

Ανάπτυξη Εφαρμογής Λογισμικού για Διαχείριση Διαδικασιών Μαθημάτων Εκπαιδευτηρίου

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Από

ΣΑΜΑΡΑΣ ΟΔΥΣΣΕΑΣ

Επίβλεψη: Μάντακας Μάριος, Επίκουρος Καθηγητής

Άρτα 2015

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Τόπος, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

- 1) Μάντακας Μάριος
- 2) Αντωνιάδης Νικόλαος
- 3) Λιαροκάπης Δημήτριος

Ο Προϊστάμενος του Τμήματος

Περιεχόμενα

Εισαγωγή	5
Απαιτήσεις Λογισμικού	6
Σκοπός και Εύρος του Λογισμικού	6
Κατηγορίες Χρηστών	7
Λειτουργικές Απαιτήσεις	10
Μη Λειτουργικές Απαιτήσεις	16
Απαιτήσεις Ασφαλείας	16
Πλατφόρμα Ανάπτυξης Εφαρμογής	17
Απαιτήσεις Λογισμικού Χρήστη	17
Λεξικό Δεδομένων	18
Ανάλυση και Σχεδιασμός	20
Λογικός Σχεδιασμός Βάσης Δεδομένων	20
Χρήστες	21
Διδακτική Περίοδος και Μάθημα	22
Μάθημα_Περίοδος και Τμήμα	22
Μάθημα_Περίοδος και Συνάντηση	22
Καθηγητής και Μάθημα_Περίοδος	22
Φοιτητής και Μάθημα_Περίοδος	23
Φοιτητής και Τμήμα	23
Παρακολούθηση	23
Περιπτώσεις Χρήσης	25
Πίνακας Περιπτώσεων Χρήσης	25
Αναλυτικές Περιπτώσεις Χρήσης	27
Ανάλυση και Σχεδιασμός ανά Περίπτωση Χρήσης	31
Εγγραφή Χρήστη	31
Επισκόπηση Παρουσιών	35
Διεπαφή Χρήστη	38
Πίνακας Ιστοσελίδων που υλοποιήθηκαν	38
Αναλυτικός σχεδιασμός σελίδων	40
Υλοποίηση	45
Σχήμα της Βάσης Δεδομένων	45
Υλοποίηση Διεπαφής Χρήστη	49

Αποθηκευμένος Κώδικας στο Σχήμα της Βάσης Δεδομένων.....	55
Κώδικας σε Σελίδες	63
Υλοποίηση Ασφάλειας.....	65
Αυθεντικοποίηση και Εξουσιοδότηση.....	65
Κρυπτογράφηση Κωδικών Σύνδεσης	66
Λειτουργικές Δοκιμές.....	67
Δοκιμές Εγκυρότητας Δεδομένων.....	67
Δοκιμές Ασφάλειας	68
Αναφορές	69
Παράρτημα Α. – Κώδικας SQL-DDL δημιουργίας Σχήματος Βάσης Δεδομένων.....	71
Παράρτημα Β. – Κώδικας Πακέτων του Σχήματος Βάσης Δεδομένων	84

Εισαγωγή

Η παρούσα πτυχιακή εργασία αφορά την ανάπτυξη τμήματος εφαρμογής λογισμικού για διαχείριση διαδικασιών μαθημάτων εκπαιδευτηρίου. Η πτυχιακή εργασία περιλαμβάνει την παρούσα έκθεση και σχετικό λογισμικό επίδειξης.

Η ανάπτυξη της εφαρμογής λογισμικού έγινε με βάση των αρχών ευέλικτης επαναληπτικής μεθοδολογίας, αν και στο παρόν κείμενο εκτίθεται μόνο το τελικό αποτέλεσμα, και καλύπτει θέματα δημιουργίας απαιτήσεων λογισμικού, ανάλυσης και σχεδιασμού, υλοποίησης, εγκατάστασης και δοκιμών.

Η υλοποίηση έγινε με χρήση της πλατφόρμας ταχείας ανάπτυξης λογισμικού Oracle Application Express, της βάσης δεδομένων Oracle, και της γλώσσας προγραμματισμού Oracle PL/SQL. Συμπληρωματικά, χρησιμοποιήθηκε το περιβάλλον Oracle SQL-Developer για τον προγραμματισμό και την αντίστροφη σχεδίαση διαγραμμάτων βάσης δεδομένων από κώδικα SQL.

Μεγάλο μέρος της πτυχιακής εργασίας αφορούσε την εκμάθηση αυτής της πλατφόρμας με χρήση διδακτικών κειμένων, βιβλίων, του Oracle Forum και άλλων πηγών από τον παγκόσμιο ιστό, και τη μελέτη θεμάτων απαιτήσεων λογισμικού, ανάλυσης, σχεδιασμού και δοκιμών.

Απαιτήσεις Λογισμικού

Σκοπός και Εύρος του Λογισμικού

Σκοπός του λογισμικού είναι η αυτοματοποίηση της ανάθεσης των φοιτητών σε τμήματα εργαστηρίου των μαθημάτων ενός εκπαιδευτηρίου, καθώς και η αυτοματοποίηση καταχώρησης και καταγραφής παρουσιών στα εργαστήρια αυτά από τους καθηγητές.

Επιτρέπει στη γραμματεία του εκπαιδευτηρίου να ορίζει διδακτικές περιόδους, μαθήματα ανά διδακτική περίοδο, να επικυρώνει τους λογαριασμούς των καθηγητών και να αναθέτει μαθήματα στους καθηγητές.

Επιτρέπει στους καθηγητές του εκπαιδευτηρίου να εγγραφούν μόνοι τους, να ορίζουν μαθήματα, να ορίζουν τμήματα για τα μαθήματα που τους έχουν ανατεθεί από τη γραμματεία, να αναθέτουν φοιτητές στα τμήματα που έχουν δημιουργήσει και να επεξεργάζονται τις αναθέσεις. Επίσης, για την καταχώρηση των παρουσιών, επιτρέπει στους καθηγητές να επεξεργάζονται τις συναντήσεις των μαθημάτων, να δημιουργούν νέες και να καταχωρούν τις παρουσίες των φοιτητών σε κάθε συνάντηση.

Επιτρέπει στους φοιτητές να εγγράφονται μόνοι τους, να μπορούν να επιλέξουν μαθήματα από τα διαθέσιμα της διδακτικής περιόδου και να δηλώσουν προτίμηση για τα τμήματα των μαθημάτων που έχουν επιλέξει. Επίσης, επιτρέπει στους φοιτητές να έχουν επισκόπηση των παρουσιών για τα μαθήματα που έχουν παρακολουθήσει.

Έχει τη μορφή εφαρμογής διαδικτύου, στην οποία οι χρήστες έχουν πρόσβαση μέσω προγράμματος περιήγησης διαδικτύου και η οποία επικοινωνεί με σύστημα διαχείρισης βάσης δεδομένων για την αποθήκευση των δεδομένων.

Μπορεί να χρησιμοποιηθεί αυτούσιο είτε να είναι κομμάτι του συνολικού πληροφοριακού συστήματος ενός εκπαιδευτηρίου. Παραδείγματος χάρη μπορεί να υλοποιηθεί διεπαφή με το ήδη υπάρχων σύστημα εγγραφής των φοιτητών για τη δημιουργία των λογαριασμών και διεπαφή με το σύστημα βαθμολογίας, στο οποίο μπορεί να παρέχει δεδομένα σχετικά με τις παρουσίες των φοιτητών στα μαθήματα.

Πλεονεκτήματα που προσφέρει το λογισμικό σε σχέση με την ισχύουσα διαδικασία, είναι η εξοικονόμηση χρόνου για το διδακτικό προσωπικό, η ταχύτερη και ομαλότερη διαδικασία εγγραφής στα εργαστήρια για τους φοιτητές, καθώς και μεγαλύτερη ασφάλεια και αξιοπιστία στην διαδικασία καταγραφής παρουσιών.

Κατηγορίες Χρηστών

Στο λογισμικό θα υπάρχουν οι παρακάτω κατηγορίες χρηστών:

	Συντομογραφία
Χρήστης	χ
Φοιτητής	φ
Καθηγητής	κ
Γραμματεία	γ

Οι 3 τελευταίοι είναι υποκλάσεις του πρώτου.

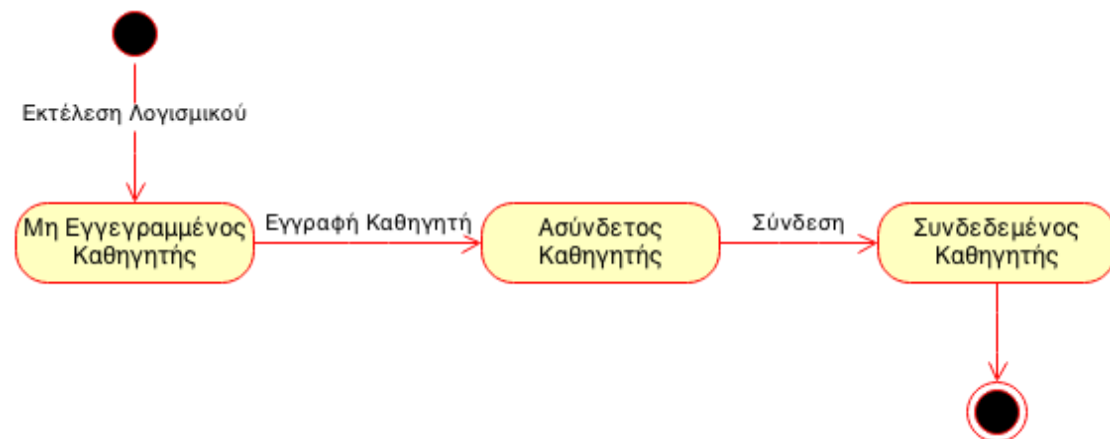
Επί πλέον συντομογραφία

κφ = κ ή φ

Καταστάσεις χρηστών:

Κατάσταση		Χρήστης	Γραμματεία	Καθηγητής	Φοιτητής
Μη εγγεγραμμένος				X	X
Εγγεγραμμένος				X	X
Ασύνδετος		X	X	X	X
Συνδεδεμένος		X	X	X	X
Μη Επικυρωμένος		X		X	X
Επικυρωμένος		X		X	X

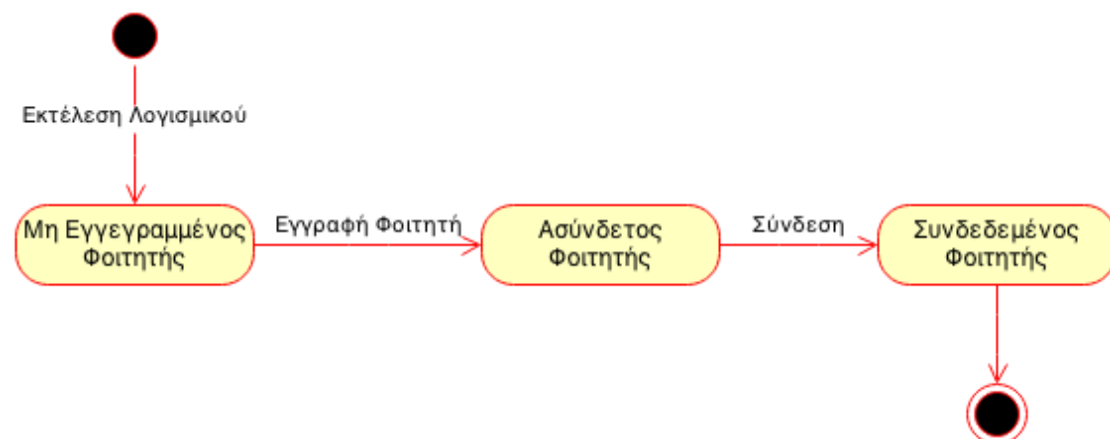
Η εικόνα 1 δείχνει το διάγραμμα καταστάσεων UML με τη διαδικασία με την οποία γίνεται η εναλλαγή καταστάσεων του χρήστη Καθηγητή.



Εικόνα 1 - Εναλλαγή καταστάσεων Καθηγητή

Αρχικά ο χρήστης Καθηγητής, όταν εκτελέσει το λογισμικό, θεωρείται Μη Εγγεγραμμένος. Αφού ολοκληρώσει τη λειτουργία της Εγγραφής Καθηγητή, θεωρείται Ασύνδετος Καθηγητής και αφού ολοκληρώσει τη λειτουργία της Σύνδεσης, θεωρείται Συνδεδεμένος Καθηγητής.

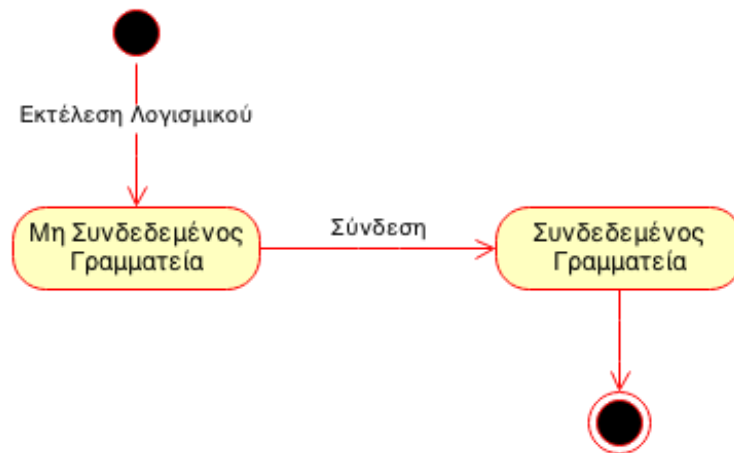
Η εικόνα 2 δείχνει το διάγραμμα καταστάσεων UML με την αντίστοιχη διαδικασία για το χρήστη Φοιτητή.



Εικόνα 2 - Εναλλαγή καταστάσεων Φοιτητή

Αρχικά ο χρήστης Φοιτητής, όταν εκτελέσει το λογισμικό, θεωρείται Μη Εγγεγραμμένος. Αφού ολοκληρώσει τη λειτουργία της Εγγραφής Φοιτητή, θεωρείται Ασύνδετος Φοιτητής και αφού ολοκληρώσει τη λειτουργία της Σύνδεσης, θεωρείται Συνδεδεμένος Φοιτητής.

Ο χρήστης Γραμματεία προϋπάρχει και δεν εγγράφεται, οπότε η διαδικασία εναλλαγής καταστάσεων (εικόνα 3) είναι η εξής:



Εικόνα 3 - Εναλλαγή καταστάσεων Γραμματεία

Αρχικά ο χρήστης Γραμματεία, όταν εκτελέσει το λογισμικό, θεωρείται Μη Συνδεδεμένος Γραμματεία. Αφού ολοκληρώσει τη λειτουργία Σύνδεση, θεωρείται Συνδεδεμένος Γραμματεία.

Λειτουργικές Απαιτήσεις

Οι λειτουργικές απαιτήσεις είναι ιεραρχικά ταξινομημένες. Η ταξινόμηση γίνεται με βάση κατ' αρχάς το χρήστη, και στη συνέχεια τη λειτουργία, (λειτουργία, υπολειτουργία κλπ.). Η αρίθμηση των απαιτήσεων γίνεται με λεκτικά που αντανakλούν αυτήν την ταξινόμηση, π.χ. η υπολειτουργία εγγραφή, της λειτουργίας αυθεντικοποίησης, την οποία χρησιμοποιούν ο Φοιτητής και ο Καθηγητής, αριθμείται ως εξής:

κφ.αυθεντικοποίηση.εγγραφή

ΠΝΟ : πρέπει να οριστεί (η απαίτηση δεν είναι ακόμα πλήρης).

Οι όροι ορίζονται στο Λεξικό Δεδομένων.

Αριθμός	Απαίτηση	Σπουδαιότητα	Έκδοση
κφ.αυθεντικοποίηση.εγγραφή	Το λογισμικό θα επιτρέπει στο Φοιτητή και στον Καθηγητή να εγγράφονται στο σύστημα, εισάγοντας τα δεδομένα εγγραφής.	Απαραίτητη	1.0
κφ.αυθεντικοποίηση.επικύρωση	Το λογισμικό θα επιτρέπει στο Φοιτητή και στον Καθηγητή να επικυρώσει την εγγραφή του, ακολουθώντας σύνδεσμο επικύρωσης ο οποίος θα αποστέλλεται στο email εγγραφής.	Απαραίτητη	1.0
χ.αυθεντικοποίηση.σύνδεση	Το λογισμικό θα επιτρέπει στο Χρήστη να συνδεθεί στην εφαρμογή, εισάγοντας τα δεδομένα σύνδεσης.	Απαραίτητη	1.0
χ.αυθεντικοποίηση.αποσύνδεση	Το λογισμικό θα επιτρέπει στο Χρήστη να αποσυνδεθεί από την εφαρμογή.	Απαραίτητη	1.0
κφ.αυθεντικοποίηση.ανάκτηση	Το λογισμικό θα επιτρέπει στο Φοιτητή και στον Καθηγητή, να ανακτήσουν των	Απαραίτητη	1.0

Αριθμός	Απαίτηση	Σπουδαιότητα	Έκδοση
	κωδικό σύνδεσης τους, εισάγοντας το email εγγραφής.		
χ.αυθεντικοποίηση.αλλαγή.δεδομένα	Το λογισμικό θα επιτρέπει στο Χρήστη, να αλλάξει τα προσωπικά δεδομένα του.	Απαραίτητη	1.0
χ.αυθεντικοποίηση.αλλαγή.κωδικός	Το λογισμικό θα επιτρέπει στο Χρήστη, να αλλάξει τον κωδικό σύνδεσης του.	Απαραίτητη	1.0
γ.περίοδοι.δημιουργία	Το λογισμικό θα επιτρέπει στη Γραμματεία να δημιουργήσει διδακτικές περιόδους, δίνοντας τα δεδομένα ορισμού διδακτικής περιόδου.	Απαραίτητη	1.0
γ.περίοδοι.μάθημα.προσθήκη	Το λογισμικό θα επιτρέπει στη Γραμματεία να προσθέσει μαθήματα σε κάθε διδακτική περίοδο. Κατά τη διαδικασία αυτή το λογισμικό θα δημιουργεί αυτόματα συναντήσεις για το μάθημα στη διδακτική περίοδο, με βάση τον αριθμό των εβδομάδων της διδακτικής περιόδου.	Απαραίτητη	1.0
γ.περίοδοι.μάθημα.αφαίρεση	Το λογισμικό θα επιτρέπει στη Γραμματεία να αφαιρέσει μαθήματα από μία διδακτική περίοδο.	Απαραίτητη	1.0
γ.μαθήματα.δημιουργία	Το λογισμικό θα επιτρέπει στη Γραμματεία να δημιουργήσει μαθήματα, ορίζοντας το όνομα του μαθήματος.	Απαραίτητη	1.0
γ.μαθήματα.διαγραφή	Το λογισμικό θα επιτρέπει στη Γραμματεία να διαγράψει μαθήματα.	Απαραίτητη	1.0
γ.μαθήματα.ανάθεση	Το λογισμικό θα επιτρέπει στη Γραμματεία να αναθέσει μαθήματα σε Καθηγητές ανά διδακτική περίοδο.	Απαραίτητη	1.0
γ.μαθήματα.ανάθεση.κατάργηση	Το λογισμικό θα επιτρέπει στη Γραμματεία να καταργήσει αναθέσεις μαθημάτων	Απαραίτητη	1.0

Αριθμός	Απαίτηση	Σπουδαιότητα	Έκδοση
	σε Καθηγητές ανά διδακτική περίοδο.		
γ.καθηγητές.επικύρωση	Το λογισμικό θα επιτρέπει στη Γραμματεία να επικυρώσει τις εγγραφές των Καθηγητών.	Απαραίτητη	1.0
γ.φοιτητές.εισαγωγή	Το λογισμικό θα επιτρέπει στη Γραμματεία να εισάγει δεδομένα Φοιτητών από αρχείο Excel.	Επιθυμητή	2.0
κ.μαθήματα.επισκόπηση	Το λογισμικό θα επιτρέπει στον Καθηγητή να έχει επισκόπηση των μαθημάτων που του έχουν ανατεθεί ανά διδακτική περίοδο.	Απαραίτητη	1.0
κ.μαθήματα.δημιουργία	Το λογισμικό θα επιτρέπει στον Καθηγητή να δημιουργήσει μαθήματα ανά διδακτική περίοδο, ορίζοντας το όνομα του μαθήματος. Το λογισμικό θα αναθέτει αυτόματα το μάθημα στον Καθηγητή που το δημιούργησε.	Απαραίτητη	1.0
κ.τμήματα.επισκόπηση	Το λογισμικό θα επιτρέπει στον Καθηγητή να έχει επισκόπηση των τμημάτων των μαθημάτων που του έχουν ανατεθεί ανά διδακτική περίοδο.	Απαραίτητη	1.0
κ.τμήματα.δημιουργία	Το λογισμικό θα επιτρέπει στον Καθηγητή να δημιουργήσει τμήματα για όλα τα μαθήματα που του έχουν ανατεθεί ανά διδακτική περίοδο, ορίζοντας τα δεδομένα ορισμού τμήματος.	Απαραίτητη	1.0
κ.τμήματα.ενημέρωση	Το λογισμικό θα επιτρέπει στον Καθηγητή να ενημερώσει τα δεδομένα ορισμού τμήματος, των τμημάτων των μαθημάτων που του έχουν ανατεθεί ανά διδακτική περίοδο.	Απαραίτητη	1.0

Αριθμός	Απαίτηση	Σπουδαιότητα	Έκδοση
κ.τμήματα.διαγραφή	Το λογισμικό θα επιτρέπει στον Καθηγητή να διαγράψει τμήματα των μαθημάτων που του έχουν ανατεθεί ανά διδακτική περίοδο.	Απαραίτητη	1.0
κ.τμήματα.φοιτητές.ανάθεση	Το λογισμικό θα επιτρέπει στον Καθηγητή να αναθέσει φοιτητές σε τμήματα των μαθημάτων που του έχουν ανατεθεί ανά διδακτική περίοδο. Κατά τη διαδικασία αυτή θα επιτρέπει στον Καθηγητή να βλέπει τις δηλώσεις προτίμησης τμημάτων των Φοιτητών..	Απαραίτητη	1.0
κ.τμήματα.φοιτητές.ανάθεση.κατάργηση	Το λογισμικό θα επιτρέπει στον Καθηγητή να καταργήσει τις αναθέσεις των Φοιτητών σε τμήματα των μαθημάτων που του έχουν ανατεθεί ανά διδακτική περίοδο.	Απαραίτητη	1.0
κ.τμήματα.ανάθεση.αυτόματη	Το λογισμικό θα επιτρέπει στον Καθηγητή να αναθέσει τους Φοιτητές αυτόματα σε τμήματα των μαθημάτων που του έχουν ανατεθεί ανά διδακτική περίοδο.	Επιθυμητή	2.0
κ.τμήματα.συγχώνευση	Το λογισμικό θα επιτρέπει στον Καθηγητή να συγχωνεύσει τμήματα των μαθημάτων που του έχουν ανατεθεί ανά διδακτική περίοδο.	Επιθυμητή	2.0
κ.τμήματα.διαχωρισμός	Το λογισμικό θα επιτρέπει στον Καθηγητή να διαχωρίσει ένα τμήμα σε δύο τμήματα.	Επιθυμητή	2.0
κ.συναντήσεις.επισκόπηση	Το λογισμικό θα επιτρέπει στον Καθηγητή να έχει επισκόπηση των συναντήσεων των μαθημάτων που του έχουν ανατεθεί ανά διδακτική περίοδο.	Απαραίτητη	1.0
κ.συναντήσεις.δημιουργία	Το λογισμικό θα επιτρέπει στον Καθηγητή να δημιουργήσει συναντήσεις για τα μαθήματα που του έχουν ανατεθεί ανά διδακτική περίοδο, ορίζοντας τον προγραμματισμό της συνάντησης. Η αρίθμηση της συνάντησης θα ορίζεται	Απαραίτητη	1.0

Αριθμός	Απαίτηση	Σπουδαιότητα	Έκδοση
	αυτόματα, αυξανόμενη κατά ένα για κάθε συνάντηση σε κάθε μάθημα ανά διδακτική περίοδο.		
κ.συναντήσεις.ενημέρωση	Το λογισμικό θα επιτρέπει στον Καθηγητή να ενημερώσει την κατάσταση των συναντήσεων μαθήματος, για τα μαθήματα που του έχουν ανατεθεί ανά διδακτική περίοδο.	Απαραίτητη	1.0
κ.παρουσίες.καταχώρηση	Το λογισμικό θα επιτρέπει στον Καθηγητή να καταχωρήσει τις παρουσίες και το τμήμα παρακολούθησης των φοιτητών στις συναντήσεις ενός μαθήματος ανά διδακτική περίοδο, με προαιρετική επιλογή του τμήματος, και αναζήτηση. Η αναζήτηση θα γίνεται με τα εξής κριτήρια: • Όνομα • Επίθετο • Αριθμός Μητρώου	Απαραίτητη	1.0
κ.παρουσίες.μεταφόρτωση	Το λογισμικό θα επιτρέπει στον Καθηγητή να μεταφορτώσει την αναφορά των παρουσιών των φοιτητών, στις συναντήσεις των μαθημάτων ανά διδακτική περίοδο, σε αρχείο Excel.	Απαραίτητη	1.0
κ.παρουσίες.επισκόπηση	Το λογισμικό θα επιτρέπει στον Καθηγητή να έχει επισκόπηση των παρουσιών όλων των Φοιτητών σε κάθε συνάντηση μαθήματος ανά διδακτική περίοδο.	Απαραίτητη	1.0
κ.παρουσίες.επισκόπηση.μεταφόρτωση	Το λογισμικό θα επιτρέπει στον Καθηγητή να μεταφορτώσει την αναφορά των παρουσιών όλων των Φοιτητών σε κάθε συνάντηση μαθήματος ανά διδακτική περίοδο, σε αρχείο Excel.	Απαραίτητη	1.0

Αριθμός	Απαίτηση	Σπουδαιότητα	Έκδοση
κ.κατωχύρωση.κριτήρια	Το λογισμικό θα επιτρέπει στον Καθηγητή να ορίσει κριτήρια κατοχύρωσης για κάθε μάθημα ανά διδακτική περίοδο. Τα κριτήρια θα πρέπει να οριστούν (ΠΝΟ).	Επιθυμητή	2.0
κ.πρόγραμμα.ορισμός	Το λογισμικό θα επιτρέπει στον Καθηγητή να ορίσει εβδομαδιαίο ωρολόγιο πρόγραμμα, μαθήματα που δεν πραγματοποιούνται και μαθήματα αναπλήρωσης.	Επιθυμητή	2.0
κ.ενημέρωση.ανάθεση	Το λογισμικό θα αποστέλλει ενημερωτικό email στο Φοιτητή όταν ο Καθηγητής τον αναθέσει σε τμήμα μαθήματος.	Επιθυμητή	2.0
κ.ενημέρωση.κατωχύρωση	Το λογισμικό θα αποστέλλει ενημερωτικό email στο Φοιτητή, όταν ο Καθηγητής ορίσει την κατοχύρωση του σε μάθημα ανά διδακτική περίοδο.	Επιθυμητή	2.0
φ.μαθήματα.επιλογή	Το λογισμικό θα επιτρέπει στον Φοιτητή να επιλέξει μαθήματα ανά διδακτική περίοδο.	Απαραίτητη	1.0
φ.τμήματα.επιλογή	Το λογισμικό θα επιτρέπει στο Φοιτητή να δηλώσει προτίμηση για τα τμήματα των μαθημάτων που έχει δηλώσει ανά διδακτική περίοδο.	Απαραίτητη	1.0
φ.τμήματα.επιλογή.αδυναμία	Το λογισμικό θα επιτρέπει στο Φοιτητή, σε περίπτωση αδυναμίας παρακολούθησης τμήματος, να επιλέξει τον λόγο αδυναμίας και να εισάγει την εξήγηση.	Απαραίτητη	1.0
φ.παρουσίες.ιστορικό	Το λογισμικό θα επιτρέπει στο Φοιτητή να έχει επισκόπηση του ιστορικού των παρακολουθήσεων του, για όλες τις διδακτικές περιόδους και όλα τα μαθήματα τα οποία έχει παρακολουθήσει.	Απαραίτητη	1.0

Μη Λειτουργικές Απαιτήσεις

Απαιτήσεις Ασφαλείας

Αριθμός	Απαίτηση	Σπουδαιότητα	Έκδοση
α.κρυπτογράφηση.κωδικός	Το λογισμικό θα αποθηκεύει τους κωδικούς εγγραφής των χρηστών σε κρυπτογραφημένη μορφή, στη βάση δεδομένων.	Απαραίτητη	1.0
α.δεδομένα.χρήστη	Ένας χρήστης, συνδεδεμένος ή ασύνδετος, δε θα μπορεί να έχει πρόσβαση (ανάγνωση/εγγραφή) σε αντικείμενα (π.χ. σελίδες, πεδία κλπ.) άλλου συγκεκριμένου χρήστη.	Απαραίτητη	1.0
α.αυθεντικοποίηση.χρήστη	Ένας ασύνδετος χρήστης, δε θα μπορεί να έχει πρόσβαση (ανάγνωση/εγγραφή) σε αντικείμενα (π.χ. σελίδες, πεδία κλπ.) συνδεδεμένου χρήστη.	Απαραίτητη	1.0

Πλατφόρμα Ανάπτυξης Εφαρμογής

Το λογισμικό θα τρέχει σε λειτουργικό σύστημα Oracle Linux 6.5 ή συμβατό λειτουργικό σύστημα, με εγκατεστημένη τη βάση δεδομένων Oracle Database Express-Edition 11g Release 2 και το Oracle Application Express 4.2. Ο εξυπηρετητής της βάσης δεδομένων θα έχει πρόσβαση στο διαδίκτυο. Ο προγραμματισμός του λογισμικού θα γίνει στη γλώσσα προγραμματισμού Oracle PL/SQL.

Απαιτήσεις Λογισμικού Χρήστη

Για την πρόσβαση στο λογισμικό από χρήστη, θα απαιτείτε φυλλομετρητής διαδικτύου με ενεργοποιημένο JavaScript. Θα υποστηρίζονται οι παρακάτω εκδόσεις φυλλομετρητών διαδικτύου:

- Microsoft Internet Explorer 7.0 ή μεγαλύτερη έκδοση
- Mozilla Firefox 14 ή μεγαλύτερη έκδοση
- Google Chrome 21 ή μεγαλύτερη έκδοση
- Apple Safari 5.0 ή μεγαλύτερη έκδοση

Λεξικό Δεδομένων

δεδομένα εγγραφής κφ:

- όνομα σύνδεσης κφ,
- + κωδικός σύνδεσης κφ,
- + email κφ,
- + όνομα κφ,
- + επίθετο κφ,
- + αριθμός μητρώου φ

όνομα σύνδεσης κφ: αλφαριθμητικό με 6 χαρακτήρες τουλάχιστον, μοναδικό για κάθε χρήστη

κωδικός σύνδεσης κφ: αλφαριθμητικό με 6 χαρακτήρες τουλάχιστον

email κφ: έγκυρης μορφής email, μοναδικό για κάθε χρήστη

όνομα κφ: το όνομα του χρήστη

επίθετο κφ: το επίθετο του χρήστη

αριθμός μητρώου φ: αριθμός θετικός ακέραιος, μοναδικός για κάθε φοιτητή

δεδομένα σύνδεσης χ:

- όνομα σύνδεσης χ,
- + κωδικός σύνδεσης χ

προσωπικά δεδομένα κφ:

- email κφ,
- + όνομα κφ,
- + επίθετο κφ,
- + αριθμός μητρώου φ

σύνδεσμος επικύρωσης: σύνδεσμος που παραπέμπει στην εφαρμογή και επικυρώνει την εγγραφή χρήστη, μοναδικός για κάθε χρήστη

διδασκτική περίοδος: περίοδος ορισμένου χρόνου στην οποία διδάσκονται μαθήματα

δεδομένα ορισμού διδακτικής περιόδου:

- όνομα διδακτικής περιόδου,
- + ημερομηνία έναρξης,

+ ημερομηνία λήξης

μάθημα: μάθημα εκπαιδευτηρίου

όνομα μαθήματος: αλφαριθμητικό, μοναδικό για κάθε μάθημα

τμήμα μαθήματος: τμήμα ενός μαθήματος σε μία διδακτική περίοδο

δεδομένα ορισμού τμήματος:

όνομα τμήματος,

+ ημέρα,

+ ώρα έναρξης,

+ ώρα λήξης

συνάντηση μαθήματος: συνάντηση ενός μαθήματος σε μία διδακτική περίοδο

προγραμματισμός συνάντησης:

τακτικός,

|| έκτακτος

κατάσταση συνάντησης:

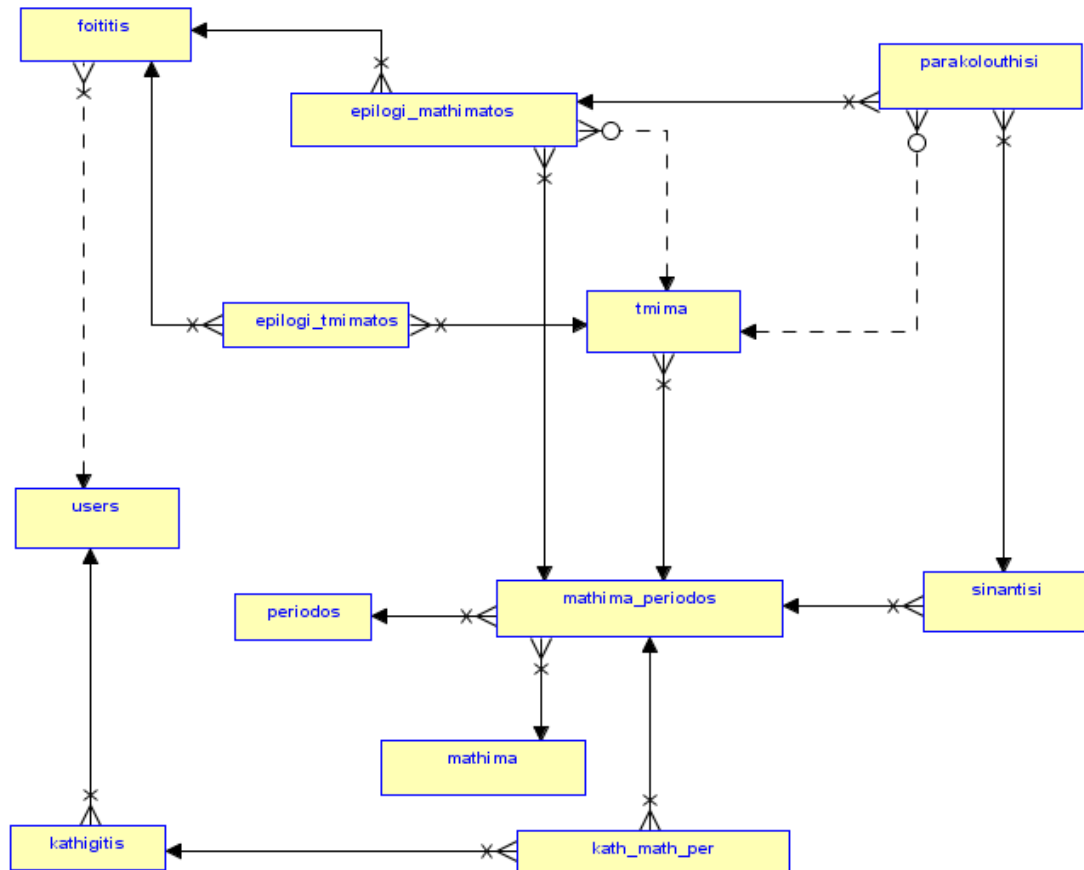
προγραμματισμένη,

|| πραγματοποιημένη,

|| ακυρωμένη

Ανάλυση και Σχεδιασμός

Λογικός Σχεδιασμός Βάσης Δεδομένων



Εικόνα 4 - Σχεσιακό μοντέλο βάσης δεδομένων

Ο λογικός σχεδιασμός της βάσης δεδομένων περιλάμβανε τη δημιουργία μοντέλου οντοτήτων-συσχετίσεων βάσει των απαιτήσεων λογισμικού με χρήση κανόνων λογικού σχεδιασμού, που διευκολύνουν τη δημιουργία σχήματος σε τουλάχιστον 3^η κανονική μορφή και τη μετατροπή του μοντέλου οντοτήτων-συσχετίσεων σε κώδικα SQL-DDL.

Παρακάτω, οι οντότητες αναφέρονται με κεφαλαίο αρχικό γράμμα ονόματος και οι πραγματώσεις τους με πεζό, π.χ. ένας φοιτητής της οντότητας Φοιτητής.

Χρήστες

Φοιτητής και Καθηγητής είναι οντότητες γιατί έχουν χαρακτηριστικά (απαίτηση: κφ.αυθεντικοποίηση.εγγραφή). Οι οντότητες μοιράζονται χαρακτηριστικά και θεωρήθηκαν αρχικά ειδικές οντότητες μιας γενικής οντότητας Χρήστης (σχέση γενίκευσης).

Δεδομένου ότι οι φοιτητές μπορούν να εγγραφούν μόνοι τους ως χρήστες του λογισμικού (απαίτηση: κφ.αυθεντικοποίηση.εγγραφή), αλλά και να εγγραφούν από τη Γραμματεία (απαίτηση: γ.φοιτητές.εισαγωγή) η αρχική αυτή θεώρηση τροποποιείται ως εξής:

Η οντότητα Φοιτητής αφορά όλους τους φοιτητές που εγγράφονται και με τους δύο τρόπους και συσχετίζεται 0..1 προς 0..1 με την οντότητα Χρήστης (ένας φοιτητής μπορεί να αντιστοιχεί σε έναν το πολύ χρήστη και ένας χρήστης μπορεί να αντιστοιχεί σε έναν το πολύ φοιτητή). Τα χαρακτηριστικά τα οποία εισάγει η Γραμματεία για κάθε φοιτητή είναι τα:

- Όνομα
- Επίθετο
- Αριθμός Μητρώου

Άρα αυτά είναι και τα χαρακτηριστικά της οντότητας Φοιτητής.

Στην οντότητα Χρήστης μένουν τα χαρακτηριστικά τα οποία έχουν μόνο οι φοιτητές που εγγράφηκαν μόνοι τους, τα οποία είναι κοινά με τους καθηγητές που επίσης εγγράφονται μόνοι τους και είναι τα:

- Όνομα Χρήστη
- Κωδικός
- Email

Αντίστοιχα η οντότητα Καθηγητής έχει τα χαρακτηριστικά:

- Όνομα
- Επίθετο
- Επικυρωμένος

Η οντότητα Καθηγητής συσχετίζεται 0..1 προς 1 με την οντότητα Χρήστης, διότι κάθε καθηγητής πρέπει να αντιστοιχεί σε χρήστη εφόσον οι καθηγητές γράφονται πάντα μόνοι τους. Το χαρακτηριστικό Επικυρωμένος χρησιμοποιείται για την επικύρωση των καθηγητών από τη γραμματεία (απαίτηση: γ.καθηγητές.επικύρωση)

Διδακτική Περίοδος και Μάθημα

Η Διδακτική Περίοδος και το Μάθημα είναι οντότητες γιατί έχουν χαρακτηριστικά. Η οντότητα Διδακτική Περίοδος έχει τα χαρακτηριστικά Όνομα, Ημερομηνία Έναρξης και Ημερομηνία Λήξης (απαίτηση: γ.περίοδοι.δημιουργία). Η οντότητα Μάθημα έχει το χαρακτηριστικό Όνομα (απαίτηση: γ.μαθήματα.δημιουργία, κ.μαθήματα.δημιουργία). Οι οντότητες μεταξύ τους συσχετίζονται 0..N προς 0..N, σε μία διδακτική περίοδο μπορεί να διδάσκονται πολλά μαθήματα και ένα μάθημα μπορεί να διδάσκεται σε πολλές διδακτικές περιόδους. Η συσχέτιση εκφράζεται με την οντότητα Μάθημα_Περίοδος.

Μάθημα_Περίοδος και Τμήμα

Το Τμήμα είναι οντότητα και έχει τα χαρακτηριστικά Όνομα, Ημέρα, Ώρα Έναρξης και Ώρα Λήξης (απαίτηση: κ.τμήματα.δημιουργία). Έχει σχέση με πληθικότητα 1 προς 0..N με την οντότητα Μάθημα_Περίοδος, ένα μάθημα σε μία περίοδο μπορεί να έχει πολλά τμήματα.

Μάθημα_Περίοδος και Συνάντηση

Η Συνάντηση είναι οντότητα και έχει τα χαρακτηριστικά Αρίθμηση, Προγραμματισμός, Κατάσταση (απαίτηση: κ.συναντήσεις.δημιουργία, κ.συναντήσεις.ενημέρωση). Έχει σχέση με πληθικότητα 1 προς 0..N με την οντότητα Μάθημα_Περίοδος, ένα μάθημα σε μία περίοδο μπορεί να έχει πολλές συναντήσεις.

Καθηγητής και Μάθημα_Περίοδος

Μεταξύ των οντοτήτων Καθηγητής Μάθημα και Περίοδος υπάρχει τριαδική σχέση πολλά-προς-πολλά-προς-πολλά, ένας καθηγητής μπορεί να διδάσκει πολλά μαθήματα σε πολλές διδακτικές περιόδους και ένα μάθημα μπορεί να διδάσκεται από πολλούς καθηγητές σε πολλές διδακτικές περιόδους. Οι οντότητες μεταξύ τους δεν έχουν συναρτησιακές εξαρτήσεις, οπότε για να εκφραστεί η σχέση τους χρειάζονται τα πρωτεύοντα κλειδιά και των τριών. Η τριαδική σχέση εκφράζεται με την οντότητα Καθηγητής_Μάθημα_Περίοδος η οποία έχει σχέση 0..N με την οντότητα Καθηγητής και 0..N με την οντότητα Μάθημα_Περίοδος (απαίτηση: γ.μαθήματα.ανάθεση). Χρησιμοποιήθηκαν δύο ενδιάμεσες οντότητας για την έκφραση της τριαδικής σχέσης, διότι η οντότητα Μάθημα_Περίοδος σχετίζεται και με τις οντότητες Τμήμα και Συνάντηση στις οποίες σχέσεις δεν χρειάζεται το πρωτεύον κλειδί της οντότητας Καθηγητής. Με αυτόν τον τρόπο δεν παραβιάζεται η δεύτερη κανονική μορφή του σχήματος της βάσης δεδομένων.

Φοιτητής και Μάθημα_Περίοδος

Μεταξύ των οντοτήτων Φοιτητής Μάθημα και Περίοδος υπάρχει τριαδική σχέση πολλά-προς-πολλά-προς-πολλά, ένας φοιτητής μπορεί να διαλέξει πολλά μαθήματα σε πολλές διδακτικές περιόδους και ένα μάθημα μπορεί να επιλεγεί από πολλούς φοιτητές σε πολλές διδακτικές περιόδους. Οι οντότητες μεταξύ τους δεν έχουν συναρτησιακές εξαρτήσεις, οπότε για να εκφραστεί η σχέση τους χρειάζονται τα πρωτεύοντα κλειδιά και των τριών. Η τριαδική σχέση εκφράζεται με την οντότητα Επιλογή_Μαθήματος η οποία έχει σχέση 0..N με την οντότητα Φοιτητής και 0..N με την οντότητα Μάθημα_Περίοδος (απαίτηση: φ.μαθήματα.επιλογή). Όπως και στην περίπτωση της τριαδικής σχέσης μεταξύ Καθηγητή, Μάθημα και Περίοδος, χρησιμοποιήθηκαν δύο ενδιαμέσες οντότητες για την έκφραση της τριαδικής σχέσης, διότι η οντότητα Μάθημα_Περίοδος σχετίζεται και με τις οντότητες Τμήμα και Συνάντηση στις οποίες σχέσεις δεν χρειάζεται το πρωτεύων κλειδί της οντότητας Φοιτητής. Με αυτόν τον τρόπο δεν παραβιάζεται η δεύτερη κανονική μορφή του σχήματος της βάσης δεδομένων.

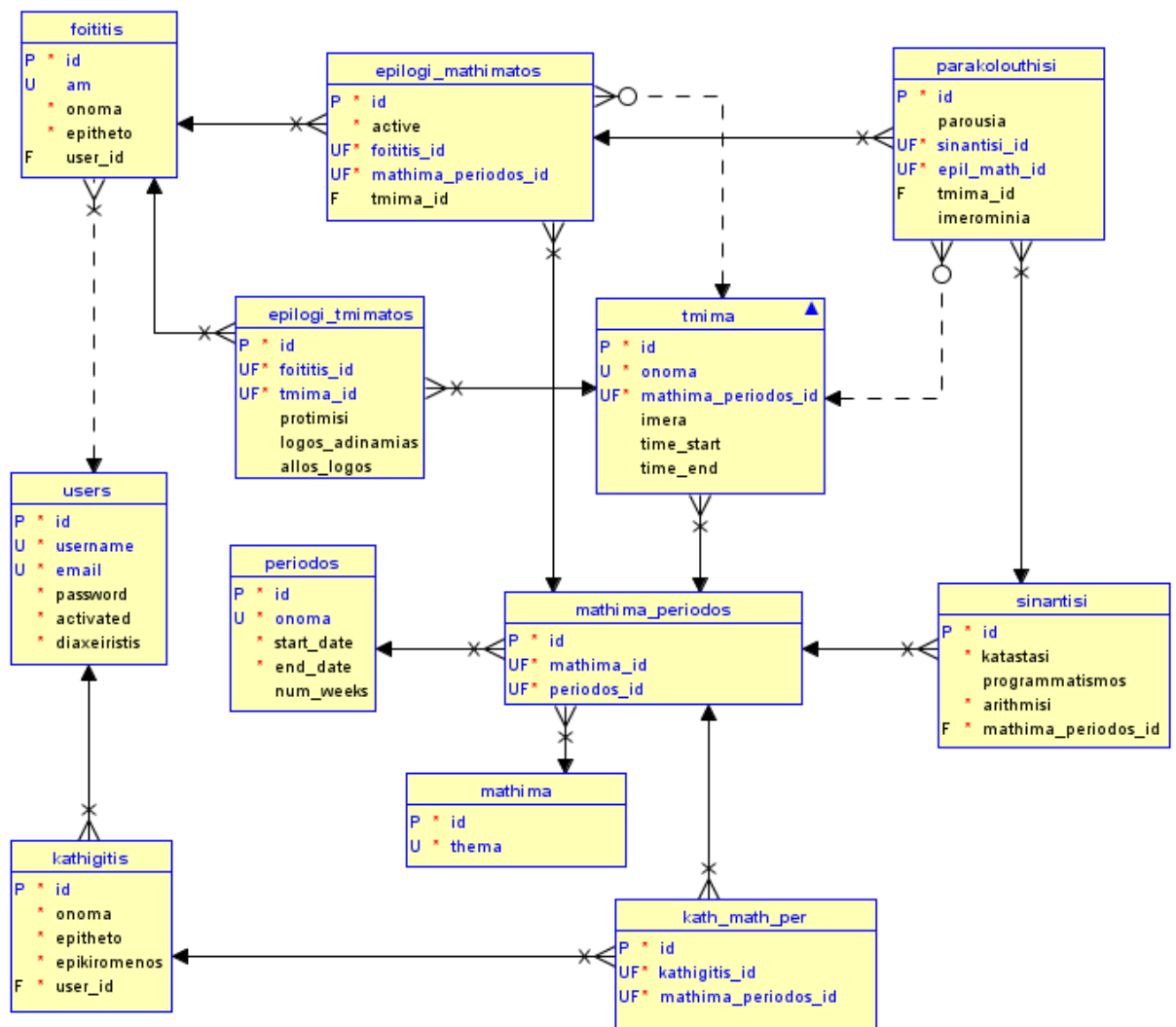
Φοιτητής και Τμήμα

Μεταξύ των οντοτήτων Φοιτητής και Τμήμα υπάρχει συσχέτιση 0..N προς 0..N, ένας φοιτητής μπορεί να δηλώσει προτίμηση για πολλά τμήματα, και ένα τμήμα μπορεί να έχει δηλωθεί από πολλούς φοιτητές (απαίτηση: φ.τμήματα.επιλογή). Η συσχέτιση εκφράζεται με την οντότητα Επιλογή_Τμήματος η οποία έχει σαν χαρακτηριστικά την Προτεραιότητα, τον Λόγο Αδυναμίας Παρακολούθησης και την Εξήγηση (απαίτηση: φ.τμήματα.επιλογή, φ.τμήματα.επιλογή.αδυναμία).

Επίσης υπάρχει δεύτερη συσχέτιση 0..N προς 0..N μεταξύ των οντοτήτων Φοιτητής και Τμήμα για την ανάθεση του φοιτητή σε τμήμα μαθήματος από καθηγητή. Η συσχέτιση αυτή εκφράζεται μέσω της οντότητας Επιλογή_Μαθήματος η οποία έχει ξένο κλειδί στην οντότητα Τμήμα και δείχνει το τμήμα ανάθεσης για την συγκεκριμένη επιλογή μαθήματος από φοιτητή.

Παρακολούθηση

Η οντότητα Παρακολούθηση έχει σαν βασικό χαρακτηριστικό την Παρουσία. Έχει συσχέτιση 1 προς 0..N με την οντότητα Επιλογή_Μαθήματος, ένας φοιτητής μπορεί να έχει πολλές παρακολουθήσεις σε κάθε μάθημα ανά διδακτική περίοδο. Επίσης έχει συσχέτιση 1 προς 0..N με την οντότητα Συνάντηση, κάθε συνάντηση μπορούν να την παρακολουθήσουν πολλοί φοιτητές. Τέλος, έχει συσχέτιση 0..1 προς 0..N με την οντότητα Τμήμα, η οποία αναφέρεται στο τμήμα στο οποίο προσήλθε ο φοιτητής για τη συγκεκριμένη παρακολούθηση, το οποίο μπορεί να είναι διαφορετικό από το τμήμα στο οποίο έχει ανατεθεί ο φοιτητής ή να μην έχει οριστεί ακόμη (απαίτηση: κ.παρουσίες.καταχώρηση)



Εικόνα 5 – Σχήμα Βάσης Δεδομένων

Περιπτώσεις Χρήσης

Οι περιπτώσεις χρήσης εκπονήθηκαν παράλληλα και ολοκληρώθηκαν μετά τον λογικό σχεδιασμό της βάσης δεδομένων.

Πίνακας Περιπτώσεων Χρήσης

Ο παρακάτω πίνακας δίνει όλες τις περιπτώσεις χρήσης που υλοποιήθηκαν.

Αριθμός	Όνομα	Χρήστης
κφ.αυθεντικοποίηση.εγγραφή ή	Εγγραφή Χρήστη	Καθηγητής, Φοιτητής
χ.αυθεντικοποίηση.σύνδεση	Σύνδεση Χρήστη	Καθηγητής, Φοιτητής, Γραμματεία
κφ.ανάκτηση.κωδικού	Ανάκτηση Κωδικού	Καθηγητής, Φοιτητής
κφ.επικύρωση	Επικύρωση Λογαριασμού	Καθηγητής, Φοιτητής
κφ.επικύρωση.αποστολή	Αποστολή Συνδέσμου Επικύρωσης	Καθηγητής, Φοιτητής
χ.δεδομένα.αλλαγή	Αλλαγή Προσωπικών Δεδομένων	Καθηγητής, Φοιτητής, Γραμματεία
χ.κωδικός.αλλαγή	Αλλαγή Κωδικού Σύνδεσης	Καθηγητής, Φοιτητής, Γραμματεία
χ.αποσύνδεση	Αποσύνδεση Χρήστη	Καθηγητής, Φοιτητής, Γραμματεία
γ.περίοδοι.δημιουργία	Δημιουργία Διδακτικής Περίοδου	Γραμματεία
γ.μαθήματα.δημιουργία	Δημιουργία Μαθήματος	Γραμματεία
γ.μαθήματα.διαγραφή	Διαγραφή Μαθήματος	Γραμματεία
γ.μάθηματα.προσθήκη	Προσθήκη Μαθήματος σε Διδακτική Περίοδο	Γραμματεία
γ.μαθήματα.αφαίρεση	Αφαίρεση Μαθήματος από Διδακτική Περίοδο	Γραμματεία
γ.καθηγητές.ανάθεση	Ανάθεση Καθηγητή σε	Γραμματεία

Αριθμός	Όνομα	Χρήστης
	Μάθημα	
γ.καθηγητές.αφαίρεση	Διαγραφή Ανάθεσης Καθηγητή σε Μάθημα	Γραμματεία
γ.καθηγητές.επικύρωση	Επικύρωση Λογαριασμού Καθηγητή	Γραμματεία
κ.μαθήματα.δημιουργία	Δημιουργία Μαθήματος	Καθηγητής
κ.τμήματα.δημιουργία	Δημιουργία Τμήματος Μαθήματος	Καθηγητής
κ.τμήματα.ενημέρωση	Ενημέρωση Τμημάτων Μαθήματος	Καθηγητής
κ.τμήματα.προσθήκη	Προσθήκη Φοιτητή σε Τμήμα Μαθήματος	Καθηγητής
κ.τμήματα.αφαίρεση	Αφαίρεση Φοιτητή από Τμήμα Μαθήματος	Καθηγητής
κ.συναντήσεις.δημιουργία	Δημιουργία Συνάντησης Μαθήματος	Καθηγητής
κ.συναντήσεις.ενημέρωση	Ενημέρωση Συναντήσεων Μαθήματος	Καθηγητής
κ.παρουσίες.καταχώρηση	Καταχώρηση Παρουσιών Συνάντησης	Καθηγητής
κ.παρουσίες.επισκόπηση	Επισκόπηση Παρουσιών Μαθήματος	Καθηγητής
φ.μαθήματα.επιλογή	Επιλογή Μαθήματος	Φοιτητής
φ.τμήματα.δήλωση	Δήλωση Προτίμησης σε Τμήμα Μαθήματος	Φοιτητής
φ.ιστορικό.επισκόπηση	Επισκόπηση Ιστορικού Παρακολουθήσεων	Φοιτητής

Αναλυτικές Περιπτώσεις Χρήσης

Παρατίθεται μια επιλογή περιπτώσεων χρήσης που αφορούν το χρήστη καθηγητής. Οι υπόλοιπες παραλείπονται για λόγους οικονομίας της παρουσίασης.

Use Case

Αριθμός: χ.αυθεντικοποίηση.σύνδεση

Όνομα: Σύνδεση Χρήστη

Actor: Καθηγητής, Φοιτητής, Γραμματεία

Απαίτηση Λογισμικού: χ.αυθεντικοποίηση.σύνδεση

Βασική Ροή:

Ο Χρήστης ζητά να συνδεθεί στο λογισμικό.

Το λογισμικό ζητάει από το Χρήστη να εισάγει το όνομα χρήστη και τον κωδικό σύνδεσης.

Ο Χρήστης εισάγει το όνομα χρήστη και τον κωδικό σύνδεσης.

Το λογισμικό επαληθεύει το όνομα χρήστη και τον κωδικό σύνδεσης και συνδέει το Χρήστη.

Εναλλακτική Ροή:

Ο Χρήστης εισάγει μη έγκυρο όνομα χρήστη ή κωδικό σύνδεσης.

Το λογισμικό εμφανίζει μήνυμα σφάλματος και ζητάει το όνομα χρήστη και τον κωδικό σύνδεσης ξανά.

Use Case

Αριθμός: κ.τμήματα.δημιουργία

Όνομα: Δημιουργία Τμήματος Μαθήματος

Actor: Καθηγητής

Απαίτηση Λογισμικού: κ.τμήματα.δημιουργία

Βασική Ροή:

Ο Καθηγητής ζητάει να δημιουργήσει τμήμα μαθήματος.

Το λογισμικό ζητάει από τον Καθηγητή να επιλέξει τη διδακτική περίοδο και
και το μάθημα στο οποίο θέλει να δημιουργήσει τμήμα

Ο Καθηγητής επιλέγει τη διδακτική περίοδο και το μάθημα στο οποίο θέλει να
δημιουργήσει τμήμα.

Το λογισμικό ζητάει από τον Καθηγητή να εισάγει το όνομα, την ημέρα, την ώρα
έναρξης και την ώρα λήξης του τμήματος.

Ο Καθηγητής εισάγει το όνομα, την ημέρα, την ώρα έναρξης και την ώρα λήξης του
τμήματος.

Το λογισμικό δημιουργεί τμήμα για το επιλεγμένο μάθημα στην επιλεγμένη
διδακτική περίοδο, με όνομα, ημέρα, ώρα έναρξης και ώρα λήξης αυτά που έχει
εισάγει ο Καθηγητής.

Εναλλακτική Ροή:

Ο Καθηγητής δεν εισάγει όνομα τμήματος ή εισάγει όνομα τμήματος που χρησιμοποιείται
ήδη.

Το λογισμικό εμφανίζει μήνυμα σφάλματος και ζητάει από τον Καθηγητή να εισάγει
το όνομα τμήματος ξανά.

Use Case

Αριθμός: κ.συναντήσεις.δημιουργία

Όνομα: Δημιουργία Συνάντησης Μαθήματος

Actor: Καθηγητής

Απαίτηση Λογισμικού: κ.συναντήσεις.δημιουργία

Βασική Ροή:

Ο Καθηγητής ζητάει να δημιουργήσει συνάντηση μαθήματος.

Το λογισμικό ζητάει από τον Καθηγητή να επιλέξει διδακτική περίοδο και μάθημα στο οποίο θέλει να δημιουργήσει συνάντηση.

Ο Καθηγητής επιλέγει τη διδακτική περίοδο και το μάθημα στο οποίο θέλει να δημιουργήσει συνάντηση.

Το λογισμικό ζητάει από τον Καθηγητή να εισάγει τον προγραμματισμό και την κατάσταση της συνάντησης.

Ο Καθηγητής εισάγει τον προγραμματισμό και την κατάσταση της συνάντησης.

Το λογισμικό δημιουργεί συνάντηση στο επιλεγμένο μάθημα για την επιλεγμένη διδακτική περίοδο, με αρίθμηση τον αριθμό του συνόλου των συναντήσεων μαθήματος, αυξανόμενο κατά ένα και τον προγραμματισμό και την κατάσταση που έχει εισάγει ο Καθηγητής.

Εναλλακτική Ροή:

Ο Καθηγητής δεν εισάγει τον προγραμματισμό ή την κατάσταση της συνάντησης.

Το λογισμικό εμφανίζει μήνυμα σφάλματος και ζητάει από τον Καθηγητή να εισάγει τον προγραμματισμό και την κατάσταση της συνάντησης ξανά.

Use Case

Αριθμός: κ.παρουσίες.καταχώρηση

Όνομα: Καταχώρηση Παρουσιών

Actor: Καθηγητής

Αντίστοιχη Απαίτηση Λογισμικού: κ.παρουσίες.καταχώρηση

Βασική Ροή:

Ο Καθηγητής επιλέγει να καταχωρήσει τις παρουσίες σε ένα μάθημα.

Το λογισμικό ζητάει από τον Καθηγητή να επιλέξει διδακτική περίοδο, μάθημα, συνάντηση και προαιρετικά το τμήμα στο οποίο θέλει να καταχωρήσει τις παρουσίες.

Ο Καθηγητής επιλέγει την διδακτική περίοδο, το μάθημα, την συνάντηση και προαιρετικά το τμήμα στο οποίο θέλει να καταχωρήσει παρουσίες.

Το λογισμικό εμφανίζει αναφορά με τους φοιτητές του επιλεγμένου μαθήματος ή αν έχει επιλεγεί τμήμα, μόνο του επιλεγμένου τμήματος.

Το λογισμικό επιτρέπει στον Καθηγητή να αναζητήσει φοιτητές με βάση το όνομα, το επίθετο ή τον αριθμό μητρώου.

Η αναφορά περιέχει το όνομα, το επίθετο, τον αριθμό μητρώου, το τμήμα ανάθεσης, το τμήμα παρακολούθησης και την παρουσία.

Το λογισμικό ζητάει από τον Καθηγητή να εισάγει την παρουσία και προαιρετικά το τμήμα παρακολούθησης για κάθε φοιτητή.

Ο Καθηγητής εισάγει τις παρουσίες και τα τμήματα παρακολούθησης για κάθε φοιτητή.

Το λογισμικό αποθηκεύει τις αλλαγές και εμφανίζει την αναφορά ξανά, με τις παρουσίες και τα τμήματα παρακολούθησης που έχει εισάγει ο Καθηγητής.

Το λογισμικό εμφανίζει σύνδεσμο για τη μεταφόρτωση της αναφοράς σε αρχείο Excel.

Εναλλακτική Ροή:

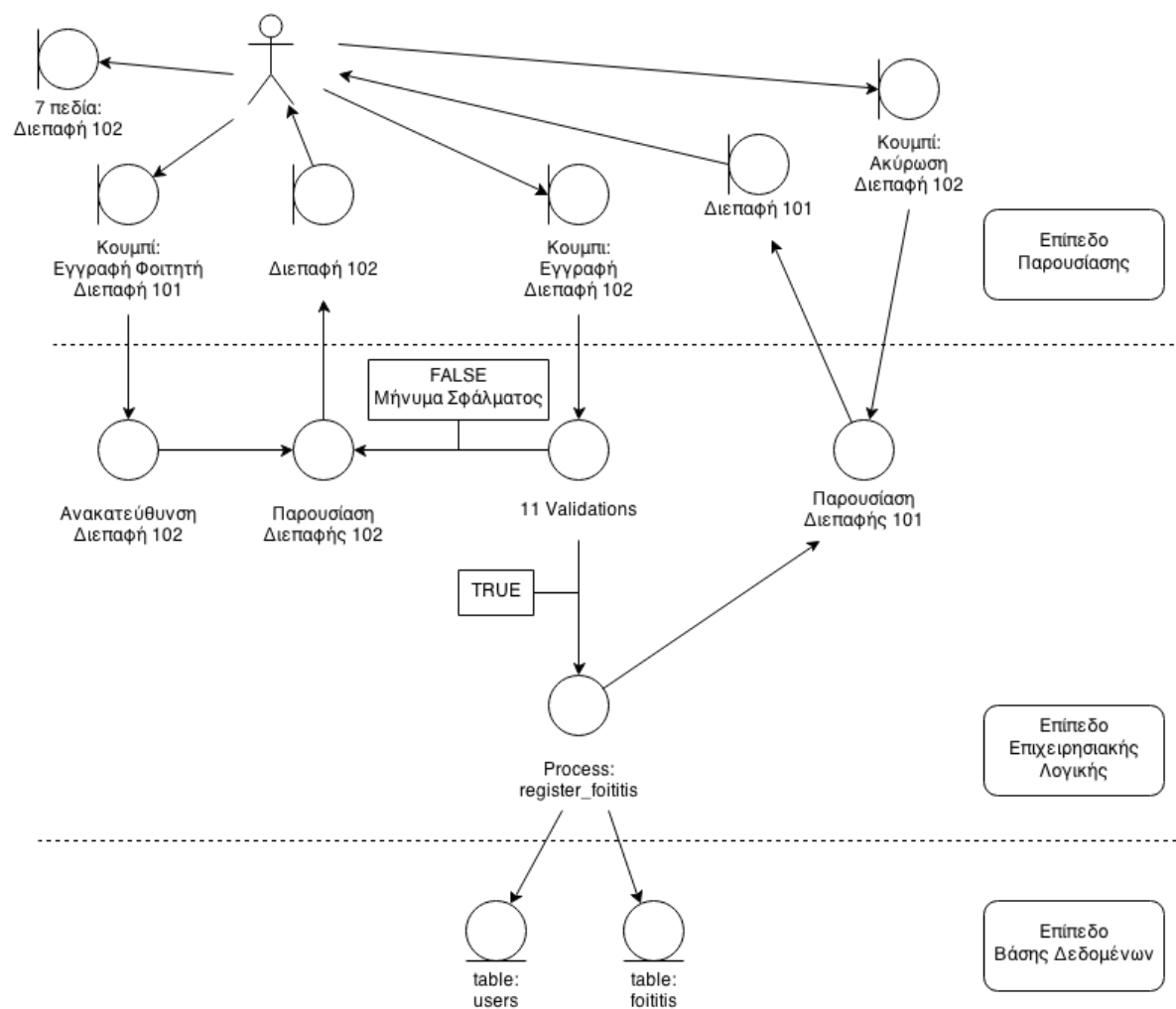
Εάν δεν υπάρχουν φοιτητές εμφανίζεται το μήνυμα «Δεν βρέθηκαν φοιτητές».

Ανάλυση και Σχεδιασμός ανά Περίπτωση Χρήσης

Δίνονται μερικά παραδείγματα ανάλυσης και σχεδιασμού με χρήση διαγραμμάτων στοιβαρότητας UML.

Εγγραφή Χρήστη

Αντίστοιχη περίπτωση χρήσης: κφ.αυθεντικοποίηση.εγγραφή



Εικόνα 6 - Διάγραμμα Στιβαρότητας Εγγραφής Χρήστη

Για την υλοποίηση της εγγραφής χρήστη, ο Χρήστης από την σελίδα 101 – Σύνδεση Χρήστη πατά το κουμπί Εγγραφή Φοιτητή ή Εγγραφή Καθηγητή ανάλογα με τον ρόλο με τον οποίο θέλει να εγγραφεί στο λογισμικό. Τα κουμπιά τον παραπέμπουν στην σελίδα 102 – Εγγραφή Φοιτητή ή 103 – Εγγραφή Καθηγητή αντίστοιχα.

Διεπαφές Χρήστη

Στο επίπεδο παρουσίασης οι διεπαφές έχουν τα παρακάτω αντικείμενα:

Σελίδα 101 – Σύνδεση Χρήστη

Region – Σύνδεση Χρήστη

Text Field – Όνομα Χρήστη

Password Field – Κωδικός

Button – Σύνδεση

Button – Εγγραφή Φοιτητή

Button – Εγγραφή Καθηγητή

Button – Ανάκτηση Κωδικού

Σελίδα 102 – Εγγραφή Φοιτητή

Region – Εγγραφή Φοιτητή

Text Field – Όνομα Χρήστη

Password Field – Κωδικός

Password Field – Κωδικός Ξανά

Text Field – Ηλεκτρονικό Ταχυδρομείο

Text Field – Όνομα

Text Field – Επίθετο

Text Field – Αριθμός Μητρώου

Button – Εγγραφή

Button – Ακύρωση

Σελίδα 102 – Εγγραφή Καθηγητή

Region – Εγγραφή Καθηγητή

Text Field – Όνομα Χρήστη

Password Field – Κωδικός

Password Field – Κωδικός Ξανά

Text Field – Ηλεκτρονικό Ταχυδρομείο

Text Field – Όνομα

Text Field – Επίθετο

Button – Εγγραφή

Button – Ακύρωση

Εικόνα 7 - Διεπαφή Εγγραφής Φοιτητή

Στο επίπεδο επιχειρησιακής λογικής η σελίδα 102 – Εγγραφή Φοιτητή έχει τα παρακάτω αντικείμενα:

Όνομα	Τύπος	Σκοπός
P102_USERNAME_NN	Validation	Ελέγχει εάν το όνομα χρήστη είναι κενό
P102_USERNAME_LENGTH	Validation	Ελέγχει εάν το όνομα χρήστη είναι τουλάχιστον 3 χαρακτήρες
P102_USERNAME_EXISTS	Validation	Ελέγχει εάν το όνομα χρήστη χρησιμοποιείται ήδη
P102_PASSWORD_NN	Validation	Ελέγχει εάν ο κωδικός είναι κενός
P102_PASSWORD_LENGTH	Validation	Ελέγχει εάν ο κωδικός είναι τουλάχιστον 6 χαρακτήρες
P102_PASSWORDS_MATCH	Validation	Ελέγχει εάν η επαλήθευση του κωδικού είναι σωστή
P102_EMAIL_NN	Validation	Ελέγχει εάν το email είναι κενό

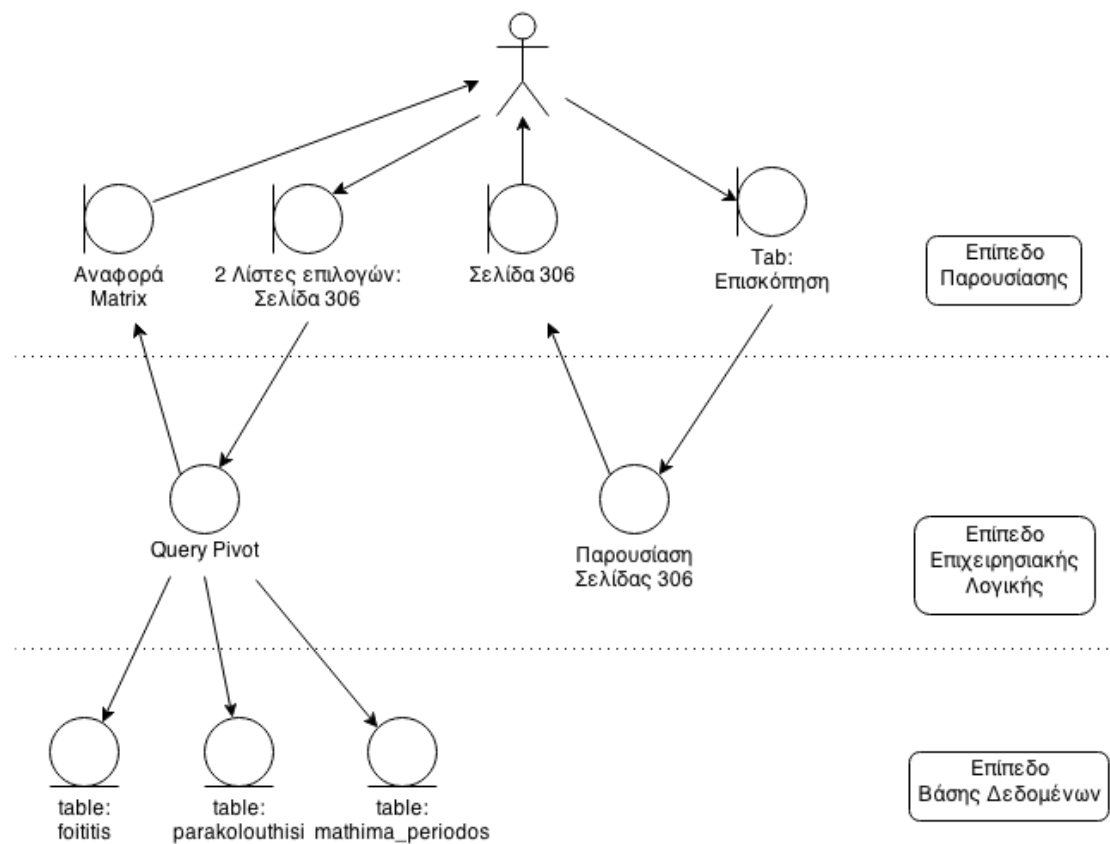
P102_EMAIL_REGEX	Validation	Ελέγχει εάν το email είναι έγκυρο
P102_EMAIL_EXISTS	Validation	Ελέγχει εάν το email χρησιμοποιείται
P102_AM_REGEX	Validation	Ελέγχει εάν ο αριθμός μητρώου είναι έγκυρος
P102_AM_EXISTS	Validation	Ελέγχει εάν ο αριθμός μητρώου χρησιμοποιείται
P102_REGISTER_FOITITIS	Process	Καλεί την μέθοδο <code>register_foititis</code>
P102_REDIRECT	Branch	Ανακατευθύνει τον χρήστη στην σελίδα 101 – Σύνδεση Χρήστη μετά την υποβολή της σελίδας

Ο Χρήστης εισάγει τα δεδομένα και πατάει το κουμπί Εγγραφή. Εκτελούνται διαδοχικά τα validations. Εάν αποτύχουν εμφανίζεται μήνυμα σφάλματος. Εάν επιτύχουν εκτελείτε η process P102_REGISTER_FOITITIS η οποία καλεί την μέθοδο `register_foititis` και εγγράφει τον χρήστη ως Φοιτητή. Εάν επιτύχει και η process τότε εκτελείτε η ανακατεύθυνση στην σελίδα 101 – Σύνδεση Χρήστη, όπου ο Χρήστης μπορεί να συνδεθεί με το όνομα χρήστη και τον κωδικό με τα οποία εγγράφηκε.

Αντίστοιχη είναι και η διαδικασία για την εγγραφή Καθηγητή στη διεπαφή 103 - Εγγραφή Καθηγητή.

Επισκόπηση Παρουσιών

Αντίστοιχη περίπτωση χρήσης: κ.παρουσίες.επισκόπηση



Εικόνα 8 - Διάγραμμα Στιβαρότητας Επισκόπησης Παρουσιών

Ο Καθηγητής κάνει κλικ στο tab επισκόπηση και το λογισμικό τον παραπέμπει στην σελίδα 306 – Επισκόπηση Παρουσιών. Ο Καθηγητής επιλέγει από τις δύο λίστες επιλογών την διδακτική περίοδο και το μάθημα για το οποίο θέλει να δει την επισκόπηση των παρουσιών. Το λογισμικό εκτελεί το αποθηκευμένο pivot query σύμφωνα με τις επιλογές του Καθηγητή και εμφανίζει την αναφορά matrix.

Διεπαφή Χρήστη

Σελίδα – 306 Επισκόπηση Παρουσιών

Tabset – TS1

Περιοχή – Περίοδος και Μάθημα

Λίστα Επιλογών – Περίοδος

Λίστα Επιλογών – Μάθημα

Αναφορά – Επισκόπηση Παρουσιών

Πεδίο – Όνομα

Πεδίο – Επίθετο

Πεδίο – ΑΜ

Πεδία – Παρουσίες Συναντήσεων

Σύνδεσμος – Μεταφόρτωση

Τμήματα	Συναντήσεις	Αναθέσεις	Παρουσίες	Επισκόπηση
---------	-------------	-----------	-----------	------------

Περίοδος:

Μαθημα:

Επισκόπηση Παρουσιών

Όνομα	Επίθετο	ΑΜ	1	2	3	4	5	6	7	8	9	10	11	12	13	14
ονομα1	epitheto1	1000	0	0	0	0	0	0	0	0	0	-	-	-	-	-
ονομα2	epitheto2	2000	0	1	0	0	1	0	1	0	1	-	-	-	-	-
ονομα3	epitheto3	3000	1	0	1	1	0	1	0	1	0	-	-	-	-	-
ονομα4	epitheto4	4000	0	1	0	0	1	0	1	0	1	-	-	-	-	-
ονομα5	epitheto5	5000	0	0	0	0	0	0	0	0	0	-	-	-	-	-
ονομα6	epitheto6	6000	1	0	1	1	0	1	0	1	0	-	-	-	-	-
ονομα7	epitheto7	7000	0	1	0	0	1	0	1	0	1	-	-	-	-	-
ονομα8	epitheto8	8000	1	0	1	1	0	1	0	1	0	-	-	-	-	-
Οδυσσέας	Σαμαράς	5148	1	0	1	1	0	1	0	1	0	-	-	-	-	-

[Μεταφόρτωση](#)

Εικόνα 9 - Διεπαφή Επισκόπησης Παρουσιών

Για την υλοποίηση της αναφοράς χρησιμοποιείται η μέθοδος `pivot` της Oracle. Η μέθοδος χρησιμοποιεί δύο πεδία ενός `query` και δημιουργεί στήλες με βάση τις τιμές του ενός πεδίου και σε κάθε σειρά υπάρχουν οι τιμές του δεύτερου πεδίου για την τιμή της στήλης. Έτσι στην αναφορά της διεπαφής, αντί να υπάρχουν πολλαπλές εγγραφές φοιτητή για κάθε συνάντηση και την παρουσία, με τη μέθοδο `pivot` υπάρχουν στήλες για όλες τις συναντήσεις και σε κάθε σειρά οι αντίστοιχες παρουσίες.

Η μέθοδος `pivot` έχει τον περιορισμό ότι ο αριθμός των πεδίων, στην προκειμένη περίπτωση ο αριθμός των συναντήσεων, δεν μπορεί να είναι δυναμικός και πρέπει το σύνολο των πεδίων να γραφεί στον κώδικα του `query`. Έτσι στην υλοποίηση της πρώτης έκδοσης ο μέγιστος αριθμός συναντήσεων για τις οποίες μπορεί να γίνει επισκόπηση, είναι 20. Για δυναμικό αριθμό πεδίων, υπάρχει η μέθοδος `pivot xml`, η οποία όμως επιστρέφει κώδικα `xml` και δεν μπορεί να χρησιμοποιηθεί από τις αναφορές του APEX. Σε μελλοντική έκδοση του λογισμικού, θα μπορούσε να γραφεί μέθοδος η οποία θα επεξεργάζεται τον κώδικα `xml` και θα επιστρέφει αναφορά του APEX ή πίνακα `html`.

Διεπαφή Χρήστη

Παρακάτω δίνεται ο Πίνακας των ιστοσελίδων της διεπαφής χρήστη που υλοποιήθηκαν

Πίνακας Ιστοσελίδων που υλοποιήθηκαν

Αριθμός	Όνομα	Use Case	Χρήστης
101	Σύνδεση Χρήστη	χ.αυθεντικοποίηση.σύνδεση	Χρήστης
102	Εγγραφή Φοιτητή	κφ.αυθεντικοποιηση.εγγραφή	Φοιτητής
103	Εγγραφή Καθηγητή	κφ.αυθεντικοποιηση.εγγραφή	Καθηγητής
104	Ανάκτηση Κωδικού	κφ.ανάκτηση.κωδικού	Καθηγητής Φοιτητής
105	Προφίλ Χρήστη	κφ.δεδομένα.αλλαγή κφ.κωδικός.αλλαγή	Καθηγητής Φοιτητής
106	Επεξεργασία Προφίλ	κφ.δεδομένα.αλλαγή κφ.κωδικός.αλλαγή	Καθηγητής Φοιτητής
201	Μαθήματα Φοιτητή	φ.μαθήματα.επιλογή	Φοιτητής
202	Δήλωση Προτίμησης Τμημάτων	φ.τμήματα.δήλωση	Φοιτητής
203	Ιστορικό Παρακολουθήσεων	φ.ιστορικό.επισκόπηση	Φοιτητής
301	Μαθήματα Καθηγητή	κ.μαθήματα.δημιουργία	Καθηγητής
302	Τμήματα Μαθήματος	κ.τμήματα.δημιουργία κ.τμήματα.επεξεργασία κ.τμήματα.διαγραφή	Καθηγητής
303	Συναντήσεις Μαθήματος	κ.συναντήσεις.δημιουργία κ.συναντήσεις επεξεργασία	Καθηγητής
304	Αναθέσεις Φοιτητών σε Τμήματα	κ.τμήματα.προσθήκη κ.τμήματα.αφαίρεση	Καθηγητής
305	Καταχώρηση Παρουσιών	κ.παρουσίες.καταχώρηση	Καθηγητής
306	Επισκόπηση Παρουσιών	κ.παρουσίες.επισκόπηση	Καθηγητής
401	Καθηγητές	γ.καθηγητές.επικύρωση	Γραμματεία

Αριθμός	Όνομα	Use Case	Χρήστης
402	Διδακτικές Περίοδοι	γ.περίοδοι.δημιουργία	Γραμματεία
403	Μαθήματα Περιόδου	γ.μάθημα.προσθήκη γ.μάθημα.αφαίρεση	Γραμματεία
404	Μαθήματα	γ.μάθημα.δημιουργία γ.μάθημα.διαγραφή	Γραμματεία
405	Αναθέσεις	γ.καθηγητές.ανάθεση γ.καθηγητές.αφαίρεση	Γραμματεία

Αναλυτικός σχεδιασμός σελίδων

Παρακάτω δίνεται ο αναλυτικός σχεδιασμός των σελίδων που υλοποιήθηκαν για τον χρήστη Καθηγητή.

Κοινά αντικείμενα σε όλες τις σελίδες είναι τα:

Navigation Bar

Σύνδεσμος – Προφίλ (σελίδα 105 – Προφίλ Χρήστη)

Σύνδεσμος – Αποσύνδεση

Tabset

Tab – Λογαριασμός

Tab – Μαθήματα (σελίδα 301 – Μαθήματα Καθηγητή)

Tab – Τμήματα (σελίδα 302 – Τμήματα Μαθήματος)

Tab – Συναντήσεις (σελίδα 303 – Συναντήσεις Μαθήματος)

Tab – Αναθέσεις (σελίδα 304 – Αναθέσεις Φοιτητών σε Τμήματα)

Tab – Παρουσίες (σελίδα 305 – Καταχώρηση Παρουσιών)

Tab – Επισκόπηση (σελίδα 306 – Επισκόπηση Παρουσιών)

Ο Καθηγητής μπορεί να χρησιμοποιήσει τους συνδέσμους του Navigation Bar, για να έχει πρόσβαση στην σελίδα του προφίλ χρήστη, ώστε να μπορεί να επεξεργαστεί τα προσωπικά δεδομένα του και να αλλάξει τον κωδικό σύνδεσης του ή να αποσυνδεθεί από το λογισμικό κάνοντας κλικ στον σύνδεσμο Αποσύνδεση.

Κάνοντας κλικ στα Tabs του Tabset, ο Καθηγητής μπορεί να περιηγηθεί σε όλες τις σελίδες στις οποίες έχει πρόσβαση. Σε όλες τις σελίδες υπάρχουν λίστες επιλογών για διδακτική περίοδο ή μάθημα, το λογισμικό αποθηκεύει τις επιλογές του Καθηγητή κατά την περιήγηση, ώστε οι αναφορές κάθε σελίδας να εμφανίζουν δεδομένα σχετικά με την τελευταία επιλογή του Καθηγητή. Για την υλοποίηση αυτής της λειτουργίας χρησιμοποιήθηκαν δύο global items, ένα για την διδακτική περίοδο και ένα για το μάθημα. Οι λίστες επιλογών έχουν ρυθμιστεί ώστε όταν αλλάξει η τιμή τους να γίνεται η υποβολή της σελίδας και οι σελίδες έχουν μεθόδους που εκτελούνται κατά την υποβολή και αποθηκεύουν τις τιμές από τις λίστες επιλογών στα αντίστοιχα global items.

Σχεδιασμός σελίδων Καθηγητή:

Σελίδα 301 – Μαθήματα Καθηγητή

Περιοχή – Περίοδος

Λίστα Επιλογών – Περίοδος

Αναφορά – Τα Μαθήματα μου

Πεδίο – Μάθημα

Πεδίο – Τμήματα

Περιοχή – Δημιουργία Μαθήματος

Πεδίο Κειμένου – Όνομα Μαθήματος

Κουμπί – Δημιουργία

Σελίδα 302 – Τμήματα Μαθήματος

Περιοχή – Περίοδος Μάθημα

Λίστα Επιλογών – Περίοδος

Λίστα Επιλογών – Μάθημα

Αναφορά – Τμήματα Μαθήματος

Πεδίο – Τμήμα

Λίστα Επιλογών – Ημέρα

Λίστα Επιλογών – Έναρξη

Λίστα Επιλογών – Λήξη

Κουμπί – Αποθήκευση

Περιοχή – Δημιουργία Τμήματος

Πεδίο Κειμένου – Όνομα Τμήματος

Λίστα Επιλογών – Ημέρα

Λίστα Επιλογών – Έναρξη

Λίστα Επιλογών – Λήξη

Κουμπί – Δημιουργία

Σελίδα 303 – Συναντήσεις Μαθήματος

Περιοχή – Περίοδος Μάθημα

Λίστα Επιλογών – Περίοδος

Λίστα Επιλογών – Μάθημα

Αναφορά – Συναντήσεις Μαθήματος

Πεδίο – Συνάντηση

Λίστα Επιλογών – Κατάσταση

Λίστα Επιλογών – Προγραμματισμός

Κουμπί – Αποθήκευση

Περιοχή – Δημιουργία Συνάντησης

Λίστα Επιλογών – Κατάσταση

Λίστα Επιλογών – Προγραμματισμός

Κουμπί – Δημιουργία

Σελίδα 304 – Αναθέσεις Φοιτητών σε Τμήματα

Περιοχή – Περίοδος Μάθημα Τμήμα

Λίστα Επιλογών – Περίοδος

Λίστα Επιλογών – Μάθημα

Λίστα Επιλογών – Τμήμα

Αναφορά – Φοιτητές Τμήματος

Πεδίο – Όνομα

Πεδίο – Επίθετο

Πεδίο – ΑΜ

Checkbox – Αφαίρεση

Κουμπί – Αφαίρεση

Αναφορά – Διαθέσιμοι Φοιτητές

Πεδίο – Όνομα

Πεδίο – Επίθετο

Πεδίο – ΑΜ

Πεδίο – Τμήμα

Πεδίο – Προτεραιότητα

Πεδίο – Λόγος Αδυναμίας

Πεδίο – Εξήγηση

Checkbox – Προσθήκη

Κουμπί – Προσθήκη

Σελίδα 305 – Καταχώρηση Παρουσιών

Περιοχή – Περίοδος Μάθημα Συνάντηση Τμήμα

Λίστα Επιλογών – Περίοδος

Λίστα Επιλογών – Μάθημα

Λίστα Επιλογών – Συνάντηση

Λίστα Επιλογών – Τμήμα

Περιοχή – Αναζήτηση

Πεδίο Κειμένου – Αναζήτηση

Λίστα Επιλογών – Προβολή

Κουμπί – Αναζήτηση

Κουμπί – Επαναφορά

Αναφορά – Παρουσίες

Πεδίο – Όνομα

Πεδίο – Επίθετο

Πεδίο – ΑΜ

Πεδίο – Τμήμα Ανάθεσης

Λίστα Επιλογών – Τμήμα Παρακολούθησης

Checkbox – Παρουσία

Σύνδεσμος – Μεταφόρτωση

Κουμπί – Αποθήκευση

Σελίδα 306 – Επισκόπηση Παρουσιών

Περιοχή – Περίοδος και Μάθημα

Λίστα Επιλογών – Περίοδος

Λίστα Επιλογών – Μάθημα

Αναφορά – Επισκόπηση Παρουσιών

Πεδίο – Όνομα

Πεδίο – Επίθετο

Πεδίο – ΑΜ

Πεδία – Παρουσίες Συναντήσεων

Σύνδεσμος – Μεταφόρτωση

Υλοποίηση

Σχήμα της Βάσης Δεδομένων

Το Σχήμα της Βάσης Δεδομένων υλοποιήθηκε σε γλώσσα Oracle SQL. Παρακάτω δίνεται σαν παράδειγμα ο κώδικας για την δημιουργία της τριαδικής σχέσης, μεταξύ Καθηγητή, Μάθημα και Περίοδος.

Αρχικά γίνεται η δημιουργία των πινάκων:

```
--Πίνακας με τα στοιχεία των καθηγητών
drop table kathigitis cascade constraints;
create table kathigitis(
    id number not null,
    onoma varchar(50) not null,
    epitheto varchar(50) not null,
    epikiromenos number not null, --Πρέπει να επικυρωθεί από άλλον
    καθηγητή
    user_id number not null
);
```

```
--Πίνακας που παριστάνει τα μαθήματα
drop table mathima cascade constraints;
create table mathima(
    id number not null,
    thema varchar2(180) not null
);
```

```
--Πίνακας που παριστάνει τις διδακτικές περιόδους
drop table periodos cascade constraints;
create table periodos(
    id number not null,
    onoma varchar(180) not null,
    start_date date not null,
    end_date date not null,
    num_weeks number
);
```

Επειδή η δυαδική σχέση μεταξύ Μαθήματος και περιόδου, χρησιμοποιείται σε πολλές τριαδικές σχέσεις, για λόγους ευκολίας των SQL queries, δημιουργήθηκε ο ενδιάμεσος πίνακας με την σχέση πολλά-προς-πολλά μεταξύ Μαθήματος και Περιόδου ο οποίος χρησιμοποιείτε για να δείξει τις τριαδικές σχέσεις:

```
--Πίνακας σχέσης μαθημάτων και διδακτικών περιόδων
drop table mathima_periodos cascade constraints;
create table mathima_periodos(
    id          number not null,
    mathima_id  number not null,
    periodos_id number not null
);
```

Ο Ενδιάμεσος πίνακας με την σχέση πολλά-προς-πολλά μεταξύ του Καθηγητή και του ενδιάμεσου πίνακα με την σχέση πολλά-προς-πολλά του Μαθήματος και της περιόδου, εκφράζει την τριαδική σχέση πολλά-προς-πολλά-προς-πολλά μεταξύ Καθηγητή, Μαθήματος και Περιόδου:

```
drop table kath_math_per cascade constraints;
create table kath_math_per(
    id number not null,
    kathigitis_id number not null,
    mathima_periodos_id number not null
);
```

Παρακάτω δίνονται οι περιορισμοί με τα πρωτεύοντα και τα ξένα κλειδιά των παραπάνω πινάκων:

<pre>alter table kathigitis primary key(id);</pre>	<pre>add constraint kathigitis_pk</pre>
<pre>alter table mathima primary key(id);</pre>	<pre>add constraint mathima_pk</pre>
<pre>alter table periodos primary key(id);</pre>	<pre>add constraint periodos_pk</pre>
<pre>alter table mathima_periodos primary key(id);</pre>	<pre>add constraint mathima_periodos_pk</pre>
<pre>alter table kath_math_per primary key(id);</pre>	<pre>add constraint kath_math_per_pk</pre>

```
alter table mathima_periodos add constraint
mathima_periodos_mathima_id_fk foreign key(mathima_id)
references mathima(id) on delete cascade;
alter table mathima_periodos add constraint
mathima_periodos_did_per_id_fk foreign key(periodos_id)
references periodos(id) on delete cascade;
alter table kath_math_per add constraint
kath_math_per_kath_id_fk foreign key(kathigitis_id)
references kathigitis(id) on delete cascade;
alter table kath_math_per add constraint
kath_math_per_math_per_fk foreign key(mathima_periodos_id)
references mathima_periodos(id) on delete cascade;
```

```
alter table mathima      add constraint mathima_thema_uq
unique(thema);
alter table periodos     add constraint periodos_onoma_uq
unique(onoma);
```

```
alter table mathima_periodos add constraint mathima_periodos_uq
unique(mathima_id, periodos_id);
```

```
alter table kath_math_per      add constraint kath_math_per_uq
unique(kathigitis_id, mathima_periodos id);
```

```
DROP SEQUENCE "KATHIGITIS_SEQ";
CREATE SEQUENCE "KATHIGITIS_SEQ" MINVALUE 1 MAXVALUE
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20
NOORDER NOCYCLE;
/
```

Και το αντίστοιχο trigger:

```
create or replace trigger "BI_KATHIGITIS"
  before insert on "KATHIGITIS"
  for each row
begin
  if :NEW."ID" is null then
    select "KATHIGITIS_SEQ".nextval into :NEW."ID" from sys.dual;
  end if;
end;
/
```


Υλοποίηση Διεπαφής Χρήστη

Παρακάτω δίνονται οι εικόνες από την υλοποίηση των σελίδων για τον χρήστη Καθηγητή.

Μαθήματα

Τμήματα

Συναντήσεις

Αναθέσεις

Παρουσίες

Καθηγητή

Περίοδος:

Εαρινή 2013-2014

Τα μαθήματά μου

Μάθημα	Τμήματα
Ασφάλεια Πληροφοριακών Συστημάτων	Τμήματα
Δομημένος	Τμήματα
Εισαγωγή στην Πληροφορική	Τμήματα
Πληροφοριακά Συστήματα Επιχειρήσεων	Τμήματα

1 - 4

Δημιουργία Μαθήματος

Όνομα Μαθήματος:

Δημιουργία

Εικόνα 10 - Σελίδα 301 - Μαθήματα Καθηγητή

Τμήματα

Συναντήσεις

Αναθέσεις

Παρουσίες

Ε

Τα Μαθήματος

Περίοδος:

Εαρινή 2013-2014

Μάθημα:

Ασφάλεια Πληροφοριακών Συστημάτων

Τμήματα Μαθήματος

Τμήμα	Ημέρα	Έναρξη	Λήξη
ΑΠΣ Τμήμα 2	Δευτέρα	12 πμ	2 μμ
ΑΠΣ Τμήμα 1	Τρίτη	8 πμ	10 πμ
ΑΠΣ Τμήμα 3	Πέμπτη	4 μμ	6 μμ
ΑΠΣ Τμήμα 4	Παρασκευή	10 πμ	12 πμ

1 - 4

Αποθήκευση

Δημιουργία Τμήματος

Τμήμα:

ΑΠΣ Τμήμα 5

Ημέρα:

Τετάρτη

Έναρξη:

6 μμ

Λήξη:

8 μμ

Δημιουργία

Εικόνα 11 - Σελίδα 302 - Τμήματα Μαθημάτων

Τμήματα
Συναντήσεις
Αναθέσεις
Παρουσίες
Επι...

ντήσεις Μαθήματος

Περίοδος:
Εαρινή 2013-2014

Μάθημα:
Ασφάλεια Πληροφοριακών Συστημάτων

Συναντήσεις Μαθήματος

Συνάντηση	Κατάσταση	Προγραμματισμός
1	Πραγματοποιημένη	Τακτικός
2	Πραγματοποιημένη	Τακτικός
3	Πραγματοποιημένη	Τακτικός
4	Πραγματοποιημένη	Τακτικός
5	Πραγματοποιημένη	Τακτικός
6	Προγραμματισμένη	Τακτικός
7	Πραγματοποιημένη	Τακτικός
8	Ακυρωμένη	Έκτακτος
9	Προγραμματισμένη	Έκτακτος

1 - 9

Αποθήκευση

Δημιουργία Συνάντησης

Κατάσταση:
-Επιλέξτε Κατάσταση-

Προγραμματισμός:
-Επιλέξτε Προγραμματισμό-

Δημιουργία

Εικόνα 12 - Σελίδα 303 - Συναντήσεις Μαθήματος

Τμήματα	Συναντήσεις	Αναθέσεις	Παρουσίες	Επισκόπηση
---------	-------------	------------------	-----------	------------

στα Μαθήματα > Αναθέσεις Φοιτητών σε Τμήματα

Περίοδος:

Μάθημα:

Τμήμα:

Φοιτητές Τμήματος

Όνομα	Επίθετο	ΑΜ	Αφαίρεση
Οδυσσέας	Σαμαράς	5148	☐
ονομα7	epitheto7	7000	☐

1 - 2

Αφαίρεση

Διαθέσιμοι Φοιτητές

Όνομα	Επίθετο	ΑΜ	Τμήμα	Προτεραιότητα	Λόγος Αδυναμίας	Εξήγηση	Προσθήκη
ονομα5	epitheto5	5000	ΑΠΣ Τμήμα 1	1	-	-	☒
ονομα1	epitheto1	1000	ΑΠΣ Τμήμα 1	0	Τελευταίο Λεωφορείο	Πηγαίνω Πρέβεζα	☐

1 - 2

Προσθήκη

Εικόνα 13 - Σελίδα 304 - Αναθέσεις Φοιτητών σε Τμήματα

Τμήματα	Συναντήσεις	Αναθέσεις	Παρουσίες	Επισκόπηση
---------	-------------	-----------	------------------	------------

Περίοδος: Εαρινή 2013-2014
 Μάθημα: Ασφάλεια Πληροφοριακών Συστημάτων

Συνάντηση: 2
 Τμήμα: -Επιλέξτε Τμήμα-

🔍 Αναζήτηση
 Προβολή 15

Παρουσίες

Όνομα	Επίθετο	ΑΜ	Τμήμα Ανάθεσης	Τμήμα Παρακολούθησης	Παρουσία
Οδυσσέας	Σαμαράς	5148	ΑΠΣ Τμήμα 1	- ▼	☐
ονομα2	epitheto2	2000	ΑΠΣ Τμήμα 3	- ▼	☒
ονομα3	epitheto3	3000	ΑΠΣ Τμήμα 2	ΑΠΣ Τμήμα 1 ▼	☐
ονομα4	epitheto4	4000	ΑΠΣ Τμήμα 2	- ▼	☒
ονομα6	epitheto6	6000	ΑΠΣ Τμήμα 2	ΑΠΣ Τμήμα 3 ▼	☐
ονομα7	epitheto7	7000	ΑΠΣ Τμήμα 1	- ▼	☒
ονομα8	epitheto8	8000	ΑΠΣ Τμήμα 3	- ▼	☐

1 - 7

[Μεταφόρτωση](#)

Εικόνα 14 - Σελίδα 305 - Καταχώρηση Παρουσιών

Τμήματα	Συναντήσεις	Αναθέσεις	Παρουσίες	Επισκόπηση
---------	-------------	-----------	-----------	-------------------

Περίοδος: Εαρινή 2013-2014

Μαθημα: Ασφάλεια Πληροφοριακών Συστημάτων

Επισκόπηση Παρουσιών																
Όνομα	Επίθετο	ΑΜ	1	2	3	4	5	6	7	8	9	10	11	12	13	14
onoma1	epitheto1	1000	0	0	0	0	0	0	0	0	0	-	-	-	-	-
onoma2	epitheto2	2000	0	1	0	0	1	0	1	0	1	-	-	-	-	-
onoma3	epitheto3	3000	1	0	1	1	0	1	0	1	0	-	-	-	-	-
onoma4	epitheto4	4000	0	1	0	0	1	0	1	0	1	-	-	-	-	-
onoma5	epitheto5	5000	0	0	0	0	0	0	0	0	0	-	-	-	-	-
onoma6	epitheto6	6000	1	0	1	1	0	1	0	1	0	-	-	-	-	-
onoma7	epitheto7	7000	0	1	0	0	1	0	1	0	1	-	-	-	-	-
onoma8	epitheto8	8000	1	0	1	1	0	1	0	1	0	-	-	-	-	-
Οδυσσέας	Σαμαράς	5148	1	0	1	1	0	1	0	1	0	-	-	-	-	-

[Μεταφόρτωση](#)

Εικόνα 15 - Σελίδα 306 - Επισκόπηση Παρουσιών

Αποθηκευμένος Κώδικας στο Σχήμα της Βάσης Δεδομένων

Στο Σχήμα της Βάσης Δεδομένων υπάρχει αποθηκευμένος κώδικας με μεθόδους και συναρτήσεις, χωρισμένος σε πακέτα ανάλογα με την κατηγορία των λειτουργιών που εκτελούν.

Υπάρχουν πέντε διαφορετικά πακέτα:

- Το πακέτο `authentication` το οποίο περιέχει μεθόδους και συναρτήσεις σχετικές με την αυθεντικοποίηση και την εξουσιοδότηση των χρηστών.
- Το πακέτο `registration` με μεθόδους και συναρτήσεις σχετικές με την εγγραφή των χρηστών.
- Το πακέτο `administration` με μεθόδους και συναρτήσεις σχετικές με την διαχείριση διδακτικών περιόδων, μαθημάτων, τμημάτων μαθήματος και συναντήσεων μαθήματος.
- Το πακέτο `parousies` με μεθόδους και συναρτήσεις σχετικές με την καταχώρηση παρουσιών .
- Το πακέτο `diloseis` με μεθόδους και συναρτήσεις σχετικές με την δήλωση μαθημάτων και προτίμησης τμημάτων από τους φοιτητές

Παρακάτω δίνεται σαν παράδειγμα ο κώδικας του πακέτου `registration` το οποίο περιέχει :

- Συνάρτηση που ελέγχει εάν το όνομα χρήστη χρησιμοποιείται ήδη και επιστρέφει `boolean`
- Συνάρτηση που ελέγχει εάν το ηλεκτρονικό ταχυδρομείο χρησιμοποιείται ήδη και επιστρέφει `boolean`
- Συνάρτηση που ελέγχει εάν ο αριθμός μητρώου φοιτητή χρησιμοποιείται ήδη και επιστρέφει `boolean`
- Συνάρτηση που εγγράφει χρήστη και επιστρέφει το πρωτεύων κλειδί του
- Μέθοδο που εγγράφει φοιτητή
- Μέθοδο που εγγράφει καθηγητή
- Μέθοδο που επικυρώνει χρήστη
- Μέθοδο που αποστέλλει νέο κωδικό στο ηλεκτρονικό ταχυδρομείο του χρήστη
- Μέθοδο που αποστέλλει σύνδεσμο επιβεβαίωσης στο ηλεκτρονικό ταχυδρομείο του χρήστη
- Συνάρτηση που επιστρέφει το ηλεκτρονικό ταχυδρομείο του συνδεδεμένου χρήστη

- Συνάρτηση που επιστρέφει τον αριθμό μητρώου του συνδεδεμένου φοιτητή


```

create or replace package registration as

    --Έλεγχος για τους περιορισμούς μοναδικότητας
    function username_exists(p_username in varchar2) return boolean;
    function email_exists(p_email in varchar2) return boolean;
    function am_exists(p_am in number) return boolean;

    --Εγγραφεί τον χρήστη και επιστρέφει το id
    function register_user(
        p_username in varchar2,
        p_password in varchar2,
        p_email in varchar2
    ) return number;

    --Εγγράφει τον φοιτητή
    procedure register_foititis(
        p_username in varchar2,
        p_password in varchar2,
        p_email in varchar2,
        p_onoma in varchar2,
        p_epitheto in varchar2,
        p_am in number
    );

    --Εγγραφεί τον καθηγητή
    procedure register_kathigitis(
        p_username in varchar2,
        p_password in varchar2,
        p_email in varchar2,
        p_onoma in varchar2,
        p_epitheto in varchar2
    );

    --Επικυρώνει τον χρήστη
    procedure activate_user( p_hashed_password in varchar2);

    --Δημιουργεί και αποστέλλει νέο κωδικό
    procedure email_new_password(p_email in varchar2);

    --Στέλνει σύνδεσμο επιβεβαίωσης
    procedure send_activation_link(p_user_id in number);

    --Επιστρέφει το email του χρήστη με βάση την μεταβλητή Apex
    v('APP_USER')
    function get_user_email return varchar2;
    --Επιστρέφει το AM του φοιτητή με βάση το user_id
    function get_foititis_am return number;

end registration;
/

```

```

create or replace package body registration as

function username_exists(p_username in varchar2) return boolean
as
    l_id users.id%type;
begin
    select id
        into l_id
        from users
        where username = p_username;
    return true;
    exception when no_data_found
        then return false;
end username_exists;

function email_exists(p_email in varchar2) return boolean
as
    l_id users.id%type;
begin
    select id
        into l_id
        from users
        where email = p_email;
    return true;
    exception when no_data_found
        then return false;
end email_exists;

function am_exists(p_am in number) return boolean
as
    l_user_id foititis.user_id%type;
begin
    select user_id
        into l_user_id
        from foititis
        where am = p_am;
    return true;
    exception when no_data_found
        then return false;
end am_exists;

procedure send_activation_link(p_user_id in number)
is
    l_hashed_password users.password%type;
    l_email users.email%type;
    l_username users.username%type;
    l_host varchar2(500);
    l_body varchar2(1000);
    l_body_html varchar2(4000);
    l_activation_link varchar2(500);
    l_email_host varchar2(500);
    crlf VARCHAR2(2) := chr(13)||chr(10);
begin
    select email, username, password

```

```

        into l_email, l_username, l_hashed_password
        from users
        where id = p_user_id;

        l_host := owa_util.get_cgi_env('HTTP_HOST');
        l_activation_link := 'http://' || l_host || '/apex/'

        || 'f?p=112:101:' || ':' || ':NO' || '::' || 'P101_ACTIVATE:' || l_hashed_passw
ord;

        l_body := 'Για να δείτε αυτό το email χρησιμοποιήστε λογισμικό
που να υποστηρίζει HTML' || crlf;

        l_body_html := '<html><head></head><body><h3>Τα στοιχεία του
λογαριασμού σας είναι τα εξής</h3>' || crlf;
        l_body_html := l_body_html || '<p>' || 'Όνομα Χρήστη:
' || l_username || '</p>' || crlf;
        l_body_html := l_body_html || '<p>' || 'Ηλεκτρονικό Ταχυδρομείο:
' || l_email || '</p>' || crlf;

        l_body_html := l_body_html || '<p>' || 'Για να ενεργοποιήσετε τον
λογαριασμό σας κάντε κλικ εδώ:' || crlf
        || '<a
href="' || l_activation_link || '">' || l_activation_link || '</a></p>';
        l_body_html := l_body_html || '</body></html>' || crlf;

        l_email_host := substr(owa_util.get_cgi_env('HTTP_HOST'), 1,
instr(owa_util.get_cgi_env('HTTP_HOST'), ':') - 1);

        apex_mail.send(
            p_to => l_email,
            p_from => 'admin@' || l_email_host,
            p_body => l_body,
            p_body_html => l_body_html,
            p_subj => 'Ενεργοποίηση Λογαριασμού - Εφαρμογή Διαχείρισης
Εργαστηρίων Εκπαιδευτηρίου'
        );

    end send_activation_link;

    function register_user(
        p_username in varchar2,
        p_password in varchar2,
        p_email in varchar2
    ) return number
    as
        v_id users.id%type;
        v_hashed_password users.password%type;
    begin
        select users_seq.nextval into v_id from sys.dual;
        v_hashed_password := authentication.hash_password(p_username,
p_password);
        insert into users(
            id,

```

```

        username,
        password,
        email,
        activated,
        diaxeiristis
    ) values (
        v_id,
        p_username,
        v_hashed_password,
        p_email,
        0,
        0
    );

    send_activation_link(v_id);

    return v_id;
end register_user;

procedure register_foititis(
    p_username in varchar2,
    p_password in varchar2,
    p_email in varchar2,
    p_onoma in varchar2,
    p_epitheto in varchar2,
    p_am in number
)
as
    v_user_id foititis.user_id%type;
begin
    v_user_id := register_user(
        p_username,
        p_password,
        p_email
    );
    insert into foititis(
        am,
        user_id,
        onoma,
        epitheto
    ) values (
        p_am,
        v_user_id,
        p_onoma,
        p_epitheto
    );
end register_foititis;

procedure register_kathigitis(
    p_username in varchar2,
    p_password in varchar2,
    p_email in varchar2,
    p_onoma in varchar2,
    p_epitheto in varchar2

```

```

)
as
    v_user_id kathigitis.user_id%type;
begin
    v_user_id := register_user(
        p_username,
        p_password,
        p_email
    );
    insert into kathigitis(
        onoma,
        epitheto,
        user_id,
        epikiromenos
    ) values (
        p_onoma,
        p_epitheto,
        v_user_id,
        0
    );
end register_kathigitis;

procedure activate_user( p_hashed_password in varchar2)
is
begin
    update users
        set activated = 1
        where password = p_hashed_password;
end;

procedure email_new_password(p_email in varchar2)
is
    l_username users.username%type;
    l_password users.password%type;
    l_hashed_password users.password%type;
begin
    select username
    into l_username
    from users
    where upper(users.email) = upper(p_email);
    --Δημιουργία νέου password
    select dbms_random.string('A', 8)
    into l_password
    from sys.dual;

    --Κρυπτογράφηση του password
    l_hashed_password := authentication.hash_password(l_username,
l_password);

    --Ενημέρωση του password
    update users
        set password = l_hashed_password
        where upper(users.email) = upper(p_email);

```

```

        APEX_MAIL.SEND(
            p_to => p_email,
            p_from => 'admin@edee.teiep.gr',
            p_subj => 'Αλλαγή κωδικού στην Εφαρμογή Διαχείρισης
Εργασιών Εκπαιδευτηρίου' ,
            p_body => 'Νέα στοιχεία σύνδεσης '
            || 'Όνομα Χρήστη: ' || l_username
            || 'Κωδικός: ' || l_password
        );
    end email_new_password;

function get_user_email return varchar2
is
    l_user_email users.email%type;
begin
    select email
    into l_user_email
    from users
    where id = authentication.get_user_id;
    return l_user_email;
end get_user_email;

function get_foititis_am return number
is
    l_foititis_am foititis.am%type;
begin
    select am
    into l_foititis_am
    from foititis
    where id = authentication.get_foititis_id;
    return l_foititis_am;
end get_foititis_am;

end registration;
/

```

Κώδικας σε Σελίδες

Παρακάτω δίνεται ο κώδικας στις σελίδες για τις περιπτώσεις χρήσης κφ.εγγραφή.χρήστη και κ.παρουσίες.επισκόπηση.

Στην σελίδα 102 – Εγγραφή Φοιτητή υπάρχει η μέθοδος P102_REGISTER_FOITITIS η οποία εκτελείται κατά την υποβολή της σελίδας και περνάει σαν παραμέτρους στην αποθηκευμένη μέθοδο `registration.register_foititis` τους σχήματος της βάσης δεδομένων, τις τιμές που έχει εισάγει ο χρήστης στα αντικείμενα της σελίδας. Η μέθοδος `registration.register_foititis` δημιουργεί νέα εγγραφή φοιτητή την οποία την αντιστοιχεί με νέα εγγραφή χρήστη και αποστέλλει σύνδεσμο επικύρωσης στο email εγγραφής.

```
registration.register_foititis(  
    p_username => :P102_USERNAME,  
    p_password => :P102_PASSWORD,  
    p_email => :P102_EMAIL,  
    p_onoma => :P102_ONOMA,  
    p_epitheto => :P102_EPITHETO,  
    p_am => :P102_AM  
)
```

Στην σελίδα 103 – Εγγραφή Καθηγητή υπάρχει αντίστοιχη μέθοδος P102_REGISTER_KATHIGITIS η οποία κατά την υποβολή της σελίδας καλεί την αποθηκευμένη μέθοδο `registration.register_kathgitis`.

```
registration.register_kathgitis(  
    p_username => :P103_USERNAME,  
    p_password => :P103_PASSWORD,  
    p_email => :P103_EMAIL,  
    p_onoma => :P103_ONOMA,  
    p_epitheto => :P103_EPITHETO  
);
```

Στην σελίδα 306 – Επισκόπηση παρουσιών υπάρχει ο κώδικας του pivot query για την αναφορά matrix.

```
select * from (
  select
    f.onoma as "Όνομα",
    f.epitheto as "Επίθετο",
    f.am, s.arithmisi sinantisi,
    p.parousia parousia
  from
    foititis f,
    sinantisi s,
    parakolouthisi p,
    epilogi_mathimatos em
  where f.id = em.foititis_id
  and em.id = p.epil_math_id
  and em.mathima_periodos_id = :p306_mathima_periodos
  and s.mathima_periodos_id = :p306_mathima_periodos
  and p.sinantisi_id = s.id
) pivot (
  max(parousia) for sinantisi in
  (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20)
)
```

Το query επιστρέφει σειρές με το όνομα, το επίθετο και τον αριθμό μητρώου κάθε φοιτητή και εκτελεί την μέθοδο pivot στο πεδίο της παρουσίας για 20 συναντήσεις του μαθήματος. Έτσι αντί να υπάρχουν πολλαπλές σειρές για κάθε φοιτητή για τις παρουσίες του σε όλες τις συναντήσεις, σε μία σειρά υπάρχουν όλες οι παρουσίες του.

Υλοποίηση Ασφάλειας

Αυθεντικοποίηση και Εξουσιοδότηση

Η ασφάλεια του λογισμικού υλοποιήθηκε σε δύο επίπεδα. Αρχικά πιστοποιείται η ταυτότητα του χρήστη, χρησιμοποιώντας ένα Custom Authentication Scheme του APEX μέσω της διαδικασίας της σύνδεσης χρήστη. Κατόπιν πιστοποιείται ο ρόλος του χρήστη και τα δικαιώματα πρόσβασης που έχει, χρησιμοποιώντας Authorization Schemes του APEX.

Για την υλοποίηση ενός Custom Authentication Scheme απαιτείται η υλοποίηση μιας μεθόδου αυθεντικοποίησης η οποία δέχεται σαν παραμέτρους ένα όνομα χρήστη και έναν κωδικό, ελέγχει την εγκυρότητα τους και επιστρέφει true εάν είναι επιτυχής και false αν αποτύχει.

Στο λογισμικό η μέθοδος που εκτελεί αυτή την λειτουργία είναι η `authenticate_user` του πακέτου `authentication`. Κατά την σύνδεση χρήστη η μέθοδος συγκρίνει το όνομα χρήστη και τον κωδικό σύνδεσης που εισάγει ο χρήστης της με τις εγγραφές του πίνακα `users` για να ελέγξει την εγκυρότητα τους. Αφού πιστοποιηθεί η ταυτότητα του χρήστη, το APEX δημιουργεί νέα σύνοδο χρήστη και αποθηκεύει την τιμή του ονόματος χρήστη στην βοηθητική μεταβλητή `APP_USER`. Η μεταβλητή χρησιμοποιείται σε συνδυασμό με την τιμή της συνόδου χρήστη, από το APEX, ώστε να πιστοποιείται η ταυτότητα του χρήστη καθώς αυτός περιηγείται στις σελίδες του λογισμικού. Κάθε σελίδα μπορεί να έχει σαν περιορισμό ο χρήστης να είναι αυθεντικοποιημένος για να έχει πρόσβαση σε αυτήν. Ο περιορισμός αυτός υπάρχει σε όλες τις σελίδες, εκτός των σελίδων 101 – Σύνδεση Χρήστη, 102 – Εγγραφή Φοιτητή, 103 – Εγγραφή Καθηγητή, 104 – Ανάκτηση κωδικού (απαίτηση: α.αυθεντικοποίηση.χρήστη). Επίσης για την προστασία των δεδομένων κάθε χρήστη, η τιμή της μεταβλητής `APP_USER` χρησιμοποιείται ως παράμετρος σε όλες τις αναφορές και αντικείμενα που περιέχουν προσωπικά δεδομένα, έτσι ώστε πάντα να εμφανίζονται μόνο τα δεδομένα του συνδεδεμένου χρήστη (απαίτηση: α.δεδομένα.χρήστη).

Για την υλοποίηση ενός Authorization Scheme, απαιτείται η υλοποίηση μιας μεθόδου η οποία επιστρέφει true εάν είναι επιτυχής και false αν αποτύχει. Το Authorization Scheme κατόπιν μπορεί να προστεθεί σε οποιαδήποτε σελίδα ή αντικείμενο και ελέγχει αν θα είναι προσβάσιμο ή όχι από τον συνδεδεμένο χρήστη. Στο λογισμικό υλοποιήθηκαν τα παρακάτω Authorization Schemes τα οποία με την βοήθεια της μεταβλητής `APP_USER` πιστοποιούν τον ρόλο του συνδεδεμένου χρήστη:

- **is_unconfirmed_user:** ελέγχει εάν ο Χρήστης δεν έχει επικυρωθεί
- **is_unconfirmed_kathigitis:** ελέγχει εάν ο Χρήστης είναι μη επικυρωμένος Καθηγητής
- **is_foititis:** ελέγχει εάν ο Χρήστης είναι Φοιτητής και έχει επικυρωθεί
- **is_kathigitis:** ελέγχει εάν ο Χρήστης είναι Καθηγητής και έχει επικυρωθεί
- **is_grammateia:** ελέγχει εάν ο Χρήστης είναι Γραμματεία

Τα παραπάνω Authorization Schemes προστέθηκαν στις αντίστοιχες σελίδες και tabs. Έτσι εξασφαλίζεται ότι κάθε χρήστης έχει πρόσβαση μόνο στις σελίδες που προορίζονται για τον ρόλο του και δεν έχει πρόσβαση στις σελίδες που προορίζονται για άλλο ρόλο χρήστη (απαίτηση: α.δεδομένα.χρήστη).

Κρυπτογράφηση Κωδικών Σύνδεσης

Για την κρυπτογράφηση των κωδικών σύνδεσης, χρησιμοποιήθηκε το πακέτο `dbms_obfuscation_toolkit` της Oracle και συγκεκριμένα η μέθοδος `md5`. Η μέθοδος παίρνει σαν παράμετρο ένα αλφαριθμητικό και χρησιμοποιώντας τον αλγόριθμο MD5 επιστρέφει το άθροισμα ελέγχου με κρυπτογράφηση 128-bit.

Στο λογισμικό υλοποιήθηκε η μέθοδος `hash_password` η οποία παίρνει σαν παραμέτρους το όνομα χρήστη και των κωδικό σύνδεσης ενός χρήστη, τα ενώνει μαζί με ένα τυχαίο αλφαριθμητικό και τα περνάει σαν παράμετρο στην μέθοδο `dbms_obfuscation_toolkit.md5` για να πάρει το κρυπτογραφημένο άθροισμα ελέγχου.

```
function hash_password(  
    p_username in varchar2,  
    p_password in varchar2  
) return varchar2  
as  
    l_password varchar2(255);  
    l_salt      varchar2(255) :=  
'2345USFG0JN2T3HW89EFG0BN23R5SDFGAKL';  
begin  
    l_password := utl_raw.cast_to_raw(  
        dbms_obfuscation_toolkit.md5(  
            input_string => p_password || substr(l_salt, 4, 14) ||  
                upper(p_username) || substr(l_salt, 5, 10)  
        )  
    );  
    return l_password;  
end hash_password;
```

Η μέθοδος χρησιμοποιείται κατά την εγγραφή χρήστη (use case: κφ.αυθεντικοποίηση.εγγραφή) για να δημιουργήσει τον κρυπτογραφημένο κωδικό σύνδεσης, ο οποίος και τελικά αποθηκεύεται στην βάση δεδομένων (απαίτηση: α.κρυπτογράφηση.κωδικοί) . Επίσης χρησιμοποιείται κατά την σύνδεση χρήστη (use case: χ.αυθεντικοποίηση.σύνδεση) και κατά την ανάκτηση κωδικού (use case: κφ.ανάκτηση.κωδικού) για να δημιουργήσει και να ελέγξει την εγκυρότητα του κωδικού σύνδεσης.

Λειτουργικές Δοκιμές

Με βάση τις απαιτήσεις λογισμικού, σχεδιάστηκαν και εκτελέστηκαν λειτουργικές δοκιμές. Παρατίθεται μέρος του σχεδίου δοκιμών, με έμφαση στις δοκιμές ασφάλειας

Δοκιμές Εγκυρότητας Δεδομένων

Δοκιμή	Χρήστης	Δράση	Αποτέλεσμα
χ.σύνδεση.κωδικός	Μη Συνδεδεμένος Χρήστης	Σύνδεση με λάθος κωδικό	Αποτυχία σύνδεσης με μήνυμα σφάλματος
κφ.εγγραφή.όνομα.χρήστη	Φοιτητής Καθηγητής	Εγγραφή με όνομα χρήστη που χρησιμοποιείται	Αποτυχία εγγραφής με μήνυμα σφάλματος
κφ.εγγραφή.email	Φοιτητής Καθηγητής	Εγγραφή με email που χρησιμοποιείται	Αποτυχία εγγραφής με μήνυμα σφάλματος
φ.εγγραφή.αμ	Φοιτητής	Εγγραφή με ΑΜ που χρησιμοποιείται	Αποτυχία εγγραφής με μήνυμα σφάλματος
γ.μαθήματα.όνομα	Γραμματεία	Δημιουργία μαθήματος με όνομα που χρησιμοποιείται	Αποτυχία δημιουργίας μαθήματος με μήνυμα σφάλματος
κ.τμήματα.όνομα	Καθηγητής	Δημιουργία Τμήματος με όνομα που χρησιμοποιείται	Αποτυχία δημιουργίας τμήματος με μήνυμα σφάλματος

Δοκιμές Ασφάλειας

Δοκιμή	Χρήστης	Δράση	Αποτέλεσμα
μσχ.πρόσβαση	Μη Συνδεδεμένος Χρήστης	Εισαγωγή URL διεπαφής Συνδεδεμένου Χρήστη	Απαγόρευση πρόσβασης και ανακατεύθυνση στην διεπαφή Σύνδεσης Χρήστη
μεχ.πρόσβαση	Μη Επικυρωμένος Χρήστης	Εισαγωγή URL διεπαφής Συνδεδεμένου Χρήστη	Απαγόρευση Πρόσβασης
μεκ.πρόσβαση	Μη Επικυρωμένος Καθηγητής	Εισαγωγή URL διεπαφής Συνδεδεμένου Χρήστη	Απαγόρευση Πρόσβασης
φ.πρόσβαση	Φοιτητής	Εισαγωγή URL διεπαφής Καθηγητή ή Γραμματείας	Απαγόρευση Πρόσβασης
φ.αντικείμενα	Φοιτητής	Αλλαγή της τιμής αντικειμένου διεπαφής μέσω παραμέτρου στη URL	Δεν εμφανίζονται δεδομένα ή εμφανίζονται μόνο τα δεδομένα του συνδεδεμένου φοιτητή
φ.δεδομένα	Φοιτητής	Εισαγωγή URL διεπαφής με δεδομένα άλλου Φοιτητή	Δεν εμφανίζονται δεδομένα ή εμφανίζονται μόνο τα δεδομένα του συνδεδεμένου φοιτητή
κ.πρόσβαση	Καθηγητής	Εισαγωγή URL διεπαφής Φοιτητή ή Γραμματείας	Απαγόρευση Πρόσβασης
κ.αντικείμενα	Καθηγητής	Αλλαγή της τιμής αντικειμένου διεπαφής μέσω παραμέτρου στη URL	Δεν εμφανίζονται δεδομένα ή εμφανίζονται μόνο τα δεδομένα του συνδεδεμένου Καθηγητή
κ.δεδομένα	Καθηγητής	Εισαγωγή URL διεπαφής με δεδομένα άλλου Καθηγητή	Δεν εμφανίζονται δεδομένα ή εμφανίζονται μόνο τα δεδομένα του συνδεδεμένου Καθηγητή
δ.πρόσβαση	Γραμματεία	Εισαγωγή URL διεπαφής Φοιτητή ή Καθηγητή	Απαγόρευση Πρόσβασης

Αναφορές

- Aust, D., D'Souza, M. G., Gault, D., Gielis, D., Hartman, R., Hichwa, M., . . . Scott, J. (2011). *Expert Oracle Application Express*.
- Bruegge, B., & Dutoit, A. H. (2000). *Object Oriented Software Engineering Using UML Patterns and Java, 3rd Edition*.
- Fox, T., Fox, J., & Spendolini, S. (2008). *Pro Oracle Application Express 4*.
- Gault, D., Canell, K., D'Souza, M. G., & Hilaire, T. (2013). *Beginning Oracle Application Express 4.2*.
- Oracle. (n.d.). *Oracle Application Express API Reference*. Ανάκτηση από http://docs.oracle.com/cd/E37097_01/doc/doc.42/e35127.pdf
- Oracle. (n.d.). *Oracle Application Express Application Builder User's Guide*. Ανάκτηση από http://docs.oracle.com/cd/E37097_01/doc/doc.42/e35125.pdf
- Oracle. (n.d.). *Oracle Application Express Installation Guide*. Ανάκτηση από http://docs.oracle.com/cd/E37097_01/doc/install.42/e35123.pdf
- Oracle. (n.d.). *Oracle Application Express SQL Workshop Guide*. Ανάκτηση από http://docs.oracle.com/cd/E37097_01/doc/doc.42/e35128.pdf
- Oracle. (n.d.). *Oracle Data Warehousing Guide*. Ανάκτηση από http://docs.oracle.com/cd/E11882_01/server.112/e25554.pdf
- Oracle. (n.d.). *Oracle Database Advanced Application Developer's Guide*. Ανάκτηση από http://docs.oracle.com/cd/E11882_01/appdev.112/e41502.pdf
- Oracle. (n.d.). *Oracle Database Express Edition 2 Day + Application Express Developer's Guide*. Ανάκτηση από http://docs.oracle.com/cd/E17781_01/doc.112/e18644.pdf
- Oracle. (n.d.). *Oracle Database Express Edition 2 Day DBA*. Ανάκτηση από http://docs.oracle.com/cd/E17781_01/server.112/e18804.pdf
- Oracle. (n.d.). *Oracle Database Express Edition 2 Day Developer's Guide*. Ανάκτηση από http://docs.oracle.com/cd/E17781_01/appdev.112/e18147.pdf
- Oracle. (n.d.). *Oracle Database Express Edition Getting Started Guide*. Ανάκτηση από http://docs.oracle.com/cd/E17781_01/admin.112/e18585.pdf
- Oracle. (n.d.). *Oracle Database Express Edition Installation Guide*. Ανάκτηση από http://docs.oracle.com/cd/E17781_01/install.112/e18802.pdf
- Oracle. (n.d.). *Oracle Database PL/SQL Language Reference*. Ανάκτηση από http://docs.oracle.com/cd/E11882_01/appdev.112/e25519.pdf
- Oracle. (n.d.). *Oracle Database PL/SQL Packages and Types Reference*. Ανάκτηση από http://docs.oracle.com/cd/E11882_01/appdev.112/e40758.pdf

Oracle. (n.d.). *Oracle Database SQL Language Reference*. Ανάκτηση από http://docs.oracle.com/cd/E11882_01/server.112/e41084.pdf

Oracle. (n.d.). *Oracle SQL Developer User's Guide*. Ανάκτηση από http://docs.oracle.com/cd/E11882_01/doc.112/e12152.pdf

Pathak, V. (2013). *Oracle APEX Reporting 4.2*.

Rosenberg, S. (2001). *Applying Use Case Driven Object Modeling with UML, An Annotated e-Commerce Example*.

Teory, T., Lightstone, S., Nadeau, T., & Jagadish, H. V. (2011). *Database Modeling and Design, 5th edition: Logical Design*.

Wieggers, K. (2003). *Software Requirements 2nd ed*. Microsoft.

IEEE. (1998). Recommended Practice For Software Requirements Specification. *Std 830*.

Μάντακας, Μ. (2010). *Σχεδιασμός Πληροφοριακών Συστημάτων, έκδοση 4.2*.

Παράρτημα Α. – Κώδικας SQL-DDL δημιουργίας Σχήματος Βάσης Δεδομένων

```
/*
  Εφαρμογή Διαχείρισης Εργαστηρίων Εκπαιδευτηρίου

  Έκδοση: 1.0

  Ανάπτυξη: Σαμαράς Οδυσσέας
*/

/* TABLES */

--Πίνακας για τα κοινά πεδία όλων των χρηστών
drop table users cascade constraints;
create table users(
  id          number          not null,
  username    varchar2(80)    not null,
  email       varchar2(180)   not null,
  password    varchar2(255)   not null, --κρυπτογραφημένη μορφή
  activated   number not null, --επιβεβαίωση μέσω email
  diaxeiristis number not null
);

--Πίνακας με τα στοιχεία των φοιτητών
drop table foititis cascade constraints;
create table foititis(
  id          number not null,
  am          number,
  onoma       varchar2(50) not null,
  epitheto   varchar2(50) not null,
  user_id     number --Χωρίς περιορισμό not null σε περίπτωση που
  εισαχθούν φοιτητές από αρχείο
);

--Πίνακας με τα στοιχεία των καθηγητών
drop table kathigitis cascade constraints;
create table kathigitis(
  id number not null,
  onoma varchar(50) not null,
  epitheto varchar(50) not null,
  epikiromenos number not null, --Πρέπει να επικυρωθεί από άλλον
  καθηγητή
  user_id number not null
);

--Πίνακας που παριστάνει τα μαθήματα
drop table mathima cascade constraints;
```

```

create table mathima(
    id      number      not null,
    thema varchar2(180) not null
);

--Πίνακας που παριστάνει τις διδακτικές περιόδους
drop table periodos cascade constraints;
create table periodos(
    id      number      not null,
    onoma varchar(180)  not null,
    start_date date not null,
    end_date date not null,
    num_weeks number
);

--Πίνακας σχέσης μαθημάτων και διδακτικών περιόδων
drop table mathima_periodos cascade constraints;
create table mathima_periodos(
    id          number not null,
    mathima_id  number not null,
    periodos_id number not null
);

drop table kath_math_per cascade constraints;
create table kath_math_per(
    id number not null,
    kathigitis_id number not null,
    mathima_periodos_id number not null
);

--Πίνακας για τις αίθουσες του εκπαιδευτηρίου
drop table aithousa cascade constraints;
create table aithousa(
    id      number      not null,
    onoma varchar2(20)  not null
);

--Πίνακας με τα τμήματα των μαθημάτων
drop table tmima cascade constraints;
create table tmima(
    id          number      not null,
    onoma       varchar2(80) not null,
    mathima_periodos_id number not null,
    aithousa_id number,
    imera       number,
    time_start  number,
    time_end    number
);

--Πίνακας με τις δηλώσεις μαθημάτων των φοιτητών
drop table epilogi_mathimatos cascade constraints;
create table epilogi_mathimatos(
    id          number not null,
    active      number not null,

```



```

--katoxirosi          number not null,
--status              number not null,
foititis_id           number not null,
mathima_periodos_id   number not null,
tmima_id              number --Αντιστοιχεί στο τμήμα που έχει
ανατεθεί ο φοιτητής
);

--Πίνακας με τις συναντήσεις των μαθημάτων
drop table sinantisi cascade constraints;
create table sinantisi(
    id                  number not null,
    katastasi           number not null,
    programmatismos     number,
    arithmisi           number not null, --Αυξάνεται κατά 1 για
κάθε συνάντηση σε κάθε μάθημα
    mathima_periodos_id number not null
);

--Πίνακας με τις παρακολουθήσεις των φοιτητών
drop table parakolouthisi cascade constraints;
create table parakolouthisi(
    id                  number not null,
    parousia           number, --Δείχνει αν ήταν παρών η απών ο φοιτητής
    sinantisi_id        number not null,
    epil_math_id        number not null,
    tmima_id            number,
    imerominia          date
);

--Πίνακας σχέσης μεταξύ τμημάτων και συναντήσεων ενός μαθήματος
drop table tmima_sinantisi cascade constraints;
create table tmima_sinantisi(
    id number not null,
    sinantisi_id number not null,
    tmima_id number not null,
    pragmatopoihmeni number not null --Δείχνει εάν έγινε η συνάντηση
σε αυτό το τμήμα
);

--Πίνακας με τις δηλώσεις των τμημάτων των φοιτητών
drop table epilogi_tmimatos cascade constraints;
create table epilogi_tmimatos(
    id number not null,
    foititis_id number not null,
    tmima_id number not null,
    protimisi number, --Βαθμός προτίμησης
    logos_adinamias number, --Σε περίπτωση αδυναμίας παρακολούθησης
allos_logos varchar2(255) --Ο φοιτητής μπορεί να δώσει δικό του
λόγο αδυναμίας
);

/* TABLES ΓΙΑ ΛΙΣΤΕΣ ΤΙΜΩΝ */

```

```

--Πίνακας για τις καταστάσεις των συναντήσεων
drop table sinantisi_katastasi cascade constraints;
create table sinantisi_katastasi(
    id          number          not null,
    display_value varchar2(50)  not null
);

--Πίνακας για τους προγραμματισμούς των συναντήσεων
drop table programmatismos cascade constraints;
create table programmatismos(
    id          number          not null,
    display_value varchar2(30)  not null
);

--Πίνακας για τους λόγους αδυναμίας παρακολούθησης τμήματος
drop table logos_adinamias cascade constraints;
create table logos_adinamias(
    id number not null,
    display_value varchar2(50)
);

--Πίνακας για τις ημέρες
drop table imera cascade constraints;
create table imera(
    id number not null,
    display_value varchar2(50)
);

drop table ora cascade constraints;
create table ora(
    id number not null,
    display_value varchar2(50)
);

/* CONSTRAINTS */

--Primary Keys
alter table users          add constraint users_pk
primary key(id);
alter table foititis       add constraint foititis_pk
primary key(id);
alter table kathigitis     add constraint kathigitis_pk
primary key(id);
alter table mathima        add constraint mathima_pk
primary key(id);
alter table periodos       add constraint periodos_pk
primary key(id);
alter table aithousa       add constraint aithousa_pk
primary key(id);
alter table tmima          add constraint tmima_pk
primary key(id);
alter table epilogi_mathimatos add constraint epilogi_mathimatos_pk
primary key(id);

```

```

alter table mathima_periodos    add constraint mathima_periodos_pk
primary key(id);
alter table kath_math_per      add constraint kath_math_per_pk
primary key(id);
alter table sinantisi          add constraint sinantisi_pk
primary key(id);
alter table parakolouthisi     add constraint parakolouthisi_pk
primary key(id);
alter table tmima_sinantisi    add constraint tmima_sinantisi_pk
primary key(id);
alter table epilogi_tmimatos   add constraint epilogi_tmimatos_pk
primary key(id);

alter table sinantisi_katastasi add constraint
sinantisi_katastasi_pk primary key(id);
alter table programmatismos    add constraint programmatismos_pk
primary key(id);
alter table no_yes             add constraint no_yes_pk
primary key(id);
alter table protimisi_epilogis add constraint
protimisi_epilogis_pk primary key(id);
alter table logos_adinamias    add constraint logos_adinamias_pk
primary key(id);
alter table imera              add constraint imera_pk
primary key(id);
alter table ora                add constraint ora_pk
primary key(id);

--Foreign Keys
alter table foititis           add constraint foititis_user_id_fk
foreign key(user_id)          references users(id) on delete
cascade;
alter table kathigitis         add constraint kathigitis_user_id_fk
foreign key(user_id)          references users(id) on delete
cascade;
alter table mathima_periodos    add constraint
mathima_periodos_mathima_id_fk foreign key(mathima_id)
references mathima(id) on delete cascade;
alter table mathima_periodos    add constraint
mathima_periodos_did_per_id_fk foreign key(periodos_id)
references periodos(id) on delete cascade;
alter table kath_math_per      add constraint
kath_math_per_kath_id_fk      foreign key(kathigitis_id)
references kathigitis(id) on delete cascade;
alter table kath_math_per      add constraint
kath_math_per_math_per_fk     foreign key(mathima_periodos_id)
references mathima_periodos(id) on delete cascade;
alter table tmima              add constraint
tmima_mathima_periodos_id_fk  foreign key(mathima_periodos_id)
references mathima_periodos(id) on delete cascade;
alter table tmima              add constraint tmima_aithousa_id_fk
foreign key(aithousa_id)      references aithousa(id) on delete
set null;

```

```

alter table epilogi_mathimatos add constraint
epil_math_foititis_id_fk      foreign key(foititis_id)
references foititis(id) on delete cascade;
alter table epilogi_mathimatos add constraint
epil_math_math_per_id_fk      foreign key(mathima_periodos_id)
references mathima_periodos(id) on delete cascade;
alter table epilogi_mathimatos add constraint epil_math_tmima_id_fk
foreign key(tmima_id)         references tmima(id) on delete set
null;
alter table sinantisi          add constraint
mathima_periodos_id_fk        foreign key(mathima_periodos_id)
references mathima_periodos(id) on delete cascade;
alter table parakolouthisi     add constraint
parakolouthisi_sinantisi_id_fk foreign key(sinantisi_id)
references sinantisi(id) on delete cascade;
alter table parakolouthisi     add constraint
parakolouthisi_epil_math_id_fk foreign key(epil_math_id)
references epilogi_mathimatos(id) on delete cascade;
alter table parakolouthisi     add constraint
parakolouthisi_tmima_id_fk    foreign key(tmima_id)
references tmima(id) on delete set null;
alter table tmima_sinantisi    add constraint
tmima_sinantisi_tmima_id_fk    foreign key(tmima_id)
references tmima(id) on delete cascade;
alter table tmima_sinantisi    add constraint
tmima_sinantisi_sin_id_fk      foreign key(sinantisi_id)
references sinantisi(id) on delete cascade;
alter table epilogi_tmimatos   add constraint
epilogi_tmimatos_foit_id_fk    foreign key(foititis_id)
references foititis(id) on delete cascade;
alter table epilogi_tmimatos   add constraint
epilogi_tmimatos_tmima_id_fk   foreign key(tmima_id)
references tmima(id) on delete cascade;

alter table sinantisi          add constraint
sinantisi_katastasi_fk        foreign key(katastasi)
references sinantisi_katastasi(id);
alter table sinantisi          add constraint sinantisi_prog_fk
foreign key(programmatismos)   references programmatismos(id);

--Uniques
alter table users              add constraint users_username_uq
unique(username);
alter table users              add constraint users_email_uq
unique(email);
alter table foititis           add constraint foititis_am_uq
unique(am);
alter table mathima            add constraint mathima_thema_uq
unique(thema);
alter table periodos           add constraint periodos_onoma_uq
unique(onoma);
alter table aithousa           add constraint aithousa_onoma_uq
unique(onoma);

```

```

alter table tmima            add constraint tmima_onoma_uq
unique(mathima_periodos_id, onoma);
alter table epilogi_mathimatos add constraint epil_math_uq
unique(foititis_id, mathima_periodos_id);
alter table epilogi_tmimatos  add constraint epil_tmima_uq
unique(foititis_id, tmima_id);
alter table mathima_periodos  add constraint mathima_periodos_uq
unique(mathima_id, periodos_id);
alter table kath_math_per     add constraint kath_math_per_uq
unique(kathigitis_id, mathima_periodos_id);
alter table parakolouthisi    add constraint parakolouthisi_uq
unique(sinantisi_id, epil_math_id);

alter table sinantisi_katastasi add constraint
sinantisi_katastasi_uq unique(display_value);
alter table programmatismos    add constraint programmatismos_uq
unique(display_value);
alter table protimisi_epilogis add constraint
protimisi_epilogis_uq unique(display_value);
alter table logos_adinamias    add constraint logos_adinamias_uq
unique(display_value);

--Checks
alter table kathigitis        add constraint
kathigitis_epikiromenos_ck check(epikiromenos in(0, 1));
alter table periodos          add constraint periodos_active_ck
check(active in(0,1));
alter table parakolouthisi    add constraint
parakolouthisi_parousia_ck check(parousia in (0, 1));
alter table tmima_sinantisi   add constraint pragmatopoihmeni_ck
check(pragmatopoihmeni in(0, 1));
/

/* SEQUENCES */
DROP SEQUENCE "USERS_SEQ";
CREATE SEQUENCE "USERS_SEQ" MINVALUE 1 MAXVALUE
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20
NOORDER NOCYCLE;
/
DROP SEQUENCE "KATHIGITIS_SEQ";
CREATE SEQUENCE "KATHIGITIS_SEQ" MINVALUE 1 MAXVALUE
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20
NOORDER NOCYCLE;
/
DROP SEQUENCE "FOITITIS_SEQ";
CREATE SEQUENCE "FOITITIS_SEQ" MINVALUE 1 MAXVALUE
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20
NOORDER NOCYCLE;
/
DROP SEQUENCE "MATHIMA_SEQ";
CREATE SEQUENCE "MATHIMA_SEQ" MINVALUE 1 MAXVALUE
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20
NOORDER NOCYCLE;
/

```

```
DROP SEQUENCE "PERIODOS_SEQ";
CREATE SEQUENCE    "PERIODOS_SEQ" MINVALUE 1 MAXVALUE
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20
NOORDER NOCYCLE;
/
DROP SEQUENCE "AITHOUSA_SEQ";
CREATE SEQUENCE    "AITHOUSA_SEQ" MINVALUE 1 MAXVALUE
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20
NOORDER NOCYCLE;
/
DROP SEQUENCE "TMIMA_SEQ";
CREATE SEQUENCE    "TMIMA_SEQ" MINVALUE 1 MAXVALUE
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20
NOORDER NOCYCLE;
/
DROP SEQUENCE "MATHIMA_PERIODOS_SEQ";
CREATE SEQUENCE    "MATHIMA_PERIODOS_SEQ" MINVALUE 1 MAXVALUE
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20
NOORDER NOCYCLE;
/
DROP SEQUENCE "KATH_MATH_PER_SEQ";
CREATE SEQUENCE    "KATH_MATH_PER_SEQ" MINVALUE 1 MAXVALUE
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20
NOORDER NOCYCLE;
/
DROP SEQUENCE "EPILOGI_MATHIMATOS_SEQ";
CREATE SEQUENCE    "EPILOGI_MATHIMATOS_SEQ" MINVALUE 1 MAXVALUE
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20
NOORDER NOCYCLE;
/
DROP SEQUENCE "SINANTISI_SEQ";
CREATE SEQUENCE    "SINANTISI_SEQ" MINVALUE 1 MAXVALUE
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20
NOORDER NOCYCLE;
/
DROP SEQUENCE "PARAKOLOUTHISI_SEQ";
CREATE SEQUENCE    "PARAKOLOUTHISI_SEQ" MINVALUE 1 MAXVALUE
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20
NOORDER NOCYCLE;
/
DROP SEQUENCE "TMIMA_SINANTISI_SEQ";
CREATE SEQUENCE    "TMIMA_SINANTISI_SEQ" MINVALUE 1 MAXVALUE
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20
NOORDER NOCYCLE;
/
DROP SEQUENCE "EPILOGI_TMIMATOS_SEQ";
CREATE SEQUENCE    "EPILOGI_TMIMATOS_SEQ" MINVALUE 1 MAXVALUE
99999999999999999999999999999999 INCREMENT BY 1 START WITH 1 CACHE 20
NOORDER NOCYCLE;
/

/* TRIGGERS */

create or replace trigger "BI USERS"
```

```

    before insert on "USERS"
    for each row
begin
    if :NEW."ID" is null then
        select "USERS_SEQ".nextval into :NEW."ID" from sys.dual;
    end if;
end;
/

create or replace trigger "BI_KATHIGITIS"
    before insert on "KATHIGITIS"
    for each row
begin
    if :NEW."ID" is null then
        select "KATHIGITIS_SEQ".nextval into :NEW."ID" from sys.dual;
    end if;
end;
/

create or replace trigger "BI_FOITITIS"
    before insert on "FOITITIS"
    for each row
begin
    if :NEW."ID" is null then
        select "FOITITIS_SEQ".nextval into :NEW."ID" from sys.dual;
    end if;
end;
/

create or replace trigger "BI_MATHIMA"
    before insert on "MATHIMA"
    for each row
begin
    if :NEW."ID" is null then
        select "MATHIMA_SEQ".nextval into :NEW."ID" from sys.dual;
    end if;
end;
/

create or replace trigger "BI_PERIODOS"
    before insert on "PERIODOS"
    for each row
begin
    if :NEW."ID" is null then
        select "PERIODOS_SEQ".nextval into :NEW."ID" from sys.dual;
    end if;
end;
/

create or replace trigger "BI_AITHOUSA"
    before insert on "AITHOUSA"
    for each row
begin

```

```

        if :NEW."ID" is null then
            select "AITHOUSA_SEQ".nextval into :NEW."ID" from sys.dual;
        end if;
    end;
/

create or replace trigger "BI_TMIMA"
    before insert on "TMIMA"
    for each row
begin
    if :NEW."ID" is null then
        select "TMIMA_SEQ".nextval into :NEW."ID" from sys.dual;
    end if;
end;
/

create or replace trigger "BI_MATHIMA_PERIODOS"
    before insert on "MATHIMA_PERIODOS"
    for each row
begin
    if :NEW."ID" is null then
        select "MATHIMA_PERIODOS_SEQ".nextval into :NEW."ID" from
sys.dual;
    end if;
end;
/

create or replace trigger "BI_KATH_MATH_PER"
    before insert on "KATH_MATH_PER"
    for each row
begin
    if :NEW."ID" is null then
        select "KATH_MATH_PER_SEQ".nextval into :NEW."ID" from sys.dual;
    end if;
end;
/

create or replace trigger "BI_EPILOGI_MATHIMATOS"
    before insert on "EPILOGI_MATHIMATOS"
    for each row
begin
    if :NEW."ID" is null then
        select "EPILOGI_MATHIMATOS_SEQ".nextval into :NEW."ID" from
sys.dual;
    end if;
end;
/

create or replace trigger "BI_SINANTISI"
    before insert on "SINANTISI"
    for each row
begin
    if :NEW."ID" is null then

```



```

        select "SINANTISI_SEQ".nextval into :NEW."ID" from sys.dual;
    end if;
end;
/

create or replace trigger "BI_PARAKOLOUTHISI"
before insert on "PARAKOLOUTHISI"
for each row
begin
    if :NEW."ID" is null then
        select "PARAKOLOUTHISI_SEQ".nextval into :NEW."ID" from
sys.dual;
    end if;
end;
/

create or replace trigger "BI_TMIMA_SINANTISI"
before insert on "TMIMA_SINANTISI"
for each row
begin
    if :NEW."ID" is null then
        select "TMIMA_SINANTISI_SEQ".nextval into :NEW."ID" from
sys.dual;
    end if;
end;
/

create or replace trigger "BI_EPILOGI_TMIMATOS"
before insert on "EPILOGI_TMIMATOS"
for each row
begin
    if :NEW."ID" is null then
        select "EPILOGI_TMIMATOS_SEQ".nextval into :NEW."ID" from
sys.dual;
    end if;
end;
/

/* DATA ΓΙΑ ΛΙΣΤΕΣ ΤΙΜΩΝ */

insert into sinantisi_katastasi(id, display_value)
values(1, 'Προγραμματισμένη');

insert into sinantisi_katastasi(id, display_value)
values(2, 'Πραγματοποιημένη');

insert into sinantisi_katastasi(id, display_value)
values(3, 'Ακυρωμένη');

insert into programmatismos(id, display_value)
values(1, 'Τακτικός');

insert into programmatismos(id, display_value)
values(2, 'Έκτακτος');

```

```
insert into no_yes(id, display_value)
values(0, 'Όχι');

insert into no_yes(id, display_value)
values(1, 'Ναι');

insert into protimisi_epilogis(id, display_value)
values(1, 'Πρώτη Επιλογή');

insert into protimisi_epilogis(id, display_value)
values(2, 'Εναλλακτική Επιλογή');

insert into protimisi_epilogis(id, display_value)
values(3, 'Αδυναμία Παρακολούθησης');

insert into logos_adinamias(id, display_value)
values(1, 'Εργασία');

insert into logos_adinamias(id, display_value)
values(2, 'Τελευταίο Λεωφορείο');

insert into logos_adinamias(id, display_value)
values(3, 'Προσωπικός Λόγος');

insert into logos_adinamias(id, display_value)
values(4, 'Άλλος Λόγος');

insert into imera(id, display_value)
values(1, 'Δευτέρα');

insert into imera(id, display_value)
values(2, 'Τρίτη');

insert into imera(id, display_value)
values(3, 'Τετάρτη');

insert into imera(id, display_value)
values(4, 'Πέμπτη');

insert into imera(id, display_value)
values(5, 'Παρασκευή');

insert into imera(id, display_value)
values(6, 'Σάββατο');

insert into imera(id, display_value)
values(7, 'Κυριακή');

insert into ora(id, display_value)
values(8, '8:00');

insert into ora(id, display_value)
values(9, '9:00');
```

```
insert into ora(id, display_value)
values(10, '10:00');

insert into ora(id, display_value)
values(11, '11:00');

insert into ora(id, display_value)
values(12, '12:00');

insert into ora(id, display_value)
values(13, '13:00');

insert into ora(id, display_value)
values(14, '14:00');

insert into ora(id, display_value)
values(15, '15:00');

insert into ora(id, display_value)
values(16, '16:00');

insert into ora(id, display_value)
values(17, '17:00');

insert into ora(id, display_value)
values(18, '18:00');

insert into ora(id, display_value)
values(19, '19:00');

insert into ora(id, display_value)
values(20, '20:00');

commit;
/
```

Παράρτημα Β. – Κώδικας Πακέτων του Σχήματος Βάσης Δεδομένων

```
create or replace package parousies as

    --Δημιουργεί νέα συνάντηση σε μάθημα
    procedure create_sinantisi(p_mathima_periodos_id in number,
    p_katastasi in number, p_programmatismos in number);

    --Θέτει φοιτητή ως παρών σε συνάντηση μαθήματος
    procedure set_parousia(p_epil_math_id in number, p_sinantisi_id in
    number);

    --Θέτει το τμήμα παρακολούθησης φοιτητή
    procedure set_tmima_parakolouthisis(p_epil_math_id in number,
    p_sinantisi_id in number, p_tmima_id in number);

end parousies;
/

create or replace package body parousies as

    procedure create_sinantisi(p_mathima_periodos_id in number,
    p_katastasi in number, p_programmatismos in number)
    is
        l_arithmisi sinantisi.arithmisi%type;
        l_sinantisi_id sinantisi.id%type;
        cursor epiloges_mathimatos is
            select id, tmima_id
            from epilogi_mathimatos
            where mathima_periodos_id = p_mathima_periodos_id;
        cursor tmimata_mathimatos is
            select id
            from tmima
            where mathima_periodos_id = p_mathima_periodos_id;
        begin
            select count(arithmisi) into l_arithmisi
            from sinantisi
            where mathima_periodos_id = p_mathima_periodos_id;
            l_arithmisi := l_arithmisi + 1;
            select sinantisi_seq.nextval into l_sinantisi_id from sys.dual;
            insert into sinantisi(id, katastasi, programmatismos, arithmisi,
            mathima_periodos_id)
                values(l_sinantisi_id, p_katastasi, p_programmatismos,
            l_arithmisi, p_mathima_periodos_id);
            for epil_math in epiloges_mathimatos
            loop
                insert into parakolouthisi(parousia, sinantisi_id,
            epil_math_id, tmima_id)
                    values(0, l_sinantisi_id, epil_math.id, epil_math.tmima_id);
```

```

        end loop;
        for tmima_mathimatos in tmimata_mathimatos
        loop
            insert into tmima_sinantisi(sinantisi_id, tmima_id,
pragmatopoihmeni)
                values(l_sinantisi_id, tmima_mathimatos.id, 0);
        end loop;
    end create_sinantisi;

    procedure set_parousia(p_epil_math_id in number, p_sinantisi_id in
number)
    is
    begin
        update parakolouthisi
            set parousia = 1
            where epil_math_id = p_epil_math_id
            and sinantisi_id = p_sinantisi_id;
    end set_parousia;

    procedure set_tmima_parakolouthisis(p_epil_math_id in number,
p_sinantisi_id in number, p_tmima_id in number)
    is
    begin
        update parakolouthisi
            set tmima_id = p_tmima_id
            where epil_math_id = p_epil_math_id
            and sinantisi_id = p_sinantisi_id;
    end set_tmima_parakolouthisis;

end parousies;
/

```

```

create or replace package authentication as

```

```

    --Κρυπτογραφεί τον κωδικό
    function hash_password(
        p_username in varchar2,
        p_password in varchar2
    ) return varchar2;

    --Αυθεντικοποιεί τον χρήστη
        function authenticate_user(
            p_username in varchar2,
            p_password in varchar2
        ) return boolean;

    --Επιστρέφει το id του χρήστη με βάση της μεταβλητής APEX
v('APP_USER')
    function get_user_id return number;
    --Επιστρέφει το id του φοιτητή με βάση το user_id
    function get_foititis_id return number;
    --Επιστρέφει το id του καθηγητή με βάση το user_id
    function get_kathigitis_id return number;

```

```

--Authorizations
function is_diaxeiristis return boolean;
function is_activated_user return boolean;
function is_not_activated_user return boolean;
function is_not_activated_foititis return boolean;
function is_not_activated_kathigitis return boolean;
function is_kathigitis return boolean;
function is_unconfirmed_kathigitis return boolean;
function is_foititis return boolean;

end authentication;
/

create or replace package body authentication as

function hash_password(
    p_username in varchar2,
    p_password in varchar2
) return varchar2
as
    l_password varchar2(255);
    l_salt      varchar2(255) :=
'2345USFG0JN2T3HW89EFG0BN23R5SDFGAKL';
begin
    l_password := utl_raw.cast_to_raw(
        dbms_obfuscation_toolkit.md5(
            input_string => p_password || substr(l_salt, 4, 14) ||
                upper(p_username) || substr(l_salt, 5, 10)
        )
    );
    return l_password;
end hash_password;

function authenticate_user(
    p_username in varchar2,
    p_password in varchar2
) return boolean
is
    l_username          users.username%type :=
upper(p_username);
    l_password          users.password%type;
    l_hashed_password   varchar2(1000);
    l_count             number;
begin
    --Έλεγχος εάν υπάρχει το username
    select count(*)
    into l_count
    from users
    where upper(username) = l_username;

    --Εάν υπάρχει το username τότε κρυπτογραφούμε την παράμετρο
    p_password
    if l_count > 0 then

```

```

        l_hashed_password := hash_password(l_username,
p_password);

    --Ανάκτηση του αποθηκευμένου password για το username
    select password
        into l_password
        from users
        where upper(username) = l_username;

    --Σύγκριση του αποθηκευμένου password με την παράμετρο
    if l_hashed_password = l_password then
        return true;
    else
        return false;
    end if;

    --Εάν δεν υπάρχει το username
    else
        return false;
    end if;
    return true;
end authenticate_user;

function get_user_id return number
is
    l_user_id users.id%type;
begin
    select id
        into l_user_id
        from users
        where upper(username) = upper(V('APP_USER'));
    return l_user_id;
end get_user_id;

function get_foititis_id return number
is
    l_foititis_id foititis.id%type;
begin
    select id
        into l_foititis_id
        from foititis
        where user_id = get_user_id;
    return l_foititis_id;
end get_foititis_id;

function get_kathigitis_id return number
is
    l_kathigitis_id kathigitis.id%type;
begin
    select id
        into l_kathigitis_id
        from kathigitis
        where user_id = get_user_id;
    return l_kathigitis_id;
end get_kathigitis_id;

```

```

function is_diaxeiristis return boolean
is
    l_user_id number;
begin
    select id
        into l_user_id
        from users
        where id = get_user_id
        and diaxeiristis = 1;
    return true;
    exception when no_data_found then return false;
end is_diaxeiristis;

```

```

function is_activated_user return boolean
is
    l_user_id number;
begin
    select id
        into l_user_id
        from users
        where id = get_user_id
        and activated = 1;
    return true;
    exception when no_data_found then return false;
end is_activated_user;

```

```

function is_not_activated_user return boolean
is
    l_user_id number;
begin
    select id
        into l_user_id
        from users
        where id = get_user_id
        and activated = 0;
    return true;
    exception when no_data_found then return false;
end is_not_activated_user;

```

```

function is_kathigitis return boolean is
    l_user_id number;
    l_kathigitis_id number;
begin
    l_user_id := get_user_id;
    select k.id
        into l_kathigitis_id
        from kathigitis k, users u
        where k.user_id = l_user_id
        and u.id = k.user_id
        and k.epikiromenos = 1
        and u.activated = 1;
    return true;
    exception when no_data_found then return false;

```



```

end is_kathigitis;

function is_unconfirmed_kathigitis return boolean is
    l_user_id number;
    l_kathigitis_id number;
begin
    l_user_id := get_user_id;

    select k.id
        into l_kathigitis_id
        from kathigitis k, users u
        where k.user_id = l_user_id
        and u.id = k.user_id
        and u.activated = 1
        and epikiromenos = 0;
    return true;
    exception when no_data_found then return false;
end is_unconfirmed_kathigitis;

function is_foititis return boolean is
    l_user_id number;
    l_foititis_id number;
begin
    l_user_id := get_user_id;
    select f.id
        into l_foititis_id
        from foititis f, users u
        where f.user_id = l_user_id
        and f.user_id = u.id
        and u.activated = 1;
    return true;
    exception when no_data_found then return false;
end is_foititis;

function is_not_activated_foititis return boolean is
    l_user_id number;
    l_foititis_id number;
begin
    l_user_id := get_user_id;
    select f.id
        into l_foititis_id
        from foititis f, users u
        where f.user_id = l_user_id
        and f.user_id = u.id
        and u.activated = 0;
    return true;
    exception when no_data_found then return false;
end is_not_activated_foititis;

function is_not_activated_kathigitis return boolean is
    l_user_id number;
    l_kathigitis_id number;
begin
    l_user_id := get_user_id;

```

```

select k.id
  into l_kathigitis_id
  from kathigitis k, users u
  where k.user_id = l_user_id
  and u.id = k.user_id
  and k.epikiromenos = 1
  and u.activated = 0;
return true;
exception when no_data_found then return false;
end is_not_activated_kathigitis;

end authentication;
/

create or replace package registration as

  --Έλεγχοι για τους περιορισμούς μοναδικότητας
  function username_exists(p_username in varchar2) return boolean;
  function email_exists(p_email in varchar2) return boolean;
  function am_exists(p_am in number) return boolean;
  function password_exists(p_password in varchar2) return boolean;

  --Εγγραφεί τον χρήστη και επιστρέφει το id
  function register_user(
    p_username in varchar2,
    p_password in varchar2,
    p_email in varchar2
  ) return number;

  --Εγγράφει τον φοιτητή
  procedure register_foititis(
    p_username in varchar2,
    p_password in varchar2,
    p_email in varchar2,
    p_onoma in varchar2,
    p_epitheto in varchar2,
    p_am in number
  );

  --Εγγραφεί τον καθηγητή
  procedure register_kathigitis(
    p_username in varchar2,
    p_password in varchar2,
    p_email in varchar2,
    p_onoma in varchar2,
    p_epitheto in varchar2
  );

  --Επικυρώνει τον χρήστη
  procedure activate_user( p_hashed_password in varchar2);

```

```

--Δημιουργεί και αποστέλλει νέο κωδικό
procedure email_new_password(p_email in varchar2);

--Στέλνει σύνδεσμο επιβεβαίωσης
procedure send_activation_link(p_user_id in number);

--Επιστρέφει το email του χρήστη με βάση την μεταβλητή Apex
v('APP_USER')
function get_user_email return varchar2;
--Επιστρέφει το AM του φοιτητή με βάση το user_id
function get_foititis_am return number;

end registration;
/

create or replace package body registration as

function username_exists(p_username in varchar2) return boolean
as
    l_id users.id%type;
begin
    select id
    into l_id
    from users
    where username = p_username;
    return true;
exception when no_data_found
    then return false;
end username_exists;

function email_exists(p_email in varchar2) return boolean
as
    l_id users.id%type;
begin
    select id
    into l_id
    from users
    where email = p_email;
    return true;
exception when no_data_found
    then return false;
end email_exists;

function am_exists(p_am in number) return boolean
as
    l_user_id foititis.user_id%type;
begin
    select user_id
    into l_user_id
    from foititis
    where am = p_am;
    return true;

```

```

        exception when no_data_found
            then return false;
    end am_exists;

function password_exists(p_password in varchar2) return boolean
is
    l_password users.password%type;
begin
    select password
        into l_password
        from users
        where password = p_password;
    return true;
    exception when no_data_found
        then return false;
end password_exists;

procedure send_activation_link(p_user_id in number)
is
    l_hashed_password users.password%type;
    l_email users.email%type;
    l_username users.username%type;
    l_host varchar2(500);
    l_body varchar2(1000);
    l_body_html varchar2(4000);
    l_activation_link varchar2(500);
    l_email_host varchar2(500);
    crlf VARCHAR2(2) := chr(13)||chr(10);
begin
    select email, username, password
        into l_email, l_username, l_hashed_password
        from users
        where id = p_user_id;

    l_host := owa_util.get_cgi_env('HTTP_HOST');
    l_activation_link := 'http://'||l_host||'/apex/'

    ||'f?p=112:101:'||':'||':NO'||'::'||'P101_ACTIVATE:'||l_hashed_passw
ord;

    l_body := 'Για να δείτε αυτό το email χρησιμοποιήστε λογισμικό
που να υποστηρίζει HTML'||crlf;

    l_body_html := '<html><head></head><body><h3>Τα στοιχεία του
λογαριασμού σας είναι τα εξής</h3>'||crlf;
    l_body_html := l_body_html||'<p>'||'Όνομα Χρήστη:
'||l_username||'</p>'||crlf;
    l_body_html := l_body_html||'<p>'||'Ηλεκτρονικό Ταχυδρομείο:
'||l_email||'</p>'||crlf;

    l_body_html := l_body_html||'<p>'||'Για να ενεργοποιήσετε τον
λογαριασμό σας κάντε κλικ εδώ:'||crlf
    ||'<a
href="'||l_activation_link||'">'||l_activation_link||'</a></p>';

```

```

l_body_html := l_body_html || '</body></html>' || crlf;

l_email_host := substr(owa_util.get_cgi_env('HTTP_HOST'), 1,
instr(owa_util.get_cgi_env('HTTP_HOST'), ':') -1);

apex_mail.send(
    p_to => l_email,
    p_from => 'admin@'||l_email_host,
    p_body => l_body,
    p_body_html => l_body_html,
    p_subj => 'Ενεργοποίηση Λογαριασμού - Εφαρμογή Διαχείρισης
Εργαστηρίων Εκπαιδευτηρίου'
);

end send_activation_link;

function register_user(
    p_username in varchar2,
    p_password in varchar2,
    p_email in varchar2
) return number
as
    v_id users.id%type;
    v_hashed_password users.password%type;
begin
    select users_seq.nextval into v_id from sys.dual;
    v_hashed_password := authentication.hash_password(p_username,
p_password);
    insert into users(
        id,
        username,
        password,
        email,
        activated,
        diaxeiristis
    ) values (
        v_id,
        p_username,
        v_hashed_password,
        p_email,
        0,
        0
    );

    send_activation_link(v_id);

    return v_id;
end register_user;

procedure register_foititis(
    p_username in varchar2,
    p_password in varchar2,
    p_email in varchar2,
    p_onoma in varchar2,

```

```

        p_epitheto in varchar2,
        p_am in number
    )
as
    v_user_id foititis.user_id%type;
begin
    v_user_id := register_user(
        p_username,
        p_password,
        p_email
    );
    insert into foititis(
        am,
        user_id,
        onoma,
        epitheto
    ) values (
        p_am,
        v_user_id,
        p_onoma,
        p_epitheto
    );
end register_foititis;

procedure register_kathigitis(
    p_username in varchar2,
    p_password in varchar2,
    p_email in varchar2,
    p_onoma in varchar2,
    p_epitheto in varchar2
)
as
    v_user_id kathigitis.user_id%type;
begin
    v_user_id := register_user(
        p_username,
        p_password,
        p_email
    );
    insert into kathigitis(
        onoma,
        epitheto,
        user_id,
        epikiromenos
    ) values (
        p_onoma,
        p_epitheto,
        v_user_id,
        0
    );
end register_kathigitis;

procedure activate_user( p_hashed_password in varchar2)
is

```

```

begin
    update users
        set activated = 1
        where password = p_hashed_password;
end;

procedure email_new_password(p_email in varchar2)
is
    l_username users.username%type;
    l_password users.password%type;
    l_hashed_password users.password%type;
begin
    select username
    into l_username
    from users
    where upper(users.email) = upper(p_email);
    --Δημιουργία νέου password
    select dbms_random.string('A', 8)
    into l_password
    from sys.dual;

    --Κρυπτογράφηση του password
    l_hashed_password := authentication.hash_password(l_username,
l_password);

    --Ενημέρωση του password
    update users
        set password = l_hashed_password
        where upper(users.email) = upper(p_email);

    APEX_MAIL.SEND(
        p_to => p_email,
        p_from => 'admin@edee.teiep.gr',
        p_subj => 'Αλλαγή κωδικού στην Εφαρμογή Διαχείρισης
Εργαστηρίων Εκπαιδευτηρίου' ,
        p_body => 'Νέα στοιχεία σύνδεσης '
        || 'Όνομα Χρήστη: ' || l_username
        || 'Κωδικός: ' || l_password
    );
end email_new_password;

function get_user_email return varchar2
is
    l_user_email users.email%type;
begin
    select email
    into l_user_email
    from users
    where id = authentication.get_user_id;
    return l_user_email;
end get_user_email;

function get_foititis_am return number
is

```

```

        l_foititis_am foititis.am%type;
begin
    select am
        into l_foititis_am
        from foititis
        where id = authentication.get_foititis_id;
    return l_foititis_am;
end get_foititis_am;

end registration;
/

create or replace package administration as

    --Δημιουργεί νέα περίοδο
    procedure create_periodos(p_onoma in varchar2, p_start_date in
date, p_end_date in date);

    --Δημιουργεί νέο μάθημα
    procedure create_mathima(p_thema in varchar2);
    --Δημιουργεί νέο μάθημα και επιστρέφει το id
    function create_mathima_f(p_thema in varchar2) return number;
    --Διαγράφει ένα μάθημα
    procedure delete_mathima(p_id in varchar2);

    --Αναθέτει καθηγητή σε μάθημα
    procedure create_kath_math_per(p_kathigitis_id in number,
p_mathima_periodos_id in number);
    --Διαγράφει ανάθεση καθηγητή σε μάθημα
    procedure delete_kath_math_per(p_kath_math_per_id in number);

    --Προσθέτει μάθημα σε διδακτική περίοδο
    procedure create_mathima_periodos(p_periodos_id in number,
p_mathima_id in number);
    --Προσθέτει μάθημα σε διδακτική περίοδο και επιστρέφει το id
    function create_mathima_periodos_f(p_periodos_id in number,
p_mathima_id in number) return number;
    --Αφαιρεί μάθημα από διδακτική περίοδο
    procedure remove_mathima_periodos(p_mathima_periodos_id in
number);

    --Δημιουργεί τμήμα μαθήματος
    procedure create_tmima(
        p_onoma in varchar2,
        p_mathima_periodos_id in number,
        p_imera in number,
        p_time_start in number,
        p_time_end in number,
        p_aithousa_id in number
    );

    --Προσθέτει φοιτητή σε τμήμα

```



```

    procedure epikirose_dilos_i_tmimatos(p_epilogi_mathimatos_id in
number, p_tmima_id in number);
    --Αφαίρει φοιτητή από τμήμα
    procedure afairese_apo_tmima(p_epilogi_mathimatos_id in number);

end administration;
/

```

```

create or replace package body administration as

```

```

    procedure create_periodos(p_onoma in varchar2, p_start_date in
date, p_end_date in date)
    is
        l_start_week number;
        l_end_week number;
        l_start_year number;
        l_end_year number;
        l_num_weeks number;
    begin
        l_start_week := to_number(to_char(p_start_date, 'WW'));
        l_end_week := to_number(to_char(p_end_date, 'WW'));
        l_start_year := to_number(to_char(p_start_date, 'YYYY'));
        l_end_year := to_number(to_char(p_end_date, 'YYYY'));
        l_num_weeks := (l_end_week - l_start_week + 52 * (l_end_year -
l_start_year)) - 4;
        insert into periodos(onoma, start_date, end_date, num_weeks)
            values(p_onoma, p_start_date, p_end_date, l_num_weeks);
    end create_periodos;

```

```

    procedure create_mathima(p_thema in varchar2)
    is
    begin
        insert into mathima(thema)
            values(p_thema);
    end create_mathima;

```

```

    function create_mathima_f(p_thema in varchar2) return number
    is
        l_mathima_id mathima.id%type;
    begin
        select mathima_seq.nextval into l_mathima_id from sys.dual;

        insert into mathima(id, thema)
            values(l_mathima_id, p_thema);

        return l_mathima_id;
    end create_mathima_f;

```

```

    procedure delete_mathima(p_id in varchar2)
    is
    begin
        delete from mathima
            where id = p_id;
    end;

```

```

    procedure create_kath_math_per(p_kathigitis_id in number,
p_mathima_periodos_id in number)
    is
    begin
        insert into kath_math_per(kathigitis_id, mathima_periodos_id)
            values(p_kathigitis_id, p_mathima_periodos_id);
    end create_kath_math_per;

    procedure delete_kath_math_per(p_kath_math_per_id in number)
    is
    begin
        delete from kath_math_per
            where id = p_kath_math_per_id;
    end delete_kath_math_per;

    procedure create_mathima_periodos(p_periodos_id in number,
p_mathima_id in number)
    is
        l_mathima_periodos_id mathima_periodos.id%type;
        l_num_sinantiseis number;
    begin
        select mathima_periodos_seq.nextval into l_mathima_periodos_id
from sys.dual;

        select num_weeks
        into l_num_sinantiseis
        from periodos
        where id = p_periodos_id;

        insert into mathima_periodos(id, periodos_id, mathima_id)
            values(l_mathima_periodos_id, p_periodos_id, p_mathima_id);

        for i in 1..l_num_sinantiseis
        loop
            parousies.create_sinantisi(l_mathima_periodos_id, 1, 1);
        end loop;
    end create_mathima_periodos;

    function create_mathima_periodos_f(p_periodos_id in number,
p_mathima_id in number) return number
    is
        l_mathima_periodos_id mathima_periodos.id%type;
        l_num_sinantiseis number;
    begin
        select mathima_periodos_seq.nextval into l_mathima_periodos_id
from sys.dual;

        select num_weeks
        into l_num_sinantiseis
        from periodos
        where id = p_periodos_id;

        insert into mathima_periodos(id, periodos_id, mathima_id)

```

```

        values(l_mathima_periodos_id, p_periodos_id, p_mathima_id);

    for i in 1..l_num_sinantiseis
    loop
        parousies.create_sinantisi(l_mathima_periodos_id, 1, 1);
    end loop;

    return l_mathima_periodos_id;
end create_mathima_periodos_f;

procedure remove_mathima_periodos(p_mathima_periodos_id in number)
is
begin
    delete
    from mathima_periodos
    where id = p_mathima_periodos_id;
end remove_mathima_periodos;

procedure create_tmima(
    p_onoma in varchar2,
    p_mathima_periodos_id in number,
    p_imera in number,
    p_time_start in number,
    p_time_end in number,
    p_aithousa_id in number
)
is
    l_tmima_id tmima.id%type;

    cursor epiloges_mathimaton is
        select foititis_id
        from epilogi_mathimatos
        where mathima_periodos_id = p_mathima_periodos_id;

begin
    select tmima_seq.nextval into l_tmima_id from sys.dual;

    insert into tmima(id, onoma, mathima_periodos_id, imera,
time_start, time_end, aithousa_id)
        values(l_tmima_id, p_onoma, p_mathima_periodos_id, p_imera,
p_time_start, p_time_end, p_aithousa_id);

    for i in epiloges_mathimaton
    loop
        insert into epilogi_tmimatos(foititis_id, tmima_id)
            values(i.foititis_id, l_tmima_id);
    end loop;

end create_tmima;

procedure epikirose_dilos_i_tmimatos(
    p_epilogi_mathimatos_id in number,
    p_tmima_id in number
)

```

```

is
begin
    update epilogi_mathimatos
    set tmima_id = p_tmima_id
    where id = p_epilogi_mathimatos_id;
end epikirose_dilos_i_tmimatos;

procedure afairese_apo_tmima(p_epilogi_mathimatos_id in number)
is
begin
    update epilogi_mathimatos
    set tmima_id = null
    where id = p_epilogi_mathimatos_id;
end afairese_apo_tmima;

end administration;
/

create or replace package diloseis as

    --Δημιουργεί δήλωση μαθήματος φοιτητή
    procedure dilose_mathima(p_foititis_id in number,
p_mathima_periodos_id in number);

    --Δημιουργεί δήλωση τμήματος φοιτητή
    procedure dilose_tmima(
        p_foititis_id in number,
        p_tmima_id in number,
        p_protimisi in number,
        p_logos_adinamias in number,
        p_allos_logos in varchar2,
        p_adinamia in number
    );

end diloseis;
/

create or replace package body diloseis as

    procedure dilose_mathima(
        p_foititis_id in number,
        p_mathima_periodos_id in number
    )
    is
        l_epil_math_id epilogi_mathimatos.id%type;

        cursor sinantiseis is
            select id
            from sinantisi
            where mathima_periodos_id = p_mathima_periodos_id;

        cursor tmimata is
            select id
            from tmima

```

```

        where mathima_periodos_id = p_mathima_periodos_id;
    begin
        select epilogi_mathimatos_seq.nextval into l_epil_math_id from
sys.dual;

        insert into epilogi_mathimatos(id, foititis_id,
mathima_periodos_id, active)
            values(l_epil_math_id, p_foititis_id, p_mathima_periodos_id,
1);

        for i in sinantiseis
        loop
            insert into parakolouthisi(parousia, sinantisi_id,
epil_math_id)
                values(0, i.id, l_epil_math_id);
        end loop;

        for i in tmimata
        loop
            insert into epilogi_tmimatos(foititis_id, tmima_id)
                values(authentication.get_foititis_id, i.id);
        end loop;

    end dilose_mathima;

procedure dilose_tmima(
    p_foititis_id in number,
    p_tmima_id in number,
    p_protimisi in number,
    p_logos_adinamias in number,
    p_allos_logos in varchar2,
    p_adinamia in number
)
is
begin
    insert into epilogi_tmimatos(
        foititis_id,
        tmima_id,
        protimisi,
        logos_adinamias,
        allos_logos,
        adinamia
    ) values (
        p_foititis_id,
        p_tmima_id,
        p_protimisi,
        p_logos_adinamias,
        p_allos_logos,
        p_adinamia
    );
end dilose_tmima;

end diloseis;
/

```