







# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ  
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ  
ΥΠΟΛΟΓΙΣΤΩΝ

Ανάπτυξη Συστήματος Ηλεκτρονικής Μάθησης - Αξιοποίηση Βάσης  
Δεδομένων με Ιστορικά και Πολιτισμικά Στοιχεία

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Εμμανουήλ Ι. Θάνος

Επιβλέπων: Γιάννης Μαΐστρος  
Επίκουρος Καθηγητής Ε.Μ.Π.

Εξωτερικοί Επιβλέποντες (Universitat Politècnica de València) :

Eva Onaindia

Antonio Garrido

Αν. Καθηγήτρια U.P.V.

Αν. Καθηγητής U.P.V.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 29η Απριλίου 2013.

.....

Γιάννης Μαΐστρος

Επ. Καθηγητής Ε.Μ.Π.

.....

Στέλλα Μαρκαντωνάτου

Κύρια Ερευνήτρια Ι.Ε.Λ.

.....

Κώστας Κοντογιάννης

Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Απρίλιος 2013

.....

Εμμανουήλ Ι. Θάνος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Εμμανουήλ Ι. Θάνος, 2013

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς το συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

# Περίληψη

Σκοπός της διπλωματικής αυτής εργασίας είναι ο σχεδιασμός και η ανάπτυξη Συστήματος Ηλεκτρονικής Μάθησης (ΣΗΜ) που αξιοποιεί τεκμηριωμένο υλικό πολιτιστικής βάσης δεδομένων. Η βάση δεδομένων που χρησιμοποιήσαμε είναι αυτή του Μουσείου του Κοργιαλένιου Ιδρύματος της Κεφαλλονιάς, η οποία περιέχει ψηφιοποιημένο υλικό και πληροφορίες για τη σύγχρονη ιστορία του νησιού.

Στο πλαίσιο της εργασίας μελετήσαμε τις προδιαγραφές του Σχεδιασμού Ηλεκτρονικής Μάθησης (e-learning planning) και τις μεθόδους που χρησιμοποιεί για την εξατομίκευση της Ηλεκτρονικής Μάθησης (e-learning) ανάλογα με τις εκπαιδευτικές ανάγκες μαθητών, ή γενικότερα εκπαιδευομένων. Επεκτείναμε τη βάση δεδομένων του Μουσείου ώστε να καλύψει την εκπαιδευτική χρήση της, αναπτύσσοντας μια καινούργια βάση την οποία και χρησιμοποιεί το σύστημά μας.

Το ΣΗΜ σχεδιάστηκε και αναπτύχθηκε με το Σύστημα Διαχείρισης Περιεχομένου (CMS) **Joomla! 2.5** και οι λειτουργίες του διαμορφώθηκαν ως ξεχωριστά αρχεία γλώσσας **PHP**. Κεντρική λειτουργία του είναι η παραγωγή εκπαιδευτικών διαδρομών από τους εκπαιδευτικούς, με σκοπό την ανάθεσή τους στους μαθητές τους. Οι διαδρομές αυτές αποτελούνται από αντικείμενα μάθησης, τα οποία αντιστοιχούν στα διαφορετικά εκθέματα του μουσείου, με τις εικόνες και τις πληροφορίες που τα συνοδεύουν στη βάση δεδομένων.

Κατά τη δημιουργία μιας διαδρομής, ο χρήστης-εκπαιδευτικός μέσα από πλούσιο σύνολο επιλογών καθορίζει τις κατηγορίες αντικειμένων που θα περιέχει η ψηφιακή διαδρομή και τα εκπαιδευτικά χαρακτηριστικά της, με βάση τις μαθησιακές ανάγκες και προτιμήσεις των μαθητών στους οποίους πρόκειται να την αναθέσει. Το ΣΗΜ επεξεργάζεται τις πληροφορίες αυτές και δημιουργεί δύο αρχεία γλώσσας **PDDL**, και αυτά εισάγονται στο σχεδιαστή (planner) **LPG-td-1.0**, ο οποίος και παράγει την τελική διαδρομή.

Παράλληλα, το ΣΗΜ προσφέρει και άλλες εκπαιδευτικές λειτουργίες, όπως τη δημιουργία σχετικών διαγωνισμάτων για να απαντηθούν διαδραστικά από τους μαθητές, την αξιολόγηση των στοιχείων της βάσης δεδομένων ως αντικειμένων μάθησης κατά την προετοιμασία του συστήματος και τη γενικότερη παρακολούθηση της πορείας των μαθητών. Απευθύνεται κυρίως σε εκπαιδευτικούς και μαθητές πρωτοβάθμιας εκπαίδευσης, αλλά αποτελεί ένα παράδειγμα αξιοποίησης πολιτιστικών βάσεων δεδομένων για εκπαιδευτικούς σκοπούς.

## Λέξεις-Κλειδιά

Σύστημα Ηλεκτρονικής Μάθησης, Σχεδιασμός Ηλεκτρονικής Μάθησης, Σχεδιαστής, Γλώσσα PDDL, Γράφοι Σχεδιασμού, Αντικείμενα Μάθησης, Εκπαιδευτικές Διαδρομές, Πολιτιστικές Βάσεις Δεδομένων



## Abstract

Purpose of this project is the design and implementation of a Learning Management System (LMS) utilizing documented material of a cultural database. The database used is the one of the Museum of the **Korgialeneion Foundation of Kefalonia**, which contains digitized material and information regarding the modern history of the island.

As part of the project, we studied the specifications of **e-learning planning** and the methods used for the personalization of the e-learning process according to the educational needs of the students. We extended the Museum's database in order to cover its educational use, developing a new database to be used by our system.

The LMS was designed and developed with the **Joomla! 2.5** Content Management System and its functions were formed as separate PHP files. Its main function is the creation of learning routes and their assignment to students by the teachers. These routes consist of Learning Objects that correspond to the exhibits of the museum, along with images and information related to them in the database.

When creating a route, the teacher-user defines through a variety of options the categories of the learning objects the route is going to contain, as well as its educational features, based on the learning preferences of the students to be assigned this route. The LMS processes this information and produces two PDDL files that are used as the inputs to the **LPG-td-1.0** planner, which in turn produces the final route.

The LMS also offers other educational functions, such as the creation of tests to be answered by the students, the evaluation of the items of the database as learning objects, at the preparatory phase, and the monitoring of the students' tasks. It is primarily offered to teachers and students of primary school, but is also an example of utilizing cultural databases for educational purpose.

### Keywords

Learning Management System, e-learning planning, Planner, PDDL Language, Planning graphs, Learning Objects, Learning routes, Cultural Databases





# Περιεχόμενα

<b>1</b>	<b>ΕΙΣΑΓΩΓΗ.....</b>	<b>13</b>
1.1	ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ .....	13
1.2	ΟΡΓΑΝΩΣΗ ΤΟΥ ΤΟΜΟΥ.....	14
<b>2</b>	<b>ΤΕΧΝΟΛΟΓΙΕΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΜΑΘΗΣΗΣ .....</b>	<b>15</b>
2.1	ΗΛΕΚΤΡΟΝΙΚΗ ΜΑΘΗΣΗ (E-LEARNING).....	15
2.1.1	Συστήματα Ηλεκτρονικής Μάθησης.....	15
2.1.2	Εφαρμογές στην Τέχνη και την Ιστορία.....	15
2.2	ΣΧΕΔΙΑΣΜΟΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ .....	17
2.2.1	Γλώσσες αναπαράστασης προβλημάτων σχεδιασμού .....	18
2.2.2	Η γλώσσα PDDL.....	20
2.2.3	Ευρετικές Συναρτήσεις και Γράφοι Σχεδιασμού .....	24
2.2.4	Σχεδιασμός με γράφους .....	26
2.2.5	Ο σχεδιαστής LPG.....	27
2.3	ΠΡΟΔΙΑΓΡΑΦΕΣ ΤΟΥ ΣΧΕΔΙΑΣΜΟΥ ΗΛΕΚΤΡΟΝΙΚΗΣ ΜΑΘΗΣΗΣ.....	29
<b>3</b>	<b>ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ .....</b>	<b>31</b>
3.1	ΑΡΧΙΚΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΤΟΥ ΚΟΡΓΙΑΛΕΝΕΙΟΥ ΜΟΥΣΕΙΟΥ .....	31
3.2	Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ CORGIALEN .....	35
3.2.1	Αντικείμενα Μάθησης (Learning Objects)-Μαθησιακά Χαρακτηριστικά (Learning Attributes).....	36
3.2.2	Πίνακες Διαχείρισης.....	40
<b>4</b>	<b>ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ .....</b>	<b>43</b>
4.1	ΧΡΗΣΤΕΣ-ΕΠΙΠΕΔΑ ΠΡΟΣΒΑΣΗΣ .....	43
4.1.1	Public .....	43
4.1.2	Registered .....	43
4.1.3	Student .....	44
4.1.4	Teacher .....	44
4.1.5	Expert.....	44
4.1.6	Super User-Administrator.....	44
4.2	ΛΕΙΤΟΥΡΓΙΕΣ.....	45
4.2.1	e-learning.....	46
4.2.2	Γενικές Πληροφορίες.....	57
4.2.3	User Menu.....	57
4.2.4	Login.....	57
<b>5</b>	<b>ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ .....</b>	<b>59</b>
5.1	WEBMATRIX .....	59
5.2	JOOMLA!.....	60
5.2.1	Περιεχόμενα Ιστοσελίδας.....	61
5.2.2	Η Βάση Δεδομένων του Joomla.....	63
5.2.3	Articles .....	65
5.2.4	Menus.....	69

5.2.5	<i>Χρήστες-Επίπεδα Πρόσβασης</i> .....	73
<b>6</b>	<b>ΑΝΑΛΥΣΗ ΛΕΙΤΟΥΡΓΙΩΝ ΑΡΧΕΙΩΝ PHP</b> .....	<b>76</b>
6.1	ΔΗΜΙΟΥΡΓΙΑ ΔΙΑΔΡΟΜΩΝ.....	76
6.1.1	<i>TeachersRoutes.php</i> .....	76
6.1.2	<i>Αρχεία PDDL</i> .....	79
6.1.3	<i>AssignRoutesToStudents.php</i> .....	84
6.2	ΤΕΚΜΗΡΙΩΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ ΜΑΘΗΣΗΣ.....	86
6.2.1	<i>ExpertChanges.php</i> .....	86
6.3	ΔΙΑΔΡΟΜΕΣ.....	87
6.3.1	<i>Course.php</i> .....	87
6.3.2	<i>Routeview.php</i> .....	87
6.3.3	<i>Objectinfo.php</i> .....	88
6.4	ΔΙΑΓΩΝΙΣΜΑΤΑ.....	89
6.4.1	<i>TestEditor.php</i> .....	89
6.4.2	<i>AssignTestToStudents.php</i> .....	90
6.4.3	<i>TestToChoose.php</i> .....	90
6.4.4	<i>TestToDo.php</i> .....	90
6.5	ΟΙ ΜΑΘΗΤΕΣ ΜΟΥ.....	91
6.5.1	<i>StudentsToTeachers.php</i> .....	91
6.5.2	<i>StudentRoutes.php</i> .....	91
6.5.3	<i>StudentTests.php</i> .....	92
<b>7</b>	<b>ΕΠΙΛΟΓΟΣ</b> .....	<b>93</b>
7.1	ΣΥΝΟΨΗ.....	93
7.2	ΠΡΟΕΚΤΑΣΕΙΣ.....	94
<b>8</b>	<b>ΒΙΒΛΙΟΓΡΑΦΙΑ</b> .....	<b>96</b>

## Κατάλογος Σχημάτων

PDDL Domain File 2.2-1.....	21
PDDL Problem File 2.2-2.....	23
Planning Graph 2.2-3.....	25
Σχεδιασμός Ηλεκτρονικής Μάθησης 2.3-1 .....	30
Korgialenios E-R diagram 3.1-1 .....	32
Πίνακες από Korgialenios DB 3.1-2.....	34
Corgialen E-R Diagram 3.2-1 .....	35
Corgialen.Object 3.2-2.....	39
Corgialen.Information 3.2-3 .....	39
Corgialen.Photos 3.2-4 .....	39
Corgialen.Routes 3.2-5 .....	40
Corgialen.StudentRoutes 3.2-6.....	40
Corgialen.TeacherStudents 3.2-7.....	41
Corgialen.Tests 3.2-8.....	41
Corgialen.questions 3.2-9 .....	41
Corgialen.testque 3.2-10.....	42
Corgialen.Studenttests 3.2-11 .....	42
Κατηγορίες Επιπέδων Πρόσβασης 4.1-1 .....	43
Αρχική Σελίδα Εφαρμογής 4.2-1 .....	45
Δημιουργία Διαδρομών 4.2-2 .....	46
Assign Route to Students 4.2-3.....	49
Τεκμηρίωση Αντικειμένων Μάθησης 4.2-4 .....	50
Νέο Διαγώνισμα 4.2-5 .....	51
Διαμόρφωση διαγωνίσματος 4.2-6 .....	51
Assign Test to students 4.2-7 .....	52
Τα τεστ μου 4.2-8 .....	52
Test To Do 4.2-9.....	53
Αποτελέσματα απαντήσεων 4.2-10 .....	53
Courses 4.2-11 .....	54
Route View 4.2-12.....	55
ObjectInfo 4.2-13.....	55
SelectStudents 4.2-14 .....	56
Student Tests 4.2-15 .....	56
Student Routes 4.2-16.....	57
Contact 4.2-17.....	57
Your Profile 4.2-18.....	58
Login 4.2-19 .....	58
WebMatrix 5.1-1.....	59
Joomla Directory 5.2-1 .....	61
Site Administration 5.2-2.....	62
Joomla\administrator 5.2-3 .....	62
Joomla Database E-R Diagram 5.2-4.....	63

Featured Articles 5.2-5 .....	65
Edit Article 5.2-6 .....	66
Categories 5.2-7 .....	67
Edit Article 5.2-8 .....	67
Menus 5.2-9 .....	69
Menu Manager 5.2-10.....	69
our articles' items 5.2-11.....	70
About Kefalonia items 5.2-12.....	71
Edit Menu Item 5.2-13.....	71
Module Manager 5.2-14 .....	72
Position-7 5.2-15.....	72
User Manager 5.2-16 .....	73
Edit User 5.2-17.....	73
User Groups 5.2-18.....	74
Edit Group 5.2-19 .....	74
Viewing Access Levels 5.2-20 .....	75
Edit Viewing Access Levels 5.2-21.....	75
Domain PDDL File 6.1-1.....	80
Problem PDDL File 6.1-2.....	82

# 1 ΕΙΣΑΓΩΓΗ

## 1.1 ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη μιας διαδικτυακής πλατφόρμας, η οποία προσφέρει ένα περιβάλλον ηλεκτρονικής μάθησης, βασισμένο στα στοιχεία πολιτιστικής κληρονομιάς που περιέχει η βάση δεδομένων του μουσείου του Κοργιαλένιου Ιδρύματος της Κεφαλλονιάς. Η εφαρμογή αξιοποιεί πληροφορίες και ψηφιοποιημένο υλικό από το Κοργιαλένιο Μουσείο και μπορεί να χρησιμοποιηθεί για εκπαιδευτικούς σκοπούς εκμάθησης ιστορίας και πολιτισμικών στοιχείων της περιοχής, από δασκάλους και μαθητές πρωτοβάθμιας εκπαίδευσης.

Η εργασία πραγματοποιήθηκε κατά ένα μέρος στο Πολυτεχνείο της Βαλένθια (Universitat Politècnica de València) το εαρινό εξάμηνο του 2011 (2/2011-7/2011), υπό την επίβλεψη των καθηγητών Eva Onaindia και Antonio Garrido, και κατά το υπόλοιπο στο ΕΜΠ, υπό την επίβλεψη του Καθηγητή Γιάννη Μαΐστρου και με τη συνεργασία της Δρος Στέλλας Μαρκαντωνάτου (ΙΕΛ), της Καθηγήτριας Κλειώς Σγουροπούλου και της Κας Κατερίνας Μαραβέγια (Κοργιαλένιο Ίδρυμα).

Για την ανάπτυξη του συστήματος αυτού χρησιμοποιήθηκαν:

- Το εργαλείο ανάπτυξης διαδικτυακών εφαρμογών Microsoft WebMatrix.
- Το σύστημα διαχείρισης περιεχομένου (CMS) Joomla! 2.5.
- Η γλώσσα συγγραφής σεναρίων PHP.
- Η γλώσσα υπερκειμένων HTML.
- Η γλώσσα προγραμματισμού PDDL (Planning Domain Definition Language).
- Το σύστημα διαχείρισης βάσεων δεδομένων MySQL.
- Ο σχεδιαστής lpg-td-1.0.

## 1.2 ΟΡΓΑΝΩΣΗ ΤΟΥ ΤΟΜΟΥ

Ο τόμος έχει οργανωθεί στα ακόλουθα κεφάλαια:

Στο κεφάλαιο 2 γίνεται γενική αναφορά στις μεθόδους ηλεκτρονικής μάθησης και το Σχεδιασμό Τεχνητής Νοημοσύνης και περιγράφονται αναλυτικά η γλώσσα PDDL, η λειτουργία του Iprg-td και οι προδιαγραφές του Σχεδιασμού Ηλεκτρονικής Μάθησης.

Στο κεφάλαιο 3 παρουσιάζονται οι βάσεις δεδομένων που δημιουργήθηκαν και χρησιμοποιήθηκαν.

Στο κεφάλαιο 4 περιγράφεται η μορφή και οι λειτουργίες των ιστοσελίδων που δημιουργήθηκαν.

Στο κεφάλαιο 5 παρουσιάζεται ο σχεδιασμός του συστήματος και ο συνδυασμός των συστατικών μερών του μέσω του συστήματος διαχείρισης περιεχομένου Joomla.

Στο κεφάλαιο 6 περιγράφονται αναλυτικά τα αρχεία PHP που περιέχουν το σύνολο των βασικών λειτουργιών της εφαρμογής.

Τέλος, στο κεφάλαιο 7 γίνεται μια πρώτη αξιολόγηση των τρόπων αξιοποίησης του συστήματος και των πιθανών μελλοντικών επεκτάσεων.

## 2 ΤΕΧΝΟΛΟΓΙΕΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΜΑΘΗΣΗΣ

### 2.1 ΗΛΕΚΤΡΟΝΙΚΗ ΜΑΘΗΣΗ (e-learning)

**Ηλεκτρονική Μάθηση (e-learning)** είναι η διαδικασία εκπαίδευσης με χρήση Τεχνολογιών Πληροφορίας και Επικοινωνιών. Λόγω της ανάπτυξης των μέσων αυτών και της ευρείας χρήσης του διαδικτύου, η πλειονότητα των διαδικασιών «ηλεκτρονικής μάθησης» πραγματοποιούνται εξ αποστάσεως («εξ αποστάσεως εκπαίδευση» - distant learning).

Στην πραγματικότητα, η ηλεκτρονική μάθηση περιλαμβάνει όλα τα ηλεκτρονικά μέσα που αναπαράγουν ήχο και προβάλλουν κείμενο, εικόνα, φωτογραφίες και βίντεο, και μπορεί να λαμβάνει χώρα μέσα ή έξω από την αίθουσα διδασκαλίας. Μπορεί ακόμη να είναι ασύγχρονη, με την πορεία και τα χρονικά της πλαίσια να αφήνονται στην ευχέρεια του εκπαιδευόμενου, ή σύγχρονη, ρυθμιζόμενη στην περίπτωση αυτή από τον εκπαιδευτή-καθοδηγητή.

Λόγω της διαδεδομένης χρήσης της στην εκπαιδευτική διαδικασία και του σημαντικού της ρόλου σήμερα, η ηλεκτρονική μάθηση έχει αναχθεί σε διεπιστημονικό πεδίο που αξιοποιεί την ανάπτυξη των νέων τεχνολογιών και ενσωματώνει τεχνικές από πολλές διαφορετικές επιστήμες - όπως μεταξύ άλλων εκπαιδευτικές θεωρίες, αναγνώριση και μοντελοποίηση προφίλ (profiling), αναπαράσταση γνώσης και μεθόδους Τεχνητής Νοημοσύνης.<sup>[2]</sup>

#### 2.1.1 Συστήματα Ηλεκτρονικής Μάθησης

Σύστημα Ηλεκτρονικής Μάθησης (Learning Management System) είναι μια εφαρμογή λογισμικού που στοχεύει στη διαχείριση, την οργάνωση, την παράδοση και την αξιολόγηση σειράς μαθημάτων ή εκπαιδευτικού προγράμματος. Τα συστήματα ηλεκτρονικής μάθησης παρέχουν στους συμμετέχοντες στην εκπαιδευτική διαδικασία ένα πλαίσιο λειτουργιών που διευκολύνουν τις παραπάνω ενέργειες, όπως τη διανομή εκπαιδευτικού υλικού, τον καθορισμό των εκπαιδευτικών στόχων και τη γενικότερη επίβλεψη της πορείας των μαθημάτων μέσω της συλλογής και παρουσίασης δεδομένων. Τα πιο διαδεδομένα συστήματα ηλεκτρονικής μάθησης γενικής χρήσης είναι το Blackboard και το Moodle.<sup>[2]</sup>

#### 2.1.2 Εφαρμογές στην Τέχνη και την Ιστορία

Τα περισσότερα από τα διασημότερα παγκοσμίως μουσεία, διαθέτουν σήμερα ιστοσελίδες που δίνουν στον επισκέπτη τη δυνατότητα ψηφιακής περιήγησης στα εκθέματά τους. Ο επισκέπτης μπορεί να διαλέξει την κατηγορία εκθεμάτων για την οποία ενδιαφέρεται και να δει σχετικές φωτογραφίες και

πληροφορίες, κάτι που αναμφίβολα αποτελεί μια διαδικασία ηλεκτρονικής μάθησης, χωρίς βέβαια να εντάσσεται σε κάποια σειρά μαθημάτων ή τυπική εκπαιδευτική διαδικασία.

Πολλά πανεπιστήμια χρησιμοποιούν επίσης περιβάλλοντα e-learning που αξιοποιούν ψηφιοποιημένο υλικό από μουσεία και πινακοθήκες, κυρίως για σειρές μαθημάτων Ιστορίας της Τέχνης.

Σε μαθητές, ένα τέτοιο περιβάλλον μπορεί πολύ αποδοτικά να συνδυαστεί με το μάθημα της Ιστορίας. Η δυνητική (virtual) περιήγηση σε εκθέματα τέχνης και πολιτισμού μιας περιοχής σε μια ιστορική περίοδο, παράλληλα με τη διδασκαλία και την εκμάθηση των αντίστοιχων ιστορικών στοιχείων, θα μπορούσαν να κινήσουν ακόμη περισσότερο το ενδιαφέρον του μαθητή, με το συνδυασμό πληροφοριών και εικόνας, και να τον βοηθήσουν στη συνδυαστική απομνημόνευση των πληροφοριών μέσω μιας ευχάριστης για αυτόν εκπαιδευτικής διαδικασίας.



## 2.2 ΣΧΕΔΙΑΣΜΟΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

Ο «σχεδιασμός» ως έννοια γενικά αναφέρεται στη διαδικασία εύρεσης σειράς ενεργειών για την επίτευξη στόχου. **Σχεδιασμός** στην Τεχνητή Νοημοσύνη (AI Planning) είναι το πεδίο εκείνο που αναφέρεται στη δημιουργία και κατανόηση των στρατηγικών, οι οποίες θα επιτρέψουν σε κάποιο λογισμικό να βρει μέσα από σύνολο αυστηρά ορισμένων ενεργειών του «**κόσμου**» (domain) που μελετά, τη σειρά εκείνων που θα οδηγήσουν στην επίτευξη ενός ή περισσότερων προκαθορισμένων στόχων.

Δεδομένης μιας περιγραφής του συνόλου των πιθανών αρχικών καταστάσεων (initial states) του κόσμου, του συνόλου των απαιτούμενων στόχων (goals) και του συνόλου των πιθανών ενεργειών (actions), το **Πρόβλημα Σχεδιασμού** (Planning Problem) είναι το πρόβλημα εύρεσης σειράς ενεργειών, που εγγυημένα θα οδηγήσει από κάποια από τις αρχικές καταστάσεις σε κάποια από αυτές που καλύπτουν όλους τους στόχους. Κάθε ενέργεια μπορεί να εκτελεστεί μόνο σε συγκεκριμένο σύνολο καταστάσεων, αυτές που καλύπτουν τις προϋποθέσεις της (preconditions), ενώ η εκτέλεσή της προκαλεί κάποιο σύνολο αποτελεσμάτων (effects) στον κόσμο.

Τα προβλήματα σχεδιασμού μπορούν να κατηγοριοποιηθούν ανάλογα με τις ιδιότητες που διακρίνουν τα επιμέρους συστατικά τους. Για παράδειγμα, οι πιθανές ενέργειες μπορεί να είναι ντετερμινιστικές ή μη ντετερμινιστικές, να έχουν ή όχι συγκεκριμένη διάρκεια, να μπορούν να εκτελεστούν ταυτόχρονα ή να υπάρχουν περιορισμοί. Αντίστοιχα οι καταστάσεις μπορεί να είναι συνεχείς ή διακριτές και παρατηρήσιμες πλήρως ή μερικώς. Τέλος, ο στόχος μπορεί να αποτελείται από ζητούμενες καταστάσεις, από βελτιστοποιήσεις τιμών συναρτήσεων ή από συνδυασμό των δύο αυτών στοιχείων.

Η απλούστερη κατηγορία προβλημάτων σχεδιασμού είναι τα **Κλασικά Προβλήματα Σχεδιασμού** (Classical Planning Problems). Σε αυτήν ανήκει και το πρόβλημα σχεδιασμού των εκπαιδευτικών διαδρομών με το οποίο θα ασχοληθούμε. Τα χαρακτηριστικά της είναι μία μοναδική, γνωστή αρχική κατάσταση, και ντετερμινιστικές ενέργειες χωρίς ορισμένη διάρκεια, από τις οποίες μόνο μία μπορεί να εκτελείται κάθε στιγμή. Αφού η αρχική κατάσταση είναι γνωστή και οι ενέργειες ντετερμινιστικές, η κατάσταση του κόσμου μετά από την εκτέλεση οποιασδήποτε σειράς ενεργειών είναι πλήρως προβλέψιμη.

Τα λογισμικά που χρησιμοποιούνται για την επίλυση των προβλημάτων Σχεδιασμού Τεχνητής Νοημοσύνης αναφέρονται συνήθως ως **Σχεδιαστές** (Planners) και αποτελούν ειδική κατηγορία «έξυπνων προγραμμάτων» (intelligent agents). Είναι συστήματα προγραμματισμένα να αντιμετωπίζουν τα προβλήματα σχεδιασμού χρησιμοποιώντας αλγορίθμους και τεχνικές Τεχνητής Νοημοσύνης, για να παράγουν ένα τελικό σχέδιο λύσης των προβλημάτων. Ένα τέτοιο σύστημα είναι και ο **LPG-TD**, τον οποίο και χρησιμοποιήσαμε στην παρούσα εργασία. Η λειτουργία του περιγράφεται σε ξεχωριστό κεφάλαιο.<sup>[1], [2], [3], [5]</sup>

## 2.2.1 Γλώσσες αναπαράστασης προβλημάτων σχεδιασμού

Από τα παραπάνω, γίνεται κατανοητό πως στο σχεδιασμό Τεχνητής Νοημοσύνης καθοριστικό ρόλο θα έχει η αναπαράσταση των προβλημάτων σχεδιασμού, η οποία θα πρέπει να επιτρέπει στους σχεδιαστές να αξιοποιούν τη λογική δομή του προβλήματος. Οι γλώσσες που χρησιμοποιούνται στο σχεδιασμό πρέπει να είναι αρκετά εκφραστικές, ώστε να μπορούν να περιγράψουν πολλά είδη προβλημάτων, αλλά ταυτόχρονα αρκετά αυστηρές, με την έννοια της καθοριστικότητας, δίνοντας τη δυνατότητα σε αποτελεσματικούς αλγορίθμους να τις αξιοποιήσουν.

Η βασική γλώσσα αναπαράστασης των κλασικών σχεδιαστών είναι η γλώσσα **STRIPS** (Stanford Research Institute Problem Solver). Αναπτύχθηκε για τη λειτουργία του ομώνυμου σχεδιαστή το 1971 στο Stanford Research Institute και αποτελεί το πρότυπο στο οποίο βασίζονται οι περισσότερες γλώσσες αναπαράστασης προβλημάτων σχεδιασμού (STRIPS-like languages). Εδώ θα περιγράψουμε τα βασικά της χαρακτηριστικά, που θα μας βοηθήσουν να επεκταθούμε στην περιγραφή της γλώσσας **PDDL** (Planning Domain Definition Language), την οποία και χρησιμοποιεί η εφαρμογή μας.

- Αναπαράσταση καταστάσεων:

Μπορούμε να σκεφτούμε μια κατάσταση στην οποία βρισκόμαστε, σαν ένα στιγμιότυπο του κόσμου, στον οποίο λαμβάνει χώρα το πρόβλημα σχεδιασμού. Δηλαδή, σε κάθε κατάσταση που βρισκόμαστε, ισχύουν διαφορετικά «στοιχεία» από τα οποία αποτελείται ο κόσμος του προβλήματος που τέθηκε. Με την εκτέλεση κάποιας ενέργειας μεταβαίνουμε από μια κατάσταση σε κάποια άλλη (ή μπορούμε να παραμείνουμε στην ίδια). Μια κατάσταση αναπαρίσταται σε μια σύζευξη από γεγονότα, για παράδειγμα  $At(Plane1, Melbourne) \wedge At(Plane2, Sydney)$ . Επίσης, ισχύει η **Υπόθεση Κλειστού Κόσμου** (closed world-assumption), δηλαδή τα γεγονότα που υπάρχουν στον κόσμο αλλά στα οποία δεν γίνεται αναφορά στην αναπαράσταση κάποιας κατάστασης, θεωρούνται ψευδή για τη συγκεκριμένη κατάσταση.

- Αναπαράσταση στόχων:

Οι στόχοι θεωρούνται ως μερικές καταστάσεις και παριστάνονται με σύζευξη γεγονότων. Μια κατάσταση  $s$  ικανοποιεί κάποιο στόχο  $g$ , εάν η  $s$  περιέχει όλα τα γεγονότα (πιθανόν όχι μόνο), τα οποία ανήκουν στον  $g$ . Για παράδειγμα, η κατάσταση που παριστάνεται με την πρόταση  $Πλούσιος \wedge Διάσημος \wedge Επιτυχημένος$  ικανοποιεί τον στόχο  $Επιτυχημένος \wedge Διάσημος$ .

- Αναπαράσταση ενεργειών:

Μια ενέργεια χωρίζεται σε δύο μέρη: τις προϋποθέσεις (preconditions) και τα αποτελέσματα (effects) της εκτέλεσής της. Για παράδειγμα, για να εκφράσουμε την ενέργεια του ταξιδιού με αεροπλάνο από ένα μέρος σε ένα άλλο χρησιμοποιούμε την ακόλουθη ενέργεια:

Action:  $Fly(p, from, to)$ ,  
Precond:  $At(p, from) \wedge Plane(p) \wedge Airport(from) \wedge Airport(to)$   
Effect:  $\neg At(p, from) \wedge At(p, to)$ ,

Η παραπάνω αναπαράσταση της ενέργειας είναι για την ακρίβεια ένα **σχήμα ενεργειών** (action schema), με την έννοια ότι αναπαριστά πολλές ενέργειες που προκύπτουν από ανάθεση διαφόρων τιμών στις μεταβλητές  $p$ ,  $from$  και  $to$ .

Γενικά, ένα σχήμα ενεργειών αποτελείται από τρία μέρη:

1. το όνομα της ενέργειας και τη λίστα παραμέτρων, τα οποία χρησιμεύουν ως αναγνωριστικά για την ενέργεια.
2. τις προϋποθέσεις (preconditions), οι οποίες παριστάνονται με την μορφή σύζευξης γεγονότων που καθορίζουν τι πρέπει να ισχύει σε κάποια κατάσταση πριν την εκτέλεση κάποιας ενέργειας. Μεταβλητές οι οποίες εμφανίζονται στις προϋποθέσεις θα πρέπει να προστεθούν στην λίστα παραμέτρων της ενέργειας.
3. τα αποτελέσματα (effects) της εκτέλεσης της συγκεκριμένης ενέργειας, τα οποία παριστάνονται με την μορφή σύζευξης γεγονότων, περιγράφοντας έτσι το πώς περνάμε από μια κατάσταση σε κάποια άλλη μέσω της ενέργειας. Μεταβλητές οι οποίες εμφανίζονται στα αποτελέσματα θα πρέπει επίσης να προστεθούν στη λίστα παραμέτρων της ενέργειας. Ένα γεγονός  $P$  το οποίο εμφανίζεται στο αποτέλεσμα της ενέργειας θα ισχύει και στην κατάσταση στην οποία μεταβαίνουμε με την εκτέλεσή της. Ένα ψευδές γεγονός  $\neg P$  το οποίο εμφανίζεται στο αποτέλεσμα της ενέργειας θα είναι ψευδές στην κατάσταση στην οποία μεταβαίνουμε με την εκτέλεση της ενέργειας, κάτι που σημαίνει πως αν ίσχυε προηγουμένως η ισχύς του θα πρέπει να αναιρεθεί. Για να διευκολύνουν την αναγνωσιμότητα, κάποιοι σχεδιαστές χωρίζουν τα αποτελέσματα στην προστιθέμενη λίστα (add list), για αληθή γεγονότα, και στην αφαιρούμενη λίστα (delete list), για τα ψευδή.

Έχοντας ορίσει τη σύνταξη, μπορούμε τώρα να ορίσουμε τη σημασιολογία, περιγράφοντας τον τρόπο με τον οποίο οι ενέργειες επηρεάζουν τις καταστάσεις. Αρχικά, λέμε ότι μια ενέργεια είναι **εφαρμόσιμη** σε μια κατάσταση όταν ισχύει η προϋπόθεσή της (η σύζευξη των προϋποθέσεών της). Σε ένα σχήμα ενεργειών, όπως το παραπάνω, ο έλεγχος της εφαρμοσιμότητας μιας ενέργειας θα περιλαμβάνει μια μετάθεση  $\theta$  των μεταβλητών της προϋπόθεσής της. Για παράδειγμα, έστω ότι η τρέχουσα κατάσταση  $s$  περιγράφεται από την ακόλουθη πρόταση:

$At(P1, JFK) \wedge At(P2, SFO) \wedge Plane(P1) \wedge Plane(P2) \wedge Airport(JFK) \wedge Airport(SFO)$  :

Η κατάσταση ικανοποιεί την προϋπόθεση:

$At(p, from) \wedge Plane(p) \wedge Airport(from) \wedge Airport(to)$

με τη μετάθεση

$\{p=P1, from=JFK, to=SFO\}$ .

Έτσι, η ενέργεια  $Fly(P1, JFK, SFO)$  όπως ορίστηκε είναι εφαρμόσιμη.

Ξεκινώντας από την κατάσταση  $s$ , το αποτέλεσμα της εκτέλεσης μιας εφαρμόσιμης ενέργειας  $a$ , είναι μια κατάσταση  $s'$  που είναι ίδια με την  $s$  με τη διαφορά ότι κάθε αληθές γεγονός  $P$  στα

αποτελέσματα της  $a$  προστίθεται στην  $s'$  ενώ κάθε ψευδές γεγονός  $\neg P$  αφαιρείται από την  $s'$ . Δηλαδή, μετά την ενέργεια  $Fly(P1, JFK, SFO)$  η τρέχουσα κατάσταση γίνεται

$At(P1, SFO) \wedge At(P2, SFO) \wedge Plane(P1) \wedge Plane(P2) \wedge Airport(JFK) \wedge Airport(SFO)$ .

Αξίζει να σημειωθεί το ότι αν ένα γεγονός των αποτελεσμάτων ισχύει ήδη στην  $s$  δεν προστίθεται ξανά, ενώ αντίστοιχα αν ένα ψευδές γεγονός των αποτελεσμάτων δεν ισχύει ούτε και στην  $s$ , τότε απλά αγνοείται από τα αποτελέσματα. Ισχύει επίσης η **υπόθεση STRIPS** (STRIPS assumption): Κάθε γεγονός που δεν αναφέρεται στα αποτελέσματα παραμένει αναλλοίωτο.

Τελικά, μπορούμε να ορίσουμε το σχέδιο λύσης (solution) ενός προβλήματος σχεδιασμού. Στην απλούστερη μορφή του, είναι μια σειρά ενεργειών, που όταν εκτελούνται από την αρχική κατάσταση, οδηγούν σε μια κατάσταση που ικανοποιεί το στόχο (ή τη σύζευξη των επιμέρους στόχων).<sup>[3]</sup>

## 2.2.2 Η γλώσσα PDDL

Η γλώσσα STRIPS αποτελεί τη βάση των γλωσσών αναπαράστασης προβλημάτων σχεδιασμού, αλλά δεν μπορεί να εκφράσει επαρκώς κάποια πραγματικά προβλήματα σχεδιασμού. Για παράδειγμα, ο κυριότερος περιορισμός της, που αφορά την απουσία συναρτήσεων από τα σχήματα ενεργειών, αποκλείει κάθε πρόβλημα που περιέχει στόχο βελτιστοποίησης. Γι' αυτό, τα τελευταία χρόνια έχουν αναπτυχθεί πολλές παραλλαγές της, με σκοπό να αντιμετωπίσουν προβλήματα πολυπλοκότερων «κόσμων» και ενεργειών. Οι διάφοροι σχεδιαστικοί φορμαλισμοί που χρησιμοποιούνται στην Τεχνητή Νοημοσύνη, έχουν συστηματοποιηθεί σε μια πρότυπη γλώσσα που ονομάζεται **PDDL** (Planning Domain Definition Language).

Μια από τις βασικές αρχές της PDDL είναι ότι η περιγραφή του κόσμου και η περιγραφή του προβλήματος περιέχονται σε διαφορετικά μεταξύ τους αρχεία: το domain file και το problem file αντίστοιχα.

### 2.2.2.1 PDDL Domain File

Στον πίνακα 2.2-1 φαίνεται ένα απλό παράδειγμα αρχείου domain.pddl.

Αρχικά ορίζουμε το όνομα του κόσμου (εδώ είναι «domain corgialeneion»).

Στη συνέχεια δίνουμε τις **απαιτήσεις** (requirements). Επειδή η PDDL είναι μια πολύ γενική γλώσσα και οι περισσότεροι σχεδιαστές υποστηρίζουν συνήθως υποσύνολά της, στα αρχεία domain δηλώνονται οι απαιτήσεις, δηλαδή τα υποσύνολα των χαρακτηριστικών της γλώσσας που χρησιμοποιούνται στο αρχείο. Οι πιο συνηθισμένες είναι οι εξής:

- **strips**: Το απλούστερο υποσύνολο της PDDL, που περιλαμβάνει μόνο ό,τι και η STRIPS.

```

(define (domain corgialeneion)

(:requirements :strips :equality :durative-actions :typing :fluents)

(:types student level - object)

(:predicates
(at ?x - (either student object) ?l - level)
(done ?o - object)
(next ?l1 - level ?l2 - level)
)

(:functions
(levelvalue ?l - level)
(atrvalue ?o - object)
(totalvalue)
(time ?o - object)
(totaltime)
)

(:action obj
:parameters (?s - student ?l1 ?l2 - level ?o - object )
:precondition
(and
(at ?o ?l2)
      (next ?l1 ?l2)
(or ( (at ?o ?l2)(at ?o ?l2) ) )
)
:effect (and
(not (at ?s ?l1))
(at ?s ?l2)
(done ?o)
(increase (levelvalue ?l2) (atrvalue ?o))
(increase (totalvalue) (atrvalue ?o) )
(increase (totaltime) (time ?o) )
)
)
)
)

```

**PDDL Domain File 2.2-1**

- **equality:** Η απαίτηση αυτή δηλώνει ότι ο «κόσμος» χρησιμοποιεί το κατηγορημα «= $\Rightarrow$ », υποστηρίζεται δηλαδή η ισότητα.
- **typing:** Το domain χρησιμοποιεί «τύπους», με την έννοια που χρησιμοποιείται στη λογική πρώτης τάξης.
- **fluents:** Επιτρέπονται οι ορισμοί συναρτήσεων και η χρήση αποτελεσμάτων (effects) με τελεστές ανάθεσης και αριθμητικές συνθήκες. Οι συναρτήσεις μπορούν να επιστρέψουν μόνο αριθμό.

Ύστερα δηλώνονται οι **τύποι** (types) που θα χρησιμοποιηθούν. Στο παράδειγμα είναι οι τύποι *student* και *level*, ενώ ως προεπιλογή υπάρχει πάντα και ο γενικός τύπος **object**.

Στην επόμενη γραμμή ορίζονται τα **κατηγορήματα** (predicates), μέσω του ορισμού των τύπων των παραμέτρων τους. Για παράδειγμα, το κατηγορήμα (*next ?I1 – level ?I2 - level*) ορίζει μια σχέση μεταξύ των παραμέτρων *I1* και *I2*, που είναι και οι δύο τύπου *level*. Πριν από κάθε ξεχωριστή παράμετρο πρέπει να υπάρχει ερωτηματικό.

Με τον ίδιο τρόπο ορίζουμε τις **συναρτήσεις** (functions) που χρησιμοποιεί ο κόσμος του προβλήματος.

Στη συνέχεια δηλώνονται οι **ενέργειες** (actions), ή καλύτερα τα σχήματα ενεργειών. Εκτός από το όνομα κάθε ενέργειας, η δήλωσή της περιέχει 3 μέρη:

1. Λίστα παραμέτρων (**parameters**): οι παράμετροι που χρησιμοποιούνται στις προϋποθέσεις και τα αποτελέσματα καθώς και οι τύποι τους.
2. προϋπόθεση (**precondition**): τα γεγονότα που αποτελούν τις προϋποθέσεις που πρέπει να ισχύουν για να μπορεί να εκτελεστεί η ενέργεια, σε συζευκτική κανονική μορφή.
3. αποτέλεσμα (**effect**): τα γεγονότα που έχει ως αποτελέσματα η εκτέλεση της ενέργειας στην κατάσταση του κόσμου, και πάλι σε συζευκτική κανονική μορφή.

Στο παράδειγμα του αρχείου *domain* της εικόνας έχουμε ένα μοναδικό σχήμα ενεργειών, που στην πραγματικότητα μέσω των παραμέτρων του μεταφράζεται σε έναν μεγάλο αριθμό ενεργειών, ο οποίος εξαρτάται από το αρχείο του προβλήματος.

### 2.2.2.2 PDDL Problem File

Στο πίνακα 2.2-1 φαίνεται ένα απλό παράδειγμα αρχείου *problem.pddl*.

Σε ένα αρχείο που περιγράφει το πρόβλημα σχεδιασμού, εκτός από το όνομά του και το όνομα του αρχείου *domain* που θα χρησιμοποιήσει διακρίνονται 4 μέρη:

- **objects**: Εδώ ορίζονται τα αντικείμενα κάθε τύπου που εμφανίζονται στο πρόβλημα.
- **init**: Η περιγραφή της αρχικής κατάστασης είναι ουσιαστικά μια λίστα που περιλαμβάνει όλα τα γεγονότα που είναι αληθή κατά την εκκίνηση του προβλήματος. Όλα τα υπόλοιπα θεωρούνται αρχικώς ψευδή. Επίσης, στο μέρος αυτό αρχικοποιούνται οι τιμές των συναρτήσεων (fluents) που θα χρησιμοποιηθούν.
- **goal**: Ορίζονται οι στόχοι του προβλήματος σε συζευκτική κανονική μορφή.
- **metric**: Εδώ ουσιαστικά δηλώνουμε το δείκτη που θα καθορίσει την ποιότητα της παραγόμενης λύσης. Στην περίπτωση του προβλήματος της εικόνας ο σχεδιαστής καλείται

```

(define (problem corgProblem)
  (:domain corgialeneion)

  (:objects
   student1 – student
   level-1 – level
   level0 – level
   object298 – object
   level1 – level
   object423 – object
  )

  (:init
   (at student1 level0)
   (next level-1 level0)
   (next level0 level1)
   (= (levelvalue level0) 0)
   (at object298 level0)
   =(atrvalue object298) 21)
   =(time object298) 5)
   (next level1 level2)
   (= (levelvalue level1) 0)
   (at object423 level1)
   =(atrvalue object423) 23)
   =(time object423) 3)
  )

  (:goal
   ( and
   (at student1 level1)
   (> (levelvalue level0)0 )
   (> (levelvalue level1)12)
   (< (totaltime)12 )
   ))

  (:metric maximize (totalvalue)))

```

**PDDL Problem File 2.2-2**

να βρει από τις πιθανές λύσεις του προβλήματος εκείνη που μεγιστοποιεί κατά το δυνατόν την τιμή της συνάρτησης total-value.

Τα 2 ξεχωριστά αρχεία PDDL αποτελούν τις εισόδους του σχεδιαστή, ο οποίος θα προσπαθήσει να δημιουργήσει το σχέδιο λύσης του προβλήματος που μελετάμε, αν αυτό υπάρχει.<sup>[3], [10]</sup>

### 2.2.3 Ευρετικές Συναρτήσεις και Γράφοι Σχεδιασμού

Για να επιλύσουν τα προβλήματα σχεδιασμού, οι σχεδιαστές χρησιμοποιούν τεχνικές που βασίζονται σε αλγόριθμους αναζήτησης. Ξεκινώντας από την αρχική, κάθε κατάσταση μπορεί να οδηγηθεί σε πλήθος διαφορετικών κόμβων-καταστάσεων με την εφαρμογή των εφαρμόσιμων ενεργειών, και το ζητούμενο είναι να οδηγηθούμε σε μια κατάσταση όπου ικανοποιούνται οι στόχοι.

Η εξέταση όλων των δυνατών καταστάσεων στις οποίες ένας αλγόριθμος αναζήτησης μπορεί να οδηγηθεί σε κάθε βήμα, είναι μια διαδικασία πολύ χρονοβόρα αν γίνεται με τυχαίο τρόπο, γι' αυτό και οι σχεδιαστές χρησιμοποιούν τις **ευρετικές συναρτήσεις** (heuristic functions ή απλά heuristics), για να επιλέγουν μεταξύ των διαθέσιμων κόμβων εκείνους που εκτιμάται πως απέχουν λιγότερο από το στόχο.

Μια ευρετική συνάρτηση  $h$  πρέπει να είναι τέτοια ώστε η ποσότητα  $h(n)$  να αντιστοιχεί στο εκτιμώμενο κόστος της φθηνότερης διαδρομής από τον κόμβο  $n$  σε έναν οποιοδήποτε κόμβο στόχου, δηλαδή μια κατάσταση που ικανοποιεί το στόχο του προβλήματος. Η  $h$  μπορεί να είναι οποιαδήποτε συνάρτηση για την οποία να ισχύει  $h(n)=0$  σε περίπτωση που ο  $n$  είναι κόμβος στόχου.

Οι Ευρετικές Συναρτήσεις αξιολογούν με βάση τις υπάρχουσες πληροφορίες τις διαθέσιμες επιλογές που έχει ένας αλγόριθμος αναζήτησης σε κάθε βήμα, ώστε να αποφασίσουν ποιον κόμβο πρέπει να ακολουθήσει ο αλγόριθμος στο βήμα αυτό. Μια από τις δομές που χρησιμοποιούνται από αρκετούς σχεδιαστές για τη δημιουργία ευρετικών συναρτήσεων, είναι ο **Γράφος Σχεδιασμού** (Planning Graph).

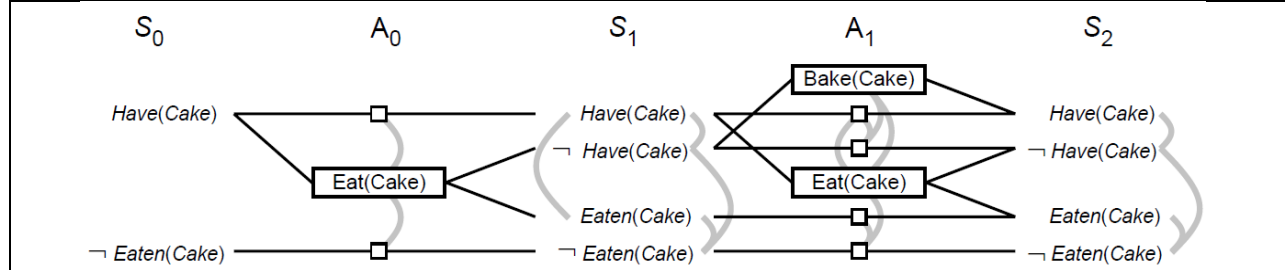
Ο γράφος σχεδιασμού αποτελείται από αριθμημένα επίπεδα που αντιστοιχούν σε χρονικές στιγμές της διαδικασίας εύρεσης του σχεδίου, με το επίπεδο 0 να αντιστοιχεί στην αρχική κατάσταση. Κάθε επίπεδο περιλαμβάνει είτε ένα σύνολο γεγονότων (επίπεδο καταστάσεων) είτε ένα σύνολο ενεργειών(επίπεδο ενεργειών). Γενικά, τα γεγονότα είναι όλα εκείνα που θα μπορούσαν να ισχύουν εκείνη τη χρονική στιγμή ανάλογα με τις ενέργειες που έχουν εκτελεστεί μέχρι τότε, ενώ οι ενέργειες είναι όλες εκείνες των οποίων οι προϋποθέσεις θα μπορούσαν να ικανοποιούνται εκείνη τη χρονική στιγμή, ανάλογα με το ποια γεγονότα όντως ισχύουν.

Στο σχήμα 2.2-3 φαίνεται το πρόβλημα “Have cake and eat cake too”, καθώς και ο γράφος σχεδιασμού του. Ο γράφος ξεκινά με το επίπεδο κατάστασης  $S0$ , που αντιστοιχεί στην αρχική κατάσταση του προβλήματος. Συνεχίζει με το επίπεδο ενεργειών  $A0$ , όπου τοποθετούμε όλες τις ενέργειες των οποίων οι προϋποθέσεις ικανοποιούνται στο προηγούμενο επίπεδο. Κάθε ενέργεια συνδέεται με τις προϋποθέσεις της στο  $S0$  και τα αποτελέσματά της στο  $S1$ , εισάγοντας έτσι στο  $S1$  νέα γεγονότα που απουσίαζαν από το  $S0$ . Για την αναπαράσταση της αδράνειας, που θα επιτρέψει σε κάποιο γεγονός να παραμείνει αληθές στην περίπτωση που καμιά ενέργεια δεν το επηρεάσει, χρησιμοποιούνται οι ενέργειες διατήρησης, οι οποίες προστίθενται με τη μορφή «κενών» ακμών (στο σχήμα έχουν τη μορφή  $\text{---}\square\text{---}$ ) για κάθε γεγονός, χρησιμοποιώντας το ως προϋπόθεση αλλά και αποτέλεσμα στα διαφορετικά επίπεδα.

Στο επίπεδο  $A0$  υπάρχουν όλες οι ενέργειες που μπορούν να εκτελεστούν στην κατάσταση  $S0$ , αλλά ταυτόχρονα σημειώνονται και οι συγκρούσεις μεταξύ τους που θα τις απέτρεπαν από το να συμβούν μαζί. Οι γκρι γραμμές στο σχήμα συμβολίζουν τις **σχέσεις αμοιβαίου αποκλεισμού**. Αντίστοιχα, στο



$Init(Have(Cake))$   
 $Goal(Have(Cake) \wedge Eaten(Cake))$   
 $Action(Eat(Cake))$   
 PRECOND:  $Have(Cake)$   
 EFFECT:  $\neg Have(Cake) \wedge Eaten(Cake)$   
 $Action(Bake(Cake))$   
 PRECOND:  $\neg Have(Cake)$   
 EFFECT:  $Have(Cake)$



Planning Graph 2.2-3

επίπεδο  $S_1$  τα γεγονότα  $Eat(Cake)$  και  $Have(cake)$  είναι αμοιβαίως αποκλειόμενα, αφού ανάλογα με την επιλογή των ενεργειών στο επίπεδο  $A_0$ , ένα από τα δύο μπορεί να ισχύει.

Για την κατασκευή του γράφου συνεχίζουμε με τον ίδιο τρόπο μέχρι να καταλήξουμε σε κάποιο σημείο όπου 2 συνεχόμενα επίπεδα του γράφου είναι πανομοιότυπα. Σε αυτό το σημείο λέμε ότι ο γράφος έχει σταθεροποιηθεί, αφού κάθε επόμενο επίπεδο θα είναι πανομοιότυπο οπότε δε χρειάζεται να προχωρήσουμε στο σχεδιασμό τους.

Καταλήγουμε, έτσι, σε μια δομή όπου κάθε επίπεδο  $A_i$  περιέχει όλες τις ενέργειες που είναι εφαρμόσιμες στο  $S_i$ , μαζί με τους περιορισμούς που καθορίζουν το ποιες ενέργειες δεν μπορούν να εκτελεστούν μαζί. Και κάθε επίπεδο  $S_i$  περιέχει όλα τα γεγονότα που μπορεί να προκύψουν από οποιαδήποτε επιλογή ενεργειών του  $A_{i-1}$ , μαζί με τους περιορισμούς που αφορούν το ποιο συνδυασμοί τους είναι αδύνατοι.

Ένας γράφος σχεδιασμού, όταν κατασκευαστεί, δίνει αρκετές πληροφορίες για το πρόβλημα που περιγράφει. Για παράδειγμα, ένα γεγονός που δεν εμφανίζεται στο τελικό στάδιο του γράφου, δεν μπορεί να επιτευχθεί με την εφαρμογή κανενός σχεδίου. Αυτή η παρατήρηση μπορεί να χρησιμοποιηθεί σε μια αναζήτηση ως εξής: Κάθε κατάσταση που περιέχει ένα μη επιτεύξιμο γεγονός, έχει κόστος  $h(n)=\infty$ .

Η ιδέα αυτή μπορεί να γενικευτεί για τη δημιουργία μιας ευρετικής συνάρτησης. Μπορούμε να εκτιμήσουμε το κόστος της επίτευξης κάθε γεγονότος που περιλαμβάνεται στο στόχο με την τιμή του

επιπέδου στο οποίο εμφανίζεται πρώτη φορά στο γράφο σχεδιασμού(level cost). Στο παράδειγμά μας αυτό μεταφράζεται σε κόστος 0 για το *Have(Cake)* και 1 για το *Eaten(Cake)*.<sup>[3], [6]</sup>

#### 2.2.4 Σχεδιασμός με γράφους

Ο σχεδιασμός με γράφους είναι μια τεχνική που χρησιμοποιεί τη δομή των γράφων σχεδιασμών για να εξαγάγει τη λύση ενός προβλήματος σχεδιασμού. Κατά την περιγραφή των κυριότερων στοιχείων του Σχεδιασμού με Γράφους, θα χρειαστεί να δώσουμε και κάποιους ορισμούς απαραίτητους για την περιγραφή των βασικών προδιαγραφών της λειτουργίας του σχεδιαστή LPG, που γίνεται στο αμέσως επόμενο κεφάλαιο.

Για την εξαγωγή της λύσης του προβλήματος, ο γράφος σχεδιασμού  $G$  ξεκινά να κατασκευάζεται με τον ίδιο τρόπο που περιγράψαμε, μέχρι να φτάσουμε σε κάποιο επίπεδο κατάστασης  $S_i$  όπου ισχύουν όλα τα γεγονότα που περιλαμβάνονται στο στόχο, χωρίς καμία σχέση αμοιβαίου αποκλεισμού μεταξύ τους.

Χωρίς βλάβη της γενικότητας, υποθέτουμε ότι οι κόμβοι στόχων του τελευταίου επιπέδου του γράφου που περιγράψαμε αντιστοιχούν στις προϋποθέσεις μιας ξεχωριστής ενέργειας  $a_{end}$ , που είναι η τελική ενέργεια κάθε έγκυρου σχεδίου που μπορεί να λύνει το πρόβλημα.

Ένας **γράφος ενεργειών**(action graph)  $A$  του  $G$ , είναι ένας υπογράφος του  $G$  που περιέχει την  $a_{end}$  και είναι τέτοιος ώστε, αν μια ενέργεια  $a$  του  $G$  περιέχεται στον  $A$ , τότε και τα γεγονότα του  $G$  που αντιστοιχούν στις προϋποθέσεις και τα αποτελέσματα της  $a$  περιέχονται επίσης στον  $A$ , μαζί με τις ακμές που τα ενώνουν με την  $a$ .

Ένα γεγονός  $q$  σε ένα επίπεδο  $i$  λέμε ότι «**υποστηρίζεται**» σε έναν υπογράφο  $G'$  του  $G$ , αν είτε στον  $G'$  υπάρχει στο επίπεδο  $i-1$  μια ενέργεια που έχει ως αποτέλεσμα το γεγονός  $q$ , είτε το επίπεδο  $i$  είναι το αρχικό( $i=0$ ).

Ένας **γράφος λύσης**(solution graph) για τον  $G$ , είναι ένας γράφος ενεργειών  $A_s$  του  $G$  τέτοιος ώστε όλες οι προϋποθέσεις κάθε ενέργειας που εμφανίζεται στον  $A_s$  υποστηρίζονται και ταυτόχρονα δεν υπάρχει καμία σχέση αμοιβαίου αποκλεισμού μεταξύ των ενεργειών.

Ο γράφος λύσης αποτελεί ένα έγκυρο σχέδιο για το πρόβλημα, δηλαδή μια λύση του προβλήματος. Αν δε βρεθεί ένας γράφος λύσης, ο γράφος σχεδιασμού  $G$  επεκτείνεται κατά δύο επίπεδα και η διαδικασία επαναλαμβάνεται. Συνθήκη τερματισμού είναι η κατασκευή του πλήρους γράφου σχεδιασμού, δηλαδή η εύρεση δύο πανομοιότυπων επιπέδων.

Δεδομένου ενός γράφου σχεδιασμού  $G$ , η διαδικασία αναζήτησης ενός γράφου λύσης ξεκινά από έναν αρχικό γράφο ενεργειών  $A$  του  $G$  (αρχικό μερικό σχέδιο), ο οποίος επιλέγεται συνήθως τυχαία, και τον μετασχηματίζει σε ένα γράφο λύσης μέσω της επαναληπτικής εφαρμογής τροποποιήσεων που κάθε φορά βελτιώνουν το μερικό σχέδιο. Οι δύο βασικές τροποποιήσεις είναι η πρόσθεση και η αφαίρεση μιας ενέργειας από τον τρέχοντα γράφο ενεργειών, μαζί με τις σχετικές ακμές σε κάθε περίπτωση.

Σε κάθε βήμα της αναζήτησης, το σύνολο των ενεργειών που μπορεί να προστεθούν ή να αφαιρεθούν καθορίζεται από τις «**ασυνέχειες**» (constraint violations) του τρέχοντος γράφου. Μια

ασυνέχεια είναι είτε μια σχέση αποκλεισμού μεταξύ ενεργειών του τρέχοντος γράφου, είτε ένα μη υποστηριζόμενο γεγονός που αποτελεί προϋπόθεση μιας ενέργειας στον τρέχοντα γράφο. Αυτό σημαίνει πως από τη στιγμή που υπολογίζεται ένας αρχικός γράφος ενεργειών, κάθε βήμα αναζήτησης διαλέγει τυχαία μια ασυνέχεια στον τρέχοντα γράφο. Αν αυτή αντιστοιχεί σε ένα μη υποστηριζόμενο γεγονός, τότε για να τη «λύσει» μπορεί είτε να προσθέσει μια ενέργεια που να το υποστηρίζει, είτε να αφαιρέσει μια ενέργεια της οποίας αποτελεί προϋπόθεση, μαζί με την ακμή που τη συνδέει με αυτήν. Αν αντιστοιχεί σε μια σχέση αμοιβαίου αποκλεισμού, τότε μπορεί να αφαιρέσει μια από τις αμοιβαίως αποκλειόμενες ενέργειες, και πάλι μαζί με τις ακμές με τις οποίες συνδέεται.

Δεδομένου ενός γράφου σχεδιασμού  $G$ , ενός γράφου ενεργειών  $A$  του  $G$  και μιας ασυνέχειας  $s$  στον  $A$ , η **γειτονιά αναζήτησης**  $N(s, A)$  της  $s$  στον  $A$ , είναι το σύνολο των γράφων ενεργειών που προκύπτουν από τον  $A$  με την εφαρμογή μιας τροποποίησης που λύνει την  $s$ . Σε κάθε βήμα της αναζήτησης, τα στοιχεία της γειτονιάς αξιολογούνται με βάση μια ευρετική συνάρτηση και το στοιχείο με την καλύτερη αξιολόγηση επιλέγεται ως ο πιθανός επόμενος υπογράφος. Η γενική ευρετική συνάρτηση  $F$  του  $A$  ορίζεται ως εξής:

$$F(A) = \sum_{a \in G} (\text{mutex}(a, A) + \text{precond}(a, A)),$$

όπου

$$\text{mutex}(a, A) = \begin{cases} 0 & \text{if } a \notin A \\ \text{me}(a, A) & \text{if } a \in A \end{cases}$$

και

$$\text{precond}(a, A) = \begin{cases} 0 & \text{if } a \notin A \\ \text{pre}(a, A) & \text{if } a \in A \end{cases}$$

και  $\text{me}(a, A)$  είναι ο αριθμός των ενεργειών στον  $A$  που είναι αμοιβαία αποκλειόμενες με την  $a$ , ενώ  $\text{pre}(a, A)$  είναι ο αριθμός των γεγονότων που είναι προϋποθέσεις της  $a$  και δεν υποστηρίζονται στον  $A$ . Είναι εύκολο να δούμε ότι όταν η αναζήτηση φτάνει σε έναν γράφο ενεργειών  $A$  με  $F(A) = 0$ , τότε ο  $A$  είναι ένας γράφος λύσης. <sup>[3], [6], [7], [9], [11]</sup>

## 2.2.5 Ο σχεδιαστής LPG

Ο **LPG** (Local search for Planning Graphs) είναι ένας σχεδιαστής βασισμένος σε αλγορίθμους τοπικής αναζήτησης και γράφους σχεδιασμού, ο οποίος χειρίζεται αρχεία PDDL που μπορεί να περιέχουν αριθμητικές ποσότητες και ενέργειες με διάρκειες. Σχεδιάστηκε από τους Alfonso Gerevini και Ivan Serina το 2002 και είναι προγραμματισμένος σε γλώσσα C. Μπορεί να λύσει προβλήματα που απαιτούν είτε παραγωγή είτε προσαρμογή σχεδίου ενός προβλήματος σχεδιασμού.

Το βασικό σύστημα αναζήτησης του LPG είναι εμπνευσμένο από το Walksat, μια αποδοτική διαδικασία που λύνει προβλήματα SAT. Το πεδίο αναζήτησης του αποτελείται από γράφους ενεργειών,

όπως τους περιγράψαμε, που αντιστοιχούν σε μερικά σχέδια λύσης. Κάθε βήμα αναζήτησης ισοδυναμεί με τη μετατροπή ενός γράφου ενεργειών σε έναν άλλον, μέσω συγκεκριμένων τροποποιήσεων των γράφων. Ο LPG εκμεταλλεύεται τη συμπαγή δομή του γράφου σχεδιασμού για να ορίσει τη γειτονιά αναζήτησης και να αξιολογήσει τα στοιχεία της χρησιμοποιώντας μια παραμετροποιημένη συνάρτηση αξιολόγησης που ορίζει διαφορετικές ευρετικές συναρτήσεις της γενικής μορφής που περιγράψαμε παραπάνω. Οι παράμετροι χρησιμοποιούνται κάθε φορά για να αποτιμούν τους διάφορους τύπους ασυνεχειών του τρέχοντος μερικού σχεδίου και αξιολογούνται δυναμικά κατά τη διάρκεια της αναζήτησης με τη χρήση διακριτών πολλαπλασιαστών Lagrange.

Η συνάρτηση αξιολόγησης χρησιμοποιεί ευρετικές συναρτήσεις για να εκτιμήσει τελικά το κόστος αναζήτησης και το κόστος εκτέλεσης της ικανοποίησης μιας -πιθανόν αριθμητικής- προϋπόθεσης. Οι διάρκειες των ενεργειών και οι αριθμητικές ποσότητες παριστάνονται στους γράφους ενεργειών και μοντελοποιούνται στη συνάρτηση αξιολόγησης. Σε χρονικά domains, δηλαδή domains με ενέργειες με διάρκεια, οι ενέργειες ταξινομούνται με τη χρήση ενός γράφου προτεραιοτήτων, που διατηρείται κατά τη διάρκεια της αναζήτησης και λαμβάνει υπόψη τις σχέσεις αμοιβαίου αποκλεισμού του γράφου σχεδιασμού.

Η έκδοση LPG-td του σχεδιαστή, που είναι και αυτή που χρησιμοποιείται στην παρούσα εργασία, είναι μια επέκταση του LPG που επιτρέπει το χειρισμό των νέων στοιχείων που περιλαμβάνει η 2.2 έκδοση της PDDL.<sup>[6], [7], [8], [9]</sup>

## 2.3 ΠΡΟΔΙΑΓΡΑΦΕΣ ΤΟΥ ΣΧΕΔΙΑΣΜΟΥ ΗΛΕΚΤΡΟΝΙΚΗΣ ΜΑΘΗΣΗΣ

Τα εκπαιδευτικά συστήματα πάντοτε στόχευαν σε κάποιο βαθμό στην εξατομίκευση του μαθήματος λαμβάνοντας υπόψη τις ανάγκες κάθε μαθητή. Με τον παραδοσιακό τρόπο διδασκαλίας, που ακολουθείται κατά την τυπική εκπαιδευτική διαδικασία στην αίθουσα διδασκαλίας, δεν είναι πάντα εύκολο να ληφθούν υπόψη οι διαφορετικές εκπαιδευτικές ανάγκες και προτιμήσεις του κάθε μαθητή.

Η Ηλεκτρονική Μάθηση εισάγει πολλές και σημαντικές καινοτομίες στην εκπαίδευση, προσφέροντας έναν πολλά υποσχόμενο τρόπο για να διευκολύνει και να ενισχύσει την εκπαιδευτική διαδικασία, δημιουργώντας ευέλικτα μαθήματα με το συνδυασμό **αντικειμένων μάθησης** (Learning Objects). Ως αντικείμενα μάθησης ορίζονται οι ψηφιακοί πόροι που μπορούν να χρησιμοποιηθούν και να επαναχρησιμοποιηθούν για την υποστήριξη της γνώσης. Οι τεχνικές σχεδιασμού Τεχνητής Νοημοσύνης έχουν αποδειχθεί πολύ επαρκείς για τη δημιουργία εκπαιδευτικών διαδρομών, δηλαδή διαδρομών που αποτελούνται από αντικείμενα μάθησης, οι οποίες είναι πλήρως εναρμονισμένες με τις αρχές της εκπαίδευσης.

Πιο συγκεκριμένα, ο στόχος του σχεδιασμού σε ένα περιβάλλον ηλεκτρονικής μάθησης είναι να παρέχει σχέδια λύσεων εξατομικευμένα ως προς το μαθητή, προσφέροντας μια ευέλικτη εκπαιδευτική διαδικασία όπου το μάθημα και τα αντικείμενα μάθησης είναι εναρμονισμένα με τις ανάγκες, το υπόβαθρο, τις μαθησιακές προτιμήσεις και γενικότερα το προφίλ του κάθε μαθητή.

Ο **Σχεδιασμός Ηλεκτρονικής Μάθησης** (e-learning Planning) ακολουθεί τις αρχές του σχεδιασμού Τεχνητής Νοημοσύνης, εξειδικεύοντάς τις στις ανάγκες της εκπαίδευσης. Τα προβλήματα σχεδιασμού Ηλεκτρονικής Μάθησης μπορούν, επομένως, να περιγραφούν με τον ίδιο τρόπο με τα προβλήματα σχεδιασμού Τεχνητής Νοημοσύνης, με τα δομικά τους στοιχεία να αντιστοιχίζονται με εκπαιδευτικούς όρους.

Σε ένα πρόβλημα σχεδιασμού Ηλεκτρονικής Μάθησης, ο στόχος είναι αντίστοιχος με το στόχο ενός μαθήματος, δηλαδή να αποκτήσει ο μαθητής γνώση πάνω σε ένα γνωστικό αντικείμενο. Οι ενέργειες αντιστοιχούν στις εργασίες που μπορεί να ακολουθήσει για να αποκτήσει αυτή τη γνώση, όπως για παράδειγμα την παρακολούθηση μιας διάλεξης ή την ανάγνωση ενός συγγράματος, ενώ τα γεγονότα αντιστοιχούν στα Αντικείμενα Μάθησης και ουσιαστικά αντιπροσωπεύουν την απόκτηση μερικών γνώσεων. Κατ' αντιστοιχία με τα κλασικά προβλήματα σχεδιασμού, τα Αντικείμενα Μάθησης μπορούν να αντιπροσωπεύουν τις προϋποθέσεις και τα αποτελέσματα των εργασιών.

Ο καθορισμός των στοιχείων αυτών του προβλήματος γίνεται από τον εκπαιδευτικό. Όπως σε κάθε εκπαιδευτική διαδικασία, ο εκπαιδευτικός πρέπει να καθορίσει τη γνώση που πρέπει να μεταδοθεί και τους διαθέσιμους τρόπους με τους οποίους μπορεί αυτή να μεταδοθεί. Η διαφορά είναι ότι τώρα μπορεί ευκολότερα να καθορίσει διαφορετικούς τρόπους απόκτησης της απαιτούμενης γνώσης από κάθε μαθητή, ανάλογα με το εκπαιδευτικό του προφίλ.

Για παράδειγμα, έστω ότι δύο μαθητές, ο Γιάννης και η Λένα, θέλουν να αποκτήσουν βασικές δεξιότητες προγραμματισμού σε Java. Ο Γιάννης είναι μαθητής της θεωρητικής κατεύθυνσης και δεν έχει γνώσεις προγραμματισμού, ενώ η Λένα ασχολείται γενικότερα με τον προγραμματισμό και έχει βασικές γνώσεις της Pascal. Θα μπορούσαμε κατ' αρχάς να σκεφτούμε ότι ο Γιάννης ξεκινά από «χαμηλότερο»

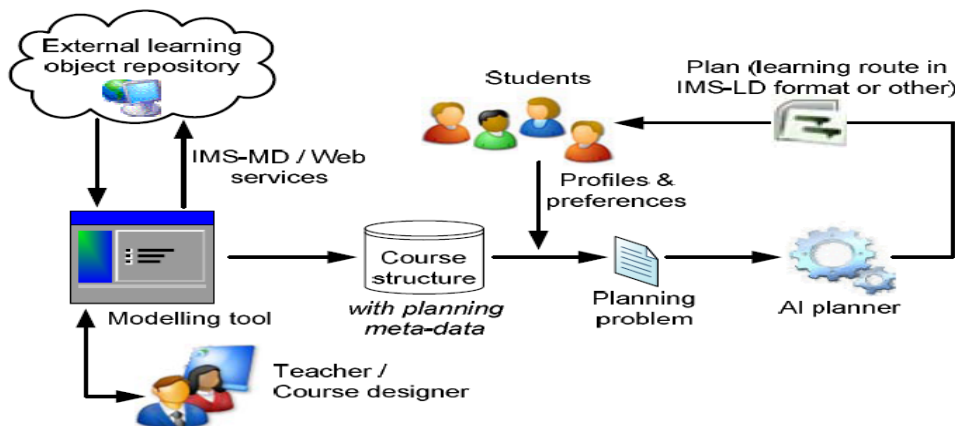
επίπεδο γνώσεων του αντικειμένου και θα χρειαστεί περισσότερες εργασίες για την απόκτηση της απαιτούμενης γνώσης. Επίσης, η Λένα θα μπορούσε να χρησιμοποιήσει ένα Αντικείμενο Μάθησης βασισμένο στις ομοιότητες και τις διαφορές της Java και της Pascal, ενώ ο Γιάννης όχι.

Γενικότερα, ο καθορισμός των στοιχείων ενός προβλήματος σχεδιασμού Ηλεκτρονικής Μάθησης επιτυγχάνεται σε 3 στάδια:

- **Επίπεδο αντίληψης** (conceptual view): Σε αυτό το στάδιο ο εκπαιδευτικός καθορίζει τα περιεχόμενα του μαθήματος σε θεωρητικό επίπεδο, δηλαδή τι γνώση πρόκειται να μεταδοθεί.
- **Επίπεδο εργασιών** (task view): Εδώ γίνεται μια αποδόμηση του επιπέδου αντίληψης σε συγκεκριμένες εργασίες. Καθορίζεται δηλαδή το πώς μπορεί να μεταδοθεί αυτή η γνώση.
- **Επίπεδο προσαρμογής** (adaptation view): Στο τελευταίο αυτό στάδιο, ο εκπαιδευτικός χρησιμοποιεί πληροφορίες που αφορούν τους μαθητές, το υπόβαθρο και το προφίλ τους, για να καθορίσει το *σε ποιόν* θα μεταδοθεί η γνώση και με *ποιο τρόπο*.

Αφού καθοριστούν τα στοιχεία αυτά, η μοντελοποίησή τους για τις ανάγκες του σχεδιασμού αποτελεί ευθύνη του Συστήματος Ηλεκτρονικής Μάθησης. Αυτό δίνει στον εκπαιδευτικό τη δυνατότητα να αναπαραστήσει και να χρησιμοποιήσει τα στοιχεία που καθορίστηκαν παραπάνω, ώστε να οριστεί πλήρως το πρόβλημα σχεδιασμού και ο «κόσμος» του προβλήματος. Η διαδικασία αυτή διαφέρει ανάλογα με το είδος του μαθήματος και την πολυπλοκότητα των εργασιών που του αντιστοιχούν. Συνήθως γίνεται μέσω κάποιων μενού επιλογών, ενώ σε κάποια εξελιγμένα Συστήματα Ηλεκτρονικής Μάθησης χρησιμοποιούνται και γραφικές αναπαραστάσεις.

Μετά τη μοντελοποίηση του μαθήματος και του προφίλ των μαθητών, το Σύστημα Ηλεκτρονικής Μάθησης αναλαμβάνει να συντάξει τις πληροφορίες αυτές σαν ένα πρόβλημα σχεδιασμού για να επιλυθεί από κάποιον σχεδιαστή. Ο σχεδιαστής παράγει το σχέδιο επίλυσης, που αντιστοιχεί σε μια εκπαιδευτική διαδρομή, η οποία μεταφορτώνεται στο σύστημα δίνοντας στο μαθητή τη δυνατότητα να πλοηγηθεί ώστε να εκτελέσει τις ενέργειες που την απαρτίζουν και να φτάσει στον τελικό στόχο, που αντιστοιχεί στην απόκτηση της ζητούμενης γνώσης.<sup>[4], [5], [12], [13]</sup>



Σχεδιασμός Ηλεκτρονικής Μάθησης 2.3-1

### 3 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Στο κεφάλαιο αυτό περιγράφεται ο σχεδιασμός και η μορφή της Βάσης Δεδομένων που αναπτύχθηκε στο πλαίσιο της παρούσας εργασίας. Αρχικά θα αναφερθούμε στη βάση δεδομένων του μουσείου, από την οποία αντλήσαμε αρκετά στοιχεία, και στη συνέχεια θα αναλύσουμε την ανάπτυξη της βάσης που χρησιμοποιείται για τη λειτουργία του συστήματός μας.

Κατα τη διάρκεια της ανάλυσης αυτής θα μελετήσουμε τα στοιχεία που εισαγάγαμε για την εφαρμογή, καθώς και τους σκοπούς που αυτά εξυπηρετούν. Η χρήση των δεδομένων κατά τη λειτουργία της εφαρμογής εξηγείται πλήρως στο κεφάλαιο που αναφέρεται στην περιγραφή του συστήματος, αλλά και σε αυτό που περιγράφει τις λειτουργίες των αρχείων PHP.

Αξίζει να σημειωθεί ότι η εφαρμογή χρησιμοποιεί μια ακόμα βάση δεδομένων, τη βάση δεδομένων του Joomla, η οποία δημιουργείται αυτόματα με τη δημιουργία της ιστοσελίδας και περιέχει τα δομικά συστατικά του συνολικού συστήματος. Γι' αυτό και αναλύεται ξεχωριστά στο κεφάλαιο που αναφέρεται στο σχεδιασμό και την ανάπτυξη του συστήματος.

Για την οργάνωση και τη διαχείριση των βάσεων χρησιμοποιήθηκε το σύστημα διαχείρισης βάσεων δεδομένων MySQL, μέσω της διεπαφής του WebMatrix. Η επικοινωνία τους με την εφαρμογή γίνεται με εντολές γλώσσας SQL που εκτελούνται μέσα στον κώδικα PHP των αρχείων.

#### 3.1 ΑΡΧΙΚΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΤΟΥ ΚΟΡΓΙΑΛΕΝΕΙΟΥ ΜΟΥΣΕΙΟΥ

Η βάση δεδομένων **korgialenios** είναι η βάση του Κοργιαλένιου Ιδρύματος, όπου είναι οργανωμένο το ψηφιοποιημένο υλικό του μουσείου. Αναπτύχθηκε με βάση το πρότυπο **CIDOC Conceptual Reference Model (CRM)**, το οποίο παρέχει μια τυπική δομή για την περιγραφή των εννοιών και των σχέσεων που χρησιμοποιούνται για την τεκμηρίωση στοιχείων πολιτιστικής κληρονομιάς. Το διάγραμμα οντοτήτων-συσχετίσεων της βάσης αυτής ακολουθεί τα πρότυπα οντολογιών ISO. Σχεδιάστηκε από τους Στέλλα Μαρκαντωνάτου και Σωκράτη Σοφιανόπουλο στο πλαίσιο του έργου **Ψηφιοποίηση, ψηφιακή επεξεργασία, μετάφραση περιεχομένου τεκμηρίωσης και ανάπτυξη διαδραστικής εφαρμογής (Νοέμβριος-Δεκέμβριος 2008), Υπόεργο 2 της Πράξης «Εμπλουτισμός και ανάδειξη ψηφιακού πολιτιστικού αποθέματος Κοργιαλενείου Ιδρύματος Κεφαλληνίας» του Επιχειρησιακού προγράμματος «Κοινωνία της Πληροφορίας».**

Στη βάση αυτή αποκτήσαμε πρόσβαση μετά από σχετική άδεια των υπευθύνων του ιδρύματος. Δε χρησιμοποιείται άμεσα από την εφαρμογή μας, αλλά αρκετά στοιχεία της χρησιμοποιήθηκαν για την ανάπτυξη της βάσης δεδομένων **corgialen**, που εξετάζουμε στο επόμενο υποκεφάλαιο και περιέχει μεταξύ άλλων τα αντικείμενα μάθησης που χρησιμοποιήσαμε. Για το λόγο αυτό θα κάνουμε μια γενική περιγραφή και θα αναφερθούμε επιγραμματικά στους πίνακες των οποίων τα δεδομένα χρησιμοποιήσαμε.

Στο σχήμα 3.1-1 της επόμενης σελίδας φαίνεται το διάγραμμα οντοτήτων συσχετίσεων της βάσης **korgialenios**. Αποτελείται από 58 πίνακες που περιέχουν διαφόρων ειδών δεδομένα με πληροφορίες για





τα εκθέματα του μουσείου. Τα εκθέματα μπορεί να είναι είτε αντικείμενα, όπως εργαλεία, φορεσιές και έργα τέχνης, είτε, κυρίως, φωτογραφίες που απεικονίζουν τοποθεσίες, πρόσωπα και ιστορικά γεγονότα.

Οι πίνακες της βάσης μπορούν να διαχωριστούν σε 4 κατηγορίες:

- Ιστορικές πληροφορίες: Εδώ ανήκουν οι πίνακες που περιέχουν πληροφορίες ιστορικού περιεχομένου για τα εκθέματα. Αφορούν την κατασκευή των αντικειμένων, την εποχή και το σκοπό για τον οποίο χρησιμοποιήθηκαν, ενώ για τις φωτογραφίες περιέχουν πληροφορίες για τα γεγονότα, τις τοποθεσίες ή τα πρόσωπα που απεικονίζουν. Κάποια παραδείγματα είναι οι πίνακες *Kataskeuh*, *Kataskeuasths*, *Kataskeuh\_Topos*, *Antikeimeno\_Xrhsh*, *Antikeimeno\_Sxetiko\_Gegonos*, *Bibliografia* και *Apeikonish*.
- Πληροφορίες σχετικές με το μουσείο: Οι σχετικοί πίνακες αφορούν στοιχεία που καταγράφουν τη θέση των εκθεμάτων στο μουσείο, τη συλλογή στην οποία ενδεχομένως ανήκουν, τον τρόπο με τον οποίο αποκτήθηκαν, τις εκθέσεις στις οποίες συμμετείχαν κλπ. Μερικοί τέτοιοι πίνακες είναι οι *Antikeimeno\_Theseis*, *Antikeimeno\_Daneismos*, *Antikeimeno\_Apokthsi* και *Ekthesi*.
- Πληροφορίες εμφάνισης: Πίνακες με δεδομένα που αφορούν την εξωτερική μορφή του εκθέματος, όπως οι πίνακες *Antikeimeno*, *Shmansh*, *Antikeimeno\_Proarthma*.
- Πληροφορίες αρχείου: Πίνακες που περιέχουν δεδομένα καταχώρησης του εκθέματος σε αρχεία του μουσείου, όπως οι πίνακες *Anaparastasi*, *Arxeio*, *Bibliografia* και *Anaparastasi\_Foreas*.

Η επιλογή των δεδομένων που χρησιμοποιήθηκαν έγινε με βάση τη φύση και τον εκπαιδευτικό σκοπό της εφαρμογής, αλλά και την πληρότητα των αντίστοιχων πινάκων. Στο σχήμα 3.1-2 φαίνονται οι πίνακες από τους οποίους αντλήσαμε δεδομένα για τη δημιουργία της βάσης που χρησιμοποιεί η εφαρμογή.

Ο βασικός πίνακας είναι ο πίνακας *Antikeimeno*, που περιλαμβάνει όλα τα εκθέματα του μουσείου και είναι ο μόνος 100% πλήρης, με 635 εγγραφές. Περιέχει πεδία με την ονομασία κάθε εκθέματος, μια χαρακτηριστική μικρή περιγραφή του και πληροφορίες που αφορούν στη θέση του στο μουσείο, τη συλλογή στην οποία ανήκει, καθώς και στα στοιχεία της εξωτερικής του μορφής, όπως τα χρώματα και οι διαστάσεις του.

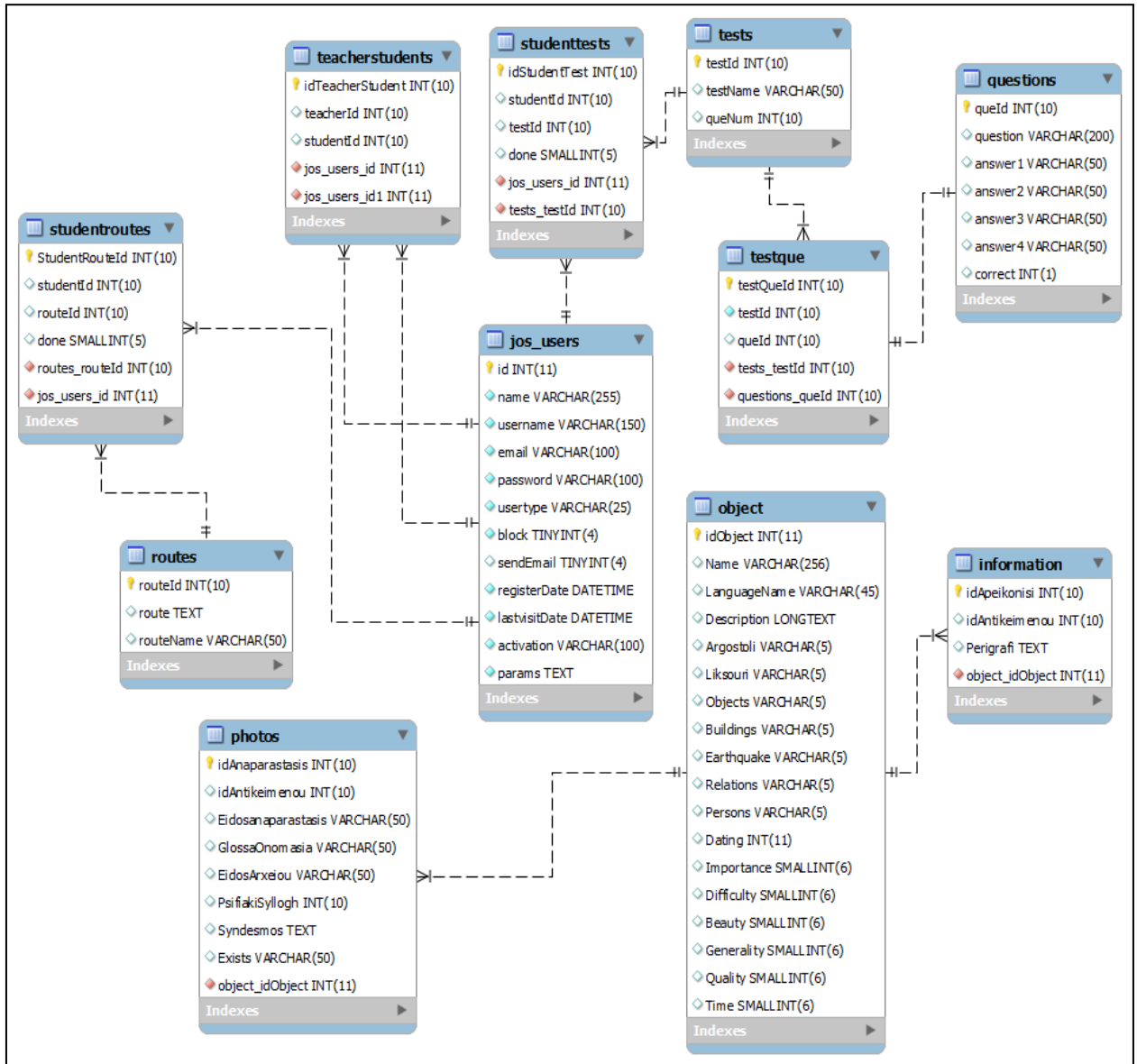
Ο πίνακας *Anaparastasi* είναι ιδιαίτερα χρήσιμος γιατί περιέχει τα ονόματα των αρχείων με τις φωτογραφίες των εκθεμάτων, με τη μορφή της διεύθυνσης των αρχείων στο server του μουσείου. Το πεδίο *Syndesmos* περιέχει τη συγκεκριμένη πληροφορία, ενώ στα υπόλοιπα πεδία υπάρχουν αντίστοιχες πληροφορίες όπως το είδος του ψηφιακού αρχείου (*tiff*, *jpeg* κλπ) και η ημερομηνία της καταχώρησης. Στον πίνακα αυτόν υπάρχουν περίπου 300 εγγραφές, αλλά από αρχεία αυτά είχαμε στη διάθεσή μας λιγότερα από 200.

Από τον πίνακα *Kataskeuh* χρησιμοποιήσαμε κυρίως το πεδίο *Xronologisi*, που για τα αντικείμενα περιέχει τη χρονολογία που χρησιμοποιούνταν, ενώ για τις φωτογραφίες την ημερομηνία φωτογράφισης.

Τέλος, ο πίνακας *Apeikonish* στο πεδίο *Perigrafh* περιέχει μια πλήρη περιγραφή του αντικειμένου με το αντίστοιχο ταυτοτικό πεδίο *id*, σε κάποιες περιπτώσεις μαζί με εκτενέστερες πληροφορίες όταν



### 3.2 Η ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ CORGIALEN



Corgialen E-R Diagram 3.2-1

Η βάση δεδομένων Corgialen αναπτύχθηκε στο πλαίσιο της εργασίας για τους σκοπούς της εφαρμογής. Αποτελείται από 10 πίνακες, επικοινωνεί μέσω του κώδικα php με τον πίνακα jos\_users, ο οποίος περιέχει τα δεδομένα χρηστών και ανήκει στη βάση δεδομένων του Joomla, η οποία αναλύεται στο κεφάλαιο που αφορά τη σχεδίαση και ανάπτυξη του συστήματος.

Οι πίνακες Object, information και photos περιέχουν τα αντικείμενα μάθησης της εφαρμογής. Περιέχουν δεδομένα που προέρχονται από τη βάση Korgialenios, αλλά και τα στοιχεία που έχουμε προσθέσει για την τήρηση των προδιαγραφών του σχεδιασμού ηλεκτρονικής μάθησης. Οι υπόλοιποι δημιουργήθηκαν για να αποθηκεύονται και να χρησιμοποιούνται απαραίτητα δεδομένα που παράγονται κυρίως κατά τη διάρκεια λειτουργίας της εφαρμογής.

### **3.2.1 Αντικείμενα Μάθησης (Learning Objects)-Μαθησιακά Χαρακτηριστικά (Learning Attributes)**

Στους πίνακες αυτούς περιέχονται τα αντικείμενα μάθησης, δηλαδή οι πληροφορίες και οι εικόνες των εκθεμάτων που αποτελούν τις εκπαιδευτικές διαδρομές. Πέρα από τα δεδομένα που πήραμε από τη βάση Korgialenios, στον πίνακα object της βάσης Corgialen έχουμε εισάγει για τις εκπαιδευτικές ανάγκες της εφαρμογής πεδία που αντιστοιχούν στα μαθησιακά χαρακτηριστικά των αντικειμένων μάθησης.

Τα πεδία αυτά αναλύονται παρακάτω και είναι ιδιαίτερα σημαντικά για την εξατομίκευση των εκπαιδευτικών διαδρομών που δημιουργούνται από τους εκπαιδευτικούς μέσα από την εφαρμογή, ανάλογα με τους εκπαιδευόμενους στους οποίους έχουν σκοπό να τις αναθέσουν. Χαρακτηρίζουν τη δυσκολία, τη σημασία, τη γενικότητα της γνώσης, την ποιότητα και την ομορφιά της εικόνας και το χρόνο που απαιτείται για την κατανόηση των αντικειμένων, καθαρά ως αντικειμένων μάθησης, αξιολογώντας τις ιδιότητες αυτές σε κλίμακα από το 0 ως το 10, εκτός του χρόνου, που μπορεί όπως θα δούμε να διαφέρει. Ανάλογα με τις τιμές αυτές και τα όρια που θέτει ο εκπαιδευτικός, καθορίζονται αργότερα τα αντικείμενα μάθησης που θα αποτελέσουν τη ζητούμενη κάθε φορά εκπαιδευτική διαδρομή, καθώς και σε αρκετές περιπτώσεις τη σειρά με την οποία αυτά θα εμφανιστούν στη διαδρομή.

#### **1) object**

Ο πίνακας αυτός δημιουργήθηκε ως μια επέκταση του πίνακα Antikeimeno της βάσης Korgialenios. Είναι η βάση της εκπαιδευτικής λειτουργίας της εφαρμογής, αφού οι εγγραφές του αποτελούν τα ξεχωριστά αντικείμενα μάθησης (Learning Objects) τα οποία συνθέτουν τις εκπαιδευτικές διαδρομές που παράγει η εφαρμογή. Γι' αυτό και θα εξετάσουμε ξεχωριστά τα διάφορα πεδία του:

- idObject:

Πρόκειται για το Id του εκθέματος, όπως ακριβώς και στον πίνακα Antikeimeno.

- Name:

Το όνομα που δείχνει τη φύση του εκθέματος. Στις περισσότερες εγγραφές είναι Photo.

- LanguageName:

Η γλώσσα της περιγραφής, που ακολουθεί στο επόμενο πεδίο.

- Description:

Μια επιγραμματική περιγραφή του εκθέματος. Για τις ανάγκες της ανάλυσης και επεξεργασίας των δεδομένων της βάσης στο UPV, μεταφράσαμε το αντίστοιχο πεδίο των εγγραφών της Korgialenios στα Αγγλικά, οπότε το πεδίο αυτό περιέχει την περιγραφή του κάθε αντικειμένου στην Αγγλική γλώσσα.

- Argostoli, Liksouri, Objects, Buildings, Earthquake, Relations, Persons:

Για τις ανάγκες της εφαρμογής χωρίσαμε τα αντικείμενα σε 7 κατηγορίες, ανάλογα με το θέμα τους. Σκοπός ήταν να μπορεί να επιλέξει ο δάσκαλος που θα δημιουργήσει τις εκπαιδευτικές διαδρομές τις κατηγορίες αντικειμένων που θέλει κάθε φορά. Μετά από μελέτη των αντικειμένων και συνεκτιμώντας το γεγονός ότι θα έπρεπε ως ένα σημείο να ισομοιαστούν, καταλήξαμε στις 7 αυτές κατηγορίες. Οι εγγραφές έχουν στα πεδία αυτά το χαρακτήρα 'y' ή 'n', ανάλογα με το αν ανήκουν ή όχι τα αντικείμενα στην αντίστοιχη κατηγορία.

Στην κατηγορία **Argostoli** ανήκουν τα αντικείμενα, κυρίως φωτογραφίες, που απεικονίζουν γεγονότα που έλαβαν χώρα στο Αργοστόλι, ή τοποθεσίες και κτήρια της πόλης. Το αντίστοιχο ισχύει και για την κατηγορία **Liksouri**, για την πόλη του Ληξουρίου.

Στην κατηγορία **Objects** ανήκουν τα αντικείμενα της καθημερινής ζωής που βρίσκονται ως εκθέματα στο μουσείο.

Στην κατηγορία **Buildings** ανήκουν οι φωτογραφίες που απεικονίζουν κτήρια, ενώ στην κατηγορία **Persons** αυτές που απεικονίζουν προσωπικότητες που διαδραμάτισαν σημαντικό ρόλο στην ιστορία του νησιού.

Στην κατηγορία **Earthquake** ανήκουν οι φωτογραφίες με θέματα που σχετίζονται με το μεγάλο σεισμό που έπληξε το νησί το 1953.

Τέλος, στην κατηγορία **Relations**, ανήκουν οι φωτογραφίες που μέσω των προσώπων ή των γεγονότων που απεικονίζουν, δίνουν πληροφορίες για τις σχέσεις των ανθρώπων του νησιού με χώρες του εξωτερικού.

Αξίζει να σημειωθεί ότι υπάρχει μεγάλος βαθμός επικάλυψης μεταξύ των κατηγοριών, αφού είναι πολλές οι περιπτώσεις που τα αντικείμενα ανήκουν σε παραπάνω από μια κατηγορίες. Όταν από το δάσκαλο επιλεγθούν 2 ή περισσότερες κατηγορίες, η εφαρμογή το αντιμετωπίζει ως διάζευξη, ανακτώντας από τη βάση όσα αντικείμενα ανήκουν τουλάχιστον σε μια από αυτές.

- Dating:

Στο πεδίο αυτό καταχωρείται το ημερολογιακό έτος που εκτιμάται ότι κατασκευάστηκε το αντικείμενο ή τραβήχτηκε η φωτογραφία, ανάλογα με το είδος του εκθέματος, όπως και στον πίνακα *Kataskeuh* της βάσης δεδομένων Korgialenios που είδαμε πριν. Στις περιπτώσεις που δεν υπήρχε αυτή η πληροφορία στη βάση του μουσείου, την εξαγάγαμε από την περιγραφή των αντικειμένων ανατρέχοντας

και σε άλλες πηγές. Η σημασία του είναι μεγάλη για τη δημιουργία των εκπαιδευτικών διαδρομών, αφού καθορίζει σε μεγάλο βαθμό τη σειρά των αντικειμένων μάθησης μέσα στις παραγόμενες διαδρομές.

- Importance, Difficulty, Beauty, Generality, Quality, Time:

Τα πεδία αυτά αντιστοιχούν στα μαθησιακά χαρακτηριστικά που αναφέραμε παραπάνω.

Το πεδίο **Importance** είναι δείκτης της σχετικής σημασίας που έχει για την ιστορία το αντικείμενο μάθησης. Όσο μεγαλύτερη είναι η τιμή του πεδίου Importance, τόσο πιθανότερο είναι το αντικείμενο να περιέχεται σε μια εκπαιδευτική διαδρομή που αφορά κάποια από τις κατηγορίες στις οποίες ανήκει.

Το **Difficulty** είναι δείκτης της δυσκολίας που εκτιμάται ότι θα συναντήσει ένας μαθητής κατά την προσπάθειά του να αφομοιώσει τη γνώση που θα του δώσει το αντίστοιχο αντικείμενο μάθησης. Σκοπός της εισαγωγής του ήταν να δοθεί στον εκπαιδευτικό ένα κριτήριο για να αποκλείει κάποια από τα αντικείμενα όταν οι δημιουργούμενες διαδρομές απευθύνονται σε μαθητές μικρότερης τάξης, αλλά και να ρυθμίζει στην πορεία ενός μαθήματος τα περιεχόμενα μιας σειράς από εκπαιδευτικές διαδρομές, αυξάνοντας κλιμακωτά τα όρια δυσκολίας.

Το πεδίο **Beauty** αφορά την ομορφιά της φωτογραφίας ή του απεικονιζόμενου αντικειμένου και εισήχθη ως δείκτης για το ενδιαφέρον που θα προκαλέσει στο μαθητή η θέαση του αντικειμένου ή και η ανάγνωση της σχετικής πληροφορίας. Σκοπός της εισαγωγής του ήταν η ύπαρξη ενός ακόμα κριτηρίου για την επιλογή των αντικειμένων μάθησης των διαδρομών, με γνώμονα τις ηλικίες των μαθητών στους οποίους θα ανατεθούν.

Το πεδίο **Generality** αφορά τη γενικότητα της γνώσης που προσφέρει το αντικείμενο μάθησης. Η εισαγωγή του είχε σαν κύριο στόχο να υπάρχει ένας επιπλέον παράγοντας που θα καθορίζει τη σειρά εμφάνισης των αντικειμένων όταν αυτά ανήκουν στην ίδια χρονολογία.

Το πεδίο **Quality** αφορά στην ποιότητα της διαθέσιμης φωτογραφίας και επίσης συμβαδίζει με την πιθανότητα να συμπεριληφθεί το αντικείμενο σε μια εκπαιδευτική διαδρομή.

Τέλος, το πεδίο **Time** αποτελεί μια εκτίμηση του χρόνου σε λεπτά που θα χρειαστεί ο μαθητής για να επεξεργαστεί τις πληροφορίες του αντίστοιχου αντικειμένου μάθησης. Η εισαγωγή του είχε σα στόχο τη ρύθμιση από τον εκπαιδευτικό του χρόνου που θα διαρκέσει η παραγόμενη διαδρομή. Η υπάρχουσα τιμή του καθορίστηκε με βάση το μέγεθος του κειμένου που αντιστοιχεί σε κάθε αντικείμενο στον πίνακα information, αλλά και του πεδίου Difficulty.

Τα παραπάνω πεδία, χρησιμοποιούνται, όπως θα δούμε, για τον καθορισμό του στόχου του εκάστοτε προβλήματος σχεδιασμού ηλεκτρονικής μάθησης που προκύπτει κατά τη δημιουργία μιας εκπαιδευτικής διαδρομής, αλλά και ως δείκτες ποιότητας του τελικού σχεδίου, δηλαδή της παραγόμενης διαδρομής.

Τις τιμές αυτών των χαρακτηριστικών επιτρέπεται να τις αλλάζουν μόνο όσοι έχουν πρόσβαση στην αντίστοιχη σελίδα της εφαρμογής, αυτοί δηλαδή που ανήκουν στην κατηγορία «τεκμηριωτές» (experts).

Στον σχήμα 3.2-2 φαίνονται μερικά παραδείγματα εγγραφών του πίνακα Object της βάσης δεδομένων Corgialen.

idObject	Name	LanguageName	Description	Argostoli	Liksouri	Objects	Buildings
3	Photo	ENG	Argostoli. Earthquakes in y	n	n	n	n
4	Photo	ENG	Argostoli. Before the ear n	n	n	n	y
5	Photo	ENG	Argostoli. 1916 to 1917. N y	n	n	n	n
6	Photo	ENG	Argostoli. 1916 to 1917. N y	n	n	n	n

Earthquake	Relations	Persons	Dating	importance	difficulty	beauty	generality	quality	time
y	n	n	1953	6	2	4	7	2	13
n	n	n	1951	9	10	7	1	0	12
n	y	n	1916	9	8	9	0	2	8
n	y	n	1916	3	1	4	9	8	2

**Corgialen.Object 3.2-2**

## 2) information

Ο πίνακας αυτός περιέχει στο πεδίο Perigrafi ένα κείμενο με πληροφορίες σχετικές με το έκθεμα με το αντίστοιχο Id. Τα κείμενα αυτά υπήρχαν στον πίνακα Apeikonish της Korgialenios και χρησιμοποιούνται μαζί με τις φωτογραφίες ως αντικείμενα μάθησης. Μερικά παράδειγμα εγγραφών φαίνονται στο σχήμα 3.2-3:

idApeikonisi	idAntikeimenou	Perigrafi
1	63	Ιονική Τράπεζα. Αρμονικός συνδυασμός νεοκλασικής και
2	9	Πάνος Αποστολάτος, Καροτσιέρης και νεκροπομπός για όσ
3	500	Η οικία της οικογένειας του κόντε Τζώρτζη Άννινου με ωρα
5	207	Ο φωτισμός της πόλης του Αργοστολίου ήταν ένα θέμα πο

**Corgialen.Information 3.2-3**

## 3) photos

Ο πίνακας Photos περιέχει στοιχεία του πίνακα Anaparastasi της βάσης Korgialenios, μεταξύ των οποίων και το πεδίο Syndesmos με τη διεύθυνση και το όνομα του αρχείου που περιέχει τη φωτογραφία του εκθέματος με το αντίστοιχο Id. Το πεδίο που προσθέσαμε είναι το Exists, που δηλώνει το αν υπάρχει το αντίστοιχο αρχείο στο σύστημά μας, αν έχουμε δηλαδή τη φωτογραφία στη διάθεσή μας. Κάποια παραδείγματα εγγραφών φαίνονται στο ακόλουθο σχήμα:

idAnaparastasi	idAntikeimenou	Eidosanaparastasis	GlossaOn	EidosArxeiou	PsifiakiSyllogi	Syndesmos	Exists
1	3	Photo	GRE	TIF		1 My Disc (E)\Seismoi 1953\EQ75.tif	n
2	4	Photo	GRE	TIF		2 My Disc (E)\TPA1\tp1332.tif	y
3	5	Photo	GRE	TIF		3 My Disc (E)\TPA1\tp111.tif	y
4	6	Photo	GRE	TIF		4 My Disc (E)\TPA1\tp118.tif	y

**Corgialen.Photos 3.2-4**

### 3.2.2 Πίνακες Διαχείρισης

Στους πίνακες αυτούς αποθηκεύονται δεδομένα που παράγονται κατά τη διάρκεια χρήσης του συστήματος και είναι απαραίτητα για τις εκπαιδευτικές λειτουργίες που προσφέρει.

#### 4) routes

Εδώ αποθηκεύονται με τη μορφή συμβολοσειρών οι εκπαιδευτικές διαδρομές που παράγονται από την εφαρμογή. Το πεδίο route αποτελείται από τα Id των αντικειμένων μάθησης της διαδρομής με τη σειρά που αυτά θα παρουσιαστούν. Στο πεδίο routeName εκχωρείται το όνομα που δίνει ο εκπαιδευτικός που δημιούργησε την αντίστοιχη διαδρομή.

routeId	route	routeName
10	298 423 32	Liksouri1
11	301 533 436 272 332 292 231	Seismos
12	302 301 100 432 274 272 333 292 231	Adiadromh

Corgialen.Routes 3.2-5

#### 5) studentroutes

Εδώ καταχωρούνται οι σχέσεις μαθητών και διαδρομών, μετά την ανάθεση των τελευταίων στους πρώτους από τους δασκάλους τους. Το πεδίο done είναι 0 όταν ο μαθητής με userId το studentId δεν έχει «εκτελέσει» τη διαδρομή με Id το αντίστοιχο routeId, και 1 στην αντίθετη περίπτωση. Υπενθυμίζεται ότι το userId είναι πεδίο του πίνακα jos\_users που βρίσκεται στη βάση του Joomla. Στο ακόλουθο σχήμα έχουμε παραδείγματα εγγραφών του πίνακα StudentRoutes:

StudentRouteId	studentId	routeId	done
1	54	1	0
2	51	1	0
3	53	23	0
4	54	23	0

Corgialen.StudentRoutes 3.2-6

#### 6) teacherstudents

Εδώ εκφράζονται οι σχέσεις δασκάλου-μαθητών. Κάθε εγγραφή δηλώνει ότι ο δάσκαλος με userId το teacherId έχει στην τάξη του το μαθητή με το αντίστοιχο studentId.



idTeacherStudent	teacherId	studentId
1	50	55
2	50	56
3	50	59

**Corgialen.TeacherStudents 3.2-7**

### 7) tests

Εδώ αποθηκεύονται τα διαγωνίσματα πολλαπλής επιλογής που δημιουργούν οι δάσκαλοι για τους μαθητές. Το πεδίο queNum δηλώνει τον αριθμό των ερωτήσεων από τις οποίες αποτελείται το αντίστοιχο test.

testId	testName	queNum
1	myTest	5
2	TEST1	2
3	TheTest	2
4	First Test	10

**Corgialen.Tests 3.2-8**

### 8) questions

Εδώ αποθηκεύονται οι ερωτήσεις των διαγωνισμάτων πολλαπλής επιλογής. Στο πεδίο question αποθηκεύεται η ερώτηση, στα 4 επόμενα οι πιθανές απαντήσεις και στο πεδίο correct ο αριθμός της σωστής απάντησης.

queId	question	answer1	answer2	answer3	answer4	correct
0	In what y	1914	1922	1953	1959	1
1	Who was	Ioannis K	Georgios L	Georgios P	Marinos Ko	1
2	Who was	Ioannis K	Georgios L	Georgios P	Marinos Ko	3
3	In what y	1924	1929	1942	1953	2

**Corgialen.questions 3.2-9**

### 9) testque

Εδώ δηλώνεται το ποιες ερωτήσεις περιλαμβάνει κάθε διαγώνισμα. Για παράδειγμα στο επόμενο σχήμα φαίνεται ότι το διαγώνισμα με testId=4 περιλαμβάνει τις ερωτήσεις με queId 5, 6, 7 και 8.

testQueId	testId	queId
3	4	5
4	4	6
5	4	7
6	4	8

**Corgialen.testque 3.2-10**

### 10) studenttests

Τέλος, ο πίνακας studenttests είναι παρόμοιος με τον studentroutes και αντιστοιχίζει τα διαγωνίσματα στους μαθητές στους οποίους τα ανέθεσε ο δάσκαλος που τα συνέθεσε. Αυτή τη φορά όμως, το πεδίο done περιέχει την επίδοση του μαθητή στο αντίστοιχο διαγώνισμα, ενώ είναι προεπιλεγμένα 0 σε περίπτωση που ο μαθητής δεν έχει προσπαθήσει να το κάνει.

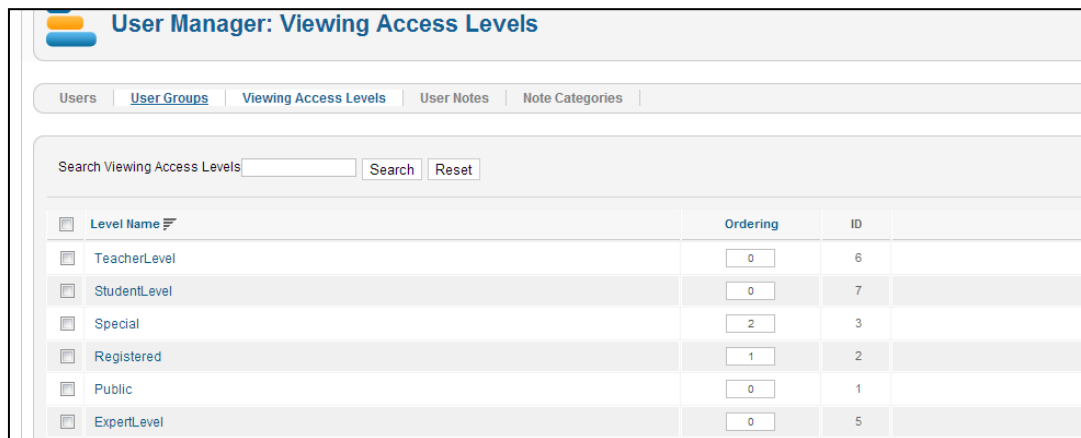
idStudentTest	studentId	testId	done
1	51	1	2
2	53	1	0
3	52	2	0

**Corgialen.Studenttests 3.2-11**

## 4 ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ

### 4.1 ΧΡΗΣΤΕΣ-ΕΠΙΠΕΔΑ ΠΡΟΣΒΑΣΗΣ

Οι χρήστες της ιστοσελίδας μπορούν να διαχωριστούν στις 6 παρακάτω κατηγορίες. Οι κατηγορίες αυτές αποτελούν ταυτόχρονα και τα διαφορετικά επίπεδα πρόσβασης του συστήματος με βάση τις λειτουργίες που μπορούν να εκτελέσουν.



<input type="checkbox"/> Level Name	Ordering	ID
<input type="checkbox"/> TeacherLevel	0	6
<input type="checkbox"/> StudentLevel	0	7
<input type="checkbox"/> Special	2	3
<input type="checkbox"/> Registered	1	2
<input type="checkbox"/> Public	0	1
<input type="checkbox"/> ExpertLevel	0	5

Κατηγορίες Επιπέδων Πρόσβασης 4.1-1

#### 4.1.1 Public

Στην κατηγορία αυτή ανήκουν οι απλοί επισκέπτες της ιστοσελίδας. Έχουν πολύ περιορισμένες δυνατότητες αφού μπορούν μόνο να περιηγηθούν στα άρθρα που περιέχουν γενικές πληροφορίες για το μουσείο και το νησί. Δεν έχουν πρόσβαση σε καμία εκ των εκπαιδευτικών λειτουργιών αν δεν εγγραφούν (Register) στο σύστημα. Το επίπεδο πρόσβασης που αντιστοιχεί στην κατηγορία αυτή των χρηστών είναι το *Public* και περιλαμβάνει λειτουργίες προσβάσιμες σε όλους τους χρήστες.

#### 4.1.2 Registered

Εδώ ανήκουν οι χρήστες που έχουν ήδη επισκεφτεί το σύστημα τουλάχιστον μία φορά και έχουν δημιουργήσει λογαριασμό, κάνοντας εγγραφή. Έχουν καταχωρηθεί, δηλαδή, στη βάση δεδομένων του συστήματος. Αυτόματα, οι χρήστες αυτοί τοποθετούνται στην κατηγορία Student και αν δεν κάνουν αίτηση για καταχώρηση σε κάποια από τις υπόλοιπες κατηγορίες, μπορούν να εκτελέσουν τις λειτουργίες που αντιστοιχούν στους μαθητές.

### 4.1.3 Student

Η κατηγορία αυτή αντιστοιχεί στους μαθητές, που επισκέπτονται την ιστοσελίδα στα πλαίσια του σχολικού μαθήματος. Το επίπεδο στο οποίο έχουν πρόσβαση είναι το *StudentLevel*. Αυτό, πέρα από τις σελίδες γενικών πληροφοριών, περιλαμβάνει εκπαιδευτικές λειτουργίες. Οι μαθητές έχουν πρόσβαση σε σελίδες όπου μπορούν να ακολουθήσουν τις εκπαιδευτικές διαδρομές και να λύσουν τα διαγωνίσματα που τους έχουν αναθέσει οι δάσκαλοί τους. Κάποια στοιχεία που αφορούν τις ενέργειές τους, όπως για παράδειγμα η θέαση των αντικειμένων μιας διαδρομής ή ο βαθμός τους σε ένα διαγώνισμα, αποθηκεύονται στη βάση ώστε να μπορούν να αξιολογηθούν από τους δασκάλους.

### 4.1.4 Teacher

Εδώ ανήκουν οι δάσκαλοι που χρησιμοποιούν το σύστημα και το επίπεδο πρόσβασης που τους αντιστοιχεί είναι το *TeacherLevel*. Μέσα από τις σελίδες στις οποίες έχουν πρόσβαση μπορούν όπως θα δούμε να δημιουργήσουν εκπαιδευτικές διαδρομές και να τις αναθέσουν στους μαθητές που θα επιλέξουν. Αντίστοιχα μπορούν να δημιουργήσουν και να αναθέσουν στους μαθητές διαγωνίσματα πολλαπλής επιλογής και να δουν την απόδοσή τους σε αυτά.

### 4.1.5 Expert

Στην κατηγορία αυτοί ανήκουν οι «τεκμηριωτές». Αυτοί μπορεί να είναι δάσκαλοι, καθηγητές ή ιστορικοί, οι οποίοι είναι ικανοί να αξιολογήσουν τα μαθησιακά χαρακτηριστικά των αντικειμένων μάθησης (Learning Objects) που εξηγήσαμε κατά την περιγραφή της βάσης δεδομένων του συστήματος. Το επίπεδο πρόσβασης που τους αντιστοιχεί είναι το *ExpertLevel*, το οποίο περιλαμβάνει τη σελίδα όπου καταχωρούνται οι τιμές των μαθησιακών χαρακτηριστικών κάθε αντικειμένου, όπως θα δούμε παρακάτω.

### 4.1.6 Super User-Administrator

Η κατηγορία αυτή περιλαμβάνει τους διαχειριστές της ιστοσελίδας, δηλαδή τα πρόσωπα στα οποία έχει ανατεθεί η συντήρησή της, που μπορεί να είναι ένα ή περισσότερα. Το επίπεδο πρόσβασης που τους αντιστοιχεί είναι το *Special*. Οι διαχειριστές έχουν πρόσβαση στη λειτουργία back-end της ιστοσελίδας και στα στοιχεία της βάσης. Στις αρμοδιότητές τους συμπεριλαμβάνονται η αλλαγή κατηγοριών στους χρήστες όταν αυτό κρίνεται αναγκαίο, η εισαγωγή νέων δεδομένων ή αντικειμένων μάθησης, η εκτέλεση των απαραίτητων ενεργειών για την αλλαγή του κωδικού πρόσβασης κάποιου χρήστη όταν αυτός τον ξεχάσει κλπ.

## 4.2 ΛΕΙΤΟΥΡΓΙΕΣ

Η ιστοσελίδα έχει δημιουργηθεί χρησιμοποιώντας το πλαίσιο λειτουργιών του Joomla!. Η αρχική σελίδα παρουσιάζει στα αριστερά τις επιλογές του επισκέπτη:

**Κοργιαλένιο Ίδρυμα**

Το Νησί

**Το Κοργιαλένιο Ίδρυμα**

Μια γενική διαδρομή!

Επικοινωνία

**User Menu**

Your Profile

**Login**

Login

**e-learning**

Δημιουργία Διαδρομών

Τεκμηρίωση Αντικειμένων Μάθησης

Διαγωνίσματα

Διαδρομές

Οι μαθητές μου

**Το Κοργιαλένιο Ίδρυμα**

Η Κοργιαλένεια Βιβλιοθήκη Αργοστολίου ιδρύθηκε το 1924, σε εκτέλεση της διαθήκης του μεγάλου ευεργέτη της Κεφαλληνίας Μαρίνου Κοργιαλένιου, σύμφωνα με τα όσα προβλέπονταν στο άρθρο 27 της διαθήκης του από 15-6-1910 ("Κληροδότημα προς Ίδρυση Δημοσίας Βιβλιοθήκης εν Αργοστολίου").

Ο Κοργιαλένιος είχε αφήσει συνολικό ποσό 10.000 λιρών για την ανέγερση της βιβλιοθήκης (4.000 λίρες), για την αγορά βιβλίων (1.000 λίρες) και για τη συντήρηση της Βιβλιοθήκης (5.000 λίρες). Κληροδότησε επίσης και τη συλλογή των βιβλίων του.

Η Βιβλιοθήκη ξεκίνησε τη λειτουργία της το 1925, ενώ το 1926 με απόφαση του Υπουργείου Παιδείας συγχωνεύθηκε με αυτήν η παλαιότερη Δημοσία Βιβλιοθήκη Κεφαλληνίας, που λειτουργούσε στο Αργοστόλι από το 1887. Το πρώτο υλικό της Κοργιαλένειας Βιβλιοθήκης αποτέλεσαν τα βιβλία της παλιάς Βιβλιοθήκης, και η συλλογή του Μαρίνου Κοργιαλένιου. Πρώτος και μοναδικός σε όλη τη διάρκεια του 20ού αιώνα διευθυντής της Βιβλιοθήκης ήταν ο Γεράσιμος Χ. Μοσχόπουλος, άνθρωπος με ευρύτατη μόρφωση και εξαιρετικό ζήλο για το Ίδρυμα, ο οποίος πρόσφερε τις υπηρεσίες του στη Βιβλιοθήκη για περίπου 4 δεκαετίες και μετά από αιπίες συνθήκες. Η Βιβλιοθήκη με τον Μοσχόπουλο στο τιμόνι και υπό την διοίκηση του Κοργιαλένιου Κληροδοτήματος, στεγασμένη στο πολυτελές κτίριό της, είχε αναδειχθεί ήδη από τη δεκαετία του 1930 σε πνευματικό φάρο της πόλεως του Αργοστολίου.

Ο πόλεμος του 1940, η κατοχή και η εξόνωση των κεφαλαίων του Κληροδοτήματος όμως οδήγησαν τη Βιβλιοθήκη σε σοβαρή κρίση. Οι σεισμοί του 1953, που κατέστρεψαν το κτίριο ήταν το τελειωτικό χτύπημα. Το βιβλιακό υλικό ευτυχώς περισώθηκε χάρη στην αυταπάρνηση των υπαλλήλων του κληροδοτήματος, και το κράτος καθιέρωσε με το Νόμο 2823/1954 την τακτική επιχορήγηση της Βιβλιοθήκης, για να εξασφαλιστεί η συνέχιση της λειτουργίας της.

Μετά τους σεισμούς του 1953 το υλικό της βιβλιοθήκης μεταφέρθηκε στα γιουγκοσλαβικά "τολ", που είχαν σπηθεί πρόχειρα στην πύλη και κατόπιν στο κτίριο του Χαροκοπέιου Ιδρύματος.

Με επίμονη προσπάθεια του Κοργιαλένιου Διοικητικού Συμβουλίου και με την ευγενική χορηγία των

Αρχική Σελίδα Εφαρμογής 4.2-1

Τα μενού που βρίσκονται στα αριστερά της σελίδας είναι διαθέσιμα σε οποιοδήποτε σημείο της περιήγησης και υποδεικνύουν τις επιλογές του χρήστη. Το μενού **Κοργιαλένιο Ίδρυμα** περιέχει σελίδες με γενικές πληροφορίες για το Κοργιαλένιο Ίδρυμα και την ιστορία του νησιού, το **e-learning** περιέχει τις σελίδες όπου εκτελούνται οι εκπαιδευτικές λειτουργίες της ιστοσελίδας, ενώ το **Login** οδηγεί τον επισκέπτη στη σελίδα που μπορεί να συνδεθεί στο σύστημα ή να αποσυνδεθεί από αυτό, ή και να δημιουργήσει έναν καινούργιο λογαριασμό. Όταν κάποιος χρήστης συνδεθεί, εμφανίζεται ως επιλογή και το μενού User Profile.

Πατώντας κάποια από τις επιλογές στα αριστερά ο χρήστης οδηγείται στην αντίστοιχη σελίδα. Για τον απλό επισκέπτη, δηλαδή αυτόν που δεν έχει εισέλθει στο σύστημα κάνοντας Login, προσβάσιμες είναι μόνο οι σελίδες του πρώτου μενού με τις γενικές πληροφορίες. Οι επιμέρους λειτουργίες περιγράφονται αναλυτικά παρακάτω.

## 4.2.1 e-learning

Οι σελίδες που συνδέονται με το μενού αυτό είναι ο πυρήνας των εκπαιδευτικών λειτουργιών και περιλαμβάνουν την ουσιαστική χρησιμότητα του συστήματος. Προσφέρουν στους δασκάλους τη δυνατότητα να δημιουργήσουν εκπαιδευτικές διαδρομές, αξιοποιώντας τη βάση δεδομένων του συστήματος, καθώς και διαγωνίσματα με ερωτήσεις πολλαπλής επιλογής για την αξιολόγηση των μαθητών.

Στις σελίδες αυτές ο μαθητής μπορεί να εισέλθει για να «ακολουθήσει» τις διαδρομές που του υποδεικνύει το σύστημα και να απαντήσει τα διαγωνίσματα-ερωτήσεις που του έχουν ανατεθεί.

Τέλος, με την επιλογή «Τεκμηρίωση Αντικειμένων Μάθησης», οι «τεκμηριωτές» μπορούν να εισέλθουν και να αξιολογήσουν τα μαθησιακά χαρακτηριστικά των αντικειμένων της βάσης.

### 4.2.1.1 Δημιουργία Διαδρομών

ΔΙΑΛΕΞΕ ΚΑΤΗΓΟΡΙΕΣ	ΜΑΘΗΣΙΑΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ	ΕΠΙΛΟΓΕΣ ΣΧΕΔΙΟΥ
<input type="checkbox"/> Αργιστόλι	Importance from 0 to 10	Μέγιστος αριθμός αντικειμένων σε κάθε επίπεδο: 10
<input type="checkbox"/> Ληξούρι	Difficulty from 0 to 10	Importance weight: 1
<input type="checkbox"/> Αντικείμενα	Generality from 0 to 10	Difficulty weight: 1
<input type="checkbox"/> Σεισμός	Beauty from 0 to 10	Generality weight: 1
<input type="checkbox"/> Διπλωματία	Quality from 0 to 10	Beauty weight: 1
<input type="checkbox"/> Προσωπικότητες	Time from 1 to 10	Quality weight: 1
<input type="checkbox"/> Σημαντικά κτίρια		
Δώσε όνομα στην νέα διαδρομή		<b>Δημιουργία διαδρομής</b>

Δημιουργία Διαδρομών 4.2-2

Η σελίδα αυτή είναι προσβάσιμη μόνο από τους δασκάλους. Εδώ οι τελευταίοι δημιουργούν τις εκπαιδευτικές διαδρομές και στη συνέχεια τις αναθέτουν στους μαθητές τους. Τον ακριβή τρόπο λειτουργίας της θα τον εξηγήσουμε πλήρως στην ανάλυση του αντίστοιχου αρχείου PHP και των αρχείων PDDL που αυτό παράγει. Τα αρχεία PDDL είναι οι περιγραφές του κόσμου (domain file) και του προβλήματος (problem file) που θα αποτελέσουν τις εισόδους του σχεδιαστή LPG, ο οποίος θα τα χρησιμοποιήσει για να παράξει το σχέδιο λύσης του προβλήματος, δηλαδή μια εκπαιδευτική διαδρομή που πληροί τις προϋποθέσεις που έθεσε ο δάσκαλος.

Στα 2 πρώτα πλαίσια της σελίδας ο δάσκαλος κάνει ένα φιλτράρισμα των αντικειμένων που θα χρησιμοποιήσει η εφαρμογή για την παραγωγή της διαδρομής, ενώ στο τρίτο επεμβαίνει ως ένα μικρό βαθμό στο σχέδιο που παράγεται από το σχεδιαστή.

Στο πλαίσιο «ΔΙΑΛΕΞΕ ΚΑΤΗΓΟΡΙΕΣ» ο δάσκαλος επιλέγει τις κατηγορίες αντικειμένων που θέλει να περιέχει η διαδρομή. Μπορεί να επιλέξει όσες κατηγορίες θέλει και σε πρώτη φάση να αποκλείσει τα αντικείμενα που δεν ανήκουν σε καμία από αυτές. Για παράδειγμα, επιλέγοντας τις δύο πρώτες κατηγορίες, η εφαρμογή θα κρατήσει μόνο τα αντικείμενα που ανήκουν είτε στην κατηγορία “*Argostoli*” είτε στην κατηγορία “*Liksouri*”. Επίσης, στο κάτω μέρος του πλαισίου ζητάται από το χρήστη να ορίσει ένα όνομα για την παραγόμενη διαδρομή.

Στη συνέχεια, στο πλαίσιο «ΜΑΘΗΣΙΑΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ» ο δάσκαλος καθορίζει τα όρια των τιμών των μαθησιακών χαρακτηριστικών των αντικειμένων. Για παράδειγμα, αν ο προηγούμενος χρήστης ορίσει ότι το πεδίο *Beauty* θέλει να έχει τιμή από 5 ως 10, αφήνοντας τα άλλα ως έχουν, η εφαρμογή θα κρατήσει από τα αντικείμενα που έχουν μείνει μετά το πρώτο φιλτράρισμα μόνο όσα έχουν τις αντίστοιχες τιμές *Beauty*.

Στο πλαίσιο «ΕΠΙΛΟΓΕΣ ΣΧΕΔΙΟΥ» ο δάσκαλος ουσιαστικά προσδιορίζει τα στοιχεία που θα καθορίσουν την ποιότητα της παραγόμενης διαδρομής, άρα και την παραγωγή του σχεδίου λύσης από το σχεδιαστή. Η πλήρης εξήγηση της σημασίας των επιλογών του γίνεται στην ανάλυση των αρχείων PHP. Στο σημείο αυτό αναφέρουμε απλώς ότι αμέσως πριν τη δημιουργία των αρχείων PDDL, τα αντικείμενα μάθησης που έχουν προκύψει μετά τα φιλτραρίσματα διαχωρίζονται σε πλειάδες, αρχικά ανάλογα με τη χρονολογία τους. Τα αντικείμενα που αντιστοιχούν σε πλειάδες χαμηλότερου επιπέδου εισάγονται στη διαδρομή νωρίτερα, ώστε η διαδρομή να ακολουθεί χρονολογική πορεία. Από κάθε μια από τις πλειάδες αυτές, το τελικό σχέδιο πρέπει να περιέχει ένα σύνολο αντικειμένων μάθησης που αθροιστικά να καλύπτουν ένα ελάχιστο όριο συνδυασμού τιμών μαθησιακών χαρακτηριστικών.

Ο συνδυασμός αυτός καθορίζεται από τη βαρύτητα που εκτιμά ο δάσκαλος για κάθε text field (Importance weight, Difficulty weight κλπ) στο πλαίσιο «ΕΠΙΛΟΓΕΣ ΣΧΕΔΙΟΥ», έχοντας υπόψη του τους μαθητές στους οποίους θα αναθέσει τη διαδρομή μάθησης. Για παράδειγμα, έστω ότι ένα αντικείμενο μάθησης (έκθεμα) έχει τιμές μαθησιακών χαρακτηριστικών Importance=1, Difficulty=2, Generality=3, Beauty=4, Quality=5 και ο δάσκαλος θέτει τα βάρη 1, 3, 5, 7 και 9 αντίστοιχα. Τότε ο συνδυασμός που κρίνει τη συμβολή του συγκεκριμένου αντικειμένου στην ποιότητα του σχεδίου λύσης, οπότε και την πιθανότητα επιλογής του από το σχεδιαστή, είναι το άθροισμα  $1*1+2*3+3*5+4*7+5*9=95$ .

Όσον αφορά στις πλειάδες που αναφέραμε, στην περίπτωση που σε κάποιες χρονολογίες υπάρχει μεγάλο πλήθος αντικειμένων μάθησης, αποφασίσαμε ότι έπρεπε να εισαχθεί ένα κριτήριο περαιτέρω διαχωρισμού ανάλογα με τα μαθησιακά χαρακτηριστικά των αντικειμένων. Σκοπός μας ήταν στις περιπτώσεις αυτές να εμφανίζονται νωρίτερα στη διαδρομή τα αντικείμενα που έχουν μεγαλύτερη σημασία ή προσφέρουν γενικότερη γνώση, ώστε να γίνεται πιο εύκολη η αφομοίωση της γνώσης από τους μαθητές.

Για το λόγο αυτό, στο πρώτο text field του πλαισίου «ΕΠΙΛΟΓΕΣ ΣΧΕΔΙΟΥ», ο δάσκαλος καθορίζει το μέγιστο αριθμό αντικειμένων μάθησης που μπορεί να περιέχει κάθε επίπεδο. Σε περίπτωση που μια πλειάδα, δηλαδή ένα σύνολο αντικειμένων που αντιστοιχούν στην ίδια χρονολογία, έχει πλήθος

μεγαλύτερο από το όριο αυτό, τότε τα αντικείμενα αυτά ταξινομούνται σε φθίνουσα σειρά ανάλογα με το άθροισμα *importance+generality* που αντιστοιχεί στο καθένα και διαχωρίζονται σε μικρότερες πλειάδες, το πλήθος καθεμιάς από τις οποίες δεν πρέπει να ξεπερνά το όριο που τέθηκε.

Με τον τρόπο αυτό, το όριο λειτουργεί ως ένας δείκτης ελευθερίας του σχεδιαστή. Όσο λιγότερα αντικείμενα αντιστοιχούν σε κάθε επίπεδο, τόσο πιο περιορισμένες είναι οι επιλογές του σχεδιαστή, αφού πρέπει να επιλέξει ανάμεσα σε λιγότερα αντικείμενα κάθε φορά. Αντίθετα, στις πλειάδες που περιέχουν μεγάλο αριθμό αντικειμένων, ο σχεδιαστής έχει να επιλέξει από ένα σύνολο συνδυασμών αυτόν που εξυπηρετεί καλύτερα την λύση του προβλήματος που αντιμετωπίζει.

Στα 2 τελευταία πλαίσια, ο δάσκαλος πρέπει να σκεφτεί τον/τους μαθητές στους οποίους θα αναθέσει τη διαδρομή και να κάνει τις κατάλληλες επιλογές ώστε να εξατομικεύσει τη διαδρομή για τις ανάγκες των μαθητών.

Για παράδειγμα, αν έχει σκοπό να απευθυνθεί σε μαθητές μικρής ηλικίας, θα πρέπει να επιλέξει αντικείμενα με μικρό βαθμό δυσκολίας(*difficulty*) και μεγάλο βαθμό ομορφιάς(*beauty*) και ποιότητας(*quality*), ώστε να τους κεντρίσει περισσότερο το ενδιαφέρον. Θα πρέπει επίσης στο τρίτο πλαίσιο να δώσει μικρό βάρος στη δυσκολία και μεγάλο στη γενικότητα(*generality*).

Αν πάλι θέλει να απευθυνθεί σε μαθητές που έχουν δει αρκετές εκπαιδευτικές διαδρομές, τότε μπορεί να αυξήσει το κάτω όριο της δυσκολίας και να μειώσει το βάρος της σημασίας(*importance*), αφού κατά πάσα πιθανότητα θα έχει συμπεριλάβει τα αντίστοιχα αντικείμενα μάθησης αρκετές φορές σε προηγούμενες διαδρομές.

Αφού καθορίσει τις κατηγορίες, τα όρια των μαθησιακών χαρακτηριστικών και τις επιλογές σχεδίου, ο χρήστης πατά το πλήκτρο «ΔΗΜΙΟΥΡΓΙΑ ΔΙΑΔΡΟΜΗΣ». Με την ενέργεια αυτή γίνονται όλες οι λειτουργίες που περιγράψαμε, και θα εξηγήσουμε λεπτομερώς στην ανάλυση του αντίστοιχου αρχείου PHP, και ο σχεδιαστής παράγει την τελική διαδρομή, η οποία αποθηκεύεται στον πίνακα *Routes* της βάσης δεδομένων *Corgialen*.

Ο χρήστης οδηγείται στη συνέχεια στη σελίδα *Assign Routes to Students* (σχήμα 4.2-2), όπου επιλέγει τους μαθητές στους οποίους θέλει να αναθέσει τη διαδρομή που δημιουργήθηκε, η οποία εμφανίζεται στο πάνω μέρος της οθόνης. Πατώντας το πλήκτρο «Ανάθεση Διαδρομής» δημιουργούνται οι αντίστοιχες εγγραφές στον πίνακα *studentroutes*, που συνδέουν τη νέα διαδρομή με τους μαθητές στους οποίους αυτή ανατέθηκε.



**Κοργιαλένιο Ίδρυμα**

- Το Νησί
- Το Κοργιαλένιο Ίδρυμα
- Επικοινωνία

**User Menu**

- Your Profile

**Login**

- Login

**e-learning**

- Δημιουργία Διαδρομών
- Τεκμηρίωση Αντικειμένων Μάθησης
- Διαγωνίσματα
- Διαδρομές
- Οι μαθητές μου

### Assign Routes to Students

**Route:DIPLWMATIA1**

Learning route: 59 139 77 122 5 143 88 138 35 530 258 302 301 400 161 272 359 148 387 292

- Vasilis Vasilopoulos (id 50)**
- Hristos Haropoulos (id 51)**
- Kostas Hatzipetridis (id 52)**
- Giannis Feratzidis (id 53)**
- Giorgos Mendopoulos (id 54)**
- Alexis Alexandris (id 55)**
- Giorgos Georgiou (id 56)**
- Nikos Nikolaou (id 57)**
- Alekos Aleksiou (id 58)**
- Argiris Argiriou (id 59)**
- Maria Papadopoulou (id 60)**

**Ανάθεση Διαδρομής**

**Assign Route to Students 4.2-3**

#### 4.2.1.2 ΤΕΚΜΗΡΙΩΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ ΜΑΘΗΣΗΣ

Object	importance	difficulty	beauty	generality	quality	time
Object 3	2	6	7	6	8	1
Object 4	8	4	9	4	7	10
Object 5	9	2	6	8	5	8
Object 6	3	7	6	4	7	1
Object 9	8	6	5	10	6	9
Object 10						

#### Τεκμηρίωση Αντικειμένων Μάθησης 4.2-4

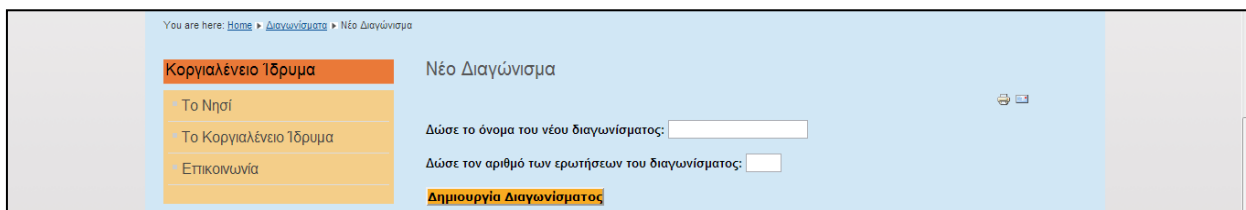
Η σελίδα αυτή είναι η μόνη στην οποία έχουν πρόσβαση μόνο οι «τεκμηριωτές» (Experts). Ανοίγοντάς τη βλέπουμε τα πεδία που φαίνονται στο σχήμα, ένα για κάθε αντικείμενο της βάσης (κάθε σελίδα περιέχει το πολύ 100 τέτοια πεδία και με συνδέσμους next και previous στο τέλος κάθε σελίδας μεταφερόμαστε στην επόμενη ή προηγούμενη 100αδα).

Τα πεδία αυτά περιέχουν το Id του αντικειμένου, έναν σύνδεσμο *view* που οδηγεί στη σελίδα απεικόνισης του αντικειμένου, 6 πλαίσια με τις υπάρχουσες τιμές των μαθησιακών χαρακτηριστικών του και ένα πλήκτρο Submit, ενώ περνώντας το ποντίκι πάνω από το πεδίο, εμφανίζεται η μικρή περιγραφή που περιέχεται στο πεδίο *Description* της αντίστοιχης εγγραφής του πίνακα *object*. Ο τεκμηριωτής έχει τη δυνατότητα να αλλάξει τις τιμές οποιουδήποτε χαρακτηριστικού οποιουδήποτε αντικειμένου και πατώντας το αντίστοιχο Submit button να καταχωρήσει τις καινούργιες τιμές στον πίνακα *object* της βάσης *corgialen*. Στην περίπτωση που θέλει να πραγματοποιήσει αλλαγές σε πλήθος αντικειμένων, τότε με το πλήκτρο **submit All** που φαίνεται στην αρχή της σελίδας, καταχωρεί στη βάση τις τιμές των χαρακτηριστικών όλων των αντικειμένων μάθησης που εμφανίζονται στη συγκεκριμένη σελίδα, δηλαδή των 100 αντικειμένων που φαίνονται εκείνη τη στιγμή.

#### 4.2.1.3 Διαγωνίσματα

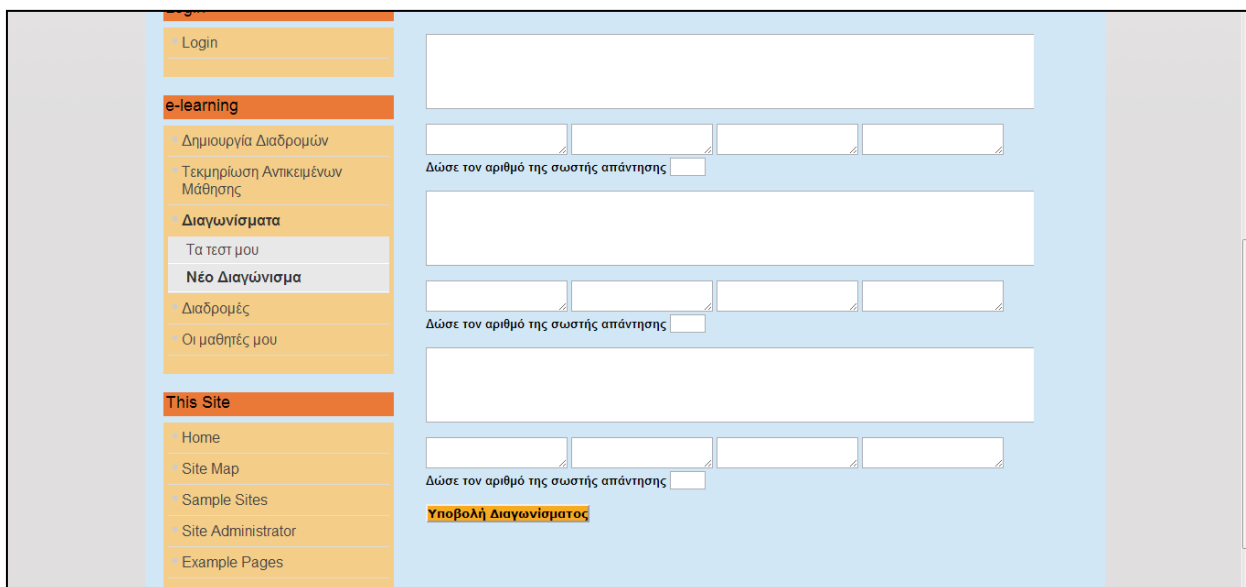
Η επιλογή αυτού του συνδέσμου μπορεί να οδηγήσει σε 2 διαφορετικές σελίδες, ανάλογα με το αν ο χρήστης είναι δάσκαλος ή μαθητής.

Ο δάσκαλος θα οδηγηθεί στη σελίδα Νέο Διαγώνισμα, που φαίνεται στο σχήμα 4.2-5, όπου μπορεί να διαμορφώσει ένα καινούργιο διαγώνισμα και να το αναθέσει στους μαθητές του. Αρχικά του ζητείται να δώσει το όνομα του διαγωνίσματος και τον αριθμό των ερωτήσεων πολλαπλής επιλογής που αυτό θα περιέχει.

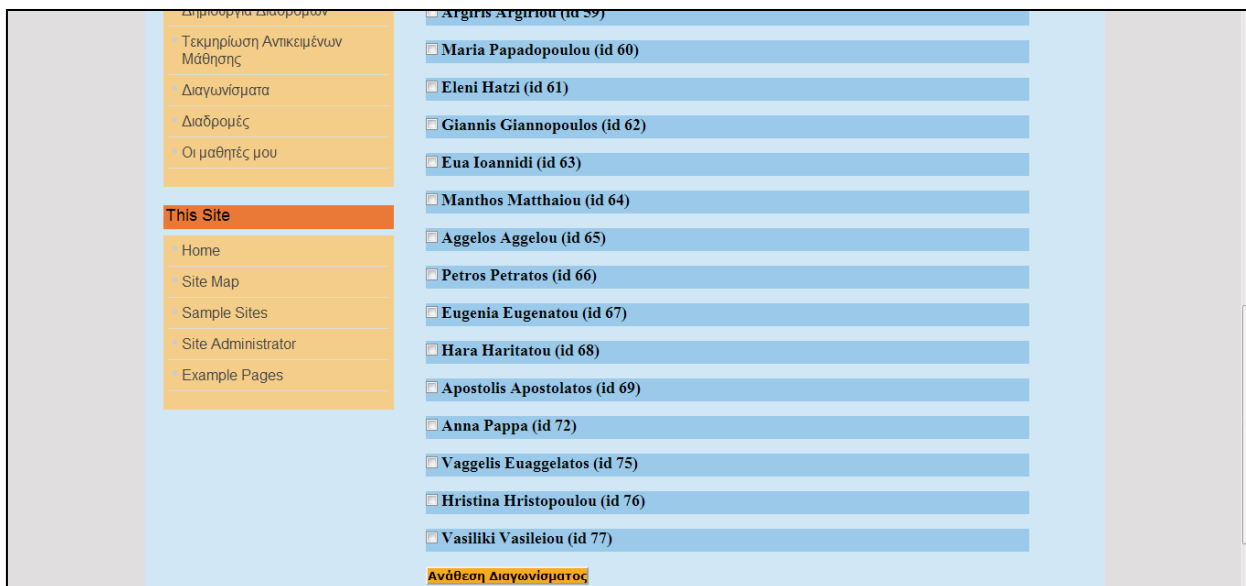


**Νέο Διαγώνισμα 4.2-5**

Στη συνέχεια οδηγείται σε μια σελίδα της μορφής του σχήματος 4.2-6, όπου πρέπει να γράψει τις ερωτήσεις και τις πιθανές απαντήσεις για κάθε ερώτηση και να δώσει τον αριθμό κάθε σωστής απάντησης. Πατώντας το πλήκτρο Υποβολή Διαγωνίσματος οδηγείται στη σελίδα του σχήματος 4.2-7, όπου αναθέτει το καινούργιο τεστ στους μαθητές που επιθυμεί.

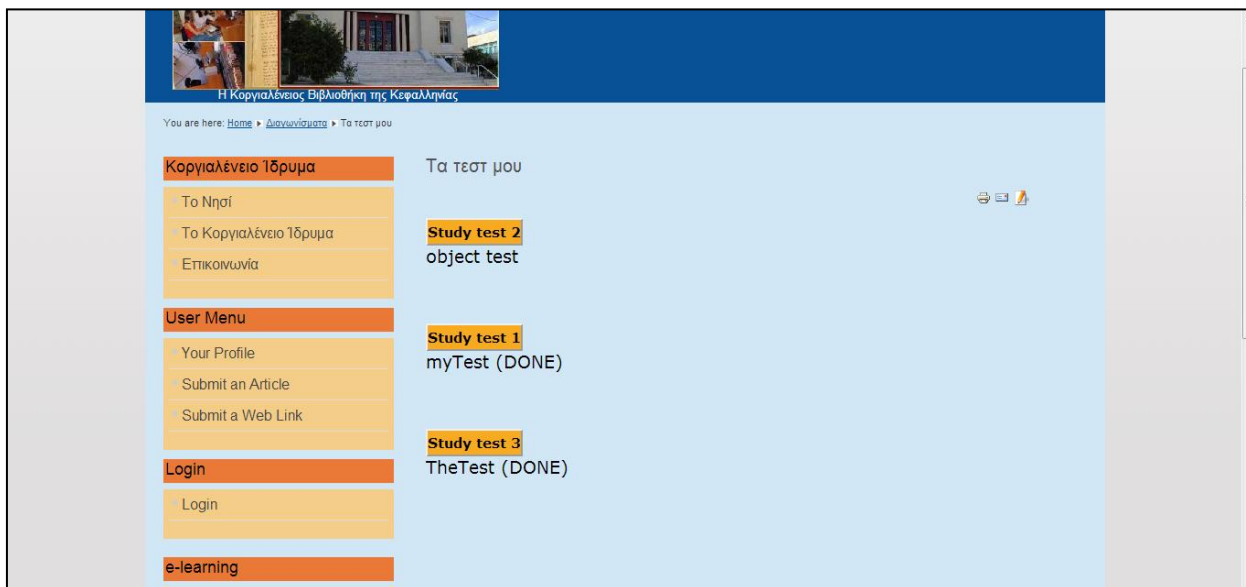


**Διαμόρφωση διαγωνίσματος 4.2-6**

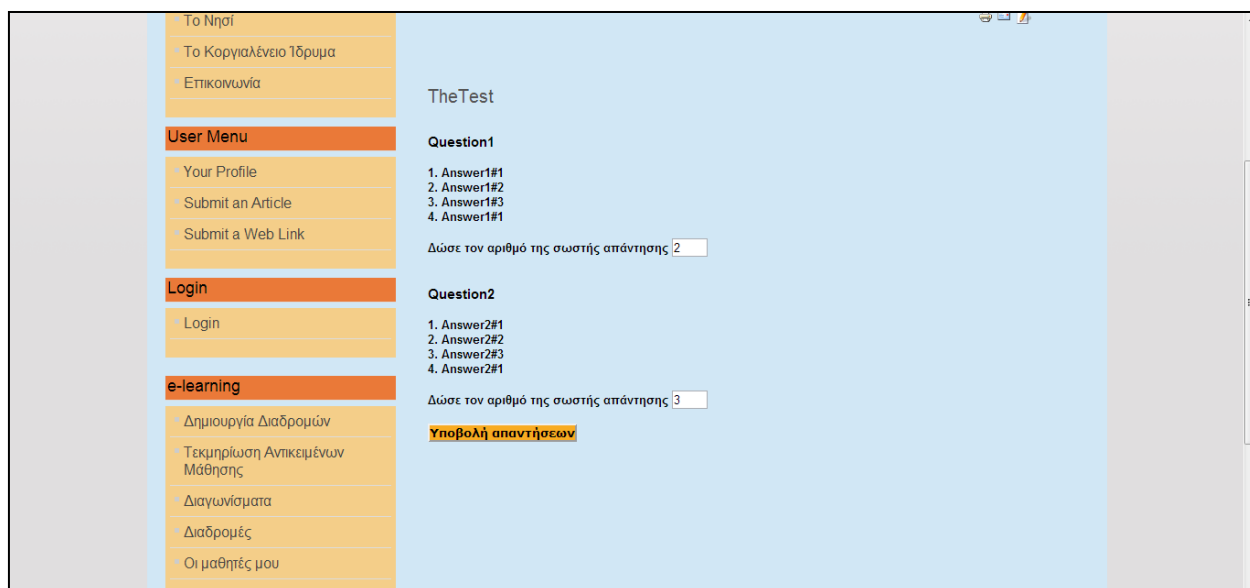


### Assign Test to students 4.2-7

Αν ο τρέχων χρήστης είναι εκπαιδευόμενος, έχει εισέλθει δηλαδή με στοιχεία λογαριασμού μαθητή, τότε επιλέγοντας το σύνδεσμο «Διαγωνίσματα» θα οδηγηθεί στη σελίδα «Τα Τεστ μου», που φαίνεται στο σχήμα 4.2-8. Εκεί μπορεί να δει τα διαγωνίσματα που του έχουν ανατεθεί. Αν έχει λύσει κάποιο από αυτά, δίπλα στο όνομα του αντίστοιχου διαγωνίσματος εμφανίζεται η ένδειξη DONE. Πατώντας το Study test οδηγείται στη σελίδα λύσης του αντίστοιχου διαγωνίσματος, που φαίνεται στο σχήμα 4.2-9.

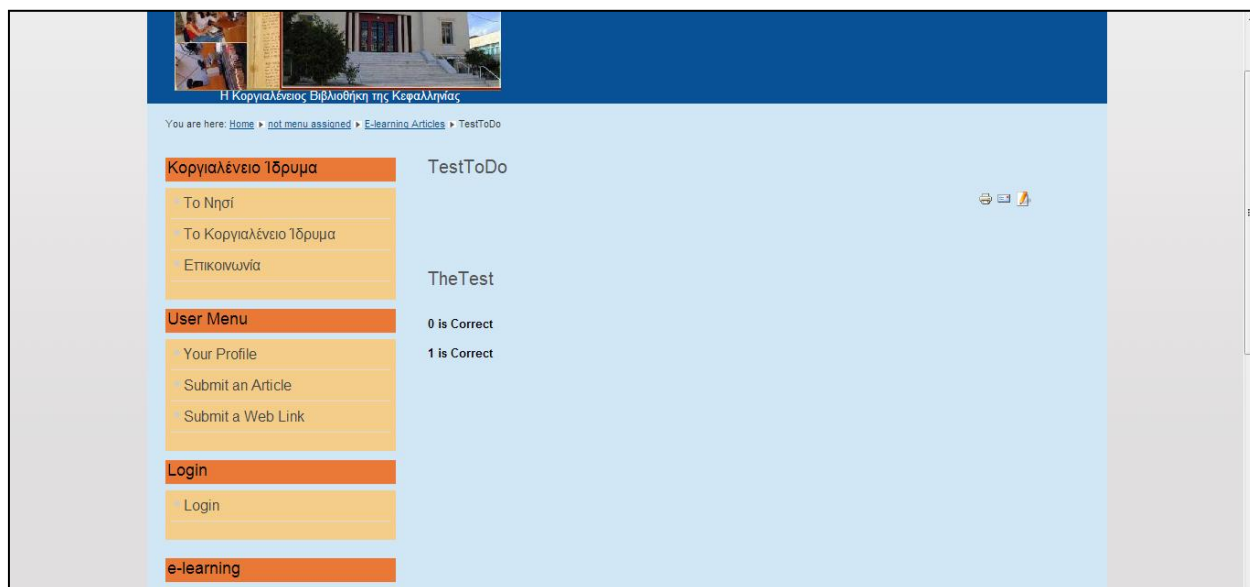


### Τα τεστ μου 4.2-8



#### Test To Do 4.2-9

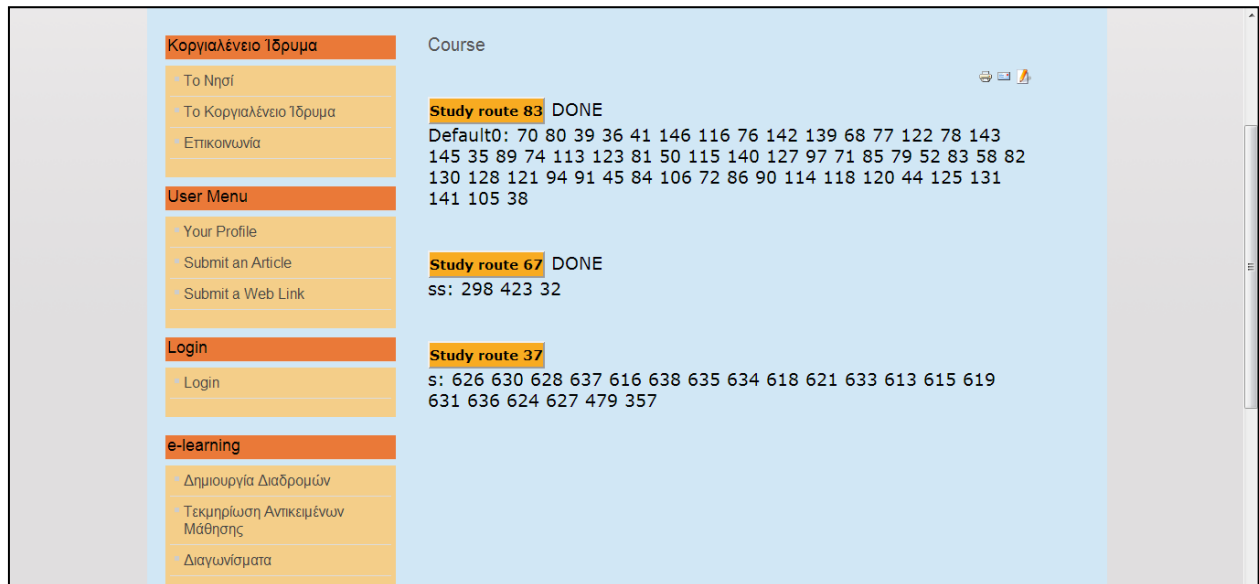
Εδώ ο μαθητής καλείται να ασχοληθεί με το διαγώνισμα. Για κάθε ερώτηση τοποθετεί στο πεδίο τον αριθμό της σωστής απάντησης. Για την υποβολή των απαντήσεων του πατά το αντίστοιχο Submit Button, για να εμφανιστούν τα αποτελέσματά του, όπως φαίνεται στο ακόλουθο σχήμα:



#### Αποτελέσματα απαντήσεων 4.2-10

Ταυτόχρονα η επίδοσή του αποθηκεύεται στο πεδίο “done” του πίνακα “studenttests”.

#### 4.2.1.4 Διαδρομές



#### Courses 4.2-11

Η επιλογή του συνδέσμου «Διαδρομές» οδηγεί το μαθητή στη σελίδα όπου φαίνονται οι εκπαιδευτικές διαδρομές που του έχουν ανατεθεί. Για κάθε μια υπάρχει ένα πεδίο που αποτελείται από ένα Submit Button, το όνομα της διαδρομής και τα αντικείμενα μάθησης που την αποτελούν. Δίπλα στο Submit Button υπάρχει η ένδειξη DONE στην περίπτωση που ο μαθητής έχει «ακολουθήσει» τη διαδρομή αυτή σε προηγούμενη επίσκεψή του.

Επιλέγοντας μία από τις διαθέσιμες διαδρομές, ο μαθητής οδηγείται στη σελίδα Route View που φαίνεται στο σχήμα 4.2-12. Εκεί εμφανίζονται στη σειρά όλα τα αντικείμενα της εκπαιδευτικής διαδρομής. Περνώντας το ποντίκι πάνω από κάποιο σύνδεσμο εμφανίζεται η περιγραφή του αντικειμένου από το πεδίο *Description* του πίνακα *Object*. Πατώντας κάποιον από τους συνδέσμους ανοίγει σε ξεχωριστή καρτέλα η αντίστοιχη σελίδα απεικόνισης, που φαίνεται στο σχήμα 4.2-13.


Εκεί ο μαθητής μπορεί να δει την εικόνα και να διαβάσει την πληροφορία που αντιστοιχεί στο αντικείμενο αυτό. Στο κάτω μέρος της σελίδας υπάρχουν σύνδεσμοι που τον οδηγούν στο προηγούμενο και το επόμενο αντικείμενο της διαδρομής.

<b>Κοργιαλένιο Ίδρυμα</b>	Route View
To Νησί	
To Κοργιαλένιο Ίδρυμα	
Επικοινωνία	
<b>User Menu</b>	
Your Profile	
Submit an Article	
Submit a Web Link	
<b>Login</b>	
Login	
<b>e-learning</b>	
Δημιουργία Διαδρομών	
Τεκμηρίωση Αντικειμένων Μάθησης	
Διαγωνίσματα	
Διαδρομές	

<b>1- Object70</b>
<b>2- Object80</b>
<b>3- Object39</b>
<b>4- Object36</b>
<b>5- Object41</b>
<b>6- Object146</b>
<b>7- Object116</b>
<b>8- Object76</b>
<b>9- Object142</b>

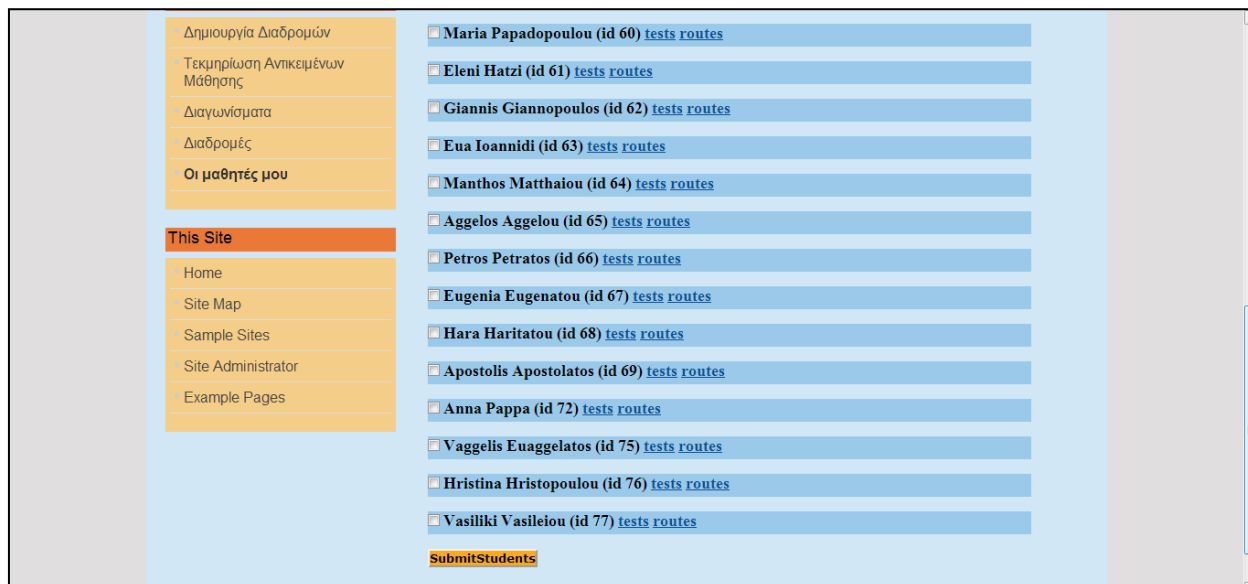
Route View 4.2-12

Login	 <p>Πρόκειται για τον πρώτο θαλασσόμυλο που συναντάμε στα δεξιά μας στη διαδρομή από Αργιστόλι Φανάρι. Το 1835 ο άγγλος Steven ανακάλυψε ότι στην περιοχή αυτή το θαλασσινό νερό απορροφάται με ορμητικότητα μέσα από τα χάσματα της γειτονικής βραχώδους ακτής. Το περίεργο είναι ότι αυτά τα χάσματα δέχονται σταθερά συγκεκριμένη ποσότητα νερού, με αποτέλεσμα η εισροή του νερού σε αυτά να είναι διαρκής. Ο Steven σκέφθηκε να χρησιμοποιήσει την ενέργεια που παράγονταν από την κίνηση του νερού και να κατασκευάσει εκεί ένα θαλασσόμυλο για την παραγωγή αλευριού. Το εσωτερικό των θαλασσόμυλων ως προς την τεχνολογία είναι πανομοιότυπο με εκείνο των ανεμόμυλων και η διαδικασία και λειτουργία της άλεσης ακολουθεί το ίδιο σύστημα και τις ίδιες πρακτικές. Εκεί που διαφέρουν ριζικά είναι στην εξωτερική διάταξη της φτερωτής, αφού αυτή έχει διαμορφωθεί με τέτοιο τρόπο ώστε να κινείται από το νερό. Ο θαλασσόμυλος Steven's τα καταπιά χρόνια έγινε γνωστός ως Μύλος του Ταμπάκη ή Μύλος της Κυράς Βασιλικής. Τα τελευταία χρόνια ο μύλος και η περιοχή γύρω από αυτόν αναπτύχθηκε ριζικά. Λίγο βορειότερα το ίδιο φαινόμενο ανακάλυψαν το 1859 οι αδελφοί Μηλιαρέση όπου κατασκεύασαν εκεί τον ομώνυμο θαλασσόμυλο.</p> <p><a href="#">προηγούμενο αντικείμενο</a> <a href="#">επόμενο αντικείμενο</a></p>
<b>e-learning</b>	
Δημιουργία Διαδρομών	
Τεκμηρίωση Αντικειμένων Μάθησης	
Διαγωνίσματα	
Διαδρομές	
Οι μαθητές μου	
<b>This Site</b>	
Home	
Site Map	
Sample Sites	
Site Administrator	
Example Pages	

Powered by Joomla!

ObjectInfo 4.2-13

#### 4.2.1.5 Οι μαθητές μου



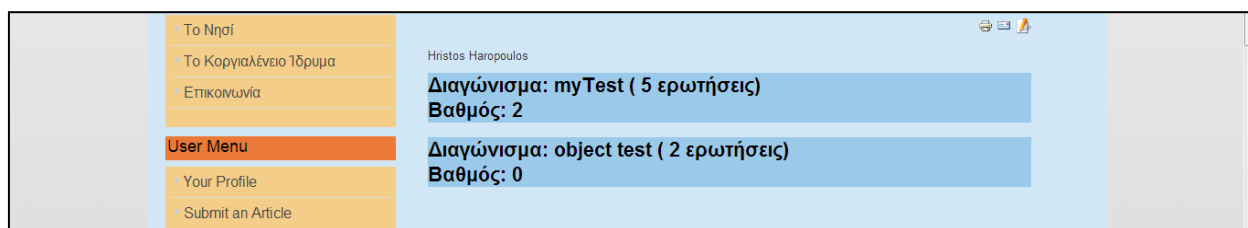
#### SelectStudents 4.2-14

Στη σελίδα αυτή έχουν πρόσβαση οι δάσκαλοι που χρησιμοποιούν το σύστημα. Εδώ μπορούν να ορίσουν τους μαθητές της τάξης τους, δηλαδή αυτούς στους οποίους μπορούν να αναθέτουν διαδρομές και διαγωνίσματα, αλλά και να ελέγχουν αν αυτοί έχουν «ακολουθήσει» τις διαδρομές ή έχουν απαντήσει στα διαγωνίσματα.

Αρχικά είναι επιλεγμένα τα checkboxes που αντιστοιχούν στους μαθητές που ήδη ανήκουν στην τάξη του τρέχοντος χρήστη. Επιλέγοντας τους μαθητές που θέλει και πατώντας το πλήκτρο SubmitStudents, ο δάσκαλος ορίζει τους μαθητές της τάξης του. Οι σύνδεσμοι tests και routes δίπλα σε κάθε όνομα ανοίγουν σε νέα παράθυρα σελίδες της μορφής των σχημάτων 4.2-15 και 4.2-16 αντίστοιχα.

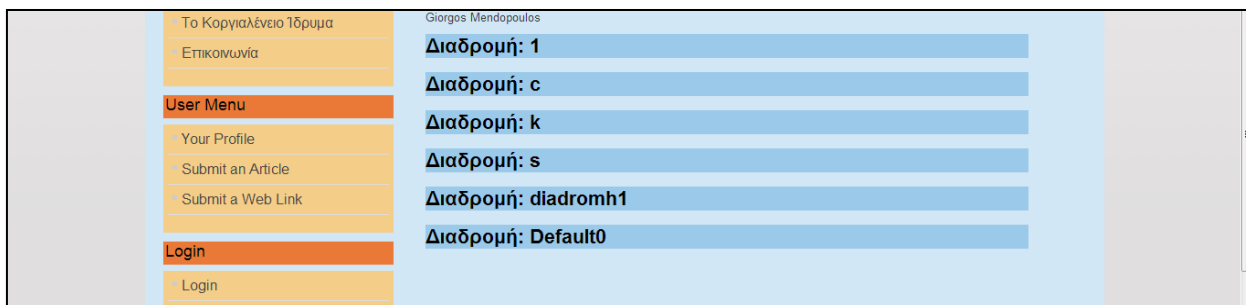
Στη σελίδα Student Tests εμφανίζονται τα διαγωνίσματα που έχουν ανατεθεί στον επιλεγμένο μαθητή καθώς και η αντίστοιχη επίδοσή του, που είναι 0 αν δεν τα έχει απαντήσει.

Στη σελίδα Student Routes εμφανίζονται οι εκπαιδευτικές διαδρομές που του έχουν ανατεθεί, με την ένδειξη DONE αν τις έχει ακολουθήσει.



#### Student Tests 4.2-15

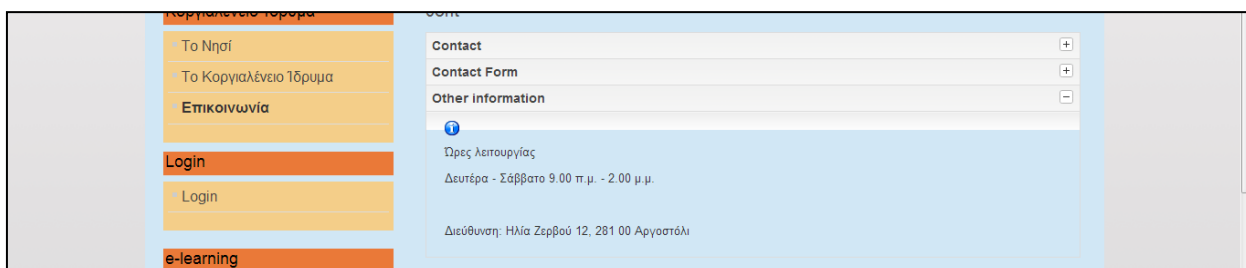




**Student Routes 4.2-16**

## 4.2.2 Γενικές Πληροφορίες

Στο μενού «Κοργιαλένιο Ίδρυμα» αντιστοιχούν 4 σελίδες: η αρχική, η οποία φαίνεται στο σχήμα 4.1-1, η σελίδα «Μια γενική διαδρομή!», που εμφανίζεται σαν επιλογή όταν είναι επιλεγμένος ο σύνδεσμος «Το Κοργιαλένιο Ίδρυμα», μια σελίδα παρόμοια με την αρχική που περιέχει γενικές πληροφορίες για το νησί της Κεφαλονιάς και η σελίδα επικοινωνίας, που περιέχει τους τρόπους επικοινωνίας με το Ίδρυμα και τα ωράρια λειτουργίας του μουσείου, και φαίνεται στο σχήμα 4.2-17.



**Contact 4.2-17**

## 4.2.3 User Menu

Στο μενού αυτό αντιστοιχεί μόνο μια σελίδα, η “Your Profile”, που φαίνεται στο σχήμα 4.2-18. Εκεί ο συνδεδεμένος χρήστης μπορεί να δει τα προσωπικά στοιχεία που έχει εισάγει και να τα αλλάξει ακολουθώντας το σύνδεσμο “Edit Profile”.

## 4.2.4 Login

Στο μενού αυτό αντιστοιχεί η σελίδα Login, που φαίνεται στο σχήμα 4.2-19. Εκεί ο επισκέπτης, ακολουθώντας το σύνδεσμο “Don’t have an account” μπορεί να δημιουργήσει έναν καινούργιο λογαριασμό στο σύστημα. Αν είναι ήδη εγγεγραμμένος μπορεί να συνδεθεί εισάγοντας το User Name και τον κωδικό του ή να ακολουθήσει τους αντίστοιχους συνδέσμους σε περίπτωση που έχει ξεχάσει κάτι

από αυτά. Τέλος, ένας συνδεδεμένος χρήστης μπορεί να επιλέξει το σύνδεσμο Login για να αποσυνδεθεί, πατώντας το πλήκτρο Logout που θα εμφανιστεί σε αυτή την περίπτωση.

**Κοργιαλένιο Ίδρυμα**

- Το Νησί
- Το Κοργιαλένιο Ίδρυμα
- Επικοινωνία

**User Menu**

- Your Profile

**Login**

- Login

**e-learning**

- Δημιουργία Διαδρομών
- Τεκμηρίωση Αντικειμένων Μάθησης
- Διαγωνίσματα
- Διαδρομές
- Οι μαθητές μου

**Profile**

Name: Anna Pappa  
Username: anna  
Registered Date: Tuesday, 09 April 2013  
Last visited date: Thursday, 18 April 2013

**Basic Settings**

Editor: No Information Entered  
Time zone: No Information Entered  
Frontend language: No Information Entered

**User Profile**

Address 1: No Information Entered  
Address 2: No Information Entered  
City: No Information Entered  
Region: No Information Entered  
Country: No Information Entered  
Postal / ZIP Code: No Information Entered  
Phone: No Information Entered  
Web site: No Information Entered  
Favourite Book: No Information Entered  
About Me: No Information Entered  
Date of Birth: Thursday, 07 October 1976

[Edit Profile](#)

**Your Profile 4.2-18**

You are here: [Home](#) > [Login](#)

**Κοργιαλένιο Ίδρυμα**

- Το Νησί
- Το Κοργιαλένιο Ίδρυμα
- Επικοινωνία

**Login**

- Login

**e-learning**

- Δημιουργία Διαδρομών

User Name:

Password:

Remember me

- [Forgot your password?](#)
- [Forgot your username?](#)
- [Don't have an account?](#)

**Login 4.2-19**

## 5 ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

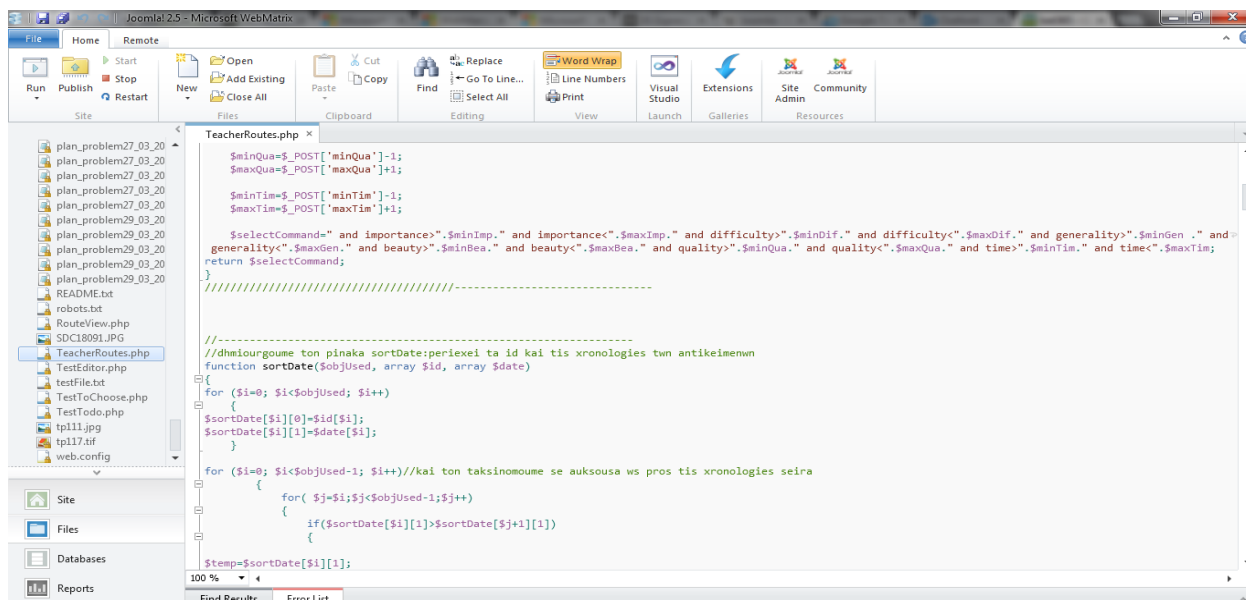
### 5.1 WebMatrix

Το **WebMatrix** είναι ένα ελεύθερο εργαλείο ανάπτυξης διαδικτυακών εφαρμογών της Microsoft. Η έκδοση WebMatrix2, που χρησιμοποιήσαμε για την ανάπτυξη του συστήματος, δίνει στο σχεδιαστή τη δυνατότητα να δημιουργήσει ιστοσελίδες βασιζόμενος είτε σε ενσωματωμένα διαθέσιμα πρότυπα, είτε σε κάποιο από τα δημοφιλή συστήματα διαχείρισης περιεχομένου (Content Management System) ανοιχτού κώδικα, όπως το Drupal και το Joomla. Στην τελευταία περίπτωση, μέσα από το πλαίσιο λειτουργιών του WebMatrix, ο σχεδιαστής μπορεί να τροποποιήσει ως ένα βαθμό το σύστημα διαχείρισης περιεχομένου, ώστε να το προσαρμόσει στις ανάγκες της εφαρμογής του.

Το WebMatrix υποστηρίζει την ανάπτυξη σε διάφορες γλώσσες προγραμματισμού, όπως η PHP και η ASP.NET, ενώ το πακέτο εγκατάστασης του περιλαμβάνει την έκδοση IIS Express 7.5 του εξυπηρετητή ιστού των Windows, τον οποίο και χρησιμοποιήσαμε.

Για την ανάπτυξη του συστήματός μας επιλέξαμε να χρησιμοποιήσουμε το σύστημα διαχείρισης περιεχομένου Joomla!, και πιο συγκεκριμένα την έκδοση 2.5. Για τη διαχείριση των βάσεων δεδομένων επιλέξαμε το σύστημα MySQL. Με την εγκατάσταση του Joomla και την επιλογή του ως CMS για την ανάπτυξη ιστοσελίδων, μαζί με την αρχική ιστοσελίδα δημιουργείται και ο φάκελος στον οποίο αποθηκεύονται τα απαραίτητα αρχεία, καθώς και η βάση δεδομένων με τα δομικά της στοιχεία, όπως θα δούμε στο επόμενο υποκεφάλαιο που αναφέρεται στη δομή του Joomla.

Οι λειτουργίες της εφαρμογής μας αναπτύχθηκαν ως ξεχωριστά αρχεία PHP στο περιβάλλον του WebMatrix, μέσα στο φάκελο της ιστοσελίδας μας, ώστε να μπορούν να κληθούν, όπως θα δούμε, από αυτήν, με τη βοήθεια των δυνατοτήτων που προσφέρει το Joomla.



The screenshot shows the Microsoft WebMatrix interface for Joomla! 2.5. The left sidebar displays a file explorer with a tree view of the Joomla! installation directory, including folders like 'plan\_problem27\_03\_20' and files like 'TeacherRoutes.php'. The main area is a code editor for 'TeacherRoutes.php', containing PHP code for handling routes and sorting data. The code includes variables for min/max quality and time, a function for selecting commands based on various criteria, and a sorting function for a list of objects.

```
TeacherRoutes.php
$minQua=$_POST['minQua']-1;
$maxQua=$_POST['maxQua']+1;

$minTim=$_POST['minTim']-1;
$maxTim=$_POST['maxTim']+1;

$selectCommand=" and importance>". $minImp." and importance<". $maxImp." and difficulty>". $minDif." and difficulty<". $maxDif." and generality>". $minGen ." and
generality<". $maxGen." and beauty>". $minBea." and beauty<". $maxBea." and quality>". $minQua." and quality<". $maxQua." and time>". $minTim." and time<". $maxTim;
return $selectCommand;
}

//-----
//dhmiourgoume ton pinaka sortDate:perixeit ta id kai tis xronologies twm antikeimewn
function sortDate($objUsed, array $id, array $date)
{
    for ($i=0; $i<$objUsed; $i++)
    {
        $sortDate[$i][0]=$id[$i];
        $sortDate[$i][1]=$date[$i];
    }

    for ($i=0; $i<$objUsed-1; $i++)//kai ton taksinomoume se auksousa ws pros tis xronologies seira
    {
        for( $j=$i;$j<$objUsed-1;$j++)
        {
            if($sortDate[$i][1]>$sortDate[$j+1][1])
            {
                $temp=$sortDate[$i][1];
            }
        }
    }
}
```

WebMatrix 5.1-1

## 5.2 JOOMLA!

Τα **Συστήματα Διαχείρισης Περιεχομένου - ΣΔΠ** (Content Management Systems) είναι προγράμματα που επιτρέπουν τη σύγχρονη (online) τροποποίηση των περιεχομένων ενός δικτυακού τόπου. Ο διαχειριστής ενημερώνει online (λειτουργία back-end) το περιεχόμενο του ΣΔΠ που είναι εγκατεστημένο σε κάποιον εξυπηρετητή ιστού και οι αλλαγές γίνονται μέσω του διαδικτύου διαθέσιμες στους επισκέπτες του δικτυακού τόπου (λειτουργία front-end).

Το Joomla είναι ένα ελεύθερο ΣΔΠ ανοιχτού κώδικα που χρησιμοποιείται για τη δημοσίευση περιεχομένου στο διαδίκτυο ή σε τοπικά δίκτυα. Είναι γραμμένο σε γλώσσα PHP και χρησιμοποιεί τεχνικές αντικειμενοστραφούς προγραμματισμού και σχεδιαστικά πρότυπα, ενώ αποθηκεύει δεδομένα σε μια βάση MySQL. Οι σελίδες του είναι δυναμικές και το περιεχόμενό τους ενημερώνεται κάθε φορά που κάποιος επισκέπτης ανοίγει μια σελίδα.

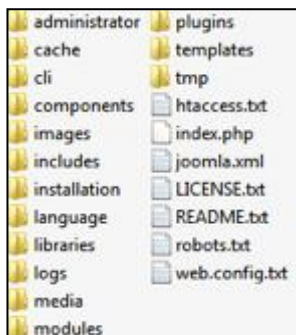
Ένα σημαντικό χαρακτηριστικό του Joomla είναι ότι χρησιμοποιεί **επεκτάσεις** (extensions), δηλαδή κομμάτια λογισμικού που μπορούν να εγκατασταθούν στο σύστημα και να επεκτείνουν τις δυνατότητες που δίνει η αρχική εγκατάσταση. Οι επεκτάσεις αντιστοιχούν σε **λειτουργικές μονάδες** (components), **ενότητες** (modules), **εξαρτήματα** (plugins), **γλώσσες** (languages) και **σχεδιαστικά πρότυπα** (templates), στοιχεία που αναλύουμε παρακάτω.

Λόγω των δυνατοτήτων που προσφέρει κυρίως στον τομέα της διαχείρισης περιεχομένου, το Joomla χρησιμοποιείται ευρέως για την κατασκευή ιστοσελίδων και για εκπαιδευτικούς σκοπούς. Η οργάνωση του περιεχομένου διευκολύνεται σε μεγάλο βαθμό από το γραφικό περιβάλλον χρήστη και η ανανέωσή του χαρακτηρίζεται από ανεξαρτησία τόσο σε επίπεδο περιεχομένου όσο και στο επίπεδο των διαφόρων, αν υπάρχουν, διαχειριστών ή χρηστών. Διευκολύνεται επίσης η διαχείριση των χρηστών και επισκεπτών της ιστοσελίδας και ο καθορισμός των επιπέδων πρόσβασης, ενώ μεγάλη σημασία έχει και η χρήση των επεκτάσεων. Χάρη σε αυτές καθίσταται ιδιαίτερα ευέλικτη η επέκταση των λειτουργιών του συστήματος με την επιλογή των κατάλληλων από τις διαθέσιμες επεκτάσεις, ανάλογα με τις ανάγκες του διαχειριστή. Τέλος, δίνεται στον τελευταίο η δυνατότητα να επέμβει στην εμφάνιση της ιστοσελίδας στο επίπεδο που επιθυμεί, κατ' αρχάς επιλέγοντας από μια μεγάλη ποικιλία το σχεδιαστικό πλαίσιο που θα χρησιμοποιήσει και στη συνέχεια επεμβαίνοντας και στον κώδικα CSS του πλαισίου εφ' όσον το θεωρεί απαραίτητο.

Στην επιλογή του Joomla ως ΣΔΠ για το σύστημά μας, εκτός από τα παραπάνω, σημαντικό ρόλο έπαιξε το γεγονός ότι χρησιμοποιεί για την αποθήκευση δεδομένων βάση MySQL, κάτι που διευκόλυνε τη διαχείρισή της παράλληλα με τη βάση δεδομένων που δημιουργήσαμε για το σύστημα. Επίσης, το Joomla διευκολύνει όπως θα δούμε παρακάτω τη σύνδεση της ιστοσελίδας με αρχεία PHP, κάτι που βοήθησε την εισαγωγή ανεξάρτητων λειτουργιών που αναπτύξαμε για τις ανάγκες της εφαρμογής μας.

Για να κατανοήσουμε τη δομή του Joomla, θα αναφερθούμε στα σημαντικότερα περιεχόμενα του φακέλου της ιστοσελίδας, αλλά και στη βάση δεδομένων που αποθηκεύονται τα δεδομένα της, ενώ στη συνέχεια θα περιγράψουμε αναλυτικά τον τρόπο με τον οποίο αξιοποιήσαμε τις δυνατότητες και τα χαρακτηριστικά στοιχεία του Joomla για την κατασκευή του συστήματος.<sup>[14]</sup>

## 5.2.1 Περιεχόμενα Ιστοσελίδας



**Joomla Directory 5.2-1**

Στο σχήμα 5.2-1 παρουσιάζονται τα περιεχόμενα του φακέλου που δημιουργήθηκε με την εξαγωγή του πακέτου του Joomla.

Ο φάκελος includes περιέχει τα αρχεία εκτέλεσης PHP με τις δομικές λειτουργίες της ιστοσελίδας, που εκτελούνται αυτόματα για να συμπεριληφθεί στην ιστοσελίδα το περιεχόμενο που έχει οριστεί.

Ο φάκελος libraries περιέχει όλες τις βιβλιοθήκες του Joomla, αλλά και τις εξωτερικές που πιθανώς χρησιμοποιεί η ιστοσελίδα για τις διάφορες περιοχές εφαρμογών.

Στο φάκελο components περιέχονται οι **λειτουργικές μονάδες**, δηλαδή αρχεία επεκτάσεων που είναι απαραίτητες για την εκτέλεση των επιμέρους λειτουργιών της ιστοσελίδας, όπως οι λειτουργίες login, register, search κλπ.

Στο φάκελο modules περιέχονται οι **ενότητες** του Joomla. Ανήκουν στην κατηγορία των επεκτάσεων και ουσιαστικά αντιστοιχούν στους χώρους απεικόνισης των λειτουργικών μονάδων μέσα στην ιστοσελίδα.

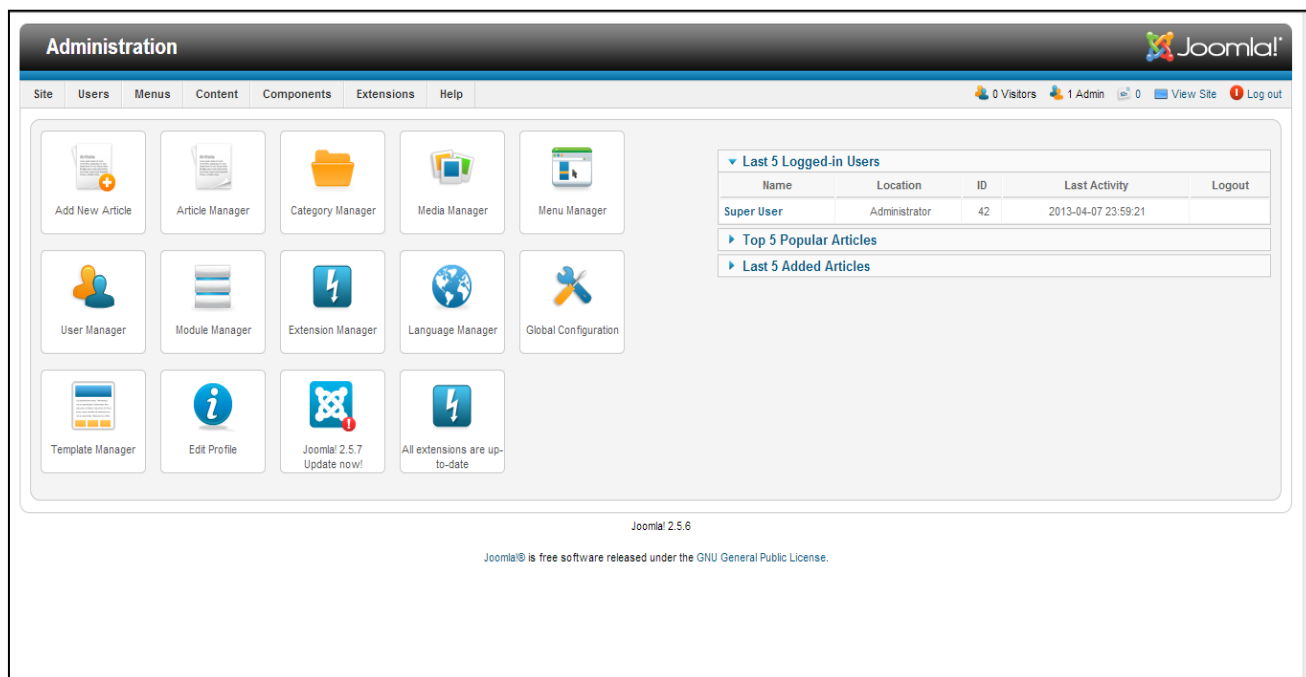
Ο φάκελος language περιέχει τα αρχεία γλωσσών που χρησιμοποιούνται και αφορούν κυρίως τις δυνατότητες μετάφρασης του περιεχομένου της ιστοσελίδας, οι οποίες προσφέρονται με την εγκατάσταση των αντίστοιχων επεκτάσεων.

Ο φάκελος plugins περιέχει τα **εξαρτήματα**, που αποτελούν επεκτάσεις οι οποίες εκτελούνται μαζί με τις εφαρμογές με τις οποίες τις συνδέει ο χρήστης και χειρίζονται διάφορα συμβάντα. Για παράδειγμα, ένα εξάρτημα θα μπορούσε να χρησιμοποιείται κατά την υποβολή ενός άρθρου για να φιλτράρει απαγορευμένες λέξεις.

Ο φάκελος Templates περιέχει τα **σχεδιαστικά πρότυπα**, δηλαδή τους διάφορους τύπους εμφάνισης που μπορεί να έχει ο «σκελετός» της ιστοσελίδας.

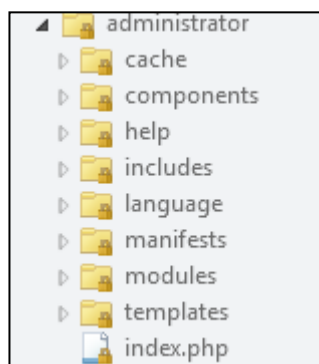
Για το τέλος αφήσαμε το φάκελο administrator, που περιέχει τα απαραίτητα στοιχεία για τη διαχείριση της ιστοσελίδας και εξηγείται αναλυτικότερα:

Η διαχείριση του περιεχομένου μιας ιστοσελίδας του Joomla γίνεται μέσω της διεπαφής **Site Administrator**. Το κομμάτι αυτό μπορεί να θεωρηθεί είτε ως μια ξεχωριστή ιστοσελίδα είτε ως τμήμα της πρώτης ιστοσελίδας, στο οποίο όμως έχει πρόσβαση μόνο κάποιος διαχειριστής (administrator).



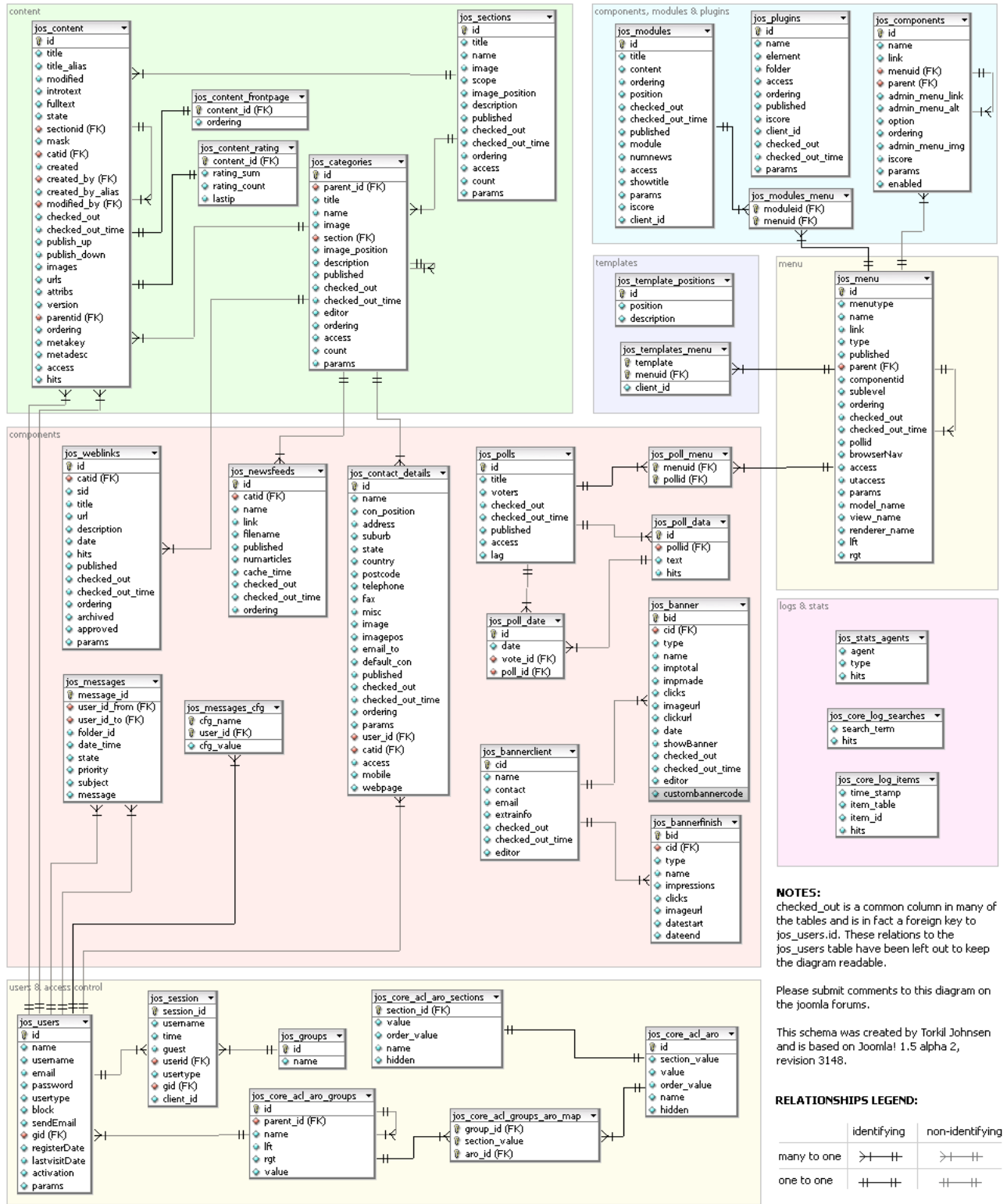
### Site Administration 5.2-2

Ως εκ τούτου, στο φάκελο administrator περιέχονται τα αρχεία που είναι απαραίτητα για τη διεπαφή Site Administrator, δηλαδή για τη διαχείριση της ιστοσελίδας. Σε αυτά συμπεριλαμβάνονται μεταξύ άλλων τα λειτουργικά μέρη (components), οι ενότητες(modules) και τα πρότυπα (templates) που είδαμε προηγουμένως, αφού όπως είπαμε το Site Administration έχει τη μορφή μιας ξεχωριστής ιστοσελίδας.<sup>[14]</sup>



### Joomla\administrator 5.2-3

## 5.2.2 Η Βάση Δεδομένων του Joomla



Joomla Database E-R Diagram 5.2-4

Στο σχήμα 5.2-4 βλέπουμε το διάγραμμα οντοτήτων-συσχετίσεων της βάσης δεδομένων που χρησιμοποιεί μια ιστοσελίδα του Joomla. Στο σύστημά μας η βάση αυτή έχει το όνομα my\_site\_db221. Όπως βλέπουμε, οι πίνακες της βάσης χωρίζονται σε 7 μεγάλες κατηγορίες:

## 1. Content

Οι πίνακες αυτοί περιέχουν δεδομένα που αφορούν τα **άρθρα** (articles) της ιστοσελίδας. Τα άρθρα είναι κομμάτια περιεχομένου, συνήθως κειμένου ή κώδικα HTML, και το κάθε ένα αποτελεί το βασικό περιεχόμενο της σελίδας που το περιέχει. Ανάλογα με τις επιλογές του διαχειριστή μπορεί να ομαδοποιούνται σε **τομείς** (sections), ή **κατηγορίες** (categories) οι οποίες με τη σειρά τους μπορεί να ανήκουν σε ξεχωριστούς τομείς, αλλά μπορεί επίσης και να μην κατηγοριοποιηθούν. Εκτός από την ιεραρχία αυτή, οι σχετικοί πίνακες περιέχουν ρυθμίσεις που αφορούν τις επιλογές εμφάνισης και τη θέση των περιεχομένων τους.

## 2. Menu

Εδώ έχουμε μόνο έναν πίνακα που αφορά τα μενού που περιέχονται στις σελίδες. Περιέχει πληροφορίες για το είδος και τα περιεχόμενα κάθε μενού και συνδέεται με τις διάφορες επεκτάσεις που μπορεί να χρησιμοποιεί και τα σχεδιαστικά πρότυπα που το περιέχουν.

## 3. Templates

Ο πίνακας templates\_positions περιέχει τις θέσεις στις οποίες χωρίζεται η οθόνη μιας σελίδας, δηλαδή τα πιθανά σημεία εισαγωγής διαφορετικών περιεχομένων, όταν η ιστοσελίδα χρησιμοποιεί το αντίστοιχο σχεδιαστικό πρότυπο, ενώ ο πίνακας templates\_menu συνδέεται με τα μενού που αυτό περιέχει.

## 4. Components, modules and plugins

Τα δεδομένα αυτά αφορούν τις επεκτάσεις που αναφέραμε παραπάνω και τις πιθανές συνδέσεις τους με κάποια μενού. Κάθε μενού είναι ένα σύνολο από menu items, δηλαδή στοιχεία που όταν επιλεγθούν οδηγούν σε κάποια άλλη σελίδα ή εκτελούν μια λειτουργία, οπότε είναι πιθανό να χρησιμοποιούν κάποια επέκταση.

## 5. Components

Οι πίνακες αυτοί αντιστοιχούν στις ξεχωριστές λειτουργικές μονάδες και μπορεί να συνδέονται με πλήθος άλλων πινάκων, ανάλογα με τις λειτουργίες που αυτές εξυπηρετούν.

## 6. Logs and stats

Εδώ περιέχονται πληροφορίες για τα αρχεία καταγραφής (log files) που δημιουργούνται κατά την εκτέλεση.

## 7. User and access control

Οι πίνακες αυτοί είναι οι μόνοι που χρησιμοποιήσαμε άμεσα, από τον κώδικα PHP και όχι μέσω της διεπαφής του Site Administrator. Περιέχουν πληροφορίες που αφορούν τους χρήστες, όπως τα



προσωπικά τους στοιχεία, τις κατηγορίες τους και τα επίπεδα πρόσβασης, καθώς και πληροφορίες για το υλικό που πιθανώς έχουν δημοσιεύσει στην ιστοσελίδα. [14]

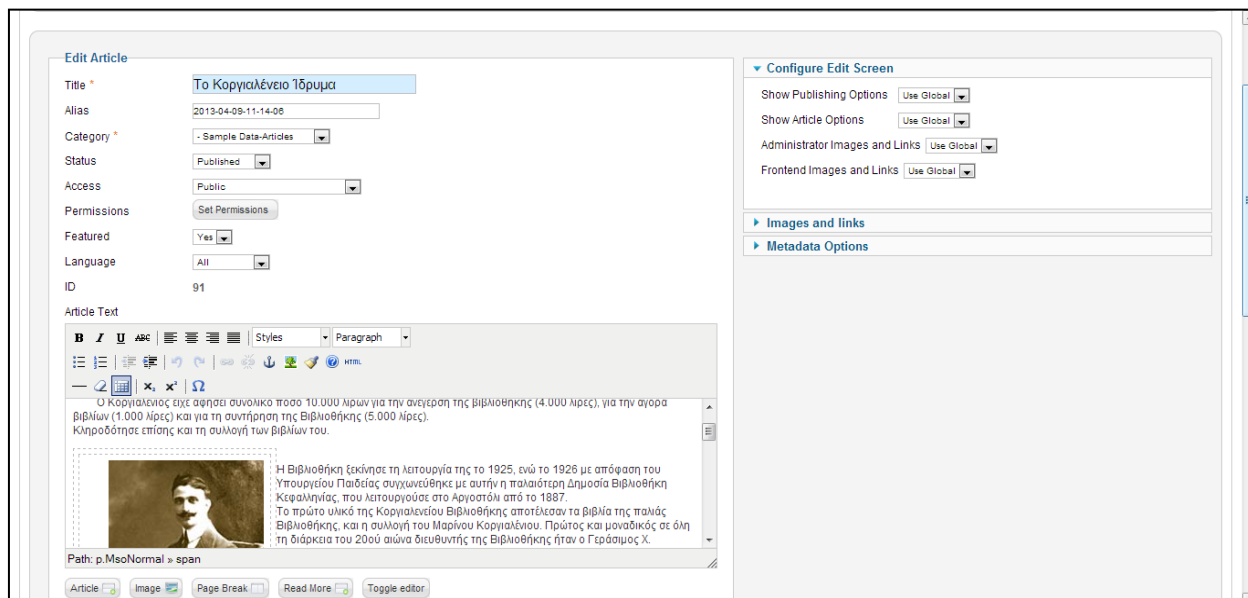
## 5.2.3 Articles

Article Manager: Featured Articles										
Articles   Categories   <b>Featured Articles</b>										
Filter	Search	Clear	- Select Status -		- Select Access -		- Select Language -			
<input type="checkbox"/>	Title	Status	Category	Ordering	Access	Created by	Date	Hits	Language	ID
<input type="checkbox"/>	Assign Routes to Students (Alias: assign-routes-to-students)	✓	Teachers	7	Public	Super User	2013-03-27	122	All	85
<input type="checkbox"/>	Assign Tests to Students (Alias: assign-tests-to-students)	✓	Teachers	6	Public	Super User	2013-04-05	43	All	89
<input type="checkbox"/>	Contacts (Alias: contact)	✓	Components	21	Public	Super User	2011-01-01	11	All	9
<input type="checkbox"/>	Course (Alias: course)	✓	E-learning Articles	22	Registered	Super User	2013-03-27	157	All	86
<input type="checkbox"/>	Expert Changes (Alias: newart1)	✓	Experts	23	ExpertLevel	Super User	2012-12-03	326	All	72
<input type="checkbox"/>	Expert Changes 2 (Alias: expert-changes-2)	✓	Experts	15	Public	Super User	2013-02-22	17	All	74
<input type="checkbox"/>	Expert Changes 3 (Alias: newart3)	✓	Experts	20	Public	Super User	2013-02-22	3	All	75
<input type="checkbox"/>	Expert Changes 4 (Alias: expert-changes-4)	✓	Experts	19	Public	Super User	2013-02-22	3	All	76
<input type="checkbox"/>	Expert Changes 5 (Alias: expert-changes-5)	✓	Experts	18	Public	Super User	2013-02-22	4	All	77
<input type="checkbox"/>	Το Κοργιάλειο Ίδρυμα (Alias: 2013-04-09-11-14-06)	✓	Sample Data-Articles	4	Public	Super User	2013-04-09	83	All	91
<input type="checkbox"/>	Τα τεστ μου (Alias: choose)	✓	Tests for students	9	Public	Super User	2013-03-14	102	All	82
<input type="checkbox"/>	Νέο Διαγώνισμα (Alias: test-editor)	✓	Tests for students	10	Public	Super User	2013-03-14	111	All	81
<input type="checkbox"/>	Κεφαλοιά (Alias: 2013-04-09-11-06-29)	✓	Sample Data-Articles	5	Public	Super User	2013-04-09	12	All	90
<input type="checkbox"/>	Διαδρομές Μαθητών (Alias: 2013-04-15-22-40-45)	✓	Teachers	1	Public	Super User	2013-04-15	19	All	94
<input type="checkbox"/>	Διαγώνισμα Μαθητών (Alias: 2013-04-15-22-33-39)	✓	Teachers	2	Public	Super User	2013-04-15	27	All	93
<input type="checkbox"/>	TestToDo (Alias: test-exam)	✓	E-learning Articles	8	Public	Super User	2013-03-14	77	All	83
<input type="checkbox"/>	Select Students (Alias: select-students)	✓	Teachers	3	Public	Super User	2013-04-09	40	All	92
<input type="checkbox"/>	Route View (Alias: route-view)	✓	E-learning Articles	13	Public	Super User	2013-03-27	135	All	87
<input type="checkbox"/>	Object Info (Alias: object-info)	✓	Sample Data-Articles	11	Public	Super User	2013-03-27	349	All	88
<input type="checkbox"/>	Login Module (Alias: login-module)	✓	User Modules	12	Public	Super User	2011-01-01	5	All	28
<input type="checkbox"/>	Learning routes (Alias: learning-routes)	✓	Teachers	14	Public	Super User	2013-03-21	726	All	84
<input type="checkbox"/>	Expert Changes 7 (Alias: expert-changes-7)	✓	Experts	16	Public	Super User	2013-02-22	6	All	79
<input type="checkbox"/>	Expert Changes 6 (Alias: expert-changes-6)	✓	Experts	17	Public	Super User	2013-02-22	6	All	78

Featured Articles 5.2-5

Το σύστημά μας χρησιμοποιεί 23 άρθρα, τα οποία φαίνονται στο σχήμα 5.2-5. Από αυτά, τα 2 περιέχουν προκαθορισμένες λειτουργικές μονάδες του Joomla: Το Contacts είναι μια σελίδα που παρουσιάζει τους τρόπους επικοινωνίας με το Κοργιαλένιο Ίδρυμα, με βάση ένα πρότυπο πλαίσιο, ενώ το Login επιτελεί τις λειτουργίες σύνδεσης και εγγραφής των χρηστών.

Τα υπόλοιπα σχεδιάστηκαν για τους σκοπούς του συστήματος, με το σχεδιαστή άρθρων του Joomla που φαίνεται στο σχήμα 5.2-6. Από αυτά, τα 2 που αναφέρουν πληροφορίες για το ίδρυμα και το νησί περιέχουν μόνο κείμενο και εικόνες, ενώ όλα τα υπόλοιπα έχουν λειτουργικό χαρακτήρα και συνδέονται με αρχεία PHP, με τρόπο που θα δούμε παρακάτω.



**Edit Article 5.2-6**

Όπως φαίνεται στην εικόνα, ο σχεδιαστής άρθρων δίνει τη δυνατότητα του καθορισμού του περιεχομένου του άρθρου μέσα από τον editor που περιλαμβάνει, ενώ περιέχει διάφορα πεδία για ρυθμίσεις που αφορούν στοιχεία όπως η κατηγορία στην οποία ανήκει, το επίπεδο πρόσβασης και οι επιλογές εμφάνισης.

Εδώ αξίζει να σημειωθεί ότι όσον αφορά στα άρθρα που συνδέονται με τα αρχεία PHP, η μορφή τους καθορίζεται σε μεγάλο βαθμό από τον κώδικα HTML που περιέχουν τα αρχεία αυτά. Για το λόγο αυτό οι παρεμβάσεις μας μέσω του γραφικού περιβάλλοντος χρήστη του Joomla στη συνολική εμφάνιση της ιστοσελίδας ήταν περιορισμένες και αφορούσαν το σχεδιαστικό πλαίσιο που επιλέξαμε και τη γενική μορφή της.

Ως σχεδιαστικό πλαίσιο επιλέξαμε το Beez\_20-Default Template, το οποίο υπήρχε ως διαθέσιμη επιλογή χωρίς την εγκατάσταση κάποιας επέκτασης. Τις περαιτέρω αλλαγές τις πραγματοποιήσαμε μέσα από το CSS αρχείο personal.css του φακέλου του αντίστοιχου σχεδιαστικού πλαισίου και αφορούσαν τροποποιήσεις στα χρώματα, τη γραμματοσειρά και τις εικόνες που εμφανίζονται στο γενικό φόντο της ιστοσελίδας.

### 5.2.3.1 Categories

Uncategorised (Alias: uncategorised)	✓	•	1	Public	All	9
E-learning Articles (Alias: learning)	✓	•	2	Public	All	78
Tests for students (Alias: tests-for-students)	✓	•	1	Public	All	79
Teachers (Alias: teachers)	✓	•	2	TeacherLevel	All	80
Students (Alias: students)	✓	•	3	StudentLevel	All	81
Experts (Alias: experts)	✓	•	4	ExpertLevel	All	82

#### Categories 5.2-7

Για τα άρθρα που κατασκευάσαμε δημιουργήσαμε την κατηγορία e-learning Articles, η οποία χωρίζεται σε 3 υποκατηγορίες: Teachers, Students και Experts. Στις υποκατηγορίες αυτές ορίσαμε τα επίπεδα πρόσβασης για τους χρήστες που τις χρησιμοποιούν, με τρόπο που περιγράφεται στο αντίστοιχο υποκεφάλαιο. Οι ρυθμίσεις αυτές κληρονομούνται στα άρθρα που ανήκουν στην κάθε μία.

Τα άρθρα που περιέχουν γενικές πληροφορίες και στα οποία έχουν πρόσβαση όλοι οι επισκέπτες εισήχθησαν κατευθείαν στη γενική κατηγορία E-learning Articles.

### 5.2.3.2 Σύνδεση με PHP files

**Edit Article**

Title \*

Alias

Category \*

Status

Access

Permissions

Featured

Language

ID

Article Text

**B I U ABC** | | Styles Paragraph

```
<?php
include "AssignRoutesToStudents.php";

$routeId = $_GET["idValue"];
StudentSelection($routeId);
?>
```

Path:

javascript:void(0);

**Configure Edit Screen**

Show Publishing Options

Show Article Options

Administrator Images and Links

Frontend Images and Links

[Images and links](#)

[Metadata Options](#)

#### Edit Article 5.2-8

Για τη σύνδεση των άρθρων με τα αντίστοιχα αρχεία PHP που περιέχουν τις λειτουργίες της εφαρμογής, χρησιμοποιήθηκε η επέκταση **DirectPHP**, την οποία εγκαταστήσαμε από τη σελίδα <http://extensions.joomla.org/extensions>.

Η επέκταση αυτή δίνει τη δυνατότητα στο χρήστη να ενσωματώσει κώδικα PHP στον editor του σχεδιαστή άρθρων. Ο κώδικας που περικλείεται στα αναγνωριστικά «<? ?>» της PHP εκτελείται κατά την κλήση της σελίδας που περιέχει το άρθρο.

Για να συμπεριλάβουμε στα άρθρα τον κώδικα που υπήρχε στα αρχεία PHP που αναπτύξαμε ανεξάρτητα, χρησιμοποιήσαμε την εντολή **include “file.php”**; της PHP. Με την εντολή αυτή, ο κώδικας που περιέχεται στο αρχείο file.php συμπεριλαμβάνεται στο αρχείο που περιέχει την εντολή, δίνοντάς του πρόσβαση σε όλες τις κλάσεις και τις συναρτήσεις του.

Με τον τρόπο αυτό, έχοντας ορίσει στα αρχεία PHP την κύρια («main») συνάρτηση που εκτελεί το σύνολο των λειτουργιών του αρχείου, πρέπει απλώς να καλέσουμε τη συνάρτηση αυτή. Για παράδειγμα, στο σχήμα 5.2-6 φαίνεται η σελίδα της διαχείρισης του άρθρου «Assign Routes to Students». Με την εντολή *include “AssignRoutesToStudents.php”* συμπεριλαμβάνεται ο κώδικας του αρχείου «AssignRoutesToStudents.php» που βρίσκεται στο φάκελο της ιστοσελίδας, ενώ με την εντολή *StudentSelection()*; εκτελείται η ομώνυμη συνάρτηση που έχει οριστεί σε αυτό, η οποία εκτελεί όλες τις ζητούμενες λειτουργίες του αρχείου.

Αν για να εκτελεστεί η συνάρτηση χρειάζεται κάποιο επιπλέον όρισμα, για παράδειγμα το `userId` του τρέχοντος χρήστη ή κάποιο `argument` που περνά μέσω του `url` από την προηγούμενη σελίδα, όπως συμβαίνει στην παραπάνω περίπτωση αλλά και σε άλλες που θα δούμε κατά την ανάλυση των αρχείων PHP, στον κώδικα PHP που γράφουμε στο σχεδιαστή άρθρων προσθέτουμε τις ανάλογες εντολές.

Στην περίπτωση που θέλουμε να πάρουμε μια τιμή που έχει περαστεί μέσω του `url` από την προηγούμενη σελίδα, αυτό γίνεται με τη μέθοδο `$_GET[‘value’]`, όπου `value` η παράμετρος του `url` που περιέχει την τιμή που θέλουμε.

Στην περίπτωση που θέλουμε κάποιο στοιχείο του χρήστη που είναι συνδεδεμένος και εκτελεί τις λειτουργίες, χρησιμοποιούμε τη μέθοδο `getUser()` από τη βιβλιοθήκη `JFactory` που είναι εγκατεστημένη στο Joomla. Για παράδειγμα, αν θέλουμε το `userId` του χρήστη εκτελούμε τις εντολές:

```
$user = & JFactory::getUser();
```

```
$uid=$user->id;
```

Με τον τρόπο αυτό η μεταβλητή `$uid` αποκτά το `id` του τρέχοντος χρήστη.

## 5.2.4 Menus

The screenshot shows a user profile page. On the left, there is a vertical navigation menu with several categories: 'Κοργιαλένιο Ίδρυμα', 'User Menu', 'Login', and 'e-learning'. Under 'User Menu', there is a link for 'Your Profile'. The main content area on the right is divided into three sections: 'Profile', 'Basic Settings', and 'User Profile'. The 'Profile' section shows the user's name as Anna Pappa, username as 'anna', registered date as Tuesday, 09 April 2013, and last visited date as Saturday, 27 April 2013. The 'Basic Settings' section shows 'Editor', 'Time zone', and 'Frontend language' all set to 'No Information Entered'. The 'User Profile' section shows various fields like 'Address 1', 'City', 'Region', etc., all set to 'No Information Entered', except for 'Date of Birth' which is Thursday, 07 October 1976. There is an 'Edit Profile' link at the bottom of the profile section.

**Menus 5.2-9**

Τα ξεχωριστά μενού που χρησιμοποιούμε στην ιστοσελίδα φαίνονται στα αριστερά των σελίδων. Τα 3 από αυτά εμφανίζονται κάθε στιγμή, ενώ το User Menu, που οδηγεί στη σελίδα διαχείρισης προφίλ χρήστη, εμφανίζεται όταν έχει συνδεθεί στο σύστημα κάποιος εγγεγραμμένος χρήστης.

Τα μενού διαχειρίζονται μέσω της σελίδας διαχείρισης μενού (Menu Manager, σχήμα 5.2-1) του Site Administrator. Εκεί αντιστοιχίζονται σε αυτά τα **menu items**, που έχουν το ρόλο των συνδέσμων για τις σελίδες στις οποίες οδηγούν, καθώς και τα **modules** που καθορίζουν τη θέση των μενού στις σελίδες.

The screenshot shows the 'Menu Manager: Menus' interface. At the top, there are navigation tabs for 'Menus' and 'Menu Items'. Below the tabs is a table listing various menu items. The table has columns for 'Title', 'Number of Menu Items' (subdivided into '# Published', '# Unpublished', and '# Trashed'), 'Modules Linked to the Menu', and 'ID'. The table contains the following data:

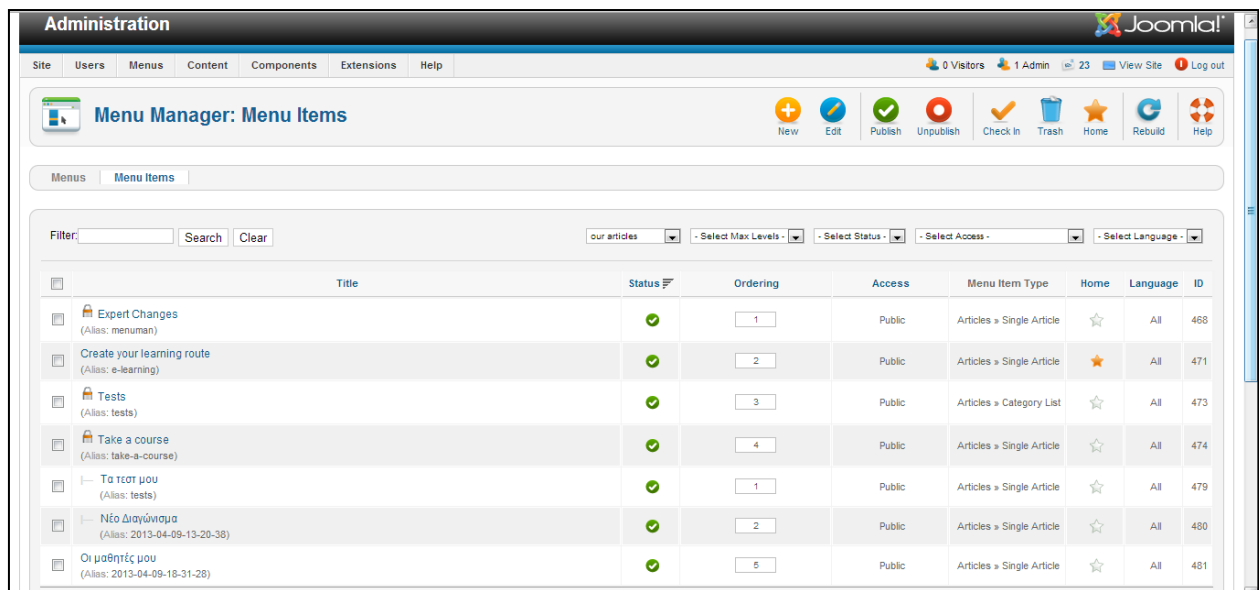
Title	Number of Menu Items			Modules Linked to the Menu	ID
	# Published	# Unpublished	# Trashed		
User Menu (Menu type usermenu)	3	0	0	• User Menu (Registered in position-7)	2
Top (Menu type top)	3	0	0	• Top (Public in position-1)	3
About Kefalonia (Menu type aboutjoomla)	3	0	0	• Extensions (Public in position-7) • About Kefalonia and the foundation (Public in position-7) • Top Menu (Atomic Template) (Public in atomic-topmenu)	4
Australian Parks (Menu type parks)	7	0	0	• Australian Parks (Public in position-5)	5
Main Menu (Menu type mainmenu)	10	0	2	• Main Menu (Public in position-7) • Site Map (Public in sitemapload) • This Site (Public in position-7) • Menu Example (Public in position-7)	6
Fruit Shop (Menu type fruitshop)	8	0	0	• Fruit Shop (Public in position-5)	7

**Menu Manager 5.2-10**

### 5.2.4.1 Menu Items

Τα μενού είναι σύνολα από menu items. Κάθε menu item καθορίζει ουσιαστικά το url της σελίδας στην οποία θα οδηγήσει η επιλογή του, καθώς και ρυθμίσεις που αφορούν την εμφάνιση αυτής της σελίδας.

Υπάρχουν διάφοροι τύποι menu items, ανάλογα με το είδος της σελίδας στην οποία οδηγούν. Για παράδειγμα το menu item **Login** του ομώνυμου μενού είναι τύπου **Login Form** και οδηγεί στη σελίδα υποβολής φόρμας εισόδου στο σύστημα. Ο τύπος των menu items που δημιουργήσαμε για την εφαρμογή είναι **Single Article**, αφού οδηγούν στα άρθρα που χρησιμοποιεί.

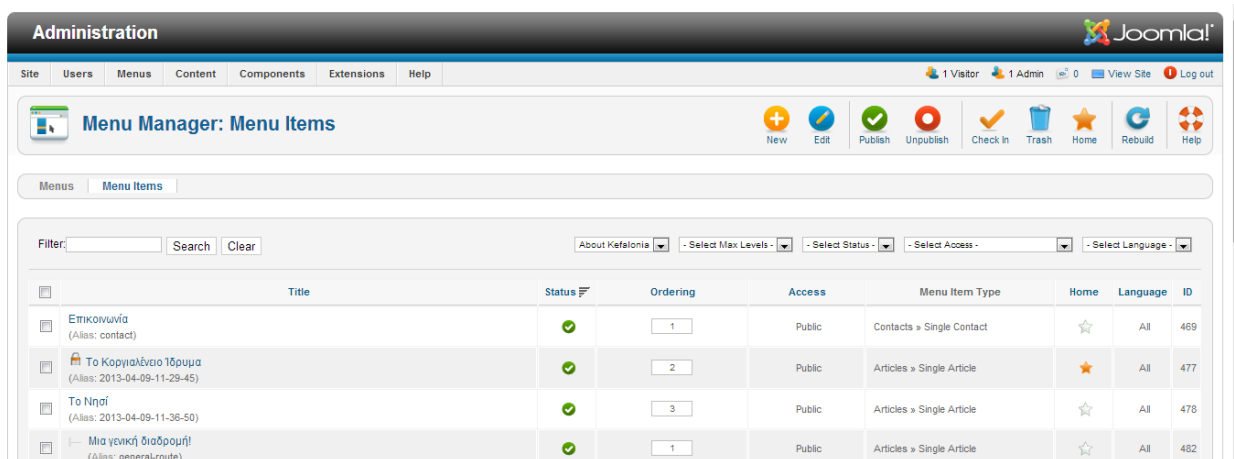


	Title	Status	Ordering	Access	Menu Item Type	Home	Language	ID
<input type="checkbox"/>	Expert Changes (Alias: menuman)	<input checked="" type="checkbox"/>	1	Public	Articles » Single Article	<input type="checkbox"/>	All	468
<input type="checkbox"/>	Create your learning route (Alias: e-learning)	<input checked="" type="checkbox"/>	2	Public	Articles » Single Article	<input checked="" type="checkbox"/>	All	471
<input type="checkbox"/>	Tests (Alias: tests)	<input checked="" type="checkbox"/>	3	Public	Articles » Category List	<input type="checkbox"/>	All	473
<input type="checkbox"/>	Take a course (Alias: take-a-course)	<input checked="" type="checkbox"/>	4	Public	Articles » Single Article	<input type="checkbox"/>	All	474
<input type="checkbox"/>	Τα τεστ μου (Alias: tests)	<input checked="" type="checkbox"/>	1	Public	Articles » Single Article	<input type="checkbox"/>	All	479
<input type="checkbox"/>	Νέο Διαγώνισμα (Alias: 2013-04-09-13-20-38)	<input checked="" type="checkbox"/>	2	Public	Articles » Single Article	<input type="checkbox"/>	All	480
<input type="checkbox"/>	Οι μαθητές μου (Alias: 2013-04-09-18-31-28)	<input checked="" type="checkbox"/>	5	Public	Articles » Single Article	<input type="checkbox"/>	All	481

#### our articles' items 5.2-11

Τα menu items που αντιστοιχούν στο μενού **our articles** (e-learning στη σελίδα) φαίνονται στο σχήμα 5.2-9. Το “Τεκμηρίωση Αντικειμένων Μάθησης” οδηγεί στη σελίδα όπου οι τεκμηριωτές μπορούν να δώσουν τις τιμές των χαρακτηριστικών των αντικειμένων μάθησης. Το “Δημιουργία Διαδρομών” οδηγεί σε αυτήν όπου οι δάσκαλοι μπορούν να δημιουργήσουν μια καινούργια διαδρομή. Το “Διαδρομές” οδηγεί στη σελίδα όπου οι μαθητές βλέπουν τις διαθέσιμες διαδρομές που τους έχουν αντιστοιχίσει για να επιλέξουν αυτή που θα ακολουθήσουν και το “Οι μαθητές μου” σε αυτήν όπου οι δάσκαλοι δηλώνουν τους μαθητές που ανήκουν στην τάξη τους και βλέπουν τις διαδρομές και τα διαγωνίσματά τους. Το «Διαγωνίσματα» οδηγεί τους δασκάλους στη σελίδα που δημιουργούν διαγωνίσματα πολλαπλής επιλογής και τους μαθητές σε αυτήν που βλέπουν τα διαγωνίσματα που τους έχουν ανατεθεί και μπορούν να τα απαντήσουν.

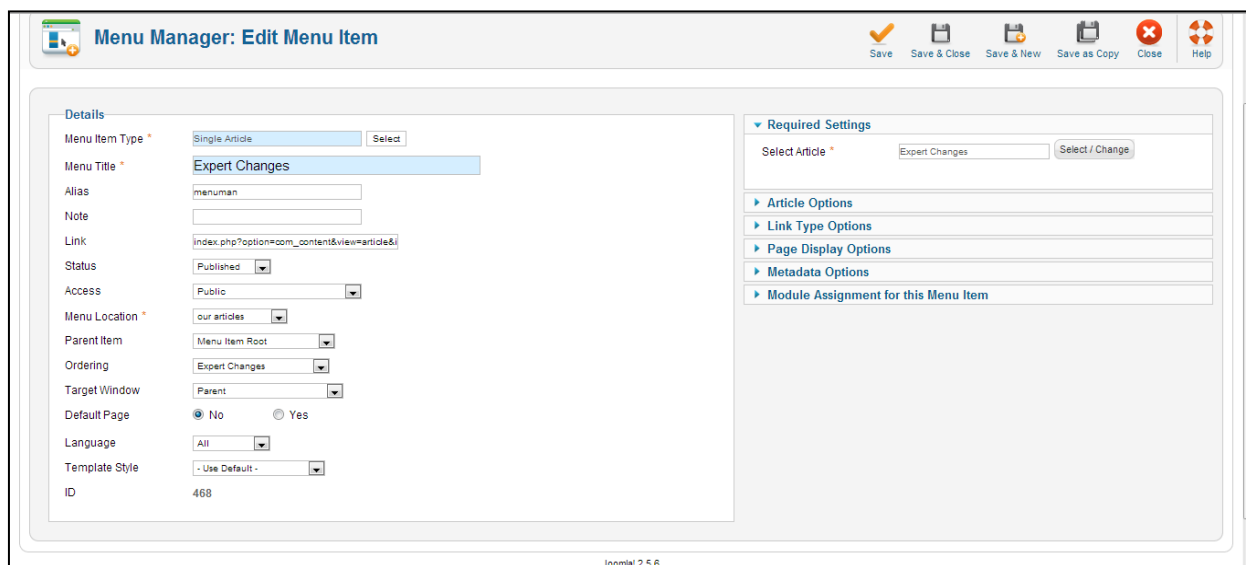
Στο σχήμα 5.2-10 φαίνονται τα menu items του μενού About Kefalonia. Το “Επικοινωνία” οδηγεί στη σελίδα επικοινωνίας, το «Το Κοργιαλένιο Ίδρυμα» στη σελίδα με τις πληροφορίες για το ίδρυμα και το «Το Νησί» σε αυτή με τις πληροφορίες για την Κεφαλονιά.



#### About Kefalonia items 5.2-12

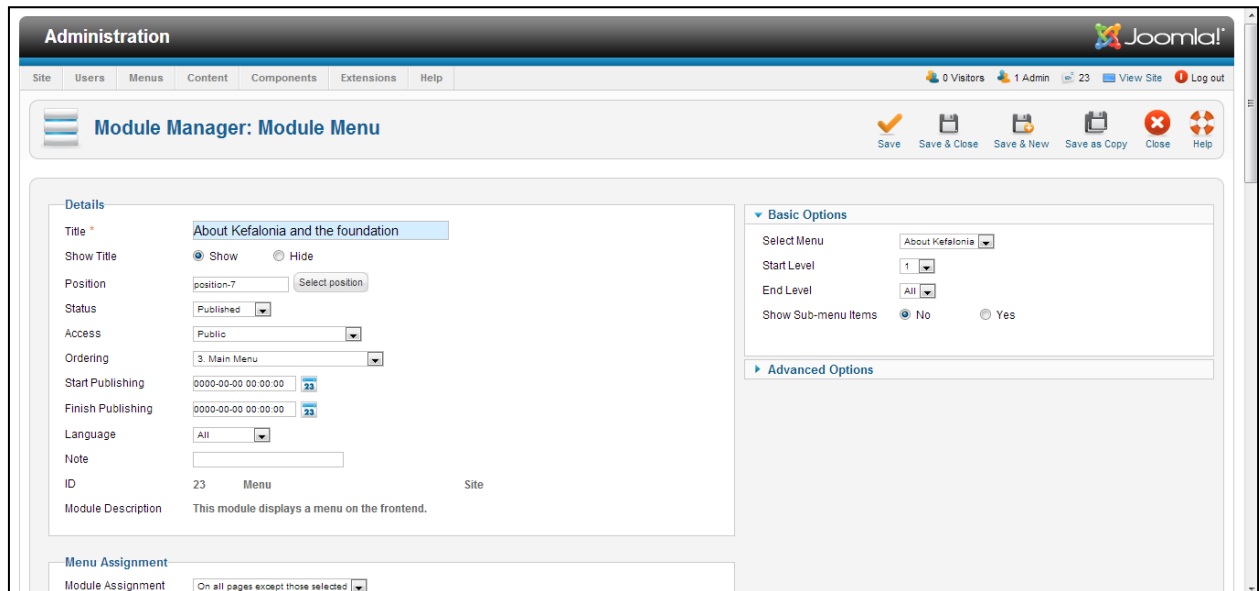
Το menu item “Your Profile” του μενού “User Profile” οδηγεί όπως είπαμε στη σελίδα διαχείρισης προφίλ του χρήστη.

Η διαχείριση των διαφόρων menu items γίνεται στις σελίδες Edit Menu Item, όπως αυτή του παρακάτω σχήματος. Εκεί καθορίζεται ο τύπος τους, το μενού που τα περιέχει και η σειρά με την οποία εμφανίζονται σε αυτό, καθώς και η σελίδα στην οποία οδηγεί η επιλογή τους (Select Article).



#### Edit Menu Item 5.2-13

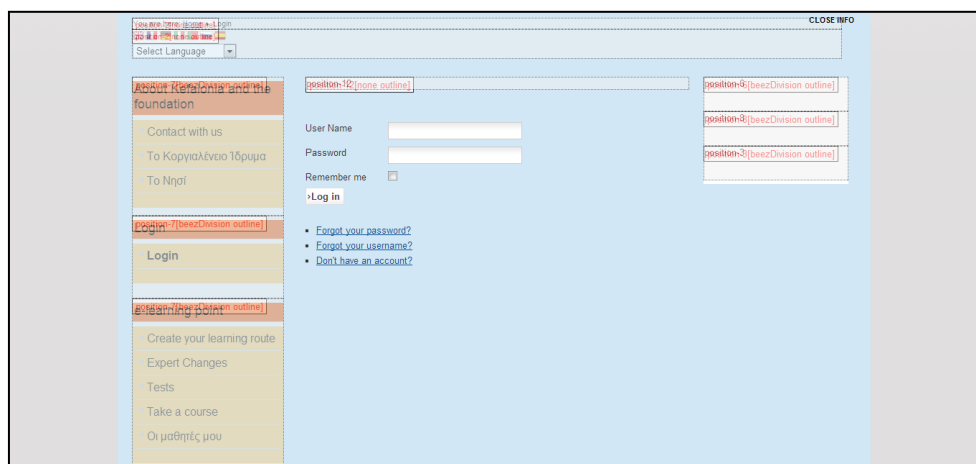
### 5.2.4.2 Ενότητες (Modules)



Module Manager 5.2-14

Όπως έχουμε αναφέρει, οι ενότητες έχουν το ρόλο του «κουτιού» μέσα στο οποίο θα εμφανίζεται μια λειτουργική μονάδα μέσα στη σελίδα. Κάθε ενότητα διαμορφώνεται στη σελίδα Module Menu, όπου ορίζεται η θέση της, η σειρά εμφάνισής της στην περίπτωση που έχουμε παραπάνω από μια σε κάποια θέση, η λειτουργική μονάδα στην οποία αντιστοιχίζεται και άλλες επιλογές που αφορούν την εμφάνισή της.

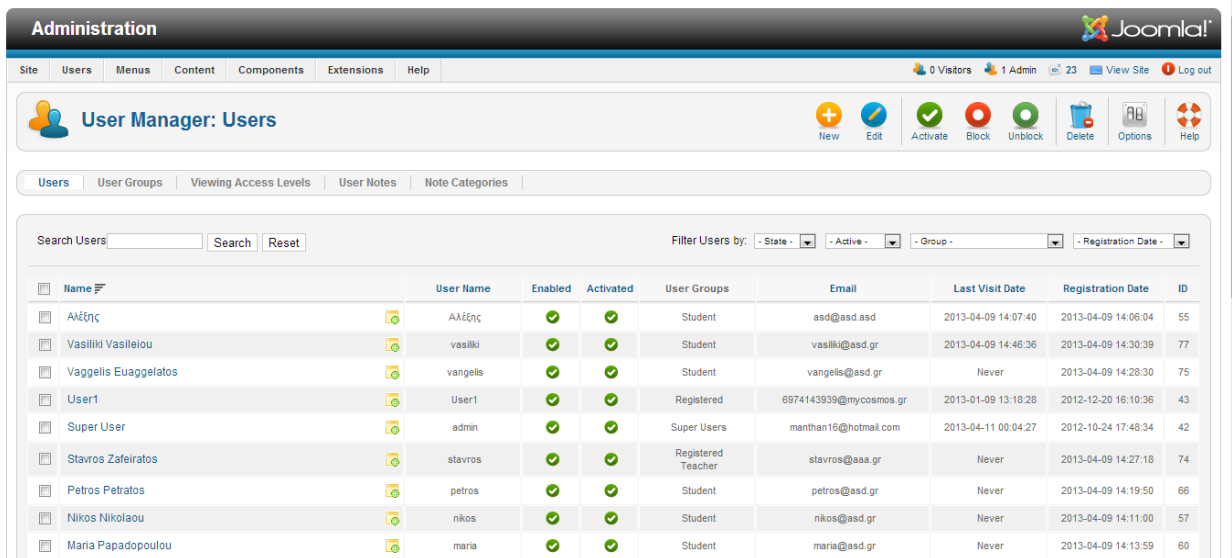
Τις ενότητες που χρησιμοποιήσαμε για την εμφάνιση των μενού τις τοποθετήσαμε στη θέση 7 του χώρου της σελίδας. Στο σχεδιαστικό πλαίσιο που επιλέξαμε, αυτή η θέση αντιστοιχεί στα αριστερά της σελίδας, όπως φαίνεται στο σχήμα, ενώ αντίστοιχα καθορίσαμε και τη σειρά εμφάνισής τους μέσω των Module Menus.



Position-7 5.2-15



## 5.2.5 Χρήστες-Επίπεδα Πρόσβασης



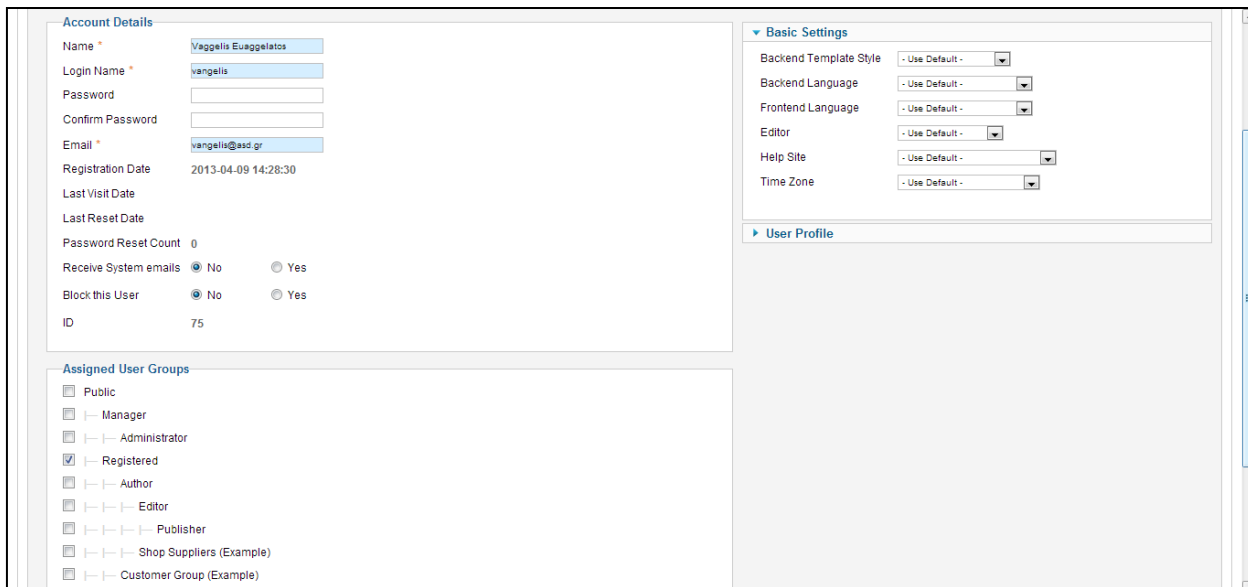
The screenshot shows the Joomla! Administration interface for User Manager. At the top, there are navigation tabs for Site, Users, Menus, Content, Components, Extensions, and Help. The main header displays 'User Manager: Users' with various action icons like New, Edit, Activate, Block, Unblock, Delete, Options, and Help. Below the header, there are filters for State, Active, Group, and Registration Date. A search bar is also present. The main content area is a table listing users with columns for Name, User Name, Enabled, Activated, User Groups, Email, Last Visit Date, Registration Date, and ID.

<input type="checkbox"/>	Name	User Name	Enabled	Activated	User Groups	Email	Last Visit Date	Registration Date	ID
<input type="checkbox"/>	Αλέξης	alexis	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Student	asd@asd.asd	2013-04-09 14:07:40	2013-04-09 14:06:04	55
<input type="checkbox"/>	Vasiliki Vasileiou	vasiliki	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Student	vasiliki@asd.gr	2013-04-09 14:46:36	2013-04-09 14:30:39	77
<input type="checkbox"/>	Vaggelis Euaggelatos	vangelis	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Student	vangelis@asd.gr	Never	2013-04-09 14:28:30	75
<input type="checkbox"/>	User1	User1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Registered	6974143939@mycosmos.gr	2013-01-09 13:18:28	2012-12-20 16:10:36	43
<input type="checkbox"/>	Super User	admin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Super Users	manthan16@hotmail.com	2013-04-11 00:04:27	2012-10-24 17:48:34	42
<input type="checkbox"/>	Stavros Zafeiratos	stavros	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Registered Teacher	stavros@aaa.gr	Never	2013-04-09 14:27:18	74
<input type="checkbox"/>	Petros Petratos	petros	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Student	petros@asd.gr	Never	2013-04-09 14:19:50	66
<input type="checkbox"/>	Nikos Nikolaou	nikos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Student	nikos@asd.gr	Never	2013-04-09 14:11:00	57
<input type="checkbox"/>	Maria Papadopoulou	maria	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Student	maria@asd.gr	Never	2013-04-09 14:13:59	60

User Manager 5.2-16

Η διαχείριση των εγγεγραμμένων χρηστών γίνεται από τον διαχειριστή μέσω της σελίδας User Manager του Site Administrator, που φαίνεται στο σχήμα 5.2-14.

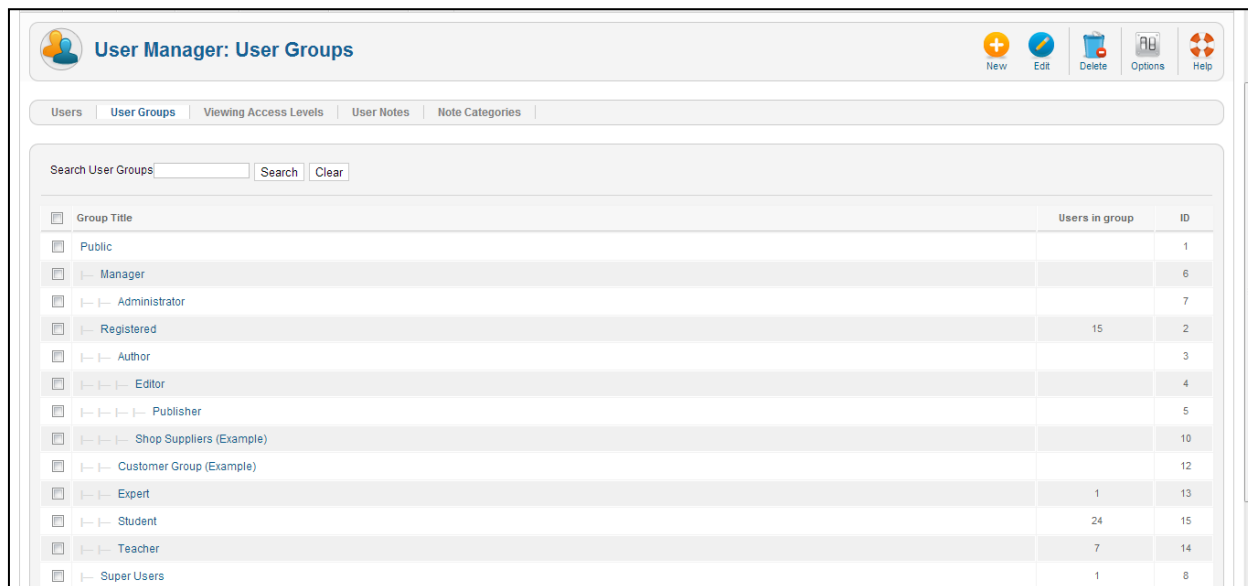
Επιλέγοντας κάποιο χρήστη οδηγούμαστε στη σελίδα Edit User, όπου ο διαχειριστής μπορεί να δει και να αλλάξει τα στοιχεία του χρήστη. Η σελίδα αυτή χρησιμοποιείται κυρίως για την αλλαγή του User Group του χρήστη, σε περίπτωση που αυτός κρίθηκε ότι ανήκει σε κάποιο διαφορετικό επίπεδο πρόσβασης από το καταχωρημένο.



The screenshot shows the Joomla! Administration interface for editing a user. The page is divided into several sections: Account Details, Assigned User Groups, Basic Settings, and User Profile. The Account Details section includes fields for Name, Login Name, Password, Confirm Password, Email, Registration Date, Last Visit Date, Last Reset Date, Password Reset Count, and options to receive system emails and block the user. The Assigned User Groups section shows a list of user groups with checkboxes, where 'Registered' is selected. The Basic Settings section includes dropdown menus for Backend Template Style, Backend Language, Frontend Language, Editor, Help Site, and Time Zone. The User Profile section is currently empty.

Edit User 5.2-17

### 5.2.5.1 Ομάδες Χρηστών



Group Title	Users in group	ID
Public		1
Manager		6
Administrator		7
Registered	15	2
Author		3
Editor		4
Publisher		5
Shop Suppliers (Example)		10
Customer Group (Example)		12
Expert	1	13
Student	24	15
Teacher	7	14
Super Users	1	8

**User Groups 5.2-18**

Όπως έχουμε προαναφέρει, οι χρήστες της ιστοσελίδας μπορούν να χωριστούν σε 5 κατηγορίες. Οι κατηγορίες αυτές δημιουργούνται και διαχειρίζονται στη σελίδα User Groups του Site Administrator. Επιλέγοντας μια από τις ομάδες χρηστών οδηγούμαστε στη σελίδα Edit Group που φαίνεται στο σχήμα 5.2.17 και εκεί μπορούμε απλώς να της δώσουμε όνομα και να ορίσουμε το «γονιό» της κατηγορίας αυτής.



Administration Joomla!

Site Users Menus Content Components Extensions Help 0 Visitors 1 Admin 23 View Site Log out

User Manager: Edit Group Save Save & Close Save & New Save as Copy Close Help

User Group Details

Group Title \* Student

Group Parent \* Registered

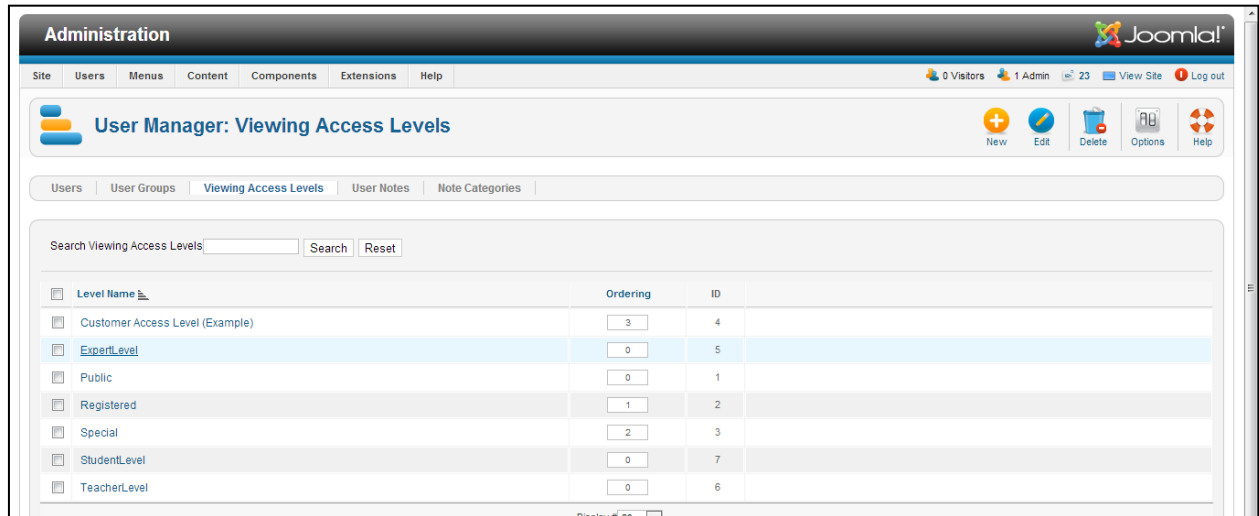
Joomla! 2.5.6  
Joomla!® is free software released under the GNU General Public License.

**Edit Group 5.2-19**

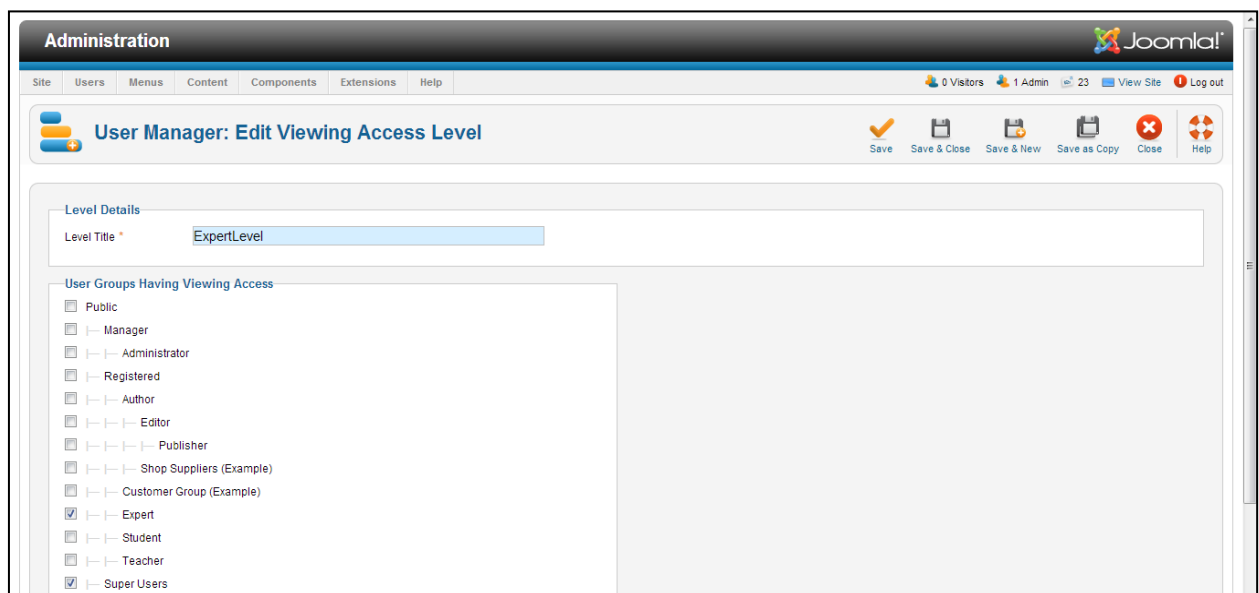
Οι κατηγορίες που δημιουργήσαμε για τους σκοπούς της ιστοσελίδας ήταν 3: **Student**, **Expert** και **Teacher**. Γονιός τους είναι η κατηγορία **Registered**, που όπως και η **Public** υπήρχε στο Joomla σαν προεπιλογή. Σε αυτές πρέπει να προστεθεί και η κατηγορία **Super User**, που αντιστοιχεί στους διαχειριστές του συστήματος, αυτούς δηλαδή που έχουν πλήρη πρόσβαση στο Site Administrator.

### 5.2.5.2 Επίπεδα Πρόσβασης

Η δημιουργία και η διαχείριση των επιπέδων πρόσβασης του συστήματος γίνεται στη σελίδα Viewing Access Levels του Site Administrator, που φαίνεται στο σχήμα 5.2-18. Δημιουργώντας ένα καινούργιο επίπεδο πρόσβασης ή επιλέγοντας κάποιο από τα διαθέσιμα, οδηγούμαστε στη σελίδα Edit Viewing Access Levels. Εκεί, όπως φαίνεται στο σχήμα 5.2-19, καθορίζουμε τις κατηγορίες χρηστών στις οποίες αντιστοιχεί το επίπεδο πρόσβασης, δηλαδή τους χρήστες που μπορούν να δουν το περιεχόμενο των σελίδων που έχουν χαρακτηριστεί με το επίπεδο πρόσβασης αυτό.



Viewing Access Levels 5.2-20



Edit Viewing Access Levels 5.2-21

## 6 ΑΝΑΛΥΣΗ ΛΕΙΤΟΥΡΓΙΩΝ ΑΡΧΕΙΩΝ PHP

### 6.1 ΔΗΜΙΟΥΡΓΙΑ ΔΙΑΔΡΟΜΩΝ

Είναι το αρχείο PHP που δημιουργεί τις εισόδους του σχεδιαστή, δηλαδή τα PDDL domain και PDDL problem, με βάση τις επιλογές του χρήστη (δασκάλου) στη σελίδα «Δημιουργία Διαδρομών». Στη συνέχεια, αφού η διαδρομή παραχθεί και αποθηκευτεί στη βάση δεδομένων Corgialen, περνά ως είσοδος στο επόμενο PHP αρχείο, όπου γίνεται η ανάθεσή της στους μαθητές.

#### 6.1.1 TeachersRoutes.php

Το αρχείο αυτό είναι το πιο πολύπλοκο από αυτά που χρησιμοποιεί η εφαρμογή, αφού είναι αυτό που δημιουργεί τις διαδρομές. Επεξεργάζεται τα δεδομένα που δίνει ο χρήστης στη σελίδα «Δημιουργία Διαδρομών» για να διαμορφώσει τα αρχεία PDDL που θα αποτελέσουν τις εισόδους του σχεδιαστή, καλεί μέσα από τη γραμμή εντολών την εκτέλεση της διαδικασίας λύσης του σχεδίου, παίρνει την παραγόμενη διαδρομή και αποθηκεύει τα δεδομένα της στη βάση.

Η βασική συνάρτηση του αρχείου είναι η LR(), η οποία και εκτελείται από τις εντολές που εισαγάγαμε στο αντίστοιχο άρθρο του Joomla, με τον τρόπο που έχουμε αναφέρει.

Αρχικά ορίζονται οι καθολικές (global) μεταβλητές που χρησιμοποιεί το αρχείο, οι οποίες όπως θα δούμε είναι πολλές και αναγκαίες. Το πρόγραμμα αρχίζει να εκτελείται με το πάτημα του πλήκτρου «Δημιουργία Διαδρομής» με την συνθήκη “if(isset(\$\_POST['submit']))”.

Η πρώτη συνάρτηση που καλείται στη συνέχεια είναι η **ConnectToCorg()**. Με τις εντολές **mysql\_connect** και **@mysql\_select\_db(corgialen)** επιτυγχάνεται η σύνδεση με τη βάση δεδομένων του συστήματος. Η συνάρτηση χρησιμοποιεί τις επιλογές (checkboxes) που έχει επιλέξει ο δάσκαλος στο πλαίσιο «ΔΙΑΛΕΞΕ ΚΑΤΗΓΟΡΙΕΣ» για να διαμορφώσει την εντολή SQL με την οποία θα γίνει η ανάκτηση δεδομένων από τη βάση. Για κάθε επιλεγμένη κατηγορία, στην εντολή προστίθεται η συμβολοσειρά " **or [Category] = 'y' "**, όπου y=Argostoli, Liksourι κλπ. Η συνάρτηση επιστρέφει την εντολή που διαμορφώνεται.

Στη συνέχεια καλείται η συνάρτηση **Filter()**. Αυτή παίρνει τα δεδομένα του δεύτερου πλαισίου της σελίδας, δηλαδή τα όρια των Μαθησιακών χαρακτηριστικών και τη ζητούμενη διάρκεια της διαδρομής, και πάλι με τη μέθοδο POST, και επιστρέφει επίσης μια εντολή SQL που ανακτά τα αντικείμενα της βάσης των οποίων τα 5 πρώτα μαθησιακά χαρακτηριστικά (εκτός του **time**) είναι μέσα στα ζητούμενα πλαίσια.

Οι δύο εντολές εκτελούνται ως σύζευξη με την εντολή **mysql\_query** και τα αντικείμενα μάθησης που πληρούν τις προϋποθέσεις ανακτώνται από τη βάση δεδομένων, πιο συγκεκριμένα από τον πίνακα *object*. Τα μαθησιακά χαρακτηριστικά τους(importance, difficulty, generality, beauty, quality, time), η χρονολογία τους(dating) και το ταυτοτικά στοιχείο (ID) τους αποθηκεύονται σε πίνακες. Τότε καλείται η

συνάρτηση **SortDate()**, η οποία επιστρέφει έναν πίνακα *objUsed* x 2, όπου *objUsed* το πλήθος των αντικειμένων που έχουν ανακτηθεί. Ο πίνακας αυτός στη μια στήλη έχει τα ID των αντικειμένων μάθησης και στην άλλη τη χρονολογία τους και είναι ταξινομημένος ως προς αυτήν. Ο πίνακας **SortDate** δίνεται ως όρισμα στη συνάρτηση **CreateLevels(\$sortDate, \$objUsed)**, η οποία επιστρέφει έναν πίνακα που περιέχει όλα τα αντικείμενα που είχε ο **sortDate**, αλλά σε κάθε γραμμή του περιέχει αντικείμενα (ID) που ανήκουν στην ίδια χρονολογία. Οι γραμμές είναι ταξινομημένες σε αύξουσα σειρά, δηλαδή η πρώτη γραμμή αντιστοιχεί στην παλαιότερη χρονολογία που εμφανίζεται.

Στη συνέχεια ανακτώνται με τη μέθοδο POST τα δεδομένα του τρίτου πλαισίου της σελίδας «Δημιουργία Διαδρομών», δηλαδή το όριο του πλήθους των αντικειμένων κάθε επιπέδου και τα βάρη των μαθησιακών χαρακτηριστικών. Το όριο χρησιμοποιείται μαζί με τον πίνακα που έχει δημιουργηθεί από την **CreateLevels()** ως είσοδος για τη συνάρτηση **FinalLevels()**. Η συνάρτηση αυτή μετατρέπει τον υπάρχοντα πίνακα σε έναν παρόμοιο ο οποίος ικανοποιεί τον περιορισμό του μέγιστου αριθμού αντικειμένων για κάθε επίπεδο, αν ο πρώτος δεν τον ικανοποιεί. Για κάθε γραμμή που έχει μεγαλύτερο αριθμό αντικειμένων από το επιτρεπτό όριο, ταξινομεί τα αντικείμενά της σε φθίνουσα σειρά ως προς το άθροισμα *importance+generality*, και τη «σπάει» σε μικρότερες, τοποθετώντας τα αντικείμενα με αυτή τη σειρά στις καινούργιες γραμμές, τοποθετώντας σε κάθε μια τόσα αντικείμενα όσα υποδεικνύει το επιτρεπτό όριο (εκτός από την τελευταία που εισάγονται όσα έχουν περισσέψει). Οι γραμμές αυτές αντιστοιχούν όπως θα δούμε στα επίπεδα από τα οποία ο σχεδιαστής θα επιλέγει τα αντικείμενα που θα αποτελέσουν τη διαδρομή.

Για τον πίνακα που επιστρέφει η **FinalLevels()**, η συνάρτηση **AttributeSum()** εκτελείται για κάθε ένα από τα 6 μαθησιακά χαρακτηριστικά και υπολογίζει το άθροισμα των τιμών του χαρακτηριστικού για τα αντικείμενα **κάθε γραμμής**, επιστρέφοντας 6 μονοδιάστατους πίνακες (*impSum[]*, *difSum[]* κλπ) μήκους *levels*, όπου *levels* το πλήθος των επιπέδων του πίνακα **FinalLevels**. Αντίστοιχα η συνάρτηση **AtrTotal** υπολογίζει το συνολικό άθροισμα κάθε χαρακτηριστικού (*impTot*, *difTot* κλπ) για όλα τα αντικείμενα του πίνακα.

Στη συνέχεια υπολογίζεται ο πίνακας **atrcombo[]**. Ο πίνακας αυτός είναι μονοδιάστατος και στη θέση **atrcombo[objId]** έχει το άθροισμα:  

$$\$wimp * \$importance[\$objId] + \$wdif * \$difficulty[\$objId] + \$wgen * \$generality[\$objId] + \$wbea * \$beauty[\$objId] + \$wqua * \$quality[\$objId],$$

όπου *wimp*, *wdif* κλπ τα βάρη που έχει θέσει ο δάσκαλος στο πλαίσιο «ΕΠΙΛΟΓΕΣ ΣΧΕΔΙΟΥ» και *importance[\$objId]*, *difficulty[\$objId]* κλπ οι τιμές των μαθησιακών χαρακτηριστικών του αντικειμένου (εκτός του **time**) με ID=objId.

Με την ίδια λογική υπολογίζεται ο μονοδιάστατος πίνακας  

$$\$atrSum[\$i] = \$wimp * \$impSum[\$i] + \$wdif * \$difSum[\$i] + \$wgen * \$genSum[\$i] + \$wbea * \$beaSum[\$i] + \$wqua * \$quaSum[\$i],$$

μήκους *levels*, καθώς και η τιμή  

$$\$atrTot = \$wimp * \$impTot + \$wdif * \$difTot + \$wgen * \$genTot + \$wbea * \$beaTot + \$wqua * \$quaTot;$$

Οι πίνακες και οι τιμές αυτές θα διαμορφώσουν, όπως θα δούμε, το αρχείο προβλήματος PDDL που παράγει το αρχείο.

Στη συνέχεια το πρόγραμμα καλεί τις συναρτήσεις **CreateDomain()** και **CreateProblem()**, οι οποίες παράγουν τα αρχεία PDDL, η δημιουργία και η δομή των οποίων αναλύεται στο επόμενο κεφάλαιο. Τα αρχεία PDDL αποθηκεύονται ως

```
$domainFile = $dompro."domain".$dt.".txt";
```

```
και $problemFile = $dompro."problem".$dt.".txt";
```

όπου *dompro* το path του φακέλου που περιέχει τον LPG και *dt* η ημερονηνία της στιγμής της δημιουργίας τους με ακρίβεια δευτερολέπτου, ώστε να εξασφαλίζεται η μοναδικότητα του ονόματος κάθε αρχείου.

Μετά τη δημιουργία τους, καλείται η συνάρτηση **LpgPlannerLearningRoute()**. Η συνάρτηση αυτή ξεκινά με την εντολή `shell_exec($dompro."lpg-td-1.0 -o ".$domainFile." -f ".$problemFile." -n 1 -cputime 10 ")`; Η εντολή αυτή εκτελεί μέσα από τη γραμμή εντολών το σχεδιαστή LPG, με εισόδους *domainFile* και *problemFile*, ενώ το όρισμα `-n 1` ορίζει ότι ζητάμε μόνο ένα σχέδιο λύσης. Το όρισμα `-cputime 10` ορίζει ότι ο σχεδιαστής θα ψάξει για κάποιο σχέδιο λύσης το πολύ για 10 δευτερόλεπτα.

Στην πραγματικότητα, οι συναρτήσεις **CreateProblem()** και **LpgPlannerLearningRoute()** μπορεί να χρειαστεί να εκτελεστούν παραπάνω από μια φορά για να προκύψει το σχέδιο λύσης. Επειδή όμως για να κατανοήσουμε τις συνθήκες υπό τις οποίες γίνεται η κλήση τους πρέπει να μελετήσουμε το αρχείο προβλήματος PDDL, η πλήρης εξήγηση του τρόπου λειτουργίας τους γίνεται στο υποκεφάλαιο 6.1.2.3.

Η λύση που τελικά παράγεται από την εκτέλεση του σχεδιαστή αποθηκεύεται στο .SOL αρχείο `$solFile="plan_problem".$dt.".txt_1.SOL"`; Το αρχείο αυτό περιέχει αρχικά κάποια στοιχεία του προβλήματος και στη συνέχεια το σχέδιο λύσης, δηλαδή μια ακολουθία από συμβολοσειρές της μορφής **(OBJ STUDENT1 LEVEL0 LEVEL1 OBJECT630)**.

Η συμβολοσειρά αυτή αντιστοιχεί στην ενέργεια OBJ, που δείχνει ότι ο μαθητής STUDENT1 πρέπει σε αυτό το σημείο της διαδρομής να «δει» το αντικείμενο OBJECT630 που ανήκει στο LEVEL1. Η συνάρτηση ανοίγει και διαβάζει το αρχείο .SOL και με τις συναρτήσεις *split()*, *strtok()* και *intval()* της PHP ακολουθεί τη σειρά των ID των αντικειμένων (πχ 630 στο παράδειγμα), που διαμορφώνουν τη διαδρομή. Η διαδρομή αποθηκεύεται ως μια συμβολοσειρά που περιέχει την ακολουθία των ID των αντικειμένων μάθησης, πχ "630 622 623 600 378". Τέλος, το ID της καινούργιας διαδρομής (το πρώτο διαθέσιμο), η διαδρομή με τη μορφή συμβολοσειράς και το όνομά της αποθηκεύονται ως μια καινούργια εγγραφή στον πίνακα *routes* της βάσης *corgialen*.

Το πρόγραμμα με την εντολή *header()* της PHP οδηγεί στη σελίδα *Assign Routes To Students*, που περιγράφεται στο υποκεφάλαιο 6.1.3, περνώντας μέσω του url το ID της διαδρομής που παράχθηκε:

```
header("Location:http://localhost:64895/index.php?option=com_content&view=article&id=85  
&Itemid=476&idValue=".$newRouteId);
```

Η τιμή *id=85* είναι το id του άρθρου *Assign Routes To Students*, όπως έχει αποθηκευτεί από το Joomla. Το *Itemid* είναι το id του menu item που χρησιμοποιεί το άρθρο για την εμφάνισή του. Στα άρθρα στα οποία οδηγούμαστε μόνο μέσω κάποιου άλλου άρθρου, και όχι κατευθείαν από το μενού, έχουμε αντιστοιχίσει το menu item με *id=476*, ώστε να ισχύουν οι επιλογές εμφάνισης που θέλουμε και όχι οι προεπιλεγμένες. Τέλος, η τιμή *idValue* περιέχει, όπως βέπουμε, το id της διαδρομής που μόλις παράχθηκε.

Στο τέλος του αρχείου γράφουμε τον απαραίτητο HTML κώδικα για την εμφάνιση της σελίδας, όπως περιγράφεται στο κεφάλαιο 4.2.1.1. Δημιουργούμε δηλαδή τα checkboxes, τα textfields και τα

submit buttons που χρησιμοποιούμε στη σελίδα και ορίζουμε τη μορφή τους και τις επιλογές εμφάνισης. Οι επιλογές αυτές καθορίζονται κυρίως με χρήση κώδικα CSS, δηλαδή εντολές της μορφής *STYLE= "height: 40px; color: #000; font-family: Verdana; "*, ξεχωριστά όμως για κάθε div που χρησιμοποιούμε.

## 6.1.2 Αρχεία PDDL

Τα αρχεία PDDL δημιουργούνται στους υποκαταλόγους (directories) που αναφέραμε ως αρχεία κειμένου με την εντολή *foopen* και «γράφονται» από τον κώδικα PHP του *TeacherRoutes.php* με τις εντολές *fwrite*. Κάθε εντολή *fwrite* γράφει μια καινούργια γραμμή στα αρχεία, παραδείγματα των οποίων φαίνονται στα σχήματα 6.1-1 και 6.1-2. Τα αρχεία *domain* είναι ίδια για κάθε πρόβλημα, ενώ τα *problem* είναι αυτά που περιέχουν τα διαφορετικά στοιχεία κάθε φορά.

### 6.1.2.1 Αρχείο Domain PDDL

Ένα παράδειγμα αρχείου *Domain.pddl* που παράχθηκε από την εφαρμογή είναι αυτό του σχήματος 6.1-1. Αρχικά ορίζεται το όνομά του (*Corgialeneion*) και στη συνέχεια οι απαιτήσεις (requirements). Όπως έχουμε δει, οι απαιτήσεις του παραδείγματος δηλώνουν ότι το *domain* χρησιμοποιεί τη γλώσσα STRIPS, τύπους και συναρτήσεις.

Με κόκκινα γράμματα φαίνονται οι τύποι που χρησιμοποιούνται. Ο γενικός είναι ο *object* που υπάρχει ως προεπιλογή και ως υποσύνολά του ορίζονται οι τύποι *student* και *level*. Ο τύπος *level* αντιστοιχεί στα επίπεδα στα οποία έχουν διαχωριστεί τα αντικείμενα μάθησης, ενώ ο *student* χρησιμοποιείται για να ορίζεται κάθε φορά η θέση του μαθητή, δηλαδή σε ποιο επίπεδο βρίσκεται τη δεδομένη χρονική στιγμή του σχεδίου λύσης. Τα αντικείμενα μάθησης αντιστοιχούν κατευθείαν στο γενικό τύπο *object*.

Στη συνέχεια, με μαύρα γράμματα ορίζονται τα κατηγορήματα που χρησιμοποιούνται. Το *at(x lev-y)* συμβολίζει το γεγονός ότι το επίπεδο στο οποίο ανήκει το *object x* είναι το *lev-y* αν το *x* είναι *object*, ιδιότητα που ορίζεται αρχικά και παραμένει αμετάβλητη κατά τη διάρκεια του σχεδίου, ενώ αν το *x* είναι τύπου *student* συμβολίζει ότι το *lev-y* είναι το τρέχον επίπεδο στο οποίο βρίσκεται ο μαθητής κάποια δεδομένη στιγμή, δηλαδή το επίπεδο στο οποίο ανήκει το τελευταίο αντικείμενο μάθησης που είδε. Το *done(?o)* συμβολίζει το γεγονός ότι το αντικείμενο *?o* έχει συμπεριληφθεί ήδη στη διαδρομή, ενώ το *next(?I1 ?I2)* δηλώνει ότι το επίπεδο *?I2* είναι το ακριβώς επόμενο του *?I1*.

Με πορτοκαλί γράμματα ορίζονται οι συναρτήσεις που χρησιμοποιεί ο κόσμος του προβλήματος. Οι συναρτήσεις που παραμένουν αμετάβλητες κατά τη διάρκεια του σχεδίου είναι οι (*minval ?l*), (*atrvalue ?o*) και (*time ?o*), όπου *l* είναι τύπου *level* και *o* τύπου *object*. Η (*atrvalue ?o*) είναι η τιμή της μεταβλητής *atrcombo[objId]* που αναφέραμε στο αρχείο PHP και συμβολίζει την τιμή του συνδυασμού των μαθησιακών χαρακτηριστικών που αντιστοιχεί στο αντικείμενο *?o*. Η (*minval ?l*) είναι αυτή που συμβολίζει το ελάχιστο άθροισμα από *atrvalue* που πρέπει να συμπληρώνουν τα αντικείμενα μάθησης

που επιλέγονται από το επίπεδο  $?l$ . Η  $(time ?o)$  είναι η τιμή του μαθησιακού χαρακτηριστικού  $time$  του αντικειμένου  $?o$ . Οι τιμές των συναρτήσεων αυτών αρχικοποιούνται όπως θα δούμε στο αρχείο `problem`.

Η συνάρτηση  $(levelvalue ?l)$  περιέχει το τρέχον άθροισμα  $atrvalue$  που έχουν τα `object` του επιπέδου  $?l$ , ενώ οι  $totalvalue$  και  $totaltime$  περιέχουν το τρέχον άθροισμα  $atrvalue$  και  $time$  όλων των αντικειμένων που έχουν επιλεχθεί μέχρι τη δεδομένη στιγμή από το σχεδιαστή. Οι συναρτήσεις αυτές αρχικοποιούνται στην τιμή 0 στην αρχή του προβλήματος.

```
(define (domain corgialeneion)

  (:requirements :strips :typing :fluents)

  (:types student level - object)

  (:predicates
   (at ?x - (either student object) ?l - level)
   (done ?o - object)
   (next ?l1 - level ?l2 - level))

  (:functions
   (levelvalue ?l - level)
   (minval?l - level)
   (atrvalue ?o - object)
   (totalvalue)
   (time ?o - object)
   (totaltime)
  )

  (:action obj
   :parameters (?s - student ?l1 ?l2 - level ?o - object )
   :precondition (and (at ?o ?l2)
                      (next ?l1 ?l2)
                      (or
                       (at ?s ?l1)
                       (at ?s ?l2)
                      )
                      (not (done ?o))
                      (> (levelvalue ?l1) (minval ?l1)))
  )
  :effect (and (not (at ?s ?l1))
               (at ?s ?l2)
               (done ?o)
               (increase (levelvalue ?l2)
                .(atrvalue ?o).)
               (increase (totalvalue)
                (atrvalue ?o) )
               (increase (totaltime)
                (time ?o) )
  ))
)
```

#### Domain PDDL File 6.1-1

Με τα μπλε γράμματα φαίνεται το μοναδικό σχήμα ενεργειών `obj` του κόσμου του προβλήματος. Όπως έχουμε πει, ένα σχήμα ενεργειών μεταφράζεται σε πολλές ενέργειες με τις διάφορες αποτιμήσεις των παραμέτρων του. Οι παράμετροι του `obj` είναι ένας μαθητής( $?s$ ), 2 επίπεδα( $?l1$ ,  $?l2$ ) και ένα αντικείμενο μάθησης( $?o$ ), το οποίο χαρακτηρίζει και την ενέργεια. Στην πραγματικότητα, η ενέργεια μεταφράζεται ως «ο μαθητής βλέπει το αντικείμενο μάθησης  $?o$ », δηλαδή ότι το  $?o$  συμπεριλαμβάνεται στην εκπαιδευτική διαδρομή στη συγκεκριμένη θέση. Το  $?l2$  είναι το επίπεδο όπου βρίσκεται το αντικείμενο  $?o$  ενώ το  $?l1$  είναι το προηγούμενό του.



Αυτό γίνεται πιο ξεκάθαρο στην προϋπόθεση (precondition) της ενέργειας, που είναι η σύζευξη (and) 5 γεγονότων:

- (at ?o ?I2): Το αντικείμενο μάθησης ?o ανήκει στο επίπεδο ?I2.
- (next ?I1 ?I2): Το επίπεδο ?I2 είναι το επόμενο του ?I1.
- (or (at ?s ?I1) (at ?s ?I2) ): Ο μαθητής πρέπει να βρίσκεται ή στο ?I1 ή στο ?I2.
- (not (done ?o)): το αντικείμενο μάθησης ?o δεν πρέπει να έχει συμπεριληφθεί προηγουμένως στη διαδρομή.
- (> (levelvalue ?I1) (minval ?I1)): Το συνολικό άθροισμα των τιμών atrvalue των αντικειμένων που έχουν επιλεγθεί από το προηγούμενο level πρέπει να ξεπερνούν την ελάχιστη τιμή που έχει οριστεί.

Αντίστοιχα ορίζεται και το αποτέλεσμα(effect) της ενέργειας:

- (not (at ?s ?I1): Ο μαθητής δε βρίσκεται πλέον στο επίπεδο ?I1, αν αυτό ίσχυε πριν.
- (at ?s ?I2): Ο μαθητής βρίσκεται στο ?I2, κάτι που μπορεί να ίσχυε και πριν.
- (done ?o): Το αντικείμενο που επιλέχθηκε έχει συμπεριληφθεί στη διαδρομή, άρα δεν μπορεί να ξαναεπιλεγθεί.
- (increase (levelvalue ?I2) (atrvalue ?o) ): Η τιμή levelvalue του επιπέδου ?I2 αυξάνεται κατά την τιμή atrvalue του αντικειμένου ?o.
- (increase (totalvalue) (atrvalue ?o) ): Το ίδιο και η τιμή totalvalue.
- (increase (totaltime) (time ?o) ): Η τιμή totaltime αυξάνεται κατά την τιμή time που αντιστοιχεί στο αντικείμενο μάθησης ?o.

### 6.1.2.2 Problem PDDL File

Ένα παράδειγμα αρχείου Problem.pddl φαίνεται στο σχήμα 6.1-2 της επόμενης σελίδας. Αρχικά ορίζεται το όνομά του και το domain που του αντιστοιχεί.

Με τα πράσινα γράμματα φαίνονται οι ορισμοί των αντικειμένων που θα χρησιμοποιηθούν και του τύπου στον οποίο ανήκει το καθένα, δηλαδή μαθητής, επίπεδα και αντικείμενα μάθησης. Τα κόκκινα γράμματα αντιστοιχούν στην αρχική κατάσταση του κόσμου (init). Εδώ αρχικοποιούνται οι τιμές των κατηγορημάτων και των συναρτήσεων.

Στην αρχική κατάσταση ο μαθητής βρίσκεται στο επίπεδο 0. Το επίπεδο -1 είναι εικονικό και ό,τι αναφέρεται σε αυτό στο problem ((= (levelvalue level-1) 9999), (next level4 level-1), (at object99999 level-1)) χρησιμοποιείται για λειτουργικούς λόγους, αφού όπως είδαμε στην περιγραφή του domain, για να επιλεγθεί ένα αντικείμενο μάθησης από ένα επίπεδο πρέπει να έχουν επιλεγθεί προηγουμένως κάποια από το προηγούμενο.

Για κάθε επίπεδο ορίζεται αρχικά το επόμενο του ((next level0 level1)). Αρχικοποιείται σε 0 η συνάρτηση levelvalue που του αντιστοιχεί ((= (levelvalue level0) 0)) και ορίζεται η τιμή της συνάρτησης

<pre> (define (problem corgProblem) (:domain corgialeneion) (:objects student1 – student level-1 – level level0 – level object219 – object object214 – object object432 – object object368 – object object148 – object object165 – object level1 – level object255 – object object363 – object object160 – object object19 – object object171 – object object209 – object level2 – level object253 – object object444 – object object459 – object object212 – object object466 – object object384 – object level3 – level object461 – object level4 – level object221 – object object292 – object object99999 - object ) (:init (at student1 level0) (next level-1 level0) (= (minval level-1)0) (next level0 level1) (= (levelvalue level0) 0) (= (minval level0)171.5) (at object219 level0) (= (atrvalue object219) 59) (= (time object219) 7) (at object214 level0) (= (atrvalue object214) 57) (= (time object214) 4) (at object432 level0) (= (atrvalue object432) 67) (= (time object432) 9) (at object368 level0) (= (atrvalue object368) 57) (= (time object368) 5) (at object148 level0) (= (atrvalue object148) 54) (= (time object148) 1) (at object165 level0) (= (atrvalue object165) 49) ) ) </pre>	<pre> (= (time object165) 5) (next level1 level2 ) (= (levelvalue level1) 0) (= (minval level1)161) (at object255 level1) (= (atrvalue object255) 43) (= (time object255) 3) (at object363 level1) (= (atrvalue object363) 64) (= (time object363) 5) (at object160 level1) (= (atrvalue object160) 62) (= (time object160) 5) (at object19 level1) (= (atrvalue object19) 62) (= (time object19) 2) (at object171 level1) (= (atrvalue object171) 41) (= (time object171) 3) (at object209 level1) (= (atrvalue object209) 50) (= (time object209) 2) (next level2 level3) (= (levelvalue level2) 0) (= (minval level2)144.5) (at object253 level2) (= (atrvalue object253) 37) (= (time object253) 1) (at object444 level2) (= (atrvalue object444) 58) (= (time object444) 1) (at object459 level2) (= (atrvalue object459) 62) (= (time object459) 5) (at object212 level2) (= (atrvalue object212) 47) (= (time object212) 4) (at object466 level2) (= (atrvalue object466) 43) (= (time object466) 5) (at object384 level2) (= (atrvalue object384) 42) (= (time object384) 5) (next level3 level4) (= (levelvalue level3) 0) (= (minval level3)20.5) (at object461 level3) (= (atrvalue object461) 41) (= (time object461) 1) (= (levelvalue level4) 0) (= (minval level4)53.5) (at object221 level4) (= (atrvalue object221) 42) (= (time object221) 6) (at object292 level4) (= (atrvalue object292) 65) (= (time object292) 3) (= (levelvalue level-1) 9999) (next level4 level-1) ) </pre>	<pre> (at object99999 level-1) ) (:goal ( and(at student1 level4) (&lt; (totaltime)100 ) )) (:metric maximize (totalvalue)) ) </pre>
--	---	--

**Problem PDDL File 6.1-2**

$minval$  του επιπέδου ( $= (minval\ level0)171.5$ ), που προκύπτει με τον τρόπο που θα δούμε στο επόμενο υποκεφάλαιο.

Στη συνέχεια, τα αντικείμενα μάθησης που αντιστοιχούν στο επίπεδο αυτό, δηλαδή αυτά που περιέχονται στην αντίστοιχη γραμμή του πίνακα `FinalLevels[][]`, ορίζεται ότι ανήκουν σε αυτό ( $(at\ object219\ level0)$ ). Για κάθε ένα από αυτά ορίζεται η τιμή  $atrvalue$  του ( $(=(atrvalue\ object219)\ 59)$ ), που ισούται με την τιμή του πίνακα `atrcombo[objId]` που έχουμε αναφέρει, καθώς και η τιμή  $time$  του ( $(=(time\ object219)\ 7)$ ), που ισούται με το πεδίο  $time$  που του αντιστοιχεί στον πίνακα `object` της βάσης `corgialen`. Όπως έχουμε αναφέρει στο κεφάλαιο 2, τα γεγονότα που δεν αναφέρονται λογίζονται ως ψευδή, κάτι που σημαίνει ότι για όλα τα αντικείμενα μάθησης `objectX` ισχύει το ***not done objectX***.

Με τα μπλε γράμματα φαίνεται ο στόχος του προβλήματος, που είναι η σύζευξη των επιμέρους στόχων:

- (`and(at student1 level4)`): Ο μαθητής πρέπει στο τέλος να βρίσκεται στο τελευταίο επίπεδο(εδώ 4), δηλαδή να έχει δει τουλάχιστον 1 αντικείμενο του τελευταίου επιπέδου.
- (`< (totaltime)100`): Η συνολική διάρκεια της διαδρομής σε λεπτά να μην ξεπερνάει αυτή που έχει ορίσει ο δάσκαλος.

Τέλος, με πορτοκαλί γράμματα ορίζεται ο δείκτης ποιότητας (*metric*) του σχεδίου. Για τα αρχεία προβλημάτων μας έχουμε ορίσει ότι αυτός θα είναι η αύξηση της τιμής  $totalvalue$ , που σημαίνει ότι η ποιότητα του σχεδίου καθορίζεται από το συνολικό άθροισμα των τιμών  $atrvalue$  των αντικειμένων που θα επιλεγθούν. Σκοπός του σχεδιαστή είναι η παραγωγή ενός σχεδίου που καλύπτει το στόχο που έχει οριστεί και έχει τη μεγαλύτερη δυνατή  $totalvalue$ , στο επίπεδο που αυτό είναι εφικτό.

### 6.1.2.3 Οι στόχοι του σχεδίου

Η συνάρτηση `CreateProblem` που είδαμε στο υποκεφάλαιο 6.1.2.1 έχει 8 παραμέτρους:  
`CreateProblem (array $finlevels, $lev, array $attribute, array $attributeSum, $attributeTotal, $totalTime, $maxDurationDesired, $maxDuration) ();`

Τις 5 πρώτες τις έχουμε αναλύσει ήδη στην περιγραφή του αρχείου PHP. Η  $totalTime$  είναι η τιμή  $$timeTot$  που έχουμε αναφέρει του `TeacherRoutes.php` και είναι ίση με το άθροισμα των εκτιμώμενων διαρκειών όλων των αντικειμένων του πίνακα `FinalLevels`, δηλαδή όλων των υποψήφιων αντικειμένων της διαδρομής. Η τιμή αυτή είναι μια εκτίμηση, λοιπόν, της διάρκειας που θα είχε η διαδρομή αν περιελάμβανε όλα τα αντικείμενα του πίνακα `FinalLevels`. Η παράμετρος  $maxDurationDesired$  είναι η ζητούμενη μέγιστη διάρκεια που ορίζει ο δάσκαλος μέσω της σελίδας «Δημιουργία Διαδρομών», ενώ η  $maxDuration$  είναι αρχικά ίση με τη  $maxDurationDesired$  και χρησιμοποιείται με τον ακόλουθο τρόπο:

Στην αρχή της `CreateProblem()` υπολογίζουμε το συντελεστή  $a = \frac{maxDuration}{totalTime}$ . Με βάση το συντελεστή αυτόν υπολογίζονται οι τιμές  $minval$  κάθε επιπέδου του αρχείου προβλήματος:

```
 $minSum = a * attributeSum[i];$   
fwrite($fh, "\r\n". "(= (minval level". $i. ")". $minSum. ")");
```

Δηλαδή η τιμή  $\text{minval}$  κάθε επιπέδου  $x$  του προβλήματος ισούται με το  $\text{minSum}$ , δηλαδή το γινόμενο του συντελεστή  $a$  επί το άθροισμα  $\text{atrcombo}$  όλων των αντικειμένων μάθησης του επιπέδου.

Αν ο σχεδιαστής δε βρει λύση για το πρόβλημα αυτό, που σημαίνει ότι με μέγιστη διάρκεια  $\text{maxDurationDesired}$  δε γίνεται να ικανοποιηθούν τα ελάχιστα  $\text{minval}$  κάθε επιπέδου, τότε το αρχείο `Problem.pddl` ξαναγράφεται με  $\text{\$maxDuration}=\text{\$maxDuration}*0.75$ , αλλά με ίδιο  $\text{maxDurationDesired}$ . Μειώνουμε δηλαδή το συντελεστή  $a$  κατά 75% χωρίς να μειώσουμε τη μέγιστη ζητούμενη διάρκεια της διαδρομής. Και στη συνέχεια εκτελούμε και πάλι τη συνάρτηση `LpgPlannerLearningRoute()`. Αν και πάλι δε βρεθεί λύση, δοκιμάζουμε άλλη μια μείωση 75%, και αν αποτύχει και η 3<sup>η</sup> προσπάθεια τότε δεν παράγεται κάποια διαδρομή και η εφαρμογή εμφανίζει μήνυμα που δίνει μια εκτίμηση της ελάχιστης διάρκειας για τα συγκεκριμένα δεδομένα.

Ο σχεδόν αποκλειστικός λόγος για να μη βρεθεί σχέδιο λύσης σε μια τέτοια περίπτωση είναι ο σχετικά μεγάλος αριθμός επιπέδων που μπορεί να έχει το πρόβλημα. Όπως έχουμε πει, από κάθε επίπεδο πρέπει να επιλέγεται τουλάχιστον 1 αντικείμενο μάθησης. Αυτό σημαίνει πως αν έχουμε  $N$  επίπεδα, τότε η διαδρομή θα αποτελείται από τουλάχιστον  $N$  αντικείμενα μάθησης, άρα θα έχει ελάχιστη διάρκεια  $\sum_{i=1}^N \text{time}[i]$ , όπου  $\text{time}[i]$  η διάρκεια του αντικειμένου με το ελάχιστο  $\text{time}$  από το επίπεδο  $i$ .

Για το λόγο αυτό, η εκτίμηση που εμφανίζεται υπολογίζεται με βάση τον αριθμό επιπέδων του πίνακα `FinalLevels` και ισούται με το γινόμενο του αριθμού αυτού, ο οποίος επίσης τυπώνεται μαζί με την εκτίμηση, επί τη μέση διάρκεια των αντικειμένων μάθησης. Αν ο μεγάλος αριθμός επιπέδων που έχει προκύψει είναι αποτέλεσμα του μικρού ορίου που έχει θέσει ο δάσκαλος στον αριθμό αντικειμένων κάθε επιπέδου, τότε η απαιτούμενη διάρκεια μπορεί να επιτευχθεί μειώνοντας τα επίπεδα, δηλαδή επιλέγοντας ένα μεγαλύτερο όριο στον αριθμό αντικειμένων κάθε επιπέδου. Αν οφείλεται στο μεγάλο αριθμό διαφορετικών χρονολογιών που εμφανίζονται στα υπονήφια αντικείμενα, δεν υπάρχει κάποιος τρόπος να επηρεάσουμε άμεσα το σχέδιο, αφού όπως έχουμε πει οι διαφορετικές χρονολογίες αντιστοιχούν σε ξεχωριστά επίπεδα. Σε αυτήν την περίπτωση ο χρήστης (δάσκαλος) θα πρέπει να περιορίσει τον αριθμό των υπονήφια αντικειμένων πριν τη δημιουργία του σχεδίου, ρυθμίζοντας αυστηρότερα όρια σε κάποια μαθησιακά χαρακτηριστικά, ανάλογα πάντα με τις ανάγκες της διαδρομής και των μαθητών στους οποίους θα την αναθέσει.

### 6.1.3 AssignRoutesToStudents.php

Όπως έχουμε αναφέρει παραπάνω, στη σελίδα αυτή οδηγούμαστε μετά τη δημιουργία μιας εκπαιδευτικής διαδρομής, μέσω του αρχείου `TeacherRoutes.php`, με την εντολή `header()` της PHP. Με την εντολή αυτή περνάμε και μέσω του `url` το `id` της διαδρομής που μόλις παράχθηκε.

Το `id` αυτό λαμβάνεται από το πρόγραμμα με τη μέθοδο `GET: $routeId = $_GET['idValue'];`. Στη συνέχεια καλείται η κεντρική συνάρτηση του αρχείου, η `StudentSelection($routeId,$userId)`, όπου `$userId` είναι το `Id` του τρέχοντος συνδεδεμένου χρήστη, δηλαδή του δασκάλου που δημιούργησε τη διαδρομή.

Αρχικά συνδεόμαστε στη βάση δεδομένων κατά τα γνωστά, για να ανακτήσουμε την εκπαιδευτική διαδρομή από τον πίνακα `routes` της βάσης `orgialen` χρησιμοποιώντας το `id` της. Αφού εμφανίσουμε τη

διαδρομή, ανακτούμε από τον πίνακα *teacherstudents* τα id των μαθητών που ανήκουν στην τάξη του δασκάλου με Id=userId. Συνδεόμαστε στη βάση του Joomla για να ανακτήσουμε και τα ονόματά τους και για κάθε έναν από αυτούς δημιουργούμε με κώδικα HTML ένα πεδίο που περιέχει ένα checkbox, το όνομα και το id του.

Εισάγουμε, επίσης, ένα submit button. Όταν αυτό πατηθεί, το πρόγραμμα ελέγχει ποια checkboxes είναι επιλεγμένα (*if (isset(\$\_POST['student'.\$stId[\$i]])*)) και για κάθε ένα από αυτά δημιουργεί μια καινούργια εγγραφή στον πίνακα *studentroutes*, αναθέτοντας στο μαθητή τη διαδρομή που δημιουργήθηκε στην προηγούμενη σελίδα.

## 6.2 ΤΕΚΜΗΡΙΩΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ ΜΑΘΗΣΗΣ

Στην σελίδα αυτή αξιολογούνται τα μαθησιακά χαρακτηριστικά των αντικειμένων μάθησης που υπάρχουν στη βάση, δηλαδή των εκθεμάτων του μουσείου.

Στην πραγματικότητα τα αρχεία PHP που αντιστοιχούν στη λειτουργία αυτή είναι παραπάνω από 1, αφού αντιστοιχεί μια σελίδα σε κάθε 100 αντικείμενα. Αυτό γίνεται για τη διευκόλυνση της επεξεργασίας των δεδομένων, αφού σε περίπτωση μαζικής καταχώρησης τιμών αυτό γίνεται σε μικρότερες ομάδες αντικειμένων, αλλά έχει και χρηστική λειτουργία αφού και ο «ειδικός» μπορεί να δουλέψει με μικρότερες ομάδες αντικειμένων.

### 6.2.1 ExpertChanges.php

Θα περιγράψουμε το αρχείο του πρώτου άρθρου αυτής της κατηγορίας, δηλαδή αυτού που αντιστοιχεί στην πρώτη εκατοντάδα αντικειμένων. Με αντίστοιχο τρόπο περιγράφονται και όλα τα υπόλοιπα αρχεία *ExpertChangesN.php* με μόνη διαφορά ότι σε αυτά γίνεται επεξεργασία άλλων αντικειμένων.

Η βασική συνάρτηση που χρησιμοποιούμε είναι η *exch1()*. Με την κλήση της αρχικά η εφαρμογή συνδέεται στη βάση *corgialen* κατά τα γνωστά. Στη συνέχεια ανακτά τα Learning attributes των 100 πρώτων αντικειμένων, ταξινομημένων με βάση το *id* τους, και τα κρατά σε πίνακες (*difficulty[]*, *importance[]* κλπ). Χρησιμοποιώντας κώδικα HTML δημιουργούμε για κάθε αντικείμενο ένα σύνολο από text fields που περιέχουν τις υπάρχουσες τιμές των χαρακτηριστικών, με το αντίστοιχο label, και ένα submit button για την καταχώρηση. Στην αρχή της σελίδας τοποθετούμε ένα ακόμα submit button με όνομα *submitAll*, που χρησιμοποιείται στην περίπτωση συνολικής καταχώρησης των χαρακτηριστικών όλων των αντικειμένων της σελίδας.

Στην περίπτωση που πατηθεί το submit button κάποιου αντικειμένου με *id=\$objId*, τότε καλείται η συνάρτηση *update(\$objId)*, η οποία διαβάζει με τη μέθοδο POST της HTML τις τιμές των text fields κάθε αντικειμένου και τις καταχωρεί στη βάση, με την εντολή UPDATE της γλώσσας SQL. Αν επιλεγθεί το *SubmitAll*, τότε η *update(\$objId)* καλείται για όλα τα *\$objId* των αντικειμένων της σελίδας (εδώ 100). Σε κάθε πάτημα Submit button πραγματοποιείται και μια ανανέωση της σελίδας με την εντολή `<META HTTP-EQUIV=Refresh CONTENT="0">`, ώστε να εμφανιστούν στα text fields οι νέες τιμές που καταχωρήθηκαν.

Σε όλα τα αρχεία *ExpertChangesN.php* τοποθετείται στο τέλος της σελίδας ένας σύνδεσμος (*href*) για την επόμενη και ένας για την προηγούμενη σελίδα *ExpertChanges*, ώστε να μπορεί ο χρήστης να μεταφερθεί στη σελίδα αξιολόγησης των αντικειμένων της επόμενης και της προηγούμενης 100άδας.

## 6.3 ΔΙΑΔΡΟΜΕΣ

Εδώ περιγράφονται τα άρθρα που αφορούν στην εκτέλεση των εκπαιδευτικών διαδρομών από τους μαθητές. Το `Course.php` παρουσιάζει τις διαθέσιμες διαδρομές στο μαθητή για να επιλέξει μία από αυτές, το `Routeview.php` παρουσιάζει με τη σειρά τα αντικείμενα που αποτελούν την επιλεγθείσα διαδρομή, ενώ όταν ο μαθητής επιλέξει το αντικείμενο που θα μελετήσει εκτελείται το `Objectinfo.php` και παρουσιάζει την εικόνα και τις πληροφορίες που αντιστοιχούν στο επιλεγθέν αντικείμενο.

### 6.3.1 Course.php

Εδώ έχουμε μία συνάρτηση, την `DisplayChoices($userId)`. Η ανάκτηση του `Id` του τρέχοντος χρήστη γίνεται από τις εντολές που εισάγουμε στο σχεδιαστή άρθρων του Joomla με τον τρόπο που έχουμε περιγράψει.

Αρχικά το πρόγραμμα συνδέεται στη βάση και επιλέγει από τον πίνακα `studentroutes` τις εγγραφές που έχουν στο πεδίο `studentId` το `Id` του συνδεδεμένου μαθητή. Αποθηκεύει στον πίνακα `$routeId[]` τα `Id` των διαδρομών των εγγραφών αυτών και στον πίνακα `$done[]` το πεδίο `done` του πίνακα, το οποίο καθορίζει αν ο μαθητής έχει ακολουθήσει τη διαδρομή σε προηγούμενη επίσκεψη. Στη συνέχεια συνδέεται στον πίνακα `routes` της βάσης για να ανακτήσει τις διαδρομές, δηλαδή τις συμβολοσειρές που αποτελούνται από τα αντικείμενα μάθησης. Τις διαδρομές τις αποθηκεύει στον πίνακα `$route[]` και για κάθε μια δημιουργεί με κώδικα HTML ένα πεδίο που περιέχει ένα submit button, το όνομα και τα αντικείμενα μάθησης της διαδρομής, καθώς και την ένδειξη `DONE` αν ο μαθητής έχει ξαναεπιλέξει τη διαδρομή. Ανάλογα με το submit button που θα πατήσει ο μαθητής, με τη συνάρτηση `header("url")` της `php` οδηγείται στη σελίδα `Route View` περνώντας στην παράμετρο `idValue` του `url` την τιμή του `Id` της διαδρομής που επιλέχθηκε.

### 6.3.2 Routeview.php

Η συνάρτηση που περιέχει τη λειτουργία του αρχείου είναι η `followRoute($routeId)`. Η τιμή `$routeId` λαμβάνεται με τη μέθοδο `GET` της HTML από το `url` με το οποίο οδήγησε στη σελίδα αυτή η προηγούμενη, με τον τρόπο που περιγράψαμε στο 6.3.1. Το πρόγραμμα συνδέεται στη βάση και με το SQL query `"SELECT route FROM routes WHERE routeId=".$routeId` ανακτά τη διαδρομή με το αντίστοιχο `Id`. Στη συνέχεια χωρίζει τη συμβολοσειρά της διαδρομής στα αντικείμενα που την αποτελούν με τη συνάρτηση `split`, τα μετατρέπει σε ακέραιους με τη μέθοδο `intval` και τα αποθηκεύει στον πίνακα `$object []`.

Για κάθε ένα αντικείμενο του πίνακα αυτού δημιουργεί ένα Submit Button το οποίο όταν πατηθεί ανοίγει σε ξεχωριστό παράθυρο τη σελίδα `Objectinfo` περνώντας στην παράμετρο `idValue` του `url` το `Id` του αντικειμένου που επιλέχθηκε, στην παράμετρο `arr` τον πίνακα `object[]`, με τη βοήθεια των μεθόδων `serialize` και `urlencode` της PHP, και τέλος στην παράμετρο `place` τη θέση του συγκεκριμένου

αντικειμένου μάθησης στον πίνακα `object[]`. Οι παράμετροι αυτές αποσκοπούν στην αυτονομία του επόμενου αρχείου, με τον τρόπο που θα δούμε αμέσως.

### 6.3.3 Objectinfo.php

Η συνάρτηση που περιέχει τη λειτουργία του αρχείου είναι η `display($objId)`. Η τιμή `$objId`, ο πίνακας που περιλαμβάνει όλα τα αντικείμενα μάθησης της διαδρομής στην οποία ανήκει το αντικείμενο και η θέση του αντικειμένου σε αυτόν, λαμβάνονται με τη μέθοδο `GET` της `HTML` από το `url` με το οποίο οδήγησε στη σελίδα αυτή η προηγούμενη, με τον τρόπο που περιγράψαμε στο 6.3.2. Ειδικότερα όσον αφορά στον πίνακα `object[]`, η ανάκτησή του γίνεται με χρήση των συναρτήσεων `urldecode()` και `unserialize()` αντίστοιχα. Όπως είπαμε, η σελίδα απεικόνισης ανοίγει σε ξεχωριστό παράθυρο και τα στοιχεία αυτά ήταν απαραίτητα για να μπορεί ο μαθητής να περιηγηθεί με συνδέσμους «επόμενο» και «προηγούμενο» σε όλα τα αντικείμενα της διαδρομής που επέλεξε.

Συνδεόμαστε κατά τα γνωστά στη βάση και εκτελώντας το query `"SELECT syndesmos FROM photos WHERE idAntikeimenou=".$objId` ανακτάμε από τον πίνακα `photos` το `path` της φωτογραφίας του αντικειμένου με το αντίστοιχο `Id` και το αποθηκεύουμε στη μεταβλητή `$foto`.

Αυτό είναι μια συμβολοσειρά της μορφής `«My Disk (E)\TPA1\tp1240.tif»`, όπως ήταν αποθηκευμένο στο `server` του μουσείου. Τα αρχεία που έχουμε εμείς έχουν το ίδιο όνομα (στην περίπτωση μας `tp1240`) αλλά φυσικά διαφέρει ο υποκατάλογος (`directory`). Επίσης τα αρχεία `tif` δεν υποστηρίζονται από τους περισσότερους φυλλομετρητές (`browsers`) και γι' αυτό τα μετατρέψαμε σε `jpeg`. Με τη συνάρτηση `split` κρατάμε από τη συμβολοσειρά το όνομα του αρχείου και με τη συνάρτηση `strtok($foto,".")` κρατάμε το όνομα του αρχείου χωρίς την κατάληξη. Προσθέτοντας την κατάληξη `.jpeg` έχουμε το όνομα του αρχείου που θέλουμε, το οποίο είναι αποθηκευμένο στο φάκελο της ιστοσελίδας.

Αντίστοιχα, με το query `"SELECT perigrifi FROM information WHERE idAntikeimenou=".$objId` ανακτούμε από τον πίνακα `information` της βάσης `corgialen` το κείμενο που περιέχει τις πληροφορίες για το συγκεκριμένο αντικείμενο.

Τελικά ο κώδικας `HTML` απεικονίζει στη σελίδα τη φωτογραφία και το κείμενο που αντιστοιχούν στο αντικείμενο. Σε περίπτωση που για το συγκεκριμένο αντικείμενο δεν έχουμε στη διάθεσή μας κάποια εικόνα ή κάποια σχετική πληροφορία, εμφανίζεται σχετικό μήνυμα.

Στο τέλος της σελίδας, υπάρχουν 2 σύνδεσμοι, «προηγούμενο αντικείμενο» και «επόμενο αντικείμενο» αντίστοιχα, οι οποίοι αν πατηθούν οδηγούν στη σελίδα απεικόνισης του αντίστοιχου αντικειμένου στη σειρά της εκπαιδευτικής διαδρομής, περνώντας και πάλι μέσω του `url` τα απαιτούμενα στοιχεία που είδαμε.



## 6.4 ΔΙΑΓΩΝΙΣΜΑΤΑ

Τα αρχεία που περιγράφουμε στο κεφάλαιο αυτό είναι όσα σχετίζονται με το σχεδιασμό των διαγωνισμάτων πολλαπλής επιλογής από τους εκπαιδευτικούς, την αποθήκευσή τους στη βάση και την εκτέλεσή τους από τους εκπαιδευόμενους.

### 6.4.1 TestEditor.php

Το αρχείο αυτό δημιουργεί το πλαίσιο στο οποίο ο χρήστης (δάσκαλος) γράφει τις ερωτήσεις και τις πιθανές απαντήσεις.

Αρχικά συνδεόμαστε στη βάση και με την εκτέλεση των queries "SELECT \* FROM Questions WHERE queId = (SELECT MAX(queId) FROM Questions)" και "SELECT \* FROM tests WHERE testId = (SELECT MAX(testId) FROM tests)" βρίσκουμε τις τελευταίες καταχωρήσεις των πινάκων ώστε να βρούμε τα καινούργια Id που θα πρέπει να χρησιμοποιήσουμε για τις ερωτήσεις και το διαγώνισμα.

Στη συνέχεια με κώδικα HTML δημιουργούνται 2 text fields κι ένα submit button για την καταχώρηση του ονόματος και του αριθμού των ερωτήσεων που θα περιλαμβάνει το διαγώνισμα. Με το πάτημα του submit button τα περιεχόμενα των text fields καταχωρούνται στις μεταβλητές \$testName και \$questionsNum και δημιουργείται με την εντολή INSERT της SQL μια καινούργια εγγραφή στον πίνακα tests που περιέχει τα στοιχεία του διαγωνίσματος που ονομάσαμε.

Ανάλογα με την τιμή της μεταβλητής \$questionsNum, δημιουργούνται ισάριθμα πεδία, ένα για κάθε ερώτηση, που περιέχει ένα text area για την εγγραφή της ερώτησης, 4 μικρότερα για τις πιθανές απαντήσεις και ένα text field για την καταχώρηση του αριθμού της σωστής απάντησης. Δημιουργείται επίσης ένα ακόμη submit button, με το πάτημα του οποίου τα περιεχόμενα των πεδίων αυτών που αντιστοιχούν σε κάθε ερώτηση διαβάζονται με τη μέθοδο POST, αποθηκεύονται σε μεταβλητές και με την εκτέλεση της εντολής `mysql_query("INSERT INTO questions (question, answer1, answer2, answer3, answer4, correct, queId) VALUES ('.$question.', '.$ans1.', '.$ans2.', '.$ans3.', '.$ans4.', '.$corAnswer.', '.$queId.');")`, που εκτελεί το αντίστοιχο query, αποθηκεύονται σαν καινούργιες εγγραφές στον πίνακα questions.

Ύστερα εκτελείται για κάθε νέα ερώτηση το query "INSERT into testque VALUES ('.\$ourTestId.', '.\$queId.' ',\$testQueid.');" με το οποίο δημιουργείται η απαραίτητη εγγραφή στον πίνακα testque, που καθορίζει τις ερωτήσεις οι οποίες περιλαμβάνονται σε κάθε διαγώνισμα. Για την αναγνώριση και επεξεργασία και των ελληνικών χαρακτήρων, προηγουμένως έχει εκτελεστεί το query "SET NAMES 'utf8'", που επιτρέπει την κωδικοποίηση ελληνικών χαρακτήρων στον MySQL.

Τέλος, με την εντολή header() οδηγούμαστε στη σελίδα που περιγράφεται αμέσως μετά, όπου γίνεται η ανάθεση των διαγωνισμάτων στους μαθητές, περνώντας στο url το Id του καινούργιου διαγωνίσματος.

## 6.4.2 AssignTestToStudents.php

Το αρχείο αυτό είναι παρόμοιο με το AssignRoutesToStudents.php. Αυτή τη φορά η συνάρτηση *StudentSelection* είναι *StudentSelection(\$testId,\$userId)*. Η τιμή *\$testId* είναι που λαμβάνεται από το url και το *\$userId* είναι το Id του συνδεδεμένου χρήστη, δηλαδή του δασκάλου που έφτιαξε το διαγώνισμα. Συνδεόμαστε στη βάση *corgialen* και από τον πίνακα *teacherstudents* βρίσκουμε τους μαθητές που ανήκουν στην τάξη του δασκάλου. Δημιουργούνται και πάλι με κώδικα HTML τα checkboxes που αντιστοιχούν σε κάθε μαθητή, και με το πάτημα του submit button εισάγονται στον πίνακα *studenttests* καινούργιες εγγραφές, που αντιστοιχίζουν σε κάθε επιλεγμένο μαθητή το νέο διαγώνισμα.

## 6.4.3 TestToChoose.php

Αυτή είναι η σελίδα στην οποία οι μαθητές βλέπουν τα διαγωνίσματα που τους έχουν ανατεθεί και μπορούν να επιλέξουν κάποιο για να λύσουν. Ο κώδικας είναι παρόμοιος με αυτόν του αρχείου *Course.php*, όπου ο μαθητής επιλέγει τη διαδρομή που θέλει να ακολουθήσει από τις διαθέσιμες.

Βασική συνάρτηση είναι η *DisplayTestChoices(\$userId)*, όπου *userId* το Id του συνδεδεμένου μαθητή. Το πρόγραμμα συνδέεται στη βάση και επιλέγει από τον πίνακα *studenttests* τις εγγραφές που έχουν στο πεδίο *studentId* το Id του συνδεδεμένου μαθητή. Αποθηκεύει στον πίνακα *\$testId[]* τα *testId* των εγγραφών αυτών, στον πίνακα *\$StId[]* τα *studentTestId* (primary key) και στον πίνακα *\$done[]* τα πεδία *done*, τα οποία καθορίζουν για κάθε εγγραφή αν ο μαθητής έχει ξαναπροσπαθήσει να λύσει το διαγώνισμα σε προηγούμενη επίσκεψη. Στη συνέχεια ανακτά από τον πίνακα *tests* της βάσης τα ονόματα των διαγωνισμάτων και για κάθε ένα από αυτά δημιουργεί με κώδικα HTML ένα πεδίο που περιέχει ένα submit button και το όνομα του διαγωνίσματος, καθώς και την ένδειξη *DONE* αν ο μαθητής έχει ξαναλύσει το διαγώνισμα. Ανάλογα με το submit button που θα πατήσει ο μαθητής, με τη συνάρτηση *header("url")* της *php* οδηγείται στη σελίδα *Test to do*, περνώντας στην παράμετρο *idValue* του url την τιμή του Id του διαγωνίσματος που επιλέχθηκε και στην παράμετρο *StidValue* την τιμή του *studentTestId* της αντίστοιχης εγγραφής του πίνακα *studenttests*.

## 6.4.4 TestToDo.php

Το πρόγραμμα παίρνει από το url το Id του διαγωνίσματος και το *studentTestId*, συνδέεται στη βάση *corgialen* και ανακτά από τον πίνακα *tests* το όνομα και τον αριθμό των ερωτήσεων του. Στη συνέχεια, από τον πίνακα *testque* ανακτά τα Id των ερωτήσεων που αντιστοιχούν στις εγγραφές με *testId* το Id του επιλεγμένου διαγωνίσματος.

Για κάθε μια ερώτηση, το πρόγραμμα ανακτά από τον πίνακα *questions* τα κείμενα της ερώτησης και των 4 απαντήσεων, καθώς και τον αριθμό της σωστής απάντησης, και με κώδικα HTML δημιουργεί ένα πεδίο για κάθε ερώτηση, το οποίο περιέχει την ερώτηση, τις πιθανές απαντήσεις και ένα textfield για την εισαγωγή του αριθμού της σωστής απάντησης.

Στο τέλος της σελίδας υπάρχει ένα Submit Button. Όταν πατηθεί αυτό, το πρόγραμμα διαβάζει τον αριθμό της δοθείσας απάντησης για κάθε ερώτηση και τυπώνει για κάθε μία αντίστοιχο μήνυμα σωστού ή λάθους, συγκρίνοντάς το με τη σωστή απάντηση. Επίσης, υπολογίζει τον αριθμό των σωστών απαντήσεων και τον αποθηκεύει στο πεδίο DONE της εγγραφής με το ζητούμενο studentTestId, χρησιμοποιώντας την εντολή UPDATE της SQL.

## 6.5 Οι μαθητές μου

### 6.5.1 StudentsToTeachers.php

Το αρχείο αυτό διαχειρίζεται τη σελίδα στην οποία οι δάσκαλοι επιλέγουν τους μαθητές της τάξης τους. Η κεντρική συνάρτηση είναι η SelectStudents(\$userId), όπου userId το Id του τρέχοντος χρήστη.

Αρχικά το πρόγραμμα συνδέεται στη βάση corgialen και ανακτά από τον πίνακα teacherstudents τα studentId των εγγραφών που έχουν στο πεδίο teacherId το userId, δηλαδή τα Id των μαθητών που ανήκουν στην τάξη του συνδεδεμένου δασκάλου. Για κάθε ένα από αυτά αποθηκεύει σε έναν πίνακα \$isChecked[\$Id] την τιμή "checked", όπου Id το Id που αντιστοιχεί στο μαθητή.

Στη συνέχεια συνδέεται στη βάση του Joomla και ανακτά τα στοιχεία όλων των χρηστών που ανήκουν στην κατηγορία students. Για κάθε έναν από αυτούς δημιουργεί με κώδικα HTML ένα πεδίο το οποίο περιέχει το όνομα και το Id του μαθητή, ένα checkbox και 2 συνδέσμους, tests και routes, οι οποίοι με εντολές href της HTML οδηγούν στις σελίδες που περιγράφονται στα αμέσως επόμενα υποκεφάλαια και εμφανίζουν τις διαδρομές και τα διαγωνίσματα που αντιστοιχούν σε κάθε μαθητή. Στον ορισμό κάθε checkbox περιλαμβάνεται η τιμή \$isChecked[\$Id], οπότε στα πεδία που αντιστοιχούν στους μαθητές που ήδη ανήκουν στην τάξη του δασκάλου τα checkboxes εμφανίζονται καταρχάς επιλεγμένα.

Στο τέλος της σελίδας δημιουργείται ένα Submit Button με το όνομα submit students. Όταν αυτό πατηθεί, το πρόγραμμα και επιτελεί τις εξής λειτουργίες: Αρχικά διαγράφει όλες τις υπάρχουσες εγγραφές του πίνακα teacherstudents που αντιστοιχούν στον τρέχοντα χρήστη. Στη συνέχεια, για κάθε επιλεγμένο checkbox, προσθέτει στον πίνακα αυτόν μια εγγραφή που συνδέει τον τρέχοντα χρήστη με τον αντίστοιχο μαθητή.

### 6.5.2 StudentRoutes.php

Το αρχείο αυτό διαχειρίζεται τη σελίδα στην οποία ο δάσκαλος μπορεί να δει τις διαδρομές που έχουν ανατεθεί στο μαθητή που επέλεξε προηγουμένως, πατώντας το σύνδεσμο routes που αντιστοιχούσε στο πεδίο του. Η εντολή href που οδήγησε στη σελίδα αυτή περνά στην παράμετρο idValue το Id και στην παράμετρο name το όνομα του επιλεγμένου μαθητή.

Με τη μέθοδο GET παίρνουμε τις τιμές αυτές και τυπώνουμε το όνομα του μαθητή. Η κεντρική συνάρτηση που χρησιμοποιείται είναι η *ViewRouteStudent(\$stId)*, όπου *stId* το Id του επιλεγμένου μαθητή. Το πρόγραμμα συνδέεται στη βάση *corgialen* και ανακτά από τον πίνακα *studentroutes* τα Id των διαδρομών που έχουν ανατεθεί στο μαθητή, καθώς και τα πεδία *done* των εγγραφών αυτών. Για κάθε μια διαδρομή, ανακτά από τον πίνακα *routes* το όνομά της και δημιουργεί ένα πεδίο που περιέχει το όνομα της διαδρομής και την ένδειξη *done*, σε περίπτωση που το αντίστοιχο πεδίο είναι 1.

### 6.5.3 StudentTests.php

Αντίστοιχα λειτουργεί και ο σύνδεσμος *tests* κάθε πεδίου της σελίδας «Οι μαθητές μου». Ο δάσκαλος οδηγείται στη σελίδα που βλέπει τα διαγωνίσματα που έχουν ανατεθεί στον επιλεγμένο μαθητή και την επίδοσή του αν έχει προσπαθήσει να τα λύσει. Η εντολή *href* που οδήγησε στη σελίδα αυτή πέρασε και πάλι στην παράμετρο *idValue* το Id και στην παράμετρο *name* το όνομα του επιλεγμένου μαθητή.

Με τη μέθοδο GET παίρνουμε τις τιμές αυτές και τυπώνουμε το όνομα του μαθητή. Η κεντρική συνάρτηση που χρησιμοποιείται αυτή τη φορά είναι η *ViewTestStudent(\$stId)*, όπου *stId* το Id του επιλεγμένου μαθητή. Το πρόγραμμα συνδέεται στη βάση *corgialen* και ανακτά από τον πίνακα *studenttests* τα Id των διαγωνισμάτων που έχουν ανατεθεί στο μαθητή, καθώς και τα πεδία *done* των εγγραφών αυτών. Για κάθε ένα από αυτά τα διαγωνίσματα ανακτά από τον πίνακα *test* το όνομα και τον αριθμό των ερωτήσεων του και δημιουργεί ένα πεδίο που περιέχει το όνομα του διαγωνίσματος, μια ένδειξη με τον αριθμό των ερωτήσεων που περιέχει, καθώς και το πεδίο *done* της αντίστοιχης εγγραφής, δηλαδή το βαθμό του μαθητή στο συγκεκριμένο διαγώνισμα. Αν ο μαθητής δεν έχει προσπαθήσει να το λύσει το πεδίο *done* είναι προεπιλεγμένα 0. <sup>[15]</sup>

## 7 ΕΠΙΛΟΓΟΣ

### 7.1 Σύνοψη

Στην εργασία αυτή αναπτύξαμε ένα Σύστημα Ηλεκτρονικής Μάθησης (ΣΗΜ) το οποίο αξιοποιεί τα στοιχεία πολιτιστικής κληρονομιάς που είναι αποθηκευμένα σε μια υπάρχουσα βάση δεδομένων ενός πραγματικού μουσείου, του Κοργιαλένειου Μουσείου της Κεφαλονιάς. Ο στόχος μας ήταν διπλός: η ανάδειξη των στοιχείων αυτών ως αντικειμένων μάθησης και των δυνατοτήτων σύνδεσής τους με την εκπαιδευτική διαδικασία, και ταυτόχρονα η μελέτη των προδιαγραφών που πρέπει να καλύπτει ο σχεδιασμός ενός Συστήματος Ηλεκτρονικής Μάθησης τέτοιου σκοπού.

Το ψηφιοποιημένο υλικό και οι πληροφορίες που αποθηκεύονται στις βάσεις δεδομένων μουσείων και βιβλιοθηκών μπορούν να αξιοποιηθούν από εκπαιδευτικούς φορείς με τη βοήθεια των ΣΗΜ. Ένα σύστημα σαν και αυτό που αναπτύξαμε μπορεί να χρησιμοποιηθεί σε τοπικό επίπεδο στο πλαίσιο μαθημάτων όπως η ιστορία και η γεωγραφία και να βοηθήσει στην άντληση μαθησιακού υλικού από την άμεση εμπειρία των μαθητών.

Αλλά και σε ευρύτερο επίπεδο, ένα τέτοιο ΣΗΜ θα έδινε τη δυνατότητα στους μαθητές να αποκτήσουν πρόσβαση σε εκθέματα και πληροφορίες που θα χρειαζόταν να ταξιδέψουν για να δουν και θα μπορούσε να προσφέρει ολοκληρωμένη γνώση για το μάθημα της Ιστορίας. Για παράδειγμα, ένα Σύστημα Ηλεκτρονικής Μάθησης που θα αξιοποιούσε μια παρόμοια βάση δεδομένων με εκθέματα του Αρχαιολογικού Μουσείου της Αθήνας, θα έδινε στους μαθητές –κυρίως της επαρχίας– τη δυνατότητα να δουν τα εκθέματα του μουσείου και να μάθουν ιστορικές πληροφορίες που συνδέονται με αυτά, παράλληλα με την παράδοση στο πλαίσιο του σχολικού μαθήματος. Με τον τρόπο αυτό θα συνέβαλλε στην αφομοίωση της γνώσης που θα λάμβαναν από τη διδασκαλία, χρησιμοποιώντας τη δύναμη της εικόνας, αλλά και λόγω του χαρακτήρα που δίνει στη μάθηση η χρήση ΤΠΕ.

Το σύστημά μας παρέχει ένα πλαίσιο λειτουργιών που θα μπορούσε να χρησιμοποιηθεί από τους εκπαιδευτικούς για να δώσουν στους μαθητές τη δυνατότητα να γνωρίσουν αρκετά στοιχεία της σύγχρονης ιστορίας και του πολιτισμού του νησιού της Κεφαλονιάς, αλλά και των άλλων νησιών του Ιονίου που έχουν παρόμοιες εμπειρίες, ανεξάρτητα από τον εκπαιδευτικό χαρακτήρα του. Αποτελεί όμως ταυτόχρονα ένα παράδειγμα του τρόπου με τον οποίο μπορούμε να αξιοποιήσουμε εκπαιδευτικά κάθε τεκμηριωμένη βάση δεδομένων τέτοιου είδους, αφού με συγκεκριμένες μετατροπές μπορεί να συμπεριλάβει αντίστοιχα δεδομένα διαφορετικών βάσεων. Επίσης, μπορεί εύκολα να τροποποιηθεί και στο επίπεδο των εκπαιδευτικών λειτουργιών ανάλογα με τους σκοπούς της μάθησης και τις προτιμήσεις των προσώπων που πρόκειται να το χρησιμοποιήσουν.

Επιπλέον, μπορεί να χρησιμοποιηθεί για την ανάπτυξη διαθεματικών εργασιών (projects) με την πληροφορική ως ένα πυλώνα. Θέματα πληροφορικής που θα είχαν ενδιαφέρον είναι η ίδια η λογική ενός λογισμικού διδασκαλίας, το θέμα των πολιτιστικών βάσεων δεδομένων, η συζήτηση περί ανοιχτού λογισμικού και δημόσιων δεδομένων κλπ. Αυτά θα μπορούσαν να συνδυαστούν με θέματα ιστορίας, γεωγραφίας, αισθητικής αλλά και θέματα χρήσης υλικών, εξέλιξης της τεχνολογίας κλπ, τα οποία εύκολα αναδεικνύονται από το υλικό που περιέχεται στις πολιτιστικές βάσεις δεδομένων.

## 7.2 Προεκτάσεις

Στην εφαρμογή που αναπτύξαμε προσπαθήσαμε να τηρήσουμε όσο το δυνατόν τις προδιαγραφές του σχεδιασμού Ηλεκτρονικής Μάθησης, όπως τις έχουμε περιγράψει στο κεφάλαιο 2.3. Η λειτουργία δημιουργίας των εκπαιδευτικών διαδρομών, που αποτελεί και τη βασική λειτουργία του συστήματος, χρησιμοποιεί σε απλή μορφή τα 3 στάδια στα οποία γίνεται ο καθορισμός των στοιχείων ενός προβλήματος Ηλεκτρονικής Μάθησης.

Ο εκπαιδευτικός καθορίζει μέσα από ένα πλαίσιο επιλογών το αντικείμενο γνώσης, που στην περίπτωση μας είναι η διάζευξη κάποιων εκ των διαθέσιμων κατηγοριών των εκθεμάτων. Η επιλογή του αυτή είναι ένας ορισμός του **επιπέδου αντίληψης** (conceptual view).

Ο διαχωρισμός των κατηγοριών προέκυψε μετά από μελέτη των αντικειμένων που είχαμε στη διάθεσή μας και οι κατηγορίες στις οποίες καταλήξαμε ελέγχονται ακόμα και στο επίπεδο του συγκεκριμένου συστήματος. Σε κάθε περίπτωση οι επιλογές αυτές θα πρέπει να αξιολογούνται από «τεκμηριωτές» (με την έννοια που χρησιμοποιούμε στο σύστημά μας) εξειδικευμένους σε θέματα ιστορίας και εκπαίδευσης.

Ειδικότερα, όμως, στην περίπτωση που κάποιο αντίστοιχο ΣΗΜ στοχεύει στην αξιοποίηση βάσεων δεδομένων με πολύ περισσότερα και διαφορετικών τύπων στοιχεία, ένας τέτοιος διαχωρισμός κατηγοριών φυσικά δε θα ήταν αρκετός. Το ΣΗΜ θα έπρεπε να προβλέπει έναν πολυεπίπεδο διαχωρισμό με διαφορετικά κριτήρια, όπως χρονολογία, τοποθεσία, είδος εκθεμάτων, πεδίο γνώσης κλπ. Ο εκπαιδευτικός θα πρέπει να έχει τη δυνατότητα να καθορίζει συγκεκριμένα και αποτελεσματικά το επίπεδο αντίληψης, ώστε να μπορεί να οργανώνει το μάθημα ανάλογα με την ύλη που θέλει να καλύψει.

Όσον αφορά στο **επίπεδο εργασιών** (task view), το σύστημα χρησιμοποιεί ουσιαστικά ένα είδος αντικειμένων μάθησης, που αποτελείται κάθε φορά από εικόνα και πληροφορία σχετικές με κάποιο έκθεμα του μουσείου. Για το λόγο αυτό μπορούμε να πούμε ότι το επίπεδο εργασιών είναι στην περίπτωση μας προκαθορισμένο από τα περιεχόμενα της βάσης δεδομένων.

Σε ένα ολοκληρωμένο ΣΗΜ είναι αναγκαία η πρόβλεψη διαφορετικών ειδών αντικειμένων μάθησης. Το πρόβλημα σχεδιασμού πρέπει να περιλαμβάνει μεγάλη ποικιλία εργασιών, με σκοπό την ανάθεσή τους στους μαθητές ανάλογα με το προφίλ των τελευταίων, και τα αντικείμενα μάθησης δε χρειάζεται να είναι προσβάσιμα μέσω του ΣΗΜ. Ο εκπαιδευτικός πρέπει να είναι σε θέση να καθορίζει εργασίες όπως η ανάγνωση ενός βιβλίου ή ενός άρθρου, η επίσκεψη σε ένα μουσείο ή η παρακολούθηση ενός ντοκυμαντέρ. Και το ΣΗΜ να παράγει ένα σχέδιο λύσης που να περιλαμβάνει το συνδυασμό τέτοιου είδους εργασιών ανάλογα με τις προτιμήσεις και τις ανάγκες κάθε μαθητή.

Γι' αυτό το λόγο θα ήταν εξαιρετικά αποδοτική η χρήση ενός μηχανισμού ανατροφοδότησης της εκπαιδευτικής λειτουργίας του συστήματος στο **επίπεδο προσαρμογής** (adaptation view). Στην εφαρμογή μας, το ρόλο του μηχανισμού αυτού έχει αποκλειστικά ο εκπαιδευτικός. Αυτός είναι που παρακολουθεί την πρόοδο του μαθήματος και των εκπαιδευομένων και πρέπει να είναι σε θέση να χρησιμοποιεί τα συμπεράσματά του κάθε φορά που παράγει εκπαιδευτικές διαδρομές για τους μαθητές του. Και σε αυτό αποσκοπούν οι υπόλοιπες εκπαιδευτικές λειτουργίες του συστήματος, όπως η δημιουργία διαγωνισμάτων και η παρακολούθηση των εργασιών και των επιδόσεων των μαθητών. Τα

δεδομένα που παράγονται από τις λειτουργίες αυτές και αποθηκεύονται στη βάση, θα μπορούσαν να αποτελούν επιπλέον στοιχεία της περιγραφής του προβλήματος σχεδιασμού και να αξιοποιούνται από το ΣΗΜ για την παραγωγή σχεδίων λύσης που θα αποσκοπούσαν στην πλήρη εξατομίκευση των εκπαιδευτικών διαδρομών, λειτουργώντας ανεξάρτητα για κάθε μαθητή. Για την αξιοποίηση των στοιχείων αυτών θα ήταν φυσικά απαραίτητη η χρήση πολύπλοκων τεχνικών Σχεδιασμού Τεχνητής Νοημοσύνης (AI Planning) και ο καθορισμός πολύ αυστηρών προδιαγραφών μοντελοποίησης των δεδομένων.

Ο καθορισμός των στοιχείων του επιπέδου προσαρμογής στην παραγωγή των εκπαιδευτικών διαδρομών της εφαρμογής μας γίνεται με την αξιοποίηση των μαθησιακών χαρακτηριστικών. Η επιλογή των υποψήφιων αντικειμένων γίνεται με βάση τα όρια που θέτει ο χρήστης στις τιμές των μαθησιακών χαρακτηριστικών, ανάλογα με τους μαθητές στους οποίους έχει σκοπό να αναθέσει τη διαδρομή. Επίσης, στην εύρεση των σχεδίων λύσης από το σχεδιαστή LPG, σημαντικό ρόλο έχουν οι βαρύτητες που επιλέγει ο χρήστης για κάθε μαθησιακό χαρακτηριστικό.

Ο τρόπος με τον οποίο η εφαρμογή αξιοποιεί τα μαθησιακά χαρακτηριστικά των αντικειμένων μάθησης και τα δεδομένα της βάσης του μουσείου σίγουρα επιδέχεται πολλές βελτιώσεις. Η περιγραφή του προβλήματος μπορεί να γίνει με πολλούς διαφορετικούς τρόπους, ανάλογα με τα στοιχεία που έχει στη διάθεσή του το ΣΗΜ. Άλλωστε, ακόμη και τα ίδια τα μαθησιακά χαρακτηριστικά που χρησιμοποιούμε και ο τρόπος αξιολόγησής τους, είναι απαραίτητο να έχουν προταθεί από ειδικούς.

Ειδικότερα, τα αρχεία PDDL μπορούν να δημιουργηθούν με βάση διάφορες μεθόδους και διαφορετικές λογικές ανάλογα με το σκοπό της εκπαιδευτικής λειτουργίας που επιτελούν, κάτι που θα τα επηρέαζε τόσο σε επίπεδο δομής όσο και σε επίπεδο στόχων. Ακόμη και στον τρόπο με τον οποίο διαμορφώνονται στην εφαρμογή μας υπάρχουν στοιχεία που μπορούν να διαφοροποιηθούν.

Το χαρακτηριστικότερο παράδειγμα είναι η χρήση των επιπέδων. Ο λόγος που καταλήξαμε στην ομαδοποίηση των αντικειμένων σε επίπεδα ανάλογα με τη χρονολογία τους ήταν η προσπάθεια ανάδειξης της έννοιας της ιστορικής περιόδου. Λόγω της άνισης κατανομής των εκθεμάτων χρονολογικά, αλλά και της έλλειψης πληροφοριών σε πολλές περιπτώσεις, ο μόνος τρόπος με τον οποίο μπορούσαμε να επιτύχουμε κάτι τέτοιο ήταν να χρησιμοποιήσουμε το ημερολογιακό έτος ως σημείο αναφοράς. Σε αυτό οφείλεται και ο περιορισμός της αναγκαστικής επιλογής αντικειμένων μάθησης από κάθε επίπεδο, που όπως είδαμε δημιούργησε αρκετές δυσκολίες στην εύρεση του σχεδίου λύσης από το σχεδιαστή. Αντιμετωπίζοντας τις χρονολογίες ως διαφορετικές ιστορικές περιόδους, που πρέπει να εξετασθούν ξεχωριστά, καταλήξαμε σε αυτήν την πρόταση περιγραφής του προβλήματος σχεδιασμού, η οποία θα λειτουργούσε πολύ πιο αποτελεσματικά στην περίπτωση ο εκπαιδευτικός είχε τη δυνατότητα να ορίσει εκείνος τις ιστορικές περιόδους και το διαχωρισμό των αντικειμένων μάθησης σε αυτές, αν βέβαια το ΣΗΜ επεξεργαζόταν περισσότερα και διαφορετικά κατανεμημένα αντικείμενα μάθησης.

Τέλος, το σύνολο των εκπαιδευτικών λειτουργιών του ΣΗΜ θα μπορούσε να παρέχει περισσότερες ευκολίες στους εκπαιδευτικούς. Ένα ολοκληρωμένο ΣΗΜ πρέπει να καλύπτει λειτουργίες όπως η ανάθεση και υποβολή εργασιών, η καταχώρηση σημειώσεων και η οργάνωση και παρουσίαση της πορείας του μαθήματος, στοιχεία τα οποία δεν περιλαμβάνονται στο σύστημά μας.

## 8 ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας, Η. Σακελλαρίου. «Τεχνητή Νοημοσύνη – Γ Έκδοση» (2006)
- [2] <http://en.wikipedia.org/>
- [3] Stuart Russel, Peter Norvig. “Artificial Intelligence: A modern approach” (2009)
- [4] Eva Onaindia, Antonio Garrido, Oscar Sapena. ” Automated planning for personalised course composition” (2009)
- [5] Eva Onaindia, Antonio Garrido, Oscar Sapena. ” LRNPlanner: Planning Personalized and Contextualized E-learning Routes” (2008)
- [6] Alfonso Gerevini, Ivan Serina. “LPG: A planner based on Local Search for Planning Graphs with Action Costs” (2002)
- [7] Alfonso Gerevini, Ivan Serina, Alessandro Saetti, Paolo Toninelli. “Planning in PDDL 2.2 domains with LPG-TD” (2004)
- [8] <http://zeus.ing.unibs.it/lpg/>
- [9] Alfonso Gerevini, Ivan Serina, Alessandro Saetti, Paolo Toninelli. “Planning with numerical expressions in LPG” (2004)
- [10] <http://users.cecs.anu.edu.au/~patrik/pddlman/writing.html>
- [11] Avrim Blum, Merrick Furst. “Fast Planning Through Planning Graph Analysis” (1995)
- [12] Lluvia Morales, Antonio Garrido, Ivan Serina. “Planning and Execution in a Personalised E-learning Setting” (2011)
- [13] Lluvia Morales, Antonio Garrido, Ivan Serina. “Applying Case-Based Planning to Personalized E-learning” (2011)
- [14] <http://docs.joomla.org/>
- [15] <http://www.php.net/>
- [16] <http://www.cidoc-crm.org/>