



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**“Σχεδιασμός και Ανάπτυξη Συστήματος για την Ασφαλή
διεξαγωγή εξετάσεων στο ΠΑΔΑ σε καιρό Πανδημίας Covid-19”**

ΠΑΝΑΓΙΩΤΗΣ ΚΑΛΑΜΑΡΑΣ
A.M. 711161040

Εισηγητής: ΙΩΑΝΝΗΣ ΒΟΓΙΑΤΖΗΣ, ΚΑΘΗΓΗΤΗΣ

(Κενό φύλλο)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΓΙΑ ΤΗΝ ΑΣΦΑΛΗ
ΔΙΕΞΑΓΩΓΗ ΕΞΕΤΑΣΕΩΝ ΣΤΟ ΠΑΔΑ ΣΕ ΚΑΙΡΟ ΠΑΝΔΗΜΙΑΣ COVID-19**

**Παναγιώτης Καλαμάρας
Α.Μ. 711161040**

Εισηγητής:

ΙΩΑΝΝΗΣ ΒΟΓΙΑΤΖΗΣ, ΚΑΘΗΓΗΤΗΣ

Εξεταστική Επιτροπή:

**ΙΩΑΝΝΗΣ ΒΟΓΙΑΤΖΗΣ, ΚΑΘΗΓΗΤΗΣ
ΧΡΗΣΤΟΣ ΤΡΟΥΣΣΑΣ, ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ
ΓΕΩΡΓΙΟΣ ΜΕΛΕΤΙΟΥ, ΕΔΙΠ**

Ημερομηνία εξέτασης 11/07/2023

(Κενό φύλλο)

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Καλαμάρας Παναγιώτης του Στυλιανού, με αριθμό μητρώου 711161040 φοιτητής του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Βεβαιώνω ότι είμαι συγγραφέας αυτής της Διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών
Π. Καλαμάρας

(Κενό φύλλο)

ΕΥΧΑΡΙΣΤΙΕΣ

Αρχικά, θα ήθελα να ευχαριστήσω ιδιαίτερα τον Καθηγητή μου κ. Βογιατζή Ιωάννη για την άψογη συνεργασία αλλά και την άμεση επικοινωνία που είχαμε, καθ' όλη την διάρκεια της εκπόνησης της διπλωματικής μου εργασίας. Επίσης, θεωρώ ιδιαίτερη τιμή μου, την συνεργασία μου με τους Καθηγητές κ. Μελετίου Γεώργιο και κ. Ματιάτο Σπυρίδων, για την αναζήτηση χρήσιμης πληροφορίας σε σχέση με την λειτουργία του τμήματος Μηχανικών Πληροφορικής και Υπολογιστών. Αυτές οι πληροφορίες χρησίμευσαν αρκετά, τόσο κατά το θεωρητικό μέρος της εργασίας μου (ανάλυση και μελέτη των προβλημάτων που προκύπτουν στην καθημερινή λειτουργία του τμήματος εν καιρώ **Covid-19**) καθώς και στο πρακτικό μέρος της (μελέτη των βασικών παραμέτρων του συστήματος «**UNIWA EXAM PLANNER**»). Έπειτα, θα ήθελα να ευχαριστήσω την οικογένεια μου, για όλη αυτήν την αγάπη και υποστήριξη που μου παρείχαν και που συνεχίζουν να μου παρέχουν εννοείται και τέλος τους υπέροχους φίλους μου, οι οποίοι ήταν πάντα εκεί, στα εύκολα και στα δύσκολα αυτής της περιόδου που πέρασε.

(Κενό φύλλο)

ΠΕΡΙΛΗΨΗ

Στην παρούσα Διπλωματική Εργασία θα δούμε την μελέτη, την ανάλυση καθώς και την επιμέρους επίλυση των προβλημάτων, που προκύπτουν κατά την καθημερινή λειτουργία και πιο συγκεκριμένα στις εξεταστικές περιόδους στο Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, κατά την επιστροφή των φοιτητών στη δια ζώσης εκπαίδευση εν καιρώ

Covid-19. Η εργασία αποτελείται από ένα θεωρητικό αλλά και από ένα πρακτικό μέρος με τις ανάλογες μελέτες που πραγματοποιήθηκαν.

Εξαρχής, στο θεωρητικό μέρος της εργασίας, γίνεται η μελέτη των προβλημάτων που αντιμετωπίζει τόσο το τμήμα όσο και οι φοιτητές του στην καθημερινή του λειτουργία και η εξαγωγή χρήσιμης πληροφορίας για προτάσεις φοιτητών, οι οποίες μπορούν να υπάρξουν πιθανές λύσεις αυτών. Έπειτα, μέσα από την εκτενέστερη ανάλυση αυτών των προτάσεων, την μελέτη των λύσεων που προκύπτουν καθώς και την μεταξύ τους σύγκριση, προκύπτει η καλύτερη μέθοδος επίλυσης του προβλήματος.

Στην συνέχεια, στο πρακτικό μέρος της εργασίας, ο αναγνώστης βλέπει την πλήρη και περιεκτική περιγραφή του συστήματος «**UNIWA EXAM PLANNER**», το οποίο και θα αποτελέσει μέρος της τελικής μας λύσης στην κατάσταση. Ακολουθεί η περιγραφή της λειτουργικότητας του συστήματος (η οποία αποτελείται από Web Services) με την βοήθεια του εργαλείου **ADONIS Business Process Management Toolkit**, της βάσης δεδομένων του «**examsdb**» μέσω του περιβάλλοντος **MySQL Workbench** καθώς και η παρουσίαση του τελικού πηγαίου κώδικα σε γλώσσα **Java**, μαζί με παραδείγματα χρήσης και οδηγίες εγκατάστασης των απαραίτητων εργαλείων για την εκτέλεση και την παραγωγή του εκτελέσιμου προγράμματος για τον τελικό χρήστη. Τέλος, οι τεχνολογίες που χρησιμοποιήθηκαν κατά την υλοποίηση της εφαρμογής είναι το **Vaadin Designer** για το κομμάτι του **Frontend** και το **Spring Boot** για την κατασκευή των **Web Services** και του **Backend** τμήματος του συστήματος.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Προγραμματισμός Η/Υ

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: UNIWA EXAM PLANNER, examsdb, Java, MySQL Workbench, Spring Boot, Web Services, Vaadin Designer, Covid-19, ADONIS Business Process Management Toolkit, Frontend, Backend

ABSTRACT

In this Diplomatic Thesis we will see the study, the analysis as well as the partial solution of the problems that arise during the daily operation and more specifically during the examination periods at the Department of Informatics and Computer Engineering of the School of Engineering of the University of Western Attica, upon the return of the students to lifelong learning in time

Covid-19. The work consists of a theoretical and a practical part with the corresponding studies carried out.

From the beginning, in the theoretical part of the work, the problems faced by both the department and its students in its daily operation are studied and the extraction of useful information for student proposals, which can be possible solutions to them. Then, through the more extensive analysis of these proposals, the study of the resulting solutions as well as the comparison between them, the best method of solving the problem emerges.

Then, in the practical part of the paper, the reader sees the complete and comprehensive description of the "**UNIWA EXAM PLANNER**" system, which will be part of our final solution to the situation. The following is the description of the functionality of the system (which consists of **Web Services**) with the help of the **ADONIS Business Process Management Toolkit**, the "**examsdb**" database through the **MySQL Workbench environment** as well as the presentation of the final source code in **Java** language, along with usage examples and installation instructions of the necessary tools to run and produce the executable program for the end user. Finally, the technologies used during the implementation of the application are **Vaadin Designer** for the **Frontend** part and **Spring Boot** for the construction of **Web Services** and the **Backend** part of the system.

SCIENTIFIC AREA: Computer programming

KEYWORDS: UNIWA EXAM PLANNER, examsdb, Java, MySQL Workbench, Spring Boot, Web Services, Vaadin Designer, Covid-19, ADONIS Business Process Management Toolkit, Frontend, Backend

Πίνακας Περιεχομένων Διπλωματικής

ΕΙΣΑΓΩΓΗ	1
Κεφάλαιο 1° - Σημερινή Κατάσταση Λειτουργίας του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών του Πανεπιστημίου Δυτικής Αττικής.	3
1.1 Καθημερινή Λειτουργία του Τμήματος εν καιρό Covid-19.....	3
1.2 Πρώτες Εντυπώσεις των Φοιτητών.....	5
1.3 Συμπεράσματα βάσει της γνώμης των Φοιτητών πάνω στην Σημερινή Κατάσταση. 14	
1.3.1 Ανάλυση των Προβλημάτων που προκύπτουν κατά την Καθημερινή Λειτουργία του Τμήματος.	14
1.3.2 Εξαγωγή Προτάσεων των Φοιτητών ως Πιθανές Λύσεις.....	16
Κεφάλαιο 2° – Ανάλυση των Διαθέσιμων Προτάσεων των Φοιτητών και Εξαγωγή της Καλύτερης Λύσης.	17
2.1 Μελέτη Προτάσεων και Εξαγωγή Βέλτιστων Δυνατών Λύσεων.....	17
2.2 Ανάλυση και Σύγκριση Διαθέσιμων Λύσεων.	20
2.3 Καλύτερη Βέλτιστη Λύση και Συμπεράσματα Μελέτης των Φοιτητών.	21
Κεφάλαιο 3° – Ανάλυση και Σχεδίαση Αρχιτεκτονικής Συστήματος «UNIWA EXAM PLANNER».....	22
3.1 Γενική Περιγραφή Συστήματος.	22
3.2 Διαγραμματική Απεικόνιση της Αρχιτεκτονικής του Συστήματος.	23
3.3 Ανάλυση της Λειτουργίας του Συστήματος σε επιμέρους Διαδικασίες.	28
Κεφάλαιο 4° – Οι Τεχνολογίες Σχεδίασης και Ανάπτυξης του Συστήματος «UNIWA EXAM PLANNER».....	32
4.1 Γενική Περιγραφή των Τεχνολογιών Συστήματος.	32
4.2 Ανάλυση Ορολογίας στις Τεχνολογίες Σχεδίασης και Ανάπτυξης του Συστήματος «UNIWA EXAM PLANNER».	33
Κεφάλαιο 5° – Περιγραφή της Βάσης Δεδομένων MySQL και των Spring Boot Web Services του Συστήματος.....	36
5.1 Διαγραμματική Απεικόνιση Βάσης Δεδομένων MySQL – Διάγραμμα Σχέσεων Οντοτήτων E.R. (Entity Relationship - Diagram).....	36
5.2 Ανάλυση Δομής των Πινάκων της Βάσης Δεδομένων.	37
5.3 Ανάλυση Συστήματος σε Web Services.	44

Κεφάλαιο 6 ^ο – Οδηγός Εγκατάστασης Απαραίτητων Εργαλείων και Χρήσης Συστήματος...	47
6.1 Βοηθητική Εγκατάσταση Εργαλείων και Μέθοδος Παραγωγής Εκτελέσιμου Κώδικα.	47
6.2 Οδηγίες Χρήσεις Συστήματος με Ενδεικτικές Οθόνες ανά Περίπτωση.	49
6.2.1 Περίπτωση Χρήστη Γραμματείας	51
6.2.2 Περίπτωση Χρήστη Φοιτητή.	65
6.2.3 Περίπτωση Χρήστη Καθηγητή.	78
ΤΕΛΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ	80
ΠΑΡΑΡΤΗΜΑΤΑ – ΚΩΔΙΚΑΣ ΕΡΓΑΣΙΑΣ	81
ΠΑΡΑΡΤΗΜΑ Α’	81
Κώδικας από αρχείο «examsdb ver13.0.sql»	81
ΠΑΡΑΡΤΗΜΑ Β’	88
Κώδικας από αρχείο «Application.java»	88
Κώδικας από αρχείο «LoginService.java»	89
Κώδικας από αρχείο «SecretariatModel.java»	90
Κώδικας από αρχείο «SecretariatRepository.java»	91
Κώδικας από αρχείο «SecretariatService.java»	91
Κώδικας από αρχείο «StudentModel.java»	93
Κώδικας από αρχείο «StudentRepository.java»	95
Κώδικας από αρχείο «StudentService.java»	96
Κώδικας από αρχείο «ProfessorModel.java»	108
Κώδικας από αρχείο «ProfessorRepository.java»	110
Κώδικας από αρχείο «ProfessorService.java»	111
Κώδικας από αρχείο «ClassAvailKey.java»	112
Κώδικας από αρχείο «ClassAvailModel.java»	113
Κώδικας από αρχείο «ClassAvailRepository.java»	114
Κώδικας από αρχείο «ClassModel.java»	115
Κώδικας από αρχείο «ClassRepository.java»	116
Κώδικας από αρχείο «ExamModel.java»	117
Κώδικας από αρχείο «ExamRepository.java»	119
Κώδικας από αρχείο «ExamClassKey.java»	119
Κώδικας από αρχείο «ExamClassModel.java»	120
Κώδικας από αρχείο «ExamClassRepository.java»	121
Κώδικας από αρχείο «ExamProfessorsKey.java»	122
Κώδικας από αρχείο «ExamProfessorsModel.java»	123

Κώδικας από αρχείο «ExamProfessorsRepository.java» :	124
Κώδικας από αρχείο «ExamStudentsKey.java» :	124
Κώδικας από αρχείο «ExamStudentsModel.java» :	125
Κώδικας από αρχείο «ExamStudentsRepository.java» :	126
Κώδικας από αρχείο «LessonModel.java» :	127
Κώδικας από αρχείο «LessonRepository.java» :	128
Κώδικας από αρχείο «LessonPerStudentKey.java» :	129
Κώδικας από αρχείο «LessonPerStudentModel.java» :	130
Κώδικας από αρχείο «LessonPerStudentRepository.java» :	131
Κώδικας από αρχείο «TimeslotModel.java» :	132
Κώδικας από αρχείο «TimeslotRepository.java» :	133
Κώδικας από αρχείο «StudentMainView.java» :	133
Κώδικας από αρχείο «ApplyView.java» :	135
Κώδικας από αρχείο «StudentApplicationsView.java» :	139
Κώδικας από αρχείο «SecretariatMainView.java» :	142
Κώδικας από αρχείο «CreateExam.java» :	144
Κώδικας από αρχείο «CreateExamprof.java» :	149
Κώδικας από αρχείο «UpdateExamInfoView.java» :	154
Κώδικας από αρχείο «ExamInfoView.java» :	159
Κώδικας από αρχείο «AllocateExamRoomView.java» :	161
Κώδικας από αρχείο «ProfessorMainView.java» :	165
Κώδικας από αρχείο «ProfessorApplicationsView.java» :	166
Κώδικας από αρχείο «LoginView.java» :	169
Κώδικας από αρχείο «login-view.ts» :	170
Κώδικας από αρχείο «secretariat-main-view.ts»:	171
Κώδικας από αρχείο «create-exam.ts» :	172
Κώδικας από αρχείο «create-examprof.ts» :	173
Κώδικας από αρχείο «allocate-exam-room-view.ts» :	174
Κώδικας από αρχείο «update-exam-info-view.ts» :	175
Κώδικας από αρχείο «exam-info-view.ts» :	176
Κώδικας από αρχείο «student-main-view.ts» :	177
Κώδικας από αρχείο «apply-view.ts» :	178
Κώδικας από αρχείο «student-applications-view.ts» :	178
Κώδικας από αρχείο «professor-main-view.ts» :	179
Κώδικας από αρχείο «professor-applications-view.ts» :	180
Κώδικας από αρχείο «application.properties» :	181

ΕΙΚΟΝΕΣ

Εικόνα 1 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση της Ταχύτητας Ελέγχων Εγγράφων κατά την Είσοδο στο Ίδρυμα.	5
Εικόνα 2 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση της Ακρίβειας Ελέγχων Εγγράφων κατά την Είσοδο στο Ίδρυμα.	5
Εικόνα 3 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση του Ποσοστού ύπαρξης των κατάλληλων Μέτρων προστασίας από τον Covid-19.	6
Εικόνα 4 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση του Ποσοστού εφαρμογής των κατάλληλων Μέτρων προστασίας από τον Covid-19.	6
Εικόνα 5 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση της ποιότητας στην Δια Ζώσης Εκπαίδευση ως προς τις ώρες διδασκαλίας βάση της δυσκολίας των μαθημάτων.	6
Εικόνα 6 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση της ποιότητας στην Δια Ζώσης Εκπαίδευση ως προς τις ώρες διδασκαλίας βάση του όγκου της ύλης των μαθημάτων.	7
Εικόνα 7 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση της ποιότητας στην Δια Ζώσης Εκπαίδευση ως προς την διαθεσιμότητα αιθουσών.	7
Εικόνα 8 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση της ποιότητας στην Δια Ζώσης Εκπαίδευση ως προς την τήρηση των απαραίτητων μέτρων προστασίας κατά του Covid-19.	7
Εικόνα 9 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση της ποιότητας στην Εκπαίδευση ως προς το ποσοστό κινδύνου εμφάνισης κρούσματος και επιστροφής στην τηλεκπαίδευση.	8
Εικόνα 10 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση της ποιότητας στην Δια Ζώσης Εκπαίδευση ως προς τα εργαστηριακά μαθήματα.	8
Εικόνα 11 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση των Μέτρων Προστασίας σε περίπτωση εμφάνισης κρούσματος στους χώρους του Πανεπιστημίου.	8
Εικόνα 12 - Απαντήσεις ως προς το Ερώτημα «Ποιες οδηγίες έχουν δοθεί για τις εξετάσεις;».	9
Εικόνα 13 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση των οδηγιών που έχουν δοθεί για τις εξετάσεις.	9
Εικόνα 14 - Απαντήσεις φοιτητών πάνω στο ερώτημα «Ποια είναι η άποψη σας για τις δια ζώσης διαλέξεις έναντι της τηλεκπαίδευσης;».	10
Εικόνα 15 - Απαντήσεις των φοιτητών ως προς το ερώτημα «Ποιες προτάσεις θα κάνατε, για την βελτίωση της ποιότητας φοίτησης σας στο Πανεπιστήμιο Δυτικής Αττικής την τρέχουσα περίοδο;».	12
Εικόνα 16 - ΣΥΝΔΕΣΗ ΧΡΗΣΤΗ (LOGIN) 1.0 - Business Process Management Diagram.	23
Εικόνα 17 - ΜΕΝΟΥ ΕΠΙΛΟΓΩΝ ΧΡΗΣΤΗ 1.0 - Business Process Management Diagram.	23
Εικόνα 18 - ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ ΕΞΕΤΑΣΤΙΚΗΣ 1.0 - Business Process Management Diagram.	24
Εικόνα 19 - ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ ΜΑΘΗΜΑΤΩΝ ΕΞΕΤΑΣΤΙΚΗΣ 1.0 - Business Process Management Diagram.	24

Εικόνα 20 - ΕΝΗΜΕΡΩΣΗ ΔΕΔΟΜΕΝΩΝ ΕΞΕΤΑΣΤΙΚΗΣ 1.0 - Business Process Management Diagram.	25
Εικόνα 21- ΟΡΙΣΜΟΣ ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ ΑΙΘΟΥΣΩΝ 1.0 - Business Process Management Diagram.	25
Εικόνα 22 - ΕΜΦΑΝΙΣΗ ΥΠΑΡΧΟΥΣΑΣ ΠΛΗΡΟΦΟΡΙΑΣ ΕΞΕΤΑΣΤΙΚΗΣ 1.0 - Business Process Management Diagram.....	26
Εικόνα 23 - ΔΗΛΩΣΗ ΑΙΤΗΣΗΣ ΕΞΕΤΑΣΗΣ ΜΑΘΗΜΑΤΟΣ 1.0 - Business Process Management Diagram.	26
Εικόνα 24 - ΕΜΦΑΝΙΣΗ ΔΙΑΘΕΣΙΜΩΝ ΑΙΤΗΣΕΩΝ ΦΟΙΤΗΤΗ 1.0 - Business Process Management Diagram.	27
Εικόνα 25 - ΕΜΦΑΝΙΣΗ ΔΙΑΘΕΣΙΜΩΝ ΑΙΤΗΣΕΩΝ ΦΟΙΤΗΤΩΝ 1.0 - Business Process Management Diagram.....	27
Εικόνα 26 - ΡΟΛΟΙ ΧΡΗΣΤΩΝ ΣΥΣΤΗΜΑΤΟΣ 1.0 - Working Environment Model Diagram.....	28
Εικόνα 27 - Διάγραμμα E.R.D. (Entity Relationship Diagram) της MySQL Βάσης Δεδομένων examsdb.	36
Εικόνα 28 - Σύνδεσμος Πρόσβασης στο Σύστημα από τον Χρήστη μέσω Browser..	49
Εικόνα 29 - Οθόνη Εισόδου.....	49
Εικόνα 30 - Επιτυχής Σύνδεση στο Σύστημα	50
Εικόνα 31 - Περιπτώσεις αποτυχίας σύνδεσης στο Σύστημα.....	51
Εικόνα 32 - Μενού επιλογών Γραμματείας.....	51
Εικόνα 33 - Φόρμα Καταχώρησης Νέας Εξεταστικής.....	52
Εικόνα 34 - Περίπτωση Λάθους - Άδεια φόρμα κατά την καταχώρηση στοιχείων εξεταστικής.....	53
Εικόνα 35 - Περίπτωση επιτυχούς καταχώρησης νέας Εξεταστικής.....	54
Εικόνα 36 - Περίπτωση Διπλότυπης Εγγραφής.....	54
Εικόνα 37 - Περίπτωση Λάθους Έτους.....	55
Εικόνα 38 - Περίπτωση Λάθους Ημερομηνίας Έναρξης/Λήξης.....	55
Εικόνα 39 - Περίπτωση Λάθους Ημερομηνίας βάση Περιόδου.....	56
Εικόνα 40 - Περίπτωση Λάθους Ημερομηνίας βάση Έτους Εξεταστικής.....	56
Εικόνα 41 - Περίπτωση άδειας φόρμας καταχώρησης μαθημάτων ανά εξεταστική περίοδο.	57
Εικόνα 42 - Περίπτωση Λάθος Έτους - Καταχώρηση Μαθήματος Εξεταστικής.....	58
Εικόνα 43 - Περίπτωση μαθήματος με περίοδο που δεν ανήκει στην επερχόμενη εξεταστική περίοδο.	59
Εικόνα 44 - Η κατηγορία εξεταστικής δεν υπάρχει στο σύστημα.....	59
Εικόνα 45 - Επιτυχής Υποβολή Μαθήματος σε Εξεταστική.....	60
Εικόνα 46 - Ενημέρωση Εξεταστικής.....	62
Εικόνα 47 - Επιλογή Διαθεσιμότητας Αιθουσών.....	63
Εικόνα 48 - Δέσμευση Χρονοδιαγράμματος Αίθουσας.....	63
Εικόνα 49 - Αποδέσμευση Χρονοδιαγράμματος Αίθουσας.....	64
Εικόνα 50 - Εμφάνιση Πληροφοριών Εξεταστικής.....	64
Εικόνα 51 - Μενού Επιλογών Φοιτητή.....	65
Εικόνα 52 - Οθόνη Δήλωσης Μαθήματος.....	65
Εικόνα 53 - Επιτυχής Εγγραφή Μαθήματος (Γραμμική Άλγεβρα).....	66
Εικόνα 54 - Επιτυχής Εγγραφής Μαθήματος (Α&Π).....	67
Εικόνα 55 - Επιτυχής Εγγραφή Σεπτέμβριος 2023 Γραμμική Άλγεβρα 1#.....	68
Εικόνα 56 - Επιτυχής Εγγραφή Σεπτέμβριος 2023 Γραμμική Άλγεβρα 2#.....	69

Εικόνα 57 - Επιτυχής Εγγραφή Εμβόλιμης Εξεταστικής.	70
Εικόνα 58 - Επιτυχής Εγγραφή Εξεταστικής Βελτίωσης Βαθμού.	71
Εικόνα 59 - Περίπτωση Δήλωσης Μαθήματος Εξεταστικής Περιόδου που δεν έχει οριστεί ακόμα.....	72
Εικόνα 60 - Περίπτωση Δήλωσης Μαθήματος από Καθηγητή που δεν έχει οριστεί ακόμα.....	72
Εικόνα 61 - Περίπτωση Δήλωσης Μαθήματος από άλλο Περίοδο (Εξάμηνο Φοιτητή Μεγαλύτερο του Μαθήματος).....	73
Εικόνα 62 - Περίπτωση Δήλωσης Μαθήματος από άλλο Περίοδο (Εξάμηνο Φοιτητή Μικρότερο του Μαθήματος).	73
Εικόνα 63 - Περίπτωση Διπλότυπης Δήλωσης Μαθήματος.	74
Εικόνα 64 - Περίπτωση Αποτυχίας Δήλωσης Μαθήματος σε Εμβόλιμη Εξεταστική (Φοιτητής κάτω του 9ου Εξαμήνου).....	75
Εικόνα 65 - Περίπτωση Αποτυχίας Δήλωσης Μαθήματος σε Εξεταστική Βελτίωσης Βαθμού 1#.	75
Εικόνα 66 - Περίπτωση Αποτυχίας Δήλωσης Μαθήματος σε Εξεταστική Βελτίωσης Βαθμού 2#.	76
Εικόνα 67 - Περίπτωση Ανεπαρκούς Διαθεσιμότητας Αιθουσών.....	76
Εικόνα 68 - Οθόνη Προβολής Αιτήσεων Φοιτητή.	77
Εικόνα 69 - Διαγραφή Δήλωσης Μαθήματος Φοιτητή.	77
Εικόνα 70 - Μενού Επιλογών Καθηγητή.	78
Εικόνα 71 - Εμφάνιση Αιτήσεων Φοιτητών ανά Εξετάσεων Καθηγητή.	79

ΠΙΝΑΚΕΣ

Πίνακας 1 - Πληροφορίες Διαθέσιμων Αιθουσών Τμήματος.....	4
Πίνακας 2 - Μελέτη Προτάσεων Φοιτητών ως προς τα Πλεονεκτήματα/Μειονεκτήματα τους.	20
Πίνακας 3 - Ανάλυση Συστήματος σε επί μέρους Διαδικασίες.	30
Πίνακας 4 - Ανάλυση Διαδικασιών σε Υποδιαδικασίες.	31
Πίνακας 5 - Ανάλυση Πινάκων Βάσης Δεδομένων και των μεταξύ τους Συσχετίσεων.	38
Πίνακας 6 - Ανάλυση των Πεδίων των Πινάκων και της χρησιμότητας της Πληροφορίας τους.....	43
Πίνακας 7 - Ανάλυση των Web Services του Συστήματος.	46

ΕΙΣΑΓΩΓΗ

Οι Έλληνες, έχοντας πλέον ήδη μια δεκαετία προβλημάτων στην πλάτη τους, τόσο οικονομικών αλλά και συνάμα κοινωνικών (βλέπε οικονομική κρίση του 2010) και με την εμφάνιση της κρίσης του **Covid19** τα τελευταία 3 χρόνια, αυτό έχει αποτέλεσμα μια μέτρια καθημερινότητα σε πολλούς τομείς της ζωής όπως είναι και η Παιδεία. Πιο συγκεκριμένα, όσον αφορά την τριτοβάθμια εκπαίδευση, αρκετά ιδρύματα (όπως και το ΠΑ.Δ.Α.) στην χώρα παρουσιάζουν κατά καιρούς μείωση προσωπικού, ακόμα και ολόκληρες συγχωνεύσεις σε τμήματα και αυτό σε συνδυασμό με την συνεχή αύξηση των εισαχθέντων φοιτητών ανά τμήμα, το καθιστά αδύναμο ως προς την εξυπηρέτηση των αναγκών τους. Πέρα από την καθημερινή λειτουργία ενός τμήματος, πρέπει να γίνει αντιληπτό ποια θα είναι ειδικότερα η κλίματα του προβλήματος σε εξεταστικές περιόδους, όπου μια παραπάνω θα πρέπει να καλυφθούν οι ανάγκες των φοιτητών και ακόμα περισσότερο των ευπαθών ομάδων. Και εδώ έρχεται η παρούσα εργασία, όπου ο στόχος της είναι η βελτιστοποίηση της παρούσας κατάστασης, κατά τις εξεταστικές περιόδους του τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της σχολής Μηχανικών στο Πανεπιστήμιο Δυτικής Αττικής.

Αυτή λοιπόν, αποτελείται από ένα θεωρητικό αλλά και ένα πρακτικό μέρος, τα οποία διαιρούνται στην έρευνα, την ανάλυση, την μελέτη, την σχεδίαση και την τελική υλοποίηση της καλύτερης δυνατής λύσης για την βελτιστοποίηση της καθημερινής λειτουργίας τους τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής. Πιο συγκεκριμένα το κομμάτι της λειτουργίας του τμήματος που θα παρουσιαστεί αφορά τις διάφορες εξεταστικές του περιόδους, ως προς την καλύτερη δυνατή αξιοποίηση των διαθέσιμων υποδομών του, σε σχέση με το πλήθος των φοιτητών που έχει εγγεγραμμένους.

Ως πρώτο κομμάτι της δουλειάς λοιπόν, θα είναι η μελέτη της σημερινής καθημερινής λειτουργίας του τμήματος και τα προβλήματα που προκύπτουν, ενώ στην συνέχεια η αναζήτηση και εύρεση ορισμένων προτάσεων προς βελτίωση της παρούσας κατάστασης. Βοηθητικά και για την καλύτερη διεξαγωγή της έρευνας, θα γίνει χρήση μέσω της πλατφόρμας Google του εργαλείου Google Form. Μέσα από το εργαλείο αυτό και μέσα από την συμπλήρωση σχετικής φόρμας ερωτημάτων από μια μερίδα φοιτητών του τμήματος, θα προκύψει όλη η απαραίτητη πληροφορία σε σχέση με τα προβλήματα αλλά και τις ανάλογες προτάσεις προς την επίλυση τους.

Στην συνέχεια της μελέτης, θα υπάρξει η ανάλυση κάποιων προτάσεων από τους φοιτητές και ουσιαστικά έτσι θα βρεθούν και ορισμένες λύσεις που μέσα από την σύγκριση τους θα προκύψουν και οι καλύτερες. Στις οποίες, θα γίνει και η ενσωμάτωση του Πληροφοριακού Συστήματος «**UNIWA EXAM PLANNER**».

Έπειτα, θα ακολουθήσει η σχεδίαση και η διαγραμματική απεικόνιση του συγκεκριμένου πληροφοριακού συστήματος, μέσα από το εργαλείο **ADONIS Business Process Management Toolkit**. Με την δημιουργία **B.P.M.** διαγραμμάτων (**Business Process Model Diagrams**) καθώς και ενός **W.E.M.** διαγράμματος (**Working Environment Model Diagram**), θα γίνει η απεικόνιση της δομής των διαδικασιών αλλά και των ρόλων που έχουν οι χρήστες του συστήματος, ενώ παρακάτω θα ακολουθήσει και η περαιτέρω ανάλυση τους, μέσα από την λεκτική περιγραφή αυτών μέσω πινάκων.

Επιπρόσθετα, θα ακολουθήσει η ανάλυση της λειτουργίας που επιτελεί η κάθε μια από τις τεχνολογίες που χρησιμοποιήθηκαν στην παρούσα εργασία αλλά και η λεκτική τους περιγραφή όπως και η παρουσίαση τους ως οντότητες μεμονωμένα.

Επιπλέον, μέσα από ένα **E.R.** διάγραμμα (**Entity Relationship Diagram – E.R.D.**), το οποίο παράχθηκε μέσα από το εργαλείο **Workbench** της **MySQL**, θα προκύψει η διαγραμματική απεικόνιση και της βάσης δεδομένων «**examsdb**» όπως και νωρίτερα, θα ακολουθήσει η ανάλυση μέσω λεκτικής περιγραφής με χρήση δομής πινάκων τόσο στους πίνακες της βάσης και στην μεταξύ τους συσχέτιση, όσο και στα πεδία αυτών και στην πληροφορία που περιέχουν. Μετά, θα ακολουθήσει και η περιγραφή σε μορφή πίνακα των **Web Services** που θα αποτελέσουν το πληροφοριακό σύστημα στην τελική του μορφή.

Τέλος, θα γίνει η αναλυτική παρουσίαση της μεθοδολογίας εγκατάστασης όλων των απαραίτητων εργαλείων για την εκτέλεση του πληροφοριακού συστήματος αλλά και της παραγωγής του εκτελέσιμου κώδικα. Θα ακολουθήσει, ένα τμήμα της εργασίας με οδηγίες χρήσης του πληροφοριακού συστήματος και η εργασία ολοκληρώθηκε με αρκετά χρήσιμα συμπεράσματα αλλά και ιδέες για πιθανές βελτιώσεις του συστήματος στο μέλλον αλλά και τον κώδικα **Java** του συστήματος και της **MySQL** βάσης του σε μορφή παραρτημάτων. Για την παραγωγή του κώδικα του συστήματος, θα γίνει χρήση του **Intellij IDEA** και οι τεχνολογίες του **Frontend** και **Backend** είναι το **Vaadin Designer** καθώς και το **Spring Boot Web Services** αντίστοιχα.

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

- E.R.D. (Entity Relationship Diagram)
- B.P.M.N. (Business Process Model and Notation)
- B.P.M. (Business Process Management)
- W.E.M. (Working Environment Model)

Κεφάλαιο 1^ο - Σημερινή Κατάσταση Λειτουργίας του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών του Πανεπιστημίου Δυτικής Αττικής.

1.1 Καθημερινή Λειτουργία του Τμήματος εν καιρό Covid-19.

Η κατάσταση στην χώρα μας τα τελευταία χρόνια (από το 2010 κυρίως και μετά) μπορεί να χαρακτηριστεί από μέτρια έως και ως αρκετά δύσκολη σε διάφορους τομείς. Ένας από αυτούς τους τομείς είναι και η Παιδεία. Πιο συγκεκριμένα, όσον αφορά την Τριτοβάθμια Εκπαίδευση, με την πάροδο του χρόνου αρκετά ιδρύματα έχουν υποστεί διαφόρων ειδών περικοπές σε προσωπικό αλλά και σε υποδομές καθώς ακόμα υπάρχουν και περιπτώσεις συγχωνεύσεων ολόκληρων εκπαιδευτικών ιδρυμάτων. Αυτό το γεγονός σε συνδυασμό με την ετήσια αύξηση των εισαχθέντων φοιτητών και συνεπώς του συνόλου των φοιτητών ανά σχολή, τμήμα και ίδρυμα έχει ως αποτέλεσμα την παρουσίαση ενός κλίματος ασφυκτικού και μη λειτουργικού ως προς τις ανάγκες που παρουσιάζονται στην καθημερινότητα. Έτσι, έχοντας φτάσει πλέον σχεδόν μιάμιση δεκαετία μετά από το 2010 και αντιμετωπίζοντας πλέον την εμφάνιση του **Covid-19** στην Ελλάδα, έπειτα από δύο αρκετά μακροχρόνιες αλλά και αρκετά εξαντλητικές καραντίνες, οι φοιτητές πλέον κάθε εκπαιδευτικού ιδρύματος έχουν επιστρέψει στα έδρανα τους. Συνεπώς, το τμήμα Μηχανικών Πληροφορικής και Υπολογιστών της σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, έχοντας είδη στην πλάτη του όλα τα προαναφερθέντα προβλήματα σε συνδυασμό με τα πολλαπλά μέτρα της πολιτείας (μάσκες, απολύμανση και αερισμός αιθουσών) τα οποία θα πρέπει να εφαρμοστούν αναγκαστικά, έχει ως αποτέλεσμα να δημιουργηθεί σοβαρό πλήγμα στην καθημερινή λειτουργία του και αρκετοί από τους φοιτητές να μην μπορούν να εξυπηρετήσουν της καθημερινές τους ανάγκες καταλλήλως. Σκεφτείτε όμως, ότι λόγω του συνωστισμού που παρουσιάζεται το πρόβλημα μπορεί να είναι μεγαλύτερο κατά την διάρκεια των εξεταστικών περιόδων του τμήματος, όπου σίγουρα ορισμένοι φοιτητές θα αποτελούν και ευπαθείς ομάδες. Έτσι, πιθανόν ορισμένοι από αυτούς να χάσουν μέχρι και τις εξετάσεις τους σε ένα ή περισσότερα μαθήματα. Εδώ λοιπόν, εστιάζει η συγκεκριμένη διπλωματική εργασία, όπου μέσα από τις μελέτες της αλλά και από την τελική υλοποίηση ενός πληροφοριακού συστήματος, του "**UNIWA EXAM PLANNER**", θα έχει ως στόχο της την αποφυγή των οποιαδήποτε εμποδίων στέκονται ανάμεσα στον φοιτητή και στην καλύτερη εμπειρία φοίτησης του στο τμήμα.

Έτσι λοιπόν, το πρόβλημα που παρουσιάζει το τμήμα, είναι ιδιαίτερα πιο έντονο κατά τις εξετάσεις, στις οποίες βάση συγκεκριμένου εβδομαδιαίου προγράμματος το οποίο ορίζεται πρόωρα από τους αρμόδιους του τμήματος, πρέπει να μπορέσουν τόσο οι φοιτητές (περίπου 4000) ως προς το σύνολο τους να καταφέρουν να κατανεμηθούν στις διάφορες διαθέσιμες αίθουσες αλλά και το κάθε μάθημα να μην φτάσει να προκαλέσει το οποιαδήποτε πρόβλημα στα υπόλοιπα μαθήματα ως προς την διαθεσιμότητα αυτών των αιθουσών και για άλλες εξετάσεις μέσα στην ίδια εξεταστική περίοδο. Χαρακτηριστικό Παράδειγμα σε αυτήν την περίπτωση είναι, ένα μάθημα που εξετάζεται κατά την χειμερινή περίοδο 2023 να εξετάζεται παράλληλα και κατά την εμβόλιμη περίοδο 2023 στην χειμερινή περίοδο και μια ή περισσότερες αίθουσες να χρησιμοποιούνται και στις δύο διαφορετικές εξεταστικές περιόδους.

Επιπρόσθετα, έπειτα από την περαιτέρω ενασχόληση με το αντικείμενο της λειτουργίας του τμήματος κατά τις εξεταστικές περιόδους του, αλλά και της αναζήτησης πληροφοριών, καταλήξαμε στα εξής:

- 1) Στην παρούσα κατάσταση λειτουργίας του τμήματος, έχουμε στην διάθεση μας 5 αίθουσες και το Αμφιθέατρο Πληροφορικής.
- 2) Εκτός των παραπάνω, έχουμε και άλλες 6 εργαστηριακές αίθουσες και 1-2 μικρότερες βοηθητικές αίθουσες, αλλά αυτές μας χρησιμεύουν για τις εξετάσεις σπάνια έως καθόλου.
- 3) Οι αίθουσες αυτές έχουν συνολική χωρητικότητα καθώς και πλήθος διαθέσιμων θέσεων (πλην των κενών θέσεων) βάση του παρακάτω πίνακα.

Όνομα Αίθουσας	Συνολικό Πλήθος Θέσεων	Καθαρό Πλήθος Διαθέσιμων Θέσεων
ΑΜΦ	250	120
113, 114	81	54
201, 202, 219	68	34

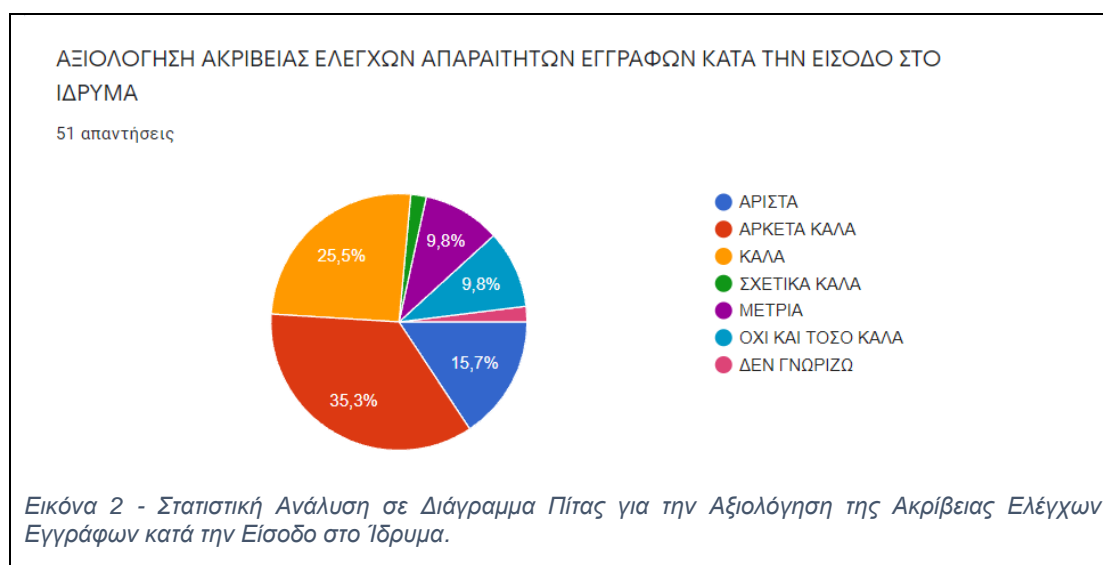
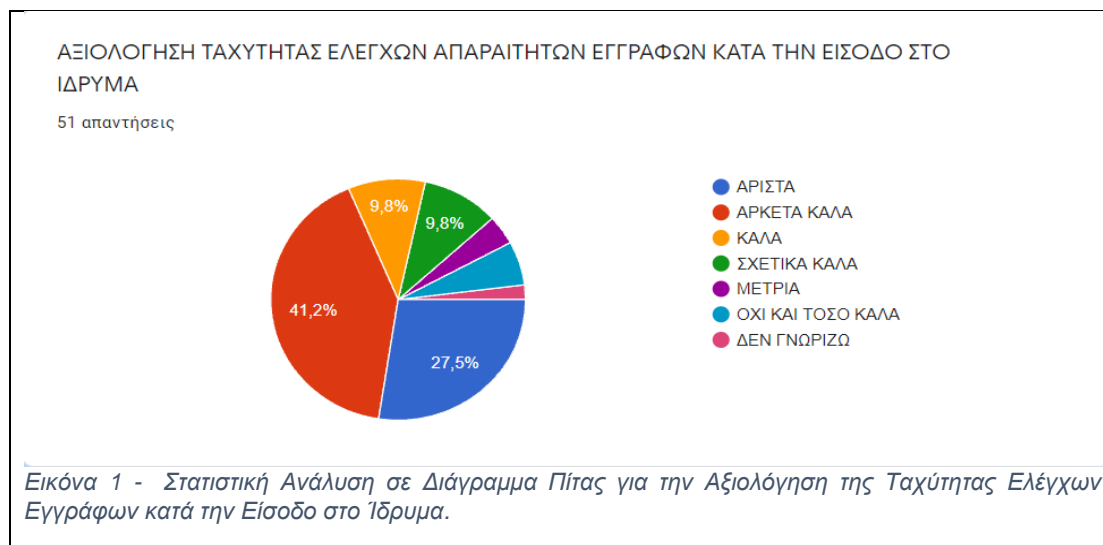
Πίνακας 1 - Πληροφορίες Διαθέσιμων Αιθουσών Τμήματος.

- 4) Τα αναγκαία μέτρα προς διεξαγωγή των εξετάσεων είναι:
 - Η υποχρεωτική χρήση μάσκας.
 - Η τοποθέτηση ενός φοιτητή ανά θρανίο για τα πολυπληθή μαθήματα και η εξάντληση της διαθεσιμότητας των χώρων, ώστε να εξασφαλίζεται η μεγαλύτερη δυνατή απόσταση μεταξύ εξεταζόμενων για τα λιγότερο πολυπληθή μαθήματα.
 - Ο έλεγχος στις πύλες εισόδου στο ΠΑΔΑ και είσοδος μόνον στους εμβολιασμένους ή με βεβαίωση αρνητικού rapid test.
 - Η τακτική απολύμανση χώρων, ιδανικά πριν κάθε εξέταση.
 - Η τήρηση αποστάσεων των εξεταζόμενων.
 - Η μείωση της ώρας εξέτασης ανά μάθημα στις 2,5 (έναντι 3), ώστε να υπάρχει επαρκής χρόνος για τον αερισμό των αιθουσών και την αποφυγή συνωστισμού μεταξύ εισερχόμενων και εξερχόμενων φοιτητών.

Συμπερασματικά, βάση των παραπάνω δεδομένων, καταλήγουμε πολύ εύκολα στο γεγονός ότι αρκετοί φοιτητές δεν μπορούν να εξυπηρετηθούν κατάλληλα κατά την διεξαγωγή των εξετάσεων τους, αρκετά από τα πρωτόκολλα δεν εφαρμόζονται (μερικώς ή ολικώς), είναι αρκετά χρονοβόρες οι διαδικασίες που απαιτούνται για την τήρηση τους και τέλος οι διαθέσιμες αίθουσες και οι γενικότερες εγκαταστάσεις του πανεπιστημίου δεν είναι σε θέση να καλύψουν έναν τόσο μεγάλο πλήθος φοιτητών, το οποίο ανά πάσα στιγμή τις χρειάζεται για την ομαλή λειτουργία του τμήματος και ιδιαίτερα κατά την περίοδο των εξετάσεων.

1.2 Πρώτες Εντυπώσεις των Φοιτητών.

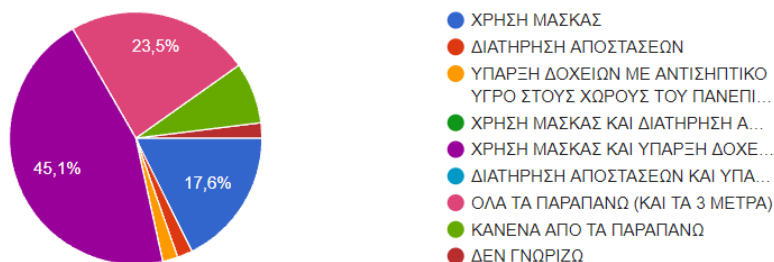
Στην συνέχεια της μελέτης του αντικειμένου της εργασίας, πραγματοποιήθηκε μια έρευνα με σκοπό την εξαγωγή της γνώμης των φοιτητών πάνω στην παρούσα κατάσταση λειτουργίας του τμήματος. Αυτή η έρευνα¹ πραγματοποιήθηκε μέσω της πλατφόρμας της Google και συγκεκριμένα μέσα από το εργαλείο «Google Form», όπου και παράχθηκε μια γενική εικόνα αρκετά εύκολα και σε ελάχιστο πλήθος ημερών για την σημερινή καθημερινή λειτουργία του τμήματος μέσα από τις απαντήσεις των φοιτητών του στις ερωτήσεις της φόρμας. Έτσι λοιπόν, όπως φαίνεται και παρακάτω, ακολουθούν τα αποτελέσματα αυτής της έρευνας σε μορφή εικόνων.



¹ Περίοδος Διεξαγωγής της Έρευνας: Νοέμβριος - Δεκέμβριος του 2021

ΑΝΑΦΟΡΑ ΜΕΤΡΩΝ ΠΟΥ ΕΦΑΡΜΟΖΟΝΤΑΙ ΚΑΤΑ ΤΗΣ ΔΙΑΣΠΟΡΑΣ ΤΟΥ COVID-19 ΣΤΟΥΣ ΧΩΡΟΥΣ ΤΟΥ ΙΔΡΥΜΑΤΟΣ

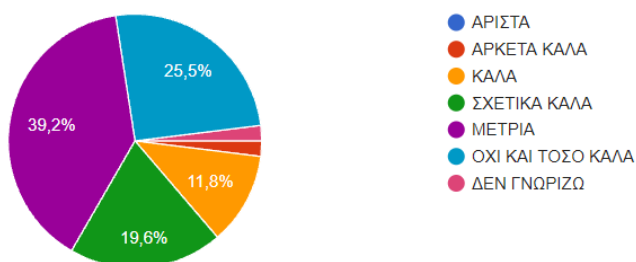
51 απαντήσεις



Εικόνα 3 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση του Ποσοστού ύπαρξης των κατάλληλων Μέτρων προστασίας από τον Covid-19.

ΑΞΙΟΛΟΓΗΣΗ ΤΗΡΗΣΗΣ ΜΕΤΡΩΝ ΚΑΤΑ ΤΗΣ ΔΙΑΣΠΟΡΑΣ ΤΟΥ COVID-19 ΣΤΟΥΣ ΧΩΡΟΥΣ ΤΟΥ ΙΔΡΥΜΑΤΟΣ

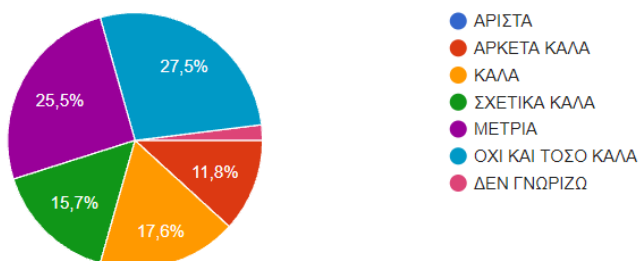
51 απαντήσεις



Εικόνα 4 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση του Ποσοστού εφαρμογής των κατάλληλων Μέτρων προστασίας από τον Covid-19.

ΑΞΙΟΛΟΓΗΣΗ ΠΟΙΟΤΗΤΑΣ ΤΗΣ ΔΙΑ ΖΩΣΗΣ ΕΚΠΑΙΔΕΥΣΗΣ (ΩΣ ΠΡΟΣ ΤΙΣ ΩΡΕΣ ΔΙΔΑΣΚΑΛΙΑΣ ΒΑΣΗ ΔΥΣΚΟΛΙΑΣ ΜΑΘΗΜΑΤΩΝ)

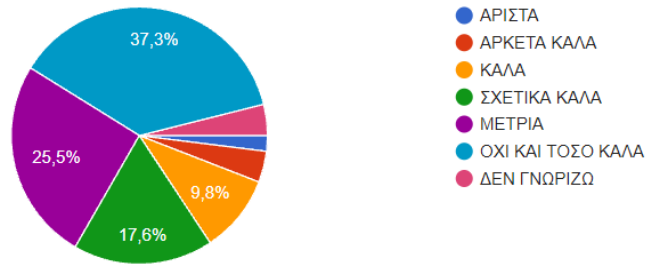
51 απαντήσεις



Εικόνα 5 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση της ποιότητας στην Δια Ζώσης Εκπαίδευση ως προς τις ώρες διδασκαλίας βάση της δυσκολίας των μαθημάτων.

ΑΞΙΟΛΟΓΗΣΗ ΠΟΙΟΤΗΤΑΣ ΤΗΣ ΔΙΑ ΖΩΣΗΣ ΕΚΠΑΙΔΕΥΣΗΣ (ΩΣ ΠΡΟΣ ΤΙΣ ΩΡΕΣ ΔΙΔΑΣΚΑΛΙΑΣ ΒΑΣΗ ΟΓΚΟΥ ΥΛΗΣ ΜΑΘΗΜΑΤΩΝ)

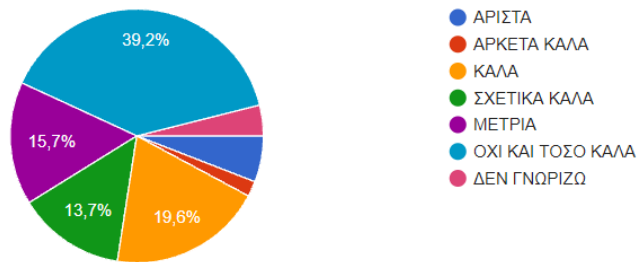
51 απαντήσεις



Εικόνα 6 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση της ποιότητας στην Δια Ζώσης Εκπαίδευση ως προς τις ώρες διδασκαλίας βάση του όγκου της ύλης των μαθημάτων.

ΑΞΙΟΛΟΓΗΣΗ ΠΟΙΟΤΗΤΑΣ ΤΗΣ ΔΙΑ ΖΩΣΗΣ ΕΚΠΑΙΔΕΥΣΗΣ (ΩΣ ΠΡΟΣ ΤΙΣ ΔΙΑΘΕΣΙΜΕΣ ΑΙΘΟΥΣΕΣ ΔΙΔΑΣΚΑΛΙΑΣ)

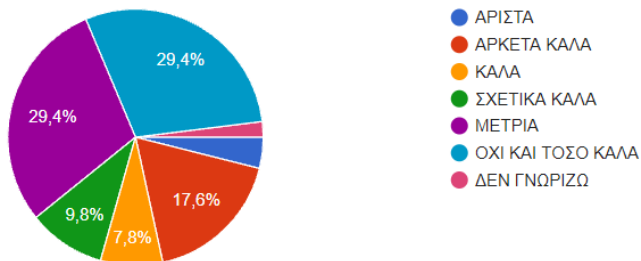
51 απαντήσεις



Εικόνα 7 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση της ποιότητας στην Δια Ζώσης Εκπαίδευση ως προς την διαθεσιμότητα αιθουσών.

ΑΞΙΟΛΟΓΗΣΗ ΠΟΙΟΤΗΤΑΣ ΤΗΣ ΔΙΑ ΖΩΣΗΣ ΕΚΠΑΙΔΕΥΣΗΣ (ΩΣ ΠΡΟΣ ΤΗΝ ΤΗΡΗΣΗ ΤΩΝ ΑΠΑΡΑΙΤΗΤΩΝ ΜΕΤΡΩΝ ΠΡΟΣΤΑΣΙΑΣ ΚΑΤΑ ΤΟΥ ΙΟΥ)

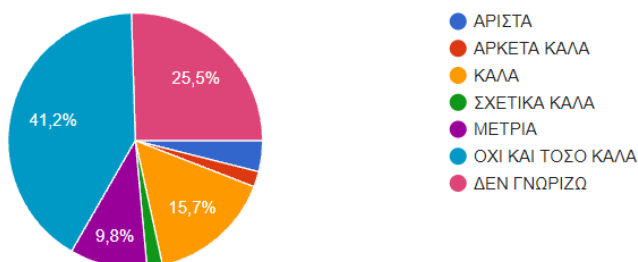
51 απαντήσεις



Εικόνα 8 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση της ποιότητας στην Δια Ζώσης Εκπαίδευση ως προς την τήρηση των απαραίτητων μέτρων προστασίας κατά του Covid-19.

ΑΞΙΟΛΟΓΗΣΗ ΠΟΙΟΤΗΤΑΣ ΕΚΠΑΙΔΕΥΣΗΣ (ΩΣ ΠΡΟΣ ΤΗΝ ΠΕΡΙΠΤΩΣΗ ΕΜΦΑΝΙΣΗΣ ΚΡΟΥΣΜΑΤΟΣ - ΕΠΙΣΤΡΟΦΗ ΣΕ ΤΗΛΕΚΠΑΙΔΕΥΣΗ)

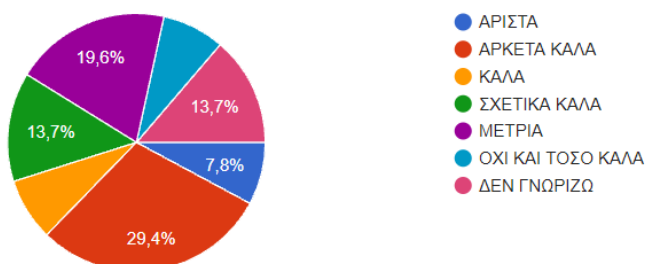
51 απαντήσεις



Εικόνα 9 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση της ποιότητας στην Εκπαίδευση ως προς το ποσοστό κινδύνου εμφάνισης κρούσματος και επιστροφής στην τηλεεκπαίδευση.

ΑΞΙΟΛΟΓΗΣΗ ΠΟΙΟΤΗΤΑΣ ΕΚΠΑΙΔΕΥΣΗΣ (ΩΣ ΠΡΟΣ ΤΑ ΕΡΓΑΣΤΗΡΙΑ)

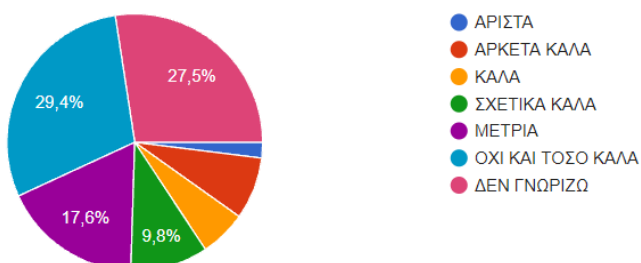
51 απαντήσεις



Εικόνα 10 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση της ποιότητας στην Δια Ζώσης Εκπαίδευση ως προς τα εργαστηριακά μαθήματα.

ΑΞΙΟΛΟΓΗΣΗ ΟΔΗΓΙΩΝ/ΜΕΤΡΩΝ ΠΟΥ ΕΦΑΡΜΟΖΟΝΤΑΙ ΣΕ ΠΕΡΙΠΤΩΣΗ ΕΜΦΑΝΙΣΗΣ ΚΡΟΥΣΜΑΤΟΣ ΣΤΟΥΣ ΧΩΡΟΥΣ ΤΟΥ ΠΑΝΕΠΙΣΤΗΜΙΟΥ

51 απαντήσεις



Εικόνα 11 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση των Μέτρων Προστασίας σε περίπτωση εμφάνισης κρούσματος στους χώρους του Πανεπιστημίου.

ΠΟΙΕΣ ΟΔΗΓΙΕΣ ΕΧΟΥΝ ΔΟΘΕΙ ΓΙΑ ΤΙΣ ΕΞΕΤΑΣΕΙΣ; (ΑΝ ΕΧΟΥΝ ΔΟΘΕΙ)

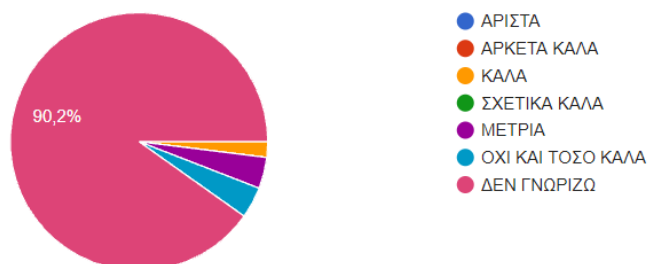
13 απαντήσεις



Εικόνα 12 - Απαντήσεις ως προς το Ερώτημα «Ποιες οδηγίες έχουν δοθεί για τις εξετάσεις;».

ΑΞΙΟΛΟΓΗΣΗ ΑΥΤΩΝ ΤΩΝ ΟΔΗΓΙΩΝ ΩΣ ΠΡΟΣ ΤΙΣ ΕΞΕΤΑΣΕΙΣ

51 απαντήσεις



Εικόνα 13 - Στατιστική Ανάλυση σε Διάγραμμα Πίτας για την Αξιολόγηση των οδηγιών που έχουν δοθεί για τις εξετάσεις.

ΣΥΜΠΕΡΑΣΜΑΤΙΚΑ ΠΟΙΑ ΕΙΝΑΙ Η ΑΠΟΨΗ ΣΑΣ ΓΙΑ ΤΙΣ ΔΙΑ ΖΩΣΗΣ ΔΙΑΛΕΞΕΙΣ ΕΝΑΝΤΙ ΤΗΣ ΤΗΛΕΚΠΑΙΔΕΥΣΗΣ;

28 απαντήσεις

Η τηλεκπαίδευση δεν έχει κανένα αρνητικό και δεν καταλαβαίνω γιατί κάνουμε δια ζώσης και βάζουμε τη ζωή μας σε κίνδυνο και χάνονται ώρες για καθαρισμό των αιθουσών ακόμη και στα εργαστήρια

Προσωπικά θα προτιμούσα τηλεδιάσκεψη γιατί κάποια πράγματα που δεν βλέπω καλά στον πίνακα στο αμφιθέατρο σίγουρα θα τα βλέπω από τον υπολογιστή και θα ακούω και καλύτερα.

Ίσως έπρεπε να συνεχιστεί η τηλεκπαίδευση, καθώς κάποιοι ίσως είμαστε ασυμπτωματικοί χωρίς να το γνωρίζουμε και να γίνεται διασπορά του ιού χωρίς να το αντιλαμβανόμαστε εγκαίρως.

Είμαστε πολλά άτομα και ο κοβιντ παραμονεύει

Καθόλου καλή

Το μόνο πλεονέκτημα είναι στα εργαστήρια, αλλιώς η τηλεκπαίδευση είναι προτιμότερη.

Στα τμήματα που είναι δυνατή η τηλεκπαίδευση χωρίς να εμποδίζεται η ομαλή τελετουργία του μαθήματος θεωρώ πως είναι καλή ως προοπτική ή τουλάχιστον ένα υβριδικό μοντέλο αυτής.

Εικόνα 14 - Απαντήσεις φοιτητών πάνω στο ερώτημα «Ποια είναι η άποψη σας για τις δια ζώσης διαλέξεις έναντι της τηλεκπαίδευσης;».

ΣΥΜΠΕΡΑΣΜΑΤΙΚΑ ΠΟΙΑ ΕΙΝΑΙ Η ΑΠΟΨΗ ΣΑΣ ΓΙΑ ΤΙΣ ΔΙΑ ΖΩΣΗΣ ΔΙΑΛΕΞΕΙΣ ΕΝΑΝΤΙ ΤΗΣ ΤΗΛΕΚΠΑΙΔΕΥΣΗΣ;

28 απαντήσεις

Χάνεται πολύτιμος χρόνος μαθήματος για την μερική τήρηση των μέτρων η οποία είναι πρακτικά αδύνατη καθώς οι αίθουσες είναι γεμάτες καθημερινά

Δεν μπορεί ο καθηγητής να κάνει ταυτόχρονα μάθημα και να ελέγχει ποιος έχει βγάλει τη μάσκα κατά τη διάρκεια του μαθήματος. Σημαντική μερίδα φοιτητών βγάζουν τη μάσκα κατά τη διάρκεια των διαλέξεων. Πρέπει να γυρίσουμε στην τηλεκπαίδευση γιατί δεν τηρούνται τα μέτρα σε βαθμό που να είμαστε ασφαλείς στα αμφιθέατρα

Καλύτερα τηλεκπαίδευση

ανάλογα το μάθημα, το τέλειο θα ήταν μεικτά για όσα μαθήματα έχουν μεγάλο όγκο φοιτητών που τα παρακολουθούν γιατί σε αυτά τα μαθήματα δεν τηρούνται αποστάσεις ούτε για αστέιο (μιλάμε για ορισμένα μαθήματα που άτομα παρακολούθησαν την διάλεξη όρθιοι λόγω έλλειψης θέσεων)

Τα μέτρα τηρούνται, φυσικά όμως υπάρχουν τα άτομα που δεν ενδιαφέρονται για τους υπόλοιπους που δεν τηρούν την χρήση μάσκας σε κλειστούς χώρους πχ διαδρόμους

ΕΠΡΕΠΕ ΝΑ ΣΥΝΕΧΙΣΤΕΙ Η ΤΗΛΕΚΠΑΙΔΕΥΣΗ

ΣΥΜΠΕΡΑΣΜΑΤΙΚΑ ΠΟΙΑ ΕΙΝΑΙ Η ΑΠΟΨΗ ΣΑΣ ΓΙΑ ΤΙΣ ΔΙΑ ΖΩΣΗΣ ΔΙΑΛΕΞΕΙΣ ΕΝΑΝΤΙ ΤΗΣ ΤΗΛΕΚΠΑΙΔΕΥΣΗΣ;

28 απαντήσεις

Είναι καλύτερης ποιότητας κυρίως από την άποψη ότι γίνονται στον χώρο που πρέπει , στη τηλεκπαίδευση ήταν αρκετά εύκολο να αφαιρεθεί κάποιος .

Τηλεκπαίδευση μέχρι να φύγει ο κοβιντ

Καλύτερη η δια ζώσης έστω και με την τήρηση των μέτρων

Αναγκαία μόνο για τα εργαστήρια που έχουν ανάγκη τη φυσική παρουσία. Τα εργαστήρια που δεν είναι αναγκαία η φυσική παρουσία είναι τα ίδια με την τηλεκπαίδευση

Η τηλεκπαίδευση δουλευε πολυ καλυτερα απο τα δια ζωσης μαθηματα.

Σε τμήματα μηχανικών η τηλεκπαίδευση επέφερε καλύτερα αποτελέσματα από ότι η δια ζώσης. Και η δικαιολογία καθηγητών ότι δεν μπορούν και δεν τους αρέσει η τηλεκπαίδευση είναι απλά δικαιολογίες. Στην πράξη η τηλεκπαίδευση έκανε πιο εύκολο το μάθημα ακόμα και όταν τα τμήματα ήταν μερικές 100αδες άτομα.

Να υπήρχε η δυνατότητα ταυτόχρονης συγχρόνως τηλεκπαίδευσης (υβριδικό σύστημα) και όχι αποκλειστικά δια ζώσης . Επί παραδείγματι για ανοσοκαταστολμένους.

ΣΥΜΠΕΡΑΣΜΑΤΙΚΑ ΠΟΙΑ ΕΙΝΑΙ Η ΑΠΟΨΗ ΣΑΣ ΓΙΑ ΤΙΣ ΔΙΑ ΖΩΣΗΣ ΔΙΑΛΕΞΕΙΣ ΕΝΑΝΤΙ ΤΗΣ ΤΗΛΕΚΠΑΙΔΕΥΣΗΣ;

28 απαντήσεις

Τα μέτρα δεν εφαρμόζονται, είμαστε σε κίνδυνο, δεν υπάρχουν ούτε αντισηπτικά η σαπούνι στις τουαλέτες, δεν υπάρχουν μέτρα και το μόνο που ισχύει είναι η χρήση μάσκας μέσα στις αίθουσες όταν γίνεται το μάθημα. Τίποτα άλλο

Είναι αρκετά διαφορετική η κάθε μορφή εκπαίδευσης. Λόγω συμφόρησης στις αίθουσες το μάθημα γίνεται πιο δυσνόητο και τα μέτρα στη τελική δεν τηρούνται στο 100%. Κάτι που με τη τηλεκπαίδευση δεν υπήρχε και κυλούσε το μάθημα πολύ ομαλά τόσο οι θεωρίες όσο και τα εργαστήρια

Η τηλεκπαίδευση ήταν ένα αναγκαίο κακό και καλό θα ήταν να μην ξαναγίνει.

Η δια ζώσης με τόσα άτομα είναι χειρότερη από την τηλεκπαίδευση

Δεν επιθυμώ να απαντήσω σε αυτό.

χαλί μαύρο

Η φετινή δια ζώσης, καμία σχέση με τη δια ζώσης πριν 2 χρόνια. Μείωση των κανονικών ωρών (συνεπώς η ύλη δεν πρόκειται να βγει και πέφτει μεγαλύτερο φόρτο εργασίας στον φοιτητή), μικρές αίθουσες (δεδομένου ότι πρέπει να τηρούνται κ οι αποστάσεις) κ.λ.π. Προτιμότερη η τηλεκπαίδευση..

Η φετινή δια ζώσης, καμία σχέση με τη δια ζώσης πριν 2 χρόνια. Μείωση των κανονικών ωρών (συνεπώς η ύλη δεν πρόκειται να βγει και πέφτει μεγαλύτερο φόρτο εργασίας στον φοιτητή), μικρές αίθουσες (δεδομένου ότι πρέπει να τηρούνται κ οι αποστάσεις) κ.λ.π. Προτιμότερη η τηλεκπαίδευση.. Τουλάχιστον τότε δεν χάναμε κατά μέσο όρο 3 ώρες στα ΜΜΜ για να παρακολουθήσουμε μάθημα της 1μιση ώρας (που κανονικά θα ήταν 3ωρο) και αυτό αν προλάβουμε θέση.

Θα ήταν ικανοποιητική αν υπήρχε τήρηση μέτρων

ΥΠΑΡΧΕΙ ΚΑΠΟΙΑ/ΕΣ ΠΡΟΤΑΣΗ/ΕΙΣ ΠΟΥ ΘΑ ΘΕΛΑΤΕ ΝΑ ΚΑΝΕΤΕ ΓΙΑ ΤΗΝ ΒΕΛΤΙΩΣΗ ΤΗΣ ΠΟΙΟΤΗΤΑΣ ΦΟΙΤΗΣΗΣ ΣΑΣ ΤΗΝ ΤΡΕΧΟΥΣΑ ΠΕΡΙΟΔΟ ΣΤΟ ΠΑ.Δ.Α. ;

23 απαντήσεις

να κανουμε τηλεκπαιευση	^
Φέτος λόγω covid 19 πιστεύω θα ήταν καλύτερη η τηλεκπαίδευση	
Μια πρόταση θα ήταν να υπάρχουν περισσότερες αίθουσες διδασκαλίας για να μοιράζεται σωστά ο αριθμός των φοιτητών.	
Ναι αλλαγή μερικών καθηγητών και εξεταστική μέσω τηλεδιάσκεψης για να μην κολλήσουμε όλοι κοβιντ όχι τίποτα άλλο	
Κατάληψη ή να γίνονται ξανά τα μαθήματα online	
Όχι.	
Διαχωρισμός των τμημάτων σε μικρότερα και μείωση της ύλης καθώς και να ανεβαίνουν τα μαθήματα σε διάφορες πλατφόρμες για παρακολούθηση των μαθημάτων από τους μαθητές	
Πραγματική εφαρμογή των μέτρων. Λέμε ότι εφαρμόζονται τα μέτρα ενώ ταυτόχρονα κυκλοφορεί μεγάλη μερίδα φοιτητών στους διαδρόμους στα κτήρια χωρίς μάσκες ή με μυτούλες έξω από τη μάσκα.	v

Εικόνα 15 - Απαντήσεις των φοιτητών ως προς το ερώτημα «Ποιες προτάσεις θα κάνατε, για την βελτίωση της ποιότητας φοίτησης σας στο Πανεπιστήμιο Δυτικής Αττικής την τρέχουσα περίοδο;».

ΥΠΑΡΧΕΙ ΚΑΠΟΙΑ/ΕΣ ΠΡΟΤΑΣΗ/ΕΙΣ ΠΟΥ ΘΑ ΘΕΛΑΤΕ ΝΑ ΚΑΝΕΤΕ ΓΙΑ ΤΗΝ ΒΕΛΤΙΩΣΗ ΤΗΣ ΠΟΙΟΤΗΤΑΣ ΦΟΙΤΗΣΗΣ ΣΑΣ ΤΗΝ ΤΡΕΧΟΥΣΑ ΠΕΡΙΟΔΟ ΣΤΟ ΠΑ.Δ.Α. ;

23 απαντήσεις

να εφαρμοστεί το μεικτό σύστημα διδασκαλίας από όσους καθηγητές το επιθυμούν ή (προσωρινά λόγω της κατάστασης στις ΜΕΘ σε όλη την χώρα αυτή την περίοδο(Δεκέμβριος 2021) να είναι δια ζώσης μόνο τα εργαστήρια	^
ΟΧΙ	
Σωστή τήρηση των μέτρων και επισημάνσεις των καθηγητών όταν παρατηρούν μη τήρηση τους.	
περισσότερη φροντίδα στις τουαλέτες και περισσότερα παγκάκια και υπόστεγα για αποφυγή συνωστισμού	
Τηλεκπαίδευση	
Χρήση βίντεο και επιπλέον υλικού που υπήρχε διαθέσιμο στην τηλεκπαίδευση για την περίπτωση των παιδιών που πρέπει να λείψουν από το πανεπιστήμιο λόγω καραντίνας	
Επιστροφή στην τηλεκπαίδευση ή τουλάχιστον δυνατότητα τηλεκπαίδευσης για τα εργαστήρια, γιατί δεν έχουν θέσεις για άτομα και οι μισοί δεν μπόρεσαν να παρακολουθήσουν εργαστήρια.	v

ΥΠΑΡΧΕΙ ΚΑΠΟΙΑ/ΕΣ ΠΡΟΤΑΣΗ/ΕΙΣ ΠΟΥ ΘΑ ΘΕΛΑΤΕ ΝΑ ΚΑΝΕΤΕ ΓΙΑ ΤΗΝ ΒΕΛΤΙΩΣΗ ΤΗΣ ΠΟΙΟΤΗΤΑΣ ΦΟΙΤΗΣΗΣ ΣΑΣ ΤΗΝ ΤΡΕΧΟΥΣΑ ΠΕΡΙΟΔΟ ΣΤΟ ΠΑ.Δ.Α. ;

23 απαντήσεις

Επαναφορά της τηλεκπαίδευσης για όλα τα θεωρητικά μαθήματα και optional η δια ζώσης στα εργαστήρια αν η φύση των μαθημάτων είναι πρακτική. Εξαιταση από κοντά μόνο εάν μπορούν να εφαρμοστούν τα μέτρα και χωρίς εκπτώσεις σε αυτά. Διαφορετικά εξαιταση από απόσταση. Δεν υπάρχει δικαιολογία στην ασφάλεια και στην υγεία. Και πόσο μάλλον όταν καθηγητές ραδυνατουν να χρησιμοποιήσουν τα εργαλεία , να υποχρεωθούν σε ένα γρήγορο σεμινάριο για την ορθή χρήση τους

ΥΠΑΡΧΕΙ ΚΑΠΟΙΑ/ΕΣ ΠΡΟΤΑΣΗ/ΕΙΣ ΠΟΥ ΘΑ ΘΕΛΑΤΕ ΝΑ ΚΑΝΕΤΕ ΓΙΑ ΤΗΝ ΒΕΛΤΙΩΣΗ ΤΗΣ ΠΟΙΟΤΗΤΑΣ ΦΟΙΤΗΣΗΣ ΣΑΣ ΤΗΝ ΤΡΕΧΟΥΣΑ ΠΕΡΙΟΔΟ ΣΤΟ ΠΑ.Δ.Α. ;

23 απαντήσεις

Να ισχύουν τα μέτρα προστασίας,
Να υπάρχουν τα απαραίτητα για απολύμανση των φοιτητών (σαπούνι, χαρτί, αντισηπτικό)
Περισσότερες θέσεις στα εργαστήρια γιατί μένουμε απ έξω λόγω περιορισμού θέσεων.
Αν δεν τα μπορούν αυτά να τα εφαρμόσουν να το γυρίσουν σε τηλεκπαίδευση μη μας βάζουν σε κίνδυνο

Υβριδική εκπαίδευση λόγω της κατάστασης

Να ελέγχουν rapid tests και τους εμβολιασμένους

Λιγότερα άτομα στην αίθουσα και περισσότερα αμφιθέατρα

Όχι.

Λίγη κατανόηση για όλους τους φοιτητές που αφιερώνουν τόσες ώρες στα ΜΜΜ για να πάνε στη σχολή να παρακολουθήσουν μισό μάθημα, κενό, μάθημα, κενό κλπ χωρίς να τηρούνται πάντα όλα τα μέτρα κατά της μετάδοσης covid (φοιτητές με μάσκα που καλύπτει μόνο στόμα, μη τήρηση των αποστάσεων εντός και εκτός αιθουσών, μη χρήση αντισηπτικού κλπ). Έχουμε διπλάσιο τουλάχιστον φόρτο εργασίας κ αυτό διότι δεν γίνεται ολόκληρο μάθημα, έχουμε πλήθος εργασιών που δεν προλαβαίνουμε να κάνουμε, εφόσον θέλουμε έξτρα χρόνο για να μάθουμε αυτά που οι καθηγητές δεν προλαβαίνουν να μας διδάξουν στην 1μιση ώρα που έχει "αφαιρεθεί" από το μάθημα Επίσης, σωστή και πλήρη ενημέρωση σχετικά με τα κρούσματα covid. Δεν είναι αρκετό και σωστό να ενημερώνονται μόνο φοιτητές που ήταν στο ίδιο μάθημα με το κρούσμα, δεδομένου ότι όλες οι αίθουσες χρησιμοποιούνται και από άλλους φοιτητές κ καθηγητές την ίδια μέρα. Ενημέρωση για όλους τους φοιτητές που τυχόν είχαν μάθημα στην ίδια αίθουσα τουλάχιστον!

1.3 Συμπεράσματα βάσει της γνώμης των Φοιτητών πάνω στην Σημερινή Κατάσταση.

Μέσα από την συλλογή αλλά και την ανάλυση των αποτελεσμάτων τα οποία προέκυψαν από την έρευνα που διεξάχθηκε, παράχθηκαν δύο μορφές χρήσιμης πληροφορίας. Η πρώτη είναι ποια προβλήματα έχει φέρει στο τμήμα η παρούσα περίοδος που διανύουμε και συγκεκριμένα η ύπαρξη του **Covid-19** και η δεύτερη και τελευταία αλλά και εξίσου χρήσιμη πληροφορία που πήραμε από την έρευνα, είναι ορισμένες αρκετά αξιόλογες προτάσεις από τους φοιτητές για την βελτίωση αυτής της κατάστασης. Στις επόμενες υπό ενότιες, ακολουθούν αναλυτικότερα αυτές οι πληροφορίες που προέκυψαν από την περαιτέρω ανάλυση των αποτελεσμάτων της έρευνας αυτής.

1.3.1 Ανάλυση των Προβλημάτων που προκύπτουν κατά την Καθημερινή Λειτουργία του Τμήματος.

Θεωρήθηκε απαραίτητο να πραγματοποιηθεί μια έρευνα, μέσω της οποίας υπήρξε μάλιστα αρκετά σημαντική συμμετοχή από αρκετούς φοιτητές του Πανεπιστημίου Δυτικής Αττικής και σε συνδυασμό με τις δυνατότητες του εργαλείου της Google «Google Form», προέκυψαν τα παρακάτω συμπεράσματα για την σημερινή κατάσταση λειτουργίας του τμήματος:

1. Παρατηρήθηκε αρκετά συχνά, η μη τήρηση των μέτρων προστασίας βάση των υγειονομικών κανονισμών της πολιτείας κατά του **Covid-19** στους χώρους του Πανεπιστημίου. Για παράδειγμα αρκετοί φοιτητές παραπονήθηκαν για την έλλειψη σαπουνιών, αντσηπτικών και γενικά ειδών υγιεινής στις τουαλέτες, όπως επίσης και για το ότι δεν ήταν τυπικοί στην χρήση μάσκας στους διάφορους χώρους του τμήματος, από τις αίθουσες διδασκαλίας μέχρι και τους κύριους διαδρόμους του ιδρύματος.
2. Επιπλέον, γνωστοποιήθηκε ότι δεν υπήρχαν, τα απαραίτητα χρονικά διαστήματα για τον σωστό αερισμό αλλά και την απολύμανση των αιθουσών του τμήματος, κατά τα διάφορα διαλείμματα μεταξύ των διαλέξεων και των εξετάσεων.
3. Άλλο ένα πρόβλημα, που οφείλεται στο μεγάλο πλήθος των φοιτητών, είναι ότι γεμίζουν πλήρως όλες οι διαθέσιμες αίθουσες, σε βαθμό που δεν είναι δυνατή η σωστή παρακολούθηση των εξετάσεων απ' όλους. Χαρακτηριστικά παραδείγματα, είναι η δημιουργία υπερβολικού θορύβου από την μαζική παρουσίαση των φοιτητών στις ακαδημαϊκές αίθουσες, η μη δυνατή εύρεση θέσης απ' όλους στον χώρο της κάθε αίθουσας, καθώς επίσης και η μη τήρηση των αντίστοιχων υγειονομικών μέτρων, όπως είναι η τήρηση των απαραίτητων αποστάσεων βάση οδηγιών της πολιτείας για την προστασία κατά του **Covid-19**.

4. Επιπρόσθετα, άλλο ένα σοβαρό ζήτημα είναι η μείωση διδακτικών ωρών ανά μάθημα και συνεπώς η δυσφορία που παρουσιάζει τόσο το διδακτικό προσωπικό αλλά και οι φοιτητές στην αφομοίωση της προβλεπόμενης ετήσιας ύλης, ως προς τον όγκο και την δυσκολία κατανόησης ανά μάθημα στο πρόγραμμα σπουδών. Το συγκεκριμένο πρόβλημα μπορεί να είναι υπεύθυνο και για αρκετά χαμηλά ποσοστά επιτυχίας από τους φοιτητές, ανά τις εξεταστικές περιόδους του τμήματος.
5. Άλλο ένα πρόβλημα που συναντάμε αρκετά συχνά, οφείλεται λόγω της συμφόρησης των αιθουσών και της μη δυνατής τήρησης όλων των απαραίτητων μέτρων και κανονισμών για την προστασία από τον **Covid-19** στους χώρους του Πανεπιστημίου κατά την διάρκεια των εξετάσεων, με αποτέλεσμα να είναι πιο άμεση και η διασπορά του ιού (περίπτωση ασυμπτωματικού κρούσματος).
6. Πέρα από την μη σωστή τήρηση, ύπαρξη των απαραίτητων μέτρων και οδηγιών για την προστασία από τον κορονοϊό, υπάρχει πρόβλημα και στην περίπτωση που θα υπάρξει κάποιο κρούσμα και συγκεκριμένα όσον αφορά την διαδικασία αντιμετώπισης μιας τέτοιας κατάστασης. Όπως φαίνεται, μια τέτοια κατάσταση έχει ως αποτέλεσμα να καθιστά αρκετά δύσκολη την όλη λειτουργία του τμήματος, καθώς επίσης είναι αρκετά χρονοβόρα ως προς την διεκπεραίωση της, με χαρακτηριστικό παράδειγμα την καραντίνα σε μεγάλες ομάδες φοιτητών για κάποιο διάστημα αλλά και την απαραίτητη απολύμανση των χώρων του πανεπιστημίου. Εντελώς συμπληρωματικά και συγκεκριμένα στην περίπτωση μιας εξεταστικής περιόδου, αυτό μπορεί να προκαλέσει την αναβολή διάφορων εξετάσεων και έτσι η εξεταστική περίοδος να πάρει αρκετά μεγάλη παράταση χρονικά, εξαιτίας των πολλαπλών μαθημάτων που αναγκαστικά θα πρέπει να αλλάξουν ημερομηνία εξέτασης δημιουργώντας μια γενική σύγχυση και στο ετήσιο ακαδημαϊκό ημερολόγιο του τμήματος.
7. Επίσης, σε ότι αφορά τους φοιτητές που θα έπαιρναν πτυχίο το 2021, έγινε αναγκαστικά η απονομή στο Πολιτιστικό Κέντρο του Δήμου Αιγάλεω Γιάννης Ρίτσος.
8. Τέλος, το πρόβλημα με την συμφόρηση των αιθουσών στις θεωρίες υπάρχει και στα Εργαστήρια και έτσι δεν μπορούν όλοι οι φοιτητές να παρακολουθήσουν σωστά ένα εργαστηριακό μάθημα. Πιο συγκεκριμένα, μιλάμε για την μη δυνατή εύρεση διαθέσιμων θέσεων και την ύπαρξη αρκετών υπεράριθμων φοιτητών, που δεν κατάφεραν να μπουν σε κάποιο από τα αρχικά προβλεπόμενα τμήματα που έχει ορίσει ο εκάστοτε καθηγητής για το κάθε εργαστηριακό σκέλος ανά μάθημα. Το ίδιο μπορεί να συμβεί και κατά την περίοδο των εργαστηριακών εξετάσεων.

1.3.2 Εξαγωγή Προτάσεων των Φοιτητών ως Πιθανές Λύσεις.

Μέσα από την έρευνα που διεξάχθηκε αλλά και την μελέτη των αποτελεσμάτων που συγκεντρώθηκαν στην τρέχουσα φάση της εργασίας, προέκυψαν αρκετές αξιόλογες προτάσεις, μέσα από τις απαντήσεις των φοιτητών στο ερωτηματολόγιο της φόρμας της Google οι οποίες ακολουθούν ενδεικτικά στην συνέχεια:

1. Μια ιδέα αρκετά χρήσιμη, θα μπορούσε να είναι η χρήση υβριδικής εκπαίδευσης, δηλαδή με λίγα λόγια η καθιέρωση σε δια ζώσης μαθήματα μόνο για τα εργαστήρια και η εφαρμογή της τηλεκπαίδευσης για τις θεωρητικές διαλέξεις.
2. Άλλη μια πρόταση από φοιτητή του τμήματος, είναι η ύπαρξη βίντεο και επιπλέον εκπαιδευτικού υλικού, το οποίο μάλιστα υπήρχε διαθέσιμο κατά την περίοδο της τηλεκπαίδευσης, για την περίπτωση φοιτητών που θα είναι πιθανά ή επιβεβαιωμένα κρούσματα και θα πρέπει να μείνουν σε καραντίνα για ορισμένες μέρες.
3. Επιπλέον, καλό θα ήταν να υπάρχει η σωστή τήρηση των μέτρων αποφυγής του ιού. Με άλλα λόγια, η τήρηση αποστάσεων, η χρήση μάσκας σε όλους τους χώρους αλλά και η ύπαρξη όλων των κατάλληλων μέσων όπως χαρτί υγείας, σαπούνι και αντισηπτικό για την σωστή απολύμανση και προστασία των φοιτητών κατά την παραμονή τους στους χώρους τους τμήματος.
4. Κάτι που θα μπορούσε να βοηθήσει ουσιαστικά στην καλύτερη δυνατή διεξαγωγή και των εργαστηριακών μαθημάτων, θα ήταν το να υπάρξουν περισσότερες διαθέσιμες θέσεις στα τμήματα των εργαστηρίων, αν και εφόσον αυτό είναι εφικτό. Για παράδειγμα, θα μπορούσε να γίνει χρήση και ορισμένων διαθέσιμων αιθουσών που υπάρχουν για τις θεωρητικές διαλέξεις σε ορισμένες περιπτώσεις.
5. Επιπρόσθετα, θα ήταν καλό να κριθεί απαραίτητη η λήψη «Rapid Test», ακόμα και σε εμβολιασμένους, κατά την είσοδο τους στους χώρους του Πανεπιστημίου Δυτικής Αττικής, με κύριο στόχο την αποφυγή πιθανόν κρουσμάτων (ασυμπτωματικών και μη).
6. Μια ακόμα καλή ιδέα, θα μπορούσε να είναι ο διαχωρισμός των τμημάτων σε ακόμα μικρότερα, για την καλύτερη δυνατή παρακολούθηση από τους φοιτητές αλλά και την μείωση της ύλης ανάλογα με τον όγκο της.
7. Τέλος, αρκετά σημαντικό θα ήταν η ύπαρξη της καλύτερης δυνατής αντιμετώπισης των περιπτώσεων ύπαρξης κρουσμάτων στο Πανεπιστήμιο, η έγκαιρη και ταχεία εφαρμογή των αντίστοιχων μέτρων και οδηγιών βάσει της πολιτείας αλλά και η παροχή της καλύτερης δυνατής ενημέρωσης του προσωπικού και των σπουδαστών του τμήματος, ώστε να βρίσκονται σε επιφυλακή ανά πάσα στιγμή για νέα πιθανά κρούσματα.

Κεφάλαιο 2^ο – Ανάλυση των Διαθέσιμων Προτάσεων των Φοιτητών και Εξαγωγή της Καλύτερης Λύσης.

2.1 Μελέτη Προτάσεων και Εξαγωγή Βέλτιστων Δυνατών Λύσεων.

Στο συγκεκριμένο κεφάλαιο θα παρουσιαστεί μια εκτενέστερη μελέτη των προτάσεων αλλά και πιθανών λύσεων που προέκυψαν από τους φοιτητές του τμήματος Μηχανικών Πληροφορικής και Υπολογιστών, με σκοπό την εξαγωγή της καλύτερης λύσης για την παρούσα κατάσταση λειτουργίας του ιδρύματος.

Μέσα λοιπόν από την λεπτομερή εξέταση των προτάσεων των φοιτητών, που καταγράφηκαν στο προηγούμενο κεφάλαιο, προέκυψαν οι παρακάτω πιθανές λύσεις πάνω στο πρόβλημα μας, καθώς και η ανάλυση τους στα Πλεονεκτήματα και Μειονεκτήματα τους στο παρακάτω πίνακάκι που ακολουθεί:

Πιθανές Λύσεις (Προτάσεις Φοιτητών)	Πλεονεκτήματα	Μειονεκτήματα
Χρήση υβριδικού συστήματος φοίτησης και ύπαρξη δια ζώσης εκπαίδευσης μόνο για τα εργαστήρια και τηλεκπαίδευσης για τις θεωρίες.	1)Μερική αποσυμφόρηση των αιθουσών, λόγω ύπαρξης τηλεκπαίδευσης στα θεωρητικά μέρη των μαθημάτων. 2)Αύξηση της διαθεσιμότητας των αιθουσών για χρήση τους και σε άλλες εκπαιδευτικές διαδικασίες, όπως για παράδειγμα σε περιπτώσεις εκδηλώσεων. 3)Καλύτερη αφομοίωση της ύλης από τους φοιτητές και πιο ξεκούραστη διδασκαλία, αφού υποστηρίζεται η ύπαρξη ενός ενιαίου τμήματος για τις θεωρίες. 4)Βελτιωμένη παρακολούθηση από τους φοιτητές, χωρίς να υπάρχει ιδιαίτερη φασαρία λόγω πολυκοσμίας ή και δυσκολία ως προς τον χώρο και την επιλογή θέσης. 5)Πιθανή σταδιακή μείωση των κρουσμάτων, λόγω του ότι τα εργαστηριακά τμήματα είναι αρκετά μικρότερα σε πλήθος φοιτητών από τα θεωρητικά τμήματα και πιο εύκολα διαχειρίσιμα ως προς την τήρηση των οδηγιών και των μέτρων προστασίας από τον Covid-19. 6)Σημαντική εξοικονόμηση χρόνου από την μείωση των ελέγχων που απαιτούνται στα απαραίτητα πιστοποιητικά κατά την είσοδο στους χώρους του τμήματος. 7)Ταχύτερη προσέλευση φοιτητών ως προς τα θεωρητικά μέρη των μαθημάτων. 8)Λιγότερες καθυστερήσεις από την πλευρά της απολύμανσης και του σωστού αερισμού του χώρου που είναι απαραίτητη τόσο πριν αλλά και μετά από τις διαλέξεις.	1)Αποχή από την αμεσότητα που παρέχει η δια ζώσης φοίτηση, για παράδειγμα σε περιπτώσεις επίλυσης πιθανών αποριών αλλά και ως προς τα φυσικά μέσα που περιλαμβάνει μια αίθουσα διδασκαλίας. 2)Πολύ μεγάλη πιθανότητα ύπαρξης τεχνικών προβλημάτων και μη διαθεσιμότητας φοιτητών να παρακολουθήσουν ορισμένες από τις διαλέξεις, κυρίως λόγω ακραίων καιρικών φαινομένων που μπορούν να προκαλέσουν πιθανές βλάβες, τόσο τον χειμώνα αλλά και το καλοκαίρι. 3)Υπάρχει εξίσου θέμα και στα εργαστήρια όπως και στις θεωρίες τόσο σε έλλειψη αιθουσών για διδασκαλία, εξοπλισμού αλλά και στην ύπαρξη πολλαπλών υπεράριθμων τμημάτων (όσο και αν ανέβει το όριο διαθέσιμων θέσεων).

<p>Ύπαρξη εκπαιδευτικού βοηθητικού υλικού, ανεβασμένο στο Uniwa Eclass, με σκοπό την μελέτη των φοιτητών που αποτελούν κρούσματα και βρίσκονται σε καραντίνα.</p>	<p>1)Με αυτόν τον τρόπο, είναι δυνατή και η μελέτη από το σπίτι για κάθε φοιτητή που αποτελεί κρούσμα και βρίσκεται σε καραντίνα. 2)Μέσω της ευκολότερης και ταχύτερης εφαρμογής της καραντίνας στα κρούσματα, έχουμε και μείωση πιθανόν μελλοντικών κρουσμάτων και σχετική αύξηση της ασφάλειας και αξιοπιστίας του τμήματος Μηχανικών Πληροφορικής και Υπολογιστών, ως προς την εφαρμογή των μέτρων και οδηγιών των ειδικών.</p>	<p>1)Πιθανόν να παρουσιαστούν, σημαντικές δαπάνες σε κόστος, με σκοπό την ενημέρωση των υποδομών του τμήματος για την παροχή μεγαλύτερης χωρητικότητας αλλά και ταχύτητας επεξεργασίας των δεδομένων στο Eclass, όπως στο ανέβασμα και κατέβασμα διαφανειών και άλλου διαθέσιμου εκπαιδευτικού υλικού. 2)Σε περίπτωση που παρουσιαστούν ορισμένα τεχνικά προβλήματα, αυτό θα έχει ως αποτέλεσμα την μη δυνατή πρόσβαση των χρηστών στην συγκεκριμένη υπηρεσία, από κάποιες ώρες μέχρι και μέρες σε πολύ ακραίες καταστάσεις. 3)Μπορεί να υπάρξουν και ορισμένες χρονικές δαπάνες, αφού το διδακτικό προσωπικό θα πρέπει συνεχώς να ενημερώνει το υλικό που ανεβάζει στην συγκεκριμένη πλατφόρμα.</p>
<p>Αυστηρή τήρηση όλων των υγειονομικών κανονισμών από την πολιτεία καθώς και ύπαρξη όλων των απαραίτητων ειδών υγιεινής στις τουαλέτες (αντισηπτικό, σαπούνι κλπ.) αλλά και γενικότερα στους διάφορους χώρους του Πανεπιστημίου.</p>	<p>1)Πιθανή μείωση μελλοντικών κρουσμάτων αλλά και βελτίωση του ποσοστού προστασίας απέναντι στην διασπορά του Covid-19. 2)Μείωση του ποσοστού των κρουσμάτων ως προς τους φοιτητές και το προσωπικό του τμήματος Μηχανικών Πληροφορικής και Υπολογιστών.</p>	<p>1)Απαραίτητη προϋπόθεση, η ύπαρξη του κατάλληλου χρονικού πλαισίου, για την σωστή και ανά τακτά χρονικά διαστήματα ενημέρωση και ευαισθητοποίηση των φοιτητών για να τηρούνται τα ανάλογα μέτρα όσο το δυνατόν καλύτερα. 2)Ίσως υπάρξουν και ορισμένες χρηματικές δαπάνες, για τον λόγο ότι θα είναι απαραίτητη η αγορά ορισμένων υλικών ή και ολόκληρου εξοπλισμού (π.χ. συσκευές αυτόματου ψεκασμού του αντισηπτικού υγρού), πέρα από την τήρηση των υγειονομικών κανονισμών για την προστασία των φοιτητών από την διασπορά του ιού.</p>
<p>Αύξηση του ορίου διαθέσιμων θέσεων στα διάφορα εργαστηριακά τμήματα για την μείωση των υπεράριθμων φοιτητών.</p>	<p>1)Μείωση των υπεράριθμων φοιτητών. 2)Καλύτερη διαχείριση των υπεράριθμων και πιθανή ένταξη τους στα είδη υπάρχοντα ή και σε νέα εργαστηριακά τμήματα.</p>	<p>1)Απαραίτητη προϋπόθεση και πάλι, να υπάρχουν οι αναγκαίες υποδομές αλλά και ο κατάλληλος εξοπλισμός για να το στηρίξουν αυτό. 2)Συνεπώς, απαιτείται και το ανάλογο κόστος για να το υποστηρίξουμε όλο αυτό.</p>
<p>Εφαρμογή Rapid Test, ακόμα και σε φοιτητές που έχουν εμβολιαστεί πλήρως, βάση των απαραίτητων δόσεων του εμβολίου όπως έχει ορίσει και η πολιτεία.</p>	<p>1)Μείωση της διασποράς και της ύπαρξης πιθανόν μελλοντικών κρουσμάτων και ιδιαίτερα των ασυμπτωματικών.</p>	<p>1)Σημαντικές χρονικές δαπάνες, ως προς τους διάφορους ελέγχους που θα είναι απαραίτητοι κατά την είσοδο των φοιτητών στους διάφορους χώρους του Πανεπιστημίου Αυτικής Αττικής. 2)Επιβάρυνση οικονομική ως προς τους φοιτητές, για την διεξαγωγή των Rapid ανά τακτά χρονικά διαστήματα.</p>

<p>Διάρθρωση του θεωρητικού μέρους κάθε μαθήματος, σε πολλαπλά εβδομαδιαία τμήματα με λιγότερο πλήθος φοιτητών.</p>	<p>1)Μερική αποσυμφόρηση των αιθουσών, λόγω ύπαρξης μικρότερου αριθμού φοιτητών στις θεωρητικές διαλέξεις. 2)Καλύτερη αφομοίωση της ύλης και πιο ξεκούραστη διδασκαλία, εφόσον και οι φοιτητές που παρακολουθούν είναι λιγότεροι και άρα πιο εύκολα διαχειρίσιμοι ως προς τις απορίες τους. 3)Καλύτερη παρακολούθηση, χωρίς να υπάρχει ιδιαίτερη φασαρία λόγω πολυκοσμίας ή και δυσφορία ως προς τον χώρο και την επιλογή θέσης. 4)Πιθανή σταδιακή μείωση και των κρουσμάτων, λόγω του ότι τα θεωρητικά τμήματα είναι αρκετά μικρότερα σε πλήθος φοιτητών από τα αρχικά ενιαία τμήματα της θεωρίας και πιο εύκολα διαχειρίσιμα ως προς την τήρηση των οδηγιών και των μέτρων προστασίας από τον Covid-19. 5)Σημαντική εξοικονόμηση χρόνου από την μείωση των ελέγχων που απαιτούνται στα απαραίτητα πιστοποιητικά κατά την είσοδο στους χώρους του τμήματος. 6)Ταχύτερη προσέλευση φοιτητών ως προς τα θεωρητικά μέρη των μαθημάτων.</p>	<p>1)Υπάρχει περίπτωση να μην καλυφθεί πλήρως η ύλη, λόγω μειωμένων ωρών διδασκαλίας, αναγκαστικά βάζει πάντα του εβδομαδιαίου ωρολογίου προγράμματος.</p>
<p>Καλύτερη αντιμετώπιση των πιθανών μελλοντικών κρουσμάτων, τα οποία εμφανίζονται κατά την καθημερινή λειτουργία της σχολής στο τμήμα Μηχανικών Πληροφορικής και Υπολογιστών.</p>	<p>1)Η έγκαιρη για παράδειγμα εφαρμογή της καραντίνας σε επιβεβαιωμένα κρούσματα, συμβάλει στην προστασία τόσο του ίδιου του κρούσματος από να χειροτερεύσει η κατάσταση της υγείας του αλλά και των υπολοίπων, από την περαιτέρω διασπορά του ιού στους ίδιους.</p>	<p>1)Απαραίτητη προϋπόθεση, η ύπαρξη του κατάλληλου χρονικού πλαισίου, για την σωστή και ανά τακτά χρονικά διαστήματα ενημέρωση και ευαισθητοποίηση του διδακτικού προσωπικού αλλά και των φοιτητών για την ανάλογη αντιμετώπιση σε περίπτωση κρούσματος ή και πολλαπλών κρουσμάτων. 2)Απαιτείται η ύπαρξη και του αντίστοιχου εκπαιδευτικού υλικού στην σελίδα του Eclass ή και σε οποιαδήποτε άλλο ηλεκτρονικό μέσο κρίνει απαραίτητο το τμήμα Μηχανικών Πληροφορικής και Υπολογιστών για την εξ' αποστάσεως φοίτηση. 3)Αν μιλάμε για μεγάλη ποσότητα πληροφορίας βάση της ύλης σε ένα μάθημα, μιλάμε και για εξοπλισμό μεγάλων δυνατοτήτων και συνεπώς για σημαντικές οικονομικές δαπάνες για την ενημέρωση του εξοπλισμού που διαθέτει το τμήμα, για την βελτίωση της ποιότητας των υπηρεσιών του, μέσω του Eclass ή και οποιαδήποτε άλλης σελίδας είναι στην διάθεση του φοιτητή. 4)Σε περίπτωση που παρουσιαστούν ορισμένα τεχνικά προβλήματα, αυτό θα έχει ως αποτέλεσμα την μη δυνατή πρόσβαση των χρηστών στην συγκεκριμένη υπηρεσία, από κάποιες ώρες μέχρι και μέρες σε πολύ ακραίες καταστάσεις.</p>

		5)Μπορεί να υπάρξουν και ορισμένες χρονικές δαπάνες, αφού το διδακτικό προσωπικό θα πρέπει συνεχώς να ενημερώνει το υλικό που ανεβάζει στον συγκεκριμένο ιστότοπο.
--	--	--

Πίνακας 2 - Μελέτη Προτάσεων Φοιτητών ως προς τα Πλεονεκτήματα/Μειονεκτήματα τους.

2.2 Ανάλυση και Σύγκριση Διαθέσιμων Λύσεων.

Συνεπώς, μέσα από την λεπτομερή ανάλυση των πληροφοριών που παρουσιάζονται στον παραπάνω πίνακα, βάση των λύσεων που προέκυψαν και σε συνδυασμό με τα πλεονεκτήματα αλλά και τα μειονεκτήματα τους, εύκολα κανείς καταλήγει στα παρακάτω συμπεράσματα:

1. Όσον αφορά την περίπτωση ύπαρξης υβριδικής εκπαίδευσης, πιθανόν να μπορούσε να γίνει κάτι σαν υπαρκτό σενάριο, μόνο με την προϋπόθεση ότι δεν θα υπάρξουν σημαντικές τεχνικές βλάβες ή και προβλήματα, με σκοπό την διατήρηση της συνέπειας και της ακεραιότητας των υπηρεσιών του συστήματος προς τους φοιτητές του τμήματος.
2. Η λύση που αφορά το να γίνει πρόταση εφαρμογής ανώτερου ορίου χωρητικότητας και συνεπώς αύξησης των θέσεων όσον αφορά τα εργαστηριακά τμήματα δεν είναι εφικτή. Ο λόγος για αυτό, είναι ότι δεν υφίστανται το να υπάρξει μεγαλύτερος αριθμός διαθέσιμων θέσεων στα είδη υπάρχοντα εργαστηριακά τμήματα, από την στιγμή που δεν υπάρχουν και οι ανάλογες υποδομές και εγκαταστάσεις συμπεριλαμβανομένου του εξοπλισμού για να υποστηριχθεί κάτι τέτοιο. Θα απαιτούνταν μεγάλες σχετικά οικονομικές δαπάνες για να γίνει κάτι τέτοιο και δεν υπάρχει λόγος να γίνει μια τέτοια κίνηση στην παρούσα φάση που διανύει το τμήμα αλλά και η χώρα. Και αυτός είναι και ο κύριος λόγος που δεν μπορεί να εφαρμοστεί πλήρως ούτε η πρόταση για διαίρεση των υπάρχων τμημάτων σε μικρότερα, για τις θεωρητικές διαλέξεις του τμήματος.
3. Η ιδέα της προσθήκης βοηθητικού εκπαιδευτικού υλικού, το οποίο υπήρξε και κατά την περίοδο της καραντίνας αναρτημένο στο site της σχολής, θα μπορούσε να εφαρμοστεί ως μια πιθανή λύση του προβλήματος της καλύτερης αντιμετώπισης των μελλοντικών κρουσμάτων αλλά και της μελέτης από το σπίτι. Έτσι, συνδυαστικά θα μπορούσε να εφαρμοστεί και το σκεπτικό της καλύτερης αντιμετώπισης της ύπαρξης κρουσμάτων αλλά και της καλύτερης αντιμετώπισης της διασποράς του ιού στους χώρους του τμήματος.
4. Η καλύτερη τήρηση των διάφορων κανονισμών και οδηγιών της πολιτείας κατά της διασποράς του **Covid-19** αλλά και η ύπαρξη των διάφορων απαραίτητων ειδών υγιεινής για την καλύτερη δυνατή απολύμανση όπως για παράδειγμα αντισηπτικών στις τουαλέτες, θα μπορούσε να εφαρμοστεί σχετικά εύκολα. Αρκετά σημαντικό για την καλύτερη εφαρμογή αυτής της λύσης, θα ήταν η καλή ενημέρωση αλλά και η επιβολή κυρώσεων σε όσους δεν συμμορφώνονται με τους νέους κανονισμούς του τμήματος. Και όσον αφορά τα υλικά, το κόστος ίσως να αυξηθεί ελάχιστα, παρόλα αυτά δεν είναι κάτι το απρόσιτο η αγορά και κατά συνέπεια η χρήση τους από τους φοιτητές.
5. Ο έλεγχος με «Rapid Test» ακόμα και στους εμβολιασμένους φοιτητές θα ήταν μια αρκετά καλή ιδέα, για την αντιμετώπιση της διασποράς του **Covid-19** και συγκεκριμένα όσον αφορά τις περιπτώσεις ασυμπτωματικών κρουσμάτων. Απλώς, το μόνο ίσως αρνητικό που μπορεί να επιφέρει κάτι τέτοιο είναι οι χρονικές και οικονομικές δαπάνες, τόσο στο κομμάτι των ελέγχων αλλά και στα

χρήματα που θα πρέπει να καταβάλουν καθημερινά στα φαρμακεία οι φοιτητές αλλά και το προσωπικό του τμήματος για την λήψη των τεστ αυτών.

2.3 Καλύτερη Βέλτιστη Λύση και Συμπεράσματα Μελέτης των Φοιτητών.

Οι τρεις από τις λύσεις που προκύπτουν ως βέλτιστες για την βελτίωση της προαναφερθέντας κατάστασης στο τμήμα είναι οι παρακάτω:

- Η ύπαρξη βοηθητικού εκπαιδευτικού υλικού αναρτημένου στον Ιστότοπο του Πανεπιστημίου, για την κάλυψη των αναγκών όσων φοιτούν από το σπίτι και απομακρυσμένα λόγω καραντίνας.
- Η εφαρμογή rapid test ακόμα και στην περίπτωση των πλήρη εμβολιασμένων ατόμων, για την καλύτερη δυνατή αντιμετώπιση της διασποράς και ειδικότερα των ασυμπτωματικών κρουσμάτων του **Covid-19**.
- Και τέλος, η καλύτερη τήρηση των υγειονομικών μέτρων και κανονισμών που έχει ορίσει το κράτος καθώς και η ύπαρξη όλων των απαραίτητων ειδών υγιεινής στους χώρους του τμήματος για τους φοιτητές.
- Συμπληρωματικά, πρέπει να γίνει σαφές ότι αναφερόμαστε στην ύπαρξη αναρτημένου βοηθητικού εκπαιδευτικού υλικού προς τους φοιτητές. Μαζί με αυτό το μέτρο γίνεται συνδυαστικά και η εφαρμογή της καλύτερης δυνατής αλλά και έγκαιρης αντιμετώπισης των κρουσμάτων που προκύπτουν μεταξύ των φοιτητών του τμήματος αλλά και του εκπαιδευτικού προσωπικού.

Βέβαια, αν το σκεφτεί κανείς καλύτερα, αυτά τα τρία μέτρα είδη έχουν εφαρμοστεί και εφαρμόζονται από το τμήμα στην καθημερινή του λειτουργία. Συνεπώς, αυτό που είναι αρκετά σημαντικό στην παρούσα φάση, είναι η εύρεση μιας όσο το δυνατόν πιο καλής λύσης, που να είναι σε θέση να απαλύνει την δυσφορία που παρουσιάζεται συγκεκριμένα κατά τις εξεταστικές περιόδους του τμήματος και η οποία να έχει και μια καινοτομία τεχνολογικά μέσα της συμβαδίζοντας και με τις τάσεις της σημερινής εποχής. Με βάση τα παραπάνω δεδομένα, προκύπτει και η «γέννηση» ενός συστήματος με τίτλο «**UNIWA EXAM PLANNER**». Το οποίο θα είναι υπεύθυνο για την καλύτερη διαχείριση του όγκου των φοιτητών και την καλύτερη δυνατή κατανομή τους, στις διαθέσιμες θέσεις των αιθουσών του τμήματος κατά τις εξεταστικές περιόδους του. Το συγκεκριμένο σύστημα, αναλύεται περαιτέρω στο επόμενο κεφάλαιο που ακολουθεί.

Κεφάλαιο 3^ο – Ανάλυση και Σχεδίαση Αρχιτεκτονικής Συστήματος «UNIWA EXAM PLANNER».

3.1 Γενική Περιγραφή Συστήματος.

Το σύστημα το οποίο θα αποτελέσει την λύση στην παρούσα κατάσταση των εξεταστικών περιόδων του τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής θα είναι το «**UNIWA EXAM PLANNER**». Ουσιαστικά, αυτό θα είναι μια διαδικτυακή πλατφόρμα, μέσα από την οποία και αναλόγως τον ρόλο του χρήστη θα του παρέχονται και οι αντίστοιχες υπηρεσίες του συστήματος. Αν ο χρήστης για παράδειγμα, αποτελεί μέλος της Γραμματείας του τμήματος, αυτό σημαίνει ότι θα είναι και υπεύθυνος για την ενημέρωση του με τα αντίστοιχα στοιχεία της επερχόμενης εξεταστικής περιόδου καθώς και ορισμένες χρήσιμες πληροφορίες σε σχέση με τα χρονοδιαγράμματα που αφορούν την διαθεσιμότητα των αιθουσών. Αν από την άλλη ο χρήστης είναι Καθηγητής, τότε η χρήση που θα κάνει ο συγκεκριμένος, θα είναι καθαρά ως παρατηρητής της εξέλιξης των εγγραφών των φοιτητών στα προς εξέταση μαθήματα που έχει στην διάθεση του. Και τέλος, αν ο χρήστης είναι Φοιτητής του τμήματος, ο οποίος και αποτελεί τον κύριο χρήστη του συστήματος, οι υπηρεσίες που θα του παρέχονται από αυτό θα έχουν ως στόχο την εγγραφή του σε ένα μάθημα προς εξέταση, την αναίρεση μιας εγγραφής σε περίπτωση που κάτι έχει αλλάξει καθώς και την προβολή του συνόλου των δηλώσεων του με τις ανάλογες πληροφορίες που θα του φανούν χρήσιμες για την ημέρα των εξετάσεων (πχ την αίθουσα, την θέση του όπως και την ώρα προσέλευσης). Για την πρόσβαση του οποιαδήποτε χρήστη στο σύστημα, θα πρέπει πρώτα να γίνεται πρόσβαση μέσω των ανάλογων στοιχείων πρόσβασης που θα έχει ο καθένας εξαρχής στην διάθεση του.

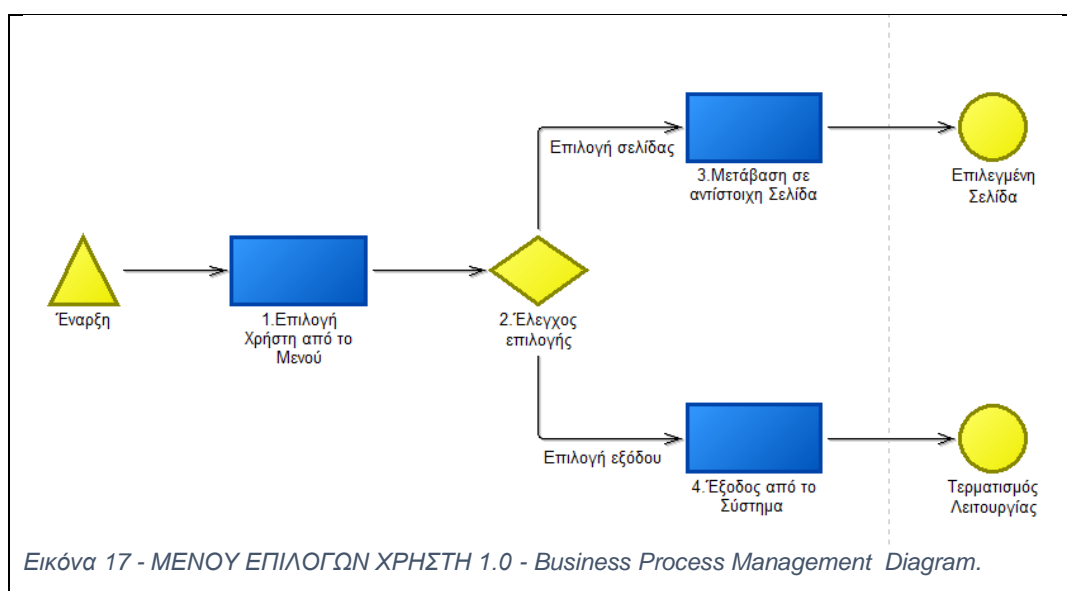
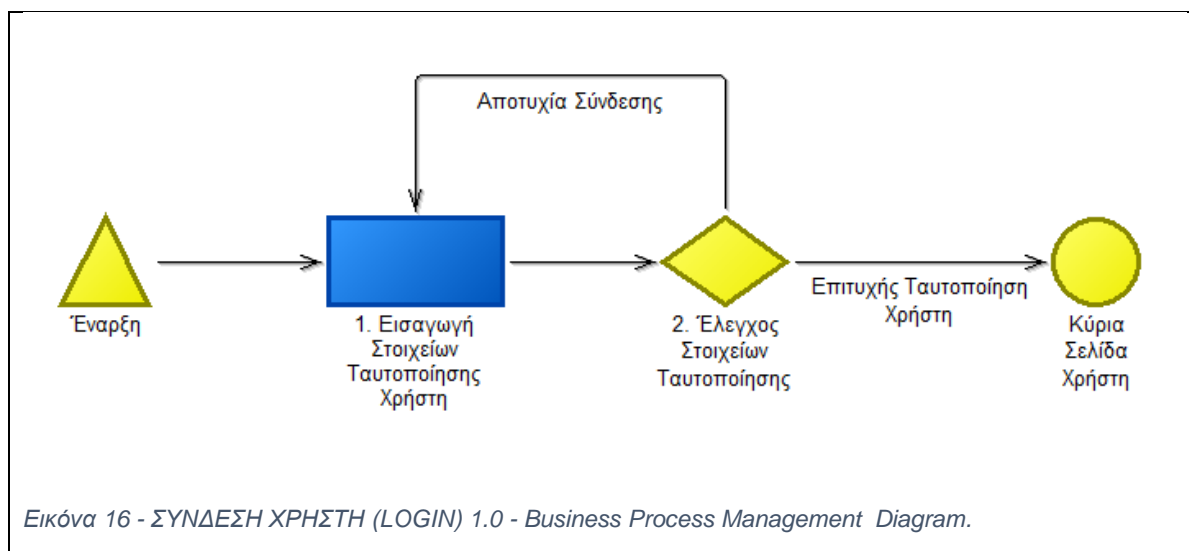
Ως κύριοι στόχοι του συστήματος, είναι οι εξής:

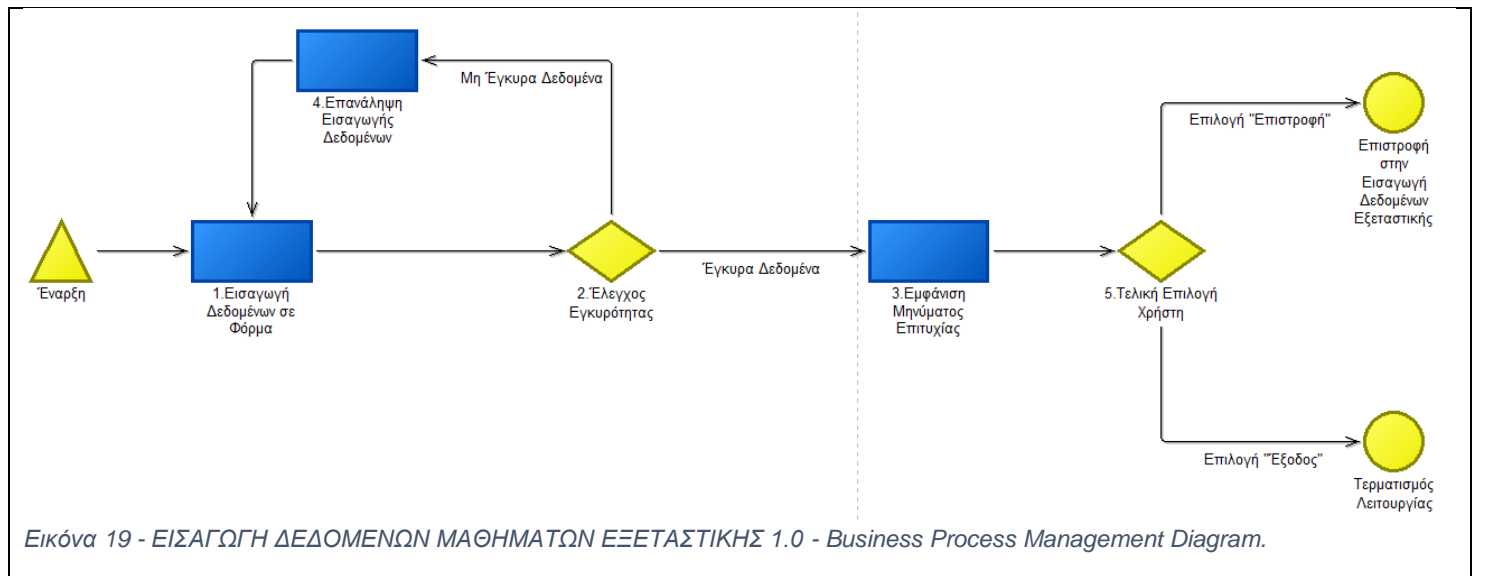
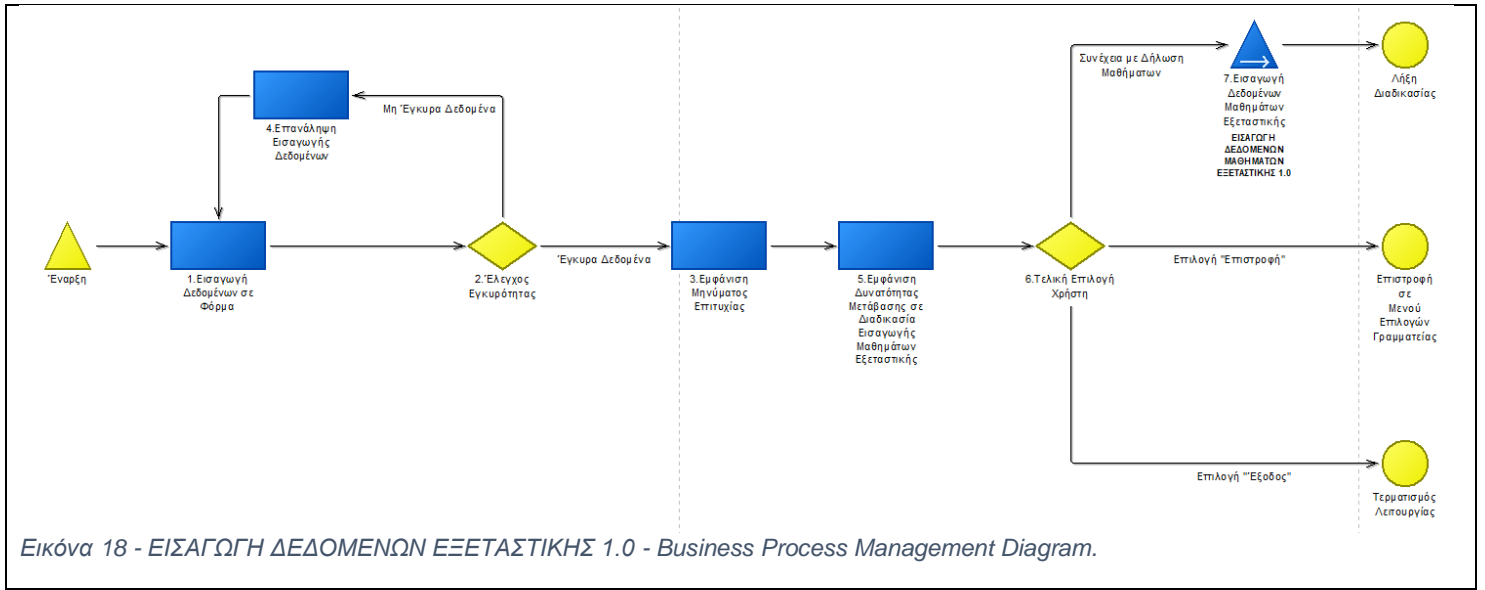
- Εξαρχής, η βελτιστοποίηση της διαδικασίας εξέτασης μαθήματος ως προς την οργάνωση της διάθεσης των υποδομών αλλά και του προσωπικού του τμήματος για τους φοιτητές.
- Έπειτα, η καλύτερη δυνατή τήρηση των μέτρων προστασίας από τον **Covid19**, μέσα από την διαχείριση του πλήθους των φοιτητών σε κάθε εξεταζόμενο μάθημα.
- Στην συνέχεια, η ταχύτερη εφαρμογή των απαραίτητων ελέγχων ταυτοποίησης καθώς και ύπαρξης των απαραίτητων δικαιολογητικών βάση κανονισμών της πολιτείας για την αντιμετώπιση εξάπλωσης του Ιού.
- Τέλος, η εξυπηρέτηση ακόμα και των φοιτητών και προσωπικού που ανήκουν σε ευπαθείς ομάδες, στους οποίους απαιτείται ιδιαίτερος βαθμός προστασίας απέναντι τους.

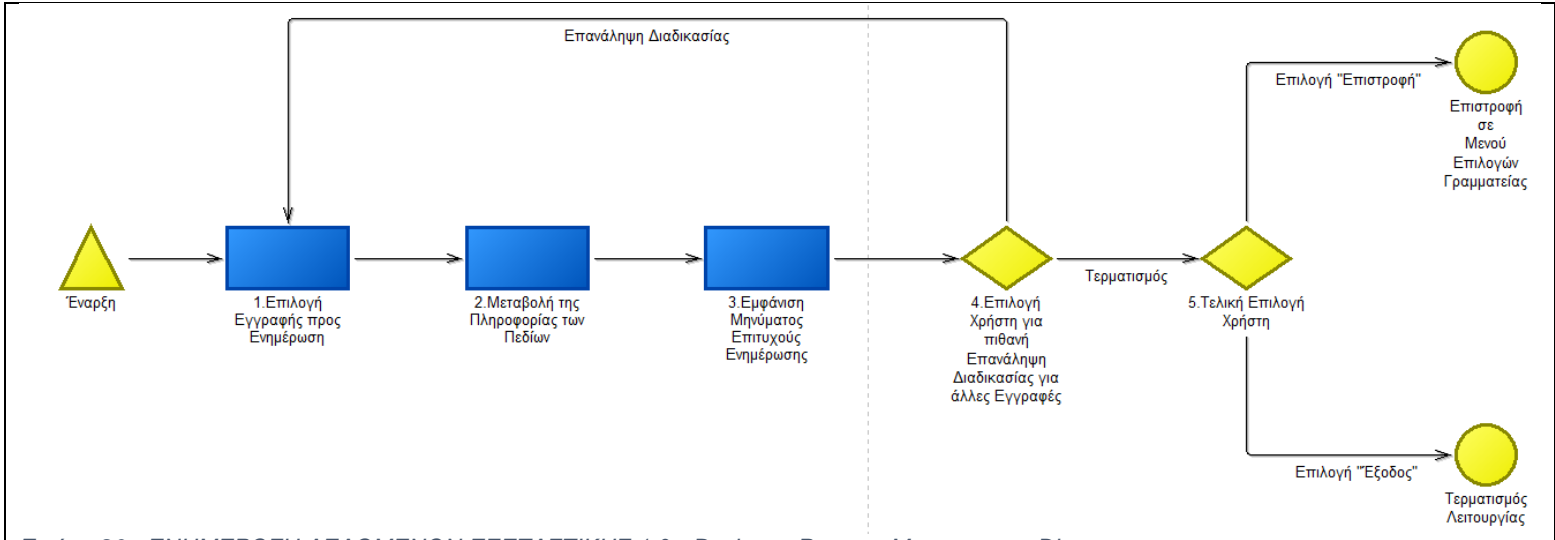
Το σύστημα θα αποτελείται από ένα **Backend** αλλά και ένα **Frontend** τμήμα που στην ουσία θα είναι υπεύθυνα για την κάλυψη της λειτουργικότητας αλλά και του γραφικού του περιβάλλοντος προς τον χρήστη και θα είναι κωδικοποιημένα μέσω των τεχνολογιών **Vaadin Designer** και **Spring Boot Web Services** αντίστοιχα. Επίσης, θα υπάρξει και μια Βάση Δεδομένων σε γλώσσα **MySQL**, μέσω της οποίας θα γίνεται η αποθήκευση αλλά και η μεταφορά της πληροφορίας από και προς τον τελικό χρήστη μέσω του συστήματος. Για την παραγωγή του συστήματος θα γίνει χρήση της γλώσσας προγραμματισμού **Java**, όπως και του περιβάλλοντος του εργαλείου **Intellij Idea IDE**.

3.2 Διαγραμματική Απεικόνιση της Αρχιτεκτονικής του Συστήματος.

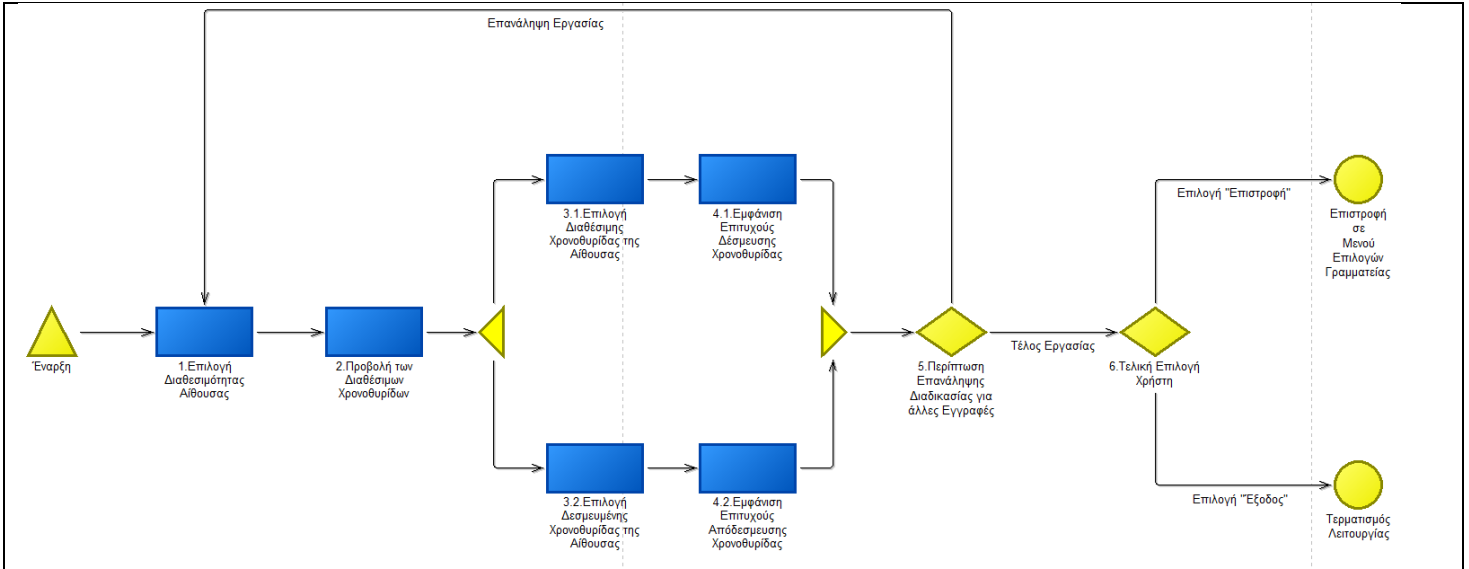
Ως προς το κομμάτι της ανάλυσης και σχεδίασης της αρχιτεκτονικής του συστήματος, έγινε χρήση του **ADONIS - Business Process Management Toolkit** και με την βοήθεια του περιβάλλοντος αλλά και των δυνατοτήτων αυτού του εργαλείου, κατασκευάστηκαν ορισμένα **B.P.M.** διαγράμματα (**Business Process Management diagrams**). Αυτό έγινε με σκοπό την αναλυτική και καλύτερη απεικόνιση των διαδικασιών του συστήματος «**UNIWA EXAM PLANNER**». Τα διαγράμματα αυτά ακολουθούν στην συνέχεια, καθώς επίσης στην επόμενη ενότητα 3.3 υπάρχει και η λεκτική ανάλυση τους σε μορφή δομής πίνακα.



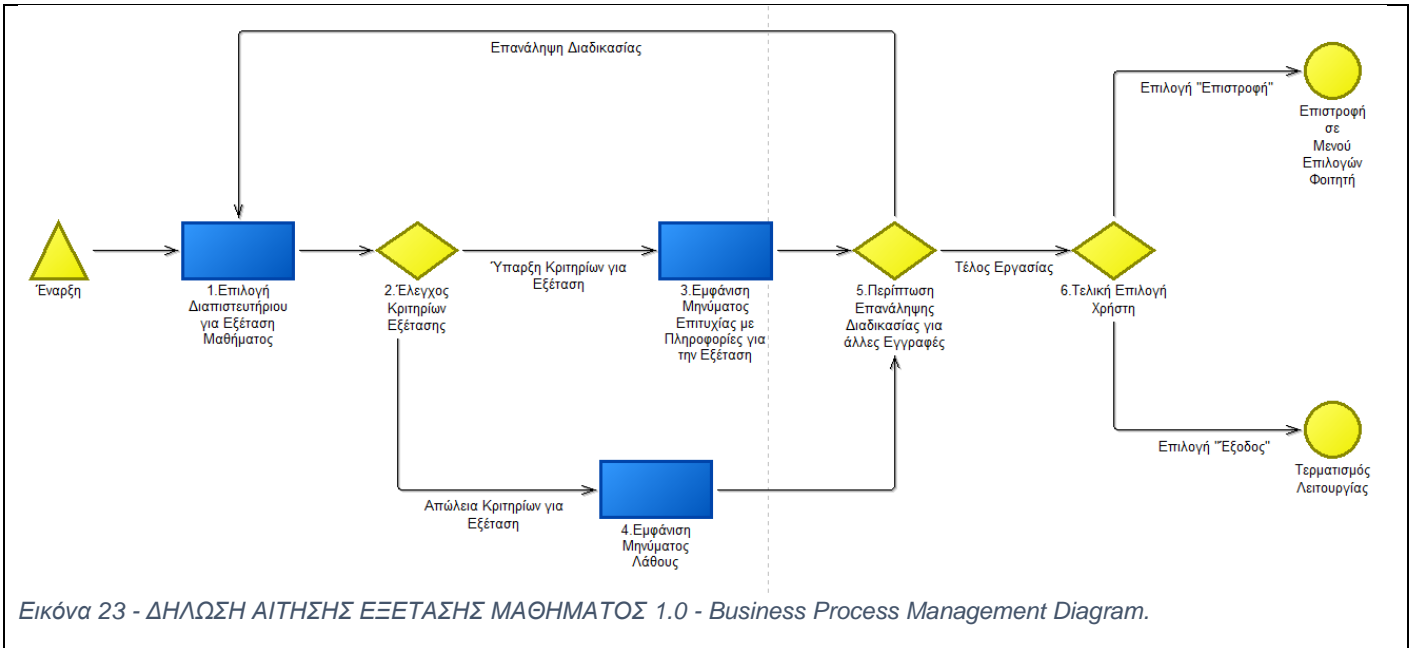
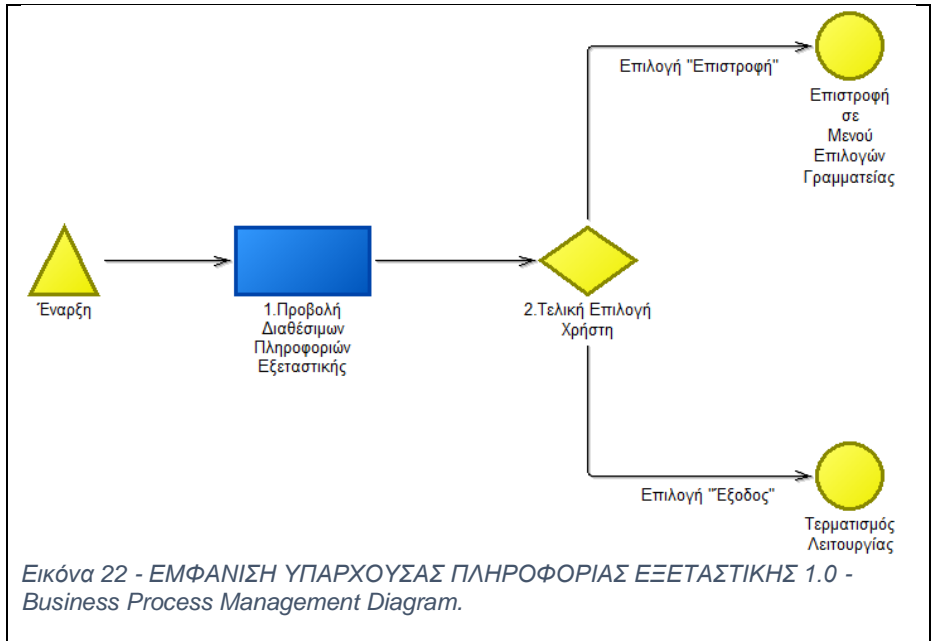


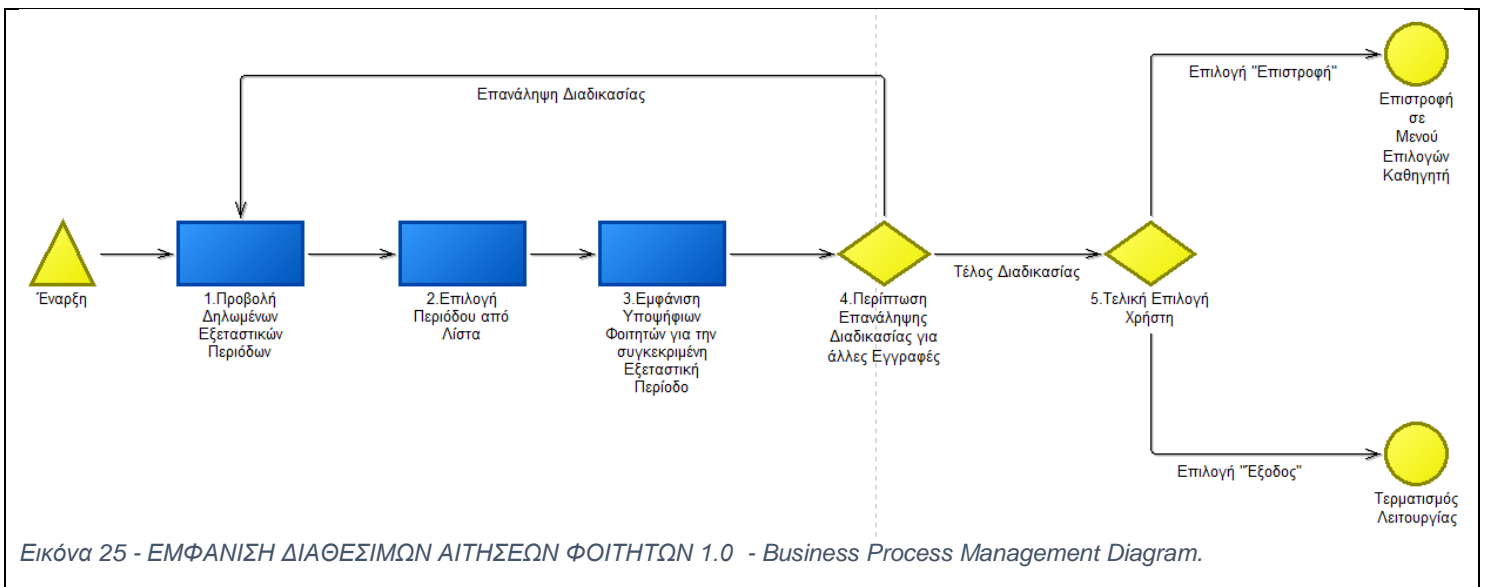
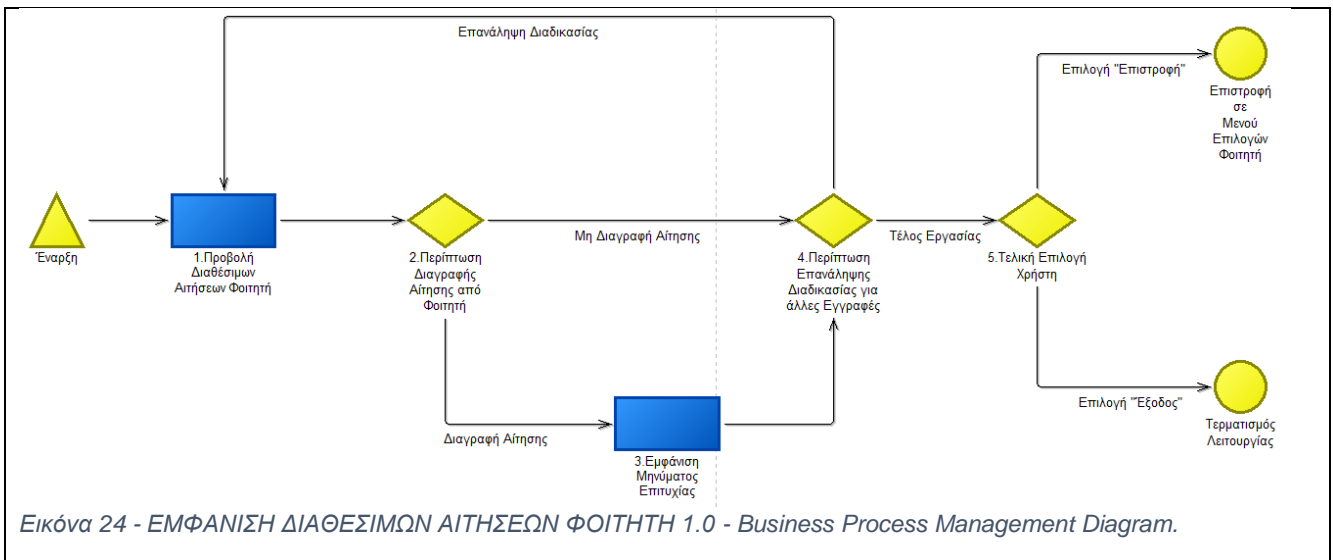


Εικόνα 20 - ΕΝΗΜΕΡΩΣΗ ΔΕΔΟΜΕΝΩΝ ΕΞΕΤΑΣΤΙΚΗΣ 1.0 - Business Process Management Diagram.

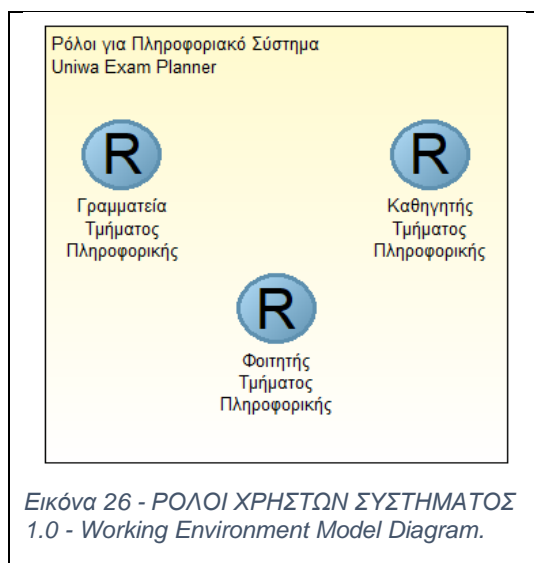


Εικόνα 21 - ΟΡΙΣΜΟΣ ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ ΑΙΘΟΥΣΩΝ 1.0 - Business Process Management Diagram.





Συμπληρωματικά, για την περιγραφή των ρόλων που έχουν οι χρήστες στο συγκεκριμένο σύστημα, έγινε η δημιουργία και του παρακάτω **W.E.M. Διαγράμματος (Working Environment Model Diagram)**.



3.3 Ανάλυση της Λειτουργίας του Συστήματος σε επιμέρους Διαδικασίες.

Το σύστημα «**UNIWA EXAM PLANNER**» διαθέτει μια λειτουργικότητα, η οποία αναλύεται σε επιμέρους διαδικασίες, οι οποίες είναι πολύ εύκολα ορατές στον πίνακα που ακολουθεί, βάση πάντα και των διαγραμμάτων που παρουσιάστηκαν στο προηγούμενο υποκεφάλαιο.

Διαδικασία	Χρήστες	Δεδομένα Εισόδου	Δεδομένα Εξόδου	Λειτουργία
Σύνδεση στο Σύστημα (Login).	Γραμματεία, Φοιτητής και Καθηγητής.	Όνομα Χρήστη (username) και Κωδικός Πρόσβασης (password).	Αρχική Σελίδα βάση του Ρόλου του Χρήστη ή Μήνυμα Λάθους και επιστροφή στην αρχική σελίδα εισόδου του συστήματος.	Η συγκεκριμένη διαδικασία, δέχεται τα στοιχεία εισόδου του χρήστη και βάση αυτά και της ιδιότητας του, ψάχνει να βρει αν υπάρχει αντιστοίχιση με την βάση δεδομένων του συστήματος. Εφόσον κάτι τέτοιο ισχύει, παρέχεται η πρόσβαση στο σύστημα, διαφορετικά απορρίπτεται η πρόσβαση και γίνεται εμφάνιση ενός μηνύματος σφάλματος στην οθόνη του χρήστη.
Μενού Επιλογών (Main Page).	Γραμματεία, Φοιτητής και Καθηγητής.	Η επιλογή του χρήστη.	Αντίστοιχη επιλεγμένη σελίδα.	Η συγκεκριμένη διαδικασία, αποτελεί μια αρκετά απλή διαδικασία, με βάση την οποία ο χρήστης διαθέτει ένα συγκεκριμένο αριθμό επιλογών και μέσα από την επιλογή του, μεταβαίνει και στις αντίστοιχες υπηρεσίες του συστήματος.

Δήλωση Εξεταστικής στο Σύστημα (Apply Exam).	Γραμματεία	Εισερχόμενη πληροφορία για την επερχόμενη εξεταστική περίοδο.	Μήνυμα Επιτυχίας ή Αποτυχίας της Εισόδου των Δεδομένων στο Σύστημα.	Σε αυτήν την διαδικασία, ο χρήστης αφού συμπληρώσει μια φόρμα και τα δεδομένα είναι σωστά δομημένα και έγκυρα για το σύστημα, εισάγονται και του επιστρέφεται ένα μήνυμα επιτυχίας, διαφορετικά παίρνει μια ένδειξη λανθασμένης καταχώρησης τους. Στην συνέχεια και αφού δηλαδή ο χρήστης, έχει κάνει τουλάχιστον μια επιτυχής καταχώρηση εξεταστικής στο σύστημα, του δίνεται η δυνατότητα να μεταβεί και στην υποδιαδικασία, για την δήλωση μαθημάτων στην εξεταστική. Συμπληρωματικά, όπως και σε οποιαδήποτε άλλη διαδικασία του συστήματος μας, υπάρχει η δυνατότητα επιστροφής στο αρχικό μενού ή εξόδου από το σύστημα μέσα από τις αντίστοιχες επιλογές.
Ενημέρωση Παραμέτρων Εξεταστικής (Update Exam).	Γραμματεία	Τα νέα δεδομένα στα πεδία της εγγραφής για την εκάστοτε εξεταστική.	Μήνυμα επιτυχούς ενημέρωσης της εξεταστικής.	Εδώ ο χρήστης επιλέγει την εκάστοτε εξεταστική που επιθυμεί, της μεταβάλλει τα πεδία και επιλέγει αποθήκευση παίρνοντας πίσω και το αντίστοιχο μήνυμα επιτυχούς αποθήκευσης των νέων δεδομένων. Εννοείται ότι και εδώ υπάρχει η δυνατότητα επιστροφής στο αρχικό μενού ή εξόδου από το σύστημα.
Δέσμευση Αίθουσών Εξεταστικής (Assign Classroom).	Γραμματεία	Επιλεγμένη Χρονοθυρίδα.	Μήνυμα επιτυχούς δέσμευσης ή αποδέσμευσης της χρονοθυρίδας της εκάστοτε αίθουσας.	Στην παρούσα διαδικασία, ο χρήστης επιλέγει μια αίθουσα, βλέπει ποιες είναι οι διαθέσιμες χρονοθυρίδες της και ποιες οι είδη πιθανόν δεσμευμένες και επιλέγει αναλόγως τις αντίστοιχες χρονοθυρίδες με σκοπό την δέσμευση ή αποδέσμευση τους. Στο τέλος, λαμβάνει και την αντίστοιχη ένδειξη από το σύστημα για την επιτυχή ολοκλήρωση της διαδικασίας. Εννοείται ότι και εδώ υπάρχει η δυνατότητα επιστροφής στο αρχικό μενού ή εξόδου από το σύστημα.
Παρουσίαση Πληροφοριών Εξεταστικής (Exam View).	Γραμματεία	-	Όλες οι δηλωμένες εξεταστικές του συστήματος μαζί με τις αίθουσες και τις δεσμευμένες χρονοθυρίδες τους.	Αυτή η διαδικασία, είναι σε θέση να προβάλει στον χρήστη όλα τα δεδομένα σε σχέση με τις είδη δηλωμένες εξεταστικές περιόδους καθώς και τις δεσμευμένες χρονοθυρίδες μαζί με τις αίθουσες τους από τον χρήστη Γραμματεία. Εννοείται ότι και εδώ όπως και πριν, υπάρχει η δυνατότητα επιστροφής στο αρχικό μενού ή εξόδου από το σύστημα.

Δήλωση Μαθημάτων προς Εξέταση (Apply Exam Lesson).	Φοιτητής	Επιλεγμένο Διαπιστευτήριο από τον Φοιτητή για το εκάστοτε μάθημα.	Μήνυμα επιτυχούς εγγραφής, μαζί με πληροφορίες που αφορούν την εξέταση ή ένδειξη λάθους.	Εδώ ο χρήστης Φοιτητής, επιλέγει από το μάθημα που επιθυμεί να δηλώσει για εξέταση το αντίστοιχο Διαπιστευτήριο και αναλόγως αν το δικαιούται βάση κριτηρίων να εξετασθεί ή όχι, παίρνει και την αντίστοιχη ένδειξη από το σύστημα. Σε περίπτωση επιτυχούς δήλωσης, ο χρήστης μαζί με την αντίστοιχη ένδειξη δέχεται και τις πληροφορίες για την εκάστοτε εξέταση (πχ Καθηγητής, Ώρα/Μέρα εξέτασης και Αίθουσα). Και πάντα υπάρχει η δυνατότητα επιστροφής στο αρχικό μενού ή εξόδου από το σύστημα.
Παρουσίαση Πληροφοριών και πιθανή Ακύρωση Εγγεγραμμένων Μαθημάτων Εξεταστικής (Exam Lesson View).	Φοιτητής	-	Καρτέλα μαθημάτων εξεταστικής που έχει δηλώσει ο εκάστοτε Φοιτητής. Μήνυμα επιτυχίας σε περίπτωση επιλογής ακύρωσης μιας εγγραφής από το σύστημα.	Στην παρούσα διαδικασία, ο χρήστης Φοιτητής έχει την δυνατότητα να βλέπει όλες τις δηλώσεις εξετάσεων που έχει πραγματοποιήσει καθώς και την δυνατότητα να ακυρώσει μια αίτηση του αν το επιθυμεί. Σε οποιαδήποτε περίπτωση ο χρήστης με την ακύρωση μιας εγγραφής, θα λάβει και αντίστοιχη ειδοποίηση από το σύστημα για την ολοκλήρωση της συγκεκριμένης διαδικασίας. Και πάντα υπάρχει η δυνατότητα επιστροφής στο αρχικό μενού ή εξόδου από το σύστημα.
Παρουσίαση Υποψήφιων στις Εξετάσεις ανά Μάθημα Καθηγητή (Exam Lesson Students View).	Καθηγητής	-	Καρτέλα με τις εξεταστικές περιόδους που αφορούν τον συγκεκριμένο Καθηγητή καθώς και τους Φοιτητές που είναι δηλωμένοι ανά περίοδο.	Σε αυτήν την διαδικασία ο χρήστης Καθηγητής έχει μια λίστα από τις εξεταστικές περιόδους που τον αφορούν και επιλέγοντας όποια επιθυμεί, μπορεί ανά πάσα στιγμή να δει και το ποιοι Φοιτητές θα συμμετάσχουν. Και πάντα υπάρχει η δυνατότητα επιστροφής στο αρχικό μενού ή εξόδου από το σύστημα.

Πίνακας 3 - Ανάλυση Συστήματος σε επί μέρους Διαδικασίες.

Συμπληρωματικά, παρακάτω ακολουθεί ένας πίνακας με τις υποδιαδικασίες του συστήματος.

Υποδιαδικασία	Κύρια Διαδικασία	Χρήστες	Δεδομένα Εισόδου	Δεδομένα Εξόδου	Λειτουργία
Δήλωση Μαθημάτων Εξεταστικής (Assign Exam Lessons).	Δήλωση Εξεταστικής στο Σύστημα (Apply Exam).	Γραμματεία	Πληροφορίες για το προς εξέταση μάθημα στην συγκεκριμένη εξεταστική περίοδο.	Μήνυμα Επιτυχίας ή Αποτυχίας ως προς την είσοδο των δεδομένων.	Στην παρούσα υποδιαδικασία, μπορούμε να μεταβούμε μόνο εφόσον υπάρχει τουλάχιστον μια εξεταστική καταχωρισμένη στο σύστημα μας, κατά προτίμηση η ίδια με τα μαθήματα που επιθυμούμε να δηλώσουμε στην συνέχεια (διαφορετικά υπάρχει περίπτωση σφάλματος). Αφού ο χρήστης Γραμματεία, εισάγει την απαραίτητη πληροφορία του στην φόρμα του συστήματος και γίνει έλεγχος εγκυρότητας, παράγεται και εμφανίζεται ένα μήνυμα επιτυχίας ή αποτυχίας. Συμπληρωματικά, εννοείται ότι υπάρχει η δυνατότητα επιστροφής στην κύρια διαδικασία ή και η δυνατότητα εξόδου από το σύστημα.

Πίνακας 4 - Ανάλυση Διαδικασιών σε Υποδιαδικασίες.

Κεφάλαιο 4^ο – Οι Τεχνολογίες Σχεδίασης και Ανάπτυξης του Συστήματος «UNIWA EXAM PLANNER».

4.1 Γενική Περιγραφή των Τεχνολογιών Συστήματος.

Οι τεχνολογίες που αξιοποιήθηκαν, για την αρχική μελέτη του αντικειμένου της σημερινής κατάστασης που παρουσιάζεται στο τμήμα Μηχανικών Πληροφορικής και Υπολογιστών της σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής προκύπτουν έπειτα από αρκετή έρευνα και αναζήτηση με σκοπό να υπάρχει και ένα τελικό αποτέλεσμα από αυτές, όσο το δυνατόν πιο προηγμένο τεχνολογικά αλλά και που να υποστηρίζει τα σημερινά πρότυπα και απαιτήσεις από την λειτουργία ενός πληροφοριακού συστήματος.

Η διεξαγωγή της απαραίτητης μελέτης της παρούσας κατάστασης και η γενικότερη εξόρυξη χρήσιμης πληροφορίας από τους φοιτητές του τμήματος για την καθημερινή λειτουργία του τμήματος έγινε μέσω του εργαλείου της **Google «Google Form»**. Μέσω αυτής της πλατφόρμας στήθηκε ένα ερωτηματολόγιο, το οποίο αφού συμπληρώθηκε και με αρκετά υψηλή συμμετοχή από τους φοιτητές, βοήθησε στην εξαγωγή ορισμένων συμπερασμάτων τόσο για τα προβλήματα που υπάρχουν όσο και για ορισμένες πιθανές λύσεις για αυτές τις καταστάσεις. Στην συνέχεια, για την ανάλυση και την σχεδίαση της αρχιτεκτονικής του συστήματος «**UNIWA EXAM PLANNER**», έγινε χρήση του εργαλείου **ADONIS - Business Process Management Toolkit**, μέσω του οποίου κατασκευάστηκαν ορισμένα διαγράμματα τα οποία απεικονίζουν τις διάφορες διαδικασίες του συστήματος, οι οποίες απαρτίζουν και την γενική του λειτουργικότητα στο σύνολο τους καθώς και τους ρόλους των χρηστών του. Επιπλέον, κατά την φάση ανάλυσης και σχεδίασης του **E.R.D. (Entity Relationship Diagram)** και **E.E.R. (Enhanced Entity-Relationship)** για την βάση δεδομένων μας «**examsdb**», έγινε χρήση του περιβάλλοντος **Workbench** που παρέχει το εργαλείο της **Oracle MySQL**.

Κατά την κωδικοποίηση της βάσης δεδομένων βέβαια, έγινε χρήση καθαρού πηγαίου κώδικα σε **MySQL**, καθώς έγινε η εκτέλεση αλλά και ο έλεγχος εγκυρότητας του μέσω του τερματικού (**MySQL Command Line Terminal**). Τέλος, στο κομμάτι της κωδικοποίησης, του ελέγχου αλλά και της εφαρμογής σε κατάσταση πλήρους λειτουργίας του συστήματος «**UNIWA EXAM PLANNER**», έγινε χρήση των τεχνολογιών **Spring Boot Web Service** και **Vaadin Designer** για το **Backend** και το **Frontend** τμήμα του αντίστοιχα, πάντα σε γλώσσα **Java** και όλα αυτά μέσα από το περιβάλλον του εργαλείου **Intellij IDEA**. Συμπληρωματικά, για την διαγραμματική αναπαράσταση της επικοινωνίας μεταξύ του χρήστη και του διακομιστή (**client-server diagram**) έγινε χρήση του εργαλείου **Modelio** καθώς επίσης και για την καλύτερη δόμηση του συστήματος ως **Java Project** στο **Intellij IDEA** περιβάλλον, κάναμε χρήση του εργαλείου **Maven**.

4.2 Ανάλυση Ορολογίας στις Τεχνολογίες Σχεδίασης και Ανάπτυξης του Συστήματος «UNIWA EXAM PLANNER».

Το εργαλείο **ADONIS** αποτελεί μια web based εφαρμογή σε **HTML 5**, η οποία είναι πλήρως συμβατή με το **BPMN 2.0** αλλά και με άλλα διεθνή πρότυπα όπως το **BPMM**, το **DMN** και το **ISO 9000**. Συγκεκριμένα κυκλοφόρησε το **1995** από τον Όμιλο **BPMS (Business Process Management Systems)** στο Ινστιτούτο Επιστήμης Υπολογιστών και Επιχειρηματικής Πληροφορικής του Πανεπιστημίου της Βιέννης, το οποίο αργότερα εξελίχθηκε στην εταιρεία **BOC Group**. Η τελευταία έκδοση του **ADONIS** είναι η **13.0** που κυκλοφόρησε τον **Δεκέμβριο του 2021**, ενώ από το **2008** και μετά κυκλοφόρησε και μια δωρεάν έκδοση του η **Community Edition** με πάνω από **200.000 εγγεγραμμένους χρήστες παγκοσμίως**, χωρίς αυτό βέβαια να επηρεάζει τις είδη υπάρχουσες εμπορικές εκδόσεις του εργαλείου **Starter και Enterprise** αντίστοιχα. Μια χαρακτηριστική δυνατότητα που παρέχει το εργαλείο στον χρήστη του, είναι η μοντελοποίηση επιχειρησιακών διαδικασιών χρησιμοποιώντας τη σημείωση **BPMN** προσομοίωσης διεργασιών και γνώσεων βάσει δεδομένων.

(Wikipedia, [en.wikipedia.org/wiki/ADONIS_\(software\)](https://en.wikipedia.org/wiki/ADONIS_(software)), 2022)

Το εργαλείο **Workbench** της **MySQL**, ο σύγχρονος αντικαταστάτης του **DBDesigner 4** από το **fabFORCE.net** και του πακέτου λογισμικού **MySQL GUI Tools Bundle**, αποτελεί ένα οπτικό εργαλείο σχεδίασης βάσεων δεδομένων, το οποίο είναι υπεύθυνο για την ανάπτυξη, την διαχείριση, το σχεδιασμό, τη δημιουργία αλλά και τη συντήρηση της βάσης δεδομένων σε **SQL** σε ένα ενιαίο ολοκληρωμένο περιβάλλον ανάπτυξης για το σύστημα βάσεων δεδομένων **MySQL**. Συμπληρωματικά, το **MySQL Workbench** αποτελεί κατασκευάσμα των γλωσσών προγραμματισμού **C++**, **C** και **Python**, συμβατό με κάθε λειτουργικό σύστημα και με τελευταία έκδοση του το **8.0.22** η οποία κυκλοφόρησε **12 Σεπτεμβρίου το 2020**. Η ιστορία ξεκινάει, πίσω στο **2002** με **2003** όταν ο Αυστριακός **Michael G. Zinner**, κατασκεύασε το εργαλείο **DBDesigner 4** σε **Delphi 7** και **Kylix 3** για την πλατφόρμα του **fabFORCE.net** και το οποίο ήταν διαθέσιμο για **MS Windows/Mac OS X** και **Linux** και προς τα τέλη του **2003** προσεγγίστηκε από εκπροσώπους της **MySQL AB** για να αναλάβει την ανάπτυξη εργαλείων γραφικής διεπαφής χρήστη (**GUI**) για τη **MySQL** με αποτέλεσμα έτσι να προκύψει το **MySQL GUI Tools Bundle**. Ωστόσο, το πακέτο εργαλείων GUI, έχει αντικατασταθεί από το **MySQL Workbench**, από τις εκδόσεις beta του **MySQL Workbench 5.2**, παρόλο που η ομάδα Υποστήριξης **MySQL** συνέχισε να παρέχει βοήθεια για το πακέτο μέχρι τις **30 Ιουνίου 2010**.

(Wikipedia, https://en.wikipedia.org/wiki/MySQL_Workbench, 2023)

Η **MySQL** είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων με πάνω από 11 εκατομμύρια χρήστες και με υπεύθυνη εταιρία ανάπτυξης του την **Oracle** στις υποδομές **Sun Microsystems**. Το όνομα του συστήματος αυτού, προκύπτει από το όνομα της κόρης του **Ulf Michael Widenius** (γνωστός και ως **Monty**), τη **My** και τρέχει σε έναν εξυπηρετητή (**server**) παρέχοντας πρόσβαση σε πολλούς χρήστες (**clients**) σε ένα σύνολο βάσεων δεδομένων (**databases**). Ανήκει και χρηματοδοτείται από την **MySQL AB** στην Σουηδία, η οποία σήμερα ανήκει στην **Oracle Corporation**, με ενσωμάτωση σε πολλές διαδεδομένες διαδικτυακές υπηρεσίες όπως το **Flickr**, το **YouTube**, η **Wikipedia**, το **Google**, το **Facebook** και το **Twitter**.

(Wikipedia, <https://en.wikipedia.org/wiki/MySQL>, 2023)

Η **Java** είναι μια αντικειμενοστραφής γλώσσα η οποία σχεδιάστηκε από την εταιρεία **Sun Microsystems**. Η γέννηση της στηρίχτηκε στην ανάγκη για την δυνατότητα ανάπτυξης λογισμικού σε μικρό-συσκευές, έτσι στις αρχές του 1991 και μέσα στο **Sun**, ο τότε προγραμματιστής **James Gosling**, έπειτα από πειραματισμούς πάνω στην **C++** κατασκεύασε ορισμένες πειραματικές γλώσσες (όπως την **C++++**, η οποία μετέπειτα ονομάστηκε σε **C#**) και μετά από λίγο καιρό ακόμα και βάση μια πρόταση στο επιτελείο της εταιρίας προέκυψε η γλώσσα **Oak** σαν το δένδρο (βελανιδιά) το οποίο βρισκόταν έξω από το γραφείο του **Gosling**. Κύριο πλεονέκτημα της **Java** σε σχέση με άλλες γλώσσες, είναι η ανεξαρτησία του λειτουργικού συστήματος από την πλατφόρμα που τρέχει. Με άλλα λόγια, ένα τέτοιο πρόγραμμα μπορεί να τρέξει το ίδιο σε ένα σύστημα με **Windows**, **Linux**, **Unix** ή και **Macintosh** (σύντομα και σε διάφορες παιχνιδοκοινότητες όπως πχ **Playstation**) χωρίς να απαιτείται να ξαναγίνει μεταγλώττιση (**compile**) ή αλλαγή του πηγαίου κώδικα (**source code**). Αυτό επιτυγχάνεται με την ύπαρξη εικονικής μηχανής (**virtual machine – vm**), μέσω του οποίου αγνοείται τόσο το υλικό μέρος του συστήματος όσο και το λογισμικό του και έτσι ο κώδικας αποτελεί το ίδιο κατανοητός ως προς το επίπεδο του συμβολικού κώδικα και κώδικα μηχανής, που παράγεται από αυτόν στην επικοινωνία του με το σύστημα. Για την ανάπτυξη ενός προγράμματος σε **Java** δεν χρειάζεται τίποτα άλλο πέρα ένας επεξεργαστής κειμένου όπως το **Notepad – Windows** ή **vi – Unix** καθώς επίσης και ένα από τα ολοκληρωμένα περιβάλλοντα ανάπτυξης που υπάρχουν στην αγορά (**IDE**), όπως το **Netbeans**, το **Apache**, το **Eclipse** και το **Intellij Idea**. Τέλος, λόγω του είδη καταχωρημένου ονόματος **Oak**, η γλώσσα άλλαξε το όνομα της **Java** (αγαπημένη ποικιλία καφέ των συγκεκριμένων προγραμματιστών) και έτσι έκανε την επίσημη εμφάνιση της η γλώσσα **Java** το **Μάρτιο του 1995** όταν η **Sun** την ανακοίνωσε στο συνέδριο **Sun World 1995** καθώς επίσης στις **13 Νοεμβρίου του 2006** έγινε πλέον μια γλώσσα ανοιχτού κώδικα (**GPL**) όσον αφορά το μεταγλωττιστή (**javac**) και το πακέτο ανάπτυξης (**JDK, Java Development Kit**).

(Wikipedia, <https://el.wikipedia.org/wiki/Java>, 2022)

Το **Spring Framework** είναι ένα λογισμικό ανοιχτού κώδικα, το οποίο βοηθάει σημαντικά στην ανάπτυξη **J2EE λογισμικού** και βασίζεται στη γλώσσα προγραμματισμού **Java**. Μια εφαρμογή **Spring Boot** μπορεί να γραφτεί χρησιμοποιώντας μεγάλη ποικιλία γλωσσών προγραμματισμού, συμπεριλαμβανομένων των **Java** και **Kotlin**. Οι ιδιότητες διαμόρφωσης για την εφαρμογή **Spring Boot** μπορούν να καθοριστούν στο αρχείο **application.properties** ή **application.yml** (πχ **server.port**, **spring.application.name**). Το **Spring Boot** έχει μια κλάση **SpringBootServletInitializer**, η οποία είναι μια εξειδίκευση του **WebApplicationInitializer** και στην συγκεκριμένη μορφή της κλάσης, ο χρήστης δεν χρειάζεται πλέον να κάνει δικιά του υλοποίηση της κλάσης.

Ο σχολιασμός **@SpringBootApplication**, επιτρέπει στην εφαρμογή **Spring Boot** να διαμορφώνει αυτόματα τις βιβλιοθήκες τρίτων και τις ανιχνευμένες δυνατότητες και συνδυάζει τους εξής τρεις σχολιασμούς:

- 1) **@SpringBootConfiguration** : μια εξειδίκευση του σχολιασμού **@Configuration** για το **Spring** και ουσιαστικά δηλώνει την ιδιότητα της συγκεκριμένης κλάσης ως η κλάση διαμόρφωσης του **project**.
- 2) **@EnableAutoConfiguration** : η δήλωση της ενεργοποίησης αυτόματης διαμόρφωσης του **Spring Boot** project για το **Spring**.
- 3) **@ComponentScan** : η δήλωση για την αναζήτηση όλων των τμημάτων του **Spring Boot** από την παρούσα κλάση.

Η τελευταία σταθερή (stable) έκδοση του **Spring Boot** προς χρήση είναι η 6.0.8.

(Wikipedia, https://en.wikipedia.org/wiki/Spring_Boot, 2023)

(Wikipedia, https://en.wikipedia.org/wiki/Spring_Framework, 2023)

Το **Vaadin** είναι μια προηγμένη τεχνολογία ανάπτυξης διαδικτυακών εφαρμογών ανοιχτού κώδικα για **Java**. Περιλαμβάνει ένα σύνολο εργαλείων που επιτρέπουν την κατασκευή σύγχρονων γραφικών διεπαφών (**GUI**), χρησιμοποιώντας μόνο τη γλώσσα προγραμματισμού **Java** (αντί για **HTML** και **JavaScript**), μόνο **TypeScript** ή συνδυασμό και των δύο. Η ανάπτυξη του ξεκίνησε για πρώτη φορά ως προσαρμογέας πάνω από το πλαίσιο **web** ανοιχτού κώδικα **Millstone 3** που κυκλοφόρησε το έτος 2002. Κατά τη διάρκεια του 2006, αυτή η ιδέα αναπτύχθηκε στη συνέχεια ξεχωριστά ως εμπορικό προϊόν. Στις 20 Μαΐου 2009, το **IT Mill Toolkit** άλλαξε το όνομά του σε **Vaadin Framework**. Αργότερα, η **IT Mill Ltd**, η εταιρεία πίσω από το ανοιχτού κώδικα **Vaadin Framework**, άλλαξε το όνομά της σε **Vaadin Ltd**.

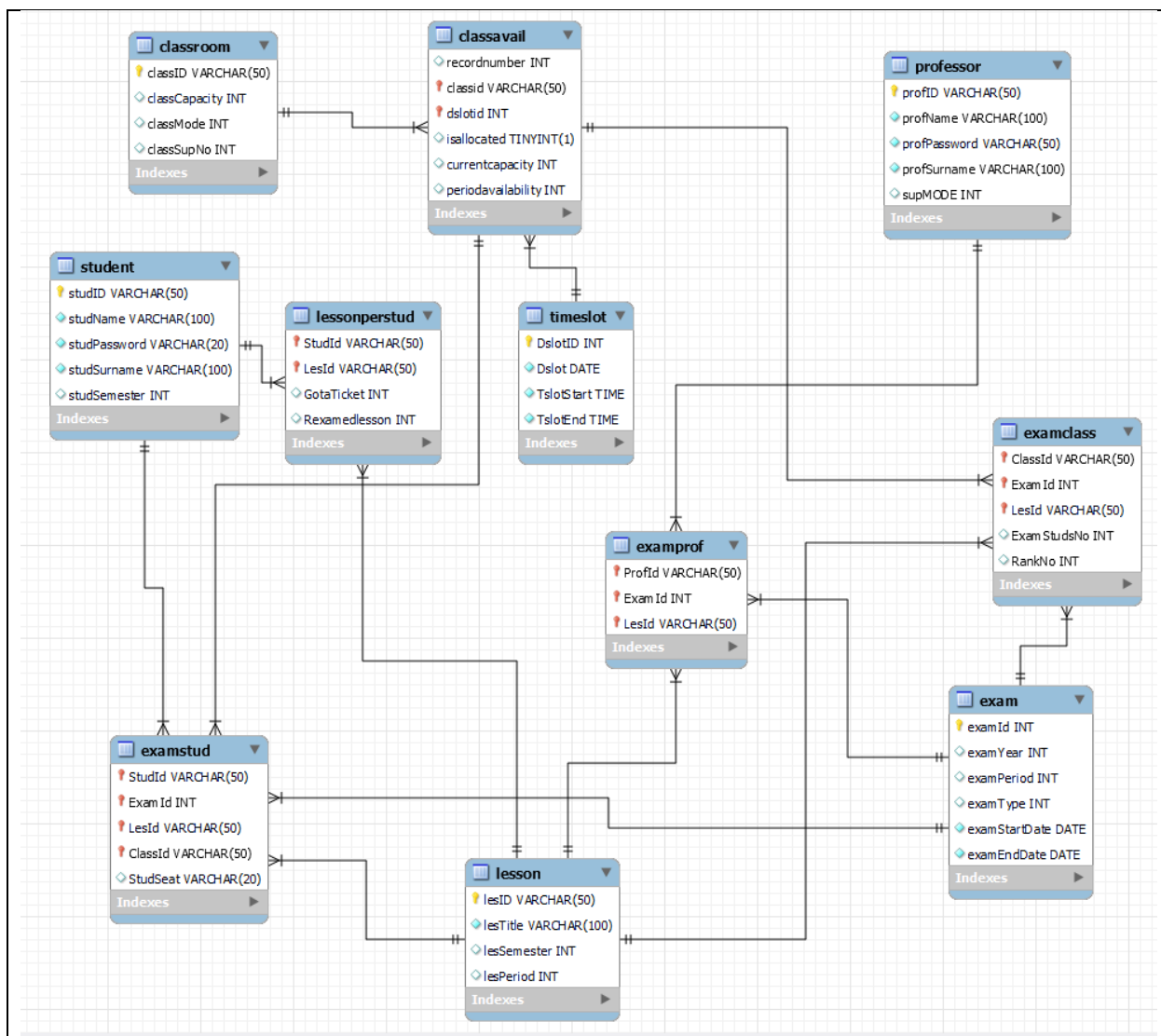
Το **Vaadin Flow** διαθέτει αρχιτεκτονική από την πλευρά του διακομιστή που σημαίνει ότι το μεγαλύτερο μέρος της λογικής διεπαφής χρήστη εκτελείται με ασφάλεια στον διακομιστή μειώνοντας την έκθεση σε εισβολείς. Από την πλευρά του πελάτη, το **Vaadin Flow** είναι χτισμένο πάνω στα πρότυπα του **Web Component**. Η επικοινωνία πελάτη/διακομιστή γίνεται αυτόματα μέσω **WebSocket** ή **HTTP** με ελαφριά μηνύματα **JSON** που ενημερώνουν τόσο τη διεπαφή χρήστη στο πρόγραμμα περιήγησης όσο και την κατάσταση διεπαφής χρήστη στον διακομιστή. Οι προβολές δημιουργούνται από προσαρμοσμένες κλάσεις **Java** που υλοποιούν ένα άλλο στοιχείο διεπαφής χρήστη (προσαρμοσμένο ή παρέχεται από το πλαίσιο). Αυτές οι τάξεις προβολής σχολιάζονται με **@Route** για να εκτεθούν στο πρόγραμμα περιήγησης με μια συγκεκριμένη διεύθυνση **URL**. Τα στοιχεία διεπαφής χρήστη του **Vaadin** λειτουργούν με συμβάντα ποντικίου και αφής, μπορούν να προσαρμοστούν με **CSS**, περιλαμβάνουν υποστήριξη πληκτρολογίου και αναγνωστών οθόνης και υποστηρίζουν γλώσσες από δεξιά προς τα αριστερά.

(Wikipedia, <https://en.wikipedia.org/wiki/Vaadin>, 2023)

Κεφάλαιο 5^ο – Περιγραφή της Βάσης Δεδομένων MySQL και των Spring Boot Web Services του Συστήματος.

5.1 Διαγραμματική Απεικόνιση Βάσης Δεδομένων MySQL – Διάγραμμα Σχέσεων Οντοτήτων E.R. (Entity Relationship - Diagram)

Η διαγραμματική απεικόνιση της Βάσης Δεδομένων «**examsdb**», προκύπτει μέσω του εργαλείου **MySQL Workbench** και αποτελεί το παρακάτω **Διάγραμμα E.R.D. (Entity Relationship Diagram)**:



Εικόνα 27 - Διάγραμμα E.R.D. (Entity Relationship Diagram) της MySQL Βάσης Δεδομένων examsdb.

5.2 Ανάλυση Δομής των Πινάκων της Βάσης Δεδομένων.

Στον παρακάτω πίνακα που ακολουθεί, απεικονίζεται η περαιτέρω ανάλυση του παραπάνω **E.R.D. Διαγράμματος**, μέσω του οποίου κάναμε την διαγραμματική απεικόνιση της δομής της Βάσης Δεδομένων μας «**examsdb**»:

Τίτλος Πίνακα	Πρωτεύον Κλειδί Πίνακα	Ξένα Κλειδιά Πίνακα	Πεδία Πίνακα	Συσχετιζόμενος Πίνακας	Είδος Συσχέτισης	Περιγραφή (Χρήση Πίνακα)
timeslot	DslotID	-	DslotID, Dslot, TslotStart, TslotEnd	-	-	Ο συγκεκριμένος πίνακας περιέχει τις χρονοθυρίδες σε σχέση με την διαθεσιμότητα των αιθουσών του τμήματος.
classroom	classID	-	classID, classCapacity, classMode, classSupNo	-	-	Ο συγκεκριμένος πίνακας αντιπροσωπεύει τις αίθουσες του τμήματος.
classAvail	classid, dslotid	classid, dslotid	recordnumber, classid, dslotid, isallocated, currentcapacity, periodavailability	Πίνακες timeslot και classroom.	Πολλά προς πολλά μεταξύ των πινάκων timeslot και classroom.	Ο συγκεκριμένος πίνακας αντιπροσωπεύει την διαθεσιμότητα των αιθουσών του τμήματος.
lesson	lesID	-	lesID, lesTitle, lesSemester, lesPeriod	-	-	Ο συγκεκριμένος πίνακας περιέχει τα μαθήματα του τμήματος.
student	studID	-	studID, studName, studPassword, studSurname, studSemester	-	-	Ο συγκεκριμένος πίνακας περιέχει τα στοιχεία των φοιτητών του τμήματος.
professor	profID	-	profID, profName, profPassword, profSurname, supMODE	-	-	Ο συγκεκριμένος πίνακας περιέχει τα στοιχεία των καθηγητών του τμήματος.
secretariat	seclID	-	seclID, seclName, seclPassword, seclSurname	-	-	Ο συγκεκριμένος πίνακας περιέχει τα στοιχεία της γραμματείας του τμήματος.

exam	examId	-	examId, examYear, examPeriod, examType, examStartDate, examEndDate	-	-	Ο συγκεκριμένος πίνακας περιέχει τα στοιχεία των εξεταστικών περιόδων.
lessonperstud	StudId, LesId	StudId, LesId	StudId, LesId, GotaTicket, Rexamedlesson	Πίνακες student και lesson.	Πολλά προς πολλά μεταξύ των πινάκων student και lesson.	Ο συγκεκριμένος πίνακας περιέχει τα στοιχεία των προς εξέταση μαθημάτων ανά φοιτητή του τμήματος.
examStud	StudId, ExamId, LesId, ClassId	StudId, ExamId, LesId, ClassId	StudId, ExamId, LesId, ClassId, StudSeat	Πίνακες student, exam, lesson και classAvail.	Πολλά προς πολλά μεταξύ των πινάκων student, exam, lesson και classAvail.	Ο συγκεκριμένος πίνακας περιέχει τα στοιχεία των προς εξέταση μαθημάτων που εξετάζεται ένας φοιτητής σε μια αίθουσα σε μια εξεταστική περίοδο.
examProf	ProfId, ExamId, LesId	ProfId, ExamId, LesId	ProfId, ExamId, LesId	Πίνακες professor, exam και lesson.	Πολλά προς πολλά μεταξύ των πινάκων professor, exam και lesson.	Ο συγκεκριμένος πίνακας περιέχει τα στοιχεία των προς εξέταση μαθημάτων που αφορούν ένα καθηγητή σε μια εξεταστική περίοδο.
examClass	ClassId, ExamId, LesId	ClassId, ExamId, LesId	ClassId, ExamId, LesId, ExamStudsNo, RankNo	Πίνακες classAvail, exam και lesson.	Πολλά προς πολλά μεταξύ των πινάκων classAvail, exam και lesson.	Ο συγκεκριμένος πίνακας περιέχει τα στοιχεία των προς εξέταση μαθημάτων που ανήκουν σε μια εξεταστική περίοδο για μια αίθουσα του τμήματος.

Πίνακας 5 - Ανάλυση Πινάκων Βάσης Δεδομένων και των μεταξύ τους Συσχετίσεων.

Συμπληρωματικά, ακολουθεί και ο παρακάτω πίνακας, με σκοπό την περαιτέρω ανάλυση των πεδίων του κάθε πίνακα, στο είδους καθώς και την χρησιμότητας της πληροφορίας τους:

Τίτλος Πίνακα	Τίτλος Πεδίου	Είδος Δεδομένων	Χρησιμότητα Πληροφορίας
timeslot	DslotID	Ακέραιος Αύξων Αριθμός INT AUTO_INCREMENT	Το πρωτεύον κλειδί του πίνακα και αναγνωριστικό της κάθε εγγραφής, το οποίο μάλιστα σαν πεδίο συμπληρώνεται αυτόματα από το σύστημα ως αύξων αριθμός με πρωταρχική τιμή το 0. (πχ. 0,1,2)
timeslot	Dslot	Ημερομηνία – DATE NOT NULL (Έτος-Μήνας-Μέρα)	Η Ημερομηνία της χρονοθυρίδας ενός μαθήματος ή διαφορετικά της διαθεσιμότητας της αίθουσας που θα γίνει η εξέταση.
timeslot	TslotStart	Χρόνος - TIME NOT NULL (Ωρα:Λεπτά:Δευτερόλεπτα)	Η Ώρα Έναρξης μιας χρονοθυρίδας ή διαφορετικά της διαθεσιμότητας μιας αίθουσας που θα γίνει η εξέταση.
timeslot	TslotEnd	Χρόνος – TIME NOT NULL (Ωρα:Λεπτά:Δευτερόλεπτα)	Η Ώρα Λήξης μιας χρονοθυρίδας ή διαφορετικά της διαθεσιμότητας μιας αίθουσας που θα γίνει η εξέταση.
classroom	classID	Γραμματοσειρά διάστασης 50 χαρακτήρων. VARCHAR (50) NOT NULL	Το πρωτεύον κλειδί του πίνακα και αναγνωριστικό της κάθε εγγραφής.
classroom	classCapacity	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Η χωρητικότητα της αίθουσας.
classroom	classMode	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Μια σήμανση για το αν η αίθουσα μας έχει 1 ή 2 κενές θέσεις ανά φοιτητή (με 0 και 1 αντίστοιχα) κατά την χρήση της στις εξετάσεις. Πεδίο καθαρά για την πληρότητα της πληροφορίας του πίνακα.
classroom	classSupNo	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Ελάχιστος αριθμός από επιτηρητές για την αίθουσα. Πεδίο που υπάρχει, καθαρά για λόγους πληρότητας της πληροφορίας.
classAvail	recordnumber	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Αριθμός που αντιπροσωπεύει την σημαντικότητα της κάθε εγγραφής. Αντιστοιχεί με το πεδίο RankNo στον πίνακα examClass. Υπάρχει καθαρά για την λειτουργικότητα του συστήματος, κατά την δήλωση της συμμετοχής του φοιτητή σε μια εξέταση και την αναζήτηση της εκάστοτε διαθέσιμης αίθουσας.
classAvail	classid	Γραμματοσειρά διάστασης 50 χαρακτήρων. VARCHAR (50) NOT NULL	Μέρος πρωτεύων κλειδιού και δευτερέων κλειδί για την συσχέτιση με τον πίνακα classroom.

classAvail	dslotid	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Μέρος πρωτεύων κλειδιού και δευτερεύων κλειδί για την συσχέτιση με τον πίνακα timeslot.
classAvail	isallocated	Λογική τιμή με προκαθορισμένη τιμή αληθείας ως ψευδής. BOOLEAN DEFAULT false	Ένας πεδίο που λειτουργεί ως λογικός τελεστής και δηλώνει την διαθεσιμότητα μιας αίθουσας. Δηλώνεται από την Γραμματεία ως true και αφού γίνει δέσμευση της αίθουσας από τον Φοιτητή επιστρέφει πάλι ως false.
classAvail	currentcapacity	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Η τρέχουσα χωρητικότητα της αίθουσας, η οποία έχει ως αρχική τιμή την χωρητικότητα της αίθουσας πριν από την χρήση της και σταδιακά αυξομειώνεται.
classAvail	periodavailability	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Η εξεταστική περίοδος που είναι διαθέσιμη η αίθουσα. Για παράδειγμα κατά την χειμερινή εξεταστική περίοδο η τιμή στο πεδίο είναι 0.
lesson	lesID	Γραμματοσειρά διάστασης 50 χαρακτήρων. VARCHAR (50) NOT NULL	Το πρωτεύων κλειδί του πίνακα και αναγνωριστικό της κάθε εγγραφής.
lesson	lesTitle	Γραμματοσειρά διάστασης 100 χαρακτήρων. VARCHAR (100) NOT NULL	Ο τίτλος του μαθήματος.
lesson	lesSemester	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Το εξάμηνο που ανήκει το μάθημα.
lesson	lesPeriod	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Η περίοδος που ανήκει το εξάμηνο του μαθήματος.
student	studID	Γραμματοσειρά διάστασης 50 χαρακτήρων. VARCHAR (50) NOT NULL	Το πρωτεύων κλειδί του πίνακα και αναγνωριστικό της κάθε εγγραφής. Επίσης, αποτελεί και το όνομα χρήστη (username) κατά την είσοδο στο σύστημα για τον Φοιτητή.
student	studName	Γραμματοσειρά διάστασης 100 χαρακτήρων. VARCHAR (100) NOT NULL	Το όνομα του Φοιτητή.
student	studPassword	Γραμματοσειρά διάστασης 20 χαρακτήρων. VARCHAR (20) NOT NULL	Ο κωδικός πρόσβασης (password) του Φοιτητή στο σύστημα.
student	studSurname	Γραμματοσειρά διάστασης 100 χαρακτήρων. VARCHAR (100) NOT NULL	Το επίθετο του Φοιτητή.
student	studSemester	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Το τρέχον εξάμηνο, στο οποίο και έχει εγγραφεί ο Φοιτητής.

professor	profID	Γραμματοσειρά διάστασης 50 χαρακτήρων. VARCHAR (50) NOT NULL	Το πρωτεύον κλειδί του πίνακα και αναγνωριστικό της κάθε εγγραφής. Επίσης, αποτελεί και το όνομα χρήστη (username) κατά την είσοδο στο σύστημα για τον Καθηγητή.
professor	profName	Γραμματοσειρά διάστασης 100 χαρακτήρων. VARCHAR (100) NOT NULL	Το όνομα του Καθηγητή.
professor	profPassword	Γραμματοσειρά διάστασης 50 χαρακτήρων. VARCHAR (50) NOT NULL	Ο κωδικός πρόσβασης (password) του Καθηγητή στο σύστημα.
professor	profSurname	Γραμματοσειρά διάστασης 100 χαρακτήρων. VARCHAR (100) NOT NULL	Το επίθετο του Καθηγητή.
professor	supMODE	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Ένα αναγνωριστικό που μας δηλώνει αν ένας Καθηγητής είναι και Επιτηρητής ή όχι. Πεδίο που υπάρχει, καθαρά για λόγους πληρότητας της πληροφορίας.
secretariat	seclID	Γραμματοσειρά διάστασης 50 χαρακτήρων. VARCHAR (50) NOT NULL	Το πρωτεύων κλειδί του πίνακα και αναγνωριστικό της κάθε εγγραφής. Επίσης, αποτελεί και το όνομα χρήστη (username) κατά την είσοδο στο σύστημα για τον χρήστη Γραμματεία.
secretariat	seclName	Γραμματοσειρά διάστασης 100 χαρακτήρων. VARCHAR (100) NOT NULL	Το όνομα του χρήστη στην Γραμματεία.
secretariat	seclPassword	Γραμματοσειρά διάστασης 50 χαρακτήρων. VARCHAR (50) NOT NULL	Ο κωδικός πρόσβασης (password) του χρήστη Γραμματεία στο σύστημα.
secretariat	seclSurname	Γραμματοσειρά διάστασης 100 χαρακτήρων. VARCHAR (100) NOT NULL	Το επίθετο του χρήστη Γραμματεία.
exam	examId	Ακέραιος Αύξων Αριθμός INT AUTO_INCREMENT	Το πρωτεύων κλειδί του πίνακα και αναγνωριστικό της κάθε εγγραφής.
exam	examYear	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Το έτος της Εξεταστικής.
exam	examPeriod	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Η Εξεταστική περίοδος. Για παράδειγμα, Χειμερινή Περίοδος με τιμή πεδίου 0.
exam	examType	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Το είδος της εξεταστικής. Για παράδειγμα, Εμβόλιμη Εξεταστική με τιμή πεδίου 1.
exam	examStartDate	Ημερομηνία – DATE NOT NULL (Έτος-Μήνας-Μέρα)	Ημερομηνία Έναρξης της Εξεταστικής.
exam	examEndDate	Ημερομηνία – DATE NOT NULL (Έτος-Μήνας-Μέρα)	Ημερομηνία Λήξης της Εξεταστικής.

lessonperstud	StudId	Γραμματοσειρά διάστασης 50 χαρακτήρων. VARCHAR (50) NOT NULL	Μέρος πρωτεύων κλειδιού και δευτερεύων κλειδί για την συσχέτιση με τον πίνακα student.
lessonperstud	LesId	Γραμματοσειρά διάστασης 50 χαρακτήρων. VARCHAR (50) NOT NULL	Μέρος πρωτεύων κλειδιού και δευτερεύων κλειδί για την συσχέτιση με τον πίνακα lesson.
lessonperstud	GotaTicket	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Το πεδίο που δηλώνεται αν ένα μάθημα έχει δηλωθεί είδη για εξέταση ή όχι, με 1 και 0 αντίστοιχα για τιμή πεδίου.
lessonperstud	Rexamedlesson	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Το πεδίο που δηλώνεται αν ένα μάθημα έχει δηλωθεί είδη για εξέταση βελτίωσης βαθμού ή όχι, με 1 και 0 αντίστοιχα.
examStud	StudId	Γραμματοσειρά διάστασης 50 χαρακτήρων. VARCHAR (50) NOT NULL	Μέρος πρωτεύων κλειδιού και δευτερεύων κλειδί για την συσχέτιση με τον πίνακα student.
examStud	ExamId	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Μέρος πρωτεύων κλειδιού και δευτερεύων κλειδί για την συσχέτιση με τον πίνακα exam.
examStud	LesId	Γραμματοσειρά διάστασης 50 χαρακτήρων. VARCHAR (50) NOT NULL	Μέρος πρωτεύων κλειδιού και δευτερεύων κλειδί για την συσχέτιση με τον πίνακα lesson.
examStud	ClassId	Γραμματοσειρά διάστασης 50 χαρακτήρων. VARCHAR (50) NOT NULL	Μέρος πρωτεύων κλειδιού και δευτερεύων κλειδί για την συσχέτιση με τον πίνακα classAvail.
examStud	StudSeat	Γραμματοσειρά διάστασης 20 χαρακτήρων. VARCHAR (20) NOT NULL	Η θέση του φοιτητή στην αίθουσα για την εξέταση του σε ένα μάθημα. Για παράδειγμα "00", "01".
examProf	ProfId	Γραμματοσειρά διάστασης 50 χαρακτήρων. VARCHAR (50) NOT NULL	Μέρος πρωτεύων κλειδιού και δευτερεύων κλειδί για την συσχέτιση με τον πίνακα professor.
examProf	ExamId	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Μέρος πρωτεύων κλειδιού και δευτερεύων κλειδί για την συσχέτιση με τον πίνακα exam.
examProf	LesId	Γραμματοσειρά διάστασης 50 χαρακτήρων. VARCHAR (50) NOT NULL	Μέρος πρωτεύων κλειδιού και δευτερεύων κλειδί για την συσχέτιση με τον πίνακα lesson.
examClass	ClassId	Γραμματοσειρά διάστασης 50 χαρακτήρων. VARCHAR (50) NOT NULL	Μέρος πρωτεύων κλειδιού και δευτερεύων κλειδί για την συσχέτιση με τον πίνακα classAvail.
examClass	ExamId	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Μέρος πρωτεύων κλειδιού και δευτερεύων κλειδί για την συσχέτιση με τον πίνακα exam.
examClass	LesId	Γραμματοσειρά διάστασης 50 χαρακτήρων. VARCHAR (50) NOT NULL	Μέρος πρωτεύων κλειδιού και δευτερεύων κλειδί για την συσχέτιση με τον πίνακα lesson.

examClass	ExamStudsNo	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Το τρέχον πλήθος φοιτητών που έχουν δηλώσει συμμετοχή στις εξετάσεις για την συγκεκριμένη αίθουσα.
examClass	RankNo	Ακέραιος Αριθμός με προεπιλεγμένη τιμή το 0 σε περίπτωση μη εισαγωγής πληροφορίας στο πεδίο. INT DEFAULT 0	Ο βαθμός σημαντικότητας της αίθουσας.Πεδίο συσχετιζόμενο με το πεδίο recordnumber του πίνακα classAvail, το οποίο χρησιμεύει η ύπαρξη του στην λειτουργικότητα του συστήματος και συγκεκριμένα κατά την δέσμευση αίθουσας από τον φοιτητή για την δήλωση ενός μαθήματος προς εξέταση στην τρέχουσα περίοδο.

Πίνακας 6 - Ανάλυση των Πεδίων των Πινάκων και της χρησιμότητας της Πληροφορίας τους.

Μια ακόμα υποσημείωση όσον αφορά τις τιμές των πεδίων των πινάκων της βάσης δεδομένων **examsdb**, είναι ότι όπου υπάρχει η ένδειξη NOT NULL σημαίνει ότι το συγκεκριμένο πεδίο δεν μπορεί να πάρει την τιμή NULL και συνεπώς να μην δηλωθεί χωρίς τιμή κατά την δημιουργία μιας εγγραφής στον συγκεκριμένο πίνακα. Κύρια χρήση αυτής της παραμέτρου στην δήλωση, είναι για πεδία που είναι απαραίτητα να έχουν μια τιμή όπως τα πρωτεύων και δευτερεύων κλειδιά.

5.3 Ανάλυση Συστήματος σε Web Services.

Αντίστοιχα με την προηγούμενη ενότητα όπου και έγινε η ανάλυση αλλά και η καταγραφή της δομής της βάσης δεδομένων μας σε μορφή δομής πίνακα, υπάρχει λεπτομερώς στον παρακάτω πίνακα που ακολουθεί και η δομή του συστήματος «UNIWA EXAM PLANNER», πλήρως αναλυμένη ως προς τα διάφορα **web services** που απαρτίζουν την λειτουργικότητα του.

Τίτλος Web Service	Path Web Service	Είσοδος	Έξοδος	Λειτουργία
LoginView (LoginService, SecretariatService, StudentService, ProfessorService).	Route Path "/"	Όνομα Χρήστη και Κωδικός Πρόσβασης (username/password).	Αντίστοιχη αρχική σελίδα βάση των στοιχείων εισόδου ή διαφορετικά ένα μήνυμα λάθους αναλόγως την περίπτωση βέβαια.	Μέσω της κλάσης LoginView.java και του αντίστοιχου typescript αρχείου, έχουμε την σελίδα εισόδου (Login) στην οθόνη. Αφού λοιπόν ο χρήστης εισάγει τα στοιχεία του σε αυτήν και πατήσει Enter, τα στοιχεία αυτά θα εισαχθούν στην μέθοδο GetAccess και μέσω του loginService θα έχουμε πίσω μια τιμή, η οποία εφόσον τα στοιχεία θα είναι έγκυρα θα είναι μεταξύ 1 και 3 και αναλόγως θα πάρουμε και την ανάλογη σελίδα πίσω. Μαζί με την σελίδα, γίνεται αρχικοποίηση και ενός Session με τα στοιχεία του χρήστη μέσω του ονόματος χρήστη (username).
SecretariatMainView	"secretariat"	Επιλογή του χρήστη από το μενού.	Αντίστοιχη υπηρεσία βάση της επιλογής του μενού.	Ουσιαστικά αυτή η υπηρεσία, λειτουργεί σαν ένα βασικό μενού επιλογών, στο οποίο ο χρήστης γραμματεία, επιλέγει την αντίστοιχη υπηρεσία που επιθυμεί να επισκεφτεί στην συνέχεια.
CreateExam	"secretariat/exam-create"	Έτος, Περίοδος, Είδος και Ημερομηνία Έναρξης και Λήξης Εξεταστικής.	Μήνυμα επιτυχούς καταχώρησης ή Μήνυμα λάθους στον Χρήστη.	Εδώ ουσιαστικά έχουμε μια φόρμα για την καταχώρηση των δεδομένων μιας εξεταστικής στο σύστημα από την γραμματεία, με τους αντίστοιχους ελέγχους και συνθήκες για την διασφάλιση της εγκυρότητας των δεδομένων.

CreateExamprof	"/secretariat/createxamprof"	Έτος Εξεταστικής, Είδος Εξεταστικής, Όνομα Καθηγητή και Τίτλος Μαθήματος.	Μήνυμα επιτυχούς καταχώρησης ή Μήνυμα λάθους στον Χρήστη.	Εδώ ουσιαστικά έχουμε μια ακόμα φόρμα για την εισαγωγή των δεδομένων ενός μαθήματος ως προς την εξεταστική που ανήκει και τον καθηγητή που το διδάσκει στο σύστημα. Επίσης, υπάρχουν και οι απαραίτητοι έλεγχοι για την διασφάλιση της εγκυρότητας των δεδομένων μας, πριν την εισαγωγή τους στο σύστημα από την γραμματοεπικοινωνία.
UpdateExamInfoView	"/secretariat/update-exam"	Νέα στοιχεία όσον αφορά το Έτος, την Περίοδο, το Είδος και την Ημερομηνία Έναρξης και Λήξης τις Εξεταστικής.	Μήνυμα επιτυχούς καταχώρησης και ενημέρωσης της εγγραφής για την εκάστοτε εξεταστική περίοδο.	Στο συγκεκριμένο service, ο χρήστης γραμματοεπικοινωνία, έχει την δυνατότητα να επεξεργαστεί τα δεδομένα μιας ή πολλαπλών εξεταστικών περιόδων.
AllocateExamRoomView	"secretariat/allocate-room"	Η επιλεγμένη διαθέσιμη ή δεσμευμένη χρονοθυρίδα της εκάστοτε αίθουσας, από τον χρήστη.	Μήνυμα επιτυχούς δέσμευσης ή αποδέσμευσης της εκάστοτε χρονοθυρίδας και ενημέρωση της σελίδας, ώστε να φαίνεται η νέα κατάσταση της διαθεσιμότητας των χρονοθυρίδων των αιθουσών.	Σε αυτό το service, γίνεται η επιλογή των διαθέσιμων ή και μη χρονοθυρίδων ανά αίθουσα μέσω της γραμματοεπικοινωνία, με σκοπό την δέσμευση ή αποδέσμευση τους και γενικότερα την δήλωση διαθεσιμότητας ή και την αφαίρεση της, για μια αίθουσα σε μια εξεταστική περίοδο στο σύστημα.
ExamInfoView	"secretariat/exam-info"	-	Μια συνολική παρουσίαση των πληροφοριών που έχουν δηλωθεί στο σύστημα, ως προς τις εξεταστικές περιόδους αλλά και τις χρονοθυρίδες ανά αίθουσα του τμήματος.	Εδώ έχουμε ένα service, το οποίο η δουλειά του είναι να εμφανίζει όσες πληροφορίες έχουν περαστεί στο σύστημα ως προς τις εξεταστικές περιόδους και την διαθεσιμότητα των αιθουσών, μέσα από τις αντίστοιχες χρονοθυρίδες τους.
StudentMainView	"student"	Επιλογή του χρήστη από το μενού.	Αντίστοιχη υπηρεσία βάση της επιλογής του μενού.	Ουσιαστικά αυτή η υπηρεσία, λειτουργεί σαν ένα βασικό μενού επιλογών, στο οποίο ο χρήστης φοιτητής, επιλέγει την αντίστοιχη υπηρεσία που επιθυμεί να επισκεφτεί στην συνέχεια.

ApplyView (Student Service)	"/student/apply"	Επιλογή διαπιστευτηρίου σε ένα μάθημα.	Μήνυμα επιτυχούς καταχώρησης και στοιχεία εξέτασης μαθήματος ή ένδειξη λάθους προς το χρήστη.	Στο συγκεκριμένο service, ο χρήστης φοιτητής επιλέγει ένα από τα διαθέσιμα διαπιστευτήρια σε ένα ή περισσότερα από τα μαθήματα που έχει δηλωμένα και εφόσον είναι σε θέση να εξετασθεί, κάνει την δήλωση του στο σύστημα. Αυτή η λειτουργία γίνεται μέσω της insertrecord μεθόδου της StudentService.java κλάσης όπως και εντελώς συμπληρωματικά για την αρχικοποίηση ορισμένων χρήσιμων παραμέτρων του συστήματος που συμβάλουν στην λειτουργία της insertrecord, έχουμε πρώτα την κλήση της InitiateSession από την ίδια κλάση.
StudentApplicationsView (Student Service)	"student/applications"	Επιλογή ακύρωσης μιας αίτησης για εξέταση μαθήματος από το φοιτητή.	Μια συνολική παρουσίαση των πληροφοριών που έχουν δηλωθεί στο σύστημα ως προς τις δηλώσεις του εκάστοτε φοιτητή, με δυνατότητα ακύρωσης ενός ή και πολλαπλών αιτήσεων.	Εδώ έχουμε ένα service, το οποίο η δουλειά του είναι, να εμφανίζει όλες τις πληροφορίες έχουν περαστεί στο σύστημα ως προς τα δηλωμένα προς εξέταση μαθήματα από το χρήστη φοιτητή καθώς και υπάρχει δυνατότητα ακύρωσης αίτησης. Η λειτουργία της ακύρωσης γίνεται σε συνδυασμό με την StudentService κλάση και συγκεκριμένα, μέσα από την κλήση της deleterecord μεθόδου της.
ProfessorMainView	"professor"	Επιλογή του χρήστη από το μενού.	Αντίστοιχη υπηρεσία βάση της επιλογής του μενού.	Ουσιαστικά αυτή η υπηρεσία, λειτουργεί σαν ένα βασικό μενού επιλογών, στο οποίο ο χρήστης καθηγητής, επιλέγει την αντίστοιχη υπηρεσία που επιθυμεί να επισκεφτεί στην συνέχεια.
ProfessorApplicationsView	"professor/applications"	-	Μια συνολική παρουσίαση των πληροφοριών που έχουν δηλωθεί στο σύστημα ως προς τις δηλώσεις των φοιτητών ανά μάθημα καθηγητή.	Εδώ έχουμε ένα service, το οποίο ουσιαστικά παρουσιάζει τις εξεταστικές περιόδους στις οποίες ανήκει ο εκάστοτε καθηγητής και πατώντας πάνω σε κάθε μια από αυτές, είναι ορατοί οι φοιτητές που θα συμμετάσχουν σε ένα ή περισσότερα μαθήματα για τον συγκεκριμένο καθηγητή.

Πίνακας 7 - Ανάλυση των Web Services του Συστήματος.

Κεφάλαιο 6° – Οδηγός Εγκατάστασης Απαραίτητων Εργαλείων και Χρήσης Συστήματος.

6.1 Βοηθητική Εγκατάσταση Εργαλείων και Μέθοδος Παραγωγής Εκτελέσιμου Κώδικα.

Για την υλοποίηση του συστήματος «**UNIWA EXAM PLANNER**», της βάσης δεδομένων «**examsdb**» καθώς και την παραγωγή του εκτελέσιμου κώδικα εφόσον κάποια στιγμή είναι επιθυμητό, μετά από σχετική έρευνα στο διαδίκτυο για την συμβατότητα μεταξύ των εκδόσεων των διάφορων τεχνολογιών της εποχής, συγκεντρώθηκαν και εγκαταστάθηκαν τα παρακάτω στο σύστημα μας:

1. **Java 18**
2. **Maven**
3. **Vaadin 14.8.18**
4. **Spring Boot 2.6.3**
5. **Oracle MySQL**

Η εγκατάσταση λοιπόν αλλά και η ρύθμιση για την καλύτερη λειτουργία τους, έγινε βάση της παρακάτω μεθοδολογίας:

1. Η εγκατάσταση του JDK 18 έγινε βάση του παρακάτω συνδέσμου:
https://download.oracle.com/java/18/archive/jdk-18_windows-x64_bin.exe
(Oracle, 2023)
2. Η ρύθμιση του **java_home** στις παραμέτρους του συστήματος πραγματοποιήθηκε από τον οδηγό του συνδέσμου που ακολουθεί:
https://confluence.atlassian.com/doc/setting-the-java_home-variable-in-windows-8895.html
(Atlassian, 2022)
3. Για το κατέβασμα και την εγκατάσταση της **MySQL**, χρησιμοποιήθηκαν οι οδηγίες της συγκεκριμένης ιστοσελίδας:
<https://dev.mysql.com/downloads/installer/>
(MySQL, 2023)
4. Η χορήγηση δικαιωμάτων στον χρήστη της βάσης δεδομένων, έγινε μέσα από την εκτέλεση των παρακάτω δύο SQL queries:
I) GRANT ALL PRIVILEGES ON *.* TO root@'localhost' WITH GRANT OPTION;
II) FLUSH PRIVILEGES;
5. Η εκτέλεση του SQL script που συνοδεύει το project, έχει ως στόχο τον ορισμό αλλά και την εισαγωγή της απαραίτητης πληροφορίας στην βάση δεδομένων του συστήματος.
6. Η εκτέλεση του project, γίνεται χρησιμοποιώντας το maven wrapper:
I) Εκτελείται η εξής εντολή μέσα στον φάκελο του project, ώστε να τρέξει το πρόγραμμα στην port 8080:
mnhw spring-boot:run

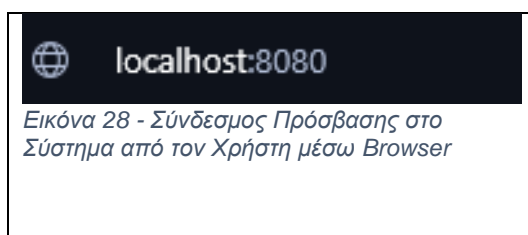
Η ανάπτυξη της εφαρμογής ως εκτελέσιμο αρχείο αλλά και η εκτέλεση του, γίνεται βάση της παρακάτω μεθόδου:

1. Η δημιουργία του αρχείου jar, γίνεται βάση της παρακάτω μεθοδολογίας:
 - i. Σε Windows μέσω της εντολής “*mvnw clean install*” χωρίς το “./” στην αρχή.
 - ii. Το “uniwaexamsproj-1.0-SNAPSHOT.jar” βρίσκεται στον φάκελο “target” .
2. Η εκτέλεση του αρχείου jar, επιτυγχάνεται μέσω της παρακάτω εντολής:
 - i. “Java -jar uniwaexamsproj-1.0-SNAPSHOT.jar”
3. Για την δημιουργία μιας υπηρεσίας στα Windows, η οποία θα τρέχει από μόνη της το εκτελέσιμο αρχείο .jar της εφαρμογής, χρειάζεται μόνο μια επίσκεψη στον οδηγό που παρέχεται με τον παρακάτω σύνδεσμο: <https://medium.com/@chamathdilanka.official/creating-windows-service-to-run-an-executable-jar-822924b28f5e>
(Guruge, 2019)

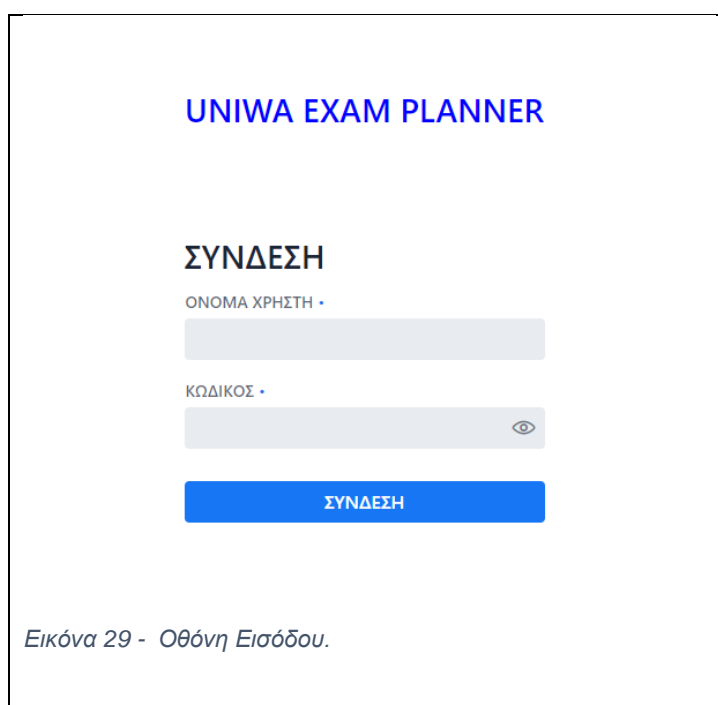
6.2 Οδηγίες Χρήσεις Συστήματος με Ενδεικτικές Οθόνες ανά Περίπτωση.

Πρώτη υπηρεσία του συστήματος προς ένα χρήστη κατά την είσοδο του στην αρχική σελίδα του συστήματος «**UNIWA EXAM PLANNER**», είναι η αξιόπιστη και εύκολη είσοδος του, βάση πάνω απ' όλα της ιδιότητας αυτού (Μέλος Γραμματείας, Φοιτητής ή και Καθηγητής) αλλά και των στοιχείων εισόδου που έχει εξαρχής στην διάθεση του και τα οποία διακρίνονται σε όνομα χρήστη και κωδικός πρόσβασης (username και password αντίστοιχα).

Επίσης, πρέπει να σημειωθεί ότι το σύστημα ακούει στην πόρτα 8080 και ο κάθε χρήστης έχει πρόσβαση σε αυτό μέσω του συνδέσμου που φαίνεται παρακάτω.



Εφόσον ο χρήστης επισκεφτεί τον παραπάνω σύνδεσμο στον περιηγητή του (browser), θα του εμφανιστεί ως αρχική σελίδα η παρακάτω, μέσω της οποίας θα συνδεθεί ουσιαστικά και στο σύστημα.



UNIWA EXAM PLANNER

ΣΥΝΔΕΣΗ

ΟΝΟΜΑ ΧΡΗΣΤΗ •

ΚΩΔΙΚΟΣ •

ΣΥΝΔΕΣΗ

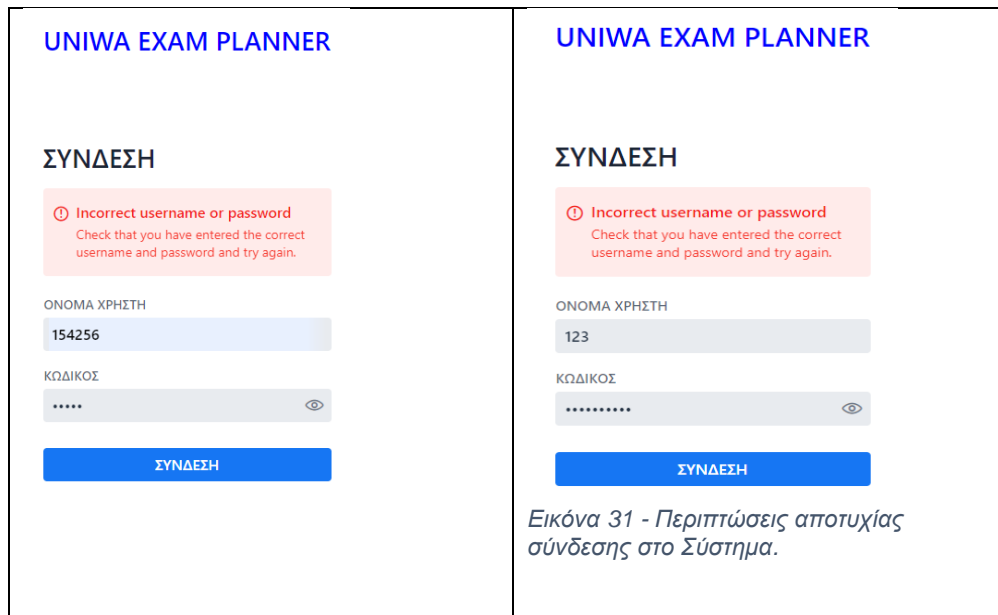
Εικόνα 29 - Οθόνη Εισόδου.

Για αυτό ας πάρουμε ενδεικτικά, την περίπτωση που έχουμε τον χρήστη Μαρία Γεωργίου, ως μέλος του προσωπικού της Γραμματείας του τμήματος, και η οποία επιθυμεί την πρόσβαση της στο σύστημα. Αν και εφόσον δώσει τα σωστά στοιχεία της στην παρακάτω οθόνη που είναι και η αρχική του συστήματος, θα έχει πρόσβαση στο αρχικό μενού της Γραμματείας, όπως φαίνεται και στις παρακάτω 2 εικόνες:

<p>UNIWA EXAM PLANNER</p> <p>ΣΥΝΔΕΣΗ</p> <p>ΟΝΟΜΑ ΧΡΗΣΤΗ</p> <input type="text" value="154256"/> <p>ΚΩΔΙΚΟΣ</p> <input type="password" value="....."/> <p>ΣΥΝΔΕΣΗ</p>	<p>ΛΕΙΤΟΥΡΓΙΕΣ ΓΡΑΜΜΑΤΕΙΑΣ</p> <div data-bbox="887 640 1267 1142"><p>ΝΕΑ ΕΞΕΤΑΣΤΙΚΗ</p><p>ΕΝΗΜΕΡΩΣΗ ΕΞΕΤΑΣΤΙΚΗΣ</p><p>ΕΝΗΜΕΡΩΣΗ ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ ΔΙΔΟΥΣΩΝ</p><p>ΕΜΦΑΝΙΣΗ ΠΛΗΡΟΦΟΡΙΩΝ ΕΞΕΤΑΣΤΙΚΗΣ</p><p>ΕΞΟΔΟΣ</p></div> <p><i>Εικόνα 30 - Επιτυχής Σύνδεση στο Σύστημα (Χρήστης - Γραμματεία).</i></p>
--	--

Από την άλλη βέβαια, αν ο χρήστης αυτός ή και οποιοσδήποτε άλλος εισάγει λάθος στοιχεία στην αρχική οθόνη, θα πάρει μηνύματα ενδεικτικά σαν τα παρακάτω, για κάθε μια από τις πιθανές περιπτώσεις συμπλήρωσης λάθους στοιχείων στην φόρμα:

<p>UNIWA EXAM PLANNER</p> <p>ΣΥΝΔΕΣΗ</p> <p>ΟΝΟΜΑ ΧΡΗΣΤΗ •</p> <input type="text"/> <p>ΚΩΔΙΚΟΣ •</p> <input type="password"/> <p>ΣΥΝΔΕΣΗ</p>	<p>UNIWA EXAM PLANNER</p> <p>ΣΥΝΔΕΣΗ</p> <p>ΟΝΟΜΑ ΧΡΗΣΤΗ</p> <input type="text" value="154256"/> <p>ΚΩΔΙΚΟΣ •</p> <input type="password"/> <p>ΣΥΝΔΕΣΗ</p>
---	--

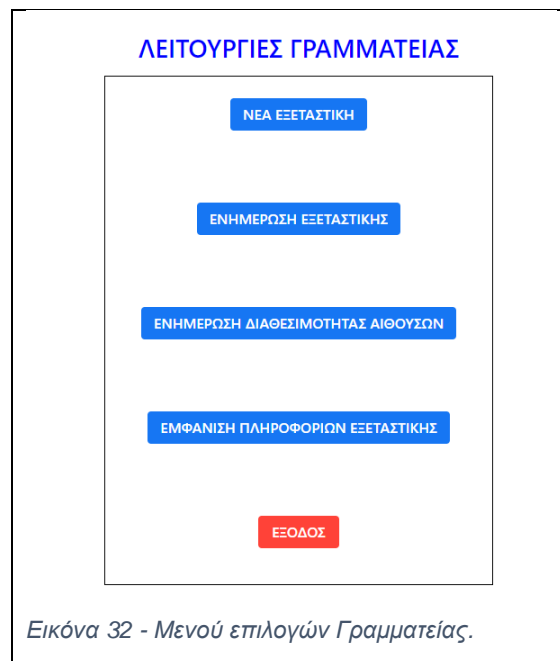


Εικόνα 31 - Περιπτώσεις αποτυχίας σύνδεσης στο Σύστημα.

6.2.1 Περίπτωση Χρήστη Γραμματείας.

Η Γραμματεία αποτελεί τον αρχικό χρήστη του συστήματος, ο οποίος είναι υπεύθυνος για την ενημέρωση του με χρήσιμες πληροφορίες, πάνω στην επερχόμενη εξεταστική περίοδο αλλά και στην διαθεσιμότητα των αιθουσών του τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής.

Παρακάτω ακολουθεί η πρώτη οθόνη που βλέπει ο χρήστης Γραμματεία, με το που εισαχθεί στο σύστημα και είναι ουσιαστικά το αρχικό του μενού:



Εικόνα 32 - Μενού επιλογών Γραμματείας.

Στην συνέχεια ο χρήστης, επιλέγει είτε να εισάγει μια νέα εξεταστική στο σύστημα μέσω της επιλογής «**ΝΕΑ ΕΞΕΤΑΣΤΙΚΗ**», είτε να ενημερώσει μια είδη υπάρχουσα μέσω της επιλογής «**ΕΝΗΜΕΡΩΣΗ ΕΞΕΤΑΣΤΙΚΗΣ**», να ενημερώσει την διαθεσιμότητα των

αιθουσών επιλέγοντας «**ΕΝΗΜΕΡΩΣΗ ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ ΑΙΘΟΥΣΩΝ**» ή τέλος να επιλέξει «**ΕΜΦΑΝΙΣΗ ΠΛΗΡΟΦΟΡΙΩΝ ΕΞΕΤΑΣΤΙΚΗΣ**» για να δει τις συνολικές πληροφορίες από την προ υπάρχουσα χρήση του συστήματος και τα δεδομένα που έχει είδη εισαγάγει.

Επιλέγοντας «**ΝΕΑ ΕΞΕΤΑΣΤΙΚΗ**», ο χρήστης βλέπει την παρακάτω φόρμα:

Η οθόνη αποτελείται από τις επιλογές «**ΥΠΟΒΟΛΗ**» για υποβολή των δεδομένων της φόρμας στο σύστημα, «**ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ**» για καθάρισμα ολόκληρης της φόρμας και επαναφοράς των πεδίων της, στις προκαθορισμένες τιμές τους, «**ΕΠΙΣΤΡΟΦΗ**» για επιστροφή στην προηγούμενη οθόνη του συστήματος και «**ΕΞΟΔΟΣ**» για έξοδο από το Σύστημα και επιστροφή στην αρχική οθόνη εισόδου στο σύστημα «**LOGIN**». Έπειτα, όπως αρκετά εύκολα φαίνεται στην παραπάνω εικόνα, ο χρήστης έχει στην διάθεση του τις απαραίτητες οδηγίες ανά πεδίο της φόρμας, με σκοπό την σωστή εισαγωγή των δεδομένων. Βέβαια, ακόμα και στην περίπτωση της λάθος εισαγωγής δεδομένων, το σύστημα είναι άρτια εξοπλισμένο με μεγάλο αριθμό μηχανισμών και ελέγχων, με σκοπό την ενημέρωση του χρήστη για το λάθος που προέκυψε και σε ποιο συγκεκριμένα πεδίο της φόρμας.

Εδώ, βλέπουμε ενδεικτικά ορισμένες περιπτώσεις εισαγωγής μη κατάλληλων δεδομένων για την φόρμα, ξεκινώντας με την περίπτωση που επιλέγεται σε άδεια φόρμα το κουμπί της υποβολής.

ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΝΕΑΣ ΕΞΕΤΑΣΤΙΚΗΣ

ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ

ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ

ΠΕΡΙΟΔΟΣ ΕΞΕΤΑΣΤΙΚΗΣ

ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ

ΗΜΕΡ/ΝΙΑ ΕΝΑΡΞΗΣ

ΗΜΕΡ/ΝΙΑ ΛΗΞΗΣ

Δεν έχει εισαχθεί σωστά το Έτος Εξεταστικής!

Μήνας εξεταστικής μη συμβατός με την περίοδο εξεταστικής!

Δεν έχει εισαχθεί σωστά η Κατηγορία Εξεταστικής!

Δεν έχει εισαχθεί σωστά η Ημερομηνία (Εναρξης/Λήξης) Εξεταστικής!

Η εισαγωγή των δεδομένων δεν πραγματοποιήθηκε λόγω ασυμβατότητας των στοιχείων που εισήχθησαν!

ΥΠΟΒΟΛΗ ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ

ΕΞΟΔΟΣ ΕΠΙΣΤΡΟΦΗ

Εικόνα 34 - Περίπτωση Λάθους - Άδεια φόρμα κατά την καταχώρηση στοιχείων εξεταστικής.

Ενώ στην συνέχεια έχουμε παρακάτω την περίπτωση που τα δεδομένα είναι σωστά δομημένα και εισάγονται επιτυχώς στο σύστημα.

ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΝΕΑΣ ΕΞΕΤΑΣΤΙΚΗΣ

ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ

ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ
2023

ΠΕΡΙΟΔΟΣ ΕΞΕΤΑΣΤΙΚΗΣ
ΧΕΙΜΕΡΙΝΟ

ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ
ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗ

ΗΜΕΡ/ΝΙΑ ΕΝΑΡΞΗΣ
2023-2-10

ΗΜΕΡ/ΝΙΑ ΛΗΞΗΣ
2023-3-10

ΥΠΟΒΟΛΗ ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ

ΕΞΟΔΟΣ ΕΠΙΣΤΡΟΦΗ

ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ
ΕΤΟΣ

ΠΕΡΙΟΔΟΣ ΕΞΕΤΑΣΤΙΚΗΣ

ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ

ΗΜΕΡ/ΝΙΑ ΕΝΑΡΞΗΣ
ΕΤΟΣ-ΜΗΝΑΣ-ΜΕΡΑ

ΗΜΕΡ/ΝΙΑ ΛΗΞΗΣ
ΕΤΟΣ-ΜΗΝΑΣ-ΜΕΡΑ

ΥΠΟΒΟΛΗ ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ

ΕΙΣΑΓΩΓΗ ΜΑΘΗΜΑΤΩΝ ΕΞΕΤΑΣΤΙΚΗΣ ΕΞΟΔΟΣ ΕΠΙΣΤΡΟΦΗ

Η εισαγωγή των δεδομένων έγινε με επιτυχία!

Εικόνα 35 - Περίπτωση επιτυχούς καταχώρησης νέας Εξεταστικής.

Συμπληρωματικά εντελώς, ακολουθούν παρακάτω και άλλες περιπτώσεις λάθος εισαγωγής δεδομένων για την συγκεκριμένη φόρμα:

ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΝΕΑΣ ΕΞΕΤΑΣΤΙΚΗΣ

ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ

ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ
ΕΤΟΣ

ΠΕΡΙΟΔΟΣ ΕΞΕΤΑΣΤΙΚΗΣ

ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ

ΗΜΕΡ/ΝΙΑ ΕΝΑΡΞΗΣ
ΕΤΟΣ-ΜΗΝΑΣ-ΜΕΡΑ

ΗΜΕΡ/ΝΙΑ ΛΗΞΗΣ
ΕΤΟΣ-ΜΗΝΑΣ-ΜΕΡΑ

ΥΠΟΒΟΛΗ ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ

ΕΙΣΑΓΩΓΗ ΜΑΘΗΜΑΤΩΝ ΕΞΕΤΑΣΤΙΚΗΣ ΕΞΟΔΟΣ ΕΠΙΣΤΡΟΦΗ

Η εισαγωγή των δεδομένων δεν πραγματοποιήθηκε λόγω του ότι η εγγραφή υπάρχει ήδη στο σύστημα!

Εικόνα 36 - Περίπτωση Διπλότυπης Εγγραφής.

ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΝΕΑΣ ΕΞΕΤΑΣΤΙΚΗΣ

ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ

ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ
2022

ΠΕΡΙΟΔΟΣ ΕΞΕΤΑΣΤΙΚΗΣ
ΧΕΙΜΕΡΙΝΟ

ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ
ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗ

ΗΜΕΡ/ΝΙΑ ΕΝΑΡΞΗΣ
2023-2-10

ΗΜΕΡ/ΝΙΑ ΛΗΞΗΣ
2023-3-10

ΥΠΟΒΟΛΗ ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ

ΕΙΣΑΓΩΓΗ ΜΑΘΗΜΑΤΩΝ ΕΞΕΤΑΣΤΙΚΗΣ ΕΞΟΔΟΣ ΕΠΙΣΤΡΟΦΗ

Δεν έχει εισαχθεί σωστά το Έτος Εξεταστικής!

Η εισαγωγή των δεδομένων δεν πραγματοποιήθηκε λόγω ασυμβατότητας των στοιχείων που εισήχθησαν!

Εικόνα 37 - Περίπτωση Λάθους Έτους.

ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΝΕΑΣ ΕΞΕΤΑΣΤΙΚΗΣ

ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ

ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ
2023

ΠΕΡΙΟΔΟΣ ΕΞΕΤΑΣΤΙΚΗΣ
ΧΕΙΜΕΡΙΝΟ

ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ
ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗ

ΗΜΕΡ/ΝΙΑ ΕΝΑΡΞΗΣ
2023-3-10

ΗΜΕΡ/ΝΙΑ ΛΗΞΗΣ
2023-2-10

ΥΠΟΒΟΛΗ ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ

ΕΙΣΑΓΩΓΗ ΜΑΘΗΜΑΤΩΝ ΕΞΕΤΑΣΤΙΚΗΣ ΕΞΟΔΟΣ ΕΠΙΣΤΡΟΦΗ

Μήνας εξεταστικής μη συμβατός με την περίοδο εξεταστικής!

Δεν έχει εισαχθεί σωστά η Ημερομηνία (Έναρξης/Λήξης) Εξεταστικής!

Η εισαγωγή των δεδομένων δεν πραγματοποιήθηκε λόγω ασυμβατότητας των στοιχείων που εισήχθησαν!

Εικόνα 38 - Περίπτωση Λάθους Ημερομηνίας Έναρξης/Λήξης.

Σε αυτό το σημείο, ο χρήστης βλέπει πέρα από το βασικό μήνυμα λάθους και ένα επιπλέον, που σχετίζεται με την περίοδο και τον μήνα της ημερομηνίας, γιατί έχει προκαθοριστεί στο σύστημα εσωτερικά να είναι από Φλεβάρη μέχρι Μάρτιο η χειμερινή περίοδος και ως μήνας έναρξης ο Φλεβάρης και άρα ο Μάρτιος δεν αναγνωρίζεται ως κάτι που ανήκει στην χειμερινή εξεταστική περίοδο του τμήματος.

ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΝΕΑΣ ΕΞΕΤΑΣΤΙΚΗΣ

ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ

ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ
2023

ΠΕΡΙΟΔΟΣ ΕΞΕΤΑΣΤΙΚΗΣ
ΧΕΙΜΕΡΙΝΟ

ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ
ΕΞΕΤΑΣΤΙΚΗ ΕΣΑΜΗ

ΗΜΕΡ/ΝΙΑ ΕΝΑΡΞΗΣ
2023-6-10

ΗΜΕΡ/ΝΙΑ ΛΗΞΗΣ
2023-7-10

ΥΠΟΒΟΛΗ ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ

ΕΙΣΑΓΩΓΗ ΜΑΘΗΜΑΤΩΝ ΕΞΕΤΑΣΤΙΚΗΣ ΕΞΟΔΟΣ ΕΠΙΣΤΡΟΦΗ

ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΝΕΑΣ ΕΞΕΤΑΣΤΙΚΗΣ

ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ

ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ
ΕΤΟΣ

ΠΕΡΙΟΔΟΣ ΕΞΕΤΑΣΤΙΚΗΣ

ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ

ΗΜΕΡ/ΝΙΑ ΕΝΑΡΞΗΣ
ΕΤΟΣ-ΜΗΝΑΣ-ΜΕΡΑ

ΗΜΕΡ/ΝΙΑ ΛΗΞΗΣ
ΕΤΟΣ-ΜΗΝΑΣ-ΜΕΡΑ

ΥΠΟΒΟΛΗ ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ

ΜΑΘΗΜΑΤΩΝ ΕΞΕΤΑΣΤΙΚΗΣ ΕΞΟΔΟΣ ΕΠΙΣΤΡΟΦΗ

Δεν έχει εισαχθεί σωστά η Περίοδος Εξεταστικής!

Η εισαγωγή των δεδομένων δεν πραγματοποιήθηκε λόγω ασυμβατότητας των στοιχείων που εισήχθησαν!

Εικόνα 39 - Περίπτωση Λάθους Ημερομηνίας βάση Περιόδου.

ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΝΕΑΣ ΕΞΕΤΑΣΤΙΚΗΣ

ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ

ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ
2023

ΠΕΡΙΟΔΟΣ ΕΞΕΤΑΣΤΙΚΗΣ
ΧΕΙΜΕΡΙΝΟ

ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ
ΕΞΕΤΑΣΤΙΚΗ ΕΣΑΜΗ

ΗΜΕΡ/ΝΙΑ ΕΝΑΡΞΗΣ
2022-2-10

ΗΜΕΡ/ΝΙΑ ΛΗΞΗΣ
2022-3-10

ΥΠΟΒΟΛΗ ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ

ΕΙΣΑΓΩΓΗ ΜΑΘΗΜΑΤΩΝ ΕΞΕΤΑΣΤΙΚΗΣ ΕΞΟΔΟΣ ΕΠΙΣΤΡΟΦΗ

ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΝΕΑΣ ΕΞΕΤΑΣΤΙΚΗΣ

ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ

ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ
ΕΤΟΣ

ΠΕΡΙΟΔΟΣ ΕΞΕΤΑΣΤΙΚΗΣ

ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ

ΗΜΕΡ/ΝΙΑ ΕΝΑΡΞΗΣ
ΕΤΟΣ-ΜΗΝΑΣ-ΜΕΡΑ

ΗΜΕΡ/ΝΙΑ ΛΗΞΗΣ
ΕΤΟΣ-ΜΗΝΑΣ-ΜΕΡΑ

ΥΠΟΒΟΛΗ ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ

ΜΑΘΗΜΑΤΩΝ ΕΞΕΤΑΣΤΙΚΗΣ ΕΞΟΔΟΣ ΕΠΙΣΤΡΟΦΗ

Δεν έχει εισαχθεί σωστά το Έτος Εξεταστικής!

Η εισαγωγή των δεδομένων δεν πραγματοποιήθηκε λόγω ασυμβατότητας των στοιχείων που εισήχθησαν!

Εικόνα 40 - Περίπτωση Λάθους Ημερομηνίας βάση Έτους Εξεταστικής.

Εφόσον, γίνει η 1^η επιτυχής υποβολή της φόρμας και άρα υπάρχει έστω μια εξεταστική καταχωρημένη στο σύστημα μας, δίνεται η δυνατότητα μέσω μιας νέας επιλογής στην φόρμα που είναι η «**ΕΙΣΑΓΩΓΗ ΜΑΘΗΜΑΤΩΝ ΕΞΕΤΑΣΤΙΚΗΣ**» να έχει πρόσβαση ο χρήστης και σε μια δεύτερη φόρμα, μέσω της οποίας θα εισάγει και τα μαθήματα που θα είναι προς εξέταση, επιλέγοντας μάλιστα το Έτος της Εξεταστικής αλλά και τον Καθηγητή του συγκεκριμένου μαθήματος. Οι επιλογές επιστροφής σε προηγούμενη οθόνη και εξόδου από το σύστημα είναι επίσης διαθέσιμες, σε αυτήν αλλά και σε οποιαδήποτε από τις οθόνες του συστήματος μας.

Παρακάτω ακολουθεί η περίπτωση άδειας φόρμας και τι θα συμβεί αν πατηθεί η Υποβολή πριν εισαχθεί η απαραίτητη πληροφορία στην φόρμα.

ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΜΑΘΗΜΑΤΩΝ ΑΝΑ ΕΞΕΤΑΣΤΙΚΗ

ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ

ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ

ΕΤΟΣ

ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ

▼

ΕΠΩΝΥΜΟ ΔΙΔΑΣΚΟΝΤΑ

▼

ΤΙΤΛΟΣ ΜΑΘΗΜΑΤΟΣ

▼

ΥΠΟΒΟΛΗ

ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ

ΕΞΟΔΟΣ

ΕΠΙΣΤΡΟΦΗ

ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΜΑΘΗΜΑΤΩΝ ΑΝΑ ΕΞΕΤΑΣΤΙΚΗ

ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ

ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ

ΕΤΟΣ

ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ

▼

ΕΠΩΝΥΜΟ ΔΙΔΑΣΚΟΝΤΑ

▼

ΤΙΤΛΟΣ ΜΑΘΗΜΑΤΟΣ

▼

Δεν έχει εισαχθεί σωστά το Έτος Εξεταστικής!

ΥΠΟΒΟΛΗ

ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ

Δεν έχει εισαχθεί σωστά η Κατηγορία Εξεταστικής!

ΕΞΟΔΟΣ

ΕΠΙΣΤΡΟΦΗ

Δεν έχει εισαχθεί σωστά το Επώνυμο του Καθηγητή!

Δεν έχει εισαχθεί σωστά ο Τίτλος του Μαθήματος!

Η εισαγωγή των δεδομένων δεν πραγματοποιήθηκε λόγω ασυμβατότητας των στοιχείων που εισήχθησαν!

Εικόνα 41 - Περίπτωση άδειας φόρμας καταχώρησης μαθημάτων ανά εξεταστική περίοδο.

ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΜΑΘΗΜΑΤΩΝ ΑΝΑ ΕΞΕΤΑΣΤΙΚΗ

ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ

ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ
2022

ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ
ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗ ▼

ΕΠΩΝΥΜΟ ΔΙΔΑΣΚΟΝΤΑ
ΜΑΓΟΣ ▼

ΤΙΤΛΟΣ ΜΑΘΗΜΑΤΟΣ
ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΣ ▼

ΥΠΟΒΟΛΗ **ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ**

ΕΞΟΔΟΣ **ΕΠΙΣΤΡΟΦΗ**

ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΜΑΘΗΜΑΤΩΝ ΑΝΑ ΕΞΕΤΑΣΤΙΚΗ

ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ

ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ
ΕΤΟΣ

ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ
▼

ΕΠΩΝΥΜΟ ΔΙΔΑΣΚΟΝΤΑ
▼

ΤΙΤΛΟΣ ΜΑΘΗΜΑΤΟΣ
▼

ΥΠΟΒΟΛΗ **ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ**

ΕΞΟΔΟΣ **ΕΠΙΣΤΡΟΦΗ**

Δεν έχει εισαχθεί σωστά το Έτος Εξεταστικής!

Η περίοδος του εξαμήνου του Μαθήματος δεν αντιστοιχεί σε καμία Εξεταστική Περίοδο του Συστήματος!

Η εισαγωγή των δεδομένων δεν πραγματοποιήθηκε λόγω ασυμβατότητας των στοιχείων που εισήχθησαν!

Εικόνα 42 - Περίπτωση Λάθος Έτους - Καταχώρηση Μαθήματος Εξεταστικής.

Συμπληρωματικά μάλιστα, εδώ η ειδοποίηση που βγαίνει για την περίοδο του μαθήματος στον χρήστη, αφορά το μάθημα των Αλγόριθμων και Πολυπλοκότητας, το οποίο ανήκει σε Εαρινή Περίοδο, ενώ μέχρι στιγμής έχει καταχωρηθεί μόνο η Εξεταστική για την Χειμερινή Περίοδο και συνεπώς δεν θα μπορούσε έτσι και αλλιώς να δηλωθεί το συγκεκριμένο μάθημα.

Παρακάτω ακολουθεί η περίπτωση μαθήματος με περίοδο που δεν ανήκει στην επερχόμενη εξεταστική περίοδο, που να έχει δηλωθεί προηγουμένως στο σύστημα από την Γραμματεία.

<p>ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΜΑΘΗΜΑΤΩΝ ΑΝΑ ΕΞΕΤΑΣΤΙΚΗ</p> <p>ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ</p> <p>ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ 2023</p> <p>ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗ</p> <p>ΕΠΩΝΥΜΟ ΔΙΔΑΣΚΟΝΤΑ ΜΑΓΟΣ</p> <p>ΤΙΤΛΟΣ ΜΑΘΗΜΑΤΟΣ ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΚ</p> <p>ΥΠΟΒΟΛΗ ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ</p> <p>ΕΞΟΔΟΣ ΕΠΙΣΤΡΟΦΗ</p>	<p>ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΜΑΘΗΜΑΤΩΝ ΑΝΑ ΕΞΕΤΑΣΤΙΚΗ</p> <p>ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ</p> <p>ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ ΕΤΟΣ</p> <p>ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ</p> <p>ΕΠΩΝΥΜΟ ΔΙΔΑΣΚΟΝΤΑ</p> <p>ΤΙΤΛΟΣ ΜΑΘΗΜΑΤΟΣ</p> <p>ΥΠΟΒΟΛΗ ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ</p> <p>ΕΞΟΔΟΣ ΕΠΙΣΤΡΟΦΗ</p> <p>Η περίοδος του εξαμήνου του Μαθήματος δεν αντιστοιχεί σε καμία Εξεταστική Περίοδο του Συστήματος!</p> <p>Η εισαγωγή των δεδομένων δεν πραγματοποιήθηκε λόγω ασυμβατότητας των στοιχείων που εισήχθησαν!</p>
---	---

Εικόνα 43 - Περίπτωση μαθήματος με περίοδο που δεν ανήκει στην επερχόμενη εξεταστική περίοδο.

<p>ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΜΑΘΗΜΑΤΩΝ ΑΝΑ ΕΞΕΤΑΣΤΙΚΗ</p> <p>ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ</p> <p>ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ 2023</p> <p>ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ ΕΜΒΟΛΙΜΗ ΕΞΕΤΑΣΤ</p> <p>ΕΠΩΝΥΜΟ ΔΙΔΑΣΚΟΝΤΑ ΜΑΓΟΣ</p> <p>ΤΙΤΛΟΣ ΜΑΘΗΜΑΤΟΣ ΑΛΓΟΡΥΘΜΙΚΗ</p> <p>ΥΠΟΒΟΛΗ ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ</p> <p>ΕΞΟΔΟΣ ΕΠΙΣΤΡΟΦΗ</p>	<p>ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΜΑΘΗΜΑΤΩΝ ΑΝΑ ΕΞΕΤΑΣΤΙΚΗ</p> <p>ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ</p> <p>ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ ΕΤΟΣ</p> <p>ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ</p> <p>ΕΠΩΝΥΜΟ ΔΙΔΑΣΚΟΝΤΑ</p> <p>ΤΙΤΛΟΣ ΜΑΘΗΜΑΤΟΣ</p> <p>ΥΠΟΒΟΛΗ ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ</p> <p>ΕΞΟΔΟΣ ΕΠΙΣΤΡΟΦΗ</p> <p>Η περίοδος του εξαμήνου του Μαθήματος δεν αντιστοιχεί σε καμία Εξεταστική Περίοδο του Συστήματος!</p> <p>Η εισαγωγή των δεδομένων δεν πραγματοποιήθηκε λόγω ασυμβατότητας των στοιχείων που εισήχθησαν!</p>
---	---

Εικόνα 44 - Η κατηγορία εξεταστικής δεν υπάρχει στο σύστημα.

ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΜΑΘΗΜΑΤΩΝ ΑΝΑ ΕΞΕΤΑΣΤΙΚΗ

ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ

ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ

2023

ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ

ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗ ▼

ΕΠΩΝΥΜΟ ΔΙΔΑΣΚΟΝΤΑ

ΜΑΓΟΣ ▼

ΤΙΤΛΟΣ ΜΑΘΗΜΑΤΟΣ

ΑΛΓΟΡΥΘΜΙΚΗ ▼

ΥΠΟΒΟΛΗ

ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ

ΕΞΟΔΟΣ

ΕΠΙΣΤΡΟΦΗ

ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΜΑΘΗΜΑΤΩΝ ΑΝΑ ΕΞΕΤΑΣΤΙΚΗ

ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ

ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ

ΕΤΟΣ

ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ

▼

ΕΠΩΝΥΜΟ ΔΙΔΑΣΚΟΝΤΑ

▼

ΤΙΤΛΟΣ ΜΑΘΗΜΑΤΟΣ

▼

ΥΠΟΒΟΛΗ

ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ

ΕΞΟΔΟΣ

ΕΠΙΣΤΡΟΦΗ

Η εισαγωγή των δεδομένων έγινε με επιτυχία!

Εικόνα 45 - Επιτυχής Υποβολή Μαθήματος σε Εξεταστική.

Αφού ο χρήστης επιστρέψει στο αρχικό μενού της εφαρμογής, επιλέγει την 2^η επιλογή «**ΕΝΗΜΕΡΩΣΗ ΕΞΕΤΑΣΤΙΚΗΣ**» και περνάει σε μια άλλη οθόνη που ουσιαστικά αποτελεί την λειτουργία για την ενημέρωση των δεδομένων στις εξεταστικές που έχουν ήδη δηλωθεί. Το μόνο που έχει να κάνει ο χρήστης είναι να πάει στην εγγραφή που θέλει να επεξεργαστεί και να επιλέξει «**ΕΠΕΞΕΡΓΑΣΙΑ ΕΓΓΡΑΦΗΣ**», μετά να συμπληρώσει τα πεδία με τις διαφορετικές πληροφορίες όπως επιθυμεί και τέλος να επιλέξει την επιλογή «**ΕΝΗΜΕΡΩΣΗ**» για την αποθήκευση των αλλαγών και την ενημέρωση ουσιαστικά των πληροφοριών της συγκεκριμένης εξεταστικής περιόδου. Τα συγκεκριμένα βήματα φαίνονται πολύ εύκολα στις παρακάτω εικόνες που ακολουθούν.

ΕΝΗΜΕΡΩΣΗ ΕΞΕΤΑΣΤΙΚΗΣ

ΕΞΕΤΑΣΕΙΣ:

ΕΤΟΣ	ΠΕΡΙΟΔΟΣ	ΚΑΤΗΓΟΡΙΑ	ΕΝΑΡΞΗΣ	ΛΗΞΗΣ	
2023	ΧΕΙΜΕΡΙΝΗ	ΜΑΘΗΜΑ ΣΕ ΑΥΤΗΝ ΤΗΝ ΠΕΡΙΟΔΟ	10-02-2023	10-03-2023	ΕΠΕΞΕΡΓΑΣΙΑ ΕΓΓΡΑΦΗΣ

[ΕΞΟΔΟΣ](#) [ΕΠΙΣΤΡΟΦΗ](#)

ΕΝΗΜΕΡΩΣΗ ΕΞΕΤΑΣΤΙΚΗΣ

ΕΞΕΤΑΣΕΙΣ:

ΕΤΟΣ	ΠΕΡΙΟΔΟΣ	ΚΑΤΗΓΟΡΙΑ	ΕΝΑΡΞΗΣ	ΛΗΞΗΣ	
2023	ΕΑΡΙΝΗ	ΒΑΘΜΟΣ ΕΞΑΜΗΝΟΥ	10/2/2023	10/3/2023	ΕΝΗΜΕΡΩΣΗ

[ΕΞΟΔΟΣ](#) [ΕΠΙΣΤΡΟΦΗ](#)

ΕΝΗΜΕΡΩΣΗ ΕΞΕΤΑΣΤΙΚΗΣ

ΕΞΕΤΑΣΕΙΣ:

ΕΤΟΣ	ΠΕΡΙΟΔΟΣ	ΚΑΤΗΓΟΡΙΑ	ΕΝΑΡΞΗ	ΛΗΞΗ	
2023	ΕΑΡΙΝΗ	ΜΑΘΗΜΑ ΣΕ ΑΥΤΗΝ ΤΗΝ ΠΕΡΙΟΔΟ	10-02-2023	10-03-2023	ΕΠΕΞΕΡΓΑΣΙΑ ΕΓΓΡΑΦΗΣ

[ΕΞΟΔΟΣ](#) [ΕΠΙΣΤΡΟΦΗ](#)

Η ενημέρωσή σας έγινε με επιτυχία!

Εικόνα 46 - Ενημέρωση Εξεταστικής.

Η 3^η επιλογή στο αρχικό μενού της Γραμματείας «**ΕΝΗΜΕΡΩΣΗ ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ ΑΙΘΟΥΣΩΝ**», αποσκοπεί στην δέσμευση των αιθουσών του τμήματος για την χρήση τους μετέπειτα κατά την εξεταστική περίοδο από τους διάφορους φοιτητές του. Επομένως, επιλέγοντας την επιλογή στο αρχικό μενού «**ΕΝΗΜΕΡΩΣΗ ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ ΑΙΘΟΥΣΩΝ**», ο χρήστης θα μεταβεί στην συγκεκριμένη οθόνη η οποία όπως φαίνεται σε αρχική φάση μας δείχνει σε 3 πλαίσια τις διαθέσιμες αίθουσες, τα διαθέσιμα χρονοδιαγράμματα τους, καθώς και ποια από αυτά είναι δεσμευμένα, που στην προκειμένη περίπτωση δεν είναι κανένα ακόμα.

ΕΝΗΜΕΡΩΣΗ ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ ΑΙΘΟΥΣΩΝ

ΑΙΘΟΥΣΕΣ

ID	ΧΩΡΗΤΙΚΟΤΗΤΑ	ΚΕΝΑ ΑΝΑ ΘΕΣΕΙΣ	ΠΛΗΘΟΣ ΕΠΙΤΗ...	
AMPHIBIGINF...	150	ΜΙΑ ΚΕΝΗ ΘΕ...	2	ΔΙΑΘΕΣΙΜΟΤΗΤΑ
AMPHISMALLI...	50	ΜΙΑ ΚΕΝΗ ΘΕ...	2	ΔΙΑΘΕΣΙΜΟΤΗΤΑ
AMPHYSICS	200	ΜΙΑ ΚΕΝΗ ΘΕ...	2	ΔΙΑΘΕΣΙΜΟΤΗΤΑ

ΔΙΑΘΕΣΙΜΕΣ ΧΡΟΝΟΘΥΡΙΔΕΣ

ID	ΗΜΕΡΟΜΗΝΙΑ	ΕΝΑΡΞΗ	ΤΕΛΟΣ
----	------------	--------	-------

ΧΡΟΝΟΘΥΡΙΔΕΣ

ID ΤΑΞΗ	ID ΧΡΟΝΟΘΗΚΗ
---------	--------------

[ΕΞΟΔΟΣ](#) [ΕΠΙΣΤΡΟΦΗ](#)

Η παρακάτω εικόνα αποτελεί την αρχική κατάσταση της οθόνης για την ενημέρωση διαθεσιμότητας των αιθουσών και για να είναι ορατές ανά αίθουσα οι διαθέσιμες χρονοθυρίδες τους, δηλαδή το πότε είναι χρονικά διαθέσιμες για χρήση, πρέπει να γίνει η επιλογή του κουμπιού «**ΔΙΑΘΕΣΙΜΟΤΗΤΑ**».

ΕΝΗΜΕΡΩΣΗ ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ ΑΙΘΟΥΣΩΝ

ΑΙΘΟΥΣΕΣ

ID	ΧΩΡΗΤΙΚΟΤΗΤΑ	ΚΕΝΑ ΑΝΑ ΘΕΣΕΙΣ	ΠΛΗΘΟΣ ΕΠΙΘ...	
AMPHIBIGINF...	150	ΜΙΑ ΚΕΝΗ ΘΕ...	2	ΔΙΑΘΕΣΙΜΟΤΗΤΑ
AMPHISMALLI...	50	ΜΙΑ ΚΕΝΗ ΘΕ...	2	ΔΙΑΘΕΣΙΜΟΤΗΤΑ
AMPHYSICS	200	ΜΙΑ ΚΕΝΗ ΘΕ...	2	ΔΙΑΘΕΣΙΜΟΤΗΤΑ

ΔΙΑΘΕΣΙΜΕΣ ΧΡΟΝΟΘΥΡΙΔΕΣ

ID	ΗΜΕΡΟΜΗΝΙΑ	ΕΝΑΡΞΗ	ΤΕΛΟΣ	
1	16-02-2023	09:00:00	12:00:00	ΕΠΙΛΟΓΗ
2	27-06-2023	14:00:00	17:00:00	ΕΠΙΛΟΓΗ
5	22-09-2023	13:00:00	16:00:00	ΕΠΙΛΟΓΗ

ΧΡΟΝΟΘΥΡΙΔΕΣ

ID ΤΑΞΗ	ID ΧΡΟΝΟΘΗΚΗ	

ΕΞΟΔΟΣ
ΕΠΙΣΤΡΟΦΗ

Εικόνα 47 - Επιλογή Διαθεσιμότητας Αιθουσών.

Στην συνέχεια στα δεξιά μας είναι ορατές οι διαθέσιμες χρονοθυρίδες και υπάρχει η δυνατότητα μέσω της επιλογής «**ΕΠΙΛΟΓΗ**» να γίνει επιλογή όσων επιθυμεί ο χρήστης Γραμματεία από αυτές. Στην συνέχεια οι επιλεγμένες χρονοθυρίδες εμφανίζονται στο κάτω μέρος της οθόνης.

ΕΝΗΜΕΡΩΣΗ ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ ΑΙΘΟΥΣΩΝ

ΑΙΘΟΥΣΕΣ

ID	ΧΩΡΗΤΙΚΟΤΗΤΑ	ΚΕΝΑ ΑΝΑ ΘΕΣΕΙΣ	ΠΛΗΘΟΣ ΕΠΙΘ...	
AMPHIBIGINF...	150	ΜΙΑ ΚΕΝΗ ΘΕ...	2	ΔΙΑΘΕΣΙΜΟΤΗΤΑ
AMPHISMALLI...	50	ΜΙΑ ΚΕΝΗ ΘΕ...	2	ΔΙΑΘΕΣΙΜΟΤΗΤΑ
AMPHYSICS	200	ΜΙΑ ΚΕΝΗ ΘΕ...	2	ΔΙΑΘΕΣΙΜΟΤΗΤΑ

ΔΙΑΘΕΣΙΜΕΣ ΧΡΟΝΟΘΥΡΙΔΕΣ

ID	ΗΜΕΡΟΜΗΝΙΑ	ΕΝΑΡΞΗ	ΤΕΛΟΣ	
2	27-06-2023	14:00:00	17:00:00	ΕΠΙΛΟΓΗ
5	22-09-2023	13:00:00	16:00:00	ΕΠΙΛΟΓΗ

ΧΡΟΝΟΘΥΡΙΔΕΣ

ID ΤΑΞΗ	ID ΧΡΟΝΟΘΗΚΗ	
AMPHIBIGINFORMAT	1	ΑΦΑΙΡΕΣΗ

ΕΞΟΔΟΣ
ΕΠΙΣΤΡΟΦΗ

Η εγγραφή σας έγινε με επιτυχία!

Εικόνα 48 - Δέσμευση Χρονοδιαγράμματος Αίθουσας.

Αντιθέτως, αν πάλι είτε λόγω λάθους ή επειδή ο χρήστης έχει αλλάξει γνώμη, είναι επιθυμητή η αποδέσμευση ενός χρονοδιαγράμματος, υπάρχει στην διάθεση του χρήστη η επιλογή «**ΑΦΑΙΡΕΣΗ**». Έτσι η χρονοθυρίδα φεύγει από τις επιλεγμένες και ξαναεμφανίζεται πάνω δεξιά μαζί με τις υπόλοιπες διαθέσιμες, με συνοδεία ανάλογου μηνύματος, που μας ενημερώνει ότι η ενέργεια αυτή έγινε επιτυχώς, όπως αντίστοιχα και κατά την δέσμευση χρονοθυρίδας.

ΕΝΗΜΕΡΩΣΗ ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ ΑΙΘΟΥΣΩΝ

ΑΙΘΟΥΣΕΣ

ID	ΧΩΡΗΤΙΚΟΤΗΤΑ	ΚΕΝΑ ΑΝΑ ΘΕΣΕΙΣ	ΠΛΗΘΟΣ ΕΠΙΤΗ...	
AMPHIBIGINF...	150	ΜΙΑ ΚΕΝΗ ΘΕ...	2	ΔΙΑΘΕΣΙΜΟΤΗΤΑ
AMPHISMAILL...	50	ΜΙΑ ΚΕΝΗ ΘΕ...	2	ΔΙΑΘΕΣΙΜΟΤΗΤΑ
AMPHYSICS	200	ΜΙΑ ΚΕΝΗ ΘΕ...	2	ΔΙΑΘΕΣΙΜΟΤΗΤΑ

ΔΙΑΘΕΣΙΜΕΣ ΧΡΟΝΟΘΥΡΙΔΕΣ

ID	ΗΜΕΡΟΜΗΝΙΑ	ΕΝΑΡΞΗ	ΤΕΛΟΣ	
1	16-02-2023	09:00:00	12:00:00	ΕΠΙΛΟΓΗ
2	27-06-2023	14:00:00	17:00:00	ΕΠΙΛΟΓΗ
5	22-09-2023	13:00:00	16:00:00	ΕΠΙΛΟΓΗ

ΧΡΟΝΟΘΥΡΙΔΕΣ

ID ΤΑΣΗ	ID ΧΡΟΝΟΘΗΚΗ

ΕΞΟΔΟΣ
ΕΠΙΣΤΡΟΦΗ

Η διαγραφή έγινε με επιτυχία!

Εικόνα 49 - Αποδέσμευση Χρονοδιαγράμματος Αίθουσας.

Και τέλος, επιλέγοντας ο χρήστης την τελευταία επιλογή στο αρχικό μενού της Γραμματείας «**ΕΜΦΑΝΙΣΗ ΠΛΗΡΟΦΟΡΙΩΝ ΕΞΕΤΑΣΤΙΚΗΣ**», θα μεταβεί σε μια ακόμα νέα οθόνη μέσω της οποίας θα δει μια σύνοψη των όσων έχουν καταχωρηθεί μέχρι στιγμής στο σύστημα από τον ίδιο νωρίτερα, που ουσιαστικά μιλάμε για μια καρτέλα με πληροφορίες εξεταστικής καθώς και τις αίθουσες και χρονοθυρίδες που έχουν δεσμευτεί για χρήση στην συγκεκριμένη εξεταστική.

ΕΜΦΑΝΙΣΗ ΠΛΗΡΟΦΟΡΙΩΝ ΕΞΕΤΑΣΤΙΚΗΣ

ΕΞΕΤΑΣΕΙΣ:

ID	ΕΤΟΣ	ΠΕΡΙΟΔΟΣ	ΚΑΤΗΓΟΡΙΑ	ΕΝΑΡΞΗ	ΛΗΞΗ
1	2023	ΧΕΙΜΕΡΙΝΗ	ΜΑΘΗΜΑ ΣΕ ΑΥΤΗ...	10-02-2023	10-03-2023

ΑΙΘΟΥΣΕΣ:

ID ΑΙΘΟΥΣΑΣ	ID ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ
AMPHIBIGINFORMAT	1

ΕΞΟΔΟΣ
ΕΠΙΣΤΡΟΦΗ

Εικόνα 50 - Εμφάνιση Πληροφοριών Εξεταστικής.

6.2.2 Περίπτωση Χρήστη Φοιτητή.

Ο χρήστης Φοιτητής είναι ο κύριος χρήστης του συστήματος και ο λόγος για τον οποίο κατασκευάστηκε όλη η εφαρμογή «**UNIWA EXAM PLANNER**». Στην αρχική οθόνη του, όπως φαίνεται παρακάτω υπάρχει η επιλογή «**ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ**» και «**ΠΡΟΒΟΛΗ ΑΙΤΗΣΕΩΝ**» καθώς όπως είναι ορατό από πάνω τους αναγράφεται και το ID του συγκεκριμένου φοιτητή. Ο συγκεκριμένος φοιτητής είναι ο Παναγιώτης Καλαμάρας με ID ή καλύτερα Α.Μ. (Αριθμό Μητρώου) «**cs123412**».



Επιλέγοντας την 1^η επιλογή, ο χρήστης επισκέπτεται μια νέα οθόνη που ουσιαστικά γίνεται η δήλωση στις εξετάσεις.



Για να γίνει λοιπόν αυτό, θα πρέπει να επιλέξει ο χρήστης την επιλογή «**ΕΠΙΛΟΓΗ**» εφόσον έχει εξαρχής ορίσει ποιο από τα «**ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ**» επιθυμεί και συνεπώς σε τι είδος εξέτασης αναφέρεται και παρακάτω ακολουθεί η περίπτωση επιτυχούς εγγραφής για το Χειμερινό Εξάμηνο.

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	<input type="checkbox"/> ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗΝΟΥ <input type="checkbox"/> ΕΜΒΟΛΙΜΗ ΕΞΕΤΑΣΤΙΚΗ <input type="checkbox"/> ΕΞΕΤΑΣΤΙΚΗ ΓΙΑ ΔΙΟΡΘΩΣΗ ΒΑΘΜΟΥ
ICE1-3001	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ	

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ	ΕΠΙΛΟΓΗ
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	<input checked="" type="checkbox"/> ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗΝΟΥ	<input type="button" value="ΕΠΙΛΟΓΗ"/>
ICE1-3001	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ	<input type="checkbox"/>	<input type="button" value="ΕΠΙΛΟΓΗ"/>

Η ΕΓΓΡΑΦΗ ΣΑΣ ΠΡΑΓΜΑΤΟΠΟΙΗΘΗΚΕ!
Υπεύθυνος Καθηγητής : ΚΩΝΣΤΑΝΤΙΝΟΣ
Θέση : 00
Αίθουσα: AMPHIBIGINFORMAT
Ημερομηνία : 16-02-2023
Ωρα Έναρξης : 09:00:00
Ωρα Λήξης : 12:00:00

Εικόνα 53 - Επιτυχής Εγγραφή Μαθήματος (Γραμμική Αλγεβρα).

Αντίστοιχα με την προηγούμενη περίπτωση, παρακάτω έχουμε την επιτυχή εγγραφή για τον φοιτητή Αναστασόπουλο Γεώργιο στο Εαρινό Εξάμηνο και συγκεκριμένα στο 2^ο βάση του προγράμματος σπουδών για το 2023.

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ		
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ <input type="text"/>	ΕΠΙΛΟΓΗ
ICE1-3001	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΙ	ΕΠΙΛΟΓΗ

[ΕΞΟΔΟΣ](#) [ΕΠΙΣΤΡΟΦΗ](#)

Η ΕΓΓΡΑΦΗ ΣΑΣ ΠΡΑΓΜΑΤΟΠΟΙΗΘΗΚΕ!
Υπεύθυνος Καθηγητής: ΜΑΓΟΣ
Θέση : 00
Αίθουσα: AMPHIBIGINFORMAT ×
Ημερομηνία : 27-06-2023
Ώρα Έναρξης : 14:00:00
Ώρα Λήξης : 17:00:00

Έπειτα για τον φοιτητή Καλαμάρα Παναγιώτη αλλά και για τον Λικούρη Μιχαήλ, βλέπουμε παρακάτω την δήλωση τους στην εξεταστική Σεπτεμβρίου και συγκεκριμένα στο μάθημα Γραμμική Άλγεβρα:

- Καλαμάρας – Γραμμική Άλγεβρα - Σεπτέμβριος 2023

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗΝΟΥ ΕΜΒΟΛΙΜΗ ΕΞΕΤΑΣΤΙΚΗ ΕΞΕΤΑΣΤΙΚΗ ΓΙΑ ΔΙΟΡΘΩΣΗ ΒΑΘΜΟΥ <input checked="" type="checkbox"/> ΕΞΕΤΑΣΤΙΚΗ ΣΕΠΤΕΜΒΡΙΟΥ
ICE1-3001	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ	

ΕΞΟΔΟΣ
ΕΠΙΣΤΡΟΦΗ

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ	ΕΠΙΛΟΓΗ
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	ΕΞΕΤΑΣΤΙΚΗ ΣΕΠΤ	<input type="checkbox"/>
ICE1-3001	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ		<input type="checkbox"/>

ΕΞΟΔΟΣ
ΕΠΙΣΤΡΟΦΗ

Η ΕΓΓΡΑΦΗ ΣΑΣ ΠΡΑΓΜΑΤΟΠΟΙΗΘΗΚΕ
 Υπεύθυνος Καθηγητής: ΚΩΝΣΤΑΝΤΙΝΟΣ
 Ούση: 00
 Δίαιουσα: ΔΗΜΗΤΡΙΟΥ
 Ημερομηνία: 22-09-2023
 Ώρα Έναρξης: 13:00:00
 Ώρα Λήξης: 16:00:00

Εικόνα 55 - Επιτυχής Εγγραφή Σεπτέμβριος 2023 Γραμμική Άλγεβρα 1#.

- Λικούρης – Γραμμική Άλγεβρα – Σεπτέμβριος 2023

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ	ΕΠΙΛΟΓΗ
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ ΕΞΕΤΑΣΤΙΚΗ ΣΕΠΤ	ΕΠΙΛΟΓΗ
ICE1-1011	ΜΑΘΗΜΑΤΙΚΗ ΑΝΑΛΥΣΗ Ι	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ	ΕΠΙΛΟΓΗ
ICE1-3001	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗ...	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ	ΕΠΙΛΟΓΗ
ICE1-5002	ΑΛΓΟΡΥΘΜΙΚΗ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ	ΕΠΙΛΟΓΗ

ΕΞΟΔΟΣ **ΕΠΙΣΤΡΟΦΗ**

Η ΕΓΓΡΑΦΗ ΣΑΣ ΠΡΑΓΜΑΤΟΠΟΙΗΘΗΚΕ!
Υπεύθυνος Καθηγητής: ΚΩΝΣΤΑΝΤΙΝΟΣ
Θέση: 01
Λίθουσα: AMRHIBISINFORMAT ×
Ημερομηνία: 22-09-2023
Ώρα Έναρξης: 13:00:00
Ώρα Λήξης: 16:00:00

Εικόνα 56 - Επιτυχής Εγγραφή Σεπτέμβριος 2023 Γραμμική Άλγεβρα 2#.

Ακολουθεί η δήλωση της φοιτήτριας Καρακίτσου Ελένη για το μάθημα Γραμμική Άλγεβρα κατά την Εμβόλιμη Χειμερινής Περιόδου 2023.

- Καρακίτσου – Γραμμική Άλγεβρα - Εμβόλιμη Χειμερινό 2023

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	<div style="border: 1px solid gray; padding: 5px;"><div style="border-bottom: 1px solid gray; padding: 2px;">ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗΝΟΥ</div><div style="padding: 2px;"><input checked="" type="checkbox"/> ΕΜΒΟΛΙΜΗ ΕΞΕΤΑΣΤΙΚΗ</div><div style="border-bottom: 1px solid gray; padding: 2px;">ΕΞΕΤΑΣΤΙΚΗ ΓΙΑ ΔΙΟΡΘΩΣΗ ΒΑΘΜΟΥ</div><div style="padding: 2px;">ΕΞΕΤΑΣΤΙΚΗ ΣΕΠΤΕΜΒΡΙΟΥ</div></div>

ΕΞΟΔΟΣ ΕΠΙΣΤΡΟΦΗ

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	<div style="border: 1px solid gray; padding: 5px;"><div style="border-bottom: 1px solid gray; padding: 2px;">ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗΝΟΥ</div><div style="padding: 2px;"><input checked="" type="checkbox"/> ΕΜΒΟΛΙΜΗ ΕΞΕΤ</div><div style="border-bottom: 1px solid gray; padding: 2px;">ΕΞΕΤΑΣΤΙΚΗ ΓΙΑ ΔΙΟΡΘΩΣΗ ΒΑΘΜΟΥ</div><div style="padding: 2px;">ΕΞΕΤΑΣΤΙΚΗ ΣΕΠΤΕΜΒΡΙΟΥ</div></div> ΕΠΙΛΟΓΗ

ΕΞΟΔΟΣ ΕΠΙΣΤΡΟΦΗ

Η ΕΓΓΡΑΦΗ ΣΑΣ ΠΡΑΓΜΑΤΟΠΟΙΗΘΗΚΕ!
Υπεύθυνος Καθηγητής : ΚΩΝΣΤΑΝΤΙΝΟΣ
Θέση : 00
Αίθουσα: ΑΜΡΗΒΙΒΙΓΙΝΦΟΡΜΑΤ ×
Ημερομηνία : 16-02-2023
Ώρα Έναρξης : 09:00:00
Ώρα Λήξης : 12:00:00

Εικόνα 57 - Επιτυχής Εγγραφή Εμβόλιμης Εξεταστικής.

Όπως και η δήλωση για βελτίωση βαθμού στο μάθημα Α&Π (Αλγόριθμοι και Πολυπλοκότητα) για το Εαρινό Εξάμηνο 2023 από τον φοιτητή Λικούρη Μιχαήλ.

- Λικούρης - Α&Π - Βελτίωση Βαθμού Εαρινό 2023

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ	ΕΠΙΛΟΓΗ
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	<input type="text"/>	<input type="button" value="ΕΠΙΛΟΓΗ"/>
ICE1-1011	ΜΑΘΗΜΑΤΙΚΗ ΑΝΑΛΥΣΗ Ι	<input type="text"/>	<input type="button" value="ΕΠΙΛΟΓΗ"/>
ICE1-3001	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗ...	<input type="text"/>	<input type="button" value="ΕΠΙΛΟΓΗ"/>
ICE1-5002	ΑΛΓΟΡΥΘΜΙΚΗ	<input type="text"/>	<input type="button" value="ΕΠΙΛΟΓΗ"/>

ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗΝΟΥ
ΕΜΒΟΛΙΜΗ ΕΞΕΤΑΣΤΙΚΗ
 ΕΞΕΤΑΣΤΙΚΗ ΓΙΑ ΔΙΟΡΘΩΣΗ ΒΑΘΜΟΥ
ΕΞΕΤΑΣΤΙΚΗ ΣΕΠΤΕΜΒΡΙΟΥ

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ	ΕΠΙΛΟΓΗ
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	<input type="text"/>	<input type="button" value="ΕΠΙΛΟΓΗ"/>
ICE1-1011	ΜΑΘΗΜΑΤΙΚΗ ΑΝΑΛΥΣΗ Ι	<input type="text"/>	<input type="button" value="ΕΠΙΛΟΓΗ"/>
ICE1-3001	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗ...	<input type="text"/>	<input type="button" value="ΕΠΙΛΟΓΗ"/>
ICE1-5002	ΑΛΓΟΡΥΘΜΙΚΗ	<input type="text"/>	<input type="button" value="ΕΠΙΛΟΓΗ"/>

Η ΕΓΓΡΑΦΗ ΣΑΣ ΠΡΑΓΜΑΤΟΠΟΙΗΘΗΚΕ!
Υπεύθυνος Καθηγητής : ΜΑΓΟΣ
Θέση : 00
Αίθουσα: AMRHIBIGINFORMAT
Ημερομηνία : 27-06-2023
Ώρα Έναρξης : 14:00:00
Ώρα Λήξης : 17:00:00

Εικόνα 58 - Επιτυχής Εγγραφή Εξεταστικής Βελτίωσης Βαθμού.

Στην συνέχεια όμως, όπως φαίνεται παρακάτω υπάρχουν και ορισμένες περιπτώσεις κατά την δήλωση ενός μαθήματος στο σύστημα που προκύπτουν προβλήματα, όπως:

- Η δήλωση μαθήματος για εξεταστική που δεν υπάρχει ακόμη στο σύστημα.

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ	ΕΠΙΛΟΓΗ
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	<input type="text"/>	<input type="button" value="ΕΠΙΛΟΓΗ"/>
ICE1-3001	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ ΕΞΕΤΑΣΤΙΚΗ ΕΣΑΙ	<input type="button" value="ΕΠΙΛΟΓΗ"/>

ΔΕΝ ΥΠΑΡΧΕΙ ΕΞΕΤΑΣΤΙΚΗ ΣΤΟ ΣΥΣΤΗΜΑ ΓΙΑ ΝΑ ΔΗΛΩΘΕΙ ΤΟ ΣΥΓΚΕΚΡΙΜΕΝΟ ΜΑΘΗΜΑ! ×

Εικόνα 59 - Περίπτωση Δήλωσης Μαθήματος Εξεταστικής Περιόδου που δεν έχει οριστεί ακόμα.

- Η δήλωση μαθήματος, για καθηγητή ο οποίος δεν είναι περασμένος στο σύστημα.

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ	ΕΠΙΛΟΓΗ
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ ΕΞΕΤΑΣΤΙΚΗ ΕΣΑΙ	<input type="button" value="ΕΠΙΛΟΓΗ"/>
ICE1-3001	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ <input type="text"/>	<input type="button" value="ΕΠΙΛΟΓΗ"/>

ΠΡΟΣΠΑΘΕΙΤΕ ΝΑ ΔΗΛΩΣΕΤΕ ΜΑΘΗΜΑ ΓΙΑ ΤΟ ΟΠΟΙΟ ΔΕΝ ΥΠΑΡΧΕΙ ΚΑΘΗΓΗΤΗΣ ΠΟΥ ΝΑ ΣΧΕΤΙΖΕΤΑΙ ΜΕ ΤΗΝ ΕΞΕΤΑΣΗ ΣΤΟ ΣΥΣΤΗΜΑ! ×

Εικόνα 60 - Περίπτωση Δήλωσης Μαθήματος από Καθηγητή που δεν έχει οριστεί ακόμα.

- Περίπτωση λάθος εξαμήνου. Όπως η παρακάτω, στην οποία ο Παναγιώτης Καλαμάρας, φοιτητής 3^{ου} εξαμήνου (χειμερινό εξάμηνο) επιθυμεί να δηλώσει το μάθημα Αλγόριθμοι και Πολυπλοκότητα που ανήκει στο 2^ο εξάμηνο σπουδών (εαρινή περίοδο).

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ	ΕΠΙΛΟΓΗ
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	<input type="text"/>	<input type="button" value="ΕΠΙΛΟΓΗ"/>
ICE1-3001	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΙ	<input type="button" value="ΕΠΙΛΟΓΗ"/>

ΠΡΟΣΠΑΘΕΙΤΕ ΝΑ ΔΗΛΩΣΕΤΕ ΜΑΘΗΜΑ ΑΠΟ ΔΙΑΦΟΡΕΤΙΚΟ ΕΞΑΜΗΝΟ! ✕

Εικόνα 61 - Περίπτωση Δήλωσης Μαθήματος από άλλο Περίοδο (Εξάμηνο Φοιτητή Μεγαλύτερο του Μαθήματος).

- Η περίπτωση δήλωσης μαθήματος με μεγαλύτερο εξάμηνο από το εξάμηνο του φοιτητή. Όπου ο Γεώργιος Αναστασόπουλος στο 2^ο εξάμηνο επιθυμεί να δηλώσει το μάθημα Γραμμικής Άλγεβρας που είναι στο 3^ο εξάμηνο σπουδών.

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ	ΕΠΙΛΟΓΗ
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΙ	<input type="button" value="ΕΠΙΛΟΓΗ"/>
ICE1-3001	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ <input type="text"/>	<input type="button" value="ΕΠΙΛΟΓΗ"/>

ΠΡΟΣΠΑΘΕΙΤΕ ΝΑ ΔΗΛΩΣΕΤΕ ΜΑΘΗΜΑ ΑΠΟ ΔΙΑΦΟΡΕΤΙΚΟ ΕΞΑΜΗΝΟ! ✕

Εικόνα 62 - Περίπτωση Δήλωσης Μαθήματος από άλλο Περίοδο (Εξάμηνο Φοιτητή Μικρότερο του Μαθήματος).

- Η περίπτωση δήλωσης διπλότυπου μαθήματος για τον φοιτητή Παναγιώτη Καλαμάρα.

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ	ΕΠΙΛΟΓΗ
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	ΕΞΕΤΑΣΤΙΚΗ ΕΣΑΓ	ΕΠΙΛΟΓΗ
ICE1-3001	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ	ΕΠΙΛΟΓΗ

ΕΞΟΔΟΣ
ΕΠΙΣΤΡΟΦΗ

ΤΟ ΜΑΘΗΜΑ ΕΙΝΑΙ ΕΙΔΗ ΔΗΛΩΜΕΝΟ! ✕

Εικόνα 63 - Περίπτωση Διπλότυπης Δήλωσης Μαθήματος.

- Η περίπτωση να μην ικανοποιείται κάποιο από τα κριτήρια για το συγκεκριμένο Διαπιστευτήριο του Φοιτητή. Όπως και παρακάτω, όπου ο φοιτητής Λικούρης Μιχαήλ (5^{ου} εξαμήνου φοίτησης) δεν μπορεί να συμμετάσχει σε Εμβόλιμη Εξεταστική για το μάθημα Αλγόριθμοι και Πολυπλοκότητα (2^{ου} εξαμήνου) εφόσον δεν έχει περάσει το 9^ο εξάμηνο.

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ	ΕΠΙΛΟΓΗ
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ		ΕΠΙΛΟΓΗ
ICE1-1011	ΜΑΘΗΜΑΤΙΚΗ ΑΝΑΛΥΣΗ Ι		
ICE1-3001	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗ...		
ICE1-5002	ΑΛΓΟΡΥΘΜΙΚΗ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ	ΕΠΙΛΟΓΗ

ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗΝΟΥ

ΕΜΒΟΛΙΜΗ ΕΞΕΤΑΣΤΙΚΗ

ΕΞΕΤΑΣΤΙΚΗ ΓΙΑ ΔΙΟΡΘΩΣΗ ΒΑΘΜΟΥ

ΕΞΕΤΑΣΤΙΚΗ ΣΕΠΤΕΜΒΡΙΟΥ

ΕΞΟΔΟΣ
ΕΠΙΣΤΡΟΦΗ

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ		
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ [input type="text"]	ΕΠΙΛΟΓΗ
ICE1-1011	ΜΑΘΗΜΑΤΙΚΗ ΑΝΑΛΥΣΗ Ι	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ [input type="text"]	ΕΠΙΛΟΓΗ
ICE1-3001	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗ...	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ ΕΜΒΟΛΙΜΗ ΕΞΕΤ	ΕΠΙΛΟΓΗ
ICE1-5002	ΑΛΓΟΡΥΘΜΙΚΗ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ [input type="text"]	ΕΠΙΛΟΓΗ

ΕΞΟΔΟΣ

ΕΠΙΣΤΡΟΦΗ

ΤΟ ΕΞΑΜΗΝΟ ΕΙΝΑΙ ΚΑΤΩ ΑΠΟ ΤΟ 9ο, ΣΥΝΕΠΩΣ ΔΕΝ ΕΙΝΑΙ ΔΥΝΑΤΗ Η ΣΥΜΜΕΤΟΧΗ ΣΕ ΕΜΒΟΛΙΜΗ ΕΞΕΤΑΣΤΙΚΗ!

Εικόνα 64 - Περίπτωση Αποτυχίας Δήλωσης Μαθήματος σε Εμβόλιμη Εξεταστική (Φοιτητής κάτω του 9ου Εξαμήνου).

- Και τέλος, έχουμε περιπτώσεις προβλημάτων σε δηλώσεις για εξέταση βελτίωσης βαθμού. Όπως στην συνέχεια, όπου ο φοιτητής Λικούρης Μιχαήλ για το μάθημα Αλγόριθμοι και Πολυπλοκότητα είναι είδη 1 φορά εξεταζόμενος.

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ		
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ [input type="text"]	ΕΠΙΛΟΓΗ
ICE1-1011	ΜΑΘΗΜΑΤΙΚΗ ΑΝΑΛΥΣΗ Ι	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ [input type="text"]	ΕΠΙΛΟΓΗ
ICE1-3001	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗ...	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ ΕΞΕΤΑΣΤΙΚΗ ΓΙΑ ↓	ΕΠΙΛΟΓΗ
ICE1-5002	ΑΛΓΟΡΥΘΜΙΚΗ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ [input type="text"]	ΕΠΙΛΟΓΗ

ΕΞΟΔΟΣ

ΕΠΙΣΤΡΟΦΗ

ΤΟ ΜΑΘΗΜΑ ΕΧΕΙ ΕΠΑΝΕΞΕΤΑΣΘΕΙ ΓΙΑ ΒΕΛΤΙΩΣΗ ΒΑΘΜΟΥ ΕΙΔΗ 1 ΦΟΡΑ!

Εικόνα 65 - Περίπτωση Αποτυχίας Δήλωσης Μαθήματος σε Εξεταστική Βελτίωσης Βαθμού 1#.

- Και όταν επίσης, δεν είναι σε θέση να δηλώσει την αλγοριθμική, διότι έχει είδη 3 μαθήματα προς εξέταση για βελτίωση βαθμού, στα οποία πιθανόν να έχει εξεταστεί και σε παλιότερες εξεταστικές περιόδους του τμήματος.

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ	ΕΠΙΛΟΓΗ
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	<input type="checkbox"/>	<input type="checkbox"/>
ICE1-1011	ΜΑΘΗΜΑΤΙΚΗ ΑΝΑΛΥΣΗ Ι	<input type="checkbox"/>	<input type="checkbox"/>
ICE1-3001	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗ...	<input type="checkbox"/>	<input type="checkbox"/>
ICE1-5002	ΑΛΓΟΡΥΘΜΙΚΗ	<input type="checkbox"/>	<input type="checkbox"/>

Ο ΦΟΙΤΗΤΗΣ ΕΧΕΙ ΕΙΔΗ ΕΞΕΤΑΣΘΕΙ ΣΕ 3 ΜΑΘΗΜΑΤΑ ΓΙΑ ΒΕΛΤΙΩΣΗ ΒΑΘΜΟΥ! ×

Εικόνα 66 - Περίπτωση Αποτυχίας Δήλωσης Μαθήματος σε Εξεταστική Βελτίωσης Βαθμού 2#.

- Τέλος, η περίπτωση να μην υπάρχει διαθέσιμη αίθουσα (πράγμα αρκετά σπάνιο).

ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ

ΜΑΘΗΜΑΤΑ:

ID	ΤΙΤΛΟΣ	ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ	ΕΠΙΛΟΓΗ
ICE1-1002	ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ	<input type="checkbox"/>	<input type="checkbox"/>
ICE1-3001	ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ	<input type="checkbox"/>	<input type="checkbox"/>

ΟΛΕΣ ΟΙ ΑΙΘΟΥΣΕΣ ΕΙΝΑΙ ΔΕΣΜΕΥΜΕΝΕΣ ΓΙΑ ΤΗΝ ΠΑΡΟΥΣΑ ΕΞΕΤΑΣΤΙΚΗ ΠΕΡΙΟΔΟ! ×

Εικόνα 67 - Περίπτωση Ανεπαρκούς Διαθεσιμότητας Αιθουσών

Και τέλος, επιλέγοντας την «**ΠΡΟΒΟΛΗ ΑΙΤΗΣΕΩΝ**», ο χρήστης έχει την δυνατότητα τόσο να δει τις διαθέσιμες εγγραφές του στο σύστημα όσο και μέσω της επιλογής «**ΑΚΥΡΩΣΗ**» να τις αφαιρέσει, σε περίπτωση που κάτι έχει αλλάξει ή έχει κάνει κάποιο λάθος κατά την δημιουργία της συγκεκριμένης εγγραφής στο σύστημα. Η συγκεκριμένη διαδικασία φαίνεται εύκολα στις παρακάτω 2 εικόνες.

ΠΡΟΒΟΛΗ ΑΙΤΗΣΕΩΝ

ID ΕΞΕΤΑΣΤΙΚΗΣ	ID ΜΑΘΗΜΑΤΟΣ	ID ΑΙΘΟΥΣΑΣ	ΗΜΕΡΟΜΗΝΙΑ	ΩΡΑ ΕΝΑΡΞΗΣ	ΩΡΑ ΛΗΞΗΣ	ΑΙΘΟΥΣΑ	ΘΕΣΗ	
1	ICE1-1002	AMPHIBIGINFOR...	2023-02-16 00:00:...	09:00:00	12:00:00	AMPHIBIGINFOR...	00	ΑΚΥΡΩΣΗ

ΕΞΟΔΟΣ

ΕΠΙΣΤΡΟΦΗ

Εικόνα 68 - Οθόνη Προβολής Αιτήσεων Φοιτητή.

ΠΡΟΒΟΛΗ ΑΙΤΗΣΕΩΝ

ID ΕΞΕΤΑΣΤΙΚΗΣ	ID ΜΑΘΗΜΑΤΟΣ	ID ΑΙΘΟΥΣΑΣ	ΗΜΕΡΟΜΗΝΙΑ	ΩΡΑ ΕΝΑΡΞΗΣ	ΩΡΑ ΛΗΞΗΣ	ΑΙΘΟΥΣΑ	ΘΕΣΗ
----------------	--------------	-------------	------------	-------------	-----------	---------	------

ΕΞΟΔΟΣ

ΕΠΙΣΤΡΟΦΗ

Η ΕΞΕΤΑΣΗ ΑΚΥΡΩΘΗΚΕ!

Εικόνα 69 - Διαγραφή Δήλωσης Μαθήματος Φοιτητή.

6.2.3 Περίπτωση Χρήστη Καθηγητή.

Ο χρήστης Καθηγητής έχει μόνο μια ιδιότητα και αυτή είναι να παρακολουθεί ανά πάσα στιγμή το πόσοι και ποιοι φοιτητές έχουν συμμετοχή για τα μαθήματα του στις εξετάσεις. Αυτό λοιπόν γίνεται μέσα από την επιλογή «**ΕΜΦΑΝΙΣΗ ΑΙΤΗΣΕΩΝ ΦΟΙΤΗΤΩΝ**» στο αρχικό μενού, η οποία μας μεταφέρει σε μια ακόμα νέα οθόνη, όπου όπως πολύ καλά φαίνεται παρακάτω είναι ορατές στον χρήστη 2 καρτέλες, η «**ΕΞΕΤΑΣΕΙΣ**» και η «**ΦΟΙΤΗΤΕΣ**» για την εμφάνιση των εξεταστικών, αλλά και των φοιτητών ανά εξεταστική περίοδο στις οποίες ανήκουν τα μαθήματα που έχουν δηλώσει. Αυτό γίνεται λοιπόν, βάση της επιλογής «**ΕΜΦΑΝΙΣΗ**» δίπλα από κάθε μάθημα όπως φαίνεται παρακάτω.



ΔΙΑΘΕΣΙΜΕΣ ΑΙΤΗΣΕΙΣ ΦΟΙΤΗΤΩΝ

ΕΞΕΤΑΣΕΙΣ:

ID ΕΞΕΤΑΣΤΙΚΗΣ	ΠΕΡΙΟΔΟΣ	Κατηγορία Μαθήματ...	Ημερομηνία Έναρξη	Ημερομηνία Λήξη	
1	ΧΕΙΜΕΡΙΝΗ	Μάθημα σε αυτήν ...	10-02-2023	10-03-2023	ΕΜΦΑΝΙΣΗ

ΦΟΙΤΗΤΕΣ:

ID	Όνομα	Επώνυμο
----	-------	---------

ΕΞΟΔΟΣ

ΔΙΑΘΕΣΙΜΕΣ ΑΙΤΗΣΕΙΣ ΦΟΙΤΗΤΩΝ

ΕΞΕΤΑΣΕΙΣ:

ID ΕΞΕΤΑΣΤΙΚΗΣ	ΠΕΡΙΟΔΟΣ	Κατηγορία Μαθήματ...	Ημερομηνία Έναρξη	Ημερομηνία Λήξη	
1	ΧΕΙΜΕΡΙΝΗ	Μάθημα σε αυτών ...	10-02-2023	10-03-2023	ΕΜΦΑΝΙΣΗ

ΦΟΙΤΗΤΕΣ:

ID	Όνομα	Επώνυμο
cs123412	ΠΑΝΑΓΙΩΤΗΣ	ΚΑΛΑΜΑΡΑΣ

ΕΞΟΔΟΣ

Εικόνα 71 - Εμφάνιση Αιτήσεων Φοιτητών ανά Εξετάσεων Καθηγητή.

Και μέσα από την ολοκλήρωση της χρήσης του συστήματος με τον χρήστη Καθηγητή, ολοκληρώνεται ουσιαστικά και το παρών κεφάλαιο και η υπό ενότητα με τις οδηγίες χρήσης του συστήματος «**UNIWA EXAM PLANNER**».

ΤΕΛΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ

Το σύστημα «**UNIWA EXAM PLANNER**» έχει την καλύτερη δυνατή λειτουργία για την ταχύτερη εξυπηρέτηση των φοιτητών ως προς την δήλωση των μαθημάτων που θα εξεταστούν αλλά και την ενημέρωση του διδακτικού προσωπικού για τα ποσοστά παρακολούθησης των εξετάσεων, στα μαθήματα που διδάσκουν. Επιπλέον, πέρα από το σύστημα, αποτελεί αρκετά ικανοποιητική και η λειτουργία της βάσης δεδομένων «**examsdb**», η οποία και υλοποιήθηκε ώστε μέσα από την επικοινωνία της με το σύστημα, να χρησιμεύει στην αποθήκευση, ανάκληση, επεξεργασία αλλά και διαγραφή της περιττής πληροφορίας που ανταλλάσσεται μεταξύ του συστήματος και του τελικού χρήστη μέσω των επιλογών της γραφικής του διεπαφής.

Τα θετικά αποτελέσματα που προκύπτουν από την τελική εφαρμογή του συστήματος σε λειτουργία είναι τα εξής:

- a) Έχουμε καλύτερη οργάνωση της διάθεσης και χρήσης των απαραίτητων υποδομών από τους εμπλεκόμενους. Για παράδειγμα, ο κάθε Λέκτορας έχει την δυνατότητα μέσω της ανά πάσα στιγμή ενημέρωσης του από το σύστημα για το πλήθος των φοιτητών που θα παρακολουθήσουν τις εξετάσεις, να μπορεί καλύτερα και ταχύτερα να γνωρίζει και πόσες θέσεις χρειάζονται, ποιες αίθουσες είναι πιο κατάλληλες και πόσοι επιτηρητές χρειάζονται.
- b) Παρουσιάζεται ταχύτερη διεξαγωγή των εξετάσεων, εφόσον δεν απαιτούνται τόσοι πολλοί έλεγχοι ως προς την τήρηση του παρουσιολογίου των εξετάσεων.
- c) Υπάρχει μεγαλύτερη ευκολία στην τήρηση των μέτρων, εφόσον εφαρμόζεται και η καλύτερη κατανομή των φοιτητών μέσω του συστήματος στις διαθέσιμες αίθουσες του τμήματος.
- d) Και τέλος, έτσι υπάρχει και μεγαλύτερη διευκόλυνση στην παρακολούθηση των εξετάσεων από τους φοιτητές και ιδιαίτερα απ' όσους ανήκουν σε ευπαθείς ομάδες και υπό άλλες προϋποθέσεις πιθανόν να έχαναν τις εξετάσεις ενός μαθήματος ή και ολόκληρης της περιόδου.

Στην συνέχεια, το συγκεκριμένο σύστημα έχει και κάποιες προοπτικές για μελλοντική εξέλιξη και επέκταση του, εφόσον θα μπορούσε να επεκταθεί για την παροχή και άλλων υπηρεσιών οι οποίες θα αφορούσαν την δήλωση μαθημάτων κατά την έναρξη ενός εξαμήνου, αλλά και την εγγραφή σε εργαστηριακά τμήματα για μια περίοδο φοίτησης στο τμήμα.

ΠΑΡΑΡΤΗΜΑΤΑ – ΚΩΔΙΚΑΣ ΕΡΓΑΣΙΑΣ

ΠΑΡΑΡΤΗΜΑ Α'

Κώδικας από αρχείο «examsdb ver13.0.sql» :

```
#Exams_Database - Panos_Kalamaras - 15_4_2023 - Version - 13.0
#Setting Up the User of our Database.. (Commands changed a little bit!)
GRANT ALL PRIVILEGES ON *.* TO root@'localhost' WITH GRANT OPTION;
FLUSH PRIVILEGES;

DROP DATABASE examsdb;
CREATE DATABASE examsdb DEFAULT CHARACTER SET Utf8 COLLATE Utf8_general_ci;
USE examsdb;

#Table for the Exams Timeslots for the Exams..
CREATE TABLE IF NOT EXISTS timeslot (
DslotID INT AUTO_INCREMENT PRIMARY KEY, #A number for each Exam Timeslot Registry..
Dslot DATE NOT NULL , #The Exams Date..
TslotStart TIME NOT NULL, #The Exams Starting Timestamp..
TslotEnd TIME NOT NULL #The Exams Ending Timestamp..
);

#Table for the Classes..
CREATE TABLE IF NOT EXISTS classroom (
classID VARCHAR (50) NOT NULL PRIMARY KEY, #Class Codename..
classCapacity INT DEFAULT 0, #Class No of Seats (the pure number without the blank
seats)..
classMode INT DEFAULT 0, #A flag for the number of blank seats taken per seat.. (0:
1 blank seat between, 1: 2 blank seats between)..
classSupNo INT DEFAULT 0 #Min No of Supervisors per Classroom during an Exam..
);

#Table for the Classroom Usage Timeslots for the Exams..
CREATE TABLE IF NOT EXISTS classAvail (
recordnumber INT DEFAULT 0, #Record personal number.. (used instead of classroom
ImportanceRank)
```

```

classid VARCHAR (50) NOT NULL, #Class Codename..
dslotid INT DEFAULT 0, #A number for each Exam Timeslot Registry..
isallocated BOOLEAN DEFAULT false, #A flag whether the classroom being available is
available to be used or not.. (true:1 & false:0)
currentcapacity INT DEFAULT 0, #The Current Capacity of the Classroom.. (At First
Current=Initial Capacity from Classroom Table..)
periodavailability INT DEFAULT 0, #The Type of the Semester the class is available..
(0:Winter,1:Summer,2:September)
PRIMARY KEY(classid,dslotid),
FOREIGN KEY(classid) REFERENCES classroom(classID),
FOREIGN KEY(dslotid) REFERENCES timeslot(DslotID)
);

```

#Table for the Lessons..

```

CREATE TABLE IF NOT EXISTS lesson (
lesID VARCHAR(50) NOT NULL PRIMARY KEY, #Lesson Id in the system..
lesTitle VARCHAR(100) NOT NULL, #Lesson Title in the system..
lesSemester INT DEFAULT 0, #The Semester the lesson belongs..
lesPeriod INT DEFAULT 0 #A flag for the period the semester belongs to.. (0: Winter
Semester, 1: Spring Semester)..
);

```

#Table for the Students..

```

CREATE TABLE IF NOT EXISTS student (
studID VARCHAR(50) NOT NULL PRIMARY KEY, #Students ID in the system..
studName VARCHAR(100) NOT NULL, #Students Name in the system..
studPassword VARCHAR(20) NOT NULL, #Student password to gain access into the system..
studSurname VARCHAR(100) NOT NULL, #Students Surname in the system..
studSemester INT DEFAULT 0 #Students Semester..
);

```

#Table for the Professors..

```

CREATE TABLE IF NOT EXISTS professor (
profID VARCHAR(50) NOT NULL PRIMARY KEY, #Professors ID in the system..
profName VARCHAR(100) NOT NULL, #Professors Name in the system..
profPassword VARCHAR(50) NOT NULL, #Professor password to gain access into the
system..
profSurname VARCHAR(100) NOT NULL, #Professors Surname in the system..

```

```
supMODE INT DEFAULT 0 #A flag which tells us whether a Professor is also available
to be a Supervisor or not (0:False, 1:True)..
```

```
);
```

```
#Table for the Secretariat..
```

```
CREATE TABLE IF NOT EXISTS secretariat (  
secrID VARCHAR(50) NOT NULL PRIMARY KEY, #Secretary ID in the system..  
secrName VARCHAR(100) NOT NULL, #Secretary Name in the system..  
secrPassword VARCHAR(50) NOT NULL, #Secretary password to gain access into the  
system..
```

```
secrSurname VARCHAR(100) NOT NULL #Secretary Surname in the system..
```

```
);
```

```
#Table for the Exams..
```

```
CREATE TABLE IF NOT EXISTS exam (  
examId INT AUTO_INCREMENT PRIMARY KEY, #Exams ID in the system..  
examYear INT DEFAULT 0, #The Year the exams take place..  
examPeriod INT DEFAULT 0, #The Type of the Semester the exams take place..  
(0:Winter,1:Summer,2:September)
```

```
examType INT DEFAULT 0, #The Type of the Exams that take place.. (0:Lesson in this  
Period ,1:Lesson from another Period, 2:Lesson we want to Fix the Grade)
```

```
examStartDate DATE NOT NULL, #The Date the exams start..
```

```
examEndDate DATE NOT NULL #The Date the exams end..
```

```
);
```

```
CREATE UNIQUE INDEX unique_exam ON exam(examYear, examPeriod, examType,  
examStartDate,examEndDate); #TO SECURE THE UNIQUENESS OF A SINGLE LINE INTO THE  
TABLE..
```

```
#Table for the Lessons being passed by a Student..
```

```
CREATE TABLE IF NOT EXISTS lessonperstud(  
StudId VARCHAR(50) NOT NULL , #Students ID in the system..
```

```
LesId VARCHAR(50) NOT NULL , #Lesson Id in the system..
```

```
GotaTicket INT DEFAULT 0, #A Flag showing us whether a lesson is already registered  
for examination or not.. (0:False , 1:True)
```

```
Rexamedlesson INT DEFAULT 0, #A Flag the shows if our Lesson has been rexamed or not  
before.. (0:False , 1:True)
```

```
PRIMARY KEY (StudId,LesId),
```

```

FOREIGN KEY (StudId) REFERENCES student(studID),
FOREIGN KEY (LesId) REFERENCES lesson(lesID)
);

```

#Table for the Exams Students

```

CREATE TABLE IF NOT EXISTS examStud (
StudId VARCHAR(50) NOT NULL , #Students ID in the system..
ExamId INT DEFAULT 0 , #Exams ID in the system..
LesId VARCHAR(50) NOT NULL , #Lesson Id in the system..
ClassId VARCHAR (50) NOT NULL , #Class Codename..
StudSeat VARCHAR (20) DEFAULT '00', #Students Seat in the Classroom during the Exams
of a Lesson.. ("00","01","02","03" etc)
PRIMARY KEY (StudId,ExamId,LesId,ClassId),
FOREIGN KEY (StudId) REFERENCES student(studID),
FOREIGN KEY (ExamId) REFERENCES exam(examId),
FOREIGN KEY (LesId) REFERENCES lesson(lesID),
FOREIGN KEY (ClassId) REFERENCES classAvail(classid)
);

```

#Table for the Exams Professors/Supervisors

```

CREATE TABLE IF NOT EXISTS examProf (
ProfId VARCHAR(50) NOT NULL, #Professors ID in the system..
ExamId INT DEFAULT 0, #Exams ID in the system..
LesId VARCHAR(50) NOT NULL, #Lesson Id in the system..
PRIMARY KEY (ProfId,ExamId,LesId),
FOREIGN KEY (ProfId) REFERENCES professor(profID),
FOREIGN KEY (ExamId) REFERENCES exam(examId),
FOREIGN KEY (LesId) REFERENCES lesson(lesID)
);

```

#Table for the Exams Classrooms

```

CREATE TABLE IF NOT EXISTS examClass (
ClassId VARCHAR (50) NOT NULL, #Class Codename..
ExamId INT DEFAULT 0, #Exams ID in the system..
LesId VARCHAR(50) NOT NULL, #Lesson Id in the system..
ExamStudsNo INT DEFAULT 0, #The Number of Students they will Participate..
);

```

RankNo INT DEFAULT 0, #This number reveals the Importance of each Classroom and which we should use first from the others.. (0: Most Important, 1:Less and so on..)

PRIMARY KEY (ClassId,ExamId,LesId),

FOREIGN KEY (ClassId) REFERENCES classAvail(classid),

FOREIGN KEY (ExamId) REFERENCES exam(examId),

FOREIGN KEY (LesId) REFERENCES lesson(lesID)

);

#INSERT VALUES..

INSERT INTO student(studID, studName, studSurname, studSemester, studPassword) VALUES ('cs123412','XXX','XXX',3,'PanKal');

INSERT INTO student(studID, studName, studSurname, studSemester, studPassword) VALUES ('cs123413','XXX','XXX',2,'GeogAnas');

INSERT INTO student(studID, studName, studSurname, studSemester, studPassword) VALUES ('cs123414','XXX','XXX',5,'MikLiku');

INSERT INTO student(studID, studName, studSurname, studSemester, studPassword) VALUES ('cs123415','XXX','XXX',12,'ElKarak');

INSERT INTO professor(profID, profName, profSurname, profPassword, supMODE) VALUES ('cp148562','YYY','YYY','Mag123\$',0);

INSERT INTO professor(profID, profName, profSurname, profPassword, supMODE) VALUES ('cp248219','YYY','YYY','X@rKonst@22\$',1);

INSERT INTO secretariat(secrID, secrName, secrSurname, secrPassword) VALUES ('154256','ZZZ','ZZZ','1234Maria\$');

-- ΤΑ ΣΤΟΙΧΕΙΑ ΣΤΑ ΠΑΡΑΚΑΤΩ ΜΑΘΗΜΑΤΑ, ΠΙΘΑΝΟΝ ΝΑ ΕΙΝΑΙ ΚΑΙ ΠΛΑΣΜΑΤΙΚΑ. (πχ Κωδικοί Συστήματος/Εξάμηνα και Περίοδοι που ανήκουν)..

INSERT INTO lesson(lesID,lesTitle,lesPeriod,lesSemester) VALUES ('ICE1-3001','ΑΛΓΟΡΙΘΜΟΙ ΚΑΙ ΠΟΛΥΠΛΟΚΟΤΗΤΑ',1,2);

INSERT INTO lesson(lesID,lesTitle,lesPeriod,lesSemester) VALUES ('ICE1-1002','ΓΡΑΜΜΙΚΗ ΑΛΓΕΒΡΑ',0,3);

INSERT INTO lesson(lesID,lesTitle,lesPeriod,lesSemester) VALUES ('ICE1-5002','ΑΛΓΟΡΥΘΜΙΚΗ',0,1);

INSERT INTO lesson(lesID,lesTitle,lesPeriod,lesSemester) VALUES ('ICE1-1011','ΜΑΘΗΜΑΤΙΚΗ ΑΝΑΛΥΣΗ Ι',0,1);

INSERT INTO classroom (classID,classCapacity, classMode,classSupNo) VALUES ('AMPHIBIGINFORMAT',150,0,2); -- Μεγάλο Αμφιθέατρο

INSERT INTO classroom (classID,classCapacity, classMode,classSupNo) VALUES ('AMPHISMALLINFORMAT',50,0,2); -- Μικρό Αμφιθέατρο

INSERT INTO classroom (classID,classCapacity, classMode,classSupNo) VALUES ('AMPHYSICS',200,0,2); -- Αμφιθέατρο Φυσικής - Σαν Plan C!

INSERT INTO timeslot (Dslot, TslotStart, TslotEnd) VALUES ('2023-02-16', '9:00:00', '12:00:00');

```

INSERT INTO timeslot (Dslot, TslotStart, TslotEnd) VALUES ('2023-06-27','14:00:00','17:00:00');
INSERT INTO timeslot (Dslot, TslotStart, TslotEnd) VALUES ('2023-02-10','18:00:00','21:00:00');
INSERT INTO timeslot (Dslot, TslotStart, TslotEnd) VALUES ('2023-07-10','17:00:00','20:00:00');
INSERT INTO timeslot (Dslot, TslotStart, TslotEnd) VALUES ('2023-09-22','13:00:00','16:00:00');
INSERT INTO timeslot (Dslot, TslotStart, TslotEnd) VALUES ('2023-06-09','8:00:00','11:00:00');

INSERT INTO classAvail (recordnumber,classID, DslotID,periodavailability) VALUES (1,'AMPHIBIGINFORMAT',1,0);
INSERT INTO classAvail (recordnumber,classID, DslotID,periodavailability) VALUES (2,'AMPHIBIGINFORMAT',2,1);
INSERT INTO classAvail (recordnumber,classID, DslotID,periodavailability) VALUES (3,'AMPHIBIGINFORMAT',5,2);
INSERT INTO classAvail (recordnumber,classID, DslotID,periodavailability) VALUES (4,'AMPHISMALLINFORMAT',2,1);
INSERT INTO classAvail (recordnumber,classID, DslotID,periodavailability) VALUES (5,'AMPHISMALLINFORMAT',4,1);
INSERT INTO classAvail (recordnumber,classID, DslotID,periodavailability) VALUES (6,'AMPHYSICS',6,1);

INSERT INTO lessonperstud (StudId,LesId) VALUES ('cs123413','ICE1-3001'); -- Κανονική Εξέταση Χειμερινό
INSERT INTO lessonperstud (StudId,LesId) VALUES ('cs123413','ICE1-1002'); -- Κανονική Εξέταση Εαρινό
INSERT INTO lessonperstud (StudId,LesId) VALUES ('cs123412','ICE1-1002'); -- Επαναληπτική Σεπτεμβρίου - Εξαρχής Χειμερινού
INSERT INTO lessonperstud (StudId,LesId) VALUES ('cs123412','ICE1-3001'); -- Κανονική Εξέταση Εαρινό
INSERT INTO lessonperstud (StudId,LesId) VALUES ('cs123414','ICE1-3001'); -- Βελτίωση Βαθμού
INSERT INTO lessonperstud (StudId,LesId) VALUES ('cs123414','ICE1-1002'); -- Βελτίωση Βαθμού 2ο Μάθημα
INSERT INTO lessonperstud (StudId,LesId) VALUES ('cs123414','ICE1-5002'); -- Βελτίωση Βαθμού 3ο Μάθημα
INSERT INTO lessonperstud (StudId,LesId) VALUES ('cs123414','ICE1-1011'); -- Βελτίωση Βαθμού 4ο Μάθημα - Κάποιο θα μείνει εκτός!
INSERT INTO lessonperstud (StudId,LesId) VALUES ('cs123415','ICE1-1002'); -- Εξέταση Εμβόλιμη

```

#JUST FOR TESTING

select * from student;

select * from professor;

select * from secretariat;

select * from lesson;

select * from classroom;

select * from timeslot;

select * from lessonperstud;

ΠΑΡΑΡΤΗΜΑ Β'

Κώδικας από αρχείο «Application.java» :

```
package net.uniwa.uniwaexamsproj;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.autoconfigure.domain.EntityScan;
import
org.springframework.boot.context.properties.EnableConfigurationProperties;
import
org.springframework.boot.web.servlet.support.SpringBootServletInitializer;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

//The entry point of the Spring Boot application..
//Starting here with some useful annotations..
@SpringBootApplication //Marks the main class and entry point of our project..
@EnableConfigurationProperties()
@ComponentScan({"net.uniwa.uniwaexamsproj.*"}) //Tells Spring in which packages
you have annotated classes that should be managed by Spring..
@EntityScan(basePackages = {"net.uniwa.uniwaexamsproj.*"}) //Scan JPA Entities
and Enable Jpa Repositories for Database communications..
@EnableJpaRepositories("net.uniwa.uniwaexamsproj.*")
public class Application extends SpringBootServletInitializer {
public static void main(String[] args) {
SpringApplication.run(Application.class, args);
}
}
```


Κώδικας από αρχείο «LoginService.java» :

```
package net.uniwa.uniwaexamsproj.login;

import net.uniwa.uniwaexamsproj.professor.ProfessorService;
import net.uniwa.uniwaexamsproj.secretariat.SecretariatModel;
import net.uniwa.uniwaexamsproj.professor.ProfessorModel;
import net.uniwa.uniwaexamsproj.secretariat.SecretariatService;
import net.uniwa.uniwaexamsproj.student.StudentModel;
import net.uniwa.uniwaexamsproj.student.StudentService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class LoginService
{
    @Autowired
    StudentService stdsrv;
    @Autowired
    ProfessorService profsrv;
    @Autowired
    SecretariatService secrsrv;

    //The Methods of the Login being called from the Login Controller..
    public int GetAccess(String user, String pass) //Here we check according to
    the role of the user that logs in, in which menu we got to transfer the user..
    {
        int flag = 0; //Flag Values:0 for not found, 1 for Student, 2 for Professor &
        3 for Secretariat..
        flag = stdsrv.getAccess(user,pass); //Here we call the getaccess method in
        student service to get a flag value..

        if(flag == 0) //If the value is still 0, which means we don't have access as
        a Student..
        {
            flag = profsrv.getAccess(user,pass); //We call the same method from Professor
            Service to check if the user is a Professor..

            if(flag == 0) //If the value is still 0 and so the user is nor a Student
            neither a Professor, we call the method from Secretariat Service..
            {
                flag = secrsrv.getAccess(user,pass);
                return flag; //Here the flag has value 3 and the User is a Secretariat..
            }
            else
            {
                return flag; //Here the flag has value 2 and the User is a Professor..
            }
        }
        else
        {
            return flag; //Here the flag has value 1 and the User is a Student..
        }
    }

    public boolean isAllowed(Object user , String role) //Here we check if the
    user Object is instance of a specific role or not..
    {

        if(user == null ) return false;
    }
}
```

```

switch(role) {
case "Student": return user instanceof StudentModel;
//This returns true or false whether the object user is of instance
StudentModel or not..

case "Professor": return user instanceof ProfessorModel;
//This returns true or false whether the object user is of instance
ProfessorModel or not..

case "Secretariat": return user instanceof SecretariatModel; //This returns
true or false whether the object user is of instance SecretariatModel or not..

}
return false;
}
}

```

Κώδικας από αρχείο «SecretariatModel.java» :

```

package net.uniwa.uniwaexamsproj.secretariat;

import javax.persistence.*;

//This is the Class that has access on the Entity of the Table Secretariat in
our Database..
@Entity
@Table(name = "secretariat")
public class SecretariatModel
{
//Below we set the Columns..
@Id
@Column(name="secrID")
private String ID;

@Column(name="secrName")
private String Name;

@Column(name="secrSurname")
private String Surname;

@Column(name="secrPassword")
private String Password;

public String getID()
{
return ID;
}

public String getName()
{
return Name;
}

public String getSurname()
{
return Surname;
}
}

```

```
public String getPassword() {return Password;}
}
```

Κώδικας από αρχείο «SecretariatRepository.java» :

```
package net.uniwa.uniwaexamsproj.secretariat;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

//This class is the Repository of the Model Class for Table Secretariat..
@Repository
public interface SecretariatRepository extends
JpaRepository<SecretariatModel,String> //Here we pass the model class we will
manage and also the type of the Primary Key parameter..
{
//Contains all the CRUD Operations (Create,Read,Update,Delete) for the
Database..
}
```

Κώδικας από αρχείο «SecretariatService.java» :

```
package net.uniwa.uniwaexamsproj.secretariat;

import net.uniwa.uniwaexamsproj.classavailability.ClassAvailKey;
import net.uniwa.uniwaexamsproj.classavailability.ClassAvailModel;
import net.uniwa.uniwaexamsproj.classavailability.ClassAvailRepository;
import net.uniwa.uniwaexamsproj.classes.ClassModel;
import net.uniwa.uniwaexamsproj.classes.ClassRepository;
import net.uniwa.uniwaexamsproj.exam.ExamModel;
import net.uniwa.uniwaexamsproj.exam.ExamRepository;
import net.uniwa.uniwaexamsproj.examclass.ExamClassRepository;
import net.uniwa.uniwaexamsproj.examprof.ExamProfessorsKey;
import net.uniwa.uniwaexamsproj.examprof.ExamProfessorsModel;
import net.uniwa.uniwaexamsproj.examprof.ExamProfessorsRepository;
import net.uniwa.uniwaexamsproj.lesson.LessonModel;
import net.uniwa.uniwaexamsproj.lesson.LessonRepository;
import net.uniwa.uniwaexamsproj.professor.ProfessorModel;
import net.uniwa.uniwaexamsproj.professor.ProfessorRepository;
import net.uniwa.uniwaexamsproj.timeslot.TimeslotModel;
import net.uniwa.uniwaexamsproj.timeslot.TimeslotRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.*;
import javax.transaction.Transactional;

//This class is the Service which Uses the methods of the Repository class
given by it's Object..
@Service
public class SecretariatService
{
@Autowired
private SecretariatRepository secrep;

@Autowired
private ExamRepository examrep;
```

```

@Autowired
private ClassAvailRepository classavailrep;

@Autowired
private ExamClassRepository examclassrep;

@Autowired
private ExamProfessorsRepository examprofrep;

@Autowired
private ClassRepository classrep;

@Autowired
private TimeslotRepository timerep;

@Autowired
private LessonRepository lesrep;

@Autowired
private ProfessorRepository profrep;

public int getAccess(String user, String pass) //The GetAccess Method, same
as Student & Professor Service Class..
{
Optional<SecretariatModel> secretary = secrep.findById(user);

if(!secretary.isEmpty())
{
SecretariatModel secr = secretary.get();
if(pass.equals(secr.getPassword()))
{
return 3;
}
else
{
return 0;
}
}
else
{
return 0;
}
}

//Here we get the Available Timeslots of a Classroom using it's ID..
@Transactional
public List<TimeslotModel> getAvailableTimeslotsByClass(String classID) {
List<ClassAvailModel> alist = classavailrep.findAll();
List<TimeslotModel> timeslots = new ArrayList<>();

for(ClassAvailModel a : alist) {
if(a.getId().getClassID().equals(classID)&&!a.getAllocated())
timeslots.add(timerep.getById(a.getId().getDTslotID()));
}

return timeslots;
}

```

```

//Here we get the all the Allocated Timeslots of a Classroom by using it's
ID..
@Transactional
public List<ClassAvailModel> getAllocatedTimeslotsByClass(String classID) {
List<ClassAvailModel> alist = classavailrep.findAll();

alist.removeIf(e->          !e.getId().getClassID().equals(classID)      ||
!e.getAllocated());

return alist;
}

//Here we Save a ClassAvail Entity to the Classroom..
public void setclassavailinfo(String classID, Integer tSlotID, boolean b,
Integer Period)
{
//Here we INSERT these data into the ClassAvail Table!
ClassAvailModel classavailentity = new ClassAvailModel();
ClassAvailKey classavailkey = new ClassAvailKey();
Optional<TimeslotModel> tslotobj;
Optional<ClassModel> classobj;
classobj = classrep.findById(classID);
tslotobj = timerep.findById(tSlotID);
classavailentity.setClassavailobj(classobj);
classavailentity.setTslotavailobj(tslotobj);
classavailkey.setClassID(classID);
classavailkey.setDTslotID(tSlotID);
classavailentity.setId(classavailkey);
classavailentity.setAllocated(b);
classavailentity.setPeriodavail(Period);
Integer                                tmprecnum                =
classavailrep.findById(classavailkey).get().getRecordNo();
classavailentity.setRecordNo(tmprecnum);
classavailrep.save(classavailentity);
}
}

```

Κώδικας από αρχείο «StudentModel.java» :

```

package net.uniwa.uniwaexamsproj.student;

import net.uniwa.uniwaexamsproj.examstud.ExamStudentsModel;
import net.uniwa.uniwaexamsproj.lessonperstudent.LessonPerStudentModel;
import javax.persistence.*;
import java.util.Set;

//This is the Class that has access on the Entity of the Table Students in our
Database..

@Entity
@Table(name = "student")
public class StudentModel {

```

```

// Below we set the Columns..
@Id
@Column(name = "studID")
private String ID;

@Column(name = "studName")
private String Name;

@Column(name = "studSurname")
private String Surname;

@Column(name = "studSemester")
private Integer Semester;

@Column(name = "studPassword")
private String Password;

@OneToMany(mappedBy = "studobj", fetch = FetchType.EAGER)
private Set<ExamStudentsModel> examstud;

@OneToMany(mappedBy = "StudPassobj")
private Set<LessonPerStudentModel> lessonpass;

public String getID() {
return ID;
}

public String getName() {
return Name;
}

public String getSurname() {
return Surname;
}

public Integer getSemester() {

```

```

return Semester;
}

public String getPassword() {
return Password;
}

public Set<ExamStudentsModel> getExamstud() {
return examstud;
}

public Set<LessonPerStudentModel> getLessonpass() {
return lessonpass;
}
}

```

Κώδικας από αρχείο «StudentRepository.java» :

```

package net.uniwa.uniwaexamsproj.student;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

//This class is the Repository of the Model Class for Table Students..
@Repository
public interface StudentRepository extends JpaRepository<StudentModel,String> //Here
we pass the model class we will manage and also the type of the Primary Key parameter..
{
//Contains all the CRUD Operations (Create,Read,Update,Delete) for the Database..

//Here we find if the Student is of Semester 9th and Above, using his ID..
@Query(value = "SELECT COUNT(*) FROM student WHERE studSemester >= 9 AND studID =
?1", nativeQuery = true)
Integer SemesterInstantExamValidation(String StudID);

//Here we find the Student using his ID..
@Query(value = "SELECT * FROM student WHERE studID = ?1", nativeQuery = true)

```

```
StudentModel StudentDetails(String StudID);  
}
```

Κώδικας από αρχείο «StudentService.java» :

```
package net.uniwa.uniwaexamsproj.student;  
  
import net.uniwa.uniwaexamsproj.classavailability.ClassAvailKey;  
import net.uniwa.uniwaexamsproj.classavailability.ClassAvailModel;  
import net.uniwa.uniwaexamsproj.classavailability.ClassAvailRepository;  
import net.uniwa.uniwaexamsproj.classes.ClassModel;  
import net.uniwa.uniwaexamsproj.classes.ClassRepository;  
import net.uniwa.uniwaexamsproj.exam.ExamModel;  
import net.uniwa.uniwaexamsproj.exam.ExamRepository;  
import net.uniwa.uniwaexamsproj.examclass.ExamClassKey;  
import net.uniwa.uniwaexamsproj.examclass.ExamClassModel;  
import net.uniwa.uniwaexamsproj.examclass.ExamClassRepository;  
import net.uniwa.uniwaexamsproj.examprof.ExamProfessorsRepository;  
import net.uniwa.uniwaexamsproj.examstud.ExamStudentsKey;  
import net.uniwa.uniwaexamsproj.examstud.ExamStudentsModel;  
import net.uniwa.uniwaexamsproj.examstud.ExamStudentsRepository;  
import net.uniwa.uniwaexamsproj.lesson.LessonModel;  
import net.uniwa.uniwaexamsproj.lesson.LessonRepository;  
import net.uniwa.uniwaexamsproj.lessonperstudent.LessonPerStudentKey;  
import net.uniwa.uniwaexamsproj.lessonperstudent.LessonPerStudentModel;  
import net.uniwa.uniwaexamsproj.lessonperstudent.LessonPerStudentRepository;  
import net.uniwa.uniwaexamsproj.professor.ProfessorRepository;  
import net.uniwa.uniwaexamsproj.timeslot.TimeslotModel;  
import net.uniwa.uniwaexamsproj.timeslot.TimeslotRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import org.springframework.transaction.annotation.Transactional;  
  
import javax.validation.Valid;  
import java.text.SimpleDateFormat;  
import java.util.*;
```



```
//This class is the Service which Uses the methods of the Repository class given by  
it's Object..
```

```
@Service
```

```
public class StudentService {
```

```
@Autowired
```

```
private StudentRepository studrep;
```

```
@Autowired
```

```
private ExamRepository examrep;
```

```
@Autowired
```

```
private ExamStudentsRepository examstudrep;
```

```
@Autowired
```

```
private ExamProfessorsRepository examprofrep;
```

```
@Autowired
```

```
private ExamClassRepository examclassrep;
```

```
@Autowired
```

```
private ClassRepository classrep;
```

```
@Autowired
```

```
private ClassAvailRepository classavailrep;
```

```
@Autowired
```

```
private LessonPerStudentRepository lessonpassrep;
```

```
@Autowired
```

```
private LessonRepository lessonrepo;
```

```
@Autowired
```

```
private ProfessorRepository profrep;
```

```
@Autowired
```

```

private TimeslotRepository tslotrep;

SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");

private static String StudentID; //To save the ID of each student has access..

private static List<Integer> InitialCapacity = new ArrayList<>(); //To keep all the
Initial Capacities of the Classrooms..

private static List<Integer> CurrentCapacity = new ArrayList<>(); //To keep all the
Current Capacities of the Classrooms..

private static List<Integer[][]> Classes = new ArrayList<>(); //To keep all Classes
Status around the Marked and Free Seats inside them..

public int getAccess(String user, String pass) { //The same method as in the
Secretariat & Professor Service Class..

Optional<StudentModel> student = studrep.findById(user);

if (!student.isEmpty()) {
StudentModel stud = student.get();
if (pass.equals(stud.getPassword())) {
return 1;
} else {
return 0;
}
} else {
return 0;
}
}

public void InitiateSession(String ID) {
//Step 1st - Student ID Initiation
StudentID = ID; // Initiate ID of each user..
Integer NotFirstTimeFlag = 0;

//Step 2nd - Initial/Current Capacity Initiation
//Find the Classes that are Available and binded into the system and also calculate
the number..

List<ClassAvailModel> classavail = classavailrep.findAll();

```

```

Iterator<ClassAvailModel> itclassavail = classavail.listIterator();
//Here we Initiate the sizes and also inform the ClassAvail CurrentSizes..
while(itclassavail.hasNext())
{
ClassAvailModel tmpclassavail = itclassavail.next();
Integer periodavail = tmpclassavail.getPeriodavail();
ClassAvailKey tmpclassavailkey = tmpclassavail.getId();
Optional<ClassModel> tmpclass = classrep.findById(tmpclassavailkey.getClassID());

//Check if this is not the first time on the system initiating..
if(!examstudrep.findAll().isEmpty())
{
NotFirstTimeFlag = 1;
}

if(!tmpclass.isEmpty())
{
InitialCapacity.add(tmpclass.get().getCapacity());
CurrentCapacity.add(tmpclass.get().getCapacity());

if(NotFirstTimeFlag == 0) //We Initiate the ClassAvail Capacity with the Current only
the first time..
{
tmpclassavail.setPeriodavail(periodavail);
tmpclassavail.setCurrentcapacity(tmpclass.get().getCapacity());
classavailrep.save(tmpclassavail); //Initiate Current Capacity to Initial..
}
}
}

//Step 3rd - Classes List Initiation
//Initiate the Classes Structure List of 2D Integer Arrays..
itclassavail = classavail.listIterator(); //Reset the iterator..

while (itclassavail.hasNext())
{

```

```

Optional<ClassModel> tmpclass =
classrep.findById(itclassavail.next().getId().getClassID());

Integer Capacity = tmpclass.get().getCapacity(); //The total capacity without the
Blanks..

Integer RowSeats = 10; //Let's say each Class has 10 Seats per Row..
Integer Rows = (Capacity/RowSeats); //Here we calculate the Rows of each Class..
Integer[][] Class = new Integer[Rows][RowSeats]; //Init each Class..

//We initialize the whole array with '0'..
for (int i = 0; i < Rows; i++)
{
for (int j = 0; j < RowSeats; j++)
{
Class[i][j] = 0;
}
}

Classes.add(Class); //Inform the List with Each Array Class..
}

//Bonus Step - Case we restart the system..
//We Refresh the Data into the System Tables from the Database Tables..
//Read all the ExamStud..
List<ExamStudentsModel> examstuds = examstudrep.findAll();
Iterator<ExamStudentsModel> itexamstuds = examstuds.listIterator();

while(itexamstuds.hasNext())
{
ExamStudentsModel tmpexamstuds = itexamstuds.next();
String tmpSeat = tmpexamstuds.getStudentSeats();
String ClassID = tmpexamstuds.getExamstudkey().getClassID();
Integer ExamID = tmpexamstuds.getExamstudkey().getExamID();
String LessonID = tmpexamstuds.getExamstudkey().getLessonID();
ExamClassKey tmpexamclasskey = new ExamClassKey();
tmpexamclasskey.setClassID(ClassID);
tmpexamclasskey.setExamID(ExamID);
}

```

```

tmpexamclasskey.setLessonID(LessonID);
Integer i,j;

//First we calculate the coordinates of the Seat into the Class..
i = Integer.valueOf(tmpSeat.charAt(0) - '0');
j = Integer.valueOf(tmpSeat.charAt(1) - '0');

//Then we get the class itself..
Integer Position = examclassrep.GetRankNoByID(ClassID,ExamID,LessonID);

//We fill that Seat Position into the Class and we're done..
Integer[][] Classroom = Classes.get(Position);
Classroom[i][j]=1;

CurrentCapacity.add(Position,(examclassrep.findById(tmpexamclasskey).get().getClass
obj().getCapacity()-examclassrep.findById(tmpexamclasskey).get().getStudno()));
//Initial Capacity - StudNo => Current Capacity..
}

}

@Transactional
public String insertrecord(String StudID, Integer ExamID, String LessonID, Integer
Credential)
{
//For each kind of situation return a string and the rest of the program will handle
it and
//inform the final user for the state of it's registration..

//Check for Class Availability..
//If the Class is already Available into the ExamClass Table..
String ExamClassID = "";
Integer ClassAvailID = 0;
Integer classavailfound = 0;
String Seat = "";
Integer i, j;
Integer ClassPosition;
ExamClassModel tmpexamclassmodel = new ExamClassModel();

```

```

ClassAvailModel tmpclassavail = null;
Integer ValidSemester = studrep.SemesterInstantExamValidation(StudID);
Integer RexamCheck = lessonpassrep.LessonRexamCheck(StudID, LessonID);
Integer RexamCount = lessonpassrep.LessonRexamCount(StudID);
Integer LesSemester = lessonrepo.findById(LessonID).get().getSemester();
Integer StudSemester = studrep.getByID(StudID).getSemester();
String ProfID = examprofrep.FindProfID(ExamID, LessonID); //Find ProfID via
Exam/Lesson ID's..

if ((Credential.equals(0) && StudSemester.equals(LesSemester))
|| (Credential.equals(1) && ValidSemester.equals(1))
|| (Credential.equals(2) && RexamCheck.equals(0) && (RexamCount < 3))
|| (Credential.equals(3) && (StudSemester >= LesSemester)))
{
//Case ExamProf does not contain a record for the Professor and the Lesson in this
Specific Exam..
if (ProfID == null) {
return "PROFESSOR NOT FOUND";
}

if (lessonpassrep.LessonTicketCheck(StudID, LessonID).equals(1)) //Lesson already
assigned to the Student..
{
return "LESSON CHECKED";
}

ExamClassID = examclassrep.ClassAvailability(ExamID, LessonID);

//If we need to allocate one from ClassAvail Table..
if (ExamClassID == null)
{
ClassAvailID = classavailrep.ClassroomSearch(examrep.getByID(ExamID).getPeriod());
//We call, with the ExamPeriod, given by the FindByID() into Exam Repository using
ExamID..

if (ClassAvailID == null) //No Classroom is available..
{
//No Classroom Available..

```

```

return "NO CLASS AVAILABLE";
}
else //We have found a new Class from ClassAvail..
{
//Setting up ExamClassID parameter from before & ExamClassModel Object..
classavailfound = 1;
ExamClassID = classavailrep.findById(ClassAvailID);
examclassrep.InsertExamClass(ExamClassID, ExamID, LessonID, ClassAvailID);
classavailrep.UpdateRecordNo(ClassAvailID); //In order not to assign two times the
same classroom for another lesson..
}
}
else
{
tmpexamclassmodel = examclassrep.ClassSearchByKey(ExamClassID, ExamID, LessonID);
}

//Calculate the Position..
if (classavailfound.equals(1)) {
ClassPosition = ClassAvailID; //Since we have found ClassAvailID which is RecordNo,
we provide it here.. (ExamClass Record Doesn't Exist!)
} else //In this case, we have to search for the RecordNo.. (ExamClass Record Already
Exist!)
{
ClassPosition = tmpexamclassmodel.getRank(); //Calculate the position of the Class
Element inside the List<Integer[][]>..
}

Integer ReducedCapacity =
(classavailrep.GetByRecordNo(ClassPosition).getCurrentCapacity() - 1);
CurrentCapacity.add(ClassPosition, ReducedCapacity); //Inform the List..
Integer[][] TempClass = Classes.get(ClassPosition); // Get the Class Entity..
Integer Row = (InitialCapacity.get(ClassPosition) / 10); //Here we calculate the rows
& let's say the j is 10..

outerloop:
for (i = 0; i < Row; i++) {

```

```

for (j = 0; j < 10; j++) {
if (TempClass[i][j] == 0) {
TempClass[i][j] = 1; // Get this Seat from the Table of the Class..
Seat = new String(i.toString() + j.toString());
break outerloop;
}
}
}

examclassrep.ReduceCapacity(ClassPosition); //Here we increase ExamStudNo by 1.

classavailrep.UpdateCapacity(ClassPosition, ReducedCapacity); //Update Class Current
Capacity in ClassAvail Table by (-1) using RankNo..

examstudrep.SaveExamStud(StudID, ExamID, LessonID, ExamClassID, Seat); //Here we Save
ExamStud Record to the Table..

lessonpassrep.LessonRegistration(StudID, LessonID); //Here we inform the database
that the Lesson has been registered!

if (Credential.equals(2))
{
lessonpassrep.LessonRexamAssignment(StudID, LessonID); //Here we assign the rexamed
field of the lesson [Credential == 2]
}

//Here we prepare the Final Message & Return this to be printed into the view to the
user..

if (classavailfound.equals(1))
{
tmpclassavail = classavailrep.GetByRecordNo(ClassAvailID); //Here we Find via
RecordNo the ClassAvail Record..}
}
else
{
tmpclassavail = classavailrep.GetByRecordNo(ClassPosition); //Here we Find via Rank
the ClassAvail Record..}
}

```



```

return "Υπεύθυνος Καθηγητής : " + profrep.findById(ProfID).get().getSurname()
+ "</br>Θέση : " + Seat
+ "</br>Αίθουσα: " + tmpclassavail.getId().getClassID()
+ "</br>Ημερομηνία : " + sdf.format(tmpclassavail.getTslotavailobj().getDate())
+ "</br>Ωρα Έναρξης : " + tmpclassavail.getTslotavailobj().getStart()
+ "</br>Ωρα Λήξης : " + tmpclassavail.getTslotavailobj().getEnd();

}

else //Here we send multiple error messages, according to the occasion..
{
//Case we haven't the same semester as the lesson we want to sign for..
if (Credential.equals(0) || Credential.equals(3))
{
return "NOT THE SAME SEMESTER";
}

else if (Credential.equals(1)) //Case we have Credential 1 & We are under 9th
Semester..
{
return "UNDER 9TH SEMESTER";
}

else if (Credential.equals(2) && RexamCheck.equals(1))
{
return "LESSON REXAMED";
}

else if (Credential.equals(2) && (RexamCount == 3))
{
return "ALREADY 3 LESSONS REXAMED";
}

}

return ""; //A completely for typical use return..
}

@Transactional

public void deleterecord(String StudID, Integer ExamID, String LessonID, String
ClassID) {

//Get the Seat for further Calculations before deleting ExamStud Record..

```

```

String DeletedSeat = examstudrep.GetSeatInUse(StudID, ExamID, LessonID, ClassID);

examstudrep.DeleteExamStud(StudID, ExamID, LessonID, ClassID);

//Set GotaTicket to 0 from Lesson Per Student Table..
lessonpassrep.LessonRelease(StudID, LessonID);

//Get ExamClassKey and ExamClassModel for further calculations..
ExamClassModel tmpexamclass = examclassrep.ClassSearchByKey(ClassID, ExamID,
LessonID);

Integer ClassPosition = tmpexamclass.getRank(); //Get the Position of the Class
Entity..
Integer[][] TempClass = Classes.get(ClassPosition); //Get the Class Entity..
Integer i, j;

//Here we get the position of the Classroom Seat..
i = Integer.valueOf(DeletedSeat.charAt(0) - '0');
j = Integer.valueOf(DeletedSeat.charAt(1) - '0');

//Make the seat back to available..
TempClass[i][j] = 0;
Integer IncreasedCapacity=0;

IncreasedCapacity = (CurrentCapacity.get(ClassPosition) + 1); //Increase the Current
Capacity by 1..
CurrentCapacity.add(ClassPosition, IncreasedCapacity); //Inform the List..

if(tmpexamclass.getStudno() >= 0)
{
examclassrep.IncreaseCapacity(ClassPosition); //We just Reduce ExamStudNo by 1..
}

//If the class is full, we can delete the record from examclass and release the
allocation from classavail..
if (IncreasedCapacity <= tmpexamclass.getClassobj().getCapacity()) //Capacity equal
to the Initial Capacity of the class.. (Class Empty)
{

```

```

classavailrep.UpdateCapacity(ClassPosition,IncreasedCapacity); //Update Class
Current Capacity in ClassAvail Table by (+1) using RankNo..
}

//Erase the Record and set IsAllocated to false into the Exam Table..
lessonpassrep.LessonRelease(StudID,LessonID);
}

//Here get the Available Class that has a specific ClassID..
public ClassAvailModel getClassAvail(List<ClassAvailModel> classavail, String
ClassID, Integer ExamID, String LessonID) {

Integer RankNumber = examclassrep.GetRankNoByID(ClassID,ExamID,LessonID);

Iterator<ClassAvailModel> itclassavail = classavail.listIterator();

while (itclassavail.hasNext()) {
ClassAvailModel tmpclassavail = itclassavail.next();

if (tmpclassavail.getRecordNo().equals(RankNumber)) {
return tmpclassavail;
}
}

return null;
}
}

```

Κώδικας από αρχείο «ProfessorModel.java» :

```
package net.uniwa.uniwaexamsproj.professor;

import net.uniwa.uniwaexamsproj.examprof.ExamProfessorsModel;
import javax.persistence.*;
import java.util.Set;

//This is the Class that has access on the Entity of the Table Professors in our
Database..

@Entity
@Table(name = "professor")
public class ProfessorModel
{
//Below we set the Columns..

@Id
@Column(name="profID")
private String ID;

@Column(name="profName")
private String Name;

@Column(name="profSurname")
private String Surname;

@Column(name="profPassword")
private String Password;

@Column(name="supMODE")
private String SupervisorMode;

@OneToMany(mappedBy = "profobj")
private Set<ExamProfessorsModel> examprof;

public void setExamprof(Set<ExamProfessorsModel> examprof) {this.examprof =
examprof;}
```

```
public String getID()
{
return ID;
}
```

```
public String getName()
{
return Name;
}
```

```
public String getSurname()
{
return Surname;
}
```

```
public String getPassword()
{
return Password;
}
```

```
public String getSupervisorMode()
{
return SupervisorMode;
}
```

```
public Set<ExamProfessorsModel> getExamprof() {return examprof;}
}
```

Κώδικας από αρχείο «ProfessorRepository.java» :

```
package net.uniwa.uniwaexamsproj.professor;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

import java.util.ArrayList;
import java.util.List;

//This class is the Repository of the Model Class for Table Professors..
@Repository
public interface ProfessorRepository extends JpaRepository<ProfessorModel,String>
//Here we pass the model class we will manage and also the type of the Primary Key
parameter..
{
//Here we find all the Professor Surnames..
@Query(value = "SELECT profSurname FROM professor", nativeQuery = true)
ArrayList<String> findprofsurname();

//Here we find the Professor ID with a Specific Surname..
@Query(value = "SELECT profID FROM professor WHERE profSurname=?1", nativeQuery =
true)
String findprofid(String profSurnam);
}
```

Κώδικας από αρχείο «ProfessorService.java» :

```
package net.uniwa.uniwaexamsproj.professor;

import net.uniwa.uniwaexamsproj.examclass.ExamClassRepository;
import net.uniwa.uniwaexamsproj.examprof.ExamProfessorsRepository;
import net.uniwa.uniwaexamsproj.examstud.ExamStudentsRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.Optional;

//This class is the Service which Uses the methods of the Repository class given by
it's Object..

@Service
public class ProfessorService
{
    @Autowired
    private ProfessorRepository profrep;

    @Autowired
    private ExamStudentsRepository examstudrep;

    @Autowired
    private ExamProfessorsRepository examprofrep;

    @Autowired
    private ExamClassRepository examclassrep;

    public int getAccess(String user, String pass) //Here we check if the user with
    username "user" and password "pass", exists in the database as Professor and the
    id's are valid to getaccess or not..
    {
        Optional<ProfessorModel> professor = profrep.findById(user);

        if(!professor.isEmpty())
```

```

{
ProfessorModel prof = professor.get();

if(pass.equals(prof.getPassword())) //Case Username and Password are valid &
Professor exists as user in the Registry..

{
return 2;
}

else //Case the Password is incorrect..

{
return 0;
}
}

else //Case the Professor doesn't exists at all..

{
return 0;
}
}
}
}

```

Κώδικας από αρχείο «ClassAvailKey.java» :

```

package net.uniwa.uniwaexamsproj.classavailability;

import javax.persistence.Column;
import javax.persistence.Embeddable;
import java.io.Serializable;

@Embeddable
public class ClassAvailKey implements Serializable
{
@Column(name = "classid")
private String classID;

@Column(name = "dslotid")
private Integer DTslotID;

public void setClassID(String classID) {this.classID = classID;}

public void setDTslotID(Integer DTslotID) {this.DTslotID = DTslotID;}

public String getClassID()
{
return classID;
}

public Integer getDTslotID() {return DTslotID;}

```



```
}
```

Κώδικας από αρχείο «ClassAvailModel.java» :

```
package net.uniwa.uniwaexamsproj.classavailability;

import net.uniwa.uniwaexamsproj.classes.ClassModel;
import net.uniwa.uniwaexamsproj.timeslot.TimeslotModel;
import javax.persistence.*;
import java.util.Optional;

@Entity
@Table (name = "classAvail")
public class ClassAvailModel
{
    @EmbeddedId
    private ClassAvailKey id;

    @ManyToOne
    @MapsId("classID")
    @JoinColumn(name = "classid")
    private ClassModel classavailobj;

    @ManyToOne
    @MapsId("DTslotID")
    @JoinColumn(name = "dslotid")
    private TimeslotModel tslotavailobj;

    @Column(name="recordnumber")
    private Integer recordNo;

    @Column(name="isallocated")
    private boolean isAllocated;

    @Column(name="currentcapacity")
    private Integer currentcapacity;

    @Column(name="periodavailability")
    private Integer periodavail;

    public void setId(ClassAvailKey id) {
        this.id = id;
    }

    public void setClassavailobj(Optional<ClassModel> classavailobj) {
        this.classavailobj = classavailobj.get();
    }

    public void setTslotavailobj(Optional<TimeslotModel> tslotavailobj) {
        this.tslotavailobj = tslotavailobj.get();
    }

    public ClassAvailKey getId()
    {
        return id;
    }

    public ClassModel getClassavailobj() {return classavailobj;}
}
```

```

public TimeslotModel getTslotavailobj() {return tslotavailobj;}

public boolean getAllocated() {
return isAllocated;
}

public void setAllocated(boolean isAllocated) {
this.isAllocated = isAllocated;
}

public Integer getRecordNo() {return recordNo;}

public void setRecordNo(Integer recordNo) {this.recordNo = recordNo;}

public Integer getCurrentcapacity() {
return currentcapacity;
}

public void setCurrentcapacity(Integer currentcapacity) {
this.currentcapacity = currentcapacity;
}

public Integer getPeriodavail() {return periodavail;}

public void setPeriodavail(Integer periodavail) {this.periodavail =
periodavail;}
}

```

Κώδικας από αρχείο «ClassAvailRepository.java» :

```

package net.uniwa.uniwaexamsproj.classavailability;

import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Modifying;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

@Repository
public interface ClassAvailRepository extends JpaRepository<ClassAvailModel,
ClassAvailKey> {
//Query to find ClassAvailPeriod by ID..
@Query(value = "SELECT periodavailability FROM classAvail WHERE classid=?1 AND
dslotid=?2", nativeQuery = true)
Integer getPeriodbyID(String classID,Integer dslotID);

//Query to find the Classid using RecordNumber..
@Query(value = "SELECT classid FROM classAvail WHERE recordnumber=?1",
nativeQuery = true)
String findById(Integer classAvailID);

//Query to find the ClassAvailModel which contains RecordNumber..
@Query(value = "SELECT * FROM classAvail WHERE recordnumber=?1", nativeQuery
= true)
ClassAvailModel GetByRecordNo(Integer RecNo);

//Query to find the ClassAvailModels that are allocated or not..
@Query(value = "SELECT * FROM classAvail WHERE isallocated=?1", nativeQuery =
true)

```

```

List<ClassAvailModel> findByAllocated(boolean isallocated);

//We use TRUE here, cause we allocate them by the Secretariat User..
@Query(value = "SELECT recordnumber FROM classAvail WHERE recordnumber=(SELECT
MIN(recordnumber) FROM classavail WHERE isallocated=TRUE AND
periodavailability=?1)", nativeQuery = true)
Integer ClassroomSearch(Integer ExamPeriod);

//Here we UPDATE each ClassAvailModel that has a specific RecordNumber with
Allocated Flag False..
@Modifying
@Query(value = "UPDATE classAvail SET isallocated = 0 WHERE recordnumber =
?1", nativeQuery = true)
void UpdateRecordNo(Integer RankNo);

//Here we UPDATE each ClassAvailModel that has a specific RecordNumber with a
specific Capacity..
@Modifying
@Query(value = "UPDATE classAvail SET currentcapacity = ?2 WHERE recordnumber
= ?1", nativeQuery = true)
void UpdateCapacity(Integer RankNo, Integer CurrentCapacity);
}

```

Κώδικας από αρχείο «ClassModel.java» :

```

package net.uniwa.uniwaexamsproj.classes;

import net.uniwa.uniwaexamsproj.classavailability.ClassAvailModel;
import net.uniwa.uniwaexamsproj.examclass.ExamClassModel;
import net.uniwa.uniwaexamsproj.examstud.ExamStudentsModel;
import javax.persistence.*;
import java.util.List;
import java.util.Set;

@Entity
@Table (name="classroom")
public class ClassModel
{
    @Id
    @Column(name ="classID")
    private String ID;

    @Column(name ="classCapacity")
    private Integer Capacity;

    @Column(name ="classMode")
    private Integer BlankSeatMode;

    @Column(name ="classSupNo")
    private Integer MinSupNo;

    @OneToMany(mappedBy = "classavailobj",fetch = FetchType.EAGER)
    private List<ClassAvailModel> classavail;

    @OneToMany(mappedBy = "classtudobj",fetch = FetchType.EAGER)
    private Set<ExamStudentsModel> examstud ;

    @OneToMany(mappedBy = "classobj", fetch=FetchType.EAGER)

```

```

private Set<ExamClassModel> examclass ;

public void setClassavail(List<ClassAvailModel> classavail) {this.classavail
= classavail;}

public void setExamstud(Set<ExamStudentsModel> examstud) {this.examstud =
examstud;}

public void setExamclass(Set<ExamClassModel> examclass) {this.examclass =
examclass;}

public String getID()
{
return ID;
}

public Integer getCapacity()
{
return Capacity;
}

public void setCapacity(Integer Capacity) {this.Capacity = Capacity;}

public Integer getBlankSeatMode()
{
return BlankSeatMode;
}

public Integer getMinSupNo()
{
return MinSupNo;
}

public List<ClassAvailModel> getClassavail() {return classavail;}

public Set<ExamStudentsModel> getExamstud() {
return examstud;
}

public Set<ExamClassModel> getExamclass() {return examclass;}

public void reduceCapacity() {Capacity=Capacity-1;}

public void increaseCapacity() {Capacity=Capacity+1;}
}

```

Κώδικας από αρχείο «ClassRepository.java» :

```

package net.uniwa.uniwaexamsproj.classes;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface ClassRepository extends JpaRepository<ClassModel,String>

```

```

{
//Contains all the CRUD Operations (Create,Read,Update,Delete) for the Database..
}

```

Κώδικας από αρχείο «ExamModel.java» :

```

package net.uniwa.uniwaexamsproj.exam;

import net.uniwa.uniwaexamsproj.examclass.ExamClassModel;
import net.uniwa.uniwaexamsproj.examprof.ExamProfessorsModel;
import net.uniwa.uniwaexamsproj.examstud.ExamStudentsModel;
import javax.persistence.*;
import java.util.Date;
import java.util.Set;

@Entity
@Table(name = "exam")
public class ExamModel
{
    @Id
    @Column(name="examId")
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Integer ID;

    @Column(name="examYear")
    private String Year;

    @Column(name="examPeriod")
    private Integer Period;

    @Column(name="examType")
    private Integer Type;

    @Column(name="examStartDate")
    private Date StartDate;

    @Column(name="examEndDate")
    private Date EndDate;

    @OneToMany(mappedBy = "examstudobj", fetch = FetchType.EAGER)
    private Set<ExamStudentsModel> examstud ;

    @OneToMany(mappedBy = "examprofobj", fetch = FetchType.EAGER)
    private Set<ExamProfessorsModel> examprof;

    @OneToMany(mappedBy = "examclassobj", fetch = FetchType.EAGER)
    private Set<ExamClassModel> examclass;

    public void setID(Integer ID) {this.ID = ID;}

    public void setYear(String year) {Year = year;}

    public void setPeriod(Integer period) {Period = period;}

    public void setType(Integer type) {Type = type;}
}

```

```

public void setStartDate(Date startDate) {StartDate = startDate;}

public void setEndDate(Date endDate) {EndDate = endDate;}

public void setExamstud(Set<ExamStudentsModel> examstud) {this.examstud =
examstud;}

public void setExamprof(Set<ExamProfessorsModel> examprof) {this.examprof =
examprof;}

public void setExamclass(Set<ExamClassModel> examclass) {this.examclass =
examclass;}

public Integer getID()
{
return ID;
}

public String getYear()
{
return Year;
}

public Integer getPeriod()
{
return Period;
}

public Integer getType()
{
return Type;
}

public Date getStartDate()
{
return StartDate;
}

public Date getEndDate()
{
return EndDate;
}

public Set<ExamStudentsModel> getExamstud() {
return examstud;
}

public Set<ExamProfessorsModel> getExamprof() {
return examprof;
}

public Set<ExamClassModel> getExamclass() {return examclass;}
}

```

Κώδικας από αρχείο «ExamRepository.java» :

```
package net.uniwa.uniwaexamsproj.exam;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

import java.util.ArrayList;
import java.util.List;

@Repository
public interface ExamRepository extends JpaRepository<ExamModel,Integer>
{
    //Query in which we find the ExamID of an Exam using ExamYear, Period & Type..
    @Query(value = "SELECT examId FROM exam WHERE examYear=?1 AND examPeriod=?2
    AND examType=?3", nativeQuery = true)
    Integer findexam(String ExYear, Integer ExPeriod, Integer Credential);
}
```

Κώδικας από αρχείο «ExamClassKey.java» :

```
package net.uniwa.uniwaexamsproj.examclass;

import javax.persistence.Column;
import javax.persistence.Embeddable;
import java.io.Serializable;

@Embeddable
public class ExamClassKey implements Serializable
{
    @Column(name = "ClassId")
    private String classID;

    @Column(name = "ExamId")
    private Integer examID;

    @Column(name = "LesId")
    private String lessonID;

    public void setClassID(String classID) {this.classID = classID;}

    public void setExamID(Integer examID) {this.examID = examID;}

    public void setLessonID(String lessonID) {this.lessonID = lessonID;}

    public String getClassID() {return classID;}

    public Integer getExamID() {return examID;}

    public String getLessonID() {return lessonID;}
}
```

Κώδικας από αρχείο «ExamClassModel.java» :

```
package net.uniwa.uniwaexamsproj.examclass;

import net.uniwa.uniwaexamsproj.classes.ClassModel;
import net.uniwa.uniwaexamsproj.exam.ExamModel;
import net.uniwa.uniwaexamsproj.lesson.LessonModel;
import javax.persistence.*;
import java.util.Optional;

@Entity
@Table (name = "examClass")
public class ExamClassModel {

    @EmbeddedId
    private ExamClassKey examclasskey;

    @Column(name = "ExamStudsNo")
    private Integer studno;

    @Column(name = "RankNo")
    private Integer rank;

    @ManyToOne
    @MapsId("classID")
    @JoinColumn(name = "ClassId")
    private ClassModel classobj;

    @ManyToOne
    @MapsId("examID")
    @JoinColumn(name = "ExamId")
    private ExamModel examclassobj;

    @ManyToOne
    @MapsId("lessonID")
    @JoinColumn(name = "LesId")
    private LessonModel lesclassobj;

    public void setClassobj(Optional<ClassModel> classobj) {this.classobj =
classobj.get();}

    public void setExamclassobj(Optional<ExamModel> examclassobj)
{this.examclassobj = examclassobj.get();}

    public void setLesclassobj(Optional<LessonModel> lesclassobj)
{this.lesclassobj = lesclassobj.get();}

    public void setExamclasskey(ExamClassKey examclasskey) {this.examclasskey =
examclasskey;}

    public void setStudno(Integer studno) {this.studno = studno;}

    public ExamClassKey getExamclasskey() {
return examclasskey;
}

    public ClassModel getClassobj() {
return classobj;
}
```



```

public ExamModel getExamclassobj() {return examclassobj;}

public LessonModel getLesclassobj() {return lesclassobj;}

public Integer getStudno() {return studno;}

public void increaseStudNo() {studno=studno+1;}

public void reduceStudNo() {studno=studno-1;}

public Integer getRank() {return rank;}

public void setRank(Integer rank) {this.rank = rank;}
}

```

Κώδικας από αρχείο «ExamClassRepository.java» :

```

package net.uniwa.uniwaexamsproj.examclass;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Modifying;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

@Repository
public interface ExamClassRepository extends
JpaRepository<ExamClassModel, ExamClassKey>
{
//Here we use this Query to find if there is any Class Available using ExamClass
& Classroom Tables in INNER JOIN..
@Query(value = "SELECT examclass.ClassId FROM examclass INNER JOIN classroom
ON examclass.ClassId = classroom.classID WHERE (ExamId=?1 AND LesId=?2) AND
(ExamStudsNo < classroom.classCapacity)", nativeQuery = true)
String ClassAvailability(Integer ExamID, String LessonID);

//Here we find a specific ExamClassModel that contains a specific Class/Exam
& Lesson ID..
@Query(value = "SELECT * FROM examclass WHERE ClassId = ?1 AND ExamId = ?2 AND
LesId = ?3", nativeQuery = true)
ExamClassModel ClassSearchByKey(String ClassID, Integer ExamID, String
LessonID);

//Here we make an Insert into the ExamClass Table..
@Modifying
@Query(value = "INSERT INTO examclass VALUES (?1,?2,?3,0,?4)", nativeQuery =
true)
void InsertExamClass(String ClassID, Integer ExamID, String LessonID, Integer
ClassRank);

//Here we Update the ExamStudNo by -1 using the RankNo of the Registry..
@Modifying
@Query(value = "UPDATE examClass SET ExamStudsNo=(ExamStudsNo - 1) WHERE
RankNo=?1", nativeQuery = true)
void IncreaseCapacity(Integer RankNo);

//Here we Update the ExamStudNo by +1 using the RankNo of the Registry..
@Modifying
@Query(value = "UPDATE examClass SET ExamStudsNo=(ExamStudsNo + 1) WHERE
RankNo=?1", nativeQuery = true)

```

```

void ReduceCapacity(Integer RankNo);

//Here we get the RankNo of an ExamClass Record using the Class/Exam & Lesson
ID..
@Query(value = "SELECT RankNo FROM examclass WHERE ClassId=?1 AND ExamId=?2
AND LesId=?3", nativeQuery = true)
Integer GetRankNoByID(String ClassID, Integer ExamID, String LessonID);
}

```

Κώδικας από αρχείο «ExamProfessorsKey.java» :

```

package net.uniwa.uniwaexamsproj.examprof;

import javax.persistence.Column;
import javax.persistence.Embeddable;
import java.io.Serializable;

@Embeddable
public class ExamProfessorsKey implements Serializable
{
    @Column(name = "ProfId")
    private String profID;

    @Column(name = "ExamId")
    private Integer examID;

    @Column(name = "LesId")
    private String lessonID;

    public void setProfID(String profID) {this.profID = profID;}

    public void setExamID(Integer examID) {this.examID = examID;}

    public void setLessonID(String lessonID) {this.lessonID = lessonID;}

    public String getProfID() {
    return profID;
    }

    public Integer getExamID() {
    return examID;
    }

    public String getLessonID() {
    return lessonID;
    }
}

```

Κώδικας από αρχείο «ExamProfessorsModel.java» :

```
package net.uniwa.uniwaexamsproj.examprof;

import net.uniwa.uniwaexamsproj.exam.ExamModel;
import net.uniwa.uniwaexamsproj.lesson.LessonModel;
import net.uniwa.uniwaexamsproj.professor.ProfessorModel;
import javax.persistence.*;
import java.util.Optional;

@Entity
@Table (name = "examProf")
public class ExamProfessorsModel
{
    @EmbeddedId
    private ExamProfessorsKey examprofkey;

    @ManyToOne
    @MapsId("profID")
    @JoinColumn(name = "ProfId")
    private ProfessorModel profobj;

    @ManyToOne
    @MapsId("examID")
    @JoinColumn(name = "ExamId")
    private ExamModel examprofobj;

    @ManyToOne
    @MapsId("lessonID")
    @JoinColumn(name = "LesId")
    private LessonModel lesprofobj;

    public void setProfobj(Optional<ProfessorModel> profobj) {this.profobj =
    profobj.get();}

    public void setExamprofobj(Optional<ExamModel> examprofobj) {this.examprofobj
    = examprofobj.get();}

    public void setLesprofobj(Optional<LessonModel> lesprofobj) {this.lesprofobj
    = lesprofobj.get();}

    public void setExamprofkey(ExamProfessorsKey examprofkey) {this.examprofkey =
    examprofkey;}
    public ExamProfessorsKey getExamprofkey() {
    return examprofkey;
    }

    public ProfessorModel getProfobj() {
    return profobj;
    }

    public ExamModel getExamprofobj() {return examprofobj;}

    public LessonModel getLesprofobj() {return lesprofobj;}
}
```

Κώδικας από αρχείο «ExamProfessorsRepository.java» :

```
package net.uniwa.uniwaexamsproj.examprof;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public interface ExamProfessorsRepository extends
JpaRepository<ExamProfessorsModel, ExamProfessorsKey>
{
//Contains all the CRUD Operations (Create,Read,Update,Delete) for the
Database..
@Query(value = "SELECT ProfId FROM examProf WHERE ExamId=?1 AND LesId=?2",
nativeQuery = true)
String FindProfID(Integer ExamID, String LessonID);

//Here we get an ExamProfessor Entity using Professor & Exam ID..
@Query(value = "SELECT * FROM examProf WHERE ProfId=?1 AND ExamId=?2",
nativeQuery = true)
List<ExamProfessorsModel> getProfDetails(String ProfID, Integer ExamID);
}
```

Κώδικας από αρχείο «ExamStudentsKey.java» :

```
package net.uniwa.uniwaexamsproj.examstud;

import javax.persistence.Column;
import javax.persistence.Embeddable;
import java.io.Serializable;

@Embeddable
public class ExamStudentsKey implements Serializable
{
@Column(name = "StudId")
private String studentID;

@Column(name = "ExamId")
private Integer examID;

@Column(name = "LesId")
private String lessonID;

@Column(name = "ClassId")
private String classID;

public void setStudentID(String studentID) {this.studentID = studentID;}

public void setExamID(Integer examID) {this.examID = examID;}

public void setLessonID(String lessonID) {this.lessonID = lessonID;}

public void setClassID(String classID) {this.classID = classID;}

public String getStudentID() {
```

```

return studentID;
}

public Integer getExamID() {
return examID;
}

public String getLessonID() {
return lessonID;
}

public String getClassID() {
return classID;
}
}

```

Κώδικας από αρχείο «ExamStudentsModel.java» :

```

package net.uniwa.uniwaexamsproj.examstud;

import net.uniwa.uniwaexamsproj.lesson.LessonModel;
import net.uniwa.uniwaexamsproj.classes.ClassModel;
import net.uniwa.uniwaexamsproj.exam.ExamModel;
import net.uniwa.uniwaexamsproj.student.StudentModel;
import javax.persistence.*;

@Entity
@Table (name = "examStud")
public class ExamStudentsModel
{
    @EmbeddedId
    private ExamStudentsKey examstudkey;

    @ManyToOne
    @MapsId("studentID")
    @JoinColumn(name = "StudId")
    private StudentModel studobj;

    @ManyToOne
    @MapsId("examID")
    @JoinColumn(name = "ExamId")
    private ExamModel examstudobj;

    @ManyToOne
    @MapsId("lessonID")
    @JoinColumn(name = "LesId")
    private LessonModel lestudobj;

    @ManyToOne(cascade = CascadeType.PERSIST)
    @MapsId("classID")
    @JoinColumn(name = "ClassId")
    private ClassModel classtudobj;

    @Column(name = "StudSeat")
    private String StudentSeats;

    public void setExamstudkey(ExamStudentsKey examstudkey) {this.examstudkey =
examstudkey;}
}

```

```

public void setStudentSeats(String studentSeats) {StudentSeats =
studentSeats;}

public ExamStudentsKey getExamstudkey() {return examstudkey;}

public StudentModel getStudobj() {
return studobj;
}

public ExamModel getExamstudobj() {return examstudobj;}

public LessonModel getLestudobj() {return lestudobj;}

public ClassModel getClasstudobj() {return classtudobj;}

public String getStudentSeats() {return StudentSeats;}

public void setClasstudobj(ClassModel classtudobj) {this.classtudobj =
classtudobj;}

public void setStudobj(StudentModel studobj) {
this.studobj = studobj;
}

public void setExamstudobj(ExamModel examstudobj) {
this.examstudobj = examstudobj;
}

public void setLestudobj(LessonModel lestudobj) {
this.lestudobj = lestudobj;
}
}

```

Κώδικας από αρχείο «ExamStudentsRepository.java» :

```

package net.uniwa.uniwaexamsproj.examstud;

import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Modifying;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

@Repository
public interface ExamStudentsRepository extends
JpaRepository<ExamStudentsModel, ExamStudentsKey> {

//Here we find all the ExamStudent Records that have a specific Student ID..
@Query(value = "SELECT * FROM examStud WHERE StudId=?1", nativeQuery = true)
List<ExamStudentsModel> findByStudentId(String studentId);

//Here we Insert a new ExamStud Record into the Database..
@Modifying
@Query(value = "INSERT INTO examStud VALUES (?1,?2,?3,?4,?5)", nativeQuery =
true)
void SaveExamStud(String StudID, Integer ExamID, String LesID, String ClassID,
String seat);

//Here we get the Student Seat using Student/Exam/Lesson & Class ID's..

```

```

@Query(value = "SELECT StudSeat FROM examStud WHERE StudId=?1 AND ExamId=?2
AND LesId=?3 AND ClassId=?4", nativeQuery = true)
String GetSeatInUse(String StudID, Integer ExamID, String LesID, String
ClassID);

//Here we Delete an ExamStud Record using Student/Exam/Lesson & Class ID's..
@Modifying
@Query(value = "DELETE FROM examstud WHERE StudId = ?1 AND ExamId = ?2 AND
LesId = ?3 AND ClassId = ?4", nativeQuery = true)
void DeleteExamStud(String StudID, Integer ExamID, String LesID, String
ClassID);

//Here we get the Student ID of an ExamStudent Record using Lesson & Exam
ID's..
@Query(value = "SELECT StudId FROM examStud WHERE LesId=?1 AND ExamId=?2",
nativeQuery = true)
List<String> GetStudentIDs(String LesID, Integer ExamID);
}

```

Κώδικας από αρχείο «LessonModel.java» :

```

package net.uniwa.uniwaexamsproj.lesson;

import net.uniwa.uniwaexamsproj.examclass.ExamClassModel;
import net.uniwa.uniwaexamsproj.examprof.ExamProfessorsModel;
import net.uniwa.uniwaexamsproj.examstud.ExamStudentsModel;
import net.uniwa.uniwaexamsproj.lessonperstudent.LessonPerStudentModel;
import javax.persistence.*;
import org.hibernate.annotations.Fetch;
import org.hibernate.annotations.FetchMode;
import java.util.List;
import java.util.Set;

@Entity
@Table(name = "lesson")
public class LessonModel
{
    @Id
    @Column(name = "lesID")
    private String ID;

    @Column(name = "lesTitle")
    private String Title;

    @Column(name = "lesSemester")
    private Integer Semester;

    @Column(name = "lesPeriod")
    private Integer Period;

    @OneToMany(mappedBy = "lestudobj", fetch = FetchType.EAGER)
    @Fetch(value = FetchMode.SUBSELECT)
    private List<ExamStudentsModel> examstud ;

    @OneToMany(mappedBy = "lesprofobj", fetch = FetchType.EAGER)
    private Set<ExamProfessorsModel> examprof;

    @OneToMany(mappedBy = "lesclassobj", fetch = FetchType.EAGER)

```

```

@Fetch(value = FetchType.SUBSELECT)
private List<ExamClassModel> examclass;

@OneToMany(mappedBy = "LesPassobj", fetch = FetchType.EAGER)
private Set<LessonPerStudentModel> lessonpass;

public void setExamstud(List<ExamStudentsModel> examstud) {this.examstud =
examstud;}

public void setExamprof(Set<ExamProfessorsModel> examprof) {this.examprof =
examprof;}

public void setExamclass(List<ExamClassModel> examclass) {this.examclass =
examclass;}

public String getID() {return ID;}

public String getTitle() {
return Title;
}

public Integer getSemester() {
return Semester;
}

public Integer getPeriod() {
return Period;
}

public List<ExamStudentsModel> getExamstud() {
return examstud;
}

public Set<ExamProfessorsModel> getExamprof() {
return examprof;
}

public List<ExamClassModel> getExamclass() {
return examclass;
}
}

```

Κώδικας από αρχείο «LessonRepository.java» :

```

package net.uniwa.uniwaexamsproj.lesson;

import net.uniwa.uniwaexamsproj.examstud.ExamStudentsModel;
import net.uniwa.uniwaexamsproj.lessonperstudent.LessonPerStudentModel;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

import java.util.ArrayList;
import java.util.List;

@Repository
public interface LessonRepository extends JpaRepository<LessonModel,String>
{

```



```

//Here we find the Student Courses using Lesson & Student ID..
@Query(value = "SELECT lesson.* FROM lesson INNER JOIN lessonperstud ON
lesson.lesID=lessonperstud.lesID WHERE studID=?1", nativeQuery = true)
ArrayList<LessonModel> findStudentCourses(String studentId);

//Here we find all the Lesson Titles..
@Query(value = "SELECT lesTitle FROM lesson", nativeQuery = true)
ArrayList<String> findlesontitles();

//Here we find the Lesson ID of the Lesson with a specific Title..
@Query(value = "SELECT lesID FROM lesson WHERE lesTitle=?1", nativeQuery =
true)
String findlesid(String lessonTitle);

//Here we find the Lesson with a specific ID..
@Query(value = "SELECT * FROM lesson WHERE lesID=?1", nativeQuery = true)
LessonModel FindLessonByID(String LessonID);
}

```

Κώδικας από αρχείο «LessonPerStudentKey.java» :

```

package net.uniwa.uniwaexamsproj.lessonperstudent;

import javax.persistence.Column;
import javax.persistence.Embeddable;
import java.io.Serializable;

@Embeddable
public class LessonPerStudentKey implements Serializable
{
    @Column(name="StudId")
    private String StudentID;

    @Column(name="LesId")
    private String LessonID;

    public String getStudentID() {return StudentID;}

    public String getLessonID() {return LessonID;}

    public void setStudentID(String studentID) {
    StudentID = studentID;
    }

    public void setLessonID(String lessonID) {
    LessonID = lessonID;
    }
}

```

Κώδικας από αρχείο «LessonPerStudentModel.java» :

```
package net.uniwa.uniwaexamsproj.lessonperstudent;

import net.uniwa.uniwaexamsproj.lesson.LessonModel;
import net.uniwa.uniwaexamsproj.student.StudentModel;
import javax.persistence.*;

@Entity
@Table(name = "lessonperstud")
public class LessonPerStudentModel {

    @EmbeddedId
    private LessonPerStudentKey lessonpasskey;

    @ManyToOne
    @MapsId("StudentID")
    @JoinColumn(name = "StudId")
    private StudentModel StudPassobj;

    @ManyToOne
    @MapsId("LessonID")
    @JoinColumn(name = "LesId")
    private LessonModel LesPassobj;

    @Column(name="GotaTicket")
    private Integer GotaTicket;

    @Column(name="RexamedLesson")
    private Integer RexamedLesson;

    public StudentModel getStudPassobj() {return StudPassobj;}

    public LessonModel getLesPassobj() {return LesPassobj;}

    public Integer getGotaTicket() {return GotaTicket;}

    public void setGotaTicket(Integer gotaTicket) {GotaTicket = gotaTicket;}

    public LessonPerStudentKey getLessonpasskey() {return lessonpasskey;}

    public void setLessonpasskey(LessonPerStudentKey lessonpasskey)
    {this.lessonpasskey = lessonpasskey;}

    public Integer getRexamedLesson() {return RexamedLesson;}

    public void setRexamedLesson(Integer rexamedLesson) {RexamedLesson =
    rexamedLesson;}
}
```

Κώδικας από αρχείο «LessonPerStudentRepository.java» :

```
package net.uniwa.uniwaexamsproj.lessonperstudent;

import net.uniwa.uniwaexamsproj.examstud.ExamStudentsModel;
import net.uniwa.uniwaexamsproj.lesson.LessonModel;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Modifying;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public interface LessonPerStudentRepository extends
JpaRepository<LessonPerStudentModel, LessonPerStudentKey>
{
    //Here we find if a Lesson has been Rexamed before or not..
    @Query(value = "SELECT COUNT(*) FROM lessonperstud WHERE StudId=?1 AND
LesId=?2 AND Rexamedlesson=1;", nativeQuery = true)
    Integer LessonRexamCheck(String StudentID, String LessonID);

    //Here we count the Rexamed Lessons..
    @Query(value = "SELECT COUNT(*) FROM lessonperstud WHERE StudId=?1 AND
Rexamedlesson=1;", nativeQuery = true)
    Integer LessonRexamCount(String StudentID);

    //Here we find if a Lesson has been signed for examination or not..
    @Query(value = "SELECT COUNT(*) FROM lessonperstud WHERE StudId=?1 AND
LesId=?2 AND GotaTicket=1;", nativeQuery = true)
    Integer LessonTicketCheck(String StudentID, String LessonID);

    //Here we Assign a Lesson for Rexamination..
    @Modifying
    @Query(value = "UPDATE lessonperstud SET Rexamedlesson=1 WHERE StudId=?1 AND
LesId=?2", nativeQuery = true)
    void LessonRexamAssignment(String StudentID, String LessonID);

    //Here we Register a Lesson for Examination..
    @Modifying
    @Query(value = "UPDATE lessonperstud SET GotaTicket=1 WHERE StudId=?1 AND
LesId=?2", nativeQuery = true)
    void LessonRegistration(String StudentID, String LessonID);

    //Here we Unsign a Lesson for Examination..
    @Modifying
    @Query(value = "UPDATE lessonperstud SET GotaTicket=0 WHERE StudId=?1 AND
LesId=?2", nativeQuery = true)
    void LessonRelease(String StudentID, String LessonID);
}
```

Κώδικας από αρχείο «TimeslotModel.java» :

```
package net.uniwa.uniwaexamsproj.timeslot;

import net.uniwa.uniwaexamsproj.classavailability.ClassAvailModel;

import javax.persistence.*;

import java.util.Set;
import java.sql.Time;
import java.util.Date;

@Entity
@Table (name="timeslot")
public class TimeslotModel
{
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column (name = "DslotID")
    private Integer ID;

    @Column (name = "Dslot")
    private Date Date;

    @Column (name = "TslotStart")
    private Time Start;

    @Column (name = "TslotEnd")
    private Time End;

    @OneToMany(mappedBy = "tslotavailobj")
    private Set<ClassAvailModel> classavail;

    public Integer getID()
    {
        return ID;
    }

    public Date getDate()
    {
        return Date;
    }

    public Time getStart()
    {
        return Start;
    }

    public Time getEnd()
    {
        return End;
    }

    public Set<ClassAvailModel> getClassavail() {return classavail;}
}
```

Κώδικας από αρχείο «TimeslotRepository.java» :

```
package net.uniwa.uniwaexamsproj.timeslot;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface TimeslotRepository extends
JpaRepository<TimeslotModel,Integer>
{
//Contains all the CRUD Operations (Create,Read,Update,Delete) for the
Database..
}
```

Κώδικας από αρχείο «StudentMainView.java» :

```
package net.uniwa.uniwaexamsproj.student;

import org.springframework.beans.factory.annotation.Autowired;
import com.vaadin.flow.component.Tag;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.dependency.JsModule;
import com.vaadin.flow.component.html.H3;
import com.vaadin.flow.component.littemplate.LitTemplate;
import com.vaadin.flow.component.polymertemplate.Id;
import com.vaadin.flow.router.BeforeEnterEvent;
import com.vaadin.flow.router.BeforeEnterObserver;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.VaadinSession;
import net.uniwa.uniwaexamsproj.login.LoginService;
import net.uniwa.uniwaexamsproj.login.LoginView;

//A Designer generated component for the student-view template..
//Designer will add and remove fields with @Id mappings but does not overwrite
or otherwise change this file..

//Some basic annotations to map the route of the class, to give a tag and to
associate it with the typescript file in the frontend package..
@Route("student")
@Tag("student-main-view")
@JsModule("./src/views/student/student-main-view.ts")
public class StudentMainView extends LitTemplate implements
BeforeEnterObserver { //Inherit the components/methods of LitTemplate for the
View & Implement the BeforeEnterObserver Interface for the BeforeEnterMethod..

//Declaration of the Login & Student Service and the Buttons of the Template
in the View of our Frontend Page..
@Autowired
LoginService loginService;

@Autowired
StudentService stdService;

@Id("apply")
Button applyBtn;
```

```

@Id("display")
Button displayBtn;

@Id("exit")
Button exitBtn;

@Id("studentId")
H3 studentId;

//Creates a new StudentView..

public StudentMainView() {

//Get User from Vaadin Session and print his ID to the view..
StudentModel std = (StudentModel)
VaadinSession.getCurrent().getAttribute("user");

//As long we got the ID, we print it to the view..
if(std!= null) {
studentId.setText("ID MAØHTH: " + std.getID());
}

//Navigate to apply section by using the Apply Button..
applyBtn.addClickListener(event ->{
UI.getCurrent().navigate("student/apply");
});

//Navigate to application section by using the Display Button..
displayBtn.addClickListener(event ->{
UI.getCurrent().navigate("student/applications");
});

//Exit from Session and Return to Home Page by clicking Exit Button..
exitBtn.addClickListener(event ->{
VaadinSession.getCurrent().setAttribute("user", null);
UI.getCurrent().navigate("/");
});
}

//Verify if user is authenticated and has role Student..
@Override
public void beforeEnter(BeforeEnterEvent event) {
if (!LoginView.class.equals(event.getNavigationTarget())&&
! loginService.isAllowed(VaadinSession.getCurrent().getAttribute("user"),
"Student")) {
event.rerouteTo(LoginView.class);
}
}
}

```

Κώδικας από αρχείο «ApplyView.java» :

```
package net.uniwa.uniwaexamsproj.student;

import net.uniwa.uniwaexamsproj.exam.ExamRepository;
import net.uniwa.uniwaexamsproj.lessonperstudent.LessonPerStudentRepository;
import org.springframework.beans.factory.annotation.Autowired;

import com.vaadin.flow.component.Html;
import com.vaadin.flow.component.Tag;
import com.vaadin.flow.component.Text;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.button.ButtonVariant;
import com.vaadin.flow.component.dependency.JsModule;
import com.vaadin.flow.component.grid.Grid;
import com.vaadin.flow.component.html.Div;
import com.vaadin.flow.component.icon.Icon;
import com.vaadin.flow.component.littemplate.LitTemplate;
import com.vaadin.flow.component.notification.Notification;
import com.vaadin.flow.component.notification.NotificationVariant;
import com.vaadin.flow.component.orderedlayout.FlexComponent.Alignment;
import com.vaadin.flow.component.orderedlayout.HorizontalLayout;
import com.vaadin.flow.component.orderedlayout.VerticalLayout;
import com.vaadin.flow.component.polymertemplate.Id;
import com.vaadin.flow.component.select.Select;
import com.vaadin.flow.router.BeforeEnterEvent;
import com.vaadin.flow.router.BeforeEnterObserver;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.VaadinSession;
import net.uniwa.uniwaexamsproj.lesson.LessonModel;
import net.uniwa.uniwaexamsproj.lesson.LessonRepository;
import net.uniwa.uniwaexamsproj.login.LoginService;
import net.uniwa.uniwaexamsproj.login.LoginView;

import java.time.Year;
import java.util.Iterator;
import java.util.List;
import java.util.Optional;

//A Designer generated component for the apply-view template..
//Designer will add and remove fields with @Id mappings but does not overwrite
or otherwise change this file..

//Some basic annotations to map the route of the class, to give a tag and to
associate it with the typescript file in the frontend package..
@Route("/student/apply")
@Tag("apply-view")
@JsModule("./src/views/student/apply-view.ts")
public class ApplyView extends LitTemplate implements BeforeEnterObserver {
//Inherit the components/methods of LitTemplate for the View & Implement the
BeforeEnterObserver Interface for the BeforeEnterMethod..

//Declaration of the Login & Student Service and the Buttons of the Template
in the View of our Frontend Page..
@Autowired
LoginService loginService;

@Autowired
StudentService stdService;
```

```

@Id("lessons")
Grid<LessonModel> lessonsGrid;

@Id("layout1")
VerticalLayout layout1;

StudentModel student;

String selectedCredential;

@Id("return")
Button returnBtn;

@Id("exit")
Button exitBtn;

//Create another Student View..
public ApplyView(ExamRepository examrep, LessonRepository lessrep) {

//Define LessonGrid columns and fill with values..
lessonsGrid.addColumn(LessonModel::getID).setHeader("ID");
lessonsGrid.addColumn(LessonModel::getTitle).setHeader("ΤΙΤΛΟΣ");

//Define column select credential..
lessonsGrid.addComponentColumn(c -> {

Select<String> credentialsSlct = new Select<>();

credentialsSlct.addValueChangeListener(event -> {
selectedCredential = event.getValue();
});

credentialsSlct.getElement().setAttribute("label", "ΔΙΑΠΙΣΤΕΥΤΗΡΙΑ");

credentialsSlct.setItems("ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗΝΟΥ", "ΕΜΒΟΛΙΜΗ
ΕΞΕΤΑΣΤΙΚΗ", "ΕΞΕΤΑΣΤΙΚΗ ΓΙΑ ΔΙΟΡΘΩΣΗ ΒΑΘΜΟΥ", "ΕΞΕΤΑΣΤΙΚΗ ΣΕΠΤΕΜΒΡΙΟΥ");

return credentialsSlct;
}).setWidth("150px").setFlexGrow(0);

//Define save column..
lessonsGrid.addComponentColumn(c -> {
Button button = new Button("ΕΠΙΛΟΓΗ");

button.addClickListener(e -> {
int d = 0;

//Map selected credential with its value..
switch (selectedCredential) {
case "ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗΝΟΥ":
d = 0;
break;
case "ΕΜΒΟΛΙΜΗ ΕΞΕΤΑΣΤΙΚΗ":
d = 1;
break;
case "ΕΞΕΤΑΣΤΙΚΗ ΓΙΑ ΔΙΟΡΘΩΣΗ ΒΑΘΜΟΥ":
d = 2;
break;
case "ΕΞΕΤΑΣΤΙΚΗ ΣΕΠΤΕΜΒΡΙΟΥ":
d = 3;

```



```

break;
}
student = (StudentModel) VaadinSession.getCurrent().getAttribute("user");

stdService.InitiateSession(student.getID());

//Insert application and get a success message..
try {
Integer ExamID=0;
String CurrentYear = String.valueOf(Year.now().getValue());

if(d == 3) //In case we've picked September, we search for September Period
and Credential '0'..
{
ExamID = examrep.findexam(CurrentYear,2,0);
}
else
{
ExamID = examrep.findexam(CurrentYear,c.getPeriod(),d);
}

if(ExamID != null){

String msg = stdService.insertrecord(student.getID(), ExamID, c.getID(), d);

Notification notification = new Notification();

//If student registered show success else show error..
Div text = null;

if(msg.equals("LESSON CHECKED"))
{
notification.addThemeVariants(NotificationVariant.LUMO_ERROR);
text = new Div(new Text("ΤΟ ΜΑΘΗΜΑ ΕΙΝΑΙ ΕΙΔΗ ΔΗΛΩΜΕΝΟ!"));
}
else if(msg.equals("PROFESSOR NOT FOUND"))
{
notification.addThemeVariants(NotificationVariant.LUMO_ERROR);
text = new Div(new Text("ΠΡΟΣΠΑΘΕΙΤΕ ΝΑ ΔΗΛΩΣΕΤΕ ΜΑΘΗΜΑ ΓΙΑ ΤΟ ΟΠΟΙΟ ΔΕΝ
ΥΠΑΡΧΕΙ ΚΑΘΗΓΗΤΗΣ ΠΟΥ ΝΑ ΣΧΕΤΙΖΕΤΑΙ ΜΕ ΤΗΝ ΕΞΕΤΑΣΗ ΣΤΟ ΣΥΣΤΗΜΑ!"));
}
else if(msg.equals("NOT THE SAME SEMESTER"))
{
notification.addThemeVariants(NotificationVariant.LUMO_ERROR);
text = new Div(new Text("ΠΡΟΣΠΑΘΕΙΤΕ ΝΑ ΔΗΛΩΣΕΤΕ ΜΑΘΗΜΑ ΑΠΟ ΔΙΑΦΟΡΕΤΙΚΟ
ΕΞΑΜΗΝΟ!"));
}
else if(msg.equals("NO CLASS AVAILABLE"))
{
notification.addThemeVariants(NotificationVariant.LUMO_ERROR);
text = new Div(new Text("ΟΛΕΣ ΟΙ ΑΙΘΟΥΣΕΣ ΕΙΝΑΙ ΔΕΣΜΕΥΜΕΝΕΣ ΓΙΑ ΤΗΝ ΠΑΡΟΥΣΑ
ΕΞΕΤΑΣΤΙΚΗ ΠΕΡΙΟΔΟ!"));
}
else if(msg.equals("UNDER 9TH SEMESTER"))
{
notification.addThemeVariants(NotificationVariant.LUMO_ERROR);
text = new Div(new Text("ΤΟ ΕΞΑΜΗΝΟ ΕΙΝΑΙ ΚΑΤΩ ΑΠΟ ΤΟ 9ο, ΣΥΝΕΠΩΣ ΔΕΝ ΕΙΝΑΙ
ΔΥΝΑΤΗ Η ΣΥΜΜΕΤΟΧΗ ΣΕ ΕΜΒΟΛΙΜΗ ΕΞΕΤΑΣΤΙΚΗ!"));
}
}
}

```

```

}
else if(msg.equals("LESSON REXAMED"))
{
notification.addThemeVariants(NotificationVariant.LUMO_ERROR);
text = new Div(new Text("ΤΟ ΜΑΘΗΜΑ ΕΧΕΙ ΕΠΑΝΕΞΕΤΑΣΘΕΙ ΓΙΑ ΒΕΛΤΙΩΣΗ ΒΑΘΜΟΥ ΕΙΔΗ
1 ΦΟΡΑ!"));
}
else if(msg.equals("ALREADY 3 LESSONS REXAMED"))
{
notification.addThemeVariants(NotificationVariant.LUMO_ERROR);
text = new Div(new Text("Ο ΦΟΙΤΗΤΗΣ ΕΧΕΙ ΕΙΔΗ ΕΞΕΤΑΣΘΕΙ ΣΕ 3 ΜΑΘΗΜΑΤΑ ΓΙΑ
ΒΕΛΤΙΩΣΗ ΒΑΘΜΟΥ!"));
}
else
{
notification.addThemeVariants(NotificationVariant.LUMO_SUCCESS);
text = new Div(new Html("<div>Η ΕΓΓΡΑΦΗ ΣΑΣ ΠΡΑΓΜΑΤΟΠΟΙΗΘΗΚΕ! </br>" + msg +
"<div>"));
}

Button closeButton = new Button(new Icon("lumo", "cross"));
closeButton.addThemeVariants(ButtonVariant.LUMO_TERTIARY_INLINE);
closeButton.getElement().setAttribute("aria-label", "Close");

//Close notification when clicking close button..
closeButton.addClickListener(event -> {
notification.close();
});

//Center the notification and display it..
HorizontalLayout layout = new HorizontalLayout(text, closeButton);
layout.setAlignItems(Alignment.CENTER);

notification.add(layout);
notification.open();
}
else
{
throw new IndexOutOfBoundsException();
}
}

catch(IndexOutOfBoundsException exceptionval) //For the case we try to
register a lesson for examination to which there is no examination registry
in the system before..
{
//If student registered show success else show error..
Div text = null;
Notification notification = new Notification();
notification.addThemeVariants(NotificationVariant.LUMO_ERROR);
text = new Div(new Text("ΔΕΝ ΥΠΑΡΧΕΙ ΕΞΕΤΑΣΤΙΚΗ ΣΤΟ ΣΥΣΤΗΜΑ ΓΙΑ ΝΑ ΔΗΛΩΘΕΙ ΤΟ
ΣΥΓΚΕΚΡΙΜΕΝΟ ΜΑΘΗΜΑ!"));

Button closeButton = new Button(new Icon("lumo", "cross"));
closeButton.addThemeVariants(ButtonVariant.LUMO_TERTIARY_INLINE);
closeButton.getElement().setAttribute("aria-label", "Close");

//Close notification when clicking close button..
closeButton.addClickListener(event -> {
notification.close();
}

```

```

});

//Center the notification and display it..
HorizontalLayout layout = new HorizontalLayout(text, closeButton);
layout.setAlignItems(Alignment.CENTER);

notification.add(layout);
notification.open();
}
});

return button;
}).setWidth("150px").setFlexGrow(0);

StudentModel std = (StudentModel)
VaadinSession.getCurrent().getAttribute("user");
lessonsGrid.setItems(lessrep.findStudentCourses(std.getID()));

//Navigate to previous page by clicking Return Button..
returnBtn.addClickListener(event -> {
UI.getCurrent().navigate("/student/");
});

//Remove user from session and return to home page by Clicking Exit Button..
exitBtn.addClickListener(event -> {
VaadinSession.getCurrent().setAttribute("user", null);
UI.getCurrent().navigate("/");
});
}

//Check if user is authenticated and has role Student..
@Override
public void beforeEnter(BeforeEnterEvent event) {
if (!LoginView.class.equals(event.getNavigationTarget())
&& !loginService.isAllowed(VaadinSession.getCurrent().getAttribute("user"),
"Student")) {
event.rerouteTo(LoginView.class);
}
}
}
}

```

Κώδικας από αρχείο «StudentApplicationsView.java» :

```

package net.uniwa.uniwaexamsproj.student;

import java.text.SimpleDateFormat;

import net.uniwa.uniwaexamsproj.examclass.ExamClassRepository;
import net.uniwa.uniwaexamsproj.timeslot.TimeslotRepository;
import org.springframework.beans.factory.annotation.Autowired;
import com.vaadin.flow.component.Tag;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.dependency.JsModule;
import com.vaadin.flow.component.grid.Grid;
import com.vaadin.flow.component.littemplate.LitTemplate;
import com.vaadin.flow.component.notification.Notification;
import com.vaadin.flow.component.polymertemplate.Id;

```

```

import com.vaadin.flow.router.BeforeEnterEvent;
import com.vaadin.flow.router.BeforeEnterObserver;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.VaadinSession;
import net.uniwa.uniwaexamsproj.classavailability.ClassAvailRepository;
import net.uniwa.uniwaexamsproj.examstud.ExamStudentsModel;
import net.uniwa.uniwaexamsproj.examstud.ExamStudentsRepository;
import net.uniwa.uniwaexamsproj.login.LoginService;
import net.uniwa.uniwaexamsproj.login.LoginView;

//A Designer generated component for the applications-view template..
//Designer will add and remove fields with @Id mappings but does not overwrite
or otherwise change this file..

//Some basic annotations to map the route of the class, to give a tag and to
associate it with the typescript file in the frontend package..
@Route("student/applications")
@Tag("student-applications-view")
@JsModule("./src/views/student/student-applications-view.ts")
public class StudentApplicationsView extends LitTemplate implements
BeforeEnterObserver { //Inherit the components/methods of LitTemplate for the
View & Implement the BeforeEnterObserver Interface for the BeforeEnterMethod..

//Declaration of the Login & Student Service, Student Model and the Buttons
of the Template in the View of our Frontend Page..
@Autowired
LoginService loginService;

@Autowired
StudentService stdService;

StudentModel student;

@Id("applications")
Grid<ExamStudentsModel> applicationsGrid;

@Id("return")
Button returnBtn;

@Id("exit")
Button exitBtn;

//Declare and Use European Date Format..
SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");

//Create a new Student View..
public StudentApplicationsView(ExamStudentsRepository exStRepo,
ClassAvailRepository classAvailRepo, ExamClassRepository examclassrep,
TimeslotRepository tslotrep) {

//Get Current user..
student = (StudentModel) VaadinSession.getCurrent().getAttribute("user");

//Define-Fill Columns with values..
applicationsGrid.addColumn(e -> e.getExamstudkey().getExamID()).setHeader("ID
ΕΞΕΤΑΣΤΙΚΗΣ");
applicationsGrid.addColumn(e -> e.getLestudobj().getID()).setHeader("ID
ΜΑΘΗΜΑΤΟΣ");

```

```

applicationsGrid.addColumn(e                                     ->
e.getExamstudkey().getClassID()).setHeader("ID ΑΙΘΟΥΣΑΣ");

//Get Timeslot Date from Exam Student Model..
applicationsGrid.addColumn(e -> {
if (!e.getClasstudobj().getClassavail().isEmpty())
return          stdService.getClassAvail(e.getClasstudobj().getClassavail(),
e.getExamstudkey().getClassID(),
e.getExamstudkey().getExamID(),e.getExamstudkey().getLessonID()).getTslotava
ilobj().getDate();

return "";

}).setHeader("ΗΜΕΡΟΜΗΝΙΑ");

//Get Timeslot Starting Date..
applicationsGrid.addColumn(e -> {
if (!e.getClasstudobj().getClassavail().isEmpty())
return          stdService.getClassAvail(e.getClasstudobj().getClassavail(),
e.getExamstudkey().getClassID(),e.getExamstudkey().getExamID(),e.getExamstud
key().getLessonID()).getTslotavailobj().getStart();

return "";

}).setHeader("ΩΡΑ ΞΑΡΞΗΣ");

//Get Timeslot Ending Date..
applicationsGrid.addColumn(e -> {
if (!e.getClasstudobj().getClassavail().isEmpty())
return          stdService.getClassAvail(e.getClasstudobj().getClassavail(),
e.getExamstudkey().getClassID(),e.getExamstudkey().getExamID(),e.getExamstud
key().getLessonID()).getTslotavailobj().getEnd();

return "";

}).setHeader("ΩΡΑ ΛΗΞΗΣ");

//Get Class ID, as long the Class is Available..
applicationsGrid.addColumn(e -> {
if (!e.getClasstudobj().getClassavail().isEmpty())
return e.getClasstudobj().getID();

return "";

}).setHeader("ΑΙΘΟΥΣΑ");

//Get the Seats..
applicationsGrid.addColumn(ExamStudentsModel::getStudentSeats).setHeader("ΘΕ
ΣΗ");

//Declare the Delete Button and it's work..
applicationsGrid.addComponentColumn(c -> {
Button button = new Button("ΑΚΥΡΩΣΗ");

button.addClickListener(e -> {
stdService.InitiateSession(student.getID());

//Delete the current application..
stdService.deleterecord(student.getID(),c.getExamstudkey().getExamID(),
c.getLestudobj().getID(), c.getClasstudobj().getID());

```

```

//Fill with values and refresh the Application Grid..
applicationsGrid.setItems(exStRepo.findByStudentId(student.getID()));

applicationsGrid.getDataProvider().refreshAll();

Notification.show("Η ΕΞΕΤΑΣΗ ΑΚΥΡΩΘΗΚΕ!"); //Notify the user everything went
as planned..

});

return button;
}).setWidth("150px").setFlexGrow(0);

//Fill Application Grid with values via Repository..
applicationsGrid.setItems(exStRepo.findByStudentId(student != null ?
student.getID() : ""));

//Navigate to previous page by clicking Return Button..
returnBtn.addClickListener(event -> {
UI.getCurrent().navigate("/student/");
});

//Exit from Session and Return to Home Page..
exitBtn.addClickListener(event -> {
VaadinSession.getCurrent().setAttribute("user", null);
UI.getCurrent().navigate("/");
});
}

//Check if user is authenticated and has role Student..
@Override
public void beforeEnter(BeforeEnterEvent event) {
if (!LoginView.class.equals(event.getNavigationTarget())
&& !loginService.isAllowed(VaadinSession.getCurrent().getAttribute("user"),
"Student")) {
event.rerouteTo(LoginView.class);
}
}
}
}

```

Κώδικας από αρχείο «SecretariatMainView.java» :

```

package net.uniwa.uniwaexamsproj.secretariat;

import org.springframework.beans.factory.annotation.Autowired;
import com.vaadin.flow.component.Tag;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.dependency.JsModule;
import com.vaadin.flow.component.littemplate.LitTemplate;
import com.vaadin.flow.component.polymertemplate.Id;
import com.vaadin.flow.router.BeforeEnterEvent;
import com.vaadin.flow.router.BeforeEnterObserver;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.VaadinSession;
import net.uniwa.uniwaexamsproj.login.LoginService;
import net.uniwa.uniwaexamsproj.login.LoginView;

```

```

//A Designer generated component for the secreteriat-view template.
//The Designer will add and remove fields with @Id mappings but does not
overwrite or otherwise change this file.

//Some basic annotations to map the route of the class, to give a tag and to
associate it with the typescript file in the frontend package..
@Route("secretariat")
@Tag("secretariat-main-view")
@JsModule("./src/views/secretariat/secretariat-main-view.ts")
public class SecretariatMainView extends LitTemplate implements
BeforeEnterObserver{ //Inherit the components/methods of LitTemplate for the
View & Implement the BeforeEnterObserver Interface for the BeforeEnterMethod..

//Declaration of the LoginService and the Buttons of the Template in the View
of our Frontend Page..
@Autowired
LoginService loginService;

@Id("insertExam")
Button insertExam;

@Id("updateExam")
Button updateExam;

@Id("allocateRoom")
Button allocateRoom;

@Id("examInfo")
Button examInfo;

@Id("exit")
Button exit;

//Creates a new SecreteriatView..
public SecretariatMainView() {

//Navigate to Insert Exam Menu by Clicking Insert Button..
insertExam.addClickListener(event ->{
UI.getCurrent().navigate("secretariat/exam-create");
});

//Navigate to Update Exam Menu by Clicking Update Button..
updateExam.addClickListener(event ->{
UI.getCurrent().navigate("/secretariat/update-exam");
});

//Navigate to Allocate Room Menu by Clicking Allocate Room Button..
allocateRoom.addClickListener(event ->{
UI.getCurrent().navigate("/secretariat/allocate-room");
});

//Navigate to Exam Info Menu by Clicking Exam Info Button..
examInfo.addClickListener(event ->{
UI.getCurrent().navigate("secretariat/exam-info");
});

//Navigate to Home Menu & Free Session by Clicking Exit Button..
exit.addClickListener(event ->{
VaadinSession.getCurrent().setAttribute("user", null);
UI.getCurrent().navigate("/");
});
}

```

```

}

//Check if the user is Authenticated and the role is Secretariat..
@Override
public void beforeEnter(BeforeEnterEvent event) {
if (!LoginView.class.equals(event.getNavigationTarget())&&
!loginService.isAllowed(VaadinSession.getCurrent().getAttribute("user"),
"Secretariat")) {
event.rerouteTo(LoginView.class);
}
}
}
}
}

```

Κώδικας από αρχείο «CreateExam.java» :

```

package net.uniwa.uniwaexamsproj.secretariat;

import com.vaadin.flow.component.Tag;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.combobox.ComboBox;
import com.vaadin.flow.component.dependency.JsModule;
import com.vaadin.flow.component.littemplate.LitTemplate;
import com.vaadin.flow.component.notification.Notification;
import com.vaadin.flow.component.polymertemplate.Id;
import com.vaadin.flow.component.textfield.TextField;
import com.vaadin.flow.router.BeforeEnterEvent;
import com.vaadin.flow.router.BeforeEnterObserver;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.VaadinSession;
import net.uniwa.uniwaexamsproj.exam.ExamModel;
import net.uniwa.uniwaexamsproj.exam.ExamRepository;
import net.uniwa.uniwaexamsproj.login.LoginService;
import net.uniwa.uniwaexamsproj.login.LoginView;
import org.aspectj.weaver.ast.Not;
import org.hibernate.engine.jdbc.spi.SqlExceptionHelper;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.dao.DataIntegrityViolationException;

import java.sql.Date;
import java.sql.SQLException;
import java.time.Month;
import java.time.Year;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;
import java.util.ListIterator;

@Route("secretariat/exam-create")
@Tag("create-exam")
@JsModule("./src/views/secretariat/create-exam.ts")
public class CreateExam extends LitTemplate implements BeforeEnterObserver{

    @Autowired
    LoginService loginService;

    @Id("year")
    TextField Year;

```



```

@Id("period")
ComboBox Period;

@Id("type")
ComboBox Type;

@Id("start")
TextField StartDate;

@Id("end")
TextField EndDate;

@Id("submit")
Button submitBtn;

@Id("clear")
Button clearBtn;

@Id("examproflink")
Button examproflinkBtn;

@Id("exit")
Button exitBtn;

@Id("return")
Button returnBtn;

public CreateExam(ExamRepository examrep)
{
if(!(examrep.findAll().isEmpty())) //Here we check if the Exam Table has any
record or not in order to make accesible or not the CreateExamProf Form View..
{
examproflinkBtn.setVisible(true);
}
else
{
examproflinkBtn.setVisible(false);
}

//Just initiate the Textfield/Combo Boxes of the Create Exam Form..
Year.setValue("ΕΤΟΣ");
Period.clear();
Type.clear();
StartDate.setValue("ΕΤΟΣ-ΜΗΝΑΣ-ΜΕΡΑ");
EndDate.setValue("ΕΤΟΣ-ΜΗΝΑΣ-ΜΕΡΑ");
ArrayList<String> periods = new ArrayList();
periods.add("ΧΕΙΜΕΡΙΝΟ");
periods.add("ΕΑΡΙΝΟ");
periods.add("ΣΕΠΤΕΜΒΡΙΟΣ");
Period.setItems(periods);
ArrayList<String> types = new ArrayList();
types.add("ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗΝΟΥ");
types.add("ΕΜΒΟΛΙΜΗ ΕΞΕΤΑΣΤΙΚΗ");
types.add("ΕΞΕΤΑΣΤΙΚΗ ΔΙΟΡΘΩΣΗΣ ΒΑΘΜΟΥ");
Type.setItems(types);

//Below we got the Submit Button actions..
submitBtn.setOnClickListener(event -> {
Boolean FalseYearFlag=false;
Boolean FalsePeriodFlag=false;
Boolean FalseTypeFlag=false;

```

```

Boolean FalseDateFlag=false;
String ExamPeriod="";
String ExamType="";

//Below we get each value from the Form..
String ExamYear = Year.getValue();
try
{
ExamPeriod = Period.getValue().toString();
}

catch(NullPointerException NullEx)
{
ExamPeriod="";
}

try
{
ExamType = Type.getValue().toString();
}

catch(NullPointerException NullEx)
{
ExamType="";
}

String ExamStart = StartDate.getValue();
String ExamEnd = EndDate.getValue();

ExamModel tmpexamrec = new ExamModel();

//Split the two Dates given and get the month and the year..
String[] StartDateParts = ExamStart.split("-");
String[] EndDateParts = ExamEnd.split("-");
String StartYear = StartDateParts[0];
String EndYear = EndDateParts[0];
String CurrentYear = java.time.Year.now().toString();
String StartMonth = StartDateParts[1];
String EndMonth = EndDateParts[1];
String MonthPeriod; //Flag to check whether our Dates and Exam Periods are
valid..

//Here we check if the Month of the Exam Start and End Dates are valid,
according to the Period..
if((StartMonth.equals("2") && EndMonth.equals("3"))||(StartMonth.equals("02")
&& EndMonth.equals("03")))
{
MonthPeriod = "XEIMEPINO";
}
else if((StartMonth.equals("6") && EndMonth.equals("7"))||(StartMonth.equals("06") && EndMonth.equals("07")))
{
MonthPeriod = "EAPINO";
}
else if((StartMonth.equals("9") || StartMonth.equals("09")) &&
EndMonth.equals("10"))
{
MonthPeriod = "ΣΕΠΤΕΜΒΡΙΟΣ";
}
else //Not valid Month Case..

```

```

{
MonthPeriod = "AKYPO";
}

//Here we check Year validation..
if(!ExamYear.equals("ΕΤΟΣ")      &&      !ExamYear.equals(""))      &&
(Integer.parseInt(CurrentYear) == Integer.parseInt(ExamYear))
&& ExamYear.equals(StartYear) && ExamYear.equals(EndYear))
{
tmpexamrec.setYear(ExamYear);
}
else //Not valid Year Case..
{
FalseYearFlag=true;
Notification.show("Δεν έχει εισαχθεί σωστά το Έτος Εξεταστικής!");
}

//Here we check Period validation..
if(ExamPeriod.equals("ΧΕΙΜΕΡΙΝΟ") && MonthPeriod.equals("ΧΕΙΜΕΡΙΝΟ"))
{
tmpexamrec.setPeriod(0);
}
else if(ExamPeriod.equals("ΕΑΡΙΝΟ") && MonthPeriod.equals("ΕΑΡΙΝΟ"))
{
tmpexamrec.setPeriod(1);
}
else          if(ExamPeriod.equals("ΣΕΠΤΕΜΒΡΙΟΣ")          &&
MonthPeriod.equals("ΣΕΠΤΕΜΒΡΙΟΣ"))
{
tmpexamrec.setPeriod(2);
}
else if(MonthPeriod.equals("AKYPO")) //Not Valid Month/Period Case..
{
FalsePeriodFlag=true;
Notification.show("Μήνας εξεταστικής μη συμβατός με την περίοδο
εξεταστικής!");
}
else //Not Valid Period Case..
{
FalsePeriodFlag=true;
Notification.show("Δεν έχει εισαχθεί σωστά η Περίοδος Εξεταστικής!");
}

//Here we check the Exam Type validation..
if(ExamType.equals("ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗΝΟΥ"))
{
tmpexamrec.setType(0);
}
else if(ExamType.equals("ΕΜΒΟΛΙΜΗ ΕΞΕΤΑΣΤΙΚΗ"))
{
tmpexamrec.setType(1);
}
else if(ExamType.equals("ΕΞΕΤΑΣΤΙΚΗ ΔΙΟΡΘΩΣΗΣ ΒΑΘΜΟΥ"))
{
tmpexamrec.setType(2);
}
else //Not valid Exam Type Case..
{
FalseTypeFlag=true;
}

```

```

Notification.show("Δεν έχει εισαχθεί σωστά η Κατηγορία Εξεταστικής!");
}

//Here we check the Date Validation..
if(!ExamStart.equals("ΕΤΟΣ-ΜΗΝΑΣ-ΜΕΡΑ") && !ExamEnd.equals("ΕΤΟΣ-ΜΗΝΑΣ-ΜΕΡΑ")
&& !ExamEnd.equals("") && !ExamStart.equals(""))
&& Date.valueOf(ExamStart).before(Date.valueOf(ExamEnd)) &&
!Date.valueOf(ExamStart).equals(Date.valueOf(ExamEnd)))
{
tmpexamrec.setStartDate(Date.valueOf(ExamStart));
tmpexamrec.setEndDate(Date.valueOf(ExamEnd));
}
else //Not Valid Date Case..
{
FalseDateFlag=true;
Notification.show("Δεν έχει εισαχθεί σωστά η Ημερομηνία (Έναρξης/Λήξης)
Εξεταστικής!");
}

try
{
//As long as everything is valid, we save the record in the Exam Table..
if(FalseYearFlag.equals(false) && FalsePeriodFlag.equals(false) &&
FalseTypeFlag.equals(false) && FalseDateFlag.equals(false))
{
examrep.save(tmpexamrec);
Notification.show("Η εισαγωγή των δεδομένων έγινε με επιτυχία!");
}
else //Else we inform the user..
{
Notification.show("Η εισαγωγή των δεδομένων δεν πραγματοποιήθηκε λόγω
ασυμβατότητας των στοιχείων που εισήχθησαν!");
}
}

catch(DataIntegrityViolationException e)
{
Notification.show("Η εισαγωγή των δεδομένων δεν πραγματοποιήθηκε λόγω του ότι
η εγγραφή υπάρχει είδη στο σύστημα!");
}

//After we insert our record into the Exam Table, we reset the form values and
we set visible the button to make accesible the Create ExamProf Form..
if(!(examrep.findAll().isEmpty()))
{
examproflinkBtn.setVisible(true);
}
else
{
examproflinkBtn.setVisible(false);
}

Year.setValue("ΕΤΟΣ");
Period.clear();
Type.clear();
StartDate.setValue("ΕΤΟΣ-ΜΗΝΑΣ-ΜΕΡΑ");
EndDate.setValue("ΕΤΟΣ-ΜΗΝΑΣ-ΜΕΡΑ");

});

//Here we have our Clear Button, that sets the init values to the Form..

```

```

clearBtn.addClickListener(event -> {
if(!(examrep.findAll().isEmpty()))
{
examproflinkBtn.setVisible(true);
}
else
{
examproflinkBtn.setVisible(false);
}
}

Year.setValue("ΕΤΟΣ");
StartDate.setValue("ΕΤΟΣ-ΜΗΝΑΣ-ΜΕΡΑ");
EndDate.setValue("ΕΤΟΣ-ΜΗΝΑΣ-ΜΕΡΑ");
Period.clear();
Type.clear();
Notification.show("Η εκκαθάριση της φόρμας έγινε με επιτυχία!");
});

//Here we have the Button that allow the user to navigate to the next View
which is a Form to Create an ExamProf Object and write it to the Database..
examproflinkBtn.addClickListener(event ->
UI.getCurrent().navigate("/secretariat/createxamprof"));

//Return to previous page via Return Button..
returnBtn.addClickListener(event ->
UI.getCurrent().navigate("/secretariat/"));

//Free the Session and return to Home Page via Exit Button..
exitBtn.addClickListener(event -> {
VaadinSession.getCurrent().setAttribute("user", null);
UI.getCurrent().navigate("/");
});
}

@Override
public void beforeEnter(BeforeEnterEvent event) {
if (!LoginView.class.equals(event.getNavigationTarget())
&& !loginService.isAllowed(VaadinSession.getCurrent().getAttribute("user"),
"Secretariat")) {
event.rerouteTo(LoginView.class);
}
}
}

```

Κώδικας από αρχείο «CreateExamprof.java» :

```

package net.uniwa.uniwaexamsproj.secretariat;

import com.vaadin.flow.component.Tag;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.combobox.ComboBox;
import com.vaadin.flow.component.dependency.JsModule;
import com.vaadin.flow.component.littemplate.LitTemplate;
import com.vaadin.flow.component.notification.Notification;
import com.vaadin.flow.component.polymertemplate.Id;
import com.vaadin.flow.component.textfield.TextField;
import com.vaadin.flow.router.BeforeEnterEvent;
import com.vaadin.flow.router.BeforeEnterObserver;

```

```

import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.VaadinSession;
import net.uniwa.uniwaexamsproj.exam.ExamModel;
import net.uniwa.uniwaexamsproj.exam.ExamRepository;
import net.uniwa.uniwaexamsproj.examprof.ExamProfessorsKey;
import net.uniwa.uniwaexamsproj.examprof.ExamProfessorsModel;
import net.uniwa.uniwaexamsproj.examprof.ExamProfessorsRepository;
import net.uniwa.uniwaexamsproj.lesson.LessonModel;
import net.uniwa.uniwaexamsproj.lesson.LessonRepository;
import net.uniwa.uniwaexamsproj.login.LoginService;
import net.uniwa.uniwaexamsproj.login.LoginView;
import net.uniwa.uniwaexamsproj.professor.ProfessorModel;
import net.uniwa.uniwaexamsproj.professor.ProfessorRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.dao.DataIntegrityViolationException;

import java.util.*;

@Route("/secretariat/createxamprof")
@Tag("create-examprof")
@JsModule("./src/views/secretariat/create-examprof.ts")
public class CreateExamprof extends LitTemplate implements
BeforeEnterObserver{
    @Autowired
    LoginService loginService;

    @Id("year")
    TextField Year;

    @Id("type")
    ComboBox Type;

    @Id("profname")
    ComboBox ProfName;

    @Id("lesstitle")
    ComboBox LessonTitle;

    @Id("submit")
    Button submitBtn;

    @Id("clear")
    Button clearBtn;

    @Id("exit")
    Button exitBtn;

    @Id("return")
    Button returnBtn;

    public CreateExamprof(ExamRepository examrep, LessonRepository lesrep,
    ProfessorRepository profrep , ExamProfessorsRepository examprofrep)
    {
        //Just setting some values to the Form..
        Year.setValue("ΕΤΟΣ");
        Type.clear();
        ProfName.clear();
        LessonTitle.clear();
        ArrayList<String> types = new ArrayList();
        types.add("ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗΝΟΥ");
        types.add("ΕΜΒΟΛΙΜΗ ΕΞΕΤΑΣΤΙΚΗ");
    }
}

```

```

types.add("ΕΞΕΤΑΣΤΙΚΗ ΔΙΟΡΘΩΣΗΣ ΒΑΘΜΟΥ");
types.add("ΕΞΕΤΑΣΤΙΚΗ ΣΕΠΤΕΜΒΡΙΟΥ");
Type.setItems(types);
ArrayList<String> profsurnames = profrep.findprofsurname();
ArrayList<String> lestitles = lesrep.findlesstontitles();
ProfName.setItems(profsurnames);
LessonTitle.setItems(lestitles);

//Below we got all the Actions the Submit Button does..
submitBtn.addClickListener(event -> {
Boolean FalseYearFlag=false;
Boolean FalseTypeFlag=false;
Boolean FalseProfNameFlag=false;
Boolean FalseLesTitleFlag=false;
Integer PeriodValue = -1;
Integer TypeValue = 0;
String ExYear = Year.getValue();
String ExamType = "";
String PrName = "";
String LsTitle = "";
String CurrentYear = java.time.Year.now().toString();

//After we got our vars initiated, we got our values from the form..
try
{
ExamType = Type.getValue().toString();
}

catch(NullPointerException NullEx)
{
ExamType="";
}

try
{
PrName = ProfName.getValue().toString();
}

catch(NullPointerException NullEx)
{
PrName = "";
}

try
{
LsTitle = LessonTitle.getValue().toString();
}

catch(NullPointerException NullEx)
{
LsTitle = "";
}

//Below we check the validation of each value..
if(ExYear.equals("ΕΤΟΣ") || ExYear.equals("") || !ExYear.equals(CurrentYear))
{
FalseYearFlag=true;
Notification.show("Δεν έχει εισαχθεί σωστά το Έτος Εξεταστικής!");
}

//Here we check the Exam Type validation..

```

```

if(ExamType.equals("ΕΞΕΤΑΣΤΙΚΗ ΕΞΑΜΗΝΟΥ"))
{
TypeValue = 0;
}
else if(ExamType.equals("ΕΜΒΟΛΙΜΗ ΕΞΕΤΑΣΤΙΚΗ"))
{
TypeValue = 1;
}
else if(ExamType.equals("ΕΞΕΤΑΣΤΙΚΗ ΔΙΟΡΘΩΣΗΣ ΒΑΘΜΟΥ"))
{
TypeValue = 2;
}
else if(ExamType.equals("ΕΞΕΤΑΣΤΙΚΗ ΣΕΠΤΕΜΒΡΙΟΥ"))
{
TypeValue = 3;
}
else //Not valid Exam Type Case..
{
FalseTypeFlag=true;
Notification.show("Δεν έχει εισαχθεί σωστά η Κατηγορία Εξεταστικής!");
}

if(PrName.equals(""))
{
FalseProfNameFlag=true;
Notification.show("Δεν έχει εισαχθεί σωστά το Επώνυμο του Καθηγητή!");
}

if(LsTitle.equals(""))
{
FalseLesTitleFlag=true;
Notification.show("Δεν έχει εισαχθεί σωστά ο Τίτλος του Μαθήματος!");
}

if(FalseLesTitleFlag.equals(false))
{
String LessonID = lesrep.findlesid(LsTitle);
LessonModel Lesson = lesrep.FindLessonByID(LessonID);

if(TypeValue.equals(3) && (examrep.findexam(ExYear,2,0) == null))
{
FalseLesTitleFlag = true;
Notification.show("Η περίοδος του εξαμήνου του Μαθήματος δεν αντιστοιχεί σε
καμία Εξεταστική Περίοδο του Συστήματος!");
}
else
if(!TypeValue.equals(3) &&
examrep.findexam(ExYear,Lesson.getPeriod(),TypeValue) == null)
{
FalseLesTitleFlag = true;
Notification.show("Η περίοδος του εξαμήνου του Μαθήματος δεν αντιστοιχεί σε
καμία Εξεταστική Περίοδο του Συστήματος!");
}
else
{
PeriodValue = Lesson.getPeriod();
}
}

try {

```



```

//If all the values are ok and valid, we prepare our ExamProf object and we
save it to the database..
if (FalseYearFlag.equals(false) && FalseTypeFlag.equals(false) &&
FalseProfNameFlag.equals(false) && FalseLesTitleFlag.equals(false) &&
!PeriodValue.equals(-1))
{
Integer ExamKey = null;
String LesKey = null;
String ProfKey = null;

if(!TypeValue.equals(3))
{
ExamKey = examrep.findexam(ExYear, PeriodValue,TypeValue);
}
else
{
ExamKey = examrep.findexam(ExYear, 2,0);
}

ProfKey = profrep.findprofid(PrName);
LesKey = lesrep.findlesid(LsTitle);

ExamProfessorsModel ExamProfRecord = new ExamProfessorsModel();
ExamProfessorsKey ExamProfKey = new ExamProfessorsKey();

ExamProfKey.setExamID(ExamKey);
ExamProfKey.setProfID(ProfKey);
ExamProfKey.setLessonID(LesKey);

ExamProfRecord.setExamprofkey(ExamProfKey);
ExamProfRecord.setExamprofobj(examrep.findById(ExamKey));
ExamProfRecord.setLesprofobj(lesrep.findById(LesKey));
ExamProfRecord.setProfobj(profrep.findById(ProfKey));

examprofrep.save(ExamProfRecord);

Notification.show("Η εισαγωγή των δεδομένων έγινε με επιτυχία!");
} else //Here we inform our user that some or all of the data that has given
to us are invalid..
{
Notification.show("Η εισαγωγή των δεδομένων δεν πραγματοποιήθηκε λόγω
ασυμβατότητας των στοιχείων που εισήχθησαν!");
}
}

catch(DataIntegrityViolationException e)
{
Notification.show("Η εισαγωγή των δεδομένων δεν πραγματοποιήθηκε λόγω του ότι
η εγγραφή υπάρχει είδη στο σύστημα!");
}

//Reset the Form values..
Year.setValue("ΕΤΟΣ");
Type.clear();
ProfName.clear();
LessonTitle.clear();
});

//Below we got the Button that Clears the Form actions..
clearBtn.addClickListener(event -> {

```

```

Year.setValue("ΕΤΟΣ");
Type.clear();
ProfName.clear();
LessonTitle.clear();

Notification.show("Η εκκαθάριση της φόρμας έγινε με επιτυχία!");
});

//Return to previous page via Return Button..
returnBtn.addClickListener(event                                     ->
UI.getCurrent().navigate("secretariat/exam-create"));

//Free the Session and return to Home Page via Exit Button..
exitBtn.addClickListener(event -> {
VaadinSession.getCurrent().setAttribute("user", null);
UI.getCurrent().navigate("/");
});
}

@Override
public void beforeEnter(BeforeEnterEvent event) {
if (!LoginView.class.equals(event.getNavigationTarget())
&& !loginService.isAllowed(VaadinSession.getCurrent().getAttribute("user"),
"Secretariat")) {
event.rerouteTo(LoginView.class);
}
}
}

```

Κώδικας από αρχείο «UpdateExamInfoView.java» :

```

package net.uniwa.uniwaexamsproj.secretariat;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.time.LocalDate;
import java.time.ZoneId;
import java.time.format.DateTimeFormatter;
import java.util.List;
import java.util.stream.Collectors;
import java.util.stream.IntStream;
import com.vaadin.flow.component.notification.Notification;
import net.uniwa.uniwaexamsproj.examclass.ExamClassRepository;
import net.uniwa.uniwaexamsproj.examprof.ExamProfessorsRepository;
import org.springframework.beans.factory.annotation.Autowired;
import com.vaadin.flow.component.Tag;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.combobox.ComboBox;
import com.vaadin.flow.component.datepicker.DatePicker;
import com.vaadin.flow.component.dependency.JsModule;
import com.vaadin.flow.component.grid.Grid;
import com.vaadin.flow.component.grid.editor.Editor;
import com.vaadin.flow.component.littemplate.LitTemplate;
import com.vaadin.flow.component.orderedlayout.HorizontalLayout;
import com.vaadin.flow.component.polymertemplate.Id;
import com.vaadin.flow.component.select.Select;

```

```

import com.vaadin.flow.component.textfield.TextField;
import com.vaadin.flow.data.binder.Binder;
import com.vaadin.flow.router.BeforeEnterEvent;
import com.vaadin.flow.router.BeforeEnterObserver;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.VaadinSession;
import net.uniwa.uniwaexamsproj.exam.ExamModel;
import net.uniwa.uniwaexamsproj.exam.ExamRepository;
import net.uniwa.uniwaexamsproj.login.LoginService;
import net.uniwa.uniwaexamsproj.login.LoginView;
import java.util.Date;

//A Designer generated component for the update-exam-info-view template..
//Designer will add and remove fields with @Id mappings but does not overwrite
or otherwise change this file..

//Some basic annotations to map the route of the class, to give a tag and to
associate it with the typescript file in the frontend package..
@Route("/secretariat/update-exam")
@Tag("update-exam-info-view")
@JsModule("./src/views/secretariat/update-exam-info-view.ts")
public class UpdateExamInfoView extends LitTemplate implements
BeforeEnterObserver { //Inherit the components/methods of LitTemplate for the
View & Implement the BeforeEnterObserver Interface for the BeforeEnterMethod..

//Declaration of the LoginService and the Buttons of the Template in the View
of our Frontend Page..
@Autowired
LoginService loginService;

@Autowired
SecretariatService secrServ;

@Autowired
ExamClassRepository examclassrep;

@Autowired
ExamProfessorsRepository examprofrep;

@Id("return")
Button returnBtn;

@Id("exit")
Button exitBtn;

@Id("exams")
Grid<ExamModel> examsGrid;

//Some initiations & the set of the Date Format for our view..
SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");

Integer selectedId;

private static final String Initial_ID = "XEIM2023"; //To check each time
whether we need or not the Initializing Records Button..

private static Integer initcount = 0; //A counter to use the file load function
for ExamClass/Prof Table only once at the beginning..

//Creates a new UpdateExamInfoView..

```

```

@SuppressWarnings("unchecked")
public UpdateExamInfoView(ExamRepository examRepo) { //Pass the Repositories
inside to have communication with the Database without SQL Queries..

//Define ExamsGrid columns and define associated variables..
Editor<ExamModel> editor = examsGrid.getEditor();

//Define Columns
examsGrid.setSizeUndefined();

Grid.Column<ExamModel>          yearColumn          =
examsGrid.addColumn(ExamModel::getYear).setHeader("ΕΤΟΣ");

//Define associated variables..
Grid.Column<ExamModel> periodColumn = examsGrid.addColumn(e -> {
switch (e.getPeriod()) {
case 0:
return "ΧΕΙΜΕΡΙΝΗ";
case 1:
return "ΕΑΡΙΝΗ";
case 2:
return "ΣΕΠΤΕΜΒΡΙΟΥ";
}
return "";
}).setHeader("ΠΕΡΙΟΔΟΣ");

//Define associated variables..
Grid.Column<ExamModel> typeColumn = examsGrid.addColumn(e -> {
switch (e.getType()) {
case 0:
return "ΜΑΘΗΜΑ ΣΕ ΑΥΤΗΝ ΤΗΝ ΠΕΡΙΟΔΟ";
case 1:
return "ΜΑΘΗΜΑ ΑΛΛΗΣ ΠΕΡΙΟΔΟΥ";
case 2:
return "ΜΑΘΗΜΑ ΠΡΟΣ ΔΙΟΡΘΩΣΗ ΒΑΘΜΟΥ";
}
return "";
}).setHeader("ΚΑΤΗΓΟΡΙΑ");

Grid.Column<ExamModel>          startColumn          =          examsGrid.addColumn(e-
>sdf.format(e.getStartDate())).setHeader("ΕΝΑΡΞΗΣ");
Grid.Column<ExamModel>          endColumn            =          examsGrid.addColumn(e-
>sdf.format(e.getEndDate())).setHeader("ΛΗΞΗΣ");

//Define Edit column that contains Edit button..
Grid.Column<ExamModel> editColumn = examsGrid.addComponentColumn(exam -> {
Button editButton = new Button("ΕΠΕΞΕΡΓΑΣΙΑ ΕΓΓΡΑΦΗΣ");
editButton.addClickListener(e -> {
selectedId = exam.getID();
if (editor.isOpen())
editor.cancel();
});

//Open Editing Mode..
examsGrid.getEditor().editItem(exam);
});

return editButton;
}).setWidth("250px").setFlexGrow(0);

Binder<ExamModel> binder = new Binder<>(ExamModel.class);
editor.setBinder(binder);

```

```

editor.setBuffered(true);

TextField idTField = new TextField();

LocalDate now = LocalDate.now(ZoneId.systemDefault());

List<Integer> selectableYears = IntStream.range(now.getYear() - 5,
now.getYear() + 7).boxed()
.collect(Collectors.toList());

ComboBox<Integer> yearPicker = new ComboBox<>("", selectableYears);

binder.forField(yearPicker).bind(e -> Integer.parseInt(e.getYear()), (e, v) -
> e.setYear(v.toString()));
yearColumn.setEditorComponent(yearPicker);

Select<String> periods = new Select<>();
periods.setItems("ΧΕΙΜΕΡΙΝΗ", "ΕΑΡΙΝΗ", "ΣΕΠΤΕΜΒΡΙΟΥ");

//Define specific setters and getters for the period to match values and
texts..
binder.forField(periods).bind(e -> {
switch (e.getPeriod()) {
case 0:
return "ΧΕΙΜΕΡΙΝΗ";
case 1:
return "ΕΑΡΙΝΗ";
case 2:
return "ΣΕΠΤΕΜΒΡΙΟΥ";
}
return null;
}, (e, v) -> {
switch (v) {
case "ΧΕΙΜΕΡΙΝΗ":
e.setPeriod(0);
break;
case "ΕΑΡΙΝΗ":
e.setPeriod(1);
break;
case "ΣΕΠΤΕΜΒΡΙΟΥ":
e.setPeriod(2);
break;
}
});

periodColumn.setEditorComponent(periods);

Select<String> types = new Select<>();
types.setItems("ΒΑΘΜΟΣ ΕΞΑΜΗΝΟΥ", "ΕΠΑΝΑΛΑΜΒΑΝΩ", "ΔΙΟΡΘΩΣΗ");

//Set specific setters and getters for types variable..
binder.forField(types).bind(e -> {
switch (e.getType()) {
case 0:
return "ΒΑΘΜΟΣ ΕΞΑΜΗΝΟΥ";
case 1:
return "ΕΠΑΝΑΛΑΜΒΑΝΩ";
case 2:
return "ΔΙΟΡΘΩΣΗ";
}
return null;
}

```

```

}, (e, v) -> {
switch (v) {
case "ΒΑΘΜΟΣ ΕΞΑΜΗΝΟΥ":
e.setType(0);
break;
case "ΕΠΑΝΑΛΑΜΒΑΝΩ":
e.setType(1);
break;
case "ΔΙΟΡΘΩΣΗ":
e.setType(2);
break;
}
});
typeColumn.setEditorComponent(types);

DatePicker startField = new DatePicker("");
DateTimeFormatter df = DateTimeFormatter.ofPattern("dd-MM-yyyy");

binder.forField(startField).bind(e                                     ->
e.getStartDate().toInstant().atZone(ZoneId.systemDefault()).toLocalDate(),
(e, v)                                                                ->
e.setStartDate(Date.from(v.atStartOfDay(ZoneId.systemDefault()).toInstant())
));
startColumn.setEditorComponent(startField);

DatePicker endField = new DatePicker("");
binder.forField(endField).bind(e                                     ->
e.getStartDate().toInstant().atZone(ZoneId.systemDefault()).toLocalDate(),
(e, v)                                                                ->
e.setEndDate(Date.from(v.atStartOfDay(ZoneId.systemDefault()).toInstant()))
);
endColumn.setEditorComponent(endField);

//Update-Save Exam Info..
Button saveButton = new Button("ΕΝΗΜΕΡΩΣΗ", e -> {
editor.save();
updateGrid(examRepo);
Notification.show("Η ενημέρωση σας έγινε με επιτυχία!");
});

editor.addSaveListener(item -> {
//Delete the previous object if the id changed..
if(!selectedId.equals(item.getItem().getID()))
examRepo.deleteById(selectedId);

examRepo.save(item.getItem());
});

HorizontalLayout actions = new HorizontalLayout(saveButton);
actions.setPadding(false);
editColumn.setEditorComponent(actions);

examsGrid.setItems(examRepo.findAll());

//Return to previous page via Return Button..
returnBtn.addClickListener(event                                     ->
UI.getCurrent().navigate("/secretariat/"));

//Free the Session and return to Home Page via Exit Button..
exitBtn.addClickListener(event -> {
VaadinSession.getCurrent().setAttribute("user", null);
}
}

```

```

UI.getCurrent().navigate("/");
});

}

//Here we update the data on the grid and we also refresh the page by that..
private void updateGrid(ExamRepository examrep) {
examsGrid.setItems(examrep.findAll());
}

//Check if Authenticated and user is Secretariat..
@Override
public void beforeEnter(BeforeEnterEvent event) {
if (!LoginView.class.equals(event.getNavigationTarget())
&& !loginService.isAllowed(VaadinSession.getCurrent().getAttribute("user"),
"Secretariat")) {
event.rerouteTo(LoginView.class);
}
}
}
}

```

Κώδικας από αρχείο «ExamInfoView.java» :

```

package net.uniwa.uniwaexamsproj.secretariat;

import java.text.SimpleDateFormat;
import org.springframework.beans.factory.annotation.Autowired;
import com.vaadin.flow.component.Tag;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.dependency.JsModule;
import com.vaadin.flow.component.grid.Grid;
import com.vaadin.flow.component.littemplate.LitTemplate;
import com.vaadin.flow.component.polymertemplate.Id;
import com.vaadin.flow.router.BeforeEnterEvent;
import com.vaadin.flow.router.BeforeEnterObserver;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.VaadinSession;
import net.uniwa.uniwaexamsproj.classavailability.ClassAvailModel;
import net.uniwa.uniwaexamsproj.classavailability.ClassAvailRepository;
import net.uniwa.uniwaexamsproj.exam.ExamModel;
import net.uniwa.uniwaexamsproj.exam.ExamRepository;
import net.uniwa.uniwaexamsproj.login.LoginService;
import net.uniwa.uniwaexamsproj.login.LoginView;

//A Designer generated component for the exam-info-view template..
//Designer will add and remove fields with @Id mappings but does not overwrite
or otherwise change this file..

//Some basic annotations to map the route of the class, to give a tag and to
associate it with the typescript file in the frontend package..
@Route("secretariat/exam-info")
@Tag("exam-info-view")
@JsModule("./src/views/secretariat/exam-info-view.ts")
public class ExamInfoView extends LitTemplate implements BeforeEnterObserver
{ //Inherit the components/methods of LitTemplate for the View & Implement the
BeforeEnterObserver Interface for the BeforeEnterMethod..

```

```

//Declaration of the Login-Secretariat Service and the Buttons of the Template
in the View of our Frontend Page..
@Autowired
LoginService loginService;

@Autowired
SecretariatService secretService;

@Id("return")
Button returnBtn;

@Id("exit")
Button exit;

@Id("exams")
Grid<ExamModel> examsGrid;

@Id("classes")
Grid<ClassAvailModel> classesGrid;

//Use European Date Format..
SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");

//Creates a new ExamInfoView..

public ExamInfoView(ExamRepository examRepo, ClassAvailRepository
classavailRepo) {
//Define examsGrid columns..
examsGrid.addColumn(ExamModel::getID).setHeader("ID");
examsGrid.addColumn(ExamModel::getYear).setHeader("ΕΤΟΣ");

//Associate Exam Period values at the frontend to the values used at the
backend (database tables)..
examsGrid.addColumn(e -> {
switch (e.getPeriod()) {
case 0:
return "ΧΕΙΜΕΡΙΝΗ";
case 1:
return "ΕΑΡΙΝΗ";
case 2:
return "ΣΕΠΤΕΜΒΡΙΟΥ";
}
return "";
}).setHeader("ΠΕΡΙΟΔΟΣ");

//Associate Exam Type values at the frontend to the values used at the backend
(database tables)..
examsGrid.addColumn(e -> {
switch (e.getType()) {
case 0:
return "ΜΑΘΗΜΑ ΣΕ ΑΥΤΗΝ ΤΗΝ ΠΕΡΙΟΔΟ";
case 1:
return "ΜΑΘΗΜΑ ΑΛΛΗΣ ΠΕΡΙΟΔΟΥ";
case 2:
return "ΜΑΘΗΜΑ ΠΡΟΣ ΔΙΟΡΘΩΣΗ ΒΑΘΜΟΥ";
}
return "";
}).setHeader("ΚΑΤΗΓΟΡΙΑ");

//Define some Exams & Classes Grid Columns..
examsGrid.addColumn(e->sdf.format(e.getStartDate())).setHeader("ΕΝΑΡΞΗΣ");

```



```

examsGrid.addColumn(e->sdf.format(e.getEndDate())).setHeader("ΛΗΞΗΣ");

classesGrid.addColumn(c -> c.getId().getClassID()).setHeader("ID ΑΙΘΟΥΣΑΣ");
classesGrid.addColumn(c -> c.getId().getDtslotID()).setHeader("ID
ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ");

//Fill Class Grid Table with Values..
classesGrid.setItems(classavailRepo.findByAllocated(true));

//Fill Exam Grid Table with Values..
examsGrid.setItems(examRepo.findAll());

//Return to previous page via Return Button..
returnBtn.addClickListener(event -> {
UI.getCurrent().navigate("/secretariat/");
});

//Free the Session and return to Home Page via Exit Button..
exit.addClickListener(event -> {
VaadinSession.getCurrent().setAttribute("user", null);
UI.getCurrent().navigate("/");
});

}

//Check if Authenticated and user is Secretariat..
@Override
public void beforeEnter(BeforeEnterEvent event) {
if (!LoginView.class.equals(event.getNavigationTarget())
&& !loginService.isAllowed(VaadinSession.getCurrent().getAttribute("user"),
"Secretariat")) {
event.rerouteTo(LoginView.class);
}
}
}
}

```

Κώδικας από αρχείο «AllocateExamRoomView.java» :

```

package net.uniwa.uniwaexamsproj.secretariat;

import java.text.SimpleDateFormat;

import net.uniwa.uniwaexamsproj.classavailability.ClassAvailRepository;
import org.springframework.beans.factory.annotation.Autowired;
import com.vaadin.flow.component.Tag;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.dependency.JsModule;
import com.vaadin.flow.component.grid.Grid;
import com.vaadin.flow.component.littemplate.LitTemplate;
import com.vaadin.flow.component.notification.Notification;
import com.vaadin.flow.component.polymertemplate.Id;
import com.vaadin.flow.router.BeforeEnterEvent;
import com.vaadin.flow.router.BeforeEnterObserver;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.VaadinSession;
import net.uniwa.uniwaexamsproj.classavailability.ClassAvailModel;
import net.uniwa.uniwaexamsproj.classes.ClassModel;
import net.uniwa.uniwaexamsproj.classes.ClassRepository;

```

```

import net.uniwa.uniwaexamsproj.exam.ExamRepository;
import net.uniwa.uniwaexamsproj.login.LoginService;
import net.uniwa.uniwaexamsproj.login.LoginView;
import net.uniwa.uniwaexamsproj.timeslot.TimeslotModel;
import net.uniwa.uniwaexamsproj.timeslot.TimeslotRepository;

//A Designer generated component for the allocate-exam-room-view template..
//Designer will add and remove fields with @Id mappings but does not overwrite
or otherwise change this file..

//Some basic annotations to map the route of the class, to give a tag and to
associate it with the typescript file in the frontend package..
@Route("secretariat/allocate-room")
@Tag("allocate-exam-room-view")
@JsModule("./src/views/secretariat/allocate-exam-room-view.ts")
public class AllocateExamRoomView extends LitTemplate implements
BeforeEnterObserver { //Inherit the components/methods of LitTemplate for the
View & Implement the BeforeEnterObserver Interface for the BeforeEnterMethod..

//Declaration of the LoginService and the Buttons of the Template in the View
of our Frontend Page..
@Autowired
LoginService loginService;

@Autowired
SecretariatService secretService;

@Id("classes")
Grid<ClassModel> classesGrid;

@Id("timeslots")
Grid<TimeslotModel> timeslotsGrid;

@Id("allocated")
Grid<ClassAvailModel> allocatedGrid;

@Id("return")
Button returnBtn;

@Id("exit")
Button exitBtn;

//Some initiations for the selected Timeslot and Class vars & the set of the
Date Format for our view..
private TimeslotModel selectedTime = null;

private ClassModel selectedClass = null;

private ClassAvailModel selectedclassavail = null;

SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");

//Creates a new AllocateExamRoomView..
public AllocateExamRoomView(ClassRepository classRepo, ExamRepository
examRepo, TimeslotRepository timeslotRepo, ClassAvailRepository
classavailrep) { //Pass the Repositories inside to have communication with the
Database without SQL Queries..

//Define AllocatedGrid Columns..
allocatedGrid.addColumn(e -> e.getId().getClassID()).setHeader("ID TAΞH");

```

```

allocatedGrid.addColumn(e      ->      e.getId().getDTslotID()).setHeader("ID
XPONOΘHKH");

//Define TimeslotsGrid Columns..
timeslotsGrid.addColumn(TimeslotModel::getID).setHeader("ID");
timeslotsGrid.addColumn(e->sdf.format(e.getDate())).setHeader("ΗΜΕΡΟΜΗΝΙΑ");
timeslotsGrid.addColumn(TimeslotModel::getStart).setHeader("ΕΝΑΡΞΗ");
timeslotsGrid.addColumn(TimeslotModel::getEnd).setHeader("ΤΕΛΟΣ");

//Fill with Values TimeslotsGrid and AllocatedGrid after clicking Allocate
Button..
timeslotsGrid.addComponentColumn(c -> {
Button button = new Button("ΕΠΙΛΟΓΗ");
//Create a Button for Selection and Below we write the code for it's work
after being clicked..
button.addClickListener(e -> {
Integer    Period    =    classavailrep.getPeriodbyID(selectedClass.getID(),
c.getID()); //Find the Period of each record to update the database..

secretService.setclassavailinfo(selectedClass.getID(),    c.getID(),    true,
Period); //Here we call the setclassavailinfo() method from the
SecretariatService.java class to inform the Table with the selection of the
class and the timeslot..

//Reload the SelectedClass Values into the Timeslot Grid..
timeslotsGrid.setItems(secretService.getAvailableTimeslotsByClass(selectedCl
ass.getID()));

//Fill with Values the AllocatedGrid with the new Values..
allocatedGrid.setItems(secretService.getAllocatedTimeslotsByClass(selectedCl
ass.getID()));

//Show the Notification via message on screen to the User to let his know
everything went good..
Notification.show("Η εγγραφή σας έγινε με επιτυχία!");
});

return button;
}).setWidth("150px").setFlexGrow(0); //Set coordinates for the last result in
the view..

//Define ClassesGrid Columns..
classesGrid.addColumn(ClassModel::getID).setHeader("ID");
classesGrid.addColumn(ClassModel::getCapacity).setHeader("ΧΩΡΗΤΙΚΟΤΗΤΑ");
classesGrid.addColumn(e->{
switch(e.getBlankSeatMode()) {
case 0: return "ΜΙΑ ΚΕΝΗ ΘΕΣΗ";
case 1: return "ΔΥΟ ΚΕΝΕΣ ΘΕΣΕΙΣ";
default: return "";
}
}).setHeader("ΚΕΝΑ ΑΝΑ ΘΕΣΕΙΣ");

classesGrid.addColumn(ClassModel::getMinSupNo).setHeader("ΠΛΗΘΟΣ
ΕΠΙΤΗΡΗΤΩΝ");

classesGrid.addComponentColumn(c -> {
Button editButton = new Button("ΔΙΑΘΕΣΙΜΟΤΗΤΑ");

editButton.addClickListener(e -> {

selectedClass = c;

```

```

//Fill with Values the TimeslotsGrid after a select of the class..
timeslotsGrid.setItems(secretService.getAvailableTimeslotsByClass(c.getID())
);

//Fill with Values the AllocatedGrid after the selection of a class and a
timeslot..
allocatedGrid.setItems(secretService.getAllocatedTimeslotsByClass(c.getID())
);

});

return editButton;
}).setWidth("150px").setFlexGrow(0);

allocatedGrid.addComponentColumn(c -> {
Button deleteButton = new Button("ΑΦΑΙΡΕΣΗ");

deleteButton.addClickListener(e -> {

//Keep the two id's to set the new records on the grid..
String tmpclassid = c.getId().getClassID();
Integer tmpdslotid = c.getId().getDTslotID();

Integer Period = classavailrep.getPeriodbyID(tmpclassid, tmpdslotid); //Find
the Period of each record to update the database..

secretService.setclassavailinfo(tmpclassid, tmpdslotid, false, Period);
//Here we call the setclassavailinfo() method from the SecretariatService.java
class to inform the Table with the selection of the class and the timeslot..

//Fill with Values the AllocatedGrid after the selection of a class and a
timeslot..
allocatedGrid.setItems(secretService.getAllocatedTimeslotsByClass(tmpclassid
));

//Reload the SelectedClass Values into the Timeslot Grid..
timeslotsGrid.setItems(secretService.getAvailableTimeslotsByClass(selectedCl
ass.getID()));

Notification.show("Η διαγραφή έγινε με επιτυχία!");
});

return deleteButton;
}).setWidth("150px").setFlexGrow(0);

classesGrid.setItems(classRepo.findAll());

//Return to previous page by clicking the Return Button..
returnBtn.addClickListener(event -> {
UI.getCurrent().navigate("/secretariat/");
});

//Return to the Home Page & Free the Session by clicking the Exit Button..
exitBtn.addClickListener(event -> {
VaadinSession.getCurrent().setAttribute("user", null);
UI.getCurrent().navigate("/");
});
}

```

```

//The BeforeEnter Method but Overriden..
//Here we check if the User has Logged in and is a Secretariat Stuff..
@Override
public void beforeEnter(BeforeEnterEvent event) {
if (!LoginView.class.equals(event.getNavigationTarget()))
&& !loginService.isAllowed(VaadinSession.getCurrent().getAttribute("user"),
"Secretariat")) {
event.rerouteTo(LoginView.class);
}
}
}
}

```

Κώδικας από αρχείο «ProfessorMainView.java» :

```

package net.uniwa.uniwaexamsproj.professor;

import org.springframework.beans.factory.annotation.Autowired;
import com.vaadin.flow.component.Tag;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.dependency.JsModule;
import com.vaadin.flow.component.littemplate.LitTemplate;
import com.vaadin.flow.component.polymertemplate.Id;
import com.vaadin.flow.router.BeforeEnterEvent;
import com.vaadin.flow.router.BeforeEnterObserver;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.VaadinSession;
import net.uniwa.uniwaexamsproj.login.LoginService;
import net.uniwa.uniwaexamsproj.login.LoginView;

/**
 * A Designer generated component for the professor-view template.
 *
 * Designer will add and remove fields with @Id mappings but
 * does not overwrite or otherwise change this file.
 */
@Route("professor")
@Tag("professor-main-view")
@JsModule("./src/views/professor/professor-main-view.ts")
public class ProfessorMainView extends LitTemplate implements
BeforeEnterObserver{

    @Autowired
    LoginService loginService;

    @Id("applications")
    Button applicationsBtn;

    @Id("exit")
    Button exitBtn;

    public ProfessorMainView() {
//Here we redirect the user to the Application View, using the Application
Button..
applicationsBtn.addClickListener(event ->{
UI.getCurrent().navigate("professor/applications");
}
}
}

```

```

});

//Here we redirect the user to the Login View, using the Exit Button..
exitBtn.addClickListener(event ->{
VaadinSession.getCurrent().setAttribute("user", null);
UI.getCurrent().navigate("/");
});
}

//BeforeEnter is activated before the view rendering..
//Checks if the user is logged in and Role is professor..
@Override
public void beforeEnter(BeforeEnterEvent event) {
if (!LoginView.class.equals(event.getNavigationTarget()) &&
!loginService.isAllowed(VaadinSession.getCurrent().getAttribute("user"),
"Professor")) {
event.rerouteTo(LoginView.class);
}
}
}
}

```

Κώδικας από αρχείο «ProfessorApplicationsView.java» :

```

package net.uniwa.uniwaexamsproj.professor;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

import net.uniwa.uniwaexamsproj.examprof.ExamProfessorsKey;
import net.uniwa.uniwaexamsproj.examprof.ExamProfessorsModel;
import net.uniwa.uniwaexamsproj.examprof.ExamProfessorsRepository;
import net.uniwa.uniwaexamsproj.examstud.ExamStudentsRepository;
import net.uniwa.uniwaexamsproj.student.StudentRepository;
import org.springframework.beans.factory.annotation.Autowired;
import com.vaadin.flow.component.Tag;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.button.Button;
import com.vaadin.flow.component.dependency.JsModule;
import com.vaadin.flow.component.grid.Grid;
import com.vaadin.flow.component.littemplate.LitTemplate;
import com.vaadin.flow.component.polymertemplate.Id;
import com.vaadin.flow.router.BeforeEnterEvent;
import com.vaadin.flow.router.BeforeEnterObserver;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.VaadinSession;
import net.uniwa.uniwaexamsproj.exam.ExamModel;
import net.uniwa.uniwaexamsproj.exam.ExamRepository;
import net.uniwa.uniwaexamsproj.examstud.ExamStudentsModel;
import net.uniwa.uniwaexamsproj.login.LoginService;
import net.uniwa.uniwaexamsproj.login.LoginView;
import net.uniwa.uniwaexamsproj.student.StudentModel;

@Route("professor/applications")
@Tag("professor-applications-view")
@JsModule("./src/views/professor/professor-applications-view.ts")

```

```

public class ProfessorApplicationsView extends LitTemplate implements
BeforeEnterObserver{

    @Autowired
    LoginService loginService;

    @Id("exit")
    Button exit;

    @Id("exams")
    Grid<ExamModel> examsGrid;

    @Id("students")
    Grid<StudentModel> studentsGrid;

    //Here we use date European Date format..
    SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");

    public ProfessorApplicationsView(ExamRepository examRepo,
ExamProfessorsRepository examprofrep, ExamStudentsRepository examstudrep,
StudentRepository studrep) {

        //Define exams grid columns..
        examsGrid.addColumn(ExamModel::getID).setHeader("ID ΕΞΕΤΑΣΤΙΚΗΣ");

        //Map period grid values to text..
        examsGrid.addColumn(e -> {
            switch (e.getPeriod()) {
                case 0:
                    return "ΧΕΙΜΕΡΙΝΗ";
                case 1:
                    return "ΕΑΡΙΝΗ";
                case 2:
                    return "ΣΕΠΤΕΜΒΡΙΟΥ";
            }
            return "";
        }).setHeader("ΠΕΡΙΟΔΟΣ");

        //Map type column values to text..
        examsGrid.addColumn(e -> {
            switch (e.getType()) {
                case 0:
                    return "Μάθημα σε αυτήν την περίοδο";
                case 1:
                    return "Μάθημα από μια άλλη περίοδο";
                case 2:
                    return "Μάθημα που θέλουμε να διορθώσουμε τον βαθμό";
            }
            return "";
        }).setHeader("Κατηγορία Μαθήματος");
        examsGrid.addColumn(e->sdf.format(e.getStartDate())).setHeader("Ημερομηνία Έναρξη");
        examsGrid.addColumn(e->sdf.format(e.getEndDate())).setHeader("Ημερομηνία Λήξη");

        //The student grid must be defined before the calling to the display button
        in the examsGrid..
        studentsGrid.addColumn(StudentModel::getID).setHeader("ID");
        studentsGrid.addColumn(StudentModel::getName).setHeader("Όνομα");
        studentsGrid.addColumn(StudentModel::getSurname).setHeader("Επώνυμο");
    }
}

```

```

examsGrid.addComponentColumn(c -> {
Button button = new Button("ΕΜΦΑΝΙΣΗ");

button.addClickListener(e -> {
List<StudentModel> list = new ArrayList();
List<String> StudIDs = new ArrayList<String>();
List<ExamProfessorsModel> examprof = new ArrayList<ExamProfessorsModel>();

ProfessorModel prf = (ProfessorModel)
VaadinSession.getCurrent().getAttribute("user"); //Here we get the
ProfessorModel via Session..
examprof = examprofrep.getProfDetails(prf.getID(),c.getID()); //Here we get
ExamProfModel via ProfModel (ProfID) to use ExamID & LesID..
Iterator<ExamProfessorsModel> examprofite = examprof.listIterator();

while(examprofite.hasNext()) //For each Professor/Exam ID Set..
{
ExamProfessorsKey tmpexamprofkey = examprofite.next().getExamprofkey();
StudIDs=examstudrep.GetStudentIDs(tmpexamprofkey.getLessonID(),
tmpexamprofkey.getExamID()); //Here we get all the Student IDs via the
ExamProfModel using ExamID & LesID..
}

Iterator<String> StudIDsIte = StudIDs.listIterator();
while (StudIDsIte.hasNext()) //Set the Team of Students..
{
String tmpstud = StudIDsIte.next();
list.add(studrep.StudentDetails(tmpstud));
}

studentsGrid.setItems(list);
});

return button;
}).setWidth("150px").setFlexGrow(0);

examsGrid.setItems(examRepo.findAll());

//Navigate to home page and delete user from session..
exit.addClickListener(event ->{
VaadinSession.getCurrent().setAttribute("user", null);
UI.getCurrent().navigate("/");
});
}

//Check if user is connected and role is Professor..
@Override
public void beforeEnter(BeforeEnterEvent event) {
if (!LoginView.class.equals(event.getNavigationTarget()) &&
!loginService.isAllowed(VaadinSession.getCurrent().getAttribute("user"),
"Professor")) {
event.rerouteTo(LoginView.class);
}
}
}
}

```


Κώδικας από αρχείο «LoginView.java» :

```
package net.uniwa.uniwaexamsproj.login;

import org.springframework.beans.factory.annotation.Autowired;
import com.vaadin.flow.component.ComponentEventListener;
import com.vaadin.flow.component.Tag;
import com.vaadin.flow.component.UI;
import com.vaadin.flow.component.dependency.JsModule;
import com.vaadin.flow.component.littemplate.LitTemplate;
import com.vaadin.flow.component.login.AbstractLogin;
import com.vaadin.flow.component.login.LoginForm;
import com.vaadin.flow.component.login.LoginI18n;
import com.vaadin.flow.component.orderedlayout.VerticalLayout;
import com.vaadin.flow.component.polymertemplate.Id;
import com.vaadin.flow.router.Route;
import com.vaadin.flow.server.VaadinSession;
import net.uniwa.uniwaexamsproj.professor.ProfessorRepository;
import net.uniwa.uniwaexamsproj.secretariat.SecretariatRepository;
import net.uniwa.uniwaexamsproj.student.StudentRepository;

//A Designer generated component for the login-view template..
//Designer will add and remove fields with @Id mappings but does not overwrite
or otherwise change this file..

//Some basic annotations to map the route of the class, to give a tag and to
associate it with the typescript file in the frontend package..
@Route("/")
@Tag("login-view")
@JsModule("./src/views/login-view.ts")
public class LoginView extends LitTemplate { //Inherit the components/methods
of LitTemplate for the View

//Declaration of a Layout, Login Service and some Repos..
@Id("layout")
VerticalLayout vlayout;

@Autowired
LoginService loginService;

@Autowired
StudentRepository stdRepo;

@Autowired
ProfessorRepository profRepo;

@Autowired
SecretariatRepository secrRepo;

//Create a new LoginView..

public LoginView() {

LoginI18n i18n = LoginI18n.createDefault(); //Use I18n for Greek Language..
i18n.getForm().setTitle("ΣΥΝΔΕΣΗ");
i18n.getForm().setPassword("ΚΩΔΙΚΟΣ");
i18n.getForm().setUsername("ΟΝΟΜΑ ΧΡΗΣΤΗ");
i18n.getForm().setSubmit("ΣΥΝΔΕΣΗ");
```

```

LoginForm loginForm = new LoginForm();
loginForm.setForgotPasswordButtonVisible(false); //Set Forgot Password Button
Invisible..
loginForm.setI18n(i18n);

vlayout.add(loginForm); //Add Login Form to the Layout..

//Here we Declare what happens next, after we press the Login Submit Button
and we send the Username/Password to the System..
loginForm.addLoginListener((ComponentEventListener<AbstractLogin.LoginEvent>
) loginEvent -> {

String username = loginEvent.getUsername(); //Read the Username/Password..
String password = loginEvent.getPassword();

if (username.isEmpty() || password.isEmpty()) { //Check if they exist..
loginForm.setError(true);
return;
}

int flag = loginService.GetAccess(username, password); //Send them to
GetAccess for Validation check..

if (flag > 0) { //Check the validation flag, save to a vaadin session with the
appropriate role and redirect to the correct main view page according to the
role..

switch (flag) {
case 1:
VaadinSession.getCurrent().setAttribute("user", stdRepo.getById(username));
UI.getCurrent().navigate("/student");
break;
case 2:
VaadinSession.getCurrent().setAttribute("user", profRepo.getById(username));
UI.getCurrent().navigate("/professor");
break;
case 3:
VaadinSession.getCurrent().setAttribute("user", secrRepo.getById(username));
UI.getCurrent().navigate("/secretariat");
break;
}
return;
}

loginForm.setError(true); //Or just set Error and Retry..
});
}

}

```

Κώδικας από αρχείο «login-view.ts» :

```

import { LitElement, html, css, customElement } from 'lit-element';
import '@vaadin/vaadin-ordered-layout/src/vaadin-vertical-layout.js';
import '@vaadin/vaadin-split-layout/src/vaadin-split-layout.js';

@customElement('login-view')
export class LoginView extends LitElement {

```

```

static get styles() {
return css`
:host {
display: block;
height: 100%;
}
`;
}

render() {
return html`
<vaadin-vertical-layout style="width: 100%; height: 100%;">
<vaadin-vertical-layout theme="spacing" style="align-self: center; margin:
var(--lumo-space-xl); padding: var(--lumo-space-xl); border: 1px solid black;
align-items: center;" id="layout">
<h2 style="align-self: center; margin: var(--lumo-space-
xl);color:blue;">UNIWA EXAM PLANNER</h2>
</vaadin-vertical-layout>
</vaadin-vertical-layout>
`;
}

// Remove this method to render the contents of this view inside Shadow DOM
createRenderRoot() {
return this;
} }

```

Κώδικας από αρχείο «secretariat-main-view.ts»:

```

import { LitElement, html, css, customElement } from 'lit-element';
import '@vaadin/vaadin-ordered-layout/src/vaadin-vertical-layout.js';
import '@vaadin/vaadin-button/src/vaadin-button.js';

@customElement('secretariat-main-view')
export class SecretariatMainView extends LitElement {
static get styles() {
return css`
:host {
display: block;
height: 100%;
}
`;
}

render() {
return html`
<vaadin-vertical-layout style="width: 100%; height: 100%; margin: var(--lumo-
space-xl); padding: var(--lumo-space-xl);">
<h2 style="align-self: center;color:blue;">ΛΕΙΤΟΥΡΓΙΕΣ ΓΡΑΜΜΑΤΕΙΑΣ</h2>
<vaadin-vertical-layout theme="spacing" style="align-self: center; border:
1px solid black; align-items: center;">
<vaadin-button theme="primary" style="margin: var(--lumo-space-xl);"
id="insertExam">
NEA ΕΞΕΤΑΣΤΙΚΗ
</vaadin-button>
<vaadin-button theme="primary" style="margin: var(--lumo-space-xl);"
id="updateExam">
ΕΝΗΜΕΡΩΣΗ ΕΞΕΤΑΣΤΙΚΗΣ
</vaadin-button>

```

```

<vaadin-button theme="primary" style="margin: var(--lumo-space-xl);"
id="allocateRoom">
ΕΝΗΜΕΡΩΣΗ ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ ΑΙΘΟΥΣΩΝ
</vaadin-button>
<vaadin-button theme="primary" style="margin: var(--lumo-space-xl);"
id="examInfo">
ΕΜΦΑΝΙΣΗ ΠΛΗΡΟΦΟΡΙΩΝ ΕΞΕΤΑΣΤΙΚΗΣ
</vaadin-button>
<vaadin-button theme="primary error" style="margin: var(--lumo-space-xl);
align-self: center;" id="exit">
ΕΞΟΔΟΣ
</vaadin-button>
</vaadin-vertical-layout>
</vaadin-vertical-layout>
`;
}

// Remove this method to render the contents of this view inside Shadow DOM
createRenderRoot() {
return this;
}
}

```

Κώδικας από αρχείο «create-exam.ts» :

```

import { LitElement, html, css, customElement } from 'lit-element';
import '@vaadin/vaadin-ordered-layout/src/vaadin-vertical-layout.js';
import '@vaadin/vaadin-ordered-layout/src/vaadin-horizontal-layout.js';
import '@vaadin/vaadin-button/src/vaadin-button.js';
import '@vaadin/vaadin-text-field';

@customElement('create-exam')
export class CreateExam extends LitElement {
static get styles() {
return css`
:host {
display: block;
height: 100%;
}
`;
}

render() {
return html`
<vaadin-vertical-layout style="width: 100%; height: 100%; align-items:
center;">
<h2 style="align-self: center; color:red; margin: var(--lumo-space-
xl);">ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΝΕΑΣ ΕΞΕΤΑΣΤΙΚΗΣ</h2>
<h3 style="align-self: center; color:green;">ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ
ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ</h3></br>
<vaadin-text-field id="year" label="ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ"></vaadin-text-field>
<vaadin-combo-box id="period" label="ΠΕΡΙΟΔΟΣ ΕΞΕΤΑΣΤΙΚΗΣ"></vaadin-combo-
box>
<vaadin-combo-box id="type" label="ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ"></vaadin-combo-box>
<vaadin-text-field id="start" label="ΗΜΕΡ/ΝΙΑ ΕΝΑΡΞΗΣ"></vaadin-text-field>
<vaadin-text-field id="end" label="ΗΜΕΡ/ΝΙΑ ΛΗΞΗΣ"></vaadin-text-field></br>
<vaadin-horizontal-layout theme="spacing" style="margin: var(--lumo-space-1);
align-self: center;">
<vaadin-button theme="primary" id="submit">

```

```

ΥΠΟΒΟΛΗ
</vaadin-button>
<vaadin-button theme="primary" id="clear">
ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ
</vaadin-button>
</vaadin-horizontal-layout>
<vaadin-horizontal-layout theme="spacing" style="margin: var(--lumo-space-1);
align-self: center;">
<vaadin-button theme="primary" id="examproflink">
ΕΙΣΑΓΩΓΗ ΜΑΘΗΜΑΤΩΝ ΕΞΕΤΑΣΤΙΚΗΣ
</vaadin-button>
<vaadin-button theme="primary error" id="exit">
ΕΞΟΔΟΣ
</vaadin-button>
<vaadin-button theme="primary" id="return">
ΕΠΙΣΤΡΟΦΗ
</vaadin-button>
</vaadin-horizontal-layout>
</vaadin-vertical-layout>
`;
}

// Remove this method to render the contents of this view inside Shadow DOM
createRenderRoot() {
return this;
}
}

```

Κώδικας από αρχείο «create-examprof.ts» :

```

import { LitElement, html, css, customElement } from 'lit-element';

@customElement('create-examprof')
export class CreateExamprof extends LitElement {
static get styles() {
return css`
:host {
display: block;
height: 100%;
}
`;
}

render() {
return html`
<vaadin-vertical-layout style="width: 100%; height: 100%; align-items:
center;">
<h2 style="align-self: center; color:red; margin: var(--lumo-space-
x1);">ΦΟΡΜΑ ΚΑΤΑΧΩΡΗΣΗΣ ΜΑΘΗΜΑΤΩΝ ΑΝΑ ΕΞΕΤΑΣΤΙΚΗ</h2>
<h3 style="align-self: center; color:green;">ΣΥΜΠΛΗΡΩΣΤΕ ΒΑΣΕΙ ΟΔΗΓΙΩΝ ΤΙΣ
ΑΠΑΡΑΙΤΗΤΕΣ ΤΙΜΕΣ</h3></br>
<vaadin-text-field id="year" label="ΕΤΟΣ ΕΞΕΤΑΣΤΙΚΗΣ"></vaadin-text-field>
<vaadin-combo-box id="type" label="ΚΑΤΗΓΟΡΙΑ ΕΞΕΤΑΣΤΙΚΗΣ"></vaadin-combo-box>
<vaadin-combo-box id="profname" label="ΕΠΩΝΥΜΟ ΔΙΔΑΣΚΟΝΤΑ"></vaadin-combo-
box>
<vaadin-combo-box id="lesstitle" label="ΤΙΤΛΟΣ ΜΑΘΗΜΑΤΟΣ"></vaadin-combo-
box>
<vaadin-horizontal-layout theme="spacing" style="margin: var(--lumo-space-1);
align-self: center;">

```

```

<vaadin-button theme="primary" id="submit">
ΥΠΟΒΟΛΗ
</vaadin-button>
<vaadin-button theme="primary" id="clear">
ΚΑΘΑΡΙΣΜΑ ΦΟΡΜΑΣ
</vaadin-button>
</vaadin-horizontal-layout>
<vaadin-horizontal-layout theme="spacing" style="margin: var(--lumo-space-1);
align-self: center;">
<vaadin-button theme="primary error" id="exit">
ΕΞΟΔΟΣ
</vaadin-button>
<vaadin-button theme="primary" id="return">
ΕΠΙΣΤΡΟΦΗ
</vaadin-button>
</vaadin-horizontal-layout>
</vaadin-vertical-layout>
`;
}

// Remove this method to render the contents of this view inside Shadow DOM
createRenderRoot() {
return this;
}
}

```

Κώδικας από αρχείο «allocate-exam-room-view.ts» :

```

import { LitElement, html, css, customElement } from 'lit-element';
import '@vaadin/vaadin-ordered-layout/src/vaadin-vertical-layout.js';
import '@vaadin/vaadin-ordered-layout/src/vaadin-horizontal-layout.js';
import '@vaadin/vaadin-button/src/vaadin-button.js';
import '@vaadin/vaadin-grid/src/vaadin-grid.js';

@customElement('allocate-exam-room-view')
export class AllocateExamRoomView extends LitElement {
static get styles() {
return css`
:host {
display: block;
height: 100%;
}
`;
}

render() {
return html`
<vaadin-vertical-layout style="width: 100%; height: 100%; align-items:
center;">
<h2 style="color:blue; align-self: center;">ΕΝΗΜΕΡΩΣΗ ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ
ΑΙΘΟΥΣΩΝ</h2>
<vaadin-horizontal-layout theme="spacing" style="width: 100%; align-self:
center; justify-content: center;">
<vaadin-vertical-layout theme="spacing" style="width: 40%;">
<h3 style="color:green">ΑΙΘΟΥΣΕΣ</h3>
<vaadin-grid id="classes" style="flex-grow: 1; flex-shrink: 0;"></vaadin-
grid>
</vaadin-vertical-layout>
<vaadin-vertical-layout theme="spacing" style="width: 50%;">

```

```

<h3 style="color:green">ΔΙΑΘΕΣΙΜΕΣ ΧΡΟΝΟΘΥΡΙΑΔΕΣ</h3>
<vaadin-grid id="timeslots" dir="timeslots"></vaadin-grid>
</vaadin-vertical-layout>
</vaadin-horizontal-layout>
<vaadin-vertical-layout theme="spacing" style="width: 40%;">
<h3 style="color:green">ΧΡΟΝΟΘΥΡΙΑΔΕΣ</h3>
<vaadin-grid id="allocated" style="height:200px;flex-grow: 1; flex-shrink:
0;"></vaadin-grid>
</vaadin-vertical-layout>
<vaadin-horizontal-layout theme="spacing" style="align-self: center;">
<vaadin-button theme="primary error" id="exit">
ΕΞΟΔΟΣ
</vaadin-button>
<vaadin-button theme="primary" id="return">
ΕΠΙΣΤΡΟΦΗ
</vaadin-button>
</vaadin-horizontal-layout>
</vaadin-vertical-layout>
`;
}

// Remove this method to render the contents of this view inside Shadow DOM
createRenderRoot() {
return this;
}
}

```

Κώδικας από αρχείο «update-exam-info-view.ts» :

```

import { LitElement, html, css, customElement } from 'lit-element';
import '@vaadin/vaadin-ordered-layout/src/vaadin-horizontal-layout.js';
import '@vaadin/vaadin-grid/src/vaadin-grid.js';
import '@vaadin/vaadin-ordered-layout/src/vaadin-vertical-layout.js';
import '@vaadin/vaadin-button/src/vaadin-button.js';

@customElement('update-exam-info-view')
export class UpdateExamInfoView extends LitElement {
static get styles() {
return css`
:host {
display: block;
height: 100%;
}
`;
}

render() {
return html`
<vaadin-vertical-layout style="width: 100%; height: 100%; align-items:
center;">
<h2 style="align-self: center; color:blue; margin: var(--lumo-space-
xl);">ΕΝΗΜΕΡΩΣΗ ΕΞΕΤΑΣΤΙΚΗΣ</h2>
<vaadin-horizontal-layout theme="spacing" style="align-self: center; justify-
content: center;">
<vaadin-vertical-layout style="align:center;">
<h3 style="align-self: flex-start;color:green;">ΕΞΕΤΑΣΕΙΣ</h3>
<vaadin-grid id="exams" style="width: 1800px;"></vaadin-grid>
</vaadin-vertical-layout>
</vaadin-horizontal-layout>

```

```

<vaadin-horizontal-layout theme="spacing" style="margin: var(--lumo-space-1);
align-self: center;">
<vaadin-button theme="primary error" id="exit">
ΕΞΟΔΟΣ
</vaadin-button>
<vaadin-button theme="primary" id="return">
ΕΠΙΣΤΡΟΦΗ
</vaadin-button>
</vaadin-horizontal-layout>
</vaadin-vertical-layout>
`;
}

// Remove this method to render the contents of this view inside Shadow DOM
createRenderRoot() {
return this;
}
}

```

Κώδικας από αρχείο «exam-info-view.ts» :

```

import { LitElement, html, css, customElement } from 'lit-element';
import '@vaadin/vaadin-grid/src/vaadin-grid.js';
import '@vaadin/vaadin-ordered-layout/src/vaadin-vertical-layout.js';
import '@vaadin/vaadin-ordered-layout/src/vaadin-horizontal-layout.js';
import '@vaadin/vaadin-split-layout/src/vaadin-split-layout.js';
import '@vaadin/vaadin-button/src/vaadin-button.js';

@customElement('exam-info-view')
export class ExamInfoView extends LitElement {
static get styles() {
return css`
:host {
display: block;
height: 100%;
}
`;
}

render() {
return html`
<vaadin-vertical-layout style="width: 100%; height: 100%; align-items:
center;">
<h2 style="align-self: center;color:blue;">ΕΜΦΑΝΙΣΗ ΠΛΗΡΟΦΟΡΙΩΝ
ΕΞΕΤΑΣΤΙΚΗΣ</h2>
<vaadin-horizontal-layout theme="spacing">
<vaadin-vertical-layout theme="spacing">
<h3 style="align-self: flex-start;color:green;">ΕΞΕΤΑΣΕΙΣ:</h3>
<vaadin-grid id="exams" style="width: 1100px;"></vaadin-grid>
</vaadin-vertical-layout>
<vaadin-vertical-layout theme="spacing">
<h3 style="color:green">ΑΙΘΟΥΣΕΣ:</h3>
<vaadin-grid id="classes" style="width: 500px;"></vaadin-grid>
</vaadin-vertical-layout>
</vaadin-horizontal-layout>
<vaadin-split-layout style="align-self: center;">
<vaadin-button theme="primary error" id="exit">
ΕΞΟΔΟΣ
</vaadin-button>

```



```

<vaadin-button theme="primary" id="return">
ΕΠΙΣΤΡΟΦΗ
</vaadin-button>
</vaadin-split-layout>
</vaadin-vertical-layout>
`;
}

// Remove this method to render the contents of this view inside Shadow DOM
createRenderRoot() {
return this;
}
}

```

Κώδικας από αρχείο «student-main-view.ts» :

```

import { LitElement, html, css, customElement } from 'lit-element';
import '@vaadin/vaadin-ordered-layout/src/vaadin-vertical-layout.js';
import '@vaadin/vaadin-button/src/vaadin-button.js';

@customElement('student-main-view')
export class SecretariatMainView extends LitElement {
static get styles() {
return css`
:host {
display: block;
height: 100%;
}
`;
}

render() {
return html`
<vaadin-vertical-layout style="width: 100%; height: 100%;">
<h2 style="color:blue; align-self: center;">ΛΕΙΤΟΥΡΓΙΕΣ ΦΟΙΤΗΤΗ</h2>
<vaadin-vertical-layout theme="spacing-xl" style="align-self: center; flex-
wrap: wrap; margin: var(--lumo-space-xl); padding: var(--lumo-space-xl);
border: 1px solid black; align-items: center; width: 400px;">
<h3 id="studentId" style="color:green; align-self: flex-start;">ID
ΜΑΘΗΤΗ:</h3>
<vaadin-button theme="primary" id="apply">
ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ
</vaadin-button>
<vaadin-button theme="primary" id="display">
ΠΡΟΒΟΛΗ ΑΙΤΗΣΕΩΝ
</vaadin-button>
<vaadin-button theme="primary error" id="exit">
ΕΞΟΔΟΣ
</vaadin-button>
</vaadin-vertical-layout>
</vaadin-vertical-layout>
`;
}

// Remove this method to render the contents of this view inside Shadow DOM
createRenderRoot() {
return this;
}
}

```

Κώδικας από αρχείο «apply-view.ts» :

```
import { LitElement, html, css, customElement } from 'lit-element';
import '@vaadin/vaadin-ordered-layout/src/vaadin-vertical-layout.js';
import '@vaadin/vaadin-ordered-layout/src/vaadin-horizontal-layout.js';
import '@vaadin/vaadin-grid/src/vaadin-grid.js';
import '@vaadin/vaadin-button/src/vaadin-button.js';

@customElement('apply-view')
export class ApplyView extends LitElement {
  static get styles() {
    return css`
      :host {
        display: block;
        height: 100%;
      }
    `;
  }

  render() {
    return html`
      <vaadin-vertical-layout style="align-items: center;">
        <h2 style="color:blue;">ΔΗΛΩΣΗ ΣΥΜΜΕΤΟΧΗΣ ΣΕ ΕΞΕΤΑΣΕΙΣ</h2>
        <vaadin-horizontal-layout theme="spacing">
          <vaadin-vertical-layout theme="spacing-xl" id="layout1">
            <h3 style="color:green; align-self: flex-start; padding: var(--lumo-space-m);">ΜΑΘΗΜΑΤΑ:</h3>
            <vaadin-grid id="lessons" style="width: 900px;"></vaadin-grid>
          </vaadin-vertical-layout>
        </vaadin-horizontal-layout>
        <vaadin-horizontal-layout theme="spacing-xl" style="margin: var(--lumo-space-xl); padding: var(--lumo-space-xl);">
          <vaadin-button theme="primary error" id="exit">
            ΕΞΟΔΟΣ
          </vaadin-button>
          <vaadin-button theme="primary" id="return">
            ΕΠΙΣΤΡΟΦΗ
          </vaadin-button>
        </vaadin-horizontal-layout>
      </vaadin-vertical-layout>
    `;
  }

  // Remove this method to render the contents of this view inside Shadow DOM
  createRenderRoot() {
    return this;
  }
}
```

Κώδικας από αρχείο «student-applications-view.ts» :

```
import { LitElement, html, css, customElement } from 'lit-element';
import '@vaadin/vaadin-ordered-layout/src/vaadin-vertical-layout.js';
import '@vaadin/vaadin-grid/src/vaadin-grid.js';
import '@vaadin/vaadin-ordered-layout/src/vaadin-horizontal-layout.js';
import '@vaadin/vaadin-button/src/vaadin-button.js';
```

```

@customElement('student-applications-view')
export class StudentApplicationsView extends LitElement {
  static get styles() {
    return css`
      :host {
        display: block;
        height: 100%;
      }
    `;
  }

  render() {
    return html`
      <vaadin-vertical-layout style="width: 100%; height: 700px; align-items:
      center;">
        <h2 style="color:blue">ΠΡΟΒΟΛΗ ΑΙΤΗΣΕΩΝ</h2>
        <vaadin-grid id="applications" style="width: 1550px; align-self: center;
        height: 300px;" dir="applications"></vaadin-grid>
        <vaadin-horizontal-layout theme="spacing">
          <vaadin-button theme="primary error" id="exit">
            ΕΞΟΔΟΣ
          </vaadin-button>
          <vaadin-button theme="primary" id="return">
            ΕΠΙΣΤΡΟΦΗ
          </vaadin-button>
        </vaadin-horizontal-layout>
      </vaadin-vertical-layout>
    `;
  }

  // Remove this method to render the contents of this view inside Shadow DOM
  createRenderRoot() {
    return this;
  }
}

```

Κώδικας από αρχείο «professor-main-view.ts» :

```

import { LitElement, html, css, customElement } from 'lit-element';
import '@vaadin/vaadin-ordered-layout/src/vaadin-vertical-layout.js';
import '@vaadin/vaadin-button/src/vaadin-button.js';

@customElement('professor-main-view')
export class ProfessorMainView extends LitElement {
  static get styles() {
    return css`
      :host {
        display: block;
        height: 100%;
      }
    `;
  }

  render() {
    return html`
      <vaadin-vertical-layout style="width: 100%; height: 100%; align-items:
      center;">
    `;
  }
}

```

```

<h2 style="color:blue; margin: var(--lumo-space-xl);">ΛΕΙΤΟΥΡΓΙΕΣ
ΚΑΘΗΓΗΤΗ</h2>
<vaadin-vertical-layout theme="spacing-xl" style="align-items: center;
margin: var(--lumo-space-xl); align-self: center; border: 1px solid black;
padding: var(--lumo-space-xl);">
<vaadin-button theme="primary" id="applications">
ΕΜΦΑΝΙΣΗ ΑΙΤΗΣΕΩΝ ΦΟΙΤΗΤΩΝ
</vaadin-button>
<vaadin-button theme="primary error" id="exit">
ΕΞΟΔΟΣ
</vaadin-button>
</vaadin-vertical-layout>
</vaadin-vertical-layout>
`;
}

// Remove this method to render the contents of this view inside Shadow DOM
createRenderRoot() {
return this;
}
}

```

Κώδικας από αρχείο «professor-applications-view.ts» :

```

import { LitElement, html, css, customElement } from 'lit-element';
import '@vaadin/vaadin-button/src/vaadin-button.js';
import '@vaadin/vaadin-ordered-layout/src/vaadin-vertical-layout.js';
import '@vaadin/vaadin-ordered-layout/src/vaadin-horizontal-layout.js';
import '@vaadin/vaadin-grid/src/vaadin-grid.js';

@customElement('professor-applications-view')
export class ProfessorApplicationsView extends LitElement {
static get styles() {
return css`
:host {
display: block;
height: 100%;
}
`;
}

render() {
return html`
<vaadin-vertical-layout style="width: 100%; height: 100%; align-items:
center;">
<h2 style="color:blue; align-self: center; margin: var(--lumo-space-
1);">ΔΙΑΘΕΣΙΜΕΣ ΑΙΤΗΣΕΙΣ ΦΟΙΤΗΤΩΝ</h2>
<vaadin-horizontal-layout theme="spacing">
<vaadin-vertical-layout theme="spacing" style="width: 100%;">
<h3 style="color:green; align-self: flex-start;">ΕΞΕΤΑΣΕΙΣ</h3>
<vaadin-grid id="exams" style="width: 1050px;"></vaadin-grid>
</vaadin-vertical-layout>
<vaadin-vertical-layout theme="spacing" style="width: 100%;">
<h3 style="color:green;">ΦΟΙΤΗΤΕΣ</h3>
<vaadin-grid id="students" style="width: 600px;"></vaadin-grid>
</vaadin-vertical-layout>
</vaadin-horizontal-layout>
<vaadin-button theme="primary error" style="align-self: center;" id="exit">
ΕΞΟΔΟΣ

```

```

</vaadin-button>
</vaadin-vertical-layout>
`;
}

// Remove this method to render the contents of this view inside Shadow DOM
createRenderRoot() {
return this;
}
}

```

Κώδικας από αρχείο «application.properties» :

```

#server runs on port 8080..
server.port = 8080

#h2 console is enabled to test live the changes on the database..
spring.h2.console.enabled=true

#h2 console is available at http://localhost:8090/console..
spring.h2.console.path=/console

#whitelabel pages are disabled and we have the normal 3 digit error html page
back..
server.error.whitelabel.enabled=false

#this is the url of our database..
spring.datasource.url = jdbc:mysql://localhost:3306/examsdb

#these are the credentials to login into our database as admin..
spring.datasource.username = root
spring.datasource.password = rootP

#the dialect to communicate with our database through jpa hibernate..
spring.jpa.properties.hibernate.dialect =
org.hibernate.dialect.MySQL5InnoDBDialect

#we use this due to the case sensitivity and in order to run correct in any
case our CRUD queries..
spring.jpa.hibernate.naming.physical-
strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl

spring.jpa.generate-ddl=true

spring.jpa.hibernate.ddl-auto=update

logging.level.org.hibernate.SQL=DEBUG
logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE

```

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Atlassian. (2022, 11 29). *confluence.atlassian.com*. Ανάκτηση από *atlassian.com*:
https://confluence.atlassian.com/doc/setting-the-java_home-variable-in-windows-8895.html
- Guruge, C. D. (2019, 10 2). *medium.com*. Ανάκτηση από *medium.com*:
<https://medium.com/@chamathdilanka.official/creating-windows-service-to-run-an-executable-jar-822924b28f5e>
- MySQL. (2023, 1 1). *dev.mysql.com*. Ανάκτηση από *mysql.com*:
<https://dev.mysql.com/downloads/installer/>
- Oracle. (2023, 1 1). *oracle.com*. Ανάκτηση από *oracle.com*:
https://download.oracle.com/java/18/archive/jdk-18_windows-x64_bin.exe
- Wikipedia. (2022, 8 12). *en.wikipedia.org/wiki/ADONIS_(software)*. Ανάκτηση από *ADONIS_(software)*: [en.wikipedia.org/wiki/ADONIS_\(software\)](https://en.wikipedia.org/wiki/ADONIS_(software))
- Wikipedia. (2022, 2 10). <https://el.wikipedia.org/wiki/Java>. Ανάκτηση από el.wikipedia.org/wiki/Java: <https://el.wikipedia.org/wiki/Java>
- Wikipedia. (2023, 6 25). <https://en.wikipedia.org/wiki/MySQL>. Ανάκτηση από en.wikipedia.org/wiki/MySQL: <https://en.wikipedia.org/wiki/MySQL>
- Wikipedia. (2023, 5 16). https://en.wikipedia.org/wiki/MySQL_Workbench. Ανάκτηση από [MySQL_Workbench](https://en.wikipedia.org/wiki/MySQL_Workbench): https://en.wikipedia.org/wiki/MySQL_Workbench
- Wikipedia. (2023, 7 2). https://en.wikipedia.org/wiki/Spring_Boot. Ανάκτηση από [Spring Boot](https://en.wikipedia.org/wiki/Spring_Boot): https://en.wikipedia.org/wiki/Spring_Boot
- Wikipedia. (2023, 7 2). https://en.wikipedia.org/wiki/Spring_Framework. Ανάκτηση από [Spring Framework](https://en.wikipedia.org/wiki/Spring_Framework): https://en.wikipedia.org/wiki/Spring_Framework
- Wikipedia. (2023, 6 10). <https://en.wikipedia.org/wiki/Vaadin>. Ανάκτηση από [Vaadin](https://en.wikipedia.org/wiki/Vaadin): <https://en.wikipedia.org/wiki/Vaadin>