



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής  
Πρόγραμμα Μεταπτυχιακών Σπουδών στην  
«Πληροφορική»

**Μεταπτυχιακή Διατριβή**

Τίτλος Διατριβής	Διαδικτυακή εφαρμογή εκμάθησης της αρχαίας ελληνικής γλώσσας με εργαλείο δημιουργίας περιεχομένου  A web-based application for learning ancient Greek supported by an integrated authoring tool
Όνοματεπώνυμο Φοιτητή	Πιέρρος Αναϊπάκος
Πατρώνυμο	Ιωάννης
Αριθμός Μητρώου	ΜΠΠΛ 12005
Επιβλέπων	Μαρία Βίρβου, Καθηγήτρια

# Περιεχόμενα

1. Εισαγωγή.....	3
2. Περιγραφή της εφαρμογής.....	4
3. Το μοντέλο RUP.....	5
4. Περιγραφή του πληροφοριακού συστήματος.....	6
4.1. Διάγραμμα περιπτώσεων χρήσης(Use Case Diagram).....	6
4.2. Διάγραμμα κλάσεων(Class Diagram).....	10
4.3. Διάγραμμα καταστάσεων(State Machine Diagram).....	11
4.4. Διάγραμμα δραστηριοτήτων(Activity Diagram).....	13
4.5. Διάγραμμα συνεργασίας(Collaboration Diagram).....	15
4.6. Διάγραμμα ανάπτυξης(Deployment Diagram).....	16
4.7. Διάγραμμα πακέτων(Package Diagram).....	17
5. Ανάλυση απαιτήσεων.....	18
5.1. Σχεδιασμός.....	21
5.2. Έλεγχος.....	22
6. Εγχειρίδιο χρήσης.....	25

## 1. Εισαγωγή

Η αλματώδης και συνεχής ανάπτυξη της τεχνολογίας που παρατηρείται τις τελευταίες δεκαετίες και αποσκοπεί στην διευκόλυνση των ανθρώπινων και καθημερινών αναγκών έχει ως αποτέλεσμα τη χρήση υπολογιστικών συστημάτων από όλους τους ανθρώπους ανεξαρτήτως ηλικίας.

Από τα πρώτα κιόλας χρόνια ενός παιδιού χρησιμοποιούνται συστήματα τα οποία απασχολούν και ταυτόχρονα μαθαίνουν-εκπαιδεύουν τα παιδιά εισάγοντας τα κλιμακωτά σε έναν "ηλεκτρονικό" κόσμο.

Στην πτυχιακή αυτή εργασία θα σχεδιαστεί, δημιουργηθεί και υλοποιηθεί ένα διαδικτυακό εκπαιδευτικό μοντέλο εκμάθησης της αρχαίας ελληνικής σύνταξης και γραμματικής που στόχο έχει να εξαίρει στους μαθητές τη μαγεία της μητρικής τους γλώσσας και να αντιληφθούν πως υπήρξε βάση για πολλές άλλες. Σκοπός επίσης της εφαρμογής είναι να προσφέρει τη δυνατότητα και την εξοικείωση σε όσους τη χρησιμοποιήσουν με τις νέες τεχνολογίες μετατρέποντας τη "βαρετή" ώρα του διαβάσματος σε πιο ενδιαφέρουσα, διαδραστικότερη και περισσότερο αποδοτική.

## **2. Περιγραφή της εφαρμογής**

Η εφαρμογή μπορεί να χρησιμοποιηθεί από δύο διαφορετικούς ρόλους, τους μαθητές και τους καθηγητές. Οι μαθητές αρχικά έχουν τη δυνατότητα να

συμπληρώσουν μια φόρμα με τα προσωπικά τους στοιχεία και στη συνέχεια χρησιμοποιώντας ορισμένα από αυτά, όπως για παράδειγμα το ηλεκτρονικό ταχυδρομείο και τον κωδικό εισαγωγής, να εισέρχονται στην εφαρμογή. Πριν γίνει η εισαγωγή πραγματοποιείται υποχρεωτική ταυτοποίηση των στοιχείων. Εφόσον ολοκληρωθεί επιτυχώς η ταυτοποίηση οι μαθητές αρχικά παρακολουθούν σε θεωρητικό επίπεδο τη διδακτέα ύλη και στο τέλος κάθε θεματικής ενότητας προβαίνουν στη συμπλήρωση ενός ερωτηματολογίου σχετικό με την εκάστοτε ύλη. Το ερωτηματολόγιο περιλαμβάνει ερωτήσεις πολλαπλής επιλογής και σωστού λάθους. Με την ολοκλήρωση της διαδικασίας ο μαθητής γνωρίζει ποιες απαντήσεις ήταν σωστές, ποιες λάθος και ποιο σημείο της ύλης πρέπει να ξαναδιαβάσει για να καλύψει τα κενά του. Στο τέλος του διαγωνίσματος θα βγαίνει ένα ποσοστό το οποίο θα επιτρέπει στο μαθητή είτε να προχωρήσει στη μελέτη της επόμενης ενότητας είτε θα τον παραπέμπει στο κομμάτι εκείνο της ύλης που πρέπει να ξαναμελετήσει ώστε να επαναλάβει το διαγώνισμα.

Από την άλλη πλευρά, οι καθηγητές λειτουργούν ως "διαχειριστές" του συστήματος. Εισάγουν και εκείνοι τα στοιχεία τους στο σύστημα και ακολουθείται αντίστοιχα η διαδικασία της ταυτοποίησης με αυτή των μαθητών. Μέσα στις αρμοδιότητες τους είναι να εισάγουν νέες θεματικές ενότητες ή αντίστοιχα να τις αφαιρούν. Επίσης παρακολουθούν την καρτέλα του κάθε μαθητή με τις βαθμολογίες του τόσο στις ερωτήσεις πολλαπλής επιλογής όσο και στις ερωτήσεις σωστού λάθους.

### **3. Το μοντέλο RUP**

Το μοντέλο Rational Unified Process είναι η διαδικασία που προτείνουν ο Booch, Rumbaugh και Jacobson για την ανάπτυξη λογισμικού. Το μοντέλο αυτό προτείνει ο κύκλος ζωής λογισμικού να είναι επαναληπτικός. Η ανάπτυξη δηλαδή να προχωρά σε μια σειρά επαναλήψεων μέχρι να εξελιχθεί το τελικό προϊόν. Η διαδικασία αυτή αποτελείται από ένα σύνολο οδηγιών σχετικά με τις τεχνικές και οργανωτικές απόψεις της ανάπτυξης Λογισμικού και αφορά κυρίως την Ανάλυση Απαιτήσεων και το Σχεδιασμό.

Είναι δομημένη σε δύο διαστάσεις :

1) Χρόνο - Χωρισμός του κύκλου ζωής σε φάσεις και επαναλήψεις.

## 2) Τμήματα διαδικασίας - Καλά ορισμένες εργασίες.

Η δόμηση έργου σύμφωνα με τη διάσταση των τμημάτων διαδικασίας περιλαμβάνει τις ακόλουθες δραστηριότητες:

- 1) Σύλληψη απαιτήσεων (Requirements capture): Μια αφήγηση του τι πρέπει να κάνει το σύστημα.
- 2) Ανάλυση και σχεδιασμός (Analysis and design): Μια περιγραφή του πως θα υλοποιηθεί το σύστημα.
- 3) Υλοποίηση (Implementation): Η παραγωγή του κώδικα.
- 4) Έλεγχος (Test): Η επαλήθευση του συστήματος.

Τα τμήματα της διαδικασίας και οι φάσεις στις οποίες αφιερώνουν περισσότερο χρόνο είναι:

- Μοντελοποίηση επιχείρησης: έναρξη και εκπόνηση μελέτης
- Σύλληψη απαιτήσεων: εκπόνηση μελέτης
- Ανάλυση και σχεδιασμός: εκπόνηση μελέτης και κατασκευή
- Υλοποίηση: κατασκευή
- Έλεγχος: εκπόνηση μελέτης, κατασκευή και μεταβίβαση

- Διανομή: μεταβίβαση
- Βοηθητικά τμήματα: σταθερά σε όλες τις φάσεις
- Διαμόρφωση και διαχείριση αλλαγών: κατασκευή και μεταβίβαση
- Διαχείριση: με μερικές αυξομειώσεις σταθερά και στις τέσσερις φάσεις
- Περιβάλλον: έναρξη

## **4. Περιγραφή πληροφοριακού συστήματος**

Στην παράγραφο αυτή της εργασίας θα περιγραφεί το πληροφοριακό σύστημα που αναφέρθηκε αναλυτικά παραπάνω και αφορά τις διεργασίες των ρόλων. Παρακάτω θα αναπαρασταθούν γραφικά οι παραπάνω συσχετίσεις μέσω διαγραμμάτων στη γλώσσα μοντελοποίησης UML.


### **4.1 Διάγραμμα περιπτώσεων χρήσης (User case diagram)**





Στο διάγραμμα περιπτώσεων χρήσης, πρέπει να δοθεί έμφαση στη λειτουργικότητα του πληροφοριακού συστήματος, έτσι όπως είναι ορατό από τους εξωτερικούς χρήστες. Στην περίπτωσή μας οι χειριστές του συστήματος, οι οποίοι είναι οι εκάστοτε μαθητές ή καθηγητές, προβαίνουν σε μία σειρά



ενεργειών, οι οποίες τους αποδίδουν κάποιο αποτέλεσμα που έχει αξία για εκείνους.

Παρακάτω παρατίθεται πίνακας με την περιγραφή των δομικών στοιχείων του Διαγράμματος Χρήσης:

Όροι	Περιγραφή	Σχήμα
Περίπτωση χρήσης (use case)	<p>Περιγράφει ποια είναι η λειτουργία ενός συστήματος /κλάσης/ διαπροσωπίας</p> <p>Είναι ένα σύνολο από δραστηριότητες που επιτελούνται από το σύστημα ώστε να παραχθεί κάποια συμπεριφορά που είναι αισθητή από κάποιο δράστη.</p>	

Δράστης (actor)	Ένα σύνολο από συναφείς ρόλους που παίζουν οι «χρήστες» του συστήματος κατά τη διάρκεια των περιπτώσεων χρήσης. Οι δράστες ενεργούν έξω από το σύστημα.	 ActorName
Όριο του συστήματος (system boundary)	Ορίζει το όριο το οποίο διαχωρίζει το σύστημα από τους δράστες που αλληλεπιδρούν με το σύστημα.	
Εταιρικότητα (association)	Η συμμετοχή και ο τρόπος επικοινωνίας ενός δράστη με μία περίπτωση χρήσης.	
Γενίκευση (generalization)	Μια σχέση ταξινόμησης ανάμεσα σε μια γενική περίπτωση και μια πιο ειδικευμένη περίπτωση χρήσης.	

<p>Επέκταση (extend)</p>	<p>Μια σχέση ανάμεσα σε περιπτώσεις χρήσης που ορίζει εναλλακτικούς τρόπους διαχείρισης της βασικής περίπτωσης χρήσης</p>	<p><code>&lt;&lt;extend&gt;&gt;</code> -----&gt;</p>
<p>Συνυπολογισμός (include)</p>	<p>Μια σχέση ανάμεσα σε μια βασική και μια ή περισσότερες μερικές περιπτώσεις χρήσης που ορίζει ότι η βασική περίπτωση χρήσης απαιτεί τη λειτουργία των μερικών περιπτώσεων χρήσης.</p>	<p><code>&lt;&lt;include&gt;&gt;</code> -----&gt;</p>

Για το ρόλο του καθηγητή έχουμε τις ακόλουθες περιπτώσεις χρήσης:

Use Case: Εισαγωγή στο σύστημα

Ο καθηγητής έχει τη δυνατότητα να μπαίνει στο σύστημα, να διαχειρίζεται τις καρτέλες των μαθητών και να οργανώνει την ύλη του μαθήματος είτε εισάγοντας είτε αφαιρώντας θεματικές ενότητες.

Use Case: Διαχείριση καρτέλας χρηστών(μαθητών)

Ο καθηγητής έχει τη δυνατότητα να παρακολουθεί τις καρτέλες των μαθητών με τις επιμέρους βαθμολγίες.

Use Case: Διαχείριση θεματικών ενοτήτων

Ο καθηγητής έχει τη δυνατότητα να προσθέτει, να συμπληρώνει και να διαγράφει κατάλληλο υλικό για τη διδασκαλία του μαθήματος.

Για το ρόλο του μαθητή έχουμε τις ακόλουθες περιπτώσεις χρήσης:

Use Case: Εισαγωγή στο σύστημα

Ο μαθητής έχει τη δυνατότητα να μπει στο σύστημα ώστε να διαβάσει τη διδακτέα ύλη, να εξεταστεί-αξιολογηθεί και να παρακολουθεί το αρχείο δραστηριοτήτων του.

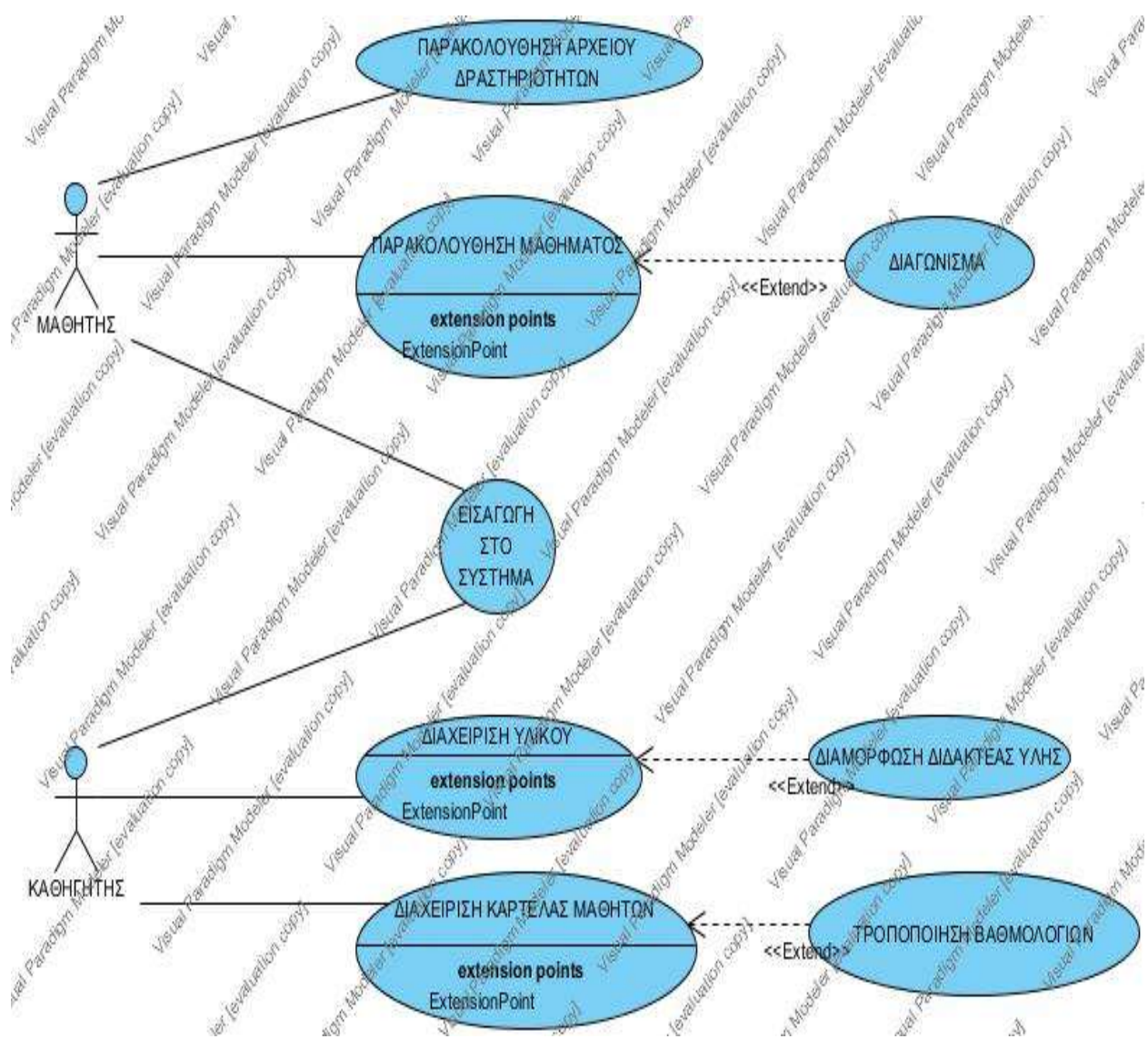
Use Case: Παρακολούθηση μαθημάτων

Ο μαθητής έχει τη δυνατότητα να παρακολουθήσει σε θεωρητικό επίπεδο θεματικές ενότητες και αφού ολοκληρώσει αυτή τη διαδικασία να εξεταστεί και να αξιολογηθεί. Εφόσον πληροί τις προϋποθέσεις(συγκεντρώνοντας ποσοστό από 80% και άνω στις δύο κατηγορίες των διαγωνισμάτων), συνεχίζει στην επόμενη ενότητα αλλιώς επαναλαμβάνει το διάβασμα και το διαγώνισμα.

Use Case: Παρακολούθηση αρχείου δραστηριοτήτων

Ο μαθητής έχει τη δυνατότητα να παρακολουθεί το αρχείο δραστηριοτήτων με τις στατιστικές του επιδόσεις.

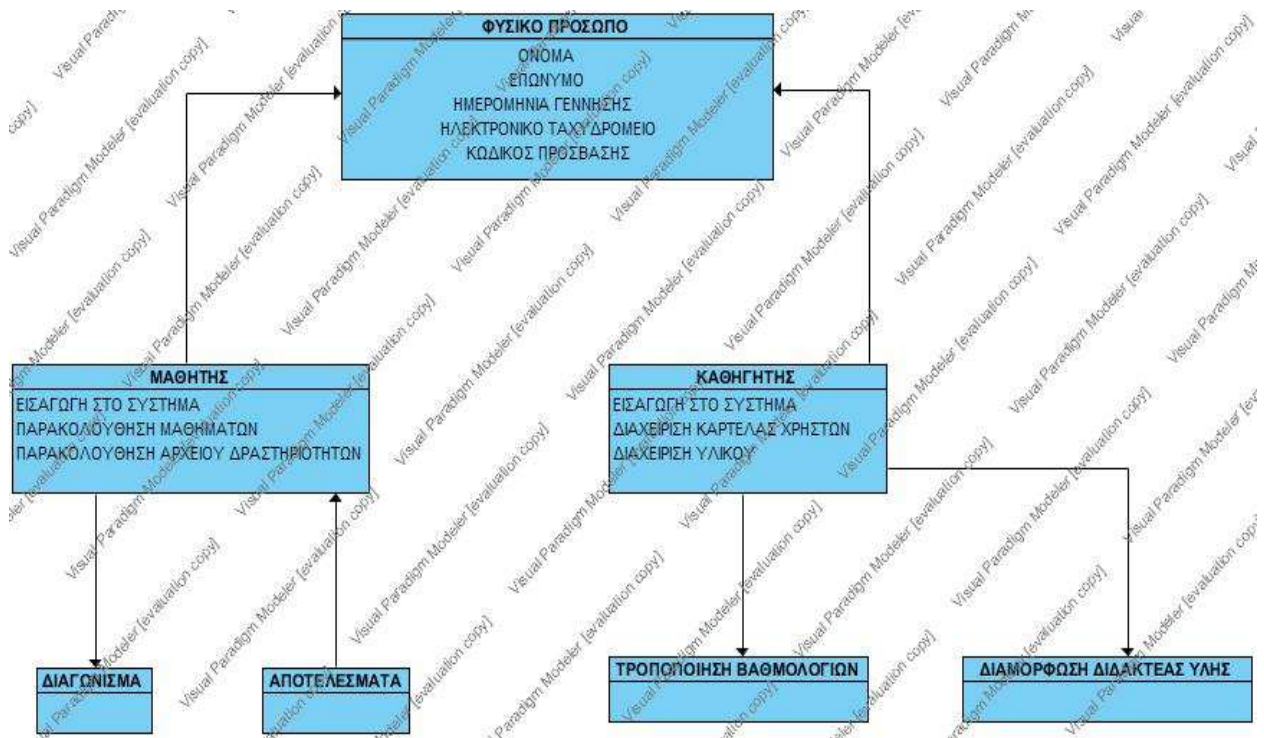
Επομένως το διάγραμμα περιπτώσεων χρήσης που σχηματίζεται είναι το ακόλουθο:



## 4.2. Διάγραμμα κλάσεων (Class diagram)

Για την υλοποίηση του διαγράμματος κλάσεων είναι χρήσιμο να καταγράψουμε τις κλάσεις του πληροφοριακού μας συστήματός και τις μεταξύ τους συσχετίσεις. Αρχικά έχουμε την κλάση «Φυσικό πρόσωπο», με χαρακτηριστικά τα προσωπικά στοιχεία των φυσικών προσώπων, είτε αυτά είναι μαθητές είτε αυτά είναι καθηγητές. Τέτοια στοιχεία είναι το όνομα, το επώνυμο, η ημερομηνία γέννησης, το ηλεκτρονικό ταχυδρομείο και ο κωδικός πρόσβασης. Η κλάση «Φυσικό Πρόσωπο» έχει δύο υποκλάσεις, την κλάση «Μαθητής» και την κλάση «Καθηγητής». Τα επιμέρους στοιχεία της κλάσης "Μαθητής" είναι η εισαγωγή στο σύστημα, η παρακολούθηση μαθημάτων και η παρακολούθηση του αρχείου δραστηριοτήτων. Επιπλέον, τα επιμέρους στοιχεία της κλάσης "Καθηγητής" είναι η εισαγωγή στο σύστημα, η διαχείριση καρτέλας των μαθητών και η διαχείριση της διδακτέας ύλης. Αυτές οι δύο υποκλάσεις κληρονομούν τα χαρακτηριστικά της μητρικής τους κλάσης αλλά έχουν και τα δικά τους χαρακτηριστικά.

Ακολουθεί το διάγραμμα κλάσεων με τις κατάλληλες συσχετίσεις:

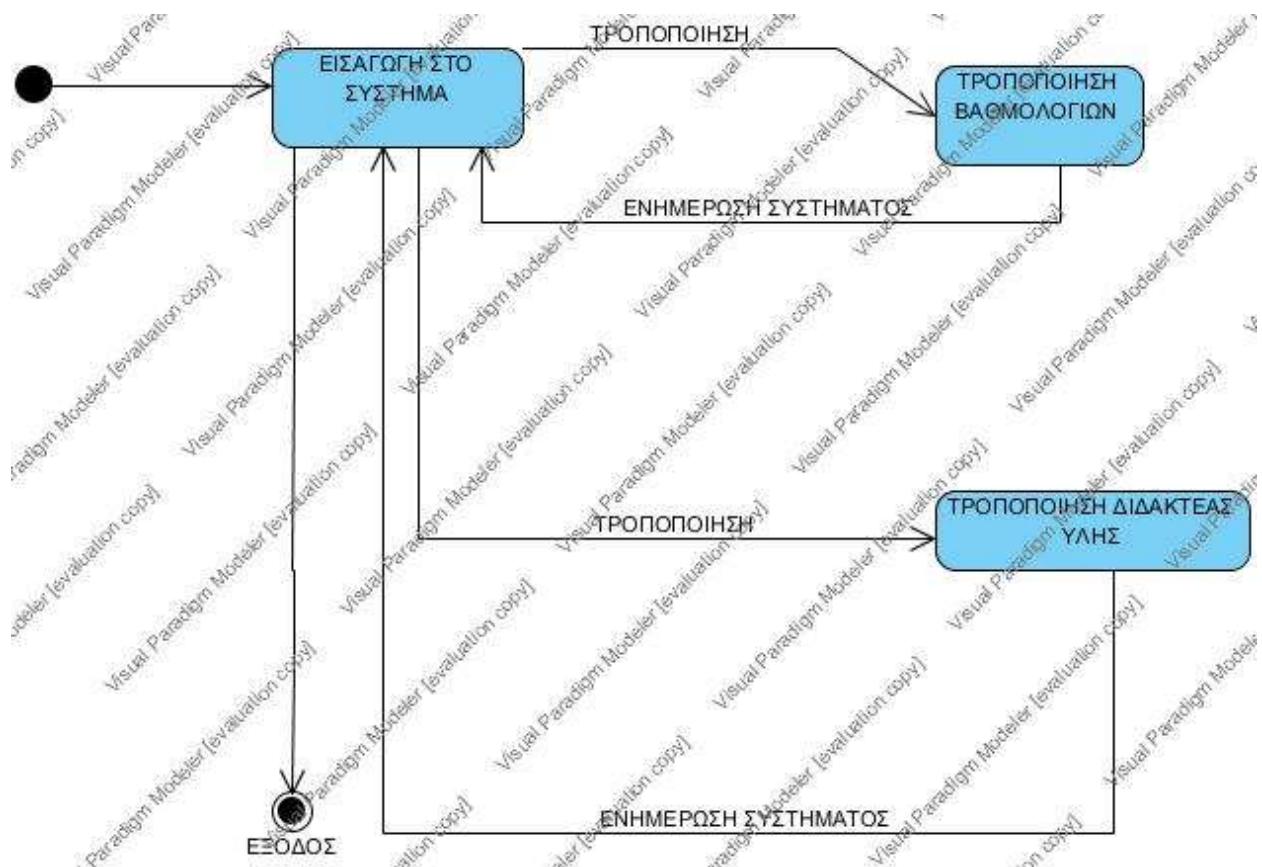


### 4.3. Διάγραμμα καταστάσεων (State machine diagram)

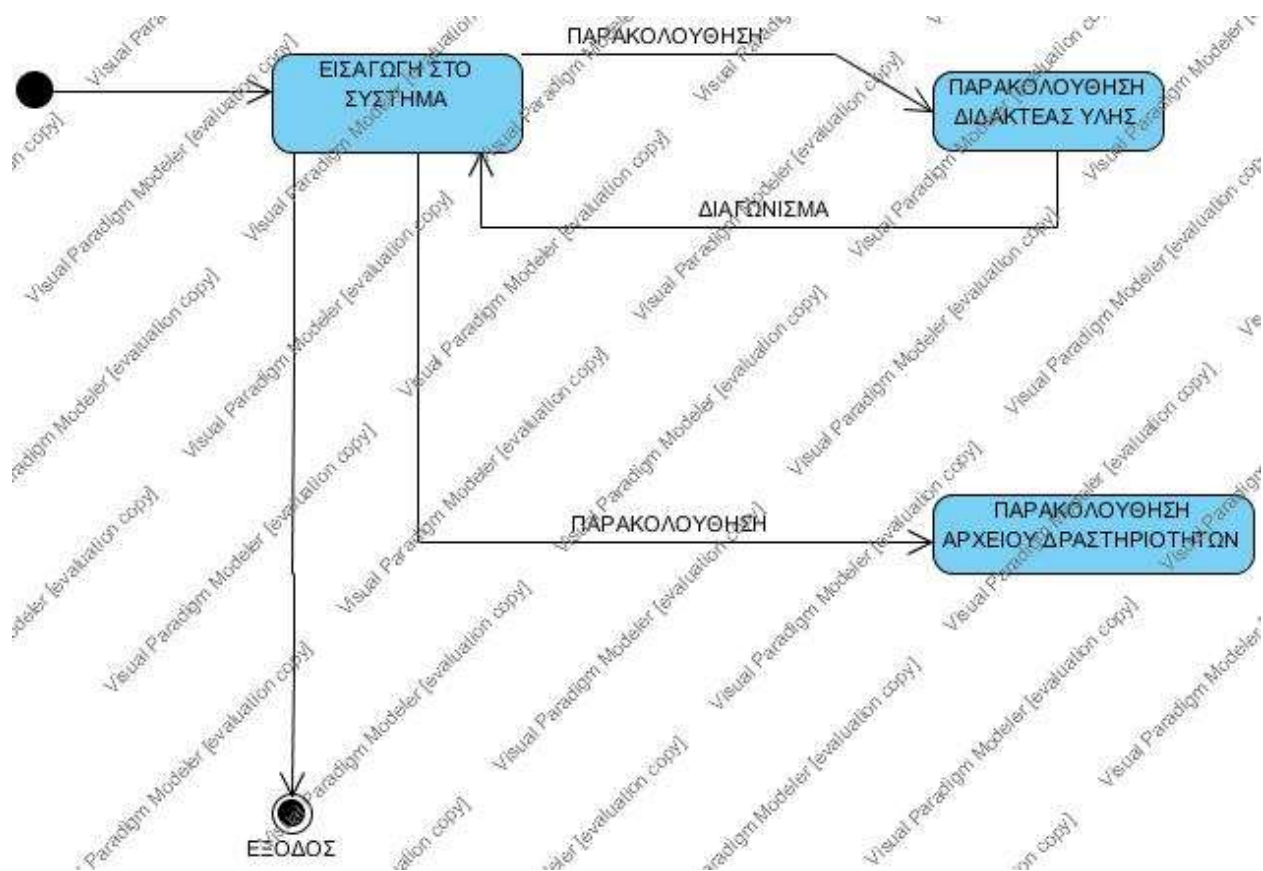
Στο διάγραμμα καταστάσεων θα απεικονιστεί η δυναμική συμπεριφορά των αντικειμένων της κλάσης "Μαθητής" και "Καθηγητής" και ο τρόπος που μεταβάλλεται η κατάσταση τους ως αντίδραση σε συμβάντα. Οι μεταβάσεις καταστάσεων του συστήματος έχουν ως εξής:



- Για τον καθηγητή



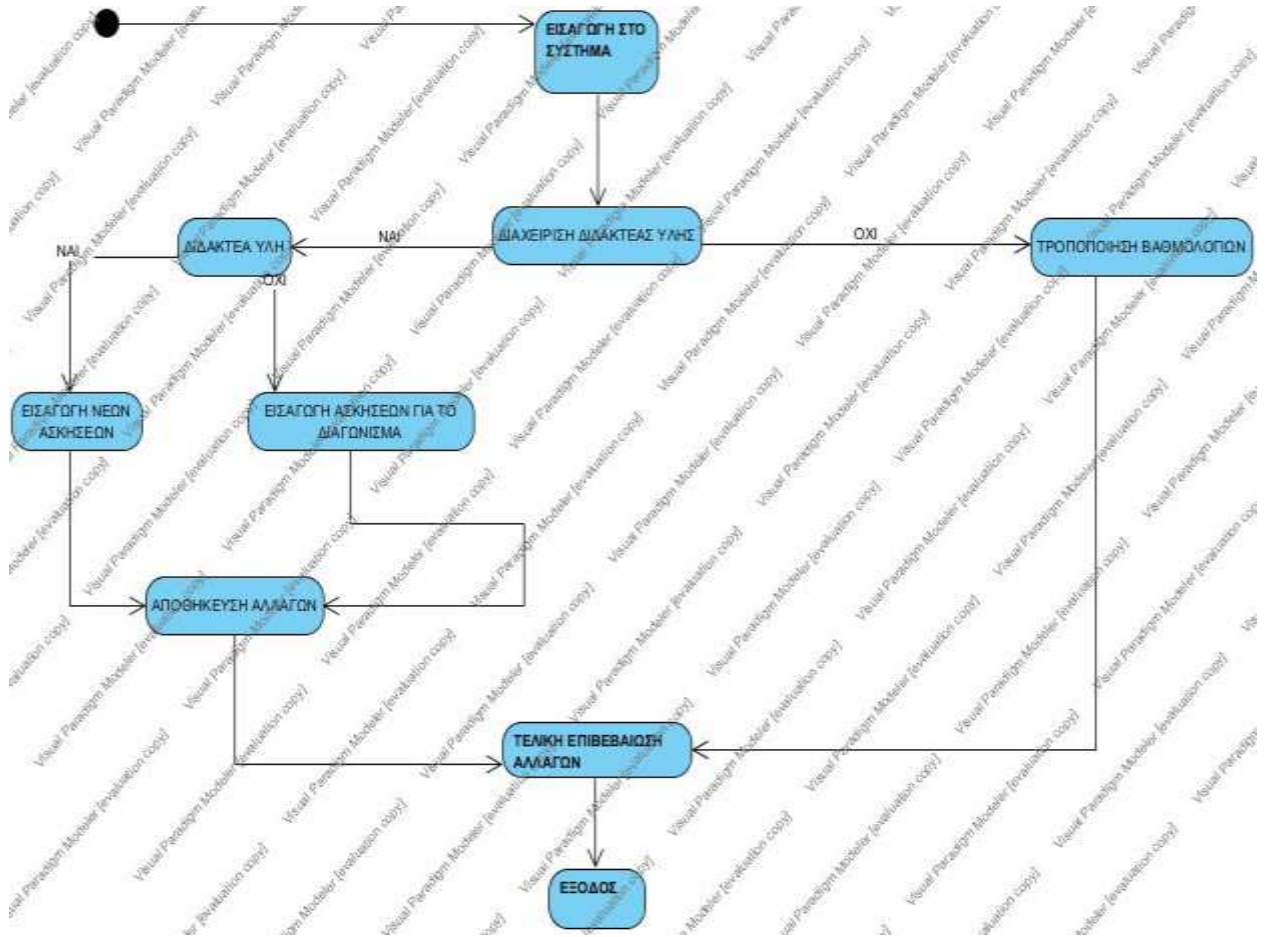
- Για το μαθητή



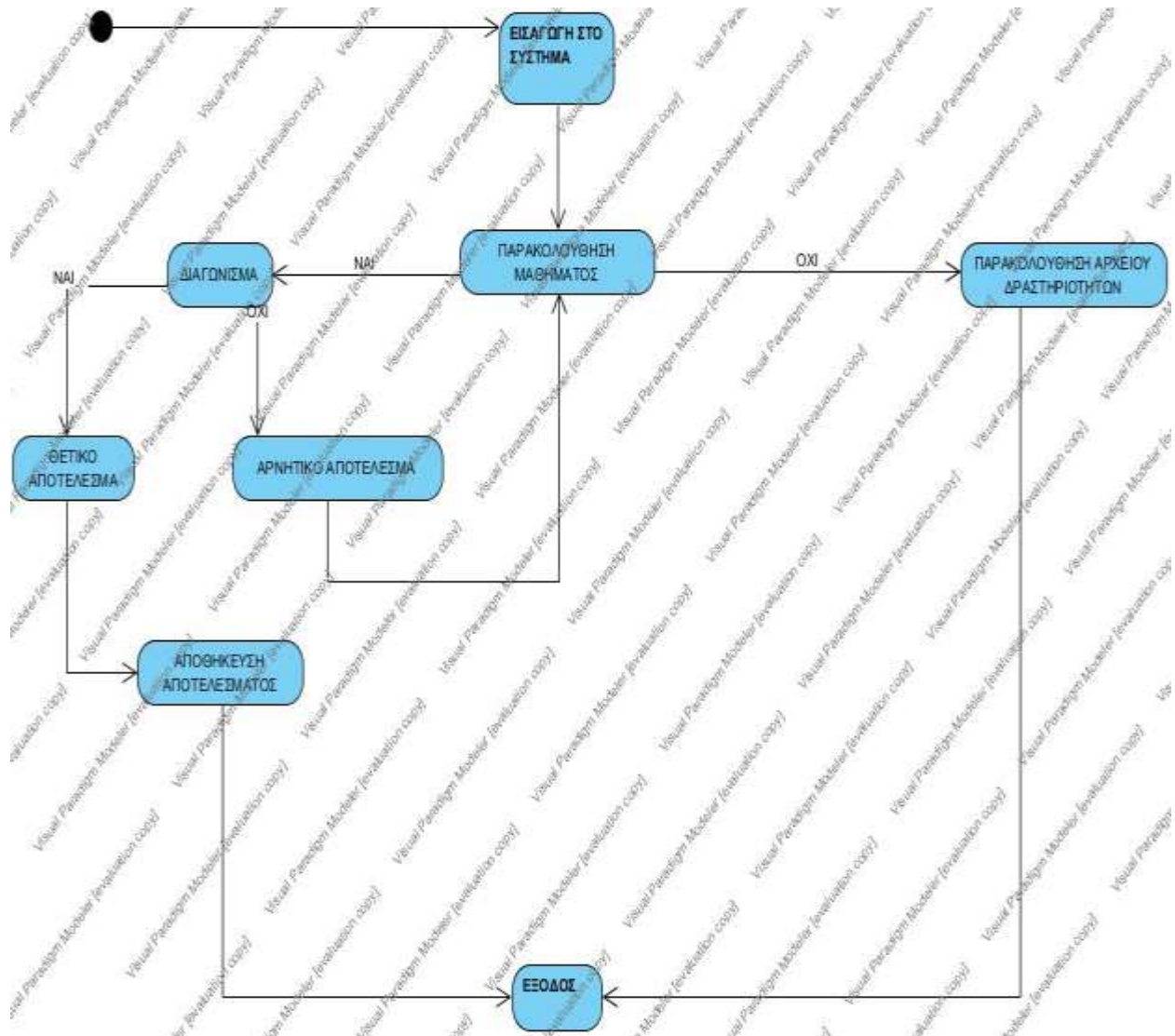
#### **4.4. Διάγραμμα δραστηριοτήτων (Activity diagram)**

Οι καταστάσεις ενεργειών και οι μεταβάσεις που λαμβάνουν χώρα κατά την ολοκλήρωση των ενεργειών διατυπώνονται ως εξής:

- Για τον καθηγητή



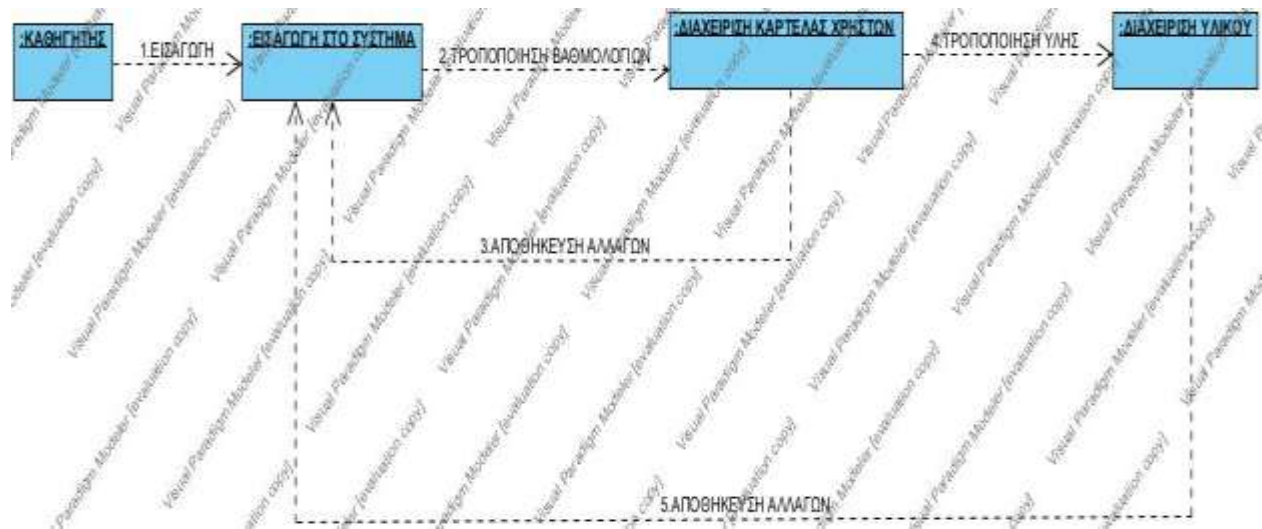
- Για το μαθητή



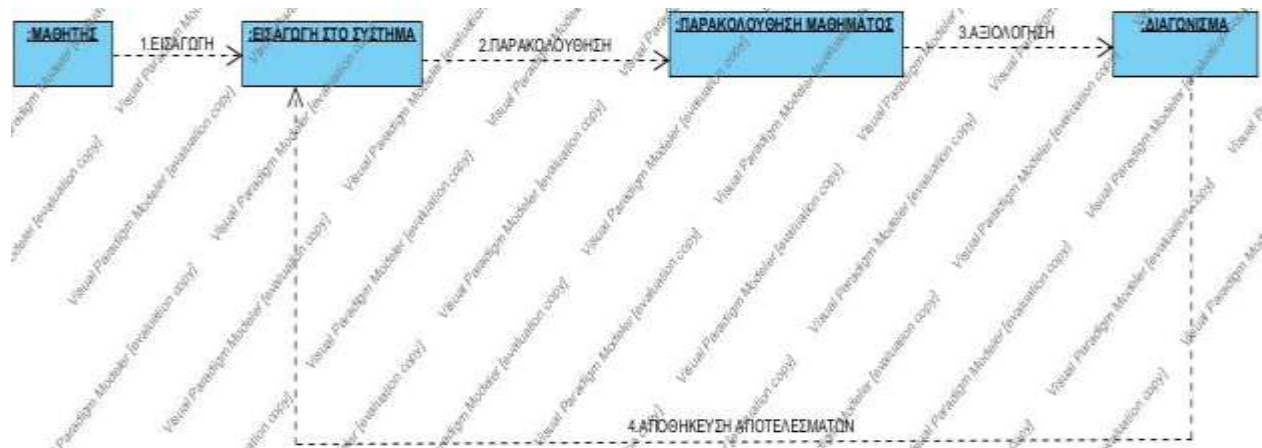
#### **4.5. Διάγραμμα συνεργασίας (Collaboration diagram)**

Στο διάγραμμα συνεργασίας παρουσιάζεται η διαμόρφωση του τρόπου με τον οποίο εμπλέκονται τα αντικείμενα σε ένα σενάριο. Στην περίπτωση μας το διάγραμμα συνεργασίας είναι το ακόλουθο:

- Για τον καθηγητή



- Για το μαθητή



#### 4.6. Διάγραμμα ανάπτυξης (Deployment diagram)

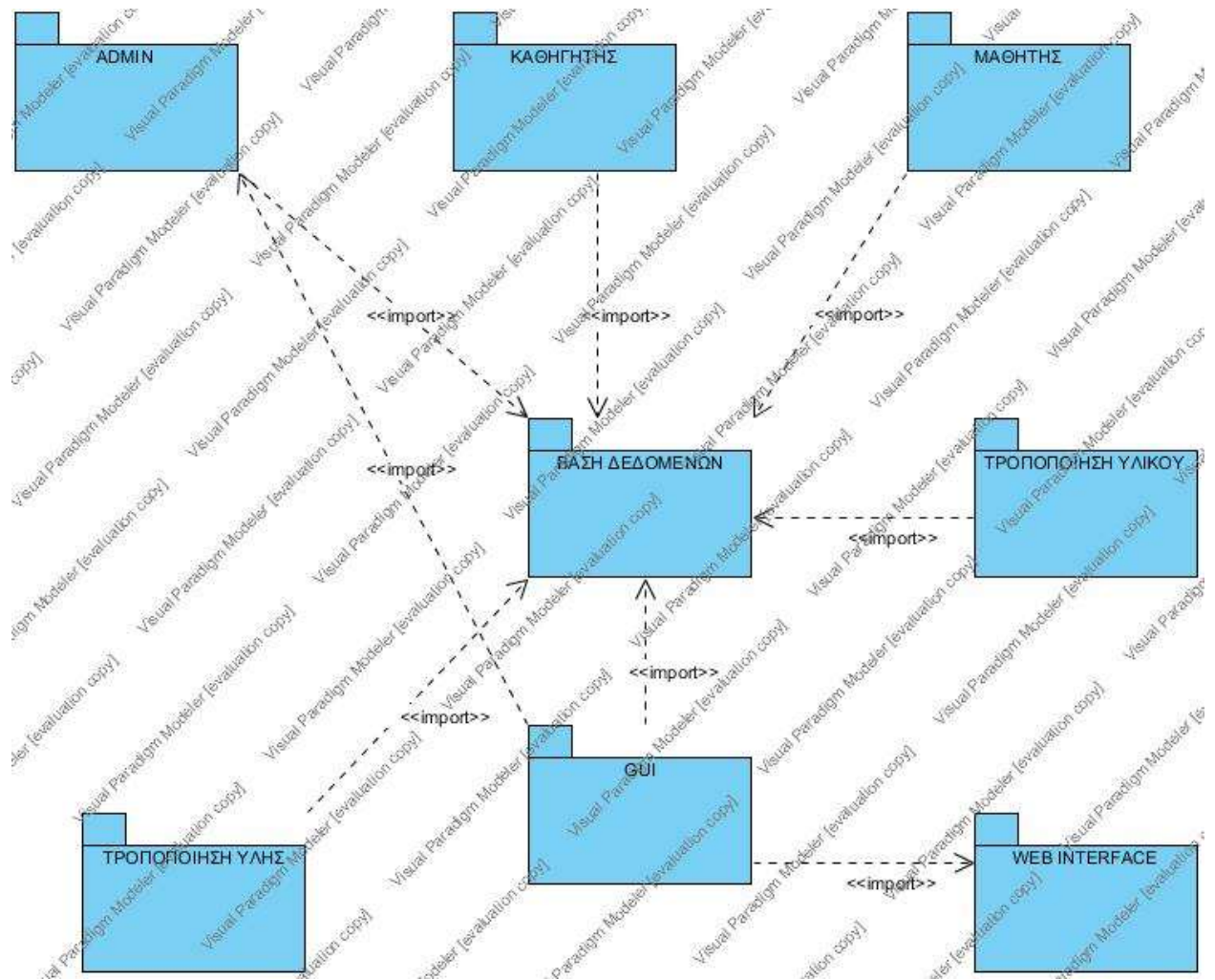
Στο διάγραμμα ανάπτυξης θα παρουσιαστεί η απεικόνιση της τοπολογίας υλικού (hardware topology) ώστε να χρησιμοποιηθεί για την ανάπτυξη των απαιτούμενων στοιχείων υλικού (deployment software componets).





#### **4.7. Διάγραμμα πακέτων (Package diagram)**

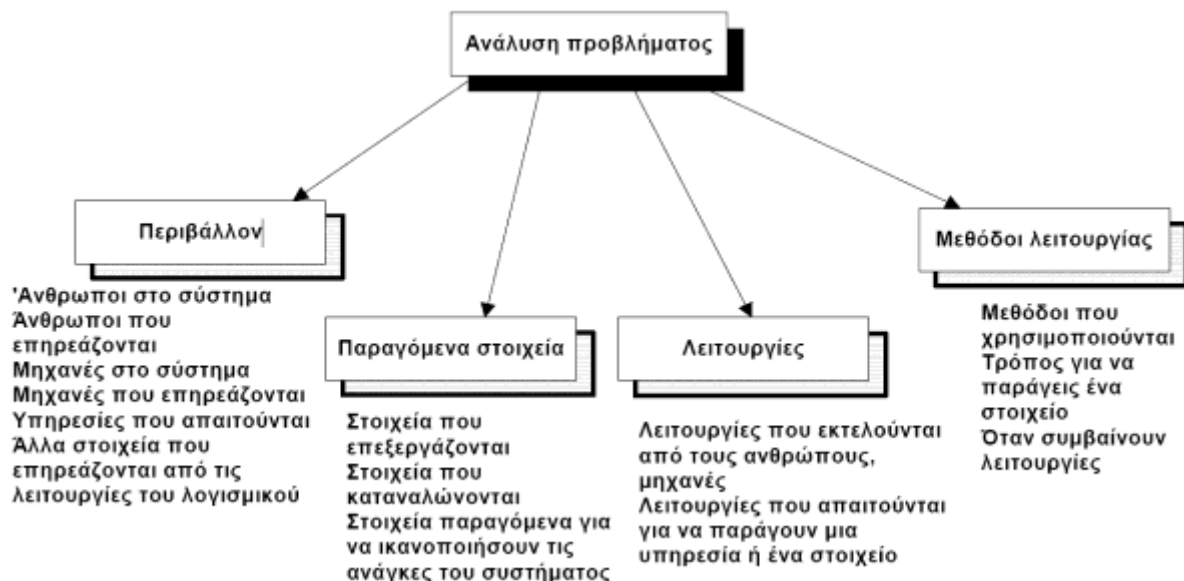
Τέλος, το διάγραμμα πακέτων είναι το ακόλουθο:



## 5. Ανάλυση Απαιτήσεων – Διάγραμμα Ροής Δεδομένων

Στην παράγραφο αυτή προδιαγράφονται οι απαιτήσεις από το σύστημα λογισμικού, γίνεται δηλαδή καθορισμός του τι θα κάνει συνολικά το σύστημα.

Η ανάλυση προβλήματος ή απαιτήσεων καθορίζει το διάστημα των προϊόντων μιας διαδικασίας λογισμικού (Davis, 1993). Στην πραγματικότητα, η ανάλυση προβλήματος καθορίζει το πλαίσιο για τις πιθανές λύσεις λογισμικού σε ένα



πρόβλημα, το οποίο έχει τα συστατικά που παρουσιάζονται στο παρακάτω σχήμα.

Με άλλα λόγια, η ανάλυση προβλήματος οδηγεί στον προσδιορισμό του περιβάλλοντος (άνθρωποι επηρεασμένοι από ένα προϊόν λογισμικού, μηχανές που χρησιμοποιούν ή επηρεάζονται από το λογισμικό, υπηρεσίες που εκτελούνται, και άλλα στοιχεία όπως κυκλοφοριακή ροή ή χρόνος επικοινωνίας), στοιχεία που παράγονται, κύριες λειτουργίες που εκτελούνται από τους

ανθρώπους και μηχανές ώστε να παράγουν ένα επιθυμητό προϊόν, μέθοδοι που απαιτούνται, και προγραμματισμό των λειτουργιών. Τρεις βασικές αρχές κατά τη διάρκεια της ανάλυσης προβλήματος έχουν προσδιοριστεί: διαίρεση, αφαίρεση, και υλοποίηση (Yeh & Zave, 1980).

- Η διαίρεση αθροίζει τις δομικές σχέσεις μεταξύ των αντικειμένων, των λειτουργιών, και των καταστάσεων, και απλοποιεί (χωρίζει σε διαμερίσματα) τις δομές που αναλύονται.
- Η αφαίρεση προσδιορίζει τις “γενικές / συγκεκριμένες” δομικές σχέσεις μεταξύ των αντικειμένων, των λειτουργιών, και των καταστάσεων.
- Η προβολή παρέχει μια «άποψη» των δομικών σχέσεων μεταξύ των αντικειμένων, των λειτουργιών, ή των καταστάσεων. Οι οργανωτικές προοπτικές των αντικειμένων, των λειτουργιών, και των καταστάσεων είναι πολύ χρήσιμες στην ανάπτυξη μιας κατανόησης ενός προβλήματος και της λύσης του.


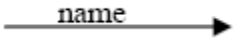
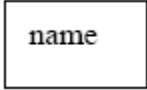
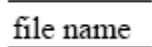

Η ανάλυση προβλήματος παρέχει ένα απαραίτητο ενδιάμεσο στάδιο που οδηγεί στην ανάπτυξη ενός προσδιορισμού απαιτήσεων λογισμικού. Ο προσδιορισμός απαιτήσεων ενδιαφέρεται κυρίως για την περιγραφή των αντικειμένων, των λειτουργιών, και των καταστάσεων σχετικών με ένα πρόβλημα.

Ένα διάγραμμα που διευκρινίζει τις διαδικασίες και τη ροή των δεδομένων μεταξύ τους καλείται Διάγραμμα Ροής Δεδομένων (Data Flow Diagram, DFD). Ένα

DFD εκθέτει πιθανές μορφές ροής πληροφορίας σε ένα σύστημα, θέσεις αποθήκευσης για τα δεδομένα, και μετασχηματισμούς των δεδομένων καθώς διατρέχουν ένα σύστημα.

Τα DFDs παρέχουν μια από τις παλαιότερες τεχνολογίες για την ανάλυση προβλημάτων που εισάγεται από τον DeMarco (1978) καθώς επίσης και από τον Gane και Sarson (1979). Τα DFDs είναι εύχρηστα, και πολύ χρήσιμα για να γεφυρώσουν το κενό μεταξύ των άτυπων περιγραφών ενός συστήματος σε μια δήλωση της ανάγκης και της ανάπτυξης των περιγραφών για το πώς οι πληροφορίες ρέουν σε ένα σύστημα.

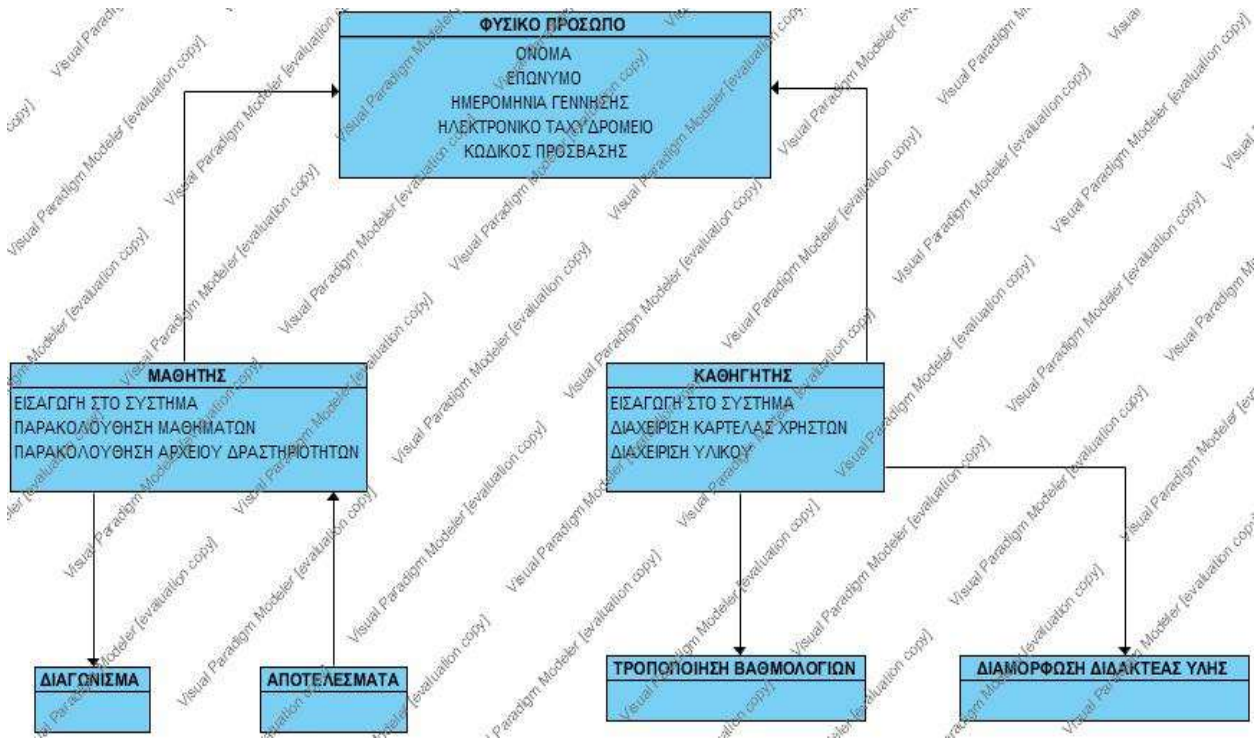
Η σημειογραφία για τα DFDs του De Marco συνοψίζεται στον παρακάτω πίνακα. Οι φυσαλίδες (κύκλοι με ονόματα) συμβολίζουν τους μετασχηματισμούς. Τα ονόματα πρέπει να είναι μοναδικά. Ένα βέλος καθορίζει την κατεύθυνση της ροής των δεδομένων. Οι ετικέτες των βελών προσδιορίζουν τα δεδομένα. Τα βέλη αντιπροσωπεύουν μια πορεία για τα ρέοντα δεδομένα, αλλά δεν καθορίζουν μια σειρά των γεγονότων. Τα ορθογώνια (επίσης αποκαλούμενα terminators) δείχνουν τις πηγές και τους προορισμούς των δεδομένων. Τέλος, οι παράλληλες γραμμές δείχνουν τα αρχεία, βάσεις δεδομένων, μόνιμες αποθήκες για τα δεδομένα, ενώ το ανοικτό ορθογώνιο δείχνει τις αποθήκες των δεδομένων.

Σύμβολο DFD	Σημασία
	<p>Προσδιορίζει έναν μετασχηματισμό (δραστηριότητα) που χρησιμοποιείται να επεξεργαστεί τα δεδομένα εισόδου και να αποκτήσει τα δεδομένα εξόδου</p>
	<p>Διευκρινίζει την κατεύθυνση της ροής των ονομασμένων δεδομένων</p>
	<p>Προσδιορίζει έναν terminator (τελικό αποδέκτη) για τα δεδομένα (πηγή ή προορισμός για τα δεδομένα)</p>
	<p>Προσδιορίζει μια μόνιμη θέση (αρχείο, βάση δεδομένων, ή αποθήκη) για τα δεδομένα</p>
	<p>Προσδιορίζει τις αποθήκες των δεδομένων</p>

## 5.1. Σχεδιασμός

Στη φάση του σχεδιασμού το λογισμικό διαιρείται σε ενότητες ,οι οποίες σχεδιάζονται και αργότερα κατασκευάζονται ξεχωριστά. Οι δομικές αυτές μονάδες θα διαιρεθούν με τη σειρά τους σε άλλες απλούστερες, έτσι ώστε το πολύπλοκο πρόβλημα της κατασκευής λογισμικού για τα επιμέρους να μετατραπεί σε απλούστερα τμήματα.





**ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΕΩΝ ΜΟΝΤΕΛΟΥ**

## **5.2. Έλεγχος**

Η φάση αυτή είναι ιδιαίτερως σημαντική. Ο κώδικας πρέπει να ελεγχθεί μέσα στο λειτουργικό σύστημα. Οι δομικές μονάδες ,όπως προέκυψαν από την παραπάνω σχεδίαση του λογισμικού, ελέγχονται στην αρχή ξεχωριστά και έπειτα γίνεται η συνένωση και ο έλεγχος του συνόλου.

### **Επαλήθευση**

Πρόκειται για τον έλεγχο του λογισμικού σχετικά με την ανάπτυξη του σωστού προϊόντος λογισμικού, δηλαδή επιβεβαιώνει ότι το σύστημα έχει αναπτυχθεί σωστά.

### **Επικύρωση**

Πρόκειται για τον έλεγχο του λογισμικού από το χρήστη σχετικά με την ανάπτυξη του σωστού προϊόντος, δηλαδή επιβεβαιώνει ότι το υπό ανάπτυξη σύστημα πληρεί τις απαιτήσεις του χρήστη.

Πιο συγκεκριμένα, η λίστα ελέγχου μιας επιθεώρησης που πρέπει να γίνεται σε κάθε βήμα, αποτελείται από τα παρακάτω στοιχεία:

- Καθορισμός συστήματος
- 1. Είναι οι κεντρικές λειτουργίες ορισμένες καλά ;
- 2. Έχουν καθοριστεί οι περιορισμοί σχεδιασμού ;
- 3. Είναι η λύση τεχνολογικά εφικτή ;
- 4. Υπάρχει καθορισμός του μηχανισμού επαλήθευσης και επικύρωσης ;
- 5. Υπάρχει συνέπεια μεταξύ όλων των στοιχείων του συστήματος ;

- Σχεδιασμός Έργου Λογισμικού

1. Είναι η ορολογία σαφής ;
2. Είναι οι πόροι αρκετοί για το έργο ;
3. Είναι οι πόροι διαθέσιμοι ;
4. Έχει γίνει ο προϋπολογισμός χρησιμοποιώντας 2 διαφορετικές μεθόδους ;
5. Έχουν συμφιλιωθεί οι διαφορές στους προϋπολογισμούς ;
6. Είναι οι προθεσμίες ρεαλιστικές ;

- Ανάλυση Απαιτήσεων

1. Είναι η ανάλυση του Πεδίου πλήρης, συνεπής και ακριβής ;
2. Έχουν καθοριστεί οι εξωτερικές και εσωτερικές διεπαφές ;
3. Είναι οι απαιτήσεις συνεπείς με το πλάνο, τους πόρους και τον προϋπολογισμό ;
4. Υπάρχουν κριτήρια επικύρωσης ;

- Σχεδιασμός Λογισμικού
  1. Υπάρχουν όλες οι απαιτήσεις στην αρχιτεκτονική του λογισμικού ;
  2. Υπάρχει καλή τμηματικότητα ;
  3. Έχει ληφθεί υπόψη η συντηρησιμότητα ;
  4. Έχουν εκτιμηθεί όλοι οι παράγοντες ποιότητας ;
  5. Οι αλγόριθμοι πετυχαίνουν τις αναμενόμενες λειτουργίες ;
  6. Είναι οι αλγόριθμοι λογικά σωστοί ;
  7. Έχει καθοριστεί η μέθοδος αποσφαλμάτωσης ;
  8. Είναι οι λεπτομέρειες του σχεδιασμού υλοποιήσιμες ;

- Κωδικοποίηση
  1. Έχει γίνει καλή μετάφραση του σχεδιασμού σε κώδικα ;
  2. Έχουν χρησιμοποιηθεί καλά οι συμβάσεις της γλώσσας ;
  3. Υπάρχουν λανθασμένα ή ασαφή σχόλια ;
  4. Είναι οι τύποι δεδομένων σωστοί ;

- Έλεγχος λογισμικού

1. Υπάρχει έγκαιρη επίδειξη των κυριότερων λειτουργιών ;
2. Υπάρχουν εργαλεία και πόροι ελέγχου ;
3. Υπάρχει πλάνο ελέγχου ;

- Συντήρηση

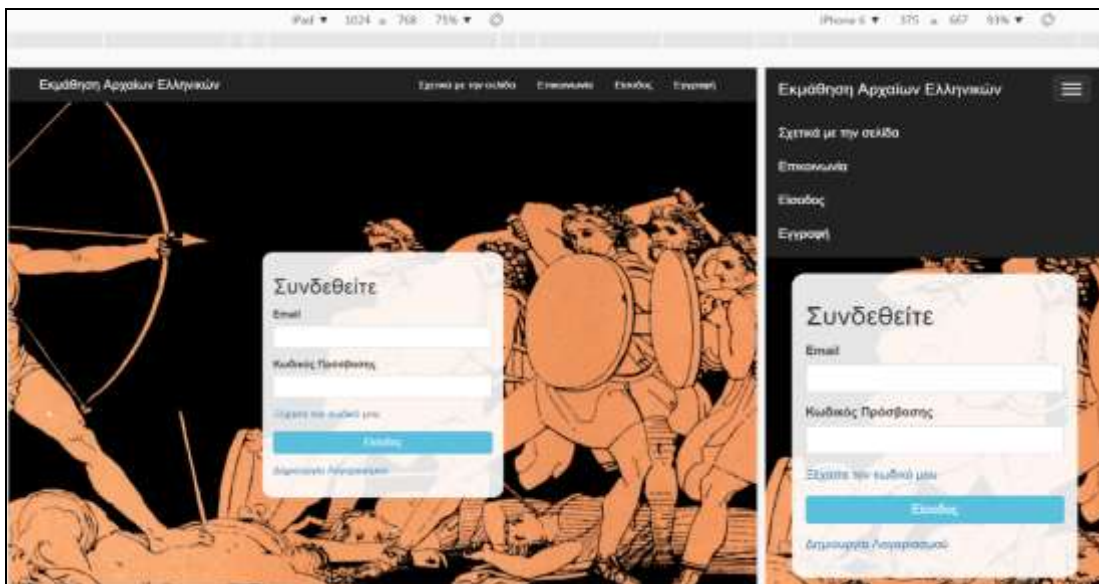
1. Έχουν ληφθεί υπόψη τα δευτερεύοντα αποτελέσματα μιας αλλαγής ;
2. Έχει η αίτηση για αλλαγή τεκμηριωθεί, αξιολογηθεί και εγκριθεί ;
3. Έχει η αλλαγή τεκμηριωθεί και αναφερθεί σε όλα τα ενδιαφερόμενα σημεία ;
4. Έχει γίνει μια τελική Επιθεώρηση Αποδοχής των αλλαγών ;

Η αξιοπιστία είναι ένας υψηλού επιπέδου δείκτης της λειτουργικής ετοιμότητας ενός συστήματος. Το κλειδί για τη βελτίωση της αξιοπιστίας λογισμικού είναι η κατασκευή μιας ακριβούς ιστορίας των λαθών, των ελαττωμάτων, και των ατελειών που συνδέονται με τις αποτυχίες λογισμικού. Τα λάθη, οι ατέλειες, τα ελαττώματα, και οι αποτυχίες έχουν μια σχέση αιτίας αποτελέσματος μεταξύ τους. Μια ατέλεια είναι μια ανωμαλία προϊόντος (π.χ., παραλείψεις ή ατέλειες που βρίσκονται κατά τη διάρκεια μιας ανάπτυξης λογισμικού). Ένα λάθος είναι μια ανθρώπινη ενέργεια που οδηγεί σε ένα ελάττωμα λογισμικού. Ένα ελάττωμα είναι μια τυχαία κατάσταση που αναγκάζει μια λειτουργική μονάδα να μην εκτελέσει τη απαραίτητη λειτουργία της (π.χ., ένα κουμπί σε ένα menu bar δεν αποκρίνεται στο ποντίκι) ή μια εκδήλωση ενός λάθους στην ανάπτυξη του λογισμικού.

Μια αποτυχία είναι μια προσωρινή διακοπή της δυνατότητας της λειτουργικής μονάδας να εκτελέσει μια απαραίτητη λειτουργία (π.χ. μια οθόνη κλειδώνει, αρνείται να αποκριθεί σε οποιοδήποτε χτύπημα ποντικιού). Ένας κύριος στόχος μιας αποτελεσματικής ανάπτυξης λογισμικού είναι να ανιχνεύσει τις αιτίες των αποτυχιών.

## **6. Εγχειρίδιο χρήσης**

Αυτή η ενότητα εξηγεί τη χρήση και την υλοποίησης της διεπαφής της εφαρμογής (front-end). Για την υλοποίηση όλων των σελίδων χρησιμοποιήθηκε HTML5 σε συνδυασμό με CSS3 και Javascript (jQueryversion 1.11<sup>1</sup>). Επιπλέον χρησιμοποιήθηκαν οι επεκτάσεις στο Bootstrapframeworkversion 3.3.7<sup>2</sup> για την δημιουργία ανταποκρίσιμου σχεδιασμού (responsivedesign) μέσω της χρήσης των κατάλληλων πλεγμάτων οι διαστάσεις των οποίων ανταποκρίνονται αυτόματα στο μέγεθος της οθόνης του χρήστη. Για παράδειγμα η Εικόνα 1 δείχνει το πως προβάλλεται η αρχική σελίδα σε μια συσκευή iPad με ανάλυση 1024 x 768, και σε μία συσκευή iPhone 6 με ανάλυση 375 x 667 (με βάση το Chrome Developer Tools).



**Εικόνα 1:** Η δομή της διεπαφής είναι βασισμένη στα πλεγμάτα του Bootstrapframeworkπου επιτρέπουν την ανταπόκριση του σχεδιασμού στο μέγεθος της οθόνης της συσκευής που χρησιμοποιεί ο χρήστης.

<sup>1</sup><https://jquery.com/>

<sup>2</sup><http://getbootstrap.com/>



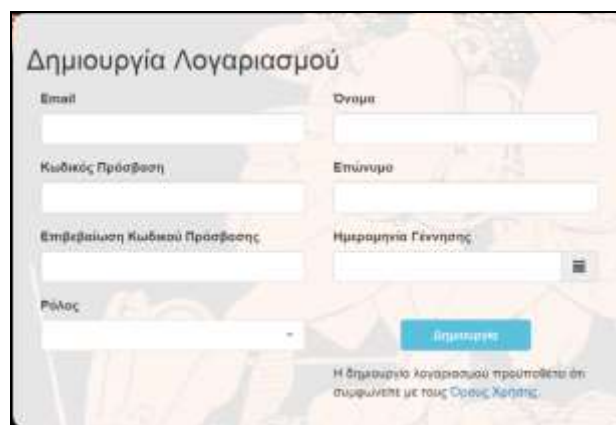
**Σημείωση:** Σε ένα περιβάλλον παραγωγής τα στοιχεία που υποβάλλονται από τις αντίστοιχες φόρμες για την αυθεντικοποίηση ενός χρήστη πρέπει να κρυπτογραφούνται με χρήση του HTTPSπρωτοκόλλου. Στην περίπτωση μας δεν χρησιμοποιήσαμε HTTPS διότι δουλεύουμε αποκλειστικά σε περιβάλλον ανάπτυξης. Παρόλα αυτά η αγορά και η εγκατάσταση ενός κλειδιού SSLείναι ανεξάρτητη από την προγραμματιστική ανάπτυξη της εφαρμογής και επομένως θεωρήθηκε εκτός του πλαισίου της εργασίας.

## Αρχική Σελίδα

Η αρχική σελίδα περιέχει τις φόρμες για την είσοδο στην εφαρμογή (sign-in), και τη δημιουργία νέου λογαριασμού χρήστη (sign-up). Και οι δύο φόρμες υποβάλλονται με ασύγχρονη (AJAX) μέθοδο POST στα ανάλογα URLs εφόσον τα απαιτούμενα πεδία έχουν συμπληρωθεί σωστά. Για τον έλεγχο των πεδίων χρησιμοποιήθηκε η javascript βιβλιοθήκη ParsleyJS<sup>3</sup>, η οποία δέχεται ως input τους κανόνες με βάση τους οποίους επιστρέφει μια τιμή true ή false για την εγκυρότητα των τιμών που υπέβαλε ο χρήστης.

### Δημιουργία Λογαριασμού (sign-up)

Το πρώτο βήμα για ένα νέο χρήστη είναι η δημιουργία ενός λογαριασμού για να αποκτήσει πρόσβαση στην εφαρμογή. Η Εικόνα 2 δείχνει τη φόρμα εγγραφής που βρίσκεται στη διεύθυνση [/#sign\\_up](#).



Εικόνα 2: Η φόρμα για τη δημιουργία ενός νέου λογαριασμού χρήστη

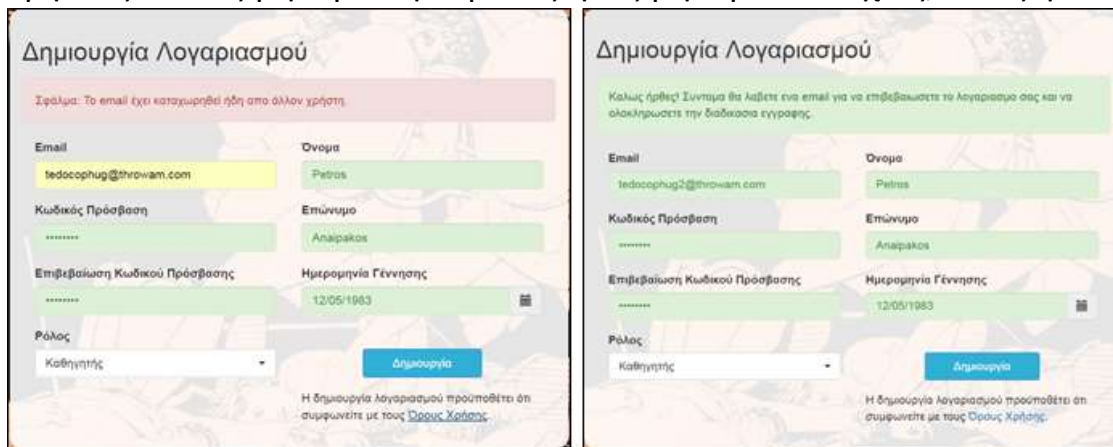
<sup>3</sup><http://parsleyjs.org/>

Όλα τα πεδία της φόρμας είναι απαραίτητο να συμπληρωθούν για να υποβληθεί η φόρμα εγγραφής, και επιπλέον ορίζουμε τους παρακάτω ελέγχους για την επαλήθευση των στοιχείων:

1. Το email να έχει format<username>@<slid>.<tld>
2. Τα πεδία του κωδικού πρόσβασης να συμφωνούν μεταξύ τους
3. Η ημερομηνία γέννησης να έχει το formatDD/MM/YYYY
4. Η ημερομηνία γέννησης να μην είναι στο μέλλον

Αν η επαλήθευση των στοιχείων είναι επιτυχής η φόρμα υποβάλλεται στη διεύθυνση [/register](#) για επεξεργασία από το back-end της εφαρμογής. Η διεπαφή περιμένει να λάβει ως απάντηση ένα JSONarray με το ακόλουθο format: {"error": Boolean, "message": String}

Ανάλογα με την τιμή του "error" (true ή false), το περιεχόμενο του "message" εμφανίζεται ως μήνυμα σφάλματος ή ως μήνυμα επιτυχίας, όπως φαίνεται στην

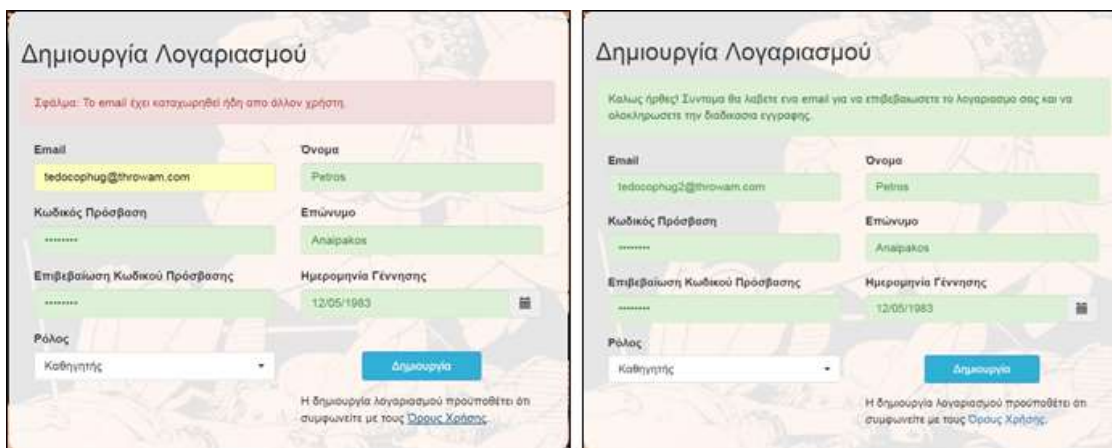


The image shows two side-by-side screenshots of a registration form titled "Δημιουργία Λογαριασμού".

The left screenshot shows an error message: "Σφάλμα: Το email έχει καταχωρηθεί ήδη από άλλον χρήστη." (Error: The email has already been registered by another user). The form fields are: Email (tedosorhug@throwam.com), Password (P@lms), Confirm Password (P@lms), First Name (P@lms), Last Name (Απαρκαός), Birth Date (12/05/1983), and Role (Καθηγητής). A blue "Δημιουργία" button is visible.

The right screenshot shows a success message: "Καλώς ήρθες! Συντομα θα λάβεις ένα email για να επαβεβαιώσεις το λογαριασμό σου και να ολοκληρώσεις την διαδικασία εγγραφής." (Welcome! You will soon receive an email to verify your account and complete the registration process). The form fields are the same as in the left screenshot, but the error message is gone.

Εικόνα 3.



**Εικόνα 3: Εμφάνιση μηνυμάτων σφάλματος(αριστερά)και επιτυχίας (δεξιά)κατά την εγγραφή χρήστη.**

Αν η αίτηση εγγραφής ολοκληρωθεί επιτυχώς τα στοιχεία του χρήστη θα καταχωρηθούν στη βάση δεδομένων αλλά ο λογαριασμός δεν θα ενεργοποιηθεί προτού επιβεβαιωθεί ότι η διεύθυνση email είναι σωστή. Ο χρήστης θα λάβει ένα email στην διεύθυνση που έδωσε με ένα link που περιέχει τον κωδικό ενεργοποίησης και το ID του χρήστη, ως εξής:

</activate?x=e7f6903fa76e0b40deda3adf481b2f8d42da54c3&y=43>

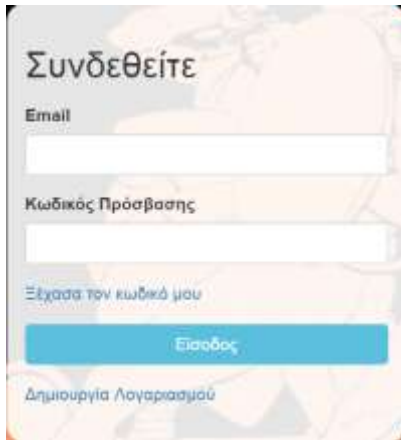
Το link είναι έγκυρο για 1 ώρα και αν ο χρήστης δεν το επισκεφτεί πριν λήξει ο λογαριασμός του θα παραμείνει ανενεργός. Το link μπορεί να χρησιμοποιηθεί μόνο μια φορά και μετά ακυρώνεται.

Σύνδεση στην Εφαρμογή (log-in)

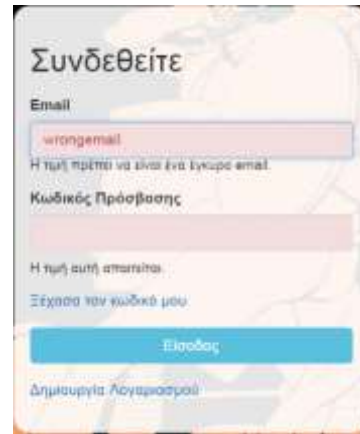
Η σύνδεση στην εφαρμογή γίνεται μέσω της φόρμας στη διεύθυνση [/#sign\\_in](#)(Εικόνα 4). Ο χρήστης πρέπει να συμπληρώσει και τα δύο πεδία της φόρμας για να επιτραπεί η υποβολή της, και το πεδίο του email πρέπει να έχει το format<username>@<slid>.<tld>. Επιπλέον των πεδίων που εμφανίζονται η

φόρμα σύνδεσης περιέχει ένα κρυμμένο πεδίο που περιέχει ένα κωδικό για την αποφυγή επιθέσεων CSRF (CrossSiteRequestForgery).

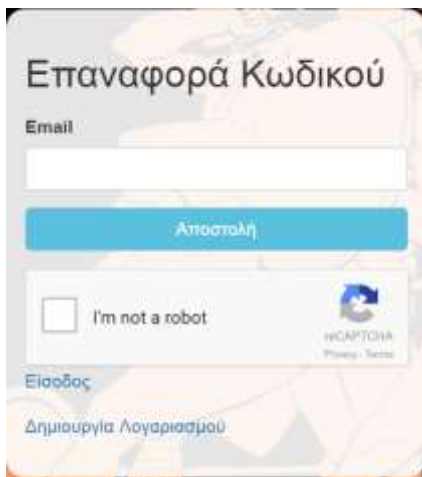
Η φόρμα υποβάλλεται στη διεύθυνση [/login](#) για επεξεργασία από την ανάλογη PHP μέθοδο που αυθεντικοποιεί το χρήστη. Όπως για τη φόρμα εγγραφής, ο server απαντάει με ένα JSONarray που περιέχει μια Boolean τιμή για το αν η σύνδεση ήταν επιτυχής και το ανάλογο. Σε περίπτωση που η σύνδεση είναι επιτυχής ο χρήστης κατευθύνεται στις εσωτερικές σελίδες της εφαρμογής, ενώ αν η σύνδεση απέτυχε εμφανίζεται το αντίστοιχο μήνυμα σφάλματος, όπως φαίνεται στην Εικόνα 5.



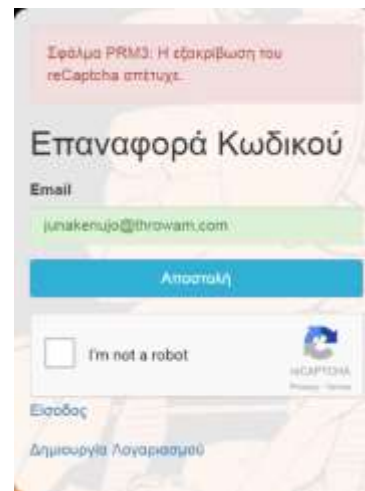
Εικόνα 4: Η φόρμα σύνδεσης (log-in)



Εικόνα 5: Υλοποίηση ελέγχου εγκυρότητας των πεδίων στη φόρμα σύνδεσης



Εικόνα 6: Η φόρμα επαναφοράς του κωδικού πρόσβασης



Εικόνα 7: Η αίτηση επαναφοράς κωδικού απαιτεί την επιβεβαίωση μέσω reCaptcha για να αποφεύγονται επιθέσεις στους κωδικούς χρηστών

## Επαναφορά Κωδικού Πρόσβασης

Αν ο χρήστης ξεχάσει τον κωδικό πρόσβασης μπορεί να ζητήσει την επαναφορά του κωδικού μέσω της φόρμας σύνδεσης επιλέγοντας «Ξέχασα τον κωδικό μου» αντί για την υποβολή της φόρμας (Εικόνα 6). Η εφαρμογή ελέγχει αν ο λογαριασμός του email υπάρχει και αν ο χρήστης έχει επιβεβαιωθεί μέσω του

GooglereCaptcha<sup>4</sup> για να αποφεύγεται η αυτοματοποιημένη επίθεση σε λογαριασμούς χρηστών μέσω scripts (Εικόνα 7). Η φόρμα υποβάλλεται μέσω AJAXPOSTHTTP αίτηση στην διεύθυνση </reset/request>. Αν ο χρήστης συμπληρώσει τα σωστά στοιχεία θα λάβει ένα email που θα παρέχει ένα link μιας χρήσης (παρόμοιο με αυτό της ενεργοποίησης τους λογαριασμού) το οποίο ο χρήστης μπορεί να επισκεφτεί για να αποκτήσει πρόσβαση στη φόρμα δημιουργίας νέου κωδικού πρόσβασης (Εικόνα 8).

</reset/new?x=fcbd8219610f8a80ca252f3d24706b75ef612321&y=48>

Το link λήγει σε 1 ώρα από τη δημιουργία του και αν χρησιμοποιηθεί μία φορά απενεργοποιείται και δεν μπορεί να χρησιμοποιηθεί ξανά για την δημιουργία νέου κωδικού. Όπως και στη φόρμα εγγραφής νέου χρήστη, ο χρήστης πρέπει να εισάγει τον ίδιο κωδικό δύο φορές για να υποβληθεί η φόρμα μέσω AJAXHTTPPOST αίτησης στη διεύθυνση </reset/new>.

Όταν ένας χρήστης συνδεθεί επιτυχώς θα κατευθυνθεί στην κατάλληλη σελίδα με βάση το ρόλο του. Ο χρήστης μπορεί να αποσυνδεθεί (log-out) πατώντας το κουμπί έξοδος που βρίσκεται στην κορυφή κάθε σελίδας. Αν ο χρήστης παραμείνει ανενεργός για πάνω από 20 λεπτά θα αποσυνδεθεί αυτόματα για λόγους ασφαλείας. Αν ο χρήστης προσπαθήσει να επισκεφθεί σελίδες στις οποίες ο ρόλος του δεν επιτρέπει πρόσβαση θα αποσυνδεθεί αυτόματα.

---

<sup>4</sup><https://www.google.com/recaptcha/intro/index.html>

Εισάγετε τον νέο κωδικό πρόσβασης

Κωδικός Πρόσβασης

Επιβεβαίωση Κωδικού Πρόσβασης

Αποστολή

Εισόδος

Δημιουργία Λογαριασμού

Εικόνα 8: Η φόρμα επαναδημιουργίας κωδικού πρόσβασης

Σφάλμα PRM9: Άκυρος σύνδεσμος επαναφοράς του κωδικού πρόσβασης.

Συνδεθείτε

Email

Κωδικός Πρόσβασης

Ξεχάσα τον κωδικό μου

Εισόδος

Δημιουργία Λογαριασμού

Εικόνα 9: Προσπάθεια επαναχρησιμοποίησης του ίδιου link δημιουργίας νέου κωδικού θα οδηγήσει σε σφάλμα.







## Σελίδες Καθηγητή

Όταν ένας καθηγητής συνδεθεί στην εφαρμογή μπορεί να εκτελέσει μία από τις ακόλουθες ενέργειες:

1. Δημιουργία νέου μαθήματος.
2. Απαρίθμηση των μαθημάτων που έχει δημιουργήσει.
3. Επεξεργασία ή διαγραφή υπάρχοντος μαθήματος.
4. Δημιουργία ασκήσεων πολλαπλής επιλογής.
5. Δημιουργία ασκήσεων σωστού/λάθους.
6. Επεξεργασία των ασκήσεων.
7. Απαρίθμηση των ασκήσεων.
8. Έλεγχος των βαθμών των μαθητών για κάθε μάθημα.

Διαχείριση Μαθημάτων

Show 10 entries Search:

Επεξεργασία *	Τίτλος Μαθήματος	Ημερομηνία δημιουργίας μαθήματος	Τελευταία ενημέρωση μαθήματος	Βαθμοί
	1.ΕΙΔΗ ΜΕΤΟΧΩΝ ΣΤΗΝ ΑΡΧΑΙΑ ΕΛΛΗΝΙΚΗ ΓΛΩΣΣΑ	October 30, 2016 09:54	October 30, 2016 10:45	
	2.ΚΑΤΗΓΟΡΟΥΜΕΝΑ ΣΤΗΝ ΑΡΧΑΙΑ ΕΛΛΗΝΙΚΗ ΓΛΩΣΣΑ	October 30, 2016 09:52	October 30, 2016 10:13	
	3.ΕΤΕΡΟΠΤΩΤΟΙ ΠΡΟΣΔΙΟΡΙΣΜΟΙ ΣΤΗΝ ΑΡΧΑΙΑ ΕΛΛΗΝΙΚΗ ΓΛΩΣΣΑ	October 30, 2016 11:18	October 30, 2016 12:15	

Showing 1 to 3 of 3 entries

Δημιουργία νέου μαθήματος  
GET /admin/new/

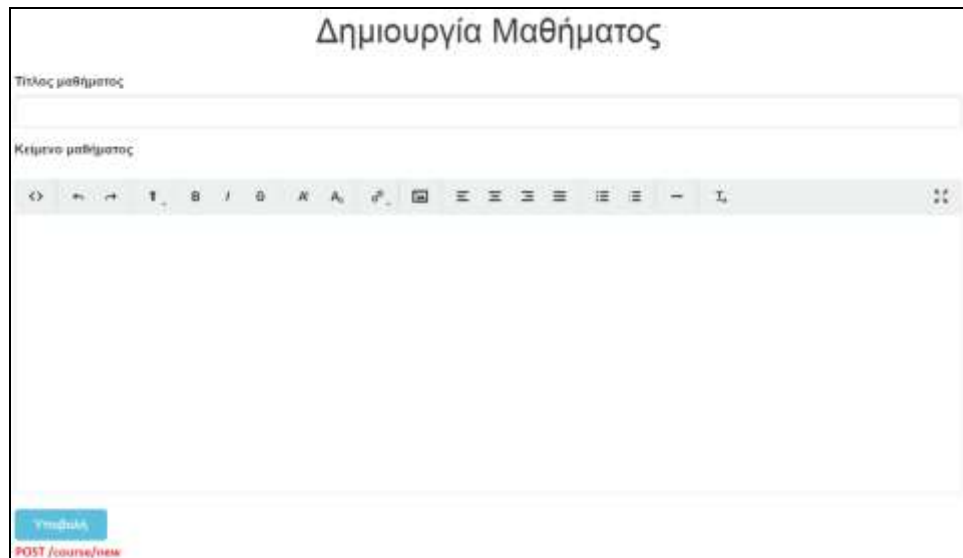
Previous 1 Next

+ Δημιουργία νέου μαθήματος

Εικόνα 10: Η διεπαφή απαρίθμησης και διαχείρισης των μαθημάτων



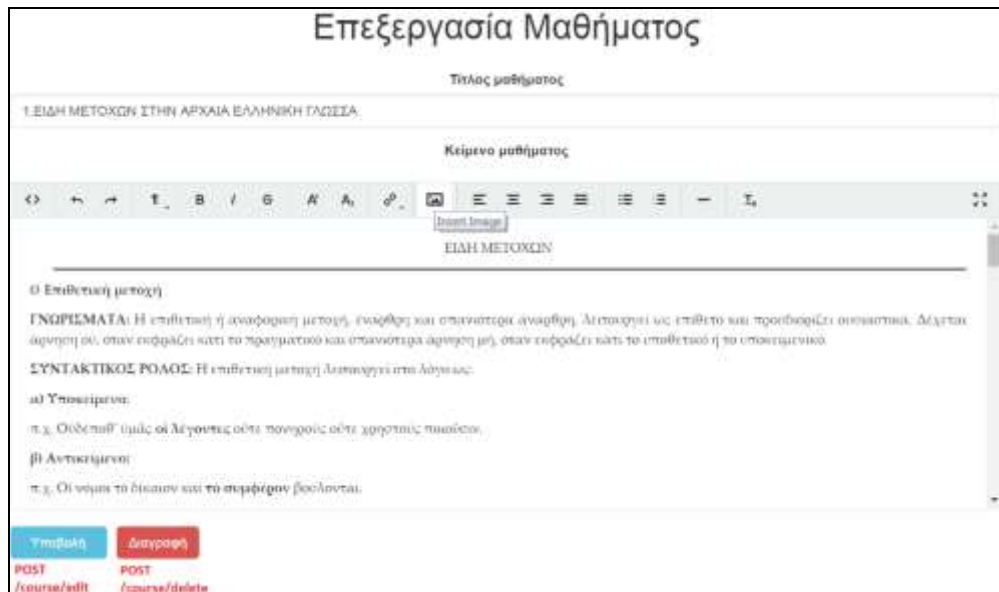
Η σελίδα απαρίθμησης και διαχείρισης των μαθημάτων είναι η κύρια σελίδα για έναν καθηγητή (Εικόνα 10). Ο πίνακας των μαθημάτων μπορεί να ταξινομηθεί με βάση κάθε στήλη, ενώ το πεδίο Search επιτρέπει στον χρήστη να φιλτράρει όλα τα πεδία των μαθημάτων με βάση κάποια λέξη. Η σελίδα δημιουργίας νέου μαθήματος φαίνεται στην Εικόνα 11. Ο καθηγητής έχει στη διάθεσή του μια φόρμα που υποστηρίζει μορφοποίηση WYIWIYG (WhatYouSeelsWhatYouGet) που υλοποιήθηκε με την jQuery βιβλιοθήκη Trumbowyg<sup>5</sup>. Ο καθηγητής εκτός από κείμενο μπορεί να κάνει embed εικόνες και βίντεο για να δημιουργήσει διαδραστικά μαθήματα.



**Εικόνα 11: Σελίδα δημιουργίας νέου μαθήματος. Η φόρμα υποβάλλεται με AJAXPOSTHTTPμέθοδο στη διεύθυνση /course/new**

Για να αποθηκευτεί το μάθημα απαιτείται η συμπλήρωση και του τίτλου και του κειμένου του μαθήματος, και αν η δημιουργία γίνει επιτυχώς η εφαρμογή θα επιστρέψει `successstatus`, ενώ αν αποτύχει θα επιστρέψει `errorstatus`.

<sup>5</sup><https://alex-d.github.io/Trumbowyg/>



**Εικόνα 12: Σελίδα επεξεργασίας μαθήματος. Η υποβολή αλλαγών γίνεται με AJAXHTPPOSTμέθοδο στη διεύθυνση /course/editενώ η διαγραφή γίνεται με AJAXHTPPOSTμέθοδο στη διεύθυνση /course/delete**

Η **Error! Unknown switch argument.** δείχνει τη σελίδα επεξεργασίας ενός υπάρχοντος μαθήματος, όπου φαίνεται και η μορφοποίηση του κειμένου. Το μορφοποιημένο κείμενο αποθηκεύεται ως HTML στη βάση δεδομένων για να μπορεί να προβληθεί στο χρήστη σε ένα <div>HTML στοιχείο χωρίς να χαθεί η μορφοποίηση. Ένα hiddeninput πεδίο περιέχει το ID του μαθήματος. Ένας καθηγητής μπορεί να επεξεργαστεί μόνο μαθήματα που έχει δημιουργήσει, και ακόμα και αν αλλάξει το ID του μαθήματος πριν την υποβολή της φόρμας, η επεξεργασία θα αποτύχει αν το αλλαγμένο ID δεν ανήκει σε μάθημα του καθηγητή που είναι συνδεδεμένος.

### Βαθμοί Μαθήματος

Show 10 entries Search:

Email	Όνομα	Επώνυμο	Τελευταία προσπάθεια	Βαθμός Πολλαπλής Επιλογής	Βαθμός Σωστού/Λάθους
panaipakos@gmail.com	Pierros	Anaipakos	October 30, 2016 13:13	100%	100%

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



### Εικόνα 13: Η οθόνη προβολής των βαθμών των μαθητών για ένα μάθημα

Η Εικόνα 13 δείχνει την οθόνη προβολής των βαθμών που έχουν πάρει οι μαθητές στις ασκήσεις πολλαπλής επιλογής και τις ασκήσεις σωστού/λάθους ξεχωριστά. Ο πίνακας με τους βαθμούς μπορεί να ταξινομηθεί με βάση την τιμή της κάθε στήλης, και επίσης ο καθηγητής μπορεί να ψάξει για συγκεκριμένα ονόματα, ημερομηνίες ή emails των μαθητών. Ο καθηγητής μπορεί να δημιουργήσει ασκήσεις μέσω της οθόνης διαχείρισης των ασκήσεων που ανοίγει επιλέγοντας το αντίστοιχο κουμπί «Ασκήσεις» που βρίσκεται στο μενού στην κορυφή κάθε σελίδας, δίπλα από την επιλογή «Έξοδος». Η Εικόνα 14 δείχνει την διεπαφή διαχείρισης των ασκήσεων.

Αρχαίων Ελληνικών Μαθήματα **Ασκήσεις** Έξοδος

## Διαχείριση Ασκήσεων

Show 10 entries Search:

Επεξεργασία	Οδηγίες	Ερώτηση	Τελευταία ενημέρωση ερώτησης	Τύπος ερώτησης	Μάθημα
 <small>Επεξεργασία άσκησης GET /tests/edit/{id}</small>	Να αναγνωρίσετε τις μετοχές στις παρακάτω προτάσεις:	Ο μη θαρρείς ού ποιδεύεται. Προσδιορίστε τη μετοχή: Ο μη θαρρείς	October 30, 2016 12:50	≡	1.ΕΙΔΗ ΜΕΤΟΧΩΝ ΣΤΗΝ ΑΡΧΑΙΑ ΕΛΛΗΝΙΚΗ ΓΛΩΣΣΑ
	Να αναγνωρίσετε τις μετοχές στις παρακάτω προτάσεις:	Συλλαμβάνο Κόρον ως άποκτενών. Προσδιορίστε τη μετοχή: άποκτενών	October 30, 2016 12:53	≡	1.ΕΙΔΗ ΜΕΤΟΧΩΝ ΣΤΗΝ ΑΡΧΑΙΑ ΕΛΛΗΝΙΚΗ ΓΛΩΣΣΑ

Showing 1 to 10 of 30 entries Previous 1 2 3 Next

[+ Δημιουργία νέας άσκησης πολλαπλής επιλογής](#) [+ Δημιουργία νέας άσκησης σωστού/λάθους](#)

Δημιουργία νέων ασκήσεων GET /tests/new και GET /tests/newbinary

**Εικόνα 14: Η οθόνη διαχείρισης των ασκήσεων**

Η Εικόνα 15 και η Εικόνα 16 δείχνουν τις φόρμες δημιουργίας νέων ερωτήσεων πολλαπλής επιλογής και σωστού/λάθους. Και στις δύο περιπτώσεις πρέπει να συμπληρωθούν όλα τα πεδία, ενώ στο κάτω μέρος της κάθε φόρμας ο καθηγητής επιλέγει το μάθημα στο οποίο θα αντιστοιχεί η ερώτηση.

Δημιουργία Άσκησης Πολλαπλής Επιλογής

Οδηγίες

Ερώτηση

Σωστή απάντηση

Λάθος απάντησης

Μάθημα: 1. ΕΙΔΗ ΜΕΤΟΧΩΝ ΣΤΗΝ ΑΡΧΑΙΑ ΕΛΛΗΝΙΚΗ ΓΛΩΣΣΑ

Υποβολή POST /test/new

**Εικόνα 15: Φόρμα για τη δημιουργία νέας άσκησης πολλαπλής επιλογής. Κάθε ερώτηση πολλαπλής επιλογής έχει μία σωστή και 3 λάθος ερωτήσεις.**

Δημιουργία Άσκησης Σωστού/Λάθους

Οδηγίες

Ερώτηση

Σωστό  Λάθος

Μάθημα: 1. ΕΙΔΗ ΜΕΤΟΧΩΝ ΣΤΗΝ ΑΡΧΑΙΑ ΕΛΛΗΝΙΚΗ ΓΛΩΣΣΑ

Υποβολή

**Εικόνα 16: Φόρμα δημιουργίας νέας άσκησης σωστού/λάθους.**

Τέλος, ο καθηγητής μπορεί να επεξεργαστεί τις ερωτήσεις που έχει ήδη δημιουργήσει επιλέγοντας το κουμπί της επεξεργασίας από την οθόνη διαχείρισης των ασκήσεων. Σε αυτή την περίπτωση τα πεδία είναι ήδη συμπληρωμένα με τα στοιχεία που ήδη έχει εισάγει ο καθηγητής για να γίνονται εύκολα μικρές αλλαγές στα υπάρχοντα κείμενα, όπως διόρθωση ορθογραφικών λαθών (Εικόνα 17).

Επεξεργασία Άσκησης

Οδηγίες

Να επιλέξετε τη σωστή απάντηση.

Ερώτηση

Η πόλις Αγησίλαον είλαντο βασιλεία. βασιλεία - Κατηγορούμενο του Υποκειμένου

Σωστό  Λάθος

Μάθημα: 1. ΕΙΔΗ ΜΕΤΟΧΩΝ ΣΤΗΝ ΑΡΧΑΙΑ ΕΛΛΗΝΙΚΗ ΓΛΩΣΣΑ

Υποβολή POST /test/edit

Εικόνα 17: Φόρμα επεξεργασίας άσκησης σωστού/λάθους. Η φόρμα υποβάλλεται με AJAXΗΤΤΟPOST μέθοδο στη διεύθυνση /test/edit

## Σελίδες Μαθητή

Ο μαθητής μπορεί να κάνει τις ακόλουθες λειτουργίες

1. Να απαριθμήσει τα μαθήματα που είναι διαθέσιμα.
2. Να ξεκινήσει το διάβασμα ενός μαθήματος.
3. Να απαντήσει στις ερωτήσεις πολλαπλής επιλογής.
4. Να απαντήσει στις ερωτήσεις σωστού/λάθους.

Η οθόνη προβολής των διαθέσιμων μαθημάτων είναι η προεπιλεγμένη σελίδα που εμφανίζεται στο μαθητή μετά το log-in (**Error! Unknown switch argument.**). Ο μαθητής μπορεί να βλέπει και να επιλέγει μαθήματα από διαφορετικούς καθηγητές.

Έναρξη *	Τίτλος Μαθήματος	Ημερομηνία ολοκλήρωσης μαθήματος	Βαθμός Πολλαπλής Επιλογής	Βαθμός Σωστού/Λάθους
	1.ΕΙΔΗ ΜΕΤΟΧΩΝ ΣΤΗΝ ΑΡΧΑΙΑ ΕΛΛΗΝΙΚΗ ΓΛΩΣΣΑ	-		
	2.ΚΑΤΗΓΟΡΟΥΜΕΝΑ ΣΤΗΝ ΑΡΧΑΙΑ ΕΛΛΗΝΙΚΗ ΓΛΩΣΣΑ	-		
	3.ΕΤΕΡΟΠΤΩΤΟΙ ΠΡΟΣΔΙΟΡΙΣΜΟΙ ΣΤΗΝ ΑΡΧΑΙΑ ΕΛΛΗΝΙΚΗ ΓΛΩΣΣΑ	-		

Showing 1 to 3 of 3 entries

**Εικόνα 18: Η σελίδα με τα μαθήματα που μπορεί να παρακολουθήσει ένας μαθητής**

Κάθε μάθημα έχει ένα κουμπί έναρξης, αλλά ο μαθητής πρέπει να ολοκληρώσει με επιτυχία το πρώτο μάθημα προτού προχωρήσει στο επόμενο. Επομένως στην αρχή μόνο το πρώτο κουμπί έναρξης είναι ενεργοποιημένο ενώ το επόμενο θα ενεργοποιηθεί εφόσον ο μαθητής πετύχει σκορ τουλάχιστον 80% και στις ερωτήσεις πολλαπλής επιλογής, και στις ερωτήσεις σωστού/λάθους.

**1.ΕΙΔΗ ΜΕΤΟΧΩΝ ΣΤΗΝ ΑΡΧΑΙΑ ΕΛΛΗΝΙΚΗ ΓΛΩΣΣΑ**

Παναγιώτης Αναστασίου 11/09/2018

---

ΕΙΔΗ ΜΕΤΟΧΩΝ

© Επιθετική μετοχή

**ΓΝΩΡΙΣΜΑΤΑ:** Η επιθετική ή ανάφορμη μετοχή, έναρξη και επανόνοτα έναρξη, λειτουργεί ως επιθετο και προσδίδει ποσοσιακά. Δεχεται άρνηση ού, όταν εισφράζει κάτι το πραγματικό και επανώνοτα άρνηση μη, όταν εισφράζει κάτι το υποθετικό ή το υποκειμενικό.

**ΣΥΝΤΑΚΤΙΚΟΣ ΡΟΛΟΣ:** Η επιθετική μετοχή λειτουργεί στο λόγο ως

α) Υποκειμενο:

π.χ. Ουδ' ουδ' ήμεις οι Αιγόντες ούτε πολερούς ούτε χρηστούς πισυόσιν.

β) Αντικείμενο:

π.χ. Οι νόμοι τού θεούσιν και τού συμφέρον βουλήσινται.

**Εικόνα 19: Σελίδα με το περιεχόμενο του μαθήματος**

Όταν ο μαθητής ξεκινήσει ένα μάθημα μπορεί να διαβάσει το κείμενο του μαθήματος και να δει ποίος είναι ο καθηγητής του μαθήματος και τότε έγινε η

τελευταία ενημέρωση του κειμένου (Εικόνα 19). Μόλις ο μαθητής αποφασίσει ότι είναι να προχωρήσει στις ασκήσεις επιλέγει το κουμπί «Ασκήσεις > » στο τέλος της σελίδας που τον κατευθύνει πρώτα στις ερωτήσεις πολλαπλής επιλογής (Εικόνα 20).

## 1.ΕΙΔΗ ΜΕΤΟΧΩΝ ΣΤΗΝ ΑΡΧΑΙΑ ΕΛΛΗΝΙΚΗ

### Ασκήσεις Πολλαπλής Επιλογής

Να αναγνωρίσετε τις μετοχές στις παρακάτω προτάσεις:

Ό μη δαρείς ού παιδεύεται. Προσδιορίστε τη μετοχή: Ό μη δαρείς

- αιτιολογική μετοχή
- τροπική μετοχή
- επιθετική μετοχή
- κατηγορηματική μετοχή

---

Συλλαμβάνει Κῦρον ὡς ἀποκτενῶν. Προσδιορίστε τη μετοχή: ἀποκτενῶν

- αιτιολογική μετοχή
- υποθετική μετοχή
- τελική μετοχή
- επιθετική μετοχή

**Εικόνα 20: Ερωτήσεις πολλαπλής επιλογής που είναι συνδεδεμένες με το τρέχον μάθημα**

Ο χρήστης πρέπει να απαντήσει όλες τις ερωτήσεις για να υποβάλει τις απαντήσεις. Η φόρμα υποβάλλεται με AJAXPOSTHTTP μέθοδο στη διεύθυνση </student/course/correct/{id}> που επιστρέφει τις απαντήσεις και το συνολικό σκορ του μαθητή, όπως φαίνεται στην **Error! Unknown switch argument..**



# 1.ΕΙΔΗ ΜΕΤΟΧΩΝ ΣΤΗΝ ΑΡΧΑΙΑ ΕΛΛΗΝΙΚΗ

## Ασκήσεις Πολλαπλής Επιλογής

Να αναγνωρίσετε τις μετοχές στις παρακάτω προτάσεις:

📊 Συνολικό σκορ: 0/2

❌ **Ο μή δαρεις ού παιδεύεται. Προσδιορίστε τη μετοχή: Ο μή δαρεις**

- κατηγορηματική μετοχή
- επιθετική μετοχή
- αιτιολογική μετοχή
- τροπική μετοχή

🟢 **Συλλαμβάνει Κύρον ως άποκτενών. Προσδιορίστε τη μετοχή: άποκτενών**

- επιθετική μετοχή
- τελική μετοχή
- αιτιολογική μετοχή
- υποθετική μετοχή

**Εικόνα 21: Τα αποτελέσματα για τις απαντήσεις που έδωσε ο χρήστης**

Ο χρήστης μπορεί να πάει πίσω στο μάθημα για να ξαναδιαβάσει αν θέλει, ή να προχωρήσει στις ερωτήσεις σωστού/λάθους (Εικόνα 22). Μετά μπορεί να ξανά προσπαθήσει τις ασκήσεις μέχρι να πετύχει το απαιτούμενο σκορ. Η σειρά των ερωτήσεων και των απαντήσεων αλλάζει τυχαία μεταξύ κάθε φόρτωσης της σελίδας για να μην είναι δυνατή η προσπάθεια κάθε πιθανής επιλογής σειριακά. Μόλις ο μαθητής ολοκληρώσει τις ασκήσεις ο πίνακας μαθημάτων θα ενημερωθεί με τα τελευταία σκορ του (Εικόνα 23).

## 1.ΕΙΔΗ ΜΕΤΟΧΩΝ ΣΤΗΝ ΑΡΧΑΙΑ ΕΛΛ

Ασκήσεις Σωστού/Λάθους

Να επιλέξετε τη σωστή απάντηση

Συνολικό σκορ: 0.8

❖ **Είρηνος** οδεν Αρτεμισίαν μοχομένην. – Η μετοχή μοχομένην είναι κατηγορηματική?

Λάθος  
 Σωστό

❖ **Ο** Λήγων τούτο φεύδεται – Η μετοχή **Ο** Λήγων είναι επιθετική?

Λάθος  
 Σωστό

❖ **Ἦλθεν** τρέχων ἐκ Μαραθῶνος – Η μετοχή τρέχων είναι χρονική?

Λάθος  
 Σωστό

❖ **Αισχύνομαι** αὐ βοηθήσας. – Η μετοχή αὐ βοηθήσας είναι εναντιωματική?

Λάθος  
 Σωστό

❖ **Θηβαίοι** λυσαντες τας σπονδας ἐστράτευσαν ἐπὶ Πλαταιων. – Η μετοχή λυσαντες είναι χρονική?

Λάθος  
 Σωστό

← Πρώτη
Απάντη

Εικόνα 22: Αποτελέσματα στις απαντήσεις για τις ασκήσεις σωστού/λάθους

## Μαθήματα

Show 10 entries Search:

Έναρξη	Τίτλος Μαθήματος	Ημερομηνία ολοκλήρωσης μαθήματος	Βαθμός Παλλαπλής Επιλογής	Βαθμός Σωστού/Λάθους
▶	1.ΕΙΔΗ ΜΕΤΟΧΩΝ ΣΤΗΝ ΑΡΧΑΙΑ ΕΛΛΗΝΙΚΗ ΓΛΩΣΣΑ	11-09-2016 01:52:33	0.2	0.8
▶	2.ΚΑΤΗΓΟΡΟΥΜΕΝΑ ΣΤΗΝ ΑΡΧΑΙΑ ΕΛΛΗΝΙΚΗ ΓΛΩΣΣΑ	-		
▶	3.ΕΤΕΡΟΠΤΩΤΟΙ ΠΡΟΣΔΙΟΡΙΣΜΟΙ ΣΤΗΝ ΑΡΧΑΙΑ ΕΛΛΗΝΙΚΗ ΓΛΩΣΣΑ	-		

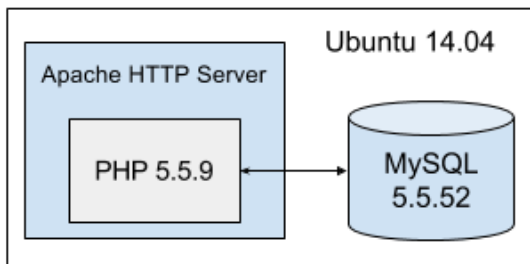
Showing 1 to 3 of 3 entries Previous 1 Next

Εικόνα 23: Η λίστα μαθημάτων ενημερωμένη με τους βαθμούς του χρήστη. Τα κουμπιά έναρξης για τα μαθήματα 2 και 3 παραμένουν απενεργοποιημένα

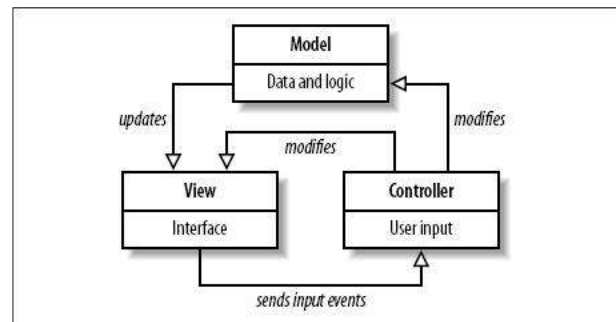
γιατί ο μαθητής απέτυχε να απαντήσει σωστά τουλάχιστον 80% των ερωτήσεων και στα δυο σετ ασκήσεων.

### Υλοποίησης της Πλευράς του Εξυπηρετητή (back-end)

Η υλοποίηση της εφαρμογής βασίστηκε στην αρχιτεκτονική που φαίνεται στην Εικόνα 4. Η χρήση του συνδυασμού PHP/MySQL/ApacheHTTPServer επιλέχθηκε για την ευρεία υποστήριξη από λειτουργικά συστήματα και την μεγάλη κοινότητα των χρηστών που προσφέρει πληθώρα πηγών. Ως πρότυπο σχεδίασης της εφαρμογής επιλέξαμε το MVC (Model - View - Controller) διότι επιτρέπει τον ξεκάθαρο διαχωρισμό των λειτουργιών σε κλάσεις ώστε η ανάπτυξη της εφαρμογής να γίνεται αρθρωτά (modular), να διευκολύνεται η επέκταση των λειτουργιών χωρίς επιπλέον πολυπλοκότητα στην διαδικασία της υλοποίησης και να είναι δυνατή η επαναχρησιμοποίηση στοιχείων της εφαρμογής. Η Εικόνα 25 Εικόνα 25 δείχνει την αλληλεπίδραση των οντοτήτων του μοντέλου MVC.



Εικόνα 24: Επισκόπηση της αρχιτεκτονικής



Εικόνα 25: Η εφαρμογή υλοποιήθηκε με βάση το πρότυπο σχεδίασης MVC (Model-View-Controller) για καλύτερο διαχωρισμό των λειτουργιών των διαφόρων κλάσεων της εφαρμογής.

Οι **ελεγκτές (controllers)** είναι οι κεντρικές οντότητες που ελέγχουν την εκτέλεση κάθε πράξης που απαιτεί ο χρήστης από το σύστημα, και στέλνουν τις κατάλληλες εντολές στα αντίστοιχα μοντέλα (models) για να επεξεργαστούν τα δεδομένα που χρειάζονται για να ολοκληρωθεί μια πράξη. Αφού ένας ελεγκτής λάβει τα κατάλληλα δεδομένα από το αντίστοιχο μοντέλο, επικοινωνεί με τις κατάλληλες όψεις (views) για να παρουσιαστεί στο χρήστη το αποτέλεσμα της πράξης. Ανάλογα με το αποτέλεσμα ο ελεγκτής παρουσιάζει τη νέα κατάσταση των δεδομένων ή ένα μήνυμα λάθους, αν η αίτηση του χρήστη δεν ικανοποιήθηκε. Κάθε ελεγκτής υλοποιεί μόνο τις πράξεις μίας συγκεκριμένης λειτουργίας του συστήματος για να υπάρχει σαφής διαχωρισμός των λειτουργιών μεταξύ των ελεγκτών, ενώ ένας ελεγκτής επικοινωνεί μόνο με τα μοντέλα και τις όψεις που είναι υπεύθυνα για τις αντίστοιχες λειτουργίες, αλλά όχι με άλλους ελεγκτές για να αποφεύγονται οριζόντιες αλληλεξαρτήσεις που δημιουργούν πολυπλοκότητες στην υλοποίηση. Συνολικά υλοποιήσαμε τους παρακάτω ελεγκτές:

- `\VClass\Controllers\Registration.php`: Διαχειρίζεται τις αιτήσεις για δημιουργία ενός νέου λογαριασμού χρήστη.
- `\VClass\Controllers>Login.php`: Διαχειρίζεται τις αιτήσεις για την αυθεντικοποίηση ενός υπάρχοντος χρήστη κατά τη σύνδεση.
- `\VClass\Controllers>PasswordReset.php`: Διαχειρίζεται τις αιτήσεις για την ανάκτηση νέου κωδικού πρόσβασης.
- `\VClass\Controllers\Homepage.php`: Διαχειρίζεται τις αιτήσεις για την εμφάνιση της αρχικής σελίδας της εφαρμογής.
- `\VClass\Controllers\Admin.php`: Διαχειρίζεται τις αιτήσεις για την διαχείριση των μαθημάτων.
- `\VClass\Controllers\Student.php`: Διαχειρίζεται τις αιτήσεις σχετικά με τις λειτουργίες των μαθητών.

Για να γνωρίζει η εφαρμογή το ποιος ελεγκτής είναι υπεύθυνος για μία λειτουργία χρησιμοποιούμε ένα δρομολογητή HTTP αιτήσεων που καθορίζει τους ελεγκτές και τις αντίστοιχες μεθόδους που θα καλούνται ανάλογα με την

HTTP μέθοδο και το URL της αίτησης. Για την υλοποίηση του δρομολογητή χρησιμοποιήθηκε η βιβλιοθήκη FastRoute<sup>6</sup>.

Τα **μοντέλα (models)** υλοποιούν τις λειτουργίες *CRUD* (Create, Read, Update and Delete) πάνω στη βάση δεδομένων, επομένως τα μοντέλα περιέχουν τη λογική της επεξεργασίας των δεδομένων. Για κάθε ελεγκτή υπάρχει το αντίστοιχο μοντέλο που εκτελεί τις μεθόδους του μόνο αν του ζητηθεί από τον ελεγκτή:

- `\VClass\Controllers\RegistrationModel.php`
- `\VClass\Controllers>LoginModel.php`
- `\VClass\Controllers>PasswordResetModel.php`
- `\VClass\Controllers\AdminModel.php`
- `\VClass\Controllers\StudentModel.php`
- `\VClass\Controllers\UserModel.php`

Βλέπουμε ότι υπάρχει ένα επιπλέον μοντέλο (`UserModel.php`) που περιέχει τις κοινές μεθόδους μεταξύ των καθηγητών και των μαθητών. Όταν ένα μοντέλο ολοκληρώσει την επεξεργασία των δεδομένων επιστρέφει το αποτέλεσμα στον ελεγκτή ο οποίος κάλεσε το μοντέλο.

Οι **όψεις (views)** είναι υπεύθυνες για την παρουσίαση του αποτελέσματος στον χρήστη. Μία όψη μπορεί να περιέχει και στατικά και δυναμικά στοιχεία. Για παράδειγμα ένα στατικό στοιχείο είναι ένα μενού πλοήγησης, ενώ η βαθμολογία ενός μαθήματος είναι παράδειγμα δυναμικού περιεχομένου που καθορίζει ο αντίστοιχος ελεγκτής. Επομένως οι όψεις μπορεί να υλοποιηθούν σαν templates πάνω στα οποία ο ελεγκτής μπορεί να αλλάξει τα δυναμικά πεδία. Το πλεονέκτημα της χρήσης templates είναι ότι η σχεδίαση των σελίδων μπορεί να αλλάξει χωρίς να χρειαστεί να πειραχθεί η λογική της εφαρμογής. Για τη

---

<sup>6</sup><https://github.com/nikic/FastRoute>

δημιουργία των templatesχρησιμοποιήθηκε η μηχανή Twig<sup>7</sup> που απλοποιεί εισαγωγή δυναμικού κώδικα σε στατικές HTML σελίδες.

---

<sup>7</sup><http://twig.sensiolabs.org/>