

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή **στα Πληροφοριακά και Επικοινωνιακά Συστήματα**



**Σχεδιασμός και υλοποίηση πληροφοριακού συστήματος
γραμματειακής υποστήριξης προγραμμάτων σπουδών
Τριτοβάθμιας Εκπαίδευσης**

Βασίλειος Τσουκαλάς

**Επιβλέπων Καθηγητής
Χρήστος Γκουμόπουλος**

Σεπτέμβριος 2015

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Σχεδιασμός και υλοποίηση πληροφοριακού συστήματος
γραμματειακής υποστήριξης προγραμμάτων σπουδών
Τριτοβάθμιας Εκπαίδευσης

Βασίλειος Τσουκαλάς

Επιβλέπων Καθηγητής
Χρήστος Γκουμόπουλος

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε
προς μερική εκπλήρωση των απαιτήσεων για απόκτηση

μεταπτυχιακού τίτλου σπουδών
στα Πληροφοριακά Συστήματα

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών
του Ανοικτού Πανεπιστημίου Κύπρου

Σεπτέμβριος 2015

Περίληψη

Στόχος της μεταπτυχιακής διατριβής είναι να υποστηρίξει την εκτέλεση των προγραμμάτων σπουδών των Ελληνικών εκπαιδευτικών ιδρυμάτων τριτοβάθμιας εκπαίδευσης με πιο οικονομικό και ποιοτικό τρόπο σε σχέση με σήμερα. Κάθε τριτοβάθμιο εκπαιδευτικό ίδρυμα διαθέτει πληροφοριακό σύστημα για την εν μέρει υποστήριξη των προγραμμάτων σπουδών του. Είναι γεγονός πως τα περισσότερα εκπαιδευτικά ιδρύματα διαθέτουν το ίδιο πληροφοριακό σύστημα και η τάση είναι πως τα υπάρχοντα πληροφοριακά συστήματα παραμένουν. Υπάρχουν σημαντικές λειτουργίες οι οποίες δεν υποστηρίζονται από τα υπάρχοντα συστήματα υποστήριξης προγραμμάτων σπουδών όπως είναι οι δηλώσεις ομάδων παρακολούθησης μαθημάτων και κατανομή φοιτητών σε αυτές, η αυτόματη δημιουργία εβδομαδιαίων ωρολογίων προγραμμάτων κ.α. Αρκετά ιδρύματα ή μεμονωμένα τμήματα ιδρυμάτων έχουν κάνει προσπάθειες να χρησιμοποιήσουν άλλα συστήματα τα όποια θα υποστηρίζουν τις παραπάνω –επιπλέον– σημαντικές λειτουργίες. Ωστόσο για διάφορους λόγους: ασφάλειας, τεχνικής φύσης, συνεργασίας, εμπιστοσύνης και συντονισμού αυτά τα συστήματα δε μπορούν να συνδεθούν με το υφιστάμενο γραμματειακό σύστημα υποστήριξης προγραμμάτων σπουδών, με αποτέλεσμα να υπάρχει πρόβλημα στην ακεραιότητα των δεδομένων που διαχειρίζονται. Λύση σε αυτά τα προβλήματα δίνουν τα Business Process Management systems (BPMs) τα οποία σε συνδυασμό με τις τεχνολογίες Java EE και REST επιτρέπουν την δημιουργία ενός «πυρήνα» λογισμικών που θα περιλαμβάνει νέες εφαρμογές, υπηρεσίες και βάσεις δεδομένων που θα είναι ενοποιημένες σε πραγματικό χρόνο με υπάρχουσες εφαρμογές, υπηρεσίες και βάσεις δεδομένων. Οι τεχνολογίες αυτές εξασφαλίζουν τη διαλειτουργικότητα των εφαρμογών-υπηρεσιών. Πάνω σε αυτή τη λογική υλοποιήθηκαν έξι νέα υποσυστήματα (που συνεργάζονται μεταξύ τους) με μια νέα βάση δεδομένων ενώ διασυνδέονται με αρκετά υπάρχοντα υποσυστήματα και υπηρεσίες του ΤΕΙ ΑΜΘ και αντιμετωπίζουν σε σημαντικό βαθμό τις επιπλέον λειτουργίες που απαιτούνται για να θεωρηθεί αποδοτική η εκτέλεση των προγραμμάτων σπουδών. Ερευνήσαμε τις μεθοδολογίες και τις τεχνολογίες δημιουργίας ωρολογίων προγραμμάτων, ξεχωρίσαμε το ελεύθερο λογισμικό FET και δημιουργήσαμε ένα υποσύστημα που το ενοποιεί με τα υπόλοιπα υποσυστήματα. Στον «πυρήνα» λογισμικών κατασκευάσαμε και ενσωματώσαμε εφαρμογή Android ώστε να μπορούν να κάνουν δηλώσεις ομάδων παρακολούθησης οι φοιτητές από τις κινητές συσκευές τους. Με βάση τις υποθέσεις και τις στατιστικές διαδικασίες που παρουσιάζονται στο κεφάλαιο της ερευνητικής πρότασης αναμένουμε σημαντική αύξηση της απόδοσης με ποιοτικό και οικονομικό τρόπο, αλλά τα οριστικά στατιστικά αποτελέσματα των μετρήσεων –όπως περιγράφεται στο πείραμα– θα

προκύψουν μετά από ένα χρόνο λειτουργίας του συστήματος που υλοποιήσαμε. Δεν υπήρξε το περιθώριο εφαρμογής του συστήματος λόγω χρονικών συγκυριών. Ωστόσο στο πλάνο αξιολόγησης προδιαγράψαμε τη διαδικασία συγκέντρωσης των στοιχείων για την τροφοδότηση των στατιστικών διαδικασιών και τον τρόπο ερμηνείας των αποτελεσμάτων που θα προκύψουν. Τέλος για την αξιολόγηση του συστήματος που υλοποιήσαμε αξιοποιήσαμε μια ομάδα χρηστών συζητήσαμε μαζί τους, διατυπώσαμε ερωτήσεις, ενώ από τις απαντήσεις τους προέκυψαν χρήσιμα θετικά και αρνητικά συμπεράσματα.

Λόγω του μεγέθους της μεταπτυχιακής διατριβής και προς διευκόλυνση της μελέτης από τους αναγνώστες έχουμε συγγράφει στην επόμενη παράγραφο εκτεταμένη περίληψη του έργου μας.

Summary

The aim of the thesis is to support the implementation of study programs in Greek higher education institutions in a more economic and qualitative way with respect to the current situation. Currently, each higher education institution has an information system to partially support its study programs. In practice, most educational institutions have identical information systems and the trend is that existing IT systems remain. However, there are important operations which are not supported by existing study programs support systems such as the group's courses students allocation, the automatic creation of weekly timetables, etc. Many institutions or individual departments have made efforts to use other systems supporting the above -extra- important functionality. However for various safety or technical in nature reasons or other human cooperation, trust and coordination reasons these systems cannot be connected to the existing main management system, resulting in data integrity problems. Solution to these problems give the Business Process Management systems (BPMs) which in combination with technologies such as Java EE and REST allow the creation of a software "core", including new applications, services and databases that are integrated in real time with existing applications, services and databases. These technologies ensure the interoperability of different applications and services. Following this approach we implemented six new application modules and a new database that integrates with the existing subsystems and services of the Technological Educational Institute of Eastern Macedonia and Thrace and confronts in a significant degree the additional functionality required. We investigated implementation technologies and methodologies for the timetables' generation problem and we choose the FET open software. Based on the FET engine we created a subsystem that integrates with the other subsystems. We

developed an Android application so that students can allocate themselves in groups of courses and we integrated this application with the software “core” mentioned previously. According to the assumptions and the statistical procedures which are presented in the research proposal chapter we expect significant increase of performance in a qualitative and economic way, but the final statistical results of the measurements -as described in the experiment; would arise after a year of operation of the new system in a production environment. The system could not be evaluated in the context of this thesis due to time constraints. However the evaluation plan has been outlined; in addition, the concentration process of data to supply the statistical procedures and how to interpret the results obtained have been specified. Finally, for an informal assessment of the system developed, we collaborated with a user group by discussing with them design and implementation issues and asking questions, while from the answers emerged useful positive and negative feedback.

Εκτεταμένη περίληψη

Για την βελτίωση της εκτέλεσης των προγραμμάτων σπουδών διατυπώσαμε ερευνητική πρόταση τριών υποθέσεων, εστίασαμε στις δυνατότητες αυτόματης δημιουργίας ωρολογίων προγραμμάτων και πραγματοποίησης ηλεκτρονικών δηλώσεων ομάδων παρακολούθησης μαθημάτων από τους φοιτητές, επιλέξαμε ποιοτικά εργαλεία υλοποίησης και υλοποιήσαμε σύστημα που θα υποστηρίζει αυτές τις δυνατότητες.

Οι τρεις υποθέσεις της ερευνητικής πρότασης φιλοδοξούν στην ελαχιστοποίηση του χρόνου δημιουργίας ωρολογίων προγραμμάτων ακαδημαϊκών περιόδων, στην μείωση των παραπόνων από τους φοιτητές που δηλώνουν ομάδες παρακολούθησης μαθημάτων και στην ελαχιστοποίηση του χρόνου ενασχόλησης από το προσωπικό του τμήματος πληροφορικής για την υποστήριξη των προγραμμάτων σπουδών. Για τον έλεγχο της εγκυρότητας των υποθέσεων έχει καταγραφεί η μεθοδολογία εκτέλεσης του πειράματος, η διαδικασία συγκέντρωσης δεδομένων, οι στατιστικές διαδικασίες επεξεργασίας των δεδομένων και η ερμηνεία των αποτελεσμάτων που θα προκύψουν.

Το σύστημα που υλοποιήσαμε αποτελείται από έξι υποσυστήματα. Οι λειτουργίες που υποστηρίζει κάθε υποσύστημα παρουσιάζονται στη συνέχεια.

Υποσύστημα Εκπαιδευτικού

- Προβολή και επεξεργασία τμημάτων ιδρύματος.
- Προβολή και επεξεργασία εκπαιδευτικών επιλεγμένου τμήματος.
- Προβολή και επεξεργασία μαθημάτων τμήματος.
- Προβολή και επεξεργασία ομάδων παρακολούθησης επιλεγμένου μαθήματος.
- Προβολή και επεξεργασία ακαδημαϊκών περιόδων τμήματος.
- Προβολή ιστορικού ακαδημαϊκών περιόδων με στατιστικά στοιχεία επιτυχιών των φοιτητών (με στόχο την έμμεση υποβοήθηση των αναθέσεων και την πρόβλεψη ομάδων παρακολούθησης).

- Αυθεντικοποίηση χρήστη με LDAP ή και με τη βάση δεδομένων του συστήματος.
- Αυτόματη δημιουργία και δημοσίευση ωρολογίων προγραμμάτων.
- Προβολή ωρολογίου προγράμματος ενεργής ακαδημαϊκής περιόδου.
- Συγχρονισμός των ομάδων παρακολούθησης με το δημιουργημένο ωρολόγιο πρόγραμμα, ώστε να υπάρχει ταύτιση στις ημέρες και στις ώρες.
- Εκκίνηση από τον προϊστάμενο (BPM) διεργασίας προετοιμασίας ακαδημαϊκής περιόδου.
- Εκκίνηση από τον προϊστάμενο (BPM) διεργασίας συγχρονισμού δηλώσεων ακαδημαϊκής περιόδου.
- Διαχείριση από τον προϊστάμενο και τον υπεύθυνο εκπαιδευτικό της (BPM) διεργασίας κορεσμού ομάδων παρακολούθησης μαθήματος.
- Προβολή Google Map με το τμήμα που ανήκει ο εκπαιδευτικός.

Υποσύστημα Φοιτητή

- Προβολή μαθημάτων που έχει δηλώσει στο γραμματειακό σύστημα.
- Δήλωση ομάδων παρακολούθησης μαθημάτων.
- Αυθεντικοποίηση χρήστη με LDAP ή και με τη βάση δεδομένων του συστήματος.
- Προβολή ωρολογίου προγράμματος ενεργής ακαδημαϊκής περιόδου.
- Εκκίνηση από το φοιτητή της (BPM) διεργασίας διαχείρισης κορεσμού ομάδων παρακολούθησης μαθήματος.
- Προβολή Google Map με το τμήμα που ανήκει ο φοιτητής.

Υποσύστημα Συγχρονισμού

- Συγχρονισμός βάσεων δεδομένων του γραμματειακού συστήματος (της υπάρχουσας υποδομής) και του νέου συστήματος υποστήριξης προγραμμάτων σπουδών με στοιχεία που αφορούν ανά τμήμα: ακαδημαϊκές περιόδους, ενεργούς φοιτητές, μαθήματα, εκπαιδευτικούς, δηλώσεις φοιτητών.
- Δυνατότητα συγχρονισμού συστήματος που δε βασίζεται στο σύνηθες γραμματειακό σύστημα που χρησιμοποιούν τα περισσότερα ιδρύματα, με το νέο σύστημα υποστήριξης προγραμμάτων σπουδών.
- Διαχείριση (BPM) διεργασίας προετοιμασίας ακαδημαϊκής περιόδου.
- Διαχείριση (BPM) διεργασίας συγχρονισμού λογαριασμού φοιτητή που δημιουργήθηκε εκτάκτως από τη γραμματεία.
- Διαχείριση (BPM) διεργασίας συγχρονισμού δηλώσεων ακαδημαϊκής περιόδου.

Υποσύστημα ωρολογίου προγράμματος

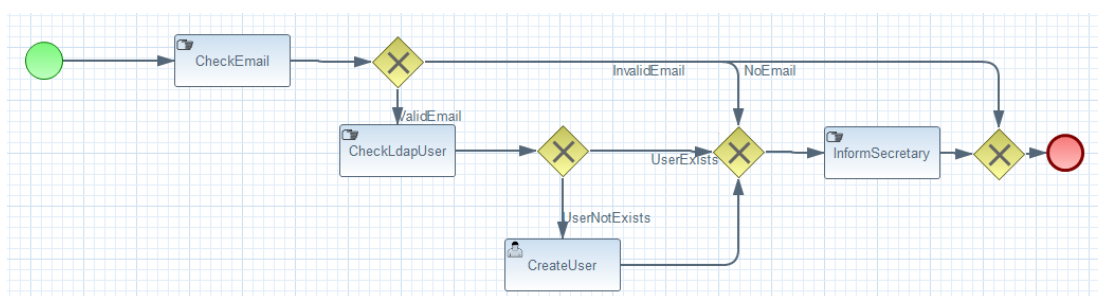
- Διασύνδεση με το εργαλείο (FET) παράγωγης ωρολογίου προγράμματος
- Τροφοδότηση περιορισμών (αίθουσες, εκπαιδευτικοί, ακαδημαϊκή περίοδος, μαθήματα, ομάδες παρακολούθησης, ημέρες της εβδομάδας, ώρες ημέρας, προτεινόμενες αίθουσες ανά ομάδα παρακολούθησης, διασύνδεση εκπαιδευτικού-μαθήματος-ομάδας παρακολούθησης και διάρκειας ομάδας παρακολούθησης, περιορισμό διαδοχικών ομάδων παρακολούθησης, ορισμό συγκεκριμένης ημέρας που θα πραγματοποιηθεί η ομάδα παρακολούθησης) ωρολογίου προγράμματος.
- Ανάλυση και μορφοποίηση του παραγόμενου ωρολογίου προγράμματος.

Υποσύστημα παροχής διεργασιών και Android back-end

- Εκτέλεση διεργασίας συγχρονισμού δηλώσεων ακαδημαϊκής περιόδου.
- Εκτέλεση διεργασίας προετοιμασίας νέας ακαδημαϊκής περιόδου.

- Εκτέλεση διεργασίας κορεσμού ομάδων παρακολούθησης μαθήματος περιόδου.
- Εκτέλεση διεργασίας συγχρονισμού λογαριασμού νέου χρήστη.
- Παροχή των διεργασιών (Export Processes) ως υπηρεσίες REST.
- Παροχή λειτουργιών Android (Export μεθόδων που διασυνδέονται με τη βάση δεδομένων του συστήματος) ως υπηρεσία REST.

Καθεμία από τις τέσσερις διεργασίες που παρέχει το υποσύστημα έχουν αρχικά σχεδιαστεί με το εργαλείο Eclipse Kepler. Το αποτέλεσμα του σχεδιασμού είναι ένα αρχείο XML που ακολουθεί το πρότυπο σχεδίασης BPMN των συστημάτων BPM (Business Process Management systems). Κάθε τέτοιο σχέδιο (αρχείο XML) με χρήση ειδικής βιβλιοθήκης (jBPM6) μπορεί να εκτελείται σε πραγματικό χρόνο. Αυτή η δυνατότητα εκτέλεσης του μοντέλου – σε πραγματικό χρόνο- σε συνδυασμό με τις τεχνολογίες JAVA EE και REST δίνει τη δυνατότητα να εκτελεστούν λειτουργίες που για να θεωρηθούν επιτυχημένες απαιτούν τη συντονισμένη συνεργασία διαφορετικών εφαρμογών που χρησιμοποιούν διαφορετικές κατηγορίες χρηστών. Παράδειγμα ενός τέτοιου μοντέλου φαίνεται στο διάγραμμα που ακολουθεί, η αναλυτική περιγραφή των μοντέλων και της παραμετροποίησης εκτέλεσης των διεργασιών BPM παρουσιάζονται στην ενότητα 2.3, και τα διαγράμματα κλάσεων που τα υλοποιούν με τις τεχνολογίες JAVA EE και REST παρουσιάζονται στην ενότητα 2.2.5.



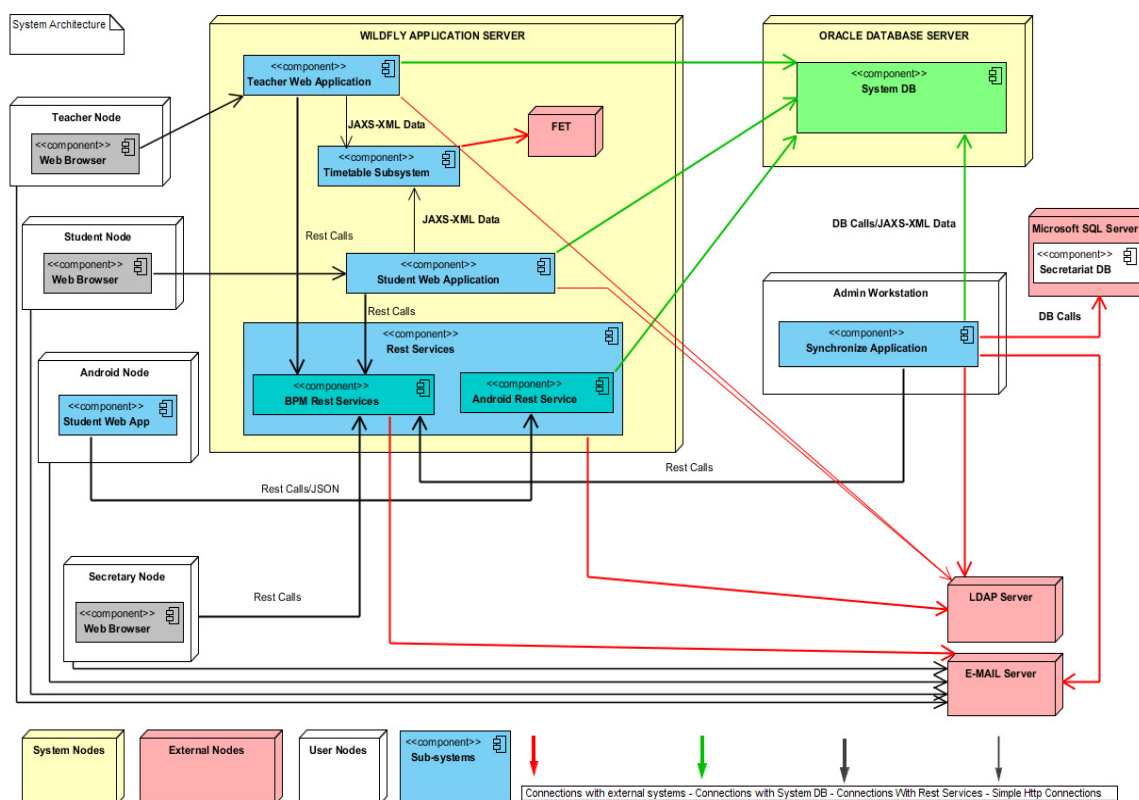
BPMN μοντέλο διεργασίας συγχρονισμού λογαριασμού νέου χρήστη

Υποσύστημα φοιτητή Android

- Προβολή μαθημάτων που έχει δηλώσει στο γραμματειακό σύστημα.
- Δήλωση ομάδων παρακολούθησης μαθημάτων.

- Αυθεντικοποίηση χρήστη με LDAP ή και με τη βάση δεδομένων του συστήματος.

Το υλοποιημένο σύστημα συγχρονίζει τη βάση δεδομένων του με την βάση του γραμματειακού συστήματος και διασυνδέεται με υπάρχοντα συστήματα της υποδομής του ιδρύματος ΤΕΙ ΑΜΘ. Η αρχιτεκτονική του φαίνεται στο διάγραμμα που ακολουθεί αλλά η περιγραφή γίνεται στην ενότητα 3.3. Στο κεφάλαιο «Εισαγωγή» της μεταπτυχιακής διατριβής απαριθμούνται η συνεισφορά της, οι τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν.



Διάγραμμα Αρχιτεκτονικής συστήματος

Λέξεις Κλειδιά

Αξιολόγηση λογισμικού, Αυτόματη δημιουργία ωρολογίων προγραμμάτων, Δηλώσεις ομάδων παρακολούθησης μαθημάτων, Διαλειτουργικότητα συστημάτων, Διασύνδεση ετερογενών συστημάτων, Διατύπωση Ερευνητικής Πρότασης, Επαλήθευση Ερευνητικών Υποθέσεων, Μοντέλο αναπαράστασης-εκτέλεσης διεργασιών, Μοντελοποίηση Σχεσιακής Βάσης Δεδομένων, Μοντελοποίηση UML, Περιορισμοί ωρολογίων προγραμμάτων, Συγχρονισμός συστημάτων, Συντονισμός χρηστών, Υλοποίηση βάσης δεδομένων, Υπηρεσίες REST, Υποστήριξη προγραμμάτων σπουδών, Ωρολόγιο πρόγραμμα, Android back-end, BPM, BPMN, Data Modeler, Java EE, MVC.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω ολόκληρο το προσωπικό που συμμετείχε στο πρόγραμμα σπουδών «Πληροφοριακά και Επικοινωνιακά Συστήματα» του Ανοικτού Πανεπιστημίου Κύπρου για την άριστη ποιότητα εκτέλεσης του προγράμματος σπουδών. Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της μεταπτυχιακής διατριβής κ.Γκουμόπουλο Χρήστο για τις σημαντικές υποδείξεις του, την άμεση απόκρισή του, το υπόβαθρο που μας δημιούργησε ως καθηγητής στο αντικείμενο της «Αντικειμενοστρεφούς Ανάλυσης & Σχεδίασης», καθώς και την δυνατότητα που μου προσέφερε να εργαστώ ενεργά σε αρκετά μεγάλη ποικιλία τεχνολογιών υλοποίησης επιχειρησιακών εφαρμογών. Θα ήθελα να ευχαριστήσω τον κ. Μουρλά Κωνσταντίνο για το υπόβαθρο που μας δημιούργησε ως καθηγητής στο αντικείμενο «Ερευνητικές Μέθοδοι» χωρίς το οποίο δε θα μπορούσε να συγγραφεί η «ερευνητική πρόταση» της μεταπτυχιακής διατριβής. Θα ήθελα ιδιαίτερα να ευχαριστήσω τον προϊστάμενο του γενικού τμήματος πληροφορικής του ιδρύματος Ανατολικής Μακεδονίας και Θράκης κ. Μίζα Χαρίλαο για τη συμμετοχή του στην αξιολόγηση του συστήματος και για την δυνατότητα πρόσβασης που μου παρείχε στους απαιτούμενους εξυπηρετητές του ιδρύματος: χωρίς αυτήν την πρόσβαση δεν θα ήταν εφικτή η υλοποίηση της μεταπτυχιακής διατριβής. Θα ήθελα να ευχαριστήσω τους φίλους, δασκάλους και συνεργάτες μου Αλεβίζο Θεόδωρο-Καθηγητής τμήματος Βιομηχανικής Πληροφορικής ΤΕΙ Καβάλας- και Σκουρλά Χρήστο-Καθηγητής τμήματος ΤΕΙ Αθήνας- που συνεισέφεραν ενεργά στη διαμόρφωση του επιστημονικού υποβάθρου μου στον τομέα «Προγραμματισμού και Βάσεων Δεδομένων», αλλά και ευρύτερα στον διαπροσωπικό και επαγγελματικό τομέα. Θα ήθελα να ευχαριστήσω το φίλο και συνεργάτη Μπέλση Πέτρο για την έμπνευση των τεχνολογιών υλοποίησης όπως διαμορφώθηκαν στο «Θέμα» της μεταπτυχιακής διατριβής και στην ενεργή συμμετοχή του στην αξιολόγηση του συστήματος. Θα ήθελα να ευχαριστήσω τον κ.Μωυσιάδη Ελευθέριο προϊστάμενο τμήματος Μηχανικών Πληροφορικής ΤΕΙ ΑΜΘ και τους φοιτητές Δημητριάδη Δημήτρη και Μιμίδου Ελένη για τη συμμετοχή τους στην αξιολόγηση του συστήματος και στην επαλήθευση της διαδικασίας εγκατάστασης του συστήματος κάνοντας χρήση των οδηγιών που υπάρχουν στο συνοδευτικό υλικό της μεταπτυχιακής διατριβής. Τέλος, θα ήθελα να ευχαριστήσω τους γονείς και τους φίλους για την υπομονή, την κατανόηση και την υποστήριξη που μου παρείχαν καθόλη τη διάρκεια παρακολούθησης του προγράμματος σπουδών.

Περιεχόμενα

ΠΕΡΙΛΗΨΗ	III
SUMMARY	IV
ΕΚΤΕΤΑΜΕΝΗ ΠΕΡΙΛΗΨΗ.....	VI
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ	X
ΕΥΧΑΡΙΣΤΙΕΣ.....	XI
ΠΕΡΙΕΧΟΜΕΝΑ	XII
ΕΙΣΑΓΩΓΗ	16
ΑΝΤΙΚΕΙΜΕΝΟ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΔΙΑΤΡΙΒΗΣ.....	16
<i>Αποτελέσματα</i>	16
<i>Τεχνολογίες</i>	18
<i>Εργαλεία</i>	18
ΟΡΓΑΝΩΣΗ ΚΕΙΜΕΝΟΥ	20
ΕΡΕΥΝΗΤΙΚΗ ΠΡΟΤΑΣΗ	22
1.1 ΔΗΛΩΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ.....	22
1.2 ΕΠΙΣΚΟΠΗΣΗ ΤΗΣ ΒΙΒΛΙΟΓΡΑΦΙΑΣ	23
1.2.1 <i>Ωρολόγια προγράμματα (Timetables)</i>	24
1.2.2 <i>Τεχνολογικά εργαλεία</i>	25
1.2.3 <i>Business Process Management systems</i>	26
1.3 ΔΗΛΩΣΗ ΤΗΣ ΥΠΟΘΕΣΗΣ	28
1.3.1 <i>Υπόθεση 1</i>	28
1.3.2 <i>Υπόθεση 2</i>	29
1.3.3 <i>Υπόθεση 3</i>	29
1.4 ΣΠΟΥΔΑΙΟΤΗΤΑ ΤΗΣ ΜΕΛΕΤΗΣ	29
1.5 ΜΕΘΟΔΟΛΟΓΙΑ	31
1.5.1 <i>Πληθυσμοί</i>	31
1.5.2 <i>Όργανα</i>	32
1.5.3 <i>Διαδικασίες</i>	33
1.6 ΑΝΑΛΥΣΗ ΔΕΔΟΜΕΝΩΝ	34
1.6.1 <i>Υπόθεση 1</i>	34
1.6.2 <i>Υπόθεση 2</i>	36
1.6.3 <i>Υπόθεση 3</i>	39
1.7 ΧΡΟΝΟΔΙΑΓΡΑΜΜΑ ΕΡΓΑΣΙΩΝ.....	41
ΑΝΑΛΥΣΗ/ΣΧΕΔΙΑΣΗ ΣΥΣΤΗΜΑΤΟΣ	44
2.1 ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ.....	44
2.1.1 <i>Υπάρχοντα ή εξωτερικά υποσυστήματα και υπηρεσίες</i>	44
2.1.2 <i>Νέα υποσυστήματα</i>	45
2.2 ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΑΣΕΩΝ	50
2.2.1 <i>Διάγραμμα κλάσεων υποσυστήματος φοιτητών</i>	50
2.2.2 <i>Διάγραμμα κλάσεων υποσυστήματος καθηγητών</i>	51
2.2.3 <i>Διάγραμμα κλάσεων ωρολογίου προγράμματος</i>	52
2.2.4 <i>Διάγραμμα κλάσεων υποσυστήματος συγχρονισμού</i>	57
2.2.5 <i>Διάγραμμα κλάσεων υποσυστήματος διεργασιών</i>	58
2.3 ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΔΙΕΡΓΑΣΙΩΝ(BPM PROCESSES)	67
2.3.1 <i>Διεργασία συγχρονισμού λογαριασμού νέου χρήστη (NewUser BPM Process)</i>	67

2.3.2	Διεργασία προετοιμασίας νέας ακαδημαϊκής περιόδου (NewAcadPeriod BPM Process)	73
2.3.3	Διεργασία συγχρονισμού δηλώσεων ακαδημαϊκής περιόδου (SyncPeriodRegistrations BPM Process)	82
2.3.4	Διεργασία κορεσμού ομάδων παρακολούθησης μαθήματος περιόδου (LessonGroupsFull BPM Process)	85
ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ.....		93
3.1	ΔΙΕΠΑΦΕΣ ΥΠΟΣΥΣΤΗΜΑΤΩΝ	93
3.1.1	Διεπαφή υποσυστήματος φοιτητών.....	93
3.1.2	Διεπαφή υποσυστήματος εκπαιδευτικών.....	99
3.1.3	Διεπαφή υποσυστήματος ωρολογίου προγράμματος	111
3.1.4	Διεπαφή υποσυστήματος συγχρονισμού.....	112
3.1.5	Διεπαφή υποσυστήματος διεργασιών.	121
3.1.6	Διεπαφή υποσυστήματος Android	136
3.2	ΥΛΟΠΟΙΗΣΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ	139
3.2.1	Απαιτήσεις-Περιορισμοί.....	139
3.2.2	Μοντέλο Οντοτήτων Συσχετίσεων (ERD).....	140
3.2.3	Διάγραμμα UML	141
3.3	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ	143
3.3.1	Κόμβοι χρηστών.....	144
3.3.2	Κόμβοι συστήματος.....	144
3.3.3	Εξωτερικοί κόμβοι	145
3.4	ΔΙΑΔΙΚΑΣΙΑ ΕΓΚΑΤΑΣΤΑΣΗΣ ΣΥΣΤΗΜΑΤΟΣ	147
3.4.1	Εγκατάσταση ΣΔΒΣ	147
3.4.2	Εγκατάσταση εξυπηρετητή εφαρμογών.....	148
3.4.3	Εγκατάσταση εφαρμογής Android φοιτητών	148
3.4.4	Εργαλεία ανάπτυξης-υλοποίησης του συστήματος	148
ΑΞΙΟΛΟΓΗΣΗ ΣΥΣΤΗΜΑΤΟΣ		150
4.1	ΠΛΑΝΟ ΕΠΑΛΗΘΕΥΣΗΣ ΕΡΕΥΝΗΤΙΚΩΝ ΥΠΟΘΕΣΕΩΝ	150
4.1.1	Υπόθεση 1.....	150
4.1.2	Υπόθεση 2.....	152
4.1.3	Υπόθεση 3.....	154
4.2	ΑΞΙΟΛΟΓΗΣΗ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ.....	156
4.2.1	Χρήστες αξιολόγησης του συστήματος	156
4.2.2	Αξιολόγηση από χρήστες κατηγορίας Μηχανικού Λογισμικών.....	157
4.2.3	Αξιολόγηση από χρήστες κατηγορίας Προϊσταμένου τμήματος.....	167
4.2.4	Αξιολόγηση από χρήστες κατηγορίας Διαχειριστή Συστημάτων.....	168
4.2.5	Αξιολόγηση από χρήστες κατηγορίας Φοιτητή	170
4.2.6	Αποτελέσματα αξιολόγησης συστήματος	171
ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ		172
5.1	ΣΥΝΤΑΚΤΗΣ ΩΡΟΛΟΓΙΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ.....	172
5.2	ΠΡΟΒΛΕΨΗ ΔΗΛΩΣΕΩΝ ΜΕΛΛΟΝΤΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΠΕΡΙΟΔΩΝ	173
5.3	ΑΣΦΑΛΕΙΑ ΔΕΔΟΜΕΝΩΝ ΣΤΗΝ ΕΠΙΚΟΙΝΩΝΙΑ ΤΟΥ ΥΠΟΣΥΣΤΗΜΑΤΟΣ ΥΠΗΡΕΣΙΩΝ REST	173
5.4	ΛΕΙΤΟΥΡΓΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....	174
ΕΠΙΛΟΓΟΣ		175
ΒΙΒΛΙΟΓΡΑΦΙΑ		176
ΚΛΑΣΕΙΣ ΥΠΟΣΥΣΤΗΜΑΤΩΝ		1

A.1	ΠΕΡΙΓΡΑΦΗ ΚΛΑΣΕΩΝ ΥΠΟΣΥΣΤΗΜΑΤΟΣ ΦΟΙΤΗΤΩΝ.....	1
A.2	ΠΕΡΙΓΡΑΦΗ ΚΛΑΣΕΩΝ ΥΠΟΣΥΣΤΗΜΑΤΟΣ ΕΚΠΑΙΔΕΥΤΙΚΩΝ.....	13
A.3	ΠΕΡΙΓΡΑΦΗ ΚΛΑΣΕΩΝ ΥΠΟΣΥΣΤΗΜΑΤΟΣ ΩΡΟΛΟΓΙΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ	30
A.4	ΠΕΡΙΓΡΑΦΗ ΚΛΑΣΕΩΝ ΥΠΟΣΥΣΤΗΜΑΤΟΣ ΣΥΓΧΡΟΝΙΣΜΟΥ	37
A.5	ΠΕΡΙΓΡΑΦΗ ΚΛΑΣΕΩΝ ΥΠΟΣΥΣΤΗΜΑΤΟΣ ΠΑΡΟΧΗΣ ΔΙΕΡΓΑΣΙΩΝ.....	50
A.5.1	<i>Κλάσεις παροχής διεργασίας συγχρονισμού δηλώσεων ακαδημαϊκής περιόδου.....</i>	50
A.5.2	<i>Κλάσεις παροχής διεργασίας προετοιμασίας νέας ακαδημαϊκής περιόδου.....</i>	54
A.5.3	<i>Κλάσεις παροχής διεργασίας κορεσμού ομάδων παρακολούθησης.....</i>	58
A.5.4	<i>Κλάσεις παροχής διεργασίας συγχρονισμού λογαριασμού νέου χρήστη.....</i>	62
A.6	ΠΕΡΙΓΡΑΦΗ ΚΛΑΣΕΩΝ ΥΠΟΣΥΣΤΗΜΑΤΟΣ ANDROID ΦΟΙΤΗΤΩΝ	65
ΥΛΟΠΟΙΗΣΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ		74
B.1	ΠΕΡΙΓΡΑΦΗ ΠΙΝΑΚΩΝ	74
B.2	ΠΕΡΙΓΡΑΦΗ ΟΨΕΩΝ (VIEWS).....	84
B.3	ΠΕΡΙΓΡΑΦΗ ΑΠΟΘΗΚΕΥΜΕΝΩΝ ΔΙΑΔΙΚΑΣΙΩΝ, ΣΥΝΑΡΤΗΣΕΩΝ ΚΑΙ ΣΚΑΝΔΑΛΙΣΜΩΝ (STORED PROCEDURES, FUNCTIONS AND TRIGGERS).....	87

Εισαγωγή

Αντικείμενο μεταπτυχιακής διατριβής

Αντικείμενο της μεταπτυχιακής διατριβής είναι η πιλοτική κατασκευή ενός αποδοτικού συστήματος υποστήριξης της εκτέλεσης προγραμμάτων σπουδών των Ελληνικών εκπαιδευτικών ιδρυμάτων τριτοβάθμιας εκπαίδευσης. Για την εξασφάλιση και τη μέτρηση αυτής της απόδοσης έπρεπε να ερευνησουμε τις απαιτήσεις των εκπαιδευτικών ιδρυμάτων, να ορίσουμε τη μεθοδολογία, να επιλέξουμε τα κατάλληλα εργαλεία και τεχνικές σχεδίασης-υλοποίησης και να υλοποιήσουμε το σύστημα. Ως μελέτη περίπτωσης για την έρευνα και την υλοποίηση βασιστήκαμε στο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Ανατολικής Μακεδονίας και Θράκης (ΤΕΙ ΑΜΘ).

Αποτελέσματα

Τα αποτελέσματα της μεταπτυχιακής διατριβής συνοψίζονται ως εξής:

- Ανάπτυξη ερευνητικής πρότασης τριών υποθέσεων και της διαδικασίας επαλήθευσης των υποθέσεων με επιστημονικό τρόπο. Ο έλεγχος ωστόσο των υποθέσεων δεν έγινε στο πλαίσιο της διατριβής για πρακτικούς λόγους, αλλά καθορίστηκε η μεθοδολογία και το πλάνο επαλήθευσής τους (συγκέντρωση των δεδομένων, οι στατιστικές διαδικασίες και ο τρόπος ερμηνείας των αποτελεσμάτων).
- Εξέταση μεθοδολογιών κατασκευής ωρολογίων προγραμμάτων (timetables), επιλογή της προσέγγισης FET και υλοποίηση μιας εφαρμογής που συνδέεται με την μηχανή FET.
- Ανάλυση/σχεδίαση και υλοποίηση υποσυστήματος φοιτητή.
- Ανάλυση/σχεδίαση και υλοποίηση υποσυστήματος εκπαιδευτικού.

- Ανάλυση/σχεδίαση και υλοποίηση υποσυστήματος ωρολογίου προγράμματος.
- Ανάλυση/σχεδίαση και υλοποίηση υποσυστήματος συγχρονισμού.
- Ανάλυση/σχεδίαση και υλοποίηση υποσυστήματος παροχής διεργασιών και Android back-end.
- Ανάλυση/σχεδίαση και υλοποίηση υποσυστήματος φοιτητών σε έκδοση Android.
- Μοντελοποίηση συστήματος με διαγράμματα κλάσεων UML για κάθε υποσύστημα.
- Μοντελοποίηση βάσης δεδομένων με συμβολισμό Elmasri-Navathe και με διάγραμμα UML.
- Εξοικείωση στην υλοποίηση εφαρμογών σε τεχνολογία JAVA EE από μηδενική βάση.
- Εξοικείωση στη σχεδίαση/υλοποίηση συστήματος BPM από μηδενική βάση.
- Εξοικείωση στην ανάπτυξη εφαρμογών Android από μηδενική βάση.
- Συγχρονισμός και διασύνδεση του συστήματος που υλοποιήσαμε με υπάρχοντα συστήματα (LDAP, Microsoft SQL Server) και υπηρεσίες με ομαλό/αποδοτικό τρόπο.
- Εκτέλεση εργασιών –με επιτυχία- που για την ολοκλήρωσή τους απαιτείται συντονισμός αρκετών και διαφορετικής φύσης χρηστών ή συστημάτων/υπηρεσιών.
- Υλοποίηση υπηρεσιών REST.
- Διασύνδεση εφαρμογών Android με back-end κάνοντας χρήση υπηρεσιών REST.
- Διασύνδεση υλοποιημένων υποσυστημάτων κάνοντας χρήση υπηρεσιών REST.

Τεχνολογίες

Οι τεχνολογίες που χρησιμοποιήθηκαν στη μεταπτυχιακή διατριβή είναι οι εξής:

- PL/SQL (Procedural Language/SQL)
- UML (Unified Modeling Language)
- BPM (Business Process Management)
- BPMN 2 (Business Process Model and Notation)
- JAVA SE/EE (Standard/Enterprise Editions): Context and Dependency Injection(CDI), Bean Validation, Java Persistence API (JPA), Object Relational Mapping, Java Server Faces (JSF)
- MVC (Model View Controller)
- XML and JSON Processing
- Restfull Web Services (REST)
- Maven
- Android
- Open LDAP

Η χρήση τους και η ακριβή τους έκδοση παρουσιάζεται στο κεφάλαιο «Υλοποίηση συστήματος».

Εργαλεία

Τα εργαλεία που χρησιμοποιήθηκαν στη μεταπτυχιακή διατριβή είναι τα εξής:

- Visual Paradigm

- SmartDraw CI
- Oracle Database Server
- Oracle SQL*PLUS
- Oracle SQL Developer & Data Modeler
- Microsoft SQL Server
- Μεταγλωττιστής Java EE7
- Netbeans 8 EE
- Novell LDAP Classes for Java
- jTDS
- JBOSS -Wildfly- Application Server
- PrimeFaces
- Apache Maven
- JBPM6
- Eclipse Kepler
- JAXS-RS
- Android Studio
- FET

Η χρήση τους και η ακριβή τους έκδοση παρουσιάζεται στο κεφάλαιο «Υλοποίηση συστήματος».

Οργάνωση κειμένου

Το κυρίως περιεχόμενο της μεταπτυχιακής διατριβής διαφείται σε πέντε κεφάλαια.

Το πρώτο κεφάλαιο «Ερευνητική Πρόταση» πραγματεύεται την έρευνα της μεταπτυχιακής διατριβής με επιστημονικό τρόπο (Scientific Research). Για τη συγγραφή της ερευνητικής πρότασης ακολουθήσαμε τη δομή που περιγράφεται στο κεφάλαιο 20 του βιβλίου *Introduction to Research in Education* [1][2]. Ξεκινήσαμε με τη δήλωση του προβλήματος και κάναμε επισκόπηση της βιβλιογραφίας σε θέματα που αφορούν τα ωρολόγια προγράμματα και τα τεχνολογικά εργαλεία υλοποίησης. Περιγράφουμε την αρχιτεκτονική MVC και την έννοια της *BPM διεργασίας*. Στη συνέχεια διατυπώσαμε τρεις υποθέσεις –που στοχεύουν στην εξοικονόμηση χρόνου και στη μείωση προβλημάτων, παρουσιάσαμε τη σπουδαιότητα της μελέτης μας, ορίσαμε τους πληθυσμούς, τα όργανα μέτρησης, τη διαδικασία εκτέλεσης του πειράματος, οργανώσαμε τα δεδομένα των υποθέσεων και καθορίσαμε τις στατιστικές διαδικασίες τους. Τέλος καθορίσαμε το χρονοδιάγραμμα των εργασιών μας.

Το δεύτερο κεφάλαιο «Ανάλυση/Σχεδίαση Συστήματος» πραγματεύεται την ανάλυση-σχεδίαση της μεταπτυχιακής διατριβής. Ξεκινήσαμε με την καταγραφή των λειτουργικών απαιτήσεων καθορίζοντας την ανάγκη υλοποίησης έξι νέων υποσυστημάτων λογισμικού (*υποσύστημα φοιτητών, υποσύστημα εκπαιδευτικών, υποσύστημα ωρολογίου προγράμματος, υποσύστημα συγχρονισμού, υποσύστημα REST παροχής διεργασιών και Android-backend, υποσύστημα φοιτητών-Android*) και μιας νέας βάσης δεδομένων που θα συνεργάζονται με την υπάρχουσα υποδομή του ΤΕΙ ΑΜΘ. Στη συνέχεια για κάθε υποσύστημα αναπτύξαμε το διάγραμμα κλάσεων (με το εργαλείο Visual Paradigm). Λόγω του μεγέθους των διαγραμμάτων κλάσεων υπάρχουν δυο εκδόσεις για κάθε υποσύστημα: η συμπυκνόμενη και η λεπτομερής. Στο κείμενο της μεταπτυχιακής διατριβής παρουσιάζουμε την συμπυκνόμενη έκδοση, ενώ η λεπτομερής έκδοση υπάρχει στο συνοδευτικό υλικό. Κάθε διάγραμμα κλάσεων συμπληρώνει λεπτομέρειες απαιτήσεων που δεν είναι τόσο προφανείς στις λειτουργικές απαιτήσεις. Με το εργαλείο Eclipse μοντελοποιήσαμε στη γλώσσα BPMN (Business Process Model and Notation Version) σημαντικές διεργασίες BPM (Business Process Management) που είναι απαραίτητες για το συντονισμό και τον έλεγχο της ροής των ενεργειών των διαφορετικών υποσυστημάτων που χρησιμοποιούν διαφορετικοί χρήστες.

Στο τρίτο κεφάλαιο «Υλοποίηση Συστήματος» αρχικά παρουσιάζονται οι διεπαφές των υποσυστημάτων που υλοποιήθηκαν. Κάθε περιβάλλον διεπαφής παρουσιάζεται ως σενάριο

χρήσης από την πλευρά του χρήστη. Στη συνέχεια παρουσιάζεται η σχεδίαση της βάσης δεδομένων του συστήματος διατυπώνοντας τους περιορισμούς της, το μοντέλο οντοτήτων συσχετίσεων με συμβολισμό Elmasri-Navathe και συμβολισμό UML, και η υλοποίηση του σχήματος σχέσεων (πίνακες), οι όψεις (views), οι αποθηκευμένες διαδικασίες, οι συναρτήσεις και οι σκανδαλισμοί (stored procedures, functions και triggers). Ακολούθως περιγράφεται η αρχιτεκτονική του συστήματος. Τέλος, περιγράφεται η διαδικασία εγκατάστασης του συστήματος και τα εργαλεία σχεδίασης και υλοποίησης που χρησιμοποιήθηκαν.

Στο τέταρτο κεφάλαιο «Αξιολόγηση Συστήματος» παρέχεται το πλάνο επαλήθευσης των ερευνητικών υποθέσεων και περιγράφεται η αξιολόγηση του λογισμικού.

Τέλος, στο πέμπτο κεφάλαιο «Μελλοντικές Επεκτάσεις» προτείνεται η ενσωμάτωση νέων δυνατοτήτων (και σε συνδυασμό με τα συμπεράσματα του κεφαλαίου 4) που θα πρέπει να έχει το σύστημα ώστε να αλλάξει η πιλοτική του φύση σε προϊόν προς λειτουργία. Οι νέες δυνατότητες που προτείνονται είναι η κατασκευή συντάκτη ωρολογίου προγράμματος, η πρόβλεψη δηλώσεων μελλοντικών ακαδημαϊκών περιόδων με τεχνολογίες εξόρυξης δεδομένων και η διασφάλιση του υποσυστήματος REST.

Κεφάλαιο 1

Ερευνητική πρόταση

Οι πρόσφατες οικονομικές μεταρρυθμίσεις στον ελλαδικό χώρο φέρνουν αρκετούς οργανισμούς και επιχειρήσεις αντιμέτωπους με την πρόκληση υποστήριξης νέων, ποιοτικών και συνεχώς αυξανόμενων παρεχόμενων υπηρεσιών πληροφορικής ενώ ταυτόχρονα παρέχεται μειωμένο προσωπικό τεχνικής υποστήριξης. Τα τριτοβάθμια ελληνικά ιδρύματα δεν αποτελούν εξαίρεση και θα πρέπει να βρεθούν τρόποι διατήρησης-ενίσχυσης της συνοχής της συνολικής λειτουργίας τους βασιζόμενα στους ελάχιστους πόρους που διαθέτουν.

1.1 Δήλωση του προβλήματος

Τα τριτοβάθμια ιδρύματα διαθέτουν κάποιο πληροφοριακό γραμματειακό σύστημα για την υποστήριξη των προγραμμάτων σπουδών κάθε τμήματος. Τα περισσότερα ιδρύματα χρησιμοποιούν το ίδιο πληροφοριακό σύστημα. Το γραμματειακό σύστημα για κάθε τμήμα ιδρύματος διαχειρίζεται κυρίως τα εξής: προγράμματα σπουδών τμήματος, μαθήματα προγράμματος σπουδών, μαθήματα που έχουν ανατεθεί σε καθηγητές, δηλώσεις μαθημάτων και βαθμολόγια φοιτητών.

Ωστόσο για την ομαλή εκτέλεση των προγραμμάτων σπουδών εσωτερικά κάθε τμήμα εκτελεί και άλλες λειτουργίες όπως:

- *Αναθέσεις μόνιμων και εκτάκτων εκπαιδευτικών σε μαθήματα.* Κάθε τμήμα δεν έχει γνώση εγκαίρως για το πλήθος των φοιτητών που θα παρακολουθήσει κάποιο μάθημα και συνεπώς δεν είναι σε θέση να κάνει έγκαιρα τις αναθέσεις μαθημάτων σε εκπαιδευτικούς. Συγκεκριμένα κάθε τμήμα δε μπορεί να γνωρίζει: α) πόσοι φοιτητές εν δυνάμει θα παρακολουθήσουν κάποιο μάθημα πριν ολοκληρωθεί η προθεσμία δηλώσεων μαθημάτων από φοιτητές, β) ακόμα και αν γνωρίζει τους εν δυνάμει φοιτητές δε μπορεί να γνωρίζει πόσοι από αυτούς θα παρακολουθούν εν γένει τα μαθήματα. Για τα εργαστηριακά μαθήματα το πρόβλημα είναι οξύτερο καθώς το γραμματειακό σύστημα δεν υποστηρίζει δηλώσεις εργαστηριακών ομάδων παρακολούθησης,

συνεπώς δεν μπορεί να υπάρχει συνολική εικόνα ακόμα και αν ολοκληρωθούν οι δηλώσεις ομάδων των εργαστηριακών μαθημάτων. Οι αναθέσεις των εκπαιδευτικών θα πρέπει να γίνονται όχι μόνο βάσει των δηλώσεων μαθημάτων αλλά και βάσει των δηλώσεων ομάδων παρακολούθησης μαθημάτων. Αυτό κρίνεται απαραίτητο καθώς οι δηλώσεις ομάδων παρακολούθησης έχουν πολλές φορές διαφορετικά κριτήρια για να τα παρακολουθήσει κάποιος ή μπορεί να ακολουθούν άλλους κανονισμούς. Για παράδειγμα σε κάποια τμήματα δεν επιτρέπεται σε φοιτητές που έχουν παρακολουθήσει ανεπιτυχώς εργαστηριακό μάθημα να το παρακολουθήσουν ξανά.

- *Δημιουργία ομάδων παρακολούθησης μαθημάτων.* Στα εργαστηριακά κυρίως μαθήματα ο αριθμός των ομάδων παρακολούθησης εξαρτάται από τους διαθέσιμους καθηγητές, από τις εργαστηριακές αίθουσες, από το σύνολο των φοιτητών που έχουν δηλώσει το μάθημα ή /και το σύνολο των φοιτητών που θα το παρακολουθήσουν (αν αυτό υπάρχει τρόπος να προβλεφθεί).

- *Δημιουργία εβδομαδιαίου ωρολογίου προγράμματος βάσει κριτηρίων.* Εμφανίζονται σημαντικές δυσκολίες στην κατασκευή ωρολογίου προγράμματος αν δεν έχουν καθοριστεί οι ομάδες παρακολούθησης μαθημάτων. Αξίζει να σημειώσουμε πως αρκετά θεωρητικά μαθήματα πραγματοποιούνται σε εργαστηριακές αίθουσες.

- *Κατανομή των φοιτητών σε ομάδες παρακολούθησης μαθημάτων βάσει κριτηρίων.* Για τη δίκαιη τοποθέτηση των φοιτητών σε ομάδες παρακολούθησης είναι απαραίτητο να καθοριστούν κάποια κριτήρια, όπως π.χ. το εξάμηνο παρακολούθησης.

1.2 Επισκόπηση της Βιβλιογραφίας

Μετά από ανασκόπηση της βιβλιογραφίας συμπεράναμε πως για την αντιμετώπιση των ζητημάτων της μεταπτυχιακής διατριβής θα πρέπει να ερευνηθούν τα παρακάτω.

α. Εξέταση μεθοδολογιών/μηχανισμών κατασκευής ωρολογίων προγραμμάτων / χρονοδιαγραμμάτων (timetables) [3],[4].

β. Τα τεχνολογικά εργαλεία που θα επιτρέψουν την υλοποίηση ενός συστήματος και θα προσεγγίζει τα ζητήματα της μεταπτυχιακής διατριβής με τρόπο ποιοτικό, οικονομικό και αποτελεσματικό για τα τριτοβάθμια ιδρύματα.

γ. Εξέταση τεχνολογιών εξόρυξης γνώσης για την δυνατότητα προβλέψεων των δηλώσεων [3],[4].

Έμφαση θα δοθεί στα ωρολόγια προγράμματα και στις τεχνολογίες υλοποίησης, αλλά όχι στις τεχνολογίες εξόρυξης δεδομένων.

1.2.1 Ωρολόγια προγράμματα (Timetables)

Αρκετά τριτοβάθμια ιδρύματα συνεχίζουν να επιλύουν προβλήματα ωρολογίων προγραμμάτων με χειρονακτικό τρόπο, παρά το γεγονός πως υπάρχουν αρκετά συστήματα που μπορούν να εφαρμοστούν [5]. Ο βασικός στόχος ενός προβλήματος ωρολογίου προγράμματος είναι η ανάθεση χρονικών περιόδων σε αίθουσες διδασκαλίας με τέτοιο τρόπο ώστε να ικανοποιούνται στο βέλτιστο οι προτιμήσεις των καθηγητών και να ελαχιστοποιηθούν οι συγκρούσεις μεταξύ των φοιτητών [6],[7]. Επιπλέον κάθε ωρολόγιο πρόγραμμα τροποποιείται σε σημαντικό βαθμό και αρκετές φορές μέχρι να θεωρηθεί οριστικό, ενώ η πλήρης αυτοματοποίησή του αποτελεί πρόκληση [6]. Τροποποιημένες εκδόσεις μπορεί να προκύψουν ακόμα και μετά τη δημοσίευσή του. Η κατασκευή ενός ωρολογίου προγράμματος προϋποθέτει την τροφοδότησή του με στοιχεία εισόδου και περιορισμούς, όπως κτίρια, αίθουσες και η χωρητικότητά τους, χρονικοί περίοδοι διαλέξεων, αργίες, μαθήματα, πλήθος ομάδων παρακολούθησης φοιτητών ανά μάθημα, μέγεθος ομάδας φοιτητών, αίθουσες με ειδικό εξοπλισμό κατάλληλες για συγκεκριμένα μαθήματα, ανάθεση εκπαιδευτικού προσωπικού σε μαθήματα, προτιμήσεις εκπαιδευτικού προσωπικού, διασπορά μιας διάλεξης μέσα στην εβδομάδα, συμπυκνωμένη κατανομή διαλέξεων μαθημάτων ανά ακαδημαϊκό εξάμηνο μέσα στην εβδομάδα, ακολουθία εκτέλεσης διαλέξεων ίδιων ή διαφορετικών μαθημάτων [6],[7],[8]. Οι περιορισμοί ενός ωρολογίου προγράμματος συνήθως χωρίζονται σε αυστηρούς (hard constraints) που πρέπει ρητά να τηρηθούν και σε λιγότερο αυστηρούς (soft constraints) οι οποίοι θα ληφθούν υπόψη στο μέτρο που δεν έρχονται σε σύγκρουση με αυστηρούς [8],[9]. Το είδος του ωρολογίου προγράμματος π.χ. εβδομαδιαίο ωρολόγιο πρόγραμμα διαλέξεων ή ωρολόγιο πρόγραμμα εξετάσεων είναι σημαντικό να εξεταστεί καθώς κάθε είδος διέπεται από διαφορετικά στοιχεία εισόδου και περιορισμούς [8]. Μια άλλη διάσταση ωρολογίου προγράμματος που θα μπορούσε να έχει ενδιαφέρον είναι η μερική αναθεώρηση του ωρολογίου προγράμματος για ομάδες παρακολούθησης μαθημάτων αρκετό διάστημα μετά την έναρξη του ακαδημαϊκού εξαμήνου, καθώς μπορεί να διαφοροποιηθούν τα στοιχεία εισόδου ή και οι περιορισμοί. Για την επίλυση των προβλημάτων ωρολογίων προγραμμάτων έχουν χρησιμοποιηθεί αρκετές μέθοδοι: ακολουθιακοί (sequential),

ομαδοποίησης (clustering), βασισμένοι σε περιορισμούς (constraint-based) και μετα-ευρετικοί (meta-heuristic). Οι αλγόριθμοι της κατηγορίας μετα-ευρετικής μεθόδου παρήγαγαν υψηλής ποιότητας αποτελέσματα και είναι οι εξής: γενετικοί και μεμετικοί αλγόριθμοι (GMA), Simulated Annealing(SA), Tabu Search (TS), Graph Coloring Algorithmh (GCA) και Mixed Integer Programming (MIP). Μεταξύ αυτών οι πιο συχνές προσεγγίσεις επίλυσης που έχουν γίνει είναι οι MIP και GMA [8]. Τα ωρολόγια προγράμματα έχουν μεγάλη επίδραση στην καθημερινή ζωή των ανθρώπων που τα χρησιμοποιούν. Έχουν απασχολήσει τη διεθνή κοινότητα από πολύ παλιά [7] και αναμφίβολα η έρευνα θα συνεχιστεί για πολύ ακόμη.

Για την κατασκευή ωρολογίου προγράμματος στο σύστημά μας προσφύγαμε στην ενσωμάτωση έτοιμων συστημάτων λογισμικού καθώς η εξαρχής σχεδίαση και υλοποίηση ωρολογίων προγραμμάτων αποτελεί δύσκολο έργο και δεν αποτελεί αντικείμενο της εργασίας μας. Από τα διαθέσιμα έτοιμα συστήματα ωρολογίων προγραμμάτων [10] διαπιστώσαμε πως η πλειοψηφία τους είναι για εμπορική χρήση, ενώ λιγοστά είναι διαθέσιμα για ελεύθερη χρήση. Από τα διαθέσιμα συστήματα ως πιο κατάλληλο κρίναμε το FET [11],[12] καθώς χρησιμοποιείται αρκετά διεθνώς ενώ προσφέρεται η δυνατότητα ενσωμάτωσής του σε άλλα συστήματα.

1.2.2 Τεχνολογικά εργαλεία

Για την υλοποίηση του συστήματος προτείνεται η χρήση των εξής τεχνολογιών/εργαλείων: Oracle Database 12g (PL/SQL functions, procedures, triggers), Java/Java EE [13] (Context and Dependency Injection, Bean Validation, Java Persistence API, Object Relational Mapping, Java Server Faces, XML and JSON Processing, Restful Web Services [14]), Maven [15], PrimeFaces [16], Netbeans 8 EE [17], Eclipse, JBOSS Application Server(Wildfly Edition 8.2.0) [18], jBPM6 [19][20] και Android Studio [21]. Πρόκειται για ισχυρά εργαλεία στον τομέα ανάπτυξης επιχειρησιακών εφαρμογών με δωρεάν χρήση - με εξαίρεση τη βάση δεδομένων Oracle.

Η αρχιτεκτονική των διαδικτυακών εφαρμογών που βασίζονται στις παραπάνω τεχνολογίες ακολουθεί το πρότυπο Model View Controller (MVC) που έχει ως βασικό στόχο το διαχωρισμό του κώδικα διεπαφής του χρήστη (user interface) από τον κώδικα της επιχειρησιακής λογικής (business logic). Όταν ο κώδικας αυτών των δυο τμημάτων δεν είναι διαχωρισμένος τότε το λογισμικό είναι πολύ δύσκολο να συντηρηθεί και δεν επιδέχεται κλιμάκωση, ενώ ο σαφής διαχωρισμός τους επιτρέπει εύκολες ενημερώσεις και στο τμήμα της διεπαφής αλλά και στο τμήμα της επιχειρησιακής λογικής. Σε αυτή την αρχιτεκτονική το τμήμα Model αναπαριστά τα δεδομένα της εφαρμογής, το τμήμα View αντιστοιχεί στη διεπαφή και το τμήμα Controller

(λογική της εφαρμογής/application logic) διαχειρίζεται τη διασύνδεση των άλλων δυο τμημάτων [13]. Η αρχιτεκτονική MVC ορίζει πως κάθε αντικείμενο της εφαρμογής θα ανήκει ξεκάθαρα είτε στο τμήμα Model, είτε στο τμήμα View είτε στο τμήμα Controller [21].

1.2.3 Business Process Management systems

Εξαιρετικό ενδιαφέρον παρουσιάζουν τα Business Process Management systems (BPMs) [19][20]. Η ιδέα των BPMs είναι πως οι λειτουργίες που εκτελεί ένας οργανισμός ή μια επιχείρηση αντιμετωπίζονται ως διεργασίες (processes). Κάθε διεργασία -που θα μπορούσε ίσως να οριστεί και ως ροή εργασιών (work flow)- μοντελοποιείται ως ένα σύνολο εργασιών (Tasks) τα οποία εκτελούνται με κάποια σειρά: Υπάρχει Αρχή-->Εργασία1-->Εργασία2-->Εργασία3-->Τέλος. Η μετάβαση από εργασία (Task) σε εργασία (Task) δεν είναι απαραίτητα ακολουθιακή, αλλά μπορεί να υπάρχουν διακλαδώσεις, μπορεί κάποια Tasks να εκτελούνται παράλληλα ή μπορεί να ενεργοποιούνται/απενεργοποιούνται και με γεγονότα (Events). Επίσης υπάρχουν διάφορα ήδη Tasks (π.χ. Human Tasks, Service Tasks κ.α.). Επιπλέον κατά τη μετάβαση από Task σε Task μπορεί να στέλνονται δεδομένα εισόδου/εξόδου με χρήση μεταβλητών ή αντικειμένων. Η μοντελοποίηση γίνεται με την τυποποιημένη γλώσσα BPMN 2.0 (Business Process Model and Notation Version 2.0) [19], ενώ η σχεδίαση του μοντέλου γίνεται από γραφικά εργαλεία (όπως το Eclipse-KEPLER [22]) και το αποτέλεσμα είναι ένα αρχείο XML [20]. Αν κατασκευαστεί ένα μοντέλο μιας διεργασίας με τη γλώσσα BPMN τότε αυτό το μοντέλο το βλέπουν όλοι οι επιθυμητοί χρήστες (Business Users, Developers, απλοί χρήστες) που εμπλέκονται στη συγκεκριμένη διεργασία και βοηθά στο συντονισμό τους. Η δυναμική ενός συστήματος BPM είναι πως εξασφαλίζει την οργάνωση και το συντονισμό του συνόλου των χρηστών (που σχετίζονται με τη διεργασία) που μπορεί να εργάζονται σε διαφορετικά τμήματα εντός (ή κ εκτός) του οργανισμού. Κάθε χρήστης χρησιμοποιεί συνήθως και κάποια(ες) εφαρμογές. Δηλαδή με την έναρξη μιας διεργασίας ειδοποιείται την κατάλληλη στιγμή ο κατάλληλος χρήστης για να εκτελέσει την εργασία (task) που του έχει ανατεθεί, και μόλις το ολοκληρώσει αυτόματα ενημερώνεται ο επόμενος κατάλληλος χρήστης ή κάποια αυτοματοποιημένη (automated service) εργασία κλπ. Η ειδοποίηση των χρηστών μπορεί να γίνει με πολλούς τρόπους (e-mail, social media, sms, application subsystem κλπ). Η υλοποίηση ενός συστήματος BPM επιτρέπει τη διασύνδεση των εφαρμογών(συστήματα/υποσυστήματα) που χρησιμοποιούν οι χρήστες ώστε να γίνεται άμεσα η ενημέρωσή τους και προάγει σημαντικά την ποιότητα των υπηρεσιών που προσφέρονται στον οργανισμό ή στην επιχείρηση.

Πλεονεκτήματα

Τα πλεονεκτήματα [23] ενός BPM system συνοψίζονται στα εξής:

Visualization: Οι επιθυμητοί χρήστες έχουν εικόνα της διεργασίας και της φάσης εξέλιξής της.

Monitoring: Οι Business Managers μπορούν να παρακολουθούν την εξέλιξη των εργασιών, ενώ μπορούν να οριστούν μετρικές (metrics) για κάθε εργασία ή διεργασία.

Auditing/Decision Support: Από στοιχεία που καταγράφονται και από τις μετρικές, οι Business Managers μπορούν να εκτιμήσουν την εκτέλεση των εργασιών-διεργασιών για να παρθούν αποφάσεις προς βελτιστοποίηση της απόδοσης.

Δωρεάν υλοποίηση: Η υλοποίηση ενός BPM συστήματος μπορεί να υλοποιηθεί χωρίς κόστος κάνοντας χρήση ελεύθερων, αλλά ισχυρών λογισμικών.

Μειονεκτήματα

Στα μειονεκτήματα συγκαταλέγονται τα εξής:

Πολυπλοκότητα: Η διασύνδεση διαφορετικών ομοιογενών και ανομοιογενών συστημάτων είναι αναμφίβολα δύσκολο έργο. Η ομάδα ανάπτυξης του λογισμικού πρέπει να έχει την απαραίτητη εμπειρία και εξοικείωση με τις συγκεκριμένες τεχνολογίες.

Εκπαίδευση χρηστών: Οι χρήστες του οργανισμού θα πρέπει να εκπαιδευτούν ώστε να εξοικειωθούν με τη φιλοσοφία των συστημάτων BPM.

Χρόνος ενσωμάτωσης: Η ενσωμάτωση ενός BPM system απαιτεί προσεκτικό σχεδιασμό, ενώ σε συνδυασμό με την ετερογένεια των χρηστών και της πολυπλοκότητας απαιτεί αρκετό χρόνο μέχρι την υλοποίηση.

1.3 Δήλωση της Υπόθεσης

Το βασικό ερευνητικό ερώτημα είναι το εξής: Μπορεί να βελτιωθεί η διαχείριση της εκτέλεσης των προγραμμάτων σπουδών από τα τμήματα των ελληνικών τριτοβάθμιων ιδρυμάτων ενοποιώντας τις υπάρχουσες λειτουργίες που προσφέρονται από το υφιστάμενο πληροφοριακό σύστημα με τις λειτουργίες που αναφέρθηκαν στην ενότητα 1.1; Δεδομένου ότι το υφιστάμενο πληροφοριακό σύστημα θα διατηρηθεί προτείνουμε την δημιουργία ενός υποστηρικτικού πληροφοριακού συστήματος το οποίο θα είναι ενοποιημένο/συνδεδεμένο σε πραγματικό χρόνο με το υφιστάμενο γραμματειακό σύστημα ώστε οι μεταβολές π.χ. σε δηλώσεις μαθημάτων ή τροποποιήσεις βαθμών από το γραμματειακό σύστημα, συνταξιοδοτήσεις καθηγητών κ.α. να αντικατοπτρίζονται ομαλά στο νέο σύστημα χωρίς να δημιουργούνται προβλήματα στις άλλες υποστηριζόμενες λειτουργίες.

Ευελπιστούμε πως τα εκπαιδευτικά ιδρύματα που θα υποστηρίζονται από ένα τέτοιο σύστημα θα προσφέρουν καλύτερη ποιότητα εκτέλεσης των προγραμμάτων σπουδών τους με παράλληλη εξοικονόμηση πόρων σε σχέση με τα υπόλοιπα εκπαιδευτικά ιδρύματα.

Στις επόμενες παραγράφους διατυπώνουμε συγκεκριμένες ερευνητικές υποθέσεις. Ο έλεγχος ωστόσο των υποθέσεων δεν έγινε στο πλαίσιο της διατριβής για πρακτικούς λόγους, αλλά καθορίστηκε η μεθοδολογία και το πλάνο επαλήθευσής τους (συγκέντρωση των δεδομένων, οι στατιστικές διαδικασίες και ο τρόπος ερμηνείας των αποτελεσμάτων).

1.3.1 Υπόθεση 1

Στο προτεινόμενο σύστημα υποστήριξης προγραμμάτων σπουδών η αυτοματοποιημένη δυνατότητα κατασκευής ωρολογίου προγράμματος σε συνδυασμό με τη διασύνδεσή του με το υφιστάμενο γραμματειακό σύστημα, αναμένουμε να περιορίσει σημαντικά το χρόνο οριστικοποίησης του ωρολογίου προγράμματος ανά ακαδημαϊκό εξάμηνο. Αντιθέτως ιδρύματα που διαθέτουν συστήματα χωρίς να είναι συνδεδεμένα με το γραμματειακό τους σύστημα καθώς και ιδρύματα που δε διαθέτουν συστήματα αναμένουμε να έχουν αυξημένο χρόνο και κόπο οριστικοποίησης των ωρολογίων προγραμμάτων τους.

1.3.2 Υπόθεση 2

Αναμένουμε σημαντική μείωση των παραπόνων από τους φοιτητές που θα εκτελούν δηλώσεις ομάδων μαθημάτων με χρήση του συστήματος υποστήριξης προγραμμάτων σπουδών σε σχέση με τους φοιτητές που πραγματοποιούν δηλώσεις ομάδων μαθημάτων με τον παραδοσιακό χειρόγραφο τρόπο ή με άλλα συστήματα που δεν είναι συνδεδεμένα με το υφιστάμενο γραμματειακό σύστημα.

1.3.3 Υπόθεση 3

Στο προτεινόμενο σύστημα υποστήριξης προγραμμάτων σπουδών η αυτοματοποιημένη δυνατότητα δηλώσεων ομάδων μαθημάτων σε συνδυασμό με τη διασύνδεσή του με το υφιστάμενο γραμματειακό σύστημα, αναμένουμε να περιορίσει σημαντικά το χρόνο που απαιτείται ανθρώπινη παρέμβαση από το τεχνικό προσωπικό του τμήματος πληροφορικής του ιδρύματος. Αντιθέτως ιδρύματα που διαθέτουν συστήματα χωρίς να είναι συνδεδεμένα με το γραμματειακό τους σύστημα απαιτούν αυξημένη τεχνική υποστήριξη, ενώ ιδρύματα που διαθέτουν συστήματα δεν απαιτούν τεχνική υποστήριξη αλλά δεν παρέχουν και ποιότητα στην εκτέλεση των προγραμμάτων σπουδών τους.

1.4 Σπουδαιότητα της μελέτης

Λόγω μείωσης του διαθέσιμου προσωπικού και των οικονομικών πόρων πολλές φορές δεν είναι εφικτή η ανάθεση μαθημάτων σε εκπαιδευτικό προσωπικό πριν ξεκινήσει το ακαδημαϊκό εξάμηνο. Θα ήταν ιδανικό αν ξεκινούσε η ανάθεση μαθημάτων μετά την ολοκλήρωση των δηλώσεων ομάδων μαθημάτων ή αν μπορούσε να γίνει από νωρίς σωστή πρόβλεψη των ομάδων μαθημάτων. Χωρίς πληροφοριακό σύστημα να υποστηρίζει τη λειτουργία της ανάθεσης εμφανίζονται φαινόμενα ανάθεσης μαθημάτων σε εκπαιδευτικούς μετά την έναρξη του ακαδημαϊκού έτους/εξαμήνου ή φαινόμενα τροποποίησης συμβάσεων ανάθεσης μαθημάτων. Δεν είναι απίθανο να χαθούν μαθήματα από το ακαδημαϊκό εξάμηνο για τέτοιους λόγους.

Οι δημιουργίες ομάδων παρακολούθησης μαθημάτων σε αρκετά ιδρύματα πραγματοποιούνται χειρόγραφα με μεγάλους συνωστισμούς φοιτητών. Αρκετοί φοιτητές ή εκπαιδευτικοί (για διαφορετικούς ουσιαστικούς λόγους) αδυνατούν να παρουσιαστούν αυτοπροσώπως στην αίθουσα για εγγραφή σε ομάδα παρακολούθησης. Επιπλέον σπαταλιέται πολύτιμος χρόνος για

όλους. Ακόμα και αν χρησιμοποιούνται κάποια άλλα συστήματα πληροφορικής για τη δημιουργία των ομάδων αυτά αντιμετωπίζουν πρόβλημα ενημέρωσης αν δεν είναι ταυτόχρονα ενοποιημένα με το υφιστάμενο γραμματειακό σύστημα. Για παράδειγμα, ένας φοιτητής μπορεί να δηλώσει ομάδα χωρίς να έχει δηλώσει το μάθημα επίσημα στο γραμματειακό σύστημα ή μπορεί να το έχει δηλώσει σωστά αλλά να κάνει τροποποίηση της δήλωσής του αργότερα στη γραμματεία. Συμβαίνει επίσης πολύ συχνά οι περίοδοι δηλώσεων μαθημάτων στο γραμματειακό σύστημα να συμπίπτουν χρονικά με τις δηλώσεις ομάδων μαθημάτων. Αυτό έχει ως αποτέλεσμα να εμφανίζονται εσφαλμένες λίστες ομάδων παρακολούθησης μαθημάτων.

Η δημιουργία του εβδομαδιαίου ωρολογίου προγράμματος πολύ συχνά δημιουργείται χρονικά παράλληλα με τις δηλώσεις ομάδων παρακολούθησης μαθημάτων ή και δηλώσεις μαθημάτων ή και με τις αναθέσεις, οπότε δεν υπάρχει σωστή εικόνα με αποτέλεσμα να ενημερώνεται συνεχώς και να παρατηρούνται καθυστερήσεις στην ουσιαστική έναρξη του εξαμήνου. Επιπλέον, η κατασκευή του πραγματοποιείται με χειρόγραφο τρόπο. Θα ήταν ιδανική η αυτοματοποιημένη κατασκευή του βάση κριτηρίων που θέτουν οι εκπαιδευτικοί και με βάση την πραγματική εικόνα των δηλώσεων ομάδων μαθημάτων η οποία μεταβάλλεται δυναμικά. Πολύ συχνά εμφανίζεται το φαινόμενο το ωρολόγιο πρόγραμμα να δημιουργείται με ιδιότελή κριτήρια με αποτέλεσμα να αδικούνται εκπαιδευτικοί ή φοιτητές.

Ανάλογα προβλήματα εμφανίζονται αν δεν εκτελεστούν οι δηλώσεις ομάδων των φοιτητών με αυτοματοποιημένο τρόπο βάση κριτηρίων.

Η συνολική αντίληψη που διαμορφώνουν οι χρήστες για τα πληροφοριακά συστήματα αντανακλούν τη φήμη των ιδρυμάτων, εξασφαλίζουν κλίμα ενδιαφέροντος και επηρεάζουν είτε άμεσα είτε έμμεσα την απόδοση/ποιότητα της εκπαιδευτικής διαδικασίας. Η μη αποτελεσματική επίλυση των παραπάνω ζητημάτων δημιουργεί αρνητική φήμη, μειώνει το κλίμα ενδιαφέροντος και ζημιώνει την απόδοση/ποιότητα της εκπαιδευτικής διαδικασίας.

Το υποστηρικτικό σύστημα που παρουσιάζεται σε αυτή τη διατριβή στοχεύει στην αντιμετώπιση των παραπάνω ζητημάτων.

1.5 Μεθοδολογία

Στις επόμενες παραγράφους παρουσιάζεται με λεπτομέρεια πώς θα πραγματοποιηθεί η έρευνα ώστε να εξεταστούν οι υποθέσεις της ενότητας 1.3 . Για το σκοπό αυτό εξετάζουμε τους πληθυσμούς ενδιαφέροντος, τα όργανα μετρήσεων και τις διαδικασίες εκτέλεσης του πειράματος.

1.5.1 Πληθυσμοί

Οι πληθυσμοί ενδιαφέροντος αυτής της μελέτης είναι οι εκπαιδευτικοί, οι φοιτητές και το τεχνικό προσωπικό του γενικού τμήματος πληροφορικής. Το πείραμα θα διεξαχθεί στο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Ανατολικής Μακεδονίας και Θράκης (ΤΕΙ ΑΜΘ). Το συγκρότημα του ιδρύματος αποτελείται από 10 εκπαιδευτικά τμήματα:

Μηχανικών Πληροφορικής (10 εκπαιδευτικοί, 1100 φοιτητές),

Λογιστική (15 εκπαιδευτικοί, 2500 φοιτητές),

Μηχανολογία (12 εκπαιδευτικοί, 1000 φοιτητές),

Διοίκηση Επιχειρήσεων (18 εκπαιδευτικοί, 2600 φοιτητές),

Τεχνολογία Πετρελαίου (10 εκπαιδευτικοί, 1150 φοιτητές),

Ηλεκτρολογία (11 εκπαιδευτικοί, 1200 φοιτητές),

Αρχιτεκτονική Τοπίου (8 εκπαιδευτικοί, 700 φοιτητές),

Δασοπονία (7 εκπαιδευτικοί, 750 φοιτητές),

Νοσηλευτική (12 εκπαιδευτικοί, 900 φοιτητές).

Το τεχνικό προσωπικό του γενικού τμήματος πληροφορικής του ιδρύματος αποτελείται από 4 άτομα.

1.5.2 Όργανα

Για τη δυνατότητα σύγκρισης του συμβατικού/παραδοσιακού τρόπου εκτέλεσης προγραμμάτων σπουδών σε σχέση με το νέο σύστημα υποστήριξης εισάγουμε τους όρους ποιότητα συστήματος και οικονομία συστήματος. *Ποιοτικό* είναι το σύστημα που υποστηρίζει αρκετές και χωρίς προβλήματα λειτουργίες που σχετίζονται με την εκτέλεση των προγραμμάτων σπουδών. *Οικονομικό* είναι το σύστημα που εξασφαλίζει τα πλεονεκτήματα του ποιοτικού συστήματος με την ελάχιστη χρονική ανθρώπινη υποστήριξη.

Για τη μέτρηση της ποιότητας και της οικονομίας των δυο τρόπων εκτέλεσης των προγραμμάτων σπουδών χρησιμοποιούμε τα εξής όργανα:

Ρυθμός απόδοσης δημιουργίας ωρολογίου προγράμματος $Etm(p)$: Ορίζεται ως ο χρόνος σε λεπτά (m) οριστικοποίησης ενός ωρολογίου προγράμματος (t) ενός τμήματος σε μια ακαδημαϊκή περίοδο(p). Για τους υπολογισμούς απαιτούνται δυο ακαδημαϊκές περίοδοι για κάθε τμήμα. Στην πρώτη ακαδημαϊκή περίοδο θα υπολογιστεί ο χρόνος οριστικοποίησης του ωρολογίου προγράμματος με το παραδοσιακό τρόπο ενώ στη δεύτερη ακαδημαϊκή περίοδο θα μετρηθεί ο χρόνος με χρήση του συστήματος υποστήριξης προγραμμάτων σπουδών.

Ρυθμός εμφάνισης παραπόνων από φοιτητές κατά τις δηλώσεις ομάδων μαθημάτων σε συγκεκριμένη ακαδημαϊκή περίοδο τμήματος $Elp(p)$: Ορίζεται ως το άθροισμα των παραπόνων(lp) για τη δήλωση ομάδων μαθημάτων όλων των μαθημάτων(N) προς το πλήθος των ενεργών φοιτητών της ακαδημαϊκής περιόδου (sn).

$$Elp(p) = \frac{\sum_{l=1}^N lp_l}{sn}$$

Για τους υπολογισμούς μας απαιτούνται δυο ακαδημαϊκές περίοδοι για κάθε τμήμα. Στην πρώτη ακαδημαϊκή περίοδο θα υπολογιστεί το άθροισμα των παραπόνων όταν οι δηλώσεις ομάδων γίνονται με το παραδοσιακό τρόπο ενώ στη δεύτερη ακαδημαϊκή περίοδο θα υπολογιστεί το άθροισμα των παραπόνων όταν οι δηλώσεις ομάδων γίνονται με το σύστημα υποστήριξης προγραμμάτων σπουδών.

Ρυθμός ενασχόλησης του προσωπικού του γενικού τμήματος πληροφορικής για την υποστήριξη της εκτέλεσης ενός προγράμματος σπουδών ενός τμήματος για συγκεκριμένη ακαδημαϊκή περίοδο $Esm(p)$: Ορίζεται ως ο χρόνος σε λεπτά (sm) που απασχολείται το τμήμα πληροφορικής για την

υποστήριξη ενός προγράμματος σπουδών συγκεκριμένων απαιτήσεων σε μια ακαδημαϊκή περίοδο(p). Για τους υπολογισμούς μας απαιτούνται δυο ακαδημαϊκές περιόδους για κάθε τμήμα. Στην πρώτη ακαδημαϊκή περίοδο θα υπολογιστεί ο χρόνος υποστήριξης με το παραδοσιακό τρόπο ενώ στη δεύτερη ακαδημαϊκή περίοδο θα μετρηθεί ο χρόνος με χρήση του συστήματος υποστήριξης προγραμμάτων σπουδών.

1.5.3 Διαδικασίες

Για τη δίκαιη εκτέλεση του πειράματος στο ίδρυμα Ανατολικής Μακεδονίας και Θράκης θα απαιτηθεί χρόνος δυο ακαδημαϊκών ετών ή τεσσάρων ακαδημαϊκών περιόδων. Στις δυο πρώτες ακαδημαϊκές περιόδους τα προγράμματα σπουδών θα εκτελεστούν με τον παραδοσιακό τρόπο, ενώ στις επόμενες δύο τα προγράμματα σπουδών θα εκτελεστούν χρησιμοποιώντας το σύστημα υποστήριξης προγραμμάτων σπουδών. Ο λόγος για τον οποίο χρειαζόμαστε τέσσερις ακαδημαϊκές περιόδους και όχι δύο είναι γιατί στη χειμερινή περίοδο πραγματοποιούνται διαφορετικά μαθήματα από την εαρινή περίοδο. Συνεπώς θεωρούμε πιο ορθό να γίνουν συγκριτικές μετρήσεις μεταξύ παραδοσιακού και νέου τρόπου εκτέλεσης προγραμμάτων σπουδών πάνω στα ίδια μαθήματα, ώστε να εξασφαλιστεί μεγαλύτερη ακρίβεια στις μετρήσεις.

Σε περίπτωση που κατά την εξέλιξη του πειράματος πραγματοποιηθεί σημαντική δομική μεταβολή κάποιου μαθήματος (π.χ. αλλάξει ακαδημαϊκή περίοδο ή από μεικτό μάθημα γίνει απλό θεωρητικό ή εργαστηριακό ή οτιδήποτε άλλο που θα μπορούσε να επηρεάσει τη συνέπεια των μετρήσεων του όλου πειράματος) τότε οι μετρήσεις του μαθήματος θα εξαιρεθούν από τις συνολικές μετρήσεις του πειράματος.

Οι μετρήσεις των παραπάνω οργάνων πραγματοποιούνται στα δυο ακαδημαϊκά έτη, μια με τον παραδοσιακό τρόπο και μια με το σύστημα υποστήριξης των προγραμμάτων σπουδών.

Για το μέγεθος του συγκεκριμένου ιδρύματος θα ήταν ιδανικό -για την αποτελεσματική εκτέλεση των μετρήσεων των οργάνων του πειράματος- την τεχνική υποστήριξη των προγραμμάτων σπουδών να την αναλάβει ένας μόνο διαχειριστής από το τμήμα πληροφορικής.

Για την αποτελεσματική λειτουργία του συστήματος υποστήριξης προγραμμάτων σπουδών θα χρειαστεί εκπαίδευση των χρηστών που θα το χρησιμοποιήσουν. Η εκπαίδευση αυτή θα πρέπει να γίνει με τη λήξη του πρώτου ακαδημαϊκού έτους και πριν την έναρξη του δεύτερου. Αυτό

κρίνεται σκόπιμο καθώς σε οποιαδήποτε άλλη χρονική στιγμή η εκπαίδευση θα μπορούσε να επηρεάσει τα αποτελέσματα των μετρήσεων.

1.6 Ανάλυση Δεδομένων

Σε αυτή την ενότητα σχεδιάζουμε την οργάνωση των δεδομένων και ορίζουμε τις στατιστικές διαδικασίες αξιολόγησης για καθεμιά από τις υποθέσεις της ενότητας 1.3.

1.6.1 Υπόθεση 1

Στο προτεινόμενο σύστημα υποστήριξης προγραμμάτων σπουδών η αυτοματοποιημένη δυνατότητα κατασκευής ωρολογίου προγράμματος σε συνδυασμό με τη διασύνδεσή του με το υφιστάμενο γραμματειακό σύστημα, αναμένουμε να *περιορίσει σημαντικά το χρόνο οριστικοποίησης* του ωρολογίου προγράμματος ανά ακαδημαϊκό εξάμηνο. Αντιθέτως, ιδρύματα που διαθέτουν συστήματα χωρίς να είναι συνδεδεμένα με το γραμματειακό τους σύστημα καθώς και ιδρύματα που δε διαθέτουν συστήματα αναμένουμε να *έχουν αυξημένο χρόνο οριστικοποίησης* των ωρολογίων προγραμμάτων τους.

Οργάνωση δεδομένων της υπόθεσης

Μεταβλητή	Πεδίο Τιμών	Τύπος μεταβλητής		Κλίμακα μέτρησης
Τύπος συστήματος	{Νέο Σύστημα, Άλλο σύστημα (μη ενοποιημένο με το γραμματειακό ή καθόλου σύστημα)}	Ανεξάρτητη		Ονομαστική/Κατηγορική
Χρόνος(σε λεπτά) για οριστικοποίηση όλων των ωρολογίων προγραμμάτων του ιδρύματος	$N=\{0,1,2,3,\dots\}$	Εξαρτημένη	Δισδιάστατος Πίνακας [3]	Αναλογιών/Συνεχής
Χρόνος(σε λεπτά) για οριστικοποίηση	$N=\{0,1,2,3,\dots\}$	Εξαρτημένη	Τρισδιάστατος Πίνακας[3][Πλήθος	Αναλογιών/Συνεχής

ωρολογίου προγράμματος/Τμήμα			Τμημάτων]	
---------------------------------	--	--	-----------	--

Πίνακας 1.1: Οργάνωση δεδομένων υπόθεσης 1

Οργάνωση των δεδομένων σε πίνακα: **Frequency Distribution**

Παρουσίαση των δεδομένων με τα γραφήματα: **Histogram** και **Frequency Polygon, Scatter plot**.

Στατιστικές διαδικασίες της υπόθεσης

Μεταβλητή	Statistical procedure		Περιγραφή
Τύπος συστήματος			
Χρόνος(σε λεπτά) για οριστικοποίηση όλων των ωρολογίων προγραμμάτων του ιδρύματος	Central Tendency	-	
	Variability	-	
	Location	-	
	Correlation	-	
Χρόνος(σε λεπτά) για οριστικοποίηση ωρολογίου προγράμματος/Τμήμα	Central Tendency	Difference between means, Effect size (Δ)	Αναμένουμε αρνητική ισχυρή συσχέτιση.
	Variability	Difference between standard deviation of variances	
	Location	z-scores,	Η διαφορά των χρόνων οριστικοποίησης ωρολογίου προγράμματος ενός τμήματος στην κατανομή του συστήματος υποστήριξης προγραμμάτων σπουδών και στην κατανομή

		stanine scores	του παραδοσιακού τρόπου εκτέλεσης προγράμματος σπουδών. Ερμηνεία της διαφοράς των χρόνων οριστικοποίησης ωρολογίου προγράμματος ενός τμήματος στην κατανομή του συστήματος υποστήριξης προγραμμάτων σπουδών και στην κατανομή του παραδοσιακού τρόπου εκτέλεσης προγράμματος σπουδών (για να φανεί η σημαντικότητά της)
	Correlation	Διαφορά Pearson r για τις μεταβλητές των χρόνων οριστικοποίησης ωρολογίου προγράμματος των τμημάτων για τις δυο ομάδες (νέο σύστημα υποστήριξης κ παραδοσιακό σύστημα)	Αναμένουμε αρνητική ισχυρή συσχέτιση.

Πίνακας 1.2: Στατιστικές διαδικασίες υπόθεσης 1

1.6.2 Υπόθεση 2

Αναμένουμε σημαντική μείωση των παραπόνων από τους φοιτητές που θα εκτελούν δηλώσεις ομάδων μαθημάτων με χρήση του συστήματος υποστήριξης προγραμμάτων σπουδών σε σχέση με τους φοιτητές που πραγματοποιούν δηλώσεις ομάδων μαθημάτων με τον παραδοσιακό χειρόγραφο τρόπο ή με άλλα συστήματα που δεν είναι συνδεδεμένα με το υφιστάμενο γραμματειακό σύστημα.

Οργάνωση δεδομένων της υπόθεσης

Μεταβλητή	Πεδίο Τιμών	Τύπος μεταβλητής	Κλίμακα μέτρησης

Τύπος συστήματος	{Νέο Σύστημα, Άλλο σύστημα (μη ενοποιημένο με το γραμματειακό ή καθόλου σύστημα)}	Ανεξάρτητη		Ονομαστική/Κατηγορική
Πλήθος παραπόνων κατά τη διαδικασία δήλωσης ομάδων μαθημάτων του ιδρύματος	$N=\{0,1,2,3,\dots\}$	Εξαρτημένη	Δισδιάστατος Πίνακας [3]	Αναλογιών/Συνεχής
Πλήθος παραπόνων κατά τη διαδικασία δήλωσης ομάδων μαθημάτων του ιδρύματος /Τμήμα	$N=\{0,1,2,3,\dots\}$	Εξαρτημένη	Τρισδιάστατος Πίνακας[3][Πλήθος Τμημάτων]	Αναλογιών/Συνεχής

Πίνακας 1.3: Οργάνωση δεδομένων υπόθεσης 2

Οργάνωση των δεδομένων σε πίνακα: **Frequency Distribution**

Παρουσίαση των δεδομένων με τα γραφήματα: **Histogram** και **Frequency Polygon, Scatter plot**.

Στατιστικές διαδικασίες της υπόθεσης

Μεταβλητή	Statistical procedure		Περιγραφή
Τύπος συστήματος			
Πλήθος παραπόνων κατά τη διαδικασία δήλωσης ομάδων μαθημάτων του ιδρύματος	Central Tendency	-	
	Variability	-	
	Location	-	

	Correlation	-	
Πλήθος παραπόνων κατά τη διαδικασία δήλωσης ομάδων μαθημάτων του ιδρύματος /Τμήμα	Central Tendency	Difference between means, Effect size (Δ)	Αναμένουμε αρνητική ισχυρή συσχέτιση.
	Variability	Difference between standard deviation of variances	
	Location	z-scores, stanine scores	Η διαφορά του πλήθους παραπόνων κατά τη διαδικασία της δήλωσης ομάδων μαθημάτων ενός τμήματος στην κατανομή του συστήματος υποστήριξης προγραμμάτων σπουδών και στην κατανομή του παραδοσιακού τρόπου εκτέλεσης προγράμματος σπουδών. Ερμηνεία της διαφοράς του πλήθους παραπόνων κατά τη διαδικασία της δήλωσης ομάδων μαθημάτων ενός τμήματος στην κατανομή του συστήματος υποστήριξης προγραμμάτων σπουδών και στην κατανομή του παραδοσιακού τρόπου εκτέλεσης προγράμματος σπουδών (για να φανεί η σημαντικότητά της)
	Correlation	Διαφορά Pearson r για τις μεταβλητές του πλήθους παραπόνων των τμημάτων για τις δυο ομάδες (νέο σύστημα υποστήριξης κ παραδοσιακό σύστημα)	Αναμένουμε αρνητική ισχυρή συσχέτιση.

Πίνακας 1.4 Στατιστικές διαδικασίες υπόθεσης 2

1.6.3 Υπόθεση 3

Στο προτεινόμενο σύστημα υποστήριξης προγραμμάτων σπουδών η αυτοματοποιημένη δυνατότητα δηλώσεων ομάδων μαθημάτων σε συνδυασμό με τη διασύνδεσή του με το υφιστάμενο γραμματειακό σύστημα, αναμένουμε να περιορίσει σημαντικά το χρόνο που απαιτείται ανθρώπινη παρέμβαση από το τεχνικό προσωπικό του τμήματος πληροφορικής του ιδρύματος. Αντιθέτως, ιδρύματα που διαθέτουν συστήματα χωρίς να είναι συνδεδεμένα με το γραμματειακό τους σύστημα απαιτούν αυξημένη τεχνική υποστήριξη, ενώ ιδρύματα που διαθέτουν συστήματα δεν απαιτούν τεχνική υποστήριξη αλλά δεν παρέχουν και ποιότητα στην εκτέλεση των προγραμμάτων σπουδών τους.

Οργάνωση δεδομένων της υπόθεσης

Μεταβλητή	Πεδίο Τιμών	Τύπος μεταβλητής		Κλίμακα μέτρησης
Τύπος συστήματος	{Νέο Σύστημα, Άλλο σύστημα (μη ενοποιημένο με το γραμματειακό ή καθόλου σύστημα)}	Ανεξάρτητη		Ονομαστική/Κατηγορική
Χρόνος(σε λεπτά) για την τεχνική υποστήριξη των δηλώσεων ομάδων μαθημάτων του ιδρύματος	$N=\{0,1,2,3,\dots\}$	Εξαρτημένη	Δισδιάστατος Πίνακας [3]	Αναλογιών/Συνεχής
Χρόνος(σε λεπτά) για την τεχνική υποστήριξη των δηλώσεων ομάδων μαθημάτων του ιδρύματος /Τμήμα	$N=\{0,1,2,3,\dots\}$	Εξαρτημένη	Τρισδιάστατος Πίνακας[3][Πλήθος Τμημάτων]	Αναλογιών/Συνεχής

Πίνακας 1.5: Οργάνωση δεδομένων υπόθεσης 3

Οργάνωση των δεδομένων σε πίνακα: **Frequency Distribution**

Παρουσίαση των δεδομένων με τα γραφήματα: **Histogram** και **Frequency Polygon, Scatter plot**.

Στατιστικές διαδικασίες της υπόθεσης

Μεταβλητή	Statistical procedure		Περιγραφή
Τύπος συστήματος			
Χρόνος(σε λεπτά) για την τεχνική υποστήριξη των δηλώσεων ομάδων μαθημάτων του ιδρύματος	Central Tendency	-	
	Variability	-	
	Location	-	
	Correlation	-	
Χρόνος(σε λεπτά) για την τεχνική υποστήριξη των δηλώσεων ομάδων μαθημάτων του ιδρύματος /Τμήμα	Central Tendency	Difference between means, Effect size (Δ)	Αναμένουμε αρνητική ισχυρή συσχέτιση.
	Variability	Difference between standard deviation of variances	
	Location	z-scores,	Η διαφορά των χρόνων για την τεχνική υποστήριξη των δηλώσεων ομάδων μαθημάτων ενός τμήματος στην κατανομή του συστήματος υποστήριξης προγραμμάτων σπουδών και στην κατανομή του παραδοσιακού τρόπου εκτέλεσης προγράμματος σπουδών. Ερμηνεία της διαφοράς για την τεχνική υποστήριξη των δηλώσεων ομάδων

		stanine scores	μαθημάτων ενός τμήματος στην κατανομή του συστήματος υποστήριξης προγραμμάτων σπουδών και στην κατανομή του παραδοσιακού τρόπου εκτέλεσης προγράμματος σπουδών (για να φανεί η σημαντικότητά της)
	Correlation	Διαφορά Pearson r για τις μεταβλητές των χρόνων δηλώσεων ομάδων μαθημάτων των τμημάτων για τις δυο ομάδες (νέο σύστημα υποστήριξης κ παραδοσιακό σύστημα)	Αναμένουμε αρνητική ισχυρή συσχέτιση.

Πίνακας 1.6 Στατιστικές διαδικασίες υπόθεσης 3

1.7 Χρονοδιάγραμμα Εργασιών

Εργασίες	Διάρκεια(ημέρες)	Έναρξη	Λήξη
<p>1.Βιβλιογραφική μελέτη*</p> <p>-Βιβλιογραφική μελέτη στην μεθοδολογία σχεδιασμού και υλοποίησης εφαρμογών σύνθετων πληροφοριακών συστημάτων με σύγχρονες Τεχνολογίες Web 2.0</p> <p>-Βιβλιογραφική μελέτη στην μεθοδολογία σχεδιασμού και υλοποίησης εφαρμογών για την πλατφόρμα Android</p> <p><i>*Οι εργασίες της βιβλιογραφικής μελέτης μπορούν να βρεθούν στο συνοδευτικό υλικό της μεταπτυχιακής διατριβής.</i></p>	30	16 Οκτ 2014	15 Νοε 2014
<p>2. Ανάλυση & Σχεδιασμός Συστήματος</p> <p>- Καταγραφή Απαιτήσεων</p> <p>- Εξέταση/Κατανόηση εγκατεστημένων συστημάτων για</p>	60	16 Νοε 2014	15 Ιαν 2015

<p>ενδεχόμενη ενοποίηση.</p> <p>-Σχεδίαση Βάσης Δεδομένων</p> <p>-Αντικειμενοστρεφής Ανάλυση Συστήματος</p>			
<p>3. <i>Ανάπτυξη Λογισμικού</i></p> <p>-Εξοικείωση με τις αναφερόμενες τεχνολογίες Web 2.0 και με τεχνολογίες της πλατφόρμας Android</p> <p>-Συγγραφή Κώδικα</p> <p>-Εντοπισμός Σφαλμάτων</p> <p>-Συγγραφή Τεκμηρίωσης Λογισμικού</p>	60	16 Ιαν 2015	15 Μαρ 2015
<p>4. <i>Εγκατάσταση & Επίβλεψη Συστήματος</i></p> <p>-Σχεδιασμός</p> <p>-Διαχείριση εγκατάστασης & συντονισμός</p> <p>-Έλεγχος & επαλήθευση εγκατάστασης</p> <p>-Παράδοση συστήματος</p> <p>-Αξιολόγηση συστήματος</p>	30	16 Μαρ 2015	15 Απρ 2015
<p>5. <i>Δημοσίευση Επιστημονικού Έργου</i></p> <p>-Αναζήτησης Ενδιαφέροντος σε Συνέδρια ή Περιοδικά</p> <p>-Προετοιμασία Εργασιών προς δημοσίευση των αποτελεσμάτων της έρευνας</p> <p>-Επισκόπηση υπάρχουσας έρευνας και αναζήτηση ερωτημάτων για συνέχιση της έρευνας.</p>	30	15 Απρ 2015	16 Μάη 2015

Πίνακας 1.7 Χρονοδιάγραμμα εργασιών

Κεφάλαιο 2

Ανάλυση/Σχεδίαση Συστήματος

Στο παρόν κεφάλαιο παρουσιάζονται οι λειτουργικές απαιτήσεις του συστήματος και τα διαγράμματα κλάσεων των υποσυστημάτων, η μοντελοποίηση της βάσης δεδομένων και των διεργασιών BPM.

2.1 Λειτουργικές απαιτήσεις

Η υλοποίηση του συστήματος απαιτεί την ανάπτυξη κάποιων υποσυστημάτων και τη διασύνδεση με υπάρχοντα υποσυστήματα ή υπηρεσίες. Τα υποσυστήματα αυτά περιγράφονται στις επόμενες παραγράφους.

2.1.1 Υπάρχοντα ή εξωτερικά υποσυστήματα και υπηρεσίες

Το τεχνολογικό εκπαιδευτικό ίδρυμα Ανατολικής Μακεδονίας και Θράκης διαθέτει υπηρεσία αυθεντικοποίησης και υποσύστημα ηλεκτρονικής γραμματείας.

Υπηρεσία αυθεντικοποίησης (LDAP)

Η υπηρεσία αυθεντικοποίησης (LDAP Server) χρησιμοποιείται για την σύνδεση των χρηστών του ιδρύματος από αρκετές άλλες υπηρεσίες που υπάρχουν στο ίδρυμα. Οι χρήστες μπορεί να είναι μόνιμο ή έκτακτο εκπαιδευτικό προσωπικό, φοιτητές και διοικητικό / τεχνικό προσωπικό. Στο νέο σύστημα επομένως οι χρήστες θα πρέπει να κάνουν σύνδεση με χρήση του LDAP Server.

Υποσύστημα Ηλεκτρονικής Γραμματείας

Το υποσύστημα ηλεκτρονικής γραμματείας είναι λογισμικό που ασχολείται με τη διαχείριση της πορείας των φοιτητών του ιδρύματος. Το σύστημα αυτό για κάθε τμήμα του ιδρύματος διαχειρίζεται κυρίως τα εξής: *προγράμματα σπουδών τμήματος, μαθήματα προγράμματος*

σπουδών, μαθήματα που έχουν ανατεθεί σε καθηγητές, δηλώσεις μαθημάτων και βαθμολόγια φοιτητών. Η πρωτογενής δημιουργία των λογαριασμών των φοιτητών γίνεται επίσης σε αυτό το σύστημα. Το νέο σύστημα θα συνδεθεί με το γραμματειακό σύστημα ώστε να αντλεί πληροφορίες από αυτό.

Λογισμικό FET

Για την κατασκευή του ωρολογίου προγράμματος θα χρησιμοποιηθεί το λογισμικό FET. Το λογισμικό FET μεταξύ των άλλων επιτρέπει την εκτέλεσή του σε μορφή κονσόλας (command line), ενώ η είσοδος και η έξοδος μπορούν να είναι αρχεία XML. Το νέο υποσύστημα ωρολογίου προγράμματος που περιγράφεται στην ενότητα 2.1.2 χρησιμοποιεί το λογισμικό FET.

2.1.2 Νέα υποσυστήματα

Το νέο σύστημα υποστηρίζει αρκετές λειτουργίες και η αρχιτεκτονική του είναι σημαντικό να διαιρείται σε αρκετά υποσυστήματα. Έτσι θα κατασκευαστούν τα υποσυστήματα φοιτητή, εκπαιδευτικού, ενδιάμεσο (interface) ωρολογίου προγράμματος, συγχρονισμού, υποστήριξης διεργασιών και Android.

Υποσύστημα Φοιτητή

Κάθε φοιτητής θα χρησιμοποιεί αυτό το υποσύστημα ώστε να μπορεί να δηλώσει σε κάθε μάθημα την ομάδα παρακολούθησης από τις διαθέσιμες που υπάρχουν. Στο φοιτητή εμφανίζονται μαθήματα που έχει δηλώσει στο υποσύστημα της ηλεκτρονικής γραμματείας. Ο φοιτητής έχει τη δυνατότητα να ακυρώσει ομάδα παρακολούθησης-και να δηλώσει κάποια άλλη- αν το επιθυμεί μέσα στο διάστημα που επιτρέπονται οι δηλώσεις ομάδων μαθημάτων.

Κατά την είσοδό τους στο σύστημα η αυθεντικοποίηση γίνεται είτε με χρήση της υπηρεσίας LDAP που διαθέτει κάθε ίδρυμα, είτε από τη βάση δεδομένων του συστήματος.

Υποσύστημα Εκπαιδευτικού/Καθηγητή

Το υποσύστημα καθηγητή μπορεί να το χρησιμοποιεί ο διαχειριστής του συστήματος, και οι καθηγητές. Ο καθηγητής έχει τη δυνατότητα να βλέπει τα στοιχεία που σχετίζονται στο τμήμα που ανήκει, όπως καθηγητές, μαθήματα, ομάδες μαθημάτων και λεπτομέρειες που αφορούν

κάθε ομάδα παρακολούθησης μαθημάτων (όπως ποιοι φοιτητές έχουν δηλώσει κάποια ομάδα ή τις διαθέσιμες θέσεις της ομάδας). Αν ο καθηγητής είναι προϊστάμενος έχει επιπλέον τη δυνατότητα να τροποποιήσει τις ομάδες παρακολούθησης μαθημάτων του τμήματός του. Η δυνατότητα αυτή κρίνεται σκόπιμη, καθώς υπάρχει ενδεχόμενο να υπάρξει ανάγκη να προστεθούν νέες ομάδες αν οι υπόλοιπες έχουν γεμίσει ή υπάρχει ενδεχόμενο να πρέπει να αφαιρεθούν ομάδες αν έχουν ελάχιστα άτομα δηλωμένα.

Για υποβοήθηση στον καθορισμό ομάδων παρακολούθησης καθώς και στις αναθέσεις των εκπαιδευτικών ο προϊστάμενος θα μπορεί να βλέπει κάποια στατιστικά στοιχεία. Τα στοιχεία αυτά θα αφορούν ιστορικό περιόδων ανά μάθημα και περιλαμβάνουν φοιτητές που δήλωσαν το μάθημα, επιτυγχόντες ή μη επιτυγχόντες κλπ.

Κατά την είσοδό τους στο σύστημα η αυθεντικοποίηση γίνεται με χρήση της υπηρεσίας LDAP που διαθέτει κάθε ίδρυμα.

Υποσύστημα Συγχρονισμού

Απαιτήσεις που αφορούν την εκκίνηση του συστήματος

Το υποσύστημα συγχρονισμού συγχρονίζει τη βάση δεδομένων του συστήματος με τα στοιχεία που ενδιαφέρουν από τη βάση δεδομένων του γραμματειακού συστήματος. Πιο συγκεκριμένα, από το υποσύστημα της ηλεκτρονικής γραμματείας συλλέγονται οι ενεργοί φοιτητές, τα μαθήματα με βάση το ενεργό πρόγραμμα σπουδών, οι εκπαιδευτικοί των τμημάτων, οι τρέχουσες δηλώσεις των φοιτητών, καθώς και ιστορικό περιόδων ανά μάθημα για την προβολή των στατιστικών υποβοήθησης του υποσυστήματος των καθηγητών. Τα στοιχεία αυτά αφού διαβαστούν από τη βάση δεδομένων του γραμματειακού συστήματος μετατρέπονται σε XML και στη συνέχεια εισάγονται στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών, ώστε οι καθηγητές και οι φοιτητές να βλέπουν στα υποσυστήματά τους την εικόνα που αναλογεί.

Απαιτήσεις που αφορούν τις ομάδες παρακολούθησης μαθημάτων

Κατά την εισαγωγή, τροποποίηση ή διαγραφή μιας ομάδας από κάποιο μάθημα θα στέλνεται ειδοποίηση με e-mail στους φοιτητές που έχουν δηλώσει το μάθημα από το σύστημα ηλεκτρονικής γραμματείας.

Όταν γεμίσουν όλες οι ομάδες μαθημάτων θα αποστέλλεται ειδοποίηση στον υπεύθυνο καθηγητή μαθήματος καθώς και στον προϊστάμενο του τμήματος. Θεωρούμε πως οι ομάδες παρακολούθησης ενός μαθήματος έχουν γεμίσει όταν κάποιος φοιτητής μπορέσει και δηλώσει μια ειδική ομάδα παρακολούθησης που λέγεται ΠΛΕΟΝΑΖΟΝΤΕΣ.

Για την πιο δίκαιη κατανομή των φοιτητών θα χρησιμοποιηθεί το κριτήριο του εξαμήνου παρακολούθησης. Για την εφαρμογή αυτού του κριτηρίου θα οριστεί ένα διάστημα ημερομηνιών όπου οι φοιτητές θα μπορούν να δηλώνουν μόνο μαθήματα που συμπίπτουν με το εξάμηνό τους. Μετά από αυτή την ημερομηνία θα μπορούν να δηλώνουν ομάδα σε όλα τα μαθήματα της δήλωσής τους.

Αν η περίοδος δήλωσης ομάδων παρακολούθησης συμπίπτει με την περίοδο δηλώσεων μαθημάτων στο σύστημα ηλεκτρονικής γραμματείας τότε συντρέχουν λόγοι να γίνεται τακτικά συγχρονισμός των συστημάτων. Για παράδειγμα αν κάποιος φοιτητής έχει δηλώσει μάθημα στην ηλεκτρονική γραμματεία και έχει δηλώσει και ομάδα παρακολούθησης υπάρχει ενδεχόμενο αργότερα να αποφασίσει να αφαιρέσει από τη δήλωσή του το εν λόγω μάθημα, και σε αυτή την περίπτωση θα πρέπει αυτόματα να αφαιρείται ο φοιτητής από την ομάδα παρακολούθησης.

Αρκετά από τα μαθήματα έχουν ομάδες παρακολούθησης ενώ κάποια άλλα δεν έχουν και ο προϊστάμενος του τμήματος θα πρέπει να επιλέξει ποια θα είναι τα μαθήματα που έχουν ομάδες παρακολούθησης. Μόλις οριστικοποιηθούν οι ομάδες παρακολούθησης οι φοιτητές θα μπορούν να δηλώνουν ομάδες μαθημάτων. Εναλλακτικά μπορεί να θεωρηθεί πως τα μαθήματα που έχουν μόνο μια ομάδα παρακολούθησης δε θα χρειάζονται να δηλώσουν ομάδα.

Υποσύστημα ωρολογίου προγράμματος

Για την δημιουργία των ωρολογίων προγραμμάτων ανά τμήμα χρησιμοποιείται το λογισμικό FET. Ωστόσο το σύστημα αυτό τρέχει σε μορφή κονσόλας και δέχεται είσοδο XML αρχείο, ενώ το παραγόμενο ωρολόγιο πρόγραμμα είναι σε XML επίσης. Έτσι έχει κατασκευαστεί ένα νέο υποσύστημα το οποίο αναλαμβάνει την ευέλικτη κατασκευή του αρχείου εισόδου με τους επιθυμητούς περιορισμούς, καθώς και τη διαμόρφωση του παραγόμενου ωρολογίου προγράμματος από XML μορφή σε διεπαφή προβολής στους χρήστες.

Απαιτήσεις που αφορούν τη δημιουργία του ωρολογίου προγράμματος

Για την κατασκευή του ωρολογίου προγράμματος θα χρησιμοποιηθεί το λογισμικό FET. Το λογισμικό FET μεταξύ των άλλων επιτρέπει την εκτέλεση του σε μορφή κονσόλας (command line), ενώ η είσοδος και η έξοδος μπορούν να είναι αρχεία XML.

Η επιτυχής χρήση του λογισμικού απαιτεί την τροφοδότηση απαραίτητων στοιχείων εισόδου όπως είναι οι καθηγητές, οι αίθουσες, τα μαθήματα της περιόδου καθώς και κάποιοι περιορισμοί. Αρκετά από αυτά τα στοιχεία (όπως οι καθηγητές, τα μαθήματα κ.α.) υπάρχουν στο σύστημα της ηλεκτρονικής γραμματείας και ενδεχομένως μπορούν να τροφοδοτηθούν αυτόματα στο λογισμικό. Το αποτέλεσμα της εκτέλεσης του λογισμικού είναι ένα αρχείο εξόδου σε μορφή XML που περιλαμβάνει το ωρολόγιο πρόγραμμα.

Το XML αρχείο εξόδου μπορεί να χρησιμοποιηθεί για να προβληθεί το ωρολόγιο πρόγραμμα στο υποσύστημα φοιτητών και καθηγητών.

Σχεδίαση της διαχείρισης του ωρολογίου προγράμματος

Για να μπορέσει να χρησιμοποιηθεί το (εξωτερικό) λογισμικό FET, απαιτείται η τροφοδότησή (με αρχείο εισόδου xml) του με συγκεκριμένα στοιχεία, όπως μαθήματα, καθηγητές, αίθουσες και χωρητικότητα, ώρες διάλεξης μαθήματος, και ο αριθμός των ομάδων για κάθε μάθημα. Κάποια από αυτά τα στοιχεία υπάρχουν στη βάση δεδομένων του γραμματειακού συστήματος (μαθήματα, καθηγητές), ενώ κάποια άλλα δεν υπάρχουν (αίθουσες, ώρες διάλεξης μαθήματος, αριθμός των ομάδων για κάθε μάθημα). Τα στοιχεία εισόδου που δεν υπάρχουν θα πρέπει να συμπληρωθούν από τον προϊστάμενο από το υποσύστημα καθηγητή.

Αφού συμπληρωθούν όλες οι απαιτούμενες πληροφορίες θα κατασκευαστεί το αρχείο εισόδου xml και θα γίνει κλήση του εργαλείου FET. Το FET θα επιστρέψει το ωρολόγιο πρόγραμμα (αρχείο εξόδου σε μορφή xml).

Στη συνέχεια θα γίνει ανάλυση του αρχείου εξόδου xml και α) θα δημιουργηθούν στο σύστημα οι ομάδες των μαθημάτων, β) θα αναπαρασταθεί σε μορφή πίνακα το ωρολόγιο πρόγραμμα.

Αφού γίνουν αυτά ο προϊστάμενος και οι καθηγητές μπορούν να βλέπουν το ωρολόγιο πρόγραμμα και να προτείνουν (μικρο) αλλαγές. Αν ο προϊστάμενος αποφασίσει να κάνει

αλλαγές, δεν θα τις κάνει με χρήση του προγράμματος FET αλλά θα επέμβει απευθείας στην επεξεργασία μαθημάτων.

Μετά την οποιαδήποτε αλλαγή θα πρέπει να ενημερωθεί αντίστοιχα και το ωρολόγιο πρόγραμμα.

Μόλις οριστικοποιηθεί το ωρολόγιο πρόγραμμα ο προϊστάμενος θα επιλέξει δημοσίευση του ωρολογίου προγράμματος στους καθηγητές και στους φοιτητές του τμήματος με αποστολή σχετικού e-mail.

Ενδεχόμενες μεταβολές στο ωρολόγιο πρόγραμμα μετά τη δημοσίευση του είναι επιτρεπτές και θα αναδημοσιεύεται με αποστολή σχετικού e-mail.

Υποσύστημα υποστήριξης διεργασιών(BPM processes)

Το υποσύστημα αυτό ασχολείται με την υλοποίηση των διεργασιών και την παροχή τους ως υπηρεσία στα υπόλοιπα υποσυστήματα. Κάθε διεργασία για να εκτελεστεί απαιτεί ένα σύνολο κλάσεων με τον απαιτούμενο κώδικα για την εκτέλεση των εργασιών (domain specific tasks). Οι κλάσεις κάθε διεργασίας αποθηκεύονται σε ξεχωριστό πακέτο (java package).

Για να μπορέσει το υποσύστημα αυτό να προσφέρει την εκτέλεση των διεργασιών στα υπόλοιπα υποσυστήματα, λειτουργεί ως υπηρεσία REST. Για κάθε διεργασία έχουν κατασκευαστεί οι κατάλληλες μέθοδοι (rest methods) τις οποίες τα υποσυστήματα μπορούν να καλούν, ως rest clients.

Οι διεργασίες που υποστηρίζονται από το σύστημα περιγράφονται στην ενότητα 2.5

Υποσύστημα Android

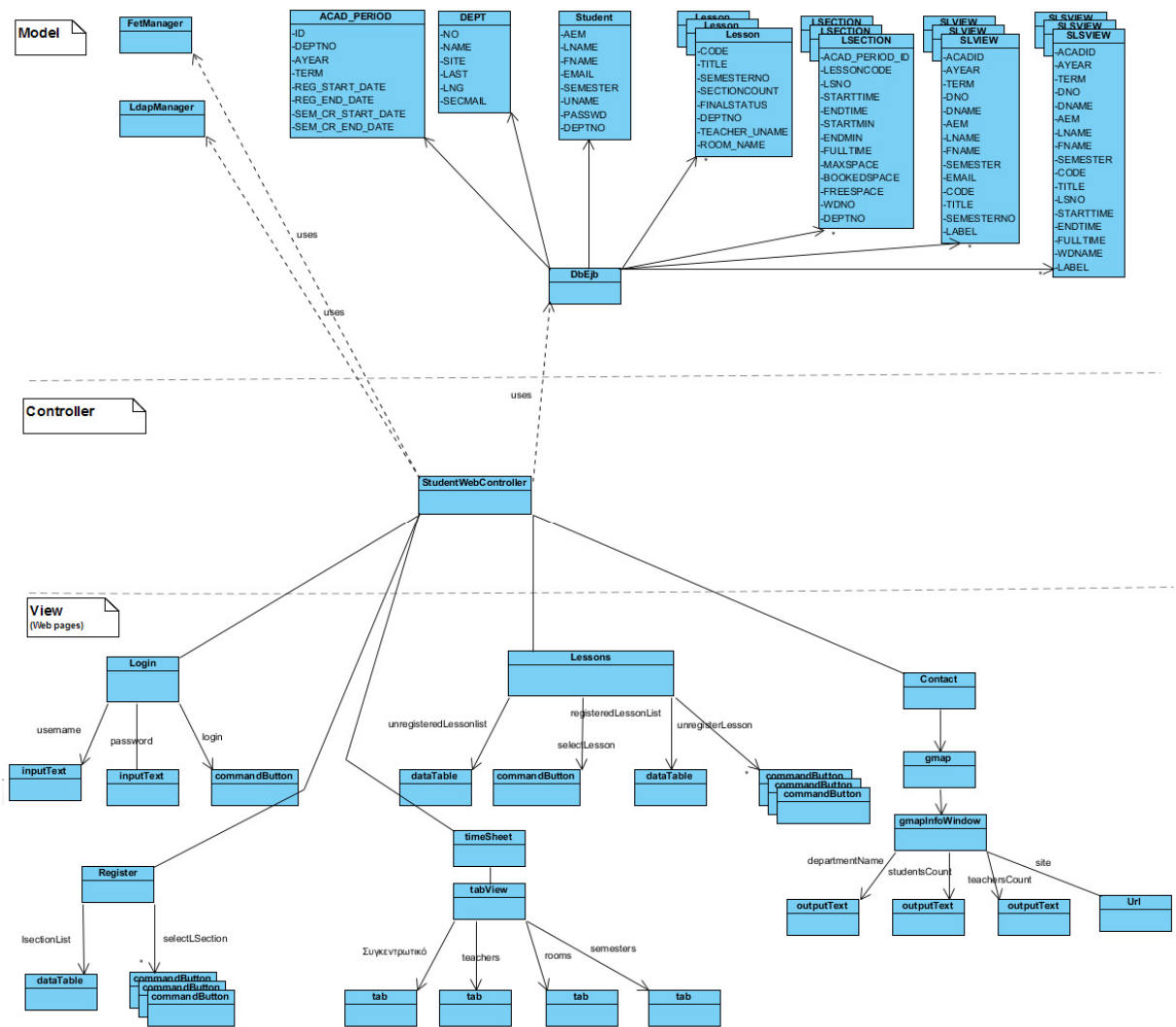
Το υποσύστημα Android είναι μια μικρή εφαρμογή που θα χρησιμοποιούν οι φοιτητές ώστε να μπορούν να κάνουν τις δηλώσεις ομάδων παρακολούθησης των μαθημάτων τους από συσκευές Android.

2.2 Διαγράμματα κλάσεων

Στις επόμενες υποενότητες παρουσιάζονται τα διαγράμματα κλάσεων των υποσυστημάτων. Για κάθε υποσύστημα παρουσιάζουμε ένα (συμπυκνωμένο) διάγραμμα κλάσεων, ενώ στη συνέχεια περιγράφονται αναλυτικά οι λεπτομέρειες κάθε κλάσης. Τα διαγράμματα κλάσεων του υποσυστήματος φοιτητών και του υποσυστήματος εκπαιδευτικών έχουν σχεδιαστεί με βάση την αρχιτεκτονική Model View Controller (MVC), καθώς σε αυτή βασίζονται τα εργαλεία υλοποίησης. Να σημειώσουμε επίσης πως τα διαγράμματα κλάσεων εμφανίζονται σε συμπυγμένη μορφή στις επόμενες σελίδες λόγω χώρου, ενώ οι πλήρεις εκδόσεις των διαγραμμάτων κλάσεων υπάρχουν στο συνοδευτικό υλικό της μεταπτυχιακής διατριβής.

2.2.1 Διάγραμμα κλάσεων υποσυστήματος φοιτητών

Το υποσύστημα φοιτητών βασίζεται στην αρχιτεκτονική MVC. Στο διάγραμμα κλάσεων που ακολουθεί (Σχήμα 2.1) στο τμήμα Model παρουσιάζονται οι κλάσεις που αντιστοιχούν στους πίνακες της βάσης, στην επικοινωνία με τον LDAP και στο εσωτερικό υποσύστημα ωρολογίου προγράμματος. Στο τμήμα View παρουσιάζονται οι ιστοσελίδες της εφαρμογής με τα επιμέρους αντικείμενα προβολής στοιχείων στο χρήστη-φοιτητή, ενώ στο τμήμα Controller εμφανίζεται η κλάση StudentWebController η οποία έχει το ρόλο της γεφύρωσης των δυο άλλων τμημάτων.

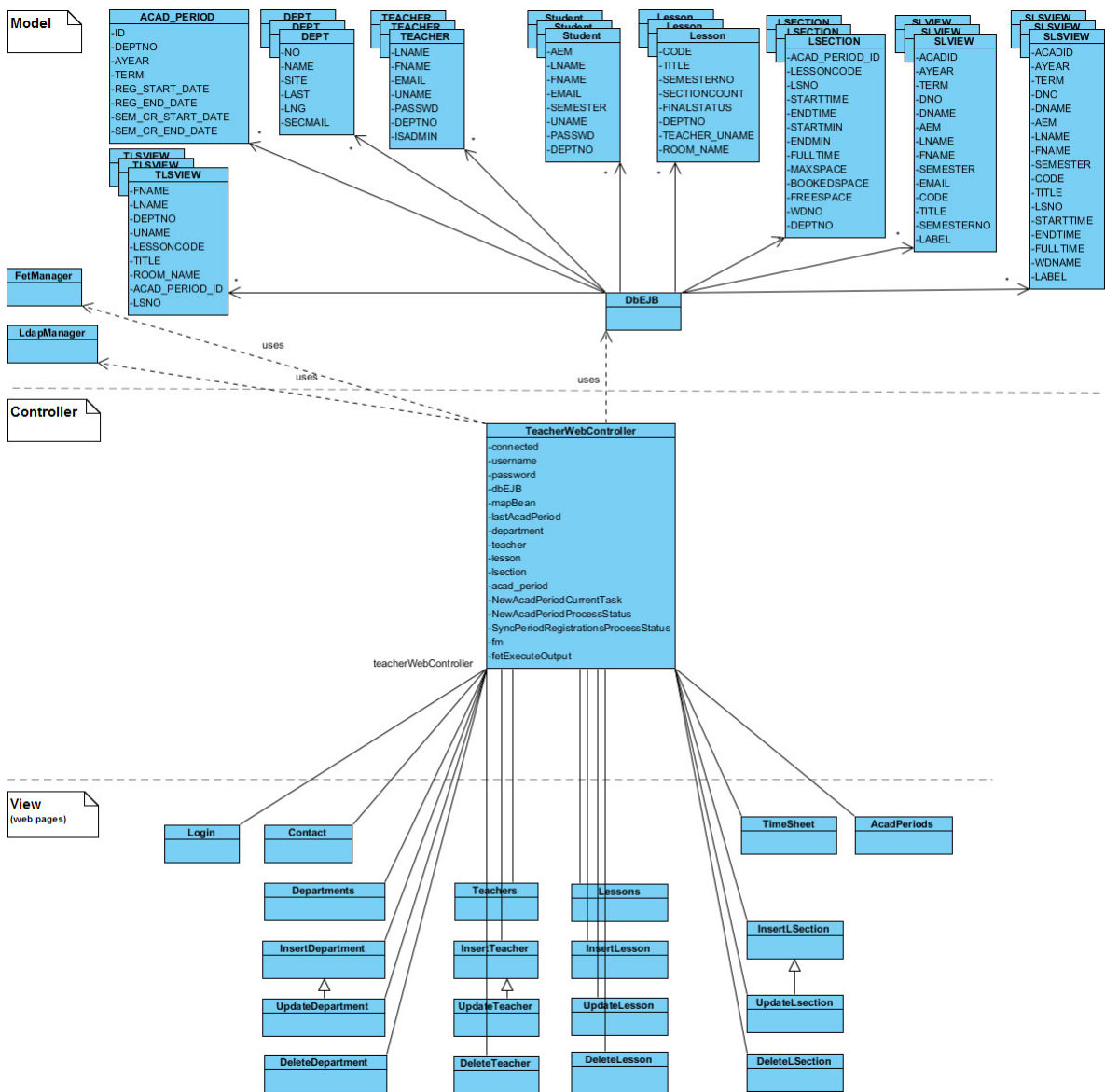


Σχήμα 2.1: Διάγραμμα κλάσεων υποσυστήματος φοιτητών

Η λεπτομερής περιγραφή των κλάσεων του υποσυστήματος φοιτητών παρουσιάζεται στην ενότητα Α.1 του παραρτήματος.

2.2.2 Διάγραμμα κλάσεων υποσυστήματος καθηγητών

Το υποσύστημα καθηγητών βασίζεται στην αρχιτεκτονική MVC. Στο διάγραμμα κλάσεων που ακολουθεί (Σχήμα 2.2) στο τμήμα Model παρουσιάζονται οι κλάσεις που αντιστοιχούν στους πίνακες της βάσης, στην επικοινωνία με τον LDAP και στο εσωτερικό υποσύστημα ωρολογίου προγράμματος. Στο τμήμα View παρουσιάζονται οι ιστοσελίδες της εφαρμογής με τα επιμέρους αντικείμενα προβολής στοιχείων στους χρήστες (εκπαιδευτικοί, προϊστάμενοι και διαχειριστής), ενώ στο τμήμα Controller εμφανίζεται η κλάση TeacherWebController η οποία έχει το ρόλο της γεφύρωσης των δυο άλλων τμημάτων.



Σχήμα 2.2: Διάγραμμα κλάσεων υποσυστήματος εκπαιδευτικών

Η λεπτομερής περιγραφή των κλάσεων του υποσυστήματος εκπαιδευτικών παρουσιάζεται στην ενότητα A.2 του παραρτήματος.

2.2.3 Διάγραμμα κλάσεων ωρολογίου προγράμματος

Το υποσύστημα ωρολογίου προγράμματος αποτελεί ένα ενδιάμεσο διεπαφής (interface) του εξωτερικού υποσυστήματος FET και των υποσυστημάτων εκπαιδευτικού και φοιτητών που το χρησιμοποιούν. Το εξωτερικό υποσύστημα FET δέχεται ως είσοδο αρχείο XML και παράγει ως έξοδο αρχείο XML. Το υποσύστημα ωρολογίου προγράμματος αναλαμβάνει την κατασκευή του αρχείου εισόδου με τους επιθυμητούς περιορισμούς, καθώς και τη διαμόρφωση του παραγόμενου ωρολογίου προγράμματος από XML μορφή σε διεπαφή προβολής στους χρήστες.

Για να γίνει εύκολα η επεξεργασία των XML αρχείων χρησιμοποιείται η αρχιτεκτονική JAXB (Java Architecture for XML Binding) [13],[26]. Για να ενσωματωθεί η εν λόγω αρχιτεκτονική απαιτείται προσεκτική εξέταση των XML tags του XML αρχείου που χρησιμοποιεί το FET και η κατασκευή αντίστοιχων κλάσεων για κάθε XML tag. Όταν κατασκευαστεί μια τέτοια αντιστοιχία τότε είναι πολύ εύκολη η μετατροπή ενός XML αρχείου σε αντικείμενα κλάσεων της εφαρμογής ή η μετατροπή των αντικειμένων κλάσεων της εφαρμογής σε παραγόμενο αρχείο XML. Στη λίστα κώδικα που ακολουθεί εμφανίζεται ενδεικτική δομή του FET XML αρχείου. Το διάγραμμα κλάσεων του ωρολογίου προγράμματος που ακολουθεί (Σχήμα 2.3) έχει κατασκευαστεί σε αυτή τη λογική. Αποτελεί κυρίως μια διασύνδεση των XML tags του εξωτερικού υποσυστήματος FET με αντίστοιχες κλάσεις του υποσυστήματος ωρολογίου προγράμματος, ενώ διαθέτει κάποιες επιπλέον κλάσεις με μεθόδους διεπαφής για την ανταλλαγή δεδομένων από τα υποσυστήματα φοιτητή και εκπαιδευτικού.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<fet>
  <Institution_Name>ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.</Institution_Name>
  <Comments>II Comments</Comments>
  <Days_List>
    <Number_of_Days>5</Number_of_Days>
    <Day>
      <Name>ΔΕΥΤΕΡΑ</Name>
    </Day>
    <Day>
      <Name>ΤΡΙΤΗ</Name>
    </Day>
    <Day>
      <Name>ΤΕΤΑΡΤΗ</Name>
    </Day>
    <Day>
      <Name>ΠΕΜΠΤΗ</Name>
    </Day>
    <Day>
      <Name>ΠΑΡΑΣΚΕΥΗ</Name>
    </Day>
  </Days_List>
  <Hours_List>
    <Number_of_Hours>14</Number_of_Hours>
    <Hour>
      <Name>08:00</Name>
    </Hour>
    <Hour>
      <Name>09:00</Name>
    </Hour>
    <Hour>
      <Name>10:00</Name>
    </Hour>
    ...
    <Hour>
      <Name>20:00</Name>
    </Hour>
  </Hours_List>
</fet>
```

```

    <Name>21:00</Name>
  </Hour>
</Hours_List>
<Students_List>
  <Year>
    <Name>20142</Name>
    <Number_of_Students>0</Number_of_Students>
    <Group>
      <Name>ΔΟΚ-Θ</Name>
      <Number_of_Students>0</Number_of_Students>
      <Subgroup>
        <Name>ΠΛΕΟΝΑΖΟΝΤΕΣ</Name>
        <Number_of_Students>0</Number_of_Students>
      </Subgroup>
      <Subgroup>
        <Name>ΤΕΛΙΚΗ ΕΞΕΤΑΣΗ</Name>
        <Number_of_Students>0</Number_of_Students>
      </Subgroup>
    </Group>
    <Group>
      <Name>ΔΟΚ-E</Name>
      <Number_of_Students>0</Number_of_Students>
      <Subgroup>
        <Name>ΠΛΕΟΝΑΖΟΝΤΕΣ</Name>
        <Number_of_Students>0</Number_of_Students>
      </Subgroup>
      <Subgroup>
        <Name>ΤΕΛΙΚΗ ΕΞΕΤΑΣΗ</Name>
        <Number_of_Students>0</Number_of_Students>
      </Subgroup>
    </Group>
  </Year>
</Students_List>
<Teachers_List>
  <Teacher>
    <Name>ΠΡΟΙΣΤΑΜΕΝΟΣ 1</Name>
  </Teacher>
  <Teacher>
    <Name>ΚΑΘΗΓΗΤΗΣ Α</Name>
  </Teacher>
  <Teacher>
    <Name>ΚΑΘΗΓΗΤΗΣ Β</Name>
  </Teacher>
</Teachers_List>
<Subjects_List>
  <Subject>
    <Name>ΔΟΚ-Θ</Name>
  </Subject>
  <Subject>
    <Name>ΔΟΚ-E</Name>
  </Subject>
</Subjects_List>
<Rooms_List>
  <Room>
    <Name>A1</Name>
    <Capacity>50</Capacity>
  </Room>
  <Room>
    <Name>A2</Name>
    <Capacity>30</Capacity>
  </Room>
</Rooms_List>

```

```

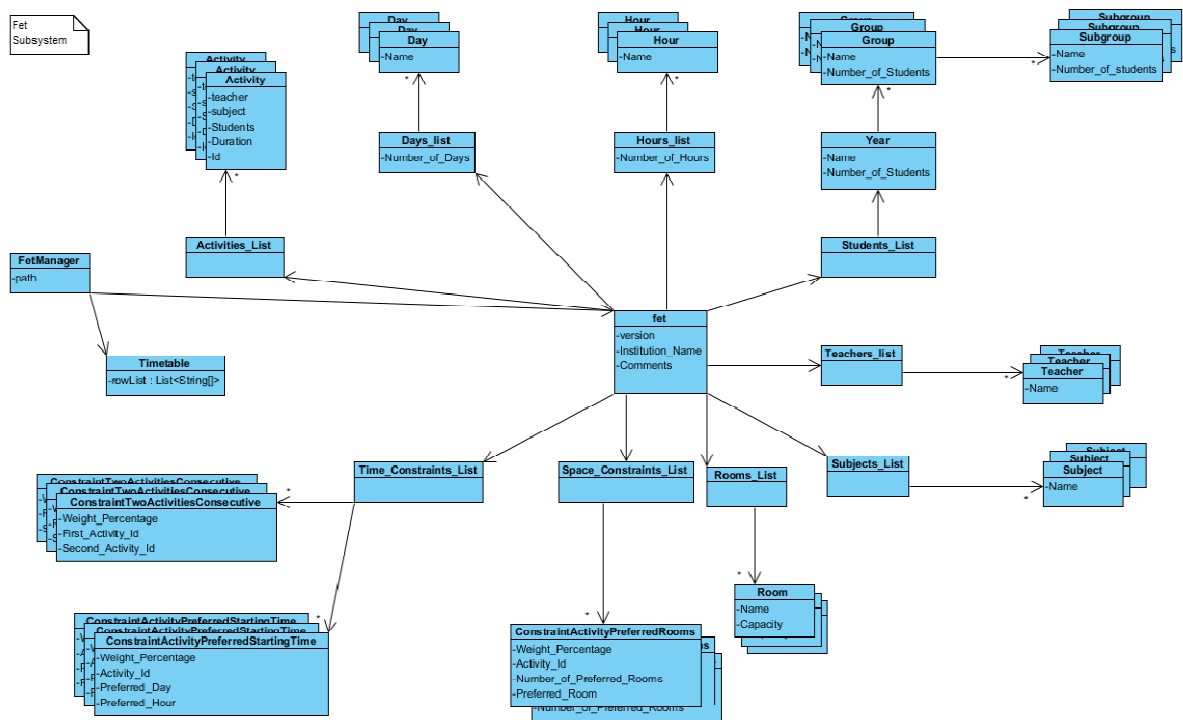
<Time_Constraints_List>
  <ConstraintBasicCompulsoryTime>
    <Weight_Percentage>100</Weight_Percentage>
    <Active>true</Active>
    <Comments></Comments>
  </ConstraintBasicCompulsoryTime>
  <ConstraintTwoActivitiesConsecutive>
    <Weight_Percentage>100</Weight_Percentage>
    <First_Activity_Id>1</First_Activity_Id>
    <Second_Activity_Id>2</Second_Activity_Id>
    <Active>true</Active>
  </ConstraintTwoActivitiesConsecutive>
  <ConstraintTwoActivitiesConsecutive>
    <Weight_Percentage>100</Weight_Percentage>
    <First_Activity_Id>3</First_Activity_Id>
    <Second_Activity_Id>4</Second_Activity_Id>
    <Active>true</Active>
  </ConstraintTwoActivitiesConsecutive>
</Time_Constraints_List>
<Activities_List>
  <Activity>
    <Teacher>ΚΑΘΗΓΗΤΗΣ Α</Teacher>
    <Subject>ΔΟΚ-Θ</Subject>
    <Students>ΠΛΕΟΝΑΖΟΝΤΕΣ</Students>
    <Duration>2</Duration>
    <Total_Duration>2</Total_Duration>
    <Id>1</Id>
    <Activity_Group_Id>0</Activity_Group_Id>
    <Active>true</Active>
    <Comments></Comments>
  </Activity>
  <Activity>
    <Teacher>ΚΑΘΗΓΗΤΗΣ Α</Teacher>
    <Subject>ΔΟΚ-Θ</Subject>
    <Students>ΤΕΛΙΚΗ ΕΞΕΤΑΣΗ</Students>
    <Duration>2</Duration>
    <Total_Duration>2</Total_Duration>
    <Id>2</Id>
    <Activity_Group_Id>0</Activity_Group_Id>
    <Active>true</Active>
    <Comments></Comments>
  </Activity>
  <Activity>
    <Teacher>ΚΑΘΗΓΗΤΗΣ Β</Teacher>
    <Subject>ΔΟΚ-Ε</Subject>
    <Students>ΠΛΕΟΝΑΖΟΝΤΕΣ</Students>
    <Duration>2</Duration>
    <Total_Duration>2</Total_Duration>
    <Id>3</Id>
    <Activity_Group_Id>0</Activity_Group_Id>
    <Active>true</Active>
    <Comments></Comments>
  </Activity>
  <Activity>
    <Teacher>ΚΑΘΗΓΗΤΗΣ Β</Teacher>
    <Subject>ΔΟΚ-Ε</Subject>
    <Students>ΤΕΛΙΚΗ ΕΞΕΤΑΣΗ</Students>
    <Duration>2</Duration>
    <Total_Duration>2</Total_Duration>
    <Id>4</Id>
    <Activity_Group_Id>0</Activity_Group_Id>
    <Active>true</Active>

```

```

    <Comments></Comments>
  </Activity>
</Activities_List>
<Space_Constraints_List>
  <ConstraintBasicCompulsorySpace>
    <Weight_Percentage>100</Weight_Percentage>
    <Active>true</Active>
    <Comments></Comments>
  </ConstraintBasicCompulsorySpace>
  <ConstraintActivityPreferredRooms>
    <Weight_Percentage>100</Weight_Percentage>
    <Activity_Id>1</Activity_Id>
    <Number_of_PREFERRED_Rooms>1</Number_of_PREFERRED_Rooms>
    <Preferred_Room>A1</Preferred_Room>
    <Active>true</Active>
    <Comments></Comments>
  </ConstraintActivityPreferredRooms>
  <ConstraintActivityPreferredRooms>
    <Weight_Percentage>100</Weight_Percentage>
    <Activity_Id>2</Activity_Id>
    <Number_of_PREFERRED_Rooms>1</Number_of_PREFERRED_Rooms>
    <Preferred_Room>A1</Preferred_Room>
    <Active>true</Active>
    <Comments></Comments>
  </ConstraintActivityPreferredRooms>
  <ConstraintActivityPreferredRooms>
    <Weight_Percentage>100</Weight_Percentage>
    <Activity_Id>3</Activity_Id>
    <Number_of_PREFERRED_Rooms>1</Number_of_PREFERRED_Rooms>
    <Preferred_Room>A2</Preferred_Room>
    <Active>true</Active>
    <Comments></Comments>
  </ConstraintActivityPreferredRooms>
  <ConstraintActivityPreferredRooms>
    <Weight_Percentage>100</Weight_Percentage>
    <Activity_Id>4</Activity_Id>
    <Number_of_PREFERRED_Rooms>1</Number_of_PREFERRED_Rooms>
    <Preferred_Room>A2</Preferred_Room>
    <Active>true</Active>
    <Comments></Comments>
  </ConstraintActivityPreferredRooms>
</Space_Constraints_List>
</fet>

```

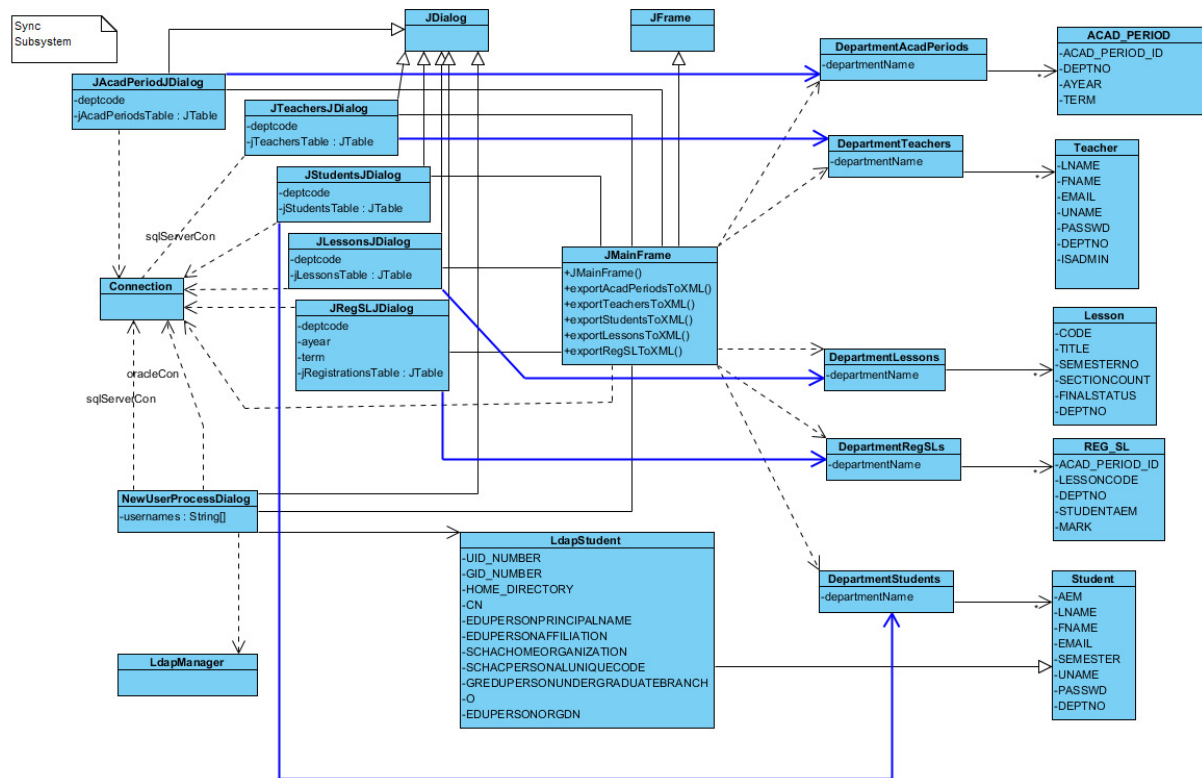


Σχήμα 2.3: Διάγραμμα κλάσεων υποσυστήματος ωρολογίου προγράμματος

Η λεπτομερής περιγραφή των κλάσεων του υποσυστήματος εκπαιδευτικών παρουσιάζεται στην ενότητα Α.3 του παραρτήματος.

2.2.4 Διάγραμμα κλάσεων υποσυστήματος συγχρονισμού

Το υποσύστημα συγχρονισμού ακολουθεί την κλασική αρχιτεκτονική client-server (Desktop Application + Database). Το υποσύστημα εκκινείται με αντικείμενο της κύριας κλάσης-φόρμας JMainFrame από την οποία υπάρχουν επιλογές για την εκκίνηση των φορμών ακαδημαϊκών περιόδων (JAcadPeriodJDialog), εκπαιδευτικών (JTeachersJDialog), φοιτητών (JStudentsJDialog), μαθημάτων (JLessonsJDialog), δηλώσεων μαθημάτων (JRegSLJDialog) και λογαριασμού νέου χρήστη (NewUserProcessDialog). Στην κύρια φόρμα προβάλλονται πινακάκια με τα στιγμιότυπα και την κατάσταση των τεσσάρων BPM διεργασιών ενώ η ολοκλήρωση των εργασιών (tasks) που αφορούν το διαχειριστή γίνεται στις παραπάνω φόρμες. Το διάγραμμα κλάσεων του υποσυστήματος συγχρονισμού παρουσιάζεται στο σχήμα 2.4.

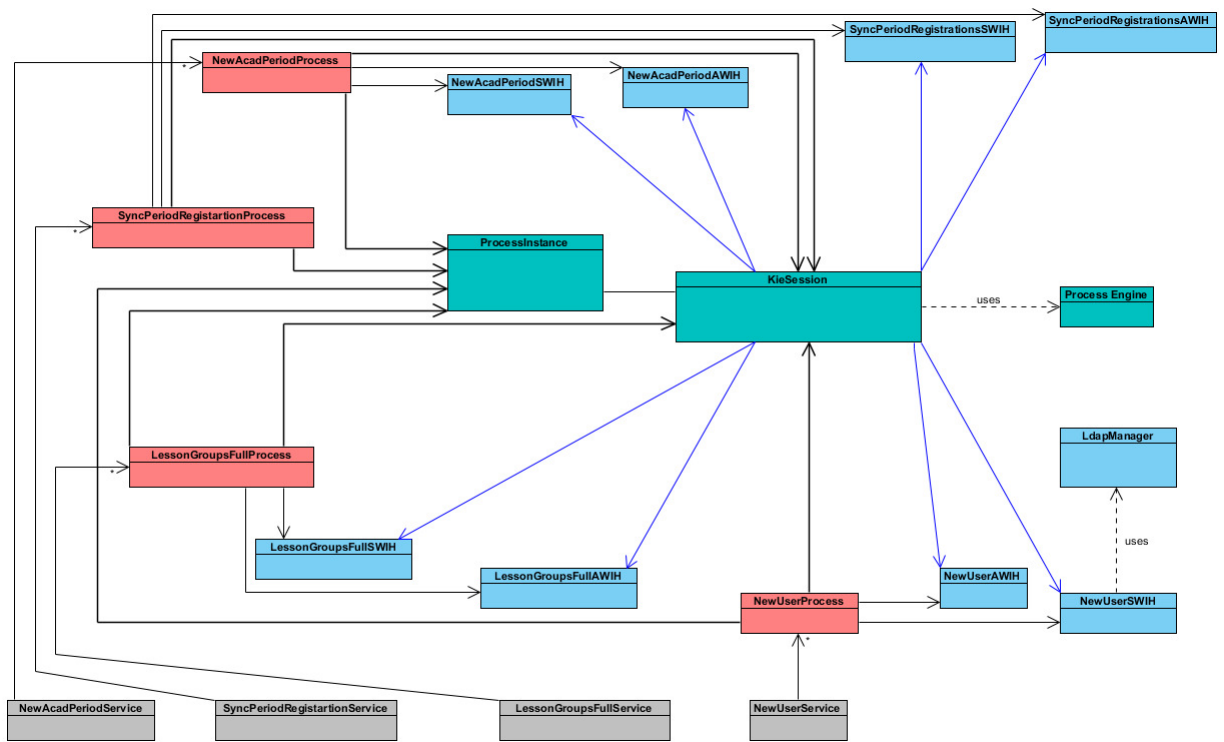


Σχήμα 2.4: Διάγραμμα κλάσεων υποσυστήματος συγχρονισμού

Η λεπτομερής περιγραφή των κλάσεων του υποσυστήματος συγχρονισμού παρουσιάζεται στην ενότητα Α.4 του παραρτήματος.

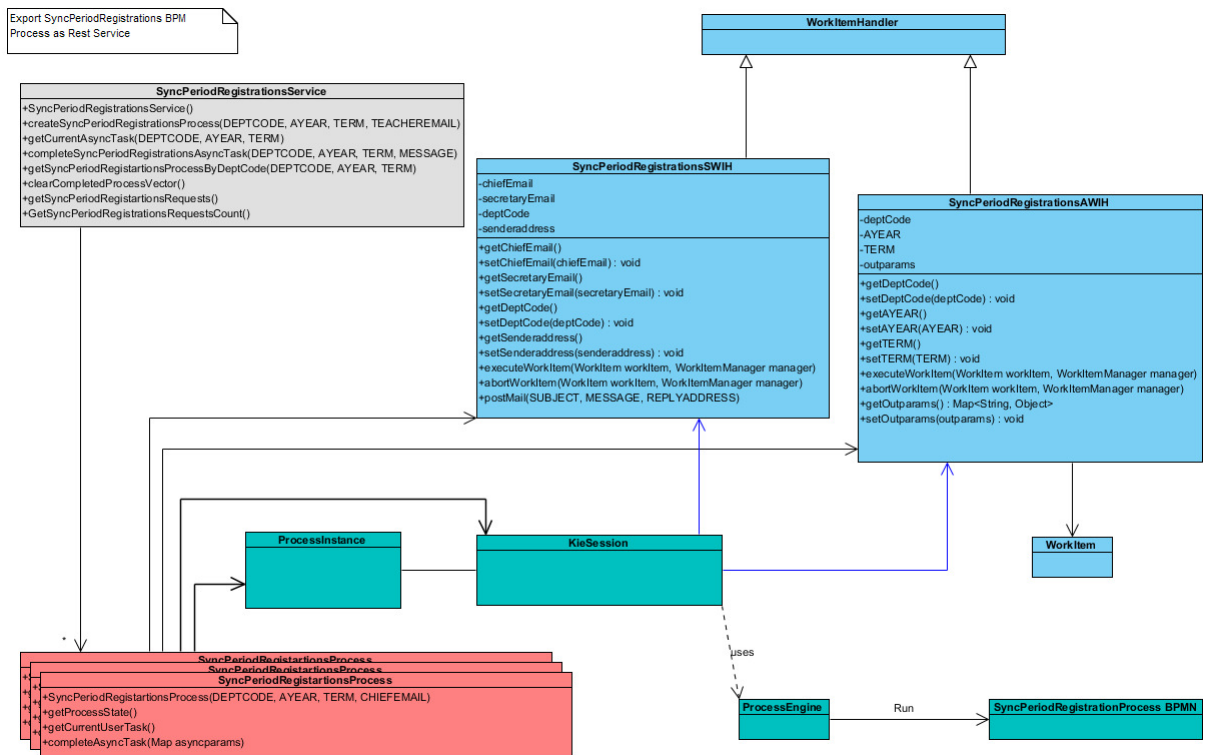
2.2.5 Διάγραμμα κλάσεων υποσυστήματος διεργασιών

Το ακόλουθο διάγραμμα κλάσεων (Σχήμα 2.5) περιλαμβάνει τις τέσσερις BPM διεργασίες που υποστηρίζει το σύστημά μας- διεργασία συγχρονισμού δηλώσεων ακαδημαϊκής περιόδου (*SyncPeriodRegistrations BPM Process*), διεργασία προετοιμασίας νέας ακαδημαϊκής περιόδου (*NewAcadPeriod BPM Process*), διεργασία κορεσμού ομάδων παρακολούθησης μαθήματος περιόδου (*LessonGroupsFull BPM Process*) και διεργασία συγχρονισμού λογαριασμού νέου χρήστη (*NewUser BPM Process*)- με τις επιμέρους κλάσεις τους. Περιλαμβάνει επίσης τις κλάσεις/υπηρεσίες Rest – **SyncPeriodRegistartion Rest Service**, **NewAcadPeriod Rest Service**, **LessonGroupsFull Rest Service** και **NewUser Rest Service** - οι οποίες προσφέρουν τις BPM διεργασίες στα υποσυστήματα που τις χρησιμοποιούν. Για λόγους απλοποίησης στα επόμενα σχήματα έχουν σχεδιαστεί χωριστά UML διαγράμματα για κάθε διεργασία.



Σχήμα 2.5: Διάγραμμα κλάσεων υποσυστήματος διεργασιών

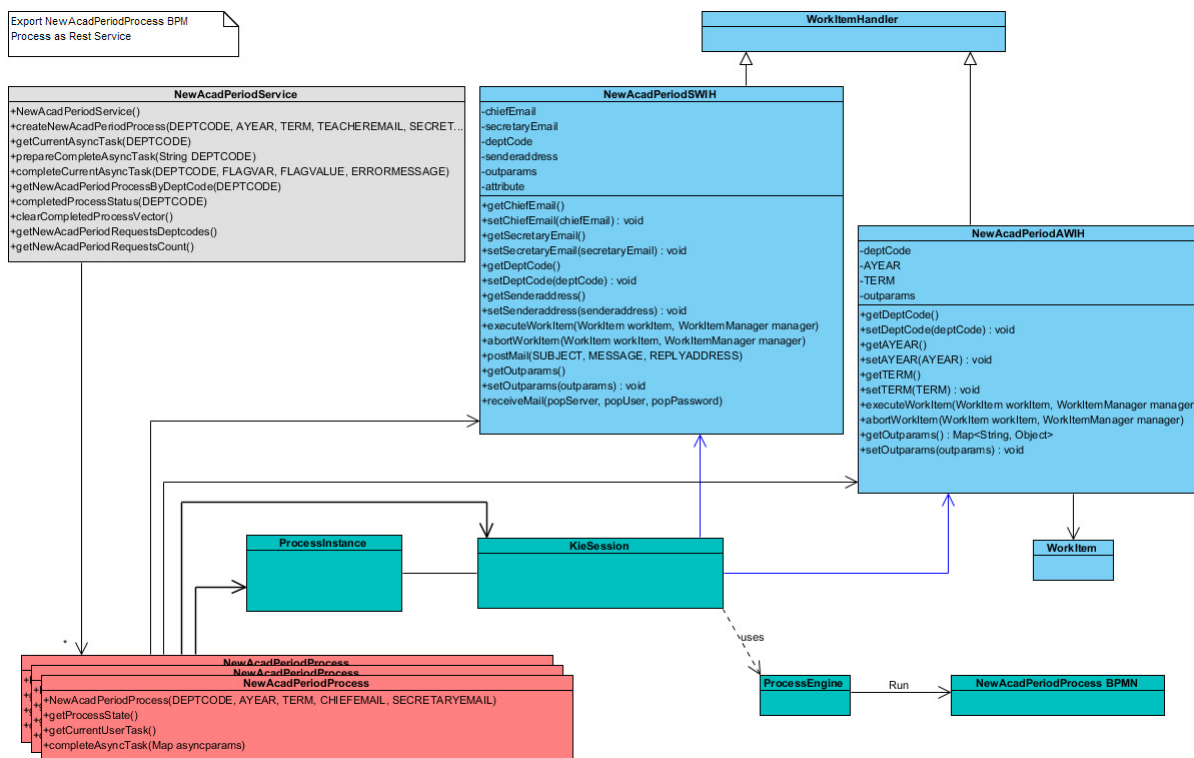
Διάγραμμα κλάσεων (Σχήμα 2.6) παροχής της διεργασίας συγχρονισμού δηλώσεων ακαδημαϊκής περιόδου (*SyncPeriodRegistrations BPM Process*) ως υπηρεσία Rest.



Σχήμα 2.6: Διάγραμμα κλάσεων παροχής της διεργασίας SyncPeriodRegistrations

Η λεπτομερής περιγραφή των κλάσεων παροχής της διεργασίας συγχρονισμού δηλώσεων ακαδημαϊκής περιόδου παρουσιάζεται στην παράγραφο A.5.1 του παραρτήματος.

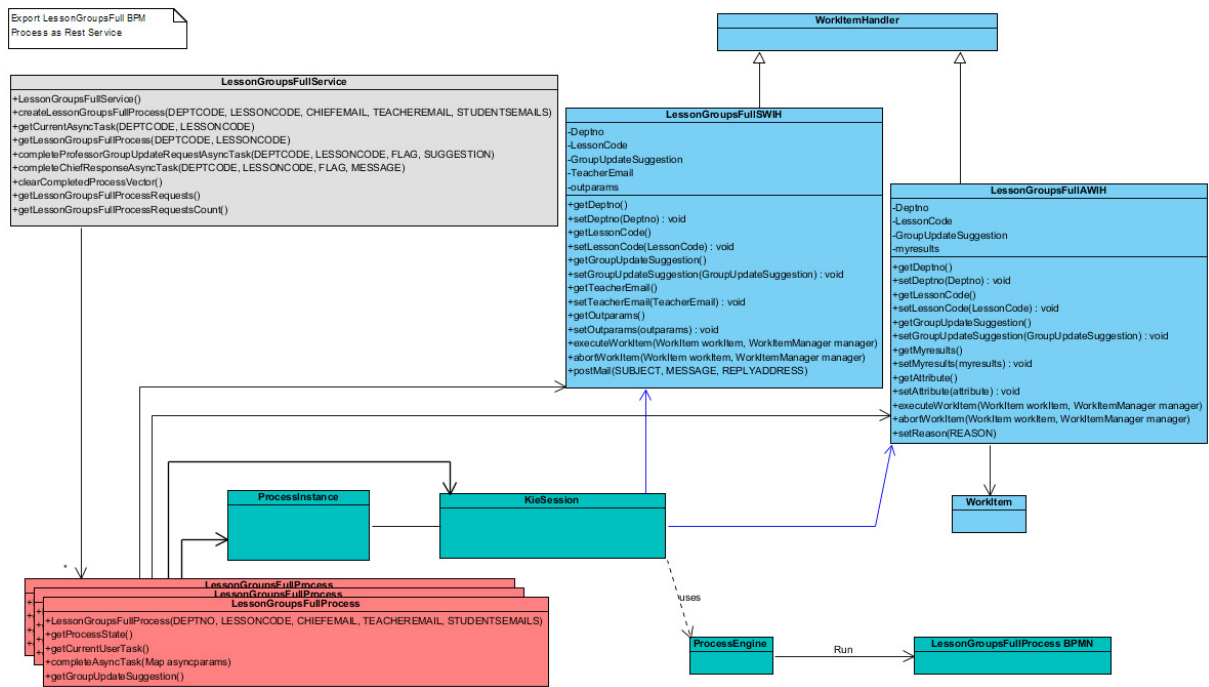
Διάγραμμα κλάσεων (Σχήμα 2.7) παροχής της διεργασίας προετοιμασίας νέας ακαδημαϊκής περιόδου (*NewAcadPeriod BPM Process*) ως υπηρεσία Rest.



Σχήμα 2.7: Διάγραμμα κλάσεων παροχής της διεργασίας NewAcadPeriod

Η λεπτομερής περιγραφή των κλάσεων παροχής της διεργασίας προετοιμασίας νέας ακαδημαϊκής περιόδου παρουσιάζεται στην παράγραφο A.5.2 του παραρτήματος.

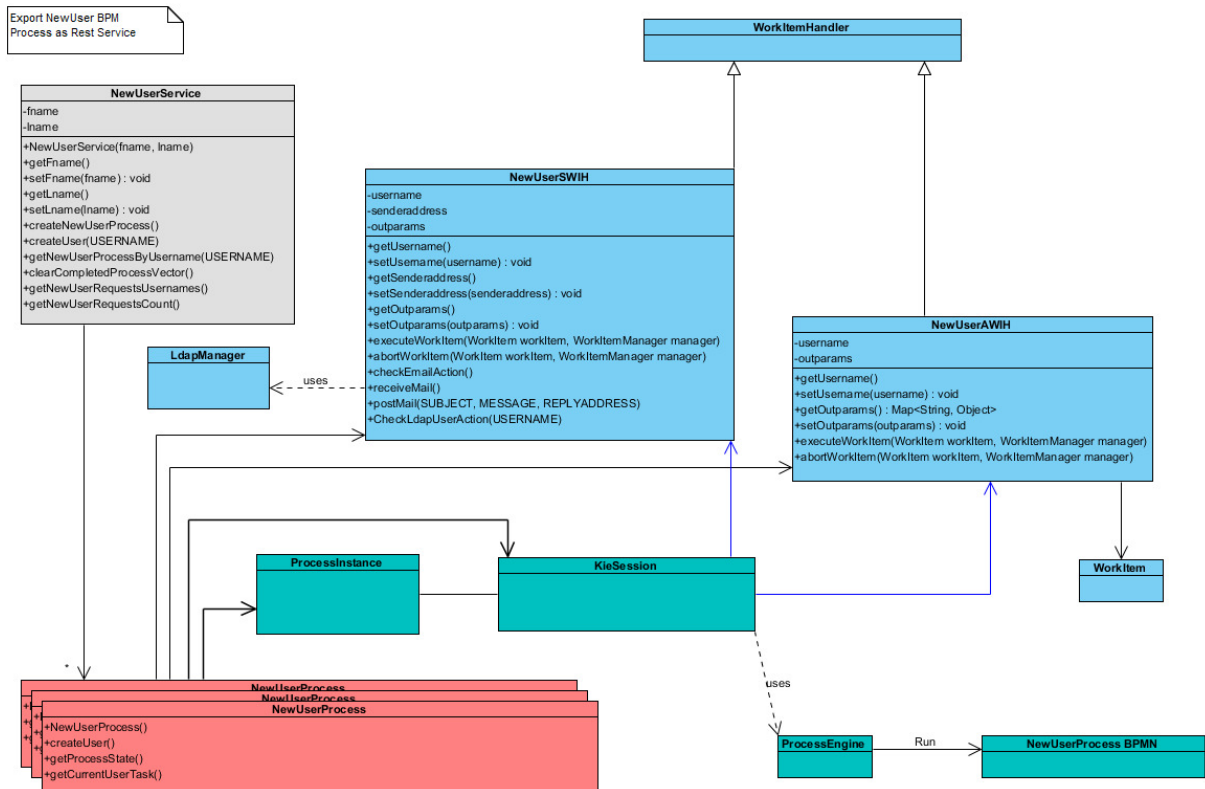
Διάγραμμα κλάσεων (Σχήμα 2.8) παροχής της διεργασίας κορεσμού ομάδων παρακολούθησης μαθήματος περιόδου (*LessonGroupsFull BPM Process*) ως υπηρεσία Rest.



Σχήμα 2.8: Διάγραμμα κλάσεων παροχής της διεργασίας LessonGroupsFull

Η λεπτομερής περιγραφή των κλάσεων παροχής της διεργασίας κορεσμού ομάδων παρακολούθησης παρουσιάζεται στην παράγραφο Α.5.3 του παραρτήματος.

Διάγραμμα κλάσεων (Σχήμα 2.9) παροχής της διεργασίας συγχρονισμού λογαριασμού νέου χρήστη (*NewUser BPM Process*) ως υπηρεσία Rest.



Σχήμα 2.9: Διάγραμμα κλάσεων παροχής της διεργασίας NewUser

Η λεπτομερής περιγραφή των κλάσεων παροχής της διεργασίας συγχρονισμού λογαριασμού νέου χρήστη παρουσιάζεται στην παράγραφο Α.5.4 του παραρτήματος 2.2.6

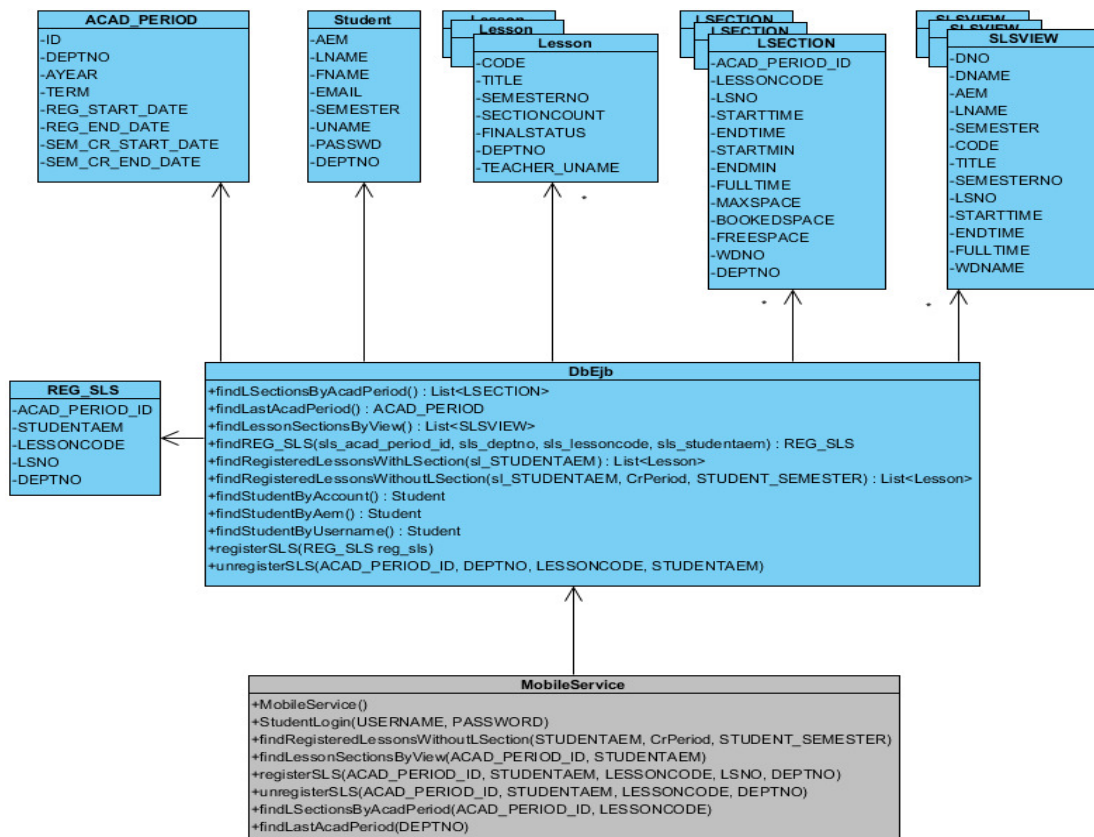
Διάγραμμα κλάσεων υποσυστήματος Android

Το υποσύστημα Android αποτελείται από δυο επιμέρους τμήματα: την Android εφαρμογή που θα έχουν εγκατεστημένη οι φοιτητές στις συσκευές Android και το τμήμα Rest Service back-end που θα λειτουργεί ως διαμεσολαβητής μεταξύ της εφαρμογής Android και της βάσης δεδομένων του συστήματος. Να σημειώσουμε πως δεν θεωρείται αποδοτική, ούτε και ασφαλής η απευθείας σύνδεση της εφαρμογής Android με τη βάση δεδομένων και για αυτό το λόγο απαιτείται το τμήμα Android Rest back-end.

Διάγραμμα κλάσεων (Σχήμα 2.10) υποσυστήματος Android: **Rest Service back-end**

Η κλάση **MobileService** δέχεται απομακρυσμένες κλήσεις REST από το τμήμα Android της εφαρμογής φοιτητών τις οποίες διεκπεραιώνουν οι μέθοδοί της (rest methods). Οι απομακρυσμένες αυτές κλήσεις αφορούν αιτήματα προς της βάση δεδομένων του συστήματος

υποστήριξης προγραμμάτων σπουδών. Τα αιτήματα αυτά προωθούνται προς τη βάση δεδομένων με χρήση της κλάσης **DbEjb** η οποία χρησιμοποιεί κλάσεις που είναι συνδεδεμένες με τους πίνακες της βάσης.



Σχήμα 2.10 Διάγραμμα κλάσεων Android Rest Service back-end

Η κλάση **DbEjb** χρησιμοποιείται για τη διασύνδεση της υπηρεσίας REST του υποσυστήματος Android με τη βάση δεδομένων του συστήματος; αναλαμβάνει την αναζήτηση δεδομένων από τη βάση δεδομένων και επιστρέφει τα δεδομένα σε μορφή αντικειμένων στην κλάση **MobileService**. Αντίστροφα δέχεται αντικείμενα προς ενημέρωση και πραγματοποιεί τις απαιτούμενες αλλαγές στη βάση δεδομένων. Οι λειτουργίες αυτές που σχετίζονται με τη βάση πραγματοποιούνται κάνοντας χρήση των παρακάτω μεθόδων.

Η μέθοδος **findLSectionsByAcadPeriod** επιστρέφει τις ομάδες παρακολούθησης μαθήματος για συγκεκριμένο μάθημα και συγκεκριμένη ακαδημαϊκή περίοδο.

Η μέθοδος **findLastAcadPeriod** επιστρέφει την τελευταία ακαδημαϊκή περίοδο.

Η μέθοδος **findLessonSectionsByView** επιστρέφει τις ομάδες παρακολούθησης μαθήματος για συγκεκριμένο φοιτητή σε λίστα αντικειμένων τύπου **SLSVIEW**.

Η μέθοδος **findRegisteredLessonsWithLSection** επιστρέφει μια λίστα με τα μαθήματα στα οποία ο φοιτητής έχει δηλώσει ομάδα παρακολούθησης.

Η μέθοδος **findREG_SLS** βρίσκει τη δήλωση μιας ομάδας παρακολούθησης μαθήματος και επιστρέφει ένα αντικείμενο REG_SLS.

Η μέθοδος **findRegisteredLessonsWithoutLSection** επιστρέφει μια λίστα με τα μαθήματα στα οποία ο φοιτητής δεν έχει δηλώσει ομάδα παρακολούθησης.

Οι μέθοδοι **findStudentByAccount**, **findStudentByAem**, και **findStudentByUsername** επιστρέφουν το φοιτητή με κριτήρια αναζήτησης αντίστοιχα το μητρώο, το όνομα λογαριασμού χρήστη και κωδικό, το όνομα λογαριασμού χρήστη.

Η μέθοδος **registerSLS** δέχεται ένα αντικείμενο REG_SLS που αντιστοιχεί σε μια δήλωση μιας ομάδας μαθήματος από φοιτητή και ενημερώνει τη βάση.

Η μέθοδος **unregisterSLS** διαγράφει μια δηλωμένη ομάδα παρακολούθησης μαθήματος ενός φοιτητή από τη βάση.

Η κλάση **MobileService** παρέχει τις απαραίτητες μεθόδους Rest ώστε η εφαρμογή των φοιτητών του υποσυστήματος android να μπορέσει να επικοινωνήσει με τη βάση δεδομένων του συστήματος (oracle database back-end). Κάθε μέθοδος rest από αυτές που ακολουθούν έχει μια αντίστοιχη μέθοδο στην κλάση DbEjb.

Η μέθοδος **StudentLogin**(USERNAME, PASSWORD) δέχεται το όνομα λογαριασμού και τον κωδικό ενός φοιτητή και επιστρέφει ένα αντικείμενο τύπου Student αν τα στοιχεία είναι έγκυρα. Η εγκυρότητα των στοιχείων εξασφαλίζεται καλώντας την αντίστοιχη μέθοδο **findStudentByAccount** s από την κλάση DbEjb.

Η μέθοδος **findRegisteredLessonsWithoutLSection**(STUDENT_AEM, CrPeriod, STUDENT_SEMESTER) δέχεται ως όρισμα το μητρώο του φοιτητή και επιστρέφει λίστα με τα μαθήματα που έχει δηλώσει ο φοιτητής στο σύστημα ηλεκτρονικής γραμματείας αλλά δεν έχει δηλώσει ομάδα παρακολούθησης. Η μέθοδος καλεί την αντίστοιχη μέθοδο **findRegisteredLessonsWithoutLSection** από την κλάση DbEjb. Η παράμετρος CrPeriod εκφράζει αν είναι ενεργοποιημένο το κριτήριο προτεραιότητας εξαμήνου ή όχι. Αν είναι ενεργοποιημένο η παράμετρος STUDENT_SEMESTER περιέχει το εξάμηνο του φοιτητή.

Η μέθοδος **findLessonSectionsByView**(ACAD_PERIOD_ID, STUDENTAEM) επιστρέφει τις ομάδες μαθήματος για συγκεκριμένο φοιτητή σε λίστα αντικειμένων τύπου SLSVIEW. Η μέθοδος καλεί την αντίστοιχη μέθοδο findLessonSectionsByView από την κλάση DbEjb

Η μέθοδος **registerSLS**(ACAD_PERIOD_ID, STUDENTAEM, LESSONCODE, LSNO, DEPTNO) δηλώνει στο φοιτητή με μητρώο STUDENTAEM στο μάθημα με κωδικό LESSONCODE την ομάδα παρακολούθησης με κωδικό LSNO ενημερώνοντας τη βάση καλώντας την αντίστοιχη μέθοδο registerSLS από την κλάση DbEjb

Η μέθοδος **unregisterSLS**(ACAD_PERIOD_ID, STUDENTAEM, LESSONCODE, DEPTNO) αφαιρεί δήλωση από το φοιτητή με μητρώο STUDENTAEM την ομάδα παρακολούθησης του μαθήματος με κωδικό LESSONCODE ενημερώνοντας τη βάση καλώντας την αντίστοιχη μέθοδο unregisterSLS από την κλάση DbEjb.

Η μέθοδος **findLSectionsByAcadPeriod**(ACAD_PERIOD_ID, LESSONCODE) επιστρέφει τις ομάδες μαθήματος για το μάθημα με κωδικό LESSONCODE και συγκεκριμένη ακαδημαϊκή περίοδο. Η μέθοδος καλεί την αντίστοιχη μέθοδο findLSectionsByAcadPeriod από την κλάση DbEjb.

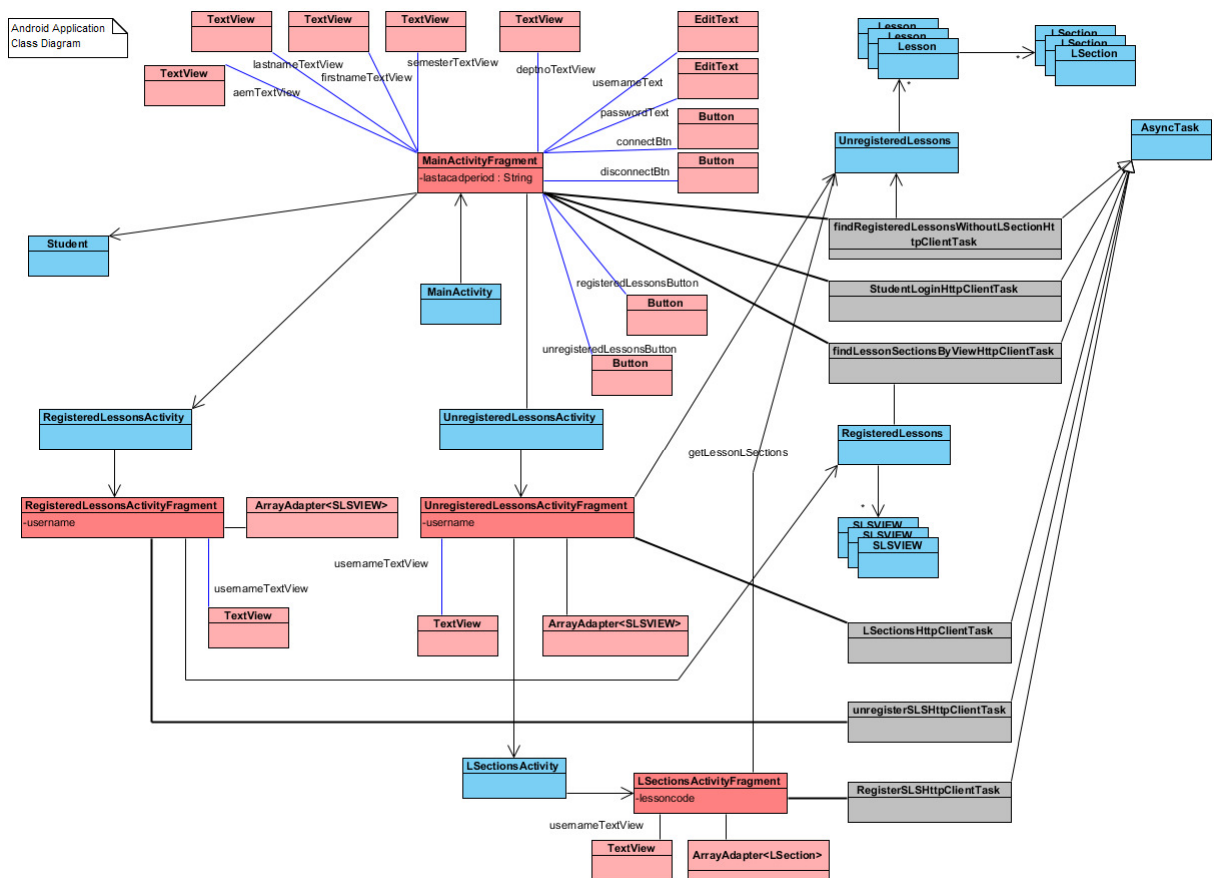
Η μέθοδος **findLastAcadPeriod**(DEPTNO) επιστρέφει την τελευταία ακαδημαϊκή περίοδο καλώντας την αντίστοιχη μέθοδο findLastAcadPeriod από την κλάση DbEjb.

Οι κλάσεις **ACAD_PERIOD**, **Student**, **Lesson**, **LSection**, **SLSVIEW** είναι ίδιες με αυτές του υποσυστήματος φοιτητών.

Η κλάση **REG_SLS** εκφράζει τη δήλωση ομάδας παρακολούθησης μαθήματος ενός φοιτητή σε συγκεκριμένη ακαδημαϊκή περίοδο και περιέχει τον κωδικό ακαδημαϊκής περιόδου (ACAD_PERIOD_ID), το μητρώο του φοιτητή (STUDENTAEM), τον κωδικό του μαθήματος (LESSONCODE), τον κωδικό ομάδας παρακολούθησης (LSNO) και το τμήμα του φοιτητή (DEPTNO).

Διάγραμμα κλάσεων (Σχήμα 2.11) υποσυστήματος Android: **Εφαρμογή φοιτητών**

Το διάγραμμα κλάσεων της Android εφαρμογής φοιτητών αποτελείται κυρίως από τις κλάσεις που λειτουργούν ως οθόνες διεπαφής –με τα οπτικά αντικείμενα τους- : κύρια οθόνη διεπαφής (MainActivityFragment), δηλωμένα μαθήματα (RegisteredLessonsActivityFragment), μη δηλωμένα μαθήματα (UnregisteredLessonsActivityFragment), ομάδες παρακολούθησης μαθήματος (LSectionsActivityFragment), καθώς και από τις κλάσεις που κάνουν αιτήματα κλήσεις rest προς το Android Rest back-end: αίτηση σύνδεσης φοιτητή (StudentLoginHttpClientTask), αίτηση για εύρεση μαθημάτων χωρίς δηλωμένη ομάδα παρακολούθησης (findRegisteredLessonsWithoutLSectionHttpClientTask), αίτηση για εύρεση μαθημάτων με τις δηλωμένες ομάδες παρακολούθησης (findLessonSectionsByViewHttpClientTask), αίτηση για εύρεση ομάδων παρακολούθησης μαθήματος (LSectionsHttpClientTask), αίτηση για ακύρωση δήλωσης μαθήματος (unregisterSLSHttpClientTask), αίτηση για δήλωση ομάδας παρακολούθησης σε μάθημα (RegisterSLSHttpClientTask).



Σχήμα 2.11 Διάγραμμα κλάσεων εφαρμογής Android φοιτητών

Η λεπτομερής περιγραφή των κλάσεων του υποσυστήματος φοιτητών Android παρουσιάζεται στην ενότητα 6 του παραρτήματος Α.

2.3 Μοντελοποίηση διεργασιών(BPM Processes)

Για την ικανοποιητική(αποδοτική, ποιοτική) υποστήριξη των λειτουργικών απαιτήσεων εκτός από την κλασική μοντελοποίηση των επιμέρους νέων υποσυστημάτων, απαιτείται η σχεδίαση και η υλοποίηση σημαντικών διεργασιών οι οποίες θα διασυνδέονται με τα υπάρχοντα αλλά και τα νέα υποσυστήματα. Έχουμε μοντελοποιήσει τέσσερις σημαντικές BPM διεργασίες οι οποίες αναλύονται στις επόμενες παραγράφους. Η μοντελοποίηση έχει γίνει με την τυποποιημένη γλώσσα BPMN 2.0 (Business Process Model and Notation Version 2.0) [19]. Στη φάση της υλοποίησης κάθε μοντέλο εκτελείται σε πραγματικό χρόνο από τη μηχανή εκτέλεσης διεργασιών (Process Engine) [19]. Κάθε μοντέλο για να μπορέσει να εκτελεστεί από το process engine χρειάζεται και την ανάλογη παραμετροποίηση. Η παραμετροποίηση περιλαμβάνει τις καθολικές μεταβλητές της διεργασίας, τις τοπικές μεταβλητές κάθε εργασίας (task) και τον ορισμό των συνθηκών ροής εκτέλεσης. Το process engine δεν επιτρέπει στις εργασίες να βλέπουν απευθείας τις καθολικές μεταβλητές της διεργασίας. Για να αποκτήσει πρόσβαση κάποια εργασία εκτός από τις τοπικές μεταβλητές της περιλαμβάνει και κάποιες άλλες τοπικές μεταβλητές που έχουν την έννοια του *Input/Output Data Mapping* [20]. Η παραμετροποίηση αυτή περιγράφεται στην παράγραφο σχεδίαση διεργασίας σε κάθε BPM διεργασία που ακολουθεί.

2.3.1 Διεργασία συγχρονισμού λογαριασμού νέου χρήστη (NewUser BPM Process)

Η διεργασία αυτή εξασφαλίζει πως αν η γραμματεία δημιουργήσει μεμονωμένα ένα χρήστη στο γραμματειακό σύστημα τότε ο λογαριασμός του θα συγχρονιστεί άμεσα με τον LDAP Server του ιδρύματος καθώς και με το σύστημα υποστήριξης προγραμμάτων σπουδών. Η διεργασία αυτή ακολουθεί την εξής επιχειρησιακή λογική.

Η γραμματεία δημιουργεί τα στοιχεία σύνδεσης του νέου χρήστη στο γραμματειακό σύστημα.

Η γραμματεία αποστέλλει e-mail στο support@teiemt.gr (test1@teiemt.gr). Το e-mail έχει συγκεκριμένο θέμα (π.χ. *NewUser :testuser1*).

Περιοδικά δημιουργείται νέα διεργασία (process) διαχείρισης νέων χρηστών LDAP.

Η διεργασία διαβάζει αν έχει έρθει e-mail.

Αν δεν έχει έρθει e-mail η διεργασία τερματίζεται, διαφορετικά ελέγχει αν το μήνυμα είναι εντάξει (δηλαδή αν έχει κατάλληλο θέμα και αν το περιεχόμενο περιέχει τη σωστή μορφή που αφορά το όνομα λογαριασμού-username).

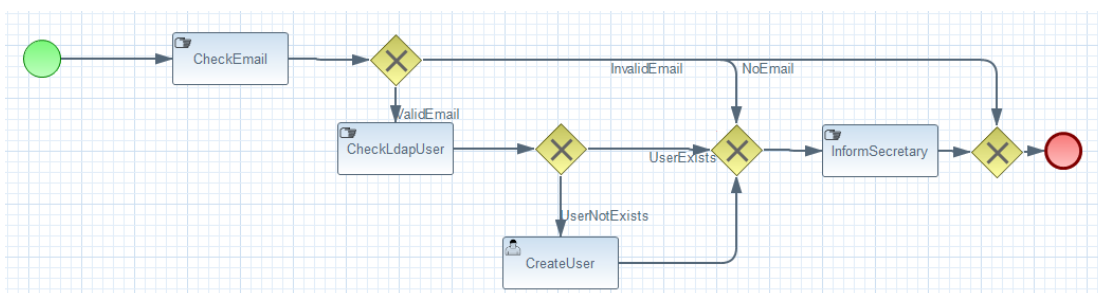
Αν το μήνυμα δεν είναι ορθό αποστέλλεται e-mail στη γραμματεία αποστολέα με ενημέρωση σφάλματος και η διεργασία τερματίζεται.

Αν το μήνυμα είναι ορθό τότε γίνεται αναζήτηση στον LDAP για την ύπαρξη του χρήστη.

Αν ο χρήστης υπάρχει ήδη στον LDAP αποστέλλεται e-mail στη γραμματεία με κατάλληλη ενημέρωση και στη συνέχεια η διεργασία τερματίζεται.

Αν ο χρήστης δεν υπάρχει στον LDAP τότε αναζητούνται τα στοιχεία του από τη βάση δεδομένων του γραμματειακού συστήματος, δημιουργείται στον LDAP με βάση τα στοιχεία που βρέθηκαν, ο χρήστης δημιουργείται επίσης στη βάση του πληροφοριακού συστήματος υποστήριξης προγραμμάτων σπουδών, η γραμματεία ενημερώνεται με e-mail και η διεργασία τερματίζεται.

Σχεδιασμός Διεργασίας (Σχήμα 2.12)



Σχήμα 2.12: BPMN μοντέλο διεργασίας NewUser

Μεταβλητές-Συνθήκες ελέγχου (Πίνακες 2.1-2.9)

Παράμετροι Διεργασίας		
Id	NewUserProcess	

Καθολικές μεταβλητές διεργασίας	CheckEmailStatus	Κατάσταση Email
	CheckLdapUserStatus	Κατάσταση νέου χρήστη στον LDAP
	InformSecretaryMessage	Μήνυμα προς τη γραμματεία
	username	Αιτούμενο όνομα λογαριασμού χρήστη
Χρήστες (Actors) που συμμετέχουν στη διεργασία	Γραμματεία, Διαχειριστής	
Υποσυστήματα που χρησιμοποιούν τη διεργασία	Υποσύστημα Συγχρονισμού	

Πίνακας 2.1: Παράμετροι διεργασίας NewUser

Παράμετροι εργασίας CheckEmail		
Name	CheckEmail	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
Τοπικές μεταβλητές εξόδου	outputInformSecretaryMessage	Μήνυμα προς τη γραμματεία
	outputCheckEmailStatus	Κατάσταση Email
	outputUsername	Αιτούμενο όνομα λογαριασμού χρήστη

Πίνακας 2.2: Παράμετροι εργασίας CheckEmail

Παράμετροι εργασίας CheckLdapUser

Name	CheckLdapUser	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
	inputUsername	Αιτούμενο όνομα λογαριασμού χρήστη
Τοπικές μεταβλητές εξόδου	outputInformSecretaryMessage	Μήνυμα προς τη γραμματεία
	outputCheckLdapUserStatus	

Πίνακας 2.3: Παράμετροι εργασίας CheckLdapUser

Παράμετροι εργασίας CreateUser		
Name	CreateUser	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
	inputUsername	Αιτούμενο όνομα λογαριασμού χρήστη
Τοπικές μεταβλητές εξόδου	outputInformSecretaryMessage	Μήνυμα προς τη γραμματεία

Πίνακας 2.4: Παράμετροι εργασίας CreateUser

Παράμετροι εργασίας InformSecretary		
Name	InformSecretary	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
	inputInformSecretaryMessage	Μήνυμα προς τη γραμματεία

Πίνακας 2.5: Παράμετροι εργασίας InformSecretary

Πύλη 1		
Τύπος	Diverging	
Συνθήκες διακλάδωσης	InvalidEmail	Μη έγκυρο e-mail
	ValidEmail	Έγκυρο e-mail
	Noemail	Δεν ήρθε e-mail

Πίνακας 2.6: Παράμετροι Πύλης 1 της διεργασίας

Πύλη 2		
Τύπος	Diverging	
Συνθήκες διακλάδωσης	UserExists	Ο χρήστης υπάρχει στον LDAP
	UserNotExists	Ο χρήστης δεν υπάρχει στον LDAP

Πίνακας 2.7: Παράμετροι Πύλης 2 της διεργασίας

Πύλη 3		
Τύπος	Converging	

Πίνακας 2.8: Παράμετροι Πύλης 3 της διεργασίας

Πύλη 4		
Τύπος	Converging	

Πίνακας 2.9: Παράμετροι Πύλης 4 της διεργασίας

Περιγραφή εκτέλεσης

Το υποσύστημα συγχρονισμού δημιουργεί ένα στιγμιότυπο αυτής της διεργασίας καλώντας τη Rest μέθοδο **createNewUserProcess** που προσφέρεται από το υποσύστημα παροχής διεργασιών.

Με τη δημιουργία της διεργασίας το BPM Process Engine αυτομάτως καλεί την πρώτη εργασία (task): *CheckEmail*. Επειδή η εργασία είναι domain specific (όπως είναι όλες οι εργασίες που εμείς έχουμε υλοποιήσει) εκτελείται ο κώδικας που έχουμε γράψει για τον έλεγχο του e-mail. Αν δεν έχει έρθει e-mail η διεργασία τερματίζεται. Αν έχει έρθει e-mail αλλά δεν έχει τη μορφή που πρέπει τότε εκτελείται η επόμενη εργασία *InformSecretary* η οποία στέλνει ενημερωτικό e-mail στην αντίστοιχη γραμματεία.

Αν το e-mail είναι έγκυρο τότε καλείται η εργασία *CheckLapUser*. Η εργασία αυτή διαβάζει το username που υπήρχε στο e-mail και ελέγχει αν ο χρήστης υπάρχει στον LDAP του ιδρύματος. Αν ο χρήστης υπάρχει εκτελείται η εργασία *InformSecretary* η οποία στέλνει ενημερωτικό e-mail στην αντίστοιχη γραμματεία.

Σημείωση

Όλα οι προηγούμενες εργασίες τρέχουν σε *synchronous mode*, και η εκτέλεση τους γίνεται αυτομάτως χωρίς να απαιτούν ανθρώπινη παρέμβαση. Μια ειδική κατηγορία, τα Human Tasks συνήθως τρέχουν σε *asynchronous mode* που σημαίνει πως διακόπτεται η εκτέλεση της ροής από το process engine. Όταν ο χρήστης κάνει την εργασία που πρέπει ενημερώνεται το process engine πως ολοκληρώθηκε η εργασία και η ροή της διεργασίας συνεχίζεται.

Αν ο χρήστης δεν υπάρχει εκτελείται το Human task *CreateUser* σε *asynchronous mode*. Το υποσύστημα συγχρονισμού καλεί τη rest μέθοδο **getNewUserRequestsUsernames** η οποία επιστρέφει τους χρήστες που αναμένουν να δημιουργηθούν. Ο χρήστης αναζητά τα στοιχεία του χρήστη από τη βάση δεδομένων του γραμματειακού συστήματος και με βάση αυτά τα στοιχεία δημιουργεί το χρήστη στον ldap του ιδρύματος και στη συνέχεια προσθέτει το χρήστη στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών. Στη συνέχεια καλεί τη rest

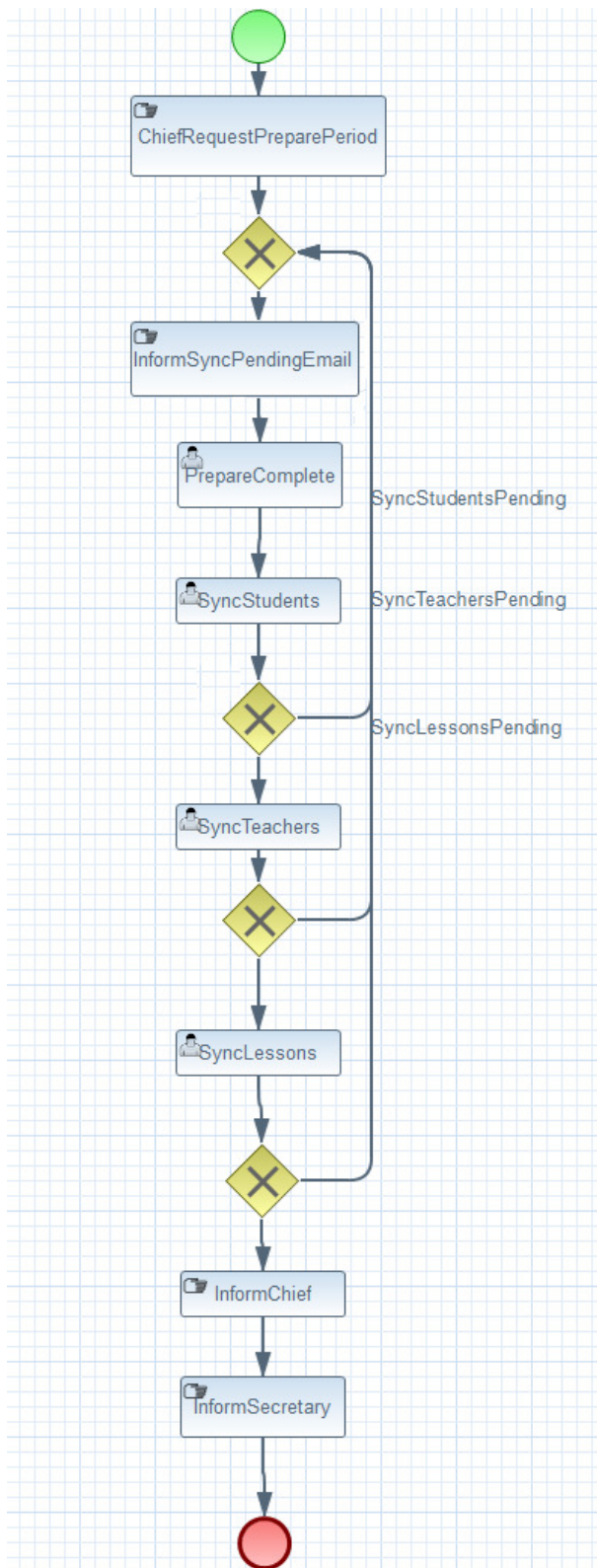
μέθοδο **CreateUser** σηματοδοτώντας την ολοκλήρωση της εργασίας *CreateUser* και το process engine δίνει τον έλεγχο στην εργασία *InformSecretary* (για ενημέρωση με e-mail της γραμματείας).

Το process engine τερματίζει τη διεργασία.

2.3.2 Διεργασία προετοιμασίας νέας ακαδημαϊκής περιόδου (NewAcadPeriod BPM Process)

Για να δημιουργηθεί μια νέα ακαδημαϊκή περίοδο είναι απαραίτητος ο συγχρονισμός του συστήματος υποστήριξης προγραμμάτων σπουδών με στοιχεία (ενεργών φοιτητών περιόδου, καθηγητών περιόδου και μαθημάτων περιόδου) που βρίσκονται στο γραμματειακό σύστημα. Για να γίνει ορθά ο συγχρονισμός είναι απαραίτητος ο συντονισμός εργασιών που εκτελούν ο διαχειριστής, ο προϊστάμενος και η γραμματεία κάθε τμήματος. Έτσι ο προϊστάμενος αιτείται τη δημιουργία της νέας ακαδημαϊκής περιόδου, η γραμματεία φροντίζει την επικαιροποίηση των στοιχείων του γραμματειακού συστήματος - ώστε να έχει νόημα ο συγχρονισμός από το διαχειριστή- και ο διαχειριστής αναλαμβάνει το συγχρονισμό των στοιχείων.

Σχεδίαση Διεργασίας (Σχήμα 2.13)



Σχήμα 2.13 BPMN μοντέλο διεργασίας NewAcadPeriod

Μεταβλητές-Συνθήκες ελέγχου(Πίνακες 2.10-2.22)

Παράμετροι Διεργασίας		
Id	NewAcadPeriodProcess	
Καθολικές μεταβλητές διεργασίας	DeptCode	Κωδικός τμήματος
	AYEAR	Έτος ακαδημαϊκής περιόδου
	TERM	Περίοδος ακαδημαϊκής περιόδου
	ChiefEmail	E-mail προϊσταμένου
	SecretaryEmail	E-mail γραμματείας τμήματος
	SyncStudentsDone	Κατάσταση ολοκλήρωσης συγχρονισμού φοιτητών
	SyncTeachersDone	Κατάσταση ολοκλήρωσης συγχρονισμού εκπαιδευτικών
	SyncLessonsDone	Κατάσταση ολοκλήρωσης συγχρονισμού μαθημάτων
	InvalidEmailMessage	Μήνυμα εσφαλμένου e-mail
	SyncPendingMessage	Μήνυμα συγχρονισμού σε εκκρεμότητα
	SyncCompleteMessage	Μήνυμα ολοκλήρωσης συγχρονισμού

Χρήστες (Actors) που συμμετέχουν στη διεργασία	Προϊστάμενος, Γραμματεία, Διαχειριστής	
Υποσυστήματα που χρησιμοποιούν τη διεργασία	Υποσύστημα Εκπαιδευτικού, Υποσύστημα Συγχρονισμού	

Πίνακας 2.10: Παράμετροι διεργασίας NewAcadPeriod

Παράμετροι εργασίας ChiefRequestPreparePeriod		
Name	ChiefRequestPreparePeriod	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
	inputChiefEmail	E-mail προϊσταμένου
	inputSecretaryEmail	E-mail γραμματείας τμήματος
	inputDeptCode	Κωδικός τμήματος
Τοπικές μεταβλητές εξόδου	outputSyncStudentsDone	Κατάσταση ολοκλήρωσης συγχρονισμού φοιτητών
	outputSyncTeachersDone	Κατάσταση ολοκλήρωσης συγχρονισμού εκπαιδευτικών
	outputSyncLessonsDone	Κατάσταση ολοκλήρωσης συγχρονισμού μαθημάτων
	outputSyncPendingMessage	Μήνυμα συγχρονισμού σε εκκρεμότητα

Πίνακας 2.11: Παράμετροι εργασίας ChiefRequestPreparePeriod

Παράμετροι εργασίας InformSyncPendingEmail		
Name	InformSyncPendingEmail	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
	inputSyncPendingMessage	Μήνυμα συγχρονισμού σε εκκρεμότητα

Πίνακας 2.12: Παράμετροι εργασίας InformSyncPendingEmail

Παράμετροι εργασίας PrepareComplete		
Name	PrepareComplete	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
	inputDeptCode	Κωδικός τμήματος
	inputAYEAR	Έτος ακαδημαϊκής περιόδου
	inputTERM	Περίοδος ακαδημαϊκής περιόδου

Πίνακας 2.13: Παράμετροι εργασίας PrepareComplete

Παράμετροι εργασίας SyncStudents		
Name	SyncStudents	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
	inputDeptCode	Κωδικός τμήματος
	inputAYEAR	Έτος ακαδημαϊκής περιόδου

	inputTERM	Περίοδος ακαδημαϊκής περιόδου
Τοπικές μεταβλητές εξόδου	outputSyncStudentsDone	Κατάσταση ολοκλήρωσης συγχρονισμού φοιτητών
	outputSyncPendingMessage	Μήνυμα συγχρονισμού σε εκκρεμότητα

Πίνακας 2.14: Παράμετροι εργασίας SyncStudents

Παράμετροι εργασίας SyncTeachers		
Name	SyncTeachers	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
Τοπικές μεταβλητές εξόδου	outputSyncTeachersDone	Κατάσταση ολοκλήρωσης συγχρονισμού εκπαιδευτικών
	outputSyncPendingMessage	Μήνυμα συγχρονισμού σε εκκρεμότητα

Πίνακας 2.15: Παράμετροι εργασίας SyncTeachers

Παράμετροι εργασίας SyncLessons		
Name	SyncLessons	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
Τοπικές μεταβλητές εξόδου	outputSyncLessonsDone	Κατάσταση ολοκλήρωσης συγχρονισμού μαθημάτων

	outputSyncPendingMessage	Μήνυμα συγχρονισμού σε εκκρεμότητα
	outputSyncCompleteMessage	Μήνυμα ολοκλήρωσης συγχρονισμού

Πίνακας 2.16: Παράμετροι εργασίας SyncLessons

Παράμετροι εργασίας InformChief		
Name	InformChief	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
	inputSyncCompleteMessage	Μήνυμα ολοκλήρωσης συγχρονισμού

Πίνακας 2.17: Παράμετροι εργασίας InformChief

Παράμετροι εργασίας InformSecretary		
Name	InformSecretary	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
	inputSyncCompleteMessage	Μήνυμα ολοκλήρωσης συγχρονισμού

Πίνακας 2.18: Παράμετροι εργασίας InformSecretary

Πύλη 1		
Τύπος	Converging	

Πίνακας 2.19: Παράμετροι Πύλης 1 της διεργασίας

Πύλη 2		
Τύπος	Diverging	
Συνθήκες διακλάδωσης	SyncStudentsPending==true	Οι φοιτητές δεν έχουν συγχρονιστεί
	SyncStudentsPending==false	Οι φοιτητές έχουν συγχρονιστεί

Πίνακας 2.20: Παράμετροι Πύλης 2 της διεργασίας

Πύλη 3		
Τύπος	Diverging	
Συνθήκες διακλάδωσης	SyncTeachersPending==true	Οι εκπαιδευτικοί δεν έχουν συγχρονιστεί
	SyncTeachersPending==false	Οι εκπαιδευτικοί δεν έχουν συγχρονιστεί

Πίνακας 2.21: Παράμετροι Πύλης 3 της διεργασίας

Πύλη 4		
Τύπος	Diverging	
Συνθήκες διακλάδωσης	SyncLessonsPending==true	Τα μαθήματα δεν έχουν συγχρονιστεί
	SyncLessonsPending==false	Τα μαθήματα έχουν συγχρονιστεί

Πίνακας 2.22: Παράμετροι Πύλης 4 της διεργασίας

Περιγραφή εκτέλεσης

Από το υποσύστημα εκπαιδευτικού ο προϊστάμενος αιτείται την δημιουργία της νέας ακαδημαϊκής περιόδου στο σύστημα υποστήριξης προγραμμάτων σπουδών. Η αίτηση υποβάλλεται καλώντας τη Rest μέθοδο **createNewAcadPeriodProcess** από το υποσύστημα παροχής διεργασιών η οποία εκκινεί τη διεργασία. Έτσι εκτελείται η πρώτη εργασία (*ChiefRequestPreparePeriod*) όπου στέλνεται ενημερωτικό e-mail στη γραμματεία του τμήματος ώστε να επικαιροποιήσει τους ενεργούς φοιτητές, τους εκπαιδευτικούς και τα μαθήματα της περιόδου.

Η γραμματεία διαβάζει το e-mail -στο οποίο εμπεριέχεται ειδικός σύνδεσμος (που καλεί τη Rest μέθοδο **prepareCompleteAsyncTask** από το υποσύστημα παροχής διεργασιών) τον οποίο επιλέγει αφού προετοιμάσει τα στοιχεία των φοιτητών, των εκπαιδευτικών και των μαθημάτων για τη νέα ακαδημαϊκή περίοδο (εργασία *PrepareComplete*). Μόλις επιλέξει το σύνδεσμο ολοκληρώνεται η εργασία και ενεργοποιείται η εργασία *SyncStudents* που πρέπει να ολοκληρωθεί από το διαχειριστή στο υποσύστημα συγχρονισμού.

Το υποσύστημα συγχρονισμού εκτελεί σε τακτά χρονικά διαστήματα ελέγχους για εκκρεμείς διεργασίες. Αν ο διαχειριστής δει σε εκκρεμότητα την εργασία *SyncStudents* πρέπει να συγχρονίσει τους ενεργούς φοιτητές του τμήματος. Αν για οποιοδήποτε λόγο διαπιστώσει πως οι ενεργοί φοιτητές δεν είναι επικαιροποιημένοι τότε η εργασία *SyncStudents* ολοκληρώνεται και η ροή της διεργασία στέλνεται στην εργασία *InformSyncPendingEmail* όπου στέλνεται ξανά e-mail στη γραμματεία.

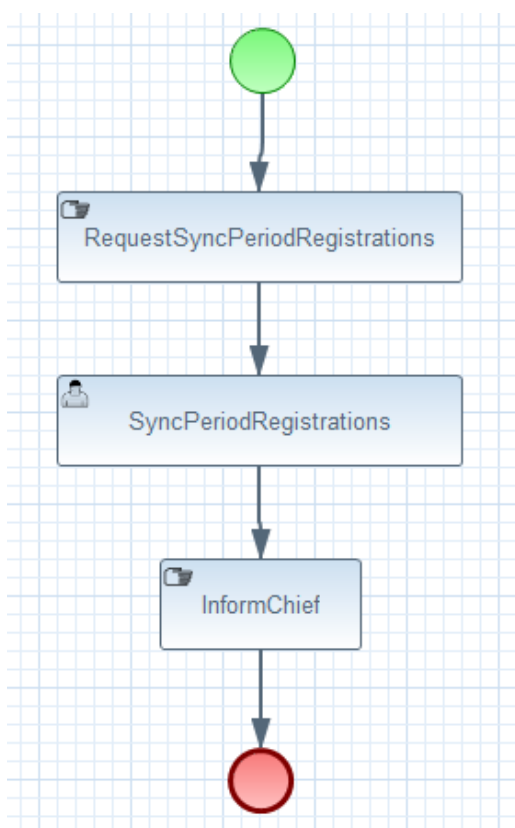
Αν οι ενεργοί φοιτητές είναι επικαιροποιημένοι τότε ο διαχειριστής συγχρονίζει τα στοιχεία των φοιτητών του γραμματειακού συστήματος με το σύστημα υποστήριξης προγραμμάτων σπουδών και στη συνέχεια καλείται η Rest μέθοδος **completeCurrentAsyncTask** η οποία ολοκληρώνει την τρέχουσα εργασία. Έτσι ο έλεγχος δίνεται στην εργασία *SyncTeachers* και στη συνέχεια στην εργασία *SyncLessons* όπου εκτελούνται με την ίδια λογική όπως και η εργασία *SyncStudents*.

Αφού συγχρονιστούν όλα τα στοιχεία αποστέλλεται e-mail στον προϊστάμενο(εργασία *InformChief*) και στη γραμματεία (εργασία *InformSecretary*) για να έχουν γνώση πως η ακαδημαϊκή περίοδος δημιουργήθηκε με επιτυχία, ώστε να συνεχίσουν περεταίρω τις ενέργειές τους.

2.3.3 Διεργασία συγχρονισμού δηλώσεων ακαδημαϊκής περιόδου (SyncPeriodRegistrations BPM Process)

Όταν γίνει συγχρονισμός δηλώσεων σε μια ακαδημαϊκή περίοδο τότε οι δηλώσεις μαθημάτων των φοιτητών μεταφέρονται από το σύστημα ηλεκτρονικής γραμματείας στο σύστημα υποστήριξης προγραμμάτων σπουδών. Συγχρονισμός δηλώσεων μπορεί να γίνει πολλές φορές κατά την τρέχουσα ακαδημαϊκή περίοδο (όταν οι δηλώσεις ομάδων παρακολούθησης μαθημάτων βρίσκονται σε εξέλιξη) είτε –μια μόνον φορά- σε προγενέστερη ακαδημαϊκή περίοδο. Στη δεύτερη περίπτωση ο συγχρονισμός έχει νόημα ώστε να έχουν εικόνα ο προϊστάμενος και οι εκπαιδευτικοί για το ιστορικό του μαθήματος (επιτυχόντες, μη επιτυχόντες, σύνολο φοιτητών που έχουν δηλώσει το μάθημα κλπ).

Σχεδίαση Διεργασίας (Σχήμα 2.14)



Σχήμα 2.14 BPMN μοντέλο διεργασίας SyncPeriodRegistrations

Μεταβλητές-Συνθήκες ελέγχου (Πίνακες 2.23-2.26)

Παράμετροι Διεργασίας		
Id	SyncPeriodRegistrationProcess	
Καθολικές μεταβλητές διεργασίας	DeptCode	Κωδικός τμήματος
	AYEAR	Έτος ακαδημαϊκής περιόδου
	TERM	Περίοδος ακαδημαϊκής περιόδου
	ChiefEmail	E-mail προϊσταμένου
	SyncCompleteMessage	Μήνυμα ολοκλήρωσης συγχρονισμού
Χρήστες (Actors) που συμμετέχουν στη διεργασία	Προϊστάμενος, Διαχειριστής	
Υποσυστήματα που χρησιμοποιούν τη διεργασία	Υποσύστημα Εκπαιδευτικού, Υποσύστημα Συγχρονισμού	

Πίνακας 2.23: Παράμετροι διεργασίας SyncPeriodRegistrations

Παράμετροι εργασίας RequestSyncPeriodRegistrations		
Name	RequestSyncPeriodRegistrations	
Τοπικές μεταβλητές εισόδου	inputChiefEmail	E-mail προϊσταμένου

Πίνακας 2.24: Παράμετροι εργασίας RequestSyncPeriodRegistrations

Παράμετροι εργασίας SyncPeriodRegistrations		
Name	SyncPeriodRegistrations	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
	inputDeptCode	Έτος ακαδημαϊκής περιόδου
	inputAYEAR	Περίοδος ακαδημαϊκής περιόδου
	inputTERM	E-mail προϊσταμένου
Τοπικές μεταβλητές εξόδου	outputSyncCompleteMessage	Μήνυμα ολοκλήρωσης συγχρονισμού

Πίνακας 2.25: Παράμετροι εργασίας SyncPeriodRegistrations

Παράμετροι εργασίας InformChief		
Name	InformChief	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
	inputSyncCompleteMessage	Μήνυμα ολοκλήρωσης συγχρονισμού

Πίνακας 2.26: Παράμετροι εργασίας InformChief

Περιγραφή εκτέλεσης

Από το υποσύστημα εκπαιδευτικού ο προϊστάμενος αιτείται συγχρονισμό δηλώσεων για το τμήμα του και για συγκεκριμένη ακαδημαϊκή περίοδο. Η αίτηση πραγματοποιείται εκτελώντας τη Rest μέθοδο **createSyncPeriodRegistrationsProcess** από το σύστημα παροχής διεργασιών η οποία εκκινεί τη διεργασία και αμέσως εκτελείται η πρώτη εργασία: *RequestSyncPeriodRegistrations* που περιλαμβάνει τα στοιχεία της αίτησης.

Στη συνέχεια τίθεται σε εκκρεμότητα (asynchronous mode) η εργασία συγχρονισμού δηλώσεων (*SyncPeriodRegistrations*), όπου αναμένεται από το διαχειριστή του συστήματος να εκτελέσει το συγχρονισμό.

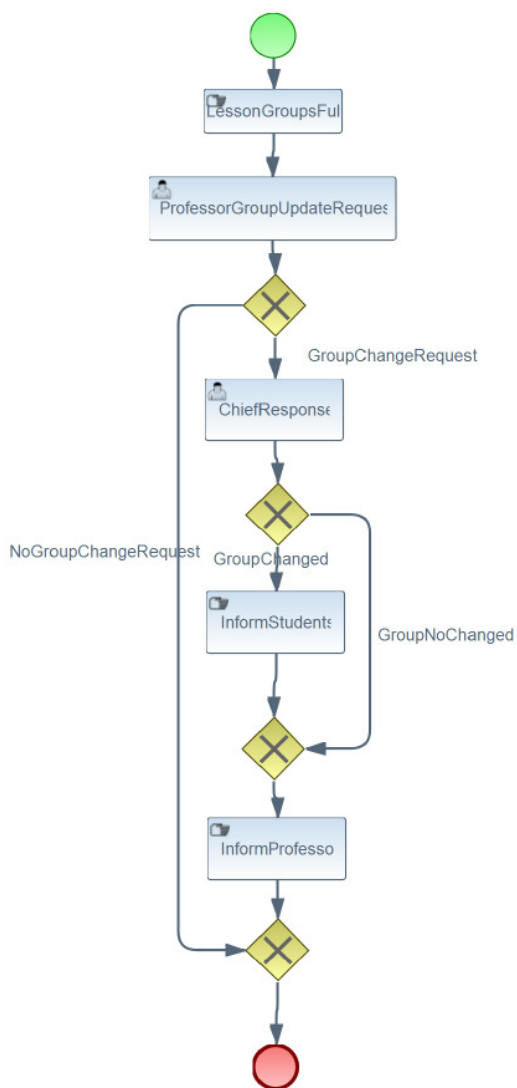
Ο διαχειριστής από το υποσύστημα συγχρονισμού εκτελεί το συγχρονισμό και έτσι προετοιμάζεται το μήνυμα ενημέρωσης προς τον προϊστάμενο του τμήματος. Στη συνέχεια ο διαχειριστής σηματοδοτεί την ολοκλήρωσή της εργασίας *SyncPeriodRegistrations* εκτελώντας τη Rest μέθοδο **completeSyncPeriodRegistrationsAsyncTask** από το υποσύστημα παροχής διεργασιών.

Η εργασία *InformChief* ενημερώνει με email τον προϊστάμενο του τμήματος και η διεργασία τερματίζεται.

2.3.4 Διεργασία κορεσμού ομάδων παρακολούθησης μαθήματος περιόδου (LessonGroupsFull BPM Process)

Κατά τη διαδικασία δηλώσεων ομάδων παρακολούθησης από τους φοιτητές στα μαθήματα τους υπάρχει πάντα το ενδεχόμενο οι ομάδες να γεμίσουν. Σε αυτή την περίπτωση οι φοιτητές μπορούν να δηλώσουν μια ειδική ομάδα παρακολούθησης με κωδικό ομάδας: ΠΛΕΟΝΑΖΟΝΤΕΣ. Κατά τη δήλωση σε αυτήν την ομάδα ενημερώνεται ο υπεύθυνος εκπαιδευτικός του μαθήματος ώστε να κάνει εκτιμήσεις αν χρειάζεται τροποποίηση στις ομάδες των μαθημάτων του. Με βάση τις εκτιμήσεις του αιτείται στον προϊστάμενο του τμήματος ενημέρωση των ομάδων παρακολούθησης. Ο προϊστάμενος με τη σειρά του κάνει τις απαραίτητες ενέργειες ενημέρωσης και τελικά ενημερώνονται οι φοιτητές που έχουν δηλώσει το μάθημα.

Σχεδιασμός Διεργασίας (Σχήμα 2.15)



Σχήμα 2.15: BPMN μοντέλο διεργασίας LessonGroupsFull

Μεταβλητές-Συνθήκες ελέγχου(Πίνακες 2.27-2.36)

Παράμετροι Διεργασίας			
Id		LessonGroupsFullProcess	
Καθολικές διεργασίας	μεταβλητές	Deptno	Κωδικός τμήματος
		LessonCode	Κωδικός μαθήματος
		ChiefEmail	E-mail προϊσταμένου

		τμήματος
	TeacherEmail	E-mail υπεύθυνου εκπαιδευτικού μαθήματος
	StudentsEmails	E-mails φοιτητών που έχουν δηλωμένο το μάθημα
	GroupChangeFlag	Κατάσταση μεταβολής ομάδων παρακολούθησης από τον προϊστάμενο
	GroupChangeRequestFlag	Κατάσταση υποβολής αιτήματος μεταβολής ομάδων παρακολούθησης προς τον προϊστάμενο
	GroupUpdateSuggestion	Εισήγηση υπεύθυνου μαθήματος προς τον προϊστάμενο του τμήματος
	ChiefResponseMessage	Απάντηση προϊσταμένου προς την εισήγηση του υπεύθυνου μαθήματος και στους φοιτητές που έχουν δηλώσει το μάθημα
Χρήστες (Actors) που συμμετέχουν στη διεργασία	Φοιτητής, Εκπαιδευτικός, Προϊστάμενος	
Υποσυστήματα που χρησιμοποιούν τη διεργασία	Υποσύστημα φοιτητών, Υποσύστημα Εκπαιδευτικών	

Πίνακας 2.27: Παράμετροι διεργασίας LessonGroupsFull

Παράμετροι εργασίας LessonGroupsFull		
Name	LessonGroupsFull	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
	inputDeptno	Κωδικός τμήματος
	inputLessonCode	Κωδικός μαθήματος
	inputTeacherEmail	E-mail υπεύθυνου εκπαιδευτικού μαθήματος

Πίνακας 2.28: Παράμετροι εργασίας LessonGroupsFull

Παράμετροι εργασίας ProfessorGroupUpdateRequest		
Name	ProfessorGroupUpdateRequest	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
Τοπικές μεταβλητές εξόδου	outputGroupChangeRequestFlag	Κατάσταση υποβολής αιτήματος μεταβολής ομάδων παρακολούθησης προς τον προϊστάμενο
	outputGroupUpdateSuggestion	Εισήγηση υπεύθυνου μαθήματος προς τον προϊστάμενο του τμήματος

Πίνακας 2.29: Παράμετροι εργασίας ProfessorGroupUpdateRequest

Παράμετροι εργασίας ChiefResponse		

Name	ChiefResponse	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
	inputDeptno	Κωδικός τμήματος
	inputLessonCode	Κωδικός μαθήματος
	inputGroupUpdateSuggestion	Εισήγηση υπεύθυνου μαθήματος προς τον προϊστάμενο του τμήματος
Τοπικές μεταβλητές εξόδου	outputGroupChangeFlag	Κατάσταση μεταβολής ομάδων παρακολούθησης από τον προϊστάμενο
	outputChiefResponseMessage	Απάντηση προϊσταμένου προς την εισήγηση του υπεύθυνου μαθήματος και στους φοιτητές που έχουν δηλώσει το μάθημα

Πίνακας 2.30: Παράμετροι εργασίας ChiefResponse

Παράμετροι εργασίας InformStudents		
Name	InformStudents	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
	inputStudentsEmails	E-mails φοιτητών που έχουν δηλωμένο το μάθημα
	inputChiefResponseMessage	Απάντηση προϊσταμένου προς την εισήγηση του

		υπεύθυνου μαθήματος και στους φοιτητές που έχουν δηλώσει το μάθημα
--	--	--

Πίνακας 2.31: Παράμετροι εργασίας InformStudents

Παράμετροι εργασίας InformProfessor		
Name	InformProfessor	
Τοπικές μεταβλητές εισόδου	inputTaskname	Όνομα εργασίας
	inputTeacherEmail	E-mail υπεύθυνου εκπαιδευτικού μαθήματος
	inputGroupChangeFlag	Κατάσταση μεταβολής ομάδων παρακολούθησης από τον προϊστάμενο
	inputChiefResponseMessage	Απάντηση προϊσταμένου προς την εισήγηση του υπεύθυνου μαθήματος

Πίνακας 2.32: Παράμετροι εργασίας InformProfessor

Πύλη 1		
Τύπος	Diverging	
Συνθήκες διακλάδωσης	GroupChangeRequest	Αίτημα ενημέρωσης των ομάδων παρακολούθησης προς τον προϊστάμενο
	NoGroupChangeRequest	Επιλογή για μη αποστολή αιτήματος ενημέρωσης των

		ομάδων παρακολούθησης προς τον προϊστάμενο
--	--	--

Πίνακας 2.33: Παράμετροι Πύλης 1 της διεργασίας

Πύλη 2		
Τύπος	Diverging	
Συνθήκες διακλάδωσης	GroupChanged	Ο προϊστάμενος ενημέρωσε τις ομάδες παρακολούθησης του μαθήματος
	GroupNoChanged	Ο προϊστάμενος δεν ενημέρωσε τις ομάδες παρακολούθησης του μαθήματος

Πίνακας 2.34: Παράμετροι Πύλης 2 της διεργασίας

Πύλη 3		
Τύπος	Converging	

Πίνακας 2.35: Παράμετροι Πύλης 3 της διεργασίας

Πύλη 4		
Τύπος	Converging	

Πίνακας 2.36: Παράμετροι Πύλης 4 της διεργασίας

Περιγραφή εκτέλεσης

Αν από το υποσύστημα φοιτητή σε κάποιο μάθημα δηλωθεί επιτυχώς η ομάδα ΠΛΕΟΝΑΖΟΝΤΕΣ τότε εκτελείται η Rest μέθοδος **createLessonGroupsFullProcess** του υποσυστήματος παροχής

διεργασιών και εκκινείται η διεργασία *LessonGroupsFullProcess* και ο έλεγχος δίνεται στην εργασία *LessonGroupsFull* η οποία στέλνει e-mail στον υπεύθυνο εκπαιδευτικό του μαθήματος.

Στη συνέχεια εκτελείται η εργασία *ProfessorGroupUpdateRequest* για να αποφασίσει ο υπεύθυνος καθηγητής αν χρειάζεται να γίνει ενημέρωση των ομάδων παρακολούθησης του μαθήματος. Αν ο υπεύθυνος αποφασίσει θετικά τότε γίνεται αίτημα με την εισήγησή του στον προϊστάμενο καλώντας την Rest μέθοδο **completeProfessorGroupUpdateRequestAsyncTask** από το υποσύστημα παροχής διεργασιών η οποία ολοκληρώνει την εργασία.

Στη συνέχεια τον έλεγχο παίρνει η εργασία *ChiefResponse*. Αν ο προϊστάμενος απορρίψει το αίτημα (καλώντας τη Rest μέθοδο *completeChiefResponseAsyncTask*) τότε ενημερώνεται ο υπεύθυνος εκπαιδευτικός (εργασία *InformProfessor*) και η διεργασία τερματίζεται.

Αν ο προϊστάμενος αποδεχτεί το αίτημα και κάνει τροποποίηση στις ομάδες παρακολούθησης του μαθήματος (π.χ. να δημιουργήσει νέο τμήμα ή να αυξήσει τη χωρητικότητα) τότε καλείται η Rest μέθοδος **completeChiefResponseAsyncTask** (με τις ανάλογες παραμέτρους) και ολοκληρώνεται η εργασία *ChiefResponse*. Στη συνέχεια ο έλεγχος έρχεται στην εργασία *InformStudents* που στέλνει e-mail στους φοιτητές που έχουν δηλώσει το μάθημα. Τέλος ο έλεγχος πηγαίνει στην εργασία *InformProfessor* που στέλνει email στον υπεύθυνο εκπαιδευτικό και η διεργασία τερματίζεται.

Κεφάλαιο 3

Υλοποίηση συστήματος

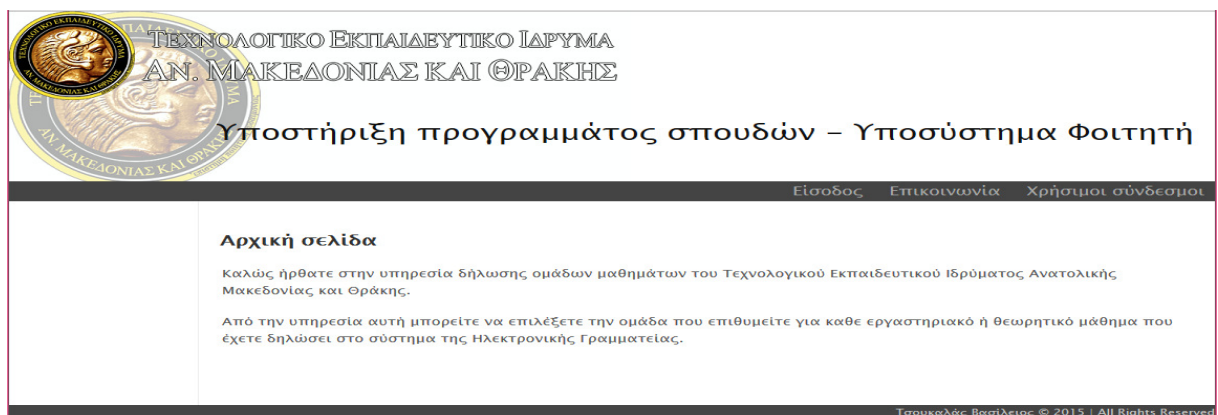
Στο κεφάλαιο αυτό παρουσιάζεται η υλοποίηση του συστήματος που αποτελείται από έξι υποσυστήματα: υποσύστημα φοιτητών, υποσύστημα εκπαιδευτικών, υποσύστημα ωρολογίου προγράμματος, υποσύστημα συγχρονισμού, υποσύστημα παροχής διεργασιών και υποσύστημα Android (*Android Rest backend* και *Android εφαρμογή φοιτητών*). Παρουσιάζονται οι διεπαφές και η χρήση των υποσυστημάτων, η υλοποίηση της βάσης δεδομένων, η αρχιτεκτονική του συστήματος, η διαδικασία εγκατάστασης και τα εργαλεία σχεδίασης – ανάπτυξης του λογισμικού των υποσυστημάτων.

3.1 Διεπαφές υποσυστημάτων

Στις επόμενες παραγράφους παρουσιάζονται η διεπαφή καθενός από τα έξι υποσυστήματα.

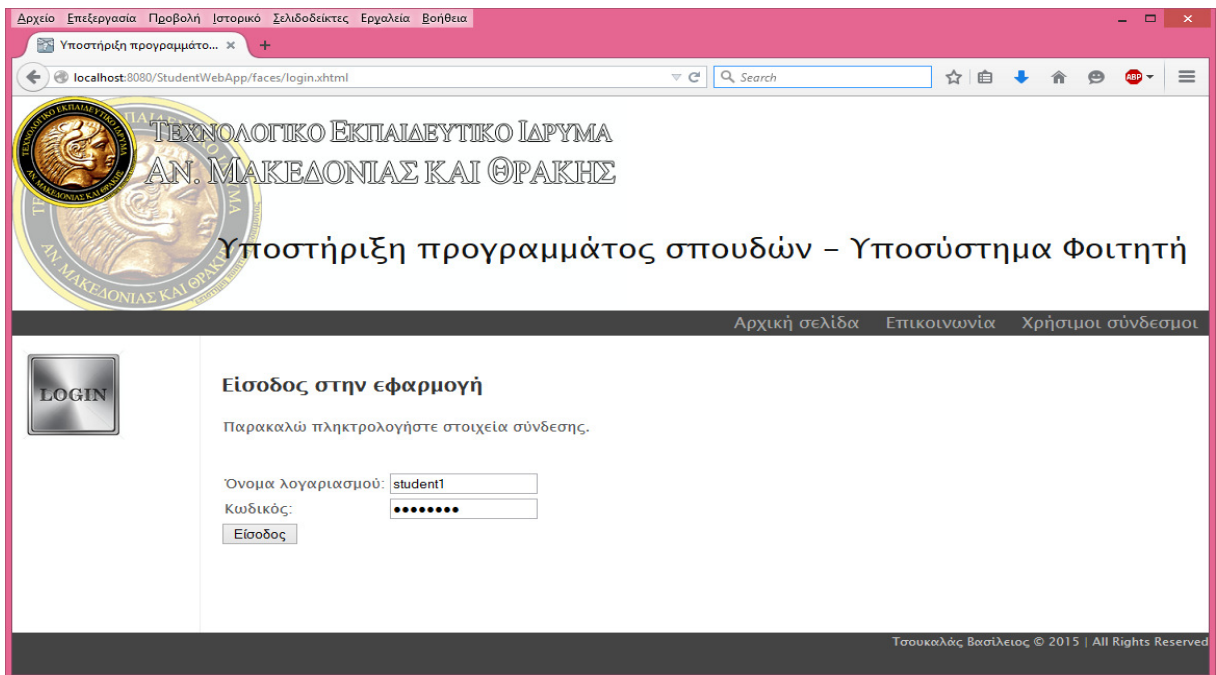
3.1.1 Διεπαφή υποσυστήματος φοιτητών

Η αρχική σελίδα εμφανίζεται στην εικόνα 3.1. Ο φοιτητής επιλέγει είσοδο από το μενού.



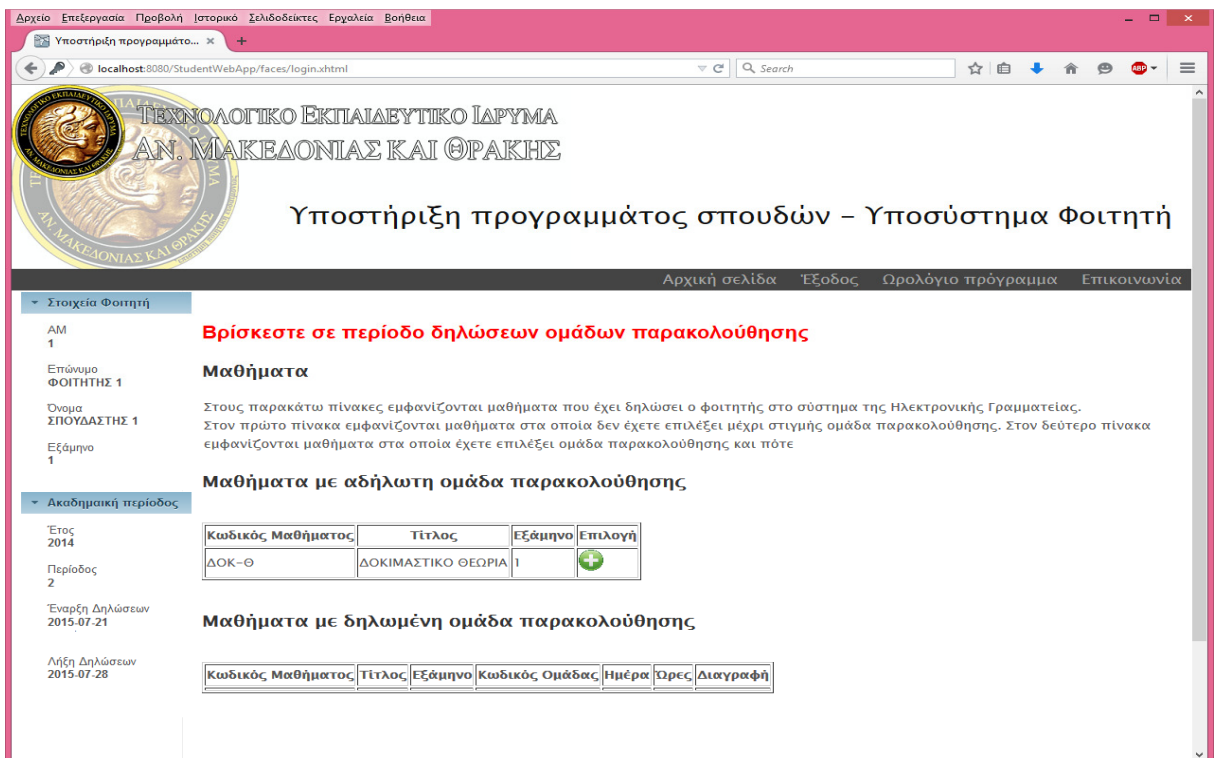
Εικόνα 3.1: Αρχική σελίδα

Στην εικόνα 3.2 εμφανίζεται η σελίδα σύνδεσης στο υποσύστημα.




Εικόνα 3.2: Σελίδα σύνδεσης

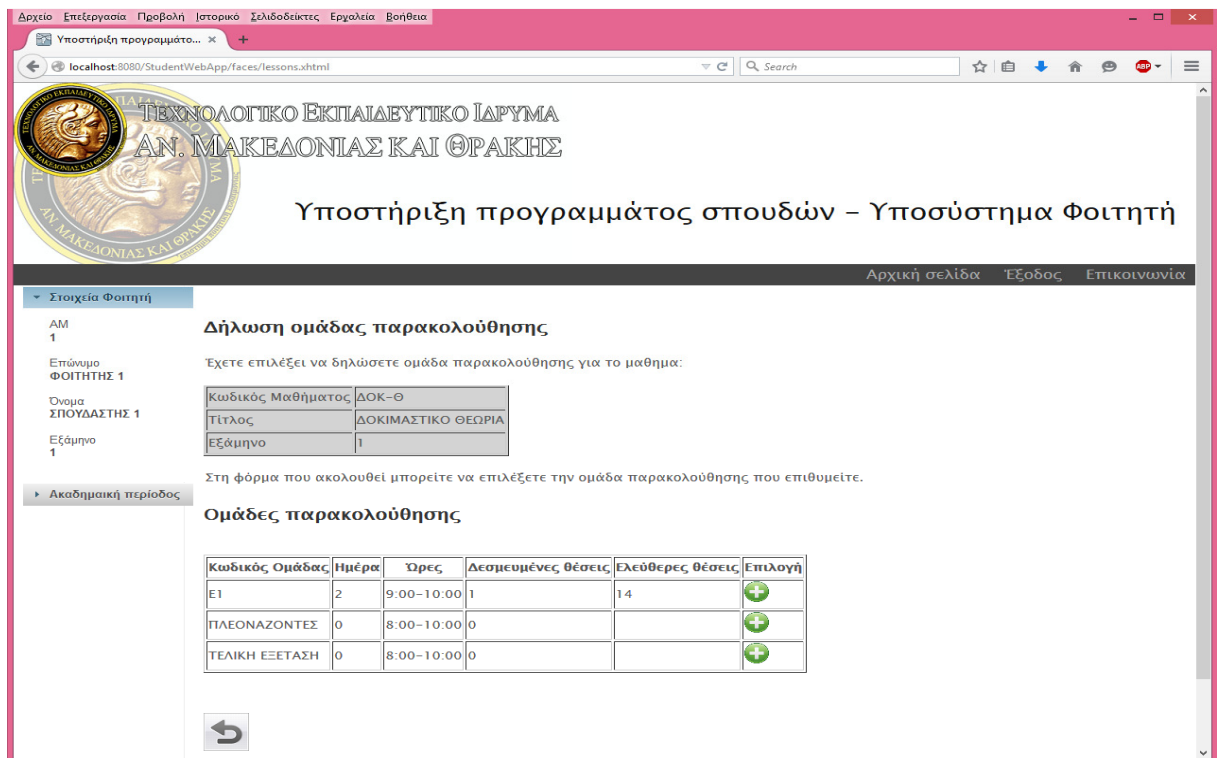
Μετά από επιτυχή είσοδο εμφανίζεται η σελίδα με τα μαθήματα του φοιτητή (εικόνα 3.3).



Εικόνα 3.3: Σελίδα μαθημάτων φοιτητή

Στο 1^ο πινακάκι εμφανίζονται τα μαθήματα που έχει δηλώσει στο σύστημα ηλεκτρονικής γραμματείας, αλλά δεν έχει επιλέξει ομάδα παρακολούθησης. Στο 2^ο πινακάκι εμφανίζονται τα μαθήματα που έχει δηλώσει στο σύστημα ηλεκτρονικής γραμματείας και έχει επιλέξει ομάδα

παρακολούθησης. Όταν ο χρήστης επιλέξει  για να δηλώσει ομάδα παρακολούθησης σε κάποιο μάθημα εμφανίζεται η παρακάτω σελίδα (Εικόνα 3.4).



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΑΝ. ΜΑΚΕΔΟΝΙΑΣ ΚΑΙ ΘΡΑΚΗΣ

Υποστήριξη προγραμμάτος σπουδών - Υποσύστημα Φοιτητή

Αρχική σελίδα Έξοδος Επικοινωνία

Στοιχεία Φοιτητή

ΑΜ
1

Επώνυμο
ΦΟΙΤΗΤΗΣ 1

Όνομα
ΣΠΟΥΔΑΣΤΗΣ 1

Εξάμηνο
1

Ακαδημαϊκή περίοδος




Δήλωση ομάδας παρακολούθησης

Έχετε επιλέξει να δηλώσετε ομάδα παρακολούθησης για το μάθημα:

Κωδικός Μαθήματος	ΔΟΚ-Θ
Τίτλος	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ
Εξάμηνο	1

Στη φόρμα που ακολουθεί μπορείτε να επιλέξετε την ομάδα παρακολούθησης που επιθυμείτε.

Ομάδες παρακολούθησης

Κωδικός Ομάδας	Ημέρα	Ώρες	Δασμευμένες θέσεις	Ελεύθερες θέσεις	Επιλογή
E1	2	9:00-10:00	1	14	
ΠΛΕΟΝΑΖΟΝΤΕΣ	0	8:00-10:00	0		
ΤΕΛΙΚΗ ΕΞΕΤΑΣΗ	0	8:00-10:00	0		

Εικόνα 3.4: Σελίδα δήλωσης ομάδας παρακολούθησης

Αν ο χρήστης επιλέξει την ομάδα E1 η δήλωση θα αποθηκευτεί στη βάση δεδομένων και θα εμφανιστεί η σελίδα μαθημάτων (Εικόνα 3.5).

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΑΝ. ΜΑΚΕΔΟΝΙΑΣ ΚΑΙ ΘΡΑΚΗΣ

Υποστήριξη προγράμματος σπουδών - Υποσύστημα Φοιτητή

Αρχική σελίδα Έξοδος Ωρολόγιο πρόγραμμα Επικοινωνία

Στοιχεία Φοιτητή

AM
1

Επώνυμο
ΦΟΙΤΗΤΗΣ 1

Όνομα
ΣΠΟΥΔΑΣΤΗΣ 1

Εξάμηνο
1

Ακαδημαϊκή περίοδος

Βρίσκεστε σε περίοδο δηλώσεων ομάδων παρακολούθησης

Μαθήματα

Στους παρακάτω πίνακες εμφανίζονται μαθήματα που έχει δηλώσει ο φοιτητής στο σύστημα της Ηλεκτρονικής Γραμματείας. Στον πρώτο πίνακα εμφανίζονται μαθήματα στα οποία δεν έχετε επιλέξει μέχρι στιγμής ομάδα παρακολούθησης. Στον δεύτερο πίνακα εμφανίζονται μαθήματα στα οποία έχετε επιλέξει ομάδα παρακολούθησης και τότε

Μαθήματα με αδήλωτη ομάδα παρακολούθησης

Κωδικός Μαθήματος	Τίτλος	Εξάμηνο	Επιλογή
ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ	1	

Μαθήματα με δηλωμένη ομάδα παρακολούθησης

Κωδικός Μαθήματος	Τίτλος	Εξάμηνο	Κωδικός Ομάδας	Ημέρα	Ώρες	Διαγραφή
ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ	1	E1	ΤΡΙΤΗ	9-10	✖

Εικόνα 3.5: Σελίδα μαθημάτων φοιτητή

Στις παρακάτω εικόνες (3.6, 3.7, 3.8, 3.9) εμφανίζονται οι προβολές της σελίδας ωρολογίου προγράμματος

Αρχική σελίδα Μαθήματα Έξοδος Επικοινωνία

Ωρολόγιο πρόγραμμα

Στην σελίδα αυτή μπορείτε να δείτε το ωρολόγιο πρόγραμμα του τμήματος.

Συγκεντρωτικό Καθηγητές Αίθουσες Εξάμηνα

Προβολή ανά μάθημα, ομάδα μαθήματος, καθηγητή και αίθουσα διδασκαλίας.

	ΔΕΥΤΕΡΑ	ΤΡΙΤΗ	ΤΕΤΑΡΤΗ	ΠΕΜΠΤΗ	ΠΑΡΑΣΚΕΥΗ
08:00					
09:00				ΔΟΚ-Θ ΟΜΑΔΑ 1 - ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Β - Α2	
10:00	ΔΟΚ-Ε ΟΜΑΔΑ 1 - ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Α - Α1			ΔΟΚ-Θ ΟΜΑΔΑ 2 - ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Β - Α2	
11:00	ΔΟΚ-Ε ΟΜΑΔΑ 2 - ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Α - Α1			ΔΟΚ-Θ ΟΜΑΔΑ 3 - ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Β - Α2	
12:00					
13:00					
14:00					
15:00					
16:00					
17:00					
18:00					
19:00					
20:00					
21:00					

Εικόνα 3.6: Συγκεντρωτική προβολή

Αρχείο Επεξεργασία Προβολή Ιστορικό Σελιδοδείκτες Εργαλεία Βοήθεια

http://195.130.92.231:8080/StudentWebApp/faces/timesheet.xhtml

ΑΝ. ΜΑΚΕΔΟΝΙΑΣ ΚΑΙ ΘΡΑΚΗΣ

Αρχική σελίδα Μαθήματα Έξοδος Επικοινωνία

Ωρολόγιο πρόγραμμα

Στην σελίδα αυτή μπορείτε να δείτε το ωρολόγιο πρόγραμμα του τμήματος.

Συγκεντρωτικό Καθηγητές Αίθουσες Εξάμηνα

Προβολή ανά καθηγητή, μάθημα και ομάδα μαθήματος.

ΠΡΟΙΣΤΑΜΕΝΟΣ ΜΠ ΠΡΟΙΣΤΑΜΕΝΟΣ ΜΠ ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Α ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Β

-	ΔΕΥΤΕΡΑ	ΤΡΙΤΗ	ΤΕΤΑΡΤΗ	ΠΕΜΠΤΗ	ΠΑΡΑΣΚΕΥΗ
08:00					
09:00				ΔΟΚ-Θ ΟΜΑΔΑ 1 - ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Β - Α2	
10:00				ΔΟΚ-Θ ΟΜΑΔΑ 2 - ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Β - Α2	
11:00				ΔΟΚ-Θ ΟΜΑΔΑ 3 - ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Β - Α2	
12:00					
13:00					
14:00					
15:00					
16:00					
17:00					
18:00					
19:00					
20:00					
21:00					

Εικόνα 3.7: Προβολή καθηγητών

Αρχείο Επεξεργασία Προβολή Ιστορικό Σελιδοδείκτες Εργαλεία Βοήθεια

http://195.130.92.231:8080/StudentWebApp/faces/timesheet.xhtml

ΑΝ. ΜΑΚΕΔΟΝΙΑΣ ΚΑΙ ΘΡΑΚΗΣ

Αρχική σελίδα Μαθήματα Έξοδος Επικοινωνία

Ωρολόγιο πρόγραμμα

Στην σελίδα αυτή μπορείτε να δείτε το ωρολόγιο πρόγραμμα του τμήματος.

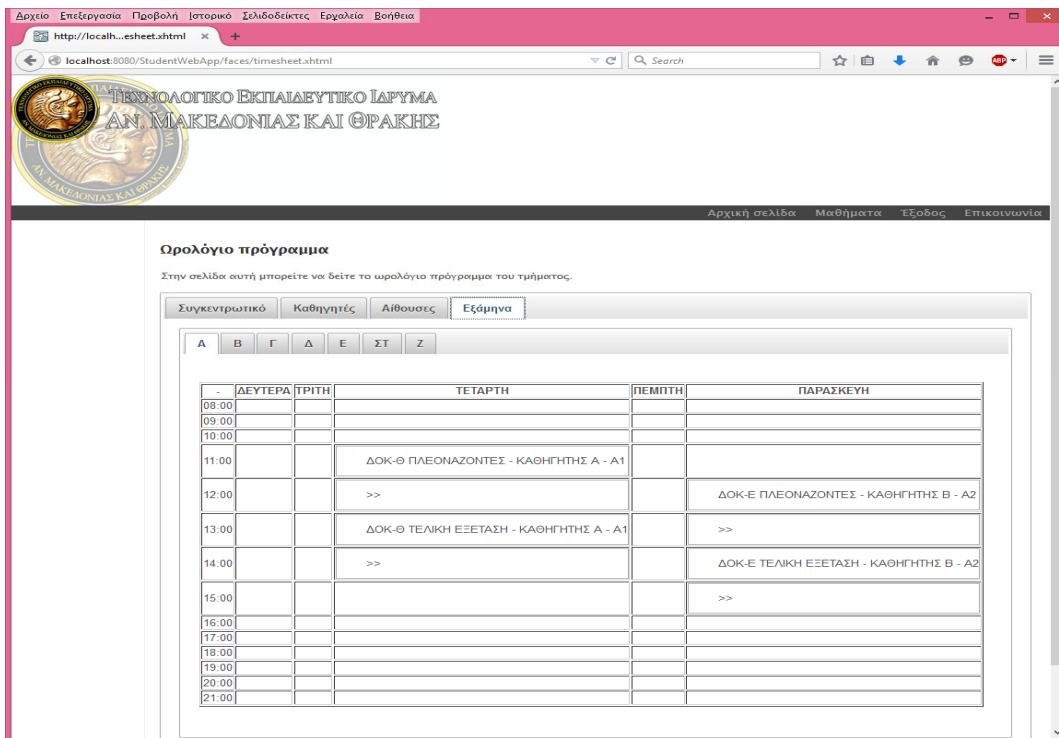
Συγκεντρωτικό Καθηγητές Αίθουσες Εξάμηνα

Προβολή ανά αίθουσα, μάθημα, ομάδα μαθήματος.

A1 A2

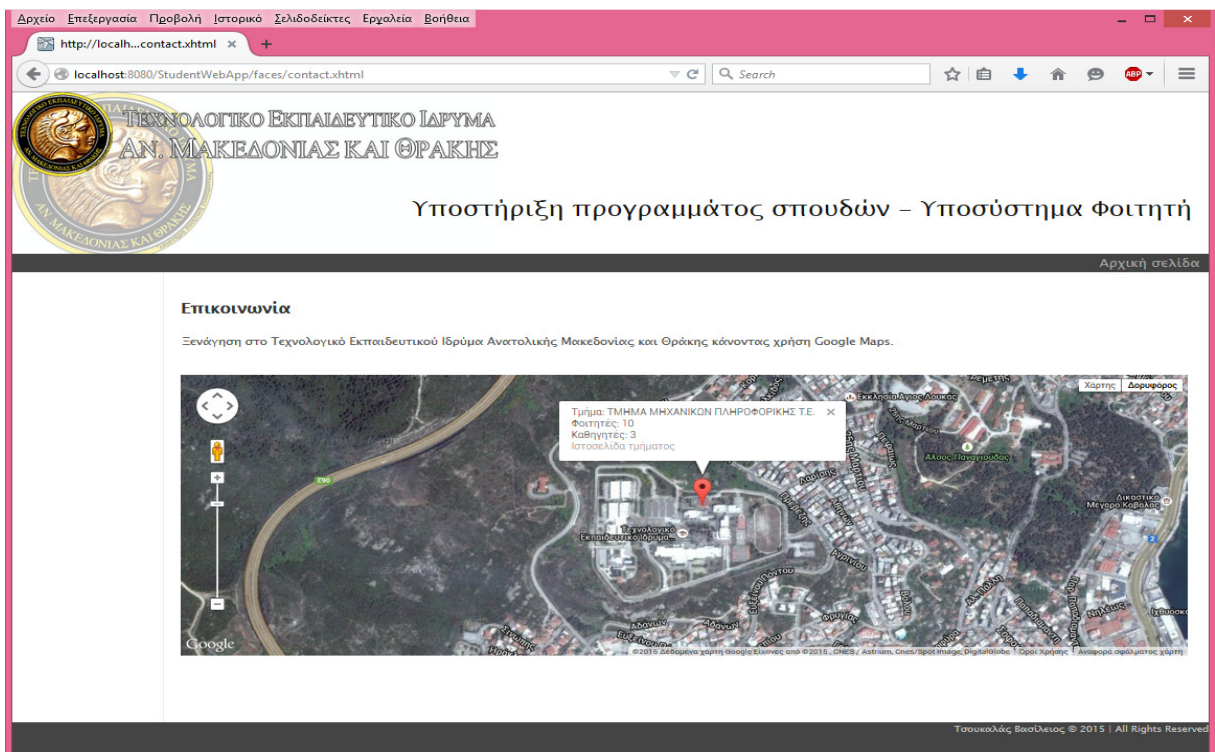
-	ΔΕΥΤΕΡΑ	ΤΡΙΤΗ	ΤΕΤΑΡΤΗ	ΠΕΜΠΤΗ	ΠΑΡΑΣΚΕΥΗ
08:00					
09:00					
10:00				ΔΟΚ-Ε ΟΜΑΔΑ 1 - ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Α - Α1	
11:00				ΔΟΚ-Ε ΟΜΑΔΑ 2 - ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Α - Α1	
12:00					
13:00					
14:00					
15:00					
16:00					
17:00					
18:00					
19:00					
20:00					
21:00					

Εικόνα 3.8: Προβολή αιθουσών



Εικόνα 3.9: Προβολή ανά εξάμηνο

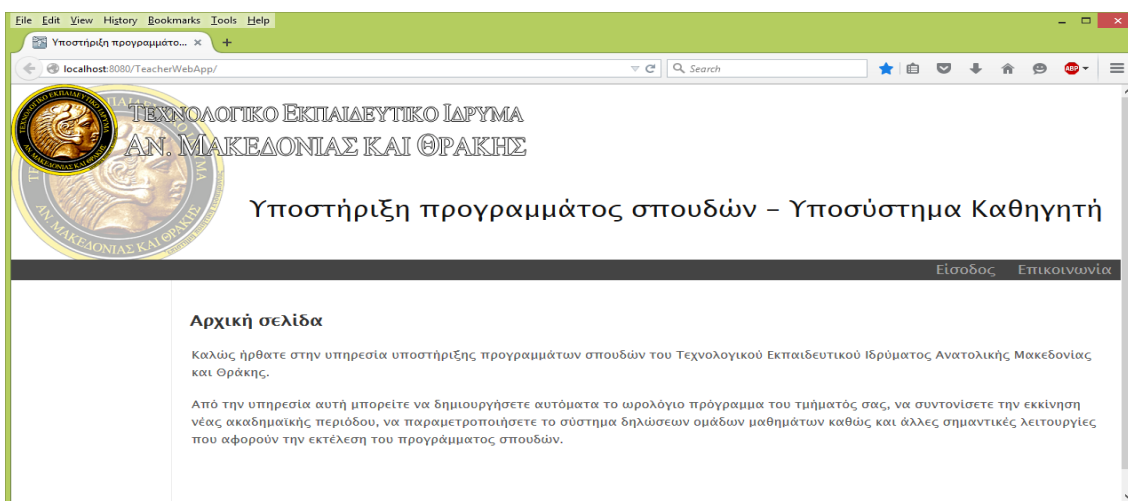
Στη σελίδα επικοινωνία (Εικόνα 3.10) φαίνεται η τοποθεσία του τμήματος ενός φοιτητή που έχει συνδεθεί στο υποσύστημα. Όταν ο φοιτητής επιλέξει την τοποθεσία εμφανίζονται το όνομα και ο ιστότοπος του τμήματός του, καθώς και τα πλήθη των φοιτητών-καθηγητών.



Εικόνα 3.10: Σελίδα επικοινωνίας

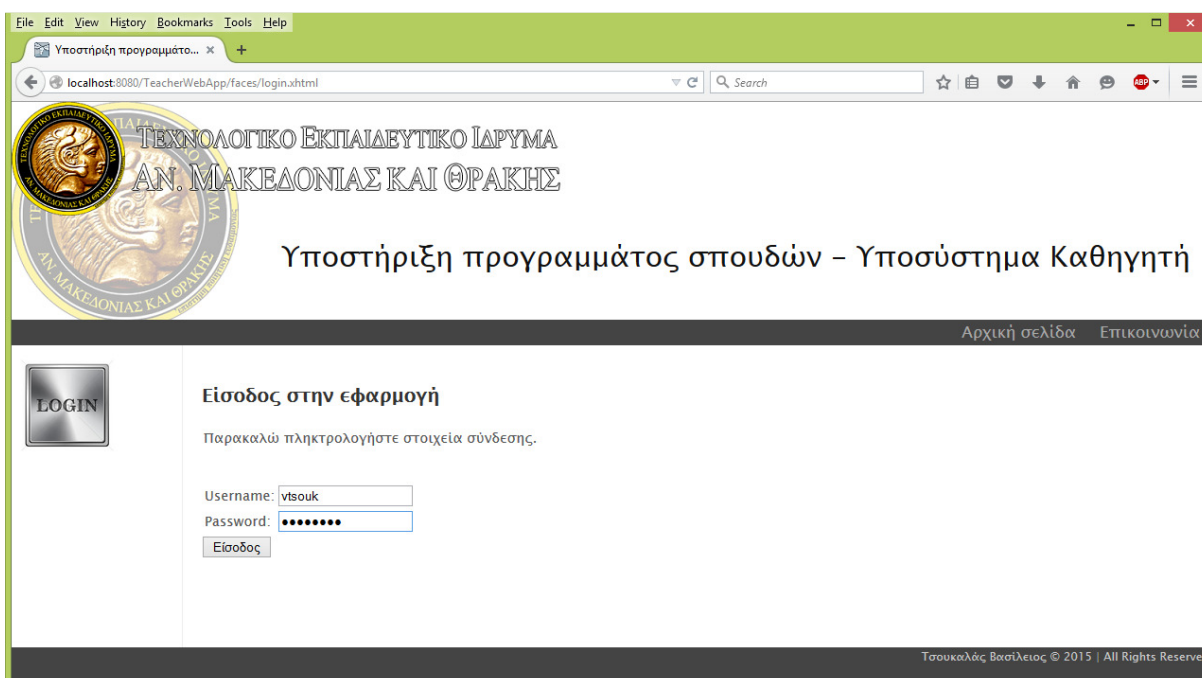
3.1.2 Διεπαφή υποσυστήματος εκπαιδευτικών

Η αρχική σελίδα εμφανίζεται στην εικόνα 3.11. Ο εκπαιδευτικός επιλέγει είσοδο από το μενού.



Εικόνα 3.11: Αρχική σελίδα

Ο εκπαιδευτικός και ο διαχειριστής μπορούν να συνδεθούν στην παρακάτω σελίδα (Εικόνα 3.12).



Εικόνα 3.12: Σελίδα σύνδεσης

Με την επιλογή *Τμήματα* από το μενού προβάλλονται στο διαχειριστή τα τμήματα του εκπαιδευτικού ιδρύματος. Ο διαχειριστής μπορεί να επεξεργάζεται τα τμήματα όπως φαίνεται στην εικόνα 3.13:

Υποστήριξη προγράμματος σπουδών - Υποσύστημα Καθηγητή

Αρχική σελίδα Τμήματα Καθηγητές Μαθήματα Ακαδημαϊκές Περιόδους Ωρολόγιο πρόγραμμα Έξοδος Επικοινωνία

▼ Στοιχεία καθηγητή

Επώνυμο
ΤΣΟΥΚΑΛΑΣ

Όνομα
ΒΑΣΙΛΕΙΟΣ

e-mail
vtsouk@teiemt.gr

username
vtsouk

Τμήμα
0

Διαχειριστής

Τμήματα Εκπαιδευτικού Ιδρύματος

Στον παρακάτω πίνακα μπορείτε να επεξεργαστείτε τα τμήματα του ιδρύματος

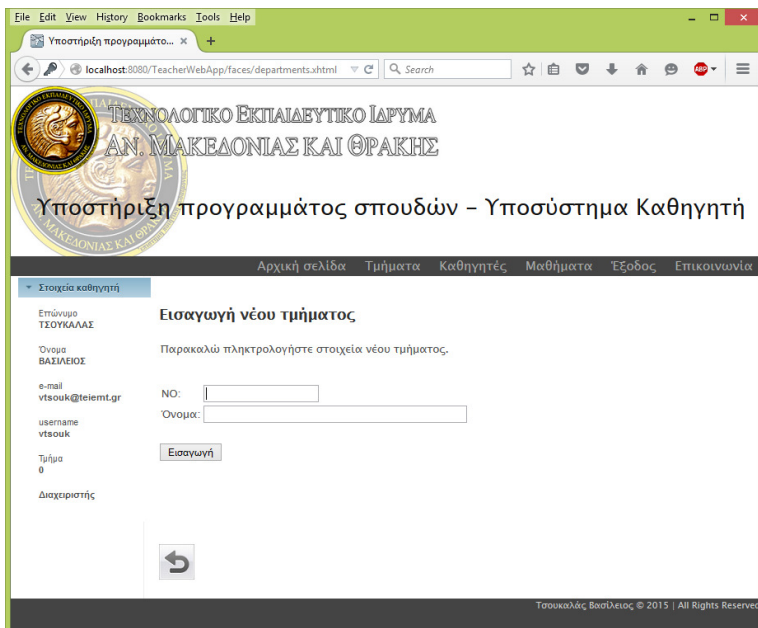
Προβολή τμημάτων

Κωδικός Τμήματος	Όνομα	Επεξεργασία	Διαγραφή
0	ΚΕΝΤΡΟ ΔΙΑΧΕΙΡΙΣΗΣ ΔΙΚΤΥΟΥ		
326	ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.		
327	ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.		
328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.		
329	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΤΕΧΝΟΛΟΓΙΑΣ ΠΕΤΡΕΛΑΙΟΥ ΚΑΙ ΦΥΣΙΚΟΥ ΑΕΡΙΟΥ Τ.Ε.		
330	ΤΜΗΜΑ ΔΑΣΟΠΟΝΙΑΣ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗΣ ΦΥΣΙΚΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ		
331	ΤΜΗΜΑ ΛΟΓΙΣΤΙΚΗΣ ΚΑΙ ΧΡΗΜΑΤΟΟΙΚΟΝΟΜΙΚΗΣ		
332	ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ		
411	ΤΜΗΜΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΤΟΠΙΟΥ		
471	ΤΜΗΜΑ ΟΙΝΟΛΟΓΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΠΟΤΩΝ		
550	ΤΜΗΜΑ ΝΟΣΗΛΕΥΤΙΚΗΣ		

Εικόνα 3.13: Σελίδα τμημάτων

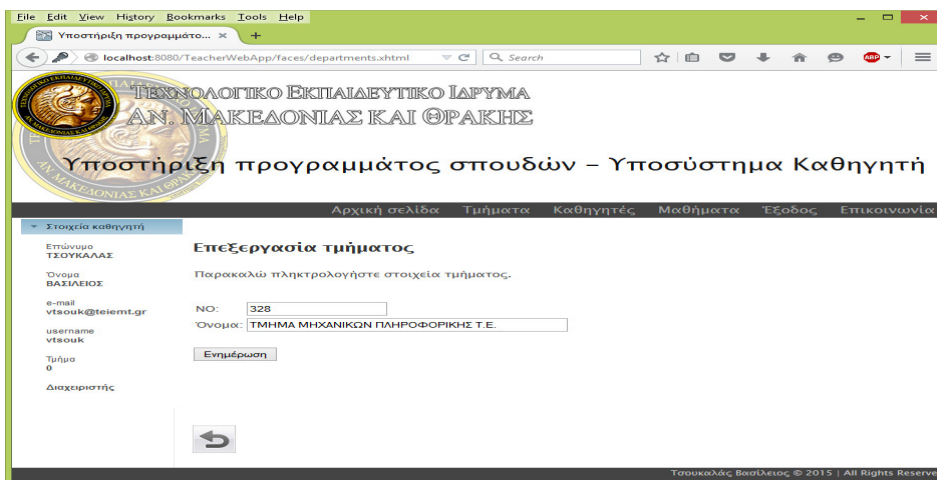
Αριστερά της εικόνας υπάρχουν τα στοιχεία του χρήστη-διαχειριστή που έχει συνδεθεί στο υποσύστημα εκπαιδευτικού. Επιλέγοντας μπορεί να γίνει προσθήκη νέου τμήματος στο εκπαιδευτικό ίδρυμα. Στον πίνακα προβάλλονται τα τμήματα του ιδρύματος ενώ δίπλα σε κάθε τμήμα υπάρχει η δυνατότητα διαγραφής και επεξεργασίας των στοιχείων του.

Στην εικόνα 3.14 φαίνεται η σελίδα προσθήκης νέου τμήματος.



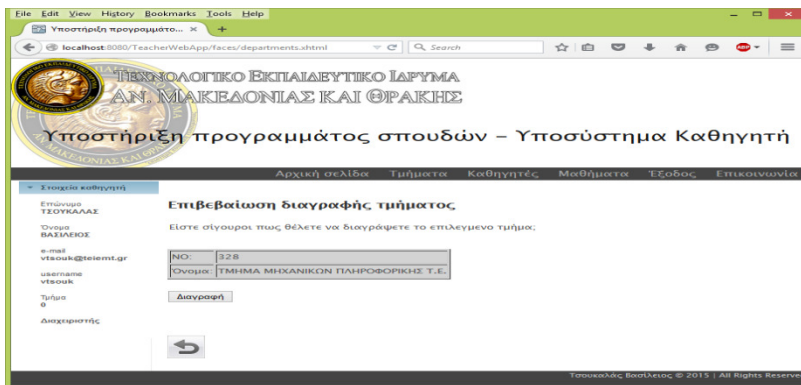
Εικόνα 3.14: Σελίδα εισαγωγής νέου τμήματος

Στην εικόνα 3.15 φαίνεται η σελίδα επεξεργασίας του τμήματος ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε..



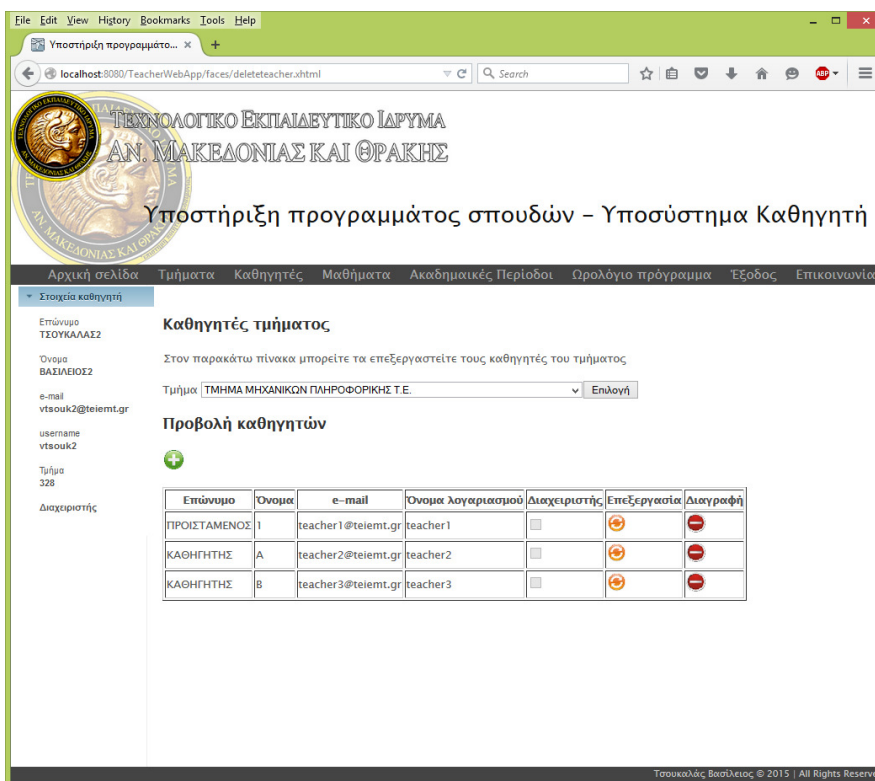
Εικόνα 3.15: Σελίδα επεξεργασίας τμήματος

Στην εικόνα 3.16 εμφανίζεται η σελίδα διαγραφής τμήματος.



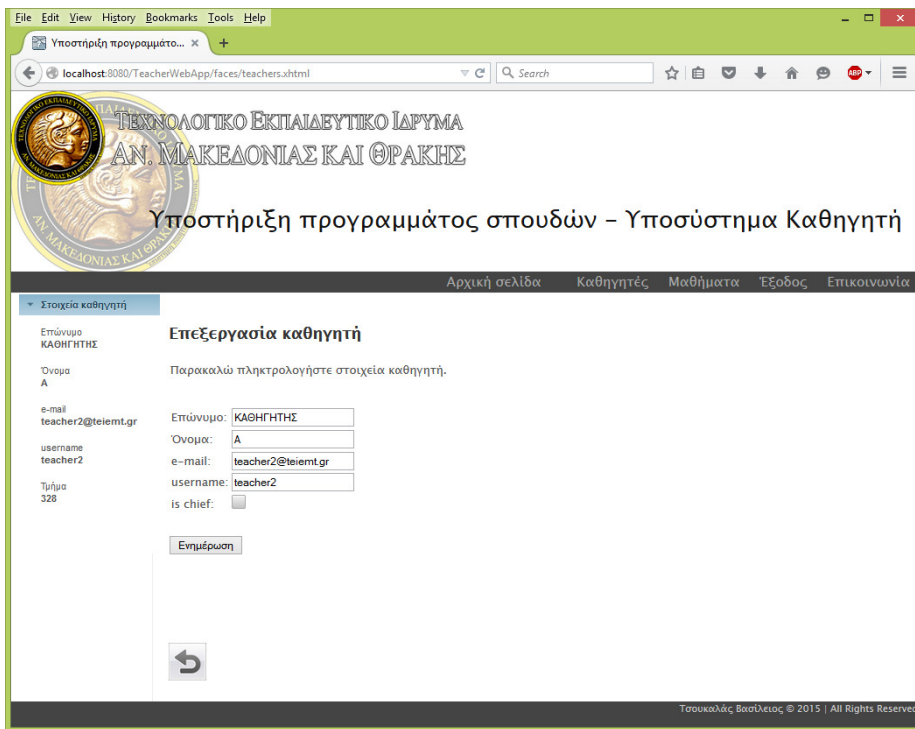
Εικόνα 3.16: Σελίδα διαγραφής τμήματος

Με την επιλογή *Καθηγητές* από το μενού προβάλλονται οι εκπαιδευτικοί του ιδρύματος όπως φαίνεται στην εικόνα 3.17.



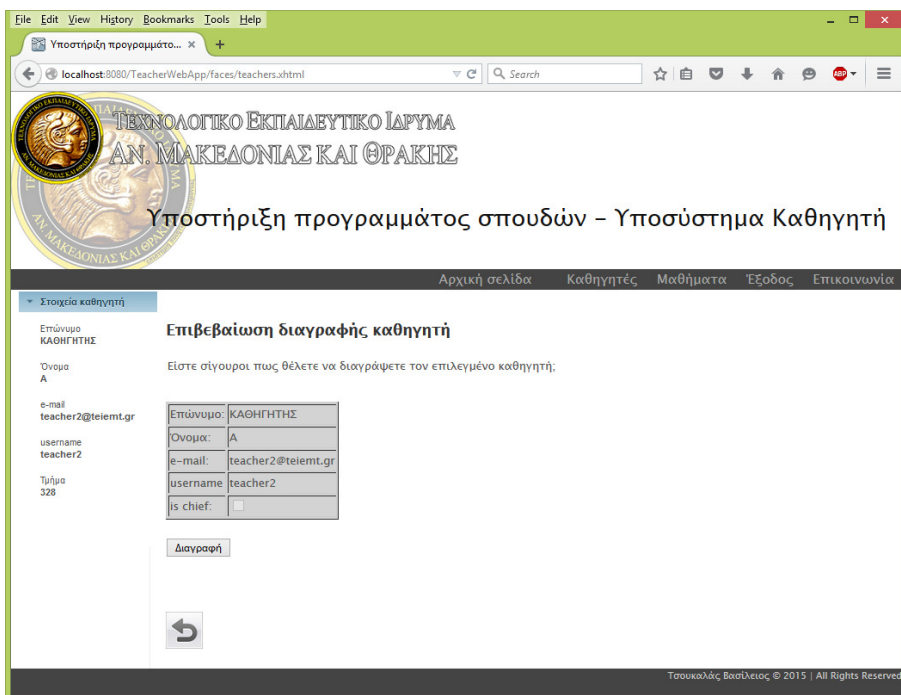
Εικόνα 3.17: Σελίδα εκπαιδευτικών

Επιλέγοντας επεξεργασία δίπλα από κάποιο εκπαιδευτικό μπορεί να γίνει επεξεργασία των στοιχείων του, όπως φαίνεται παρακάτω (Εικόνα 3.18).



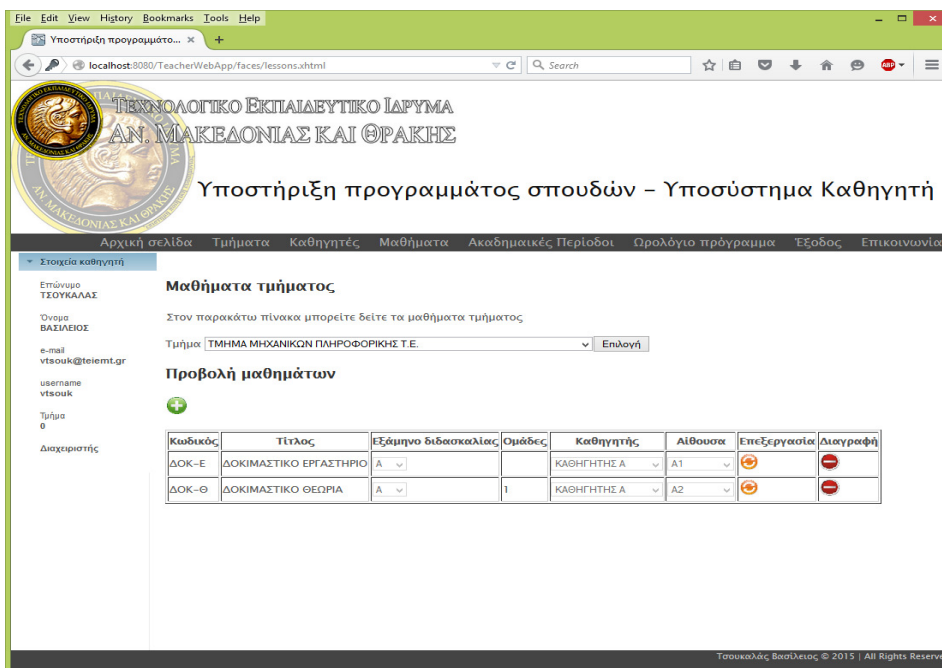
Εικόνα 3.18: Σελίδα επεξεργασίας στοιχείων εκπαιδευτικού

Επιλέγοντας διαγραφή δίπλα από κάποιο εκπαιδευτικό μπορούμε να τον διαγράψουμε, όπως φαίνεται παρακάτω (Εικόνα 3.19).



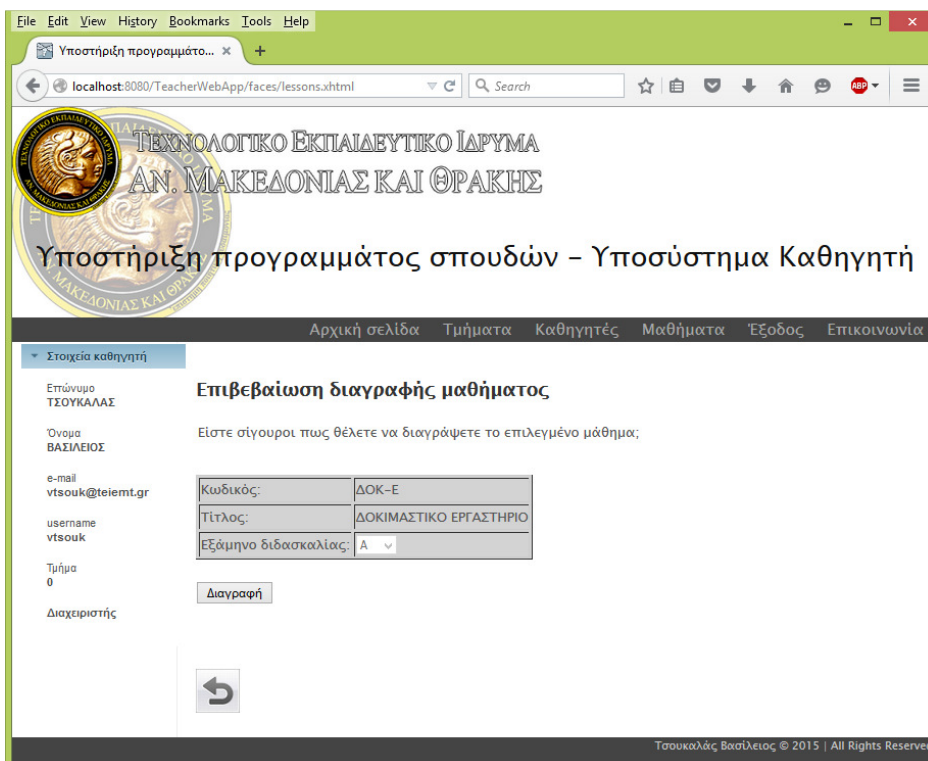
Εικόνα 3.19: Σελίδα διαγραφής εκπαιδευτικού

Με την επιλογή *Μαθήματα* από το μενού προβάλλονται τα μαθήματα ανά τμήμα του ιδρύματος όπως φαίνεται στην εικόνα 3.20.



Εικόνα 3.20: Σελίδα μαθημάτων τμήματος

Επιλέγοντας διαγραφή δίπλα από κάποιο μάθημα μπορούμε να το διαγράψουμε, όπως φαίνεται παρακάτω (Εικόνα 3.21).



Εικόνα 3.21: Σελίδα διαγραφής μαθήματος

Επιλέγοντας επεξεργασία δίπλα από κάποιο μάθημα μπορεί να γίνει επεξεργασία των στοιχείων του, προβολή των ομάδων παρακολούθησης, να δούμε το ιστορικό επιτυχιών ανά ακαδημαϊκή περίοδο, καθώς και ποιοι φοιτητές έχουν δηλώσει το μάθημα στο εξωτερικό σύστημα της ηλεκτρονικής γραμματείας την τελευταία ακαδημαϊκή περίοδο. (Εικόνα 3.22)

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΑΝ. ΜΑΚΕΔΟΝΙΑΣ ΚΑΙ ΘΡΑΚΗΣ**

Υποστήριξη προγραμμάτων σπουδών – Υποσύστημα Καθηγητή

Αρχική σελίδα Τμήματα Καθηγητές Μαθήματα Έξοδος Επικοινωνία

▼ Στοιχεία καθηγητή

Επώνυμο: ΤΣΟΥΚΑΛΑΣ
Όνομα: ΒΑΣΙΛΕΙΟΣ
e-mail: vtsouk@teiemt.gr
username: vtsouk
Τμήμα: 0
Διαχειριστής

Επεξεργασία Μαθήματος

Παρακαλώ πληκτρολογήστε στοιχεία μαθήματος.

Κωδικός: ΔΟΚΕ
Τίτλος: ΔΟΚΙΜΑΣΤΙΚΟ ΕΡΓΑΣΤΗΡΙΟ
Εξάμηνο διδασκαλίας: Α
Καθηγητής: ΚΑΘΗΓΗΤΗΣ Α
Αίθουσα: Α1

Ενμέριωση

Προβολή ομάδων

ACAD_PERIOD_ID	Κωδικός Ομάδας	Ημέρες	Ώρες	Χωρητικότητα	Δεσμευμένες θέσεις	Ελεύθερες θέσεις	Επεξεργασία	Διαγραφή
32820142	ΠΛΕΟΝΑΖΟΝΤΕΣ	--	999	0	0	999	☺	☹
32820142	ΤΕΛΙΚΗ ΕΞΕΤΑΣΗ	--	999	0	0	999	☺	☹

Ιστορικό επιτυχιών ανά ακαδημαϊκή περίοδο

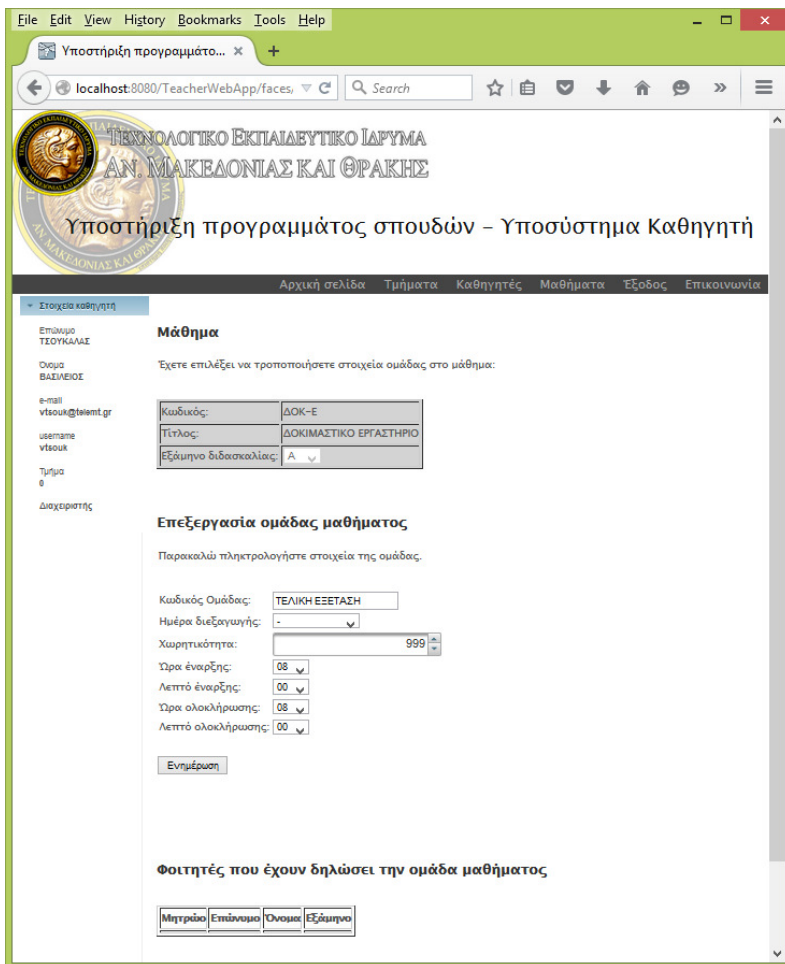
Κωδικός	Κωδικός τμήματος	Ακαδημαϊκό Έτος	Περίοδος	Επιτυχίες	Αποτυχίες	Μη παρακολουθήσαντες
32820142	328	2014	2	0	0	6
32820141	328	2014	1	0	0	0
32820132	328	2013	2	0	0	0
32820131	328	2013	1	0	0	0

Φοιτητές που έχουν δηλώσει το μάθημα στην Ηλεκτρονική Γραμματεία

Μητρώο	Επώνυμο	Όνομα	Εξάμηνο
10	ΦΟΙΤΗΤΗΣ 10	ΣΠΟΥΔΑΣΤΗΣ 10	A
5	ΦΟΙΤΗΤΗΣ 5	ΣΠΟΥΔΑΣΤΗΣ 5	A
6	ΦΟΙΤΗΤΗΣ 6	ΣΠΟΥΔΑΣΤΗΣ 6	A
7	ΦΟΙΤΗΤΗΣ 7	ΣΠΟΥΔΑΣΤΗΣ 7	A
8	ΦΟΙΤΗΤΗΣ 8	ΣΠΟΥΔΑΣΤΗΣ 8	A
9	ΦΟΙΤΗΤΗΣ 9	ΣΠΟΥΔΑΣΤΗΣ 9	A

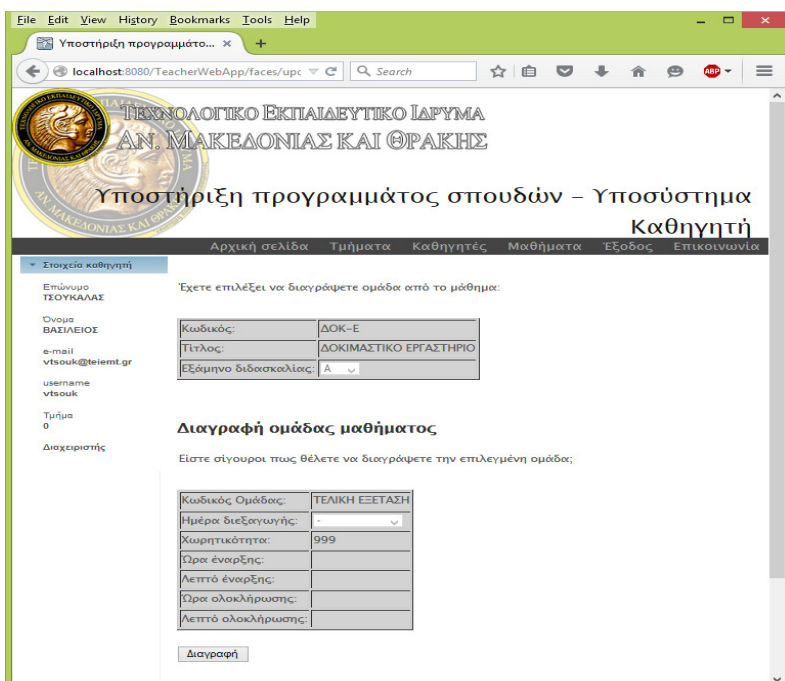
Εικόνα 3.22: Σελίδα επεξεργασίας μαθήματος, προβολής ομάδων, ιστορικού επιτυχιών και φοιτητές μαθήματος

Στον πίνακα προβολής ομάδων παρακολούθησης δίπλα από κάθε ομάδα έχουμε τη δυνατότητα επεξεργασίας ή διαγραφής της. Στην εικόνα 3.23 φαίνεται η σελίδα επεξεργασίας μιας ομάδας. Προβάλλεται το μάθημα στο οποίο ανήκει η ομάδα, μπορούμε να επεξεργαστούμε τα στοιχεία της ομάδας και να δούμε ποιοι φοιτητές έχουν δηλώσει την ομάδα παρακολούθησης.





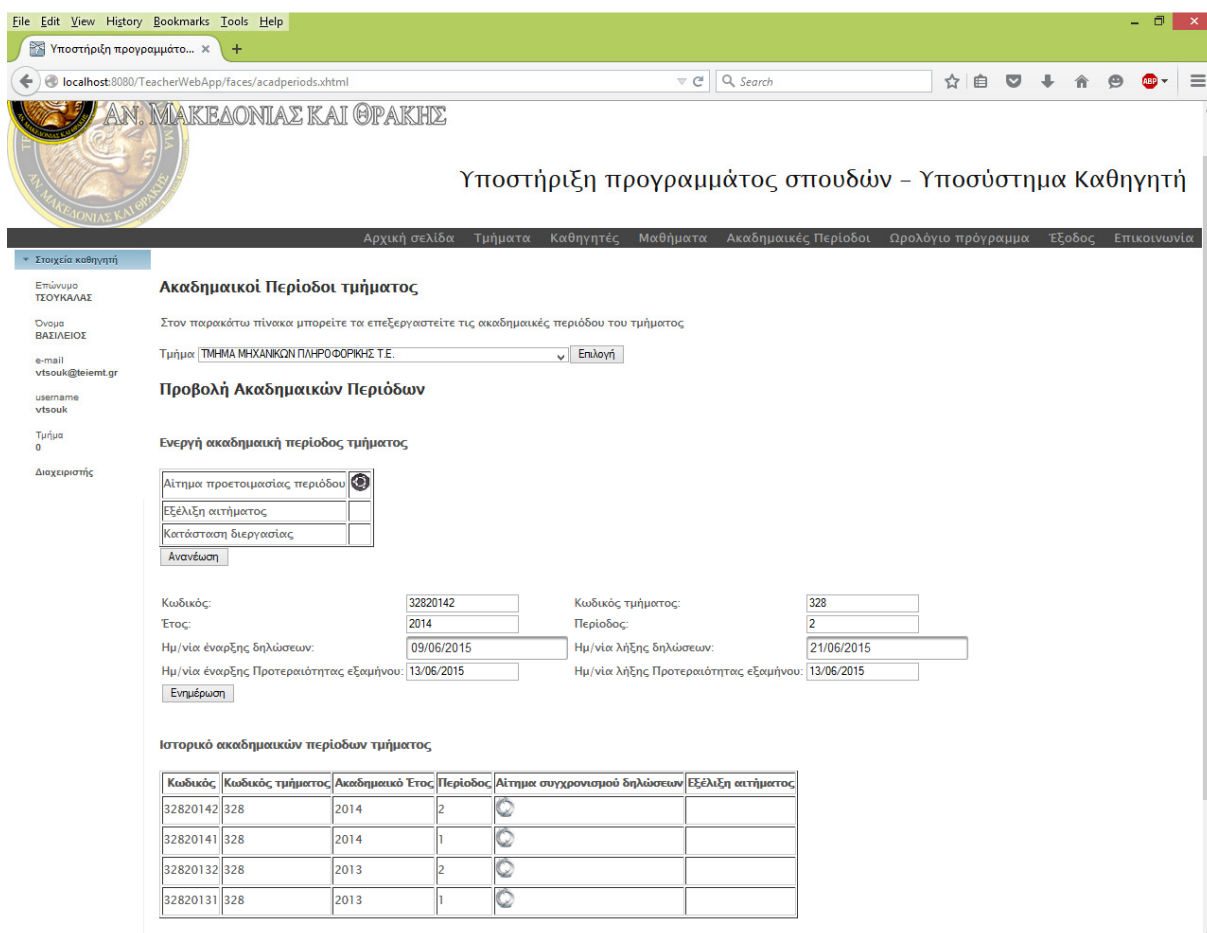
Εικόνα 3.23: Σελίδα επεξεργασίας ομάδας μαθήματος

Επιλέγοντας διαγραφή δίπλα από κάποια ομάδα μπορούμε να τη διαγράψουμε, όπως φαίνεται παρακάτω. (Εικόνα 3.24)







Εικόνα 3.24: Σελίδα διαγραφής ομάδας μαθήματος

Στην επιλογή *Ακαδημαϊκές Περιόδους* από το μενού, ο προϊστάμενος ενός τμήματος μπορεί ζητήσει την προετοιμασία της ακαδημαϊκής περιόδου (εκκινώντας το BPM Process *NewAcadPeriod*, επιλέγοντας το εικονίδιο ), να ρυθμίσει τις παραμέτρους μιας ακαδημαϊκής περιόδου και να ζητήσει συγχρονισμό δηλώσεων για παλαιότερες ακαδημαϊκές περιόδους (εκκινώντας το BPM Process *SyncRegistrationsAcadPeriod*, επιλέγοντας το εικονίδιο  δίπλα από κάποια περίοδο). (Εικόνα 3.25)



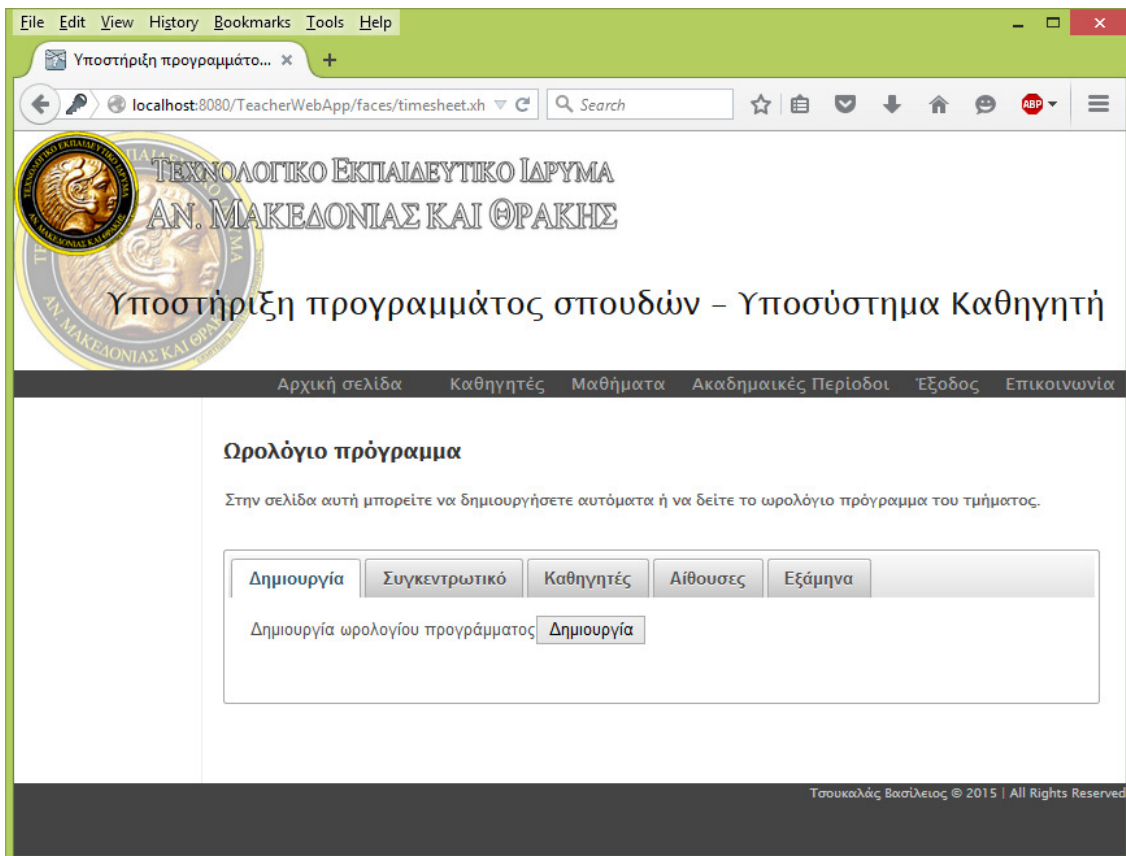
The screenshot shows a web browser window displaying the 'Υποστήριξη προγράμματος σπουδών - Υποσύστημα Καθηγητή' (Teacher Support System). The page is titled 'Ακαδημαϊκοί Περίοδοι τμήματος' (Academic Periods of the Department). It features a navigation menu with options like 'Αρχική σελίδα', 'Τμήματα', 'Καθηγητές', 'Μαθήματα', 'Ακαδημαϊκές Περιόδους', 'Ωρολόγιο πρόγραμμα', 'Έξοδος', and 'Επικοινωνία'. The main content area includes a sidebar with user details (Name: ΤΣΟΥΚΑΛΑΣ, Email: vtsouk@teiemt.gr) and a main section with the following elements:

- Ακαδημαϊκοί Περίοδοι τμήματος:** A message indicating that the table below can be used to manage academic periods.
- Προβολή Ακαδημαϊκών Περιόδων:** A dropdown menu for the department (currently 'ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.') and an 'Επιλογή' button.
- Ενεργή ακαδημαϊκή περίοδος τμήματος:** A table with columns for 'Αίτημα προετοιμασίας περιόδου' (gear icon), 'Εξέλιξη αιτήματος', and 'Κατάσταση διεργασίας'. Below the table is an 'Ανανέωση' button.
- Κωδικός:** 32820142, **Κωδικός τμήματος:** 328, **Έτος:** 2014, **Περίοδος:** 2.
- Ημ/νία έναρξης δηλώσεων:** 09/06/2015, **Ημ/νία λήξης δηλώσεων:** 21/06/2015.
- Ημ/νία έναρξης Προτεραιότητας εξαμήνου:** 13/06/2015, **Ημ/νία λήξης Προτεραιότητας εξαμήνου:** 13/06/2015.
- Ενημέρωση:** A button to update the information.
- Ιστορικό ακαδημαϊκών περιόδων τμήματος:** A table with the following data:

Κωδικός	Κωδικός τμήματος	Ακαδημαϊκό Έτος	Περίοδος	Αίτημα συγχρονισμού δηλώσεων	Εξέλιξη αιτήματος
32820142	328	2014	2		
32820141	328	2014	1		
32820132	328	2013	2		
32820131	328	2013	1		

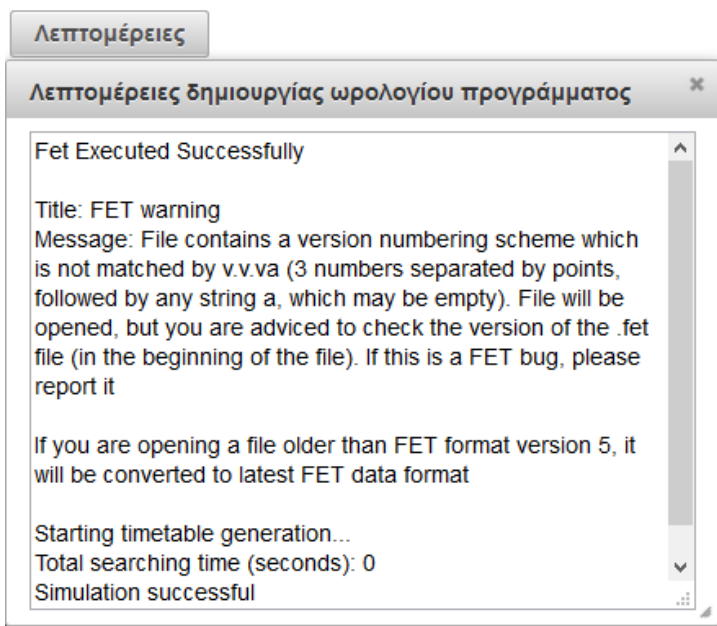
Εικόνα 3.25: Σελίδα ακαδημαϊκών περιόδων

Στις παρακάτω εικόνες εμφανίζονται οι προβολές της σελίδας ωρολογίου προγράμματος. Στην εικόνα 3.26 ο προϊστάμενος δημιουργεί με αυτόματο τρόπο το ωρολόγιο πρόγραμμα της περιόδου επιλέγοντας **Δημιουργία**.



Εικόνα 3.26: Σελίδα δημιουργίας ωρολογίου προγράμματος

Μετα την επιλογή δημιουργίας ο προϊστάμενος μπορεί να δει το αποτέλεσμα της εκτέλεσης του εξωτερικού συστήματος FET, όπως φαίνεται στην εικόνα 3.27.




Εικόνα 3.27: Σελίδα αποτελεσμάτων εκτέλεσης του συστήματος FET

Στις επόμενες εικόνες (3.28, 3.29, 3.30, 3.31) μπορούμε να δούμε τις διάφορες προβολές του ωρολογίου προγράμματος: συγκεντρωτική, ανά καθηγητή, ανά αίθουσα και ανά εξάμηνο.

Δημιουργία **Συγκεντρωτικό** Καθηγητές Αίθουσες Εξάμηνα

Προβολή ανά μάθημα, ομάδα μαθήματος, καθηγητή και αίθουσα διδασκαλίας.

-	ΔΕΥΤΕΡΑ	ΤΡΙΤΗ	ΤΕΤΑΡΤΗ	ΠΕΜΠΤΗ	ΠΑΡΑΣΚΕΥΗ
08:00					ΔΟΚ-Θ ~ Ε1 ~ ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Β ~ Α2
09:00					>>
10:00					>>
11:00					ΔΟΚ-Θ ~ Ε2 ~ ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Β ~ Α2
12:00					>>
13:00					ΔΟΚ-Θ ~ Ε3 ~ ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Β ~ Α2
14:00					>>
15:00					>>
16:00					ΔΟΚ-Θ ~ Ε4 ~ ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Β ~ Α2
17:00					>>
18:00					>>
19:00					
20:00					
21:00					

Συγχρονισμός ομάδων παρακολούθησης 

Εικόνα 3.28: Συγκεντρωτική προβολή ωρολογίου προγράμματος, και συγχρονισμός του παραγόμενου προγράμματος με τις ομάδες παρακολούθησης

Αρχείο Επεξεργασία Προβολή Ιστορικό Σελιδοδείκτες Εργαλεία Βοήθεια

Υποστήριξη προγραμμάτ... 195.130.92.231:8080/TeacherWebApp/faces/timesheet.xhtml

ΑΝ. ΜΑΚΕΔΟΝΙΑΣ ΚΑΙ ΘΡΑΚΗΣ

Υποστήριξη προγραμμάτων σπουδών – Υποσύστημα Καθηγητή

Αρχική σελίδα Καθηγητές Μαθήματα Ακαδημαϊκές Περίοδοι Έξοδος Επικοινωνία

Ωρολόγιο πρόγραμμα

Στην σελίδα αυτή μπορείτε να δημιουργήσετε αυτόματα ή να δείτε το ωρολόγιο πρόγραμμα του τμήματος.

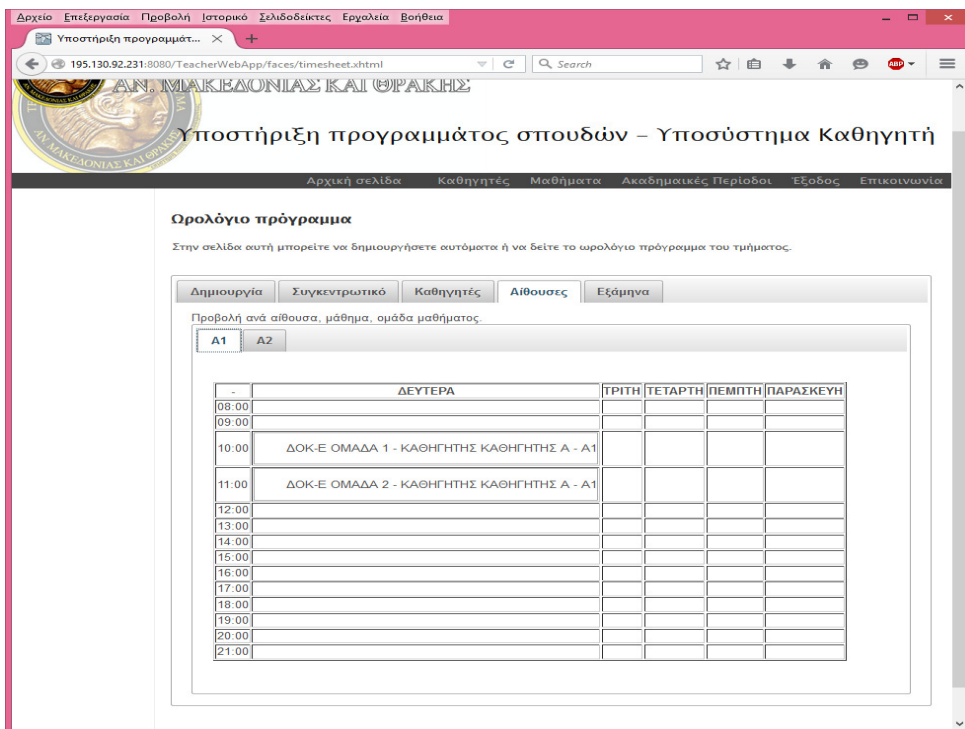
Δημιουργία **Συγκεντρωτικό** Καθηγητές Αίθουσες Εξάμηνα

Προβολή ανά καθηγητή, μάθημα και ομάδα μαθήματος.

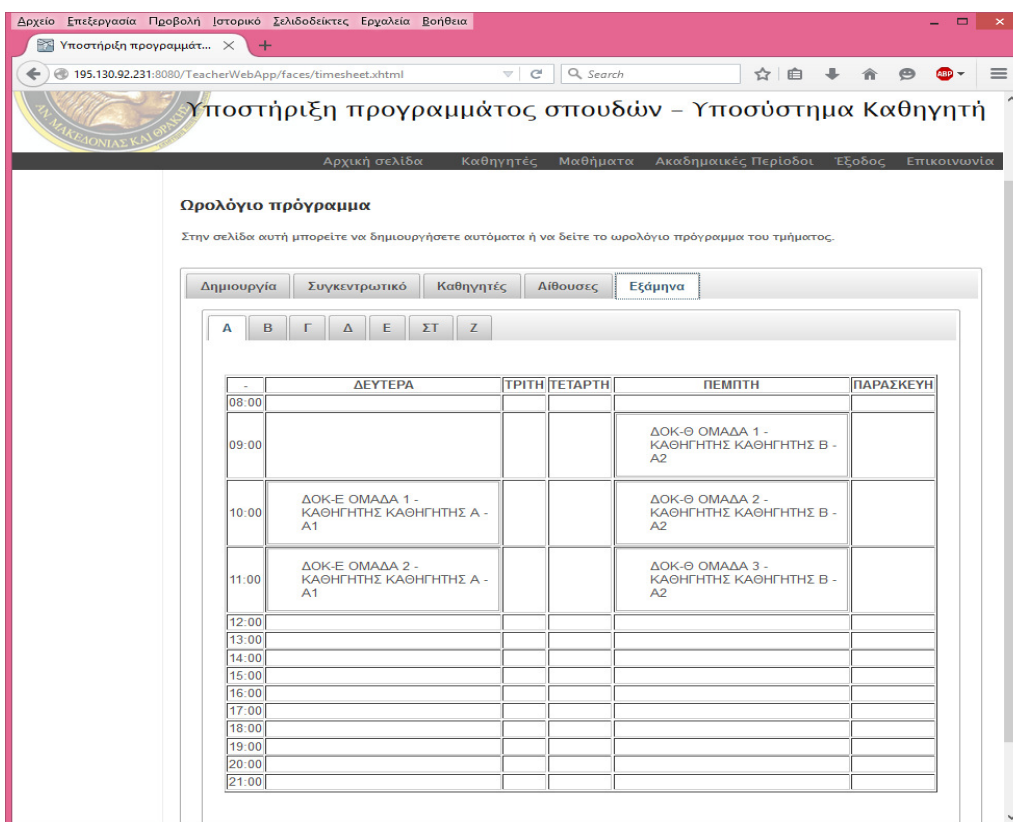
ΠΡΟΙΣΤΑΜΕΝΟΣ ΜΠ ΠΡΟΙΣΤΑΜΕΝΟΣ ΜΠ ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Α ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Β

-	ΔΕΥΤΕΡΑ	ΤΡΙΤΗ	ΤΕΤΑΡΤΗ	ΠΕΜΠΤΗ	ΠΑΡΑΣΚΕΥΗ
08:00					
09:00				ΔΟΚ-Θ ΟΜΑΔΑ 1 - ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Β - Α2	
10:00				ΔΟΚ-Θ ΟΜΑΔΑ 2 - ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Β - Α2	
11:00				ΔΟΚ-Θ ΟΜΑΔΑ 3 - ΚΑΘΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ Β - Α2	
12:00					
13:00					
14:00					
15:00					
16:00					
17:00					
18:00					
19:00					
20:00					
21:00					

Εικόνα 3.29: Προβολή καθηγητών

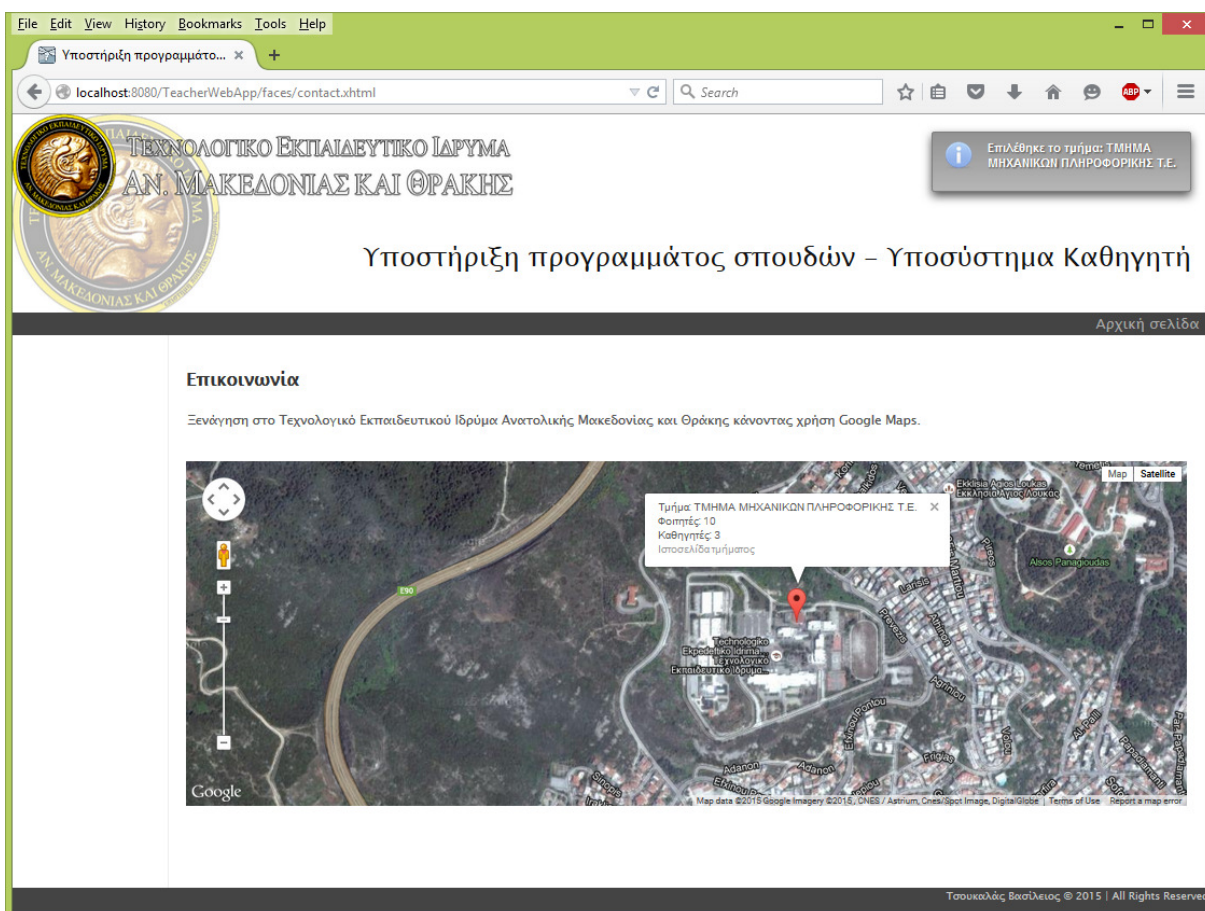


Εικόνα 3.30: Προβολή αιθουσών



Εικόνα 3.31: Προβολή ανά εξάμηνο

Στη σελίδα επικοινωνία (Εικόνα 3.32) φαίνεται η τοποθεσία του τμήματος ενός εκπαιδευτικού που έχει συνδεθεί στο υποσύστημα. Όταν ο εκπαιδευτικός επιλέξει την τοποθεσία εμφανίζονται το όνομα και ο ιστότοπος του τμήματός του, καθώς και τα πλήθη των φοιτητών-καθηγητών.



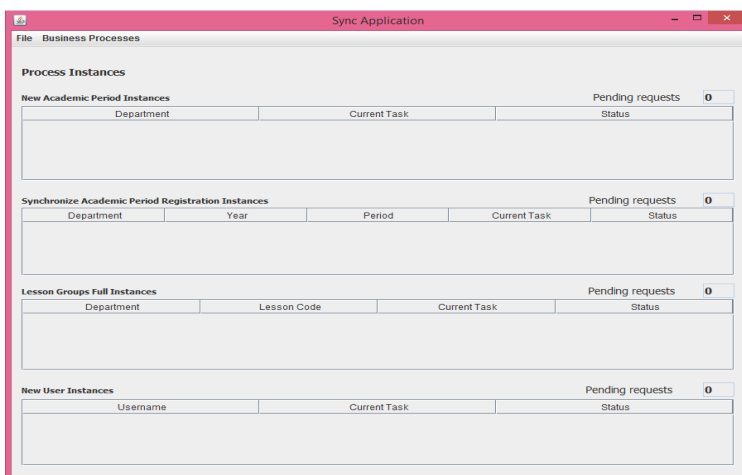
Εικόνα 3.32: Σελίδα επικοινωνίας

3.1.3 Διεπαφή υποσυστήματος ωρολογίου προγράμματος

Το υποσύστημα ωρολογίου προγράμματος δεν έχει φόρμες διεπαφής. Το υποσύστημα αυτό χρησιμοποιείται ως βιβλιοθήκη (ενδιάμεσο-interface) που ενσωματώνεται στα υποσυστήματα φοιτητών και εκπαιδευτικών για την επεξεργασία και την προβολή ωρολογίων προγραμμάτων.

3.1.4 Διεπαφή υποσυστήματος συγχρονισμού

Η κύρια φόρμα JMainFrame του υποσυστήματος συγχρονισμού φαίνεται στην εικόνα 3.33 .



Εικόνα 3.33: Κύρια φόρμα

Στο πίνακάκι *New Academic Period Instances* εμφανίζονται οι διεργασίες νέων ακαδημαϊκών περιόδων που βρίσκονται σε εξέλιξη για οποιοδήποτε τμήμα του ιδρύματος.

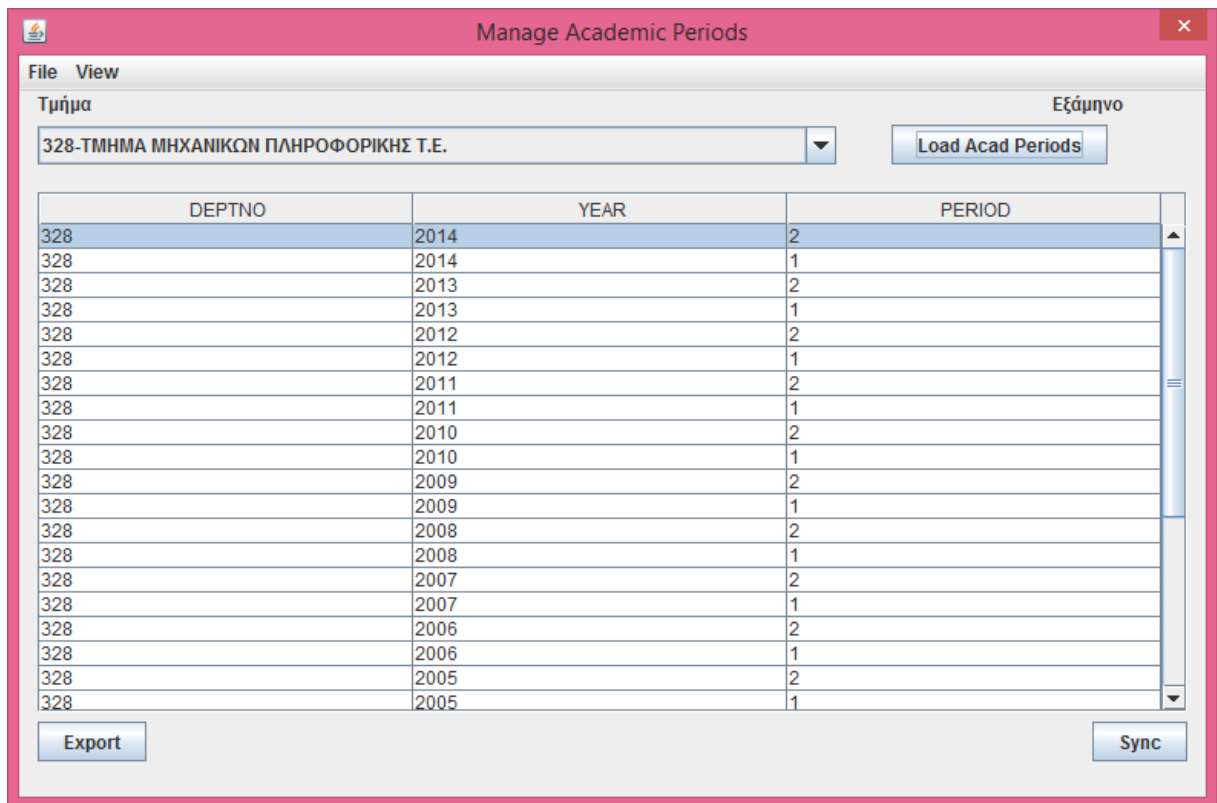
Στο πίνακάκι *Synchronize Academic Period Registration Instances* εμφανίζονται οι διεργασίες συγχρονισμού δηλώσεων ακαδημαϊκών περιόδων που έχουν αιτηθεί από τους προϊστάμενους των τμημάτων.

Στο πίνακάκι *Lesson Groups Full Instances* εμφανίζονται οι διεργασίες όπου οι ομάδες παρακολούθησης των μαθημάτων έχουν κορεστεί (Φοιτητές έχουν δηλώσει την ομάδα παρακολούθησης 'ΠΛΕΟΝΑΖΟΝΤΕΣ').

Στο πίνακάκι *New User Instances* εμφανίζονται οι νέοι λογαριασμοί χρηστών που έχουν αιτηθεί από τις γραμματείες των τμημάτων για να συγχρονιστούν με τον LDAP Server του ιδρύματος και με τη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών.

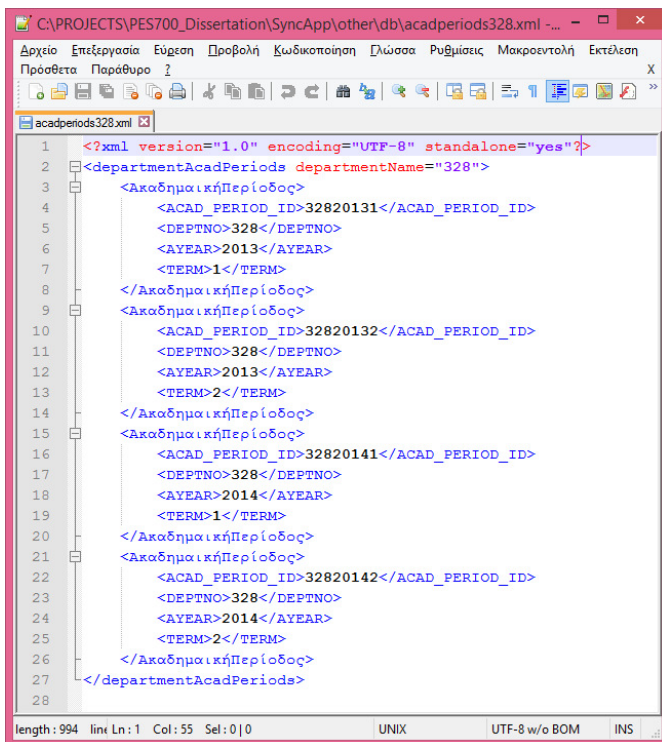
Σε κάθε είδος διεργασίας υπάρχει ένδειξη για το πλήθος των διεργασιών (Pending requests) που εκκρεμούν.

Η φόρμα συγχρονισμού ακαδημαϊκών περιόδων φαίνεται στην εικόνα 3.34 .



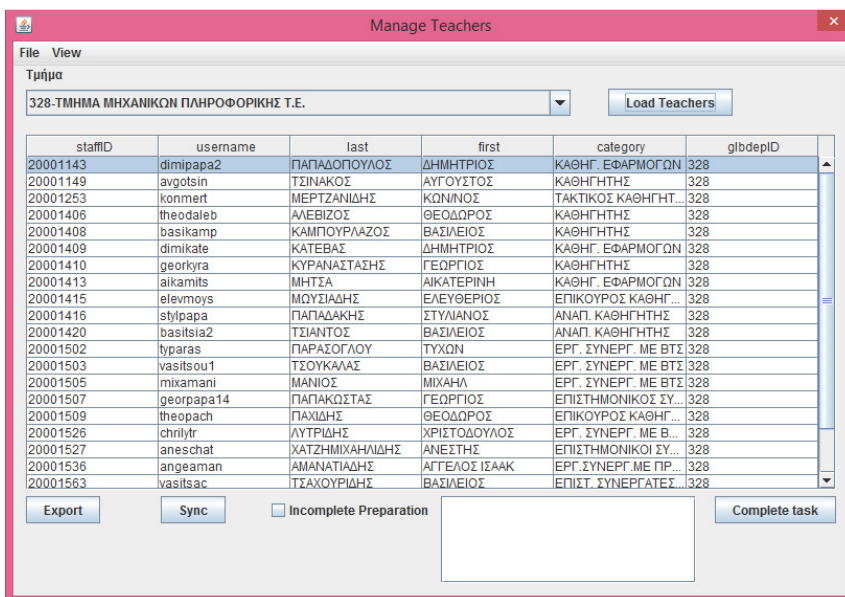
Εικόνα 3.34: Φόρμα συγχρονισμού ακαδημαϊκών περιόδων τμήματος

Ο διαχειριστής επιλέγει το τμήμα του ιδρύματος από τη λίστα των τμημάτων και στη συνέχεια επιλέγει το κουμπί *Load Acad Periods* για να διαβαστούν όλες οι ακαδημαϊκές περίοδοι του τμήματος από τη βάση δεδομένων του γραμματειακού συστήματος. Μόλις επιλεγεί το κουμπί *Export* τα δεδομένα εξάγονται σε αρχείο XML. Μόλις επιλεγεί το κουμπί *Sync* τα δεδομένα διαβάζονται από το XML αρχείο και εισάγονται στη βάση. Η δομή του αρχείου XML φαίνεται στην εικόνα 3.35.



Εικόνα 3.35: Δομή αρχείου XML ακαδημαϊκών περιόδων

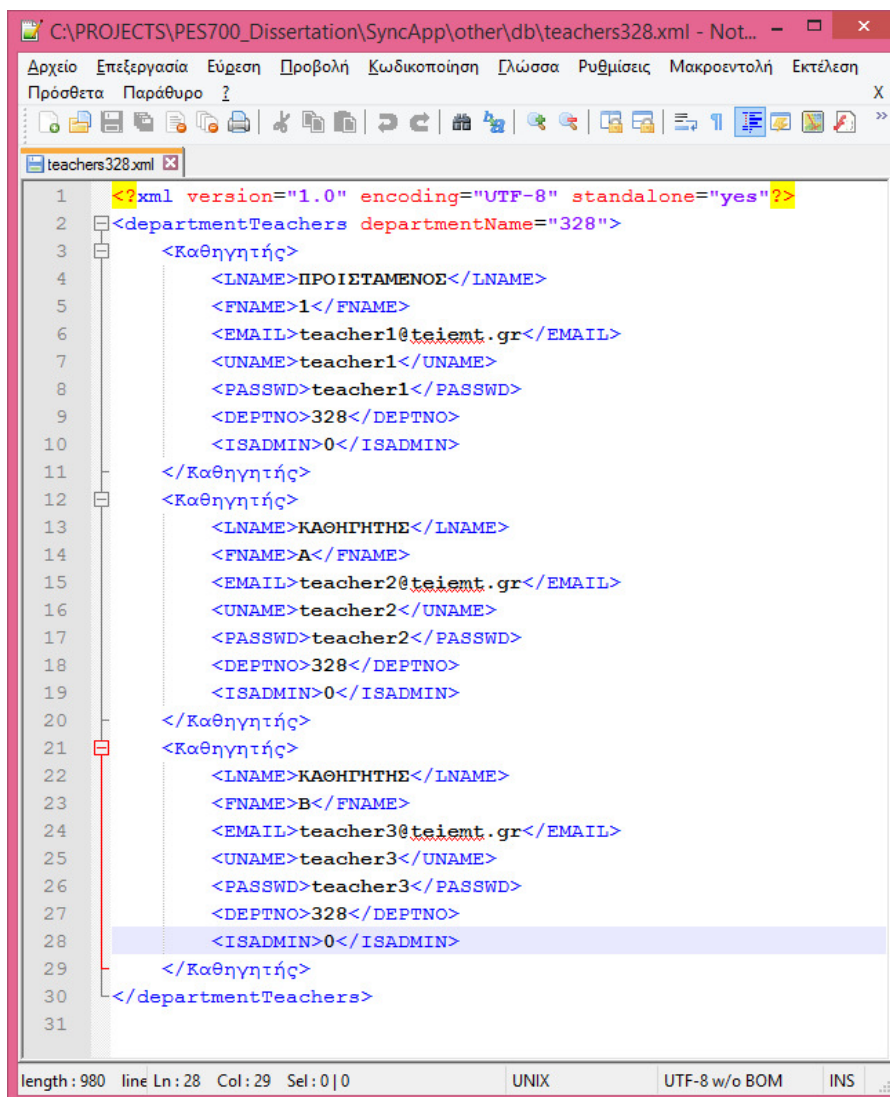
Η φόρμα συγχρονισμού εκπαιδευτικών φαίνεται στην εικόνα 3.36.



Εικόνα 3.36: Φόρμα συγχρονισμού εκπαιδευτικών τμήματος

Ο διαχειριστής επιλέγει το τμήμα του ιδρύματος από τη λίστα των τμημάτων και στη συνέχεια επιλέγει το κουμπί *Load Teachers* για να διαβαστούν όλοι οι εκπαιδευτικοί του τμήματος από τη βάση δεδομένων του γραμματειακού συστήματος. Μόλις επιλέγει το κουμπί *Export* τα δεδομένα

εξάγονται σε αρχείο XML. Μόλις επιλεγεί το κουμπί *Sync* τα δεδομένα διαβάζονται από το XML αρχείο και εισάγονται στη βάση. Η δομή του αρχείου XML φαίνεται στην εικόνα 3.37.

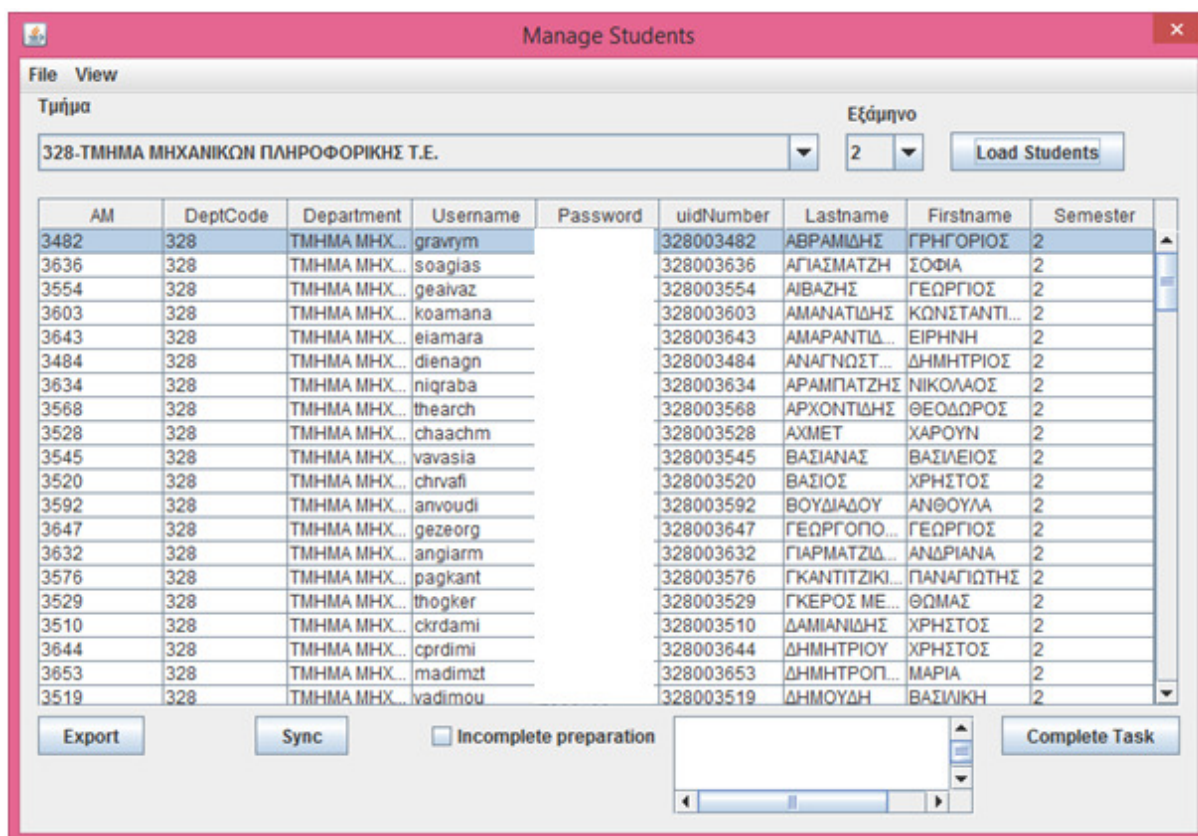


```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <departmentTeachers departmentName="328">
3   <Καθηγητής>
4     <LNAME>ΠΡΟΙΣΤΑΜΕΝΟΣ</LNAME>
5     <FNAME>1</FNAME>
6     <EMAIL>teacher1@teiemt.gr</EMAIL>
7     <UNAME>teacher1</UNAME>
8     <PASSWD>teacher1</PASSWD>
9     <DEPTNO>328</DEPTNO>
10    <ISADMIN>0</ISADMIN>
11  </Καθηγητής>
12  <Καθηγητής>
13    <LNAME>ΚΑΘΗΓΗΤΗΣ</LNAME>
14    <FNAME>Α</FNAME>
15    <EMAIL>teacher2@teiemt.gr</EMAIL>
16    <UNAME>teacher2</UNAME>
17    <PASSWD>teacher2</PASSWD>
18    <DEPTNO>328</DEPTNO>
19    <ISADMIN>0</ISADMIN>
20  </Καθηγητής>
21  <Καθηγητής>
22    <LNAME>ΚΑΘΗΓΗΤΗΣ</LNAME>
23    <FNAME>Β</FNAME>
24    <EMAIL>teacher3@teiemt.gr</EMAIL>
25    <UNAME>teacher3</UNAME>
26    <PASSWD>teacher3</PASSWD>
27    <DEPTNO>328</DEPTNO>
28    <ISADMIN>0</ISADMIN>
29  </Καθηγητής>
30 </departmentTeachers>
31
```

Εικόνα 3.37: Δομή αρχείου XML εκπαιδευτικών

Αν βρίσκεται σε εξέλιξη η BPM διεργασία συγχρονισμού νέας ακαδημαϊκής περιόδου και βρίσκεται σε εκκρεμότητα η εργασία (task) συγχρονισμού εκπαιδευτικών τότε ο διαχειριστής μπορεί να ολοκληρώσει την εργασία επιλέγοντας το κουμπί *Complete task*. Αν ο διαχειριστής διαπιστώσει πως υπάρχει κάποιο πρόβλημα με τη λίστα των εκπαιδευτικών τότε τσεκάρει το αντικείμενο *Incomplete Preparation* και τοποθετεί μήνυμα στο πλαίσιο *κειμένου*, ώστε να ενημερωθούν οι χρήστες που σχετίζονται με τη διεργασία.

Η φόρμα συγχρονισμού φοιτητών φαίνεται στην εικόνα 3.38.



Εικόνα 3.38: Φόρμα συγχρονισμού φοιτητών τμήματος

Ο διαχειριστής επιλέγει το τμήμα του ιδρύματος από τη λίστα των τμημάτων και στη συνέχεια επιλέγει το κουμπί *Load Students* για να διαβαστούν οι φοιτητές του τμήματος από τη βάση δεδομένων του γραμματειακού συστήματος. Μόλις επιλεγεί το κουμπί *Export* τα δεδομένα εξάγονται σε αρχείο XML. Μόλις επιλεγεί το κουμπί *Sync* τα δεδομένα διαβάζονται από το XML αρχείο και εισάγονται στη βάση. Η δομή του αρχείου XML φαίνεται στην εικόνα 3.39.

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
2 <departmentStudents departmentName="328">
3   <φοιτητής>
4     <ΑΕΜ>1</ΑΕΜ>
5     <LNAME>ΦΟΙΤΗΤΗΣ 1</LNAME>
6     <FNAME>ΕΠΟΥΔΑΣΤΗΣ 1</FNAME>
7     <EMAIL>student1@teiemt.gr</EMAIL>
8     <SEMESTER>1</SEMESTER>
9     <UNAME>student1</UNAME>
10    <PASSWD>student1</PASSWD>
11    <DEPTNO>328</DEPTNO>
12  </φοιτητής>
13  <φοιτητής>
14    <ΑΕΜ>2</ΑΕΜ>
15    <LNAME>ΦΟΙΤΗΤΗΣ 2</LNAME>
16    <FNAME>ΕΠΟΥΔΑΣΤΗΣ 2</FNAME>
17    <EMAIL>student2@teiemt.gr</EMAIL>
18    <SEMESTER>2</SEMESTER>
19    <UNAME>student2</UNAME>
20    <PASSWD>student2</PASSWD>
21    <DEPTNO>328</DEPTNO>
22  </φοιτητής>
23 </departmentStudents>

```

Εικόνα 3.39: Δομή αρχείου XML φοιτητών

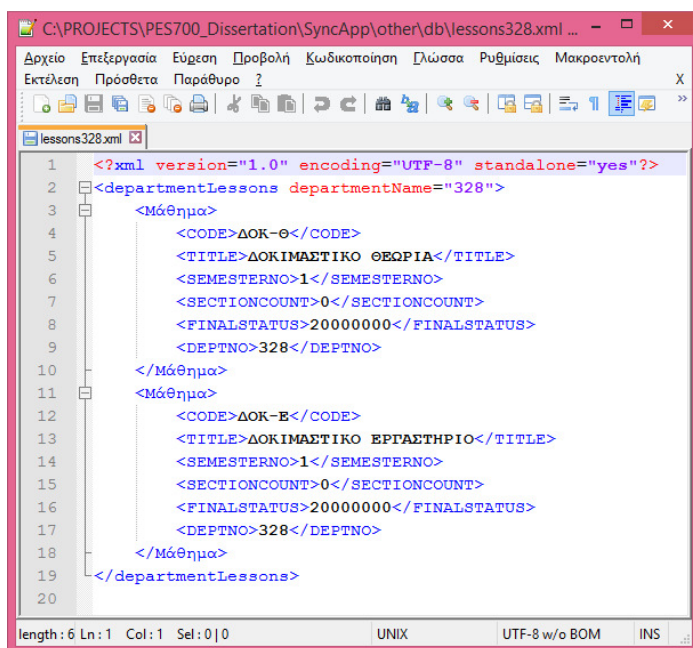
Αν βρίσκεται σε εξέλιξη η BPM διεργασία συγχρονισμού νέας ακαδημαϊκής περιόδου και βρίσκεται σε εκκρεμότητα η εργασία (task) συγχρονισμού φοιτητών τότε ο διαχειριστής μπορεί να ολοκληρώσει την εργασία επιλέγοντας το κουμπί *Complete task*. Αν ο διαχειριστής διαπιστώσει πως υπάρχει κάποιο πρόβλημα με τη λίστα των φοιτητών τότε τσεκάρει το αντικείμενο *Incomplete Preparation* και τοποθετεί μήνυμα στο πλαίσιο κειμένου, ώστε να ενημερωθούν οι χρήστες που σχετίζονται με τη διεργασία.

Η φόρμα συγχρονισμού μαθημάτων φαίνεται στην εικόνα 3.40.

CODE	TITLE	SEMESTERNO	SECTIONCOUNT	FINALSTATUS	DEPTNO
101ΓΥΥΚ	ΑΓΓΛΙΚΗ ΤΕΧΝΙΚΗ Ο...	1	0	20000000	328
102ΓΥΥΚ-Ε	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠ...	1	0	20000001	328
102ΓΥΥΚ-Θ	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠ...	1	0	20000000	328
103ΓΥΥΚ	ΔΙΑΚΡΙΤΑ ΜΑΘΗΜΑΤΙ...	1	0	20000000	328
104ΓΥΥΚ-Ε	ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡ...	1	0	20000001	328
104ΓΥΥΚ-Θ	ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡ...	1	0	20000000	328
105ΓΥΥΚ-Ε	ΗΛΕΚΤΡΙΚΑ ΚΥΚΛΩΜ...	1	0	20000001	328
105ΓΥΥΚ-Θ	ΗΛΕΚΤΡΙΚΑ ΚΥΚΛΩΜ...	1	0	20000000	328
106ΓΥΥΚ	ΜΑΘΗΜΑΤΙΚΑ Ι	1	0	20000000	328
201ΕΥΥΚ-Ε	ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΒΑΣΕ...	2	0	20000001	328
201ΕΥΥΚ-Θ	ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΒΑΣΕ...	2	0	20000000	328
202ΓΥΥΚ-Ε	ΗΛΕΚΤΡΟΝΙΚΑ ΚΥΚΛ...	2	0	20000001	328
202ΓΥΥΚ-Θ	ΗΛΕΚΤΡΟΝΙΚΑ ΚΥΚΛ...	2	0	20000000	328
203ΓΥΥΚ-Ε	ΘΕΩΡΙΕΣ ΜΑΘΗΣΗΣ ...	2	0	20000001	328
203ΓΥΥΚ-Θ	ΘΕΩΡΙΕΣ ΜΑΘΗΣΗΣ ...	2	0	20000000	328
204ΓΥΥΚ	ΜΑΘΗΜΑΤΙΚΑ ΙΙ	2	0	20000000	328
205ΓΥΥΚ	ΜΕΘΟΔΟΛΟΓΙΑ ΕΠΙΣ...	2	0	20000000	328
206ΕΥΥΚ-Ε	ΤΕΧΝΙΚΕΣ ΠΡΟΓΡΑΜ...	2	0	20000001	328
206ΕΥΥΚ-Θ	ΤΕΧΝΙΚΕΣ ΠΡΟΓΡΑΜ...	2	0	20000000	328
301ΕΥΥΚ	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΔΟ...	3	0	20000000	328

Εικόνα 3.40: Φόρμα συγχρονισμού μαθημάτων τμήματος

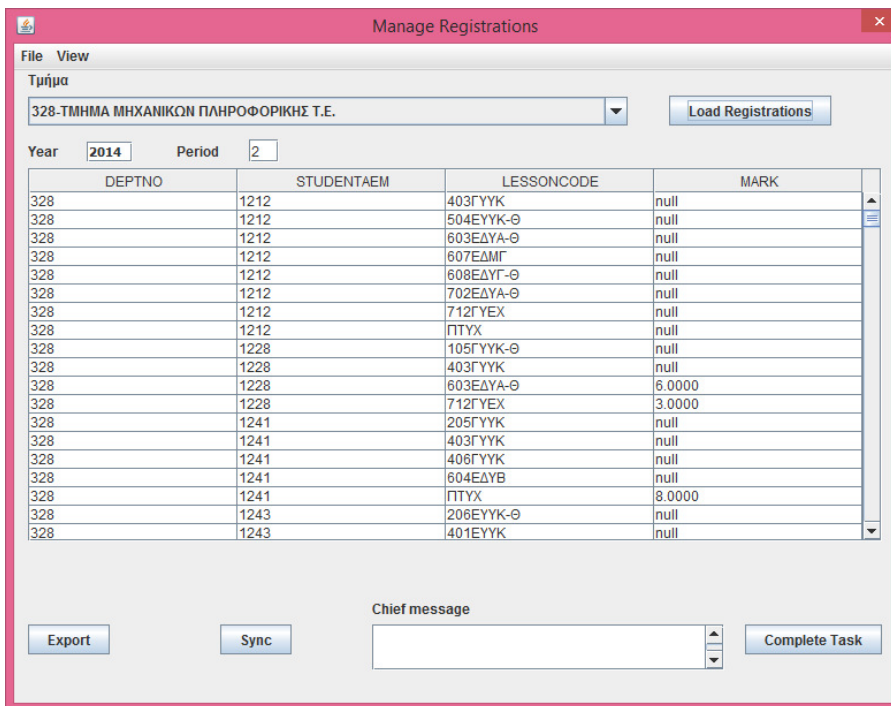
Ο διαχειριστής επιλέγει το τμήμα του ιδρύματος από τη λίστα των τμημάτων και στη συνέχεια επιλέγει το κουμπί *Load Lessons* για να διαβαστούν τα μαθήματα του τμήματος από τη βάση δεδομένων του γραμματειακού συστήματος. Μόλις επιλεγεί το κουμπί *Export* τα δεδομένα εξάγονται σε αρχείο XML. Μόλις επιλεγεί το κουμπί *Sync* τα δεδομένα διαβάζονται από το XML αρχείο και εισάγονται στη βάση. Η δομή του αρχείου XML φαίνεται στην εικόνα 3.41.



Εικόνα 3.41: Δομή αρχείου XML μαθημάτων

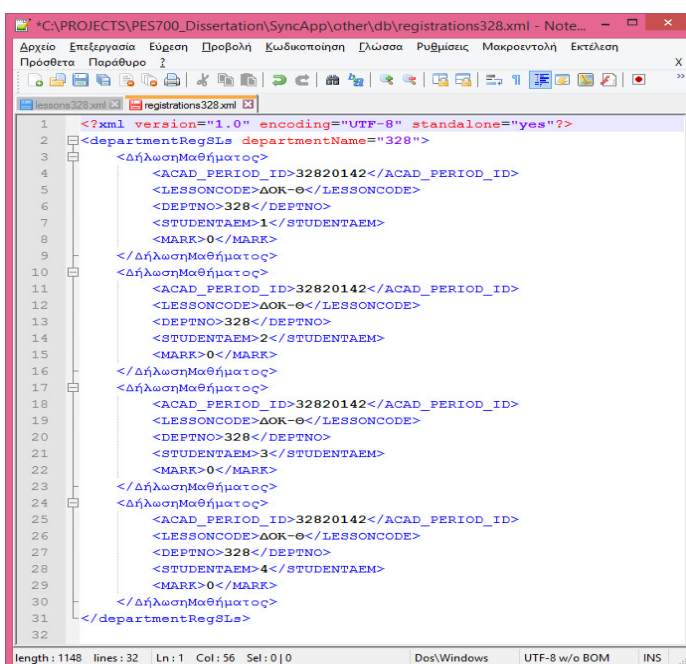
Αν βρίσκεται σε εξέλιξη η BPM διεργασία συγχρονισμού νέας ακαδημαϊκής περιόδου και βρίσκεται σε εκκρεμότητα η εργασία (task) συγχρονισμού μαθημάτων τότε ο διαχειριστής μπορεί να ολοκληρώσει την εργασία επιλέγοντας το κουμπί *Complete task*. Αν ο διαχειριστής διαπιστώσει πως υπάρχει κάποιο πρόβλημα με τη λίστα των μαθημάτων τότε τσεκάρει το αντικείμενο *Incomplete Preparation* και τοποθετεί μήνυμα στο πλαίσιο *κειμένου*, ώστε να ενημερωθούν οι χρήστες που σχετίζονται με τη διεργασία.

Η φόρμα συγχρονισμού δηλώσεων μαθημάτων φαίνεται στην εικόνα 3.42.



Εικόνα 3.42: Φόρμα συγχρονισμού δηλώσεων μαθημάτων φοιτητών τμήματος

Ο διαχειριστής επιλέγει το τμήμα του ιδρύματος από τη λίστα των τμημάτων, το ακαδημαϊκό έτος και την περίοδο και στη συνέχεια επιλέγει το κουμπί *Load Registrations* για να διαβαστούν οι δηλώσεις μαθημάτων από τη βάση δεδομένων του γραμματειακού συστήματος. Μόλις επιλέγει το κουμπί *Export* τα δεδομένα εξάγονται σε αρχείο XML. Μόλις επιλέξει το κουμπί *Sync* τα δεδομένα διαβάζονται από το XML αρχείο και εισάγονται στη βάση. Η δομή του αρχείου XML φαίνεται στην εικόνα 3.43.



Εικόνα 3.43: Δομή αρχείου XML δηλώσεων μαθημάτων φοιτητών

Αν βρίσκεται σε εξέλιξη η BPM διεργασία συγχρονισμού δηλώσεων ακαδημαϊκής περιόδου και βρίσκεται σε εκκρεμότητα η εργασία (task) συγχρονισμού δηλώσεων μαθημάτων τότε ο διαχειριστής μπορεί να ολοκληρώσει την εργασία επιλέγοντας το κουμπί *Complete task*. Αν ο διαχειριστής διαπιστώσει πως υπάρχει κάποιο πρόβλημα με τη λίστα των δηλώσεων μαθημάτων τότε τσεκάρει το αντικείμενο *Incomplete Preparation* και τοποθετεί μήνυμα στο πλαίσιο κειμένου, ώστε να ενημερωθούν οι χρήστες που σχετίζονται με τη διεργασία.

Η φόρμα συγχρονισμού λογαριασμού νέου χρήστη μπορεί να ανοίξει όταν δημιουργηθεί BPM διεργασία συγχρονισμού νέου χρήστη από τη γραμματεία. Η φόρμα φαίνεται στην εικόνα 3.44.

The image shows a web application window titled "New User Requests Process". Inside, there is a form for creating a new user. At the top, there is a dropdown menu with "aggetata2" selected and a "Load user details" button. Below this is a section titled "New User details" containing several input fields for user information. The fields are organized as follows:

- Αριθμός Μητρώου:** 123
- Κωδικός τμήματος:** 328
- Όνομα τμήματος:** ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.
- Username:** aggetata2
- Password:** masked with asterisks
- e-mail:** @teiemt.gr
- Επώνυμο (sn):** ΤΑΤΑΣ
- Όνομα (givenName):** ΑΓΓΕΛΟΣ
- UidNumber:** 100001
- gidNumber:** 500
- Όνοματεπώνυμο (cn):** ΤΑΤΑΣ ΑΓΓΕΛΟΣ
- edupersonprincipalname:** @teikav.edu.gr
- edupersonaffiliation:** student
- schachomeorganization:** teikav.edu.gr
- eduPersonOrgDN:** dc=teikav,dc=edu,dc=gr
- o:** Eastern Macedonia and Thrace Institute of Technology

At the bottom of the form, there are two buttons: "Create User" and "Complete CreateUser Task".

Εικόνα 3.44: Φόρμα συγχρονισμού λογαριασμού νέου χρήστη

Το κουμπί *Load User details* διαβάζει τα στοιχεία του νέου χρήστη από το γραμματειακό σύστημα και τα τοποθετεί στα πλαίσια κειμένου της φόρμας. Επιπλέον συμπληρώνει και τα

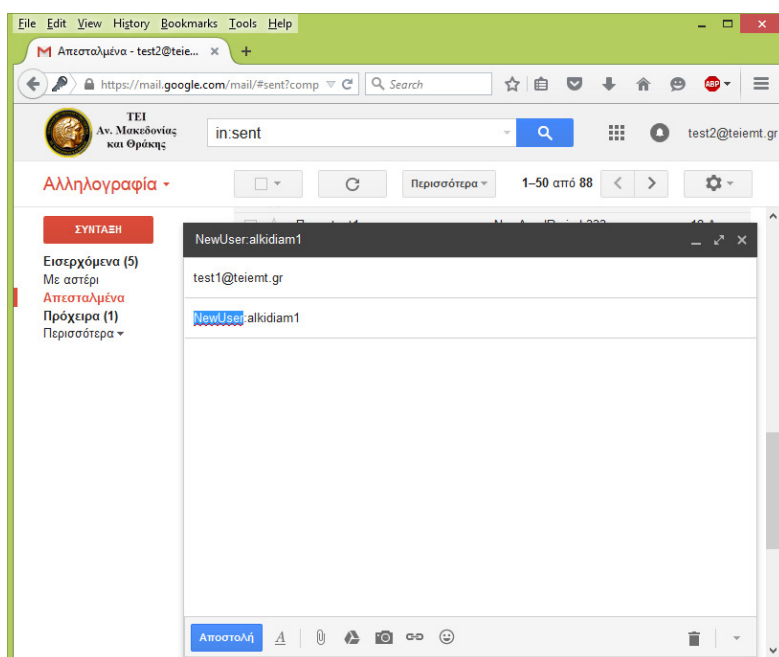
επιπλέον πεδία χρήστη που απαιτεί ο LDAP Server. Ο διαχειριστής επιλέγει το κουμπί *Create User* για να εισαχθεί ο νέος χρήστης στον LDAP και στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών. Με το κουμπί *Complete CreateUser Task* ολοκληρώνεται η BPM διεργασία συγχρονισμού λογαριασμού νέου χρήστη.

3.1.5 Διεπαφή υποσυστήματος διεργασιών.

Το υποσύστημα διεργασιών δεν έχει φόρμες διεπαφής. Λειτουργεί ως υπηρεσία REST (rest service) που ενσωματώνει τις διεργασίες BPM και παρέχει (κάνει export) τις απαιτούμενες μεθόδους (Rest Client methods) ώστε να μπορούν να χρησιμοποιηθούν οι διεργασίες από τα υποσυστήματα. Ωστόσο στις επόμενες παραγράφους περιγράφουμε από την πλευρά των χρηστών το βασικό σενάριο χρήσης κάθε διεργασίας.

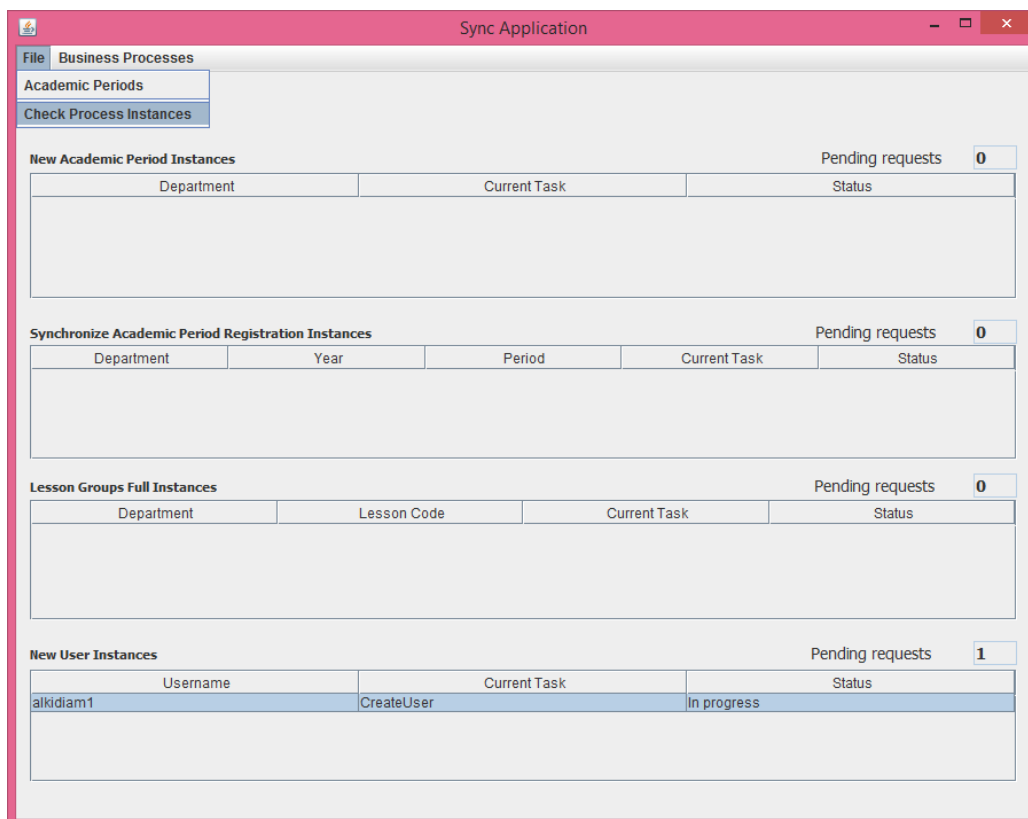
Περίπτωση χρήσης της διεργασίας συγχρονισμού λογαριασμού νέου χρήστη

Η γραμματεία (π.χ. με λογαριασμό test2@teiemt.gr) στέλνει e-mail στο test1@teiemt.gr (support email account) δίνοντας στο θέμα του μηνύματος τα στοιχεία (με τη μορφή) **NewUser:alkdiam1**. (Εικόνα 3.45)



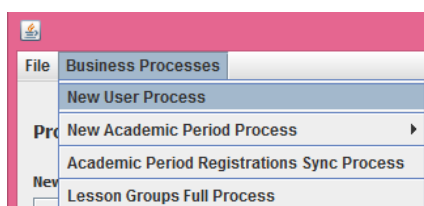
Εικόνα 3.45: Αποστολή e-mail από γραμματεία

Ο διαχειριστής από το υποσύστημα συγχρονισμού εκτελεί την επιλογή *Check Process Instances*, εκκινείται η BPM διεργασία και εφόσον ο χρήστης που έστειλε η γραμματεία δεν υπάρχει στον LDAP Server εμφανίζεται στο πινακάκι *New User Instances* όπως φαίνεται στην εικόνα 3.46.



Εικόνα 3.46: Φόρμα προβολής ενεργών διεργασιών

Εφόσον υπάρχει BPM διεργασία συγχρονισμού λογαριασμού νέου χρήστη σε εκκρεμότητα ο διαχειριστής επιλέγει τη διαχείριση της αίτησης όπως φαίνεται στην εικόνα 2.47 .



Εικόνα 3.47: Μενού ανοίγματος φορμών συγχρονισμού-διαχείρισης εργασιών

Εμφανίζεται η φόρμα διαχείρισης του αιτήματος. (Εικόνα 3.48)

Εικόνα 3.48: Φόρμα διαχείρισης διεργασίας NewUser

Ο διαχειριστής επιλέγει το κουμπί *Load user details* ώστε να βρεθούν τα στοιχεία του χρήστη από το υφιστάμενο σύστημα της ηλεκτρονικής γραμματείας του ιδρύματος. Εμφανίζεται η φόρμα σύνδεσης με τη βάση δεδομένων του γραμματειακού συστήματος όπως φαίνεται στην εικόνα 3.49.

Εικόνα 3.49: Φόρμα σύνδεσης με τη βάση δεδομένων της ηλεκτρονικής γραμματείας

Μετα από επιτυχή σύνδεση εμφανίζονται τα στοιχεία του φοιτητή όπως φαίνεται στην εικόνα 3.50.

New User Requests Process

New User Requests
alkidiam1

New User details

Αριθμός Μητρώου
1234

Κωδικός τμήματος
328

Όνομα τμήματος
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

Username
alkidiam1

Password
●●●●●●●●

e-mail
alkidiam1@teiemt.gr

Επώνυμο (sn)
ΔΙΑΜΑΝΤΟΠΟΥΛΟΣ

Όνομα (givenName)
ΑΛΕΞΑΝΔΡΟΣ

UidNumber
3455367677

gidNumber
500

Όνοματεπώνυμο (cn)
ΔΙΑΜΑΝΤΟΠΟΥΛΟΣ ΑΛΕΞΑΝΔΡΟΣ

edupersonprincipalname
alkidiam1@teikav.edu.gr

edupersonaffiliation
student

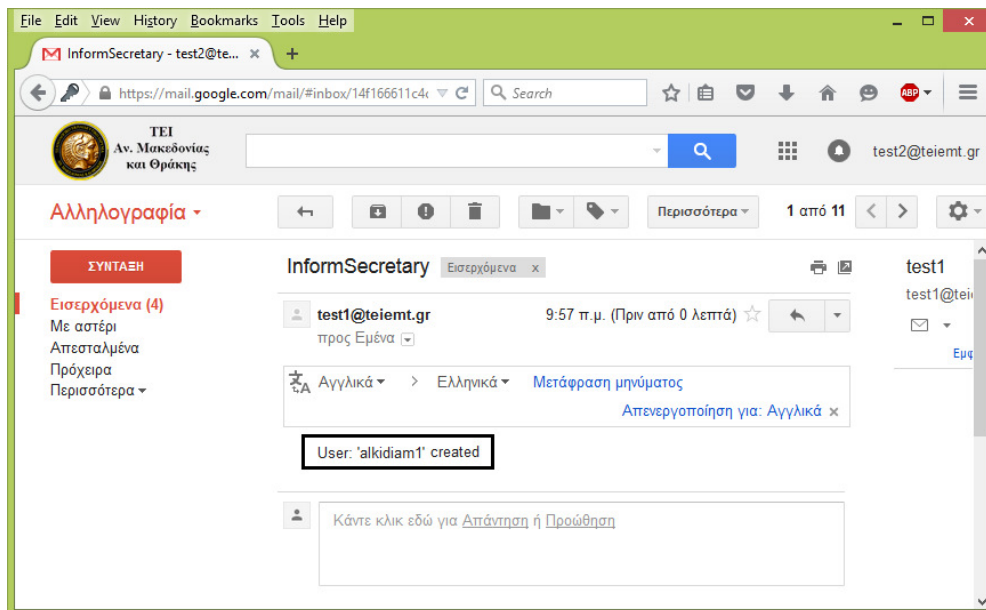
schachomeorganization
teikav.edu.gr

eduPersonOrgDN
dc=teikav,dc=edu,dc=gr

o
Eastern Macedonia and Thrace Institute of Technology

Εικόνα 3.50: Φόρμα διαχείρισης διεργασίας NewUser

Με την επιλογή *Create User* δημιουργείται ο χρήστης στον LDAP Server και στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών, ενώ με την επιλογή *Complete CreateUser Task* στέλνεται μήνυμα στη μηχανή επεξεργασίας διεργασιών (process engine) για την ολοκλήρωσή της εργασίας (task) CreateUser. Με την ολοκλήρωσή της εργασίας αποστέλλεται ενημερωτικό μήνυμα στη γραμματεία όπως φαίνεται στην εικόνα 3.51.



Εικόνα 3.51: Παραλαβή e-mail από τη γραμματεία

Περίπτωση χρήσης της διεργασίας προετοιμασίας νέας ακαδημαϊκής περιόδου

Ο προϊστάμενος από το υποσύστημα εκπαιδευτικού θέλει να προετοιμάσει τη νέα ακαδημαϊκή περίοδο (έστω για την χειμερινή περίοδο του ακαδημαϊκού έτους 2014). Επιλέγει από το μενού *Ακαδημαϊκές Περιόδου* και προβάλλεται μεταξύ άλλων πληροφοριών ένας πίνακας με την ενεργή ακαδημαϊκή περίοδο τμήματος όπως φαίνεται στην εικόνα 3.52 .

Επίπλομο
ΠΡΟΙΣΤΑΜΕΝΟΣ

Όνομα
1

e-mail
teacher1@teiemt.gr

username
teacher1

Τμήμα
328

Ακαδημαϊκοί Περιόδοι τμήματος

Στον παρακάτω πίνακα μπορείτε να επεξεργαστείτε τις ακαδημαϊκές περιόδου του τμήματος

Τμήμα: ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε. Επιλογή

Προβολή Ακαδημαϊκών Περιόδων

Ενεργή ακαδημαϊκή περίοδος τμήματος

Αίτημα προετοιμασίας περιόδου	<input type="checkbox"/>
Εξέλιξη αιτήματος	<input type="checkbox"/>
Κατάσταση διεργασίας	<input type="checkbox"/>

Κωδικός:

Έτος:

Ημ/νία έναρξης δηλώσεων:

Ημ/νία έναρξης Προτεραιότητας εξαμήνου:

Κωδικός τμήματος:

Περίοδος:


Ημ/νία λήξης δηλώσεων:

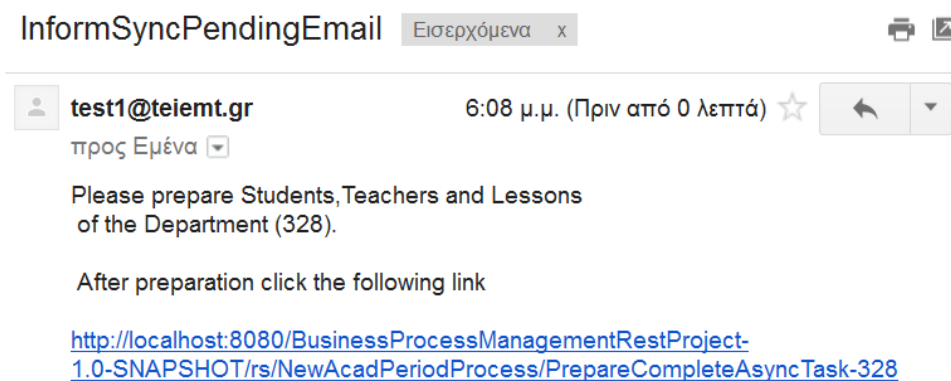
Ημ/νία λήξης Προτεραιότητας εξαμήνου:

Ιστορικό ακαδημαϊκών περιόδων τμήματος

Κωδικός	Κωδικός τμήματος	Ακαδημαϊκό Έτος	Περίοδος	Αίτημα συγχρονισμού δηλώσεων	Εξέλιξη αιτήματος
32820142	328	2014	2	<input checked="" type="checkbox"/>	
32820141	328	2014	1	<input checked="" type="checkbox"/>	

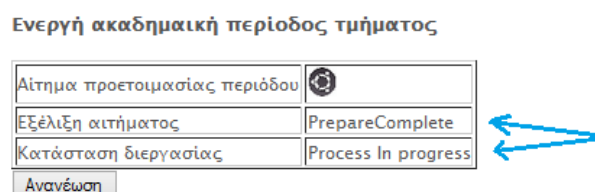
Εικόνα 3.52: Προβολή πίνακα ενεργής ακαδημαϊκής περιόδου

Στον πίνακα με την ενεργή ακαδημαϊκή περίοδο του τμήματος επιλέγει το εικονίδιο . Με την επιλογή αυτή εκκινείται η διεργασία προετοιμασίας νέας ακαδημαϊκής περιόδου και αποστέλλεται e-mail από το λογαριασμό support (π.χ. test1@teiemt.gr) στη γραμματεία του τμήματος (π.χ. test2@teiemt.gr) να προετοιμάσει τα στοιχεία των φοιτητών, των εκπαιδευτικών και των μαθημάτων. Το e-mail φαίνεται στην εικόνα 3.53.



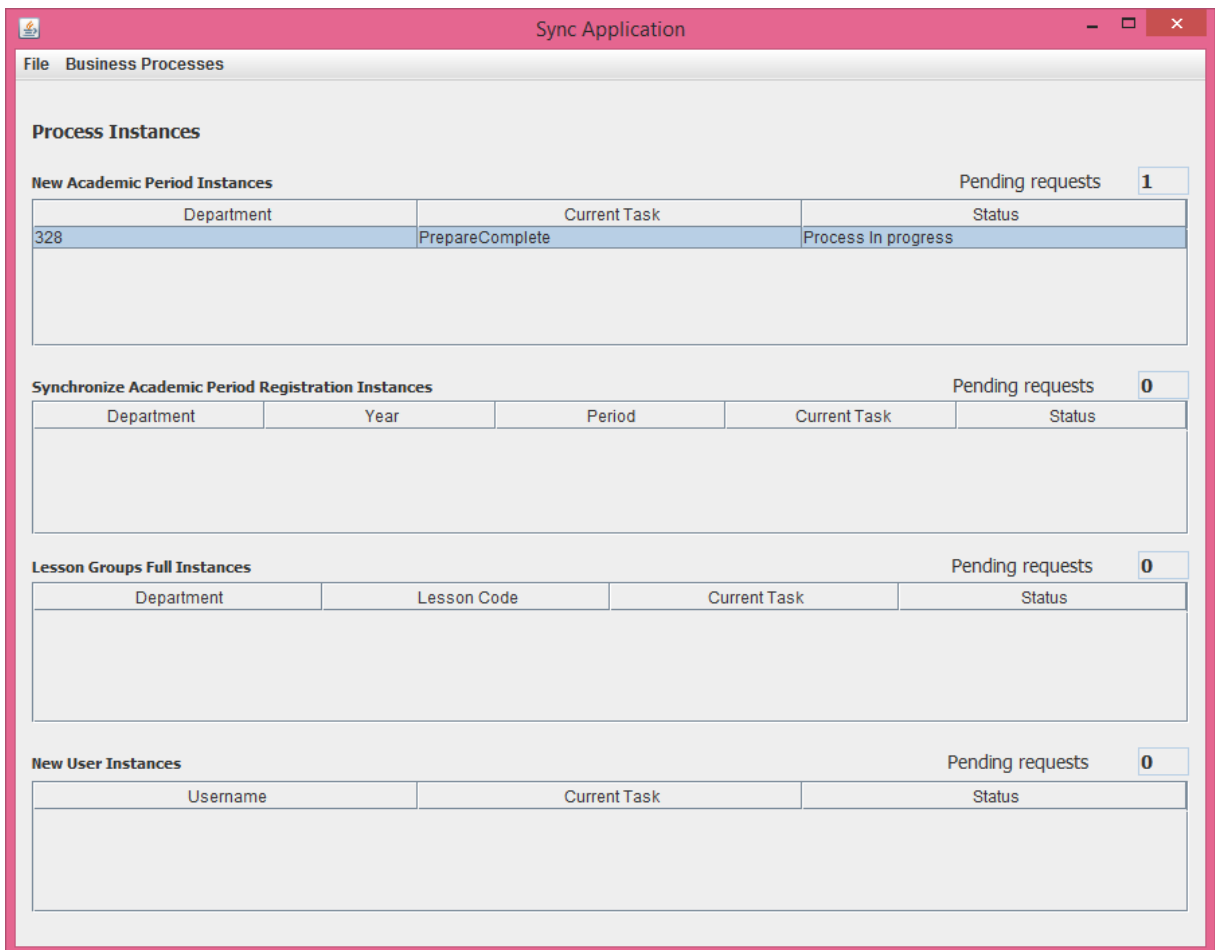
Εικόνα 3.53: Παραλαβή e-mail από τη γραμματεία

Στο μεταξύ ο προϊστάμενος μπορεί να βλέπει την εξέλιξη του αιτήματος και την κατάσταση της διεργασίας στον πίνακα ενεργής ακαδημαϊκής περιόδου τμήματος που φαίνεται στην εικόνα 3.54.



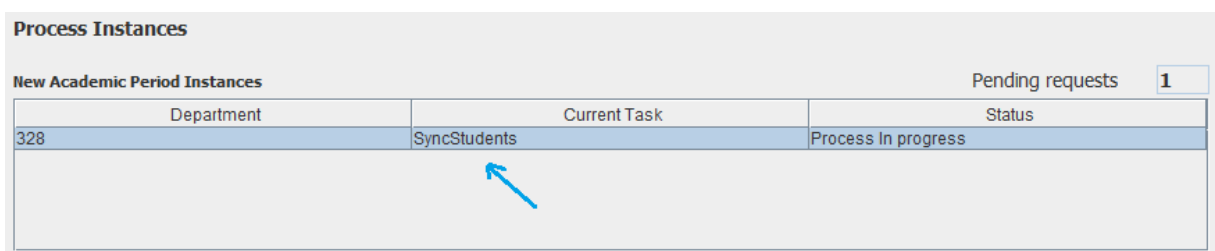
Εικόνα 3.54: Προβολή εξέλιξης αιτήματος και κατάστασης διεργασίας

Ο διαχειριστής επίσης από το υποσύστημα συγχρονισμού θα έχει ένδειξη (επιλέγοντας File→Check Process Instances) πως εκκρεμεί η παραπάνω εργασία (task) PrepareComplete, όπως φαίνεται στην εικόνα 3.55.



Εικόνα 3.55: Προβολή τρέχουσας εργασίας και κατάστασης διεργασίας

Μόλις η γραμματεία κάνει την προετοιμασία φοιτητών, εκπαιδευτικών και μαθημάτων και επιλέξει το σύνδεσμο που της είχε έρθει στο e-mail ο διαχειριστής μπορεί να δει και να επεξεργαστεί την επόμενη σε εξέλιξη εργασία (task).



Εικόνα 3.56: Προβολή τρέχουσας εργασίας και κατάστασης διεργασίας

Σημείωση: Ο διαχειριστής για να βλέπει την εξέλιξη των διεργασιών πρέπει να επιλέγει από το υποσύστημα συγχρονισμού (επιλέγοντας File→Check Process Instances). Προσωρινά, δεν έχει ενσωματωθεί timer -ώστε να γίνεται αυτόματα ανανέωση της κατάστασης των διεργασιών- για να υπάρχει καλύτερη εποπτεία στη συμπεριφορά του συστήματος κατά την ανάπτυξή του.

Στη συνέχεια ο διαχειριστής επιλέγει από το μενού *Business Processes* την επιλογή *New Academic Period Process* → *Synchronize Department Students* όπως φαίνεται στην εικόνα 3.57.

Business Processes	
New User Process	
New Academic Period Process	Synchronize Department Students
Academic Period Registrations Sync Process	Synchronize Department Teachers
Lesson Groups Full Process	Synchronize Department Lessons

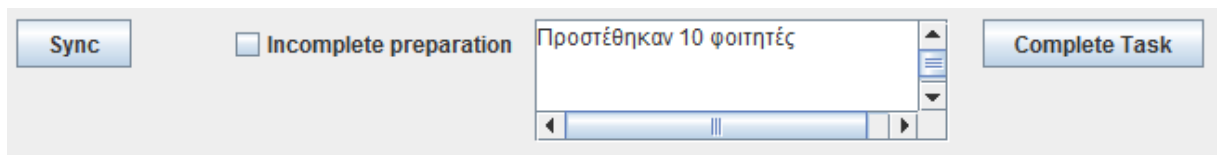
Εικόνα 3.57: Επιλογή συγχρονισμού φοιτητών τμήματος

Φορτώνει τους φοιτητές για το τμήμα που ζήτησε συγχρονισμό όπως φαίνεται στην εικόνα 3.58, και επιλέγει *Export, Sync*.

AM	DeptCode	Department	Username	Password	uidNumber	Lastname	Firstname	Semester
2722	328	ΤΜΗΜΑ ΜΗΧ...			328002722			10
3482	328	ΤΜΗΜΑ ΜΗΧ...			328003482			2
3180	328	ΤΜΗΜΑ ΜΗΧ...			328003180			6
1581	328	ΤΜΗΜΑ ΜΗΧ...			328001581			21
3399	328	ΤΜΗΜΑ ΜΗΧ...			328003399			4
2947	328	ΤΜΗΜΑ ΜΗΧ...			328002947			10
3636	328	ΤΜΗΜΑ ΜΗΧ...			328003636			2
3276	328	ΤΜΗΜΑ ΜΗΧ...			328003276			6
1564	328	ΤΜΗΜΑ ΜΗΧ...			328001564			21
2615	328	ΤΜΗΜΑ ΜΗΧ...			328002615			11
2411	328	ΤΜΗΜΑ ΜΗΧ...			328002411			16
1196	328	ΤΜΗΜΑ ΜΗΧ...			328001196			24
739	328	ΤΜΗΜΑ ΜΗΧ...			328000739			27
2463	328	ΤΜΗΜΑ ΜΗΧ...			328002463			15
3554	328	ΤΜΗΜΑ ΜΗΧ...			328003554			2
1013	328	ΤΜΗΜΑ ΜΗΧ...			328001013			25
3471	328	ΤΜΗΜΑ ΜΗΧ...			328003471			4
2570	328	ΤΜΗΜΑ ΜΗΧ...			328002570			13
1666	328	ΤΜΗΜΑ ΜΗΧ...			328001666			21
2983	328	ΤΜΗΜΑ ΜΗΧ...			328002983			8

Εικόνα 3.58: Φόρμα συγχρονισμού φοιτητών τμήματος

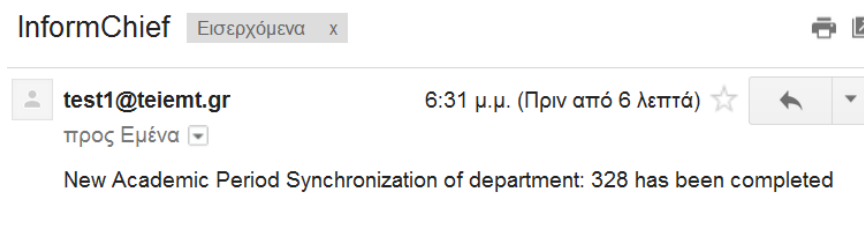
Μετά την επιλογή *Sync* εμφανίζονται στο πλαίσιο κειμένου *Chief message* πόσοι φοιτητές προστέθηκαν, ώστε να ενημερωθεί ο προϊστάμενος του τμήματος όπως φαίνεται στην εικόνα 3.59.



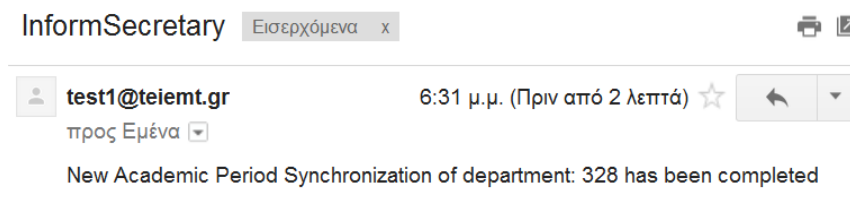
Εικόνα 3.59: Συγχρονισμός φοιτητών και ενημέρωση προϊστάμενου

Τέλος ο διαχειριστής επιλέγει *Complete Task* για να σηματοδοτηθεί το τέλος της εργασίας (task) και να πάρει τη θέση του η εργασία SyncTeachers και στη συνέχεια η εργασία SyncLessons , ώστε να συγχρονιστούν οι εκπαιδευτικοί και τα μαθήματα του τμήματος.

Μόλις ο διαχειριστής συγχρονίσει τα μαθήματα και σηματοδοτήσει το τέλος της εργασίας SyncLessons στέλνεται ενημερωτικό e-mail στον προϊστάμενο και στη γραμματεία του τμήματος. (Εικόνες 3.60, 3.61)



Εικόνα 3.60: Παραλαβή ενημερωτικού e-mail από τον προϊστάμενο



Εικόνα 3.61: Παραλαβή ενημερωτικού e-mail από τη γραμματεία

Ο προϊστάμενος μπορεί να διαπιστώσει πως η προετοιμασία της νέας ακαδημαϊκής περιόδου ολοκληρώθηκε αν ξαναγυρίσει στη σελίδα ακαδημαϊκών περιόδων όπως φαίνεται στην εικόνα 3.62.

Προβολή Ακαδημαϊκών Περιόδων

Ενεργή ακαδημαϊκή περίοδος τμήματος

Αίτημα προετοιμασίας περιόδου	
Εξέλιξη αιτήματος	
Κατάσταση διεργασίας	Process Completed
Ανανέωση	




Εικόνα 3.62: Διαπίστωση ολοκλήρωσης διεργασίας

Περίπτωση χρήσης της διεργασίας συγχρονισμού δηλώσεων ακαδημαϊκής περιόδου

Έστω πως ο προϊστάμενος από το υποσύστημα εκπαιδευτικού θέλει να κάνει συγχρονισμό δηλώσεων για την χειμερινή περίοδο του ακαδημαϊκού έτους 2014. Επιλέγει από το μενού την επιλογή *Ακαδημαϊκές Περιόδους* και προβάλλεται μεταξύ άλλων πληροφοριών ένας πίνακας με το ιστορικό ακαδημαϊκών περιόδων όπως φαίνεται στην εικόνα 3.63.

Εικόνα 3.63: Ιστορικό ακαδημαϊκών περιόδων

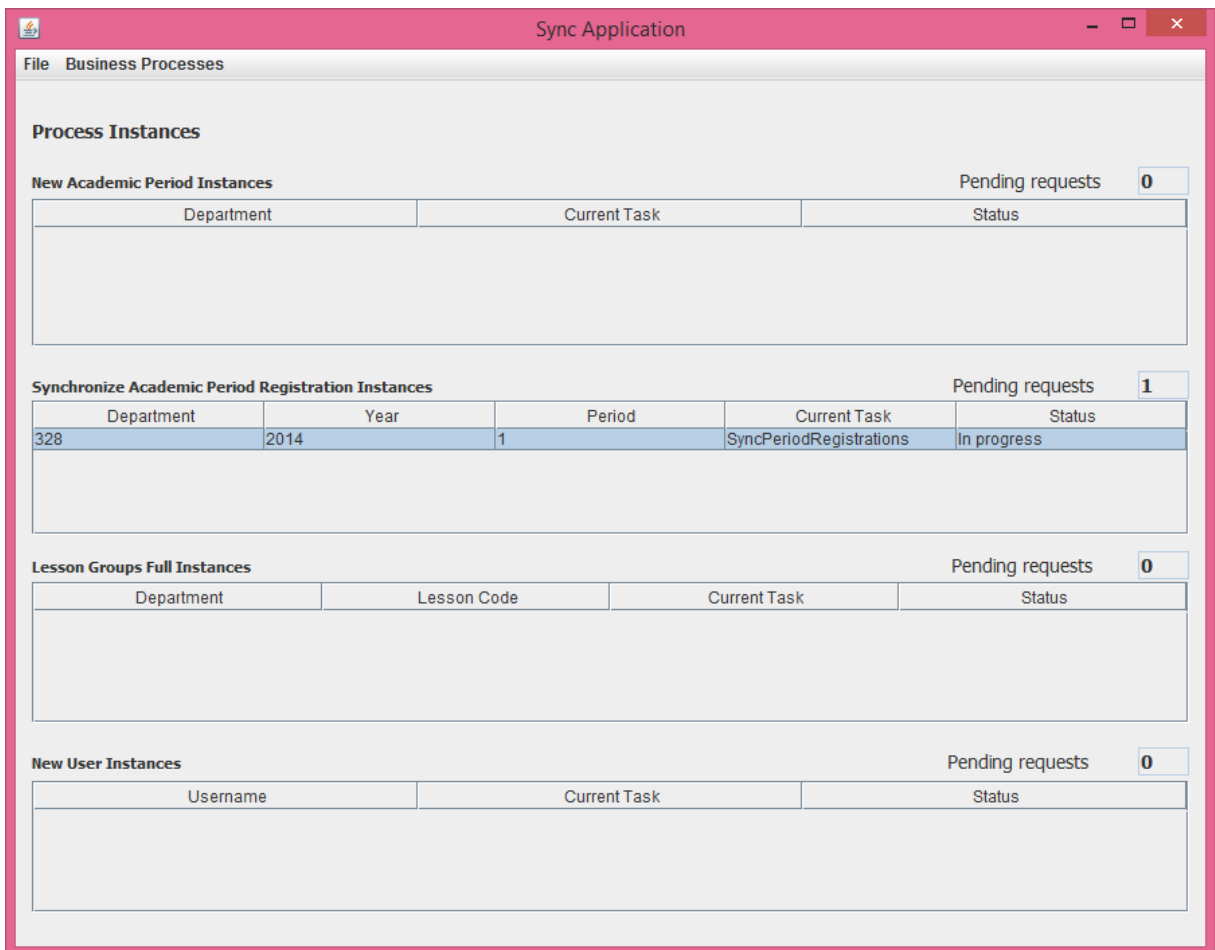
Στον πίνακα με το ιστορικό ακαδημαϊκών περιόδων τμήματος επιλέγει το εικονίδιο  στην ακαδημαϊκή περίοδο που επιθυμεί συγχρονισμό δηλώσεων. Με την επιλογή αυτή εκκινείται η διεργασία συγχρονισμού δηλώσεων για την επιλεγμένη ακαδημαϊκή περίοδο και αναμένεται από το διαχειριστή να κάνει το συγχρονισμό των δηλώσεων. Στην εικόνα 3.64 στη στήλη εξέλιξη αιτήματος φαίνεται η εργασία (task) που εκκρεμεί να ολοκληρωθεί.

Ιστορικό ακαδημαϊκών περιόδων τμήματος

Κωδικός	Κωδικός τμήματος	Ακαδημαϊκό Έτος	Περίοδος	Αίτημα συγχρονισμού δηλώσεων	Εξέλιξη αιτήματος
32820142	328	2014	2		
32820141	328	2014	1		SyncPeriodRegistrations
32820132	328	2013	2		
32820131	328	2013	1		

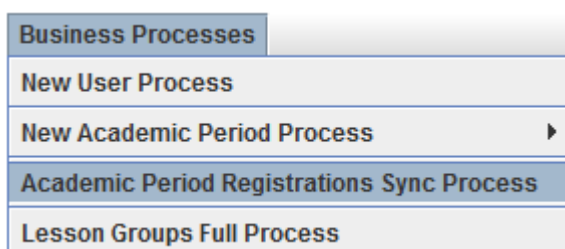
Εικόνα 3.64: Εκκίνηση διεργασίας SyncPeriodRegistrations

Ο διαχειριστής από το υποσύστημα συγχρονισμού θα έχει ένδειξη πως εκκρεμεί η παραπάνω εργασία, όπως φαίνεται στην εικόνα 3.65.



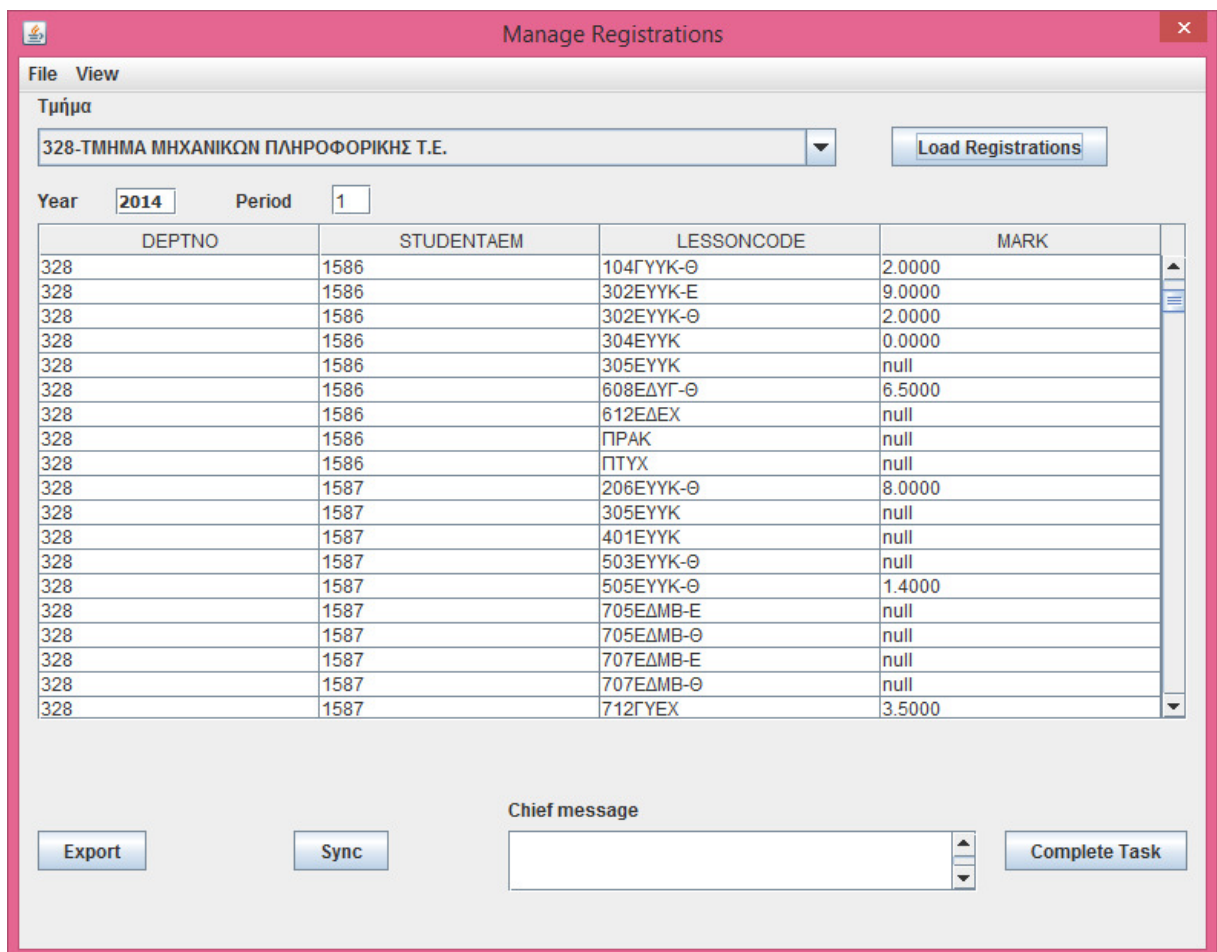
Εικόνα 3.65: Προβολή τρέχουσας εργασίας και κατάστασης διεργασίας

Επιλέγει από το μενού *Business Processes* την επιλογή *Academic Period Registrations Sync Process* όπως φαίνεται στην εικόνα 3.66 .



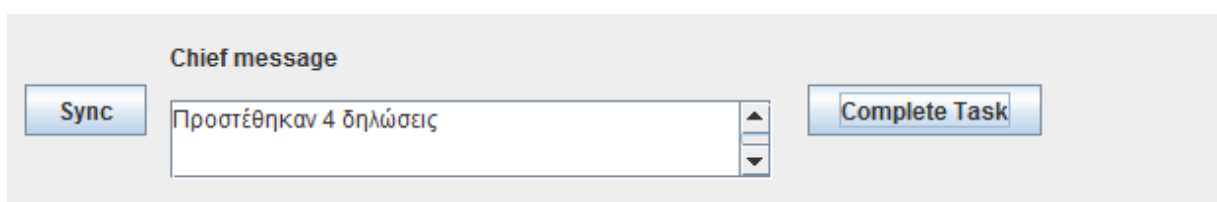
Εικόνα 3.66: Επιλογή συγχρονισμού δηλώσεων ακαδημαϊκής περιόδου τμήματος

Φορτώνει τις δηλώσεις για την ακαδημαϊκή περίοδο όπως φαίνεται στην εικόνα 3.67, και επιλέγει *Export, Sync*.



Εικόνα 3.67: Φόρμα συγχρονισμού δηλώσεων μαθημάτων ακαδημαϊκής περιόδου τμήματος

Μετα την επιλογή *Sync* εμφανίζονται οι δηλώσεις που προστέθηκαν στο πλαίσιο κειμένου *Chief message* για να ενημερωθεί ο προϊστάμενος του τμήματος όπως φαίνεται στην εικόνα 3.68.



Εικόνα 3.68: Συγχρονισμός δηλώσεων και ενημέρωση προϊσταμένου

Τέλος ο διαχειριστής επιλέγει *Complete Task* για να σηματοδοτηθεί το τέλος της εργασίας.

Ο προϊστάμενος ειδοποιείται με e-mail όπως δείχνει η εικόνα 3.69.

test1@teiemt.gr 10:36 π.μ. (Πριν από 25 λεπτά) ☆

προς Εμένα ▾

Προστέθηκαν 4 δηλώσεις

Εικόνα 3.69: Παραλαβή e-mail από τον προϊστάμενο

Ο προϊστάμενος μπορεί να διαπιστώσει πως ο συγχρονισμός πραγματοποιήθηκε με επιτυχία αν επιλέξει την επεξεργασία κάποιου μαθήματος και προσέξει τον πίνακα με το ιστορικό επιτυχιών ανά ακαδημαϊκή περίοδο, όπως φαίνεται στην εικόνα 3.70.

Επεξεργασία Μαθήματος

Παρακαλώ πληκτρολογήστε στοιχεία μαθήματος.

Κωδικός: ΔΟΚ-Θ

Τίτλος: ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ

Εξέμνηνο διδασκαλίας: Α ▾

Καθηγητής: ΚΑΘΗΓΗΤΗΣ Α ▾

Αίθουσα: Α1 ▾

Ενημέρωση

Προβολή ομάδων



ACAD_PERIOD_ID	Κωδικός Ομάδας	Ημέρα	Ώρες	Χωρητικότητα	Δεσμευμένες θέσεις	Ελεύθερες θέσεις	Επεξεργασία	Διαγραφή
32820142	E1	ΤΡΙΤΗ ▾	9:00-10:00	15	2	13		
32820142	ΠΛΕΟΝΑΖΟΝΤΕΣ	- ▾	8:00-10:00	999	0	999		
32820142	ΤΕΛΙΚΗ ΕΞΕΤΑΣΗ	- ▾	8:00-10:00	999	0	999		

Ιστορικό επιτυχιών ανά ακαδημαϊκή περίοδο

Κωδικός	Κωδικός τμήματος	Ακαδημαϊκό Έτος	Περίοδος	Επιτυχίες	Αποτυχίες	Μη παρακολούθησαντες
32820142	328	2014	2	0	0	4
32820141	328	2014	1	2	1	1
32820132	328	2013	2	0	0	0
32820131	328	2013	1	0	0	0

Εικόνα 3.70: Επιβεβαίωση συγχρονισμού από τον προϊστάμενο

Περίπτωση χρήσης διεργασίας κορεσμού ομάδων παρακολούθησης ενός μαθήματος

Από το υποσύστημα φοιτητών ο φοιτητής μη έχοντας τη δυνατότητα να δηλώσει ομάδα παρακολούθησης -επειδή όλες οι ομάδες έχουν γεμίσει- δηλώνει την ομάδα ΠΛΕΟΝΑΖΟΝΤΕΣ όπως φαίνεται στην εικόνα 3.71.

▼ Στοιχεία Φοιτητή

AM
1

Επίσημο
ΦΟΙΤΗΤΗΣ 1

Όνομα
ΣΠΟΥΔΑΣΤΗΣ 1

Εξάμηνο
1

Δήλωση ομάδας παρακολούθησης

Έχετε επιλέξει να δηλώσετε ομάδα παρακολούθησης για το μαθημα:

Κωδικός Μαθήματος	ΔΟΚ-Θ
Τίτλος	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ
Εξάμηνο	1

Στη φόρμα που ακολουθεί μπορείτε να επιλέξετε την ομάδα παρακολούθησης που επιθυμείτε.

► Ακαδημαϊκή περίοδος

Ομάδες παρακολούθησης

Κωδικός Ομάδας	Ημέρα	Ώρες	Δεσμευμένες θέσεις	Ελεύθερες θέσεις	Επιλογή
E1	2	9:00-10:00	1	0	+
ΠΛΕΟΝΑΖΟΝΤΕΣ	0	8:00-10:00	0		+
ΤΕΛΙΚΗ ΕΞΕΤΑΣΗ	0	8:00-10:00	0		+

Εικόνα 3.71: Δήλωση ομάδας παρακολούθησης ΠΛΕΟΝΑΖΟΝΤΕΣ από φοιτητή

Με την επιλογή της ομάδας ΠΛΕΟΝΑΖΟΝΤΕΣ εκκινείται η διεργασία διαχείρισης κορεσμού ομάδων παρακολούθησης και εμφανίζεται το popup παράθυρο που φαίνεται στην εικόνα 3.72.

Υποστήριξη προγράμματος σπουδών – Υποσύστημα Φοιτητή

An alert sent to the teacher for solving this inconvenience. Please be alerted.

▼ Στοιχεία Φοιτητή

AM
1

Επίσημο
ΦΟΙΤΗΤΗΣ 1

Όνομα
ΣΠΟΥΔΑΣΤΗΣ 1

Εξάμηνο
1

► Ακαδημαϊκή περίοδος

Βρίσκεστε σε περίοδο δηλώσεων ομάδων παρακολούθησης

Μαθήματα

Στους παρακάτω πίνακες εμφανίζονται μαθήματα που έχει δηλώσει ο φοιτητής στο σύστημα της Ηλεκτρονικής Γραμματείας. Στον πρώτο πίνακα εμφανίζονται μαθήματα στα οποία δεν έχετε επιλέξει μέχρι στιγμής ομάδα παρακολούθησης. Στον δεύτερο πίνακα εμφανίζονται μαθήματα στα οποία έχετε επιλέξει ομάδα παρακολούθησης και τότε

Μαθήματα με αδήλωτη ομάδα παρακολούθησης

Κωδικός Μαθήματος	Τίτλος	Εξάμηνο	Επιλογή
ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ	1	

Μαθήματα με δηλωμένη ομάδα παρακολούθησης

Κωδικός Μαθήματος	Τίτλος	Εξάμηνο	Κωδικός Ομάδας	Ημέρα	Ώρες	Διαγραφή
ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ	1	ΠΛΕΟΝΑΖΟΝΤΕΣ	-	8-10	⊖

Εικόνα 3.72: Ένδειξη στο φοιτητή εκκίνησης της διεργασίας LessonGroupsFull

Ο υπεύθυνος εκπαιδευτικός ενημερώνεται πως οι ομάδες παρακολούθησης του μαθήματος του έχουν γεμίσει. Η ενημέρωση γίνεται αυτόματα με e-mail που φαίνεται στην εικόνα 3.73.

Inform Teacher Εισερχόμενα x

test1@teiemt.gr 9:24 μ.μ. (Πριν από 0 λεπτά) ☆

προς Εμένα ▾

The lesson: ΔΟΚ-Θ of the Department: 328 is full
Please make any necessary actions

Εικόνα 3.73: Παραλαβή e-mail από υπεύθυνο εκπαιδευτικό

Αφού δει το e-mail του ο υπεύθυνος εκπαιδευτικός πρέπει να επιλέξει επεξεργασία του μαθήματος που φαίνεται στον e-mail (ΔΟΚ-Θ) και θα του εμφανιστεί η σελίδα της εικόνας 3.74. Εκεί ο υπεύθυνος εκπαιδευτικός μπορεί να δει την εκκρεμότητα της διεργασίας κορεσμού και έχει τη δυνατότητα να απορρίψει το αίτημα ή να το προωθήσει στον προϊστάμενο του τμήματος.

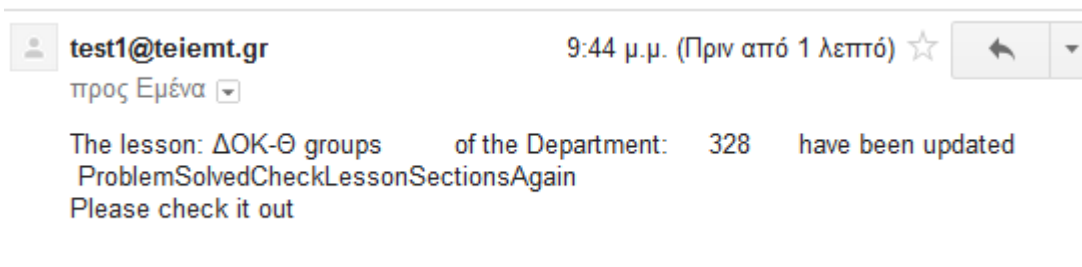
ACAD_PERIOD_ID	Κωδικός Ομάδας	Ημέρα	Ώρες	Κωριτικότητα	Δεσμευμένες θέσεις	Ελεύθερες θέσεις	Επεξεργασία	Διαγραφή
32820142	Ε1	ΤΡΙΤΗ	9.00-10.00	1	1	0	🔴	🔴
32820142	ΠΛΕΟΝΑΖΟΝΤΕΣ	-	8.00-10.00	999	1	998	🔴	🔴
32820142	ΤΕΛΙΚΗ ΕΞΕΤΑΣΗ	-	8.00-10.00	999	0	999	🔴	🔴

Εικόνα 3.74: Ένδειξη εκκρεμούς διεργασίας κορεσμού ομάδων παρακολούθησης μαθήματος

Αν ο υπεύθυνος εκπαιδευτικός επιλέξει προώθηση αιτήματος τότε ο προϊστάμενος πρέπει να επιλέξει επεξεργασία του μαθήματος και θα εμφανιστεί η κατάσταση της διεργασίας που φαίνεται στην εικόνα 3.75.

Εικόνα 3.75: Αποδοχή ή απόρριψη αιτήματος από προϊστάμενο

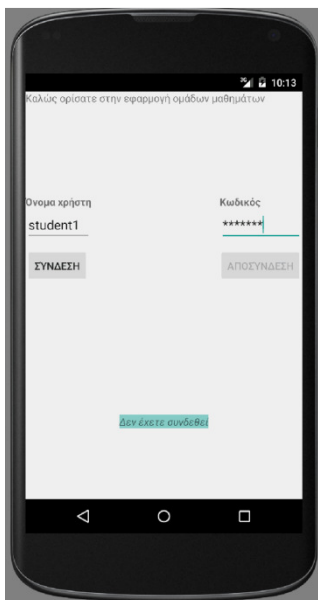
Αν ο προϊστάμενος επιλέξει αποδοχή αιτήματος (δεδομένου ότι έχει τροποποιήσει τις ομάδες παρακολούθησης του μαθήματος) αποστέλλεται ενημερωτικό e-mail (εικόνα 3.76) στον υπεύθυνο εκπαιδευτικό και στους φοιτητές που έχουν δηλώσει το μάθημα



Εικόνα 3.76: Παραλαβή ενημερωτικού e-mail σε υπεύθυνο καθηγητή και φοιτητές

3.1.6 Διεπαφή υποσυστήματος Android

Στην εικόνα 3.77 εμφανίζεται η αρχική οθόνη της εφαρμογής Android. Ο φοιτητής πληκτρολογεί τα στοιχεία σύνδεσης και επιλέγει το κουμπί **ΣΥΝΔΕΣΗ**.



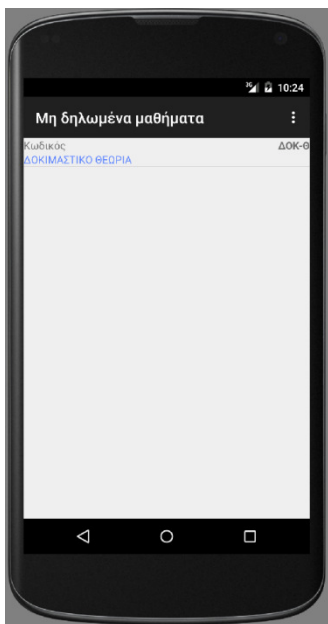
Εικόνα 3.77: Αρχική οθόνη

Μετά από επιτυχημένη είσοδο εμφανίζεται η κύρια οθόνη (Εικόνα 3.78) όπου εμφανίζονται τα στοιχεία του φοιτητή (μητρώο, επώνυμο, όνομα, εξάμηνο, κωδικός τμήματος) και μπορεί να επιλέξει προβολή μαθημάτων με δηλωμένη ή αδήλωτη ομάδα παρακολούθησης.



Εικόνα 3.78: Κύρια οθόνη

Όταν επιλέξει το κουμπί *ΑΔΗΛΩΤΑ ΜΑΘΗΜΑΤΑ* εμφανίζονται τα μαθήματα στα οποία ο φοιτητής δεν έχει δηλώσει ομάδα παρακολούθησης, όπως φαίνεται στην εικόνα 3.79.



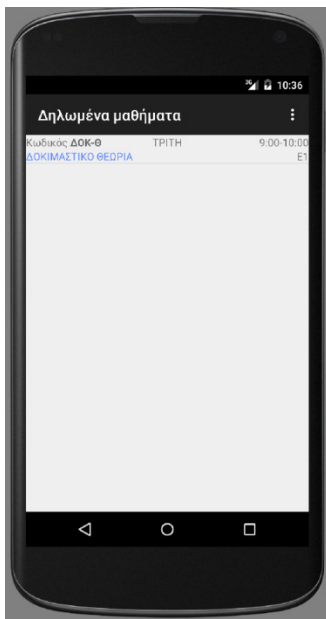
Εικόνα 3.79: Οθόνη μαθημάτων με αδήλωτες ομάδες παρακολούθησης

Όταν επιλέξει κάποιο από τα μη δηλωμένα μαθήματα εμφανίζεται η επομένη οθόνη (Εικόνα 3.80) ώστε να διαλέξει ομάδα παρακολούθησης.



Εικόνα 3.80: Οθόνη δήλωσης ομάδων παρακολούθησης μαθήματος

Όταν από την κύρια οθόνη επιλέξει το κουμπί *ΔΗΛΩΜΕΝΑ ΜΑΘΗΜΑΤΑ* εμφανίζονται τα μαθήματα (και η ομάδα παρακολούθησης) στα οποία ο φοιτητής έχει δηλώσει ομάδα παρακολούθησης, όπως φαίνεται στην εικόνα 3.81.



Εικόνα 3.81: Οθόνη μαθημάτων με δηλωμένες ομάδες παρακολούθησης

3.2 Υλοποίηση βάσης δεδομένων

Στις επόμενες παραγράφους περιγράφονται οι απαιτήσεις-περιορισμοί της βάσης δεδομένων του συστήματος, η μοντελοποίηση της βάσης, οι όψεις, οι αποθηκευμένες διαδικασίες, συναρτήσεις και σκανδαλισμοί. Για τη μοντελοποίηση χρησιμοποιήσαμε δυο συμβολισμούς: Συμβολισμό Elmasri-Navathe [27] και Διάγραμμα UML.

3.2.1 Απαιτήσεις-Περιορισμοί

Το σύστημα υποστήριξης προγραμμάτων σπουδών θα αποθηκεύει πληροφορίες που αφορούν μόνο ένα εκπαιδευτικό ίδρυμα. Σε κάθε ίδρυμα οι πληροφορίες που ενδιαφέρουν είναι τα τμήματα, οι αίθουσες κάθε τμήματος, οι εκπαιδευτικοί, οι φοιτητές, τα μαθήματα, οι ομάδες παρακολούθησης ανά μάθημα, οι ακαδημαϊκές περίοδοι, οι δηλώσεις μαθημάτων και οι δηλώσεις ομάδων παρακολούθησης μαθημάτων.

Κάθε τμήμα έχει μοναδικό κωδικό, όνομα, διεύθυνση ιστοσελίδας, γεωγραφικό πλάτος, γεωγραφικό μήκος και τη διεύθυνση ηλεκτρονικής αλληλογραφίας της γραμματείας που το εξυπηρετεί. Κάθε τμήμα περιλαμβάνει στο πρόγραμμα σπουδών του αρκετά μαθήματα, αίθουσες διδασκαλίας, εκπαιδευτικούς, φοιτητές και ακαδημαϊκές περιόδους.

Κάθε αίθουσα έχει όνομα, διαθέσιμες θέσεις και ανήκει αποκλειστικά σε ένα τμήμα.

Κάθε εκπαιδευτικός έχει επώνυμο, όνομα, διεύθυνση ηλεκτρονικής αλληλογραφίας, όνομα λογαριασμού και κωδικό λογαριασμού. Επιπλέον ένας εκπαιδευτικός μπορεί να είναι προϊστάμενος ενός τμήματος ή να έχει οριστεί ως διαχειριστής του συνόλου του συστήματος. Ακόμα ένας εκπαιδευτικός μπορεί να έχει οριστεί ως υπεύθυνος σε ένα ή περισσότερα μαθήματα ενός τμήματος. Ένας εκπαιδευτικός ανήκει αποκλειστικά σε ένα μόνο τμήμα.

Κάθε φοιτητής έχει αριθμό μητρώου τμήματος, επώνυμο, όνομα, διεύθυνση ηλεκτρονικής αλληλογραφίας, τρέχον εξάμηνο φοίτησης, όνομα λογαριασμού και κωδικό λογαριασμού. Κάθε φοιτητής ανήκει σε ένα μόνο τμήμα και μπορεί να δηλώνει μαθήματα και ομάδες παρακολούθησης μαθημάτων σε διαφορετικές ακαδημαϊκές περιόδους. Φοιτητές διαφορετικών τμημάτων μπορεί να έχουν τον ίδιο αριθμό μητρώου.

Κάθε μάθημα έχει κωδικό, τίτλο, εξάμηνο διδασκαλίας και πλήθος ομάδων παρακολούθησης. Ένα μάθημα μπορεί να παραμετροποιητή με τέτοιο τρόπο ώστε οι φοιτητές που το έχουν παρακολουθήσει ανεπιτυχώς σε παρελθοντική ακαδημαϊκή περίοδο να μη μπορούν να δηλώσουν κανονική ομάδα παρακολούθησης, αλλά μόνο την ειδική ομάδα παρακολούθησης ΤΕΛΙΚΗ ΕΞΕΤΑΣΗ. Κάθε μάθημα μπορεί να διεξαχθεί μόνο σε μια αίθουσα και έχει υπεύθυνο έναν μόνο εκπαιδευτικό.

Κάθε ομάδα παρακολούθησης ανήκει σε ένα μόνο μάθημα, διεξάγεται συγκεκριμένη μέρα και ώρες (με ακρίβεια λεπτού) της εβδομάδας και έχει περιορισμένες θέσεις. Οι ομάδες παρακολούθησης μπορεί να αλλάζουν σε διαφορετική ακαδημαϊκή περίοδο.

Κάθε ακαδημαϊκή περίοδος τμήματος έχει έτος και περίοδο. Στην ακαδημαϊκή περίοδο ορίζονται οι ημερομηνίες έναρξης-λήξης δηλώσεων ομάδων παρακολούθησης και οι ημερομηνίες έναρξης-λήξης κριτηρίου προτεραιότητας εξαμήνου.

Ένα μάθημα μπορεί να δηλώνεται από πολλούς φοιτητές σε διαφορετικές ακαδημαϊκές περιόδους. Σε κάθε δήλωση ανά περίοδο ενδιαφέρει η βαθμολογία του φοιτητή στο μάθημα.

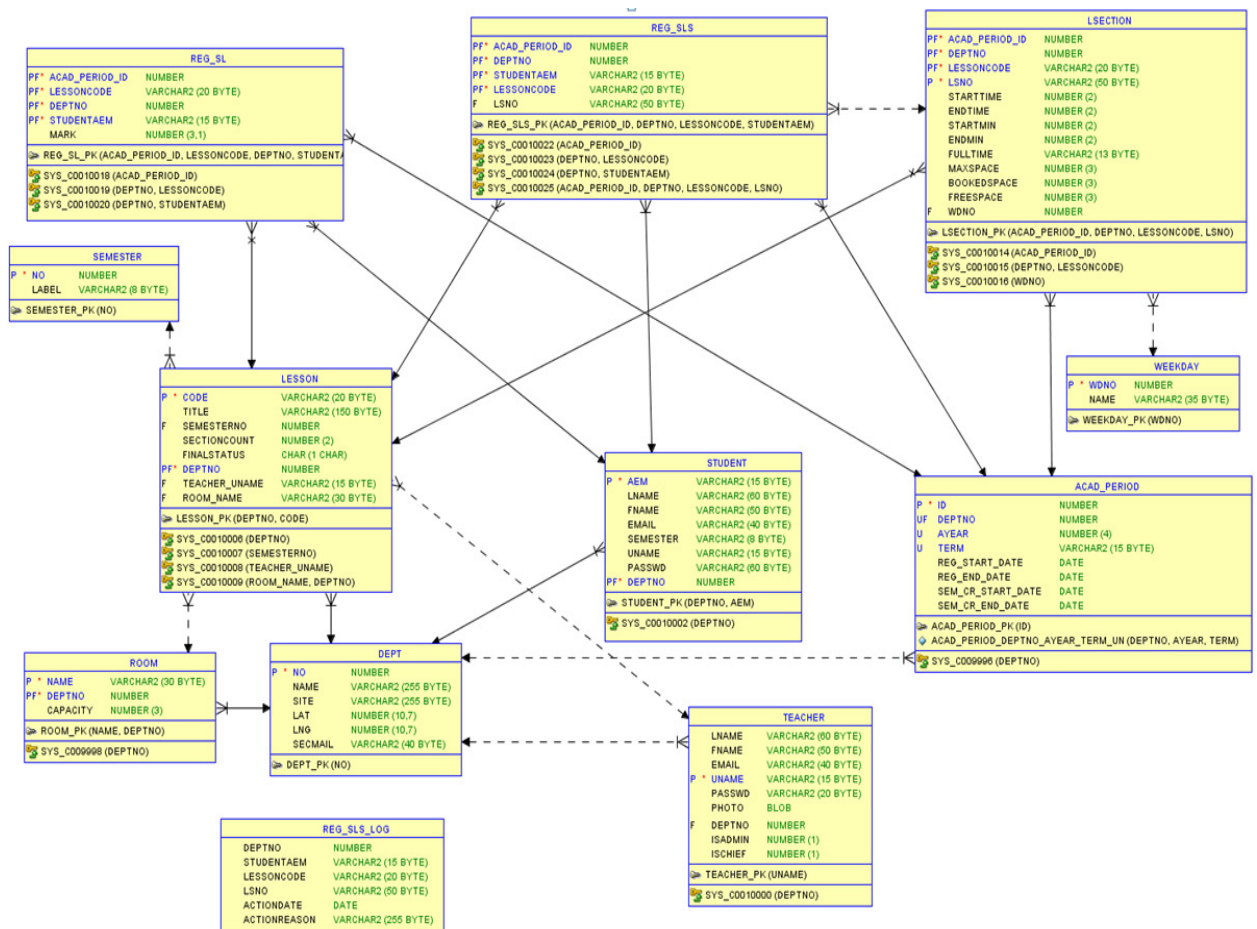
Μια ομάδα παρακολούθησης ενός μαθήματος μπορεί να δηλώνεται από πολλούς φοιτητές, αλλά μια μόνο φορά ανά φοιτητή και σε μια μόνο ακαδημαϊκή περίοδο. Σε κάθε ακαδημαϊκή περίοδο κατασκευάζονται σε κάθε μάθημα διαφορετικές ομάδες παρακολούθησης.

Ενδιαφέρει να υπάρχει αρχείο καταγραφής των δηλώσεων ομάδων παρακολούθησης των φοιτητών, για την αντιμετώπιση παράπονων των φοιτητών. Στο αρχείο καταγραφής θα καταγράφονται οι ενέργειες εισαγωγή και διαγραφή των δηλώσεων.

3.2.2 Μοντέλο Οντοτήτων Συσχετίσεων (ERD)

Το ακόλουθο διάγραμμα (Σχήμα 3.1) μοντελοποιεί σε συμβολισμό Elmasri-Navathe [27] τη βάση δεδομένων του συστήματος λαμβάνοντας υπόψη τις απαιτήσεις-περιορισμούς της προηγούμενης παραγράφου.

Developer [28]. Με την επιλογή Data Modeler το UML διάγραμμα κατασκευάστηκε αυτόματα από την ήδη υλοποιημένη βάση δεδομένων του συστήματος.

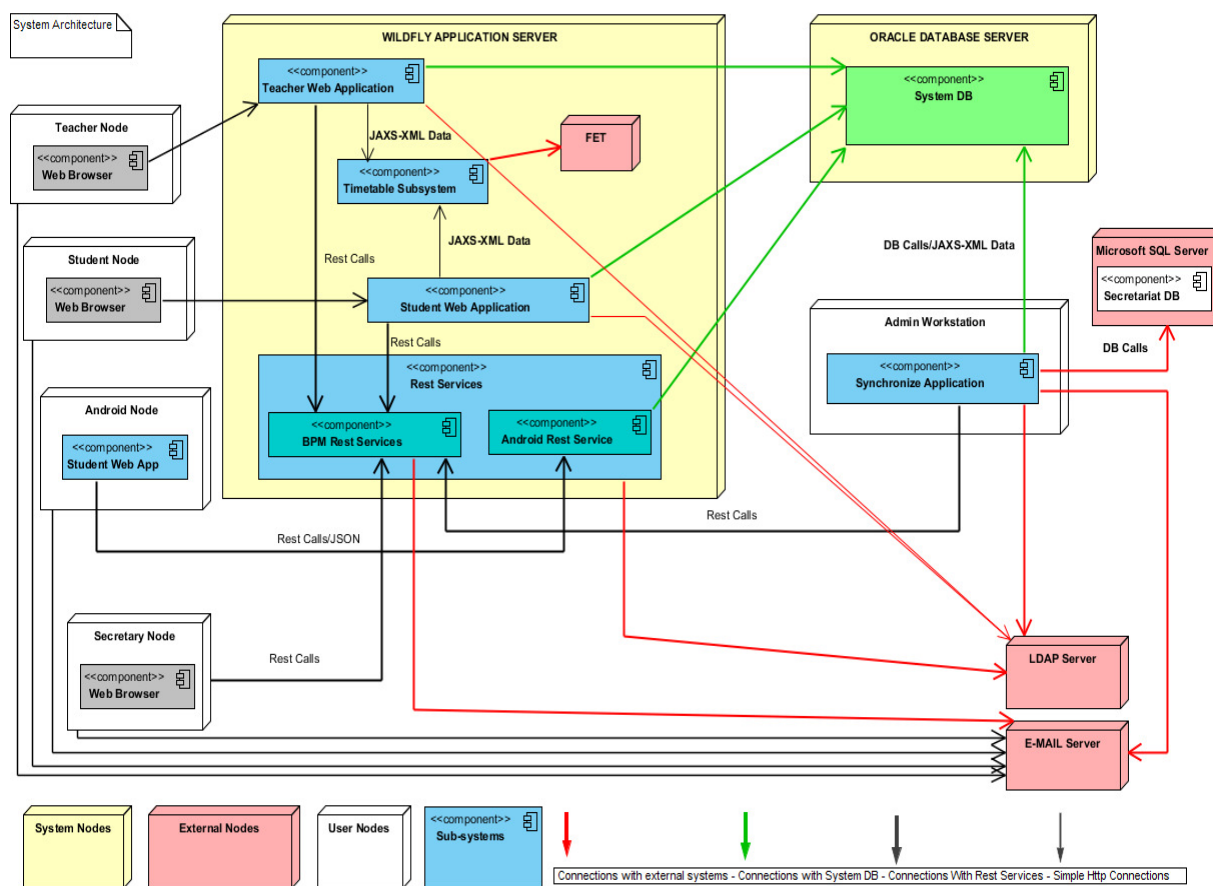


Σχήμα 3.2: Διάγραμμα UML

Στο παράρτημα Β παρουσιάζονται με λεπτομέρεια οι πίνακες, οι όψεις, δείγμα δεδομένων των πινάκων και των όψεων, οι αποθηκευμένες διαδικασίες, οι αποθηκευμένες συναρτήσεις και οι σκανδαλισμοί της βάσης δεδομένων που υλοποιήσαμε.

3.3 Αρχιτεκτονική συστήματος

Η αρχιτεκτονική του συστήματος που φαίνεται στο σχήμα 3.3 δείχνει τους κόμβους του συστήματος, των χρηστών, των εξωτερικών υποσυστημάτων και τη διασύνδεση που έχουν με τα εσωτερικά υποσυστήματα. Οι κόμβοι των χρηστών (**User Nodes**) απεικονίζονται σε ορθογώνιο παραλληλεπίπεδο με λευκό φόντο. Οι κόμβοι του συστήματος (**System Nodes**) απεικονίζονται σε ορθογώνιο παραλληλεπίπεδο με κίτρινο φόντο. Οι εξωτερικοί κόμβοι (**External Nodes**) απεικονίζονται σε ορθογώνιο παραλληλεπίπεδο με ροζ φόντο. Οι πράσινες γραμμές διασυνδέουν τα εσωτερικά υποσυστήματα με τη βάση δεδομένων (System DB). Οι κόκκινες γραμμές διασυνδέουν τα εσωτερικά υποσυστήματα με τα εξωτερικά υποσυστήματα. Οι μαύρες γραμμές διασυνδέουν το υποσύστημα παροχής διεργασιών και Android back-end με τα εσωτερικά υποσυστήματα. Οι γκρι γραμμές δείχνουν απλή HTTP επικοινωνία. Στις επόμενες παραγράφους περιγράφονται οι επιμέρους κόμβοι της αρχιτεκτονικής.



Σχήμα 3.3: Αρχιτεκτονική συστήματος

3.3.1 Κόμβοι χρηστών

Ο κόμβος του εκπαιδευτικού (**Teacher Node**) ανοίγει οποιοδήποτε πρόγραμμα περιήγησης (Web Browser) για να χρησιμοποιήσει την εφαρμογή του εκπαιδευτικού (Teacher Web Application) που βρίσκεται εγκατεστημένη στον κόμβο-εξυπηρέτη (WILDFLY APPLICATION SERVER). Ο εκπαιδευτικός λαμβάνει e-mail από τον κόμβο-E-MAIL Server κατά την εκτέλεση διεργασιών BPM που σχετίζονται με αυτόν. Ο κόμβος του εκπαιδευτικού μπορεί να χρησιμοποιηθεί από τους χρήστες εκπαιδευτικός, προϊστάμενος και διαχειριστής.

Ο φοιτητής μπορεί να χρησιμοποιήσει το σύστημα μέσα από δυο κόμβους. Στην κλασική περίπτωση (**Student Node**) ο φοιτητής ανοίγει οποιοδήποτε πρόγραμμα περιήγησης (Web Browser) για να χρησιμοποιήσει την εφαρμογή του φοιτητή (Student Web Application) που βρίσκεται εγκατεστημένη στον κόμβο-εξυπηρετητή (WILDFLY APPLICATION SERVER).

Στην περίπτωση που ο φοιτητής διαθέτει συσκευή Android (**Android Node**) μπορεί να εγκαταστήσει στη συσκευή του το υποσύστημα Student Web App που επικοινωνεί με το Android Rest Service.

Ο φοιτητής λαμβάνει e-mail από τον κόμβο-E-MAIL Server κατά την εκτέλεση διεργασιών BPM που σχετίζονται με αυτόν.

Ο κόμβος της γραμματείας (**Secretary Node**) επικοινωνεί με το σύστημα αποστέλλοντας και λαμβάνοντας ηλεκτρονικά μηνύματα μέσω E-MAIL Server.

Ο κόμβος του διαχειριστή (**Admin Workstation**) έχει εγκατεστημένο το υποσύστημα συγχρονισμού (Synchronize Application) που επικοινωνεί με τη βάση δεδομένων του συστήματος (System DB), με το υποσύστημα παροχής διεργασιών (BPM Rest Services), με τη βάση δεδομένων (Secretariat DB) του εξωτερικού υποσυστήματος Microsoft SQL Server (χρησιμοποιώντας τη βιβλιοθήκη **jTDS [29]**), με τον LDAP Server και με τον E-MAIL Server.

3.3.2 Κόμβοι συστήματος

Ο κόμβος εξυπηρετητής εφαρμογών (**WILDFLY APPLICATION SERVER**) φιλοξενεί την εφαρμογή του εκπαιδευτικού (Teacher Web Application), την εφαρμογή του φοιτητή (Student Web Application), το υποσύστημα ωρολογίου προγράμματος (Timetable Subsystem), το

εξωτερικό υποσύστημα FET, το υποσύστημα παροχής διεργασιών (BPM Rest Services) και το back-end του υποσυστήματος Android (Android Rest Service).

Ο εξυπηρετητής διασυνδέει την εφαρμογή του εκπαιδευτικού (Teacher Web Application), του φοιτητή (Student Web Application) και του back-end του υποσυστήματος Android (Android Rest Service) με τη βάση δεδομένων του συστήματος (System DB) που βρίσκεται εγκατεστημένη στο σύστημα διαχείρισης βάσεων δεδομένων (κόμβος ORACLE DATABASE SERVER).

Ο εξυπηρετητής διασυνδέει επίσης την εφαρμογή του εκπαιδευτικού (Teacher Web Application), την εφαρμογή του φοιτητή (Student Web Application) και το υποσύστημα παροχής διεργασιών (BPM Rest Services) με τον LDAP Server του ιδρύματος.

Η εφαρμογή του εκπαιδευτικού και του φοιτητή επικοινωνούν με το υποσύστημα ωρολογίου προγράμματος ανταλλάσσοντας δεδομένα σε μορφή XML χρησιμοποιώντας την τεχνολογία JAXS-RS [26].

Το υποσύστημα παροχής διεργασιών (BPM Rest Services) και το Android back-end (Android Rest Service) αποτελούν -στην ουσία- ένα υποσύστημα παροχής υπηρεσιών REST.

Ο κόμβος – εξυπηρετή βάσεων δεδομένων (**ORACLE DATABASE SERVER**) φιλοξένει τη βάση δεδομένων του συστήματος και δέχεται ερωτήματα (SQL Queries) από τα υποσυστήματα εκπαιδευτικού, φοιτητή, Android backend και συγχρονισμού.

3.3.3 Εξωτερικοί κόμβοι

Ο κόμβος **Microsoft SQL Server** διαθέτει τη βάση δεδομένων του γραμματειακού συστήματος και δέχεται αιτήματα αναζήτησης (SQL Select Queries) από το υποσύστημα συγχρονισμού (Synchronize Application).

Ο κόμβος **LDAP Server** περιέχει τους χρήστες του ιδρύματος. Τα υποσυστήματα εκπαιδευτικού, φοιτητή και παροχής διεργασιών κάνουν αιτήματα αυθεντικοποίησης στον LDAP Server. Το υποσύστημα συγχρονισμού κάνει αιτήματα αναζήτησης και εισαγωγής λογαριασμών στον LDAP Server. Όλα τα υποσυστήματα χρησιμοποιούν τη βιβλιοθήκη της **Novell** [24] για να επικοινωνήσουν με τον LDAP Server.

Ο κόμβος **FET** περιλαμβάνει το εξωτερικό υποσύστημα που χρησιμοποιείται για την δημιουργία του ωρολογίου προγράμματος. Το υποσύστημα ωρολογίου προγράμματος (Timetable Subsystem) χρησιμοποιεί αυτόν τον κόμβο για να στείλει παραμέτρους εισόδου και να παραλάβει το παραγόμενο ωρολόγιο πρόγραμμα σε XML μορφή.

Ο κόμβος **E-MAIL Server** περιέχει του λογαριασμούς ηλεκτρονικής αλληλογραφίας του ιδρύματος. Ο κόμβος αυτός χρησιμοποιείται για ανταλλαγή μηνυμάτων από όλους τους κόμβους χρηστών του συστήματος καθώς και από το υποσύστημα παροχής διεργασιών (BPM Rest Services).

3.4 Διαδικασία εγκατάστασης συστήματος

Η εγκατάσταση του συστήματος είναι μια διαδικασία που περιλαμβάνει την εγκατάσταση του συστήματος διαχείρισης βάσεων δεδομένων (ΣΔΒΣ) -μαζί με τη δημιουργία της βάσης του συστήματος-, την εγκατάσταση του εξυπηρέτη εφαρμογών (Application Server) που εμπεριέχει τα υποσύστημα φοιτητών, εκπαιδευτικών, παροχής διεργασιών και Android back-end, και της εφαρμογής Android των φοιτητών. Σε αυτή την ενότητα περιγράφουμε περιληπτικά τη διαδικασία εγκατάστασης καθώς και τα εργαλεία ανάπτυξης και σχεδίασης που χρησιμοποιήσαμε.

Στο συνοδευτικό υλικό της μεταπτυχιακής διατριβής υπάρχει το αρχείο **Readme.txt** με βήμα προς βήμα οδηγίες εγκατάστασης ολόκληρου του συστήματος και της αρχικής παραμετροποίησής του.

3.4.1 Εγκατάσταση ΣΔΒΣ

Η εγκατάσταση του συστήματος διαχείρισης βάσεων δεδομένων περιλαμβάνει την εγκατάσταση του προϊόντος Oracle Database έκδοσης 12.1.0, τη δημιουργία ενός χρήστη βάσης δεδομένων και τη δημιουργία του σχήματος της βάσης του συστήματος υποστήριξης προγραμμάτων σπουδών. Τα βήματα είναι τα εξής:

-Κατεβάζουμε την έκδοση Oracle Database έκδοσης 12.1.0 από τη διεύθυνση <http://www.oracle.com/technetwork/indexes/downloads/> . Για δοκιμαστική χρήση μπορούμε εναλλακτικά να χρησιμοποιήσουμε την έκδοση Oracle 11g Express Edition.

-Δημιουργούμε το χρήστη C##scott με κωδικό tiger και με δικαιώματα CONNECT, RESOURCE, CREATE VIEW και φροντίζουμε ο χρήστης να έχει UNLIMITED TABLESPACE.

-Από το εργαλείο της SQL PLUS Oracle ή καλύτερα τον Oracle SQL Developer συνδεόμαστε ως χρήστης C##SCOTT και τρέχουμε το περιεχόμενο του αρχείου PROJECTS\PES700_Dissertation\database\CREATE.sql που βρίσκεται στο συνοδευτικό DVD της μεταπτυχιακής διατριβής.

3.4.2 Εγκατάσταση εξυπηρετητή εφαρμογών

Για την εγκατάσταση του εξυπηρετητή εφαρμογών αντιγράφουμε από το συνοδευτικό DVD το φάκελο wildfly-8.2.0.Final στο σκληρό δίσκο του συστήματος και εκτελούμε την εξής εντολής από command prompt:

```
c:\wildfly-8.2.0.Final\bin>standalone.bat -c standalone-full.xml -b=hostURL  
-bmanagement=localhost
```

3.4.3 Εγκατάσταση εφαρμογής Android φοιτητών

Για την εγκατάσταση της εφαρμογής Android τοποθετούμε το αρχείο εγκατάστασης StudentAndroidApp.apk -που βρίσκεται στο συνοδευτικό υλικό της μεταπτυχιακής διατριβής- στη συσκευή Android που επιθυμούμε και το κάνουμε εγκατάσταση.

3.4.4 Εργαλεία ανάπτυξης-υλοποίησης του συστήματος

Για τη σχεδίαση των UML διαγραμμάτων κλάσεων των υποσυστημάτων καθώς και του διαγράμματος της αρχιτεκτονικής του συστήματος χρησιμοποιήσαμε το εργαλείο **Visual Paradigm** έκδοσης 12.1.

Η βάση δεδομένων του συστήματος δημιουργήθηκε στο σύστημα διαχείρισης βάσεων δεδομένων (ΣΔΒΣ) της εταιρείας **ORACLE** έκδοσης 12g και η διασύνδεση του συστήματος με τη βάση έγινε με τη βιβλιοθήκη **ojdbc** έκδοσης 7. Για τη διαχείριση του ΣΔΒΔ χρησιμοποιήθηκαν τα εργαλεία της εταιρείας ORACLE, **SQL*PLUS** έκδοσης 12.1 και **SQL Developer** έκδοσης 4.1.

Για το διάγραμμα UML της βάσης δεδομένων χρησιμοποιήθηκε το εργαλείο **Data Modeler**, τμήμα του SQL Developer.

Για το διάγραμμα UML της βάσης δεδομένων χρησιμοποιήθηκε το εργαλείο **Data Modeler**, τμήμα του SQL Developer.

Για το μοντέλο οντοτήτων συσχετίσεων (ERD) σε συμβολισμό Elmasri-Navathe χρησιμοποιήθηκε το εργαλείο **SmartDraw CI** έκδοσης 11 της εταιρείας **SmartDraw**.

Για τη διασύνδεση του υποσυστήματος συγχρονισμού με τη βάση δεδομένων του γραμματειακού συστήματος που βρίσκεται στο προϊόν **Microsoft SQL Server 2012** χρησιμοποιήθηκε η βιβλιοθήκη **JTDS** έκδοσης 1.2.5.

Για τη διασύνδεση του συστήματος με τον **Open LDAP** (έκδοσης 3) Server χρησιμοποιήθηκε η βιβλιοθήκη **LDAP Classes for Java** της εταιρείας **Novell**.

Τα υποσυστήματα εφαρμογών διαδικτύου (φοιτητή, εκπαιδευτικού, παροχής διεργασιών και Android back-end) αναπτύχθηκαν στο περιβάλλον ανάπτυξης **NETBEANS 8 EE**, βασίζονται στον μεταγλωττιστή **JAVA EE** έκδοσης 7 και μεταφορτώθηκαν (έγιναν deploy) στον εξυπηρετητή διαδικτυακών εφαρμογών **Wildfly** έκδοσης 8.2.0 της εταιρείας **JBOSS**. Η αρχιτεκτονική αυτών των εφαρμογών βασίζεται στο πρότυπο **MVC (Model View Controller)**, ενώ στα υποσυστήματα φοιτητή και εκπαιδευτικού το περιβάλλον διεπαφής – τμήμα View- βασίστηκε στο πρότυπο **JSF 2.1** και χρησιμοποιήθηκε η σουίτα JSF αντικειμένων **PrimeFaces** έκδοσης 5.2. Επιπλέον στο υποσύστημα παροχής διεργασιών χρησιμοποιήθηκε η τεχνολογία Apache **Maven** 3.2.5 για τη δόμηση των πακέτων – βιβλιοθηκών, και οι βιβλιοθήκες **JBPM6** για την εκτέλεση των BPM διεργασιών.

Για τη σχεδίαση των διεργασιών BPM και την εξαγωγή των μοντέλων σε αρχεία XML (με επέκταση **.bpmn**) χρησιμοποιήθηκε το **Eclipse Kepler** έκδοσης 2.0 και η γλώσσα σχεδίασης των διεργασιών ήταν η **BPMN (Business Process Model and Notation) 2.0**.

Για τη διασύνδεση των υποσυστημάτων χρησιμοποιήθηκε η αρχιτεκτονική **REST, JAXS-RS, XML** και **JSON**. Οι βιβλιοθήκες τους είναι ενσωματωμένες στο μεταγλωττιστή Java EE 7.

Για αποστολή δοκιμαστικών αιτημάτων REST χρησιμοποιήθηκε το εργαλείο Google Chrome **Rest Console** έκδοσης 4.0.2.

Για την ανάπτυξη του υποσυστήματος Android φοιτητών χρησιμοποιήθηκε το περιβάλλον ανάπτυξης **Android Studio** έκδοσης 1.0.

Το (εξωτερικό) υποσύστημα δημιουργίας ωρολογίου προγράμματος **FET** έχει έκδοση 5.27.

Για την επεξεργασία της βιβλιογραφίας σε αυτό το βιβλίο χρησιμοποιήθηκε η εφαρμογή **Zotero** έκδοση 4.0.27.

Κεφάλαιο 4

Αξιολόγηση συστήματος

Για την αξιολόγηση του συστήματός ενδιαφερόμαστε για δυο άξονες: Ο πρώτος άξονας αφορά τη διαδικασία επαλήθευσης των ερευνητικών υποθέσεων που παρουσιάζονται στο κεφάλαιο της ερευνητικής πρότασης και ο δεύτερος άξονας ασχολείται με την αξιολόγηση του λογισμικού. Ωστόσο η διαδικασία επαλήθευσης των ερευνητικών υποθέσεων δε θα πραγματοποιηθεί στα πλαίσια της διατριβής.

4.1 Πλάνο επαλήθευσης ερευνητικών υποθέσεων

Για την επαλήθευση των ερευνητικών υποθέσεων θα κατασκευάσουμε ερωτηματολόγιο (για τη συγκέντρωση δεδομένων), θα πραγματοποιήσουμε ανάλυση των δεδομένων με χρήση των στατιστικών διαδικασιών και στη συνέχεια θα ερμηνεύσουμε τα αποτελέσματά τους. Στις επόμενες παραγράφους εξετάζουμε χωριστά τις υποθέσεις που περιλαμβάνονται στο κεφάλαιο διατύπωσης της ερευνητικής πρότασης, λαμβάνοντας υπόψη τη μεθοδολογία εκτέλεσης του πειράματος.

4.1.1 Υπόθεση 1

Σε αυτήν την υπόθεση ενδιαφερόμαστε να εξετάσουμε αν θα μειωθεί ο χρόνος οριστικοποίησης των ωρολογίων προγραμμάτων σε κάθε τμήμα με χρήση του νέου συστήματος σε σχέση με τη μη ύπαρξη συστήματος ή ύπαρξη οποιουδήποτε άλλου συστήματος.

Συγκέντρωση δεδομένων

Στο 1^ο ακαδημαϊκό έτος (χωρίς το νέο σύστημα) θα καταγραφεί από τον προϊστάμενο κάθε τμήματος ο χρόνος ενασχόλησής του για την κατασκευή των ωρολογίων προγραμμάτων των δυο ακαδημαϊκών εξαμήνων. Ο χρόνος ενασχόλησης θα καταγραφεί και στο 2^ο ακαδημαϊκό έτος αντίστοιχα με χρήση του νέου συστήματος.

Αφού συγκεντρωθούν τα στοιχεία θα καταρτιστεί πίνακας ανά τμήμα που θα περιλαμβάνει τους χρόνους ενασχόλησης των προϊσταμένων για τα δυο ακαδημαϊκά έτη. (Πίνακας 4.1)

Τμήμα	Χρόνος ενασχόλησης προϊσταμένου για δημιουργία ωρολογίων προγραμμάτων χωρίς χρήση συστήματος ή χρήση άλλου συστήματος	Χρόνος ενασχόλησης προϊσταμένου για δημιουργία ωρολογίων προγραμμάτων με χρήση του νέου συστήματος
Μηχανικών Πληροφορικής		
Λογιστική		
Μηχανολογία		
Διοίκηση Επιχειρήσεων		
Τεχνολογία Πετρελαίου		
Ηλεκτρολογία		
Αρχιτεκτονική Τοπίου		
Δασοπονία		
Νοσηλευτική		

Πίνακας 4.1: Χρόνοι ενασχόλησης για κατασκευή ωρολογίων προγραμμάτων ανά τμήμα.

Υπολογισμός στατιστικών διαδικασιών

Με βάση τα δεδομένα του πίνακα θα υπολογίσουμε τη μέση τιμή, τη μεσαία τιμή, τη διακύμανση, την τυπική απόκλιση των χρόνων με χρήση ή όχι του νέου συστήματος, το z-score κάθε τμήματος, τη συσχέτιση Pearson r μεταξύ των χρόνων με χρήση ή όχι του νέου συστήματος και το Effect size Δ .

Ερμηνεία των αποτελεσμάτων

Με βάση την υπόθεσή μας αναμένουμε αρνητική ισχυρή συσχέτιση. Για τη συσχέτιση *Pearson r* αναμένουμε τιμή που να πλησιάζει το -1, ενώ για το Effect Size (Δ) αναμένουμε τιμή που να πλησιάζει το -0.8.

Για το *z-score* αναμένουμε τιμή που να πλησιάζει το 0 καθώς αναμένουμε ελάχιστο χρόνο κατασκευής των ωρολογίων προγραμμάτων από τους προϊσταμένους για κάθε τμήμα με χρήση του νέου συστήματος και μεγάλο χρόνο ενασχόλησης χωρίς χρήση του νέου συστήματος.

4.1.2 Υπόθεση 2

Σε αυτήν την υπόθεση ενδιαφερόμαστε να εξετάσουμε αν θα μειωθούν τα παράπονα των φοιτητών κατά τη διαδικασία δήλωσης των ομάδων παρακολούθησης για κάθε τμήμα με χρήση του νέου συστήματος σε σχέση με τη μη ύπαρξη συστήματος ή ύπαρξη οποιουδήποτε άλλου συστήματος.

Συγκέντρωση δεδομένων

Με τη λήξη του 1^{ου} ακαδημαϊκού έτους, καθώς και με τη λήξη του 2^{ου} ακαδημαϊκού έτους θα συμπληρωθεί από τους φοιτητές η εξής ερώτηση:

Έχετε παράπονα από τον τρόπο που πραγματοποιήθηκαν οι δηλώσεις αυτό το ακαδημαϊκό έτος; Αξιολογήστε το μέγεθος των παραπόνων σας με βάση την παρακάτω κλίμακα:

0-Κανένα παράπονο, 1-Ελάχιστα παράπονα, 2-Μέτρια παράπονα, 3-Αρκετά παράπονα, 4-Πολλά παράπονα, 5-Απίστευτα παράπονα

Αφού συγκεντρωθούν τα στοιχεία θα καταρτιστεί πίνακας ανά τμήμα που θα περιλαμβάνει τη μέση τιμή των παραπόνων με βάση τις απαντήσεις των φοιτητών. (Πίνακας 4.2)

Τμήμα	Μέση τιμή παραπόνων χωρίς χρήση συστήματος ή χρήση άλλου συστήματος	Μέση τιμή παραπόνων με χρήση του νέου συστήματος
--------------	--	---

Μηχανικών Πληροφορικής		
Λογιστική		
Μηχανολογία		
Διοίκηση Επιχειρήσεων		
Τεχνολογία Πετρελαίου		
Ηλεκτρολογία		
Αρχιτεκτονική Τοπίου		
Δασοπονία		
Νοσηλευτική		

Πίνακας 4.2: Παράπονα φοιτητών ανά τμήμα.

Υπολογισμός στατιστικών διαδικασιών

Με βάση τα δεδομένα του πίνακα θα υπολογίσουμε τη μέση τιμή, τη μεσαία τιμή, τη διακύμανση, την τυπική απόκλιση των παραπόνων με χρήση ή όχι του νέου συστήματος, το *z-score* κάθε τμήματος, τη συσχέτιση *Pearson r* μεταξύ των παραπόνων με χρήση ή όχι του νέου συστήματος και το *Effect size Δ*.

Ερμηνεία των αποτελεσμάτων

Με βάση την υπόθεση μας αναμένουμε αρνητική ισχυρή συσχέτιση. Για τη συσχέτιση *Pearson r* αναμένουμε τιμή που να πλησιάζει το -1, ενώ για το *Effect Size (Δ)* αναμένουμε τιμή που να πλησιάζει το -0.8.

Για το *z-score* αναμένουμε τιμή που να πλησιάζει το 0 καθώς αναμένουμε ελάχιστα παράπονα των φοιτητών για κάθε τμήμα με χρήση του νέου συστήματος και πολλά παράπονα χωρίς χρήση του νέου συστήματος.

4.1.3 Υπόθεση 3

Σε αυτήν την υπόθεση ενδιαφερόμαστε να εξετάσουμε αν θα μειωθεί ο χρόνος ενασχόλησης του τεχνικού προσωπικού του τμήματος πληροφορικής για υποστήριξη λειτουργιών που αφορούν την υποστήριξη προγράμματος σπουδών με χρήση του νέου συστήματος σε σχέση με τη μη ύπαρξη συστήματος ή ύπαρξη οποιουδήποτε άλλου συστήματος.

Συγκέντρωση δεδομένων

Στο 1^ο ακαδημαϊκό έτος (χωρίς το νέο σύστημα) θα καταγραφεί από τον προϊστάμενο του τμήματος πληροφορικής ο χρόνος ενασχόλησης για την υποστήριξη λειτουργιών που αφορούν τα προγράμματα σπουδών. Θα καταγραφεί ο χρόνος υποστήριξης χωριστά για κάθε τμήμα του ιδρύματος. Οι χρόνοι υποστήριξης θα καταγραφούν και στο 2^ο ακαδημαϊκό έτος αντίστοιχα με χρήση του νέου συστήματος.

Αφού συγκεντρωθούν τα στοιχεία θα καταρτιστεί πίνακας ανά τμήμα που θα περιλαμβάνει τους χρόνους υποστήριξης του τμήματος πληροφορικής για τα δυο ακαδημαϊκά έτη. (Πίνακας 4.3)

Τμήμα	Χρόνος ενασχόλησης τμήματος πληροφορικής για την υποστήριξη του προγράμματος σπουδών χωρίς χρήση συστήματος ή χρήση άλλου συστήματος	Χρόνος ενασχόλησης τμήματος πληροφορικής για την υποστήριξη του προγράμματος σπουδών με χρήση του νέου συστήματος
Μηχανικών Πληροφορικής		
Λογιστική		
Μηχανολογία		
Διοίκηση Επιχειρήσεων		
Τεχνολογία Πετρελαίου		

Ηλεκτρολογία		
Αρχιτεκτονική Τοπίου		
Δασοπονία		
Νοσηλευτική		

Πίνακας 4.3: Χρόνοι υποστήριξης προγραμμάτων σπουδών ανά τμήμα.

Υπολογισμός στατιστικών διαδικασιών

Με βάση τα δεδομένα του πίνακα θα υπολογίσουμε τη μέση τιμή, τη μεσαία τιμή, τη διακύμανση, την τυπική απόκλιση των χρόνων με χρήση ή όχι του νέου συστήματος, το *z-score* κάθε τμήματος, τη συσχέτιση *Pearson r* μεταξύ των χρόνων με χρήση ή όχι του νέου συστήματος και το *Effect size* Δ .

Ερμηνεία των αποτελεσμάτων

Με βάση την υπόθεσή μας αναμένουμε αρνητική ισχυρή συσχέτιση. Για τη συσχέτιση *Pearson r* αναμένουμε τιμή που να πλησιάζει το -1, ενώ για το *Effect Size* (Δ) αναμένουμε τιμή που να πλησιάζει το -0.8.

Για το *z-score* αναμένουμε τιμή που να πλησιάζει το 0 καθώς αναμένουμε ελάχιστο χρόνο υποστήριξης των προγραμμάτων σπουδών από το τμήμα πληροφορικής για κάθε τμήμα με χρήση του νέου συστήματος και μεγάλο χρόνο υποστήριξης χωρίς χρήση του νέου συστήματος.

4.2 Αξιολόγηση του λογισμικού

Για την αξιολόγηση του λογισμικού βασιζόμαστε στην εργασία “Software Evaluation: Criteria-based Assessment” [30] που στηρίζεται στο πρότυπο ISO/IEC 9126-1 και περιλαμβάνει ερωτηματολόγια ανά κατηγορία αξιολόγησης του λογισμικού. Πρόκειται για μια αξιολόγηση του λογισμικού με όρους βιωσιμότητας(sustainability), συντηρησιμότητας (maintainability) και ευχρηστίας (usability). Η αξιολόγηση εξετάζει αν το λογισμικό και το έργο που το αναπτύσσει συμμορφώνεται με αρκετά χαρακτηριστικά και αν διαθέτει την απαραίτητη ποιότητα ώστε να θεωρηθεί βιώσιμο.

Για την εκτέλεση της αξιολόγησης αξιοποιήσαμε διαφορετικές κατηγορίες χρηστών και τους ζητήσαμε να συνεργαστούν μαζί μας μέσω συνέντευξης και με συμπλήρωση ερωτηματολογίου αφού τους παρουσιάσαμε το σύστημα που υλοποιήσαμε. Κάθε κατηγορία χρήστη απάντησε σε ερωτήματα που του αναλογούν. Στις επόμενες παραγράφους παρουσιάζουμε τις κατηγορίες χρηστών, τις ερωτήσεις που τους θέσαμε, τις απαντήσεις που πήραμε και τις αδυναμίες ή ελλείψεις που εντοπίσαμε και πρέπει να επιλυθούν για τη μελλοντική συμμόρφωση του συστήματος.

4.2.1 Χρήστες αξιολόγησης του συστήματος

Για την αξιολόγηση του συστήματος αξιοποιήσαμε τους εξής χρήστες (τα έντονα γράμματα επισημαίνουν την κατηγορία του χρήστη ως αξιολογητή του συστήματος):

- Έναν **Μηχανικό Λογισμικού** και **Project Manager** Ευρωπαϊκών προγραμμάτων, Βρυξέλες
- Έναν **Διαχειριστή Συστημάτων** και Δικτύων, Προϊστάμενος γενικού τμήματος Πληροφορικής, ΤΕΙ ΑΜΘ
- Τον **Προϊστάμενο τμήματος** Μηχανικών Πληροφορικής, ΤΕΙ ΑΜΘ.
- Έναν **Απόφοιτο τμήματος** Βιομηχανικής Πληροφορικής, ΤΕΙ ΑΜΘ
- Μία **Φοιτήτρια τμήματος** Μηχανικών Πληροφορικής, ΤΕΙ ΑΜΘ.

4.2.2 Αξιολόγηση από χρήστες κατηγορίας Μηχανικού Λογισμικών

Ο χρήστης αυτού το ρόλου συμπλήρωσε ερωτηματολόγιο που σχετίζεται με την ευχρηστία, τη βιωσιμότητα και συντηρησιμότητα του συστήματος. Οι ερωτήσεις και οι απαντήσεις παρουσιάζονται ανά τομέα αξιολόγησης στους πίνακες 4.4 έως 4.14.

Usability, Understandability	Ναι/Όχι, Συνοδευτικά Σχόλια
	Μηχανικός Λογισμικού
Είναι κατανοητό που απευθύνεται το λογισμικό και σε ποιους χρήστες;	Ναι
Είναι κατανοητό ποιες είναι οι βασικές λειτουργίες του λογισμικού;	Ναι
Είναι κατανοητό ποιες είναι οι προχωρημένες λειτουργίες του λογισμικού;	Ναι
Υπάρχει υψηλού επιπέδου περιγραφή στο τι κάνει το λογισμικό;	Ναι
Υπάρχει διαθέσιμη σύνοψη της αρχιτεκτονικής του συστήματος με διαγράμματα;	Ναι
Είναι διαθέσιμες περιγραφές που προορίζονται για περιπτώσεις χρήσης;	Ναι, Χρειάζεται πιο αναλυτική περιγραφή

Πίνακας 4.4 : Ευχρηστία, κατανοησιμότητα

Usability, Documentation	Ναι/Όχι, Συνοδευτικά Σχόλια
	Μηχανικός Λογισμικού
Η τεκμηρίωση των χρηστών είναι ποιοτική, ολοκληρωμένη, ακριβής, κατάλληλη και διαυγής;	Όχι, Χρειάζεται σημαντική δουλειά στην τεκμηρίωση χρήστη που δεν έχει γίνει
Περιέχει υψηλού επιπέδου επισκόπηση του λογισμικού;	Ναι
Είναι χωρισμένη σε ενότητες με βάση την κατηγορία του χρήστη;	Όχι
Δηλώνει το αναμενόμενο υπόβαθρο και την εμπειρία για κάθε κατηγορία χρήστη;	Όχι
Είναι οργανωμένο σε κατά μέρους εργασίες (task-oriented)	Ναι, Υπάρχουν ακόμα ελλείψεις
Αποτελείται από σαφείς οδηγίες βήμα προς βήμα.	Ναι, μόνο στις κατά μέρους εργασίες που υπάρχουν
Δίνει παραδείγματα για το τι θα δει ο χρήστης σε μορφή εικόνων;	Ναι
Παρέχονται βήμα προς βήμα λύσεις-κατευθύνσεις για προβλήματα ή μηνύματα λάθους που θα εμφανιστούν;	Όχι
Τα μενού περιγράφονται ακριβώς όπως εμφανίζονται;	Ναι, Χρειάζεται και άλλη προσπάθεια

Σε περίπτωση που η γλώσσα ανάπτυξης είναι Java οι κλάσεις επεξηγούνται την πρώτη φορά που αναφέρονται;	Ναι
Ο έλεγχος έκδοσης υπάρχει μέσα στον κώδικα;	Όχι
Η τεκμηρίωση υπάρχει στον ιστότοπο;	Όχι

Πίνακας 4.5 : Ευχρηστία, Τεκμηρίωση

Usability, <u>Buildability</u>	Ναι/Όχι, Συνοδευτικά Σχόλια
	Μηχανικός Λογισμικού
Είναι εύκολο να ικανοποιηθούν οι προϋποθέσεις ώστε να γίνει build το λογισμικό σε κάποια πλατφόρμα;	Ναι
Ο ιστότοπος έχει οδηγίες για το πώς γίνεται build το λογισμικό;	Όχι
Παρέχεται αυτοματοποιημένος τρόπος ώστε να γίνει build το λογισμικό;	Ναι, στο περιβάλλον NetBeans 8 EE
Παρέχεται αυτοματοποιημένος τρόπος Dependency Management;	Ναι, Μόνο στο υποσύστημα παροχής υπηρεσιών REST
Είναι διαθέσιμες όλες οι third-party	Ναι

dependencies;	

Πίνακας 4.6: Ευχρηστία, Οικοδομησιμότητα

Usability, <u>Installability</u>	Ναι/Όχι, Συνοδευτικά Σχόλια
	Μηχανικός Λογισμικού
Είναι εύκολο να ικανοποιηθούν οι προϋποθέσεις εγκατάστασης του λογισμικού σε κάποια πλατφόρμα;	Ναι
Είναι εύκολη η εγκατάσταση του λογισμικού σε κάποια πλατφόρμα;	Ναι
Είναι εύκολο να επαληθευτεί πως η εγκατάσταση ολοκληρώθηκε και το λογισμικό είναι έτοιμο προς χρήση;	Ναι
Παρέχονται οδηγίες εγκατάστασης του λογισμικού;	Ναι
Παρέχονται τεστ επαλήθευσης της εγκατάστασης;	Όχι
Όταν αποσυμπίεζεται ένα αρχείο εγκατάστασης τα αρχεία τοποθετούνται κάτω από έναν υποφάκελο;	Ναι
Το εγκατεστημένο λογισμικό είναι οργανωμένο σε υποφακέλους που δεικνύουν το περιεχόμενο των αρχείων	Ναι

τους;	
Η διανομή του λογισμικού παρέχει αρχείο README.txt με πληροφορίες για το έργο, επικοινωνίας, πνευματικής ιδιοκτησίας κλπ.	Όχι
Όλες οι οθόνες ή σελίδες διεπαφής του λογισμικού παρέχουν πληροφορίες για το έργο, επικοινωνίας, πνευματικής ιδιοκτησίας κλπ.	Όχι
Δίνεται η δυνατότητα κατά την εγκατάσταση ο χρήστης να ορίσει το φάκελο που θα εγκατασταθεί το λογισμικό;	Όχι
Κατά την απεγκατάσταση αφαιρούνται όλα τα αρχεία και τα ίχνη του λογισμικού;	Όχι, Δεν έχει κατασκευαστεί εργαλείο απεγκατάστασης

Πίνακας 4.7 : Ευχρηστία, Εγκαταστασιμότητα

Usability, <u>Learnability</u>	Ναι/Όχι, Συνοδευτικά Σχόλια
	Μηχανικός Λογισμικού
Ήταν εύκολο να μάθεις πως εκτελούνται οι απλές λειτουργίες	Ναι
Ήταν εύκολο να μάθεις πως εκτελούνται οι προχωρημένες λειτουργίες	Ναι

Παρέχεται εγχειρίδιο τύπου «Getting started guide»	Όχι
Παρέχονται οδηγίες για αρκετές περιπτώσεις χρήσης;	Ναι
Παρέχονται οδηγίες για όλες τις περιπτώσεις χρήσης;	Όχι
Παρέχεται API documentation για τους developers	Ναι

Πίνακας 4.8: Ευχρηστία, Ευκολία εκμάθησης

Sustainability and maintainability, Identity	Ναι/Όχι, Συνοδευτικά Σχόλια
	Μηχανικός Λογισμικού
Έχει αποδοθεί στο λογισμικό όνομα domain ;	Όχι
Έχει logo το λογισμικό;	Όχι
Έχει αποδοθεί ξεχωριστό όνομα στο λογισμικό ώστε να ξεχωρίζει στον τομέα εφαρμογής;	Όχι

Πίνακας 4.9 : Βιωσιμότητα και συντηρησιμότητα, Ταυτότητα

Sustainability and maintainability, Copyright, Licencing	Ναι/Όχι, Συνοδευτικά Σχόλια
	Μηχανικός Λογισμικού
Είναι ξεκάθαρο ποιος έγραψε το λογισμικό και έχει τα πνευματικά	Ναι

δικαιώματα;	
Δηλώνει ο ιστότοπος τα πνευματικά δικαιώματα;	Όχι
Δηλώνει ο ιστότοπος ποιος ανέπτυξε το λογισμικό;	Ναι
Υπάρχουν τα πνευματικά δικαιώματα σε κάθε αρχείο του πηγαίου κώδικα καταχωρημένο σε μια σταθερά (Constant) του προγράμματος;	Όχι
Έχει κάθε πηγαίο αρχείο επικεφαλίδα με την άδεια χρήσης;	Όχι
Έχει υιοθετηθεί κατάλληλη άδεια χρήσης;	Όχι

Πίνακας 4.10: Βιωσιμότητα και συντηρησιμότητα, Πνευματικά δικαιώματα και Αδειοδότηση

Sustainability and maintainability, Accessibility, Testability	Ναι/Όχι, Συνοδευτικά Σχόλια
	Μηχανικός Λογισμικού
Η διανομή του λογισμικού είναι ελεύθερα διαθέσιμη;	Όχι
Η διανομή του πηγαίου κώδικα του λογισμικού είναι ελεύθερα διαθέσιμη;	Όχι
Διαθέτει το έργο Unit Tests;	Όχι
Διαθέτει το έργο Integration Tests;	Όχι

Διαθέτει το έργο GUI test Frameworks;	Όχι
Διαθέτει το έργο τεστ συμμόρφωσης με Coding Standards;	Όχι, Το σύνολο της ελεγκσιμότητας δεν έχει γίνει σημαντική δουλειά

Πίνακας 4.11: Βιωσιμότητα και συντηρησιμότητα, Προσβασιμότητα και Ελεγκσιμότητα

Sustainability and maintainability, Portability	Ναι/Όχι, Συνοδευτικά Σχόλια
	Μηχανικός Λογισμικού
Μπορεί το λογισμικό να χρησιμοποιηθεί σε άλλες πλατφόρμες;	Ναι
Μπορεί το λογισμικό να λειτουργήσει σε πλατφόρμα Windows;	Ναι
Μπορεί το λογισμικό να λειτουργήσει σε πλατφόρμα Linux	Ναι
Μπορεί το λογισμικό να λειτουργήσει σε πλατφόρμα, Unix	Ναι
Μπορεί το λογισμικό να λειτουργήσει σε πλατφόρμα MacOS	Ναι
Οι διαδικτυακές εφαρμογές του λογισμικού λειτουργούν με όλους τους δημοφιλείς περιηγητές διαδικτύου (Web Browsers);	Ναι

Πίνακας 4.12: Βιωσιμότητα και συντηρησιμότητα, Φορητότητα

Sustainability and maintainability, Analyzability	Ναι/Όχι, Συνοδευτικά Σχόλια
	Μηχανικός Λογισμικού
Είναι εύκολη η κατανόηση των αρχείων πηγαίου κώδικα και πως ταιριάζουν στην αρχιτεκτονική υλοποίησης;	Ναι
Ο πηγαίος κώδικας είναι δομημένος σε τμήματα και πακέτα ;	Ναι
Παρέχονται τα αρχεία project του περιβάλλοντος ανάπτυξης (IDE);	Ναι
Υπάρχει τεκμηρίωση της δομής της αποθήκης του πηγαίου κώδικα (source code repository) σε σχέση με τα τμήματα του λογισμικού;	Όχι
Είναι σχολιασμένος ο πηγαίος κώδικας;	Όχι
Είναι μορφοποιημένος σωστά ο πηγαίος κώδικας;	Ναι
Έχουν δοθεί ενδεδειγμένα ονόματα στα πακέτα, τις κλάσεις, τις συναρτήσεις και στις μεταβλητές;	Ναι
Ο αυτόματα παραγόμενος πηγαίος κώδικας είναι σε διαφορετικούς υποφακέλους από τον υπόλοιπο πηγαίο κώδικα;	Όχι

Τα πρότυπα κωδικοποίησης του πηγαίου κώδικα του έργου είναι συμβατά με τα πρότυπα κωδικοποίησης της κοινότητας (Java coding Standards);	Ναι
---	-----

Πίνακας 4.13 : Βιωσιμότητα και συντηρησιμότητα, Αναλυσιμότητα

Sustainability and maintainability, Changeability, Evolvability, Interoperability	Ναι/Όχι, Συνοδευτικά Σχόλια
	Μηχανικός Λογισμικού
Είναι εύκολη η αντιμετώπιση ενδεχομένων προβλημάτων, η τροποποίηση και η προσθήκη νέων λειτουργιών;	Ναι
Έχει ορίσει το έργο πολιτική συνεισφοράς προς αυτό;	Όχι
Έχει καθοριστεί η προοπτική εξέλιξης του έργου;	Όχι
Χρησιμοποιεί το λογισμικό open standards;	Ναι
Είναι εφικτή η διαλειτουργικότητά του με third-party στοιχεία λογισμικού;	Ναι
Είναι ώριμα τα open standards που χρησιμοποιεί;	Ναι
Παρέχει τεστ που να επιδεικνύουν τη συμμόρφωση με τα open standards;	Όχι

Πίνακας 4.14: Βιωσιμότητα και συντηρησιμότητα, Ανοχή σε αλλαγές, Μετατρεψιμότητα, Διαλειτουργικότητα

4.2.3 Αξιολόγηση από χρήστες κατηγορίας Προϊσταμένου τμήματος

Ο χρήστης αυτού το ρόλου συμπλήρωσε ερωτηματολόγιο που σχετίζεται με τη ευχρηστία του συστήματος. Οι ερωτήσεις και οι απαντήσεις παρουσιάζονται στους πίνακα 4.15 και 4.16 .

Usability, <u>Understandability</u>	Ναι/Όχι, Συνοδευτικά Σχόλια
	Προϊστάμενος τμήματος
Είναι κατανοητό που απευθύνεται το λογισμικό και σε ποιους χρήστες;	Ναι
Είναι κατανοητό ποιες είναι οι βασικές λειτουργίες του λογισμικού;	Ναι
Είναι κατανοητό ποιες είναι οι προχωρημένες λειτουργίες του λογισμικού;	Ναι
Υπάρχει υψηλού επιπέδου περιγραφή στο τι κάνει το λογισμικό;	Ναι

Πίνακας 4.15 : Ευχρηστία, κατανοησιμότητα

Usability, <u>Learnability</u>	Ναι/Όχι, Συνοδευτικά Σχόλια
	Προϊστάμενος τμήματος
Ήταν εύκολο να μάθεις πως εκτελούνται οι απλές λειτουργίες	Ναι
Ήταν εύκολο να μάθεις πως εκτελούνται	Ναι, Υπάρχει κάποια πολυπλοκότητα στις

οι προχωρημένες λειτουργίες	διεργασίες που συμμετείχα
Παρέχεται εγχειρίδιο τύπου «Getting started guide»	Όχι
Παρέχονται οδηγίες για αρκετές περιπτώσεις χρήσης;	Ναι
Παρέχονται οδηγίες για όλες τις περιπτώσεις χρήσης;	Όχι

Πίνακας 4.16 : Ευχρηστία, Ευκολία εκμάθησης

4.2.4 Αξιολόγηση από χρήστες κατηγορίας Διαχειριστή Συστημάτων

Ο χρήστης αυτού το ρόλου συμπλήρωσε ερωτηματολόγιο που σχετίζεται με την εγκατάσταση του συστήματος. Οι ερωτήσεις και οι απαντήσεις παρουσιάζονται στους πίνακες 4.17 και 4.18 .

Usability,Understandability	Ναι/Όχι, Συνοδευτικά Σχόλια
	Διαχειριστής Συστημάτων και Δικτύων
Είναι κατανοητός ο στόχος του λογισμικού και τι κάνει;	Ναι
Είναι κατανοητό που απευθύνεται το λογισμικό και σε ποιους χρήστες;	Ναι
Είναι κατανοητό ποιες είναι οι βασικές λειτουργίες του λογισμικού;	Ναι
Είναι κατανοητό ποιες είναι οι προχωρημένες λειτουργίες του λογισμικού;	Ναι

Υπάρχει διαθέσιμη σύνοψη της αρχιτεκτονικής του συστήματος με διαγράμματα;	Ναι

Πίνακας 4. 17: Ευχρηστία, κατανοησιμότητα

Usability, <u>Installability</u>	Ναι/Όχι, Συνοδευτικά Σχόλια
	Διαχειριστής Συστημάτων και Δικτύων
Είναι εύκολο να ικανοποιηθούν οι προϋποθέσεις εγκατάστασης του λογισμικού σε κάποια πλατφόρμα;	Ναι
Είναι εύκολη η εγκατάσταση του λογισμικού σε κάποια πλατφόρμα;	Ναι
Είναι εύκολο να επαληθευτεί πως η εγκατάσταση ολοκληρώθηκε και το λογισμικό είναι έτοιμο προς χρήση;	Ναι
Παρέχονται οδηγίες εγκατάστασης του λογισμικού;	Ναι
Παρέχονται τεστ επαλήθευσης της εγκατάστασης;	Όχι
Όταν αποσυμπιέζεται ένα αρχείο εγκατάστασης τα αρχεία τοποθετούνται κάτω από έναν υποφάκελο;	Ναι
Το εγκατεστημένο λογισμικό είναι οργανωμένο σε υποφακέλους που	Ναι

δεικνύουν το περιεχόμενο των αρχείων τους;	
Η διανομή του λογισμικού παρέχει αρχείο README.txt με πληροφορίες για το έργο, επικοινωνίας, πνευματικής ιδιοκτησίας κλπ.	Όχι
Όλες οι οθόνες ή σελίδες διεπαφής του λογισμικού παρέχουν πληροφορίες για το έργο, επικοινωνίας, πνευματικής ιδιοκτησίας κλπ.	Όχι
Δίνεται η δυνατότητα κατά την εγκατάσταση ο χρήστης να ορίσει το φάκελο που θα εγκατασταθεί το λογισμικό;	Όχι
Κατά την απεγκατάσταση αφαιρούνται όλα τα αρχεία και τα ίχνη του λογισμικού;	Όχι, Δεν έχει κατασκευαστεί εργαλείο απεγκατάστασης

Πίνακας 4.18: Ευχρηστία, Εγκαταστασιμότητα

4.2.5 Αξιολόγηση από χρήστες κατηγορίας Φοιτητή

Οι χρήστες αυτού του ρόλου συμπλήρωσαν ερωτηματολόγιο που σχετίζεται με την ευχρηστία του συστήματος. Οι ερωτήσεις και οι απαντήσεις τους παρουσιάζονται στους πίνακες 4.19 και 4.20.

Usability, Understandability	Ναι/Όχι, Συνοδευτικά Σχόλια	
	Απόφοιτος τμήματος	Φοιτητής τμήματος
Είναι κατανοητός ο στόχος του λογισμικού και τι κάνει;	Ναι	Ναι

Είναι κατανοητό ποιες είναι οι βασικές λειτουργίες του λογισμικού;	Ναι	Ναι
Υπάρχει υψηλού επιπέδου περιγραφή στο τι κάνει το λογισμικό;	Ναι	Ναι

Πίνακας 4.19 : Ευχρηστία, κατανοησιμότητα

Usability, <u>Learnability</u>	Ναι/Όχι, Συνοδευτικά Σχόλια	
	Απόφοιτος τμήματος	Φοιτητής τμήματος
Ήταν εύκολο να μάθεις πως εκτελούνται οι λειτουργίες;	Ναι	Ναι
Παρέχεται εγχειρίδιο τύπου «Getting started guide»	Όχι	Όχι

Πίνακας 4.20 : Ευχρηστία, Ευκολία εκμάθησης

4.2.6 Αποτελέσματα αξιολόγησης συστήματος

Από τις απαντήσεις των χρηστών είναι προφανές πως το λογισμικό συγκεντρώνει θετικά σχόλια σε θέματα: κατανοησιμότητας (Understandability), οικοδομησιμότητας (Buildability), εγκατάστασης (Installability), φορητότητας (Portability), αναλυσιμότητας (Analyzability), μετατρεψιμότητας (Changeability) και διαλειτουργικότητας (Interoperability).

Ωστόσο είναι προφανές πως το έργο/λογισμικό έχει πολύ σοβαρές ελλείψεις στην τεκμηρίωση (Documentation), στην ελεγχσιμότητα (Testability), δεν έχει οριστεί η ταυτότητά του (Identity), τα πνευματικά δικαιώματα (Copyright) και η αδειοδότηση της χρήσης του (Licensing), ενώ δεν έχει καθοριστεί η προοπτική εξέλιξής του (Evolvability). Στόχος όμως της μεταπτυχιακής διατριβής ήταν να κατασκευαστεί ένα πρωτότυπο και όχι ένα προϊόν.

Κεφάλαιο 5

Μελλοντικές Επεκτάσεις

Η παρούσα εργασία απέδωσε ένα σύνθετο πληροφοριακό σύστημα υποστήριξης προγραμμάτων σπουδών που αποτελείται από αρκετά υποσυστήματα, ενώ διασυνδέεται με αρκετά εξωτερικά υποσυστήματα. Μπορεί να λειτουργήσει σε πιλοτική μορφή από τα ελληνικά εκπαιδευτικά ιδρύματα, αλλά επιδέχεται αρκετές προσθήκες, ώστε να αποτελέσει προϊόν. Μερικές από τις σημαντικές εργασίες που προτείνονται είναι οι εξής: δυνατότητα επεξεργασίας του ωρολογίου προγράμματος σπουδών, ενσωμάτωση εργαλείων εξόρυξης δεδομένων για την πρόβλεψη δηλώσεων μαθημάτων μελλοντικών ακαδημαϊκών περιόδων, ασφαλής χρήση των υπηρεσιών REST, εμπλουτισμό των λειτουργικών απαιτήσεων.

5.1 Συντάκτης ωρολογίου προγράμματος

Το υποσύστημα ωρολογίου προγράμματος που έχει υλοποιηθεί μπορεί να ορίσει αρκετούς περιορισμούς που θα κάλυπταν τις περισσότερες προτιμήσεις κατασκευής ωρολογίων προγραμμάτων από τα εκπαιδευτικά ιδρύματα. Ωστόσο δεν έχει υλοποιηθεί κάποιος επεξεργαστής ή συντάκτης ωρολογίου προγράμματος ώστε οι προϊστάμενοι να παραμετροποιούν με βάση τις προτιμήσεις των τμημάτων. Στο υλοποιημένο υποσύστημα όταν δημιουργείται ένα ωρολόγιο πρόγραμμα κατασκευάζεται ωρολόγιο πρόγραμμα με μια μόνο συγκεκριμένη παραμετροποίηση (η οποία περιλαμβάνει πολλούς περιορισμούς) την οποία οι προϊστάμενοι δεν μπορούν να αλλάξουν. Ενδεικτικά ένας συντάκτης ωρολογίου προγράμματος θα μπορούσε να προσφέρει στους προϊσταμένους των τμημάτων τις εξής δυνατότητες:

- Προσδιορισμός του διαθέσιμου εύρους ημερών του εβδομαδιαίου προγράμματος (π.χ. Δευτέρα έως Παρασκευή)

- Προσδιορισμός του εύρους των ωρών ανά ημέρα (π.χ. Δευτέρα 08:00 – 17:00, Τρίτη 10:00 – 21:00 κλπ.

- Να καθορίσει σε ποια μαθήματα οι ώρες των ομάδων παρακολούθησης θα είναι διαδοχικές.
- Να καθορίσει ποια μαθήματα θα μπορούν να πραγματοποιούνται σε συγκεκριμένο εύρος ημερών.
- Να καθορίσει ποια μαθήματα θα μπορούν να πραγματοποιούνται σε συγκεκριμένο εύρος ωρών.
- Μερική αναθεώρηση ωρολογίου προγράμματος.

5.2 Πρόβλεψη δηλώσεων μελλοντικών ακαδημαϊκών περιόδων

Στο υποσύστημα εκπαιδευτικών οι προϊστάμενοι μπορούν να δουν κάποια στατιστικά στοιχεία που αφορούν δηλώσεις μαθημάτων προηγούμενων περιόδων καθώς και δηλώσεις ομάδων παρακολούθησης μαθημάτων των ακαδημαϊκών περιόδων από την εγκατάσταση του συστήματος και μελλοντικά. Θα ήταν βολικό ωστόσο η ύπαρξη ενός υποσυστήματος που θα χρησιμοποιεί τα παραπάνω στατιστικά στοιχεία και να κάνει εκτιμήσεις δηλώσεων ομάδων παρακολούθησης για τις μελλοντικές ακαδημαϊκές περιόδους, ώστε οι προϊστάμενοι των τμημάτων να κάνουν καλύτερους προγραμματισμούς. Το νέο προτεινόμενο υποσύστημα θα μπορούσε να κάνει ενδεχομένως και προβλέψεις που αφορούν τις αναθέσεις των εκπαιδευτικών σε μαθήματα.

Για την στήριξη αυτού το υποσυστήματος θα εξετάζαμε τη μεθοδολογία εξόρυξης δεδομένων CRISP. Η μεθοδολογία αυτή εισηγείται έξι στάδια για την ανάπτυξη επιτυχημένων μοντέλων εξόρυξης δεδομένων: κατανόηση του εκπαιδευτικού ιδρύματος (Institutional Understanding), κατανόηση δεδομένων (Data Understanding), προετοιμασία δεδομένων (Data Preparation), μοντελοποίηση (Modeling), αξιολόγηση (Evaluation) και ανάπτυξη (Deployment). [3]

5.3 Ασφάλεια δεδομένων στην επικοινωνία του υποσυστήματος υπηρεσιών REST

Η υλοποίηση των παρεχόμενων υπηρεσιών Rest στηρίζεται στο HTTP πρωτόκολλο. Τα αιτήματα εξυπηρέτησης γίνονται με κλήσεις Rest μεθόδων σε μορφή απλού κειμένου (plain text

messages), χωρίς κρυπτογράφηση. Για μια ασφαλή υλοποίηση των αναφερομένων υπηρεσιών προτείνεται η χρήση του πρωτοκόλλου **OAuth 2.0**. [14]

5.4 Λειτουργικές επεκτάσεις

Για λόγους απλοποίησης έχουμε περιορίσει κάποιες λειτουργικές απαιτήσεις. Για παράδειγμα έχουμε θεωρήσει πως ένας εκπαιδευτικός δε μπορεί να διδάσκει ή να είναι υπεύθυνος σε μαθήματα άλλων τμημάτων. Παρομοίως έχουμε θεωρήσει ότι σε μια αίθουσα διδάσκονται αποκλειστικά μαθήματα ενός τμήματος. Αυτοί οι περιορισμοί δεν ευσταθούν πάντα σε πραγματικές συνθήκες και πρέπει να τροποποιηθούν.

Επίλογος

Στη μεταπτυχιακή διατριβή επιχειρήσαμε την ερευνητική και την τεχνική προσέγγιση για την ανάπτυξη ενός συστήματος που θα προσφέρει αποδοτική εκτέλεση των προγραμμάτων σπουδών σε οποιοδήποτε ελληνικό τριτοβάθμιο εκπαιδευτικό ίδρυμα. Προσδιορίσαμε υποθέσεις και σε συνδυασμό με το υλοποιημένο σύστημα φιλοδοξούμε καλύτερη ποιότητα στις υπηρεσίες εκτέλεσης προγραμμάτων σπουδών με παράλληλη εξοικονόμηση πόρων. Αν και η υλοποίηση είναι πιλοτική έχει κατασκευαστεί ένας ισχυρός πυρήνας λογισμικού και τεχνολογιών που επιτρέπουν την μετατροπή του σε προϊόν με κάποια επιπλέον προσπάθεια, ενώ επιτρέπει την κλιμάκωσή του σε νέες λειτουργίες και δυνατότητες. Με γνώμονα την εμπειρία που αποκτήσαμε στη διάρκεια εκπόνησης της διατριβής θεωρούμε ότι ακαδημαϊκά ιδρύματα τα οποία θα ενσωματώσουν στη λειτουργία τους την αρχιτεκτονική συστημάτων BPM, όπως το σύστημα που αναπτύξαμε, θα εξασφαλίσουν σημαντική εξοικονόμηση πόρων με υψηλή ποιότητα υπηρεσιών.

Βιβλιογραφία

- [1] Donald Ary, Lucy Cheser Jacobs, Chris Sorensen, and Asghar Razavieh, *Introduction to Research in Education*, 8th ed. Wadsworth, Cengage Learning, 2010.
- [2] Chitu Okoli and Kira Schabram, “A Guide to Conducting a Systematic Literature Review of Information Systems Research,” *Sprouts*, no. Working Papers on Information Systems, 2010.
- [3] Elliot N. Maltz, Kenneth E. Murphy, Michael L. Hand, “Decision support for university enrollment management: Implementation and experience,” pp. 106–123, 2007.
- [4] Jacobson-Schulte, Schmidt, and Yoshimoto, “THE USE AND SECURITY OF DATA IN THE HIGHER EDUCATION ENVIRONMENT.” 2009.
- [5] Zeb Nash, “UK higher education: sector wide study of timetabling and resource scheduling: presentation of findings and discussion,” *PATAT 2010*, vol. Queen's University Belfast, 2010.
- [6] Hana Rudova, Tomas Muller, “Rapid Development of University Course Timetables.” 2011.
- [7] J. S. Appleby, D. V. Blake, and E. A. Newman, “Techniques for Producing School Timetables on a Computer and their Application to other Scheduling Problems,” 1961.
- [8] S. A. MirHassani, F. Habibi, “Solution approaches to the course timetabling problem,” 2011.
- [9] S. D. T. B. b, E. H. E. Housos, “An integer programming formulation for a case study in university timetabling,” 2004.
- [10] “Scheduling Utilities,” http://www.dmoz.org/Computers/Software/Educational/Administration_and_School_Management/Scheduling_Uilities/.
- [11] Wikipedia, “FET.”
- [12] “Free Evolutionary Timetabling”, <http://www.lalescu.ro/liviu/fet/>
- [13] Antonio Goncalves, *Beginning Java EE 7*. Apress, 2013.
- [14] Bill Burke, *RESTful Java with JAX-RS 2.0*, 2nd ed. O'Reilly Media, 2013.
- [15] Tim O'Brien, John Casey, Brian Fox, Jason Van Zyl, Juven Xu, Thomas Locher Dan Fabulich, Eric Redmond, and Bruce Snyder, *Maven by Example*. Sonatype, 2011.
- [16] Çağatay Çivici, *PrimeFaces 5.2 User Guide*. 2013.
- [17] David R. Heffelfinger, *Java EE 7 Development with NetBeans 8*, 3rd ed. Packt Publishing, 2015.
- [18] Michal Cmil, *Java EE 7 Development with WildFly*. Packt Publishing, 2014.
- [19] *jBPM6 Developer Guide*. Packt Publishing, 2014.
- [20] The JBoss jBPM team, *jBPM6 User guide*. <http://docs.jboss.org/jbpm/v6.2/userguide/>, 2014.
- [21] Bill Phillips and Brian Hardy, *Android Programming: The Big Nerd Ranch Guide*. Big Nerd Ranch, 2013.
- [22] “KEPLER Eclipse”, <http://www.eclipse.org/kepler/>
- [23] Marian Buenosayres, “Java and BPM Developer blog.”, <https://marianbuenosayres.wordpress.com/>
- [24] Novell, “LDAP Classes for Java.”, https://www.novell.com/developer/ndk/ldap_classes_for_java.html

- [25] Community, "OpenLDAP.", <http://www.openldap.org/>
- [26] Ed Ort and Bhakti Mehta, "Java Architecture for XML Binding (JAXB)", <http://www.oracle.com/technetwork/articles/javase/index-140168.html>, 2003.
- [27] S. B. N. Ramez Elmasri, *Fundamentals of Database Systems*, 6th ed. Addison-Wesley, Pearson.
- [28] "Oracle SQL Developer," *Oracle*, www.oracle.com/technetwork/developer-tools/sql-developer/overview/
- [29] GNU LGPL, "jTDS - SQL Server and Sybase JDBC driver," *sourceforge.net*.
- [30] Mike Jackson, Steve Crouch and Rob Baxter, "Software Evaluation: Criteria-based Assessment", Software Sustainability Institute, 2011

Παράρτημα Α

Κλάσεις υποσυστημάτων

Στις επόμενες ενότητες παρουσιάζεται λεπτομερής περιγραφή των κλάσεων κάθε υποσυστήματος.

A.1 Περιγραφή κλάσεων υποσυστήματος φοιτητών

DEPT
-NO -NAME -SITE -LAT -LNG -SECMAIL
+DEPT() +getNO() +setNO(NO) : void +getName() +setName(NAME) : void +getSITE() +setSITE(SITE) : void +getLAT() +setLAT(LAT) : void +getLNG() +setLNG(LNG) : void +getSECMAIL() +setSECMAIL(SECMAIL) : void

Η κλάση **DEPT** περιέχει τα χαρακτηριστικά τμήματος του ιδρύματος: κωδικός (**NO**), όνομα (**NAME**), ηλεκτρονική διεύθυνση (**SITE**), γεωγραφικό πλάτος (**LAT**), γεωγραφικό μήκος (**LNG**) και διεύθυνση ηλεκτρονικού ταχυδρομείου γραμματείας τμήματος (**SECMAIL**).

ACAD_PERIOD
-ID -DEPTNO -AYEAR -TERM -REG_START_DATE -REG_END_DATE -SEM_CR_START_DATE -SEM_CR_END_DATE
+getID() +setID(ID) : void +getDEPTNO() +setDEPTNO(DEPTNO) : void +getAYEAR() +setAYEAR(AYEAR) : void +getTERM() +setTERM(TERM) : void +getREG_START_DATE() +setREG_START_DATE(REG_START_DATE) : void +getREG_END_DATE() +setREG_END_DATE(REG_END_DATE) : void +getSEM_CR_START_DATE() +setSEM_CR_START_DATE(SEM_CR_START_DATE) : void +getSEM_CR_END_DATE() +setSEM_CR_END_DATE(SEM_CR_END_DATE) : void +ACAD_PERIOD()

Η κλάση **ACAD_PERIOD** περιέχει τα χαρακτηριστικά μιας ακαδημαϊκής περιόδου: κωδικός ακαδημαϊκής περιόδου (**ID**), κωδικός τμήματος (**DEPTNO**), ακαδημαϊκό έτος (**AYEAR**), ακαδημαϊκή περίοδος (**TERM**), περίοδος δηλώσεων ομάδων παρακολούθησης μαθημάτων (**REG_START_DATE, REG_END_DATE**), περίοδος ενεργοποίησης κριτηρίου εξαμήνου δηλώσεων ομάδων (**SEM_CR_START_DATE, SEM_CR_END_DATE**).

Student
-AEM -LNAME -FNAME -EMAIL -SEMESTER -UNAME -PASSWD -DEPTNO
+getAEM() +setAEM(AEM) : void +getLNAME() +setLNAME(LNAME) : void +getFNAME() +setFNAME(FNAME) : void +getEmail() +setEMAIL(EMAIL) : void +getSEMESTER() +setSEMESTER(SEMESTER) : void +getUNAME() +setUNAME(UNAME) : void +getPASSWD() +setPASSWD(PASSWD) : void +getDEPTNO() +setDEPTNO(DEPTNO) : void +Student()

Η κλάση **Student** περιέχει τα χαρακτηριστικά του φοιτητή: αριθμός μητρώου (**AEM**), επώνυμο(**LNAME**), όνομα (**FNAME**), διεύθυνση ηλεκτρονικής αλληλογραφίας (**EMAIL**), τρέχον εξάμηνο φοίτησης (**SEMESTER**), όνομα λογαριασμού (**UNAME**), κωδικός λογαριασμού (**PASSWD**), κωδικός τμήματος στο οποίο ανήκει ο φοιτητής (**DEPTNO**).

Lesson
-CODE -TITLE -SEMESTERNO -SECTIONCOUNT -FINALSTATUS -DEPTNO -TEACHER_UNAME
+getCode() +setCODE(CODE) : void +getTitle() +setTitle(TITLE) : void +getSemesterNo() +setSemesterNo(SemesterNo) : void +getSectionCount() +setSectionCount(SectionCount) : void +getFinalStatus() +setFinalStatus(FinalStatus) : void +getDeptNo() +setDeptNo(DeptNo) : void +getTeacher_Uname() +setTeacher_Uname(Teacher_Uname) : void +Lesson()

Η κλάση **Lesson** περιέχει τα χαρακτηριστικά ενός μαθήματος: κωδικός μαθήματος-όπως ορίζεται από το πρόγραμμα σπουδών- (**CODE**), τίτλος(**TITLE**), εξάμηνο διδασκαλίας (**SEMESTERNO**), πλήθος ομάδων παρακολούθησης μαθήματος (**SECTIONCOUNT**), κωδικός τμήματος (**DEPTNO**), κωδικός υπεύθυνου καθηγητή (**TEACHER_UNAME**). Αν κάποιο μάθημα είναι μικτό (θεωρητικό κ εργαστηριακό) τότε δημιουργούνται δυο αντικείμενα της ίδια κλάσης χωρίς να υπάρχει σύνδεση μεταξύ τους.

LSECTION
-ACAD_PERIOD_ID -LESSONCODE -LSNO -STARTTIME -ENDTIME -STARTMIN -ENDMIN -FULLTIME -MAXSPACE -BOOKEDSPACE -FREESPACE -WDNO -DEPTNO
+getACAD_PERIOD_ID() +setACAD_PERIOD_ID(ACAD_PERIOD_ID) : void +getLessonCode() +setLessonCode(LessonCode) : void +getLSNO() +setLSNO(LSNO) : void +getStartTime() +setStartTime(StartTime) : void +getEndTime() +setEndTime(EndTime) : void +getStartMin() +setStartMin(StartMin) : void +getEndMin() +setEndMin(EndMin) : void +getFullTime() +setFullTime(FullTime) : void +getMaxSpace() +setMaxSpace(MaxSpace) : void +getBookedSpace() +setBookedSpace(BookedSpace) : void +getFreeSpace() +setFreeSpace(FreeSpace) : void +getWDNO() +setWDNO(WDNO) : void +getDeptNo() +setDeptNo(DeptNo) : void +LSECTION()

Η κλάση **LSECTION** περιέχει τα χαρακτηριστικά μια ομάδας παρακολούθησης μαθήματος: κωδικός ακαδημαϊκής περιόδου του τμήματος (**ACAD_PERIOD_ID**), κωδικός μαθήματος στο οποίο ανήκει η ομάδα παρακολούθησης (**LESSONCODE**), κωδικός ομάδας (**LSNO**), ώρα διεξαγωγής ομάδας (**STARTTIME**, **ENDTIME**, **STARTMIN**, **ENDMIN**, **FULLTIME**), μέγιστη χωρητικότητα ομάδας (**MAXSPACE**), δεσμευμένες θέσεις ομάδας (**BOOKEDSPACE**), ελεύθερες θέσεις ομάδας (**FREESPACE**), ημέρα της εβδομάδας που διεξάγεται η ομάδα παρακολούθησης του μαθήματος (**WDNO**), κωδικός τμήματος (**DEPTNO**).

SLVIEW
-ACADID
-AYEAR
-TERM
-DNO
-DNAME
-AEM
-LNAME
-FNAME
-SEMESTER
-EMAIL
-CODE
-TITLE
-SEMESTERNO
-LABEL
+getACADID()
+setACADID(ACADID) : void
+getAYEAR()
+setAYEAR(AYEAR) : void
+getTERM()
+setTERM(TERM) : void
+getDNO()
+setDNO(DNO) : void
+getDNAME()
+setDNAME(DNAME) : void
+getAEM()
+setAEM(AEM) : void
+getLNAME()
+setLNAME(LNAME) : void
+getFNAME()
+setFNAME(FNAME) : void
+getSEMESTER()
+setSEMESTER(SEMESTER) : void
+getEMAIL()
+setEMAIL(EMAIL) : void
+getCode()
+setCODE(CODE) : void
+getTitle()
+setTitle(TITLE) : void
+getSEMESTERNO()
+setSEMESTERNO(SEMESTERNO) : void
+getLabel()
+setLabel(LABEL) : void

Η κλάση **SLVIEW** περιέχει τα χαρακτηριστικά μια όψης που περιέχει στοιχεία φοιτητή και στοιχεία ενός μαθήματος που έχει δηλώσει στο γραμματειακό σύστημα. Τα χαρακτηριστικά αυτά είναι: κωδικός ακαδημαϊκής περιόδου (**ACADID**), έτος ακαδημαϊκής περιόδου (**AYEAR**), ακαδημαϊκή περίοδος (**TERM**), κωδικός τμήματος (**DNO**), όνομα τμήματος (**DNAME**), αριθμός μητρώου φοιτητή (**AEM**), επώνυμο (**LNAME**), όνομα (**FNAME**), εξάμηνο φοιτητή (**SEMESTER**), διεύθυνση ηλεκτρονικής αλληλογραφίας (**EMAIL**), κωδικός μαθήματος (**CODE**), τίτλος (**TITLE**), αριθμός εξαμήνου διδασκαλίας (**SEMESTERNO**), λεκτικό εξαμήνου διδασκαλίας (**LABEL**).

SLSVIEW
-DNO -DNAME -AEM -LNAME -SEMESTER -CODE -TITLE -SEMESTERNO -LSNO -STARTTIME -ENDTIME -FULLTIME -WDNAME
+getDNO() +setDNO(DNO) : void +getDNAME() +setDNAME(DNAME) : void +getAEM() +setAEM(AEM) : void +getLNAME() +setLNAME(LNAME) : void +getSEMESTER() +setSEMESTER(SEMESTER) : void +getCode() +setCODE(CODE) : void +getTitle() +setTitle(TITLE) : void +getSEMESTERNO() +setSEMESTERNO(SEMESTERNO) : void +getLSNO() +setLSNO(LSNO) : void +getSTARTTIME() +setSTARTTIME(STARTTIME) : void +getENDTIME() +setENDTIME(ENDTIME) : void +getFULLTIME() +setFULLTIME(FULLTIME) : void +getWDNAME() +setWDNAME(WDNAME) : void

Η κλάση **SLSVIEW** περιέχει τα χαρακτηριστικά μια όψης που περιέχει στοιχεία φοιτητή, στοιχεία τμήματος και στοιχεία μιας ομάδας παρακολούθησης ενός μαθήματος που έχει δηλώσει στο γραμματειακό σύστημα. Τα χαρακτηριστικά αυτά είναι: κωδικός τμήματος (**DNO**), όνομα τμήματος (**DNAME**), αριθμός μητρώου φοιτητή (**AEM**), επώνυμο (**LNAME**), όνομα (**FNAME**), εξάμηνο φοιτητή (**SEMESTER**), κωδικός μαθήματος (**CODE**), τίτλος (**TITLE**), εξάμηνο διδασκαλίας μαθήματος (**SEMESTERNO**), κωδικός ομάδας παρακολούθησης μαθήματος (**LSNO**), ώρες διεξαγωγής ομάδας μαθήματος (**STARTTIME, ENDTIME, FULLTIME, WDNAME**).

DbEjb
+registerSLS()
+findREG_SLS(): REG_SLS
+findRegisteredLessons(): List<Lesson>
+findRegisteredLessonsWithoutLSection(): List<Lesson>
+findRegisteredLessonsWithLSection(): List<Lesson>
+findLessonById(): Lesson
+findStudentByAem(): Student
+findStudentByAccount(): Student
+findStudentByUsername(): Student
+findLSectionsByAcadPeriod(): List<LSECTION>
+findLessonSectionsByView(): List<SLSVIEW>
+findLastAcadPeriod(): ACAD_PERIOD
+findLessonStudentsEmails(): List<String>
+findDepartmentChiefEmail(): String
+findLessonTeacherEmail(): String
+findDepartmentLessons(): List<Lesson>
+findAllDepts(): List<DEPT>
+findDeptById(): DEPT
+findStudentsCountByDept(): Long
+findTeachersCountByDept(): Long
+removeLSection()

Η κλάση **DbEjb** χρησιμοποιείται για τη διασύνδεση του υποσυστήματος με τη βάση δεδομένων του συστήματος: αναλαμβάνει την αναζήτηση δεδομένων από τη βάση δεδομένων και επιστρέφει τα δεδομένα σε μορφή αντικειμένων στο αντικείμενο της κλάσης **StudentWebController**. Αντίστροφα δέχεται αντικείμενα προς εισαγωγή, τροποποίηση και διαγραφή και πραγματοποιεί τις απαιτούμενες αλλαγές στη βάση δεδομένων. Οι λειτουργίες αυτές που σχετίζονται με τη βάση πραγματοποιούνται κάνοντας χρήση των παρακάτω μεθόδων.

Η μέθοδος **registerSLS** δέχεται ένα αντικείμενο **REG_SLS** που αντιστοιχεί σε μια δήλωση μιας ομάδας παρακολούθησης μαθήματος από φοιτητή και ενημερώνει τη βάση του συστήματος.

Η μέθοδος **findREG_SLS** βρίσκει τη δήλωση μιας ομάδας παρακολούθησης μαθήματος και επιστρέφει ένα αντικείμενο **REG_SLS**.

Η μέθοδος **findRegisteredLessons** επιστρέφει μια λίστα με τα μαθήματα που έχει δηλώσει ο φοιτητής στο γραμματειακό σύστημα.

Η μέθοδος **findRegisteredLessonsWithoutLSection** επιστρέφει μια λίστα με τα μαθήματα στα οποία ο φοιτητής δεν έχει δηλώσει ομάδα παρακολούθησης.

Η μέθοδος **findRegisteredLessonsWithLSection** επιστρέφει μια λίστα με τα μαθήματα στα οποία ο φοιτητής έχει δηλώσει ομάδα παρακολούθησης.

Η μέθοδος **findREG_SLS** βρίσκει τη δήλωση μιας ομάδας μαθήματος και επιστρέφει ένα αντικείμενο REG_SLS.

Η μέθοδος **findLessonById** επιστρέφει το μάθημα (αντικείμενο Lesson) με το Id της παραμέτρου.

Οι μέθοδοι **findStudentByAem**, **findStudentByAccount**, **findStudentByUsername** επιστρέφουν το φοιτητή με κριτήρια αναζήτησης αντίστοιχα το μητρώο, το όνομα χρήστη και κωδικό, το όνομα χρήστη.

Η μέθοδος **findLSectionsByAcadPeriod** επιστρέφει τις ομάδες παρακολούθησης μαθήματος για συγκεκριμένο μάθημα και συγκεκριμένη ακαδημαϊκή περίοδο.

Η μέθοδος **findLessonSectionsByView** επιστρέφει τις ομάδες παρακολούθησης μαθήματος για συγκεκριμένο φοιτητή σε λίστα αντικειμένων τύπου SLSVIEW.

Η μέθοδος **findLastAcadPeriod** επιστρέφει την τελευταία ακαδημαϊκή περίοδο ενός τμήματος.

Η μέθοδος **findLessonStudentsEmails** επιστρέφει τα emails των φοιτητών που έχουν δηλώσει συγκεκριμένο μάθημα σε συγκεκριμένη ακαδημαϊκή περίοδο.

Η μέθοδος **findDepartmentChiefEmail** επιστρέφει το email του προϊσταμένου συγκεκριμένου τμήματος.

Η μέθοδος **findLessonTeacherEmail** επιστρέφει το email του υπεύθυνου εκπαιδευτικού συγκεκριμένου μαθήματος.

Η μέθοδος **findDepartmentLessons** επιστρέφει λίστα με τα μαθήματα συγκεκριμένου τμήματος.

Η μέθοδος **findAllDepts** επιστρέφει λίστα με τα τμήματα του ιδρύματος.

Η μέθοδος **findDeptById** επιστρέφει το τμήμα με το id της παραμέτρου.

Η μέθοδος **findStudentsCountByDept** επιστρέφει το πλήθος των ενεργών φοιτητών συγκεκριμένου τμήματος.

Η μέθοδος **findTeachersCountByDept** επιστρέφει το πλήθος των εκπαιδευτικών συγκεκριμένου τμήματος.

Η μέθοδος **removeLSection** δέχεται ένα αντικείμενο τύπου REG_SLS (για δήλωση ομάδας παρακολούθησης μαθήματος) και τη διαγράφει από τη βάση δεδομένων.

StudentWebController
-connected -username -password -dbEJB -lastAcadPeriod -dept -student -lesson -lsection -reg_sls -slsview -fm -Title
+getConnection() +setConnected(connected) : void +getUsername() +setUsername(username) : void +getPassword() +setPassword(password) : void +getDbEJB() +setDbEJB(dbEJB) : void +getLastAcadPeriod() +setLastAcadPeriod(lastAcadPeriod) : void +getDept() +setDept(dept) : void +getStudent() +setStudent(student) : void +getLesson() +setLesson(lesson) : void +getLsection() +setLsection(lsection) : void +getReg_sls() +setReg_sls(reg_sls) : void +getSlsview() +setSlsview(slsview) : void +getFm() +setFm(fm) : void +getTitle() +setTitle(title) : void +doLogin() +doLogout() +doRemoveLSection() +doSelectLSection() +doSelectLesson() +getCellDataList() +isCrPeriodValid() +isRegistrationPeriodValid() +loadLastAcadPeriod() +loadTimeSheet()

Όλες οι προηγούμενες κλάσεις αποτελούν το τμήμα Model της αρχιτεκτονικής MVC. Όλες οι επόμενες κλάσεις αποτελούν το τμήμα View της αρχιτεκτονικής MVC. Η κλάση **StudentWebController** αποτελεί το ενδιάμεσο τμήμα Controller. Συνεπώς η εργασία της κλάσης αυτής είναι να δέχεται αιτήματα από το τμήμα View (κλάσεις που ακολουθούν) να τα μεταβιβάζει στο τμήμα Model και αν το τμήμα Model επιστρέφει δεδομένα/αντικείμενα από τη βάση δεδομένων ή από τον FetManager κλπ να τα προωθεί προς εμφάνιση στις κλάσεις του τμήματος View. Αρκετά από τα χαρακτηριστικά που εμφανίζονται στην κλάση StudentWebController έχουν προέλθει από το Model. Τέτοια χαρακτηριστικά/αντικείμενα είναι τα **lastAcadPeriod**, **dept**, **student**, **lesson**, **lsection**, **reg_sls**, **slsview**, **fm** (FetManager). Τα αντικείμενα αυτά μπορούν να χρησιμοποιηθούν απευθείας από τις κλάσεις του τμήματος View. Επιπλέον, το χαρακτηριστικό **dbEJB** χρησιμοποιείται για τη διασύνδεση του Controller με το Model. Το χαρακτηριστικό **connected** δείχνει αν ο χρήστης φοιτητής είναι συνδεδεμένος, ενώ τα **username** και **password** είναι τα στοιχεία που έχει δώσει ο χρήστης για να συνδεθεί.

Η μέθοδος **doLogin** επιχειρεί να αυθεντικοποιήσει ένα φοιτητή στον ldap server του ιδρύματος. Αρχικά θα πρέπει ο χρήστης να έχει δημιουργηθεί στη βάση δεδομένων του συστήματος και στη συνέχεια γίνεται αίτημα προς τον ldap. Αν τα στοιχεία σύνδεσης γίνουν δεκτά τότε φορτώνεται στο υποσύστημα η τελευταία ακαδημαϊκή περίοδος (`loadLastAcadPeriod`), το ωρολόγιο πρόγραμμα (`loadTimeSheet`) και ο google map (`loadMapBean`).

Η μέθοδος **doLogout** αποσυνδέει το χρήστη από το υποσύστημα φοιτητή.

Η μέθοδος **doRemoveLSection** διαγράφει τη δήλωση μιας ομάδας παρακολούθησης ενός μαθήματος από συγκεκριμένο φοιτητή και από συγκεκριμένη ακαδημαϊκή περίοδο.

Η μέθοδος **doSelectLSection** δέχεται ως παράμετρο τον κωδικό ενός μαθήματος και το κωδικό της ομάδας παρακολούθησης του μαθήματος και εισάγει τη δήλωση ομάδας παρακολούθησης του φοιτητή που είναι συνδεδεμένος στο υποσύστημα.

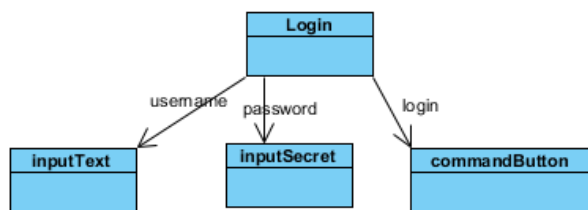
Η μέθοδος **doSelectLesson** δέχεται ως παράμετρο τον κωδικό ενός μαθήματος και επιστρέφει το αντικείμενο Lesson.

Η μέθοδος **isCrPeriodValid** ελέγχει αν η τρέχουσα ημερομηνία είναι περίοδος που είναι ενεργοποιημένο το κριτήριο προτεραιότητας εξαμήνου.

Η μέθοδος **isRegistrationPeriodValid** ελέγχει αν η τρέχουσα ημερομηνία είναι σε περίοδο δηλώσεων.

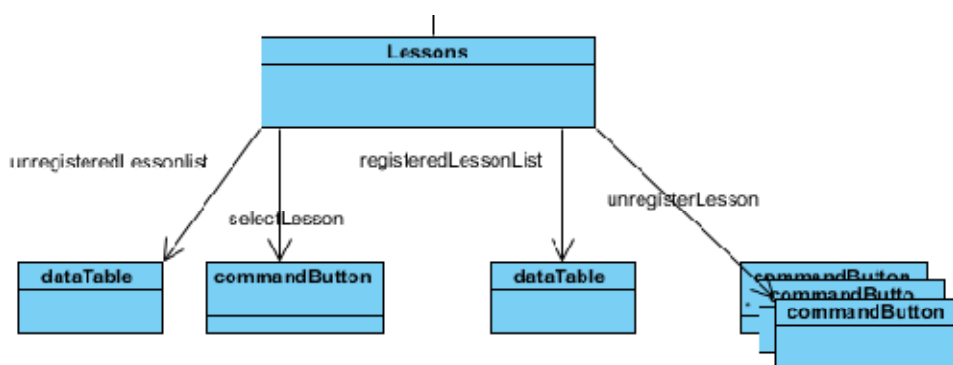
Η μέθοδος **loadLastAcadPeriod** επιστρέφει το αντικείμενο ACAD_PERIOD της τελευταίας ακαδημαϊκής περιόδου του τμήματος στο οποίο ανήκει ο συνδεδεμένος φοιτητής.

Η μέθοδος **loadTimeSheet** φορτώνει το ωρολόγιο πρόγραμμα του τμήματος στο οποίο ανήκει ο συνδεδεμένος φοιτητής.



Σχήμα A.1: : Διάγραμμα κλάσεων σελίδας σύνδεσης

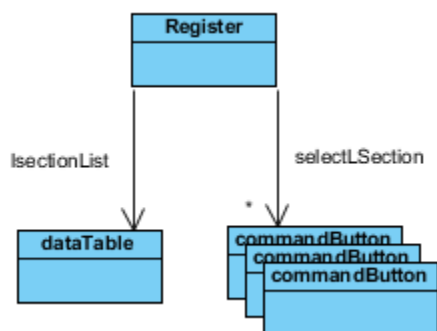
Η σελίδα **Login** (Σχήμα A.1) χρησιμοποιείται για τη σύνδεση του φοιτητή. Το όνομα χρήστη και και ο κωδικός συμπληρώνονται σε δυο αντικείμενα τύπου **inputText** και **inputSecret** αντίστοιχα (JSF Components). Όταν ο φοιτητής επιλέξει το αντικείμενο login τύπου **commandButton** τα στοιχεία σύνδεσης τοποθετούνται στα χαρακτηριστικά username,password του studentWebController και στη συνέχεια εκτελείται η μέθοδος doLogin (του studentWebController).



Σχήμα A.2: Διάγραμμα κλάσεων σελίδας μαθημάτων

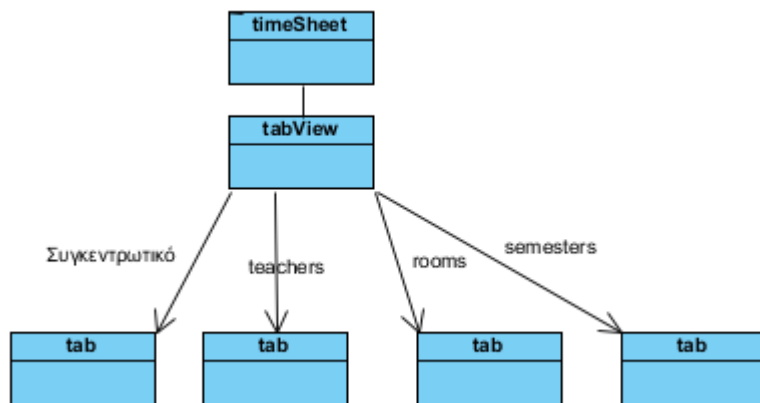
Στη σελίδα **Lessons** (Σχήμα A.2) εμφανίζονται τα μαθήματα που έχει δηλώσει ο φοιτητής στο γραμματειακό σύστημα. Από αυτά τα μαθήματα στο αντικείμενο **unregisteredLessonList** (τύπου dataTable) εμφανίζονται τα μαθήματα στα οποία δεν έχει δηλωθεί ομάδα

παρακολούθησης, ενώ στο αντικείμενο **registeredLessonList** εμφανίζονται τα μαθήματα και η ομάδα παρακολούθησης που έχει δηλωθεί. Το αντικείμενο **selectLesson** (τύπου `commandButton`) επιλέγει μάθημα ώστε σε επόμενη σελίδα να δηλωθεί ομάδα παρακολούθησης. Το αντικείμενο **unregisterLesson** ακυρώνει ομάδα παρακολούθησης.



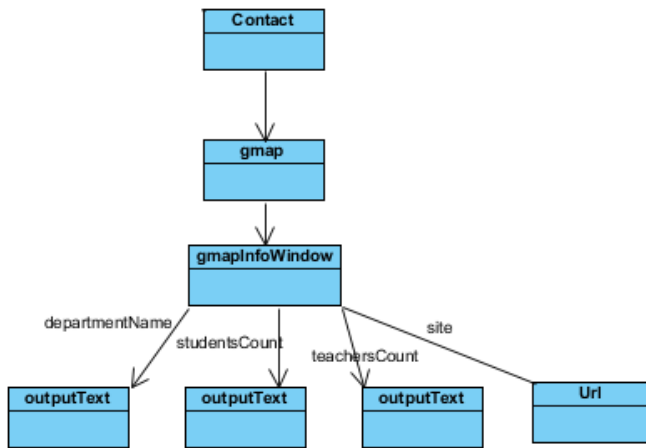
Σχήμα A.3: Διάγραμμα κλάσεων σελίδας δήλωσης ομάδας παρακολούθησης

Στη σελίδα **Register** (Σχήμα A.3) ο φοιτητής δηλώνει ομάδα παρακολούθησης μαθήματος. Το αντικείμενο **lsectionList** (τύπου `dataTable`) εμφανίζει τις ομάδες παρακολούθησης του επιλεγμένου μαθήματος. Όταν χρήστης επιλέγει το αντικείμενο **selectLSection** (τύπου `commandButton`) τότε εκτελείται η μέθοδος `doSelectLSection` (του `studentWebController`) που αποθηκεύει στη βάση δεδομένων την επιλεγμένη ομάδα παρακολούθησης.



Σχήμα A.4: Διάγραμμα κλάσεων σελίδας προβολής ωρολογίου προγράμματος

Η σελίδα **timeSheet** (Σχήμα A.4) περιέχει το ωρολόγιο πρόγραμμα ενός τμήματος. Το περιεχόμενο προβάλλεται σε ένα αντικείμενο τύπου `tabView`. Το ωρολόγιο πρόγραμμα έχει τέσσερις προβολές (αντικείμενα τύπου `tab`) : συγκεντρωτική, προβολή ανά εκπαιδευτικό, ανά αίθουσα και ανά εξάμηνο διδασκαλίας.



Σχήμα A.5: Διάγραμμα κλάσεων σελίδας επικοινωνίας

Η σελίδα **Contact** (Σχήμα A.5) περιέχει ένα αντικείμενο google map τύπου **gmap** το οποίο έχει ως σημείο ενδιαφέροντος το γεωγραφικό πλάτος και μήκος του τμήματος που ανήκει ο φοιτητής. Όταν ο φοιτητής επιλέξει το σημείο ενδιαφέροντος εμφανίζεται ένα παράθυρο popup τύπου **gmapInfoWindow** το οποίο περιέχει το όνομα του τμήματος, το πλήθος των φοιτητών, των εκπαιδευτικών (αντικείμενα τύπου **outputText**) και τον υπερσύνδεσμο που οδηγεί στην ιστοσελίδα του τμήματος.



Η κλάση **LdapManager** ασχολείται με την αυθεντικοποίηση των χρηστών στον LDAP Server του ιδρύματος. Η κλάση χρησιμοποιεί τη βιβλιοθήκη της Novell [24] για τη διαχείριση οποιουδήποτε open ldap server [25]. Στο χαρακτηριστικό **LDAP_HOST** βάζουμε τη διεύθυνση του LDAP server, ενώ στο χαρακτηριστικό **LOGIN_DN** το χρήστη που θα συνδεθεί. Το χαρακτηριστικό **ldapCon** έχει το αντικείμενο σύνδεσης με τον LDAP Server. Οι μέθοδοι **connect**, **disconnect** επιχειρούν σύνδεση και αποσύνδεση από τον LDAP.

Η κλάση **FetManager** περιγράφεται στο υποσύστημα ωρολογίου προγράμματος.

A.2 Περιγραφή κλάσεων υποσυστήματος εκπαιδευτικών

Οι κλάσεις **ACAD_PERIOD**, **DEPT**, **TEACHER**, **Student**, **Lesson**, **LSection**, **SLVIEW**, **SLSVIEW**, **LdapManager** καθώς και η σελίδα **Contact** είναι ίδιες με αυτές του υποσυστήματος φοιτητών.

Η κλάση **FetManager** περιγράφεται στο υποσύστημα ωρολογίου προγράμματος.

TLSVIEW
-FNAME -LNAME -DEPTNO -UNAME -LESSONCODE -TITLE -ROOM_NAME -ACAD_PERIOD_ID -LSNO
+TLSVIEW() +getFNAME() +setFNAME(FNAME) : void +getLNAME() +setLNAME(LNAME) : void +getDEPTNO() +setDEPTNO(DEPTNO) : void +getUNAME() +setUNAME(UNAME) : void +getLESSONCODE() +setLESSONCODE(LESSONCODE) : void +getTitle() +setTitle(TITLE) : void +getROOM_NAME() +setROOM_NAME(ROOM_NAME) : void +getACAD_PERIOD_ID() +setACAD_PERIOD_ID(ACAD_PERIOD_ID) : void +getLSNO() +setLSNO(LSNO) : void

Η κλάση **TLSVIEW** περιέχει τα χαρακτηριστικά μια όψης που περιέχει στοιχεία εκπαιδευτικού, μαθήματος και στοιχεία μιας ομάδας παρακολούθησης ενός μαθήματος. Τα χαρακτηριστικά αυτά είναι: όνομα (**FNAME**), επώνυμο (**LNAME**), κωδικός τμήματος (**DEPTNO**), όνομα λογαριασμού (**UNAME**), κωδικός μαθήματος (**LESSONCODE**), τίτλος (**TITLE**), αίθουσα διδασκαλίας (**ROOM_NAME**), κωδικός ακαδημαϊκής περιόδου (**ACAD_PERIOD_ID**) και κωδικός ομάδας παρακολούθησης μαθήματος (**LSNO**).

DbEJB
+createDEPT()
+findAllDepts()
+findDeptById()
+removeDeptById()
+updateDeptById()
+createTEACHER()
+removeTeacherById()
+updateTeacherById()
+findDepartmentTeachers()
+findTeacherByAccount()
+findTeachersCountByDept()
+createLESSON()
+removeLessonById()
+updateLessonById()
+findDepartmentLessons()
+createLSECTION()
+removeLSectionById()
+updateLSectionById()
+findLessonLSections()
+findAllSemesters()
+findAllWeekDays()
+findDepartmentAcadPeriods()
+findLastAcadPeriod()
+updateAcadPeriod()
+findDepartmentRooms()
+findTLVIEW()
+findStudentsCountByDept()
+findLessonStudents()
+findLSectionStudents()

Η κλάση **DbEjb** χρησιμοποιείται για τη διασύνδεση του υποσυστήματος με τη βάση δεδομένων του συστήματος: αναλαμβάνει την αναζήτηση δεδομένων από τη βάση δεδομένων και επιστρέφει τα δεδομένα σε μορφή αντικειμένων στο αντικείμενο τύπου `TeacherWebController`. Αντίστροφα δέχεται αντικείμενα προς εισαγωγή, τροποποίηση και διαγραφή και πραγματοποιεί τις απαιτούμενες αλλαγές στη βάση δεδομένων. Οι λειτουργίες αυτές που σχετίζονται με τη βάση πραγματοποιούνται κάνοντας χρήση των παρακάτω μεθόδων.

Η μέθοδος **createDEPT** εισάγει ένα τμήμα στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών.

Η μέθοδος **findAllDepts** επιστρέφει από τη βάση μια λίστα με τα τμήματα του ιδρύματος.

Η μέθοδος **findDeptById** επιστρέφει το τμήμα που έχει κωδικό το `Id` της παραμέτρου.

Η μέθοδος **removeDeptById** διαγράφει το τμήμα που έχει κωδικό το `Id` της παραμέτρου.

Η μέθοδος **updateDeptById** ενημερώνει το τμήμα με βάση τις τιμές που έχει το αντικείμενο της παραμέτρου.

Η μέθοδος **createTEACHER** εισάγει στη βάση έναν εκπαιδευτικό σε ένα τμήμα.

Η μέθοδος **removeTeacherById** διαγράφει τον εκπαιδευτικό που έχει κωδικό το Id της παραμέτρου.

Η μέθοδος **updateTeacherById** ενημερώνει τον εκπαιδευτικό με βάση τις τιμές που έχει το αντικείμενο της παραμέτρου.

Η μέθοδος **findDepartmentTeachers** δέχεται ως παράμετρο τον κωδικό ενός τμήματος και επιστρέφει μια λίστα με τους εκπαιδευτικούς του τμήματος.

Η μέθοδος **findTeacherByAccount** επιστρέφει τον εκπαιδευτικό που έχει το όνομα του λογαριασμού και τον κωδικό της παραμέτρου.

Η μέθοδος **findTeachersCountByDept** επιστρέφει το πλήθος των εκπαιδευτικών του τμήματος της παραμέτρου.

Η μέθοδος **createLESSON** εισάγει στη βάση ένα μάθημα σε ένα τμήμα.

Η μέθοδος **removeLessonById** διαγράφει το μάθημα που έχει κωδικό το Id της παραμέτρου.

Η μέθοδος **updateLessonById** ενημερώνει το μάθημα με βάση τις τιμές που έχει το αντικείμενο της παραμέτρου.

Η μέθοδος **findDepartmentLessons** επιστρέφει λίστα με τα μαθήματα του τμήματος της παραμέτρου.

Η μέθοδος **createLSECTION** εισάγει στη βάση μια ομάδα παρακολούθησης σε ένα μάθημα.

Η μέθοδος **removeLSectionById** διαγράφει την ομάδα παρακολούθησης που έχει κωδικό το Id της παραμέτρου.

Η μέθοδος **updateLSectionById** ενημερώνει μια ομάδα παρακολούθησης με βάση τις τιμές που έχει το αντικείμενο της παραμέτρου.

Η μέθοδος **findLessonLSections** επιστρέφει λίστα με τις ομάδες παρακολούθησης του μαθήματος της παραμέτρου.

Η μέθοδος **findAllSemesters** επιστρέφει λίστα με τα εξάμηνα.

Η μέθοδος **findAllWeekDays** επιστρέφει λίστα με τις ημέρες της εβδομάδος.

Η μέθοδος **findDepartmentAcadPeriods** επιστρέφει λίστα με τις ακαδημαϊκές περιόδους του τμήματος της παραμέτρου.

Η μέθοδος **findLastAcadPeriod** επιστρέφει την τελευταία ακαδημαϊκή περίοδο του τμήματος της παραμέτρου.

Η μέθοδος **updateAcadPeriod** ενημερώνει την ακαδημαϊκή περίοδο με βάση τις τιμές που έχει το αντικείμενο της παραμέτρου.

Η μέθοδος **findDepartmentRooms** επιστρέφει λίστα με τις αίθουσες του τμήματος της παραμέτρου.

Η μέθοδος **findTLSVIEW** επιστρέφει λίστα καθηγητών με τα μαθήματα και τις ομάδες παρακολούθησης της ακαδημαϊκής περιόδου της παραμέτρου.

Η μέθοδος **findStudentsCountByDept** επιστρέφει το πλήθος των φοιτητών του τμήματος της παραμέτρου.

Η μέθοδος **findLessonStudents** επιστρέφει τους φοιτητές που έχουν δηλώσει σε συγκεκριμένη ακαδημαϊκή περίοδο το μάθημα της παραμέτρου.

Η μέθοδος **findLSectionStudents** επιστρέφει τους φοιτητές που έχουν δηλώσει σε συγκεκριμένη ακαδημαϊκή περίοδο την ομάδα παρακολούθησης της παραμέτρου.

TeacherWebController
-connected -username -password -dbEJB -mapBean -lastAcadPeriod -department -teacher -lesson -lsection -acad_period -NewAcadPeriodCurrentTask -NewAcadPeriodProcessStatus -SyncPeriodRegistrationsProcessStatus -fm -fetExecuteOutput
+TeacherWebController() +getConnectioned() +setConnected(connected) : void +getUsername() +setUsername(username) : void +getPassword() +setPassword(password) : void +getDbEJB() +setDbEJB(dbEJB) : void +getMapBean() +setMapBean(mapBean) : void +getLastAcadPeriod() +setLastAcadPeriod(lastAcadPeriod) : void +getDepartment() +setDepartment(department) : void +getTeacher() +setTeacher(teacher) : void +getLesson() +setLesson(lesson) : void +getLsection() +setLsection(lsection) : void +getAcad_period() +setAcad_period(acad_period) : void +getNewAcadPeriodCurrentTask() +setNewAcadPeriodCurrentTask(NewAcadPeriodCurrentTask) : void +getNewAcadPeriodProcessStatus() +setNewAcadPeriodProcessStatus(NewAcadPeriodProcessStatus) : void +getSyncPeriodRegistrationsProcessStatus() +setSyncPeriodRegistrationsProcessStatus(SyncPeriodRegistrationsProcessStatus) : void +getFm() +setFm(fm) : void +getFetExecuteOutput() +setFetExecuteOutput(fetExecuteOutput) : void +doLogin() +prepareInsertDepartment() +prepareUpdateDepartment() +prepareDeleteDepartment() +doInsertDepartment() +doUpdateDepartment() +doDeleteDepartment() +prepareInsertTeacher() +prepareUpdateTeacher() +prepareDeleteTeacher() +doInsertTeacher() +doUpdateTeacher() +doDeleteTeacher() +selectTeacherDepartment() +selectTimeSheetDepartment() +prepareInsertLesson() +prepareUpdateLesson() +prepareDeleteLesson() +doInsertLesson() +doUpdateLesson() +doDeleteLesson() +selectLessonDepartment() +prepareInsertLSection() +prepareUpdateLSection() +prepareDeleteLSection() +doInsertLSection() +doUpdateLSection() +doDeleteLSection() +selectAcadPeriodDepartment() +loadLastAcadPeriod() +doUpdateLastAcadPeriodLesson() +loadTimeSheet() +generateFetInput() +generateTimetable() +chiefRequestPreparePeriod() +chiefNewAcadPeriodGetCurrentAsyncTask() +requestSyncPeriodRegistrations() +getLessonGroupsFullRequest() +CompleteProfessorGroupUpdateRequest() +CompleteChiefResponse()

Όλες οι προηγούμενες κλάσεις αποτελούν το τμήμα Model της αρχιτεκτονικής MVC. Όλες οι επόμενες κλάσεις αποτελούν το τμήμα View της αρχιτεκτονικής MVC. Η κλάση **TeacherWebController** αποτελεί το ενδιάμεσο τμήμα Controller. Συνεπώς η εργασία της κλάσης αυτής είναι να δέχεται αιτήματα από το τμήμα View (κλάσεις που ακολουθούν) να τα μεταβιβάζει στο τμήμα Model και αν το τμήμα Model επιστρέφει δεδομένα/αντικείμενα από τη βάση δεδομένων ή από τον FetManager κλπ να τα προωθεί προς εμφάνιση στις κλάσεις του τμήματος View. Αρκετά από τα χαρακτηριστικά που εμφανίζονται στην κλάση TeacherWebController έχουν προέλθει από το Model. Τέτοια χαρακτηριστικά/αντικείμενα είναι τα **lastAcadPeriod, department, teacher, lesson, lsection, acadperiod, fm** (FetManager). Τα αντικείμενα αυτά μπορούν να χρησιμοποιηθούν απευθείας από τις κλάσεις του τμήματος View. Επιπλέον, το χαρακτηριστικό **dbEJB** χρησιμοποιείται για τη διασύνδεση του Controller με το Model. Το χαρακτηριστικό **connected** δείχνει αν ο χρήστης φοιτητής είναι συνδεδεμένος, ενώ τα **username** και **password** είναι τα στοιχεία που έχει δώσει ο χρήστης για να συνδεθεί. Το χαρακτηριστικό **fetExecuteOutput** επιστρέφει πληροφορίες εξόδου κατά την αυτόματη δημιουργία του ωρολογίου προγράμματος από το εξωτερικό υποσύστημα FET. Τα χαρακτηριστικά **NewAcadPeriodCurrentTask** και **NewAcadPeriodProcessStatus** περιέχουν το όνομα της τρέχουσας εργασίας (task) και την κατάσταση της BPM διεργασίας προετοιμασίας νέας ακαδημαϊκής περιόδου ώστε ο προϊστάμενος ενός τμήματος να βλέπει την εξέλιξη της διεργασίας. Το χαρακτηριστικό **SyncPeriodRegistrationsProcessStatus** περιέχει την κατάσταση της BPM διεργασίας συγχρονισμού δηλώσεων ακαδημαϊκής περιόδου.

Η μέθοδος **doLogin** επιχειρεί να συνδέσει τον εκπαιδευτικό ή τον διαχειριστή στο υποσύστημα. Αν η σύνδεση είναι επιτυχής και επιπλέον ο χρήστης είναι εκπαιδευτικός φορτώνεται η τελευταία ακαδημαϊκή περίοδος, προετοιμάζεται η προβολή του ωρολογίου προγράμματος, και προετοιμάζεται ο google map (loadMapBean).

Η μέθοδος **prepareInsertDepartment** προβάλλει τη σελίδα εισαγωγής νέου τμήματος.

Η μέθοδος **prepareUpdateDepartment** προβάλλει τη σελίδα ενημέρωσης τμήματος.

Η μέθοδος **prepareDeleteDepartment** προβάλλει τη σελίδα διαγραφής τμήματος.

Η μέθοδος **doInsertDepartment** εισάγει ένα τμήμα με βάση τα στοιχεία εισόδου της σελίδας.

Η μέθοδος **doUpdateDepartment** ενημερώνει τα στοιχεία ενός τμήματος με βάση τα στοιχεία εισόδου της σελίδας.

Η μέθοδος **doDeleteDepartment** διαγράφει το επιλεγμένο τμήμα.

Η μέθοδος **prepareInsertTeacher** προβάλλει τη σελίδα εισαγωγής νέου εκπαιδευτικού.

Η μέθοδος **prepareUpdateTeacher** προβάλλει τη σελίδα ενημέρωσης εκπαιδευτικού.

Η μέθοδος **prepareDeleteTeacher** προβάλλει τη σελίδα διαγραφής εκπαιδευτικού.

Η μέθοδος **doInsertTeacher** εισάγει έναν εκπαιδευτικό στο επιλεγμένο τμήμα με βάση τα στοιχεία εισόδου της σελίδας.

Η μέθοδος **doUpdateTeacher** ενημερώνει τα στοιχεία ενός εκπαιδευτικού με βάση τα στοιχεία εισόδου της σελίδας.

Η μέθοδος **doDeleteTeacher** διαγράφει τον επιλεγμένο εκπαιδευτικό.

Η μέθοδος **selectTeacherDepartment** επιλέγει το τμήμα του εκπαιδευτικού.

Η μέθοδος **selectTimesheetDepartment** επιλέγει το ωρολόγιο πρόγραμμα του τμήματος του εκπαιδευτικού.

Η μέθοδος **prepareInsertLesson** προβάλλει τη σελίδα εισαγωγής νέου μαθήματος.

Η μέθοδος **prepareUpdateLesson** προβάλλει τη σελίδα ενημέρωσης μαθήματος.

Η μέθοδος **prepareDeleteLesson** προβάλλει τη σελίδα διαγραφής μαθήματος.

Η μέθοδος **doInsertLesson** εισάγει ένα μάθημα στο επιλεγμένο τμήμα με βάση τα στοιχεία εισόδου της σελίδας.

Η μέθοδος **doUpdateLesson** ενημερώνει τα στοιχεία ενός μαθήματος με βάση τα στοιχεία εισόδου της σελίδας.

Η μέθοδος **doDeleteLesson** διαγράφει το επιλεγμένο μάθημα.

Η μέθοδος **selectLessonDepartment** επιλέγει ένα μάθημα.

Η μέθοδος **prepareInsertLSection** προβάλλει τη σελίδα εισαγωγής νέας ομάδας παρακολούθησης σε ένα μάθημα.

Η μέθοδος **prepareUpdateLSection** προβάλλει τη σελίδα ενημέρωσης μιας ομάδας παρακολούθησης.

Η μέθοδος **prepareDeleteLSection** προβάλλει τη σελίδα διαγραφής μιας ομάδας παρακολούθησης.

Η μέθοδος **doInsertLSection** εισάγει μιας ομάδα παρακολούθησης στο επιλεγμένο μάθημα με βάση τα στοιχεία εισόδου της σελίδας.

Η μέθοδος **doUpdateLSection** ενημερώνει τα στοιχεία μιας ομάδας παρακολούθησης μαθήματος με βάση τα στοιχεία εισόδου της σελίδας.

Η μέθοδος **doDeleteLSection** διαγράφει την επιλεγμένη ομάδα παρακολούθησης από κάποιο μάθημα.

Η μέθοδος **selectAcadPeriodDepartment** επιλέγει ακαδημαϊκή περίοδο τμήματος.

Η μέθοδος **loadLastAcadPeriod** βρίσκει την τελευταία ακαδημαϊκή περίοδο ενός τμήματος.

Η μέθοδος **doUpdateLastAcadPeriodLesson** ενημερώνει την τελευταία ακαδημαϊκή περίοδο με βάση τα στοιχεία της σελίδας.

Η μέθοδος **loadTimeSheet** φορτώνει το ωρολόγιο πρόγραμμα συγκεκριμένου τμήματος.

Η μέθοδος **generateFetInput** προετοιμάζει σε XML μορφή τα στοιχεία εισόδου και εκτελεί το εξωτερικό σύστημα FET ώστε να παρόχθιε το ωρολόγιο πρόγραμμα. Οι πληροφορίες αποτελεσμάτων της εκτέλεσης τοποθετούνται στο χαρακτηριστικό `fetExecuteOutput`.

Η μέθοδος **generateTimetable** προβάλλει το ωρολόγιο πρόγραμμα καλώντας τις μεθόδους `generateFetInput` και `loadTimeSheet`.

Η μέθοδος **chiefRequestPreparePeriod** δημιουργεί αίτημα προετοιμασίας νέας ακαδημαϊκής περιόδου, εκκινώντας τη BPM διεργασία **NewAcadPeriodProcess** (προετοιμασία νέας ακαδημαϊκής περιόδου). Η αίτηση υποβάλλεται κάνοντας Rest Client κλήση (**Create** method) στο υποσύστημα διεργασιών.

Η μέθοδος **chiefNewAcadPeriodGetCurrentAsyncTask** επιστρέφει την τρέχουσα εργασία (BPM process Task) και την κατάσταση της BPM διεργασίας (NewAcadPeriodProcess). Οι πληροφορίες βρίσκονται κάνοντας Rest Client κλήσεις (**GetCurrentAsyncTask** , **CompletedProcessStatus** methods) στο υποσύστημα διεργασιών.

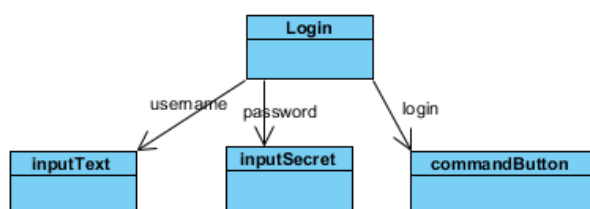
Η μέθοδος **requestSyncPeriodRegistrations** δημιουργεί αίτημα συγχρονισμού δηλώσεων επιλεγμένης ακαδημαϊκής περιόδου, εκκινώντας τη BPM διεργασία **SyncPeriodRegistrationsProcess**. Η αίτηση υποβάλλεται κάνοντας Rest Client κλήση (**Create** method) στο υποσύστημα διεργασιών.

Η μέθοδος **getSyncPeriodRegistrationsProcessStatus** επιστρέφει το τρέχον BPM process Task της διεργασίας **SyncPeriodRegistrationsProcess**. Η πληροφορία βρίσκεται κάνοντας Rest Client κλήση (**GetCurrentAsyncTask** method) στο υποσύστημα διεργασιών.

Η μέθοδος **getLessonGroupsFullRequest** επιστρέφει το τρέχον BPM process Task της διεργασίας **LessonGroupsFullProcess** (κορεσμού ομάδων παρακολούθησης μαθήματος). Η πληροφορία βρίσκεται κάνοντας Rest Client κλήση (**GetCurrentAsyncTask** method) στο υποσύστημα διεργασιών.

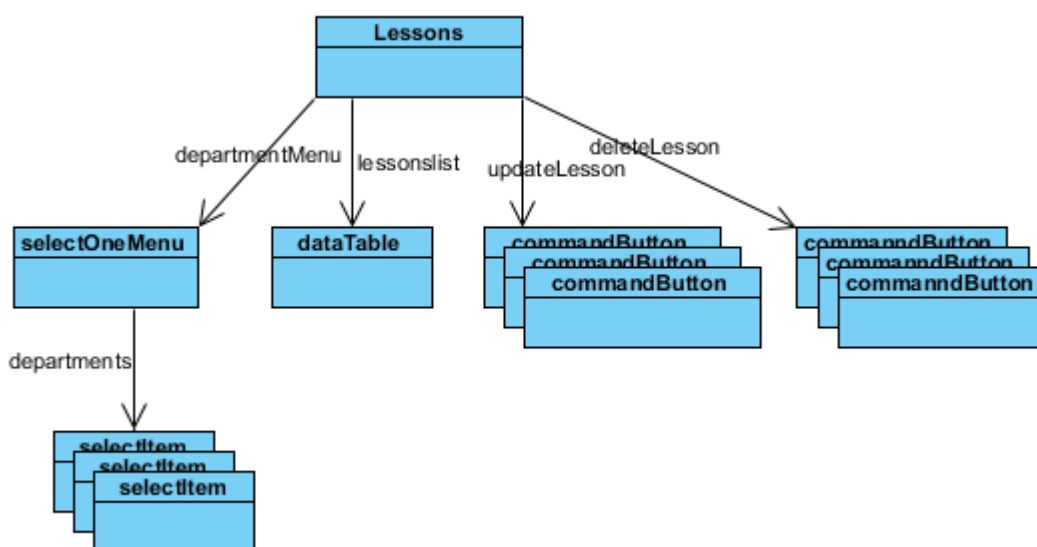
Η μέθοδος **CompleteProfessorGroupUpdateRequest** ολοκληρώνει το BPM process Task **CompleteProfessorGroupUpdateRequestAsyncTask** της διεργασίας **LessonGroupsFullProcess** καλώντας τη Rest method **CompleteProfessorGroupUpdateRequestAsyncTask** από το υποσύστημα διεργασιών.

Η μέθοδος **CompleteChiefResponse** ολοκληρώνει το BPM process Task **CompleteChiefResponseAsyncTask** της διεργασίας **LessonGroupsFullProcess** καλώντας τη Rest μέθοδο **CompleteChiefResponseAsyncTask** από το υποσύστημα διεργασιών.



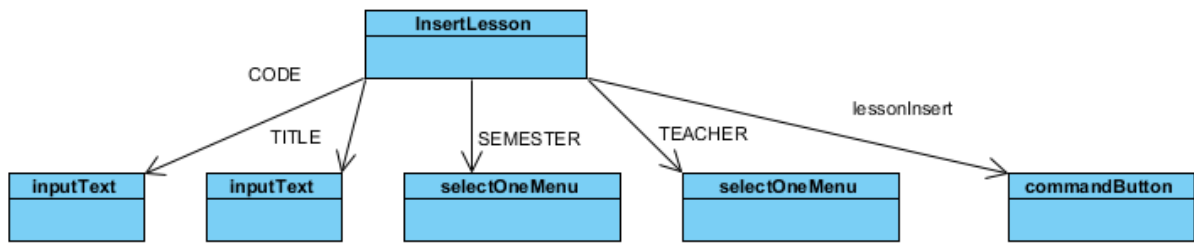
Σχήμα A.6: Διάγραμμα κλάσεων σελίδας σύνδεσης

Η σελίδα **Login** (Σχήμα A.6) χρησιμοποιείται για τη σύνδεση του εκπαιδευτικού ή του διαχειριστή. Το όνομα χρήστη και ο κωδικός συμπληρώνονται σε δυο αντικείμενα τύπου **inputText** και **inputSecret** αντίστοιχα (JSF Components). Όταν ο εκπαιδευτικός επιλέξει το αντικείμενο **login** τύπου **commandButton** τα στοιχεία σύνδεσης τοποθετούνται στα χαρακτηριστικά **username,password** του **teacherWebController** και στη συνέχεια εκτελείται η μέθοδος **doLogin** (του **teacherWebController**).



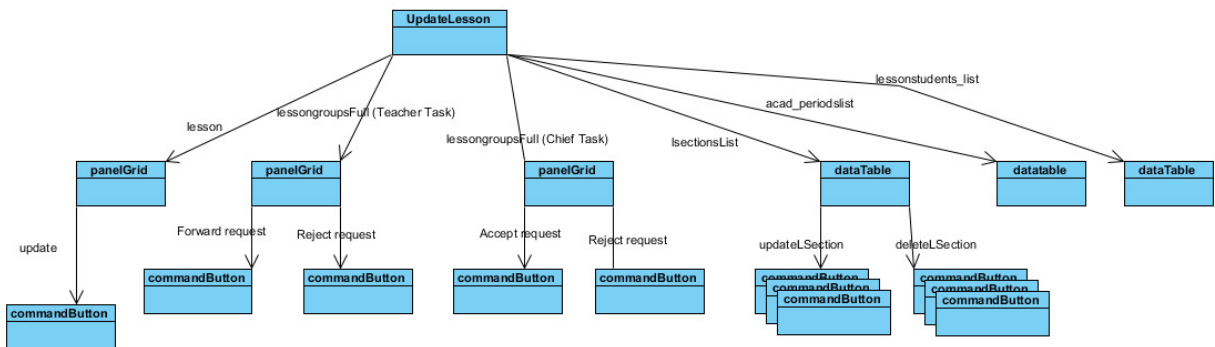
Σχήμα A.7: Διάγραμμα κλάσεων σελίδας μαθημάτων

Στη σελίδα **Lessons** (Σχήμα A.7) εμφανίζονται το τμήμα του εκπαιδευτικού (αντικείμενο τύπου **selectItem**) από το σύνολο των τμημάτων του ιδρύματος (αντικείμενο τύπου **selectOneMenu**) και τα μαθήματα του τμήματος (αντικείμενο τύπου **dataTable**). Το αντικείμενο **updateLesson** (τύπου **commandButton**) επιλέγει μάθημα προς επεξεργασία. Το αντικείμενο **deleteLesson** (τύπου **commandButton**) επιλέγει μάθημα προς διαγραφή.



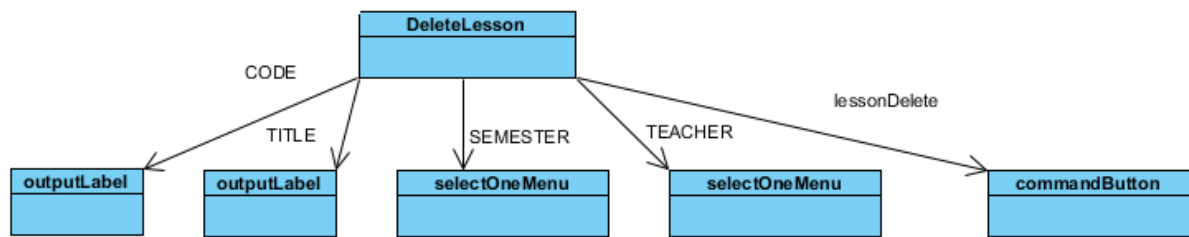
Σχήμα A.8: Διάγραμμα κλάσεων σελίδας εισαγωγής μαθήματος

Η σελίδα **InsertLesson** (Σχήμα A.8) εισάγει στη βάση δεδομένων του συστήματος νέο μάθημα σε συγκεκριμένο τμήμα με κωδικό (αντικείμενο **CODE**), όνομα (αντικείμενο **TITLE**), εξάμηνο (αντικείμενο **SEMESTER**) και υπεύθυνο εκπαιδευτικό (αντικείμενο **TEACHER**). Η εισαγωγή εκτελείται όταν ο χρήστης επιλέξει το αντικείμενο **lessonInsert** (τύπου **commandButton**).



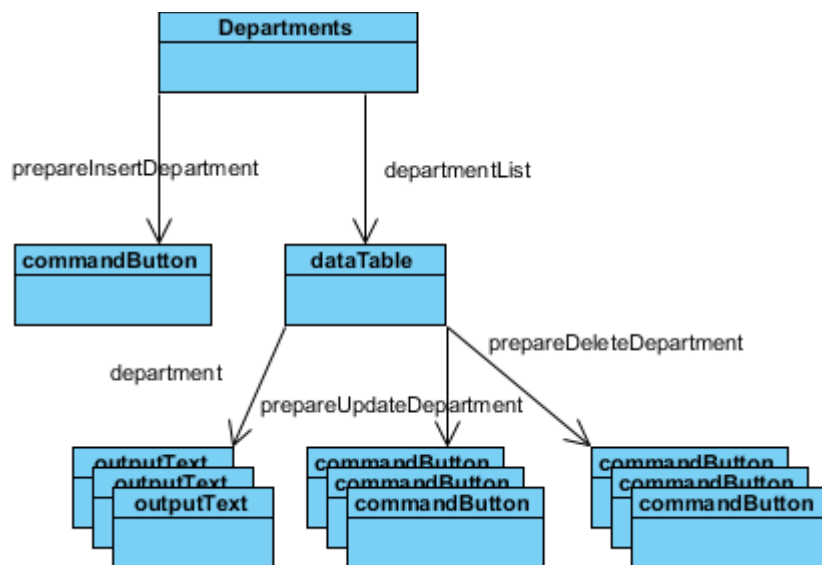
Σχήμα A.9: Διάγραμμα κλάσεων σελίδας ενημέρωσης μαθήματος

Στη σελίδα **UpdateLesson** (Σχήμα A.9) γίνονται τα εξής: Στο αντικείμενο **lesson** (τύπου **panelGrid**) εμφανίζονται τα στοιχεία του μαθήματος, όπου ο χρήστης μπορεί να ενημερώσει με νέα στοιχεία επιλέγοντας το αντικείμενο **update** (τύπου **commandButton**). Στο αντικείμενο **lessonstudents_list** ο εκπαιδευτικός βλέπει ποιοι φοιτητές έχουν δηλώσει το συγκεκριμένο μάθημα την τελευταία ακαδημαϊκή περίοδο. Στο αντικείμενο **acad_periodslist** ο εκπαιδευτικός βλέπει ιστορικό επιτυχιών ανά ακαδημαϊκή περίοδο. Στο αντικείμενο **lsections_list** ο προϊστάμενος μπορεί να δει και να επιλέξει για τροποποίηση (**updateLSection**) ή διαγραφή (**deleteLSection**) τις ομάδες παρακολούθησης του μαθήματος. Στο αντικείμενο **lessongroupsFull (Teacher Task)** ο υπεύθυνος εκπαιδευτικός του μαθήματος μπορεί να προωθήσει (**forward request**) στον προϊστάμενο ή να απορρίψει τη BPM διεργασία κορεσμού ομάδων παρακολούθησης μαθήματος. Στο αντικείμενο **lessongroupsFull (Chief Task)** ο προϊστάμενος μπορεί να αποδεχτεί (**accept request**) ή να απορρίψει το αίτημα (BPM process) κορεσμού ομάδων παρακολούθησης.



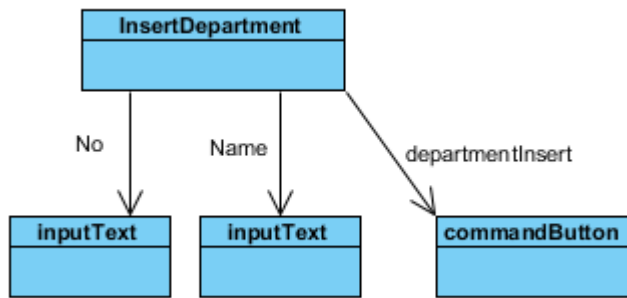
Σχήμα A.10: Διάγραμμα κλάσεων σελίδας διαγραφής μαθήματος

Η σελίδα **DeleteLesson** (Σχήμα A.10) διαγράφει από τη βάση δεδομένων του συστήματος το επιλεγμένο μάθημα με τα στοιχεία που προβάλλονται: κωδικός (αντικείμενο **CODE** τύπου outputLabel), όνομα (αντικείμενο **TITLE**), εξάμηνο (αντικείμενο **SEMESTER**) και υπεύθυνο εκπαιδευτικό (αντικείμενο **TEACHER**). Η εισαγωγή εκτελείται όταν ο χρήστης επιλέξει το αντικείμενο **lessonDelete** (τύπου commandButton).



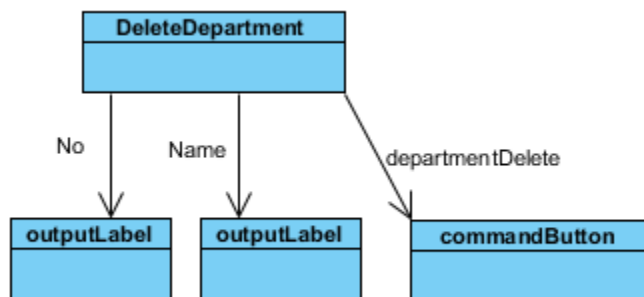
Σχήμα A.11: Διάγραμμα κλάσεων σελίδας τμημάτων

Η σελίδα **Departments** (Σχήμα A.11) προβάλλει τα τμήματα του ιδρύματος. Το αντικείμενο **prepareInsertDepartment** (τύπου commandButton) όταν επιλεγεί ανοίγει τη σελίδα InsertDepartment. Το αντικείμενο **departmentList** (τύπου dataTable) προβάλλει τα τμήματα και δίπλα σε κάθε τμήμα υπάρχουν τα αντικείμενα **prepareUpdateDepartment** και **prepareDeleteDepartment** που όταν επιλεγούν ανοίγουν οι σελίδες UpdateDepartment και DeleteDepartment αντίστοιχα.



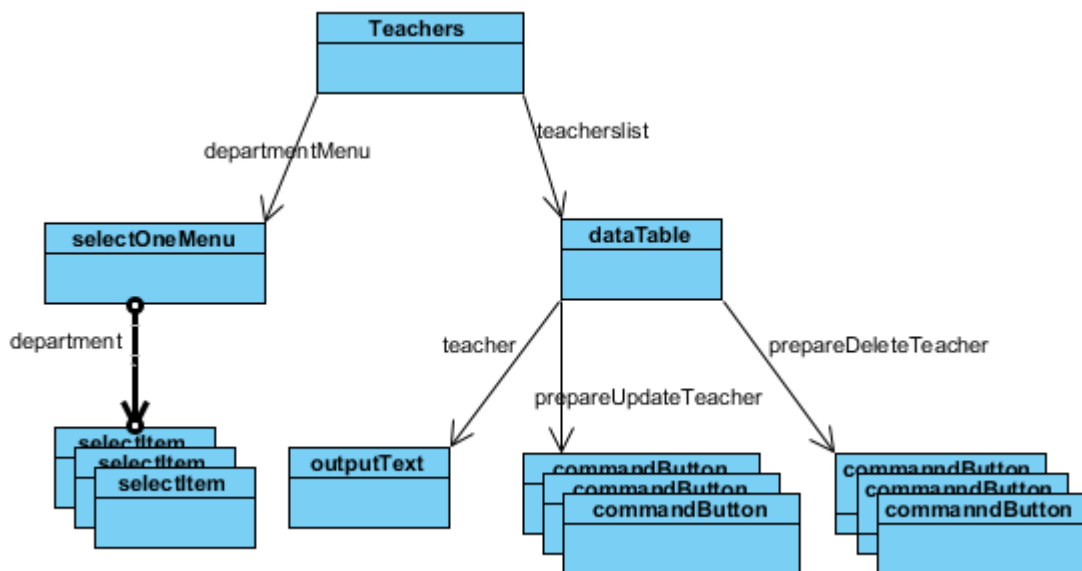
Σχήμα A.12: Διάγραμμα κλάσεων σελίδας εισαγωγής τμήματος

Η σελίδα **InsertDepartment** (Σχήμα A.12) εισάγει στη βάση δεδομένων του συστήματος νέο τμήμα με κωδικό και όνομα αυτά που έχει πληκτρολογήσει ο χρήστης στα αντικείμενα **No** και **Name** (τύπου `inputText`). Η εισαγωγή εκτελείται όταν ο χρήστης επιλέξει το αντικείμενο **departmentInsert** (τύπου `commandButton`).



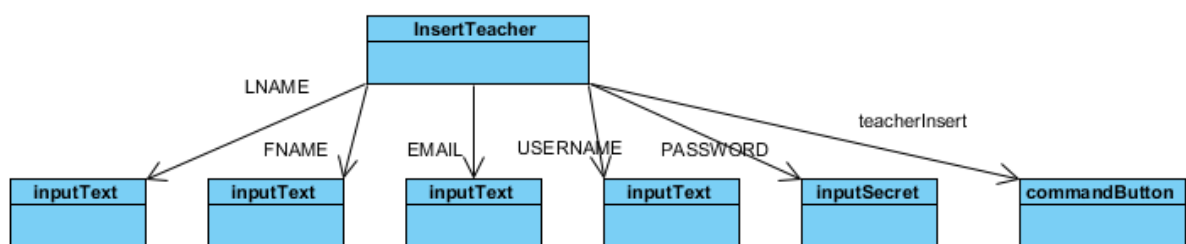
Σχήμα A.13: Διάγραμμα κλάσεων σελίδας διαγραφής τμήματος

Η σελίδα **DeleteDepartment** (Σχήμα A.13) διαγράφει από τη βάση δεδομένων του συστήματος το επιλεγμένο τμήμα με τον κωδικό και το όνομα που προβάλλονται στα αντικείμενα **No** και **Name** (τύπου `outputLabel`). Η διαγραφή εκτελείται όταν ο χρήστης επιλέξει το αντικείμενο **departmentDelete** (τύπου `commandButton`).



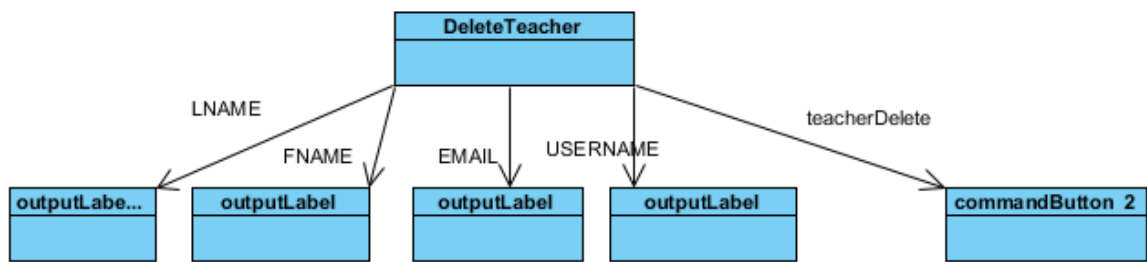
Σχήμα A.14: Διάγραμμα κλάσεων σελίδας εκπαιδευτικών

Η σελίδα **Teachers** (Σχήμα A.14) προβάλλει τους εκπαιδευτικούς του ιδρύματος. Το αντικείμενο **prepareInsertTeacher** (τύπου `commandButton`) όταν επιλεγεί ανοίγει τη σελίδα **InsertTeacher**. Το αντικείμενο **teacherList** (τύπου `dataTable`) προβάλλει τους εκπαιδευτικούς και δίπλα σε κάθε εκπαιδευτικό υπάρχουν τα αντικείμενα **prepareUpdateTeacher** και **prepareDeleteTeacher** που όταν επιλεγούν ανοίγουν οι σελίδες **UpdateTeacher** και **DeleteTeacher** αντίστοιχα. Στο αντικείμενο **departmentMenu** (τύπου `selectOneMenu`) γίνεται επιλογή του τμήματος ώστε να εμφανιστούν οι εκπαιδευτικοί συγκεκριμένου τμήματος.



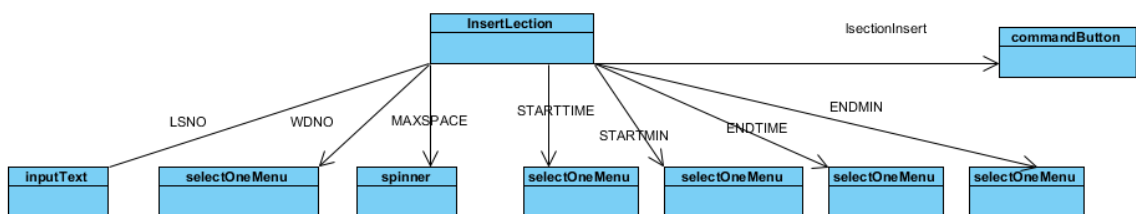
Σχήμα A.15: Διάγραμμα κλάσεων σελίδας εισαγωγής εκπαιδευτικού

Η σελίδα **InsertTeacher** (Σχήμα A.15) εισάγει στη βάση δεδομένων του συστήματος νέο εκπαιδευτικό με τα στοιχεία: επώνυμο (`LNAME`), όνομα (`FNAME`), διεύθυνση ηλεκτρονικής αλληλογραφίας (`EMAIL`), όνομα λογαριασμού (`USERNAME`) και κωδικό (`PASSWORD`) που έχει πληκτρολογήσει ο χρήστης. Η εισαγωγή εκτελείται όταν ο χρήστης επιλέξει το αντικείμενο **teacherInsert** (τύπου `commandButton`).



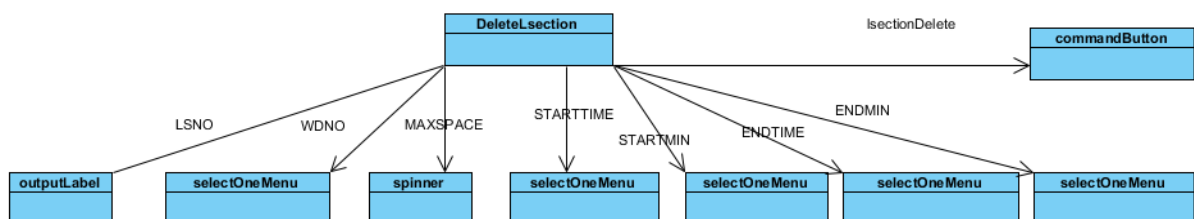
Σχήμα A.16: Διάγραμμα κλάσεων σελίδας διαγραφής εκπαιδευτικού

Η σελίδα **DeleteTeacher** (Σχήμα A.16) διαγράφει από τη βάση δεδομένων του συστήματος τον επιλεγμένο εκπαιδευτικό με τα προβαλλόμενα στοιχεία: επώνυμο (LNAME), όνομα (FNAME), διεύθυνση ηλεκτρονικής αλληλογραφίας (EMAIL), όνομα λογαριασμού (USERNAME) (τύπου outputLabel). Η διαγραφή εκτελείται όταν ο χρήστης επιλέξει το αντικείμενο **teacherDelete** (τύπου commandButton).



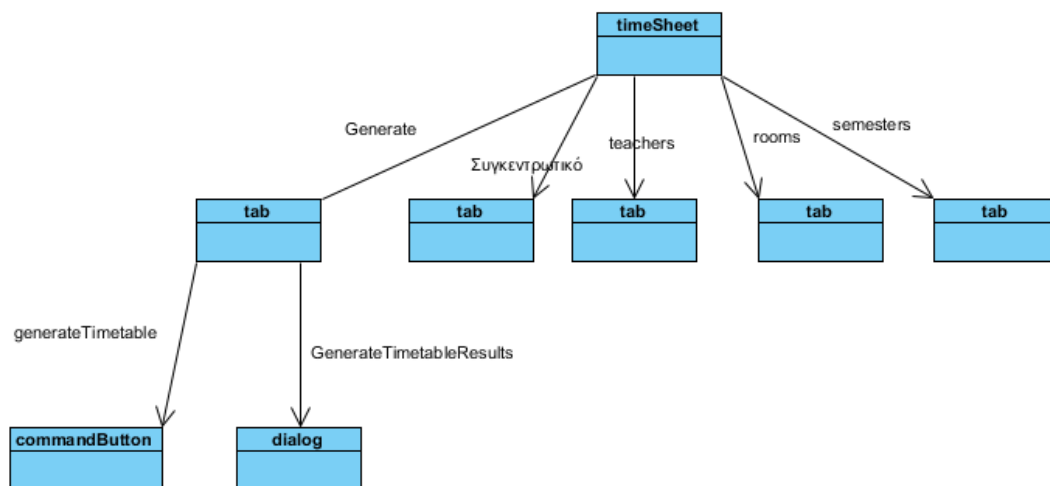
Σχήμα A.17: Διάγραμμα κλάσεων σελίδας εισαγωγής ομάδας παρακολούθησης

Η σελίδα **InsertLSection** (Σχήμα A.17) εισάγει στη βάση δεδομένων του συστήματος νέα ομάδα παρακολούθησης σε συγκεκριμένο μάθημα με κωδικό ομάδας (αντικείμενο LSNO), ημέρα της εβδομάδας (αντικείμενο WDNO), χωρητικότητα (αντικείμενο **MAXSPACE** τύπου spinner), ώρα και λεπτό έναρξης (STARTTIME, STARTMIN), ώρα και λεπτό λήξης (ENDTIME, ENDMIN). Η εισαγωγή εκτελείται όταν ο χρήστης επιλέξει το αντικείμενο **lsectionInsert** (τύπου commandButton).



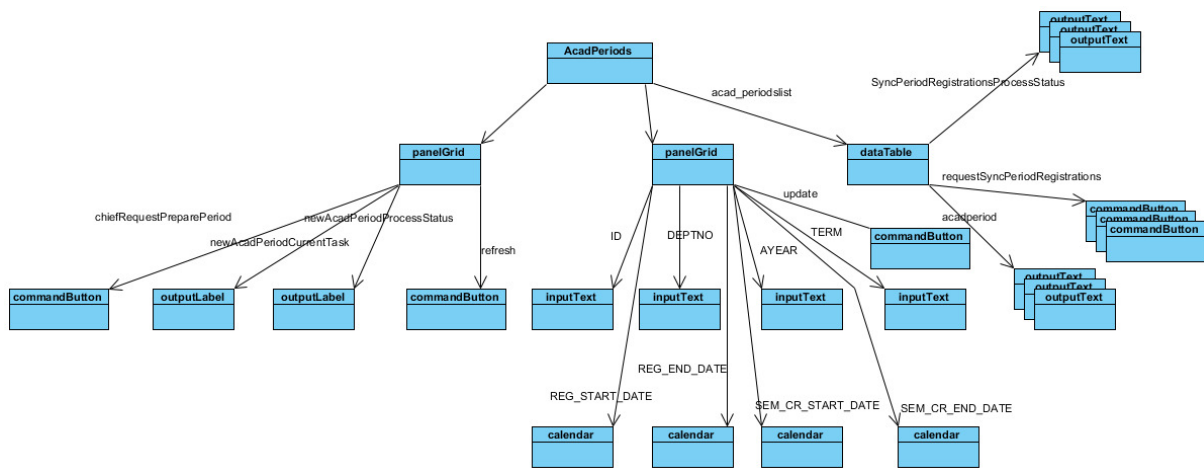
Σχήμα A.18: Διάγραμμα κλάσεων σελίδας διαγραφής ομάδας παρακολούθησης

Η σελίδα **DeleteLSection** (Σχήμα A.18) διαγράφει από τη βάση δεδομένων του συστήματος την επιλεγμένη ομάδα παρακολούθησης που προβάλλεται με τα στοιχεία: κωδικός ομάδας (αντικείμενο LSNO), ημέρα της εβδομάδας (αντικείμενο WDNO), χωρητικότητα (αντικείμενο **MAXSPACE** τύπου spinner), ώρα και λεπτό έναρξης(STARTTIME, STARTMIN), ώρα και λεπτό λήξης(ENDTIME, ENDMIN). Η διαγραφή εκτελείται όταν ο χρήστης επιλέξει το αντικείμενο **lsectionDelete** (τύπου `commandButton`).



Σχήμα A.19: Διάγραμμα κλάσεων σελίδας δημιουργίας και προβολής ωρολογίων προγράμματος

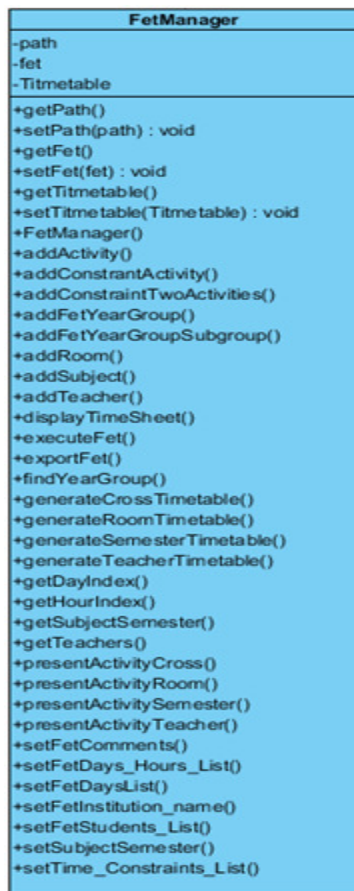
Η σελίδα **timeSheet** (Σχήμα A.19) περιέχει το ωρολόγιο πρόγραμμα ενός τμήματος. Το περιεχόμενο προβάλλεται σε ένα αντικείμενο τύπου `tabView`. Το ωρολόγιο πρόγραμμα έχει τέσσερις προβολές (αντικείμενα τύπου `tab`) : συγκεντρωτική, προβολή ανά εκπαιδευτικό, ανά αίθουσα και ανά εξάμηνο διδασκαλίας. Στο `tab Generate` μπορεί να δημιουργηθεί το ωρολόγιο πρόγραμμα επιλέγοντας το αντικείμενο **generateTimetable**, ενώ τα αποτελέσματα της προσπάθειας δημιουργίας εμφανίζονται στο αντικείμενο **GenerateTimetableResults** (τύπου `dialog - popup`).



Σχήμα A.20: Διάγραμμα κλάσεων σελίδας ακαδημαϊκών περιόδων

Η σελίδα **AcadPeriods** (Σχήμα A.20) προβάλλει ακαδημαϊκές περιόδους και εκκινεί τις BPM διεργασίες προετοιμασίας νέας ακαδημαϊκής περιόδου και συγχρονισμού δηλώσεων ακαδημαϊκών περιόδων. Το αντικείμενο **chiefRequestPreparePeriod** (τύπου `commandButton`) προκαλεί τη δημιουργία της `NewAcadPeriod` διεργασίας, ενώ τα αντικείμενα **newAcadPeriodCurrentTask** και **newAcadPeriodProcessStatus** προβάλλουν την τρέχουσα εργασία (task) και την κατάσταση της διεργασίας αντίστοιχα (τύπου `outputLabel`). Το αντικείμενο **refresh** (τύπου `commandButton`) προκαλεί ανανέωση των ενδείξεων της διεργασίας. Τα αντικείμενα **ID** (κωδικός ακαδημαϊκής περιόδου), **DEPTNO** (κωδικός τμήματος), **AYEAR** (ακαδημαϊκό έτος), **TERM** (ακαδημαϊκή περίοδος), **REG_START_DATE** (ημερομηνία έναρξης δηλώσεων), **REG_END_DATE** (ημερομηνία λήξης δηλώσεων), **SEM_CR_START_DATE** (ημερομηνία έναρξης προτεραιότητας εξαμήνου κατά τη δήλωση), **SEM_CR_END_DATE** (ημερομηνία λήξης προτεραιότητας εξαμήνου κατά τη δήλωση) αποτελούν τα στοιχεία της τελευταίας ενεργής ακαδημαϊκής περιόδου που ενημερώνονται με το αντικείμενο **update**. Στο αντικείμενο **acad_periodslist** (τύπου `dataTable`) εμφανίζεται λίστα με ιστορικό ακαδημαϊκών περιόδων. Σε κάθε γραμμή της λίστας υπάρχει η ακαδημαϊκή περίοδος (αντικείμενο **acadperiod**), κουμπί για τη δυνατότητα αιτήματος (δημιουργία της BPM διεργασίας `SyncPeriodRegistrations`) συγχρονισμού δηλώσεων της περιόδου (αντικείμενο **requestSyncPeriodRegistrations**) και προβολή της κατάστασης (αντικείμενο **SyncPeriodRegistrationsProcessStatus**) της διεργασίας.

A.3 Περιγραφή κλάσεων υποσυστήματος ωρολογίου προγράμματος



Η κλάση **FetManager** ασχολείται με τη διαχείριση του ωρολογίου προγράμματος. Το χαρακτηριστικό **path** περιέχει τη διαδρομή εκτέλεσης της εξωτερικής εφαρμογής FET (fet-cl.exe). Για την επιτυχή εκτέλεση του προγράμματος FET απαιτείται η τροφοδοσία του με ένα αρχείο εισόδου μορφής XML. Το αρχείο εισόδου περιλαμβάνει τους περιορισμούς που έχουμε θέσει ως προϋπόθεση για να παραχθεί το ωρολόγιο πρόγραμμα. Η μέθοδος **exportFet** παράγει το αρχείο εισόδου XML, ενώ η μέθοδος **executeFet** παράγει το ωρολόγιο πρόγραμμα σε ένα αρχείο εξόδου XML εκτελώντας το αρχείο fet-cl.exe του εξωτερικού λογισμικού FET. Οι περιορισμοί τοποθετούνται με χρήση των μεθόδων **setFetInstitution_Name**, **setFetComments**, **setFetDaysList**, **setFetDaysHours_List**, **setFetStudents_List**, **addFetYearGroup**, **addFetYearGroupSubgroup**, **addTeacher**, **addSubject**, **setSubjectSemester**, **addRoom**, **addActivity**, **setTime_Constraints_List**, **addConstraintActivityPreferredRooms** και **addConstraintTwoActivitiesConsecutive**. Η μέθοδος **generateCrossTimetable** χρησιμοποιείται για την συγκεντρωτική προβολή ανά μάθημα, ομάδα μαθήματος, καθηγητή και αίθουσα διδασκαλίας. Η μέθοδος **generateTeacherTimetable** χρησιμοποιείται για την προβολή

ανά καθηγητή, μάθημα και ομάδα μαθήματος. Η μέθοδος **generateRoomTimetable** χρησιμοποιείται για την προβολή ανά αίθουσα, μάθημα, ομάδα μαθήματος. Η μέθοδος **generateSemesterTimetable** χρησιμοποιείται για την προβολή ανά εξάμηνο.

Day
-Name
+Day() +getName() +setName(Name) : void

Η κλάση **Day** εκφράζει τον περιορισμό μιας ημέρας (χαρακτηριστικό **Name**) της εβδομάδας.

Days_list
-Number_of_Days
+Days_list() +getNumber_of_Days() +setNumber_of_Days(Number_of_Days) : void

Η κλάση **Days_List** περιέχει λίστα με τις ημέρες (Day) της εβδομάδας που ορίζονται ως περιορισμοί για το ωρολόγιο πρόγραμμα.

Hour
-Name
+Hour() +getName() +setName(Name) : void

Η κλάση **Hour** εκφράζει μια συγκεκριμένη ώρα (χαρακτηριστικό **Name**) που θα αποτελεί περιορισμό για όλες τις ημέρες που έχουν οριστεί ως περιορισμοί.

Hours_list
-Number_of_Hours
+Hours_list() +getNumber_of_Hours() +setNumber_of_Hours(Number_of_Hours) : void

Η κλάση **Hours_List** περιέχει λίστα με τις ώρες (Hour) των ημερών που έχουν οριστεί ως περιορισμοί για το ωρολόγιο πρόγραμμα.

Subgroup
-Name -Number_of_students
+Subgroup() +getName() +setName(Name) : void +getNumber_of_students() +setNumber_of_students(Number_of_students) : void

Η κλάση **Subgroup** εκφράζει την ομάδα παρακολούθησης. Το χαρακτηριστικό Name είναι ο κωδικός της ομάδας. Το χαρακτηριστικό Number_Of_Students είναι η χωρητικότητα της ομάδας.

Group
-Name -Number_of_Students
+Group() +getName() +setName(Name) : void

Η κλάση **Group** εκφράζει ένα μάθημα που περιλαμβάνει τις ομάδες παρακολούθησης (Subgroup). Το χαρακτηριστικό Name είναι ο κωδικός του μαθήματος. Το χαρακτηριστικό Number_Of_Students είναι η χωρητικότητα του μαθήματος.

Year
-Name -Number_of_Students
+Year() +getName() +setName(Name) : void +getNumber_of_Students() +setNumber_of_Students(Number_of_Students) : void

Η κλάση **Year** εκφράζει την ακαδημαϊκή περίοδο (χαρακτηριστικό **Name**) ενός τμήματος που περιλαμβάνει τα μαθήματα (Group). Το χαρακτηριστικό Number_Of_Students είναι η χωρητικότητα της τάξης/περιόδου.

Students_List
+Students_List()

Η κλάση **Students_List** περιέχει μια ακαδημαϊκή περίοδο (Year).

Teacher
-Name
+Teacher() +getName() +setName(Name) : void

Η κλάση **Teacher** εκφράζει έναν εκπαιδευτικό. Το χαρακτηριστικό **Name** είναι το ονοματεπώνυμο του εκπαιδευτικού.

Teachers_list
+Teachers_list()

Η κλάση **Teacher_List** περιέχει το σύνολο των εκπαιδευτικών (Teacher).

Subject
-Name
+Subject()
+getName()
+setName(Name) : void

Η κλάση **Subject** εκφράζει ένα μάθημα. Το χαρακτηριστικό Name είναι ο κωδικός του μαθήματος.

Subjects_List
+Subjects_List()

Η κλάση **Subjects_List** περιέχει το σύνολο των μαθημάτων (Subject).

Room
-Name
-Capacity
+Room()
+getName()
+setName(Name) : void
+getCapacity()
+setCapacity(Capacity) : void

Η κλάση **Room** περιέχει τα στοιχεία μιας αίθουσας: όνομα (**Name**) αίθουσας και χωρητικότητα (**Capacity**).

Rooms_List
+Rooms_List()

Η κλάση **Rooms_List** περιέχει το σύνολο των αιθουσών (Room).

ConstraintActivityPreferredRooms
-Weight_Percentage -Activity_Id -Number_of_PREFERRED_Rooms -Preferred_Room
+ConstraintActivityPreferredRooms() +getWeight_Percentage() +setWeight_Percentage(Weight_Percentage) : void +getActivity_Id() +setActivity_Id(Activity_Id) : void +getNumber_of_PREFERRED_Rooms() +setNumber_of_PREFERRED_Rooms(Number_of_PREFERRED_Rooms) : void +getPreferred_Room() +setPreferred_Room(Preferred_Room) : void

Η κλάση **ConstraintActivityPreferredRooms** περιέχει τις προτεινόμενες αίθουσες ανά ομάδα παρακολούθησης. Το χαρακτηριστικό **Activity_Id** εκφράζει την ομάδα παρακολούθησης στην οποία ανήκει αυτός ο περιορισμός. Το χαρακτηριστικό **Weight_Percentage** περιέχει τη βαρύτητα κατά την επιλογή αυτής της αίθουσας. Το χαρακτηριστικό **Number_of_PREFERRED_Rooms** είναι το πλήθος των προτεινόμενων αιθουσών. Το χαρακτηριστικό **Preferred_Room** είναι το όνομα της προτεινόμενης αίθουσας.

Space_Constraints_List
+Space_Constraints_List()

Η κλάση **Space_Constraints_List** περιέχει τους προτεινόμενους περιορισμούς αιθουσών (**ConstraintActivityPreferredRooms**) ανά ομάδα παρακολούθησης.

Activity
-teacher -subject -Students -Duration -Id
+Activity() +getTeacher() +setTeacher(teacher) : void +getSubject() +setSubject(subject) : void +getStudents() +setStudents(Students) : void +getDuration() +setDuration(Duration) : void +getId() +setId(Id) : void

Η κλάση **Activity** διασυνδέει εκπαιδευτικό (χαρακτηριστικό **teacher**), μάθημα (**subject**), ομάδα παρακολούθησης (**Students**) και ορίζει τη διάρκεια της ομάδας παρακολούθησης (**Duration**).

Κάθε αντικείμενο Activity έχει επίσης και το χαρακτηριστικό **Id** που το χαρακτηρίζει για να μπορέσει να πάρει μέρος σε άλλους περιορισμούς.

Activities_List
+Activities_List()

Η κλάση **Activities_List** περιέχει ένα σύνολο δραστηριοτήτων (Activity).

ConstraintTwoActivitiesConsecutive
-Weight_Percentage
-First_Activity_Id
-Second_Activity_Id
+ConstraintTwoActivitiesConsecutive()
+getWeight_Percentage()
+setWeight_Percentage(Weight_Percentage) : void
+getFirst_Activity_Id()
+setFirst_Activity_Id(First_Activity_Id) : void
+getSecond_Activity_Id()
+setSecond_Activity_Id(Second_Activity_Id) : void

Η κλάση **ConstraintTwoActivitiesConsecutive** ορίζει περιορισμό πως δυο συγκεκριμένες δραστηριότητες (Activity) θα είναι διαδοχικές μεταξύ τους. Πρώτη θα βρίσκεται η δραστηριότητα με χαρακτηριστικό το **First_Activity_Id** και θα την ακολουθεί η δραστηριότητα με χαρακτηριστικό το **Second_Activity_Id**.

ConstraintActivityPreferredStartingTime
-Weight_Percentage
-Activity_Id
-Preferred_Day
-Preferred_Hour

Η κλάση **ConstraintActivityPreferredStartingTime** ορίζει πως μια συγκεκριμένη δραστηριότητα (χαρακτηριστικό **Activity_Id**) θα πραγματοποιηθεί συγκεκριμένη ημέρα (χαρακτηριστικό **Preferred_Day**) και ώρα (χαρακτηριστικό **Preferred_Hour**).

Time_Constraints_List
+Time_Constraints_List()

Η κλάση **Time_Constraints_List** περιέχει είτε σύνολο περιορισμών **ConstraintTwoActivitiesConsecutive** ή σύνολο περιορισμών **ConstraintActivityPreferredStartingTime**.

Timetable
-rowList : List<String[]>
+Timetable() +getRowList() : List<String[]> +setRowList(rowList : List<String[]>) : void +addRow()

Η κλάση **Timetable** χρησιμοποιείται για την προβολή του ωρολογίου προγράμματος. Τα κελιά του ωρολογίου προγράμματος αποθηκεύονται στο χαρακτηριστικό **rowList**. Η μέθοδος `addRow` προσθέτει μια γραμμή στο ωρολόγιο πρόγραμμα.

fet
-version -Institution_Name -Comments
+fet() +getVersion() +setVersion(version) : void +getInstitution_Name() +setInstitution_Name(Institution_Name) : void +getComments() +setComments(Comments) : void

Η κλάση **fet** έχει κεντρική θέση στο υποσύστημα, καθώς διασυνδέεται με όλες τις επιμέρους κλάσεις που αποτελούν τη δομή και τους περιορισμούς του ωρολογίου προγράμματος. Η κλάση `FetManager` -η οποία επεξεργάζεται ένα ωρολόγιο πρόγραμμα και χρησιμοποιείται ως `interface` για τη διασύνδεση/χρήση του εξωτερικού συστήματος FET από τα υποσυστήματα (που χρησιμοποιούν ωρολόγιο πρόγραμμα)- αποκτά πρόσβαση και ενημερώνει τα χαρακτηριστικά όλων των υπόλοιπων κλάσεων του ωρολογίου προγράμματος χρησιμοποιώντας την κλάση `fet`. Το χαρακτηριστικό **Institution_Name** έχει το όνομα ενός τμήματος από το εκπαιδευτικό ίδρυμα.

A.4 Περιγραφή κλάσεων υποσυστήματος συγχρονισμού

JMainFrame
+JMainFrame() +exportAcadPeriodsToXML() +exportTeachersToXML() +exportStudentsToXML() +exportLessonsToXML() +exportRegSLToXML() +syncAcadPeriodsAction() +insertAcadPeriod() +syncTeachersAction() +insertTeacher() +syncStudentsAction() +insertStudent() +syncLessonsAction() +insertLesson() +syncRegistrationsAction() +insertRegistration() +doOracleConnect() +doOracleDisconnect() +AcadPeriodsAction() +TeachersAction() +StudentsAction() +LessonsAction() +RegSLsAction() +newUserProcessAction() +checkProcessInstancesAction() +checkNewAcadPeriodRequestsAction() +checkSyncRegistrationsRequestsAction() +checkLessonGroupsFullRequestsAction() +checkNewUserRequestsAction() +newAcadPeriodCompleteTask() +completeSyncPeriodRegistrationsAsyncTask()

Η κλάση **JMainFrame** είναι η κεντρική φόρμα του υποσυστήματος συγχρονισμού. Περιέχει μεθόδους για την εξαγωγή ακαδημαϊκών περιόδων, εκπαιδευτικών, φοιτητών, μαθημάτων, δηλώσεων μαθημάτων σε XML αρχεία και αντίστοιχες μεθόδους για την εισαγωγή τους στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών. Για την εύρεση των παραπάνω στοιχείων επικοινωνεί με επιμέρους φόρμες. Περιέχει μεθόδους για την ανίχνευση ενεργών BPM διεργασιών, την προβολή των καταστάσεών τους σε αντικείμενα της φόρμας και την ολοκλήρωση των σχετικών BPM εργασιών (tasks).

Η μέθοδος **exportAcadPeriodsToXML** εξάγει τις ακαδημαϊκές περιόδους από το οπτικό αντικείμενο `jAcadPeriodsTable` σε αρχείο XML.

Η μέθοδος **exportTeachersToXML** εξάγει τους εκπαιδευτικούς από το οπτικό αντικείμενο `jTeachersTable` σε αρχείο XML.

Η μέθοδος **exportStudentsToXML** εξάγει τους φοιτητές από το οπτικό αντικείμενο `jStudentsTable` σε αρχείο XML.

Η μέθοδος **exportLessonsToXML** εξάγει τα μαθήματα από το οπτικό αντικείμενο `jLessonsTable` σε αρχείο XML.

Η μέθοδος **exportRegSLToXML** εξάγει τις δηλώσεις από το οπτικό αντικείμενο `jRegistrationsTable` σε αρχείο XML.

Η μέθοδος **syncAcadPeriodsAction** εισάγει τις ακαδημαϊκές περιόδους ενός τμήματος από το αρχείο XML στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών. Το αρχείο XML έχει αρκετές ακαδημαϊκές περιόδους και κάθε ακαδημαϊκή περίοδος γίνεται εισαγωγή στη βάση με τη μέθοδο `insertAcadPeriod`.

Η μέθοδος **insertAcadPeriod** εισάγει στη βάση του συστήματος μια νέα ακαδημαϊκή περίοδο για συγκεκριμένο τμήμα.

Η μέθοδος **syncTeachersAction** εισάγει τους εκπαιδευτικούς ενός τμήματος από το αρχείο XML στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών. Το αρχείο XML έχει αρκετούς εκπαιδευτικούς και κάθε εκπαιδευτικός γίνεται εισαγωγή στη βάση με τη μέθοδο `insertTeacher`.

Η μέθοδος **insertTeacher** εισάγει στη βάση του συστήματος έναν εκπαιδευτικό για συγκεκριμένο τμήμα.

Η μέθοδος **syncStudentsAction** εισάγει τους φοιτητές ενός τμήματος από το αρχείο XML στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών. Το αρχείο XML έχει αρκετούς φοιτητές και κάθε φοιτητής γίνεται εισαγωγή στη βάση με τη μέθοδο `insertStudent`.

Η μέθοδος **insertStudent** εισάγει στη βάση του συστήματος έναν φοιτητή για συγκεκριμένο τμήμα.

Η μέθοδος **syncLessonsAction** εισάγει τα μαθήματα ενός τμήματος από το αρχείο XML στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών. Το αρχείο XML έχει αρκετά μαθήματα και κάθε μάθημα γίνεται εισαγωγή στη βάση με τη μέθοδο `insertLesson`.

Η μέθοδος **insertLesson** εισάγει στη βάση του συστήματος ένα μάθημα για συγκεκριμένο τμήμα.

Η μέθοδος **syncRegistrationsAction** εισάγει τις δηλώσεις φοιτητών ενός τμήματος από το αρχείο XML στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών. Το αρχείο XML έχει αρκετές δηλώσεις φοιτητών και κάθε δήλωση γίνεται εισαγωγή στη βάση με τη μέθοδο `insertRegistration`.

Η μέθοδος **insertRegistration** εισάγει στη βάση του συστήματος μια νέα ακαδημαϊκή περίοδο για συγκεκριμένο τμήμα.

Η μέθοδος **doOracleConnect** συνδέει το υποσύστημα συγχρονισμού με τη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών.

Η μέθοδος **doOracleDisconnect** αποσυνδέει το υποσύστημα συγχρονισμού από τη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών.

Η μέθοδος **AcadPeriodsAction** εμφανίζει τη φόρμα `JAcadPeriodJDialog`.

Η μέθοδος **TeachersAction** εμφανίζει τη φόρμα `JTeachersJDialog`.

Η μέθοδος **StudentsAction** εμφανίζει τη φόρμα `JStudentsJDialog`.

Η μέθοδος **LessonsAction** εμφανίζει τη φόρμα `JLessonsJDialog`.

Η μέθοδος **RegSLsAction** εμφανίζει τη φόρμα `JRegSLJDialog`.

Η μέθοδος **newUserProcessAction** εμφανίζει τη φόρμα `NewUserProcessDialog`.

Η μέθοδος **checkProcessInstancesAction** ελέγχει αν υπάρχουν ενεργές BPM διεργασίες και εμφανίζει την τρέχουσα εργασία (task) και την κατάσταση αυτών των διεργασιών στη φόρμα `JMainFrame`.

Η μέθοδος **checkNewAcadPeriodRequestsAction** ελέγχει αν υπάρχει ενεργή η BPM διεργασία `NewAcadPeriod` και εμφανίζει την τρέχουσα εργασία (task) και την κατάσταση αυτής της διεργασίας στη φόρμα `JMainFrame`.

Η μέθοδος **checkSyncRegistrationsRequestsAction** ελέγχει αν υπάρχει ενεργή η BPM διεργασία **SyncRegistrations** και εμφανίζει την τρέχουσα εργασία (task) και την κατάσταση αυτής της διεργασίας στη φόρμα JMainFrame.

Η μέθοδος **checkLessonGroupsFullRequestsAction** ελέγχει αν υπάρχει ενεργή η BPM διεργασίας **LessonGroupsFull** και εμφανίζει την τρέχουσα εργασία (task) και την κατάσταση αυτής της διεργασίας στη φόρμα JMainFrame.

Η μέθοδος **checkNewUserRequestsAction** ελέγχει αν υπάρχει ενεργή η BPM διεργασία **NewUserRequest** και εμφανίζει την τρέχουσα εργασία (task) και την κατάσταση αυτής της διεργασίας στη φόρμα JMainFrame.

Η μέθοδος **newAcadPeriodCompleteTask** ολοκληρώνει την τρέχουσα εργασία της διεργασίας NewAcadPeriod καλώντας τη rest μέθοδο CompleteCurrentAsyncTask του υποσυστήματος υποστήριξης διεργασιών.

Η μέθοδος **completeSyncPeriodRegistrationsAsyncTask** ολοκληρώνει την τρέχουσα εργασία της διεργασίας SyncPeriodRegistrations καλώντας τη rest μέθοδο CompleteSyncPeriodRegistrationsAsyncTask του υποσυστήματος υποστήριξης διεργασιών.

Connection

Η κλάση **Connection** χρησιμοποιείται για τη διασύνδεση του υποσυστήματος με τη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών.

ACAD_PERIOD
-ACAD_PERIOD_ID -DEPTNO -AYEAR -TERM
+ACAD_PERIOD() +getACAD_PERIOD_ID() +setACAD_PERIOD_ID(ACAD_PERIOD_ID) : void +getDEPTNO() +setDEPTNO(DEPTNO) : void +getAYEAR() +setAYEAR(AYEAR) : void +getTERM() +setTERM(TERM) : void

Η κλάση **ACAD_PERIOD** περιέχει τα χαρακτηριστικά μιας ακαδημαϊκής περιόδου: κωδικός (**ACAD_PERIOD_ID**), κωδικός τμήματος (**DEPTNO**), ακαδημαϊκό έτος (**AYEAR**) και ακαδημαϊκή περίοδος (**TERM**).

DepartmentAcadPeriods
-departmentName
+DepartmentAcadPeriods() +getDepartmentName() +setDepartmentName(departmentName) : void +addAcadPeriod()

Η κλάση **DepartmentAcadPeriods** περιέχει τις ακαδημαϊκές περιόδους ενός τμήματος. Το χαρακτηριστικό **departmentName** είναι το όνομα του τμήματος. Η μέθοδος **addAcadPeriod** προσθέτει μια ακαδημαϊκή περίοδο στο τμήμα.

Lesson
-CODE -TITLE -SEMESTERNO -SECTIONCOUNT -FINALSTATUS -DEPTNO
+Lesson() +getCODE() +setCODE(CODE) : void +getTitle() +setTitle(TITLE) : void +getSEMESTERNO() +setSEMESTERNO(SEMESTERNO) : void +getSECTIONCOUNT() +setSECTIONCOUNT(SECTIONCOUNT) : void +getFINALSTATUS() +setFINALSTATUS(FINALSTATUS) : void +getDEPTNO() +setDEPTNO(DEPTNO) : void

Η κλάση **Lesson** περιέχει τα χαρακτηριστικά ενός μαθήματος: κωδικός (**CODE**), τίτλος(**TITLE**), εξάμηνο διδασκαλίας (**SEMESTERNO**), πλήθος ομάδων μαθήματος (**SECTIONCOUNT**), κωδικός τμήματος (**DEPTNO**). Αν κάποιο μάθημα είναι μικτό (θεωρητικό κ εργαστηριακό) τότε δημιουργούνται δυο αντικείμενα της ίδια κλάσης χωρίς να υπάρχει σύνδεση μεταξύ τους.

DepartmentLessons
-departmentName
+DepartmentLessons() +getDepartmentName() +setDepartmentName(departmentName) : void +addLesson()

Η κλάση **DepartmentLessons** περιέχει τα μαθήματα ενός τμήματος. Το χαρακτηριστικό **departmentName** είναι το όνομα του τμήματος. Η μέθοδος **addLesson** προσθέτει ένα νέο μάθημα.

Student
-AEM -LNAME -FNAME -EMAIL -SEMESTER -UNAME -PASSWD -DEPTNO
+Student() +getAEM() +setAEM(AEM) : void +getLNAME() +setLNAME(LNAME) : void +getFNAME() +setFNAME(FNAME) : void +getEMAIL() +setEMAIL(EMAIL) : void +getSEMESTER() +setSEMESTER(SEMESTER) : void +getUNAME() +setUNAME(UNAME) : void +getPASSWD() +setPASSWD(PASSWD) : void +getDEPTNO() +setDEPTNO(DEPTNO) : void

Η κλάση **Student** περιέχει τα χαρακτηριστικά του φοιτητή: αριθμός μητρώου (**AEM**), επώνυμο(**LNAME**), όνομα (**FNAME**), διεύθυνση ηλεκτρονικής αλληλογραφίας (**EMAIL**), τρέχον εξάμηνο φοίτησης (**SEMESTER**), όνομα λογαριασμού (**UNAME**), κωδικός λογαριασμού (**PASSWD**), κωδικός τμήματος στο οποίο ανήκει ο φοιτητής (**DEPTNO**).

DepartmentStudents
-departmentName
+DepartmentStudents() +getDepartmentName() +setDepartmentName(departmentName) : void +addStudent()

Η κλάση **DepartmentStudents** περιέχει τους φοιτητές ενός τμήματος. Το χαρακτηριστικό **departmentName** είναι το όνομα του τμήματος. Η μέθοδος **addStudent** προσθέτει ένα νέο φοιτητή.

Teacher
-LNAME -FNAME -EMAIL -UNAME -PASSWORD -DEPTNO -ISADMIN
+Teacher() +getLNAME() +setLNAME(LNAME) : void +getFNAME() +setFNAME(FNAME) : void +getEmail() +setEmail(EMAIL) : void +getUNAME() +setUNAME(UNAME) : void +getPASSWORD() +setPASSWORD(PASSWORD) : void +getDEPTNO() +setDEPTNO(DEPTNO) : void +getISADMIN() +setISADMIN(ISADMIN) : void

Η κλάση **Teacher** περιέχει τα χαρακτηριστικά ενός εκπαιδευτικού: επώνυμο(**LNAME**), όνομα (**FNAME**), διεύθυνση ηλεκτρονικής αλληλογραφίας (**EMAIL**), όνομα λογαριασμού (**UNAME**), κωδικός λογαριασμού (**PASSWORD**), κωδικός τμήματος στο οποίο ανήκει ο εκπαιδευτικός (**DEPTNO**). Το χαρακτηριστικό **ISADMIN** έχει τιμή 1 αν ο χρήστης είναι διαχειριστής του συστήματος.

DepartmentTeachers
-departmentName
+DepartmentTeachers() +getDepartmentName() +setDepartmentName(departmentName) : void +addTeacher()

Η κλάση **DepartmentTeachers** περιέχει τους εκπαιδευτικούς ενός τμήματος. Το χαρακτηριστικό **departmentName** είναι το όνομα του τμήματος. Η μέθοδος **addTeacher** προσθέτει ένα νέο εκπαιδευτικό.

REG_SL
-LESSONCODE -ACAD_PERIOD_ID -DEPTNO -STUDENTAEM -MARK
+REG_SL() +getLESSONCODE() +setLESSONCODE(LESSONCODE) : void +getACAD_PERIOD_ID() +setACAD_PERIOD_ID(ACAD_PERIOD_ID) : void +getDEPTNO() +setDEPTNO(DEPTNO) : void +getSTUDENTAEM() +setSTUDENTAEM(STUDENTAEM) : void +getMARK() +setMARK(MARK) : void

Η κλάση **REG_SL** περιέχει τη δήλωση και το βαθμό (αν έχει) ενός φοιτητή για ένα μάθημα σε μια συγκεκριμένη ακαδημαϊκή περίοδο. Τα χαρακτηριστικά της κλάσης είναι: κωδικός ακαδημαϊκής περιόδου (**ACAD_PERIOD_ID**), κωδικός μαθήματος (**LESSONCODE**), κωδικός τμήματος (**DEPTNO**), μητρώο φοιτητή (**STUDENTAEM**) και βαθμός (**MARK**).

DepartmentRegSLs
-departmentName
+DepartmentRegSLs() +getDepartmentName() +setDepartmentName(departmentName) : void +addRegSL()

Η κλάση **DepartmentRegSLs** περιέχει τις δηλώσεις των φοιτητών ενός τμήματος. Το χαρακτηριστικό **departmentName** είναι το όνομα του τμήματος. Η μέθοδος **addRegSL** προσθέτει μια νέα δήλωση.

JAcadPeriodJDialog
-deptcode -jAcadPeriodsTable : JTable
+JAcadPeriodJDialog() +getDeptcode() +setDeptcode(deptcode) : void +getJAcadPeriodsTable() : JTable +setJAcadPeriodsTable(jAcadPeriodsTable : JTable) : void +loadAcadPeriodsAction() +exportBtnAction() +SyncBtnAction()

Η φόρμα **JAcadPeriodJDialog** χρησιμοποιείται για την εξαγωγή των στοιχείων ακαδημαϊκών περιόδων ενός συγκεκριμένου τμήματος (χαρακτηριστικό **deptcode**) - από το σύστημα της

ηλεκτρονικής γραμματείας- σε ένα αρχείο XML και στη συνέχεια την εισαγωγή αυτών των περιόδων στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών.

Η μέθοδος **loadAcadPeriodsAction** διαβάζει από τη βάση δεδομένων του γραμματειακού συστήματος τις ακαδημαϊκές περιόδους του επιλεγμένου τμήματος και τα τοποθετεί στο οπτικό αντικείμενο **jAcadPeriodsTable** (τύπου JTable).

Η μέθοδος **exportBtnAction** εξάγει τις ακαδημαϊκές περιόδους από το οπτικό αντικείμενο **jAcadPeriodsTable** σε αρχείο XML.

Η μέθοδος **SyncBtnAction** εισάγει τις ακαδημαϊκές περιόδους από αρχείο XML στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών.

JTeachersJDialog
-deptcode
-jTeachersTable : JTable
+JTeachersJDialog()
+getDeptcode()
+setDeptcode(deptcode) : void
+getJTeachersTable() : JTable
+setJTeachersTable(jTeachersTable : JTable) : void
+loadTeachersAction()
+exportBtnAction()
+SyncBtnAction()
+CompleteTaskBtnAction()

Η φόρμα **jTeachersJDialog** χρησιμοποιείται για την εξαγωγή των εκπαιδευτικών ενός συγκεκριμένου τμήματος (χαρακτηριστικό **deptcode**) -από το σύστημα της ηλεκτρονικής γραμματείας- σε ένα αρχείο XML και στη συνέχεια την εισαγωγή των εκπαιδευτικών αυτού του αρχείου στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών.

Η μέθοδος **loadTeachersAction** διαβάζει από τη βάση δεδομένων του γραμματειακού συστήματος τους εκπαιδευτικούς του επιλεγμένου τμήματος και τα τοποθετεί στο οπτικό αντικείμενο **jTeachersTable** (τύπου JTable).

Η μέθοδος **exportBtnAction** εξάγει τους εκπαιδευτικούς από το οπτικό αντικείμενο **jTeachersTable** σε αρχείο XML.

Η μέθοδος **SyncBtnAction** εισάγει τους εκπαιδευτικούς από το αρχείο XML στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών.

Η μέθοδος **CompleteTaskBtnAction** ολοκληρώνει τη BPM εργασία (task) συγχρονισμού εκπαιδευτικών προς την ολοκλήρωση της ακαδημαϊκής περιόδου που εκκρεμεί (αν η αντίστοιχη διεργασία βρίσκεται σε εκκρεμότητα για αυτή την εργασία).

JStudentsJDialog
-deptcode
-jStudentsTable : JTable
+JStudentsJDialog()
+getDeptcode()
+setDeptcode(deptcode) : void
+getJStudentsTable() : JTable
+setJStudentsTable(jStudentsTable : JTable) : void
+loadStudentsAction()
+exportBtnAction()
+SyncBtnAction()
+CompleteTaskBtnAction()

Η φόρμα **jStudentsJDialog** χρησιμοποιείται για την εξαγωγή των φοιτητών ενός συγκεκριμένου τμήματος (χαρακτηριστικό **deptcode**) -από το σύστημα της ηλεκτρονικής γραμματείας- σε ένα αρχείο XML και στη συνέχεια την εισαγωγή των φοιτητών αυτού του αρχείου στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών.

Η μέθοδος **loadStudentsAction** διαβάζει από τη βάση δεδομένων του γραμματειακού συστήματος τους φοιτητές του επιλεγμένου τμήματος και τα τοποθετεί στο οπτικό αντικείμενο **jStudentsTable** (τύπου JTable).

Η μέθοδος **exportBtnAction** εξάγει τους φοιτητές από το οπτικό αντικείμενο jStudentsTable σε αρχείο XML.

Η μέθοδος **SyncBtnAction** εισάγει τους φοιτητές από το αρχείο XML στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών.

Η μέθοδος **CompleteTaskBtnAction** ολοκληρώνει την BPM εργασία(task) συγχρονισμού φοιτητών προς την ολοκλήρωση της ακαδημαϊκής περιόδου που εκκρεμεί (αν η αντίστοιχη διεργασία βρίσκεται σε εκκρεμότητα για αυτή την εργασία).

JLessonsJDialog
-deptcode -jLessonsTable : JTable
+JLessonsJDialog() +getDeptcode() +setDeptcode(deptcode) : void +getJLessonsTable() : JTable +setJLessonsTable(jLessonsTable : JTable) : void +loadLessonsAction() +exportBtnAction() +SyncBtnAction() +CompleteTaskBtnAction()

Η φόρμα **jLessonsJDialog** χρησιμοποιείται για την εξαγωγή των μαθημάτων ενός συγκεκριμένου τμήματος (χαρακτηριστικό **deptcode**) -από το σύστημα της ηλεκτρονικής γραμματείας- σε ένα αρχείο XML και στη συνέχεια την εισαγωγή των μαθημάτων αυτού του αρχείου στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών.

Η μέθοδος **loadLessonsAction** διαβάζει από τη βάση δεδομένων του γραμματειακού συστήματος τα μαθήματα του επιλεγμένου τμήματος και τα τοποθετεί στο οπτικό αντικείμενο **jLessonsTable** (τύπου JTable).

Η μέθοδος **exportBtnAction** εξάγει τα μαθήματα από το οπτικό αντικείμενο jLessonsTable σε αρχείο XML.

Η μέθοδος **SyncBtnAction** εισάγει τα μαθήματα από το αρχείο XML στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών.

Η μέθοδος **CompleteTaskBtnAction** ολοκληρώνει τη BPM εργασία (task) συγχρονισμού μαθημάτων προς την ολοκλήρωση της ακαδημαϊκής περιόδου που εκκρεμεί (αν η αντίστοιχη διεργασία βρίσκεται σε εκκρεμότητα για αυτή την εργασία).

JRegSLJDialog
-deptcode -ayear -term -jRegistrationsTable : JTable
+JRegSLJDialog() +getDeptcode() +setDeptcode(deptcode) : void +getAyear() +setAyear(ayear) : void +getTerm() +setTerm(term) : void +getJRegistrationsTable() : JTable +setJRegistrationsTable(jRegistrationsTable : JTable) : void +loadRegistrationsAction() +exportBtnAction() +SyncBtnAction() +CompleteTaskBtnAction()

Η φόρμα **JRegSLJDialog** χρησιμοποιείται για την εξαγωγή των δηλώσεων ενός τμήματος (από το σύστημα της ηλεκτρονικής γραμματείας), για μια συγκεκριμένη ακαδημαϊκή περίοδο σε ένα αρχείο XML και στη συνέχεια την εισαγωγή των δηλώσεων αυτού του αρχείου στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών. Η ακαδημαϊκή περίοδος προσδιορίζεται από τα χαρακτηριστικά: κωδικός τμήματος (**deptcode**), έτος ακαδημαϊκής περιόδου (**ayear**) και περίοδος (**term**).

Η μέθοδος **loadRegistrationsAction** διαβάζει από τη βάση δεδομένων του γραμματειακού συστήματος της δηλώσεις συγκεκριμένης ακαδημαϊκής περιόδου ενός τμήματος και τα τοποθετεί στο οπτικό αντικείμενο **jRegistrationsTable** (τύπου JTable).

Η μέθοδος **exportBtnAction** εξάγει τις δηλώσεις από το οπτικό αντικείμενο **jRegistrationsTable** σε αρχείο XML.

Η μέθοδος **SyncBtnAction** εισάγει τις δηλώσεις από αρχείο XML στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών.

Η μέθοδος **CompleteTaskBtnAction** ολοκληρώνει τη BPM εργασία (task) συγχρονισμού δηλώσεων που εκκρεμεί (αν η αντίστοιχη διεργασία βρίσκεται σε εκκρεμότητα για αυτή την εργασία).

NewUserProcessDialog
-usernames : String[]
+NewUserProcessDialog()
+getUsernames() : String []
+setUsernames(usernames : String []) : void
+completeCreateUserTask()
+getRstNewUserRequests()
+createUserAction()
+addLdapStudentUserEntry()
+loadRequestUserDetailsAction()

Η φόρμα **NewUserProcessDialog** χρησιμοποιείται για το συγχρονισμό του λογαριασμού ενός χρήστη-φοιτητή που μόλις δημιουργήθηκε στη βάση δεδομένων του γραμματειακού συστήματος. Μόλις η γραμματεία ενημερώσει για τη δημιουργία του φοιτητή-λογαριασμού δημιουργείται ο αντίστοιχος χρήστης στον LDAP Server του ιδρύματος καθώς και στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών. Ο LDAP Server απαιτεί ένα σύνολο χαρακτηριστικών για την έγκυρη δημιουργία του χρήστη. Τα χαρακτηριστικά αυτά περιγράφονται από την κλάση **LdapStudent**.

Η μέθοδος **completeCreateUserTask** ολοκληρώνει τη BPM εργασία(task) δημιουργίας νέου χρήστη που εκκρεμεί (αν η αντίστοιχη διεργασία βρίσκεται σε εκκρεμότητα για αυτή την εργασία).

Η μέθοδος **getRstNewUserRequests** επιστρέφει τα νέα αιτήματα λογαριασμών-χρηστών που εκκρεμούν προς δημιουργία. Για να βρεθούν τα αιτήματα γίνεται rest κλήση της μεθόδου GetNewUserRequests από το υποσύστημα παροχής διεργασιών. Τα αιτήματα τοποθετούνται στο χαρακτηριστικό **usernames**.

Η μέθοδος **createUserAction** διαβάζει τα στοιχεία από τα πεδία της φόρμας και καλεί τη μέθοδο addLdapStudentUserEntry για να δημιουργηθεί ο χρήστης στον Ldap Server. Επίσης προσθέτει το χρήστη στο σύστημα υποστήριξης προγραμμάτων σπουδών.

Η μέθοδος **addLdapStudentUserEntry** προσθέτει έναν χρήστη στον Ldap Server.

Η μέθοδος **loadRequestUserDetailsAction** διαβάζει τα στοιχεία ενός χρήστη με κριτήριο το username από τη βάση δεδομένων του γραμματειακού και τα τοποθετεί στα πεδία της φόρμας. Συμπληρώνει επίσης και τις τιμές των πεδίων που αφορούν τον LDAP Server.

LdapStudent
-UID_NUMBER -GID_NUMBER -HOME_DIRECTORY -CN -EDUPERSONPRINCIPALNAME -EDUPERSONAFFILIATION -SCHACHOMEORGANIZATION -SCHACPERSONALUNIQUECODE -GREDUPERSONUNDERGRADUATEBRANCH -O -EDUPERSONORGDN
+LdapStudent() +getUID_NUMBER() +setUID_NUMBER(UID_NUMBER) : void +getGID_NUMBER() +setGID_NUMBER(GID_NUMBER) : void +getHOME_DIRECTORY() +setHOME_DIRECTORY(HOME_DIRECTORY) : void +getCN() +setCN(CN) : void +getEDUPERSONPRINCIPALNAME() +setEDUPERSONPRINCIPALNAME(EDUPERSONPRINCIPALNAME) : void +getEDUPERSONAFFILIATION() +setEDUPERSONAFFILIATION(EDUPERSONAFFILIATION) : void +getSCHACHOMEORGANIZATION() +setSCHACHOMEORGANIZATION(SCHACHOMEORGANIZATION) : void +getSCHACPERSONALUNIQUECODE() +setSCHACPERSONALUNIQUECODE(SCHACPERSONALUNIQUECODE) : void +getGREDUPERSONUNDERGRADUATEBRANCH() +setGREDUPERSONUNDERGRADUATEBRANCH(GREDUPERSONUNDERGRADUATEBRANCH) : void +getO() +setO(O) : void +getEDUPERSONORGDN() +setEDUPERSONORGDN(EDUPERSONORGDN) : void

Η κλάση **LdapStudent** περιλαμβάνει ένα σύνολο χαρακτηριστικών για την έγκυρη δημιουργία ενός χρήστη στον LDAP Server. Τα χαρακτηριστικά αυτά είναι το σύνολο των χαρακτηριστικών του φοιτητή (κλάση **Student**), καθώς και τα χαρακτηριστικά που φαίνονται στην κλάση LdapStudent.

A.5 Περιγραφή κλάσεων υποσυστήματος παροχής διεργασιών

Στις επόμενες παραγράφους περιγράφονται με λεπτομέρεια οι κλάσεις παροχής των διεργασιών του υποσυστήματος.

A.5.1 Κλάσεις παροχής διεργασίας συγχρονισμού δηλώσεων ακαδημαϊκής περιόδου

SyncPeriodRegistrationsProcess
+SyncPeriodRegistrationsProcess(DEPTCODE, AYEAR, TERM, CHIEFEMAIL) +getProcessState() +getCurrentUserTask() +completeAsyncTask(Map asyncparams)

Η κλάση **SyncPeriodRegistrationsProcess** διαχειρίζεται τη διεργασία συγχρονισμού δηλώσεων ακαδημαϊκής περιόδου και λειτουργεί ως ενδιάμεσο (interface) μεταξύ του υποσυστήματος παροχής διεργασιών ως υπηρεσίες Rest και της βιβλιοθήκης κλάσεων εκτέλεσης των BPM διεργασιών.

Ο δημιουργός της κλάσης **SyncPeriodRegistrationsProcess** δημιουργεί ένα αίτημα προς τη μηχανή επεξεργασίας διεργασιών (Process Engine) ώστε να εκτελέσει τη διεργασία **SyncPeriodRegistrationProcess** - **BPMN** (που περιγράφεται στο xml αρχείο SyncPeriodRegistrationProcess.bpmn) με παραμέτρους τον κωδικό τμήματος, το ακαδημαϊκό έτος, την ακαδημαϊκή περίοδο και την διεύθυνση ηλεκτρονικής αλληλογραφίας προϊσταμένου τμήματος. Το αίτημα μαζί με τις παραμέτρους υποβάλλονται με χρήση του αντικείμενου τύπου **KieSession**, το οποίο επιστρέφει ένα αντικείμενο τύπου **ProcessInstance**. Ο δημιουργός επίσης δημιουργεί τα αντικείμενα των κλάσεων **SyncPeriodRegistrationsSWIH** και **SyncPeriodRegistrationsAWIH** (που είναι υπεύθυνα για το συγχρονισμό της εκτέλεσης των

σύγχρονων και ασύγχρονων εργασιών διεργασίας -process tasks) και τα καταχωρεί επίσης στο KieSession. Το KieSession μεταφέρει όλες αυτές τις πληροφορίες (αίτημα, παραμέτρους, αρχείο SyncPeriodRegistrationProcess.bpmn και τα παραγόμενα αντικείμενα της κλάσης WorkItemHandler στο **ProcessEngine** (εσωτερικό αντικείμενο της βιβλιοθήκης) το οποίο - τελικώς- εκτελεί το BPM process. Το αντικείμενο τύπου **WorkItem** παρέχει πρόσβαση στην τρέχουσα εργασία (task).

Η μέθοδος **getProcessState** επιστρέφει την κατάσταση της διεργασίας χρησιμοποιώντας το αντικείμενο τύπου ProcessInstance.

Η μέθοδος **getCurrentUserTask** επιστρέφει το όνομα της τρέχουσας (ασύγχρονης) BPM εργασίας (task).

Η μέθοδος **completeAsyncTask** αιτείται την ολοκλήρωση της τρέχουσας ασύγχρονης εργασίας (task).

SyncPeriodRegistrationsSWIH
-chiefEmail
-secretaryEmail
-deptCode
-senderaddress
+getChiefEmail()
+setChiefEmail(chiefEmail) : void
+getSecretaryEmail()
+setSecretaryEmail(secretaryEmail) : void
+getDeptCode()
+setDeptCode(deptCode) : void
+getSenderaddress()
+setSenderaddress(senderaddress) : void
+executeWorkItem(WorkItem workItem, WorkItemManager manager)
+abortWorkItem(WorkItem workItem, WorkItemManager manager)
+postMail(SUBJECT, MESSAGE, REPLYADDRESS)

Η κλάση **SyncPeriodRegistrationsSWIH** εκτελεί και συγχρονίζει την εκτέλεση των σύγχρονων εργασιών της διεργασίας. Αν και το αντικείμενο τύπου SyncPeriodRegistrationsSWIH δημιουργείται από την κλάση SyncPeriodRegistartionsProcess η εκτέλεση των εργασιών (tasks) γίνεται από το Process Engine.

Κάθε φορά που ενεργοποιείται μια σύγχρονη εργασία (task) εκτελείται η μέθοδος **executeWorkItem(WorkItem workItem, WorkItemManager manager)**. Επειδή κάθε φορά εκτελείται η ίδια μέθοδος ο κώδικας αυτής της μεθόδου περιέχει αρκετές εντολές If (μια για κάθε process task).

Η μέθοδος **abortWorkItem**(WorkItem workItem, WorkItemManager manager) ακυρώνει την εκτέλεση μιας εργασίας.

Η μέθοδος **postMail** αποστέλλει e-mail . Η μέθοδος **executeWorkItem** καλεί την **postMail** εκ μέρους της εργασίας "InformChief" ώστε να ενημερωθεί ο προϊστάμενος του τμήματος.

SyncPeriodRegistrationsAWIH
-deptCode -AYEAR -TERM -outparams
+getDeptCode() +setDeptCode(deptCode) : void +getAYEAR() +setAYEAR(AYEAR) : void +getTERM() +setTERM(TERM) : void +executeWorkItem(WorkItem workItem, WorkItemManager manager) +abortWorkItem(WorkItem workItem, WorkItemManager manager) +getOutparams() : Map<String, Object> +setOutparams(outparams) : void

Η κλάση **SyncPeriodRegistrationsAWIH** εκτελεί και συγχρονίζει την εκτέλεση των ασύγχρονων εργασιών. Αν και το αντικείμενο τύπου **SyncPeriodRegistrationsAWIH** δημιουργείται από την κλάση **SyncPeriodRegistrationsProcess** η εκτέλεση των εργασιών γίνεται από τη μηχανή διεργασιών(**Process Engine**).

Κάθε φορά που ενεργοποιείται μια ασύγχρονη εργασία εκτελείται η μέθοδος **executeWorkItem**(**WorkItem** workItem, **WorkItemManager** manager). Επειδή κάθε φορά εκτελείται η ίδια μέθοδος ο κώδικας αυτής της μεθόδου περιέχει αρκετές εντολές **If** (μια για κάθε process task).

Η μέθοδος **abortWorkItem**(**WorkItem** workItem, **WorkItemManager** manager) ακυρώνει την εκτέλεση μιας εργασίας.

SyncPeriodRegistrationsService
+SyncPeriodRegistrationsService()
+createSyncPeriodRegistrationsProcess(DEPTCODE, AYEAR, TERM, TEACHEREMAIL)
+getCurrentAsyncTask(DEPTCODE, AYEAR, TERM)
+completeSyncPeriodRegistrationsAsyncTask(DEPTCODE, AYEAR, TERM, MESSAGE)
+getSyncPeriodRegistrationsProcessByDeptCode(DEPTCODE, AYEAR, TERM)
+clearCompletedProcessVector()
+getSyncPeriodRegistrationsRequests()
+GetSyncPeriodRegistrationsRequestsCount()

Η κλάση **SyncPeriodRegistrationsService** διαχειρίζεται πολλαπλά αιτήματα δημιουργίας συγχρονισμού δηλώσεων ακαδημαϊκών περιόδων και συντονίζει την παράλληλη εκτέλεση των ενεργών διεργασιών SyncPeriodRegistrationsProcess. Όλα τα αντικείμενα διεργασιών αποθηκεύονται σε μια λίστα αντικειμένων (Vector).

Ο δημιουργός της κλάσης **SyncPeriodRegistrationsService** δημιουργεί ένα κενό Vector από αιτήματα δημιουργίας συγχρονισμού δηλώσεων ακαδημαϊκών περιόδων.

Η μέθοδος **createSyncPeriodRegistrationsProcess**(DEPTCODE, AYEAR, TERM, TEACHEREMAIL) αιτείται με βάση τις παραμέτρους (κωδικός τμήματος, ακαδημαϊκό έτος, ακαδημαϊκή περίοδος, διεύθυνση ηλεκτρονικής αλληλογραφίας) και δημιουργεί ένα αντικείμενο SyncPeriodRegistrationsProcess για τη διαχείριση μιας διεργασίας συγχρονισμού δηλώσεων ακαδημαϊκής περιόδου.

Η μέθοδος **getCurrentAsyncTask**(DEPTCODE, AYEAR, TERM) επιστρέφει το όνομα της τρέχουσας ασύγχρονης εργασίας για τη διεργασία που προσδιορίζεται από κωδικό τμήματος, ακαδημαϊκό έτος και ακαδημαϊκή περίοδο.

Η μέθοδος **completeSyncPeriodRegistrationsAsyncTask**(DEPTCODE, AYEAR, TERM, MESSAGE) ολοκληρώνει την τρέχουσα ασύγχρονη εργασία που προσδιορίζεται από κωδικό τμήματος, ακαδημαϊκό έτος και ακαδημαϊκή περίοδο. Το μήνυμα MESSAGE στέλνεται ως παράμετρος εξόδου της τρέχουσας εργασίας.

Η μέθοδος **getSyncPeriodRegistrationsProcessByDeptCode**(DEPTCODE, AYEAR, TERM) επιστρέφει τη διεργασία (αντικείμενο SyncPeriodRegistrationsProcess) που προσδιορίζεται από κωδικό τμήματος, ακαδημαϊκό έτος και ακαδημαϊκή περίοδο.

Η μέθοδος **clearCompletedProcessVector** μηδενίζει το Vector από αιτήματα δημιουργίας συγχρονισμού δηλώσεων ακαδημαϊκών περιόδων.

Η μέθοδος **getSyncPeriodRegistrationsRequests** επιστρέφει τις ενεργές διεργασίες που προσδιορίζεται από κωδικό τμήματος, ακαδημαϊκό έτος και ακαδημαϊκή περίοδο.

Η μέθοδος **GetSyncPeriodRegistrationsRequestsCount** επιστρέφει το πλήθος των ενεργών διεργασιών συγχρονισμού δηλώσεων ακαδημαϊκών περιόδων.

A.5.2 Κλάσεις παροχής διεργασίας προετοιμασίας νέας ακαδημαϊκής περιόδου

NewAcadPeriodProcess
+NewAcadPeriodProcess(DEPTCODE, AYEAR, TERM, CHIEFEMAIL, SECRETARYEMAIL) +getProcessState() +getCurrentUserTask() +completeAsyncTask(Map asyncparams)

Η κλάση **NewAcadPeriodProcess** διαχειρίζεται τη διεργασία προετοιμασίας νέας ακαδημαϊκής περιόδου και λειτουργεί ως ενδιάμεσο (interface) μεταξύ του υποσυστήματος παροχής διεργασιών ως υπηρεσίες Rest και της βιβλιοθήκης κλάσεων εκτέλεσης των BPM Processes.

Ο δημιουργός της κλάσης **NewAcadPeriodProcess** δημιουργεί ένα αίτημα προς το Process Engine ώστε να εκτελέσει τη διεργασία **NewAcadPeriodProcess - BPMN** (που περιγράφεται στο xml αρχείο NewAcadPeriodProcess.bpmn) με παραμέτρους τον κωδικό τμήματος, το ακαδημαϊκό έτος, την ακαδημαϊκή περίοδο, τη διεύθυνση ηλεκτρονικής αλληλογραφίας προϊσταμένου τμήματος και την ηλεκτρονική διεύθυνση της γραμματείας του τμήματος. Το αίτημα μαζί με τις παραμέτρους υποβάλλονται με χρήση του αντικείμενου τύπου **KieSession**, το οποίο επιστρέφει ένα αντικείμενο τύπου **ProcessInstance**. Ο δημιουργός επίσης δημιουργεί τα αντικείμενα των κλάσεων **SyncPeriodRegistrationsSWIH** και **SyncPeriodRegistrationsAWIH** (που είναι υπεύθυνα για το συγχρονισμό της εκτέλεσης των σύγχρονων και ασύγχρονων εργασιών) και τα καταχωρεί επίσης στο KieSession. Το KieSession μεταφέρει όλες αυτές τις πληροφορίες (αίτημα, παραμέτρους, αρχείο NewAcadPeriodProcess.bpmn και τα παραγόμενα αντικείμενα της κλάσης WorkItemHandler στο **ProcessEngine** (εσωτερικό αντικείμενο της βιβλιοθήκης) το οποίο εκτελεί το BPM process. Το αντικείμενο τύπου **WorkItem** παρέχει πρόσβαση στην τρέχουσα ενεργή εργασία.

Η μέθοδος **getProcessState** επιστρέφει την κατάσταση της διεργασίας χρησιμοποιώντας το αντικείμενο τύπου `ProcessInstance`.

Η μέθοδος **getCurrentUserTask** επιστρέφει το όνομα της τρέχουσας (ασύγχρονης) εργασίας.

Η μέθοδος **completeAsyncTask** αιτείται την ολοκλήρωση της τρέχουσας ασύγχρονης εργασίας.

NewAcadPeriodSWIH
-chiefEmail
-secretaryEmail
-deptCode
-senderaddress
-outparams
-attribute
+getChiefEmail()
+setChiefEmail(chiefEmail) : void
+getSecretaryEmail()
+setSecretaryEmail(secretaryEmail) : void
+getDeptCode()
+setDeptCode(deptCode) : void
+getSenderaddress()
+setSenderaddress(senderaddress) : void
+executeWorkItem(WorkItem workItem, WorkItemManager manager)
+abortWorkItem(WorkItem workItem, WorkItemManager manager)
+postMail(SUBJECT, MESSAGE, REPLYADDRESS)
+getOutparams()
+setOutparams(outparams) : void

Η κλάση **NewAcadPeriodSWIH** εκτελεί και συγχρονίζει την εκτέλεση των σύγχρονων εργασιών. Αν και το αντικείμενο τύπου `SyncPeriodRegistrationsSWIH` δημιουργείται από την κλάση `NewAcadPeriodProcess` η εκτέλεση των εργασιών γίνεται από το `Process Engine`.

Κάθε φορά που ενεργοποιείται μια σύγχρονη εργασία εκτελείται η μέθοδος **executeWorkItem(WorkItem workItem, WorkItemManager manager)**. Επειδή κάθε φορά εκτελείται η ίδια μέθοδος ο κώδικας αυτής της μεθόδου περιέχει αρκετές εντολές `If` (μια για κάθε `process task`). Η μέθοδος αυτή διεκπεραιώνει τις εξής εργασίες (`tasks`): *ChiefRequestPreparePeriod, InformSyncPendingEmail, InformChief* και *InformSecretary*.

Η μέθοδος **abortWorkItem(WorkItem workItem, WorkItemManager manager)** ακυρώνει την εκτέλεση μιας εργασίας.

Η μέθοδος **postMail** αποστέλλει e-mail.

NewAcadPeriodAWIH
-deptCode -AYEAR -TERM -outparams
+getDeptCode() +setDeptCode(deptCode) : void +getAYEAR() +setAYEAR(AYEAR) : void +getTERM() +setTERM(TERM) : void +executeWorkItem(WorkItem workItem, WorkItemManager manager) +abortWorkItem(WorkItem workItem, WorkItemManager manager) +getOutparams() : Map<String, Object> +setOutparams(outparams) : void

Η κλάση **NewAcadPeriodAWIH** εκτελεί και συγχρονίζει την εκτέλεση των ασύγχρονων εργασιών. Αν και το αντικείμενο τύπου NewAcadPeriodAWIH δημιουργείται από την κλάση NewAcadPeriodAWIH η εκτέλεση των εργασιών γίνεται από το Process Engine.

Κάθε φορά που ενεργοποιείται μια ασύγχρονη εργασία εκτελείται η μέθοδος **executeWorkItem(WorkItem workItem, WorkItemManager manager)**. Επειδή κάθε φορά εκτελείται η ίδια μέθοδος ο κώδικας αυτής της μεθόδου περιέχει αρκετές εντολές If (μια για κάθε process task). Η μέθοδος αυτή διεκπεραιώνει τα εξής εργασίες: *PrepareComplete*, *SyncStudents*, *SyncTeachers* και *SyncLessons*.

Η μέθοδος **abortWorkItem(WorkItem workItem, WorkItemManager manager)** ακυρώνει την εκτέλεση μιας εργασίας.

NewAcadPeriodService
+NewAcadPeriodService() +createNewAcadPeriodProcess(DEPTCODE, AYEAR, TERM, TEACHEREMAIL, SECRET... +getCurrentAsyncTask(DEPTCODE) +prepareCompleteAsyncTask(String DEPTCODE) +completeCurrentAsyncTask(DEPTCODE, FLAGVAR, FLAGVALUE, ERRORMESSAGE) +getNewAcadPeriodProcessByDeptCode(DEPTCODE) +completedProcessStatus(DEPTCODE) +clearCompletedProcessVector() +getNewAcadPeriodRequestsDeptcodes() +getNewAcadPeriodRequestsCount()

Η κλάση **NewAcadPeriodService** διαχειρίζεται πολλαπλά αιτήματα προετοιμασίας νέων ακαδημαϊκών περιόδων και συντονίζει την παράλληλη εκτέλεση των ενεργών διεργασιών NewAcadPeriodProcess. Όλα τα αντικείμενα διεργασιών αποθηκεύονται σε μια λίστα αντικειμένων (Vector).

Ο δημιουργός της κλάσης **NewAcadPeriodService** δημιουργεί ένα κενό Vector από αιτήματα προετοιμασίας νέων ακαδημαϊκών περιόδων.

Η μέθοδος **createNewAcadPeriodProcess**(DEPTCODE, AYEAR, TERM, TEACHEREMAIL, SECRETARYEMAIL) αιτείται με βάση τις παραμέτρους (κωδικός τμήματος, ακαδημαϊκό έτος, ακαδημαϊκή περίοδος, διεύθυνση ηλεκτρονικής αλληλογραφίας προϊσταμένου και διεύθυνση ηλεκτρονικού ταχυδρομείου γραμματείας τμήματος) και δημιουργεί ένα αντικείμενο NewAcadPeriodProcess για τη διαχείριση μιας διεργασίας προετοιμασίας νέας ακαδημαϊκής περιόδου.

Η μέθοδος **getCurrentAsyncTask**(DEPTCODE) επιστρέφει το όνομα της τρέχουσας ασύγχρονης εργασίας για τη διεργασία που προσδιορίζεται από τον κωδικό του τμήματος.

Η μέθοδος **prepareCompleteAsyncTask**(DEPTCODE) ολοκληρώνει την τρέχουσα ασύγχρονη εργασία "prepareComplete" που προσδιορίζεται από τον κωδικό τμήματος.

Η μέθοδος **completeCurrentAsyncTask** (DEPTCODE, FLAGVAR, FLAGVALUE, ERRORMESSAGE) ολοκληρώνει την τρέχουσα ασύγχρονη εργασία που προσδιορίζεται από κωδικό τμήματος. Η μέθοδος αυτή χρησιμοποιείται για την ολοκλήρωση των εργασιών: *SyncStudents*, *SyncTeachers* και *SyncLessons*. Το μήνυμα ERRORMESSAGE στέλνεται ως παράμετρος εξόδου την τρέχουσα ασύγχρονη εργασίας σε περίπτωση μη σωστής προετοιμασίας της ανάλογης εργασίας από τη γραμματεία. Οι μεταβλητές FLAGVAR, FLAGVALUE αποτελούν παραμέτρους εξόδου που αποστέλλονται από τον διαχειριστή του υποσυστήματος συγχρονισμού και δηλώνουν την επιτυχία ή αποτυχία ολοκλήρωσης της εργασίας.

Η μέθοδος **getNewAcadPeriodProcessByDeptCode**(DEPTCODE) επιστρέφει τη διεργασία (αντικείμενο NewAcadPeriodProcess) που προσδιορίζεται από τον κωδικό του τμήματος.

Η μέθοδος **completedProcessStatus**(DEPTCODE) επιστρέφει την κατάσταση της διεργασίας που προσδιορίζεται από τον κωδικό τμήματος.

Η μέθοδος **clearCompletedProcessVector** μηδενίζει το Vector από αιτήματα προετοιμασίας νέων ακαδημαϊκών περιόδων.

Η μέθοδος **getNewAcadPeriodRequestsDeptcodes** επιστρέφει τις ενεργές διεργασίες που προσδιορίζεται από τον κωδικό του τμήματος.

Η μέθοδος **getNewAcadPeriodRequestsCount** επιστρέφει το πλήθος των ενεργών διεργασιών προετοιμασίας νέων ακαδημαϊκών περιόδων.

A.5.3 Κλάσεις παροχής διεργασίας κορεσμού ομάδων παρακολούθησης.

```
LessonGroupsFullProcess
+LessonGroupsFullProcess(DEPTNO, LESSONCODE, CHIEFEMAIL, TEACHEREMAIL, STUDENTSEMAILS)
+getProcessState()
+getCurrentUserTask()
+completeAsyncTask(Map asyncparams)
+getGroupUpdateSuggestion()
```

Η κλάση **LessonGroupsFullProcess** διαχειρίζεται τη διεργασία κορεσμού ομάδων παρακολούθησης ενός μαθήματος και λειτουργεί ως ενδιάμεσο (interface) μεταξύ του υποσυστήματος παροχής διεργασιών ως υπηρεσίες Rest και της βιβλιοθήκης κλάσεων εκτέλεσης των BPM διεργασιών.

Ο δημιουργός της κλάσης **LessonGroupsFullProcess** δημιουργεί ένα αίτημα προς το Process Engine ώστε να εκτελέσει τη διεργασία **LessonGroupsFullProcess - BPMN** (που περιγράφεται στο xml αρχείο LessonGroupsFullProcess.bpmn) με παραμέτρους τον κωδικό τμήματος, τον κωδικό μαθήματος, την διεύθυνση ηλεκτρονικής αλληλογραφίας προϊσταμένου τμήματος, την ηλεκτρονική διεύθυνση υπεύθυνου του μαθήματος και τις ηλεκτρονικές διευθύνσεις των φοιτητών που έχουν δηλωμένο το μάθημα στο εξωτερικό σύστημα της ηλεκτρονικής γραμματείας. Το αίτημα μαζί με τις παραμέτρους υποβάλλονται με χρήση του αντικείμενου τύπου **KieSession**, το οποίο επιστρέφει ένα αντικείμενο τύπου **ProcessInstance**. Ο δημιουργός επίσης δημιουργεί τα αντικείμενα των κλάσεων **LessonGroupsFullSWIH** και **LessonGroupsFullAWIH** (που είναι υπεύθυνα για το συγχρονισμό της εκτέλεσης των σύγχρονων και ασύγχρονων εργασιών) και τα καταχωρεί επίσης στο KieSession. Το KieSession μεταφέρει όλες αυτές τις πληροφορίες (αίτημα, παραμέτρους, αρχείο LessonGroupsFullProcess.bpmn και τα παραγόμενα αντικείμενα της κλάσης WorkItemHandler στο **ProcessEngine** (εσωτερικό αντικείμενο της βιβλιοθήκης) το οποίο εκτελεί τη BPM διεργασία. Το αντικείμενο τύπου **WorkItem** παρέχει πρόσβαση στην τρέχουσα ενεργή εργασία.

Η μέθοδος **getProcessState** επιστρέφει την κατάσταση της διεργασίας χρησιμοποιώντας το αντικείμενο τύπου ProcessInstance.

Η μέθοδος **getCurrentUserTask** επιστρέφει το όνομα της τρέχουσα ασύγχρονης ενεργή εργασίας.

Η μέθοδος **completeAsyncTask(Map asyncparams)** αιτείται την ολοκλήρωση της τρέχουσας ενεργής εργασίας.

Η μέθοδος **getGroupUpdateSuggestion** επιστρέφει την εισήγηση του υπεύθυνου εκπαιδευτικού προς τον προϊστάμενο αφού εξέτασε το αίτημα κορεσμού.

LessonGroupsFullSWIH
-Deptno
-LessonCode
-GroupUpdateSuggestion
-TeacherEmail
-outparams
+getDeptno()
+setDeptno(Deptno) : void
+getLessonCode()
+setLessonCode(LessonCode) : void
+getGroupUpdateSuggestion()
+setGroupUpdateSuggestion(GroupUpdateSuggestion) : void
+getTeacherEmail()
+setTeacherEmail(TeacherEmail) : void
+getOutparams()
+setOutparams(outparams) : void
+executeWorkItem(WorkItem workItem, WorkItemManager manager)
+abortWorkItem(WorkItem workItem, WorkItemManager manager)
+postMail(SUBJECT, MESSAGE, REPLYADDRESS)

Η κλάση **LessonGroupsFullSWIH** εκτελεί και συγχρονίζει την εκτέλεση των σύγχρονων εργασιών. Αν και το αντικείμενο τύπου LessonGroupsFullSWIH δημιουργείται από την κλάση LessonGroupsFullProcess η εκτέλεση των εργασιών γίνεται από το Process Engine.

Κάθε φορά που ενεργοποιείται μια σύγχρονη εργασία εκτελείται η μέθοδος **executeWorkItem(WorkItem workItem, WorkItemManager manager)**. Επειδή κάθε φορά εκτελείται η ίδια μέθοδος ο κώδικας αυτής της μεθόδου περιέχει αρκετές εντολές If (μια για κάθε process task). Η μέθοδος αυτή διεκπεραιώνει τα εξής tasks: *LessonGroupsFull*, *InformStudents* και *InformProfessor*.

Η μέθοδος **abortWorkItem(WorkItem workItem, WorkItemManager manager)** ακυρώνει την εκτέλεση μιας εργασίας.

Η μέθοδος **postMail** αποστέλλει e-mail.

LessonGroupsFullAWIH
-Deptno -LessonCode -GroupUpdateSuggestion -myresults
+getDeptno() +setDeptno(Deptno) : void +getLessonCode() +setLessonCode(LessonCode) : void +getGroupUpdateSuggestion() +setGroupUpdateSuggestion(GroupUpdateSuggestion) : void +getMyresults() +setMyresults(myresults) : void +getAttribute() +setAttribute(attribute) : void +executeWorkItem(WorkItem workItem, WorkItemManager manager) +abortWorkItem(WorkItem workItem, WorkItemManager manager)

Η κλάση **LessonGroupsFullAWIH** εκτελεί και συγχρονίζει την εκτέλεση των ασύγχρονων εργασιών. Αν και το αντικείμενο τύπου LessonGroupsFullAWIH δημιουργείται από την κλάση LessonGroupsFullProcess η εκτέλεση των εργασιών γίνεται από το Process Engine.

Κάθε φορά που ενεργοποιείται μια ασύγχρονη εργασία εκτελείται η μέθοδος **executeWorkItem(WorkItem workItem, WorkItemManager manager)**. Επειδή κάθε φορά εκτελείται η ίδια μέθοδος ο κώδικας αυτής της μεθόδου περιέχει αρκετές εντολές If (για κάθε process task). Η μέθοδος αυτή διεκπεραιώνει τις εξής εργασίες: *ProfessorGroupUpdateRequest* και *ChiefResponse*.

Η μέθοδος **abortWorkItem(WorkItem workItem, WorkItemManager manager)** ακυρώνει την εκτέλεση μιας εργασίας.

LessonGroupsFullService
+LessonGroupsFullService() +createLessonGroupsFullProcess(DEPTCODE, LESSONCODE, CHIEFEMAIL, TEACHEREMAIL, STUDENTSEMAILS) +getCurrentAsyncTask(DEPTCODE, LESSONCODE) +getLessonGroupsFullProcess(DEPTCODE, LESSONCODE) +completeProfessorGroupUpdateRequestAsyncTask(DEPTCODE, LESSONCODE, FLAG, SUGGESTION) +completeChiefResponseAsyncTask(DEPTCODE, LESSONCODE, FLAG, MESSAGE) +clearCompletedProcessVector() +getLessonGroupsFullProcessRequests() +getLessonGroupsFullProcessRequestsCount()

Η κλάση **LessonGroupsFullService** διαχειρίζεται πολλαπλά αιτήματα κορεσμού ομάδων παρακολούθησης μαθημάτων και συντονίζει την παράλληλη εκτέλεση των ενεργών διεργασιών LessonGroupsFullProcess. Όλα τα αντικείμενα διεργασιών αποθηκεύονται σε μια λίστα αντικειμένων (Vector).

Ο δημιουργός της κλάσης **LessonGroupsFullService** δημιουργεί ένα κενό Vector από αιτήματα κορεσμού ομάδων παρακολούθησης μαθημάτων.

Η μέθοδος **createLessonGroupsFullProcess**(DEPTCODE, LESSONCODE, CHIEFEMAIL, TEACHEREMAIL, STUDENTSEMAILS) αιτείται με βάση τις παραμέτρους (κωδικό τμήματος, κωδικό μαθήματος, ηλεκτρονική διεύθυνση προϊστάμενου τμήματος, ηλεκτρονική διεύθυνση υπεύθυνου του μαθήματος και ηλεκτρονικές διευθύνσεις των φοιτητών που έχουν δηλωμένο το μάθημα στο εξωτερικό σύστημα της ηλεκτρονικής γραμματείας) και δημιουργεί ένα αντικείμενο LessonGroupsFullProcess για τη διαχείριση μιας διεργασίας κορεσμού ομάδων παρακολούθησης ενός μαθήματος.

Η μέθοδος **getCurrentAsyncTask**(DEPTCODE, LESSONCODE) επιστρέφει το όνομα της τρέχουσας ασύγχρονης εργασίας για τη διεργασία που προσδιορίζεται από τον κωδικό τμήματος και τον κωδικό μαθήματος.

Η μέθοδος **getLessonGroupsFullProcess**(DEPTCODE, LESSONCODE) επιστρέφει τη διεργασία (αντικείμενο LessonGroupsFullProcess) που προσδιορίζεται από τον κωδικό τμήματος και τον κωδικό μαθήματος.

Η μέθοδος **completeProfessorGroupUpdateRequestAsyncTask**(DEPTCODE, LESSONCODE, FLAG, SUGGESTION) ολοκληρώνει την εργασία “ProfessorGroupUpdateRequest” της διεργασίας που προσδιορίζεται από τον κωδικό τμήματος και τον κωδικό μαθήματος. Η παράμετρος FLAG παίρνει τιμές TRUE ή FALSE που σημαίνει την προώθηση (από τον υπεύθυνο εκπαιδευτικό προς τον προϊστάμενο) ή την απόρριψη του αιτήματος κορεσμού ομάδων παρακολούθησης. Η παράμετρος SUGGESTION είναι η εισήγηση του υπεύθυνου εκπαιδευτικού.

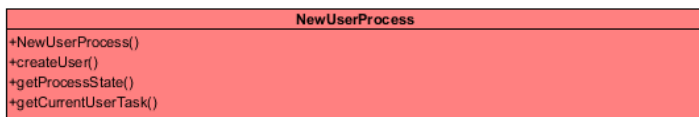
Η μέθοδος **completeChiefResponseAsyncTask**(DEPTCODE, LESSONCODE, FLAG, MESSAGE) ολοκληρώνει την εργασία “ChiefResponse” της διεργασίας που προσδιορίζεται από τον κωδικό τμήματος και τον κωδικό μαθήματος. Η παράμετρος FLAG παίρνει τιμές TRUE ή FALSE που σημαίνει την αποδοχή (από τον προϊστάμενο) ή την απόρριψη του αιτήματος κορεσμού ομάδων παρακολούθησης. Η παράμετρος MESSAGE είναι η απάντηση του υπεύθυνου προϊστάμενου.

Η μέθοδος **clearCompletedProcessVector** μηδενίζει το Vector από αιτήματα κορεσμού δηλώσεων ομάδων παρακολούθησης μαθημάτων.

Η μέθοδος **getLessonGroupsFullProcessRequests** επιστρέφει τις ενεργές διεργασίες που προσδιορίζονται από κωδικό τμήματος και κωδικό μαθήματος.

Η μέθοδος **getLessonGroupsFullProcessRequestsCount** επιστρέφει το πλήθος των ενεργών διεργασιών κορεσμού δηλώσεων ομάδων παρακολούθησης μαθημάτων.

A.5.4 Κλάσεις παροχής διεργασίας συγχρονισμού λογαριασμού νέου χρήστη



Η κλάση **NewUserProcess** διαχειρίζεται τη διεργασία συγχρονισμού λογαριασμού νέου φοιτητή τμήματος και λειτουργεί ως ενδιάμεσο (interface) μεταξύ του υποσυστήματος παροχής διεργασιών ως υπηρεσίες Rest και της βιβλιοθήκης κλάσεων εκτέλεσης των BPM διεργασιών.

Ο δημιουργός της κλάσης **NewUserProcess** δημιουργεί ένα αίτημα προς το Process Engine ώστε να εκτελέσει τη διεργασία **NewUserProcess - BPMN** (που περιγράφεται στο xml αρχείο **NewUserProcess.bpmn**). Το αίτημα υποβάλλεται με χρήση του αντικείμενου τύπου **KieSession**, το οποίο επιστρέφει ένα αντικείμενο τύπου **ProcessInstance**. Ο δημιουργός επίσης δημιουργεί τα αντικείμενα των κλάσεων **NewUserSWIH** και **NewUserAWIH** (που είναι υπεύθυνα για το συγχρονισμό της εκτέλεσης των σύγχρονων και ασύγχρονων εργασιών) και τα καταχωρεί επίσης στο **KieSession**. Το **KieSession** μεταφέρει όλες αυτές τις πληροφορίες (αίτημα, αρχείο **NewUserProcess.bpmn** και τα παραγόμενα αντικείμενα της κλάσης **WorkItemHandler** στο **ProcessEngine** (εσωτερικό αντικείμενο της βιβλιοθήκης) το οποίο εκτελεί την BPM διεργασία. Το αντικείμενο τύπου **WorkItem** παρέχει πρόσβαση στην τρέχουσα ενεργή εργασία.

Η μέθοδος **getProcessState** επιστρέφει την κατάσταση της διεργασίας χρησιμοποιώντας το αντικείμενο τύπου **ProcessInstance**.

Η μέθοδος **getCurrentUserTask** επιστρέφει το όνομα της τρέχουσας (ασύγχρονης) εργασίας.

NewUserSWIH
-username -senderaddress -outparams
+getUsername() +setUsername(username) : void +getSenderaddress() +setSenderaddress(senderaddress) : void +getOutparams() +setOutparams(outparams) : void +executeWorkItem(WorkItem workItem, WorkItemManager manager) +abortWorkItem(WorkItem workItem, WorkItemManager manager) +checkEmailAction() +receiveMail() +postMail(SUBJECT, MESSAGE, REPLYADDRESS) +CheckLdapUserAction(USERNAME)

Η κλάση **NewUserSWIH** εκτελεί και συγχρονίζει την εκτέλεση των σύγχρονων εργασιών. Αν και το αντικείμενο τύπου NewUserSWIH δημιουργείται από την κλάση NewUserProcess η εκτέλεση των εργασιών γίνεται από το Process Engine.

Κάθε φορά που ενεργοποιείται μια σύγχρονη εργασία εκτελείται η μέθοδος **executeWorkItem(WorkItem workItem, WorkItemManager manager)**. Επειδή κάθε φορά εκτελείται η ίδια μέθοδος ο κώδικας αυτής της μεθόδου περιέχει αρκετές εντολές If (μια για κάθε process task). Η μέθοδος αυτή διεκπεραιώνει τις εξής εργασίες: *CheckEmail*, *CheckLdapUser* και *InformSecretary*.

Η μέθοδος **abortWorkItem(WorkItem workItem, WorkItemManager manager)** ακυρώνει την εκτέλεση μιας εργασίας.

Η μέθοδος **checkEmailAction** ελέγχει αν έχει έρθει email καθώς και αν το email αυτό έχει μορφή που αφορά αίτημα συγχρονισμού νέου χρήστη.

Η μέθοδος **receiveMail** διαβάζει ένα email από συγκεκριμένο λογαριασμό.

Η μέθοδος **postMail** αποστέλλει e-mail.

Η μέθοδος **CheckLdapUserAction(USERNAME)** ψάχνει στον LDAP Server του ιδρύματος αν ο χρήστης με το όνομα λογαριασμού της παραμέτρου υπάρχει ή όχι.

NewUserAWIH
-username -outparams
+getUsername() +setUsername(username) : void +getOutparams() : Map<String, Object> +setOutparams(outparams) : void +executeWorkItem(WorkItem workItem, WorkItemManager manager) +abortWorkItem(WorkItem workItem, WorkItemManager manager)

Η κλάση **NewUserAWIH** εκτελεί και συγχρονίζει την εκτέλεση των ασύγχρονων εργασιών. Αν και το αντικείμενο τύπου **NewUserAWIH** δημιουργείται από την κλάση **NewUserProcess** η εκτέλεση των εργασιών γίνεται από το **Process Engine**.

Κάθε φορά που ενεργοποιείται μια ασύγχρονη εργασία εκτελείται η μέθοδος **executeWorkItem(WorkItem workItem, WorkItemManager manager)**. Επειδή κάθε φορά εκτελείται η ίδια μέθοδος ο κώδικας αυτής της μεθόδου περιέχει αρκετές εντολές **If** (μια για κάθε **process task**). Η μέθοδος αυτή διεκπεραιώνει την εργασία: *CreateUser*.

Η μέθοδος **abortWorkItem(WorkItem workItem, WorkItemManager manager)** ακυρώνει την εκτέλεση μιας εργασίας.

NewUserService
-fname -lname
+NewUserService(fname, lname) +getFname() +setFname(fname) : void +getLname() +setLname(lname) : void +createNewUserProcess() +createUser(USERNAME) +getNewUserProcessByUsername(USERNAME) +clearCompletedProcessVector() +getNewUserRequestsUsernames() +getNewUserRequestsCount()

Η κλάση **NewUserService** διαχειρίζεται πολλαπλά αιτήματα συγχρονισμού λογαριασμών φοιτητών και συντονίζοντας την παράλληλη εκτέλεση των ενεργών διεργασιών **NewUserProcess**. Όλα τα αντικείμενα διεργασιών αποθηκεύονται σε μια λίστα αντικειμένων (**Vector**).

Ο δημιουργός της κλάσης **NewUserService** δημιουργεί ένα κενό **Vector** από αιτήματα συγχρονισμού λογαριασμών φοιτητών.

Η μέθοδος **createNewUserProcess** αιτείται και δημιουργεί ένα αντικείμενο **NewUserProcess** για τη διαχείριση μιας διεργασίας συγχρονισμού λογαριασμού ενός φοιτητή.

Η μέθοδος **createUser(USERNAME)** ολοκληρώνει το task **CreateUser**.

Η μέθοδος **getNewUserProcessByUsername(USERNAME)** επιστρέφει τη διεργασία (αντικείμενο **NewUserProcess**) που προσδιορίζεται από το όνομα λογαριασμού της παραμέτρου.

Η μέθοδος **clearCompletedProcessVector** μηδενίζει το **Vector** από αιτήματα συγχρονισμού λογαριασμού φοιτητών.

Η μέθοδος **getNewUserRequestsUsernames** επιστρέφει τις ενεργές διεργασίες που προσδιορίζονται από τα ονόματα λογαριασμού των φοιτητών.

Η μέθοδος **getNewUserRequestsCount** επιστρέφει το πλήθος των ενεργών διεργασιών συγχρονισμού λογαριασμού των φοιτητών.

A.6 Περιγραφή κλάσεων υποσυστήματος Android φοιτητών

Student
-AEM -LNAME -FNAME -EMAIL -SEMESTER -UNAME -PASSWD -DEPTNO
+getAEM() +setAEM(AEM) : void +getLNAME() +setLNAME(LNAME) : void +getFNAME() +setFNAME(FNAME) : void +getEmail() +setEMAIL(EMAIL) : void +getSEMESTER() +setSEMESTER(SEMESTER) : void +getUNAME() +setUNAME(UNAME) : void +getPASSWD() +setPASSWD(PASSWD) : void +getDEPTNO() +setDEPTNO(DEPTNO) : void +Student()

Η κλάση **Student** περιέχει τα χαρακτηριστικά του φοιτητή: αριθμός μητρώου (**AEM**), επώνυμο (**LNAME**), όνομα (**FNAME**), διεύθυνση ηλεκτρονικής αλληλογραφίας (**EMAIL**), τρέχον εξάμηνο

φοίτησης (**SEMESTER**), όνομα λογαριασμού (**UNAME**), κωδικός λογαριασμού (**PASSWD**), κωδικός τμήματος στο οποίο ανήκει ο φοιτητής (**DEPTNO**).

Lesson
-CODE -TITLE
+Lesson() +getCode() +setCode(CODE) : void +getTitle() +setTitle(TITLE) : void +addLSection(String NO, Integer DAY, String FULLTIME) +getLSections() : ArrayList<LSection> +clearLSections()

Η κλάση **Lesson** περιέχει τα χαρακτηριστικά ενός μαθήματος: κωδικός (**CODE**), τίτλος(**TITLE**). Κάθε αντικείμενο περιέχει ομάδες παρακολούθησης σε μια λίστα (`ArrayList<LSection>`).

Η μέθοδος **addLSection**(String NO, Integer DAY, String FULLTIME) δέχεται ως παραμέτρους τον κωδικό, ημέρα και την ώρα διεξαγωγής μιας ομάδας παρακολούθησης και την προσθέτει στη λίστα.

Η μέθοδος **getLSections** επιστρέφει τη λίστα με τις ομάδες παρακολούθησης του μαθήματος.

Η μέθοδος **clearLSections** διαγράφει τις ομάδες παρακολούθησης από τη λίστα.

LSection
-NO -DAYNO -FULLTIME
+LSection() +getNO() +setNO(NO) : void +getDAYNO() +setDAYNO(DAYNO) : void +getFULLTIME() +setFULLTIME(FULLTIME) : void +getDayName()

Η κλάση **LSECTION** περιέχει τα βασικά χαρακτηριστικά μια ομάδας παρακολούθησης μαθήματος: κωδικός ομάδας (**LSNO**), τον αριθμό της ημέρας της εβδομάδας που διεξάγεται η ομάδα μαθήματος (**DAYNO**) και ώρες διεξαγωγής ομάδας μαθήματος (**FULLTIME**).

Η μέθοδος **getDayName** επιστρέφει το όνομα της ημέρας της ομάδας παρακολούθησης σε κείμενο.

SLSVIEW
-DNO
-DNAME
-AEM
-LNAME
-SEMESTER
-CODE
-TITLE
-SEMESTERNO
-LSNO
-STARTTIME
-ENDTIME
-FULLTIME
-WDNAME
+getDNO()
+setDNO(DNO) : void
+getDNAME()
+setDNAME(DNAME) : void
+getAEM()
+setAEM(AEM) : void
+getLNAME()
+setLNAME(LNAME) : void
+getSEMESTER()
+setSEMESTER(SEMESTER) : void
+getCODE()
+setCODE(CODE) : void
+getTitle()
+setTitle(TITLE) : void
+getSEMESTERNO()
+setSEMESTERNO(SEMESTERNO) : void
+getLSNO()
+setLSNO(LSNO) : void
+getSTARTTIME()
+setSTARTTIME(STARTTIME) : void
+getENDTIME()
+setENDTIME(ENDTIME) : void
+getFULLTIME()
+setFULLTIME(FULLTIME) : void
+getWDNAME()
+setWDNAME(WDNAME) : void

Η κλάση **SLSVIEW** περιέχει τα χαρακτηριστικά μια όψης που περιέχει στοιχεία φοιτητή και στοιχεία μιας ομάδας παρακολούθησης ενός μαθήματος που έχει δηλώσει στο γραμματειακό σύστημα. Τα χαρακτηριστικά αυτά είναι: κωδικός τμήματος (**DNO**), όνομα τμήματος (**DNAME**), αριθμός μητρώου φοιτητή (**AEM**), επώνυμο (**LNAME**), όνομα (**FNAME**), εξάμηνο φοιτητή (**SEMESTER**), κωδικός μαθήματος (**CODE**), τίτλος (**TITLE**), εξάμηνο μαθήματος (**SEMESTERNO**), κωδικός ομάδας μαθήματος (**LSNO**), ώρες διεξαγωγής ομάδας μαθήματος (**STARTTIME, ENDTIME, FULLTIME, WDNAME**).

UnregisteredLessons
+UnregisteredLessons()
+addLesson(Lesson l)
+removeLesson(String Code)
+Lesson getLesson(String Code)
+getmLessons() : ArrayList<Lesson>
+clear()

Η κλάση **UnregisteredLessons** περιέχει μια λίστα (ArrayList<Lesson>) με τα μαθήματα όπου ο φοιτητής δεν έχει δηλώσει ομάδα παρακολούθησης.

Η μέθοδος **addLesson(Lesson l)** προσθέτει το μάθημα της παραμέτρου στη λίστα.

Η μέθοδος **removeLesson**(String Code) αφαιρεί το μάθημα με τον κωδικό της παραμέτρου από τη λίστα.

Η μέθοδος **getLesson**(String Code) ψάχνει στη λίστα και επιστρέφει το αντικείμενο (Lesson) του μαθήματος που έχει τον κωδικό της παραμέτρου.

Η μέθοδος **getmLessons** επιστρέφει τη λίστα με τα μαθήματα.

Η μέθοδος **clear** διαγράφει τα μαθήματα από τη λίστα.

RegisteredLessons
+RegisteredLessons()
+addLessonView(SLSVIEW sIs)
+getLessonView(String C) : SLSVIEW
+getmLessonsView() : ArrayList<SLSVIEW>
+clear()

Η κλάση **RegisteredLessons** περιέχει μια λίστα (ArrayList<SLSVIEW>) με τις ομάδες παρακολούθησης που έχει δηλώσει ο φοιτητής. Στη λίστα αυτή εμπεριέχονται και τα μαθήματα που ο φοιτητής έχει δηλώσει ομάδα παρακολούθησης.

Η μέθοδος **addLessonView**(SLSVIEW sIs) προσθέτει την ομάδα παρακολούθησης ενός μαθήματος (της παραμέτρου) στη λίστα.

Η μέθοδος **getLessonView**(String C) ψάχνει στη λίστα και επιστρέφει την ομάδα παρακολούθησης (αντικείμενο SLSVIEW) που έχει τον κωδικό μαθήματος της παραμέτρου.

Η μέθοδος **getmLessonsView** επιστρέφει τη λίστα με τις δηλωμένες ομάδες παρακολούθησης.

Η μέθοδος **clear** διαγράφει τις ομάδες παρακολούθησης από τη λίστα.

MainActivity
+onCreate()

Η κλάση **MainActivity** είναι η κλάση εκκίνησης της εφαρμογής Android. Η κλάση αυτή ενεργοποιεί την κύρια οθόνη διεπαφής (MainActivityFragment) της εφαρμογής.

Η κλάση **MainActivityFragment** είναι η κύρια οθόνη διεπαφής με το χρήστη. Περιλαμβάνει ένα σύνολο από οπτικά αντικείμενα (Widgets), όπως το όνομα του λογαριασμού(**EditText**), ο κωδικός(**EditText**), το μητρώο(**TextView**), το επώνυμο(**TextView**), το όνομα(**TextView**), το εξάμηνο(**TextView**), το τμήμα(**TextView**), κουμπιά (**Button**) σύνδεσης, αποσύνδεσης, προβολής μαθημάτων με δηλωμένη ή αδήλωτη ομάδα παρακολούθησης. Η κλάση περιέχει επίσης το χαρακτηριστικό **lastacadperiod** που εκφράζει την τελευταία ακαδημαϊκή περίοδο.

Η μέθοδος **setWidgetsStatus** καθορίζει ποια από τα οπτικά αντικείμενα θα είναι ενεργά και ποια όχι με βάση τις επιλογές που κάνει ο χρήστης.

Η μέθοδος **onCreateView** δημιουργεί τα οπτικά αντικείμενα και τους θέτει αρχικές τιμές και διαχειριστές συμβάντων.

Οι μέθοδοι **usernameText_addTextChangedListener**, **passwordText_addTextChangedListener** δημιουργούν αντικείμενα τύπου Listener και παρακολουθούν την πληκτρολόγηση κειμένου στα αντικείμενα usernameText και passwordText. Όταν ο χρήστης πληκτρολογήσει κείμενο ενεργοποιείται το κουμπί σύνδεσης.

Η μέθοδος **connectBtn_setOnClickListener** δημιουργεί ένα αντικείμενο τύπου Listener και ενεργοποιείται όταν ο χρήστης επιλέξει το κουμπί σύνδεσης. Κατά την ενεργοποίηση η μέθοδος χρησιμοποιεί την κλάση StudentLoginHttpClientTask για να ταχτοποιηθούν τα στοιχεία του χρήστη και να συνδεθεί στο σύστημα.

Η μέθοδος **unregisteredLessonsButton_setOnClickListener** δημιουργεί ένα αντικείμενο τύπου Listener και ενεργοποιείται όταν ο χρήστης επιλέξει το κουμπί unregisteredLessonsButton. Κατά την ενεργοποίηση η μέθοδος χρησιμοποιεί την κλάση findRegisteredLessonsWithoutLSectionHttpClientTask για να βρεθούν τα μαθήματα στα οποία ο φοιτητής δεν έχει δηλώσει ομάδα παρακολούθησης. Στη συνέχεια ενεργοποιεί το αντικείμενο της κλάσης UnregisteredLessonsActivity ώστε να προβληθούν τα μαθήματα που βρέθηκαν.

Η μέθοδος **registeredLessonsButton_setOnClickListener** δημιουργεί ένα αντικείμενο τύπου Listener και ενεργοποιείται όταν ο χρήστης επιλέξει το κουμπί registeredLessonsButton. Κατά την ενεργοποίηση η μέθοδος χρησιμοποιεί την κλάση findLessonSectionsByViewHttpClientTask για να βρεθούν τα μαθήματα στα οποία ο φοιτητής έχει δηλώσει ομάδα παρακολούθησης. Στη

συνέχεια ενεργοποιεί το αντικείμενο της κλάσης `RegisteredLessonsActivity` ώστε να προβληθούν τα μαθήματα με τις ομάδες παρακολούθησης που βρέθηκαν.

StudentLoginHttpClientTask
+doInBackground()

Η κλάση **StudentLoginHttpClientTask** πραγματοποιεί δυο κλήσεις REST στο Mobile Service. Η πρώτη κλήση καλεί τη rest μέθοδο `StudentLogin` και γίνεται προσπάθεια αυθεντικοποίησης των στοιχείων του φοιτητή. Η δεύτερη κλήση (που πραγματοποιείται μετα από επιτυχημένη αυθεντικοποίηση του φοιτητή) καλεί τη rest μέθοδο `findLastAcadPeriod` ώστε να βρεθεί η τελευταία ενεργή ακαδημαϊκή περίοδος του φοιτητή.

findRegisteredLessonsWithoutLSectionHttpClientTask
+doInBackground()

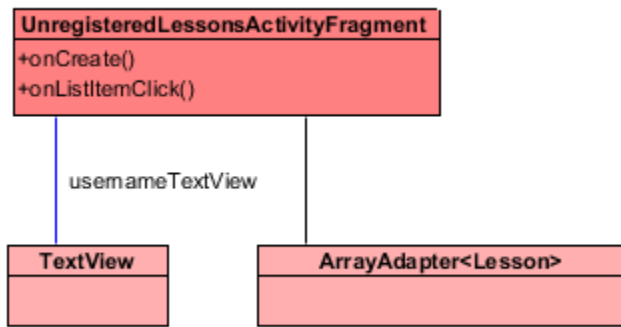
Η κλάση **findRegisteredLessonsWithoutLSectionHttpClientTask** πραγματοποιεί μια κλήση REST στο Mobile Service. Καλείται η rest μέθοδος `findRegisteredLessonsWithoutLSection` και επιστρέφονται τα μαθήματα στα οποία ο συνδεδεμένος φοιτητής δεν έχει δηλώσει ομάδα παρακολούθησης.

findLessonSectionsByViewHttpClientTask
+doInBackground()

Η κλάση **findLessonSectionsByViewHttpClientTask** πραγματοποιεί μια κλήση REST στο Mobile Service. Καλείται η rest μέθοδος `findLessonSectionsByView` και επιστρέφονται τα μαθήματα με τις ομάδες παρακολούθησης όπου ο συνδεδεμένος φοιτητής έχει δηλώσει ομάδα παρακολούθησης.

UnregisteredLessonsActivity

Η κλάση **UnregisteredLessonsActivity** είναι η κλάση εκκίνησης της οθόνης διεπαφής (`UnregisteredLessonsActivityFragment`) που προβάλλει τα μαθήματα στα οποία ο φοιτητής δεν έχει δηλώσει ομάδα παρακολούθησης.

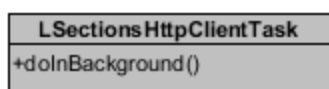


Σχήμα A.21: Διάγραμμα κλάσεων οθόνης διεπαφής μαθημάτων χωρίς ομάδα παρακολούθησης

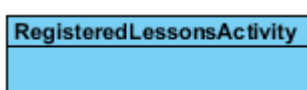
Η κλάση **UnregisteredLessonsActivityFragment** είναι η οθόνη διεπαφής που προβάλλει τα μαθήματα στα οποία ο φοιτητής δεν έχει δηλώσει ομάδα παρακολούθησης. Περιλαμβάνει τα οπτικά αντικείμενα : όνομα του λογαριασμού(**TextView**) και μια λίστα με τα μαθήματα (αντικείμενο τύπου **ArrayAdapter<Lesson>**).

Η μέθοδος **onCreate** δημιουργεί τα δυο οπτικά αντικείμενα και τους θέτει αρχικές τιμές και διαχειριστές συμβάντων. Στο οπτικό αντικείμενο τύπου **ArrayAdapter<Lesson>** τοποθετεί τα μαθήματα από το αντικείμενο τύπου **UnregisteredLessons**.

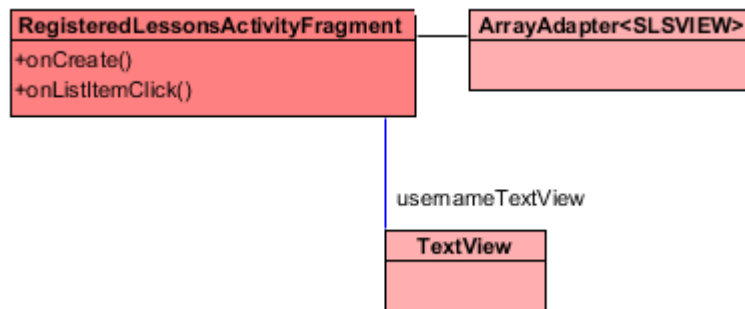
Η μέθοδος **onListItemClick** εκτελείται αυτόματα όταν ο χρήστης επιλέξει κάποιο από τα μαθήματα. Κατά την ενεργοποίηση η μέθοδος χρησιμοποιεί την κλάση **LSectionsHttpClientTask** για να βρεθούν οι ομάδες παρακολούθησης του μαθήματος. Στη συνέχεια για να μπορέσει να δηλώσει ο φοιτητής ομάδα παρακολούθησης η μέθοδος εκκινεί αντικείμενο της κλάσης **LSectionsActivity**.



Η κλάση **LSectionsHttpClientTask** πραγματοποιεί μια κλήση REST στο Mobile Service. Καλείται η rest μέθοδος **findLSectionsByAcadPeriod** και έτσι βρίσκονται οι ομάδες παρακολούθησης του επιλεγμένου μαθήματος.



Η κλάση **RegisteredLessonsActivity** είναι η κλάση εκκίνησης της οθόνης διεπαφής (RegisteredLessonsActivityFragment) που προβάλλει τα μαθήματα με τις ομάδες παρακολούθησης όπου ο συνδεδεμένος φοιτητής έχει δηλώσει ομάδα παρακολούθησης.

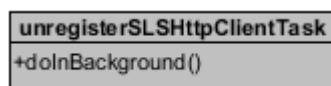


Σχήμα A.22: Διάγραμμα κλάσεων οθόνης διεπαφής μαθημάτων με δηλωμένη ομάδα παρακολούθησης

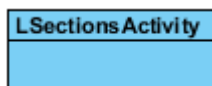
Η κλάση **RegisteredLessonsActivityFragment** είναι η οθόνη διεπαφής που προβάλλει μαθήματα με τις ομάδες παρακολούθησης όπου ο συνδεδεμένος φοιτητής έχει δηλώσει ομάδα παρακολούθησης. Περιλαμβάνει τα οπτικά αντικείμενα : όνομα του λογαριασμού(**TextView**) και μια λίστα με τα μαθήματα (αντικείμενο τύπου **ArrayAdapter<SLSVIEW>**).

Η μέθοδος **onCreate** δημιουργεί τα δυο οπτικά αντικείμενα και τους θέτει αρχικές τιμές και διαχειριστές συμβάντων. Στο οπτικό αντικείμενο τύπου **ArrayAdapter<SLSVIEW>** τοποθετεί τα μαθήματα (με τις δηλωμένες ομάδες παρακολούθησης) από το αντικείμενο τύπου RegisteredLessons.

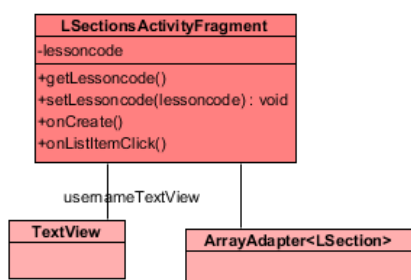
Η μέθοδος **onItemClick** εκτελείται αυτόματα όταν ο χρήστης επιλέξει κάποιο από τα μαθήματα. Κατά την ενεργοποίηση η μέθοδος χρησιμοποιεί την κλάση **unregisterSLSHttpClientTask** ώστε να ακυρωθεί η δήλωση μαθήματος με τη δηλωμένη ομάδα παρακολούθησης. Στη συνέχεια κλείνει η φόρμα διεπαφής και επιστρέφει στην προηγούμενη.



Η κλάση **unregisterSLSHttpClientTask** πραγματοποιεί μια κλήση REST στο Mobile Service. Καλείται η rest μέθοδος unregisterSLS και έτσι ακυρώνεται η ομάδα παρακολούθησης.



Η κλάση **LSectionsActivity** είναι η κλάση εκκίνησης της οθόνης διεπαφής (LSectionsActivityFragment) που προβάλλει τις ομάδες παρακολούθησης ενός μαθήματος ώστε να μπορέσει ο φοιτητής να δηλώσει ομάδα.

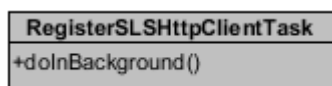


Σχήμα A.23: Διάγραμμα κλάσεων οθόνης διεπαφής ομάδων παρακολούθησης μαθήματος

Η κλάση **LSectionsActivityFragment** είναι η οθόνη διεπαφής που προβάλλει τις ομάδες παρακολούθησης ενός μαθήματος ώστε να μπορέσει ο φοιτητής να δηλώσει ομάδα. Περιλαμβάνει τα οπτικά αντικείμενα : όνομα του λογαριασμού(**TextView**) και μια λίστα με τις ομάδες παρακολούθησης ενός μαθήματος (αντικείμενο τύπου **ArrayAdapter<LSection>**). Περιλαμβάνει επίσης τον κωδικό μαθήματος ως χαρακτηριστικό (**lessoncode**).

Η μέθοδος **onCreate** δημιουργεί τα δυο οπτικά αντικείμενα και τους θέτει αρχικές τιμές και διαχειριστές συμβάντων. Στο οπτικό αντικείμενο τύπου **ArrayAdapter<LSection>** βρίσκει και τοποθετεί τις ομάδες παρακολούθησης από το αντικείμενο τύπου **UnregisteredLessons**.

Η μέθοδος **onListItemClick** εκτελείται αυτόματα όταν ο χρήστης επιλέξει κάποια από τις ομάδες παρακολούθησης του μαθήματος. Κατά την ενεργοποίηση η μέθοδος χρησιμοποιεί την κλάση **RegisterSLSHttpClientTask** για να δηλωθεί η επιλεγμένη ομάδα παρακολούθησης.



Η κλάση **RegisterSLSHttpClientTask** πραγματοποιεί μια κλήση REST στο Mobile Service. Καλείται η rest μέθοδος **registerSLS** και έτσι δηλώνεται η ομάδα παρακολούθησης.

Παράρτημα Β

Υλοποίηση βάσης δεδομένων

Στις επόμενες ενότητες παρουσιάζεται λεπτομερής περιγραφή των κλάσεων κάθε υποσυστήματος.

B.1 Περιγραφή πινάκων

DEPT	
P * NO	NUMBER
NAME	VARCHAR2 (255 BYTE)
SITE	VARCHAR2 (255 BYTE)
LAT	NUMBER (10,7)
LNG	NUMBER (10,7)
SECMAIL	VARCHAR2 (40 BYTE)
DEPT_PK (NO)	

Ο πίνακας τμημάτων (**DEPT**) αποτελείται από τις στήλες κωδικός (NO), όνομα(NAME), διεύθυνση ιστοσελίδας(SITE), γεωγραφικό πλάτος(LAT), γεωγραφικό μήκος(LNG) και τη διεύθυνση ηλεκτρονικής αλληλογραφίας (SECMAIL) της γραμματείας που το εξυπηρετεί.

Κύριο κλειδί του πίνακα είναι η στήλη **NO**.

Δείγμα δεδομένων φαίνεται στον πίνακα B.1

NO	NAME	SITE	LAT	LNG	SECMAIL
328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.	http://iiwm.teikav.edu.gr	40,92993	24,379547	test2@teiemt.gr
332	ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ	http://abd.teikav.edu.gr	40,929282	24,378099	test3@teiemt.gr

Πίνακας B.1: Δείγμα δεδομένων πίνακα

ROOM	
P *	NAME VARCHAR2 (30 BYTE)
PF *	DEPTNO NUMBER
	CAPACITY NUMBER (3)
ROOM_PK (NAME, DEPTNO)	
	(DEPTNO)

Ο πίνακας αιθουσών (**ROOM**) αποτελείται από τις στήλες όνομα (NAME) και διαθέσιμες θέσεις (CAPACITY).

Κύριο κλειδί του πίνακα είναι ο συνδυασμός των στηλών (**NAME, DEPTNO**).

Ξένο κλειδί του πίνακα είναι η στήλη *DEPTNO* που συνδέεται με τη στήλη *DEPT.NO*.

Δείγμα δεδομένων φαίνεται στον πίνακα B.2

NAME	DEPTNO	CAPACITY
A1	328	50
A2	328	30

Πίνακας B.2: Δείγμα δεδομένων πίνακα

TEACHER	
	LNAME VARCHAR2 (60 BYTE)
	FNAME VARCHAR2 (50 BYTE)
	EMAIL VARCHAR2 (40 BYTE)
P *	UNAME VARCHAR2 (15 BYTE)
	PASSWD VARCHAR2 (20 BYTE)
	PHOTO BLOB
F	DEPTNO NUMBER
	ISADMIN NUMBER (1)
	ISCHIEF NUMBER (1)
TEACHER_PK (UNAME)	
	SYS_C0010000 (DEPTNO)

Ο πίνακας εκπαιδευτικών (**TEACHER**) αποτελείται από τις στήλες επώνυμο (LNAME), όνομα (FNANE), διεύθυνση ηλεκτρονικής αλληλογραφίας (EMAIL), όνομα λογαριασμού (UNAME), κωδικό λογαριασμού (PASSWD), σημαία διαχειριστή συστήματος (ISADMIN) και σημαία προϊσταμένου (ISCHIEF).

Κύριο κλειδί του πίνακα είναι η στήλη **UNAME**.

Ξένο κλειδί του πίνακα είναι η στήλη *DEPTNO* που συνδέεται με τη στήλη *DEPT.NO*.

Δείγμα δεδομένων φαίνεται στον πίνακα B.3

LNAME	FNAME	EMAIL	UNAME	PASSWD	PHOTO	DEPTNO	ISADMIN	ISCHIEF
ΤΣΟΥΚΑΛΑΣ	ΒΑΣΙΛΕΙΟΣ	vtsouk@teiemt.gr	vtsouk	123	(null)	0	1	0
ΠΡΟΙΣΤΑΜΕΝΟΣ	111	teacher1@teiemt.gr	teacher1	teacher1	(null)	328	0	1
ΚΑΘΗΓΗΤΗΣ	A	teacher2@teiemt.gr	teacher2	teacher2	(null)	328	0	0
ΚΑΘΗΓΗΤΗΣ	B	teacher3@teiemt.gr	teacher3	teacher3	(null)	328	0	0

Πίνακας B.3: Δείγμα δεδομένων πίνακα

STUDENT	
P * AEM	VARCHAR2 (15 BYTE)
LNAME	VARCHAR2 (60 BYTE)
FNAME	VARCHAR2 (50 BYTE)
EMAIL	VARCHAR2 (40 BYTE)
SEMESTER	VARCHAR2 (8 BYTE)
UNAME	VARCHAR2 (15 BYTE)
PASSWD	VARCHAR2 (60 BYTE)
PF* DEPTNO	NUMBER
STUDENT_PK (DEPTNO, AEM)	
SYS_C0010002 (DEPTNO)	

Ο πίνακας φοιτητών (**STUDENT**) αποτελείται από τις στήλες αριθμό μητρώου τμήματος (AEM), επώνυμο (LNAME), όνομα (FNAME), διεύθυνση ηλεκτρονικής αλληλογραφίας (EMAIL), τρέχον εξάμηνο φοίτησης (SEMESTER), όνομα λογαριασμού (UNAME) και κωδικό λογαριασμού (PASSWD).

Κύριο κλειδί του πίνακα είναι ο συνδυασμός των στηλών **DEPTNO,AEM**.

Ξένο κλειδί του πίνακα είναι η στήλη *DEPTNO* που συνδέεται με τη στήλη *DEPT.NO*.

Δείγμα δεδομένων φαίνεται στον πίνακα B.4

AEM	LNAME	FNAME	EMAIL	SEMESTER	UNAME	PASSWD	DEPTNO
1	ΦΟΙΤΗΤΗΣ 1	ΣΠΟΥΔΑΣΤΗΣ 1	student1@teiemt.gr	1	student1	student1	328
2	ΦΟΙΤΗΤΗΣ 2	ΣΠΟΥΔΑΣΤΗΣ 2	student2@teiemt.gr	2	student2	student2	328
3	ΦΟΙΤΗΤΗΣ 3	ΣΠΟΥΔΑΣΤΗΣ 3	student3@teiemt.gr	3	student3	student3	328
4	ΦΟΙΤΗΤΗΣ 4	ΣΠΟΥΔΑΣΤΗΣ 4	student4@teiemt.gr	4	student4	student4	328
5	ΦΟΙΤΗΤΗΣ 5	ΣΠΟΥΔΑΣΤΗΣ 5	student5@teiemt.gr	5	student5	student5	328
6	ΦΟΙΤΗΤΗΣ 6	ΣΠΟΥΔΑΣΤΗΣ 6	student6@teiemt.gr	6	student6	student6	328
7	ΦΟΙΤΗΤΗΣ 7	ΣΠΟΥΔΑΣΤΗΣ 7	student7@teiemt.gr	7	student7	student7	328
8	ΦΟΙΤΗΤΗΣ 8	ΣΠΟΥΔΑΣΤΗΣ 8	student8@teiemt.gr	8	student8	student8	328
9	ΦΟΙΤΗΤΗΣ 9	ΣΠΟΥΔΑΣΤΗΣ 9	student9@teiemt.gr	9	student9	student9	328
10	ΦΟΙΤΗΤΗΣ 10	ΣΠΟΥΔΑΣΤΗΣ 10	student10@teiemt.gr	10	student10	student10	328

Πίνακας Β.4: Δείγμα δεδομένων πίνακα

LESSON	
P *	CODE VARCHAR2 (20 BYTE)
	TITLE VARCHAR2 (150 BYTE)
F	SEMESTERNO NUMBER
	SECTIONCOUNT NUMBER (2)
	FINALSTATUS CHAR (1 CHAR)
PF *	DEPTNO NUMBER
F	TEACHER_UNAME VARCHAR2 (15 BYTE)
F	ROOM_NAME VARCHAR2 (30 BYTE)
LESSON_PK (DEPTNO, CODE)	
SYS_C0010006 (DEPTNO)	
SYS_C0010007 (SEMESTERNO)	
SYS_C0010008 (TEACHER_UNAME)	
SYS_C0010009 (ROOM_NAME, DEPTNO)	

Ο πίνακας μαθημάτων (**LESSON**) αποτελείται από τις στήλες κωδικό (CODE), τίτλο (TITLE), εξάμηνο διδασκαλίας (SEMESTERNO), πλήθος ομάδων παρακολούθησης (SECTIONCOUNT), σημαία τελικής εξέτασης (FINALSTATUS).

Κύριο κλειδί του πίνακα είναι ο συνδυασμός των στηλών **DEPTNO, CODE**.

Ξένα κλειδιά του πίνακα είναι:

Η στήλη *DEPTNO* που συνδέεται με τη στήλη *DEPT.NO*.

Η στήλη *SEMESTERNO* που συνδέεται με τη στήλη *SEMESTER.NO*.

Η στήλη *TEACHER_UNAME* που συνδέεται με τη στήλη *TEACHER.UNAME*.

Ο συνδυασμός των στηλών *DEPTNO, ROOM_NAME* που συνδέονται με τις στήλες *DEPT.NO ROOM.NAME*.

Δείγμα δεδομένων φαίνεται στον πίνακα Β.5

CODE	TITLE	SEMESTERNO	SECTIONCOUNT	FINALSTATUS	DEPTNO	TEACHER_UNAME	ROOM_NAME
ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ	1	1	T	328	teacher2	A1
ΔΟΚ-Ε	ΔΟΚΙΜΑΣΤΙΚΟ ΕΡΓΑΣΤΗΡΙΟ	1	2	T	328	teacher3	A2

Πίνακας Β.5: Δείγμα δεδομένων πίνακα

LSECTION	
PF*	ACAD_PERIOD_ID NUMBER
PF*	DEPTNO NUMBER
PF*	LESSONCODE VARCHAR2 (20 BYTE)
P *	LSNO VARCHAR2 (50 BYTE)
	STARTTIME NUMBER (2)
	ENDTIME NUMBER (2)
	STARTMIN NUMBER (2)
	ENDMIN NUMBER (2)
	FULLTIME VARCHAR2 (13 BYTE)
	MAXSPACE NUMBER (3)
	BOOKEDSPACE NUMBER (3)
	FREESPACE NUMBER (3)
F	WDNO NUMBER
LSECTION_PK (ACAD_PERIOD_ID, DEPTNO, LESSONCODE, LSNO)	
SYS_C0010014 (ACAD_PERIOD_ID)	
SYS_C0010015 (DEPTNO, LESSONCODE)	
SYS_C0010016 (WDNO)	

Ο πίνακας ομάδων παρακολούθησης (**LSECTION**) αποτελείται από τις στήλες κωδικός ομάδας (LSNO), αριθμός ημέρας έναρξης (WDNO), ώρα έναρξης (STARTTIME), λεπτό έναρξης (STARTMIN), ώρα λήξης (ENDTIME), λεπτό λήξης (ENDMIN), ώρες διεξαγωγής με μορφοποιημένο τρόπο (FULLTIME), μέγιστη χωρητικότητα (MAXSPACE), δεσμευμένες θέσεις (BOOKEDSPACE), ελεύθερες θέσεις (FREESPACE).

Κύριο κλειδί του πίνακα είναι ο συνδυασμός των στηλών **ACAD_PERIOD_ID,DEPTNO,LESSONCODE, LSNO**.

Ένα κλειδιά του πίνακα είναι:

Η στήλη *ACAD_PERIOD_ID* που συνδέεται με τη στήλη *ACAD_PERIOD.ID*.

Ο συνδυασμός των στηλών *DEPTNO,LESSONCODE* που συνδέονται με τις στήλες *DEPT.NO* *LESSON.CODE*.

Η στήλη *WDNO* που συνδέεται με τη στήλη *WEEKDAY.WDNO*.

Δείγμα δεδομένων φαίνεται στον πίνακα Β.6

ACAD_PERIOD_ID	DEPTNO	LESSONCODE	LSNO	STARTTIME	ENDTIME	STARTMIN	ENDMIN	FULLTIME	MAXSPACE	BOOKEDSPACE	FREESPACE	WDNO
32820142	328 ΔΟΚ-Θ	E1		9	10	0	0	9:00-10:00	1	1	0	2
32820142	328 ΔΟΚ-Θ	ΠΛΕΟΝΑΖΟΝΤΕΣ		(null)	(null)	(null)	(null)	:-:	999	0	999	0
32820142	328 ΔΟΚ-Θ	ΤΕΛΙΚΗ ΕΞΕΤΑΣΗ		(null)	(null)	(null)	(null)	:-:	999	0	999	0
32820142	328 ΔΟΚ-Ε	E1		8	10	0	0	8:00-10:00	20	0	20	3
32820142	328 ΔΟΚ-Ε	E2		10	12	0	0	10:00-12:00	20	0	20	3
32820142	328 ΔΟΚ-Ε	ΠΛΕΟΝΑΖΟΝΤΕΣ		(null)	(null)	(null)	(null)	:-:	999	0	999	0
32820142	328 ΔΟΚ-Ε	ΤΕΛΙΚΗ ΕΞΕΤΑΣΗ		(null)	(null)	(null)	(null)	:-:	999	0	999	0

Πίνακας Β.6: Δείγμα δεδομένων πίνακα

SEMESTER	
P * NO	NUMBER
LABEL	VARCHAR2 (8 BYTE)
SEMESTER_PK (NO)	

Ο βοηθητικός πίνακας **SEMESTER** λειτουργεί ως ευρετήριο αντιστοιχιών αριθμού εξαμήνων και λεκτικού εξαμήνων. Αποτελείται από τις στήλες αριθμός εξαμήνου (NO) και λεκτικό εξαμήνου (LABEL).

Κύριο κλειδί του πίνακα είναι η στήλη **NO**.

Δείγμα δεδομένων φαίνεται στον πίνακα Β.7

NO	LABEL
0	-
1	A
2	B
3	Γ
4	Δ
5	E
6	ΣΤ
7	Z

Πίνακας Β.7: Δείγμα δεδομένων πίνακα

WEEKDAY	
P * WDNO	NUMBER
NAME	VARCHAR2 (35 BYTE)
WEEKDAY_PK (WDNO)	

Ο βοηθητικός πίνακας **WEEKDAY** λειτουργεί ως ευρετήριο αντιστοιχιών αριθμού ημέρας της εβδομάδας και λεκτικού ημέρας της εβδομάδας. Αποτελείται από τις στήλες αριθμός ημέρας (WDNO) και λεκτικό ημέρας (NAME).

Κύριο κλειδί του πίνακα είναι η στήλη WDNO.

Δείγμα δεδομένων φαίνεται στον πίνακα B.8

WDNO	NAME
0	-
1	ΔΕΥΤΕΡΑ
2	ΤΡΙΤΗ
3	ΤΕΤΑΡΤΗ
4	ΠΕΜΠΤΗ
5	ΠΑΡΑΣΚΕΥΗ
6	ΣΑΒΒΑΤΟ

Πίνακας B.8: Δείγμα δεδομένων πίνακα

ACAD_PERIOD	
P *	ID NUMBER
UF	DEPTNO NUMBER
U	AYEAR NUMBER (4)
U	TERM VARCHAR2 (15 BYTE)
	REG_START_DATE DATE
	REG_END_DATE DATE
	SEM_CR_START_DATE DATE
	SEM_CR_END_DATE DATE
ACAD_PERIOD_PK (ID)	
ACAD_PERIOD_DEPTNO_AYEAR_TERM_UN (DEPTNO, AYEAR, TERM)	
SYS_C009996 (DEPTNO)	

Ο πίνακας ακαδημαϊκών περιόδων (**ACAD_PERIOD**) τμήματος αποτελείται από τις στήλες κωδικός ακαδημαϊκής περιόδου (ID), ακαδημαϊκό έτος (AYEAR), ακαδημαϊκή περίοδος (TERM), ημερομηνίες έναρξης (REG_START_DATE) – λήξης (REG_END_DATE) δηλώσεων ομάδων παρακολούθησης και οι ημερομηνίες έναρξης (SEM_CR_START_DATE) – λήξης (SEM_CR_END_DATE) κριτηρίου προτεραιότητας εξαμήνου.

Κύριο κλειδί του πίνακα είναι η στήλη **ID**.

Εναλλακτικό υποψήφιο κλειδί του πίνακα είναι ο συνδυασμός των στηλών **DEPTNO,AYEAR,TERM**.

Ξένο κλειδί είναι η στήλη *DEPTNO* που συνδέεται με τη στήλη *DEPT.NO*.

Δείγμα δεδομένων φαίνεται στον πίνακα B.9

ID	DEPTNO	AYEAR	TERM	REG_START_DATE	REG_END_DATE	SEM_CR_START_DATE	SEM_CR_END_DATE
32820131	328	2013	1	(null)	(null)	(null)	(null)
32820132	328	2013	2	(null)	(null)	(null)	(null)
32820141	328	2014	1	(null)	(null)	(null)	(null)
32820142	328	2014	2	12/08/15	15/08/15	13/06/15	13/06/15

Πίνακας B.9: Δείγμα δεδομένων πίνακα

REG_SL	
PF* ACAD_PERIOD_ID	NUMBER
PF* LESSONCODE	VARCHAR2 (20 BYTE)
PF* DEPTNO	NUMBER
PF* STUDENTAEM	VARCHAR2 (15 BYTE)
MARK	NUMBER (3,1)
REG_SL_PK (ACAD_PERIOD_ID, LESSONCODE, DEPTNO, STUDENTAEM)	
SYS_C0010018 (ACAD_PERIOD_ID)	
SYS_C0010019 (DEPTNO, LESSONCODE)	
SYS_C0010020 (DEPTNO, STUDENTAEM)	

Ο πίνακας δηλώσεων μαθημάτων παρακολούθησης (**REG_SL**) αποτελείται από τη στήλη βαθμολογία (MARK) του φοιτητή στο μάθημα.

Κύριο κλειδί του πίνακα είναι ο συνδυασμός των στηλών **ACAD_PERIOD_ID**, **DEPTNO**, **LESSONCODE**, **STUDENTAEM**.

Ξένα κλειδιά του πίνακα είναι τα εξής:

Η στήλη *ACAD_PERIOD_ID* που συνδέεται με τη στήλη *ACAD_PERIOD.ID*.

Ο συνδυασμός των στηλών *DEPTNO,LESSONCODE* που συνδέονται με τις στήλες *DEPT.NO* *LESSON.CODE*.

Ο συνδυασμός των στηλών *DEPTNO,STUDENTAEM* που συνδέονται με τις στήλες *DEPT.NO* *STUDENT.AEM*.

Δείγμα δεδομένων φαίνεται στον πίνακα B.10

ACAD_PERIOD_ID	LESSONCODE	DEPTNO	STUDENTAEM	MARK
32820141	ΔΟΚ-Θ	328	1	7
32820141	ΔΟΚ-Θ	328	2	8
32820141	ΔΟΚ-Θ	328	3	3
32820141	ΔΟΚ-Θ	328	4	0
32820142	ΔΟΚ-Θ	328	1	0
32820142	ΔΟΚ-Θ	328	2	0
32820142	ΔΟΚ-Θ	328	3	0
32820142	ΔΟΚ-Θ	328	4	0
32820142	ΔΟΚ-Ε	328	5	0
32820142	ΔΟΚ-Ε	328	6	0
32820142	ΔΟΚ-Ε	328	7	0
32820142	ΔΟΚ-Ε	328	8	0
32820142	ΔΟΚ-Ε	328	9	0
32820142	ΔΟΚ-Ε	328	10	0

Πίνακας Β.10: Δείγμα δεδομένων πίνακα

REG_SLS	
PF*	ACAD_PERIOD_ID NUMBER
PF*	DEPTNO NUMBER
PF*	STUDENTAEM VARCHAR2 (15 BYTE)
PF*	LESSONCODE VARCHAR2 (20 BYTE)
F	LSNO VARCHAR2 (50 BYTE)
REG_SLS_PK (ACAD_PERIOD_ID, DEPTNO, LESSONCODE, STUDENTAEM)	
SYS_C0010022 (ACAD_PERIOD_ID)	
SYS_C0010023 (DEPTNO, LESSONCODE)	
SYS_C0010024 (DEPTNO, STUDENTAEM)	
SYS_C0010025 (ACAD_PERIOD_ID, DEPTNO, LESSONCODE, LSNO)	

Πίνακας δηλώσεων ομάδων παρακολούθησης (**REG_SLS**).

Κύριο κλειδί του πίνακα είναι ο συνδυασμός των στηλών **ACAD_PERIOD_ID**, **DEPTNO**, **LESSONCODE**, **STUDENTAEM**.

Ξένα κλειδιά του πίνακα είναι τα εξής:

Η στήλη *ACAD_PERIOD_ID* που συνδέεται με τη στήλη *ACAD_PERIOD.ID*.

Ο συνδυασμός των στηλών *DEPTNO*, *LESSONCODE* που συνδέονται με τις στήλες *DEPT.NO* και *LESSON.CODE*.

Ο συνδυασμός των στηλών *DEPTNO*, *STUDENTAEM* που συνδέονται με τις στήλες *DEPT.NO* *STUDENT.AEM*.

Ο συνδυασμός των στηλών *ACAD_PERIOD_ID, DEPTNO, LESSONCODE, LSNO* που συνδέονται με τις στήλες *ACAD_PERIOD.ID, DEPT.NO, LESSON.CODE, LSECTION.LSNO*.

Δείγμα δεδομένων φαίνεται στον πίνακα B.11

ACAD_PERIOD_ID	DEPTNO	STUDENTAEEM	LESSONCODE	LSNO
32820142	328 2		ΔΟΚ-Θ	E1
32820142	328 5		ΔΟΚ-E	E2
32820142	328 1		ΔΟΚ-Θ	E1

Πίνακας B.11: Δείγμα δεδομένων πίνακα

REG_SLS_LOG	
DEPTNO	NUMBER
STUDENTAEEM	VARCHAR2 (15 BYTE)
LESSONCODE	VARCHAR2 (20 BYTE)
LSNO	VARCHAR2 (50 BYTE)
ACTIONDATE	DATE
ACTIONREASON	VARCHAR2 (255 BYTE)

Ο πίνακας αρχείου καταγραφής δηλώσεων ομάδων παρακολούθησης (**REG_SLS_LOG**) αποτελείται από τις στήλες κωδικός τμήματος (DEPTNO), μητρώο φοιτητή (STUDENTAEEM), κωδικός μαθήματος (LESSONCODE), κωδικός ομάδας παρακολούθησης (LSNO), ημερομηνία ενέργειας δήλωσης (ACTIONDATE), είδος ενέργειας (ACTIONREASON).

Δείγμα δεδομένων φαίνεται στον πίνακα B.12

DEPTNO	STUDENTAEEM	LESSONCODE	LSNO	ACTIONDATE	ACTIONREASON
328 1		ΔΟΚ-Θ	ΠΛΕΟΝΑΖΟΝΤΕΣ	12/08/15	INSERT
328 5		ΔΟΚ-E	E2	18/08/15	INSERT
328 1		ΔΟΚ-Θ	E1	09/08/15	INSERT
328 1		ΔΟΚ-Θ	E1	10/06/15	DELETE

Πίνακας B.12: Δείγμα δεδομένων πίνακα

B.2 Περιγραφή όψεων (views)

Όψη TLSVIEW

Η όψη TLSVIEW προβάλλει τους εκπαιδευτικούς με τα μαθήματά τους και τις ομάδες παρακολούθησης. Τα χαρακτηριστικά της όψης είναι: όνομα (**FNAME**), επώνυμο (**LNAME**), κωδικός τμήματος (**DEPTNO**), όνομα λογαριασμού (**UNAME**), κωδικός μαθήματος (**LESSONCODE**), τίτλος (**TITLE**), αίθουσα (**ROOM_NAME**), κωδικός ακαδημαϊκής περιόδου (**ACAD_PERIOD_ID**) και κωδικός ομάδας παρακολούθησης μαθήματος (**LSNO**).

Δείγμα δεδομένων της όψης φαίνεται στον πίνακα B.13

LNAME	FNAME	UNAME	DEPTNO	LESSONCODE	TITLE	ROOM_NAME	ACAD_PERIOD_ID	LSNO
ΚΑΘΗΓΗΤΗΣ Α	teacher2	teacher2	328	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ	A1	32820142 E1	
ΚΑΘΗΓΗΤΗΣ Α	teacher2	teacher2	328	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ	A1	32820142 ΠΛΕΟΝΑΖΟΝΤΕΣ	
ΚΑΘΗΓΗΤΗΣ Α	teacher2	teacher2	328	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ	A1	32820142 ΤΕΛΙΚΗ ΕΞΕΤΑΣΗ	
ΚΑΘΗΓΗΤΗΣ Β	teacher3	teacher3	328	ΔΟΚ-Ε	ΔΟΚΙΜΑΣΤΙΚΟ ΕΡΓΑΣΤΗΡΙΟ	A2	32820142 E1	
ΚΑΘΗΓΗΤΗΣ Β	teacher3	teacher3	328	ΔΟΚ-Ε	ΔΟΚΙΜΑΣΤΙΚΟ ΕΡΓΑΣΤΗΡΙΟ	A2	32820142 E2	
ΚΑΘΗΓΗΤΗΣ Β	teacher3	teacher3	328	ΔΟΚ-Ε	ΔΟΚΙΜΑΣΤΙΚΟ ΕΡΓΑΣΤΗΡΙΟ	A2	32820142 ΠΛΕΟΝΑΖΟΝΤΕΣ	
ΚΑΘΗΓΗΤΗΣ Β	teacher3	teacher3	328	ΔΟΚ-Ε	ΔΟΚΙΜΑΣΤΙΚΟ ΕΡΓΑΣΤΗΡΙΟ	A2	32820142 ΤΕΛΙΚΗ ΕΞΕΤΑΣΗ	

Πίνακας B.13: Δείγμα δεδομένων όψης

Όψη SLVIEW

Η όψη SLVIEW προβάλλει λεπτομέρειες των δηλώσεων μαθημάτων που έχουν κάνει οι φοιτητές στο γραμματειακό σύστημα. Τα χαρακτηριστικά της όψης είναι: κωδικός ακαδημαϊκής περιόδου (**ACADID**), έτος ακαδημαϊκής περιόδου (**AYEAR**), ακαδημαϊκή περίοδος (**TERM**), κωδικός τμήματος (**DNO**), όνομα τμήματος (**DNAME**), αριθμός μητρώου φοιτητή (**AEM**), επώνυμο (**LNAME**), όνομα (**FNAME**), εξάμηνο φοιτητή (**SEMESTER**), διεύθυνση ηλεκτρονικής αλληλογραφίας (**EMAIL**), κωδικός μαθήματος (**CODE**), τίτλος (**TITLE**), αριθμός εξαμήνου (**SEMESTERNO**), λεκτικό εξαμήνου (**LABEL**).

Δείγμα δεδομένων της όψης φαίνεται στον πίνακα B.14

ACADID	AYEAR	TERM	DNO	DNAME	AEM	LNAME	FNAME	SEMESTER	EMAIL	CODE	TITLE	SEMESTERNO	LABEL
32820141	2014 1		328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛ...	4	ΦΟΙΤΗΤΗΣ 4	ΣΠΟΥΔΑΣΤΗΣ 4	4	student4@teiemt.gr	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ		1 A
32820141	2014 1		328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛ...	3	ΦΟΙΤΗΤΗΣ 3	ΣΠΟΥΔΑΣΤΗΣ 3	3	student3@teiemt.gr	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ		1 A
32820141	2014 1		328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛ...	2	ΦΟΙΤΗΤΗΣ 2	ΣΠΟΥΔΑΣΤΗΣ 2	2	student2@teiemt.gr	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ		1 A
32820141	2014 1		328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛ...	1	ΦΟΙΤΗΤΗΣ 1	ΣΠΟΥΔΑΣΤΗΣ 1	1	student1@teiemt.gr	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ		1 A
32820142	2014 2		328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛ...	4	ΦΟΙΤΗΤΗΣ 4	ΣΠΟΥΔΑΣΤΗΣ 4	4	student4@teiemt.gr	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ		1 A
32820142	2014 2		328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛ...	3	ΦΟΙΤΗΤΗΣ 3	ΣΠΟΥΔΑΣΤΗΣ 3	3	student3@teiemt.gr	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ		1 A
32820142	2014 2		328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛ...	2	ΦΟΙΤΗΤΗΣ 2	ΣΠΟΥΔΑΣΤΗΣ 2	2	student2@teiemt.gr	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ		1 A
32820142	2014 2		328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛ...	1	ΦΟΙΤΗΤΗΣ 1	ΣΠΟΥΔΑΣΤΗΣ 1	1	student1@teiemt.gr	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ		1 A
32820142	2014 2		328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛ...	9	ΦΟΙΤΗΤΗΣ 9	ΣΠΟΥΔΑΣΤΗΣ 9	9	student9@teiemt.gr	ΔΟΚ-Ε	ΔΟΚΙΜΑΣΤΙΚΟ ΕΡΓΑΣΤΗΡΙΟ		1 A
32820142	2014 2		328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛ...	8	ΦΟΙΤΗΤΗΣ 8	ΣΠΟΥΔΑΣΤΗΣ 8	8	student8@teiemt.gr	ΔΟΚ-Ε	ΔΟΚΙΜΑΣΤΙΚΟ ΕΡΓΑΣΤΗΡΙΟ		1 A
32820142	2014 2		328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛ...	7	ΦΟΙΤΗΤΗΣ 7	ΣΠΟΥΔΑΣΤΗΣ 7	7	student7@teiemt.gr	ΔΟΚ-Ε	ΔΟΚΙΜΑΣΤΙΚΟ ΕΡΓΑΣΤΗΡΙΟ		1 A
32820142	2014 2		328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛ...	6	ΦΟΙΤΗΤΗΣ 6	ΣΠΟΥΔΑΣΤΗΣ 6	6	student6@teiemt.gr	ΔΟΚ-Ε	ΔΟΚΙΜΑΣΤΙΚΟ ΕΡΓΑΣΤΗΡΙΟ		1 A
32820142	2014 2		328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛ...	5	ΦΟΙΤΗΤΗΣ 5	ΣΠΟΥΔΑΣΤΗΣ 5	5	student5@teiemt.gr	ΔΟΚ-Ε	ΔΟΚΙΜΑΣΤΙΚΟ ΕΡΓΑΣΤΗΡΙΟ		1 A
32820142	2014 2		328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛ...	10	ΦΟΙΤΗΤΗΣ 10	ΣΠΟΥΔΑΣΤΗΣ 10	10	student10@teiemt.gr	ΔΟΚ-Ε	ΔΟΚΙΜΑΣΤΙΚΟ ΕΡΓΑΣΤΗΡΙΟ		1 A

Πίνακας B.14: Δείγμα δεδομένων όψης

Όψη SLSVIEW

Η όψη SLSVIEW προβάλλει λεπτομέρειες των δηλώσεων ομάδων μαθημάτων που έχουν κάνει οι φοιτητές στο σύστημα υποστήριξης προγραμμάτων σπουδών. Τα χαρακτηριστικά της όψης είναι: κωδικός τμήματος (**DNO**), όνομα τμήματος (**DNAME**), αριθμός μητρώου φοιτητή (**AEM**), επώνυμο (**LNAME**), όνομα (**FNAME**), εξάμηνο φοιτητή (**SEMESTER**), κωδικός μαθήματος (**CODE**), τίτλος (**TITLE**), εξάμηνο μαθήματος (**SEMESTERNO**), κωδικός ομάδας παρακολούθησης μαθήματος (**LSNO**), ώρες διεξαγωγής ομάδας μαθήματος (**STARTTIME**, **ENDTIME**, **FULLTIME**, **WDNAME**)

Δείγμα δεδομένων της όψης φαίνεται στον πίνακα B.15

ACADID	AYEAR	TERM	DNO	DNAME	AEM	LNAME	FNAME	SEMESTER	CODE	TITLE	SEMESTERNO	LSNO	STARTTIME	ENDTIME	FULLTIME	WDNAME	LABEL
32820142	2014 2		328	ΤΜΗΜΑ ΜΗΧ...	2	ΦΟΙΤΗΤΗΣ 2	ΣΠΟΥΔΑΣΤΗΣ 2	2	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕ...		1 E1	9	10 9:00-10:00		ΤΡΙΤΗ	A
32820142	2014 2		328	ΤΜΗΜΑ ΜΗΧ...	1	ΦΟΙΤΗΤΗΣ 1	ΣΠΟΥΔΑΣΤΗΣ 1	1	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕ...		1 E1	9	10 9:00-10:00		ΤΡΙΤΗ	A
32820142	2014 2		328	ΤΜΗΜΑ ΜΗΧ...	5	ΦΟΙΤΗΤΗΣ 5	ΣΠΟΥΔΑΣΤΗΣ 5	5	ΔΟΚ-Ε	ΔΟΚΙΜΑΣΤΙΚΟ ΕΡ...		1 E2	10	12 10:00-12:00		ΤΕΤΑΡΤΗ	A

Πίνακας B.15: Δείγμα δεδομένων όψης

Όψη NOTSLSVIEW

Η όψη NOTSLSVIEW προβάλλει λεπτομέρειες των μαθημάτων που οι φοιτητές έχουν δηλώσει στο σύστημα της ηλεκτρονικής γραμματείας, αλλά εκκρεμεί η δήλωση ομάδων παρακολούθησης μαθημάτων στο σύστημα υποστήριξης προγραμμάτων σπουδών. Τα χαρακτηριστικά της όψης είναι: κωδικός ακαδημαϊκής περιόδου (**ACADID**), έτος ακαδημαϊκής περιόδου (**AYEAR**), ακαδημαϊκή περίοδος (**TERM**), κωδικός τμήματος (**DNO**), όνομα τμήματος (**DNAME**), αριθμός μητρώου φοιτητή (**AEM**), επώνυμο (**LNAME**), όνομα (**FNAME**), εξάμηνο φοιτητή (**SEMESTER**), διεύθυνση ηλεκτρονικής αλληλογραφίας (**EMAIL**), κωδικός μαθήματος (**CODE**), τίτλος (**TITLE**), αριθμός εξαμήνου (**SEMESTERNO**), λεκτικό εξαμήνου (**LABEL**).

Δείγμα δεδομένων της όψης φαίνεται στον πίνακα Β.16

ACADID	AYEAR	TERM	DNO	DNAME	AEM	LNAME	FNAME	SEMESTER	EMAIL	CODE	TITLE	SEMESTERNO	LABEL
2820142	2014	2	328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.	6	ΦΟΙΤΗΤΗΣ	6	6	student6@teiemt.gr	ΔΟΚ-Ε	ΔΟΚΙΜΑΣΤΙΚΟ ΕΡΓΑΣΤΗΡΙΟ		1 Α
2820141	2014	1	328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.	4	ΦΟΙΤΗΤΗΣ	4	4	student4@teiemt.gr	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ		1 Α
2820142	2014	2	328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.	3	ΦΟΙΤΗΤΗΣ	3	3	student3@teiemt.gr	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ		1 Α
2820142	2014	2	328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.	10	ΦΟΙΤΗΤΗΣ	10	10	student10@teiemt.gr	ΔΟΚ-Ε	ΔΟΚΙΜΑΣΤΙΚΟ ΕΡΓΑΣΤΗΡΙΟ		1 Α
2820142	2014	2	328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.	9	ΦΟΙΤΗΤΗΣ	9	9	student9@teiemt.gr	ΔΟΚ-Ε	ΔΟΚΙΜΑΣΤΙΚΟ ΕΡΓΑΣΤΗΡΙΟ		1 Α
2820141	2014	1	328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.	1	ΦΟΙΤΗΤΗΣ	1	1	student1@teiemt.gr	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ		1 Α
2820141	2014	1	328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.	3	ΦΟΙΤΗΤΗΣ	3	3	student3@teiemt.gr	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ		1 Α
2820142	2014	2	328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.	4	ΦΟΙΤΗΤΗΣ	4	4	student4@teiemt.gr	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ		1 Α
2820141	2014	1	328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.	2	ΦΟΙΤΗΤΗΣ	2	2	student2@teiemt.gr	ΔΟΚ-Θ	ΔΟΚΙΜΑΣΤΙΚΟ ΘΕΩΡΙΑ		1 Α
2820142	2014	2	328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.	8	ΦΟΙΤΗΤΗΣ	8	8	student8@teiemt.gr	ΔΟΚ-Ε	ΔΟΚΙΜΑΣΤΙΚΟ ΕΡΓΑΣΤΗΡΙΟ		1 Α
2820142	2014	2	328	ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.	7	ΦΟΙΤΗΤΗΣ	7	7	student7@teiemt.gr	ΔΟΚ-Ε	ΔΟΚΙΜΑΣΤΙΚΟ ΕΡΓΑΣΤΗΡΙΟ		1 Α

Πίνακας Β.16: Δείγμα δεδομένων όψης

Όψη LESSONSTATVIEW

Η όψη LESSONSTATVIEW προβάλλει στατιστικά επιτυχιών, αποτυχιών και μη συμμετεχόντων φοιτητών ανά μάθημα και ανά ακαδημαϊκή περίοδο. Τα χαρακτηριστικά της όψης είναι κωδικός ακαδημαϊκής περιόδου (**ACAD_PERIOD_ID**), κωδικός μαθήματος (**LESSONCODE**), είδος στατιστικής ομάδας (**STATUS**) και πλήθος φοιτητών ανά στατιστική ομάδα (**C**).

Δείγμα δεδομένων της όψης φαίνεται στον πίνακα Β.17

ACAD_PERIOD_ID	LESSONCODE	STATUS	C
32820141	ΔΟΚ-Θ	FAILED	1
32820142	ΔΟΚ-Θ	MISSING	4
32820141	ΔΟΚ-Θ	MISSING	1
32820141	ΔΟΚ-Θ	SUCCEED	2
32820142	ΔΟΚ-Ε	MISSING	6

Πίνακας Β.17: Δείγμα δεδομένων όψης

B.3 Περιγραφή Αποθηκευμένων Διαδικασιών, Συναρτήσεων και Σκανδαλισμών (Stored Procedures, Functions and Triggers)

Αποθηκευμένη Διαδικασία DEL_LESSON

Η αποθηκευμένη διαδικασία DEL_LESSON(DEPTNO_P IN NUMBER, CODE_P IN VARCHAR2) δέχεται ως παραμέτρους τον κωδικό τμήματος (**DEPTNO_P**) και τον κωδικό μαθήματος (**CODE_P**) και διαγράφει τις ομάδες παρακολούθησης και το μάθημα της παραμέτρου.

Αποθηκευμένη Συνάρτηση CHECKLSECTIONAVAIL

Η αποθηκευμένη συνάρτηση CHECKLSECTIONAVAIL(**ACAD_PERIOD_ID_P** IN ACAD_PERIOD.ID%TYPE, **DEPTNO_P** IN LSECTION.DEPTNO%TYPE, **LESSONCODE_P** IN LSECTION.LESSONCODE%TYPE, **LSNO_P** IN LSECTION.LSNO%TYPE) δέχεται ως παραμέτρους τον κωδικό της ακαδημαϊκής περιόδου (**ACAD_PERIOD_ID_P**), τον κωδικό τμήματος (**DEPTNO_P**), τον κωδικό μαθήματος (**LESSONCODE_P**) και τον κωδικό της ομάδας παρακολούθησης (**LSNO_P**) και επιστρέφει τιμή **TRUE** αν είναι διαθέσιμη ή τιμή **FALSE** αν η ομάδα παρακολούθησης είναι γεμάτη.

Αποθηκευμένη Συνάρτηση CHECKPLEONAZONTES

Η αποθηκευμένη συνάρτηση CHECKPLEONAZONTES(**ACAD_PERIOD_ID_P** IN ACAD_PERIOD.ID%TYPE, **DEPTNO_P** IN LSECTION.DEPTNO%TYPE, **LESSONCODE_P** IN LSECTION.LESSONCODE%TYPE) δέχεται ως παραμέτρους τον κωδικό της ακαδημαϊκής περιόδου (**ACAD_PERIOD_ID_P**), τον κωδικό τμήματος (**DEPTNO_P**) και τον κωδικό μαθήματος (**LESSONCODE_P**) και επιστρέφει τιμή **TRUE** αν όλες οι κανονικές ομάδες παρακολούθησης έχουν γεμίσει, αλλιώς επιστρέφει τιμή **FALSE**.

Αποθηκευμένη Συνάρτηση GET_LAST_ACAD_ID

Η αποθηκευμένη συνάρτηση GET_LAST_ACAD_ID(**DEPTNO_P** IN DEPT.NO%TYPE) δέχεται ως παράμετρο τον κωδικό τμήματος (**DEPTNO_P**) και επιστρέφει τον κωδικό της τελευταίας ακαδημαϊκής περιόδου του.

Σκανδαλισμοί(Triggers) **UPPER_DEPT, UPPER_TEACHER, UPPER_STUDENT**

Οι σκανδαλισμοί UPPER_DEPT, UPPER_TEACHER, UPPER_STUDENT εξασφαλίζουν πως κατά την εισαγωγή ενός τμήματος ή εκπαιδευτικού ή φοιτητή στη βάση δεδομένων του συστήματος υποστήριξης προγραμμάτων σπουδών τα ονόματά τους θα αποθηκευτούν με κεφαλαία γράμματα.

Σκανδαλισμός(Trigger) **INS_LESSON**

Ο σκανδαλισμός INS_LESSON εξασφαλίζει πως κατά την εισαγωγή ενός νέου μαθήματος στη βάση θα προστεθούν αυτόματα οι ειδικές ομάδες παρακολούθησης ΠΛΕΟΝΑΖΟΝΤΕΣ και ΤΕΛΙΚΗ ΕΞΕΤΑΣΗ.

Σκανδαλισμός(Trigger) **INS_DEL_LSECTION**

Ο σκανδαλισμός INS_DEL_LSECTION εξασφαλίζει πως κατά την εισαγωγή ή τη διαγραφή μιας ομάδας παρακολούθησης ενός μαθήματος η στήλη SECTIONCOUNT του πίνακα LESSON θα ενημερώνεται ανάλογα ώστε να διατηρείται η ακεραιότητα της βάσης. Ο σκανδαλισμός εξασφαλίζει επίσης πως κατά την εισαγωγή μιας ομάδας παρακολούθησης θα ενημερώνονται οι υπολογιζόμενες στήλες (calculated fields) FULLTIME, FREESPACE και BOOKEDSPACE του πίνακα LSECTION.

Σκανδαλισμός(Trigger) **UPDATE_LSECTION**

Ο σκανδαλισμός UPDATE_LSECTION εξασφαλίζει πως κατά την ενημέρωση των στοιχείων μιας ομάδας παρακολούθησης θα ενημερώνονται οι υπολογιζόμενες στήλες (calculated fields) FULLTIME και FREESPACE του πίνακα LSECTION.

Σκανδαλισμός(Trigger) **INS_DEL_REG_SLS**

Ο σκανδαλισμός INS_DEL_REG_SLS εξασφαλίζει πως κατά την εισαγωγή ή τη διαγραφή μιας δήλωσης ομάδας παρακολούθησης ενός μαθήματος θα ελέγχεται η διαθεσιμότητα της ομάδας που επιχειρείται να δηλωθεί (καλώντας τη συνάρτηση CHECKLSECTIONAVAIL) και αποτρέπεται η δήλωση αν δεν υπάρχει διαθεσιμότητα. Αν η δήλωση επιτρέπεται ή αν πρόκειται για διαγραφή δήλωσης από το φοιτητή τότε ενημερώνονται ανάλογα οι στήλες BOOKEDSPACE και FREESPACE του πίνακα LSECTION. Επίσης αν η ομάδα δήλωσης αφορά την ειδική ομάδα

ΠΛΕΟΝΑΖΟΝΤΕΣ ο σκανδαλισμός ελέγχει αν επιτρέπεται ή όχι η δήλωση της (καλώντας τη συνάρτηση CHECKPLEONAZONTES) και αν δεν επιτρέπεται τότε αποτρέπεται η δήλωση.