



ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ  
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ

**Μεταπτυχιακή Εργασία**

**Σύγκριση scripting γλωσσών προγραμματισμού  
και διερεύνηση της καταλληλότητας τους για  
εργασίες συντήρησης λογισμικού**

**Φοιτητής: Πέτρου Βασίλειος Α.Μ. 21/09**

**Επιβλέπων Καθηγητής: κος Αλέξανδρος Χατζηγεωργίου**

**Θεσσαλονίκη 2011**

Copyright © Πέτρου Βασίλειος, 2011  
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Η έγκριση της μεταπτυχιακής εργασίας από το Τμήμα Εφαρμοσμένης Πληροφορικής του Πανεπιστημίου Μακεδονίας δεν υποδηλώνει απαραίτητως και απόδοχη των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

## Περίληψη

Σκοπός της διπλωματικής εργασίας είναι να αναλύσει και να συγκρίνει δύο scripting γλώσσες προγραμματισμού, την πολύ διαδεδομένη PHP και την νεοεισερχόμενη Ruby. Η σύγκριση αυτή μπορεί να οδηγήσει και σε βαθύτερα συμπεράσματα όπως για το ποια γλώσσα είναι ακαδημαϊκά και επαγγελματικά καταλληλότερη.

Αρχικά θα υλοποιηθεί μια βασική μορφή ενός ηλεκτρονικού περιβάλλοντος διαχείρισης πτυχιακών εργασιών με βασικές λειτουργίες και στις 2 γλώσσες. Στη συνέχεια θα προταθούν σενάρια ανάπτυξης της λειτουργικότητας και μαζί με την υλοποίηση αυτών θα γίνει παράλληλα καταγραφή των βημάτων υλοποίησης και όλων των σημαντικών στοιχείων που σχετίζονται με τη διαδικασία για να προκύψουν τα κατάλληλα συμπεράσματα.

Λέξεις Κλειδιά : «Ruby, Ruby on Rails, PHP, Apache, MySQL, Model-View-Controller, Box-Plot»

## Summary

The main purpose of this thesis is to analyze and compare two scripting languages, the very popular PHP and the “newcomer” Ruby. This Comparison can guide to deeper conclusions. eg. Which language is more appropriate for academic and business purpose.

First of all, an initial interface of an electronic environment in management and publication of thesis will be developed with basic functions in both languages separately. Then, there will be some proposals upgrading the functionality of the environment. During programming, all the development steps and the significant details will be recorded to guide us to conclusions.

Keywords : «Ruby, Ruby on Rails, PHP, Apache, MySQL, Model-View-Controler, Box-Plot»

## Πίνακας Περιεχομένων

Ευχαριστίες .....	9
1. Περιγραφή και Ανάλυση Προβλήματος .....	10
1.1 Περιγραφή και Ανάλυση Προβλήματος .....	10
1.1.1 Το γενικό πρόβλημα .....	10
1.1.2 Το ειδικό πρόβλημα .....	10
1.2 Αντιμετώπιση του προβλήματος .....	11
1.2.1 Τρέχουσα κατάσταση .....	11
1.2.2 Προσέγγιση λύσης .....	11
2. Ανάπτυξη με PHP .....	14
2.1 Λογισμικά Ανοικτού Κώδικα (Open Source Code) .....	14
2.1.1 Τι είναι το Λογισμικό Ανοικτού Κώδικα (Open Source) ; .....	14
2.1.2 Τα πλεονεκτήματα του Open Source .....	14
2.1.3 Τα μειονεκτήματα του Open Source .....	16
2.2 PHP .....	17
2.2.1 Τι είναι η PHP ; .....	17
2.2.2 Τι μπορεί να κάνει η PHP ; .....	18
2.3 HTML .....	20
2.3.1 Τι είναι η HTML ; .....	20
2.3.2 Τι είναι οι ετικέτες της HTML ; .....	20
2.4 APACHE .....	21
2.4.1 Τι είναι ο Apache ; .....	21
2.4.2 Πως δημιουργήθηκε ο Apache ; .....	22
2.4.3 Χαρακτηριστικά του Apache .....	23
2.5 MySQL .....	24
2.5.1 Τι είναι η MySQL ; .....	24
2.5.2 Ιστορία της MySQL .....	25
2.6 Λόγοι χρησιμοποίησης της PHP και της MySQL .....	25
2.6.1 Μερικά από τα πλεονεκτήματα της PHP .....	26
2.6.2 Μερικά από τα πλεονεκτήματα της MySQL .....	27
3. Ανάπτυξη με RUBY .....	29
3.1 Ιστορία .....	29
3.1.1 Ετυμολογία του ονόματος "Ruby" .....	29
3.1.2 Αρχική δημοσιοποίηση .....	30

3.1.3 Ruby 1.0 .....	30
3.1.4 Ruby on Rails.....	30
3.1.5 Ruby 1.9.1 .....	33
3.2 Φιλοσοφία .....	33
3.3 Χαρακτηριστικά .....	34
3.4 Σημασιολογία .....	35
3.5 Σύνταξη .....	36
3.6 Διαφορές στη συμπεριφορά σε σχέση με άλλες γλώσσες .....	37
3.7 Αλληλεπίδραση.....	38
3.8 Υλοποιήσεις .....	38
3.9 Αποθήκες κώδικα και βιβλιοθήκες .....	39
4. Περιγραφή της Βάσης Δεδομένων .....	40
4.1 Οι πίνακες γενικά.....	40
4.2 Οι πίνακες αναλυτικά .....	40
4.3 Η βάση περιγραφικά .....	43
4.4 Διάγραμμα Οντοτήτων Συσχετίσεων της βάσης δεδομένων.....	49
5. Δομή του ιστοτόπου .....	50
5.1 Σύντομη περιγραφή της δομής του ιστοτόπου.....	50
5.1.1 Κεντρική Σελίδα .....	50
5.1.2 Επιλογές Φοιτητή .....	52
5.1.3 Επιλογές Καθηγητή.....	54
6. Σύγκριση Scripting Γλωσσών Προγραμματισμού.....	56
6.1 Συγκριτικά Διαγράμματα.....	56
6.2 Συγκριτικοί Πίνακες .....	58
6.3 Συγκριτικοί Αλγόριθμοι .....	61
6.3.1 Πρόγραμμα “Hallo World” .....	61
6.3.2 Αλγόριθμος “Binnary Trees” .....	62
6.3.3 Αλγόριθμος “Recursive” .....	65
6.3.4 Αλγόριθμος “FANNKUCH” .....	68
7. Ανάλυση Κώδικα Βασικής Λειτουργικότητας σε PHP και Ruby .....	72
7.1 Σενάριο : Αυθεντικοποίηση Χρηστών.....	72
7.1.1 Σενάριο : Αυθεντικοποίηση Χρηστών με PHP .....	72
7.1.2 Σενάριο : Αυθεντικοποίηση Χρηστών με Ruby .....	75
7.2 Σενάριο : Δημιουργία/Επεξεργασία χρηστών .....	82

7.2.1 Σενάριο : Δημιουργία/Επεξεργασία χρηστών με PHP .....	82
7.2.2 Σενάριο : Δημιουργία/Επεξεργασία χρηστών με Ruby .....	86
7.3 Σενάριο : Δημιουργία/Επεξεργασία πτυχιακών .....	95
7.3.1 Σενάριο : Δημιουργία/Επεξεργασία πτυχιακών με PHP .....	95
7.3.2 Σενάριο : Δημιουργία/Επεξεργασία πτυχιακών με Ruby.....	107
8. Ανάπτυξη & Σύγκριση Σεναρίων Επέκτασης σε PHP και Ruby.....	117
8.1 Σενάριο : Πλήθος Πτυχιακών Καθηγητή ανά κατηγορία και συνολικά .....	117
8.1.1 Σενάριο : Πλήθος Πτυχιακών Καθηγητή ανά κατηγορία και συνολικά με PHP .....	117
8.1.2 Σενάριο : Πλήθος Πτυχιακών Καθηγητή ανά κατηγορία και συνολικά με Ruby .....	118
8.2 Σενάριο : Μέσος Όρος Ημερών Εκπόνησης Πτυχιακών Καθηγητή .....	119
8.2.1 Σενάριο : Μέσος Όρος Ημερών Εκπόνησης Πτυχιακών Καθηγητή με PHP .....	119
8.2.2 Σενάριο : Μέσος Όρος Ημερών Εκπόνησης Πτυχιακών Καθηγητή με RUBY .....	120
8.3 Σενάριο : Πλήθος Πτυχιακών Όλων των Καθηγητών ανά κατηγορία και συνολικά .....	121
8.3.1 Σενάριο : Πλήθος Πτυχιακών Όλων των Καθηγητών ανά κατηγορία και συνολικά με PHP.....	121
8.3.2 Σενάριο : Πλήθος Πτυχιακών Όλων των Καθηγητών ανά κατηγορία και συνολικά με RUBY .....	122
8.4 Σενάριο : Μέσος Όρος Ημερών Εκπόνησης Πτυχιακών Όλων των Καθηγητών .....	123
8.4.1 Σενάριο : Μέσος Όρος Ημερών Εκπόνησης Πτυχιακών Όλων των Καθηγητών με PHP .....	123
8.4.2 Σενάριο : Μέσος Όρος Ημερών Εκπόνησης Πτυχιακών Όλων των Καθηγητών με RUBY.....	124
8.5 Σενάριο : Ορισμός Ποσοστού Εξέλιξης από τον Φοιτητή για την Εργασία που εκπονεί.....	125
8.5.1 Σενάριο : Ορισμός Ποσοστού Εξέλιξης από τον Φοιτητή για την Εργασία που εκπονεί με PHP .....	125
8.5.2 Σενάριο : Ορισμός Ποσοστού Εξέλιξης από τον Φοιτητή για την Εργασία που εκπονεί με RUBY.....	129
8.6 Συγκριτικός Πίνακας PHP - RUBY.....	132
9. ΣΥΜΠΕΡΑΣΜΑΤΑ.....	133
Παράρτημα Α : ScreenShots Ιστοσελίδας .....	136
Α.1 Οδηγίες χρήσης (πρώτα βήματα) .....	136

Σύγκριση scripting γλωσσών προγραμματισμού  
και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

---

A.1.1 Αρχική σελίδα.....	136
A.2 Οδηγίες χρήσης (φοιτητής) .....	137
A.2.1 Είσοδος φοιτητών .....	137
A.2.2 Λίστα Πτυχιακών .....	138
A.3 Οδηγίες χρήσης (καθηγητής) .....	143
A.3.1 Είσοδος καθηγητή .....	143
A.3.2 Οι πτυχιακές μου.....	144
A.4 Οδηγίες χρήσης (διαχειριστής) .....	160
A.4.1 Διαχείριση usernames.....	160
Βιβλιογραφία.....	162
Ιστοσελίδες .....	165



## Ευχαριστίες

Για την επίτευξη της παρούσας διπλωματικής εργασίας θα ήθελα να ευχαριστήσω τον υπεύθυνο καθηγητή μου κύριο Αλέξανδρο Χατζηγεωργίου για την υποστήριξη, την υπομονή, την όλη καθοδήγηση του και την πάντοτε άμεση ανταπόκριση του όσες φορές κι αν τον χρειάστηκα κατά τη διάρκεια της εκπόνησης.

Επίσης θα ήθελα να ευχαριστήσω τους γονείς μου και την αδερφή μου για την αγάπη τους, τη στήριξη και την εμπιστοσύνη τους σε κάθε μου βήμα.

Τέλος θα ήθελα να ευχαριστήσω τους φίλους μου, για τη κατανόηση και τη συμπαράσταση τους. Ειδικά όμως να ευχαριστήσω τον πολύ αγαπητό μου φίλο, συμφοιτητή και συνάδελφο Λούπο Κωνσταντίνο για την πολύτιμη βοήθεια του στο κομμάτι της Ruby.

## 1. Περιγραφή και Ανάλυση Προβλήματος

### 1.1 Περιγραφή και Ανάλυση Προβλήματος

#### 1.1.1 Το γενικό πρόβλημα

Τα τελευταία χρόνια ολοένα και περισσότεροι απόφοιτοι των Λυκείων της χώρας, γίνονται δεκτοί στα ανώτατα εκπαιδευτικά ιδρύματα (Πανεπιστήμια και Τ.Ε.Ι.). Η έγκυρη, σαφής και εφ' όλης της ύλης ενημέρωση, σ' όλους τους πρωτοετείς φοιτητές αλλά και σ' αυτούς που είναι σε μεγαλύτερα εξάμηνα, είναι αναγκαία για την καλύτερη γενικότερα, λειτουργία του πανεπιστημίου. Είναι συχνό φαινόμενο, φοιτητές να έρχονται αντιμέτωποι με το άγνωστο και να μην μπορούν να εντοπίσουν μια πηγή έγκυρης και ταχείας ενημέρωσης. Παρατηρείται συχνά, μεγάλος συνωστισμός στις γραμματείες από φοιτητές που θα ήθελαν μια απλή πληροφορία.

#### 1.1.2 Το ειδικό πρόβλημα

Μια από τις πληροφορίες που ζητούν νέοι φοιτητές αλλά και φοιτητές μεγάλου εξαμήνου αφορά τις **πτυχιακές εργασίες**. Είναι παρά πολύ συχνό το φαινόμενο, φοιτητές να φθάνουν σε μεγάλα εξάμηνα και να μην έχουν ιδέα για το τι είναι η πτυχιακή εργασία. Το βασικότερο πρόβλημα όμως δεν είναι αυτό. Είναι η όλη διαδικασία που χρειάζεται μέχρις ότου ο σπουδαστής να εντοπίσει κάποιον καθηγητή, να βρει ενδιαφέρον θέμα και να το εκπονήσει. Αυτή η διαδικασία είναι δύσκολη, επίπονη και ιδιαίτερα χρονοβόρα για τον φοιτητή. Τα προβλήματα όμως δεν σταματούν εδώ. Ακόμα και όταν τελικά ο φοιτητής καταφέρει να βρει καθηγητή και θέμα πτυχιακής για εκπόνηση, ακόμα και τότε μπορεί να του παρουσιαστεί ένα πολύ βασικό πρόβλημα, το οποίο αφορά στη μεταξύ τους επικοινωνία (καθηγητή - φοιτητή) καθ' όλη τη διάρκεια της εκπόνησης. Πολλοί φοιτητές απογοητεύονται και παρατάνε τις πτυχιακές τους, λόγω της έλλειψης συχνής επικοινωνίας με τους υπεύθυνους καθηγητές. Σαν συμπέρασμα θα 'λεγε κανείς ότι είναι παράλογο, τμήματα όπως αυτό της Πληροφορικής να αντιμετωπίζουν τέτοια προβλήματα, ενώ είναι προφανές ότι ασχολείται με νέες ψηφιακές τεχνολογίες που παρέχουν ευκολίες στον άνθρωπο. Η εύρεση λύσης επομένως με χρήση νέων τεχνολογιών, είναι επιβεβλημένη.

## 1.2 Αντιμετώπιση του προβλήματος

### 1.2.1 Τρέχουσα κατάσταση

Αρχικά η λύση που κυριάρχησε και παραμένει ως σήμερα είναι η προφορική ενημέρωση. Δηλαδή αυτό που συμβαίνει στην ουσία είναι, όταν ένας σπουδαστής ενδιαφέρεται να μάθει τι είναι η πτυχιακή και άλλες σχετικές πληροφορίες ρωτάει κάποιον φοιτητή από μεγαλύτερο εξάμηνο και μαθαίνει. Αυτή η προσέγγιση όμως, δεν αποτελεί ουσιαστική λύση καθώς τα προβλήματα παραμένουν. Θα πρέπει από μόνος του ο φοιτητής να ψάξει καθηγητή, να προτείνει ή να του προτείνουν θέματα κ.ο.κ. Δηλαδή δεν μπορεί να αποφύγει όλη αυτή τη χρονοβόρα διαδικασία. Αυτή η λύση επομένως αποτελεί μια βολική κατάσταση μόνο για όσους δεν επιθυμούν τη βελτίωση της οργάνωσης του ιδρύματος.

Μια βελτίωση, η οποία δυστυχώς δεν παρατηρείται συχνά, είναι η πρωτοβουλία κάποιων καθηγητών να δημοσιεύουν μέσω της προσωπικής τους ιστοσελίδας ή με αποστολή email προτεινόμενες πτυχιακές. Υπάρχει επίσης και διάσπαρτα σχετικά κείμενα στις ιστοσελίδες των ιδρυμάτων που αναφέρεται στο πρόγραμμα σπουδών και εξηγούν περιληπτικά τη διαδικασία εκπόνησης πτυχιακής εργασίας. Γιατί όλα αυτά όμως να μην είναι συγκεντρωμένα σ' ένα περιβάλλον που θα παρέχει:

- μεγαλύτερη ευκολία στους καθηγητές για τη διαχείριση των πτυχιακών τους εργασιών και
- τη καλύτερη δυνατή ενημέρωση στους φοιτητές

### 1.2.2 Προσέγγιση λύσης

Τη δημιουργία αυτού του περιβάλλοντος, προσπαθεί να προσεγγίσει αυτή η μεταπτυχιακή εργασία με τη χρήση 2 διαφορετικών scripting γλωσσών προγραμματισμού ώστε να πραγματοποιηθεί παράλληλα και η σύγκριση τους και να προκύψουν παράλληλα και ερευνητικά συμπεράσματα για το ποια είναι περισσότερο αποτελεσματική, εύχρηστη, ευέλικτη, λειτουργική και πιο εύκολη. Οι δύο συγκριθείς γλώσσες είναι :

- PHP
- RUBY

Η διαδικασία που αφορά αυτό το περιβάλλον και τον **φοιτητή**, έχει ως εξής :

- Ο φοιτητής, επισκέπτεται την ιστοσελίδα πτυχιακών εργασιών και επιλέγει την είσοδο φοιτητών. Στη συνέχεια από τη λίστα πτυχιακών, μελετάει τα θέματα. Μπορεί να βρει μια προτεινόμενη πτυχιακή που θα τον ενδιέφερε

να εκπονήσει, μπορεί όμως να βρει και κάποια σε εξέλιξη ή ολοκληρωμένη που να του δώσει ιδέες για κάτι παρεμφερές ή για κάποια επέκταση.

- Στη συνέχεια για οποιαδήποτε πρόταση νέου θέματος(να προτείνει δηλαδή δικό του θέμα, το οποίο μπορεί να είναι παρεμφερές κάποιου άλλου ή ακόμα και επέκταση ολοκληρωμένης πτυχιακής), ή για εκδήλωση ενδιαφέροντος για συγκεκριμένη προτεινόμενη πτυχιακή, μπορεί ο φοιτητής να επικοινωνήσει είτε άμεσα είτε με email με τον καθηγητή.
- Εφόσον ο φοιτητής πληροί τις προϋποθέσεις για τη προτεινόμενη πτυχιακή που επέλεξε, στη συνέχεια του γίνεται ανάθεση και του αποστέλλεται αυτομάτως με email ο κωδικός πτυχιακής.
- Για όλη τη διάρκεια που ο φοιτητής θα εκπονεί τη πτυχιακή, θα μπορεί μέσα από ένα εσωτερικό σύστημα διαχείρισης να μεταβάλει το ποσοστό εξέλιξης της εκπόνησης, με την χρήση του κωδικού πτυχιακής. Αυτή η διαδικασία ισχύει για όσο η πτυχιακή είναι “σε εξέλιξη”.

Η διαδικασία που αφορά αυτό το περιβάλλον και τον **καθηγητή**, έχει ως εξής :

- Ο καθηγητής αρχικά εισάγει το username και το password που του έχει δοθεί, για τον server του τμήματος. Αν εισάγει σωστά τα στοιχεία και δεν του επιτρέπεται η πρόσβαση τότε πιθανότατα να μην είναι καταχωρημένο το username του, στη βάση του συστήματος. Για να λυθεί το πρόβλημα θα πρέπει να επικοινωνήσει με τον διαχειριστή του συστήματος, όπως αυτός θα έχει οριστεί.
- Κατά την είσοδο του ο καθηγητής, αντικρίζει τρεις βασικές επιλογές. Επιλέγει “Λίστα πτυχιακών” αν θέλει να μελετήσει τις πτυχιακές που υπάρχουν ήδη από άλλους συναδέλφους του, να πάρει ιδέες για κάτι καινούργιο, να έχει μια γενική εικόνα για το πώς θα κινηθεί και για ν’ αποφύγει τυχόν αντιγραφές με προϋπάρχοντα θέματα.  
*Σημείωση : Η λίστα πτυχιακών είναι αυτή ακριβώς που βλέπουν και οι σπουδαστές.*
- Επιλέγοντας οι πτυχιακές μου, ακολουθούν τέσσερις επιλογές.
  1. Προτεινόμενες πτυχιακές
  2. Σε εξέλιξη πτυχιακές
  3. Ολοκληρωμένες πτυχιακές
  4. Όλες οι κατηγορίες
- Ο καθηγητής επιλέγει “Προτεινόμενες πτυχιακές”, όταν θέλει να δημιουργήσει κάποιο νέο θέμα πτυχιακής, να δει όλα τα προϋπάρχοντα θέματα πτυχιακής που έχει ως τώρα προτείνει και δεν έχουν αναλάβει φοιτητές, να μεταβάλλει το περιεχόμενο τους και να διαγράψει κάποιο ή

κάποια από αυτά, αν το επιθυμεί. Τέλος μπορεί να κάνει ανάθεση προτεινόμενης πτυχιακής σε σπουδαστές που ενδιαφέρονται.

- Ο καθηγητής επιλέγει “Σε εξέλιξη πτυχιακές”, όταν θέλει να δει όλα τα θέματα πτυχιακής που έχει αναλάβει και είναι σε εξέλιξη, να μεταβάλλει το περιεχόμενο τους και να διαγράψει κάποιο ή κάποια από αυτά, αν το επιθυμεί. Μπορεί επίσης να δει δίπλα από τον τίτλο της εργασίας το ποσοστό εξέλιξης της, όπως το χει ορίσει ο φοιτητής, να δηλώσει την ολοκλήρωση κάποιου θέματος εφόσον έχει παραδοθεί και παρουσιαστεί από τον φοιτητή ή τους φοιτητές που το είχαν αναλάβει. Τέλος, για περιπτώσεις φοιτητών που παρατούν την εκπόνηση κάποιας πτυχιακής, υπάρχει και η δυνατότητα της ακύρωσης και επαναφοράς του θέματος στις προτεινόμενες πτυχιακές.
- Ο καθηγητής επιλέγει “Ολοκληρωμένες πτυχιακές”, όταν θέλει να δει όλα τα θέματα πτυχιακής που έχει αναλάβει κι έχουν ολοκληρωθεί. Να φορτώσει κάποιο αρχείο που είναι σχετικό με κάποια πτυχιακή ή ακόμα και να διαγράψει φορτωμένο αρχείο. Μπορεί επίσης να εισάγει και στοιχεία από πτυχιακές που έχουν παραδοθεί στο παρελθόν, να τις μεταβάλλει και να τις διαγράψει αν το επιθυμεί.
- Ο καθηγητής επιλέγει “Όλες οι κατηγορίες”, όταν θέλει να δει όλα τα θέματα πτυχιακής που έχει αναλάβει και είναι οποιασδήποτε μορφής, προτεινόμενα, σε εξέλιξη, ολοκληρωμένα...
- Μια ακόμη επιλογή που έχει ο καθηγητής είναι η διαχείριση του προφίλ του, μέσα από την οποία μπορεί να αλλάξει τον κωδικό πρόσβασης του, το ονοματεπώνυμο κλπ.
- Τέλος, ως διαχειριστής (administrator) διαθέτει και την επιλογή της διαχείρισης των χρηστών, μέσα από την οποία μπορεί να διαχειριστεί τους χρήστες του συστήματος, τροποποιώντας τα στοιχεία τους ή διαγράφοντας τους.

Όλα αυτά συνθέτουν το βασικό σενάριο, στο οποίο στηρίζεται το δυναμικό αυτό περιβάλλον.

Το επόμενο κεφάλαιο αναφέρεται στα εργαλεία που χρησιμοποιήθηκαν για την πρακτική υλοποίηση αυτού του περιβάλλοντος, στη μία περίπτωση, όπου έγινε χρήση της scripting γλώσσας προγραμματισμού PHP.

## 2. Ανάπτυξη με PHP

Στο κεφάλαιο αυτό γίνεται μια συνοπτική αναφορά στις τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της συγκεκριμένης πτυχιακής με τον ένα τρόπο. Οι τεχνολογίες αυτές είναι οι εξής : PHP, HTML, APACHE και MYSQL οι οποίες είναι λογισμικά ανοικτού κώδικα (Open Source Code).

### 2.1 Λογισμικά Ανοικτού Κώδικα (Open Source Code)

#### 2.1.1 Τι είναι το Λογισμικό Ανοικτού Κώδικα (Open Source) ;

Για τον επιχειρηματία πρόκειται απλώς για λογισμικό που διατίθεται δωρεάν και με όλο του τον κώδικα. (Το λογισμικό "γράφεται" με μια γλώσσα προγραμματισμού και στη συνέχεια αυτός ο "πηγαίος κώδικας", δηλαδή ό,τι γράφτηκε στη γλώσσα, μετατρέπεται σε εκτελέσιμο πρόγραμμα.)

Αντίθετα, στο εμπορικό λογισμικό ο χρήστης όχι μόνο πληρώνει για να λάβει το πρόγραμμα που θα "τρέχει" στον Η/Υ του, αλλά δεν έχει στη διάθεσή του και τον πηγαίο κώδικα. Δεν μπορεί λοιπόν να αλλάξει το πρόγραμμα ο ίδιος και πρέπει να ζητά (και να πληρώνει) όποια τροποποίηση επιθυμεί μόνο στον αρχικό κατασκευαστή.

Η παραπάνω ερμηνεία αν και είναι λίγο απλοϊκή είναι αρκετή για τους σκοπούς αυτής της εργασίας. Ιδεολογικά υπάρχουν σημαντικές διαφορές ανάμεσα στο λογισμικό ανοικτού κώδικα (Open Source) και το ελεύθερο λογισμικό (Free Software). Και στις δύο περιπτώσεις ο κώδικας πρέπει να είναι διαθέσιμος σε όλους για διαφορετικό όμως λόγο. Στη πρώτη περίπτωση η ελεύθερη διαθεσιμότητα κάνει τα προγράμματα καλύτερα, ενώ στη δεύτερη προσφέρει ελευθερία στον χρήστη καθώς αυτός δεν περιορίζεται από τις αποφάσεις του κατασκευαστή. Το ότι τα προγράμματα γίνονται καλύτερα λοιπόν αποτελεί ένα ευχάριστο αλλά δευτερεύον χαρακτηριστικό. Οι οπαδοί του Free Software θα το χρησιμοποιούσαν ακόμη κι αν το εμπορικό λογισμικό ήταν καλύτερης ποιότητας από το δικό τους.

#### 2.1.2 Τα πλεονεκτήματα του Open Source

Οι περισσότερες από τις συγκρίσεις που ακολουθούν γίνονται μεταξύ Windows και GNU/Linux μια και τα πρώτα αποτελούν το de facto μονοπώλιο στο επιχειρηματικό λογισμικό, ενώ το δεύτερο τη δημοφιλέστερη πλατφόρμα εφαρμογών σε Open Source. Υπάρχουν λοιπόν πολύ περισσότερα διαθέσιμα στοιχεία.

**1. Αξιοπιστία** - Πολλές έρευνες (Zdnet, Bloor Research, Syscontrol AG, Netcraft κ.λπ.) έχουν δείξει την ανωτερότητα εφαρμογών όπως το λειτουργικό σύστημα

GNU/Linux και ο Apache web server. Για παράδειγμα, σε δοκιμή του Zdnet διάρκειας 10 μηνών τα Windows NT "κράσαραν" κατά μέσο όρο μια φορά κάθε 6 εβδομάδες και απαιτούσαν 30 λεπτά για την επισκευή τους ενώ το GNU/Linux ποτέ!

Αν αυτό φαίνεται "παράλογο" (το δωρεάν λειτουργεί πιο αξιόπιστα από το πληρωμένο) θα μπορούσε κάποιος να αναλογιστεί ότι οι εφαρμογές Open Source είναι αποτέλεσμα εθελοντικής εργασίας. Δεν υπάρχει λοιπόν πίεση από το Marketing και τις πωλήσεις να παρουσιαστεί κάτι, ακόμη κι αν δεν είναι έτοιμο ή καλά δοκιμασμένο, ενώ όλοι μπορούν να δουν, να σχολιάσουν και να διορθώσουν τη δουλειά των άλλων.

**2. Αποδοτικότητα** - Μεγάλος αριθμός συγκριτικών δοκιμών έχει αποδείξει την ταχύτητα και αποδοτικότητα πολλών Open Source εφαρμογών βασισμένων στο GNU/Linux (π.χ. PC Magazine, Sys Admin magazine, SPEC Consortium, IBM, Fastcenter, Ziff Davis, Mindcraft κ.λπ.).

**3. Επεκτασιμότητα** - Οι περισσότερες επιχειρηματικές εφαρμογές επιθυμούν χαμηλό κόστος εκκίνησης με εύκολη και γρήγορη αναβάθμιση, αν αποδειχθεί ότι η υπηρεσία που δημιουργήθηκε είναι δημοφιλής. Λόγω του ελεύθερου κώδικά τους οι εφαρμογές Open Source μπορούν εύκολα να τοποθετηθούν σε πολλά διαφορετικά είδη (πλατφόρμες) υπολογιστών, αλλά και να λειτουργήσουν "εν παραλλήλω" σε μεγάλα συστήματα υψηλών επιδόσεων.

**4. Ασφάλεια δεδομένων** - Όλοι οι κρυπτογράφοι γνωρίζουν πως όποιο σύστημα ελέγχεται από πολλούς ανθρώπους είναι ασφαλέστερο από εκείνο που διαχειρίζονται μόνο λίγοι (όσο "καταρτισμένοι ή ευφυείς" κι αν είναι αυτοί). Χάρη στον ανοικτό τους κώδικα οι εφαρμογές Open Source "ελέγχονται" από χιλιάδες ανθρώπους και τα όποια προβλήματά τους ανακαλύπτονται πολύ γρήγορα. Για παράδειγμα, όπως έδειξε έρευνα της Attrition.org το 59% των defaced (τους άλλαξαν τη home page) sites στο Internet έτρεχαν Windows, ενώ μόνο το 21% GNU/Linux, πράγμα που θα ήταν φυσιολογικό μόνο αν στο δίκτυο υπήρχαν 3 φορές περισσότερα συστήματα Windows απ' ό,τι GNU/Linux (κάτι που φυσικά δεν συμβαίνει). Άλλη έρευνα (SecurityPortal) έδειξε πως η ταχύτητα διευθέτησης όσων προβλημάτων εμφανίζονται στο GNU/Linux είναι πολύ μεγαλύτερη απ' ό,τι στα Windows.

**5. Προστασία από ιούς** - Υπάρχουν περίπου 60.000 ιοί για Windows και περίπου 40 για GNU/Linux.

**6. Χαμηλότερο κόστος λειτουργίας (Total Cost of Ownership)** - Το Open Source λογισμικό (λειτουργικά συστήματα, βάσεις δεδομένων κ.λπ.) παρέχεται δωρεάν και για απεριόριστο αριθμό χρηστών. Επίσης, η επιχείρηση μπορεί να αξιοποιήσει παλαιότερο εξοπλισμό (αφού είναι ελεύθερη να κάνει τροποποιήσεις στον κώδικα), μειώνοντας έτσι τα έξοδά της. Χαρακτηριστική εδώ είναι η περίπτωση του Amazon.com. Ελάχιστοι γνωρίζουν ότι το γνωστό βιβλιοπωλείο κατάφερε να καταστεί κερδοφόρο επειδή στράφηκε στο Open Source λογισμικό, μειώνοντας

θεαματικά τα έξοδά του. Επίσης, πολλές χρηματιστηριακές επιχειρήσεις χρησιμοποιούν Linux γεγονός που κρατούν κρυφό, θεωρώντας το ως ένα σημαντικό ανταγωνιστικό πλεονέκτημα!

**7. Ευκολότερες και φθηνότερες επεκτάσεις** - Επειδή ο χρήστης διαθέτει πρόσβαση στον κώδικα του λογισμικού που χρησιμοποιεί μπορεί να ζητήσει επεκτάσεις ή αλλαγές από οποιονδήποτε. Ο κατασκευαστής δηλαδή δεν διαθέτει μονοπωλιακή δύναμη επάνω του και δεν μπορεί να τον "εκβιάσει" με υποχρεωτικές αναβαθμίσεις ή υπέρογκες χρεώσεις για μικρές εργασίες. Αν ο χρήστης λοιπόν δεν είναι ικανοποιημένος από τη συνεργασία ενός προμηθευτή μπορεί να απευθυνθεί σε κάποιον άλλον, χωρίς να απαιτηθεί η αντικατάσταση των εφαρμογών που χρησιμοποιεί.

**8. Διαρκής και εκτενής υποστήριξη** - Οι κατασκευαστές εμπορικού λογισμικού διορθώνουν μόνο τα πιο ενοχλητικά από τα προβλήματα των εφαρμογών τους και υποχρεώνουν τους πελάτες τους να αγοράζουν νέες αναβαθμίσεις ακόμη κι αν δεν τις χρειάζονται, σταματώντας την υποστήριξη των παλαιότερων εκδόσεων του προγράμματος. Στις εφαρμογές Open Source όμως η επιχείρηση μπορεί να διορθώσει η ίδια ό,τι την "πονάει" στην εφαρμογή και δεν υποχρεώνεται να αναβαθμίσει τίποτε αν δεν το επιθυμεί.

**9. Το μέλλον ανήκει στο Open Source** - Το 49% των Ευρωπαίων CIOs (Chief Information Officers) δηλώνει ότι θα υιοθετήσει εφαρμογές Open Source στην επιχείρησή του (Φεβρουάριος 2002) λόγω χαμηλότερου κόστους λειτουργίας (54%), χαμηλότερης δαπάνης αγοράς (24%), μεγαλύτερου ελέγχου (22%) και υψηλότερης ασφάλειας δεδομένων (22%).

### 2.1.3 Τα μειονεκτήματα του Open Source

**1. Μερίδιο αγοράς** - Αν και ο Apache αποτελεί τον δημοφιλέστερο web server στο Internet και το GNU/Linux διαθέτει ένα μεγάλο ποσοστό της αγοράς των Internet Servers, στην Ελλάδα υπάρχουν ακόμη πολύ λίγοι τεχνικοί με εμπειρία σε παρόμοια συστήματα. Αναμφίβολα πάντως οι γνώσεις τους είναι πολύ μεγαλύτερες από εκείνες του μέσου windows administrator ο οποίος συνήθως μπορεί να εκτελέσει μόνο τις πολύ βασικές λειτουργίες ενός συστήματος και αγνοεί τις πιο προχωρημένες δυνατότητες και εφαρμογές του. (Είναι πιο εύκολο να μάθει κάποιος τις βασικές λειτουργίες ενός windows συστήματος, αλλά δυσκολεύεται κατόπιν πολύ περισσότερο απ' ό,τι στο GNU/Linux για να κατανοήσει και να υλοποιήσει κάτι προχωρημένο.)

**2. Ενδοεταιρικός καταλογισμός ευθυνών** - Όπως λέει ένα γνωστό αμερικάνικο ρητό "Κανείς δεν απολύθηκε επειδή αγόρασε IBM". Ένα προϊόν της Microsoft ή της Oracle μπορεί να κοστίζει πολύ περισσότερα χρήματα και να προσφέρει λιγότερα απ' όσα μια εφαρμογή Open Source, αλλά όποιο πρόβλημα και αν παρουσιαστεί, το στέλεχος που εισηγήθηκε την αγορά του μπορεί πάντοτε να ισχυριστεί πως "έκανα μια επώνυμη επιλογή ενός μεγάλου ονόματος για να έχουμε το καλύτερο". Αν όμως



πάει στραβά το παραμικρό σε μια εφαρμογή Open Source (και στην πληροφορική πάντα κάτι θα λειτουργήσει στραβά) η μομφή "γιατί εμπιστευθήκαμε κάτι φτηνιάρικο;" δύσκολα μπορεί να απαντηθεί με επιτυχία, ειδικά αν ο ερωτών είναι ο άσχετος από υπολογιστές προϊστάμενος, όπως συμβαίνει συνήθως. Τα πράγματα θα ήταν καλύτερα αν υπήρχαν και στη χώρα μας εταιρείες εξειδικευμένες στην υποστήριξη open source εφαρμογών, αλλά απ' όσο γνωρίζω αυτό δεν συμβαίνει.

## 2.2 PHP

### 2.2.1 Τι είναι η PHP ;

Η PHP<sup>[1]</sup>, της οποίας τα αρχικά αντιπροσωπεύουν το "PHP: Hypertext Preprocessor" είναι μια ευρέως χρησιμοποιούμενη, ανοιχτού κώδικα, γενικού σκοπού scripting γλώσσα προγραμματισμού, η οποία είναι ειδικά κατάλληλη για ανάπτυξη εφαρμογών για το Web και μπορεί να ενσωματωθεί στην HTML.

Απλή απάντηση, αλλά τι σημαίνει; Ένα παράδειγμα:

#### Παράδειγμα 1-1. Ένα εισαγωγικό παράδειγμα

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>

    <?php
    echo "Hi, I'm a PHP script!";
    ?>

  </body>
</html>
```

Το PHP script είναι διαφορετικό από ένα script γραμμένο σε άλλες γλώσσες προγραμματισμού όπως η Perl ή η C : Αντί να γράφετε ένα πρόγραμμα με πολλές εντολές για να εξάγετε HTML, γράφετε ένα HTML script με κάποιο ενσωματωμένο κώδικα για να κάνει κάτι (σε αυτή την περίπτωση, να εμφανίζει κάποιο κείμενο). Ο κώδικας PHP είναι εσώκλειστος σε ειδικά tags (ετικέτες) αρχής και τέλους που σας επιτρέπουν να μεταφέρεστε μέσα και έξω από το "PHP mode" (PHP τρόπο λειτουργίας).

Αυτό που διαχωρίζει την PHP από κάτι σαν client-side Javascript είναι ότι ο κώδικας εκτελείται στον server (εξηγηρητητή). Αν είχατε ένα script σαν το παραπάνω στον server σας, ο client θα έπαιρνε τα αποτελέσματα της εκτέλεσης αυτού του script, χωρίς να υπάρχει κανένας τρόπος να καταλάβει τι κώδικας

υπάρχει από κάτω. Μπορείτε ακόμη να ρυθμίσετε τον web server σας να χειρίζεται όλα τα HTML αρχεία σας με την PHP, και τότε πραγματικά δεν υπάρχει τρόπος ο χρήστης να καταλάβει τι έχετε κάτω από το μανίκι σας.

Τα καλύτερο πράγμα στην PHP είναι ότι είναι εξαιρετικά απλή για ένα νεοφερμένο αλλά προσφέρει πολλά προηγμένα χαρακτηριστικά για ένα επαγγελματία προγραμματιστή. Θα πρέπει να μην τρομάζει κάποιος όταν διαβάζει την μακροσκελή λίστα με τα χαρακτηριστικά της PHP. Μπορεί να εξοικειωθεί μέσα σε πολύ λίγο χρόνο και να αρχίσει να γράφει απλά script σε λίγες ώρες.

Αν και η ανάπτυξη της PHP εστιάζεται σε server-side scripting, μπορεί κάποιος να κάνει πολύ περισσότερα με αυτή.

### 2.2.2 Τι μπορεί να κάνει η PHP ;

Η PHP επικεντρώνεται κυρίως στο server-side scripting, έτσι μπορεί κανείς να κάνει οτιδήποτε μπορεί να κάνει ένα άλλο CGI πρόγραμμα, όπως να μαζέψει δεδομένα, να παράγει δυναμικό περιεχόμενο σελίδων, ή να στείλει και να πάρει cookies. Αλλά η PHP μπορεί να κάνει πολύ περισσότερα.

Υπάρχουν τρεις κύριοι τομείς που χρησιμοποιείται ένα PHP script.

- Server-side scripting. Αυτό είναι το πιο παραδοσιακό και το κύριο πεδίο για την PHP. Απαιτούνται τρία πράγματα για να δουλέψει αυτό. Ο PHP μεταγλωττιστής (parser) (CGI ή server module), ένας webserver (εξηγηρητής σελίδων) και ένας web browser ("φυλλομετρητής"). Πρέπει να τρέξει ο webserver, με μια συνδεδεμένη εγκατάσταση της PHP. Μπορούν να προσπελαστούν τα αποτελέσματα του PHP προγράμματος με ένα web browser, βλέποντας την σελίδα PHP μέσα από τον server.
- Command line scripting. Μπορεί να φτιαχτεί ένα PHP script για να τρέχει χωρίς server ή browser. Χρειάζεται μόνο ο PHP μεταγλωττιστής για να χρησιμοποιηθεί με αυτό τον τρόπο. Αυτός ο τύπος είναι ιδανικός για script που εκτελούνται συχνά με τη χρήση της cron (σε Unix ή Linux) ή με τον Task Scheduler (στα Windows). Αυτά τα script μπορούν επίσης να χρησιμοποιηθούν για απλές εργασίες επεξεργασίας κειμένου.
- Εγγραφή client-side GUI εφαρμογών (Γραφικά περιβάλλοντα χρηστών). Η PHP ίσως να μην είναι η πιο καλή γλώσσα για να γράψει κανείς παραθυρικές εφαρμογές, αλλά αν κάποιος ξέρει PHP πολύ καλά και θέλει να χρησιμοποιήσει κάποια προχωρημένα χαρακτηριστικά της PHP στις client-side εφαρμογές του, μπορεί επίσης να χρησιμοποιήσει το PHP-GTK για αυτού του είδους τα προγράμματα. Το PHP-GTK είναι μια επέκταση της PHP και δεν συμπεριλαμβάνεται στην κύρια διανομή.

Η PHP μπορεί να χρησιμοποιηθεί σε όλα τα κύρια λειτουργικά συστήματα, συμπεριλαμβανομένου του Linux, πολλών εκδοχών του Unix (HP-UX, Solaris και OpenBSD), Microsoft Windows, Mac OS X, RISC OS και πιθανώς σε άλλα. Η PHP

υποστηρίζει επίσης τους Apache, Microsoft Internet Information Server, Personal Web Server, Netscape και iPlanet servers, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd, και πολλούς άλλους webserver. Για την πλειοψηφία των server η PHP έχει ένα module, για τους υπόλοιπους η PHP μπορεί να λειτουργήσει ως ένας CGI επεξεργαστής.

Έτσι με την PHP παρέχεται ελευθερία επιλογής ενός λειτουργικού συστήματος και ενός web server. Επιπλέον, παρέχεται η ελευθερία να χρησιμοποιήσει κάποιος συναρτησιακό (procedural) ή αντικειμενοστρεφή (object oriented) προγραμματισμό ή μια ανάμειξη τους. Αν και η παρούσα έκδοση δεν υποστηρίζει όλα τα πρότυπα χαρακτηριστικά, μεγάλες βιβλιοθήκες κώδικα και μεγάλες εφαρμογές (συμπεριλαμβανομένης και της βιβλιοθήκης PEAR) είναι γραμμένες μόνο με αντικειμενοστρεφή κώδικα.

Με την PHP δεν περιορίζεται κανείς στο να εξάγει HTML. Οι δυνατότητες της PHP συμπεριλαμβάνουν την εξαγωγή εικόνων, αρχείων PDF, ακόμη και ταινίες Flash (χρησιμοποιώντας τα libswf και Ming) παράγονται αμέσως. Μπορεί επίσης να εξάγει εύκολα οποιοδήποτε κείμενο όπως XHTML και οποιοδήποτε άλλο XML αρχείο. Η PHP μπορεί να δημιουργεί αυτόματα αυτά τα αρχεία και να τα αποθηκεύει στο σύστημα αρχείων, αντί να τα εκτυπώνει, αποτελώντας έτσι μια server-side cache για το δυναμικό του περιεχόμενο.

Ένα από τα πιο δυνατά και σημαντικά χαρακτηριστικά της PHP είναι η υποστήριξη που έχει για ένα μεγάλο σύνολο βάσεων δεδομένων. Η συγγραφή μιας σελίδας που υποστηρίζει βάσεις δεδομένων είναι εξαιρετικά απλή. Οι εξής βάσεις δεδομένων υποστηρίζονται μέχρι στιγμής:

Adabas D	Ingres	Oracle (OCI7 and OCI8)
dBase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro (read-only)	mSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Velocis
Informix	ODBC	Unix dbm

Υπάρχει επίσης μια αφαιρετική επέκταση DBX βάσεων δεδομένων (DBX database abstraction extension) που επιτρέπει διάφανα τη χρήση οποιασδήποτε βάσης δεδομένων που υποστηρίζεται από αυτή την επέκταση. Επιπλέον η PHP υποστηρίζει το ODBC, το Open Database Connection standard (Ανοιχτό πρότυπο Σύνδεσης Βάσεων δεδομένων) έτσι μπορεί κανείς να συνδεθεί σε οποιαδήποτε βάση δεδομένων που υποστηρίζει αυτό το παγκόσμιο πρότυπο.

Η PHP έχει επίσης υποστήριξη για επικοινωνία με άλλες υπηρεσίες χρησιμοποιώντας πρωτόκολλα όπως LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (στα Windows) και αμέτρητα άλλα. Μπορεί επίσης κάποιος να ανοίξει raw network

sockets και να αλληλεπιδράσει με οποιοδήποτε άλλο πρωτόκολλο. Η PHP έχει ακόμη υποστήριξη για την περίπλοκη ανταλλαγή δεδομένων WDDX μεταξύ σχεδόν όλων των Web programming γλωσσών. Μιλώντας για δια-επικοινωνία, η PHP υποστηρίζει instantiation αντικειμένων Java και τα χρησιμοποιεί διάφανα σαν αντικείμενα PHP. Μπορεί τέλος κάποιος να χρησιμοποιήσει την CORBA επέκταση για να προσπελάσει απομακρυσμένα αντικείμενα.

Η PHP έχει εξαιρετικά χρήσιμα χαρακτηριστικά επεξεργασίας κειμένων, από την POSIX επέκταση ή τις Perl regular expressions μέχρι XML parsing αρχείων. Για τη μεταγλώττιση και την πρόσβαση αρχείων XML, υποστηρίζονται τα πρότυπα SAX και DOM. Μπορεί να χρησιμοποιηθεί η XSLT επέκταση για να μετατραπούν τα XML αρχεία σε άλλες μορφές.

Καθώς χρησιμοποιεί κανείς την PHP στον τομέα του ecommerce, θα βρεί τις Cybercash payment, CyberMUT, VeriSign Payflow Pro και CCVS συναρτήσεις χρήσιμες για τα online προγράμματα πληρωμής.

Τελευταίο αλλά σημαντικό, έχουμε πολλές άλλες ενδιαφέρουσες επεκτάσεις, τις mhoGoSearch search engine συναρτήσεις, πολλά εργαλεία συμπίεσης (gzip, bz2), μετατροπές ημερολογίου, μεταφράσεις...

Όπως είναι προφανές, αυτή η ενότητα δεν είναι αρκετή για να απαριθμήσει όλα τα χαρακτηριστικά και πλεονεκτήματα της PHP.

## 2.3 HTML

### 2.3.1 Τι είναι η HTML ;

**Με απλά λόγια**, η HTML είναι μια γλώσσα η οποία λέει σε ένα υπολογιστή πως να εμφανίσει μια ιστοσελίδα. Τα HTML αρχεία είναι στην ουσία απλά αρχεία κειμένου (text files-ASCII) με ειδικές "ετικέτες" (tags) ή κώδικες που ένα πρόγραμμα, ο web browser, ξέρει να μεταφράζει και να εμφανίζει τα αποτελέσματα στη οθόνη.

### 2.3.2 Τι είναι οι ετικέτες της HTML ;

Όταν ο web browser εμφανίζει μία σελίδα, σαν και αυτή που διαβάζει κανείς τώρα, στην ουσία διαβάζει από ένα αρχείο απλού κειμένου και κοιτάει για ειδικούς κώδικες ή αλλιώς "ετικέτες" που περιλαμβάνονται μεταξύ των συμβόλων < και >.

Η γενική μορφή μιας HTML ετικέτας είναι:

```
<tag_name>Σώμα κειμένου</tag_name>
```

Για παράδειγμα, ο τίτλος αυτής της ενότητας χρησιμοποιεί **header** ετικέτα:

```
<h3>Τι είναι οι ετικέτες της HTML;</h3>
```

Αυτή η ετικέτα λέει στον web browser να εμφανίσει το κείμενο **“Τι είναι οι ετικέτες της HTML;”** σε στύλ header 3ου επιπέδου. Οι HTML ετικέτες δίνουν εντολή στον web browser να κάνει έντονο (bold) το κείμενο, πλαγιαστό (italicize), ή να το μετατρέψει σε σύνδεσμο σε μία άλλη ιστοσελίδα. Είναι απαραίτητο να σημειώσουμε ότι η ετικέτα κλεισίματος,

```
</tag_name>
```

περιέχει το χαρακτήρα "/" (slash), ο οποίος λέει στον web browser τότε σταματάει η ετικέτα. Πολλές ετικέτες χρησιμοποιούν αυτόν τον χαρακτήρα, τον οποίο αν ξεχάσει κανείς να τον βάλει, ο web browser θα συνεχίσει την ετικέτα για το επόμενο κείμενο του αρχείου, και το αποτέλεσμα θα είναι δυσνόητο.

**ΣΗΜΕΙΩΣΗ: Είναι το ίδιο για τον web browser εάν βάλετε κεφαλαία ή μικρά. Για παράδειγμα, <h3>...</h3> δεν διαφέρει από το <H3>...</H3>**

Εάν κάνει κάποιος τυπογραφικό λάθος στην HTML, δεν θα κρεμάσει το σύστημα αλλά θα εμφανιστεί λάθος η ιστοσελίδα. Αυτό που έχει να κάνει, είναι να μπει μέσα στην HTML και να διορθώσει το λάθος.

Ο browser έχει ανοιχτό ένα μικρό λεξικό. Το ενδιαφέρον θέμα της HTML είναι ότι αν ο browser δεν καταλαβαίνει τι να κάνει με την ετικέτα, την αγνοεί. Για παράδειγμα στο παρακάτω έγγραφο, η header ετικέτα σε αυτή την ενότητα πραγματικά μοιάζει σαν αυτήν:

```
<wigggle><h3>Τι είναι οι ετικέτες της HTML;</h3></wigggle>
```

αλλά απο τότε που ο browser πιθανότατα δεν αναγνωρίζει την ετικέτα **<wigggle>**, συνεχίζει με αυτά που αναγνωρίζει.

## 2.4 APACHE

### 2.4.1 Τι είναι ο Apache ;

Ο Apache<sup>[2]</sup>, είναι ένας εξυπηρετητής του παγκόσμιου ιστού (www/http server) που σχεδιάστηκε για να αντικαταστήσει τον NCSA http server. Σαν μεταγενέστερος διορθώνει πολλά λάθη τα οποία υπήρχαν στον NCSA και συγχρόνως ενσωματώνει καινούργια χαρακτηριστικά σύμφωνα με τις τελευταίες τεχνολογίες Δυναμικών Διαμοιραζόμενων Αντικειμένων (DSO). Η αρχιτεκτονική DSO

(Dynamic Shared Object Support) επιτρέπει στον διακομηστή να φορτώνει δυναμικά κατά την εκτέλεση του, κατάλληλες βιβλιοθήκες που χρειάζονται για την λειτουργία του, με αποτέλεσμα να γίνεται εξοικονόμηση μνήμης. Ένα χρήσιμο χαρακτηριστικό που διαθέτει, είναι τα λεγόμενα Virtual Hosts, με τα οποία μπορούμε να προσπελάσουμε πολλαπλές διευθύνσεις στον ίδιο εξυπηρετητή. Το βασικό του όμως πλεόνηκτημα είναι ότι διαθέτει API (Application Programmable Interface) το οποίο μπορεί να παραμετροποιηθεί κατάλληλα ώστε να προσαρμόζεται σύμφωνα με τις ανάγκες κάθε χρήστη / διαχειριστή.

Ο Apache αναπτύσσεται από κοινού από μια ομάδα εθελοντών που βρίσκονται σε όλο τον κόσμο, χρησιμοποιώντας το Διαδίκτυο για να επικοινωνούν, σχεδιάζουν, και αναπτύσσουν τον εξυπηρετητή και τη σχετική τεκμηρίωσή του. Αυτοί οι εθελοντές είναι γνωστοί ως ομάδα Apache. Επιπλέον, οι εκατοντάδες των χρηστών έχουν συμβάλει με ιδέες, κώδικα, και τεκμηρίωση στο πρόγραμμα. Αυτή η ενότητα περιγράφει εν συντομία την ιστορία της ομάδας Apache.

## 2.4.2 Πως δημιουργήθηκε ο Apache ;

Το Φεβρουάριο του 1995, το δημοφιλέστερο λογισμικό εξυπηρετητών στον Παγκόσμιο Ιστό ήταν το public domain HTTP daemon που αναπτύχθηκε από τον Rob McCool στο εθνικό κέντρο για εφαρμογές υπερυπολογιστών του Πανεπιστήμιου του Ιλλινόις.. Εντούτοις, η ανάπτυξη του httpd είχε χρονοτριβήσει αφότου άφησε ο Rob την NCSA στα μέσα του 1994, και πολλοί webmasters είχαν αναπτύξει τις δικές τους επεκτάσεις και διορθώσεις λαθών (bug fixes) που χρειάστηκαν για κοινή διανομή. Μια μικρή ομάδα αυτών των webmasters, ερχόμενοι σε επαφή μέσω του ιδιωτικού ηλεκτρονικού ταχυδρομείου, συγκεντρώθηκαν με σκοπό το συντονισμό των αλλαγών που έκριναν απαραίτητες. Ο Brian Behlendorf και Cliff Skolnick έβαλαν μαζί έναν κατάλογο διευθύνσεων, κοινό χώρο πληροφοριών, και logins για τους υπεύθυνους για την ανάπτυξη πυρήνων σε μια μηχανή στην περιοχή Bay Area της Καλιφόρνιας, με εύρος ζώνης που δίνεται από το HotWired. Μέχρι το τέλος του Φεβρουαρίου, οκτώ συνεισφέροντες πυρήνων διαμόρφωσαν το ίδρυμα της αρχικής ομάδας Apache:

Brian Behlendorf, Roy T. Fielding, Rob Hartill  
David Robinson, Cliff Skolnick, Randy Terbush  
Robert S. Thau, Andrew Wilson

με πρόσθετες συνεισφορές από τους

Eric Hagberg, Frank Peters, Nicolas Pioch

Χρησιμοποιώντας το NCSA httpd 1.3 ως βάση, προσθέσανε όλα τα δημοσιευμένα bug fixes και σημαντικές αυξήσεις που θα μπορούσαν να βρουν, εξετάσανε το αποτέλεσμα στους εξυπηρετητές τους, και κάνανε την πρώτη επίσημη

δημόσια διανομή (0.6.2) του Apache εξυπηρετητή τον Απρίλιο του 1995. Από σύμπτωση, η NCSA ξαναξεκίνησε την ανάπτυξη των εξυπηρετητών κατά τη διάρκεια της ίδιας περιόδου, και ο Brandon Long και Beth Frank της ομάδας ανάπτυξης εξυπηρετητών NCSA ένωσαν τον κατάλογο το Μάρτιο ως τιμητικά μέλη έτσι ώστε τα δύο προγράμματα να μπορούν να μοιραστούν τις ιδέες.

Ο πρώτος Apache server ήταν μια μεγάλη επιτυχία, παρόλα αυτά το codebase χρειαζόταν μια γενική εξέταση και έναν επανασχεδιασμό. Κατά τη περίοδο από τον Μάιο ως τον Ιούνιο του 1995, ενώ ο Rob Hartill και το υπόλοιπο της ομάδας εστίασαν στην εφαρμογή των νέων χαρακτηριστικών γνωρισμάτων των εκδόσεων 0.7.x και στην ενίσχυση της γρήγορα αυξανόμενης κοινότητας χρηστών Apache, ο Robert Thau σχεδίασε μια νέα αρχιτεκτονική εξυπηρετητών (με κωδικό Shambhala) που περιέλαβε μια πολύμορφη δομή και ένα API για καλύτερη επεκτασιμότητα, pool-based κατανομή μνήμης, και ένα προσαρμοστικό pre-forking πρότυπο διαδικασίας. Η ομάδα μεταπήδησε σε αυτήν την νέα βάση εξυπηρετητή τον Ιούλιο και πρόσθεσε τα χαρακτηριστικά γνωρίσματα από τις εκδόσεις 0.7.x, δημιουργώντας έτσι μια νέα έκδοση, τον Apache 0.8.8 τον Αύγουστο.

Μετά από την εκτενή beta δοκιμή και με την περαιτέρω προσθήκη νέου συνόλου τεκμηρίωσης και πολλών νέων χαρακτηριστικών γνωρισμάτων υπό μορφή τυποποιημένων ενοτήτων, το Apache 1.0 απελευθερώθηκε την 1η Δεκεμβρίου, 1995.

Σε λιγότερο από ένα έτος μετά από την ομάδα που διαμορφώθηκε, ο Apache server θεωρήθηκε από την NCSA httpd ως ο κορυφαίος εξυπηρετητής στο διαδίκτυο και σύμφωνα με την έρευνα της εταιρίας Netcraft, διατηρεί τη θέση αυτή μέχρι και σήμερα.

Το 1999, τα μέλη της ομάδας Apache διαμόρφωσαν το ίδρυμα λογισμικού Apache για να παρέχουν οργανωτική, νομική, και οικονομική ενίσχυση για τον Apache HTTP server. Το ίδρυμα έχει τοποθετήσει το λογισμικό σε μια στερεά θέση για τη μελλοντική ανάπτυξη.

## 2.4.3 Χαρακτηριστικά του Apache

### 2.4.3.1 Η δομή του αρχείου διαμόρφωσης του Apache

Το Apache διατηρεί όλες τις παραμέτρους διαμόρφωσης του σε αρχεία απλού κειμένου. Το κύριο αρχείο παραμέτρων διαμόρφωσης ονομάζεται httpd.conf. Το αρχείο αυτό περιέχει :

- **Ντιρεκτίβες** : διαμορφώνουν συγκεκριμένες ρυθμίσεις του Apache, όπως οι παράμετροι για τον έλεγχο πρόσβασης, την απόδοση και τη λειτουργία στο δίκτυο.

- **Περιέκτες :** καθορίζουν το πλαίσιο στο οποίο αναφέρονται αυτές οι ρυθμίσεις.

### 2.4.3.2 Αρχεία διαμόρφωσης ανά κατάλογο

Το Apache χρησιμοποιεί αρχεία διαμόρφωσης ανά κατάλογο (per directory configuration files) με τα οποία δίνει στις ντιρεκτίβες τη δυνατότητα να υπάρχουν έξω από το κύριο αρχείο διαμόρφωσης httpd.conf. Αυτά τα αρχεία μπορούν να τοποθετούνται στο επίπεδο του συστήματος αρχείων.

Η ενεργοποίηση των αρχείων αυτών έχει αρνητική επίδραση στην απόδοση. Το Apache είναι υποχρεωμένο να εκτελεί “δαπανηρές” λειτουργίες αναζήτησης αυτών των αρχείων σε κάθε αίτηση που λαμβάνει, ακόμη κι αν δεν υπάρχουν.

Εξ’ ορισμού τα αρχεία διαμόρφωσης ανά κατάλογο φέρουν το όνομα .htaccess. Για να εξακριβώσει κάποιος εάν μια ντιρεκτίβα μπορεί να χρησιμοποιηθεί σε αρχεία διαμόρφωσης ανά κατάλογο, ελέγχει εάν το πεδίο Context : στον ορισμό της ντιρεκτίβας περιέχει το .htaccess.

### 2.4.3.3 Αρχεία καταγραφής του Apache

Εξ’ ορισμού, ο Apache περιλαμβάνει δύο αρχεία καταγραφής (log files) :

- `access_log` : χρησιμοποιείται για την παρακολούθηση των αιτήσεων που στέλνουν τα client συστήματα στον server.
- `error_log` : χρησιμοποιείται για την καταγραφή σημαντικών συμβάντων, όπως π.χ. τα σφάλματα που συμβαίνουν ή η επανεκκίνηση του server.

## 2.5 MySQL

### 2.5.1 Τι είναι η MySQL ;

Η **MySQL**<sup>[3][4]</sup> είναι μια πολύ γρήγορη, πολυ-νηματική, αξιόπιστη και συμβατή με το πρότυπο ANSI SQL '92 βάση δεδομένων. Επίσης ανήκει στην κατηγορία των σχεσιακών βάσεων δεδομένων. Το γεγονός αυτό την κάνει αρκετά ευέλικτη στην διαχείριση μεγάλου όγκου πληροφορίας. Όλα τα αποθηκευμένα δεδομένα είναι καταχωρημένα υπο μορφή πινάκων, και υπάρχει μια αλληλοεξάρτηση μεταξύ αυτών, μέσω των σχέσεων των πινάκων της βάσης (relationship). Επίσης υποστηρίζει την φυσική και λογική ανεξαρτησία των δεδομένων, όπως αυτά περιγράφονται στα πρότυπα των σχεσιακών βάσεων (RDBMS). Το τμήμα SQL (Structured Query Language) της MySQL, είναι το υπεύθυνο για την πρόσβαση και διαχείριση των πινάκων. Τα πρότυπα που ακολουθεί μέχρι



στιγμής, είναι το πρότυπο SQL '92 και ODBC επιπέδου 0 – 2. Το λογισμικό MySQL, δίνεται μαζί με τον πηγαίο κωδικά του, με λίγα λόγια ανηκεί στην κατηγορία του ανοιχτού λογισμικού (Open Source Software), όπου ο καθένας μπορεί να κάνει αλλαγές στον κώδικα σύμφωνα με τις ανάγκες του. Επίσης διανέμεται με βάση την άδεια αντιγραφής κατά GNU (GPL, General Public License).

## 2.5.2 Ιστορία της MySQL

Η MySQL δημιουργήθηκε το 1979, όταν ο Michael "Monty" Widenius δημιούργησε ένα σύστημα βάσεων δεδομένων το UNIREG για τη σουηδική επιχείρηση TcX. Το UNIREG, εντούτοις, δεν είχε μια δομημένη γλώσσα διατύπωσης ερωτήσεων (SQL queries). Η TcX απέσυρε το σύστημα, στα μέσα της δεκαετίας του '90 και άρχισε την αναζήτηση εναλλακτικών λύσεων. Μια από εκείνες τις εναλλακτικές λύσεις ήταν η mSQL, ένα ανταγωνιστικό πρόγραμμα διαχείρισης βάσεων δεδομένων (ΠΔΒΔ) που δημιουργείται από το David Hughes.

Η mSQL δεν λειτούργησε για την TcX, έτσι ο Widenius αποφάσισε να δημιουργήσει έναν νέο εξυπηρετητή βάσεων δεδομένων που προσαρμόστηκε στις συγκεκριμένες απαιτήσεις του. Εκείνο το σύστημα, ολοκληρωμένο και απελευθερωμένο σε μια μικρή ομάδα τον Μάιο του 1996, έγινε η πρώτη έκδοση αυτού που είναι σήμερα γνωστό ως MySQL.

Μερικούς μήνες αργότερα, η MySQL 3.11 είδε την πρώτη δημόσια απελευθέρωσή της ως διανομή για Solaris. Σήμερα, η MySQL είναι διαθέσιμη για μια ευρεία ποικιλία πλατφόρμων, συμπεριλαμβανομένου Linux, MacOS, και Windows.

## 2.6 Λόγοι χρησιμοποίησης της PHP και της MySQL

Όταν κάποιος δημιουργεί μια τοποθεσία ηλεκτρονικού εμπορίου ή γενικότερα μια δυναμική ιστοσελίδα, υπάρχουν πολλά διαφορετικά προϊόντα που θα μπορούσε να χρησιμοποιήσει.

Είναι απαραίτητο να επιλέξει υλικό για το Web διακομιστή (H/Y υψηλών δυνατοτήτων με μεγάλη επεξεργαστική ικανότητα και μεγάλη χωρητικότητα μνήμης), ένα λειτουργικό σύστημα, ένα λογισμικό Web διακομιστή, ένα σύστημα διαχείρισης βάσης δεδομένων και μια γλώσσα προγραμματισμού ή script.

Κάποιες από τις επιλογές εξαρτώνται από κάποιες άλλες επιλογές. Για παράδειγμα, δεν τρέχουν όλα τα λειτουργικά συστήματα σε όλους τους υπολογιστές, ούτε οι γλώσσες script μπορούν να συνδεθούν με όλες τις βάσεις δεδομένων κ.λ.π.

Μια από τις ωραίες λειτουργίες της PHP είναι ότι είναι διαθέσιμη για τα Windows, για πολλές εκδόσεις του UNIX και για οποιοδήποτε λειτουργικό Web

διακομιστή. Η MySQL είναι εξίσου ευέλικτη. Παρακάτω αναφέρονται μερικά από τα πλεονεκτήματα της PHP και της MySQL.

### 2.6.1 Μερικά από τα πλεονεκτήματα της PHP

Κάποιοι από τους βασικούς ανταγωνιστές της PHP είναι οι Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP) και Allaire Cold Fusion.

Σε σύγκριση με αυτά τα προϊόντα, η PHP έχει πολλά πλεονεκτήματα, όπως :

- Υψηλή απόδοση
- Διασυνδέσεις με πολλά διαφορετικά συστήματα βάσεων δεδομένων
- Ενσωματωμένες βιβλιοθήκες για πολλές συνηθισμένες Web διαδικασίες
- Χαμηλό κόστος
- Ευκολία μάθησης και χρήσης
- Μεταφερισμότητα
- Διαθεσιμότητα του κώδικα προέλευσης

Ακολουθεί μια πιο λεπτομερής περιγραφή αυτών των πλεονεκτημάτων.

#### α) Απόδοση

Η PHP είναι πολύ αποτελεσματική. Με ένα φθηνό διακομιστή, μπορούν να εξυπηρετηθούν εκατομμύρια επισκέψεων καθημερινά. Οι δοκιμές που δημοσιεύθηκαν από την Zend Technologies (<http://www.zend.com>), δείχνουν ότι η PHP ξεπερνά τους ανταγωνιστές της.

#### β) Ολοκλήρωση με Βάσεις Δεδομένων

Η PHP έχει εγγενείς συνδέσεις για πολλά συστήματα βάσεων δεδομένων. Εκτός από την MySQL, μπορεί να γίνει απ' ευθείας σύνδεση μεταξύ άλλων και με τις βάσεις δεδομένων PostgreSQL, mSQL, Oracle, dbm, filePro, Hyperwave, Informix, Interbase και Sybase.

Χρησιμοποιώντας Open Database Connectivity Standard (ODBC), μπορεί κάποιος να συνδεθεί σε οποιαδήποτε βάση δεδομένων παρέχει ένα πρόγραμμα οδήγησης ODBC. Αυτό περιλαμβάνει και τα προϊόντα της Microsoft products, μεταξύ άλλων.

#### γ) Ενσωματωμένες Βιβλιοθήκες

Επειδή η PHP σχεδιάστηκε για να χρησιμοποιείται στο Web, έχει πολλές ενσωματωμένες βιβλιοθήκες, που εκτελούν χρήσιμες λειτουργίες σχετικές με το Web. Μπορεί κάποιος να δημιουργήσει εικόνες GIF δυναμικά, να συνδεθεί με άλλες υπηρεσίες δικτύων, να στέλνει ηλεκτρονικό ταχυδρομείο, να δουλεύει με cookies και να δημιουργεί PDF έγγραφα : όλα αυτά με λίγες γραμμές κώδικα.

#### **δ) Κόστος**

Η PHP είναι δωρεάν. Για την απόκτηση της τελευταίας έκδοσης, χωρίς χρέωση, μπορεί κάποιος να επισκεφθεί την ηλεκτρονική σελίδα <http://www.php.net>

#### **ε) Εκμάθηση της PHP**

Η σύνταξη της PHP βασίζεται σε άλλες γλώσσες προγραμματισμού. Κυρίως στη C και στη Perl. Αν γνωρίζει κάποιος μια από αυτές τις δύο γλώσσες, ή μια γλώσσα παρόμοια με την C, όπως την C++ ή την Java, θα αρχίσει να προγραμματίζει με την PHP σχεδόν αμέσως.

#### **στ) Μεταφερσιμότητα**

Η PHP είναι διαθέσιμη για πολλά λειτουργικά συστήματα. Μπορεί κάποιος να γράψει κώδικα PHP για δωρεάν συστήματα τύπου Unix, όπως στο Linux και στο FreeBSD, για εμπορικές εκδόσεις του Unix, όπως το Solaris και το IRIX, ή για διαφορετικές εκδόσεις των Microsoft Windows.

Ο κώδικας σε PHP συνήθως θα δουλεύει χωρίς αλλαγές στα συστήματα που τρέχουν τη συγκεκριμένη scripting γλώσσα προγραμματισμού.

#### **ζ) Κώδικας Προέλευσης**

Η πρόσβαση στον κώδικα προέλευσης της PHP είναι επιτρεπτή. Αντίθετα με εμπορικά, κλειστά προϊόντα, αν υπάρχει θέληση για μεταβολή ή προσθήκη στην γλώσσα μπορεί να γίνει.

### **2.6.2 Μερικά από τα πλεονεκτήματα της MySQL**

Μερικοί από τους μεγαλύτερους ανταγωνιστές της MySQL είναι οι PostgreSQL, Microsoft SQL Server και Oracle.

Η MySQL έχει πολλά πλεονεκτήματα, όπως υψηλή απόδοση, χαμηλό κόστος, εύκολη ενσωμάτωση στο σύστημα και εκμάθηση και ο κωδικός προέλευσης είναι διαθέσιμος (Open Source Code).

Σε σύγκριση με αυτά τα προϊόντα, η MySQL έχει πολλά πλεονεκτήματα, όπως :

- Υψηλή απόδοση
- Χαμηλό κόστος
- Ευκολία μάθησης και χρήσης
- Μεταφερσιμότητα
- Διαθεσιμότητα του κώδικα προέλευσης

Ακολουθεί μια πιο λεπτομερής περιγραφή αυτών των πλεονεκτημάτων.

**α) Απόδοση**

Η MySQL είναι χωρίς αμφιβολία γρήγορη. Μπορεί κάποιος να δει τη σελίδα δοκιμών στο <http://web.mysql.com/benchmark.html>. Πολλές από αυτές τις δοκιμές δείχνουν ότι η MySQL είναι αρκετά πιο γρήγορη από τον ανταγωνισμό.

**β) Χαμηλό Κόστος**

Η MySQL είναι διαθέσιμη δωρεάν, με άδεια ανοικτού κώδικα (Open Source) ή με χαμηλό κόστος, αν ληφθεί εμπορική άδεια την οποία απαιτεί κάποια εφαρμογή.

**γ) Ευκολία Χρήσης**

Οι περισσότερες μοντέρνες βάσεις δεδομένων χρησιμοποιούν SQL. Αν κάποιος έχει χρησιμοποιήσει ένα άλλο σύστημα διαχείρισης βάσεων δεδομένων, δεν θα πρέπει να έχει πρόβλημα να προσαρμοστεί σε αυτό. Η MySQL είναι επίσης ευκολότερη από παρόμοια προϊόντα.

**δ) Μεταφερσιμότητα**

Η MySQL μπορεί να χρησιμοποιηθεί σε πολλά διαφορετικά συστήματα UNIX, όπως επίσης και στα Microsoft Windows.

**ε) Κώδικας Προέλευσης**

Όπως και με την PHP, μπορεί κάποιος να πάρει και να τροποποιήσει τον κώδικα προέλευσης της MySQL.

Το επόμενο κεφάλαιο αναφέρεται στα εργαλεία που χρησιμοποιήθηκαν για την πρακτική υλοποίηση αυτού του περιβάλλοντος, στη δεύτερη περίπτωση, όπου έγινε χρήση της γλώσσας προγραμματισμού RUBY.

### 3. Ανάπτυξη με RUBY

Η **Ruby**<sup>[5]</sup> είναι μια δυναμική, ανακλαστική, αντικειμενοστραφής γλώσσα προγραμματισμού γενικής χρήσης που συνδυάζει μια σύνταξη επηρεασμένη από την Perl με χαρακτηριστικά από τη Smalltalk. Η Ruby προήλθε από την Ιαπωνία στα μέσα της δεκαετίας του 1990 και αρχικά σχεδιάστηκε και αναπτύχθηκε από τον Yukihiro "Matz" Matsumoto. Βασικές της επιρροές είναι η Perl, η Smalltalk, η Eiffel και η Lisp.

Η Ruby υποστηρίζει πολλαπλά παραδείγματα προγραμματισμού όπως ο συναρτησιακός προγραμματισμός, ο αντικειμενοστραφής προγραμματισμός, ο προστακτικός προγραμματισμός και ο ανακλαστικός (reflective) προγραμματισμός. Έχει σύστημα δυναμικών τύπων και αυτόματη διαχείριση μνήμης, επομένως μοιάζει σε κάποια χαρακτηριστικά της με την Python, την Perl, τη Lisp, τη Dylan, την Pike και τη CLU<sup>[6]</sup>.

Η πρότυπη υλοποίηση 1.8.7 της Ruby είναι γραμμένη σε C, σαν μια διερμηνευόμενη γλώσσα ενός περάσματος. Προς το παρόν δεν υπάρχει κάποιο επίσημο πρότυπο αναφοράς για τη γλώσσα Ruby, επομένως η αρχική υλοποίηση θεωρείται το *ντε φάκτο* σημείο αναφοράς. Υπάρχουν αρκετές (ολοκληρωμένες ή σε ανάπτυξη) εναλλακτικές υλοποιήσεις της γλώσσας, συμπεριλαμβανομένων των YARV, JRuby, Rubinius, IronRuby, MacRuby και HotRuby, κάθε μια από τις οποίες και έχει διαφορετική προσέγγιση, με τις IronRuby, JRuby και MacRuby να προσφέρουν just-in-time compilation και τη MacRuby να προσφέρει επιπλέον ahead-of-time compilation. Ο κώδικας της επίσημης έκδοσης 1.9 χρησιμοποιεί τη YARV, όπως και αυτός της έκδοσης 2.0 (σε ανάπτυξη), η οποία και θα αντικαταστήσει την πιο αργή Ruby MRI<sup>[7]</sup>.

#### 3.1 Ιστορία

Η αρχική ιδέα για τη Ruby προέκυψε στις 24 Φεβρουαρίου 1993 από το Yukihiro Matsumoto που είχε σκοπό να δημιουργήσει μια νέα γλώσσα που θα ισορροπούσε μεταξύ συναρτησιακού και προστακτικού προγραμματισμού. Ο Matsumoto έχει δηλώσει: "Ήθελα μια γλώσσα σεναρίων πιο ισχυρή από την Perl και πιο αντικειμενοστρεφή από την Python. Για αυτό αποφάσισα να σχεδιάσω τη δική μου γλώσσα".

##### 3.1.1 Ετυμολογία του ονόματος "Ruby"

Το όνομα "Ruby"<sup>[8]</sup> αποφασίστηκε στη διάρκεια μιας ηλεκτρονικής συνομιλίας μεταξύ του Matsumoto και του Keiju Ishitsuka στις 24 Φεβρουαρίου του 1993, πριν γραφεί κώδικας για τη γλώσσα. Αρχικά προτάθηκαν δύο ονόματα: "Coral" και "Ruby", με το τελευταίο να είναι η επιλογή του Matsumoto σε ένα μήνυμα ηλεκτρονικής αλληλογραφίας στη συνέχεια στον Ishitsuka. Ο Matsumoto αργότερα είχε δηλώσει ότι ένας παράγοντας υπέρ της επιλογής του ονόματος

"Ruby" ήταν το ότι βρισκόταν στην "πέτρα γενεθλίων" (birthstone) ενός από τους συναδέλφους του.

### 3.1.2 Αρχική δημοσιοποίηση

Η πρώτη δημόσια έκδοση της Ruby 0.95 ανακοινώθηκε σε εγχώριες ομάδες νέων (newsgroups) της Ιαπωνίας την 21η Δεκεμβρίου 1995. Στη συνέχεια τρεις εκδόσεις της Ruby έγιναν διαθέσιμες στο διάστημα δύο ημερών, ταυτόχρονα με τη δημιουργία της λίστας ηλεκτρονικού ταχυδρομείου ruby-list (στην Ιαπωνική γλώσσα), η οποία ήταν η πρώτη λίστα για τη νέα γλώσσα.

Ήδη σε αυτή τη φάση της ανάπτυξης ήταν παρόντα αρκετά από τα χαρακτηριστικά που έγιναν γνώριμα από τις επόμενες εκδόσεις της Ruby, όπως η αντικειμενοστρεφής σχεδίαση, οι κλάσεις με κληρονομικότητα, τα mixin, οι απαριθμητές (iterators), τα κλεισίματα (closures), ο χειρισμός εξαιρέσεων (exception handling), και η συλλογή σκουπιδιών (garbage collection).

### 3.1.3 Ruby 1.0

Η Ruby έφτασε την έκδοση 1.0 στις 25 Δεκεμβρίου 1996.

Μετά την έκδοση Ruby 1.3 το 1999, άρχισε η πρώτη λίστα ταχυδρομείου στην αγγλική γλώσσα, η ruby-talk, που υπήρξε η αρχή ενός αυξανόμενου ενδιαφέροντος για τη γλώσσα εκτός Ιαπωνίας. Το Σεπτέμβριο του 2000, τυπώθηκε το πρώτο βιβλίο στην αγγλική, το "Programming Ruby", το οποίο και αργότερα δημοσιεύθηκε ελεύθερα για το κοινό, ωθώντας και άλλο τη χρήση της Ruby στον αγγλόφωνο κόσμο.

### 3.1.4 Ruby on Rails

Το 2005, το ενδιαφέρον για τη γλώσσα Ruby εκτοξεύθηκε σε συνδυασμό με το Ruby on Rails<sup>[16]</sup>, ένα πλαίσιο ανάπτυξης Web εφαρμογών (web application framework) γραμμένο σε Ruby. Το Rails συχνά θεωρείται ότι έκανε "διάσημη" τη Ruby και συνδέεται τόσο στενά με αυτήν που συχνά οι νέοι προγραμματιστές σε Ruby θεωρούν τη γλώσσα και το πλαίσιο ταυτόσημα.

Το Ruby on Rails, συχνά Rails ή RoR<sup>[17]</sup>, είναι ένα πλαίσιο ανάπτυξης λογισμικού Ιστού ανοιχτού κώδικα για τη γλώσσα προγραμματισμού Ruby. Προορίζεται για χρήση σε συνδυασμό με ευέλικτες μεθοδολογίες ανάπτυξης (agile development methodologies), οι οποίες χρησιμοποιούνται από τους προγραμματιστές Ιστού για ταχεία ανάπτυξη λογισμικού (rapid application development).

Το Ruby on Rails<sup>[18]</sup> προήλθε από τη δουλειά του David Heinemeier Hansson στο Basecamp, ένα εργαλείο διαχείρισης project από την εταιρεία 37signals (η οποία τώρα είναι εταιρεία ανάπτυξης λογισμικού Ιστού). Ο David Hansson αρχικά

κυκλοφόρησε το Rails σαν ανοιχτό κώδικα τον Ιούλιο του 2004 αλλά δεν επέτρεπε σε άλλους προγραμματιστές να συνεισφέρουν κώδικα στο εγχείρημα μέχρι το Φεβρουάριο του 2005. Τον Αύγουστο του 2006 υπήρξε κομβικό σημείο για το Rails, όταν η Apple ανακοίνωσε ότι θα κυκλοφορούσε το Ruby on Rails μαζί με το Mac OS X v10.5 "Leopard", το οποίο κυκλοφόρησε τον Οκτώβριο του 2007.

Η έκδοση 2.3 του Rails κυκλοφόρησε στις 15 Μαρτίου του 2009. Βασικά νέα χαρακτηριστικά του Rails ήταν τα πρότυπα (templates), οι μηχανές (engines), το Rack και οι εμφωλευμένες φόρμες μοντέλων.

- Τα πρότυπα επιτρέπουν στον προγραμματιστή να δημιουργεί το σκελετό μιας εφαρμογής με ειδικά gems και ρυθμίσεις (configurations).
- Οι μηχανές επιτρέπουν τη χρήση τμημάτων εφαρμογών σε άλλες εφαρμογές, συμπεριλαμβανομένων των χαρακτηριστικών "routes", "view paths" και "models".
- Η διαπροσωπεία Rack και το Metal επιτρέπουν στον προγραμματιστή να γράφει βελτιστοποιημένα κομμάτια κώδικα που μπορούν να δρομολογούνται (route) σε σχέση με την ActionController.

Στις 23 Δεκεμβρίου 2008 ξεκίνησε το Merb, ένα άλλο πλαίσιο ανάπτυξης εφαρμογών Ιστού, και το Rails<sup>[19]</sup> ανακοίνωσε ότι επρόκειτο να συνεργαστούν. Η ομάδα του Rails ανακοίνωσαν ότι θα συνεργάζονταν με το εγχείρημα Merb για να φέρουν "τις καλύτερες ιδέες του Merb" στο Rails 3, δίνοντας τέλος στο φαινόμενο παρόμοιας δουλειάς που γινόταν σε δύο σημεία, στις δύο κοινότητες.

### 3.1.4.1 Τεχνική σύνοψη

Όπως πολλά πλαίσια Ιστού, το Rails χρησιμοποιεί αρχιτεκτονική Model-View-Controller<sup>[9]</sup> (MVC) για να οργανώσει τον προγραμματισμό των εφαρμογών.

Το Ruby on Rails περιλαμβάνει εργαλεία που διευκολύνουν κοινές προγραμματιστικές εργασίες, όπως η δημιουργία σκελετών προγραμμάτων (scaffolding) που μπορεί να δημιουργήσει αυτόματα κάποια από τα μοντέλα (models) και τις όψεις (views) που χρειάζεται μια βασική σελίδα Ιστού. Επίσης περιέχει τον WEBrick, έναν απλό εξυπηρετητή Ιστού σε Ruby, και το Rake, ένα σύστημα κατασκευής προγραμμάτων. Αυτά τα εργαλεία, σε συνδυασμό με το Rails, προσφέρουν ένα βασικό περιβάλλον ανάπτυξης.

Το Ruby on Rails βασίζεται σε έναν εξυπηρετητή Ιστού για την εκτέλεσή του. Συνήθως προτιμάται ο Mongrel έναντι του WEBrick αλλά μπορεί να χρησιμοποιηθεί και ο Lighttpd, ο Abyss, ο Apache (σαν μονάδα κώδικα - π.χ. Passenger - ή μέσω του CGI, του FastCGI ή του mod\_ruby), και πολλοί άλλοι. Από το 2008 ο εξυπηρετητής Passenger δείχνει να προτιμάται αντί για τον Mongrel. Πρόσφατα, παρατηρήθηκε συχνή χρήση του εξυπηρετητή Unicorn.

Το Rails είναι επίσης γνωστό για την εκτενή χρήση των βιβλιοθηκών JavaScript Prototype και Script.aculo.us για Ajax. Το Rails αρχικά έκανε χρήση ελαφρών κλήσεων SOAP για web services - αργότερα αυτό αντικαταστάθηκε από RESTful web services.

Από την έκδοση 2.0, το Ruby on Rails προσφέρει σαν μορφές εξόδου HTML και XML, με τη δεύτερη να χρησιμοποιείται και στα RESTful web services.

Το Ruby on Rails βασίζεται στη Ruby 1.8.6. Η έκδοση 3.0 δεν υποστηρίζει την Ruby 1.8.6 και χρειάζεται Ruby 1.8.7 για να λειτουργήσει.

#### **3.1.4.1.1 Δομή**

Το Ruby on Rails είναι χωρισμένο σε διάφορα πακέτα, το ActiveRecord (ένα σύστημα αντικειμενοστρεφούς-σχεσιακής αντιστοίχισης (object-relational mapping) για την πρόσβαση σε βάσεις δεδομένων), το ActiveResource (παρέχει web services), το ActionPack, το ActiveSupport και το ActionMailer. Πριν την έκδοση 2.0, το Rails περιλάμβανε και το πακέτο Action Web Service που τώρα αντικαθίσταται από το Active Resource. Εκτός από τα βασικά πακέτα, οι προγραμματιστές μπορούν να δημιουργήσουν plugins για να επεκτείνουν τα υπάρχοντα πακέτα.

#### **3.1.4.1.2 Εγκατάσταση και χρήση τελικών εφαρμογών**

Το Ruby on Rails συχνά εγκαθίσταται μέσω του RubyGems, ενός διαχειριστή πακέτων[16] που περιλαμβάνεται στη Ruby. Πολλές διανομές Linux υποστηρίζουν την εγκατάσταση του Rails και των εξαρτήσεών του μέσω του διαχειριστή πακέτων τους.

Το Ruby on Rails συχνά συνδέεται με μια βάση δεδομένων όπως η MySQL και έναν εξυπηρετητή Ιστού όπως ο Apache.

#### **3.1.4.2 Φιλοσοφία και σχεδίαση**

Το Ruby on Rails<sup>[10]</sup> δίνει έμφαση στις αρχές Σύμβαση έναντι Ρύθμισης (Convention over Configuration, CoC), και Μην επαναλαμβάνεσαι (Don't Repeat Yourself, DRY).

"Σύμβαση έναντι Ρύθμισης" σημαίνει ότι ο προγραμματιστής χρειάζεται μόνο να ορίζει τα μη συμβατικά χαρακτηριστικά της εφαρμογής του. Για παράδειγμα, αν στο μοντέλο υπάρχει μια κλάση Sale, ο αντίστοιχος πίνακας στη βάση θα ονομαστεί sales. Μόνο αν κάποιος αποκλίνει από αυτήν τη σύμβαση, για παράδειγμα αποκαλώντας τον "products sold", θα πρέπει να γράψει κώδικα για αυτά τα ονόματα. Συνήθως αυτό οδηγεί σε λιγότερη επανάληψη κώδικα.

"Μην επαναλαμβάνεσαι" σημαίνει ότι η πληροφορία υπάρχει πάντα σε ένα μέρος, χωρίς αμφιβολίες. Για παράδειγμα, ο προγραμματιστής που χρησιμοποιεί το



ActiveRecord, δε χρειάζεται να ορίσει τα ονόματα των στηλών των πινάκων στη βάση όταν ορίζει κλάσεις. Το Ruby on Rails βρίσκει μόνο του την πληροφορία αυτή από τη βάση δεδομένων με βάση το όνομα της κλάσης.

### 3.1.4.3 Εμπορικά σήματα

Το Μάρτιο του 2007 ο David Heinemeier Hansson έκανε τρεις αιτήσεις για εμπορικά σήματα σχετικά με το Rails στο USPTO. Αυτές οι αιτήσεις αφορούσαν τη φράση "RUBY ON RAILS", τη λέξη "RAILS" και το επίσημο λογότυπο του Rails. Αποτέλεσμα αυτών ήταν το καλοκαίρι του 2007 ο Hansson να αρνηθεί στον εκδοτικό οίκο Apress την άδεια να χρησιμοποιήσει το λογότυπο του Rails στο εξώφυλλο ενός νέου βιβλίου για το Rails που είχε γραφτεί από σημαντικά μέλη της κοινότητας. Αυτό το επεισόδιο δημιούργησε μια ήπια αντίδραση στην κοινότητα του Rails. Σε απάντηση σε αυτήν την αντίδραση ο Hansson δήλωσε το εξής:

"I only grant promotional use (of the Rails logo) for products I'm directly involved with. Such as books that I've been part of the development process for or conferences where I have a say in the execution. I would most definitely seek to enforce all the trademarks of Rails".

### 3.1.4.4 Υποδοχή

Έχουν υπάρξει αρνητικά σχόλια για την απόδοση του Rails σε βαριές εφαρμογές. Αναφέρονται διάφορες διακοπές της λειτουργίας του Twitter το 2007 και το 2008, που το ώθησαν να μεταφέρει τμήμα της υποδομής του σε κώδικα Scala (που εκτελείται στην αφηρημένη μηχανή της Java) για το σύστημα μηνυμάτων τους (message queuing system) και άλλο ενδιάμεσο κώδικα. Ο κώδικας της εμφάνισης της σελίδας εξακολουθεί να βασίζεται στο Ruby on Rails.

### 3.1.5 Ruby 1.9.1

Η Ruby 1.9.1 εισάγει κάποιες σημαντικές αλλαγές σε σχέση με την έκδοση 1.8.6. Κάποια παραδείγματα ακολουθούν:

- Τοπικές μεταβλητές (μεταβλητές που είναι ορατές μόνο στην ενότητα/block στην οποία δηλώθηκαν)
- Επιπλέον σύνταξη για λ-εκφράσεις (fun = -(a,b) { puts a + b })
- Υποστηρίζονται κωδικοποιήσεις χαρακτήρων ανά συμβολοσειρά

## 3.2 Φιλοσοφία

Ο Matsumoto έχει πει ότι η Ruby<sup>[11]</sup> έχει σχεδιαστεί για παραγωγικότητα και ευχάριστη χρήση από τον προγραμματιστή, ακολουθώντας τις αρχές της καλής σχεδίασης διεπαφών χρήστη. Τονίζει ότι η σχεδίαση συστημάτων πρέπει να δίνει έμφαση στις ανθρώπινες ανάγκες, και όχι σε αυτές του υπολογιστή:

Συχνά οι άνθρωποι, ειδικά οι μηχανικοί υπολογιστών, εστιάζουν στους υπολογιστές. Σκέφτονται: "Κάνοντας αυτό, ο υπολογιστής θα τρέξει πιο γρήγορα. Κάνοντας αυτό, ο υπολογιστής θα τρέξει πιο αποδοτικά. Κάνοντας αυτό, ο υπολογιστής θα ...". Εστιάζουν την προσοχή τους στους υπολογιστές. Αλλά στην πραγματικότητα πρέπει να εστιάσουμε στους ανθρώπου, στο πώς οι άνθρωποι προγραμματίζουν ή χειρίζονται τις εφαρμογές των υπολογιστών. Είμαστε τα αφεντικά. Είναι οι σκλάβοι.

Η Ruby λέγεται ότι ακολουθεί την "αρχή της ελάχιστης έκπληξης", το οποίο σημαίνει ότι η γλώσσα πρέπει να συμπεριφέρεται έτσι ώστε να μην μπερδεύονται οι έμπειροι χρήστες της. Ο Matsumoto έχει πει ότι ο βασικός στόχος της σχεδιάσής του ήταν να δημιουργήσει μια γλώσσα, την οποία θα απολάμβανε ο ίδιος να χρησιμοποιεί, ελαχιστοποιώντας την προγραμματιστική δουλειά και τα πιθανά διφορούμενα σημεία. Έχει πει ότι δεν είχε εφαρμόσει την αρχή της ελάχιστης έκπληξης στη Ruby, αλλά η φράση έχει πια συνδεθεί στενά με αυτήν. Η φράση η ίδια έχει υπάρξει πηγή εκπλήξεων, αφού οι αρχάριοι χρήστες πιστεύουν ότι σημαίνει ότι οι συμπεριφορές της Ruby προσπαθούν να μοιάσουν σε αυτές άλλων γνώριμων γλωσσών. Σε μια συζήτηση το Μάιο του 2005 στην ομάδα `comp.lang.ruby`, ο Matsumoto προσπάθησε να κρατήσει μια απόσταση μεταξύ της Ruby και της παραπάνω "αρχής", εξηγώντας ότι, αφού κάθε σχεδιαστική απόφαση θα καταπλήξει κάποιον, χρησιμοποιεί ένα προσωπικό μέτρο για την αποτίμηση της "έκπληξης". Αν αυτό το προσωπικό μέτρο παραμένει συνεπές, θα υπάρχουν λίγες εκπλήξεις για αυτούς που το γνωρίζουν.

Ο Matsumoto το περιέγραψε ως εξής σε μια συνέντευξη:

Κάθε άνθρωπος έχει ένα προσωπικό παρελθόν. Κάποιος μπορεί να ήρθε από την Python, κάποιος άλλος από την Perl, και μπορεί να εκπλαγούν από διάφορες όψεις της γλώσσας. Τότε έρχονται και μου λένε, 'ένιωσα έκπληξη από αυτό το χαρακτηριστικό της γλώσσας, η Ruby παραβαίνει την αρχή της ελάχιστης έκπληξης.' Περίμενε. Περίμενε. Η αρχή της ελάχιστης έκπληξης δεν είναι μόνο για εσένα. Η αρχή της ελάχιστης έκπληξης σημαίνει την αρχή της ελάχιστης δικής μου έκπληξης. Και αυτό σημαίνει ότι ισχύει όταν έχεις μάθει τη Ruby αρκετά καλά. Για παράδειγμα, ήμουν προγραμματιστής της C++ πριν αρχίσω τη σχεδίαση της Ruby. Προγραμματίζα αποκλειστικά σε C++ για δύο με τρία χρόνια. Και μετά από δύο χρόνια προγραμματισμού με C++, η γλώσσα αυτή εξακολουθεί να με εκπλήσσει.

### 3.3 Χαρακτηριστικά

- Πλήρως αντικειμενοστρεφής με κληρονομικότητα, `mixin`, και μετακλάσεις
- Δυναμικοί τύποι και `Duck typing`
- Τα πάντα είναι εκφράσεις (ακόμα και οι εντολές) και τα πάντα εκτελούνται προστακτικά (ακόμα και οι δηλώσεις)
- Σύντομη και ευέλικτη σύνταξη που ελαχιστοποιεί το "συντακτικό θόρυβο" και είναι η βάση για γλώσσες συγκεκριμένων πεδίων (`domain specific languages`)

- Δυναμική ανάκλαση και τροποποίηση αντικειμένων στο χρόνο εκτέλεσης για τη διευκόλυνση του μεταπρογραμματισμού
- Κλεισίματα, απαριθμητές και γεννήτριες, με μια μοναδική σύνταξη ενοτήτων (blocks)
- Ρητός συμβολισμός για δυναμικούς πίνακες, πίνακες κατακερματισμού (hashes), κανονικές εκφράσεις και σύμβολα
- Ενσωμάτωση κώδικα σε συμβολοσειρές (variable interpolation)
- Παράμετροι με εξορισμού τιμές (Default arguments)
- Τέσσερα επίπεδα εμβέλειας μεταβλητών: καθολική, κλάσης, στιγμιότυπου κλάσης και τοπική, που σημειώνονται με σύμβολα (sigils) και χρήση κεφαλαίων-μικρών ανά περίπτωση
- Αυτόματη συλλογή σκουπιδιών
- Συνέχειες (continuations) πρώτης τάξης
- Αυστηροί κανόνες έμμεσης μετατροπής τιμών αληθείας (τα πάντα είναι αληθή εκτός του false και του nil)
- Χειρισμός εξαιρέσεων
- Υπερφόρτωση τελεστών
- Ενσωματωμένη υποστήριξη για ρητούς αριθμούς, μιγαδικούς αριθμούς και αριθμητική άπειρης ακρίβειας
- Ρυθμιζόμενη διανομή των κλήσεων μεθόδων (dispatch) (με τις `method_missing` και `const_missing`)
- Εγγενή νήματα και συνεργατικές ίνες (cooperative fibers)
- Αρχική υποστήριξη για το πρότυπο Unicode και πολλαπλές κωδικοποιήσεις χαρακτήρων (αν και με κάποια προβλήματα μέχρι την έκδοση 1.9)
- plug-in API σε C
- Το κέλυφος Ruby
- Κεντρική διαχείριση πακέτων μέσω των RubyGems
- Έχει υλοποιηθεί σε όλες τις σημαντικές πλατφόρμες
- Μεγάλη βασική βιβλιοθήκη

### 3.4 Σημασιολογία

Η Ruby είναι αντικειμενοστρεφής<sup>[12]</sup>: κάθε τύπος δεδομένων είναι αντικείμενο, συμπεριλαμβανομένων και των κλάσεων και των τύπων που άλλες γλώσσες θεωρούν βασικούς (όπως οι ακέραιοι, οι τιμές αλήθειας και η τιμή "nil"). Κάθε συνάρτηση είναι μια μέθοδος. Οι τιμές με όνομα (μεταβλητές) πάντα είναι αναφορές σε αντικείμενα, και όχι τα ίδια τα αντικείμενα. Η Ruby υποστηρίζει κληρονομικότητα με δυναμική διανομή μεθόδων (dynamic dispatch), mixin και μεθόδους singleton (που ανήκουν και ορίζονται μόνο σε ένα αντικείμενο και όχι σε ολόκληρη την κλάση). Αν και η Ruby δεν υποστηρίζει πολλαπλή κληρονομικότητα, οι κλάσεις μπορούν να εισάγουν μονάδες κώδικα (modules) σαν mixin. Η διαδικαστική σύνταξη υποστηρίζεται αλλά όλες οι μέθοδοι που ορίζονται εκτός της εμβέλειας ενός συγκεκριμένου αντικειμένου είναι στην πραγματικότητα μέθοδοι της κλάσης Object. Επειδή η κλάση αυτή είναι η βασική κλάση κάθε άλλης βάσης, οι αλλαγές είναι ορατές σε όλες τις κλάσεις και τα αντικείμενα.

Η Ruby έχει περιγραφεί σαν γλώσσα προγραμματισμού πολλών παραδειγμάτων: επιτρέπει διαδικαστικό προγραμματισμό (ο ορισμός συναρτήσεων/μεταβλητών εκτός κλάσεων τις κάνει μέρος της κλάσης ρίζας, 'self' Object), με αντικειμενοστρεφή χαρακτηριστικά (τα πάντα είναι αντικείμενα) ή συναρτησιακά χαρακτηριστικά και συνέχειες - όλες οι εντολές έχουν τιμές, και οι συναρτήσεις επιστρέφουν την τελευταία τιμή τους). Επιτρέπει ενδοσκόπηση (introspection), ανάκλαση (reflection) και μεταπρογραμματισμό, και υποστηρίζει νήματα βασισμένα στο διερμηνέα. Η Ruby έχει δυναμικό σύστημα τύπων και υποστηρίζει τον παραμετρικό πολυμορφισμό.

Σύμφωνα με τη λίστα συχνών ερωτήσεων της Ruby:

"Αν σας αρέσει η Perl, θα σας αρέσει η Ruby και θα νιώσετε άνετα με τη σύνταξη της. Αν σας αρέσει η Smalltalk, θα σας αρέσει η Ruby και θα νιώσετε άνετα με τη σημασιολογία της. Αν σας αρέσει η Python, μπορεί (ή μπορεί και όχι) να σας ξενίσει η τεράστια διαφορά στη φιλοσοφία σχεδίασης μεταξύ της Python και των Ruby/Perl."

### 3.5 Σύνταξη

Η σύνταξη<sup>[13]</sup> της Ruby είναι παρόμοια με αυτή της Perl και της Python. Οι ορισμοί κλάσεων και μεθόδων ξεχωρίζουν από λέξεις-κλειδιά. Σε αντίθεση με την Perl, δεν είναι απαραίτητο οι μεταβλητές να έχουν στην αρχή τους ένα αναγνωριστικό σύμβολο. Όταν αυτό όμως χρησιμοποιείται αλλάζει η σημασία της εμβέλειας της μεταβλητής. Η εμφανέστερη διαφορά σε σχέση με τη C και την Perl είναι ότι συνήθως χρησιμοποιούνται λέξεις-κλειδιά για τον ορισμό λογικών ενοτήτων κώδικα, χωρίς αγκύλες (δηλ., ζευγάρι { και }). Για πρακτικούς λόγους, δεν υπάρχει διαχωρισμός μεταξύ εκφράσεων και εντολών. Οι αλλαγές γραμμής είναι σημαντικές και αποτελούν το τέλος μιας εντολής - επίσης μπορεί να χρησιμοποιηθεί ένα ελληνικό ερωτηματικό στη θέση τους. Σε αντιδιαστολή με την Python, οι εσοχές (indentation) δεν έχουν σημασία.

Μια από τις διαφορές της Ruby σε σχέση με την Python και την Perl είναι ότι η Ruby κρατά όλες τις μεταβλητές ενός αντικειμένου σαν ιδιωτικές (private) μέσα σε μια κλάση και τις εκθέτει προς τα έξω μόνο μέσω μεθόδων πρόσβασης (attr\_writer, attr\_reader, κλπ.). Σε αντίθεση με τις μεθόδους "getter" και "setter" άλλων γλωσσών όπως η C++ ή η Java, οι μέθοδοι πρόσβασης στη Ruby δημιουργούνται με μια γραμμή κώδικα με χρήση μεταπρογραμματισμού. Επειδή η κλήση αυτών των μεθόδων δε χρειάζεται παρενθέσεις, είναι εύκολο να αλλάξουμε μια μεταβλητή ενός αντικειμένου σε μια πλήρη συνάρτηση, χωρίς να αλλάξουμε ούτε μια γραμμή κώδικα ή refactoring, επιτυγχάνοντας παρόμοια λειτουργικότητα με τα μέλη ιδιοτήτων της C# και της VB.NET. Οι περιγραφείς ιδιοτήτων της Python (property descriptors) είναι παρόμοιοι αλλά έχουν ένα κόστος στη φάση της ανάπτυξης του λογισμικού. Αν κάποιος αρχίσει στην Python με μια δημόσια μεταβλητή ενός

αντικειμένου και στη συνέχεια αλλάξει την υλοποίησή της σε μια ιδιωτική μεταβλητή που φαίνεται μέσα από έναν περιγραφέα ιδιοτήτων, ο εσωτερικός κώδικας της κλάσης μπορεί να χρειαστεί να διορθωθεί ώστε να χρησιμοποιεί την ιδιωτική μεταβλητή και όχι τη δημόσια. Η Ruby αφαιρεί αυτήν την επιλογή της σχεδίασης, επιβάλλοντας όλες οι μεταβλητές αντικειμένων να είναι ιδιωτικές, αλλά παρέχει έναν απλό τρόπο να ορίζονται μέθοδοι `get` και `set`. Αυτό συμφωνεί με την ιδέα ότι στη Ruby, κανείς δεν έχει άμεση πρόσβαση στα εσωτερικά μέλη μιας κλάσης από το εξωτερικό της αλλά μπορεί μόνο να στείλει ένα μήνυμα στην κλάση και να λάβει μια απάντηση.

Δείτε τα παραδείγματα για δείγματα κώδικα που δείχνουν τη σύνταξη της Ruby.

### 3.6 Διαφορές στη συμπεριφορά σε σχέση με άλλες γλώσσες

Κάποια χαρακτηριστικά που διαφέρουν σημαντικά σε σχέση με γλώσσες όπως η C ή η Perl ή η PHP<sup>[14]</sup>:

- Η σύνταξη της γλώσσας διακρίνει μεταξύ κεφαλαίων και μικρών στα αναγνωριστικά (identifiers), θεωρώντας ότι οι μεταβλητές με κεφαλαία είναι σταθερές.
- Τα σύμβολα `$` και `@` δεν δείχνουν τον τύπο της μεταβλητής όπως στην Perl, αλλά την εμβέλειά της.
- Για να δείξετε έναν αριθμό κινητής υποδιαστολής χωρίς δεκαδικό μέρος, πρέπει να τελειώνει σε `0` (`99.0`) ή να μετατρέπεται ρητά (`99.to_f`). Δεν αρκεί να προστεθεί η τελεία (`99.`) γιατί οι αριθμοί ακολουθούν τη σύνταξη των μεθόδων.
- Η αποτίμηση τιμών αληθείας δεδομένων που δεν είναι αυτού του τύπου είναι αυστηρή: το `0`, το `""` και το `[]` αποτιμώνται σε `true`. Στη C, η έκφραση `0 ? 1 : 0` αποτιμάται σε `0` (δηλ. `false`). Στη Ruby, όμως, επιστρέφει `1`, γιατί όλοι οι αριθμοί αποτιμώνται σε `true`; μόνο το `nil` και το `false` αποτιμώνται σε `false`. Ένα αποτέλεσμα αυτού του κανόνα είναι ότι οι μέθοδοι της Ruby κατά σύμβαση — για παράδειγμα, οι αναζητήσεις κανονικών εκφράσεων — όταν επιτύχουν επιστρέφουν αριθμούς, συμβολοσειρές, λίστες ή άλλες τιμές που δεν είναι `false` αλλά επιστρέφουν `nil` όταν αποτύχουν. Αυτή η σύμβαση χρησιμοποιείται επίσης και στη Smalltalk, όπου μόνο τα ειδικά αντικείμενα `true` και `false` μπορούν να χρησιμοποιηθούν σε μια έκφραση αλήθειας.
- Οι εκδόσεις πριν την 1.9 χρησιμοποιούν απλούς ακέραιους για να αναπαραστήσουν τους χαρακτήρες, όπως στη C. Αυτό μπορεί να προκαλέσει εκπλήξεις όταν τεμαχίζονται οι συμβολοσειρές: το `"abc"[0]` επιστρέφει `97` (τον κώδικα ASCII του πρώτου χαρακτήρα της συμβολοσειράς) - για να πάρουμε `"a"` πρέπει να χρησιμοποιηθεί η `"abc"[0,1]` (υποσυμβολοσειρά μήκους 1) ή η `"abc"[0].chr`.
- Η έκφραση `statement until expression`, σε αντίθεση με τις αντίστοιχες εντολές άλλων γλωσσών (π.χ. `do { statement } while (!(expression))`; στη C/C++/...), στην πραγματικότητα δεν τρέχει ποτέ την εντολή `statement` αν η

έκφραση expression είναι ήδη αληθής. Αυτό συμβαίνει γιατί η statement until expression είναι στην πραγματικότητα μια πιο φιλική μορφή (syntactic sugar) της until expression; statement; end, η αντίστοιχη της οποίας στη C/C++ είναι η while (!(expression)) { statement; }, όπως η statement if expression είναι ισοδύναμη με την if (expression) { statement; }. Όμως, η σύνταξη begin statement end until expression στη Ruby θα εκτελέσει την εντολή statement μια φορά ακόμα και αν η έκφραση expression είναι ήδη αληθής, μια συμπεριφορά που μοιάζει με αυτή του "do-while" άλλων γλωσσών.

- Επειδή οι σταθερές είναι αναφορές σε αντικείμενα, η αλλαγή αυτού στο οποίο δείχνει μια μεταβλητή δημιουργεί μια προειδοποίηση αλλά η τροποποίηση του ίδιου του αντικειμένου όχι. Για παράδειγμα, ο κώδικας Greeting << " world!" if Greeting == "Hello" δεν εμφανίζει κάποιο μήνυμα λάθους η προειδοποίησης. Αυτό είναι παρόμοιο με τις μεταβλητές final της Java ή με τους δείκτες const σε μη-const αντικείμενα στη C++, αλλά η Ruby επιτρέπει το "πάγωμα" ενός αντικειμένου, σε αντίθεση με τη Java.

Κάποια χαρακτηριστικά που διαφέρουν σε σχέση με άλλες γλώσσες:

- Οι συνηθισμένοι τελεστές για εκφράσεις συνθήκης, and και or, δεν ακολουθούν τους κανονικούς κανόνες προτεραιότητας: ο and δεν έχει προτεραιότητα έναντι του or. Η Ruby επίσης έχει τελεστές εκφράσεων || και && που δουλεύουν με τον κλασικό τρόπο.

### 3.7 Αλληλεπίδραση

Η επίσημη διανομή της Ruby περιλαμβάνει το "irb"<sup>[15]</sup>, έναν αλληλεπιδραστικό διερμηνέα γραμμής εντολών που μπορεί να χρησιμοποιηθεί για γρήγορες δοκιμές κώδικα. Το παρακάτω κομμάτι κώδικα αναπαριστά ένα παράδειγμα χρήσης του irb:

```
<source lang="ruby"> $ irb irb(main):001:0> puts "Hello, World" Hello, World => nil
irb(main):002:0> 1+2 => 3 </source>
```

### 3.8 Υλοποιήσεις

Η έκδοση 1.9 της Ruby, αποτελείται από μια μοναδική υλοποίηση γραμμένη σε C που χρησιμοποιεί μια εικονική μηχανή ειδική για Ruby.

Η έκδοση 1.8 της Ruby έχει δυο βασικές υλοποιήσεις: τον επίσημο διερμηνέα της Ruby, ο οποίος συχνά αποκαλείται και "Ο διερμηνέας του Matz για τη Ruby" ή MRI, και είναι αυτός που χρησιμοποιείται πιο συχνά, και τη JRuby, μια υλοποίηση βασισμένη σε Java που εκτελείται στην εικονική μηχανή της Java.

Υπάρχουν και άλλες λιγότερο γνωστές ή νεότερες υλοποιήσεις όπως η Cardinal (υλοποιημένη στην εικονική μηχανή Parrot), η IronRuby, η MacRuby, η MagLev, η Rubinius, η Ruby.NET, η XRuby και η HotRuby (εκτελεί κώδικα Ruby σε φυλλομετρητές ιστού και στην πλατφόρμα Flash).

Το πόσο ώριμη είναι μια υλοποίηση της Ruby συνήθως μετράται από το αν είναι ικανή να εκτελεί το πλαίσιο Ruby on Rails (Rails), γιατί είναι πολύπλοκο στην υλοποίηση και χρησιμοποιεί πολλά χαρακτηριστικά που είναι μόνο διαθέσιμα στη Ruby. Το σημείο στο οποίο μια υλοποίηση μπορεί να το πετύχει αυτό, ονομάζεται η μοναδικότητα των Rails (The Rails singularity). Μέχρι το Μάιο του 2010, μόνο η βασική υλοποίηση (MRI) και η JRuby μπορούν να εκτελέσουν το Rails χωρίς τροποποιήσεις σε ένα περιβάλλον παραγωγής. Η Rubinius κυκλοφόρησε την έκδοση 1.0 που μπορεί να εκτελεί το Rails, αλλά δεν είναι σίγουρο ότι μπορεί να χρησιμοποιηθεί σε περιβάλλοντα παραγωγής ακόμα. Η IronRuby μπορεί να τρέξει κάποιες δοκιμαστικές περιπτώσεις του Rails, αλλά ακόμα δεν είναι έτοιμη για περιβάλλοντα παραγωγής.

Η Ruby είναι διαθέσιμη σε πολλά λειτουργικά συστήματα όπως το Linux, το Mac OS X, τα Microsoft Windows, τα Windows CE και οι πιο πολλές εκδόσεις του Unix.

Η Ruby 1.9 πρόσφατα μεταφέρθηκε στο Symbian OS 9.x.

### 3.9 Αποθήκες κώδικα και βιβλιοθήκες

Το Ruby Application Archive (RAA) και το RubyForge, λειτουργούν σαν αποθήκες κώδικα (repositories) για πολλές εφαρμογές και βιβλιοθήκες της Ruby, περιλαμβάνοντας πάνω από 7000 αντικείμενα. Αν και ο αριθμός των εφαρμογών που είναι διαθέσιμος δεν είναι ίδιος με αυτόν των κοινοτήτων της Perl ή της Python, υπάρχουν πολλά εργαλεία που ενθαρρύνουν την περαιτέρω ανάπτυξη κώδικα στη γλώσσα.

Ο RubyGems έχει γίνει ο βασικός διαχειριστής πακέτων για βιβλιοθήκες σε Ruby. Ο σκοπός του μοιάζει με το CPAN της Perl αλλά η χρήση του μοιάζει πιο πολύ με το apt-get.

Πρόσφατα πολλές παλιές και νέες βιβλιοθήκες στεγάζονται στο GitHub, το οποίο εστιάζει στο Git και είχε βασική υποστήριξη για πακετάρισμα RubyGems.

## 4. Περιγραφή της Βάσης Δεδομένων

Στο κεφάλαιο αυτό, γίνεται περιγραφή της βάσης δεδομένων των πτυχιακών εργασιών. Αναφέρονται οι πίνακες και αναλύονται τα πεδία του καθενός από αυτούς. Στη συνέχεια, περιγράφεται το πώς λειτουργεί η βάση και αναφέρονται σχετικές SQL εντολές. Τέλος, παρουσιάζεται το διάγραμμα οντοτήτων – συσχετίσεων της βάσης δεδομένων μαζί με μια σύντομη αναφορά για το τι σημαίνει το κάθε τι που αφορά το διάγραμμα αυτό.

### 4.1 Οι πίνακες γενικά

Η βάση της εφαρμογής αποτελείται από τους παρακάτω πίνακες όπου καταχωρούνται :

- **Project** : τα στοιχεία των πτυχιακών εργασιών.  
(project id, project title, teacher login, pointer, description, number of students, requested lessons, category, material, date create, date begin, date end, student code, upload file, message code, old,percent)
- **Teacher** : τα στοιχεία των καθηγητών.  
(teacher login, password, teacher name, is admin, last created )
- **Student** : τα στοιχεία των φοιτητών.  
(student code, student names, student email)
- **Examination** : τα στοιχεία που αφορούν την εξέταση της πτυχιακής.  
(project id, adviser, examiner, final gradation, details)

### 4.2 Οι πίνακες αναλυτικά

- **Project**
  - Project id :Κωδικός πτυχιακής. Δημιουργείτε αυτόματα με την βοήθεια ενός μηχανισμού και αποτελείτε από τα πέντε πρώτα γράμματα του username του καθηγητή (αν το username του καθηγητή έχει μέγεθος τρία ή τέσσερα γράμματα τότε τοποθετούνται δύο ή ένα μηδενικά μπροστά, αντίστοιχα. Π.χ. maya0, stef0. Ο λόγος που γίνεται αυτή η προσθήκη, είναι για να έχει ο κωδικός σταθερό μέγεθος),μια κάτω παύλα, ένα λατινικό X ή ένα λατινικό E (Χειμερινό ή Εαρινό Εξάμηνο αντίστοιχα), το τρέχον ακαδημαϊκό έτος υπό τη μορφή 00-01,μια ακόμα κάτω παύλα και τέλος έναν Α.Α. (ο οποίος είναι πάντοτε τριψήφιος , υπολογίζεται με την βοήθεια του δείκτη last created του πίνακα Teacher και είναι ένας δείκτης που αφορά τον ίδιο τον καθηγητή και μόνο). Παράδειγμα : **achat\_X04-05\_008**  
Τύπος : varchar(16)



- Project title : Τίτλος Πτυχιακής. Μαζί με το Project id, αποτελούν τα κύρια κλειδιά του πίνακα.  
Τύπος : varchar(150)
- Teacher login : Είναι το username του καθηγητή που εισηγείται της πτυχιακής.  
Τύπος : varchar(8)
- Pointer : Δείκτης κατάστασης πτυχιακής. Αρχικά όταν δημιουργούμε μια πτυχιακή ο pointer παίρνει default τιμή ίση με μηδέν. Κατόπιν, όταν ανατεθεί μια προτεινόμενη πτυχιακή, παύει να είναι 'Προτεινόμενη' και γίνεται 'Σε Εξέλιξη', δηλαδή ο pointer παίρνει τη τιμή 1. Όταν κάποια στιγμή ολοκληρωθεί η πτυχιακή και κληθεί η διαδικασία δήλωσης ολοκλήρωσης πτυχιακής, τότε η πτυχιακή παύει να είναι 'Σε Εξέλιξη' και γίνεται 'Ολοκληρωμένη', δηλαδή ο pointer παίρνει τη τιμή 2.  
Τύπος : int(11)
- Description : Περιγραφή της πτυχιακής.  
Τύπος : text
- Number of students : Αριθμός φοιτητών για ανάθεση πτυχιακής.  
Τύπος : varchar(20)
- Requested lessons : Προαπαιτούμενα μαθήματα.  
Τύπος : varchar(80)
- Category : Κατηγορία που ανήκει η πτυχιακή.  
Τύπος : varchar(80)
- Material : Βιβλιογραφία  
Τύπος : varchar(100)
- Date create : Ημερομηνία δημιουργίας πτυχιακής.  
Τύπος : date
- Date begin : Ημερομηνία έναρξης / ανάθεσης πτυχιακής  
Τύπος : date
- Date end : Ημερομηνία λήξης / ολοκλήρωσης πτυχιακής.  
Τύπος : date
- Student code : Κωδικός φοιτητή ή ομάδας φοιτητών που αναλαμβάνουν τη πτυχιακή  
Τύπος : varchar(30)

- UploadFile : Όνομα σχετικού αρχείου προς φόρτωση.  
Τύπος : varchar(100)
- M\_code : Κωδικός Πτυχιακής για Διαχείριση από πλευράς Φοιτητή.  
Τύπος : varchar(30)
- Old : Δείκτης περιόδου πτυχιακής. Αν θέλει κάποιος καθηγητής να εισάγει πτυχιακές παλαιάς περιόδου για να τις έχει όλες μαζί, επιλέγει 'εισαγωγή ολοκληρωμένης πτυχιακής', τις εισάγει και ο δείκτης old παίρνει τιμή 1. Αυτό τον κάνει να τις ξεχωρίζει από τις ολοκληρωμένες που δημιουργούνται κατά την διαδικασία δήλωσης ολοκλήρωσης πτυχιακής καθώς οι πρώτες μπορούν μετέπειτα να μεταβληθούν ή να διαγραφούν ενώ οι δεύτερες όχι.  
Τύπος : int(11)
- Percent : Ποσοστό Εξέλιξης Πτυχιακής  
Τύπος : int(3)
- **Teacher**
  - Teacher login : Το username του καθηγητή. Το κύριο κλειδί του πίνακα.  
Τύπος : varchar(30)
  - Password : Το password του καθηγητή.  
Τύπος : varchar(8)
  - Teacher name : Το ονοματεπώνυμο του καθηγητή.  
Τύπος : varchar(30)
  - Is admin : Εάν ο καθηγητής είναι και διαχειριστής του συστήματος τότε η μεταβλητή αυτή έχει τη τιμή 1. Διαφορετικά έχει τη τιμή 0.  
Τύπος : int(1)
  - Last created : Το πεδίο αυτό λειτουργεί σαν δείκτης. Τοποθετείται το Project id της πιο προσφάτως δημιουργημένης πτυχιακής του καθηγητή. Για να υπολογίζεται σωστά και ξεχωριστά για τον κάθε καθηγητή ο Α.Α. που αποτελεί το τελευταίο κομμάτι κατά την σύνθεση του Project id, κρατάμε τον κωδικό της τελευταίας πτυχιακής και κατά τη δημιουργία μιας επομένης, κρατάμε τους τρεις τελευταίους χαρακτήρες (που αποτελούν και τον Α.Α.), προσθέτουμε το νούμερο κατά ένα και βρίσκουμε τον αμέσως επόμενο Α.Α. για τον συγκεκριμένο καθηγητή.  
Τύπος : varchar(16)

- **Student**
  - Student code : Κωδικός φοιτητή ή ομάδας φοιτητών που αναλαμβάνουν τη πτυχιακή  
Τύπος : varchar(30)
  - Student names : Ονοματεπώνυμο φοιτητή ή φοιτητών που αναλαμβάνουν πτυχιακή.  
Τύπος : varchar(100)
  - Student email : Ηλεκτρονικό ταχυδρομείο φοιτητή.  
Τύπος : varchar(50)
  
- **Examination**
  - Project id : Κωδικός πτυχιακής.  
Τύπος : varchar(16)
  - Adviser : Ονοματεπώνυμο εισηγητή.  
Τύπος : varchar(50)
  - Examiner : Ονοματεπώνυμο εξεταστή.  
Τύπος : varchar(50)
  - Final gradation : Τελική βαθμολογία πτυχιακής που προκύπτει από τον μέσο όρο της βαθμολογίας των εξεταστών.  
Τύπος : varchar(5)
  - Details : Παρατηρήσεις και σχόλια από τους εξεταστές μετά την παρουσίαση της πτυχιακής.  
Τύπος : longtext

### 4.3 Η βάση περιγραφικά

Κατά την είσοδο καθηγητή, ζητείται το username και το password από τον server της σχολής. Μετά από έγκυρη εισαγωγή στοιχείων, αν ο καθηγητής μπαίνει στην ιστοσελίδα για πρώτη φορά, εμφανίζεται μια φόρμα εισαγωγής ονόματος και επιθέτου του καθηγητή. Συγκεκριμένα, γίνεται έλεγχος στον πίνακα Teacher, αν υπάρχει εγγραφή με teacher login ίσο με το username που εισήγαγε ο καθηγητής. Αν συμβαίνει αυτό, τότε δεν εμφανίζεται καθόλου η φόρμα εισαγωγής προσωπικών στοιχείων. Το πεδίο last created του πίνακα Teacher παραμένει κενό έως ότου ο καθηγητής, δημιουργήσει κάποια πτυχιακή. Δηλαδή όπως μπορεί να γίνει αντιληπτό, το πεδίο αυτό λειτουργεί ως δείκτης, επισημαίνοντας την τελευταία πτυχιακή που δημιουργήθηκε από τον καθηγητή. Ο λόγος για τον οποίο συμβαίνει

αυτό, θα εξηγηθεί λίγο παρακάτω, στο κομμάτι που αναφέρεται στην δημιουργία του κωδικού πτυχιακής (project id).

Στη συνέχεια ο καθηγητής μπορεί να δημιουργήσει είτε προτεινόμενη πτυχιακή, είτε ολοκληρωμένη που έχει όμως δημιουργηθεί, εξελιχθεί και ολοκληρωθεί κατά το παρελθόν και υπήρξε εισηγητής, ο ίδιος.

Ο τρόπος με τον οποίο μπορούμε να ξεχωρίσουμε μια ολοκληρωμένη πτυχιακή και μια ολοκληρωμένη πτυχιακή που εξελίχθηκε κατά το παρελθόν, είναι η χρήση ενός δείκτη που ονομάζεται old και βρίσκεται στον πίνακα Project. Έχει δύο καταστάσεις. Όταν είναι 0, τότε η πτυχιακή ολοκληρώθηκε κατά την παρούσα περίοδο σταδιακά. Δηλαδή δημιουργήθηκε μέσα από το σύστημα ως προτεινόμενη, στη συνέχεια ανατέθηκε σε φοιτητή ή ομάδα φοιτητών και τελικά ολοκληρώθηκε. Ενώ όταν είναι 1, τότε η πτυχιακή ολοκληρώθηκε κατά το παρελθόν και δηλώνεται απ' απευθείας ως ολοκληρωμένη στο σύστημα.

Για τις ολοκληρωμένες πτυχιακές του παρελθόντος παρέχεται και η δυνατότητα διαγραφής και μεταβολής των στοιχείων. Αντίθετα στις ολοκληρωμένες πτυχιακές που εξελίχθηκαν σταδιακά, η μεταβολή και διαγραφή των στοιχείων τους, δεν επιτρέπεται. Ο λόγος είναι καθαρά τυπικός, καθώς δεν θέλουμε να γίνεται καμιά παραβίαση στοιχείων σε πτυχιακές που εξελίχθηκαν και ολοκληρώθηκαν μέσα από αυτό το περιβάλλον.

Γενικά, για τις πτυχιακές υπάρχει ένας πολύ βασικός δείκτης, ο pointer. Αυτός ο δείκτης μπορεί να πάρει τρεις τιμές. Αν ο δείκτης έχει τη τιμή 0, τότε μιλάμε για προτεινόμενη πτυχιακή, αν ο δείκτης έχει τη τιμή 1, τότε μιλάμε για πτυχιακή σε εξέλιξη, ενώ αν ο δείκτης έχει τη τιμή 2, τότε μιλάμε για ολοκληρωμένη πτυχιακή.

Είναι πολύ σημαντικό το γεγονός πως ανάλογα με τα υπάρχοντα δεδομένα, εμφανίζονται και οι ανάλογες επιλογές για τον καθηγητή. Αυτό σημαίνει πως αν ο συγκεκριμένος καθηγητής δεν έχει προτεινόμενες πτυχιακές, θα του παρέχεται μόνο η επιλογή "προσθήκη νέου θέματος" διαφορετικά του παρέχονται και οι επιλογές, "μεταβολή θέματος" και "διαγραφή θέματος". Ο λόγος είναι αυτονόητος καθώς πώς να μεταβάλεις ή να διαγράψεις κάτι που δεν υπάρχει. Πάνω σ' αυτή τη λογική, βασίζεται ολόκληρο το περιβάλλον.

Επιλέγοντας ο καθηγητής "προσθήκη νέου θέματος", εισάγει τα εξής στοιχεία:

- Τίτλος πτυχιακής
- Αριθμός σπουδαστών
- Περιγραφή
- Προαπαιτούμενα μαθήματα
- Κατηγορία
- Βιβλιογραφία

## Σύγκριση scripting γλωσσών προγραμματισμού και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

---

Τα στοιχεία αυτά εισάγονται σε μια νέα εγγραφή στον πίνακα Project. Ο δείκτης pointer παίρνει αυτομάτως την τιμή 0 και η ημερομηνία δημιουργίας την τρέχουσα ημερομηνία.

Ακολουθούν οι εντολές SQL που χρησιμοποιούνται για τη δημιουργία νέας εγγραφής στον πίνακα project :

```
$query="insert into project  
  
values('$code','$title','$valid_user',0,'$description','$nos','$pre_lessons','$category','$  
$material',' ',' ',' ',' ',' ',' ');  
  
$sql="update project set date_create = curdate() where project_title='$title'";
```

Ο κωδικός πτυχιακής, project id, δημιουργείται αυτόματα, με την βοήθεια ενός μηχανισμού. Συγκεκριμένα, αποτελείται από τέσσερα μέρη :

- Στο πρώτο μέρος, εισάγονται τα πέντε πρώτα λατινικά γράμματα του username του καθηγητή που δημιουργεί την πτυχιακή. Αν το username του καθηγητή αποτελείται από τρία ή τέσσερα γράμματα, συμπληρώνεται η πεντάδα με μηδενικά. (Αυτό συμβαίνει για να έχουμε σταθερό μέγεθος κωδικού, 16 χαρακτήρες). Ακολουθεί το σύμβολο της κάτω παύλας '\_ '.
- Στο δεύτερο μέρος, εισάγεται το λατινικό X ή το λατινικό E για τρέχον χειμερινό ή εαρινό εξάμηνο αντίστοιχα.
- Στο τρίτο μέρος, εισάγεται το τρέχον ακαδημαϊκό έτος υπό τη μορφή 02-03. Ακολουθεί το σύμβολο της κάτω παύλας '\_ '.
- Στο τέταρτο μέρος, εισάγεται ο Α.Α. της πτυχιακής που αντιστοιχεί στον συγκεκριμένο καθηγητή. Ο αριθμός αυτός έχει πάντοτε σταθερό μήκος, 3 χαρακτήρες. Π.χ. 005. Η αρίθμηση των πτυχιακών για τον κάθε καθηγητή ξεχωριστά γίνεται με την βοήθεια του πεδίου last created του πίνακα teacher, το οποίο κρατά την τελευταία πτυχιακή που δημιούργησε ο καθηγητής, κρατά τα τρία τελευταία στοιχεία που είναι ο Α.Α. και τον προσθέτει κατά ένα για να βρει τον Α.Α. της επομένης.

Παράδειγμα κωδικού πτυχιακής : achat\_X04-05\_020

Επόμενο σενάριο, αποτελεί η περίπτωση όπου φοιτητής εντοπίζει, μέσα από την λίστα πτυχιακών που του παρέχεται κατά την είσοδο φοιτητή, προτεινόμενο θέμα πτυχιακής που τον ενδιαφέρει να εκπονήσει. Σε τούτη τη περίπτωση ή στέλνει email για να δηλώσει το ενδιαφέρον του ή βρίσκει τον ίδιο τον καθηγητή και το συζητάει.

Ο καθηγητής όταν θέλει να αναθέσει μια προτεινόμενη πτυχιακή, επιλέγει μέσα από “τις πτυχιακές μου” και τις “προτεινόμενες πτυχιακές”, “ανάθεση

πτυχιακής”. Όταν γίνεται ανάθεση μιας πτυχιακής, ζητούνται από τον καθηγητή τα παρακάτω στοιχεία :

- Ονοματεπώνυμο φοιτητή ή φοιτητών που αναλαμβάνουν τη πτυχιακή. (Στη περίπτωση ομάδας φοιτητών, ζητείται να εισάγεται “παύλα” ανάμεσα).
- Email φοιτητή ή ομάδας φοιτητών (Ζητείται ENA email και όχι περισσότερα).

Στη συνέχεια τα πεδία pointer, date\_begin (ημερομηνία έναρξης) και m\_code του πίνακα project ενημερώνονται.

Επίσης δημιουργείται νέα εγγραφή στον πίνακα student με τα στοιχεία του φοιτητή ή των φοιτητών που ανέλαβαν τη πτυχιακή.

Συγκεκριμένα, ο pointer παίρνει τη τιμή 1 (περνάει δηλαδή σε κατάσταση “Σε Εξέλιξη”), το date\_begin παίρνει την τρέχουσα ημερομηνία και το m\_code παίρνει αυτομάτως μια τιμή μέσα από ένα μηχανισμό και αντιστοιχεί στο κωδικό πτυχιακής που λαμβάνει ο φοιτητής για να την διαχειρίζεται

Ακολουθούν οι εντολές Sql που χρησιμοποιούνται για την ενημέρωση εγγραφής στον πίνακα project και τη δημιουργία νέας εγγραφής στον πίνακα project :

```
$sql="update project set s_code='$s_code', pointer=1, m_code='$code' where project_title='$titles';"
```

```
$sql="update project set date_begin = curdate() where project_title='$titles';"
```

```
$sql2="insert into student values('$s_code','$s_names','$s_email');"
```

Μια πτυχιακή “σε εξέλιξη” μπορεί ένας καθηγητής και να μεταβάλει αλλά και να διαγράψει. Αν ο φοιτητής παρατήρει τη πτυχιακή που είχε αναλάβει, δίχως να την ολοκληρώσει, τότε ο καθηγητής επιλέγει “ακύρωση & επαναφορά στις προτεινόμενες”. Τούτο έχει ως αποτέλεσμα, η πτυχιακή να μετατρέπεται ξανά ως “προτεινόμενη”. Συγκεκριμένα στη βάση:

Για τον πίνακα project,

- ο pointer μηδενίζεται και τα πεδία s\_code, m\_code και date\_begin εκκενώνονται.

```
Εντολή SQL : $sql = "update project set pointer=0, date_begin = from_unixtime('$date'), s_code=",
```

```
m_code=" where project_title='$titles';"
```

Για τον πίνακα student,

- η εγγραφή με s\_code ίσο με το s\_code της πτυχιακής, διαγράφεται. Κοινώς, διαγράφονται τα στοιχεία σπουδαστών που σχετίζονταν με τη συγκεκριμένη πτυχιακή.

```
Εντολή SQL : $sql7 = "delete from student where s_code='$s_code';"
```

Σε περίπτωση πάντως που όλα κυλήσουν ομαλά και ο φοιτητής δεν παρατήρει την πτυχιακή που ανέλαβε, το επόμενο βήμα είναι η “δήλωση ολοκλήρωσης πτυχιακής”. Κατά την διαδικασία αυτή, εισάγονται τα παρακάτω στοιχεία :

- Η ημερομηνία λήξης της πτυχιακής,

Σύγκριση scripting γλωσσών προγραμματισμού  
και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

---

- τα ονοματεπώνυμα των εξεταστών
- η τελική βαθμολογία της εργασίας και
- παρατηρήσεις

Η ημερομηνία λήξης δεν εισάγεται αυτόματα με τη τρέχουσα ημερομηνία διότι ο καθηγητής μπορεί να μην κάνει συχνή χρήση της ιστοσελίδας και να ενημερώσει πολύ αργότερα την ολοκλήρωση της πτυχιακής.

Στη βάση :

Για τον πίνακα project,

- ο pointer παίρνει τη τιμή 2, το πεδίο m\_code εκκενώνεται και η ημερομηνία λήξης παίρνει τη τιμή που εισήγαγε ο καθηγητής.  
Εντολή SQL: `$sql = "update project set pointer=2, date_end = from_unixtime('$date'), m_code="`  
`where project_title='$titles';`

Για τον πίνακα examination,

- εισάγονται τα στοιχεία εξέτασης της πτυχιακής. Συγκεκριμένα δημιουργείται νέα εγγραφή με τα ονοματεπώνυμα των εξεταστών καθώς και με τη τελική βαθμολογία της εργασίας. Ως κωδικός της εγγραφής, εισάγεται αυτομάτως ο κωδικός της πτυχιακής.  
Εντολή SQL: `$sql3="insert into examination`  
`values('$score','$examiner1','$examiner2','$gradation')";`

Ο καθηγητής έχει τη δυνατότητα να δημιουργήσει απευθείας ολοκληρωμένες πτυχιακές. Σε τούτη τη περίπτωση εισάγονται απευθείας σε μια νέα εγγραφή στον πίνακα project τα παρακάτω στοιχεία :

- Τίτλος πτυχιακής
- Αριθμός φοιτητών
- Περιγραφή
- Προαπαιτούμενα μαθήματα
- Κατηγορία
- Βιβλιογραφία
- Ημερομηνία δημιουργίας
- Ημερομηνία έναρξης
- Ημερομηνία λήξης

Στον πίνακα student :

- Ονοματεπώνυμο φοιτητή ή φοιτητών που αναλαμβάνουν τη πτυχιακή. (Στη περίπτωση ομάδας φοιτητών, ζητείται να εισάγεται “παύλα” ανάμεσα).
- Email φοιτητή ή ομάδας φοιτητών (Ζητείται ENA email και όχι περισσότερα).

Στον πίνακα examination :

- τα ονοματεπώνυμα των εξεταστών
- η τελική βαθμολογία της εργασία και
- παρατηρήσεις

Σύγκριση scripting γλωσσών προγραμματισμού  
και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

---

Στον πίνακα project ο δείκτης pointer παίρνει αυτομάτως την τιμή 2 και ο δείκτης old τη τιμή 1. Ακολουθούν οι εντολές SQL που χρησιμοποιούνται για τη δημιουργία ολοκληρωμένης πτυχιακής :

```
$query="insert into project  
values('$code','$title','$valid_user',2,'$description','$nos','$pre_lessons','$category',  
'$material',' ','','$s_code',' ','1');" ;
```

```
$sql = "update project set date_create = from_unixtime('$date1'), date_begin =  
from_unixtime('$date2'),  
date_end = from_unixtime('$date3') where project_title='$title'";
```

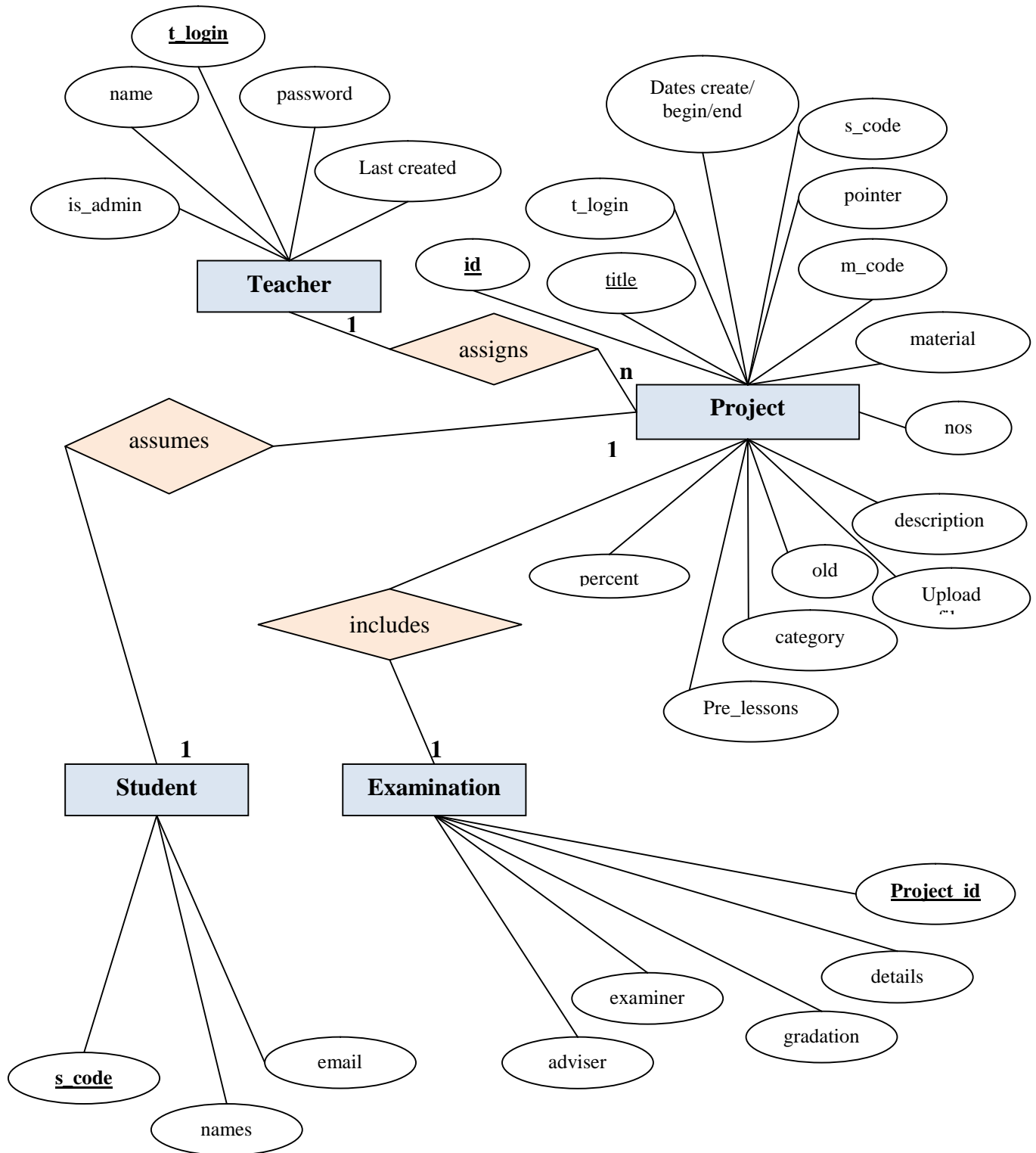
```
$sql7="insert into student values('$s_code','$s_names','$s_email');" ;
```

```
$sql2="insert into examination values('$code','$examer1','$examer2','$gradation','$details');" ;
```

```
$query2="update teacher set last_created='$code' where t_login='$valid_user'";
```



#### 4.4 Διάγραμμα Οντοτήτων Συσχετίσεων της βάσης δεδομένων



Εικόνα 4.1

## 5. Δομή του ιστοτόπου

Στο κεφάλαιο αυτό παρουσιάζεται ένα γενικό πλάνο και μια σύντομη περιγραφή της δομής του ιστοτόπου.

Το κεφάλαιο χωρίζεται σε τρία βασικά μέρη :

- Κεντρική σελίδα
- Επιλογές φοιτητή
- Επιλογές καθηγητή

Το γενικό πλάνο που αφορά το κάθε μέρος ξεχωριστά, δίνει μια απλή και εύκολα κατανοητή εικόνα για το πώς μπορεί κάποιος να κάνει πλοήγηση στον ιστοτόπο και τι επιλογές μπορεί να συναντήσει.

Για όλες αυτές τις επιλογές υπάρχει μια σύντομη περιγραφή που αποσκοπεί στην εύκολη κατανόηση τους.

Περισσότερες λεπτομέρειες και πληροφορίες δίνονται στο ένθετο εγχειρίδιο χρήσης της πτυχιακής αυτής.

### 5.1 Σύντομη περιγραφή της δομής του ιστοτόπου

#### 5.1.1 Κεντρική Σελίδα

Στην κεντρική σελίδα διακρίνονται 2 βασικές επιλογές :

- Είσοδος Φοιτητών
- Είσοδος Καθηγητών

Όπως φαίνεται και από τα σχήματα, κατά την είσοδο του καθηγητή και μόνο ζητούνται τα username και password που έχει για τον server της σχολής. Αντιθέτως, ο φοιτητής έχει απευθείας πρόσβαση στην ιστοσελίδα.

Ο διαχωρισμός Καθηγητή και Διαχειριστή Καθηγητή οφείλεται στο γεγονός ότι δεν υπάρχει ξεκάθαρος διαχωρισμός των username καθηγητών και φοιτητών με τη χρήση κάποιου δείκτη, πράγμα που σημαίνει ότι θα μπορούσε κάποιος φοιτητής να εισχωρήσει στο σύστημα ως καθηγητής. Επομένως έχει ορισθεί ένας καθηγητής ως administrator και έχει μια επιπλέον επιλογή που είναι η **Διαχείριση Χρηστών**.

Μ' αυτή την επιλογή, μπορεί ουσιαστικά να ενημερώνει κάθε εξάμηνο το σύστημα για τυχών νέους καθηγητές. Στη περίπτωση που κάποιοι καθηγητές φεύγουν, μπορεί να τους διαγράψει από το σύστημα. Όπως είναι προφανές, διαγράφονται μαζί και όλες οι προτεινόμενες και σε εξέλιξη πτυχιακές τους, ενώ οι

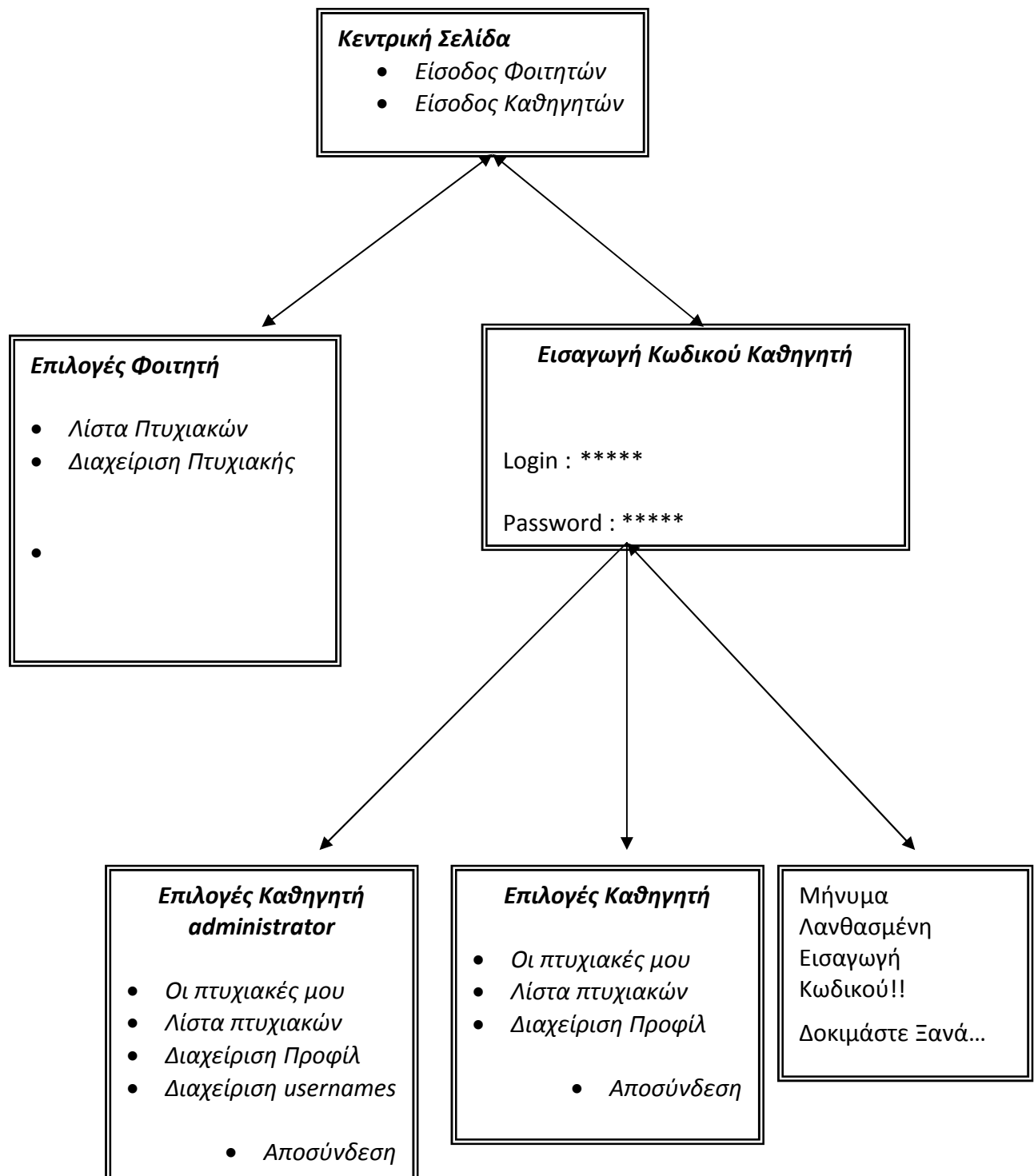
Σύγκριση scripting γλωσσών προγραμματισμού  
και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

---

ολοκληρωμένες τους, παραμένουν στο αρχείο χωρίς όμως πλέον να εμφανίζεται το email τους για επικοινωνία.

Ο ρόλος του Διαχειριστή στο σύστημα είναι πολύ εύκολος και σύντομος. Στο επόμενο κεφάλαιο αναφέρονται τα σημεία κώδικα που αφορούν τον διαχειριστή ώστε να μπορεί να γίνει είτε αλλαγή του username που θα είναι ο διαχειριστής, είτε πλήρης απενεργοποίηση του Διαχειριστή σε περίπτωση εύρεσης εναλλακτικής λύσης.

Στη συνέχεια ακολουθεί το πλάνο που αφορά την κεντρική σελίδα.



### 5.1.2 Επιλογές Φοιτητή

Ο φοιτητής κατά την είσοδο του στην ιστοσελίδα, αντικρίζει δύο επιλογές :

- Λίστα πτυχιακών
- Διαχείριση Πτυχιακής

Η “λίστα πτυχιακών” όπως φαίνεται και από τα σχήματα του γενικού πλάνου, είναι μια λίστα από πτυχιακές :

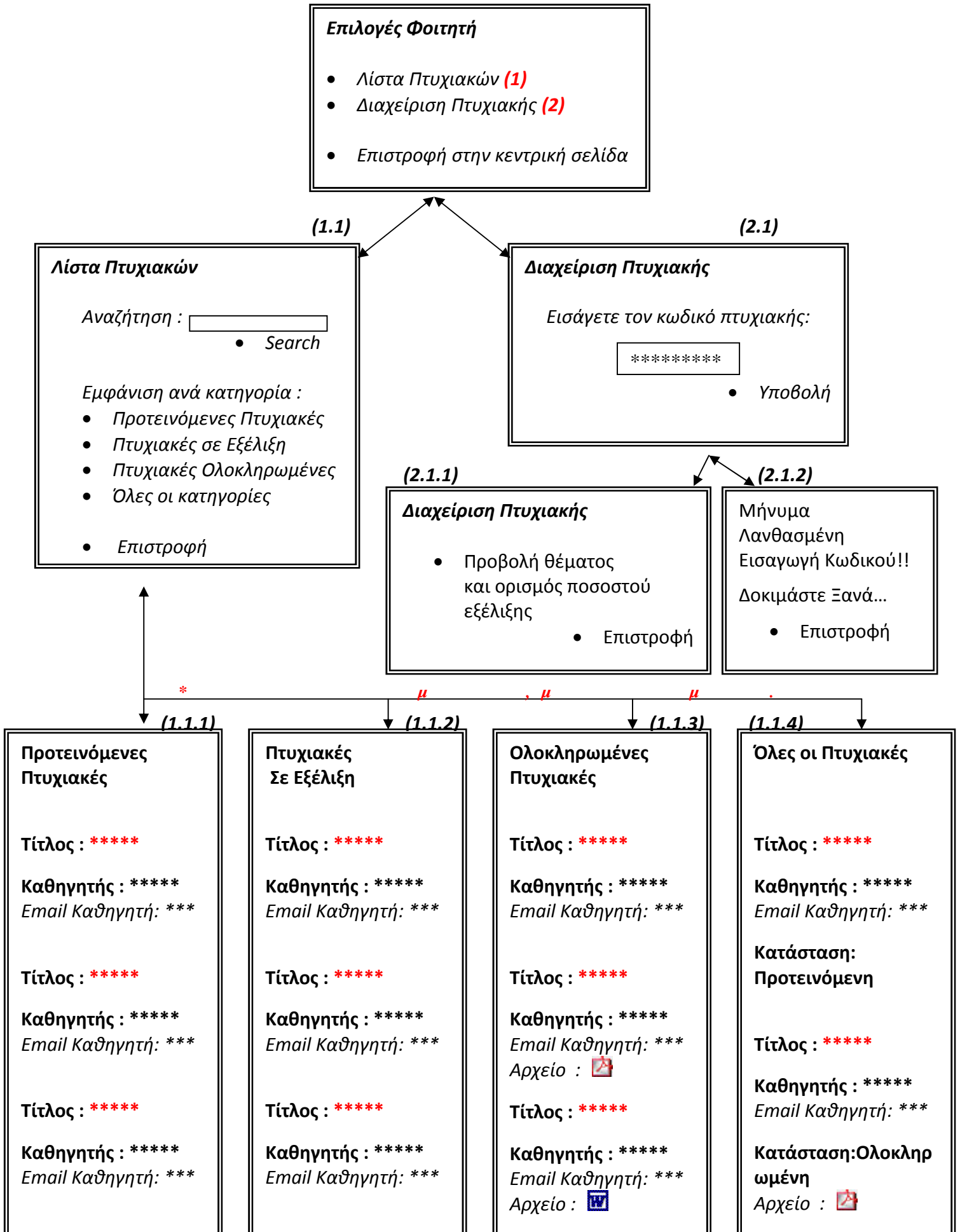
- Προτεινόμενες
- Σε εξέλιξη
- Ολοκληρωμένες
- Όλες οι κατηγορίες

Η επιλογή αυτή περιλαμβάνει και μια μηχανή αναζήτησης με τη χρήση λέξης ή πρότασης κλειδί καθώς και επιλογές ταξινομημένης εμφάνισης. Παρόμοια επιλογή έχουν και οι καθηγητές.

Η “Διαχείριση Πτυχιακής” είναι μια επιλογή που αφορά αποκλειστικά όσους φοιτητές έχουν μια πτυχιακή που είναι σε εξέλιξη. Απαιτείται η εισαγωγή κωδικού που αποστέλλεται με email από τον καθηγητή κατά την ανάθεση πτυχιακής. Είναι ένα εργαλείο που παρέχει δυνατότητα διαχείρισης της πτυχιακής από τον φοιτητή.

Στη συνέχεια ακολουθεί το πλάνο που αφορά τις επιλογές φοιτητή.

Σύγκριση scripting γλωσσών προγραμματισμού  
και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού



### 5.1.3 Επιλογές Καθηγητή

Ο καθηγητής κατά την είσοδο του στην ιστοσελίδα, πρέπει να εισάγει έγκυρα τα username και password του στον server της σχολής. Στη συνέχεια, αντικρίζει τρεις βασικές επιλογές :

- Οι πτυχιακές μου
- Λίστα πτυχιακών
- Διαχειριση Προφίλ

Η “λίστα πτυχιακών” είναι παρόμοια με αυτή των φοιτητών.

Η επιλογή “Οι πτυχιακές μου” παρέχουν συγκεκριμένες επιλογές για κάθε τύπο πτυχιακής του καθηγητή.

Συγκεκριμένα, για τις προτεινόμενες πτυχιακές, οι επιλογές είναι :

- Προσθήκη νέου θέματος
- Μεταβολή θέματος
- Διαγραφή θέματος
- Ανάθεση πτυχιακής

Για τις πτυχιακές σε εξέλιξη, οι επιλογές είναι :

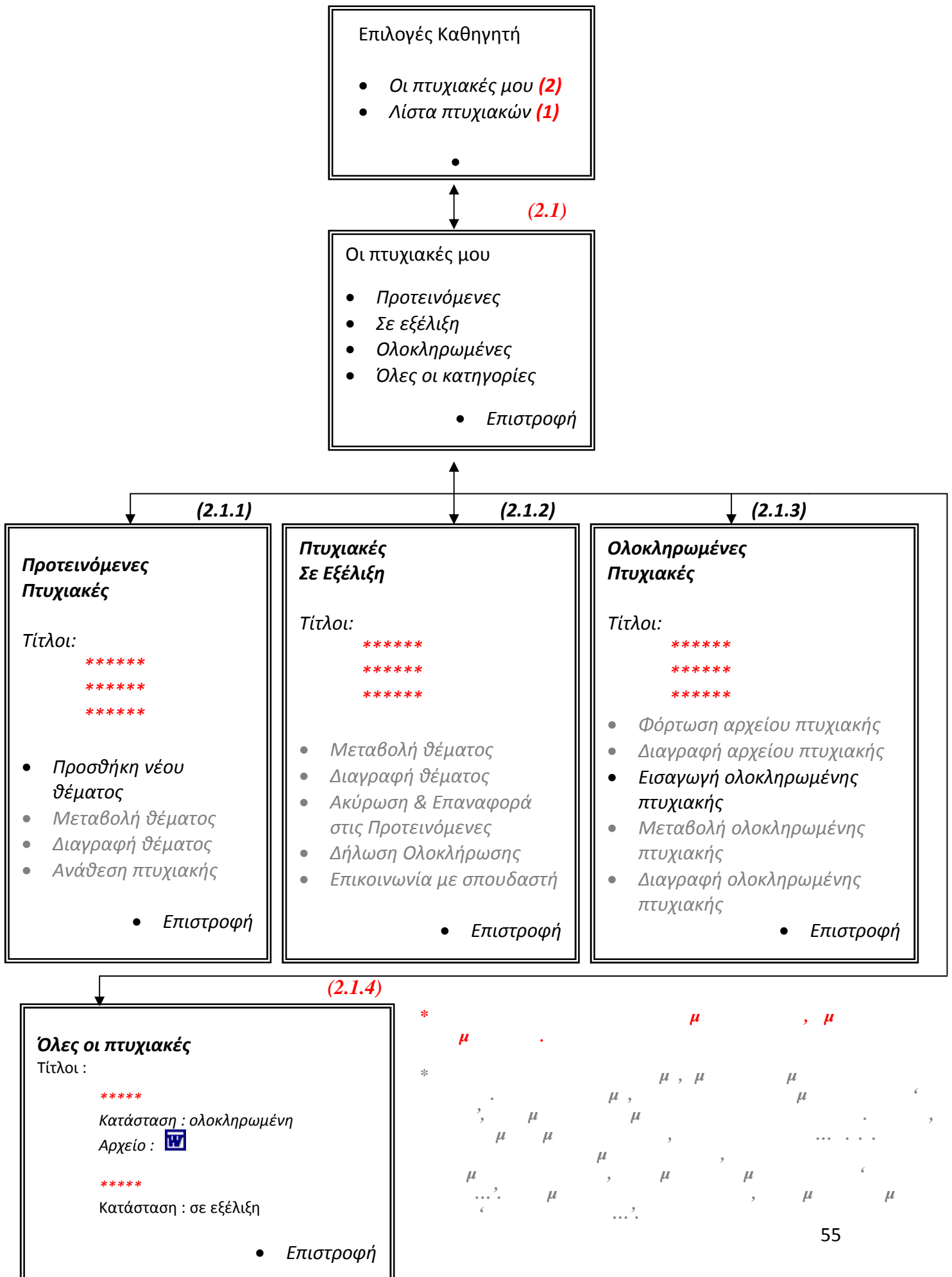
- Μεταβολή θέματος
- Διαγραφή θέματος
- Ακύρωση και επαναφορά στις προτεινόμενες
- Δήλωση ολοκλήρωσης πτυχιακής
- Επικοινωνία με φοιτητή

Τέλος, για τις ολοκληρωμένες πτυχιακές, οι επιλογές είναι :

- Φόρτωση αρχείου πτυχιακής
- Διαγραφή αρχείου πτυχιακής
- Εισαγωγή ολοκληρωμένης πτυχιακής
- Μεταβολή ολοκληρωμένης πτυχιακής
- Διαγραφή ολοκληρωμένης πτυχιακής

Στη συνέχεια ακολουθεί το πλάνο που αφορά τις επιλογές καθηγητή.

Σύγκριση scripting γλωσσών προγραμματισμού  
και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

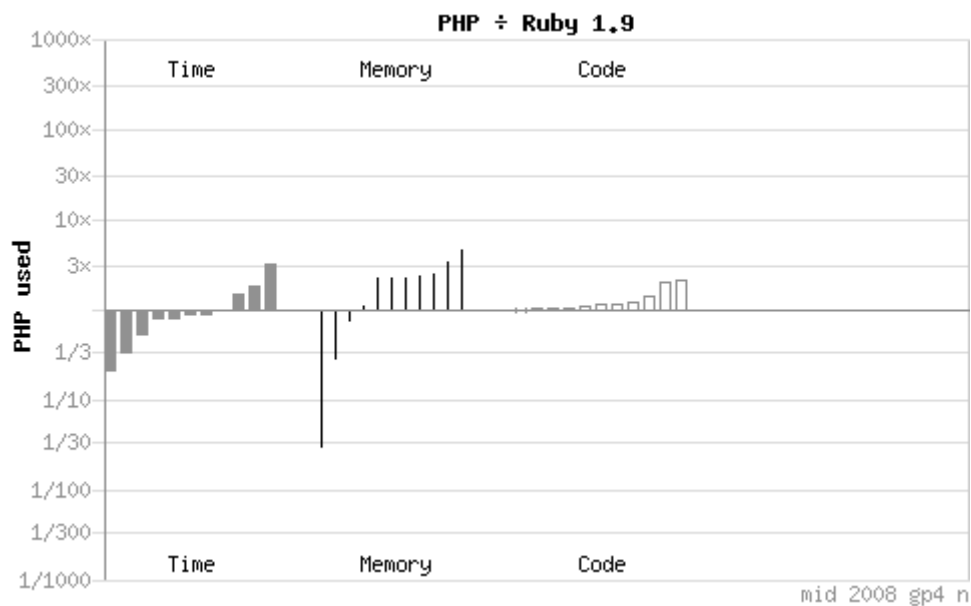


## 6. Σύγκριση Scripting Γλωσσών Προγραμματισμού

### 6.1 Συγκριτικά Διαγράμματα

Το παρακάτω διάγραμμα εμφανίζει τριών ειδών συγκρίσεις ταχύτητας και μεγέθους σε επίπεδο μνήμης, χρόνου και κώδικα για την έκδοση 1.9 της Ruby και την PHP<sup>[20]</sup>.

Κάθε ράβδος αντικατοπτρίζει έναν X αλγόριθμο. Η οριζόντια γραμμή παριστάνει τον χρόνο του ταχύτερου προγράμματος σε PHP και η θέση της ράβδου την εν συγκρίσει τιμή του αντίστοιχου προγράμματος σε Ruby.



Εικόνα 6.1

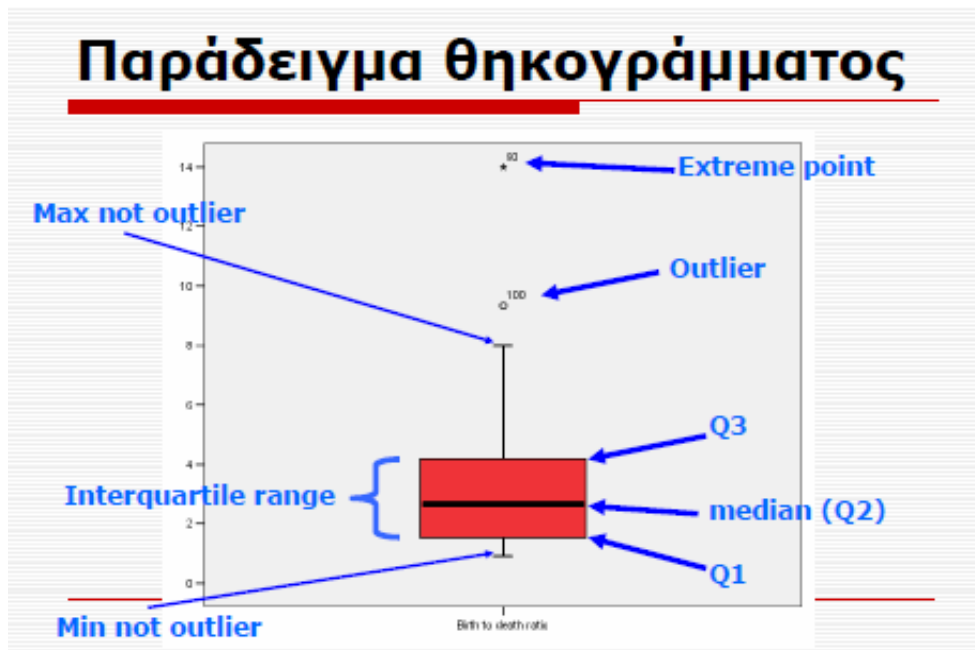
Για παράδειγμα, βλέπουμε από τη πρώτη ράβδο η οποία αντικατοπτρίζει έναν X αλγόριθμο ότι απαιτείται περίπου το 1/4 του χρόνου σε Ruby σε σχέση με τον χρόνο που απαιτείται σε PHP, το 1/30 σε μνήμη σε σχέση με τη PHP και περίπου την ίδια ποσότητα γραμμών κώδικα. Αντίστοιχα ερμηνεύονται και οι υπόλοιποι ράβδοι

#### Διερευνητική ανάλυση δεδομένων - Θηκόγραμμα(box-plot)

Γραφική παράσταση που παριστάνει την κατανομή των δεδομένων και συγκεκριμένα:

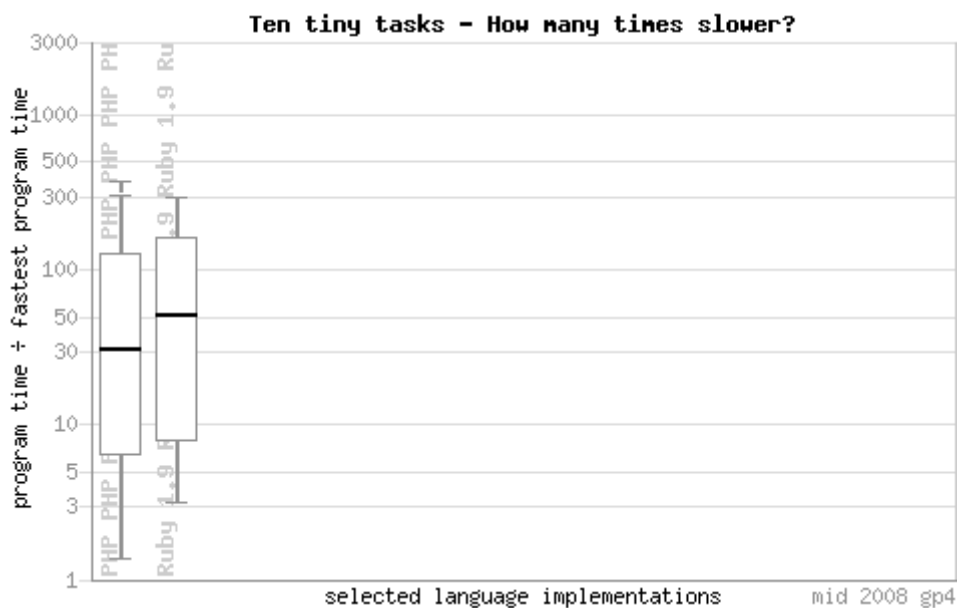
- Διάμεσο
- Τεταρτημόρια και ενδοτεταρτημοριακό εύρος
- Μέγιστη και ελάχιστη τιμή που δεν είναι στατιστικά ακραίες
- Τις ακραίες τιμές(outliers and extreme points)





Εικόνα 6.2

Το παρακάτω θηκόγραμμα παρουσιάζει συγκρίσεις της PHP και της Ruby ως προς τον χρόνο εκτέλεσης (CPU secs) στην υλοποίηση 10 διαφορετικών αλγορίθμων.



Εικόνα 6.3

## 6.2 Συγκριτικοί Πίνακες

Παρακάτω ακολουθεί πίνακας που δείχνει για κάθε ξεχωριστό αλγόριθμο, πόσο χρόνος απαιτείται από την Ruby σε σχέση με τη PHP. Δηλαδή για τα binary trees βλέπει κανείς πως με τη Ruby απαιτείται το 1/3 του χρόνου εκτέλεσης που απαιτείται με την PHP και το 1/5 της μνήμης αντίστοιχα.

Benchmark	Time	Memory	Code
binary-trees	$\frac{1}{3}$	$\frac{1}{5}$	±
recursive	$\frac{1}{2}$	$\frac{1}{2}$	±
fasta	$\frac{1}{2}$	34×	±
k-nucleotide	±	±	$\frac{1}{2}$
regex-dna	±	$\frac{1}{3}$	$\frac{1}{2}$
n-body	±	$\frac{1}{2}$	±
nsieve	±	4×	±
spectral-norm	±	$\frac{1}{2}$	±
partial-sums	2×	$\frac{1}{2}$	±
mandelbrot	3×	$\frac{1}{2}$	±
pidigits	5×	±	±

± Περίπου σε ίσα επίπεδα

### Αναλυτικός Συγκριτικός Πίνακας

Program Source Code	CPU secs	Elapsed secs	Memory KB	Code B	≈ CPU Load
<a href="#">pidigits</a>					
<a href="#">PHP</a>	2.24		7,312	537	
<a href="#">Ruby 1.9</a>	10.42		6,86	518	
<a href="#">mandelbrot</a>					
<a href="#">PHP</a>	245.48		5,484	395	
<a href="#">Ruby 1.9</a>	753.97		2,424	349	
<a href="#">partial-sums</a>					
<a href="#">PHP</a>	17.81		5,504	367	
<a href="#">Ruby 1.9</a>	33.82		2,448	362	
<a href="#">spectral-norm</a>					
<a href="#">PHP</a>	2,946.15		7,188	315	
<a href="#">Ruby 1.9</a>	3,756.99		3,284	292	

Σύγκριση scripting γλωσσών προγραμματισμού  
και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

<a href="#">_nsieve</a>					
<a href="#">PHP</a>	13.31		10,516	221	
<a href="#">Ruby 1.9</a>	16.63		37,428	227	
<a href="#">_n-body</a>					
<a href="#">PHP</a>	2,381.86		5,636	1289	
<a href="#">Ruby 1.9</a>	2,717.75		2,468	1137	
<a href="#">_regex-dna</a>					
<a href="#">PHP</a>	9.20		106,752	675	
<a href="#">Ruby 1.9</a>	10.43		31,844	323	
<a href="#">_k-nucleotide</a>					
<a href="#">PHP</a>	79.78		30,336	818	
<a href="#">Ruby 1.9</a>	82.07		39,628	420	
<a href="#">_fasta</a>					
<a href="#">PHP</a>	1,755.91		5,576	1029	
<a href="#">Ruby 1.9</a>	1,167.73		189,448	744	
<a href="#">_recursive</a>					
<a href="#">PHP</a>	25.44		6,316	315	
<a href="#">Ruby 1.9</a>	14.21		2,532	301	
<a href="#">_binary-trees</a>					
<a href="#">PHP</a>	173.25		81,648	493	
<a href="#">Ruby 1.9</a>	54.41		17,864	409	
<a href="#">_nsieve-bits</a>					
<a href="#">PHP</a>	26.77		17,128	317	
<a href="#">Ruby 1.9</a>	Failed				
<a href="#">_reverse-complement</a>					
<a href="#">PHP</a>	0.10		5,112	508	
<a href="#">Ruby 1.9</a>	Failed				

Στον παρακάτω πίνακα<sup>[21]</sup>, κάθε γραμμή παρουσιάζει μετρήσεις για κάθε ξεχωριστό πρόγραμμα γραμμένο σε γλώσσα PHP και Ruby για N επαναλήψεις.

Program Source Code	N	CPU secs PHP	CPU secs RUBY	Mem. KB PHP	Mem. KB RUBY	Code Line PHP	Code Line RUBY
binary-trees #2	12	7.27	2.47	10,34	3,752	493	409
binary-trees #2	14	35.99	10.97	24,6	7,844	493	409
binary-trees #2	16	173.25	54.41	81,648	17,864	493	409
fannkuch (deprecated)	9	7.62	Failed	5,5		484	334
fannkuch (deprecated) #2	9	99.46	8.43	5,5	2,7	484	334
fannkuch (deprecated) #2	10	1,165.02	102.30	5,5	2,728	484	334
fannkuch (deprecated) #2	11	17.66	1,301.14	5,576	2,728	1029	334
fasta	250,000	175.45	13.35	5,576	3,48	1029	744
fasta	2,500,000	1,755.91	123.99	5,576	19,844	1029	744
fasta	25,000,000	0.08	1,167.73	6,28	189,448	68	744

Σύγκριση scripting γλωσσών προγραμματισμού  
και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

Program Source Code	N	CPU secs PHP	CPU secs RUBY	Mem. KB PHP	Mem. KB RUBY	Code Line PHP	Code Line RUBY
hello-world	1	13.68	0.06	6,576	3,504	68	44
hello-world	200	0.94	3.44	16,66	3,516	818	44
k-nucleotide #2	10,000	8.35	1.02	25,948	8,64	818	420
k-nucleotide #2	100,000	79.78	8.61	30,336	26,916	818	420
k-nucleotide #2	1,000,000	0.48	82.07	5,488	39,628	395	420
mandelbrot #3	120	9.43	1.28	5,488	2,424	395	349
mandelbrot #3	600	245.48	31.09	5,484	2,424	395	349
mandelbrot #3	3,000	Failed	753.97		2,424	2582	349
meteor-contest	2,098	23.36	43.82	5,636	3,164	1289	2768
meteor-contest #2	2,098	240.70	34.82	5,64	3,068	1289	2786
n-body #2	200,000	2,381.86	27.27	5,636	3,236	1289	1137
n-body #2	2,000,000	3.32	273.22	6,764	2,468	221	1137
n-body #2	20,000,000	6.62	2,717.75	8,016	2,468	221	1137
nsieve	7	13.31	Failed	10,516		221	270
nsieve #2	7	26.77	4.06	17,128	11,176	317	227
nsieve #2	8	55.52	8.26	28,776	19,932	317	227
nsieve #2	9	115.15	16.63	52,08	37,428	317	227
nsieve-bits #2	9	0.26	Failed	5,504		367	346
partial-sums	25,000	1.89	0.39	5,504	2,448	367	362
partial-sums	250,000	17.81	3.42	5,504	2,448	367	362
partial-sums	2,500,000	2.63	33.82	5,84	2,448	736	362
pidigits	500	26.49	0.45	6,048	6,588	736	518
pidigits	1,500	82.16	3.65	6,308	6,708	736	518
pidigits	2,500	0.20	10.42	1,82	6,86	537	518
recursive #2	3	0.89	2.06	6,704	2,424	537	301
recursive #2	7	2.24	14.21	7,312	2,532	537	301
recursive #2	11	3.52	Failed	5,488		315	301
regex-dna	100,000	25.44	2.24	6,316	9,144	315	323
regex-dna	300,000	879.94	6.43	18,088	19,248	315	323
regex-dna	500,000	2.01	10.43	91,452	31,844	675	323
reverse-complement	25,000	5.79	Failed	99,84		675	268
spectral-norm	500	9.20	31.22	106,752	2,456	675	292
spectral-norm	3,000	0.10	1,191.00	5,112	2,716	508	292
spectral-norm	5,500	0.34	3,756.99	6,984	3,284	508	292
sum-file	1,000	2.58	1.30	39,696	2,388	508	75
sum-file	11,000	23.41	13.79	5,64	2,388	315	75
sum-file	21,000	879.00	26.31	6,48	2,388	315	75
sum-file #2	1,000	2,946.15	1.36	7,188	2,388	315	65
sum-file #2	11,000	2.10	14.51	5,46	2,388	146	65
sum-file #2	21,000	21.85	27.83	5,46	2,4	146	65
thread-ring	100,000	41.99	Failed	5,456		146	342
thread-ring #2	100,000		Failed				228

## 6.3 Συγκριτικοί Αλγόριθμοι

### 6.3.1 Πρόγραμμα “Hallo World”

#### 6.3.1.1 Πρόγραμμα “Hallo World” σε PHP

Κάθε γραμμή πίνακα εμφανίζει μετρήσεις για την ανάπτυξη του προγράμματος σε γλώσσα **PHP**, για N επαναλήψεις.

<u>N</u>	<u>CPU secs</u>	<u>Elapsed secs</u>	<u>Memory KB</u>	<u>Code B ≈ CPU Load</u>
1	0.08		6,280	68
200	13.68		6,576	68

#### Πηγαίος Κώδικας :

```
#!/usr/bin/php
<?php
print "hello world\n";
?>
```

#### make, command-line, and program output logs

```
RUNNING SCRIPT: hello.php
```

```
=====
COMMAND LINE (%A is single numeric argument):
repeat %A /usr/bin/php ../hello.php
```

#### 6.3.1.2 Πρόγραμμα “Hello World” σε Ruby

Κάθε γραμμή πίνακα εμφανίζει μετρήσεις για την ανάπτυξη του προγράμματος σε γλώσσα **Ruby**, για N επαναλήψεις.

<u>N</u>	<u>CPU secs</u>	<u>Elapsed secs</u>	<u>Memory KB</u>	<u>Code B ≈ CPU Load</u>
1	0.06		3,504	44
200	3.44		3,516	44

#### Πηγαίος Κώδικας :

```
#!/usr/bin/ruby
# -*- mode: ruby -*-
# $Id: hello.1.yarv.code,v 1.1 2008-08-06 18:34:23 igouy-guest Exp $

puts "hello world"
```

#### make, command-line, and program output logs

```
RUNNING SCRIPT: hello.yarv
```

```
=====
COMMAND LINE (%A is single numeric argument):
repeat %A /opt/ruby/ruby ../hello.yarv
```

## 6.3.2 Αλγόριθμος “Binnary Trees”

### 6.3.2.1 Αλγόριθμος “Binnary Trees” σε PHP

Κάθε γραμμή πίνακα εμφανίζει μετρήσεις για την ανάπτυξη του προγράμματος σε γλώσσα **PHP**, για N επαναλήψεις.

<u>N</u>	<u>CPU secs</u>	<u>Elapsed secs</u>	<u>Memory KB</u>	<u>Code B</u>	<u>≈ CPU Load</u>
12	7.27		10,340	493	
14	35.99		24,600	493	
16	173.25		81,648	493	

#### Πηγαίος Κώδικας :

```
<?php
```

```
function bottomUpTree($item, $depth)
{
    if($depth)
    {
        --$depth;
        $newItem = $item<<1;
        return array(
            bottomUpTree($newItem - 1, $depth),
            bottomUpTree($newItem, $depth),
            $item
        );
    }
    return array(NULL, NULL, $item);
}
```

```
function itemCheck($treeNode)
{
    $check = 0;
    do
    {
        $check += $treeNode[2];
        if(NULL == $treeNode[0])
        {
            return $check;
        }
        $check -= itemCheck($treeNode[1]);
        $treeNode = $treeNode[0];
    }
    while(TRUE);
}
```

```
$minDepth = 4;
$n = ($argc == 2) ? $argv[1] : 1;
$maxDepth = max($minDepth + 2, $n);
```

## Σύγκριση scripting γλωσσών προγραμματισμού και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

---

```
$stretchDepth = $maxDepth + 1;

$stretchTree = bottomUpTree(0, $stretchDepth);
printf("stretch tree of depth %d\t check: %d\n",
$stretchDepth, itemCheck($stretchTree));
unset($stretchTree);

$longLivedTree = bottomUpTree(0, $maxDepth);

$iterations = 1 << ($maxDepth);
do
{
    $check = 0;
    for($i = 1; $i <= $iterations; ++$i)
    {
        $t = bottomUpTree($i, $minDepth);
        $check += itemCheck($t);
        unset($t);
        $t = bottomUpTree(-$i, $minDepth);
        $check += itemCheck($t);
        unset($t);
    }

    printf("%d\t trees of depth %d\t check: %d\n",
$iterations<<1, $minDepth, $check);

    $minDepth += 2;
    $iterations >>= 2;
}
while($minDepth <= $maxDepth);

printf("long lived tree of depth %d\t check: %d\n",
$maxDepth, itemCheck($longLivedTree));
?>
```

### [make, command-line, and program output logs](#)

RUNNING SCRIPT: binarytrees.php-2.php

=====

COMMAND LINE (%A is single numeric argument):

/usr/bin/php -n -d memory\_limit=128M ../binarytrees.php-2.php %A

PROGRAM OUTPUT

=====

stretch tree of depth 17	check: -1
131072 trees of depth 4	check: -131072
32768 trees of depth 6	check: -32768
8192 trees of depth 8	check: -8192
2048 trees of depth 10	check: -2048
512 trees of depth 12	check: -512
128 trees of depth 14	check: -128
32 trees of depth 16	check: -32
long lived tree of depth 16	check: -1

### 6.3.2.2 Αλγόριθμος “Binnary Trees” σε Ruby

Κάθε γραμμή πίνακα εμφανίζει μετρήσεις για την ανάπτυξη του προγράμματος σε γλώσσα **Ruby**, για N επαναλήψεις.

#### N CPU secs Elapsed secs Memory KB Code B ≈ CPU Load

12	2.47	3,752	409
14	10.97	7,844	409
16	54.41	17,864	409

#### Πηγαίος Κώδικας :

```
def item_check(tree)
  if tree[0] == nil
    tree[1]
  else
    tree[1] + item_check(tree[0]) - item_check(tree[2])
  end
end

def bottom_up_tree(item, depth)
  if depth > 0
    item_item = 2 * item
    depth -= 1
    [bottom_up_tree(item_item - 1, depth), item, bottom_up_tree(item_item, depth)]
  else
    [nil, item, nil]
  end
end

max_depth = ARGV[0].to_i
min_depth = 4

max_depth = min_depth + 2 if min_depth + 2 > max_depth

stretch_depth = max_depth + 1
stretch_tree = bottom_up_tree(0, stretch_depth)

puts "stretch tree of depth #{stretch_depth}\t check: #{item_check(stretch_tree)}"
stretch_tree = nil

long_lived_tree = bottom_up_tree(0, max_depth)

min_depth.step(max_depth + 1, 2) do |depth|
  iterations = 2**(max_depth - depth + min_depth)

  check = 0

  for i in 1..iterations
    temp_tree = bottom_up_tree(i, depth)
    check += item_check(temp_tree)
  end

  temp_tree = bottom_up_tree(-i, depth)
end
```



## Σύγκριση scripting γλωσσών προγραμματισμού και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

---

```
check += item_check(temp_tree)
end

puts "#{iterations * 2}\t trees of depth #{depth}\t check: #{check}"
end

puts "long lived tree of depth #{max_depth}\t check: #{item_check(long_lived_tree)}"
```

### [make, command-line, and program output logs](#)

RUNNING SCRIPT: binarytrees.yarv-2.yarv

=====

COMMAND LINE (%A is single numeric argument):  
/opt/ruby/ruby ../binarytrees.yarv-2.yarv %A

PROGRAM OUTPUT

=====

stretch tree of depth 17	check: -1
131072 trees of depth 4	check: -131072
32768 trees of depth 6	check: -32768
8192 trees of depth 8	check: -8192
2048 trees of depth 10	check: -2048
512 trees of depth 12	check: -512
128 trees of depth 14	check: -128
32 trees of depth 16	check: -32
long lived tree of depth 16	check: -1

## 6.3.3 Αλγόριθμος “Recursive”

### 6.3.3.1 Αλγόριθμος “Recursive” σε PHP

. Κάθε γραμμή πίνακα εμφανίζει μετρήσεις για την ανάπτυξη του προγράμματος σε γλώσσα **PHP**, για N επαναλήψεις.

#### [N CPU secs Elapsed secs Memory KB Code B ≈ CPU Load](#)

3	3.52	5,488	315
7	25.44	6,316	315
11	879.94	18,088	315

#### [Πηγαίος Κώδικας :](#)

```
function ack($m, $n){
    if($m == 0) return $n+1;
    if($n == 0) return ack($m-1, 1);
    return ack($m - 1, ack($m, ($n - 1)));
}
```

```
function fib($n){
    if($n < 2) { return 1; }
    else { return fib($n - 2) + fib($n - 1); }
}
```

## Σύγκριση scripting γλωσσών προγραμματισμού και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

---

```
function tak($x,$y,$z){
  if($y < $x){ return tak( tak($x-1,$y,$z), tak($y-1,$z,$x), tak($z-1,$x,$y)); }
  else { return $z; }
}

$N = $argv[1];
echo "Ack(3,$N): ", ack(3,$N),"\\n";
printf("Fib(%.1f): %.1f\\n", 27.0+$N, fib(27.0+$N));
$N--; printf("Tak(%d,%d,%d): %d\\n", 3*$N,2*$N,$N, tak(3*$N,2*$N,$N));

printf("Fib(3): %d\\n", fib(3));
printf("Tak(3.0,2.0,1.0): %.1f\\n", tak(3.0,2.0,1.0));
?>
```

### [make, command-line, and program output logs](#)

```
RUNNING SCRIPT: recursive.php-2.php
=====
COMMAND LINE (%A is single numeric argument):
/usr/bin/php -n $PHPOPTS ../recursive.php-2.php %A

PROGRAM OUTPUT
=====
Ack(3,11): 16381
Fib(38.0): 63245986.0
Tak(30,20,10): 11
Fib(3): 3
Tak(3.0,2.0,1.0): 2.0
```

### 6.3.3.2 Αλγόριθμος “Recursive” σε Ruby

. Κάθε γραμμή πίνακα εμφανίζει μετρήσεις για την ανάπτυξη του προγράμματος σε γλώσσα **Ruby**, για N επαναλήψεις.

#### [N CPU secs Elapsed secs Memory KB Code B ≈ CPU Load](#)

3	2.06		2,424	301
7	14.21		2,532	301
11	Failed			301

#### [Πηγαίος Κώδικας :](#)

```
def ack(m, n)
  if m == 0 then
    n + 1
  else if n == 0 then
    ack(m - 1, 1)
  else
    ack(m - 1, ack(m, n - 1))
  end
end
end

# -----
```

## Σύγκριση scripting γλωσσών προγραμματισμού και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

---

```
def fib(n)
  if n > 1 then
    fib(n - 2) + fib(n - 1)
  else
    1
  end
end

# -----
def tak(x, y, z)
  if y < x then
    tak(tak(x - 1.0, y, z), tak(y - 1.0, z, x), tak(z - 1.0, x, y))
  else z
  end
end

# -----
n = (ARGV.shift || 1).to_i

printf("Ack(3,%d): %d\n", n, ack(3, n));
printf("Fib(%.1f): %.1f\n", 27.0 + n, fib(27.0 + n));

n -= 1;
printf("Tak(%d,%d,%d): %d\n", n * 3, n * 2, n, tak(n * 3, n * 2, n));

printf("Fib(3): %d\n", fib(3));
printf("Tak(3.0,2.0,1.0): %.1f\n", tak(3.0, 2.0, 1.0));
```

### [make, command-line, and program output logs](#)

RUNNING SCRIPT: recursive.yarv-2.yarv

```
=====
COMMAND LINE (%A is single numeric argument):
/opt/ruby/ruby ../recursive.yarv-2.yarv %A
../recursive.yarv-2.yarv:15:in `ack': stack level too deep (SystemStackError)
    from ../recursive.yarv-2.yarv:17:in `ack'
    from ../recursive.yarv-2.yarv:17:in `ack'
    from ../recursive.yarv-2.yarv:17:in `ack'
    from ../recursive.yarv-2.yarv:17:in `ack'
    from ../recursive.yarv-2.yarv:17:in `ack'
    from ../recursive.yarv-2.yarv:17:in `ack'
    from ../recursive.yarv-2.yarv:17:in `ack'
    from ../recursive.yarv-2.yarv:17:in `ack'
    from ../recursive.yarv-2.yarv:17:in `ack'
    ... 6225 levels...
    from ../recursive.yarv-2.yarv:17:in `ack'
    from ../recursive.yarv-2.yarv:17:in `ack'
    from ../recursive.yarv-2.yarv:17:in `ack'
    from ../recursive.yarv-2.yarv:45:in `<main>'
TEST FAILED - Child processed exited with error code (1)
```

PROGRAM OUTPUT

```
=====
```

### 6.3.4 Αλγόριθμος “FANNKUCH”

#### 6.3.4.1 Αλγόριθμος “Fannkuch” σε PHP

Κάθε γραμμή πίνακα εμφανίζει μετρήσεις για την ανάπτυξη του προγράμματος σε γλώσσα **PHP**, για N επαναλήψεις.

##### N CPU secs Elapsed secs Memory KB Code B ≈ CPU Load

9	7.62	5,500	484
10	99.46	5,500	484
11	1,165.02	5,500	484

##### Πηγαίος Κώδικας :

```
function Fannkuch($n){
    $check = 0;
    $perm = array();
    $perm1 = array();
    $count = array();
    $maxPerm = array();
    $maxFlipsCount = 0;
    $m = $n - 1;

    for ($i=0; $i<$n; $i++) $perm1[$i] = $i;
    $r = $n;

    while (TRUE) {
        // write-out the first 30 permutations
        if ($check < 30){
            for($i=0; $i<$n; $i++) echo $perm1[$i]+1;
            echo "\n";
            $check++;
        }

        while ($r != 1){ $count[$r-1] = $r; $r--; }
        if (! ($perm1[0]==0 || $perm1[$m] == $m)){
            for($i=0; $i<$n; $i++) $perm[$i] = $perm1[$i];
            $flipsCount = 0;

            while ( ! (($k=$perm[0]) == 0) ) {
                $k2 = ($k+1) >> 1;
                for($i=0; $i<$k2; $i++) {
                    $temp = $perm[$i]; $perm[$i] = $perm[$k-$i]; $perm[$k-$i] = $temp;
                }
                $flipsCount++;
            }

            if ($flipsCount > $maxFlipsCount) {
                $maxFlipsCount = $flipsCount;
                for($i=0; $i<$n; $i++) $maxPerm[$i] = $perm1[$i];
            }
        }
    }
}
```

## Σύγκριση scripting γλωσσών προγραμματισμού και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

---

```
while (TRUE) {
  if ($r == $n) return $maxFlipsCount;
  $perm0 = $perm1[0];
  $i = 0;
  while ($i < $r) {
    $j = $i + 1;
    $perm1[$i] = $perm1[$j];
    $i = $j;
  }
  $perm1[$r] = $perm0;

  $count[$r] = $count[$r] - 1;
  if ($count[$r] > 0) break;
  $r++;
}
}
}

$n = $argv[1];
printf("Pfannkuchen(%d) = %d\n", $n, Fannkuch($n));
?>
```

### [make, command-line, and program output logs](#)

RUNNING SCRIPT: fannkuch.php

=====

COMMAND LINE (%A is single numeric argument):

/usr/bin/php \$PHPOPTS ../fannkuch.php %A

PROGRAM OUTPUT

=====

```
1234567891011
2134567891011
2314567891011
3214567891011
3124567891011
1324567891011
2341567891011
3241567891011
3421567891011
4321567891011
4231567891011
2431567891011
3412567891011
4312567891011
4132567891011
1432567891011
1342567891011
3142567891011
4123567891011
1423567891011
1243567891011
2143567891011
2413567891011
4213567891011
2345167891011
```

```
3245167891011
3425167891011
4325167891011
4235167891011
2435167891011
Pfannkuchen(11) = 51
```

### 6.3.4.2 Αλγόριθμος “Fannkuch” σε Ruby

Κάθε γραμμή πίνακα εμφανίζει μετρήσεις για την ανάπτυξη του προγράμματος σε γλώσσα **Ruby**, για N επαναλήψεις.

N CPU secs Elapsed secs Memory KB Code B ≈ CPU Load

9	8.43		2,700	334
10	102.30		2,728	334
11	1,301.14		2,728	334

#### Πηγαίος Κώδικας :

```
def fannkuch(n)
  maxFlips, m, r, check = 0, n-1, n, 0
  count = (1..n).to_a
  perm = (1..n).to_a

  while true
    if check < 30
      puts "#{perm.join}"
      check += 1
    end

    while r != 1
      count[r-1] = r
      r -= 1
    end

    if perm[0] != 1 and perm[m] != n
      perml = perm.clone #.dup
      flips = 0
      while (k = perml.first) != 1
        perml = perml.slice!(0, k).reverse + perml
        flips += 1
      end
      maxFlips = flips if flips > maxFlips
    end
  end
  while true
    return maxFlips if r == n
    perm.insert r, perm.shift
    break if (count[r] -= 1) > 0
    r += 1
  end
end
end
```

Σύγκριση scripting γλωσσών προγραμματισμού  
και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

---

```
N = (ARGV[0] || 1).to_i
puts "Pfannkuchen(#{N}) = #{fannkuch(N)}"
```

[make, command-line, and program output logs](#)

RUNNING SCRIPT: fannkuch.yarv-2.yarv

Fri Jul 11 09:40:14 PDT 2008

=====  
COMMAND LINE (%A is single numeric argument):

/opt/ruby/ruby ../fannkuch.yarv-2.yarv %A

PROGRAM OUTPUT

=====

```
1234567891011
2134567891011
2314567891011
3214567891011
3124567891011
1324567891011
2341567891011
3241567891011
3421567891011
4321567891011
4231567891011
2431567891011
3412567891011
4312567891011
4132567891011
1432567891011
1342567891011
3142567891011
4123567891011
1423567891011
1243567891011
2143567891011
2413567891011
4213567891011
2345167891011
3245167891011
3425167891011
4325167891011
4235167891011
2435167891011
Pfannkuchen(11) = 51
```

## 7. Ανάλυση Κώδικα Βασικής Λειτουργικότητας σε PHP και Ruby

### 7.1 Σενάριο : Αυθεντικοποίηση Χρηστών

#### 7.1.1 Σενάριο : Αυθεντικοποίηση Χρηστών με PHP

Η αυθεντικοποίηση γίνεται στο βασικό script αρχείο των καθηγητών το teacher.php. Συγκεκριμένα, κατά την επίσκεψη στις επιλογές καθηγητή, εμφανίζεται το script teacher.php το οποίο αρχικά κάνει include τα αρχεία login.php το οποίο κάνει σύνδεση με τη MySQL και το select\_db.php το οποίο επιλέγει τη βάση δεδομένων που θα χρησιμοποιηθεί.

login.php

```
<?php
$conn = mysql_connect ("localhost","project_uomuser","1234");
if (!$conn)
{
    echo "Communication Error! Unable to connect to Data Base...";
    exit;
}
mysql_query("SET NAMES 'utf8'");
?>
```

select\_db.php

```
<?php
mysql_select_db("project_uom");
?>
```

Στην αρχή του script πραγματοποιείται έλεγχος για ενεργό session (σύνοδος λειτουργίας). Εφόσον δεν έχει γίνει ακόμη η αυθεντικοποίηση το session είναι κενό, οπότε μια βοηθητική μεταβλητή "pointer" έχει τη τιμή 0 και συνεπώς εμφανίζεται στην οθόνη η φόρμα εισαγωγής username και password.

Εφόσον εισαχθούν τα δύο αυτά στοιχεία και πατηθεί το πλήκτρο login, ξαναεκτελείται το script teacher.php στο οποίο αρχικά γίνεται έλεγχος των στοιχείων σε σύγκριση με τον πίνακα teacher της βάσης δεδομένων. Εφόσον επαληθευτεί ότι τα στοιχεία είναι υπαρκτά, η βοηθητική μεταβλητή "exists" παίρνει τη τιμή 1, πράγμα που σημαίνει ότι στη συνέχεια ενεργοποιείται το session και δημιουργείται ουσιαστικά η session μεταβλητή \$\_SESSION["valid\_user"]

```
session_register("valid_user");
```

Από το script teacher.php



```
<?php
$queryCHARSET = "SET CHARACTER_SET_RESULTS=utf-8";
$resultCHARSET = mysql_query($queryCHARSET);

    $uid = $_POST["userid"];
    $pwd = $_POST["pwd"];
    $lines="select * from teacher where t_login='$uid' and t_password='$pwd'";
$result=mysql_query($lines);
$number_of_rows=mysql_num_rows($results);
if ($number_of_rows == 0)
    $exist=0;
    else
        $exist=1;

/* Efosa ta stoixeia prosvasis einai sosta, dimiourgoume
mia synodo leitourgias (session) me to username tou xrhsth */
if($exist==1){
    $valid_user=$_POST["userid"];
    session_register("valid_user");
}

if (session_is_registered("valid_user"))
    $pointer=1;
else
    if (isset($_POST["userid"]))
        $pointer=0;
    else
        $pointer=0;
?>
```

Στη συνέχεια εφόσον έχει δημιουργηθεί το session και ο χρήστης έχει κάνει έγκυρα το Login του, η βοηθητική μεταβλητή pointer παίρνει τη τιμή 1 με αποτέλεσμα να εμφανίζεται πλέον το μενού επιλογών του καθηγητή (teachermenu.php) και όχι η φόρμα πρόσβασης (login form).

Από το script teacher.php

```
<?php
if ($pointer==1){
    if ($_GET["profil"]==1){
        include("profil.php");
    }
    else
        if ($_GET["users"]==1){
            include("users.php");
        }
    else
        if ($_GET["prlist"]==1){
            $back = $_GET["back"];
            include("prlist.php");
        }
}
```

```
}
else
  if ($_GET["projects"]==1){
    include("projects.php");
  }
  else
    include("teachermenu.php");
}
else
  include("loginmenu.php");
?>
```

Πάνω παρατηρούμε φωλεμένα if τα οποία αφορούν στις επιλογές του teachermenu.php. Αν επιλέξουμε για παράδειγμα την διαχείριση των χρηστών ουσιαστικά καλείται το ίδιο script με μια παράμετρο που καλεί κάποιο σχετικό script. Δηλαδή, καλείται το teacher.php?users=1  
Αυτό σημαίνει πως όταν φτάσει σε αυτά τα φωλεμένα if θα ισχύει η συνθήκη \$\_GET["users"]==1, οπότε και θα εμφανιστεί το περιεχόμενο του script users.php το οποίο κάνει τη διαχείριση χρηστών include("users.php");

#### Teachermenu.php

```
<?php
echo "<td align=\"center\"><img src=\"images/teacher_icon.jpg\" alt=\"\" width=\"240\"
height=\"240\" border=\"0\"></td><td align=\"center\">";
$sql="select * from teacher where t_login='$valid_user'";
$result=mysql_query($sql);
while($row=mysql_fetch_array($result)){
  $name=$row['t_name'];
  $isadmin=$row['is_admin'];
}
if($name=="")
  echo "<p class=\"headers\">Συνδεδεμένος Χρήστης : ".$valid_user."</p>";
else
  echo "<p class=\"headers\">Συνδεδεμένος Χρήστης : ".$name."</p>";

$lines1="select m_code from project where pointer=1 and t_login='$valid_user'";
$result1=mysql_query($lines1);
while($newArray=mysql_fetch_array($result1)){
  $m_code=$newArray['m_code'];
  $lines2="select * from inbox where m_code='$m_code' and m_readen=0 and
m_to=1";
  $result2=mysql_query($lines2);
  $number_of_rows2=mysql_num_rows($result2);
  if ($number_of_rows2 > 0)
    $new=1;
}

echo "<p class=\"mainheader\">Επιλογές Καθηγητή</p>";
echo "<p><a href=\"teacher.php?projects=1\" target=\"_self\">Οι Πτυχιακές
```

```
μου</a></p>";  
echo "<p><a href=\"teacher.php?prlist=1&back=1\" target=\"_self\">Λίστα  
Πτυχιακών</a></p>";  
echo "<p><a href=\"teacher.php?profil=1\" target=\"_self\">Διαχείριση Προφίλ</a></p>";  
if ($isAdmin==1)  
echo "<p><a href=\"teacher.php?users=1\" target=\"_self\">Διαχείριση  
Χρηστών</a></p><p>&nbsp;</p>";  
else  
echo "<p>&nbsp;</p>";  
echo "<p><a href=\"teacher.php?logout=1\" target=\"_self\">Αποσύνδεση</a></p>";  
echo "</td></tr>";  
echo "<tr><td colspan=\"2\"><br><p align=\"center\"><a href=\"index.html\"  
target=\"_self\">Επιστροφή</a></td></tr>";  
>
```

Κατά την έξοδο του (logout) ο χρήστης, πατά το πλήκτρο “αποσύνδεση” το οποίο καλεί πάλι τον εαυτό του με μια συνοδευτική βοηθητική μεταβλητή logout, δηλαδή το teacher.php?logout=1 και ουσιαστικά καταστρέφει το session επανεμφανίζοντας έτσι τη φόρμα πρόσβασης, όπως και στην αρχή.

```
/*Elegchos gia logout  
an i metavliti logout=1 tote katastrefetai to session  
kai kanei logout o xristis kai i selida epanerxete sto  
simeio pou zita stoixeia (username kai password) gia prosvasi.. */  
  
if($_GET["logout"]==1){  
    $result = session_unregister("valid_user");  
    session_destroy();  
}
```

Στην αρχή κάθε script είναι απαραίτητη η εντολή

```
session_start();
```

όπου κάνει έναρξη της συνόδου λειτουργίας.

### 7.1.2 Σενάριο : Αυθεντικοποίηση Χρηστών με Ruby

Η αυθεντικοποίηση έγινε με την βοήθεια του Authlogic plugin το οποίο ουσιαστικά φτιάχνει τις μεθόδους για την διαδικασία της αυθεντικοποίησης.

**Το μοντέλο** που θα επικοινωνεί με την βάση είναι το εξής:

```
class TeacherSession < Authlogic::Session::Base  
  
  def to_key  
  
    new_record? ? nil : [ self.send(self.class.primary_key) ]  
  
  end
```

```
end
```

Το μοντέλο το μόνο που κάνει είναι να ελέγχει αν υπάρχει το session. Δεν δημιουργείται κάποιο πεδίο ή κάποιος πίνακας.

Οι **μέθοδοι(controllers)** που χρησιμοποιήθηκαν είναι οι εξής:

```
class ApplicationController < ActionController::Base
  protect_from_forgery

  helper :all
  #Του λέμε ότι οι παρακάτω μέθοδοι current_teacher_session και current_teacher
  θα #χρησιμοποιούνται μέσα σε όποιο controller θέλουμε ώστε να ελέγχουμε αν
  υπάρχει ή όχι #ο καθηγητής.
  helper_method :current_teacher_session, :current_teacher
  filter_parameter_logging :password
  filter_parameter_logging :password, :password_confirmation

private

  #Authlogic Session Creation
  #Μέθοδος που βρίσκει αν υπάρχει το session του συγκεκριμένου καθηγητή
  def current_teacher_session
    logger.debug "ApplicationController::current_teacher_session"
    return @current_teacher_session if defined?(@current_teacher_session)
    @current_teacher_session = TeacherSession.find
  end

  #Μέθοδος που βρίσκει τον καθηγητή αν αυτός συνδεθεί
  def current_teacher
    logger.debug "ApplicationController::current_teacher"
    return @current_teacher if defined?(@current_teacher)
    @current_teacher = current_teacher_session &&
    current_teacher_session.teacher
  end
end
```

```
end

#Μέθοδος που ελέγχει αν έχει αν έχει συνδεθεί ο καθηγητής. Αν όχι εμφανίζει το
μήνυμα #«You must be logged in to access this page» και τότε τον επιστρέφει στο
#new_teacher_session_url που είναι ουσιαστικά το σημείο που εμφανίζει την
φόρμα της #σύνδεσης του καθηγητή.
def require_teacher
  logger.debug "ApplicationController::require_teacher"
  unless current_teacher
    store_location
    flash[:notice] = "You must be logged in to access this page"
    redirect_to new_teacher_session_url
    return false
  end
end

#Όταν δεν χρειάζεται να γνωρίζουμε αν είναι συνδεδεμένος ή όχι ένας καθηγητής.
def require_no_teacher
  logger.debug "ApplicationController::require_no_teacher"
  if current_teacher
    store_location
    flash[:notice] = "You must be logged out to access this page"
    redirect_to root_url
    return false
  end
end

#Μέθοδος που ελέγχει το σημείο του url. Δεν το χρειαζόμαστε.
def store_location
  session[:return_to] = request.request_uri
end
```

```
end

#Μέθοδος που επιστρέφει το url για το οποίο έγινε το αίτημα. Πχ αν καλέσουμε το
#http://www.railsgreece.gr/project/new χειροκίνητα αυτό θα μας πάει στο login
page και #μετά σε αυτή τη διεύθυνση. Δεν χρησιμοποιείται κάπου στο πρόγραμμα.
def redirect_back_or_default(default)
  redirect_to(session[:return_to] || default)
  session[:return_to] = nil
end
end
```

Οι παραπάνω μέθοδοι δημιουργούνται στο application controller έτσι ώστε να είναι προσβάσιμες από οποιοδήποτε controller κάτω από αυτό. Όλες οι άλλες οι κλάσεις κληρονομούν τις μεθόδους της ApplicationController όπως βλέπουμε και κάτω στην ProjectsController.

Πχ αν κάποιος πάει να μπει να δει τις πτυχιακές θα γίνει έλεγχος. Το before\_filter τρέχει πρώτα την μέθοδο require\_teacher πρώτου εκτελέσει οποιαδήποτε μέθοδος του ProjectsController.

```
class ProjectsController < ApplicationController
  #Άμα βγάλουμε το before_filter δεν θα πραγματοποιηθεί η διαδικασία της
  #αυθεντικοποίησης όταν κάποιος θα πάει να μπει να δει τις πτυχιακές. Η
  require_teacher #καλεί την μέθοδο που βρίσκεται στην ApplicationController.
  before_filter :require_teacher
  def index
  ....
  ....
```

Οι μέθοδοι που συνδέονται με τις προβολές δημιουργίας και καταστροφής του session είναι οι παρακάτω:

```
class TeacherSessionsController < ApplicationController
  layout "teachers"
  #Λέμε ποια μέθοδο από το application controller θα χρησιμοποιήσει σε κάθε μια
  από τις #παρακάτω μεθόδους
  before_filter :require_no_teacher, :only => [:new, :create]
  before_filter :require_teacher, :only => :destroy
```

```
def new
  @teacher_session = TeacherSession.new
end

#Πέρνει τα στοιχεία από την φόρμα εισόδου και δημιουργεί το session
def create
  @teacher_session = TeacherSession.new(params[:teacher_session])
  if @teacher_session.save
    redirect_to teacher_menu_url
  else
    render :action => :new
  end
end

def destroy
  current_teacher_session.destroy
  redirect_to new_teacher_session_url
end

end
```

**Η προβολή** δημιουργίας του session είναι:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Πτυχιακές Εργασίες - Πανεπιστήμιο Μακεδονίας</title>
<%= stylesheet_link_tag 'template' %>
</head>

<body>
<table width="800" border="0" align="center">
```

```
<tr>
  <td align="center"><%= image_tag ('/images/teacher_icon.jpg', :size=>"240x240",
:header=>"0")%> </td>
  <td>
    <p class="mainheader" align="center">Έλεγχος πρόσβασης</p>
    <br />
    <p class="headers" align="center">Παρακαλώ εισάγετε το username και το
password.</p>

    <% form_for @teacher_session, :url => {:action => "create"} do |f| %>
    <%= f.error_messages %>
    <table align="center" border="0">
      <table border="0" align="center">
        <tr><td class="maintext"><%= f.label "username :" %></td>
          <td><%= f.text_field :login %></td></tr>
        <tr><td class="maintext"><%= f.label "password :" %></td>
          <td><%= f.password_field :password %></td></tr>
        <tr><td colspan=2 align=right><%= f.submit "login" %></td></tr>
      </table>
    <% end %>

    <p></p>
    <p></p>
  </td>
</tr>
<tr>
  <td colspan="2" align="center">
    <p align="center" class="headers">
      Σε περίπτωση που εισάγετε σωστά το username και το password και δεν έχετε
πρόσβαση,
```



```
</br>παρακαλώ επικοινωνήστε με τον διαχειριστή του συστήματος κάνοντας  
κλικ  
  <a href="mailto:projects@uom.gr">εδω</a>  
</p>  
</td>  
</tr>  
</table>  
<table align="center" border="0">  
  <tr>  
    <td>  
      <br /><%= link_to "Επιστροφή", :root %>  
    </td>  
  </tr>  
</table>  
<%=# if !current_teacher %>  
  <%=#= link_to "Σύνδεση", login_path %>  
  <%=# else %>  
  <%=#Καλωσήλθες: <%= current_teacher.login %>  
  <%=#= link_to "Αποσύνδεση", logout_path %>  
  <%=# end %>  
</body>  
</html>
```

Για να μπορέσει να λειτουργήσει η αυθεντικοποίηση με το Authlogic πρέπει να δημιουργηθούν εξτρά πεδία μέσα στη βάση. Το **migration** είναι το παρακάτω:

```
class CreateTeachers < ActiveRecord::Migration  
  def self.up  
    create_table :teachers do |t|  
      t.string :login  
      t.string :name  
      t.boolean :is_admin  
    end  
  end  
end
```

```
#Εξτρά πεδία υποχρεωτικά (τα 3 παρακάτω)
```

```
  t.string :encrypted_password, :null => false
```

```
  t.string :password_salt, :null => false
```

```
  t.string :persistence_token, :null => false
```

```
#Εξτρά πεδία προαιρετικά (τα 2 παρακάτω)
```

```
  t.datetime :current_login_at
```

```
  t.datetime :last_login_at
```

```
  t.string :last_project_created
```

```
  t.timestamps
```

```
end
```

```
end
```

```
def self.down
```

```
  drop_table :teachers
```

```
end
```

```
end
```

## 7.2 Σενάριο : Δημιουργία/Επεξεργασία χρηστών

### 7.2.1 Σενάριο : Δημιουργία/Επεξεργασία χρηστών με PHP

Όπως αναφέρθηκε και στο σενάριο της αυθεντικοποίησης, εφόσον δημιουργηθεί το session και αποκτήσει πρόσβαση ο χρήστης, βλέπει το μενού επιλογών ως καθηγητής μέσα από το script teachemenu.php. Κάνοντας κλικ στην επιλογή “διαχείριση χρηστών” ουσιαστικά παραπέμπεται στο link `teacher.php?users=1`

και εφόσον το `$_GET['users'] = 1` εμφανίζεται το περιεχόμενο του script `users.php`

`users.php`

```
<?php
  echo "<td align=\"left\" colspan=\"2\">";
    $sql="select * from teacher where t_login!='$valid_user";
    $result = mysql_query($sql);
    echo "<table border=0>";
    echo " <tr><td colspan=6><p class=\"mainheader\">Δημιουργία Νέου
Χρήστη</p></td></tr>";
    if($_GET["newrec"]==1){
      $kodikos = $_GET["kodikos"];
```



Σύγκριση scripting γλωσσών προγραμματισμού  
και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

```
$nam = $_GET["nam"];
$chkadmin = $_GET["chkadmin"];
if($pword != " and $nam != "){
    $supsql2="update teacher set t_password='$pword', t_name='$nam',
is_admin='$chkadmin' where t_login='$kodikos";
    $supresult2=mysql_query($supsql2);
    echo "<tr><td colspan=6><p align=\"center\" class=\"maintext\">Η
τροποποίηση έγινε με επιτυχία.</p></td></tr>";
}
else
    echo "<tr><td colspan=6><p align=\"center\" class=\"maintext\">Σφάλμα!
Παρακαλώ επαναλάβετε συμπληρώνοντας όλα τα πεδία.</p></td></tr>";
}
echo "<tr><td width=\"10\"></td><td class=\"headers\">Username</td>";
echo "<td width=\"10\"></td><td class=\"headers\">Password</td>";
echo "<td width=\"10\"></td><td class=\"headers\">Όνοματεπώνυμο</td>";
echo "<td width=\"10\"></td><td class=\"headers\">Διαχειριστής</td>";
echo "<td width=\"20\"></td><td width=\"20\"></td></tr>";
while($row=mysql_fetch_array($result)){
    $login=$row['t_login'];
    $password=$row['t_password'];
    $name=$row['t_name'];
    $isadmin=$row['is_admin'];
    echo "<tr><td width=\"10\"></td><td class=\"maintext\">.$login.</td>";
    echo "<td width=\"20\"></td><td><input type=\"text\" Name=\"password\"
id=\"password_$.login\" value=\"\".$.password.\"\" SIZE=\"20\"></td>";
    echo "<td width=\"40\"></td><td><input type=\"text\" Name=\"name\"
id=\"name_$.login\" value=\"\".$.name.\"\" SIZE=\"40\"></td>";
    if($isadmin == 1)
        echo "<td width=\"10\"></td><td><input type=\"checkbox\"
name=\"checkadmin\" id=\"checkadmin_$.login\" value=\"\".$.isadmin.\"\" checked></td>";

    else
        echo "<td width=\"10\"></td><td><input type=\"checkbox\"
name=\"checkadmin\" id=\"checkadmin_$.login\" value=\"\".$.isadmin.\"\"></td>";
    echo "<script language=\"JavaScript\" type=\"text/javascript\">var aaa=\";
var aa_$.login.\" = \"$.login.\";</script>";
    echo "<td align=\"center\"><input type=\"image\"
src=\"buttons/modify.jpg\" onClick=\"changerecord(aa_$.login.);\"></td>";
    echo "<td align=\"center\"><input type=\"image\"
src=\"buttons/delete.jpg\" onClick=\"deleterecord(aa_$.login.);\"></td></tr>";
}
echo "</table>";
echo "</td></tr>";
echo "<tr><td colspan=2\"><br><p align=\"center\"><a href=\"teacher.php\"
target=\"_self\">Επιστροφή</a></td></tr>";
?>
```

Για την δημιουργία, τροποποίηση και διαγραφή των χρηστών αλλά και του προφίλ του ίδιου χρήστη, καλούνται procedures γραμμένες σε κώδικα javascript που βρίσκονται στο βασικό script "teacher.php".

#### Από το script teacher.php

```
<script language="JavaScript" type="text/javascript">
function newrecord(){
  oVDivCb = document.getElementById("crt4" ).value;
  if(oVDivCb.checked = "true")
    oVDiv4 = 1;
  else
    oVDiv4 = 0;
  var sure=window.confirm("Επιβεβαίωση δημιουργίας");
  if(sure==true){
    oVDiv1 = document.getElementById("crt1" ).value;
    oVDiv2 = document.getElementById("crt2" ).value;
    oVDiv3 = document.getElementById("crt3" ).value;

    window.location="teacher.php?users=1&newrec=1&kodikos="+oVDiv1+"&pwd="+oVDiv2
    +"&nam="+oVDiv3+"&chkadmin="+oVDiv4;
  }
  else
    window.location="teacher.php?users=1";
}
function deleterecord(aaa){
  var sure=window.confirm("Επιβεβαίωση διαγραφής");
  if(sure==true){
    window.location="teacher.php?users=1&delete=1&kodikos="+aaa;
  }
  else
    window.location="teacher.php?users=1";
}
function changerecord(aaa){
  oVDivCb = document.getElementById("checkadmin_"+aaa ).value;
  if(oVDivCb.checked = "true")
    oVDiv3 = 1;
  else
    oVDiv3 = 0;
  var sure=window.confirm("Επιβεβαίωση αλλαγών");
  if(sure==true){
    oVDiv1 = document.getElementById("password_"+aaa ).value;
    oVDiv2 = document.getElementById("name_"+aaa ).value;

    window.location="teacher.php?users=1&change=1&kodikos="+aaa+"&pwd="+oVDiv1+"&
    nam="+oVDiv2+"&chkadmin="+oVDiv3;
  }
  else
    window.location="teacher.php?users=1";
}
function changeprofil(a1){
  var sure=window.confirm("Επιβεβαίωση αλλαγών");
  if(sure==true){
    oVDiv1 = document.getElementById("password_"+a1 ).value;
```

```
oVDiv2 = document.getElementById("name_"+a1 ).value;

window.location="teacher.php?profil=1&change=1&kodikos="+a1+"&pword="+oVDiv1+"&nam="+oVDiv2;
}
else
  window.location="teacher.php?profil=1";
}
</script>
```

### 7.2.2 Σενάριο : Δημιουργία/Επεξεργασία χρηστών με Ruby

Για την δημιουργία των χρηστών, χρησιμοποιήθηκε το scaffold το οποίο ουσιαστικά φτιάχνει μόνο του τις μεθόδους και τις φόρμες προσθήκης, επεξεργασίας και διαγραφής. Παραμετροποιήθηκε κατάλληλα ώστε να παίζουν τα νέα πεδία πχ `crypted_password` κτλ.

Το μοντέλο που χρησιμοποιείται για τους χρήστες είναι το ίδιο όπως αναφέρθηκε στην αυθεντικοποίηση.

Οι μέθοδοι controllers είναι οι παρακάτω:

```
class TeachersController < ApplicationController

  #before_filter :require_teacher

  # GET /teachers

  # GET /teachers.xml

  def index

    @teachers = Teacher.find(:all, :conditions => ['login != ?', current_teacher.login])

#εμφανίζει όλους τους χριστες εκτος απο τον sindedemeno

    respond_to do |format|

      format.html # index.html.erb

      format.xml { render :xml => @teachers }

    end

  end

  # GET /teachers/1

  # GET /teachers/1.xml

  def show
```

```
@teacher = Teacher.find(params[:id])

respond_to do |format|
  format.html # show.html.erb
  format.xml { render :xml => @teacher }
end

end

# GET /teachers/new
# GET /teachers/new.xml
def new
  @teacher = Teacher.new

  respond_to do |format|
    format.html # new.html.erb
    format.xml { render :xml => @teacher }
  end
end

# GET /teachers/1/edit
def edit
  @teacher = Teacher.find(params[:id])
end

# POST /teachers
# POST /teachers.xml
def create
  @teacher = Teacher.new(params[:teacher])

  respond_to do |format|
```

```
if @teacher.save
  format.html { redirect_to(teachers_url, :notice => 'Ο χρήστης δημιουργήθηκε με
επιτυχία!') }
  format.xml { render :xml => @teacher, :status => :created, :location =>
@teacher }
else
  format.html { render :action => "new" }
  format.xml { render :xml => @teacher.errors, :status => :unprocessable_entity }
end
end
end

# PUT /teachers/1
# PUT /teachers/1.xml
def update
  @teacher = Teacher.find(params[:id])

  respond_to do |format|
    if @teacher.update_attributes(params[:teacher])
      format.html { redirect_to(:back, :notice => '? Ο χρήστης ενημερώθηκε με
επιτυχία!') }
      format.xml { head :ok }
    else
      format.html { render :action => "edit" }
      format.xml { render :xml => @teacher.errors, :status => :unprocessable_entity }
    end
  end
end

# DELETE /teachers/1
```



```
# DELETE /teachers/1.xml

def destroy

  @teacher = Teacher.find(params[:id])

  @teacher.destroy

  respond_to do |format|
    format.html { redirect_to(teachers_url) }
    format.xml { head :ok }
  end
end

def show_admin

  #@admin = Teacher.find(:first, :conditions=> {:t_login => ["admin"]})
  #@teacher = Teacher.find(:first)
  @teacher = current_teacher
  #@admin=123;
  #@teachers = Teacher.find(:all, :conditions => {:t_login => "admin"})
end
end
```

Οι προβολές που αντιστοιχούνε στις μεθόδους:

**H index** που εμφανίζει όλους τους χρήστες εκτός απ'τον συνδεδεμένο:

```
<td align="left" colspan="2">
  <table border=0>
    <tr>
      <td colspan=6><p class="mainheader">Διαχείριση χρηστών</p></td>
    </tr>
    <tr>
      <td width="10"></td>
      <td class="headers">Username</td>
      <td width="10"></td>
```

```
<td class="headers">Password</td>
<td width="10"></td>
<td class="headers">Όνοματεπώνυμο</td>
<td width="10"></td>
<td class="headers">Διαχειριστής</td>
<td width="20"></td>
<td width="20"></td>
</tr>

<% @teachers.each do |teacher| %>
<tr>
  <td width="10"></td>
  <td><%=h teacher.login %></td>
  <td width="10"></td>
  <td><%=h teacher.password %></td>
  <td width="10"></td>
  <td><%=h teacher.name %></td>
  <td width="10"></td>
  <td>
    <% if teacher.is_admin == false %>
      <%= "Όχι" %>
    <%else%>
      <%= "Ναί" %>
    <%end%>
  </td>
  <td width="20"></td>
  <td width="20"></td>
  <td>
    #Κουμπί για την επεξεργασία του χρήστη
```

```
<%= link_to(image_tag("/images/edit_icon.png",:size=>"20x20",:border=>"0"),
edit_teacher_path(teacher))%>
#Κουμπί για την διαγραφή του χρήστη
  <%= link_to(image_tag("/images/delete_icon.jpg",:size=>"20x20",:border=>"0"),
teacher, :confirm => 'Είστε σίγουρος?', :method=>:delete)%></td>
  <%= link_to 'Διαγραφή', :action=>"destroy", :confirm=>"Sigouros?" %>
</tr>
<% end %>
</table>

<p align="center"><%= link_to 'Νέος χρήστης', :new_teacher %></p>
<p align="center"><%= link_to 'Επιστροφή', :teacher_menu %></p>
</td>
```

Για την επεξεργασία **edit** των χρηστών η προβολή είναι η εξής:

```
<td align="left" colspan="2">
  <p class="mainheader">Επεξεργασία χρήστη</p>

  <%= render 'form' %>

  <%= form_for(@teacher) do |f| %>
  <% if @teacher.errors.any? %>
    <div id="error_explanation">
      <h2><%= pluralize(@teacher.errors.count, "error") %> prohibited this teacher
from being saved:</h2>

      <ul>
        <% @teacher.errors.full_messages.each do |msg| %>
          <li><%= msg %></li>
        <% end %>
      </ul>
    </div>
  </td>
```

```
<% end %>
<table>
<tr>
  <td width="10"></td>
  <td class="headers">Username</td>
  <td width="10"></td>
  <td class="headers">Password</td>
  <td width="10"></td>
  <td class="headers">Password</td>
  <td width="10"></td>
  <td class="headers">Όνοματεπώνυμο</td>
  <td width="10"></td>
  <td class="headers">Διαχειριστής</td>
  <td width="20"></td>
</tr>

<tr>
  <td width="10"></td>
  <td><%= @teacher.login %></td>
  <td width="10"></td>
  <td><%= f.text_field :password %></td>
  <td width="10"></td>
  <td><%= f.text_field :password_confirmation %></td>
  <td width="10"></td>
  <td><%= f.text_field :name %></td>
  <td width="10"></td>
  <td><%= f.check_box :is_admin %> </td>
  <td width="20"></td>
  <td width="20"><%= f.submit %></td>
</tr>
```

```
</table>
<% end %>

<p align="center"><%= link_to 'Επιστροφή', teachers_path %></p>
```

Ενώ για τη δημιουργία νέου χρήστη **new** η προβολή είναι η εξής:

```
<td align="left" colspan="2">
  <p class="mainheader">Προσθήκη νέου χρήστη</p>

  <%= form_for(@teacher) do |f| %>
    <% if @teacher.errors.any? %>
      <div id="error_explanation">
        <h2><%= pluralize(@teacher.errors.count, "error") %> prohibited this teacher
from being saved:</h2>

        <ul>
          <% @teacher.errors.full_messages.each do |msg| %>
            <li><%= msg %></li>
          <% end %>
        </ul>
      </div>
    <% end %>
  </td>
  <td align="left" colspan="2">
    <table>
      <tr>
        <td width="10"></td>
        <td class="headers">Username</td>
        <td width="10"></td>
        <td class="headers">Password</td>
        <td width="10"></td>
        <td class="headers">Επαλήθευση Password</td>
        <td width="10"></td>
```

```
<td class="headers">Όνοματεπώνυμο</td>
<td width="10"></td>
<td class="headers">Διαχειριστής</td>
<td width="20"></td>
</tr>

<tr>
<td width="10"></td>
<td><%=f.text_field :login %></td>
<td width="10"></td>
<td><%=f.text_field :password %></td>
<td width="10"></td>
<td><%=f.text_field :password_confirmation %></td>
<td width="10"></td>
<td><%=f.text_field :name %></td>
<td width="10"></td>
<td><%=f.check_box :is_admin %> </td>
<td width="20"></td>
<td width="20"><%= f.submit %></td>
</tr>
</table>
<% end %>
<p align="center"><%= link_to 'Επιστροφή', teachers_path %></p>
```

## 7.3 Σενάριο : Δημιουργία/Επεξεργασία πτυχιακών

### 7.3.1 Σενάριο : Δημιουργία/Επεξεργασία πτυχιακών με PHP

Κατά την είσοδο στη διαχείριση των προτεινόμενων πτυχιακών γίνεται αρχικά σύνδεση με τη βάση (login.php) και επιλογή της βάσης (select\_db.php) και στη συνέχεια με τη κατάλληλη mysql εντολή, εμφανίζονται όλες οι προτεινόμενες πτυχιακές του συνδεδεμένου καθηγητή.

```
$lines="select * from project where pointer=0 and t_login='$valid_user';"
```

Η μεταβλητή pointer όταν έχει τη τιμή 0 σημαίνει πως η πτυχιακή είναι προτεινόμενη, όταν έχει τη τιμή 1, τότε η πτυχιακή είναι σε εξέλιξη και όταν έχει τη τιμή 2 τότε είναι ολοκληρωμένη.

Οι επιλογές που έχει ο καθηγητής στο κομμάτι των προτεινόμενων πτυχιακών εξαρτώνται από τη τιμή της μεταβλητής number\_of\_rows η οποία περιέχει το πλήθος των προτεινόμενων πτυχιακών του συγκεκριμένου καθηγητή.

Αν το πλήθος αυτό είναι μηδέν, τότε όπως είναι λογικό η μόνη επιλογή που εμφανίζεται είναι η «δημιουργία νέας προτεινόμενης πτυχιακής». Ενώ αν προυπάρχουν προτεινόμενες, εμφανίζονται επιπρόσθετα και οι επιλογές για τροποποίηση, διαγραφή και ανάθεση.

Από το script suggested.php, το οποίο καλείται μέσω του projects.php

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="css/styles.css" rel="stylesheet">
<title>Πτυχιακές Εργασίες - Πανεπιστήμιο Μακεδονίας</title>
</head>
<?php

include("login.php");
include("select_db.php");

$queryCHARSET = "SET CHARACTER_SET_RESULTS=utf-8";
$resultCHARSET = mysql_query($queryCHARSET);

echo "<table><tr><td>&nbsp;</td>";
echo "<td class=\"headers\">Προτεινόμενες Πτυχιακές</td></tr></table>";

$lines="select * from project where pointer=0 and t_login='$valid_user';"
$result=mysql_query($lines);
$number_of_rows=mysql_num_rows($result);
if ($number_of_rows == 0){
    echo "<table><tr><td>&nbsp;</td>";
```

Σύγκριση scripting γλωσσών προγραμματισμού  
και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

```
        echo "<td class=\"smalltext\">Δεν υπάρχουν προτεινόμενες  
πτυχιακές.</td></tr></table>";  
    else{  
        echo "<table>";  
        while($newArray=mysql_fetch_array($result)){  
            $title=$newArray['project_title'];  
            $pointer=$newArray['pointer'];  
            echo "<tr><td>&nbsp;</td><td><a  
href=\"myprojects/each_project.php?title=$title.$pointer\">$title</a></td><tr>";  
        }  
        echo "</table>";  
    }  
    echo "<p align=\"center\" class=\"headers\">Επιλογές :</p>";  
    echo "<p align=\"center\"><a href=\"myprojects/add_project.php\">προσθήκη νέου  
θέματος</a></p>";  
    if ($number_of_rows > 0){  
        echo "<p align=\"center\"><a  
href=\"myprojects/change_project.php?situation=$pointer\">τροποποίηση  
θέματος</a></p>";  
        echo "<p align=\"center\"><a  
href=\"myprojects/delete_project.php?situation=$pointer\">διαγραφή θέματος</a></p>";  
        echo "<p align=\"center\"><a href=\"myprojects/offer_project.php\"  
target=\"_self\">ανάθεση πτυχιακής</a></p>";  
    }  
    echo "</td></tr><tr><td>&nbsp;</td></tr><tr><td><a href=\"teacher.php?projects=1\"  
target=\"_self\">Επιστροφή</a></td></tr></table>";  
?>  
</body>  
</html>
```

add\_project.php, για τη δημιουργία προτεινόμενης πτυχιακής

```
<?php  
    session_start();  
  
    if($valid_user==""){  
        echo "<h2>Sorry, You are not authorized to view this page!!</h2>";  
        exit;  
    }  
    include("header.htm");  
    echo "<table><tr><td>&nbsp;</td>";  
    echo "        <td class=\"mainheader\">Προτεινόμενες  
Πτυχιακές.</td></tr><tr><td>&nbsp;</td><td>";  
  
    echo "<p class=\"headers\">Προσθήκη νέου θέματος</p>";  
  
    include("create_code.php");  
  
    include("login.php");  
    include("select_db.php");  
    if($_GET['add'] == 1){
```



Σύγκριση scripting γλωσσών προγραμματισμού  
και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

```
$check = "select * from project where project_id='$code' or project_title='$title';  
//ELEGXOS MONADIKOTITAS KLEIDIWN TOY PINAKA  
$result0 = mysql_query($check);  
$number_of_rows2 = mysql_num_rows($result0);  
if($number_of_rows == 0)  
    if($title == "" || $nos == "" || $description == "" || $pre_lessons == "" || $category == "" ||  
$material == "")  
        echo "<p class=\"mainheader\">Τα στοιχεία δεν έχουν εισαχθεί με επιτυχία. Όσα πεδία  
έχουν * είναι υποχρεωτικά</p>";  
    else{  
        $query="insert                into                project  
values('$code','$title','$valid_user',0,$description','$nos','$pre_lessons','$category','$materi  
al',' ',' ',' ',' ',' ',' ');  
        $result=mysql_query($query);  
        include("creation_date.php");  
  
        $query2="update teacher set last_created='$code' where t_login='$valid_user';  
        $result2 = mysql_query($query2);  
  
        echo "<p class=\"mainheader\">Τα στοιχεία έχουν εισαχθεί με επιτυχία.</p>";  
    }  
else  
    echo "<p class=\"mainheader\">Τα στοιχεία δεν έχουν εισαχθεί με επιτυχία. Πιθανότατα  
να υπάρχει ήδη τίτλος με αυτό το όνομα. Παρακαλώ δώστε κάποιο άλλο όνομα.</p>";  
}  
$query3_1="select last_created from teacher where t_login='$valid_user';  
$result3_1 = mysql_query($query3_1);  
$number_of_rows=mysql_num_rows($result3_1);  
while($newArray=mysql_fetch_array($result3_1)){  
    $lc=$newArray['last_created'];  
}  
if($lc=="")  
    $code = "$p1"."_"."$p2"."$p3"."_"."001";  
else{  
    $counter=substr($lc, -3);  
    $cnt = $counter+1;  
    if($cnt > 1 && $cnt < 10)  
        $p4 = "00".$cnt;  
    else  
        if($cnt >= 10 && $cnt < 100)  
            $p4 = "0".$cnt;  
        else  
            $p4 = "00".$cnt;  
    $code = "$p1"."_"."$p2"."$p3"."_"."$p4";  
}  
  
echo "<p class=\"maintext\">Κωδικός : ";  
echo "$code<br></br>";  
echo "<form method=post action=\"add_project.php?code=$code&add=1\">";  
?>  
<p    class="maintext">*    Τίτλος    :<br><textarea    name="title"    Rows="4"
```

Σύγκριση scripting γλωσσών προγραμματισμού  
και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

```
Cols="70"></textarea></br></p>
<p class="maintext">(Παρακαλώ μην εισάγετε στον τίτλο το σύμβολο της τελείας και της
διπλής αποστρόφου)</font><br>
<br><p class="maintext">*Αριθμός Φοιτητών : <input type="text" Name="nos"
SIZE="20"></br>
<br><p class="maintext">*Περιγραφή :<br><textarea name="description" Rows="4"
Cols="70"></textarea></br>
<p class="maintext">*Προαπαιτούμενα Μαθήματα :<br><textarea name="pre_lessons"
Rows="4" Cols="70"></textarea></br>
<br><p class="maintext">*Κατηγορία : <br><textarea name="category" Rows="4"
Cols="70"></textarea></br>
<br><p class="maintext">*Βιβλιογραφία : <br><textarea name="material" Rows="4"
Cols="70"></textarea></br>
<p class="maintext"><b><i>Όσα πεδία έχουν </i></b>*<b><i>είναι υποχρεωτικά..</i></b>
<p><input Type=submit value="Καταχώρηση">
</form>
<?php
echo "</td></tr><tr><td>&nbsp;</td><td><a href=\"../teacher.php?projects=1&cat=1\"
target=\"_self\">Επιστροφή</a></td></tr>";
echo "</table>";
include("footer.htm");
?>
```

change\_project.php, για τη τροποποίηση προτεινόμενης πτυχιακής

```
<?php
    session_start();

    if($valid_user==""){
        echo "<h2>Sorry, You are not authorized to view this page!!</h2>";
        exit;
    }
    include("header.htm");
    echo "<table><tr><td>&nbsp;</td>";
    if ($_GET['situation']==1)
        echo " <td class=\"mainheader\">Πτυχιακές Σε
Εξέλιξη<br></td></tr><tr><td>&nbsp;</td><td>";
    else
        echo " <td class=\"mainheader\">Προτεινόμενες
Πτυχιακές<br></td></tr><tr><td>&nbsp;</td><td>";
    echo "<p class=\"headers\">Τροποποίηση θέματος</p>";

    include("login.php");
    include("select_db.php");
    if ($_GET['situation']==1)
        $lines="select * from project where pointer=1 and t_login='$valid_user'";
    else
        $lines="select * from project where pointer=0 and t_login='$valid_user'";
    $result=mysql_query($lines);
    $number_of_rows=mysql_num_rows($result);
    $title=$_POST['title'];
```



Σύγκριση scripting γλωσσών προγραμματισμού  
και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

```
if ($situation==1)
echo          "<td>&nbsp;</td></tr><tr><td          align=\"right\">&nbsp;</td><td><a
href=\"../teacher.php?projects=1&cat=2\" target=\"_self\">Επιστροφή</a></td></tr><tr>";
  else
echo          "<td>&nbsp;</td></tr><tr><td          align=\"right\">&nbsp;</td><td><a
href=\"../teacher.php?projects=1&cat=1\" target=\"_self\">Επιστροφή</a></td></tr><tr>";
  exit;
}

if(strstr($title, ".")){
  echo "<p class=\"mainheader\">Λυπούμαστε, δεν επιτρέπεται να μπαίνει τελεία στον
τίτλο. Παρακαλώ επαναλάβετε την εισαγωγή των στοιχείων.</p>";
  if ($situation==1)
echo          "<td>&nbsp;</td></tr><tr><td          align=\"right\">&nbsp;</td><td><a
href=\"../teacher.php?projects=1&cat=2\" target=\"_self\">Επιστροφή</a></td></tr><tr>";
  else
echo          "<td>&nbsp;</td></tr><tr><td          align=\"right\">&nbsp;</td><td><a
href=\"../teacher.php?projects=1&cat=1\" target=\"_self\">Επιστροφή</a></td></tr><tr>";
  exit;
}

mysql_select_db("projects");

$date1 = mktime("", "", "", $month1, $day1, $year1);
$date2 = mktime("", "", "", $month2, $day2, $year2);

if($month1=="" || $day1=="" || $year1=="")
  $p1=0;
else
  $p1=1;
if($month2=="" || $day2=="" || $year2=="")
  $p2=0;
else
  $p2=1;

if($p1==0 && $p2==0){
  $query="update          project          set
project_title='$title',description='$description',nos='$nos',pre_lessons='$pre_lessons',categ
ory='$category',material='$material'          where          t_login='$valid_user'          and
project_title='$part[0]";
  $query2="update student set s_names='$s_names', s_email='$s_email' where
s_code='$part[2]";
}
else
  if($p1==1 && $p2==1){
    $query="update          project          set
project_title='$title',description='$description',nos='$nos',pre_lessons='$pre_lessons',categ
ory='$category',material='$material', date_create = from_unixtime('$date1'), date_begin =
from_unixtime('$date2') where t_login='$valid_user' and project_title='$part[0]";
    $query2="update student set s_names='$s_names', s_email='$s_email'
```

```
where s_code='$part[2]';
    }
    else
        if($p1==0 && $p2==1){
            $query="update                project                set
project_title='$title',description='$description',nos='$nos',pre_lessons='$pre_lessons',category='$category',material='$material', date_begin = from_unixtime('$date2') where
t_login='$valid_user' and project_title='$part[0]";
            $query2="update student set s_names='$s_names',
s_email='$s_email' where s_code='$part[2]";
        }
        else
            if($p1==1 && $p2==0){
                $query="update                project                set
project_title='$title',description='$description',nos='$nos',pre_lessons='$pre_lessons',category='$category',material='$material', date_create = from_unixtime('$date1') where
t_login='$valid_user' and project_title='$part[0]";
                $query2="update student set s_names='$s_names',
s_email='$s_email' where s_code='$part[2]";
            }

$result=mysql_query($query);
$result2=mysql_query($query2);

if($result)
    echo "<p class=\"mainheader\">Η Μεταβολή έγινε με επιτυχία.</p>";
else
    echo "<p class=\"mainheader\">Η Μεταβολή δεν έγινε με επιτυχία.</p>";
}

// $part=explode(".",$_POST['title']);

$trans1=$_POST['titles'];
$trans2=$situation;
$sql="select
project_title,description,nos,pre_lessons,category,material,s_code,date_format(date_create
,'%d-%m-%Y') as dcf,date_format(date_begin,'%d-%m-%Y') as
dbf,date_format(date_end,'%d-%m-%Y') as def from project where project_title='$trans1'
and t_login='$valid_user";
$result=mysql_query($sql);
while($row=mysql_fetch_array($result)){
    $s_code=$row['s_code'];
    echo
        "<form                                method=post
action=\"change_project2.php?title2=$trans1.$trans2.$s_code&change=1\">";
    echo "<p class=\"maintext\">ΤΙΤΛΟΣ :<br><textarea name=\"title\" Rows=\"4\"
Cols=\"70\">";
    echo $row["project_title"];
    echo "</textarea></br>";
    echo "<p class=\"maintext\">(Παρακαλώ μην εισάγετε στον τίτλο σύμβολα όπως τελεία ή
διπλή απόστροφος.)<br>";
```

Σύγκριση scripting γλωσσών προγραμματισμού  
και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

```
$nos=$row["nos"];
echo "<p class=\"maintext\">ΑΡΙΘΜΟΣ ΦΟΙΤΗΤΩΝ : <input type=\"text\" Name=\"nos\"
SIZE=\"20\" value=\"$nos\"></br>";
echo "<p class=\"maintext\">ΠΕΡΙΓΡΑΦΗ :<br><textarea name=\"description\" Rows=\"4\"
Cols=\"70\">";
echo $row["description"];
echo "</textarea></br>";
echo "<p class=\"maintext\">ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ ΜΑΘΗΜΑΤΑ :<br><textarea
name=\"pre_lessons\" Rows=\"4\" Cols=\"70\">";
echo $row["pre_lessons"];
echo "</textarea></br>";
echo "<p class=\"maintext\">ΚΑΤΗΓΟΡΙΑ : <br><textarea name=\"category\" Rows=\"4\"
Cols=\"70\">";
echo $row["category"];
echo "</textarea></br>";
echo "<p class=\"maintext\">ΒΙΒΛΙΟΓΡΑΦΙΑ : <br><textarea name=\"material\"
Rows=\"4\" Cols=\"70\">";
echo $row["material"];
echo "</textarea></br>";
echo "<p class=\"maintext\">ΗΜΕΡΟΜΗΝΙΑ ΔΗΜΙΟΥΡΓΙΑΣ : <br>";
echo $row["dcf"];
echo "<p class=\"maintext\">Μεταβολή της ημερομηνίας δημιουργίας της πτυχιακής :";
echo "<p class=\"maintext\">Ημέρα / Μήνας / Έτος";
echo "<br><INPUT TYPE=text name=\"day1\" SIZE=\"1\" MAXLENGTH=\"2\"> / ";
echo "<INPUT TYPE=text name=\"month1\" SIZE=\"1\" MAXLENGTH=\"2\"> / ";
echo "<INPUT TYPE=text name=\"year1\" SIZE=\"1\" MAXLENGTH=\"4\"><br>";
echo "(π.χ. 05/11/2003)<br>";

if ($situation==1){
    include("return_student2.php");
    echo "<p class=\"maintext\">ΗΜΕΡΟΜΗΝΙΑ ΕΝΑΡΞΗΣ : <br>";
    echo $row["dbf"];
    echo "<p class=\"maintext\">Μεταβολή της ημερομηνίας έναρξης της πτυχιακής :";
    echo "<p class=\"maintext\">Ημέρα / Μήνας / Έτος";
    echo "<br><INPUT TYPE=text name=\"day2\" SIZE=\"1\" MAXLENGTH=\"2\"> / ";
    echo "<INPUT TYPE=text name=\"month2\" SIZE=\"1\" MAXLENGTH=\"2\"> / ";
    echo "<INPUT TYPE=text name=\"year2\" SIZE=\"1\" MAXLENGTH=\"4\"><br>";
    echo "(π.χ. 05/11/2003)<br>";
}
echo "<p><Input Type=submit value=\"Τροποποίηση\">";
}
mysql_free_result($result);

if ($_GET['situation']==1)
echo "</td></tr><tr><td>&nbsp;</td><td><a href=\"../teacher.php?projects=1&cat=2\"
target=\"_self\">Επιστροφή</a></td></tr>";
else
echo "</td></tr><tr><td>&nbsp;</td><td><a href=\"../teacher.php?projects=1&cat=1\"
target=\"_self\">Επιστροφή</a></td></tr>";
echo "</table>";
```

```
include("footer.htm");  
?>
```

delete\_project.php, για τη διαγραφή προτεινόμενης πτυχιακής

```
<?php  
    session_start();  
  
    if($valid_user==""){  
        echo "<h2>Sorry, You are not authorized to view this page!!</h2>";  
        exit;  
    }  
    include("header.htm");  
    echo "<table><tr><td>&nbsp;</td>";  
    if ($_GET['situation']==1)  
        echo "<td class=\"mainheader\">Πτυχιακές Σε  
Εξέλιξη<br></td></tr><tr><td>&nbsp;</td><td>";  
    else  
        echo "<td class=\"mainheader\">Προτεινόμενες  
Πτυχιακές<br></td></tr><tr><td>&nbsp;</td><td>";  
    echo "<p class=\"headers\">Διαγραφή θέματος</p>";  
  
    include("login.php");  
    include("select_db.php");  
  
    if($_GET['delete']==1)  
    {  
        $title=$_POST['titles'];  
        $query1="select m_code from project where project_title='$title';  
        //edo diagrafoume tixon  
        $result1=mysql_query($query1);  
        //minimata pou sxetizontai  
        while($newArray1=mysql_fetch_array($result1)){  
            //me tin sigkekrimeni ptyxiaki  
            $code=$newArray1['m_code'];  
        }  
        $query2="delete from inbox where m_code='$code';  
        $result2=mysql_query($query2);  
  
        $sql0="select s_code from project where project_title='$title'; // edo diagrafoume ti  
sisxetisi  
        $result0=mysql_query($sql0); // tis ptyxiakis me tous faitites  
        while($newArray0=mysql_fetch_array($result0)){ // pou tin anelavan (isxuei  
gia tis  
            $s_code=$newArray0['s_code']; // ptyxiakes se ekseliksi)  
            $sql7 = "delete from student where s_code='$s_code';  
            $result7=mysql_query($sql7);  
        }  
  
        $query="delete from project where project_title='$title';  
        $result=mysql_query($query);
```

```
if($result)
  echo "<p class=\"mainheader\">Η Διαγραφή έγινε με επιτυχία.</p>";
else
  echo "<p class=\"mainheader\">Η Διαγραφή δεν έγινε με επιτυχία.</p>";
}

if ($_GET['situation']==1)
  $lines="select * from project where pointer=1 and t_login='$valid_user'";
else
  $lines="select * from project where pointer=0 and t_login='$valid_user'";
$result=mysql_query($lines);
$number_of_rows=mysql_num_rows($result);
$title=$_POST['title'];
$situation=$_GET['situation'];
echo "<FORM ACTION=\"delete_project.php?situation=$situation&delete=1\"
METHOD=POST>";
echo "<SELECT NAME=\"titles\">";
while($newArray=mysql_fetch_array($result)){
  $title=$newArray['project_title'];
  echo "<OPTION VALUE=\"\$title\">$title</OPTION>";
}
echo "</SELECT>";
echo "<p><Input Type=submit value=\"Επιλογή\">";
echo "</FORM>";

if ($_GET['situation']==1)
echo "</td></tr><tr><td>&nbsp;</td><td><a href=\"../teacher.php?projects=1&cat=2\"
target=\"_self\">Επιστροφή</a></td></tr>";
else
echo "</td></tr><tr><td>&nbsp;</td><td><a href=\"../teacher.php?projects=1&cat=1\"
target=\"_self\">Επιστροφή</a></td></tr>";
echo "</table>";
include("footer.htm");
?>
```

offer\_project.php, για την ανάθεση προτεινόμενης πτυχιακής

```
<?php
  session_start();

  if($valid_user==""){
    echo "<h2>Sorry, You are not authorized to view this page!!</h2>";
    exit;
  }
include("header.htm");
echo "<table><tr><td>&nbsp;</td>";
echo "        <td class=\"mainheader\">Προτεινόμενες
Πτυχιακές<br></td></tr><tr><td>&nbsp;</td><td>";
echo "<p class=\"headers\">Ανάθεση Πτυχιακής</p>";
```



Σύγκριση scripting γλωσσών προγραμματισμού  
και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

```
include("login.php");
include("select_db.php");

$lines="select * from project where pointer=0 and t_login='$valid_user';
$result=mysql_query($lines);
$number_of_rows=mysql_num_rows($result);
$title=$_POST['title'];
echo "<FORM ACTION=\"offer_project2.php\" METHOD=POST>";
echo "<SELECT NAME=\"titles\">";
while($newArray=mysql_fetch_array($result)){
    $title=$newArray['project_title'];
    echo "<OPTION VALUE=\"\$title\">$title</OPTION>";
}
echo "</SELECT>";
echo "<p><Input Type=submit value=\"Επιλογή\">";
echo "</FORM>";

echo "</td></tr><tr><td>&nbsp;</td><td><a href=\"../teacher.php?projects=1&cat=1\"
target=\"_self\">Επιστροφή</a></td></tr>";
echo "</table>";
include("footer.htm");
?>
```

offer\_project2.php, καλείται από το προηγούμενο script για την ανάθεση προτεινόμενης πτυχιακής

```
<?
session_start();

if($valid_user==""){
    echo "<h2>Sorry, You are not authorized to view this page!!</h2>";
    exit;
}
include("header.htm");
echo "<table><tr><td>&nbsp;</td>";
echo "        <td class=\"mainheader\">Προτεινόμενες
Πτυχιακές<br></td></tr><tr><td>&nbsp;</td><td>";
echo "<p class=\"headers\">Τροποποίηση θέματος</p>";

include("login.php");
include("select_db.php");

if($_GET['offer']==1){
    $titles=$_GET['titles'];

    if(!$_POST['s_names'] || !$_POST['s_email']){
        echo "<p class=\"mainheader\"><b>Σφάλμα!!</b> Πρέπει να εισάγετε όνομα
σπουδαστή και email.";
        echo "</td></tr><tr><td>&nbsp;</td><td><a href=\"../teacher.php?projects=1&cat=1\"
target=\"_self\">Επιστροφή</a></td></tr>";
        echo "</table>";
    }
}
```

```
include("footer.htm");
    exit;
}

include("create_m_code.php");
include("create_s_code.php");
$sql="update project set s_code='$s_code', pointer=1, m_code='$code' where
project_title='$titles'";
$result=mysql_query($sql);
include("present_date.php");

if($result){
    $sql2="insert into student values('$s_code','$s_names','$s_email')";
    $result2=mysql_query($sql2);
    echo "<p class=\"mainheader\">Η καταχώρηση έγινε με επιτυχία.</p>";
    include("send_email.php");
}
else
    echo "<p class=\"mainheader\">Η καταχώρηση δεν έγινε με επιτυχία.</p>";
}
else{
    $trans1=$_POST['titles'];
    $sql="select
project_title,description,nos,pre_lessons,category,material,s_code,date_format(date_create
,'%d-%m-%Y') as dcf,date_format(date_begin,'%d-%m-%Y') as
dbf,date_format(date_end,'%d-%m-%Y') as def from project where project_title='$trans1'
and t_login='$valid_user'";
    $result=mysql_query($sql);
    while($row=mysql_fetch_array($result)){
        $s_code=$row['s_code'];
        echo "<p class=\"maintext\">ΤΙΤΛΟΣ :<br>";
        echo $row["project_title"];
        $nos=$row["nos"];
        echo "<p class=\"maintext\">ΑΡΙΘΜΟΣ ΦΟΙΤΗΤΩΝ : $nos<br>";
        echo "<p class=\"maintext\">ΠΕΡΙΓΡΑΦΗ :<br>";
        echo $row["description"];
        echo "</br>";
        echo "<p class=\"maintext\">ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ ΜΑΘΗΜΑΤΑ :<br>";
        echo $row["pre_lessons"];
        echo "</br>";
        echo "<p class=\"maintext\">ΚΑΤΗΓΟΡΙΑ : <br>";
        echo $row["category"];
        echo "</br>";
        echo "<p class=\"maintext\">ΒΙΒΛΙΟΓΡΑΦΙΑ : <br>";
        echo $row["material"];
        echo "</br>";
        echo "<p class=\"maintext\">ΗΜΕΡΟΜΗΝΙΑ ΔΗΜΙΟΥΡΓΙΑΣ : <br>";
        echo $row["dcf"];
        include("return_student2.php");
        echo "<p class=\"maintext\">ΗΜΕΡΟΜΗΝΙΑ ΕΝΑΡΞΗΣ : <br>";
```

```
echo $row["dbf"];
}
echo "<form method=post action=\"offer_project2.php?titles=$trans1&offer=1\">";
echo "<p class=\"maintext\">Παρακαλώ δώστε τα ονοματεπώνυμα των φοιτητών που
θ'αναλάβουν την πτυχιακή : ";
echo "<br><input type=\"text\" name=\"s_names\" size=\"90\"></br>";
echo "<p class=\"maintext\">*Για περισσότερα από ένα ονοματεπώνυμα, παρακαλώ
εισάγετε ανάμεσα παύλα!<br>π.χ. Πέτρου Βασίλειος-Λούπος Κωνσταντίνος.</br>";
echo "<p class=\"maintext\">Παρακαλώ εισάγετε το email του φοιτητή :";
echo "<br><input type=\"text\" name=\"s_email\" size=\"90\"></br>";
echo "<p class=\"maintext\">*Εισάγετε ΜΟΝΟ ένα email (ακόμα και στην περίπτωση που
περισσότεροι από ένας, αναλαμβάνουν την ίδια πτυχιακή).";
echo "<p><Input Type=submit value=\"Ανάθεση\"></form>";
}
echo "</td></tr><tr><td>&nbsp;</td><td><a href=\"../teacher.php?projects=1&cat=1\"
target=\"_self\">Επιστροφή</a></td></tr>";
echo "</table>";
include("footer.htm");
?>
```

### 7.3.2 Σενάριο : Δημιουργία/Επεξεργασία πτυχιακών με Ruby

Το μοντέλο που χρησιμοποιείται είναι το **Projects**. Μέσα σε αυτό ορίζονται οι συσχετίσεις με τους υπόλοιπους πίνακες. Ο Projects παίρνει ξένα κλειδιά του teacher και του student.

```
class Project < ActiveRecord::Base
  belongs_to :teacher # Μία πτυχιακή ανήκει σε έναν καθηγητή
  belongs_to :student #Μία πτυχιακή ανήκει σε κάποιο μαθητή
  ...
End
```

Ο controller των προτεινόμενων πτυχιακών είναι ο εξής:

```
class Project::SuggestedController < ApplicationController
  layout "projects"
  #Κάνει την αυθεντικοποίηση κατα πόσο ένας καθηγητής είναι συνδεδεμένος
  before_filter :require_teacher
  #Εμφανίζει όλες τις προτεινόμενες πτυχιακές του συνδεδεμένου καθηγητή
  def index
    @suggested = Project.find_all_by_status_and_teacher_id("suggested",
current_teacher.id)
```

```
end
#Καλεί την φόρμα για την δημιουργία μία νέας προτεινόμενης πτυχιακής
def new
  @project = Project.new
  create_project_code

  respond_to do |format|
    format.html # new.html.erb
    format.xml { render :xml => @project }
  end
end
#Δημιουργεί την νέα προτεινόμενη πτυχιακή
def create
  @project = Project.new(params[:project])
  respond_to do |format|
    if @project.save
      @current_teacher.update_attribute(:last_project_created, @project.code)
      format.html { redirect_to(project_suggested_index_url, :notice => 'Η πτυχιακή
δημιουργήθηκε με επιτυχία!') }
      format.xml { render :xml => @teacher, :status => :created, :location =>
@teacher }
    else
      format.html { redirect_to(:back, :error=>'Πρέπει να εισάγετε όλα τα απαραίτητα
πεδία')}}
#   format.html { render :action => "new" }
      format.xml { render :xml => @teacher.errors, :status => :unprocessable_entity }
    end
  end
end
end
```

```
#Εμφανίζει την φόρμα επεξεργασίας της πτυχιακής
def edit
  @project = Project.find(params[:id])
end
#Ενημερώνει την πτυχιακή
def update
  @project = Project.find(params[:id])

  respond_to do |format|
    if @project.update_attributes(params[:project])
      format.html { redirect_to(project_suggested_index_url, :notice => 'Η πτυχιακή
ενημερώθηκε με επιτυχία!') }
      format.xml { head :ok }
    else
      format.html { render :action => "edit" }
      format.xml { render :xml => @project.errors, :status => :unprocessable_entity }
    end
  end
end

#Διαγράφει την προτεινόμενη πτυχιακή
def destroy
  @project = Project.find(params[:id])
  if @project.destroy
    format.html { redirect_to(project_suggested_index_url, :notice => 'Η πτυχιακή
διαγράφηκε με επιτυχία!') }
    format.xml { head :ok }
  end
end
```

```
def recall
end

#Εμφανίζει την φόρμα ανάθεσης
def assign
  create_student_code #Δημιουργεί τον κωδικό της κάθε πτυχιακής
  @assigned = Project.find(params[:id])
  @student = Student.new
end

#Κάνει την ανάθεση της πτυχιακής
def offer
  @student = Student.new(params[:student])
  @project = @student.pr_code #Gives the number of the project
  @current_project = Project.find(@project)
  if @student.save
    #Updates the project attributes taken from the student
    @current_project.update_attributes(:student_id=>@student.id,
:status=>'inprocess')
    redirect_to(project_suggested_index_url, :notice => "Η ανάθεση
πραγματοποιήθηκε με επιτυχία!")
  else
    redirect_to :back, :error => "Προέκυψε ένα πρόβλημα, παρακαλώ δοκιμάστε
ξανά."
  end
end

private
#Δημιουργεί τον κωδικό της του μαθητή κατά την ανάθεση της πτυχιακής
def create_student_code
  now = DateTime.now
```

```
@year=now.year
@month=now.month
@day=now.day
@hour=now.hour
@mins=now.min
@secs=now.sec

@student_code = "#{@year}#{@month}#{@day}#{@hour}#{@mins}#{@secs}"
# "#{strftime("%Y/%m/%d %H:%M:%S")}"
end

def get_parts
  #Δημιουργεί το όνομα πτυχιακής : με την μορφή admin_X10-11_
  now = DateTime.now
  @this_year = now.year
  @next_year = @this_year+1
  @previous_year = @this_year-1
  @month = now.month

  #First part
  @username = @current_teacher.login

  #Second part
  @month=now.month
  if @month>=3 and @month<=8 then
    @part2="E"
  else
    @part2="X"
  end

  #Third part
```

```
@tY = @this_year.to_s[2..-1]
@pY = @previous_year.to_s[2..-1]
@nY = @next_year.to_s[2..-1]
if @month>=1 and @month<=8 then
  @part3 = @pY.to_s+"-"+@tY.to_s
else
  @part3 = @tY.to_s+"-"+@nY.to_s
end
end

def get_part4
#Δημιουργεί το τελευταίο κομμάτι του ονόματος της πτυχιακής
  @last = @current_teacher.last_project_created
  #@last_code = @last.
  if @last
    code = @last
    last=code[-3,3]
    @part4 = last.succ
  else
    @part4 = "001"
  end
end

def create_project_code
  get_parts
  get_part4
  @generated_code = @username+"_"+@part2+@part3+"_"+@part4
end

end
```



Οι εμφανίσεις είναι οι παρακάτω. Για την **δημιουργία της πτυχιακής:**

```
<td align="left" colspan="2">
  <p class="mainheader">Προτεινόμενες Πτυχιακές.</p>
  <p class="headers">Προσθήκη νέου θέματος</p>
  <br></br>
  <p class="maintext">Κωδικός : <%= @generated_code%><br></br></p>

  <% form_for (@project,:url=>project_suggested_index_path) do |f| %>
  <br><p class="maintext">* Τίτλος :<br><%= f.text_area :title, :size=>"70x4" %> </br>
  <p class="maintext">(Παρακαλώ μην εισάγετε στον τίτλο το σύμβολο της τελείας
  και της διπλής αποστρόφου)</font><br>
  <%= f.hidden_field :code, :value=>@generated_code%>
  <%= f.hidden_field :status, :value=>"suggested" %>
  <%= f.hidden_field :teacher_id, :value=>@current_teacher.id %>
  <br><p class="maintext">*Αριθμός Φοιτητών : <%= f.text_field :no_students,
  :size=>"20" %> </br>
  <br><p class="maintext">*Περιγραφή :<br><%= f.text_area :description,
  :size=>"70x4" %></br></br>
  <p class="maintext">*Προαπαιτούμενα Μαθήματα :<br><%= f.text_area
  :pre_lessons, :size=>"70x4" %></br>
  <br><p class="maintext">*Κατηγορία : <br><%= f.text_area :category, :size=>"70x4"
  %></br>
  <br><p class="maintext">*Βιβλιογραφία : <br><%= f.text_area :material,
  :size=>"70x4" %></br>
  <br><p><%=f.hidden_field :begin_at, :size=>"20", :value=>Time.now%>
  <p class="maintext"><b><i>Όσα πεδία έχουν </i></b>*<b><i>είναι
  υποχρεωτικά.</i></b>
  <br>
  <%= f.submit "Καταχώρηση" %>
  <% end %>
  <p><%= link_to "Επιστροφή", project_suggested_index_url %> </p>
```

```
</td>
```

Για την **επεξεργασία της πτυχιακής**:

```
<td align="left" colspan="2">
  <p class="mainheader">Προτεινόμενες Πτυχιακές.</p>
  <p class="headers">Τροποίηση θέματος</p>
  <br></br>

  <% form_for (@project,:url=>project_suggested_path) do |f| %>
  <br><p class="maintext">ΤΙΤΛΟΣ :<br><%= f.text_area :title, :size=>"70x4" %> </br>
  <p class="maintext">(Παρακαλώ μην εισάγετε στον τίτλο το σύμβολο της τελείας
  και της διπλής αποστρόφου)</font><br>
  <%= f.text_field :code, :value=>@generated_code%>
  <%= f.text_field :status, :value=>"suggested" %>
  <%= f.text_field :teacher_id, :value=>@current_teacher.id %>
  <br><p class="maintext">ΑΡΙΘΜΟΣ ΦΟΙΤΗΤΩΝ : <%= f.text_field :no_students,
  :size=>"20" %> </br>
  <br><p class="maintext">ΠΕΡΙΓΡΑΦΗ :<br><%= f.text_area :description,
  :size=>"70x4" %></br></br>
  <p class="maintext">ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ ΜΑΘΗΜΑΤΑ :<br><%= f.text_area
  :pre_lessons, :size=>"70x4" %></br>
  <br><p class="maintext">ΚΑΤΗΓΟΡΙΑ : <br><%= f.text_area :category, :size=>"70x4"
  %></br>
  <br><p class="maintext">ΒΙΒΛΙΟΓΡΑΦΙΑ : <br><%= f.text_area :material,
  :size=>"70x4" %></br>
  <br><p class="maintext">ΗΜΕΡΟΜΗΝΙΑ ΔΗΜΙΟΥΡΓΙΑΣ :
  <br><%= @project.created_at.strftime("%d-%m-%Y")%>
  <br><p class="maintext"> Μεταβολή της ημερομηνίας δημιουργίας της πτυχιακής
  :</br>
  <br><%= f.date_select("created_at", :order => [:day, :month, :year]) %> </br></p>
```

```
<%= f.submit "Τροποποίηση" %>
<% end %>
<p><%= link_to "Επιστροφή", project_suggested_index_url %> </p>

</td>
```

#### Η προβολή της ανάθεσης εργασίας:

```
<td align="left" colspan="2">
  <p class="mainheader">Προτεινόμενες Πτυχιακές.</p>
  <p class="headers">Διαγραφή θέματος</p>
  <br></br>
  <%# form_for @assigned do |f| %>
    <p class="maintext">ΤΙΤΛΟΣ :<br> <%= @assigned.title%> </br>
    <p class="maintext">ΑΡΙΘΜΟΣ ΦΟΙΤΗΤΩΝ : <br><%= @assigned.no_students%>
  </br>
    <p class="maintext">ΠΕΡΙΓΡΑΦΗ :<br><%= @assigned.description%> </br>
    <p class="maintext">ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ ΜΑΘΗΜΑΤΑ :<br><%=
  @assigned.pre_lessons%></br>
    <p class="maintext">ΚΑΤΗΓΟΡΙΑ : <br><%= @assigned.category%></br>
    <p class="maintext">ΒΙΒΛΙΟΓΡΑΦΙΑ : <br><%= @assigned.material%></br>
    <p class="maintext">ΗΜΕΡΟΜΗΝΙΑ ΔΗΜΙΟΥΡΓΙΑΣ : <br><%=
  @assigned.created_at%>
    <p class="maintext">ΗΜΕΡΟΜΗΝΙΑ ΕΝΑΡΞΗΣ : <br>
  <%# end %>

  <% form_for @student,:url=>offer_project_suggested_index_path do |f| %>
  <p class="maintext">Παρακαλώ δώστε τα ονοματεπώνυμα των φοιτητών που
  θ'αναλάβουν την πτυχιακή :
    <%= f.hidden_field :pr_code, :value=>@assigned.id %>
    <%= f.hidden_field :s_code, :value=>@student_code %>
    <br><%= f.text_field :names, :size=>"60"%> </br>
```

```
<p class="maintext">*Για περισσότερα από ένα ονοματεπώνυμα, παρακαλώ  
εισάγετε ανάμεσα παύλα!<br>π.χ. Πέτρου Βασίλειος-Λούπος Κωνσταντίνος.</br>  
<p class="maintext">Παρακαλώ εισάγετε το email του φοιτητή :  
<br><%= f.email_field :email, :size=>"20" %> </br>  
<p class="maintext">*Εισάγετε ΜΟΝΟ ένα email (ακόμα και στην περίπτωση που  
περισσότεροι από ένας, αναλαμβάνουν την ίδια πτυχιακή).  
<p><%= f.submit "Ανάθεση" %></p>  
<% end %>  
<p><%= link_to "Επιστροφή", project_suggested_index_url %> </p>  
</td>
```

## 8. Ανάπτυξη & Σύγκριση Σεναρίων Επέκτασης σε PHP και Ruby

Στο κεφάλαιο αυτό παρουσιάζονται διάφορα σενάρια επέκτασης του ήδη υπάρχοντος ηλεκτρονικού περιβάλλοντος. Στόχος του "πειράματος" αυτού είναι να διερευνηθούν οι δυνατότητες συντηρησιμότητας που παρέχονται από την κάθε γλώσσα καθώς με την υλοποίηση της ίδιας ακριβώς λειτουργικής απαίτησης στα δύο συστήματα και με μέτρηση της απαιτούμενης προσπάθειας (χρόνος και LOC) παρέχεται μια εκτίμηση του βαθμού δυσκολίας για την συντήρηση του λογισμικού ώστε να "φιλοξενήσει" την επιπρόσθετη λειτουργικότητα.

### 8.1 Σενάριο : Πλήθος Πτυχιακών Καθηγητή ανά κατηγορία και συνολικά

Προστίθεται η δυνατότητα στον καθηγητή όταν μπαίνει στις επιλογές του και επιλέγει «οι πτυχιακές μου» να βλέπει έναν πίνακα με τα συνολα των πτυχιακών του (βλέπε εικόνα 8.1).

ΚΑΤΗΓΟΡΙΑ	ΠΛΗΘΟΣ
Προτεινόμενες	1
Σε εξέλιξη	4
Ολοκληρωμένες	1
Συνολικά	6

Εικόνα 8.1

#### 8.1.1 Σενάριο : Πλήθος Πτυχιακών Καθηγητή ανά κατηγορία και συνολικά με PHP

Η προσθήκη αυτή τοποθετήθηκε στο κομμάτι των καθηγητών «Είσοδος Καθηγητή», στην επιλογή «οι πτυχιακές μου»

Διαδρομή : Είσοδος Καθηγητή -> οι πτυχιακές μου

Στο script **projects.php** προσθέτουμε στο σημείο που θέλουμε να εμφανίσουμε τον πίνακα με τα πλήθη, την εντολή :

```
include("display_totals.php");
```

Το **display\_totals.php** υπολογίζει και εμφανίζει τα σύνολα των πτυχιακών του συγκεκριμένου καθηγητή.

Οι παρακάτω γραμμές κώδικα είναι για να υπολογίσουμε τα σύνολα των προτεινόμενων, των υπό εξέλιξη, των ολοκληρωμένων και του συνόλου όλων των πτυχιακών, αντίστοιχα :

```
$lines0="select * from project where pointer=0 and t_login='$valid_user';  
$result0=mysql_query($lines0);  
$nor0=mysql_num_rows($result0);  
$lines1="select * from project where pointer=1 and t_login='$valid_user';  
$result1=mysql_query($lines1);  
$nor1=mysql_num_rows($result1);  
$lines2="select * from project where pointer=2 and t_login='$valid_user';  
$result2=mysql_query($lines2);  
$nor2=mysql_num_rows($result2);  
$nor3=$nor0+$nor1+$nor2;
```

Οι παρακάτω γραμμές κώδικα είναι για να εμφανίσουμε τον πίνακα με τα σύνολα για τον συγκεκριμένο καθηγητή :

```
echo "<br><div align=\"center\"><table border=\"1\">;  
echo "<tr><td align=\"left\"><p class=\"smallheaders\">ΚΑΤΗΓΟΡΙΑ</td><td  
align=\"center\"><p class=\"smallheaders\">ΠΛΗΘΟΣ</td></tr>;  
echo "<tr><td align=\"left\"><p class=\"smalltext\">Προτεινόμενες</td><td  
align=\"center\"><p class=\"smalltext\">$nor0</td></tr>;  
echo "<tr><td align=\"left\"><p class=\"smalltext\">Σε εξέλιξη</td><td align=\"center\"><p  
class=\"smalltext\">$nor1</td></tr>;  
echo "<tr><td align=\"left\"><p class=\"smalltext\">Ολοκληρωμένες</td><td  
align=\"center\"><p class=\"smalltext\">$nor2</td></tr>;  
echo "<tr><td align=\"left\" ><p class=\"smalltext\">Συνολικά</td><td align=\"center\"><p  
class=\"smalltext\">$nor3</td></tr>;  
echo "</table></div>;
```

**Πλήθος Γραμμών Κώδικα : 17**

**Απαιτούμενος Χρόνος Υλοποίησης : 15 λεπτά**

### 8.1.2 Σενάριο : Πλήθος Πτυχιακών Καθηγητή ανά κατηγορία και συνολικά με Ruby

Στον controller αναζήτηση μόνο για την συνδεδεμένο χρήστη αναλόγως το status της πτυχιακής :

```
@suggested=Project.find_all_by_status_and_teacher_id("suggested", current_teacher.id)  
@inprocess=Project.find_all_by_status_and_teacher_id("inprocess", current_teacher.id)  
@finished=Project.find_all_by_status_and_teacher_id("finished", current_teacher.id)  
@all=@suggested.count+@inprocess.count+@finished.count
```

Στο view (το count εμφανίζει τον αριθμό) :

```
<table border="1" align="center">
  <tr><td class="smallheaders">ΚΑΤΗΓΟΡΙΑ</td><td class="smallheaders">ΠΛΗΘΟΣ</td>
  <tr><td class="smalltext">Προτεινόμενες</td><td
class="smalltext"><%= @suggested.count% ></td>
  <tr><td class="smalltext">Σε εξέλιξη</td><td
class="smalltext"><%= @inprocess.count% ></td>
  <tr><td class="smalltext">Ολοκληρωμένες</td><td
class="smalltext"><%= @finished.count% ></td>
  <tr><td class="smalltext">Συνολικά</td><td class="smalltext"><%= @all% ></td>
</table>
```

**Πλήθος Γραμμών Κώδικα : 11**

**Απαιτούμενος Χρόνος Υλοποίησης (λεπτά) : 20 λεπτά**

## 8.2 Σενάριο : Μέσος Όρος Ημερών Εκπόνησης Πτυχιακών Καθηγητή

Προστίθεται η δυνατότητα στον καθηγητή όταν μπαίνει στις επιλογές του και επιλέγει «οι πτυχιακές μου» να βλέπει τον μέσο όρο εκπόνησης πτυχιακών σε μέρες, όπως υπολογίζονται από τις ολοκληρωμένες του πτυχιακές (βλέπε εικόνα 8.2).



ΚΑΤΗΓΟΡΙΑ	ΠΛΗΘΟΣ
Προτεινόμενες	1
Σε εξέλιξη	4
Ολοκληρωμένες	1
Συνολικά	6

Μέσος Όρος Συνολικών Ημερών Εκπόνησης : 90

Εικόνα 8.2

### 8.2.1 Σενάριο : Μέσος Όρος Ημερών Εκπόνησης Πτυχιακών Καθηγητή με PHP

Η προσθήκη αυτή τοποθετήθηκε στο κομμάτι των καθηγητών «Είσοδος Καθηγητή», στην επιλογή «οι πτυχιακές μου»

Διαδρομή : Είσοδος Καθηγητή -> οι πτυχιακές μου

Στο script **projects.php** προσθέτουμε στο σημείο που θέλουμε να εμφανίσουμε τον πίνακα με τα πλήθη, την εντολή :

```
include("display_date_duration.php");
```

Το **display\_date\_details.php** υπολογίζει και εμφανίζει τον μέσο όρο εκπόνησης σε ημέρες για όλες τις ολοκληρωμένες πτυχιακές του συγκεκριμένου καθηγητή. Ουσιαστικά γίνεται ένας υπολογισμός ώστε να βρεθούν οι συνολικές ημέρες μεταξύ της ημερομηνίας έναρξης εκπόνησης (date\_begin) και της ημερομηνίας ολοκλήρωσης (date\_end) και βγαίνει ο μέσος όρος αυτών για όλες τις πτυχιακές που έχει αναλάβει ο εν λόγω καθηγητής.

Οι παρακάτω γραμμές κώδικα είναι για να υπολογίσουμε και να εμφανίσουμε αυτό τον μέσο όρο :

```
$lines3="select * from project where pointer=2 and t_login='$valid_user';  
$result3=mysql_query($lines3);  
$nor3=mysql_num_rows($result3);  
$days = 0;  
if($nor3>0){  
    while($newArray=mysql_fetch_array($result3)){  
        $date1=$newArray['date_begin'];  
        $date2=$newArray['date_end'];  
        $day1 = strtotime($date1);  
        $day2 = strtotime($date2);  
        $oneday = 60 * 60 * 24;  
        $day = $day1;  
        while($day <= $day2){  
            $days = $days + 1;  
            $day = $day + $oneday;}}  
        $total_days = $total_days + $days;  
        $total_average = $total_days / $nor3;  
        echo "<p align=\"center\" class=\"smalltext\">Μέσος Όρος Συνολικών Ημερών  
Εκπόνησης : ".$total_average."</p>";  
    else  
        echo "<p align=\"center\" class=\"smalltext\">Μέσος Όρος Συνολικών Ημερών  
Εκπόνησης : 0</p>";
```

**Πλήθος Γραμμών Κώδικα : 20**

**Απαιτούμενος Χρόνος Υλοποίησης (λεπτά) : 13 λεπτά**

## 8.2.2 Σενάριο : Μέσος Όρος Ημερών Εκπόνησης Πτυχιακών Καθηγητή με RUBY

Στο view :

```
<% all_days=0 %>  
  <% if @finished!=nil %>  
    <% for finished in @finished %>  
      <% day1 = Date.parse("#{finished.begin_at}") %>  
      <% day2 = Date.parse("#{finished.end_at}") %>
```



```
<% one_project_day = (day2-day1).abs %>
<% all_days = all_days+one_project_day %>
<%end%>
<td class="smalltext"><%= "Μέσος Όρος Συνολικών Ημερών Εκπόνησης: #{all_days}"
%> </td>
<%else%>
<td class="smalltext"><%= "Μέσος Όρος Συνολικών Ημερών Εκπόνησης: 0" %> </td>
<%end%>
```

Στο controller κάνουμε την αναζήτηση για να βρούμε όλες τις ολοκληρωμένες πτυχιακές του συνδεδεμένου καθηγητή:







```
@finished=Project.find_all_by_status_and_teacher_id("finished", current_teacher.login)
```

**Πλήθος Γραμμών Κώδικα: 13**

**Απαιτούμενος Χρόνος Υλοποίησης (λεπτά) : 16 λεπτά**

### 8.3 Σενάριο : Πλήθος Πτυχιακών Όλων των Καθηγητών ανά κατηγορία και συνολικά

Προστίθεται η δυνατότητα στον καθηγητή που είναι και διαχειρίστης (admin) όταν μπαίνει στις επιλογές του και επιλέγει «διαχείριση χρηστών» να βλέπει μαζί με τα υπόλοιπα στοιχεία των χρηστών και στήλες με τα συνολα των πτυχιακών του (βλέπε εικόνα 8.3).

Username	Password	Όνοματεπώνυμο	Διαχειριστής		Προτ.	Εξελ.	Ολοκλ.	Όλες
achat	1234	Αλέξανδρος Χατζηγεωργίου	Ναι	 	0	0	1	1
gevan	1234	Georgios Evaggelidis	Ναι	 	0	0	0	0
louposk	1234	Loupos K	Ναι	 	0	0	0	0

Νέος Χρήστης

Επιστροφή

Εικόνα 8.3

#### 8.3.1 Σενάριο : Πλήθος Πτυχιακών Όλων των Καθηγητών ανά κατηγορία και συνολικά με PHP

Η προσθήκη αυτή τοποθετήθηκε στο κομμάτι των καθηγητών «Είσοδος Καθηγητή», στην επιλογή «διαχείριση χρηστών», στην οποία έχει πρόσβαση ένας καθηγητής μόνον εφόσον είναι μαρκαρισμένος ως «admin»

Διαδρομή : Είσοδος Καθηγητή -> διαχείριση χρηστών

Στο script **users.php** προσθέτουμε τις παρακάτω εντολές για να δημιουργήσουμε νέες στήλες με τίτλους που αφορούν στα σύνολα, στον πίνακα των χρηστών :

```
echo "<td width=\"10\"></td><td class=\"headers\" align=\"center\">Προτ.</td>";  
echo "<td width=\"10\"></td><td class=\"headers\" align=\"center\">Εξελ.</td>";  
echo "<td width=\"10\"></td><td class=\"headers\" align=\"center\">Ολοκλ.</td>";  
echo "<td width=\"10\"></td><td class=\"headers\" align=\"center\">Όλες</td></tr>";
```

Στη συνέχεια, στο σημείο που εμφανίζουμε έναν έναν τους χρήστες, προσθέτουμε τις παρακάτω γραμμές κώδικα, με τις οποίες υπολογίζουμε και εμφανίζουμε τα πλήθη των πτυχιακών για τον κάθε καθηγητή ανά κατηγορία και συνολικά :

```
$lines0="select * from project where pointer=0 and t_login='$login';"  
$result0=mysql_query($lines0);  
$nor0=mysql_num_rows($result0);  
$lines1="select * from project where pointer=1 and t_login='$login';"  
$result1=mysql_query($lines1);  
$nor1=mysql_num_rows($result1);  
$lines2="select * from project where pointer=2 and t_login='$login';"  
$result2=mysql_query($lines2);  
$nor2=mysql_num_rows($result2);  
$nor3=$nor0+$nor1+$nor2;  
echo "<td width=\"10\"></td><td class=\"maintext\" align=\"center\">$nor0</td>";  
echo "<td width=\"10\"></td><td class=\"maintext\" align=\"center\">$nor1</td>";  
echo "<td width=\"10\"></td><td class=\"maintext\" align=\"center\">$nor2</td>";  
echo "<td width=\"10\"></td><td class=\"maintext\" align=\"center\">$nor3</td>";
```

**Πλήθος Γραμμών Κώδικα : 18**

**Απαιτούμενος Χρόνος Υλοποίησης (λεπτά) : 12 λεπτά**

### 8.3.2 Σενάριο : Πλήθος Πτυχιακών Όλων των Καθηγητών ανά κατηγορία και συνολικά με RUBY

Προσθέτουμε γραμμές στο view file :

```
<td width="10"></td><td class="headers">Προτ.</td>  
<td width="10"></td><td class="headers">Εξελ.</td>  
<td width="10"></td><td class="headers">Ολοκλ.</td>  
<td width="10"></td><td class="headers">Όλες.</td>
```

```
<% @suggested = Project.find_all_by_status_and_teacher_id("suggested",teacher)%>
```







```
<% @inprocess = Project.find_all_by_status_and_teacher_id("inprocess", teacher)%>
<% @finished = Project.find_all_by_status_and_teacher_id("finished", teacher)%>
<% @all = Project.find_all_by_teacher_id(teacher)%>
<td width="10"></td><td class="headers"><%= @suggested.count %> </td>
<td width="10"></td><td class="headers"><%= @inprocess.count%></td>
<td width="10"></td><td class="headers"><%= @finished.count%></td>
<td width="10"></td><td class="headers"><%= @all.count %> </td>
```

**Πλήθος Γραμμών Κώδικα : 12**

**Απαιτούμενος Χρόνος Υλοποίησης (λεπτά) : 12 λεπτά**

## 8.4 Σενάριο : Μέσος Όρος Ημερών Εκπόνησης Πτυχιακών Όλων των Καθηγητών

Προστίθεται η δυνατότητα στον καθηγητή που είναι και διαχειρίστης (admin) όταν μπαίνει στις επιλογές του και επιλέγει «διαχείριση χρηστών» να βλέπει μαζί με τα υπόλοιπα στοιχεία των χρηστών και εκτός από τις στήλες με τα συνολα των πτυχιακών του, και τον μέσο όρο του χρόνου σε ημέρες για την εκπόνηση των πτυχιακών του (βλέπε εικόνα 8.4).

Username	Password	Όνοματεπώνυμο	Διαχειριστής		Προτ.	Εξελ.	Ολοκλ.	Όλες	Μ.Ο.ημ.
achat	1234	Αλέξανδρος Χατζηγεωργίου	Ναι	 	0	0	1	1	170
gevan	1234	Georgios Evaggelidis	Ναι	 	0	0	0	0	0
louposk	1234	Loupos K	Ναι	 	0	0	0	0	0

Νέος Χρήστης

Επιστροφή

**Εικόνα 8.4**

### 8.4.1 Σενάριο : Μέσος Όρος Ημερών Εκπόνησης Πτυχιακών Όλων των Καθηγητών με PHP

Η προσθήκη αυτή τοποθετήθηκε στο κομμάτι των καθηγητών «Είσοδος Καθηγητή», στην επιλογή «διαχείριση χρηστών», στην οποία έχει πρόσβαση ένας καθηγητής μόνον εφόσον είναι μαρκαρισμένος ως «admin»

Διαδρομή : Είσοδος Καθηγητή -> διαχείριση χρηστών

Στο script **users.php** προσθέτουμε την παρακάτω εντολή για να δημιουργήσουμε νέα στήλη για τον μέσο όρο που θέλουμε να εμφανίσουμε :

```
echo "<td width=\"10\"></td><td class=\"headers\" align=\"center\">Μ.Ο.ημ.</td></tr>";
```

Στη συνέχεια, στο σημείο που εμφανίζουμε έναν έναν τους χρήστες, προσθέτουμε τις παρακάτω γραμμές κώδικα, με τις οποίες υπολογίζουμε και εμφανίζουμε τον μέσο όρο των ημερών που απαιτούνται για την εκπόνηση των εργασιών για τον κάθε καθηγητή :

```
$lines3="select * from project where pointer=2 and t_login='$login';  
$result3=mysql_query($lines3);  
$nor3=mysql_num_rows($result3);  
$days = 0;  
if($nor3 > 0){  
    while($newArray=mysql_fetch_array($result3)){  
        $date1=$newArray['date_begin'];  
        $date2=$newArray['date_end'];  
        $day1 = strtotime($date1);  
        $day2 = strtotime($date2);  
        $oneday = 60 * 60 * 24;  
        $day = $day1;  
        while($day <= $day2){  
            $days = $days + 1;  
            $day = $day + $oneday;}}  
        $total_days = $total_days + $days;  
        $total_average = $total_days / $nor3;  
        echo "<td width=\"10\"></td><td class=\"maintext\"  
align=\"center\">$total_average</td>";  
        $total_average = 0;  
        $total_days = 0;}  
else  
    echo "<td width=\"10\"></td><td class=\"maintext\" align=\"center\">0</td>";
```

**Πλήθος Γραμμών Κώδικα : 24**

**Απαιτούμενος Χρόνος Υλοποίησης (λεπτά) : 12 λεπτά**

#### 8.4.2 Σενάριο : Μέσος Όρος Ημερών Εκπόνησης Πτυχιακών Όλων των Καθηγητών με RUBY

Το finished εμφανίζει το σύνολο των ολοκληρωμένων πτυχιακών για τον κάθε καθηγητή (όπως φαίνεται παραπάνω) :

```
<% all_days=0 %>  
<% if @finished!=nil %>  
  <% for finished in @finished %>  
    <% day1 = Date.parse("#{finished.begin_at}") %>  
    <% day2 = Date.parse("#{finished.end_at}") %>  
    <% one_project_day = (day2-day1).abs %>
```

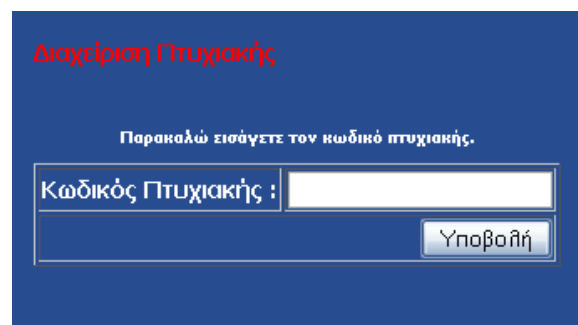
```
<% all_days = all_days+one_project_day %>
<%end%>
<td class="headers"><%= all_days %> </td>
<%else%>
    <td class="headers"><%= "0" %> </td>
<%end%>
```

**Πλήθος γραμμών κώδικα: 12**

**Απαιτούμενος Χρόνος Υλοποίησης (λεπτά) : 15 λεπτά**

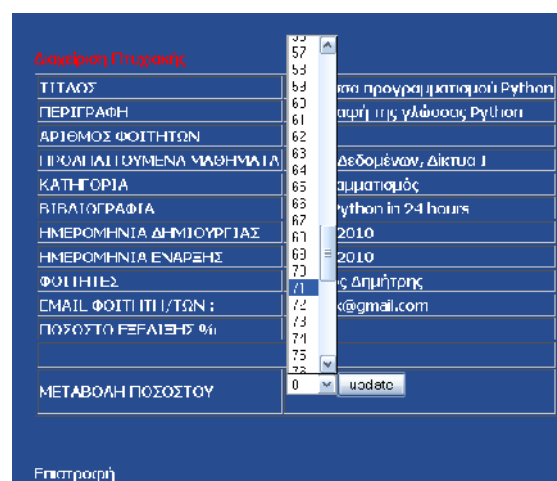
## 8.5 Σενάριο : Ορισμός Ποσοστού Εξέλιξης από τον Φοιτητή για την Εργασία που εκπονεί

Προστίθεται η δυνατότητα στον φοιτητή όταν μπαίνει στις επιλογές του και επιλέγει «διαχείριση πτυχιακής» (βλέπε εικόνα 8.5) να εισάγει τον κωδικό πτυχιακής όπως του έχει αποσταλεί κατά την ανάθεση με email.



**Εικόνα 8.5**

Εφόσον δώσει έγκυρο κωδικό βλέπει τα στοιχεία της πτυχιακής και έχει τη δυνατότητα να μεταβάλλει το ποσοστό εξέλιξης της πτυχιακής όπως εκείνος εκτιμά. (βλέπε εικόνα 8.6)



**Εικόνα 8.6**

### 8.5.1 Σενάριο : Ορισμός Ποσοστού Εξέλιξης από τον Φοιτητή για την Εργασία που εκπονεί με PHP

Η προσθήκη αυτή τοποθετήθηκε στο κομμάτι των φοιτητών «Είσοδος Φοιτητή», στην επιλογή «διαχείριση πτυχιακής», στην οποία έχει πρόσβαση ένας φοιτητής μόνον εφόσον έχει στη διάθεση του τον κωδικό πτυχιακής που του αποστέλλεται μέσω email κατά την ανάθεση της πτυχιακής.

Διαδρομή : Είσοδος Φοιτητή -> Διαχείριση Πτυχιακής

Στο script **student.php** προσθέτουμε την παρακάτω εντολή για να δημιουργήσουμε νέα επιλογή στο βασικό μενού επιλογών του φοιτητή :

```
<p><a href="percent.php?back=0" target="_self">Διαχείριση Πτυχιακής</a> </p>
```

Στη βάση δεδομένων στον πίνακα project προστέθηκε ένα πεδίο ακόμα με την παρακάτω **SQL εντολή** :

```
ALTER TABLE `project` ADD `percent` INT( 3 ) NOT NULL
```

Στη συνέχεια δημιουργούμε το **percent.php** το οποίο καλείται κατά τη διαχείριση πτυχιακής και ουσιαστικά ζητά κωδικό πτυχιακής για πρόσβαση στα στοιχεία της και μεταβολή του ποσοστού εξέλιξης της :

```
<?php
include("header.htm");
include("login.php");
include("select_db.php");
echo "<table><tr><td>&nbsp;</td>";
echo "<td class=\"mainheader\">Διαχείριση
Πτυχιακής</td></tr><tr><td>&nbsp;</td><td>";
if($_GET["percent"]==1){
    if($_GET["update"]==1){
        $mcode = $_GET['mcode'];
        $newpercent = $_POST["newpercent"];
        $sql2="update project set percent='$newpercent' where
m_code='$mcode'";
        $result2=mysql_query($sql2);}
    else
        $mcode = $_POST['mcode'];
    $sql="select
project_id,project_title,description,nos,pre_lessons,category,material,s_code,percent,date_
format(date_create,'%d-%m-%Y') as dcf,date_format(date_begin,'%d-%m-%Y') as dbf from
project where m_code='$mcode' and m_code != '' and pointer=1";
    $result=mysql_query($sql);
    $number_of_rows = mysql_num_rows($result);
if($number_of_rows > 0){
    $control = 2;
```

```
while($arow=mysql_fetch_array($aresult)){
    $code = $arow['project_id'];
    $s_code=$arow['s_code'];
    $email=$arow['s_email'];
    echo "<table border=1 align=\"top\"><tr>";
    echo "<td class=\"maintext\">ΤΙΤΛΟΣ</td>";
    echo "<td class=\"maintext\">";
    echo $arow["project_title"];
    echo "</td></tr>";
    echo "<tr>";
    echo "<td class=\"maintext\">ΠΕΡΙΓΡΑΦΗ</td>";
    echo "<td class=\"maintext\">";
    echo $arow["description"];
    echo "</td></tr>";
    echo "<tr>";
    echo "<td class=\"maintext\">ΑΡΙΘΜΟΣ ΦΟΙΤΗΤΩΝ</td>";
    echo "<td class=\"maintext\">";
    echo $arow["nos"];
    echo "</td></tr>";
    echo "<tr>";
    echo "<td class=\"maintext\">ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ ΜΑΘΗΜΑΤΑ</td>";
    echo "<td class=\"maintext\">";
    echo $arow["pre_lessons"];
    echo "</td></tr>";
    echo "<tr>";
    echo "<td class=\"maintext\">ΚΑΤΗΓΟΡΙΑ</td>";
    echo "<td class=\"maintext\">";
    echo $arow["category"];
    echo "</td></tr>";
    echo "<tr>";
    echo "<td class=\"maintext\">ΒΙΒΛΙΟΓΡΑΦΙΑ</td>";
    echo "<td class=\"maintext\">";
    echo $arow["material"];
    echo "</td></tr>";
    echo "<tr>";
    echo "<td class=\"maintext\">ΗΜΕΡΟΜΗΝΙΑ ΔΗΜΙΟΥΡΓΙΑΣ</td>";
    echo "<td class=\"maintext\">";
    if ($arow["dcf"]=='00-01-1900')
        echo "Αγνωστο";
    else
        echo $arow["dcf"];
    echo "</td></tr>";
    echo "<tr>";
    echo "<td class=\"maintext\">ΗΜΕΡΟΜΗΝΙΑ ΕΝΑΡΞΗΣ</td>";
    echo "<td class=\"maintext\">";
    if ($arow["dbf"]=='00-01-1900')
        echo "Αγνωστο";
    else
        echo $arow["dbf"];
    echo "</td></tr>";
```

```
        include("return_student.php");
        echo "<tr>";
        echo "<td class=\"maintext\">ΠΟΣΟΣΤΟ ΕΞΕΛΙΞΗΣ %</td>";
        echo "<td class=\"maintext\">";
        echo $arrow["percent"];
        echo "</td></tr>";
        echo "<tr><td colspan=2>&nbsp;</td></tr>";
        echo "<tr>";
        echo "<td class=\"maintext\">ΜΕΤΑΒΟΛΗ ΠΟΣΟΣΤΟΥ</td>";
        echo "<td class=\"maintext\">";
        echo "<form method=post
action=\"percent.php?percent=1&update=1&mcode=$mcode\" target=\"_top\"><select
name=\"newpercent\">";
        $i = 0;
        while($i <= 100){
            echo "<option value=\"\".$i.\">.$i.</option>";
            $i = $i + 1;
        }
        echo "</select>";
        echo "<input type=submit value=\"update\" name=\"update\"></form>";
        echo "</td></tr>";
    }
    echo "</table><br></br>";
    echo "</td></tr><tr><td>&nbsp;</td><td><a href=\"student.php\"
target=\"_self\">Επιστροφή</a></td></tr>";
}
else
    $control = 1;
}
if ($control<>2){
    if ($control==1)
        echo "<br><p align=\"center\" class=\"smalltext\">Λανθασμένη εισαγωγή. Η
πρόσβαση είναι αδύνατη. Παρακαλώ προσπαθήστε ξανά.</p>";
    else{
        echo "<br><p align=\"center\" class=\"smalltext\">Παρακαλώ εισάγετε τον κωδικό
πτυχιακής.</p>";
        echo "<form method=post action=\"percent.php?percent=1\">";
        echo "<table align=\"center\">";
        echo "<table border=1 align=\"center\">";
        echo "<tr><td class=\"maintext\">Κωδικός Πτυχιακής :</td>";
        echo "<td class=\"maintext\"><input type=password name=mcode
size=22></td></tr>";
        echo "<tr><td colspan=2 align=right>";
        echo "<input type=submit value=\"Υποβολή\"></td></tr>";
        echo "</table>";
        echo "</table></form>";
        echo "<p>&nbsp;</p>";
        echo "</td></tr><tr><td>&nbsp;</td><td><a href=\"student.php\"
target=\"_self\">Επιστροφή</a></td></tr>";
        echo "</table>";
```



```
}  
}  
include("footer.htm");  
?>
```

Τέλος, προσθέτουμε στο script inprocess.php την παρακάτω γραμμή ώστε να εμφανίζεται το ποσοστό εξέλιξης δίπλα από τον τίτλο της κάθε πτυχιακής που είναι σε εξέλιξη :

```
if($percent > 0) echo " (".$percent."%);
```

**Πλήθος Γραμμών Κώδικα : 118**

**Απαιτούμενος Χρόνος Υλοποίησης (λεπτά) : 40 λεπτά**

### 8.5.2 Σενάριο : Ορισμός Ποσοστού Εξέλιξης από τον Φοιτητή για την Εργασία που εκπονεί με RUBY

Εμφάνιση του Link στις επιλογές του φοιτητή:

```
<p><%= link_to "Διαχείριση πτυχιακής", students_project_code_login_url%></p>  
<p>&nbsp;</p>
```

Δημιουργία του migration για το πεδίο percent της βάσης :

```
class AddPercentInProjects < ActiveRecord::Migration  
  def self.up  
    add_column :projects, :percent, :string  
  end  
  
  def self.down  
    remove_column :projects, :percent, :string  
  end  
end
```

Το πεδίο που εισάγεται ο κωδικός :

```
<td>  
<p class="mainheader">Διαχείριση πτυχιακής</p>  
<br><br>  
<p class="smalltext">Παρακαλώ εισάγετε τον κωδικό πτυχιακής</p>  
<table border="1">  
  <% form_for :code, :url=>{:action=>"change_percent"} do |f| %>  
    <tr><td class="maintext">Κωδικός πτυχιακής:</td><td><%= f.password_field  
:message_code%> </td></tr>  
    <tr><td colspan="2" class="maintext"><%= f.submit "Υποβολή" %></td></tr>  
  <% end %>  
</table>  
<p><%= link_to "Επιστροφή", student_menu_url%> </p>  
</td>
```

## Σύγκριση scripting γλωσσών προγραμματισμού και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

Η εμφάνιση των στοιχείων της πτυχιακής και ή εμφάνιση της φόρμας αλλαγής ποσοστού:

```
<td>
  <p class="mainheader">Διαχείριση πτυχιακής</p>
  <table border="1" class="maintext">
    <tr><td>ΤΙΤΛΟΣ:</td><td><%= @project.title %> </td></tr>
    <tr><td>ΠΕΡΙΓΡΑΦΗ:</td><td><%= @project.description %> </td></tr>
    <tr><td>ΑΡΙΘΜΟΣ ΦΟΙΤΗΤΩΝ:</td><td><%= @project.no_students %> </td></tr>
    <tr><td>ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ ΜΑΘΗΜΑΤΑ:</td><td><%= @project.pre_lessons %>
  </td></tr>
    <tr><td>ΚΑΤΗΓΟΡΙΑ:</td><td><%= @project.category %> </td></tr>
    <tr><td>ΒΙΒΛΙΟΓΡΑΦΙΑ:</td><td><%= @project.material %> </td></tr>
    <tr><td>ΗΜΕΡΟΜΗΝΙΑ ΔΗΜΙΟΥΡΓΙΑΣ:</td><td><%= @project.created_at.strftime("%d-
  %m-%Y") %> </td></tr>
    <tr><td>ΗΜΕΡΟΜΗΝΙΑ ΕΝΑΡΞΗΣ:</td><td><%= @project.begin_at.strftime("%d-%m-
  %Y") %> </td></tr>
    <tr><td>ΦΟΙΤΗΤΕΣ:</td><td><%= @project.student.names %> </td></tr>
    <tr><td>EMAIL ΦΟΙΤΗΤΗ/ΤΩΝ</td><td><%= @project.student.email %> </td></tr>
    <tr><td>ΠΟΣΟΣΤΟ ΕΞΕΛΙΞΗΣ %</td><td>
    <%if @project.percent%> <%= @project.percent %> <%else%> <%= "0" %><%end%>
  </td></tr>
    <tr><td colspan="2">&nbsp;</td></tr>
    <tr><td>ΜΕΤΑΒΟΛΗ ΠΟΣΟΣΤΟΥ:</td><td>
    <% form_for @project, :url=>student_path do |f|%>
    <%= f.hidden_field :id, :value=>@project.id %>
    <%= f.select :percent, (1..100) %>
    <%= f.submit "Μεταβολή" %>
    <%end%>
  </td></tr>
  </table>

  <%= link_to "Επιστροφή", student_menu_url %>
</td>
```

Όσον αφορά τις ενέργειες που γίνονται στον controller:

```
class StudentsController < ApplicationController
  layout "projects"

  #Για την εμφάνιση των στοιχείων της πτυχιακής
  def show
    @project = Project.find(params[:id])
    @one_project = Project.new

    respond_to do |format|
      format.html # show.html.erb
      format.xml { render :xml => @student }
    end
  end
end
```

```
end

def edit
  @project = Project.find(params[:code])
end

#Για την ανανέωση του ποσοστού το παίρνει από την φόρμα
def update
  @project = Project.find(params[:id])

  respond_to do |format|
    if @project.update_attributes(params[:project])
      format.html { redirect_to(:back, :notice => 'Το ποσοστό ενημερώθηκε με επιτυχία!') }
      format.xml { head :ok }
    else
      format.html { render :action => "edit" }
      format.xml { render :xml => @student.errors, :status => :unprocessable_entity }
    end
  end
end

def project_code_login
  @code = Project.new
end

def change_percent
  @code = Project.new(params[:code])
  @message = @code.message_code
  @project_found = Project.find_by_message_code(@message)

  respond_to do |format|
    if @project_found
      format.html { redirect_to(:action=>'show',:id=>@project_found) }
    else
      format.html { redirect_to(:back, :notice => "Λανθασμένη εισαγωγή. Παρακαλώ
δοκιμάστε ξανά.") }
    end
  end
end

end
```

Στη διαχείριση των πτυχιακών σε εξέλιξη βάζουμε το παρακάτω που εμφανίζει το ποσοστό δίπλα από κάθε πτυχιακή (με bold το κείμενο που προστέθηκε):

```
<td class="smalltext">Όνοματα πτυχιακών
<table>
  <% @inprocess.each do |project| %>
  <tr>
    <td class="maintext"><%= link_to "#{project.title}",project_inproces_path(project) %>
    <% if project.percent %>
```

```
<%= "#{project.percent}%" %>
<%else%>
  <%= "" %>
<%end%>
</td>
</tr>
<% end %>
</table>
</td>
```

**Πλήθος Γραμμών Κώδικα : 92**

**Απαιτούμενος Χρόνος Υλοποίησης (λεπτά) : 60 λεπτά**

## 8.6 Συγκριτικός Πίνακας PHP – RUBY

Όπως φαίνεται και από τον παρακάτω πίνακα σε όλα τα σενάρια η Ruby on Rails χρησιμοποιεί λιγότερο κώδικα, περίπου κατά 30%, σε σύγκριση με τη PHP. Όσον αφορά στον χρόνο που δαπανείθηκε για την υλοποίηση των σεναρίων βλέπουμε σ'όλες τις περιπτώσεις ότι με την PHP είχαμε γρηγορότερο αποτέλεσμα κατά περίπου 25 - 30%. Βέβαια το θέμα του χρόνου είναι κατά κάποιον τρόπο υποκειμενικό καθώς η εμπειρία μου στη συγγραφή PHP κώδικα είναι πολύ μεγαλύτερη, οπότε το αποτέλεσμα ως προς το χρόνο μπορεί να θεωρηθεί ίσο.

Σενάριο Επέκτασης	Code Lines		Time Elapsed (minutes)	
	PHP	Ruby on Rails	PHP	Ruby on Rails
Πλήθος Πτυχιακών Καθηγητή ανά κατηγορία και συνολικά	17	11	15	20
Μέσος Όρος Ημερών Εκπόνησης Πτυχιακών Καθηγητή	20	13	15	16
Πλήθος Πτυχιακών όλων των Καθηγητών ανά κατηγορία και συνολικά	18	12	11	12
Μέσος Όρος Ημερών Εκπόνησης Πτυχιακών όλων των Καθηγητών	24	12	13	15
Ορισμός Ποσοστού Εξέλιξης από τον Φοιτητή για την Εργασία που εκπονεί	118	92	40	60
<b>Σύνολο :</b>	<b>197</b>	<b>140</b>	<b>94</b>	<b>123</b>

Θεωρώ ότι η «εξωτική» για τα ελληνικά δεδομένα, scripting γλώσσα προγραμματισμού Ruby υπόσχεται πολλά και βάσει της κίνησης της κυρίως στο εξωτερικό (προς το παρόν τουλάχιστον) αξίζει κάποιος να εμβαθύνει, αφού μπορεί να υλοποιήσει ζητούμενα με πολύ πιο απλό τρόπο αρκεί να αποκτήσει την απαραίτητη γνώση και εμπειρία στην γλώσσα αυτή.

## 9. ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην παρούσα διπλωματική εργασία, αρχικά παρουσιάστηκαν οι δύο scripting γλώσσες προγραμματισμού PHP και RUBY και στη συνέχεια αναλύθηκε ένα περιβάλλον το οποίο θα αναπτυσσόταν και από τις δύο. Έγινε αναλυτική περιγραφή της βάσης δεδομένων και όλης της δομής του περιβάλλοντος καθώς και του πηγαίου κώδικα για συγκεκριμένα σενάρια υλοποίησης. Παρουσιάστηκαν επίσης έγκυρες συγκρίσεις μεταξύ των δύο scripting γλωσσών προγραμματισμού με μετρήσεις που αφορούν μνήμη, ταχύτητα και αριθμό γραμμών κώδικα για διάφορα προγράμματα και αλγορίθμους.

Το να συγκρίνει κάποιος τη Ruby On Rails με τη PHP είναι κατά κάποιο τρόπο άδικο. Η PHP είναι απλά μια γλώσσα προγραμματισμού. Παρόλα αυτά είναι εξαιρετικά ευέλικτη και χρήσιμη. Απαιτεί ελάχιστη μνήμη και εκτελείται ταχύτατα. Είναι μια ώριμη γλώσσα με μεγάλη ποικιλία επεκτάσεων και βιβλιοθηκών. Ενώ αρχικά σχεδιάστηκε για να χρησιμοποιείται μέσα από HTML σελίδες, σταδιακά εξελίχθηκε σε full object oriented χαρακτηριστικά.

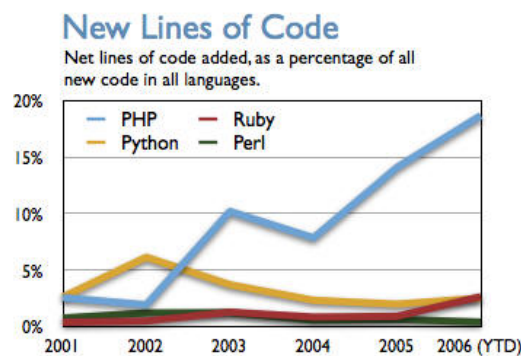


Η PHP ως γλώσσα δεν διαθέτει όλα τα χαρακτηριστικά που διαθέτει η Ruby on Rails. Αυτό έχει ως αποτέλεσμα, όταν δημιουργείται ένα Site από το μηδέν με PHP, απαιτούνται καλές γνώσεις προγραμματισμού καθώς διαδικασίες όπως η διασύνδεση με τη βάση δεδομένων προγραμματίζονται “from scratch”. Παρόλα αυτά υπάρχουν και frameworks για τη PHP με χαρακτηριστικά παρόμοια με αυτά της Ruby on Rails. Ένα από τα δυνατά πλεονεκτήματα της PHP είναι ότι ο προγραμματιστής μπορεί να επιλέξει το framework που τον βολεύει με τα χαρακτηριστικά μόνο που χρειάζεται. Αντίθετα με την Ruby on Rails είναι αναγκασμένος να χρησιμοποιήσει μόνο όσα του δίνονται. Αυτό μπορεί να αποτελέσει μεγάλο πρόβλημα καθώς περιορίζει τη δυνατότητα στο να επιτευχθούν κάποιες απαιτήσεις μιας εφαρμογής ως προς τη λειτουργία ή την εμφάνιση της.

Η Ruby on Rails είναι ένα framework ανεπτυγμένο σε Ruby γλώσσα προγραμματισμού. Ως βάση έχει σχεδιαστεί με την φιλοσοφία του αντικειμενοστραφή προγραμματισμού (object oriented). Έχει δομημένη σύνταξη και κομψή σχεδίαση. Διαθέτει κάποια πολύ δυνατά χαρακτηριστικά. Υποστηρίζει την σχεδίαση ActiveRecord όπου κάνει “map” τις εγγραφές της βάσης δεδομένων σε αντικείμενα. Το ActiveRecord κρύβει πολλές SQL εντολές από τον προγραμματιστή με τον χειρισμό αυτόματων μηχανισμών δημιουργίας, τροποποίησης και διαγραφής αντικειμένων. Κοινώς ο προγραμματιστής δε γράφει SQL εντολές και δε χρειάζεται να γνωρίζει. Δημιουργούνται αυτομάτως μέσω του ActiveRecord. Το Ruby on Rails επίσης περιλαμβάνει ένα σύστημα δημιουργίας URL’s σε μεθόδους και ένα

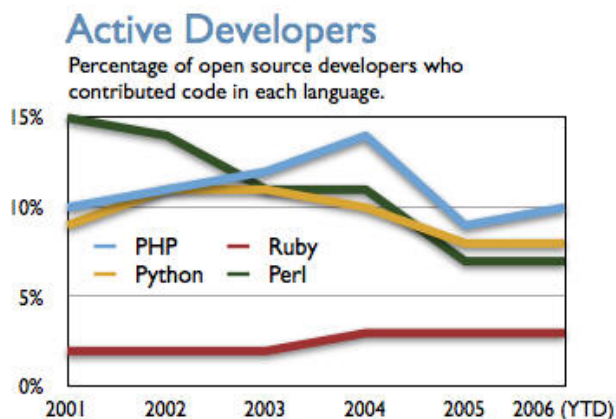
σύστημα δημιουργίας προτύπου για παραγωγή HTML κώδικα. Υπάρχουν και άλλα πολλά χαρακτηριστικά συμπεριλαμβανομένου και των συνόδων λειτουργίας.

Στο παρελθόν, η Ruby on Rails ήταν πιο περίπλοκη. Πλέον είναι το ίδιο απλή και ακόμα περισσότερο από την PHP αφού κάποια πράγματα τα κάνει αυτόματα χωρίς να απαιτούνται γνώσεις από τον προγραμματιστή (πχ. Διαχείριση στοιχείων από βάση δεδομένων). Σύμφωνα με μετρήσεις<sup>[22]</sup>, για παρόμοιες εφαρμογές απαιτεί λιγότερο κώδικα, λιγότερη μνήμη κατά 1/3 και είναι πιο γρήγορη ακόμα και 3 φορές σε σχέση με τη PHP. Παρακάτω εμφανίζεται ποσοστό κώδικα που απαιτείται επιπλέον σε μια ήδη υπάρχουσα εφαρμογή. Βλέπουμε ότι τα ποσοστά της PHP είναι πολύ μεγαλύτερα πράγμα που σημαίνει πως για μια περαιτέρω διεργασία απαιτεί περισσότερο κόπο σε σχέση με τη Ruby.



Εικόνα 9.1

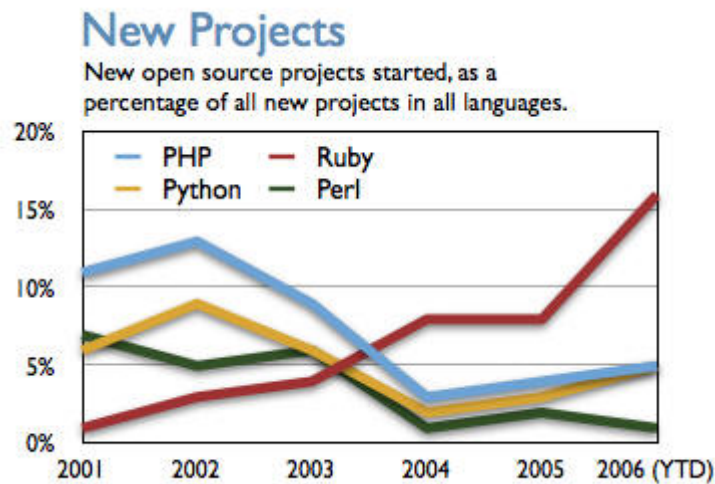
Κάποιοι μπορεί να προτιμήσουν την PHP λόγω της μεγάλης διάδοσης της και της εξαιρετικής της ευελιξίας. Υπάρχουν επίσης frameworks για PHP όπως πχ. Το CakePHP με παρόμοια χαρακτηριστικά. Παρόλα αυτά η ραγδαία ανάπτυξη της Ruby σε συνδιασμό με την αύξηση της χρήσης της σε νέες εφαρμογές (twitter – μια από τις πιο δημοφιλείς ιστοσελίδες παγκοσμίως), την εξαιρετική απλότητα της, και τις ικανοποιητικές της επιδόσεις την καθιστούν ως την «γλώσσα του νέου κύματος εφαρμογών» (the language of the next wave of applications). Παρακάτω βλέπουμε μια ραγδαία πτώση των εν' ενεργεία προγραμματιστών ως προς τη χρήση της PHP και παράλληλα μια αργή αλλά σταθερή άνοδο της Ruby, ως γλώσσα προτίμησης τους.



Εικόνα 9.2

## Σύγκριση scripting γλωσσών προγραμματισμού και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

Ήδη στο εξωτερικό οι προγραμματιστές Ruby είναι περιζήτητοι. Παρότι στην Ελλάδα η ζήτηση είναι ελάχιστη, δεν αποκλείεται στα επόμενα χρόνια να αυξηθεί το ενδιαφέρον και να δημιουργηθούν απαιτήσεις γνώσης της παρούσας γλώσσας. Παρακάτω φαίνεται κιόλας η μεγάλη και ραγδαία αύξηση του ενδιαφέροντος ως προς τον προγραμματισμό με γλώσσα Ruby σε νέες εφαρμογές και παράλληλα η δραματική μείωση στην επιλογή της PHP.



**Εικόνα 9.3**

Εκπαιδευτικά, θεωρώ ότι θα είχε πολύ μεγάλο ενδιαφέρον η προώθηση της Ruby σε συνδυασμό με 2-3 ακόμη scripting γλώσσες προγραμματισμού συμπεριλαμβανομένης και της PHP, μέσα από μια αλυσίδα μαθημάτων προγραμματισμού διαδικτύου, όπου οι φοιτητές θα επέλεγαν μετά από μια βασική παρουσίαση, τη scripting γλώσσα που προτιμούν για την υλοποίηση κάποιας εργασίας στα πλαίσια του μαθήματος. Αυτό θα μπορούσε να δώσει πολύ περισσότερα συμπεράσματα στον κόσμο των scripting γλωσσών προγραμματισμού.

## Παράρτημα Α : ScreenShots Ιστοσελίδας

### Α.1 Οδηγίες χρήσης (πρώτα βήματα)

#### Α.1.1 Αρχική σελίδα

Από την αρχική σελίδα έχουμε δύο βασικές εισόδους :

- την είσοδο καθηγητή και
- την είσοδο φοιτητή.

Για την είσοδο του, ο καθηγητής θα πρέπει να γνωρίζει τα username και password που έχει στον server της σχολής.



**Εικόνα Α.1.1**



## A.2 Οδηγίες χρήσης (φοιτητής)

### A.2.1 Είσοδος φοιτητών

Κατά την είσοδο του, ο φοιτητής, αντικρίζει δύο βασικές επιλογές (βλέπε εικόνα A.2.1):

- την λίστα πτυχιακών και
- την διαχείριση πτυχιακής



**Εικόνα A.2.1**

## A.2.2 Λίστα Πτυχιακών

Η επιλογή αυτή (βλέπε εικόνα A.2.2) περιέχει έναν μηχανισμό αναζήτησης με την εισαγωγή μιας λέξης ή πρότασης “κλειδί” και με επιλογές ταξινομημένης εμφάνισης. Για παράδειγμα κάποιος φοιτητής που ψάχνει πτυχιακές που αφορούν τα νευρωνικά δίκτυα και θέλει να εμφανισθούν κατά φθίνουσα σειρά με βάση την ημερομηνία δημιουργίας (δηλαδή αρχίζοντας από τις πιο πρόσφατες, και καταλήγοντας στις πιο παλιές), θα μπορούσε να εισάγει στο πεδίο “Εισαγωγή λέξης κλειδί”, την πρόταση νευρωνικά δίκτυα, να τσεκάρει την επιλογή “Φθίνουσα Σειρά” και να επιλέξει στην επιλογή “με βάση”, την Ημερομηνία δημιουργίας.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ  
UNIVERSITY OF MACEDONIA

Ηλεκτρονικό Σύστημα  
Διαχείρισης & Δημοσίευσης  
Πτυχιακών Εργασιών

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ  
ΦΟΙΤΗΤΗΣ : ΠΕΤΡΟΥ ΒΑΣΙΛΕΙΟΣ (Α.Μ. 21/09)  
ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ : ΧΑΤΖΙΓΕΩΡΓΙΟΥ ΑΛΕΞΑΝΔΡΟΣ

Λίστα Πτυχιακών

Αναζήτηση Πτυχιακής

Εισαγωγή λέξης κλειδί:  Αναζήτηση

Εμφάνιση κατά:  Αύξουσα Σειρά  Φθίνουσα Σειρά

με βάση   
την Ημερομηνία Δημιουργίας  
τον Τίτλο  
το username του καθηγητή

Εμφάνιση ανά Κατηγορία

Προτεινόμενες  Σε εξέλιξη  
 Ολοκληρωμένες  Όλες οι κατηγορίες

Επιστροφή

Εικόνα A.2.2

## Σύγκριση scripting γλωσσών προγραμματισμού και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

Στις Εικόνες A.2.3 και A.2.4 εμφανίζονται δύο περιπτώσεις αναζήτησης λέξης κλειδιού, όπου στην πρώτη δεν υπάρχει εγγραφή που να σχετίζεται με την λέξη κλειδί, ενώ στη δεύτερη, υπάρχει.

Ηλεκτρονικό Σύστημα  
Διαχείρισης & Δημοσίευσης  
Πτυχιακών Εργασιών

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ  
UNIVERSITY OF MACEDONIA

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ  
ΦΟΙΤΗΤΗΣ : ΠΕΤΡΟΥ ΒΑΣΙΛΕΙΟΣ (Α.Μ. 21/09)  
ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ : ΧΑΤΖΗΓΕΩΡΓΙΟΥ ΑΛΕΞΑΝΔΡΟΣ

Αναζητήστε Αναζητήστε  
Αναζήτηση Πτυχιακής

Εισαγωγή λέξης κλειδί:

Εμφάνιση κατά:  Αύξουσα σειρά  Φθίνουσα σειρά

με βάση

Δεν βρέθηκε κατάλληλη εγγραφή για το κλειδί εισαγμένο στο πλαίσιο αναζήτησης.

Επιστροφή

Εικόνα A.2.3

Ηλεκτρονικό Σύστημα  
Διαχείρισης & Δημοσίευσης  
Πτυχιακών Εργασιών

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ  
UNIVERSITY OF MACEDONIA

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ  
ΦΟΙΤΗΤΗΣ : ΠΕΤΡΟΥ ΒΑΣΙΛΕΙΟΣ (Α.Μ. 21/09)  
ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ : ΧΑΤΖΗΓΕΩΡΓΙΟΥ ΑΛΕΞΑΝΔΡΟΣ

Αναζητήστε Αναζητήστε  
Αναζήτηση Πτυχιακής

Εισαγωγή λέξης κλειδί:

Εμφάνιση κατά:  Αύξουσα σειρά  Φθίνουσα σειρά

με βάση

Βρέθηκαν μία εγγραφές.

ΤΙΤΛΟΣ :	test0
ΚΑΤΑΣΤΑΣΗ :	σε εξέλιξη
ΚΑΘΗΓΗΤΗΣ :	Διαχειριστής
ΕΜΒΛΗ ΚΑΘΗΓΗΤΗ :	admin@il.lelilhe.gr
ΕΠΗΡΕΑΦΗ :	test
ΑΡΙΘΜΟΣ ΣΠΟΥΔΑΣΤΩΝ :	1-2
ΠΡΟΑΠΛΟΤΥΜΕΝΑ ΜΑΘΗΜΑΤΑ :	test
ΚΑΤΗΓΟΡΙΑ :	test

Εικόνα A.2.4

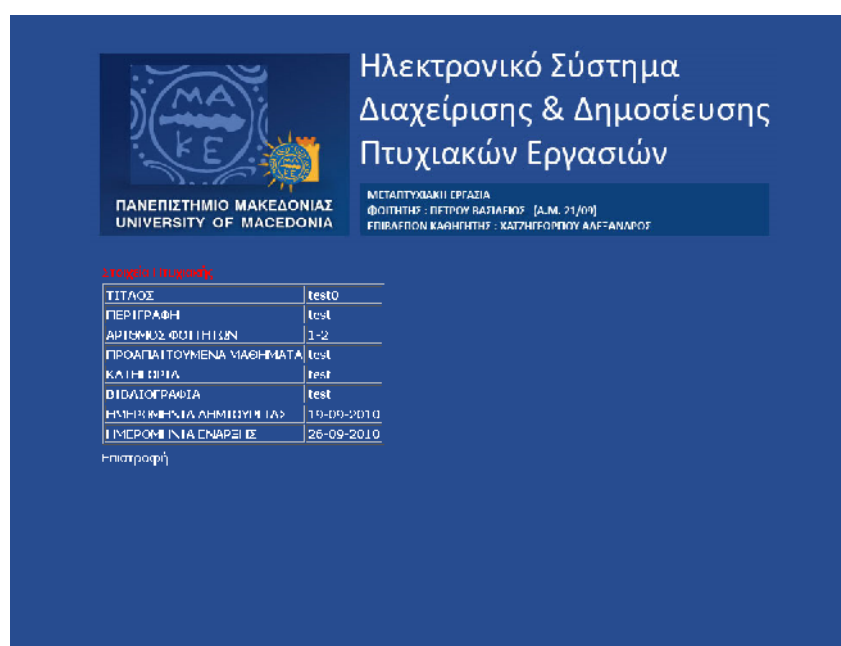
## Σύγκριση scripting γλωσσών προγραμματισμού και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

Η λίστα πτυχιακών, εκτός του μηχανισμού αναζήτησης, περιέχει επίσης και μια επιλογή εμφάνισης ανά κατηγορία. Επιλέγοντας για παράδειγμα τις πτυχιακές που βρίσκονται **σε εξέλιξη**, εμφανίζεται μια λίστα από πτυχιακές (βλέπε Εικόνα A.2.5). Συγκεκριμένα για την κάθε πτυχιακή σε εξέλιξη, εμφανίζεται ο τίτλος, το ονοματεπώνυμο του καθηγητή που την εισηγείται και το email του.



Εικόνα A.2.5

Κάνοντας κλικ πάνω στον τίτλο εμφανίζονται λεπτομέρειες που αφορούν τη συγκεκριμένη πτυχιακή (βλέπε Εικόνα A.2.6).



Εικόνα A.2.6

## Σύγκριση scripting γλωσσών προγραμματισμού και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

Και στις τρεις κατηγορίες πτυχιακών (Προτεινόμενες, σε Εξέλιξη, Ολοκληρωμένες) εμφανίζονται τα εξής στοιχεία :

- Τίτλος
- Περιγραφή
- Αριθμός φοιτητών
- Προαπαιτούμενα μαθήματα
- Κατηγορία
- Βιβλιογραφία
- Ημερομηνία δημιουργίας

Στις πτυχιακές σε εξέλιξη και στις ολοκληρωμένες εμφανίζονται επίσης τα εξής στοιχεία :

- Ημερομηνία έναρξης
- Φοιτητές
- Email φοιτητών

Τέλος στις ολοκληρωμένες εμφανίζονται επίσης τα εξής στοιχεία :

- Ημερομηνία λήξης
- Εισηγητής
- Εξεταστής
- Βαθμολογία
- Παρατηρήσεις

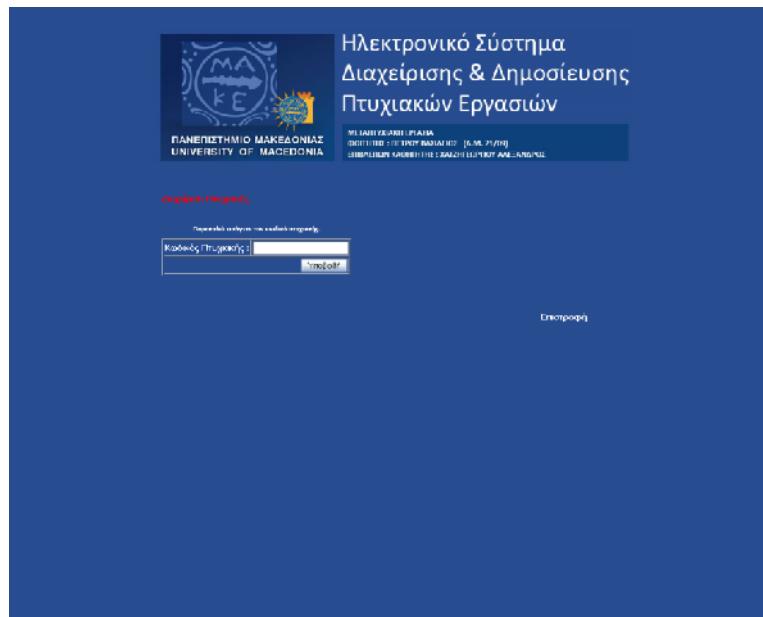
Για τις ολοκληρωμένες πτυχιακές, εμφανίζεται (όχι πάντα) και η επιλογή αρχείου (βλέπε Εικόνα A.2.7) με ένα εικονίδιο που αντικατοπτρίζει τον τύπο του αρχείου. Το αρχείο αυτό αφορά την συγκεκριμένη πτυχιακή και οι φοιτητές έχουν τη δυνατότητα να το κάνουν “download” και να το μελετήσουν.



Εικόνα A.2.7

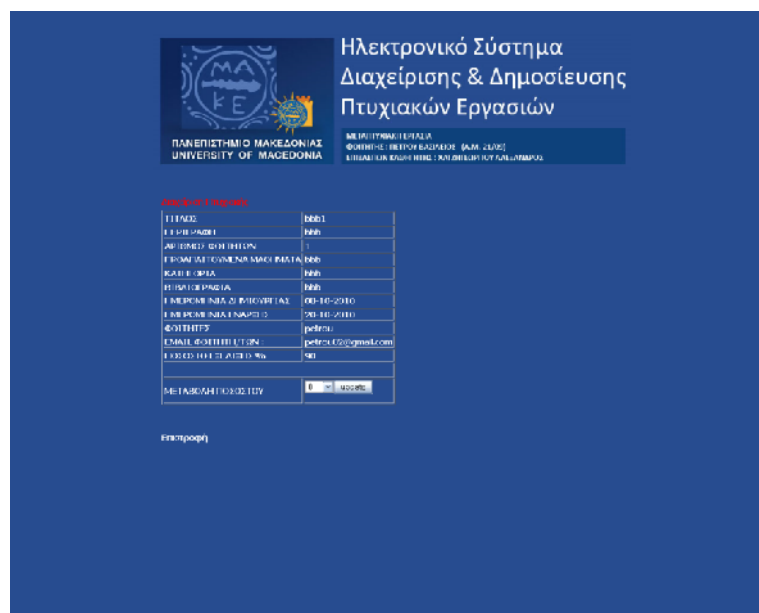
### A.2.3 Διαχείριση Πτυχιακής

Η επιλογή “διαχείριση πτυχιακής” αφορά **μόνο** τους φοιτητές που εκπονούν πτυχιακή. Δηλαδή αφορά μόνο τις πτυχιακές που είναι **σε εξέλιξη**. Για να διαχειριστεί ένας φοιτητής την πτυχιακή του, πρέπει να εισάγει τον κωδικό πτυχιακής (βλέπε εικόνα A.2.8) που του έχει αποσταλεί με email κατά την ανάθεση της πτυχιακής.



Εικόνα A.2.8

Εισάγοντας σωστά τον κωδικό πτυχιακής, έχει πλέον πρόσβαση στα στοιχεία της πτυχιακής και δυνατότητα να μεταβάλλει το ποσοστό εξέλιξης της πτυχιακής (βλέπε εικόνα A.2.9).



Εικόνα A.2.9

## A.3 Οδηγίες χρήσης (καθηγητής)

### A.3.1 Είσοδος καθηγητή

Κατά την είσοδο του, ο καθηγητής, αντικρίζει έναν έλεγχο πρόσβασης (βλέπε εικόνα 3.1) και καλείται να εισάγει τα username και password που έχει στον server της σχολής.



Εικόνα A.3.1

Σε περίπτωση όπου εισάγει τα στοιχεία σωστά και δεν του επιτρέπεται η πρόσβαση θα πρέπει να επικοινωνήσει με τον διαχειριστή του συστήματος.

Κάνοντας κλικ πάνω στο όνομα του, εκτελείτε αυτομάτως το προεπιλεγμένο πρόγραμμα αποστολής email για την αποστολή του μηνύματος. Ο καθηγητής που δεν του επιτρέπεται η πρόσβαση, δεν έχει παρά να γράψει στον διαχειριστή το ονοματεπώνυμο και το username του, ώστε ο δεύτερος να επαληθεύσει τα στοιχεία και να κάνει προσθήκη του username του στο σύστημα.



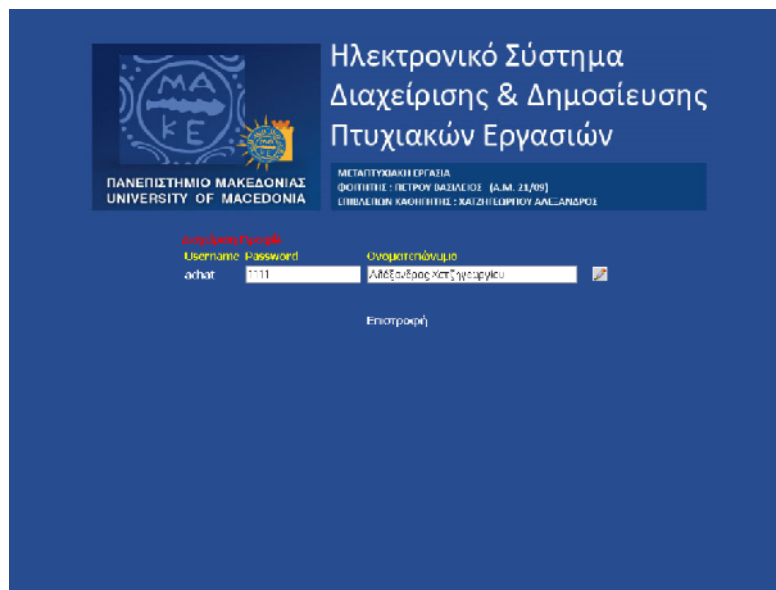
Εικόνα A.3.3

Κατά την αρχική σελίδα του, ο καθηγητής, αντικρίζει τρεις βασικές επιλογές (βλέπε εικόνα 3.3):

- τις πτυχιακές μου
- την λίστα πτυχιακών και
- την διαχείριση του προφίλ

Η λίστα πτυχιακών είναι ακριβώς η ίδια με αυτή των φοιτητών (βλέπε ενότητα 2.2) ενώ η επιλογή “οι πτυχιακές μου” παρουσιάζεται στην αμέσως επόμενη ενότητα.

Η επιλογή “ διαχείριση προφίλ ” εμφανίζει τα υπάρχοντα στοιχεία του χρήστη τα οποία μπορεί να τροποποιήσει (βλέπε εικόνα 3.4).



**Εικόνα A.3.4**

Κάνοντας κλικ στην επιλογή “αποσύνδεση” γίνεται αποσύνδεση του καθηγητή από το σύστημα και επιστροφή στην αρχική σελίδα πτυχιακών εργασιών.

### **A.3.2 Οι πτυχιακές μου**

Στο κομμάτι “οι πτυχιακές μου” (βλέπε εικόνα A.3.5), ο καθηγητής έχει τη δυνατότητα να επιλέξει μια κατηγορία και να κάνει τα εξής :

- **Επιλέγοντας “Προτεινόμενες” :**  
Μπορεί να δημιουργήσει κάποιο νέο θέμα πτυχιακής, να δει όλα τα προϋπάρχοντα θέματα πτυχιακής που έχει ως τώρα προτείνει και δεν έχουν αναλάβει σπουδαστές και να διαγράψει κάποιο ή κάποια από αυτά, αν το επιθυμεί. Τέλος, μπορεί να κάνει ανάθεση προτεινόμενης πτυχιακής, σε σπουδαστές που δηλώνουν ενδιαφέρον.



- **Επιλέγοντας “Σε εξέλιξη” :**  
Μπορεί να δει όλα τα θέματα που έχει αναλάβει και είναι σε εξέλιξη, να μεταβάλει το περιεχόμενο τους και να διαγράψει κάποιο ή κάποια από αυτά, αν το επιθυμεί. Μπορεί επίσης να δει το ποσοστό εξέλιξης της πτυχιακής όπως έχει εκτυμήσει και έχει δηλώσει ο φοιτητής, να δηλώσει την ολοκλήρωση κάποιου θέματος εφόσον έχει παραδοθεί και παρουσιαστεί από τον σπουδαστή ή τους σπουδαστές που το είχαν αναλάβει. Μπορεί τέλος να ακυρώσει κάποια πτυχιακή και να την επαναφέρει σε “Προτεινόμενη”.
- **Επιλέγοντας “Ολοκληρωμένη” :**  
Μπορεί να δει όλα τα θέματα πτυχιακής που έχει αναλάβει και έχουν ολοκληρωθεί. Να φορτώσει κάποιο αρχείο που είναι σχετικό με κάποια πτυχιακή ή ακόμα και να διαγράψει φορτωμένο αρχείο. Μπορεί επίσης να εισάγει και στοιχεία από πτυχιακές που έχουν παρελθόν, να τις μεταβάλλει και να τις διαγράψει αν το επιθυμεί.
- **Επιλέγοντας “Όλες οι κατηγορίες” :**  
Μπορεί να δει όλα τα θέματα πτυχιακής, που έχει αναλάβει ως τώρα και είναι οποιασδήποτε μορφής, προτεινόμενα, σε εξέλιξη, ολοκληρωμένα.



Εικόνα A.3.5

### A.3.2.1 Προτεινόμενες πτυχιακές

Αρχικά πρέπει να ειπωθεί πως αν ο καθηγητής δεν έχει προτεινόμενες πτυχιακές η μόνη επιλογή που του εμφανίζεται είναι η προσθήκη νέου θέματος (βλέπε εικόνα A.3.6).



**Εικόνα A.3.6**

Όπως είναι απόλυτα λογικό, οι επιλογές :

- Μεταβολή θέματος
- Διαγραφή θέματος
- Ανάθεση πτυχιακής

εμφανίζονται μόνο αν υπάρχουν προτεινόμενες πτυχιακές.

### A.3.2.1.1 Προσθήκη νέου θέματος

Επιλέγοντας “προσθήκη νέου θέματος”, εμφανίζεται μια φόρμα εισαγωγής στοιχείων νέου θέματος (βλέπε εικόνα A.3.7).

The screenshot shows a web form titled "Ηλεκτρονικό Σύστημα Διαχείρισης & Δημοσίευσης Πτυχιακών Εργασιών" (Electronic System for Management & Publication of Undergraduate Theses). The form is on a blue background and includes the following fields and text:

- Logo of the University of Macedonia (ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ UNIVERSITY OF MACEDONIA).
- Faculty information: ΜΓΠΑ ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ, ΦΟΙΤΗΤΗΣ : ΠΕΤΡΟΣ ΒΑΣΙΛΓΙΟΣ (Α.Μ. 71/09), ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ : ΧΑΤΖΗΓΕΩΡΓΙΟΥ ΑΓΓΕΛΑΡΟΣ.
- Form title: Προσθήκη νέου θέματος.
- Code field: Κωδικός : achat X10-11 001.
- Title field: \* Τίτλος : Ηλεκτρονικό σύστημα διαχείρισης και δημοσίευσης πτυχιακών εργασιών.
- Instruction: (Η παρακάτω μην εκτάγεται στον τίτλο το σύμβολο της τελείας και της διπλής απόστροφου)
- Student ID field: \*Αριθμός φοιτητών : [input field].
- Description field: \* Περιγραφή : Ηλεκτρονικό σύστημα διαχείρισης και δημοσίευσης πτυχιακών εργασιών.

Εικόνα A.3.7

**Προσοχή!!** Για λόγους σωστής λειτουργίας της ιστοσελίδας, προτείνεται η αποφυγή εισαγωγής συμβόλων όπως τελεία ή διπλή απόστροφος, στον τίτλο.

Ο κωδικός πτυχιακής αποτελείται από 16 χαρακτήρες και παράγεται αυτόματα μέσω ενός μηχανισμού.

Πρέπει να εισαχθούν όλα τα ζητούμενα στοιχεία. Συγκεκριμένα, τα στοιχεία που ζητούνται είναι:

Τίτλος πτυχιακής (αλφαριθμητικό μέχρι 150 χαρακτήρες).

Αριθμός σπουδαστών (αλφαριθμητικό μέχρι 20 χαρακτήρες).

Περιγραφή (κείμενο μέχρι 65.535 χαρακτήρες).

Προαπαιτούμενα μαθήματα (αλφαριθμητικό μέχρι 100 χαρακτήρες).

Κατηγορία (αλφαριθμητικό μέχρι 80 χαρακτήρες).

Βιβλιογραφία (αλφαριθμητικό μέχρι 100 χαρακτήρες).

## Σύγκριση scripting γλωσσών προγραμματισμού και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

Μετά τη προσθήκη νέου θέματος, επιστρέφοντας στις προτεινόμενες πτυχιακές, εμφανίζονται όλες οι επιλογές (βλέπε εικόνα Α.3.8) καθώς και οι τίτλοι όλων των προτεινόμενων πτυχιακών που έχουν δημιουργηθεί ως τώρα.

Ηλεκτρονικό Σύστημα  
Διαχείρισης & Δημοσίευσης  
Πτυχιακών Εργασιών

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ  
ΦΟΙΤΗΤΗΣ : ΠΕΤΡΟΥ ΒΑΣΙΛΕΙΟΣ (Α.Μ. 21/09)  
ΕΠΙΒΛΗΤΗΝ ΚΑΘΗΓΗΤΗ : ΧΑΤΖΙΠΕΤΡΙΟΥ ΔΑΔ - ΑΝΔΡΕΑ

Προτεινόμενες Πτυχιακές

Ηλεκτρονικό σύστημα διαχείρισης και δημοσίευσης πτυχιακών εργασιών

Επιλογές :

- προσθήκη νέου θέματος
- τροποποίηση θέματος
- διαγραφή θέματος
- ανάθεση πτυχιακής

Επιστροφή

Εικόνα Α.3.8

Κάνοντας κλικ πάνω στον τίτλο της πτυχιακής, εμφανίζονται οι λεπτομέρειες της (βλέπε εικόνα Α.3.9)

Ηλεκτρονικό Σύστημα  
Διαχείρισης & Δημοσίευσης  
Πτυχιακών Εργασιών

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ  
ΦΟΙΤΗΤΗΣ : ΠΕΤΡΟΥ ΒΑΣΙΛΕΙΟΣ (Α.Μ. 21/09)  
ΕΠΙΒΛΗΤΗΝ ΚΑΘΗΓΗΤΗ : ΧΑΤΖΙΠΕΤΡΙΟΥ ΔΑΔ - ΑΝΔΡΕΑ

Προτεινόμενη Πτυχιακή

Στοιχεία :

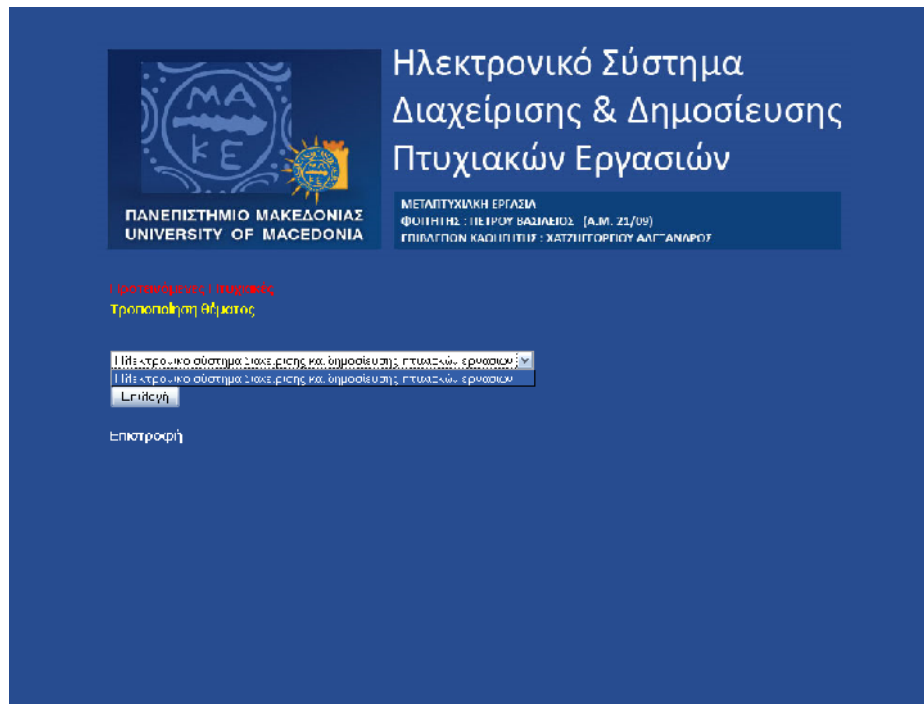
ΚΩΔΙΚΟΣ	αδhat_χ10_Π1_001
ΤΙΤΛΟΣ	Ηλεκτρονικό σύστημα διαχείρισης και δημοσίευσης πτυχιακών εργασιών
ΠΕΡΙΓΡΑΦΗ	Ηλεκτρονικό σύστημα διαχείρισης και δημοσίευσης πτυχιακών εργασιών. Ηλεκτρονικό σύστημα διαχείρισης και δημοσίευσης πτυχιακών εργασιών. Ηλεκτρονικό σύστημα διαχείρισης και δημοσίευσης πτυχιακών εργασιών.
ΑΡΙΘΜΟΣ ΦΟΙΤΗΤΩΝ	1
ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ ΜΑΘΗΜΑΤΑ	προγραμματισμός Π1, βάσεις δεδομένων Ι, βάσεις δεδομένων ΙΙ
ΚΛΕΙΔΙΑ	script προγραμματισμός
ΒΙΒΛΙΟΓΡΑΦΙΑ	PHP Programming, Alexandros Chatzigeorgiou
ΗΜΕΡΟΜΗΝΙΑ ΔΗΜΙΟΥΡΓΙΑΣ	19-10-2010

Επιστροφή

Εικόνα Α.3.9

### A.3.2.1.2 Τροποποίηση θέματος

Επιλέγοντας “**τροποποίηση θέματος**”, εμφανίζεται μια μπάρα επιλογής τίτλου προτεινόμενης πτυχιακής για μεταβολή (βλέπε εικόνα 3.10).



Εικόνα A.3.10

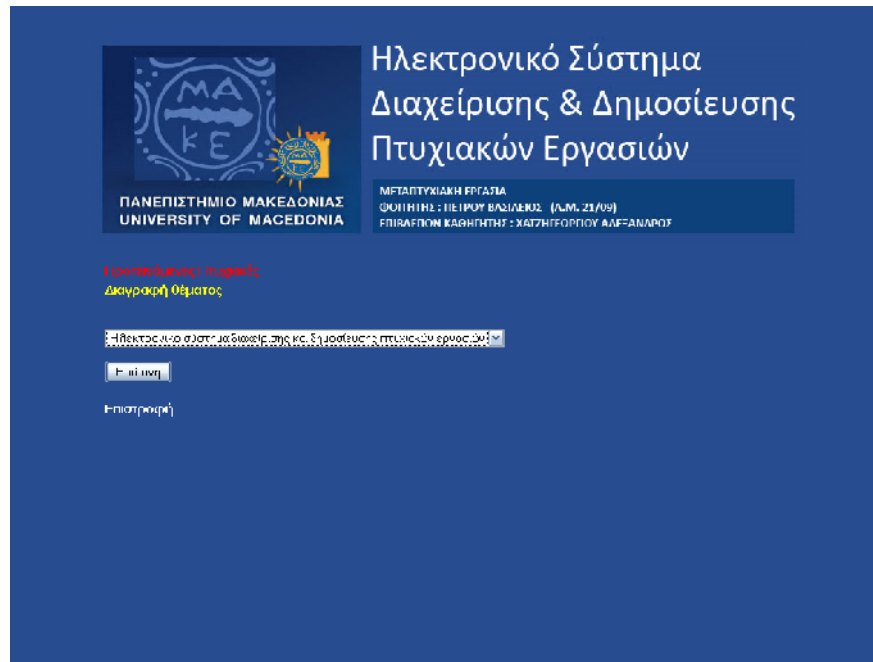
Επιλέγοντας τον τίτλο εμφανίζονται τα στοιχεία της πτυχιακής για μεταβολή (βλέπε εικόνα A.3.11).



Εικόνα A.3.11

### A.3.2.1.3 Διαγραφή θέματος

Επιλέγοντας “**Διαγραφή θέματος**”, εμφανίζονται όλοι οι τίτλοι προτεινόμενων πτυχιακών και ο καθηγητής επιλέγει ποια θέλει να διαγράψει (βλέπε εικόνα A.3.12).



Εικόνα A.3.12

### A.3.2.1.4 Ανάθεση πτυχιακής

Επιλέγοντας “**Ανάθεση πτυχιακής**”, εμφανίζεται μια μπάρα επιλογής τίτλου προτεινόμενης πτυχιακής για ανάθεση (βλέπε εικόνα A.3.13).



Εικόνα A.3.13

## Σύγκριση scripting γλωσσών προγραμματισμού και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

Μετά την επιλογή τίτλου εμφανίζονται αναλυτικά τα στοιχεία της πτυχιακής και ζητούνται δύο στοιχεία των οποίων η εγκυρότητα είναι αναγκαία (βλέπε εικόνα A.3.14). Τα στοιχεία αυτά είναι :

- Τα ονοματεπώνυμα των φοιτητών που θα αναλάβουν τη πτυχιακή (αλφαριθμητικό μέχρι 150 χαρακτήρες).
- Το email του φοιτητή (αλφαριθμητικό μέχρι 50 χαρακτήρες).

Εικόνα A.3.14

### Προσοχή!!

- Για την καλύτερη εμφάνιση, όταν έχουμε περισσότερα από ένα ονοματεπώνυμα, πρέπει να εισάγουμε ανάμεσα παύλα.

Π.χ. Πέτρου Βασίλειος-Λούπος Κωνσταντίνος

- Ακόμα και στη περίπτωση όπου περισσότεροι από ένας αναλαμβάνουν την ίδια πτυχιακή, πρέπει να εισάγεται ΜΟΝΟ ένα email.

Στη συνέχεια γίνεται έλεγχος για το αν έχουν εισαχθεί και τα δύο στοιχεία. Μετά τον έλεγχο εμφανίζεται ανάλογο μήνυμα. Η πτυχιακή πλέον αλλάζει κατηγορία και από “προτεινόμενη” γίνεται “σε εξέλιξη”. Παράλληλα με την διαδικασία της ανάθεσης αποστέλλεται και email στον φοιτητή το οποίο περιέχει τον τίτλο της πτυχιακής και τον κωδικό πτυχιακής (βλέπε εικόνα A.3.15) με τον οποίο θα μπορεί ο φοιτητής να διαχειρίζεται την εργασία του και συγκεκριμένα να ορίζει το ποσοστό εξέλιξης της. Αυτό ισχύει για όσο διάστημα η πτυχιακή είναι σε εξέλιξη.



Εικόνα A.3.15

### A.3.2.2 Πτυχιακές σε εξέλιξη

Καταρχάς, αν δεν υπάρχουν πτυχιακές σε εξέλιξη για τον καθηγητή, δεν εμφανίζεται καμιά επιλογή.

Αν υπάρχουν πτυχιακές σε εξέλιξη, εμφανίζονται οι παρακάτω επιλογές (βλέπε εικόνα A.3.16) :

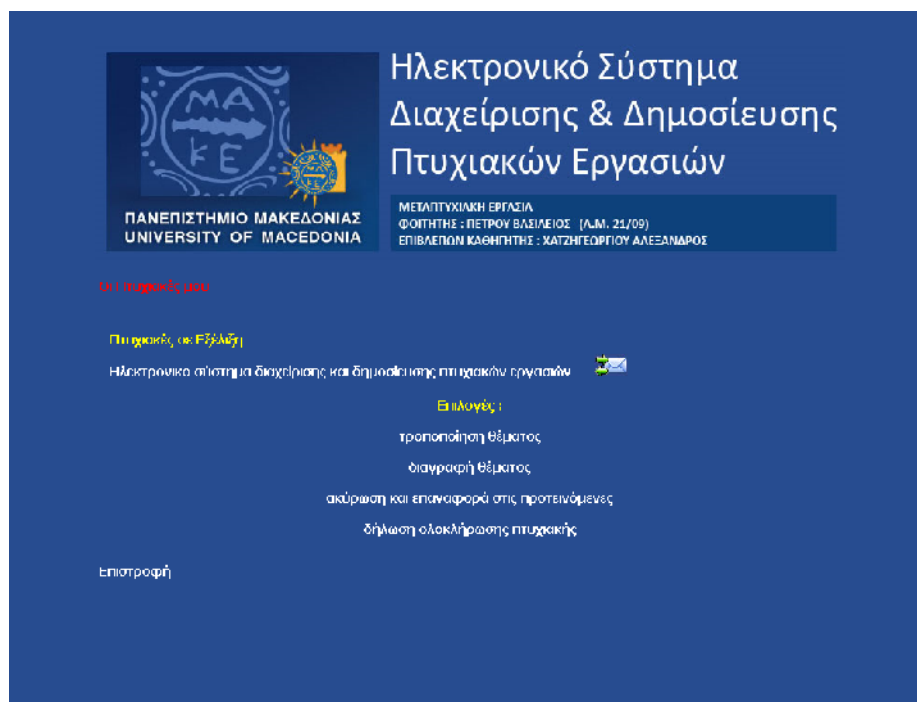
- Τροποποίηση θέματος
- Διαγραφή θέματος
- Ακύρωση και επαναφορά στις προτεινόμενες
- Δήλωση ολοκλήρωσης πτυχιακής

Η τροποποίηση θέματος και η διαγραφή θέματος λειτουργούν ακριβώς το ίδιο όπως και στις προτεινόμενες πτυχιακές.

Η επιλογή “Ακύρωση και επαναφορά στις προτεινόμενες” είναι αναγκαία σε περιπτώσεις όπως όταν ένας σπουδαστής αφήνει μια πτυχιακή χωρίς να την ολοκληρώσει. Τότε ο καθηγητής μπορεί να την επαναφέρει στις προτεινόμενες. Διαδικαστικά, ο καθηγητής επιλέγει τον τίτλο της πτυχιακής. Στη συνέχεια, εμφανίζονται οι λεπτομέρειες της κι ένα κουμπί υποβολής για να γίνει η επαναφορά.

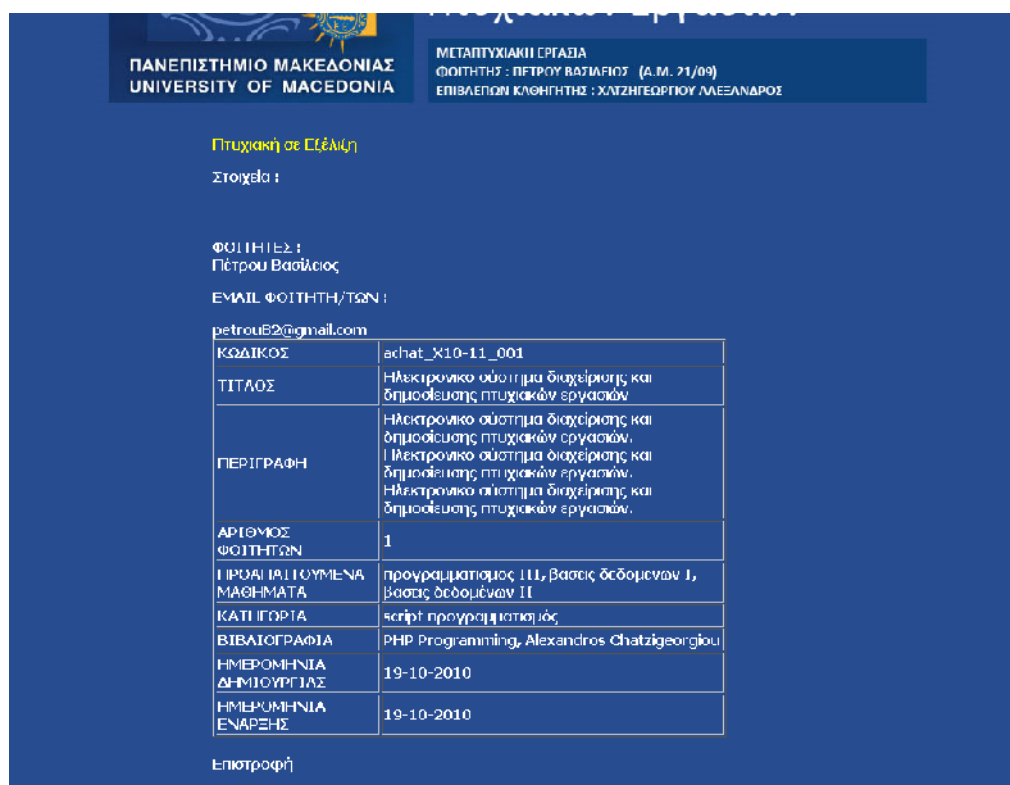


Σύγκριση scripting γλωσσών προγραμματισμού  
και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού



Εικόνα Α.3.16

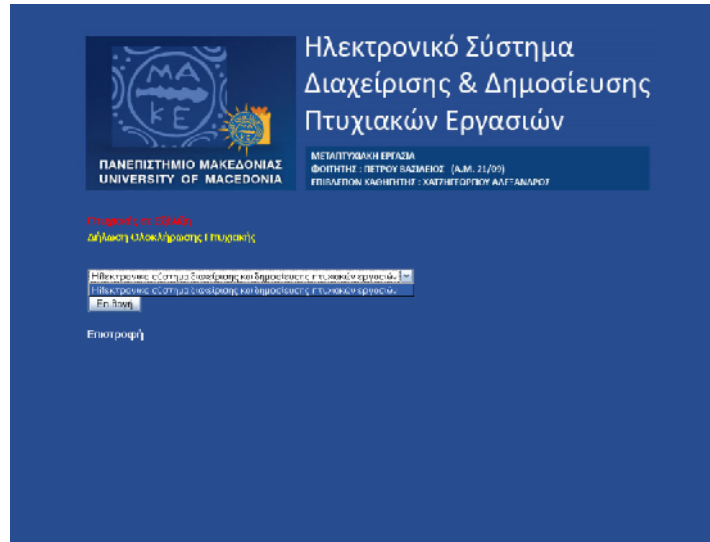
Κάνοντας κλικ πάνω στον τίτλο της πτυχιακής σε εξέλιξη, εμφανίζονται οι λεπτομέρειες της (βλέπε εικόνα Α.3.17).



Εικόνα Α.3.17

### A.3.2.2.1 Δήλωση ολοκλήρωσης πτυχιακής

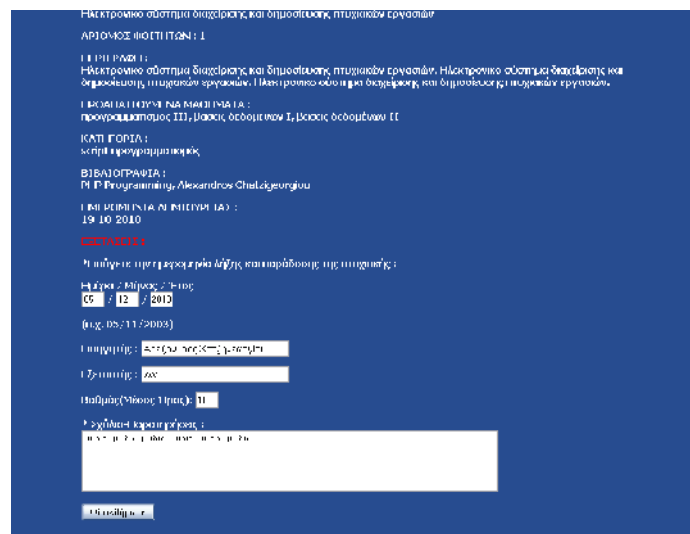
Επιλέγοντας “**Δήλωση ολοκλήρωσης πτυχιακής**”, εμφανίζεται μια μπάρα επιλογής τίτλου πτυχιακής σε εξέλιξη που έχει ολοκληρωθεί (βλέπε εικόνα A.3.18).



Εικόνα A.3.18

Στη συνέχεια εμφανίζονται οι λεπτομέρειες της πτυχιακής και μια φόρμα εισαγωγής στοιχείων (βλέπε εικόνα A.3.19). Ο καθηγητή, υποχρεούται να εισάγει τα παρακάτω στοιχεία :

- Ημερομηνία λήξης (ημερομηνία).
- Εισηγητής (Αλφαριθμητικό μέχρι 50 χαρακτήρες).
- Εξεταστής (Αλφαριθμητικό μέχρι 50 χαρακτήρες).
- Μέσος Όρος (Αλφαριθμητικό μέχρι 5 χαρακτήρες).
- Παρατηρήσεις (Κείμενο μέχρι 65.535 χαρακτήρες).



Εικόνα A.3.19

## Σύγκριση scripting γλωσσών προγραμματισμού και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού

Στη συνέχεια γίνεται η δήλωση και η πτυχιακή πλέον ανήκει στη λίστα των ολοκληρωμένων πτυχιακών (βλέπε εικόνα A.3.20).



**Εικόνα A.3.20**

### A.3.2.3 Ολοκληρωμένες πτυχιακές

Καταρχάς, αν δεν υπάρχουν πτυχιακές σε εξέλιξη για τον καθηγητή, εμφανίζεται μόνο η επιλογή “Εισαγωγή ολοκληρωμένης πτυχιακής” (βλέπε εικόνα A.3.24).

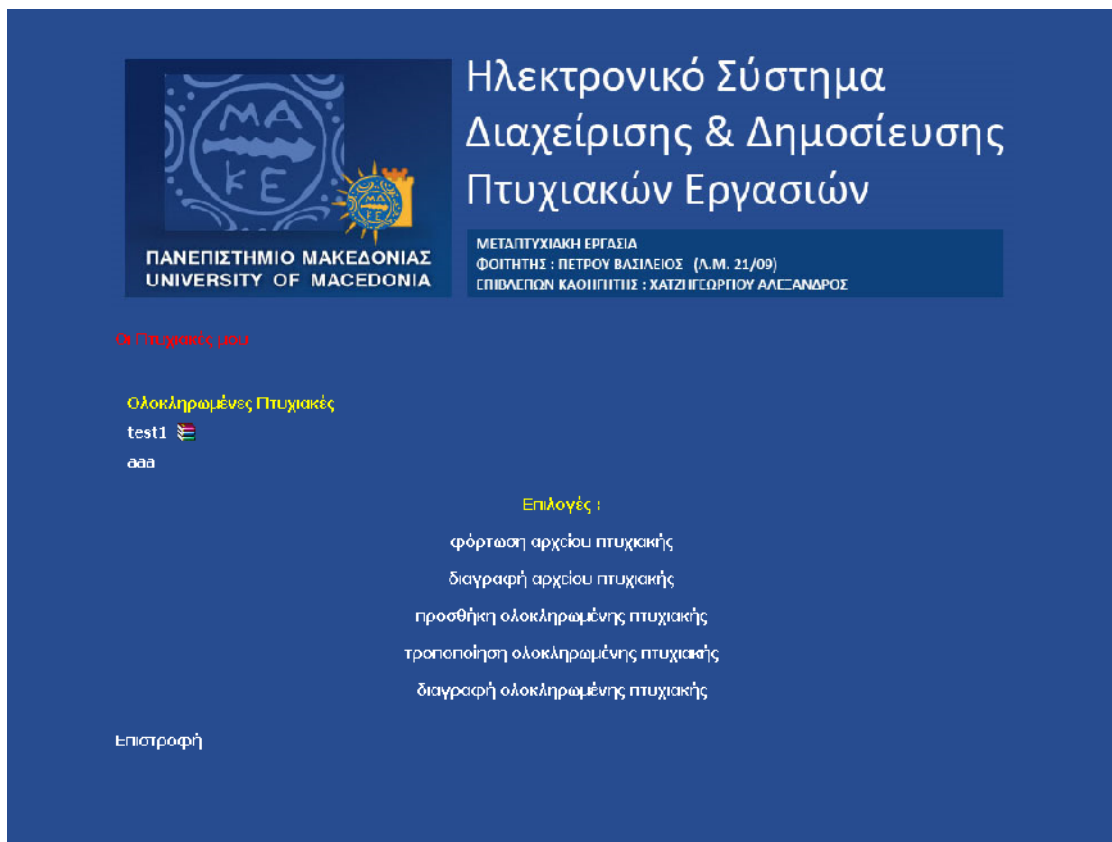


**Εικόνα A.3.24**

Η επιλογή “Εισαγωγή ολοκληρωμένης πτυχιακής”, είναι αναγκαία σε περιπτώσεις όταν ο καθηγητής θέλει να προσθέσει στη λίστα του, τις

ολοκληρωμένες πτυχιακές που εκπονήθηκαν κατά το παρελθόν και βρίσκονται στο ράφι του.

Οι πτυχιακές εργασίες που δημιουργούνται απ' ευθείας ως ολοκληρωμένες και όχι σταδιακά από προτεινόμενες σε εξέλιξη και μετά σε ολοκληρωμένες, εμφανίζονται στη λίστα μ' έναν αστερίσκο μπροστά και ο καθηγητής έχει το δικαίωμα να τις μεταβάλλει και να τις διαγράψει αν θέλει. Αντιθέτως, αυτές που έχουν δημιουργηθεί σταδιακά στο σύστημα εις γνώσιν του φοιτητή, με το που ολοκληρώνονται παύει ο καθηγητής να έχει το οποιοδήποτε δικαίωμα τροποποίησης και διαγραφής. Η διαφορά αυτή έχει να κάνει κυρίως με ηθικούς κανόνες αλλά και με το γεγονός ότι στις πτυχιακές που δημιουργούνται σταδιακά, ο καθηγητής έχει αρκετό χρόνο στο να κάνει ό,τι τροποποιήσεις χρειάζονται.







**Εικόνα A.3.25**

Κάνοντας κλικ πάνω στον τίτλο ολοκληρωμένης πτυχιακής, εμφανίζονται οι λεπτομέρειες της.

Το μόνο δικαίωμα που έχει ο καθηγητής για όλες τις ολοκληρωμένες πτυχιακές, είναι η φόρτωση (upload) και η διαγραφή σχετικού αρχείου.

Οι πτυχιακές που έχουν φορτωμένο αρχείο, εμφανίζονται στη λίστα με ένα από τα παρακάτω εικονίδια (βλέπε εικόνα A.3.25) :

-  Έγγραφο word.
-  Έγγραφο pdf.
-  Συμπιεσμένο αρχείο zip.
-  Συμπιεσμένο αρχείο rar.

Κάνοντας κλικ πάνω σε κάποιο από τα εικονίδια, εκτελείται προς ανάγνωση ή αποθήκευση, αρχείο σχετικό με την πτυχιακή.

Η μεταβολή θέματος και η διαγραφή θέματος λειτουργούν ακριβώς το ίδιο όπως και στις προτεινόμενες πτυχιακές.

#### A.3.2.3.1 Εισαγωγή ολοκληρωμένης πτυχιακής

Επιλέγοντας “εισαγωγή ολοκληρωμένης πτυχιακής”, εμφανίζεται μια φόρμα εισαγωγής στοιχείων (βλέπε εικόνα A.3.26).



\*Βιβλιογραφία :

\*Εισάγετε την ημερομηνία δημιουργίας της πτυχιακής :

Ημέρα / Μήνας / Έτος  
□ / □ / □  
(π.χ. 05/11/2003)

\*Εισάγετε την ημερομηνία έναρξης της πτυχιακής :

Ημέρα / Μήνας / Έτος  
□ / □ / □  
(π.χ. 05/11/2003)

\*Εισάγετε την ημερομηνία λήξης της πτυχιακής :

Ημέρα / Μήνας / Έτος  
□ / □ / □  
(π.χ. 05/11/2003)

\*Παρακαλώ δώστε τα ονοματεπώνυμα των φοιτητών που ανέλαβαν την πτυχιακή :

□

„Για περισσότερα από ένα ονοματεπώνυμα, παρακαλώ εισάγετε ανάμεσα καύλα!  
π.χ. Πέτρου Βασίλειος-Χριστούδης Ιωάννης.

Παρακαλώ εισάγετε το email του φοιτητή :

□

Εικόνα A.3.26

Ο κωδικός πτυχιακής αποτελείται από 16 χαρακτήρες και παράγεται αυτόματα μέσω ενός μηχανισμού.

Υποχρεωτική είναι η εισαγωγή όλων των στοιχείων με αστερίσκο στ' αριστερά.

Συγκεκριμένα, τα στοιχεία που ζητούνται είναι:

- Τίτλος πτυχιακής (αλφαριθμητικό μέχρι 150 χαρακτήρες).
- Αριθμός σπουδαστών (αλφαριθμητικό μέχρι 20 χαρακτήρες).
- Περιγραφή (κείμενο μέχρι 65.535 χαρακτήρες).
- Προαπαιτούμενα μαθήματα (αλφαριθμητικό μέχρι 100 χαρακτήρες).
- Κατηγορία (αλφαριθμητικό μέχρι 80 χαρακτήρες).
- Βιβλιογραφία (αλφαριθμητικό μέχρι 100 χαρακτήρες).
- Ημερομηνία δημιουργίας (ημερομηνία).
- Ημερομηνία έναρξης (ημερομηνία).
- Ημερομηνία λήξης (ημερομηνία).
- Ονοματεπώνυμο σπουδαστών (αλφαριθμητικό μέχρι 100 χαρακτήρες).
- Email σπουδαστών (αλφαριθμητικό μέχρι 50 χαρακτήρες).
- Ονοματεπώνυμο εισηγητή (αλφαριθμητικό μέχρι 50 χαρακτήρες).
- Ονοματεπώνυμο εξεταστή (αλφαριθμητικό μέχρι 50 χαρακτήρες).
- Τελική βαθμολογία εξέτασης (αλφαριθμητικό μέχρι 5 χαρακτήρες).
- Παρατηρήσεις Εξέτασης (κείμενο μέχρι 65.535 χαρακτήρες).

**Προσοχή!!** Στον τίτλο δεν επιτρέπεται να εισάγονται σύμβολα όπως η τελεία ή η διπλή απόστροφος.

### A.3.2.3.2 Φόρτωση αρχείου πτυχιακής

Επιλέγοντας “Φόρτωση αρχείου πτυχιακής”, εμφανίζεται μια μπάρα επιλογής τίτλου ολοκληρωμένων πτυχιακών (βλέπε εικόνα A.3.27) οι οποίες όμως, δεν έχουν φορτωμένο αρχείο.

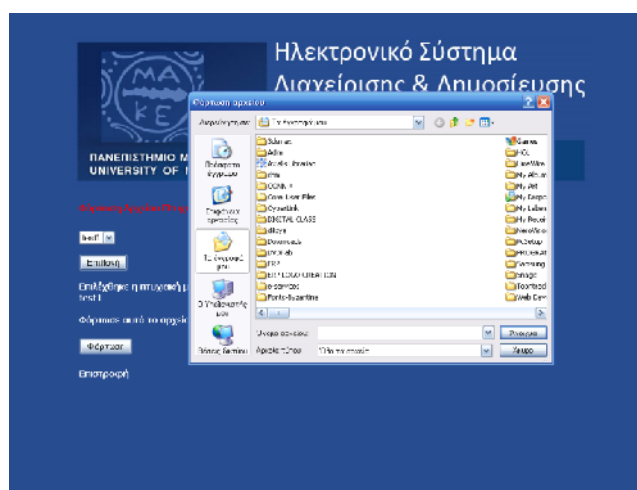


Εικόνα A.3.27

Ο καθηγητής κάνει αναζήτηση και επιλέγει το αρχείο που θέλει να φορτώσει (βλέπε εικόνα A.3.28). Στη συνέχεια κάνει κλικ στο κουμπί “Φόρτωση...”.

Τα αρχεία που μπορεί να φορτώνει, θα πρέπει :

- να μην είναι μηδενικά,
- να είναι μικρότερα από 40Mb
- και να είναι τύπου doc, pdf, zip ή rar.



Εικόνα A.3.28

### A.3.2.3.3 Διαγραφή αρχείου πτυχιακής

Επιλέγοντας “Διαγραφή αρχείου πτυχιακής”, εμφανίζεται μια μπάρα επιλογής τίτλου ολοκληρωμένων πτυχιακών (βλέπε εικόνα A.3.29) οι οποίες έχουν φορτωμένο αρχείο.



**Εικόνα A.3.29**

Στη συνέχεια το αρχείο της πτυχιακής που επέλεξε ο καθηγητής διαγράφεται και εμφανίζεται ανάλογο μήνυμα.

## A.4 Οδηγίες χρήσης (διαχειριστής)

### A.4.1 Διαχείριση usernames

Ο καθηγητής που αναλαμβάνει το κομμάτι της διαχείρισης έχει μια επιπλέον επιλογή κατά την είσοδο του η οποία είναι η “διαχείριση χρηστών” (βλέπε εικόνα A.3.30). Με αυτή την επιλογή ενημερώνει στην ουσία το σύστημα για το ποιοι καθηγητές είναι ενεργοί και ποιοι δεν είναι και αποτρέπει την είσοδο των φοιτητών, ως καθηγητές στο σύστημα.



## Σύγκριση scripting γλωσσών προγραμματισμού και διερεύνηση της καταλληλότητας τους για εργασίες συντήρησης λογισμικού



Εικόνα A.3.30

Επιλέγοντας “Διαχείριση usernames”, μπορεί να γίνει δημιουργία νέου χρήστη και να τροποποιηθούν ή να διαγραφούν οι ήδη υπάρχοντες (βλέπε εικόνα A.3.31).



Εικόνα A.3.31

## Βιβλιογραφία

- McAnally, Jeremy; Arkin, Assaf (March 28, 2009), *Ruby in Practice* (Πρώτη έκδοση), [Manning Publications](http://www.manning.com/mcanally/), σελ. 360, ISBN 1933988479, <http://www.manning.com/mcanally/>
- Thomas, Dave; Fowler, Chad; Hunt, Andy (April 28, 2009), *Programming Ruby 1.9: The Pragmatic Programmers' Guide* (Τρίτη έκδοση), [Pragmatic Bookshelf](http://pragprog.com/titles/ruby3/programming-ruby-1-9), σελ. 1000, ISBN 1934356085, <http://pragprog.com/titles/ruby3/programming-ruby-1-9>
- Flanagan, David; Matsumoto, Yukihiro (January 25, 2008), *The Ruby Programming Language* (Πρώτη έκδοση), [O'Reilly Media](http://oreilly.com/catalog/9780596516178/), σελ. 446, ISBN 0596516177, <http://oreilly.com/catalog/9780596516178/>
- Baird, Kevin (June 8, 2007), *Ruby by Example: Concepts and Code* (Πρώτη έκδοση), [No Starch Press](http://nostarch.com/ruby.htm), σελ. 326, ISBN 1593271484, <http://nostarch.com/ruby.htm>
- Fitzgerald, Michael (May 14, 2007), *Learning Ruby* (Πρώτη έκδοση), [O'Reilly Media](http://oreilly.com/catalog/9780596529864), σελ. 255, ISBN 0596529864, <http://oreilly.com/catalog/9780596529864>
- Cooper, Peter (March 26, 2007), *Beginning Ruby: From Novice to Professional* (Πρώτη έκδοση), [Apress](http://apress.com/book/view/9781590597668), σελ. 664, ISBN 1590597664, <http://apress.com/book/view/9781590597668>
- Fulton, Hal (November 4, 2006), *The Ruby Way* (Πρώτη έκδοση), [Addison-Wesley Professional](http://www.informit.com/store/product.aspx?isbn=0672328844), σελ. 888, ISBN 0596523696, <http://www.informit.com/store/product.aspx?isbn=0672328844>
- Carlson, Lucas; Richardson, Leonard (July 19, 2006), *Ruby Cookbook* (Πρώτη έκδοση), [O'Reilly Media](http://oreilly.com/catalog/9780596523695/), σελ. 906, ISBN 0596523696, <http://oreilly.com/catalog/9780596523695/>
- Ruby, Sam; Thomas, Dave; Hansson, David (March 28, 2009), *Agile Web Development with Rails* (Third έκδοση), [Pragmatic Bookshelf](http://www.pragprog.com/titles/rails3/agile-web-development-with-rails-third-edition), σελ. 850, ISBN 1934356166, <http://www.pragprog.com/titles/rails3/agile-web-development-with-rails-third-edition>
- Laurent, Simon St.; Dumbill, Edd (November 28, 2008), *Learning Rails* (First έκδοση), [O'Reilly Media](http://oreilly.com/catalog/9780596518776/), σελ. 442, ISBN 0596518773, <http://oreilly.com/catalog/9780596518776/>
- Lenz, Patrick (May 1, 2008), *Simply Rails 2* (Second έκδοση), [SitePoint](http://www.sitepoint.com/books/rails2/), σελ. 450, ISBN 0980455200, <http://www.sitepoint.com/books/rails2/>

Tate, Bruce; Hibbs, Curt (August 22, 2006), [\*Ruby on Rails: Up and Running\*](#) (First έκδοση), [O'Reilly Media](#), σελ. 182, [ISBN 0596101325](#),  
<http://oreilly.com/catalog/9780596101329/>

Holzner Ph.D., Steve (November 29, 2006), [\*Beginning Ruby on Rails\*](#) (First έκδοση), [Wrox](#), σελ. 380, [ISBN 0470069155](#),  
<http://www.wrox.com/WileyCDA/WroxTitle/Beginning-Ruby-on-Rails.productCd-0470069155.html>

Allan Hardy, Jeffrey; Carneiro Jr, Cloves; Catlin, Hampton (July 20, 2007), [\*Beginning Ruby on Rails E-Commerce: From Novice to Professional\*](#) (First έκδοση), [Wrox](#), σελ. 361, [ISBN 1590596862](#),  
<http://www.apress.com/book/view/9781590597361>

Clark, Mike (May 15, 2008), [\*Advanced Rails Recipes\*](#) (First έκδοση), Pragmatic Bookshelf, σελ. 464, [ISBN 0978739221](#),  
[http://www.pragprog.com/titles/fr\\_arr/advanced-rails-recipes](http://www.pragprog.com/titles/fr_arr/advanced-rails-recipes)

“Apache The Definitive Guide ( Second Edition )”, Ben Laurie and Peter Laurie, O'REILLY 1999.

“Apache Server Unleashed”, *Rich Bowen, Ken Coar*, 2000 by Sams Publishing.

“Apache Server 2 Bible”, Mohammed J. Kabir, Hungry Minds, Inc. 2002.

“Apache Server”, Rich Bowen, Ken Coar, et al., Sams Publishing 2000.

“MySQL Cookbook”, Paul DuBois, O'REILLY 2002.

“MySQL and Perl for the Web”, Paul DuBois, New Riders 2003.

“MySQL Enterprise Solutions”, Alexander Sasha Pachev, Wiley 2003.

“MySQL Reference Manual”, MySQL AB, MySQL AB 1997-2001.

“MySQL/PHP Database Applications, Jay Greenspan and Brand Bulger, M&T Books 2001.

“MySQL Reference Manual”, Copyright 1997-2001 MySQL AB.

“SQL Language Reference”, Pervasive Software 1998.

“SQL Performance Tuning”, Peter Gulutzan, Trudy Pelzer, Addison Wesley 2002.

“SQL Reference ( Release 3 )”, Diana Lorentz, Oracle Corporation 2000.

“Teach Your Self SQL in 21 Days ( Second Edition )”, Ryan K. Stephens, Ronald

R. Plew, Bryan Morgan, Jeff Perkins, Macmillan Computer Publishing 2000.

" Sam's Teach Yourself MySQL in 21 Days", Mark Maslakowski , Sams 2000.

"Managing and Using MySQL, 2nd Edition", George Reese\_Tim King\_Hugh E. Williams , O'Reilly Second Edition April 2002.

"Transact-SQL Cookbook", Jonathan Gennick, Ales Spetic,\_O'REILLY 2002.

"Database Management Systems, First Edition ( Συστήματα Διαχείρισης Βάσεων Δεδομένων ( Τόμος Α ) )", Raghuram Ramakrishnan, Johannes Gehrke ( Μετάφραση : Δημήτριος Δέρβος, Γεώργιος Ευαγγελίδης ), McGraw-Hill Companies 2000 ( Εκδόσεις ΤΖΙΟΛΑ 2002 ).

"Database Management Systems, Second Edition ( Συστήματα Διαχείρισης Βάσεων Δεδομένων ( Τόμος Α ) )", Raghuram Ramakrishnan, Johannes Gehrke ( Μετάφραση : Δημήτριος Δέρβος, Γεώργιος Ευαγγελίδης ), McGraw-Hill Companies 2000 ( Εκδόσεις ΤΖΙΟΛΑ 2002 ).

"Ανάπτυξη Web Εφαρμογών με PHP και MySQL", Luke Welling και Laura Thomson, Εκδόσεις Μ. ΓΚΙΟΥΡΔΑΣ

"Μάθετε PHP, MySQL και Apache σε 24 ώρες", Julie C. Meloni, Εκδόσεις Μ. ΓΚΙΟΥΡΔΑΣ

"PHP Οδηγός Προγραμματισμού", Smievski Hughes, Εκδόσεις Μ. ΓΚΙΟΥΡΔΑΣ

## Ιστοσελίδες

- [1] μ scripting PHP, <http://www.php.net>, (Ανακτήθηκε 01-05-2010)
- [2] Επίσημη σελίδα του Apache Server, <http://www.apache.org>, (Ανακτήθηκε 05-05-2010)
- [3] Επίσημη σελίδα της MySQL, <http://www.mysql.com>, (Ανακτήθηκε 07-05-2010)
- [4] MySQL, <http://www.sqlcourse.com>, (Ανακτήθηκε 07-05-2010)
- [5] Άρθρα για αρχάριους στη Ruby, [http://www.imargar.gr/index.php?option=com\\_content&view=category&id=34:--ruby&Itemid=53&layout=default](http://www.imargar.gr/index.php?option=com_content&view=category&id=34:--ruby&Itemid=53&layout=default), (Ανακτήθηκε 10-07-2010)
- [6] Rubystes: Σελίδα για Έλληνες προγραμματιστές της Ruby, <http://rubyst.es/>, (Ανακτήθηκε 10-07-2010)
- [7] Η επίσημη σελίδα της γλώσσας Ruby (Αγγλικά), <http://www.ruby-lang.org/en/>, (Ανακτήθηκε 17-07-2010)
- [8] The Great Ruby Shootout (December 2008): Συγκρίσεις μεταξύ υλοποιήσεων της Ruby. (Αγγλικά), <http://antoniochangiano.com/2008/12/09/the-great-ruby-shootout-december-2008/>, (Ανακτήθηκε 18-07-2010)
- [9] Collingbourne, Huw (June 17 2006), The Little Book Of Ruby, ελεύθερο eBook σε μορφή PDF 1.1MB (Αγγλικά), <http://www.sapphiresteel.com/The-Little-Book-Of-Ruby>, (Ανακτήθηκε 20-07-2010)
- [10] Collingbourne, Huw (April 18 2009), The Book Of Ruby, ελεύθερο eBook σε μορφή PDF 2.9MB (Αγγλικά), <http://www.sapphiresteel.com/the-book-of-ruby>, (Ανακτήθηκε 20-07-2010)
- [11] απλό εγχειρίδιο της Ruby με πολλά παραδείγματα (Αγγλικά) <http://ruby.on-page.net>, (Ανακτήθηκε 22-07-2010)
- [12] Ruby User Guide Από το Matz, δημιουργό της Ruby. Μετάφραση στα Αγγλικά. (Αγγλικά), <http://www.rubyist.net/~slagell/ruby/index.html>, (Ανακτήθηκε 23-07-2010)
- [13] Η Ruby από την πλευρά άλλων γλωσσών (Αγγλικά), <http://www.ruby-lang.org/en/documentation/ruby-from-other-languages/>, (Ανακτήθηκε 25-07-2010)
- [14] Community Filtered Ruby News (Αγγλικά), <http://www.rubyflow.com/>, (Ανακτήθηκε 27-07-2010)
- [15] Νέα της Ruby (Αγγλικά), <http://rubyreflector.com/>, (Ανακτήθηκε 28-07-2010)
- [16] Ελληνικό Ruby on Rails Βίκι, <http://wiki.rubyonrails.org/el/start>, (Ανακτήθηκε 30-07-2010)
- [17] Επίσημη σελίδα του Ruby on Rails (Αγγλικά) <http://www.rubyonrails.com/>, (Ανακτήθηκε 30-07-2010)

- [18] RubyForge.org (Αγγλικά), <http://rubyforge.org/> ,(Ανακτήθηκε 02-08-2010)
- [19] Εισαγωγή στο Ruby on Rails (Ελληνικά)  
[http://nikosd.com/files/Ruby On Rails Introduction v1.0.pdf](http://nikosd.com/files/Ruby%20On%20Rails%20Introduction%20v1.0.pdf)  
,(Ανακτήθηκε 09-08-2010)
- [20] μ scripts μ μ ,  
<http://www.hotscripts.com> ,(Ανακτήθηκε 11-05-2010)
- [21] Σύγκριση Γλωσσών Προγραμματισμού,  
<http://shootout.alioth.debian.org/gp4/benchmark.php?test=all&lang=php&lang2=yarv> ,(Ανακτήθηκε 20-10-2010)
- [22] Σύγκριση Ruby με PHP  
<http://www.feld.com/wp/archives/2006/10/php-vs-ruby.html> ,(Ανακτήθηκε 10-01-2011)