



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Ανάπτυξη Διαδικτυακής Εφαρμογής για
την Ανάλυση Δεδομένων Ηλεκτρονικής Μάθησης στο
Σύστημα Moodle χρησιμοποιώντας Τεχνικές Οπτικοποίησης**

**Βασιλειάδης Νικόλαος
Καλατζής Θεόδωρος**

Εισηγητής: Δρ Ιωάννης Ψαρομήλιγκος, Καθηγητής

**Αιγάλεω
Σεπτέμβριος 2018**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη Διαδικτυακής Εφαρμογής για την Ανάλυση Δεδομένων Ηλεκτρονικής Μάθησης στο Σύστημα Moodle χρησιμοποιώντας Τεχνικές Οπτικοποίησης

Νικόλαος Βασιλειάδης

A.M. 42432

Θεόδωρος Καλατζής

A.M. 42930

Εισηγητής:

Δρ. Ιωάννης Ψαρομήλιγκος, Καθηγητής

Εξεταστική Επιτροπή:

Γεώργιος Πρεζεράκος, Καθηγητής

Κωνσταντίνος Κουκουλέτσος, Καθηγητής

Ημερομηνία εξέτασης: 28/09/2018

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΩΝ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Οι κάτωθι υπογεγραμμένοι:

Βασιλειάδης Νικόλαος του **Ιωάννη**, με αριθμό μητρώου **42432**

Καλατζής Θεόδωρος του **Προκοπίου**, με αριθμό μητρώου **42930**

φοιτητές του τμήματος **Μηχανικών Η/Υ Συστημάτων Τ.Ε.** του **Α.Ε.Ι. Πειραιά Τ.Τ.** πριν αναλάβουμε την εκπόνηση της Πτυχιακής Εργασίας μας, δηλώνουμε ότι ενημερωθήκαμε για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασης της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

ΕΥΧΑΡΙΣΤΙΕΣ

«Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή κύριο Ιωάννη Ψαρομήλιγκο κυρίως για την εμπιστοσύνη που μας έδειξε, και την υπομονή που έκανε κατά τη διάρκεια υλοποίησης της πτυχιακής εργασίας. Όπως επίσης και για την πολύτιμη βοήθεια και καθοδήγηση του, για την επίλυση διάφορων θεμάτων. Τέλος θα ήθελα να ευχαριστήσω τον συμφοιτητή μου και πολύ καλό μου φίλο ΤΕΟ - παρά τις παραξενιές του, τον άντεξα - που μαζί καταφέραμε να ολοκληρώσουμε την πτυχιακή.»

Βασιλειάδης Νικόλαος

«Αυτή η εργασία σηματοδοτεί το τέλος ενός πολύ ξεχωριστού ταξιδιού. Ενός ταξιδιού γνώσης και εμπειριών, με πολλές ευχάριστες στιγμές, δίπλα σε τόσο υπέροχους και ξεχωριστούς ανθρώπους. Θα ήθελα να αφιερώσω ένα πολύ ιδιαίτερο ευχαριστώ στην οικογένειά μου και τους φίλους μου, που ήταν πάντα δίπλα μου σε αυτή τη διαδρομή, και ιδιαίτερη μνεία στη μοναδική παγέα “Μηχανικάγες” -με Γάμμα-, που περάσαμε τόσες καθημερινές στιγμές μαζί, πάντα με αφορμή το διάβασμα!

Επίσης, θα ήθελα να ευχαριστήσω τον καθηγητή μας κύριο Ιωάννη Ψαρομήλιγκο, για όλη την πολύτιμη βοήθεια που μας προσέφερε, την πίστη και τη συνεχή στήριξή του σε εμάς, και που άντεξε τις πολύωρες εβδομαδιαίες Skype συναντήσεις που είχαμε κάθε Τετάρτη επί αρκετούς μήνες, ώστε να καταφέρουμε να κυνηγήσουμε τα όνειρα μας. Τέλος, τον συμφοιτητή και πλέον καλό μου φίλο Νίκο που -παρά τις παραξενιές μου- με άντεξε και καταφέραμε να ολοκληρώσουμε μαζί αυτή την εργασία.

Σας ευχαριστώ.»

Καλατζής Θεόδωρος

Περίληψη

Η Ηλεκτρονική Μάθηση αποτελεί πλέον σημαντικό εργαλείο στην εκπαιδευτική διαδικασία, σε όλες τις βαθμίδες εκπαίδευσης σε ολόκληρο τον κόσμο. Υπάρχουν διαθέσιμα αρκετά Συστήματα Διαχείρισης Μάθησης, με πιο γνωστό ανοιχτού κώδικα το Moodle. Το Moodle είναι εύκολο στη χρήση τόσο για τους μαθητές, όσο και τους καθηγητές, ενώ παρέχει πολλά εργαλεία για την αξιολόγηση των μαθητευόμενων. Παρέχει επίσης στον καθηγητή βασικά στατιστικά στοιχεία και γραφήματα για την επίδοση των μαθητών στα μαθήματά του. Ενώ παρέχονται αρκετά εργαλεία, δεν είναι πάντα εύχρηστα στον καθηγητή ή δεν έχουν τη δυνατότητα για περαιτέρω παραμετροποίηση. Αυτή η εργασία έχει στόχο τη δημιουργία μίας διαδικτυακής εφαρμογής, η οποία θα αξιοποιήσει τα αρχεία καταγραφής και άλλων στοιχείων, ώστε να συμπληρώσει ή ακόμη και να προσθέσει λειτουργίες στο Moodle και να παρέχονται λειτουργίες και διαγράμματα με αυξημένη πολυπλοκότητα.

Επιστημονική περιοχή: Μαθησιακή Αναλυτική

Λέξεις κλειδιά: learning analytics, moodle, οπτικοποίηση, διαγράμματα, στατιστικά

Abstract

Nowadays, e-Learning has been an important tool in the educational process of all education levels, all over the world. There are several Learning Management Systems available in the market, with Moodle being the most famous open-source LMS. Moodle is easy to use, both for students and teachers, and provides many tools for evaluating students' progress. Also, it provides the teacher with simple statistics and graphs for students' performance in his lessons. While Moodle provides several tools, teachers do not find them practical and cannot be further parameterized. This study aims the creation of an online application that will use logs and other data, to supplement or even add functionality on Moodle and provide features and charts of increased complexity.

Scientific field: Learning Analytics

Keywords: learning analytics, moodle, visualization, charts, statistics

Περιεχόμενα

Κεφάλαιο 1: Ηλεκτρονική Μάθηση & Συστήματα.....	15
1.1 Εισαγωγή.....	15
1.2 Η Έννοια της Ηλεκτρονικής Μάθησης	15
1.3 Μορφές της Ηλεκτρονικής Μάθησης.....	16
1.3.1 Σύγχρονη	16
1.3.2 Ασύγχρονη	16
1.4 Συστήματα Διαχείρισης Μάθησης.....	17
1.5 Η πλατφόρμα του Moodle.....	18
1.6 Τι είναι το Learning Analytics	18
1.7 Γιατί Learning Analytics	19
Κεφάλαιο 2: Ανάλυση Δεδομένων Ηλεκτρονικής Μάθησης στο Moodle.....	21
2.1 Αναφορές και Γραφικές απεικονίσεις	23
2.2 Αναφορές και Φίλτρα	24
2.3 Βελτιώσεις στις αναφορές του Moodle.....	24
Κεφάλαιο 3: Ανάπτυξη Εφαρμογής Ανάλυσης Δεδομένων Ηλεκτρονικής Μάθησης	27
3.1 Η αρχιτεκτονική της Εφαρμογής μας	27
3.2 Backend.....	28
3.2.1 Εισαγωγή	28
3.2.2 Ruby on Rails: Routes, Models και Controllers.....	28
3.2.2.i Τι είναι η Ruby και το Rails;	28
3.2.2.ii Routes.....	29
3.2.2.iii Models (Active Model).....	29
3.2.2.iv Controllers.....	30
3.2.3 Ανάλυση Δομής.....	31
3.2.4 Εγκατάσταση εφαρμογής	33
3.2.4.i Προετοιμασία περιβάλλοντος.....	33
3.2.4.ii Προετοιμασία υπηρεσιών για σύνδεση	33
3.2.4.iii Σύνδεση της εφαρμογής με τις απαιτούμενες υπηρεσίες	35
3.2.5 Προσθήκη νέας λειτουργίας	36
3.2.6 Πίνακας υλοποιημένων endpoints.....	37
3.3 Frontend	41

3.3.1 Εισαγωγή	41
3.3.2 Λειτουργίες	41
3.3.2.i Πρώτη εντύπωση	42
3.3.2.ii Dashboard	42
3.3.2.iii Courses	43
3.3.2.iv Risk Analysis	45
3.3.3 Τεχνικές Λεπτομέρειες	47
3.3.3.i Εργαλεία και Frameworks	48
3.3.3.ii Αρχιτεκτονική και Δομή	48
3.3.3.iii Παράδειγμα με κώδικα	51
3.3.3.iv Singleton class	54
3.3.4 UML Διαγράμματα	55
3.3.4.i Use case diagram	55
3.3.4.ii Sequence diagram	56
Κεφάλαιο 4: Συμπεράσματα	57
4.1 Προτάσεις για μελλοντική ανάπτυξη	58
Βιβλιογραφία	59
Παράρτημα Α': Λειτουργικό αντίγραφο της εφαρμογής	63

Κατάλογος Εικόνων

Εικόνα 3.3.1 - Κεντρική οθόνη.....	42
Εικόνα 3.3.2 - Dashboard Views Widget – Toolbar	43
Εικόνα 3.3.3 - Courses submenu	43
Εικόνα 3.3.4 - Κεντρικό πάνελ μαθήματος	44
Εικόνα 3.3.5 - Widget Views - Modules Dropdown.....	44
Εικόνα 3.3.6 - Initialize Activities	45
Εικόνα 3.3.7 - Risk Analysis Overview - Step One.....	46
Εικόνα 3.3.8 - Risk Analysis Overview - Results	47
Εικόνα 3.3.9 - ExtJS File Structure.....	49

Κατάλογος Διαγραμμάτων

Διάγραμμα 3.1.1 - Βασικό Διάγραμμα Αρχιτεκτονικής Εφαρμογής.....	27
Διάγραμμα 3.2.1 - Σύντομο διάγραμμα δομής της Rails εφαρμογής	31
Διάγραμμα 3.2.2 - Αναλυτικό διάγραμμα δομής της Rails εφαρμογής.....	32
Διάγραμμα 3.3.1 - MVVM.....	48
Διάγραμμα 3.3.2 - ExtJS Pattern.....	50
Διάγραμμα 3.3.3 - Use Case Diagram	55
Διάγραμμα 3.3.4 - Sequence Diagram	56

Κατάλογος Πινάκων

Πίνακας 1.3.1 - Μορφές ηλεκτρονικής μάθησης	16
Πίνακας 3.1.1 - Οι ελεγκτές και τα αντικρίσματά τους.....	30

Κεφάλαιο 1: Ηλεκτρονική Μάθηση & Συστήματα

1.1 Εισαγωγή

Η ηλεκτρονική μάθηση αποτελεί ένα σημαντικό κομμάτι του εκπαιδευτικού συστήματος και έχει βελτιώσει σημαντικά τόσο τη σύγχρονη όσο και την ασύγχρονη εκπαίδευση. Τα Συστήματα Διαχείρισης Μάθησης είναι ευρέως διαδεδομένα και πλέον αποτελούν αναπόσπαστο κομμάτι σε όλες τις βαθμίδες εκπαίδευσης. Επιτρέπουν στους εκπαιδευτικούς να διαχειρίζονται και να διαθέτουν διαδικτυακά το εκπαιδευτικό τους υλικό ώστε να είναι διαθέσιμο στη ευχέρεια του εκπαιδευόμενου, παρέχοντας πολλά ακόμη εργαλεία και δυνατότητες, ενδεικτικά: την εγγραφή μαθητών και την ανάλυση απόδοσης των εκπαιδευόμενων.

Η χρήση νέων τεχνολογιών σε συνδυασμό με τη χρήση νέων παιδαγωγικών μεθόδων δημιουργούν νέους δρόμους στον τομέα της μάθησης. Τα τελευταία χρόνια έχουν πραγματοποιηθεί πολλές έρευνες για τη χρήση της τεχνολογίας ως εκπαιδευτικό εργαλείο, εστιάζοντας κυρίως σε θέματα μαθησιακής τεχνολογίας, διαδικτυακής μάθησης και συνεργατικής μάθησης μέσω υπολογιστή. Σύμφωνα με έρευνες, οι εκπαιδευτικοί έχουν εκφράσει διάθεση να χρησιμοποιήσουν περισσότερα εργαλεία ηλεκτρονικής μάθησης, ενώ και οι μαθητές δηλώνουν ότι μπορούν να μελετήσουν πιο παραγωγικά με χρήση τέτοιων εργαλείων.

Σύμφωνα με τους Connolly και Stansfield (2006) η ηλεκτρονική μάθηση έχει περάσει από τρεις ξεχωριστές γενιές. Η πρώτη γενιά, από το 1994 μέχρι το 1999, χαρακτηρίστηκε από μια παθητική χρήση του Διαδικτύου, όπου τα παραδοσιακά υλικά απλώς μεταφέρθηκαν σε ηλεκτρονική μορφή. Η δεύτερη γενιά από το 2000 έως το 2003 χαρακτηρίστηκε από τη μετάβαση σε υψηλότερα εύρη ζώνης (bandwidth), πλούσια μέσα για streaming, αυξημένους πόρους και η κίνηση για να δημιουργηθούν εικονικά περιβάλλοντα μάθησης που ενσωματώνουν την πρόσβαση σε εκπαιδευτικό υλικό, επικοινωνίες και φοιτητικές υπηρεσίες. Η τρίτη γενιά, που βρίσκεται σε εξέλιξη, χαρακτηρίζεται από την ενσωμάτωση μιας μεγαλύτερης συνεργασίας, της κοινωνικοποίησης, της μάθησης μέσω πρότζεκτ (project based learning) και των reflective practices, μέσω ηλεκτρονικών χαρτοφυλακίων (e-portfolios), wikis, blogs, κοινωνικά μέσα δικτύωσης, καθώς και προσομοιώσεων. Επιπλέον, η τρίτη γενιά επηρεάζεται όλο και περισσότερο από τις εξελίξεις στον τομέα των υπολογιστών κινητής τηλεφωνίας

1.2 Η Έννοια της Ηλεκτρονικής Μάθησης

Η έννοια «Ηλεκτρονική Μάθηση», ή όπως συνηθίζεται να χρησιμοποιείται ο αγγλικός όρος «e-Learning», είναι αρκετά γενική και περιλαμβάνει οποιαδήποτε μορφή εκπαίδευσης χρησιμοποιεί τους πόρους του δικτύου ή γενικότερα τις δυνατότητες των ηλεκτρονικών

υπολογιστών. Η χρήση της έχει πολλά πλεονεκτήματα, τόσο για τους εκπαιδευτικούς όσο και για τους εκπαιδευόμενους, με αυτά να συνοψίζονται στα παρακάτω χαρακτηριστικά:

- Η ευελιξία σε χρόνο και τόπο σε σχέση με την παραδοσιακή διδασκαλία μπορεί να προσελκύσει πολύ μεγάλο αριθμό συμμετεχόντων όπως εργαζόμενοι και άτομα με πολλαπλές ασχολίες.
- Το γεγονός ότι ακόμα και άτομα με ειδικές ανάγκες μπορούν να έχουν πρόσβαση σε ένα ηλεκτρονικό μάθημα.
- Η διευκόλυνση της συνεργατικότητας και της επικοινωνίας μεταξύ των εκπαιδευόμενων, ακόμα και αυτών που δεν είχαν αρχικά εκπαιδευτεί σε μια τέτοια στάση και συμπεριφορά.
- Το γεγονός ότι σε ένα ηλεκτρονικό μάθημα μπορούν να συμμετέχουν άτομα με διαφορετικό υπόβαθρο και στυλ μάθησης (learning style).
- Η δυνατότητα επικαιροποίησης και αναβάθμισης του μαθησιακού περιεχομένου.

1.3 Μορφές της Ηλεκτρονικής Μάθησης

Η ηλεκτρονική μάθηση μπορεί να είναι σύγχρονη, ασύγχρονη ή και ακόμη και συνδυασμός και των δύο. Έχει πολλαπλές μορφές, όπως κείμενο, γραφικά, ήχο, βίντεο, καθώς και συνδυασμούς αυτών, προσφέροντας πολλαπλές δυνατότητες και εμπειρίες μάθησης.

	Ίδιο μέρος	Διαφορετικό Μέρος
Ίδια χρονική στιγμή	Παραδοσιακή Μάθηση	Σύγχρονη Μάθηση από Απόσταση
Διαφορετικές χρονικές στιγμές		Ασύγχρονη Μάθηση από Απόσταση

Πίνακας 1.3.1 - Μορφές ηλεκτρονικής μάθησης

1.3.1 Σύγχρονη

Σε αυτή την περίπτωση, το μάθημα γίνεται την ίδια χρονική στιγμή, ενώ οι εκπαιδευτές και οι εκπαιδευόμενοι μπορούν να βρίσκονται σε διαφορετικό τόπο, όπου χρησιμοποιώντας τεχνολογίες τηλεδιάσκεψης βρίσκονται όλοι σε μία εικονική αίθουσα διδασκαλίας. Η διεξαγωγή του μαθήματος γίνεται με τέτοιο τρόπο ώστε να προσφέρονται οι ίδιες ή και παραπάνω δυνατότητες με αυτές που προσφέρονται σε μία κανονική αίθουσα. Ένα παράδειγμα χρήσης της σύγχρονης ηλεκτρονικής μάθησης, είναι όταν ο εκπαιδευόμενος δεν μπορεί να είναι βρίσκεται στον ίδιο χώρο λόγω απόστασης ή λόγω έλλειψης χρόνου για τη μεταφορά τους στον ίδιο χώρο. Επίσης, ο εκπαιδευόμενος έχει την δυνατότητα να ρωτήσει κατευθείαν τον καθηγητή για τυχόν απορίες.

1.3.2 Ασύγχρονη

Οι εκπαιδευόμενοι έχουν τη δυνατότητα να παρακολουθήσουν διαδίκτυακά το εκπαιδευτικό υλικό όποτε και όσες φορές επιθυμούν. Η επικοινωνία τόσο με τον καθηγητή

όσο και με τους υπόλοιπους συμμετέχοντες γίνεται μέσω ηλεκτρονικών μέσων. Το εκπαιδευτικό υλικό μπορεί να διατεθεί εξ' ολοκλήρου από την έναρξη των μαθημάτων ή σταδιακά. Ο ρυθμός διεξαγωγής καθορίζεται από τον καθηγητή σε συνεργασία με τους εκπαιδευόμενους. Παράδειγμα χρήσης της ασύγχρονης μάθησης είναι οι ηλεκτρονικές πλατφόρμες εκμάθησης προγραμματισμού, όπου υπάρχει βιντεοσκοπημένο υλικό και εργασίες ή quiz γνώσεων σε κάθε ενότητα για να συνεχίσει στην επόμενη ενότητα. Σε αυτές τις πλατφόρμες, υπάρχει δυνατότητα επικοινωνίας με τον καθηγητή μέσω email.

1.4 Συστήματα Διαχείρισης Μάθησης

Ένα σύστημα διαχείρισης μάθησης (Learning Management System - LMS) είναι μια πλατφόρμα για τη διαχείριση, την τεκμηρίωση, την παρακολούθηση, την αναφορά και την παράδοση εκπαιδευτικών μαθημάτων ή προγραμμάτων κατάρτισης.

Το ΣΔΜ είναι η κινητήρια δύναμη της ηλεκτρονικής μάθησης και εξ' αποστάσεως εκπαίδευσης, και στην πιο κοινή μορφή αποτελείται από δύο ξεχωριστά μέρη:

- Έναν server που εκτελεί τις βασικές λειτουργίες (δημιουργία, διαχείριση και παράδοση μαθημάτων, έλεγχος ταυτότητας χρηστών, προβολή δεδομένων και ειδοποιήσεων κ.λπ.)
- Ένα user interface που λειτουργεί μέσα από έναν φυλλομετρητή (browser), το οποίο χρησιμοποιείται από διαχειριστές, εκπαιδευτές και φοιτητές.

Τα ΣΔΜ χρησιμοποιούνται σε όλες τις βαθμίδες εκπαίδευσης (πρωτοβάθμια, δευτεροβάθμια και τριτοβάθμια), επειδή προσφέρουν αρκετά πλεονεκτήματα σε εκπαιδευόμενους, εκπαιδευτές και στους εκπαιδευτικούς οργανισμούς, αίροντας χωρικούς και χρονικούς περιορισμούς στη διαδικασία μάθησης.

Τα ΣΔΜ είναι ενσωματωμένα (integrated) συστήματα λογισμικού που εμπεριέχουν διάφορα εργαλεία στους εκπαιδευτικούς σχεδιαστές και στους διαχειριστές, αναφορικά με την οργάνωση και τη διανομή του μαθησιακού υλικού, τη διαχείριση των μαθημάτων, την αξιολόγηση των μαθητών, τα εργαλεία επικοινωνίας και συνεργασίας, τη διαχείριση των εκπαιδευόμενων σε ένα οργανισμό, κλπ.

Υπάρχουν πάρα πολλά συστήματα διαχείρισης μάθησης, τόσο εμπορικά όσο και ανοικτού κώδικα (open source). Μερικά παγκοσμίως γνωστά ΣΔΜ είναι τα Docedo (εμπορικό), Litmos LML (εμπορικό) και το Moodle (ανοικτού κώδικα). Το Moodle είναι ένα από τα κυρίαρχα ΣΔΜ, και θα αναφερθούμε σε αυτό στο κεφάλαιο 1.5 .

Τα ΣΔΜ χρησιμοποιούνται παγκοσμίως, σε πολλές διαφορετικές βιομηχανίες και για μια ποικιλία διαφορετικών περιπτώσεων χρήσης. Η υιοθέτηση των ΣΔΜ έχει αυξηθεί τα τελευταία χρόνια σε όλο τον κόσμο. Στην πραγματικότητα, η παγκόσμια αγορά ΣΔΜ αναμένεται να αυξηθεί σε 24% CAGR (Compound Annual Growth Rate) από το 2016 έως το 2020.

Σε ένα πιο συγκεκριμένο επίπεδο, υπάρχουν δύο βασικοί τύποι χρηστών ΣΔΜ:

- **Μαθητές:** Οι μαθητές που έχουν πρόσβαση στο ΣΔΜ θα μπορούν να δουν τον κατάλογο μαθημάτων τους, να ολοκληρώσουν τα καθορισμένα μαθήματα και τις αξιολογήσεις τους και να μετρήσουν τη δική τους πρόοδο.
- **Διαχειριστές:** Είναι υπεύθυνοι για τη διαχείριση του ΣΔΜ, συμπεριλαμβανομένης της δημιουργίας μαθημάτων, την ανάθεση συγκεκριμένων ομάδων μαθητών σε συγκεκριμένα προγράμματα μάθησης και την παρακολούθηση της προόδου των μαθητών τους.

1.5 Η πλατφόρμα του Moodle

Το Moodle (Moodle.org) είναι ένα ευρέως γνωστό ΣΔΜ ανοικτού κώδικα. Διαθέτει λεπτομερείς οδηγούς και χρήσιμες συμβουλές για το πώς να δημιουργήσετε το δικό σας ΣΔΜ, να δημιουργήσετε ηλεκτρονικά μαθήματα και να διδάξετε με το Moodle. Επίσης, διαθέτει μία μεγάλη και ενεργή κοινότητα χρηστών που αλληλεπιδρούν σε διάφορα θέματα. Πολύ σημαντικό είναι το γεγονός ότι διατίθεται δωρεάν, με άδεια GNU/GPL ως ελεύθερο λογισμικό / λογισμικό ανοικτού κώδικα και υπάρχει διαθέσιμη εφαρμογή για κινητά.

Δημιουργήθηκε το 1999 από τον Αυστραλό Martin Dougiamas. Το όνομα Moodle είναι το ακρώνυμο του Modular Object-Oriented Dynamic Learning Environment (Αρθρωτό Αντικειμενοστραφές Δυναμικό Μαθησιακό Περιβάλλον). Πρόκειται για ένα πολύ ευέλικτο, αρθρωτό λογισμικό, το οποίο ενδείκνυται για εξ' αποστάσεως μαθήματα, αλλά και για μεικτή μάθηση, καθώς συχνά χρησιμοποιείται ως υποστηρικτικό μέσο της εκπαιδευτικής διαδικασίας σε όλες τις βαθμίδες εκπαίδευσης. Τέλος, αξίζει να σημειωθεί ότι έχει μεταφραστεί σε εβδομήντα (70) γλώσσες μεταξύ των οποίων και η ελληνική και χρησιμοποιείται από το Πανελλήνιο Σχολικό Δίκτυο.

1.6 Τι είναι το Learning Analytics

Ένας συχνά χρησιμοποιούμενος ορισμός για το Learning Analytics προέρχεται από το πρώτο διεθνές συνέδριο «Learning Analytics and Knowledge» το 2011:

«Learning analytics is the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs.»

Ο Erik Duval πρότεινε τον ορισμό ότι «το Learning Analytics αφορά τη συλλογή των ιχνών που αφήνουν οι μαθητές κατά τη χρήση των ΣΔΜ και τη χρήση αυτών των ιχνών για τη βελτίωση της μάθησης».

Σε μια παρουσίαση που δόθηκε στο ανώτερο προσωπικό της βιβλιοθήκης, η Rebecca Ferguson τοποθετεί τα μαθησιακά αναλυτικά μέσα σε ένα συνεχές:

- **Υψηλού επιπέδου στοιχεία:** τα οποία μπορούν να προμηθεύσουν μια επισκόπηση εσωτερικών και εξωτερικών αναφορών και να χρησιμοποιηθούν για σκοπούς οργανωτικού προγραμματισμού.

- **Ακαδημαϊκοί αναλυτές:** Στοιχεία σχετικά με τη διατήρηση και την επιτυχία, τα οποία χρησιμοποιεί το ίδρυμα για την αξιολόγηση της απόδοσης.
- **Εξόρυξη εκπαιδευτικών δεδομένων (Educational data mining):** αναζήτηση προτύπων στα δεδομένα.
- **Learning analytics:** Η χρήση δεδομένων, τα οποία μπορεί να περιλαμβάνουν «big data», για την παροχή ευφυσής πληροφορίας για εκπαιδευόμενους και εκπαιδευτικούς.

1.7 Γιατί Learning Analytics

Big data, predictive analytics, data mining - ο κατάλογος συνεχίζεται. Αυτοί οι όροι, που πριν από 20 χρόνια δεν υπήρχαν, αποτελούν πλέον μέρος του καθημερινού λεξικού της εκπαιδευτικής κοινότητας. Περίπου 2,5 Exabytes (~2.500.000 TB) δεδομένων δημιουργούνται καθημερινά, επηρεάζοντας τη ζωή και τη σταδιοδρομία μας με απροσδόκητους τρόπους.

Αλλά τα big data δεν αφορούν τα δεδομένα. Πρόκειται για το τι κάνεις με αυτά. Πρόκειται για την ανάλυση που οδηγεί σε αποφάσεις που επηρεάζουν τους ανθρώπους. Και ένας από τους μεγαλύτερους τομείς όπου η ανάλυση αυτή μπορεί να κάνει τη διαφορά είναι η εκπαίδευση.

Τα learning analytics είναι εκεί όπου τα big data συναντούν τις παραδοσιακές ποσοτικές μεθόδους (quantitative methods) στην εκπαίδευση. Οι κυβερνήσεις, τα πανεπιστήμια και οι πάροχοι των MOOCs (Massive Open Online Course) συλλέγουν στοιχεία για τους μαθητές και το πώς μαθαίνουν. Όλα αυτά τα δεδομένα, ωστόσο, ήταν ως επί το πλείστον ανεκμετάλλευτα μέχρι την αρκετά πρόσφατη ανάπτυξη των μεθόδων και εργαλείων για να το κάνουν.

Μεγάλο μέρος των διαθέσιμων δεδομένων δεν διατίθεται σε καθαρή, καλά οργανωμένη και συγκεντρωμένη μορφή. Υπάρχει σε διάφορες μορφές σε διάφορα συστήματα και τοποθεσίες. Οι αναλυτές σήμερα χρειάζονται τις δεξιότητες για να έχουν πρόσβαση και να μετασχηματίζουν αυτά τα δεδομένα, ώστε να κατανοήσουν καλύτερα όχι μόνο αυτό που γνωρίζουν οι σπουδαστές, αλλά και το πώς το γνωρίζουν. Τα learning analytics και η εξόρυξη εκπαιδευτικών δεδομένων (educational data mining) είναι τα εργαλεία για τη μετατροπή αυτών των δεδομένων σε γνώση και εν τέλει στη βελτίωση της εκπαίδευσης.

Περισσότερο από το 87% των εκπαιδευτικών που ερωτήθηκαν από το The New America Foundation υποστηρίζουν ότι τα analytics μπορούν να αποτελέσουν ένα ισχυρό εργαλείο διδασκαλίας στην τάξη. Αλλά οι σημερινοί εκπαιδευτικοί δεν είναι ακόμη εξοπλισμένοι για να ανταποκριθούν στις απαιτήσεις του ή να επωφεληθούν από τα οφέλη του. Παρά την αυξανόμενη συνάφεια και την ισχυρή υπόσχεση των αναλυτών, υπάρχει σημαντική έλλειψη επαγγελματιών με επαρκή εμπειρία.

Το McKinsey εκτιμά ότι μόνο οι Ηνωμένες Πολιτείες θα μπορούσαν να αντιμετωπίσουν έλλειψη από 140.000 έως 190.000 ατόμων με ισχυρές αναλυτικές δεξιότητες, καθώς και 1.5 εκατομμύρια managers με την ικανότητα να λαμβάνουν αποφάσεις βάσει δεδομένων

από analytics. Το NMC Horizon Report: 2016 Higher Education Edition, που παράγεται από κοινού από την NMC και το EDUCAUSE Learning Initiative, προσδιορίζει τα learning analytics ως μία από τις ψηφιακές στρατηγικές και τεχνολογίες που αναμένεται να εισέλθουν δυναμικά στο άμεσο μέλλον.

Τώρα είναι η στιγμή να εισέλθει κάποιος στον τομέα των learning analytics και να κάνει τη διαφορά στην εκπαίδευση των σημερινών σπουδαστών και των δια βίου μαθητών. Για το σκοπό αυτό απαιτείται μια σταθερή θεμελιώδης κατανόηση των ποσοτικών μεθόδων, σε συνδυασμό με την τεχνογνωσία στην κατασκευή, τη συγχώνευση, τη διαχείριση, τον καθαρισμό και την ανάλυση δεδομένων χρησιμοποιώντας λογισμικό και τεχνολογίες αιχμής.

Κεφάλαιο 2: Ανάλυση Δεδομένων Ηλεκτρονικής Μάθησης στο Moodle

Το Moodle είναι λογισμικό ανοιχτού κώδικα εικονικού περιβάλλοντος μάθησης, είναι σχεδιασμένο γύρω από τις ανάγκες των καθηγητών με πάνω από 80 εκατομμύρια χρήστες παγκοσμίως.

Βασικά χαρακτηριστικά που κάνουν το Moodle ιδανικό εργαλείο στην Ηλεκτρονική Μάθηση:

- Είναι πλήρως διαμορφώσιμο ως προς τον σχεδιασμό και το περιβάλλον του
- Είναι απλό, αποτελεσματικό και συμβατό με φορητές συσκευές
- Μπορεί να επικοινωνεί με άλλα συστήματα, όπως το Google Drive και το YouTube
- Απλή και εύκολη δημιουργία μαθημάτων, διαχείρισης τους και μεταφορά παλιών αρχείων
- Φόρα συζητήσεων και εργαλεία ερωτηματολόγιων σε πάνω από 100 γλώσσες

Το Moodle περιέχει εργαλεία, μεταξύ άλλων:

- Ολοκληρωμένη διαχείριση χρηστών, ελεγχόμενη πρόσβαση & απόδοση ρόλων-ιεραρχίας
- Αξιολόγηση χρηστών
- Παρακολούθηση προόδου
- Οι εκπαιδευόμενοι γνωρίζουν την πορεία & τις υποχρεώσεις τους
- Ασύγχρονη αλλά και σύγχρονη επικοινωνία
- Εξαγωγή αναφορών χρήσης του συστήματος (reports)
- Έλεγχος λογοκλοπής περιεχομένου με χρήση πρόσθετων
- Mobile learning

Για να μπορέσουν τα παραπάνω εργαλεία να λειτουργήσουν αποτελεσματικά, το Moodle διατηρεί λεπτομερή αρχεία καταγραφής για όλες τις δραστηριότητες που επιτελούν οι εκπαιδευόμενοι. Συγκεκριμένα, η βάση δεδομένων του Moodle έχει πάνω από 350 πίνακες, στους οποίους περιγράφονται:

- τα δομικά στοιχεία του Moodle, όπως τα μαθήματα, οι χρήστες, οι ρόλοι, οι δραστηριότητες και οι πόροι ενός μαθήματος
- η σχέση μεταξύ τους, δηλαδή ποιοι πόροι χρησιμοποιούνται σε ένα μάθημα ή ποιοι χρήστες είναι εγγεγραμμένοι στο μάθημα
- καταγραφή των ενεργειών των χρηστών, βαθμολογίες των μαθητών

Στη διαχειριστική σελίδα ενός μαθήματος, προσφέρονται κάποιες αναφορές που βοηθούν τους διαχειριστές των μαθημάτων να έχουν μία γενική εικόνα της κατάστασης. Οι

αναφορές που προσφέρει το Moodle, με τις οποίες μπορεί να γίνει ανάλυση των δεδομένων για μάθηση είναι οι εξής:

- Αναφορές διαχειριστή (Site administration Reports)
- Αναφορές μαθήματος (Course Reports)
- Αναφορές συμμετεχόντων (Participants Reports)
- Αναφορές βαθμολογητή (Grader Reports)
- Αναφορές δραστηριοτήτων (Activity Reports)

Αναλυτικά, οι αναφορές που υπάρχουν διαθέσιμες σε επίπεδο Διαχείρισης Ιστότοπου (Site Administration Reports) είναι:

- Σχολίων (Comments)
- Επισκόπησης του Μαθήματος (Course overview)
- Αρχείων Καταγραφής (Logs)
- Ενεργών Αρχείων Καταγραφής (Live logs)
- Επισκόπησης Στατιστικών (Overview statistics)
- Στατιστικών (Statistics)

Σε επίπεδο μαθήματος (Course Reports) υπάρχουν διαθέσιμες:

- Αρχεία Καταγραφής (Logs)
- Ενεργά Αρχεία Καταγραφής (Live logs)
- Αναφορά Δραστηριότητας (Activity report)
- Επισκόπησης Στατιστικών (Overview statistics)
- Συμμετοχής στο Μάθημα (Course participation)
- Στατιστικών (Statistics)

Σε επίπεδο εκπαιδευόμενων (Participant Reports) είναι:

- Σημερινά Αρχεία καταγραφής (Today's logs)
- Όλες οι Αναφορές (All logs)
- Αναφορά Σύνοψης (Outline report)
- Πλήρης Αναφορά (Complete report)
- Στατιστικών (Statistics)
- Επισκόπηση Βαθμολογιών (Grades overview)
- Βαθμός (Grade)

Σε επίπεδο βαθμολογιών χρήστη (Grader Reports) είναι:

- Αναφορά Βαθμολογητή (Grader Report)
- Ιστορικό Βαθμολογίας (Grade History)
- Αναφορά Εκβάσεων (Outcomes Report)
- Αναφορά Επισκόπησης (Overview Report)
- Ενιαία Προβολή (Single View)
- Αναφορά Χρήστη (User Report)

Επίσης, προσφέρεται η επιλογή για στατιστικές αναφορές για δραστηριότητες τύπου Scorm ή τύπου Quiz, οι οποίες δεν παρέχουν επαρκή ανάλυση των δεδομένων στο πλαίσιο της Αναλυτικής Ακαδημαϊκών Δεδομένων.

2.1 Αναφορές και Γραφικές απεικονίσεις

Παρότι το Moodle διατηρεί ένα λεπτομερές αρχείο καταγραφής, δεδομένου ότι καταγράφει κάθε ενέργεια πλοήγησης που επιτελείται από τους χρήστες του, δεν διαθέτει ένα σύστημα αναφορών που να δίνει στον εκπαιδευτικό τη δυνατότητα για ανάλυση των ακαδημαϊκών δεδομένων. Υπάρχει η δυνατότητα εφαρμογής φίλτρων για εμφάνιση των δεδομένων χρήσης ανά μάθημα, ανά συμμετέχοντα, ανά ημέρα και ανά δραστηριότητα. Ο εκπαιδευτικός μπορεί να χρησιμοποιήσει αυτές τις αναφορές για να αντλήσει πληροφορίες για το ποιοι μαθητές είναι ενεργοί σε κάποιο μάθημα, τι ενέργειες έκαναν και πότε. Εκτός από απλές αναφορές ακατέργαστων, μη επεξεργασμένων δεδομένων που περιέχονται στα αρχεία καταγραφής, το σύστημα δεν παρέχει κάποια εξειδικευμένη λειτουργία ώστε να σχηματίσει ο εκπαιδευτικός μια χρήσιμη εικόνα για τη συμμετοχή ενός μαθητή ή μιας ομάδας μαθητών στο μάθημα.

Στην κατηγορία με αναφορές για τη διαχείριση του ιστότοπου (site administration reports), στην επιλογή «Statistics» έχουμε γραφική απεικόνιση, για το πλήθος των εισόδων και των μοναδικών συνδέσεων, για το πλήθος των επισκέψεων, για των δημοσιεύσεων, για τη συνολική δραστηριότητα (δηλ. επισκέψεις + δημοσιεύσεις), για όλους τους ρόλους του συστήματος σε μια συγκεκριμένη χρονική περίοδο. Η αναφορά «Στατιστικά» προσφέρεται επίσης και σε επίπεδο μαθήματος στην οποία μπορούμε να έχουμε γραφική απεικόνιση για το πλήθος των επισκέψεων, για των δημοσιεύσεων ή για τη συνολική δραστηριότητα (δηλ. επισκέψεις + δημοσιεύσεις), ανά ρόλο του συστήματος για μια καθορισμένη χρονική περίοδο.

Στις αναφορές που αφορούν έναν συγκεκριμένο συμμετέχοντα (participant reports) έχουμε γραφικές αναπαραστάσεις στην κατηγορία «Αναφορές δραστηριοτήτων» που περιέχει τις επιλογές «Σημερινά αρχεία καταγραφής» και «Όλες οι αναφορές», οι οποίες παρουσιάζουν το πλήθος των «Επισκέψεων» στο μάθημα, καθώς και στην αναφορά «Στατιστικά» η οποία παρουσιάζει το πλήθος των επισκέψεων, των δημοσιεύσεων και της όλης δραστηριότητας (επισκέψεις + δημοσιεύσεις) του χρήστη.

Στοιχειώδη γραφική απεικόνιση έχουμε επίσης και στην αναφορά «Graph Report» για δραστηριότητες που ακολουθούν το πρότυπο Scorm, η οποία απεικονίζει το ποσοστό των συμμετεχόντων που πέτυχαν μια συγκεκριμένη βαθμολογία. Παρόμοια γραφική αναπαράσταση βαθμολογιών έχουμε και στην αντίστοιχη αναφορά για δραστηριότητες τύπου Quiz.

Συγκεκριμένα για τις δραστηριότητες τύπου Quiz, υπάρχει πιο λεπτομερής ανάλυση που περιλαμβάνει στοιχεία όπως:

- το όνομα του Quiz
- το όνομα του μαθήματος
- τον αριθμό των βαθμολογημένων πρώτων προσπαθειών
- το συνολικό αριθμό των βαθμολογημένων προσπαθειών
- το Μέσο Όρο βαθμολογίας των αρχικών προσπαθειών
- το Μέσο Όρο βαθμολογίας όλων των προσπαθειών
- το Μέσο Όρο βαθμολογίας των τελικών προσπαθειών
- το Μέσο Όρο βαθμολογίας των υψηλότερων σε επίδοση προσπαθειών
- το Διάμεσο για όλες τις Βαθμολογίες
- την τυπική Απόκλιση για όλες τις βαθμολογίες

2.2 Αναφορές και Φίλτρα

Επίσης, υπάρχει η δυνατότητα εφαρμογής φίλτρου για συγκεκριμένη χρονική περίοδο, στο οποίο υπάρχουν τα εξής διαστήματα: 1-7 ημέρες, 1-4 εβδομάδες, 2-7 ή 2-11 μήνες, το προηγούμενο έτος, μία συγκεκριμένη ημερομηνία ή όλες τις ημέρες. Η δυνατότητα να επιλέξουμε συγκεκριμένο χρονικό διάστημα (από - έως κάποια ημερομηνία) υπάρχει μόνο στην αναφορά «Grade History» που δίνει στοιχεία για το ιστορικό των βαθμολογιών.

Μεταξύ επιπέδων παρατηρείται μία ανομοιογένεια για τις επιλογές που υπάρχουν στον περιορισμό των αναφορών σε συγκεκριμένους χρήστες. Σε επίπεδο μαθήματος (course reports), τα φίλτρα για την επιλογή των χρηστών περιορίζονται, ή σε όλους τους συμμετέχοντες ή σε συγκεκριμένο χρήστη. Στις αναφορές σε επίπεδο ιστότοπου (site) υπήρχε η δυνατότητα εφαρμογής φίλτρου σε χρήστες που κατέχουν ένα συγκεκριμένο ρόλο του συστήματος όπως μαθητής, δάσκαλος, διαχειριστής κ.α.. Η δυνατότητα για επιλογή δύο ή περισσότερων χρηστών (όχι ομάδας), υπάρχει μόνο στις αναφορές για το Ιστορικό των Βαθμολογιών «Grade History».

2.3 Βελτιώσεις στις αναφορές του Moodle

Όπως έχουμε δει μέχρι στιγμής, το Moodle παρέχει πολλά μία μεγάλη γκάμα αναφορών που δίνουν τη δυνατότητα στον εκπαιδευτικό να αναλύσει την κίνηση των μαθημάτων. Παρατηρούμε ότι λαμβάνεται μία παράμετρος (επισκέψεις, δημοσιεύσεις ή βαθμολογίες) υπόψιν, η οποία από μόνη δεν είναι παρέχει αντιπροσωπευτικά αποτελέσματα της μαθησιακής διαδικασίας και απόδοσης.

Μια ολοκληρωμένη εικόνα της συμμετοχής σε συγκεκριμένη δραστηριότητα ή στο σύνολό τους, η οποία δεν περιορίζεται μόνο στον αριθμό των επισκέψεων και των δημοσιεύσεων ενός χρήστη, αλλά λαμβάνει υπόψη δραστηριότητες τύπου Scorm και quiz ή ακόμη και άλλα δομικά στοιχεία, θα ήταν αναμφισβήτητα μία ιδιαίτερα χρήσιμη αναφορά στον καθηγητή.

Οπτική απεικόνιση που θα συγκέντρωνε δεδομένα αλληλεπίδρασης από δύο ή περισσότερες δραστηριότητες/πόρους του μαθήματος, θα βοηθούσε σε μια καλύτερη αξιολόγηση του περιεχομένου και θα αποτελούσε πιθανότατα ένα έναυσμα για την αναθεώρηση και τον ανασχεδιασμό του.

Η δυνατότητα σύγκρισης της απόδοσης ενός μαθητή ή ομάδας μαθητών σε δύο ή περισσότερες δραστηριότητες, με σκοπό την εξέταση μαθησιακών αντικειμένων, θα έδινε τη δυνατότητα να αναγνωριστούν σημεία που χρήζουν βελτίωσης στο περιεχόμενο του μαθήματος.

Στην περίπτωση που σε κάποιες από τις δραστηριότητες συναντούσαμε μεγάλο ποσοστό αποτυχίας ή επιτυχίας, θα μπορούσαμε να δράσουμε προληπτικά επιδίδοντας μεγαλύτερο μέρος της προσοχής μας σε αυτές τις περιπτώσεις. Έτσι ένα μεγάλο ποσοστό αποτυχίας σε δραστηριότητες αξιολόγησης μπορεί να αποτελεί ένδειξη ότι οι ερωτήσεις της εξέτασης δεν γίνονται κατανοητές από τους μαθητές λόγω κακής διατύπωσης τους, ενώ ένα εξαιρετικά μεγάλο ποσοστό επιτυχίας μπορεί να αποτελεί ένδειξη ότι οι ερωτήσεις είναι πολύ εύκολες, κάτι το οποίο καθιστά δύσκολο το διαχωρισμό των καλών από τους κακούς μαθητές. Και στις δύο περιπτώσεις οι ερωτήσεις στις δραστηριότητες εξέτασης χρήζουν αναδιατύπωση.

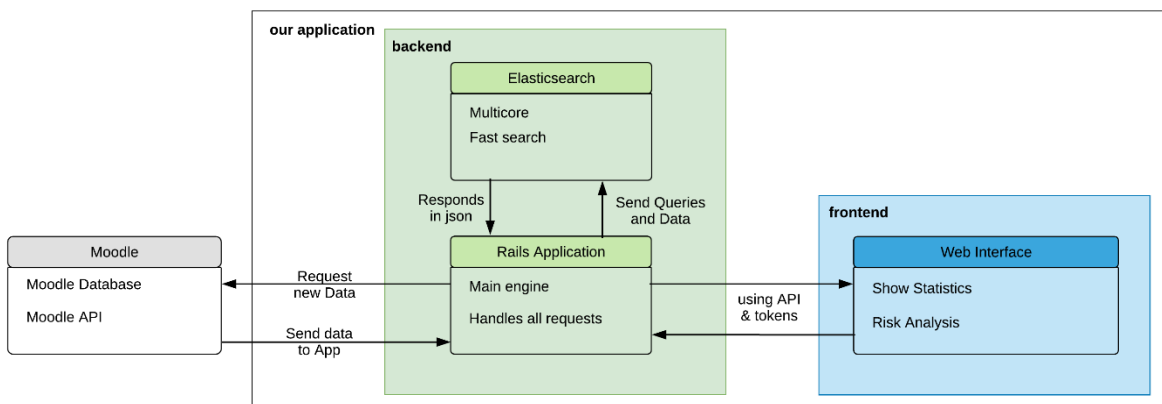
Ενδείξεις που οδηγούν σε πιθανή αναθεώρηση και ανασχεδιασμό των εκπαιδευτικών δραστηριοτήτων και του υλικού που τις υποστηρίζει, μπορεί να έχουμε από τη σύγκριση τις συμπεριφοράς των συμμετεχόντων σε δύο οι περισσότερες δραστηριότητες, ιδιαίτερα στην περίπτωση που μια μαθησιακή δραστηριότητα παρουσιάζει εξαιρετικά χαμηλό βαθμό προτίμησης σε σχέση με άλλες.

Οπτική αναπαράσταση της απόδοσης μαθητών σε διαφορετικές μαθησιακές δραστηριότητες βοηθάει επίσης σε μια παρακολούθηση, σε μια ιχνηλάτιση της μαθησιακής πορείας τους και πιθανότατα προσφέρει ενδείξεις για να ανακαλύψουμε από νωρίς τάσεις που οδηγούν σε χαμηλές αποδόσεις στο μάθημα. Τέτοιες ενδείξεις μπορεί να αποτελέσουν την αφετηρία για στρατηγική θεώρηση της εκπαιδευτικής προσέγγισης που ακολουθούμε και για μελλοντικές δράσεις.

Κεφάλαιο 3: Ανάπτυξη Εφαρμογής Ανάλυσης Δεδομένων Ηλεκτρονικής Μάθησης

3.1 Η αρχιτεκτονική της Εφαρμογής μας

Όπως φαίνεται στο Διάγραμμα 3.1.1 - Βασικό Διάγραμμα Αρχιτεκτονικής Εφαρμογής, η εφαρμογή μας χωρίζεται σε 2 κύρια μέρη, το Backend και το Frontend, και επικοινωνεί και με το Moodle.



Διάγραμμα 3.1.1 - Βασικό Διάγραμμα Αρχιτεκτονικής Εφαρμογής

Αποφασίσαμε να τα κρατήσουμε το Backend και το Frontend ανεξάρτητα, ώστε:

- Να μπορούμε να τα ενημερώνουμε ανεξάρτητα.
- Να μην περιοριστούμε στις βιβλιοθήκες γραφημάτων της Ruby.
- Να μπορούν να επεκταθούν ανεξάρτητα (π.χ. εφαρμογή σε κινητό).
- Να χρησιμοποιήσουμε περισσότερες τεχνολογίες αιχμής.

Αρχικά, το Backend χωρίζεται σε δύο επιμέρους τμήματα (αναλυτικά κεφ. 3.2.3 Ανάλυση Δομής, σελ.:31), την εφαρμογή Rails και το Elasticsearch. Η εφαρμογή Rails, επικοινωνεί με όλες τις απαραίτητες υπηρεσίες (Elasticsearch & Moodle), για να εξυπηρετήσει τα αιτήματα που δέχεται από το Frontend (αναλυτικά κεφ. 3.3.3.ii Αρχιτεκτονική και Δομή, σελ.: 48).

Το Elasticsearch χρησιμοποιείται λόγω της εξαιρετικής ταχύτητας που προσφέρει στην ανάκτηση των logs, σε αντίθεση με τη MySQL βάση δεδομένων του Moodle. Εκτός της ταχύτητας που προσφέρει, προσφέρει εργαλεία για περαιτέρω φιλτράρισμα των πληροφοριών καθώς και πολλές επιλογές για γκρουπάρισμα. Για να χρησιμοποιήσουμε το Elasticsearch, πρέπει να εισάγουμε/αντιγράψουμε τα logs του Moodle στο Elasticsearch (Παράρτημα Α': - Βήμα 13ο: – σελ.: 66).

Η εφαρμογή για το Frontend είναι ανεπτυγμένη σε JavaScript με χρήση του framework ExtJS, με το οποίο έχουμε τη δυνατότητα να φτιάξουμε μία όμορφη εφαρμογή με φιλικό User Interface. Επίσης, έχει πληθώρα βιβλιοθηκών και έτοιμων εργαλείων, τα οποία βοηθούν να προστίθενται νέες λειτουργίες εύκολα και γρήγορα.

3.2 Backend

3.2.1 Εισαγωγή

Σύνοψη κεφαλαίου:

Σε αυτό το κεφάλαιο θα αναλυθεί η δομή της εφαρμογής που αναπτύχθηκε, οι απαραίτητες ενέργειες που πρέπει να γίνουν στο Moodle ώστε να επικοινωνεί σωστά με την εφαρμογή μας, ο τρόπος εγκατάστασης της εφαρμογής και τα βήματα για την περαιτέρω ανάπτυξή της.

Δομή:

Η εφαρμογή είναι ανεπτυγμένη σε Ruby on Rails, ενώ για την εύρυθμη λειτουργία της χρειάζεται το Elasticsearch, μία λειτουργική εγκατάσταση του Moodle και πρόσβαση στην αντίστοιχη βάση δεδομένων του.

3.2.2 Ruby on Rails: Routes, Models και Controllers

3.2.2.i Τι είναι η Ruby και το Rails;

Η Ruby είναι μία αντικειμενοστρεφής γλώσσα προγραμματισμού ανοιχτού κώδικα, δυναμική, γενικής χρήσης που σχεδιάστηκε από τον Yukihiro Matz Matsumoto στην Ιαπωνία στα μέσα της δεκαετίας του '90, με πρώτη δημόσια κυκλοφορία τον Δεκέμβριο του 1995. Συνδυάζει χαρακτηριστικά από την Python, την Perl, τη Lisp, τη Dylan, την Pike και τη CLU. Γνώρισε τεράστια απήχηση και αποδοχή το 2006, κυρίως μέσω ενός πλαισίου ανάπτυξης διαδικτυακών εφαρμογών (web application framework), το Ruby on Rails.

Το Ruby on Rails είναι μία βιβλιοθήκη λογισμικού για τη Ruby και συνήθως καλείται απλώς «Rails». Το Rails είναι ένα πλαίσιο για την ανάπτυξη ιστοσελίδων, παρέχοντας στον προγραμματιστή ευκολίες για πιο γρήγορη ανάπτυξη, καλύτερη συνεργασία και ευκολότερη συντήρηση. Συνδυάζει τη Ruby με την HTML, τη CSS και τη JavaScript για να δημιουργήσει μία εφαρμογή διαδικτύου που τρέχει σε έναν web server. Αρκετά γνωστά site έχουν αναπτυχθεί χρησιμοποιώντας Ruby on Rails, ενδεικτικά Scribd, SlideShare, ThemeForest, Shopify, Basecamp, GitHub και Twitter.

3.2.2.ii Routes

Ο σκοπός του Rails Router είναι να αναγνωρίζει το URL και να προωθεί τα δεδομένα του αιτήματος στην κατάλληλη μέθοδο του κατάλληλου ελεγκτή. Όλες οι ενέργειες ορίζονται στο αρχείο `routes.rb`, ακολουθούν τη δομή

```
verb "address", to: "controller#method"[, as: "keyword"]
```

δηλαδή:

```
post "sign_in", to: "user#sign_in", as: "signin"
get "user", to: "user#info"
get "courses", to: "user#courses"
get "courses/logs", to: "course#get_logs"
delete "courses/categories", to: "course_category#delete"
```

Επίσης, ο router παρέχει βοηθητικές μεθόδους που μπορούν να υπολογίσουν αυτόματα δυναμικούς υπερσυνδέσμους για αντικείμενα ή σελίδες. Για παράδειγμα αντί για τον στατικό σύνδεσμο «<http://example.com/users/signin>», μπορούμε να χρησιμοποιήσουμε τη μέθοδο `signin_url`. Αυτό εξυπηρετεί γιατί σε περίπτωση αλλαγής του συνδέσμου σε «http://example.com/users/sign_in», αρκεί να αλλάξουμε τον σύνδεσμο στο αρχείο `routes.rb` αντί για οπουδήποτε αλλού αναφέρεται (π.χ. αρχεία `html`).

3.2.2.iii Models (Active Model)

Τα μοντέλα παρέχουν χρήσιμες βοηθητικές μεθόδους για τη διαχείρισή τους (`create`, `new`, `update`, `delete`, `destroy`). Στο όνομα των μοντέλων χρησιμοποιείται ο ενικός αριθμός (`User`), ενώ οι αντίστοιχοι πίνακες στη βάση δεδομένων είναι στον πληθυντικό (`users`). Αυτή είναι μία από τις συμβάσεις που χρησιμοποιούνται στο Rails και καλό είναι να εφαρμόζεται.

Το Rails παρέχει μία γεννήτρια για την εύκολη δημιουργία νέων μοντέλων, η οποία θα φτιάξει μία σειρά από χρήσιμα αρχεία, όπως το αρχείο για τη δημιουργία του πίνακα στη βάση δεδομένων. Παρέχουν επίσης μεθόδους για την εύκολη πρόσβαση σε συσχετισμένα μοντέλα. Έστω ότι έχουμε έναν χρήστη `user` τύπου `User`, και θέλουμε να πάρουμε τα μαθήματα του. Είναι αρκετό να κάνουμε `user.courses` το Rails θα κάνει το query (πίνακας `courses`, join τον πίνακα `users`) στη βάση δεδομένων και θα μας επιστρέψει τα μαθήματα του χρήστη τύπου `Course`.

Στην εφαρμογή μας έχουμε τα εξής μοντέλα: `User`, `CourseCategory`, `CustomActivity`, `CourseCategoryParameter`, & `MoodleDatabase`. Αναλυτικά:

- `User`: Περιέχει βασικά προσωπικά στοιχεία και το `token` του χρήστη για την αναγνώριση του από το Moodle.
- `CourseCategory`: Είναι το μοντέλο που κρατάει τις κατηγορίες του χρήστη για κάθε μάθημα.

- CustomActivity: Είναι το μοντέλο που «χαρακτηρίζει» μία δραστηριότητα με μία κατηγορία.
- CourseCategoryParameter: Είναι οι παράμετροι κάθε κατηγορίας μαθήματος και χρησιμοποιούνται στην Ανάλυση Ρίσκου.
- MoodleDatabase: Είναι το μοντέλο που κάνει τη σύνδεση με τη βάση δεδομένων του Moodle. Αυτή η σύνδεση απαιτείται, καθώς το Rails εξ' ορισμού έχει σύνδεση με μία και μόνο μία βάση δεδομένων. Χρησιμοποιείται στη διαδικασία rake import_logs_from_moodle για την αντιγραφή των logs από τη βάση δεδομένων στο Elasticsearch, καθώς και σε κάποια queries.

3.2.2.iv Controllers

Οι ελεγκτές (controllers) είναι αυτοί που περιέχουν τη λογική της εφαρμογής και είναι υπεύθυνοι για τη σωστή ροή των διαδικασιών. Κληρονομεί από τον ApplicationController και έχει μεθόδους όπως και κάθε άλλη κλάση, που μπορούν να είναι δημόσιες, ιδιωτικές ή προστατευόμενες (public, private ή protected).

Λόγω του ότι η εφαρμογή μας είναι με API, έχουμε τον ApiController ώστε να έχουμε καλύτερη οργάνωση και να μην επηρεάσουμε τα αρχεία της κύριας εφαρμογής. Κάθε ελεγκτής για το API, κληρονομεί από τον ApiController και έχει έναν συνονόματο ελεγκτή -αντίκρισμα- που κληρονομεί από τον ApplicationController.

Στην εφαρμογή μας υπάρχουν οι παρακάτω ελεγκτές: ApiController, ApplicationController, CourseCategoryController, CourseController, EventController, NoteController, NotificationController, UserController, EsController, MoodleController, RiskAnalysisController. Η εφαρμογή έχει 2 «κύριους» ελεγκτές και οι υπόλοιποι κληρονομούν από αυτούς. Έτσι, έχουμε τον παρακάτω πίνακα.

ApplicationController	Ελεγκτής «αντίκρισμα» από ApiController
CourseCategoryController	CourseCategoryController
CourseController	CourseController
EventController	EventController
NoteController	NoteController
NotificationController	NotificationController
UserController	UserController
RiskAnalysisController	RiskAnalysisController
EsController	
MoodleController	

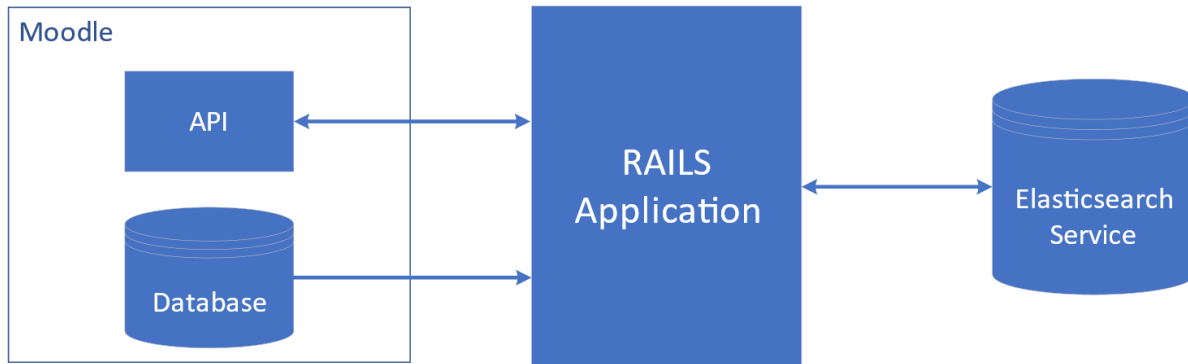
Πίνακας 3.2.1 - Οι ελεγκτές και τα αντίκριματά τους

Όλοι οι ελεγκτές περιέχουν μεθόδους σχετικές με την ονομασία τους, ενώ έχουν και ελεγκτή αντίκρισμα για το API. Οι ελεγκτές EsController και MoodleController δεν έχουν ελεγκτή αντίκρισμα γιατί δεν τους καλούμε απευθείας από κάποιο endpoint. Ο EsController περιέχει μεθόδους σχετικές με την επικοινωνία της εφαρμογής με το Elasticsearch. Ο MoodleController περιέχει βοηθητικές μεθόδους όπως

«enrolled_users», «contents» και «activities», οι οποίες επιστρέφουν δεδομένα με συγκεκριμένη δομή, όχι όπως από το Moodle API, και μπορούν να χρησιμοποιηθούν από τους υπόλοιπους ελεγκτές.

3.2.3 Ανάλυση Δομής

Η δομή της εφαρμογής παρουσιάζεται στο παρακάτω Διάγραμμα 3.2.1 - Σύντομο διάγραμμα δομής της Rails εφαρμογής.



Διάγραμμα 3.2.1 - Σύντομο διάγραμμα δομής της Rails εφαρμογής

Παρακάτω περιγράφονται πώς και γιατί επικοινωνεί η εφαρμογή μας με τα:

Moodle API: γίνεται η ταυτοποίηση του χρήστη και μέσω αυτού υπάρχει ελεγχόμενη πρόσβαση στις πληροφορίες των χρηστών/φοιτητών και των μαθημάτων του Moodle.

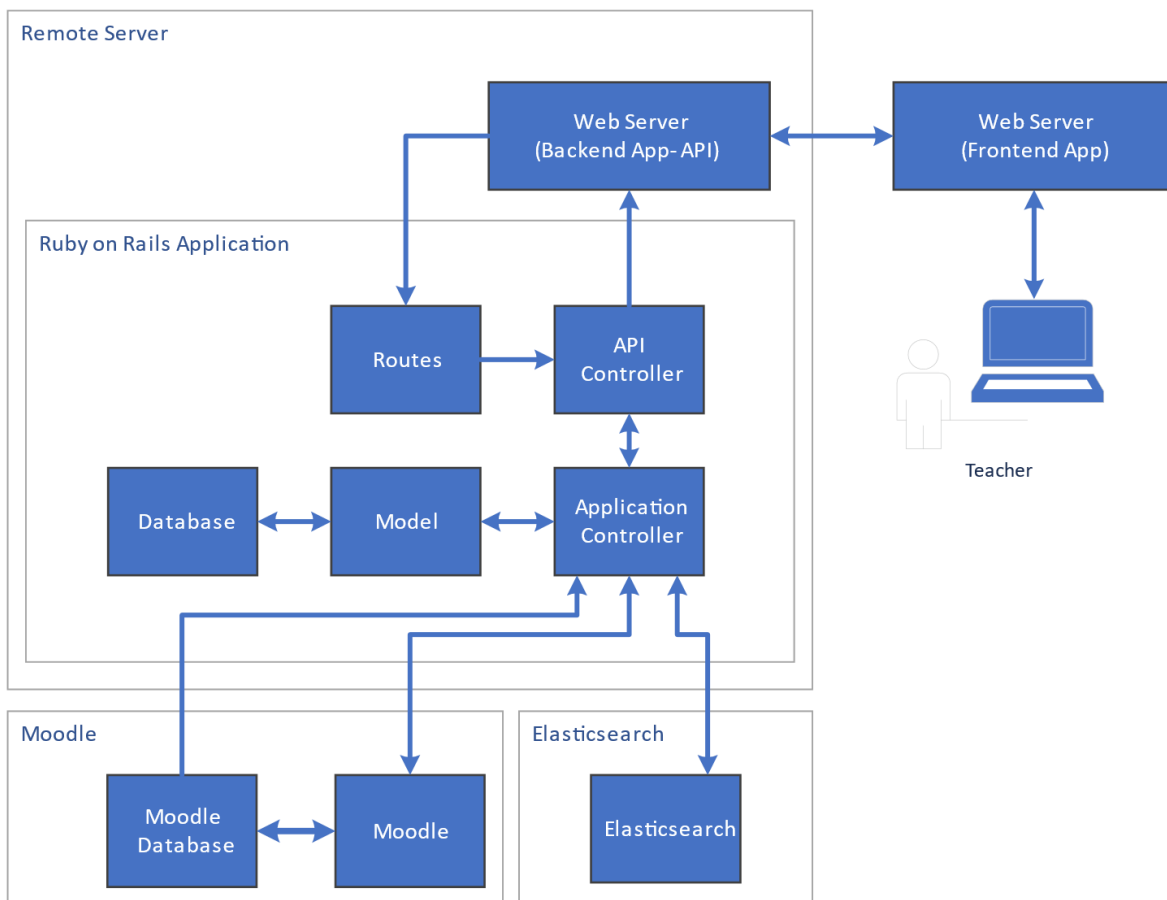
Βάση Δεδομένων του Moodle: χρησιμοποιείται κυρίως για πρόσβαση στα logs και τη αντιγραφή τους στο Elasticsearch. Χρησιμοποιείται επίσης για την εκτέλεση βελτιστοποιημένων queries, καθώς σε κάποιες περιπτώσεις η χρήση του Moodle API δεν παρέχει τις απαιτούμενες πληροφορίες ή είναι πολύ χρονοβόρο. Απαιτείται πρόσβαση σε συγκεκριμένους πίνακες και για λόγους ασφαλείας με δικαιώματα read-only.

Elasticsearch: χρησιμοποιείται για την αποθήκευση των logs του Moodle. Το Elasticsearch είναι σημαντικά ταχύτερο στην παροχή στατιστικών δεδομένων των logs σε σχέση με ένα αντίστοιχο query στη βάση δεδομένων του Moodle.

Στο Διάγραμμα 3.2.2 - Αναλυτικό διάγραμμα δομής της Rails εφαρμογής μπορούμε να δούμε καλύτερα τη δομή και την αλληλουχία που έχουν τα μέρη μεταξύ τους. Έτσι, όταν ο καθηγητής συνδέεται και χρησιμοποιεί την πλατφόρμα μας, έχουμε τα εξής στάδια:

1. Ο καθηγητής αλληλεπιδρά με το UI της FrontEnd εφαρμογής (εφεξής FE-A).
2. Ο web server (FE-A) κάνει το απαραίτητο αίτημα στον web server της BackEnd εφαρμογής (εφεξής BE-A).
3. Ο web server (BE-A) δέχεται το αίτημα και το προωθεί στον Router του Rails.
4. Ο Router θα προωθήσει το αίτημα στον κατάλληλο Ελεγκτή API, βάσει των εγγραφών στο αρχείο routes.rb.

5. Ο ελεγκτής API (API Controller) θα επικοινωνήσει με τον απαραίτητο ελεγκτή Εφαρμογής.
6. Ο ελεγκτής Εφαρμογής (Application Controller) θα επικοινωνήσει με τα απαραίτητα μοντέλα (Models) ή/και εξωτερικές υπηρεσίες (Moodle, Elasticsearch) και θα επιστρέψει τα δεδομένα στον ελεγκτή API.
7. Ο ελεγκτής API θα ελέγξει αν υπήρξε κάποιο σφάλμα και θα επιστρέψει κατάλληλη απάντηση σε μορφή JSON στον web server (BE-A).
8. Ο web server (BE-A) θα λάβει τα δεδομένα από τον ελεγκτή API και θα απαντήσει στο αίτημα του web server (FE-A).
9. Ο web server (FE-A) θα επεξεργαστεί κατάλληλα την απάντηση και θα την εμφανίσει στον καθηγητή.



Διάγραμμα 3.2.2 - Αναλυτικό διάγραμμα δομής της Rails εφαρμογής

3.2.4 Εγκατάσταση εφαρμογής

3.2.4.i Προετοιμασία περιβάλλοντος

Η εφαρμογή είναι σε γλώσσα Ruby και δύναται να λειτουργήσει σε Windows και UNIX περιβάλλοντα. Στα επόμενα στάδια θα δοθούν οδηγίες για την εγκατάσταση της εφαρμογής σε περιβάλλον UNIX (συγκεκριμένα Ubuntu 16.04), καθώς συνηθίζεται οι εφαρμογές σε αυτές τις τεχνολογίες να χρησιμοποιούνται σε UNIX και έχει μεγαλύτερη κοινότητα υποστήριξης.

Η εφαρμογή χρησιμοποιεί τις παρακάτω εκδόσεις:

- Ruby 2.3.6
- Rails 4.1.16
- Elasticsearch 5.3.0
- Moodle 3.1 – Moodle API και βάση δεδομένων
- MySQL 5.7

Στο Παράρτημα Α': (σελ. 63) παρέχονται scripts για την εγκατάσταση ολόκληρης της εφαρμογής σε Docker container.

3.2.4.ii Προετοιμασία υπηρεσιών για σύνδεση

➤ Moodle – Ενεργοποίηση Moodle Web Services

◊ *Οδηγίες για την ενεργοποίηση μπορούν να βρεθούν στη επίσημη σελίδα υποστήριξης MoodleDocs: [Enabling Web services](#).*

Για να ενεργοποιήσετε επιτυχώς το Moodle Web services, πρέπει να:

- Μεταβείτε στο Administration > Site administration > Advanced features.
- Επιλέξτε το πλαίσιο ελέγχου στο «**Enable web services**» και μετά πατήστε «**Save Changes**».
- Μεταβείτε Administration > Site administration > Plugins > Web services > Manage protocols
- Ενεργοποιήστε το πρωτόκολλο REST (Πατήστε στο εικονίδιο με το μάτι).
- Μεταβείτε στο Administration > Site administration > Users > Accounts > Add a new user
- Συμπληρώστε τα στοιχεία στη φόρμα.
- Στο «**Choose an authentication method**» επιλέξτε «Manual accounts».
- Τέλος, πατήστε «**Create user**».

- Μεταβείτε Administration > Site administration > Plugins > Web services > External services
- Πατήστε το «**Add**» στο τέλος της σελίδας.
- Συμπληρώστε τα πεδία όνομα (π.χ.: Web service for LA) και shortname (π.χ.: wsla).
- Επιλέξτε τα πλαίσια ελέγχου στα «**Enabled**» και «**Authorised users only**»
- Τέλος, πατήστε «**Add service**».
- Στο service που δημιουργήθηκε, πατήστε στο «**Functions**» και μετά «**Add functions**» στο τέλος της σελίδας.
- Επιλέξτε τα παρακάτω και μετά πατήστε «**Add functions**».

core_calendar_get_calendar_events	core_group_get_groupings
core_completion_get_activities_completion_status	core_group_get_groups
core_completion_get_course_completion_status	core_message_send_instant_messages
core_course_get_categories	core_notes_get_course_notes
core_course_get_contents	core_user_get_users
core_course_get_course_module	core_user_get_users_by_field
core_course_get_course_module_by_instance	core_webservice_get_site_info
core_course_get_courses	gradereport_user_get_grades_table
core_course_search_courses	mod_assign_get_assignments
core_enrol_get_enrolled_users	mod_assign_get_submission_status
core_enrol_get_users_courses	mod_scorm_get_scorm_attempt_count
core_fetch_notifications	mod_scorm_get_scorm_sco_tracks
core_files_get_files	mod_scorm_get_scorm_scoes
core_grades_get_grades	mod_scorm_get_scorm_user_data
core_group_get_course_groups	mod_scorm_get_scorms_by_courses
core_group_get_course_user_groups	tool_lp_search_users
core_group_get_group_members	

- Μεταβείτε στο Administration > Site administration > Plugins > Web services > Manage tokens
- Πατήστε το «**Add**» στο τέλος της σελίδας.
- Επιλέξτε τον χρήστη που έχετε δημιουργήσει για Learning Analytics.
- Επιλέξτε το service που δημιουργήσατε.
- Συμπληρώστε τα πεδία «**IP restriction**» και «**Valid Until**» όπως επιθυμείτε ή αφήστε τα κενά.
- Σημειώστε το token, καθώς θα το χρειαστείτε στο βήμα 3.2.4.iii Σύνδεση με το Moodle API.

➤ Δημιουργία χρήστη για τη βάση δεδομένων του Moodle

❖ Βεβαιωθείτε ότι έχετε αντίγραφο της βάσης δεδομένων. Μην ακολουθήσετε τις παρακάτω οδηγίες αν δεν έχετε ενημερώσει τον διαχειριστή του συστήματος.

Για την παροχή των γραφημάτων επισκεψιμότητας και της λειτουργίας Risk Analysis, είναι απαραίτητη η πρόσβαση στη βάση δεδομένων του Moodle. Για λόγους ασφαλείας, σας προτείνουμε να δημιουργηθεί νέος χρήστης με δικαιώματα read-only και πρόσβαση τουλάχιστον στους συγκεκριμένους πίνακες: mdl_course_modules_completion, mdl_course_modules, mdl_logstore_standard_log και mdl_modules.

Οι παρακάτω οδηγίες είναι ενδεικτικές για τη δημιουργία χρήστη σε βάση MySQL [5.7] και απαιτούν πρόσβαση στη βάση δεδομένων του Moodle. Απευθυνθείτε στον διαχειριστή του συστήματος Moodle.

```
CREATE USER 'newusertest'@'%' IDENTIFIED BY 'p2ssw0RdT3st';
GRANT SELECT ON moodle.mdl_course_modules_completion TO 'newusertest'@'%' ;
GRANT SELECT ON moodle.mdl_course_modules TO 'newusertest'@'%' ;
GRANT SELECT ON moodle.mdl_logstore_standard_log TO 'newusertest'@'%' ;
GRANT SELECT ON moodle.mdl_modules TO 'newusertest'@'%' ;
```

3.2.4.iii Σύνδεση της εφαρμογής με τις απαιτούμενες υπηρεσίες

- Σύνδεση με το Moodle API

Επεξεργαστείτε το αρχείο **“thesis-app/config/environments/development.rb”**

- ◊ Προϋποθέτει ότι έχει ολοκληρώσει με επιτυχία τα βήματα από το 3.2.4.ii Moodle – Ενεργοποίηση Moodle Web Services.

```
ENV['MOODLE_HOST_URL'] = '<moodle_ip_here>'
ENV['MOODLE_TOKEN'] = '<token_here>'
ENV['MOODLE_SERVICE_SHORT'] = '<service_shortname_here>'
```

Θα χρειαστεί να κάνετε επανεκκίνηση της εφαρμογής, αν τρέχει.

- Σύνδεση με τη βάση δεδομένων του Moodle

- ◊ Προϋποθέτει ότι έχετε δημιουργήσει χρήστη, όπως περιγράφονται παραπάνω στο 3.2.4.ii Δημιουργία χρήστη για τη βάση δεδομένων του Moodle.

Επεξεργαστείτε το αρχείο **“thesis-app/config/secrets.yml”**

```
development:
  ...
  database_moodle:
    :host: '<moodle_ip>'
    :name: 'moodle'
    :username: 'newusertest'
    :password: 'p2ssw0RdT3st'
    :port: 3306
```

- Σύνδεση με τη βάση δεδομένων της εφαρμογής RAILS

Επεξεργαστείτε το αρχείο **“thesis-app/config/secrets.yml”**

```
development:
  ...
  database:
    :host: '<database_ip>'
    :name: 'app_db'
    :username: 'railsuser'
    :password: 'railsuserp2ss'
    :port: 3306
```

3.2.5 Προσθήκη νέας λειτουργίας

Για να προστεθεί μία νέα λειτουργία θα πρέπει να κάνουμε τα ακόλουθα 3 βήματα:

1. Δημιουργία νέου route που θα καλεί τη σωστή μέθοδο στον σωστό ελεγκτή.

Στο αρχείο “config/routes.rb” προσθέτουμε:

```
get "/new_method", to: "example#new_method_api"
```

2. Δημιουργία νέας μεθόδου και νέου ελεγκτή - αν χρειάζεται - στον φάκελο app/controllers/api, που θα καλεί τον κύριο ελεγκτή (γραμμή 5).

Δημιουργούμε ή επεξεργαζόμαστε το αρχείο “app/controllers/api/v1/example_controller.rb” και προσθέτουμε:

```
01: class Api::V1::ExampleController < Api::V1::ApiController
02:   def new_method_api
03:     controller = ApplicationController::ExampleController.new
04:     controller.request = ActionDispatch::Request.new(request.env)
05:     response = controller.new_method
06:
07:     if response[:type] == :error
08:       success_response(type: :error, message: response[:message])
09:     else
10:       success_response(data: response[:data])
11:     end
12:   rescue => error
13:     Rails.logger.debug(error.message)
14:     error_response
15:   end
16: end
```

3. Δημιουργία νέας μεθόδου και ελεγκτή – αν χρειάζεται – στον φάκελο app/controllers.

```
01: class ExampleController < ApplicationController
02:   def new_method
03:     {data: "Data from your new method."}
04:   end
05: end
```

Πλέον το νέο endpoint `http://localhost:3000/new_endpoint` μπορεί να χρησιμοποιηθεί. Μπορείτε να το δοκιμάσετε μέσω του εργαλείου Postman.

3.2.6 Πίνακας υλοποιημένων endpoints

Όνομασία	Μέθοδος	Διαδρομή χωρίς το {{host}}/api/1	Περιγραφή ή/και Σημειώσεις
	Κεφαλίδες:		
	Parameters ή Body:		
Get user's enrolled courses	GET	/courses	Επιστρέφει τα μαθήματα που έχει πρόσβαση ο καθηγητής.
	Κεφαλίδες: Authorization: Token {{token}}		
	Parameters: user_id=1		
Get Activity Grid	GET	/courses/activities	Επιστρέφει το πώς έχει χαρακτηρίσει ο καθηγητής την κάθε δραστηριότητα για το επιλεγμένο μάθημα.
	Κεφαλίδες: Authorization: Token {{token}}		
	Parameters: course_id=11		
Create/Update Activity Grid	POST	/courses/activities	Αποθηκεύει το πώς έχει χαρακτηρίσει ο καθηγητής την κάθε δραστηριότητα για το επιλεγμένο μάθημα.
	Κεφαλίδες: Authorization: Token {{token}} Content-Type= "application/json"		
	Body: { "course_id": 11, "activities": [{ "id": 48, "1": true, "2": false }] }		
Get Course Categories	GET	/courses/categories	Επιστρέφει τις διαθέσιμες κατηγορίες για χαρακτηρισμό των δραστηριοτήτων για το επιλεγμένο μάθημα.
	Κεφαλίδες: Authorization: Token {{token}}		
	Parameters: course_id=13		
Create Course Category	POST	/courses/categories	
	Κεφαλίδες: Authorization: Token {{token}}		

	Body: name="Category name 1" course_id=13	Δημιουργεί νέα κατηγορία για το επιλεγμένο μάθημα.
Delete Course Category	DELETE /courses/categories	Διαγράφει την επιλεγμένη κατηγορία.
	Κεφαλίδες: Authorization: Token {{token}}	
	Body: id=1	
Get Course Parameters	GET /courses/categories_parameters	Επιστρέφει τους συντελεστές για όλες τις κατηγορίες και τις σταθερές του επιλεγμένου μαθήματος.
	Κεφαλίδες: Authorization: Token {{token}}	
	Parameters: course_id=13	
Create/Update Course Parameters	POST /courses/categories_parameters	Αποθηκεύει τους συντελεστές για όλες τις κατηγορίες και τις σταθερές για το επιλεγμένο μάθημα.
	Κεφαλίδες: Authorization: Token {{token}} Content-Type= "application/json"	
	Body: { "course_id": 13, "parameters": [{ "category_id": 31, "1": 2, "2": -44.25 }, { "category_id": 32, "1": 2414.35, "2": -25 }], "constants": { "1": 5.002, "2": -92 } }	
Get Contents	GET /courses/contents	Επιστρέφει τα περιεχόμενα ενός μαθήματος.
	Κεφαλίδες: Authorization: Token {{token}}	
	Parameters: course_id=13	
Get Course Categories Graph	GET /courses/custom_categories_graph	Επιστρέφει στατιστικά για το πώς έχει χαρακτηρίσει ο καθηγητής τις δραστηριότητες του επιλεγμένου μαθήματος.
	Κεφαλίδες: Authorization: Token {{token}}	
	Parameters: course_id=13	
Get Enrolled Users	GET /courses/enrolled_users	Επιστρέφει τους εγγεγραμμένους χρήστες (μαθητές και καθηγητές) του μαθήματος.
	Κεφαλίδες: Authorization: Token {{token}}	
	Parameters: course_id=13	
Get if Initialized	GET /courses/initialized_course	Επιστρέφει αν ο χρήστης έχει
	Κεφαλίδες: Authorization: Token {{token}}	

	Parameters: course_id=13	δημιουργήσει κατηγορίες για αυτό το μάθημα. Αν όχι, δεν μπορεί να μεταβεί το widget του Risk Analysis.
Get Course Logs	GET /courses/logs	Επιστρέφει στοιχεία επισκεψιμότητας για την επιλεγμένη χρονική περίοδο για το επιλεγμένο μάθημα και τους επιλεγμένους μαθητές.
	Κεφαλίδες: Authorization: Token {{token}}	
	Parameters: course_id=13 from_date=01-01-2013 to_date=2018 query=viewed view=day module=course module_ids=[-1] student_ids=[1234, 1432]	
Get Course Module Logs	GET /courses/logs	Επιστρέφει στοιχεία επισκεψιμότητας στις δραστηριότητες του επιλεγμένου μαθήματος για την επιλεγμένη χρονική περίοδο. Αν το module_ids είναι [-1], θα επιστραφούν για όλες τις δραστηριότητες.
	Κεφαλίδες: Authorization: Token {{token}}	
	Parameters: course_id=13 from_date=01-01-2013 to_date=2018 query=viewed view=day module=course_module module_ids=[12,15] student_ids=[1234, 1432]	
Get Modules	GET /courses/modules	Επιστρέφει όλες τις δραστηριότητες του μαθήματος.
	Κεφαλίδες: Authorization: Token {{token}}	
	Parameters: course_id=13	
Get Risk Analysis	GET /courses/risk_analysis	Επιστρέφει όλους τους μαθητές του επιλεγμένου μαθήματος, και το αν έχουν ολοκληρώσει επιτυχώς κάθε δραστηριότητα.
	Κεφαλίδες: Authorization: Token {{token}}	
	Parameters: course_id=13 from_date=01-01-2014 to_date=01-01-2018	
Get info for user dashboard	GET /dashboard	Επιστρέφει στατιστικά και στοιχεία για την πρώτη οθόνη της εφαρμογής.
	Κεφαλίδες: Authorization: Token {{token}}	
	Parameters: from_date=02-07-2014 to_date=27-10-2018 view=month	
Get Events	GET /events	Επιστρέφει τα προγραμματισμένα συμβάντα που έχει το Moodle.
	Κεφαλίδες: Authorization: Token {{token}}	
	Parameters: -	
Get Course Notes	GET /courses/notes	Επιστρέφει τις σημειώσεις του χρήστη στο συγκεκριμένο μάθημα. Δεν
	Κεφαλίδες: Authorization: Token {{token}}	
	Parameters: course_id=13	

			χρησιμοποιείται στην εφαρμογή μας ακόμα.
Notifications	GET	/notifications	Επιστρέφει τις ειδοποιήσεις του χρήστη. Δεν χρησιμοποιείται στην εφαρμογή μας ακόμα.
	Κεφαλίδες: Authorization: Token {{token}}		
	Parameters: -		
Send Message	POST	/send_message	Στέλνει μήνυμα στους επιλεγμένους χρήστες. Χρησιμοποιείται στο Risk Analysis, όπου ο καθηγητής μπορεί να στείλει κάποια χρήσιμη σημείωση στους μαθητές.
	Κεφαλίδες: Authorization: Token {{token}}		
	Body: message="Message to send to students" student_ids: [1, 2, 3, 4]		
Sign in	POST	/sign_in	Στέλνει τα στοιχεία σύνδεσης στο Moodle, ταυτοποιεί τον χρήστη και επιστρέφει ένα token για την περαιτέρω αναγνώριση του χρήστη.
	Κεφαλίδες: -		
	Body: username: "loginusername" password: "log1np@\$Sw0rD"		
Test Connection	GET	/test	Επιστρέφει την κατάσταση της σύνδεσης και το αν ο χρήστης είναι ταυτοποιημένος ή όχι.
	Κεφαλίδες: (Optional) Authorization: Token {{token}}		
	Parameters: -		
User Info	GET	/user	Επιστρέφονται πληροφορίες για τον συνδεδεμένο χρήστη ή για τον χρήστη με το επιλεγμένο id.
	Κεφαλίδες: Authorization: Token {{token}}		
	Parameters: user_id=3		

3.3 Frontend

3.3.1 Εισαγωγή

Σύνοψη κεφαλαίου:

Σε αυτό το κεφάλαιο θα αναφερθούμε στο user interface, τη σημαντικότητά του, τις λειτουργίες τις οποίες προσφέρουμε, τη δομή της εφαρμογής και τέλος θα παρουσιάσουμε μερικά UML διαγράμματα.

Τι είναι το user interface (UI);

Το user interface είναι ο συνδετικός κρίκος μεταξύ του χρήστη και της πλατφόρμας. Περιέχει τον τρόπο με τον οποίο η πλατφόρμα αλληλοεπιδρά με τον χρήστη, το συνολικό design και τον τρόπο με τον οποίο παρουσιάζουμε τις πληροφορίες.

Τεχνολογίες:

Στην εφαρμογή μας, για το frontend, χρησιμοποιούμε JavaScript και συγκεκριμένα το framework ExtJS 6.0.2 GPLv3 License.

Λειτουργίες:

Συνοπτικά οι λειτουργίες που προσφέρουμε είναι οι εξής:

- I. Login με τον λογαριασμό του Moodle.
 - II. Γενική επισκόπηση όλων των μαθημάτων του.
 - III. Αναλυτική επισκόπηση ανά μάθημα.
 - IV. Δυνατότητα υπολογισμού του risk analysis για τους μαθητές.
- Όπως θα δούμε στη συνέχεια, στην εφαρμογή μας παρουσιάζονται διάφορα γραφήματα (bar-chart, pie-chart), για την καλύτερη απεικόνιση των πληροφοριών.

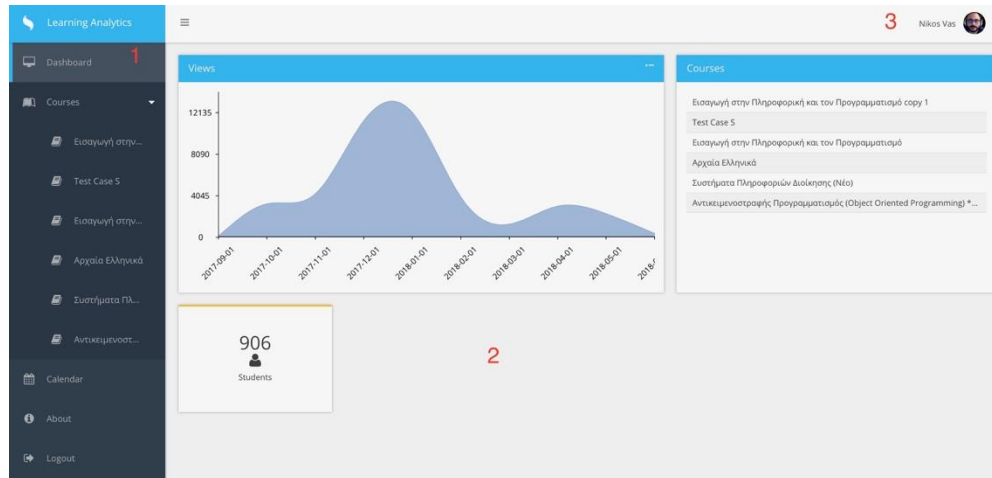
3.3.2 Λειτουργίες

Για να μπορέσει ο καθηγητής να συνδεθεί στην πλατφόρμα μας θα πρέπει να χρησιμοποιήσει τα στοιχεία σύνδεσης (email - password) από τον λογαριασμό του Moodle του. Η ταυτοποίηση γίνεται μέσω του Moodle API, όπως εξηγήθηκε στο 3.2.3 Ανάλυση Δομής. Αν τυχόν ο καθηγητής δε γνωρίζει τα στοιχεία σύνδεσης του, θα πρέπει να επικοινωνήσει με τον διαχειριστή του συστήματος του Moodle.

Μετά την επιτυχή ταυτοποίηση του καθηγητή, έχει πρόσβαση στις πληροφορίες και στα γραφήματα που παρέχονται στην εφαρμογή. Αξίζει να επισημάνουμε, ότι ο καθηγητής θα πρέπει να έχει ενεργοποιήσει μερικές ρυθμίσεις στο Moodle, έτσι ώστε να έχουμε

δικαίωμα να πάρουμε αυτές τις πληροφορίες από το Moodle. Η διαδικασία αυτή, καλύφθηκε στο 3.2.4.ii Προετοιμασία υπηρεσιών για σύνδεση.

3.3.2.i Πρώτη εντύπωση



Εικόνα 3.3.1 - Κεντρική οθόνη

Στην Εικόνα 3.3.1 - Κεντρική οθόνη φαίνεται η πρώτη οθόνη που βλέπει ο καθηγητής, μετά τη σύνδεσή του. Στα αριστερά υπάρχει το navigation menu (1), στη μέση είναι το κύριο panel (2) και επάνω δεξιά (3) εμφανίζεται το username του χρήστη.

Από το navigation menu μπορούμε να περιηγηθούμε στις εξής ενότητες και επιλογές:

1. Dashboard: Εμφανίζονται στατιστικά στοιχεία για όλα τα μαθήματα του χρήστη.
2. Courses: Εμφανίζονται στατιστικά στοιχεία για το επιλεγμένο μάθημα.
3. Calendar: Εμφανίζονται τα events του καθηγητή.
4. About: Πληροφορίες για τη πλατφόρμα.
5. Logout: Αποσύνδεση του χρήστη.

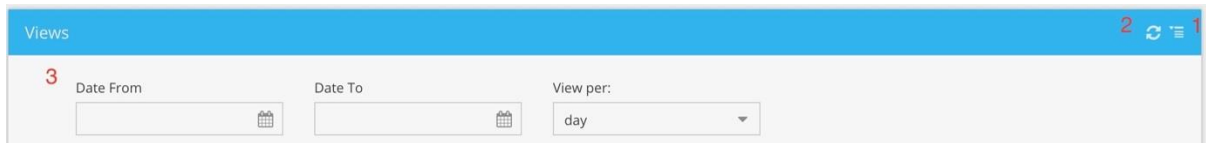
Τις κυριότερες από τις παραπάνω επιλογές θα τις αναλύσουμε στη συνέχεια.

Στο κύριο panel εμφανίζονται πληροφορίες και γραφήματα, ανάλογα με το τι έχουμε επιλέξει στο navigation menu.

3.3.2.ii Dashboard

Στο Dashboard ο καθηγητής μπορεί να έχει μια γενική επισκόπηση όλων των μαθημάτων του. Όπως φαίνεται και στην Εικόνα 3.3.1 - Κεντρική οθόνη, υπάρχουν τρία widgets. Το Courses, που εμφανίζονται όλα τα μαθήματα του καθηγητή. Το Students, όπου εμφανίζεται ο συνολικός αριθμός των φοιτητών που παρακολουθούν τα μαθήματα του καθηγητή. Το Views, όπου εμφανίζεται σε γράφημα, η επισκεψιμότητα που είχαν τα μαθήματα του.

Στο widget Views υπάρχει η δυνατότητα, τα αποτελέσματα να περιοριστούν στην επιθυμητή χρονική περίοδο που θέλει ο καθηγητής να εστιάσει.

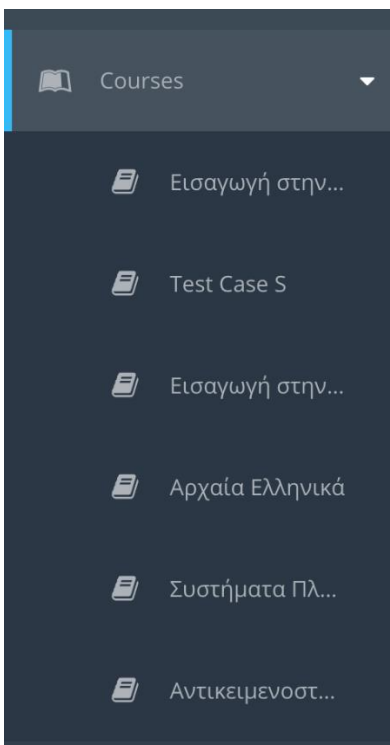


Εικόνα 3.3.2 - Dashboard Views Widget – Toolbar

Όπως φαίνεται στην Εικόνα 3.3.2, αν πατήσουμε το εικονίδιο (1), κάνει extend το widget Views και εμφανίζονται οι επιλογές που μπορούμε να κάνουμε search. Οι επιλογές αυτές (Εικόνα 3.3.2 νούμερο 3) είναι με field name, “Date From”, “Date To” και “View per”. Αυτό σημαίνει ότι μπορούμε να επιλέξουμε τη χρονική περίοδο από (Date From) έως (Date To), όπως επίσης και να επιλέξουμε τη χρονική διάρκεια (View per) με την οποία θα εμφανίζονται στο γράφημα. Η χρονική διάρκεια μπορεί να είναι ημερησίως, εβδομαδιαίως κλπ. Στη συνέχεια και αφότου ο καθηγητής έχει επιλέξει τη χρονική περίοδο, θα πρέπει να πατήσει το εικονίδιο (2) για να ενημερωθεί το γράφημα.

3.3.2.iii Courses

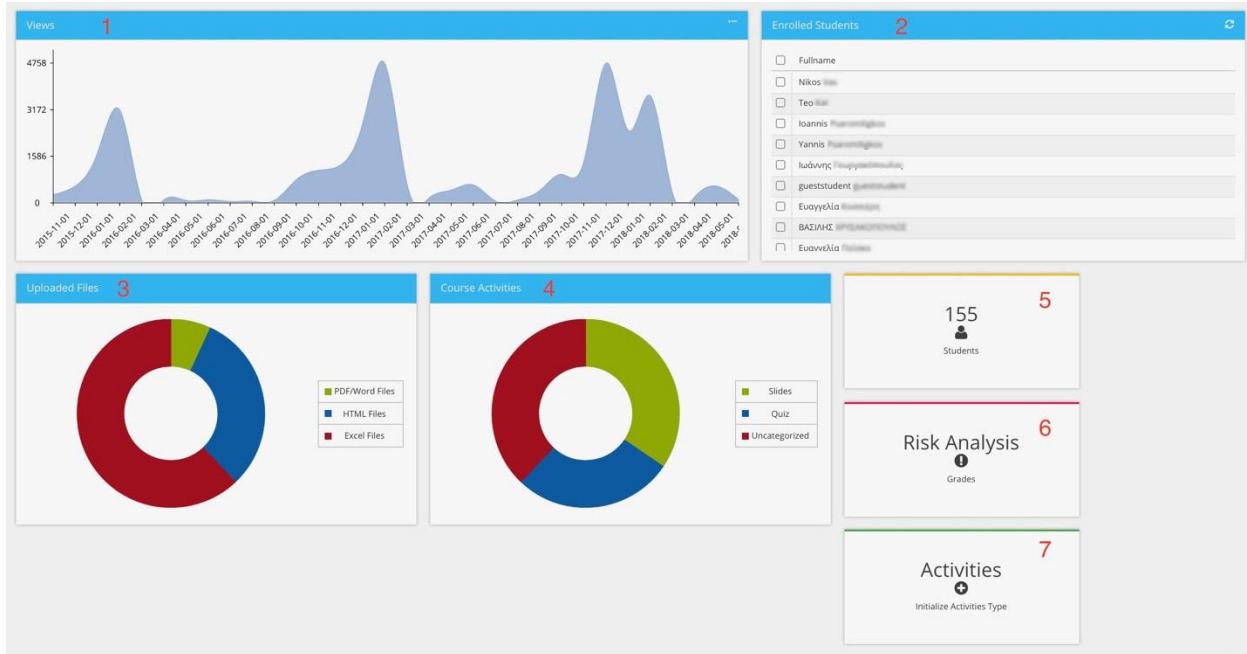
Στο navigation menu υπάρχει η επιλογή του Courses. Όταν πατηθεί το Courses κάνει extend ένα υπομενού όπως φαίνεται και στην Εικόνα 3.3.3 - Courses submenu.



Εικόνα 3.3.3 - Courses submenu

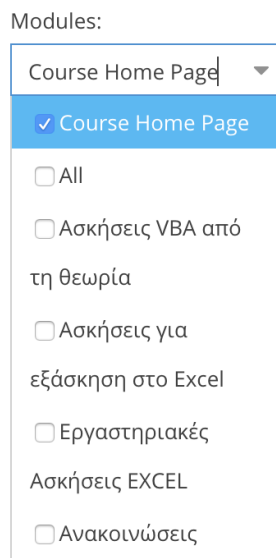
Το υπομενού εμφανίζει όλα τα μαθήματα που έχει ο καθηγητής. Επιλέγοντας ένα μάθημα, γίνεται ενημέρωση του κύριου πάνελ και εμφανίζονται πληροφορίες για το επιλεγμένο

μάθημα.



Εικόνα 3.3.4 - Κεντρικό πάνελ μαθήματος

Η Εικόνα 3.3.4 - Κεντρικό πάνελ μαθήματος δείχνει το κύριο πάνελ του μαθήματος.



Εικόνα 3.3.5 - Widget Views - Modules Dropdown

Το widget Views εμφανίζει σε γράφημα, την επισκεψιμότητα του μαθήματος. Όπως και στο widget Views του Dashboard, ο καθηγητής έχει τη δυνατότητα να κάνει search με όποια χρονική περίοδο επιθυμεί. Επιπλέον όμως έχει και τη δυνατότητα να επιλέξει όποιο module επιθυμεί (βλέπε Εικόνα 3.3.5 - Widget Views - Modules Dropdown). Τέλος, από το widget Enrolled Students (2), μπορεί να επιλέξει όποιο μαθητή επιθυμεί και από το εικονίδιο refresh επάνω δεξιά στο toolbar, μπορεί να ενημερώσει το γράφημα. Ο

καθηγητής έχει πλέον τη δυνατότητα να χρησιμοποιήσει ένα advanced search εργαλείο, όπου μπορεί να συνδυάσει τη χρονική περίοδο, τα modules και τους μαθητές.

Το widget Enrolled Students (2) εμφανίζει τους μαθητές που είναι εγγεγραμμένοι, το widget Students (5) εμφανίζει τον συνολικό αριθμό των μαθητών και το widget Uploaded Files (3) εμφανίζει σε γράφημα πίτας τον τύπο των αρχείων που έχει ανεβάσει ο καθηγητής.

3.3.2.iv Risk Analysis

Μια από τις σημαντικότερες λειτουργίες της πλατφόρμας είναι ο υπολογισμός του risk analysis. Το risk analysis δίνει στον καθηγητή μια εκτίμηση για το ποιοι μαθητές είναι σε κίνδυνο να αποτύχουν στο μάθημα.

Από την Εικόνα 3.3.4 - Κεντρικό πάνελ μαθήματος δεν αναφερθήκαμε καθόλου στα widgets Course Activities (4), Risk Analysis (6), Activities (7). Τα widgets αυτά είναι αλληλένδετα μεταξύ τους και παίζουν σημαντικό ρόλο στον υπολογισμό του risk analysis.

Activity	Slides	Quiz	Uncategorized
01 Άσκηση "Μαρκάρισε τα κελιά..."	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
02 Βρες ποιοι είναι Pass ή NotPass	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
03 Λύσε τα Τριώνυμα (1η έκδοση)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
04 Λύσε τα τριώνυμα (2η έκδοση, χρήση υποπρογραμμάτων)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
05 Το παιχνίδι Joker	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4η Ενότητα - Βιντεοσκοπημένη Διάλεξη - Μέρος 2/2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5η Ενότητα - Βιντεοσκοπημένη Διάλεξη - Μέρος 1/2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5η Ενότητα - Βιντεοσκοπημένη Διάλεξη - Μέρος 2/2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6η Ενότητα - Βιντεοσκοπημένη Διάλεξη - Μέρος 1/4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6η Ενότητα - Βιντεοσκοπημένη Διάλεξη - Μέρος 2/4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Εικόνα 3.3.6 - Initialize Activities

Για τον υπολογισμό του risk analysis, ο καθηγητής θα πρέπει, τουλάχιστον την πρώτη φορά, να πατήσει πάνω στο widget Activities (7) και να χαρακτηρίσει τη κατηγορία κάθε δραστηριότητας (activity). Όπως φαίνεται και στην εικόνα υπάρχει ένας πίνακας όπου μπορεί να κλικάρει τη κατηγορία κάθε δραστηριότητας. Οι προεπιλεγμένες κατηγορίες είναι οι «Slides», «Quiz» και «Uncategorized». Αν δεν κλικάρει σε καμία κατηγορία

αυτόματα θα χαρακτηριστούν με τη κατηγορία «Uncategorized». Οι δραστηριότητες της κατηγορίας «Uncategorized», δεν λαμβάνονται υπόψιν στον υπολογισμό του risk analysis. Επίσης, ο καθηγητής έχει τη δυνατότητα να προσθέσει ή να αφαιρέσει κατηγορίες, αλλά δεν μπορεί να αφαιρέσει την κατηγορία «Uncategorized».

Μετά την επιτυχή αποθήκευση του Initialize Activities, ενημερώνεται αυτόματα το widget Course Activities (4) όπου εμφανίζονται σε γράφημα πίτας, οι κατηγορίες των δραστηριοτήτων του μαθήματος, όπως τις έχει χαρακτηρίσει ο καθηγητής.

Risk Analysis Overview

Step 1 Results

Please select the dates to calculate the risk analysis.

Date From: 2018-01-01 Date To: 2018-07-31

If you want, you can change the default parameter's value.

A₁₁: 2.45648 A₁₂: 3.43455 C₁: 5.3453

A₂₁: -2.756 A₂₂: 3.242 C₂: 1.242

$Y_1 = A_{11} * X_1 + A_{12} * X_2 + C_1$
 $Y_2 = A_{21} * X_1 + A_{22} * X_2 + C_2$

Where:
X₁ = Slides
X₂ = Quiz

Previous Next

Εικόνα 3.3.7 - Risk Analysis Overview - Step One

Στη συνέχεια ο καθηγητής πρέπει να πατήσει στο widget Risk Analysis (6), όπου θα του εμφανίσει ένα παράθυρο σαν την Εικόνα 3.3.7 - Risk Analysis Overview - Step One, χωρίς όμως τιμές στις παραμέτρους. Στο παράθυρο αυτό, ορίζουμε τους συντελεστές των κατηγοριών.

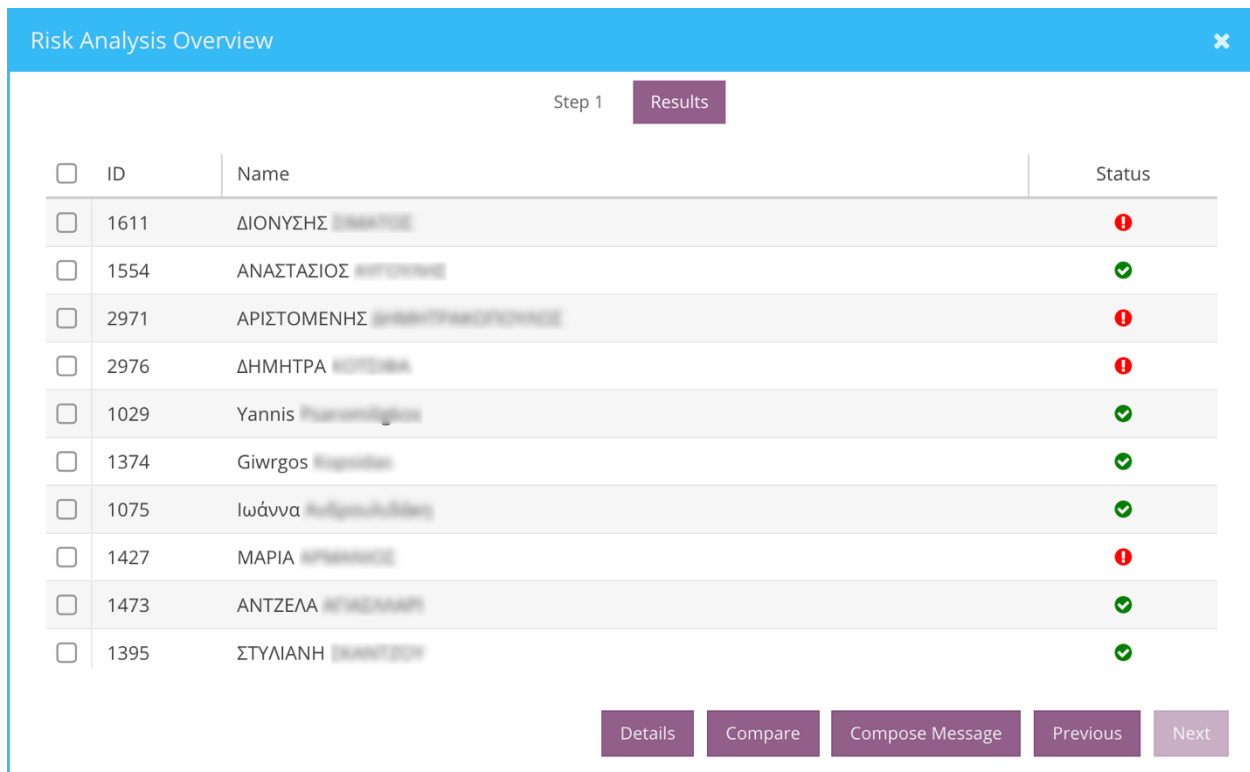
Οι αλγόριθμοι για τον υπολογισμό του risk analysis είναι οι:

1. $Y_1 = A_{11} * X_1 + A_{12} * X_2 + \dots + A_{1N} * X_N + C_1$
2. $Y_2 = A_{21} * X_1 + A_{22} * X_2 + \dots + A_{2N} * X_N + C_2$

Όπου A είναι οι συντελεστές, X είναι οι κατηγορίες που έχει ορίσει ο καθηγητής και C η σταθερά.

Ένας φοιτητής δεν βρίσκεται σε κίνδυνο όταν το $Y_1 > Y_2$.

Στην Εικόνα 3.3.7 - Risk Analysis Overview - Step One υπάρχουν μόνο 2 κατηγορίες. Παρ' όλα αυτά, το έχουμε υλοποιήσει δυναμικά και δημιουργεί τα πεδία συντελεστών, ανάλογα με τον αριθμό των κατηγοριών που έχει ορίσει ο καθηγητής από το widget Activities (7).



<input type="checkbox"/>	ID	Name	Status
<input type="checkbox"/>	1611	ΔΙΟΝΥΣΗΣ	⚠
<input type="checkbox"/>	1554	ΑΝΑΣΤΑΣΙΟΣ	✅
<input type="checkbox"/>	2971	ΑΡΙΣΤΟΜΕΝΗΣ	⚠
<input type="checkbox"/>	2976	ΔΗΜΗΤΡΑ	⚠
<input type="checkbox"/>	1029	Yannis	✅
<input type="checkbox"/>	1374	Giwrgos	✅
<input type="checkbox"/>	1075	Ιωάννα	✅
<input type="checkbox"/>	1427	ΜΑΡΙΑ	⚠
<input type="checkbox"/>	1473	ΑΝΤΖΕΛΑ	✅
<input type="checkbox"/>	1395	ΣΤΥΛΙΑΝΗ	✅

Εικόνα 3.3.8 - Risk Analysis Overview - Results

Πατώντας το κουμπί Next μας εμφανίζει τα αποτελέσματα όπως φαίνεται και στην Εικόνα 3.3.8 - Risk Analysis Overview - Results. Όσοι φοιτητές δεν βρίσκονται σε κίνδυνο, το status τους είναι ένα πράσινο εικονίδιο αλλιώς είναι ένα κόκκινο.

Στο συγκεκριμένο παράθυρο ο καθηγητής έχει μερικές επιπλέον δυνατότητες. Πατώντας το κουμπί Details εμφανίζονται όλες οι δραστηριότητες με το δικό τους status, δηλαδή αν ο φοιτητής πέρασε ή όχι τη δραστηριότητα. Πατώντας το κουμπί Compare εμφανίζεται ένα παράθυρο με ένα γράφημα, όπου δείχνει τον συνολικό αριθμό των επιτυχόντων και αποτυχόντων ανά δραστηριότητα. Τέλος, ο καθηγητής μπορεί να επιλέξει από τον πίνακα τους φοιτητές που επιθυμεί και πατώντας το κουμπί Compose Message να στείλει μαζικά ένα προσωπικό μήνυμα στο Moodle.

3.3.3 Τεχνικές Λεπτομέρειες

Στο προηγούμενο κεφάλαιο αναφερθήκαμε στις βασικές λειτουργίες της πλατφόρμας, από την πλευρά του χρήστη / καθηγητή. Σε αυτό το κεφάλαιο θα αναφερθούμε στις τεχνολογίες που χρησιμοποιήσαμε και στον τρόπο που υλοποιήσαμε τις παραπάνω λειτουργίες.

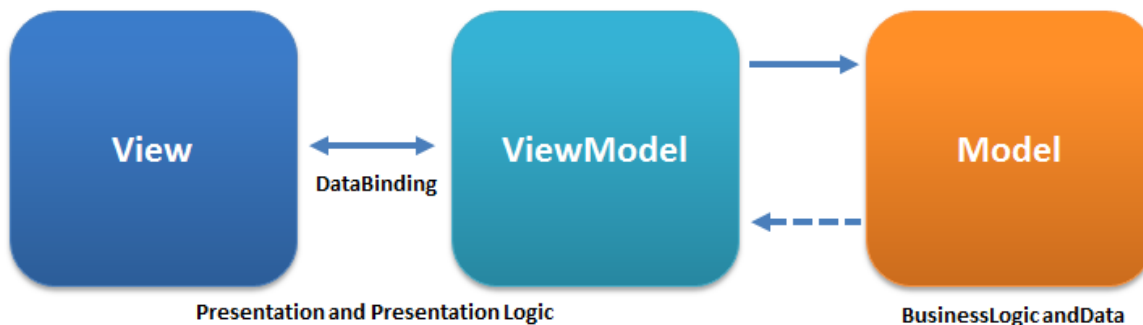
3.3.3.i Εργαλεία και Frameworks

Ένα από τα σημαντικότερα εργαλεία που χρησιμοποιήσαμε για την υλοποίηση του project είναι το Visual Studio Code. Είναι ένα open source code editor, το οποίο υποστηρίζει τη γλώσσα προγραμματισμού JavaScript και το framework ExtJS, που χρησιμοποιήσαμε για την υλοποίηση του frontend. Επίσης, σημαντικά εργαλεία είναι το Chrome DevTools, το οποίο μας δίνει τη δυνατότητα debugging JavaScript code και να ελέγξουμε τα network request / response, και το Lucidchart, με το οποίο μπορούμε να σχεδιάσουμε UML διαγράμματα όπως θα δούμε στο Κεφάλαιο 3.3.4 .

Όπως αναφέραμε και προηγουμένως, χρησιμοποιούμε τη γλώσσα προγραμματισμού JavaScript και το framework ExtJS GPL v3, το οποίο είναι της εταιρείας Sencha. Το ExtJS είναι ένα enterprise framework, όμως η εταιρεία Sencha δίνει την επιλογή για μια δωρεάν open source έκδοση του ExtJS με license GPL v3. Αυτό σημαίνει ότι όσοι επιλέξουν τη συγκεκριμένη έκδοση θα πρέπει να ακολουθήσουν τους όρους του GPL v3. Για να τρέξουμε το ExtJS θα πρέπει πρώτα να το κατεβάσουμε¹ και στη συνέχεια να ακολουθήσουμε τον οδηγό για να το εγκαταστήσουμε².

3.3.3.ii Αρχιτεκτονική και Δομή

Η αρχιτεκτονική που επιλέξαμε για το project μας, βασίζεται στο πρότυπο του MVVM (Model - View - ViewModel).



Διάγραμμα 3.3.1 - MVVM

Το MVVM είναι ένα software architectural pattern. Δηλαδή, μας καθοδηγεί στο πώς να οργανώσουμε και να δομήσουμε σωστά τον κώδικα που γράφουμε. Όπως βλέπουμε και από το Διάγραμμα 3.3.1 - MVVM, το συγκεκριμένο πρότυπο χωρίζεται σε τρεις κατηγορίες.

Model: Περιέχει τα δεδομένα και τη λογική του προγράμματος.

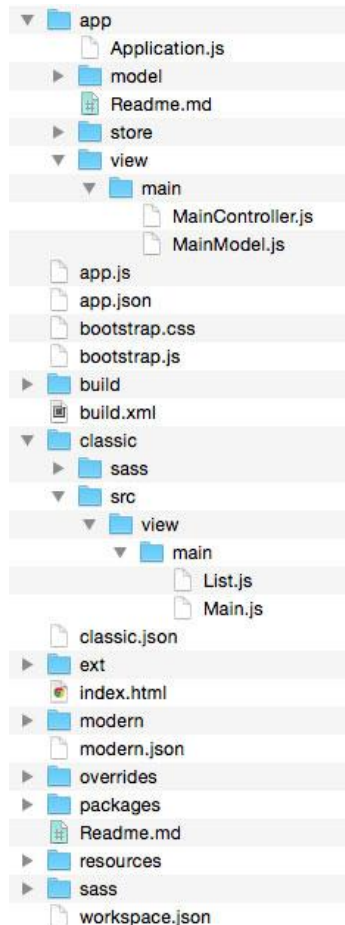
¹ <https://www.sencha.com/legal/gpl/>

² https://docs.sencha.com/extjs/6.0.2/guides/getting_started/getting_started.html

ViewModel: Είναι ο συνδεδειγμένος κρίκος μεταξύ του Model και του View. Δομεί τα δεδομένα που πήρε από το Model πριν τα παραλάβει το View.

View: Εμφανίζει τα δεδομένα στον χρήστη, δηλαδή είναι το user interface.

Το συγκεκριμένο πρότυπο είναι δημοφιλές και χρησιμοποιείται ευρέως με μικρές τροποποιήσεις, όπως και στη δική μας περίπτωση. Κατά τη διάρκεια της δημιουργίας μιας καινούργιας εφαρμογής, το ExtJS παίρνει τη πρωτοβουλία να μας γλιτώσει από πολύ χρόνο δομώντας το project μας.



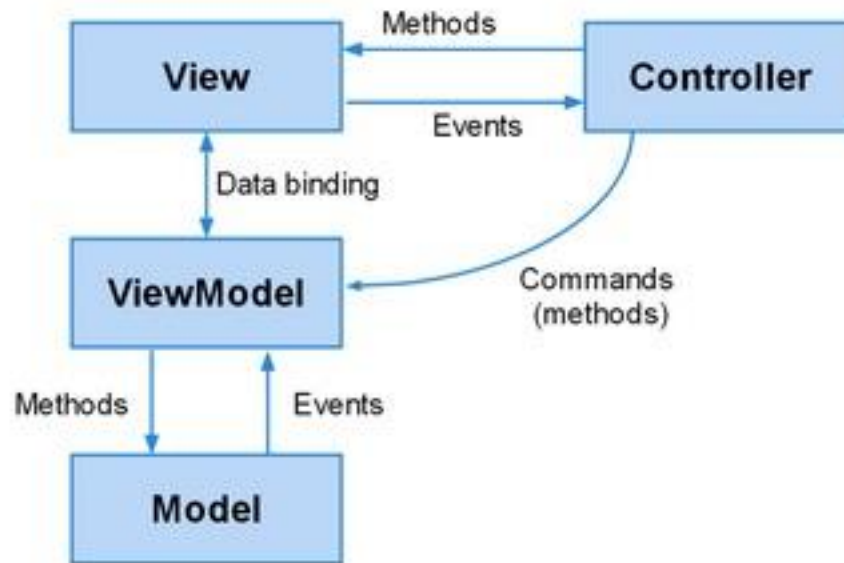
Εικόνα 3.3.9 - ExtJS File Structure

Η Εικόνα 3.3.9 - ExtJS File Structure δείχνει πώς δόμησε το project μας. Δημιούργησε φακέλους και υποφακέλους με βάση το πρότυπο του MVVM. Επίσης δημιούργησε και μερικά αρχεία τα οποία είναι σημαντικά για να μπορέσει να τρέξει το ExtJS. Οι φάκελοι στους οποίους θα επικεντρωθούμε είναι οι app, classic και modern.

Στον φάκελο app αποθηκεύουμε όλα τα Store, Model και μερικά από τα ViewModel, ViewController και View (όπου τα Model τοποθετούνται στον φάκελο /model, τα Store στον φάκελο /store και τα υπόλοιπα στον φάκελο /view). Στο Store ορίζουμε τα calls που κάνουμε στο server. Στο Model ορίζουμε τα πεδία και τον τύπο των πεδίων που

λαμβάνουμε από το server call. Αυτό σημαίνει ότι σε κάθε αρχείο store θα πρέπει να αναφέρουμε το αντίστοιχο αρχείο model. Το Store και το Model που αναφέραμε, με βάση το πρότυπο MVVM είναι το Model, δηλαδή περιέχει τα δεδομένα του προγράμματος. Στο ViewModel αναφέρουμε τα αρχεία store και ορίζουμε, πιθανόν, οπουδήποτε άλλα δεδομένα θα χρειαστούμε. Στο ViewController υλοποιούμε διάφορες μεθόδους και events. Τέλος, το View είναι το user interface, όπου επίσης δηλώνουμε τα αντίστοιχα αρχεία viewModel και viewController.

Αυτό σημαίνει ότι η αρχιτεκτονική μας είναι λίγο διαφορετική από το MVVM.



Διάγραμμα 3.3.2 - ExtJS Pattern

Το ExtJS μας δίνει τη δυνατότητα να υλοποιήσουμε ένα project και να στοχεύουμε είτε προσωπικούς υπολογιστές, είτε φορητές συσκευές, είτε και τα δύο. Αυτή είναι η διαφορά των φακέλων classic και modern. Ο φάκελος classic στοχεύει στους προσωπικούς υπολογιστές ενώ ο φάκελος modern στις φορητές συσκευές. Σε επίπεδο υλοποίησης η διαφορά βρίσκεται στο user interface (View), αλλά πιθανώς να βρίσκεται και στα αρχεία viewModel και viewController. Αυτό εξαρτάται από το τι θέλουμε να δείξουμε σε κάθε πλατφόρμα. Στη δική μας περίπτωση δε στοχεύουμε τις φορητές συσκευές και γι' αυτό τον λόγο δεν έχουμε υλοποιήσει τίποτα στον φάκελο modern.

Εν κατακλείδι, στον φάκελο app τοποθετούμε τον κοινό κώδικα ανεξαρτήτως την πλατφόρμα που στοχεύουμε. Αυτό σημαίνει σίγουρα τα δεδομένα του project μας (Model και Store) και μερικά από τα αρχεία ViewModel, ViewController και View. Τα υπόλοιπα αρχεία ViewModel, ViewController και View τα τοποθετούμε, αναλόγως με την πλατφόρμα που στοχεύουμε, στους αντίστοιχους φακέλους και με τις αντίστοιχες παραμετροποιήσεις που επιθυμούμε.

3.3.3.iii Παράδειγμα με κώδικα

Για να γίνει καλύτερα κατανοητή η προηγούμενη υποενότητα, θα δείξουμε ένα παράδειγμα με κώδικα και όλα τα στάδια της αρχιτεκτονικής που αναφέραμε προηγουμένως.

Ο πίνακας με όλα τα server calls που έχουμε χρησιμοποιήσει στο project μας, είναι ο 3.2.6 Πίνακας υλοποιημένων endpoints. Από αυτό τον πίνακα θα επιλέξουμε ένα για παράδειγμα. Συγκεκριμένα θα αναφερθούμε στο server call που κάνουμε για να πάρουμε τα μαθήματα του εκάστοτε καθηγητή.

Για να υλοποιήσουμε το συγκεκριμένο call θα πρέπει πρώτα να ξεκινήσουμε από το αρχείο Store.

```
01: Ext.define('LearningAnalytics.store.dashboard.Courses', {
02:   extend: 'Ext.data.Store',
03:
04:   storeId: 'Courses',
05:
06:   alias: 'store.courses',
07:
08:   model: 'LearningAnalytics.model.dashboard.Course',
09:
10:   autoLoad: true,
11:
12:   proxy: {
13:     type: 'ajax',
14:     actionMethods: {
15:       read: 'GET'
16:     },
17:     useDefaultXhrHeader: false,
18:     cors: true,
19:     headers: {
20:       'Authorization': ''
21:     },
22:     api: {
23:       read: '/api/1/courses'
24:     },
25:     writer: {
26:       type: 'json',
27:       allowSingle: true,
28:       writeAllFields: true
29:     },
30:     reader: {
31:       type: 'json',
32:       rootProperty: 'response.data'
```

```
33:   }
34: }
35: });
```

Στον παραπάνω κώδικα είναι ολοκληρωμένο το αρχείο store για το συγκεκριμένο call. Μερικές σημειώσεις για το αρχείο:

- Στη γραμμή 6 ορίζουμε το όνομα με το οποίο θα μπορούμε να το συνδέσουμε με τα άλλα αρχεία.
- Στη γραμμή 8 ορίζουμε το path του model.
- Στη γραμμή 15 ορίζουμε το HTTP Method.
- Στη γραμμή 19 ορίζουμε το header που θέλουμε να στείλουμε.
- Στη γραμμή 22 ορίζουμε το endpoint.

Επειδή σε κάθε server call ορίζουμε το URL και στα Headers το Authentication Token για ταυτοποίηση, για να είναι πιο εύκολα συντηρήσιμο φτιάξαμε μία singleton κλάση, στην οποία θα αναφερθούμε παρακάτω.

Στη συνέχεια φτιάχνουμε το αρχείο model. Αυτό που ορίσαμε στη γραμμή 8 του αρχείου store.

```
01: Ext.define('LearningAnalytics.model.dashboard.Course', {
02:   extend: 'LearningAnalytics.model.Base',
03:
04:   fields: [
05:     { name: 'id', type: 'int' },
06:     { name: 'fullname', type: 'string' },
07:     { name: 'shortname', type: 'string' }
08:   ]
09: });
10:
```

Στο συγκεκριμένο αρχείο ορίζουμε το όνομα και τον τύπο των πεδίων.

Αυτά τα αρχεία είναι αποθηκευμένα στον φάκελο app και συγκεκριμένα στους υποφακέλους store και model. Είναι έτοιμα να χρησιμοποιηθούν στο viewModel αρχείο μας. Το συγκεκριμένο call χρησιμοποιείται στο Dashboard.

```
01: Ext.define('LearningAnalytics.view.dashboard.DashboardModel', {
02:   extend: 'Ext.app.ViewModel',
03:   alias: 'viewmodel.dashboard',
04:
05:   requires: [
06:     'Ext.data.Store',
07:     'Ext.data.field.Integer',
08:     'Ext.data.field.String',
```

```
09:   'Ext.data.field.Boolean'  
10: ],  
11:  
12: stores: {  
13:   courses: {  
14:     autoLoad: true,  
15:     type: 'courses'  
16:   },  
17:   viewStudents: {  
18:     autoLoad: true,  
19:     type: 'viewStudents'  
20:   }  
21: },  
22:  
23: data: {  
24:   recs: null,  
25:   enrolledusercount: null,  
26:   viewStudentsChart: null  
27: }  
28: });  
29:
```

Μέσα στον φάκελο `classic/src/view/dashboard` υπάρχει το αρχείο `DashboardModel`, που είναι το `viewModel` για το `Dashboard`. Ο παραπάνω κώδικας δείχνει τη τελική έκδοση του αρχείου. Μέσα στο `viewModel` ορίζουμε τα `stores` στα οποία θα έχει πρόσβαση το αρχείο `view`. Στη συγκεκριμένη περίπτωση, στη γραμμή 12 ορίσαμε το `store courses`. Μπορούμε να χρησιμοποιήσουμε το αρχείο με το όνομα που του ορίσαμε στη γραμμή 3.

```
10: controller: 'dashboard',  
11: viewModel: {  
12:   type: 'dashboard'  
13: },
```

Στον ίδιο φάκελο με το `viewModel` υπάρχει το αρχείο `Dashboard`. Αυτό είναι το `View`, δηλαδή το `user interface` με διάφορα `css styles` που φτιάξαμε για το `Dashboard`. Οι 3 γραμμές δείχνουν τον τρόπο με τον οποίο το `view` μπορεί να έχει πρόσβαση στα αρχεία `viewModel` και `controller`. Τέλος, στον ίδιο φάκελο υπάρχει και το αρχείο `DashboardController`, το οποίο περιέχει διάφορες βοηθητικές μεθόδους.

Με την ολοκλήρωση του αρχείου `DashboardController`, έχουμε τελειώσει με το παράδειγμα και δείξαμε έναν πλήρη κύκλο με τα στάδια της αρχιτεκτονικής.

3.3.3.iv Singleton class

Το αρχείο `app/config/Runtime.js` είναι η singleton κλάση.

```
07: singleton : true,
08: config : {
09:   baseUrl: 'localhost:3000'
10: },
11:
12: constructor : function(config){
13:   this.initConfig(config);
14:   Ext.Ajax.on('requestexception', this.onRequest, this);
15:   Ext.Ajax.on('beforerequest', this.onBeforeRequest, this);
16: },
17:
18: onBeforeRequest : function(connection, options, eOpts) {
19:   options.url = this.getBaseUrl() + options.url;
20:   if (options.url.indexOf('sign_in') === -1) {
21:     var object = Ext.util.Cookies.get('AccessToken');
22:     var cookies = JSON.parse(object);
23:     var token = null;
24:     if (cookies !== null) {
25:       token = cookies.access_token
26:     }
27:     options.headers.Authorization = "Token token=" + token;
28:   }
29: },
30:
31: onRequest : function(conn, response, options, eOpts) {
32:   if (response){
33:     if (response.status === 401){
34:       Ext.Msg.alert({
35:         title:'Session timeout',
36:         message: 'Please login',
37:         buttons: Ext.Msg.OK,
38:         icon: Ext.Msg.INFO,
39:         draggable: false,
40:         fn: function(btn) {
41:           Ext.create('LearningAnalytics.view.authentication.Login');
42:           window.location.assign("");
43:         }
44:       });
45:     }
46:   }
47: },
```

Στη κλάση αυτή έχουμε ορίσει το URL που χτυπάνε όλα τα endpoints, γραμμή 9. Η μέθοδος `onBeforeRequest` εκτελείται πριν από κάθε server call. Σκοπός της είναι να ορίσει το ολοκληρωμένο URL και να περάσει το Authentication Token στο headers. Το Authentication Token μαζί με τις πληροφορίες του χρήστη, τις αποθηκεύουμε σε cookies κατά τη διάρκεια του login. Η μέθοδος `onRequest` εκτελείται όταν το HTTP response status είναι error, δηλαδή όταν είναι 4xx ή 5xx. Εμείς χειριζόμαστε το 401 Unauthorized όπου εμφανίζουμε ένα μήνυμα στον χρήστη και τον ανακατευθύνουμε στη σελίδα login.

3.3.4 UML Διαγράμματα

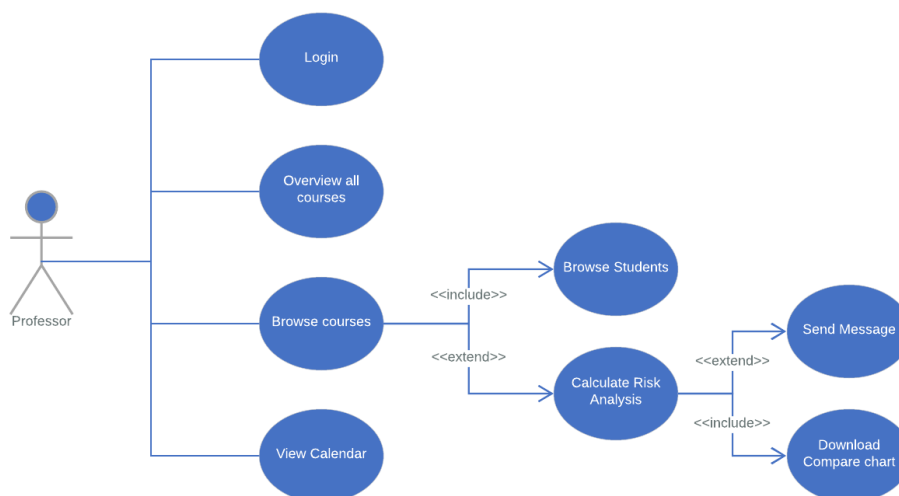
Η UML (Unified Modeling Language) πλέον είναι η πρότυπη γλώσσα μοντελοποίησης στη μηχανική λογισμικού. Χρησιμοποιείται για τη γραφική απεικόνιση, προσδιορισμό, κατασκευή και τεκμηρίωση των στοιχείων ενός συστήματος λογισμικού. Στόχος της είναι να προσφέρει μια τυποποιημένη μέθοδο μοντελοποίησης.

Με τη UML μπορούμε να σχεδιάσουμε πολλούς τύπους διαγραμμάτων. Εμείς όμως σχεδιάσαμε δύο διαγράμματα: το Use case diagram και το Sequence diagram.

3.3.4.i Use case diagram

Το use case diagram χρησιμοποιείται για την εμφάνιση των ορίων ενός συστήματος και των βασικών λειτουργιών του. Στο διάγραμμα αυτό σχεδιάζουμε τον χρήστη και τις αλληλεπιδράσεις του με το σύστημα. Το διάγραμμα δεν έχει πολλές λεπτομέρειες. Αντ' αυτού, απεικονίζει μία υψηλού επιπέδου επισκόπηση της σχέσης μεταξύ use cases, users και system.

Το Διάγραμμα 3.3.3 - Use Case Diagram δείχνει το use case diagram της πτυχιακής μας.

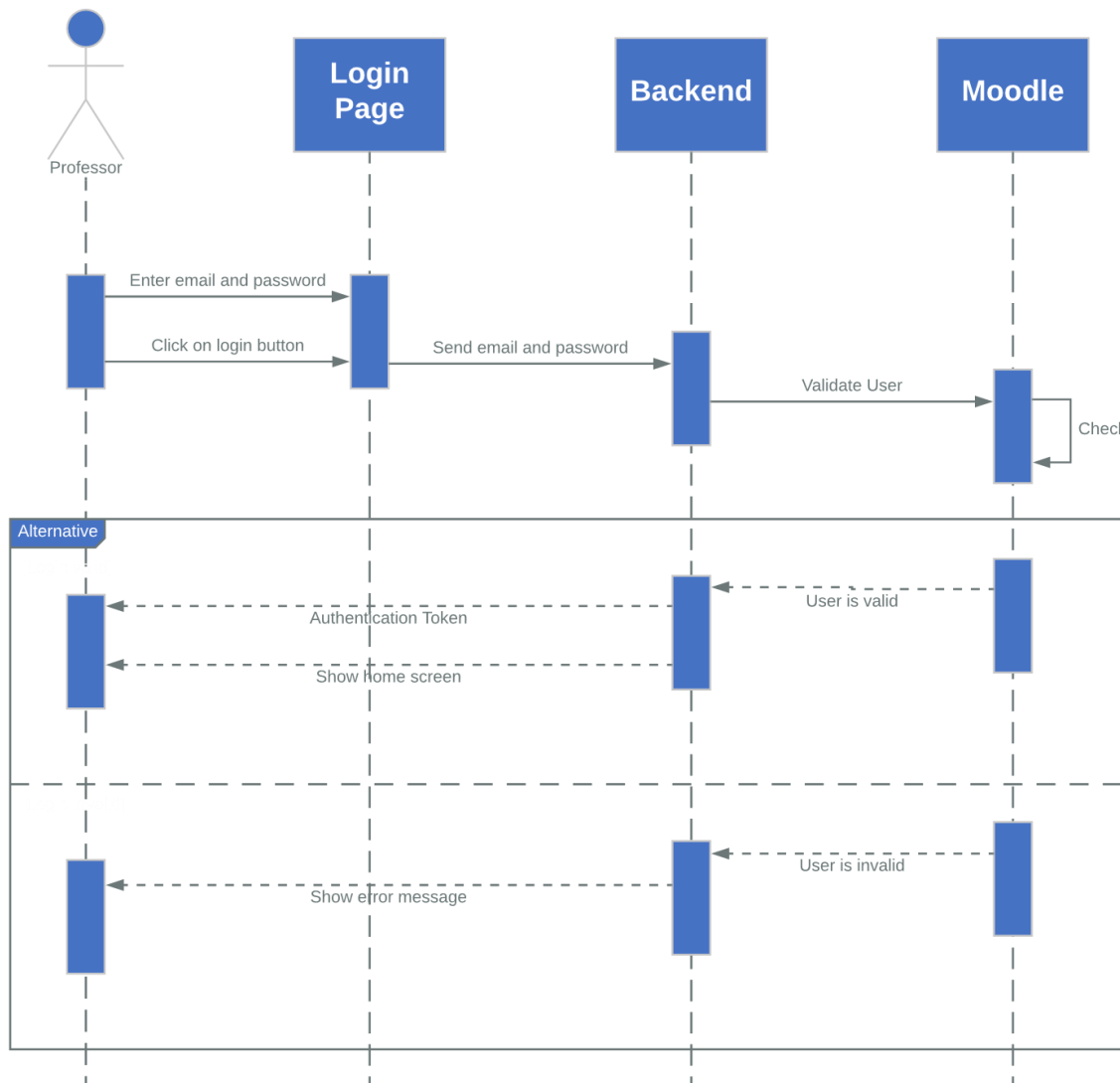


Διάγραμμα 3.3.3 - Use Case Diagram

3.3.4.ii Sequence diagram

Το sequence diagram είναι μια δημοφιλής λύση στη UML, διότι εστιάζει στις διαδικασίες, στα αντικείμενα και τα μηνύματα που ανταλλάσσονται μεταξύ τους για να εκτελέσουν μια λειτουργία. Είναι ένας τύπος αλληλεπίδρασης επειδή περιγράφει το πώς και σε ποια σειρά, μια ομάδα αντικειμένων συνεργάζονται. Η λειτουργία η οποία περιγράφεται στο Sequence diagram είναι η επεξήγηση ενός use case από το use case diagram.

Στο Διάγραμμα 3.3.4 - Sequence Diagram δείχνουμε το Sequence diagram για τη λειτουργία του login.



Διάγραμμα 3.3.4 - Sequence Diagram

Κεφάλαιο 4: Συμπεράσματα

Στην παρούσα πτυχιακή προτάθηκε η υλοποίηση ενός συστήματος για την ανάλυση δεδομένων ηλεκτρονικής μάθησης στο σύστημα Moodle. Πριν προχωρήσουμε στην υλοποίηση του συστήματος έπρεπε πρώτα να κατανοήσουμε το θεωρητικό υπόβαθρο. Οι όροι όπως, Ηλεκτρονική Μάθηση (e-Learning), Συστήματα Διαχείρισης Μάθησης (LMS), Learning Analytics και το σύστημα Moodle καλύφθηκαν στα προηγούμενα κεφάλαια, και παίζουν ένα σημαντικό ρόλο στη πτυχιακή μας.

Εν συντομία:

1. Η ηλεκτρονική μάθηση είναι οποιαδήποτε μορφή εκπαίδευσης, η οποία χρησιμοποιεί τους πόρους του διαδικτύου και γενικότερα τις δυνατότητες των ηλεκτρονικών υπολογιστών.
2. Ένα σύστημα διαχείρισης μάθησης (Learning Management System - LMS) είναι μια πλατφόρμα για τη διαχείριση, την τεκμηρίωση, την παρακολούθηση, την αναφορά και την παράδοση εκπαιδευτικών μαθημάτων ή προγραμμάτων κατάρτισης.
3. Το Moodle είναι ένα ευρέως γνωστό ΣΔΜ ανοικτού κώδικα.
4. Το Learning Analytics αφορά τη συλλογή των ιχνών που αφήνουν οι μαθητές κατά τη χρήση των ΣΔΜ και τη χρήση αυτών των ιχνών για τη βελτίωση της μάθησης.

Το σύστημα που υλοποιήσαμε είναι ένα εργαλείο Learning Analytics για το ΣΔΜ του Moodle. Ο καθηγητής συνδέεται στο σύστημα μας με τα στοιχεία λογαριασμού του Moodle του. Στο Moodle υπάρχουν όλες οι απαραίτητες πληροφορίες αλλά το πρόβλημα είναι ότι είναι αδόμετες. Μετά την επιτυχή σύνδεση, ο καθηγητής έχει πρόσβαση στο σύστημα μας και στα διάφορα γραφήματα / λειτουργίες που προσφέρουμε.

Η υλοποίηση του συστήματος μας χωρίστηκε σε δύο τεχνικά κομμάτια. Το Frontend και το Backend. Το Backend είναι υπεύθυνο για την επικοινωνία με το Moodle, για τη συλλογή / επεξεργασία των απαραίτητων πληροφοριών και τη σωστή δόμηση αυτών των πληροφοριών πριν σταλούν στον χρήστη. Το Frontend είναι ο συνδεδειμένος κρίκος του χρήστη και των πληροφοριών, είναι υπεύθυνο για την παρουσίαση των πληροφοριών σε ένα όμορφο και εύχρηστο γραφικό περιβάλλον.

Στόχος μας ήταν να προσφέρουμε ένα εύκολο και κατανοητό εργαλείο στους καθηγητές, όπου θα μπορούν να εκμεταλλευτούν όλα τα δεδομένα που έχει συλλέξει το Moodle. Παρ' όλα αυτά, το σύστημα μας δεν είναι πλήρες από άποψη λειτουργιών και υπάρχει μεγάλο περιθώριο ανάπτυξης.

Συνοπτικά οι κυριότερες λειτουργίες και γραφήματα τα οποία προσφέρουμε είναι:

1. Συνολική επισκεψιμότητα που είχαν όλα τα μαθήματα του καθηγητή.

2. Επισκεψιμότητα ανά μάθημα και την δυνατότητα αναζήτησης με συνδυασμό την χρονική περίοδο, τα modules και τους μαθητές.
3. Εμφάνιση σε γράφημα πίτας τον τύπο των αρχείων που έχει ανεβάσει ανά μάθημα.
4. Αποστολή μηνυμάτων είτε ατομικά είτε ομαδικά.
5. Εμφάνιση των events.
6. Και από τις σημαντικότερες λειτουργίες, η διαδικασία υπολογισμού του Risk Analysis. Δίνει στον καθηγητή μια εκτίμηση για το ποιοι μαθητές είναι σε κίνδυνο να αποτύχουν το μάθημα. Η διαδικασία αυτή καλύφθηκε αναλυτικά στο κεφ. 3.3.2.

4.1 Προτάσεις για μελλοντική ανάπτυξη

Παρακάτω αναφέρουμε συνοπτικά μερικές ιδέες οι οποίες πιστεύουμε θα μπορούσαν να αναπτυχθούν μελλοντικά:

- Χρήση Machine Learning για τη λειτουργία της Ανάλυσης Ρίσκου.
- Εύρεση και παρουσίαση των λιγότερο χρησιμοποιούμενων πόρων.
- Δυνατότητα προεπισκόπησης (embedded preview) των δραστηριοτήτων από την εφαρμογή μας.
- Δημιουργία Moodle Plugin, για την προβολή της εφαρμογής μας μέσω του συστήματος Moodle.
- Άνοιγμα πλατφόρμας στους μαθητές, με πληροφορίες (τι έχουν διαβάσει, τι δεν έχουν διαβάσει, τι θα πρέπει να μελετήσουν παραπάνω) και διαγράμματα για την συνολική τους επίδοση στο σύστημα Moodle.
- Μεταφορά υπολογισμού της Ανάλυσης Ρίσκου στο backend, για τη βελτίωση της επίδοσης.

Βιβλιογραφία

- [1] Talos, «Τι είναι Ruby, τι είναι Rails;», 13 February 2014. [Ηλεκτρονικό]. Available: <https://linuxmint.gr/index.php?topic=2811.0>. [Πρόσβαση 20 April 2018].
- [2] D. Kehoe, «What is Ruby on Rails?», 11 October 2013. [Ηλεκτρονικό]. Available: <http://railsapps.github.io/what-is-ruby-rails.html>. [Πρόσβαση 10 April 2018].
- [3] R. Branon και C. Essex, «Synchronous and asynchronous communication tools in distance education,» τόμ. 45, p. 36, 2001.
- [4] M. Hammond, «A review of recent papers on online discussion in teaching and learning in Higher Education,» *Online Learning (formally the Journal of Asynchronous Learning*, τόμ. 9, 2005.
- [5] G. Siemens, «Learning and Knowledge Analytics,» 5 August 2011. [Ηλεκτρονικό]. Available: <http://www.learninganalytics.net/?p=131>. [Πρόσβαση 16 08 2018].
- [6] E. Duval, «Learning Analytics and Educational Data Mining,» 30 01 2012. [Ηλεκτρονικό]. Available: <https://erikduval.wordpress.com/2012/01/30/learning-analytics-and-educational-data-mining/>. [Πρόσβαση 11 08 2018].
- [7] R. Ferguson, «Learning analytics FAQs,», 27 06 2014. [Ηλεκτρονικό]. Available: <http://www.slideshare.net/R3beccaF/learning-analytics-fa-qs>. [Πρόσβαση 11 08 2018].
- [8] A. Popovici και C. Mironov, «Students' Perception on Using eLearning Technologies,» 05 05 2015. [Ηλεκτρονικό]. Available: <https://www.sciencedirect.com/science/article/pii/S1877042815016468>. [Πρόσβαση 11 08 2018].
- [9] Lucidchart, «UML Use Case Diagram Tutorial,» [Ηλεκτρονικό]. Available: <https://www.lucidchart.com/pages/uml-use-case-diagram>. [Πρόσβαση 20 08 2018].
- [10] Lucidchart, «UML Class Diagram Tutorial,» [Ηλεκτρονικό]. Available: <https://www.lucidchart.com/pages/uml-class-diagram>. [Πρόσβαση 20 08 2018].
- [11] tutorialspoint, «MVVM – Introduction,» [Ηλεκτρονικό]. Available: https://www.tutorialspoint.com/mvvm/mvvm_introduction.htm. [Πρόσβαση 20 08 2018].
- [12] Sencha, «Introduction to Application Architecture,» [Ηλεκτρονικό]. Available: https://docs.sencha.com/extjs/6.0.0/guides/application_architecture/application_architecture.html. [Πρόσβαση 20 08 2018].

- [13 G. Siemens και R. Baker, «Learning Analytics and Educational Data Mining: Towards Communication and Collaboration,» 12 12 2016. [Ηλεκτρονικό]. Available: <http://www.columbia.edu/~rsb2162/LAKs%20reformatting%20v2.pdf>. [Πρόσβαση 20 08 2018].
- [14 Docebo, «eLearning Market Trends and Forecast 2017-2021,» 12 2016. [Ηλεκτρονικό]. Available: <https://www.docebo.com/resource/elearning-market-trends-and-forecast-2017-2021/>. [Πρόσβαση 20 08 2018].
- [15 Α. Μακρή και Δ. Βλαχόπουλος, «Ηλεκτρονική μάθηση: η πολυσημία και πολυπλοκότητα της έννοιας: Μία συστηματική βιβλιογραφική επισκόπηση,» 11 2017. [Ηλεκτρονικό]. Available: <https://eproceedings.epublishing.ekt.gr/index.php/openedu/article/viewFile/974/1325>. [Πρόσβαση 20 08 2018].
- [16 Β. Ραγάζου, Π. Β. Σιούλας και Κ. Φλώρος, «Ηλεκτρονική μάθηση. Η περίπτωση του Moodle στην επαγγελματική εκπαίδευση,» [Ηλεκτρονικό]. Available: <https://eproceedings.epublishing.ekt.gr/index.php/edusc/article/viewFile/368/330>. [Πρόσβαση 21 08 2018].
- [17 J. Lebowitz, «The Unexpected Impact Data Analytics Could Have On Your Career,» 10 05 2016. [Ηλεκτρονικό]. Available: <https://www.northeastern.edu/graduate/blog/data-analytics/>. [Πρόσβαση 26 08 2018].
- [18 M. Dean, «The Growing Impact Analytics Is Having On Education,» 07 06 2016. [Ηλεκτρονικό]. Available: <https://www.northeastern.edu/graduate/blog/learning-analytics/>. [Πρόσβαση 25 08 2018].
- [19 Lacey, «What are Learning Analytics?,» [Ηλεκτρονικό]. Available: <http://www.laceproject.eu/faqs/learning-analytics>. [Πρόσβαση 13 08 2018].
- [20 Harvard University, Institute for Quantitative Social Science, Gary King, «Preface: Big Data Is Not About The Data!,» 21 05 2016. [Ηλεκτρονικό]. Available: http://gking.harvard.edu/files/gking/files/prefaceorbigdataisnotaboutthedata_1.pdf. [Πρόσβαση 13 08 2018].
- [21 L. Rothfeld, «Smarter Education: The Rise of Big Data in the Classroom,» 03 09 2014. [Ηλεκτρονικό]. Available: <https://mashable.com/2014/09/03/education-data-video/>. [Πρόσβαση 13 08 2018].
- [22 M. Lynch, «What are the benefits of Learning Analytics?,» 27 08 2017. [Ηλεκτρονικό]. Available: <https://www.thetechedvocate.org/benefits-learning-analytics/>. [Πρόσβαση 25 08 2018].

- [23 M. Lynch, «Why are Learning Analytics so important?,» 04 08 2017. [Ηλεκτρονικό].
] Available: <https://www.thetechadvocate.org/learning-analytics-important/>.
[Πρόσβαση 25 08 2018].
- [24 McKinsey Global Institute, «Big data: The next frontier for innovation, competition,
] and productivity,» 05 2011. [Ηλεκτρονικό]. Available:
<https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/big-data-the-next-frontier-for-innovation>. [Πρόσβαση 13 08 2018].
- [25 New Media Consortium (NMC), «NMC Horizon Report > 2016 Higher Education
] Edition,» 27 01 2016. [Ηλεκτρονικό]. Available: <http://www.nmc.org/publication/nmc-horizon-report-2016-higher-education-edition/>. [Πρόσβαση 23 08 2018].

Παράρτημα Α': Λειτουργικό αντίγραφο της εφαρμογής

Παράρτημα Α':

Λειτουργικό αντίγραφο της εφαρμογής

Οι παρακάτω οδηγίες θα λειτουργήσουν σε Windows, MacOS και Linux.

```
01: sudo apt-get install docker-ce # not for Windows
02: sudo apt install docker-compose # not needed for Windows

03: sudo sysctl -w vm.max_map_count=262144 # not for Windows

04: mkdir project && cd project

05: git clone https://gist.github.com/edfd9542c6a6f4f53c7d1ed986db093c.git .
06: rm -rf .git # for Windows use: rmdir .git

07: git clone https://github.com/teokal/thesis-app
08: git clone https://github.com/teokal/thesis-app-client-side

09: docker-compose up

10: docker cp secrets.yml "$(docker-compose ps -q backend)":/myapp/config/secrets.yml
11: docker cp mapping.json "$(docker-compose ps -q backend)":/
12: docker-compose exec backend bash -c "cd / && curl -XPUT 'elasticsearch:9200/moodle_logs'
-d @mapping.json"
13: docker-compose exec backend bash -c 'rake import_logs_from_moodle'
```

Αναλυτικά (κάθε γραμμή από τις παραπάνω οδηγίες αντιστοιχεί σε ένα βήμα):

Βήμα 1ο: Εγκατάσταση του Docker Community Edition.

Για τα Windows, μπορείτε να κατεβάσετε το **Docker for Windows** από το Docker Store³.

Βήμα 2ο: Εγκατάσταση του πρόσθετου Docker Compose.

Για τα Windows, εγκαθίσταται μαζί με το **Docker for Windows**, οπότε δεν χρειάζεται κάποια επιπλέον κίνηση.

Βήμα 3ο: Αύξηση του `vm.max_map_count`.

Απαραίτητο για την επιτυχή έναρξη του Elasticsearch. Μπορείτε να βρείτε περισσότερες πληροφορίες στη σελίδα⁴ του Elasticsearch.

Βήμα 4ο: Δημιουργία και είσοδος στο φάκελο `project` όπου θα κατέβουν όλα τα απαραίτητα αρχεία για της εφαρμογής μας.

³ <https://store.docker.com/editions/community/docker-ce-desktop-windows>

⁴ <https://www.elastic.co/guide/en/elasticsearch/reference/5.6/vm-max-map-count.html>

Βήμα 5ο: Κατέβασμα των αρχείων (.env, docker-compose.yml, mapping.json, secrets.yml) που αναφέρονται παρακάτω. Τα αρχεία είναι διαθέσιμα στα συνοδευτικά αρχεία της πτυχιακής αλλά και ως Gist⁵ στο GitHub για ευκολότερο διαμοιρασμό τους.

Βήμα 6ο: Διαγραφή του φακέλου .git για τα αρχεία του gist καθώς δεν χρειάζεται.

Για τα Windows χρησιμοποιήστε rmdir .git

Βήμα 7ο: Λήψη της εφαρμογή για το Backend από το GitHub.

Βήμα 8ο: Λήψη της εφαρμογή για το FrontEnd από το GitHub.

Βήμα 9ο: Δημιουργία και έναρξη των containers με χρήση του Docker Compose.

⚠ Θα πρέπει να συμπληρώσετε το τελευταίο τμήμα (τις 4 μεταβλητές για το Moodle) του αρχείου secrets.yml με τα στοιχεία της βάσης δεδομένων του Moodle, για να προχωρήσετε.

Βήμα 10ο: Αντιγραφή του αρχείου secrets.yml μέσα στα αρχεία του backend.

⚠ Αυτό το αρχείο δεν πρέπει να γίνει ποτέ commit στο git repository, γιατί περιέχει κωδικούς που μπορεί να δώσουν ανεπιθύμητη πρόσβαση σε ξένους.

Βήμα 11ο: Αντιγραφή του αρχείου mapping.json μέσα στα αρχεία του backend.

Βήμα 12ο: Χρήση του Elasticsearch API, για τη δημιουργία περιγραφής (mapping) στο Elasticsearch για τα logs του Moodle.

Βήμα 13ο: Αντιγραφή των logs από τη βάση δεδομένων του Moodle στο Elasticsearch. Αυτή τη διαδικασία μπορεί να διαρκέσει αρκετή ώρα και μπορεί να εκτελεστεί οποτεδήποτε για την ενημέρωση της εφαρμογής μας με τα νέα logs.

Παρακάτω παρατίθενται τα περιεχόμενα των αρχείων που αναφέρθηκαν στο 5^ο βήμα.

Αρχείο .env:

```
MYSQL_ROOT_PASSWORD=mysqlr00tp@ssw0rd
MYSQL_DATABASE=thesis-app
MYSQL_USER=thesisuser
MYSQL_PASSWORD=p@ssworDdB
```

Αρχείο docker-compose.yml: (μεγάλη προσοχή στα κενά, απαιτούνται 2 ανά επίπεδο)

```
version: '3'
services:
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:5.6.11
    restart: always
    environment:
```

⁵ <https://gist.github.com/teokal/edfd9542c6a6f4f53c7d1ed986db093c>

- xpack.security.enabled=false
- ES_JAVA_OPTS=-Xms1g -Xmx1g
- discovery.type=single-node
- transport.host=127.0.0.0
- transport.tcp.port=9300
- http.host=0.0.0.0
- http.port=9200
- network.host=0.0.0.0

db:

```
image: mysql:5.7
restart: always
environment:
  MYSQL_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}
  MYSQL_DATABASE: ${MYSQL_DATABASE}
  MYSQL_USER: ${MYSQL_USER}
  MYSQL_PASSWORD: ${MYSQL_PASSWORD}
```

frontend:

```
build: './thesis-app-client-side'
volumes:
  - ./thesis-app-client-side:/project
restart: always
ports:
  - "80:1841"
command: ["sencha", "app", "watch"]
```

backend:

```
build: './thesis-app'
restart: always
ports:
  - "3000:3000"
volumes:
  - ./thesis-app:/myapp
environment:
  RAILS_ENV: development
command: bash -c "sleep 10 && rake db:migrate 2>/dev/null || rake db:setup && rails server -b 0.0.0.0"
links:
  - elasticsearch
  - db
```

Αρχείο mapping.json:

```
{
  "mappings": {
    "log": {
      "properties": {
        "id": { "type": "integer" },
        "eventname": { "type": "text" },
        "component": { "type": "keyword" },
        "action": { "type": "keyword" },
        "target": { "type": "keyword" },
        "objecttable": { "type": "keyword" },
        "objectid": { "type": "integer" },
        "crud": { "type": "keyword" },
        "edulevel": { "type": "integer" },
        "contextid": { "type": "integer" },
        "contextlevel": { "type": "integer" },
        "contextinstanceid": { "type": "integer" },
        "userid": { "type": "integer" },
        "courseid": { "type": "integer" },
        "relateduserid": { "type": "integer" },
        "anonymous": { "type": "integer" },
        "other": { "type": "text" },
        "timecreated": {
          "type": "date",
          "format": "epoch_second"
        },
      },
      "realuserid": { "type": "integer" }
    }
  }
}
```

Αρχείο `secrets.yml`: (μεγάλη προσοχή στα κενά, απαιτούνται 2 ανά επίπεδο)

```
development:
  secret_token:
  secret_key_base:
  database:
    :name: thesis-app
    :username: thesisuser
    :password: p@ssworDdB
    :host: db
    :port: 3306

database_moodle:
  :host: localhost
  :name: moodle
  :username: railsuser
  :password: r@1lsU$eR
  :port: 3306

MOODLE_DB_PREFIX: "mdl_"
MOODLE_HOST_URL: "http://localhost"
MOODLE_TOKEN: "moodle_token"
MOODLE_SERVICE_SHORT: "moodle_short_name"
```