



**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ**

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Σχεδιασμός και Υλοποίηση Εφαρμογής για τη Διαχείριση  
Αναθέσεων Μαθημάτων**

**Μπουμπούσης Κ. Άγγελος**

**Παπαδάκης Ε. Μιχαήλ**

**Επιβλέπων: Παναγιώτης Σταματόπουλος, Επίκουρος Καθηγητής ΕΚΠΑ**

**ΑΘΗΝΑ  
ΦΕΒΡΟΥΑΡΙΟΣ 2016**

## **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

Σχεδιασμός και Υλοποίηση Εφαρμογής για τη Διαχείριση Αναθέσεων Μαθημάτων

**Μπουμπούσης Κ. Άγγελος**

**A.M.: 200800103**

**Παπαδάκης Ε. Μιχαήλ**

**A.M.: 200900216**

**Επιβλέπων: Παναγιώτης Σταματόπουλος, Επίκουρος Καθηγητής ΕΚΠΑ**

## ΠΕΡΙΛΗΨΗ

Σε κάθε ακαδημαϊκή περίοδο, ένα πανεπιστημιακό τμήμα πρέπει να διαχειρίζεται τις αναθέσεις σε διδάσκοντες των μαθημάτων που προσφέρει. Κάθε τομέας του τμήματος έχει ένα σύνολο μαθημάτων για τα οποία είναι αρμόδιος να αναθέτει σε διδάσκοντες.

Η εφαρμογή που αναπτύχθηκε δίνει την δυνατότητα online διαχείρισης των μαθημάτων του προπτυχιακού και μεταπτυχιακού προγράμματος σπουδών του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών. Αυτό οφελεί τα μέλη του προσωπικού του πανεπιστημίου να συνεργάζονται εύκολα και αξιόπιστα. Διαχωρίζονται τα καθήκοντα των εργαζομένων του τμήματος, αναθέτοντας στον καθένα τις αρμοδιότητες που του αντιστοιχούν. Μέχρι σήμερα δεν υπήρχε μια κοινή πλατφόρμα στην οποία συνδεόνται όλοι οι χρήστες. Υπήρχε ένα κοινό αρχείο με όλα τα μαθήματα, στο οποίο είχαν πρόσβαση όλοι οι τομείς, με αποτέλεσμα να δημιουργείται σύγχυση, διότι μπορούσε ο κάθε τομέας να τροποποιεί μαθήματα που δεν του ανήκουν.

Οι εργαζόμενοι του πανεπιστημίου έχουν την δυνατότητα να συνδέονται στο σύστημα μέσω ενός λογαριασμού. Αφού εισέλθουν στο λογαριασμό τους, μπορούν να εισαγάγουν, να διαβάσουν, να τροποποιήσουν και να διαγράψουν αναθέσεις για κάθε μάθημα του τομέα που τους αντιστοιχεί.

**ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ:** Σχεδίαση και Υλοποίηση Λογισμικού

**ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:** PHP, Javascript

## **ABSTRACT**

In each academic year, a university department must manage assignments to professors of the courses offered. Each sector of the university department has a set of courses for which is responsible to entrust professors.

The application developed enables online management of the undergraduate and Master courses curriculum of the Department of Informatics and Telecommunications of the University of Athens. This benefits the members of the university personnel to collaborate easily and reliably. With the separation of concerns of the the department's employees, they are assigned with the corresponding responsibilities. Until today there was no platform in which all users were connected. There was a public file of all the courses, which was open to all sectors, resulting in confusion, because every sector could modify courses, which should not.

The university employees are able to connect to the system through one account. When they sign in their account they can, read, modify and delete assignments for each sector's course they belong to.

**SUBJECT AREA:** Software Development

**KEYWORDS:** PHP, Javascript

*Αφιερώνεται στις οικογένειες μας*

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Θα θέλαμε να ευχαριστήσουμε τον επιβλέποντα επικ. καθ. Παναγιώτη Σταματόπουλο για την υπομονή του, την γρήγορη ανταπόκρισή του σε ό,τι χρειαζόμασταν και την άψογη συνεργασία του.

## ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ .....	11
1. ΕΦΑΡΜΟΓΗ ΔΙΑΧΕΙΡΙΣΗΣ ΑΝΑΘΕΣΕΩΝ ΠΑΝΕΠΙΣΤΗΜΙΑΚΩΝ ΜΑΘΗΜΑΤΩΝ.....	12
1.1 Εισαγωγή στην εφαρμογή .....	12
1.2 Βασικές έννοιες της εφαρμογής.....	12
1.3 Λειτουργίες της εφαρμογής.....	13
1.3.1 Γενικές Λειτουργίες της εφαρμογής.....	13
1.3.2 Λειτουργίες διαχείρισης τμημάτων .....	17
2. ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΙ ΒΙΒΛΙΟΘΗΚΕΣ.....	20
2.1 Οργάνωση ανάπτυξης εφαρμογής και τεχνολογίες .....	20
2.2 Εργαλεία ανάπτυξης (Development tools) .....	22
2.3 Πλατφόρμα Υλοποίησης .....	23
2.4 Ανάλυση τεχνολογιών / πρόσθετων επεκτάσεων (plugins) .....	24
2.4.1 Τεχνολογίες νωτιαίου άκρου .....	24
2.4.2 Τεχνολογίες μετωπιαίου άκρου .....	25
2.4.3 Τεχνολογία AJAX Asynchronous Javascript and XML .....	26
2.4.4 Πρόσθετες επεκτάσεις (plugins).....	26
3. ΑΝΑΛΥΣΗ, ΣΧΕΔΙΑΣΜΟΣ, ΥΛΟΠΟΙΗΣΗ ΒΑΣΕΩΣ ΔΕΔΟΜΕΝΩΝ.....	27
3.1 Ανάλυση απαιτήσεων .....	27
3.2 Μοντέλο Οντοτήτων Συσχετίσεων.....	28
3.3 Περιγραφή σχέσεων στο σχεσιακό μοντέλο.....	29
4. ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ.....	33
4.1 Αρχές σχεδιασμού και μοτίβα σχεδιασμού.....	33
4.3 Μοτίβο σχεδιασμού - Μοντέλο Όψη Διαχειριστής (Model View Controller) .....	35
5. ΑΡΧΕΙΑ ΕΦΑΡΜΟΓΗΣ .....	36
5.1 Εισαγωγή .....	36
5.2 Αρχεία ρυθμίσεων.....	36
5.2.1 autoload.php.....	36
5.2.2 config.php.....	36
5.2.3 constants.php .....	36
5.2.4 database.php.....	36
5.2.5 doctypes.php .....	36
5.2.6 foreign_chars.php.....	36
5.2.7 hooks.php.....	37
5.2.8 memcached.php .....	37

5.2.9 migration.php.....	37
5.2.10 mimes.php.....	37
5.2.11 profiler.php .....	37
5.2.12 routes.php .....	37
5.2.13 smileys.php .....	37
5.2.14 user_agents.php.....	37
<b>5.3 Περιγραφή αρχείων Μοντέλου (AssignCourses\application\models\)</b> .....	<b>37</b>
<b>5.4 Περιγραφή αρχείων Όψης (AssignCourses\application\views\)</b> .....	<b>38</b>
<b>5.5 Περιγραφή αρχείων Διαχειριστή (AssignCourses\application\controllers\)</b> .....	<b>38</b>
<b>5.6 Περιγραφή αρχείων css (AssignCourses\assets\css)</b> .....	<b>39</b>
<b>5.7 Περιγραφή αρχείων Javascript (AssignCourses\assets\js)</b> .....	<b>39</b>
<b>6. ΠΡΟΣΤΑΣΙΑ ΚΑΙ ΑΣΦΑΛΕΙΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ</b> .....	<b>40</b>
6.1 Εισαγωγή .....	40
<b>7. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ</b> .....	<b>43</b>
<b>ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ</b> .....	<b>44</b>
<b>ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ</b> .....	<b>45</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ</b> .....	<b>47</b>



## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Ανεπιτυχής προσπάθεια εισόδου χρήστη στην εφαρμογή	14
Εικόνα 2: Φόρμα επεξεργασίας χρήστη	15
Εικόνα 3: Φόρμα ανάθεσης μαθήματος	16
Εικόνα 4: Στιγμιότυπο πλοήγησης μαθημάτων πρώτου εξαμήνου στο προπτυχιακό πρόγραμμα σπουδών για την περίοδο 2015-2016	17
Εικόνα 5: Προβολή τμημάτων του μαθήματος της Γραμμικής Άλγεβρας	18
Εικόνα 6: Εισαγωγή τμήματος στη της Γραμμική Άλγεβρα	19
Εικόνα 7: Μοντέλο Οντοτήτων-Συσχετίσεων	28
Εικόνα 8: Κλασσικό MVC μοντέλο	33
Εικόνα 9: Διάγραμμα ροής δεδομένων της εφαρμογής	34
Εικόνα 10: MVC μοντέλο του CodeIgniter	35

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Assigned_course	29
Πίνακας 2: Course	29
Πίνακας 3: Course_has_period	29
Πίνακας 4: Period	30
Πίνακας 5: Professor	30
Πίνακας 6: Professor_has_section	30
Πίνακας 7: Roles	30
Πίνακας 8: Roles_has_sector	31
Πίνακας 9: Section	31
Πίνακας 10: Sector	31
Πίνακας 11: Sector_has_course	31
Πίνακας 12: User	32
Πίνακας 13: Μοντέλα	37
Πίνακας 14: Όψεις	38
Πίνακας 15: Διαχειριστές	38
Πίνακας 16: Αρχεία css	39
Πίνακας 17: Αρχεία Javascript	39

## ΠΡΟΛΟΓΟΣ

Η παρούσα εργασία διενεργήθηκε στα πλαίσια εκπόνησης πτυχιακής εργασίας στο Τμήμα Πληροφορικής και Τηλεπικοινωνιών της Σχολής Θετικών Επιστημών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών. Πραγματοποιήθηκε υπό την επίβλεψη του Επίκουρου Καθηγητή κ. Παναγιώτη Σταματόπουλου. Για την υλοποίηση της συνεργαστήκαμε και τα δύο μέλη της ομάδας μέσω του λογισμικού Teamviewer, το οποίο μας εξασφάλισε την εξ αποστάσεως σύνδεση μας σε ένα υπολογιστικό σύστημα, Είχαμε άριστη συνεργασία με τον κ. Σταματόπουλο, ο οποίος μας βοήθησε στην πρόοδο του έργου.

## 1. ΕΦΑΡΜΟΓΗ ΔΙΑΧΕΙΡΙΣΗΣ ΑΝΑΘΕΣΕΩΝ ΠΑΝΕΠΙΣΤΗΜΙΑΚΩΝ ΜΑΘΗΜΑΤΩΝ

### 1.1 Εισαγωγή στην εφαρμογή

Η εφαρμογή σχεδιάστηκε για να προσφέρει στα μέλη του πανεπιστημίου τη δυνατότητα να αναθέτουν τα μαθήματα σε καθηγητές και να κατανέμουν τις αναθέσεις αυτές σε τμήματα. Οι χρήστες της εφαρμογής μόλις εισέρχονται στο σύστημα μπορούν να επιλέξουν ακαδημαϊκή περίοδο, πρόγραμμα σπουδών, εξάμηνο και μάθημα. Έπειτα μπορούν να βλέπουν τις υπάρχουσες αναθέσεις μαθημάτων και να τις τροποποιούν ή να τις διαγράψουν ανάλογα με τα δικαιώματά τους.

### 1.2 Βασικές έννοιες της εφαρμογής

Οι κύριες έννοιες της εφαρμογής μας είναι οι εξής :

- **Ακαδημαϊκή περίοδος :** Ως ακαδημαϊκή περίοδο ορίζουμε τη περίοδο διεξαγωγής από την έναρξη του χειμερινού εξαμήνου του ενός έτους μέχρι το τέλος του εαρινού εξαμήνου του επόμενου.
- **Πρόγραμμα σπουδών :** Υπάρχουν δύο προγράμματα σπουδών στα οποία χωρίζονται τα μαθήματα, το *προπτυχιακό* και το *μεταπτυχιακό*.
- **Εξάμηνα :** Το προπτυχιακό πρόγραμμα σπουδών αποτελείται από οκτώ εξάμηνα που απαριθμούνται ως εξής :
  1. *Χειμερινό Εξάμηνο* – (1ο , 3ο , 5ο, 7ο)
  2. *Εαρινό Εξάμηνο* – (2ο, 4ο, 6ο, 8ο)

Το μεταπτυχιακό πρόγραμμα σπουδών αποτελείται από δύο εξάμηνα που είναι το *χειμερινό* και το *εαρινό*.

- **Διδακτικές ώρες :** Με αυτήν την έννοια ορίζουμε το σύνολο των ωρών που διδάσκει ο καθηγητής το μάθημα ανά εβδομάδα.
- **Μαθήματα :** Κάθε πρόγραμμα σπουδών περιέχει ένα σύνολο μαθημάτων. Για μία ακαδημαϊκή περίοδο ανά εξάμηνο ορίζονται τα αντίστοιχα μαθήματα. Κάθε μάθημα κατανέμεται σε τμήματα και διδάσκεται από έναν ή περισσότερους καθηγητές. Υπάρχει περίπτωση ένα μάθημα του προπτυχιακού προγράμματος σπουδών να συνδιδάσκεται με ένα μάθημα του μεταπτυχιακού. Κάθε μάθημα έχει τρεις διαφορετικούς τύπους διδασκαλίας τη *θεωρία*, το *εργαστήριο* και το *φροντιστήριο*.
- **Τομείς :** Υπάρχουν τρεις τομείς μαθημάτων.
- **Καθηγητές :** Ο καθηγητής μπορεί να είναι είτε του τμήματος είτε από κάποια άλλη σχολή του ιδρύματος. Επιπλέον είναι δυνατόν να διδάσκει πολλά μαθήματα και από τα δύο προγράμματα σπουδών
- **Τμήματα :** Ένα τμήμα ενός μαθήματος έχει έναν καθηγητή που το διδάσκει για ένα σύνολο διδακτικών ωρών. Για κάθε τύπο διδασκαλίας του μαθήματος μπορούν να δημιουργηθούν πολλά τμήματα.

- **Υπεύθυνος ανάθεσης μαθήματος :** Για κάθε μάθημα ορίζονται από το σύστημα υπεύθυνοι ανάθεσης μαθήματος. Η αρμοδιότητα τους είναι να αναθέτουν μαθήματα σε διδάσκοντες και να επεξεργάζονται τις αναθέσεις.
- **Χρήστες :** Με τον όρο *χρήστες* θα εννοούμε τους χρήστες της εφαρμογής μας. Κάθε χρήστης ανήκει σε έναν *ρόλο*, που του δίνεται από τον διαχειριστή του συστήματος.
- **Ρόλοι χρηστών :** Οι βασικοί ρόλοι του συστήματος είναι οι εξής:
  - i. *Διαχειριστής συστήματος (administrator) :* Έχει πλήρη έλεγχο πάνω στην εφαρμογή.
  - ii. *Κεντρική γραμματεία :* Έχει πρόσβαση στη διαχείριση των αναθέσεων όλων των μαθημάτων ανεξάρτητα από τον τομέα και το προγράμματα σπουδών. Επίσης μπορεί να εισαγάγει μαθήματα στο σύστημα και να ορίζει υπεύθυνο ανάθεσης μαθήματος για την τρέχουσα ακαδημαϊκή περίοδο.
  - iii. *Γραμματεία κάθε τομέα :* Έχει πρόσβαση στη διαχείριση των τμημάτων όλων των μαθημάτων του τομέα στον οποίο ανήκει.

### 1.3 Λειτουργίες της εφαρμογής

Η εφαρμογή μας προσφέρει κάποιες γενικές λειτουργίες που σχετίζονται με την εισαγωγή και την επεξεργασία των οντοτήτων της εφαρμογής όπως μαθήματα, καθηγητές, δικαιώματα και την κύρια λειτουργία της εφαρμογής που είναι η διαχείριση των αναθέσεων ανά εξάμηνο της ακαδημαϊκής περιόδου που ο χρήστης επιλέγει.

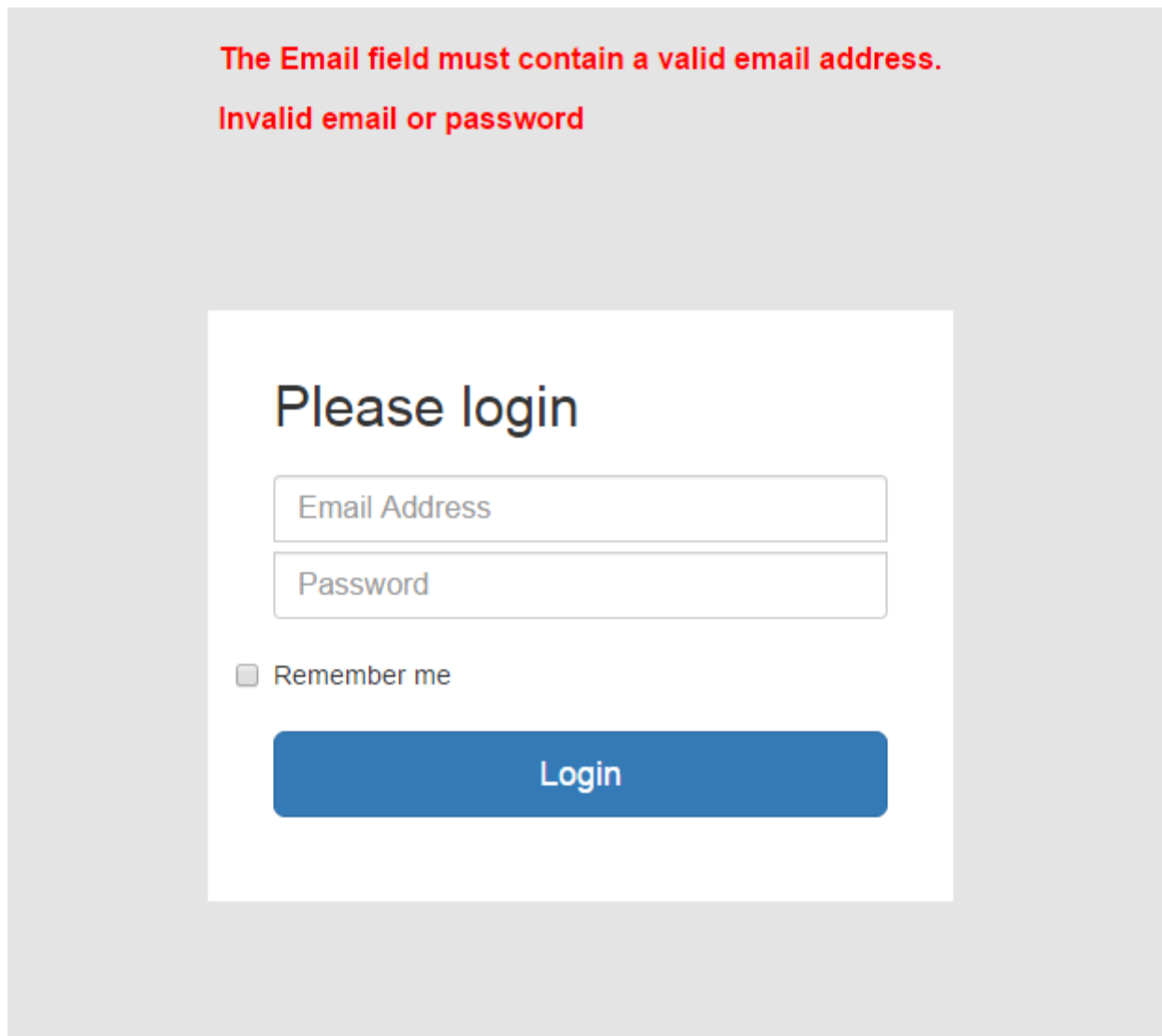
Η χρήση των τεχνολογιών jquery, angularjs και ajax μας επιτρέπουν να υπάρχει αλληλεπίδραση καθώς με κάθε αλλαγή επιλογής που κάνουμε συμβαίνουν διαφορετικές λειτουργίες στο παρασκήνιο στέλνοντας αιτήματα στον server, ενώ η εφαρμογή λειτουργεί κανονικά, χωρίς να περιμένει μέχρι να τελειώσει την εργασία του ο server.

Σε όλες τις σελίδες εμπεριέχεται το μενού, που είναι μία οριζόντια μπάρα πλοήγησης (navigation bar) που έχει επιλογές «Αρχική Σελίδα», «Χρήστες», «Μαθήματα», «Καθηγητές». Ο χρωματισμός, η μορφοποίηση, η στοίχιση, η δομή και όλα τα αντικείμενα γραφιστικής σχεδίασης παραμένουν κοινά για όλες τις σελίδες.

#### 1.3.1 Γενικές Λειτουργίες της εφαρμογής

##### Είσοδος χρήστη στο σύστημα (Log-in)

Στην πρώτη σελίδα της εφαρμογής εμφανίζεται μία φόρμα εισόδου όπου ζητάει από τον χρήστη να εισαγάγει τα στοιχεία του και το κουμπί εισόδου. Αν ο χρήστης εισάγει τα σωστά στοιχεία θα ανακατευθυνθεί στη κεντρική σελίδα της εφαρμογής, ενώ σε διαφορετική περίπτωση θα του εμφανιστούν μηνύματα λάθους πάνω από την φόρμα εισόδου, χωρίς να φορτωθεί εκ νέου η σελίδα.



Εικόνα 1: Ανεπιτυχής προσπάθεια εισόδου χρήστη στην εφαρμογή

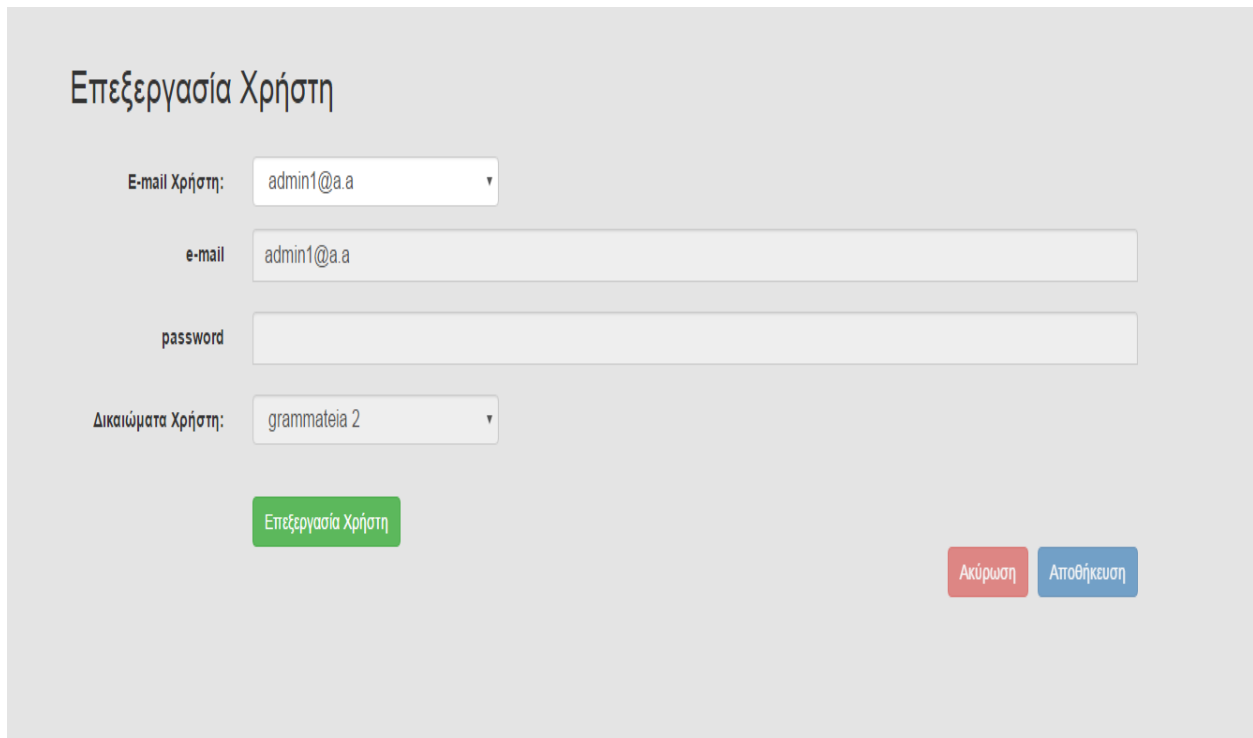
### Εγγραφή χρήστη (User registration)

Ο χρήστης με δικαιώματα διαχειριστή έχει πρόσβαση στην επιλογή της λίστας από το μενού «Χρήστες». Από εκεί έχει πρόσβαση στην εγγραφή, επεξεργασία και διαγραφή χρήστη.

Στην εισαγωγή χρήστη ο διαχειριστής πρέπει να ορίσει ένα username που θα εισάγει ο χρήστης (υπό την μορφή e-mail) σαν αναγνωριστικό όταν εισέρχεται στο σύστημα, τον κωδικό του χρήστη (8-16 χαρακτήρες) και επιλέγει τον ρόλο του από την λίστα των ρόλων. Όταν τελειώσουν οι αλλαγές το σύστημα πρέπει να πατηθεί το κουμπί «Εισαγωγή Χρήστη» και αν ο χρήστης εισαχθεί επιτυχώς στην βάση θα εμφανίσει μήνυμα επιτυχίας, αλλιώς μήνυμα αποτυχίας ανάλογα με το σφάλμα που μπορεί να είχε γίνει (όπως να μην έχει μορφή e-mail το username ή να υπάρχει ήδη ο χρήστης με το ίδιο username στην βάση δεδομένων).

## Επεξεργασία χρήστη (Edit User)

Στην επεξεργασία χρήστη πρέπει αρχικά να επιλέξουμε το όνομα χρήστη από την λίστα με τους χρήστες και έπειτα θα γεμίσουν τα στοιχεία τα οποία στην αρχή δεν είναι ενεργοποιημένα για επεξεργασία. Όταν πατήσουμε το κουμπί «επεξεργασία Χρήστη», θα φορτωθούν τα στοιχεία προς επεξεργασία του συγκεκριμένου χρήστη (e-mail, password, Δικαιώματα Χρήστη) και οι είσοδοι τους θα ενεργοποιηθούν. Στη φόρμα e-mail εμφανίζεται το τωρινό και μπορεί να γίνει αλλαγή του ονόματος του χρήστη (πάντα το καινούργιο πρέπει να κρατάει την δομή ενός e-mail). Η φόρμα του password, για λόγους ασφαλείας, είναι κενή, δηλαδή δεν φαίνεται ο παλιός κωδικός του χρήστη. Εάν δεν είναι επιθυμητή η αλλαγή του κωδικού πρόσβασης, τότε η φόρμα πρέπει να παραμείνει κενή. Στην αντίθετη περίπτωση ο νέος κωδικός πρέπει να είναι 8-16 χαρακτήρες. Στα δικαιώματα χρήστη μπορούμε να επιλέξουμε νέο δικαίωμα ή να αφήσουμε το παλιό. Για κάθε αλλαγή, αν δεν θέλει ο διαχειριστής του συστήματος να αποθηκευτούν οι αλλαγές, πατάει το κουμπί «Ακύρωση» και δεν θα γίνει καμία αλλαγή. Στην αντίθετη περίπτωση, πατάει το κουμπί «Αποθήκευση» και όπως και στην επεξεργασία θα εμφανιστεί μήνυμα επιβεβαίωσης ή αποτυχίας.



Εικόνα 2: Φόρμα επεξεργασίας χρήστη

## Διαγραφή χρήστη (Delete User)

Στην διαγραφή χρήστη μπορούν να διαγραφούν οι λογαριασμοί των χρηστών, εκτός από τους διαχειριστές συστήματος. Ένας διαχειριστής δεν μπορεί να σβηστεί.

## Ανάθεση μαθήματος

Στην ανάθεση μαθήματος υπάρχουν κουμπιά με που επιλέγονται οι ρόλοι ανάθεσης. Ο κάθε ρόλος μπορεί να επιλεγεί ξεχωριστά από τον προηγούμενο. Οι ρόλοι του διαχειριστή και της κεντρικής γραμματείας θεωρούνται πάντα ότι έχουν δικαίωμα ανάθεσης σε όλα τα μαθήματα. Επιπλέον κάθε γραμματεία ανά τομέα έχει δικαίωμα στα μαθήματα του τομέα της. Αρχικά αν το μάθημα δεν έχει ανατεθεί σε εξάμηνο του τρέχοντος έτους, δεν υπάρχει κάποιος υπεύθυνος σε αυτό. Θεωρείται ότι δεν διδάσκεται φέτος και δεν του μπορεί να γίνει ανάθεση. Για να γίνει ένα μάθημα διαθέσιμο πρέπει να ανατεθεί από εδώ. Κατά την επιλογή εξαμήνου και μαθήματος εμφανίζονται οι ρόλοι που μπορούν να είναι υπεύθυνοι για τα μαθήματα του έτους. Τα εξάμηνα για τα προπτυχιακά είναι οκτώ (8) και για τα μεταπτυχιακά ένα και δύο (1,2) και συμβολίζουν χειμερινό και εαρινό αντίστοιχα. Τα μαθήματα στην λίστα μαθημάτων εμφανίζονται με αλφαβητική σειρά.

Εικόνα 3: Φόρμα ανάθεσης μαθήματος

## Εισαγωγή ρόλου

Στην φόρμα εισαγωγής ρόλου ο διαχειριστής της πλατφόρμας εισάγει ένα νέο όνομα ρόλου. Έπειτα επιλέγει έναν δύο ή τρεις τομείς από τις επιλογή τομέα. Η επιλογή αυτή καθορίζει τα μαθήματα του τομέα που μπορεί να δει και κατά συνέπεια περιορίζει τα μαθήματα που μπορεί να γίνει υπεύθυνος.

Στην διαγραφή ρόλου ο διαχειριστής επιλέγει τους ρόλους που επιθυμεί να διαγράψει. Μπορεί να διαγράψει όλους τους ρόλους εκτός από τον ρόλο διαχειριστή, της κεντρικής γραμματείας, και των γραμματειών τομέων. Εάν ένας χρήστης ανήκει σε έναν ρόλο που είναι προς διαγραφή τότε θα εμφανιστεί μήνυμα με το όνομά του διότι πρέπει να γίνει πρώτα να γίνει εκεί αλλαγή.

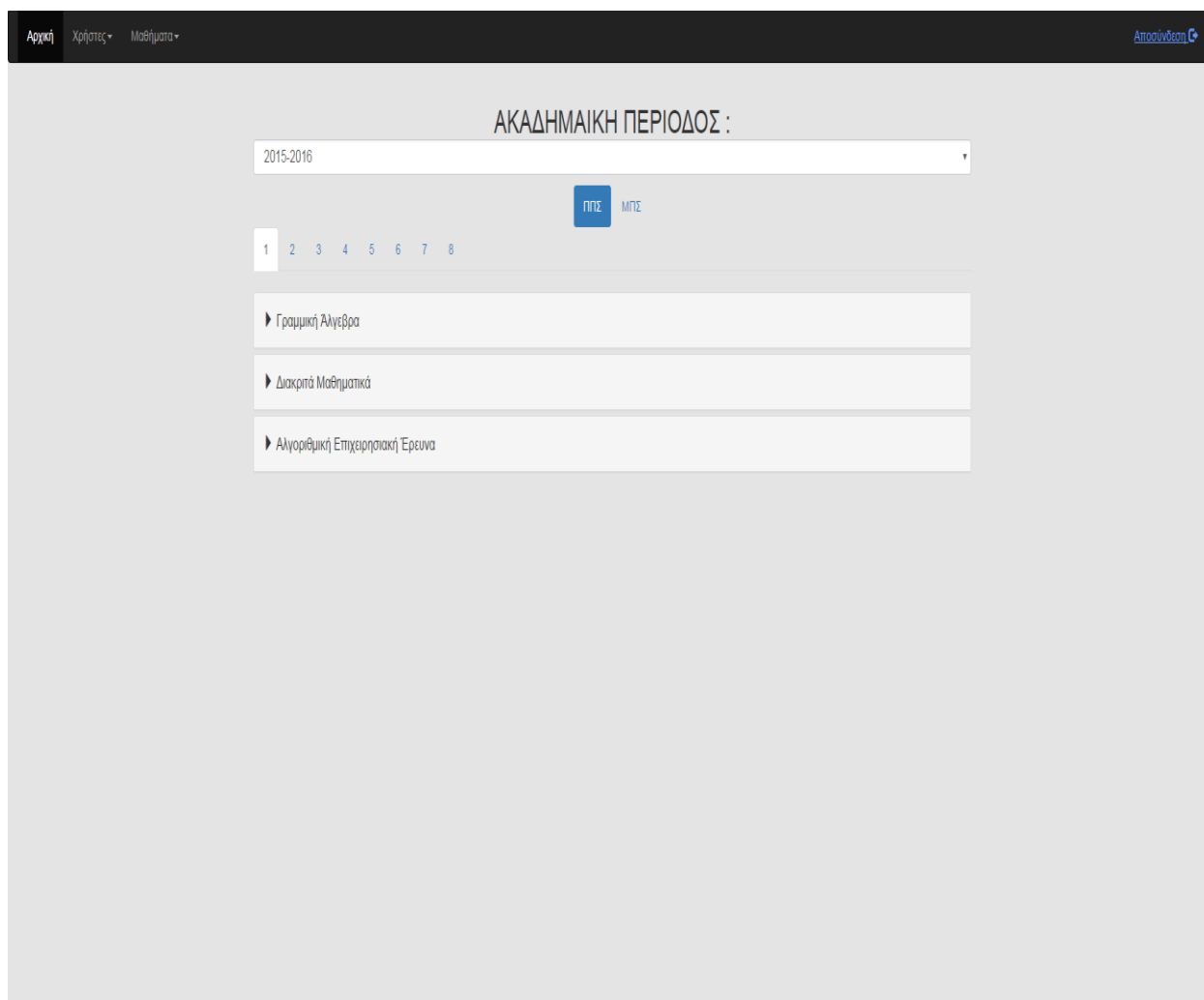


### 1.3.2 Λειτουργίες διαχείρισης τμημάτων

Όταν ο χρήστης εισέρχεται στο σύστημα βάζοντας τα σωστά στοιχεία του, βλέπει την κύρια οθόνη που περιλαμβάνει το μενού, μια λίστα επιλογών (dropdown list) των ακαδημαϊκών περιόδων και κάτω από τη λίστα είναι οι επιλογές για το προπτυχιακό και το μεταπτυχιακό πρόγραμμα σπουδών.

#### Πλοήγηση στα μαθήματα

Όταν ο χρήστης επιλέγει ακαδημαϊκή περίοδο και πρόγραμμα σπουδών, τότε εμφανίζεται ένα μενού από tabs με τα εξάμηνα. Τα tabs αλλάζουν ανάλογα με το πρόγραμμα σπουδών το οποίο επιλέγεται. Πατώντας ένα εξάμηνο αλλάζει το περιεχόμενο των μαθημάτων κάθε φορά. Εμφανίζονται τα μαθήματα με τη μορφή μίας λίστας accordion, όπου επιλέγοντας κάθε φορά ένα μάθημα σύρεται ένα πλαίσιο κάτω από το μάθημα με τις πληροφορίες των τμημάτων. Όταν επιλέγει ένα άλλο μάθημα το προηγούμενο πλαίσιο κλείνει και ανοίγει αμέσως το πλαίσιο του τελευταίου μαθήματος που πατήθηκε. Μέσα σε αυτό το κεντρικό πλαίσιο εμφανίζονται τα τμήματα σε έναν πίνακα και οι χρήστες μπορούν να κάνουν διάφορες λειτουργίες.



**Εικόνα 4: Στιγμιότυπο πλοήγησης μαθημάτων πρώτου εξαμήνου στο προπτυχιακό πρόγραμμα σπουδών για τη περίοδο 2015-2016**

## Προβολή τμημάτων ανά μάθημα

Όλοι οι χρήστες ανεξάρτητα από τον ρόλο που τους προσδιορίζει έχουν δικαίωμα να βλέπουν πληροφορίες για τα τμήματα ανά μάθημα. Στις γραμμές του πίνακα βλέπουν για κάθε τμήμα πληροφορίες όπως το όνομα του καθηγητή, τον τύπο του μαθήματος και τις διδακτικές ώρες. Αν οι χρήστες έχουν δικαιώματα εγγραφής βλέπουν και κάποιες επιπλέον λειτουργίες όπως κουμπί εισαγωγής τμήματος, κουμπί επεξεργασίας σε κάθε τμήμα, αλλά και κουμπί διαγραφής του τμήματος. Οι χρήστες που δεν έχουν δικαιώματα διαχειριστή και κεντρικής γραμματείας δε μπορούν να βλέπουν αυτά τα κουμπιά για μαθήματα παλαιότερων ακαδημαϊκών περιόδων, παρά μόνο για το τρέχον ακαδημαϊκό έτος.

ΑΚΑΔΗΜΑΙΚΗ ΠΕΡΙΟΔΟΣ :

2015-2016

ΠΠΣ ΜΠΣ

1 2 3 4 5 6 7 8

► Γραμμική Άλγεβρα

Καθηγητής	Τμήμα	Ωρες		
Αχλιόπτας Δημήτριος	Theoria	4	Επεξεργασία	-
Ρουσσοπούλου Μέμα	Theoria	2	Επεξεργασία	-

Εικόνα 5: Προβολή τμημάτων του μαθήματος της Γραμμικής Άλγεβρας

## Εισαγωγή τμήματος σε μάθημα

Στην επικεφαλίδα του τμήματος υπάρχει ένα κουμπί εισαγωγής τμήματος. Όταν πατείται αυτό το κουμπί εμφανίζεται ένα πλαίσιο που περιέχει μια φόρμα με πεδία που εισάγει ο χρήστης. Η φόρμα περιέχει μια λίστα επιλογών (dropdown list) με τους καθηγητές, μία λίστα επιλογών με τα τμήματα και μία είσοδο για τις διδακτικές ώρες. Αν ο χρήστης βάλει λάθος τιμή στις διδακτικές ώρες, του εμφανίζεται ένα μήνυμα πληροφορώντας για το λάθος του. Όταν πατηθεί το κουμπί αποδοχής, εκτελείται η εισαγωγή, ενώ όταν πατηθεί το κουμπί της ακύρωσης, ακυρώνεται. Αν εισαχθεί το τμήμα επιτυχώς στη βάση, ο χρήστης βλέπει επιτόπου μια νέα εγγραφή στον πίνακα των τμημάτων με το τμήμα που μόλις εισήχθη.

### ΑΚΑΔΗΜΑΙΚΗ ΠΕΡΙΟΔΟΣ :

2015-2016

ΠΠΣ

ΜΠΣ

1

2345678

▶

Γραμμική Άλγεβρα

Καθηγητής	Τμήμα	Ώρες		
Αχλιόπτας Δημήτριος	Theoria	4	Επεξεργασία	-
Ρουσσοπούλου Μέμα	Theoria	2	Επεξεργασία	-

Εισαγωγή Στοιχείων

Όνομα: Αχλιόπτας Δημήτριος
✓ ✕

Τμήμα: Ergasthrio

Ώρες:

Εικόνα 6: Εισαγωγή τμήματος στη Γραμμική Άλγεβρα

### Επεξεργασία τμήματος μαθήματος

Ο χρήστης όταν πατήσει το κουμπί της επεξεργασίας βλέπει το ίδιο πλαίσιο που εμφανίζεται κατά την εισαγωγή. Όταν βάλει σε κάποιο πεδίο μη επιτρεπτή τιμή, του εμφανίζονται μηνύματα alert πληροφορώντας για το λάθος που συνέβη από την επιλογή του. Αν π.χ για ένα τμήμα αλλάξει τον καθηγητή και τον τύπο του μαθήματος σε τιμές που συμπίπτουν με ένα ήδη υπάρχον τμήμα τότε θα του εμφανιστεί ανάλογο μήνυμα. Μόλις πατήσει το κουμπί της αποδοχής και ενημερωθεί η βάση, αμέσως ενημερώνεται και η γραφική διεπαφή.

### Διαγραφή τμήματος μαθήματος

Ο χρήστης όταν πατήσει το κουμπί της διαγραφής το τμήμα αμέσως σβήνεται από τη βάση αλλά και επιτόπου από τη γραφική διεπαφή που χειρίζεται ο χρήστης.

## 2. ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΙ ΒΙΒΛΙΟΘΗΚΕΣ

### 2.1 Οργάνωση ανάπτυξης εφαρμογής και τεχνολογίες

Η ανάπτυξη και η υλοποίηση της εφαρμογής, όπως και κάθε άλλης διαδικτυακής εφαρμογής μπορεί να γίνει με τη χρήση πολλών και διαφορετικών γλωσσών προγραμματισμού και εργαλείων ανάπτυξης. Οι εφαρμογές συνήθως διαιρούνται σε δύο βασικά και ανεξάρτητα μέρη: το *μετωπιαίο άκρο* (front end), που είναι η γραφική διεπαφή που διαχειρίζονται οι *χρήστες* (clients) και το *νωτιαίο άκρο* (back end), που είναι οι λειτουργίες προγραμματιστικής λογικής που χρησιμοποιεί ο *διακομιστής* (server). Οι προσεγγίσεις ανάπτυξης μιας διαδικτυακής εφαρμογής που μπορούν να ακολουθηθούν τόσο σε επίπεδο σχεδίασης όσο και υλοποίησης, είναι πάρα πολλές. Γι' αυτό, για κάθε εφαρμογή χρειάζεται να ληφθούν υπόψη οι ανάγκες, το μέγεθος η πολυπλοκότητα της, ο επιθυμητός χρόνος υλοποίησης και οι γνώσεις-δεξιότητες των προγραμματιστών που θα την αναπτύξουν. Παρακάτω θα αναφέρουμε μερικές προσεγγίσεις ανάπτυξης λογισμικού :

1. Μία προσέγγιση θα ήταν η χρήση των JavaEE τεχνολογιών (Java Servlets, JSP, JSF, JPA, κλπ...) οι οποίες είναι αναπτυγμένες απ την Oracle και κυριαρχούν στον χώρο της βιομηχανίας σε συνδυασμό με άλλα ολοκληρωμένα μοντέλα προγραμματισμού (frameworks) όπως το Spring, Struts, Wicket, Vaadin κλπ. Υπάρχουν πολλές δυνατότητες και πολλές έτοιμες λύσεις στη Java που δίνουν στους προγραμματιστές τη δυνατότητα να αναπτύξουν κώδικα πάνω στις πιο σύγχρονες μέθοδοι σχεδιασμού (*Διαχωρισμός Αρμοδιοτήτων, Αρχή σχεδιασμού της απλής υπευθυνότητας, Μοντέλο-Όψη-Ελεγκτήρας*, κλπ...). Τα κύρια πλεονεκτήματα που προσφέρουν είναι τα εξής:

- **Επεκτασιμότητα (scalability)** : Οι πλατφόρμες της Java παρέχουν τη δυνατότητα να προστίθενται συνεχώς νέες λειτουργικότητες και νέες οντότητες στο σύστημα χωρίς να χρειάζεται να γίνονται πολλές αλλαγές στο ήδη υπάρχον λογισμικό.

- **Χρόνος Παραγωγικότητας (Development Time)** : Ακολουθώντας την σχεδίαση του διαχωρισμού των αρμοδιοτήτων, κάθε προγραμματιστής αναλαμβάνει το δικό του τμήμα της λειτουργικότητας που του έχει ανατεθεί, χωρίς να χρειάζεται να κατανοήσει τον κώδικα που είναι αναπτυγμένος από κάποιους άλλους προγραμματιστές. Έτσι σε μακροπρόθεσμο επίπεδο γλιτώνεται αρκετός χρόνος στην ανάπτυξη κώδικα και στην τελική ολοκλήρωση του έργου.

- **Συντήρηση λογισμικού και Έλεγχος (Software Maintenance and Testing)** : Υπάρχουν πολλά υποστηρικτικά εργαλεία για Java όπως Maven/Ant, JavaDoc, Junit, τα οποία παρέχουν δυνατότητες συντήρησης και ελέγχων σε φιλικό περιβάλλον.

- **Λογισμικά τρίτων (Third party tools)** : Η Java παρέχει πολλές εξωτερικές βιβλιοθήκες για εξειδικευμένες και εξεζητημένες λειτουργικότητες, με αρκετή αξιοπιστία, πολύ καλούς οδηγούς ανάπτυξης (documentation) και με αρκετά παραδείγματα εκμάθησης. που επιλέχτηκε είναι η PHP. Είναι η πιο απλή και διάσημη προγραμματισμού και είναι η μόνη γλώσσα αυστηρά σχεδιασμένη για το διαδίκτυο. Η εκμάθηση της είναι εύκολη και η εκκίνηση μιας εφαρμογής σε PHP είναι πολύ απλή και γρήγορη.

2. Η άλλη προσέγγιση θα ήταν να φτιαχτεί η εφαρμογή σε περιβάλλον PHP. Τα κύρια πλεονεκτήματά της είναι τα εξής :

- **Απλότητα (Simplicity)** : Η PHP είναι η πιο απλή και διάσημη γλώσσα προγραμματισμού παγκοσμίως και είναι η μόνη γλώσσα προγραμματισμού αυστηρά σχεδιασμένη για το διαδίκτυο.
- **Εύκολη εκμάθηση (Easy Learning)** : Η εκμάθηση της PHP είναι πολύ εύκολη και δεν απαιτείται αυστηρά η γνώση και κατανόηση σύνθετων και πολύπλοκων μεθόδων σχεδίασης και ανάπτυξης λογισμικού.
- **Γρήγορη εκκίνηση και ανάπτυξη (setup and deployment)** : Η εκκίνηση μιας εφαρμογής σε PHP είναι πολύ απλή και γρήγορη, εφόσον έχουμε πρόσβαση σε κάποιον server όπως Apache. Κάθε φορά που ο χρήστης αλλάζει τον κώδικα μπορεί να ανανεώνεται απλά ο φυλλομετρητής ιστού (web browser), χωρίς να χρειάζεται να γίνεται επανεκκίνηση του server από κάποιο εξωτερικό εργαλείο.

Με βάση τις ανάγκες της εφαρμογής, αλλά και τις απαιτήσεις του συστήματος που μας παρέχει το τμήμα, καταλήξαμε ότι η καταλληλότερη προσέγγιση είναι η δεύτερη. Ο server της σχολής μας δεν προσφέρει το κατάλληλο περιβάλλον εργασίας για Java, ενώ αρκετές άλλες εφαρμογές ανεπτυγμένες σε PHP, δουλεύουν αρκετά καλά. Το μέγεθος της πολυπλοκότητας και της έκτασης της εφαρμογής δεν είναι τέτοιο που να χρειάζονται αρκετά σύνθετες μέθοδοι σχεδιασμού που παρέχει η Java, αλλά ούτε να χρειάζεται η εγκατάσταση του Apache Tomcat Container, το οποίο θα έπρεπε συνεχώς να επανεκκινείται κάθε φορά που αλλάζουν κάποια τμήματα του κώδικα της εφαρμογής (redployment).

## 2.2 Εργαλεία ανάπτυξης (Development tools)

Στη συνέχεια θα γίνει μια σύντομη περιγραφή των εργαλείων που χρησιμοποιήσαμε για την σχεδίαση και την ανάπτυξη της εφαρμογής.

**1. NetBeans 8.0.2, Php Development :** Το NetBeans-Php Development είναι ένα δωρεάν περιβάλλον ανάπτυξης εφαρμογών με λογισμικό ανοιχτού κώδικα, το οποίο υποστηρίζει τις ακόλουθες γλώσσες προγραμματισμού: PHP,HTML5/JavaScript. Δίνει τη δυνατότητα να αναπτυχθούν διάφοροι τύποι εφαρμογών και υποστηρίζει πολλές χρήσιμες λειτουργικότητες για τους προγραμματιστές όπως refactoring, προφίλ, το τονισμένο χρώμα σύνταξης, αυτόματη συμπλήρωση, ορισμό κωδικών templates κλπ. Είναι αρκετά διαδεδομένο περιβάλλον εργασίας στο χώρο των εφαρμογών του διαδικτύου. Με το πάτημα ενός κουμπιού ανοίγει αυτομάτως ο φυλλομετρητής ιστού και με τις κατάλληλες ρυθμίσεις που γίνονται μόνο μια φορά αναλαμβάνεται η αλληλεπίδραση της εφαρμογής με τον server.

**2. XAMPP :** Το XAMPP είναι το εργαλείο που εγκαταστάθηκε στα τοπικά μηχανήματα και χρησιμοποιήθηκε για την φιλοξενία του ιστοχώρου.Είναι ένα σύνολο προγραμμάτων ελεύθερου λογισμικού (open-source), ανεξαρτήτου πλατφόρμας (cross-platform), το οποίο αποτελείται από έναν εξυπηρετητή ιστοσελίδων http Apache, την βάση δεδομένων MySQL και ένα διερμηνέα για τις γλώσσες προγραμματισμού PHP και Perl. Λόγω του ότι είναι ανεξάρτητο πλατφόρμας τρέχει σε Microsoft Windows, Linux, Solaris και σε MAC OS X. Το ακρωνύμιο XAMPP αναφέρεται στα εξής αρχικά :

- a. **X** (Cross-Platform)
- b. **A**pache HTTP εξυπηρετητής
- c. **M**ySQL
- d. **P**HP
- e. **P**erl

Το XAMPP περιέχει επίσης το εργαλείο phpMyAdmin το οποίο το χρησιμοποιήσαμε για τη διαχείριση της βάσης δεδομένων

**3. XAMPP – phpMyAdmin :** Το phpMyAdmin είναι ένα εργαλείο που παρέχει το XAMPP με το οποίο διαχειριστήκαμε εύκολα τη βάση δεδομένων με τη χρήση έτοιμων γραφικών στοιχείων (widgets) με τα χρησιμοποιώντας τις διάφορες λειτουργίες που προσφέρει όπως εισαγωγή και εξαγωγή αρχείων sql,txt από και προς τη βάση και για την εκτέλεση διάφορων sql επερωτήσεων πάνω στη βάση δεδομένων.

**4. MySQL Workbench :** Το MySQL Workbench είναι ένα εργαλείο οπτικής αναπαράστασης σχημάτων βάσεων δεδομένων MySQL. Το χρησιμοποιήσαμε για την παραγωγή του διαγράμματος επεκταμένου μοντέλου οντοτήτων συσχετίσεων (EER diagram) και για την εξαγωγή του ως sql script με την δημιουργία σχήματος της βάσης δεδομένων.

**5. Chrome και Inspect Element :** Ο φυλλομετρητής ιστού που χρησιμοποιήσαμε για να βλέπουμε την εφαρμογή μας είναι ο Chrome. Το Inspect Element είναι ένα εσωτερικό εργαλείο αποσφαλμάτωσης (debugging) του Chrome, το οποίο μας επιτρέπει να έχουμε εύκολη πρόσβαση στον κώδικα της ιστοσελίδας HTML, JavaScript και CSS και να κάνουμε επιτόπου αλλαγές βλέποντας τες στην ιστοσελίδα.

## 2.3 Πλατφόρμα Υλοποίησης

Η πλατφόρμα υλοποίησης της εφαρμογής που επιλέξαμε είναι το CodeIgniter 3.0.3, το οποίο είναι ένα ολοκληρωμένο μοντέλο προγραμματισμού (framework), ανοιχτού κώδικα. Είναι πολύ απλό και εύκολο στη χρήση, δίνοντας τη δυνατότητα να αναπτυχθούν από απλές έως σύνθετες εφαρμογές.

Γιατί CodeIgniter ?

- **Πολύ χαμηλές απαιτήσεις :** Το CodeIgniter έχει χωρητικότητα μόνο δύο megabyte και το μόνο που απαιτείται είναι η PHP 5.2.4.
- **Αποδοτικότητα (Performance) :** Λόγω της απλής δομής και λειτουργίας του, το CodeIgniter δεν καταναλώνει πολλούς πόρους του συστήματος και αναλαμβάνει τη βελτιστοποίηση κώδικα. Παρέχει δυνατότητες στον προγραμματιστή να χρησιμοποιήσει σύγχρονες μεθόδους βελτιστοποίησης όπως caching, benchmarking.
- **Ελευθερία επιλογής σχεδίασης λογισμικού :** Δεν απαιτεί να ακολουθηθούν συγκεκριμένοι κανόνες συγγραφής κώδικα, επιτρέποντας στον προγραμματιστή τη δυνατότητα να χρησιμοποιεί το δικό του τρόπο και χειρισμό του κώδικα. Κύριο μοτίβο σχεδίασης είναι το MVC μοντέλο, χωρίς όμως να περιορίζεται ο προγραμματιστής σε αυτό. Το CodeIgniter παρέχει απλές λύσεις έναντι πολύπλοκων μεθόδων προγραμματισμού και επιτρέπει τον δομημένο και ευανάγνωστο κώδικα, λύνοντας προβλήματα μακροσκελών προγραμμάτων με διπλότυπο κώδικα.
- **Εύκολη εκκίνηση (easy startup) :** Δεν χρειάζονται εξωτερικά εργαλεία για την εγκατάσταση του framework , όπως το composer ή η χρήση εντολών σε τερματικό. Απλά ανεβάζονται οι φάκελοι με τα αρχεία στον server.
- **Επικοινωνία με λογισμικά τρίτων (integration with third-party tools) :** Το CodeIgniter επιτρέπει την εύκολη επικοινωνία με βιβλιοθήκες και plugins για το front-end που είναι γραμμένες σε javascript, χωρίς να χρειάζεται κάποιο εργαλείο διαχείρισης επεκτάσεων (extentions).
- **Εύκολος χειρισμός ρυθμίσεων (configurations) :** Οι πιο πολλές ρυθμίσεις γίνονται αυτομάτως από το CodeIgniter, χωρίς ο προγραμματιστής να χρειάζεται να πειράξει αρχεία ρυθμίσεων. Σε περίπτωση που θέλει να αλλάξει τις ρυθμίσεις, αυτό γίνεται εύκολα από τον φάκελο config στον οποίο βρίσκονται όλα τα αρχεία ρυθμίσεων.
- **Εύκολη εκμάθηση (easy learning) :** Είναι πολύ εύκολο στη χρήση και στη κατανόηση της δομής των αρχείων, σε σύγκριση με άλλα γνωστά frameworks, τα οποία ακολουθούν πολύπλοκες τεχνικές σχεδίασης. Επιπλέον είναι εύκολο να γράψει ο προγραμματιστής νέες βιβλιοθήκες και να αλλάξει τη συμπεριφορά των ήδη υπάρχοντων βιβλιοθηκών, χωρίς ιδιαίτερη προσπάθεια, αυξάνοντας έτσι τον χρόνο και την ποιότητα παραγωγής κώδικα.
- **Πολλές υπηρεσίες-λειτουργίες (features) σε ένα :** Παρέχει πολλές έτοιμες λύσεις στους προγραμματιστές, όπως λειτουργίες :
  - Αποστολής email (Email sender)
  - Θέματα ασφάλειας (Security libraries)
  - Συμπίεση αρχείων (Zip compression)
  - Υπηρεσία ημερολόγιου (Calendar)
  - Διαχείριση συνόδων (Session handling)
- **Καθαρό κείμενο τεκμηρίωσης (clear documentation) :** Το CodeIgniter

παρέχει ένα πολύ καλό και ακριβές κείμενο τεκμηρίωσης, καθοδηγώντας προοδευτικά τον προγραμματιστή από απλά εισαγωγικά παραδείγματα σε τμήματα κώδικα για τη λειτουργία συγκεκριμένων μεθόδων και κλάσεων της πλατφόρμας.

Άλλες πλατφόρμες που θα μπορούσαν να χρησιμοποιηθούν είναι οι Yii, Zend, Symfony. Λόγω του πιο απαιτητικού επιπέδου εκμάθησης τους και της πολυπλοκότητάς τους επιλέξαμε το Codeigniter.

## 2.4 Ανάλυση τεχνολογιών / πρόσθετων επεκτάσεων (plugins)

### 2.4.1 Τεχνολογίες νωτιαίου άκρου

- **PHP: Hypertext Preprocessor**

Η PHP είναι η πιο διάσημη και διαδεδομένη γλώσσα προγραμματισμού στο χώρο της ανάπτυξης διαδικτυακών εφαρμογών. Οι περισσότεροι εξυπηρετητές (hosts) και διακομιστές (servers) παγκοσμίως υποστηρίζουν php. Υποστηρίζει τη πολύ γρήγορη κατασκευή μιας ιστοσελίδας, χωρίς χρονοβόρες διαδικασίες. Η PHP μοιάζει πολύ με την γλώσσα C, καθώς έχει δανειστεί το συντακτικό της από αυτήν, ενώ υποστηρίζει και αντικειμενοστραφή προγραμματισμό.

Αναλαμβάνει την κεντρική διαχείριση των λειτουργιών της εφαρμογής μας και την διασύνδεση με τη βάση δεδομένων. Πιο αναλυτικά διαβάζει και επεξεργάζεται τα δεδομένα που βάζει ο χρήστης από τη γραφική διεπαφή, διασφαλίζοντας την ασφάλεια τους από κακόβουλες επιθέσεις και την αποθήκευσή τους στη βάση. Στη συνέχεια στέννει στο χρήστη τις κατάλληλες απαντήσεις ανάλογα με τα αιτήματα που ζήτησε.

- **MySQL / MySQLi Extension**

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων ανοικτού κώδικα (relational database management system), που χρησιμοποιεί την SQL (Structured Query Language), που είναι η πιο γνωστή γλώσσα παγκοσμίως για την εκτέλεση λειτουργιών όπως εισαγωγή, ανάγνωση, επεξεργασία και διαγραφή δεδομένων σε μία βάση δεδομένων. Όλες οι πληροφορίες που χρειάζεται να καταγραφούν από την εφαρμογή μας αποθηκεύονται στη βάση δεδομένων μέσω σύνδεσης με τον MySQL Server.

Το MySQLi extension είναι μία προέκταση της MySQL που λειτουργεί για τις εκδόσεις mysql 4.1 και πάνω. Η php5 παρέχει ένα mysqli interface που προσφέρουν ασφαλείς τρόπους στον χρήστη για να επικοινωνεί σωστά με τη βάση δεδομένων αποτρέποντας κάποιες κακόβουλες επιθέσεις όπως το sql.



## 2.4.2 Τεχνολογίες μετωπιαίου άκρου

- **HTML: HyperText Markup Language**

Η HTML είναι η βασική γλώσσα δόμησης σελίδων στο διαδίκτυο. Είναι μία γλώσσα προγραμματισμού. Σκόπος της είναι η καλή εμφάνιση πληροφοριών (κείμενο, εικόνες, ήχος, βίντεο, σύνδεσμοι). Περιλαμβάνει ένα σύνολο από στοιχεία (tags) για διάφορετικούς σκοπούς όπως μορφοποίηση κειμένου (πάραγραφοι, επικεφαλίδες), δημιουργία συνδέσμων (links), πινάκων (tables), φόρμες εγγραφής, λίστες, κουμπιά (buttons) κ.α. Τα tags μεταφράζονται στη κατάλληλη μορφή από τον φυλλομετρητή ιστού, με τελικό αποτέλεσμα την εμφάνιση της ιστοσελίδας στον χρήστη. Η έκδοση που χρησιμοποιούμε είναι η HTML5.

Τα αρχεία του μετωπιαίου άκρου της εφαρμογής μας είναι οι όψεις (views), τα οποία είναι ρηρ αρχεία τα οποία περιέχουν HTML κώδικα.

- **Javascript**

Η HTML σαν γλώσσα διαδικτύου έχει κάποια όρια, καθώς δεν επιτρέπει την αλληλεπίδραση μιας σελίδας με τους χρήστες μέσω δυναμικής αποστολής και λήψεων αιτημάτων από τον εξυπηρετητή (server). Αυτό το πρόβλημα έρχεται να λύσει μία scripting γλώσσα προγραμματισμού για τους φυλλομετρητές, η Javascript. Η Javascript εισάγει διαδραστικότητα στις σελίδες HTML, δίνοντας μας τη δυνατότητα να ελέγχουμε για πιθανά συμβάντα (π.χ. πάτημα ενός κουμπιού, αλλαγή στοιχείου λίστας ή πληκτρολόγηση σε κάποια είσοδο) και να εκτελούνται οι αντίστοιχες διαδικασίες ανάλογα με το συμβάν που έλαβε χώρα.

- **Βιβλιοθήκη JQuery**

Η JQuery είναι μια βιβλιοθήκη ανοιχτού κώδικα γραμμένη σε Javascript που σκοπό έχει να παρέχει στον χρήστη έτοιμες λειτουργίες που απλοποιούν την υλοποίηση κώδικα από τη πλευρά του πελάτη. Υποστηρίζει πολλούς φυλλομετρητές ιστού και πολλές λειτουργικότητες όπως :

- ο Τεχνολογίες Ajax
- ο Διαχείριση του Dom (Document Object Model) Tree
- ο Επεκτασιμότητα μέσω plug-ins
- ο Συμβάντα (Events)

- **CSS (Cascading Style Sheets)**

Το Css είναι μία απλή γλώσσα προγραμματισμού που βοηθάει στην περιγραφή και τη μορφοποίηση διάφορων html στοιχείων μέσα σε μία ιστοσελίδα.

- **Bootstrap**

Το Bootstrap είναι ένα σύνολο εργαλείων ανοιχτού κώδικα, το οποίο περιέχει έτοιμες ρυθμίσεις πάνω σε κάποια στοιχεία html και css selectors και πολλά αντικείμενα με συγκεκριμένες έτοιμες λειτουργίες, όπως δυναμικά μενού, tabs, λίστες accordion, πίνακες, κουμπιά κλπ.

- **AngularJs**

Η angularjs είναι ένα περιβάλλον προγραμματισμού σε javascript, το οποίο μας βοηθάει να γράφουμε λίγο html κώδικα αποφεύγοντας διπλότυπο κώδικα που εκτελεί την ίδια λειτουργία.

### **2.4.3 Τεχνολογία AJAX Asynchronous Javascript and XML**

Η τεχνολογία AJAX είναι σήμερα η πιο διαδεδομένη τεχνολογία προγραμματισμού στις διαδικτυακές εφαρμογές. Δίνει τη δυνατότητα στον χρήστη να αλληλεπιδράσει με την ιστοσελίδα, χωρίς να χρειάζεται επαναφόρτωση της σελίδας, ενώ βοηθάει και στον τομέα της ασφάλειας, καθώς δε χρειάζεται η αλλαγή του URL στη μπάρα διεύθυνσης του φυλλομετρητή ιστού.

### **2.4.4 Πρόσθετες επεκτάσεις (plugins)**

Έχουν χρησιμοποιηθεί κάποιες πρόσθετες επεκτάσεις σε javascript και jquery, που μας βοήθησαν για την προσφορά φιλικής διάδρασης της εφαρμογής μας με τον χρήστη.

#### **X-editable**

Το x-editable είναι μια jquery προέκταση η οποία μας επιτρέπει να τροποποιούμε μια τιμή ενός στοιχείου html. Χρησιμοποιείται για το pop-up πλαίσιο που εμφανίζεται κατά το πάτημα του κουμπιού εισαγωγής και επεξεργασίας τμήματος ενός μαθήματος.

### 3. ΑΝΑΛΥΣΗ, ΣΧΕΔΙΑΣΜΟΣ, ΥΛΟΠΟΙΗΣΗ ΒΑΣΕΩΣ ΔΕΔΟΜΕΝΩΝ

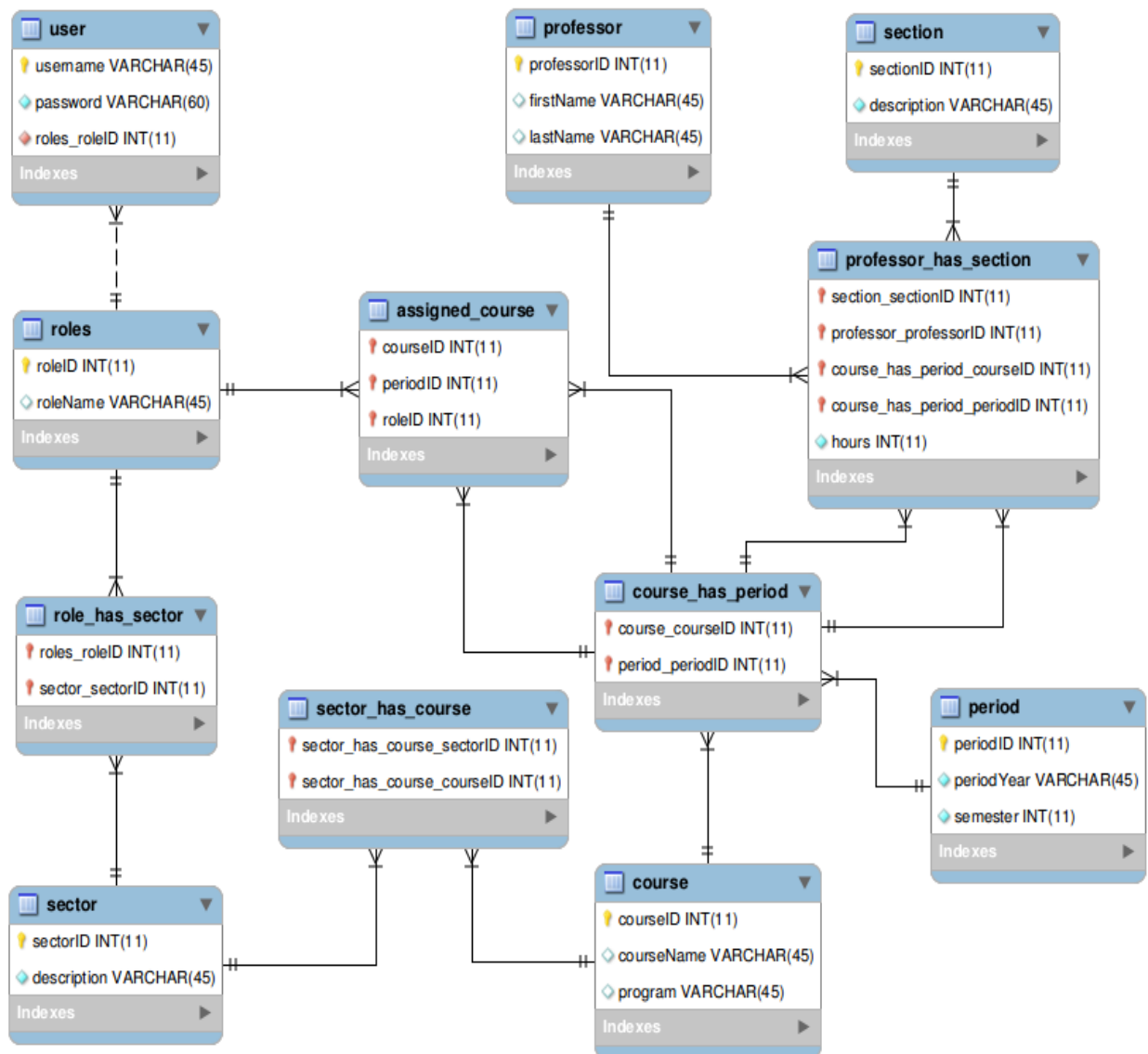
#### 3.1 Ανάλυση απαιτήσεων

Οι απαιτήσεις του συστήματος μας είναι οι εξής:

- **Απαίτηση 1:** Κάθε χρήστης έχει ένα e-mail που του έχει δοθεί από τον διαχειριστή και τον χαρακτηρίζει ένας συγκεκριμένος ρόλος.
- **Απαίτηση 2:** Οι ρόλοι των χρηστών αποτελούνται από τον διαχειριστή και τα μέλη του τμήματος.
- **Απαίτηση 3:** Οι ρόλοι έχουν διαφορετικά δικαιώματα και αντιστοιχίζονται σε έναν ή παραπάνω τομέα.
- **Απαίτηση 4:** Για να χρησιμοποιήσει ένας χρήστης την εφαρμογή πρέπει να πληκτρολογήσει το username και το password του, για να εισαχθεί στο σύστημα.
- **Απαίτηση 5:** Μόνο ο διαχειριστής του συστήματος μπορεί να κάνει εισαγωγή, επεξεργασία και διαγραφή λογαριασμού χρήστη.
- **Απαίτηση 6:** Μόνο εξουσιοδοτημένοι χρήστες, μπορούν να εισάγουν νέα μαθήματα και να τα αντιστοιχούν σε εξάμηνα.
- **Απαίτηση 7:** Μόνο εξουσιοδοτημένοι χρήστες, μπορούν να εισάγουν καθηγητές (όνομα και επωνυμο) και να δημιουργήσουν ή να επεξεργαστούν τα τμήματα του μαθήματος.
- **Απαίτηση 8:** Όλοι όσοι εισέρχονται στην εφαρμογή έχουν δικαίωμα να δουν τις αναθέσεις των μαθημάτων σε καθηγητές για όλα τα μαθήματα και όλα τα έτη.
- **Απαίτηση 9:** Κάθε μάθημα έχει ένα μοναδικό όνομα, ανήκει σε ένα μόνο πρόγραμμα σπουδών (εκτός από την περίπτωση της συνδιδασκαλίας), σε έναν ή περισσότερους τομείς και για να μπορέσουμε να του ορίσουμε τμήμα πρέπει να το αντιστοιχίσουμε πρώτα σε ένα εξάμηνο της τρέχουσας ακαδημαϊκής περιόδου.
- **Απαίτηση 10:** Σε κάθε μάθημα αντιστοιχίζονται ρόλοι υπευθύνων ανάθεσης μαθημάτων σε καθηγητές από τον διαχειριστή και την κεντρική γραμματεία.
- **Απαίτηση 11:** Κάθε ακαδημαϊκή περίοδος ξεκινά τον Σεπτέμβριο και τελειώνει τον Ιούνιο. Οι περίοδοι χωρίζονται σε εξάμηνα για τους προπτυχιακούς και χειμερινό-εαρινό για τους μεταπτυχιακούς.
- **Απαίτηση 12:** Κάθε ακαδημαϊκή περίοδος πρέπει όταν δημιουργείται, να είναι άδεια και πρέπει να εισάχθούν τα εκάστοτε μαθήματα στα εξάμηνα. Όταν δημιουργηθεί το νέο έτος, το παλαιότερο κλειδώνει και είναι προσβάσιμο μόνο για ανάγνωση.
- **Απαίτηση 13:** Κάθε μάθημα ανήκει το πολύ μια φορά σε κάθε εξάμηνο (και ακαδημαϊκό έτος).
- **Απαίτηση 14:** Σε κάθε μάθημα ανά εξάμηνο αντιστοιχίζονται τμήματα και οι καθηγητές που τα διδάσκουν καθώς και οι ώρες διδασκαλίας.
- **Απαίτηση 15:** Σε ένα μάθημα ο καθηγητής μπορεί να διδάσκει σε πολλά τμήματα αλλά όχι στο ίδιο δύο φορές (οι ώρες διδασκαλίας συναθροίζονται).

### 3.2 Μοντέλο Οντοτήτων Συσχετίσεων

Παρακάτω δίνεται το σχήμα της βάσης δεδομένων στο μοντέλο Οντοτήτων-συσχετίσεων



Εικόνα 7 :Μοντέλο Οντοτήτων-Συσχετίσεων

### 3.3 Περιγραφή σχέσεων στο σχεσιακό μοντέλο

Πίνακας 1: Assigned\_course

Assigned_course	
Int	courseID
Int	periodID
Int	roleID

Στο table Assigned\_course συνδέουμε το κάθε μάθημα ενός εξαμήνου με τους roles των χρηστών που θα είναι υπεύθυνοι ανάθεσης. Η σχέση N-M μεταξύ roles και Course\_has\_period χρησιμοποιείται για να μπορέσουμε να έχουμε για κάθε course σε κάθε χρονική περίοδο το δικό role. Έτσι επιτυγχάνουμε ευλυγισία για τις μελλοντικές αλλαγές που θα γίνουν στο πρόγραμμα σπουδών.

Οι υπεύθυνοι κάθε μαθήματος είναι πάντα οι users που έχουν ρόλο διαχειριστή, κεντρικής γραμματείας και της γραμματείας του αντίστοιχου τομέα που ανήκει το μάθημα.

Πίνακας 2: Course:

Course	
Int	courseId (Primary key)
string	courseName
string	program

Τα μαθήματα έχουν μοναδικό ID για να γίνονται αλλαγές στο όνομα του μαθήματος και πρόγραμμα σπουδών (κυρίως για διόρθωση στην περίπτωση που γίνει κάπιο λάθος). Το program (πρόγραμμα σπουδών) αντιπροσωπεύει αν το μάθημα είναι προπτυχιακό ή μεταπτυχιακό. Έγινε σαν γνώρισμα γιατί δεν υπάρχει διαφορά στην αντιμετώπισή τους από την εφαρμογή στην ανάθεση τμήματος και καθηγητή.

Πίνακας 3: Course\_has\_period

Course_has_period	
Int	course_courseID
int	period_periodID

Τα μαθήματα ορίζονται σε χρονολογικές περιόδους. Η σχέση N-M μεταξύ course και period υφίσταται για να είναι δυνατόν το κάθε μάθημα να μπορεί να εισαχθεί σε κάθε χρονολογική περίοδο. Αυτή η σχεδίαση έχει ως σκοπό την ελευθερία ανάθεσης μαθήματος σε όποιο εξάμηνο επιθυμούμε για μελλοντικές αλλαγές αλλά και την αποθήκευση της πληροφορίας για τα παλαιότερες περιόδους.

**Πίνακας 4: Period**

Period	
int	periodID
string	periodYear
int	semester

Η περίοδος χαρακτηρίζεται από ένα μοναδικό ID, την ακαδημαϊκή περίοδο που έχει την εξής μορφοποίηση «τρέχον έτος»-«επόμενο» (π.χ 2015-2016) και το εξάμηνο το οποίο είναι ένας ακέραιος αριθμός από ένα έως οκτώ. Με αυτόν τον τρόπο κρατάμε και την πληροφορία για τις παλιές περιόδους.

**Πίνακας 5: Professor**

Professor	
int	professorID
string	firstName
string	lastName

Ο κάθε καθηγητής έχει ένα μοναδικό ID για περιπτώσεις συνωνυμίας και για να μπορούν να προστεθούν και άλλα χαρακτηριστικά στο μέλλον όπως e-mail.

**Πίνακας 6: Professor\_has\_section**

Professor_has_section	
int	Section_sectionID
int	Professor_professorID
int	Course_has_period_courseID
int	Course_has_period_periodID
int	hours

Ο κάθε καθηγητής αναλαμβάνει το πολύ από ένα είδος τμήματος σε κάθε μάθημα μιας περιόδου. N-M σχέση μεταξύ του Professor και του section για να έχει ο κάθε καθηγητής πρόσβαση σε πολλά είδη τμήματος και N-M με τα δύο primary keys του Course\_has\_period θέτουμε το είδος του τμήματος και τον καθηγητή σε ένα μάθημα που έχει οριστεί σε συγκεκριμένη χρονική περίοδο. Με τις ώρες αποθηκεύουμε πόσες ώρες την εβδομάδα διδάσκει ο καθηγητής στο τμήμα.

**Πίνακας 7: Roles**

Roles	
int	roleID
string	roleName

Οι ρόλοι αντικατροπτίζουν τα δικαιώματα που έχουν οι χρήστες στο σύστημα. Αποτελούνται από το αναγνωριστικό και την περιγραφή του ρόλου.

**Πίνακας 8: Roles\_has\_sector**

Roles_has_sector	
Int	Roles_ roleID
int	Sector_sectorID

Η N-M συσχέτιση Roles με sector προσδίδει τομείς στους ρόλους. Ο κάθε ρόλος στην εφαρμογή έχει δικαιώματα για τις λειτουργίες του, στους αντίστοιχους τομείς που είναι διορισμένος.

**Πίνακας 9: Section**

Section	
Int	sectionID
String	description

Κάθε Section αναλογεί στο είδος του τμήματος που μπορεί να διδαχθεί από τους αρμόδιους. Έχουμε ένα ID σαν κύριο κλειδί και το description, σαν περιγραφή.

**Πίνακας 10: Sector**

Sector	
int	sectorID
string	description

Για κάθε τομέα αποθηκεύουμε την περιγραφή του. Στην σχολή μας έχουμε τρεις τομείς: Θεωρητική πληροφορική, Υπολογιστικά Συστήματα και Εφαρμογές και Επικοινωνίες και Επεξεργασία Σήματος. Χρησιμοποιούμε το ID ως κύριο κλειδί για λόγους απλότητας.

**Πίνακας 11: Sector\_has\_course**

Sector_has_course	
int	Sector_has_course_sectorID
int	Sector_has_course_courseID

Κάθε μάθημα μπορεί να ανήκει σε παραπάνω από έναν τομέα και κάθε τομέας μπορεί να διαθέτει περισσότερα από ένα μαθήματα εντεύθεν συνάγεται η N-M συσχέτιση μεταξύ του sector και του course. Ένα μάθημα δεν μπορεί να αλλάξει τομέα οπότε δεν συσχετίζεται με το μάθημα ανά χρονική περίοδο (Course\_has\_period), αλλά με το μάθημα (course) καθαυτό .

**Πίνακας 12: User**

User	
string	Username
string(60)	Password
int	Roles_roleID

Οι χρήστες του συστήματος καλούνται να έχουν ένα username που είναι το e-mail τους και λειτουργεί σαν αναγνωριστικό, έναν κωδικό τους που αποθηκεύεται φιλτρισμένος από μια hash συνάρτηση και έναν ρόλο από την N-1 σχέση user και role. Χρησιμοποιείται μόνο ένας ρόλος για λόγους απλότητας και ευκρίνειας καθώς ο κάθε ρόλος έχει συγκεκριμένα δικαιώματα σε όλα τα πεδία.

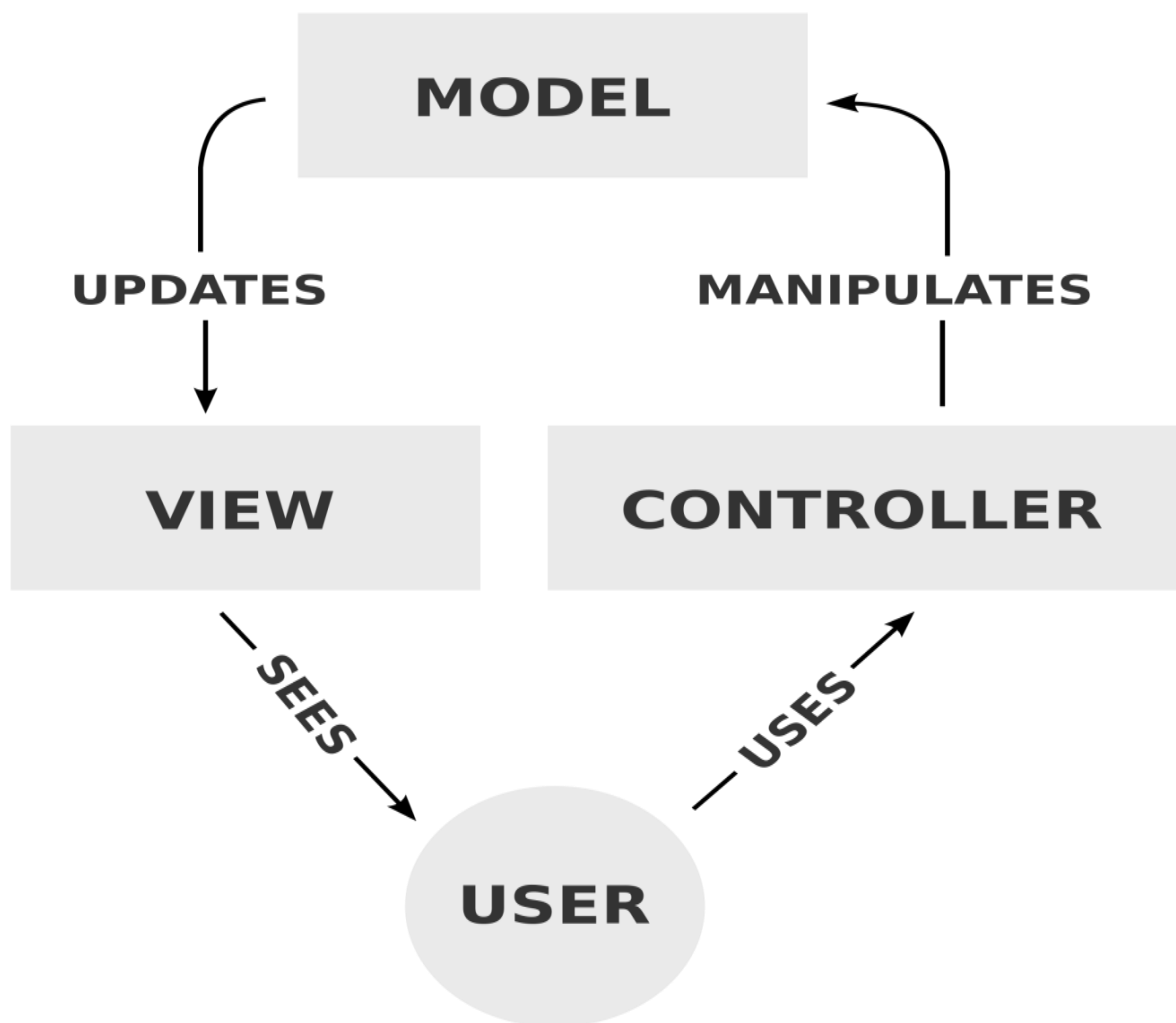


## 4. ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ

### 4.1 Αρχές σχεδιασμού και μοτίβα σχεδιασμού

Στον σχεδιασμό της εφαρμογής αποφασίσαμε ότι πρέπει να χαρακτηρίζεται από ποιότητα, λειτουργικότητα, ταχύτητα, αποδοτικότητα, ευελιξία, ασφάλεια και επεκτασιμότητα. Κρατήσαμε τις έννοιες και τις λειτουργίες της όσο πιο κοντά γίνεται στο φυσικό μοντέλο καθώς και την συνέπεια τόσο στην δομή όσο και στο τρόπο λειτουργίας της. Με τη βοήθεια του CodeIgniter πετυχαίνεται ο διαχωρισμός των αρμοδιοτήτων (separation of concerns), δηλαδή η διατήρηση της ακεραιότητας του κώδικα μίας αρμοδιότητας σε περίπτωση, όπου αλλάξει ο κώδικας μιας άλλης αρμοδιότητας. Το μοτίβο σχεδιασμού MVC πετυχαίνει αυτό το διαχωρισμό. Τέλος περιορίσαμε τα λάθη από φυσικά πρόσωπα που θα χρησιμοποιήσουν την εφαρμογή.

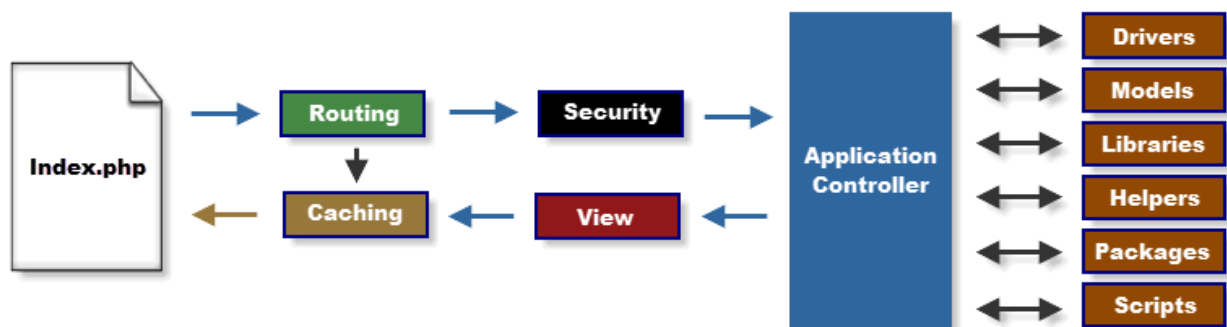
Μία τυπική υλοποίηση του MVC φαίνεται στην παρακάτω εικόνα.



Εικόνα 8: Κλασσικό MVC μοντέλο

## 4.2 Ροή δεδομένων της εφαρμογής

Η παρακάτω εικόνα δείχνει την κίνηση των δεδομένων στο σύστημα μας.



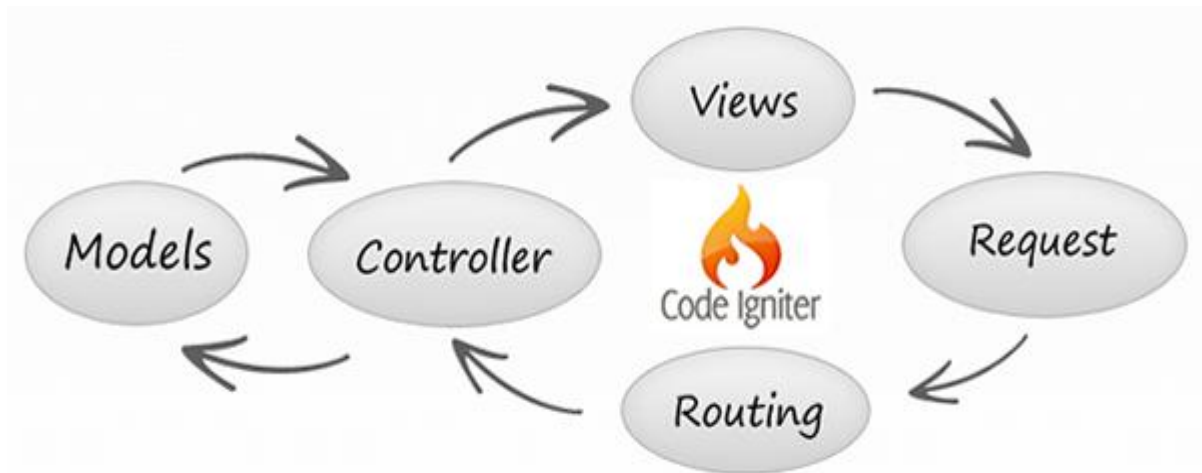
Εικόνα 9: Διάγραμμα ροής δεδομένων της εφαρμογής

Ανάλυση των βημάτων της ροής δεδομένων της εφαρμογής :

- 1) Το index.php παίζει το ρόλο του αρχικού Controller, που αρχικοποιεί τα δεδομένα που απαιτούνται για να εκκινηθεί η εφαρμογή μας.
- 2) Ο Router του CodeIgniter εξετάζει το HTTP αίτημα και αναλαμβάνει τη διαχείριση του.
- 3) Εάν υπάρχει αρχείο cache, τότε αυτό στέλνεται απευθείας στον φυλλομετρητή ιστού, παρακάμπτοντας την προκαθορισμένη διαδικασία του συστήματος.
- 4) Το επόμενο βήμα που ακολουθεί είναι η ασφάλεια. Προτού φορτωθεί ο controller της εφαρμογής, το HTTP αίτημα και όλα τα δεδομένα που εισάγει ο χρήστης από τις φόρμες περνάνε από έλεγχο και επεξεργασία για να ασφαλιστούν από κακόβουλες επιθέσεις.
- 5) Έπειτα ο Controller φορτώνει το Model, τις βασικές βιβλιοθήκες, helpers και οτιδήποτε άλλο χρειάζεται η εφαρμογή.
- 6) Έπειτα παράγεται η τελική όψη (View) και στέλνεται στον browser. Αν είναι ενεργοποιημένη η μέθοδος caching, η όψη αποθηκεύεται στην κρυφή μνήμη έτσι ώστε στα επόμενα αιτήματα να φορτωθεί αμέσως όπως προειπώθηκε στο βήμα 3.

#### 4.3 Μοτίβο σχεδιασμού - Μοντέλο Όψη Διαχειριστής (Model View Controller)

Το μοντέλο MVC είναι ένα μοτίβο σχεδίασης λογισμικού που διαχωρίζει την λογική της εφαρμογής από την παρουσίαση της. Η κεντρική ιδέα πίσω από το ΜΟΔ μοτίβο είναι η επαναχρησιμοποίηση του κώδικα και ο διαχωρισμός των αρμοδιοτήτων. Παρακάτω δείχνουμε στην εικόνα το μοντέλο MVC που ακολουθεί το CodeIgniter.



Εικόνα 10: MVC μοντέλο του CodeIgniter

##### Ανάλυση του μοτίβου MVC

1. Το Μοντέλο αποτελεί το δομικό συστατικό της online εφαρμογής, που είναι η προβολή των αντικειμένων της βάσης δεδομένων στην εφαρμογή. Ένα μοντέλο μπορεί να χρησιμοποιηθεί πολλές φορές χωρίς να χρειάζεται η επαναδημιουργία του. Τα μοντέλα έχουν απλές συναρτήσεις για επερωτήματα στην βάση δεδομένων και επικοινωνούν αποκλειστικά με τους Διαχειριστές.
2. Στην Όψη παρουσιάζεται το front-end κομμάτι της εφαρμογής. Τα πάντα που εμφανίζονται στην οθόνη των χρηστών είναι φτιαγμένα σε όψεις. Οι όψεις συνεργάζονται με τους Διαχειριστές για να κάνουν δοσοληψίες με την βάση δεδομένων όταν υπάρχει διάδραση από τον χρήστη.
3. Ο Διαχειριστής είναι ο μεσάζων που πραγματοποιεί την διασύνδεση μεταξύ Μοντέλου και Όψης. Εκεί γίνονται όλοι οι απαραίτητοι έλεγχοι και υλοποιούνται οι περισσότερες λειτουργίες της εφαρμογής.

## 5. ΑΡΧΕΙΑ ΕΦΑΡΜΟΓΗΣ

### 5.1 Εισαγωγή

Στο κεφάλαιο αυτό θα περιγράψουμε περιληπτικά την αρχιτεκτονική της εφαρμογής στο δίσκο (μέσα στο φάκελο του project) και τα σημαντικά σημεία, ώστε να είναι διαθέσιμα σε κάποιον που θέλει να επεκτείνει την εφαρμογή.

### 5.2 Αρχεία ρυθμίσεων

Η εφαρμογή συνολικά έχει τα εξής αρχεία ρυθμίσεων:

1. **autoload.php** (AssignCourses\Source Files\application\autoload.php)
2. **config.php** (AssignCourses\Source Files\application\config.php)
3. **constants.php** (AssignCourses\Source Files\application\constants.php)
4. **database.php** (AssignCourses\Source Files\application\database.php)
5. **doctypes.php** (AssignCourses\Source Files\application\doctypes.php)
6. **foreign\_chars.php** (AssignCourses\Source Files\application\foreign\_chars.php)
7. **hooks.php** (AssignCourses\Source Files\application\hooks.php)
8. **memcached.php** (AssignCourses\Source Files\application\memcached.php)
9. **migration.php** (AssignCourses\Source Files\application\migration.php)
10. **mimes.php** (AssignCourses\Source Files\application\mimes.php)
11. **profiler.php** (AssignCourses\Source Files\application\profiler.php)
12. **routes.php** (AssignCourses\Source Files\application\routes.php)
13. **smileys.php** (AssignCourses\Source Files\application\smileys.php)
14. **user\_agents.php** (AssignCourses\Source Files\application\user\_agents.php)

#### 5.2.1 autoload.php

Το code igniter φορτώνονται και να αρχικοποιεί τις επιλεγμένες βιβλιοθήκες, τους βοηθούς και τα μοντέλα να κάθε φορά που τρέχει το σύστημα.

#### 5.2.2 config.php

Περιέχει ρυθμίσεις για λειτουργίες της ιστοσελίδας που αφορούν το url, την γλώσσα, την ημερομηνία, τα αρχεία καταγραφής και τα δικαιώματα τους, ρυθμίσεις για την λειτουργία της περιόδου (session) και των cookie, προστασίας από επιθέσεις xss και csrf.

#### 5.2.3 constants.php

Εδώ υπάρχουν οι global μεταβλητές του προγράμματος.

#### 5.2.4 database.php

Εμπεριέχονται οι ρυθμίσεις για την σύνδεση του προγράμματος με την SQL βάση δεδομένων.

#### 5.2.5 doctypes.php

Καθορίζονται τα doc types που θα χρησιμοποιηθούν.

#### 5.2.6 foreign\_chars.php

Εμπεριέχονται οι χαρακτήρες για μετατροπή από το 8-bit ASCII σε αντίστοιχους των 7-bit ASCII.

### 5.2.7 hooks.php

Αρχείο που χρησιμοποιείται για την επέκταση του κώδικα του Codeigniter χωρίς να χρειάζεται η αλλαγή του πηγαίου κώδικά του.

### 5.2.8 memcached.php

Εδώ αποθηκεύονται οι διακομιστές για την αποθήκευση των cached αρχείων.

### 5.2.9 migration.php

Εδώ αποθηκεύονται οι μεταβλητές που καθορίζουν την έκδοση της βάσης δεδομένων που ενδέχεται να μεταφερθεί.

### 5.2.10 mimes.php

Εδώ περιέχεται ο πίνακας με τους τύπους των αρχείων που επιτρέπεται να μεταφορτωθούν.

### 5.2.11 profiler.php

Υπάγονται είναι οι ρυθμίσεις για την λειτουργία ανάλυσης του κώδικα (για σφάλματα).

### 5.2.12 routes.php

Εδώ αναγράφονται τα αρχεία στα οποία για την αναδρομολόγηση.

### 5.2.13 smileys.php

Εδώ γίνεται αντιστοιχία των smileys σε εικόνες.

### 5.2.14 user\_agents.php

Περιέχονται όλες οι συσκευές που αναγνωρίζονται από την εφαρμογή και ονόματα που τους αναθέτουμε για την αναφορά τους.

## 5.3 Περιγραφή αρχείων Μοντέλου (AssignCourses\application\models\)

Περιέχει όλα τα αρχεία που αλληλεπιδρούν με την βάση δεδομένων.

Πίνακας 13: Μοντέλα

<b>Course_model.php</b>	Συναρτήσεις για την εισαγωγή λήψη δεδομένων για τα μαθήματα
<b>Period_model.php</b>	Συναρτήσεις για την εισαγωγή και λήψη για δεδομένα που αφορούν την περίοδο
<b>Professor_model.php</b>	Συναρτήσεις για την εισαγωγή και λήψη δεδομένα καθηγητών
<b>Role_model.php</b>	Συναρτήσεις για την εισαγωγή και λήψη δεδομένα ρόλων
<b>Section_model.php</b>	Συναρτήσεις για την εισαγωγή και λήψη τμημάτων μαθήματος
<b>User_model.php</b>	Συναρτήσεις για την εισαγωγή και λήψη δεδομένα χρηστών του συστήματος

## 5.4 Περιγραφή αρχείων Όψης (AssignCourses\application\views\)

Τα αρχεία όψης αποτελούν το γραφικό κομμάτι που βλέπει ο χρήστης. Όλες οι αλληλεπιδράσεις του χρήστη με το σύστημα ξεκινούν από εδώ. Για κάθε πράξη του χρήστη του εμφανίζεται το αντίστοιχο αποτέλεσμα.

Πίνακας 14: Όψεις

<b>assign_course_view.php</b>	Ανάθεση υπευθύνου μαθήματος και μαθήματος στο τρέχον ακαδημαϊκό έτος.
<b>edit_user_view.php</b>	Διαχείριση στοιχείων χρήστη.
<b>login_view.php</b>	Αρχική σελίδα για είσοδο στην εφαρμογή, με αίτηση για πιστοποίηση.
<b>menu.php</b>	Το μενού πλήγησης που εισάγεται σε κάθε σελίδα.
<b>register_success_view.php</b>	Επιτυχής Εισαγωγή νέου χρήστη.
<b>register_view.php</b>	Εισαγωγή χρήστη.
<b>user_page.php</b>	Σελίδα που περιέχει όλα τα μαθήματα και τα τμήματά τους.

## 5.5 Περιγραφή αρχείων Διαχειριστή (AssignCourses\application\controllers\)

Τα αρχεία Διαχειριστή πέρνουν δεδομένα από τα αρχεία όψεως, τα ελέγχουν και τα επεξεργάζονται, επικοινωνούν μέσω των Μοντέλων με την βάση δεδομένων και επιστρέφουν το αποτέλεσμα σε μία Όψη.

Πίνακας 15: Διαχειριστές

<b>AccountController.php</b>	Δημιουργία, επεξεργασία, διαγραφή, έλεγχος των στοιχείων του λογαριασμού του χρήστη και δημιουργία συνεδρίας.
<b>CourseController.php</b>	Εισαγωγή μαθημάτων, εισαγωγή σε περίοδο, και επεξεργασία τμημάτων των μαθημάτων. ανάθεση υπευθύνων σε μαθήματα. Εισαγωγή, επεξεργασία καθηγητή.
<b>LoginController.php</b>	Μετάβαση στην σελίδα της πιστοποίησης στοιχείων.
<b>PeriodController.php</b>	Κατασκευή και ανάκτηση ακαδημαϊκών περιόδων.

## 5.6 Περιγραφή αρχείων css (AssignCourses\assets\css)

Τα αρχεία css αποδίδουν την μορφοποίησεις για την οπτική μορφή των στοιχείων των ιστοσελίδων.

Πίνακας 16: Αρχεία css

<b>Mystyles.css</b>	Για όλες τις σελίδες.
<b>Bootstrap*.css</b>	Χρησιμεύει παντού.

## 5.7 Περιγραφή αρχείων Javascript (AssignCourses\assets\js)

Τα αρχεία Javascript τα χρησιμοποιούμε για να φορτώνουμε δεδομένα και να αλλάζουμε δυναμικά το περιεχόμενο των σελίδων.

Πίνακας 17: Αρχεία Javascript

<b>app.js</b>	Αρχικό module που χρειάζεται από την AngularJS.
<b>starterController.js</b>	Controller του app.js για την εμφάνιση-απόκριση των πληροφοριών προπτυχιακών ή μεταπτυχιακών μαθημάτων στην σελίδα user_page.php
<b>courseController</b>	Controller του app.js για την φόρτωση, ανανέωση και επεξεργασία των μαθημάτων στην σελίδα user_page.php
<b>periodController.js</b>	Controller του app.js για την λειτουργία της αλλαγής της ακαδημαϊκής περιόδου στην σελίδα user_page.php
<b>bootstrap.js</b>	Αρχείο που χρειάζεται από το Bootstrap.
<b>jquery-1.11.3.js</b>	Χρήσιμο σε όλα τα views για την μεταφορά δεδομένων στους Διαχειριστές και τις δυναμικές αλλαγές στο DOM.
<b>npm.js</b>	Αυτοδημιουργούμενο αρχείο από το commonjs Grunt task.

## 6. ΠΡΟΣΤΑΣΙΑ ΚΑΙ ΑΣΦΑΛΕΙΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

### 6.1 Εισαγωγή

Η εφαρμογή διατίθεται στο διαδίκτυο με ανάγκη να έχει προδιαγραφές για την προστασία της από κακόβουλες επιθέσεις. Χρήστες και πολλά αυτοματοποιημένα προγράμματα σαρώνουν συνεχώς το διαδίκτυο προσπαθώντας να αποκτήσουν πρόσβαση σε ευαίσθητα δεδομένα εκμεταλλευόμενοι ευάλωτο κώδικα. Καθώς υπάρχουν πολλές εξειδικευμένες επιθέσεις που απαιτούν συνεχή παρακολούθηση και συντήρηση της εφαρμογής, μπορούμε να την προστατεύσουμε από τις πιο συχνές και τις πιο «δημοφιλείς» επιθέσεις.

### 6.2 Κακόβουλες διαδικτυακές επιθέσεις

Παρακάτω ακολουθούν μερικές από τις πιο σημαντικές κακόβουλες επιθέσεις.

#### 6.2.1 Ένεση SQL (SQL Injection)

Η επίθεση γίνεται αλλάζοντας δεδομένα που εισάγει ο χρήστης για να παρερμηνευτούν και να μπορέσει να αποσπάσει πληροφορίες. Μέσω αυτής της επίθεσης, μπορεί κάποιος κακόβουλος επιτιθέμενος να εκτελέσει δικά του επρωτηήματα SQL στον server, με σκοπό να αποκτήσει πρόσβαση σε ευαίσθητα δεδομένα της βάσης.

Τρόποι αντιμετώπισης :

- Όλα τα επρωτηήματα της βάσεως δεδομένων πραγματοποιούνται στον rhr διακομιστή έτσι ώστε να μην υπάρχει πρόσβαση από τον χρήστη.
- Επίσης με την βοήθεια του CodeIgniter, χρησιμοποιούμε την μέθοδο του δεσμευμένου επρωτηήματος (query binding) που καθαρίζει τα δεδομένα που εισάγει ο χρήστης από τις φόρμες (escape user input), παράγοντας έτσι ασφαλέστερα επρωτηήματα.

#### 6.2.2 Διασταύρωση κώδικα μεταξύ ιστοσελίδων (Cross Site Scripting, XSS)

Η επίθεση εκμεταλλεύεται την δυνατότητα να ανοίξει ο επιτιθέμενος δίαυλο επικοινωνίας για να εισχωρήσει δικό του κώδικα στο πρόγραμμα, κάτι που μπορεί να είναι καταστροφικό για μία εφαρμογή και για τον server.

Τρόποι αντιμετώπισης :

- Το CodeIgniter χρησιμοποιεί την τεχνική του URI Routing. Είναι αρκετά αυστηρό σε ότι σχετίζεται με τους επιτρεπόμενους χαρακτήρες στις URI συμβολοσειρές, για να μειώσει την πιθανότητα να εισάγει κάποιος κακόβουλα δεδομένα.
- Επιπλέον προστατεύουμε την εφαρμογή φιλτράροντας κάθε POST, SESSION και COOKIE μεταβλητή με το φίλτρο που παρέχεται από το



CodeIgniter την `xss_clean`. Δεν χρησιμοποιούμε άλλες ευάλωτες μεταβλητές. Επιπλέον έχουμε ενεργοποιήσει και την `global` μεταβλητή του CodeIgniter που μας εξασφαλίζει προστασία σε όλο το πρόγραμμα.

### **6.2.3 Πλαστογράφηση αίτηση μεταξύ ιστοσελίδων (Cross Site Request Forgery, CSRF)**

Η επίθεση εκμεταλεύεται την ύπαρξη μιας συνεδρίας για να εισβάλει στα δεδομένα του χρήστη. Το ιδιαίτερο στοιχείο της είναι ότι δεν φανερώνεται τίποτα στον χρήστη, απλά εκτελείται ένα κομμάτι κώδικα που υπάρχει σε μια άλλη ιστοσελίδα.

Τρόπος αντιμετώπισης : Ενεργοποιούμε την επιλογή που μας παρέχει το CodeIgniter μέσα από τον `config.php` αρχείο για να διασφαλίσει όλη την εφαρμογή από τέτοιου είδους επιθέσεις.

### **6.2.4 Εισχώρηση τοπικού αρχείου / Εισχώρηση απομακρυσμένου αρχείου (Local File Incursion - LFI, Remote File Incursion – RFI)**

Οι επιθέσεις είναι επικίνδυνες όταν μπορεί ο χρήστης να ανεβάσει ένα αρχείο στο σύστημα ή η εφαρμογή να συμπεριλαμβάνει ένα αρχείο από τον χρήστη που περιέχει κακόβουλο κώδικα.

Τρόπος αντιμετώπισης : Η εφαρμογή δεν επιτρέπει το ανέβασμα των αρχείων, και δεν χρησιμοποιεί κανένα αρχείο που να ορίζεται από τους χρήστες.

### **6.2.5 Υφαρπαγή συνόδου (Session Hijacking)**

Ο επιτιθέμενος υφαρπάζει ή ανακτά τον έλεγχο μιας συνόδου που δημιουργήθηκε μεταξύ ενός χρήστη και του διακομιστή. Αυτό μπορεί να συμβεί είτε σε επίπεδο του δικτύου είτε σε επίπεδο της εφαρμογής. Υπάρχουν πολλά έτοιμα εργαλεία που υλοποιούν και εκτελούν αυτόματα τέτοιες επιθέσεις.

Όσον αφορά τη προστασία μας από αυτές τις επιθέσεις καθώς μέσα από το `config.php` αρχείο έχουμε ενεργοποιήσει τις εξής ρυθμίσεις :

1. Κρυπτογράφηση του αναγνωριστικού της συνόδου (cookie encryption),
2. Προσδιορίσει χρόνο τερματισμού αυτής (session timeout)
3. Δυνατότητα επαλήθευσης διεύθυνσης δικτύου (IP matching).

### **6.2.6 Υποκλοπή ευαίσθητων δεδομένων της βάσης δεδομένων**

Στις βάσεις δεδομένων στις οποίες αποθηκεύονται ευαίσθητα δεδομένα των χρηστών, χρειάζεται να προσεχθεί ιδιαίτερα ο τρόπος και η μορφή αποθήκευσης των δεδομένων. Αν κάποιος κακόβουλος επιτιθέμενος αποκτήσει πρόσβαση στη βάση και δει τους κωδικούς των χρηστών στη βάση, τότε έχει αποκτήσει αμέσως πρόσβαση σε όλους τους λογαριασμούς τους.

Ακολουθήσαμε την εξής αντιμετώπιση για αυτή τη περίπτωση :

Αντί να αποθηκεύσουμε τους κωδικούς των χρηστών σε μορφή plain text στη βάση, χρησιμοποιήσαμε τον ασφαλέστερο αλγόριθμο κρυπτογραφίας bcrypt.

## **7. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ**

Η εφαρμογή που αναπτύχθηκε δίνει την δυνατότητα της διαχείρισης των αναθέσεων μαθημάτων για το Τμήμα Πληροφορικής και Τηλεπικοινωνιών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών. Επιτυγχάνει την άρτια συνεργασία των μελών-χρηστών του Τμήματος διαχωρίζοντας τις εργασιακές αρμοδιότητές τους, μέσα σε ένα εύχρηστο, φιλικό, διαδραστικό και ασφαλές περιβάλλον του συστήματος.

Η κυριότερη δυσκολία που αντιμετωπίστηκε ήταν η πλήρης και σαφής διατύπωση των απαιτήσεων του συστήματος από την αρχή της διεργασίας ανάπτυξης της εφαρμογής. Αυτό είχε σαν αποτέλεσμα την δημιουργία προβλημάτων, λόγω απρόβλεπτων αλλαγών, που επιλύθηκαν με την άριστη συνεργασία του επιβλέπων Επικουρου Καθηγητή κ. Παναγιώτη Σταματόπουλο.

Επειδή χρησιμοποιείται η αρχιτεκτονική σχεδίασης MVC μπορούν άμεσα να εισαχθούν και να αλληλεπιδράσουν νέες οντότητες και λειτουργίες με το σύστημα. Η χρήση της JQuery επιτρέπει την πρόσθεση νέων επεκτάσεων στο σύστημα. Η εφαρμογή με λίγες τροποποιήσεις μπορεί να χρησιμοποιηθεί και από άλλα Τμήματα του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών.

## ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

PHP	<i>Hypertext Preprocessor</i>
JS	Javascript
Angular.js	AngularJS
Ajax	Asynchronous JavaScript and XML
JavaEE	Java Platform, Enterprise Edition
JSP	JavaServer Pages
JSF	JavaServer Faces
JPA	Java Persistence API
MSDNAA	Microsoft Developers Network Academic Alliance
Apache	Apache HTTP Server
XAMPP	X (Cross-Platform) Apache HTTP Server MySQL or MariaDB PHP Perl
URI	Uniform Resource Identifier
HTML	HyperText Markup Language
SQL	Structured Query Language
MySqli	MySQL Improved Extension
Dom	Document Object Model
CSS	Cascading Style Sheets
XML	Extensible Markup Language
URL	Uniform Resource Locator
MVC	Model–view–controller
HTTP	Hypertext Transfer Protocol
Csrf	Cross-Site Request Forgery
Xss	Cross-site scripting
ASCII	American Standard Code for Information Interchange
LFI	Local File Inclusion
RFI	Remote File Inclusion
IP	Internet Protocol address

## ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Administrator	Διαχειριστής
navigation bar	Μπάρα πλοήγησης
Log-in	Σύνδεση
User registration	Εγγραφή χρήστη
Frameworks	Πλαίσιο
Model View Controller	Μοντέλο-Όψη-Ελεγκτήρας
Separation of concepts	Διαχωρισμός Αρμοδιοτήτων
Single Responsibility design principle	Αρχή σχεδιασμού της απλής υπευθυνότητας
Scalability	Επεκτασιμότητα
Development Time	Χρόνος ανάπτυξης
Software Maintenance and Testing	Συντήρηση Λογισμικού και Δοκιμές
Third party tools	Εργαλεία τρίτων
Documentation	Τεκμηρίωση
Simplicity	Απλότητα
Easy Learning	Ευκολία εκμάθησης
setup and deployment	εγκατάσταση και ανάπτυξη
web browser	φυλλομετρητής
redployment	ανασύνταξη
refactoring	Ανακατασκευή
templates	σχεδίοτυπο
open-source	Ανοικτού κώδικα
cross-platform	ανεξάρτητο πλατφόρμας
widgets	Προκατασκευασμένες γραφικές εφαρμογές
EER diagram	Επεκταμένο μοντέλο οντοτήτων συσχετίσεων
Performance	Απόδοση
Easy Startup	Εύκολη εκκίνηση
integration with third-party tools	Ενσωμάτωση εργαλεία τρίτων
Configuration file	Αρχείο ρυθμίσεων
easy learning	Εύκολη εκμάθηση
Email sender	Αποστολέας email
Security libraries	Βιβλιοθήκες ασφαλείας
Zip compression	Συμπίεση αρχείων
Calendar	Υπηρεσία ημερολόγιου
Session handling	Διαχείριση συνόδων
clear documentation	Καθαρό κείμενο τεκμηρίωσης
Plugins	άρθρωμα
hosts	εξυπηρετητές
servers	διακομιστής
relational database management system	Σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων
System	Σύστημα
tags	Ετικέτα
link	Σύνδεσμος
table	Πίνακας
button	Κουμπί
Events	Συμβάν
e-mail	Ηλεκτρονικό ταχυδρομείο
username	Όνομα χρήστη
password	Κωδικός

hash	Κατακερματισμός
View	Όψη
Cache	κρυφή μνήμη
Browser	πρόγραμμα περιήγησης
Controller	ελεγκτήρας
front-end	εμπρόσθιο τμήμα
back-end	οπίσθιο τμήμα
project	Έργο
session	σύνοδος
smileys	Χαμογελαστά πρόσωπα
query	ερώτημα
binding	δέσμευση
escape	Διαφυγή
user	Χρήστης
input	Είσοδος
Routing	δρομολόγηση
POST	Τοποθέτηση
COOKIE	μπισκοτάκι
Encryption	Κρυπτογράφηση
Time-out	εξωχρονισμός
matching	αντιστοίχιση

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] I. Sommerville, *Βασικές Αρχές τεχνολογίας λογισμικού*, Κλειδάριθμος, 2009
- [2] R. Elmasri, S. B. Navathe, *Θεμελιώδεις αρχές συστημάτων βάσεων δεδομένων*, Δίαυλος, 2012
- [3] Ν. Αβούρης, *Εισαγωγή στην επικοινωνία ανθρώπου-υπολογιστή*, Δίαυλος, 2000
- [4] C. Lindley, *jQuery Cookbook*, O'Reilly Media, 2009
- [5] L. Welling, L. Thomson, *PHP and MySQL Web Development (4th Edition)*, Addison-Wesley Professional, 2008
- [6] I. Alshanetsky, R. Lerdorf, *php|architect's Guide to PHP Security* (PHP Architect Nanobooks), musketeers.me, LLC, 2005
- [7] C. Snyder, M. Southwell, *Pro PHP Security*, Apress, 2006
- [8] R. Nixon, *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*, O'Reilly, 2014
- [9] Simon. "Security checklist for a CodeIgniter application launch." *Knowledge markets*: <http://stackoverflow.com/questions/14429802/security-checklist-for-a-codeigniter-application-launch>. Η καταχώρηση δημοσιεύτηκε στις 20 Ιανουαρίου 2013. [Προσπελάστηκε 17/3/2016]
- [10] Red. "Codeigniter form validation security." *Knowledge markets*: <http://stackoverflow.com/questions/13486806/codeigniter-form-validation-security?rq=1>. Η καταχώρηση δημοσιεύτηκε στις 21 Νοεμβρίου 2012. [Προσπελάστηκε 17/3/2016]
- [11] Bruno Skvorc. "The Best PHP Framework for 2015: SitePoint Survey Results." online learning platform for web developers: <http://usawc.libguides.com/content.php?pid=291999&sid=2397891>. Η καταχώρηση δημοσιεύτηκε στις 30 Μαρτίου 2015. [Προσπελάστηκε 17/3/2016]
- [12] Seralo. "PHP frameworks comparison." *Social Comparison*: <http://socialcompare.com/en/comparison/php-frameworks-comparison>. Η καταχώρηση δημοσιεύτηκε στις 14 Μαρτίου 2010. [Προσπελάστηκε 17/3/2016]
- [13] Adam. "What is the best way to prevent session hijacking?." *Knowledge markets*: <http://stackoverflow.com/questions/22880/what-is-the-best-way-to-prevent-session-hijacking>. Η καταχώρηση δημοσιεύτηκε στις 22 Αυγούστου 2008. [Προσπελάστηκε 17/3/2016]
- [14] Marcela De Vivo. "Essentials For Creating A User-Friendly Interface." community for web designers and developers: <http://www.instantshift.com/2013/08/14/creating-user-friendly-interface/>. 14 Αυγούστου 2013. [Προσπελάστηκε 17/3/2016]
- [15] Xampp tutorial, <https://www.apachefriends.org/> [Προσπελάστηκε 17/3/2016]
- [16] CodeIgniter documentation, <https://www.codeigniter.com/> [Προσπελάστηκε 17/3/2016]
- [17] Sql tutorial, [www.w3schools.com/sql/](http://www.w3schools.com/sql/) [Προσπελάστηκε 17/3/2016]
- [18] JQuery documentation, <http://api.jquery.com/> [Προσπελάστηκε 17/3/2016]
- [19] X-Editable documentation, <https://vitalets.github.io/x-editable/> [Προσπελάστηκε 17/3/2016]
- [20] AngularJs Documentation, <https://angularjs.org/> [Προσπελάστηκε 17/3/2016]

- [21] PHP examples and documentation, <http://php.net/> [Προσπελάστηκε 17/3/2016]
- [22] PHP examples and documentation, [www.w3schools.com/php/](http://www.w3schools.com/php/) [Προσπελάστηκε 17/3/2016]
- [23] Open Web Application Security Project, <https://www.owasp.org/> [Προσπελάστηκε 17/3/2016]
- [24] Html examples and documentation, [www.w3schools.com/html/](http://www.w3schools.com/html/) [Προσπελάστηκε 17/3/2016]
- [25] CSS examples and documentation, [www.w3schools.com/css/](http://www.w3schools.com/css/) [Προσπελάστηκε 17/3/2016]
- [26] *Bootstrap* examples and documentation, [www.w3schools.com/bootstrap/](http://www.w3schools.com/bootstrap/) [Προσπελάστηκε 17/3/2016]
- [27] *Bootstrap* examples and documentation, <http://getbootstrap.com/> [Προσπελάστηκε 17/3/2016]
- [28] Javascript examples and documentation, [www.w3schools.com/js/](http://www.w3schools.com/js/) [Προσπελάστηκε 17/3/2016]
- [29] Ελληνική εταιρεία ορολογίας, [www.eleto.gr/](http://www.eleto.gr/) [Προσπελάστηκε 17/3/2016]
- [30] PHP practices, <http://www.phptherightway.com/> [Προσπελάστηκε 17/3/2016]