

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ, ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

*Ανάπτυξη Πλατφόρμας Ασύγχρονης Εκπαίδευσης - Development of an
Asynchronous Learning Platform*

Διπλωματική Εργασία

Κωστάκη Ιωάννα

Υπεύθυνη Καθηγήτρια: Τσομπανοπούλου Παναγιώτα.
Επίκουρος Καθηγήτρια

Επιβλέπων Καθηγητής: Μποζάνης Παναγιώτης
Αναπληρωτής Καθηγητής

Βόλος, 2012

Ευχαριστίες

Ευχαριστώ θερμά την υπεύθυνη καθηγήτρια μου κ. Τσομπανοπούλου Παναγιώτα για την αμέριστη συμπαράσταση, υπομονή και εμπιστοσύνη που μου έδειξε κατά την διάρκεια ολοκλήρωσης της πτυχιακής μου εργασίας.

Ευχαριστώ πολύ τον καθηγητή κ. Παναγιώτη Μποζάνη που δέχτηκε να αναλάβει την επίβλεψη της διπλωματικής μου εργασίας.

Θα ήθελα να ευχαριστήσω επίσης την οικογένεια μου, που είναι δίπλα μου και με στηρίζει σε κάθε μου βήμα.

Περιεχόμενα

1. Εισαγωγή	4
1.1 Σκοπός.....	4
1.2 Σκοπός της πτυχιακής εργασίας	4
1.3 Επισκόπηση.....	7
2. Υλοποίηση της πλατφόρμας	8
2.1 Περιβάλλον Εργασίας.....	8
2.2 Τεχνολογίες που χρησιμοποιήθηκαν.....	10
2.3 Εγκατάσταση της Ruby on Rails	10
2.4 Δομή της πλατφόρμας.....	11
2.5 Επιμέρους Εργαλεία της πλατφόρμας.....	20
3. Χαρακτηριστικά της Πλατφόρμας	23
3.1 Συνολική Περιγραφή	23
3.1.1 Προοπτική της Πλατφόρμας.....	23
3.1.2 Χαρακτηριστικά Προϊόντος.....	24
3.1.3 Κατηγορίες χρηστών και τα χαρακτηριστικά τους	24
3.2 Λειτουργίες Πλατφόρμας	26
3.2.1 Μαθητής:: Επεξεργάζεται το προφίλ του.....	26
3.2.2 Μαθητής:: Παρακολουθεί ένα μάθημα	27
3.2.3 Δάσκαλος:: Δημιουργεί προφίλ δασκάλου	27
3.2.4 Δάσκαλος:: Δημιουργεί ένα μάθημα.....	28
3.2.5 Δάσκαλος:: Δημιουργεί την τάξη.....	29
3.2.6 Δάσκαλος:: Διδάσκει το μάθημα του	29
3.3 Interface.....	29
4. Η πλατφόρμα Online	34
4.1 Διεύθυνση πλατφόρμας.....	34
4.2 Heroku.....	34
5. Μελλοντικές Επεκτάσεις.....	35
Παράρτημα	37
Αναφορές.....	38

1. Εισαγωγή

1.1 Σκοπός

Σκοπός αυτού του εγγράφου είναι να περιγράψει τις απαιτήσεις λογισμικού για την πτυχιακή εργασία: Ανάπτυξη Πλατφόρμας Ασύγχρονης Εκπαίδευσης - Development of an Asynchronous Learning Platform.

1.2 Σκοπός της πτυχιακής εργασίας

Πλατφόρμα ασύγχρονης εκπαίδευσης

Σκοπός της πλατφόρμας ασύγχρονης εκπαίδευσης είναι η δημιουργία εικονικών τάξεων από τους χρήστες της πλατφόρμας που θέλουν να διδάξουν κάποιο μάθημα. Μαθητές σε αυτές τις εικονικές τάξεις μπορεί να είναι οποιοσδήποτε χρήστης θέλει να παρακολουθήσει κάτι που τον ενδιαφέρει ενώ βρίσκεται απομακρυσμένα.

Περιορισμοί στο είδος των μαθημάτων που θα διδάσκονται δεν υπάρχουν. Οπότε κάποιος χρήστης για παράδειγμα μπορεί να δημιουργήσει μία εικονική τάξη για εκμάθηση μιας ξένης γλώσσας ή μιας γλώσσας προγραμματισμού αλλά ακόμα και να δημιουργήσει και μια τάξη για μαθήματα μαγειρικής ή γυμναστικής.

Γιατί κάποιος να διδάξει στην πλατφόρμα ασύγχρονης εκπαίδευσης:

Ο κάθε χρήστης μπορεί να διδάξει στην πλατφόρμα και να δημιουργήσει μια δική του εικονική τάξη. Πολλά είναι τα παραδείγματα όπου κάποιος χρήστης θα έχει όφελος διδάσκοντας κάποιο μάθημα. Παρακάτω παραθέτω κάποια από αυτά.

Ο χρήστης που μόλις έχει αποφοιτήσει από τη σχολή του και είναι νέος δάσκαλος/καθηγητής. Δημιουργώντας μια εικονική τάξη αποκτά εμπειρία στην διδασκαλία αλλά και προϋπηρεσία.

Ο χρήστης που είναι δάσκαλος/καθηγητής σε μια πραγματική τάξη και για κάποιο λόγο δεν μπορεί να παραβρεθεί. Έτσι μπορεί να δημιουργήσει μια εικονική τάξη ώστε να μην χάσει το μάθημα του και να τα διδάξει απομακρυσμένα.

Ο χρήστης που θέλει να βγάλει οικονομικό κέρδος. Αυτό είναι δυνατόν αφού μελλοντικά θα δίνεται η δυνατότητα στον χρήστη να κοστολογεί το μάθημα του και να βγάζει κέρδος από την πλατφόρμα

Ο χρήστης που απλά έχει ελεύθερο χρόνο και θέλει να τον αξιοποιήσει δημιουργικά.

Γιατί κάποιος να παρακολουθήσει μάθημα στην πλατφόρμα ασύγχρονης εκπαίδευσης:

Αντίστοιχα με τους δημιουργούς μιας εικονικής τάξης, υπάρχουν πολλά παραδείγματα χρηστών οι οποίοι θέλουν να παρακολουθήσουν μια διαδικτυακή τάξη. Κάποια από αυτά αναφέρονται παρακάτω.

Ο χρήστης που θέλει να παρακολουθήσει κάποιο μάθημα από το χώρο του και στο χρόνο τον οποίο επιθυμεί. Όλοι και περισσότεροι χρήστες του διαδικτύου αναζητούν να παρακολουθήσουν κάτι που τους ενδιαφέρει απομακρυσμένα. Η πλατφόρμα ασύγχρονης εκπαίδευσης δίνει αυτή την δυνατότητα.

Ο χρήστης ο οποίος δεν έχει την οικονομική δυνατότητα να παρακολουθήσει κάποιο μάθημα. Εφόσον θα υπάρχουν μαθήματα που διατίθενται δωρεάν, κάθε χρήστης κάθε χρήστης θα μπορεί να παρακολουθήσει το μάθημα που επιθυμεί.

Ο χρήστης ο οποίος θέλει να παρακολουθήσει κάποιο μάθημα το οποίο δεν μπορεί να βρει στον τόπο κατοικίας του. Αν κάποιος χρήστης, για παράδειγμα θέλει να μάθει μια εξεζητημένη ξένη γλώσσα είναι πιθανό να δυσκολευτεί να βρει κάποιο δάσκαλο στο μέρος που μένει. Αντιθέτως στην πλατφόρμα εκπαίδευσης είναι ευκολότερο να βρει κάποιος μια τέτοια τάξη.

Υπάρχουσες πλατφόρμες εκπαίδευσης

Πλατφόρμες Πανεπιστημίων

Παρόμοιες πλατφόρμες εκπαίδευσης με online μαθήματα παρέχονται από τα μεγαλύτερα Πανεπιστήμια του κόσμου. Κάποια ενδεικτικά Πανεπιστήμια είναι το MIT[1], το Πανεπιστήμιο του Cambridge[2] και το Stanford[3].

Σε αυτά τα πανεπιστήμια δίνεται η δυνατότητα στον χρήστη να αναζητήσει κάποιο μάθημα που θέλει και αν προσφέρεται να το παρακολουθήσει. Η εγγραφή του χρήστη σε αυτές τις πλατφόρμες είναι ανάλογα με το πανεπιστήμιο είτε υποχρεωτική είτε όχι.

Τα online μαθήματα αυτών των πανεπιστημίων διδάσκονται από τους καθηγητές των εκάστοτε πανεπιστημίων και οποιοσδήποτε χρήστης μπορεί να τα παρακολουθήσει.

Κύρια χαρακτηριστικά αυτών των διαδικτυακών τάξεων είναι:

- Στην εισαγωγή γίνεται περιγραφή του μαθήματος, λίγα λόγια για τον διδάσκοντα, αναφέρεται το απαραίτητο υπόβαθρο του χρήστη καθώς και η εκτιμώμενη διάρκεια που θα χρειαστεί ο χρήστης για να ολοκληρώσει το μάθημα

- Η διαδικτυακή τάξη αποτελείται από αρκετά video με τις διαλέξεις των καθηγητών, από quiz και ασκήσεις που είτε δίνονται οι λύσεις στους χρήστες είτε ο χρήστης κάνει submit τις απαντήσεις του και παρουσιάζεται στον χρήστη το ποσοστό της επιτυχίας του.
- Ο χρήστης είναι υπεύθυνος για το πόσο θα διαρκέσει το μάθημα που παρακολουθεί

Πλατφόρμες σχολικών τάξεων

Εκτός από τις πλατφόρμες πανεπιστημίων, διαδικτυακή τάξη μπορούν να δημιουργήσουν και οι δάσκαλοι σχολικών τάξεων.

Έτσι μπορεί κάποιος δάσκαλος να διδάξει στους μαθητές του με την βοήθεια της τεχνολογίας.

Δηλαδή προσομοιώνεται η πραγματική τάξη με την βοήθεια διάφορων εργαλείων εκπαίδευσης όπως live-streaming ώστε ο δάσκαλος να κεντρίσει το ενδιαφέρον των μαθητών.

Λοιπές πλατφόρμες εκπαίδευσης

Μπορεί κάποιος χρήστης να βρει διάφορες πλατφόρμες που ακολουθούν κυρίως την λογική των site των πανεπιστημίων.

Ο χρήστης που θέλει να δημιουργήσει μια διαδικτυακή τάξη, ανεβάζει κάποιο υλικό και όποιος θέλει να παρακολουθήσει το μάθημα ανά πάσα στιγμή έχει πρόσβαση σε αυτά τα αρχεία. Στην ουσία μαθαίνει μόνος του από το διαθέσιμο υλικό.

Γιατί κάποιος να διαλέξει την πλατφόρμα ασύγχρονης εκπαίδευσης

Η διαφορά αυτής της πλατφόρμας με τις υπάρχουσες είναι η προσομοίωση μιας πραγματικής τάξης, με την έννοια ότι προσπαθεί να συνδυάσει την άμεση επικοινωνία με τον δάσκαλο (όπως γίνεται με τις πλατφόρμες σχολικών τάξεων) σε μια ποικιλία μαθημάτων και δραστηριοτήτων (όπως κάποιος να βρει και σε υπόλοιπες πλατφόρμες).

Όνομα πλατφόρμας ασύγχρονης εκπαίδευσης

LearnTogether

1.3 Επισκόπηση

Παρακάτω παρέχεται μια επισκόπηση αυτού του εγγράφου και της οργάνωσης του.

Στην επόμενη ενότητα (ενότητα 2) θα γίνει αναφορά στην υλοποίηση της πλατφόρμας και στις τεχνολογίες και τεχνικές που χρησιμοποιήθηκαν για την δημιουργία της.

Επίσης θα δοθεί έμφαση στην δομή του προγράμματος αλλά και σε ότι άλλο είναι απαραίτητο για την κατανόηση της διαδικασίας της ανάπτυξης της πλατφόρμας.

Θα γίνει αναφορά στα χαρακτηριστικά της πλατφόρμας και θα αναφερθούν αναλυτικά οι λειτουργίες της καθώς και οι δυνατές ενέργειες των χρηστών (ενότητα 3).

Τέλος θα δοθεί το url της πλατφόρμας μαζί με όλες τις απαραίτητες πληροφορίες για την διαδικασία του deployment (ενότητα 4) και θα δοθούν οι μελλοντικές επεκτάσεις της πλατφόρμας(ενότητα 5).

2. Υλοποίηση της πλατφόρμας

2.1 Περιβάλλον Εργασίας

Για την δημιουργία της ασύγχρονης πλατφόρμας εκπαίδευσης το λειτουργικό σύστημα που χρησιμοποιήσα είναι Ubuntu[4]. Η βάση δεδομένων που χρησιμοποιώ είναι sqlite3[5]. Τέλος ο κώδικας για την δημιουργία της πλατφόρμας είναι σε γλώσσα Ruby με framework Rails[6].

Ruby on Rails:

Λίγα λόγια:

Η Ruby on Rails είναι ένα open source[7] full-stack[8] web application framework[9] για την γλώσσα προγραμματισμού Ruby[10].

Η Ruby on Rails δεν πρέπει να συγχέεται με την Ruby, η οποία είναι μια γενικής χρήσης γλώσσα προγραμματισμού, στην οποία τρέχει η Ruby on Rails. Η Ruby από μόνη της υπάρχει πάνω από 10 χρόνια πριν την πρώτη έκδοση της Ruby on Rails. Η Rails είναι ένα full-stack framework, πράγμα που σημαίνει ότι δίνει την δυνατότητα στον web developer να συλλέξει πληροφορίες από τον web server, κάνοντας επερωτήσεις στην βάση δεδομένων και να εμφανίσει τα αποτελέσματα. Ως εκ τούτου, η Rails διαθέτει ένα σύστημα δρομολόγησης που είναι ανεξάρτητο από τον web server.

Ιστορικά:

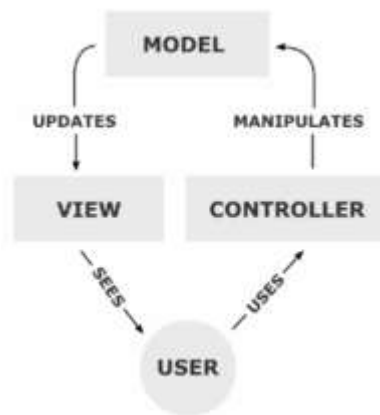
Ο David Heinemeier Hansson[11] δημιούργησε την Ruby on Rails βασισμένος στην δική του δουλειά στο Basecamp, ένα εργαλείο διαχείρισης με 37signals[12] (τώρα μια web application εταιρία).

Ο Hansson κυκλοφόρησε για πρώτη φορά την Rails ως open source τον Ιούλιο του 2004 αλλά χωρίς να δώσει τα δικαιώματα για το έργο του μέχρι τον Φεβρουάριο του 2005.

Τεχνική επισκόπηση:

Η Rails βασίζεται στο Model-View-Controller πρότυπο το οποίο χωρίζει την εφαρμογή σε τρία μέρη:

- Τον controller, που είναι η λογική της εφαρμογής. Συντονίζει την αλληλεπίδραση μεταξύ χρήστη, των views και του model. Ο controller επίσης προσφέρει σημαντικές βοηθητικές υπηρεσίες. Είναι υπεύθυνος για την δρομολόγηση των εξωτερικών αιτήσεων σε εσωτερικές δράσεις. Διαχειρίζεται helper modules μεθόδους, οι οποίες επεκτείνουν τις δυνατότητες των views χωρίς να τους επιβαρύνουν με πολλά δεδομένα. Διαχειρίζεται sessions δίνοντας την εντύπωση στους χρήστες μιας συνεχούς αλληλεπίδρασης στις εφαρμογές τους
- Τα Models, τα οποία αντιπροσωπεύουν τις πληροφορίες (δεδομένα) από την εφαρμογή και τους κανόνες να διαχειριστούν τα δεδομένα. Στις περισσότερες περιπτώσεις κάθε πίνακας στην βάση δεδομένων αντιστοιχεί σε ένα μοντέλο στην εφαρμογή. Το μεγαλύτερο μέρος της επιχειρηματικής λογικής θα επικεντρωθεί στα μοντέλα.
- Τα Views, τα οποία εμφανίζουν τα δεδομένα στο χρήστη. Η δουλειά του είναι να παρέχει τα δεδομένα στον web browser ή σε οποιοδήποτε άλλο εργαλείο χρησιμοποιείται για να κάνει κάποιο αίτημα στην εφαρμογή. Ένα αρχείο View της Rails είναι ένα ERb πρόγραμμα που μοιράζεται δεδομένα με τους controllers μέσω αμοιβαίων προσπελάσιμων μεταβλητών.



Μια τυπική συνεργασία των συνιστωσών MVC

Η δομή του framework:

Η Ruby on Rails χωρίζεται σε διάφορα πακέτα:

- ActiveRecord (Object-relational mapping[13] για πρόσβαση στην βάση δεδομένων),
- ActiveSupport(παρέχει web services),
- ActionPack,
- ActiveSupport

➤ ActionMailer

Εκτός από τα standard πακέτα, οι προγραμματιστές μπορούν να εισάγουν plugins[14] ώστε να επεκτείνουν τα υπάρχοντα πακέτα.

Deployment

Στην Ruby on Rails συχνά εγκαθίσταται, χρησιμοποιώντας RubyGems[15], ένα package manager[16] το οποίο περιλαμβάνεται με τις τρέχουσες εκδόσεις της Ruby. Πολλά δωρεάν Unix συστήματα[17] υποστηρίζουν την εγκατάσταση της Ruby on Rails και των εξαρτήσεων της μέσω των δικών τους πακέτων διαχείρισης συστήματος(package management system[18]).

2.2 Τεχνολογίες που χρησιμοποιήθηκαν

Η εφαρμογή υλοποιήθηκε με την χρήση της MVC αρχιτεκτονικής και δεν πρέπει να συγχέεται με την 3-tier αρχιτεκτονική. Οι ομοιότητες είναι πάρα πολλές γιατί και στις 2 περιπτώσεις έχουμε τον client, τον server και τα δεδομένα απλά η διαφορά είναι ότι στην 3-tier αρχιτεκτονική ο client δεν έχει την δυνατότητα να επικοινωνήσει άμεσα με τα δεδομένα κάτι που γίνεται στην MVC αρχιτεκτονική.

Βασικός δρομολογητής και διαχειριστής των request-response καθώς και υπεύθυνος για το routing της εφαρμογής είναι το framework της Rails. Με την εγκατάσταση της Rails γίνεται διαθέσιμος ένας virtual server με όνομα web-rick.

Η Rails είναι υπεύθυνη για το port assignment αυτού του server όπου αυτός τρέχει.

Κατά σύμβαση ο server αυτός ακούει στην url 0.0.0.0 (όλο το τοπικό δίκτυο) και στην πόρτα 3.000.

2.3 Εγκατάσταση της Ruby on Rails

Όπως αναφέρθηκε παραπάνω το περιβάλλον που δούλεψα για την ανάπτυξη της πλατφόρμας είναι Ubuntu 12.04. Παρακάτω περιγράφονται τα βήματα για την εγκατάσταση της Ruby on Rails:

- Βήμα 1. Εγκατάσταση git και cURL
Αρχικά πρέπει να γίνει update έτσι ώστε όλα τα πακέτα να είναι ενημερωμένα:

```
sudo apt-get update
```

Σε περίπτωση που δεν υπάρχει εγκαταστημένο curl στο σύστημα το εγκαθιστούμε:

```
sudo apt-get install git
```

➤ Βήμα 2. Εγκατάσταση RVM

Το RVM (Ruby Version Manager) είναι ένα πρόγραμμα που επιτρέπει την χρήση διάφορων εκδόσεων της Ruby σε έναν server. Οι εντολές είναι:

```
curl -L get.rvm.io | bash -s stable
```

Στην συνέχεια φορτώνουμε το RVM:

```
source ~/.rvm/scripts/rvm
```

Μετά εγκαθιστούμε επιπλέον εξαρτήσεις που καθορίζονται από τον RVM:

```
rvm requirements
```

➤ Βήμα 3. Εγκατάσταση της Ruby

Για την εγκατάσταση της Ruby δίνεται η εντολή:

```
rvm install 1.9.3
```

Στην συνέχεια επιλέγουμε την έκδοση της Ruby που θέλουμε να χρησιμοποιήσουμε (ή κάνουμε την επιλογή default ώστε να μην χρειάζεται να επιλέξουμε) την έκδοση ξανά.

➤ Βήμα 4. Εγκατάσταση της Rails

Για να εγκαταστήσουμε την Rails, απλά την εγκαθιστούμε χρησιμοποιώντας το κατάλληλο gem. Θα εγκατασταθεί αυτόματα η τελευταία έκδοση.

```
gem install rails
```

2.4 Δομή της πλατφόρμας

Όπως προαναφέρθηκε η Ruby on Rails ανήκει στα framework που υποστηρίζουν την αρχιτεκτονική MVC. Στη συνέχεια θα γίνει αναφορά των

Models, View και Controllers που δημιουργήθηκαν για την ανάπτυξη της πλατφόρμας καθώς και κάποιων ορισμών απαραίτητων για την κατανόηση της δομής του προγράμματος:

➤ **Controllers**

Με τον όρο REST(Representational State Transfer) περιγράφουμε ένα παράδειγμα αρχιτεκτονικής για web applications οι οποίες ζητάνε και διαχειρίζονται web resources χρησιμοποιώντας τις καθιερωμένες HTTP μεθόδους GET, POST,PUT και DELETE.

Στην rails μια RESTful δρομολόγηση παρέχει ταιρίασμα μεταξύ των HTTP ρημάτων και των URLs διευθύνσεων για τις ενέργειες που γίνονται στον controller.

Κάθε καταχώρηση στο route file δημιουργεί επτά διαφορετικά routes στην εφαρμογή:

HTTP Verb	action
GET	index
GET	new
POST	create
GET	show
GET	edit
PUT	update
DELETE	destroy

Σε περίπτωση που χρειάζονται άλλες ή διαφορετικές συναρτήσεις, τις προσθέτουμε στο route file και παράγεται το αντίστοιχο URL.

Κάθε μία από τις ενέργειες (action) που αντιστοιχεί στις εκάστοτε URLs απεικονίζεται ως μία συνάρτηση μέσα στον αντίστοιχο controller. Έτσι για παράδειγμα όταν επισκέπτεται κάποιος την index URL τότε η Ruby on Rails αναζητεί μέσα στον εκάστοτε controller μία συνάρτηση με όνομα index. Μετά την ολοκλήρωση της εκτέλεσης του σώματος της προηγούμενης συνάρτησης το αρχείο index.html.erb(index_view) στέλνεται στο browser σαν απάντηση.

Παρακάτω παρουσιάζονται οι controllers που δημιουργήθηκαν και μια σύντομη περιγραφή των λειτουργιών των μεθόδων τους:

application_controller

Ο application_controller δημιουργείται αυτόματα από την Ruby on Rails και οι υπόλοιποι controllers του project είναι απόγονοι του. Ο ApplicationController είναι απόγονος του ActionController::Base ο οποίος είναι η ρίζα της ιεραρχίας των controllers στην rails, παρέχοντας

έτσι μια σειρά από χρήσιμες μεθόδους κάνοντας τες διαθέσιμες στο υπόλοιπο πρόγραμμα.

Όταν κάνει κάποιος sign in (συνάρτηση `after_sign_in_path_for`) ή sign up (συνάρτηση `after_sign_up_path_for`) γίνεται redirect στο dashboard του χρήστη.

home_controller

Με την δημιουργία κάποιου controller, δημιουργείται και το αντίστοιχο view στον φάκελο `app/views` ώστε να δημιουργηθεί ο κατάλληλος html κώδικας για τις σελίδες της πλατφόρμας. Ο `home_controller` δημιουργήθηκε έτσι ώστε να αναπτυχθεί η αρχική σελίδα της πλατφόρμας. Έτσι στον `home_controller` υπάρχει η κενή συνάρτηση `index`.

courses_controller

Σε αυτόν τον controller γίνεται η δημιουργία κάποιου μαθήματος με την δημιουργία της μεταβλητής `@course` (συνάρτηση `create`). Δίνεται η δυνατότητα της ενημέρωσης του μαθήματος (συνάρτηση `update`) και της διαγραφής του (συνάρτηση `destroy`).

Ο δάσκαλος έχει την δυνατότητα να στείλει κάποιο μήνυμα στην τάξη του. Αυτό επιτυγχάνεται με την δημιουργία της μεταβλητής `@message` όπου υπάρχει το μήνυμα του δασκάλου (συνάρτηση `send_message`). Για την διαδικτυακή τάξη υπάρχει η δυνατότητα να γίνει upload ένα αρχείο(συνάρτηση `upload_file`).

Τέλος έχει δημιουργηθεί η συνάρτηση `streaming` όπου υπάρχει ο κώδικας για την δημιουργία του `streaming` που χρησιμοποιείται στην διαδικτυακή τάξη.

notifications_controller

Όταν ο χρήστης κάνει αίτηση παρακολούθησης ενός μαθήματος τότε δημιουργείται η μεταβλητή `@notification` για τον συγκεκριμένο χρήστη (συνάρτηση `create`).

Όταν ο δάσκαλο λάβει το αίτημα και κάνει δεκτό ή απορρίπτει το αίτημα του μαθητή η μεταβλητή `@notification` ενημερώνεται (συνάρτηση `update`) και ο μαθητής λαμβάνει μια αντίστοιχη κοινοποίηση στο dashboard.

profiles_controller

Οι συναρτήσεις που υπάρχουν σε αυτή την ενότητα αφορούν την δημιουργία προφίλ δασκάλου με την δημιουργία της μεταβλητής `@teacher` (συνάρτηση `edit_teacher`), την ενημέρωση αυτού του προφίλ όταν ο χρήστης αλλάξει κάποια δεδομένα(συνάρτηση `update_teacher`) καθώς και την διαγραφή του προφίλ του δασκάλου(συνάρτηση `destroy`) . Επίσης υπάρχει συνάρτηση για την ενημέρωση του προφίλ του χρήστη(συνάρτηση `update`).

Τέλος υπάρχουν συναρτήσεις για το δημόσιο προφίλ του μαθητή(συνάρτηση `public_profile`) και για το δημόσιο προφίλ του

δασκάλου(συνάρτηση show).

searches_controller

Σε αυτόν τον controller γίνεται η αναζήτηση των μαθημάτων με επερώτηση στη βάση δεδομένων. Σε περίπτωση που ο χρήστης δεν έχει κάνει login στη πλατφόρμα δεν μπορεί να δει το προφίλ του δασκάλου στα μαθήματα που εμφανίζονται από την αναζήτηση και του ζητάτε να κάνει εγγραφή πρώτα. Τα αποτελέσματα εμφανίζονται στο dashboard του χρήστη. Τέλος στην μεταβλητή @students_in_course αποθηκεύονται οι μαθητές της τάξης (συνάρτηση edit).

users_controller

Σε αυτό τον controller δημιουργείται η μεταβλητή @notification_for_me όπου αποθηκεύονται τα μηνύματα για την απόρριψη ή την αποδοχή του μαθητή από τον δάσκαλο και η μεταβλητή @notifications_from_me όπου αποθηκεύονται τα μηνύματα που έχει στείλει ο χρήστης.

Επίσης δημιουργείται η μεταβλητή @messages που αποθηκεύονται τα μηνύματα που στέλνει ο δάσκαλος στους μαθητές του(συνάρτηση dashboard).

course_days_controller

Σε αυτό τον controller γίνεται η δημιουργία του ημερολογίου, όπου θα δηλωθεί από τον δάσκαλο η μέρα που θα γίνει το μάθημα με την μεταβλητή @course_day.day και η ώρα που θα ξεκινήσει και θα τελειώσει το μάθημα, με τις μεταβλητές @course_day.start_at και @course_day.ends_at αντίστοιχα.

➤ **Models**

Σε μια προεπιλεγμένη ρύθμιση, ένα model στην Ruby on Rails αντιστοιχεί σε ένα πίνακα στην βάση δεδομένων. Κατά σύμβαση, ένα model που ονομάζεται User αντιστοιχίζεται στην βάση δεδομένων στον πίνακα users και το μοντέλο θα έχει όνομα αρχείου user.rb στον φάκελο app/models.

Σε κάθε αρχείο που περιγράφει ένα model περιέχονται οι συσχετίσεις του με τα άλλα models. Οι συσχετίσεις που χρησιμοποιήθηκαν είναι:

belongs_to - has_one: 1-1 συσχέτιση

belongs_to - has_many: 1-N συσχέτιση

Παρακάτω παρουσιάζονται τα models που δημιουργήθηκαν και μια σύντομη περιγραφή του περιεχομένου τους:

classroom

Σε αυτό το model περιέχονται όλα τα πεδία που είναι απαραίτητα για την έναρξη μιας διαδικτυακής τάξης.

Το model classroom έχει τα πεδία: teacher_id, user_id, course_id

και `upload_file`.

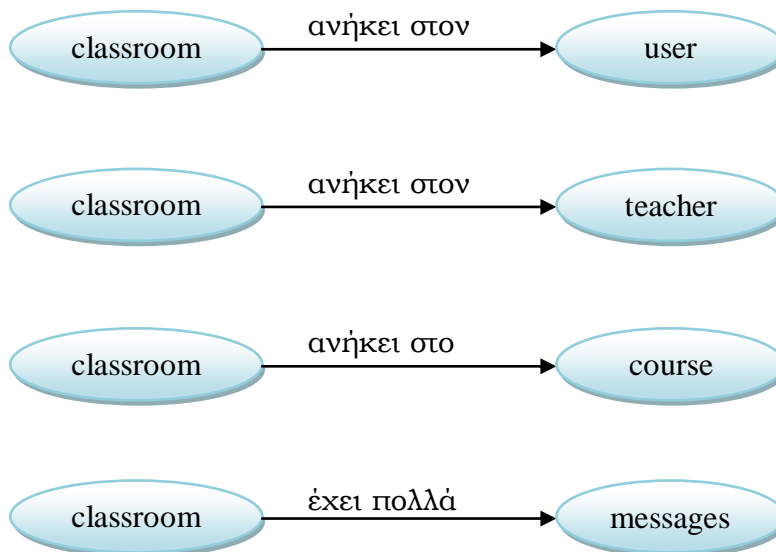
`teacher_id`: αυτό το πεδίο περιέχει το `id` του δασκάλου που κάνει το μάθημα.

`user_id`: αυτό το πεδίο περιέχει το `id` του μαθητή που παρακολουθεί το μάθημα.

`course_id`: κάθε μάθημα έχει ένα μοναδικό `id`. Αυτό το πεδίο περιέχει αυτό το `id`.

`file`: περιέχει το αρχείο που γίνεται `upload` στην διαδικτυακή τάξη.

Συσχετίσεις του `model classroom` με τα άλλα `models`:



course

Σε αυτό το `model` περιέχονται όλα τα πεδία που χρειάζονται για να προσδιοριστεί ένα μάθημα.

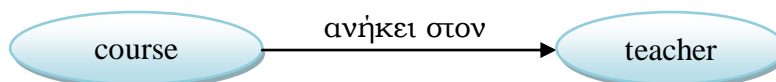
Το `model course` έχει τα πεδία: `title`, `details` και `classroom_attributes`.

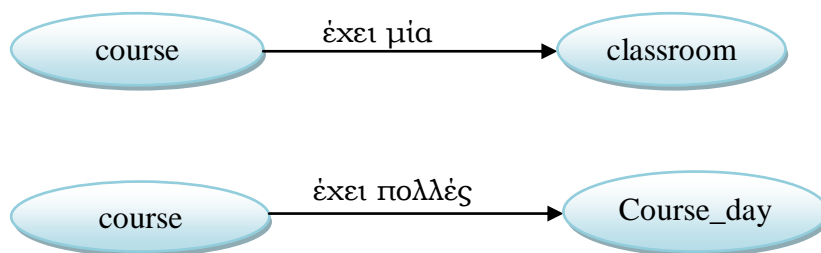
`title`: το πεδίο αυτό περιέχει το όνομα του μαθήματος.

`details`: περιέχονται οι πληροφορίες που ο δάσκαλος έχει εισάγει για το μάθημα που θα διδάξει.

`classroom_attributes`: η μεταβλητή αυτή χρειάζεται για να έχουμε πρόσβαση στις μεταβλητές της τάξης.

Συσχετίσεις του `model course` με τα άλλα `models`:





notification

Αυτό το model δημιουργήθηκε για να μπορεί να στέλνεται ειδοποίηση στον μαθητή για το αν έγινε δεκτός στο μάθημα που έχει κάνει αίτηση παρακολούθησης, από τον δάσκαλο

Το model notification έχει τα πεδία: `student_id`, `to`, `status` και `course_id`.

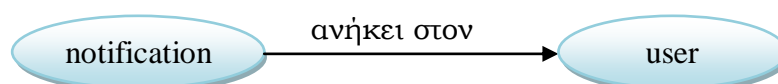
`student_id`: περιέχει το `id` του μαθητή που έκανε αίτηση στον δάσκαλο για να παρακολουθήσει το μάθημα του.

`to`: περιέχει το μάθημα για το οποίο έκανε αίτηση παρακολούθησης ο μαθητής.

`status`: στο πεδίο `status` αναφέρεται ποιο είναι το αποτέλεσμα της αίτησης, δηλαδή αν έχει γίνει δεκτή και ο μαθητής μπορεί να παρακολουθήσει το μάθημα, αν έχει απορριφθεί και ο μαθητής δεν μπορεί να παρακολουθήσει το μάθημα καθώς επίσης και αν ο δάσκαλος δεν έχει απαντήσει ακόμα στην αίτηση.

`course_id`: περιέχει το `id` του μαθήματος.

Συσχετίσεις του model notification με τα άλλα models:



teacher

Σε αυτό το model υπάρχουν όλα τα πεδία που πρέπει να συμπληρωθούν ώστε κάποιος χρήστης να μπορέσει να διδάξει κάποιο μάθημα.

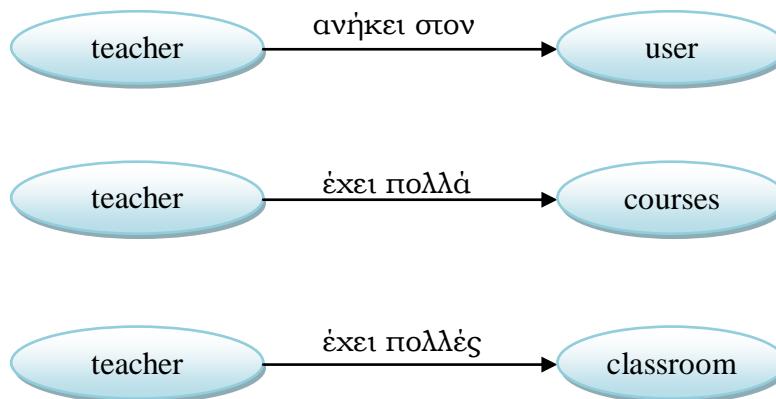
Το model teacher έχει τα πεδία: `higher_education_level`, `languages_spoken`, `years_of_experience`, `about`, `user_id`, `teacher_photo`.

`higher_education_level`: σε αυτό το πεδίο περιέχεται η ανώτερη μόρφωση του δάσκαλου. Ο δάσκαλος διαλέγει την ανώτερη μόρ από ένα σύνολο επιλογών που είναι διαθέσιμα.

`languages_spoken`: σε αυτό το πεδίο αποθηκεύονται οι γλώσσες που μιλάει ο δάσκαλος

`years_of_experience`: σε αυτό το πεδίο περιέχονται τα χρόνια εμπειρίας του δασκάλου.
`about`: σε αυτό το πεδίο ο δάσκαλος μπορεί να αποθηκεύσει όποια προσωπική του πληροφορία επιθυμεί.
`user_id`: σε αυτή τη μεταβλητή αποθηκεύεται το `id` που έχει ο δάσκαλος ως χρήστης
`teacher_photo`: περιέχει το avatar του δασκάλου.

Συσχετίσεις του model `teacher` με τα άλλα models:



course_day

Σε αυτό το model περιέχονται όλα τα πεδία που είναι απαραίτητα για να δημιουργηθεί η μέρα και η ώρα που θα γίνει το μάθημα.

Το model `course_day` περιέχει τα πεδία: `day`, `start_at`, `ends_at` και `course_id`.

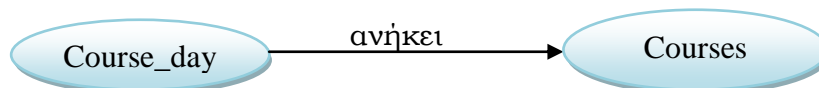
`day`: σε αυτό το πεδίο αποθηκεύεται η μέρα διεξαγωγής του μαθήματος.

`start_at`: σε αυτό το πεδίο αποθηκεύεται η ώρα που θα ξεκινήσει το μάθημα

`ends_at`: σε αυτό το πεδίο αποθηκεύεται η ώρα που θα τελειώσει το μάθημα.

`course_id`: σε αυτό το πεδίο αποθηκεύεται το `id` του μαθήματος.

Συσχετίσεις του `course_day` με τα άλλα models:

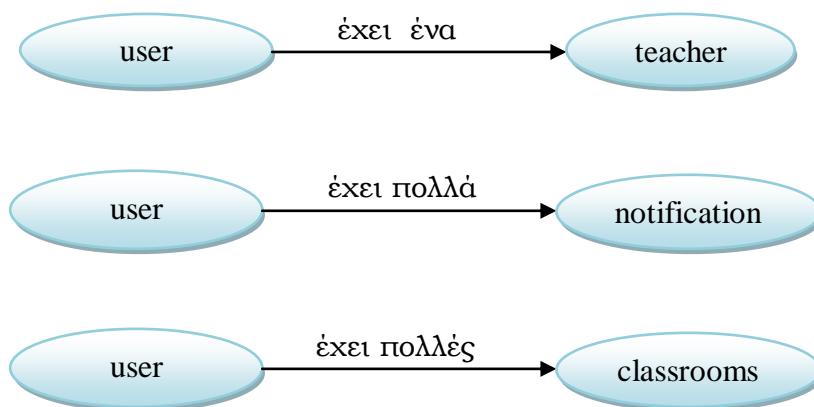


user

Σε αυτό το model περιέχονται όλα τα πεδία που μπορεί να συμπληρώσει

ο χρήστης στο προφίλ του μετά την εισαγωγή του στο σύστημα.
Το model user περιέχει τα πεδία email, password, password_confirmation, first_name, last_name, gender, country, city, age, photo_filename.
email: σε αυτό το πεδίο αποθηκεύεται το email του χρήστη κατά την εγγραφή του στη πλατφόρμα.
password: σε αυτό το πεδίο αποθηκεύεται το password του χρήστη κατά την εγγραφή του στη πλατφόρμα.
password_confirmation: σε αυτό το πεδίο αποθηκεύεται το password του χρήστη για επιβεβαίωση κατά την εγγραφή του στη πλατφόρμα.
first_name: σε αυτό το πεδίο αποθηκεύεται το όνομα χρήστη κατά την εγγραφή του στην πλατφόρμα.
last_name: σε αυτό το πεδίο αποθηκεύεται το επίθετο του χρήστη κατά την εγγραφή στην πλατφόρμα.
gender: στην μεταβλητή αυτή περιέχεται προαιρετικά το φύλο του χρήστη, ο οποίος το συμπληρώνει μετά την εισαγωγή του σύστημα και θέλοντας να επεξεργαστεί το προφίλ του.
country: στην μεταβλητή αυτή περιέχεται προαιρετικά η χώρα του χρήστη, ο οποίος το συμπληρώνει μετά την εισαγωγή του σύστημα και θέλοντας να επεξεργαστεί το προφίλ του.
city: στην μεταβλητή αυτή περιέχεται προαιρετικά η πόλη του χρήστη, ο οποίος το συμπληρώνει μετά την εισαγωγή του σύστημα και θέλοντας να επεξεργαστεί το προφίλ του.
age: στην μεταβλητή αυτή περιέχεται προαιρετικά η ηλικία του χρήστη, ο οποίος το συμπληρώνει μετά την εισαγωγή του σύστημα και θέλοντας να επεξεργαστεί το προφίλ του.
photo_filename: στην μεταβλητή αυτή περιέχεται προαιρετικά το avatar του χρήστη, ο οποίος το συμπληρώνει μετά την εισαγωγή του σύστημα και θέλοντας να επεξεργαστεί το προφίλ του.

Συσχετίσεις του model user με τα άλλα models:



➤ **Views**

Στο directory `app/views` της εφαρμογής υπάρχουν subdirectories για καθέναν από τους controllers που έχουμε δημιουργήσει. Κάθε μια από αυτές τις υποκατηγορίες έχει το ίδιο όνομα με τις συναρτήσεις του αντίστοιχου controller.

Όλα τα αρχεία που δημιουργήθηκαν έχουν σκοπό να οπτικοποιήσουν την λειτουργία της αντίστοιχης συνάρτησης στον controller.

Επομένως οι φάκελοι που έχουν δημιουργηθεί είναι:

courses

Αντίστοιχα με τις συναρτήσεις που δημιουργήθηκαν στον `course_controller`, υπάρχουν και τα κατάλληλα `.html.erb` αρχεία. Στην συνέχεια αναφέρονται τα αρχεία καθώς και μια σύντομη περιγραφή για την λειτουργία τους.

new: δημιουργία του μαθήματος

index: πάνελ των μαθημάτων, δημιουργία ημερολογίου και εμφάνιση των μαθητών της τάξης

show: δημόσιο προφίλ μαθήματος

edit: update του μαθήματος

streaming: διαδικτυακή τάξη

home

Αντίστοιχα με τις συναρτήσεις που δημιουργήθηκαν στον `home_controller`, υπάρχουν και τα κατάλληλα `.html.erb` αρχεία. Στην συνέχεια αναφέρονται τα αρχεία καθώς και μια σύντομη περιγραφή για την λειτουργία τους.

index: δημιουργία της αρχικής σελίδας

layouts

Αυτός ο φάκελος δημιουργείται από την rails και αντιστοιχεί στον `application_controller`. Καθορίζει το αρχικό προσχέδιο για όλες τις σελίδες(template).

application: αρχείο που κληρονομούν όλες οι σελίδες, για παράδειγμα τα links “dashboard-profile-teacher”.

public_layout: αρχείο για το δημόσιο προφίλ του δασκάλου, χρήστη και μαθήματος. Δεν υπάρχουν δηλαδή οι επιλογές dashboard-profile-teacher.

profiles

Αντίστοιχα με τις συναρτήσεις που δημιουργήθηκαν στον

`profile_controller`, υπάρχουν και τα κατάλληλα `.html.erb` αρχεία. Στην συνέχεια αναφέρονται τα αρχεία καθώς και μια σύντομη περιγραφή για την λειτουργία τους.

edit: επεξεργασία προφίλ χρήστη

edit_teacher: επεξεργασία προφίλ δασκάλου

public_profile: το προφίλ του χρήστη όπως το βλέπουν οι άλλοι χρήστες.

show: εμφάνιση του προφίλ του χρήστη

avatar_holder: εμφάνιση της φωτογραφίας του χρήστη ή δασκάλου αν έχει.

user

Αντίστοιχα με τις συναρτήσεις που δημιουργήθηκαν στον `user_controller`, υπάρχουν και τα κατάλληλα `.html.erb` αρχεία. Στην συνέχεια αναφέρονται τα αρχεία καθώς και μια σύντομη περιγραφή για την λειτουργία τους.

be_teacher: εμφάνιση του προφίλ του δασκάλου.

dashboard: το dashboard του χρήστη.

2.5 Επιμέρους Εργαλεία της πλατφόρμας

Για τις ανάγκες της πλατφόρμας είναι αναγκαία η ενσωμάτωση εργαλείων εκπαίδευσης. Παρακάτω παρατίθεται αναλυτική περιγραφή αυτής της διαδικασίας για καθένα από τα εργαλεία εκπαίδευσης που έχουν αναφερθεί.

Live-streaming

Το live-streaming αποτελεί ένα third part API το οποίο μπορεί να βρει κάποιος online. Συγκεκριμένα στην πλατφόρμα το third part API που χρησιμοποιήθηκε είναι το `tokbox[25]`. Όπως με κάθε API έτσι και με το `tokbox` για να μπορέσει κάποιος να τεστάρει και να ενσωματώσει το `tokbox` στο περιβάλλον ανάπτυξης (development), πρέπει να είναι κάτοχος των κλειδιών API key, API signature. Δημιουργώντας ένα λογαριασμό στο επίσημο site του `tokbox` σαν προγραμματιστής μπορεί να γίνει κάποιος κάτοχος τέτοιων κλειδιών. Έχοντας στην διάθεση του τα κλειδιά μπορεί να ενσωματώσει το interface του API με την εφαρμογή που αναπτύσσει.

Η ενσωμάτωση ποικίλει. Μπορεί κάποιος να ενσωματώσει το συγκεκριμένο API χρησιμοποιώντας javascript, Ruby, python, Java και php.

Στη συγκεκριμένη περίπτωση επειδή η εφαρμογή αναπτύχθηκε με την χρήση του framework Rails, κάνοντας χρήση της γλώσσας Ruby επιλέχθηκε σαν μέσω επικοινωνίας του API με την εφαρμογή η συγκεκριμένη γλώσσα. Για τον σκοπό

χρησιμοποιήθηκε το gem openTok το οποίο έχει προκαθορισμένες μεθόδους για το authorization και το session management.

Επειδή το live-streaming παίρνει μέρος στο front-end της εφαρμογής χρησιμοποιήθηκε η javascript σαν μέσο ανάπτυξης και ενσωμάτωσης της επιθυμητής λειτουργικότητας.

Συνοψίζοντας, το session management και το authorization παίρνουν μέρος στο back-end της εφαρμογής ενώ το session registration των requests που γίνονται από χρήστες παίρνει μέρος στο front-end.

Upload αρχεία

Στο αρχικό πλάνο σχεδίασης των απαιτήσεων της εφαρμογής υπήρχε η λειτουργικότητα υποστήριξης ανεβάσματος αρχείων. Για την υλοποίηση αυτής της λειτουργικότητας χρησιμοποιήθηκε η προεγκατεστημένη βιβλιοθήκη ανεβάσματος αρχείων που χρησιμοποιεί το framework Rails. Επειδή στο αρχικό πλάνο σχεδίασης θέλαμε το ανέβασμα των αρχείων να γίνεται χωρίς ανανέωση των σελίδων η προεγκατεστημένη βιβλιοθήκη ανεβάσματος αρχείων δεν μας βοήθησε σε αυτό τον τομέα. Για το λόγο αυτό εγκαταστάθηκε το gem remotipart το οποίο επέκτεινε τις δυνατότητες της προεγκατεστημένης βιβλιοθήκης upload αρχείων, να υποστήριζε ajax request. Όταν ο χρήστης επιλέξει το αρχείο που θέλει να κάνει upload από το front-end της εφαρμογής, ο server επεξεργάζεται την ajax αίτηση, ελέγχοντας κατά πόσο το αρχείο είναι ασφαλές (αποτρέποντας εκτελέσιμα .exe, .dmg) και στέλνει στο front-end της εφαρμογής ένα javascript response. Αυτό είναι είτε μήνυμα λάθους σε περίπτωση που το αρχείου είναι λάθος είτε κενό ή ενσωματώνει στην ήδη υπάρχουσα λίστα των uploaded αρχείων το αρχείο που επιλέχτηκε.

Ημερολόγιο

Απαραίτητο εργαλείο για να μπορέσει ο δάσκαλος να οργανώσει και να προγραμματίσει τις μέρες, ώρες και διάρκεια των μαθημάτων του ήταν η χρήση ημερολογίου.

Η χρήση ημερολογίου κατά μέγιστο βαθμό στο front-end και μόνο ένα μικρό κομμάτι της επικοινωνεί με το back-end.

Στο front-end κομμάτι υλοποιήθηκε ένας πίνακας html και τα δεδομένα των ημερών και των μηνών ενός ολοκληρωμένου ημερολογίου έγιναν populate σε αυτόν, με την χρήση του gem watu_table_builder. Για να μπορέσει η αλληλεπίδραση με τον χρήστη να είναι πιο ρεαλιστική και ευχάριστη προστέθηκαν javascript mouse events σε κάθε κελί του πίνακα το οποίο περιέχει ημερομηνία. Σε κάθε ένα από αυτά τα κελιά επίσης, προστέθηκε ένα κρυφό dialog window το οποίο περιέχει την φόρμα η οποία κάνει update το status της συγκεκριμένης ημέρας για το συγκεκριμένο μάθημα. Η διαχείριση των mouse events και η απόκρυψη/εμφάνιση των dialog γίνεται την χρήση της βιβλιοθήκης jquery.

Η ροή για το update της μέρας που επιλέγει ο καθηγητής είναι η εξής: ο καθηγητής κάνει κλικ το κελί με την ημερομηνία που τον ενδιαφέρει, το αντίστοιχο dialog που εμπεριέχει την φόρμα για την συγκεκριμένη ημερομηνία

εμφανίζεται, ο καθηγητής επιλέγει την ώρα έναρξης και την ώρα λήξης και στέλνει τα δεδομένα στον server. Ο server επεξεργάζεται την αίτηση και απαντά με μήνυμα λάθους αν η αίτηση ήταν μη αποδεκτή ή κάνει update το status της συγκεκριμένης ημερομηνίας για το συγκεκριμένο μάθημα στην βάση δεδομένων και απαντά με μήνυμα επιτυχίας.

Σε αυτό το σημείο καλό είναι να τονιστεί ότι κάθε μήνυμα αποτελεί μία εγγραφή στο πίνακα μαθημάτων στη βάση δεδομένων το οποίο έχει συσχέτιση 1-N με τον πίνακα μέρες μαθήματος.

Με βάση τα προαναφερθέντα γίνεται κατανοητό ότι όταν ο server λάβει αποδεκτό update για μια συγκεκριμένη ημερομηνία ενός συγκεκριμένου μαθήματος εγγράφει στον πίνακα μέρες μαθήματος την συγκεκριμένη ημερομηνία ώρα έναρξης, λήξης και το id δηλαδή το αναγνωριστικό του μαθήματος που αυτή συσχετίζεται.

3. Χαρακτηριστικά της Πλατφόρμας

3.1 Συνολική Περιγραφή

3.1.1 Προοπτική της Πλατφόρμας

Η πλατφόρμα ασύγχρονης εκπαίδευσης δίνει την δυνατότητα στο χρήστη να επιλέξει ανάμεσα σε δύο ρόλους αυτού του δασκάλου ή αυτού του μαθητή αλλά του δίνεται η δυνατότητα να αποκτήσει και τους δύο ρόλους δημιουργώντας το δικό του μάθημα και παρακολουθώντας οποιοδήποτε άλλο θέλει.

Πιο συγκεκριμένα για αυτές τις δύο ομάδες χρηστών:

Μαθητής

Ο χρήστης μετά την εγγραφή του στην πλατφόρμα εμφανίζεται το dashboard όπου έχει τρεις επιλογές:

- Την εμφάνιση του προφίλ του, όπου εκεί δίνεται η δυνατότητα στον χρήστη να επεξεργαστεί το προφίλ, συμπληρώνοντας πρόσθετες πληροφορίες που θέλει.
- Την αναζήτηση μαθημάτων, όπου δίνεται η δυνατότητα στον χρήστη να αναζητήσει για το μάθημα που τον ενδιαφέρει και να του εμφανιστούν τα σχετικά αποτελέσματα. Στην συνέχεια ο μαθητής θα στέλνει μια αίτηση αποδοχής στον δάσκαλο για να δώσει την δυνατότητα αυτός να παρακολουθήσει το μάθημα του.
- Την δυνατότητα να γίνει δάσκαλος, όπου δίνει την δυνατότητα στον χρήστη να δημιουργήσει το δικό του μάθημα.

Δάσκαλος

Ο χρήστης μετά την επιλογή του να γίνει δάσκαλος (που μπορεί να το επιλέξει από το dashboard), έχει δύο επιλογές:

- Να επεξεργαστεί το προφίλ του δασκάλου, ώστε να μπορέσει να δημιουργήσει κάποιο μάθημα, αλλά του δίνεται και η δυνατότητα να διαγράψει το προφίλ.
- Να επεξεργαστεί το προφίλ του κάθε μαθήματος, όπου του δίνεται η δυνατότητα να δημιουργήσει ένα μάθημα και να επεξεργαστεί τα στοιχεία του, όπου μεταξύ άλλων ορίζει τις ημερομηνίες και τις ώρες που θα είναι διαθέσιμο το μάθημα του. Επιπλέον ο δάσκαλος κάνει δεκτές ή απορρίπτει αιτήσεις μαθητών για το μάθημα του. αλλά και διαγράφει το

μάθημα. Τέλος ο δάσκαλος είναι αυτός που θα ξεκινήσει το μάθημα και στην συνέχεια θα έχουν πρόσβαση και οι μαθητές στην τάξη.

Από τα παραπάνω φαίνεται ότι τα οφέλη για τον χρήστη είναι:

- Για τον μαθητή να μάθει κάτι που θέλει με άμεσα και εύκολα από τον χώρο του
- Για τον δάσκαλο να αποκτήσει εμπειρία στην διδασκαλία και στην οργάνωση ενός μαθήματος

3.1.2 Χαρακτηριστικά Προϊόντος

Το προϊόν έχει σκοπό την ασύγχρονη εκπαίδευση μεταξύ ενός δασκάλου και των μαθητών του. Το προϊόν πρέπει να διαθέτει κάποια χαρακτηριστικά για σκοπό που θέλει να πετύχει.

Παρακάτω παρουσιάζονται κάποιες λειτουργίες που έχουν ενσωματωθεί στην πλατφόρμα:

- Video-streaming
- Δυνατότητα αποστολής μηνύματος σε όλη την τάξη
- Δυνατότητα να κάνει ο καθηγητής και ο μαθητής upload αρχεία
- Ύπαρξη ημερολογίου για τις μέρες και ώρες του κάθε μαθήματος

Λειτουργίες που θα μπορούσαν να ενσωματωθούν μελλοντικά:

- Whiteboard
- Αξιολόγηση των μαθησιακών δραστηριοτήτων και υλικών που χρησιμοποιήθηκαν
- Audio και video μηνύματα
- Online quiz

Όσον αφορά την ομαλή λειτουργία της πλατφόρμας υπάρχει σελίδα για τον administrator όπου μπορεί να διαγράψει και να επεξεργαστεί τα μαθήματα και τους users.

3.1.3 Κατηγορίες χρηστών και τα χαρακτηριστικά τους

Μαθητής:

Αυτός ο τύπος χρήστη έχει σκοπό να κάνει αναζήτηση για κάποιο μάθημα που τον ενδιαφέρει , να επιλέξει το μάθημα και αν γίνει αποδεκτός από τον δάσκαλο του μαθήματος να το παρακολουθήσει.

Δάσκαλος:

Μπορεί οποιοσδήποτε χρήστης να δημιουργήσει το δικό του μάθημα σε όποιο γνωστικό αντικείμενο επιθυμεί. Δεν υπάρχουν περιορισμού σχετικά με το ποιος μπορεί να γίνει δάσκαλος. Οργανώνει αυτός το μάθημα του όπως θέλει και είναι υπεύθυνος για την διαδικτυακή τάξη που έχει δημιουργήσει.

Administrator:

Είναι υπεύθυνος για την ομαλή λειτουργία της πλατφόρμας. Οι υπόλοιποι χρήστες της πλατφόρμας αν θεωρούν ότι κάποιος χρήστης τους προσβάλει ή ότι παρεμποδίζει την ομαλή λειτουργία της πλατφόρμας μπορεί να απευθυνθεί στον administrator ο οποίος πρέπει να λύσει όποιο πρόβλημα παρουσιαστεί.

Η πλατφόρμα προσφέρει τις παρακάτω λειτουργικότητες στους τρεις τύπους χρηστών:

Μαθητής:

1. *Να αναζητήσει κάποιο μάθημα που τον ενδιαφέρει*
 - Κάνει αναζήτηση σε κάποιο γνωστικό αντικείμενο που τον ενδιαφέρει
 - Επιλέγει το μάθημα που θέλει να παρακολουθήσει και στέλνει αίτηση παρακολούθησης στον δάσκαλο
 - Περιμένει να του σταλεί βεβαίωση αποδοχής από τον δάσκαλο ώστε να μπορεί να παρακολουθήσει το μάθημα

2. *Να παρακολουθήσει το μάθημα που έχει επιλέξει*
 - Παρακολουθεί το μάθημα, την ώρα και μέρα που έχει ορίσει ο δάσκαλος, με την βοήθεια εργαλείων εκπαίδευσης

Δάσκαλος:

3. *Να δημιουργήσει το δικό του μάθημα*
 - Έχει την δυνατότητα να δημιουργήσει, να επεξεργαστεί και να διαγράψει κάποιο μάθημα

4. *Να διδάξει στους μαθητές που έχουν επιλέξει το μάθημα του*
 - Διδάσκει το μάθημα του με τα εργαλεία εκπαίδευσης που έχει επιλέξει

Administrator

5. *Να μπορεί να έχει πρόσβαση στη βάση δεδομένων*

- Μπορεί να δει/διαγράψει/τροποποιήσει κάποια από τις εικονικές τάξεις, κάποιον από τους δασκάλους ή από κάποιον χρήστη.

3.2 Λειτουργίες Πλατφόρμας

Οι λειτουργικές απαιτήσεις οργανώνονται από τον χρήστη. Οι δύο τύποι χρηστών δάσκαλος και μαθητής μπορούν να εκτελέσουν συγκεκριμένες λειτουργίες με την εφαρμογή. Οι πράξεις αυτές ονομάζονται συνήθως περιπτώσεις χρήσης και περιγράφονται παρακάτω: Για κάθε περίπτωση χρήσης παρατίθεται μια συνοπτική περιγραφή, ακολουθία ενέργειας/ανταπόκρισης και λειτουργικές απαιτήσεις

3.2.1 Μαθητής:: Επεξεργάζεται το προφίλ του

Περιγραφή

Ο χρήστης θέλει να επεξεργαστεί το προφίλ του μετά την είσοδο του στο σύστημα.

Ενέργεια/Ανταπόκριση

1. Ο χρήστης κάνει εγγραφή ή login στην πλατφόρμα
2. Μετά την εισαγωγή στο σύστημα, εμφανίζεται στον χρήστη το dashboard όπου στο πάνω μέρος αυτής της σελίδας, αλλά και σε κάθε άλλη, βρίσκονται οι επιλογές:
 - Dashboard
 - Profile
 - Teacher
3. Ο χρήστης επιλέγοντας το profile εμφανίζονται τα πεδία First Name και Last Name που συμπληρώθηκαν όταν ο χρήστης έκανε Sign up. Επίσης εμφανίζεται η επιλογή Edit Profile

4. Επιλέγοντας το Edit Profile εμφανίζεται μια λίστα από πεδία που μπορεί να συμπληρώσει ο χρήστης. Αφού συμπληρώσει τα στοιχεία που θέλει ο χρήστης, πατάει το κουμπί Update και τότε το προφίλ του χρήστη ενημερώνεται
5. Ο χρήστης μπορεί να δει το ενημερωμένο προφίλ του πατώντας το link: Profile που βρίσκεται στο πάνω μέρος της σελίδας.

3.2.2 Μαθητής:: Παρακολουθεί ένα μάθημα

Περιγραφή

Ο χρήστης θέλει να παρακολουθήσει ένα μάθημα σε κάποιο γνωστικό αντικείμενο που τον ενδιαφέρει.

Ενέργεια/Ανταπόκριση

1. Ο χρήστης κάνει εγγραφή ή login στην πλατφόρμα
2. Μετά την εισαγωγή στο σύστημα, εμφανίζεται στον χρήστη το dashboard όπου εκεί υπάρχουν οι επιλογές να πάει στο προφίλ του και να το επεξεργαστεί, να γίνει δάσκαλος, να ξαναγυρίσει στο Dashboard και να κάνει αναζήτηση για κάποιο μάθημα που τον ενδιαφέρει. Εκεί ο χρήστης-μαθητής :
 - κάνει αναζήτηση για το μάθημα που τον ενδιαφέρει
 - του εμφανίζονται όλα τα μαθήματα που είναι διαθέσιμα σύμφωνα με την αναζήτηση που έκανε.
 - Επιλέγει το μάθημα που τον ενδιαφέρει και στέλνει αίτηση παρακολούθησης του μαθήματος στον δάσκαλο που έχει δημιουργήσει το μάθημα
3. Όταν κάνει αποδεκτή ή απορριφθεί η αίτηση από τον καθηγητή εμφανίζεται κατάλληλο μήνυμα στο dashboard του χρήστη
4. Επίσης σε αυτή την περίπτωση εμφανίζεται στο dashboard ένα link που τον μεταφέρει στην διαδικτυακή τάξη τις ώρες που λαμβάνει χώρα το μάθημα
5. Τέλος ο μαθητής θα λαμβάνει πιθανές ανακοινώσεις από τον δάσκαλο του μαθήματος που έχει επιλέξει να παρακολουθήσει, οι οποίες θα εμφανίζονται στο dashboard
6. Ο μαθητής μπορεί να ακολουθήσει την διαδικασία αναζήτησης και εγγραφής σε όποιο μάθημα επιθυμεί

3.2.3 Δάσκαλος:: Δημιουργεί προφίλ δασκάλου

Περιγραφή

Ο δάσκαλος θέλει να δημιουργήσει το προφίλ του για να μπορέσει να δημιουργήσει κάποιο μάθημα

Ενέργεια/Ανταπόκριση

1. Ο χρήστης επιλέγει το πεδίο Teacher, στο πάνω μέρος όποιας σελίδας βρίσκεται
2. Στον χρήστη εμφανίζεται το μήνυμα: You don't have a teacher profile yet, create one now.
3. Πατώντας στο create, εμφανίζεται στο χρήστη μια φόρμα με πεδία που πρέπει να συμπληρώσει
4. Όταν ο χρήστης πατήσει Update για να δημιουργηθεί το προφίλ του, τα στοιχεία που έχει συμπληρώσει εισάγονται στο προφίλ του δασκάλου
5. Μετά την δημιουργία του προφίλ δασκάλου, εμφανίζονται κάτω από το προφίλ του δύο επιλογές: Edit Teacher profile και Delete teacher profile όπου μπορεί αντίστοιχα να επεξεργαστεί και να διαγράψει το προφίλ του

3.2.4 Δάσκαλος:: Δημιουργεί ένα μάθημα

Περιγραφή

Ο δάσκαλος θέλει να δημιουργήσει ένα μάθημα που θέλει να διδάξει.

Ενέργεια/Ανταπόκριση

1. Αφού ο δάσκαλος έχει δημιουργήσει το προφίλ του τότε του εμφανίζεται η επιλογή Courses για να μπορέσει να δημιουργήσει κάποιο μάθημα
2. Πατώντας στο link Course εμφανίζεται το μήνυμα "You don't have any courses yet" και υπάρχει η επιλογή : Create a Course
3. Πατώντας το link Create a Course ο χρήστης μεταβαίνει σε μια φόρμα συμπλήρωσης στοιχείων του μαθήματος. Αφού συμπληρώσει αυτή την φόρμα πατάει Create και το μάθημα δημιουργείται
4. Εμφανίζονται το στοιχεία του μαθήματος στο Course Panel δίνοντας την δυνατότητα στο χρήστη να κάνει Edit και Delete το συγκεκριμένο μάθημα
5. Ο χρήστης πατώντας την επιλογή Create a Course μπορεί να δημιουργήσει όποιο άλλο μάθημα επιθυμεί

3.2.5 Δάσκαλος:: Δημιουργεί την τάξη

Περιγραφή

Ο δάσκαλος επιλέγει τους μαθητές που θα παρακολουθήσουν το μάθημα του κάνοντας αποδεκτές τις αιτήσεις για το μάθημα του

Ενέργεια/Ανταπόκριση

1. Στο προφίλ του καθηγητή εμφανίζονται οι αιτήσεις δηλαδή το όνομα και επώνυμο του μαθητή αλλά δίνεται και η δυνατότητα στον δάσκαλο να δει το προφίλ του μαθητή
2. Στην συνέχεια ο δάσκαλος δέχεται ή απορρίπτει την αίτηση του μαθητή
3. Τα ονόματα των μαθητών που γίνονται αποδεκτοί από τον δάσκαλο εμφανίζονται στο πάνελ των μαθημάτων στο προφίλ των μαθημάτων

3.2.6 Δάσκαλος:: Διδάσκει το μάθημα του

Περιγραφή

Ο δάσκαλος διδάσκει το μάθημα που έχει δημιουργήσει χρησιμοποιώντας τα εργαλεία εκπαίδευσης που παρέχονται

Ενέργεια/Ανταπόκριση

1. Ο δάσκαλος διδάσκει το μάθημα που έχει δημιουργήσει .
Την ώρα έναρξης του μαθήματος πατάει το αντίστοιχο link για να τον μεταφέρει στην διαδικτυακή τάξη
2. Στην συνέχεια εισέρχονται οι μαθητές που έχει επιλέξει να παρακολουθήσουν το μάθημα στο σύστημα
3. Ο δάσκαλος ξεκινάει το μάθημα
4. Ο δάσκαλος είναι αυτός που θα τερματίσει το μάθημα

3.3 Interface

Τα κύρια χαρακτηριστικά του interface (διεπαφή) της πλατφόρμας είναι η απλότητα, η χρηστικότητα και η καλαισθησία.

Για την δημιουργία του design της πλατφόρμας χρησιμοποιήθηκε σαν βάση το Twitter Bootstrap[19]. Το Twitter Bootstrap είναι ένα εργαλείο για να επιταχύνει την ανάπτυξη του front end της εφαρμογής. Περιλαμβάνει βασικό κώδικα CSS και HTML για πίνακες, φόρμες, κουμπιά, alerts και πολλά άλλα.

Στον υπάρχοντα κώδικα έχουν γίνει αρκετές αλλαγές για την προσαρμογή του Twitter Bootstrap στην πλατφόρμα ασύγχρονης εκπαίδευσης.

Για την ενσωμάτωση του στην πλατφόρμα απλά χρειάζεται η εγκατάσταση του κατάλληλου Gem. Πληροφορίες για την εγκατάσταση και τα βήματα που χρειάζεται να ακολουθηθούν βρίσκονται στο github [20].

Στην πλατφόρμα εμφανίζονται διάφορα πάνελ που εξυπηρετούν τις λειτουργίες που αναφέρθηκαν στην ενότητα 2.2. Κάποια ενδεικτικά παρουσιάζονται παρακάτω:




Εικόνα 1: Αρχική σελίδα της πλατφόρμας

Teacher added

Learn Together

Dashboard Profile Teacher Find a Course [Logout](#)



Teacher

[Create a Course](#)

[Edit Profile](#)

[Courses](#)

Skills & Professional Details

Higher education level:	Bachelor
Languages spoken:	en
Years of experience:	1
About:	Just a teacher

Personal Details


First Name:	Ioanna
Last Name:	K
Gender:	Female
Age:	25
City:	
Country:	Greece

Εικόνα 2: Προφίλ δασκάλου

User successfully updated

Learn Together

Dashboard Profile Teacher Find a Course [Logout](#)



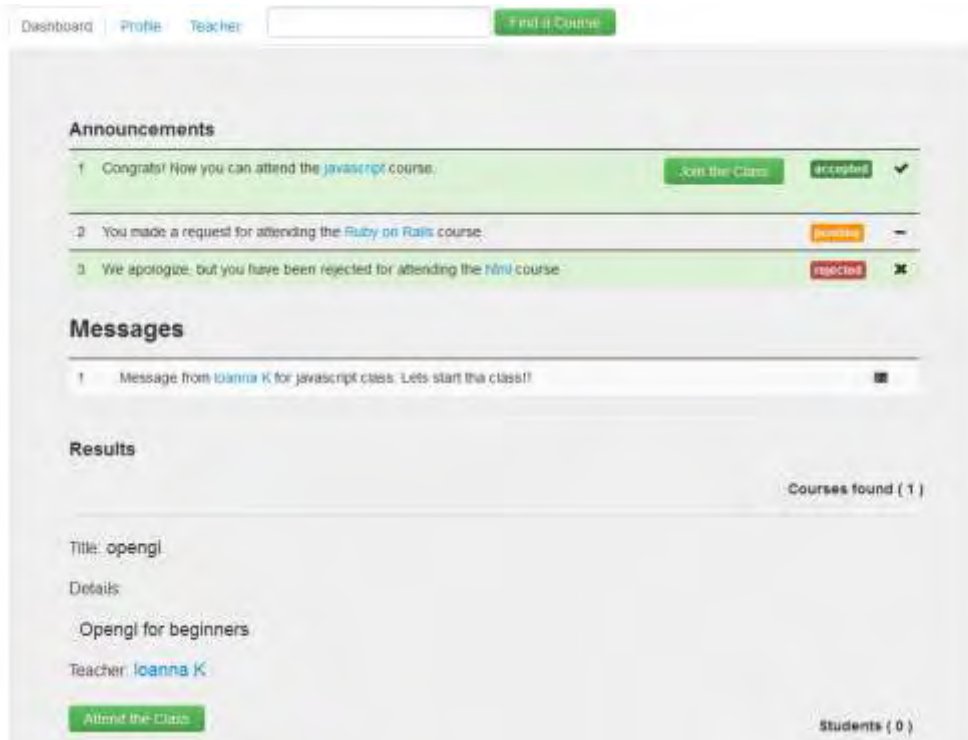
Student

[Edit Profile](#)

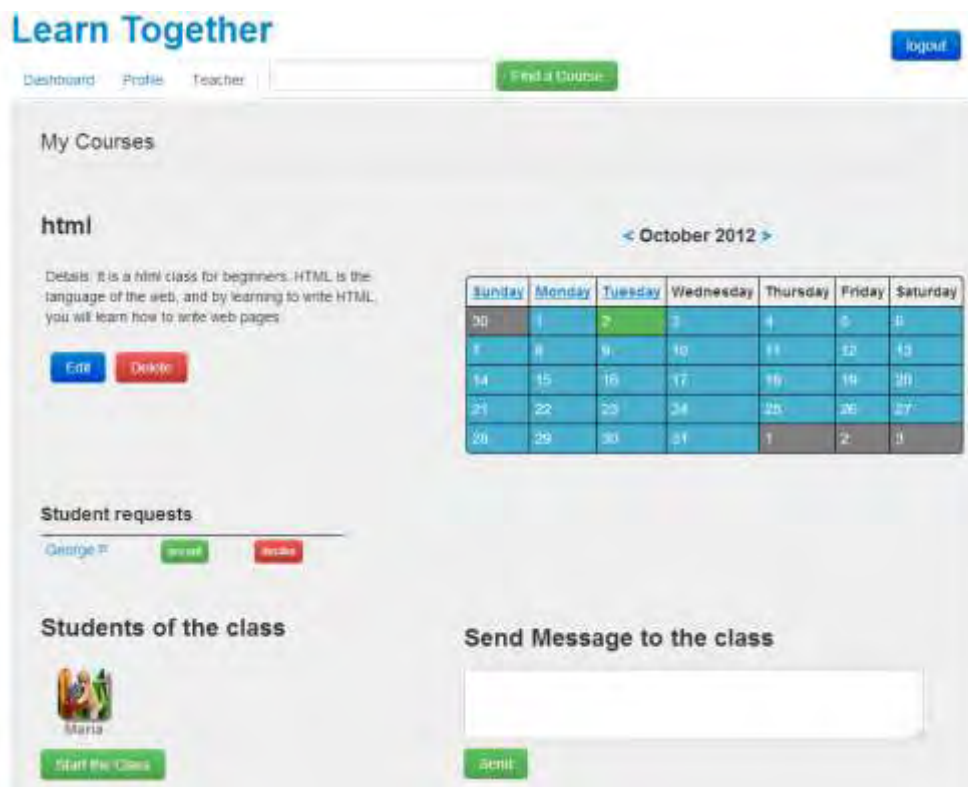
Profile Details

First Name:	Maria
Last Name:	G
Gender:	Choose
Age:	23
City:	Ypsos
Country:	Greece

Εικόνα 3: Προφίλ μαθητή



Εικόνα 4: Dashboard χρήστη



Εικόνα 5: πάνελ μαθημάτων δασκάλου



Εικόνα 6: Διαδικτυακή τάξη

4. Η πλατφόρμα Online

4.1 Διεύθυνση πλατφόρμας

Η πλατφόρμα, μετά την υλοποίηση της βρίσκεται online στην διεύθυνση:

<http://stark-warrior-8303.herokuapp.com/>

4.2 Heroku

Η cloud πλατφόρμα[21] που χρησιμοποιήθηκε και για να ανεβεί online η ασύγχρονη πλατφόρμα εκπαίδευσης είναι το heroku[22].

Το heroku είναι μια cloud πλατφόρμα(cloud platform as a service[23]) που υποστηρίζει αρκετές γλώσσες προγραμματισμού. Είναι μια από τις πρώτες cloud πλατφόρμες που δημιουργήθηκε το 2007 και υποστηρίζει την γλώσσα προγραμματισμού Ruby.

Για να κάνει κανείς deploy στο heroku υπάρχουν οδηγίες στο επίσημο site του heroku[24].

5. Μελλοντικές Επεκτάσεις

Η πλατφόρμα βρίσκεται σε αρχικό στάδιο που σημαίνει ότι δίνεται η δυνατότητα για περαιτέρω επέκταση και ανάπτυξη.

Παρακάτω παρουσιάζονται κάποιες από τις μελλοντικές επεκτάσεις της πλατφόρμας:

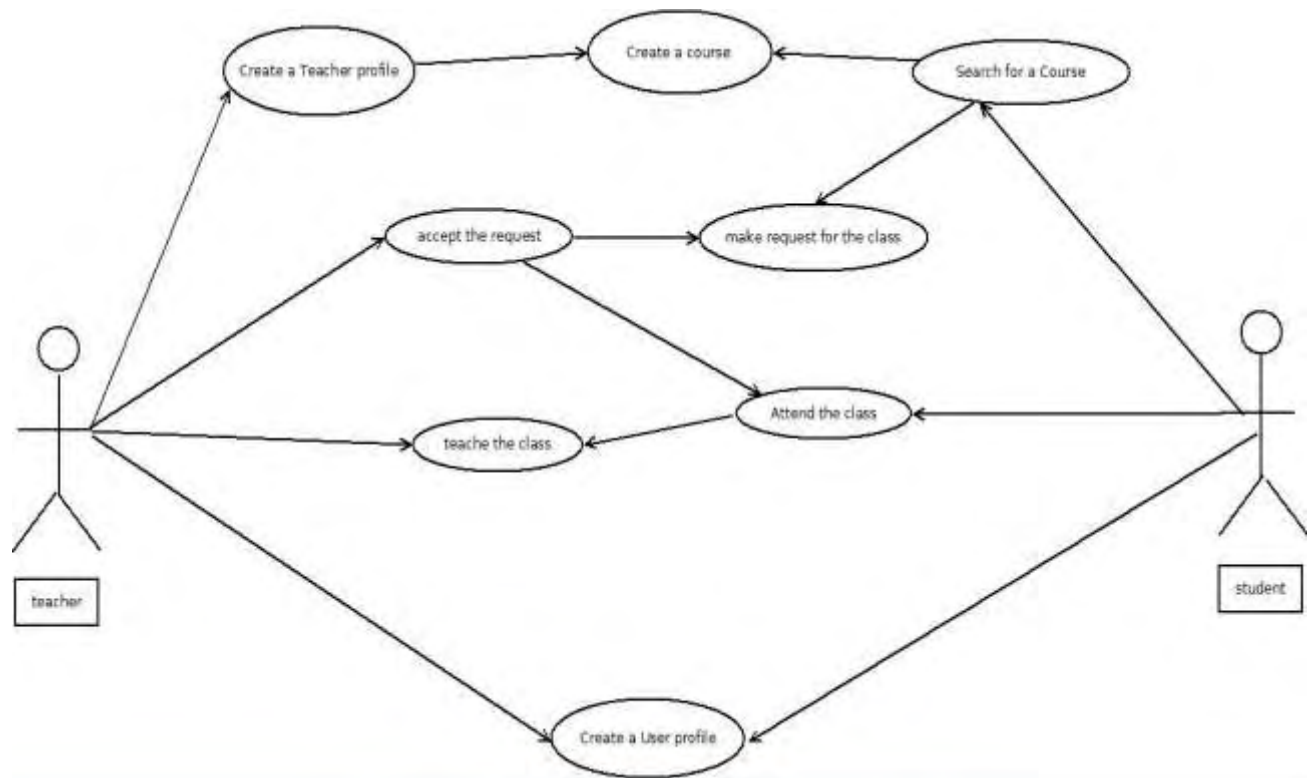
- *Δυνατότητα αξιολόγησης του δασκάλου από τους μαθητές του.*
Μετά την ολοκλήρωση των μαθημάτων θα δίνεται η δυνατότητα στους μαθητές να αξιολογήσουν το δάσκαλο καθώς και να σχολιάσουν για το πόσο ικανοποιημένοι ή όχι είναι από το μάθημα που διδάχτηκαν. Έτσι θα δημιουργηθεί ένα πιο ολοκληρωμένο προφίλ για τον δάσκαλο και οι χρήστες σε μελλοντικές αναζητήσεις θα μπορούν να βλέπουν την αξιολόγηση του δασκάλου.
- *Ενσωμάτωση εργαλείων εκπαίδευσης στην υπάρχουσα διαδικτυακή τάξη.*
Η διαδικτυακή τάξη θα εμπλουτιστεί με εργαλεία εκπαίδευσης ώστε να προσφέρονται περισσότερες δυνατότητες στον δάσκαλο για να διδάξει το μάθημα του. Κάποια από αυτά τα εργαλεία εκπαίδευσης θα μπορούσαν να είναι:
 - online quiz για να εξετάζει ο δάσκαλος τους μαθητές
 - Δυνατότητα να κάνει upload ο καθηγητής Audio και video μηνύματα.
 - Whiteboard. Δηλαδή η δυνατότητα να βλέπει ο μαθητής αυτά που γράφει ο δάσκαλος.
 - Υποστήριξη share screen
 - Chat στην διαδικτυακή τάξηΕπίσης θα μπορούσαν να αναπτυχθούν και να επεκταθούν τα ήδη υπάρχοντα εργαλεία εκπαίδευσης ώστε να εξυπηρετούν καλύτερα τις ανάγκες μιας διαδικτυακής τάξης:
 - video streaming
 - upload αρχείων
- *Δυνατότητα του δασκάλου να κοστολογεί το μάθημά του.*
Ο δάσκαλος θα έχει την δυνατότητα να χρεώνει το μαθητή για το μάθημα που διδάσκει. Επομένως τα μαθήματα που θα προσφέρονται θα είναι είτε δωρεάν είτε όχι.
- *Δυνατότητα ταξινόμησης και φίλτρου των μαθημάτων που εμφανίζονται στην πλατφόρμα.*
Ο χρήστης θα έχει την δυνατότητα μετά την αναζήτηση για κάποιο μάθημα και αφού εμφανίζονται πολλά σχετικά αποτελέσματα να τα ταξινομεί. Θα δίνονται διάφορες επιλογές ταξινόμησης όπως η τιμή του μαθήματος, τα χρόνια εμπειρίας του δασκάλου κ.α. Επίσης θα μπορέσει να βάλει φίλτρο στα μαθήματα που εμφανίζονται στα αποτελέσματα,

όπως για παράδειγμα αν θέλει τα μαθήματα που θα του εμφανιστούν να είναι δωρεάν ή όχι, σε ποια γλώσσα θα διδάσκεται το μάθημα κ.α.

- *Δυνατότητα ιδιωτικών μαθημάτων.*
Ο δάσκαλος θα μπορεί να δημιουργεί μία διαδικτυακή τάξη όπου δεν θα εμφανίζεται στα αποτελέσματα αναζήτησης όταν κάποιος χρήστη ψάχνει για αυτό το μάθημα. Όμως ο δάσκαλος θα είναι αυτός που θα στέλνει τις προσκλήσεις στους μαθητές του για να παρακολουθήσουν το μάθημα του. Αυτό θα μπορούσε να έχει πρακτική εφαρμογή για παράδειγμα σε ήδη υπάρχουσες σχολικές/φοιτητικές/φροντιστηριακές τάξεις όπου ο δάσκαλος θα θελήσει να κάνει το μάθημα απομακρυσμένα. Έτσι θα ενημερώνει τους μαθητές του για το link της διαδικτυακής τάξης ώστε να παρακολουθήσουν το μάθημα.
- *Δυνατότητα ενημέρωσης των χρηστών μέσω email.*
Οι χρήστες θα ενημερώνονται για τις ανακοινώσεις που λαμβάνουν από την πλατφόρμα και μέσω email. Έτσι ο χρήστης δεν θα καθυστερεί να δει κάποια πιθανόν σημαντική ανακοίνωση.
- *Δυνατότητα αναφοράς των χρηστών για κακόβουλους χρήστης.*
Ο χρήστης θα μπορεί να κάνει «report» για κάποιο μάθημα ή για κάποιο χρήστη στον admin, όταν νομίζει ότι η συμπεριφορά του στην πλατφόρμα δεν είναι η κατάλληλη. Ο admin θα ελέγχει το συγκεκριμένα χρήστη ή μάθημα και θα δρα κατάλληλα.

Παράρτημα

Μοντέλο Ανάλυσης



Αναφορές

- [1] Online μαθήματα στο MIT: <http://ocw.mit.edu/courses/>
- [2] Online μαθήματα στο Cambridge University: <http://www.ice.cam.ac.uk/courses/online-courses>
- [3] Online μαθήματα στο Stanford: <http://www.stanford.edu/online/courses/>
- [4] Περιβάλλον Εργασίας: <http://www.ubuntu.com/download>
- [5] Βάση δεδομένων sqlite: <http://www.sqlite.org/>
- [6] Επίσημο site της Ruby on Rails: <http://rubyonrails.org/>
- [7] Open Source: http://en.wikipedia.org/wiki/Open_source
- [8] Full-stack: http://en.wikipedia.org/wiki/Solution_stack
- [9] Web application framework: http://en.wikipedia.org/wiki/Web_application_framework
- [10] Γλώσσα προγραμματισμού Ruby: [http://en.wikipedia.org/wiki/Ruby_\(programming_language\)](http://en.wikipedia.org/wiki/Ruby_(programming_language))
- [11] David Heinemeier Hansson: http://en.wikipedia.org/wiki/David_Heinemeier_Hansson
- [12] 37signals: <http://en.wikipedia.org/wiki/37signals>
- [13] Object-relational mapping: http://en.wikipedia.org/wiki/Object-relational_mapping
- [14] Plugins: [http://en.wikipedia.org/wiki/Plug-in_\(computing\)](http://en.wikipedia.org/wiki/Plug-in_(computing))
- [15] RubyGems: <http://en.wikipedia.org/wiki/RubyGems>
- [16] Ruby on Rails: [Download: http://rubyonrails.org/download](http://rubyonrails.org/download)
- [17] Unix-like systems: http://en.wikipedia.org/wiki/Unix-like_system
- [18] package management system: http://en.wikipedia.org/wiki/Package_management_system
- [19] Official site Twitter Bootstrap: <http://twitter.github.com/bootstrap/>
- [20] Twitter Bootstrap for Rails: <https://github.com/seyhunak/twitter-bootstrap-rails>
- [21] Cloud πλατφόρμα: http://en.wikipedia.org/wiki/Cloud_platform#Platform
- [22] Heroku cloud application platform: <http://www.heroku.com/>

[23] Cloud platform as a service:

http://en.wikipedia.org/wiki/Platform_as_a_service

[24] Deploy στο Heroku: <https://devcenter.heroku.com/articles/git>

[25] Free video chat API | OpenTok: <http://www.tokbox.com/opentok/api>