



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Ολοκληρωμένο Σύστημα Ανάλυσης στοιχείων  
Αξιολόγησης Πανεπιστημιακής εκπαιδευτικής  
διαδικασίας»

Του φοιτητή Τσελάνη Βασίλειου  
Αρ. Μητρώου: 2019173

Επιβλέπων  
Αντώνης Σιδηρόπουλος  
Αναπληρωτής Καθηγητής

Θεσσαλονίκη 25/5/2024

Τίτλος Δ.Ε. Ολοκληρωμένο Σύστημα Ανάλυσης στοιχείων Αξιολόγησης Πανεπιστημιακής  
εκπαιδευτικής διαδικασίας

Κωδικός Δ.Ε. 23356

Όνοματεπώνυμο φοιτητή/τών ΒΑΣΙΛΕΙΟΣ ΤΣΕΛΑΝΙ  
Όνοματεπώνυμο εισηγητή ΑΝΤΩΝΗΣ ΣΙΔΗΡΟΠΟΥΛΟΣ

Ημερομηνία ανάληψης Δ.Ε. 20-12-2023

Ημερομηνία περάτωσης Δ.Ε. 25-5-2024

*Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.*

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Τσελάνι Βασίλειου που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.



*«Αφιερώνεται στον Pasquale Celani»*



## Πρόλογος

Ο λόγος που επέλεξα το θέμα διπλωματικής εργασίας “Ολοκληρωμένο Σύστημα Ανάλυσης στοιχείων Αξιολόγησης Πανεπιστημιακής εκπαιδευτικής διαδικασίας” ήταν επειδή επιθυμούσα να εφαρμόσω στην πράξη όλες τις γνώσεις που αποκόμισα σχετικά με την ανάπτυξη λογισμικού κατά την διάρκεια των σπουδών μου. Επίσης θεώρησα ιδιαίτερα ενδιαφέρον το γεγονός ότι το πληροφοριακό σύστημα που ανέπτυξα θα χρησιμοποιείται από τους διδάσκοντες και θα προσφέρει ένα σημαντικό αντίκτυπο στην ακαδημαϊκή κοινότητα. Η ανάπτυξη του συγκεκριμένου λογισμικού ήταν μια προσωπική πρόκληση, επειδή έχει να κάνει με την επίλυση ενός προβλήματος που είναι αρκετά σύνθετο και δεν υπάρχει μεγάλο περιθώριο σφάλματος στα αποτελέσματα. Μέσω αυτού και υπό την καθοδήγηση από τον κ. Σιδηρόπουλο αποκόμισα πολλές και σημαντικές γνώσεις που θα μου φανούν ιδιαίτερα χρήσιμες στα μελλοντικά λογισμικά που θα αναπτύξω.

## Περίληψη

Κάθε εξάμηνο πραγματοποιείται διαδικασία αξιολογήσεις όλων των μαθημάτων τους τμήματος όσο το θεωρητικό μέρος τόσο και το εργαστηριακό. Οι αξιολογήσεις των φοιτητών αποθηκεύονται στην διαδικτυακή πλατφόρμα της ΜΟΔΠΠ. Η πληροφορία οργανώνεται σε xlsx αρχεία και οι διδάσκοντες του τμήματος μπορούν να κατεβάσουν τα αρχεία μέσω της διαδικτυακής εφαρμογής. Η δομή των αρχείων xlsx δεν ευνοεί τους διδάσκοντες για να μπορούν να υπολογίσουν σύνθετα στατιστικά στοιχεία. Το θέμα της συγκεκριμένης διπλωματικής εργασίας “Ολοκληρωμένο Σύστημα Ανάλυσης στοιχείων Αξιολόγησης Πανεπιστημιακής εκπαιδευτικής διαδικασίας” σχετίζεται με την ανάπτυξη ενός πληροφοριακού συστήματος σε διαδικτυακή μορφή, που θα μπορεί να δέχεται τα xlsx αρχεία που περιέχουν τις αξιολογήσεις των φοιτητών για διάφορα πανεπιστημιακά μαθήματα ενός συγκεκριμένου εξαμήνου ώστε τα δεδομένα να αποθηκευτούν στο σύστημα με την κατάλληλη μορφή. Το κέρδος είναι ότι οι διδάσκοντες θα μπορούν να εξάγουν τα επιθυμητά στατιστικά στοιχεία που επιθυμούν χωρίς να χρειάζεται να γράψουν δικά τους προγράμματα ή να αφιερώσουν πολύ χρόνο με τις ενσωματωμένες συναρτήσεις του xlsx.

Integrated Data Analysis System for the Evaluation of the University  
Educational Process

Vasili Celani

## **Abstract**

Each semester an evaluation procedure is carried out for all the courses of the department both the theoretical part and the laboratory part. The student evaluations are stored in the online platform of the MODIP. The information is organized in xlsx files and the lecturers of the department can download the files through the respective website. The structure of the xlsx files does not favor faculty to be able to calculate complex statistics. The topic of this thesis "Integrated System for Analysis of University Educational Process Evaluation Data" is related to the development of an information system in web-based format, which can accept xlsx files containing student evaluations for various university courses of a particular semester so that the data can be stored in the system in the appropriate format. The benefit is that instructors will be able to extract the desired statistics they want without having to write their own programs or spend a lot of time with the built-in xlsx functions.

## **Ευχαριστίες**

Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας θα ήθελα να εκφράσω την ευγνωμοσύνη μου στα άτομα που συνέβαλαν στην ολοκλήρωση αυτής. Πρώτα απ' όλα, είμαι υπόχρεος στον σύμβουλο της διατριβής μου, τον καθηγητή κ. Σιδηρόπουλο, του οποίου η καθοδήγηση και η αμέριστη υποστήριξη ήταν ανεκτίμητη καθ' όλη τη διάρκεια της μελέτης.

Επίσης θα ήθελα να ευχαριστήσω τους γονείς μου αλλά και όλους εκείνους που με στήριξαν με διάφορους τρόπους, είτε μέσω συζητήσεων, είτε μέσω ηθικής υποστήριξης, είτε απλά με το να είναι εκεί.

# Περιεχόμενα

Πρόλογος.....	vi
Περίληψη .....	vii
Abstract .....	ix
Ευχαριστίες .....	x
Περιεχόμενα .....	xi
Πίνακας Σχημάτων .....	xiv
Συντομογραφίες.....	xv
Κεφάλαιο 1ο: Εισαγωγή.....	1
1.1 Ιστορικό .....	1
1.1.1 ΜΟ.ΔΙ.Π .....	1
1.1.2 Δομή Ερωτηματολογίων .....	1
1.1.3 Διαδικασία συλλογής των δεδομένων.....	5
1.2 Καθορισμός του προβλήματος.....	6
1.3 Στόχοι .....	6
1.4 Επισκόπηση του συστήματος.....	7
Κεφάλαιο 2ο: Βιβλιογραφική ανασκόπηση.....	8
2.1 Εισαγωγή .....	8
2.2 Υπάρχουσες λύσεις.....	8
2.2.1 Tableau.....	8
2.2.2 IBM SPSS Statistics.....	8
2.2.3 Apache Superset .....	9
2.3 Τεχνολογίες.....	10
2.3.1 Εισαγωγή.....	10
2.3.2 Front-end Technologies .....	10
2.3.3 Back-end Technologies .....	13
Κεφάλαιο 3ο: Σχεδιασμός συστήματος.....	21
3.1 Εισαγωγή .....	21
3.2 Περιεχόμενο .....	21
3.3 Λειτουργικές Απαιτήσεις.....	21
3.4 Μη Λειτουργικές Απαιτήσεις.....	26
3.5 Περιορισμοί.....	26
3.6 Αρχές .....	27
3.6.1 Αρχιτεκτονική στρατηγική διαστρωμάτωσης .....	28

3.6.2	Όχι επιχειρησιακή λογική στην γραφική διεπαφή.....	28
3.6.3	Έξυπνη βάση δεδομένων.....	28
3.6.4	SOLID αρχές.....	29
3.7	Αρχιτεκτονική Λογισμικού.....	29
3.7.1	Context Diagram.....	30
3.7.2	Container diagram.....	31
3.7.3	Component diagram.....	32
3.7.4	Code-level diagram.....	33
3.8	Δεδομένα.....	33
3.8.1	Οντότητα: Πανεπιστήμιο .....	34
3.8.2	Οντότητα: Τμήμα Πανεπιστημίου .....	34
3.8.3	Οντότητα: Πρόγραμμα Σπουδών .....	35
3.8.4	Οντότητα: Μάθημα.....	35
3.8.5	Οντότητα: Εξάμηνο .....	36
3.8.6	Οντότητα: Διδάσκων.....	36
3.8.7	Οντότητα: Κατηγορία Ερώτησης .....	37
3.8.8	Οντότητα: Ερώτηση.....	37
3.8.9	Οντότητα: Ερωτηματολόγιο.....	38
3.8.10	Οντότητα: Ψήφος.....	39
3.9	Διάγραμμα ER.....	40
Κεφάλαιο 4ο:	Υλοποίηση συστήματος .....	42
4.1	Εισαγωγή .....	42
4.2	Αλγόριθμοι.....	42
4.2.1	Έλεγχος xlsx αρχείου.....	42
4.2.2	Υποβολή αρχείου.....	45
4.3	APIs.....	47
4.3.1	Έλεγχος ορθότητας XLSX αρχείου .....	49
4.3.2	Καταχώρηση XLSX αρχείου.....	49
4.3.3	{ "error": "ERROR MESSAGE" } Ανάκτηση καταχωρημένων εξαμήνων.....	50
4.3.4	Ανάκτηση ερωτήσεων.....	51
4.3.5	Ανάκτηση φίλτρων .....	51
4.3.6	Ανάκτηση μαθημάτων .....	52
4.3.7	Ανάκτηση αναλυτικών πληροφοριών μαθήματος .....	53
4.4	Σχήμα Βάσης Δεδομένων .....	54
Κεφάλαιο 5ο:	Οδηγίες χρήσης.....	62

5.1	Εισαγωγή .....	62
5.2	Φόρτωση και έλεγχος δεδομένων.....	62
5.3	Προβολή Στατιστικών Στοιχείων Μαθημάτων .....	62
5.4	Αναλυτική Στατιστική Ανάλυση Μαθήματος.....	63
Κεφάλαιο 6ο:	Συζήτηση.....	64
6.1	Μεθοδολογία ανάπτυξης AGILE .....	64
6.2	Δυσκολίες .....	64
6.2.1	Δομή ερωτηματολογίων.....	65
6.2.2	Μη προκαθορισμένες απαιτήσεις .....	66
6.2.3	Διεξαγωγή στατιστικών στοιχείων .....	66
6.3	Μελλοντικές βελτιώσεις .....	66
6.3.1	NLP ανάλυση .....	66
6.3.2	Αυστηρή δομή ερωτηματολογίων .....	67
6.3.3	Πολλαπλά τμήματα.....	67
6.4	Συμπεράσματα .....	67
BIBΛΙΟΓΡΑΦΙΑ.....		68

## Πίνακας Σχημάτων

Σχήμα 1. Επικοινωνία ανάμεσα σε πελάτη και εξυπηρετή. Οι Routers δέχονται και απαντούν σε HTTP αιτήσεις. ....	14
Σχήμα 2. Επικοινωνία ανάμεσα σε πελάτη και επυπηρετή. Τα middleware φιλτράρουν τα δεδομένα προτού φτάσουν στους Routers .....	15
Σχήμα 3. Γραφική απεικόνιση του σχεδιαστικού προτύπου Dependency Injection.....	16
Σχήμα 4. Context Diagram του πληροφοριακού συστήματος. Παρουσιάζονται τα είδη των χρηστών που επικοινωνούν με το σύστημα.....	30
Σχήμα 5. Container Diagram του πληροφοριακού συστήματος. Παρουσιάζονται τα containers που αποτελούν το συνολικό πληροφοριακό σύστημα. ....	31
Σχήμα 6. Component Diagram του πληροφοριακού συστήματος. Παρουσιάζονται τα components εντός του Flask container .....	33
Σχήμα 7. Διάγραμμα οντοτήτων συσχετίσεων .....	41
Σχήμα 8. Πίνακες του σχεσιακού σχήματος της κεντρικής βάσης δεδομένων του πληροφοριακού συστήματος. ....	61
Σχήμα 9. στιγμιότυπο οθόνης του πληροφοριακού συστήματος για την ανάρτηση xlsx αρχείων .....	62
Σχήμα 10. Στιγμιότυπο οθόνης του πληροφοριακού συστήματος για την στατιστικών στοιχείων ενός συγκεκριμένου μαθήματος.....	63

## Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
ΜΟΔΙΠ	Μονάδας Διασφάλισης Ποιότητας
ΟΜΕΑ	Ομάδα Εσωτερικής Αξιολόγησης
ΤΜΠΗΣ	Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων



## Κεφάλαιο 1ο: Εισαγωγή

### 1.1 Ιστορικό

#### 1.1.1 ΜΟ.ΔΙ.Π

Η ΜΟ.ΔΙ.Π. είναι μονάδα διασφάλισης ποιότητας του διεθνούς πανεπιστημίου της Ελλάδος και παρέχει σύστημα αξιολόγησης μαθημάτων και διοικητικών υπηρεσιών σε πανεπιστημιακό περιβάλλον. Το σύστημα αυτό χρησιμοποιείται από διάφορα πανεπιστήμια της Ελλάδας για να προσφέρει τη δυνατότητα στους φοιτητές να αξιολογήσουν τα μαθήματα που παρακολουθούν αλλά και τις διοικητικές υπηρεσίες του πανεπιστημίου. Η διαδικασία διεξαγωγής ερωτηματολογίων για την αξιολόγηση των μαθημάτων και των διοικητικών υπηρεσιών του πανεπιστημίου πραγματοποιείται στα μέσα κάθε ακαδημαϊκού εξαμήνου. Οι ενεργοί φοιτητές, δηλαδή αυτοί που παρακολουθούν συστηματικά τις διαλέξεις δια ζώσης, έχουν την δυνατότητα να συμπληρώσουν τα ερωτηματολόγια τα οποία σχετίζονται με την εμπειρία τους και γενικότερα με τις απόψεις τους, ενώ οι διδάσκοντες λαμβάνουν ανώνυμες αξιολόγησης από τους φοιτητές, με σκοπό να αξιολογήσουν σύμφωνα με την εμπειρία τους και να βελτιώσουν τυχόν αδυναμίες που εντοπίζονται στην διαδικασία εκπαίδευσης. Αυτή η διαδικασία διαγωγής ερωτηματολογίων επιτρέπει στους φοιτητές να εκφράζουν ανώνυμα την άποψη τους σχετικά με την εμπειρία τους. Αυτό έχει ως θετικό αποτέλεσμα, τα δεδομένα που συλλέγονται να μπορούν να χρησιμοποιηθούν για τη βελτίωση τόσο των μαθημάτων όσο και των υπηρεσιών που προσφέρονται [1].

#### 1.1.2 Δομή Ερωτηματολογίων

Τα ερωτηματολόγια χωρίζονται σε 2 κατηγορίες, της θεωρίας και των εργαστηρίων, κατά βάση τα δυο ερωτηματολόγια είναι κοινά απλά διαφέρουν σε κάποιες ερωτήσεις και λίγο στην δομή τους. Αυτές οι διαφοροποιήσεις αντικατοπτρίζουν τη φύση των διαφορετικών μαθησιακών περιβαλλόντων μεταξύ θεωρίας και εργαστηρίων. Τα ερωτηματολόγια που διανέμονται στα θεωρητικά μαθήματα περιλαμβάνουν 35 ερωτήσεις οι οποίες είναι οι παρακάτω:

1. Οι στόχοι του μαθήματος είναι σαφείς.
2. Η ύλη που καλύφθηκε ανταποκρίνεται στους στόχους του μαθήματος.
3. Η ύλη είναι καλά οργανωμένη.
4. Το εκπαιδευτικό υλικό (κύριο βιβλίο, διαφάνειες, σημειώσεις κλπ) βοηθάει στην κατανόηση του μαθήματος.
5. Στο μάθημα γίνεται αναφορά στην ύλη άλλων μαθημάτων.
6. Στο μάθημα γίνεται σύνδεση γνώσεων με άλλα μαθήματα.
7. Το επίπεδο δυσκολίας του μαθήματος είναι ανάλογο του έτους του.
8. Ο αριθμός των Διδακτικών Μονάδων του μαθήματος είναι ανάλογες με το φόρτο εργασίας.
9. Τα φροντιστηριακά μαθήματα βοηθούν στην κατανόηση του μαθήματος.
10. Τα φροντιστηριακά μαθήματα ενισχύουν την επίδοσή μου.
11. Οι Ασκήσεις-Πράξεις βοηθούν στην κατανόηση του μαθήματος.
12. Οι Ασκήσεις-Πράξεις του μαθήματος ανταποκρίνονται στους στόχους του μαθήματος.
13. Οι Ασκήσεις-Πράξεις προσφέρουν εφαρμοσμένη γνώση.
14. Το θέμα της εργασίας δόθηκε έγκαιρα.
15. Καθορίστηκαν εκ των προτέρων τα κριτήρια βαθμολόγησης της εργασίας.

16. Υπήρχε καθοδήγηση από τον διδάσκοντα αναφορικά με την εργασία.
17. Η εργασία βοήθησε στην καλύτερη κατανόηση του μαθήματος.
18. Υπάρχει διαφάνεια στα κριτήρια βαθμολόγησης.
19. Η εξέταση ήταν σχετική με το περιεχόμενο των διαλέξεων.
20. Η εξέταση έγινε με τρόπο αξιοκρατικό.
21. Οι υποδομές των αιθουσών διδασκαλίας είναι επαρκείς σε σχέση με τον αριθμό των φοιτητών που τις χρησιμοποιούν.
22. Η ηλεκτρονική υποστήριξη του μαθήματος (e-class, e-learning κλπ) είναι επαρκής.
23. Οι Τεχνολογίες Πληροφορικής και Επικοινωνιών που χρησιμοποιούνται είναι οι κατάλληλες.
24. Επιτυγχάνει να διεγείρει το ενδιαφέρον για το αντικείμενο του μαθήματος.
25. Αναλύει και παρουσιάζει τις έννοιες με τρόπο απλό χρησιμοποιώντας παραδείγματα.
26. Είναι καλά προετοιμασμένος/η.
27. Είναι συνεπής στις υποχρεώσεις του.
28. Είναι μεταδοτικός/ή.
29. Ενθαρρύνει τους φοιτητές να διατυπώνουν απορίες και ερωτήσεις.
30. Είναι προσιτός/η στους φοιτητές.
31. Παρακολουθώ τις διαλέξεις.
32. Μελετώ συστηματικά την ύλη.
33. Χρησιμοποιώ τη βιβλιοθήκη του Ιδρύματος για τη μελέτη πρόσθετης βιβλιογραφίας σχετικής με το μάθημα.
34. Χρησιμοποιώ πηγές του διαδικτύου για τη μελέτη πρόσθετης βιβλιογραφίας σχετικής με το μάθημα.
35. Αφιερώνω εβδομαδιαία για μελέτη του συγκεκριμένου μαθήματος: 1: Καθόλου 2: λιγότερο από μια ώρα 3: 1-2 ώρες 4: 2-3 ώρες 5: περισσότερο από 3 ώρες.

ενώ τα ερωτηματολόγια για τα εργαστήρια περιλαμβάνουν 33 ερωτήσεις οι οποίες είναι οι εξής

1. Οι στόχοι του εργαστηρίου είναι σαφείς.
2. Η ύλη που καλύφθηκε ανταποκρίνεται στους στόχους του μαθήματος.
3. Η ύλη του εργαστηρίου είναι καλά οργανωμένη.
4. Το εκπαιδευτικό υλικό (π.χ. φύλλα εργασίας) που χρησιμοποιείται στο εργαστήριο βοηθάει στην κατανόησή του.
5. Το εργαστήριο είναι συμπληρωματικό του θεωρητικού μέρους του μαθήματος.
6. Το εργαστήριο με βοηθά να αποκτήσω συναφείς με το μάθημα συμπληρωματικές γνώσεις και δεξιότητες.
7. Στο εργαστήριο γίνεται αναφορά στην ύλη άλλων μαθημάτων.
8. Το επίπεδο δυσκολίας του εργαστηρίου είναι ανάλογο του έτους του.
9. Οι υποδομές του εργαστηρίου είναι επαρκείς σε σχέση με τον αριθμό των φοιτητών/τριών που τις χρησιμοποιούν.
10. Οι υποδομές του εργαστηρίου είναι σύγχρονες.
11. Ο εξοπλισμός του εργαστηρίου ανταποκρίνεται στις εκπαιδευτικές ανάγκες του μαθήματος.
12. Ο εξοπλισμός είναι τοποθετημένος σε τέτοια διάταξη που να επιτρέπει τη συνεργασία μεταξύ των φοιτητών/τριών όταν απαιτείται.
13. Το εργαστήριο είναι διαθέσιμο για χρήση από τους φοιτητές/τριες και πέραν των προγραμματισμένων ωρών μαθήματος.
14. Το εργαστήριο πληροί τις απαραίτητες συνθήκες υγιεινής.
15. Το εργαστήριο πληροί τις απαραίτητες συνθήκες ασφάλειας.

16. Οι εργαστηριακές ασκήσεις είναι καλά προετοιμασμένες.
17. Οι θεμελιώδεις αρχές των εργαστηριακών ασκήσεων εξηγούνται προσεκτικά.
18. Καθορίστηκαν εκ των προτέρων τα κριτήρια βαθμολόγησης των εργαστηριακών ασκήσεων.
19. Η υποστήριξη/επίβλεψη από το βοηθητικό προσωπικό του εργαστηρίου (π.χ. ΕΤΕΠ/ΕΔΠ) είναι ικανοποιητική.
20. Υπήρχε ανατροφοδότηση για τις εργαστηριακές ασκήσεις ώστε να μπορώ να βελτιώσω τις δεξιότητές μου.
21. Υπάρχει διαφάνεια στα κριτήρια βαθμολόγησης.
22. Η εξέταση ήταν σχετική με τις εργαστηριακές ασκήσεις.
23. Η εξέταση του εργαστηριακού μαθήματος έγινε με τρόπο αξιοκρατικό.
24. Επιτυγχάνει να διεγείρει το ενδιαφέρον για τις εργαστηριακές ασκήσεις.
25. Γνωρίζει καλά τη χρήση του εργαστηριακού εξοπλισμού.
26. Ενθαρρύνει τους φοιτητές να αναπτύξουν δεξιότητες για τη διεξαγωγή των εργαστηριακών ασκήσεων.
27. Είναι συνεπής στις υποχρεώσεις του.
28. Είναι προσιτός/ή στους φοιτητές.
29. Δεν απουσιάζω από τις εργαστηριακές ασκήσεις.
30. Συνεισφέρω με τις γνώσεις μου κατά τη διάρκεια του εργαστηρίου.
31. Βοηθώ τους συμφοιτητές μου/τρεις μου στις εργαστηριακές ασκήσεις.
32. Στο εργαστήριο αυτό απόκτησα δεξιότητες που πιστεύω ότι είναι απαραίτητες για την εφαρμογή πιο εξειδικευμένων τεχνικών σε επόμενα εργαστήρια.
33. Στο εργαστήριο αυτό απόκτησα δεξιότητες που πιστεύω θα με βοηθήσουν στην επαγγελματική μου πορεία.

Κάθε ερώτηση ανήκει σε μία συγκεκριμένη κατηγορία, οι κατηγορίες του θεωρητικού μέρους του μαθήματος παρατίθενται ως εξής:

1. **Μάθημα:** Η κατηγορία "Μάθημα" περιλαμβάνει ερωτήσεις που σχετίζονται με το πόσο ποιοτικό και αποτελεσματικό είναι το συγκεκριμένο μάθημα που αναλύεται σύμφωνα με διάφορα χαρακτηριστικά. Αρχικά εξετάζει αν ήταν σαφής οι στόχοι του μαθήματος, αν η ύλη ήταν καλά οργανωμένη στις διάφορες πλατφόρμες που οι διδάσκοντες αναρτούν όλα τα αρχεία, καθώς και αν ήταν ποιοτικό το εκπαιδευτικό υλικό. Επιπλέον, αξιολογεί το ποσοστό σύνδεσης διαφόρων γνώσεων με άλλα μαθήματα του τμήματος και πώς ενσωματώνει γνώσεις από άλλα παρόμοια γνωστικά πεδία.
2. **Φροντιστηριακά μαθήματα:** Η κατηγορία "Φροντιστηριακά μαθήματα", περιλαμβάνει ερωτήσεις που σχετίζονται με τα φροντιστηριακά μαθήματα. Φροντιστηριακά μαθήματα προσφέρονται είτε από τους ίδιους τους διδάσκοντες αλλά κυρίως από φοιτητές που έχουν την κατάλληλη τεχνογνωσία και διάθεση για να βοηθήσουν τους συμφοιτητές τους. Σε περίπτωση που δεν προσφέρονται Φροντιστηριακά μαθήματα, οι φοιτητές αφήνουν αναπάντητες τις ερωτήσεις που ανήκουν στην συγκεκριμένη κατηγορία
3. **Ασκήσεις Πράξεις:** Η κατηγορία "Ασκήσεις Πράξεις", περιλαμβάνει ερωτήσεις που σχετίζονται με τις ασκήσεις πράξεις που μοιράζουν οι καθηγητές με σκοπό να βοηθήσουν τους φοιτητές να κατανοήσουν καλύτερα το μάθημα μέσα από πρακτικές εφαρμογές. Σε περίπτωση που οι διδάσκοντες δεν προσφέρουν ασκήσεις πράξεις, οι φοιτητές αφήνουν αναπάντητες τις ερωτήσεις που ανήκουν στην συγκεκριμένη κατηγορία
4. **Εργασίες:** Η κατηγορία "Εργασίες" περιλαμβάνει ερωτήσεις που σχετίζονται με τις ασκήσεις που μοιράζουν οι διδάσκοντες. Ο κύριος στόχος αυτών των ερωτήσεων είναι να αξιολογήσουν

εάν οι εργασίες συνέβαλαν στην καλύτερη κατανόηση του μαθήματος καθώς και εάν ο διδάσκων ήταν συνεπής στην διαδικασία ανάθεσης των εργασιών. Σε περίπτωση που οι διδάσκοντες δεν προσφέρουν εργασία εξαμήνου, οι φοιτητές αφήνουν αναπάντητες τις ερωτήσεις που ανήκουν στην συγκεκριμένη κατηγορία

5. **Εξέταση του μαθήματος:** Η κατηγορία "Εξέταση του μαθήματος" περιλαμβάνει ερωτήσεις που σχετίζονται με την τελική εξέταση του μαθήματος. Ο κύριος στόχος αυτών των ερωτήσεων είναι να αξιολογήσουν εάν υπήρξε διαφάνεια στα κριτήρια αξιολόγησης, επίσης και αν το θέμα της τελικής εξέτασης εμπεριείχε ερωτήσεις που σχετίζονται άμεσα με αυτά που διδάχθηκαν στις διαλέξεις του διδάσκοντα. Οι συγκεκριμένες ερωτήσεις απαντώνται από τους φοιτητές μόνο αν η διαδικασία αξιολόγησης πραγματοποιείται αφού έχει ολοκληρωθεί η τελική εξέταση του μαθήματος.
6. **Εκπαιδευτικές υποδομές:** Η κατηγορία "Εκπαιδευτικές υποδομές" περιλαμβάνει ερωτήσεις που σχετίζονται με τις εγκαταστάσεις και τις υποδομές που παρέχονται από το πανεπιστήμιο στους φοιτητές. Ο κύριος σκοπός αυτών των ερωτήσεων είναι να αξιολογήσουν την ποιότητα των εκπαιδευτικών υποδομών, καθώς και το πως ανταποκρίνονται αυτές στις ανάγκες των φοιτητών. Επιπλέον, οι απαντήσεις σε αυτές τις ερωτήσεις μπορούν να παρέχουν στο ίδρυμα σημαντικές πληροφορίες για να ανιχνευθούν τυχόν αδυναμίες που μπορούν να βελτιωθούν για την καλύτερη εμπειρία των φοιτητών.
7. **Ο διδάσκων/ουσα:** Η κατηγορία "Ο διδάσκων/ουσα" περιλαμβάνει ερωτήσεις που σχετίζονται με τον διδάσκοντα του εκάστοτε μαθήματος. Ο κύριος στόχος αυτών των ερωτήσεων είναι να αξιολογήσουν τον διδάσκοντα για το αν καταφέρνει να διεγείρει το ενδιαφέρον των φοιτητών για το αντικείμενο του μαθήματος που διδάσκει, επίσης και αν ο διδάσκων είναι προετοιμασμένος για τις διαλέξεις που πραγματοποιεί. Είναι ζωτικής σημασίας η ανατροφοδότηση που προκύπτει από αυτές τις ερωτήσεις, επίσης δίνεται η ευκαιρία στους διδάσκοντες να αναγνωρίσουν τα σημεία στα οποία υπάρχουν πιθανώς αδυναμίες και σε περίπτωση που το επιθυμούν να εστιάσουν στη βελτίωσή τους, με αποτέλεσμα να επιτύχουν μια πιο αποτελεσματική εκπαιδευτική διαδικασία στο μέλλον.
8. **Εγώ ο φοιτητής:** Η κατηγορία "Εγώ ο φοιτητής" περιλαμβάνει ερωτήσεις που αφορούν τον συγκεκριμένο φοιτητή που παρακολουθεί το μάθημα για το οποίο κάνει την αξιολόγηση. Ο κύριος στόχος αυτών των ερωτήσεων είναι να διαπιστώσουν οι διδάσκοντες πόσο ώρα αφιερώνουν οι φοιτητές για μελέτη ή ανάγνωση και αν έρχονται στις διαλέξεις. Επίσης σε αυτήν την κατηγορία υπάρχουν ερωτήσεις για το πως οι φοιτητές ανακτούν τις πληροφορίες. Εάν χρησιμοποιούν την βιβλιοθήκη του ιδρύματος ή πηγές του διαδικτύου

Οι κατηγορίες του εργαστηριακού μέρους του μαθήματος παρατίθενται ως εξής:

1. **Μάθημα:** Η κατηγορία "Μάθημα" περιλαμβάνει ερωτήσεις που σχετίζονται με το πόσο ποιοτικό και αποτελεσματικό είναι το συγκεκριμένο εργαστηριακό μάθημα που αναλύεται σύμφωνα με διάφορα χαρακτηριστικά. Αρχικά εξετάζει αν ήταν σαφής οι στόχοι του εργαστηρίου, αν η ύλη ήταν καλά οργανωμένη στις διάφορες πλατφόρμες που οι διδάσκοντες αναρτούν όλα τα αρχεία, καθώς και αν ήταν ποιοτικό το εκπαιδευτικό υλικό. Επιπλέον, αξιολογεί το ποσοστό σύνδεσης διαφόρων γνώσεων με άλλα μαθήματα του τμήματος και πώς ενσωματώνει γνώσεις από άλλα παρόμοια γνωστικά πεδία.
2. **Εργαστηριακές υποδομές:** Η κατηγορία "Εργαστηριακές υποδομές" περιλαμβάνει ερωτήσεις που σχετίζονται με τις εγκαταστάσεις και την υποδομή των εργαστηρίων. Ο σκοπός αυτών των ερωτήσεων είναι να αξιολογήσουν εάν οι υποδομές των εργαστηρίων είναι επαρκείς σε σχέση με τον αριθμό των φοιτητών/τριών που τις χρησιμοποιούν, επίσης αν είναι σύγχρονες και αν

έχουν κατάλληλο εξοπλισμό που ευνοεί τους φοιτητές να διεξάγουν αποτελεσματικά το εργαστήριο. Επιπλέον, οι απαντήσεις σε αυτές τις ερωτήσεις μπορούν να παρέχουν στο πανεπιστήμιο σημαντικές πληροφορίες για καταστάσεις που θα μπορούσαν να βελτιωθούν. Οι απαντήσεις που δεν ήταν θετικές είναι ευκαιρία για να εμφανίσουν ελαττώματα στις εκπαιδευτικές υποδομές, με αποτέλεσμα να προσφέρονται πληροφορίες στο ίδρυμα για το ποια μέρη και συστήματα πρέπει να βελτιωθούν.

3. **Εργαστηριακές ασκήσεις:** Η κατηγορία "Εργαστηριακές ασκήσεις" περιέχει ερωτήσεις που σχετίζονται με τις ασκήσεις που πραγματοποιούν οι φοιτητές απευθείας στα εργαστήρια αλλά και τις ασκήσεις που πραγματοποιούν εκτός του εργαστηρίου για την προσωπική τους μελέτη.
4. **Εξέταση του μαθήματος:** Η κατηγορία "Εξέταση του μαθήματος" περιλαμβάνει ερωτήσεις που σχετίζονται με την τελική εξέταση του εργαστηρίου. Ο κύριος στόχος αυτών των ερωτήσεων είναι για να αναδείξουν εάν υπήρξε διαφάνεια στα κριτήρια που αξιολογήθηκαν οι φοιτητές του εργαστηρίου, επίσης και αν η τελική εξέταση ήταν σχετική με τις εργαστηριακές ασκήσεις. Οι συγκεκριμένες ερωτήσεις απαντώνται από τους φοιτητές μόνο αν η διαδικασία αξιολόγησης του εργαστηρίου έγινε μετά την τελική εξέταση του μαθήματος.
5. **Ο διδάσκων/ουσα:** Η κατηγορία "Ο διδάσκων/ουσα" περιλαμβάνει ερωτήσεις που σχετίζονται με τον διδάσκοντα του εκάστοτε μαθήματος που αξιολογείται. Ο κύριος στόχος αυτών των ερωτήσεων είναι να αξιολογήσουν τον διδάσκοντα για το αν καταφέρνει να διεγείρει το ενδιαφέρον των φοιτητών για το εργαστηριακό μέρος του μαθήματος και αν ήταν κατάλληλα προετοιμασμένος για την διεξαγωγή του εργαστηρίου. Η ανατροφοδότηση που προκύπτει από αυτές τις ερωτήσεις είναι πολύ σημαντική για τους διδάσκοντες, επειδή τους δίνεται η δυνατότητα να αναγνωρίσουν τις περιοχές στις οποίες πιθανώς έχουν αδυναμίες και σε περίπτωση που το επιθυμούν να εστιάσουν στη βελτίωσή τους, με αποτέλεσμα να επιτύχουν μια πιο αποτελεσματική εκπαιδευτική διαδικασία στο εργαστήριο.
6. **Εγώ ο φοιτητής:** Η κατηγορία "Εγώ ο φοιτητής" περιλαμβάνει ερωτήσεις που αφορούν τον συγκεκριμένο φοιτητή που παρακολουθεί το εργαστήριο για το μάθημα που αξιολογεί. Ο κύριος στόχος αυτών των ερωτήσεων είναι για να διαπιστώσουν οι διδάσκοντες πόσοι ώρα αφιερώνουν οι φοιτητές για την προσωπική τους μελέτη εκτός πανεπιστημίου. Επίσης σε αυτήν την κατηγορία υπάρχουν ερωτήσεις για το πως οι φοιτητές ψάχνουν τις πληροφορίες που σχετίζονται με το εργαστήριο. Διαπιστώνεται εάν Χρησιμοποιούν την βιβλιοθήκη του ιδρύματος ή αν αναζήτησαν τις πληροφορίες σε πηγές του διαδικτύου.

### 1.1.3 Διαδικασία συλλογής των δεδομένων

Για να ξεκινήσει η διαδικασία αξιολόγησης του μαθήματος πρέπει πρώτα ο διδάσκων του κάθε μαθήματος να εκτυπώσει τους κωδικούς αξιολόγησης από την ιστοσελίδα <https://modip.ihu.edu.gr>. Οι κωδικοί που εκτυπώνονται είναι απαραίτητοι για να διασφαλίσει το σύστημα ότι μόνο οι φοιτητές που χαρακτηρίζονται ως ενεργοί συμπληρώνουν τα ερωτηματολόγια, για αυτό τον λόγο οι κωδικοί αξιολόγησης μοιράζονται μόνο στις δια ζώσης διαλέξεις. Κατά τη διάρκεια διεξαγωγής του μαθήματος, ο διδάσκων αποφασίζει για το αν θα μοιράσει τους κωδικούς μέσω ψηφιακού αρχείου ή αν θα μοιράσει σε κάθε φοιτητή ξεχωριστά έναν μοναδικό κωδικό αξιολόγησης. Στη συνέχεια, ενημερώνει τους φοιτητές για τη διαδικασία αξιολόγησης του μαθήματος και τον τρόπο συμπλήρωσης των ερωτηματολογίων. Οι φοιτητές, από την στιγμή που κατέχουν τον κωδικό αξιολόγησης του μαθήματος, συνδέονται στην ιστοσελίδα <https://modip.ihu.edu.gr> χρησιμοποιώντας τον ιδρυματικό τους λογαριασμό. Στη συνέχεια, προχωρούν στην ανώνυμη αξιολόγηση του μαθήματος. Μετά την ολοκλήρωση του μαθήματος, ο διδάσκων έχει τη δυνατότητα να συνδεθεί οποιαδήποτε στιγμή στην

ιστοσελίδα <https://modip.ihu.edu.gr/> και να δει τη συμμετοχή των φοιτητών στην αξιολόγηση καθώς και τα αποτελέσματα αυτής για τα μαθήματά του [2].

## 1.2 Καθορισμός του προβλήματος

Οι αξιολογήσεις των φοιτητών αποθηκεύονται στην ιστοσελίδα <https://modip.ihu.edu.gr>. Οι διδάσκοντες έχουν πρόσβαση στις αξιολογήσεις των μαθημάτων για τα οποία είναι υπεύθυνοι, ενώ οι καθηγητές που είναι μέλη της Ομάδας Εσωτερικής Αξιολόγησης (ΟΜΕΑ) έχουν πρόσβαση στις αξιολογήσεις όλων των μαθημάτων. Η πληροφορία για τα μαθήματα ενός συγκεκριμένου εξαμήνου οργανώνεται σε ένα αρχείο xlsx με σχεδόν αυστηρή δομή. Ωστόσο, οι διδάσκοντες προκειμένου να αντλήσουν στατιστικά στοιχεία από αυτά τα αρχεία, είτε πρέπει να δημιουργήσουν δικά τους προγράμματα που θα αναλύσουν τα αρχεία, είτε να προβούν σε χειροκίνητη επεξεργασία. Όπως είναι εύκολα αντιληπτό, αυτή η διαδικασία είναι ιδιαίτερα χρονοβόρα και πολλές φορές μπορεί να οδηγήσει σε λάθη ή λανθασμένα συμπεράσματα αν δεν εκτελεστεί σωστά. Η διπλωματική εργασία μου αναλαμβάνει την αντιμετώπιση αυτού του προβλήματος. Συγκεκριμένα, αφορά την ανάπτυξη ενός πληροφοριακού συστήματος που θα επιτρέπει στους διδάσκοντες να ανεβάζουν τα αρχεία xlsx για κάθε εξάμηνο με εύκολο και αποτελεσματικό τρόπο. Το σύστημα θα παρέχει επίσης τη δυνατότητα προβολής των στατιστικών στοιχείων του κάθε εξαμήνου και τη σύγκρισή τους με τα προηγούμενα εξάμηνα. Αυτό έχει μεγάλη σημασία, καθώς προάγει τη διαφάνεια στα αποτελέσματα της αξιολόγησης.

## 1.3 Στόχοι

Το θέμα της διπλωματικής εργασίας έχει ως κύριο στόχο την ανάπτυξη ενός ολοκληρωμένου πληροφοριακού συστήματος που θα διαχειρίζεται και αναλύει τις αξιολογήσεις των φοιτητών όλων των πανεπιστημίων του ανήκουν στο ΔΠΠΑΕ. Στο σύστημα αυτό θα έχουν πρόσβαση μόνο διδάσκοντες, και τα μέλη της ΟΜΕΑ που θα είναι και οι διαχειριστές του συστήματος. Ένας από τους βασικούς στόχους είναι η δημιουργία μιας εύχρηστης πλατφόρμας που θα επιτρέπει στους διδάσκοντες να ανεβάζουν τα αρχεία αξιολόγησης και να βλέπουν τα στατιστικά στοιχεία των μαθημάτων που διδάσκουν. Με αυτόν τον τρόπο, οι διδάσκοντες μπορούν να παρακολουθούν την εξέλιξη της ποιότητας της διδασκαλίας τους και να λαμβάνουν αποτελεσματικά μέτρα για τη βελτίωσή της.

Το πληροφοριακό σύστημα θα μπορεί να δέχεται αρχεία της μορφής xlsx και θα πραγματοποιείται αυτόματα η διαδικασία ανάλυσης των δεδομένων χωρίς να χρειάζεται να παρέμβει κάποιος άνθρωπος στην διαδικασία. Αυτό θα εξοικονομήσει αρκετό χρόνο και πολλούς πόρους για τους διδάσκοντες και θα επιτρέψει την πιο γρήγορη πρόσβαση στα αποτελέσματα των αξιολογήσεων. Επιπλέον, η αυτοματοποίηση της διαδικασίας θα εξασφαλίσει ότι τα στατιστικά στοιχεία που εξάγονται είναι σωστά επειδή δεν θα υπάρχει κάποια ανθρώπινη παρέμβαση στην διαδικασία που θα μπορούσε να δημιουργήσει κάποιο σφάλμα.

Το πληροφοριακό σύστημα θα έχει ενσωματωμένο μηχανισμό που θα ανιχνεύει πιθανά συντακτικά λάθη στη δομή των αρχείων xlsx. Σε περίπτωση ανίχνευσης σφαλμάτων, το σύστημα θα ενημερώνει τους διδάσκοντες για το είδος του λάθους που παρουσιάστηκε, όπως για παράδειγμα, η απουσία του κωδικού του μαθήματος, το όνομα του διδάσκων δεν είναι καταχωρημένο στην βάση δεδομένων ή αν δεν έχουν συμπληρωθεί άλλα αναγκαία πεδία. Αυτό επιτρέπει στους διδάσκοντες να διορθώνουν τα λάθη εύκολα και γρήγορα, εξασφαλίζοντας την ορθότητα των δεδομένων και την ομαλή λειτουργία του πληροφοριακού συστήματος.

Τέλος, το πληροφοριακό σύστημα πρέπει να υποστηρίζει την ανάλυση δεδομένων και την δυνατότητα σύγκρισης των αποτελεσμάτων αξιολόγησης ανάμεσα σε διαφορετικά εξάμηνα. Αυτή η

λειτουργικότητα είναι πολύ σημαντική επειδή οι διδάσκοντες μπορούν να εντοπίσουν αν κάποιο μάθημα βελτιώνεται ή όχι με την πάροδο του χρόνου.

#### 1.4 Επισκόπηση του συστήματος

Το πληροφοριακό σύστημα έχει υλοποιηθεί μέσω των web τεχνολογιών, οι διδάσκοντες και μέλη της ΟΜΕΑ μπορούν να έχουν πρόσβαση στο πληροφοριακό σύστημα μέσω ενός οποιουδήποτε φυλλομετρητή. Για να μπορέσουν οι διδάσκοντες να αναρτήσουν κάποιο xlsx αρχείο ή να δουν τα στατιστικά στοιχεία κάποιου συγκεκριμένου μαθήματος πρέπει πρώτα να εξουσιοδοτηθούν από το σύστημα κάνοντας login με τους προσωπικούς τους ακαδημαϊκούς κωδικούς.

Το web app χωρίζεται σε 2 βασικές κατηγορίες

1. Η **γραφική διεπαφή** (frontend) Μέσω της γραφικής διεπαφής οι διδάσκοντες αλληλοεπιδρούν με το σύστημα. Πιο συγκεκριμένα είναι ο κώδικας που εκτελείται στην πλευρά του πελάτη. Η γραφική διεπαφή λαμβάνει όλες τα δεδομένα που χρειάζεται από το backend. Η γραφική διεπαφή χρησιμοποιεί το πρωτόκολλο HTTP για να επικοινωνήσει με τον πελάτη.
2. Ο **Web Server** (Backend) Στο backend, δηλαδή το μέρος που δεν είναι άμεσα ορατό στον πελάτη, εκτελείται όλη η επιχειρησιακή λογική. Μέσα στο πίσω μέρος υπάρχει και μία βάση δεδομένων όπου εκεί αποθηκεύονται όλες οι πληροφορίες του συστήματος.

## Κεφάλαιο 2ο: Βιβλιογραφική ανασκόπηση

### 2.1 Εισαγωγή

Στο συγκεκριμένο κεφάλαιο γίνεται βιβλιογραφική ανασκόπηση όλων των διαθέσιμων τεχνολογιών που θα μπορούσαν να χρησιμοποιηθούν για την ανάπτυξη του πληροφοριακού συστήματος ΟΜΕΑ. Αρχικά αναλύονται οι ήδη υπάρχουσες διαθέσιμες τεχνολογίες οι οποίες δεν είναι απαραίτητα ανταγωνιστικές αλλά θα μπορούσαν να συνυπάρξουν στην διαδικασία διεξαγωγής στατιστικών στοιχείων, όπως για παράδειγμα το λογισμικό Tableau, το IBM SPSS Statistics και το Apache Superset. Στο τέλος αναλύονται οι τεχνολογίες που χρησιμοποιήθηκαν για το πληροφοριακό σύστημα της ΟΜΕΑ σε συνδυασμό με την αντίστοιχη τεκμηρίωση για τον λόγο που επιλέχθηκαν.

### 2.2 Υπάρχουσες λύσεις

#### 2.2.1 Tableau

Το Tableau είναι μια ολοκληρωμένη εφαρμογή κλειστού λογισμικού η οποία σχεδιάστηκε, αναπτύχθηκε και συντηρείται από την εταιρία Salesforce. Το Tableau μπορεί να χρησιμοποιηθεί είτε μέσω Desktop εφαρμογής είτε μέσω της διαδικτυακής υπηρεσίας νέφους που παρέχει η Salesforce. Η επιλογή αυτή γίνεται σύμφωνα με τις ανάγκες της επιχείρησης. Το λογισμικό αυτό απευθύνεται σε βιομηχανίες μεγάλης κλίμακας που χρειάζεται για να διαχειριστούν μεγάλο όγκο δεδομένων. Το Tableau έχει την δυνατότητα να δέχεται δεδομένα από διάφορες πηγές όπως για παράδειγμα βάσεις δεδομένων, αρχεία xlsx και υπηρεσίες νέφους. Επίσης δεν περιορίζεται στον τύπο δεδομένων των πληροφοριών, υποστηρίζει τόσο δομημένα όσο και αδόμητα δεδομένα. Τα δεδομένα που συλλέγει τα μεταφράζει σε χρήσιμη πληροφορία ώστε οι εταιρίες να μπορούν να παίρνουν βέλτιστες αποφάσεις σε πολύ σύντομα χρονικά διαστήματα. Η πληροφορία εμφανίζεται στον τελικό χρήστη στην μορφή διαγραμμάτων, γραφημάτων και χάρτες [3].

Το δυνατό του χαρακτηριστικό είναι ότι οι χρήστες έχουν την δυνατότητα να δημιουργούν τις δικές τους οθόνες μέσω τις οποίες προσαρμόζουν πίνακες οργάνων με φίλτρα, παραμέτρους και άλλα διαδραστικά στοιχεία ελέγχου για να διευκολύνουν την διαδικασία ανάλυσης δεδομένων. Επίσης μπορούν τις συγκεκριμένες οθόνες να τις κοινοποιήσουν με άλλα μέλη της εταιρείας ή με άλλους εξωτερικούς συνεργάτες εφόσον έχουν την κατάλληλη εξουσιοδότηση [4].

Η διεπαφή του είναι πολύ απλή και προσιτή σε όλους τους χρήστες, έχει σχεδιαστεί με τέτοιο τρόπο ώστε να είναι πολύ εύχρηστη σε χρήστες που δεν έχουν προγραμματιστικό υπόβαθρο. Οι περισσότερες λειτουργίες του συστήματος επιτυγχάνονται μέσω dragging and dropping, αυτό έχει ως πλεονέκτημα ότι οι χρήστες δεν χρειάζεται να γνωρίζουν απαραίτητα προγραμματισμό για να το χρησιμοποιήσουν το Tableau. Παρόλα αυτά σε περίπτωση που οι χρήστες επιθυμούν να συλλέξουν πιο σύνθετα στατιστικά στοιχεία, το Tableau τους δίνει την δυνατότητα να γράψουν κώδικα με τις γλώσσες προγραμματισμού R και Python για να αποκτήσουν τα επιθυμητά αποτελέσματα [5].

#### 2.2.2 IBM SPSS Statistics

Το IBM SPSS Statistics, συνήθως αναφερόμενο και ως μόνο SPSS, είναι μια εφαρμογή ανάλυσης δεδομένων κλειστού λογισμικού η οποία σχεδιάστηκε από την εταιρία SPSS, στην συνέχεια όμως πιο συγκεκριμένα το 2009 εξαγοράστηκε από την εταιρία IBM και πλέον συντηρείται από την τελευταία.

Η αναφερόμενη εφαρμογή χρησιμοποιείται από ερευνητές, μαθητές, επιχειρήσεις και εκπαιδευτές. Το SPSS υπάρχει μόνο σε Desktop application, όσοι χρήστες επιθυμούν να το χρησιμοποιήσουν πρέπει πρώτα να εγκαταστήσουν την εφαρμογή τοπικά στους υπολογιστές τους, αυτό όμως έχει το πλεονέκτημα ότι δεν απαιτείται η σύνδεση στο διαδίκτυο για να επιτευχθεί η στατιστική ανάλυση. Τα δεδομένα στο SPSS μπορούν να φορτωθούν με 2 τρόπους. Είτε μέσω αρχείων xlsx και CSV, είτε μέσω σύνδεσης με σχεσιακή βάση δεδομένων. Πιο συγκεκριμένα το SPSS υποστηρίζει τις βάσεις δεδομένων IBM Db2, Microsoft SQL Server, Oracle Database, MySQL και PostgreSQL [6].

Τα πλεονεκτήματα του SPSS είναι κοινά με το Tableau. Οι χρήστες μπορούν να αναλύσουν τα δεδομένα μέσω εύχρηστης διεπαφής και δεν προϋποθέτει ότι οι χρήστες γνωρίζουν προγραμματισμό για να μπορέσουν να την εξάγουν στατιστικά στοιχεία. Εφόσον όμως υπάρχει ανάγκη για εξαγωγή σύνθετων στατιστικών στοιχείων, οι χρήστες της εφαρμογής μπορούν να χρησιμοποιήσουν τις γλώσσες προγραμματισμού R και Python για να αποκτήσουν τα εξειδικευμένα αποτελέσματα που επιθυμούν.

Η IBM προσφέρει ένα πλήρες ανεπτυγμένο οικοσύστημα εφαρμογών. Το SPSS μπορεί να συνδεθεί και με άλλες εφαρμογές της IBM όπως για παράδειγμα το IBM Watson Studio και το IBM Cognos Analytics. το IBM Watson Studio βοηθάει τους προγραμματιστές και τους αναλυτές δεδομένων να δημιουργούν μοντέλα μηχανικής μάθησης και το IBM Cognos Analytics είναι μια πλατφόρμα επιχειρηματικής τεχνητής νοημοσύνης στοχεύει στην ανάλυση πληροφοριών και επιτρέπει στους οργανισμούς να απεικονίζουν, να αναλύουν και να μοιράζονται πληροφορίες από τα δεδομένα τους. Με την ενσωμάτωση του SPSS Statistics με το IBM Watson Studio και το IBM Cognos Analytics, οι χρήστες μπορούν να αξιοποιήσουν τα πλεονεκτήματα τις κάθε εφαρμογής και να βελτιώσουν τον τρόπο εργασίας τους στην ανάλυση δεδομένων και να οδηγήσουν σε καλύτερες στρατηγικές αποφάσεις που βασίζονται σε δεδομένα [7], [8].

### 2.2.3 Apache Superset

Το Apache Superset είναι ένα πλήρες λογισμικό ανοιχτού κώδικα το οποίο αρχικά είχε σχεδιαστεί από τον προγραμματιστή Maxime Beauchemin που εκείνη την περίοδο εργαζόταν εντός της εταιρίας airbnb ως μηχανικός δεδομένων. Το 2016, η Airbnb άλλαξε την άδεια λογισμικού του Superset, κάνοντας το διαθέσιμο στην ευρύτερη κοινότητα προγραμματιστών και αναλυτών υπό την άδεια Apache License. Πλέον συνεχίζει να συντηρείται και να φιλοξενείται από τον οργανισμό Apache. Οποιοσδήποτε προγραμματιστής το επιθυμεί μπορεί να συνεισφέρει στο έργο Apache Superset μέσω του GitHub. Το apache superset ομοίως με τα λογισμικά Tableau και το SPSS, υποστηρίζει την άμεση εισαγωγή δεδομένων από αρχεία CSV και XLSX παρόλα αυτά το λογισμικό χρησιμοποιείται συνήθως με σύνδεση απευθείας πάνω σε κάποια βάση δεδομένων και μέσω της γραφικής διεπαφής οι χρήστες μπορούν να εξάγουν τα επιθυμητά αποτελέσματα [9], [10].

Από το μεγαλύτερα πλεονεκτήματα του Apache Superset είναι ότι υποστηρίζει πολλές βάσεις δεδομένων. Πιο συγκεκριμένα υποστηρίζει τις παρακάτω βάσεις δεδομένων [9]:

1. Postgresql
2. Google Big Query
3. Snowflake
4. MySQL
5. Amazon REDSHIFT
6. Amazon ATHENA
7. Druid
8. Databricks

9. ClickHouse
10. ROCKSET
11. Dremio
12. Trino
13. Oracle Database
14. Pinot
15. Presto
16. IBM DB2
17. SAP HANA
18. SQL Server
19. DORIS

Ο λόγος που ξεκίνησαν την δημιουργία του Apache Superset ήταν επειδή αισθανόντουσαν την έλλειψη στην αγορά ενός λογισμικού που θα μπορούσε να εκδημοκρατίσει την πρόσβαση στα δεδομένα και να δώσει τη δυνατότητα στους χρήστες που δεν είναι μηχανικοί δεδομένων και αναλυτές δεδομένων να απεικονίζουν δεδομένα με εύκολο και αποδοτικό τρόπο.

### 2.3 Τεχνολογίες

#### 2.3.1 Εισαγωγή

Όπως αναφέρθηκε και στην εισαγωγή, το πληροφοριακό σύστημα που αναπτύχθηκε χωρίζεται σε δύο μέρη. Το front-end και back-end. Στο συγκεκριμένο κεφάλαιο θα εξηγήσουμε τους λόγους για τους οποίους επιλέχθηκαν οι τεχνολογίες σε όλο το stack front-end, back-end και deployment επίσης θα τις συγκρίνουμε με ανταγωνιστικές τεχνολογίες που θα μπορούσαν να είχαν χρησιμοποιηθεί σε συνδυασμό με την κατάλληλη τεκμηρίωση.

#### 2.3.2 Front-end Technologies

Προτού αναλυθούν τα διάφορα front-end frameworks, πηγάζει η ανάγκη να απαντηθεί η ερώτηση: Για ποιο λόγο προτιμάται η χρήση frameworks front-end αντί τις ανάπτυξης σε native JavaScript;

1. **Γρήγορη ανάπτυξη λογισμικού:** Τα JavaScripts frameworks συνδέονται με πολλές βιβλιοθήκες που επιτρέπουν στους προγραμματιστές να συναρμολογούν τα έτοιμα components που υπάρχουν στην κάθε βιβλιοθήκη με αποτέλεσμα οι διεπαφές χρήστη να αναπτύσσονται σε σύντομο χρονικό διάστημα χωρίς να χρειάζεται οι προγραμματιστές να γράψουν τα πάντα από την αρχή. Επίσης τα περισσότερα JavaScripts frameworks έχουν πολύ εύκολη καμπύλη μάθησης, γεγονός που επιτρέπει τους προγραμματιστές που δεν είναι πολύ εξοικειωμένοι με την τρέχον τεχνολογία να μπορούν να προσαρμοστούν με εύκολο και γρήγορο τρόπο [11].
2. **Συνέπεια:** Τα JavaScripts frameworks έχουν συγκεκριμένη δομή και αρχιτεκτονική και αναγκάζουν τους προγραμματιστές να ακολουθούν μια αυστηρή δομή σε όλα τα προγράμματα που αναπτύσσουν, με αποτέλεσμα να είναι πιο εύκολη και αποδοτική η συνεργασία μεταξύ διαφόρων ομάδων που δεν έχουν άμεση επικοινωνία για να συντηρούν των κώδικα που αναπτύσσουν. Αυτή η συνέπεια οδηγεί σε μια βάση κώδικα που είναι πιο καθαρή και γίνεται πιο εύκολα η συντήρηση της από όλους τους προγραμματιστές που εμπλέκονται στην διαδικασία ανάπτυξης λογισμικού [12].
3. **Βελτιστοποίηση επιδόσεων:** Τα JavaScripts frameworks διαθέτουν συγκεκριμένες λειτουργίες που είναι πολύ σημαντικές για να έχουν καλές επιδόσεις οι διαδικτυακές εφαρμογές. Μια από αυτές τις λειτουργίες είναι ο διαχωρισμός του κώδικα, δηλαδή η διαδικασία με την οποία ο

κώδικας JavaScript δεν φορτώνεται όλος απευθείας αλλά φορτώνονται σε πολλά μικρά κομμάτια. Αυτή η μεθοδολογία ονομάζεται Lazy Loading, είναι μεθοδολογία με την οποία φορτώνει ο φυλλομετρητής μόνο όσα χρειάζεται εκείνη την στιγμή για να λειτουργήσει σωστά. Σημαντική λειτουργία είναι το Virtual DOM το οποίο είναι μια τεχνική που χρησιμοποιείται από πολλές JavaScript βιβλιοθήκες όπως για παράδειγμα η React JS και Vue. Το κύριο κίνητρο του Virtual DOM είναι η βελτίωση της απόδοσης των αλλαγών της διεπαφής χρήστη σε διαδικτυακές εφαρμογές. Το virtual dom είναι μια εικονική αναπαράσταση του κανονικού DOM. Κάθε φορά που γίνεται κάποια αλλαγή στην διεπαφή χρήστη, πρώτα ενημερώνεται το Virtual DOM επειδή είναι πιο γρήγορο και στην συνέχεια γίνεται η σύγκριση του Virtual DOM με το κανονικό DOM ώστε να γίνουν όσο το δυνατόν λιγότερες αλλαγές στον κανονικό DOM. Αυτή η διαδικασία γίνεται επειδή είναι πάρα πολύ κοστοβόρες οι αλλαγές στο κανονικό DOM επειδή πρέπει να γίνουν ξανά render στην οθόνη όλα τα στοιχεία σε σύγκριση με το Virtual DOM που γίνονται όλες οι αλλαγές εικονικά στην μνήμη του φυλλομετρητή. Αυτές οι λειτουργίες που αναφέρθηκαν παραπάνω βοηθούν στη βελτίωση τόσο της ταχύτητας όσο και στην απόκριση των διαδικτυακών εφαρμογών [12], [11].

### 2.3.2.1 React JS

Η React JS είναι μια ανοιχτού κώδικα front-end τεχνολογία, πιο συγκεκριμένα μια βιβλιοθήκη της γλώσσας προγραμματισμού JavaScript. Αναπτύχθηκε από την εταιρία Facebook το 2011 και επίσημα δημοσιεύτηκε σε όλη την κοινότητα προγραμματιστών τον Μάιο του 2013. Το γεγονός ότι είναι front-end τεχνολογία σημαίνει ότι όλος ο κώδικας της εκτελείται στον φυλλομετρητή του πελάτη. Η φιλοσοφία της React js είναι ότι χωρίζουμε την ιστοσελίδα μας σε πολλά μικρά ανεξάρτητα κομμάτια τα οποία ονομάζουμε components. Επί της ουσίας κάθε component λειτουργεί με τον ίδιο ακριβώς τρόπο μιας συνάρτησης, δέχονται παραμετρικά κάποια ορίσματα τα οποία αποκαλούνται props, γίνονται εσωτερικά κάποιοι υπολογισμοί και επιστρέφεται ένα DOM στοιχείο. Ο τρόπος με τον οποίο θα χωρίσουμε την ιστοσελίδα μας σε components είναι ευρετικός παρόλα αυτά στην βιβλιογραφία έχουν αναφερθεί κάποιες καλές πρακτικές διαχωρισμού κώδικα σε components ώστε ο κώδικας να είναι εύκολα αναβαθμίσιμος και η ιστοσελίδα που αναπτύσσεται να έχει καλύτερες επιδόσεις [12].

Η React JS λειτουργεί με Δηλωτικό τρόπο, αυτό σημαίνει ότι οι προγραμματιστές στον κώδικα τους δηλώνουν μόνο πως επιθυμούν να είναι η γραφική του διεπαφή και όχι τις ενέργειες που πρέπει να πραγματοποιηθούν ώστε να χτιστεί η γραφική διεπαφή. Αυτό έχει ως αποτέλεσμα να είναι πολύ εύκολη η διαχείριση της κατάστασης της εφαρμογής. Επιπλέον, ένα σημαντικό πλεονέκτημα της βιβλιοθήκης είναι ότι ο κώδικας που γράφεται για την δημιουργία μιας διαδραστικής ιστοσελίδας μπορεί με εύκολο τρόπο να μεταφερθεί σε άλλο περιβάλλον για την ανάπτυξη native εφαρμογών για κινητές συσκευές Android και IOS [12].

### 2.3.2.2 Angular

Η Angular είναι μια βιβλιοθήκη ανοιχτού κώδικα για την ανάπτυξη δυναμικών διεπαφών για διαδικτυακές εφαρμογές, υποστηρίζει τις γλώσσες προγραμματισμού JavaScript και TypeScript. Αρχικά δημιουργήθηκε από τον Misko Hevery ο οποίος το 2010 εργαζόταν στην Google, όταν δημοσιεύτηκε το 2010 κέρδισε αμέσως μεγάλη δημοτικότητα στην κοινότητα των προγραμματιστών. Η βασική φιλοσοφία που υπάρχει πίσω από την Angular είναι η δημιουργία εφαρμογών που αποτελούνται από μικρά, ανεξάρτητα κομμάτια, τα οποία ονομάζονται συνήθως "components" όπως ισχύει και για την React JS που περιγράφεται παραπάνω. Κάθε component λειτουργεί ως ένα αυτόνομο

τμήμα της εφαρμογής, που δέχεται εισόδους σε αποκαλούμενοι και props, εκτελώντας εσωτερικούς υπολογισμούς και επιστρέφοντας ένα στοιχείο DOM [13].

### 2.3.2.3 Vue JS

Η Vue.js είναι μια ανοιχτού κώδικα front-end τεχνολογία, επί της ουσίας μία βιβλιοθήκη της γλώσσας προγραμματισμού JavaScript που αναπτύχθηκε αρχικά από τον Evan You το 2014. Το κύριο κίνητρο ανάπτυξης της Vue Js ήταν η δημιουργία βιβλιοθήκης η οποία θα είναι πιο ελαφριά από την Angular JS και θα είχε καλύτερες επιδόσεις. Ο Evan για την δημιουργία της Vue, συνδύασε όλα τα καλά χαρακτηριστικά της React και της Angular, με σημαντική διαφορά ότι επιθυμούσε η Vue να είναι πολύ πιο απλή στην χρήση ώστε να μπορέσει προσελκύσει μεγαλύτερο πλήθος προγραμματιστών. Από την στιγμή που αναπτύχθηκε, έχει κεντρίσει το ενδιαφέρον πολλών προγραμματιστών διαδικτυακών εφαρμογών και πλέον έχει γίνει πολύ δημοφιλής στην κοινότητα ανοιχτού λογισμικού. Η βασική φιλοσοφία της Vue.js είναι παρόμοια με αυτήν της React.js, καθώς και αυτή επικεντρώνεται στον διαχωρισμό της εφαρμογής σε components. Κάθε Component λειτουργεί ως ανεξάρτητη μονάδα, λαμβάνοντας παραμέτρους που ονομάζονται props και επιστρέφοντας ένα DOM στοιχείο. Ο τρόπος με τον οποίο διαχωρίζουμε και δημιουργούμε τα components στην Vue.js είναι επίσης ευέλικτος και στην βιβλιογραφία έχουν δημοσιευτεί πολλές καλές πρακτικές για την ανάπτυξη οργανωμένων και αποδοτικών εφαρμογών. Η Vue.js για την οργάνωση των εσωτερικών της Components χρησιμοποιεί το σχεδιαστικό πρότυπο αρχιτεκτονικής MVVM. Το συγκεκριμένο σχεδιαστικό πρότυπο προσφέρει μεγάλη ευελιξία στους προγραμματιστές επειδή τους επιτρέπει να διαχωρίσουν τον κώδικα που σχετίζεται με την γραφική διεπαφή και με τον κώδικα που σχετίζεται με την επιχειρησιακή λογική. Με αποτέλεσμα οι αλλαγές που γίνονται στην γραφική διεπαφή δεν επηρεάζουν άμεσα την επιχειρησιακή λογική και αντίστροφα [14].

### 2.3.2.4 Επιλέχθηκε η React

Όπως μπορούμε να διαβάσουμε και από τις παραπάνω παραγράφους, οι JavaScript βιβλιοθήκες διαφέρουν πολύ λίγο μεταξύ τους, όλες οι λειτουργικές απαιτήσεις του πληροφοριακού συστήματος θα μπορούσαν να είχαν υλοποιηθεί και με την Angular ή Vue. Για την ανάπτυξη του λογισμικού όμως επιλέχθηκε ως front-end τεχνολογία η React Js. Παρακάτω αναλύονται οι λόγοι για τους οποίους επιλέχθηκε η React Js σε σύγκριση με την Angular και Vue.

Στην διαδικασία επιλογής τεχνολογίας για την ανάπτυξη κάποια συστήματος πληροφορικής, η εκάστοτε τεχνογνωσία των προγραμματιστών που θα εργαστούν στο έργο λογισμικού είναι το πιο σημαντικό κυριότερο κριτήριο επιλογής. Είναι σημαντικό να σημειωθεί ότι θεωρείται δεδομένο ότι όλες οι βιβλιοθήκες και τεχνολογίες που αναφέρθηκαν παραπάνω μπορούν να χρησιμοποιηθούν εξίσου αποτελεσματικά και αποδοτικά για την υλοποίηση των σημαντικών λειτουργικών απαιτήσεων της εφαρμογής. Λόγω ότι για την ανάπτυξη του πληροφοριακού συστήματος εργάστηκα ατομικά, λήφθηκε υπόψιν η προσωπική μου τεχνογνωσία. Την React Js ήδη την γνώριζα λόγο ότι την είχα χρησιμοποιήσει και σε προηγούμενα έργα ανάπτυξης λογισμικού.

Η Angular έχει μια πιο απότομη καμπύλη εκμάθησης σε σύγκριση με την React JS. Αυτό μπορεί να οδηγήσει σε πιθανές καθυστερήσεις στην ανάπτυξη του πληροφοριακού συστήματος, επειδή η ανάκτηση εμπειρίας και ικανοτήτων για την συγκεκριμένη βιβλιοθήκη απαιτεί περισσότερο χρόνο από τις άλλες. Είναι σημαντικό να επισημανθεί ότι η αύξηση του χρόνου που απαιτείται για την ολοκλήρωση ενός έργου λογισμικού δεν είναι απαραίτητα αρνητικό χαρακτηριστικό, εάν ο χρόνος επενδυθεί σωστά ώστε με την πάροδο του χρόνου να αναπτύσσεται πιο εύκολα και αποδοτικά το σύστημα. Όμως στην προκειμένη περίπτωση η συγκεκριμένη καθυστέρηση χρόνου δεν θα ωφελούσε κάπου διότι η Angular

δεν προσφέρει επιπλέον λειτουργικότητα σε σύγκριση με το React JS. Η React JS σε αντίθεση με την Angular προσφέρει μια πιο ευέλικτη και ελαφριά προσέγγιση, επιτρέποντας στους προγραμματιστές να επικεντρωθούν γρήγορα στην ανάπτυξη λειτουργικού κώδικα.

Παρά την εύκολα αντιληπτή απλότητα της Vue.js σε σχέση με την React.js, είναι σημαντικό να σημειωθεί ότι η Vue.js έχει συνήθως λιγότερη υποστήριξη από την κοινότητα του ανοικτού λογισμικού σε σύγκριση με την βιβλιοθήκη React.js. Αυτό το γεγονός μπορεί εύκολα να γίνει αντιληπτό από τους παρακάτω παράγοντες. Το πρώτο και πιο βασικό είναι η ενεργή κοινότητα προγραμματιστών και ατόμων που συνεισφέρουν σε αυτή, η react συζητιέται πολύ παραπάνω στις ιστοσελίδες που οι προγραμματίστες διαμοιράζουν τις απόψεις τους για τεχνολογίες και γενικότερα γνώσεις που σχετίζονται με τον προγραμματισμό, όπως για παράδειγμα το StackOverFlow και Reddit[15]. Επίσης, σημαντικό παράγοντας είναι ο αριθμός έργων, η React χρησιμοποιείται πολύ σε αρκετές διαδικτυακές εφαρμογές σε όλη την υφήλιο, από μικρές ομάδες ατόμων μέχρι και μεγάλες επιχειρήσεις, γεγονός που δείχνει την εμπιστοσύνη της κοινότητας στην τεχνολογία αυτή. Η React JS διαθέτει ένα μεγάλο φάσμα βιβλιοθηκών και έτοιμου κώδικα που κάνουν πιο εύκολη και αποδοτική την συνολική διαδικασία ανάπτυξης του λογισμικού.

### 2.3.3 Back-end Technologies

Back-end χαρακτηρίζεται το μέρος του συστήματος που δεν είναι άμεσα ορατό στον χρήστη. Όταν ένας χρήστης χρησιμοποιεί μια εφαρμογή αλληλοεπιδρά με το front-end. Στο back-end συνήθως ενθυλακώνεται όλη η σημαντική επιχειρησιακή λογική. Ο λόγος που η επιχειρησιακή λογική βρίσκεται στο back-end και όχι στο front-end είναι επειδή το front-end μπορεί να δεχτεί αλλαγές επειδή εκτελείται στην πλευρά του πελάτη με αποτέλεσμα οι προγραμματιστές να μην μπορούν να έχουν έλεγχο σε αυτό, σε αντίθεση με το back-end το οποίο τρέχει σε server που έχουν ελέγχουν οι προγραμματιστές. Η διεπαφή χρήστη, αν και σημαντική, από μόνη της δεν είναι αρκετή για να καλύψει όλες τις ανάγκες ενός πληροφοριακού συστήματος, καθώς χρειάζεται ένα σημείο πηγής για την αντλία των δεδομένων. Αυτό το σημείο πηγής είναι το back-end. Συνήθως, η επικοινωνία μεταξύ της γραφικής διεπαφής και του back-end πραγματοποιείται μέσω του πρωτοκόλλου HTTP ακολουθώντας την γνωστή αρχιτεκτονική REST API [16].

#### 2.3.3.1 Συστατικά των Web Frameworks

Τα web frameworks παρέχουν στους προγραμματιστές ένα σύνολο από έτοιμες λειτουργίες που σχεδόν όλες οι διαδικτυακές εφαρμογές έχουν ανάγκη. Παρακάτω αναλύονται όλα τα βασικά συστατικά των web frameworks

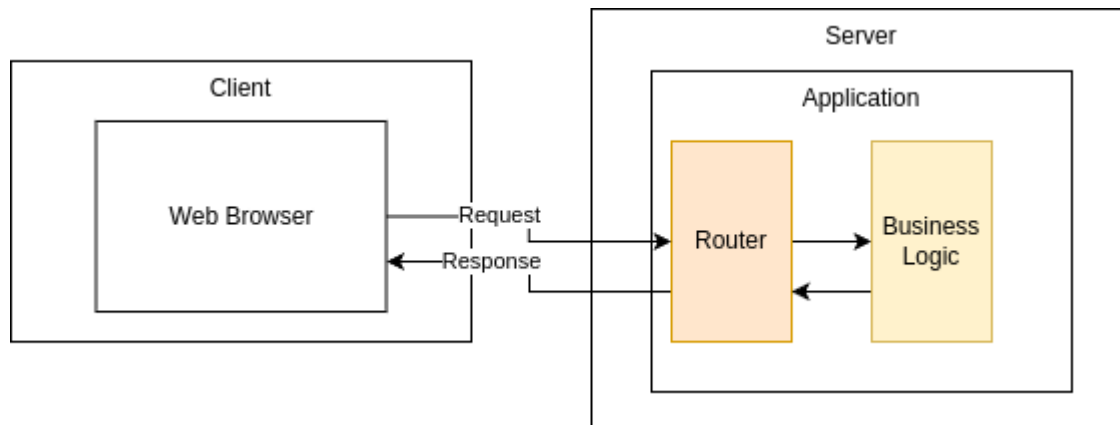
##### 2.3.3.1.1 Routers

Οι Routers ή αλλιώς δρομολογητές είναι από τα πιο σημαντικά συστατικά των frameworks, χωρίς αυτά δεν θα μπορούσε να υπάρχει επικοινωνία ανάμεσα σε πελάτες και frameworks. Ο ρόλος τους είναι να δέχονται HTTP αιτήσεις οποιουδήποτε είδους από πελάτες και ανάλογα με την αίτηση να καλούν την αντίστοιχη συνάρτηση. Μέσα στην συνάρτηση που κλήθηκε γίνονται κάποιες ενέργειες και αφού ολοκληρωθούν στέλνεται πίσω η απάντηση στον πελάτη. Λόγο ότι συνήθως η επικοινωνία ανάμεσα σε πελάτη και διακομιστή γίνεται μέσω της γνωστής δομής δεδομένων JSON, οι περισσότεροι Routers παρέχουν έτοιμες συναρτήσεις για την μετατροπή ενός αντικειμένου σε JSON[17].

Μια καλή σχεδιαστική απόφαση είναι οι Routers να περιορίζονται στον έλεγχο των παραμέτρων και στην κλήση των αντίστοιχων συναρτήσεων. Αυτό είναι σημαντικό για την διατήρηση της αρχής της διαχωρισμένης ανησυχίας γνωστή και ως (separation of concerns), όπου η λογική της εφαρμογής

ενσωματώνεται σε άλλα επίπεδα. Με αυτόν τον τρόπο, αποφεύγεται η επανάληψη κώδικα (code duplication). Με αυτόν τον τρόπο μπορούν οι προγραμματιστές να προσθέτουν νέους routers και η λειτουργικότητα δεν επηρεάζεται επειδή παραμένει ήδη ενσωματωμένη σε άλλα μέρη του συστήματος [17].

Από την οπτική γωνία των προγραμματιστών, οι routers αντιπροσωπεύουν απλές μεθόδους που μπορούν να βρίσκονται εντός ή εκτός κάποιας κλάσης και έχουν ως κύριο σκοπό να ανταποκρίνονται σε αιτήματα HTTP μέσω των αναφορών (annotations) της κάθε γλώσσας προγραμματισμού [17].



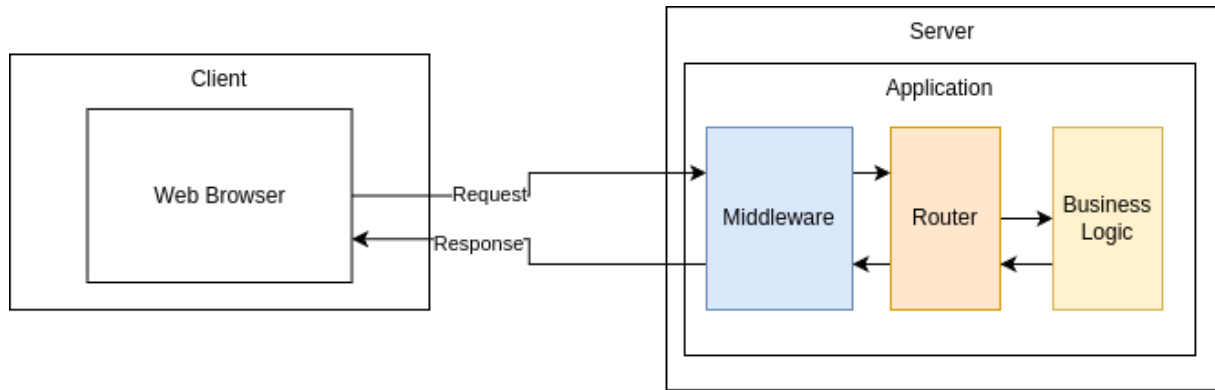
Σχήμα 1. Επικοινωνία ανάμεσα σε πελάτη και εξυπηρέτη. Οι Routers δέχονται και απαντούν σε HTTP αιτήσεις.

### 2.3.3.1.2 Middleware

Τα middleware είναι συναρτήσεις που έχουν ως σκοπό να φιλτράρουν την πληροφορία προτού η τελευταία φτάσει στους Routers. τα middleware λειτουργούν σε διασωλήνωση αφού το ένα middleware περάσει διαλέγει αν θα στείλει την πληροφορία στο επόμενο ή αν θα μπλοκάρει την αίτηση επιστρέφοντας μήνυμα σφάλματος με τον αντίστοιχο κωδικό. Γι Αυτό έχει πολύ μεγάλη σημασία, η σειρά με την οποία θα δηλώσουμε τα φίλτρα εντός του λογισμικού [18].

Τα middleware χρησιμοποιούνται σε πολλές περιπτώσεις χρήσης. Αρχικά για την αυθεντικοποίηση και εξουσιοδότηση. Στο εσωτερικό της συνάρτησης των middleware μπορεί να γίνεται έλεγχος αν το token ή το session id του χρήστη που επιθυμεί να αυθεντικοποιηθεί είναι έγκυρο. Αυτό είναι σημαντικό για την διατήρηση της αρχής της διαχωρισμένης ανησυχίας (separation of concerns), έχοντας όλη την λογική αυθεντικοποίησης στα middleware, σε περίπτωση που γίνει αλλαγή στον τρόπο αυθεντικοποίησης δεν θα χρειαστεί να γίνουν αλλαγές στους δρομολογητές ή στα υπόλοιπα στρώματα του κώδικα [18].

Τα middleware μπορούν να χρησιμοποιηθούν και για να περιορίσουν τους ρυθμούς αιτήσεων από πελάτες που έχουν ως κύριο σκοπό να βλάψουν το σύστημα. Σε περίπτωση που κάποιος χρήστης έχει δημιουργήσει ένα λογισμικό που στέλνει πολλές HTTP αιτήσεις το δευτερόλεπτο με απώτερο σκοπό να μπλοκάρει κάποιο σύστημα, τα middleware χρησιμοποιώντας συγκεκριμένους αλγόριθμους όπως για παράδειγμα Token Bucket Algorithm, Fixed Window Counter ή Sliding Window Log, μπορούν να αποφασίσουν αν θα αφήσουν την αίτηση να περάσει ή όχι στους δρομολογητές [18].



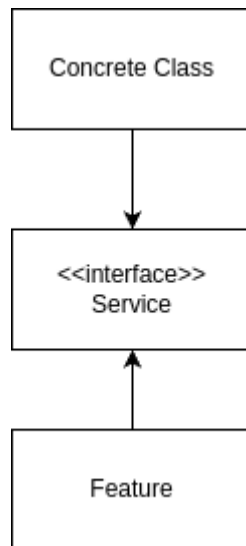
Σχήμα 2. Επικοινωνία ανάμεσα σε πελάτη και επωηρέτη. Τα middleware φιλτράρουν τα δεδομένα προτού φτάσουν στους Routers

### 2.3.3.1.3 Dependency Injection

Το Dependency Injection, είναι ένα σχεδιαστικό πρότυπο που επιτρέπει στους προγραμματιστές να κάνουν κλάσεις που να μην έχουν άμεση εξάρτηση με από άλλες κλάσεις, αλλά οι υλοποιήσεις να περνιούνται δυναμικά κατά την εκτέλεση του προγράμματος. Σε γενικές γραμμές, πρόκειται για μια τεχνική όπου ένα αντικείμενο παρέχει τις εξαρτήσεις ενός άλλου αντικειμένου, αντί το ίδιο το αντικείμενο να δημιουργεί αυτές τις εξαρτήσεις. Οι προγραμματιστές του εκάστοτε λογισμικού δηλώνουν ποιες υλοποιήσεις θα περνιούνται παραμετρικά [19]. Η έννοια του Dependency Injection σχετίζεται στενά με το την αρχή Dependency Inversion Principle η οποία αντιπροσωπεύει το γράμμα D από τις SOLID αρχές [20]. Με το Dependency Injection μπορούμε να επιτύχουμε το Dependency Inversion Principle. Όπως αναφέρει ο Uncle Bob στον βιβλίο Clean Architecture “**Entities must depend on abstractions, not on concretions. It states that the high-level module must not depend on the low-level module, but they should depend on abstractions.**” [21].

Χάρη στο Dependency Injection μπορούμε να έχουμε πιο ευέλικτο, αναβαθμίσιμο και ελέγξιμο κώδικα. Οι κλάσεις που ενθυλακώνουν όλη την επιχειρησιακή λογική, δεν είναι απαραίτητο να έχουν άμεση αναφορά στην πηγή πληροφοριών που αντλούν τα δεδομένα, αλλά δηλώνουν, ότι θα δέχονται παραμετρικά την υλοποίηση. Αυτό έχει ως πλεονέκτημα ότι μπορεί να γίνει έλεγχος της επιχειρησιακής λογικής χωρίς να υπάρχει εξάρτηση από την βάση δεδομένων. Κατά την διάρκεια εκτέλεσης των δοκιμών, το framework θα αναλάβει την ευθύνη να μεταβιβάσει παραμετρικά στη δοκιμή την ψευδή βάση δεδομένων [19].

Τα περισσότερα λογισμικά χρησιμοποιούν κάποιες τρίτες υπηρεσίες οι οποίες όμως έχουν νόημα να εκτελούνται μόνο στο Production και όχι κατά την διάρκεια της ανάπτυξης λογισμικού. Με το DI, μπορούμε να δηλώσουμε ποιες υλοποιήσεις των κλάσεων να στέλνονται δυναμικά στο Production και ποιες στο Development. Με αυτό τον τρόπο ο οργανισμός που αναπτύσσει το συγκεκριμένο λογισμικό, δεν χρειάζεται να έχει διαφορετικό κώδικα για το development και για το production. Για παράδειγμα αν αυτή η τρίτη υπηρεσία είναι παρέχει υπηρεσίες ηλεκτρονικού ταχυδρομείου, ο οργανισμός εξοικονομεί πόρους επειδή τα ηλεκτρονικά μηνύματα δεν θα στέλνονται κατά την διάρκεια ανάπτυξης λογισμικού αλλά μόνο στο λογισμικό παραγωγής [19].



Σχήμα 3. Γραφική απεικόνιση του σχεδιαστικού προτύπου **Dependency Injection**

#### 2.3.3.1.4 ORM

Το ORM (Object Relational Mapping) είναι μια τεχνική προγραμματισμού που αντιστοιχεί με αντικειμενοστραφή τρόπο τα δεδομένα που υπάρχουν εντός κάποιας σχεσιακής ή μη σχεσιακής βάσης δεδομένων. Ουσιαστικά δημιουργείται ένα επίπεδο αφαίρεσης ανάμεσα στην εφαρμογή και στην βάση δεδομένων. Για κάθε πίνακα αν πρόκειται για σχεσιακή βάση ή έγγραφο αν πρόκειται για μη σχεσιακή βάση δεδομένων υπάρχει και ένα αντίστοιχο αντικείμενο που αντιπροσωπεύει την δομή του και αυτό γίνεται για να υπάρχει αλληλεπίδραση με τις αντίστοιχες γραμμές των πινάκων της βάσης δεδομένων ή των αρχείων. Αν και το ίδιο το ORM δεν θα μπορούσε να χαρακτηριστεί ως σχεδιαστικό πρότυπο, συχνά χρησιμοποιεί εσωτερικά διάφορα σχεδιαστικά πρότυπα. Για παράδειγμα, στα ORM χρησιμοποιείται συνήθως το σχεδιαστικό πρότυπο Active Record [22].

Ένας σημαντικός παράγοντας που ευνοεί τη χρήση ORM σε μία διαδικτυακή εφαρμογή είναι η απουσία αμεσότερης σύνδεσης μεταξύ της εφαρμογής και της βάσης δεδομένων. Αυτό σημαίνει ότι, σε περίπτωση που χρειαστεί να γίνει αλλαγή της βάσης δεδομένων ο κώδικας που αφορά την επιχειρησιακή λογική θα παραμείνει καθώς έχει, οι αλλαγές περιορίζονται στις ρυθμίσεις της εφαρμογής και στους οδηγούς, ενώ ο πυρήνας του κώδικα παραμένει αναλλοίωτος. Επιπλέον, η χρήση του ORM απαλλάσσει τους προγραμματιστές από την ανάγκη να είναι εξοικειωμένοι με τη γλώσσα προγραμματισμού SQL, καθώς το ORM αναλαμβάνει τη μετάφραση των εντολών της εφαρμογής σε SQL και την εκτέλεσή τους στη βάση δεδομένων [22].

#### 2.3.3.1.5 Seeders

Οι seeders είναι κλάσεις που είναι υπεύθυνες για την δημιουργία αρχικών δεδομένων στην βάση που χρησιμοποιείται από την εφαρμογή. Το seeding μπορεί να χρησιμοποιηθεί για πολλούς σκοπούς, στην συνέχεια θα απολαύσουμε τους δύο από αυτούς. Αρχικά η δημιουργία αρχικών δεδομένων, πολλές εφαρμογές για να μπορέσουν να λειτουργήσουν χρειάζονται κάποια αρχικά δεδομένα τα οποία είναι και στατικά, δηλαδή δέχονται πολύ λίγες τροποποιήσεις με την πάροδο του χρόνου. Για το πληροφοριακό σύστημα που αναπτύχθηκε για την OMEA, το seeding χρησιμοποιήθηκε για την εισαγωγή στατικών δεδομένων όπως για παράδειγμα τα πανεπιστήμια, τμήματα, διδάσκοντες και μαθήματα. Επιπλέον, το seeding μπορεί να χρησιμοποιηθεί για την δημιουργία τυχαίων δεδομένων,

αυτό το σενάριο είναι επιθυμητό όταν χρειάζεται να γίνει έλεγχος της εφαρμογής για το πως ανταπεξέρχεται με μεγάλου όγκου δεδομένα [23].

#### 2.3.3.1.6 CLI

Το CLI (Command Line Interface) είναι ένα εργαλείο που επιτρέπει στους προγραμματιστές να αλληλοεπιδρούν με το σύστημα μέσω εντολών του τερματικού. Παρά το γεγονός ότι δεν είναι άμεσα κομμάτι του framework και ότι πολλά framework μπορεί να μην το υποστηρίζουν επειδή βασίζονται σε αρχεία με ρυθμίσεις, προσφέρουν πολύ σημαντικό αντίκτυπο στο συνολικό οικοσύστημα. Το CLI αναλαμβάνει την αυτοματοποίηση εκείνων των εργασιών που είναι αυστηρά προκαθορισμένες και δεν εμπλέκεται λογική για την πραγματοποίησή τους, όπως για παράδειγμα η δημιουργία συγκεκριμένων κλάσεων ή το scaffolding του σχήματος μια σχεσιακής βάσης δεδομένων. Η παρούσα λειτουργία έχει θετικό αντίκτυπο, καθώς ελευθερώνει τον χρόνο των προγραμματιστών από χρονοβόρες διαδικασίες και επαναλαμβανόμενες εργασίες, επιτρέποντάς τους να επικεντρώνονται σε κρίσιμες και εποικοδομητικές εργασίες [24].

#### 2.3.3.2 Web Frameworks

Παρακάτω παρουσιάζονται και αναλύονται τρία δημοφιλή web frameworks, το Flask, η ASP.NET CORE και η Laravel. Στην συνέχεια θα τεκμηριωθεί η απόφαση επιλογής framework σύμφωνα με τις ανάγκες του πληροφοριακού συστήματος της OMEA.

##### 2.3.3.2.1 Flask

Το Flask είναι ένα web framework που έχει γραφτεί για την γλώσσα προγραμματισμού Python. Δημιουργήθηκε από τον προγραμματιστή Armin Ronacher το 2010. Ο λόγος που ο Armin δημιούργησε το Flask είναι επειδή θεωρούσε ότι δεν υπήρχε κάποια βιβλιοθήκη στην γλώσσα προγραμματισμού Python που να είναι πολύ μινιμαλιστική και να περιέχει μόνο όσα χρειάζονται οι προγραμματιστές για να υλοποιήσουν web apis. Ένα από τα σημαντικά πλεονεκτήματα που προσφέρει το Flask framework είναι ότι δεν απαιτεί πολλούς υπολογιστικούς πόρους για να λειτουργήσει σωστά. Επιπλέον, ένα αξιόλογο πλεονέκτημα του Flask είναι ότι λειτουργεί για την γλώσσα προγραμματισμού Python. Η Python έχει ένα πλούσιο οικοσύστημα βιβλιοθηκών που προσφέρουν λύσεις για μια ευρεία γκάμα από καταστάσεις, από βασικές λειτουργίες όπως η δημιουργία μοντέλων μηχανικής μάθησης και η ανάλυση σύνθετων δεδομένων. Αυτές οι βιβλιοθήκες που αναφέρθηκαν μπορούν να ενσωματωθούν άνετα σε έργα Flask μέσω της εντολής pip του τερματικού. Θα μπορούσε να θεωρηθεί μόνο ως μειονέκτημα ότι η Python, λόγω ότι είναι interpreted γλώσσα προγραμματισμού μπορεί να εμφανίζει καθυστερήσεις συγκριτικά με άλλες γλώσσες προγραμματισμού που είναι compiled. Προφανώς Αυτό δεν έχει να κάνει μόνο με την γλώσσα προγραμματισμού αλλά και με την πολυπλοκότητα του κώδικα [25].

##### 2.3.3.2.2 ASP.NET CORE

Το ASP.NET CORE είναι ένα web framework που δημιουργήθηκε από την εταιρεία microsoft. Το framework έχει μια πλούσια ιστορία που ξεκινάει από την δεκαετία του 90. Γενικά το.NET ως framework έκανε τη πρώτη εμφάνιση το 2003, λόγο ότι ακόμα ήταν στα σπάργανα ήταν πάρα πολύ απλοϊκό, παρόλο αυτά έφερε έναν καινούργιο τρόπο ανάπτυξης λογισμικών για διαφορετικού είδους πλατφόρμες όπως για παράδειγμα desktop εφαρμογές ή server εφαρμογές. Το.NET ως framework είναι κλειστού λογισμικού, όμως το 2016 η microsoft δημοσίευσε το .NET Core που ουσιαστικά είναι μια ανοιχτού έκδοση της.NET. Ο λόγος που πήρε η microsoft την απόφαση να εκδώσει ανοιχτού κώδικα προϊόν, πάρα το γεγονός ότι έρχεται σε αντίφαση με τις πολιτικές πεποιθήσεις της εταιρίας ήταν επειδή

διαισθάνθηκαν την ανάγκη ύπαρξης ενός πιο βέλτιστου framework το οποίο θα μπορεί να λειτουργεί σε διάφορα λειτουργικά συστήματα όπως για παράδειγμα Windows, macOS και Linux [26].

Στο ASP.NET CORE Framework είναι από τα λίγα frameworks που δίνουν την ευελιξία στους προγραμματιστές να αποφασίσουν ποια γλώσσα προγραμματισμού θα χρησιμοποιήσουν. Η κύρια γλώσσα προγραμματισμού που χρησιμοποιείται είναι η C#, είναι compiled αντικειμενοστραφής γλώσσα προγραμματισμού, η σύνταξη είναι παρόμοια με την Java καθώς συνδυάζει πολλές πτυχές της γλώσσας προγραμματισμού C++. Το Framework υποστηρίζει επίσης τις γλώσσες προγραμματισμού Visual Basic, F# και IronPython. Ανάλογα με τις ανάγκες του εκάστοτε έργου οι προγραμματιστές επιλέγουν την γλώσσα προγραμματισμού που θα έχει την καλύτερη απόδοση [26].

### 2.3.3.2.3 Laravel

Η Laravel είναι web framework ανοιχτού λογισμικού για την γλώσσα προγραμματισμού PHP. Δημιουργήθηκε από τον προγραμματιστή Taylor Otwell, ο λόγος που ο δημιουργός του ξεκίνησε να αναπτύσσει το παρών framework ήταν επειδή αισθανόταν την έλλειψη εναλλακτικών frameworks για την γλώσσα προγραμματισμού PHP. Πιο συγκεκριμένα είχε ως κίνητρο την δημιουργία framework που θα μπορούσε να αντικαταστήσει το Codeigniter επειδή ήταν αρκετά ελλιπές. Το ντεμπούτο της πρώτης έκδοσης ήταν τον Ιούνιο του 2011, όμως όπως γίνεται και με τα περισσότερα frameworks, κυκλοφόρησαν πολλές εκδόσεις με την πάροδο του χρόνου. Πλέον η Laravel χρησιμοποιείται και σε λογισμικά μεγάλης κλίμακας όπως για παράδειγμα στο οικοσύστημα της livestreaming πλατφόρμα twitch, στην εφημερίδα The New York Times και σε εσωτερικά έργα της εταιρείας Disney. Ο λόγος που συχνά προτιμάται ως framework είναι επειδή χαρακτηρίζεται ως Προοδευτικό και Επεκτάσιμο. Προοδευτικό επειδή έχει πολύ καλογραμμένο και συντηρημένο το documentation καθώς υπάρχει και πληθώρα από εκπαιδευτικά βίντεο στο διαδίκτυο για το πως μπορούν οι προγραμματιστές να το χρησιμοποιήσουν και να επωφεληθούν πλήρως από αυτό. Τέλος, επεκτάσιμο επειδή μπορεί να ανταπεξέλθει το ίδιο αποδοτικά και αποτελεσματικά τόσο σε μικρά έργα όσο και σε λογισμικά που έχουν σημαντικές απαιτήσεις και δέχονται χιλιάδες αιτήσεις το δευτερόλεπτο [27].

### 2.3.3.2.4 Λόγοι που επιλέχθηκε η Flask

Για την υλοποίηση του πληροφοριακού συστήματος OMEA, ήταν σημαντική η απόφαση επιλογής της κατάλληλης back-end τεχνολογίας που θα μπορούσε να εξυπηρετήσει τόσο τις λειτουργικές απαιτήσεις όσο και τις μη λειτουργικές απαιτήσεις. Γενικά, όπως ισχύει για το front-end, το ίδιο ισχύει και για το back-end. Η επιλογή ενός framework επηρεάζεται κυρίως από τις μη λειτουργικές απαιτήσεις παρά από τις λειτουργικές απαιτήσεις του συγκεκριμένου λογισμικού που αναπτύσσεται. Αυτό συμβαίνει επειδή οι ανάγκες στα έργα λογισμικού είναι πανομοιότυπες και στα τελευταία εργάζονται οι ίδιοι προγραμματιστές που τα χρησιμοποιούν. Με αποτέλεσμα τα περισσότερα frameworks να λύνουν τα ίδια προβλήματα, με διαφορετικό τρόπο.

Παρακάτω παρουσιάζονται τα κριτήρια που εξετάστηκαν για την επιλογή του back-end Framework

1. **Τεχνογνωσία:** Από την στιγμή που τα περισσότερα frameworks μπορούν να λύσουν τα ίδια προβλήματα το πιο σημαντικό κριτήριο αξιολόγησης είναι η τεχνογνωσία των προγραμματιστών. Δηλαδή δίνεται σημαντική βαρύτητα στο frameworks που γνωρίζουν οι προγραμματιστές. Συμπεριλαμβάνονται υπόψη όχι μόνο η τεχνογνωσία της ομάδας που αναπτύσσει το εκάστοτε λογισμικό αλλά γενικά τι γνωρίζουν και προτιμούν οι προγραμματιστές της κοινότητας. Επειδή τα έργα λογισμικού δεν ολοκληρώνεται στην παράδοση τους αλλά συνεχίζουν να αναπτύσσονται, να προστίθενται λειτουργίες και να

διορθώνονται τυχόν σφάλματα που προέκυψαν κατά την διάρκεια ανάπτυξης του λογισμικού για όσο ο οργανισμός έχει συμφέροντα για το παρών έργο. Γι' αυτόν τον λόγο είναι ζωτικής σημασίας χαρακτηριστικό για τους οργανισμούς που δημιουργούν εφαρμογές να έχουν την δυνατότητα να αντικαταστούν εύκολα τους προγραμματιστές επειδή στον κλάδο της ανάπτυξης λογισμικού είναι συχνό φαινόμενο οι προγραμματιστές να αλλάζουν εταιρίες, αυτός ο όρος είναι γνωστός και ως Job Hobbing. Το Flask είναι πολύ γνωστή τεχνολογία με αποτέλεσμα να είναι εύκολη η διαδικασία εύρεσης προγραμματιστής που θα μπορεί να αναβαθμίσει το σύστημα.

2. **Απλότητα:** Ως απλότητα εννοείται η ευελιξία που διαθέτουν οι προγραμματιστές να προσθέτουν καινούργιες λειτουργίες σε λογισμικά που αναπτύσσονται ή σε ήδη αναπτυσσόμενα λογισμικά. Είναι πολύ σημαντικό κριτήριο αξιολόγησης πόσο μάλλον στα έργα λογισμικού που ακολουθούν ως μεθοδολογία το AGILE manifesto[28], με λίγα λόγια όταν δεν είναι ξεκάθαρες οι απαιτήσεις αλλά επιβεβαιώνονται παράλληλα με την ανάπτυξη του λογισμικού. Αυτό έχει ως αποτέλεσμα να υπάρχει πιθανότητα τροποποίησης λειτουργιών που δεν ήταν αναμενόμενες και αν το framework δεν μπορεί να ανταπεξέλθει θα χρειαστεί το λογισμικό να γραφτεί ξανά από την αρχή με συνέπεια να αργήσει να βγει στην παραγωγή. Το Flask αναδεικνύεται ως μία απλή και ευέλικτη τεχνολογία, που παρέχει στους προγραμματιστές τη δυνατότητα να προσθέτουν λειτουργίες με ευκολία.
3. **Βιβλιοθήκες:** Η ύπαρξη πληθώρας διαθέσιμων βιβλιοθηκών για το επιλεγμένο framework συνιστά σημαντικό πλεονέκτημα στη διαδικασία ανάπτυξης λογισμικού. Αυτό επιτρέπει στους προγραμματιστές να μην αναγκάζονται να δημιουργήσουν από την αρχή κάθε λειτουργία, καθώς τα περισσότερα κοινά προβλήματα έχουν ήδη επιλυθεί από άλλους προγραμματιστές. Οι λειτουργίες αυτές έχουν ενσωματωθεί σε βιβλιοθήκες που είναι διαθέσιμες για χρήση, εξοικονομώντας χρόνο και πόρους στη διαδικασία ανάπτυξης. Το οικοσύστημα του Flask, το οποίο βασίζεται στη γλώσσα προγραμματισμού Python, παρέχει μια πλούσια συλλογή από βιβλιοθήκες, κυρίως εστιάζοντας σε βιβλιοθήκες που αφορούν την επεξεργασία και ανάλυση δεδομένων. Οι βιβλιοθήκες ανάλυσης δεδομένων παρέχουν ζωτική συνδρομή, καθώς το πληροφοριακό σύστημα που αναπτύσσεται έχει ως κύριο στόχο την επεξεργασία δεδομένων [29].
4. **Υποστήριξη από την κοινότητα:** Είναι ζωτικής σημασίας το επιλεγμένο framework να έχει εκτενή υποστήριξη από την κοινότητα, καθώς κατά τη διάρκεια της ανάπτυξης λογισμικού συχνά προκύπτουν διάφορα προβλήματα που απαιτούν γνώση της εκάστοτε τεχνολογίας. Η ενεργή κοινότητα διασφαλίζει ότι υπάρχει πρόσβαση σε βοήθεια και υποστήριξη από άλλους έμπειρους προγραμματιστές, οι οποίοι μπορούν να συνεισφέρουν στην αντιμετώπιση των προβλημάτων. Επιπλέον, η χρήση λογισμικών που είναι δημοφιλή στην κοινότητα προσφέρει την πλεονεκτική δυνατότητα πρόσβασης σε καθαρές τεκμηριώσεις κώδικα και πληθώρα εκπαιδευτικού υλικού, όπως βίντεο tutorials σε δημοφιλείς πλατφόρμες όπως το YouTube. Το Flask καταγράφει 66.2 χιλιάδες αστέρια στο GitHub και διαθέτει πολλά blog και tutorials στο διαδίκτυο που εξηγούν τη λειτουργία του framework. Αυτά τα γεγονότα υποδεικνύουν ότι το Flask έχει μια ενεργή κοινότητα με πολλούς χρήστες[30].
5. **Επιδόσεις:** Τέλος, αλλά εξίσου σημαντικό, το επιλεγμένο framework πρέπει να είναι ικανό να λειτουργεί με γρήγορο και αποδοτικό τρόπο. Δεν υπάρχει ιδανικό framework που να επιλύει όλα τα προβλήματα με ταχύτητα, αλλά υπάρχουν επιλογές που είναι πιο κατάλληλες για συγκεκριμένα είδη προβλημάτων. Για την ανάπτυξη του πληροφοριακού συστήματος OMEA, η βάση δεδομένων είναι το πιο σημαντικό σύστημα και εκεί γίνονται οι περισσότεροι δύσκολοι

## Κεφάλαιο 2

υπολογισμοί οπότε οι επιδόσεις του framework δεν επηρεάζουν τόσο πολύ τις επιδόσεις του συστήματος.

## Κεφάλαιο 3ο: Σχεδιασμός συστήματος

### 3.1 Εισαγωγή

Το συγκεκριμένο κεφάλαιο, σχετίζεται με τον τομέα την μηχανικής λογισμικού, πιο συγκεκριμένα γίνεται ανάλυση των μεθοδολογιών που χρησιμοποιήθηκαν για την σχεδίαση του λογισμικού σε αφηρημένο επίπεδο. Αρχικά πραγματοποιείται ανάλυση σε επίπεδο context το οποίο αναφέρει ποιοι είναι οι χρήστες του συστήματος και ποια είναι τα εξωτερικά συστήματα που αναπτύσσονται από την εσωτερική ομάδα ανάπτυξης λογισμικού, Στην συνέχεια παρουσιάζονται οι λειτουργικές απαιτήσεις, μη λειτουργικές απαιτήσεις, οι περιορισμοί και αρχές που εφαρμόστηκαν για την ανάπτυξη του πληροφοριακού συστήματος OMEA. Το συγκεκριμένο κεφάλαιο αναφέρεται στο βιβλίο [31].

### 3.2 Περιεχόμενο

Το έργο λογισμικού το οποίο υλοποιήθηκε έχει ως κύριο σκοπό να αυτοματοποιήσει την διαδικασία εισαγωγής δεδομένων σε μια σχεσιακή βάση ώστε μετέπειτα να είναι εύκολη η εξαγωγή στατιστικών δεδομένων μέσω γραφικής διεπαφής. Το κύριο πλεονέκτημα της εφαρμογής είναι η εξοικονόμηση χρόνου για τους εκπαιδευτικούς, καθώς δεν απαιτείται η διαχείριση καθηκόντων που μπορούν να αυτοματοποιηθούν. Επιπλέον, οι εκπαιδευτικοί μπορούν να επικεντρωθούν σε ζητήματα που είναι ζωτικής σημασίας για την διδασκαλία, ενώ έχουν την εγγύηση της ορθής καταχώρησης των δεδομένων, προσφέροντας έτσι ακρίβεια και αξιοπιστία στα στατιστικά στοιχεία που εξάγουν. Χωρίς την ύπαρξη του πληροφοριακού συστήματος OMEA, οι διδάσκοντες θα έπρεπε είτε να εξάγουν τα στατιστικά απευθείας μέσω xlsx κάτι το οποίο δεν είναι καθόλου πρακτικό επειδή τα δεδομένα εσωτερικά στα αρχεία δεν έχουν την κατάλληλη δομή για να μπορούν να εξαχθούν εύκολα στατιστικά στοιχεία. Το πληροφοριακό σύστημα OMEA είναι διαδικτυακή εφαρμογή, αυτό το χαρακτηριστικό είναι πολύ σημαντικό επειδή μπορούν να έχουν πρόσβαση όλοι οι διδάσκοντες ανεξάρτητα από την συσκευή και το λειτουργικό σύστημα που κατέχουν.

Οι χρήστες του πληροφοριακού συστήματος OMEA διαιρούνται σε δύο κατηγορίες χρηστών. Οι διδάσκοντες, οι οποίοι χρησιμοποιούν το σύστημα με κύριο στόχο την πρόσβαση στις αξιολογήσεις των φοιτητών ώστε με αυτές τις πληροφορίες να μπορέσουν να βελτιώσουν τυχόν ελλείψεις στον τρόπο διδασκαλίας τους. Οι διδάσκοντες που είναι μέλη στην ομάδα εσωτερικής αξιολόγησης του τμήματος, θα έχουν τον επιπλέον ρόλο των διαχειριστών δηλαδή θα μπορούν να τροποποιήσουν τα εσωτερικά δεδομένα της εφαρμογής όπως για παράδειγμα την προσθήκη κάποιου νέου διδάσκοντα καθώς και να ρυθμίσουν τυχόν λειτουργίες όπως για παράδειγμα την προσθήκη νέου ακαδημαϊκού εξαμήνου.

### 3.3 Λειτουργικές Απαιτήσεις

Όπως αναφέρθηκε στην ενότητα Περιεχόμενο, υπάρχουν δύο κατηγορίες χρηστών. Οι διδάσκοντες και οι διδάσκοντες που είναι μέλη της OMEA θα αναφέρονται και ως διαχειριστές του συστήματος. Όπως είναι εύκολα αντιληπτό οι διαχειριστές κληρονομούν όλες τις λειτουργικές απαιτήσεις των διδασκόντων καθώς μπορούν να κάνουν επιπλέον διαχειριστικές λειτουργίες.

Κάθε λειτουργική απαίτηση έχει έναν βαθμό αναγκαιότητας, δηλαδή πόσο σημαντική είναι ύπαρξή της για να μπορέσει να λειτουργήσει ομαλά το σύστημα. Ο βαθμός αναγκαιότητας υπολογίστηκε με την μέθοδο MoSCoW method [32]. Η συγκεκριμένη μέθοδος χωρίζει τις λειτουργικές απαιτήσεις σε τέσσερις κατηγορίες.

1. **Must Have:** Η λειτουργική απαίτηση που καθιστά το χαρακτηρισμό "Must Have" υποδεικνύει ότι η λειτουργία αυτή είναι απαραίτητη για την ομαλή λειτουργία του συστήματος. Χωρίς αυτήν, το σύστημα θα είναι ατελές και η χρήση του θα είναι ανεπαρκής. Για παράδειγμα τέτοιου είδους λειτουργία όπως θα αναλυθεί και στην συνέχεια είναι η εισαγωγή των δεδομένων. Αν δεν υπάρχει η λειτουργία εισαγωγής δεδομένων το σύστημα στο γενικό του σύνολο δεν μπορεί να εξυπηρετήσει τις επιχειρησιακές ανάγκες του έργου.[32].
2. **Should Have:** Η λειτουργική απαίτηση που καθιστά το χαρακτηρισμό "Should Have" υποδεικνύει ότι η λειτουργία αυτή είναι πολύ σημαντική για το σύστημα αλλά δεν θα μπορούσε να χαρακτηριστεί και ως απαίτηση. Χωρίς την συγκεκριμένη λειτουργία το σύστημα μπορεί να εξυπηρετήσει τις επιχειρησιακές του ανάγκες όμως δεν θα είναι καθόλου πρακτικό στην χρήση του. Όπως θα αναλύσουμε και στην συνέχεια τέτοιου είδους λειτουργία είναι τα φίλτρα, χωρίς αυτά το σύστημα εξυπηρετεί τις ανάγκες του, όμως δεν είναι καθόλου πρακτικό επειδή οι διδάσκοντες δεν θα έχουν την δυνατότητα να φτάνουν σε σύντομο χρονικό διάστημα στην πληροφορία που αναζητούν [32].
3. **Could Have:** Η λειτουργική απαίτηση που καθιστά το χαρακτηρισμό "Could Have" υποδεικνύει ότι η λειτουργία αυτή θα ήταν καλό να υλοποιηθεί για το σύστημα χωρίς όμως να μπορούσε να χαρακτηριστεί ως απαραίτητη ή πολύ σημαντική για το σύστημα. Όπως θα αναλύσουμε και στην συνέχεια τέτοιου είδους λειτουργία είναι ύπαρξη δια δραστηκών διαγραμμάτων εντός της γραφικής διεπαφής. Βοηθούν πολύ άμα υπάρχουν αλλά δεν είναι αναγκαία ότι πάρα πολύ σημαντικά επειδή οι διδάσκοντες μπορούν να εξάγουν τα συμπεράσματα τους και μόνο με την χρήση αριθμών [32].
4. **Would Have:** Η λειτουργική απαίτηση που καθιστά το χαρακτηρισμό "Would Have" υποδεικνύει ότι η λειτουργία αυτή δεν προσφέρει μεγάλη αξία στο σύστημα συνεπώς θα μπορούσε και να μην υλοποιηθεί ή να υλοποιηθεί στο μέλλον σε μελλοντική αναβάθμιση του συστήματος. Όπως θα αναλύσουμε και στην συνέχεια τέτοιου είδους λειτουργία θα μπορούσε να είναι η εξαγωγή δεδομένων στην μορφή δεδομένων xlsx μέσω της διαδικτυακής γραφικής διεπαφής [32].

Παρακάτω παρουσιάζονται οι λειτουργικές απαιτήσεις του πληροφοριακού συστήματος σε πίνακες. Κάθε πίνακας αναγράφει τον χρήστη που μπορεί να κάνει την συγκεκριμένη λειτουργία, ποιες είναι οι προϋποθέσεις για να μπορέσει να πραγματοποιηθεί η συγκεκριμένη λειτουργία και ποια θα είναι τα αποτελέσματα δηλαδή τι θα έχει αλλάξει στο σύστημα όταν ολοκληρωθεί επιτυχώς η εκάστοτε λειτουργία. Επιπλέον περιγράφεται με σύντομο τρόπο η ροή της λειτουργικής απαίτησης.

Όνομα περίπτωσης χρήσης	Φόρτωση δεδομένων
Χρήστης	Διδάσκων
Περιγραφή	Κάθε διδάσκων πρέπει να έχει την δυνατότητα να μπορεί να φορτώσει αρχεία xlsx στο σύστημα.
Προϋποθέσεις	Το xlsx αρχείο που πρόκειται να φορτωθεί να είναι σε έγκυρη μορφή.
Μετασυνθήκες	Τα δεδομένα έχουν φορτωθεί στο σύστημα.
Προτεραιότητα	Must

Όνομα περίπτωσης χρήσης	Έλεγχος ορθότητας xlsx αρχείου
-------------------------	--------------------------------

Χρήστης	Διδάσκων
Περιγραφή	Κάθε διδάσκων πρέπει να έχει την δυνατότητα να μπορεί να ελέγξει αν το xlsx αρχείο που πρόκειται να αναρτηθεί ότι είναι σε έγκυρη μορφή.
Προϋποθέσεις	Ο διδάσκων να έχει στην διάθεση του κάποιο xlsx αρχείο.
Μετασυνθήκες	Εμφανίζονται στον χρήστη αν είναι σωστά δομημένο το αρχείο.
Προτεραιότητα	Should

Όνομα περίπτωσης χρήσης	Έλεγχος λαθών xlsx αρχείου
Χρήστης	Διδάσκων
Περιγραφή	Κάθε διδάσκων πρέπει να έχει την δυνατότητα να μπορεί να ελέγξει ποια είναι τα σφάλματα που έχει το xlsx αρχείο που πρόκειται να αναρτηθεί.
Προϋποθέσεις	Ο διδάσκων να έχει στην διάθεση του κάποιο xlsx αρχείο.
Μετασυνθήκες	Εμφανίζονται στον χρήστη τα τυχόν σφάλματα που έχει το xlsx αρχείο.
Προτεραιότητα	Should

Όνομα περίπτωσης χρήσης	Προβολή μαθημάτων
Χρήστης	Διδάσκων
Περιγραφή	Κάθε διδάσκων πρέπει να έχει την δυνατότητα να μπορεί να βλέπει ποια μαθήματα υπάρχουν καταχωρημένα στο σύστημα.
Προϋποθέσεις	Να έχει εισαχθεί στο σύστημα τουλάχιστον ένα ερωτηματολόγιο.
Μετασυνθήκες	Εμφανίζονται στον διδάσκοντα τα μαθήματα που υπάρχουν καταχωρημένα στο σύστημα.
Προτεραιότητα	Must

Όνομα περίπτωσης χρήσης	Προβολή Στατιστικά μαθημάτων
Χρήστης	Διδάσκων

Κεφάλαιο 3

Περιγραφή	Κάθε διδάσκων πρέπει να έχει την δυνατότητα να μπορεί να βλέπει τους μερους όρους από τα μαθήματα που υπάρχουν καταχωρημένα στο σύστημα.
Προϋποθέσεις	Να έχει εισαχθεί στο σύστημα τουλάχιστον ένα ερωτηματολόγιο.
Μετασυνθήκες	Εμφανίζονται στον διδάσκοντα οι μέση όροι από τα μαθήματα που υπάρχουν καταχωρημένα στο σύστημα
Προτεραιότητα	Must

Όνομα περίπτωσης χρήσης	Φιλτράρισμα μαθημάτων
Χρήστης	Διδάσκων
Περιγραφή	Κάθε διδάσκων πρέπει να έχει την δυνατότητα να μπορεί να ρυθμίσει τα φίλτρα ώστε να του εμφανίζονται μόνο τα μαθήματα που επιθυμεί.
Προϋποθέσεις	Να έχει εισαχθεί στο σύστημα τουλάχιστον ένα ερωτηματολόγιο.
Μετασυνθήκες	Εμφανίζονται στον διδάσκοντα μόνο τα μαθήματα που τηρούν τις προϋποθέσεις των φίλτρων.
Προτεραιότητα	Should

Όνομα περίπτωσης χρήσης	Προβολή ερωτήσεων μαθημάτων
Χρήστης	Διδάσκων
Περιγραφή	Κάθε διδάσκων πρέπει να έχει την δυνατότητα να μπορεί να βλέπει τις ερωτήσεις των ερωτηματολογίων ενός συγκεκριμένου μαθήματος σε συνδυασμό με τον τρέχον μέσο όρο της κάθε ερώτησης και τον γενικό μέσο όρο της ερώτησης.
Προϋποθέσεις	Να έχει εισαχθεί στο σύστημα τουλάχιστον ένα ερωτηματολόγιο.
Μετασυνθήκες	Εμφανίζονται στον διδάσκοντα οι ερωτήσεις από το μάθημα που επιλέχθηκε σε συνδυασμό με τα στατιστικά τους στοιχεία.
Προτεραιότητα	Must

Όνομα περίπτωσης χρήσης	Προβολή πλήθος φοιτητών
Χρήστης	Διδάσκων
Περιγραφή	Κάθε διδάσκων πρέπει να έχει την δυνατότητα να μπορεί να βλέπει το πλήθος των φοιτητών που συμμετείχαν στην διαδικασία αξιολόγησης ενός μαθήματος.
Προϋποθέσεις	Να έχει εισαχθεί στο σύστημα τουλάχιστον ένα ερωτηματολόγιο.
Μετασυνθήκες	Εμφανίζονται στον διδάσκοντα το πλήθος φοιτητών που συμμετείχαν στο ερωτηματολόγιο.
Προτεραιότητα	Should

Όνομα περίπτωσης χρήσης	Προβολή διδασκόντων μαθήματος
Χρήστης	Διδάσκων
Περιγραφή	Κάθε διδάσκων πρέπει να έχει την δυνατότητα να μπορεί να βλέπει ποιοι διδάσκοντες διεξήγαγαν ένα συγκεκριμένο μάθημα.
Προϋποθέσεις	Να έχει εισαχθεί στο σύστημα τουλάχιστον ένα ερωτηματολόγιο.
Μετασυνθήκες	Εμφανίζονται στον διδάσκοντα οι διδάσκοντες που διεξήγαγαν το μάθημα.
Προτεραιότητα	Should

Όνομα περίπτωσης χρήσης	Ανανέωση εξαμήνου
Χρήστης	μέλος OMEA
Περιγραφή	Κάθε μέλος OMEA πρέπει να έχει την δυνατότητα να μπορεί να προσθέτει εξάμηνα στο σύστημα.
Προϋποθέσεις	-
Μετασυνθήκες	Έχει καταχωρηθεί στο σύστημα ένα επιπλέον εξάμηνο.
Προτεραιότητα	Must

### 3.4 Μη Λειτουργικές Απαιτήσεις

Οι μη λειτουργικές απαιτήσεις, είναι απαιτήσεις που δεν σχετίζονται με κάποια ενέργεια αλλά έχουν να κάνουν με περιορισμούς που πρέπει να τηρούνται ώστε το πληροφοριακό σύστημα να μπορεί να ανταπεξέλθει στις επιχειρησιακές ανάγκες που έθεσαν οι ενδιαφερόμενοι του έργου. Οι μη λειτουργικές απαιτήσεις είναι τόσο σημαντικές όσο και οι λειτουργικές απαιτήσεις για αυτό πρέπει να επιλέγονται και να διαχειρίζονται με μεγάλη προσοχή. Σύμφωνα με τις μη λειτουργικές απαιτήσεις παίρνονται αποφάσεις για την αρχιτεκτονική και τις τεχνολογίες λογισμικού που θα χρησιμοποιηθούν για το εκάστοτε έργο που πρόκειται να υλοποιηθεί. Λάθος αποφάσεις στις μη λειτουργικές απαιτήσεις πιθανόν να οδηγήσουν το έργο λογισμικού σε δυσάρεστη κατάσταση.

Η επιχειρησιακές ανάγκες είναι αυτές που καθορίζουν τις μη λειτουργικές απαιτήσεις. Κάθε μη λειτουργική απαίτηση πρέπει να είναι μετρήσιμη και αυστηρά καθορισμένη ώστε να μην υπάρχει περιθώριο αμφιβολίας στους προγραμματιστές και αρχιτέκτονες που θα ασχοληθούν με την κατασκευή του λογισμικού. Επιπλέον, οι μη λειτουργικές απαιτήσεις για να έχουν αντίκτυπο πρέπει να είναι ρεαλιστικές, για παράδειγμα η μη λειτουργική απαίτηση το σύστημα να είναι διαθέσιμο για τους χρήστες 24/7 είναι πολύ δύσκολο έως ανέφικτο να επιτευχθεί.

Παρακάτω παρουσιάζονται οι μη λειτουργικές απαιτήσεις που έχουν σημαντική σημασία για το πληροφοριακό σύστημα ΟΜΕΑ. Επιπλέον, αναλύεται ο λόγος για τον οποίο αυτές οι μη λειτουργικές απαιτήσεις είναι κρίσιμες για την ορθή λειτουργία του συστήματος.

1. **Επιδόσεις:** Οι επιδόσεις αποτελούν ουσιαστικό παράγοντα για το πληροφοριακό σύστημα ΟΜΕΑ, καθώς οι διδάσκοντες χρειάζεται να έχουν γρήγορη και αποτελεσματική πρόσβαση στα επιθυμητά δεδομένα. Εντός της κεντρικής βάσης δεδομένων του συστήματος υπάρχει μεγάλος όγκος δεδομένων και πρέπει να διαχειρίζεται σωστά από τους προγραμματιστές για να μην υπάρχουν αυξημένες καθυστερήσεις.
2. **Ασφάλεια:** Τα δεδομένα που είναι αποθηκευμένα στην κεντρική βάση δεδομένων μπορούν να χαρακτηριστούν ως ευαίσθητα, καθώς αφορούν τις αξιολογήσεις των φοιτητών. Η παράδοση αυτών των δεδομένων σε μη εξουσιοδοτημένα άτομα μπορεί να προκαλέσει σοβαρή ζημία στη φήμη του τμήματος. Το πληροφοριακό σύστημα ΟΜΕΑ διασφαλίζει την ασφάλεια των δεδομένων με την χρήση firewall και VPN. Μόνο οι χρήστες που έχουν πρόσβαση στο εσωτερικό του τοπικού δικτύου μπορούν να συνδεθούν στην βάση δεδομένων. Το μοναδικό σφάλμα που θα μπορούσε να προκύψει είναι από την Flask που επικοινωνεί με τον έξω κόσμο, γι αυτό τον λόγο η Flask όταν λαμβάνει δεδομένα από τους χρήστες ελέγχει πρώτα ότι δεν έχουν σκοπό τα τελευταία να βλάψουν το σύστημα.
3. **Επεκτασιμότητα:** Η επεκτασιμότητα του συστήματος αποτελεί ουσιαστικό παράγοντα για το πληροφοριακό σύστημα ΟΜΕΑ. Το λογισμικό έχει σχεδιαστεί με τέτοιο τρόπο ώστε να μπορεί να φιλοξενήσει όλα τα ερωτηματολόγια από όλα τα πανεπιστήμια της Ελλάδας. Αυτό το γεγονός έχει ως αποτέλεσμα ότι αν εισαχθούν δεδομένα από πολλά πανεπιστήμια θα πρέπει το πληροφοριακό σύστημα να μπορεί να ανταπεξέλθει σε αρκετές αιτήσεις σε σύντομο χρονικό διάστημα.

### 3.5 Περιορισμοί

Κάθε λογισμικό για να μπορέσει να υλοποιηθεί πρέπει οι προγραμματιστές και αρχιτέκτονες που εμπλέκονται στην διαδικασία υλοποίησης να αντιμετωπίσουν τους περιορισμούς του αληθινού κόσμου. Οι περιορισμοί αν και αρνητική λέξη δεν έχουν πάντα αρνητική επίπτωση στην ανάπτυξη του λογισμικού, συνήθως ένα πρόβλημα έχει πάρα πολλές λύσεις που εξυπηρετούν τις ίδιες ανάγκες. Οι

περιορισμοί μπορεί να μειώσουν τις πιθανές λύσεις ώστε η προσοχή να επικεντρώνεται σε πολύ λιγότερες λύσεις με αποτέλεσμα οι αρχιτέκτονες λογισμικού να μπορούν να παίρνουν καλύτερες αποφάσεις. Στην συγκεκριμένη ενότητα θα αναλυθούν και συζητηθούν όλοι οι περιορισμοί που αντιμετωπίστηκαν και πως οι συγκεκριμένοι επηρέασαν την ανάπτυξη του πληροφοριακού συστήματος ΟΜΕΑ.

Ο πρώτος και πιο βασικός περιορισμός που αντιμετωπίστηκε είναι ο διαθέσιμος χρόνος για την ανάπτυξη του πληροφοριακού συστήματος ΟΜΕΑ. Το αναφερόμενο λογισμικό υλοποιήθηκε με αφορμή την διπλωματική εργασία η οποία έχει διάρκεια περίπου 5 μήνες. Λόγο περιορισμένου χρόνου δόθηκε μεγαλύτερη έμφαση στις απαραίτητες λειτουργικές απαιτήσεις που έχουν μεγαλύτερο αντίκτυπο σύμφωνα με τις επιχειρησιακές ανάγκες του έργου. Παρόλα αυτά λόγω της αυξημένης παραγωγικότητας ήταν εφικτή η υλοποίηση περισσότερων από τις βασικές λειτουργικές απαιτήσεις. Για όσον αφορά τις υπόλοιπες λειτουργικές απαιτήσεις, θα μπορέσουν να υλοποιηθούν από κάποιον άλλον συνάδελφο που αναλάβει ως διπλωματική το θέμα αναβάθμισης του συγκεκριμένου συστήματος.

Σημαντικός περιορισμός ήταν επίσης οι διαθέσιμες τεχνολογίες που μπορούσαν να χρησιμοποιηθούν για την υλοποίηση του συνολικού έργου. Λόγο ότι το συγκεκριμένο πληροφοριακό σύστημα υλοποιήθηκε με σκοπό να υπάρχει αρκετά χρόνια, έπρεπε να επιλεγθούν τεχνολογίες που χαρακτηρίζονται ως σταθερές ώστε να υπάρξει βεβαιότητα ότι το σύστημα θα συμπεριφέρεται ομαλά ακόμα και με πολλούς χρήστες και πολλά δεδομένα. Επιπλέον, εντός της λίστας των διαθέσιμων τεχνολογιών υπήρξαν τεχνολογίες που γνώριζε και η υπόλοιπη κοινότητα συναδέλφων ώστε να είναι πιο εύκολη η διαδικασία εύρεσης προγραμματιστών για την αναβάθμιση του πληροφοριακού συστήματος. Τέλος, το πληροφοριακό σύστημα έπρεπε να υλοποιηθεί μέσω της τεχνολογίας web δηλαδή να γίνει διαδικτυακή εφαρμογή, ώστε όλοι οι διδάσκοντες που το χρησιμοποιούν να μην χρειάζεται να το εγκαταστήσουν τοπικά στους υπολογιστές τους.

Τελευταίος προορισμός που αντιμετωπίστηκε ήταν η απαίτηση ότι το πληροφοριακό σύστημα ΟΜΕΑ έπρεπε να εκτελείται στον διακομιστή της ΟΜΕΑ. Ωστόσο, με τη χρήση του Docker, ήταν εφικτή η δημιουργία των αντίστοιχων εικονιδίων (images) μέσα στο αρχείο compose, με αποτέλεσμα το λογισμικό να εκτελείται στο περιβάλλον παραγωγής με ακριβώς τον ίδιο τρόπο όπως και στο περιβάλλον ανάπτυξης.

### 3.6 Αρχές

Στην συγκεκριμένη ενότητα παρουσιάζονται και αναλύονται όλες οι αρχές που χρησιμοποιήθηκαν κατά την διάρκεια σχεδίασης και υλοποίησης του πληροφοριακού συστήματος ΟΜΕΑ. Το κίνητρο της ενότητας είναι να παρέχει σε όλους του προγραμματιστές και αρχιτέκτονες που εμπλέκονται στην διαδικασία υλοποίησης του λογισμικού την κατάλληλη γνώση για να συντηρείται ορθά το σύστημα. Για να μπορέσει ένα λογισμικό να αναβαθμίζεται σωστά πολλά χρόνια πρέπει να έχει αυστηρά καθορισμένες αρχές οι οποίες ακολουθούν τις καλές πρακτικές των αρχών ανάπτυξης λογισμικού. Οι αρχές βοηθάνε τους προγραμματιστές να γνωρίζουν ποια λύση θα χρησιμοποιήσουν για να λύσουν ένα συγκεκριμένο πρόβλημα. Οι αρχές δεν είναι ίδιες για όλα τα λογισμικά αλλά τροποποιούνται ανάλογα με τις επιχειρησιακές ανάγκες που έχει του κάθε έργου. Υπάρχουν αυστηρές αρχές και μη αυστηρές αρχές. Όσες χαρακτηρίζονται ως αυστηρές πρέπει να μην παραβιάζονται σε κανένα μέρος του πληροφοριακού συστήματος με διαφορά όσες χαρακτηρίζονται ως μη αυστηρές σημαίνει ότι καλό είναι οι προγραμματιστές να τις ακολουθούν όπου μπορούν χωρίς όμως να είναι σημαντικό μειονέκτημα η έλλειψη τους.

Παρακάτω παρουσιάζονται όλες οι αρχές που χρησιμοποιήθηκαν κατά τη διάρκεια της σχεδίασης και υλοποίησης του πληροφοριακού συστήματος ΟΜΕΑ. Οι αρχές αυτές δεν σχετίζονται μόνο με το προγραμματιστικό κομμάτι, δηλαδή τον κώδικα, αλλά και με τον κλάδο της σχεδίασης λογισμικού, που αφορά τον τρόπο με τον οποίο δομείται το πρόβλημα σε μικρότερου μεγέθους προβλήματα και καθορίζονται οι ευθύνες κάθε υποσυστήματος.

### 3.6.1 Αρχιτεκτονική στρατηγική διαστρωμάτωσης

Η αρχή στρατηγικής διαστρωμάτωσης είναι ο διαχωρισμός των τεχνικών ευθυνών σε η στρώματα. Ο βασικός διαχωρισμός που γίνεται στα λογισμικά είναι σε τρία στρώματα. Το στρώμα διεπαφής, επιχειρησιακής λογικής και ανάκτησης δεδομένων από κάποια πηγή δεδομένων η οποία μπορεί να είναι ένα αρχείο ή μια βάση δεδομένων. Αυτός ο διαχωρισμός δίνει την ευελιξία στους προγραμματιστές να μπορούν να τροποποιούν συγκεκριμένα κομμάτια κώδικα ενός στρώματος χωρίς να επηρεάζονται τα άλλα στρώματα. Για παράδειγμα οι αλλαγές που γίνονται στην γραφική διεπαφή δεν πρέπει να επηρεάζουν την επιχειρησιακή λογική επειδή βρίσκονται σε διαφορετικά τεχνικά στρώματα. Κάθε στρώμα πρέπει να μπορεί να επικοινωνεί μόνο με το αμέσως από κάτω του στρώμα για να μην υπάρχουν κυκλικές εξαρτήσεις. Στο πληροφοριακό σύστημα ΟΜΕΑ, το σύστημα Flask χωρίστηκε σε 2 στρώματα, το στρώμα επικοινωνίας που έχει την ευθύνη να επικοινωνεί με τον έξω κόσμο μέσω HTTP αιτήσεων δηλαδή τους controllers και το στρώμα της επιχειρησιακής λογικής. Η σχεδιαστή απόφαση του να έχει το στρώμα της επιχειρησιακής λογικής απευθείας πρόσβαση στην βάση δεδομένων πάρθηκε επειδή όπως αναφέρεται παρακάτω, η βάση δεδομένων στο πληροφοριακό σύστημα ΟΜΕΑ είναι το πιο σημαντικό αρχιτεκτονικό σύστημα οπότε πολλές ενέργειες εκτελούνται απευθείας στην βάση δεδομένων μέσω διαδικασιών, συναρτήσεων και όψεων

### 3.6.2 Όχι επιχειρησιακή λογική στην γραφική διεπαφή

Η συγκεκριμένη αρχή της αποφυγής προσθήκης επιχειρησιακής λογικής στην γραφική διεπαφή αναφέρεται στην συγκέντρωση όλης της επιχειρησιακής λογικής σε ένα ξεχωριστό στρώμα το οποίο πρέπει να είναι ανεξάρτητο από την γραφική διεπαφή. Ο λόγος που αυτό το γεγονός είναι επιθυμητό είναι επειδή όλος ο κώδικας της γραφικής διεπαφής εκτελείται στην πλευρά του πελάτη άρα θα μπορούσε πολύ εύκολα να μεταβληθεί ο κώδικας της από μη εξουσιοδοτημένους χρήστες. Επίσης στα σύνθετα λογισμικά είναι επιθυμητό να μπορεί να τσεκάρετε η επιχειρησιακή λογική, όσο αυτή όμως είναι στην γραφική διεπαφή γίνεται πολύ δύσκολο το testing για να διασφαλιστεί ότι το σύστημα λειτουργεί ομαλά. Για το παρόν πληροφοριακό σύστημα η γραφική διεπαφή υλοποιείται με την βοήθεια της βιβλιοθήκης React JS, η οποία είναι υπεύθυνη μόνο για να ανακτά τα δεδομένα από την Flask και να τα εμφανίζει σε μορφή που είναι κατάλληλη για τους διδάσκοντες.

### 3.6.3 Έξυπνη βάση δεδομένων

Η αρχή της έξυπνης βάσης δεδομένων αναφέρεται στο γεγονός ότι η βάση δεδομένων είναι πολύ σημαντικό αρχιτεκτονικό στοιχείο για το πληροφοριακό σύστημα. Συγκεκριμένα για το πληροφοριακό σύστημα ΟΜΕΑ, η βάση δεδομένων είναι το πιο σημαντικό αρχιτεκτονικό στοιχείο επειδή το λογισμικό θα μπορούσε να χαρακτηριστεί και ως data centric εφαρμογή, δηλαδή τα δεδομένα είναι το πιο σημαντικό στοιχείο που κατέχει το σύστημα. Πολλά κομμάτια της επιχειρησιακής λογικής είναι ενθυλακωμένα στην βάση μέσω περιορισμούς στο σχήμα και αυτή η αρχιτεκτονική απόφαση πάρθηκε επειδή η βάση δεδομένων δύσκολα πρόκειται να αλλάξει σε σύγκριση με τα υπόλοιπα συστήματα της εφαρμογής όπως για παράδειγμα το Flask ή την React. Επίσης το πλεονέκτημα της αρχής έξυπνης βάσης δεδομένων είναι ότι το συνολικό σύστημα έχει πολύ καλύτερες επιδόσεις σε σύγκριση με την αρχή της

χαζής βάσης δεδομένων. Οι βάσεις δεδομένων δημιουργήθηκαν με τον σκοπό να είναι πολύ αποδοτικές στην διαχείριση μεγάλου όγκου δεδομένων σε σύγκριση με τα web frameworks. Για παράδειγμα είναι πιο γρήγορο η βάση δεδομένων να υπολογίσει απευθείας τους μέσους όρους από το να ζητήσει η Flask όλα τα μαθήματα, να τα φορτώσει στην μνήμη της και να υπολογίσει τους μέσους όρους.

### 3.6.4 SOLID αρχές

Για την ανάπτυξη του λογισμικού ακολουθήθηκαν οι γνωστές αρχές SOLID. Οι SOLID αρχές είναι πρακτικές καλής σχεδίασης και ανάπτυξης λογισμικού. Κάθε γράμμα της λέξης αντιπροσωπεύει μια αρχή. Το πρώτο γράμμα S αντιπροσωπεύει το Single Responsibility, η συγκεκριμένη αρχή δηλώνει ότι κάθε οντότητα λογισμικού (πακέτο, κλάση, συνάρτηση ) πρέπει να κάνει μόνο ένα πράγμα και πρέπει να υπάρχει μόνο ένας λόγος ώστε να υποστεί η οντότητα αλλαγές. Το δεύτερο γράμμα O αντιπροσωπεύει την αρχή Open-Close Principle, δηλαδή ότι ο κώδικας πρέπει να γράφεται με τέτοιο τρόπο ώστε αν ζητηθεί από τους ενδιαφερόμενους να γίνει προσθήκη λειτουργίας να πρέπει να προστεθεί κώδικας χωρίς να αλλάζει ο παλιός κώδικας. Το τρίτο γράμμα L αντιπροσωπεύει την αρχή Liskov Substitution το οποίο δηλώνει ότι τα αντικείμενα μιας υπερκλάσης πρέπει να μπορούν να αντικατασταθούν με αντικείμενα των υποκλάσεων της χωρίς να καταστραφεί η εφαρμογή. Παραβίαση του Liskov Substitution μπορεί να εμφανιστεί όταν κάποια κλάση κληρονομεί μια άλλη κλάση που έχει μεθόδους που δεν της είναι απαραίτητες για να λειτουργήσει. Το γράμμα I αντιπροσωπεύει την αρχή Interface Segregation, δηλαδή οι διεπαφές θα πρέπει να είναι αρκετά συγκεκριμένες. Τέλος το D αντιπροσωπεύει την αρχή Dependency Inversion Principle δηλαδή οι οντότητες λογισμικού υψηλού επιπέδου δεν θα πρέπει να εξαρτώνται από οντότητες λογισμικού χαμηλού επιπέδου, αλλά και οι δύο θα πρέπει να εξαρτώνται από τις αφαιρέσεις [33].

## 3.7 Αρχιτεκτονική Λογισμικού

Η αρχιτεκτονική του συστήματος θα μπορούσε να θεωρηθεί ως ένας χάρτης που ανάλογα με την μεγέθυνση του δείχνει και τις αντίστοιχες πληροφορίες. Όσο περισσότερη μεγέθυνση γίνεται στο αρχιτεκτονικό διάγραμμα φαίνονται παραπάνω πληροφορίες που σχετίζονται με την υλοποίηση του συστήματος, ενώ όσο γίνεται σμίκρυνση από τον χάρτη αρχιτεκτονικής φαίνονται πιο αφηρημένες πληροφορίες. Ανάλογα με την πληροφορία που θέλει να ανακτήσει ο προγραμματιστής η αρχιτέκτονας λογισμικού επιλέγει το κατάλληλο διάγραμμα.

Στην κοινότητα λογισμικού υπάρχει μεγάλη συζήτηση για το αν πρέπει οι αρχιτέκτονες λογισμικού να ασχολούνται με τις τεχνολογίες που θα χρησιμοποιηθούν. Σύμφωνα με τον Simon Brown όπως αναφέρει στο βιβλίο του [34] οι αρχιτέκτονες λογισμικού πρέπει να ασχολούνται με τεχνολογίες και τεχνικές υλοποίησης όπως για παράδειγμα πρωτόκολλα επικοινωνίας που πρέπει να χρησιμοποιηθούν για να υπάρχει αλληλεπίδραση ανάμεσα σε παραπάνω συστήματα. Διαφορετικά αναφέρει πως οι αρχιτέκτονες λογισμικού απομακρύνονται πολύ από το στάδιο ανάπτυξης με συνέπεια όλα τα διαγράμματα που σχεδιάζονται να μην έχουν πρακτικό αντίκτυπο και ως αποτέλεσμα να μην λαμβάνονται υπόψιν από τους προγραμματιστές.

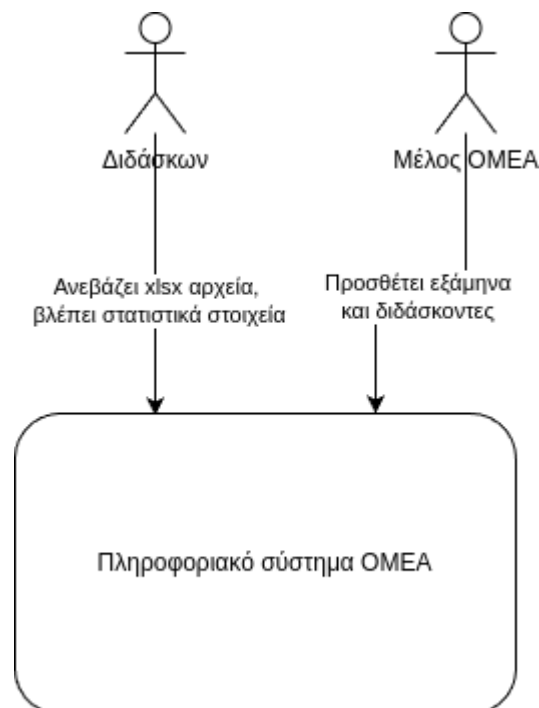
Η σχεδίαση της αρχιτεκτονικής για ένα λογισμικό συνήθως πραγματοποιείται πριν ξεκινήσει η υλοποίησή του συστήματος επειδή οργανώνει όλους του ενδιαφερόμενους που εμπλέκονται στην διαδικασία ανάπτυξης του λογισμικού. Βλέποντας τα διαγράμματα αρχιτεκτονικής μπορούν οι υπεύθυνοι του έργου να χωρίσουν του προγραμματιστές σε ομάδες με αποτέλεσμα να εργάζονται πιο αποδοτικά όλοι οι προγραμματιστές. Παρόλα αυτά, τα διαγράμματα αρχιτεκτονικής θα μπορούσαν να εξυπηρετήσουν ακόμα και μετά την υλοποίηση του συστήματος επειδή νέοι προγραμματιστές που

εισέρχονται στο έργο μπορούν μέσω των διαγραμμάτων να διαπιστώσουν πως είναι οργανωμένο και για πιο λόγο δομήθηκε με τον συγκεκριμένο τρόπο.

Για την ανάλυση της αρχιτεκτονικής του συνολικού συστήματος χρησιμοποιήθηκε η τεχνική C4 model από το βιβλίο του Simon Brown [43]. Η συγκεκριμένη μέθοδος είναι πολύ αποδοτική επειδή διαχωρίζει τις όψεις σε τέσσερις κατηγορίες που η κάθε μια περιέχει διαφορετικό επίπεδο αφαίρεσης. Κάθε επόμενο διάγραμμα περιέχει της εσωτερικές πληροφορίες του προηγούμενου. Παρακάτω παρουσιάζονται αναλυτικά και τα τέσσερα επίπεδα διαγραμμάτων σε συνδυασμό με την πληροφορία που αντιπροσωπεύουν.

### 3.7.1 Context Diagram

Το πρώτο επίπεδο ή αλλιώς Context Diagram είναι το πιο αφηρημένο διάγραμμα και παρουσιάζει το σύστημα που πρόκειται να υλοποιηθεί από τους προγραμματιστές εντός της ομάδας ανάπτυξης λογισμικού, όλων των ειδών χρηστών που αλληλοεπιδρούν με το σύστημα και όλα εξωτερικά συστήματα που χρησιμοποιούνται αλλά δεν είναι ευθύνη των προγραμματιστών εντός του οργανισμού να το συντηρούν. Για κάθε επικοινωνία συστημάτων πρέπει να αναφέρεται ο λόγος αυτής της επικοινωνίας. Κύριος σκοπός του Context Diagram είναι να παρουσιάσει την αλληλεπίδραση του συστήματος που υλοποιείται με τα εξωτερικά συστήματα. Είναι το πιο απλό από τα τέσσερα διαγράμματα, για να υλοποιηθεί αρκεί να σχεδιαστεί ένα κουτί με το όνομα του συστήματος και από γύρο του του χρήστες και εξωτερικά συστήματα. Στο σχήμα 4 παρουσιάζεται το Context Diagram του πληροφοριακού συστήματος.



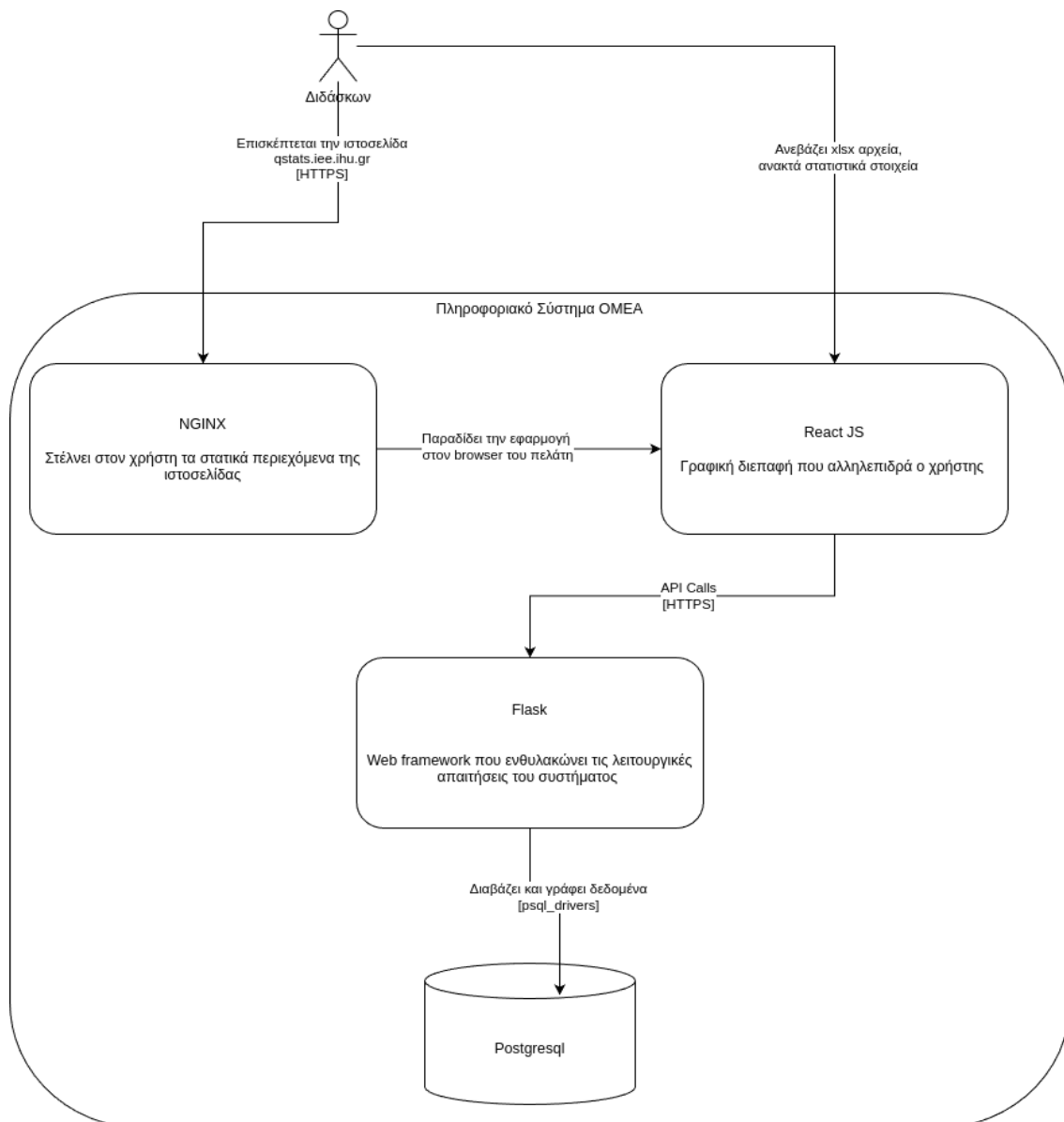
Σχήμα 4. Context Diagram του πληροφοριακού συστήματος. Παρουσιάζονται τα είδη των χρηστών που επικοινωνούν με το σύστημα.

Το παραπάνω Context Diagram παρουσιάζει τις δύο κατηγορίες χρηστών που αλληλοεπιδρούν με το σύστημα. Οι διδάσκοντες οι οποίοι αλληλοεπιδρούν με το σύστημα για να ανεβάσουν δεδομένα μέσω xlsx αρχείων είτε για να ανακτήσουν σημαντικά στατιστικά στοιχεία που σχετίζονται με τις επιδόσεις των μαθημάτων που διδάσκουν. Οι διδάσκοντες που ανήκουν στην Ομάδα Εσωτερικής Αξιολόγησης

(OMEA) χρησιμοποιούν το σύστημα για τη διαχείριση του, προκειμένου να επιτελέσουν διάφορες λειτουργίες διαχείρισης.

### 3.7.2 Container diagram

Το δεύτερο επίπεδο ή αλλιώς Container Diagram, δείχνει όλα τα containers που αποτελούν το συνολικό σύστημα που υλοποιείται. Container μπορεί να θεωρηθεί κάθε οντότητα που μπορεί να λειτουργήσει αυτόνομα και προφανώς συνεισφέρει στο συνολικό σύστημα. Για κάθε container πρέπει να αναγράφεται το όνομα του και συνοπτικά ποιες είναι οι ευθύνες του. Στο Container Diagram αναγράφονται επίσης τεχνολογικές αποφάσεις όπως για παράδειγμα ποια frameworks χρησιμοποιούνται και τα αντίστοιχα πρωτόκολλα επικοινωνίας. Πιο συγκεκριμένα, η βάση δεδομένων μπορεί θεωρηθεί container επειδή λειτουργεί αυτόνομα και έχει ως ευθύνη να παρέχει δεδομένα στα υπόλοιπα containers. Στο σχήμα 5 παρουσιάζεται το Container Diagram του πληροφοριακού συστήματος.



Σχήμα 5. Container Diagram του πληροφοριακού συστήματος. Παρουσιάζονται τα containers που αποτελούν το συνολικό πληροφοριακό σύστημα.

Το παραπάνω Container Diagram παρουσιάζει όλα τα συστήματα που αποτελούν το πληροφοριακό σύστημα OMEA. Πιο συγκεκριμένα παρουσιάζονται τα παρακάτω συστήματα

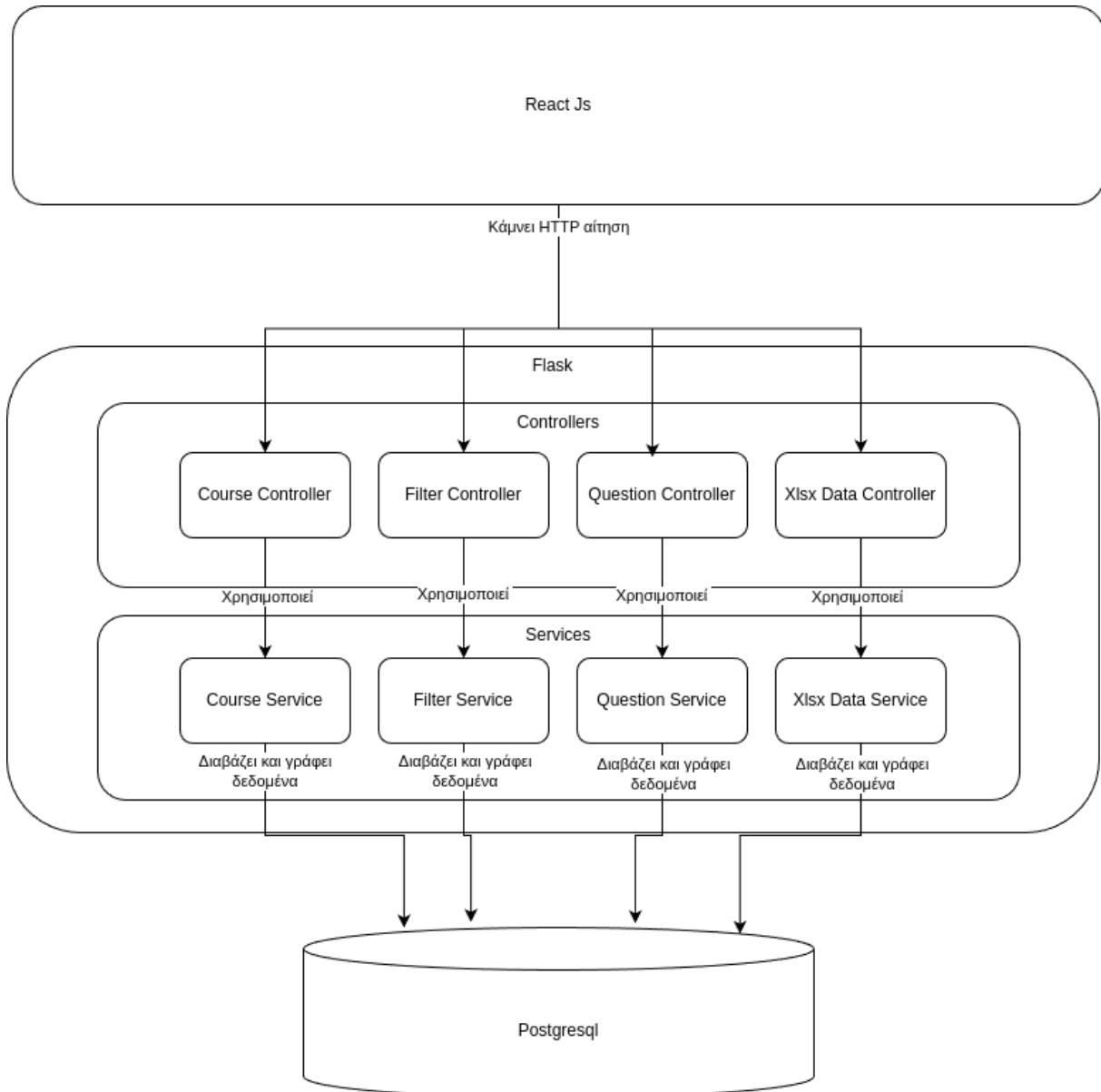
1. **NGINX:** Το container nginx είναι υπεύθυνο για να δέχεται https αιτήσεις από τους διδάσκοντες και να παραδίδει στατιστικά αρχεία όπως για παράδειγμα HTML, CSS και JavaScript. Επι της ουσίας παραδίδει όλων των κώδικα της βιβλιοθήκης React που έγινε compile με το framework Node Js. Κάνει ανακατεύθυνση τις αιτήσεις που έχουν ως όνομα τομέα qstats.iee.ihu.gr.
2. **React JS:** Η React JS αναλαμβάνει την εκτέλεση του γραφικού περιβάλλοντος στον φυλλομετρητή του χρήστη. Μέσω της React JS, οι διδάσκοντες έχουν τη δυνατότητα να αλληλοεπιδρούν με το σύστημα, επιτρέποντας τη μεταφόρτωση αρχείων xlsx και την παρουσίαση των δεδομένων σε μορφή πινάκων ή διαγραμμάτων. Για τη λήψη των δεδομένων, η React JS επικοινωνεί με το Flask μέσω αιτήσεων HTTPS.
3. **Flask:** Η λειτουργία του Flask container εστιάζεται στην υποδοχή αιτημάτων HTTPS από τη React JS και στην παροχή των απαιτούμενων δεδομένων. Η δομή ανταλλαγής δεδομένων είναι το JSON. Επιπλέον, επωμίζεται τον ρόλο της διαχείρισης και οργάνωσης όλων των λειτουργικών απαιτήσεων του συστήματος, συμπεριλαμβανομένης της αλληλεπίδρασης με τη βάση δεδομένων PostgreSQL μέσω της βιβλιοθήκης Flask-PostgreSQL.
4. **Postgresql:** Η PostgreSQL αποτελεί τον πυρήνα της αρχιτεκτονικής του συστήματος, καθώς όλα τα δεδομένα, όπως τμήματα, καθηγητές, ερωτηματολόγια και άλλα κρίσιμα στοιχεία, καταγράφονται εντός αυτής. Σύμφωνα με το ανωτέρω διάγραμμα, όλα τα containers του συστήματος OMEA εξαρτώνται από τη βάση δεδομένων, ενώ η ίδια η βάση δεδομένων δεν εξαρτάται από κανένα άλλο container του συστήματος. Επιπλέον, η ανεξαρτησία της βάσης δεδομένων από άλλα containers οδηγεί σε χρονοβόρες διαδικασίες αλλαγών σε αυτήν.

### 3.7.3 Component diagram

Το τρίτο επίπεδο ή αλλιώς Component diagram, δείχνει όλα τα components που αποτελούν το κάθε Container. Για κάθε Container που έχει εσωτερικές διαχωρισμένες λειτουργίες πρέπει να γίνει το αντίστοιχο component διάγραμμα. Component μπορεί να θεωρηθεί κάθε πακέτο ενός συστήματος. Όταν σχεδιάζετε μια εφαρμογή χωρίζεται το μεγάλο πρόβλημα σε μικρότερα πιο εύκολα προβλήματα, αυτά τα υποπροβλήματα εντός της εφαρμογής είναι τα components. Τα components αντιπροσωπεύουν την εσωτερική δομή της οργάνωσης του κώδικα του εκάστοτε Container που αναλύεται.

Το Component Diagram στο σχήμα 6 παρουσιάζει την εσωτερική δομή του Flask Container, με τις ευθύνες να διαχωρίζονται σε τεχνικά επίπεδα. Στο επίπεδο των Controllers, οι εσωτερικές κλάσεις διαχειρίζονται τα HTTP αιτήματα που προέρχονται από τη React JS, ενώ τα Services αναλαμβάνουν ένα σημαντικό μέρος της επιχειρησιακής λογικής. Επιπλέον, τα Services επικοινωνούν με τη βάση δεδομένων PostgreSQL, χωρίς να χρησιμοποιείται ORM framework, αλλά γράφονται απευθείας τα αιτήματα στη βάση δεδομένων από τη Python χρησιμοποιώντας την γλώσσα SQL.

Το Component Diagram παρουσιάζει μόνο το Flask Container, καθώς αυτό είναι το μόνο που έχει οργανωμένες πολλές λειτουργίες στο εσωτερικό του και ο διαχωρισμός σε εσωτερικά πακέτα είναι ζωτικής σημασίας λόγω της ενθυλάκωσης της επιχειρηματικής λογικής. Ως προς τη React JS, δεν υπάρχει ανάγκη για Component Diagram, καθώς η λειτουργία της είναι απλή και περιορίζεται στην παρουσίαση στατιστικών στοιχείων στην οθόνη και στη λήψη εισόδου από τον χρήστη.



Σχήμα 6. Component Diagram του πληροφοριακού συστήματος. Παρουσιάζονται τα components εντός του Flask container

### 3.7.4 Code-level diagram

Το τέταρτο και τελευταίο επίπεδο ή αλλιώς Code-level diagram, δείχνει το διάγραμμα των domain κλάσεων. Ουσιαστικά θα μπορούσε να θεωρηθεί και ως ένα UML Class Diagram όμως έχει την σημαντική διαφορά ότι δεν είναι αυστηρά καθορισμένη η σύνταξη του διαγράμματος σε σύγκριση με το UML. Αυτό το γεγονός δίνει την δυνατότητα στον αρχιτέκτονα λογισμικού να παρουσιάσει με ευέλικτο τρόπο μόνο την πληροφορία που είναι απαραίτητη για την κατανόηση της λύσης του προβλήματος

### 3.8 Δεδομένα

Στην συγκεκριμένη ενότητα παρουσιάζεται η μεθοδολογία που ακολουθήθηκε για την ανάλυση της φύσης των δεδομένων ώστε τα τελευταία να μπορέσουν στην συνέχεια να δομηθούν στο σχεσιακό σχήμα. Για κάθε σημαντική οντότητα που σχετίζεται με το πληροφοριακό σύστημα OMEA

δημιουργήθηκε ένας πίνακας που περιγράφει τα χαρακτηριστικά της οντότητα, τον τύπο δεδομένων των χαρακτηριστικών της κάθε οντότητας και ποια ιδιότητα της οντότητας αντιπροσωπεύει το κάθε χαρακτηριστικό. Για την εύρεση των σημαντικών οντοτήτων διαβάστηκαν προσεκτικά τα ερωτηματολόγια της ΟΜΕΑ τα οποία περιγράφονται στην εισαγωγή και με την βοήθεια του επιβλέπων διδάσκοντα αποφασίστηκαν ποιες οντότητες είναι σημαντικές και χρειάζεται να αναλυθούν. Για τις συσχετίσεις μεταξύ των δεδομένων έχει δημιουργηθεί ξεχωριστή ενότητα που αναλύει ποιες οντότητες σχετίζονται μεταξύ τους και ποια είναι πολλαπλότητα τους.

Το πληροφοριακό σύστημα ΟΜΕΑ κατά βάση είναι διαδικτυακή εφαρμογή με επίκεντρο στην ανάλυση δεδομένων ακαδημαϊκού σκοπού. Για Αυτό τον λόγο η σχεδίαση της δομής της βάσης δεδομένων είναι το πιο κρίσιμο κομμάτι όλης της εργασίας, επειδή άμα σχεδιαστεί λάθος το σχήμα της βάσης δεδομένων θα υπάρχει κίνδυνος να παραβιαστεί η ακεραιότητα των δεδομένων και να είναι πολύ δύσκολο με την πάροδο του χρόνου να τροποποιηθεί το σχήμα αν ήδη έχουν καταχωρηθεί δεδομένα στην βάση.

Τα δεδομένα αναλύονται σε conceptual επίπεδο χωρίς να λαμβάνονται υπόψιν οι τεχνικές λεπτομέρειες όπως για παράδειγμα σε ποια βάση δεδομένων θα καταχωρηθούν τα δεδομένα. Αυτό γίνεται επειδή η σχεδίαση του σχήματος των δεδομένων πρέπει να είναι σε αφηρημένη μορφή ώστε να μπορεί να προσαρμοστεί σε μια οποιαδήποτε τεχνολογία χωρίς να υπάρχουν περιορισμοί.

### 3.8.1 Οντότητα: Πανεπιστήμιο

Όλα τα στιγμιότυπα που προέρχονται από την οντότητα πανεπιστήμιο αντιπροσωπεύουν κάποιο πανεπιστήμιο της Ελλάδος

Χαρακτηριστικό	Τύπος Δεδομένων	Περιγραφή
Όνομα	Συμβολοσειρά	Το χαρακτηριστικό όνομα αντιπροσωπεύει το όνομα του εκάστοτε πανεπιστημίου.

### 3.8.2 Οντότητα: Τμήμα Πανεπιστημίου

Όλα τα στιγμιότυπα που προέρχονται από την οντότητα τμήμα πανεπιστημίου αντιπροσωπεύουν κάποιο πανεπιστημιακό τμήμα

Χαρακτηριστικό	Τύπος Δεδομένων	Περιγραφή
Όνομα	Συμβολοσειρά	Το χαρακτηριστικό όνομα αντιπροσωπεύει το όνομα του εκάστοτε πανεπιστημιακού τμήματος.
Πόλη	Συμβολοσειρά	Το χαρακτηριστικό Πόλη αντιπροσωπεύει το όνομα της πόλης του εκάστοτε πανεπιστημίου.
Πανεπιστήμιο	Αναφορά στην οντότητα Πανεπιστήμιο	Το χαρακτηριστικό Πανεπιστήμιο είναι μια αναφορά στην οντότητα

		Πανεπιστήμιο τύπου ένα προς πολλά και δηλώνει ότι κάθε πανεπιστημιακό τμήμα ανήκει σε μόνο ένα πανεπιστήμιο.
--	--	--------------------------------------------------------------------------------------------------------------

### 3.8.3 Οντότητα: Πρόγραμμα Σπουδών

Όλα τα στιγμιότυπα που προέρχονται από την οντότητα πρόγραμμα σπουδών αντιπροσωπεύουν κάποιο πρόγραμμα σπουδών κάποιου πανεπιστημιακού τμήματος.

Χαρακτηριστικό	Τύπος Δεδομένων	Περιγραφή
Όνομα	Συμβολοσειρά	Το χαρακτηριστικό όνομα αντιπροσωπεύει το όνομα του εκάστοτε πρόγραμμα σπουδών.
Πανεπιστημιακό Τμήμα	Αναφορά στην οντότητα Τμήμα Πανεπιστημίου	Το χαρακτηριστικό Πανεπιστημιακό Τμήμα είναι μια αναφορά στην οντότητα Τμήμα Πανεπιστημίου τύπου ένα προς πολλά και δηλώνει ότι κάθε Πρόγραμμα Σπουδών ανήκει σε μόνο ένα πανεπιστήμιο.

### 3.8.4 Οντότητα: Μάθημα

Όλα τα στιγμιότυπα που προέρχονται από την οντότητα μάθημα αντιπροσωπεύουν κάποιο μάθημα πανεπιστημίου το οποίο ανήκει σε ένα συγκεκριμένο πρόγραμμα σπουδών.

Χαρακτηριστικό	Τύπος Δεδομένων	Περιγραφή
Κωδικός Μαθήματος	Ακέραιος Αριθμός	Το χαρακτηριστικό κωδικός μαθήματος αντιπροσωπεύει τον κωδικό του μαθήματος. Ο κωδικός του μαθήματος πρέπει να είναι μοναδικός στο πρόγραμμα σπουδών που ανήκει το μάθημα.
Τίτλος	Συμβολοσειρά	Το χαρακτηριστικό Τίτλος αντιπροσωπεύει τον τίτλο του συγκεκριμένου μαθήματος.
Εξάμηνο	Ακέραιος Αριθμός	Το χαρακτηριστικό Εξάμηνο αντιπροσωπεύει τον αριθμό εξαμήνου που ανήκει το συγκεκριμένο μάθημα.

Πρόγραμμα Σπουδών	Αναφορά στην οντότητα Πρόγραμμα Σπουδών	Το χαρακτηριστικό Πρόγραμμα Σπουδών είναι μια αναφορά στην οντότητα Πρόγραμμα Σπουδών τύπου ένα προς πολλά και δηλώνει ότι κάθε μάθημα ανήκει σε μόνο ένα πρόγραμμα σπουδών.
-------------------	-----------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 3.8.5 Οντότητα: Εξάμηνο

Όλα τα στιγμιότυπα που προέρχονται από την οντότητα εξάμηνο αντιπροσωπεύουν κάποιο εξάμηνο.

Χαρακτηριστικό	Τύπος Δεδομένων	Περιγραφή
Όνομα	Συμβολοσειρά	Το χαρακτηριστικό Όνομα αντιπροσωπεύει τον το όνομα του εξαμήνου. Μπορεί να είναι οποιαδήποτε συμβολοσειρά αλλά συνιστάται η δομή $\{d\{4\}-\{d\{2\}[XEIM   EAP]$ .
Χρονιά	Ακέραιος Αριθμός	Το χαρακτηριστικό Χρονιά αντιπροσωπεύει την χρονιά που ανήκει το συγκεκριμένο εξάμηνο.

### 3.8.6 Οντότητα: Διδάσκων

Όλα τα στιγμιότυπα που προέρχονται από την οντότητα εξάμηνο αντιπροσωπεύουν κάποιο διδάσκων πανεπιστημίου.

Χαρακτηριστικό	Τύπος Δεδομένων	Περιγραφή
Κωδικός Διδάσκοντα	Ακέραιος Αριθμός	Το χαρακτηριστικό Κωδικός Διδάσκων αντιπροσωπεύει τον μοναδικό κωδικό διδάσκοντα εντός ενός συγκεκριμένου πανεπιστημιακού τμήματος.
Όνομα	Συμβολοσειρά	Το χαρακτηριστικό όνομα αντιπροσωπεύει το όνομα του συγκεκριμένου διδάσκοντα.
Επίθετο	Συμβολοσειρά	Το χαρακτηριστικό Επίθετο αντιπροσωπεύει το επίθετο του συγκεκριμένου διδάσκοντα.
Ψευδώνυμο	Συμβολοσειρά	Το χαρακτηριστικό Ψευδώνυμο αναφέρεται σε μία προσωπική

		επιλογή ονομασίας, την οποία πραγματοποιεί ο κάθε διδάσκων για τον ίδιο του τον εαυτό.
Πανεπιστημιακό Τμήμα	Αναφορά στην οντότητα Τμήμα Πανεπιστημίου	Το χαρακτηριστικό Πανεπιστημιακό Τμήμα είναι μια αναφορά στην οντότητα Τμήμα Πανεπιστημίου τύπου πολλά προς πολλά και δηλώνει ότι κάθε Διδάσκων μπορεί να διδάσκει σε πολλά πανεπιστημιακά τμήματα.

### 3.8.7 Οντότητα: Κατηγορία Ερώτησης

Όλα τα στιγμιότυπα που προέρχονται από την οντότητα Κατηγορία Ερώτησης αντιπροσωπεύουν μια κατηγορία στην οποία ανήκουν ερωτήσεις ενός ερωτηματολογίου.

Χαρακτηριστικό	Τύπος Δεδομένων	Περιγραφή
Περιγραφή	Συμβολοσειρά	Το χαρακτηριστικό Περιγραφή είναι συμβολοσειρά που περιγράφει την συγκεκριμένη κατηγορία..

### 3.8.8 Οντότητα: Ερώτηση

Όλα τα στιγμιότυπα που προέρχονται από την οντότητα Ερώτηση αντιπροσωπεύουν μια κατηγορία μια ερώτηση που υπάρχει εντός των ερωτηματολογίων της ΜΟΔΙΠ.

Χαρακτηριστικό	Τύπος Δεδομένων	Περιγραφή
Κωδικός Ερώτησης	Ακέραιος Αριθμός	Το χαρακτηριστικό Κωδικός Ερώτησης αντιπροσωπεύει των κωδικό που έχει η ερώτηση εντός των ερωτηματολογίων..
Τύπος Ερώτησης	Συμβολοσειρά [Εργαστήριο ή θεωρία]	Το χαρακτηριστικό Τύπος Ερώτησης αντιπροσωπεύει των τύπο της συγκεκριμένης ερώτησης. Μπορεί να πάρει μόνο τις τιμές Εργαστήριο ή Θεωρία.
Περιγραφή	Συμβολοσειρά	Το χαρακτηριστικό Περιγραφή είναι συμβολοσειρά που περιγράφει την συγκεκριμένη

		ερώτηση έτσι ακριβώς όπως εμφανίζεται και εντός των ερωτηματολογίων.
Κατηγορία Ερώτησης	Αναφορά στην οντότητα Κατηγορία Ερώτησης	Το χαρακτηριστικό Κατηγορία Ερώτησης είναι μια αναφορά στην οντότητα Κατηγορία Ερώτησης τύπου ένα προς πολλά και δηλώνει ότι κάθε ερώτηση μπορεί να ανήκει σε μόνο μια κατηγορία.

### 3.8.9 Οντότητα: Ερωτηματολόγιο

Όλα τα στιγμιότυπα που προέρχονται από την οντότητα Ερωτηματολόγιο αντιπροσωπεύουν ένα ερωτηματολόγιο το οποίο προέρχεται από τον οργανισμό ΜΟΔΙΠ.

Χαρακτηριστικό	Τύπος Δεδομένων	Περιγραφή
Τύπος Ερωτηματολογίου	Συμβολοσειρά [Εργαστήριο ή θεωρία]	Το χαρακτηριστικό Τύπος Ερωτηματολογίου αντιπροσωπεύει των τύπο του συγκεκριμένου ερωτηματολογίου. Μπορεί να πάρει μόνο τις τιμές Εργαστήριο ή Θεωρία.
Μάθημα	Αναφορά στην οντότητα Μάθημα	Το χαρακτηριστικό Μάθημα είναι μια αναφορά στην οντότητα Μάθημα τύπου ένα προς πολλά και δηλώνει ότι κάθε ερωτηματολόγιο μπορεί να ανήκει σε μόνο σε ένα μάθημα.
Εξάμηνο	Αναφορά στην οντότητα Εξάμηνο	Το χαρακτηριστικό Εξάμηνο είναι μια αναφορά στην οντότητα Εξάμηνο τύπου ένα προς πολλά και δηλώνει ότι κάθε ερωτηματολόγιο μπορεί να ανήκει σε μόνο σε ένα εξάμηνο.
Διδάσκοντες	Αναφορά στην οντότητα Διδάσκων	Το χαρακτηριστικό Διδάσκοντες είναι μια αναφορά στην οντότητα Διδάσκων τύπου πολλά προς πολλά και δηλώνει ότι κάθε ερωτηματολόγιο

		μπορεί να σχετίζεται με πολλούς διδάσκοντες καθώς ότι οι διδάσκοντες μπορούν να σχετίζονται με πολλά ερωτηματολόγια.
--	--	----------------------------------------------------------------------------------------------------------------------

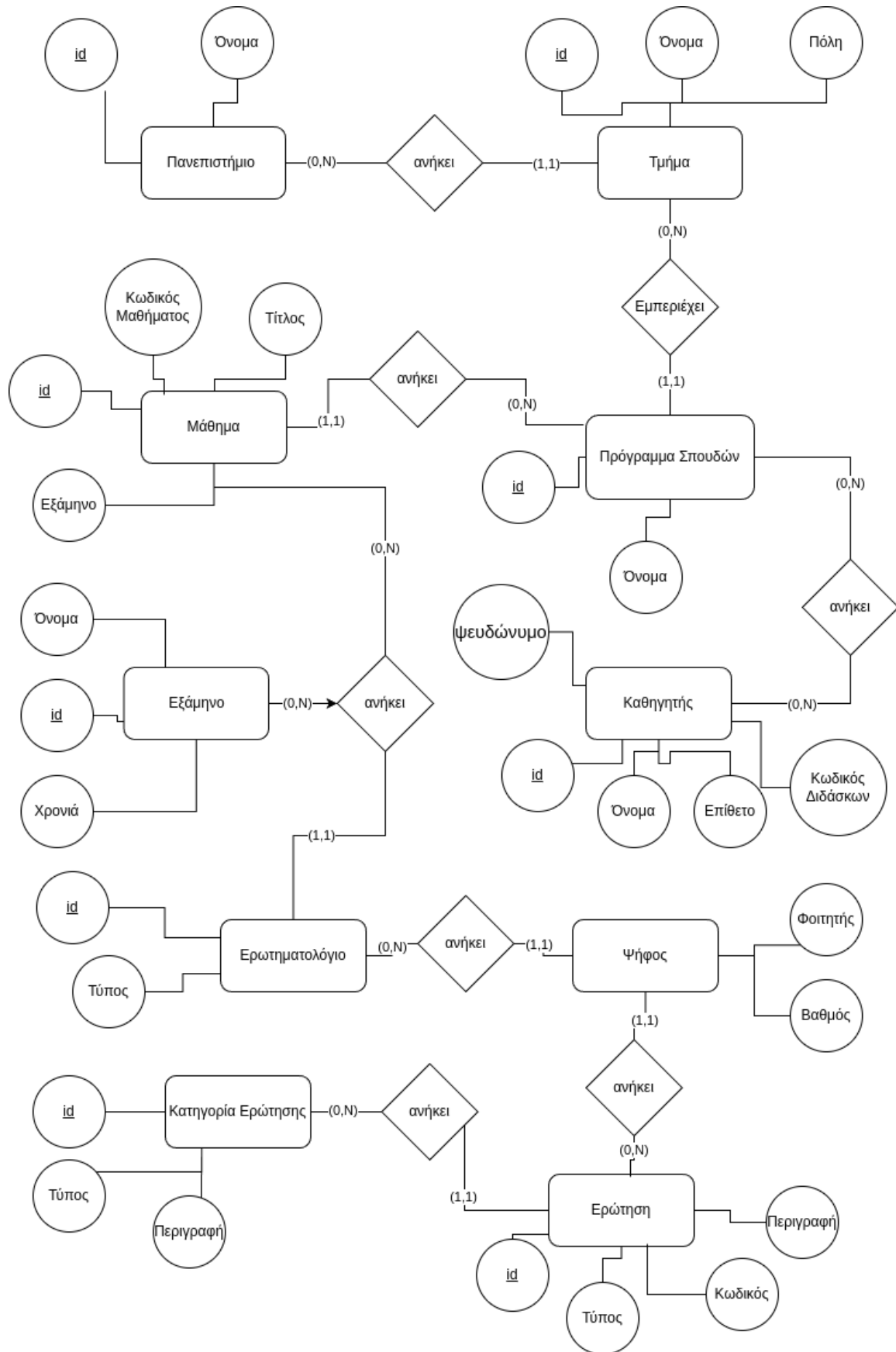
### 3.8.10 Οντότητα: Ψήφος

Όλα τα στιγμιότυπα που προέρχονται από την Ψήφος αντιπροσωπεύουν την ψήφο που καταχώρησε ένας φοιτητής για μια συγκεκριμένη ερώτηση.

Χαρακτηριστικό	Τύπος Δεδομένων	Περιγραφή
Αριθμός φοιτητή	Ακέραιος Αριθμός	Το χαρακτηριστικό Αριθμός φοιτητή αντιπροσωπεύει την σειρά του φοιτητή που απαντάει το συγκεκριμένο ερωτηματολόγιο.
Βαθμολογία	Ακέραιος Αριθμός [1 μέχρι 5]	Το χαρακτηριστικό Βαθμολογία αντιπροσωπεύει τον βαθμό της ψήφου που καταχωρεί ο φοιτητής για την συγκεκριμένη ερώτηση. Μπορεί να πάρει τις τιμές από ένα μέχρι 5.
Ερωτηματολόγιο	Αναφορά στην οντότητα Ερωτηματολόγιο	Το χαρακτηριστικό Ερωτηματολόγιο είναι μια αναφορά στην οντότητα Ερωτηματολόγιο τύπου ένα προς πολλά και δηλώνει ότι κάθε ψήφος μπορεί να ανήκει σε μόνο ένα Ερωτηματολόγιο.
Διδάσκοντες	Αναφορά στην οντότητα Διδάσκων	Το χαρακτηριστικό Διδάσκοντες είναι μια αναφορά στην οντότητα Διδάσκων τύπου πολλά προς πολλά και δηλώνει ότι κάθε ερωτηματολόγιο μπορεί να σχετίζεται με πολλούς διδάσκοντες καθώς ότι οι διδάσκοντες μπορούν να σχετίζονται με πολλά ερωτηματολόγια.

### 3.9 Διάγραμμα ER

Το διάγραμμα ER ή αλλιώς Entity-Relationship διάγραμμα χρησιμοποιείται για την δόμηση των δεδομένων σε αφηρημένο επίπεδο δηλαδή χωρίς να αναφέρονται τεχνικές λεπτομέρειες όπως για παράδειγμα η βάση δεδομένων που θα χρησιμοποιηθεί για την καταχώρηση των δεδομένων. Χρησιμοποιείται εύρος από πάρα πολλούς αναλυτές δεδομένων επειδή έχει την ευελιξία της ανοιχτής σύνταξης σε σύγκριση με την γλώσσα μοντελοποίησης UML που έχει πολύ αυστηρή και καθορισμένη σύνταξη. Η δημιουργία του διαγράμματος ER είναι συνήθως το πρώτο στάδιο προτού καταχωρηθεί το σχήμα σε σχεσιακή βάση δεδομένων, περιέχει οντότητες και συσχετίσεις μεταξύ των οντοτήτων. Στην συνέχεια το διάγραμμα μπορεί με εύκολο τρόπο να μεταφραστεί από προγραμματιστές σε κάποιο σχεσιακό ή μη σχεσιακό σχήμα [35]. Στο Σχήμα Σχήμα 7 παρουσιάζεται το διάγραμμα ER για το πληροφοριακό σύστημα ΟΜΕΑ.



Σχήμα 7. Διάγραμμα οντοτήτων συσχετίσεων

## Κεφάλαιο 4ο: Υλοποίηση συστήματος

### 4.1 Εισαγωγή

Στο συγκεκριμένο κεφάλαιο αναλύονται οι τεχνικές λεπτομέρειες του πληροφοριακού συστήματος που αναπτύχθηκε για την ΟΜΕΑ. Αρχικά παρουσιάζονται οι σημαντικοί αλγόριθμοι που εκτελούνται στο πίσω μέρος του συστήματος (Back-End) όπως για παράδειγμα το πως γίνεται έλεγχος ορθότητας των αρχείων και με ποιόν τρόπο καταχωρούνται τα δεδομένα στο σύστημα. Στην συνέχεια πραγματοποιείται εκτενής ανάλυση των REST APIs που χρησιμοποιούνται για την επικοινωνία με το front-end και το back-end. Τέλος παρουσιάζεται η διαδικασία με την οποία έγινε ο μετασχηματισμός των δεδομένων στο σχεσιακό σχήμα SQL.

### 4.2 Αλγόριθμοι

Στη συγκεκριμένη ενότητα παρουσιάζονται όλες οι ροές που είναι ζωτικής σημασίας για τη συνολική λειτουργία του συστήματος. Οι αλγόριθμοι παρουσιάζονται στη μορφή διαγραμμάτων ροής και κειμένου περιγραφής αλγορίθμου, παρέχοντας μια ολοκληρωμένη εικόνα του πως πραγματοποιούνται οι λειτουργικές απαιτήσεις εντός του συστήματος. Κάθε ροή ή αλγόριθμος παρουσιάζεται με λεπτομερή ανάλυση των βημάτων και των συνθηκών που επηρεάζουν τον τρόπο λειτουργίας του συστήματος. Το κείμενο περιγραφής αλγορίθμου παρέχει αναλυτικές επεξηγήσεις και συνοπτικές προδιαγραφές των κύριων ενεργειών που εκτελούνται σε κάθε ροή ή αλγόριθμο, ενώ τα διαγράμματα ροής ή αλλιώς flow charts προσφέρουν μια οπτική αναπαράσταση των διαδικασιών για εύκολη κατανόηση και απεικόνιση των λειτουργικών απαιτήσεων. Μέσω αυτής της συνεκτικής ανάλυσης λειτουργιών, οι αναγνώστες του κειμένου μπορούν να κατανοήσουν εύκολα την λειτουργία του συστήματος.

#### 4.2.1 Έλεγχος xlsx αρχείου

Παρακάτω παρουσιάζεται συνοπτικά ο αλγόριθμος που εκτελείται στον διακομιστή και εμπεριέχει τα βήματα που είναι αναγκαία για να πραγματοποιηθεί ο έλεγχος ενός xlsx αρχείου. Αναλύεται μόνο ο αλγόριθμος που εκτελείται στον διακομιστή λόγω ότι εκεί ενθυλακώνεται όλη η επιχειρησιακή λογική, ο φυλλομετρητής κάνει μόνο HTTP αιτήματα στον διακομιστή. Παρά το γεγονός ότι ο παρακάτω αλγόριθμος θα μπορούσε να μεταφραστεί σε οποιαδήποτε γλώσσα προγραμματισμού, ο παρακάτω αλγόριθμος είναι εμπνευσμένος από την γλώσσα προγραμματισμού Python.

```

συνάρτηση λάθη_από_xlsx(αρχείο)
    βιβλίο_εργασίας = φόρτωση_βιβλίου_εργασίας(αρχείο)
    φύλλο = βιβλίο_εργασίας.ενεργό_φύλλο
    λάθη = []
    για κάθε γραμμή στο φύλλο
        τιμή_κελιού = λήψη_τιμής_κελιού(φύλλο, γραμμή, 'A')
        αν τιμή_κελιού
            προσπάθεια
                αν είναι_έναρξη_μαθήματος(τιμή_κελιού)

```

```

αν είναι_σωστή_έναρξη_μαθήματος(τιμή_κελιού)
    πρόγραμμα_σπουδών = λήψη_προγράμματος_σπουδών(φύλλο, γραμμή)
    μάθημα = λήψη_κωδικού_μαθήματος(τιμή_κελιού)
    αν όχι_υπάρχει_πρόγραμμα_σπουδών(πρόγραμμα_σπουδών)
        προσθήκη_λάθους(λάθη, "Μη έγκυρο πρόγραμμα σπουδών", γραμμή)
    αν όχι_υπάρχει_μάθημα(μάθημα, πρόγραμμα_σπουδών)
        προσθήκη_λάθους(λάθη, "Μη έγκυρο μάθημα", γραμμή)
    καθηγητές = λήψη_καθηγητών(φύλλο, γραμμή)
    αν καθηγητές_μη_έγκυροι(καθηγητές)
        προσθήκη_λάθους(λάθη, "Μη έγκυροι καθηγητές", γραμμή)
    αλλιώς
        προσθήκη_λάθους(λάθη, "Λανθασμένο όνομα μαθήματος", γραμμή)
    εξαίρεση ως e
        προσθήκη_λάθους(λάθη, str(e), γραμμή)
    επιστροφή λάθη

```

Το πρώτο βήμα έναρξης του αλγορίθμου πραγματοποιείται από τον τρέχον ενεργό χρήστη και είναι η ανάρτηση του xlsx αρχείου μέσω της διαδικτυακής γραφικής διεπαφής. Στην συνέχεια η γραφική διεπαφή πιο συγκεκριμένα η React JS στέλνει το xlsx αρχείο στην Flask μέσω του HTTP πρωτοκόλλου επικοινωνίας. Ο λόγος που στέλνεται το αρχείο στην Flask είναι επειδή όπως αναφέρεται και στην παράγραφο των αρχών αρχιτεκτονικής λογισμικού όλες οι ενέργειες που σχετίζονται με την επιχειρησιακή λογική δεν πρέπει να εκτελούνται στην γραφική διεπαφή λόγω ότι είναι εύκολα παραβιάσιμη.

Η Flask δέχεται το αρχείο μέσω του controller xlsx\_data\_controller, στην συνέχεια μέσω κλήσης συναρτήσεων στέλνει το αρχείο στο επίπεδο service πιο συγκεκριμένα στο αρχείο xlsx\_data\_service. Στο εσωτερικό του αρχείου xlsx\_data\_service υπάρχουν πολλές συναρτήσεις που είναι υπεύθυνες για τον έλεγχο του xlsx αρχείου. Η κεντρική συνάρτηση που ξεκινάει τον έλεγχο είναι η get\_errors\_from\_xlsx η οποία παίρνει παραμετρικά το αρχείο που πρόκειται να ελεγχθεί από το σύστημα.

Η πρώτη ενέργεια που πραγματοποιείται εντός της συνάρτησης είναι η μετατροπή του αρχείου σε xlsx μορφή ώστε να είναι εφικτή η προσπέλαση κελιών με την αναφορά σε γραμμή και στήλη. Η μετατροπή αυτή γίνεται με την συνάρτηση load\_workbook η οποία είναι μέλος της βιβλιοθήκης openpyxl. Η openpyxl είναι γνωστή βιβλιοθήκη της γλώσσας προγραμματισμού Python, προέκυψε από την έλλειψη υπάρχουσας βιβλιοθήκης για την ανάγνωση/εγγραφή εγγενώς από την Python της μορφής Office Open XML [36].

Στην συνέχεια αρχικοποιείται μια άδεια λίστα με το όνομα errors. Εντός της λίστας πρόκειται να αποθηκεύσουν όλα τα σφάλματα που ανίχνευσε ο αλγόριθμός και είναι αυτά τα οποία αν υπάρχουν θα επιστραφούν στο τέλος του αλγορίθμου στον χρήστη. Εκτελείται μια δομή επανάληψης for για όσες

είναι η γραμμές του αρχείου xlsx και με την βοήθεια του αντίστοιχου index που ξεκινάει με αρχική τιμή ένα γίνεται προσπέλαση στην στήλη A στην γραμμή index και η τιμή αποθηκεύεται σε μεταβλητή με το όνομα sheet\_value.

Εάν το sheet\_value δεν είναι null περνιέται παραμετρικά στην μέθοδο isRowThatStartsCourse. Όπως είναι εύκολα αντιληπτό και από το όνομα της μεθόδου, η συγκεκριμένη μέθοδος παίρνει παραμετρικά το sheet\_value και ελέγχει εάν είναι γραμμή έναρξης μαθήματος. Αυτός ο έλεγχος είναι πολύ σημαντικός επειδή στο xlsx αρχείο που περνιέται παραμετρικά περιέχει πολλά μαθήματα από διάφορα προγράμματα σπουδών του τρέχον πανεπιστημιακού τμήματος. Γι' αυτό πρέπει να καταλάβει ο αλγόριθμος σε ποιο μάθημα αναφέρεται η συγκεκριμένη γραμμή. Ο έλεγχος γραμμής γίνεται με την βοήθεια των κανονικών εκφράσεων, εάν το sheet\_value ακολουθεί την συγκεκριμένη δομή `\"(.*)\"s*(προπτυχιακό|μεταπτυχιακό)s*$` σημαίνει ότι η τρέχον γραμμή σχετίζεται με γραμμή έναρξης μαθήματος. Το συγκεκριμένο regular expressions δηλώνει ότι πρέπει να ξεκινάει με παρενθέσεις και εντός των παρενθέσεων να υπάρχουν κάποια γράμματα και στο τέλος η συμβολοσειρά πρέπει να τελειώνει με τα λήμματα προπτυχιακό ή μεταπτυχιακό. Εφόσον θεωρηθεί ότι πρόκειται για γραμμή έναρξης μαθήματος, γίνεται στην συνέχεια έλεγχος ορθότητας της γραμμής επειδή μπορεί να ακολουθεί σωστά την δομή αλλά να είναι λάθος επειδή μπορεί να λείπει στο εσωτερικό των παρενθέσεων ο κωδικός του μαθήματος. Ο έλεγχος ορθότητας πραγματοποιείται με την βοήθεια της μεθόδου isCorrectRowThatStartsCourse, επίσης με την βοήθεια regular expression `\"(\\s*(\\d+|[MM])?\\d+)s*,s*(\\w\\s+)\"'`

Η συνάρτηση isCorrectRowThatStartsCourse επιστρέφει boolean τιμή, εάν η τιμή που επιστρέφει είναι ψευδής τότε ο αλγόριθμος προσθέτει στην λίστα errors μια συμβολοσειρά που περιγράφει την φύση του σφάλματος, διαφορετικά συνεχίζει ο αλγόριθμος με τους υπόλοιπους ελέγχους. Ο αμέσως επόμενος έλεγχος που πραγματοποιείται είναι η εξέταση ύπαρξης προγράμματος σπουδών εντός του μαθήματος. Η ενέργεια αυτή πραγματοποιείται με την βοήθεια της μεθόδου getStudyProgrammName. Η getStudyProgrammName παίρνει παραμετρικά τον αριθμό γραμμής που ξεκινάει το μάθημα και με την βοήθεια των regular expressions ελέγχει μία τις γραμμές του μαθήματος για να βρει την γραμμή που έχει στο εσωτερικό της το πρόγραμμα σπουδών που ανήκει το συγκεκριμένο μάθημα. Αν το βρει επιστρέφει η συνάρτηση επιστρέφει το πρόγραμμα σπουδών διαφορετικά η αναζήτηση συνεχίζεται μέχρι ωσότου να συναντηθεί άλλο μάθημα ή να φτάσει ο δείκτης στο τέλος του αρχείου, σε αυτή την περίπτωση ρίχνει εξαίρεση με το αντίστοιχο μήνυμα σφάλματος ότι το συγκεκριμένο μάθημα δεν περιέχει πρόγραμμα σπουδών. Όπως και στην προηγούμενη περίπτωση αν προκύψει σφάλμα, αυτό αποθηκεύεται στην λίστα με τα σφάλματα.

Στην συνέχεια αποθηκεύονται σε λίστα τα ονόματα όλων των διδασκόντων που είναι υπεύθυνοι για το συγκεκριμένο μάθημα. Η ενέργεια αυτή πραγματοποιείται με την βοήθεια της μεθόδου getProfessorsRowLineAndValue, η οποία λειτουργεί ακριβώς με τον ίδιο τρόπο όπως και οι προηγούμενες, δηλαδή με την βοήθεια των regular expressions αναζητείται γραμμή εντός του μαθήματος που περιέχει όλους τους καθηγητές. Οι καθηγητές εντός του μαθήματος διαχωρίζονται με το σύμβολο κόμμα “;”. Εφόσον οι διδάσκοντες υπάρχουν επιστρέφονται εσωτερικά της δομής δεδομένων λίστα, διαφορετικά η συνάρτηση ρίχνει εξαίρεση με μήνυμα σφάλματος που περιγράφει την φύση του προβλήματος έλλειψης διδασκόντων για το συγκεκριμένο μάθημα.

Εφόσον έχουν αποθηκευτεί σε μεταβλητές όλες οι σημαντικές πληροφορίες όπως για παράδειγμα ο κωδικός του μαθήματος, το πρόγραμμα σπουδών που ανήκει το μάθημα και οι διδάσκοντες που είναι υπεύθυνοι για το συγκεκριμένο μάθημα, σταματάει ο στατικός έλεγχος και ξεκινάει ο δυναμικός. Κάθε σημαντική πληροφορία που υπάρχει για το μάθημα πρέπει να γίνει δυναμικός έλεγχος ότι η

συγκεκριμένη έχει καταγραφεί στην βάση δεδομένων. Αυτό γίνεται επειδή μπορεί στο αρχείο xlsx να αναφέρονται διδάσκοντες οι οποίοι δεν υπάρχουν στην βάση δεδομένων είτε επειδή ο κωδικός του μαθήματος είναι λάθος ή αντιπροσωπεύει μάθημα που βρίσκεται σε διαφορετικό πρόγραμμα σπουδών.

Όλα τα παραπάνω σφάλματα εφόσον υπάρχουν επιστρέφονται στον χρήστη στην μορφή δεδομένων JSON. Μαζί με την πληροφορία του σφάλματος επιστρέφεται και η γραμμή του αρχείου xlsx που εντοπίστηκε το σφάλμα. Αυτό το γεγονός βοηθάει πολύ στους διδάσκοντες στο να κερδίζουν χρόνο στην διόρθωση σφαλμάτων του xlsx αρχείου. Οι διδάσκοντες κάνουν όλες τις διορθώσεις που αναγράφονται και στην συνέχεια υποβάλουν το αρχείο ξανά, η συγκεκριμένη ενέργεια πραγματοποιείται μέχρι ωσότου να μην επιστραφεί κανένα σφάλμα για να μπορέσουν να πάνε στο επόμενο στάδιο όπου είναι η εισαγωγή των στοιχείων στην βάση δεδομένων.

#### 4.2.2 Υποβολή αρχείου

Παρακάτω παρουσιάζεται συνοπτικά ο αλγόριθμος που εκτελείται στον διακομιστή και εμπεριέχει όλα τα βήματα που είναι αναγκαία για να καταχωρηθούν με επιτυχία τα ερωτηματολόγια στο σύστημα. Αναλύεται μόνο ο αλγόριθμος που εκτελείται στον διακομιστή λόγω ότι εκεί ενθυλακώνεται όλη η επιχειρησιακή λογική, ο φυλλομετρητής κάνει μόνο HTTP αιτήματα στον διακομιστή. Παρά το γεγονός ότι ο παρακάτω αλγόριθμος θα μπορούσε να μεταφραστεί σε οποιαδήποτε γλώσσα προγραμματισμού, ο παρακάτω αλγόριθμος είναι εμπνευσμένος από την γλώσσα προγραμματισμού Python.

```

συνάρτηση εισαγωγή_δεδομένων_xlsx(αρχείο, εξάμηνο):
    ερωτήματα = δημιουργία_αρχικών_ερωτημάτων(εξάμηνο)
    αρχικό_αναγνωριστικό_ερωτηματολογίου
    =λήψη_τελευταίου_αναγνωριστικού_ερωτηματολογίου_συν_ένα()
    φύλλο = φόρτωση_φύλλου(αρχείο)
    για κάθε γραμμή στο φύλλο:
        δεδομένα = λήψη_δεδομένων_γραμμής(φύλλο, γραμμή)
        αν είναι_γράμμη_μαθήματος(δεδομένα):
            επεξεργασία_μαθήματος(δεδομένα, γραμμή, αρχικό_αναγνωριστικό_ερωτηματολογίου,
            εξάμηνο, ερωτήματα)
    εισαγωγή_ερωτημάτων_στην_βάση_δεδομένων(ερωτήματα)
    υποβολή_εξαμήνου(εξάμηνο)
    επιστροφή ερωτήματα

```

Εφόσον πραγματοποιηθεί επιτυχώς η ενέργεια ελέγχου του xlsx αρχείου δηλαδή επιστραφεί από την Flask μια απάντηση που στο εσωτερικό της δεν υπάρχει καταγεγραμμένο κανένα λάθος, η React JS επιτρέπει τον διδάσκον να προχωρήσει στο επόμενο βήμα το οποίο είναι η υποβολή του αρχείου. Κάθε xlsx αρχείο που υποβάλλεται αντιπροσωπεύει τις βαθμολογίες των μαθημάτων για ένα συγκεκριμένο εξάμηνο το οποίο όμως δεν αναγράφεται στο εσωτερικό του αρχείου. Για αυτό τον λόγο, για να μπορέσει να καταχωρήσει ο διδάσκων το αρχείο πρέπει πρώτα να επιλέξει το εξάμηνο που πραγματοποιήθηκε η συγκεκριμένη διεξαγωγή του ερωτηματολογίου.

Τα εξάμηνα επιστρέφονται δυναμικά από την Flask σε συνδυασμό με την πληροφορία αν το συγκεκριμένο εξάμηνο έχει ήδη εισαχθεί στο παρελθόν. Σε περίπτωση που ο διδάσκων προσπαθήσει να υποβάλει ερωτηματολόγιο για εξάμηνο που έχει ήδη περαστεί στο σύστημα. Η γραφική διεπαφή θα εμφανίζει παράθυρο προειδοποίησης ότι άμα συνεχιστεί η ενέργεια θα χαθούν τα δεδομένα του ήδη καταχωρημένου εξαμήνου. Αυτό γίνεται επειδή κάθε εξάμηνο μπορεί να έχει καταχωρημένο μόνο ένα ερωτηματολόγιο.

Όταν ο διδάσκων πυροδοτήσει την ενέργεια υποβολής ερωτηματολογίου, το αρχείο `xlsx` στέλνεται στην Flask ώστε να ξεκινήσει η διαδικασία εισαγωγής των βαθμολογιών. Για να διασφαλιστεί η ακεραιότητα των δεδομένων, παρά το γεγονός ότι ήδη έχει γίνει έλεγχος ότι το αρχείο είναι σωστά δομημένο σε προηγούμενο στάδιο η Flask εκτελεί ξανά στο εσωτερικό της την συνάρτηση ελέγχου αρχείου σε περίπτωση που είχε γίνει προηγούμενος παράκαμψη του βήματος ελέγχου.

Η συνάρτηση που είναι υπεύθυνη για την εισαγωγή των βαθμολογιών στην βάση δεδομένων είναι η `insert_xlsx_data` η οποία βρίσκεται ομοίως σαν την άλλες στο εσωτερικό του αρχείου `xlsx_data_service`. Δέχεται παραμετρικά το αρχείο `xlsx` που περιέχει τα δεδομένα και το εξάμηνο που επιλέχθηκε από τον διδάσκων μέσω της γραφικής διεπαφής.

Όπως αναφέρεται και στην ενότητα αρχών αρχιτεκτονικής του πληροφοριακού συστήματος OMEA, η Flask επικοινωνεί με την βάση δεδομένων χωρίς την χρήση ORM με αποτέλεσμα να πρέπει να αναγράφονται από την Flask τα SQL ερωτήματα για την εισαγωγή των δεδομένων. Η επικοινωνία της Flask με την βάση δεδομένων πραγματοποιείται με την βοήθεια της βιβλιοθήκης `psycopg2`. Το `Psycopg2` είναι μία από της πιο δημοφιλής βιβλιοθήκες για τη γλώσσα προγραμματισμού Python για την επικοινωνία με την βάση δεδομένων PostgreSQL. Τα κύρια χαρακτηριστικά της είναι η πλήρης υλοποίηση των προδιαγραφών Python DB API 2.0 και η ασφάλεια νημάτων (πολλά νήματα μπορούν να μοιράζονται την ίδια σύνδεση). Σχεδιάστηκε για εφαρμογές με πολλά νήματα που δημιουργούν και καταστρέφουν πολλούς κέρσορες και κάνουν μεγάλο αριθμό ταυτόχρονων INSERTs ή UPDATEs [37].

Παρά το γεγονός που πριν την υποβολή του αρχείου πραγματοποιούμε πολλαπλή έλεγχοι για την ορθότητα των δεδομένων υπάρχουν πάλι πιθανότητες αν και χαμηλές να προκύψει κάποιο σφάλμα κατά την διαδικασία εισαγωγής των δεδομένων. Για αυτόν τον λόγο όλα τα ερωτήματα SQL πρέπει να εκτελούνται σε `transaction`, σε περίπτωση που αποτύχει κάποια από τις πολλές SQL εντολές να μην περαστεί καμία άλλη στο σύστημα και να επιστραφεί στον εκάστοτε διδάσκων το σχετικό μήνυμα σφάλματος που προκάλεσε το πρόβλημα στο σύστημα. Για να μπορέσει να επιτευχθεί αυτό εντός της συνάρτησης `insert_xlsx_data` αρχικοποιείται μια άδεια λίστα με όνομα `queries` αντί να υποβάλλονται τα `queries` απευθείας στην βάση γίνονται πρώτα εισαγωγή στην λίστα ώστε στο τέλος της συνάρτησης να εκτελούνται όλα τα ερωτήματα μαζί σε `transaction`.

Η ροή των εντολών εντός της συνάρτησης `insert_xlsx_data` είναι παρόμοιες με αυτές της `get_errors_from_xlsx` που αναγράφονται στην παραπάνω ενότητα. αρχικά το αρχείο που περνιέται παραμετρικά μετατρέπεται σε μορφή `xlsx` στο εσωτερικό της Python ώστε να είναι εφικτή η πρόσβαση σε γραμμές και στήλες του αρχείου. Στην συνέχεια με την βοήθεια των μεθόδων `isRowThatStartsCourse`, `isCorrectRowThatStartsCourse`, `getStudyProgramName`, `getCourseInfo`, `getProfessorsRowLineAndValue` και `get_questions_with_votes` καταχωρούνται σε μεταβλητές όλες οι πληροφορίες για το συγκεκριμένο μάθημα. Από την στιγμή που το πρόγραμμα διαθέτει όλες τις σημαντικές πληροφορίες μπορεί να δημιουργήσει την συμβολοσειρά που αντιπροσωπεύει ένα query. Δημιουργούνται τρία query, ένα για κάθε οντότητα.

Όταν ο δείκτης φτάσει στο τέλος του αρχείου καλείται συνάρτηση `insert_queries_to_db` η οποία παίρνει παραμετρικά την λίστα με τα queries, ξεκινάει σύνδεση με την βάση δεδομένων PostgreSQL και τα εκτελεί όλα τα ερωτήματα της λίστας στην βάση. Σε περίπτωση που κάποιο query προκαλέσει σφάλμα στο εσωτερικό της βάσης εκτελείται η εντολή `rollback` για να ακυρωθεί το transaction, αν όλα τα ερωτήματα εκτελεστούν επιτυχώς επιστρέφεται μήνυμα επιτυχίας και η Flask απαντάει στην React JS με HTTP απάντηση με κωδικό 200 το οποίο σηματοδοτεί ότι όλα τα δεδομένα καταχωρίστηκαν επιτυχώς.

### 4.3 APIs

Στην συγκεκριμένη ενότητα παρουσιάζονται τα endpoints του συστήματος Flask. Μέσω των συγκεκριμένων endpoints η γραφική διεπαφή React JS μπορεί και έχει πρόσβαση στα δεδομένα της εφαρμογή καθώς μπορεί και να καταχωρήσει δεδομένα στο σύστημα. Η επικοινωνία της React Js με την Flask γίνεται μέσω HTTP αιτήσεων χρησιμοποιώντας την αρχιτεκτονική REST API.

Η λέξη REST προκύπτει από των συνδυασμών των λέξεων Μεταφορά κατάστασης αναπαράστασης (Representational State Transfer). Το REST είναι αρχιτεκτονική επικοινωνίας που βασίζεται πάνω στο πρωτόκολλο του HTTP. Στην αρχιτεκτονική REST Τα APIs οργανώνονται βάση των σημαντικών οντοτήτων του συστήματος και όχι βάση των ενεργειών του συστήματος. Ένα πολύ σημαντικό χαρακτηριστικό της αρχιτεκτονική REST είναι ότι δεν αποθηκεύεται η κατάσταση όπως και στο HTTP στο οποίο βασίζεται. Αξιοποιεί όλες τις μεθόδους του HTTP πρωτοκόλλου και ανάλογα με την μέθοδο γίνεται αναπαραστήσει διαφορετικής ενέργειας. Η μέθοδος GET χρησιμοποιείται μόνο για την ανάκτηση δεδομένων και αφού κληθεί δεν θα πρέπει να έχει γίνει κάποια αλλαγή στο σύστημα, η παράμετροι στην μέθοδο GET περνιούνται μέσω query parameters για αυτό τον λόγο καλό είναι να αποφεύγεται να περνιούνται ευαίσθητα δεδομένα επειδή είναι ορατά και στους μη εξουσιοδοτημένους χρήστες [38], [39].

Η μέθοδος POST χρησιμοποιείται για την προσθήκη δεδομένων στο σύστημα και αφού ολοκληρωθεί η κλίση του API είναι πιθανό να έχουν προστεθεί καινούργια δεδομένα εσωτερικά στο σύστημα, παρά το γεγονός ότι η μέθοδος POST χρησιμοποιείται για την προσθήκη νέων δεδομένων στο σύστημα μπορεί να χρησιμοποιηθεί και για την ανάκτηση δεδομένων. Η Μέθοδος POST για την ανάκτηση δεδομένων χρησιμοποιείται όταν πρόκειται να υπάρξουν σύνθετα κριτήρια αναζήτησης ή φιλτραρίσματος, οι παράμετροι που απαιτούνται για τον προσδιορισμό του ερωτήματος μπορεί να είναι πολύ μεγάλες ή πολύπλοκες με αποτέλεσμα να μην εφικτό να περαστούς ως query attribute στο URL του αιτήματος. Σε αυτές τις περιπτώσεις, μπορεί να γίνει παράκαμψη του κανόνα και να χρησιμοποιηθεί η μέθοδος POST για να σταλούν οι παράμετροι του ερωτήματος στο σώμα της αίτησης αντί για τη διεύθυνση URL [38], [39].

Η μέθοδος PUT χρησιμοποιείται για την τροποποίηση ήδη καταχωρημένων οντοτήτων εντός του συστήματος. Αφού ολοκληρωθεί επιτυχώς η κλίση του API είναι πιθανό να έχει τροποποιηθεί η ζητούμενη οντότητα εντός του συστήματος. Για να μπορέσει να επιτευχθεί η συγκεκριμένη ενέργεια πρέπει στην κλίση του API να δοθεί παραμετρικά το μοναδικό αναγνωριστικό της οντότητας και όλα τα χαρακτηριστικά που πρόκειται να τροποποιηθούν [38], [39].

Η μέθοδος DELETE χρησιμοποιείται για τη διαγραφή ήδη καταχωρημένων οντοτήτων εντός του συστήματος. Μετά την επιτυχή ολοκλήρωση του αιτήματος API, η ζητούμενη οντότητα θα έχει είτε διαγραφεί από το σύστημα, είτε θα παραμείνει ανεπηρέαστη, ανάλογα με τις απαιτήσεις του οργανισμού. Σημειώνεται ότι ορισμένοι οργανισμοί δίνουν μεγάλη σημασία στη διατήρηση όλων των δεδομένων, για αυτό τον λόγο πολλές φορές δεν διαγράφονται τα δεδομένα απλά δεν είναι ορατά στον

τελικό χρήστη. Για να μπορέσει να επιτευχθεί η συγκεκριμένη ενέργεια πρέπει στην κλήση του API να δοθεί παραμετρικά το μοναδικό αναγνωριστικό της οντότητας που πρόκειται να διαγραφεί ή να κρυφτεί από το σύστημα [38], [39].

Τα δεδομένα της μεθόδου POST, PUT και DELETE εισάγονται στο σώμα της HTTP αίτησης. Συνήθως τα δεδομένα εσωτερικά του σώματος οργανώνονται στην μορφή δεδομένων JSON επειδή είναι βιομηχανικό πρότυπο, ελαφριά μορφή ανταλλαγής δεδομένων. Είναι μινιμαλιστικό, χωρίς περιττά στοιχεία, γεγονός που το καθιστά αποτελεσματικό για τη μετάδοση δεδομένων μέσω δικτύου. Επιπλέον είναι πολύ εύκολο να γίνει η ανάκτηση των δεδομένων εντός της γνωστής δομής δεδομένων JSON [38], [39].

Παρά το γεγονός που έχουν οριστεί συγκεκριμένες ευθύνες για την κάθε μέθοδο GET,PUT,POST και DELETE πρακτικά κάθε μέθοδος θα μπορούσε να εξυπηρετήσει τις ίδιες ανάγκες. Παρόλα αυτά καλό είναι να ακολουθούνται οι καλές πρακτικές που έχουν οριστεί από την αρχιτεκτονική ώστε να μπορούν οι προγραμματιστές εύκολα να καταλαβαίνουν τι κάνει κάθε API και πιο πρέπει να χρησιμοποιήσουν σε κάθε περίπτωση [38], [39].

Στην αρχιτεκτονική REST χρησιμοποιούνται οι ήδη κωδικοί επιστροφής του πρωτοκόλλου HTTP. Οι κωδικοί επιστροφής είναι ζωτικής σημασίας για την επικοινωνία ανάμεσα σε πελάτες όπως για παράδειγμα οι φυλλομετρητές και σε εξυπηρετές ιστού. Οι κωδικοί επιστροφής παρέχουν έναν συγκεκριμένο τρόπο στους διακομιστές να μεταδίδουν τα αποτελέσματα των αιτημάτων στους πελάτες. Μεταδίδουν αν η αίτηση ήταν επιτυχής, αν συνάντησε σφάλμα ή αν απαιτούνται να γίνουν παραπάνω ενέργειες από την χρήστη που έκανε κλήση το web api. Παρακάτω παρουσιάζονται συνοπτικά οι πιο σημαντικοί κωδικοί επιστροφής που χρησιμοποιούνται κατά κόρον από τους προγραμματιστές [40].

1. **Κωδικός 200:** Όταν επιστρέφεται ο κωδικός 200 σημαίνει ότι δεν εμφανίστηκε κανένα σφάλμα στην πλευρά του διακομιστή.
2. **Κωδικός 301:** Όταν επιστρέφεται ο κωδικός 301 σημαίνει ότι έχει γίνει οριστική μετακόμιση της διεύθυνσης URL και ο χρήστης θα πρέπει να χρησιμοποιήσει την καινούργια διεύθυνση για να γίνει η επικοινωνία με τον διακομιστή.
3. **Κωδικός 400:** Όταν επιστρέφεται ο κωδικός 400 σημαίνει ότι έκανε κάποιο λάθος ο πελάτης κατά την διαδικασία αποστολής του αιτήματος. Συνήθως ο κωδικός 400 επιστρέφεται όταν είναι λάθος οι παράμετροι του αιτήματος
4. **Κωδικός 401:** Όταν επιστρέφεται ο κωδικός 401 σημαίνει ότι ο χρήστης που κάνει το αίτημα δεν έχει αυθεντικοποιηθεί και για να έχει πρόσβαση ο πελάτης στην συγκεκριμένη πληροφορία ή για να εκτελέσει μια συγκεκριμένη ενέργεια στην πλευρά του διακομιστή πρέπει πρώτα να γίνει η διαδικασία της αυθεντικοποίησης.
5. **Κωδικός 403:** Όταν επιστρέφεται ο κωδικός 403 σημαίνει ότι έχει πραγματοποιηθεί η διαδικασία αυθεντικοποίησης αλλά ο συγκεκριμένος χρήστης που κάνει το αίτημα δεν έχει την κατάλληλη πρόσβαση για να εκτελέσει την ενέργεια του αιτήματος στην πλευρά του διακομιστή ή για να έχει πρόσβαση στην πληροφορία που προσπαθεί να ανακτήσει
6. **Κωδικός 404:** Όταν επιστρέφεται ο κωδικός 404 σημαίνει ότι δεν βρέθηκε η πληροφορία που προσπαθεί να ανακτήσει ο χρήστης που κάνει το αίτημα.
7. **Κωδικός 500:** Όταν επιστρέφεται ο κωδικός 500 σημαίνει ότι έχει πραγματοποιηθεί κάποιο σφάλμα στην πλευρά του διακομιστή και για το συγκεκριμένο σφάλμα δεν ευθύνεται ο πελάτης που κάνει το αίτημα.

Κάθε λογισμικό που αναπτύσσεται έχει τις δικές του ιδιαιτερότητες, γεγονός που καθιστά δύσκολη την αυστηρή εφαρμογή όλων των αρχών της αρχιτεκτονικής REST. Δεν είναι απαραίτητο να

ακολουθούνται αυστηρά όλες οι αρχές αν αυτές δεν προσφέρουν ευελιξία στο σύστημα. Σε συγκεκριμένα API endpoints μπορούν γίνουν παραλλαγές της αρχιτεκτονικής ώστε να είναι πιο εύκολη και αποδοτική η ανάπτυξη των APIs με αποτέλεσμα οι εκάστοτε προγραμματιστές του τρέχον λογισμικού να μπορούν με πιο εύκολο τρόπο να προσθέτουν λειτουργίες σε σύντομο χρονικό διάστημα.

Παρακάτω παρουσιάζονται οι συνοπτικές πληροφορίες για τις διαδικτυακές διεπαφές που παρέχει το Flask. Βάσει αυτών των διαδικτυακών διεπαφών, επιτρέπεται η επικοινωνία της React JS με το Flask REST API. Οι λεπτομέρειες για κάθε διαδικτυακή διεπαφή παρουσιάζονται σε μορφή ενοτήτων. Για κάθε ένα, συμπεριλαμβάνονται πληροφορίες σχετικά με τυχόν απαιτήσεις αυθεντικοποίησης, τη μέθοδο HTTP που χρησιμοποιείται, τις παραμέτρους που δέχεται και εάν είναι υποχρεωτικές.

### 4.3.1 Έλεγχος ορθότητας XLSX αρχείου

#### 4.3.1.1 Τερματικό σημείο

POST /upload\_excel

#### 4.3.1.2 Περιγραφή

Αυτό το τελικό σημείο δέχεται ένα αίτημα POST με συνημμένο ένα αρχείο xlsx που έχει στο εσωτερικό του όλα τα μαθήματα με τις αντίστοιχες βαθμολογίες των φοιτητών, το αρχείο πρέπει να είναι στην αυθεντική μορφή που κατεβαίνει από την ΜΟΔΠΠ. Στο εσωτερικό του τελικού σημείου εκτελείται μια συνάρτηση που επικυρώνει το αρχείο και επιστρέφει την κατάσταση επικύρωσης μαζί με τυχόν σφάλματα που προέκυψαν κατά τη διαδικασία επικύρωσης.

#### 4.3.1.3 Σώμα αιτήματος

Παράμετρος	Τύπος	Περιγραφή
file	File	Αρχείο Excel προς μεταφόρτωση

#### 4.3.1.4 Απαντήσεις

Στο σενάριο επιτυχία επιστρέφεται HTTP απάντηση με κωδικό 200. Στο σώμα την απάντησης υπάρχει μήνυμα που προσδιορίζει την επιτυχία σε μορφή JSON όπως αναγράφεται παρακάτω.

```
{ "status": "CORRECT" }
```

Στο σενάριο σφάλματος επιστρέφεται HTTP απάντηση με κωδικό 400. Στο σώμα της απάντησης υπάρχει μήνυμα που προσδιορίζει το σφάλμα σε μορφή JSON όπως φαίνεται παρακάτω. Σφάλματα μπορούν να προκύψουν επειδή δεν στάλθηκε κάποιο αρχείο είτε επειδή το αρχείο δεν είναι σε xlsx μορφή.

```
{ "error": "ERROR MESSAGE" }
```

### 4.3.2 Καταχώρηση XLSX αρχείου

#### 4.3.2.1 Τερματικό σημείο

POST /submit\_excel

#### 4.3.2.2 Περιγραφή

Αυτό το τελικό σημείο δέχεται ένα αίτημα POST με συνημμένο ένα αρχείο xlsx που έχει στο εσωτερικό του όλα τα μαθήματα με τις αντίστοιχες βαθμολογίες των φοιτητών, το αρχείο πρέπει να είναι στην

αυθεντική μορφή που κατεβαίνει από την ΜΟΔΠΠ. Στο εσωτερικό του τελικού σημείου εκτελείται μια συνάρτηση που αρχικά ελέγχει το αρχείο αν είναι στην κατάλληλη μορφή και δεν λείπουν δεδομένα και στην συνέχεια καταχωρούνται τα δεδομένα στην βάση δεδομένων.

#### 4.3.2.3 Σώμα αιτήματος

Παράμετρος	Τύπος	Περιγραφή
αρχείο	File	Αρχείο Excel προς μεταφόρτωση
εξάμηνο	Συμβολοσειρά	Το εξάμηνο που έγινε η διεξαγωγή του ερωτηματολογίου

#### 4.3.2.4 Απαντήσεις

Στο σενάριο επιτυχία επιστρέφεται HTTP απάντηση με κωδικό 200. Στο σώμα την απάντησης υπάρχει μήνυμα που προσδιορίζει την επιτυχία σε μορφή JSON όπως αναγράφεται παρακάτω.

```
{ "status": "CORRECT" }
```

Στο σενάριο σφάλματος επιστρέφεται HTTP απάντηση με κωδικό 400. Στο σώμα της απάντησης υπάρχει μήνυμα που προσδιορίζει το σφάλμα σε μορφή JSON όπως φαίνεται παρακάτω. Σφάλματα μπορούν να προκύψουν επειδή δεν στάλθηκε κάποιο αρχείο, επειδή το αρχείο δεν είναι σε xlsx μορφή, το αρχείο δεν είναι σωστά δομημένο ή το εξάμηνο δεν υπάρχει καταχωρημένο στην βάση δεδομένων. Σε οποιαδήποτε άλλη περίπτωση επιστρέφεται απάντηση με κωδικό κατάστασης 500.

```
{ "error": "ERROR MESSAGE" }
```

### 4.3.3 Ανάκτηση καταχωρημένων εξαμήνων

#### 4.3.3.1 Τερματικό σημείο

GET / semesters

#### 4.3.3.2 Περιγραφή

Αυτό το τελικό σημείο δέχεται ένα αίτημα GET χωρίς να έχει κανένα αρχείο συνημμένο. Στο εσωτερικό του τελικού σημείου εκτελείται μια συνάρτηση που εκτελεί ένα ερώτημα στην βάση δεδομένων για να ανακτηθούν τα εξάμηνα που είναι καταχωρημένα στο σύστημα. Για κάθε εξάμηνο επιστρέφεται και η πληροφορία για το αν έχει ήδη εισαχθεί ερωτηματολόγιο για το συγκεκριμένο εξάμηνο. Το ίδιο τελικό σημείο μπορεί να χρησιμοποιηθεί ακόμα και για να γίνει η ανάκτηση των εξαμήνων που έχουν καταχωρηθεί για ένα συγκεκριμένο μάθημα.

#### 4.3.3.3 Σώμα αιτήματος

Παράμετρος	Τύπος	Περιγραφή
Μάθημα	Ακέραιος Αριθμός	Η παράμετρος μάθημα αντιπροσωπεύει το μοναδικό αναγνωριστικό του μαθήματος. Η συγκεκριμένη παράμετρος είναι προαιρετική

#### 4.3.3.4 Απαντήσεις

Στο σενάριο επιτυχία επιστρέφεται HTTP απάντηση με κωδικό 200. Στο σώμα την απάντησης υπάρχει μήνυμα που προσδιορίζει την επιτυχία σε μορφή JSON όπως αναγράφεται παρακάτω.

```
{ "semesters": [ { "name": "2023-24EAP", "submitted": False }, ... ] }
```

Στο σενάριο σφάλματος επιστρέφεται HTTP απάντηση με κωδικό 400. Στο σώμα της απάντησης υπάρχει μήνυμα που προσδιορίζει το σφάλμα σε μορφή JSON όπως φαίνεται παρακάτω. Σφάλματα μπορούν να προκύψουν επειδή το μάθημα που δόθηκε ως παράμετρο δεν υπάρχει καταχωρημένο στην βάση δεδομένων του συστήματος. Σε περίπτωση που εμφανιστεί εσωτερικό σφάλμα για το οποίο ευθύνεται άμεσα το σύστημα επιστρέφεται απάντηση με κωδικό κατάστασης 500.

```
{ "error": "ERROR MESSAGE" }
```

#### 4.3.4 Ανάκτηση ερωτήσεων

##### 4.3.4.1 Τερματικό σημείο

GET / questions

##### 4.3.4.2 Περιγραφή

Αυτό το τελικό σημείο δέχεται ένα αίτημα GET χωρίς να έχει κανένα αρχείο συνημμένο. Στο εσωτερικό του τελικού σημείου εκτελείται μια συνάρτηση που εκτελεί ένα ερώτημα στην βάση δεδομένων για να ανακτηθούν οι ερωτήσεις που είναι καταχωρημένες στο σύστημα. Για κάθε ερώτηση επιστρέφεται ο κωδικός της, ο τύπος της ερώτησης δηλαδή αν είναι ερώτηση εργαστηρίου ή θεωρίας και η περιγραφή της ερώτησης εντός του ερωτηματολογίου.

##### 4.3.4.3 Σώμα αιτήματος

Το συγκεκριμένο τερματικό σημείο δεν δέχεται κάποια παράμετρο η αρχείο

Παράμετρος	Τύπος	Περιγραφή
-	-	-

##### 4.3.4.4 Απαντήσεις

Στο σενάριο επιτυχία επιστρέφεται HTTP απάντηση με κωδικό 200. Στο σώμα την απάντησης υπάρχει μήνυμα που προσδιορίζει την επιτυχία σε μορφή JSON όπως αναγράφεται παρακάτω.

```
[ { "Code": "14", "q_type": "Theory", "description": "Το εκπαιδευτικό υλικό (κύριο βιβλίο, διαφάνειες, σημειώσεις κλπ) βοηθάει στην κατανόηση του μαθήματος" } ]
```

Στο σενάριο σφάλματος επιστρέφεται HTTP απάντηση με κωδικό 500. Στο σώμα της απάντησης υπάρχει μήνυμα που προσδιορίζει το σφάλμα σε μορφή JSON όπως φαίνεται παρακάτω. Σφάλματα μπορούν να προκύψουν η Flask δεν κατάφερε να επικοινωνήσει επιτυχώς με την βάση δεδομένων Postgresql.

```
{ "error": "ERROR MESSAGE" }
```

#### 4.3.5 Ανάκτηση φίλτρων

##### 4.3.5.1 Τερματικό σημείο

GET / filters

#### 4.3.5.2 Περιγραφή

Αυτό το τελικό σημείο δέχεται ένα αίτημα GET χωρίς να έχει κανένα αρχείο συνημμένο. Στο εσωτερικό του τελικού σημείου εκτελείται μια συνάρτηση που εκτελεί αντίστοιχα πολλές συναρτήσεις και ερώτημα στην βάση δεδομένων για να ανακτηθούν τα φίλτρα που είναι καταχωρημένα στο σύστημα. Με την χρήση των φίλτρων ο χρήστης μπορεί να εξάγει υποσύνολα στατιστικών στοιχείων των μαθημάτων. Τα φίλτρα χωρίζονται σε κατηγορίες ανάλογα με την ιδιότητα τους. Οι κατηγορίες είναι το πρόγραμμα σπουδών, το εξάμηνο, ο τύπος των μαθημάτων (εργαστήριο ή θεωρία), διδάσκοντες, μαθήματα, κατηγορίες ερωτήσεων και οι ερωτήσεις. Για κάθε κατηγορία φίλτρων επιστρέφεται ο κωδικός της και η περιγραφή του φίλτρου.

#### 4.3.5.3 Σώμα αιτήματος

Το συγκεκριμένο τερματικό σημείο δεν δέχεται κάποια παράμετρο η αρχείο

Παράμετρος	Τύπος	Περιγραφή
-	-	-

#### 4.3.5.4 Απαντήσεις

Στο σενάριο επιτυχία επιστρέφεται HTTP απάντηση με κωδικό 200. Στο σώμα την απάντησης υπάρχει μήνυμα που προσδιορίζει την επιτυχία σε μορφή JSON όπως αναγράφεται παρακάτω.

```
{ "COURSE": { "value": "1946 - Προηγμένα Θέματα Τεχνητής Νοημοσύνης" "value_id": "74" } .....
```

Στο σενάριο σφάλματος επιστρέφεται HTTP απάντηση με κωδικό 500. Στο σώμα της απάντησης υπάρχει μήνυμα που προσδιορίζει το σφάλμα σε μορφή JSON όπως φαίνεται παρακάτω. Σφάλματα μπορούν να προκύψουν η Flask δεν κατάφερε να επικοινωνήσει επιτυχώς με την βάση δεδομένων Postgresql.

```
{ "error": "ERROR MESSAGE" }
```

### 4.3.6 Ανάκτηση μαθημάτων

#### 4.3.6.1 Τερματικό σημείο

GET / course

#### 4.3.6.2 Περιγραφή

Αυτό το τελικό σημείο δέχεται ένα αίτημα GET χωρίς να έχει κανένα αρχείο συνημμένο. Στο εσωτερικό του τελικού σημείου εκτελείται μια συνάρτηση που εκτελεί αντίστοιχα πολλές συναρτήσεις και ερώτημα στην βάση δεδομένων για να ανακτηθούν τα μαθήματα που υπάρχουν καταχωρημένα στην βάση. Το αίτημα μπορεί να δεχθεί παραμετρικά συγκεκριμένα φίλτρα που αναφέρονται στην ενότητα Ανάκτηση Φίλτρων. Σε περίπτωση που δεν δοθούν παραμετρικά κάποια φίλτρα επιστρέφονται όλα τα μαθήματα που είναι καταχωρημένα στο σύστημα. Για κάθε μάθημα επιστρέφεται ο μοναδικός κωδικός id του μαθήματος, ο εσωτερικός κωδικός του μαθήματος, ο τίτλος, το πρόγραμμα σπουδών που ανήκει, το εξάμηνο διεξαγωγής, ο μέσος όρος των ερωτήσεων και πόσοι μαθητές συμπλήρωσαν τα ερωτηματολόγια. Είναι σημαντικό να σημειωθεί ότι ο μέσος όρος των ερωτήσεων και το πλήθος των συμπληρωμένων ερωτηματολογίων διαμορφώνεται ανάλογα με τα επιλεγμένα φίλτρα.

Τα μαθήματα μπορούν και αναρτώνται χάρη στην συνάρτηση που δημιουργήθηκε εντός της Postgresql με το όνομα `get_courses_filtered`. Οι συνάρτηση δέχεται τις παρακάτω παραμέτρους `p_department_id INT`, `p_studyprogramme INT[] DEFAULT NULL`, `p_semesters INT[] DEFAULT NULL`, `p_types TEXT[] DEFAULT NULL`, `p_professors INT[] DEFAULT NULL`, `p_qgroups INT[] DEFAULT NULL`, `p_questions INT[] DEFAULT NULL`, `p_courses INT[] DEFAULT NULL`

Κάθε παράμετρος μπορεί είναι πίνακας ακεραίων αριθμών ή συμβολοσειρών επειδή μπορεί να επιλέγονται παραπάνω φίλτρα για την ίδια κατηγορία. Οι πίνακες που σχετίζονται με το ερώτημα γίνονται `join` και ανάλογα με τα φίλτρα που επιλέχθηκαν διαμορφώνεται η συνθήκη `where` του ερωτήματος.

#### 4.3.6.3 Σώμα αιτήματος

Παράμετρος	Τύπος	Περιγραφή
Φίλτρα	JSON	Η παράμετρος Φίλτρα αντιπροσωπεύει τα φίλτρα που θα εισαχθούν στο <code>where</code> του SQL ερωτήματος

#### 4.3.6.4 Απαντήσεις

Στο σενάριο επιτυχία επιστρέφεται HTTP απάντηση με κωδικό 200. Στο σώμα την απάντησης υπάρχουν τα μαθήματα στην μορφή δεδομένων JSON έτσι όπως αναγράφεται παρακάτω.

```
[ { "id": "74", "real_course_id": "1946", "title": "Προηγμένα Θέματα Τεχνητής Νοημοσύνης", "study_programme": "ΠΡΟΠΤΥΧΙΑΚΟ", "semester": "9", "avg": "4.56", "count_of": "204" } ..... ]
```

Στο σενάριο σφάλματος επιστρέφεται HTTP απάντηση με κωδικό 500. Στο σώμα της απάντησης υπάρχει μήνυμα που προσδιορίζει το σφάλμα σε μορφή JSON όπως φαίνεται παρακάτω. Σφάλματα μπορούν να προκύψουν η Flask δεν κατάφερε να επικοινωνήσει επιτυχώς με την βάση δεδομένων Postgresql είτε επειδή τα φίλτρα δεν δόθηκαν στην σωστή μορφή, σε αυτή την περίπτωση επιστρέφεται απάντηση με κωδικό σφάλματος 400.

```
{ "error": "ERROR MESSAGE" }
```

### 4.3.7 Ανάκτηση αναλυτικών πληροφοριών μαθήματος

#### 4.3.7.1 Τερματικό σημείο

```
GET /course/<int:course_id>
```

#### 4.3.7.2 Περιγραφή

Αυτό το τελικό σημείο δέχεται ένα αίτημα GET με τον μοναδικό κωδικό του μαθήματος στο τέλος του URL. Ο σκοπός του συγκεκριμένου τερματικού σημείου είναι να επιστρέψει στον χρήστη σημαντικές στατιστικές πληροφορίες για το επιλεγμένο μάθημα, όπως για παράδειγμα την αναλυτική βαθμολογία όλων των ερωτήσεων, ποιοι διδάσκοντες είναι υπεύθυνοι για το μάθημα και πόσοι φοιτητές συμπλήρωσαν το ερωτηματολόγιο του μαθήματος. Για να μπορέσουν να επιτευχθούν οι παραπάνω ενέργειες, στο εσωτερικό του τελικού σημείου εκτελούνται τρεις συναρτήσεις οι `get_course_questions_avg`, `get_basic_course_info` και η `get_course_professors`. Κάθε συνάρτηση επιστρέφει διαφορετικές πληροφορίες για το επιλεγμένο μάθημα..

### 4.3.7.3 Σώμα αιτήματος

Παράμετρος	Τύπος	Περιγραφή
Μοναδικός κωδικός μαθήματος	Ακέραιος Αριθμός	Η παράμετρος Μοναδικός κωδικός μαθήματος αντιπροσωπεύει τον κωδικό του μαθήματος για το επιλεγμένο μάθημα

### 4.3.7.4 Απαντήσεις

Στο σενάριο επιτυχία επιστρέφεται HTTP απάντηση με κωδικό 200. Στο σώμα την απάντησης υπάρχουν οι προαναφερόμενες πληροφορίες για το επιλεγμένο μάθημα στην μορφή δεδομένων JSON έτσι όπως αναγράφεται παρακάτω

```
[ { "avg_score": 5, "avg_score_all": 5, "professors": [ { "displayName": "", "semesters": [ "2021-22EAP", "2022-23EAP" ] } ], "title": "Εισαγωγή στα Λειτουργικά Συστήματα", "total_students": 115, "total_votes": 2782, "votes": [ { "avg_score": 5, "description": "Οι στόχοι του μαθήματος είναι σαφείς", "general_score": 5, "questioncode": 11 }, ..... ]
```

Στο σενάριο σφάλματος επιστρέφεται HTTP απάντηση με κωδικό 500. Στο σώμα της απάντησης υπάρχει μήνυμα που προσδιορίζει το σφάλμα σε μορφή JSON όπως φαίνεται παρακάτω. Σφάλματα μπορούν να προκύψουν η Flask δεν κατάφερε να επικοινωνήσει επιτυχώς με την βάση δεδομένων PostgreSQL είτε επειδή δεν υπάρχει καταχωρημένο μάθημα στην βάση δεδομένων με τον αποσταλμένο κωδικό, σε αυτή την περίπτωση επιστρέφεται απάντηση με κωδικό σφάλματος 400.

```
{ "error": "ERROR MESSAGE" }
```

## 4.4 Σχήμα Βάσης Δεδομένων

Το σχεσιακό σχήμα δημιουργήθηκε σύμφωνα με την ανάλυση των δεδομένων που πραγματοποιήθηκε στην ενότητα αρχιτεκτονικής του πληροφοριακού συστήματος OMEA. Στην συγκεκριμένη ενότητα αναλύεται βήμα - βήμα η διαδικασία με την οποία μετατράπηκε το διάγραμμα σχέσεων οντοτήτων σε σχεσιακό σχήμα SQL για την βάση δεδομένων PostgreSQL.

Το πρώτο βήμα είναι κάθε οντότητα να μετατραπεί σε πίνακα. Οι οντότητες στο διάγραμμα αναπαριστώνται με το τετράγωνο και στο εσωτερικό του τετραγώνου υπάρχει το όνομα της οντότητας. Οι στήλες του πίνακα θα γίνουν τα χαρακτηριστικά στοιχεία της κάθε οντότητας. Στο διάγραμμα τα χαρακτηριστικά κάθε οντότητας απεικονίζονται με το σχήμα ρόμβου και στο εσωτερικό του υπάρχει το όνομα του χαρακτηριστικού. Όσα χαρακτηριστικά έχουν υπογραμμισμένο όνομα σημαίνει ότι είναι κύριο κλειδί, δηλαδή δεν μπορεί να υπάρχει στην βάση δεδομένων στιγμιότυπο της συγκεκριμένης οντότητας με ίδια τιμή στην συγκεκριμένη στήλη. Σύμφωνα με τα προηγούμενα αναφερόμενα στη βάση δεδομένων, θα δημιουργηθούν τουλάχιστον 10 πίνακες: Πανεπιστήμιο, Τμήμα, Μάθημα, Πρόγραμμα Σπουδών, Εξάμηνο, Καθηγητής, Ερωτηματολόγιο, Ψήφος, Κατηγορία Ερώτησης και Ερώτηση, με τα αντίστοιχα χαρακτηριστικά τους.

Οι ρόμβοι στο διάγραμμα σχέσεων οντοτήτων αναπαριστούν τις συσχετίσεις μεταξύ των οντοτήτων. Εντός αυτών, υπάρχει το όνομα της συσχέτισης, το οποίο, κατά τη διαδικασία μετατροπής, δεν είναι πολύ σημαντικό· χρησιμεύει κυρίως για να τεκμηριωθεί η ύπαρξη της συγκεκριμένης συσχέτισης. Παρόλα αυτά, η πολλαπλότητα της συσχέτισης είναι στοιχείο που μπορεί να χαρακτηριστεί ως πολύ σημαντικό. Υπάρχουν τρία βασικά είδη συσχετίσεων: το "ένα προς ένα", το "πολλά προς ένα" και το "πολλά προς πολλά". Η πολλαπλότητα αναπαρίσταται στο διάγραμμα με 2 αριθμούς εντός

παρενθέσεων, το (1,1) σημαίνει ότι η συσχέτιση είναι τύπου ένα προς ένα, το (0,N) σημαίνει ότι η συσχέτιση είναι τύπου πολλά προς ένα και αν υπάρχει το και από τις δύο πλευρές σημαίνει ότι η συσχέτιση είναι τύπου πολλά προς πολλά.

Ακολουθώντας τη μεθοδολογία που περιγράφεται παραπάνω, δημιουργήθηκε το παρακάτω σχεσιακό σχήμα οντοτήτων για τη σχεσιακή βάση δεδομένων PostgreSQL. Παρακάτω παρουσιάζονται οι πίνακες επεξήγησης των οντοτήτων και οι SQL εντολές που εκτελέστηκαν στην βάση δεδομένων για την δημιουργία του αρχικού σχεσιακού σχήματος.

Όνομα πίνακα	University
Περιγραφή πίνακα	ο πίνακας University περιέχει εγγραφές για τα πανεπιστήμια που είναι καταχωρημένα στο σύστημα.
Χαρακτηριστικό <b>id</b>	Μοναδικό αναγνωριστικό για το πανεπιστήμιο (πρωτεύον κλειδί).
Χαρακτηριστικό <b>name</b>	Όνομα του πανεπιστημίου, το οποίο πρέπει να είναι μοναδικό.
SQL εντολή για την δημιουργία του πίνακα	CREATE TABLE University ( id SERIAL PRIMARY KEY, name VARCHAR UNIQUE );

Όνομα πίνακα	Department
Περιγραφή πίνακα	ο πίνακας Department περιέχει εγγραφές για τα τμήματα των πανεπιστημίων που είναι καταχωρημένα στο σύστημα.
Χαρακτηριστικό <b>id</b>	Μοναδικό αναγνωριστικό για το τμήμα (πρωτεύον κλειδί).
Χαρακτηριστικό <b>name</b>	Όνομα του τμήματος, το οποίο πρέπει να είναι μοναδικό.
Χαρακτηριστικό <b>university_id</b>	Ξένο κλειδί που έχει αναφορά στον πίνακα University.
Χαρακτηριστικό <b>city</b>	Πόλη στην οποία βρίσκεται το συγκεκριμένο τμήμα.
SQL εντολή για την δημιουργία του πίνακα	CREATE TABLE Department ( id SERIAL PRIMARY KEY, name VARCHAR UNIQUE, university_id INT REFERENCES University(id), city VARCHAR );

Όνομα πίνακα	StudyProgramme
--------------	----------------

Περιγραφή πίνακα	ο πίνακας StudyProgramme περιέχει εγγραφές για τα προγράμματα σπουδών που προσφέρονται από κάθε πανεπιστημιακό τμήμα.
Χαρακτηριστικό <b>id</b>	Μοναδικό αναγνωριστικό για το πρόγραμμα σπουδών (πρωτεύον κλειδί).
Χαρακτηριστικό <b>name</b>	Όνομα του προγράμματος σπουδών
Χαρακτηριστικό <b>department_id</b>	Ξένο κλειδί που έχει αναφορά στον πίνακα Department.
Περιορισμός μοναδικότητας UNIQUE (name, department_id)	Ο συγκεκριμένος περιορισμός εξασφαλίζει ότι το ίδιο όνομα προγράμματος σπουδών δεν μπορεί να υπάρχει στο ίδιο τμήμα.
SQL εντολή για την δημιουργία του πίνακα	CREATE TABLE StudyProgramme ( id SERIAL PRIMARY KEY, name VARCHAR, department_id INT REFERENCES Department(id), UNIQUE (name, department_id) );

Όνομα πίνακα	Course
Περιγραφή πίνακα	ο πίνακας Course περιέχει εγγραφές για τα μαθήματα που προσφέρονται στο κάθε πρόγραμμα σπουδών.
Χαρακτηριστικό <b>id</b>	Μοναδικό αναγνωριστικό για το μάθημα (πρωτεύον κλειδί).
Χαρακτηριστικό <b>real_course_id</b>	Αναγνωριστικό για το μάθημα που χρησιμοποιείται εντός του πανεπιστημίου
Χαρακτηριστικό <b>title</b>	Τίτλος του μαθήματος
Χαρακτηριστικό <b>semester</b>	Αριθμός εξαμήνου που πραγματοποιείται το μάθημα
Χαρακτηριστικό <b>study_programme_id</b>	Αναφορά στον πίνακα StudyProgramme
Περιορισμός μοναδικότητας UNIQUE (real_course_id, study_programme_id)	Ο συγκεκριμένος περιορισμός εξασφαλίζει ότι εντός ενός προγράμματος σπουδών δεν μπορεί να υπάρχει πάνω από μια φορά το ίδιο μάθημα
SQL εντολή για την δημιουργία του πίνακα	CREATE TABLE Course ( id SERIAL PRIMARY KEY, real_course_id INT, title VARCHAR, semester INT, study_programme_id INT REFERENCES StudyProgramme(id), UNIQUE (real_course_id, study_programme_id) );

Όνομα πίνακα	Semester
Περιγραφή πίνακα	ο πίνακας Semester περιέχει εγγραφές για τα ακαδημαϊκά εξάμηνα που έχουν ολοκληρωθεί
Χαρακτηριστικό <b>id</b>	Μοναδικό αναγνωριστικό για το εξάμηνο (πρωτεύον κλειδί).
Χαρακτηριστικό <b>name</b>	Όνομα του εξαμήνου, το οποίο πρέπει να είναι μοναδικό.
Χαρακτηριστικό <b>year</b>	Έτος του εξαμήνου.
SQL εντολή για την δημιουργία του πίνακα	CREATE TABLE Semester ( id SERIAL PRIMARY KEY, name VARCHAR UNIQUE, year INT );

Όνομα πίνακα	SemesterSubmitted
Περιγραφή πίνακα	ο πίνακας SemesterSubmitted καταγράφει ποια εξάμηνα έχουν υποβληθεί από κάθε τμήμα.
Χαρακτηριστικό <b>semester_id</b>	Αναφορά στον πίνακα Semester
Χαρακτηριστικό <b>department_id</b>	Αναφορά στον πίνακα Department
Σύνθετο πρωτεύον κλειδί (semester_id, department_id)	Το συγκεκριμένο σύνθετο πρωτεύον κλειδί εξασφαλίζει ότι ένα ένα εξάμηνο έχει την δυνατότητα να υποβληθεί μόνο μία φορά
SQL εντολή για την δημιουργία του πίνακα	CREATE TABLE SemesterSubmitted ( semester_id INT REFERENCES Semester(id), department_id INT REFERENCES Department(id), PRIMARY KEY (semester_id, department_id) );

Όνομα πίνακα	Professor
Περιγραφή πίνακα	ο πίνακας Professor περιέχει εγγραφές για τους διδάσκοντες των τμημάτων
Χαρακτηριστικό <b>id</b>	Μοναδικό αναγνωριστικό του διδάσκοντα (πρωτεύον κλειδί).
Χαρακτηριστικό <b>real_professor_id</b>	Αναγνωριστικό για τον διδάσκοντα που χρησιμοποιείται στο πλαίσιο του πραγματικού κόσμου.
Χαρακτηριστικό <b>displayName</b>	Όνοματεπώνυμο του διδάσκοντα
Χαρακτηριστικό <b>firstName</b>	Όνομα του διδάσκοντα

Χαρακτηριστικό <b>lastName</b>	Επώνυμο του διδάσκοντα
SQL εντολή για την δημιουργία του πίνακα	CREATE TABLE Professor ( id SERIAL PRIMARY KEY, real_professor_id INT, displayName VARCHAR, firstName VARCHAR, LastName VARCHAR );

Όνομα πίνακα	ProfessorDepartment
Περιγραφή πίνακα	ο πίνακας ProfessorDepartment συνδέει τους καθηγητές με τα τμήματα στα οποία διδάσκουν.
Χαρακτηριστικό <b>professor_id</b>	Αναφορά στον πίνακα Professor
Χαρακτηριστικό <b>department_id</b>	Αναφορά στον πίνακα Department
Σύνθετο πρωτεύον κλειδί (professor_id, department_id)	Το συγκεκριμένο σύνθετο πρωτεύον κλειδί εξασφαλίζει ότι ένας διδάσκοντας δεν μπορεί να είναι καταγεγραμμένος πάνω από μια φορά στο ίδιο τμήμα
SQL εντολή για την δημιουργία του πίνακα	CREATE TABLE ProfessorDepartment ( professor_id INT REFERENCES Professor(id), department_id INT REFERENCES Department(id), PRIMARY KEY (professor_id, department_id)

Όνομα πίνακα	QuestionGroup
Περιγραφή πίνακα	ο πίνακας QuestionGroup περιέχει εγγραφές για τις κατηγορίες των ερωτήσεων
Χαρακτηριστικό <b>id</b>	Μοναδικό αναγνωριστικό της κατηγορίας (πρωτεύον κλειδί).
Χαρακτηριστικό <b>qg_type</b>	Τύπος ερωτήσεων στην ομάδα (ENUM QuestionType).
Χαρακτηριστικό <b>qg_description</b>	Περιγραφή της ομάδας ερωτήσεων.
SQL εντολή για την δημιουργία του πίνακα	CREATE TABLE QuestionGroup ( id SERIAL PRIMARY KEY, qg_type QuestionType, qg_description VARCHAR );

Όνομα πίνακα	Question
Περιγραφή πίνακα	ο πίνακας Question περιέχει εγγραφές για τις ερωτήσεις των ερωτηματολογίων

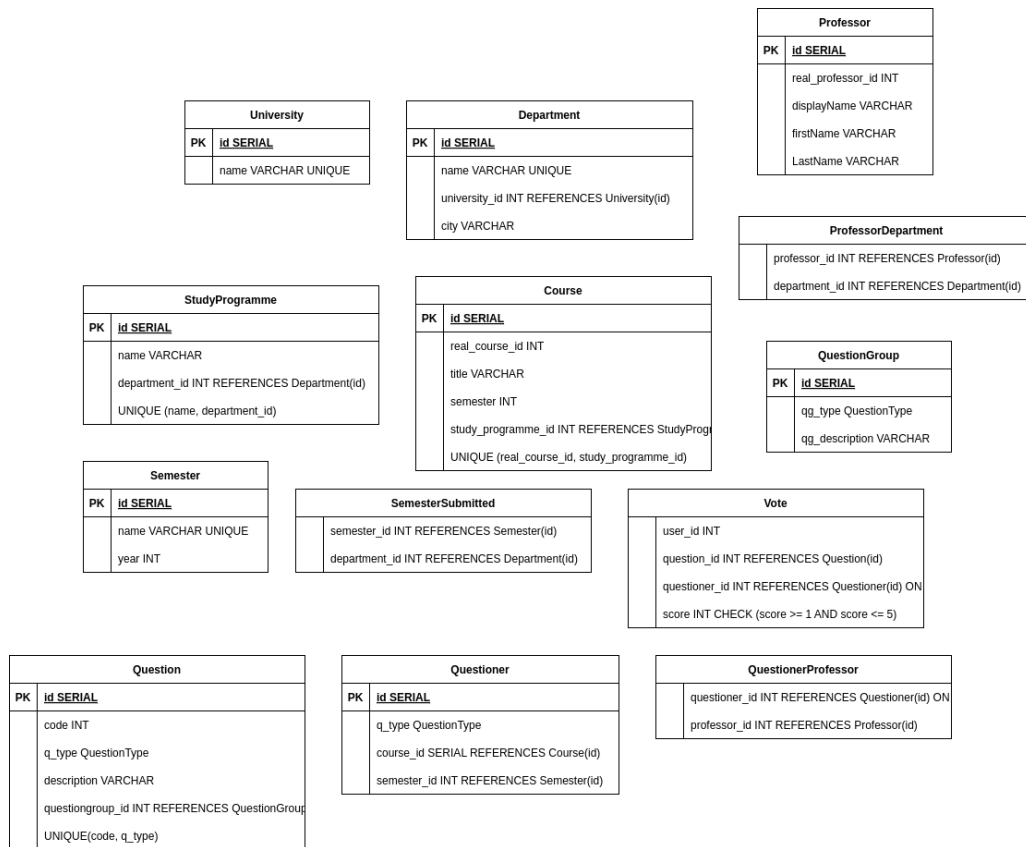
Χαρακτηριστικό <b>id</b>	Μοναδικό αναγνωριστικό της ερώτησης (πρωτεύον κλειδί).
Χαρακτηριστικό <b>code</b>	Κωδικός της ερώτησης
Χαρακτηριστικό <b>q_type</b>	Τύπος της ερώτησης (ENUM QuestionType).
Χαρακτηριστικό <b>description</b>	Περιγραφή της ερώτησης
Χαρακτηριστικό <b>questiongroup_id</b>	Αναφορά στον πίνακα QuestionGroup
Περιορισμός μοναδικότητας UNIQUE (code, q_type)	Σύνθετος μοναδικός περιορισμός που εξασφαλίζει ότι κάθε κωδικός ερώτησης είναι μοναδικός εντός του τύπου του.
SQL εντολή για την δημιουργία του πίνακα	CREATE TABLE Question ( id SERIAL PRIMARY KEY, code INT, q_type QuestionType, description VARCHAR, questiongroup_id INT REFERENCES QuestionGroup(id), UNIQUE(code, q_type) );

Όνομα πίνακα	Questioner
Περιγραφή πίνακα	ο πίνακας Questioner περιέχει εγγραφές για τα ερωτηματολόγια που είναι καταχωρημένα στο σύστημα.
Χαρακτηριστικό <b>id</b>	Μοναδικό αναγνωριστικό του ερωτηματολογίου (πρωτεύον κλειδί).
Χαρακτηριστικό <b>q_type</b>	Τύπος ερωτηματολογίου (ENUM QuestionType).
Χαρακτηριστικό <b>course_id</b>	Αναφορά στον πίνακα Course.
Χαρακτηριστικό <b>semester_id</b>	Αναφορά στον πίνακα Semester.
SQL εντολή για την δημιουργία του πίνακα	CREATE TABLE Questioner ( id SERIAL PRIMARY KEY, q_type QuestionType, course_id SERIAL REFERENCES Course(id), semester_id INT REFERENCES Semester(id) );

Όνομα πίνακα	QuestionerProfessor
Περιγραφή πίνακα	ο πίνακας QuestionerProfessor συνδέει τους καθηγητές με τα ερωτηματολόγια.
Χαρακτηριστικό <b>professor_id</b>	Αναφορά στον πίνακα Professor

Χαρακτηριστικό <b>questioner_id</b>	Αναφορά στον πίνακα Questioner
Σύνθετο πρωτεύον κλειδί (professor_id, questioner_id)	Το συγκεκριμένο σύνθετο πρωτεύον κλειδί εξασφαλίζει ότι ένας διδάσκοντας δεν μπορεί να είναι καταγεγραμμένος πάνω από μια φορά στο ίδιο ερωτηματολόγιο.
SQL εντολή για την δημιουργία του πίνακα	CREATE TABLE QuestionerProfessor ( questioner_id INT REFERENCES Questioner(id) ON DELETE CASCADE, professor_id INT REFERENCES Professor(id), PRIMARY KEY (questioner_id, professor_id) );

Όνομα πίνακα	Vote
Περιγραφή πίνακα	ο πίνακας Vote περιέχει εγγραφές για τις βαθμολογίες των φοιτητών της κάθε ερώτησης του ερωτηματολογίου.
Χαρακτηριστικό <b>user_id</b>	Αριθμός στήλης του φοιτητή εντός του ερωτηματολογίου.
Χαρακτηριστικό <b>question_id</b>	Αναφορά στον πίνακα Question.
Χαρακτηριστικό <b>questioner_id</b>	Αναφορά στον πίνακα Questioner.
Χαρακτηριστικό <b>score</b>	Η βαθμολογία που έδωσαν οι φοιτητές, με έναν περιορισμό που εξασφαλίζει ότι είναι μεταξύ 1 και 5.
SQL εντολή για την δημιουργία του πίνακα	CREATE TABLE Vote ( user_id INT, question_id INT REFERENCES Question(id), questioner_id INT REFERENCES Questioner(id) ON DELETE CASCADE, score INT CHECK (score >= 1 AND score <= 5) );



Σχήμα 8. Πίνακες του σχεσιακού σχήματος της κεντρικής βάσης δεδομένων του πληροφοριακού συστήματος.

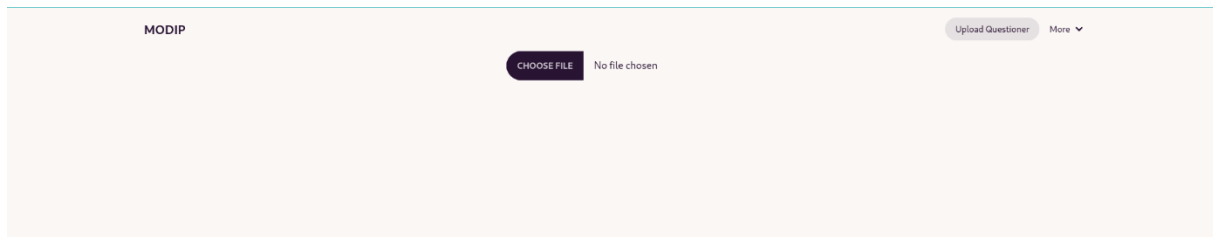
## Κεφάλαιο 5ο: Οδηγίες χρήσης

### 5.1 Εισαγωγή

Στη συγκεκριμένη ενότητα, παρουσιάζονται οι οδηγίες χρήσης του πληροφοριακού συστήματος ΟΜΕΑ για όλες τις λειτουργικές απαιτήσεις που υλοποιήθηκαν. Αναλύονται με εκτενή τρόπο οι πιο σημαντικές διαδικασίες που πρέπει να γνωρίζουν οι διδάσκοντες για να χρησιμοποιήσουν ορθά το σύστημα. Μέσω αυτής της ενότητας, οι διδάσκοντες ή γενικότερα οι χρήστες του συστήματος θα είναι σε θέση να κατανοήσουν πλήρως τις δυνατότητες του συστήματος και να το χρησιμοποιήσουν αποτελεσματικά για να φορτώσουν στο σύστημα τα ερωτηματολόγια της ΜΟΔΙΠ και να εξάγουν σημαντικά στατιστικά στοιχεία.

### 5.2 Φόρτωση και έλεγχος δεδομένων

Για την φόρτωση των δεδομένων στο πληροφοριακό σύστημα οι διδάσκοντες πρέπει πρώτα να πατήσουν το κουμπί που βρίσκεται στην πάνω δεξιά μεριά της οθόνης που έχει το όνομα Upload Questioner. Στην συνέχεια θα πραγματοποιηθεί η δρομολόγηση στην σελίδα που είναι υπεύθυνη για να την ανάρτηση των αρχείων. Στην συνέχεια πρέπει να γίνει κλικ στο κεντρικό κουμπί choose file, να επιλεγεί το xlsx αρχείο και να γίνει υποβολή. Σε περίπτωση που το xlsx αρχείο έχει λάθη στην γραφική διεπαφή θα εμφανιστούν παραπάνω πληροφορίες για τα λάθη και σε ποια γραμμή του αρχείου βρίσκονται



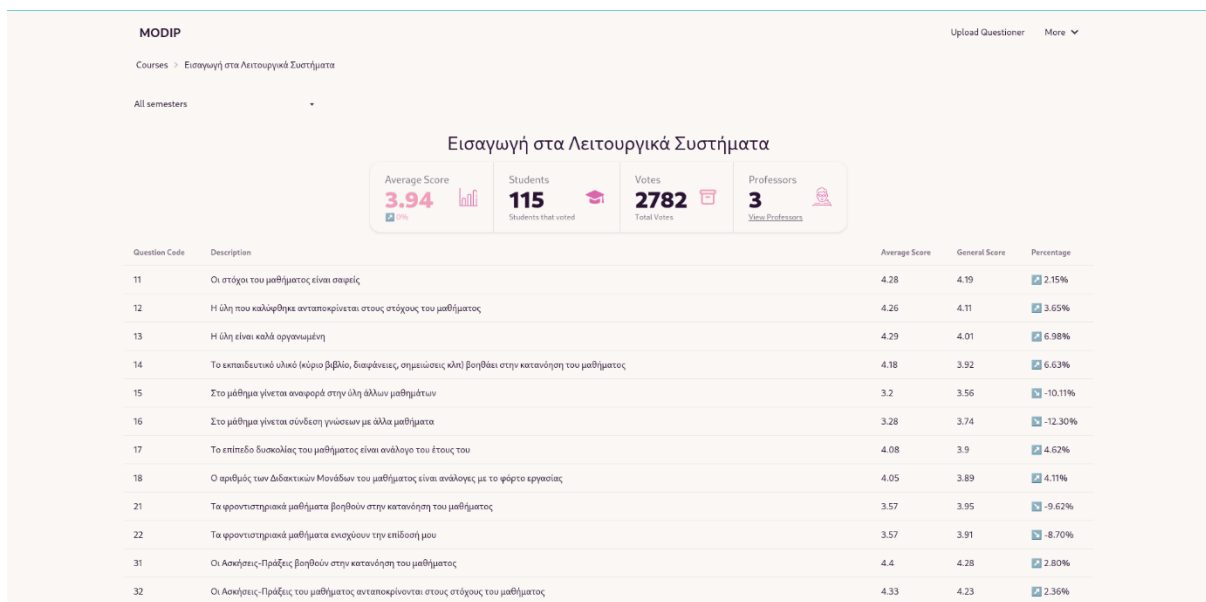
Σχήμα 9. στιγμιότυπο οθόνης του πληροφοριακού συστήματος για την ανάρτηση xlsx αρχείων

### 5.3 Προβολή Στατιστικών Στοιχείων Μαθημάτων

Για τη μετάβαση του χρήστη στη σελίδα προβολής μαθημάτων, πρέπει να πατήσει το κουμπί "More" και στη συνέχεια το κουμπί "Courses". Με αυτήν την πράξη, ο χρήστης ανακατευθύνεται στην αντίστοιχη σελίδα προβολής μαθημάτων. Όπως παρατηρείται από το παρακάτω στιγμιότυπο της εφαρμογής, η γραφική διεπαφή προσφέρει πλήρη λειτουργικότητα και πολλές επιλογές στον χρήστη. Αρχικά, ο χρήστης μπορεί να ταξινομήσει τον πίνακα μαθημάτων επιλέγοντας το κριτήριο ταξινόμησης μέσω του κουμπιού "Sort Table By". Αν ο χρήστης δεν επιθυμεί να εμφανιστούν όλα τα μαθήματα που υπάρχουν στο σύστημα, μπορεί να επιλέξει ποια μαθήματα θα εμφανίζονται μέσω των φίλτρων στο αριστερό και πάνω μέρος της οθόνης. Είναι σημαντικό να αναφερθεί ότι τα φίλτρα επηρεάζουν επίσης τη στήλη "Average Score". Για παράδειγμα, αν επιλεγεί το φίλτρο "2021-22ΕΑΡ", η στήλη "Average Score" θα αντικατοπτρίζει τους μέσους όρους των μαθημάτων για το εξάμηνο "2021-22ΕΑΡ". Τέλος, με το πάτημα του κουμπιού "Download", ο πίνακας μαθημάτων κατεβαίνει στον υπολογιστή του χρήστη σε μορφή xlsx, επιτρέποντάς του να πραγματοποιήσει εξατομικευμένες στατιστικές αναλύσεις για τα μαθήματα της επιλογής του.

## 5.4 Αναλυτική Στατιστική Ανάλυση Μαθήματος

Για να μεταβεί ο χρήστης στη σελίδα προβολής στατιστικών στοιχείων ενός συγκεκριμένου μαθήματος, πρέπει να κάνει κλικ στο όνομα του συγκεκριμένου μαθήματος. Με αυτήν την πράξη, ο χρήστης ανακατευθύνεται στην αντίστοιχη σελίδα προβολής της βαθμολογίας όλων των ερωτήσεων του μαθήματος. Όπως φαίνεται στο παρακάτω στιγμιότυπο της εφαρμογής για το μάθημα "Αντικειμενοστραφής Προγραμματισμός", ο χρήστης μπορεί να δει τη γενική βαθμολογία του μαθήματος, το σύνολο των φοιτητών που συμπλήρωσαν το ερωτηματολόγιο, τον αριθμό των μεμονωμένων ψήφων και τον αριθμό των καθηγητών που είναι υπεύθυνοι για το συγκεκριμένο μάθημα στο θεωρητικό μέρος είτε στο εργαστηριακό κομμάτι. Για να εμφανιστούν οι καθηγητές, ο χρήστης πρέπει να πατήσει το κουμπί με την ονομασία "View Professors" στην συνέχεια θα εμφανιστεί παράθυρο με τα ονόματα των καθηγητών και το εξάμηνο που συμμετείχαν. Ο πίνακας του στιγμιότυπου παρουσιάζει όλες τις ερωτήσεις που αξιολόγησαν οι φοιτητές μέσω των ερωτηματολογίων, ενώ η στήλη "Average Score" αντιπροσωπεύει τον μέσο όρο της κάθε ερώτησης για το συγκεκριμένο μάθημα. Επιπλέον, η στήλη "General Score" αντιπροσωπεύει τον μέσο όρο της κάθε ερώτησης για όλα τα μαθήματα, ενώ η στήλη "Percentage" παρουσιάζει την ποσοστιαία διαφορά μεταξύ των στηλών "Average Score" και "General Score".



Σχήμα 10. Στιγμιότυπο οθόνης του πληροφοριακού συστήματος για την στατιστικών στοιχείων ενός συγκεκριμένου μαθήματος

## Κεφάλαιο 6ο: Συζήτηση

### 6.1 Μεθοδολογία ανάπτυξης AGILE

Κατά την έναρξη του σχεδιασμού και της υλοποίησης του πληροφοριακού συστήματος OMEA, οι απαιτήσεις του λογισμικού δεν ήταν ακόμα σαφώς ορισμένες. Με αποτέλεσμα να χρησιμοποιηθεί το μανιφέστο AGILE για την βέλτιστη ανάπτυξη του λογισμικού. Το AGILE μανιφέστο είναι ένα σύνολο από καλές πρακτικές και αρχές για τη διαχείριση της διαδικασίας ανάπτυξης λογισμικού, το οποίο δημιουργήθηκε από πολύ γνωστούς μηχανικούς λογισμικού όπως για παράδειγμα τον Uncle Bob. Ο κύριος λόγος που οδήγησε τους μηχανικούς στην ανάπτυξη του μανιφέστο είναι επειδή πίστευαν ότι στην βιομηχανία ανάπτυξης λογισμικού δεν εργαζόντουσαν με ορθό τρόπο και επικεντρωνόντουσαν περισσότερο στην συγγραφή κειμένων τεκμηρίωσης παρά στην συγγραφή ποιοτικού και αναβαθμίσιμου κώδικα που μπορεί να εξυπηρετήσει συνεχές αλλαγές που ζητούνται από τους ενδιαφερόμενους που εμπλέκονται στην διαδικασία ανάπτυξης λογισμικού. Σκοπός του AGILE είναι να προσφέρει βέλτιστες πρακτικές ανάπτυξης λογισμικού, ενθαρρύνοντας την μετάδοση γνώσεων και βοηθώντας άλλους προγραμματιστές να τις εφαρμόσουν, όπως αναγράφεται και στην κεντρική ιστοσελίδα που δημιουργήθηκε για το μανιφέστο [41]. Παρά το γεγονός ότι οι αρχές που περιλαμβάνονται στο συγκεκριμένο μανιφέστο έχουν μελετηθεί εκτενώς από έμπειρους προγραμματιστές, δεν αποτελούν πάντα την ιδανική λύση για όλα τα έργα ανάπτυξης λογισμικού, όπως για παράδειγμα τα κρίσιμα συστήματα ασφαλείας όπου είναι προτιμότερο να αναπτυχθούν με μεθοδολογίες καταρράκτη [42].

Πριν από την έναρξη της υλοποίησης και σχεδίασης του πληροφοριακού συστήματος, πραγματοποιήθηκε λεπτομερής ανάλυση της δομής δεδομένων των ερωτηματολογίων της ΜΟΔΠ, τα οποία μέσω αυτών οι φοιτητές του πανεπιστημίου εκφράζουν την γνώμη τους. Μετά από την έγκριση του κ. Σιδηρόπουλου για το σχεσιακό σχήμα πινάκων της βάσης δεδομένων PostgreSQL, ξεκίνησε η διαδικασία υλοποίησης του κώδικα για τα framework Flask και React. Κάθε δύο εβδομάδες πραγματοποιούνταν διαδικτυακές συναντήσεις με τον κ. Σιδηρόπουλο μέσω της εφαρμογής Zoom. Ο κ. Σιδηρόπουλος είναι μέλος της OMEA και υπεύθυνος για την ανάπτυξη του συγκεκριμένου έργου. Στις συναντήσεις παρουσιάζονταν οι υλοποιημένες λειτουργίες και συζητούνταν πιθανές ανησυχίες σχετικά με το αποτέλεσμα. Στο τέλος της συνάντησης, γινόταν συζήτηση για το ποιες λειτουργίες πρέπει να προστεθούν στο σύστημα μέχρι την ανερχόμενη συνάντηση, λαμβάνοντας υπόψη τον διαθέσιμο χρόνο που είχα και δίνοντας έμφαση στις λειτουργικές απαιτήσεις που έχουν μεγαλύτερο αντίκτυπο στους διδάσκοντες και γενικότερα στους χρήστες που θα χρησιμοποιούν το πληροφοριακό σύστημα OMEA.

Κάθε δύο εβδομάδες, όταν το πληροφοριακό σύστημα έφτανε να είχε αρκετές καινούργιες λειτουργίες, πραγματοποιούνταν η διαδικασία της αναβάθμισης του συστήματος στον διακομιστή της OMEA. Πραγματοποιείται σύνδεση στον διακομιστή μέσω του πρωτοκόλλου SSH που μου είχε παραχωρήσει πρόσβαση ο κ. Σιδηρόπουλος. Εκεί, πρώτα κατέβαζα όλο τον κώδικα από την πλατφόρμα GitHub, στη συνέχεια δημιουργούσα εκ νέου τα images του Docker με τον καινούργιο κώδικα και εκτελούσα τα containers από τα καινούργια images. Με αυτό τον τρόπο διασφαλίζεται η συνεχής ενημέρωση του πληροφοριακού συστήματος OMEA.

### 6.2 Δυσκολίες

Κατά τη διαδικασία σχεδίασης και ανάπτυξης του πληροφοριακού συστήματος, συναντήθηκαν διάφορες δυσκολίες, οι οποίες θα αναλυθούν στις παρακάτω παραγράφους. Είναι σημαντικό να

σημειωθεί ότι οι παρακάτω δυσκολίες που αναλύονται δεν σχετίζονται μόνο την υλοποίηση του πληροφοριακού συστήματος αλλά έχουν να κάνουν γενικότερα και με την φύση του προβλήματος. Χάρη στην παρακάτω ανάλυση, θα αναδειχθεί ο τρόπος αντιμετώπισης των δυσκολιών, με αποτέλεσμα να είναι κατανοητές οι προκλήσεις που αντιμετωπίστηκαν και για ποιόν λόγο επιλέχθηκε η συγκεκριμένη λύση για το πρόβλημα.

### 6.2.1 Δομή ερωτηματολογίων

Η μεγαλύτερη δυσκολία που αντιμετωπίστηκε είναι η έλλειψη συγκεκριμένης δομής των ερωτηματολογίων της ΜΟΔΠ. Τα ερωτηματολόγια αποθηκεύονται στην μορφή xlsx αρχείων, κάθε αρχείο αφορά ένα συγκεκριμένο εξάμηνο και περιέχει τις βαθμολογίες όλων των φοιτητών που συμμετείχαν στην διαδικασία βαθμολόγησης. Το γεγονός ότι η πληροφορία αποθηκεύεται σε xlsx αρχεία και όχι σε μια αυστηρά δομημένη μορφή ενός σχεσιακού σχήματος, δημιουργεί κάποιες δυσκολίες, παρακάτω παρουσιάζονται όλες οι δυσκολίες που σχετίζονται με τα ερωτηματολόγια της ΜΟΔΠ.

1. **Όχι συγκεκριμένος αριθμός γραμμών:** Κάθε μάθημα εντός του ερωτηματολογίου δεν έχει κάποιον συγκεκριμένο αριθμό γραμμών, αυτό έχει ως μειονέκτημα ότι δύσκολα από το πρόγραμμα να αναγνωριστεί από πιο σημείο ξεκινάει και τελειώνει ένα μάθημα εντός του xlsx αρχείου. Ο τρόπος με τον οποίο αντιμετωπίστηκε το συγκεκριμένο πρόβλημα είναι με την χρήση των κανονικών εκφράσεων. Κατά την διαδικασία της σάρωσης ο αλγόριθμος που εκτελείται στην Ρυθση αναγνωρίζει ότι ξεκινάει μάθημα όταν η γραμμή είναι της μορφής `r"(.*)\s*(προπτυχιακό|μεταπτυχιακό)\s*$'` και αντίστοιχα όταν συναντήσει επόμενο μάθημα ή τελειώσει το αρχείο αναγνωρίζει το τέλος του μαθήματος.
2. **Έλλιπη δεδομένα:** Για να είναι σωστά δομημένο ένα ερωτηματολόγιο της ΜΟΔΠ, απαιτείται η παρουσία των ακόλουθων χαρακτηριστικών για κάθε μάθημα: Τίτλος, εσωτερικός κωδικός μαθήματος, πρόγραμμα σπουδών, διδάσκοντες που είναι υπεύθυνοι για το εν λόγω μάθημα, καθώς και οι ερωτήσεις με τις αντίστοιχες βαθμολογίες των φοιτητών που συμπλήρωσαν το ερωτηματολόγιο. Παρατηρήθηκε ότι σε πολλά ερωτηματολόγια υπήρχαν μαθήματα τα οποία δεν πληρούσαν τα παραπάνω χαρακτηριστικά, με αποτέλεσμα να μην είναι δυνατή η καταχώρησή τους στο σύστημα λόγω έλλειψης δεδομένων. Η απουσία αυστηρής δομής στα ερωτηματολόγια δυσκόλευε τον έλεγχο καταχώρισης των μαθημάτων στο σύστημα. Το παραπάνω πρόβλημα αντιμετωπίστηκε με την υιοθέτηση μιας μεθόδου που ανιχνεύει αυτόματα όλα τα χαρακτηριστικά κάθε μαθήματος, και σε περίπτωση έλλειψης κάποιου εμφανίζει ένα μήνυμα σφάλματος προς τον χρήστη.
3. **Λάθος δεδομένα:** Παρατηρήθηκε ότι σε ορισμένα ερωτηματολόγια που εξετάστηκαν υπήρχαν καταχωρημένα μαθήματα που είχαν όλες τις σημαντικές πληροφορίες που χρειάζεται ο αλγόριθμος του πληροφοριακού συστήματος ΜΟΔΠ για να λειτουργήσει σωστά, αλλά οι πληροφορίες των μαθημάτων ήταν λάθος όπως για παράδειγμα ο κωδικός του μαθήματος να μην είναι σωστός ή να αντιστοιχεί σε άλλο μάθημα, με αποτέλεσμα να υπάρχει πιθανότητα της παραβίασης της ακεραιότητας των δεδομένων στην κεντρική βάση δεδομένων του συστήματος. Αυτό το πρόβλημα αντιμετωπίστηκε με δύο διαφορετικές λύσεις. Αρχικά, προστέθηκε δυναμικός έλεγχος για κάθε σημαντικό στοιχείο του κάθε μαθήματος με αποτέλεσμα να επιβεβαιωθεί ότι η πληροφορία είναι ορθή. Επίσης, καταχωρήθηκαν στη βάση δεδομένων τα ανάλογα κλειδιά και περιορισμοί, για να εξασφαλιστεί ότι σε περίπτωση που προκύψει σφάλμα στην Flask, δεν θα καταχωρηθούν λάθος δεδομένα στην βάση δεδομένων.

## 6.2.2 Μη προκαθορισμένες απαιτήσεις

Κατά την διάρκεια σχεδίασης και ανάπτυξης του πληροφοριακού συστήματος ΜΟΔΠ παρουσιάστηκαν αρκετές προκλήσεις που σχετίζονται με το γεγονός ότι δεν είχε αποφασιστεί εκ των προτέρων ένας συγκεκριμένος τρόπος λειτουργίας του συστήματος. Η διαδικασία της ανάλυσης των δεδομένων από τα ερωτηματολόγια δεν είχε προκαθορισμένη λύση, κάτι που έκανε αρκετά δύσκολη τη διαδικασία ανάπτυξης του λογισμικού. Κατά τη διάρκεια της προγραμματισμού του λογισμικού, σε συνεργασία με τον κ. Σιδηρόπουλο, εξετάζαμε τακτικά τις λύσεις που υλοποιούσαμε και εάν το αποτέλεσμα που πρόκυπτε δεν ήταν ικανοποιητικό το συγκεκριμένο κομμάτι δημιουργούνταν ξανά από την αρχή. Επίσης, αποτέλεσε σημαντική πρόκληση το γεγονός ότι δεν υπήρχαν στην ακαδημαϊκή κοινότητα παραπάνω από μια διαδικτυακή εφαρμογή που εξυπηρετούσε την ίδια ανάγκη, επιπλέον δεν υπήρχαν πολλές πηγές που μπορούσα με εύκολο τρόπο να αντλήσω έμπνευση για την ανάπτυξη του πληροφοριακού συστήματος ΟΜΕΑ.

## 6.2.3 Διεξαγωγή στατιστικών στοιχείων

Μια άλλη σημαντική πρόκληση που αντιμετώπισα είναι ο υπολογισμός των στατιστικών στοιχείων των μαθημάτων με την χρήση πολλαπλών φίλτρων. Αν και ο μέσος όρος των μαθημάτων είναι σημαντικό στατιστικό στοιχείο ανάλυσης, σύμφωνα με την βιβλιογραφία αυτό το μέτρο από μόνο του δεν παρέχει τόσο σημαντική πληροφορία για την ποιότητα των μαθημάτων. Για την αντιμετώπιση αυτού του ζητήματος, ο μέσος όρος περιλάμβανε διάφορα κριτήρια, όπως το εξάμηνο διεξαγωγής του μαθήματος και οι ερωτήσεις που πρέπει να συμπεριλαμβάνονται στον μέσο όρο. Η απόκτηση όλων των μέσων όρων των μαθημάτων παρουσιάζει προκλήσεις επειδή συνεπαγόταν υψηλό υπολογιστικό κόστος. Είναι σημαντικό να εκτελεστεί αυτή η διαδικασία με τον βέλτιστο αλγοριθμικό τρόπο, ώστε οι διδάσκοντες που χρησιμοποιούν το σύστημα να μην χρειάζεται να περιμένουν αρκετό χρόνο για να έχουν πρόσβαση στην πληροφορία που χρειάζονται.

## 6.3 Μελλοντικές βελτιώσεις

Σε αυτήν την ενότητα θα παρουσιαστούν λειτουργίες που θα μπορούσαν να προστεθούν και να έχουν σημαντικό αντίκτυπο στους διδάσκοντες που χρησιμοποιούν το σύστημα, με αποτέλεσμα να βελτιωθεί η διαδικασία διεξαγωγής στατιστικών στοιχείων. Οι παρακάτω λειτουργίες μπορούν να αναπτυχθούν στο μέλλον από άλλους συναδέλφους συμφοιτητές που ενδιαφέρονται για το θέμα της διπλωματικής. Επιπλέον, θα προταθούν βελτιώσεις στις διαδικασίες που θα μπορούσαν να καταστήσουν το σύστημα πιο αποδοτικό.

### 6.3.1 NLP ανάλυση

Τα ερωτηματολόγια της ΜΟΔΠ περιλαμβάνουν, εκτός από τις ερωτήσεις επιλογής βαθμολογίας, ένα πλαίσιο όπου οι φοιτητές μπορούν να εκφράσουν ελεύθερα τη άποψη τους για τους διδάσκοντες και γενικότερα τον τρόπο διεξαγωγής του μαθήματος. Μια καλή λύση θα μπορούσε να είναι η εφαρμογή τεχνολογιών NLP (Επεξεργασία Φυσικής Γλώσσας) για την ανάλυση των απόψεων των φοιτητών που συμπλήρωσαν το ερωτηματολόγιο, προκειμένου να μετατραπεί η το γραπτό κείμενο σε ποσοτικά δεδομένα. Αυτή η προσέγγιση θα επιτρέψει την αξιολόγηση των απόψεων των φοιτητών με πιο αντικειμενικό τρόπο, καθώς τα ποσοτικά δεδομένα προσφέρουν μια πιο αντικειμενική εικόνα των πραγμάτων. Αυτή η λειτουργία μπορεί να βοηθήσει στην ανάδειξη πιθανών προβλημάτων και στην εξαγωγή συμπερασμάτων για τυχόν βελτιώσεις στη διδακτική διαδικασία.

### 6.3.2 Αυστηρή δομή ερωτηματολογίων

Για να καταστεί πιο αποδοτική η διαδικασία στατιστικής ανάλυσης στοιχείων, θα ήταν χρήσιμο εάν η ΜΟΔΠΠ μπορούσε να αποθηκεύει τα δεδομένα σε αρχεία με πιο αυστηρή και καλά δομημένη μορφή. Αυτό θα επέφερε το πλεονέκτημα της ευκολότερης ανάκτησης των δεδομένων και του πιο εύκολου ελέγχου της ορθότητας του αρχείου, επειδή θα ήταν πιο εύκολο να διαπιστωθεί από τον αλγόριθμο αν λείπει κάποιο πεδίο ή αν είναι σωστή η τιμή που αντιπροσωπεύει. Για παράδειγμα θα μπορούσαν τα μαθήματα να είχαν όλα τα χαρακτηριστικά τους στην μορφή `ΟΝΟΜΑ_ΧΑΡΑΚΤΗΡΙΣΤΙΚΟΥ : ΤΙΜΗ`. Αυτό θα είχε το πλεονέκτημα ότι θα ήταν πολύ εύκολο για τον αλγόριθμο να καταλάβει που ξεκινάει που τελειώνει κάθε μάθημα και θα μπορούσε εύκολα να διαπιστωθεί αν λείπει κάποια τιμή. Επίσης θα βοηθούσε αν όλοι διδάσκοντες είχαν το ίδιο όνομα σε όλα τα μαθήματα.

### 6.3.3 Πολλαπλά τμήματα

Η βάση δεδομένων έχει σχεδιαστεί με τρόπο που υποστηρίζει την ύπαρξη πολλών τμημάτων από διάφορα πανεπιστήμια της Ελλάδας. Ωστόσο, στην εφαρμογή αυτό δεν είναι εμφανές, καθώς παρουσιάζονται μόνο τα μαθήματα και οι βαθμολογίες που ανήκουν στο τμήμα μηχανικών πληροφορικής και ηλεκτρονικών συστημάτων του ΔΠΠΑΕ. Με την βοήθεια της λειτουργίας αυθεντικοποίησης, ανάλογα με το τμήμα του κάθε διδάσκοντα που χρησιμοποιεί το σύστημα, θα εμφανίζονται πληροφορίες και δεδομένα για το οποία τον ενδιαφέρουν και έχει εξουσιοδότηση. Η προϋπόθεση για να μπορέσει να επιτευχθεί αυτή η λειτουργία είναι τα ερωτηματολόγια της ΜΟΔΠΠ να έχουν όλα την ίδια δομή ανεξάρτητα από το τμήμα.

## 6.4 Συμπεράσματα

Η συζήτηση για τη διπλωματική εργασία επικεντρώθηκε σε διάφορα θέματα που αφορούν τη βελτίωση του πληροφοριακού συστήματος ΟΜΕΑ που υλοποιήθηκε για τις ανάγκες της διπλωματικής εργασίας. Αρχικά, αναλύθηκε η διαδικασία με την οποία συνεργάστηκα μαζί με τον κ. Σιδηρόπουλο για την ανάπτυξη του πληροφοριακού συστήματος. Στην συνέχεια παρουσιάστηκαν κάποιες προτάσεις για την βελτίωση των αρχείων που παραδίδονται από την ΜΟΔΠΠ. Συζητήθηκε για το πως η τεχνολογία NLP (Επεξεργασίας Φυσικής Γλώσσας) θα μπορούσε να βοηθήσει στην ανάλυση των σχολίων που κάνουν οι φοιτητές στα ερωτηματολόγια. Επιπλέον, συζητήθηκε η δυνατότητα επέκτασης του πληροφοριακού συστήματος ΟΜΕΑ ώστε να μπορεί να υποστηρίξει την ύπαρξη πολλών τμημάτων όχι μόνο στην βάση δεδομένων. Τέλος, προτάθηκε η αποθήκευση των δεδομένων σε αρχεία CSV για την ευκολότερη ανάκτηση και ανάλυση των δεδομένων από τα αρχεία.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] “Διεθνές Πανεπιστήμιο της Ελλάδος - Quality Assurance Unit.” Accessed: Apr. 11, 2024. [Online]. Available: <https://modip.ihu.edu.gr/>
- [2] “Διεθνές Πανεπιστήμιο της Ελλάδος - Quality Assurance Unit,” Ihu.edu.gr, 2023. <https://modip.ihu.edu.gr> (accessed May 20, 2024).
- [3] “Why choose Tableau?” Accessed: Apr. 11, 2024. [Online]. Available: <https://www.tableau.com/why-tableau>
- [4] S. Batt, T. Grealis, O. Harmon, and P. Tomolonis, “Learning Tableau: A data visualization tool,” *Journal of Economic Education*, vol. 51, no. 3–4, pp. 317–328, Aug. 2020, doi: 10.1080/00220485.2020.1804503.
- [5] H. Alasmari, “Sentimental Visualization: Semantic Analysis of Online Product Reviews Using Python and Tableau,” in *Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020. doi: 10.1109/BigData50022.2020.9391769.
- [6] “IBM SPSS Statistics.” Accessed: Apr. 11, 2024. [Online]. Available: <https://www.ibm.com/products/spss-statistics>
- [7] “IBM Cognos Analytics.” Accessed: Apr. 11, 2024. [Online]. Available: <https://www.ibm.com/products/cognos-analytics>
- [8] “IBM Watson Studio.” Accessed: Apr. 11, 2024. [Online]. Available: <https://www.ibm.com/products/watson-studio>
- [9] “Welcome | Superset.” Accessed: Apr. 11, 2024. [Online]. Available: <https://superset.apache.org/>
- [10] M. V Tatas, “October 2022 National and Kapodistrian University of Athens School of Sciences Department of Informatics and Telecommunications BSc Thesis A data analytics framework for COVID-19 analysis.”
- [11] S. Aggarwal, “Modern Web-Development using ReactJS,” 2018.
- [12] M. Persson, “JavaScript DOM Manipulation Performance Comparing Vanilla JavaScript and Leading JavaScript Front-end Frameworks,” 2020. [Online]. Available: [www.bth.se](http://www.bth.se)
- [13] M. A. J. #1, B. R. Sawant, and A. Deshmukh, “Single Page Application using AngularJS.” [Online]. Available: <http://mydomain.com/myseo#key=value>
- [14] J. Song, M. Zhang, and H. Xie, “Design and implementation of a Vue.js-based college teaching system,” *International Journal of Emerging Technologies in Learning*, vol. 14, no. 13, pp. 59–69, 2019, doi: 10.3991/ijet.v14i13.10709.
- [15] “React Frameworks - HotFrameworks.” Accessed: Apr. 12, 2024. [Online]. Available: <https://hotframeworks.com/frameworks/react/>

- [16] L. Li and W. Chou, “Design and describe REST API without violating REST: A Petri Net based approach,” in *Proceedings - 2011 IEEE 9th International Conference on Web Services, ICWS 2011*, IEEE Computer Society, 2011, pp. 508–515. doi: 10.1109/ICWS.2011.54.
- [17] Decasper, D., Dittia, Z., Parulkar, G., & Plattner, B. (2000). Router plugins: a software architecture for next-generation routers. *IEEE/ACM Transactions on Networking*, 8(1), 2–15. doi:10.1109/90.836474
- [18] D. Oktaria, J. Andrew, M. K. Ginting, M. Abdurohman, and R. Yasirandi, “Design of API Gateway as Middleware on Platform as a Service”, doi: 10.34818/indojc.2021.6.3.597.
- [19] H. Y. Yang, E. Tempero, and H. Melton, “An Empirical Study into Use of Dependency Injection in Java,” 2008, doi: 10.1109/ASWEC.2008.35.
- [20] H. Singh and S. Imtiyaz Hassan, “Effect of SOLID Design Principles on Quality of Software: An Empirical Assessment,” *Int J Sci Eng Res*, vol. 6, no. 4, 2015, [Online]. Available: <http://www.ijser.org>
- [21] W. E. Ir, “Clean Architecture.” [Online]. Available: [www.EBooksWorld.ir](http://www.EBooksWorld.ir)
- [22] T. H. Chen *et al.*, “An empirical study on the practice of maintaining object-relational mapping code in Java systems,” in *Proceedings - 13th Working Conference on Mining Software Repositories, MSR 2016*, Association for Computing Machinery, Inc, May 2016, pp. 165–176. doi: 10.1145/2901739.2901758.
- [23] “Database: Seeding - Laravel 11.x - The PHP Framework For Web Artisans.” Accessed: Apr. 12, 2024. [Online]. Available: <https://laravel.com/docs/11.x/seeding>
- [24] H. Sampath, A. Merrick, and A. MacVean, “Accessibility of command line interfaces,” in *Conference on Human Factors in Computing Systems - Proceedings*, Association for Computing Machinery, May 2021. doi: 10.1145/3411764.3445544.
- [25] A. Zainudin *et al.*, *Proceedings, IES 2019: IES, International Electronics Symposium: Surabaya, Indonesia, September 27-28, 2019: the Role of Techno-intelligence in Creating an Open Energy System Towards Energy Democracy*.
- [26] “GitHub - dotnet/aspnetcore: ASP.NET Core is a cross-platform.NET framework for building modern cloud-based web applications on Windows, Mac, or Linux.” Accessed: Apr. 12, 2024. [Online]. Available: <https://github.com/dotnet/aspnetcore>
- [27] “Laravel - The PHP Framework For Web Artisans.” Accessed: Apr. 12, 2024. [Online]. Available: <https://laravel.com/>
- [28] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, “A decade of agile methodologies: Towards explaining agile software development,” *Journal of Systems and Software*, vol. 85, no. 6. Elsevier Inc., pp. 1213–1221, 2012. doi: 10.1016/j.jss.2012.02.033.
- [29] A. Zainudin *et al.*, *Proceedings, IES 2019: IES, International Electronics Symposium: Surabaya, Indonesia, September 27-28, 2019: the Role of Techno-intelligence in Creating an Open Energy System Towards Energy Democracy*.
- [30] “GitHub - pallets/flask: The Python micro framework for building web applications.” Accessed: Apr. 12, 2024. [Online]. Available: <https://github.com/pallets/flask>

- [31] S. Brown, “The C4 model for visualising software architecture,” 2015. [Online]. Available: <http://leanpub.com/visualising-software-architecture>
- [32] L. Spiru, M. Marzan, C. Paul, M. Velciu, and A. Garleanu, “The reversed moscow method. a general framework for developing age-friendly technologies,” in *Multi Conference on Computer Science and Information Systems, MCCSIS 2019 - Proceedings of the International Conference on e-Health 2019*, IADIS Press, 2019, pp. 75–81. doi: 10.33965/eh2019\_2019101010.
- [33] Oktafiani, I., & Hendradjaya, B. (2018). Software Metrics Proposal for Conformity Checking of Class Diagram to SOLID Design Principles. 2018 5th International Conference on Data and Software Engineering (ICoDSE). doi:10.1109/icodse.2018.8705857.
- [34] A. Cardoso, G. R. Alves, T. Restivo, IEEE Education Society, and Institute of Electrical and Electronics Engineers, *Proceedings of the 2020 IEEE Global Engineering Education Conference (EDUCON) : date and venue, 27-30 April, 2020, Porto, Portugal*.
- [35] I.-Y. Song, M. Evans, and U. E. K. Park, “A Comparative Analysis of Entity-Relationship Diagrams 1,” 1995.
- [36] “openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files — openpyxl 3.1.2 documentation.” Accessed: May 03, 2024. [Online]. Available: <https://openpyxl.readthedocs.io/en/stable/>
- [37] “Psycopg – PostgreSQL database adapter for Python — Psycopg 2.9.9 documentation.” Accessed: May 03, 2024. [Online]. Available: <https://www.psycopg.org/docs/>
- [38] A. Soni and V. Ranga, “API features individualizing of web services: REST and SOAP,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 9 Special Issue, pp. 664–671, Jul. 2019, doi: 10.35940/ijitee.I1107.0789S19.
- [39] H. Wenhui, H. Yu, L. Xueyang, and X. Chen, “Study on REST API Test Model Supporting Web Service Integration,” in *2017 IEEE 3rd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS)*, IEEE, May 2017, pp. 133–138. doi: 10.1109/BigDataSecurity.2017.35.
- [40] M. Nottingham and R. Fielding, “Additional HTTP Status Codes,” Apr. 2012, doi: 10.17487/RFC6585.
- [41] “Manifesto for Agile Software Development.” Accessed: May 03, 2024. [Online]. Available: <https://agilemanifesto.org/>
- [42] M. Fowler and J. Highsmith, “The Agile Manifesto,” 2001. [Online]. Available: [www.martinfowler.com/articles/newMethodology.html](http://www.martinfowler.com/articles/newMethodology.html)
- [43] Vazquez-Ingelmo, A., Garcia-Holgado, A., & Garcia-Penalvo, F. J. (2020). C4 model in a Software Engineering subject to ease the comprehension of UML and the software. 2020 IEEE Global Engineering Education Conference (EDUCON). doi:10.1109/educon45650.2020.9125335

