορφοποίησης κλπ
- Το body με το οποίο αποστέλλονται επιπλέον πληροφορίες στον εξυπηρετητή, όπως πχ τα δεδομένα που είναι απαραίτητα για τη δημιουργία ενός νέου πόρου ή την ενημέρωση ενός υφιστάμενου.

3

Ανάλυση αναγκών

Η Ομάδα Εσωτερικής Αξιολόγησης (ΟΜ.Ε.Α) στο πλαίσιο της σύνταξης της ετήσιας έκθεσης εσωτερικής αξιολόγησης/πιστοποίησης του Τμήματος (ακαδημαϊκή μονάδα) χρειάζεται στατιστικά στοιχεία σε σχέση με την απορρόφηση των αποφοίτων στην αγορά εργασίας. Σήμερα αυτό δεν είναι δυνατόν παρά μόνο με άμεση (ένας-προς-έναν τύπου) επικοινωνία με τους αποφοίτους.

Γι' αυτόν τον σκοπό αποφασίστηκε να δημιουργηθεί η εφαρμογή IEEAlumni στην οποία οι απόφοιτοι θα καταχωρούν τις σχετικές πληροφορίες και από την οποία η ΟΜΕΑ θα εξάγει τα στατιστικά της στοιχεία.

Επειδή η ανάγκη για μια λειτουργική ως προς τις βασικές απαιτήσεις εφαρμογή, ήταν άμεση, αποφασίστηκε να υλοποιηθεί άμεσα η έκδοση IEEAlumni v.1 και να παραδοθεί προς χρήση με την έναρξη του χειμερινού εξαμήνου του ακαδημαϊκού έτους 2022-23. Κατά την ανάλυση αναγκών της έκδοσης IEEAlumni v.1 συμφωνήθηκε να υπάρχουν τρεις κατηγορίες χρηστών με διαφορετικούς ρόλους. Πρόκειται για τους χρήστες «Γραμματεία», «Μέλος ΟΜΕΑ» και «Απόφοιτος»

3.1 Γραμματεία

Ο χρήστης «Γραμματεία» είναι διοικητικός υπάλληλος του του τμήματος ΜΠΗΣ και έχει την ευθύνη της καταχώρισης των αποφοίτων του τμήματος, των βασικών τους στοιχείων και την περιγραφή των σπουδών τους στο τμήμα. Πρέπει επίσης να επεξεργάζεται τις υφιστάμενες καταχωρίσεις και να τροποποιεί όποιο πεδίο χρήζει ενημέρωσης.

Πλέον των αποφοίτων έχει την αρμοδιότητα της προσθήκης νέων χρηστών των άλλων δύο κατηγοριών, δηλαδή χρηστών τύπου «Γραμματεία» και «Ομάδα ΟΜΕΑ» καθώς και την ενημέρωση των σχετικών καρτελών.

Βασική αρμοδιότητα της γραμματείας είναι η δημιουργία νέων χρηστών της υπηρεσίας IEEAlumni και η αποστολή πρόσκλησης/υπενθύμισης για την από μέρους τους ενεργοποίηση των λογαριασμών τους. Αυτό θα πρέπει να γίνεται με αυτοματοποιημένη αποστολή προσωποποιημένων μηνυμάτων αλληλογραφίας.

Μεταξύ των άλλων η γραμματεία του Τμήματος θα πρέπει να έχει πρόσβαση στον πλήρη κατάλογο των καταχωρημένων αποφοίτων.

3.2 Μέλος ΟΜΕΑ

Η βασική ανάγκη του χρήστη «Μέλος ΟΜΕΑ» είναι η πρόσβαση στα δεδομένα προκειμένου να εξάγει στατιστικά στοιχεία. Γι' αυτό και έχει άμεση πρόσβαση στη Βάση Δεδομένων από όπου μπορεί να αντλήσει τις απαιτούμενες για το έργο του πληροφορίες.

Όμως κατά το σχεδιασμό της εφαρμογής προέκυψε μια επιπλέον απαίτηση η οποία ενώ δεν άπτεται του πυρήνα των αναγκών του, είναι ο μόνος ικανός χρήστης για την εκπλήρωση της. Προκειμένου να απαλλαγθεί η γραμματεία από την χειροκίνητη εισαγωγή των νέων αποφοίτων, θα ήταν ιδανικό να συνδεθεί εφαρμογή με το σύστημα UniPortal. Όμως το σύστημα UniPortal δεν υποστηρίζει, στην παρούσα στιγμή, το σχετικό API που θα επέτρεπε αυτήν την λειτουργικότητα. Γι' αυτό και θεωρήθηκε ότι το «Μέλος ΟΜΕΑ» μπορεί αναλάβει αυτό το έργο ακολουθώντας μια ημιαυτόματη διαδικασία.

3.3 Απόφοιτος

Ο σημερινός απόφοιτος του Τμήματος δεν έχει άμεση επαφή με το Τμήμα και με τους άλλους αποφοίτους. Η εφαρμογή IEEAlumni έρχεται να καλύψει αυτό το κενό. Όπως σε οποιαδήποτε εφαρμογή, ο χρήστης θα πρέπει να μπορεί να ενημερώνει τα προσωπικά του στοιχεία. Αποφασίστηκε να έχει τη δυνατότητα ενημέρωσης μέρους των ατομικών του στοιχείων, ενώ για τα πεδία Επίθετο, Όνομα, Πατρώνυμο, Ημ/νια Γέννησης, ΑΦΜ και ΑΜΚΑ θα πρέπει να απευθύνεται στη γραμματεία για να κάνει εκείνη τη σχετική αλλαγή.

Βασικός στόχος της εφαρμογής είναι η συλλογή δεδομένων από τους αποφοίτους. Γι' αυτό και ο απόφοιτος πρέπει να έχει τη δυνατότητα να καταχωρεί το σύνολο του επαγγελματικού του ιστορικού, όπως και των σπουδών του, επιπλέον εκείνων που έχει εκπονήσει στο τμήμα ΜΠΗΣ.

Προκειμένου να είναι ελκυστική η χρήση της εφαρμογής ώστε ο χρήστης να έχει κίνητρο για την καταχώριση αυτών των πληροφοριών, θεωρήθηκε πως πρέπει να του δοθούν κάποια

κίνητρα. Στο πλαίσιο της παρούσας εργασίας, θεωρήθηκε ως κίνητρο για τον χρήστη η παροχή σε αυτόν της δυνατότητας να αναζητεί άλλους αποφοίτους και να επικοινωνεί με αυτούς μέσω email, ακόμη και αν δεν γνωρίζει τις αντίστοιχες διευθύνσεις ηλεκτρονικού ταχυδρομείου. Επειδή στην αναζήτηση αυτή δυνητικά μπορούν να εμφανιστεί μέρος των προσωπικών του δεδομένων (στοιχεία επικοινωνίας – άλλες σπουδές – επαγγελματική εμπειρία), αποφασίστηκε να δοθεί στον χρήστη η δυνατότητα να καθορίζει μόνος του το εύρος των προσωπικών του δεδομένων που θα είναι ορατά από τρίτους.

3.4 Γενικές απαιτήσεις

Υπάρχει ανάγκη για μηχανισμούς αυθεντικοποίησης, αλλαγής κωδικών και αλλαγής λογαριασμού αλληλογραφίας. Αποφασίστηκε η σύνδεση στην εφαρμογή να γίνεται με συνδυασμό ενός από τους καταχωρημένους λογαριασμούς αλληλογραφίας - προβλέπεται η δυνατότητα καταχώρισης πέραν του ενός λογαριασμού αλληλογραφίας - και τον προσωπικό κωδικό, η δε αλλαγή του προσωπικού κωδικού και του λογαριασμού αλληλογραφίας να γίνεται με την αποστολή προσωρινού κωδικού επιβεβαίωσης.

4

IEEAlumni v.1

Επειδή η ανάγκη για μια λειτουργική ως προς τις βασικές απαιτήσεις εφαρμογή, ήταν άμεση, αποφασίστηκε να υλοποιηθεί άμεσα η έκδοση IEEAlumni v.1 και να παραδοθεί προς χρήση με την έναρξη του χειμερινού εξαμήνου του ακαδημαϊκού έτους 2022-23.

4.1 Σχεσιακό μοντέλο

Το μοντέλο της υπηρεσίας αποτελείται από έξι βασικές οντότητες που συσχετίζονται μεταξύ τους. Πρόκειται για τις:

- *users*: Η οντότητα που αφορά στα προσωπικά στοιχεία ενός χρήστη
- *graduate*: Η οντότητα που αφορά στις αποφοιτήσεις
- *other_studies*: Η οντότητα που αφορά στις σπουδές σε άλλα τμήματα
- *working_experience*: Η οντότητα που αφορά στην επαγγελματική εμπειρία
- *user_personal_data_exposure*: Η οντότητα που αφορά το βαθμό έκθεσης προσωπικών δεδομένων ενός χρήστη
- *uncommitted_users*: Η οντότητα που αφορά στους χρήστες που δεν έχουν ενεργοποιήσει το λογαριασμό τους

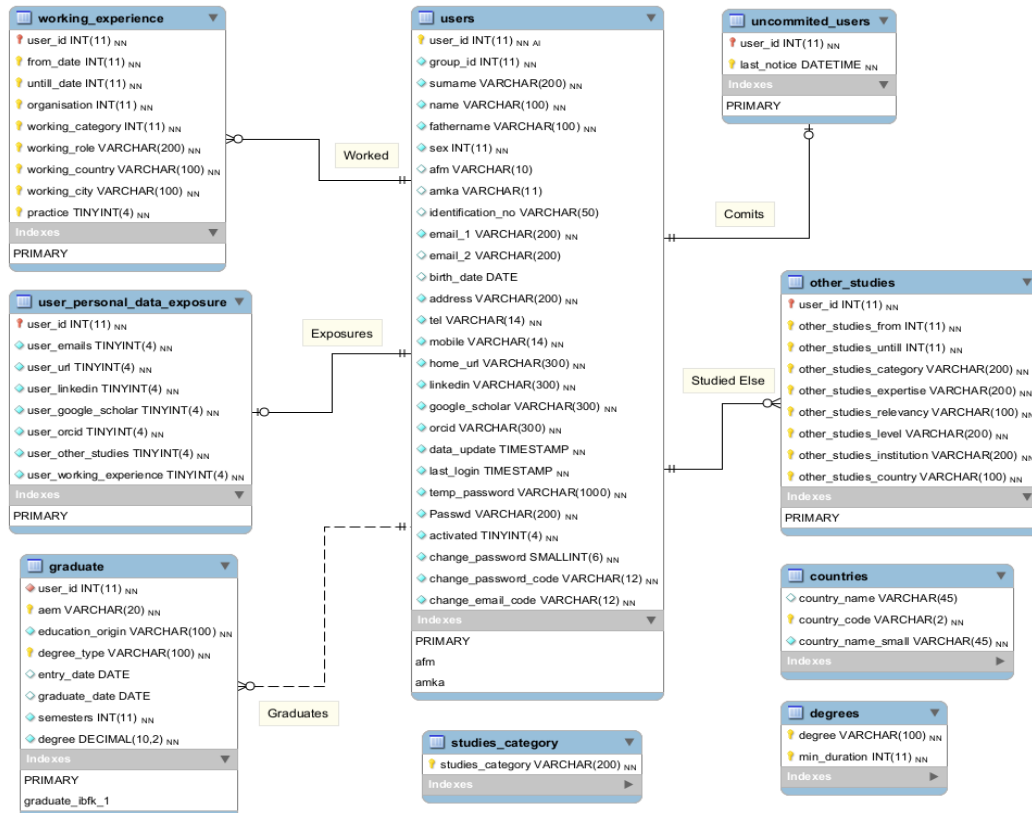
Επίσης υπάρχουν και τρεις βοηθητικές οντότητες που χρησιμοποιούνται για τη λειτουργική υποστήριξη drop down μενού και δεν έχουν κάποια συσχέτιση με τις υπόλοιπες:

- *countries*: Κατάλογος των χωρών της Γης με τις επίσημες ονομασίες τους
- *studies_category*: Κατάλογος πανεπιστημιακών τμημάτων εντός Ελλάδος

- degrees: Κατάλογος προσφερόμενων ακαδημαϊκών τίτλων από το τμήμα ΜΠΗΣ

4.2 Διάγραμμα ER

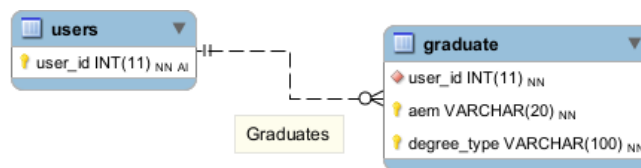
Παρακάτω ακολουθεί το ER διάγραμμα του σχεσιακού μοντέλου για την έκδοση IEEAlumni v.1 με βάση την μεθοδολογία απεικόνισης Crow's Foot Notation, όπως προέκυψε από το λογισμικό MySQL Workbench[10]:



Εικόνα 10: IEEAlumni v.1 ER

4.3 Ανάλυση σχέσεων μεταξύ οντοτήτων

4.3.1 Συσχέτιση users – graduate

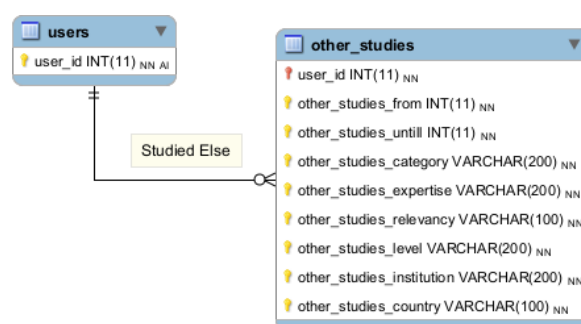


Εικόνα 11: IEEAlumni v.1 Συσχέτιση users – graduate

Πρόκειται για συσχέτιση «Ένα → Μηδέν, Ένα ή Πολλά». Ένα στιγμιότυπο της οντότητας users αν είναι τύπου «Γραμματεία» ή «Ομάδα ΟΜΕΑ» θα έχει συσχέτιση με μηδέν στιγμιότυπα της οντότητας graduate. Αλλιώς θα έχει συσχέτιση με ένα ή περισσότερα στιγμιότυπα της οντότητας graduate. Αντίστροφα ένα στιγμιότυπο της οντότητας graduate αντιστοιχεί υποχρεωτικά σε έναν και μόνο ένα στιγμιότυπο της οντότητας user.

Επιπλέον επειδή το πεδίο user_id της οντότητας graduate δεν είναι μέλος του κυρίως κλειδιού της, αλλά ταυτόχρονα ξένο κλειδί στην οντότητα users, η γραμμή που απεικονίζει τη συσχέτιση είναι διακεκομμένη (χαρακτηριστικό του λογισμικού MySQL Workbench).

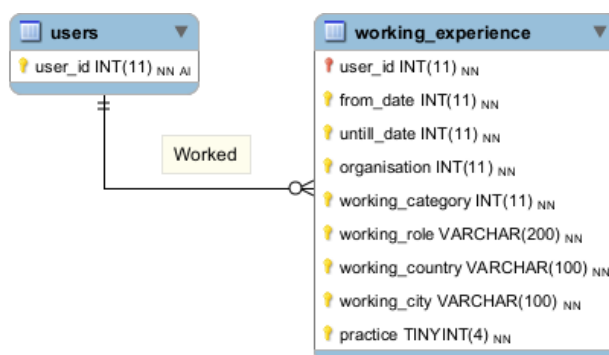
4.3.2 Συσχέτιση users – other_studies



Εικόνα 12: IEEAlumni v.1 Συσχέτιση users – other_studies

Πρόκειται για συσχέτιση «Ένα → Μηδέν, Ένα ή Πολλά». Ένα στιγμιότυπο της οντότητας users μπορεί να έχει συσχέτιση με μηδέν, ένα ή περισσότερα στιγμιότυπα της οντότητας other_studies. Αντίστροφα ένα στιγμιότυπο της οντότητας other_studies αντιστοιχεί υποχρεωτικά σε έναν και μόνο ένα στιγμιότυπο της οντότητας user.

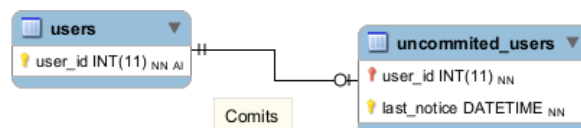
4.3.3 Συσχέτιση users – working_experience



Εικόνα 13: IEEAlumni v.1 Συσχέτιση users – working_experience

Πρόκειται για συσχέτιση «Ένα → Μηδέν, Ένα ή Πολλά». Ένα στιγμιότυπο της οντότητας users μπορεί να έχει συσχέτιση με μηδέν, ένα ή περισσότερα στιγμιότυπα της οντότητας working_experience. Αντίστροφα ένα στιγμιότυπο της οντότητας working_experience αντιστοιχεί υποχρεωτικά σε έναν και μόνο ένα στιγμιότυπο της οντότητας user.

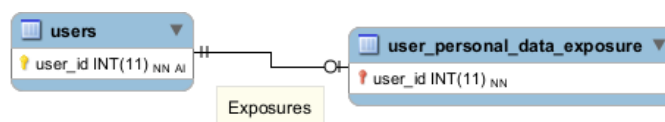
4.3.4 Συσχέτιση users – uncommitted_users



Εικόνα 14: IEEAlumni v.1 Συσχέτιση users – uncommitted_users

Πρόκειται για συσχέτιση «Ένα → Μηδέν ή Ένα». Ένα στιγμιότυπο της οντότητας users μπορεί να έχει συσχέτιση με μηδέν, ή ένα στιγμιότυπα της οντότητας uncommitted_users, επειδή ενδιαφέρει μόνο η χρονική στιγμή της τελευταίας ειδοποίησης που του στάλθηκε ή επειδή έχει κάνει επιβεβαίωση του λογαριασμού του. Αντίστροφα ένα στιγμιότυπο της οντότητας uncommitted_users αντιστοιχεί υποχρεωτικά σε έναν και μόνο ένα στιγμιότυπο της οντότητας user.

4.3.5 Συσχέτιση users – personal_data_exposure



Εικόνα 15: IEEAlumni v.1 Συσχέτιση users – personal_data_exposure

Πρόκειται για συσχέτιση «Ένα → Μηδέν ή Ένα». Ένα στιγμιότυπο της οντότητας users αν είναι τύπου «Γραμματεία» ή «Ομάδα OMEA» θα έχει συσχέτιση με μηδέν στιγμιότυπα της οντότητας personal_data_exposure. Αλλιώς θα έχει συσχέτιση με μηδέν (αν δεν έχει επιβεβαιώσει το λογαριασμό του δεν δημιουργείται στιγμιότυπο για την οντότητα personal_data_exposure) ή ένα στιγμιότυπα της οντότητας personal_data_exposure. Αντίστροφα ένα στιγμιότυπο της οντότητας personal_data_exposure αντιστοιχεί υποχρεωτικά σε έναν και μόνο ένα στιγμιότυπο της οντότητας user.

4.4 Πίνακες

4.4.1 Πίνακας *users*

Ο πίνακας *users* είναι ο κεντρικός πίνακας του σχήματος στον οποίο γίνονται όλες οι αναφορές ξένων κλειδιών. Καταγράφει τα προσωπικά στοιχεία κάθε χρήστη και περιέχει επιπλέον μερικά βοηθητικά πεδία για τον έλεγχο της διαδικασίας αλλαγής κωδικού, αλλαγής λογαριασμού αλληλογραφίας, επιβεβαίωση λογαριασμού κλπ.

4.4.2 Πίνακας *graduate*

Στον πίνακα *graduate* αποθηκεύονται οι αποφοιτήσεις από το τμήμα. Καταχωρούνται πληροφορίες όπως ο αριθμός μητρώου, ο κύκλος σπουδών που παρακολούθησε, η ημερομηνία έναρξης και ολοκλήρωσης των σπουδών, η διάρκεια τους σε εξάμηνα (επιλέχθηκε να μην γίνεται αυτόματος προσδιορισμός για την περίπτωση που έχει μεσολάβησει διακοπή φοίτησης) και ο βαθμός.

4.4.3 Πίνακας *other_studies*

Χρησιμοποιείται για την αποθήκευση των κύκλων σπουδών που ολοκλήρωσαν οι απόφοιτοι ή είναι ακόμη σε εξέλιξη και προσφέρονται από τμήματα εκτός του ΜΠΗΣ ή από άλλα ιδρύματα

4.4.4 Πίνακας *working_experience*

Χρησιμοποιείται για την αποθήκευση της επαγγελματικής εμπειρίας των αποφοίτων. Μεταξύ των άλλων στοιχείων που καταχωρούνται είναι η ένδειξη αν υπήρξε πρακτική άσκηση.

4.4.5 Πίνακας *user_personal_data_exposure*

Στον πίνακα αυτόν αποθηκεύονται οι καταχωρίσεις που έχουν σχέση με το βαθμό έκθεσης των προσωπικών δεδομένων ενός αποφοίτου.

4.4.6 Πίνακας *uncommitted_users*

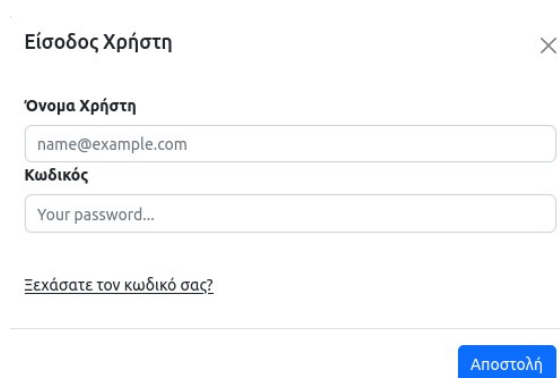
Ο πίνακας αυτός χρησιμοποιείται για τον έλεγχο των χρηστών που δεν έχουν επιβεβαιώσει τον λογαριασμό τους. Όταν του στέλνεται πρόσκληση για επιβεβαίωση του λογαριασμού, καταγράφεται η χρονοσφραγίδα της αποστολής του μηνύματος.

4.5 Λειτουργικότητα *Alumni v.1*

Οι βασικές λειτουργίες της έκδοσης *Alumni v.1* περιγράφονται παρακάτω:

4.5.1 Σύνδεση στην Εφαρμογή

Στην έκδοση Alumni v.1 υποστηρίζεται η καταχώριση έως και δύο λογαριασμών αλληλογραφίας. Η σύνδεση στην εφαρμογή γίνεται με οποιονδήποτε από τους δύο καταχωρημένους λογαριασμούς αλληλογραφίας και του προσωπικού κωδικού.



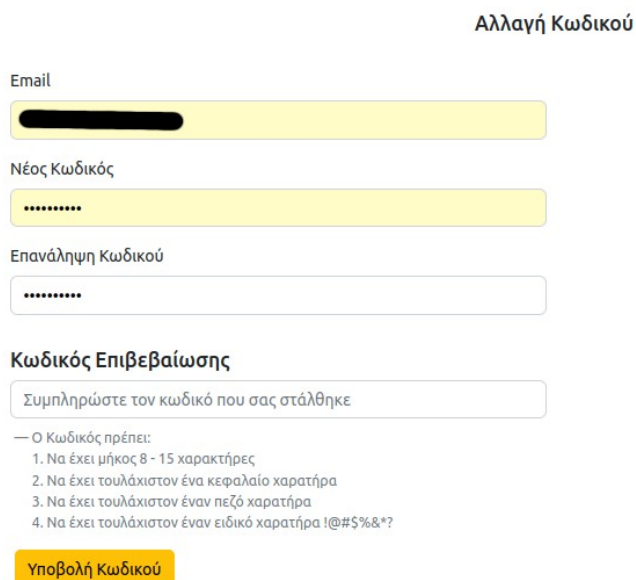
The screenshot shows a login form with the following elements:

- Title: Είσοδος Χρήστη
- Field: Όνομα Χρήστη (User Name) with the value name@example.com
- Field: Κωδικός (Password) with the placeholder text Your password...
- Link: Ξεχάσατε τον κωδικό σας?
- Button: Αποστολή (Submit)

Εικόνα 16: Σύνδεση στην Εφαρμογή

4.5.2 Αλλαγή Κωδικού

Για την αλλαγή κωδικού ο χρήστης πληκτρολογεί στη σχετική φόρμα τον λογαριασμό αλληλογραφίας του και τον επιθυμητό κωδικό. Εάν ο λογαριασμός αλληλογραφίας είναι ήδη καταχωρημένος η εφαρμογή του στέλνει κωδικό επιβεβαίωσης στον λογαριασμό αλληλογραφίας που δήλωσε.



The screenshot shows a password change form with the following elements:

- Title: Αλλαγή Κωδικού
- Field: Email (Redacted)
- Field: Νέος Κωδικός (New Password) (Redacted)
- Field: Επανάληψη Κωδικού (Repeat Password) (Redacted)
- Field: Κωδικός Επιβεβαίωσης (Confirmation Code) with the placeholder text Συμπληρώστε τον κωδικό που σας στάλθηκε
- Text: Ο Κωδικός πρέπει:
- List:
 1. Να έχει μήκος 8 - 15 χαρακτήρες
 2. Να έχει τουλάχιστον ένα κεφαλαίο χαρακτήρα
 3. Να έχει τουλάχιστον έναν πεζό χαρακτήρα
 4. Να έχει τουλάχιστον έναν ειδικό χαρακτήρα !@#%&*?
- Button: Υποβολή Κωδικού (Submit)

Figure 17: Αλλαγή Κωδικού

4.5.3 Καταχώρηση νέου Χρήστη από τη Γραμματεία

Η γραμματεία καταχωρεί χρήστες με ρόλο «Απόφοιτος», «Γραμματεία» και «Μέλος ΟΜΕΑ». Η εναλλαγή γίνεται με ένα drop down χωρίς να επαναφορτωθεί η σελίδα.

4.5.3.1 Καταχώρηση χρήστη τύπου «Απόφοιτος»

Κατά την καταχώρηση ενός νέου αποφοίτου υπάρχουν δύο ομάδες στοιχείων που καταγράφονται, τα ατομικά του στοιχεία και τα στοιχεία αποφοίτησης. Στα ατομικά στοιχεία υποχρεωτικά είναι τα πεδία "Ημ/νια Γέννησης", "Επίθετο", "Όνομα", "Πατρώνυμο", "Φύλλο", "Διεύθυνση Κατοικίας" και "Email_1" (δεν είναι υποχρεωτική η καταχώρηση 2ου λογαριασμού αλληλογραφίας)

Τύπος Χρήστη
ΑΠΟΦΟΙΤΟΣ

ΑΦΜ	ΑΜΚΑ	ΑΔΤ/Διαβατήριο	Ημ/νια Γέννησης*	Σταθερό Τηλ	Κινητό Τηλ
<input type="text"/>	<input type="text"/>	<input type="text"/>	12/11/2022	<input type="text"/>	<input type="text"/>
Επίθετο*	Όνομα*	Πατρώνυμο*	Φύλλο*		
<input type="text"/>	<input type="text"/>	<input type="text"/>	Ανδρας		
Διεύθυνση Κατοικίας*					
<input type="text"/>					
Email_1*	Email_2				
<input type="text"/>	<input type="text"/>				

— Τα πεδία με * είναι υποχρεωτικά

Εικόνα 18: Καρτέλα ατομικών στοιχείων

Τα στοιχεία αποφοίτησης αφορούν το πτυχίο που καταχωρείται και όλα τα πεδία είναι υποχρεωτικά.

Αποφοιτήσεις

ΑΕΜ	Κατηγορία Εισαγωγής	Τύπος Πτυχίου	Ημ/νια Εισαγωγής	Ημ/νια Αποφοίτησης	Διάρκεια	Βαθμός Πτυχίου
<input type="text"/>	ΓΕΛ	5-ΕΤΕΣ ΠΠΕ ΤΜΗΜΑΤΟΣ	12/11/2022	12/11/2022	10	5.00

Εικόνα 19: Καρτέλα καταχώρισης στοιχείων αποφοίτησης

4.5.3.2 Καταχώρηση χρήστη τύπου «Γραμματεία» ή «Μέλος ΟΜΕΑ»

Όταν πρόκειται για χρήστη τύπου «Γραμματεία» ή «Μέλος ΟΜΕΑ» υπάρχουν μόνο 4 πεδία προς συμπλήρωση. Και σε αυτήν την περίπτωση γίνονται οι έλεγχοι ύπαρξης στη βάση του προς καταχώρηση χρήστη.

Τύπος Χρήστη

ΓΡΑΜΜΑΤΕΙΑ

Επίθετο* Όνομα*

Email_1* Email_2

— Τα πεδία με * είναι υποχρεωτικά

Εικόνα 20: Καταχώρηση χρήστη τύπου «Γραμματεία» ή «Μέλος ΟΜΕΑ»

4.5.4 Ενημέρωση Στοιχείων Υφιστάμενου Χρήστη από τη Γραμματεία

Η ενημέρωση των στοιχείων ενός υφιστάμενου χρήστη γίνεται από το μενού της καταχώρησης νέου χρήστη. Η καρτέλα ενός υφιστάμενου χρήστη μπορεί αν ανακτηθεί με δύο τρόπους:

- Τα πεδία "ΑΦΜ", "ΑΜΚΑ", "Email_1" και το "Email_2" (αν υπάρχει δεύτερος λογαριασμός αλληλογραφίας) έχουν μοναδικές τιμές, οπότε αρκεί η πληκτρολόγηση της αντίστοιχης πληροφορίας και ακολουθεί ο έλεγχος στη βάση και η επιστροφή των αποτελεσμάτων. Η φόρμα συμπληρώνεται με τις υφιστάμενες τιμές και τα πεδία γίνονται μη επεξεργάσιμα από τον χρήστη/απόφοιτο.
- Το πεδίο "Επίθετο" δεν μπορεί να έχει μοναδικές τιμές. Γίνεται όμως έλεγχος για την ύπαρξη παρόμοιων επιθέτων (έλεγχος LIKE) και επιστρέφεται σχετικός κατάλογος από όπου μπορεί η γραμματεία να επιλέξει κάποιο αν θεωρεί ότι πληροί τα κριτήρια.

#	ΑΦΜ	ΑΜΚΑ	ΑΔΤ/ Διαβατήριο	Επίθετο	Όνομα	Πατρώνυμο	Ημ/νία Γέννησης	email_1	email_2
1	testafm	testamka	TESTID	TEST SURNAME	TEST NAME	TEST FATHERNAME	01/01/1960	██████████	
2	test afm	test amka	TEST ADT	TEST SURNAME	TEST NAME	TEST FATHERNAME	01/11/2022	██████████	

Εικόνα 21: Κατάλογος χρηστών με παρόμοιο επίθετο

Επιλέγοντας μια γραμμή από τον κατάλογο η φόρμα συμπληρώνεται με τις υφιστάμενες τιμές και τα πεδία γίνονται μη επεξεργάσιμα από τον χρήστη/απόφοιτο.

Η φόρμα είναι χωρισμένη σε δύο περιοχές, στα «Ατομικά Στοιχεία Χρήστη» και στις «Αποφοιτήσεις». Η επεξεργασία τους γίνεται με διαφορετικό τρόπο.

Για την επεξεργασία των ατομικών στοιχείων η γραμματεία πρέπει να επιλέξει «Ενημέρωση» ώστε να γίνουν επεξεργάσιμα τα πεδία και αφού κάνει τις τροποποιήσεις το «Επιβεβαίωση» (το κείμενο «Ενημέρωση» - «Επιβεβαίωση» εναλλάσσεται).

Με το «Επιβεβαίωση» αποστέλλονται με κλήση Ajax τα δεδομένα από όλα τα πεδία από σχετικά με τα «Ατομικά Στοιχεία Χρήστη» και γίνεται η αντίστοιχη ενημέρωση στη βάση. Η εναλλακτική προσέγγιση, η οποία επιλέχθηκε για την περιοχή «Αποφοιτήσεις», είναι με το που τροποποιείται ένα πεδίο με κλήση Ajax να ενημερώνεται η βάση και κατόπιν η περιοχή «Ατομικά Στοιχεία Χρήστη» να γίνεται μη επεξεργάσιμη. Σε περίπτωση λοιπόν που θα πρέπει να γίνει επέμβαση σε περισσότερα από ένα σημεία, ο χρήστης θα πρέπει να επαναλάβει τη διαδικασία.

Εικόνα 22: Ενημέρωση ατομικών στοιχείων

Η προσέγγιση που επιλέχθηκε για τις «Αποφοιτήσεις» είναι διαφορετική. Κάθε πεδίο μια υφιστάμενης αποφοίτησης μπορεί να τροποποιηθεί αυτόνομα. Για την τροποποίηση μιας εγγραφής πρέπει να επιλεγεί το εικονίδιο με το πινέλο της σχετικής γραμμής ώστε η γραμμή να γίνει επεξεργάσιμη. Όταν αλλάξει μια τιμή, γίνεται καταχώριση στη βάση και η γραμμή επανέρχεται σε κατάσταση μη επεξεργασίας. Αν την τροποποιεί κάποιος λογαριασμός αλληλογραφίας η εφαρμογή στέλνει σχετικό μήνυμα στον χρήστη.

Αποφοιτήσεις

ΑΕΜ	Κατηγορία Εισαγωγής	Τύπος Πτυχίου	Ημ/νια Εισαγωγής	Ημ/νια Αποφοίτησης	Διάρκεια	Βαθμός Πτυχίου	
123	ΓΕΛ	5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ	01/09/2014	19/09/2018	8	8.50	✓
567	ΒΑΣΙΚΟ ΠΤΥΧΙΟ ΑΕΙ/ΤΕ	ΠΜΣ ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟ	01/09/2020	21/09/2020	4	9.00	✓

Καθαρισμός Νέο Πτυχίο

Εικόνα 23: Τροποποίηση στοιχείων αποφοίτησης

Επίσης υπάρχει η δυνατότητα προσθήκης νέου πτυχίου. Πατώντας το σχετικό πλήκτρο εμφανίζεται γραμμή για την καταχώριση της πληροφορίας.

Αποφοιτήσεις

ΑΕΜ	Κατηγορία Εισαγωγής	Τύπος Πτυχίου	Ημ/νια Εισαγωγής	Ημ/νια Αποφοίτησης	Διάρκεια	Βαθμός Πτυχίου	
	ΓΕΛ	5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ	13/11/2022	13/11/2022	10	5.00	✓
123	ΓΕΛ	5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ	01/09/2014	19/09/2018	8	8.50	✓
567	ΒΑΣΙΚΟ ΠΤΥΧΙΟ ΑΕΙ/ΤΕ	ΠΜΣ ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟ*	01/09/2020	21/09/2020	4	9.00	✓

Καθαρισμός

Νέο Πτυχίο

Εικόνα 24: Προσθήκη αποφοίτησης

4.5.5 Αποστολή Μηνύματος Ενεργοποίησης Λογαριασμού

Προκειμένου να ενεργοποιηθεί ένας λογαριασμός θα πρέπει πρώτα να αποσταλεί πρόσκληση στον χρήστη με έναν προσωρινό κωδικό. Το μήνυμα ενημερώνει τον χρήστη για τη δημιουργία του λογαριασμού από την γραμματεία και τον καλεί να τον ενεργοποιήσει.

Η γραμματεία μπορεί να ελέγχει ποιοι χρήστες δεν έχουν ενεργοποιήσει τον λογαριασμό τους και να τους στέλνει υπενθυμίσεις. Από το μενού «Εκκρεμούντες Επιβεβαίωσης» προκύπτει σχετικός πίνακας

Μη Ενεργοποιημένοι Χρήστες

Εμφάνιση 10 εγγραφών ανά σελίδα

Αναζήτηση

<input type="checkbox"/>	Επίσημο	Όνομα	Σπουδές	EMAIL_1	EMAIL_2	Τελευταία ενημέρωση
<input type="checkbox"/>			5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ			2022-10-31 13:15:56
<input type="checkbox"/>			5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ			2022-10-31 13:15:57
<input type="checkbox"/>			5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ			2022-10-31 13:15:58
<input type="checkbox"/>			5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ			2022-10-31 13:16:02
<input type="checkbox"/>			5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ			2022-10-31 13:16:11
<input type="checkbox"/>			5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ			2022-10-31 13:16:01
<input type="checkbox"/>			5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ			2022-10-31 13:16:07
<input type="checkbox"/>			5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ			2022-10-31 13:16:11
<input type="checkbox"/>			5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ			2022-10-31 13:16:05
<input type="checkbox"/>			5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ			2022-10-31 13:16:00

Σελίδα 1 από 3

Προηγούμενο 1 2 3 Επόμενο

Αποστολή

Εικόνα 25: Κατάλογος χρηστών που εκκρεμεί η επιβεβαίωση

Από τον πίνακα μπορεί να επιλέξει τους χρήστες που θέλει να ενημερώσει και η εφαρμογή τους στέλνει το σχετικό μήνυμα και καταχωρεί τη σχετική χρονοσφραγίδα.

4.5.6 Κατάλογος Όλων των Αποφοίτων

Για αναλυτικότερη αναζήτηση η γραμματεία μπορεί να χρησιμοποιήσει το μενού «Κατάλογος Όλων των Αποφοίτων», όπου εφαρμόζοντας ταξινόμηση ή αναζήτηση μπορεί να εντοπίσει αυτόν που αναζητά.

Κατάλογος Όλων των Αποφοίτων

Εμφάνιση 10 εγγραφών ανά σελίδα

Αναζήτηση

Επίθετο	Όνομα	ΑΦΜ	ΑΜΚΑ	Σπουδές	EMAIL_1	EMAIL_2	Ενεργοποιημένος
				ΔΙΔΑΚΤΟΡΙΚΟ			ΟΧΙ
TEST SURNAME	TEST NAME	testafm	testamka	ΔΙΔΑΚΤΟΡΙΚΟ			ΟΧΙ
TEST SURNAME	TEST NAME	test afm	test amka	5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ ΠΜΣ ΕΠΗΡΕΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ			ΝΑΙ
				5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ			ΝΑΙ
				5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ			ΟΧΙ
				5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ			ΝΑΙ
				5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ			ΟΧΙ
				5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ			ΟΧΙ
				5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ			ΟΧΙ
				5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ			ΝΑΙ

Σελίδα 1 από 4

Προηγούμενο 1 2 3 4 Επόμενο

Εικόνα 26: Κατάλογος όλων των αποφοίτων

4.5.7 Ενημέρωση Ατομικών Στοιχείων από Απόφοιτο

Ο απόφοιτος μπορεί να τροποποιήσει μέρος των προσωπικών του στοιχείων. Τα πεδία Επίθετο, Όνομα, Πατρώνυμο, Ημ/νια Γέννησης, ΑΦΜ και ΑΜΚΑ, μπορούν να ενημερωθούν μόνο από τη γραμματεία.

Προσωπικά Στοιχεία

Επίθετο TEST SURNAME	Όνομα TEST NAME	Πατρώνυμο TEST FATHERNAME	Ημ/νια Γέννησης 18/11/1992
ΑΦΜ test afm	ΑΜΚΑ test amka	ΑΔΤ/Διαβατήριο TEST ADT	
Διεύθυνση Κατοικίας TEST ADDRESS DSSD	Σταθερό Τηλέφωνο test tel	Κινητό Τηλέφωνο test tel	
Home URL myurl	LinkedIn mylinkedin		
Google Scholar mygooglescholar	ORCID myorcid		

Ενημέρωση

Email_1
taouktsis.dipe.thess@gmail.com Αλλαγή

Email_2
 Αλλαγή

Εικόνα 27: Ενημέρωση ατομικών στοιχείων από απόφοιτο

Αν επιλέξει να ενημερώσει κάποιο email, ακολουθείται διαδικασία πιστοποίησης με την αποστολή κωδικού επιβεβαίωσης στον νέο λογαριασμό αλληλογραφίας που δήλωσε.

4.5.8 Προσθήκη/Διαγραφή Άλλων Σπουδών από Απόφοιτο

Ο απόφοιτος μπορεί να προσθέσει/διαγράψει καταχωρίσεις σε σχέση με επιπλέον σπουδές που έκανε σε άλλες σχολές ή τμήματα.

Άλλες Σπουδές

Από	Έως	Ακαδημαϊκό Ίδρυμα	Κατηγορία	Επίπεδο	Εξειδίκευση	Συνάφεια	Χώρα
2013	2017	ΑΠΘ	ΜΑΘΗΜΑΤΙΚΩΝ	ΠΡΟΠΤΥΧΙΑΚΟ	ΕΦΑΡΜΟΣΜΕΝΑ ΜΑΘΗΜ.	ΠΛΗΡΟΦΟΡΙΚΗ	GB

Εικόνα 28: Προσθήκη/Διαγραφή άλλων σπουδών από απόφοιτο

4.5.9 Προσθήκη/Διαγραφή Επαγγελματικής Εμπειρίας από Απόφοιτο

Ο απόφοιτος μπορεί να προσθέσει/διαγράψει καταχωρίσεις σε σχέση με την επαγγελματική του εμπειρία.

Επαγγελματική Σταδιοδρομία

Από	Έως	Αντικείμενο Φορέα/Εργοδότη	Αντικείμενο Εργασίας	Ρόλος (πχ Προγραμματιστής)	Χώρα	Πόλη	Πρακτική*
2018	Σήμερα	Πληροφορική	Πληροφορική	ΠΡΟΓΡΑΜΜΑΤΙΣΤΗΣ	GR	ΘΕΣΣΑΛΟΝΙΚΗ	Όχι
2022	Σήμερα	Πληροφορική & Ηλεκτρονικά/Ψηφια	Πληροφορική & Ηλεκτρονικά/Ψηφια	ΠΡΟΓΡΑΜΜΑΤΙΣΤΗΣ	GR	ΘΕΣΣΑΛΟΝΙΚΗ	Όχι

— Πρακτική*
Ναι – Έκανε πρακτική άσκηση σε αυτόν τον οργανισμό και προσελήφθη
Όχι – Προσελήφθη χωρίς να κάνει πρακτική σε αυτόν τον οργανισμό

Εικόνα 29: Προσθήκη/Διαγραφή επαγγελματικής εμπειρίας από απόφοιτο

4.5.10 Επιλογή Δημοσιοποίησης Προσωπικών Δεδομένων από Απόφοιτο

Ο απόφοιτος έχει τη δυνατότητα να αναζητήσει άλλους αποφοίτους και να δει προσωπικά τους στοιχεία εφόσον αυτοί το επιτρέπουν. Επιλέγει αντίστοιχα τα προσωπικά του στοιχεία που θα δημοσιοποιεί στους άλλους αποφοίτους.

Επίπεδο Έκθεσης Προσωπικών Δεδομένων

Πλέον του Ονοματεπώνυμου, των τίτλων σπουδών και της ημερομηνίας αποφοίτησης από το ΔΙ.ΠΑ.Ε. επιτρέπετε την κοινοποίηση των παρακάτω:

- Email
- Home Url
- LinkedIn
- Google Scholar
- ORCID
- Άλλες Σπουδές
- Επαγγελματική Σταδιοδρομία

Ενημέρωση

Εικόνα 30: Επιλογή δημοσιοποίησης προσωπικών δεδομένων από απόφοιτο

4.5.11 Αναζήτηση Άλλων Αποφοίτων

Ο απόφοιτος έχει τη δυνατότητα να αναζητήσει άλλους αποφοίτους και να δει προσωπικά τους στοιχεία εφόσον αυτοί το επιτρέπουν. Μπορεί να ενημερωθεί για την ακαδημαϊκή τους εξέλιξη, την επαγγελματική τους σταδιοδρομία, για την ιστοσελίδα τους, τον λογαριασμό αλληλογραφίας κλπ

4.5.12 Αποστολή Μηνύματος από Απόφοιτο σε Απόφοιτο

Ο απόφοιτος μπορεί από την «Αναζήτηση Αποφοίτων» να στείλει ένα σύντομο μήνυμα ηλεκτρονικής αλληλογραφίας σε έναν άλλον απόφοιτο μέσω της εφαρμογής.

Αποστολή Μηνύματος

×

Αποστολέας

Email Αποστολέα

Παραλήπτης

Θέμα

Μήνυμα

Εικόνα 31: Αποστολή Μηνύματος από Απόφοιτο σε Απόφοιτο

4.5.13 Μαζικού Τύπου Προσθήκη Νέων Αποφοίτων

Μετά από κάθε ορκωμοσία η γραμματεία εξάγει από το UniPortal σε αρχείο excel, τον κατάλογο όσων ολοκλήρωσαν τις σπουδές τους στο τρέχον εξάμηνο. Κατόπιν ένας χρήστης του τύπου «Μέλος ΟΜΕΑ» αναλαμβάνει να εκτελέσει χειροκίνητα το σχετικό κώδικα που θα αναλάβει τη μαζική καταχώριση αυτών των εγγραφών στην υπηρεσία IEEAlumni.

Ο κώδικας που εκτελείται κάνει όλους τους βασικούς ελέγχους για αποφυγή διπλοεγγραφών. Όπου υπάρχει ένδειξη διπλοεγγραφής, δεν καταχωρεί την εγγραφή και ενημερώνει το αρχείο καταγραφής με τα σχετικά μηνύματα. Για τις εγγραφές που απέτυχε η αυτόματη καταχώριση τους, θα πρέπει να γίνει χειροκίνητη καταχώριση από την γραμματεία.

5

IEEAlumni v.2

Η έκδοση IEEAlumni v.1 υλοποιήθηκε με κύριο μέλημα την παράδοση όσον το δυνατόν πιο άμεσα ενός λειτουργικό ως προς τις βασικές απαιτήσεις λογισμικού. Στην έκδοση IEEAlumni v.1 δεν υπήρξε η δυνατότητα υλοποίησης διεπαφής API. Η εμπειρία που αποκτήθηκε κατά την υλοποίηση της έκδοσης αυτής, καθώς και οι ανάγκες όπως αυτές διατυπώθηκαν από τα μέλη της ομάδας OMEA έθεσαν τη βάση για τον εκ νέου σχεδιασμό της επόμενης έκδοσης εφαρμογής, IEEAlumni v.2 η οποία θα βασίζεται στην αρχιτεκτονική REST API. Η νέα έκδοση της εφαρμογής εκτίθεται, σε μία πρώτη προσέγγιση, στη συνέχεια.

5.1 Σχεσιακό μοντέλο

Το μοντέλο μας αποτελείται από οκτώ βασικές οντότητες και πέντε βοηθητικές οντότητες

Βασικές οντότητες είναι οι:

1. users: Η οντότητα που αφορά στα προσωπικά στοιχεία ενός χρήστη. Στην οντότητα αυτή συμπεριλαμβάνεται και οι πληροφορίες που καταχωρούνταν στην οντότητα user_personal_data_exposure της έκδοσης Alumni v.1
2. degrees: Η οντότητα που αφορά στα προγράμματα σπουδών που προσφέρει το τμήμα ΜΠΗΣ.
3. roles: Η οντότητα που καθορίζει το ρόλο ενός χρήστη (Διαχειριστής, Γραμματεία, Μέλος OMEA, Απόφοιτος).

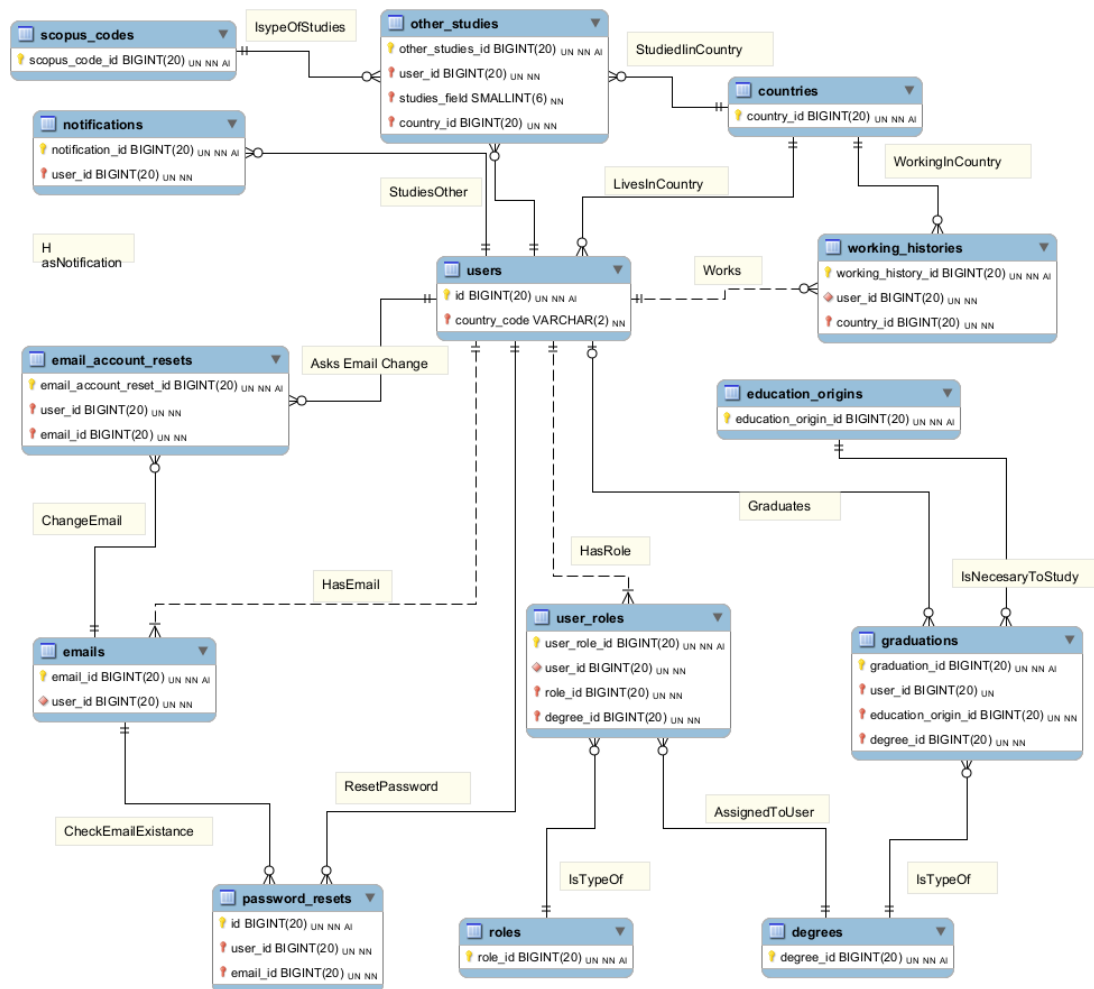
4. `graduations`: Η οντότητα που αφορά στις αποφοιτήσεις από το τμήμα ΜΠΗΣ (`graduate` στην έκδοση Alumni v.1)
5. `other_studies`: Η οντότητα που αφορά στις σπουδές σε άλλα τμήματα
6. `working_history`: Η οντότητα που αφορά στην επαγγελματική εμπειρία (`working_experience` στην έκδοση Alumni v.1)
7. `emails`: Η οντότητα που αφορά στους λογαριασμούς αλληλογραφίας που μπορεί να έχει κάθε χρήστης. Στην έκδοση Alumni v.1 ένας χρήστης μπορούσε να έχει έως δύο λογαριασμούς αλληλογραφίας και η πληροφορία αποθηκευόταν στον πίνακα `users`. Πλέον ο χρήστης μπορεί να έχει απεριόριστους λογαριασμούς αλληλογραφίας.
8. `notifications`: Η οντότητα που αφορά στις διάφορες ειδοποιήσεις που δέχονται οι χρήστες από την εφαρμογή. Στην έκδοση Alumni v.1 υπήρχε η οντότητα `uncommitted_users` που πλέον καταργείται. Σε αυτήν αποθηκευόταν η χρονική στιγμή της τελευταίας πρόσκλησης προς τους χρήστες για ενεργοποίηση του λογαριασμού τους. Η οντότητα `notifications` μπορεί να χρησιμοποιηθεί για καταγραφή όλων των ειδοποιήσεων που θα αποσταλούν από το σύστημα στους χρήστες.

Βοηθητικές οντότητες που χρησιμοποιούνται για την συμπλήρωση `drop down` μενού ή για την καταχώριση προσωρινών τιμών είναι οι:

1. `education_origins`: Η οντότητα που αφορά στους τίτλους σπουδών που μπορούν να χρησιμοποιηθούν για πρόσβαση στο τμήμα ΜΠΗΣ, δηλαδή τον τίτλο σπουδών που χρησιμοποιήθηκε προκειμένου να γίνει η εγγραφή στο τμήμα.
2. `password_resets`: Ο πίνακας αυτός χρησιμοποιείται για την καταχώριση ενός προσωρινού κωδικού για την επιβεβαίωση της πρόθεσης του χρήστη για αλλαγή του κωδικού του.
3. `email_account_reset`: Ο πίνακας αυτός χρησιμοποιείται για την καταχώριση ενός προσωρινού κωδικού για την επιβεβαίωση της πρόθεσης του χρήστη για αλλαγή ενός λογαριασμού αλληλογραφίας.
4. `countries`: Οντότητα που αφορά στον κατάλογο των χωρών της Γης με τις επίσημες ονομασίες τους
5. `scopus_codes`: Οντότητα που αφορά στις θεματικές επιστημονικές περιοχές που καλύπτονται από το Scopus Database[11]. Στην έκδοση Alumni v.1 χρησιμοποιήθηκε η οντότητα `studies_category` που περιείχε τον κατάλογο των πανεπιστημιακών τμημάτων εντός Ελλάδος

5.2 Διάγραμμα ER

Παρακάτω ακολουθεί το ER διάγραμμα του σχεσιακού μοντέλου για την έκδοση IEEAlumni v.2 με βάση την μεθοδολογία απεικόνισης Crow's Foot Notation, όπως προέκυψε από το λογισμικό MySQL Workbench:

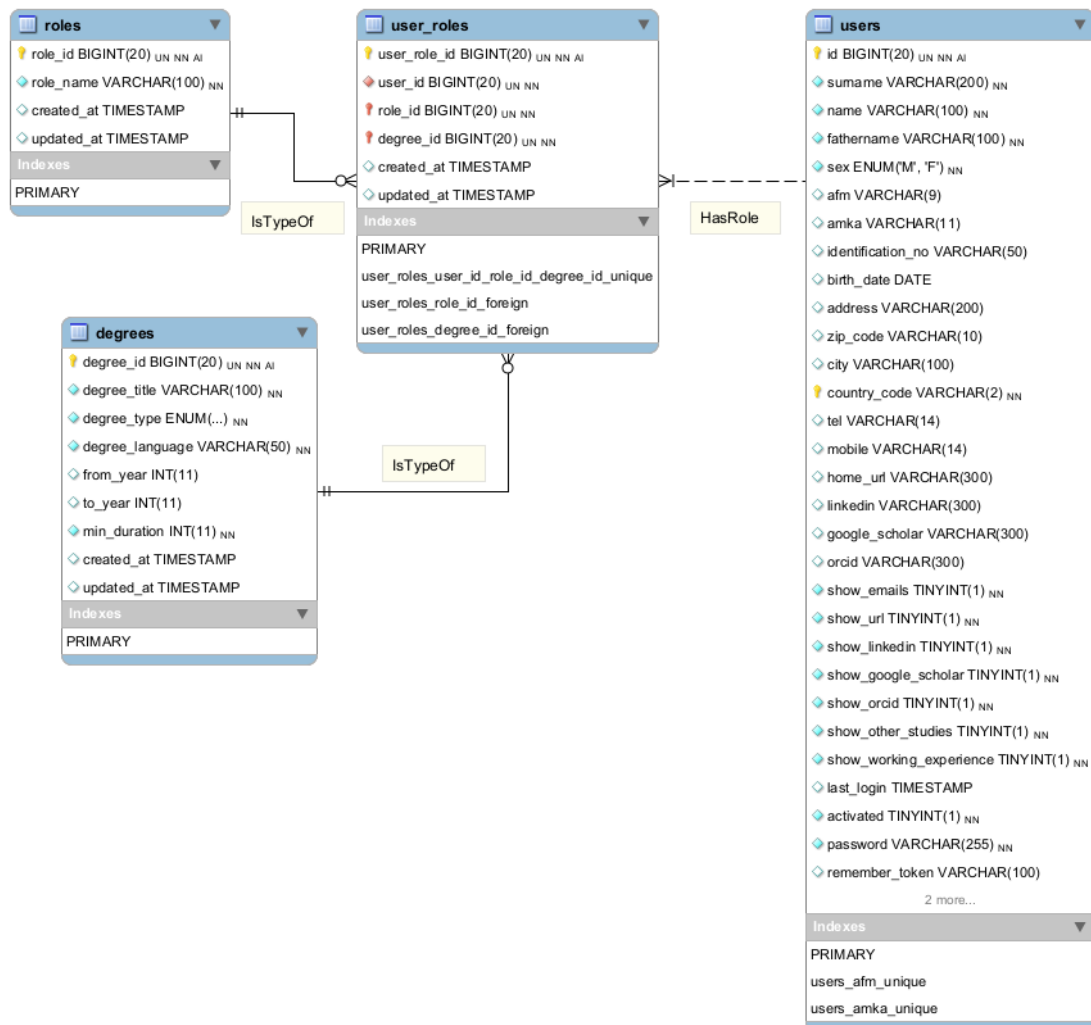


Εικόνα 32: IEEAlumni v.2 - Διάγραμμα ER

5.3 Ανάλυση σχέσεων μεταξύ οντοτήτων

Το διάγραμμα ER της έκδοσης Alumni v.1 είναι πιο απλό και περιλαμβάνει πέντε συσχετίσεις. Στην έκδοση Alumni v.2 έχουμε περισσότερες οντότητες και περισσότερες συσχετίσεις. Οι διαφορές μεταξύ των δύο υλοποιήσεων παρατίθενται στις επόμενες παραγράφους.

5.3.1 Συσχέτιση users – roles – degrees

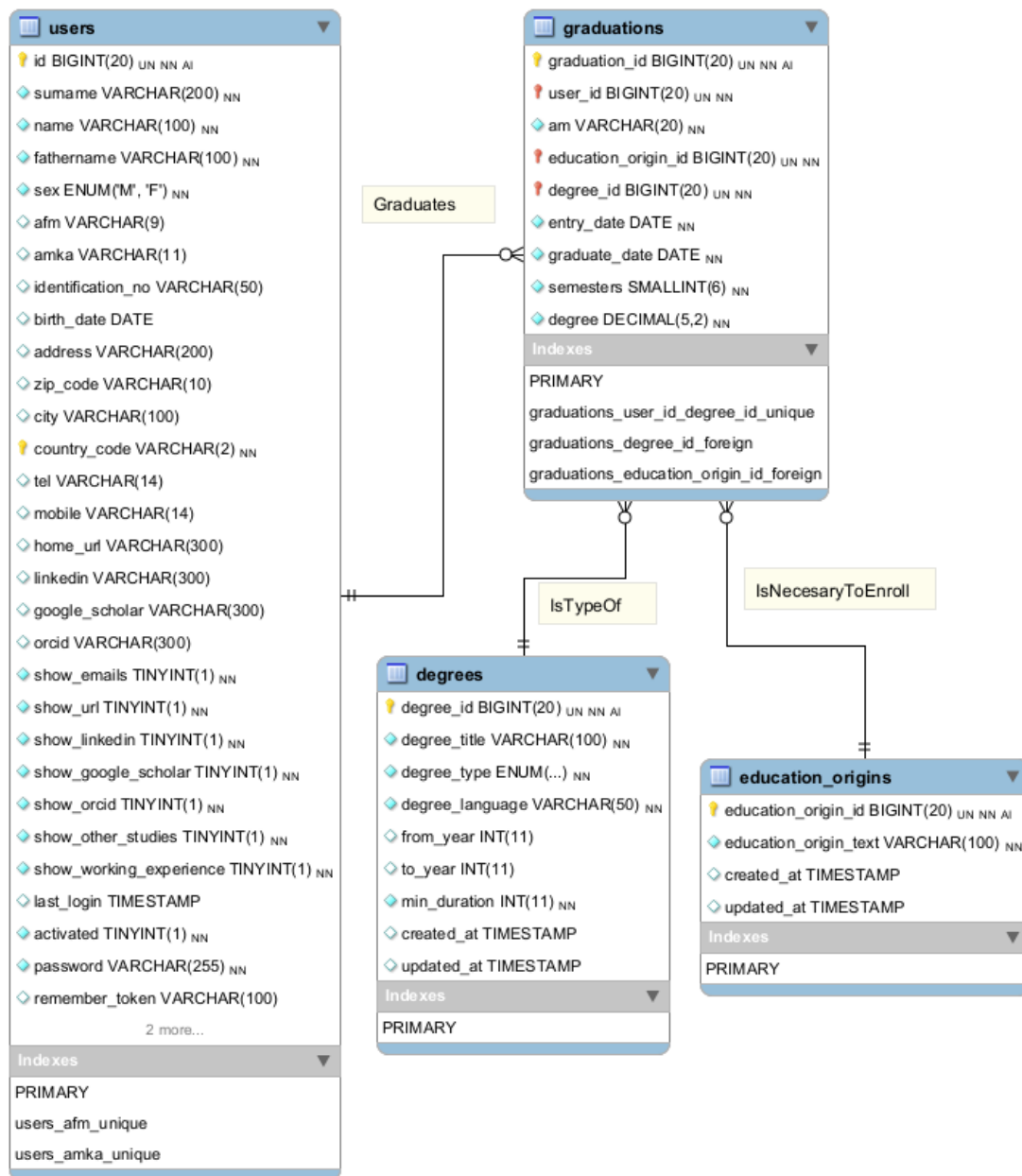


Εικόνα 33: IEEAlumni v.2 Συσχέτιση users – roles – degrees (Λεπτομέρειες)

Πρόκειται για τριαδική συσχέτιση "Πολλά προς Πολλά" μεταξύ των οντοτήτων users (χρήστες), degrees (τιμήμα), roles (ρόλοι) που υλοποιείται με τη χρήση του πίνακα user_roles. Κάθε χρήστης μπορεί να έχει πολλαπλούς ρόλους και να συσχετίζεται μέσω των ρόλων με τμήματα. Πχ ένας χρήστης μπορεί να έχει το ρόλο γραμματείας σε περισσότερα από ένα τμήματα και ταυτόχρονα να είναι απόφοιτος κάποιου από αυτά. Στην έκδοση Alumni v.1 δεν υπάρχει αυτή η ευελιξία και σε κάθε χρήστη μπορεί να ανατεθεί μοναδικός ρόλος που προσδιοριζόταν με το πεδίο group_id του πίνακα users.

Ένας χρήστης πρέπει οπωσδήποτε να συσχετίζεται με έναν ρόλο και ένα τμήμα, ενώ δεν είναι απαραίτητο (τουλάχιστον κατά την έναρξη λειτουργίας της εφαρμογής) ένας ρόλος ή ένα τμήμα να συσχετίζεται με κάποιον χρήστη.

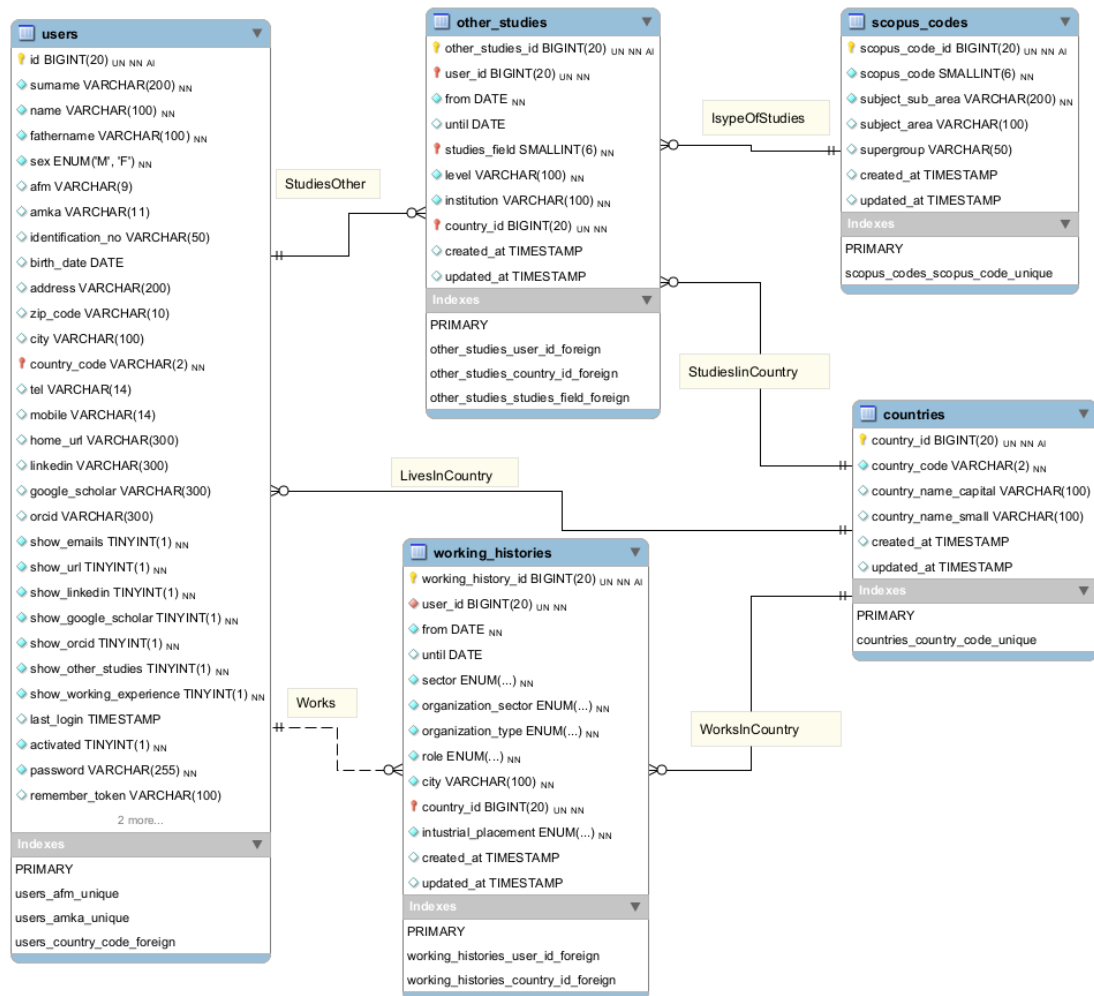
5.3.2 Συσχέτιση users – graduations



Εικόνα 34: IEEAlumni v.2 Συσχέτιση users – graduations (Λεπτομέρειες)

Ένας χρήστης μπορεί να συσχετίζεται με καμία (πχ Γραμματεία), μία ή περισσότερες αποφοιτήσεις ενώ κάθε αποφοίτηση συσχετίζεται μόνο με έναν χρήστη. Σε σχέση με την έκδοση Alumni v.1 προστίθεται επιπλέον ο πίνακας education_origins. Στον πίνακα αυτόν καταγράφονται οι διαφορετικές τιμές προέλευσης προκειμένου να εγγραφεί ένας σπουδαστής σε πρόγραμμα σπουδών του ΜΠΗΣ (πχ ΓΕΛ, Πτυχίο άλλου τμήματος, μεταπτυχιακό κ.ο.κ). Επίσης ο πίνακας degrees στον οποίο καταγράφονται τα προγράμματα σπουδών που διαθέτει το τμήμα ΜΠΗΣ έχει περισσότερα πεδία.

5.3.3 Συσχετίσεις users – other_studies/ working_history/countries



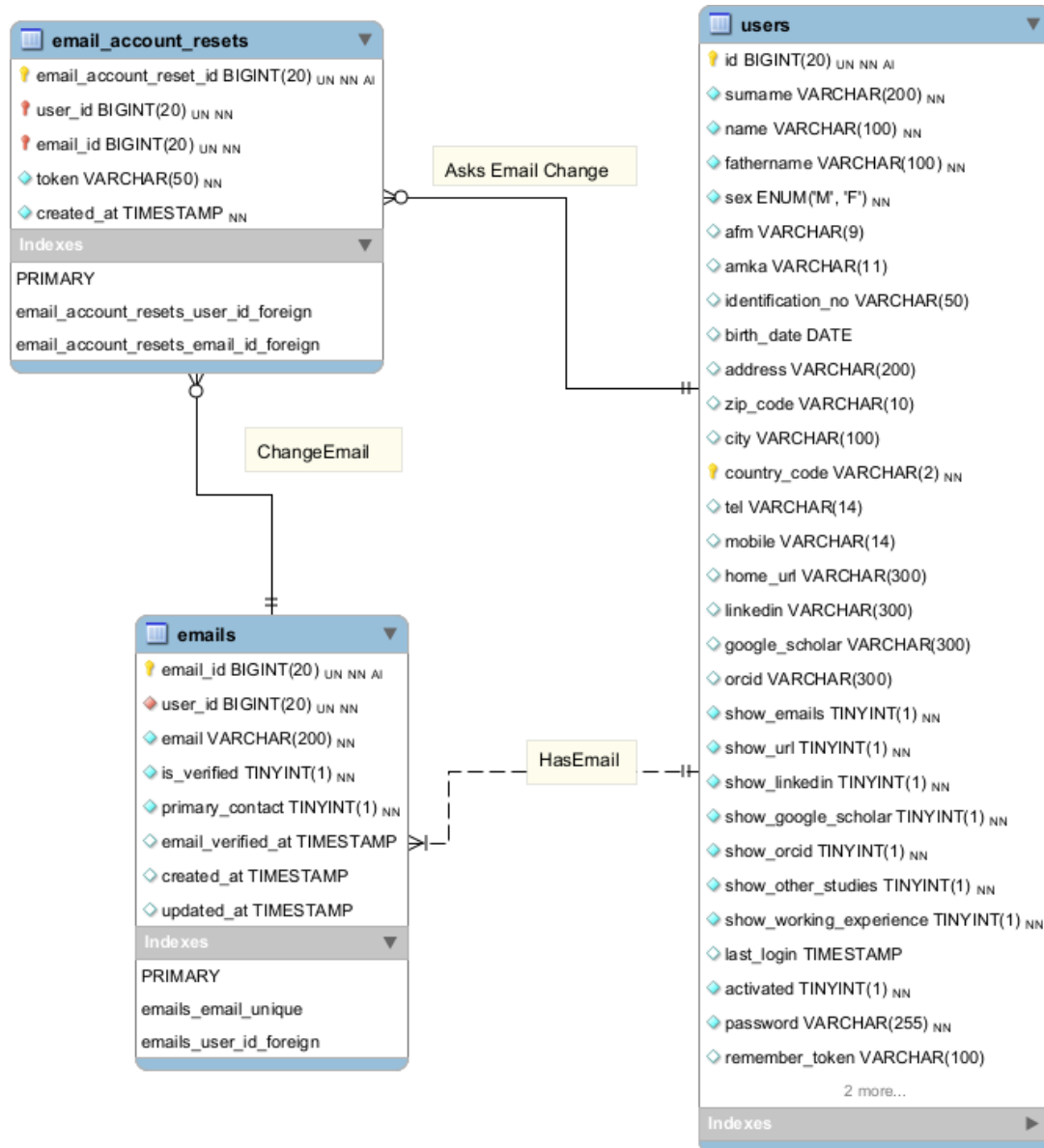
Εικόνα 35: IEEAlumni v.2 Συσχετίσεις users – other_studies/ working_history/countries (Λεπτομέρειες)

Ένας χρήσης μπορεί να έχει καθόλου, ένα ή περισσότερα πτυχία από άλλα τμήματα και να έχει καθόλου, μια ή περισσότερες εργασιακές εμπειρίες. Επίσης ένας χρήστης είναι κάτοικος μιας χώρας ενώ δεν είναι απαραίτητο όλες οι χώρες να συσχετίζονται με χρήστες.

Σε σχέση με την έκδοση Alumni v.1 έχουμε τις παρακάτω διαφορές:

- Ο πίνακας countries συσχετίζεται με τον πίνακα users με τη σχέση 'LivesInCountry'
- Ο πίνακας working_histories (ο πληθυντικός χρησιμοποιήθηκε για λόγους τυποποίησης του Laravel) έχει διαφορετική δομή από την έκδοση Alumni v.1
- Ο πίνακας scopus_codes αντικαθιστά τον studies_categories της έκδοσης Alumni v.1 και συσχετίζεται με τον other_studies

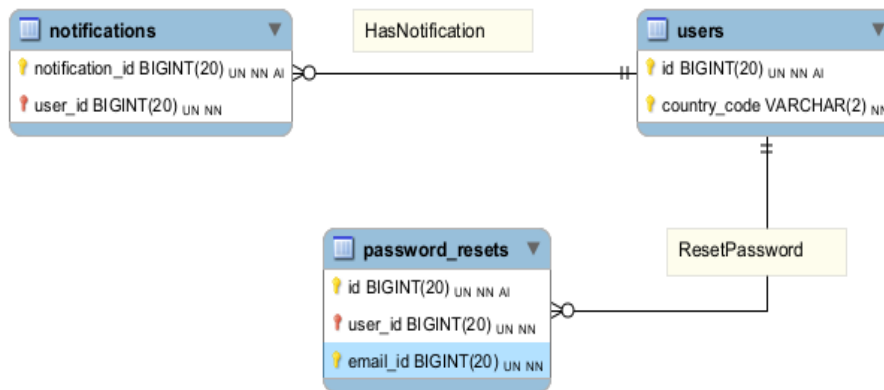
5.3.4 Συσχέτιση users – emails



Εικόνα 36: IEEAlumni v.2 Συσχέτιση users – emails (Λεπτομέρειες)

Ένας χρήστης μπορεί να έχει έναν ή περισσότερους λογαριασμούς αλληλογραφίας ενώ στην έκδοση Alumni v.1 είχε έως δύο. Για να υποστηριχθεί αυτό δημιουργήθηκε ο πίνακας emails που περιλαμβάνει και τα πεδία is_verified για να δηλωθεί αν επιβεβαιώθηκε ο λογαριασμός και primary_contact για να δηλωθεί αν είναι ο κύριος λογαριασμός επικοινωνίας του χρήστη. Επίσης υπάρχει ο πίνακας email_account_resets (ο πληθυντικός λόγω τυποποίησης του Laravel), χρησιμοποιείται κατά τη διαδικασία αλλαγής λογαριασμού αλληλογραφίας

5.3.5 Συσχέτιση users – notifications/password_resets



Εικόνα 37: IEEAlumni v.2 Συσχέτιση users – notifications/password_resets (Λεπτομέρειες)

Ένας χρήστης μπορεί να συσχετίζεται με καμία, μια ή περισσότερες ειδοποιήσεις από το σύστημα και να έχει αλλάξει κωδικό καμία, μια ή περισσότερες φορές (αν χρησιμοποιεί διαφορετικούς λογαριασμούς αλληλογραφίας)

5.4 Πίνακες

Οι πίνακες δημιουργήθηκαν με την εντολή migration του Laravel. Το Laravel επικοινωνεί με τη βάση δεδομένων μέσα από ρυθμίσεις που έχουμε κάνει στο αρχείο `.env`

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=database_name
DB_USERNAME=database_user
DB_PASSWORD=database_user_password
```

Για κάθε πίνακα τρέχουμε την εντολή `php artisan make:migration create_tablename_table` ώστε να δημιουργηθεί το αντίστοιχο αρχείο στο οποίο θα γίνει η περιγραφή της δομής του πίνακα. Εξ ορισμού με την εγκατάσταση του Laravel υπάρχουν τα παρακάτω αρχεία migration:

```
create_users_table
create_password_resets_table
create_failed_jobs_table
create_personal_access_tokens_table
```

Για τους πίνακες της εφαρμογής μας εκτελέσαμε τις εντολές:

```
php artisan make:migration create_groups_table;
php artisan make:migration create_user_group_table;
php artisan make:migration create_graduations_table;
php artisan make:migration create_other_studies_table;
php artisan make:migration create_working_history_table;
```

```
php artisan make:migration create_emails_table;
php artisan make:migration create_degrees_table;
php artisan make:migration create_email_account_reset_table;
php artisan make:migration create_countries_table;
php artisan make:migration create_notifications_table;
php artisan make:migration create_scopus_codes_table;
php artisan make:migration create_education_origins_table;
```

Για τους περιορισμούς ξένου κλειδιού δημιουργήσαμε τα αντίστοιχα αρχεία migration:

```
php artisan make:migration add_fk_to_user_group;
php artisan make:migration add_fk_to_graduations;
php artisan make:migration add_fk_to_other_studies;
php artisan make:migration add_fk_to_working_history;
php artisan make:migration add_fk_to_emails;
php artisan make:migration add_fk_to_email_account_reset;
php artisan make:migration add_fk_to_notifications;
php artisan make:migration add_fk_to_password_resets;
php artisan make:migration add_fk_to_users;
```

Για την δημιουργία των εγγραφών στη βάση πρέπει ακολουθεί η εντολή:

```
php artisan migrate;
```

5.4.1 Πίνακας users

Ο πίνακας users είναι ο κεντρικός πίνακας του σχήματος μας στον οποίο γίνονται όλες οι αναφορές ξένων κλειδιών. Σε αυτόν καταγράφονται τα προσωπικά στοιχεία κάθε χρήστη. Κύριο κλειδί είναι το πεδίο user_id και υπάρχουν δύο επιπλέον πεδία με περιορισμό μοναδικότητας, τα afm και amka που αφορούν το ΑΦΜ και το ΑΜΚΑ του χρήστη.

5.4.2 Πίνακας roles

Ο πίνακας groups αφορά στον χαρακτηρισμό του επιπέδου πρόσβασης ενός χρήστη στα δεδομένα και τις λειτουργίες της εφαρμογής.

5.4.3 Πίνακας user_roles

Επειδή ένας χρήστης μπορεί να ανήκει σε πολλά group, ο πίνακας user_group χρησιμοποιείται για τη δημιουργία συσχέτισης "Πολλά προς Πολλά" μεταξύ των πινάκων users και groups.

5.4.4 Πίνακας graduations

Στον πίνακα graduations καταχωρούνται οι αποφοιτήσιες των χρηστών από το τμήμα ΜΠΗΣ.

5.4.5 Πίνακας *degrees*

Στον πίνακα *degrees* καταγράφονται πληροφορίες σχετικές με το είδος του κύκλου σπουδών που προσφέρεται από το τμήμα ΜΠΗΣ.

5.4.6 Πίνακας *other_studies*

Στον πίνακα *other_studies* καταγράφονται οι σπουδές του χρήστη σε τμήματα εκτός ΜΠΗΣ.

5.4.7 Πίνακας *working_histories*

Στον πίνακα *working_histories* καταγράφεται η επαγγελματική προϋπηρεσία των αποφοίτων.

5.4.8 Πίνακας *education_origins*

Ο πίνακας *education_origins* χρησιμοποιείται για την αποθήκευση των προαπαιτούμενων τίτλων σπουδών που είναι απαραίτητες για την εγγραφή σε ένα κύκλο σπουδών του ΜΠΗΣ (Πχ ΓΕΛ, ΕΠΑΛ, Προπτυχιακό (αν πρόκειται να παρακολουθήσει μεταπτυχιακό πρόγραμμα) κλπ).

5.4.9 Πίνακας *emails*

Στον πίνακα *emails* καταγράφονται οι λογαριασμοί αλληλογραφίας των χρηστών. Κάθε χρήστης μπορεί να έχει απεριόριστους λογαριασμούς αλληλογραφίας, ένας εκ των οποίων χαρακτηρίζεται ως κύριος.

5.4.10 Πίνακας *notifications*

Στον πίνακα *notifications* καταχωρούνται οι ειδοποιήσεις που στέλνονται περιοδικά στους χρήστες (πχ ειδοποίηση για ενεργοποίηση του λογαριασμού τους)

5.4.11 Πίνακας *countries*

Ο πίνακας *countries* για την αποθήκευση των των επίσημων ονομασιών των χωρών της Γης.

5.4.12 Πίνακας *scopus_codes*

Ο πίνακας *scopus_codes* χρησιμοποιείται για την αποθήκευση των θεματικών επιστημονικών περιοχών που καλύπτονται από το Scopus Database.

5.4.13 Πίνακας *password_resets*

Ο πίνακας `password_resets` χρησιμοποιείται κατά τη διαδικασία ανάκτησης κωδικού για την αποθήκευση του token επιβεβαίωσης.

5.4.14 Πίνακας *email_account_reset*

Ο πίνακας `email_account_reset` χρησιμοποιείται κατά τη διαδικασία ανάκτησης κωδικού για την αποθήκευση του token επιβεβαίωσης.

5.5 *Allumni Api*

Με ένα REST API τα στοιχεία δικτύου αντιμετωπίζονται σαν πόροι, στους οποίους ο χρήστης ζητά πρόσβαση μέσω των μεθόδων GET, PUT, POST, DELETE του HTTP πρωτοκόλλου, υλοποιώντας με αυτές τις λειτουργίες CRUD (Create, Read, Update, Delete).

- GET: Η μέθοδος αυτή χρησιμοποιείται για την ανάγνωση της τιμής ενός πόρου
- PUT: Η μέθοδος αυτή χρησιμοποιείται για την ενημέρωση υφιστάμενου πόρου
- POST: Η μέθοδος αυτή χρησιμοποιείται για την καταχώριση ενός νέου πόρου
- DELETE: Η μέθοδος αυτή χρησιμοποιείται για τη διαγραφή ενός πόρου

Στο Laravel ανάλογα με το HTTP request δρομολογείται η πρόσβαση στην αντίστοιχη μέθοδο κάποιου controller. Όταν ο controller δημιουργείται με το flag `-api` τότε υλοποιεί πέντε μεθόδους και οποιοδήποτε άλλο αίτημα δεν γίνεται δεκτό:

- `index()` που απαντά σε αίτημα GET χωρίς προσδιοριστικό πόρου στο URL και επιστρέφει όλους τους πόρους μιας κατηγορίας. Πχ η κλήση `http://mywebsite/api/degrees` επιστρέφει όλες τις εγγραφές του πίνακα `degrees`
- `show()` που απαντά σε αίτημα GET με προσδιοριστικό πόρου στο URL τον πόρο που θέλουμε να ανακτηθεί. Πχ η κλήση `http://mywebsite/api/degrees/{degree_id}` επιστρέφει τα στοιχεία της εγγραφής `{degree_id}` του πίνακα `degrees`
- `store()` που απαντά σε αίτημα POST χωρίς προσδιοριστικό πόρου στο URL. Η μέθοδος αυτή δημιουργεί έναν νέο πόρο τα στοιχεία του οποίου θα πρέπει να περιγραφούν στο `body` του αιτήματος. Πχ η κλήση `http://mywebsite/api/degrees` δημιουργεί ένα νέο τμήμα. Τα στοιχεία του υπό δημιουργία τμήματος περιγράφονται στο `body` του αιτήματος
- `update()` που απαντά σε αίτημα PUT με προσδιοριστικό πόρου στο URL τον πόρο που θέλουμε να ενημερωθεί. Πχ η κλήση `http://mywebsite/api/degrees/{degree_id}`

ενημερώνει την εγγραφή {degree_id} του πίνακα degrees. Τα πεδία που θα ενημερωθούν και οι νέες τιμές περιγράφονται στο body του αιτήματος

- destroy() που απαντά σε αίτημα DELETE με προσδιοριστικό πόρου στο URL τον πόρο που θέλουμε να διαγραφεί. Πχ η κλήση http://mywebsite/api/degrees/{degree_id} διαγράφει την εγγραφή {degree_id} του πίνακα degrees.

5.5.1 Κλήσεις API

Το API μεταξύ των άλλων απαντάει στις παρακάτω κλήσεις:

Πίνακας	Μέθοδος	URL
Degrees	DELETE	http://mywebsite/api/degrees/{degree_id}
Degrees	GET	http://mywebsite/api/degrees
Degrees	GET	http://mywebsite/api/degrees/{degree_id}
Degrees	POST	http://mywebsite/api/degrees
Degrees	PUT	http://mywebsite/api/degrees/{degree_id}
Emails	DELETE	http://mywebsite/api/degrees/{degree_id}/emails/{email-to-delete}
Emails	GET	http://mywebsite/api/degrees/{degree_id}/emails
Emails	GET	http://mywebsite/api/degrees/{degree_id}/emails/{user_id}
Emails	POST	http://mywebsite/api/degrees/{degree_id}/emails
Emails	PUT	http://mywebsite/api/degrees/{degree_id}/emails/{email-to-update}
Graduations	DELETE	http://mywebsite/api/degrees/{degree_id}/graduations/{user_id}
Graduations	GET	http://mywebsite/api/degrees/{degree_id}/graduations
Graduations	GET	http://mywebsite/api/graduations
Graduations	GET	http://mywebsite/api/degrees/{degree_id}/graduations/{user_id}
Graduations	GET	http://mywebsite/api/graduations/{user_id}
Graduations	POST	http://mywebsite/api/degrees/{degree_id}/graduations

Πίνακας	Μέθοδος	URL
Graduations	PUT	http://mywebsite/api/degrees/{degree_id}/graduations/{user_id}
Notifications	GET	http://mywebsite/api/degrees/{degree_id}/users/{user_id}/notifications
Notifications	POST	http://mywebsite/api/degrees/{degree_id}/users/{user_id}/notifications
Other_Studies	DELETE	http://mywebsite/api/users/{user_id}/other-studies/{other_studies_id}
Other_Studies	GET	http://mywebsite/api/degrees/{degree_id}/other-studies
Other_Studies	GET	http://mywebsite/api/users/{user_id}/other-studies
Other_Studies	GET	http://mywebsite/api/users/{user_id}/other-studies/{other_studies_id}
Other_Studies	POST	http://mywebsite/api/users/{user_id}/other-studies
Other_Studies	PUT	http://mywebsite/api/users/{user_id}/other-studies/{other_studies_id}
UserRoles	DELETE	http://mywebsite/api/user-roles/{user_role_id}
UserRoles	GET	http://mywebsite/api/degrees/{degree_id}/user-roles/
UserRoles	GET	http://mywebsite/api/user-roles/
UserRoles	GET	http://mywebsite/api/degrees/{degree_id}/user-roles/{user_id}
UserRoles	GET	http://mywebsite/api/user-roles/{user_id}
UserRoles	POST	http://mywebsite/api/degrees/{degree_id}/user-roles/
UserRoles	PUT	http://mywebsite/api/user-roles/{user_role_id}
Users	DELETE	http://mywebsite/api/users/{user_id}
Users	GET	http://mywebsite/api/users
Users	GET	http://mywebsite/api/degrees/{degree_id}/users
Users	GET	http://mywebsite/api/degrees/{degree_id}/roles/{role_id}/users
Users	GET	http://mywebsite/api/users/{user_id}
Users	GET	http://mywebsite/api/degrees/{degree_id}/users/{user_id}
Users	POST	http://mywebsite/api/degrees/{degree_id}/roles/{role_id}/users
Users	POST	http://mywebsite/api/degrees/{degree_id}/users

Πίνακας	Μέθοδος	URL
Users	PUT	http://mywebsite/api/degrees/{degree_id}/users/{user_id}
Users		http://mywebsite/api/users/{user_id}
Working_Histories	DELETE	http://mywebsite/api/users/{user_id}/working-histories/{working_histories_id}
Working_Histories	GET	http://mywebsite/api/degrees/{degree_id}/working-histories
Working_Histories	GET	http://mywebsite/api/users/{user_id}/working-histories
Working_Histories	GET	http://mywebsite/api/users/{user_id}/working-histories/{working_histories_id}
Working_Histories	POST	http://mywebsite/api/users/{user_id}/working-histories
Working_Histories	PUT	http://mywebsite/api/users/{user_id}/working-histories/{working_histories_id}

5.5.2 Παράδειγμα: Κλήση στον πίνακα Degrees

Ανάλογα με την ταυτότητα αυτού που πραγματοποιεί την κλήση το API επιστρέφει ή όχι αποτελέσματα. Έτσι στην περίπτωση του πίνακα Degrees έχουμε:

5.5.2.1 GET: *http://mywebsite/api/degrees*

- Διαχειριστής: Επιστρέφει τα στοιχεία όλων των τμημάτων
- Γραμματεία: Επιστρέφει τα στοιχεία όλων των τμημάτων
- Απόφοιτος: Δεν έχει εξουσιοδότηση

5.5.2.2 GET: *http://mywebsite/api/degrees/{degree_id}*

- Διαχειριστής: Επιστρέφει τα στοιχεία του τμήματος με {degree_id}
- Γραμματεία: Επιστρέφει τα στοιχεία του τμήματος με {degree_id}
- Απόφοιτος: Δεν έχει εξουσιοδότηση

5.5.2.3 POST: *http://mywebsite/api/degrees*

- Διαχειριστής: Δημιουργεί ένα νέο τμήμα. Τα στοιχεία της εγγραφής τα συμπεριλαμβάνει το body του αιτήματος
- Γραμματεία: Δεν έχει εξουσιοδότηση
- Απόφοιτος: Δεν έχει εξουσιοδότηση

5.5.2.4 PUT:http://apimywebsite/degrees/{degree_id}

- Διαχειριστής: Ενημερώνει τα στοιχεία του τμήματος με {degree_id}. Τα πεδία που θα ενημερωθούν και οι νέες τιμές τους καθορίζονται στο body του αιτήματος
- Γραμματεία: Δεν έχει εξουσιοδότηση
- Απόφοιτος: Δεν έχει εξουσιοδότηση

5.5.2.5 DELETE:http://apimywebsite/degrees/{degree_id}

- Διαχειριστής: Διαγράφει το τμήμα με {degree_id}
- Γραμματεία: Δεν έχει εξουσιοδότηση
- Απόφοιτος: Δεν έχει εξουσιοδότηση

5.5.3 Παράδειγμα: Κλήση στον πίνακα Graduations

Στην περίπτωση του πίνακα Graduations η κλήση γίνεται συνδυάζοντας το τμήμα με τον χρήστη:

5.5.3.1 GET:http://mywebsite/api/degrees/{degree_id}/graduations

- Διαχειριστής: Επιστρέφει όλες τις αποφοιτήσεις του τμήματος {degree_id}
- Γραμματεία: Επιστρέφει όλες τις αποφοιτήσεις του τμήματος {degree_id} εφόσον έχει το ρόλο της γραμματείας στο συγκεκριμένο τμήμα
- Απόφοιτος: Δεν έχει εξουσιοδότηση

5.5.3.2 GET:http://mywebsite/api/degrees/{degree_id}/graduations/{user_id}

- Διαχειριστής: Επιστρέφει την αποφοίτηση του χρήστη {user_id} από το τμήμα {degree_id}
- Γραμματεία: Επιστρέφει την αποφοίτηση του χρήστη {user_id} από το τμήμα {degree_id} εφόσον έχει το ρόλο της γραμματείας στο συγκεκριμένο τμήμα
- Απόφοιτος: Δεν έχει εξουσιοδότηση

5.5.3.3 POST:http://mywebsite/api/degrees/{degree_id}/graduations

- Διαχειριστής: Δημιουργεί μια νέα αποφοίτηση για το τμήμα {degree_id}. Τα στοιχεία της εγγραφής τα συμπεριλαμβάνει το body του αιτήματος
- Γραμματεία: Δημιουργεί μια νέα αποφοίτηση για το τμήμα {degree_id} εφόσον έχει το ρόλο της γραμματείας στο συγκεκριμένο τμήμα. Τα στοιχεία της εγγραφής τα συμπεριλαμβάνει το body του αιτήματος

- Απόφοιτος: Δεν έχει εξουσιοδότηση

5.5.3.4 PUT: `http://mywebsite/api/degrees/{degree_id}/graduations/{user_id}`

- Διαχειριστής: Ενημερώνει την αποφοίτηση του {user_id} που αφορά το τμήμα {degree_id}. Τα πεδία που θα ενημερωθούν και οι νέες τιμές τους καθορίζονται στο body του αιτήματος
- Γραμματεία: Ενημερώνει την αποφοίτηση του {user_id} που αφορά το τμήμα {degree_id} εφόσον έχει το ρόλο της γραμματείας στο συγκεκριμένο τμήμα. Τα πεδία που θα ενημερωθούν και οι νέες τιμές τους καθορίζονται στο body του αιτήματος
- Απόφοιτος: Δεν έχει εξουσιοδότηση

5.5.3.5 DELETE: `http://mywebsite/api/degrees/{degree_id}/graduations/{user_id}`

- Διαχειριστής: Διαγράφει την αποφοίτηση του {user_id} που αφορά το τμήμα {degree_id}
- Γραμματεία: Διαγράφει την αποφοίτηση του {user_id} που αφορά το τμήμα {degree_id} εφόσον έχει το ρόλο της γραμματείας στο συγκεκριμένο τμήμα
- Απόφοιτος: Δεν έχει εξουσιοδότηση

5.5.4 Κλήσεις με παραμέτρους στο URL

Στο URL της μεθόδου GET μπορούμε να συμπεριλάβουμε παραμέτρους που θα λειτουργήσουν σαν φίλτρα επί του συνόλου των επιστρεφόμενων αποτελεσμάτων. Πχ αν θέλουμε τα τμήματα που η διδασκαλία γίνεται στα Ελληνικά (μπορεί να υπάρχουν και Αγγλόφωνα τμήματα) το URL θα είναι:

`http://mywebsite/api/degrees?degreeLanguage[eq]=Ελληνικά`

Θα λάβουμε απάντηση της μορφής:

```
{
  "current_page": 1,
  "data": [
    {
      "degree_id": "*****",
      "degree_title": "5-ΕΤΕΣ ΠΠΣ ΤΜΗΜΑΤΟΣ",
      "degree_type": "Προπτυχιακό ΠΕ",
      "degree_language": "Ελληνικά",
      "from_year": "*****",
      "to_year": "*****",
      "min_duration": "*****",
      "created_at": "*****",
      "updated_at": "*****"
    }
  ],
}
```

```

{
  "degree_id": *****,
  "degree_title": "Π.Μ.Σ. ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ",
  "degree_type": "Μεταπτυχιακό",
  "degree_language": "Ελληνικά",
  "from_year": *****,
  "to_year": *****,
  "min_duration": *****,
  "created_at": *****,
  "updated_at": *****
},
{
  "degree_id": *****,
  "degree_title": "Π.Μ.Σ. ΕΦΑΡΜΟΣΜΕΝΑ ΗΛΕΚΤΡΟΝΙΚΑ ΣΥΣΤΗΜΑΤΑ",
  "degree_type": "Μεταπτυχιακό",
  "degree_language": "Ελληνικά",
  "from_year": *****,
  "to_year": *****,
  "min_duration": *****,
  "created_at": *****,
  "updated_at": *****
}
],
"first_page_url": "http://mywebsite/api/degrees?page=1",
"from": 1,
"last_page": 1,
"last_page_url": "http://mywebsite/api/degrees?page=1",
"links": [
  {
    "url": null,
    "label": "&laquo; Previous",
    "active": false
  },
  {
    "url": "http://mywebsite/api/degrees?page=1",
    "label": "1",
    "active": true
  },
  {
    "url": null,
    "label": "Next &raquo;",
    "active": false
  }
],
"next_page_url": null,
"path": "http://mywebsite/api/degrees",
"per_page": 10,
"prev_page_url": null,
"to": 3,
"total": 3
}

```

6

Τεχνολογίες που

χρησιμοποιήθηκαν

6.1 Λογισμικό και υπολογιστική υποδομή

6.1.1 Εγκατάσταση Εφαρμογής

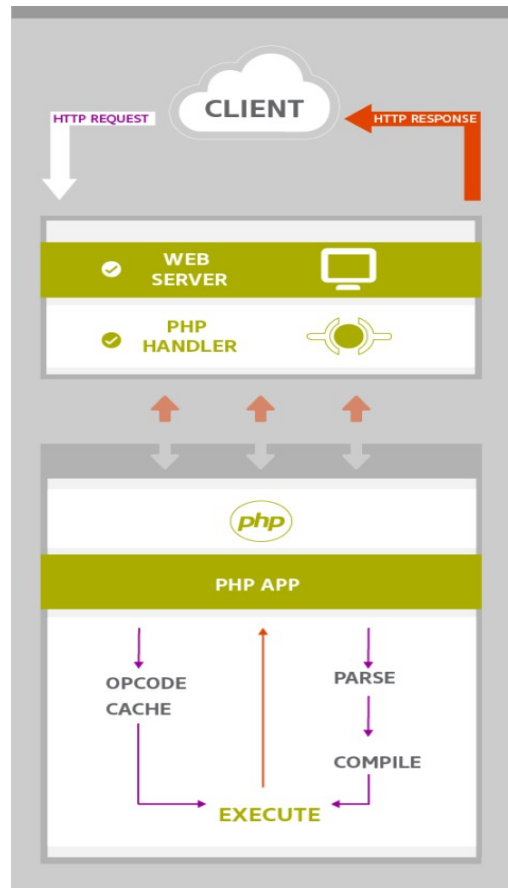
Η έκδοση Alumni v.1 εγκαταστάθηκε σε εξυπηρετητή (web server) του τμήματος ΜΠΗΣ που τρέχει σε λειτουργικό σύστημα Linux και διανομή Debian 11. Η πρόσβαση σε αυτήν γίνεται μέσω της διεύθυνσης <https://alumni.iee.ihu.gr>. Ο web server που τρέχει είναι ο Apache HTTP Server Version 2.4. Η συντήρηση του web server και τα αντίγραφα ασφαλείας είναι στην ευθύνη του τμήματος ΜΠΗΣ.

6.1.2 PHP

Για το BackEnd της εφαρμογής χρησιμοποιείται η PHP 7.4. Ο Apache υποστηρίζει αρκετούς διαφορετικούς χειριστές/διερμηνευτές της PHP (PHP Handlers) μεταξύ των οποίων τους[12]:

- CGI (Common gateway interface): Πρόκειται για τον παλαιότερο τρόπο εκτέλεσης εφαρμογών σε έναν web server και πλέον χρησιμοποιείται σπάνια. Αυτός ο handler για κάθε αίτημα εκκινεί μια νέα διεργασία CGI. Δηλαδή κάθε φορά να φορτώνεται ο διερμηνευτής της PHP με τα πρόσθετα του καθώς και τα σχετικά αρχεία παραμετροποίησης[13].

- DSO (Dynamic Shared Object) ή mod_php: Πρόκειται για τον προκαθορισμένο και έναν από τους γρηγορότερους PHP handlers. Η PHP λειτουργεί σαν μέρος του Apache και τη διερμηνεία του κώδικα την αναλαμβάνει πλέον ο web server μέσω του module mod_php. Πλεονέκτημα του είναι οι χαμηλές απαιτήσεις σε μνήμη. Θεωρείται όμως ευάλωτος όσον αφορά την ασφάλεια επειδή εκτελεί τον κώδικα σαν χρήστη nobody[12].
- FCGI: Είναι μια βελτιωμένη έκδοση του CGI handler με προσανατολισμό την βελτίωση της απόδοσης. Κρατάει τον κώδικα PHP στη μνήμη (caching) και έτσι δεν χρειάζεται να εκκινεί μια νέα διεργασία PHP σε κάθε αίτημα. Το μειονέκτημα είναι ότι καταναλώνει περισσότερη μνήμη.
- PHP-FPM (PHP FastCGI Process Manager): Πρόκειται για έναν handler εναλλακτικό του FastCGI. Επιτρέπει σε ένα ιστοχώρο να ανταποκρίνεται σε μεγάλο φόρτο διατηρώντας μια δεξαμενή από ανταποκριτές (workers) που ακούν σε διαφορετικές πόρτες. Έχει όμως υψηλές απαιτήσεις σε μνήμη. Ο web server που είναι εγκατεστημένη η εφαρμογή χρησιμοποιεί αυτόν τον PHP handler.



Εικόνα 38: PHP Execution[13]

6.1.3 Composer

Πέρα από τις προσφερόμενες βιβλιοθήκες από τους προγραμματιστές της PHP, υπάρχουν πολλές εξωτερικές βιβλιοθήκες που μπορούν να ενσωματωθούν σε μια εφαρμογή. Αυτό μπορεί να γίνει είτε χειροκίνητα είτε με εργαλεία προσθήκης και διαχείρισης βιβλιοθηκών. Ένα από αυτά είναι ο Composer.

Ο Composer είναι ένας διαχειριστής βιβλιοθηκών PHP σε επίπεδο εφαρμογής και όχι λειτουργικού συστήματος. Επιτρέπει στον προγραμματιστή να δηλώσει και εγκαταστήσει αυτόματα τις βιβλιοθήκες που χρησιμοποιεί στην εφαρμογή του, κάνοντας παράλληλα αυτόματο έλεγχο των αλληλεξαρτήσεων (dependency management)[14].

Το βασικό του αποθετήριο είναι το Packagist.org[15]. Οι χρήστες του μπορούν να ανεβάσουν τις βιβλιοθήκες που επιθυμούν να διαμοιράσουν μέσω αυτού. Οι περισσότερες διαμοιραζόμενες βιβλιοθήκες παρέχουν σύνδεσμο για το Packagist.org ή την σύνταξη της απαιτούμενη εντολή του Composer ώστε να αναλάβει αυτός την εγκατάστασή τους.

6.1.3.1 Εγκατάσταση Composer

Ο Composer μπορεί να εγκατασταθεί είτε καθολικά σαν εκτελέσιμο που βρίσκεται στο PATH (πχ σε Linux Mint στον φάκελλο /usr/local/bin) είτε σαν τοπικό αρχείο της εφαρμογής οπότε στο ριζικό φάκελλο της εφαρμογής αποθηκεύεται το αρχείο composer.phar.

- Στην περίπτωση της καθολικής εγκατάστασης η κλήση του γίνεται από οποιονδήποτε σημείο με την εντολή composer.
- Όταν η εγκατάσταση είναι τοπική για να γίνει κλήση θα πρέπει να βρισκόμαστε στο ριζικό φάκελλο της εφαρμογής και να χρησιμοποιήσουμε την εντολή phar composer.phar.

Η διαδικασία εγκατάστασης του σε περιβάλλον Linux Mint 19.3 και εκτέλεσης χωρίς να χρειάζεται να είμαστε root, έχει ως εξής:

- Λήψη και έλεγχος ακεραιότητας

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');";

php -r "if (hash_file('sha384', 'composer-setup.php') ===
'906a84df04cea2aa72f40b5f787e49f22d4c2f19492ac310e8c5a5b96ac8b6
4115ac402c8cd292b8a03482574915d1a8') { echo 'Installer
verified'; } else { echo 'Installer corrupt'; unlink('composer-
setup.php'); } echo PHP_EOL;";
```

- Για καθολική εγκατάσταση ώστε να μπορεί να γίνει κλήση από οποιοδήποτε σημείο με την εντολή composer

```
sudo php composer-setup.php --install-dir=/usr/local/bin -
filename=composer;
```

- Απόκτηση δικαιώματος εγκατάστασης πακέτων από τον τρέχον χρήστη

```
sudo chown -R $USER ~/.config/composer;
```

- Προσθήκη του φακέλλου vendor/bin του composer στο PATH (βρίσκουμε πρώτα που είναι το global directory του Composer με την εντολή composer global about)

```
echo 'export PATH="$PATH:$HOME/.config/composer/vendor/bin"' >>
~/.bashrc;
```

- Αλλαγή δικαιωμάτων στον φάκελλο που είναι τα αρχεία ρυθμίσεων του Composer στο home directory (.config/composer)

```
sudo chown -R username:username composer/;
sudo chmod -R 755 composer/;
```

Για να εγκαταστήσουμε μια βιβλιοθήκη χρησιμοποιούμε την εντολή `composer require library_name`, όπου `library_name` είναι το όνομα της βιβλιοθήκης που επιθυμούμε. Για την εγκατάσταση των απαραίτητων βιβλιοθηκών της έκδοσης Alumni v.1 τρέξαμε τις παρακάτω εντολές:

```
composer require phpmailer/phpmailer;
composer require components/jquery;
composer require twbs/bootstrap;
composer require datatables/datatables;
composer require phpoffice/phpspreadsheet;
```

Με την εκτέλεση τους δημιουργείται ο φάκελλος `vendor` στον οποίο εγκαθίστανται οι βιβλιοθήκες, ο αρχείο `composer.json` στο οποίο καταγράφονται οι εγκατεστημένες βιβλιοθήκες και οι εκδόσεις τους, καθώς και το αρχείο `composer.lock` που εξασφαλίζει ότι δεν θα αλλάξει η έκδοση τους χωρίς τη δική μας παρέμβαση. Το περιεχόμενο του `composer.json` της εφαρμογής μας είναι:

```
{
    "require": {
        "twbs/bootstrap": "^5.2",
        "components/jquery": "^3.6",
        "phpmailer/phpmailer": "^6.6",
        "datatables/datatables": "^1.10",
        "phpoffice/phpspreadsheet": "^1.25"
    }
}
```

6.1.4 *PHPMailer*

Παρόλο που η PHP διαθέτει την ενσωματωμένη συνάρτηση `mail()` για τη δημιουργία και αποστολή μηνυμάτων αλληλογραφίας, χρησιμοποιούμε την βιβλιοθήκη `phpmailer` που είναι διαθέσιμη και στο GitHub στη διεύθυνση <https://github.com/PHPMailer/PHPMailer>[16].

Η έκδοση που χρησιμοποιείται είναι η `PHPMailer 6.6` και εγκαταστάθηκε μέσω του `Composer` με την εντολή:

```
composer require phpmailer/phpmailer;
```

Για να την χρησιμοποιήσουμε αρχικά προσθέτουμε τα namespaces των αντίστοιχων τάξεων:

```
require_once('../vendor/autoload.php');
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;
```

Δημιουργούμε ένα αντικείμενο:

```
$mail = new PHPMailer(true);
```

Καθορίζουμε τις παραμέτρους, όπως πχ αν χρησιμοποιώ SMTP, το CharSet, τον mailserver (Host), την θύρα, το user name και password κλπ. Μια πολύ ενδιαφέρουσα ιδιότητα είναι το SMTPDebug που αφορά το επίπεδο εμφάνισης μηνυμάτων λάθους. Σε παραγωγικό περιβάλλον θέτουμε τιμή 0, ενώ η αναλυτικότερη περιγραφή σφαλμάτων γίνεται με τιμή 4.

Παρακάτω παραθέτουμε τις ρυθμίσεις για τον mail server που χρησιμοποιήθηκε:

```
$mail->SMTPDebug = 0;
$mail->isSMTP();
$mail->CharSet = "UTF-8";
$mail->Host = 'smtp.iee.ihu.gr';
$mail->Port = 25;
$mail->Username = "*****";
$mail->Password = "*****";
$mail->SMTPAuth = false;
$mail->SMTPOptions = array(
    'ssl' => array(
        'verify_peer' => false,
        'verify_peer_name' => false,
        'allow_self_signed' => true
    )
);
/*Αποστολέας*/
$mail->setFrom('*****', '*****');
/* Παραλήπτης */
$mail->addAddress($mail_address, '');
/* Θέμα μηνύματος */
$mail->Subject = $subject;
/* Σώμα μηνύματος */
$mail->IsHTML(true);
$mail->Body = $message;
/*Αποστολή μηνύματος*/
$mail->Send();
```

6.1.5 HTML

Η HTML (ισοδύναμα: HyperText) Markup Language είναι μια γλώσσα σήμανσης που χρησιμοποιούν τα προγράμματα περιήγησης Ιστού για να ερμηνεύσουν και να συνθέσουν κείμενο, εικόνες και άλλο υλικό σε οπτικές ή ακουστικές ιστοσελίδες. Η HTML βασίζεται στη χρήση υπερσυνδέσμων, ετικετών που οριοθετούν τα στοιχεία της και ιδιοτήτων που χρησιμοποιούνται για τον προσδιορισμό της συμπεριφοράς των στοιχείων που περιλαμβάνει η σελίδα.

Μια σελίδα HTML χωρίζεται σε δύο περιοχές, στο <header> όπου περιγράφονται τα μεταδεδομένα και κλήσεις σε αρχεία ενεργειών (javascript) και μορφοποίησης (css) και το <body> όπου γράφουμε τον κώδικα για την εμφάνιση της σελίδας.

Οι διάφοροι περιηγητές υποστηρίζουν το βασικό πυρήνα των στοιχείων της HTML αλλά όχι το σύνολο των χαρακτηριστικών της. Από το 2008 η τρέχουσα έκδοση είναι η HTML5.

6.1.6 CSS

Ο κάθε περιηγητής έχει την δική του εσωτερική λογική για την εμφάνιση των στοιχείων της ιστοσελίδας. Για την αλλαγή της εμφάνισης χρησιμοποιούνται εντολές μορφοποίησης που ορίζονται με την ετικέτα <style> ή αρχεία css.

Η CSS (Cascading Style Sheet) είναι μια απλή γλώσσα σχεδιασμού που χειρίζεται την εμφάνιση της ιστοσελίδας. Με τα CSS, υπάρχει η δυνατότητα να ελέγχονται τα χρώματα, οι γραμματοσειρές, η θέση των αντικειμένων, κλπ

6.1.7 Javascript

Η Javascript είναι μια scripting γλώσσα προγραμματισμού που μπορεί να χρησιμοποιηθεί τόσο στην πλευρά του πελάτη (FrontEnd) για να προσθέσει διαλειτουργικότητα στην ιστοσελίδα, όσο και για προγραμματισμό στους εξυπηρετητές (BackEnd).

Δημιουργήθηκε το 1995 από τον Brendan Eich[17] της εταιρείας Netscape με την επωνυμία Mocha[18]. Αργότερα μετονομάστηκε σε LiveScript και τελικά σε JavaScript, κυρίως επειδή η ανάπτυξή της επηρεάστηκε περισσότερο από τη γλώσσα προγραμματισμού Java. Στην εφαρμογή μας την χρησιμοποιούμε μόνο για τον προγραμματισμό σε επίπεδο FrontEnd.

6.1.8 JQuery

Η JQuery είναι μια βιβλιοθήκη ανοικτού κώδικα της JavaScript που επινοήθηκε από τον John Resign και λανσαρίστηκε το Ιανουάριο του 2006 στο BarCamp NYC[19]. Υποστηρίζεται από πλήθος περιηγητών και το βασικό της πλεονέκτημα είναι η απλή σύνταξη των εντολών για προγραμματισμό σε FrontEnd. Επιτρέπει την εύκολη τροποποίηση των στοιχείων του DOM, τον χειρισμό γεγονότων, τον χειρισμό του Css, η σύνταξη των κλήσεων Ajax είναι απλούστερη, και έχει ενσωματωμένη βιβλιοθήκη για animation.

Στην εφαρμογή μας χρησιμοποιείται για τον προγραμματισμό του FrontEnd και χρησιμοποιούμε την έκδοση JQuery 3.6. Εγκαταστάθηκε μέσω του Composer με την εντολή:

```
composer require components/jquery;
```

Στη JQuery βασίζονται πολλές άλλες βιβλιοθήκες JavaScript, μεταξύ των οποίων η Datatables και η Bootstrap οι οποίες χρησιμοποιούνται από την εφαρμογή.

6.1.9 Datatables

Το Datatables[20] είναι μια βιβλιοθήκη που βασίζεται στο JQuery, προσφέρει διάφορα στυλ μορφοποίησης των πινάκων και έχει ενσωματωμένες λειτουργίες σελιδοποίησης, ταξινόμησης και αναζήτησης.

Εγκαταστάθηκε η έκδοση datatables 1.10 μέσω του Composer με την εντολή:

```
composer require datatables/datatables;
```

Η βιβλιοθήκη Datatables μπορεί να χρησιμοποιηθεί με δύο λογικές όσον αφορά την επεξεργασία των δεδομένων:

- Client side processing
- Server side processing.

6.1.9.1 Client side

Όλα τα δεδομένα επιστρέφουν από τον εξυπηρετητή στον σταθμό εργασίας και η οποιαδήποτε επεξεργασία γίνεται από αυτόν. Είναι κατάλληλο για μικρό αριθμό δεδομένων γιατί δεν απασχολεί τον εξυπηρετητή.

Πχ αν έχουμε δημιουργήσει έναν πίνακα με id="example", αρκεί στη JavaScript να γράψουμε κάτι σαν το παρακάτω:

```
$(document).ready(function () {  
  
    /* Datatable initialization */  
    DataTable = $('#example').DataTable(  
    {  
        columnDefs: [  
            {targets: [1], visible: false},  
            {targets: [4], sortable: false, searchable: false},  
            {targets: [5], sortable: false, searchable: false}  
        ],  
        "order": [[0, "asc"]],  
    }  
    );  
  
});
```

Στο παραπάνω παράδειγμα ορίζουμε ότι η 2η στήλη δεν θα είναι ορατή, ότι η 5η και η 6η στήλη δεν θα ταξινομούνται ή θα υπόκεινται σε αναζήτηση περιεχομένου και ότι η τα δεδομένα του πίνακα θα ταξινομηθούν με κλειδί την πρώτη στήλη.

6.1.9.2 Server side

Ο εξυπηρετητής επιστρέφει ένα υποσύνολο των δεδομένων (όσα έχουμε ορίσει ότι θέλουμε να εμφανίζονται) και με κάθε ενέργεια αναζήτησης, ταξινόμησης, αιτήματος για την επόμενη/προηγούμενη σελίδα γίνεται κλήση για τα δεδομένα στον εξυπηρετητή. Το πλεονέκτημα είναι ο μικρός αριθμός δεδομένων που κάθε φορά ανταλλάσσει ο εξυπηρετητής με τον σταθμό εργασίας. Το μειονέκτημα είναι ότι η επεξεργασία γίνεται από τον εξυπηρετητή και ότι έχουμε αλληπαλλήλες κλήσεις. Παρακάτω παραθέτουμε τμήμα κώδικα της εφαρμογής όπου κάνουμε χρήση της server side επιλογής. Κύριο χαρακτηριστικό είναι ότι γίνεται μια κλήση Ajax:

```

var table = $('#user_data').DataTable({
    "processing": true,
    "serverSide": true,
    "ajax": "path_to_ajax_script.php",

    "columns": [
        {
            'targets': 0,
            'searchable': false,
            "class": " dt-body-center",
            "orderable": false,
            "data": null,
            "defaultContent": "",
            'className': 'dt-body-center',
            'render': function (data, type, full, meta) {
                return '<input type="checkbox">';
            }
        },

        {"data": "row_no", visible: false},
        {"data": "surname"},
        {"data": "name"},
        {"data": "studies"},
        {"data": "email_1"},
        {"data": "email_2"},
        {"data": "last_notice"},
    ],

    "order": [[1, 'asc']],

    'rowCallback': function (row, data, dataIndex) {
        // Get row ID
        var row_no = data['row_no'];

        rows_selected.forEach(function (e) {
            if (row_no === e.row_no) {
                $(row).find('input[type="checkbox"]')
                    .prop('checked', true);
                $(row).addClass('selected');
            }
        });
    }
});

```

Στο αρχείο php που καλεί η κλήση Ajax πρέπει να γράψουμε τον κώδικα που θα επιστρέψει τα δεδομένα με τη μορφή json. Ο δημιουργός της βιβλιοθήκης έχει αναπτύξει μια τάξη που εκτελεί τα ερωτήματα στη βάση και επιστρέφει τα αποτελέσματα. Πρόκειται για την τάξη SSP και ο κώδικας είναι στο αρχείο ssp.class.php. Η τάξη αυτή προβλέπει δύο μεθόδους για την εκτέλεση ερωτημάτων την simple και την complex που επιτρέπει πιο σύνθετα ερωτήματα. Σε γενικές γραμμές αυτό που αναμένει και οι δύο στην είσοδο είναι ένας και μόνο ένας πίνακας και όχι ένα join πινάκων. Αυτό ξεπερνιέται με τη χρήση όψεων ή με μια τροποποίηση που προτείνει ο Md. Emran ul hadi[21].

Ένα παράδειγμα του php αρχείου στο οποίο γίνεται η κλήση Ajax είναι[22]:

```
<?php
```

```

// DB table to use
$table = 'datatables_demo';

// Table's primary key
$primaryKey = 'id';

// Array of database columns which should be read and sent back to
DataTables.
// The `db` parameter represents the column name in the database,
while the `dt`
// parameter represents the DataTables column identifier. In this
case simple
// indexes
$columns = array(
    array( 'db' => 'first_name', 'dt' => 0 ),
    array( 'db' => 'last_name', 'dt' => 1 ),
    array( 'db' => 'position', 'dt' => 2 ),
    array( 'db' => 'office', 'dt' => 3 ),
);

// SQL server connection information
$sql_details = array(
    'user' => '',
    'pass' => '',
    'db' => '',
    'host' => ''
);

require( 'ssp.class.php' );

echo json_encode(
    SSP::simple( $_GET, $sql_details, $table, $primaryKey, $columns
)
);

```

6.1.10 Bootstrap

Πρόκειται για μια FrontEnd εργαλειοθήκη με έτοιμα css, δικά της πρόσθετα javascript. Αρχικά δημιουργήθηκε από ένα σχεδιαστή και έναν προγραμματιστή του Twitter στα μέσα του 2010 και ήταν γνωστό σαν Twitter Blueprint, ενώ το 2011 δημοσιεύτηκε ως λογισμικό ανοικτού κώδικα[23]. Στην εφαρμογή εγκαταστάθηκε η έκδοση Bootstrap 5.2 μέσω του composer με την εντολή:

```
composer require twbs/bootstrap;
```

Το Bootstrap παρέχει δικές του μορφοποιήσεις για πληθώρα αντικειμένων του DOM. Πχ μπορεί να οριστεί το χρώμα του κειμένου με χρήση της κλάσης "text-primary" που αντιστοιχεί σε γαλάζιο χρώμα και για την ακρίβεια στο rgb(13, 110, 253).

Ένα από τα ενδιαφέροντα αντικείμενα που προσφέρει το Bootstrap είναι τα modals, δηλαδή παραμετροποιήσιμα αναδυόμενα παράθυρα. Χρησιμοποιήθηκαν αντί για του κλασσικού alert της javascript καθώς και σε πολλές φόρμες καταχώρισης στοιχείων (πχ στη login φόρμα).

6.1.11 *PhpSpreadsheet*

Η PhpSpreadsheet είναι μια βιβλιοθήκη PHP για την επεξεργασία λογιστικών φύλλων, αρχείων csv, ακόμη και pdf (σε αυτήν την περίπτωση χρειάζεται επιπλέον μια από τις TCPDF, Dompdf ή mPDF που είναι σχεδιασμένες για την παραγωγή pdf)[24]. Η έκδοση που χρησιμοποιείται είναι η PhpSpreadsheet 1.25 και εγκαταστάθηκε μέσω του Composer με την εντολή:

```
composer require phpooffice/phpspreadsheet;
```

Για να την χρησιμοποιήσουμε αρχικά προσθέτουμε τα namespaces των αντίστοιχων τάξεων:

```
require_once('../vendor/autoload.php');
use PhpOffice\Ph\Spreadsheet;
use PhpOffice\PhpSpreadsheet\Worksheet;
use PhpOffice\PhpSpreadsheet\IOFactory;
use PhpOffice\PhpSpreadsheet\Reader\IReadFilter;
```

Κατόπιν πρέπει να δημιουργήσουμε έναν αντικείμενο αναγνώστη για το αρχείο και αναθέσουμε το περιεχόμενο του αρχείου.

```
$reader = IOFactory::createReaderForFile($filename);
$reader->setReadDataOnly(true);
$spreadsheet = $reader->load($filename);
```

Εφαρμόζοντας κλάσεις-φίλτρα μπορούμε να πάρουμε μέρος του αρχείου (πχ την πρώτη γραμμή). Παράδειγμα τέτοιας κλάσης είναι η κλάση ChunkReadFilter που ακολουθεί παρακάτω:

```
/** Define a Read Filter class implementing IReadFilter */
class ChunkReadFilter implements IReadFilter
{
    private $startRow = 0;

    private $endRow = 0;

    /**
     * Set the list of rows that we want to read.
     *
     * @param mixed $startRow
     * @param mixed $chunkSize
     */
    public function setRows($startRow, $chunkSize): void
    {
        $this->startRow = $startRow;
        $this->endRow = $startRow + $chunkSize;
    }

    public function readCell($columnAddress, $row, $worksheetName = '')
    {
        // Only read the heading row, and the rows that are configured in
        $this->_startRow and $this->_endRow
    }
}
```

```

        if (($row == 1) || ($row >= $this->startRow && $row < $this->endRow)) {
            return true;
        }

        return false;
    }
}

```

και με την παρακάτω συνάρτηση επιστρέφουμε την πρώτη γραμμή:

```

function get_spreadsheet_headers($reader, $filename)
{
    $chunkFilter = new ChunkReadFilter();
    $chunkFilter->setRows(1, 0);
    return $reader->load($filename);
}

```

Υπάρχει πληθώρα άλλων μεθόδων που μας επιτρέπουν να επεξεργαστούμε ένα λογιστικό φύλλο.

6.1.12 MariaDB

Η MariaDB είναι ένα Σύστημα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων. Αναπτύχθηκε από τους προγραμματιστές της MySQL όταν αυτή αγοράστηκε από την Oracle, με σκοπό να παραμείνει εφαρμογή Ανοικτού Λογισμικού[25]. Η εγκατεστημένη έκδοση είναι η 10.5.15. Η διαχείριση της μπορεί να γίνει είτε μέσω γραμμής εντολών, είτε μέσω εργαλείων όπως η PhpMyAdmin και το MySQL Workbench.

Για τις ανάγκες της έκδοσης Alumni v1, πέρα από τους πίνακες δημιουργήσαμε στη βάση views για να χρησιμοποιηθούν σε συνδυασμό με την βιβλιοθήκη Datatable και stored procedures που αναλαμβάνουν την καταχώριση των νέων χρηστών, την ενημέρωση των στοιχείων τους και την αλλαγή κωδικού.

6.1.13 Laravel

Το Laravel[26] είναι ένα πλαίσιο ανάπτυξης κώδικα (framework) για την ανάπτυξη διαδικτυακών εφαρμογών σε γλώσσα PHP που βασίζεται σε ένα άλλο framework, το Symfony και ακολουθεί τη σχεδιαστική αρχιτεκτονική Model-View-Controller.

Ένα PHP Framework αποτελείται από ένα σύνολο βιβλιοθηκών και προσφέρει έτοιμες λειτουργίες, όπως εργαλεία ελέγχου ταυτότητας, εργαλεία δρομολόγησης, πρότυπα HTML κλπ. Αυτό επιτρέπει την ταχύτερη και ποιοτικότερη ανάπτυξη της εφαρμογής, αφού στηρίζεται στην επαναχρησιμοποίηση ήδη δοκιμασμένων στοιχείων.

Η εγκατάσταση και αρχικοποίηση μιας εφαρμογής που βασίζεται στο Laravel μπορεί να γίνει με τη χρήση του composer με την εκτέλεση της εντολής:

```
composer create-project laravel/laravel project_name
```

Με την εκτέλεση της εντολής δημιουργείται η δομή φακέλλων της εφαρμογής μεταξύ των οποίων περιλαμβάνονται οι φάκελλοι:

- `app`: Εδώ βρίσκεται ο πυρήνας του κώδικα της εφαρμογής, όπως οι `Controllers` και τα `Models`
- `database`: Χρησιμοποιείται για το χτίσιμο των πινάκων της βάσης δεδομένων και την εισαγωγή πραγματικών ή δοκιμαστικών δεδομένων σε αυτή.
- `public`: Περιέχει το αρχείο `index.php` που είναι το σημείο υποδοχής των αιτημάτων. Στο φάκελλο επίσης αποθηκεύονται στοιχεία όπως εικόνες, αρχεία JavaScript και CSS.
- `routes`: Φιλοξενεί τα αρχεία δρομολόγησης των αιτημάτων που γίνονται στην εφαρμογή. Όταν αφορά API οι κανόνες δρομολόγησης γράφονται στο αρχείο `api.php`

Η τρέχουσα έκδοση του Laravel είναι η 9.x που απαιτεί PHP ≥ 8.0 , γι' αυτό για την ανάπτυξη της εφαρμογής μας χρησιμοποιήσαμε την έκδοση 8.x. που απαιτεί PHP ≥ 7.3 .

6.1.13.1 *Laravel Artisan*

Το `artisan` είναι η διεπαφή γραμμής εντολών (`command line interface`) που συμπεριλαμβάνει το Laravel . Καλείται από το ριζικό κατάλογο της εφαρμογής και παρέχει ένα σύνολο χρήσιμων κατά την ανάπτυξη της εφαρμογής εντολών.

6.1.13.2 *Eloquent*

Το Laravel συμπεριλαμβάνει το Eloquent που είναι ένα εργαλείο αντιστίχωσης αντικειμένων με σχεσιακούς πίνακες (`object-relational mapper - ORM`). Με τη χρήση του Eloquent κάθε πίνακας αντιστοιχεί σε ένα Model το οποίο αλληλεπιδρά μαζί του.

6.1.13.3 *Δημιουργία πινάκων με το Laravel*

Με το Laravel μπορούμε να δημιουργήσουμε τους πίνακες της εφαρμογής μας, να ορίσουμε τις μεταξύ τους συσχετίσεις και επίσης να προσθέσουμε σε αυτούς πειραματικά ή πραγματικά δεδομένα. Η περιγραφή των πινάκων γίνεται στα `migration` αρχεία, τα οποία βρίσκονται στον φάκελλο `database/migrations`. Για να δημιουργηθεί ένα αρχείο `migration` χρησιμοποιείται η εντολή `artisan make:migration` ε όρισμα το όνομα του επιθυμητού πίνακα μεταξύ των λέξεων `create` και `table`. Πχ για να δημιουργηθεί το `migration` αρχείο που περιγράφει τον πίνακα `countries` (η τυποποίηση της ονοματοδοσίας στο Laravel απαιτεί τα ονόματα των πινάκων να είναι στον πληθυντικό αριθμό) :

```
php artisan make:migration create_countries_table
```

Αφού γίνει η επεξεργασία των migration αρχείων, προκειμένου να δημιουργηθούν οι πίνακες στη βάση εκτελείται η εντολή:

```
php artisan migrate
```

6.1.13.4 Δημιουργία Controller, Model

Ένα αρχείο model δημιουργείται με την εντολή `artisan make:model` και αποθηκεύεται στο φάκελο `app/Models`. Πχ το μοντέλο που θα αλληλεπιδρά με τον πίνακα `countries` θα δημιουργηθεί με την εντολή

```
php artisan make:model Country
```

Τα ονοματα των μοντέλλων στο Laravel είναι στον ενικό αριθμό. Επίσης το Eloquent θεωρεί πως το κύριο κλειδί του πίνακα ονομάζεται `id`, οπότε αν θέλουμε να δώσουμε στη στήλη άλλη ονομασία πρέπει στο μοντέλο να αναθέσουμε το όνομα στη `protected $primaryKey` ιδιότητα. Οι controller δημιουργούνται με την εντολή `artisan make:controller` και στην περίπτωση που πρόκειται για API προσθέτουμε το διακόπτη `--api`. Πχ ένα controller που θα διαχειριστεί τα αιτήματα σε σχέση με τον πίνακα `countries` και θα έχει τις μεθόδους `index()`, `store()`, `show()`, `update()`, `destroy()` δημιουργείται με την εντολή:

```
php artisan make:controller CountriesController --api
```

6.1.13.5 Routes

Η πρόσβαση στους controller γίνεται μέσα από το αρχείο `routes/api.php`. Αν λοιπόν θέλουμε με το URI <http://mywebsite/countries> πρόσβαση στον `CountriesController`, τότε πρέπει να δηλώσουμε το route:

```
Route::resource('countries', CountriesController::class);
```

Ανάλογα με την HTTP μέθοδο και το URL το αίτημα θα δρομολογείται σε διαφορετική μέθοδο του controller σύμφωνα με τον παρακάτω πίνακα:

Μέθοδος HTTP	URL	Μέθοδος Controller
GET	http://mywebsite/countries	index
POST	http://mywebsite/countries	store
GET	http://mywebsite/countries/3	show
PUT	http://mywebsite/countries/3	update
DELETE	http://mywebsite/countries/3	destroy

7

Επίλογος

Παρακάτω παραθέτουμε τα πρώτα συμπεράσματα από τη λειτουργία της υλοποίησης Alumni v.1 και τα βήματα που πρέπει να ακολουθήσουν για την έκδοση Alumni v.2.

7.1 Σύνοψη και συμπεράσματα

Η έκδοση Alumni v.1 αποτελεί μια αρχική υλοποίηση και είναι ήδη σε λειτουργία. Αρχικά καταχωρήθηκαν 37 απόφοιτοι του προπτυχιακού κύκλου σπουδών του Τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων που ορκίστηκαν το Φθινόπωρο του 2022. Εκκρεμεί η καταχώριση 184 αποφοίτων παλαιότερων ετών των μεταπτυχιακών κύκλων και συγκεκριμένα 136 αποφοίτων του ΠΜΣ Ευφυείς Τεχνολογίες Διαδικτύου και 48 του ΠΜΣ Εφαρμοσμένα Ηλεκτρονικά Συστήματα στην οποία δεν προχωρήσαμε γιατί βρισκόμαστε στο στάδιο της συμπλήρωσης πεδίων πληροφορίας που δεν υπήρχαν καταχωρημένα.

Η έκδοση Alumni v.2 θα συνεχίσει να αναπτύσσεται και σύμφωνα με τον προγραμματισμό το FrontEnd θα υλοποιηθεί στο πλαίσιο επόμενης εργασίας.

8

Βιβλιογραφία

- [1] "ΜΟΔΙΠ," Μονάδα Διασφάλισης Ποιότητας ΔΠΠΑΕ. [Online]. Available: <https://www.ihu.gr/modip/>. [Accessed: 30-Dec-2022].
- [2] "ΟΜΕΑ," ΟΜΕΑ – Μονάδα Διασφάλισης Ποιότητας ΔΠΠΑΕ. [Online]. Available: <https://www.ihu.gr/modip/omea/>. [Accessed: 30-Dec-2022].
- [3] Κ. Πηγουνάκης, Εισαγωγή στις Βάσεις Δεδομένων ΙΙ. Ενότητα Μοντέλο Οντοτήτων Συσχετίσεων, [Online]. Available: https://opencourses.uoc.gr/courses/pluginfile.php/15370/mod_resource/content/4/3.Μοντέλο_Οντοτήτων_Συσχετίσεων_Κειμεν.pdf
- [4] G. C Everest, Basic data structure models explained with common example, Fifth Texas Conference on Computing Systems 1976, pages 39-46.
- [5] P. Dybka, Crow's Foot Notation. [Online]. Available: <https://vertabelo.com/blog/crow-s-foot-notation/>. [Accessed: 20-Dec-2022].
- [6] M. Jailia, A. Kumar, M. Agarwal and I. Sinha, "Behavior of MVC (Model View Controller) based Web Application developed in PHP and .NET framework," *2016 International Conference on ICT in Business Industry & Government (ICTBIG)*, Indore, India, 2016, pp. 1-5, doi: 10.1109/ICTBIG.2016.7892651.
- [7] Visual Paradigm, What is Model-View and Control?. [Online]. Available: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-model-view-control-mvc/>. [Accessed: 25-Noe-2022].

- [8] Υπουργείο Ψηφιακής Διακυβέρνησης, Δι@ύγεια - API Ανοιχτών Δεδομένων. [Online]. Available: <https://diavgeia.gov.gr/api/help>. [Accessed: 26-Noe-2022].
- [9] A. S. Gillis,, REST API (RESTful API). [Online]. Available: <https://www.techtarget.com/searcharchitecture/definition/RESTful-API>. [Accessed: 01-Dec-2022].
- [10] MySQL, MySQL WorkBench. [Online]. Available: <https://www.mysql.com/products/workbench/>. [Accessed: 10-Dec-2022].
- [11] iLovePhD, List of Subject Areas Covered By Scopus Database. [Online]. Available: <https://www.ilovephd.com/list-of-subject-areas-covered-by-scopus-database/>. [Accessed: 10-Dec-2022].
- [12] KnownHost, PHP Handlers and what they are. [Online]. Available: <https://www.knownhost.com/kb/php-handlers-and-what-they-are/>. [Accessed: 15-Dec-2022].
- [13] O. Habib, Get a handle on PHP Handlers, AppDynamics, 2015. [Online]. Available: <https://www.appdynamics.com/blog/engineering/get-handle-php-handlers/>. [Accessed: 15-Dec-2022].
- [14] Composer A Dependency Manager for PHP. [Online]. Available: <https://getcomposer.org/>. [Accessed: 10-Dec-2022].
- [15] Packgist, The PHP Package Repository. [Online]. Available: <https://packagist.org/>. [Accessed: 10-Dec-2022].
- [16] PHPMailer, PHPMailer – A full-featured email creation and transfer class for PHP. [Online]. Available: <https://github.com/PHPMailer/PHPMailer>. [Accessed: 12-Dec-2022].
- [17] K. Krippendorff, Brendan Eich: The Employee who Created JavaScript. [Online]. Available: <https://www.linkedin.com/pulse/brendan-eich-employee-who-created-javascript-kaihan>. [Accessed: 11-Dec-2022].
- [18] The A to Z of programming languages, The A-Z of Programming Languages: JavaScript. [Online]. Available: <https://a-z.readthedocs.io/en/latest/javascript.html>. [Accessed: 11-Dec-2022].
- [19] Geekforgeeks.org, Difference Between JavaScript and jQuery. [Online]. Available: <https://www.geeksforgeeks.org/difference-between-javascript-and-jquery/>. [Accessed: 08-Dec-2022].
- [20] Datatables.net, Add advanced interaction controls to your HTML tables the free & easy way. [Online]. Available: <https://datatables.net/>. [Accessed: 08-Dec-2022].
- [21] E. Hadi, Customized SSP Class For Datatables Library. [Online]. Available: <https://github.com/emran/ssp>. [Accessed: 15-Dec-2022].

- [22] Datatables.net, Server-side processing. [Online]. Available:
https://datatables.net/examples/server_side/simple.html. [Accessed: 15-Dec-2022].
- [23] Bootstrap, History. [Online]. Available:
<https://getbootstrap.com/docs/5.2/about/overview/#history>. [Accessed: 25-Noe-2022].
- [24] PhpSpreadsheet, Welcome to PhpSpreadsheet's documentation. [Online]. Available:
<https://phpspreadsheet.readthedocs.io/en/latest/>. [Accessed: 25-Noe-2022].
- [25] MariaDB, MariaDB Server. [Online]. Available: <https://mariadb.org/en/>. [Accessed: 25-Noe-2022].
- [26] Laravel, The PHP Framework for Web Artisans. [Online]. Available:
<https://laravel.com/>. [Accessed: 25-Dec-2022].

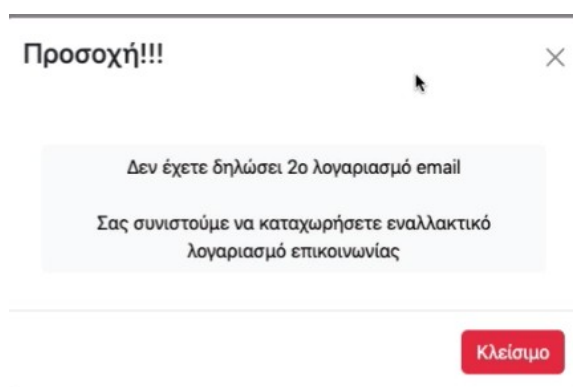
9

Παράρτημα

9.1 Οδηγίες χρήσης εφαρμογής *Alumni v.1*

Για τη διευκόλυνση των αποφοίτων έχουν συνταχθεί οι παρακάτω οδηγίες χρήσης της εφαρμογής:

1. Έχοντας λάβει το email-πρόσκληση με τα (αρχικά) username και password, κάνουμε login στην υπηρεσία (<https://alumni.iee.ihu.gr/omea>)
2. Εμφανίζεται το μήνυμα της παρακάτω εικόνας:



Εικόνα 39: Σύσταση για δήλωση 2ης διεύθυνσης email

Προς το παρόν, μπορούμε να αγνοήσουμε τη δήλωση δεύτερης διεύθυνσης email και προχωρούμε στην ενεργοποίηση του λογαριασμού μας (βήμα 3, παρακάτω).

3. Απαραίτητη ενέργεια αποτελεί η δήλωση νέου κωδικού πρόσβασης (password) και η ενεργοποίηση του λογαριασμού μας:

Αλλαγή Κωδικού

Νέος Κωδικός

Κωδικός

Επανάληψη Κωδικού

Επανάληψη Κωδικού

— Ο Κωδικός πρέπει:

1. Να έχει μήκος 8 - 15 χαρακτήρες
2. Να έχει τουλάχιστον ένα κεφαλαίο χαρακτήρα
3. Να έχει τουλάχιστον έναν πεζό χαρακτήρα
4. Να έχει τουλάχιστον έναν ειδικό χαρακτήρα !@#%&*?

Ενεργοποίηση Λογαριασμού

Εικόνα 40: Αλλαγή κωδικού και ενεργοποίηση του λογαριασμού

4. Έχοντας ενεργοποιήσει το λογαριασμό μας, μας ζητείται να κάνουμε εκ νέου login και το κάνουμε. Τώρα μπορούμε να δηλώσουμε δεύτερη διεύθυνση email την οποία και επιβεβαιώνουμε με χρήση κωδικού που αποστέλλεται στη συγκεκριμένη διεύθυνση. Σημειώνεται ότι, από τη στιγμή που δηλώνεται δεύτερη διεύθυνση email, ο χρήστης μπορεί να χρησιμοποιεί εξίσου την πρώτη ή τη τη δεύτερη διεύθυνση email ως username για πρόσβαση (login) στην υπηρεσία
5. Στη συνέχεια έχουμε τη δυνατότητα να εμπλουτίσουμε το περιεχόμενο του προσωπικού μας προφίλ με επιπλέον προσωπικά στοιχεία (Εικόνα 41), στοιχεία που αφορούν σε άλλες σπουδές μας (σε πανεπιστημιακό επίπεδο) στο τμήμα ΜΠΗΣ ή αλλού, καθώς επίσης και στοιχεία που αφορούν στην προσωπική μας επαγγελματική σταδιοδρομία (Εικόνα 42).

Προσωπικά Στοιχεία

Επίθετο	Όνομα	Πατρώνυμο	Ημέλια Γέννησης
ΔΕΡΒΟΣ	ΔΗΜΗΤΡΙΟΣ	ΠΑΤΡΩΝΥΜΟ	01/01/1960
ΑΦΜ	ΑΜΚΑ	ΑΔΤ/Διαβατήριο	
Διεύθυνση Κατοικίας	Σταθερό Τηλέφωνο	Κινητό Τηλέφωνο	
ΔΙΕΥΘΥΝΣΗ	14 Ψηφία	14 Ψηφία	
Home URL	LinkedIn		
Google Scholar	ORCID		
Ενημέρωση			

Εικόνα 41: Δήλωση προσωπικών στοιχείων

Σπουδές στο Τμήμα Μ.Π.Η.Σ.

ΑΕΜ	Τύπος Πτυχίου	Ημ/νια Εισαγωγής	Ημ/νια Αποφοίτησης	Βαθμός Πτυχίου
ΔΟΚΙΜΑΣΤΙΚΟ	ΔΙΔΑΚΤΟΡΙΚΟ	01/10/2022	22/10/2022	10.00

Άλλες Σπουδές

Από	Έως	Ακαδημαϊκό Ίδρυμα	Κατηγορία	Επίπεδο	Εξειδίκευση	Συνάφεια	Χώρα
-----	-----	-------------------	-----------	---------	-------------	----------	------

Επαγγελματική Σταδιοδρομία

Από	Έως	Εταιρεία/ Οργανισμός	Κατηγορία (πχ Πληροφορικής)	Ρόλος (πχ Προγραμματιστής)	Χώρα	Πόλη	Πρακτική*
-----	-----	-------------------------	--------------------------------	-------------------------------	------	------	-----------

— Πρακτική*

Ναι → Έκανε πρακτική άσκηση σε αυτόν τον οργανισμό και προσελήφθη

Όχι → Προσελήφθη χωρίς να κάνει πρακτική σε αυτόν τον οργανισμό

Εικόνα 42: Σπουδές και επαγγελματική σταδιοδρομία

6. Για τα (προσωπικά μας) δεδομένα του παραπάνω βήματος (43), μας δίνεται η δυνατότητα να δηλώσουμε εκείνα που επιθυμούμε να είναι ορατά από τους υπόλοιπους χρήστες/αποφοίτους (Εικόνα 43):

Επίπεδο Έκθεσης Προσωπικών Δεδομένων

Πλέον του Ονοματεπώνυμου, των τίτλων σπουδών και της ημερομηνίας αποφοίτησης από το ΔΙ.ΠΑ.Ε. επιτρέπεται την κοινοποίηση των παρακάτω:

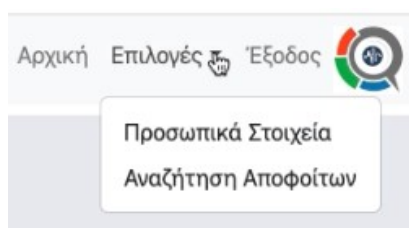
- Email
- Home Url
- LinkedIn
- Google Scholar
- ORCID
- Άλλες Σπουδές
- Επαγγελματική Σταδιοδρομία

Ενημέρωση

Εικόνα 43: Επιλογή δεδομένων προς δημοσιοποίηση

Στο βήμα που ακολουθεί (7) γίνεται απόλυτα κατανοητή η σημασία της δήλωσής μας στην Εικόνα 5.



7. Επιλέγοντας “Αναζήτηση Αποφοίτων” στο επάνω δεξιά άκρο της οθόνης (Εικόνα 44), μπορούμε να δούμε ή/και αναζητήσουμε άλλους αποφοίτους/χρήστες και (εάν το επιθυμούμε) να επικοινωνήσουμε μαζί τους (Εικόνα 45). Σε αυτό το σημείο, μπορούμε να δούμε τα προσωπικά δεδομένα που ο κάθε ένας απόφοιτος έχει αποφασίσει να δημοσιοποιεί για τον εαυτό του (βλ. βήμα 6, παραπάνω).



Εικόνα 44: Αναζήτηση Αποφοίτων

Αναζήτηση Αποφοίτων

Show 10 entries Search:

Επώνυμο	Όνομα	Τίτλος Πτυχίου	Αποφοίτηση	
TESTSURNAME	TESTNAME	5-ΕΤΕΣ ΠΤΥΧ ΤΜΗΜΑΤΟΣ	24/10/2022	
ΔΕΡΒΙΟΣ	ΔΗΜΗΤΡΙΟΣ	ΔΙΔΑΚΤΟΡΙΚΟ	22/10/2022	

Showing 1 to 2 of 2 entries Previous **1** Next

Εικόνα 45: Απόφοιτοι/χρήστες της υπηρεσίας

8. Στην περίπτωση που επιθυμούμε να επικοινωνήσουμε με έναν από τους αποφοίτους της Εικόνας 7, ακόμη και στην περίπτωση που αυτός/ή δεν δημοσιοποιεί της προσωπική του/ης διεύθυνση email, κάνουμε “κλικ” στο αντίστοιχο εικονίδιο με τον φάκελο στο δεξί άκρο της εικόνας (βλέπε Εικόνα 46):



Figure 46: Εικονίδια-φάκελοι για αποστολή email σε άλλον/η απόφοιτο

9. Για απορίες/βοήθεια, στείλτε email στη διεύθυνση: omea.iee.ihu@gmail.com γράφοντας “IEEAlumni” στο πεδίο “Θέμα/Subject” του email.