**ΥΠΟΕΡΓΟ 6 :**

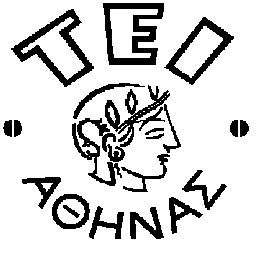
**«ΕΙΚΟΝΙΚΟ ΕΡΓΑΣΤΗΡΙΟ ΧΩΡΙΚΗΣ ΑΝΑΛΥΣΗΣ»**

**ΕΡΓΟ :**

**«ΥΠΗΡΕΣΙΕΣ ΕΙΚΟΝΙΚΩΝ ΕΡΓΑΣΤΗΡΙΩΝ ΤΟΥ ΤΕΙ ΑΘΗΝΑΣ» (MIS 304191)**

**Τεχνικός Οδηγός**

**(Maintainers' Guide)**

****

**ΤΕΙ ΑΘΗΝΑΣ**

**ΑΙΓΑΛΕΩ, ΑΥΓΟΥΣΤΟΣ 2015**

Περιεχόμενα

[Εισαγωγή 3](#_Toc427609247)

[Τεχνολογίες & Αρχιτεκτονική 3](#_Toc427609248)

[Υποσύστημα Βιβλιοθήκης Εννοιών - Τεχνικός Σχεδιασμός 4](#_Toc427609249)

[Οντότητες 4](#_Toc427609250)

[Διάγραμμα Πινάκων 5](#_Toc427609251)

[Υποσύστημα Εικονικών Ασκήσεων - Τεχνικός Σχεδιασμός 6](#_Toc427609252)

[Οντότητες 6](#_Toc427609253)

[Διάγραμμα Πινάκων 7](#_Toc427609254)

[Πηγαίος Κώδικας 8](#_Toc427609255)

[RESTful API 8](#_Toc427609256)

[Ελάχιστες Απαιτήσεις 8](#_Toc427609257)

[Οδηγίες Εγκατάστασης 9](#_Toc427609258)

[Τεχνική Υποστήριξη 9](#_Toc427609259)

## Εισαγωγή

Ο συγκεκριμένος οδηγός απευθύνεται σε τεχνικά καταρτισμένους χρήστες (τόσο προγραμματιστές όσο και διαχειριστές συστημάτων) οι οποίοι θα κληθούν να διαχειριστούν και/ή να συντηρήσουν την υπηρεσία. Περιλαμβάνει μια γενική περιγραφή της αρχιτεκτονικής της υπηρεσίας και τις τεχνολογίες από τις οποίες αποτελείται, τα ελάχιστα χαρακτηριστικά που πρέπει να έχει ένας server για την φιλοξενήσει καθώς και οδηγίες εγκατάστασης σε περίπτωση που χρειαστεί να μεταφερθεί σε νέα υπολογιστική υποδομή.

## Τεχνολογίες & Αρχιτεκτονική

Η υπηρεσία έχει αναπτυχθεί με τη χρήση των παρακάτω κύριων τεχνολογιών.

* PHP 5.4
* Symfony MVC Framework 2.6
* Doctrine ORM 2.5 (Βάση Δεδομένων MySQL)
* Sonata Admin Bundle ως διαχειριστικό περιβάλλον
* Twitter Bootstrap CSS Framework 3

Για την ανάπτυξη και των δύο υποσυστημάτων χρησιμοποιήθηκε η αρχιτεκτονική MVC, η οποία επιτρέπει τον διαχωρισμό της επιχειρησιακής λογικής (Business Logic - **Model**) από την παρουσίαση (**View**) με τη βοήθεια ενός ελεγκτή (**Controller**) ο οποίος αναλαμβάνει να δέχεται και να διεκπεραιώνει τα web αιτήματα των χρηστών. Η επικοινωνία με τη βάση δεδομένων γίνεται με τη βοήθεια του Doctrine ORM (**Object Relational Mapper**), το οποίο επιτρέπει την απευθείας αντιστοίχηση πινάκων της βάσης δεδομένων με κλάσεις του επίπεδου Model που αναφέρθηκε παραπάνω. Η συγκεκριμένη αντιστοίχηση δίνει το λεγόμενο "**persistence**" στις συγκεκριμένες οντότητες. Για το front-end χρησιμοποιήθηκε το Twitter Bootstrap CSS Framework το οποίο παρέχει εργαλεία για το σχεδιασμό φιλικών προς τον χρήστη και προσαρμοστικών (**responsive**) web εφαρμογών οι οποίες εμφανίζονται το ίδιο καλά σε όλα τα μεγέθη οθονών (από κινητές συσκευές/tablets έως και μεγάλες οθόνες).

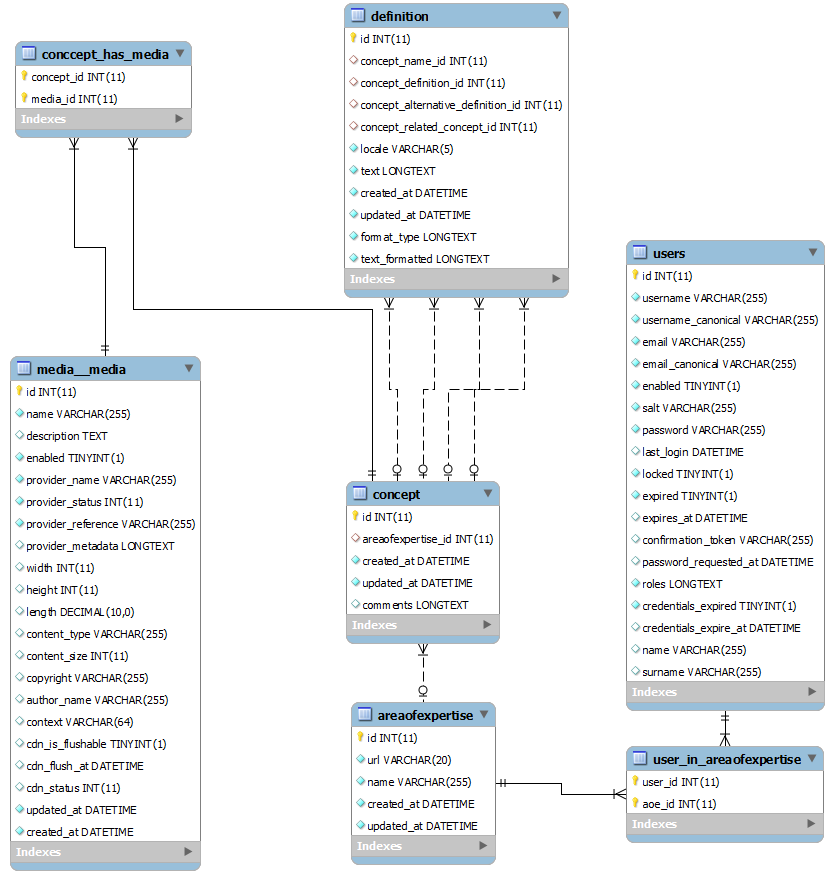
# Υποσύστημα Βιβλιοθήκης Εννοιών - Τεχνικός Σχεδιασμός

## Οντότητες

Η Βιβλιοθήκη Εννοιών αποτελείται από τις εξής οντότητες:

* **Γνωστικό Αντικείμενο** (Area of Expertise): Ένα γνωστικό αντικείμενο, το οποίο μπορεί να περιλαμβάνει έννοιες καθώς και εγγεγραμμένους χρήστες
* **Έννοια** (Concept): Μια έννοια, η οποία μπορεί να έχει ένα ή περισσότερα ονόματα σε διαφορετικές γλώσσες (αλλά μόνο ένα σε κάθε γλώσσα), έναν ορισμό και έναν ή περισσότερους εναλλακτικούς ορισμούς κ.α. Η κάθε έννοια ανήκει υποχρεωτικά σε ένα γνωστικό αντικείμενο (έννοιες που ανήκουν σε πολλαπλά καταχωρούνται μια φορά για κάθε γνωστικό αντικείμενο στο οποίο ανήκουν).
* **Ορισμός** (Definition): Βοηθητική οντότητα η οποία έχει σχέση Ν-1 με την Έννοια. Η οντότητα Ορισμός περιλαμβάνει το "περιεχόμενο" ενός ορισμού και αφορά μια συγκεκριμένη γλώσσα. Επιπλέον περιλαμβάνει ιδιότητες που αφορούν τη μορφοποίηση του κειμένου που περιέχει.
* **Μέσο** (Media): Βοηθητική οντότητα η οποία έχει σχέση Ν-1 με την Έννοια. Η συγκεκριμένη οντότητα συμμετέχει στην αποθήκευση οπτικών δεδομένων όπως εικόνες, video, αρχεία κτλ. τα οποία μπορούν να συνδεθούν με μια έννοια.

## Διάγραμμα Πινάκων



# Υποσύστημα Εικονικών Ασκήσεων - Τεχνικός Σχεδιασμός

## Οντότητες

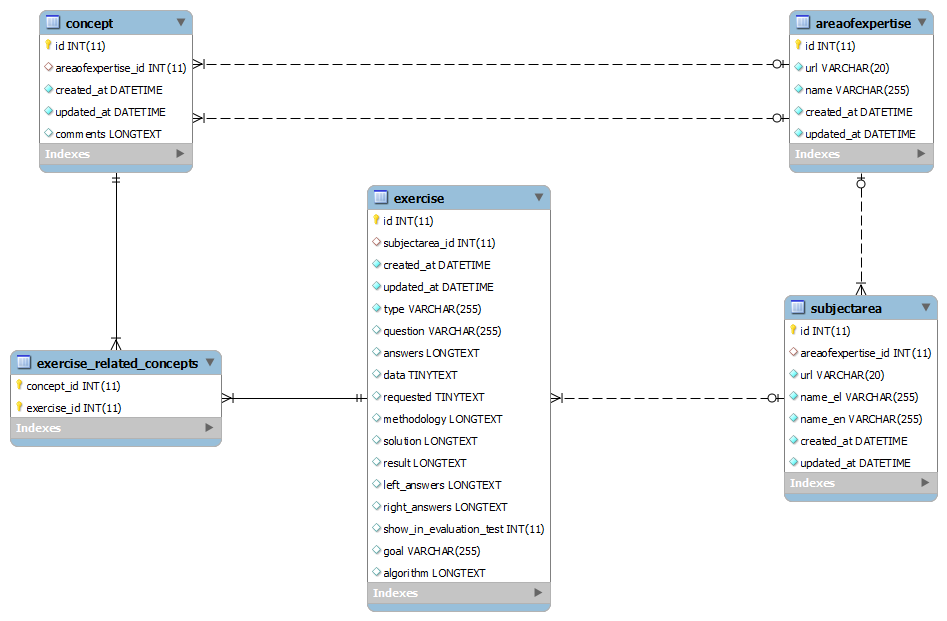
Το υποσύστημα εικονικών ασκήσεων αποτελείται από τις εξής οντότητες:

* **Θεματική Ενότητα** (Subject Area): Μια θεματική ενότητα η οποία ανήκει σε ένα γνωστικό αντικείμενο. Χρησιμοποιείται για να κατηγοριοποιήσει ασκήσεις σε ένα δεύτερο επίπεδο πέραν του γνωστικού αντικειμένου.
* **Βασική Άσκηση** (Exercise): Η βασική άσκηση αποτελεί μια αφηρημένη (abstract) οντότητα από την οποία κληρονομούν όλοι οι τύποι εικονικών ασκήσεων. Περιλαμβάνει κοινά πεδία όπως τη θεματική περιοχή στην οποία ανήκει η άσκηση και τυχόν σχετικές έννοιες (related concepts).
* **Άσκηση Πολλαπλής Επιλογής** (MultipleChoice): Οι ασκήσεις πολλαπλής επιλογής περιλαμβάνουν έναν αριθμό από πιθανές απαντήσεις, από τις οποίες μια χαρακτηρίζεται ως σωστή. Σε επίπεδο βάσης δεδομένων, οι πιθανές απαντήσεις καθώς και η σωστή απάντηση αποτυπώνονται σε μια στήλη η οποία περιέχει δεδομένα σε JSON-encoded μορφή. Για κάθε απάντηση ορίζεται το πεδίο is\_correct που δηλώνει αν η απάντηση είναι σωστή ή οχι.
* **Άσκηση Σωστό/Λάθος** (OnOff): Οι ασκήσεις σωστό λάθος αποτελούν μια υποκατηγορία των ασκήσεων Πολλαπλής Επιλογής. Η μόνη διαφοροποίηση είναι ότι στις ασκήσεις Σωστό/Λάθος υπάρχουν μόνο οι επιλογές Σωστό ή Λάθος.
* **Άσκηση Αντιστοίχησης** (Matching): Οι ασκήσεις αντιστοίχησης αποτελούνται από δύο στήλες: την αριστερή και τη δεξιά. Για κάθε επιλογή της αριστερής στήλης ορίζεται ένας πίνακας "matches" ο οποίος δηλώνει με ποιές επιλογές τις δεξιάς στήλης ταιριάζει (μια ή περισσότερες). Για να χαρακτηριστεί η άσκηση σωστή πρέπει για κάθε επιλογή της αριστερής στήλης ο χρήστης να έχει επιλέξει τις κατάλληλες αντιστοιχήσεις με τη δεξιά.
* **Λυμένες Ασκήσεις** (Solved): Οι λυμένες ασκήσεις περιλαμβάνουν τα πεδία Στόχος, Ερώτηση, Δεδομένα, Ζητούμενα, Μεθοδολογία, Αλγόριθμος, Επίλυση, Αποτέλεσμα. Οι συγκεκριμένες ασκήσεις έχουν σχεδιαστεί για να εμφανίζονται με δομή αντίστοιχη της βιβλιοθήκης εννοιών, με μέρη τα οποία αρχικά είναι κρυμμένα (collapsed) και τα οποία μπορεί ο χρήστης να εμφανίσει (expand).
* **Μη Λυμένες Ασκήσεις** (Unsolved): Οι μη λυμένες ασκήσεις έχουν αντίστοιχα χαρακτηριστικά με τις Λυμένες Ασκήσεις, με τη διαφορά ότι περιέχουν λιγότερα πεδία. Εμφανίζονται το ίδιο σε επίπεδο εμφάνισης.

Σημειώνεται ότι οι ασκήσεις αξιολόγησης δεν αποτελούν ξεχωριστό τύπο ασκήσεων αλλά προέρχονται από τις ήδη καταχωρημένες ασκήσεις Πολλαπλής Επιλογής, Σωστό Λάθος και/ή Αντιστοίχησης. Κάθε άσκηση αυτού του τύπου περιλαμβάνει ένα πεδίο "showInEvaluationTests", το οποίο μπορεί να περιέχει τις τιμές: 1 (απλή άσκηση - μη εμφάνιση στις ασκήσεις αξιολόγησης), 2 (εμφάνιση μόνο στις ασκήσεις αξιολόγησης), 3 (εμφάνιση και στις απλές ασκήσεις και στις ασκήσεις αξιολόγησης).

## Διάγραμμα Πινάκων

Όλες οι ασκήσεις καταχωρούνται στον πίνακα BaseExercise και διαχωρίζονται μέσω της στήλης "type". Για τα πεδία που δε χρησιμοποιούνται σε έναν συγκεκριμένο τύπο άσκησης οι σχετικές στήλες έχουν τιμή NULL.



# Πηγαίος Κώδικας

Ο πηγαίος κώδικας της εφαρμογής αναρτάται τακτικά στην τοποθεσία <https://github.com/teiath/vispanlab>. Επιπλέον, στην τοποθεσία <https://github.com/teiath/vispanlab/commits/master> παρέχεται πλήρες ιστορικό όλων των αλλαγών που έγιναν στον κώδικα κατά την ανάπτυξη της εφαρμογής.

# RESTful API

Η υπηρεσία παρέχει προγραμματιστική διεπαφή μέσω RESTful web services τα οποία τεκμηριώνονται στην τοποθεσία http://spatial.lab.teiath.gr/api/doc/ . Μέσω της συγκεκριμένης διεπαφής μπορεί να γίνει ανάκτηση και διαχείριση των οντοτήτων του συστήματος σε μορφή JSON ή XML (αρκεί η προσθήκη της κατάληξης στο URL του πόρου πχ. GET http://spatial.lab.teiath.gr/api/concepts/planning.**json** για ανάκτηση των εννοιών του γνωστικού αντικείμενου "Πολεοδομία" σε μορφή JSON).

# Ελάχιστες Απαιτήσεις

Τα παρακάτω αποτελούν τις ελάχιστες απαιτήσεις της εφαρμογής με βάση τον αναμενόμενο φόρτο ενός τυπικού σεναρίου χρήσης (~50 ταυτόχρονα συνδεδεμένοι χρήστες):

**Υλικό:**

* 2 CPU cores με τουλάχιστον 1.5GHz ο καθένας
* 2GB μνήμη
* 20GB χώρος δίσκου

**Λογισμικό:**

* Λειτουργικό Σύστημα Linux (προτείνεται Scientific Linux, αλλά θα λειτουργεί με οποιαδήποτε διανομή Linux)
* Apache 2.2+ web server
* PHP 5.4+ με τις επεκτάσεις curl, pdo και mysql
* MySQL 5.x

# Οδηγίες Εγκατάστασης

Τα βήματα για μια νέα εγκατάσταση της πλατφόρμας είναι τα εξής:

1. Κάνουμε clone τον πηγαίο κώδικα από το repository του Vispanlab σε έναν κατάλογο του server
2. Αντιγράφουμε το αρχείο app/config/parameters.yml.dist σε app/config/parameters.yml και ορίζουμε τις παραμέτρους που αφορούν την πρόσβαση στη βάση δεδομένων, αποστολή email, analytics κτλ.
3. Κατεβάζουμε το composer από την τοποθεσία https://getcomposer.org/ στον ίδιο κατάλογο με την εφαρμογή
4. Εκτελούμε την εντολή php composer.phar update ώστε να εγκατασταθούν τα απαραίτητα dependencies για την εκτέλεση της εφαρμογής
5. Εκτελούμε την εντολή php app/console assets:install για να εγκατασταθούν τα απαραίτητα assets (CSS, εικόνες κτλ.)
6. Ορίζουμε ως document root του web server τον φάκελο web μέσα στο Vispanlab
7. Σε αυτό το σημείο η εφαρμογή θα πρέπει να είναι προσβάσιμη. Ελέγξτε ότι λειτουργεί η αυθεντικοποίηση. Ο αρχικός διαχειριστής (Υπεύθυνος ΕΕΧΑ) μπορεί να δημιουργηθεί με την εντολή fos:user:create <username> --super

Επιπλέον, για τη διευκόλυνση των deployments και των αναβαθμίσεων, η εφαρμογή μπορεί να εγκατασταθεί και με τη χρήση του εργαλείου Capifony - οι ρυθμίσεις που αφορούν την εγκατάσταση μέσω Capifony μπορούν να οριστούν στο αρχείο app/config/deploy.rb .

# Τεχνική Υποστήριξη

Για οποιαδήποτε ερώτηση δεν καλύπτεται από τον συγκεκριμένο οδηγό, μπορείτε να επικοινωνείτε μέσω email στη διεύθυνση [dnna@dnna.gr](mailto:dnna@dnna.gr) .