



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

Τίτλος Πτυχιακής Εργασίας

ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΑΡΟΥΣΙΩΝ – ΗΛΕΚΤΡΟΝΙΚΟ ΠΑΡΟΥΣΙΟΛΟΓΙΟ

Ονομα φοιτητή

ΜΑΚΡΗΣ ΙΩΑΝΝΗΣ

Αθήνα, 2022



HAROKOPIO UNIVERSITY
SCHOOL OF DIGITAL TECHNOLOGY
DEPARTMENT INFORMATICS AND TELEMATICS

Thesis Title

PRESENCE MANAGEMENT SYSTEM

Student's Name

MAKRIS IOANNIS

Athens, 2022



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

Τριμελής Εξεταστική Επιτροπή

ΕΠΙΒΛΕΠΩΝ: ΤΣΑΔΗΜΑΣ ΑΝΑΡΓΥΡΟΣ

Ε.ΔΙ.Π., Τμήμα Πληροφορικής & Τηλεματικής,
Χαροκόπειο Πανεπιστήμιο

ΤΣΕΡΠΕΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

Αναπληρωτής Καθηγητής, Τμήμα Πληροφορικής & Τηλεματικής,
Χαροκόπειο Πανεπιστήμιο

ΚΑΜΑΛΑΚΗΣ ΘΩΜΑΣ

Καθηγητής, Τμήμα Πληροφορικής & Τηλεματικής,
Χαροκόπειο Πανεπιστήμιο

Ο ΜΑΚΡΗΣ ΙΩΑΝΝΗΣ

δηλώνω υπεύθυνα ότι:

- 1) Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλλει τα πνευματικά δικαιώματα τρίτων.
- 2) Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.
- 3) Οπου υφίστανται δικαιώματα άλλων δημιουργών έχουν διασφαλιστεί όλες οι αναγκαίες άδειες χρήσης ενώ το αντίστοιχο υλικό είναι ευδιάκριτο στην υποβληθείσα εργασία.

ΑΦΙΕΡΩΣΗ

Η παρούσα πτυχιακή εργασία είναι αφιερωμένη στην οικογένειά μου, σε όλους όσους με στήριξαν καθόλη την διάρκεια περάτωσής της και ιδιαιτέρως στην γιαγιά μου, Άννα Συρίγου που πάντα με στήριζε με όλες τις δυνάμεις που διέθετε και με βοήθαγε να είμαι προσηλωμένος στους στόχους και τις προτεραιότητές μου.

Αγαπητή μου γιαγιά σε ευχαριστώ για όλα όσα μου προσέφερες. Η δυναμική σου προσωπικότητα, ο αkéραιος χαρακτήρας σου και τα όσα κατάφερες με πολύ αγώνα και αυτοθυσία στην ζωή σου, αποτελεί και πάντοτε θα αποτελεί για μένα, παράδειγμα προς μίμηση.

Ἄρχὴ σοφίας ἢ τῶν ὀνομάτων ἐπίσκεψις.

ΑΝΤΙΣΘΕΝΗΣ Ο ΚΥΝΙΚΟΣ

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον κ. Τσαδήμα Ανάργυρο, επιβλέποντα καθηγητή της πτυχιακής μου εργασίας, για όλη την υποστήριξη που μου προσέφερε καθόλη την διάρκεια της περάτωσής της. Κατα την εκπόνηση της πτυχιακής μου εργασίας, ο κ. Τσαδήμας Ανάργυρος μου έδωσε χρήσιμες ιδέες και συμβουλές και ήταν πάντα πρόθυμος να με κατατοπίσει σε όποια δυσκολία προέκυψε.

Ο κ. Τσαδήμας Ανάργυρος με βοήθησε με τον ίδιο τρόπο και καθόλη την διάρκεια των σπουδών μου, περισσότερο όμως με ενέπνευσε ως άνθρωπος που επιμένει, εργάζεται σκληρά και πετυχαίνει τους στόχους του κάνοντας αυτό που αγαπά με μεράκι και αφοσίωση.

Στο ίδιο μήκος κύματος θα ήθελα να αποδώσω ευχαριστίες και στον κ. Τσερπέ Κωνσταντίνο που στην διάρκεια των σπουδών μου, μου έδωσε το κίνητρο να ασχοληθώ με τον προγραμματισμό και μου ενέπνευσε την αρετή της πειθαρχίας. Ευχαριστώ επίσης τον κ. Καμαλάκη Θωμά που απο την πρώτη ημέρα των σπουδών μου, με έκανε να αγαπήσω το Χαροκόπειο Πανεπιστήμιο, να κατανοήσω την αξία της πληροφορικής και να προσπαθώ να σκέφτομαι “έξω απο το κουτί”. Θα ήθελα να αποδώσω ευχαριστίες και στον κ. Βαρλάμη Ηρακλή, τον κ. Μιχαήλ Δημήτριο, τον κ. Μιχαλακέλη Χρήστο καθώς και τους υπόλοιπους καθηγητές για την στήριξη και τις συμβουλές τους καθόσο ήμουν φοιτητής.

Επιπροσθέτως ευχαριστώ πολύ την οικογένειά μου για την κατανόησή της κατα την περάτωση της πτυχιακής μου εργασίας αλλά και το κίνητρο που μου έδωσε να ασχοληθώ με αυτό που αγαπώ με συνέπεια και αφοσίωση. Απο την οικογένειά μου, ευχαριστώ ιδιαιτέρως την γιαγιά μου Άννα που είχε πάντα έννοια την περάτωση των σπουδών μου, με βοήθησε σταθερά και έμπρακτα με οποιοδήποτε μέσο και τρόπο μπόρεσε και αποτέλεσε για μένα ως προσωπικότητα, παράδειγμα προς μίμηση.

Τέλος ευχαριστώ τους φίλους συμφοιτητές μου και συνολικά το Χαροκόπειο Πανεπιστήμιο, που μέσα απο το πνεύμα της συνεργασίας, με στήριξαν στα δύσκολα και μου έδωσαν πρόσβαση σε πολύτιμες γνώσεις γύρω απο το αντικείμενό μου καθώς και εμπειρίες ζωής.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Περίληψη στα Ελληνικά.....	10
Abstract.....	11
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ.....	12
ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ.....	15
ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ.....	16
Κεφάλαιο 1: Εισαγωγή.....	17
1.1. Εισαγωγή.....	17
1.2. Στόχος Εφαρμογής.....	18
Κεφάλαιο 2: Ανάλυση του Συστήματος.....	20
2.1. Εισαγωγή.....	20
2.2. Ανάλυση Απαιτήσεων του Συστήματος.....	21
2.3. Ανάλυση και Σχεδίαση Συστήματος.....	25
2.3.1. Use Case Diagram.....	27
2.3.2. Class Diagram.....	37
2.3.3. Activity Diagrams.....	51
2.3.4. Sequence Diagrams.....	73
2.3.5. State Machine Diagrams.....	85
2.3.6. Entity Relationship Diagram.....	88
2.3.7. Λειτουργικές και Μη Λειτουργικές Απαιτήσεις.....	89
Κεφάλαιο 3: Τεχνολογίες Υλοποίησης της Εφαρμογής.....	92
3.1. Τεχνολογίες Server.....	92
3.1.1. Spring Framework - Spring Boot.....	92
3.1.2. Project Lombok.....	97
3.1.3. Apache Commons Csv.....	98
3.1.4. Apache POI-XSSF/SXSSF.....	100
3.1.5. Java Bean Validation Framework.....	102
3.1.6. JPA - Spring Data JPA.....	103

3.1.7. Java Persistence Query Language.....	109
3.1.8. Hibernate - ORM.....	111
3.1.9. MySQL.....	117
3.1.10. Spring Security.....	120
3.1.11. Json Web Token.....	123
3.2. Τεχνολογίες Client.....	131
3.2.1. Angular Framework.....	131
3.2.2. RxJS.....	143
3.2.3. Angular Material.....	148
3.2.4. Timepicker.....	158
3.2.5. Angular Flex-Layout.....	159
3.2.6. Moment.js.....	162
Κεφάλαιο 4: Οδηγίες Εγκατάστασης της Εφαρμογής.....	163
Κεφάλαιο 5: Σενάρια Χρήσης.....	165
5.1. Γενικές Λειτουργίες.....	165
5.2. Σύνδεση και Εγγραφή στην Εφαρμογή (Login, Register Page).....	168
5.3. Λειτουργίες Διαχειριστή (Admin).....	171
5.4. Λειτουργίες Καθηγητή (Teacher).....	206
5.5. Λειτουργίες Φοιτητή (Student).....	242
5.6. Λειτουργίες Γραμματέα (Secretary).....	258
Κεφάλαιο 6: Σύνοψη.....	263
6.1. Βασικά Συμπεράσματα.....	263
6.2. Προοπτικές Εξέλιξης του Συστήματος.....	264
Κεφάλαιο 7: Επίλογος.....	266
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	267
ΑΠΟΘΕΤΗΡΙΑ ΚΩΔΙΚΑ.....	270

Περίληψη στα Ελληνικά

Στον σύγχρονο κόσμο, η εξέλιξη του διαδικτύου και των νέων τεχνολογιών, βελτίωσε και εξομάλυνε τις γραφειοκρατικές διαδικασίες των πανεπιστημίων. Στον αντίποδα όμως συνεχίζουν να επιβιώνουν διαδικασίες απαρχαιωμένες και χρονοβόρες, γεγονός που δυσχεραίνει την απαιτητική ακαδημαϊκή καθημερινότητα. Κρίνεται λοιπόν αναγκαία η αυτοματοποίηση και ψηφιοποίηση τους.

Σκοπός αυτής της εργασίας είναι η αυτοματοποίηση της διαδικασίας διαχείρισης των παρουσιών των φοιτητών στις παραδόσεις των διαλέξεων, μέσω της δημιουργίας μιας διαδικτυακής εφαρμογής, για το Χαροκόπειο Πανεπιστήμιο.

Η εφαρμογή αναπτύχθηκε με την τεχνολογία Spring Boot στο επίπεδο του backend, ενώ για το frontend χρησιμοποιήθηκε το Angular Framework. Η πρόσβαση των εμπλεκόμενων χρηστών στα δεδομένα και τις ενέργειες που επιτελούν στην εφαρμογή, δίνεται με βάση συγκεκριμένα κριτήρια όπου κρίνεται απαραίτητο καθώς και με τα αντίστοιχα δικαιώματα τα οποία είναι ανάλογα με τα καθήκοντα που επιμελούνται.

Αρχικά γίνεται μία βασική περιγραφή του προβλήματος που καλείται να επιλύσει το σύστημα. Επειτα αναφέρονται οι βασικές λειτουργίες του συστήματος, οι ανάγκες που ικανοποιούν συνολικά καθώς και ο κύριος στόχος της εφαρμογής. Ακολούθως αναλύεται η βασική δομή του συστήματος και περιγράφονται οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίησή του. Παράλληλα, παρέχονται οδηγίες εγκατάστασης αλλά και σενάρια χρήσης της εφαρμογής ανα τύπο χρήστη. Τέλος παρατίθενται συμπεράσματα και προοπτικές εξέλιξης για το σύστημα.

Λέξεις κλειδιά: [Angular Framework, Spring Boot, σύστημα διαχείρισης παρουσιών]

Abstract

In the modern world, the development of the internet and new technologies has improved and simplified the bureaucratic procedures of universities. On the other hand, however, outdated and time-consuming procedures continue to survive which makes the demanding academic daily life more difficult. It is therefore necessary to automate and digitise them.

The aim of this project is to automate the process of managing the attendance of students at lectures, through the creation of a web application for the Harokopio University.

The application was developed with Spring Boot technology at the backend level, while the Angular Framework was used for the frontend. The access of the involved users to the data and the actions they perform in the application, is given based on specific criteria where it is necessary and with the corresponding rights, which are according to the tasks they are in charge of.

At first, a basic description of the problem the system is asked to solve is given. Then the main functions of the system, the needs they satisfy overall and the main purpose of the application are stated. Subsequently, the main structure of the system is analysed and the technologies used for its implementation are described. At the same time, installation instructions and scenarios for the use of the application are provided for each type of user. Finally, conclusions and prospects for the system's development are given.

Keywords: [Angular Framework, Spring Boot, presence management system]

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικ.1. Μορφή του Json Web Token [28].....	σ.124
Εικ.2. Json Web Token Header.....	σ.125
Εικ.3. Json Web Token Payload.....	σ.126
Εικ.4. Json Web Token Signature.....	σ.127
Εικ.5. Κωδικοποιημένο Json Web Token.....	σ.127
Εικ.6. Json Web Token – Κωδικοποιημένο και Αποκωδικοποιημένο.....	σ.128
Εικ.7. Μορφή Authorization Header [28].....	σ.129
Εικ.8. RxJS Downloads [42].....	σ.145
Εικ.9. Classes Sessions Material Features – Datepicker.....	σ.151
Εικ.10. Courses Schedules Material Features - Επιλογή Μαθήματος.....	σ.152
Εικ.11. Courses Schedules Material Features - Επιλογή καθηγητών.....	σ.153
Εικ.12. Classes Sessions Material Features - Επιλογή Διάλεξης.....	σ.154
Εικ.13. Classes Sessions Material Features – Λίστα.....	σ.154
Εικ.14. Material Features - Success Dialog.....	σ.156
Εικ.15. Material Features - Failure Dialog.....	σ.157
Εικ.16. Timepicker.....	σ.158
Εικ.17. Home Page.....	σ.166
Εικ.18. Account Details.....	σ.167
Εικ.19. Change Password.....	σ.167
Εικ.20. Φόρμα Login.....	σ.168
Εικ.21. Login - Bad Credentials.....	σ.169
Εικ.22. Login - Inactive Account.....	σ.170
Εικ.23. Επιτυχής Σύνδεση Χρήστη.....	σ.170
Εικ.24. Φόρμα Εγγραφής Καθηγητή.....	σ.171
Εικ.25. Μενού Διαχειριστή Συστήματος.....	σ.172
Εικ.26. Δημιουργία Σχολής.....	σ.173
Εικ.27. Προβολή στοιχείων σχολής.....	σ.174
Εικ.28. Επεξεργασία στοιχείων σχολής.....	σ.175
Εικ.29. Διαγραφή Σχολής.....	σ.176
Εικ.30. Αδυναμία Διαγραφής Σχολής.....	σ.177
Εικ.31. Δημιουργία τμήματος.....	σ.178
Εικ.32. Προβολή στοιχείων τμήματος.....	σ.179
Εικ.33. Επεξεργασία στοιχείων τμήματος.....	σ.180
Εικ.34. Διαγραφή τμήματος.....	σ.181
Εικ.35. Αδυναμία Διαγραφής Τμήματος.....	σ.182
Εικ.36. Δημιουργία μαθήματος.....	σ.183
Εικ.37. Προβολή στοιχείων μαθήματος.....	σ.184
Εικ.38. Επεξεργασία στοιχείων μαθήματος.....	σ.185
Εικ.39. Διαγραφή μαθήματος.....	σ.186
Εικ.40. Αδυναμία διαγραφής μαθήματος.....	σ.187

Εικ.41. Δημιουργία προγράμματος μαθήματος.....	σ.188
Εικ.42. Προβολή στοιχείων προγράμματος μαθήματος.....	σ.189
Εικ.43. Φοιτητές προγράμματος μαθήματος.....	σ.190
Εικ.44. Επεξεργασία στοιχείων προγράμματος μαθήματος.....	σ.191
Εικ.45. Αδυναμία επεξεργασίας προγράμματος μαθήματος.....	σ.192
Εικ.46. Διαγραφή προγράμματος μαθήματος.....	σ.193
Εικ.47. Αδυναμία διαγραφής προγράμματος μαθήματος.....	σ.194
Εικ.48. Δημιουργία αίτησης δικαιολόγησης απουσίας για έναν φοιτητή.....	σ.195
Εικ.49. Δημιουργία χρηστών.....	σ.197
Εικ.50. Ανέβασμα αρχείου φοιτητών.....	σ.198
Εικ.51. Λάθος τύπος αρχείου csv/excel.....	σ.199
Εικ.52. Ακατάλληλο περιεχόμενο αρχείου csv/excel.....	σ.200
Εικ.53. Ενημέρωση για τα σημεία των σφαλμάτων του αρχείου excel.....	σ.201
Εικ.54. Ενημέρωση διαχειριστή για το πρώτο σφάλμα του αρχείου csv/excel.....	σ.202
Εικ.55. Ενημέρωση διαχειριστή για ήδη υπάρχουσα εγγραφή του αρχείου csv/excel.....	σ.203
Εικ.56. Προβολή στοιχείων χρήστη.....	σ.204
Εικ.57. Επεξεργασία στοιχείων χρήστη.....	σ.205
Εικ.58. Φόρμα επεξεργασίας φοιτητή.....	σ.206
Εικ.59. Μενού Καθηγητή.....	σ.207
Εικ.60. Δημιουργία διάλεξης.....	σ.208
Εικ.61. Προβολή στοιχείων διάλεξης.....	σ.209
Εικ.62. Επεξεργασία στοιχείων διάλεξης.....	σ.210
Εικ.63. Αδυναμία επεξεργασίας στοιχείων διάλεξης.....	σ.211
Εικ.64. Διαγραφή διάλεξης.....	σ.212
Εικ.65. Αδυναμία διαγραφής διάλεξης.....	σ.213
Εικ.66. Δημιουργία ομάδας για φοιτητές.....	σ.213
Εικ.67. Προβολή στοιχείων ομάδας για φοιτητές.....	σ.214
Εικ.68. Επεξεργασία στοιχείων ομάδας για φοιτητές.....	σ.215
Εικ.69. Αδυναμία επεξεργασίας ομάδας φοιτητών.....	σ.216
Εικ.70. Διαγραφή ομάδας για φοιτητές.....	σ.216
Εικ.71. Αδυναμία διαγραφής ομάδας σχετιζόμενης με συνεδρία διάλεξης.....	σ.217
Εικ.72. Αδυναμία διαγραφής ομάδας με εγγεγραμμένους φοιτητές.....	σ.218
Εικ.73. Προσθήκη φοιτητή στην ομάδα.....	σ.219
Εικ.74. Αδυναμία προσθήκης φοιτητή σε ομάδα που σχετίζεται με συνεδρία διάλεξης.....	σ.220
Εικ.75. Αδυναμία προσθήκης φοιτητή σε γεμάτη ομάδα.....	σ.221
Εικ.76. Προβολή στοιχείων εγγεγραμμένου φοιτητή.....	σ.222
Εικ.77. Αφαίρεση φοιτητή από την ομάδα.....	σ.223
Εικ.78. Αδυναμία αφαίρεσης φοιτητή από ομάδα σχετιζόμενη με συνεδρία διάλεξης.....	σ.224
Εικ.79. Προσθήκη συνεδρίας διάλεξης.....	σ.224
Εικ.80. Αδυναμία δημιουργίας ήδη υπάρχουσας συνεδρίας διάλεξης.....	σ.225
Εικ.81. Ανεπαρκής ομάδα φοιτητών.....	σ.226
Εικ.82. Πεπερασμένη ημερομηνία και ώρα.....	σ.227
Εικ.83. Μη διαθέσιμη ημερομηνία, ώρα ή αίθουσα.....	σ.228
Εικ.84. Ασύμβατη με το εξάμηνο τιμή του μήνα στην ημερομηνία.....	σ.229
Εικ.85. Ο καθηγητής μετέχει σε άλλη συνεδρία διάλεξης που εξελίσσεται ταυτόχρονα.....	σ.230

Εικ.86. Προβολή στοιχείων συνεδρίας διάλεξης.....	σ.231
Εικ.87. Φοιτητές συνεδρίας διάλεξης.....	σ.232
Εικ.88 Επεξεργασία εκκρεμούς συνεδρίας διάλεξης.....	σ.234
Εικ.89. Πεπερασμένη ημερομηνία και ώρα συνεδρίας διάλεξης.....	σ.235
Εικ.90. Επεξεργασία τρέχουσας συνεδρίας διάλεξης.....	σ.235
Εικ.91. Διαγραφή συνεδρίας διάλεξης.....	σ.236
Εικ.92. Προβολή στοιχείων παρουσίας φοιτητή.....	σ.237
Εικ.93. Επιλογέας κατάστασης απουσίας.....	σ.239
Εικ.94. Επεξεργασία παρουσίας.....	σ.240
Εικ.95. Αδυναμία επεξεργασίας παρουσίας πεπερασμένης συνεδρίας διάλεξης.....	σ.241
Εικ.96. Αδυναμία επεξεργασίας παρουσίας που έχει αίτηση δικαιολόγησης.....	σ.242
Εικ.97. Μενού Φοιτητή.....	σ.243
Εικ.98. Εγγραφή σε ομάδα.....	σ.243
Εικ.99. Κλειστή προς εγγραφές ομάδα.....	σ.244
Εικ.100. Γεμάτη ομάδα.....	σ.245
Εικ.101. Απεγγραφή απο ομάδα.....	σ.246
Εικ.102. Αδυναμία απεγγραφής λόγω κλειστής ομάδας.....	σ.247
Εικ.103. Δήλωση παρουσίας.....	σ.248
Εικ.104. Ηδη δηλωμένη παρουσία.....	σ.249
Εικ.105. Κλειστή δήλωση παρουσιών.....	σ.250
Εικ.106. Αδυναμία δήλωσης παρουσίας, λόγω καταχωρημένης απουσίας.....	σ.251
Εικ.107. Δημιουργία αίτησης δικαιολόγησης απουσίας.....	σ.252
Εικ.108. Επιβεβαίωση αίτησης.....	σ.253
Εικ.109. Ηδη υποβεβλημένη αίτηση.....	σ.254
Εικ.110. Εξαντλημένο όριο δικαιολογημένων απουσιών.....	σ.255
Εικ.111. Εξάντληση ορίου αδικαιολόγητων απουσιών.....	σ.256
Εικ.112. Προβολή στοιχείων αίτησης δικαιολόγησης απουσίας.....	σ.257
Εικ.113 Μενού Γραμματέα.....	σ.259
Εικ.114. Προβολή στοιχείων αίτησης δικαιολόγησης απουσίας.....	σ.260
Εικ.115. Αξιολόγηση αίτησης δικαιολόγησης απουσίας (έγκριση ή απόρριψη).....	σ.261
Εικ.116. Αδυναμία αξιολόγησης αίτησης μετα απο 30 μέρες.....	σ.262

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχ.1: Use Case Diagram.....	σ.27
Σχ.2: Class Diagram.....	σ.37
Σχ.3: Δημιουργία Προγράμματος Μαθήματος - Activity Diagram.....	σ.52
Σχ.4: Δημιουργία Φοιτητών - Activity Diagram.....	σ.54
Σχ.5: Επεξεργασία ομάδας φοιτητών - Activity Diagram.....	σ.56
Σχ.6: Προβολή λίστας φοιτητών ανα ομάδα - Activity Diagram.....	σ.58
Σχ.7: Προσθήκη φοιτητή στην ομάδα - Activity Diagram.....	σ.59
Σχ.8: Αφαίρεση φοιτητή από την ομάδα - Activity Diagram.....	σ.60
Σχ.9: Εγγραφή / Απεγγραφή φοιτητή στην ομάδα - Activity Diagram.....	σ.62
Σχ.10: Δημιουργία Συνεδρίας Διάλεξης - Activity Diagram.....	σ.63
Σχ.11: Επεξεργασία τρέχουσας συνεδρίας διάλεξης - Activity Diagram.....	σ.66
Σχ.12: Δήλωση παρουσίας σε μια τρέχουσα διάλεξη - Activity Diagram.....	σ.68
Σχ.13: Επεξεργασία παρουσίας - Activity Diagram.....	σ.69
Σχ.14: Δημιουργία αίτησης δικαιολόγησης απουσίας - Activity Diagram.....	σ.71
Σχ.15: Αξιολόγηση Αίτησης δικαιολόγησης απουσίας - Activity Diagram.....	σ.72
Σχ.16: Σύνδεση χρήστη και πρόσβαση στις υπηρεσίες - Sequence Diagram.....	σ.74
Σχ.17: Εγγραφή και ενεργοποίηση καθηγητή - Sequence Diagram.....	σ.76
Σχ.18: Δημιουργία ομάδων και εγγραφή/απεγγραφή φοιτητή - Sequence Diagram.....	σ.78
Σχ.19: Δημιουργία και δήλωση παρουσίας - Sequence Diagram.....	σ.80
Σχ.20: Επεξεργασία Παρουσίας - Sequence Diagram.....	σ.82
Σχ.21: Δικαιολόγηση απουσίας - Sequence Diagram.....	σ.84
Σχ.22: Παρουσία - State Machine Diagram.....	σ.86
Σχ.23: Αίτηση δικαιολόγησης απουσίας - State Machine Diagram.....	σ.87
Σχ.24: Entity Relationship Diagram.....	σ.88
Σχ.25: Επίπεδα Αρχιτεκτονικής Spring Boot. [22].....	σ.95
Σχ.26: Αρχιτεκτονική Ροής Spring Boot. [22].....	σ.96
Σχ.27: Αρχιτεκτονική JPA σε επίπεδο κλάσης [23].....	σ.106
Σχ.28: ORM Tool [19].....	σ.112
Σχ.29: Αρχιτεκτονική του Hibernate στο υψηλό επίπεδο [21].....	σ.115
Σχ.30: Αρχιτεκτονική του Hibernate σε χαμηλότερο επίπεδο [21].....	σ.116
Σχ.31: Αρχιτεκτονική Πελάτη – Εξυπηρετητή [24].....	σ.118
Σχ.32: Τρόπος λειτουργίας ενός Json Web Token [53].....	σ.129
Σχ.33: Αναπαράσταση του Document Object Model.....	σ.133
Σχ.34: Αρχιτεκτονική της Angular [29].....	σ.135
Σχ.35: Angular Rendering [34].....	σ.141

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

HTTP	Hypertext Transfer Protocol
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
UML	Unified Modeling Language
MVC	Model View Controller
OOP	Object Oriented Programming
AOP	Aspect Oriented Programming
ORM	Object Relational Mapping
DAO	Data Access Object
CRUD	Create Read Update Delete
JSP	Java Server Pages
XML	Extensible Markup Language
JSON	JavaScript Object Notation
API	Application Programming Interface
DOM	Document Object Model
JWT	JSON Web Token
OAuth	Open Authorization
JPA	Java Persistence API
SQL	Structured Query Language
HQL	Hibernate Query Language
JPQL	Java Persistence Query Language
JTA	Java Transaction API
JNDI	Java Naming Directory Interface
JMS	Java Message Service
JCA	Java Cryptography Architecture
JMX	Java Management Extensions
SpEL	Spring Expression Language
RDBMS	Relational Database Management System
LDAP	Lightweight Directory Access Protocol
UI	User Interface

Κεφάλαιο 1: Εισαγωγή

1.1. Εισαγωγή

Μέχρι σήμερα η καταγραφή και η διαχείριση των παρουσιών αλλά και η δικαιολόγηση των απουσιών ανα παράδοση ενός μαθήματος γίνεται με έναν εξαιρετικά υποτυπώδη τρόπο, ενώ άλλες φορές καθόλου. Οι καθηγητές χάνουν πολύτιμο χρόνο από την παράδοση προσπαθώντας να ελέγξουν και να καταγράψουν τις παρουσίες των φοιτητών. Οι φοιτητές πολλές φορές δεν προλαβαίνουν να δικαιολογήσουν τις απουσίες τους, ενώ η γραμματεία επιβαρύνεται με επιπλέον γραφειοκρατικές διαδικασίες. Ενα ακόμα πρόβλημα που προκύπτει είναι ότι χάνονται πολύτιμα δεδομένα που θα μπορούσαν να αξιοποιηθούν για να δοθεί για παράδειγμα μια σαφή εικόνα για τον ρυθμό προσέλευσης φοιτητών στις ακαδημαϊκές διαλέξεις. Τέλος, διακυβεύεται η αξιοπιστία και η οργάνωση της διαδικασίας διαχείρισης τους. Θα ήταν λοιπόν χρήσιμο ένα σύστημα διαχείρισης παρουσιών για τις παραδόσεις των ακαδημαϊκών διαλέξεων είτε θεωρητικών, είτε εργαστηριακών.

Με γνώμονα την αναγκαιότητα αυτή, αναπτύχθηκε το παρόν διαδικτυακό σύστημα, στο οποίο ο διαχειριστής του συστήματος θα δημιουργεί ένα πρόγραμμα μαθήματος για κάθε μάθημα, στο οποίο θα ανεβάζει ένα csv ή excel αρχείο με τους αριθμούς μητρώων των φοιτητών, που είναι εγγεγραμμένοι σε αυτό. Επίσης θα ρυθμίζει το πόση ώρα θα διαρκέσουν οι θεωρητικές και εργαστηριακές διαλέξεις, τον μέγιστο αριθμό τους, και ποιοι θα είναι οι διδάσκοντες του προγράμματος για το μάθημα.

Ο καθηγητής θα μπορεί να φτιάχνει διαλέξεις εργαστηρίου ή θεωρίας καθώς και ομάδες φοιτητών ανα πρόγραμμα μαθήματος για το οποίο όμως θα έχει την δικαιοδοσία ως μέλος του. Επίσης θα μπορεί να δημιουργεί συνεδρίες για τις διαλέξεις και τις ομάδες φοιτητών που επιθυμεί, για την ημερομηνία και ώρα που πρόκειται να διεξαχθεί η παράδοση. Παράλληλα θα μπορεί να παρεμβαίνει στις ομάδες των φοιτητών προσθέτοντας ή αφαιρώντας φοιτητές, όσο ακόμα δεν υπάρχει προγραμματισμένη συνεδρία διάλεξης που να αφορά την ομάδα.

Ο φοιτητής θα πρέπει να γραφτεί σε μια ομάδα του αντίστοιχου προγράμματος μαθήματος. Έτσι θα μπορεί σε κάθε συνεδρία μίας διάλεξης στην οποία μετέχει είτε είναι εργαστηριακή, είτε θεωρητική, να εισάγεται στο σύστημα και να δηλώνει την παρουσία του όσο ο καθηγητής έχει ενεργοποιημένη την φόρμα για τις δηλώσεις παρουσιών. Κατά την διάρκεια μιας διάλεξης, ο καθηγητής θα μπορεί να ανοίξει ή να κλείσει την φόρμα δηλώσεων παρουσιών κατά το δοκούν ή να μεταβάλλει μια παρουσία σε απουσία και το αντίστροφο.

Επειτα ο φοιτητής έχει την δυνατότητα εντός σαράντα οκτώ ωρών από την στιγμή της καταχώρησης της απουσίας του και αν πληροί τις προϋποθέσεις, να υποβάλλει αίτηση δικαιολόγησης. Η γραμματεία βλέπει τις αιτήσεις των φοιτητών και έπειτα από αξιολόγηση εντός τριάντα ημερών από την αίτηση, τις εγκρίνει ή τις απορρίπτει. Τέλος, ο φοιτητής μπορεί να βλέπει εύκολα με τα κατάλληλα φίλτρα αναζήτησης, την πορεία της αίτησής του κάθε στιγμή.

1.2. Στόχος Εφαρμογής

Το σύστημα διαχείρισης παρουσιών στοχεύει στην αυτοματοποίηση και απλούστευση της διαδικασίας διαχείρισης των παρουσιών των φοιτητών για το Χαροκόπειο Πανεπιστήμιο. Παράλληλα βοηθά στην μείωση της γραφειοκρατίας και των τετριμμένων διαδικασιών κερδίζοντας χρόνο για όλα τα μέλη της ακαδημαϊκής κοινότητας και προωθεί ένα οριζόντιο μοντέλο αλληλεπίδρασης μεταξύ τους, για την διεκπεραίωση των διαδικασιών που δεσμεύεται να απλουστεύσει.

Επίσης προσφέρει μεγάλη ευελιξία σε όλες τις διαδικασίες που αφορούν όλα τα εμπλεκόμενα μέλη, θέτοντας σε προτεραιότητα τις ανάγκες και τις ιδιαιτερότητες του ανθρώπινου παράγοντα. Επιπροσθέτως προσδίδει μεγαλύτερη διαφάνεια και αξιοπιστία στην όλη διαδικασία, προσφέροντας συνάμα την δυνατότητα συστηματικής καταγραφής και μελλοντικής αξιοποίησης των δεδομένων που προκύπτουν από αυτήν.

Τέλος, συμβάλλει στην ανάδειξη της αναγκαιότητας της συνέπειας και της πειθαρχίας ως αρετές που πρέπει να μεταλαμπαδεύονται στους φοιτητές ως μέλη της ακαδημαϊκής κοινότητας.

Κεφάλαιο 2: Ανάλυση του Συστήματος

2.1. Εισαγωγή

Στο παρόν κεφάλαιο παρατίθεται διεξοδικά η ανάλυση απαιτήσεων, καθώς και το κομμάτι που αφορά την ανάλυση και σχεδίαση του συστήματος διαχείρισης παρουσιών συνολικά, με τα αντίστοιχα UML διαγράμματα.

Για κάθε UML διάγραμμα υπάρχει και μια αντίστοιχη τεκμηρίωση για την καλύτερη και αποτελεσματικότερη κατανόησή του. Επίσης στο τέλος του κεφαλαίου αναλύονται οι κυριότερες λειτουργικές και μη λειτουργικές απαιτήσεις του συστήματος.

2.2. Ανάλυση Απαιτήσεων του Συστήματος

Η ανάλυση απαιτήσεων είναι η διαδικασία καθορισμού των προσδοκιών των χρηστών για μια εφαρμογή που πρόκειται να κατασκευαστεί ή να τροποποιηθεί. Περιλαμβάνει όλες τις εργασίες που διεξάγονται για τον εντοπισμό των αναγκών των διαφόρων εμπλεκόμενων μερών. Ως εκ τούτου, ανάλυση απαιτήσεων σημαίνει ανάλυση, τεκμηρίωση, επικύρωση και διαχείριση των απαιτήσεων του εκάστοτε λογισμικού ή συστήματος. [54]

Οι απαιτήσεις του συστήματος διαχείρισης παρουσιών αναλύονται σύντομα στην ενότητα 2.2, ενώ επίσης παρέχεται πιο εκτενής ανάλυση με το αντίστοιχο UML και συγκεκριμένα use case διάγραμμα, στην ενότητα 2.3.

UML

Το όνομα UML αποτελεί συντομογραφία του Unified Modeling Language. Η UML είναι μια τυποποιημένη γλώσσα μοντελοποίησης αποτελούμενη από ένα πλήρες σύνολο διαγραμμάτων. Αναπτύχθηκε για να βοηθήσει τους προγραμματιστές συστημάτων και λογισμικού για τον προσδιορισμό, την οπτικοποίηση, την κατασκευή και την τεκμηρίωση των αντικειμένων που διέπουν ένα σύστημα λογισμικού, καθώς επίσης και για την μοντελοποίηση επιχειρηματικών διαδικασιών όχι απαραίτητα σχετικών με λογισμικό. [55]

Βασικά χαρακτηριστικά και πλεονεκτήματα της γλώσσας UML:

- Η UML αντιπροσωπεύει μια συλλογή βέλτιστων πρακτικών μηχανικής που έχουν αποδειχθεί επιτυχείς στην μοντελοποίηση μεγάλων και πολύπλοκων συστημάτων. [55]
- Η UML αποτελεί πολύ σημαντικό μέρος της ανάπτυξης αντικειμενοστραφούς λογισμικού και της διαδικασίας ανάπτυξης λογισμικού γενικά. [55]
- Η UML χρησιμοποιεί ως επι το πλείστον γραφικές σημειώσεις και εξειδικευμένα διαγράμματα για να εκφράσει αποτελεσματικά την σχεδίαση έργων λογισμικού. [55]
- Η UML βοηθά τις ομάδες που σχετίζονται με ένα έργο να επικοινωνούν, να διερευνούν τα πιθανά σχέδια και να επικυρώνουν τον αρχιτεκτονικό σχεδιασμό του λογισμικού. [55]

Τύποι UML διαγραμμάτων που χρησιμοποιήθηκαν:

- Use Case Diagram
- Class Diagram
- Activity Diagram
- Sequence Diagram
- State Machine Diagram
- Entity Relationship Diagram

2.2. Ανάλυση Απαιτήσεων Συστήματος

Για την καλύτερη οργάνωση και συντονισμό της διαδικασίας δημιουργίας του συστήματος διαχείρισης παρουσιών, αλλά και για την αποτελεσματικότερη κατανόηση των επιμέρους λειτουργιών που το διέπουν, ήταν απαραίτητο να αναλυθούν οι διαδικασίες που το σύστημα καλείται να βελτιώσει ή να επιλύσει.

Εν ολίγοις, προκειμένου να δημιουργηθεί βέλτιστη και γερή βάση, δηλαδή αποτελεσματικά σχεδιασμένα θεμέλια απαραίτητα για την ομαλή υλοποίησή του, πραγματοποιήθηκε η ανάλυση των απαιτήσεων του συστήματος.

Είναι απαραίτητο να τονιστεί ότι στην παρούσα ενότητα γίνεται μια γενική αναφορά των απαιτήσεων του συστήματος. Η ακριβέστερη και λεπτομερέστερη ανάλυσή τους γίνεται στην ενότητα 2.3.1 με το Use Case Diagram. Παρακάτω παρατίθενται οι απαιτήσεις που αφορούν το σύστημα αλλά και τους επιμέρους ρόλους του.

Διαχειριστής Συστήματος (Admin):

- Είσοδος στο σύστημα με username και κωδικό πρόσβασης. (Ο Διαχειριστής Συστήματος εισάγεται απευθείας κατά την εκκίνηση της εφαρμογής - αν δεν υπάρχει κανένας χρήστης ακόμη - μέσω του server side για λόγους ασφαλείας).
- Δημιουργία και επεξεργασία χρηστών συστήματος.
- Ανέβασμα αρχείου csv/excel με τους φοιτητές κάθε τμήματος.
- Δημιουργία σχολών και τμημάτων και επεξεργασία ή διαγραφή τους.
- Δημιουργία προγραμμάτων μαθημάτων και επεξεργασία ή διαγραφή τους.
- Κατά την δημιουργία ενός προγράμματος μαθήματος, δυνατότητα επιλογής των καθηγητών που πρόκειται να συμμετάσχουν και ανεβάσματος αρχείου csv/excel με τους φοιτητές που θα λάβουν μέρος στο μάθημα.
- Δημιουργία διαλέξεων και επεξεργασία ή διαγραφή τους.
- Δημιουργία ομάδων φοιτητών και επεξεργασία ή διαγραφή τους.
- Προσθήκη, αφαίρεση φοιτητή σε/από ομάδα.
- Προσθήκη συνεδριών διαλέξεων και αν είναι εκκρεμείς δυνατότητα επεξεργασίας ή διαγραφής τους.
- Καταχώρηση παρουσιών / απουσιών ανα φοιτητή.
- Δημιουργία αιτήσεων δικαιολόγησης απουσιών και δυνατότητα αξιολόγησής τους.

Καθηγητής (Teacher):

- Εγγραφή στο σύστημα.
- Εισαγωγή στο σύστημα με το username και password, έπειτα από την ενεργοποίηση του καθηγητή από τον Διαχειριστή Συστήματος και την μετάθεσή του σε τμήμα σχολής.
- Προβολή προσωπικού προφίλ και δυνατότητα αλλαγής του κωδικού πρόσβασης.
- Δημιουργία διαλέξεων και επεξεργασία ή διαγραφή τους.
- Δημιουργία ομάδων φοιτητών και επεξεργασία ή διαγραφή τους.
- Προσθαφαίρεση φοιτητών σε ομάδες.
- Δημιουργία συνεδριών διαλέξεων και επεξεργασία ή διαγραφή τους.
- Καταχώρηση παρουσιών ή απουσιών ανα φοιτητή όσο μια συνεδρία διάλεξης είναι τρέχουσα.

Φοιτητής (Student):

- Εισαγωγή στο σύστημα με το username και password, μετά την καταχώρησή του μέσω του αρχείου csv/excel από τον Διαχειριστή Συστήματος.
- Προβολή προσωπικού προφίλ και δυνατότητα αλλαγής του κωδικού πρόσβασης.
- Προβολή της λίστας με τις ομάδες και εγγραφή ή απεγγραφή (αν επιτρέπεται από το σύστημα) σε / από όποια ομάδα επιλέξει.
- Προβολή λίστας με τις τρέχουσες διαλέξεις και δήλωση παρουσίας σε μια από αυτές.
- Προβολή λίστας με τις αιτήσεις, πλοήγηση με τα κατάλληλα φίλτρα και δημιουργία αίτησης δικαιολόγησης μιας απουσίας που έχει λάβει χώρα τις τελευταίες 48 ώρες.

Γραμματέας:

- Εισαγωγή στο σύστημα με τα username και password, μετά την καταχώρησή του από τον διαχειριστή του συστήματος.
- Προβολή προσωπικού προφίλ και δυνατότητα αλλαγής του κωδικού πρόσβασης.
- Προβολή αιτήσεων δικαιολόγησης απουσίας και δυνατότητα αξιολόγησής τους (έγκρισης ή απόρριψης), εντός 30 ημερών από την κατάθεση της αίτησης.

Σύστημα:

- Νήμα (thread) που ρυθμίζει αυτόματα την κατάσταση του προγράμματος μαθήματος σε (τρέχον, εκκρεμές, πεπερασμένο) ανάλογα την περίοδο του τρέχοντος εξαμήνου (χειμερινό ή εαρινό).
- Υπολογισμός του ελάχιστου αριθμού μελών ομάδας από ειδικό αλγόριθμο, κατά την δημιουργία μιας συνεδρίας διάλεξης.
- Δημιουργία αδήλων παρουσιών (με status null) μετά το άνοιγμα των δηλώσεων παρουσιών για τους φοιτητές.
- Μετατροπή των αδήλων παρουσιών σε αδικαιολόγητες απουσίες, μετά το κλείσιμο των δηλώσεων παρουσιών.
- Υπολογισμός του ορίου των δικαιολογημένων απουσιών από ειδικό αλγόριθμο, κατά την δημιουργία μιας αίτησης δικαιολόγησης απουσιών.

- Υπολογισμός της ώρας περάτωσης της ομάδας, από την ώρα έναρξης που εισάγεται κατα την δημιουργία της και την διάρκεια που έχει καταχωρηθεί στο πρόγραμμα μαθήματος (είτε η ομάδα αφορά εργαστήριο είτε αφορά θεωρία).
- Υπολογισμός της ημερομηνίας και ώρας περάτωσης της συνεδρίας διάλεξης, από την ημερομηνία έναρξης που εισάγεται κατα την δημιουργία της και την ώρα περάτωσης της ομάδας που έχει επιλεγεί για αυτή τη συνεδρία διάλεξης.

2.3. Ανάλυση και Σχεδίαση Συστήματος

Για την ανάλυση και σχεδίαση του συστήματος χρησιμοποιούνται uml διαγράμματα με απώτερο στόχο να επεξηγήσουν και να αναδείξουν σε ένα ευρύ και πολύπλευρο φάσμα όλες τις κύριες πτυχές του συστήματος διαχείρισης παρουσιών. Δίνεται έτσι έμφαση σε καίρια σημεία και λειτουργίες του συστήματος, από διαφορετικές οπτικές γωνίες, με γνώμονα τον τρόπο με τον οποίο κάθε λειτουργία πρέπει να αναλυθεί.

Συγκεκριμένα η ευρεία παρουσίαση των λειτουργιών που επιτελεί κάθε ρόλος του συστήματος, αλλά και το ίδιο το σύστημα, οι αλληλεξαρτήσεις των λειτουργιών αυτών αλλά και οι μεταξύ τους συσχετίσεις, γίνεται με ένα Use Case Diagram.

Οι λειτουργίες του Use Case Diagram που χρήζουν βαθύτερης επεξήγησης, αναλύονται με Activity Diagrams τα οποία παραθέτουν την ροή των βασικότερων επιμέρους ενεργειών που πρέπει να εκτελεστούν, ώστε να έρθουν σε πέρας οι λειτουργίες που αναλύουν.

Τα Activity Diagrams για το σύστημα διαχείρισης παρουσιών εστιάζουν στην παράθεση των ενεργειών που περιγράφουν:

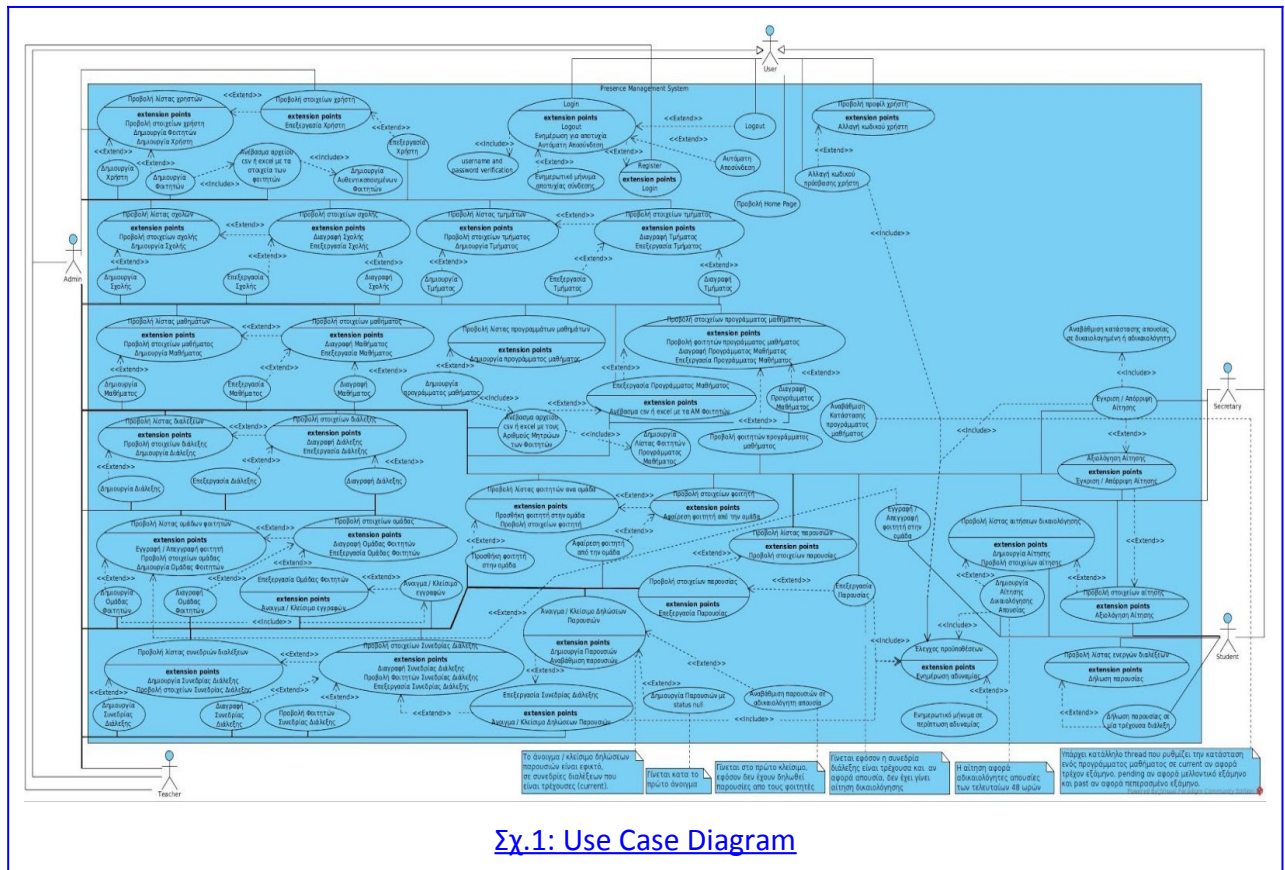
- Τα σημεία αλληλεπίδρασης των χρηστών με την client πλευρά της εφαρμογής που είναι σημαντικό να αναδειχθούν.
- Την αλληλεπίδραση της εφαρμογής από την πλευρά του client με τον χρήστη και κομμάτια της επικοινωνίας της με τον server, όπου θεωρείται απαραίτητο.
- Τους ελέγχους και λειτουργίες που γίνονται στον server και πρέπει να τονιστούν και την αλληλεπίδραση του server με την client πλευρά της εφαρμογής.

Η ανάλυση λειτουργιών με γνώμονα την επικοινωνία μεταξύ των εμπλεκομένων ρόλων και του συστήματος με χρονική σειρά γίνεται με Sequence Diagrams. Παραδείγματα λειτουργιών που αναλύθηκαν με αυτόν τον τρόπο στο παρόν σύστημα αποτελούν η “Σύνδεση του χρήστη στο σύστημα και η πρόσβασή του στις υπηρεσίες”, η “Δημιουργία ομάδων και εγγραφή/απεγγραφή”, η “Δημιουργία και δήλωση παρουσίας”, που θα περιγραφούν στην αντίστοιχη ενότητα παρακάτω.

Οι κυριότερες καταστάσεις που διέρχονται κρίσιμες οντότητες του συστήματος, όπως είναι η αίτηση και η παρουσία, και η επίδειξη των καίριων ενεργειών που τις μεταβάλλουν, αναλύονται με State Machine Diagrams. Παράλληλα η βασική ανάλυση των οντοτήτων της εφαρμογής καθώς και των μεταξύ τους σχέσεων πληθικότητας γίνεται με ένα Class Diagram.

Τέλος η σχεδίαση της βάσης δεδομένων και των συσχετίσεων μεταξύ των πινάκων γίνεται με ένα διάγραμμα οντοτήτων συσχετίσεων (ER Diagram).

2.3.1. Use Case Diagram



Στο παρόν Use Case Diagram, παρατίθενται όλοι οι ρόλοι που συμμετέχουν στο σύστημα διαχείρισης παρουσιών καθώς και οι λειτουργίες που αυτοί επιτελούν. Παρατίθενται επίσης και οι λειτουργίες που επιτελεί το ίδιο το σύστημα. Για όλες τις λειτουργίες που εμπεριέχονται στο διάγραμμα υπάρχουν οι αντίστοιχες μεταξύ τους συσχετίσεις ή αλληλεξαρτήσεις (τύπου include ή extend).

Πιο συγκεκριμένα οι συμμετέχοντες ρόλοι είναι οι εξής:

- Διαχειριστής Συστήματος (Admin)
- Καθηγητής (Teacher)
- Φοιτητής (Student)
- Γραμματέας (Secretary)

Ο ρόλος του Διαχειριστή Συστήματος (Admin), αφορά έναν και μοναδικό χρήστη στο σύστημα. Όλοι οι ρόλοι συσχετίζονται με generalization με έναν γενικότερο τύπο με το όνομα User. Αυτό γίνεται με σκοπό να διατηρηθεί καθαρό το διάγραμμα και να είναι ευδιάκριτη η συσχέτιση του User με λειτουργίες που εκτελούνται από όλους τους ρόλους του συστήματος, οι οποίες είναι οι εξής:

- Είσοδος στο Σύστημα
- Αποσύνδεση
- Προβολή Home Page
- Προβολή Προφίλ χρήστη
- Αλλαγή κωδικού πρόσβασης χρήστη

Κατα την είσοδο ενός χρήστη στο σύστημα και εφόσον ο χρήστης είναι ενεργοποιημένος από τον Διαχειριστή του Συστήματος, πραγματοποιείται η αυθεντικοποίησή του μέσω του username και του password που εισάγει.

Η λειτουργία **“Είσοδος στο σύστημα”** συνδέεται με συσχέτιση include με την λειτουργία **“Αυθεντικοποίηση ονόματος χρήστη και κωδικού πρόσβασης”**, διότι μετά την είσοδο ενός χρήστη στο σύστημα, εκείνο σίγουρα θα επιχειρήσει την αναγνώρισή του.

Σε περίπτωση αποτυχίας ο χρήστης ενημερώνεται με κατάλληλο μήνυμα. Επίσης αν μείνει συνδεδεμένος στο σύστημα χωρίς να εκτελέσει κάποια ενέργεια για ένα συγκεκριμένο χρονικό διάστημα γίνεται από το σύστημα αυτόματη αποσύνδεση. Τέλος μπορεί να επιλέξει ο ίδιος την αποσύνδεσή του από το σύστημα.

Οι λειτουργίες **“Αποσύνδεση”**, **“Αυτόματη Αποσύνδεση”** και **“Ενημερωτικό μήνυμα αποτυχίας σύνδεσης”** συνδέονται με συσχέτιση extend με την λειτουργία **“Είσοδος στο σύστημα”** καθώς δεν θα συμβούν σίγουρα μετά την είσοδο του χρήστη στο σύστημα. Παρομοίως κατα την λειτουργία **“Προβολή προφίλ χρήστη”** ο χρήστης μπορεί να μην εκτελέσει την λειτουργία **“Αλλαγή κωδικού πρόσβασης χρήστη”** και για αυτό η συσχέτιση μεταξύ των δύο αυτών λειτουργιών γίνεται επίσης με extend.

Για όλες τις λειτουργίες του συστήματος πλην των λειτουργιών **“Προβολή Home Page”**, **“Είσοδος στο σύστημα”**, **“Εγγραφή στο σύστημα”** ο χρήστης θα πρέπει πρώτα να έχει πραγματοποιήσει την λειτουργία **“Είσοδος στο σύστημα”**. Υπάρχει δηλαδή μια συσχέτιση τύπου **include** με όλες αυτές τις λειτουργίες, η οποία σχηματικά παραλείπεται για λόγους ευμορφίας του διαγράμματος.

Για κάθε μία από τις λειτουργίες του διαγράμματος, ο χρήστης ανάλογα με τον ρόλο του και την δικαιοδοσία που κατέχει μπορεί να επιτελεί τις λειτουργίες που περιγράφονται στο διάγραμμα Use Case. Με στόχο την καλύτερη παρουσίαση των λειτουργιών αυτών γίνεται η διάκρισή τους σε θεματικές ενότητες όπως αυτές υπάρχουν και στην εφαρμογή και είναι οι εξής:

- Σχολές (Πανεπιστημίου)
- Τμήματα (Σχολών)
- Μαθήματα (ανα Τμήμα)
- Προγράμματα Μαθημάτων (Σε ένα μάθημα αντιστοιχεί και ένα πρόγραμμα μαθήματος)
- Διαλέξεις (ανα Πρόγραμμα Μαθήματος)
- Ομάδες Φοιτητών (ανα Πρόγραμμα Μαθήματος)
- Φοιτητές ανα Ομάδα
- Συνεδρίες Διαλέξεων
- Παρουσίες (ανα Συνεδρία Διάλεξης)
- Αιτήσεις Δικαιολόγησης Απουσίας
- Χρήστες

Παρακάτω παρουσιάζονται οι λειτουργίες με επίκεντρο τον εκάστοτε ρόλο και ανα θεματική ενότητα. Παρουσιάζονται επίσης και οι λειτουργίες του συστήματος.

Διαχειριστής Συστήματος (Admin):

Ο διαχειριστής του συστήματος έχει πρόσβαση στις περισσότερες από τις λειτουργίες του συστήματος και σε όλες τις σχολές και τα τμήματα του πανεπιστημίου.

Χρήστες:

- Προβολή και πλοήγηση στην λίστα χρηστών.
- Προβολή στοιχείων χρήστη.
- Δημιουργία (Μπορούν να δημιουργηθούν χρήστες με όλους τους ρόλους εκτός από ρόλο Admin που είναι μόνο ένας στο σύστημα) και επεξεργασία χρήστη συστήματος (Στην επεξεργασία συμπεριλαμβάνεται και η ενεργοποίηση του εγγεγραμμένου καθηγητή και η μετάθεσή του σε τμήμα).
- Ανέβασμα csv/excel αρχείου με τους προς εγγραφή φοιτητές στο σύστημα.

Σχολές:

- Προβολή λίστας σχολών του πανεπιστημίου.
- Προβολή στοιχείων σχολής.
- Δημιουργία, επεξεργασία, διαγραφή, σχολής του πανεπιστημίου.

Τμήματα:

- Προβολή λίστας τμημάτων των σχολών του πανεπιστημίου.
- Προβολή στοιχείων τμήματος.
- Δημιουργία, επεξεργασία, διαγραφή τμήματος των σχολών του πανεπιστημίου.

Μαθήματα:

- Προβολή λίστας μαθημάτων (όλων ή ανα τμήμα του πανεπιστημίου).
- Προβολή στοιχείων μαθήματος.
- Δημιουργία, επεξεργασία, διαγραφή μαθήματος.

Προγράμματα Μαθημάτων:

- Προβολή προγραμμάτων μαθημάτων (όλων ή ανα τμήμα).
- Προβολή στοιχείων προγράμματος μαθήματος και λίστας φοιτητών που περιλαμβάνει.
- Δημιουργία, επεξεργασία, διαγραφή προγράμματος μαθήματος.
- Ανέβασμα csv/excel αρχείου με τους αριθμούς μητρώων των φοιτητών που πρόκειται να συμπεριληφθούν στο πρόγραμμα μαθήματος κατά την δημιουργία του ή πιθανώς και κατά την επεξεργασία του (εάν το επιτρέπει το σύστημα).

Διαλέξεις:

- Προβολή λίστας διαλέξεων (Με βάση το επιλεγμένο τμήμα, πρόγραμμα μαθήματος και είδος διάλεξης (εργαστηριακή ή θεωρητική)).
- Προβολή στοιχείων διάλεξης.
- Δημιουργία, επεξεργασία, διαγραφή διάλεξης.

Ομάδες Φοιτητών:

- Προβολή λίστας ομάδων φοιτητών (Με βάση το επιλεγμένο τμήμα και πρόγραμμα μαθήματος).
- Προβολή στοιχείων ομάδας φοιτητών.
- Δημιουργία, επεξεργασία, διαγραφή ομάδας φοιτητών.
- Ανοιγμα ή κλείσιμο εγγραφών φοιτητών στην ομάδα κατά την επεξεργασία της ομάδας.

Φοιτητές ανα Ομάδα:

- Προβολή λίστας φοιτητών ανα Ομάδα (Με βάση το επιλεγμένο τμήμα, πρόγραμμα μαθήματος και τύπο ομάδας (θεωρητική ή εργαστηριακή)).
- Προβολή στοιχείων φοιτητή ομάδας.
- Προσθήκη φοιτητή στην ομάδα.
- Αφαίρεση φοιτητή από την ομάδα.

Συνεδρία διάλεξης:

- Προβολή λίστας συνεδριών διαλέξεων (Με βάση το επιλεγμένο τμήμα, πρόγραμμα μαθήματος, τύπο διάλεξης και διάλεξη).
- Προβολή στοιχείων συνεδρίας διάλεξης.
- Προβολή λίστας φοιτητών συνεδρίας διάλεξης.
- Δημιουργία, επεξεργασία και διαγραφή συνεδρίας διάλεξης.
- Ανοιγμα ή Κλείσιμο δηλώσεων παρουσιών κατά την επεξεργασία της τρέχουσας συνεδρίας διάλεξης.

Παρουσίες:

- Προβολή λίστας παρουσιών συνεδρίας διάλεξης (Με βάση το επιλεγμένο τμήμα, πρόγραμμα μαθήματος, τύπο διάλεξης, διάλεξη και συνεδρία διάλεξης).
- Προβολή στοιχείων παρουσίας.
- Επεξεργασία παρουσίας (Επικύρωση παρουσίας ή μετατροπή της σε απουσία όσο η συνεδρία διάλεξης είναι ακόμη τρέχουσα).

Αιτήσεις Δικαιολόγησης Απουσίας:

- Προβολή λίστας αιτήσεων (ανά τμήμα).
- Προβολή στοιχείων αίτησης.
- Δημιουργία αίτησης δικαιολόγησης για κάποιον φοιτητή.
- Αξιολόγηση αίτησης (Εγκριση ή απόρριψη).

Καθηγητής (Teacher):

Ο καθηγητής έχει πρόσβαση μόνο σε λειτουργίες που αφορούν αντικείμενα σχετικά με το πρόγραμμα μαθήματος που συμμετέχει και φυσικά περιορίζεται στο τμήμα στο οποίο ανήκει.

Διαλέξεις:

- Προβολή λίστας διαλέξεων (Με βάση το πρόγραμμα μαθήματος και είδος διάλεξης (εργαστηριακή ή θεωρητική)).
- Προβολή στοιχείων διάλεξης.
- Δημιουργία, επεξεργασία, διαγραφή διάλεξης.

Ομάδες Φοιτητών:

- Προβολή λίστας ομάδων φοιτητών (Με βάση το επιλεγμένο πρόγραμμα μαθήματος).
- Προβολή στοιχείων ομάδας φοιτητών.
- Δημιουργία, επεξεργασία, διαγραφή ομάδας φοιτητών.
- Ανοιγμα ή κλείσιμο εγγραφών φοιτητών στην ομάδα κατά την επεξεργασία της ομάδας.

Φοιτητές ανα Ομάδα:

- Προβολή λίστας φοιτητών ανα Ομάδα (Με βάση το επιλεγμένο πρόγραμμα μαθήματος και τύπο ομάδας (θεωρητική ή εργαστηριακή)).
- Προβολή στοιχείων φοιτητή ομάδας.
- Προσθήκη φοιτητή στην ομάδα.
- Αφαίρεση φοιτητή από την ομάδα.

Συνεδρία διάλεξης:

- Προβολή λίστας συνεδριών διαλέξεων (Με βάση το επιλεγμένο πρόγραμμα μαθήματος, τύπο διάλεξης και διάλεξη).
- Προβολή στοιχείων συνεδρίας διάλεξης.
- Προβολή λίστας φοιτητών συνεδρίας διάλεξης.

- Δημιουργία, επεξεργασία και διαγραφή συνεδρίας διάλεξης.
- Ανοιγμα ή Κλείσιμο δηλώσεων παρουσιών κατα την επεξεργασία της τρέχουσας συνεδρίας διάλεξης.

Παρουσίες:

- Προβολή λίστας παρουσιών συνεδρίας διάλεξης (Με βάση το επιλεγμένο πρόγραμμα μαθήματος, τύπο διάλεξης, διάλεξη και συνεδρία διάλεξης).
- Προβολή στοιχείων παρουσίας.
- Επεξεργασία παρουσίας (Επικύρωση παρουσίας ή μετατροπή της σε απουσία όσο η συνεδρία διάλεξης είναι ακόμη τρέχουσα).

Φοιτητής (Student):

Ο φοιτητής έχει πρόσβαση μόνο σε λειτουργίες που αφορούν αντικείμενα στα οποία συμμετέχει ή είναι ιδιοκτήτης και φυσικά περιορίζεται στο τμήμα στο οποίο ανήκει.

Ομάδες Φοιτητών:

- Προβολή λίστας ομάδων φοιτητών (Με βάση το επιλεγμένο πρόγραμμα μαθήματος που συμμετέχει).
- Εγγραφή σε ομάδα.
- Απεγγραφή από ομάδα.

Τρέχουσα Διάλεξη (Συνεδρία Διάλεξης):

- Δήλωση παρουσίας (σε μία τρέχουσα διάλεξη που συμμετέχει).

Αιτήσεις Δικαιολόγησης Απουσίας:

- Προβολή λίστας αιτήσεων.
- Προβολή στοιχείων αίτησης.
- Δημιουργία αίτησης δικαιολόγησης απουσίας.

Γραμματέας (Secretary):

Ο γραμματέας έχει πρόσβαση μόνο σε λειτουργίες που αφορούν το τμήμα στο οποίο ανήκει.

Αιτήσεις Δικαιολόγησης Απουσίας:

- Προβολή λίστας αιτήσεων.
- Προβολή στοιχείων αίτησης.
- Αξιολόγηση αίτησης δικαιολόγησης απουσίας (έγκριση / απόρριψη).

Σύστημα:

Το σύστημα επιτελεί κάποιες απαραίτητες για την εφαρμογή λειτουργίες, ορισμένες από τις οποίες παρουσιάζονται στο παραπάνω Use Case Diagram ενώ άλλες αναλύονται σε Activity ή Sequence Diagrams τα οποία παρατίθενται παρακάτω.

Λειτουργίες του συστήματος όπως απεικονίζονται στο Use Case Diagram:

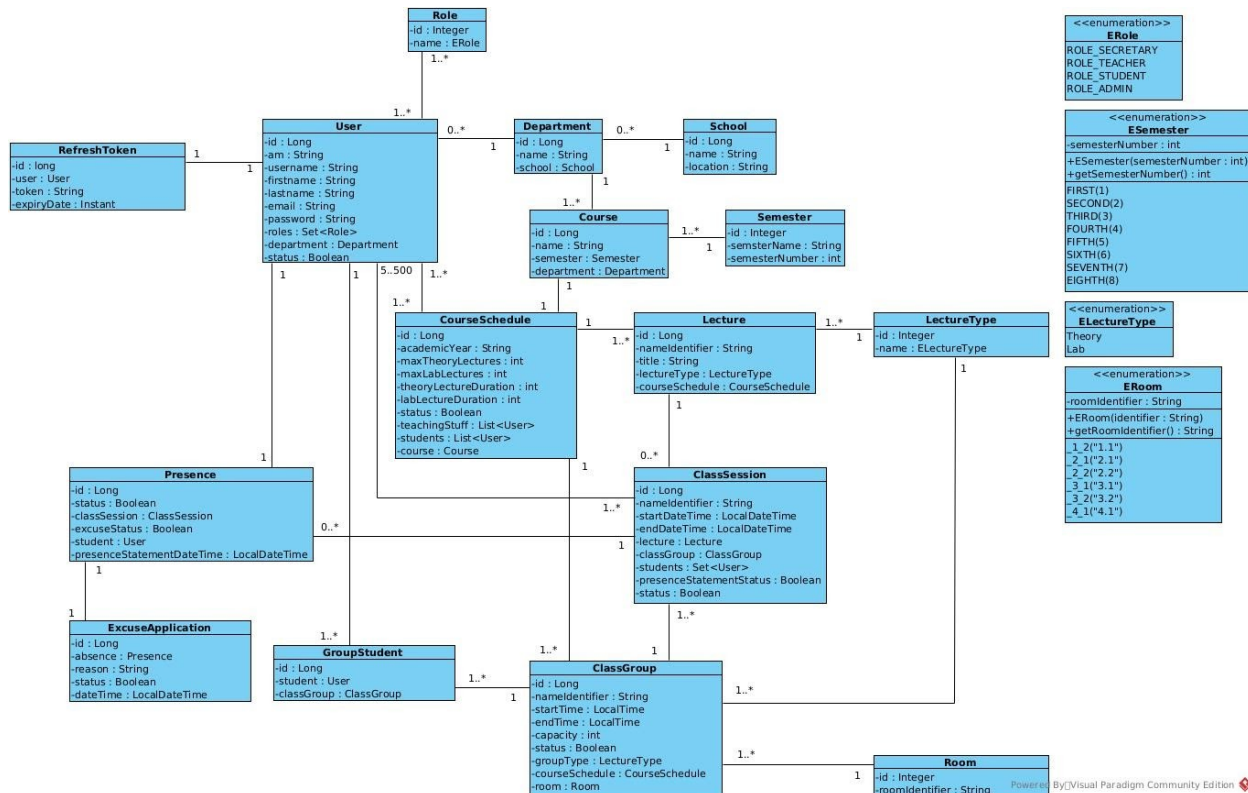
- Αυθεντικοποίηση ονόματος χρήστη και κωδικού πρόσβασης.
- Ενημερωτικό μήνυμα αποτυχίας σύνδεσης.
- Αυτόματη αποσύνδεση.
- Δημιουργία αυθεντικοποιημένων φοιτητών.
- Δημιουργία λίστας φοιτητών προγράμματος μαθήματος
- Αναβάθμιση κατάστασης προγράμματος μαθήματος
- Δημιουργία αδήλων παρουσιών (με status null).
- Αναβάθμιση αδήλων παρουσιών σε αδικαιολόγητες απουσίες.
- Αναβάθμιση κατάστασης (status) απουσίας σε δικαιολογημένη ή αδικαιολόγητη.
- Έλεγχος προϋποθέσεων (Γίνεται από το σύστημα για λειτουργίες που περιλαμβάνονται στο Use Case).
- Ενημερωτικό μήνυμα σε περίπτωση αδυναμίας πραγματοποίησης (Γίνεται σε περίπτωση αποτυχίας του ελέγχου προϋποθέσεων).

Επεξηγηματικές σημειώσεις Use Case διαγράμματος:

Παρακάτω παρατίθενται κάποιες επεξηγηματικές σημειώσεις καθώς και οι ενέργειες τις οποίες αυτές αφορούν:

- **“Άνοιγμα / Κλείσιμο δηλώσεων παρουσιών”**: Το άνοιγμα / κλείσιμο δηλώσεων παρουσιών είναι εφικτό σε συνεδρίες διαλέξεων που είναι τρέχουσες (current).
- **“Δημιουργία αδήλων παρουσιών με status null”**: Η λειτουργία “Δημιουργία αδήλων παρουσιών με status null”, γίνεται κατά το πρώτο άνοιγμα των δηλώσεων της τρέχουσας διάλεξης.
- **“Αναβάθμιση αδήλων παρουσιών σε αδικαιολόγητες απουσίες”**: Γίνεται από το πρώτο κλείσιμο των δηλώσεων (και όσες φορές αυτό επαναληφθεί), εφόσον δεν έχουν δηλωθεί παρουσίες από τους φοιτητές.
- **“Επεξεργασία παρουσίας”**: Γίνεται εφόσον η συνεδρία διάλεξης είναι τρέχουσα και αν αφορά απουσία, δεν έχει γίνει αίτηση δικαιολόγησης.
- **“Δημιουργία αίτησης δικαιολόγησης απουσίας”**: Η αίτηση αφορά αδικαιολόγητες απουσίες των τελευταίων 48 ωρών.
- **“Αναβάθμιση κατάστασης προγράμματος μαθήματος”**: Υπάρχει κατάλληλο thread που ρυθμίζει την κατάσταση ενός προγράμματος μαθήματος σε current αν αφορά τρέχον εξάμηνο, pending αν αφορά μελλοντικό εξάμηνο και past αν αφορά πεπερασμένο εξάμηνο.

2.3.2. Class Diagram



Σχ.2: Class Diagram

Στο παρόν Class Diagram αναπαρίστανται οι οντότητες (entities) από τις οποίες δομείται το σύστημα με τα χαρακτηριστικά που τις διέπουν και τις μεταξύ τους συσχετίσεις. Το διάγραμμα αυτό συμβάλλει στην βαθύτερη κατανόηση των συστατικών στοιχείων των οντοτήτων αλλά και του είδους κάθε συσχέτισης. Ειδικότερα μέσα από το ποιόν των συσχετίσεων και των εμπλεκόμενων μεταξύ τους κλάσεων, γίνεται ευκολότερα αντιληπτή η κεντρική ιδέα δόμησης που ακολουθήθηκε.

Παρακάτω αναλύεται διεξοδικά η κάθε οντότητα (entity) και συσχέτιση του διαγράμματος.

Οι οντότητες που πρόκειται να αναλυθούν παρακάτω έχουν ως χαρακτηριστικό ένα **μοναδικό id** αυτόματα παραγόμενο από το σύστημα, ώστε να ξεχωρίζονται οι εγγραφές τους στην βάση δεδομένων.

Role:

Η οντότητα Role, αποτελείται από ένα μοναδικό id και το πεδίο name που είναι τύπου ERole και αφορά το enumeration με τους ρόλους.

User:

Η οντότητα του User, αναπαριστά κάθε χρήστη που εμπλέκεται στο σύστημα.

Περιγραφή χαρακτηριστικών:

- **username, password:** Είναι τα χαρακτηριστικά που χρησιμοποιούνται για την αυθεντικοποίηση του χρήστη.
- **firstname, lastname, email, roles:** Απαρτίζουν μαζί με το username και το password τα βασικά προσωπικά στοιχεία του κάθε χρήστη.
- **am:** Είναι ο αριθμός μητρώου του φοιτητή. Είναι μοναδικός και παίρνει τιμή μόνο όταν ο χρήστης έχει τον ρόλο του φοιτητή, διαφορετικά καταχωρείται ως null.
- **department:** Είναι το τμήμα της σχολής που ανήκει ο κάθε χρήστης του συστήματος εκτός του Διαχειριστή για τον οποίο έχει τιμή null.
- **status:** Αφορά την κατάσταση του χρήστη, εκτός του Διαχειριστή για τον οποίο έχει τιμή null. Δείχνει αν ο χρήστης είναι ενεργοποιημένος ή όχι, ώστε να έχει πρόσβαση στον λογαριασμό του.

Περιορισμοί μεταξύ χαρακτηριστικών της οντότητας:

- Θα πρέπει για κάθε χρήστη, ο συνδυασμός του am (αν είναι φοιτητής), username και email, να είναι μοναδικός.

Συσχετίσεις:

- Η οντότητα User συνδέεται με την οντότητα Role με συσχέτιση “**many to many**” καθώς ένας χρήστης μπορεί να έχει πάνω από έναν ρόλο αλλά και πολλοί ρόλοι να αντιστοιχούν σε έναν και μόνο χρήστη.

RefreshToken:

Η οντότητα του RefreshToken, αφορά ένα Json Web Token το οποίο χρησιμοποιείται για την αυθεντικοποίηση του χρήστη ώστε να έχει πρόσβαση σε πόρους του συστήματος σε περίπτωση που έχει λήξει το βασικό accessToken του.

Περιγραφή χαρακτηριστικών:

- **user:** Αφορά τον χρήστη στον οποίο ανήκει το refreshToken.
- **token:** Είναι μοναδικό hashed token για την πιστοποίηση αυθεντικότητας του χρήστη.
- **expiryDate:** Είναι η ημερομηνία λήξης του refreshToken μετά το πέρας της οποίας, δεν είναι δυνατή η αυθεντικοποίηση του χρήστη με αυτό.

Συσχετίσεις:

- Η οντότητα RefreshToken συνδέεται με την οντότητα User με συσχέτιση **“one to one”** καθώς ένα refreshToken, αντιστοιχεί σε έναν και μόνο χρήστη.

School:

Η οντότητα του School, αναπαριστά κάθε σχολή του πανεπιστημίου.

Περιγραφή χαρακτηριστικών:

- **name, location:** Το όνομα και η τοποθεσία της σχολής.

Department:

Η οντότητα του Department, αναπαριστά κάθε τμήμα για κάθε σχολή του πανεπιστημίου.

Περιγραφή χαρακτηριστικών:

- **name:** Το όνομα του εκάστοτε τμήματος.
- **school:** Η σχολή στην οποία αντιστοιχεί το τμήμα

Συσχετίσεις:

- Η οντότητα Department συνδέεται με την οντότητα School με συσχέτιση “many to one” καθώς πολλά τμήματα μπορούν να αντιστοιχούν σε μια σχολή.

Semester:

Η οντότητα του Semester, αναπαριστά τα ακαδημαϊκά εξάμηνα φοίτησης στο πανεπιστήμιο.

Περιγραφή χαρακτηριστικών:

- **semesterName:** Το όνομα του εκάστοτε ακαδημαϊκού εξαμήνου. Το όνομα του εξαμήνου δημιουργείται από το σύστημα, ανάλογα με το εξάμηνο που αντιπροσωπεύει. Για παράδειγμα το πρώτο ακαδημαϊκό εξάμηνο αντιστοιχεί στην ονομασία “First”.
- **semesterNumber:** Το νούμερο του ακαδημαϊκού εξαμήνου. Το νούμερο του εξαμήνου δημιουργείται από το σύστημα, ανάλογα με το εξάμηνο που αντιπροσωπεύει. Προκύπτει από τον enumeration τύπο ESemester στον οποίο αντιστοιχίζεται το νούμερο κάθε εξαμήνου με την ονομασία του. Για παράδειγμα το πρώτο ακαδημαϊκό εξάμηνο με όνομα “First”, αντιστοιχίζεται στο νούμερο “1”.

Course:

Η οντότητα του Course, αναπαριστά κάθε μάθημα ανα τμήμα σχολής του πανεπιστημίου.

Περιγραφή χαρακτηριστικών:

- **name:** Το όνομα του εκάστοτε μαθήματος.
- **semester:** Το εξάμηνο που αντιστοιχεί το μάθημα.

- **department:** Το τμήμα στο οποίο αντιστοιχεί το μάθημα.

Συσχετίσεις:

- Η οντότητα Course συνδέεται με την οντότητα Semester με συσχέτιση “many to one” καθώς πολλά μαθήματα μπορούν να αντιστοιχούν στο ίδιο εξάμηνο.
- Η οντότητα Course συνδέεται με την οντότητα Department με συσχέτιση “many to one” καθώς πολλά μαθήματα μπορούν να αντιστοιχούν σε ένα τμήμα.

CourseSchedule:

Η οντότητα του CourseSchedule, αναπαριστά το πρόγραμμα μαθήματος που αντιστοιχεί σε κάθε μάθημα.

Περιγραφή χαρακτηριστικών:

- **academicYear:** Το ακαδημαϊκό έτος του προγράμματος μαθήματος το οποίο είναι το τρέχον έτος κατά την δημιουργία του.
- **maxTheoryLectures:** Ο μέγιστος αριθμός των διαλέξεων θεωρητικού τύπου που μπορούν να δημιουργηθούν για το πρόγραμμα μαθήματος.
- **maxLabLectures:** Ο μέγιστος αριθμός των διαλέξεων εργαστηριακού τύπου που μπορούν να δημιουργηθούν για το πρόγραμμα μαθήματος.
- **theoryLectureDuration:** Η διάρκεια των διαλέξεων θεωρητικού τύπου που χρησιμεύει στις θεωρητικού τύπου ομάδες που αντιστοιχούν στο πρόγραμμα μαθήματος.
- **labLectureDuration:** Η διάρκεια των διαλέξεων εργαστηριακού τύπου που χρησιμεύει στις εργαστηριακού τύπου ομάδες που αντιστοιχούν στο πρόγραμμα μαθήματος.
- **status:** Η κατάσταση του προγράμματος μαθήματος η οποία δείχνει αν το πρόγραμμα μαθήματος είναι εκκρεμές (πριν γίνει τρέχον), τρέχον ή πεπερασμένο.
- **teachingStuff:** Το σύνολο των καθηγητών που συμμετέχουν στην διδασκαλία του μαθήματος στο οποίο αντιστοιχεί το πρόγραμμα. Οι καθηγητές μπορούν να είναι από ένας έως τρεις.
- **students:** Το σύνολο των φοιτητών που αντιστοιχούν στο πρόγραμμα μαθήματος. Προσάπτονται σε μια λίστα φοιτητών που προκύπτει μετά απο αναγνώριση των αριθμών μητρώων τους, αφότου έχουν ανέβει στο σύστημα μέσω ειδικού αρχείου csv ή excel.

- **course:** Το μάθημα στο οποίο αντιστοιχεί το πρόγραμμα μαθήματος.

Περιορισμοί μεταξύ χαρακτηριστικών της οντότητας:

- Θα πρέπει για κάθε πρόγραμμα μαθήματος ο συνδυασμός του μαθήματος με το ακαδημαϊκό έτος να είναι μοναδικός (δηλαδή το κάθε μάθημα να είναι μοναδικό για κάθε ακαδημαϊκό έτος).

Συσχετίσεις:

- Η οντότητα CourseSchedule συνδέεται με την οντότητα Course με συσχέτιση “one to one” καθώς ένα πρόγραμμα μαθήματος αντιστοιχεί σε ένα και μόνο μάθημα.
- Η οντότητα CourseSchedule συνδέεται με την οντότητα User με συσχέτιση “**many to many**” καθώς πολλά προγράμματα μαθημάτων μπορούν να έχουν πολλούς καθηγητές (από έναν έως τρεις) ή πολλούς φοιτητές.

LectureType:

Η οντότητα LectureType, αναπαριστά τον τύπο διάλεξης ο οποίος μπορεί να είναι είτε θεωρητικός είτε εργαστηριακός. Αποτελείται από ένα μοναδικό id και το πεδίο name που είναι τύπου ELectureType και αφορά το enumeration με τους τύπους διαλέξεων.

Lecture:

Η οντότητα Lecture, αναπαριστά κάθε διάλεξη που αντιστοιχεί σε ένα πρόγραμμα μαθήματος.

Περιγραφή χαρακτηριστικών:

- **namelidentifier:** Το όνομα της διάλεξης το οποίο ακολουθεί συγκεκριμένη τυποποίηση. Τα ονόματα των διαλέξεων φτιάχνονται από το σύστημα για κάθε μια διάλεξη που πρόκειται να δημιουργηθεί σύμφωνα με τον μέγιστο αριθμό διαλέξεων που έχει οριστεί στο πρόγραμμα μαθήματος. Το όνομα έχει ως πρόθεμα τον τύπο διάλεξης και επίθεμα το νούμερο που αντιστοιχεί σε αυτήν κατα σειρά. Για παράδειγμα αν ο μέγιστος αριθμός διαλέξεων εργαστηριακού τύπου είναι 8, θα δημιουργηθούν 8

ονόματα ως εξής: lab_1, lab_2, ..., lab_8. Ο καθηγητής ή ο διαχειριστής συστήματος επιλέγει απλώς ένα από τα ονόματα αυτά σύμφωνα με το ποία διάλεξη θέλει να δημιουργήσει.

- **title:** Ο τίτλος της διάλεξης.
- **lectureType:** Ο τύπος που αντιστοιχεί στην διάλεξη. Μπορεί να είναι “εργαστήριο” ή “θεωρία”.
- **courseSchedule:** Το πρόγραμμα μαθήματος για το οποίο δημιουργείται η διάλεξη.

Περιορισμοί μεταξύ χαρακτηριστικών της οντότητας:

- Θα πρέπει για κάθε διάλεξη ο συνδυασμός του παραγόμενου από το σύστημα ονόματός της και του προγράμματος μαθήματος που αντιστοιχεί, να είναι μοναδικός.

Συσχετίσεις:

- Η οντότητα Lecture συνδέεται με την οντότητα LectureType με συσχέτιση “many to one” καθώς πολλές διαλέξεις μπορούν να είναι του ίδιου τύπου.
- Η οντότητα Lecture συνδέεται με την οντότητα CourseSchedule με συσχέτιση “many to one” καθώς πολλές διαλέξεις μπορούν να αντιστοιχούν σε ένα συγκεκριμένο πρόγραμμα μαθήματος.

Room:

Η οντότητα Room, αναπαριστά την αίθουσα στην οποία μια ομάδα φοιτητών παρακολουθεί μια διάλεξη.

Περιγραφή χαρακτηριστικών:

- **roomIdentifier:** Το όνομα της αίθουσας το οποίο ακολουθεί συγκεκριμένη τυποποίηση. Πιο συγκεκριμένα το όνομα κάθε αίθουσας παράγεται από το σύστημα και προκύπτει από τον enumeration τύπο ERoom ο οποίος αντιστοιχίζει ένα enumeration όνομα με την αντίστοιχη αρίθμηση της αίθουσας. Για παράδειγμα το όνομα _1_1 στο enumeration ERoom, αντιστοιχίζεται με την αρίθμηση “1.1” για την αντίστοιχη αίθουσα.

ClassGroup:

Η οντότητα ClassGroup, αναπαριστά κάθε ομάδα φοιτητών που αντιστοιχεί σε ένα πρόγραμμα μαθήματος.

Περιγραφή χαρακτηριστικών:

- **namelidentifier:** Το όνομα της ομάδας το οποίο ακολουθεί συγκεκριμένη τυποποίηση. Πιο συγκεκριμένα το όνομα κάθε ομάδας είναι μοναδικό ανα τύπο ομάδας (θεωρία ή εργαστήριο) για κάθε πρόγραμμα μαθήματος. Τα ονόματα των ομάδων φτιάχνονται εν μέρει από το σύστημα.
- Το όνομα έχει ως πρόθεμα την λέξη “group” και επίθεμα το νούμερο που έχει εισαχθεί από τον καθηγητή ή διαχειριστή του συστήματος κατα την δημιουργία του.
- Για παράδειγμα αν ο καθηγητής ή ο διαχειριστής του συστήματος επιλέξει το νούμερο 1, θα δημιουργηθεί για την ομάδα το όνομα “group_1”. Αυτό γίνεται για μεγαλύτερη ευελιξία στην ονοματοδοσία των ομάδων, καθώς μπορεί κάποια να πρέπει να διαγραφεί ή να αλλάξει αρίθμηση και δεν υπάρχει προκαθορισμένος μέγιστος αριθμός τους.
- **startTime:** Η ώρα έναρξης της παρακολούθησης συνεδριών διαλέξεων για την συγκεκριμένη ομάδα φοιτητών.
- **endTime:** Η ώρα περάτωσης της παρακολούθησης συνεδριών διαλέξεων για την συγκεκριμένη ομάδα φοιτητών.
- **capacity:** Η χωρητικότητα της ομάδας σε φοιτητές.
- **status:** Η κατάσταση της ομάδας ως προς την αποδοχή εγγραφών φοιτητών σε αυτήν. Δηλαδή “ανοικτή / κλειστή” για εγγραφές.
- **groupType:** Ο τύπος της ομάδας φοιτητών. Μπορεί να είναι είτε ομάδα που αφορά διαλέξεις θεωρίας, είτε ομάδα που αφορά διαλέξεις εργαστηρίου.
- **courseSchedule:** Το πρόγραμμα μαθήματος το οποίο αντιστοιχεί στην ομάδα.
- **room:** Η αίθουσα παρακολούθησης συνεδριών διαλέξεων για την ομάδα.

Συσχετίσεις:

- Η οντότητα ClassGroup συνδέεται με την οντότητα LectureType με συσχέτιση “many to one” καθώς πολλές ομάδες φοιτητών μπορούν να αφορούν διαλέξεις του ίδιου τύπου.

- Η οντότητα ClassGroup συνδέεται με την οντότητα CourseSchedule με συσχέτιση “**many to one**” καθώς πολλές ομάδες φοιτητών μπορούν να αντιστοιχούν σε ένα συγκεκριμένο πρόγραμμα μαθήματος.
- Η οντότητα ClassGroup συνδέεται με την οντότητα Room με συσχέτιση “**many to one**” καθώς πολλές ομάδες φοιτητών μπορούν να παρακολουθούν συνεδρίες διαλέξεων στην ίδια αίθουσα.

GroupStudent:

Η οντότητα GroupStudent, αναπαριστά την εγγραφή του φοιτητή σε μια ομάδα.

Περιγραφή χαρακτηριστικών:

- **student:** Ο φοιτητής που εγγράφεται στην ομάδα.
- **classGroup:** Η ομάδα στην οποία εγγράφεται ο φοιτητής.

Περιορισμοί μεταξύ χαρακτηριστικών της οντότητας:

- Θα πρέπει κάθε εγγραφή φοιτητή σε κάθε ομάδα να είναι μοναδική.

Συσχετίσεις:

- Η οντότητα GroupStudent συνδέεται με την οντότητα User με συσχέτιση “**many to one**” καθώς ίδιος φοιτητής μπορεί να εγγραφεί σε πολλές ομάδες.
- Η οντότητα GroupStudent συνδέεται με την οντότητα ClassGroup με συσχέτιση “**many to one**” καθώς πολλές εγγραφές φοιτητών μπορούν να αφορούν την ίδια ομάδα.

ClassSession:

Η οντότητα ClassSession, αναπαριστά την συνεδρία μιας διάλεξης, δηλαδή την διεξαγωγή της.

Περιγραφή χαρακτηριστικών:

- **namelIdentifier:** Το όνομα της συνεδρίας διάλεξης το οποίο ακολουθεί συγκεκριμένη τυποποίηση. Πιο συγκεκριμένα το όνομα κάθε συνεδρίας διάλεξης είναι μοναδικό ανα τύπο συνεδρίας διάλεξης (θεωρία ή εργαστήριο) για κάθε συνδυασμό ομάδας και διάλεξης του ίδιου προγράμματος μαθήματος.

Τα ονόματα των συνεδριών διαλέξεων φτιάχνονται εν μέρει από το σύστημα.

Το όνομα έχει ως πρόθεμα την λέξη “session” και επίθεμα το νούμερο που έχει εισαχθεί από τον καθηγητή ή διαχειριστή του συστήματος κατά την δημιουργία του.

Για παράδειγμα αν ο καθηγητής ή ο διαχειριστής του συστήματος επιλέξει το νούμερο 1, θα δημιουργηθεί συνεδρία διάλεξης με το όνομα “session_1”. Αυτό γίνεται για μεγαλύτερη ευελιξία στην ονοματοδοσία των συνεδριών διαλέξεων, καθώς μπορεί κάποια να πρέπει να διαγραφεί ή να αλλάξει αρίθμηση και δεν υπάρχει προκαθορισμένος μέγιστος αριθμός τους.

- **startDateTime:** Η ημερομηνία και ώρα έναρξης της συνεδρίας διάλεξης. Η ημερομηνία εισάγεται από τον καθηγητή ή διαχειριστή συστήματος κατά την δημιουργία της συνεδρίας διάλεξης ενώ η ώρα προέρχεται από την ώρα έναρξης παρακολούθησης της ομάδας φοιτητών που επιλέχθηκε για την συνεδρία διάλεξης.
- **endDateTime:** Η ημερομηνία και ώρα περάτωσης της συνεδρίας διάλεξης. Η ημερομηνία είναι η ίδια με την ημερομηνία έναρξης της συνεδρίας διάλεξης ενώ η ώρα προέρχεται από την ώρα περάτωσης παρακολούθησης της ομάδας φοιτητών που επιλέχθηκε για την συνεδρία διάλεξης.
- **lecture:** Η διάλεξη που αντιστοιχεί στην συνεδρία διάλεξης.
- **classGroup:** Η ομάδα φοιτητών που αντιστοιχεί στην συνεδρία διάλεξης. Στην ομάδα φοιτητών θα πρέπει να αντιστοιχεί ο ελάχιστος αριθμός φοιτητών που υπολογίζεται από το σύστημα καθώς και να είναι κλειστή προς εγγραφές.
- **students:** Οι φοιτητές που αντιστοιχούν στην συνεδρία διάλεξης. Οι φοιτητές της συνεδρίας διάλεξης προέρχονται από τους φοιτητές που είναι εγγεγραμμένοι στην επιλεγμένη για την συνεδρία διάλεξης ομάδα.
- **presenceStatementStatus:** Η κατάσταση δηλώσεων παρουσιών της διάλεξης. Η δηλώσεις μπορεί να είναι “ανοικτές” ή “κλειστές”. Αρχικά είναι κλειστές. Η κατάσταση δηλώσεων παρουσιών μπορεί να μεταβληθεί όταν η διάλεξη είναι τρέχουσα.
- **status:** Η κατάσταση της συνεδρίας διάλεξης η οποία μπορεί να είναι **εκκρεμής (pending)**, **τρέχουσα (current)** ή **πεπερασμένη (past)**.

Περιορισμοί μεταξύ χαρακτηριστικών της οντότητας:

Για κάθε συνεδρία διάλεξης θα πρέπει ο συνδυασμός ομάδας φοιτητών και διάλεξης να είναι μοναδικός.

Συσχετίσεις:

- Η οντότητα ClassSession συνδέεται με την οντότητα Lecture με συσχέτιση **“many to one”** καθώς πολλές συνεδρίες διάλεξης μπορούν να αντιστοιχούν σε μία διάλεξη. Δηλαδή η ίδια διάλεξη μπορεί να αναπαραχθεί για περισσότερες από μία φορές.
- Η οντότητα ClassSession συνδέεται με την οντότητα ClassGroup με συσχέτιση **“many to one”** καθώς πολλές συνεδρίες διάλεξης μπορούν να παρακολουθούνται από την ίδια ομάδα φοιτητών.

Presence:

Η οντότητα Presence, αναπαριστά την παρουσία του φοιτητή σε μια συνεδρία διάλεξης στην ομάδα της οποίας ανήκει.

Περιγραφή χαρακτηριστικών:

- **status:** Η κατάσταση της παρουσίας η οποία μπορεί να έχει την τιμή **null** που αντιστοιχεί σε **“αδήλωτη” παρουσία**, **true** η οποία αντιστοιχεί σε **“παρουσία”** και **false** η οποία αντιστοιχεί σε **“απουσία”**.
- **classSession:** Η συνεδρία διάλεξης της παρουσίας.
- **excuseStatus:** Η κατάσταση δικαιολόγησης απουσίας. Σε περίπτωση που η κατάσταση παρουσίας είναι **false** δηλαδή **“απουσία”**, η κατάσταση δικαιολόγησης απουσίας μπορεί να είναι **true** **“δικαιολογημένη”** ή **false** **“αδικαιολόγητη”**. Αν η κατάσταση παρουσίας δεν είναι false δηλαδή **“απουσία”**, τότε η κατάσταση δικαιολόγησης απουσίας έχει την τιμή null.
- **student:** Ο φοιτητής στον οποίο αντιστοιχεί η παρουσία.
- **presenceStatementDateTime:** Η ημερομηνία και ώρα δημιουργίας της **“αδήλωτης”** παρουσίας ή της δηλωμένης απουσίας ή παρουσίας.

Περιορισμοί μεταξύ χαρακτηριστικών της οντότητας:

Για κάθε παρουσία θα πρέπει ο συνδυασμός του φοιτητή και της συνεδρίας διάλεξης να είναι μοναδικός. Δηλαδή να αντιστοιχεί μια παρουσία ανα φοιτητή και συνεδρία διάλεξης.

Συσχετίσεις:

- Η οντότητα Presence συνδέεται με την οντότητα ClassSession με συσχέτιση “**many to one**” καθώς μια συνεδρία διάλεξης μπορεί να έχει πολλές παρουσίες φοιτητών.
- Η οντότητα Presence συνδέεται με την οντότητα User με συσχέτιση “**one to one**” καθώς μια παρουσία αντιστοιχεί σε έναν και μόνο φοιτητή.

ExcuseApplication:

Η οντότητα ExcuseApplication, αναπαριστά την αίτηση δικαιολόγησης απουσίας του φοιτητή για μια αδικαιολόγητη απουσία που καταχωρήθηκε εντός των τελευταίων 48 ωρών σε μια συνεδρία διάλεξης στην οποία συμμετέχει.

Περιγραφή χαρακτηριστικών:

- **absence:** Η αδικαιολόγητη απουσία που ο φοιτητής θέλει να δικαιολογήσει.
- **reason:** Ο λόγος για τον οποίο προέκυψε η απουσία.
- **status:** Η κατάσταση της αίτησης δικαιολόγησης. Μπορεί να είναι null δηλαδή εκκρεμής, true δηλαδή εγκεκριμένη ή false απορριμμένη.
- **dateTime:** Η ημερομηνία και ώρα κατάθεσης της αίτησης από τον φοιτητή ή τον διαχειριστή του συστήματος.

Περιορισμοί μεταξύ χαρακτηριστικών της οντότητας:

Η αίτηση δικαιολόγησης απουσίας, θα πρέπει να είναι μοναδική για κάθε απουσία.

Συσχετίσεις:

- Η οντότητα ExcuseApplication συνδέεται με την οντότητα Presence με συσχέτιση “**one to one**” καθώς μια αίτηση δικαιολόγησης αντιστοιχεί σε μία και μόνο παρουσία.

ERole:

Η οντότητα ERole, είναι ο enumeration τύπος που περιέχει τα είδη ρόλων.

Αυτοί οι ρόλοι είναι:

- Διαχειριστής Συστήματος (Admin)
- Καθηγητής (Teacher)
- Φοιτητής (Student)
- Γραμματέας (Secretary)

ESemester:

Η οντότητα ESemester, είναι ο enumeration τύπος που περιέχει τα ονόματα των ακαδημαϊκών εξαμήνων φοίτησης τα οποία αντιστοιχίζονται με ανάλογους με την ονομασία τους αριθμούς.

Περιγραφή χαρακτηριστικών:

- **semesterNumber:** Το νούμερο που αντιστοιχεί σε κάθε όνομα εξαμήνου στο enumeration. Για παράδειγμα το όνομα εξαμήνου **"First"**, αντιστοιχίζεται με το νούμερο **"1"**. Όταν χρειαστεί να αντληθεί κάπου στο σύστημα η πληροφορία για το εξάμηνο τότε με βάση το όνομα που έχει το enumeration και με την αντίστοιχη συνάρτηση getter, επιτυγχάνεται η πρόσβαση στο νούμερο που του αντιστοιχεί. Έτσι όταν χρειαστεί να διαπιστωθεί αν ένα εξάμηνο είναι χειμερινό ή εαρινό, αυτό γίνεται μετά από έλεγχο στο αντίστοιχο νούμερό του (αν είναι μονό ή ζυγό).

ELectureType:

Η οντότητα ELectureType, είναι το enumeration που περιέχει τα ονόματα που αφορούν τους τύπους διάλεξης. Οι τύποι αυτοί μπορούν να είναι **"Θεωρία" (Theory)** ή **"Εργαστήριο" (Lab)**.

ERoom:

Η οντότητα ERoom, είναι ο enumeration τύπος που περιέχει τα ονόματα των αιθουσών παρακολούθησης συνεδριών διαλέξεων από τις ομάδες φοιτητών, τα οποία αντιστοιχίζονται με ανάλογους με την ονομασία τους αριθμούς.

Περιγραφή χαρακτηριστικών:

- **roomIdIdentifier:** Το όνομα που αφορά την αρίθμηση της αίθουσας, το οποίο αντιστοιχίζεται στο όνομα του enumeration για την αίθουσα. Για παράδειγμα το enumeration με όνομα _1_1, αφορά την αίθουσα με αρίθμηση “1.1”.

Παρατήρηση σχετικά με το διάγραμμα:

Στο Class Diagram του συστήματος διαχείρισης παρουσιών, παρατηρείται κύκλος μεταξύ των οντοτήτων CourseSchedule, Lecture, ClassGroup και ClassSession. Αυτό συμβαίνει διότι στο σύστημα διαχείρισης παρουσιών η διάλεξη όπως και η ομάδα φοιτητών αντιστοιχούν σε ένα συγκεκριμένο πρόγραμμα μαθήματος.

Παράλληλα, η συνεδρία διάλεξης αφορά τον συνδυασμό της διάλεξης που πρόκειται να διεξαχθεί με την ομάδα φοιτητών που θα παρακολουθήσει την διεξαγωγή της. Ταυτόχρονα όμως, τόσο η διάλεξη όσο και η ομάδα φοιτητών της συνεδρίας διάλεξης θα πρέπει να σχετίζονται με το ίδιο πρόγραμμα μαθήματος.

Επιπλέον στο σύστημα δίνεται η δυνατότητα για δημιουργία πολλών συνεδριών διαλέξεων που μπορούν να αφορούν την ίδια διάλεξη και να την παρακολουθούν διαφορετικές ομάδες φοιτητών για το ίδιο πρόγραμμα μαθήματος. Δηλαδή μια διάλεξη μπορεί να έχει πολλαπλές διεξαγωγές.

Ετσι ανάλογα με την χωρητικότητα των αιθουσών ή τις απαιτήσεις του κάθε τμήματος του πανεπιστημίου που διεξάγονται οι συνεδρίες διαλέξεων, επιτυγχάνεται η ευελιξία στην

οργάνωση και διαχείριση τους και κατ' επέκταση η αποτελεσματικότερη διαχείριση των παρουσιών των φοιτητών που συμμετέχουν σε αυτές.

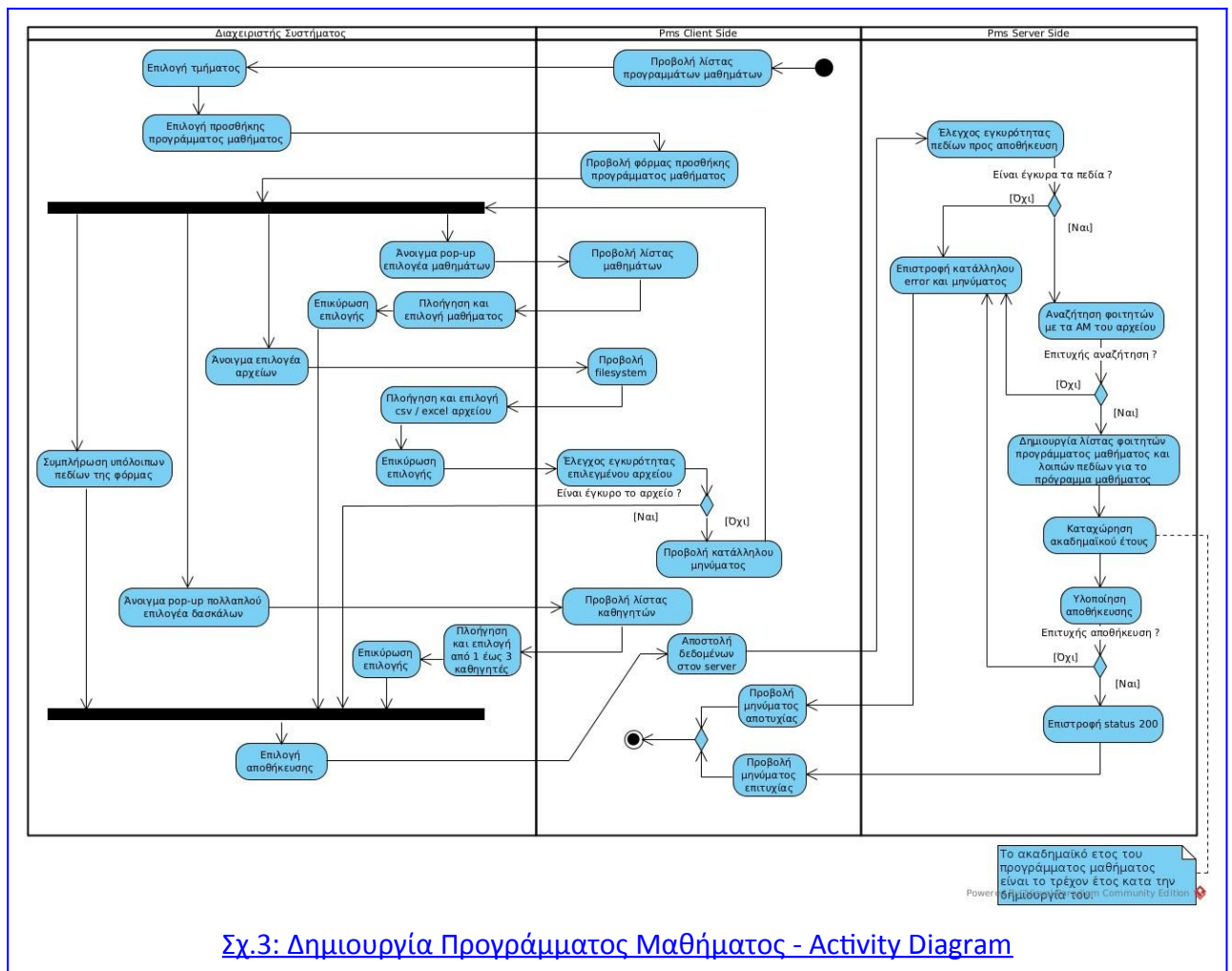
2.3.3. Activity Diagrams

Στην ενότητα αυτή παρατίθενται τα Activity Diagrams για την εις βάθος ανάλυση ορισμένων λειτουργιών που παρουσιάζονται στο Use Case διάγραμμα.

Η διεξοδική ανάλυση των λειτουργιών αυτών επιτυγχάνεται μέσω της περιγραφής της αλληλεπίδρασης του χρήστη με την client πλευρά της εφαρμογής και της ανάδειξης ορισμένων καίριων ενεργειών (κυρίως ελέγχων) που υλοποιούνται από την server πλευρά της εφαρμογής.

Είναι σημαντικό να αναφερθεί ότι οι ενέργειες που επιτελεί ο ρόλος του καθηγητή ή του γραμματέα στα παρακάτω Activity διαγράμματα αλλά και του φοιτητή στο διάγραμμα **“Δημιουργία αίτησης δικαιολόγησης απουσίας”**, επιτελούνται και απο τον διαχειριστή του συστήματος.

Δημιουργία Προγράμματος Μαθήματος



Σχ.3: Δημιουργία Προγράμματος Μαθήματος - Activity Diagram

Στο παραπάνω Activity διάγραμμα αναλύεται η λειτουργία της δημιουργίας προγράμματος μαθήματος. Το πρόγραμμα μαθήματος δημιουργείται από τον διαχειριστή του συστήματος. Αφού επιλέξει το τμήμα για τα μαθήματα του οποίου θέλει να δημιουργήσει τα αντίστοιχα προγράμματα μαθημάτων, επιλέγει την “προσθήκη προγράμματος μαθήματος” και συμπληρώνει και υποβάλλει την αντίστοιχη φόρμα.

Στο παραπάνω διάγραμμα παρουσιάζονται κάποιες ενέργειες στις οποίες αξίζει να γίνει αναφορά. Πρόκειται για ενέργειες που απορρέουν από χρήσιμες δυνατότητες που δίνονται στον χρήστη από την client πλευρά του συστήματος. Παράλληλα ενέργειες σημαντικές για την

έκβαση της δημιουργίας του προγράμματος μαθήματος επιτελεί και η server πλευρά του συστήματος.

Οι ενέργειες αυτές παρατίθενται παρακάτω και είναι:

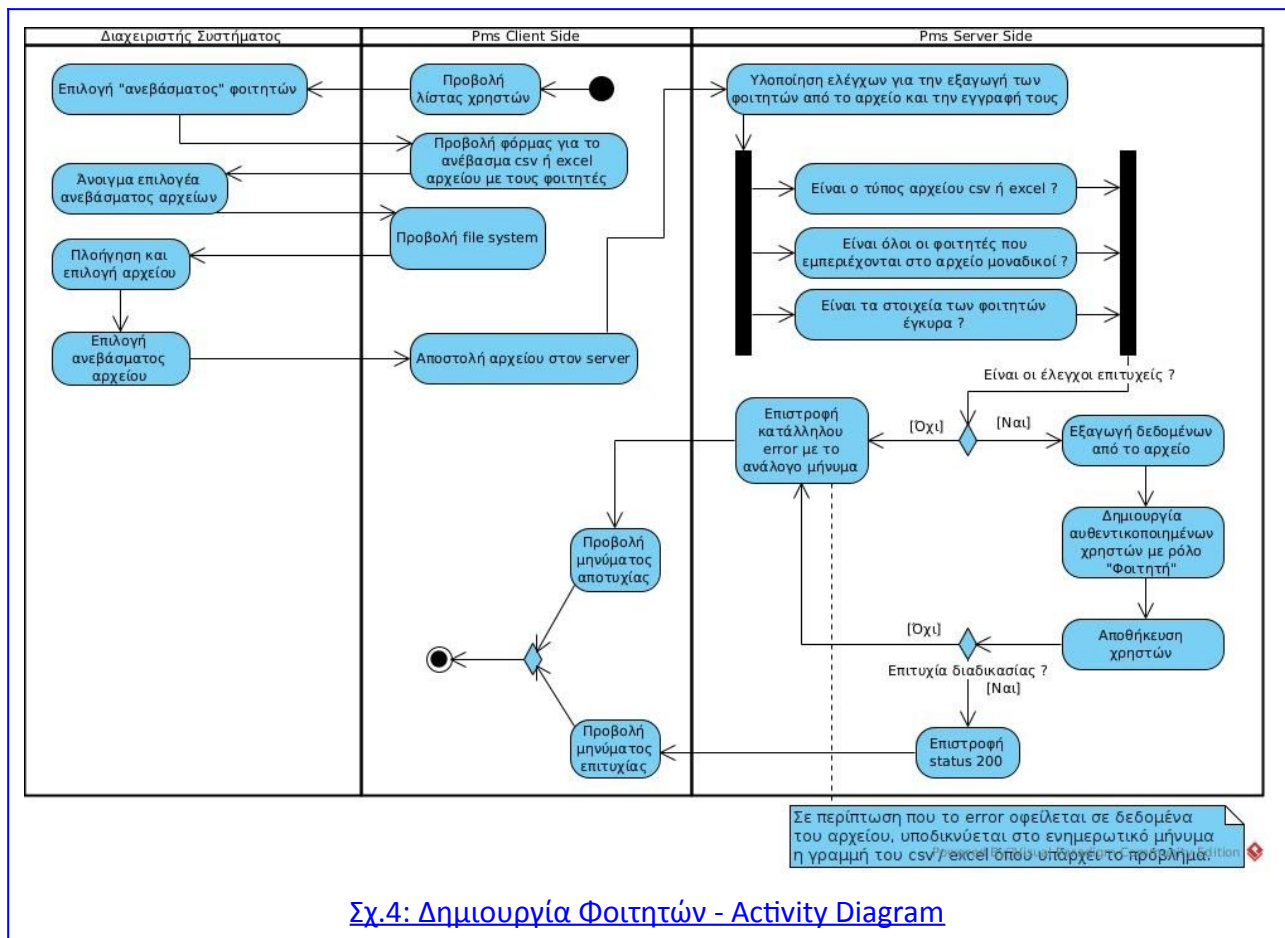
Απο την client πλευρά της εφαρμογής:

- Κατα την επιλογή μαθήματος η εμφάνιση ενός pop-up με λίστα των μαθημάτων του τμήματος ανα 3 ή 5 σε κάθε σελίδα με δυνατότητα ταξινόμησής τους με βάση το όνομα, το εξάμηνο ή το id τους. Υπάρχει επίσης και η δυνατότητα αναζήτησής τους, κάνοντας την πλοήγηση πιο εύκολη, γρήγορη και αποτελεσματική. Η σελιδοποίηση, το φιλτράρισμα και η ταξινόμηση γίνονται από την πλευρά του server καθιστώντας την εφαρμογή πιο γρήγορη και αποδοτική.
- Κατα την επιλογή καθηγητών του προγράμματος μαθήματος η εμφάνιση pop-up με παρόμοια κατασκευή με αυτήν που προαναφέρθηκε για την επιλογή μαθημάτων με την διαφορά ότι προσφέρει την πολλαπλή επιλογή για τους καθηγητές (από 1 έως 3) κατα μήκος όλων των σελίδων της λίστας.
- Κατα το ανέβασμα csv ή excel αρχείου με τους αριθμούς μητρώων των φοιτητών, ο έλεγχος για την εγκυρότητα του αρχείου.
- Η εμφάνιση κατάλληλου ενημερωτικού μηνύματος στον χρήστη σχετικού με την έκβαση της διαδικασίας.

Από την server πλευρά της εφαρμογής:

- Η αναζήτηση των φοιτητών των οποίων οι αριθμοί μητρώων αντιστοιχούν σε αυτούς που περιέχει το αρχείο csv ή excel και η δημιουργία κατάλληλης λίστας με αυτούς.
- Ο ορισμός και υπολογισμός του τρέχοντος ακαδημαϊκού έτους του προγράμματος μαθήματος ανάλογα την χρονολογία της δημιουργίας του.
- Οι έλεγχοι που αφορούν την διαδικασία δημιουργίας του προγράμματος μαθήματος και την επιστροφή κατάλληλου μηνύματος σε περίπτωση αποτυχίας.

Δημιουργία φοιτητών



Στο παραπάνω Activity διάγραμμα αναλύεται η λειτουργία που αφορά την δημιουργία των φοιτητών απο τον διαχειριστή του συστήματος. Οι εγγεγραμμένοι στο σύστημα φοιτητές δημιουργούνται από τον διαχειριστή του συστήματος, ο οποίος ανεβάζει ένα αρχείο csv ή excel με τα στοιχεία τους.

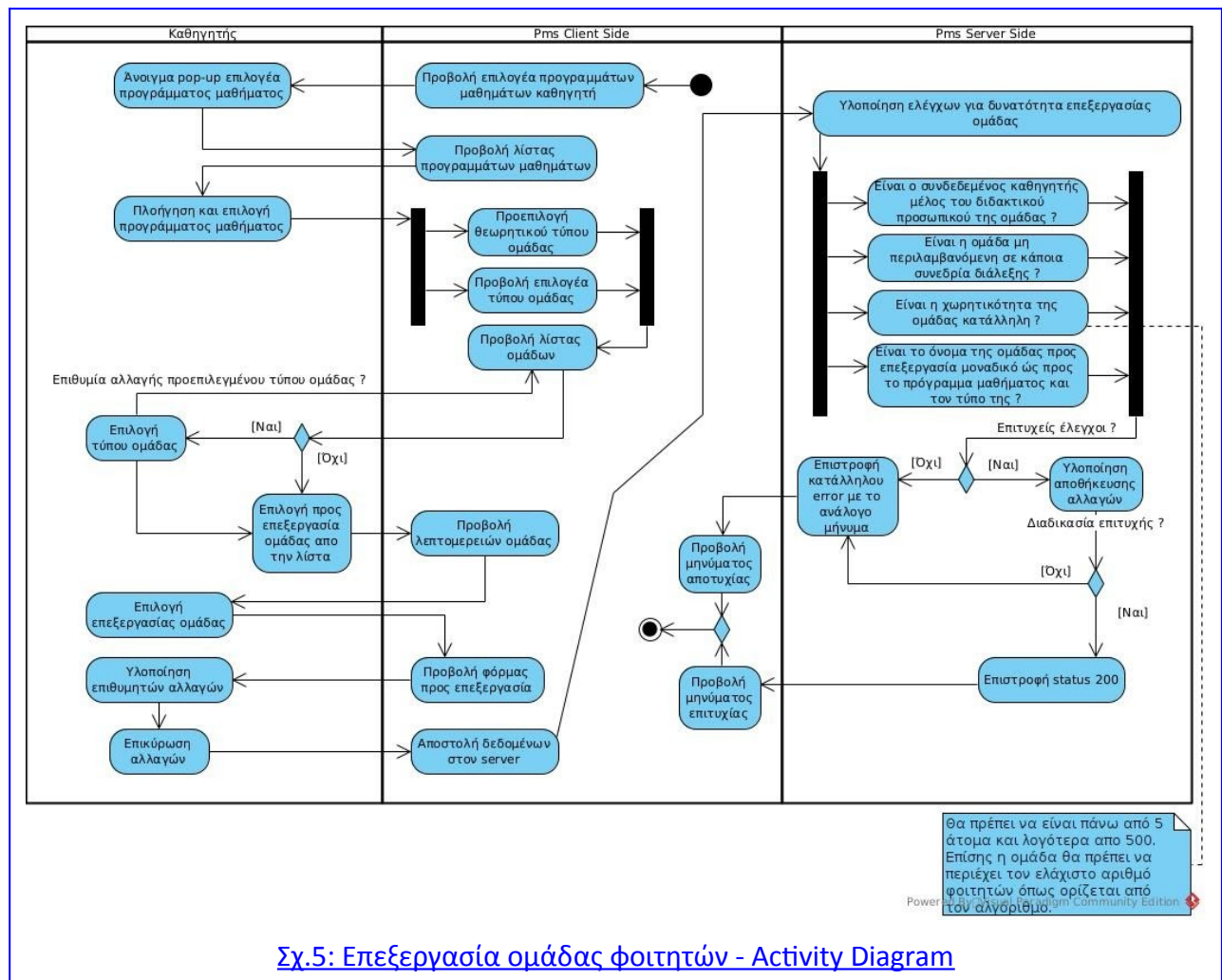
Σε αυτό το διάγραμμα ιδιαίτερη σημασία έχουν οι ενέργειες που γίνονται από την server πλευρά της εφαρμογής. Οι ενέργειες αυτές αφορούν κατάλληλους ελέγχους σχετικούς με το αν πληρούνται οι προϋποθέσεις για την εξαγωγή των φοιτητών από το csv ή excel αρχείο.

Παρουσιάζονται επίσης οι ενέργειες που αφορούν την εξαγωγή των δεδομένων των φοιτητών από το αρχείο, την δημιουργία αυθεντικοποιημένων χρηστών με ρόλο φοιτητή, την τελική αποθήκευσή τους και την δημιουργία και αποστολή κατάλληλου μηνύματος σε περίπτωση αποτυχίας.

Είναι σημαντικό να τονιστεί ότι σε περίπτωση αποτυχίας εξαγωγής των φοιτητών από το αρχείο και αποθήκευσής τους, εάν το error οφείλεται σε δεδομένα του αρχείου ο server δημιουργεί μήνυμα στο οποίο υποδεικνύεται η γραμμή του csv / ή excel αρχείου όπου υπάρχει το πρόβλημα, γίνεται δηλαδή validation σε κάθε δεδομένο του αρχείου.

Τέλος για την client πλευρά της εφαρμογής αξίζει να αναφερθεί ότι ανάλογα με την έκβαση της διαδικασίας παρουσιάζει το αντίστοιχο ενημερωτικό μήνυμα.

Επεξεργασία ομάδας φοιτητών



Στο παραπάνω Activity διάγραμμα αναλύεται η λειτουργία της επεξεργασίας ομάδας φοιτητών. Την λειτουργία αυτή επιτελεί ο καθηγητής ή ο διαχειριστής του συστήματος ο οποίος χρησιμοποιώντας κατάλληλα την λειτουργικότητα που του δίνεται από το client side της εφαρμογής μπορεί να αναζητήσει και να επιλέξει την ομάδα φοιτητών την οποία θέλει να επεξεργαστεί, να πραγματοποιήσει τις αλλαγές που θέλει και να υποβάλει τα στοιχεία της φόρμας που επεξεργάστηκε.

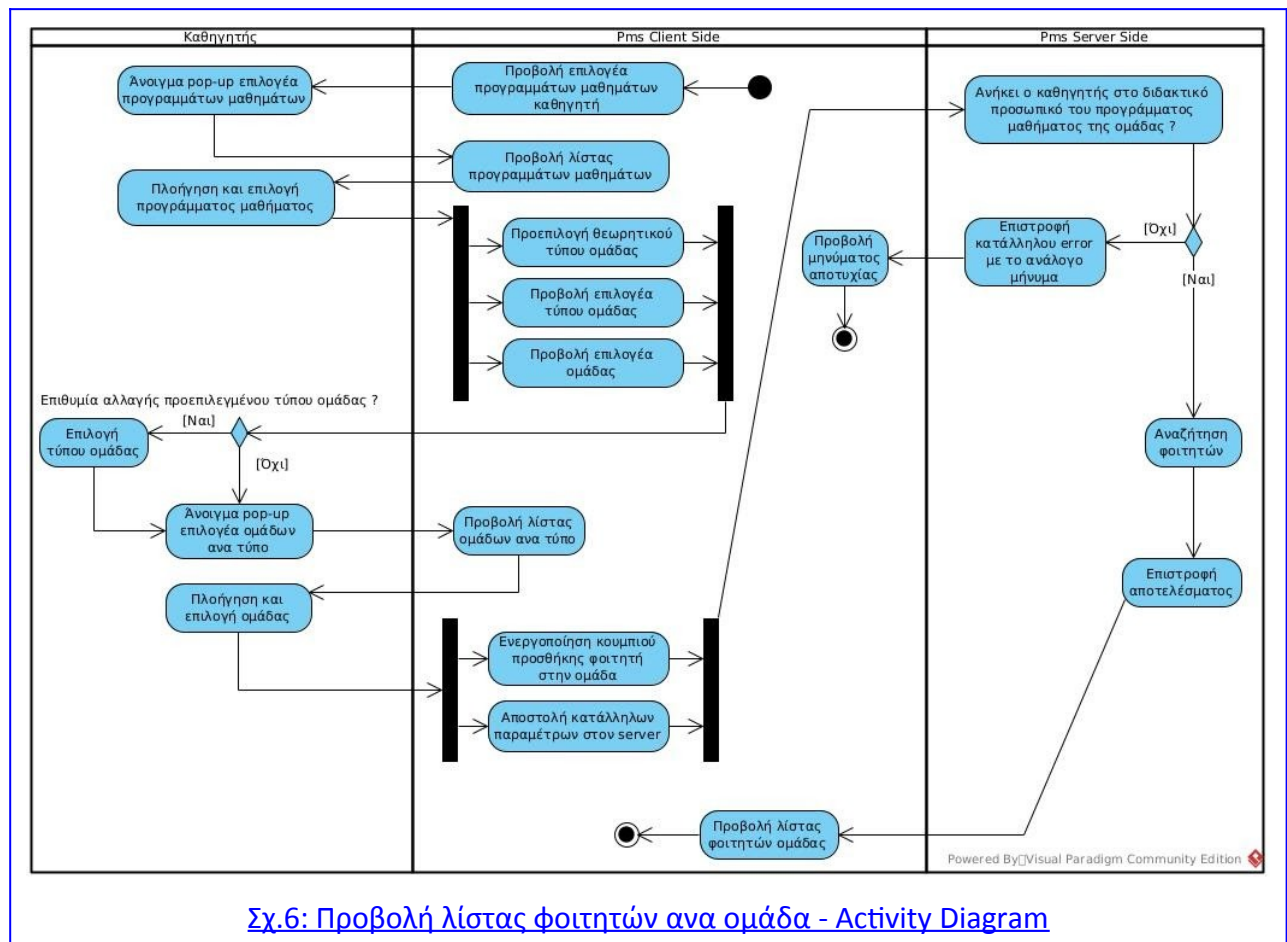
Μετά την αποστολή των δεδομένων στο server side, εκείνο υλοποιεί τους κατάλληλους ελέγχους και ενέργειες για την περάτωση της διαδικασίας και επιστρέφει το κατάλληλο http status. Σε περίπτωση αποτυχίας παρέχει ενημερωτικό μήνυμα. Το client side προβάλλει το κατάλληλο ενημερωτικό μήνυμα ανάλογα με την έκβαση της διαδικασίας.

Ενέργεια του server side που παρουσιάζεται στο διάγραμμα και χρήζει επεξήγησης:

- Έλεγχος για το αν η χωρητικότητα της ομάδας είναι η κατάλληλη:

Η χωρητικότητα θα πρέπει να είναι μεγαλύτερη του αριθμού 5 και μικρότερη του αριθμού 500.

Προβολή λίστας φοιτητών ανα ομάδα



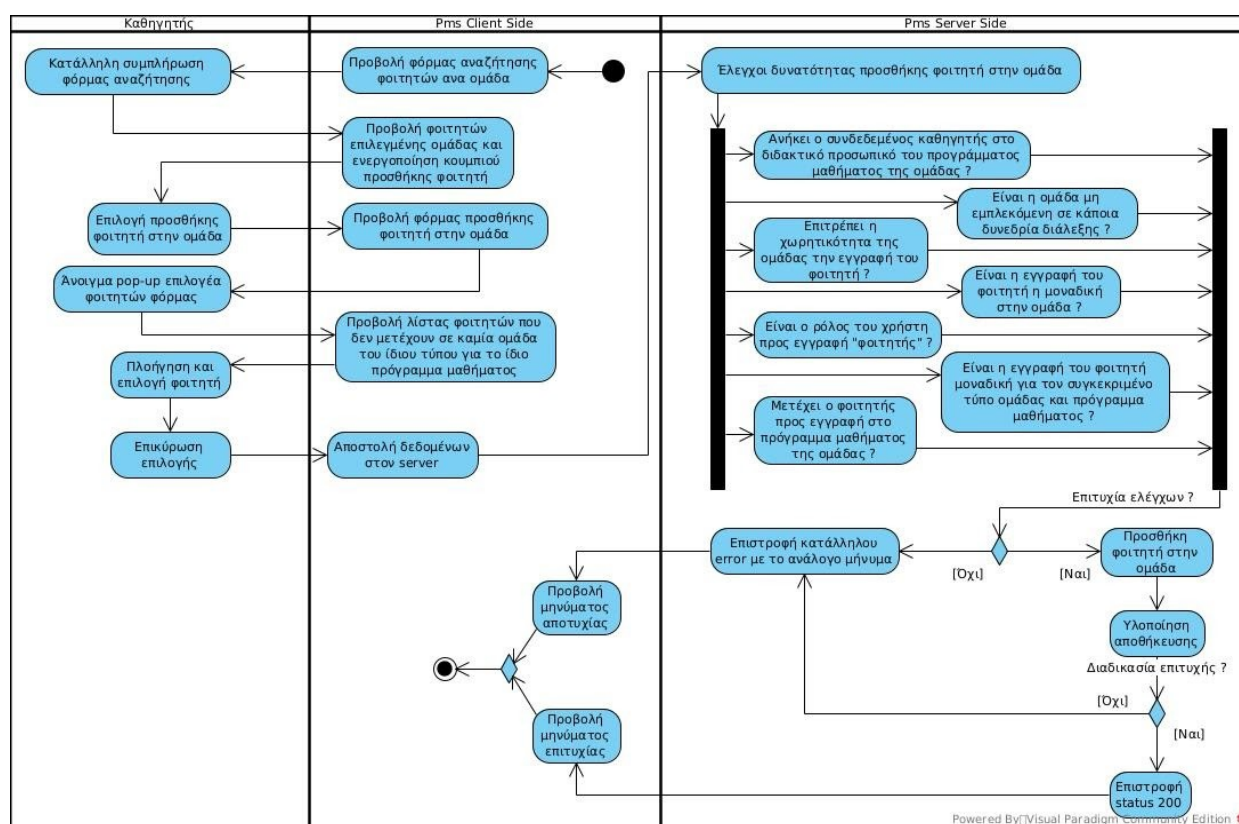
Σχ.6: Προβολή λίστας φοιτητών ανα ομάδα - Activity Diagram

Στο παραπάνω Activity διάγραμμα γίνεται μια εκτενής περιγραφή της διαδικασίας που αφορά την λειτουργία της προβολής της λίστας των φοιτητών ανα ομάδα. Η λειτουργία της προβολής της λίστας των φοιτητών ανα ομάδα αφορά τον ρόλο του καθηγητή, αλλά και του διαχειριστή του συστήματος. Στο διάγραμμα δίνεται ιδιαίτερη έμφαση στην αλληλεπίδραση του χρήστη με την client πλευρά της εφαρμογής με απώτερο στόχο την καλύτερη κατανόηση της λειτουργίας που αναλύεται αλλά και του εύρους των δυνατοτήτων που προσφέρει η client εφαρμογή.

Ο καθηγητής επιλέγει πρώτα πρόγραμμα μαθήματος από τον pop-up επιλογέα (αν είναι διαχειριστής πρέπει να επιλέξει πρώτα τμήμα). Επειτα το client σύστημα του εμφανίζει έναν επιλογέα σχετικό με τον τύπο της ομάδας (θεωρητικός ή εργαστηριακός) και παράλληλα έναν

pop-up επιλογέα για να επιλέξει την ομάδα που θέλει. Αφού γίνει η επιλογή των απαιτούμενων στοιχείων, η client εφαρμογή αποστέλλει τα δεδομένα στον server όπου γίνονται οι σχετικές ενέργειες που περιγράφονται στο διάγραμμα. Σε περίπτωση επιτυχίας η client εφαρμογή προβάλλει στον χρήστη τους φοιτητές της επιλεγμένης ομάδας. Σε περίπτωση αποτυχίας προβάλλει κατάλληλο μήνυμα.

Προσθήκη φοιτητή στην ομάδα



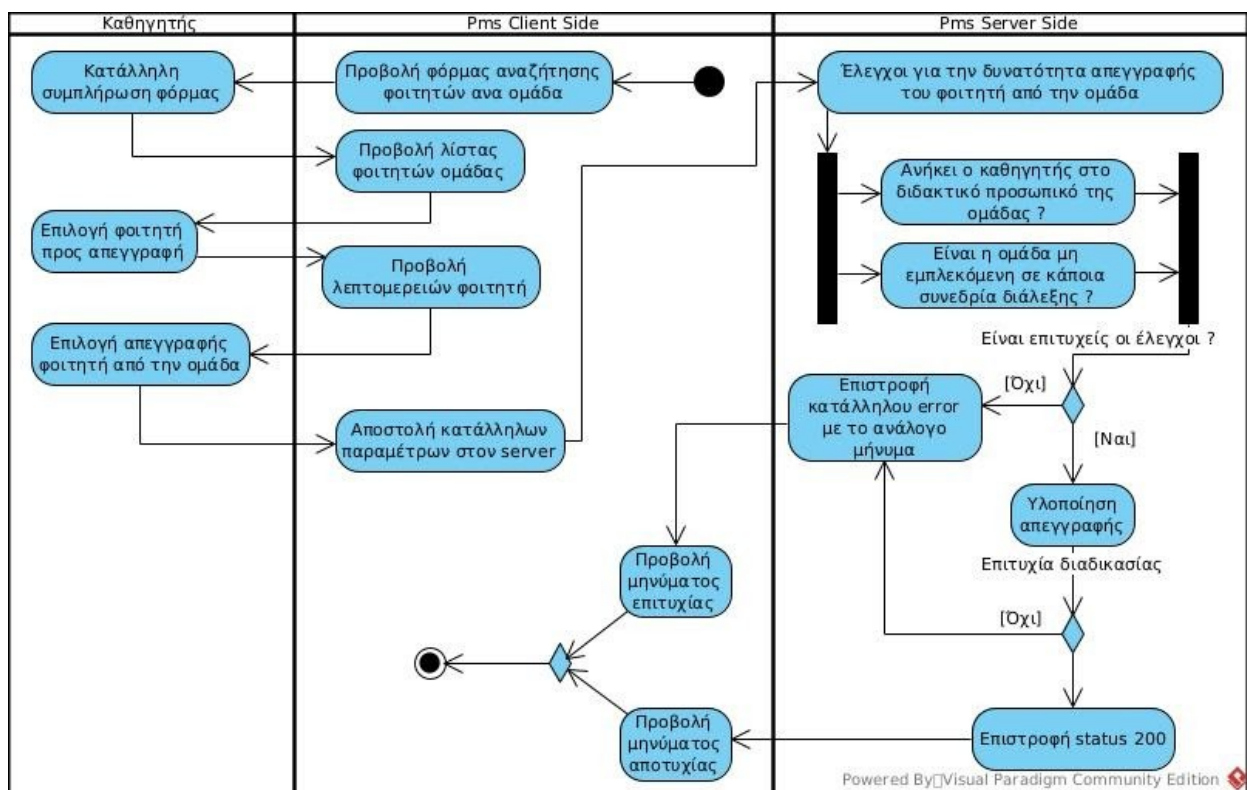
Σχ.7: Προσθήκη φοιτητή στην ομάδα - Activity Diagram

Στο παραπάνω Activity διάγραμμα αναλύεται η λειτουργία της προσθήκης φοιτητή σε ομάδα. Η λειτουργία αυτή επιτελείται από τον καθηγητή αλλά και τον διαχειριστή του συστήματος. Στο διάγραμμα αυτό δίνεται έμφαση στις κρίσιμες ενέργειες που επιτελεί η server πλευρά της εφαρμογής.

Αρχικά ο καθηγητής ή ο διαχειριστής του συστήματος συμπληρώνει τα απαιτούμενα πεδία που του προβάλλει το client side προκειμένου να ενεργοποιηθεί το κουμπί για την προσθήκη του φοιτητή. Επειτα το client side προβάλλει την φόρμα προσθήκης φοιτητή, όπου ο καθηγητής ή ο διαχειριστής συστήματος επιλέγει τον φοιτητή από ένα pop-up με τους φοιτητές του προγράμματος μαθήματος που δεν συμμετέχουν ακόμα σε ομάδα και το client side στέλνει τα δεδομένα στον server.

Ο server πραγματοποιεί σημαντικούς ελέγχους και υλοποιεί την αποθήκευση της εγγραφής. Επειτα στέλνει το κατάλληλο http status στο client side και σε περίπτωση αποτυχίας δημιουργεί κατάλληλο μήνυμα. Το client side με την σειρά του ενημερώνει τον χρήστη με κατάλληλο μήνυμα για την έκβαση της διαδικασίας.

Αφαίρεση φοιτητή από την ομάδα



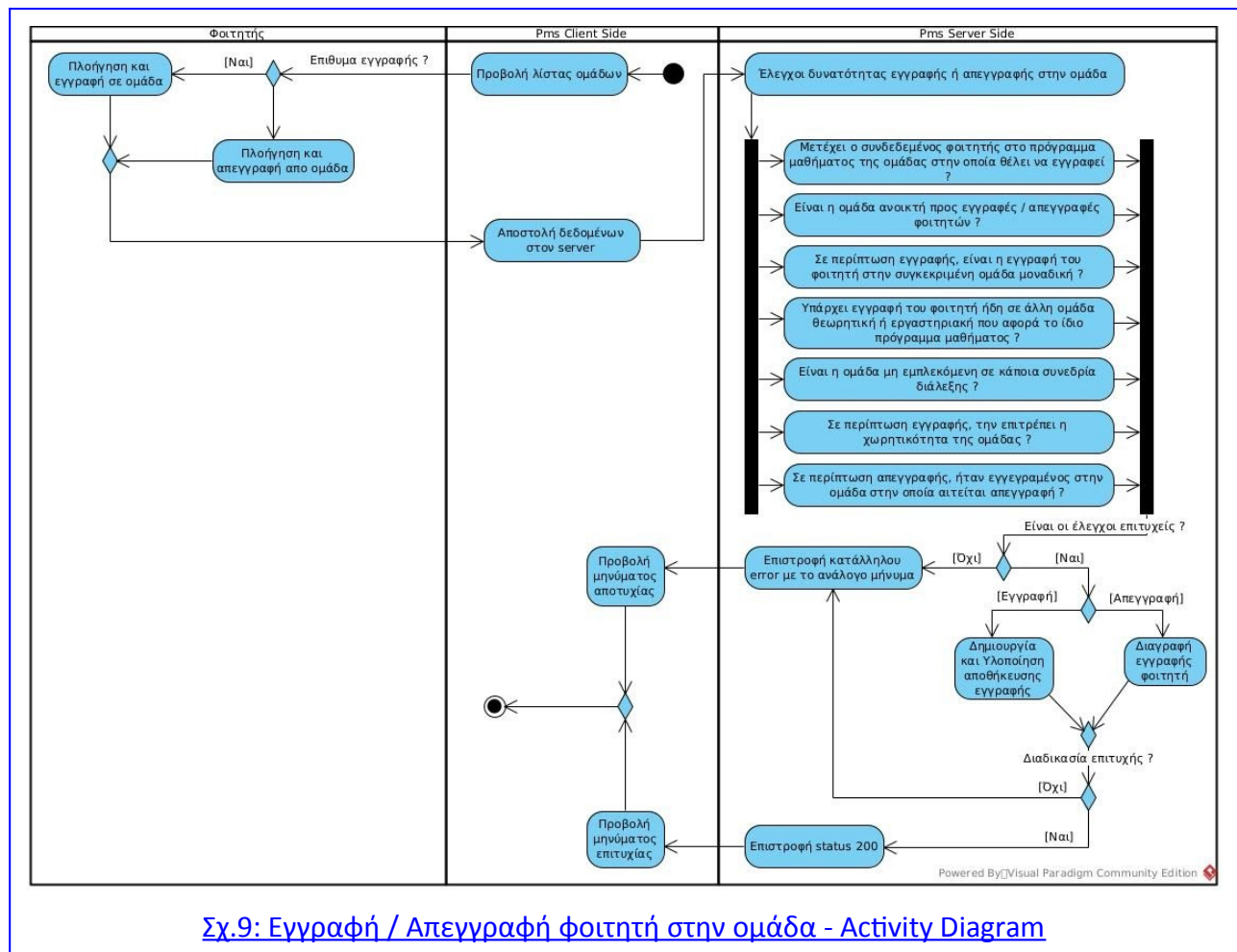
Σχ.8: Αφαίρεση φοιτητή από την ομάδα - Activity Diagram

Στο παραπάνω Activity διάγραμμα αναλύεται η λειτουργία της αφαίρεσης φοιτητή από ομάδα. Η λειτουργία αυτή επιτελείται από τον καθηγητή ή διαχειριστή του συστήματος. Όπως και στο προηγούμενο activity διάγραμμα που αφορούσε την προσθήκη φοιτητή στην ομάδα, έτσι και στο παρόν διάγραμμα δίνεται έμφαση στις κρίσιμες ενέργειες που επιτελεί το server side της εφαρμογής.

Αρχικά ο καθηγητής ή ο διαχειριστής του συστήματος συμπληρώνει κατάλληλα την φόρμα αναζήτησης που το προβάλλει το client side. Επειτα το client side του προβάλλει την λίστα με τους φοιτητές και ο καθηγητής ή ο διαχειριστής επιλέγει τον φοιτητή που θέλει να απεγγραφεί από την ομάδα. Στην συνέχεια το client side προβάλλει τις λεπτομέρειες του φοιτητή προς απεγγραφή και ύστερα ο καθηγητής ή ο διαχειριστής του συστήματος επιλέγει “απεγγραφή από την ομάδα”.

Μετά την αποστολή των κατάλληλων παραμέτρων από το client side, ο server υλοποιεί τις ανάλογες ενέργειες για την διαγραφή του φοιτητή. Επειτα αποστέλλει το κατάλληλο http status στον client ενώ σε περίπτωση αποτυχίας δημιουργεί και ανάλογο μήνυμα. Ο client τέλος προβάλλει μήνυμα σχετικό με την έκβαση της διαδικασίας.

Εγγραφή / Απεγγραφή φοιτητή στην ομάδα



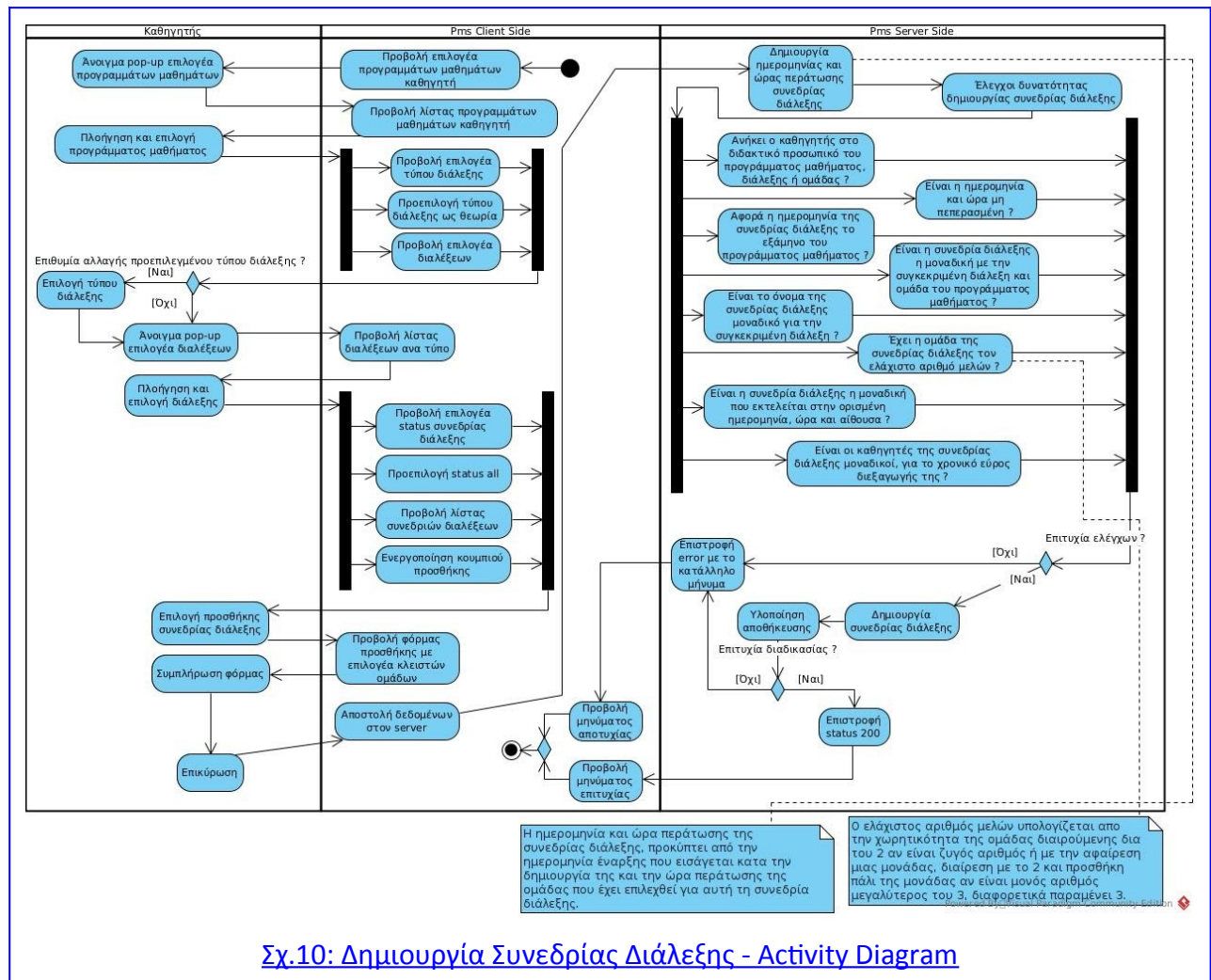
Σχ.9: Εγγραφή / Απεγγραφή φοιτητή στην ομάδα - Activity Diagram

Στο παραπάνω Activity διάγραμμα αναλύεται η λειτουργία της εγγραφής ή απεγγραφής του φοιτητή από την ομάδα. Ο φοιτητής χρησιμοποιώντας κατάλληλα την λειτουργικότητα που του δίνεται από το client side της εφαρμογής, μπορεί να πλοηγηθεί στην λίστα με τις ομάδες του ανάλογου τύπου (θεωρητικού ή εργαστηριακού) και προγράμματος μαθήματος και να εγγραφεί ή να απεγγραφεί σε κάποια από αυτές.

Επειτα το client side αποστέλλει τα δεδομένα που αφορούν την εγγραφή ή απεγγραφή του φοιτητή στο server side του συστήματος. Μετά την αποστολή των δεδομένων στο server side, εκείνο υλοποιεί τους κατάλληλους ελέγχους και ενέργειες για την περάτωση της διαδικασίας

και επιστρέφει το κατάλληλο http status. Σε περίπτωση αποτυχίας παρέχει ενημερωτικό μήνυμα. Το client side προβάλλει ενημερωτικό μήνυμα ανάλογο της έκβασης της διαδικασίας.

Δημιουργία Συνεδρίας Διάλεξης



Σχ.10: Δημιουργία Συνεδρίας Διάλεξης - Activity Diagram

Στο παραπάνω Activity διάγραμμα αναλύεται η λειτουργία της δημιουργίας συνεδρίας διάλεξης. Η λειτουργία αυτή επιτελείται από τον καθηγητή ή τον διαχειριστή του συστήματος. Στο διάγραμμα αυτό, δίνεται έμφαση τόσο στις ενέργειες που αφορούν την λειτουργικότητα του client side αλλά και του server side.

Αρχικά ο καθηγητής επιλέγει από ειδικό pop-up επιλογέα που του προβάλλει το client side, το πρόγραμμα μαθήματος για το οποίο πρόκειται να φτιάξει μια συνεδρία διάλεξης. Σε περίπτωση που ο χρήστης είναι διαχειριστής του συστήματος, θα πρέπει πρώτα να επιλέξει και τμήμα.

Επειτα το client side σύστημα εμφανίζει επιλογέα τύπου διάλεξης και pop-up επιλογέα διάλεξης. Ο καθηγητής ή ο διαχειριστής του συστήματος επιλέγει τύπο διάλεξης (εργαστηριακή ή θεωρητική) καθώς και διάλεξη. Ακολούθως το client side εμφανίζει έναν dropdown επιλογέα επιλογής status της συνεδρίας διάλεξης προεπιλεγμένο στην επιλογή “all” και παράλληλα ενεργοποιεί το κουμπί για την προσθήκη συνεδρίας διάλεξης. Ο καθηγητής ή ο διαχειριστής του συστήματος επιλέγει το κουμπί προσθήκης συνεδρίας διάλεξης και μετά την εμφάνιση της αντίστοιχης φόρμας από το client side, την συμπληρώνει και υποβάλλει τα στοιχεία και το client side στέλνει τα δεδομένα στο server side.

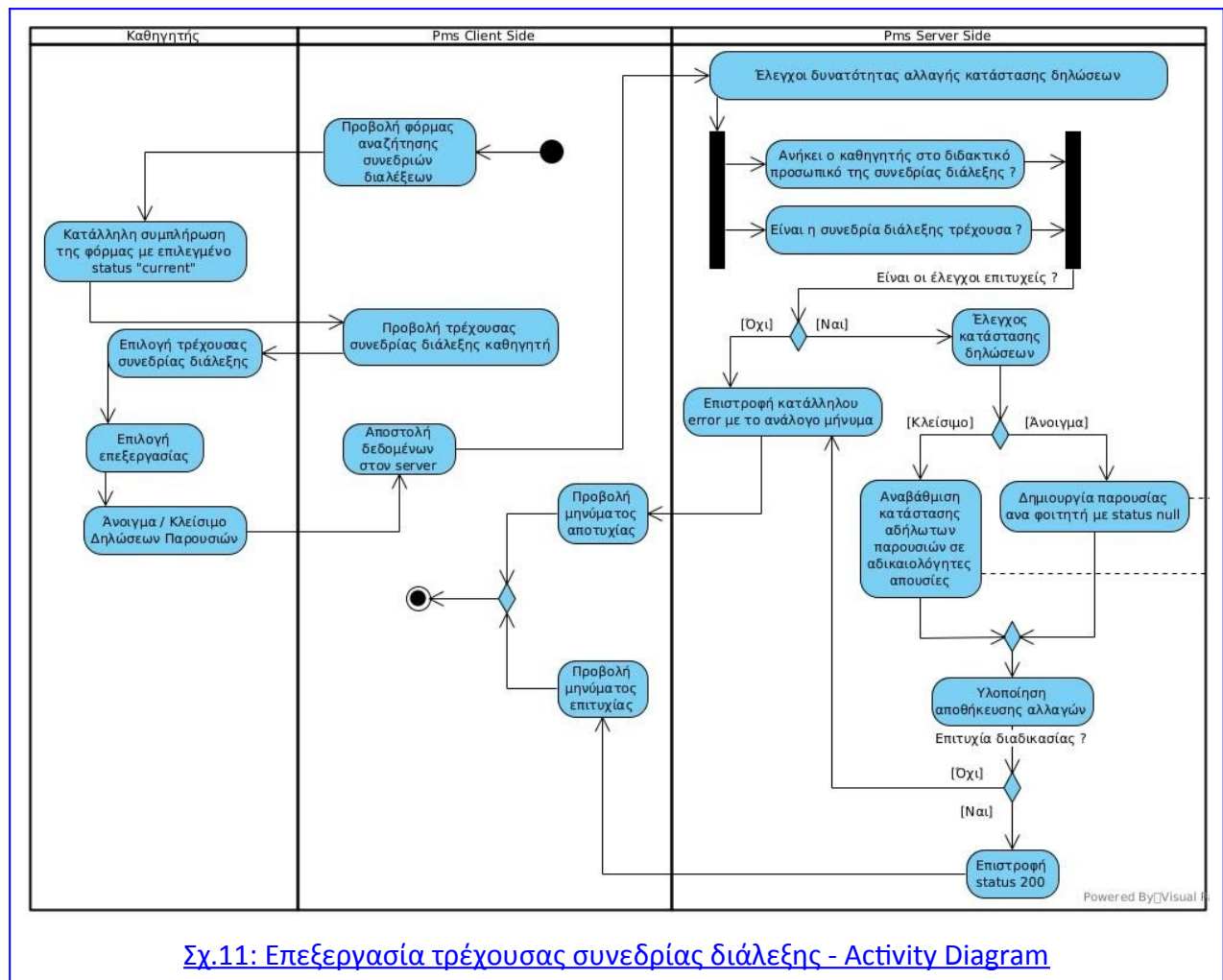
Υστερα από τις παραπάνω ενέργειες το server side επιχειρεί την αποθήκευση της συνεδρίας διάλεξης και στέλνει στο client side το κατάλληλο http status. Αν για κάποιο λόγο κάποια διαδικασία απο τις παραπάνω αποτύχει, το server side δημιουργεί και κατάλληλο μήνυμα. Το client side τέλος, προβάλλει ενημερωτικό μήνυμα ανάλογο της έκβασης της διαδικασίας.

Ενέργειες του server side που παρουσιάζονται στο διάγραμμα και χρήζουν επεξήγησης:

- Έλεγχος για το αν ο καθηγητής ανήκει στο διδακτικό προσωπικό του προγράμματος μαθήματος, διάλεξης ή ομάδας που αφορούν την προς δημιουργία συνεδρία διάλεξης: Σε περίπτωση που ο χρήστης είναι διαχειριστής του συστήματος ο έλεγχος παραλείπεται.
- Έλεγχος για το αν το όνομα της προς δημιουργία συνεδρίας διάλεξης είναι μοναδικό ως προς την διάλεξη που αφορά: Θα πρέπει για κάθε διάλεξη να αντιστοιχούν συνεδρίες της διάλεξης με μοναδική ονομασία.
- Έλεγχος για το αν η ομάδα της προς δημιουργία συνεδρίας διάλεξης έχει τον ελάχιστο αριθμό μελών: Ο ελάχιστος αριθμός μελών της ομάδας υπολογίζεται από το client side με βάση την χωρητικότητά της. Αν η χωρητικότητα της ομάδας είναι ίση με 5 άτομα, τότε ο ελάχιστος αριθμός μελών είναι 5, διαφορετικά αν η χωρητικότητα της ομάδας είναι μεγαλύτερη των 5 ατόμων ο υπολογισμός του ελαχίστου αριθμού μελών γίνεται ως εξής:

- Αν η χωρητικότητα της ομάδας είναι μονός αριθμός, ο ελάχιστος αριθμός μελών υπολογίζεται με την αφαίρεση του αριθμού ένα από την χωρητικότητα της ομάδας, διαίρεση με το δύο και προσθήκη του αριθμού ένα στο αποτέλεσμα.
- Αν η χωρητικότητα της ομάδας είναι ζυγός αριθμός, ο ελάχιστος αριθμός μελών υπολογίζεται με την διαίρεση της χωρητικότητας της ομάδας δια του δύο.
- Δημιουργία ημερομηνίας και ώρας περάτωσης της συνεδρίας διάλεξης: Η ημερομηνία και ώρα περάτωσης της συνεδρίας διάλεξης, προκύπτει από την ημερομηνία έναρξης που εισάγεται κατά την δημιουργία της και την ώρα περάτωσης της ομάδας που έχει επιλεγεί για την παρακολούθησή της.

Επεξεργασία τρέχουσας συνεδρίας διάλεξης



Σχ.11: Επεξεργασία τρέχουσας συνεδρίας διάλεξης - Activity Diagram

Στο παραπάνω Activity διάγραμμα αναλύεται η λειτουργία της επεξεργασίας τρέχουσας συνεδρίας διάλεξης. Την λειτουργία αυτή επιτελεί ο καθηγητής ή ο διαχειριστής του συστήματος ο οποίος χρησιμοποιώντας κατάλληλα την λειτουργικότητα που του δίνεται από το client side της εφαρμογής μπορεί να αναζητήσει και να επιλέξει την τρέχουσα συνεδρία διάλεξης την οποία θέλει να επεξεργαστεί, ανοίγοντας ή να κλείνοντας τις δηλώσεις παρουσιών που σχετίζονται με αυτή.

Υστερα από την υποβολή της φόρμας στην οποία έγινε το άνοιγμα ή το κλείσιμο των δηλώσεων παρουσιών της τρέχουσας συνεδρίας διάλεξης, το client side, στέλνει τα δεδομένα

στο server side. Μετά την αποστολή των δεδομένων στο server side, εκείνο υλοποιεί τους κατάλληλους ελέγχους και ενέργειες για την περάτωση της διαδικασίας και επιστρέφει το κατάλληλο http status. Σε περίπτωση αποτυχίας παρέχει ενημερωτικό μήνυμα. Το client side προβάλλει ενημερωτικό μήνυμα ανάλογο της έκβασης της διαδικασίας.

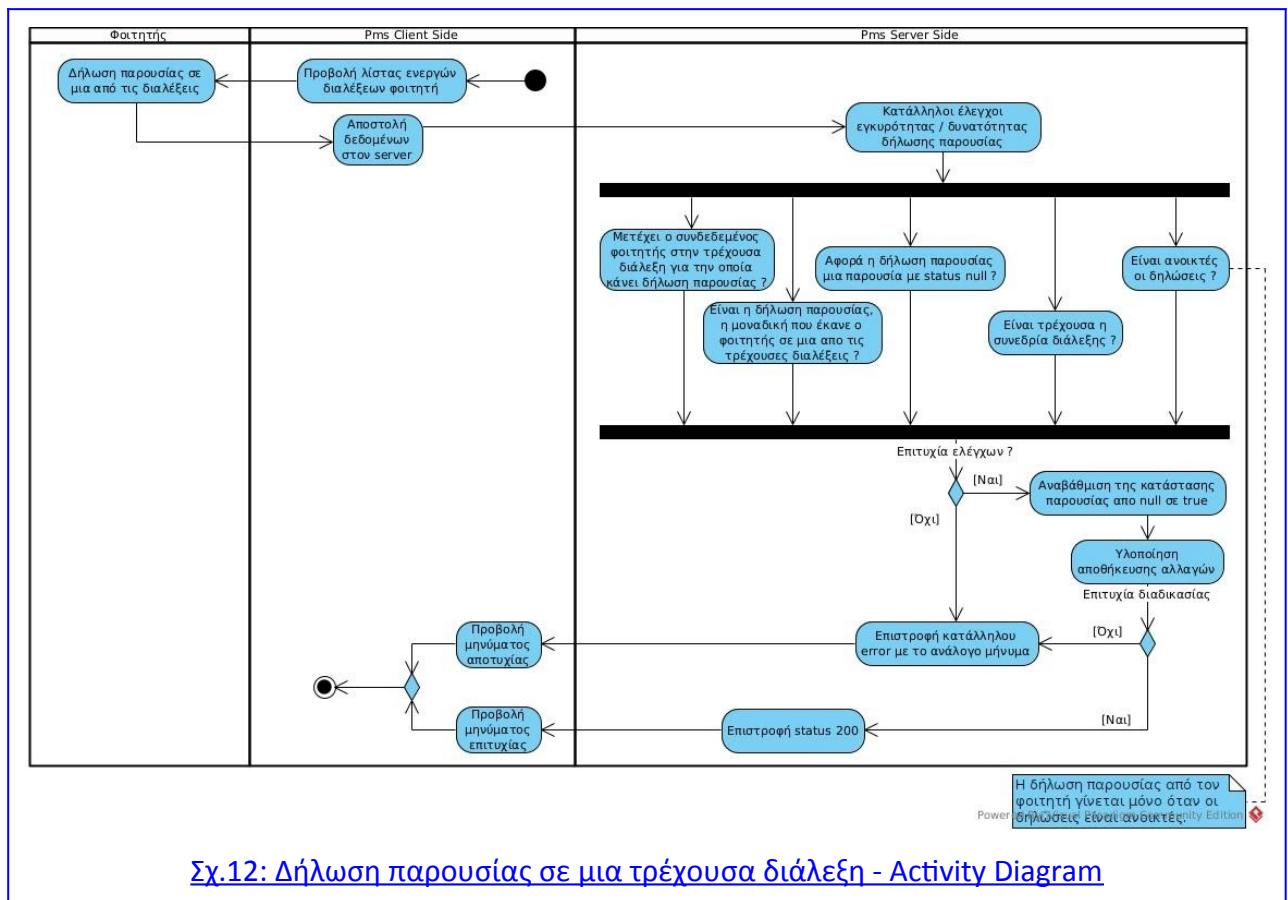
Ενέργειες του server side που παρουσιάζονται στο διάγραμμα και χρήζουν επεξήγησης:

- **Δημιουργία παρουσίας ανα φοιτητή με status null:** Κατά το άνοιγμα των δηλώσεων παρουσιών από τον καθηγητή ή διαχειριστή του συστήματος, το σύστημα δημιουργεί για κάθε φοιτητή που συμμετέχει στην τρέχουσα συνεδρία διάλεξης μια αδήλωτη παρουσία, δηλαδή μια παρουσία με status null.

Αν ο εκάστοτε φοιτητής δηλώσει την παρουσία του σε μια τρέχουσα συνεδρία διάλεξης, τότε το status της αναβαθμίζεται από το σύστημα σε true και ο φοιτητής χαρακτηρίζεται παρών.

- **Αναβάθμιση κατάστασης αδήλωτων παρουσιών σε αδικαιολόγητες απουσίες:** Σε περίπτωση που κάποιος φοιτητής δεν δηλώσει την παρουσία του σε μια τρέχουσα διάλεξη, το σύστημα αναβαθμίζει το status της παρουσίας σε false δηλαδή πλέον χαρακτηρίζεται ως απουσία. Η απουσία αυτή έχει excuse status false και χαρακτηρίζεται με τη σειρά της ως αδικαιολόγητη. Το σύστημα θέτει τον φοιτητή ως απών από την τρέχουσα συνεδρία διάλεξης.

Δήλωση παρουσίας σε μια τρέχουσα διάλεξη



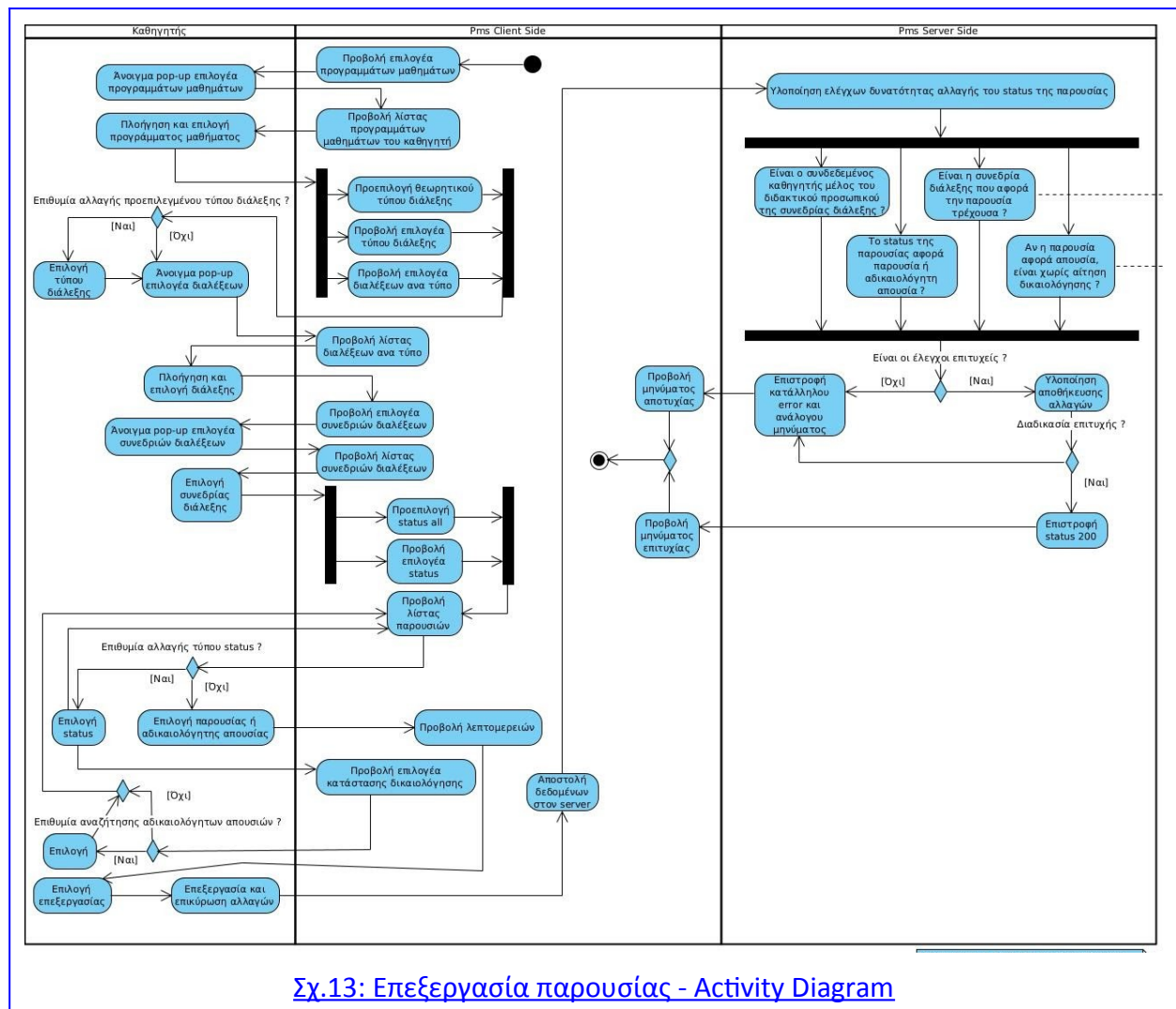
Στο παραπάνω Activity διάγραμμα αναλύεται η λειτουργία της δήλωσης παρουσίας σε μια τρέχουσα συνεδρία διάλεξης. Την λειτουργία αυτή επιτελεί ο φοιτητής ο οποίος χρησιμοποιώντας κατάλληλα την λειτουργικότητα που του δίνεται από το client side της εφαρμογής μπορεί να δηλώσει την παρουσία του σε μια τρέχουσα συνεδρία διάλεξης.

Υστερα από την δήλωση της παρουσίας του φοιτητή στην τρέχουσα συνεδρία διάλεξης, το client side, στέλνει τα δεδομένα στο server side. Μετά την αποστολή των δεδομένων στο server side, εκείνο υλοποιεί τους κατάλληλους ελέγχους και ενέργειες για την περάτωση της διαδικασίας και επιστρέφει το κατάλληλο http status. Σε περίπτωση αποτυχίας παρέχει ενημερωτικό μήνυμα. Το client side προβάλλει ενημερωτικό μήνυμα ανάλογο της έκβασης της διαδικασίας.

Ενέργεια του server side που παρουσιάζεται στο διάγραμμα και χρήζει επεξήγησης:

- **Ελεγχος για το αν είναι ανοικτές οι δηλώσεις παρουσιών:** Η δήλωση παρουσίας απο τον φοιτητή, γίνεται μόνο όταν οι δηλώσεις είναι ανοικτές. Δηλαδή θα πρέπει ο καθηγητής ή ο διαχειριστής του συστήματος να έχει ανοίξει τις δηλώσεις παρουσιών για την τρέχουσα διάλεξη.

Επεξεργασία παρουσίας



Στο παραπάνω Activity διάγραμμα αναλύεται η λειτουργία της επεξεργασίας παρουσίας. Την λειτουργία αυτή επιτελεί ο καθηγητής ή ο διαχειριστής του συστήματος ο οποίος

χρησιμοποιώντας κατάλληλα την λειτουργικότητα που του δίνεται από το client side της εφαρμογής μπορεί να αναζητήσει και να επεξεργαστεί την παρουσία ενός φοιτητή για μια συνεδρία διάλεξης. Υστερα απο την επεξεργασία της παρουσίας του φοιτητή απο τον καθηγητή ή τον διαχειριστή του συστήματος, το client side, στέλνει τα δεδομένα στο server side.

Μετά την αποστολή των δεδομένων στο server side, εκείνο υλοποιεί τους κατάλληλους ελέγχους και ενέργειες για την περάτωση της διαδικασίας και επιστρέφει το κατάλληλο http status. Σε περίπτωση αποτυχίας παρέχει ενημερωτικό μήνυμα. Το client side προβάλλει ενημερωτικό μήνυμα ανάλογο της έκβασης της διαδικασίας.

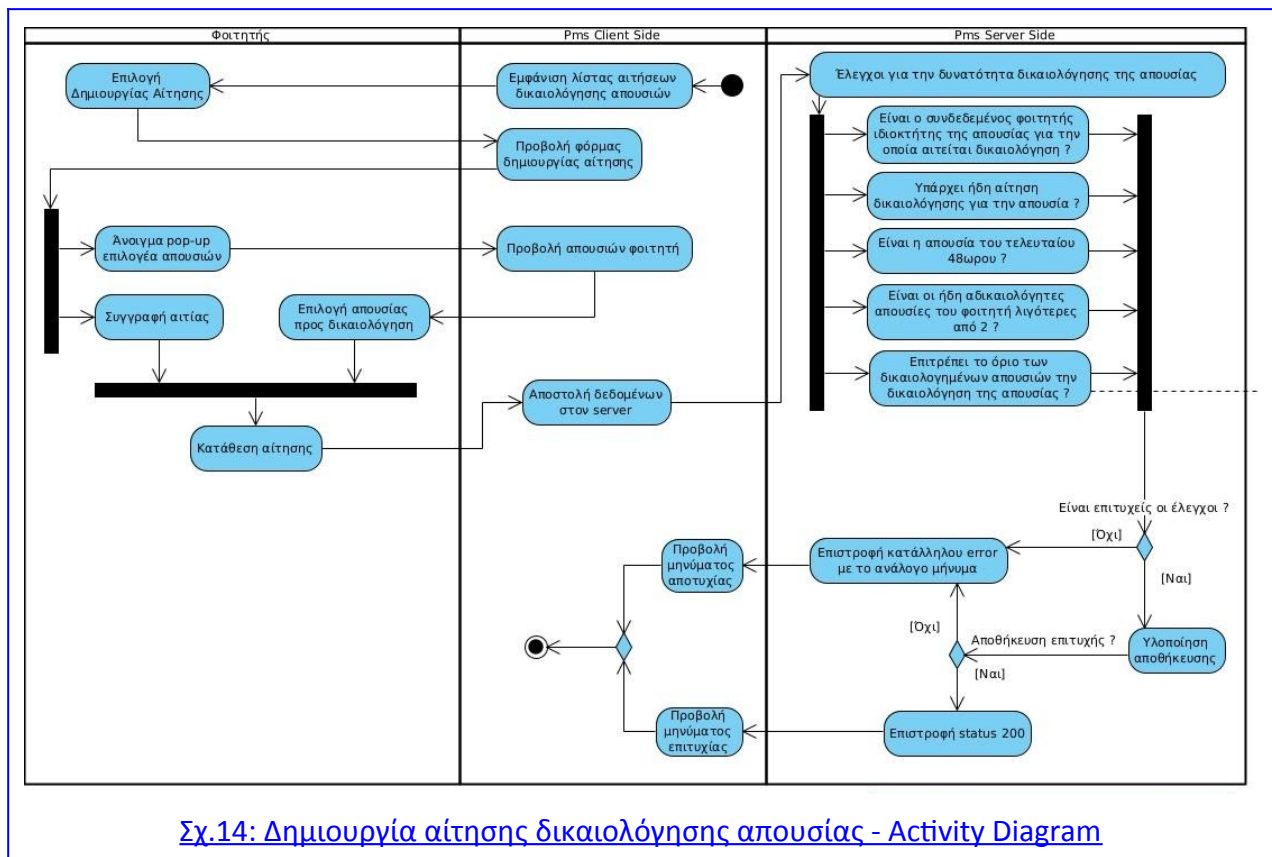
Ενέργειες του server side που παρουσιάζονται στο διάγραμμα και χρήζουν επεξήγησης:

- **Ελεγχος για το αν η συνεδρία διάλεξης που αφορά την παρουσία είναι τρέχουσα:** Για να μπορεί ο καθηγητής ή ο διαχειριστής του συστήματος να επεξεργαστεί μια παρουσία, πρέπει η παρουσία να απευθύνεται σε μια τρέχουσα συνεδρία διάλεξης.
- **Ελεγχος για το αν η παρουσία που χαρακτηρίζεται ως απουσία είναι χωρίς αίτηση δικαιολόγησης:** Για να μπορεί ο καθηγητής ή ο διαχειριστής του συστήματος να επεξεργαστεί μια απουσία, θα πρέπει να μην έχει γίνει ακόμη αίτηση δικαιολόγησης για εκείνη.

Ενέργεια του καθηγητή ή του διαχειριστή συστήματος που παρουσιάζεται στο διάγραμμα και χρήζει επεξήγησης:

- **Επιλογή παρουσίας ή αδικαιολόγητης απουσίας:** Μόνο οι παρουσίες ή οι αδικαιολόγητες απουσίες μπορούν να τεθούν υπο επεξεργασία. Δηλαδή ο καθηγητής ή ο διαχειριστής του συστήματος μπορεί να αλλάξει την κατάσταση μιας παρουσίας σε αδικαιολόγητη απουσία και το αντίστροφο. Σε καμία περίπτωση όμως δεν μπορεί να τεθεί προς αλλαγή κατάστασης μια δικαιολογημένη απουσία.

Δημιουργία αίτησης δικαιολόγησης απουσίας



Στο παραπάνω Activity διάγραμμα αναλύεται η λειτουργία της αίτησης δικαιολόγησης απουσίας. Την λειτουργία αυτή επιτελεί ο φοιτητής ο οποίος χρησιμοποιώντας κατάλληλα την λειτουργικότητα που του δίνεται από το client side της εφαρμογής, μπορεί να αναζητήσει, να επιλέξει και να δικαιολογήσει την απουσία των τελευταίων 48 ωρών που τον ενδιαφέρει υποβάλλοντας αίτηση δικαιολόγησης.

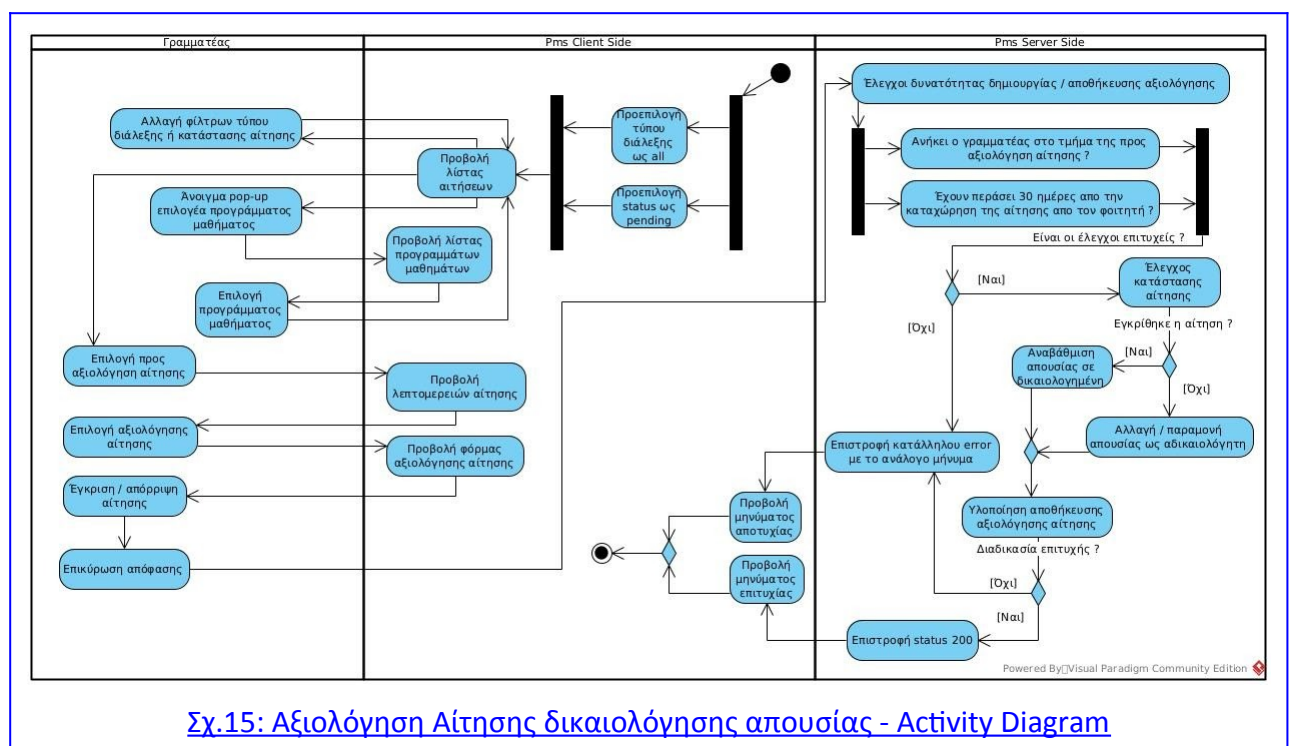
Υστερα από την υποβολή της αίτησης δικαιολόγησης από τον φοιτητή, το client side, στέλνει τα δεδομένα στο server side. Μετά την αποστολή των δεδομένων στο server side, εκείνο υλοποιεί τους κατάλληλους ελέγχους και ενέργειες για την περάτωση της διαδικασίας και επιστρέφει το κατάλληλο http status. Σε περίπτωση αποτυχίας παρέχει ενημερωτικό μήνυμα. Το client side προβάλλει ενημερωτικό μήνυμα ανάλογο της έκβασης της διαδικασίας.

Ενέργεια του server side που παρουσιάζεται στο διάγραμμα και χρήζει επεξήγησης:

- **Έλεγχος για το αν το όριο των δικαιολογημένων απουσιών επιτρέπει την δικαιολόγηση της απουσίας:** Το όριο των δικαιολογημένων απουσιών υπολογίζεται απο το σύστημα δυναμικά με τον εξής αλγόριθμο:
 - Αν ο αριθμός των διαλέξεων (εργαστηριακών ή θεωρητικών) που σχετίζονται με την απουσία είναι ζυγός, τότε οι απουσίες που μπορούν να δικαιολογηθούν είναι ο αριθμός αυτός δια του 2.
 - Αν είναι μονός, είναι ο αριθμός αυτός συν μια μονάδα, έπειτα διαιρούμενος με το 2 και αφαιρουμένης της μονάδας.
 - Αν ο αριθμός των διαλέξεων (εργαστηριακών ή θεωρητικών) είναι ο αριθμός 1, τότε η απουσία που την αφορά μπορεί να δικαιολογηθεί.

Κάτι ακόμα που πρέπει να αναφερθεί είναι ότι και **ο διαχειριστής του συστήματος μπορεί να δημιουργήσει αίτηση δικαιολόγησης**, απλώς με την διαφορά ότι επιλέγει εκείνος ποιον φοιτητή θα αφορά η αίτηση.

Αξιολόγηση αίτησης δικαιολόγησης απουσίας



Σχ.15: Αξιολόγηση Αίτησης δικαιολόγησης απουσίας - Activity Diagram

Στο παραπάνω Activity διάγραμμα αναλύεται η λειτουργία της αξιολόγησης της αίτησης δικαιολόγησης απουσίας. Την λειτουργία αυτή επιτελεί ο γραμματέας ή ο διαχειριστής του συστήματος ο οποίος χρησιμοποιώντας κατάλληλα την λειτουργικότητα που του δίνεται από το client side της εφαρμογής, μπορεί να αναζητήσει, να επιλέξει και να αξιολογήσει μια αίτηση δικαιολόγησης απουσίας μέσα στο χρονικό πλαίσιο 30 ημερών απο την υποβολή της.

Υστερα απο την αξιολόγηση της αίτησης, το client side, στέλνει τα δεδομένα στο server side. Μετά την αποστολή των δεδομένων στο server side, εκείνο υλοποιεί τους κατάλληλους ελέγχους και ενέργειες για την περάτωση της διαδικασίας και επιστρέφει το κατάλληλο http status. Σε περίπτωση αποτυχίας παρέχει ενημερωτικό μήνυμα. Το client side προβάλλει ενημερωτικό μήνυμα ανάλογο της έκβασης της διαδικασίας.

Ενέργεια του server side που παρουσιάζεται στο διάγραμμα και χρήζει επεξήγησης:

- **Αλλαγή / Παραμονή απουσίας ως αδικαιολόγητη:** Αν ο γραμματέας ή ο διαχειριστής του συστήματος απορρίψει την αίτηση δικαιολόγησης, τότε η απουσία παραμένει αδικαιολόγητη.

Επίσης ο γραμματέας ή ο διαχειριστής του συστήματος μπορεί να αλλάξει την πορεία της αξιολόγησής του εντός του χρονικού πλαισίου των 30 ημερών απο την υποβολή της. Έτσι για παράδειγμα, μπορεί μια αίτηση που είχε εγκριθεί προηγουμένως καταλάθος, στην συνέχεια να απορριφθεί. Αυτό σημαίνει ότι η δικαιολογημένη απουσία θα αλλάξει σε αδικαιολόγητη.

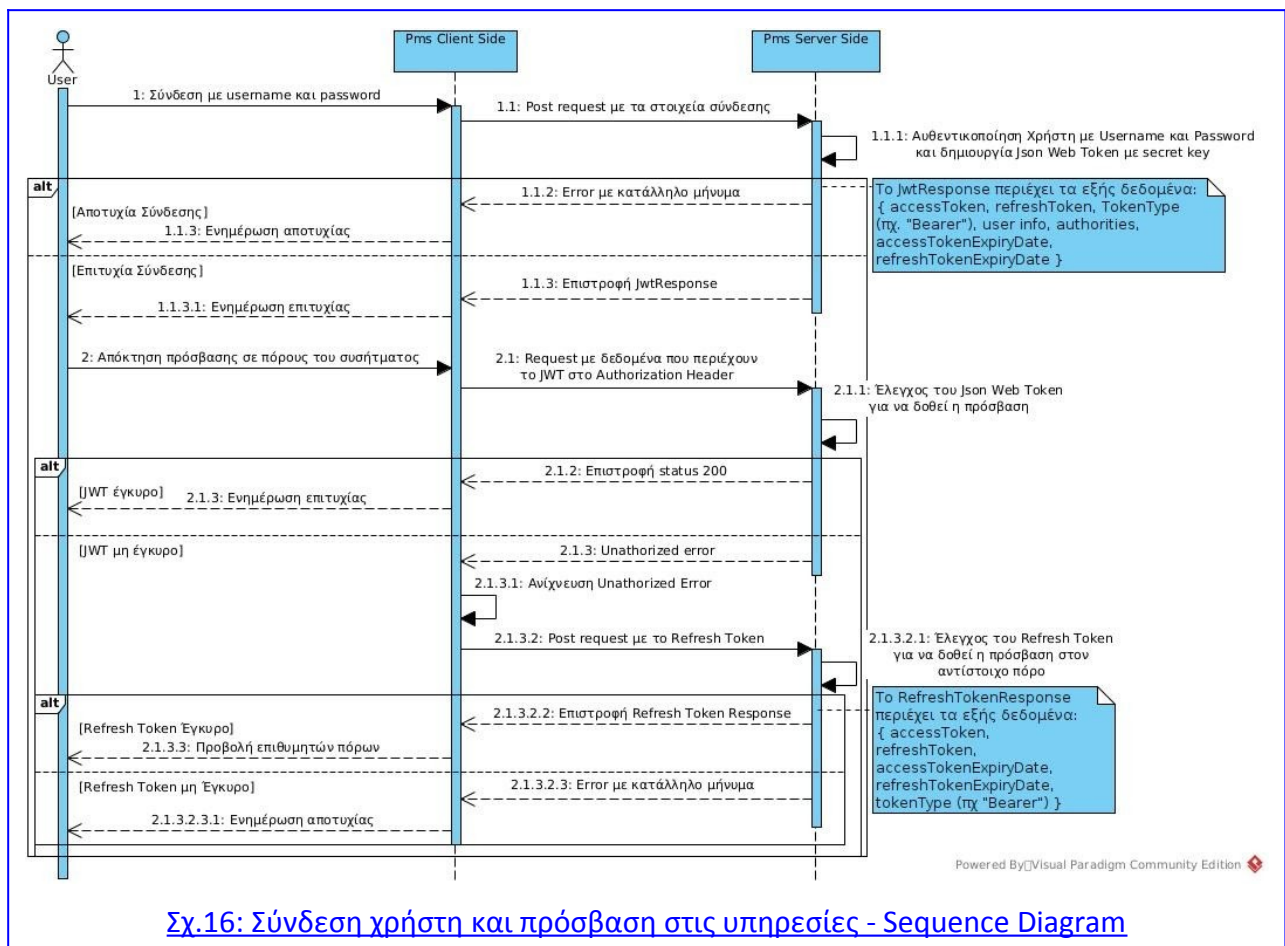
2.3.4. Sequence Diagrams

Στην ενότητα αυτή παρατίθενται τα Sequence Diagrams. Τα Sequence Diagrams που παρατίθενται παρακάτω, συμβάλλουν στην ανάλυση πιο σύνθετων λειτουργιών ή συνδιασμών τους, καθώς και σημαντικών πτυχών του συστήματος διαχείρισης παρουσιών.

Ειδικότερα τα διαγράμματα αυτά εστιάζουν στην αλληλεπίδραση και την επικοινωνία μεταξύ των ρόλων, του client side και του server side του συστήματος με χρονολογική σειρά. Είναι σημαντικό να αναφερθεί ότι οι ενέργειες που επιτελεί ο ρόλος του καθηγητή ή του γραμματέα

στα παρακάτω Sequence διαγράμματα αλλά και του φοιτητή στο διάγραμμα “Δικαιολόγηση απουσίας”, επιτελούνται και απο τον διαχειριστή του συστήματος.

Σύνδεση χρήστη και πρόσβαση στις υπηρεσίες



Σχ.16: Σύνδεση χρήστη και πρόσβαση στις υπηρεσίες - Sequence Diagram

Το παραπάνω Sequence διάγραμμα αναλύει το πως γίνεται η σύνδεση του κάθε χρήστη οποιουδήποτε ρόλου στο σύστημα και η πρόσβασή του στις υπηρεσίες του συστήματος. Ο χρήστης υποβάλλοντας το username και το password του στο client side επιχειρεί την είσοδό του στο σύστημα. Το client side στέλνει τα δεδομένα που εισήγαγε ο χρήστης στο server side.

Το server side με την σειρά του υλοποιεί τους κατάλληλους ελέγχους και ενέργειες για την ταυτοποίηση του χρήστη και επιστρέφει το κατάλληλο http status στο client side. Πιο συγκεκριμένα, σε περίπτωση που ο χρήστης αναγνωριστεί επιτυχώς, το server side δημιουργεί και επιστρέφει στο client side ένα JWT για τον χρήστη, διαφορετικά επιστρέφει error με κατάλληλο μήνυμα. Το client side προβάλλει στον χρήστη ενημερωτικό μήνυμα ανάλογο της έκβασης της διαδικασίας.

Μετα την σύνδεσή του ο χρήστης επιχειρεί να αποκτήσει πρόσβαση σε πόρους του συστήματος που απαιτούν την ταυτοποίησή του. Το client side στέλνει ένα Authorization Header με το JWT του χρήστη στο server side.

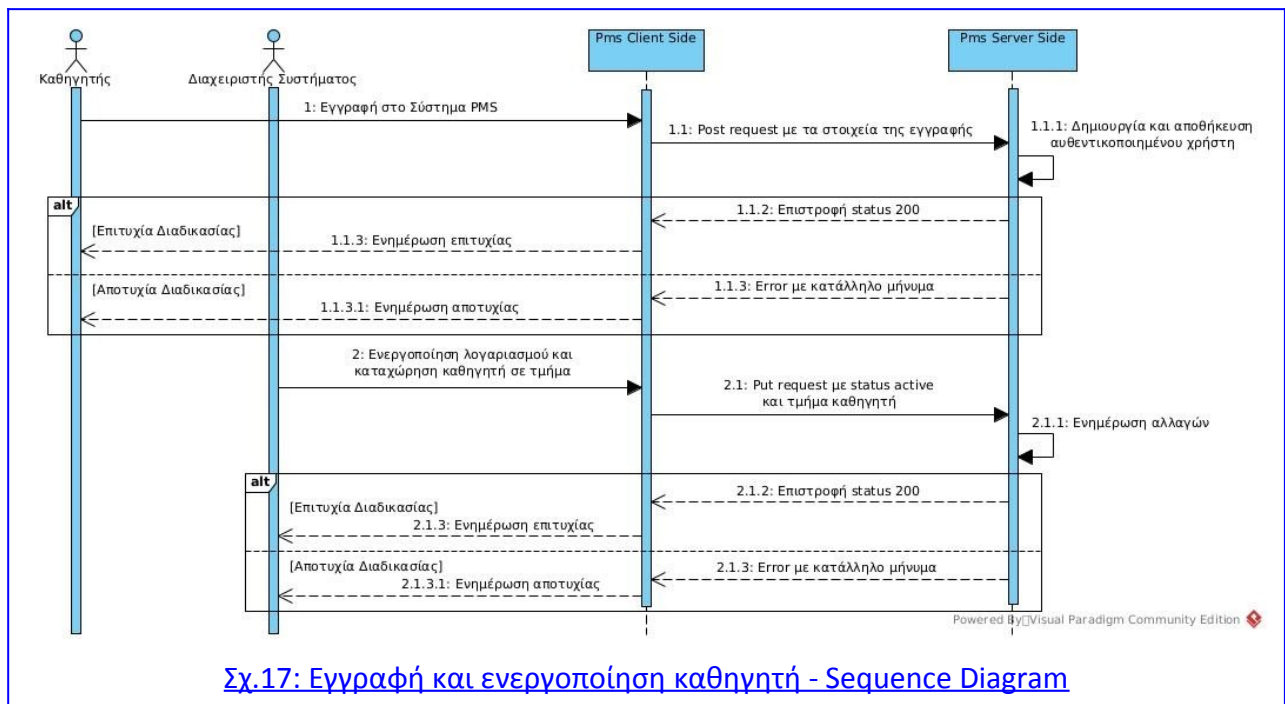
To server side ελέγχει το JWT:

- Σε περίπτωση επιτυχίας δίνει την επιθυμητή πρόσβαση στον χρήστη και επιστρέφει στο client side status 200.
- Σε περίπτωση αποτυχίας, επιστρέφει Unauthorized error στο client side. Το client side με τους κατάλληλους μηχανισμούς ανιχνεύει το error και ξαναστέλνει στο server side το αίτημα πρόσβασης στον πόρο με το refresh token του χρήστη.

To server side ελέγχει το refresh token:

- Αν ο έλεγχος είναι επιτυχής δημιουργεί ένα refresh token response με ανανεωμένα tokens για τον χρήστη και τα επιστρέφει στο client side. Το client side προβάλλει τον επιθυμητό πόρο στον χρήστη.
- Αν ο έλεγχος αποτύχει το server side στέλνει στο client side error με κατάλληλο μήνυμα. Το client side ενημερώνει τον χρήστη για την αποτυχία.

Εγγραφή και ενεργοποίηση καθηγητή



Το παραπάνω Sequence διάγραμμα αναλύει το πως γίνεται η εγγραφή του καθηγητή στο σύστημα και η ενεργοποίησή του απο τον διαχειριστή του συστήματος. Ο καθηγητής υποβάλλοντας τα απαραίτητα στοιχεία της αντίστοιχης φόρμας στο client side επιχειρεί την εγγραφή του στο σύστημα. Το client side στέλνει τα δεδομένα που εισήγαγε ο καθηγητής στο server side.

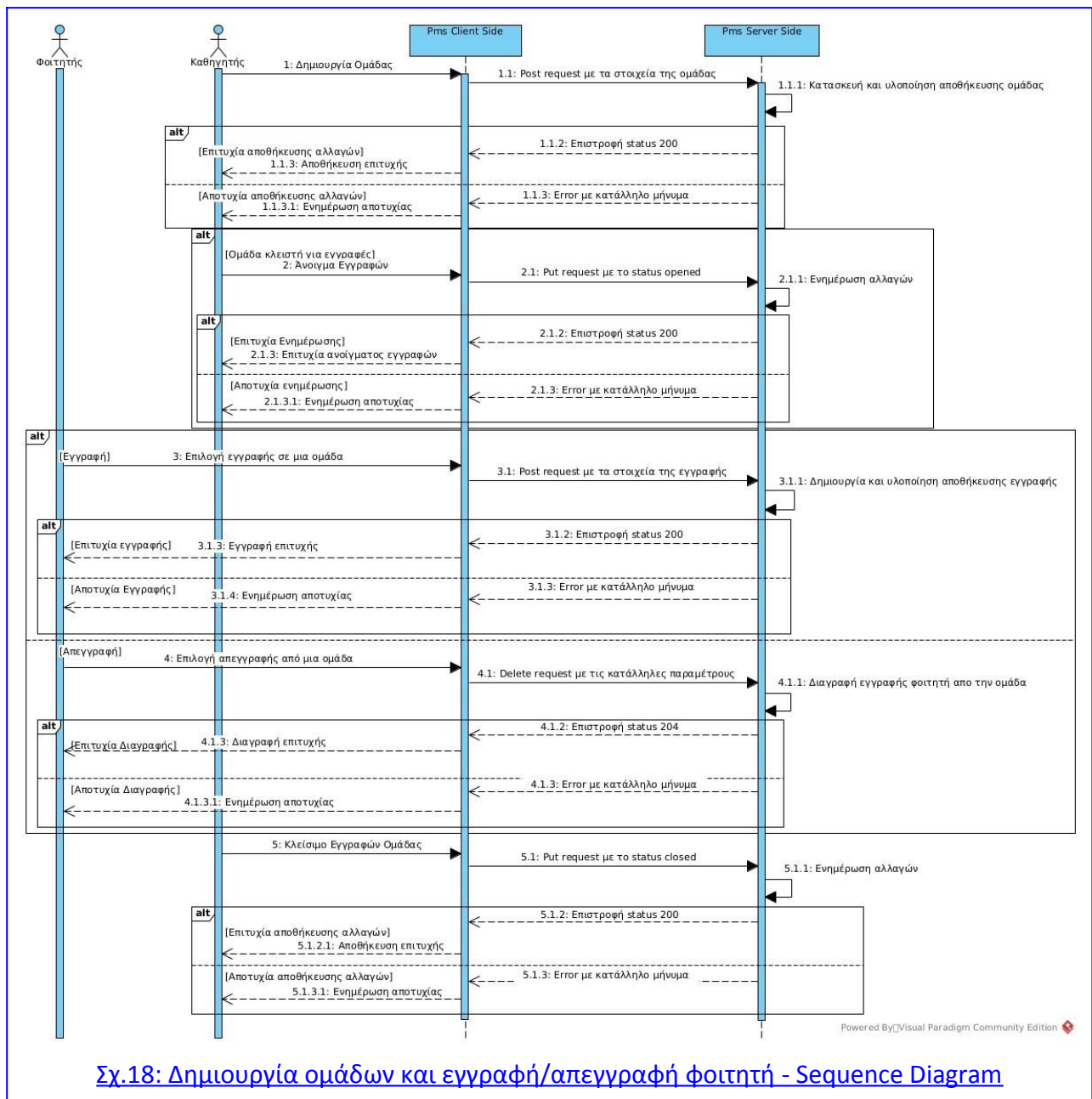
Το server side του συστήματος δημιουργεί και αποθηκεύει τον αυθεντικοποιημένο καθηγητή. Σε περίπτωση αποτυχίας επιστρέφει στο client side http status 200, ενώ αν η διαδικασία αποτύχει επιστρέφει το ανάλογο error με το κατάλληλο μήνυμα. Το client side ενημερώνει τον καθηγητή για την έκβαση της διαδικασίας.

Επειτα ο διαχειριστής του συστήματος πηγαίνοντας στην αντίστοιχη φόρμα επεξεργασίας χρηστών, ενεργοποιεί τον καθηγητή και τον καταχωρεί σε κάποιο τμήμα του πανεπιστημίου. Στην συνέχεια το client side αποστέλλει τα αντίστοιχα δεδομένα των αλλαγών που προέκυψαν

στο server side, το οποίο ενημερώνει το σύστημα για τις αλλαγές και του επιστρέφει http status ανάλογο της έκβασης της διαδικασίας. Σε περίπτωση αποτυχίας το http status συνοδεύεται και απο κατάλληλο μήνυμα.

Τέλος το client side ενημερώνει τον διαχειριστή του συστήματος με μήνυμα ανάλογο της έκβαση της διαδικασίας.

Δημιουργία ομάδων και εγγραφή/απεγγραφή φοιτητή



Το παραπάνω Sequence διάγραμμα αναλύει την διαδικασία δημιουργίας ομάδας φοιτητών απο τον καθηγητή και την εγγραφή ή απεγγραφή τους απο αυτήν.

Σε αυτό το διάγραμμα παρουσιάζεται με χρονική σειρά:

- Η διαδικασία αλληλεπίδρασης του καθηγητή, του client side και του server side σχετικά με:
 - Την δημιουργία ομάδας φοιτητών.
 - Το άνοιγμα εγγραφών των φοιτητών στην ομάδα.
 - Το κλείσιμο των εγγραφών των φοιτητών στην ομάδα.
- Η διαδικασία αλληλεπίδρασης του φοιτητή, του client side και του server side σχετικά με:
 - Την εγγραφή του στην ομάδα.
 - Την απεγγραφή του απο την ομάδα.

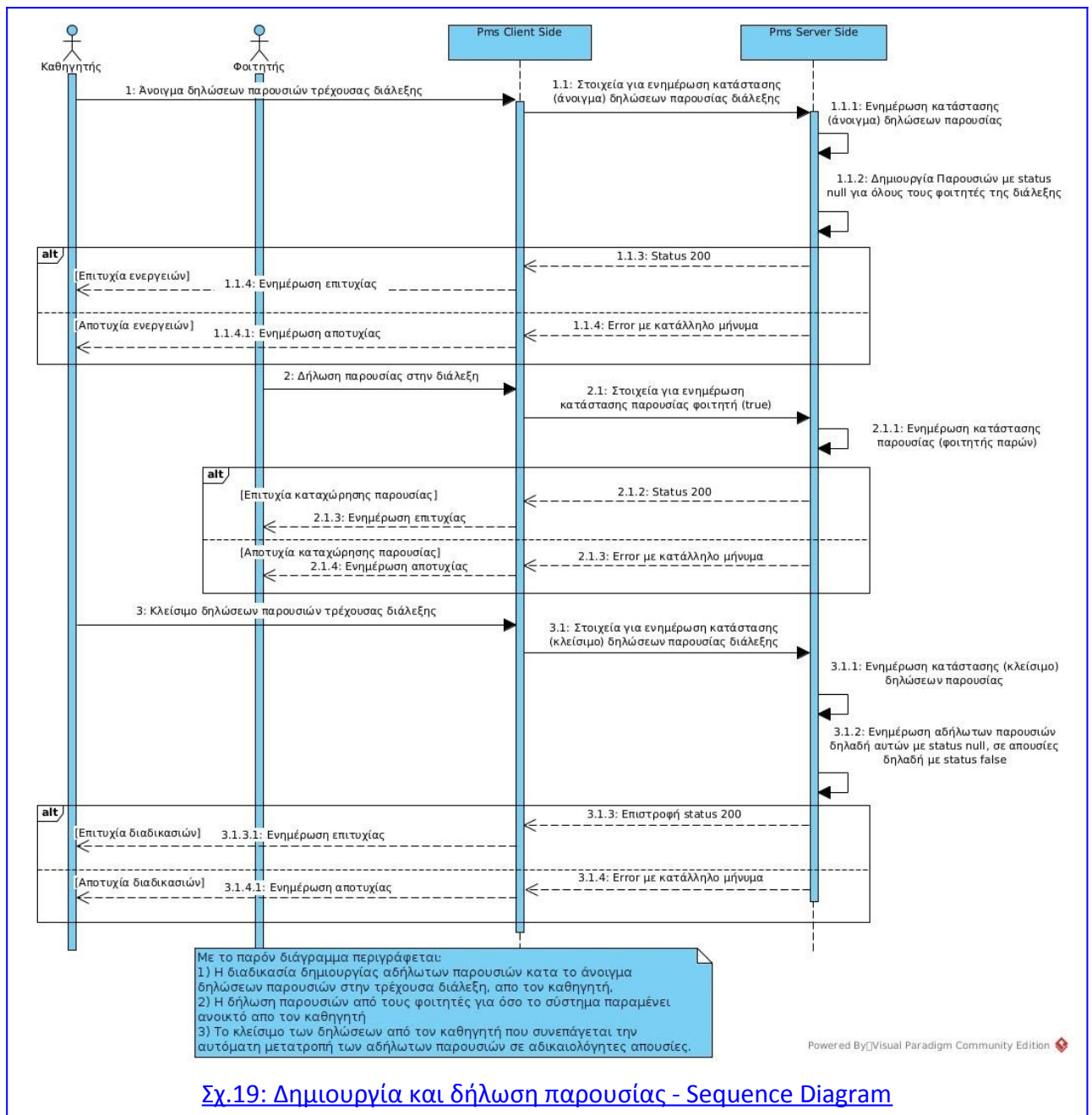
Το παραπάνω διάγραμμα στοχεύει κυρίως στην σαφέστερη προβολή της χρονικής αλληλουχίας μεταξύ των λειτουργιών που αφορούν:

- Την δημιουργία ομάδας απο τον καθηγητή.
- Το ανοίγμα / κλείσιμο των εγγραφών της ομάδας.
- Την εγγραφή ή απεγγραφή των φοιτητών στην ομάδα.

Παράλληλα με το διάγραμμα αυτό γίνεται κατανοητή, εκτός απο την χρονική και η λογική σειρά των γεγονότων που αφορούν τις λειτουργίες αυτές. Γίνεται προφανές λοιπόν, ότι πρώτα ο καθηγητής πρέπει να έχει δημιουργήσει μια ομάδα φοιτητών της οποίας να έχει ανοίξει τις εγγραφές. Επειτα οι φοιτητές μπορούν να εγγραφούν σε μια ομάδα ή να απεγγραφούν απο μια ομάδα. Και στις δύο περιπτώσεις η ομάδα θα πρέπει να είναι ανοικτή προς εγγραφές. Επιπλέον όποτε κρίνεται απαραίτητο ο καθηγητής μπορεί να θέσει την ομάδα κλειστή ως προς τις εγγραφές. Τέλος στο διάγραμμα διαφαίνεται η αλληλεπίδραση των εμπλεκόμενων μερών, ανάλογα με την έκβαση των διαδικασιών που περιγράφονται.

Είναι σημαντικό να τονιστεί μια λεπτομέρεια που δεν απεικονίζεται στο διάγραμμα: Ο καθηγητής μπορεί κατω απο ορισμένες συνθήκες να εγγράψει ή να απεγγράψει φοιτητές σε μια ομάδα, ανεξάρτητα απο το αν οι εγγραφές της ομάδας είναι ανοικτές ή όχι.

Δημιουργία και δήλωση παρουσίας



Το παραπάνω Sequence διάγραμμα αναλύει την διαδικασία δημιουργίας παρουσίας και την δήλωση της παρουσίας από τον φοιτητή.

Σε αυτό το διάγραμμα παρουσιάζεται:

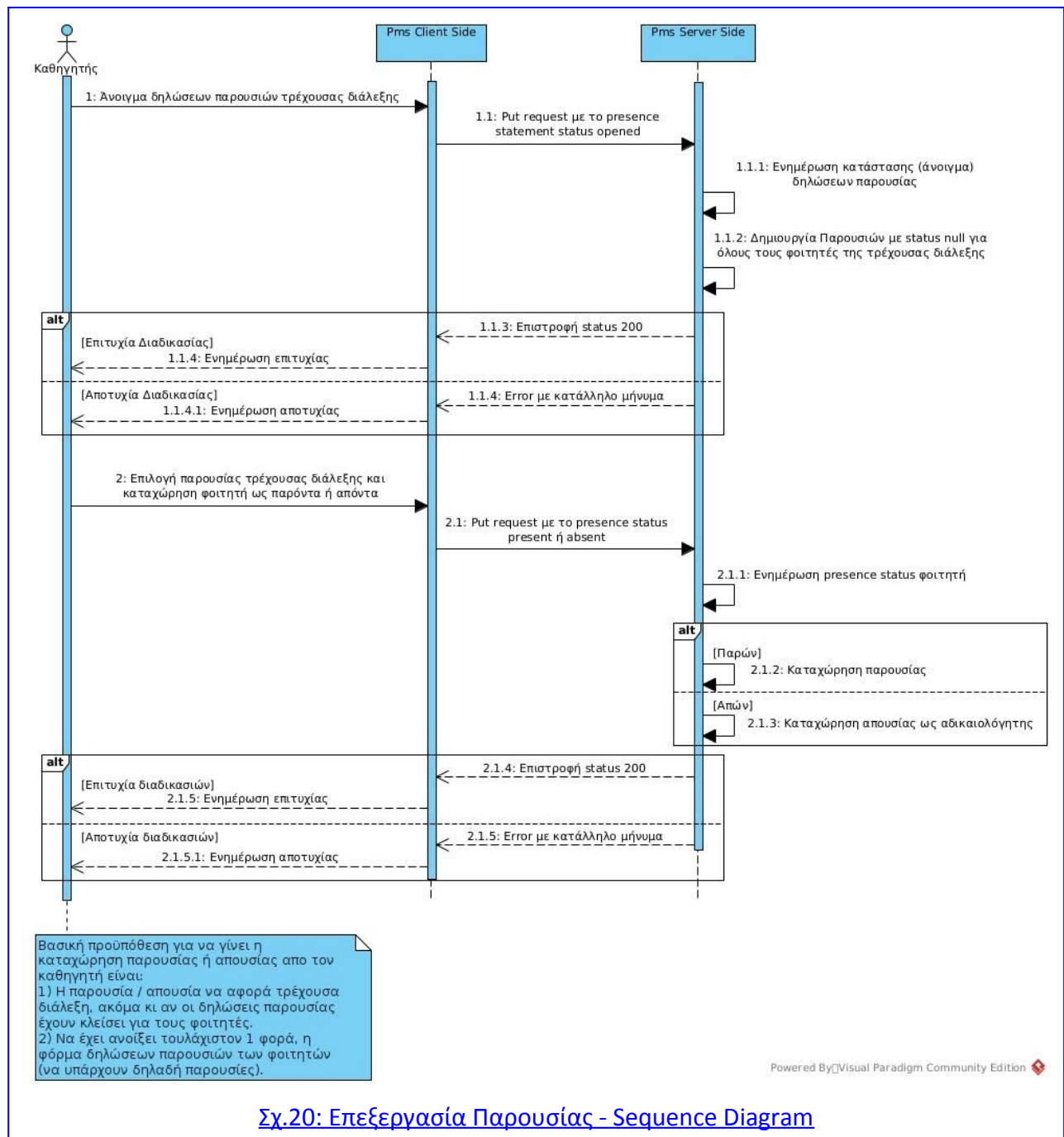
- Η αλληλεπίδραση του καθηγητή, του client side και του server side σχετικά με:
 - Το άνοιγμα των δηλώσεων παρουσιών μιας τρέχουσας συνεδρίας διάλεξης.
 - Το κλείσιμο των δηλώσεων παρουσιών μιας τρέχουσας συνεδρίας διάλεξης.
- Η αλληλεπίδραση του φοιτητή, του client side και του server side σχετικά με την δήλωση παρουσίας σε μια τρέχουσα διάλεξη.

Η χρονική σειρά των κεντρικών λειτουργιών του διαγράμματος είναι η ακόλουθη:

- Άνοιγμα των δηλώσεων παρουσιών στην τρέχουσα διάλεξη από τον καθηγητή και έπειτα δημιουργία αδήλων παρουσιών από το σύστημα.
- Δήλωση παρουσιών από τους φοιτητές για όσο οι δηλώσεις παραμένουν ανοικτές από τον καθηγητή.
- Κλείσιμο των δηλώσεων παρουσιών από τον καθηγητή που συνεπάγεται την αυτόματη μετατροπή των αδήλων παρουσιών σε αδικαιολόγητες απουσίες.

Τέλος στο διάγραμμα διαφαίνεται η αλληλεπίδραση των εμπλεκόμενων μερών, ανάλογα με την έκβαση των διαδικασιών που περιγράφονται.

Επεξεργασία παρουσίας



Σχ.20: Επεξεργασία Παρουσίας - Sequence Diagram

Το παραπάνω Sequence διάγραμμα αναλύει την διαδικασία επεξεργασίας παρουσίας απο τον καθηγητή. Σε αυτό το διάγραμμα παρουσιάζεται η αλληλεπίδραση του καθηγητή, του client side και του server side σχετικά με την επεξεργασία των παρουσιών.

Η χρονική σειρά των κεντρικών λειτουργιών του διαγράμματος είναι η ακόλουθη:

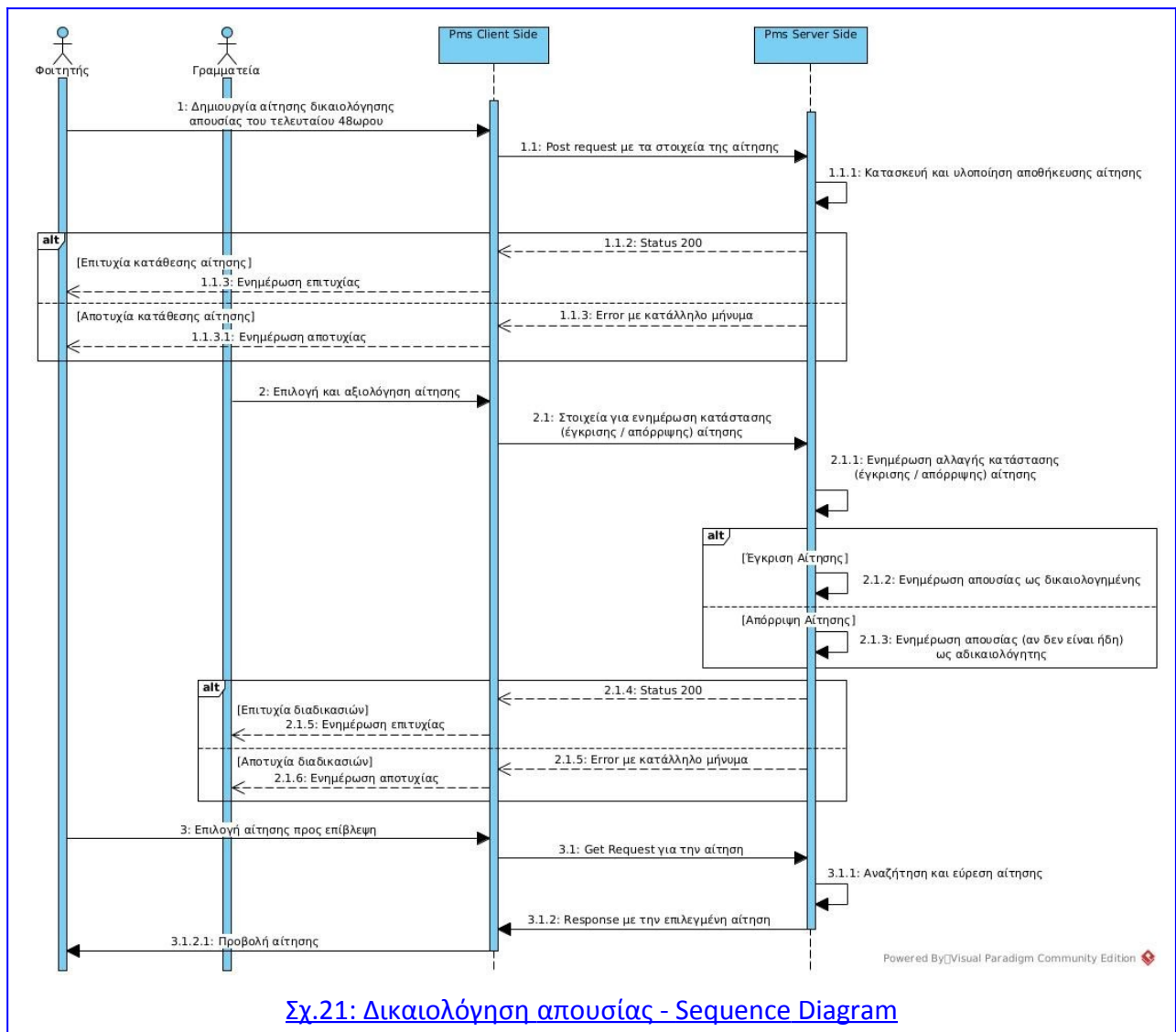
- Ανοιγμα των δηλώσεων παρουσιών στην τρέχουσα διάλεξη, απο τον καθηγητή (ως προαπαιτούμενη διαδικασία για την ύπαρξή παρουσιών) και έπειτα δημιουργία αδήλωτων παρουσιών απο το σύστημα.
- Επιλογή απο τον καθηγητή μιας παρουσίας της τρέχουσας διάλεξης και καταχώρηση του φοιτητή ως παρόντα ή απόντα.
- Ενημέρωση της κατάστασης παρουσίας του φοιτητή απο το σύστημα.

Επιπλέον στο διάγραμμα διαφαίνεται η αλληλεπίδραση των εμπλεκόμενων μερών, ανάλογα με την έκβαση των διαδικασιών που περιγράφονται.

Τέλος, είναι σημαντικό να τονιστεί ότι βασική προϋπόθεση για να γίνει η καταχώρηση παρουσίας ή απουσίας απο τον καθηγητή είναι:

- Η παρουσία / απουσία να αφορά τρέχουσα διάλεξη, ακόμα κι αν οι δηλώσεις παρουσίας έχουν κλείσει για τους φοιτητές.
- Να έχει ανοίξει τουλάχιστον 1 φορά, η φόρμα δηλώσεων παρουσιών των φοιτητών (να υπάρχουν δηλαδή παρουσίες).

Δικαιολόγηση απουσίας



Σχ.21: Δικαιολόγηση απουσίας - Sequence Diagram

Το παραπάνω Sequence διάγραμμα αναλύει την διαδικασία δικαιολόγησης απουσίας.

Σε αυτό το διάγραμμα παρουσιάζεται:

- Η αλληλεπίδραση του γραμματέα, του client side και του server side σχετικά με την αξιολόγηση της αίτησης δικαιολόγησης απουσίας.
- Η αλληλεπίδραση του φοιτητή, του client side και του server side σχετικά με:

- Την δημιουργία αίτησης δικαιολόγησης απουσίας των τελευταίων 48 ωρών.
- Την επίβλεψη της πορείας της αίτησης.

Η χρονική σειρά των κεντρικών λειτουργιών του διαγράμματος είναι η ακόλουθη:

- Δημιουργία αίτησης δικαιολόγησης απουσίας απο τον φοιτητή για μια απουσία του των τελευταίων 48 ωρών.
- Επιλογή και αξιολόγηση, δηλαδή έγκριση ή απόρριψη της αίτησης απο τον γραμματέα.
- Επιλογή της αίτησης προς επίβλεψη απο τον φοιτητή.

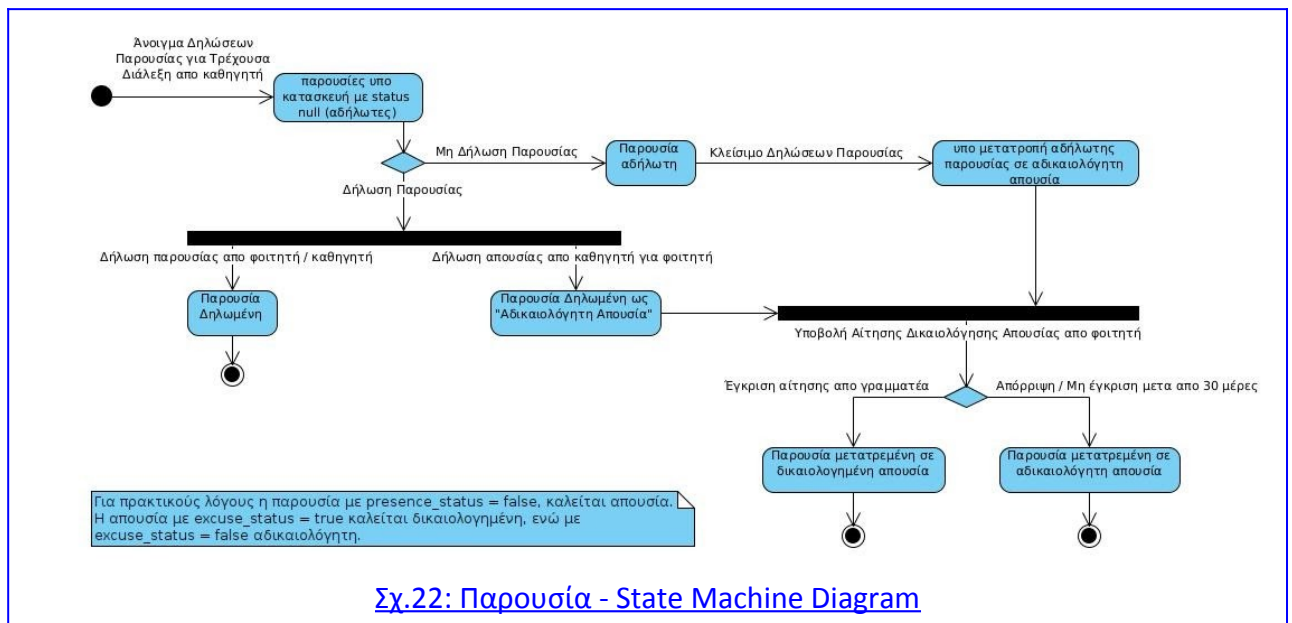
Τέλος στο διάγραμμα διαφαίνεται η αλληλεπίδραση των εμπλεκόμενων μερών, ανάλογα με την έκβαση των διαδικασιών που περιγράφονται.

2.3.5. State Machine Diagrams

Στην ενότητα αυτή παρατίθενται τα State Machine Diagrams. Τα State Machine Diagrams που παρουσιάζονται παρακάτω, συμβάλλουν στην αναλυτική περιγραφή και ανάδειξη των πιο σημαντικών καταστάσεων απο τις οποίες διέρχονται οι κρίσιμες οντότητες του συστήματος διαχείρισης παρουσιών, καθώς και των ενεργειών που τις προκαλούν.

Επίσης είναι αναγκαίο να τονιστεί ότι οι ενέργειες που φαίνονται στα παρακάτω διαγράμματα και αφορούν τον γραμματέα και τον καθηγητή, καθώς και οι ενέργειες του φοιτητή που αφορούν την δημιουργία αίτησης, υλοποιούνται εξίσου και απο τον διαχειριστή του συστήματος.

Παρουσία

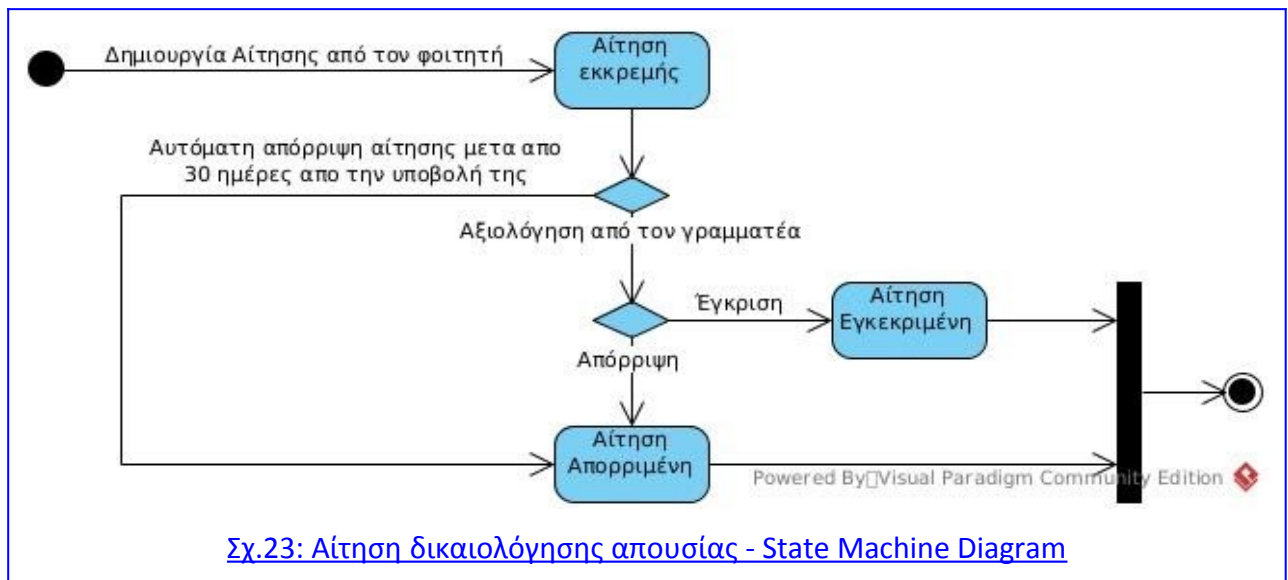


Στο παραπάνω State Machine Diagram αναλύονται οι καταστάσεις απο τις οποίες διέρχεται η κρίσιμη οντότητα της παρουσίας. Φαίνονται επίσης και οι ενέργειες που προκαλούν την εναλλαγή αυτών των καταστάσεων.

Είναι χρήσιμο να τονιστεί ότι για πρακτικούς λόγους:

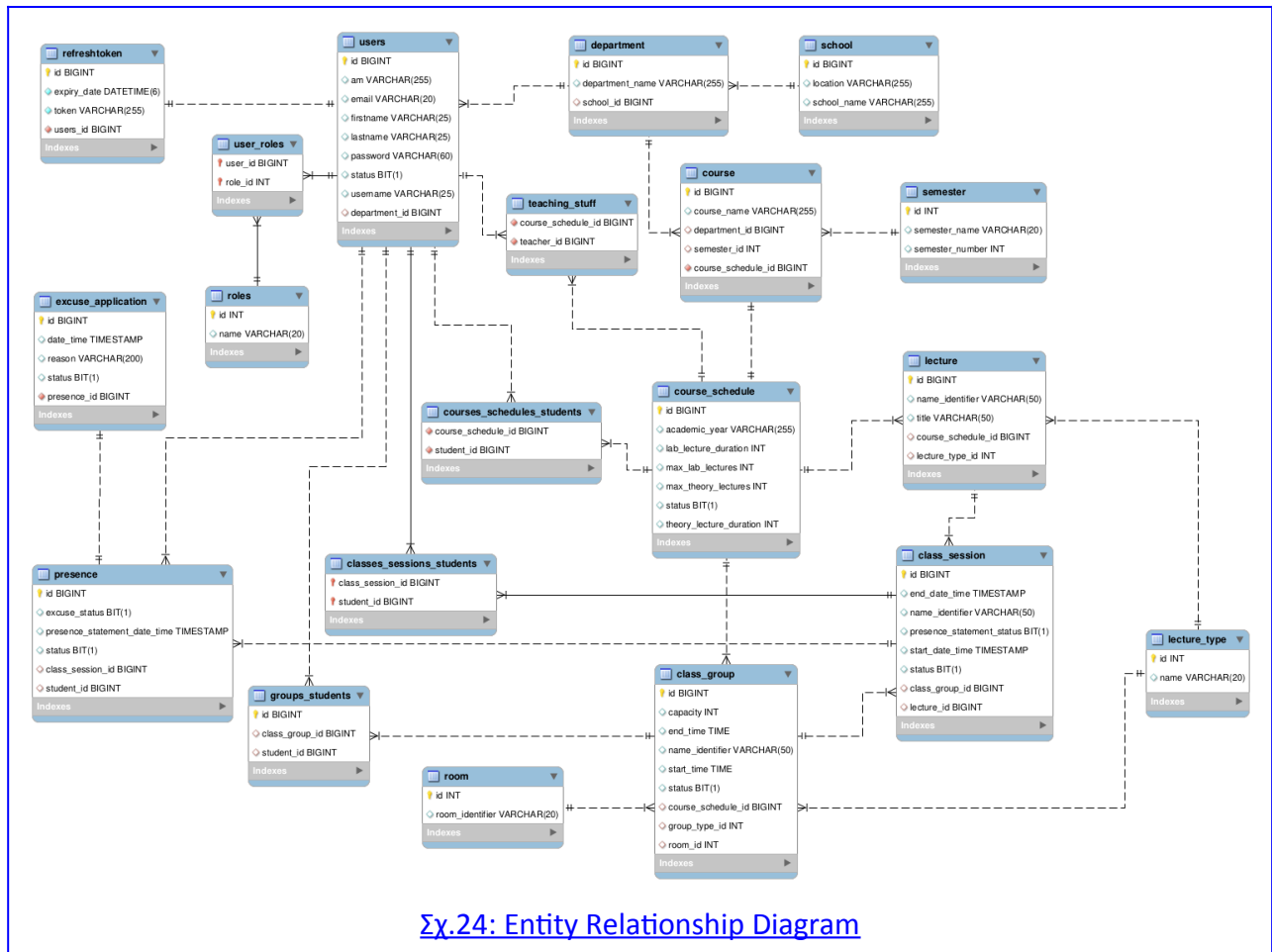
- Η παρουσία με status = false, καλείται απουσία.
- Η απουσία με excuse_status = true, καλείται δικαιολογημένη.
- Η απουσία με excuse_status = false, καλείται αδικαιολόγητη.
- Η παρουσία με status = true, καλείται παρουσία και έχει excuse_status = null.

Αίτηση δικαιολόγησης απουσίας



Στο παραπάνω State Machine Diagram αναλύονται οι καταστάσεις απο τις οποίες διέρχεται η κρίσιμη οντότητα της αίτησης δικαιολόγησης απουσίας. Φαίνονται επίσης και οι ενέργειες που προκαλούν την εναλλαγή αυτών των καταστάσεων.

2.3.6. Entity Relationship Diagram



Σχ.24: Entity Relationship Diagram

Στο παραπάνω Entity Relationship Diagram παρατίθενται αναλυτικά όλοι οι πίνακες της βάσης δεδομένων που αφορούν τις οντότητες του συστήματος διαχείρισης παρουσιών.

Σε κάθε έναν πίνακα απεικονίζονται τα χαρακτηριστικά κάθε οντότητας καθώς και το ρόλο έχουν στην βάση δεδομένων (όπως primary key ή foreign key), με την χρήση κατάλληλων συμβόλων.

Τέλος στο διάγραμμα φαίνονται και οι συσχετίσεις μεταξύ των οντοτήτων των πινάκων της βάσης δεδομένων.

2.3.7. Λειτουργικές και μη Λειτουργικές απαιτήσεις

Το σύστημα διαχείρισης παρουσιών διέπεται τόσο από λειτουργικές, όσο και μη λειτουργικές απαιτήσεις οι κυριότερες από τις οποίες παρατίθενται στην παρούσα ενότητα.

Λειτουργικές απαιτήσεις:

- **Εύχρηστη διεπαφή χρήστη:** Η διεπαφή του χρήστη αναπτύσσεται με το πανίσχυρο **Angular Framework** που παρέχει πληθώρα απο δυνατότητες για την δημιουργία frontend single page εφαρμογών. Επίσης, η δόμησή του user interface, βασίζεται στις αρχές του **Master - Detail Pattern**.
- **Διαχείριση σφαλμάτων (exception handling):** Στο server side του συστήματος, έχει αναπτυχθεί ειδικός μηχανισμός διαχείρισης των exceptions της εφαρμογής. Έτσι η διαχείριση των exceptions γίνεται με τρόπο ευέλικτο ανάλογα με τις εκάστοτε συνθήκες. Ειδικότερα μπορεί να δημιουργηθούν νέα exceptions με συγκεκριμένα ενημερωτικά μηνύματα, ανάλογα με την περίπτωση. Τα exceptions αυτά, μπορεί να τα ανιχνεύσει η server side εφαρμογή με τον ειδικό μηχανισμό που έχει αναπτυχθεί.
- **Παροχή ενημέρωσης στον χρήστη για την έκβαση των διαδικασιών στις οποίες εμπλέκεται:** Το client side του συστήματος διαχείρισης παρουσιών, διαθέτει ειδικό μηχανισμό ανίχνευσης των exceptions απο το server side, αλλά και της επιτυχούς έκβασης των διαδικασιών που καλείται να φέρει σε πέρας το σύστημα στο σύνολό του.
Έτσι ο χρήστης ενημερώνεται για την έκβαση κάθε κρίσιμης ενέργειάς του στο σύστημα, μέσω κατάλληλων ενημερωτικών μηνυμάτων που εμφανίζει το client side στην διεπαφή του.

- **Διαγραφή ή επεξεργασία πόρων του συστήματος με προϋποθέσεις:** Οι λειτουργίες επεξεργασίας ή διαγραφής οντοτήτων στο σύστημα επιτρέπονται μόνο υπό ορισμένες προϋποθέσεις. Για παράδειγμα, μια συνεδρία διάλεξης είναι επεξεργάσιμη όταν ακόμη είναι εκκρεμής (pending) ή τρέχουσα (current). Επίσης μια συνεδρία διάλεξης μπορεί να διαγραφεί μόνο όσο ακόμη είναι εκκρεμής (pending).
- **Περιορισμοί του συστήματος που αφορούν την διεκπεραίωση διαδικασιών:**
Για παράδειγμα, ο φοιτητής μπορεί να απεγγραφεί από μια ομάδα μόνο όσο η ομάδα είναι ανοιχτή προς εγγραφές για τους φοιτητές, ο γραμματέας μπορεί να αξιολογήσει μια αίτηση εντός 30 ημερών μετά την υποβολή της ή ο φοιτητής μπορεί να δικαιολογήσει απουσίες που καταχωρήθηκαν τις τελευταίες 48 ώρες.
Επίσης ο φοιτητής δικαιούται έως δύο οριστικά αδικαιολόγητες απουσίες, ενώ οι δικαιολογημένες απουσίες που του αναλογούν καθορίζονται από ειδικό αλγόριθμο στο σύστημα.

Μη Λειτουργικές απαιτήσεις:

- **Τα αρχεία csv ή excel πρέπει να έχουν συγκεκριμένη μορφή:** Συγκεκριμένα θα πρέπει να έχουν συγκεκριμένες επικεφαλίδες με μια καθορισμένη σειρά.
 - Στην περίπτωση που το csv ή excel αρχείο αφορά τους μαθητές που πρόκειται να καταχωρηθούν στο σύστημα οι επικεφαλίδες πρέπει να είναι όπως παρουσιάζονται παρακάτω και με την εξής σειρά:
AM, First Name, Last Name, Username, Email, Password, Department
 - Στην περίπτωση που το csv ή excel αρχείο αφορά τους φοιτητές που πρόκειται να καταχωρηθούν σε πρόγραμμα μαθήματος, πρέπει να έχει την επικεφαλίδα **AM** που ουσιαστικά αφορά τον αριθμό μητρώου του φοιτητή.
- **Τα στοιχεία του Διαχειριστή Συστήματος (Admin), πρέπει να εισάγονται απευθείας στην βάση δεδομένων, μέσω του server side κατά την εκκίνηση του συστήματος (αν δεν υπάρχει κανένας χρήστης ακόμη), για λόγους ασφαλείας.**
- **Το σύστημα να είναι ασφαλές:** Χρησιμοποιείται Spring Security που σε συνδυασμό με το JWT, παρέχει ασφάλεια στο σύστημα.

- **Γρήγορη αναζήτηση σε μεγάλο όγκο δεδομένων:** Στο σύστημα διαχείρισης παρουσιών, παρέχεται server side σελιδοποίηση, ταξινόμηση και φιλτράρισμα (pagination, sorting, filtering) για τους πίνακες των δεδομένων που παρουσιάζονται στο client side.
- **Γρήγορη απόδοση του συστήματος:** Σε όλη την έκταση της εφαρμογής, απο το client side έως το server side, συνδιάστηκαν οι κατάλληλες τεχνολογίες με τέτοιο τρόπο ώστε να επιφέρουν την μέγιστη δυνατή απόδοση. Ενδεικτικά λειτουργίες όπως το eager ή lazy loading των δεδομένων, χρησιμοποιήθηκαν τόσο στο server side όσο και στο client side της εφαρμογής. Παράλληλα με την χρήση της γλώσσας java, τα ερωτήματα στην βάση δεδομένων γίνονται γρήγορα και αποδοτικά. Επίσης, το server side pagination, sorting και filtering όπως προαναφέρθηκε, συμβάλλει στην ταχύτατη διαχείριση μεγάλου όγκου δεδομένων που σε συνδυασμό με την κατάλληλη δόμηση του client side, ελαφραίνουν σημαντικά το δίκτυο. Τέλος το client side είναι αρμονικά δομημένο ώστε να αξιοποιεί στο μέγιστο τις δυνατότητες που του παρέχονται απο την σχεδίαση του server side, κάνοντας το σύστημα συνολικά αποδοτικό.
- **Ορθή κατανομή του κέντρου βάρους των διαδικασιών μεταξύ του server side και του client side της εφαρμογής:** Η εφαρμογή είναι δομημένη ώστε να επιβαρύνει τον server τόσο όσο πρέπει και το δίκτυο τόσο όσο χρειάζεται. Στα πλαίσια αυτής της απαίτησης υπάρχει ξεχωριστή client side εφαρμογή ανεπτυγμένη σε Angular 12, η οποία τρέχει στην εκάστοτε συσκευή του κάθε τελικού χρήστη και όχι στο server side. Έτσι το server side όντας ελαφρυμένο απο τον όγκο της frontend εφαρμογής, αναλαμβάνει διαδικασίες, όπως η σελιδοποίηση (pagination), η ταξινόμηση (sorting), το φιλτράρισμα (filtering), που αν γίνονταν στο client side θα το επιβάρυναν σημαντικά.

Κεφάλαιο 3: Τεχνολογίες Υλοποίησης της Εφαρμογής

3.1. Τεχνολογίες Server

Στα παρακάτω υποκεφάλαια παρατίθενται οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη του server side της εφαρμογής.

3.1.1. Spring Framework - Spring Boot

Γιατί χρησιμοποιήθηκε το Spring ?

Το Spring κάνει τον προγραμματισμό με την γλώσσα Java γρηγορότερο, ευκολότερο και ασφαλέστερο για τον καθένα. Η εστίαση του Spring στην ταχύτητα, την απλότητα και την παραγωγικότητα το έκανε το δημοφιλέστερο Java framework παγκοσμίως. [1]

“Χρησιμοποιούμε πολλά εργαλεία του Spring Framework και αποκομίζουμε τα οφέλη, του να έχουμε εναλλακτικές λύσεις και να μην ανησυχούμε γράφοντας υπερβολικά πολύ κώδικα, πράγμα που μας γλιτώνει χρόνο και ενέργεια” [1]

Sean Graham, Application Transformation Lead, DICK’s Sporting Goods

Το server side του συστήματος διαχείρισης παρουσιών αναπτύχθηκε με το εργαλείο Spring Boot το οποίο ανήκει στο Spring Framework. Η έκδοση του Spring Boot που χρησιμοποιήθηκε ήταν η 2.4.5.

Spring Framework:

Το Spring Framework είναι λογισμικό ανοικτού κώδικα που παρέχει την δυνατότητα δημιουργίας περιεκτικού κώδικα καθώς και μοντέλα παραμετροποίησης για σύγχρονες επαγγελματικές εφαρμογές βασισμένες σε γλώσσα Java για οποιαδήποτε πλατφόρμα ανάπτυξης.[2]

Πλεονεκτήματα Spring Framework:

- Βασικές τεχνολογίες: dependency injection, events, resources, i18n, validation, data binding, type conversion, SpEL, AOP. [2]
- Testing: mock objects, TestContext framework, Spring MVC Test, WebTestClient. [2]
- Πρόσβαση στα δεδομένα (Data Access): transactions, DAO support, JDBC, ORM, Marshalling XML. [2]
- Spring MVC και Spring WebFlux frameworks. [2]
- Σύνδεση με άλλα υποσυστήματα: remoting, JMS, JCA, JMX, email, tasks, scheduling, cache. [2]
- Γλώσσες: Kotlin, Groovy, dynamic languages. [2]

Spring Boot:

Το Spring Boot είναι ένα εργαλείο ανοικτού κώδικα που διευκολύνει την ανάπτυξη αυτόνομων επαγγελματικών εφαρμογών και microservices με το Spring Framework γρήγορα και αξιόπιστα. Οι εφαρμογές που αναπτύσσονται με Spring Framework τρέχουν σε Java Virtual Machine. Το Spring Boot έχοντας συνοψίσει πολλές από τις λειτουργίες της πλατφόρμας του spring αλλά και άλλων βιβλιοθηκών συμβάλλει στην δημιουργία μεστού και καθαρού κώδικα. [3]

Πλεονεκτήματα Spring Boot:

- Δημιουργία αυτόνομων εφαρμογών. [3]

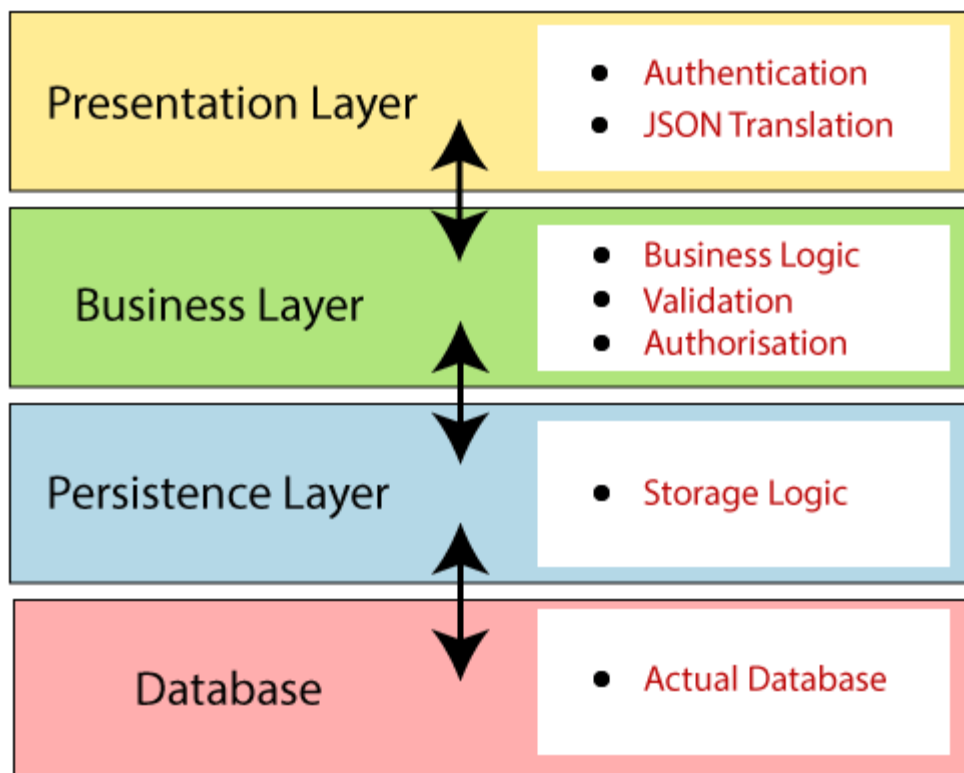
- Απευθείας ενσωματωμένος Tomcat, Jetty ή Undertow χωρίς να χρειάζεται η επεξεργασία WAR αρχείων. [3]
- Παροχή διευκολύνσεων στην αρχικοποίηση της εφαρμογής μέσω του spring initializr.
- Αυτοματοποιημένη παραμετροποίηση για το Spring και για διάφορες βιβλιοθήκες όποτε είναι δυνατό. [3]
- Παροχή δυνατοτήτων όπως είναι μετρικές ή έλεγχοι ποιότητας του προγράμματος (health checks) και αυτόματες παραμετροποιήσεις. [3]
- Παροχή δυνατοτήτων που αποβλέπουν στην αποφυγή παραγωγής μεγάλης ποσότητας κώδικα για λειτουργίες που μπορούν να αυτοματοποιηθούν. Για παράδειγμα με την χρήση των annotation, αποφεύγεται η δημιουργία και συντήρηση αρχείων XML όπου απαιτείται.[3]

Αρχιτεκτονική του Spring Boot:

Το Spring Boot ακολουθεί μια πολυεπίπεδη αρχιτεκτονική στην οποία το κάθε επίπεδο επικοινωνεί με το αμέσως επόμενο ή προηγούμενο, βασίζεται δηλαδή σε μια ιεραρχική δομή. [22]

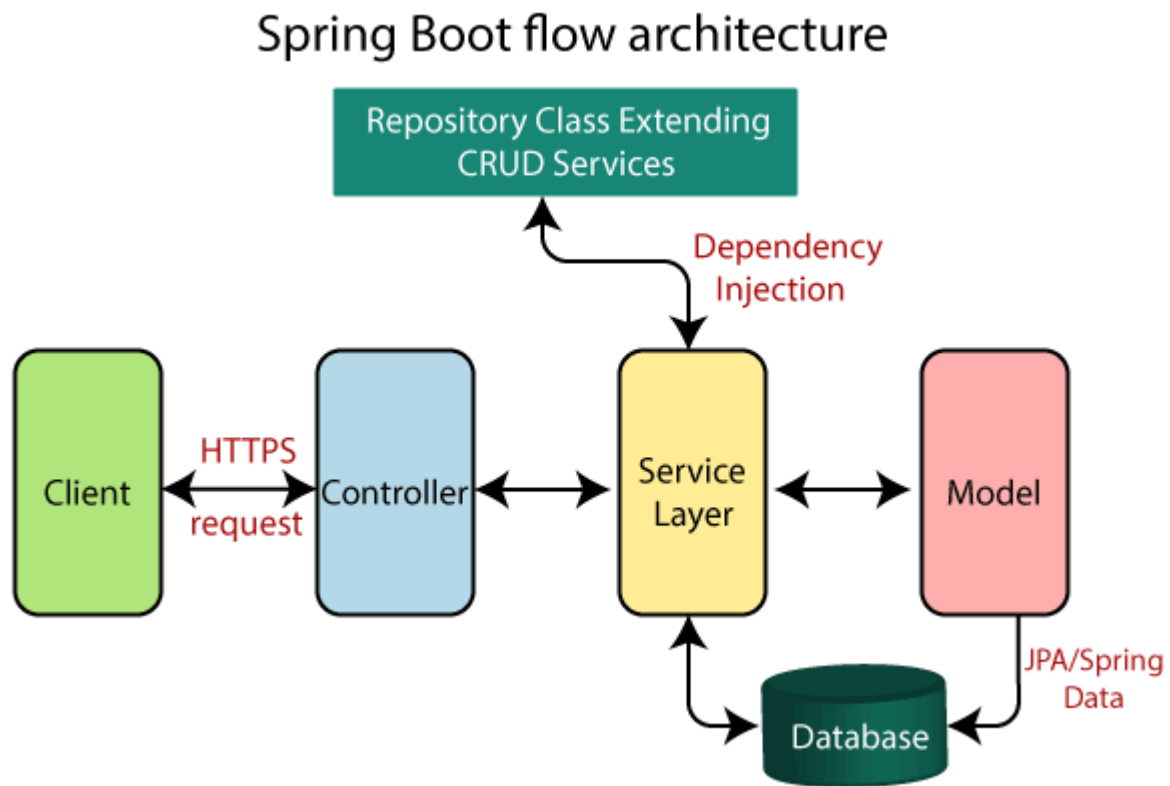
Τα βασικά αυτά επίπεδα είναι τα εξής:

- **Presentation Layer:** Το Presentation Layer διαχειρίζεται τα HTTP requests, μεταφράζει τις JSON παραμέτρους σε αντικείμενα και αυθεντικοποιεί το request και το μεταφέρει στο business layer. Εν ολίγοις αποτελείται από views (τμήμα frontend). [22]
- **Business Layer:** Το Business layer διαχειρίζεται όλο το business logic. Αποτελείται από κλάσεις τύπου service και χρησιμοποιεί αυτά τα services που του παρέχονται από τα επίπεδα που αφορούν την πρόσβαση στα δεδομένα (data access layers). Πραγματοποιεί επίσης **εξουσιοδότηση (authorization)** και **επικύρωση (validation)**. [22]
- **Persistence Layer:** Το persistence layer περιέχει όλη την λογική αποθήκευσης (storage logic) και μεταφράζει τα business objects από και προς τις εγγραφές της βάσης δεδομένων. [22]
- **Database Layer:** Στο επίπεδο της βάσης δεδομένων υλοποιούνται οι λειτουργίες CRUD (create, retrieve, update, delete). [22]



Σχ.25: Επίπεδα Αρχιτεκτονικής Spring Boot. [22]

Αρχιτεκτονική ροής Spring Boot (Spring Boot Flow Architecture):



Σχ.26: Αρχιτεκτονική Ροής Spring Boot. [22]

- Υπάρχουν κλάσεις επικύρωσης (validator classes), κλάσεις προβολής (view classes) και κλάσεις βοηθητικές για διάφορες λειτουργίες (utility classes).
- Το Spring Boot χρησιμοποιεί όλες τις ενότητες (modules) του Spring όπως για παράδειγμα το Spring MVC ή το Spring Data. Η αρχιτεκτονική του Spring Boot είναι επίσης ίδια με την αρχιτεκτονική του Spring MVC με την εξής όμως διαφορά: Δεν είναι απαραίτητες οι κλάσεις τύπου DAO (Data Access Object) και DAOImpl (υλοποίησης του Data Access Object) στο Spring Boot.
- Το Spring Boot δημιουργεί ένα data access layer και πραγματοποιεί λειτουργίες CRUD.
- Ο client πραγματοποιεί HTTP requests (PUT ή GET).
- Το request ανιχνεύεται από τον Controller και ο Controller κατευθύνει και διαχειρίζεται το request αναλόγως. Υστερα καλεί το αντίστοιχο service εάν αυτό απαιτείται.
- Στο επίπεδο του service, υλοποιούνται όλες οι λειτουργίες που αφορούν το business logic και αφορούν τα δεδομένα που αντιστοιχούν στο JPA με τις αντίστοιχες κλάσεις των μοντέλων (model classes).
- Μια σελίδα JSP επιστρέφεται στον χρήστη εάν δεν συμβεί κάποιο σφάλμα.

Στην εφαρμογή διαχείρισης παρουσιών η υλοποίηση του Spring Boot server side φτάνει μέχρι το επίπεδο των controller, καθώς το client side αναπτύσσεται με το Angular Framework το οποίο αναλύεται περαιτέρω στο αντίστοιχο κεφάλαιο.

Η έκδοση του Spring Boot που χρησιμοποιείται στην εφαρμογή είναι η 2.4.5, ενώ όσον αφορά την Java χρησιμοποιείται η Java 11.

3.1.2. Project Lombok

Το “Boilerplate” είναι ένας όρος που χρησιμοποιείται για να περιγράψει τον κώδικα που επαναλαμβάνεται σε πολλά τμήματα μιας εφαρμογής με μικρές διαφοροποιήσεις. Μία από τις πιο συχνές επικρίσεις που δέχεται η γλώσσα Java οφείλεται στον όγκο αυτού του τύπου κώδικα που συναντάται σε εφαρμογές που χρησιμοποιείται. Αυτό το πρόβλημα είναι συχνά αποτέλεσμα του τρόπου σχεδιασμού σε διάφορες βιβλιοθήκες, αλλά επιδεινώνεται από τους περιορισμούς της ίδιας της γλώσσας. [4]

Το Project Lombok στοχεύει στην μείωση αυτού του φαινομένου αντικαθιστώντας τον παρόμοιο και αναγκαστικά επαναλαμβανόμενο κώδικα με τα αντίστοιχα annotations. [4]

Αν και δεν είναι ασυνήθιστο τα annotations να χρησιμοποιούνται για να υποδείξουν την χρήση, να εφαρμόσουν δεσμεύσεις ή ακόμη να δημιουργήσουν κώδικα που χρησιμοποιείται από τα frameworks, γενικά δεν χρησιμοποιούνται για την δημιουργία κώδικα προς άμεση χρήση από την εφαρμογή. Το Project Lombok με την ενσωμάτωση στο εκάστοτε IDE παρέχει annotations που εισάγουν κώδικα ο οποίος είναι άμεσα διαθέσιμος στον προγραμματιστή και μειώνει τον συχνά επαναλαμβανόμενο κώδικα. [4]

Για παράδειγμα το **annotation @Data** όταν εφαρμοστεί σε επίπεδο κλάσης δημιουργεί αυτόματα τον κώδικα των getter και setter για κάθε χαρακτηριστικό της. Το **annotation @AllArgsConstructor** επίσης σε επίπεδο κλάσης δημιουργεί τον κώδικα για τον constructor που αφορά όλα τα χαρακτηριστικά της κλάσης ενώ το **annotation @NoArgsConstructor** δημιουργεί τον κώδικα για **κενό constructor**. [4]

Τα annotation που προαναφέρθηκαν χρησιμοποιούνται σε πολλές από τις κλάσεις που αφορούν οντότητες της εφαρμογής διαχείρισης παρουσιών και καθιστούν τον κώδικα πιο μεστό και καθαρό. *Η έκδοση του Project Lombok που χρησιμοποιείται στην εφαρμογή είναι η 1.18.20.*

3.1.3. Apache Commons Csv

Η βιβλιοθήκη Apache Commons Csv αναπτύχθηκε από το Apache Software Foundation και παρέχει μια απλή διεπαφή για το διάβασμα ή γράψιμο CSV (Comma Separated Value) αρχείων διαφόρων μορφών. [5]

Οι πιο κοινές μορφοποιήσεις (formats) αρχείων είναι προκαθορισμένες στην κλάση CSVFormat και είναι οι εξής: [6]

- Microsoft Excel
- Informix UNLOAD
- Informix UNLOAD CSV
- MySQL
- Oracle
- PostgreSQL CSV
- PostgreSQL Text
- RFC 4180
- TDF

Με την κλάση CSVFormat μπορεί να καθοριστεί η μορφοποίηση (format) ενός αρχείου CSV και να αναλυθεί η είσοδος (input) του αρχείου. [7]

Η κλάση CSVFormat είναι αμετάλλακτη (immutable) δηλαδή από την στιγμή που δημιουργηθούν αντικείμενά της αυτά δεν μπορούν να αλλάξουν στην πορεία και παρέχει διάφορες λειτουργίες όπως: [7]

- Η χρήση προκαθορισμένων μορφοποιήσεων (formats) αρχείου όπως αυτές που προαναφέρθηκαν.
- Ο καθορισμός μορφοποίησης (format) για το αρχείο, δίνοντας την δυνατότητα να κληρονομηθεί μια μορφοποίηση (format) με την κλήση κατάλληλων set μεθόδων, για παράδειγμα:

```
CSVFormat.EXCEL.withNullString("N/A").withIgnoreSurroundingSpaces(true);
```

- Ο καθορισμός των ονομάτων των στηλών που θέλει ο χρήστης να χρησιμοποιήσει για την πρόσβαση στις εγγραφές του αρχείου, για παράδειγμα:

```
CSVFormat.EXCEL.withHeader("Col1", "Col2", "Col3");
```

- Η ανάλυση και εξαγωγή στοιχείων για το διάβασμα του αρχείου χρησιμοποιώντας απευθείας την μορφοποίηση (format) του αρχείου. Για παράδειγμα:

```
Reader in = ...;
```

```
CSVFormat.EXCEL.withHeader("Col1", "Col2", "Col3").parse(in);
```

- Η ασφαλής αναφορά στις στήλες του αρχείου μέσω των ονομάτων τους. Πιο συγκεκριμένα, αν το αρχείο περιέχει επικεφαλίδα (header) για κάθε στήλη, ο κώδικας μπορεί να απλοποιηθεί και η αναφορά στις στήλες του αρχείου να είναι ασφαλής χρησιμοποιώντας την μέθοδο [CSVFormat.Builder.setHeader\(String...\)](#) χωρίς τιμές: CSVFormat.EXCEL.withHeader();

Ετσι διαβάζεται η πρώτη εγγραφή του αρχείου και οι τιμές της χρησιμοποιούνται ως ονόματα των στηλών του. Επειτα παρέχεται η δυνατότητα κλήσης της CSVRecord get μεθόδου που λαμβάνει το όνομα της εκάστοτε στήλης και δίνει το περιεχόμενο της εγγραφής. **Για παράδειγμα:**

```
String value = record.get("Col1");
```

Αυτό κάνει τον κώδικα ανεπηρέαστο από αλλαγές που μπορεί να γίνουν στην σειρά των στηλών του αρχείου CSV.

Στο σύστημα διαχείρισης παρουσιών η βιβλιοθήκη **Apache Commons Csv** χρησιμοποιείται στο **server side** της εφαρμογής για το διάβασμα και την εξαγωγή των δεδομένων των φοιτητών προς εγγραφή, αλλά και την πιστοποίηση των αριθμών μητρώων τους για την ένταξή τους σε ένα πρόγραμμα μαθήματος.

Αυτό επιτυγχάνεται με τον αρχικό έλεγχο του τύπου αρχείου που ανεβαίνει, την δημιουργία νέου **αντικειμένου CSVParser** με την ακολουθία των δεδομένων του αρχείου και μέσω αυτού αποκτάται πρόσβαση στα δεδομένα προκειμένου να γίνουν οι απαραίτητες διαδικασίες.

Η έκδοση της βιβλιοθήκης Apache Commons Csv που χρησιμοποιείται στην εφαρμογή είναι η 1.8.

3.1.4. Apache POI-XSSF/SXSSF

Το Apache POI-XSSF/SXSSF αναπτύχθηκε από το Apache Software Foundation. Το POI-XSSF/SXSSF είναι ένα API δηλαδή μια διεπαφή προγραμματισμού εφαρμογών που χρησιμοποιείται για την πρόσβαση σε αρχεία της μορφοποίησης (format) Microsoft Excel. Ειδικότερα το XSSF είναι η καθαρή υλοποίηση του POI Project σε γλώσσα Java για την μορφοποίηση (format) αρχείου Excel 2007 OOXML (.xlsx). [8]

Το XSSF παρέχει την δυνατότητα δημιουργίας, επεξεργασίας, διαβάσματος και γραψίματος αρχείων τύπου XLS. Πιο συγκεκριμένα προσφέρει:

- Χαμηλού επιπέδου (low - level) δομές για πληθώρα συγκεκριμένων αναγκών. [8]
- Ένα eventmodel api για άνετη πρόσβαση μόνο για διάβασμα του αρχείου. [8]
- Ένα ολοκληρωμένο **usermodel api** για την δημιουργία, το διάβασμα και την επεξεργασία αρχείων XLS. [8]

Κάποιες από τις διευκολύνσεις που παρέχει το υψηλού επιπέδου usermodel API είναι οι εξής:

- **Δυνατότητα δημιουργίας (workbook) για τα φύλλα excel** που πρόκειται να δημιουργηθούν ή να εισαχθούν σε αυτό ως ακολουθία εγγραφών μέσω της κλάσης `org.apache.poi.xssf.usermodel.XSSFWorkbook`. [9]
- **Δυνατότητα δημιουργίας φύλλου excel** με την μέθοδο `createSheet()` για ένα υπάρχον βιβλίο εργασίας (workbook). [9]
- **Δυνατότητα ανάκτησης φύλλου excel** από ένα συγκεκριμένο σημείο του μέσω της μεθόδου `getSheetAt(int index)`. [9]

- Επειδή τα προς δημιουργία φύλλα excel δεν έχουν εξ' αρχής όνομα, το api παρέχει την μέθοδο `Workbook.setSheetName(sheetindex,"SheetName",encoding)`. Στην περίπτωση του XSSF η κωδικοποίηση είναι αυτομάτως unicode. [9]
- **Δυνατότητα δημιουργίας rows** (γραμμών για τις εγγραφές του αρχείου) μέσω της μεθόδου `createRow(rowNumber)` για ένα υπάρχον φύλλο excel. Για παράδειγμα αν ένα αρχείο excel έχει ανέβει στο σύστημα, με την χρήση της δομής επανάληψης και της παραπάνω μεθόδου μπορούν να ανακτηθούν οι εγγραφές για κάθε γραμμή του. **Είναι σημαντικό να τονιστεί ότι μόνο οι γραμμές (rows) των οποίων τα κελιά (cells) έχουν τιμές, μπορούν να προστεθούν στο φύλλο excel.** [9]
- **Δυνατότητα δημιουργίας cells** (κελιών) για κάθε γραμμή εγγραφής με την μέθοδο `createCell(column, type)`. Μόνο τα κελιά (cells) που έχουν τιμές μπορούν να προστεθούν στην γραμμή (row). Τα κελιά cell, μπορούν να έχουν τον κατάλληλο τύπο, όπως για παράδειγμα `CELL_TYPE_NUMERIC` ή `CELL_TYPE_STRING`, ανάλογα με το είδος της τιμής που πρόκειται να περιέχουν. [9]
- **Δυνατότητα καταχώρησης τιμής για κάθε κελί (cell)** με την μέθοδο `setCellValue`. [9]
- **Δυνατότητα καθορισμού των διαστάσεων των γραμμών (rows) και των κελιών (cells)** με τις μεθόδους `setRowHeight(height)` και `setColumnWidth(colindex, width)` καθώς και καθορισμού του στυλ των κελιών με διάφορες άλλες μεθόδους. [9]

Στο σύστημα διαχείρισης παρουσιών το **POI-XSSF/SXSSF API** χρησιμοποιείται στο server side της εφαρμογής για το διάβασμα, την εξαγωγή και τον έλεγχο των δεδομένων των φοιτητών προς εγγραφή αλλά και την πιστοποίηση των αριθμών μητρώων τους για την ένταξή τους σε ένα πρόγραμμα μαθήματος.

Αυτό επιτυγχάνεται με τον αρχικό έλεγχο του τύπου αρχείου που ανεβαίνει, την δημιουργία νέου βιβλίου εργασίας `Workbook` με την ακολουθία των δεδομένων του αρχείου η οποία γίνεται προσβάσιμη μέσω της δημιουργίας ενός sheet (φύλλου για το excel) που αφορά το workbook και μέσω αυτού αποκτάται πρόσβαση στα δεδομένα προκειμένου να γίνουν οι απαραίτητες διαδικασίες.

Η έκδοση του Apache POI-XSSF/SXSSF για αρχεία του τύπου ooxml που χρησιμοποιείται στην εφαρμογή είναι η 4.1.2.

3.1.5. Java Bean Validation Framework

Η επικύρωση (validation) των δεδομένων που εισάγονται από τον χρήστη είναι κοινή απαίτηση στις περισσότερες εφαρμογές. Το Java Bean Validation framework είναι πλέον εκ των πραγμάτων ένα πρότυπο διαχείρισης αυτής της αναγκαιότητας. [10]

Η πρότυπη έκδοση του framework είναι γνωστή ως **JSR 380** ή **Bean Validation 2.0**. Αναλυτικότερα το JSR 380 είναι ένας προσδιορισμός του **Java API** για **bean validation** και είναι τμήμα των **Jakarta EE** και **JavaSE**. [10]

Σκοπός του είναι να επιβεβαιώσει ότι τα χαρακτηριστικά των οντοτήτων, τηρούν ορισμένα κριτήρια, μέσω της χρήσης **annotations** όπως είναι τα **@NotNull**, **@Min** και **@Max**. Για την χρησιμοποίησή του απαιτείται έκδοση μεγαλύτερη ή ίση με την **Java 8** ώστε να είναι προσβάσιμες διάφορες νέες λειτουργικότητές του. [10]

Στο server side του συστήματος διαχείρισης παρουσιών συμπεριλήφθηκαν τα ακόλουθα εξαρτήματα (dependencies) του Java Bean Validation Framework:

- **Validation API, έκδοση 2.0.1.Final:** Σύμφωνα με τις προδιαγραφές του JSR 380 το validation-api dependency περιέχει τα πρότυπα APIs επικύρωσης (validation). [10]
- **Hibernate Validator έκδοση 6.1.7.Final:** Ο **Hibernate Validator** αποτελεί μια υλοποίηση του **Validation Api** (Validation API Reference Implementation). **Πρέπει να τονιστεί ότι δεν έχει καμία σχέση με το κομμάτι του Hibernate που αφορά το persistence** (δηλαδή τις λειτουργικότητες που έχουν σχέση με την επικοινωνία με την βάση δεδομένων). [10] Ο **Hibernate Validator** παρέχει την δυνατότητα έκφρασης περιορισμών για την διαμόρφωση της επικύρωσης (validation), μέσω ειδικών annotations. [27]

Κάποια από τα annotations που παρέχει είναι:

@NotNull, @Size(min = integer, max = integer), @Min(integer) και αρκετά ακόμη.

Ο **Hibernate Validator** είναι επεκτάσιμος καθώς παρέχει την δυνατότητα δημιουργίας περιορισμών προσαρμοσμένων στις ανάγκες που προκύπτουν απο τις εκάστοτε απαιτήσεις (custom constraints). [27]

- **Expression Language Dependency και συγκεκριμένα το javax.el (έκδοση 3.0.0) dependency του GlassFish:** Το JSR 380 υποστηρίζει την προσθήκη μηνυμάτων

εσωτερικά στα annotation επικύρωσης που μπορούν να αξιοποιηθούν στην εφαρμογή σε περίπτωση που οι περιορισμοί της επικύρωσης παραβιαστούν. [10]

Λίγα λόγια για την προγραμματιστική επικύρωση (validation):

Πολλά **frameworks** όπως το **Spring** διαθέτουν απλούς τρόπους για την ενεργοποίηση της διαδικασίας επικύρωσης απλά με την χρήση των **annotation**. Αυτό συμβαίνει για να μην χρειαστεί η αλληλεπίδραση με το **Validation Api** να γίνει προγραμματιστικά. Ωστόσο σε κάποιες περιπτώσεις μπορεί να χρειαστεί η προγραμματιστική αλληλεπίδραση με το Validation Api με την βοήθεια ειδικών κλάσεων. [10]

Πως χρησιμοποιήθηκαν οι δυνατότητες του Java Bean Validation Framework στο σύστημα διαχείρισης παρουσιών:

Στο server side του συστήματος διαχείρισης παρουσιών οι προαναφερθείσες δυνατότητες του Java Bean Validation Framework χρησιμοποιήθηκαν όπου και όπως ήταν αναγκαίο.

Για παράδειγμα σε επίπεδο κλάσης αξιοποιήθηκαν αρκετά οι δυνατότητες των annotation, ενώ σε κάποιες περιπτώσεις όπως όταν χρειάστηκε επικύρωση (validation) για τα δεδομένα των csv ή excel αρχείων αυτό έγινε προγραμματιστικά χρησιμοποιώντας τα κατάλληλα εργαλεία καθώς ήταν αδύνατη η αλληλεπίδραση με το Validation Api μέσω των annotation.

3.1.6. JPA - Spring Data JPA

JPA (Java Persistence Api)

Το Java Persistence Api σχετίζεται με το επίπεδο επικοινωνίας με την βάση δεδομένων (persistence layer) κάθε Java εφαρμογής και κατ' επέκταση με κάθε μηχανισμό στον οποίο τα αντικείμενα Java θα πρέπει να αποθηκευτούν σε μία βάση δεδομένων. [11]

Επειδή ακριβώς δεν είναι απαραίτητο όλα τα αντικείμενα που προκύπτουν να αποθηκεύονται στην βάση δεδομένων, για αυτό σε πολλές εφαρμογές είναι απαραίτητο να διατηρηθούν τα

κυριότερα από αυτά (δηλαδή τα business objects). Με βάση την αναγκαιότητα αυτή, το JPA προσφέρει στον προγραμματιστή την δυνατότητα να αποφασίζει ποια αντικείμενα Java θα κρατηθούν στην βάση δεδομένων καθώς και με ποιον τρόπο. [11]

Πιο συγκεκριμένα, το Java Persistence API (JPA) είναι ένα πρότυπο της γλώσσας Java (Java standard) για την αντιστοίχιση αντικειμένων Java σε μια σχεσιακή βάση δεδομένων. Η αντιστοίχιση αντικειμένων Java σε πίνακες μιας βάσης δεδομένων και το αντίστροφο, καλείται ORM (Object Relational Mapping). [23]

Το JPA είναι μια προσέγγιση του ORM, το οποίο αναλύεται σε σχετικό κεφάλαιο παρακάτω.

Ετσι μέσω του JPA ο προγραμματιστής μπορεί να αντιστοιχίζει, να αποθηκεύει, να ενημερώνει και να λαμβάνει δεδομένα από σχεσιακές βάσεις δεδομένων με την μορφή Java αντικειμένων και το αντίστροφο. [23]

Είναι σημαντικό να τονιστεί ότι το JPA από μόνο του δεν αποτελεί ένα εργαλείο ή framework, αλλά διασαφηνίζει ένα σύνολο από στρατηγικές που μπορούν να υλοποιηθούν από οποιοδήποτε εργαλείο ή framework, [11] ενώ μπορεί να χρησιμοποιηθεί τόσο σε Java-EE όσο και σε Java-SE εφαρμογές. [23]

JPA Eager και Lazy Loading:

Η βασική ιδέα γύρω από το JPA είναι η δημιουργία ενός αντιγράφου της βάσης δεδομένων στην κρυφή μνήμη (cache memory). Κατά την συνδιαλλαγή με την βάση δεδομένων, πρώτα θα κρατήσει αντίγραφα των δεδομένων και μόνο όταν διεκπεραιωθεί επιτυχώς η διαδικασία τότε γίνονται και οι μόνιμες αλλαγές στην βάση. [17]

Υπάρχουν δύο τρόποι πρόσβασης στα δεδομένα: [17]

Eager fetch (άμεση πρόσβαση):

- Ανάκτηση ολόκληρης της εγγραφής μέσω του πρωτεύοντος κλειδιού (primary key).

Lazy fetch (Πρόσβαση όποτε χρειαστεί):

- Έλεγχος για το αν η εγγραφή υπάρχει μέσω του πρωτεύοντος κλειδιού. Επειτα αν χρειαστεί η ανάκτηση χαρακτηριστικών της οντότητας τότε ανακτώνται οι υπόλοιπες πληροφορίες. Αυτό συμβαίνει όταν η εγγραφή ανακτάται για πρώτη φορά κατα την οποία κρατιέται αντίγραφο των πληροφοριών της οντότητας στην κρυφή μνήμη (cache memory). Έτσι σε περίπτωση που ξαναχρειαστούν οι ίδιες πληροφορίες αντλούνται αμέσως από το αντίγραφο.

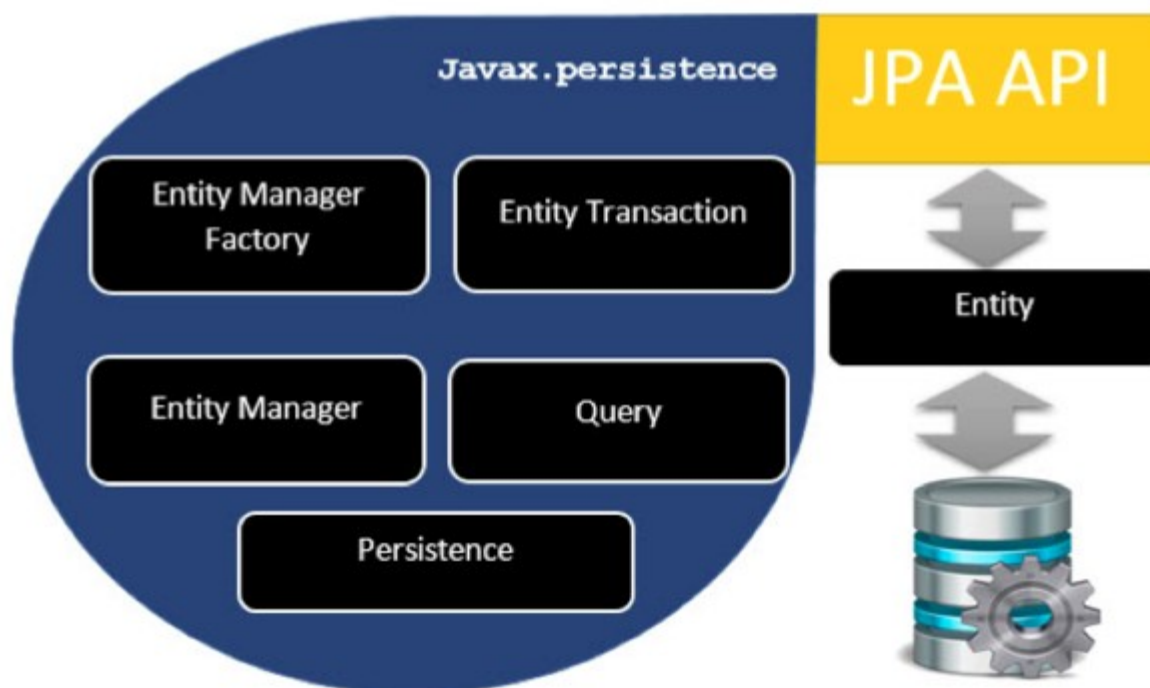
Το eager και lazy fetch του JPA, προσδίδουν ταχύτητα και αποδοτικότητα στην εφαρμογή καθώς η άντληση των δεδομένων από την βάση πραγματοποιείται μόνο όποτε και όπου είναι απαραίτητο. [17]

Αρχιτεκτονική του JPA:

Το Java Persistence API συμβάλλει στην αποθήκευση των κεντρικών οντοτήτων (business entities) ως σχεσιακές οντότητες (relational entities). Δείχνει πως ορίζεται ένα παλαιό αντικείμενο Java (plain old java object) ως μια οντότητα και πως γίνεται η διαχείριση των οντοτήτων μέσω συσχετίσεων. [23]

Αρχιτεκτονική σε επίπεδο κλάσης (Class Level Architecture): [23]

Η παρακάτω εικόνα δείχνει την αρχιτεκτονική του JPA σε επίπεδο κλάσης. Παρουσιάζει επίσης της κύριες κλάσεις (core classes) και διεπαφές (interfaces) του JPA.



Σχ.27: Αρχιτεκτονική JPA σε επίπεδο κλάσης [23]

Παρακάτω περιγράφονται οι μονάδες που απαρτίζουν την παραπάνω αρχιτεκτονική:

- **EntityManagerFactory:** Είναι η κλάση παραγωγός (factory class) του EntityManager. Επίσης δημιουργεί και διαχειρίζεται πολλαπλές διεπαφές (interfaces) του EntityManager. [23]
- **EntityManager:** Είναι μια διεπαφή (interface), για την διαχείριση των λειτουργιών αλληλεπίδρασης με το επίπεδο της βάσης δεδομένων (persistence operations) που αφορούν τα αντικείμενα. Μέσω του EntityManager μπορούν να παραχθούν ερωτήματα (queries) τύπου Query. [23]
- **Entity:** Οι οντότητες είναι τα αντικείμενα που αντιστοιχούν στην βάση δεδομένων και αποθηκεύονται ως εγγραφές της βάσης δεδομένων. [23]
- **EntityTransaction:** Έχει ένα προς ένα συσχέτιση (one-to-one relationship) με τον EntityManager. Για κάθε EntityManager, οι λειτουργίες του συντηρούνται από την κλάση EntityTransaction. [23]
- **Persistence:** Η κλάση αυτή περιέχει στατικές μεθόδους (static methods) για την απόκτηση ενός πανομοιότυπου (instance) EntityManagerFactory. [23]
- **Query:** Αυτή η διεπαφή (interface), υλοποιείται από κάθε παροχή του JPA ώστε να λαμβάνει τα σχεσιακά αντικείμενα (relational objects), που πληρούν τα ανάλογα κριτήρια. [23]

Οι παραπάνω κλάσεις και διεπαφές (interfaces), χρησιμοποιούνται για την αποθήκευση των οντοτήτων ως εγγραφές σε μια βάση δεδομένων. Επίσης απλοποιούν την δουλειά του προγραμματιστή καθώς ελατώνουν την προσπάθεια που απαιτείται για αυτή την διαδικασία επιτρέποντάς του να εστιάσει σε σημαντικότερες ενέργειες όπως την δημιουργία κώδικα για την αντιστοίχιση των κλάσεων με τους πίνακες της βάσης δεδομένων. [23]

Spring Data JPA

Το Spring Data JPA είναι τμήμα της ευρύτερης οικογένειας του Spring Data και διευκολύνει την υλοποίηση αποθετηρίων (repositories) βασισμένων στο JPA (Java Persistence Api). [12]

Αυτή η ενότητα του Spring Data, αφορά την παροχή ενισχυμένης υποστήριξης στο επίπεδο που αφορά την πρόσβαση στα δεδομένα (data access layer) η οποία βασίζεται στο JPA. Το Spring Data JPA διευκολύνει την δημιουργία των βασισμένων σε Spring εφαρμογών που χρησιμοποιούν τεχνολογίες που αφορούν την πρόσβαση σε δεδομένα. Για αρκετό καιρό, η υλοποίηση της εφαρμογής στο επίπεδο που αφορούσε την πρόσβαση στα δεδομένα (data access layer), ήταν κατα κάποιο τρόπο περίπλοκη. Υπερβολικά πολύς περιττός κώδικας έπρεπε να γραφτεί για την εκτέλεση απλών ερωτημάτων (queries) όπως επίσης για την υλοποίηση σελιδοποίησης (pagination) και ελέγχων. [12]

Το Spring Data JPA στοχεύει να βελτιώσει σημαντικά την υλοποίηση της εφαρμογής στο επίπεδο που αφορά την πρόσβαση στα δεδομένα (data access layer), μειώνοντας την προσπάθεια στο επίπεδο που πραγματικά είναι χρήσιμη. [12]

Ετσι, ο προγραμματιστής δημιουργεί τις διεπαφές των αποθετηρίων (repository interfaces), συμπεριλαμβάνοντας δικές του μεθόδους αναζήτησης (custom finder methods) και το Spring παρέχει την υλοποίηση αυτόματα. [12]

Πλεονεκτήματα του Spring Data Jpa:

- **Εξελιγμένη και ποιοτική υποστήριξη** για τη δημιουργία αποθετηρίων με βάση το Spring και το JPA. [12]
- **Υποστήριξη για Querydsl δηλαδή για ενοποιημένα ερωτήματα** για την γλώσσα Java (unified queries for Java) και κατ' επέκταση JPA ερωτήματα ασφαλούς τύπου (type-safe JPA queries). [12]
- **Δυνατότητα διαφανών ελέγχων** που αφορούν τις κλάσεις που αφορούν την ραχοκοκαλιά της εφαρμογής (domain classes). [12]
- **Υποστήριξη για σελιδοποίηση (pagination)**, εκτέλεση δυναμικών ερωτημάτων, δυνατότητα ενσωμάτωσης ιδιόχειρα φτιαγμένου κώδικα που αφορά την πρόσβαση σε δεδομένα. [12]
- **Επικύρωση (validation) των ερωτημάτων (queries)** που φέρουν το **annotation @Query** στον χρόνο εκκίνησης (bootstrap time). [12]
- Υποστήριξη για αντιστοίχιση οντοτήτων (**entity mapping**) βασισμένη σε XML. [12]
- **Διαμόρφωση αποθετηρίου** βασισμένου σε **JavaConfig** εισάγοντας το **annotation @EnableJpaRepositories**. [12]

Στο server side του συστήματος διαχείρισης παρουσιών, η τεχνολογία του Spring Data Jpa χρησιμοποιείται για την υλοποίηση της εφαρμογής στο επίπεδο που αφορά την πρόσβαση στα δεδομένα (data access layer).

Με το Spring Data JPA είναι δημιουργημένες οι διεπαφές των αποθετηρίων (δηλαδή τα repository interfaces). Επίσης έχουν χρησιμοποιηθεί πολλά από τα έτοιμα ερωτήματα (queries) που παρέχει το Spring Data JPA μεταξύ αυτών και ερωτήματα που εξυπηρετούν ταξινόμηση σε δεδομένα (sorting) και άλλα ακόμη που κατασκευάστηκαν στα πλαίσια των δυνατοτήτων του Spring Data JPA για πιο εξειδικευμένες λειτουργίες, όπως πιο πολύπλοκες αναζητήσεις ή ελέγχους. *Η έκδοση του Spring Data JPA που χρησιμοποιείται στην εφαρμογή είναι η 2.4.5.*

3.1.7. Java Persistence Query Language

Η Java Persistence Query Language (JPQL) είναι γλώσσα ερωτημάτων που ορίζεται από το JPA.

Η JPQL έχει παρόμοια δομή και συντακτικό με την SQL με μια σημαντική όμως διαφορά:

Η JPQL βασίζεται στα αντικείμενα των οντοτήτων (entity objects) για να ορίσει ένα ερώτημα (query) και όχι στους πίνακες της βάσης δεδομένων. Αυτή η ιδιαιτερότητά του κάνει εύκολη την διαδικασία δημιουργίας και εκτέλεσης ερωτημάτων αλλά πάντοτε σε τελικό στάδιο καταλήγουν σε SQL στην βάση. Οποιαδήποτε κι αν είναι η υλοποίηση του JPA θα πρέπει να μετατρέπει το JPQL ερώτημα σε SQL. [13]

Η JPQL μπορεί να χρησιμοποιηθεί για διάβασμα (SELECT), όπως επίσης και για ογκώδεις ενημερώσεις (UPDATE) και διαγραφές (DELETE).

Μπορεί επίσης να χρησιμοποιηθεί σε ένα **NamedQuery** (δηλαδή κλασικό ερώτημα) (με την βοήθεια annotations ή xml) ή σε δυναμικά ερωτήματα (dynamic queries) χρησιμοποιώντας την μέθοδο `createQuery()` του `EntityManager` API. [13] Το `EntityManager` API παρέχει διάφορες δυνατότητες σχετικές με την δημιουργία ερωτημάτων και την διαχείριση λειτουργιών σχετικές με το επίπεδο της εφαρμογής που αφορά την επικοινωνία με την βάση δεδομένων.

Πλεονεκτήματα της JPQL:

- **Δυνατότητα δημιουργίας δυναμικών ερωτημάτων:** Όπως προαναφέρθηκε τα δυναμικά ερωτήματα υλοποιούνται με την βοήθεια του `EntityManager` API. Το βασικό πλεονέκτημα της δυνατότητας δημιουργίας δυναμικών ερωτημάτων είναι ότι προσφέρουν την δυνατότητα πρόσβασης σε δεδομένα που ίσως είναι ανέφικτη ή λιγότερο αποδοτική σε κάποιες περιπτώσεις. [14]
- **Δυνατότητα δημιουργίας `namedQueries` με το `@NamedQuery` annotation.** Ένα `named query` εφαρμόζεται σε επίπεδο κλάσης και συμβάλλει στην καλύτερη οργάνωση του κώδικα και είναι αποδοτικότερο από ένα αντίστοιχο δυναμικό ερώτημα. Αυτό συμβαίνει κυρίως γιατί δεν ενσωματώνει τις παραμέτρους που εισάγονται σε αυτό με

την συμβολοσειρά του ερωτήματος, αλλά περνάει της παραμέτρους στο ερώτημα, όποτε αυτό χρειαστεί. [15]

- **Μπορεί να συνδυαστεί πολύ αποδοτικά με το @Query annotation του Spring Data JPA και να δημιουργήσει ισχυρά JPQL ερωτήματα (queries).** [16]

Πιο συγκεκριμένα, το @Query annotation έχει μεγαλύτερη προτεραιότητα έναντι των namedQueries με @NamedQuery annotation ή των namedQueries που ορίζονται σε ένα αρχείο orm.xml. [16]

Γενικά, είναι καλύτερη πρακτική η υλοποίηση ενός ερωτήματος (query) σε επίπεδο μεθόδου του αποθετηρίου (repository), παρά σε επίπεδο κλάσης ως namedQuery.

Το αποθετήριο είναι υπεύθυνο για το επίπεδο της εφαρμογής που αφορά την πρόσβαση στα δεδομένα (data access layer), άρα και καλύτερο μέρος για την τοποθέτηση των αντίστοιχων ερωτημάτων. [16]

Ετσι προκειμένου να οριστεί η εντολή SQL που θα εκτελεστεί για μία μέθοδο του αποθετηρίου (repository), αρκεί να τοποθετηθεί σε αυτήν ένα @Query annotation που θα περιέχει το κατάλληλο JPQL ερώτημα (query). [16]

Στο server side του συστήματος διαχείρισης παρουσιών, χρησιμοποιήθηκε κατα κύριο λόγο ο συνδυασμός του @Query annotation του Spring Data JPA με την γλώσσα JPQL σε επίπεδο μεθόδου του αποθετηρίου συμβάλλοντας στην δημιουργία σύνθετων ερωτημάτων (queries) αλλά με τρόπο μεστό και ξεκάθαρο, προσφέροντας την πρόσβαση που απαιτείται σε κάθε αντικείμενο ή χαρακτηριστικό αντικείμενου, μέσω των κατάλληλων συσχετίσεων ή συνθηκών που περιγράφονται σε αυτό.

Με την βοήθεια του @Query annotation και της JPQL δημιουργήθηκαν ερωτήματα που εξυπηρετούν λειτουργίες όπως:

- **Σελιδοποίηση (pagination):** Κατα την σελιδοποίηση (pagination), το server side σύστημα επιστρέφει ένα υποσύνολο εκ του συνολικού αποτελέσματος σε μια σελίδα. Αυτή η λειτουργία είναι εξαιρετικά χρήσιμη ιδιαίτερα κατα την πλοήγηση μεταξύ πολλών σελίδων με δεδομένα σε μία web εφαρμογή.

Το μεγαλύτερο πλεονέκτημα όμως είναι ότι ο όγκος των δεδομένων που στέλνονται από τον server στον client ελαχιστοποιείται. Αυτό συμβαίνει διότι κατα την σελιδοποίηση (pagination) τα δεδομένα στέλνονται από τον server τμηματικά. Ετσι βελτιώνεται η συνολική απόδοση της εφαρμογής καθώς το pagination στον server είναι ταχύτερο, δεν επιβαρύνεται το δίκτυο εφόσον “φορτώνει” μικρότερο όγκο δεδομένων και παράλληλα το client side είναι ελαφρύτερο. [16]

- **Φιλτράρισμα (filtering):** Δηλαδή κατά το φιλτράρισμα υλοποιείται αναζήτηση με βάση συγκεκριμένη παράμετρο μέσω ισχυρών JPQL ερωτημάτων. Έτσι επιτυγχάνεται μια γρήγορη και αποδοτική αναζήτηση στο server side της εφαρμογής μετατοπίζοντας το κέντρο βάρους στον server και όχι στον εκάστοτε client κάνοντας την εφαρμογή αποδοτικότερη στο σύνολό της.
- Σύνθετες αναζητήσεις.
- Αναζητήσεις που εξυπηρετούν ελέγχους.

Τέλος, πρέπει να τονιστεί ότι σε ορισμένες περιπτώσεις χρησιμοποιήθηκαν JPQL ερωτήματα μέσω της μεθόδου `createQuery` που προσφέρει ο `EntityManager` (διεπαφή που επιτρέπει την αλληλεπίδραση με το επίπεδο πρόσβασης στα δεδομένα) του JPA και αφορούν την ενημέρωση της κατάστασης (status) των προγραμμάτων μαθημάτων, ανάλογα με το τρέχον ακαδημαϊκό έτος.

Τα ερωτήματα αυτά υλοποιήθηκαν σε επίπεδο προσαρμοσμένου αποθετηρίου (custom repository), το οποίο κληρονομείται από το αντίστοιχο βασισμένο στο JPA αποθετήριο (JPA repository) και καλούνται από ειδικό νήμα (thread) όταν αλλάζει το ακαδημαϊκό εξάμηνο. Έτσι γίνονται οι κατάλληλες ενημερώσεις.

3.1.8. Hibernate - ORM

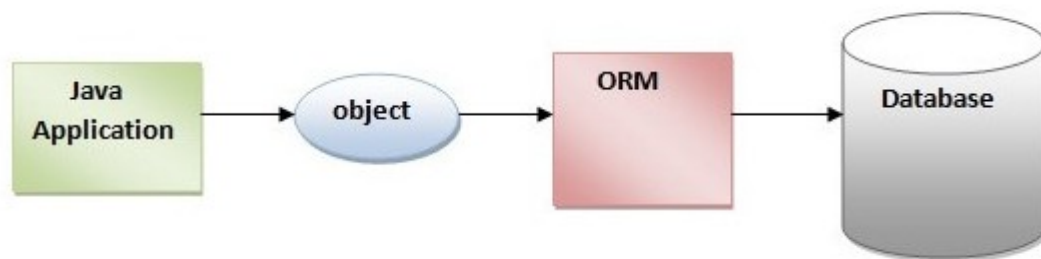
ORM

Το **ORM (Object Relational Mapping)**, είναι μια τεχνική προγραμματισμού στην οποία χρησιμοποιείται ένας περιγραφέας μεταδεδομένων για την σύνδεση του κώδικα αντικειμένου με μια σχεσιακή βάση δεδομένων. Ο κώδικας του αντικειμένου είναι γραμμένος σε μια αντικειμενοστραφής γλώσσα προγραμματισμού (OOP) όπως είναι για παράδειγμα η Java. [19]

Το ORM αντιστοιχίζει τα δεδομένα των αντικειμένων μιας αντικειμενοστραφούς γλώσσας προγραμματισμού σε δεδομένα που αναπαρίστανται με την μορφή πινάκων σε σχεσιακές βάσεις δεδομένων. [19]

ORM Εργαλείο (ORM tool)

Ενα εργαλείο ORM απλοποιεί την διαδικασία της δημιουργίας των δεδομένων αλλά και του χειρισμού και πρόσβασης σε αυτά. Η κύρια λειτουργία ενός εργαλείου ORM είναι η αναπαράσταση των δεδομένων που αποθηκεύονται στην βάση με αντικείμενα. Το εργαλείο ORM (ORM tool), χρησιμοποιεί εσωτερικά το **JDBC API** για την αλληλεπίδραση με την βάση δεδομένων. [19]



Σχ.28: ORM Tool [19]

Hibernate

Το **Hibernate** είναι ένα **Java Framework** το οποίο απλοποιεί την ανάπτυξη εφαρμογών σε Java στο επίπεδο που αφορά την αλληλεπίδραση με την βάση δεδομένων. Είναι ένα ανοικτού κώδικα **ORM (Object Relational Mapping)** εργαλείο. Το Hibernate υλοποιεί τις προδιαγραφές του **JPA (Java Persistence API)** στο επίπεδο που αφορά την πρόσβαση στα δεδομένα (data access layer). [18].

Βασικά γνωρίσματα και πλεονεκτήματα του Hibernate:

- **ORM (Object Relational Mapping):** Το Hibernate επιτρέπει στους προγραμματιστές να χειρίζονται εύκολα τις διαδικασίες που αφορούν το επίπεδο πρόσβασης και διαχείρισης των δεδομένων στα πλαίσια δημιουργίας μιας εφαρμογής. Ως ένα ORM Framework, το Hibernate αφορά την διατήρηση των δεδομένων (data persistence) καθώς εφαρμόζεται σε σχεσιακές βάσεις δεδομένων. [20]

- **JPA Provider:** Εκτός από το δικό του API, το Hibernate αποτελεί επίσης υλοποίηση των προδιαγραφών του JPA (Java Persistence API). Ως εκ τούτου μπορεί να χρησιμοποιηθεί εύκολα σε οποιοδήποτε περιβάλλον που υποστηρίζει το JPA όπως σε Java SE εφαρμογές, Java EE application servers ή Enterprise OSGi containers. [20]
- **Παροχή persistence εργαλείων προσαρμοσμένων στις αρχές της αντικειμενοστρέφειας (Idiomatic Persistence):** Το Hibernate δίνει την δυνατότητα δημιουργίας κλάσεων οντοτήτων άμεσα συνδεδεμένων με το persistence και συνυφασμένων με τα χαρακτηριστικά που διέπουν την αντικειμενοστρέφεια όπως είναι η κληρονομικότητα, ο πολυμορφισμός, η συσχέτιση, η σύνθεση και γενικά με όλα τα εργαλεία της γλώσσας Java. Το Hibernate δεν απαιτεί διεπαφές (interfaces) ή βασικές κλάσεις (base classes) για τις persistent κλάσεις και επιτρέπει το persistence για οποιαδήποτε κλάση ή δομή δεδομένων. [20]
- **Υψηλή απόδοση (High Performance):** Το Hibernate υποστηρίζει lazy initialization, και πολυάριθμες στρατηγικές ανάκτησης δεδομένων και άλλων λειτουργιών που συμβάλλουν στην μεγιστοποίηση της απόδοσης των διαδικασιών που αφορούν το persistence της εφαρμογής στην οποία εφαρμόζεται. [20]
- **Επεκτασιμότητα σε επίπεδο κλίμακας (Scalability):** Το Hibernate σχεδιάστηκε για να μπορεί να λειτουργήσει σε ένα σύμπλεγμα application server (application server cluster) και να παρέχει αρχιτεκτονική υψηλής επεκτασιμότητας σε επίπεδο κλίμακας (scalable architecture). Το Hibernate προσαρμόζεται σε οποιοδήποτε περιβάλλον: Από το οικιακό δίκτυο που εξυπηρετεί μερικούς εκατοντάδες χρήστες έως και μεγάλες εφαρμογές που αφορούν εκατοντάδες χιλιάδες. [20]
- **Αξιοπιστία (Reliability):** Το Hibernate είναι γνωστό για την εξαιρετική σταθερότητα και ποιότητά του που αποδεικνύεται από την αποδοχή του και την χρήση του από δεκάδες χιλιάδες προγραμματιστές. [20]
- **Εύκολη Παραμετροποίηση:** Το Hibernate είναι εξαιρετικά παραμετροποιήσιμο και επεκτάσιμο. [20]

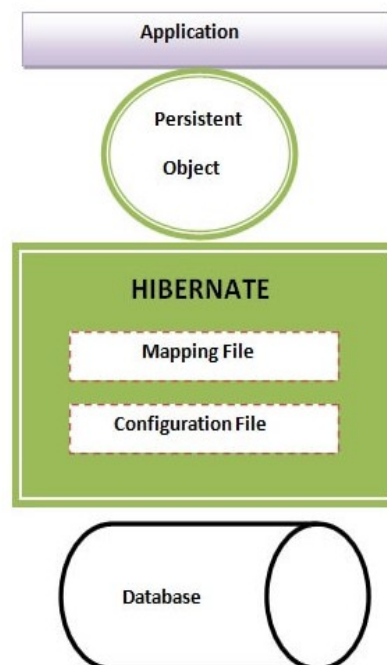
- **Παροχή ερωτημάτων ανεξάρτητων από την βάση δεδομένων (Database Independent Query):** HQL (Hibernate Query Language) είναι μια αντικειμενοστραφής έκδοση της SQL που δημιουργεί ανεξάρτητα από την βάση δεδομένων ερωτήματα (queries). Οπότε δεν χρειάζεται να γράφονται συγκεκριμένα ερωτήματα (queries) προς την βάση δεδομένων. Έτσι αν η βάση δεδομένων άλλαζε ή υπόκειντο αλλαγές για μια συγκεκριμένη εφαρμογή, χωρίς το Hibernate θα έπρεπε να αλλάξουν και όλα τα αντίστοιχα ερωτήματα, πράγμα που θα έκανε την συντήρησή της δυσκολότερη. [18]
- **Αυτόματη Δημιουργία Πινάκων (Automatic Table Creation):** Το Hibernate Framework παρέχει την δυνατότητα αυτόματης δημιουργίας πινάκων στην βάση δεδομένων, επομένως η όλη διαδικασία δεν χρειάζεται να γίνει χειροκίνητα. [18]
- **Απλοποίηση των σύνθετων συνδέσεων (Simplifies Complex Join):** Η ανάκτηση δεδομένων από πολλούς πίνακες γίνεται με ευκολία. [18]

Αρχιτεκτονική του Hibernate:

Η αρχιτεκτονική του Hibernate κατηγοριοποιείται σε 4 επίπεδα:

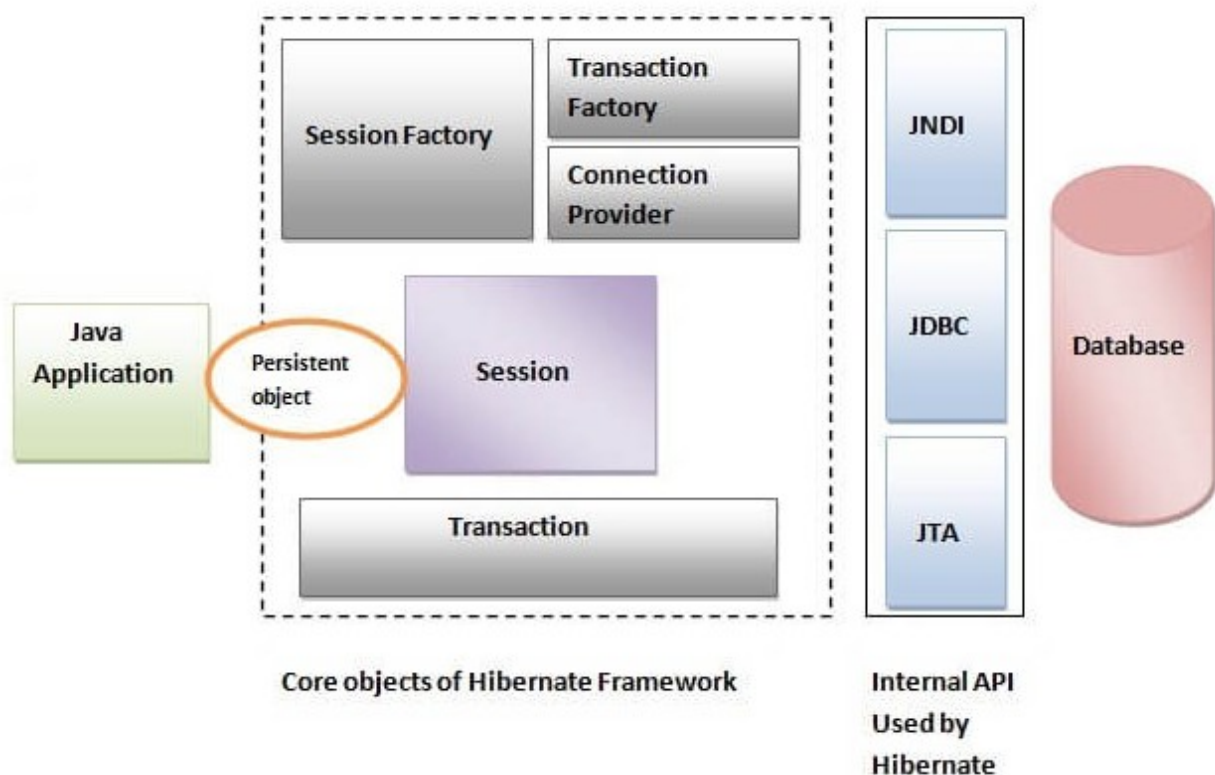
- Java application layer [21]
- Hibernate framework layer [21]
- Backhand api layer [21]
- Database layer [21]

Στην παρακάτω εικόνα παρατίθεται η αρχιτεκτονική του Hibernate στο υψηλό επίπεδο, όπου φαίνεται το mapping file και το configuration file: [21]



Σχ.29: Αρχιτεκτονική του Hibernate στο υψηλό επίπεδο [21]

Το Hibernate Framework χρησιμοποιεί αρκετά αντικείμενα όπως το session factory, το session, το transaction, μαζί με το υπάρχον Java API όπως το JDBC (Java Database Connectivity), JTA (Java Transaction API) και το JNDI (Java Naming Directory Interface). [21]



Σχ.30: Αρχιτεκτονική του Hibernate σε χαμηλότερο επίπεδο [21]

Στο παραπάνω σχήμα παρουσιάζεται αναλυτικότερα η αρχιτεκτονική του Hibernate και τα δομικά της στοιχεία.

Δομικά στοιχεία της αρχιτεκτονικής του Hibernate:

- **SessionFactory:** Το SessionFactory παράγει το session και είναι ο client για το ConnectionProvider. Κρατά προεραϊτικά δευτέρου επιπέδου κρυφή μνήμη με τα δεδομένα (second level cache). [21]
- **Session:** Το αντικείμενο Session παρέχει μια διεπαφή ανάμεσα στην εφαρμογή και τα δεδομένα που αποθηκεύονται στην βάση δεδομένων. Η διάρκεια ζωής του αντικειμένου Session είναι σύντομη και σε αυτό εγκολπίζεται η σύνδεση του JDBC. Παράλληλα στο αντικείμενο Session παράγεται το Transaction, το Query και το Criteria. Το Session κρατά υποχρεωτικά πρώτου επιπέδου κρυφή μνήμη με τα δεδομένα (first level cache). Η org.hibernate.Session διεπαφή (interface), παρέχει μεθόδους για εισαγωγή (insert), ενημέρωση (update) και διαγραφή (delete) του αντικειμένου.

Παρέχει επίσης τις μεθόδους για Transaction, Query, Criteria που παράγει (factory methods). [21]

- **Transaction:** Το αντικείμενο Transaction καθορίζει την ατομική μονάδα εργασίας (atomic unit of work). Είναι προαιρετικό. Η org.hibernate.Transaction διεπαφή (interface) παρέχει μεθόδους για την διαχείριση του Transaction. [21]
- **ConnectionProvider:** Ο ConnectionProvider παράγει τις συνδέσεις του JDBC (JDBC connections). Ο ConnectionProvider οδηγεί την εφαρμογή από τον DriverManager ή το DataSource, σε ένα γενικευμένο επίπεδο αφαίρεσης. Είναι προαιρετικός. [21]
- **TransactionFactory:** Το TransactionFactory παράγει το Transaction. Είναι προαιρετικό. [21]

Στο server side του συστήματος διαχείρισης παρουσιών, αξιοποιούνται τα περισσότερα από τα προαναφερθέντα οφέλη του Hibernate κάνοντας την υλοποίηση του JPA ελαφρύτερη, αποδοτικότερη και αποτελεσματικότερη · ως προς την ανάπτυξη και διαχείριση των διαδικασιών που αφορούν το επίπεδο πρόσβασης στα δεδομένα (data access layer) και την επικοινωνία με την βάση δεδομένων. *Η έκδοση που χρησιμοποιείται είναι η 5.4.30.Final.*

3.1.9. MySQL

Η MySQL είναι το δημοφιλέστερο λογισμικό ανοικτού κώδικα για την διαχείριση σχεσιακών βάσεων δεδομένων. Είναι γρήγορο, κλιμακωτά επεκτάσιμο (scalable), και εύχρηστο και αναπτύσσεται από την εταιρεία MySQL AB σε γλώσσα προγραμματισμού C και C++. Υποστηρίζει επίσης αρκετά λειτουργικά συστήματα όπως Windows, Linux, MacOS με τις γλώσσες C, C++ και Java. [24]

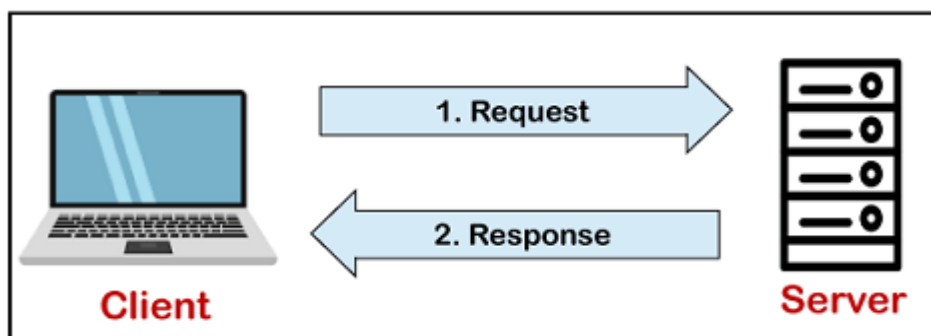
Η MySQL είναι ένα λογισμικό της κατηγορίας συστήματος διαχείρισης σχεσιακής βάσης δεδομένων (RDBMS software) που προσφέρει παροχές όπως: [24]

- Επιτρέπει στον χρήστη την υλοποίηση εντολών σε πίνακες, γραμμές, στήλες και δείκτες.
- Ορίζει τις συσχετίσεις της βάσης δεδομένων με την μορφή πινάκων.
- Παρέχει ακεραιότητα στις αναφορές μεταξύ γραμμών και στηλών των διαφόρων πινάκων.
- Επιτρέπει στον χρήστη την ενημέρωση στους δείκτες των πινάκων αυτόματα.

- Χρησιμοποιεί πολλά SQL ερωτήματα και συνδυάζει χρήσιμες πληροφορίες από πολλαπλούς πίνακες για τον τελικό χρήστη.

Αρχιτεκτονική MySQL:

Η MySQL ακολουθεί την λειτουργία της αρχιτεκτονικής πελάτη - εξυπηρετητή (Client - Server Architecture). Το μοντέλο αυτό έχει σχεδιαστεί για τους τελικούς χρήστες (end-users) που ονομάζονται πελάτες (clients), ώστε να έχουν πρόσβαση στους πόρους απο έναν κεντρικό υπολογιστή γνωστό ως διακομιστή (server), ο οποίος χρησιμοποιεί τις υπηρεσίες του δικτύου. [24]



Σχ.31: Αρχιτεκτονική Πελάτη – Εξυπηρετητή [24]

Για την απευθείας επικοινωνία με τον MySQL server ο χρήστης μπορεί να χρησιμοποιήσει το κατάλληλο MySQL GUI (Graphical User Interface). Με αυτό μπορεί να κάνει requests και να λάβει την κατάλληλη απάντηση από τον server. Κατα την υλοποίηση του συστήματος διαχείρισης παρουσιών για την πραγμάτωση αυτού του τύπου επικοινωνίας με τον server στο επίπεδο της βάσης δεδομένων, χρησιμοποιήθηκε ως GUI το MySQL Workbench.

Ο πυρήνας της MySQL βάσης δεδομένων είναι ο MySQL Server. Αυτός ο server διατίθεται ως ξεχωριστό πρόγραμμα και είναι υπεύθυνος για όλες τις οδηγίες, δηλώσεις (statements) ή εντολές της βάσης δεδομένων. [24]

Λειτουργία MySQL:

Η λειτουργία της MySQL βάσης δεδομένων με MySQL Server έχει ως εξής:

- Η MySQL δημιουργεί μια βάση δεδομένων που επιτρέπει στον χρήστη να φτιάχνει πίνακες για να αποθηκεύει και να διαχειρίζεται δεδομένα και να ορίζει τις συσχετίσεις μεταξύ των πινάκων. [24]
- Οι πελάτες μπορούν να κάνουν requests μέσω του GUI ή της γραμμής εντολών χρησιμοποιώντας ειδικές SQL εκφράσεις στην MySQL. [24]
- Τέλος, η server εφαρμογή θα αποκριθεί αναλόγως και θα παράξει το επιθυμητό αποτέλεσμα στο client-side. [24]

Πλεονεκτήματα MySQL:

- Είναι βάση δεδομένων ανοικτού κώδικα και διατίθεται προς χρήση εντελώς δωρεάν. [24]
- Είναι πολύ δυνατό πρόγραμμα το οποίο μπορεί να διαχειρίζεται ένα σύνολο λειτουργιών των πιο ακριβών και ισχυρών πακέτων. [24]
- Είναι παραμετροποιήσιμη επειδή είναι βάση δεδομένων ανοικτού κώδικα και η ειδική άδεια ανοικτού κώδικα που παρέχει διευκολύνει τους προγραμματιστές να τροποποιήσουν το SQL λογισμικό σύμφωνα με τις ανάγκες που επιτάσσει το δικό τους εξειδικευμένο περιβάλλον. [24]
- Είναι γρηγορότερη από άλλες βάσεις δεδομένων οπότε μπορεί δουλέψει αποτελεσματικά με μεγάλο όγκο δεδομένων. [24]
- Υποστηρίζει πολλά λειτουργικά συστήματα με πολλές γλώσσες όπως PHP, PERL, C, C++, Java. [24]
- Χρησιμοποιεί μια πρότυπη μορφή (standard form) της SQL. [24]
- Υποστηρίζει μεγάλες βάσεις δεδομένων και ακόμη περισσότερες από 50 εκατομμύρια γραμμές σε έναν πίνακα. Το προκαθορισμένο όριο μεγέθους για τον πίνακα είναι 4GB, αλλά γίνεται να αυξηθεί. Πιο συγκεκριμένα αν ένα λειτουργικό σύστημα μπορούσε να το αντέξει, το όριο μεγέθους ανα πίνακα θα μπορούσε θεωρητικά να αυξηθεί στα 8 εκατομμύρια terabytes (TB). [24]

Στο server side του συστήματος διαχείρισης παρουσιών χρησιμοποιήθηκε το λογισμικό βάσεων δεδομένων MySQL και συγκεκριμένα η έκδοση 8.0.23.

3.1.10. Spring Security

Το **Spring Security** είναι ένα ισχυρό και εξαιρετικά προσαρμόσιμο **framework** για την εξουσιοδότηση (authentication) και τον έλεγχο πρόσβασης των χρηστών μιας εφαρμογής. Είναι εκ των πραγμάτων το πρότυπο για την ασφάλεια στις εφαρμογές βασισμένες σε Spring. [25] Το Spring Security Framework εστιάζει στην παροχή τόσο αυθεντικοποίησης (authentication), όσο και εξουσιοδότησης (authorization) στις εφαρμογές βασισμένες σε Java. [25]

Για παράδειγμα εάν ένας χρήστης επιθυμεί πρόσβαση στο σύστημα ή σε πόρους του συστήματος το Spring Security αναγνωρίζει και ταυτοποιεί τον χρήστη που επιθυμεί την πρόσβαση (authentication). Στην συνέχεια όμως ελέγχει, αν είναι εξουσιοδοτημένος για να υλοποιήσει ενέργειες που έχουν να κάνουν με πόρους του συστήματος (authorization).

Όπως σε όλες τις υπηρεσίες του Spring, η αληθινή δύναμη του Spring Security βρίσκεται στο πόσο εύκολα μπορεί να επεκταθεί για να καλύψει προσαρμοσμένες απαιτήσεις. [25]

Παράλληλα το Spring Security Framework είναι αρκετά ευέλικτο στο κομμάτι που αφορά την εξουσιοδότηση των χρηστών (authentication). Ετσι μπορεί να ενσωματωθεί με οποιοδήποτε λογισμικό μπορεί να την υλοποιήσει. Ως εκ τούτου στην περίπτωση που ο προγραμματιστής θελήσει να το ενσωματώσει με ένα παλαιότερο σύστημα που δεν ακολουθεί κανένα πρότυπο ασφάλειας, και σε αυτή την περίπτωση, το Spring Security θα λειτουργήσει κανονικά. [26]

Πλεονεκτήματα του Spring Security:

Το Spring Security έχει αρκετά πλεονεκτήματα κάποια απο τα οποία αναφέρονται παρακάτω:

- Ολοκληρωμένη υποστήριξη για έλεγχο ταυτότητας (authentication) και εξουσιοδότηση (authorization). [25]
- Προστασία ενάντια σε επιθέσεις όπως session fixation, clickjacking ή cross site request forgery. [25]

- Ενσωμάτωση Servlet API. [25]
- Προαιρετική ενσωμάτωση με το Spring Web MVC. [25]
- Δυνατότητα μεταφοράς. [26]
- Υποστήριξη Java παραμετροποίησης [26]

Στο server side του συστήματος διαχείρισης παρουσιών το Spring Security χρησιμοποιήθηκε:

- Για τον έλεγχο της ταυτότητας των χρηστών που επιθυμούν να συνδεθούν στο σύστημα ή να αποκτήσουν πρόσβαση σε πόρους του, σε συνδυασμό με το Json Web Token (JWT) η λειτουργία του οποίου αναλύεται στο επόμενο κεφάλαιο.
- Για την προστασία των πόρων της εφαρμογής με την παραμετροποίηση ειδικού φίλτρου στο οποίο:
 - Αποκλείστηκαν οι csrf (cross site request forgery) επιθέσεις.
 - Ορίστηκε η γενική εξουσιοδότηση των κεντρικών url ανα controller και συγκεκριμένα επιτράπηκαν όλοι οι ρόλοι καθώς για μεγαλύτερη ευελιξία γίνεται έλεγχος εξουσιοδότησης σε επίπεδο μεθόδου σε κάθε Controller.
 - Ορίστηκε για κάθε request να απαιτείται αυθεντικοποίηση.
 - Εγιναν κατάλληλες ρυθμίσεις με τις οποίες φιλτράρονται τα requests με βάση το username και το password του χρήστη με ειδικό Json Web Token (JWT).
- Για την εξουσιοδότηση των χρηστών πρώτου αποκτήσουν πρόσβαση σε πόρους του συστήματος. Συγκεκριμένα χρησιμοποιήθηκε το annotation **@PreAuthorize** του Spring Security σε επίπεδο μεθόδων στους Controllers της εφαρμογής.

Μέσω του **@PreAuthorize** annotation ορίζεται το ποιός ή ποιοι ρόλοι μπορούν να έχουν πρόσβαση στους αντίστοιχους πόρους. Παράλληλα όπου απαιτείται, γίνεται έλεγχος για το αν ο συνδεδεμένος στο σύστημα χρήστης είναι εκείνος που ζητάει την πρόσβαση στον συγκεκριμένο πόρο.

Ο έλεγχος για το αν ο συνδεδεμένος χρήστης κατέχει τον αντίστοιχο ρόλο έγινε με την έκφραση **hasRole** του Spring Security. Η έκφραση **hasRole** επιστρέφει **true** αν ο συνδεδεμένος χρήστης έχει τον συγκεκριμένο ρόλο.

Ο έλεγχος για το αν ο συνδεδεμένος χρήστης είναι εκείνος που επιθυμεί την πρόσβαση σε κάποιον πόρο ή την υλοποίηση κάποιας ενέργειας σχετικά με αυτόν, έγινε με την έκφραση **authentication** (που επιτρέπει την απευθείας πρόσβαση στο τρέχον **Authentication αντικείμενο** που λαμβάνεται από το **SecurityContext**) και συγκεκριμένα με το **principal** του **authentication** το οποίο επιτρέπει την απευθείας πρόσβαση στο

principal object που αναπαριστά τον συνδεδεμένο χρήστη. Ο έλεγχος γίνεται με το `userId` του χρήστη.

Παράδειγμα:

```
@PostMapping("/create/{userId}")  
  
@PreAuthorize("(hasRole('ADMIN') or hasRole('STUDENT'))"  
              + " and #userId == authentication.principal.id")  
  
public ResponseEntity<ExcuseApplication> createExcuseApplication( ... )
```

- Για την υλοποίηση πιο **εξειδικευμένων ελέγχων** και την **πιστοποίηση των δικαιωμάτων των συνδεδεμένων χρηστών** ανάλογα με τις απαιτήσεις του συστήματος.

Για παράδειγμα τέτοιου είδους έλεγχος θα μπορούσε να είναι ο έλεγχος που αφορά το αν ο συνδεδεμένος καθηγητής ανήκει στο διδακτικό προσωπικό που αφορά ένα πρόγραμμα μαθήματος.

Για αυτή την συνθετότερη περίπτωση χρειάστηκε να γίνει επέκταση του Spring Data (Spring Data Integration), ως εξής:

Με την έκθεση του **SecurityEvaluationContextExtension Java Bean** στην κλάση **WebSecurityConfig** στην οποία γίνονται οι παραμετροποιήσεις για το Spring Security, οι κοινές εκφράσεις ασφαλείας του Spring Security έγιναν διαθέσιμες και για τα ερωτήματα που πραγματοποιούνται μέσω του **@Query** του Spring Data.

Έτσι στην εφαρμογή μπορούν να υλοποιούνται έλεγχοι του Spring Security που αφορούν τον συνδεδεμένο χρήστη, μέσα σε ερωτήματα (queries) μέσω του **@Query** annotation του Spring Data. Τα queries αυτά γίνονται σε επίπεδο μεθόδου στα αποθετήρια (repositories) της εφαρμογής καθώς έτσι είναι ταχύτερα και αποδοτικότερα (όπως αναφέρεται στην ενότητα 3.1.7).

Παράδειγμα:

```
@Query(value = "SELECT cs FROM CourseSchedule as cs JOIN cs.teachingStuff as user  
WHERE cs.id=:courseScheduleId and user.id = ?#{principal?.id}")  
  
CourseSchedule checkOwnershipByCourseScheduleId(Long courseScheduleId);
```

Στο server-side της εφαρμογής διαχείρισης παρουσιών χρησιμοποιείται η έκδοση 2.4.5 του Spring Security.

3.1.11. Json Web Token

Το **JSON Web Token (JWT)** είναι ένα πρότυπο (RFC 7519) ανοικτού κώδικα που ορίζει έναν συμπαγή και αυτοτελή τρόπο για την ασφαλή μετάδοση πληροφοριών μεταξύ των επικοινωνούντων μερών με την μορφή ενός αντικειμένου τύπου JSON. [28]

Οι πληροφορίες αυτές μπορούν να επαληθευτούν και να είναι αξιόπιστες αφού είναι **ψηφιακά υπογεγραμμένες (digitally signed)**. Τα JWT μπορούν να υπογραφούν χρησιμοποιώντας ένα μυστικό κλειδί (μέσω του HMAC αλγορίθμου) ή ένα ζεύγος δημόσιου/ιδιωτικού κλειδιού (public/private key) χρησιμοποιώντας RSA (κρυπταλγόριθμος ασύμμετρου κλειδιού που μπορεί να χρησιμοποιηθεί και ως ψηφιακή υπογραφή) ή ECDSA (αλγόριθμος ψηφιακής υπογραφής ελλειπτικής καμπύλης). [28]

Τα ψηφιακά υπογεγραμμένα (digitally signed) tokens μπορούν να επαληθεύσουν την ακεραιότητα των claims (δηλώσεις που αφορούν συνήθως δεδομένα του χρήστη αλλά και πρόσθετα δεδομένα) που περιέχονται σε αυτά. Όταν τα tokens υπογράφονται χρησιμοποιώντας ζεύγη δημόσιου/ιδιωτικού κλειδιού, η υπογραφή πιστοποιεί επίσης ότι μόνο το μέρος που κατέχει το ιδιωτικό κλειδί είναι αυτό που το υπέγραψε. [28]

Χρησιμότητα του JWT

Τα JSON Web Tokens χρησιμοποιούνται κυρίως για:

- **Εξουσιοδότηση (Authorization):** Αυτό είναι το πιο κοινό σενάριο στο οποίο χρησιμοποιείται το JWT. [28] Με το που συνδεθεί ο χρήστης στο σύστημα, κάθε επόμενο request θα περιλαμβάνει το JWT, επιτρέποντας στον χρήστη την πρόσβαση σε routes, υπηρεσίες και πόρους με βάση αυτό το token. [28]
- **Ανταλλαγή πληροφοριών (Information Exchange):** Τα JSON Web Tokens είναι ένας καλός τρόπος για την μεταφορά δεδομένων μεταξύ επικοινωνούντων μερών. Επειδή τα JWTs μπορούν να υπογραφούν - χρησιμοποιώντας για παράδειγμα ζεύγη δημόσιου/ιδιωτικού κλειδιού - μπορεί κανείς να είναι σίγουρος ότι οι αυτοί που κάνουν requests είναι αυτοί που ισχυρίζονται ότι είναι. Επιπροσθέτως, καθώς η υπογραφή υπολογίζεται μέσω του header και του payload (δομικών στοιχείων του JWT

που θα αναλυθούν παρακάτω), μπορεί να επαληθευτεί ότι το περιεχόμενό της δεν έχει αλλοιωθεί. [28]

Δομή του JWT

Τα JSON Web Tokens αποτελούνται από τρία μέρη χωρισμένα με τελείες (.), τα οποία είναι τα εξής: [28]

- Header
- Payload
- Signature

Ως εκ τούτου, ένα JWT έχει συνήθως την ακόλουθη μορφή:



xxxxxx.yyyyy.zzzzz

Εικ.1. Μορφή του Json Web Token [28]

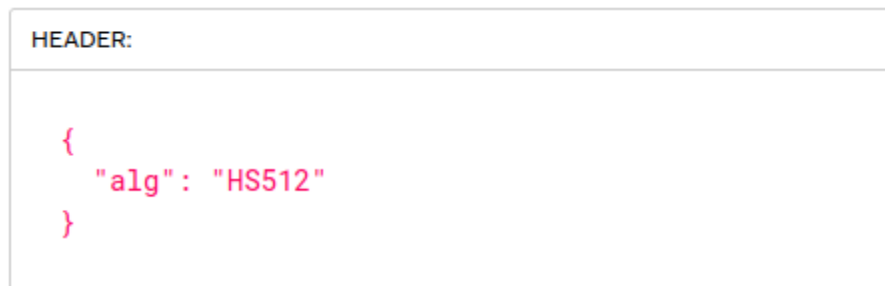
Παρακάτω αναλύονται διεξοδικά τα μέρη που συναποτελούν τα δομικά στοιχεία του JWT:

Επικεφαλίδα (Header)

Ο header τυπικά αποτελείται από δύο μέρη: τον τύπο του token, ο οποίος είναι JWT, και τον αλγόριθμο υπογραφής που χρησιμοποιείται, όπως είναι οι HMAC, SHA256, SHA512 ή ο RSA. [28]

Στο server side του συστήματος διαχείρισης παρουσιών στην παραμετροποίηση του JWT χρησιμοποιείται ο αλγόριθμος SHA512.

Ενα Header του json web token ενός χρήστη του συστήματος διαχείρισης παρουσιών φαίνεται στην παρακάτω εικόνα:



Εικ.2. Json Web Token Header

Το JSON που αποτελεί το Header, όπως είναι το παραπάνω, είναι κωδικοποιημένο με βάση το Base64Url και αποτελεί το πρώτο τμήμα του JWT. [28]

Κύρια πληροφορία (Payload)

Το δεύτερο τμήμα του token είναι το payload, το οποίο περιέχει τα claims. Τα claims αποτελούν δηλώσεις που αφορούν μια οντότητα (συνήθως τον χρήστη) αλλά και πρόσθετα δεδομένα. [28]

Στην περίπτωση του συστήματος διαχείρισης παρουσιών το payload δομείται από τις εξής πληροφορίες:

- Το **username** του εκάστοτε χρήστη για τον οποίο δημιουργείται.
- Την **στιγμή της δημιουργίας του json web token** υπολογισμένη με βάση την στιγμή έναρξης του unix.
- Την **στιγμή λήξης του json web token** που ορίζεται στην παραμετροποίηση του server-side.

Υπάρχουν τρεις τύποι claim: **registered**, **public** και **private claims** οι οποίοι αναλύονται παρακάτω: [28]

- **Registered claims:** Πρόκειται για ένα σύνολο προκαθορισμένων claims, τα οποία δεν είναι υποχρεωτικά, αλλά συνιστώνται ώστε να παρέχεται ένα σύνολο χρήσιμων διαλειτουργικών claims. Μερικά registered claims είναι τα εξής:
 - iss (issuer)
 - exp (expiration time, αφορά τον χρόνο λήξης του token)

- sub (subject, δείχνει σε ποιόν αναφέρεται το token)
- aud (audience)
- **Public claims:** Τα public claims μπορούν να οριστούν κατα βούληση από όσους χρησιμοποιούν JWT. Για να αποφευχθούν οι συγκρούσεις (collisions), θα πρέπει να ορίζονται στο IANA JSON Web Token Registry ή να ορίζονται ως URI.
- **Private claims:** Πρόκειται για προσαρμοσμένα (custom) claims που δημιουργούνται για την ανταλλαγή πληροφοριών μεταξύ των επικοινωνούντων μερών που συμφωνούν στην χρήση τους και δεν είναι ούτε registered, ούτε public claims.

Ενα παράδειγμα payload ενός json web token ενός χρήστη του συστήματος διαχείρισης παρουσιών φαίνεται στην παρακάτω εικόνα:



Εικ.3. Json Web Token Payload

Το JSON που αποτελεί το Payload, είναι κωδικοποιημένο με βάση το Base64Url και αποτελεί το δεύτερο τμήμα του JWT. [28]

Υπογραφή (Signature)

Η υπογραφή (**signature**) αποτελεί τον συγκερασμό της κωδικοποιημένης επικεφαλίδας (**header**), του κωδικοποιημένου **payload**, ένα μυστικό κλειδί (**secret key**) και τον αλγόριθμο που έχει καθοριστεί στην επικεφαλίδα (header). [28]

Ενα παράδειγμα υπογραφής (**signature**) του json web token ενός χρήστη του συστήματος διαχείρισης παρουσιών, για την κρυπτογράφηση του οποίου χρησιμοποιήθηκε ο αλγόριθμος SHA512, φαίνεται στην παρακάτω εικόνα:

VERIFY SIGNATURE

```

HMACSHA512(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded

```

Εικ.4. Json Web Token Signature

Η υπογραφή (signature) χρησιμοποιείται για να επαληθεύσει ότι το μήνυμα δεν άλλαξε στην πορεία και στην περίπτωση των json web tokens με ιδιωτικό κλειδί μπορεί να επαληθευτεί ότι ο αποστολέας του JWT είναι αυτός που ισχυρίζεται ότι είναι. [28]

Συνδυάζοντας το προηγούμενο **header**, **payload** και **signature** έχουμε το κωδικοποιημένο json web token που φαίνεται στην παρακάτω εικόνα:

```

eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0c2FkaW1hcyIsIm1hdCI6MTY0MjUzMjIzMiwiZXhwIjoxNjQyNjA0MjMyfQ.ty5coGQY9vs2Am8SREB6CcmXrT-QUX4SHXMzvaHma8tpQn-yI_044VzY03bBHswyk28EHGTmbtyERmk353ymGQ

```

Εικ.5. Κωδικοποιημένο Json Web Token

Στην παρακάτω εικόνα φαίνεται το json web token που αναλύθηκε προηγουμένως τόσο σε κωδικοποιημένη, όσο και σε αποκωδικοποιημένη μορφή:

Encoded

```
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ0c2FkaW1hcyIsIm1hdCI6MTY0MjUzMjIzMiwiaXhwIjoxNjQyNjA0MjMyfQ.ty5coGQY9vs2Am8SREB6CcmXrT-QUX4SHXMzvaHma8tpQn-yI_044VzY03bBHswyk28EHGTmbtyERmk353ymGQ
```

Decoded

HEADER:

```
{
  "alg": "HS512"
}
```

PAYLOAD:

```
{
  "sub": "tsadimas",
  "iat": 1642532232,
  "exp": 1642604232
}
```

VERIFY SIGNATURE

```
HMACSHA512(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```

Εικ.6. Json Web Token – Κωδικοποιημένο και Αποκωδικοποιημένο

Τρόπος λειτουργίας των Json Web Tokens:

Αφού πραγματοποιηθεί ο έλεγχος ταυτότητας του χρήστη που θέλει να συνδεθεί στο σύστημα και αφού συνδεθεί επιτυχώς χρησιμοποιώντας τα διαπιστευτήριά (credentials) του, θα του επιστραφεί ένα JSON Web Token. [28]

Οποτε ο χρήστης επιθυμεί να έχει πρόσβαση σε προστατευμένους πόρους του συστήματος, το client-side με το οποίο αλληλεπιδρά στέλνει το JWT συνήθως μέσω ενός Authorization header χρησιμοποιώντας το πρόθεμα Bearer. [28] Με αυτόν τον τρόπο αποκτά πρόσβαση σε προστατευμένους πόρους της εφαρμογής και ο χρήστης του συστήματος διαχείρισης παρουσιών.

Στην παρακάτω εικόνα φαίνεται πως μοιάζει το περιεχόμενο του header στην μορφή που χρησιμοποιείται στην εφαρμογή:

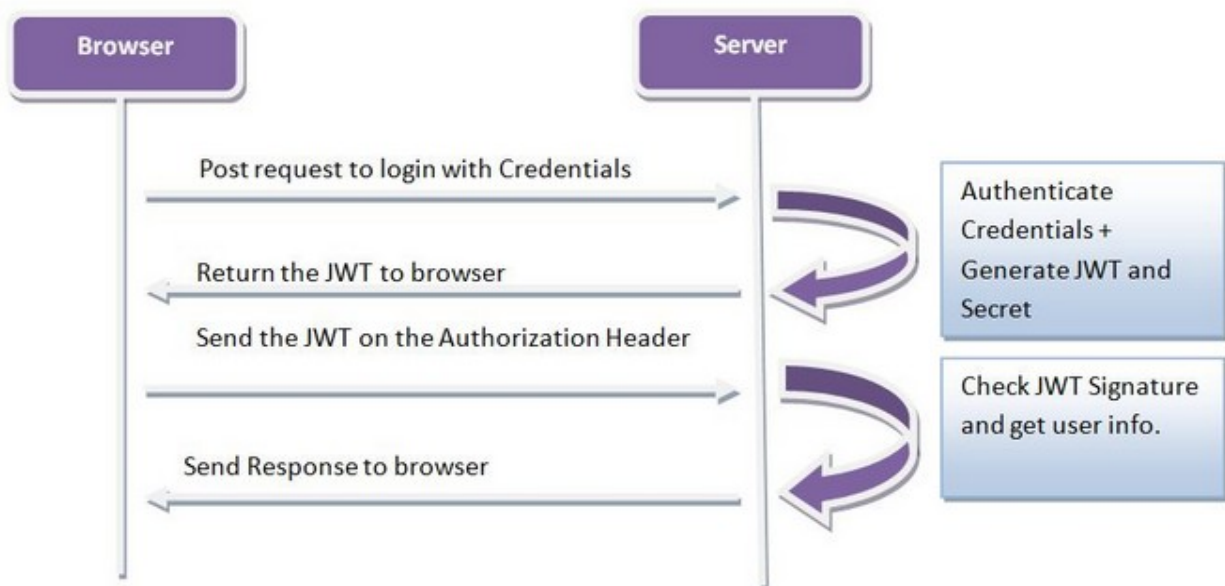
```
Authorization: Bearer <token>
```

Εικ.7. Μορφή Authorization Header [28]

Μετά την αποστολή του Authorization header με το πρόθεμα bearer, στο server-side μπορεί να το περιμένει ένας stateless μηχανισμός εξουσιοδότησης [28], πράγμα που συμβαίνει και στο σύστημα διαχείρισης παρουσιών.

Επειτα το Authorization header ελέγχεται κατάλληλα από αυτόν τον μηχανισμό και αν το περιεχόμενό του είναι κατάλληλο τότε το σύστημα δίνει την απαιτούμενη από τον χρήστη πρόσβαση στους προστατευμένους πόρους. [28]

Στην παρακάτω εικόνα περιγράφεται συνοπτικά ο βασικός τρόπος λειτουργίας ενός json web token:



Σχ.32: Τρόπος λειτουργίας ενός Json Web Token [53]

Επειδή τα JWTs είναι διαπιστευτήρια (credentials) είναι καλό να δίνεται σε αυτά μικρή διάρκεια ζωής για να περιορίζονται τα πιθανά προβλήματα ασφαλείας. [28]

Για αυτό τον λόγο το server-side του συστήματος διαχείρισης παρουσιών έχει υλοποιηθεί ώστε να παρέχει για κάθε χρήστη που συνδέεται στο σύστημα **δύο json web tokens**:

Ενα **jwt access-token** το οποίο είναι το βασικό token του χρήστη και διαρκεί 5 λεπτά, και ένα **jwt refresh-token** το οποίο είναι συμπληρωματικού χαρακτήρα και διαρκεί 10 λεπτά.

Μόλις το access-token λήξει και ο χρήστης προσπαθήσει να κάνει κάποια ενέργεια στο σύστημα, εκείνο ελέγχει αν είναι ακόμη ενεργό το refresh-token. Σε περίπτωση που το refresh-token είναι ακόμη ενεργό, το σύστημα δίνει πρόσβαση στον χρήστη παρέχοντάς του ένα νέο jwt access-token αλλά επίσης και ένα εξίσου νέο refresh-token. Αντιθέτως αν το refresh-token έχει και αυτό λήξει, τότε το σύστημα αποσυνδέει αυτόματα τον χρήστη από τον λογαριασμό του. *Στο κεφάλαιο 2 στην ενότητα 2.3.3 περιγράφεται αναλυτικά η παραπάνω διαδικασία του συστήματος.*

Πλεονεκτήματα του JWT:

- **Είναι συμπαγές** - Μπορεί να σταλεί μέσω URL/Post request/HTTP Header πράγμα που το καθιστά γρήγορο στην μετάδοση και εύχρηστο. [53]
- **Είναι αυτόνομο** - Περιέχει τα στοιχεία του χρήστη οπότε δεν χρειάζεται να πραγματοποιηθεί περεταίρω επικοινωνία με την βάση για την λήψη στοιχείων του χρήστη. [53]
- **Είναι ψηφιακά υπογεγραμμένο** - Οι πληροφορίες του επαληθεύονται και είναι αξιόπιστες. [53]
- **Παρέχει ασφαλή μεταφορά πληροφοριών** μεταξύ δύο φορέων (όπως μεταξύ του client και του server). [53]
- **Είναι σε μορφή Json** - Μπορεί να χρησιμοποιηθεί στις περισσότερες γλώσσες προγραμματισμού, καθώς μπορεί να αντιστοιχιστεί απευθείας στο αντίστοιχο αντικείμενο. [53]
- Δέχεται την χρήση τόσο τον αλγόριθμο HMAC όσο και ένα ζεύγος δημόσιου/ιδιωτικού κλειδιού (public/private key) σε συνδυασμό με το μυστικό κλειδί, για να υπογραφεί ψηφιακά. [53]

Στο server-side του συστήματος διαχείρισης παρουσιών το Json Web Token χρησιμοποιήθηκε με τον τρόπο που αναφέρθηκε παραπάνω και σε συνδυασμό με το Spring Security συνέβαλε στην διεκπεραίωση του επιπέδου της εφαρμογής που αφορά την ασφάλεια. *Η έκδοση του json web token που χρησιμοποιήθηκε είναι η 0.9.1.*

3.2. Τεχνολογίες Client

Στα παρακάτω υποκεφάλαια παρατίθενται οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη του client side της εφαρμογής.

3.2.1. Angular Framework

Η Angular είναι ένα ανοικτού κώδικα framework γραμμένο στην γλώσσα TypeScript. Το Angular Framework συντηρείται από την Google και κύριος σκοπός του είναι η ανάπτυξη single-page εφαρμογών. Ως framework η Angular έχει πολλά πλεονεκτήματα καθώς επίσης παρέχει μια πρότυπη δομή (standard structure) που δίνει την δυνατότητα στους προγραμματιστές να δημιουργούν μεγάλες εφαρμογές εύκολα συντηρίσιμες. [29]

Ωστόσο στην πραγματικότητα η Angular είναι κάτι πολύ περισσότερο απο ένα framework: Είναι μια ολόκληρη πλατφόρμα που παρέχει τα εργαλεία και την κατάλληλη δομή στους προγραμματιστές ώστε να γράφουν καθαρό κώδικα γρηγορότερα και ευκολότερα. [33]

Η Angular ικανοποιεί την δημιουργία εφαρμογών σε ένα ευρύ φάσμα, όπως εφαρμογών ιστού (web) ή κινητών συσκευών. Παρα ταύτα είναι ιδιαίτερα αποτελεσματική για συγκεκριμένες λύσεις σε επίπεδο λογισμικού οι οποίες αναλύονται παρακάτω:

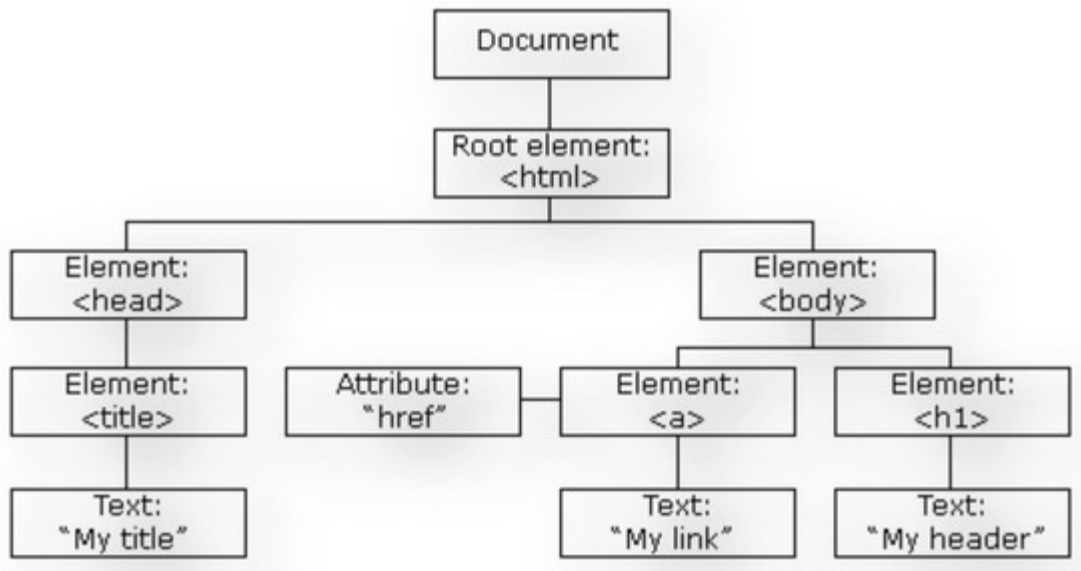
- **Μεγάλες επιχειρηματικές εφαρμογές (Large Enterprise Apps):** Η Angular χρησιμοποιείται ευρέως για την δημιουργία συστημάτων μεγάλης κλίμακας που

καλύπτουν τις ανάγκες επιχειρήσεων ή οργανισμών και όχι τις ανάγκες ενός συγκεκριμένου χρήστη. Τέτοιες εφαρμογές βοηθούν στην οργάνωση της ροής εργασιών στις επιχειρήσεις και στην διαχείριση συγκεκριμένων λειτουργιών. Τέτοια συστήματα μπορεί να είναι για παράδειγμα CRMs, ERP systems, platforms for payment processing, email marketing, content management [33] ή στην παρούσα περίπτωση ένα σύστημα διαχείρισης των παρουσιών των φοιτητών για ένα πανεπιστημιακό ίδρυμα.

- **Single-Page Applications (SPAs):** Οι single-pages εφαρμογές όπως είναι και το client-side του συστήματος διαχείρισης παρουσιών, είναι δυναμικές εφαρμογές. Αυτό σημαίνει ότι όταν ένας χρήστης πλοηγείται στην εφαρμογή, δεν πραγματοποιείται επαναφόρτωση της σελίδας. Τα δεδομένα λαμβάνονται και αποστέλλονται όπως σε κάθε συνηθισμένο σύστημα, αλλά η εφαρμογή δεν χρειάζεται να φορτώνει νέες σελίδες κάθε φορά που ο χρήστης κάνει κάποια ενέργεια στην σελίδα. [33]
- **Progressive Web Applications (PWAs):** Μια progressive Angular εφαρμογή είναι μια οικονομική λύση που επιτρέπει μέσω της χρήσης ειδικών web τεχνολογιών την εκτέλεση εφαρμογών για κινητές συσκευές σε διάφορες πλατφόρμες είτε σε σύνδεση (online) είτε εκτός σύνδεσης (offline). Μια progressive Angular εφαρμογή δουλεύει στο πρόγραμμα περιήγησης (web browser), αλλά συμπεριφέρεται όπως μια κοινή εφαρμογή. Οι Angular progressive web εφαρμογές έχουν μεγάλη απήχηση στο κοινό καθώς οι χρήστες δεν χρειάζεται να χρησιμοποιήσουν το Google Play ή το App Store για να κατεβάσουν την εφαρμογή, αλλά αρχίζουν χρησιμοποιώντας την εφαρμογή απευθείας. [33]
- **Στατικές διαδικτυακές εφαρμογές (Static web pages):** Η Angular είναι διαδεδομένη κυρίως για την δημιουργία δυναμικών ιστοσελίδων. Όμως μέσω της Angular είναι δυνατό να δημιουργηθούν και στατικές ιστοσελίδες για μια διαδικτυακή εφαρμογή με την τεχνολογία Angular Universal. Η Angular Universal είναι μια τεχνολογία για server-side rendering (SSR). Επειδή οι single page εφαρμογές δεν έχουν και τόσο καλή προσαρμογή με τις μηχανές αναζήτησης, δηλαδή με το το κομμάτι του search engine optimization (SEO), το server side rendering που παρέχει η Angular Universal λύνει αυτό το πρόβλημα. [33]

Βασικά χαρακτηριστικά της Angular:

- **Document Object Model:** Το DOM (Document Object Model) αναπαριστά ένα έγγραφο XML ή HTML ως μια δενδρική δομή στην οποία κάθε κόμβος αντιπροσωπεύει ένα μέρος αυτού του εγγράφου. [29]



Σχ.33: Αναπαράσταση του Document Object Model

Η Angular χρησιμοποιεί το DOM. Μάλιστα αν για παράδειγμα γίνονται δέκα ενημερώσεις στην ίδια σελίδα HTML, αντί να ενημερώνει μόνο τα τις ετικέτες (tags) που έχουν ενημερωθεί, η Angular θα ενημερώσει όλα την δενδρική δομή των ετικετών (tags) HTML. [29]

- **TypeScript:** Η TypeScript ορίζει ένα σύνολο τύπων στην JavaScript, το οποίο βοηθά τους χρήστες να γράφουν κώδικα που πιο κατανοητό. Όλος ο κώδικας TypeScript μεταγλωττίζεται σε JavaScript και μπορεί να τρέξει ομαλά και αποτελεσματικά σε οποιαδήποτε πλατφόρμα. [29]

Αν και η Typescript δεν είναι υποχρεωτικό να χρησιμοποιηθεί για την ανάπτυξη Angular εφαρμογών, ωστόσο προτείνεται ανεπιφύλακτα καθώς προσφέρει καλύτερη δομή στο συντακτικό της, κάνοντας παράλληλα τον κώδικα καλώς ορισμένο, καθαρό και κατ' επέκταση εύκολα συντηρήσιμο. [29] *Στο client-side του συστήματος διαχείρισης παρουσιών χρησιμοποιείται η έκδοση 4.2.3 της Typescript.*

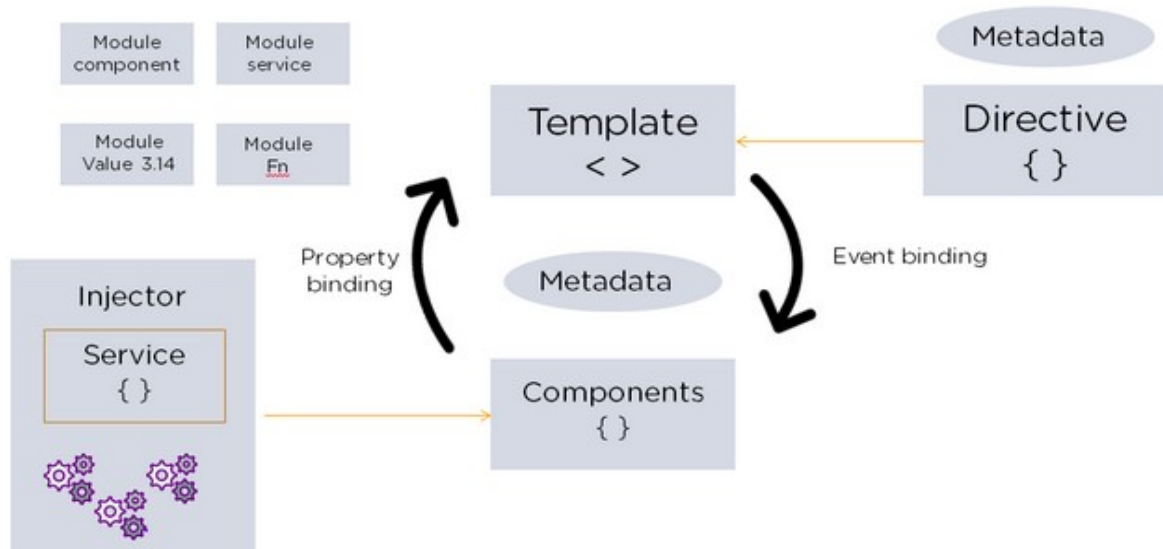
- **Data Binding:** Το Data Binding είναι μια διαδικασία που επιτρέπει στους χρήστες να χειρίζονται τα τμήματα του DOM (δηλαδή τα HTML elements) μιας ιστοσελίδας μέσω του web browser. Το Data Binding γίνεται μέσω δυναμικής HTML και δεν απαιτεί πολύπλοκες προγραμματιστικές πρακτικές. Χρησιμοποιείται σε ιστοσελίδες που περιλαμβάνουν διαδραστικά συστατικά στοιχεία (components) όπως αριθμομηχανές, διαδραστικά μαθήματα, φόρουμ καθώς και παιχνίδια. Τέλος το Data Binding επιτρέπει την καλύτερη και σταδιακή προβολή μιας ιστοσελίδας, όταν εκείνη έχει μεγάλο όγκο δεδομένων. [29]

Η Angular χρησιμοποιεί Data Binding δύο κατευθύνσεων (**two-way data binding**). Πιο συγκεκριμένα στην Angular Model-View-Controller (MVC) αρχιτεκτονική, εάν ο προγραμματιστής αλλάξει κάτι που αφορά το επίπεδο του Model το αντίστοιχο View επίπεδο προσαρμόζεται στις αλλαγές που υπέστη το Model και το αντίστροφο. Εν ολίγοις, αν τα δεδομένα της εφαρμογής αλλάξουν, αλλάζει αναλόγως και το UI. [33] Αυτό το χαρακτηριστικό επιτρέπει στο framework να συνδέει το DOM με τα δεδομένα της Angular εφαρμογής μέσω των αντίστοιχων μηχανισμών. [29] Το **two-way binding** παρέχει στα components της εφαρμογής την δυνατότητα να μοιράζονται δεδομένα. Για παράδειγμα μπορεί να χρησιμοποιηθεί με τρόπο που η εφαρμογή να αντιδρά σε events και να αναβαθμίζει τις αντίστοιχες τιμές ταυτόχρονα μεταξύ parent και child components. [31]

- **Testing:** Η Angular χρησιμοποιεί το Jasmine testing framework. Το Jasmine framework παρέχει πολλαπλές λειτουργικότητες για την δημιουργία διαφορετικών ειδών τμημάτων κώδικα που σκοπό έχουν να ελέγχουν την εύρυθμη λειτουργία τμημάτων της εφαρμογής (test cases). Οι διεργασίες για τα test γίνονται μέσω ενός εργαλείου αυτοματοποίησης για test (testing automation tool) που ονομάζεται Karma. Το Karma χρησιμοποιεί ένα αρχείο που περιέχει τις ρυθμίσεις σχετικά με το πότε θα ξεκινήσουν τα test, σχετικές αναφορές για τα test καθώς και το testing framework που χρησιμοποιείται. [29] *Στο client-side του συστήματος διαχείρισης παρουσιών χρησιμοποιείται η έκδοση 6.3.0 του Karma και η έκδοση 3.8.0 του Jasmine.*

Αρχιτεκτονική της Angular

Η Angular χρησιμοποιεί το μοντέλο αρχιτεκτονικής λογισμικού MVC (model-view-controller). Είναι δηλαδή ένα MVC framework. Η Angular παρέχει σαφείς οδηγίες σχετικά με τον τρόπο με τον οποίο θα πρέπει να δομηθεί η εφαρμογή και προσφέρει αμφίδρομη ροή δεδομένων παρέχοντας παράλληλα παραγματικό DOM. [29]



Σχ.34: Αρχιτεκτονική της Angular [29]

Παρακάτω παρατίθενται τα πιο βασικά δομικά στοιχεία μιας Angular εφαρμογής:

- **Modules:** Το Angular module είναι ένας μηχανισμός που ομαδοποιεί τα διάφορα components, directives, pipes και services τα οποία σχετίζονται μεταξύ τους. Η ομαδοποίηση αυτή, γίνεται με τέτοιο τρόπο ώστε το κάθε module να μπορεί να συνδυαστεί με άλλα modules για την δημιουργία μιας εφαρμογής. [30] Το βασικότερο Angular module είναι το root module που ονομάζεται AppModule και παρέχει τον μηχανισμό εκκίνησης της εφαρμογής. [29]
- **Components:** Τα Components αποτελούν το κύριο συστατικό στοιχείο των Angular εφαρμογών. Κάθε component αποτελείται από: [31]
 - Ένα HTML template που υποδηλώνει τι ρεντάρεται στην σελίδα.
 - Μια κλάση Typescript που ορίζει την συμπεριφορά του.
 - Προερατικά, CSS styles τα οποία εφαρμόζονται στο HTML template.

Ετσι κάθε component σε μια εφαρμογή, ορίζει μια κλάση που περιέχει δεδομένα καθώς και την λογική της εφαρμογής (application logic). Γενικά ένα component ορίζει ένα τμήμα του UI (user interface). [30]

- **Templates:** Ένα template είναι ένα κομμάτι HTML το οποίο καθοδηγεί το Angular Framework στο πως να προετοιμάσει την προβολή (δηλαδή πως να κάνει render) του αντίστοιχου component στην Angular εφαρμογή. [32]

Στα templates υπάρχουν δύο τρόποι για data binding:

- **Event binding:** Λειτουργία η οποία επιτρέπει στην εφαρμογή να ανταποκρίνεται στην είσοδο του χρήστη (user input) ενημερώνοντας τα αντίστοιχα δεδομένα της εφαρμογής που εμπλέκονται στην διαδικασία. [29]
 - **Property binding:** Το property binding είναι μια λειτουργία που βοηθά τον προγραμματιστή να ορίσει τιμές για τις ιδιότητες (properties) των HTML στοιχείων ή των directives. Μέσω του property binding μπορεί να κατασκευαστούν toggle buttons ή να κοινοποιηθούν τιμές μεταξύ διαφόρων components. [31]
-
- **Metadata:** Τα μεταδεδομένα κατευθύνουν την Angular σχετικά με το τρόπο επεξεργασίας μιας κλάσης. Έτσι εφαρμόζονται σε επίπεδο κλάσης (για παράδειγμα ως decorators) ώστε να προσδιορίσουν την αναμενόμενη συμπεριφορά της. [29]
-
- **Services:** Όταν υπάρχουν δεδομένα ή λογική τα οποία δεν σχετίζονται άμεσα με την προβολή (view) τους μέσω των components αλλά σκοπός είναι να διαμοιραστούν σε αυτά είναι απαραίτητες οι κλάσεις τύπου service. Οι κλάσεις αυτές σχετίζονται με τον @Injectable decorator. [29]
-
- **Dependency Injection:** Η λειτουργία αυτή συμβάλλει στην δημιουργία κλάσεων μεστών, περιεκτικών και αποτελεσματικών. [29] Τα Dependencies αποτελούν services ή αντικείμενα (objects) τα οποία χρειάζεται μια κλάση για να πραγματοποιήσει μια λειτουργία. Το Dependency Injection ή αλλιώς DI, είναι ένα design pattern δηλαδή μια αποδεδειγμένα καλή προγραμματιστική τακτική, στην οποία μια κλάση αιτείται τα dependencies απο εξωτερικούς πόρους και δεν τους δημιουργεί η ίδια. Το Angular DI framework παρέχει την δυνατότητα προσθήκης dependencies σε μια κλάση την στιγμή της δημιουργίας της. [31]
-
- **Directives:** Τα Directives είναι κλάσεις που προσδίδουν πρόσθετη συμπεριφορά στα HTML στοιχεία των Angular εφαρμογών και χωρίζονται στις παρακάτω κατηγορίες: [31]

- **Components:** Είναι ουσιαστικά Directives με ένα template. Αυτός ο τύπος directive είναι και ο πιο κοινός. [31]
- **Attribute Directives:** Τα directives αυτού του τύπου αλλάζουν την εμφάνιση ή την συμπεριφορά σε ένα HTML element, component ή ακόμη και σε άλλο directive. [31]
- **Structural Directives:** Είναι directives που αλλάζουν την δομή του DOM προσθέτοντας ή αφαιρώντας ολόκληρα DOM elements. [31]

Πλεονεκτήματα της Angular

Το Angular Framework έχει πολλά πλεονεκτήματα, τόσο απο επιχειρηματική (business) άποψη όσο και απο άποψη ανάπτυξης εφαρμογών (development). Ανήκει στην κατηγορία των framework που μπορούν να συνδυάσουν την επιχειρηματική λογική (business logic) με το UI, [33] καθώς και να δουλέψουν παραγωγικά με οποιουδήποτε τύπου back-end.

Η Angular προσφέρει την δυνατότητα ανάπτυξης του front-end τμήματος μια εφαρμογής με τρόπο στιβαρό, αποδοτικό και με χαμηλό κόστος. [33] Παρακάτω παρατίθενται και αναλύονται τα πλεονεκτήματα του **Angular Framework** τόσο απο **επιχειρηματική** όσο και απο **τεχνική άποψη**. [33]

Επιχειρηματικά οφέλη των λειτουργιών της Angular (Business Benefits Of Angular Features):

- **Αποτελεσματική Ανάπτυξη σε πολλές πλατφόρμες (Effective Cross-Platform Development):** Το Angular Framework παρέχει την δυνατότητα ομαλής λειτουργίας των εφαρμογών που παράγονται απο αυτό σε πολλά και διαφορετικά περιβάλλοντα.
Για παράδειγμα εκτός απο την παροχή τρόπων ανάπτυξης εφαρμογών με χαμηλό κόστος οι οποίες μπορούν να τρέχουν και σε κινητά, το framework παρέχει την δυνατότητα ανάπτυξης εφαρμογών αποκλειστικά για κινητά. [33]
- **Υψηλή ποιότητα της Εφαρμογής (High Quality of the Application):** Η Angular είναι πολύπλοκη πλατφόρμα που είναι δύσκολο να μάθει κάποιος, οπότε απαιτεί και τα κατάλληλα προσόντα από έναν προγραμματιστή. Πολλά δομικά στοιχεία της Angular

όπως είναι οι Injectors, τα Components, τα Directives, τα Pipes ή τα Services μπορεί να κάνουν την εκμάθησή της απο έναν προγραμματιστή λίγο δύσκολη. [33]

Απεναντίας το γεγονός ότι το Angular framework δίνει την δυνατότητα στον προγραμματιστή να δημιουργήσει οτιδήποτε σκεφτεί παρέχοντάς του ένα ολοκληρωμένο σύνολο ενσωματωμένων δυνατοτήτων, υπερνικά την οποιαδήποτε δυσκολία στην εκμάθησή του. [33]

Συνολικά η δομή που προσφέρει το Angular framework, προάγει τις καλές προγραμματιστικές πρακτικές και ακολούθως βοηθά στην παραγωγή προϊόντος υψηλής ποιότητας.

- **Βελτιωμένη ταχύτητα και απόδοση (Improved Speed and Performance):**

Η ποικιλομορφία των δυνατοτήτων της Angular όπως είναι για παράδειγμα τα instances, το συντακτικό των templates, το Angular CLI, ή οι routers, διευκολύνουν το έργο του προγραμματιστή και επιτρέπουν την γρήγορη φόρτωση της εφαρμογής. Το framework είναι συμβατό με οποιουδήποτε τύπου back-end και επιτυγχάνει την εμφάνιση των συλλεχθέντων δεδομένων στο UI με τρόπο αποτελεσματικό. [33]

Τεχνικά οφέλη των λειτουργιών της Angular (Technical Benefits Of Angular):

- **Γρήγορη διαδικασία ανάπτυξης (Faster Development Process):** Το Angular framework επιτρέπει σε έναν προγραμματιστή να δημιουργεί Angular εφαρμογές γρηγορότερα καθιστώντας τις ιδιαίτερα αποδοτικές χάρη στα τεχνικά πλεονεκτήματα που προσφέρει. [33]

- **Λεπτομερής Τεκμηρίωση (Detailed Documentation):** Οι προγραμματιστές της Angular φροντίζουν να κάνουν την Angular προσιτή και εύκολη στην εκμάθηση. Αυτοί παρέχουν προσεκτικά γραμμένη τεκμηρίωση και συμπεριλαμβάνουν σε αυτή εξαιρετικά παραδείγματα κώδικα για καλύτερη διευκρίνιση. Έτσι ο προγραμματιστής που χρησιμοποιεί την Angular μπορεί εύκολα να βρει λύση σε οποιοδήποτε πρόβλημα προκύπτει κατά την δημιουργία μιας εφαρμογής. [33]

- **Angular CLI:** Η ειδική διεπαφή γραμμής εντολών (Angular command-line) που προσφέρει η Angular, κάνει την δουλειά του προγραμματιστή ευκολότερη επειδή προσφέρει ένα σύνολο απο βοηθητικά εργαλεία [33] για την παραγωγή κώδικα για λειτουργίες όπως:

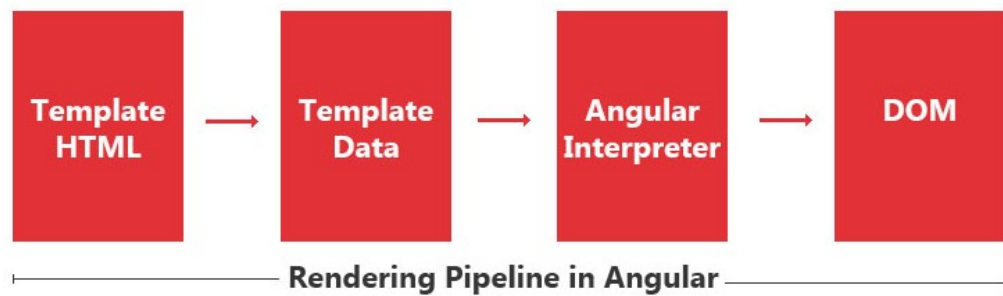
- Δημιουργία ενός Angular project απο το μηδέν. [35]

- Δημιουργία components, directives, services, καθώς και άλλων δομικών στοιχείων. [35]
- Έλεγχος του κώδικα. [35]
- Εκκίνηση της εφαρμογής. [35]
- Εκκίνηση ειδικών unit test. [35]

Εκτός απο τις ισχυρές ενσωματωμένες λειτουργίες που αναφέρονται παραπάνω, η Angular μπορεί να επεκταθεί μέσω τρίτων βιβλιοθηκών (third-party libraries) με στόχο την επίλυση ασυνήθιστων ή πολύπλοκων προβλημάτων. [33] Οι βιβλιοθήκες αυτές μπορούν να εγκατασταθούν στην εφαρμογή επίσης μέσω του Angular CLI.

- **Two-Way Data Binding:** Το two-way data binding το οποίο αναλύθηκε και προηγουμένως, είναι μια λειτουργία που γλιτώνει χρόνο και αυτοματοποιεί κάποιες διαδικασίες στο επίπεδο παραγωγής κώδικα. [33]
- **Differential Loading:** Οι νέες εκδόσεις της Angular επιτρέπουν την δημιουργία καναλιών δύο τύπων: Ένα σχετικό με τα εργαλεία που περιέχουν τον κώδικα και ένα σχετικό με τους πόρους που απαιτούνται για αποτελεσματικό προγραμματισμό. Για να είναι η εφαρμογή συμβατή με τους περισσότερους browsers, το ένα κανάλι χρησιμοποιείται για τους μοντέρνους browsers που υποστηρίζουν E2015+ και το άλλο για τους παλαιότερους που υποστηρίζουν την ES5 JS έκδοση. Με το differential loading οι browsers μπορούν να φορτώνουν λιγότερο κώδικα και polyfills (αρχεία σχετικά με την εφαρμογή) καθιστώντας την εφαρμογή περισσότερο παραγωγική. [33]
- **Υποστήριξη απο την Google (Google Support):** Η Google δημιούργησε την Angular για να την χρησιμοποιήσει εσωτερικά για τους δικούς της επίσημους ιστότοπους και να λύσει προβλήματα στα εσωτερικά συστήματά της. Η Angular παρέχει ενημερώσεις κάθε έξι μήνες προσφέροντας μικρές σταδιακές αλλαγές για κάθε νέα έκδοση για την άνετη και συνεχή εξέλιξη του framework. [33]
- **Μεγάλη Κοινότητα Προγραμματιστών (Large Developer Community):** Η Angular είναι ευρέως δημοφιλής μεταξύ των προγραμματιστών ανα τον κόσμο. Ήδη απο τις απαρχές της AngularJS (η παλαιότερη έκδοση της Angular), υπήρχε μια δυναμική κοινότητα γύρω απο το framework. Οι ειδικοί της Angular συνεισφέρουν συνεχώς στο framework, μοιράζονται την εμπειρία τους στον προγραμματισμό, συζητούν συγκεκριμένα θέματα και βρίσκουν τρόπους να επιλύουν τα προβλήματα απο κοινού. [33]

- **Ευανάγνωστος και εύκολα ελέγξιμος κώδικας (Readable and Testable Code):** Αυτό που κάνει τον κώδικα της Angular λογικό, συνεπή και εύκολα ακολουθήσιμο από έναν εξειδικευμένο προγραμματιστή front-end εφαρμογών, είναι τα δύο κυριότερα δομικά του στοιχεία:
 - **Modules:** Το γεγονός ότι η Angular είναι σπονδυλωτή σχετίζεται άμεσα με την δομή του framework που όπως προαναφέρθηκε περιλαμβάνει modules, components, directives, pipes και services. Η υποδιαίρεση της δομής του framework σε modules και components καθιστά την εφαρμογή άκρως κατάλληλη για στοχευμένους ελέγχους σε διάφορα τμήματά της (unit-testing). Κάθε τμήμα κώδικα ελέγχεται ξεχωριστά κατά την διάρκεια της διαδικασίας ανάπτυξης πράγμα που επιτρέπει σχολαστικό έλεγχο της ποιότητας. [33]
 - **Components:** Οι νέες εκδόσεις της Angular χαρακτηρίζονται από μια δομή βασισμένη στα components στην οποία όλα τα δομικά στοιχεία είναι ανεξάρτητα μεταξύ τους. Ενώ το πρότυπο MVC διαχωρίζει την δομή της εφαρμογής σε διαφορετικά επίπεδα, τα components περιλαμβάνουν τα χαρακτηριστικά αυτά σε επίπεδο κλάσης. Έτσι, όποιο component κι αν δημιουργεί ο προγραμματιστής, η συνολική δομή της εφαρμογής διατηρείται, γεγονός που συμβάλλει στην συνοχή του κώδικα. [33]
- **Ελαφρύτερες Web Εφαρμογές (More Lightweight Web Applications):** Για την γρηγορότερη φόρτωση των εφαρμογών, το Angular framework παρέχει τις εξής βελτιώσεις:
 - **Lazy-Load Modules:** Τα Modules είναι τα δομικά στοιχεία της εφαρμογής τα οποία διαιρούν τα κύρια components (business components). Οι μεγάλες εφαρμογές γραμμένες σε Angular μπορούν να χρησιμοποιούν **lazy-load modules** για να αναπαράγουν διάφορα components της εφαρμογής βασιζόμενα στο που είναι ο χρήστης στην εφαρμογή. Αυτή η λειτουργία συμβάλλει στην βελτίωση της απόδοσης πολλών εφαρμογών ελαχιστοποιώντας το μέγεθος της εφαρμογής κατά την αρχική φόρτωσή της, [33] καθώς η φόρτωση των Component γίνεται τμηματικά ανάλογα με τις ανάγκες του χρήστη.
 - **Ivy Renderer:** Ο renderer είναι ένας μηχανισμός που λαμβάνει εντολές γραμμένες από έναν προγραμματιστή και τις μεταφράζει για το DOM, μια διεπαφή διαδικτυακών σελίδων (web page interface) που συμβάλλει στον έλεγχο του περιεχομένου, της δομής και του τρόπου εμφάνισής τους. Ο Ivy renderer επιτρέπει την δημιουργία μικρότερων υποσυνόλων δεδομένων που απαιτούνται στην προαναφερόμενη διαδικασία, επιταχύνοντας έτσι την εφαρμογή συνολικά. [33]



Σχ.35: Angular Rendering [34]

- **Αποτελεσματικά πρότυπα για την επίλυση προβλημάτων (Efficient Problem-Solving Patterns):** Όπως προαναφέρθηκε η Angular προσφέρει ένα πανίσχυρο εργαλείο, το dependency injection καθώς και services για την επίλυση διαφόρων προβλημάτων που έχουν σχέση με το επίπεδο παραγωγής της εφαρμογής επιταχύνοντας την διαδικασία ανάπτυξης του λογισμικού. [33]
- **Dependency Injection:** Όπως αναφέρθηκε και παραπάνω, το dependency injection είναι ένα design pattern που καθιστά την εφαρμογή στην οποία εφαρμόζεται σπονδυλωτή και αποτελεσματική. Είναι ένας τρόπος δημιουργίας αντικειμένων που εξαρτώνται από άλλα αντικείμενα. Με το dependency injection η Angular καθιστά δυνατή την ανάθεση ορισμένων server-side services στα αντίστοιχα τμήματα της client-side εφαρμογής. [33]
- **Angular Services:** Τα components της Angular δεν είναι σωστό να επικοινωνούν άμεσα με το server side της εφαρμογής, αλλά πρέπει να επικεντρώνονται στην αναπαράσταση των δεδομένων αυτών και στην παροχή κατάλληλης πρόσβασης στα αντίστοιχα services. Τα services βοηθούν να συνδυαστεί η επιχειρηματική λογική (business logic) και το UI (διεπαφή χρήστη) τόσο όσο χρειάζεται ώστε ο κώδικας να είναι πιο καθαρός. Ένας προγραμματιστής μπορεί να εισάγει (δηλαδή να κάνει import) ένα service μόνο μια φορά στον κώδικα και έπειτα να το χρησιμοποιήσει οπουδήποτε το χρειαστεί ξανά. Ως εκ τούτου ένα ειδικά εξειδικευμένος προγραμματιστής, εργάζεται γρηγορότερα καθώς χρειάζεται να παράξει λιγότερο κώδικα. [33]
- **Material Design Library:** Το Angular Material για το οποίο γίνεται αναφορά στο μεθεπόμενο κεφάλαιο είναι μια βιβλιοθήκη που επιτρέπει την εφαρμογή ειδικών Material Design στοιχείων (elements). Το Material Design είναι ένα σύστημα σχεδιασμού φτιαγμένο από την Google που χρησιμοποιείται για την δημιουργία εξαιρετικά ευέλικτων και παραγωγικών διεπαφών χρήστη (user interfaces). Οι

προγραμματιστές επιλέγουν το Angular Material επειδή μόλις εξοικειωθούν με αυτό μπορούν εύκολα να το ενσωματώσουν σε οποιαδήποτε Angular εφαρμογή επιθυμούν. [33]

Κάποιες δημοφιλείς εφαρμογές που αναπτύχθηκαν με την Angular είναι οι εξής:

- Gmail
- YouTubeTV
- UDACITY
- Upwork
- Forbes
- PayPal
- Office
- The Weather Channel
- Εφαρμογές για το Netflix

Το client-side του συστήματος διαχείρισης παρουσιών έχει αναπτυχθεί με την βοήθεια του Angular Framework, αποκομίζοντας τα πλεονεκτήματα και τα οφέλη που προαναφέρθηκαν σε μεγάλο βαθμό. Απο την δημιουργία ειδικών modules με τα επιμέρους components μέχρι την αξιοποίηση των services, του data binding, του lazy loading, τον συγκερασμό εξειδικευμένων βιβλιοθηκών (third-party libraries) αλλά και πολλών άλλων λειτουργιών, ο τρόπος δόμησης της Angular έδωσε ένα μεγάλο εύρος επιλογών και λύσεων εναρμονίζοντας το business logic της εφαρμογής με τις αντίστοιχες λειτουργίες στο user interface. ***Η έκδοση της Angular που χρησιμοποιήθηκε είναι η 12.0.0.***

3.2.2. RxJS

Reactive Programming:

Το **Reactive Programming** είναι ένα είδος ασύγχρονου προγραμματισμού που σχετίζεται με τις ροές δεδομένων (data streams) και την διάδοση της αλλαγής (propagation of change) [36] που μπορεί να προκύψει για παράδειγμα, από ένα συμβάν (event).

Behavioral Design Patterns - Observer Pattern:

Τα Behavioral Design Patterns σχετίζονται με αλγορίθμους και την ανάθεση αρμοδιοτήτων μεταξύ των αντικειμένων. Ο Observer είναι ένα behavioral design pattern που βοηθά τον προγραμματιστή να ορίσει έναν μηχανισμό ο οποίος ενημερώνει πολλά αντικείμενα για τυχόν συμβάντα (events) που αφορούν αντικείμενα την κατάσταση των οποίων παρακολουθεί. [38] Για να ενημερωθούν τα αντίστοιχα αντικείμενα θα πρέπει να έχει εκδηλωθεί ένα ενδιαφέρον από αυτά για το συγκεκριμένο Observable το οποίο καθορίζει ο προγραμματιστής.

Iterator Pattern: Το Iterator Pattern είναι επίσης ένα behavioral pattern που επιτρέπει στον προγραμματιστή να διατρέχει τα στοιχεία μιας συλλογής (collection) χωρίς να χρειάζεται να εκθέτει την υποκείμενη αναπαράστασή της που μπορεί να αφορά λίστα, στοίβα, δένδρο ή διάφορες άλλες δομές. [38]

Functional Programming: Ο συναρτησιακός προγραμματισμός (functional programming) είναι ένα είδος προγραμματισμού που προσπαθεί να δεσμεύσει τα πάντα σε μια καθαρά μαθηματική προσέγγιση συναρτησιακού τύπου. [39]

Collection: Το collection είναι μια κλάση που χρησιμοποιείται για την αναπαράσταση ενός συνόλου δεδομένων του ίδιου τύπου ως μια ενιαία μονάδα. [40]

RxJS

Το **RxJS (Reactive Extensions for JavaScript)** είναι μια βιβλιοθήκη κατάλληλη για Reactive Programming που χρησιμοποιεί observables για να διευκολύνει την σύνταξη ασύγχρονου κώδικα ή κώδικα που βασίζεται σε callbacks. [36]

Παρέχει έναν βασικό τύπο το **Observable**, τύπους-δορυφόρους (Observer, Schedulers, Subject) καθώς επίσης και τελεστές (map, filter, reduce, every) που επιτρέπουν τον χειρισμό ασύγχρονων συμβάντων με την μορφή συλλογών δεδομένων (collections). [37]

Το **RxJS** συνδιάζει το **Observer pattern** με το **Iterator pattern** και το **functional programming** με **collections** για να καλύψει την ανάγκη για έναν ιδανικό τρόπο διαχείρισης των ακολουθιών απο συμβάντα (sequences of events). [37]

Τα βασικά στοιχεία του RxJS που συμβάλλουν στην ασύγχρονη διαχείριση συμβάντων (async event management), είναι τα εξής: [37]

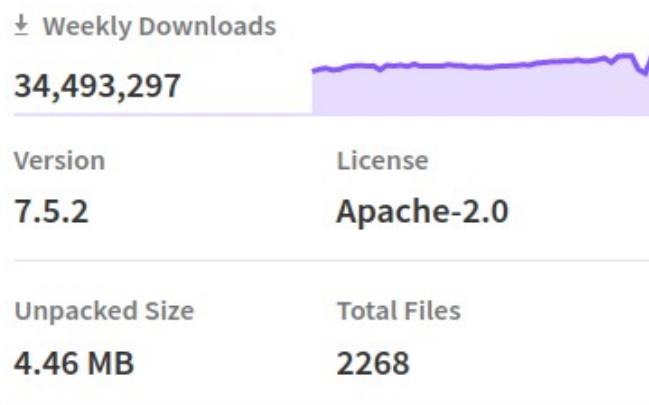
- **Observable:** Αναπαριστά την ιδέα ενός μηχανισμού που παράγει μια συλλογή (collection) απο τιμές ή συμβάντα (events).
- **Observer:** Είναι μια συλλογή απο callbacks που γνωρίζει πότε να αφουγκράζεται τις τιμές που εκπέμπονται από το Observable.
- **Subscription:** Αναπαριστά την εκτέλεση ενός Observable, και είναι χρήσιμο κυρίως για την διακοπή της διαδικασίας.
- **Operators:** Είναι καθαρές συναρτήσεις που επιτρέπουν ένα συγκεκριμένο ύφος functional programming για την διαχείριση των συλλογών (collections) παρέχοντας λειτουργίες όπως είναι οι map, filter, concat, reduce καθώς και αρκετές άλλες.
- **Subject:** Είναι ισοδύναμο με το EventEmitter (ειδική γεννήτρια συμβάντων) και είναι ο μόνος τρόπος πολλαπλής αποστολής μιας τιμής ή ενός συμβάντος (event) σε πολλαπλούς παρατηρητές (Observers).
- **Schedulers:** Είναι κεντρικοί αποστολείς για τον έλεγχο της παραλληλίας, επιτρέποντας τον συντονισμό του πότε και πόσο να γίνεται μια λειτουργία, όπως για παράδειγμα ένας υπολογισμός.

Το RxJS παρέχει βοηθητικές συναρτήσεις που χρησιμεύουν για την δημιουργία και διαχείριση των **Observables**. Ενδεικτικά μπορούν να χρησιμοποιηθούν για: [36]

- Την μετατροπή υπάρχοντος κώδικα για ασύγχρονες λειτουργίες σε Observables.
- Την επανάληψη των τιμών σε μια ροή δεδομένων (stream)
- Την αντιστοίχιση τιμών σε διαφορετικούς τύπους.
- Το φιλτράρισμα ροών (streams).
- Την σύνθεση πολλαπλών ροών (streams).

Πλεονεκτήματα της χρήσης του RxJS:

- **Το RxJS αναπτύσσεται ραγδαία:** Το RxJS είναι ένα ισχυρό εργαλείο που καθιστά το reactive programming περισσότερο ελκυστικό και προσιτό για τους προγραμματιστές, ενώ αναπτύσσεται και εξελίσσεται ραγδαία: [41]
 - Έχει περισσότερα απο 2.4 εκατομμύρια αποθετήρια στο GitHub.
 - Έχει περισσότερα απο 16 χιλιάδες σχετικά πακέτα στο NPM.
 - Ο αριθμός των εβδομαδιαίων downloads από το NPM έχει ξεπεράσει τα 34 εκατομμύρια.



Εικ.8. RxJS Downloads [42]

- **Ελαστικότητα στην χρήση:** Το RxJS μπορεί εύκολα να χρησιμοποιηθεί και με άλλες Javascript βιβλιοθήκες και frameworks όπως είναι η Angular, το ReactJS, το Vue.js, το Node.js και πολλά ακόμα. Το RxJS υποστηρίζεται τόσο από την JavaScript, όσο και από την Typescript. [41]

- **Είναι μεγάλη και χρήσιμη βιβλιοθήκη:** Το RxJS είναι μια τεράστια βιβλιοθήκη που όπως προαναφέρθηκε παρέχει εργαλεία για τον χειρισμό ασύγχρονων εργασιών (asynchronous tasks). [41]
- **Παρέχει API υψηλής ποιότητας:** Το RxJS παρέχει ένα εύχρηστο API που κρύβει από τον προγραμματιστή την πολυπλοκότητα των διεργασιών που καλείται να επιλύσει επιτρέποντάς του να εστιάσει σε σημεία που αφορούν το κομμάτι της λογικής ανάπτυξης του προϊόντος. [41]
- **Διευκολύνει την διαδικασία ανάπτυξης:** Η πληθώρα των εργαλείων που παρέχει πολλά από τα οποία προαναφέρθηκαν, σε συνδυασμό με το reactive programming διευκολύνουν την διαδικασία ανάπτυξης του προϊόντος αφού δίνει στον προγραμματιστή αρκετές εναλλακτικές για λύσεις. [41]
- **Παρέχει καλή βελτιστοποίηση και προστασία από διαρροές μνήμης (memory leaks protection):** Το RxJS έχει ένα στέρεο υπόβαθρο που εξασφαλίζει στους προγραμματιστές υψηλή ποιότητα βελτιστοποίησης. Επίσης μεγάλος αριθμός προγραμματιστών έχει συμβάλει στην δοκιμή και βελτιστοποίηση αυτής της βιβλιοθήκης με στόχο την παροχή εργαλείων για την καλύτερη αντιμετώπιση των διαρροών μνήμης. [41]
- **Έχει μικρό και περιεκτικό αποτύπωμα:** Το RxJS παρέχει βελτιστοποίηση υψηλής ποιότητας και έχει σπονδυλωτή αρχιτεκτονική (modular architecture). Αυτά τα χαρακτηριστικά του το καθιστούν μικρό και ο όγκος του κατά την διαδικασία παραγωγής αφορά μόνο εκείνα τα τμήματα που πρόκειται να χρησιμοποιηθούν. [41]
- **Είναι επεκτάσιμο:** Το RxJS παρέχει όπως προαναφέρθηκε ένα ευρύ σύνολο τυποποιημένων τελεστών, ωστόσο κάποιες φορές για την ικανοποίηση εξειδικευμένων αναγκών είναι απαραίτητη η δημιουργία νέων και χάρη στην επεκτασιμότητα που προσφέρει αυτό είναι εφικτό. [41]

- **Είναι ανεξάρτητο από άλλες βιβλιοθήκες:** Το RxJS δεν εξαρτάται από άλλες βιβλιοθήκες αλλά διαθέτει μια αυτάρκη βιβλιοθήκη. Για αυτό τον λόγο το μέγεθος του RxJS project είναι βέλτιστο και εξαρτάται από τις βιβλιοθήκες που χρησιμοποιεί. [41]

Σύμφωνα με το NPM, μέχρι στιγμής, το RxJS έχει μόνο μια εξάρτηση, την tslib. Αυτή η μοναδική εξάρτηση χρησιμοποιείται για την παροχή υποστήριξης για την γλώσσα Typescript. [41]

Το γεγονός ότι το RxJS είναι ανεξάρτητο από άλλες βιβλιοθήκες, κάνει πιο εύκολο το να υιοθετηθεί ως εξάρτηση για άλλα πακέτα όπως API SDKs, βοηθητικά πακέτα, server-side προγράμματα, webhook υλοποιήσεις, front-end event managers καθώς και για αρκετά άλλα. [41]

- **Μεγάλη και ενεργή κοινότητα:** Η πολυμορφική και μεγάλη κοινότητα του RxJS είναι ένα από τα μεγαλύτερα πλεονεκτήματα του και το καθιστά ακόμα πιο ισχυρό και δημοφιλές. Οι συμμετέχοντες σε αυτή την κοινότητα βοηθούν ο ένας τον άλλον στην επίλυση προβλημάτων και την απάντηση σε ερωτήσεις στο StackOverflow και στο Gitter. [41]

- **Επαρκής τεκμηρίωση (documentation):** Το reactive programming σε συνδυασμό με το RxJS δεν είναι και πολύ απλό στην εκμάθηση, αλλά παρέχει εκτενή τεκμηρίωση για την αντιμετώπιση αυτής της δυσκολίας. Ο επίσημος δικτυακός τόπος του RxJS παρέχει πόρους καθοδηγούμενους από την κοινότητα (community-driven resources) οι οποίοι παρέχουν εξειδικευμένη επεξήγηση των εννοιών, των βέλτιστων πρακτικών, των προτύπων και των παραδειγμάτων και βοηθούν τους χρήστες να ενισχύουν τις θεωρητικές γνώσεις αλλά και τις δεξιότητές τους. [41]

- **Τακτικές ενημερώσεις (Regular Updates):** Ένα βασικό πλεονέκτημα του RxJS είναι επίσης ότι συντηρείται σωστά. Παρέχει τακτικές ενημερώσεις στο GitHub, πράγμα που το καθιστά αξιόπιστο. Οι τακτικές και συνεπείς ενημερώσεις για την συνολική βελτίωση του RxJS το καθιστούν πιο ισχυρό, αξιόπιστο και βολικό. [41]

Στο client-side του συστήματος διαχείρισης παρουσιών το η βιβλιοθήκη RxJS ενσωματώθηκε στην Angular εφαρμογή και συνέβαλε στην δημιουργία σημαντικών μηχανισμών ιδανικών για την διεκπεραίωση ασύγχρονων λειτουργιών όπως για παράδειγμα είναι η επικοινωνία των επιμέρους components και dialogs με τα services, η ενημέρωση των δυναμικών πινάκων που

προβάλλονται στο UI, η πρόσβαση στις παραμέτρους των url, η διαχείριση των σφαλμάτων (error handling), καθώς και πολλές άλλες λειτουργίες.

Η έκδοση του RxJS που χρησιμοποιήθηκε στην εφαρμογή ήταν η 6.6.0.

3.2.3. Angular Material

Το Angular Material είναι μια βιβλιοθήκη που βοηθά το σχεδιασμό του user interface της εφαρμογής στην οποία συγκαταλέγεται, με δομημένο τρόπο. [43]

Το Angular Material αναπτύχθηκε από την Google το 2014 ως Angular Material JS και ήταν αρχικά σχεδιασμένο για AngularJS προγραμματιστές. Εκείνη την εποχή συμπεριλήφθηκε στο AngularJS ώστε να κάνει τις παραγόμενες από αυτό εφαρμογές ελκυστικές και με ταχύτερες επιδόσεις. [43]

Στην συνέχεια όμως η Google έγραψε τον κώδικα για το Angular Material JS ξανά από την αρχή και αφαίρεσε το ακρωνύμιο JS που έφερε μέχρι τότε μετονομάζοντάς το σε Angular Material. Τα στοιχεία που παρέχει το Angular Material για την ανάπτυξη του UI των εφαρμογών είναι γνωστά ως Angular Material Components ή Angular Materials. [43]

Τα δομικά στοιχεία που αποτελούν το Angular Material (δηλαδή τα components του), συμβάλλουν στην κατασκευή ελκυστικών, πρακτικών και λειτουργικών ιστοσελίδων και εφαρμογών ιστού (web pages). Γενικά με το Angular Material επιτυγχάνεται η δημιουργία ενός ευέλικτου και ταχύτερου ιστότοπου (website). [43]

Το Angular Material είναι ελκυστικό προς τους χρήστες και εναρμονίζεται με όλο το εύρος των elements και components μιας Angular εφαρμογής. Παράλληλα συμβάλλει στον σχεδιασμό εφαρμογών με μοναδική μορφή και στυλ. [43] Τα components του Angular Material,

καθιστούν τον ιστότοπο ή την εφαρμογή στην οποία περιλαμβάνονται σχεδιαστικά ευέλικτα καθώς συνδυάζουν τις κλασικές αρχές του επιτυχημένου σχεδιασμού με την τεχνολογία. [43]

Χαρακτηριστικά του Angular Material:

- Έχει responsive σχεδιασμό. [43]
- Έχει πρότυπο (standard) CSS. [43]
- Τα UI Components του Angular Material συμβαδίζουν με τις αρχές του Angular Material Design. [43]
- Διαθέτει εξειδικευμένα χαρακτηριστικά για την ανάπτυξη επαγγελματικών διεπαφών χρήστη. [43]
- Μπορεί να τρέχει σε πολλές πλατφόρμες (cross-platform) και browsers και χρησιμοποιείται για την δημιουργία web components. [43]

Πλεονεκτήματα του Angular Material:

Υψηλή Ποιότητα:

- Το Angular Material παρέχει διεθνοποιημένα και προσβάσιμα components για όλους. Τα Angular Material Components είναι καλά δοκιμασμένα πράγμα που τα καθιστά αποδοτικά και αξιόπιστα. [44]
- Το Angular Material παρέχει απλά APIs με συνεπή συμπεριφορά σε όλες τις πλατφόρμες. [44]

Ευελιξία:

- Παρέχει εργαλεία που βοηθούν τους προγραμματιστές να δημιουργήσουν τα προσαρμοσμένα (custom) components τους με κοινά πρότυπα (common patterns). [44]
- Είναι προσαρμόσιμο (customizable) εντός συγκεκριμένων ορίων. [44]

Εύκολη συμβατότητα με την Angular (Frictionless):

- Το Angular Material κατασκευάστηκε απο την ομάδα της Angular για να μπορεί να ενσωματωθεί τέλεια με αυτή. [44]
- Το Angular Material μπορεί να χρησιμοποιηθεί για την σχεδίαση μιας εφαρμογής απο το μηδέν ή μπορεί να προστεθεί σε μια ήδη υπάρχουσα εφαρμογή. [44]

Επεκτασιμότητα:

- Είναι εύκολη η προσθήκη νέων CSS προσαρμοσμένων στις εκάστοτε ανάγκες που μπορούν να αντικαταστήσουν τα υπάρχοντα CSS του Angular Material. [43]
- Υποστηρίζει σκιές και χρώματα. [43]
- Τα χρώματα και οι αποχρώσεις στο Angular Material είναι ομοιόμορφα. [43]
- Η χρήση του Angular Material είναι εντελώς δωρεάν. [43]

Responsive σχεδιασμός:

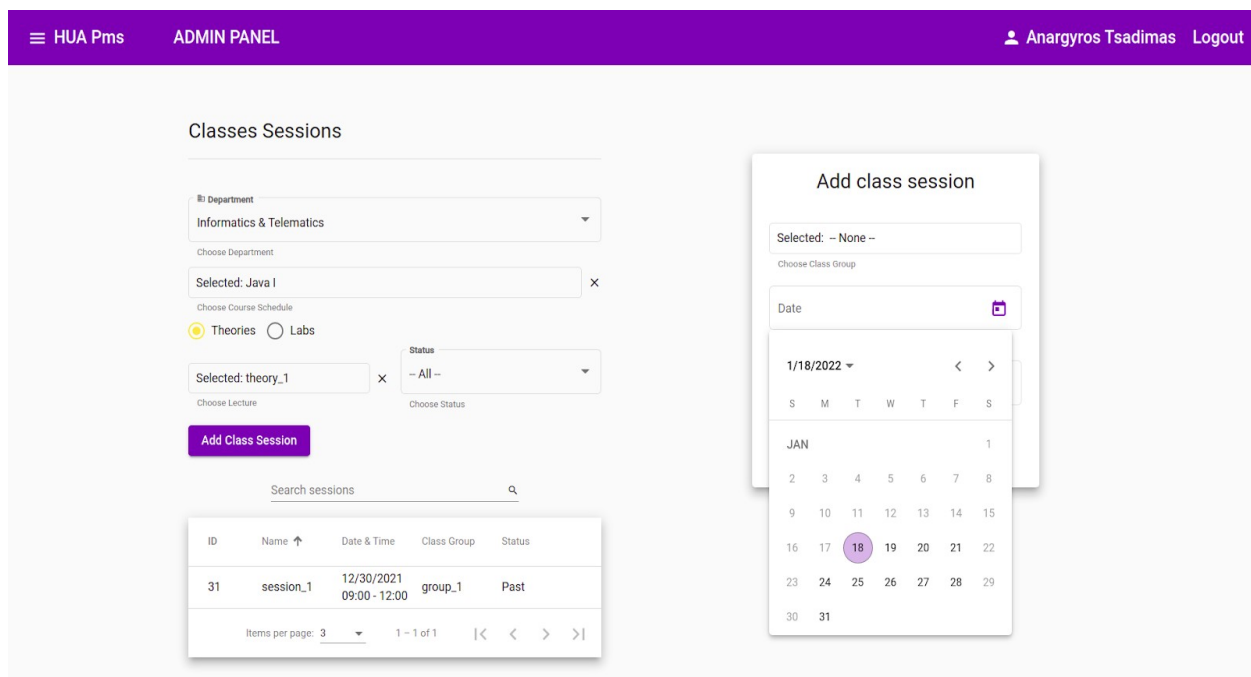
- Το Angular Material έχει responsive σχεδιασμό με αποτέλεσμα τα user interfaces που είναι υλοποιημένα με αυτό, να μπορούν να προσαρμοστούν σε οποιοδήποτε μέγεθος. [43]
- Οι ιστότοποι ή οι εφαρμογές που είναι υλοποιημένες με Angular Material είναι συμβατές με Android, iPhone, tablets και laptops. [43]

Η βιβλιοθήκη Angular Material χρησιμοποιήθηκε στο client-side του συστήματος διαχείρισης παρουσιών σε συνδυασμό με το Angular Framework καθιστώντας την σχεδίαση και υλοποίηση του user interface πρακτική και φιλική προς τον χρήστη.

Κατα την υλοποίηση του client-side της εφαρμογής, χρησιμοποιήθηκαν πολλά απο τα Components του Angular Material, άλλοτε μεμονωμένα και άλλοτε συνδυαστικά με σκοπό να καλύψουν πιο εξειδικευμένες ανάγκες ανάλογα με τις εκάστοτε απαιτήσεις. Σε αρκετές περιπτώσεις μάλιστα κάποια Angular Material Components παραμετροποιήθηκαν ώστε να εξυπηρετήσουν τις ανάγκες της εφαρμογής κατα τον βέλτιστο τρόπο.

Παρακάτω παρουσιάζονται κάποιοι σύνθετοι συνδυασμοί components που χρησιμοποιήθηκαν για να καλύψουν απαιτητικές ανάγκες στο επίπεδο του UI. Τα **Angular Material Components** εμφανίζονται με **bold**:

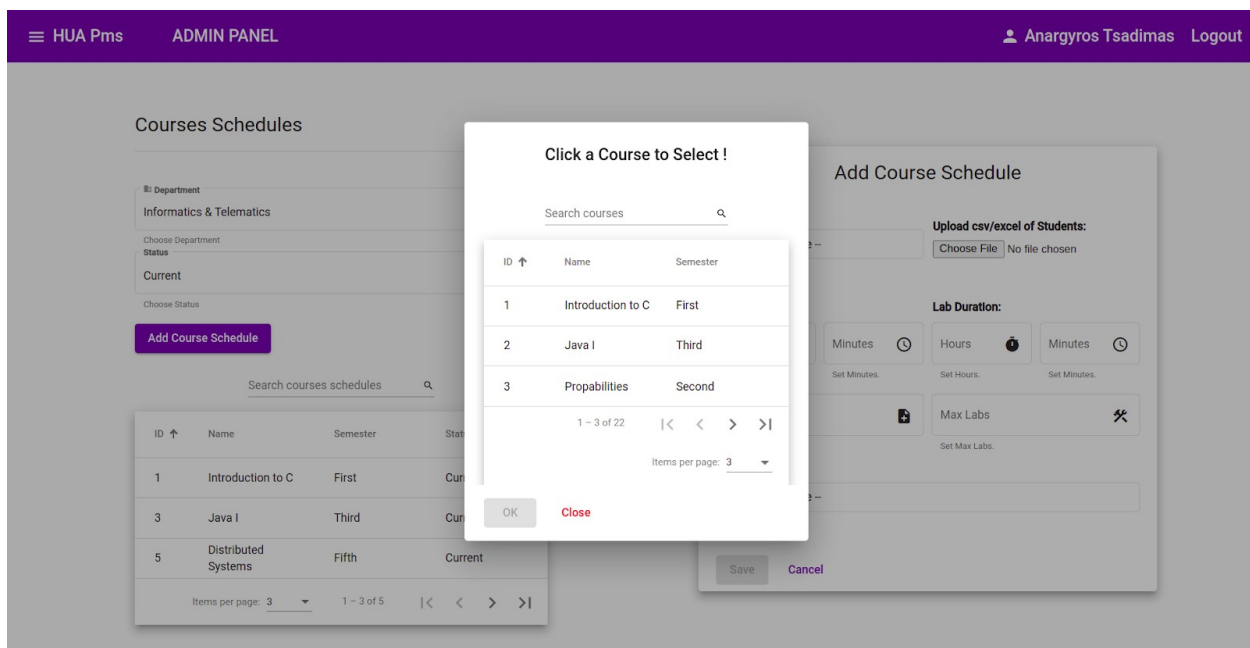
- Στην φόρμα προσθήκης μιας συνεδρίας διάλεξης χρησιμοποιείται **card** και μέσα σε αυτό **buttons** και **datepicker**.



Εικ.9. Classes Sessions Material Features - Datepicker

Στην παραπάνω εικόνα φαίνεται ο **datepicker** για την επιλογή της ημερομηνίας που θα διεξαχθεί η συνεδρία διάλεξης. Ο **datepicker** είναι παραμετροποιημένος ώστε να αποκλείονται τα σαββατοκύριακα και οι πεπερασμένες ημερομηνίες καθώς τότε δεν γίνεται να προγραμματιστεί μια συνεδρία διάλεξης. Η στέγαση της φόρμας δημιουργίας συνεδρίας διάλεξης στο **card component** με έναν **datepicker** εσωτερικά κατάλληλα παραμετροποιημένο, κάνει το **user interface** ευχάριστο και εύχρηστο για τον χρήστη.

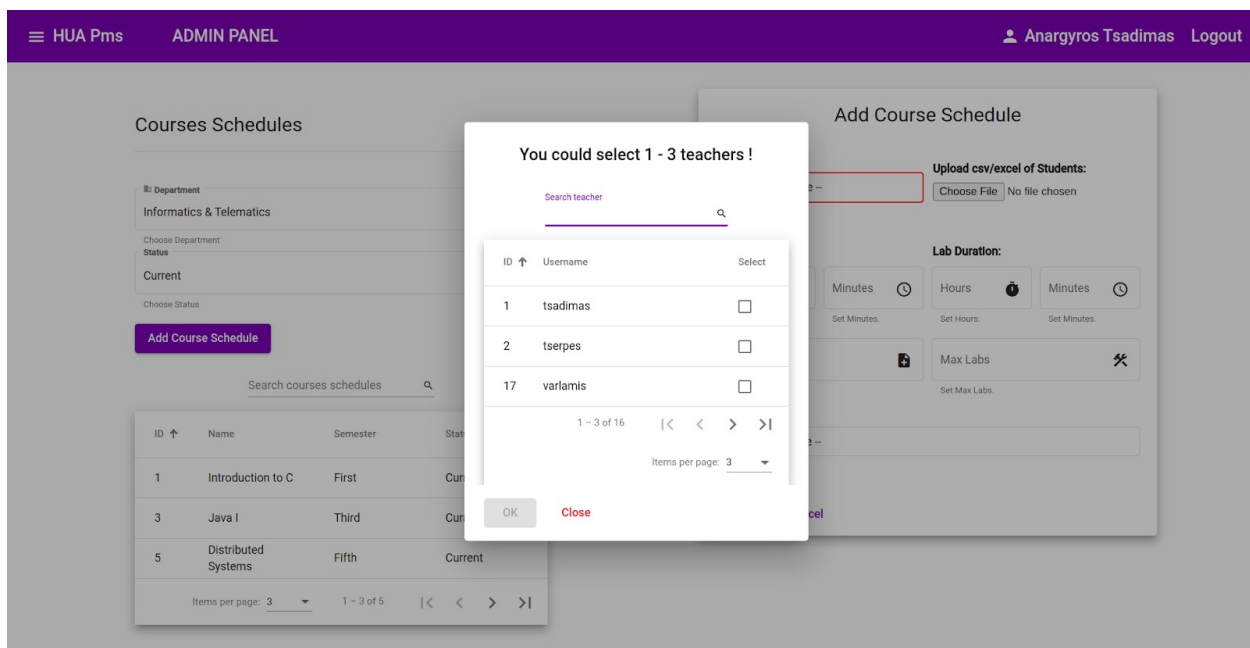
- Μέσα απο την φόρμα προσθήκης προγράμματος μαθήματος που περικλείεται σε ένα **card**, ανοίγει μέσω ενός **material button** ένα ειδικό **dialog** (pop-up) που μέσα περιέχει ένα **table** με τα μαθήματα καθώς και **material input** για την αναζήτησή τους.



Εικ.10. Courses Schedules Material Features - Επιλογή Μαθήματος

Ο συνδυασμός των παραπάνω Angular Material Components που συνέβαλε στην δημιουργία της παραπάνω κατασκευής ικανοποιεί την ανάγκη επιλογής ενός συγκεκριμένου μαθήματος ανάμεσα σε πολλά άλλα. Επειδή ακριβώς τα μαθήματα μπορεί να είναι πολλά ο παραπάνω επιλογέας καλύπτει την ανάγκη της αποτελεσματικής και φιλικής προς τον χρήστη πλοήγησης και τελικά επιλογής ή όχι κάποιου μαθήματος.

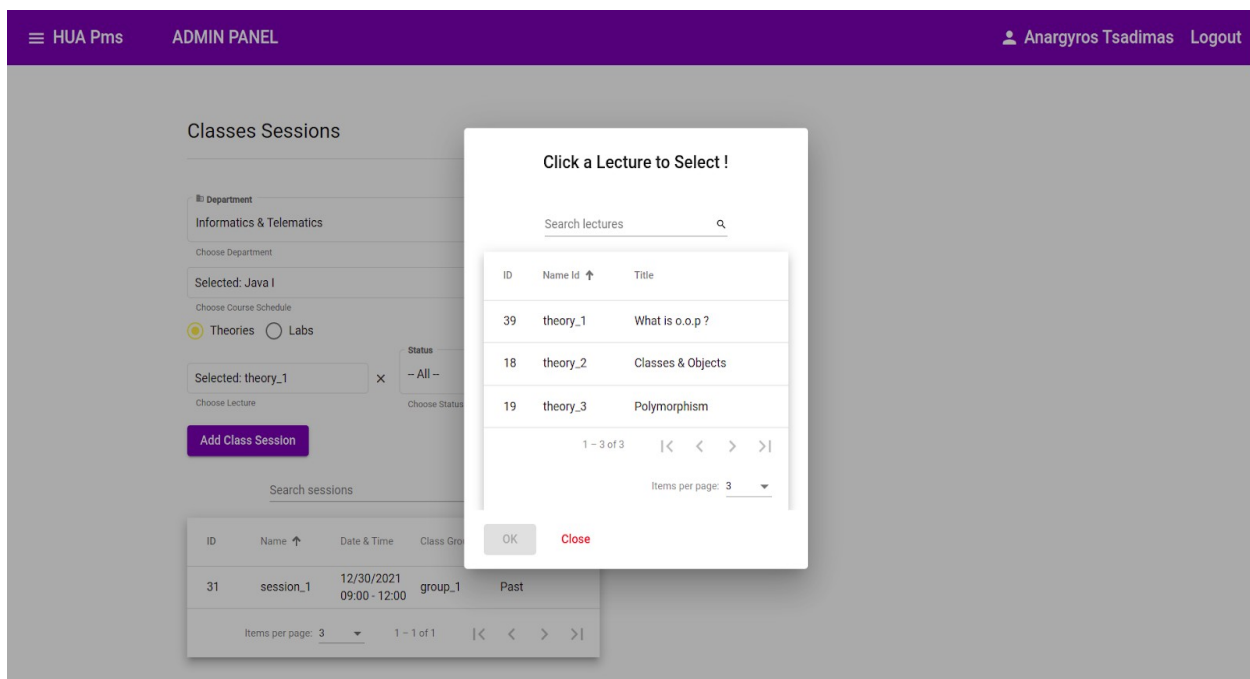
- Μέσα απο την φόρμα προσθήκης προγράμματος μαθήματος που περιέχεται μέσα σε ένα **card**, ανοίγει μέσω ενός **material button** ένα **dialog** που περιέχει:
 - Ενα **table** σε κάθε γραμμή του οποίου υπάρχει ένα **checkbox** για την επιλογή του.
 - Ενα **material input** για την αναζήτηση των καθηγητών.



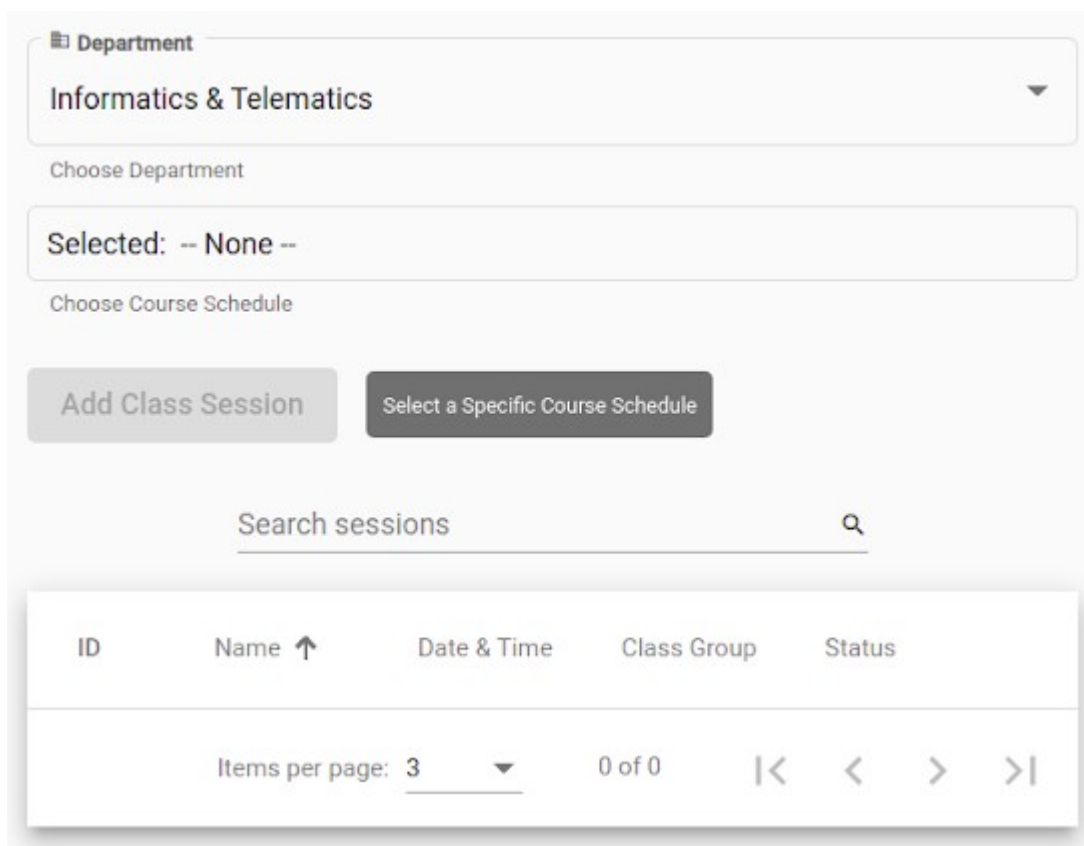
Εικ.11. Courses Schedules Material Features - Επιλογή καθηγητών

Ο συνδυασμός των παραπάνω Angular Material Components συνέβαλε στην δημιουργία του επιλογέα που φαίνεται στην εικόνα. Με βάση το γεγονός ότι οι καθηγητές είναι πολυάριθμοι, ο παραπάνω ειδικός επιλογέας καλύπτει την ανάγκη για την αποτελεσματική πλοήγηση του χρήστη. Παράλληλα με βάση τις απαιτήσεις του συστήματος ο χρήστης και συγκεκριμένα ο διαχειριστής του συστήματος, μπορεί να επιλέξει απο έναν έως τρεις καθηγητές. Ο παραπάνω ειδικός επιλογέας καλύπτει επίσης την ανάγκη της πολλαπλής επιλογής καθηγητών.

- Στην λίστα επιλογής συνεδρίας διάλεξης που φαίνεται στην παρακάτω εικόνα, χρησιμοποιούνται επίσης πολλά Components του Angular Material όπως:
 - **Button**
 - **Select**
 - **Table**
 - **Material Input**
 - **Radio Button**
 - **Dialog**
 - **Tooltip**



Εικ.12. Classes Sessions Material Features - Επιλογή Διάλεξης



Εικ.13. Classes Sessions Material Features - Λίστα

Στις παραπάνω εικόνες φαίνονται τα Angular Material Components που κατονομάζονται παραπάνω και καλύπτουν με τρόπο αποτελεσματικό, απλό και φιλικό προς τον χρήστη, την ανάγκη της στοχευμένης αναζήτησης συνεδριών διαλέξεων μέσα απο μεγάλο όγκο δεδομένων και παραμέτρων.

Ενδεικτικά για την επιλογή του πανεπιστημιακού τμήματος ή της κατάστασης της συνεδρίας διάλεξης χρησιμοποιείται **selector**, ενώ για την επιλογή προγράμματος μαθήματος και διάλεξης συνδυάζονται τα **button**, **dialog**, **table**, **input** που συνιστούν τον ειδικό επιλογέα που φαίνεται παραπάνω. Παράλληλα για την επιλογή του τύπου της συνεδρίας διάλεξης (θεωρητικού ή εργαστηριακού), χρησιμοποιείται **radio button**.

Τέλος για την προβολή των συνεδριών διαλέξεων χρησιμοποιείται **table**, ενώ για την αναζήτηση των υπαρχόντων στο **table** συνεδριών διαλέξεων χρησιμοποιείται κατάλληλο **input** αναζήτησης.

Αξίζει να αναφερθεί ότι τα κουμπιά που ανοίγουν αυτούς τους ειδικά κατασκευασμένους επιλογείς έχουν τροποποιηθεί ώστε να προβάλλεται στο εσωτερικό τους το όνομα της σχετικής οντότητας που επιλέχθηκε. Επιπροσθέτως δίνεται η δυνατότητα εκκαθάρισης της επιλεγμένης οντότητας μέσω ειδικού κουμπιού. Μάλιστα σε κάποια άλλα σημεία της εφαρμογής (όχι στα σημεία που απεικονίζονται παραπάνω) που χρησιμοποιούνται παρόμοιου τύπου κουμπιά, έχει δημιουργηθεί προσαρμοσμένος μηχανισμός επικύρωσης (custom validation).

Ενα ακόμη πράγμα για το οποίο αξίζει να γίνει αναφορά είναι το **προσαρμοσμένο angular material dialog component για την ενημέρωση του χρήστη με ειδικό μήνυμα** σε περίπτωση που μια διαδικασία αποτύχει ή επιτύχει. Το component αυτό, είναι παραμετροποιημένο ως προς τα CSS styles αλλά και τις διαστάσεις του, όπως και κάθε angular material dialog της εφαρμογής απλώς σε μεγαλύτερο βαθμό. Αυτού του είδους το **dialog** φαίνεται στις εικόνες παρακάτω.

Department

Informatics & Telematics

Choose Department

Selected: Java I

Choose Course Schedule

☐ Theories ☒ Labs

Add Class Group

Search groups

ID	Name ↑	Time	Room	Status
21	group_4	18:00 - 20:00	2.2	Closed

Items per page: 3 4 – 4 of 4 |< < > >|

Class Group added

Close

Εικ.14. Material Features - Success Dialog

Classes Sessions

Department

Informatics & Telematics

Choose Department

Selected: Java I

Choose Course Schedule

☒ Theories
 ☐ Labs

Selected: theory_1

Choose Lecture

Status

- All -

Choose Status

Add Class Session

Search sessions

ID	Name ↑	Date & Time	Class Group	Status
31	session_1	12/30/2021 09:00 - 12:00	group_1	Past

Items per page: 3

1 - 1 of 1

<<

<

>

>>

Add class session

Selected: group_1

Choose Class Group

Date

1/18/2022

Set date

Timespan

09:00 - 12:00

No.

1

Set number.

Save

Cancel

You cannot create a class session using a past date and time

Close

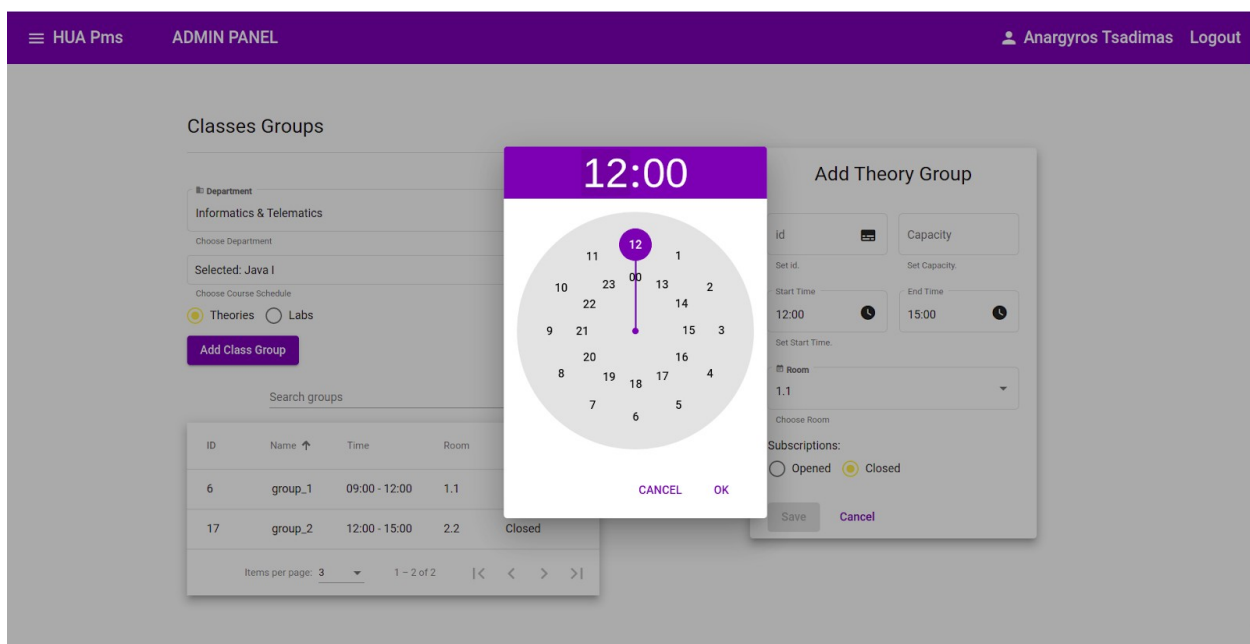
Εικ.15. Material Features - Failure Dialog

Όπως είναι προφανές, η βιβλιοθήκη Angular Material προσαρμοσμένη στην Angular εφαρμογή, προσέδωσε στο user interface του συστήματος διαχείρισης παρουσιών καλαισθησία και ευχρηστία. *Η έκδοση του Angular Material που χρησιμοποιήθηκε στην εφαρμογή είναι η 12.2.5.*

3.2.4. Timepicker

Για την κάλυψη των αναγκών της εφαρμογής σχετικά με τον καθορισμό της ώρας έναρξης της παρακολούθησης συνεδριών διαλέξεων που αφορά κάθε ομάδα φοιτητών χρησιμοποιήθηκε η βιβλιοθήκη **ngx-mat-timepicker**, καθώς η βιβλιοθήκη του Angular Material δεν παρέχει ακόμη component που να καλύπτει αυτή την ανάγκη.

Ο **ngx-mat-timepicker** είναι ένας απλός επιλογέας ώρας που συμβατός με το Angular Material. Συγκεκριμένα, παρέχει ένα dialog (pop-up) για την επιλογή ώρας το οποίο είναι αρκετά καλαίσθητο και εύχρηστο προσαρμοσμένο στο στυλ του Material. Αυτό το dialog ανοίγει μέσω ενός material input. [45] Ο **ngx-mat-timepicker** ταιριάζει αυτόματα στο επιλεγμένο θέμα (theme) του Material, όποιο κι αν είναι αυτό. [45] Τέλος επιτρέπει κάποιες παραμετροποιήσεις για παράδειγμα ως προς το χρώμα ή την μορφοποίηση (format) της ώρας (αν είναι 24ωρο ή 12ωρο). Ο timepicker που βασίστηκε στην βιβλιοθήκη **ngx-mat-timepicker** φαίνεται στην παρακάτω εικόνα.



Εικ.16. Timepicker

Η έκδοση του ngx-mat-timepicker που χρησιμοποιήθηκε στην Angular εφαρμογή του client-side του συστήματος διαχείρισης παρουσιών είναι η 12.1.0.

3.2.5. Angular Flex-Layout

CSS Flexible Box Layout

Το Css Flexible Box Layout ένα μοντέλο φτιαγμένο για την γλώσσα CSS. Συγκεκριμένα, ορίζει ένα μοντέλο βασισμένο σε CSS κουτιά (css box model), βελτιστοποιημένο για τον σχεδιασμό διεπαφών χρήστη (user interfaces) και την διάταξη των στοιχείων του UI σε μια διάσταση.

Στο μοντέλο του Css Flexible Box Layout, το σύνολο των παιδιών (children) που βρίσκονται ομαδοποιημένα σε ένα συγκεκριμένο πεδίο (πιο ορθά container), μπορούν να τοποθετηθούν προς οποιαδήποτε κατεύθυνση και να αλλάξουν δυναμικά τα μεγέθη τους είτε μεγαλώνοντας για να γεμίσουν τον αχρησιμοποίητο χώρο, είτε μικραίνοντας για να προσαρμοστούν στο μέγεθος του γονέα (parent). Τέλος, τόσο η οριζόντια όσο και η κατακόρυφη ευθυγράμμιση των παιδιών (children) μπορεί εύκολα να προσαρμοστεί, ανάλογα με τις εκάστοτε απαιτήσεις. [46]

Media Query

Το ερώτημα πολυμέσων (media query) είναι μια κατασκευή βασισμένη σε HTML/CSS που επιτρέπει στο περιεχόμενο μιας ιστοσελίδας να προσαρμόζεται στον τύπο του μέσου στο οποίο αποδίδεται η σελίδα, όπως η οθόνη ενός υπολογιστή, ενός κινητού τηλεφώνου, ή ενός tablet. Θεωρείται βασική τεχνολογία για την υλοποίηση responsive ιστοσελίδων (responsive web design) και προτάθηκε ως πρότυπο υλοποίησης μαζί με άλλες λειτουργίες του CSS3 τον Ιούνιο του 2012. [47]

Angular Flex Layout

Το Angular Flex Layout είναι μια αυτόνομη βιβλιοθήκη ανοικτού κώδικα που αναπτύχθηκε από την ομάδα της Angular για τον σχεδιασμό εξελιγμένων διατάξεων (layout) για τα user interfaces εφαρμογών. [49] Βασίζεται στην γλώσσα Typescript και μέσω της HTML καθορίζει το πως θα παραμετροποιηθούν τα στοιχεία που απεικονίζονται στην διεπαφή χρήστη [49] ενώ παρέχει ένα εξελιγμένο API το οποίο χρησιμοποιεί Flexbox CSS και mediaQuery. [48]

Το Angular Flex Layout προσφέρει στους Angular προγραμματιστές λειτουργίες για την διάταξη των στοιχείων της διεπαφής χρήστη βασισμένες σε components. Αυτό γίνεται κυρίως μέσω ενός προσαρμοσμένου API, τα observables του mediaQuery και τα εισηγμένα στυλιστικά στοιχεία του DOM flexbox-2016 CSS. [48] Στο απλό CSS flexbox οι responsive διατάξεις (responsive layouts) είναι φτιαγμένες χρησιμοποιώντας πολύπλοκο κώδικα CSS και media queries. [49]

Ο μηχανισμός του Flex Layout αυτοματοποιεί έξυπνα την διαδικασία εφαρμογής κατάλληλων δομών CSS Flexbox βασιζόμενο στην βέλτιστη ιεραρχία προβολής τους στον browser. Αυτή η αυτοματοποίηση ελαχιστοποιεί την πολυπλοκότητα και τις πιθανές δυσκολίες που συναντώνται σε μια παραδοσιακή, χειροκίνητη CSS εφαρμογή. [48]

Πιο συγκεκριμένα η χρήση του Angular Flex Layout επιτρέπει στον προγραμματιστή να δημιουργεί εύκολα σελίδες βασισμένες στην διάταξη του FlexBox, παρέχοντάς του ειδικά directives (μηχανισμούς καθοδήγησης του DOM) που μπορεί να χρησιμοποιήσει ανάλογα με τις εκάστοτε ανάγκες. Έτσι ο προγραμματιστής δεν χρειάζεται να γράψει ξεχωριστό κώδικα σε CSS για τις λειτουργίες που ικανοποιούνται από το Angular Flex Layout. [49]

Η πραγματική δύναμη του Flex Layout είναι ότι αποτελεί έναν responsive μηχανισμό. [48] Το Responsive API του, επιτρέπει στους προγραμματιστές να καθορίζουν εύκολα τις διάφορες παραμέτρους που αφορούν την προβολή των διαφόρων στοιχείων στο user interface κάνοντάς το περισσότερο ευέλικτο.

Πλεονεκτήματα του Angular Flex Layout έναντι του Angular Material Layout

Στην Angular εφαρμογή του client-side του συστήματος διαχείρισης παρουσιών για την δημιουργία μιας ευέλικτης διεπαφής χρήστη (responsive user interface), χρησιμοποιήθηκε η βιβλιοθήκη **Angular Flex Layout** και όχι το **Angular Material Layout**.

Αυτό συνέβη γιατί το Angular Material Layout απαιτεί περισσότερο και πολύπλοκο κώδικα σε CSS, εν αντιθέσει με το Angular Flex Layout που:

- Αυτοματοποιεί την διαδικασία παραμετροποίησης του user interface.
- Δεν απαιτεί την συγγραφή καθόλου κώδικα CSS, κάνοντας τον συνολικό κώδικα CSS της εφαρμογής πιο μεστό και καθαρό.
- Επιτρέπει τον χειρισμό των CSS απο ένα υψηλότερο επίπεδο μέσω ειδικών directives που παρέχει.
- Έχει αναπτυχθεί επίσης απο την ομάδα της Angular.

Αλλα πλεονεκτήματα για τα οποία προτιμήθηκε το Angular Flex Layout είναι τα εξής:

- Ο κώδικας που αποτελεί το Angular Flex Layout διευκολύνει περισσότερο την συντήρηση και την αποσφαλμάτωσή του. [50]
- Βασίζεται εξ' ολοκλήρου σε Typescript. [51]
- Χρησιμοποιεί HTML για την διάταξη των στοιχείων που εμφανίζονται στο UI. [51]
- Το Angular Flex Layout είναι ανεξάρτητη βιβλιοθήκη και δεν εξαρτάται απο το Angular Material. [51]
- Παρέχει ένα responsive API μέσω του οποίου μπορούν να παραμετροποιηθούν οι διαστάσεις, τα μεγέθη και πολλά άλλα χαρακτηριστικά των πεδίων στα οποία εφαρμόζεται προσδίδοντάς τους προσαρμοστικότητα και ευελιξία για κάθε τύπο συσκευής. [51]
- Δεν απαιτεί εξωτερικό CSS. [51]
- Περιλαμβάνει CSS Grid. [51]

Η έκδοση του Angular Flex Layout που χρησιμοποιήθηκε στην εφαρμογή είναι η 12.0.0.

3.2.6. Moment.js

Το Moment.js είναι ένα αυτόνομο framework ανοικτού κώδικα υλοποιημένο στην γλώσσα JavaScript που διευκολύνει την διαχείριση των αντικειμένων της Javascript που έχουν να κάνουν με ημερομηνίες και χρόνο. [52] Μπορεί επίσης να συνδυαστεί και με την γλώσσα προγραμματισμού Typescript.

Το Moment.js διευκολύνει την εμφάνιση, μορφοποίηση, ανάλυση, επικύρωση και τον χειρισμό των ημερομηνιών και της ώρας χρησιμοποιώντας ένα καθαρό και συνοπτικό API. [52]

Επίσης παρέχει χρήσιμα χαρακτηριστικά όπως:

- Εύχρηστο μορφοποιητή για ημερομηνίες (date formatter). [52]
- Χρονόμετρα (timers). [52]
- ISO calendars. [52]
- Google Calendar API. [52]
- Υποστήριξη για τις ζώνες ώρας (time zones support). [52]
- Υποστήριξη πολλών γλωσσών (multi-languages support). [52]
- Τοπική ώρα. (localized time). [52]
- Λειτουργίες για τον χειρισμό ημερομηνιών (date manipulation functionalities). [52]

Το **Moment.js** χρησιμοποιήθηκε στην Angular εφαρμογή του client-side του συστήματος διαχείρισης παρουσιών για την διαχείριση και κατάλληλη **μορφοποίηση**:

- Της διάρκειας των θεωρητικών και εργαστηριακών διαλέξεων σε επίπεδο προγράμματος μαθήματος.
- Της ώρας έναρξης παρακολούθησης συνεδρίας διάλεξης μιας ομάδας φοιτητών.
- Της ημερομηνίας έναρξης μιας συνεδρίας διάλεξης.

Η έκδοση του Moment.js που χρησιμοποιήθηκε στην εφαρμογή είναι η 2.29.1.

Κεφάλαιο 4: Οδηγίες εγκατάστασης της εφαρμογής

Στο παρόν κεφάλαιο αναλύεται η διαδικασία εγκατάστασης τόσο του server-side, όσο και του client-side της εφαρμογής του συστήματος διαχείρισης παρουσιών.

Οδηγίες εγκατάστασης του server-side τμήματος της εφαρμογής:

Η εφαρμογή του server side του συστήματος διαχείρισης παρουσιών μπορεί να κατέβει στον υπολογιστή με τους εξής δυο τρόπους:

- Πατώντας το εξής [link](#) και επιλέγοντας την ένδειξη “Code” και “Download ZIP”.
- Σε περίπτωση που κάποιος έχει εγκατεστημένη την τεχνολογία git στον υπολογιστή του, μπορεί να χρησιμοποιήσει στο terminal του την παρακάτω εντολή:

```
git clone git@github.com:john-makris/pms-backend.git
```

Προκειμένου να μπορέσει να τρέξει η **spring boot εφαρμογή** πρέπει να είναι εγκατεστημένες οι εξής τεχνολογίες:

- Η Java 11 (jdk 11.0.4) η οποία μπορεί να κατέβει απο τον εξής ιστότοπο: <http://jdk.java.net/java-se-ri/11>
- MySql 8.0.23 η οποία μπορεί να κατέβει απο τον ιστότοπο: <https://downloads.mysql.com/archives/community/>

Σε αυτό το σημείο πρέπει να τονιστεί ότι θα πρέπει να ενημερωθούν τα πεδία “spring.datasource.username” και “spring.datasource.password” στο αρχείο **application.properties** της εφαρμογής, με το **username** και το **password** που χρησιμοποιούνται στην βάση δεδομένων του χρήστη.

Για το τρέξιμο της εφαρμογής μπορεί να χρησιμοποιηθεί το **Eclipse IDE for Enterprise Java and Web Developers**, ή όποιο άλλο μέσο βολεύει τον χρήστη. Τέλος όσον αφορά το επίπεδο της εύχρηστης διαχείρισης της βάσης δεδομένων, προτείνεται το εργαλείο διεπαφής χρήστη **MySQL Workbench**. Ο χρήστης μπορεί να κατεβάσει την κατάλληλη έκδοση απο τον ιστότοπο <https://dev.mysql.com/downloads/workbench/>.

Οδηγίες εγκατάστασης του client-side τμήματος της εφαρμογής:

Η εφαρμογή του client side του συστήματος διαχείρισης παρουσιών μπορεί να κατέβει στον υπολογιστή με τους εξής δυο τρόπους:

- Πατώντας το εξής [link](#) και επιλέγοντας την ένδειξη “Code” και “Download ZIP”.
- Σε περίπτωση που κάποιος έχει εγκατεστημένη την τεχνολογία git στον υπολογιστή του, μπορεί να χρησιμοποιήσει στο terminal του την παρακάτω εντολή:

```
git clone git@github.com:john-makris/pms-frontend.git
```

Προκειμένου να μπορέσει να τρέξει η **Angular εφαρμογή** πρέπει να είναι εγκατεστημένες οι εξής τεχνολογίες:

- npm έκδοση 8.0.0, η οποία μπορεί να εγκατασταθεί με την εξής εντολή:
npm install npm@8.0.0 για τοπική εγκατάσταση ή **npm install -g npm@8.0.0** για εγκατάσταση globally.
- Node.js έκδοση 14.17.2, η οποία μπορεί να εγκατασταθεί από τον ιστότοπο <https://nodejs.org/en/download/>.
- Angular CLI έκδοση 11.0.5, η οποία μπορεί να εγκατασταθεί με την εντολή:
npm install -g @angular/cli@11.0.5 για εγκατάσταση globally, ή
npm install @angular/cli@11.0.5 για τοπική εγκατάσταση.

Επίσης μετά την εγκατάσταση των παραπάνω τεχνολογιών, αν δεν είναι ήδη εγκατεστημένες, θα πρέπει να υλοποιηθεί η εντολή **npm -i**, ώστε να εγκατασταθούν τα **node_modules** της Angular εφαρμογής. Για να τρέξει η client side εφαρμογή και να ανοίξει στον προκαθορισμένο browser θα πρέπει να υλοποιηθεί μέσω του Angular CLI, η εντολή **ng serve -- open**.

Το url εισόδου στο σύστημα είναι το **127.0.0.1:4200**, καθώς αυτό τρέχει τοπικά (locally).

Τέλος για να τρέξει συνολικά η εφαρμογή θα πρέπει να έχει τεθεί σε λειτουργία παράλληλα με το client side και το server side.

Κεφάλαιο 5: Σενάρια χρήσης

Σε αυτό το κεφάλαιο παρουσιάζονται τα σενάρια χρήσης για τις λειτουργίες που επιτελεί στο σύστημα διαχείρισης παρουσιών ο κάθε ρόλος που συμμετέχει σε αυτό.

Αρχικά παρουσιάζονται τα σενάρια χρήσης που αφορούν όλους τους χρήστες. Επειτα παρουσιάζονται τα σενάρια χρήσης για τις λειτουργίες που επιτελεί μόνο ο διαχειριστής του συστήματος (Admin), ο καθηγητής (Teacher), ο γραμματέας (Secretary) και ο φοιτητής (Student).

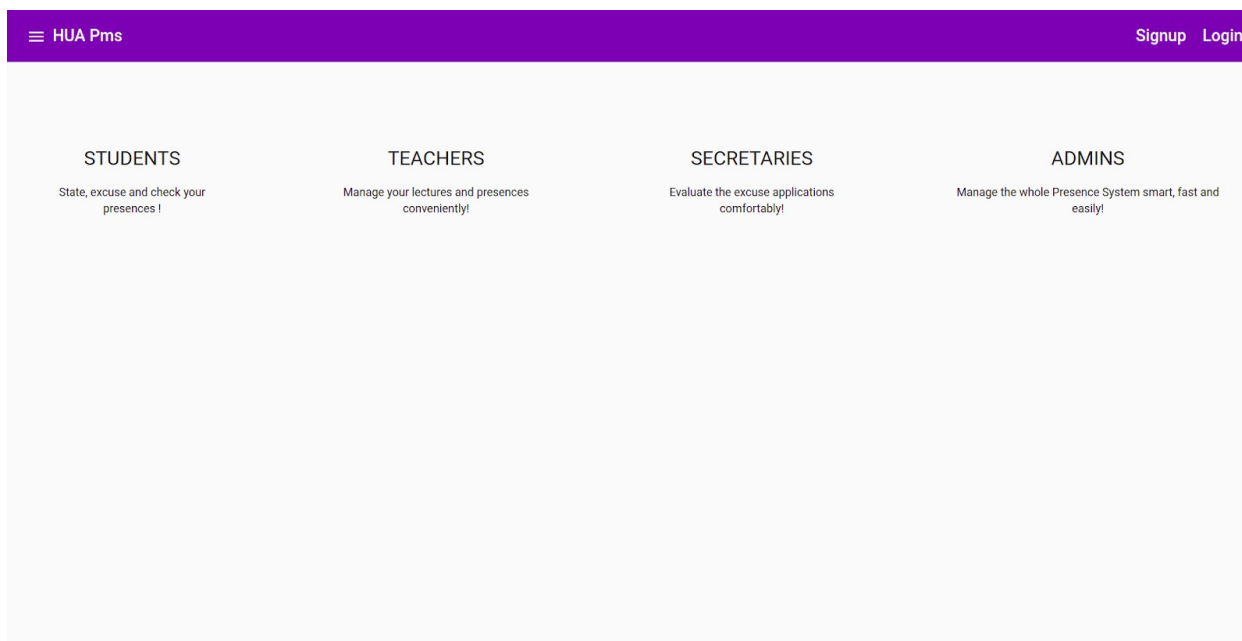
Είναι πολύ σημαντικό να τονιστεί ότι στην ενότητα που αφορά τον διαχειριστή του συστήματος, τα σενάρια χρήσης που παρουσιάζονται, σχετίζονται με τις λειτουργίες που επιτελεί στην εφαρμογή, μόνο ο διαχειριστής του συστήματος και κανένας άλλος ρόλος. Αυτό συμβαίνει διότι τα σενάρια χρήσης που παρουσιάζονται στις ενότητες που αφορούν τις λειτουργίες του καθηγητή και του γραμματέα, επιτελούνται επίσης και απο τον διαχειριστή του συστήματος.

5.1. Γενικές λειτουργίες

Σε αυτή την υποενότητα παρουσιάζονται τα σενάρια για δύο γενικές λειτουργίες του συστήματος.

Προβολή Αρχικής Σελίδας Εφαρμογής (Home Page)

Το Home Page της εφαρμογής είναι προσβάσιμο απο οποιονδήποτε χρήστη του συστήματος είτε είναι συνδεδεμένος σε αυτό είτε όχι.



Εικ.17. Home Page

Προβολή προφίλ και αλλαγή κωδικού πρόσβασης

Οποιοσδήποτε χρήστης του συστήματος είναι συνδεδεμένος σε αυτό μπορεί:

- Να βλέπει τις πληροφορίες του προφίλ του πατώντας στην ένδειξη με το όνομα και το επώνυμό του δίπλα απο το Logout πάνω δεξιά στην σελίδα ή επιλέγοντας την ένδειξη “Profile” απο το μενού που εμφανίζεται στο πλάι.

Ενα παράδειγμα προβολής προφίλ φαίνεται στην παρακάτω εικόνα.

Account Details

First name Anargyros	Last name Tsadimas
Username tsadimas	Email tsadimas@hua.gr
Roles Teacher, Admin	Department Informatics & Telematics

Change your password

Εικ.18. Account Details

- Να αλλάζει τον κωδικό πρόσβασής του, όπως φαίνεται στην παρακάτω εικόνα.

Change your Password !

Old Password	12 / 18
New Password	10 / 10
Retype new Password	10 / 10

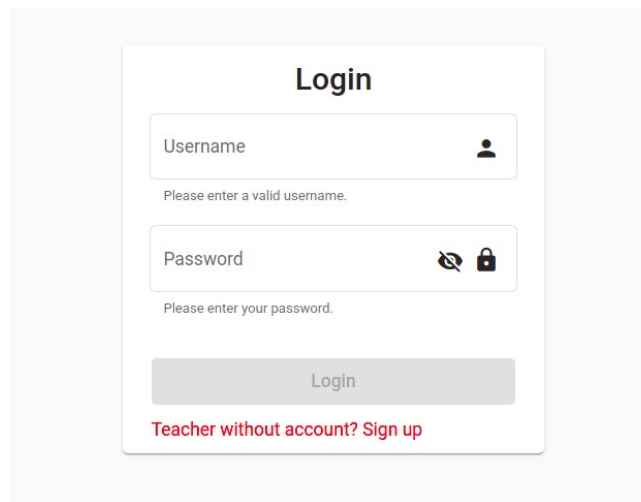
OK Close

Εικ.19. Change Password

5.2. Σύνδεση και εγγραφή στην εφαρμογή (Login, Register Page)

Σύνδεση στην εφαρμογή

Κατα την έναρξη της πλοήγησης του χρήστη, εμφανίζεται η σελίδα για login στο σύστημα.

The image shows a login form titled "Login" centered on a light gray background. The form is a white rounded rectangle containing two input fields. The first field is labeled "Username" with a user icon on the right and a placeholder text "Please enter a valid username." below it. The second field is labeled "Password" with an eye icon and a lock icon on the right, and a placeholder text "Please enter your password." below it. At the bottom of the form is a gray "Login" button. Below the button, the text "Teacher without account? Sign up" is displayed in red.

Εικ.20. Φόρμα Login

Αν ο χρήστης έχει λογαριασμό, μπορεί να συμπληρώσει το username και το password του και να συνδεθεί στο σύστημα.

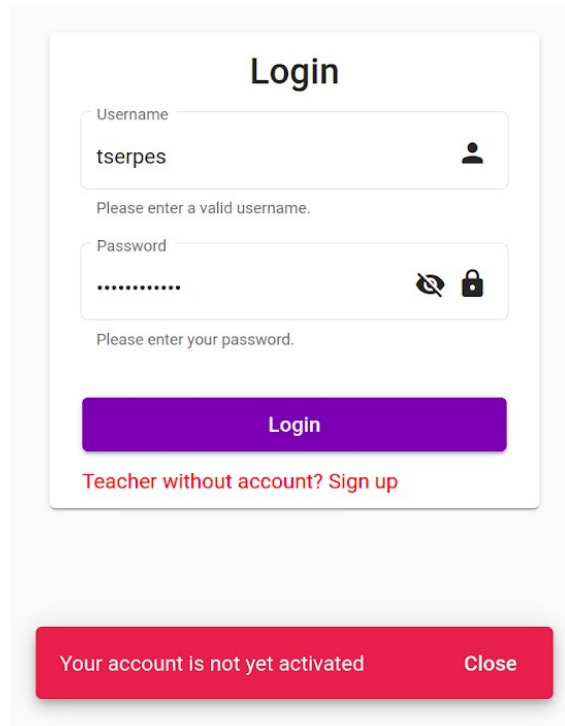
Το σενάριο σύνδεσης του χρήστη στο σύστημα συνοδεύεται απο άλλα τρία υποσενάρια:

- Ο χρήστης εισάγει λανθασμένα στοιχεία.

The screenshot displays a web login interface. At the top, the title "Login" is centered. Below it, there are two input fields. The first field, labeled "Username", contains the text "tsadimas" and has a small user icon to its right. Below this field, a red error message reads "Please enter a valid username.". The second field, labeled "Password", contains a series of dots and has icons for toggling visibility and a lock symbol to its right. Below this field, a red error message reads "Please enter your password.". A large blue button labeled "Login" is positioned below the password field. At the bottom of the form, there is a red link that says "Teacher without account? Sign up". Below the entire form, a red alert box is visible with the text "Bad credentials" on the left and a "Close" button on the right.

Εικ.21. Login - Bad Credentials

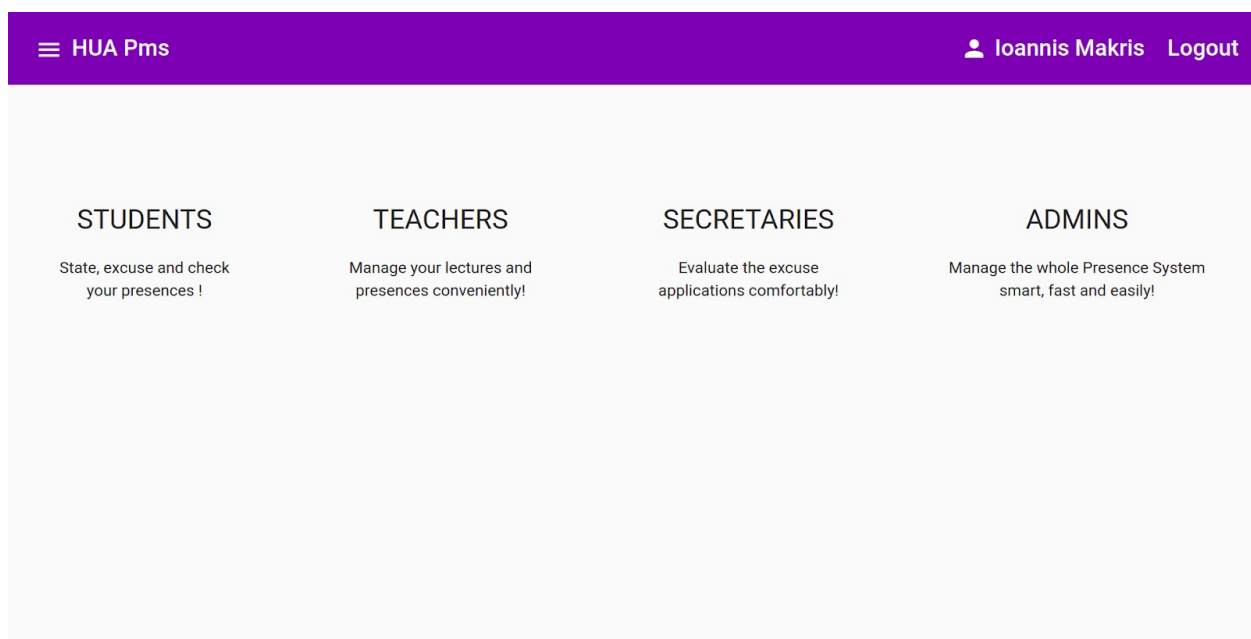
- Ο λογαριασμός του χρήστη δεν είναι ακόμη ενεργοποιημένος απο τον διαχειριστή του συστήματος.



The image shows a login interface with the title "Login". It contains two input fields: "Username" with the value "tserpes" and "Password" with masked characters ".....". Below the username field is the message "Please enter a valid username." and below the password field is "Please enter your password." There is a purple "Login" button. Below the button is a red link that says "Teacher without account? Sign up". At the bottom of the form, there is a red error message box that says "Your account is not yet activated" with a "Close" button.

Εικ.22. Login - Inactive Account

- Ο χρήστης συνδέεται με επιτυχία στο σύστημα.

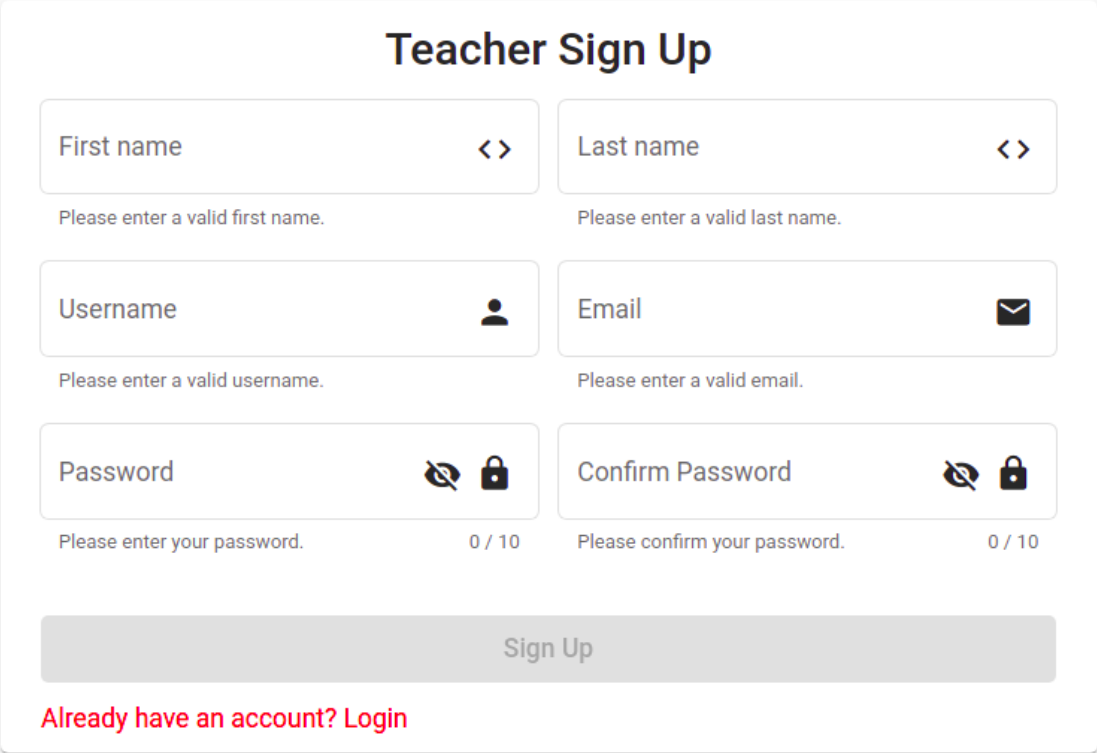


Εικ.23. Επιτυχής Σύνδεση Χρήστη

Εγγραφή στην εφαρμογή

Σε περίπτωση που ο χρήστης δεν έχει λογαριασμό ώστε να συνδεθεί στο σύστημα:

- Αν είναι φοιτητής ή γραμματέας μπορεί να επικοινωνήσει με τον διαχειριστή του συστήματος μέσω του email της σχολής ώστε να του δημιουργήσει έναν.
- Αν είναι καθηγητής, μπορεί να πλοηγηθεί στην φόρμα εγγραφής καθηγητών που φαίνεται στην παρακάτω εικόνα, συμπληρώνοντας τα απαιτούμενα στοιχεία.



The image shows a 'Teacher Sign Up' form with the following fields and elements:

- First name**: Input field with a '<>' icon. Below it, the text 'Please enter a valid first name.' is displayed.
- Last name**: Input field with a '<>' icon. Below it, the text 'Please enter a valid last name.' is displayed.
- Username**: Input field with a user icon. Below it, the text 'Please enter a valid username.' is displayed.
- Email**: Input field with an envelope icon. Below it, the text 'Please enter a valid email.' is displayed.
- Password**: Input field with an eye and lock icon. Below it, the text 'Please enter your password.' and a character count '0 / 10' are displayed.
- Confirm Password**: Input field with an eye and lock icon. Below it, the text 'Please confirm your password.' and a character count '0 / 10' are displayed.
- Sign Up**: A large grey button at the bottom of the form.
- Already have an account? Login**: A red text link below the 'Sign Up' button.

Εικ.24. Φόρμα Εγγραφής Καθηγητή

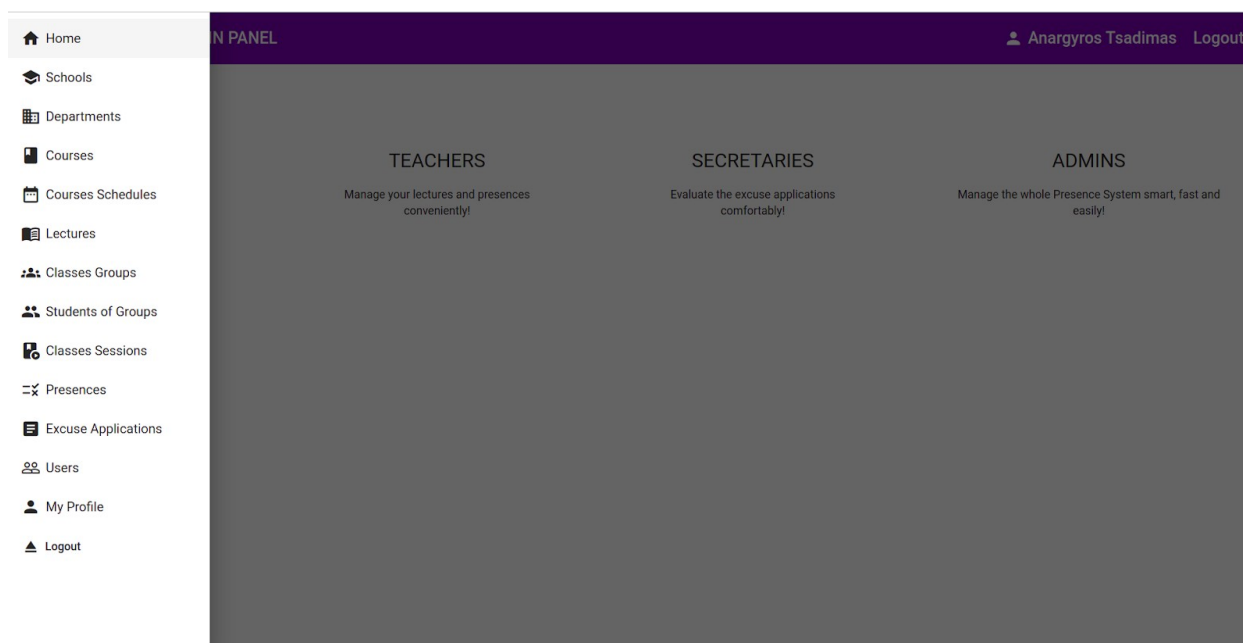
5.3. Λειτουργίες Διαχειριστή (Admin)

Στην ενότητα αυτή παρουσιάζονται τα σενάρια χρήσης για τις λειτουργίες που επιτελούνται **μόνο απο τον ρόλο του διαχειριστή συστήματος (Admin) και κανέναν άλλον ρόλο.**

Στις ενότητες που ακολουθούν και σχετίζονται με τον ρόλο του καθηγητή (Teacher) και του γραμματέα (Secretary), όλα τα σενάρια χρήσης αφορούν στην πραγματικότητα λειτουργίες που επιτελεί και ο διαχειριστής του συστήματος (Admin). Η μόνη διαφορά είναι ότι ο καθηγητής (Teacher), και ο γραμματέας (Secretary) δρουν αποκλειστικά στα πλαίσια του τμήματος στο οποίο ανήκουν και όπου έχουν την ανάλογη δικαιοδοσία, ενώ ο διαχειριστής του συστήματος (Admin) δεν υπόκειται σε τέτοιους περιορισμούς.

Πλοήγηση στο μενού επιλογών

Ο διαχειριστής αφού συνδεθεί στο σύστημα, μπορεί να δει το μενού επιλογών του και να πλοηγηθεί σε αυτό, πατώντας το κουμπί που αναπαρίσταται με τρεις οριζόντιες παύλες πάνω αριστερά στην οθόνη.



Εικ.25. Μενού Διαχειριστή Συστήματος

Δημιουργία Σχολής

Schools

Add School

Search schools

ID ↑	Name
1	School of Digital Technology
2	School of Health Science and Education
3	School of Environment, Geography and Applied Economics

1 – 3 of 3

<<

<

>

>>

Items per page: 3

Add School

Name

Please enter a valid name.

Location

Please enter a valid location.

Save

Cancel

Εικ.26. Δημιουργία Σχολής

Ο διαχειριστής του συστήματος, μπορεί να προσθέσει τις σχολές του πανεπιστημίου συμπληρώνοντας την αντίστοιχη φόρμα που εμφανίζεται δεξιά του πίνακα, μετά το πάτημα του κουμπιού “Add School”.

Προβολή στοιχείων σχολής

The screenshot displays a web interface for managing schools. On the left, a table lists three schools. Above the table is a search bar and an 'Add School' button. On the right, a 'School Details' modal is open for the first school, showing its name and location, with 'Edit' and 'Delete' buttons.

ID ↑	Name
1	School of Digital Technology
2	School of Health Science and Education
3	School of Environment, Geography and Applied Economics

1 – 3 of 3 |< < > >|

Items per page: 3

School Details

School Name: School of Digital Technology

Location: Omirou 9, Tauros 177 78

Edit Delete

Cancel

Εικ.27. Προβολή στοιχείων σχολής

Ο διαχειριστής του συστήματος, μπορεί να αναζητήσει σχολές μέσω της μπάρας αναζήτησης στο πάνω μέρος του πίνακα με κριτήριο το “Name”.

Επειτα πλοηγούμενος στον πίνακα με τις σχολές, μπορεί να επιλέξει μια σχολή πατώντας στην γραμμή του πίνακα στην οποία αντιστοιχεί και να δει περισσότερες λεπτομέρειες σχετικά με αυτή, στην καρτέλα που εμφανίζεται δεξιά του πίνακα.

Επεξεργασία στοιχείων σχολής

Schools

Add School

Search schools

ID ↑	Name
1	School of Digital Technology
2	School of Health Science and Education
3	School of Environment, Geography and Applied Economics

1 – 3 of 3

<<

<

>

>>

Items per page: 3

Edit School

Name

School of Digital Technology

Please enter a valid name.

Location

Omirou 9, Tauros 177 78

Please enter a valid location.

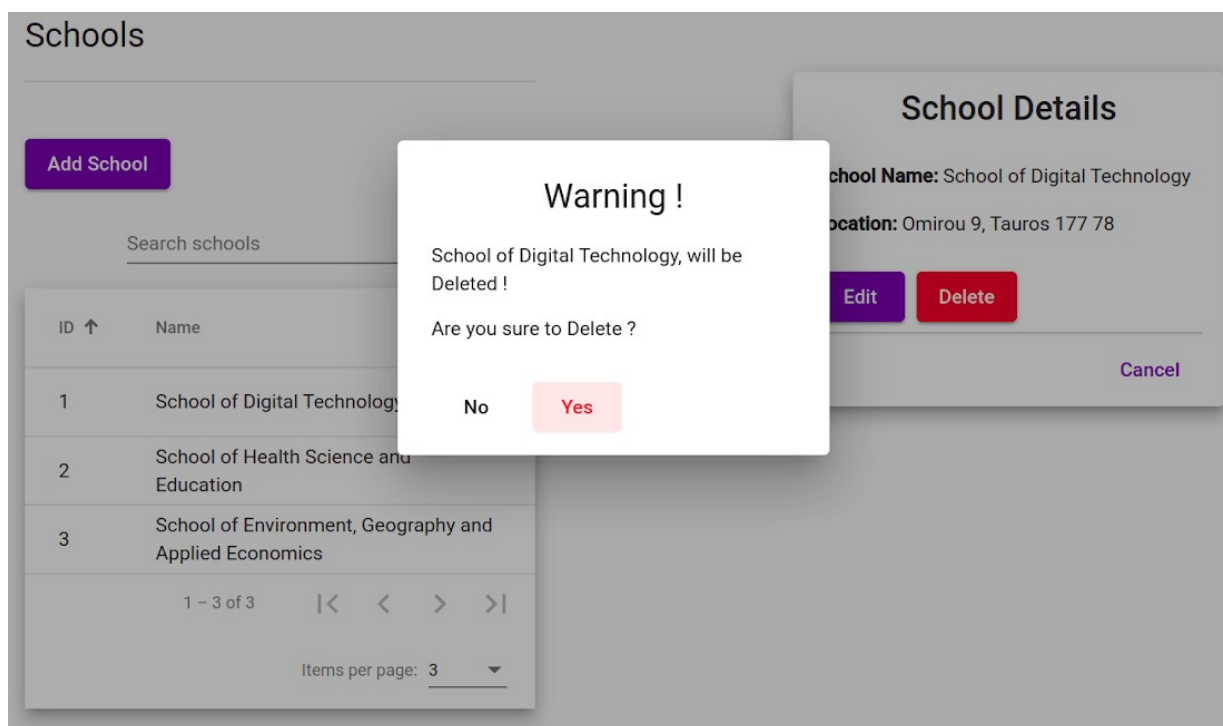
Update

Cancel

Εικ.28. Επεξεργασία στοιχείων σχολής

Στην καρτέλα προβολής των στοιχείων της σχολής, ο διαχειριστής του συστήματος μπορεί να επιλέξει το κουμπί “Edit” και να επεξεργαστεί τα στοιχεία της σχολής στην αντίστοιχη φόρμα που εμφανίζεται απο το σύστημα.

Διαγραφή σχολής



Εικ.29. Διαγραφή Σχολής

Στην καρτέλα προβολής των στοιχείων της σχολής, ο διαχειριστής του συστήματος μπορεί να επιλέξει το κουμπί “Delete” για να διαγράψει την επιλεγμένη σχολή. Το σύστημα θα εμφανίσει ένα αναδυόμενο παράθυρο το οποίο ρωτάει τον διαχειριστή του συστήματος, αν είναι βέβαιος για την διαγραφή ή όχι, οπότε αν θέλει να διαγράψει την επιλεγμένη σχολή πατάει το κουμπί “Yes”.

Για να γίνει η διαγραφή μιας σχολής, θα πρέπει να μην έχει δημιουργηθεί ακόμη κάποιο τμήμα για την σχολή αυτή, διαφορετικά εμφανίζεται στον διαχειριστή του συστήματος κατάλληλο μήνυμα όπως φαίνεται στην παρακάτω εικόνα.

Schools

Add School

Search schools

ID ↑	Name
1	School of Digital Technology
2	School of Health Science and Education
3	School of Environment, Geography and Applied Economics

1 – 3 of 3

<

>

Items

School Details

School Name: School of Digital Technology

Location: Omirou 9, Tauros 177 78

Edit

Delete

Cancel

You should first delete school's Departments !

Close

Εικ.30. Αδυναμία Διαγραφής Σχολής

Δημιουργία τμήματος

The screenshot displays the 'Departments' management interface. On the left, a table lists existing departments. On the right, a modal form titled 'Add Department' is open, allowing for the creation of a new department.

Departments Table:

ID ↑	Name
1	Informatics & Telematics
2	Nutrition and Dietetics
3	Economics and Sustainable Development

Below the table, pagination shows '1 - 3 of 5' and navigation arrows.

Add Department Modal:

- Name:** A text input field with a validation message: 'Please enter a valid name.'
- School:** A dropdown menu currently set to '-- None --'.
- Buttons:** 'Save' and 'Cancel' buttons at the bottom.

Εικ.31. Δημιουργία τμήματος

Ο διαχειριστής του συστήματος, μπορεί να προσθέσει τμήματα για κάθε σχολή του πανεπιστημίου συμπληρώνοντας την αντίστοιχη φόρμα που εμφανίζεται δεξιά του πίνακα, μετά το πάτημα του κουμπιού “Add Department”.

Προβολή στοιχείων τμήματος

The screenshot displays a web application for managing departments. On the left, the 'Departments' section features a dropdown menu for 'School' (currently set to 'All Schools'), a search bar, and an 'Add Department' button. Below this is a table listing departments:

ID ↑	Name
1	Informatics & Telematics
2	Nutrition and Dietetics
3	Economics and Sustainable Development

At the bottom of the table, it indicates '1 - 3 of 5' items with navigation arrows. On the right, a 'Department Details' modal is open for the selected department. It displays the 'Department Name' as 'Informatics & Telematics' and the 'School' as 'School of Digital Technology'. There are 'Edit' and 'Delete' buttons, and a 'Cancel' button at the bottom right.

Εικ.32. Προβολή στοιχείων τμήματος

Ο διαχειριστής του συστήματος, μπορεί να επιλέξει τα τμήματα ανα σχολή, μέσω του ειδικού επιλογέα. Παράλληλα μπορεί να αναζητήσει τμήματα μέσω της μπάρας αναζήτησης στο πάνω μέρος του πίνακα με κριτήριο το “Name”.

Επειτα πλοηγούμενος στον πίνακα με τα τμήματα, μπορεί να επιλέξει ένα τμήμα πατώντας την γραμμή του πίνακα στην οποία αντιστοιχεί. Έτσι έχει την δυνατότητα να δει περισσότερες λεπτομέρειες σχετικά με αυτό, στην καρτέλα που εμφανίζεται δεξιά του πίνακα.

Επεξεργασία στοιχείων τμήματος

Departments

School

-- All Schools --

Choose School

Add Department

Search departments

ID ↑	Name
1	Informatics & Telematics
2	Nutrition and Dietetics
3	Economics and Sustainable Development

1 – 3 of 5

Edit Department

Name

Informatics & Telematics

Please enter a valid name.

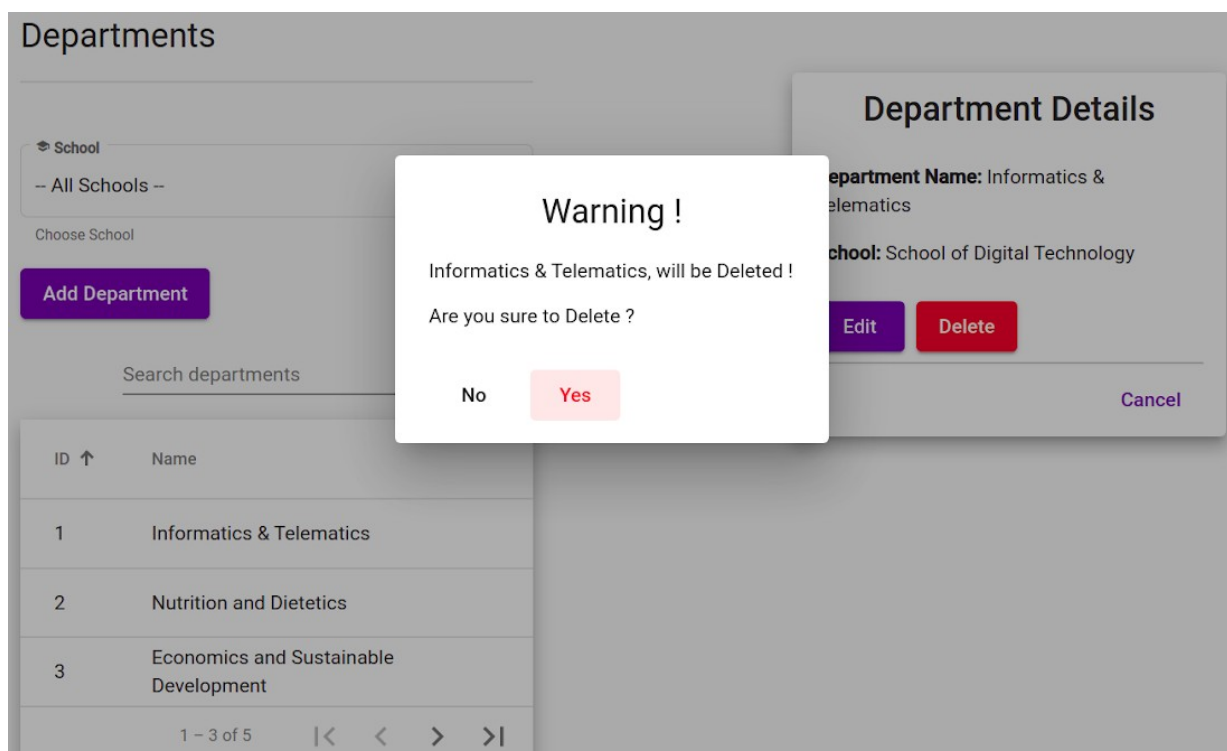
Update

Cancel

Εικ.33. Επεξεργασία στοιχείων τμήματος

Στην καρτέλα προβολής των στοιχείων του τμήματος, ο διαχειριστής του συστήματος μπορεί να επιλέξει το κουμπί “Edit” και να επεξεργαστεί τα στοιχεία του στην αντίστοιχη φόρμα που εμφανίζεται απο το σύστημα.

Διαγραφή τμήματος



Εικ.34. Διαγραφή τμήματος

Στην καρτέλα προβολής των στοιχείων του τμήματος, ο διαχειριστής του συστήματος μπορεί να επιλέξει το κουμπί “Delete” για να διαγράψει το επιλεγμένο τμήμα. Το σύστημα θα εμφανίσει ένα αναδυόμενο παράθυρο το οποίο ρωτάει τον διαχειριστή του συστήματος, αν είναι βέβαιος για την διαγραφή ή όχι, οπότε αν θέλει να διαγράψει το επιλεγμένο τμήμα πατάει το κουμπί “Yes”.

Για να γίνει η διαγραφή ενός τμήματος, θα πρέπει να μην έχει δημιουργηθεί ακόμη κάποιο μάθημα που να αντιστοιχεί σε αυτό, διαφορετικά εμφανίζεται στον διαχειριστή του συστήματος κατάλληλο μήνυμα όπως φαίνεται στην παρακάτω εικόνα.

Departments

School

-- All Schools --

Choose School

Add Department

Search departments

ID ↑	Name
1	Informatics & Telematics
2	Nutrition and Dietetics
3	Economics and Sustainable Development

1 - 3 of 5

Department Details

Department Name: Informatics & Telematics

School: School of Digital Technology

EditDelete

Cancel

You should first delete department's courses !Close

Εικ.35. Αδυναμία Διαγραφής Τμήματος

Δημιουργία μαθήματος

The screenshot displays the 'Courses' management interface. On the left, a table lists existing courses with columns for ID, Name, and Semester. Below the table is a pagination control showing 'Items per page: 3' and '1 - 3 of 22'. On the right, an 'Add Course' modal form is open, featuring input fields for Name, Semester, and Department, along with 'Save' and 'Cancel' buttons.

Courses

Department
-- All Departments --
Choose Department

Add Course

Name

Please enter a valid name.

Semester
-- None --
Choose Semester

Department
-- None --
Choose Department

Save Cancel

ID ↑	Name	Semester
1	Introduction to C	First
2	Java I	Third
3	Propabilities	Second

Items per page: 3 1 - 3 of 22 |< < > >|

Εικ.36. Δημιουργία μαθήματος

Ο διαχειριστής του συστήματος, μπορεί να προσθέσει τα μαθήματα για κάθε τμήμα, συμπληρώνοντας την αντίστοιχη φόρμα που εμφανίζεται δεξιά του πίνακα, μετα την επιλογή προσθήκης μαθήματος.

Προβολή στοιχείων μαθήματος

The screenshot displays a web application for managing courses. On the left, the 'Courses' section features a dropdown menu for 'Department' set to 'Informatics & Telematics', an 'Add Course' button, and a search bar. Below is a table of courses:

ID ↑	Name	Semester
1	Introduction to C	First
2	Java I	Third
3	Propabilities	Second

At the bottom of the table, there is a pagination control showing 'Items per page: 3' and '1 - 3 of 22' items, along with navigation arrows. On the right, the 'Course Details' modal is open, showing the following information:

- Course Name:** Introduction to C
- Semester:** First
- Department:** Informatics & Telematics

The modal includes 'Edit' and 'Delete' buttons, and a 'Cancel' button at the bottom right.

Εικ.37. Προβολή στοιχείων μαθήματος

Ο διαχειριστής του συστήματος, μπορεί να επιλέξει τα μαθήματα ανα τμήμα. Παράλληλα μπορεί να αναζητήσει μαθήματα μέσω της μπάρας αναζήτησης στο πάνω μέρος του πίνακα με κριτήρια το “Name” ή το “Semester”.

Επειτα πλοηγούμενος στον πίνακα με τα μαθήματα, μπορεί να επιλέξει ένα μάθημα πατώντας στην γραμμή του πίνακα στην οποία αντιστοιχεί. Έτσι έχει την δυνατότητα να δει περισσότερες λεπτομέρειες σχετικά με αυτό, στην καρτέλα που εμφανίζεται δεξιά του πίνακα.

Επεξεργασία στοιχείων μαθήματος

Courses

Department

Informatics & Telematics

Choose Department

Add Course

Search courses

ID ↑	Name	Semester
1	Introduction to C	First
2	Java I	Third
3	Propabilities	Second

Items per page: 3 1 - 3 of 22 << < > >>

Edit Course

Name

Introduction to C

Please enter a valid name.

Semester

First

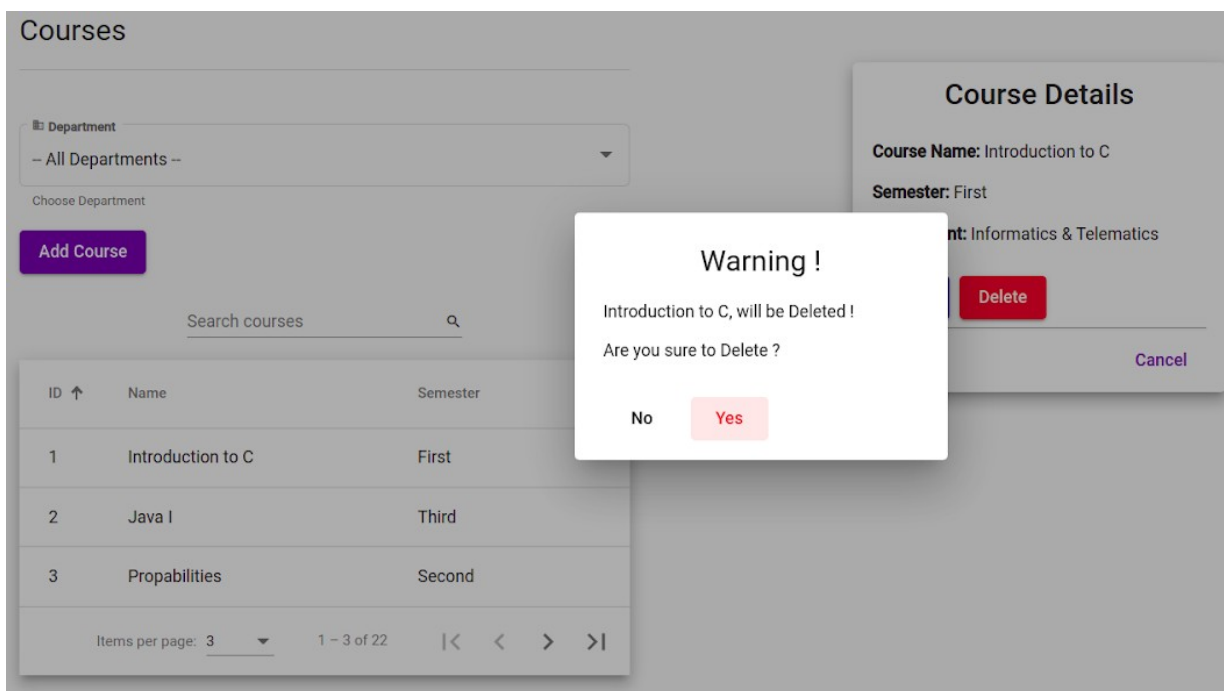
Choose Semester

Update Cancel

Εικ.38. Επεξεργασία στοιχείων μαθήματος

Στην καρτέλα προβολής των στοιχείων του μαθήματος, ο διαχειριστής του συστήματος μπορεί να επιλέξει το κουμπί “Edit” και να επεξεργαστεί τα στοιχεία του στην αντίστοιχη φόρμα που εμφανίζεται απο το σύστημα.

Διαγραφή μαθήματος



Εικ.39. Διαγραφή μαθήματος

Στην καρτέλα προβολής των στοιχείων του μαθήματος, ο διαχειριστής του συστήματος μπορεί να επιλέξει το κουμπί "Delete" για να διαγράψει το επιλεγμένο μάθημα. Το σύστημα θα εμφανίσει ένα αναδυόμενο παράθυρο το οποίο ρωτάει τον διαχειριστή του συστήματος, αν είναι βέβαιος για την διαγραφή ή όχι, οπότε αν θέλει να διαγράψει το επιλεγμένο μάθημα πατάει το κουμπί "Yes".

Για να γίνει η διαγραφή ενός μαθήματος, θα πρέπει να μην έχει δημιουργηθεί ακόμη κάποιο πρόγραμμα μαθήματος που να αντιστοιχεί σε αυτό, διαφορετικά εμφανίζεται στον διαχειριστή του συστήματος κατάλληλο μήνυμα όπως φαίνεται στην παρακάτω εικόνα.

Courses

Department

-- All Departments --

Choose Department

Add Course

Search courses

ID ↑	Name	Semester
1	Introduction to C	First
2	Java I	Third
3	Propabilities	Second

Items per page: 3 1 – 3 of 22

Course Details

Course Name: Introduction to C

Semester: First

Department: Informatics & Telematics

Edit

Delete

Cancel

You cannot delete course Introduction to C since it has already a Schedule

Close

Εικ.40. Αδυναμία διαγραφής μαθήματος

Δημιουργία προγράμματος μαθήματος

The screenshot displays the 'Courses Schedules' management interface. On the left, a sidebar contains filters for 'Department' (set to 'Informatics & Telematics') and 'Status' (set to 'Current'), along with an 'Add Course Schedule' button. Below the filters is a search bar and a table of existing courses.

ID	Name	Semester
8	Parallel Computing and Algorithms	Seventh
6	Operating Systems	Third
5	Distributed Systems	Fifth

On the right, the 'Add Course Schedule' modal is open. It includes a 'Selected' course dropdown (currently 'Introduction to C'), an 'Upload csv/excel of Students' section with a 'Choose File' button and a file name 'course_sch...students.xlsx', and duration settings for 'Theory Duration' (3 hours, 0 minutes) and 'Lab Duration' (2 hours, 0 minutes). It also has 'Max Theories' (3) and 'Max Labs' (9) fields. At the bottom, there is a 'Selected' teacher dropdown (currently 'Dimitrios Michail, Konstantinos Tserpes') and 'Save' and 'Cancel' buttons.

Εικ.41. Δημιουργία προγράμματος μαθήματος

Ο διαχειριστής του συστήματος πρέπει πρώτα να επιλέξει το τμήμα στο οποίο θέλει να φτιάξει πρόγραμμα μαθήματος για ένα μάθημα. Επειτα μπορεί να προσθέσει ένα πρόγραμμα μαθήματος για ένα μάθημα συμπληρώνοντας την αντίστοιχη φόρμα που εμφανίζεται δεξιά του πίνακα, μετά το πάτημα του κουμπιού “Add Course Schedule”.

Προβολή στοιχείων προγράμματος μαθήματος

Courses Schedules

Department

Informatics & Telematics

Choose Department

Status

Current

Choose Status

Add Course Schedule

Search courses schedules

ID ↑	Name	Semester
1	Introduction to C	First
3	Java I	Third
5	Distributed Systems	Fifth

Items per page: 3 1 - 3 of 5 < < > >|

Course Schedule Details

Course Name: Introduction to C

Department: Informatics & Telematics

Academic Year: 2021 - 2022

Semester: First

Max Theory Lectures: 4

Max Lab Lectures: 8

Theory Duration: 3 hours

Lab Duration: 2 hours

Teaching Stuff: Dimitrios Michail

Students: [Preview](#)

Status: Current

Edit

Delete

Cancel

Εικ.42. Προβολή στοιχείων προγράμματος μαθήματος

Ο διαχειριστής του συστήματος, μπορεί να επιλέξει τα μαθήματα ανα τμήμα ή κατάσταση (status). Η κατάσταση (status) των μαθημάτων μπορεί να είναι current, past και pending.

Παράλληλα μπορεί να αναζητήσει προγράμματα μαθήματος μέσω της μπάρας αναζήτησης στο πάνω μέρος του πίνακα με κριτήριο το “Name” ή το “Semester”.

Επειτα πλοηγούμενος στον πίνακα με τα προγράμματα μαθημάτων, μπορεί να επιλέξει ένα πρόγραμμα μαθήματος πατώντας την γραμμή του πίνακα στην οποία αντιστοιχεί. Έτσι έχει την δυνατότητα να δει περισσότερες λεπτομέρειες σχετικά με αυτό, στην καρτέλα που εμφανίζεται δεξιά του πίνακα.

Επειδή η πληροφορία που αφορά τους φοιτητές του προγράμματος μαθήματος έχει συνήθως μεγάλο όγκο δεδομένων, ο διαχειριστής του συστήματος μπορεί να πατήσει το κουμπί “Preview” ώστε να ανοίξει το κατάλληλο αναδυόμενο παράθυρο που περιέχει πίνακα με τους συμμετέχοντες φοιτητές.

Στην παρακάτω εικόνα φαίνεται το αναδυόμενο παράθυρο με τους φοιτητές του επιλεγμένου προγράμματος μαθήματος.

Courses Schedules

Department: Informatics & Telematics

Status: Current

Add Course Schedule

Search courses schedules

ID	Name	Semester
1	Introduction to C	First
3	Java I	Third
5	Distributed Systems	Fifth

Items per page: 3 1 - 3 of 5

List of Students:

Search student

ID	Username	First Name	Last Name
3	it21526	Ioannis	Makris
4	it21569	Giorgos	Loizidis
5	it21628	Vasilis	Matzaris

1 - 3 of 14

Items per page: 3

Close

Εικ.43. Φοιτητές προγράμματος μαθήματος

Επεξεργασία στοιχείων προγράμματος μαθήματος

The screenshot displays the 'Courses Schedules' management interface. On the left, there are filters for 'Department' (Informatics & Telematics) and 'Status' (Current), along with an 'Add Course Schedule' button. Below these is a search bar and a table of courses. The table has columns for ID, Name, and Semester, and lists three courses: 'Introduction to C' (ID 1, First Semester), 'Java I' (ID 3, Third Semester), and 'Distributed Systems' (ID 5, Fifth Semester). On the right, a modal titled 'Introduction to C Schedule' is open, allowing for the upload of a CSV/Excel file and the configuration of theory and lab durations. The modal includes input fields for hours and minutes for both theory and lab durations, as well as fields for the maximum number of theories and labs. The 'Update' button is highlighted in purple.

ID	Name	Semester
1	Introduction to C	First
3	Java I	Third
5	Distributed Systems	Fifth

Εικ.44. Επεξεργασία στοιχείων προγράμματος μαθήματος

Στην καρτέλα προβολής των στοιχείων του προγράμματος μαθήματος, ο διαχειριστής του συστήματος μπορεί να επιλέξει το κουμπί “Edit” και να επεξεργαστεί τα στοιχεία του προγράμματος μαθήματος στην αντίστοιχη φόρμα που εμφανίζεται από το σύστημα. Το πεδίο της φόρμας επεξεργασίας του προγράμματος μαθήματος που αφορά το csv/excel αρχείο με τους φοιτητές, δεν απαιτεί υποχρεωτική συμπλήρωση.

Παράλληλα μετά την επεξεργασία του προγράμματος μαθήματος, το σύστημα ελέγχει για το αν οι αλλαγές που έγιναν από τον διαχειριστή του συστήματος στα στοιχεία της φόρμας είναι εφικτές.

Σε περίπτωση που κάποια ή κάποιες αλλαγές δεν είναι εφικτές, όπως το να μικρύνει ο αριθμός των μέγιστων διαλέξεων θεωρίας, ενώ έχει δημιουργηθεί ήδη ο μέγιστος αριθμός αυτών, το σύστημα εμφανίζει κατάλληλο μήνυμα, όπως φαίνεται στην παρακάτω εικόνα.

Courses Schedules

Department

-- All Departments --

Choose Department

Add Course Schedule

Search courses schedules

ID	Name	Semester	Status
1	Introduction to C	First	Current
2	Propabilities	Second	Pending
3	Java I	Third	Current

Items per page: 31 - 3 of 7

Introduction to C Schedule

Upload csv/excel of Students:Choose FileNo file chosen

Theory Duration:

Hours3Minutes0

Set Hours. Set Minutes.

Max Theories3

Set Max Theories.

Lab Duration:

Hours2Minutes0

Set Hours. Set Minutes.

Max Labs8

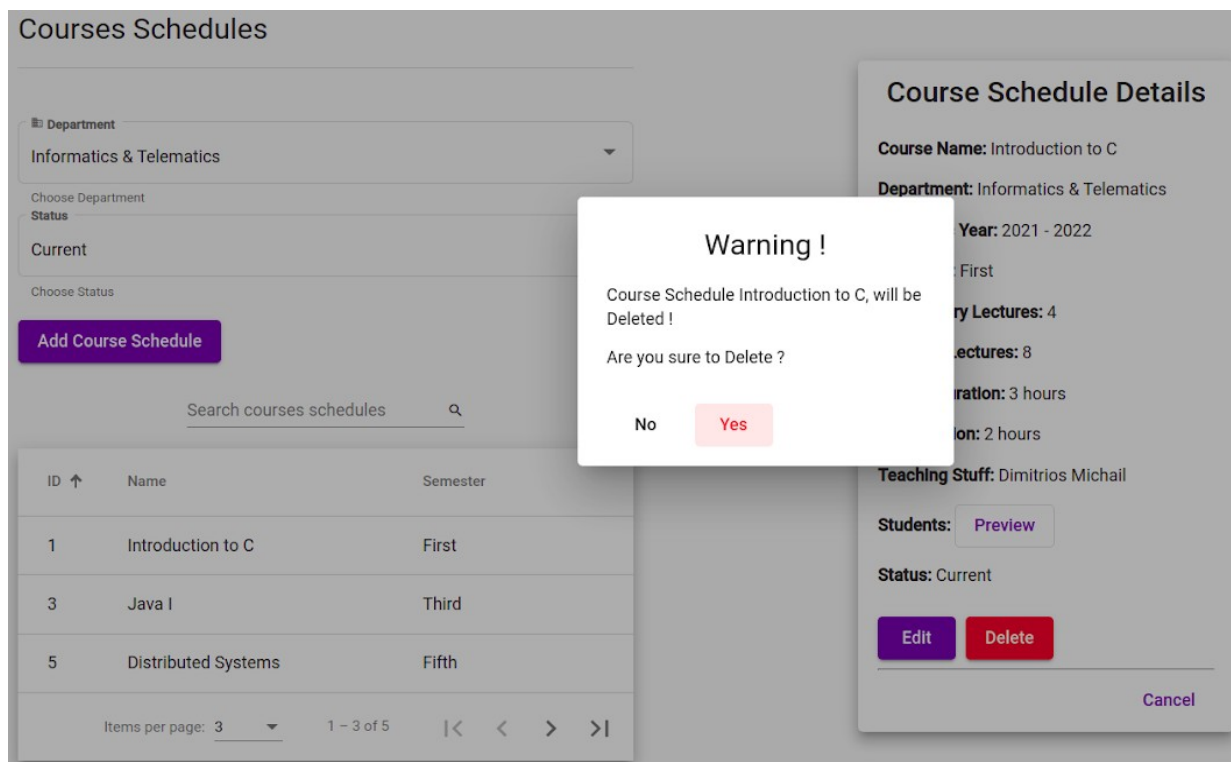
Set Max Labs.

UpdateCancel

You cannot set max theories to 3, because you have already 4 theories created for "Introduction to C" scheduleClose

Εικ.45. Αδυναμία επεξεργασίας προγράμματος μαθήματος

Διαγραφή προγράμματος μαθήματος



Εικ.46. Διαγραφή προγράμματος μαθήματος

Στην καρτέλα προβολής των στοιχείων του προγράμματος μαθήματος, ο διαχειριστής του συστήματος μπορεί να επιλέξει το κουμπί “Delete” για να διαγράψει το επιλεγμένο πρόγραμμα μαθήματος. Το σύστημα θα εμφανίσει ένα αναδυόμενο παράθυρο το οποίο ρωτάει τον διαχειριστή του συστήματος, αν είναι βέβαιος για την διαγραφή ή όχι, οπότε αν θέλει να διαγράψει το επιλεγμένο πρόγραμμα μαθήματος πατάει το κουμπί “Yes”.

Για να γίνει η διαγραφή ενός προγράμματος μαθήματος, θα πρέπει να μην έχει δημιουργηθεί ακόμη κάποια διάλεξη ή ομάδα φοιτητών που να αντιστοιχεί σε αυτό, διαφορετικά εμφανίζεται στον διαχειριστή του συστήματος κατάλληλο μήνυμα όπως φαίνεται στην παρακάτω εικόνα.

Courses Schedules

Department

Informatics & Telematics

Choose Department

Status

Current

Choose Status

Add Course Schedule

Search courses schedules

ID ↑	Name	Semester
1	Introduction to C	First
3	Java I	Third
5	Distributed Systems	Fifth

Items per page: 31 – 3 of 5

Course Schedule Details

Course Name: Introduction to C

Department: Informatics & Telematics

Academic Year: 2021 - 2022

Semester: First

Max Theory Lectures: 4

Max Lab Lectures: 8

Theory Duration: 3 hours

Lab Duration: 2 hours

Teaching Stuff: Dimitrios Michail

Students: [Preview](#)

Status: Current

[Edit](#)[Delete](#)

Cancel

You should first delete Introduction to C schedule's lectures ![Close](#)

Εικ.47. Αδυναμία διαγραφής προγράμματος μαθήματος

Δημιουργία αίτησης δικαιολόγησης απουσίας για έναν φοιτητή

The screenshot displays the 'Excuse Applications' interface. On the left, there is a form with the following fields:

- Department:** Informatics & Telematics (dropdown menu)
- Lecture Type:** -- All -- (dropdown menu)
- Status:** Pending (dropdown menu)
- Selected:** -- None -- (dropdown menu)

Below these fields is a purple button labeled 'Create Application'. Underneath the button is a search bar labeled 'Search applications' with a magnifying glass icon.

Below the search bar is a table with the following data:

Student	Course	Lecture
it21780	Introduction to C	lab_8
it21526	Introduction to C	lab_8

At the bottom of the table, there is a pagination bar showing 'Items per page: 3' and '1 - 2 of 2'.

On the right, there is a modal dialog titled 'Create Application'. It contains the following fields:

- Selected:** it21526 (with a close button 'x')
- Choose Student:** (empty field)
- Selected:** Introduction to C, lab_8 (with a close button 'x')
- Choose Absence:** (empty field)
- Reason:** I was ill and I could not be at lab_8 (with a close button 'x' and a document icon)

Below the 'Reason' field is a text prompt: 'Please enter a reason (up to 200 characters)'. At the bottom of the modal are two buttons: 'Apply' (purple) and 'Cancel' (white).

Εικ.48. Δημιουργία αίτησης δικαιολόγησης απουσίας για έναν φοιτητή

Ο διαχειριστής του συστήματος πρέπει πρώτα να επιλέξει τμήμα. Επειτα μπορεί να δημιουργήσει μια αίτηση δικαιολόγησης απουσίας για έναν φοιτητή συμπληρώνοντας την αντίστοιχη φόρμα που εμφανίζεται δεξιά του πίνακα, μετά το πάτημα του κουμπιού “Create Application”.

Μετά την συμπλήρωση της φόρμας, ο διαχειριστής του συστήματος πατάει “Apply” και το σύστημα του εμφανίζει αναδυόμενο παράθυρο, όμοιο με αυτό που παρουσιάζεται στην ενότητα 5.5, για να προχωρήσει η αίτηση. Πατώντας το κουμπί “Yes” η αίτηση καταχωρείται.

Είναι σημαντικό να αναφερθεί ότι στην φόρμα δημιουργίας αίτησης, ο επιλογέας φοιτητών περιέχει όλους τους φοιτητές του τμήματος σε έναν πίνακα που εμφανίζεται μέσα σε

αναδυόμενο παράθυρο. Η πλοήγηση στον πίνακα αυτό είναι εύκολη και γρήγορη καθώς είναι σελιδοποιημένος και στο πάνω μέρος του περιέχει μπάρα αναζήτησης φοιτητή.

Παράλληλα ο επιλογέας απουσίας στην φόρμα δημιουργίας αίτησης, αφορά τις αδικαιολόγητες απουσίες του επιλεγμένου φοιτητή τις τελευταίες 48 ώρες.

Τα σενάρια κατα τα οποία η δημιουργία αίτησης δικαιολόγησης απουσίας είναι ανέφικτη, είναι όμοια με εκείνα που αναλύονται στην ενότητα 5.5 στην παράγραφο που αφορά την αίτηση δικαιολόγησης απουσίας απο τον φοιτητή.

Το σενάριο της λειτουργίας που αναλύθηκε παραπάνω, συμβαίνει σε πολύ συγκεκριμένες περιπτώσεις που κάποιος φοιτητής αδυνατεί να αιτηθεί δικαιολόγηση για κάποια απουσία του.

Για παράδειγμα αν ο φοιτητής έχει ξεχάσει τον κωδικό πρόσβασής εισόδου του στο σύστημα και βιάζεται να προλάβει να δικαιολογήσει μια απουσία του, μπορεί να ενημερώσει μέσω email τον διαχειριστή του συστήματος και εκείνος με την σειρά του να παρέμβει και να δημιουργήσει μια αίτηση δικαιολόγησης για κάποια απουσία του.

Τα υπόλοιπα σενάρια χρήσης του διαχειριστή συστήματος που αφορούν τις αιτήσεις δικαιολόγησης απουσιών είναι πανομοιότυπα με αυτά που αφορούν τον ρόλο του γραμματέα και παρατίθενται αναλυτικά στην **ενότητα 5.6**.

Δημιουργία χρηστών

The screenshot displays the 'Users' management interface. On the left, there is a sidebar with filters for 'Department' (set to 'All Departments') and 'Role' (set to 'Teacher'). Below these filters are buttons for 'Add User' and 'Upload Students'. A search bar labeled 'Search users' is present. The main area shows a table of users with columns for ID, Username, and Roles. The table contains three rows of data. To the right, an 'Add User' modal form is open, showing fields for First name, Last name, Username, Email, Password, Roles, and Department. The form is pre-filled with 'Thomas Kamalakis', 'kamelakis', 'kamelakis@hua.gr', and 'Teacher'.

ID	Username	Roles
1	tsadimas	Admin, Teacher
2	tserpes	Teacher
17	varlamis	Teacher

Items per page: 3 1 - 3 of 16

Add User

First name: Thomas Last name: Kamalakis

Please enter first name. Please enter lastname.

Username: kamalakis Email: kamalakis@hua.gr

Please enter username. Please enter email.

Password: Password field with strength indicator (12 / 18)

Please enter password. Choose Role(s): Teacher

Department: Informatics & Telematics

Choose Department

☒ Active ☐ Inactive

Save Cancel

Εικ.49. Δημιουργία χρηστών

Ο διαχειριστής του συστήματος μπορεί να προσθέσει έναν χρήστη οποιουδήποτε ρόλου εκτός από "Admin" σε ένα τμήμα, συμπληρώνοντας την αντίστοιχη φόρμα που εμφανίζεται δεξιά του πίνακα, μετά το πάτημα του κουμπιού "Add User".

Ανέβασμα αρχείου φοιτητών

The screenshot displays the 'Users' management interface. On the left, there are filters for 'Department' (set to '-- All Departments --') and 'Role' (set to '-- All Roles --'). Below these are buttons for 'Add User' and 'Upload Students'. A search bar labeled 'Search users' is also present. The main area contains a table with the following data:

ID ↑	Username	Roles
1	tsadimas	Teacher, Admin
2	tserpes	Teacher
3	it21526	Student

At the bottom of the table, it shows 'Items per page: 3' and '1 - 3 of 31'. On the right, the 'Upload Students' modal is open, showing a file selection area with 'students.xlsx' selected and a progress bar at 100%. The modal includes 'Upload' and 'Cancel' buttons.

Εικ.50. Ανέβασμα αρχείου φοιτητών

Ο διαχειριστής του συστήματος μπορεί να προσθέσει τους φοιτητές ενός τμήματος, πατώντας το κουμπί “Upload Students” και ανεβάζοντας ένα αρχείο csv ή excel με τους φοιτητές στην καρτέλα που εμφανίζεται δεξιά του πίνακα. Το αρχείο αυτό περιέχει όλες τις πληροφορίες που είναι απαραίτητες όπως και το σε ποιο τμήμα ανήκουν οι φοιτητές.

Ωστόσο απο το σενάριο “Ανέβασμα αρχείου φοιτητών”, προκύπτουν τρία υποσενάρια:

- Το αρχείο που ανέβηκε δεν είναι του σωστού τύπου και το σύστημα εμφανίζει ενημερωτικό μήνυμα όπως φαίνεται στην παρακάτω εικόνα.

Users

Department
-- All Departments --
Choose Department

Role
-- All Roles --
Choose Role

Add User **Upload Students**

Search users

ID ↑	Username	Roles
1	tsadimas	Admin, Teacher
2	tserpes	Teacher
3	it21526	Student

Items per page: 3 1 - 3 of 31

Upload Students

Choose csv/excel file:

Choose File dummy_data.txt

0%

Upload **Cancel**

File type text/plain is not the corresponding **Close**

Εικ.51. Λάθος τύπος αρχείου csv/excel

- Το αρχείο που ανέβηκε είναι του σωστού τύπου αλλά ο τρόπος δόμησής του δεν είναι συμβατός. Το σύστημα ενημερώνει τον διαχειριστή του συστήματος με κατάλληλο μήνυμα. Στην παρακάτω εικόνα φαίνεται η περίπτωση ενός ασύμβατου αρχείου csv.

Users

Department

-- All Departments --

Choose Department

Role

-- All Roles --

Choose Role

Add User

Upload Students

Search users

ID ↑	Username	Roles
1	tsadimas	Admin, Teacher
2	tserpes	Teacher
3	it21526	Student

Items per page: 3 1 – 3 of 31

Upload Students

Choose csv/excel file:

Choose File

course_sc...tudents.csv

0%

Upload

Cancel

The content of the csv file is inappropriate Close

Εικ.52. Ακατάλληλο περιεχόμενο αρχείου csv/excel

Αν μάλιστα το ανεβασμένο αρχείο είναι και τύπου excel, το μήνυμα ενημερώνει τον διαχειριστή του συστήματος και σε ποιά σημείο του αρχείου εντοπίζονται τα προβλήματα και ποια είναι αυτά. Μια τέτοια περίπτωση σφάλματος φαίνεται στην παρακάτω εικόνα.

Users

Department

-- All Departments --

Choose Department

Role

-- All Roles --

Choose Role

Add User

Upload Students

Search users

Q

ID ↑	Username	Roles
1	tsadimas	Admin, Teacher
2	tserpes	Teacher
3	it21526	Student

Items per page: 3

1 - 3 of 31

⏪

Upload Students

Choose csv/excel file:

Choose File

course_sc...tudents.xlsx

0%

Upload

Cancel

Error in row 2: username: Username required, lastname: Last name required, firstname: First name required, email: Email required, password: Password required, departmentName: Department required

Close

Εικ.53. Ενημέρωση για τα σημεία των σφαλμάτων του αρχείου excel

- Το ανεβασμένο αρχείο είναι σωστού τύπου και συμβατό αλλά περιέχει κάποιο σφάλμα σε μια εγγραφή του. Το σύστημα ενημερώνει τον διαχειριστή για το πρώτο σφάλμα που συναντά στο αρχείο κάθε φορά (ξεκινώντας από την πρώτη γραμμή και το πρώτο κελί του και προσπελαύνοντας δεξιά και προς τα κάτω). Μια τέτοια περίπτωση σφάλματος φαίνεται στην παρακάτω εικόνα.

Users

Department

-- All Departments --

Choose Department

Role

-- All Roles --

Choose Role

Add User

Upload Students

Search users

Q

ID ↑	Username	Roles
1	tsadimas	Teacher, Admin
2	tserpes	Teacher
3	it21526	Student

Items per page: 3

1 – 3 of 31

<<

<

>

>>

Upload Students

Choose csv/excel file:

Choose File

students.xlsx

0%

Upload

Cancel

Error in row 2: am: AM must be between 5 and 8 integers

Close

Εικ.54. Ενημέρωση διαχειριστή για το πρώτο σφάλμα του αρχείου csv/excel

- Το αρχείο που ανέβηκε περιέχει ήδη καταχωρημένες εγγραφές. Το σύστημα ενημερώνει τον διαχειριστή για την πρώτη εγγραφή που συναντά κάθε φορά στο αρχείο και υπάρχει ήδη στο σύστημα.

Users

Department

-- All Departments --

Choose Department

Role

-- All Roles --

Choose Role

Add User

Upload Students

Search users

ID ↑	Username	Roles
1	tsadimas	Teacher, Admin
2	tserpis	Teacher
3	it21526	Student

Items per page: 31 – 3 of 31

Upload Students

Choose csv/excel file:

Choose Filestudents.xlsx

0%

Upload

Cancel

Error in row 2: AM 21526 already exists!Close

Εικ.55. Ενημέρωση διαχειριστή για ήδη υπάρχουσα εγγραφή του αρχείου csv/excel

Προβολή στοιχείων χρήστη

Users

Department: Informatics & Telematics
Choose Department

Role: Teacher
Choose Role

[Add User](#) [Upload Students](#)

Search users

ID ↑	Username	Roles
1	tsadimas	Teacher, Admin
2	tserpes	Teacher
17	varlamis	Teacher

Items per page: 3 1 – 3 of 16

User Details

First name: Konstantinos
Last name: Tserpes
Username: tserpes
Email: tserpes@hua.gr
Roles: Teacher
Department: Informatics & Telematics
Status: Inactive

[Edit](#)

[Close](#)

Εικ.56. Προβολή στοιχείων χρήστη

Ο διαχειριστής του συστήματος, μπορεί να επιλέξει τους χρήστες ανα τμήμα ή ρόλο. Οι ρόλοι των χρηστών μπορεί να είναι: Admin, Teacher, Secretary, Student.

Παράλληλα μπορεί να αναζητήσει χρήστες μέσω της μπάρας αναζήτησης στο πάνω μέρος του πίνακα με κριτήριο το “Username”.

Επειτα πλοηγούμενος στον πίνακα με τους χρήστες, μπορεί να επιλέξει έναν χρήστη πατώντας την γραμμή του πίνακα στην οποία αντιστοιχεί. Έτσι έχει την δυνατότητα να δει περισσότερες λεπτομέρειες σχετικά με τον χρήστη, στην καρτέλα που εμφανίζεται δεξιά του πίνακα.

Επεξεργασία στοιχείων χρήστη

The image shows a web interface for managing users. On the left, there's a 'Users' section with filters for Department (set to 'All Departments') and Role (set to 'Teacher'). Below these are buttons for 'Add User' and 'Upload Students', and a search bar. A table lists three users: ID 1 (tsadimas, Teacher, Admin), ID 2 (tserpes, Teacher), and ID 17 (varlamis, Teacher). On the right, an 'Edit User' modal is open for user 'tserpes'. It contains fields for First name (Konstantinos), Last name (Tserpes), Username (tserpes), Email (tserpes@hua.gr), Password (0/10), Roles (Teacher), and Department (Informatics & Telematics). There are also radio buttons for 'Active' (selected) and 'Inactive', and 'Update' and 'Cancel' buttons at the bottom.

ID	Username	Roles
1	tsadimas	Teacher, Admin
2	tserpes	Teacher
17	varlamis	Teacher

Εικ.57. Επεξεργασία στοιχείων χρήστη

Στην καρτέλα προβολής των στοιχείων του χρήστη, ο διαχειριστής του συστήματος μπορεί να επιλέξει το κουμπί “Edit” και να επεξεργαστεί τα στοιχεία του χρήστη στην αντίστοιχη φόρμα που εμφανίζεται από το σύστημα. Το πεδίο της φόρμας επεξεργασίας του χρήστη που αφορά το password, δεν απαιτεί υποχρεωτική συμπλήρωση.

Στο σενάριο επεξεργασίας των στοιχείων του χρήστη, είναι σημαντικό να τονιστούν δύο περιπτώσεις:

- Ο χρήστης που τίθεται προς επεξεργασία, μπορεί να είναι ένας πρόσφατα εγγεγραμμένος καθηγητής. Στην περίπτωση αυτή ο διαχειριστής του συστήματος προκειμένου να ενεργοποιήσει τον λογαριασμό του καθηγητή, πρέπει να τον κατατάξει στο αντίστοιχο τμήμα και να επιλέξει την ένδειξη “Active”.
- Εάν ο χρήστης που τίθεται προς επεξεργασία είναι φοιτητής, τότε δεν υπάρχει επιλογή για αλλαγή τμήματος στην φόρμα επεξεργασίας, όπως φαίνεται στην παρακάτω εικόνα.

Users

Department

Informatics & Telematics

Choose Department

Role

Student

Choose Role

Add User

Upload Students

Search users

ID ↑	Username	Roles
3	it21526	Student
4	it21569	Student
5	it21628	Student

Items per page: 3

1 – 3 of 14

<

>

Edit Student

First name

Ioannis

Last name

Makris

Please enter first name.

Username

it21526

Please enter lastname.

Email

it21526@hua.gr

Please enter username.

Password

21526

Please enter email.

AM

21526

Please enter password.

0 / 10

Please enter AM.

☒ Active
 ☐ Inactive

Update

Cancel

Εικ.58. Φόρμα επεξεργασίας φοιτητή

5.4. Λειτουργίες Καθηγητή (Teacher)

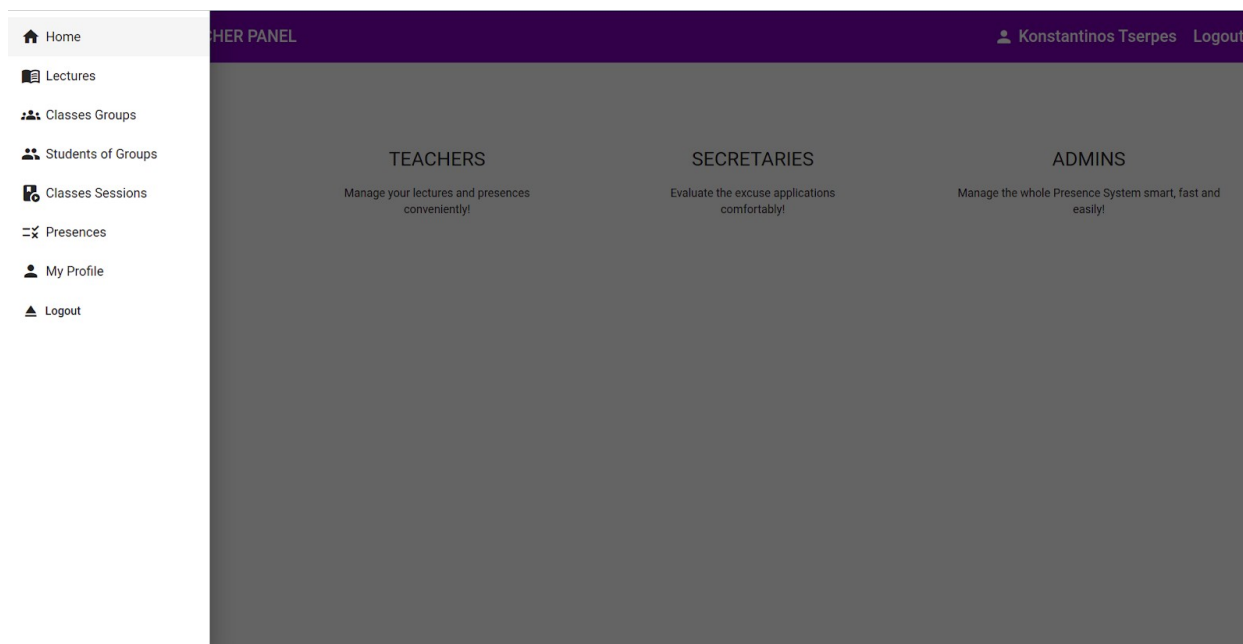
Στην ενότητα αυτή παρουσιάζονται τα σενάρια χρήσης για τις λειτουργίες που επιτελεί ο ρόλος του καθηγητή. Επειδή στο σύστημα διαχείρισης παρουσιών οι λειτουργίες αυτές επιτελούνται απο τον καθηγητή αλλά και απο τον διαχειριστή του συστήματος, δεν συμπεριλήφθηκαν στην προηγούμενη ενότητα. Ετσι για πρακτικούς λόγους παρουσιάζονται σε αυτήν.

Η βασική διαφορά σε σχέση με τον διαχειριστή συστήματος είναι ότι η δράση του καθηγητή περιορίζεται στο τμήμα στο οποίο ανήκει.

Επίσης όλες οι λειτουργίες που αφορούν τον καθηγητή και σχετίζονται άμεσα ή έμμεσα με τα προγράμματα μαθημάτων, προϋποθέτουν ο καθηγητής να ανήκει στο διδακτικό προσωπικό τους.

Πλοήγηση στο μενού επιλογών

Ο καθηγητής αφού συνδεθεί στο σύστημα, μπορεί να δει το μενού επιλογών του και να πλοηγηθεί σε αυτό, πατώντας το κουμπί που αναπαρίσταται με τρεις οριζόντιες παύλες πάνω αριστερά στην οθόνη.



Εικ.59. Μενού Καθηγητή

Δημιουργία διάλεξης

Lectures

Selected: Java I

Choose Course Schedule

☐ Theories ☒ Labs

Add Lecture

Search lectures

ID	Name Id ↑	Title
20	lab_1	Hello World in Java
21	lab_2	Create your first Class
22	lab_3	Introduction to clock demo

Items per page: 3 1 – 3 of 3

Add Lab Lecture

Identifier
lab_4

Choose Identifier

Title
Abstract Classes

Set Title.

Save Cancel

Εικ.60. Δημιουργία διάλεξης

Ο καθηγητής πρέπει πρώτα να επιλέξει το πρόγραμμα μαθήματος για το οποίο θέλει να δημιουργήσει την διάλεξη, καθώς και τον τύπο της διάλεξης. Ο τύπος της διάλεξης μπορεί να είναι θεωρητικός ή εργαστηριακός. Πατώντας το κουμπί “Add Lecture” μπορεί να προσθέσει μια διάλεξη συμπληρώνοντας την αντίστοιχη φόρμα που εμφανίζεται δεξιά του πίνακα.

Προβολή στοιχείων διάλεξης

The screenshot displays a web application interface for managing lectures. On the left, a sidebar titled 'Lectures' contains a search bar with 'Selected: Java I', radio buttons for 'Theories' and 'Labs' (with 'Labs' selected), an 'Add Lecture' button, and a search input field. Below this is a table of lectures. On the right, a 'Lecture Details' modal is open, showing details for the selected lecture.

ID	Name Id ↑	Title
20	lab_1	Hello World in Java
21	lab_2	Create your first Class
22	lab_3	Introduction to clock demo
59	lab_4	Abstract Classes

The 'Lecture Details' modal shows:

- Name:** Java I, lab_4
- Title:** Abstract Classes
- Buttons: Edit, Delete, Cancel

Εικ.61. Προβολή στοιχείων διάλεξης

Ο καθηγητής μπορεί να δει τις διαλέξεις αφού επιλέξει πρώτα πρόγραμμα μαθήματος και τύπο διάλεξης. Παράλληλα μπορεί να αναζητήσει διαλέξεις μέσω της μπάρας αναζήτησης στο πάνω μέρος του πίνακα. Η αναζήτηση μπορεί να γίνει με κριτήρια το “Name Id” ή το “Title” της διάλεξης.

Επειτα πλοηγούμενος στον πίνακα με τις διαλέξεις, μπορεί να επιλέξει μια διάλεξη πατώντας την γραμμή του πίνακα στην οποία αντιστοιχεί. Έτσι έχει την δυνατότητα να επεξεργαστεί ή να διαγράψει την διάλεξη πατώντας τα αντίστοιχα κουμπιά “Edit” ή “Delete” στην καρτέλα που εμφανίζεται δεξιά του πίνακα.

Επεξεργασία στοιχείων διάλεξης

The screenshot displays a web application for managing lectures. On the left, a sidebar titled 'Lectures' contains a search bar with 'Selected: Java I', radio buttons for 'Theories' (selected) and 'Labs', an 'Add Lecture' button, and a search input field. Below this is a table of lectures with columns 'ID', 'Name Id', and 'Title'. The table lists three items: 'theory_1' (What is o.o.p ?), 'theory_2' (Classes & Objects), and 'theory_3' (Polymorphism). At the bottom of the table, there is a pagination bar showing 'Items per page: 5' and '1 - 3 of 3'. On the right, a modal form titled 'Edit theory_3' is open. It contains a dropdown for 'Identifier' (set to 'theory_3'), a text input for 'Title' (set to 'Polymorphism'), and 'Update' and 'Cancel' buttons.

ID	Name Id	Title
39	theory_1	What is o.o.p ?
18	theory_2	Classes & Objects
19	theory_3	Polymorphism

Εικ.62. Επεξεργασία στοιχείων διάλεξης

Στην καρτέλα προβολής των στοιχείων της διάλεξης, ο καθηγητής μπορεί να επιλέξει το κουμπί “Edit” και να επεξεργαστεί τα στοιχεία της στην αντίστοιχη φόρμα που εμφανίζεται απο το σύστημα. Επίσης μετα την επεξεργασία των στοιχείων της διάλεξης, το σύστημα ελέγχει για το αν η διάλεξη αντιστοιχεί με κάποια συνεδρία διάλεξης.

Σε περίπτωση που η διάλεξη σχετίζεται με συνεδρία διάλεξης η επεξεργασία των στοιχείων της είναι αδύνατη και το σύστημα εμφανίζει κατάλληλο μήνυμα, όπως φαίνεται στην παρακάτω εικόνα.

Lectures

Selected: Java I

Choose Course Schedule

☒ Theories ☐ Labs

Add Lecture

Search lectures

ID	Name Id ↑	Title
39	theory_1	What is o.o.p ?
18	theory_2	Classes & Objects
60	theory_3	Create a clock with Java

Items per page: 3 1 - 3 of 3

Edit theory_1

Identifier
theory_1

Choose Identifier

Title
What is o.o.p ??

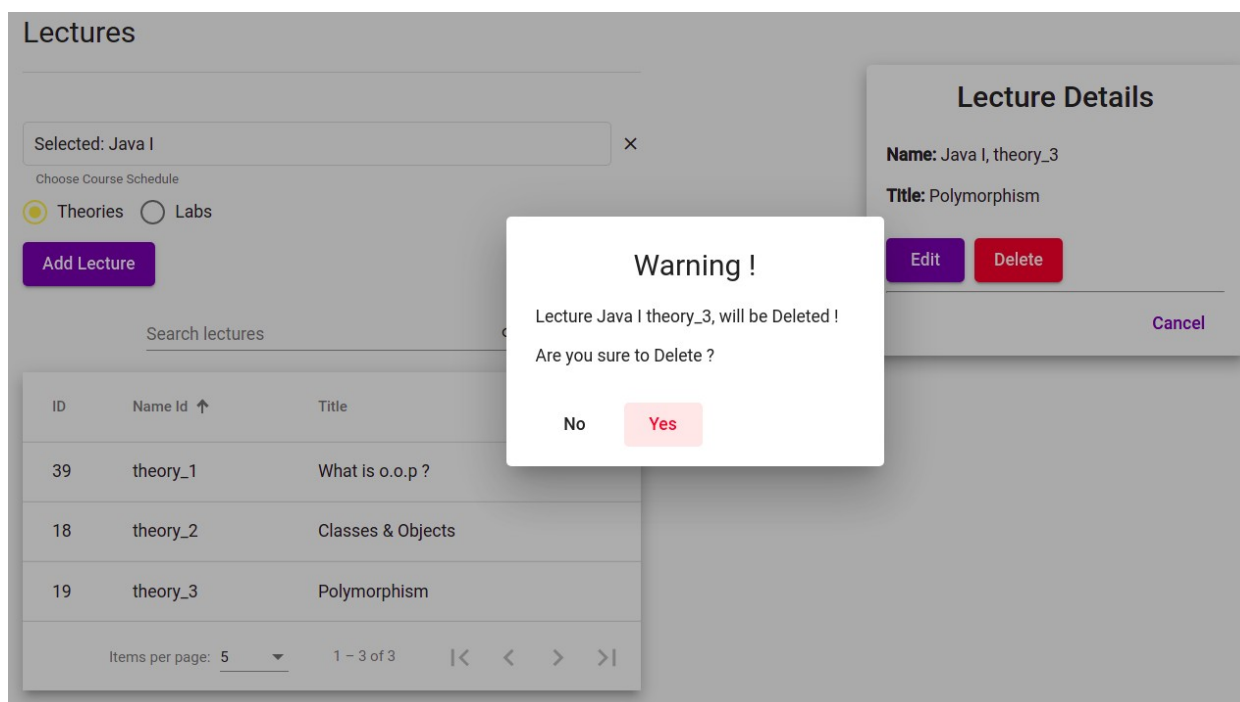
Set Title.

Update Cancel

You cannot update the lecture since it's already a part of a class session

Εικ.63. Αδυναμία επεξεργασίας στοιχείων διάλεξης

Διαγραφή διάλεξης



Εικ.64. Διαγραφή διάλεξης

Στην καρτέλα προβολής των στοιχείων της διάλεξης, ο καθηγητής μπορεί να επιλέξει το κουμπί “Delete” για να διαγράψει την επιλεγμένη διάλεξη. Το σύστημα θα εμφανίσει ένα αναδυόμενο παράθυρο το οποίο ρωτάει τον καθηγητή, αν είναι βέβαιος για την διαγραφή ή όχι, οπότε αν θέλει να διαγράψει την επιλεγμένη διάλεξη πατάει το κουμπί “Yes”.

Για να γίνει η διαγραφή μιας διάλεξης θα πρέπει να μην έχει δημιουργηθεί ακόμη κάποια συνεδρία διάλεξης που να αντιστοιχεί σε αυτήν, διαφορετικά εμφανίζεται στον καθηγητή κατάλληλο μήνυμα όπως φαίνεται στην παρακάτω εικόνα.

Lectures

Selected: Java I

Choose Course Schedule

☒ Theories
☐ Labs

Add Lecture

Search lectures

ID	Name Id ↑	Title
39	theory_1	What is o.o.p ?
18	theory_2	Classes & Objects
60	theory_3	Create a clock with Java

Items per page: 5
1 - 3 of 3

Lecture Details

Name: Java I, theory_1

Title: What is o.o.p ?

Edit
Delete

Cancel

You cannot delete theory_1 lecture of Java I schedule, since it is a part of a class session
Close

Εικ.65. Αδυναμία διαγραφής διάλεξης

Δημιουργία ομάδας για φοιτητές

Classes Groups

Selected: Parallel Computing and Algorithms

Choose Course Schedule

☐ Theories
☒ Labs

Add Class Group

Search groups

ID	Name ↑	Time	Room	Status
----	--------	------	------	--------

Items per page: 3
0 of 0

Add Lab Group

Id
1

Capacity
20

Set id.

Set Capacity.

Start Time
12:00

End Time
14:00

Set Start Time.

Room
2.2

Choose Room

Subscriptions:
☐ Opened
☒ Closed

Save

Cancel

Εικ.66. Δημιουργία ομάδας για φοιτητές

Ο καθηγητής πρέπει πρώτα να επιλέξει το πρόγραμμα μαθήματος για το οποίο θέλει να δημιουργήσει την ομάδα, καθώς και τον τύπο της ομάδας. Ο τύπος της ομάδας μπορεί να είναι θεωρητικός ή εργαστηριακός, ανάλογα με τον τύπο της διάλεξης (θεωρητική ή εργαστηριακή) για την παρακολούθηση της οποίας προορίζεται.

Πατώντας το κουμπί “Add Class Group” μπορεί να προσθέσει μια ομάδα συμπληρώνοντας την αντίστοιχη φόρμα που εμφανίζεται δεξιά του πίνακα.

Προβολή στοιχείων ομάδας για φοιτητές

The screenshot displays the 'Classes Groups' management interface. At the top, a dropdown menu shows 'Selected: Parallel Computing and Algorithms'. Below it, there are radio buttons for 'Theories' and 'Labs', with 'Labs' selected. A purple 'Add Class Group' button is present. A search bar labeled 'Search groups' is also visible. The main part of the interface is a table with the following data:

ID	Name ↑	Time	Room	Status
22	group_1	12:00 - 14:00	2.2	Closed

Below the table, there is a pagination control showing 'Items per page: 3' and '1 - 1 of 1'. To the right of the table, a 'Class Group Details' modal is open, showing the following information:

- Category:** Labs
- Course Schedule:** Parallel Computing and Algorithms
- Name:** group_1
- Time:** 12:00 - 14:00
- Room:** 2.2
- Capacity:** 20/20
- Status:** Closed

The modal includes 'Edit' and 'Delete' buttons, and a 'Cancel' button at the bottom right.

Εικ.67. Προβολή στοιχείων ομάδας για φοιτητές

Ο καθηγητής μπορεί να δει τις ομάδες αφού επιλέξει πρώτα πρόγραμμα μαθήματος και τύπο. Παράλληλα μπορεί να αναζητήσει ομάδες μέσω της μπάρας αναζήτησης στο πάνω μέρος του πίνακα. Η αναζήτηση μπορεί να γίνει με κριτήρια το “Name”, το “Time” ή το Room της ομάδας.

Επειτα πλοηγούμενος στον πίνακα με τις ομάδες, μπορεί να επιλέξει μια ομάδα πατώντας την γραμμή του πίνακα στην οποία αντιστοιχεί. Έτσι έχει την δυνατότητα να δει περισσότερες λεπτομέρειες σχετικά με την ομάδα, στην καρτέλα που εμφανίζεται δεξιά του πίνακα.

Επεξεργασία στοιχείων ομάδας για φοιτητές

The screenshot displays the 'Classes Groups' management interface. On the left, a sidebar shows the selected course 'Parallel Computing and Algorithms' and options for 'Theories' and 'Labs'. A table lists the groups, with 'group_1' selected. On the right, the 'Edit lab_group_1' modal is open, showing fields for ID, Capacity, Start Time, End Time, Room, and Subscriptions.

Classes Groups

Selected: Parallel Computing and Algorithms

Choose Course Schedule

☐ Theories ☒ Labs

Add Class Group

Search groups

ID	Name ↑	Time	Room	Status
22	group_1	12:00 - 14:00	2.2	Closed

Items per page: 3 1 - 1 of 1

Edit lab_group_1

id: 1 Capacity: 20

Set id. Set Capacity.

Start Time: 12:00 End Time: 14:00

Set Start Time.

Room: 2.2

Choose Room

Subscriptions: ☒ Opened ☐ Closed

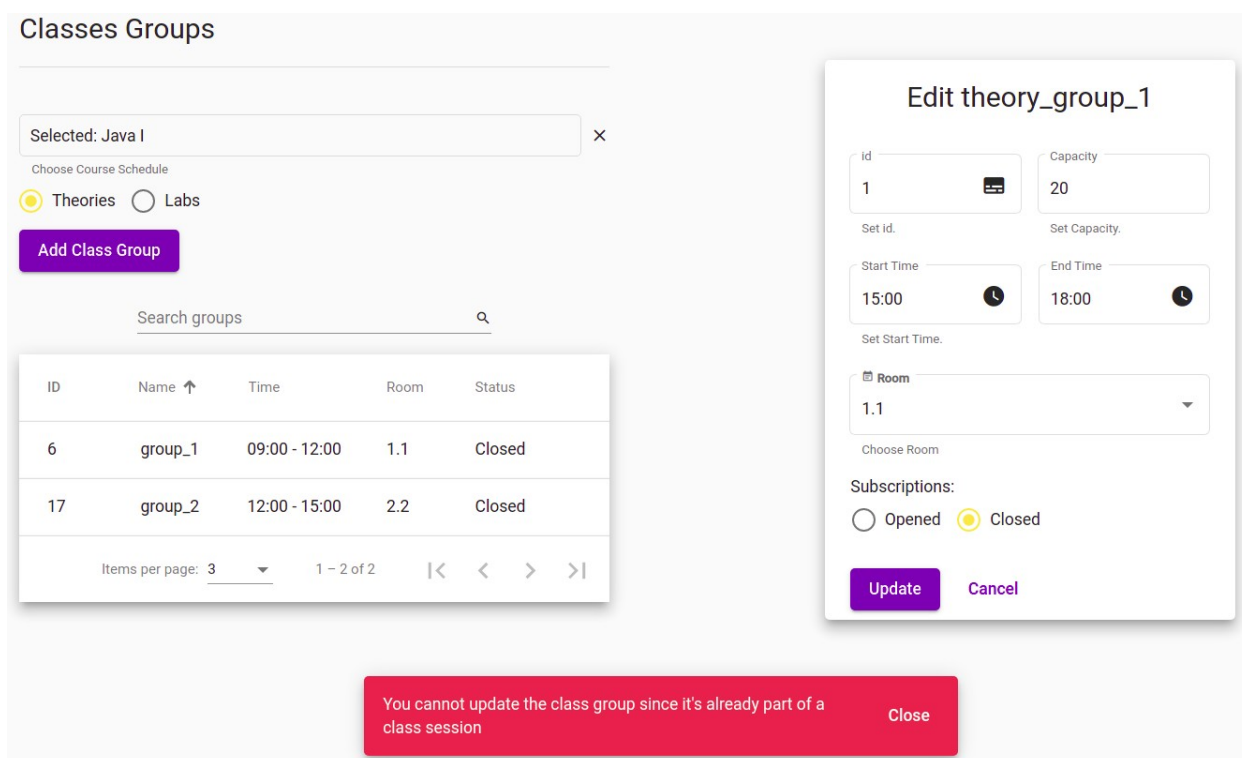
Update **Cancel**

Εικ.68. Επεξεργασία στοιχείων ομάδας για φοιτητές

Στην καρτέλα προβολής των στοιχείων της ομάδας, ο καθηγητής μπορεί να επιλέξει το κουμπί “Edit” και να επεξεργαστεί τα στοιχεία στην αντίστοιχη φόρμα που εμφανίζεται απο το σύστημα.

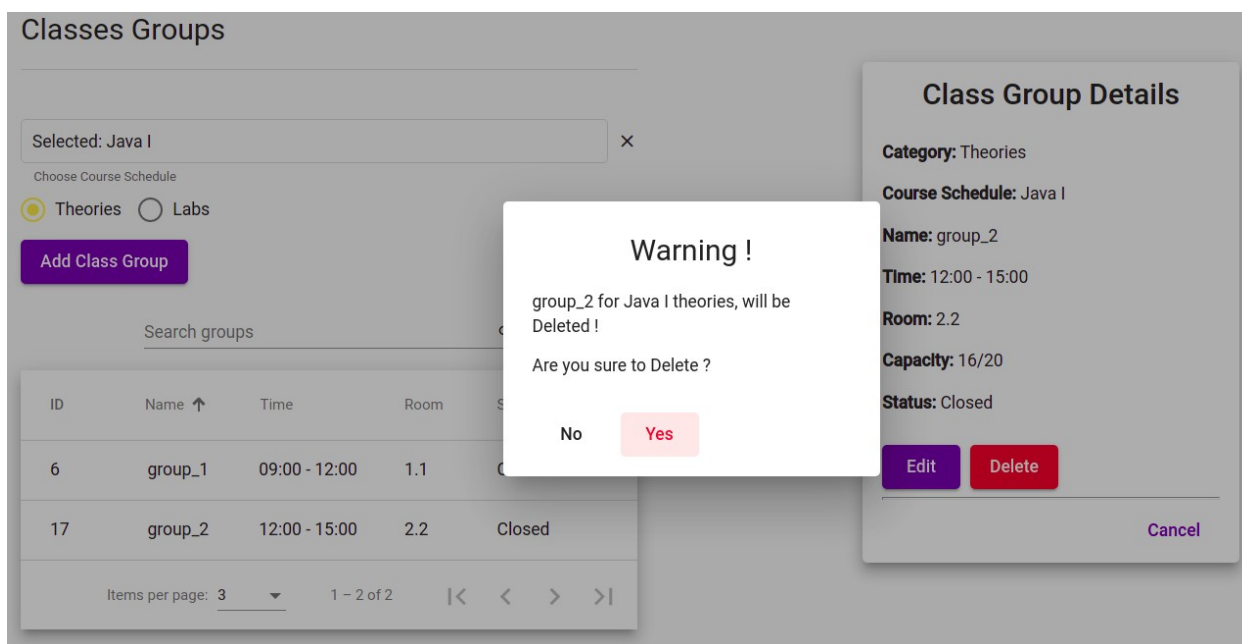
Μέσα στην διαδικασία της επεξεργασίας των στοιχείων της ομάδας, συμπεριλαμβάνεται και το “άνοιγμα” ή το “κλείσιμο” των εγγραφών που μπορούν να πραγματοποιήσουν οι φοιτητές στην ομάδα. Επίσης μετα την επεξεργασία των στοιχείων της ομάδας, το σύστημα ελέγχει αν η ομάδα σχετίζεται με κάποια συνεδρία διάλεξης.

Σε περίπτωση που η ομάδα σχετίζεται με συνεδρία διάλεξης η επεξεργασία των στοιχείων της ομάδας είναι αδύνατη και το σύστημα εμφανίζει κατάλληλο μήνυμα, όπως φαίνεται στην παρακάτω εικόνα.



Εικ.69. Αδυναμία επεξεργασίας ομάδας φοιτητών

Διαγραφή ομάδας για φοιτητές



Εικ.70. Διαγραφή ομάδας για φοιτητές

Στην καρτέλα προβολής των στοιχείων της ομάδας, ο καθηγητής μπορεί να επιλέξει το κουμπί “Delete” για να διαγράψει την επιλεγμένη ομάδα. Το σύστημα θα εμφανίσει ένα αναδυόμενο παράθυρο το οποίο ρωτάει τον καθηγητή, αν είναι βέβαιος για την διαγραφή ή όχι, οπότε αν θέλει να διαγράψει την επιλεγμένη ομάδα πατάει το κουμπί “Yes”.

Για να γίνει η διαγραφή μιας ομάδας θα πρέπει αυτή να μην σχετίζεται με κάποια συνεδρία διάλεξης, διαφορετικά εμφανίζεται στον καθηγητή κατάλληλο μήνυμα όπως φαίνεται στην παρακάτω εικόνα.

Classes Groups

Selected: Java I ×

Choose Course Schedule

☒ Theories ☐ Labs

Add Class Group

Search groups 🔍

ID	Name ↑	Time	Room	Status
6	group_1	09:00 - 12:00	1.1	Closed
17	group_2	12:00 - 15:00	2.2	Closed

Items per page: 3 1 – 2 of 2 |< < > >|

Class Group Details

Category: Theories

Course Schedule: Java I

Name: group_1

Time: 09:00 - 12:00

Room: 1.1

Capacity: 10/20

Status: Closed

Edit **Delete**

Cancel

You cannot delete theory_group_1 of Java I schedule, since it is a part of a class session **Close**

Εικ.71. Αδυναμία διαγραφής ομάδας σχετιζόμενης με συνεδρία διάλεξης

Ομως ακόμα κι αν μια ομάδα δεν σχετίζεται ακόμα με κάποια συνεδρία διάλεξης, αλλά έχει εγγεγραμμένους φοιτητές, τότε και πάλι η διαγραφή της είναι αδύνατη. Στην παρακάτω εικόνα απεικονίζεται μια τέτοια περίπτωση.

Classes Groups

Selected: Java I

Choose Course Schedule

☒ Theories ☐ Labs

Add Class Group

Search groups

ID	Name ↑	Time	Room	Status
6	group_1	09:00 - 12:00	1.1	Closed
17	group_2	12:00 - 15:00	2.2	Closed

Items per page: 3 1 - 2 of 2

Class Group Details

Category: Theories
Course Schedule: Java I
Name: group_2
Time: 12:00 - 15:00
Room: 2.2
Capacity: 16/20
Status: Closed

Edit

Delete

Cancel

You cannot delete theory_group_2 of Java I schedule, since it has student subscriptions

Close

Εικ.72. Αδυναμία διαγραφής ομάδας με εγγεγραμμένους φοιτητές

ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΑΡΟΥΣΙΩΝ – ΗΛΕΚΤΡΟΝΙΚΟ ΠΑΡΟΥΣΙΟΛΟΓΙΟ / ΜΑΚΡΗΣ ΙΩΑΝΝΗΣ

218

Προσθήκη φοιτητή στην ομάδα

The screenshot displays the 'Students of Groups' interface. On the left, there are two dropdown menus: 'Selected: Parallel Computing and Algorithms' and 'Selected: group_1'. Below these are radio buttons for 'Theory Groups' (selected) and 'Lab Groups'. A purple button labeled 'Add Student to Group' is visible. On the right, a modal window titled 'Add Student to Group' is open, showing 'Selected: it21788' and 'Choose Student' text. At the bottom of the modal are 'Save' and 'Cancel' buttons. Below the modal, there is a search bar labeled 'Search students' and a table of students.

ID	Username	First Name	Last Name
3	it21526	Ioannis	Makris
4	it21569	Giorgos	Loizidis
5	it21628	Vasilis	Matzaris

At the bottom of the table, there is a pagination bar showing 'Items per page: 3' and '1 - 3 of 5'.

Εικ.73. Προσθήκη φοιτητή στην ομάδα

Ο καθηγητής πρέπει πρώτα να επιλέξει το πρόγραμμα μαθήματος και τον τύπο της ομάδας (ομάδα θεωρίας ή ομάδα εργαστηρίου). Επειτα του δίνεται η δυνατότητα επιλογής μιας από τις ομάδες φοιτητών που αντιστοιχούν στο πρόγραμμα μαθήματος και τον τύπο ομάδας που επέλεξε.

Πατώντας το κουμπί “Add Student to Group” μπορεί να προσθέσει έναν φοιτητή στην ομάδα επιλέγοντάς τον από τον επιλογέα της αντίστοιχης φόρμας που εμφανίζεται δεξιά του πίνακα. Είναι χρήσιμο να αναφερθεί ότι στον επιλογέα της φόρμας προσθήκης φοιτητή στην ομάδα, οι φοιτητές που εμφανίζονται προς επιλογή, είναι φοιτητές που ανήκουν στο επιλεγμένο πρόγραμμα μαθήματος αλλά δεν έχουν προστεθεί ακόμη στην ομάδα.

Επίσης ο καθηγητής μπορεί να προσθέσει φοιτητές στην ομάδα, ακόμα κι αν έχει απενεργοποιήσει τις εγγραφές για εκείνους. Για να γίνει η προσθήκη ενός φοιτητή σε μια ομάδα, θα πρέπει η ομάδα να μην σχετίζεται με κάποια συνεδρία διάλεξης, διαφορετικά είναι

αδύνατη. Σε περίπτωση που η ομάδα προσθήκης φοιτητή σχετίζεται με συνεδρία διάλεξης, εμφανίζεται στον καθηγητή κατάλληλο μήνυμα όπως φαίνεται στην παρακάτω εικόνα.

The screenshot shows the 'Students of Groups' interface. At the top, there are two selection fields: 'Selected: Java I' (with a 'Choose Course Schedule' link) and 'Selected: group_2' (with a 'Choose Class Group' link). Below these are radio buttons for 'Theory Groups' and 'Lab Groups'. A purple 'Add Student to Group' button is present. A search bar labeled 'Search students' is above a table of students. The table has columns for ID, Username, First Name, and Last Name, and contains three rows of student data. A modal dialog titled 'Add Student to Group' is open, showing 'Selected: it21752' and 'Choose Student' link, with 'Save' and 'Cancel' buttons. A red error message box is overlaid on the bottom right of the interface, stating: 'You cannot subscribe a student to a group that already exists on a session'. The error box has a 'Close' button.

ID	Username	First Name	Last Name
3	it21526	Ioannis	Makris
4	it21569	Giorgos	Loizidis
5	it21628	Vasilis	Matzaris

Εικ.74. Αδυναμία προσθήκης φοιτητή σε ομάδα που σχετίζεται με συνεδρία διάλεξης

Επίσης προκειμένου ο καθηγητής να μπορεί να εγγράψει έναν φοιτητή σε μια ομάδα, θα πρέπει η ομάδα αυτή να έχει κάποια κενή θέση. Διαφορετικά επίσης η εγγραφή είναι αδύνατη και το σύστημα ενημερώνει τον καθηγητή με κατάλληλο μήνυμα.

Students of Groups

Selected: Web Development

Choose Course Schedule

☐ Theory Groups ☒ Lab Groups

Selected: group_1

Choose Class Group

Add Student to Group

Search students

ID	Username	First Name	Last Name
3	it21526	Ioannis	Makris
4	it21569	Giorgos	Loizidis
5	it21628	Vasilis	Matzaris
6	it21780	Fanis	Papazoglou
7	it21752	Ioannis	Panagiotopoulos

Items per page: 5 1 – 5 of

Add Student to Group

Selected: it21672

Choose Student

Save Cancel

You cannot subscribe a student to group, since it is full of students

Close

Εικ.75. Αδυναμία προσθήκης φοιτητή σε γεμάτη ομάδα

Προβολή στοιχείων εγγεγραμμένου φοιτητή

Students of Groups

Selected: Parallel Computing and Algorithms

Choose Course Schedule

☒ Theory Groups ☐ Lab Groups

Selected: group_1

Choose Class Group

Add Student to Group

Search students

ID	Username	First Name	Last Name
3	it21526	Ioannis	Makris
4	it21569	Giorgos	Loizidis
5	it21628	Vasilis	Matzaris

Items per page: 3 1 – 3 of 5

Student Details

Username: it21526

First Name: Ioannis

Last Name: Makris

Email: it21526@hua.gr

Unsubscribe student

Cancel

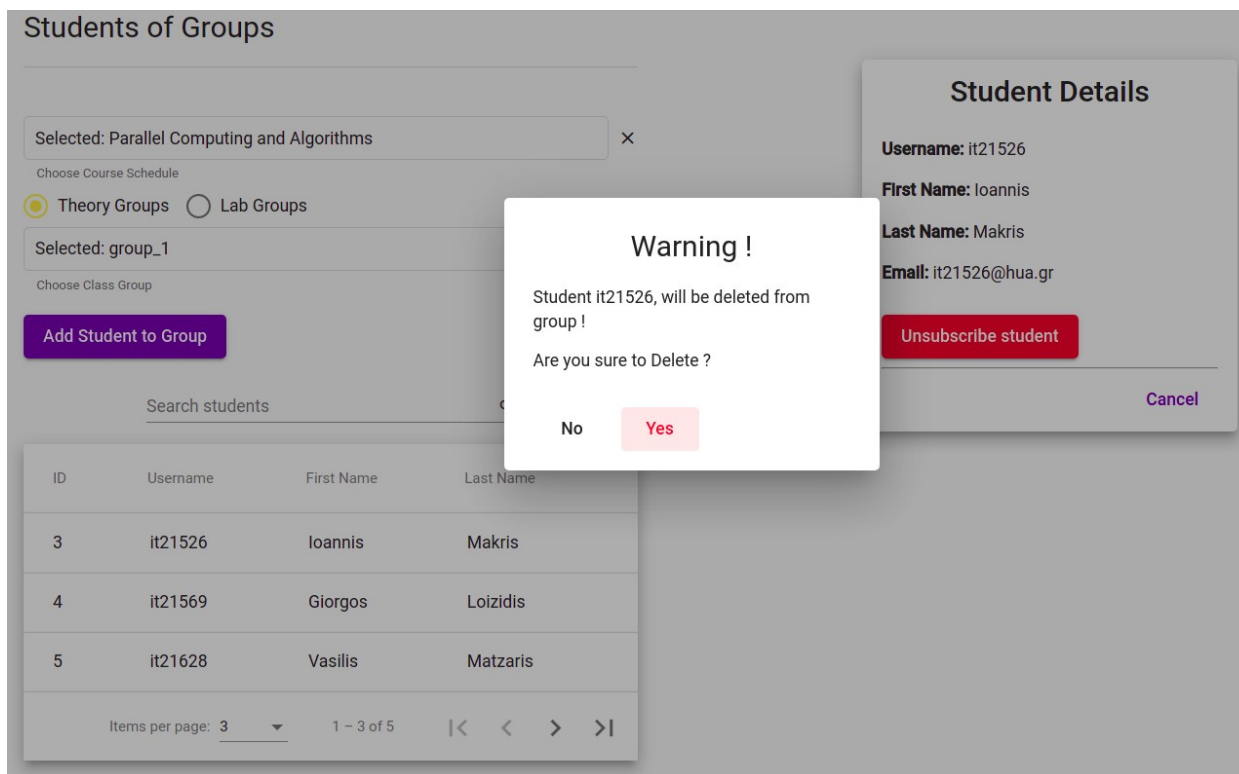
Εικ.76. Προβολή στοιχείων εγγεγραμμένου φοιτητή

Ο καθηγητής πρέπει πρώτα να επιλέξει το πρόγραμμα μαθήματος και τον τύπο της ομάδας (ομάδα θεωρίας ή ομάδα εργαστηρίου). Επειτα του δίνεται η δυνατότητα επιλογής μιας από τις ομάδες φοιτητών που αντιστοιχούν στο πρόγραμμα μαθήματος και τον τύπο ομάδας που επέλεξε.

Παράλληλα μπορεί να αναζητήσει φοιτητές της ομάδας μέσω της μπάρας αναζήτησης στο πάνω μέρος του πίνακα. Η αναζήτηση μπορεί να γίνει με κριτήρια το “Username”, το “First Name” ή το “Last Name” του φοιτητή.

Επειτα πλοηγούμενος στον πίνακα με τους φοιτητές, μπορεί να επιλέξει έναν φοιτητή πατώντας στην γραμμή του πίνακα στην οποία αντιστοιχεί. Έτσι έχει την δυνατότητα να δει περισσότερες λεπτομέρειες σχετικά με τον φοιτητή της επιλεγμένης ομάδας, στην καρτέλα που εμφανίζεται δεξιά του πίνακα.

Αφαίρεση φοιτητή απο την ομάδα



Εικ.77. Αφαίρεση φοιτητή απο την ομάδα

Στην καρτέλα προβολής των στοιχείων του φοιτητή, ο καθηγητής μπορεί να επιλέξει το κουμπί “Unsubscribe student” για να διαγράψει τον επιλεγμένο φοιτητή απο την ομάδα. Το σύστημα θα εμφανίσει ένα αναδυόμενο παράθυρο το οποίο ρωτάει τον καθηγητή, αν είναι βέβαιος για την διαγραφή του φοιτητή ή όχι απο την ομάδα.

Αν ο καθηγητής θέλει να διαγράψει τον επιλεγμένο φοιτητή απο την ομάδα πατάει το κουμπί “Yes”. Για να γίνει η διαγραφή ενός φοιτητή απο μια ομάδα, θα πρέπει η ομάδα να μην σχετίζεται με κάποια συνεδρία διάλεξης, διαφορετικά εμφανίζεται στον καθηγητή κατάλληλο μήνυμα όπως φαίνεται στην παρακάτω εικόνα.

Students of Groups

Selected: Java I

Choose Course Schedule

☒ Theory Groups
 ☐ Lab Groups

Selected: group_1

Choose Class Group

Add Student to Group

Search students

ID	Username	First Name	Last Name
4	it21569	Giorgos	Loizidis
5	it21628	Vasilis	Matzaris
6	it21780	Fanis	Papazoglou

Items per page: 3

1 – 3 of 3

Student Details

Username: it21569

First Name: Giorgos

Last Name: Loizidis

Email: it21569@hua.gr

Unsubscribe student

Cancel

You cannot unsubscribe a student from a group that already exists on a session

Close

Εικ.78. Αδυναμία αφαίρεσης φοιτητή απο ομάδα σχετιζόμενη με συνεδρία διάλεξης

Προσθήκη συνεδρίας διάλεξης

Classes Sessions

Selected: Java I

Choose Course Schedule

☐ Theories
 ☒ Labs

Selected: lab_2

Choose Lecture

Status

-- All --

Choose Status

Add Class Session

Search sessions

ID	Name	Date & Time ↑	Class Group	Status
----	------	---------------	-------------	--------

Items per page: 3

0 of 0

Add class session

Selected: group_4

Choose Class Group

Date

1/28/2022

Set date

Timespan

18:00 - 20:00

No.

1

Set number.

Save

Cancel

Εικ.79. Προσθήκη συνεδρίας διάλεξης

224

ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΑΡΟΥΣΙΩΝ – ΗΛΕΚΤΡΟΝΙΚΟ ΠΑΡΟΥΣΙΟΛΟΓΙΟ / ΜΑΚΡΗΣ ΙΩΑΝΝΗΣ

Ο καθηγητής πρέπει πρώτα να επιλέξει πρόγραμμα μαθήματος και τύπο διάλεξης (θεωρητική ή εργαστηριακή). Επειτα του δίνεται η δυνατότητα επιλογής μιας απο τις διαλέξεις που αντιστοιχούν στο πρόγραμμα μαθήματος και τον τύπο διάλεξης που επέλεξε.

Πατώντας το κουμπί “Add Class Session” μπορεί να προσθέσει μια συνεδρία για την επιλεγμένη διάλεξη, συμπληρώνοντας την αντίστοιχη φόρμα που εμφανίζεται δεξιά του πίνακα. Αξίζει να αναφερθεί ότι στον επιλογέα ομάδων της φόρμας προσθήκης συνεδρίας διάλεξης, περιλαμβάνονται μόνο οι ομάδες των οποίων οι εγγραφές είναι κλειστές προς τους φοιτητές.

Για να γίνει η δημιουργία μιας συνεδρίας για μια διάλεξη, θα πρέπει:

- Η ομάδα φοιτητών της συνεδρίας διάλεξης να μην σχετίζεται με άλλη συνεδρία για την ίδια διάλεξη. Δηλαδή να μην μπορούν να δημιουργηθούν πολλαπλές συνεδρίες παρακολούθησης για την ίδια διάλεξη, αλλά θα πρέπει να αφορούν διαφορετικές ομάδες φοιτητών. Σε αντίθετη περίπτωση το σύστημα ενημερώνει τον καθηγητή με κατάλληλο μήνυμα. Σχετικό παράδειγμα φαίνεται στην παρακάτω εικόνα.

The screenshot displays the 'Classes Sessions' management interface. On the left, a form allows selecting a course schedule (Java I), a lecture type (Theories), and a specific lecture (theory_1). A status dropdown is set to 'All'. A purple 'Add Class Session' button is visible. Below this is a search bar and a table of existing sessions.

ID	Name	Date & Time	Class Group	Status
31	session_1	12/30/2021 09:00 - 12:00	group_1	Past

At the bottom, a red error message states: 'A class session for Java I Schedule's theory_1 and group_1 already exists'.

On the right, the 'Add class session' modal is shown. It contains fields for 'Selected: group_1', 'Date' (1/27/2022), 'Set date', 'Timespan' (09:00 - 12:00), and 'No.' (2). 'Save' and 'Cancel' buttons are at the bottom.

Εικ.80. Αδυναμία δημιουργίας ήδη υπάρχουσας συνεδρίας διάλεξης

- Η επιλεγμένη ομάδα φοιτητών της προς δημιουργία συνεδρίας διάλεξης, να έχει τον ελάχιστο αριθμό εγγεγραμμένων φοιτητών που ορίζει το σύστημα διαχείρισης παρουσιών. Σε αντίθετη περίπτωση το σύστημα ενημερώνει τον καθηγητή με κατάλληλο μήνυμα. Σχετικό παράδειγμα φαίνεται στην παρακάτω εικόνα.

Classes Sessions

Selected: Java I

Choose Course Schedule

Theories

Labs

Selected: theory_1

Choose Lecture

Status

-- All --

Choose Status

Add Class Session

Search sessions

Q

ID	Name	Date & Time ↑	Class Group	Status
31	session_1	12/30/2021 09:00 - 12:00	group_1	Past

Items per page: 3

1 - 1 of 1

<< < > >>

Add class session

Selected: group_2

Choose Class Group

Date

1/27/2022

Set date

Timespan

12:00 - 15:00

No.

2

Set number.

Save

Cancel

The choosen class group has not the required minimum number of 10 students

Close

Εικ.81. Ανεπαρκής ομάδα φοιτητών

- Η επιλεγμένη ημερομηνία και ώρα διεξαγωγής της προς δημιουργίας συνεδρίας διάλεξης, να μην είναι πεπερασμένη. Σε αντίθετη περίπτωση το σύστημα ενημερώνει τον καθηγητή με κατάλληλο μήνυμα. Σχετικό παράδειγμα φαίνεται στην παρακάτω εικόνα.

Classes Sessions

Selected: Java I

Choose Course Schedule

☒ Theories ☐ Labs

Selected: theory_1

Choose Lecture

Status

-- All --

Choose Status

Add Class Session

Search sessions

ID	Name	Date & Time ↑	Class Group	Status
31	session_1	12/30/2021 09:00 - 12:00	group_1	Past

Items per page: 3

1 - 1 of 1

<<

<

>

>>

Add class session

Selected: group_2

Choose Class Group

Date

1/24/2022

Set date

Timespan

12:00 - 15:00

No.

2

Set number.

Save

Cancel

You cannot create a class session using a past date and time

Close

Εικ.82. Πεπερασμένη ημερομηνία και ώρα

- Η επιλεγμένη ημερομηνία και ώρα διεξαγωγής της προς δημιουργίας συνεδρίας διάλεξης, να μην συμπίπτει ταυτόχρονα με το χρονικό πλαίσιο διεξαγωγής κάποιας άλλης. Σε αντίθετη περίπτωση το σύστημα ενημερώνει τον καθηγητή με κατάλληλο μήνυμα.

Είναι σημαντικό να τονιστεί ότι στα πλαίσια της δημιουργίας μιας συνεδρίας διάλεξης, στην φόρμα προσθήκης συμπληρώνεται απο τον καθηγητή μόνο η ημερομηνία. Η ώρα της συνεδρίας διάλεξης προκύπτει απο την ομάδα παρακολούθησης που επιλέγει ο καθηγητής.

Στην παρακάτω εικόνα φαίνεται το ενημερωτικό μήνυμα του συστήματος σε περίπτωση που η ημερομηνία και ώρα, της προς δημιουργία συνεδρίας διάλεξης, συμπίπτει με το χρονικό πλαίσιο κάποιας άλλης.

Classes Sessions

Selected: Java I

Choose Course Schedule

☒ Theories ☐ Labs

Selected: theory_2

Choose Lecture

Status

-- All --

Choose Status

Add Class Session

Search sessions

ID	Name ↑	Date & Time	Class Group	Status
32	session_1	01/03/2022 09:00 - 12:00	group_1	Past

Items per page: 3

1 - 1 of 1

|<

<

>

>|

Add class session

Selected: group_2

Choose Class Group

Date

3/11/2022

Set date

Timespan

12:00 - 15:00

No.

2

Set number.

Save

Cancel

This is a winter semester course: Put month value between October and February

Close

Εικ.84. Ασύμβατη με το εξάμηνο τιμή του μήνα στην ημερομηνία

- Να μην συμμετέχει ο ίδιος καθηγητής σε κάποια άλλη διάλεξη που εξελίσσεται ταυτόχρονα. Σε περίπτωση που ο καθηγητής επιχειρήσει την δημιουργία συνεδρίας διάλεξης ενώ ταυτόχρονα έχει δημιουργηθεί μια άλλη στην οποία συμμετέχει, το σύστημα εμποδίζει την διαδικασία ενημερώνοντάς τον με κατάλληλο μήνυμα όπως φαίνεται στην παρακάτω εικόνα.

Classes Sessions

Selected: Web Development

Choose Course Schedule

Theories

Labs

Selected: lab_1

Choose Lecture

Status

-- All --

Choose Status

Add Class Session

Search sessions

ID	Name ↑	Date & Time	Class Group	Status
Items per page: 3 0 of 0 < < > >				

Add class session

Selected: group_1

Choose Class Group

Date

1/25/2022

Set date

Timespan

18:00 - 20:00

No.

1

Set number.

Save

Cancel

It's impossible to have the same teacher(s) is the same date and time range of another session

Close

Εικ.85. Ο καθηγητής μετέχει σε άλλη συνεδρία διάλεξης που εξελίσσεται ταυτόχρονα

Προβολή στοιχείων συνεδρίας διάλεξης

Classes Sessions

Selected: Java I

Choose Course Schedule

☐ Theories ☒ Labs

Selected: lab_2

Choose Lecture

Status

-- All --

Choose Status

Add Class Session

Search sessions

Q

ID	Name ↑	Date & Time	Class Group	Status
34	session_1	01/28/2022 18:00 - 20:00	group_4	Pending

Items per page: 3 1 - 1 of 1 << < > >>

Class Session Details

Name: session_1

Date: 01/28/2022

Time: 18:00 - 20:00

Course Schedule: Java I

Lecture: lab_2, "Create your first Class"

Class Group: group_4

Teaching Stuff: Konstantinos Tserpes, Vasileios Dalakas, Christos Sardianos

Students:

Preview

Presence Statements: Closed

Status: Pending

Edit

Delete

Cancel

Εικ.86. Προβολή στοιχείων συνεδρίας διάλεξης

Ο καθηγητής πρέπει πρώτα να επιλέξει πρόγραμμα μαθήματος και τύπο διάλεξης (θεωρητική ή εργαστηριακή). Επειτα του δίνεται η δυνατότητα επιλογής μιας απο τις διαλέξεις που αντιστοιχούν στο πρόγραμμα μαθήματος και τον τύπο διάλεξης που επέλεξε.

Επιλέγοντας διάλεξη ο καθηγητής μπορεί να δει τις συνεδρίες που σχετίζονται με αυτήν στον πίνακα. Παράλληλα έχει την δυνατότητα να φιλτράρει το ποιές συνεδρίες διαλέξεων θέλει να δει στον πίνακα, επιλέγοντας την κατάσταση συνεδρίας διάλεξης (class session status) που τον ενδιαφέρει.

Η κατάσταση συνεδρίας διάλεξης μπορεί να είναι:

- Τρέχουσα (Current): Η συνεδρία διάλεξης εκτυλίσσεται εκείνη την στιγμή.
- Εκκρεμής (Pending): Η συνεδρία διάλεξης δεν έχει ακόμη ξεκινήσει.
- Πεπερασμένη (Past): Η συνεδρία διάλεξης έχει πραγματοποιηθεί και έχει πια περάσει στο ιστορικό.

Επιπλέον μπορεί να αναζητήσει συνεδρίες διάλεξης μέσω της μπάρας αναζήτησης στο πάνω μέρος του πίνακα. Η αναζήτηση μπορεί να γίνει με κριτήρια το “Name”, το “Date & Time” ή το “Class Group”. Στην συνέχεια, πλοηγούμενος στον πίνακα με τις συνεδρίες διαλέξεων, μπορεί να επιλέξει μια συνεδρία διάλεξης πατώντας την γραμμή του πίνακα στην οποία αντιστοιχεί. Έτσι έχει την δυνατότητα να δει περισσότερες λεπτομέρειες σχετικά με την συνεδρία διάλεξης, στην καρτέλα που εμφανίζεται δεξιά του πίνακα.

Στην καρτέλα των πληροφοριών της επιλεγμένης συνεδρίας διάλεξης περιλαμβάνονται και οι φοιτητές που συμμετέχουν σε αυτήν. Όμως, ο όγκος δεδομένων των φοιτητών που συμμετέχουν στην εκάστοτε συνεδρία διάλεξης μπορεί να είναι μεγάλος. Για αυτό υπάρχει το κουμπί “Preview”, μέσω του οποίου ανοίγει αναδυόμενο παράθυρο που περιέχει σχετικό πίνακα με τις πληροφορίες των συμμετεχόντων στην συνεδρία διάλεξης φοιτητών.

Στην παρακάτω εικόνα φαίνονται τα στοιχεία των φοιτητών μιας συνεδρίας διάλεξης στο σχετικό αναδυόμενο παράθυρο.

The image shows a web application interface with a modal window titled "List of Students:". The modal contains a search bar and a table of student information. The background shows a "Classes Sessions" page with a table of sessions and a "Class Session Details" panel on the right.

ID	Username	First Name	Last Name
7	it21752	Ioannis	Panagiotopoulos
8	it21672	Giorgos	Papadopoulos
9	it213101	Ioannis	Chalkiadakis
10	it21788	Christos	Oikonomou
11	it21972	Manolis	Grigoriou

Below the table, there is a pagination control showing "1 - 5 of 5" and "Items per page: 5". A "Close" button is at the bottom left of the modal.

Εικ.87. Φοιτητές συνεδρίας διάλεξης

Είναι χρήσιμο να αναφερθεί ότι:

- Η δυνατότητα επεξεργασίας μιας συνεδρίας διάλεξης είναι εφικτή μόνο αν η συνεδρία διάλεξης είναι εκκρεμής (pending), δηλαδή δεν έχει ακόμα ξεκινήσει η διεξαγωγή της ή τρέχουσα (current). Για αυτό και στην καρτέλα με τις πληροφορίες της συνεδρίας διάλεξης, το κουμπί “Edit”, εμφανίζεται όταν η συνεδρία διάλεξης είναι εκκρεμής (pending) ή τρέχουσα (current).
- Η δυνατότητα διαγραφής μιας συνεδρίας διάλεξης είναι εφικτή μόνο αν η συνεδρία διάλεξης είναι εκκρεμής (pending), δηλαδή δεν έχει ακόμα ξεκινήσει η διεξαγωγή της. Για αυτό και στην καρτέλα με τις πληροφορίες της συνεδρίας διάλεξης, το κουμπί “Delete”, εμφανίζεται όταν η συνεδρία διάλεξης είναι εκκρεμής (pending).

Επεξεργασία συνεδρίας διάλεξης

Οπως προαναφέρθηκε η επεξεργασία μιας συνεδρίας διάλεξης είναι εφικτή μόνο αν η συνεδρία διάλεξης είναι εκκρεμής (pending) ή τρέχουσα (current). Το σενάριο της επεξεργασίας συνεδρίας διάλεξης θα αναλυθεί σε δύο υποσενάρια για κάθε περίπτωση ξεχωριστά. Αυτό συμβαίνει γιατί ανάλογα με το αν η συνεδρία διάλεξης είναι εκκρεμής (pending) ή τρέχουσα (current), γίνεται διαφορετικού τύπου επεξεργασία.

Πρώτη Περίπτωση: Η συνεδρία διάλεξης είναι εκκρεμής.

The screenshot displays the 'Classes Sessions' management interface. On the left, there are filters for 'Selected: Java I' (Course Schedule), 'Selected: lab_2' (Lecture), and 'Status: Pending'. A purple 'Add Class Session' button is visible. Below the filters is a table of sessions. The table has columns: ID, Name, Date & Time, and Class Group. One session is listed with ID 34, Name session_1, Date & Time 01/28/2022 18:00 - 20:00, and Class Group group_4. To the right, an 'Edit session_1' modal is open, showing fields for Date (1/28/2022), Timespan (18:00 - 20:00), and No. (1). The modal includes 'Update' and 'Cancel' buttons.

ID	Name	Date & Time	Class Group
34	session_1	01/28/2022 18:00 - 20:00	group_4

Εικ.88 Επεξεργασία εκκρεμούς συνεδρίας διάλεξης

Στην καρτέλα προβολής των στοιχείων της εκκρεμούς (pending) συνεδρίας διάλεξης, ο καθηγητής μπορεί να επιλέξει το κουμπί “Edit” και να επεξεργαστεί τα στοιχεία στην αντίστοιχη φόρμα που εμφανίζεται απο το σύστημα.

Οπως αναφέρθηκε προηγουμένως η ώρα διεξαγωγής της συνεδρίας διάλεξης προκύπτει απο την ομάδα φοιτητών που της αντιστοιχεί. Ετσι στην περίπτωση επεξεργασίας μιας εκκρεμούς (pending) συνεδρίας διάλεξης η ώρα διεξαγωγής της δεν επιδέχεται αλλαγές.

Οσον αφορά την ημερομηνία θα πρέπει να μην είναι πεπερασμένη σε συνδυασμό πάντα με την ώρα. Σε περίπτωση που ο καθηγητής επιλέξει ημερομηνία η οποία συναρτήσει της ώρας διεξαγωγής της διάλεξης είναι πεπερασμένη, η επεξεργασία της συνεδρίας διάλεξης είναι αδύνατη και το σύστημα ενημερώνει τον καθηγητή με το κατάλληλο ενημερωτικό μήνυμα. Η παρακάτω εικόνα αφορά μια τέτοια περίπτωση.

Classes Sessions

Selected: Java I

Choose Course Schedule

☐ Theories
 ☒ Labs

Selected: lab_2

Choose Lecture

Status

Pending

Choose Status

Add Class Session

Search sessions

ID	Name ↑	Date & Time	Class Group
34	session_1	01/25/2022 18:00 - 20:00	group_4

Items per page: 3

1 – 1 of 1

<<

<

>

>>

Edit session_1

Date

1/24/2022

Set date

Timespan

18:00 - 20:00

No.

1

Set number.

Update

Cancel

You cannot update a class session using a past date and time

Close

Εικ.89. Πεπερασμένη ημερομηνία και ώρα συνεδρίας διάλεξης

Δεύτερη Περίπτωση: Η συνεδρία διάλεξης είναι τρέχουσα.

Classes Sessions

Selected: Java I

Choose Course Schedule

☐ Theories
 ☒ Labs

Selected: lab_2

Choose Lecture

Status

Current

Choose Status

Add Class Session

Search sessions

ID	Name ↑	Date & Time	Class Group	Ps Status
34	session_1	01/25/2022 18:00 - 20:00	group_4	Closed

Items per page: 3

1 – 1 of 1

<<

<

>

>>

Edit session_1

Presence Statement:

☐ Opened
 ☒ Closed

Cancel

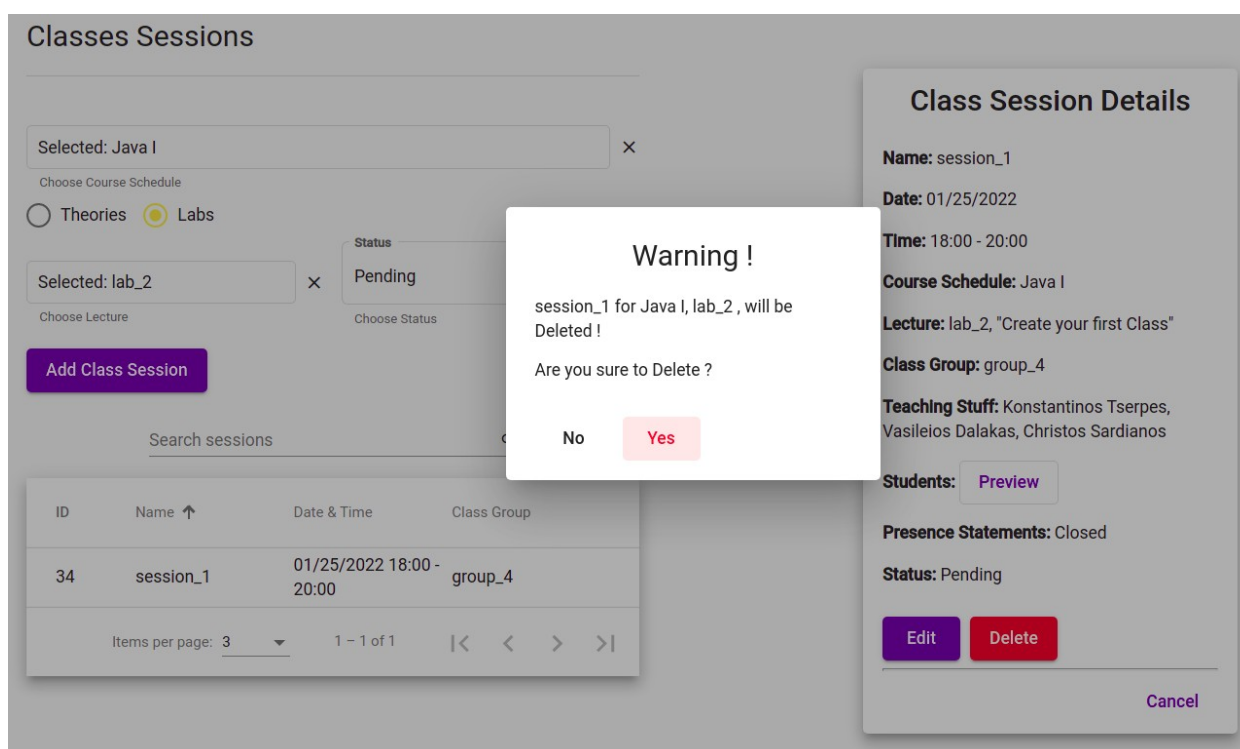
Εικ.90. Επεξεργασία τρέχουσας συνεδρίας διάλεξης

Στην καρτέλα προβολής των στοιχείων της τρέχουσας (current) συνεδρίας διάλεξης, ο καθηγητής μπορεί να επιλέξει το κουμπί “Edit” και να ανοίξει ή να κλείσει τις δηλώσεις παρουσιών για τους φοιτητές στην αντίστοιχη φόρμα που εμφανίζεται απο το σύστημα.

Είναι σημαντικό να αναφερθεί ότι κατα το πρώτο άνοιγμα των δηλώσεων παρουσιών σε μια τρέχουσα συνεδρία διάλεξης, δημιουργούνται “κενές” παρουσίες για τους φοιτητές που συμμετέχουν σε αυτήν. Επίσης κατα το πρώτο κλείσιμο των δηλώσεων παρουσιών σε μια τρέχουσα συνεδρία διάλεξης οι αδήλωτες παρουσίες των φοιτητών καταχωρούνται στο σύστημα ως αδικαιολόγητες απουσίες.

Διαγραφή συνεδρίας διάλεξης

Οπως έχει αναφερθεί και προηγουμένως το σενάριο της διαγραφής συνεδρίας διάλεξης, αφορά την συνεδρία διάλεξης που είναι εκκρεμής (pending).



Εικ.91. Διαγραφή συνεδρίας διάλεξης

Στην καρτέλα προβολής των στοιχείων της εκκρεμούς (pending) συνεδρίας διάλεξης, ο καθηγητής μπορεί να επιλέξει το κουμπί “Delete” για να διαγράψει την επιλεγμένη συνεδρία διάλεξης.

Το σύστημα θα εμφανίσει ένα αναδυόμενο παράθυρο το οποίο ρωτάει τον καθηγητή, αν είναι βέβαιος για την διαγραφή ή όχι, οπότε αν θέλει να διαγράψει την επιλεγμένη συνεδρία διάλεξης πατάει το κουμπί “Yes”.

Προβολή στοιχείων παρουσίας φοιτητή

The screenshot displays the 'Presences' management interface. On the left, there are filters for 'Selected: Introduction to C', 'Choose Course Schedule' (Theories/Labs), 'Selected: lab_8', 'Selected: session_1', and a 'Status' dropdown set to 'All'. Below these is a search bar and a table of student presences. On the right, a 'Presence Details' modal is open, showing information for student 'it21526' in 'Introduction to C', 'lab_8' session 'session_1' on '01/25/2022 16:00 - 18:00'. The status is 'Absent' with an 'Inexcusable' excuse. The modal includes 'Edit' and 'Cancel' buttons.

ID	Username	First Name	Last Name	Status
39	it21752	Ioannis	Panagiotopoulos	Present
40	it21526	Ioannis	Makris	Absent
41	it21569	Giorgos	Loizidis	Absent

Εικ.92. Προβολή στοιχείων παρουσίας φοιτητή

Ο καθηγητής πρέπει πρώτα να επιλέξει πρόγραμμα μαθήματος και τύπο διάλεξης (θεωρητική ή εργαστηριακή). Επειτα του δίνεται η δυνατότητα επιλογής μιας απο τις διαλέξεις που αντιστοιχούν στο πρόγραμμα μαθήματος και τον τύπο διάλεξης που επέλεξε.

Στην συνέχεια, ο καθηγητής πρέπει να επιλέξει την συνεδρία διάλεξης της οποίας θέλει να δει τις παρουσίες των φοιτητών. Επιλέγοντας συνεδρία διάλεξης ο καθηγητής μπορεί να δει τις παρουσίες των φοιτητών που συμμετέχουν σε αυτήν στον πίνακα. Παράλληλα έχει την δυνατότητα να φιλτράρει το είδος των παρουσιών που θέλει να δει στον πίνακα, επιλέγοντας την κατάσταση παρουσίας φοιτητή (student presence status) που τον ενδιαφέρει.

Η κατάσταση παρουσίας του φοιτητή μπορεί να είναι:

- **Εκκρεμής (Pending):** Η παρουσία δεν έχει ακόμη δηλωθεί από τον φοιτητή ή ο καθηγητής δεν έχει κλείσει ακόμη τις δηλώσεις παρουσιών για τους φοιτητές, καθόσον η συνεδρία διάλεξης εκτυλίσσεται εκείνη την στιγμή.
- **Παρών (Present):** Η παρουσία έχει δηλωθεί από τον φοιτητή ή από τον καθηγητή. Έτσι ο φοιτητής θεωρείται πλέον παρών στην τρέχουσα συνεδρία διάλεξης.
- **Απών (Absent):** Η παρουσία δεν έχει δηλωθεί από τον φοιτητή ή ο καθηγητής έχει μετατρέψει για δικούς του λόγους την παρουσία του σε απουσία. Έτσι ο φοιτητής χαρακτηρίζεται απών.

Στο σημείο αυτό πρέπει να αναφερθεί ότι στον επιλογέα της κατάστασης παρουσίας του φοιτητή, αν ο καθηγητής επιλέξει την ένδειξη “Absent” τότε δεξιά από τον επιλογέα εμφανίζεται ένας ακόμη, που αφορά όμως την κατάσταση της απουσίας (absence status).

Η κατάσταση της απουσίας του φοιτητή μπορεί να είναι:

- **Δικαιολογημένη (Excused):** Ο φοιτητής έχει κάνει αίτηση δικαιολόγησης στην γραμματεία και η αίτηση έχει εγκριθεί. Έτσι η απουσία του είναι πια δικαιολογημένη.
- **Αδικαιολόγητη (Inexcusable):** Ο φοιτητής δεν έχει κάνει αίτηση δικαιολόγησης στην γραμματεία, ή ακόμη και να έχει κάνει, η αίτησή του έχει απορριφθεί.

Στην παρακάτω εικόνα φαίνεται ο επιλογέας κατάστασης απουσίας.

Presences

Selected: Introduction to C

Choose Course Schedule

☐ Theories
☒ Labs

Selected: lab_8

Choose Lecture

Status

Absent

Choose Status

Selected: session_1

Choose Class Session

Excuse Status

-- All --

Excused

Inexcusable

Search presences

ID	Username	First Name		
41	it21569	Giorgos	Loizidis	Excused
43	it21780	Fanis	Papazoglou	Inexcusable

Items per page: 3

1 – 2 of 2

<<

<

>

>>

Εικ.93. Επιλογέας κατάστασης απουσίας

Ο καθηγητής μπορεί να αναζητήσει παρουσίες μέσω της μπάρας αναζήτησης στο πάνω μέρος του πίνακα. Η αναζήτηση μπορεί να γίνει με κριτήρια το “Username”, το “First Name” ή το “Last Name” της παρουσίας του φοιτητή που τον ενδιαφέρει.

Στην συνέχεια, πλοηγούμενος στον πίνακα με τις παρουσίες, μπορεί να επιλέξει μια παρουσία πατώντας την γραμμή του πίνακα στην οποία αντιστοιχεί. Έτσι έχει την δυνατότητα να δει περισσότερες λεπτομέρειες σχετικά με την παρουσία, στην καρτέλα που εμφανίζεται δεξιά του πίνακα.

Τέλος, στην καρτέλα που δείχνει τις λεπτομέρειες της επιλεγμένης παρουσίας, αν η παρουσία δεν αφορά ήδη δικαιολογημένη απουσία, τότε το σύστημα εμφανίζει ένα κουμπί “Edit”, ώστε ο καθηγητής να μπορεί καθόλη την διάρκεια της τρέχουσας συνεδρίας διάλεξης να μεταβάλλει μια παρουσία σε απουσία και το αντίστροφο.

Επεξεργασία παρουσίας

Presences

Selected: Introduction to C

Choose Course Schedule

☐ Theories ☒ Labs

Selected: lab_8

Choose Lecture

Selected: session_1

Choose Class Session

Status

-- All --

Choose Status

Search presences

ID	Username	First Name	Last Name	Status
39	it21752	Ioannis	Panagiotopoulos	Present
40	it21526	Ioannis	Makris	Absent
41	it21569	Giorgos	Loizidis	Absent

Items per page: 3 1 – 3 of 5

Edit it21526 Presence

☒ Present ☐ Absent

Update

Cancel

Εικ.94. Επεξεργασία παρουσίας

Στην καρτέλα προβολής των στοιχείων της παρουσίας φοιτητή, ο καθηγητής μπορεί να επιλέξει το κουμπί “Edit” και να επεξεργαστεί την κατάσταση της παρουσίας του φοιτητή, θέτοντάς τον παρών ή απών, στην αντίστοιχη φόρμα που εμφανίζεται απο το σύστημα.

Ο καθηγητής μπορεί να μεταβάλλει την κατάσταση της παρουσίας του φοιτητή μόνο όταν:

- Η συνεδρία διάλεξης είναι τρέχουσα (current). Σε αντίθετη περίπτωση η αλλαγή της κατάστασης παρουσίας είναι αδύνατη και το σύστημα ενημερώνει τον καθηγητή με κατάλληλο μήνυμα. Μια τέτοια περίπτωση φαίνεται στην παρακάτω εικόνα.

Presences

Selected: Java I

Choose Course Schedule

☒ Theories
 ☐ Labs

Selected: theory_1

Choose Lecture

Selected: session_1

Choose Class Session

Status

-- All --

Choose Status

Search presences

ID	Username	First Name	Last Name	Status
24	it21780	Fanis	Papazoglou	Present
25	it21972	Manolis	Grigoriou	Absent
26	it21628	Vasilis	Matzaris	Absent

Items per page: 3

1 – 3 of 10

<<

<

>

>>

Edit it21628 Presence

☐ Present
 ☒ Absent

Update

Cancel

You cannot update the status for a presence of a past class session

Close

Εικ.95. Αδυναμία επεξεργασίας παρουσίας πεπερασμένης συνεδρίας διάλεξης

- Σε περίπτωση που η κατάσταση της παρουσίας του φοιτητή είναι “Absent”, δηλαδή απών, θα πρέπει ο φοιτητής να μην έχει κάνει ήδη κάποια αίτηση δικαιολόγησης.

Σε αντίθετη περίπτωση η αλλαγή της κατάστασης παρουσίας είναι αδύνατη και το σύστημα ενημερώνει τον καθηγητή με κατάλληλο μήνυμα. Μια τέτοια περίπτωση φαίνεται στην παρακάτω εικόνα.

Presences

Selected: Introduction to C

Choose Course Schedule

Theories

Labs

Selected: lab_8

Choose Lecture

Selected: session_1

Choose Class Session

Status

-- All --

Choose Status

Search presences

ID	Username	First Name	Last Name	Status
42	it21628	Vasilis	Matzaris	Present
43	it21780	Fanis	Papazoglou	Absent

Items per page: 3

4 – 5 of 5

|<

<

>

|>

Edit it21780 Presence

Present

Absent

Update

Cancel

This absence had already have an excuse application

Close

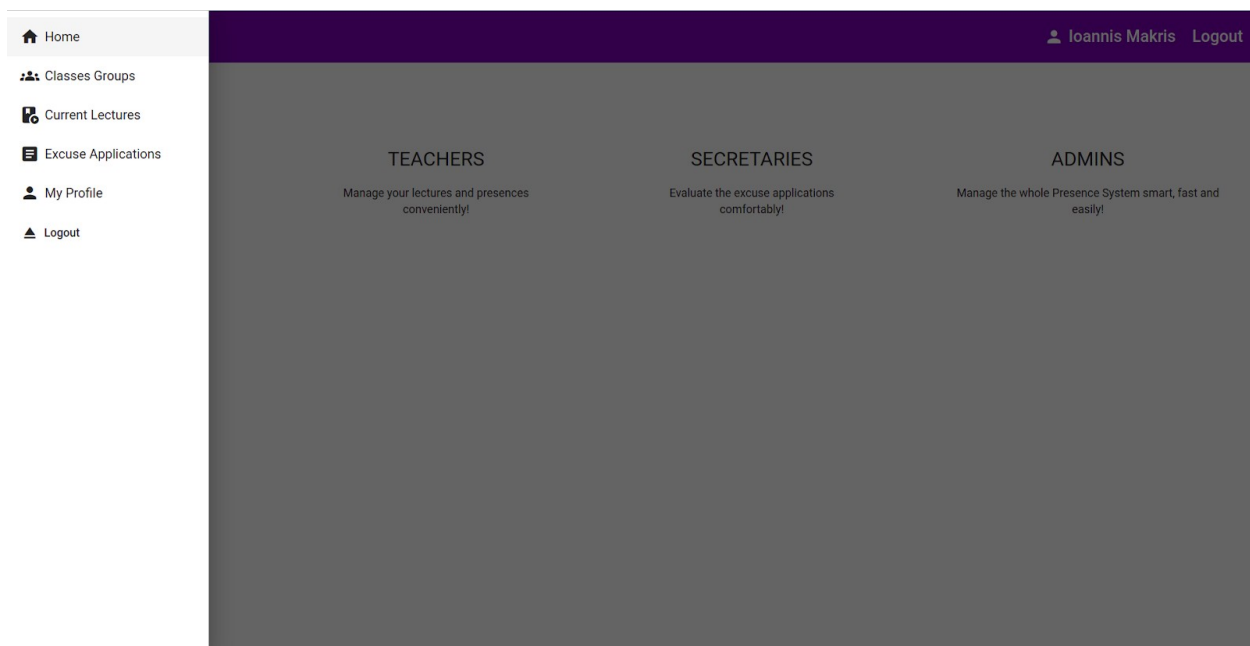
Εικ.96. Αδυναμία επεξεργασίας παρουσίας που έχει αίτηση δικαιολόγησης

5.5. Λειτουργίες Φοιτητή (Student)

Στην ενότητα αυτή παρουσιάζονται τα σενάρια χρήσης για τις λειτουργίες που επιτελεί ο ρόλος του φοιτητή. Η δράση του φοιτητή στο σύστημα διαχείρισης παρουσιών, περιορίζεται στο τμήμα στο οποίο ανήκει. Επίσης όλες οι λειτουργίες του φοιτητή προϋποθέτουν να είναι μέλος του προγράμματος μαθήματος που σχετίζεται άμεσα ή έμμεσα με αυτές.

Πλοήγηση στο μενού επιλογών

Ο φοιτητής αφού συνδεθεί στο σύστημα, μπορεί να δει το μενού επιλογών του και να πλοηγηθεί σε αυτό, πατώντας το κουμπί που αναπαρίσταται με τρεις οριζόντιες παύλες πάνω αριστερά στην οθόνη.



Εικ.97. Μενού Φοιτητή

Εγγραφή σε ομάδα

Classes Groups

Selected: Distributed Systems

Choose Course Schedule

Theories
Labs

Search groups

Name ↑	Time	Capacity	Subscription
group_2	15:30 - 17:30	7/7	<input type="checkbox"/>
group_3	18:00 - 20:00	7/7	<input checked="" type="checkbox"/>

Items per page: 3
1 – 2 of 2

You just subscribed to lab_group_3
Close

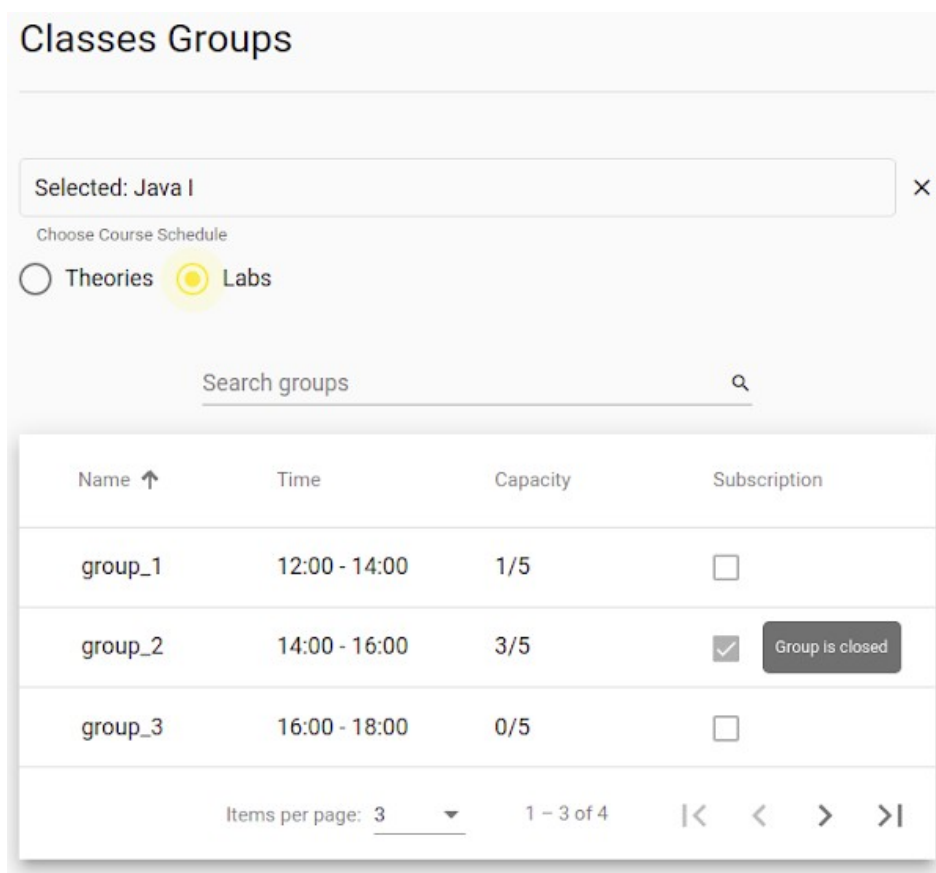
Εικ.98. Εγγραφή σε ομάδα

Ο φοιτητής πρέπει πρώτα να επιλέξει πρόγραμμα μαθήματος και τύπο ομάδας (ομάδα για παρακολούθηση θεωρίας ή ομάδα για παρακολούθηση εργαστηρίου). Επειτα μπορεί να αναζητήσει ομάδες μέσω της μπάρας αναζήτησης στο πάνω μέρος του πίνακα. Η αναζήτηση μπορεί να γίνει με κριτήρια το “Name”, ή το “Time” της ομάδας που τον ενδιαφέρει.

Στην συνέχεια πλοηγούμενος στον πίνακα μπορεί να εγγραφεί σε μια ομάδα, πατώντας το κουτάκι (checkbox) της αντίστοιχης γραμμής που βρίσκονται τα στοιχεία της ομάδας. Σε περίπτωση που το κουτάκι είναι απενεργοποιημένο (disabled), εμφανίζεται και κατάλληλο μηνυματάκι που εξηγεί στον φοιτητή για ποιον λόγο είναι αδύνατο να εγγραφεί στην συγκεκριμένη ομάδα.

Οι περιπτώσεις σύμφωνα με τις οποίες η εγγραφή είναι αδύνατη και κατ’ επέκταση το κουτάκι είναι απενεργοποιημένο είναι οι εξής:

- Η ομάδα είναι κλειστή ως προς τις εγγραφές φοιτητών.



The screenshot shows a web interface titled "Classes Groups". At the top, there is a search bar with "Selected: Java I" and a close button (X). Below the search bar, there is a section "Choose Course Schedule" with two radio buttons: "Theories" and "Labs". The "Labs" radio button is selected and highlighted with a yellow circle. Below this, there is a search bar labeled "Search groups" with a magnifying glass icon. The main part of the interface is a table with the following columns: "Name", "Time", "Capacity", and "Subscription". The table contains three rows of data:

Name ↑	Time	Capacity	Subscription
group_1	12:00 - 14:00	1/5	<input type="checkbox"/>
group_2	14:00 - 16:00	3/5	<input checked="" type="checkbox"/> Group is closed
group_3	16:00 - 18:00	0/5	<input type="checkbox"/>

At the bottom of the table, there is a pagination bar showing "Items per page: 3" with a dropdown arrow, "1 - 3 of 4", and navigation arrows (|< < > >|).

Εικ.99. Κλειστή προς εγγραφές ομάδα

- Η χωρητικότητα της ομάδας έχει εξαντληθεί.

Classes Groups

Selected: Distributed Systems

×

Choose Course Schedule

☐ Theories
 ☒ Labs

Name ↑	Time	Capacity	Subscription
group_2	15:30 - 17:30	7/7	<input type="checkbox"/> <div>Group is full</div>
group_3	18:00 - 20:00	7/7	<input checked="" type="checkbox"/>

Items per page: 3

1 – 2 of 2

Εικ.100. Γεμάτη ομάδα

Απεγγραφή από ομάδα

Classes Groups

Selected: Distributed Systems ×

Choose Course Schedule

☐ Theories ☒ Labs

Search groups 🔍

Name ↑	Time	Capacity	Subscription
group_2	15:30 - 17:30	7/7	<input type="checkbox"/>
group_3	18:00 - 20:00	6/7	<input type="checkbox"/>

Items per page: 3 ▼ 1 – 2 of 2 |< < > >|

You just unsubscribed from lab_group_3 Close

Εικ.101. Απεγγραφή από ομάδα

Ο φοιτητής πρέπει πρώτα να επιλέξει πρόγραμμα μαθήματος και τύπο ομάδας (ομάδα για παρακολούθηση θεωρίας ή ομάδα για παρακολούθηση εργαστηρίου). Επειτα μπορεί να αναζητήσει ομάδες μέσω της μπάρας αναζήτησης στο πάνω μέρος του πίνακα. Η αναζήτηση μπορεί να γίνει με κριτήρια το “Name”, ή το “Time” της ομάδας που τον ενδιαφέρει.

Στην συνέχεια πλοηγούμενος στον πίνακα μπορεί να απεγγραφεί από μια ομάδα που είναι εγγεγραμμένος, αποεπιλέγοντας το κουτάκι (checkbox) της αντίστοιχης γραμμής που βρίσκονται τα στοιχεία της ομάδας στην οποία είχε προηγουμένως εγγραφεί. Σε περίπτωση που το κουτάκι είναι απενεργοποιημένο (disabled), εμφανίζεται και κατάλληλο μηνυματάκι που εξηγεί στον φοιτητή για ποιον λόγο είναι αδύνατο να απεγγραφεί από την συγκεκριμένη ομάδα.

Η απεγγραφή είναι αδύνατη και κατ' επέκταση το κουτάκι είναι απενεργοποιημένο όταν η ομάδα είναι κλειστή προς εγγραφές απο φοιτητές.

Classes Groups

Selected: Introduction to C

×

Choose Course Schedule

☐ Theories

☒ Labs

Search groups

🔍

Name ↑	Time	Capacity	Subscription
group_1	11:00 - 13:00	0/5	<input type="checkbox"/>
group_2	16:00 - 18:00	5/5	<input checked="" type="checkbox"/> <div>Group is closed</div>

Items per page: 3

1 – 2 of 2

|< < > >|

Εικ.102. Αδυναμία απεγγραφής λόγω κλειστής ομάδας

Σε αυτό το σημείο είναι σημαντικό να αναφερθεί ότι στο σύστημα διαχείρισης παρουσιών, όταν μια ομάδα είναι κλειστή προς εγγραφές απο φοιτητές, είναι επίσης κλειστή και για απεγγραφές φοιτητών απο αυτήν (και στις δυο περιπτώσεις η διενέργεια εγγραφής / απεγγραφής αφορά τον ρόλο του φοιτητή καθώς ο καθηγητής μπορεί υπο κάποιες προϋποθέσεις που αναλύθηκαν στην ενότητα 5.4 να εγγράψει ή να διαγράψει κάποιον φοιτητή απο μια ομάδα).

Δήλωση παρουσίας

Current Lectures

Course	Lecture	Date & Time	Presence Statement
Distributed Systems	lab_5	01/26/2022 12:00 - 14:00	<input type="checkbox"/>
Parallel Computing and Algorithms	lab_8	01/26/2022 12:00 - 14:00	<input checked="" type="checkbox"/>

Items per page: 3 1 – 2 of 2 |< < > >|

Presence statement was successfully [Close](#)

Εικ.103. Δήλωση παρουσίας

Οπως φαίνεται στην παραπάνω εικόνα, για λόγους ευμορφίας στην διεπαφή χρήστη του φοιτητή, χρησιμοποιήθηκε το όνομα “Current Lectures” αντι του “Current Class Sessions”. Ο φοιτητής μπορεί αναζητήσει κάποια απο τις τρέχουσες διαλέξεις μέσω της μπάρας αναζήτησης στο πάνω μέρος του πίνακα. Η αναζήτηση μπορεί να γίνει με κριτήρια το “Course”, το “Lecture” ή το “Date & Time” της τρέχουσας διάλεξης που τον ενδιαφέρει.

Στην συνέχεια πλοηγούμενος στον πίνακα μπορεί να δηλώσει την παρουσία του σε μια τρέχουσα διάλεξη, πατώντας το κουτάκι (checkbox) της αντίστοιχης γραμμής που βρίσκονται τα στοιχεία της τρέχουσας διάλεξης.

Σε περίπτωση που το κουτάκι είναι απενεργοποιημένο (disabled), εμφανίζεται και κατάλληλο μηνυματάκι που εξηγεί στον φοιτητή για ποιον λόγο είναι αδύνατο να δηλώσει την παρουσία του στην συγκεκριμένη συνεδρία διάλεξης.

Οι περιπτώσεις σύμφωνα με τις οποίες η δήλωση παρουσίας στην τρέχουσα διάλεξη είναι αδύνατη και κατ' επέκταση το κουτάκι είναι απενεργοποιημένο είναι οι εξής:

- Ο φοιτητής έχει ήδη δηλωμένη την παρουσία του σε κάποια άλλη τρέχουσα διάλεξη που προφανώς εκτυλίσσεται ταυτόχρονα, οπότε δεν μπορεί να είναι σε πάνω απο μια παρών.

Current Lectures

Search current lectures

Course	Lecture	Date & Time	Presence Statement
Distributed Systems	lab_5	01/26/2022 12:00 - 14:00	<input type="checkbox"/>
Parallel Computing and Algorithms	lab_8	01/26/2022 12:00 - 14:00	<input checked="" type="checkbox"/>

Items per page: 3 1 - 2 of 2

You cannot make more than 1 presence statements for Lectures they are runnig in the same time

Εικ.104. Ηδη δηλωμένη παρουσία

- Οι δηλώσεις παρουσιών στην τρέχουσα διάλεξη έχουν απενεργοποιηθεί για τους φοιτητές. Στην παρακάτω εικόνα φαίνεται μια τέτοια περίπτωση.

Current Lectures

Search current lectures



Course	Lecture	Date & Time	Presence Statement
Distributed Systems	lab_5	01/26/2022 12:00 - 14:00	<input type="checkbox"/> Presence Statement is closed
Parallel Computing and Algorithms	lab_8	01/26/2022 12:00 - 14:00	<input type="checkbox"/>

Items per page: 3 1 – 2 of 2

Εικ.105. Κλειστή δήλωση παρουσιών

- Οι δηλώσεις παρουσιών είναι ακόμη ανοικτές απο τον καθηγητή αλλά ο φοιτητής έχει πάρει ήδη απουσία. Σε μια τέτοια περίπτωση, αν ο φοιτητής επιχειρήσει να δηλώσει παρουσία, το κουτάκι που πάτησε απενεργοποιείται και το σύστημα του παρέχει σχετικό ενημερωτικό μήνυμα. Στην παρακάτω εικόνα φαίνεται μια τέτοια περίπτωση.

Current Lectures

Search current lectures

Course	Lecture	Date & Time	Presence Statement
Distributed Systems	lab_5	01/26/2022 12:00 - 14:00	<input type="checkbox"/>
Parallel Computing and Algorithms	lab_8	01/26/2022 12:00 - 14:00	<input type="checkbox"/>

Items per page: 3 1 - 2 of 2 |< < > >|

You already have an absence for this session, so you cannot make a statement anymore

Close

Εικ.106. Αδυναμία δήλωσης παρουσίας, λόγω καταχωρημένης απουσίας

Στο σενάριο δήλωσης παρουσίας, η πιο συνηθισμένη περίπτωση είναι να εμφανίζεται μια μόνο τρέχουσα διάλεξη στον φοιτητή κάθε φορά. Το παραπάνω παράδειγμα, που φαίνονται δυο ταυτόχρονες συνεδρίες διάλεξης, χρησιμοποιήθηκε για να καλύψει τα σενάρια που μπορούν να προκύψουν ως υποπερίπτωση.

Δημιουργία αίτησης δικαιολόγησης απουσίας

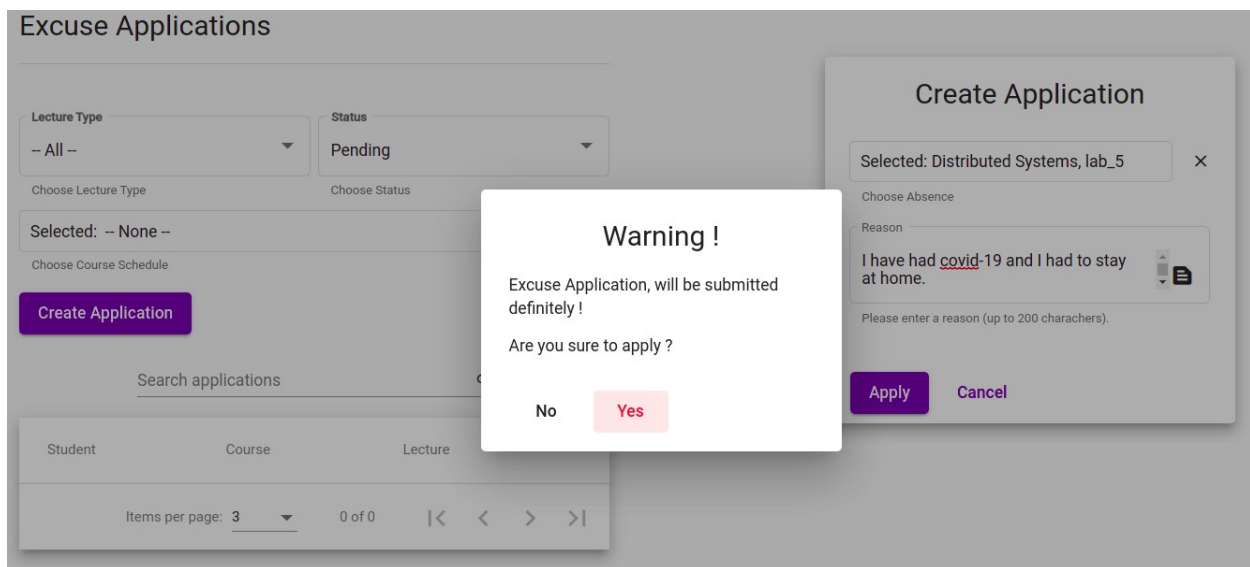
The screenshot displays the 'Excuse Applications' web interface. On the left, there are filters for 'Lecture Type' (set to '-- All --') and 'Status' (set to 'Pending'). Below these is a 'Selected: -- None --' dropdown. A purple 'Create Application' button is visible. A search bar labeled 'Search applications' is present. Below the search bar is a table with columns 'Student', 'Course', and 'Lecture'. The table shows '0 of 0' items. At the bottom of the table, it says 'Items per page: 3'. On the right, a 'Create Application' modal is open. It shows 'Selected: Distributed Systems, lab_5' and a 'Reason' text area containing 'I have had covid-19 and I had to stay at home.' The modal has 'Apply' and 'Cancel' buttons.

Εικ.107. Δημιουργία αίτησης δικαιολόγησης απουσίας

Ο φοιτητής πατώντας το κουμπί “Create Application” μπορεί να δημιουργήσει μια αίτηση δικαιολόγησης απουσίας, συμπληρώνοντας την αντίστοιχη φόρμα που εμφανίζεται δεξιά του πίνακα.

Σε αυτό το σημείο πρέπει να αναφερθεί ότι στον επιλογέα απουσιών της φόρμας δημιουργίας αίτησης, εμφανίζονται στον φοιτητή όλες οι απουσίες του για το τελευταίο 48ωρο, καθώς μόνο για αυτές επιτρέπεται να αιτηθεί δικαιολόγηση.

Στην συνέχεια πατώντας το κουμπί “Apply”, ανοίγει απο το σύστημα ένα αναδυόμενο παράθυρο, όπως φαίνεται στην παρακάτω εικόνα, για την τελική επιβεβαίωση υποβολής της δήλωσής του. Επιλέγοντας το κουμπί “Yes”, η αίτηση δικαιολόγησης για την απουσία του, υποβάλλεται στο σύστημα.



Εικ.108. Επιβεβαίωση αίτησης

Αξίζει να τονιστεί ότι ο φοιτητής μπορεί να υποβάλει μόνο μια αίτηση για κάθε απουσία που θέλει να δικαιολογήσει.

Η υποβολή αίτησης δικαιολόγησης απουσίας είναι αδύνατη στις εξής περιπτώσεις:

- Ο φοιτητής έχει ήδη υποβάλει αίτηση δικαιολόγησης για την συγκεκριμένη απουσία. Το σύστημα τον ενημερώνει με κατάλληλο μήνυμα όπως φαίνεται στην παρακάτω εικόνα.

Excuse Applications

Lecture Type
-- All --
Choose Lecture Type

Status
Pending
Choose Status

Selected: -- None --
Choose Course Schedule

Create Application

Student	Course	Lecture
Items per page: 3 0 of 0 << < > >>		

You have already made an excuse application for Distributed Systems, lab_5 absence Close

Create Application

Selected: Distributed Systems, lab_5 ×

Choose Absence

Reason
I have had covid-19 and I had to stay at home.

Please enter a reason (up to 200 characters).

Apply Cancel

Εικ.109. Ηδη υποβεβλημένη αίτηση

- Ο φοιτητής έχει ξεπεράσει το όριο των δικαιολογημένων απουσιών που του αναλογούν.

Το όριο των δικαιολογημένων απουσιών υπολογίζεται απο το σύστημα ως εξής:

Αν το σύνολο των διαλέξεων (εργαστηριακών ή θεωρητικών, αναλόγως το που απευθύνεται η απουσία κάθε φορά) του μαθήματος είναι:

- Αρτιος αριθμός, τότε διαιρούμενος με τον αριθμό δύο, μας δίνει το όριο των δικαιολογημένων απουσιών.
- Περιττός αριθμός, τότε προστιθέμενος με τον αριθμό ένα, διαιρουμένου του αποτελέσματος με το δύο και αφαιρούμενου ξανά του αριθμού ένα, έχουμε το μέγιστο όριο δικαιολογημένων απουσιών.
 - Αν ο περιττός αριθμός είναι ίσος με το 1, τότε το όριο των δικαιολογημένων απουσιών είναι ο ίδιος ο αριθμός 1.

Αν ο φοιτητής έχει εξαντλήσει το όριο των δικαιολογημένων απουσιών το σύστημα τον ενημερώνει με κατάλληλο μήνυμα όπως φαίνεται στην παρακάτω εικόνα.

The screenshot displays the 'Excuse Applications' interface. On the left, there are filters for 'Lecture Type' (set to 'All'), 'Status' (set to 'Pending'), and 'Course Schedule' (set to 'None'). A 'Create Application' button is visible. Below the filters is a table with columns 'Student', 'Course', and 'Lecture'. The table contains one entry: Student 'it21526', Course 'Introduction to C', and Lecture 'lab_8'. A search bar and pagination controls are also present. On the right, a 'Create Application' modal is open, showing the selected lecture and a text area for the reason. At the bottom, a red error message states: 'You cannot make an excuse application for Introduction to C labs since you have already excuse 4 absences !' with a 'Close' button.

Student	Course	Lecture
it21526	Introduction to C	lab_8

Εικ.110. Εξαντλημένο όριο δικαιολογημένων απουσιών

- Ο φοιτητής έχει ξεπεράσει το όριο των οριστικά αδικαιολόγητων απουσιών που του αναλογούν, δηλαδή αυτών που δεν μπορούν να δικαιολογηθούν με κάποια αίτηση.

Στο σύστημα διαχείρισης απουσιών το όριο των οριστικά αδικαιολόγητων απουσιών είναι ο αριθμός δύο.

Στην περίπτωση που ο φοιτητής έχει ήδη πάνω από δύο οριστικά αδικαιολόγητες απουσίες στις συνεδρίες διαλέξεων ενός μαθήματος, δεν μπορεί να κάνει άλλες αιτήσεις δικαιολόγησης και το σύστημα το ενημερώνει με κατάλληλο μήνυμα όπως φαίνεται στην παρακάτω εικόνα.

Excuse Applications

Lecture Type
-- All --
Choose Lecture Type

Status
Pending
Choose Status

Selected: -- None --
Choose Course Schedule

Create Application

Student	Course	Lecture
it21526	Introduction to C	lab_8

Items per page: 3 1 - 1 of 1

Create Application

Selected: Introduction to C, lab_8

Choose Absence

Reason
I could not be at lab_8 because of illness.

Please enter a reason (up to 200 characters).

Apply Cancel

You cannot make an excuse application for Introduction to C labs since you have more than 2 definitely inexcusable absences ! Close

Εικ.111. Εξάντληση ορίου αδικαιολόγητων απουσιών

ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΑΡΟΥΣΙΩΝ – ΗΛΕΚΤΡΟΝΙΚΟ ΠΑΡΟΥΣΙΟΛΟΓΙΟ / ΜΑΚΡΗΣ ΙΩΑΝΝΗΣ

256

Προβολή στοιχείων αίτησης δικαιολόγησης απουσίας

The interface is titled "Excuse Applications". It features a form with the following elements:

- Lecture Type:** A dropdown menu currently set to "All".
- Status:** A dropdown menu currently set to "Pending".
- Choose Lecture Type:** A text input field.
- Choose Status:** A text input field.
- Selected:** A dropdown menu currently set to "None".
- Choose Course Schedule:** A text input field.
- Create Application:** A purple button.
- Search applications:** A search bar with a magnifying glass icon.

Below the form is a table with the following data:

Student	Course	Lecture
it21788	Distributed Systems	lab_5

At the bottom of the table, there is a pagination bar showing "Items per page: 3" and "1 - 1 of 1".

To the right of the table is a modal window titled "Application Details" containing the following information:

- Student:** it21788
- Student's group:** group_1
- Course:** Distributed Systems
- Lecture:** lab_5
- Lecture's session date & time:** 01/26/2022 12:00 - 14:00
- Status:** Pending
- Date & time of application:** 01/26/2022 14:37:58
- Absence Reason:** I have had covid-19 and I had to stay at home.
- Cancel:** A purple button at the bottom right.

Εικ.112. Προβολή στοιχείων αίτησης δικαιολόγησης απουσίας

Ο φοιτητής πλοηγούμενος στον πίνακα με τις αιτήσεις του, μπορεί να επιλέξει μια αίτηση πατώντας την γραμμή του πίνακα στην οποία αντιστοιχεί. Έτσι έχει την δυνατότητα να δει περισσότερες λεπτομέρειες σχετικά με την αίτηση, στην καρτέλα που εμφανίζεται δεξιά του πίνακα.

Αξίζει να αναφερθεί ότι το σύστημα παρέχει στον φοιτητή μια άνετη και οργανωμένη εύρεση των αιτήσεών του, παρέχοντάς του κάποια φίλτρα αναζήτησης τα οποία μπορούν να χρησιμοποιηθούν ανεξάρτητα ή και ταυτόχρονα και είναι τα εξής:

- **Επιλογέας τύπου διάλεξης.** Ο τύπος διάλεξης μπορεί να είναι θεωρητικός ή εργαστηριακός. Έτσι ο φοιτητής μπορεί για παράδειγμα να αναζητήσει αιτήσεις του που αφορούν μόνο θεωρίες ή μόνο εργαστήρια.
- **Επιλογέας κατάστασης αίτησης.** Η κατάσταση της αίτησης μπορεί να είναι:
 - **Εκκρεμής (Pending):** Η αίτηση δικαιολόγησης δεν έχει ακόμα αξιολογηθεί από την γραμματεία.

- **Εγκεκριμένη (Approved):** Η αίτηση δικαιολόγησης έχει εγκριθεί απο την γραμματεία.
- **Απορριμμένη (Rejected):** Η αίτηση δικαιολόγησης έχει απορριφθεί απο την γραμματεία.

Ετσι ο φοιτητής μπορεί να επιλέξει να του εμφανιστούν οι αιτήσεις του με βάση μια απο τις προηγούμενες καταστάσεις απο τις οποίες μπορεί να διέλθουν.

- **Επιλογή προγράμματος μαθήματος.** Για την αποφυγή ενός αχανούς όγκου δεδομένων που αφορούν τις αιτήσεις του, ο φοιτητής μπορεί να τις πλαισιώσει ανα πρόγραμμα μαθήματος που τον ενδιαφέρει.

Σαφώς οι τρεις επιλογείς που προαναφέρθηκαν μπορούν να χρησιμοποιηθούν και συνδυαστικά, επιφέροντας μια πιο άνετη εμπειρία πλοήγησης στον φοιτητή.

Τέλος όσον αφορά τον πίνακα των αιτήσεων ο φοιτητής μπορεί να αναζητήσει αιτήσεις του μέσω της μπάρας αναζήτησης στο πάνω μέρος του πίνακα. Η αναζήτηση μπορεί να γίνει με κριτήρια το “Student”, το “Course” ή το “Lecture”.

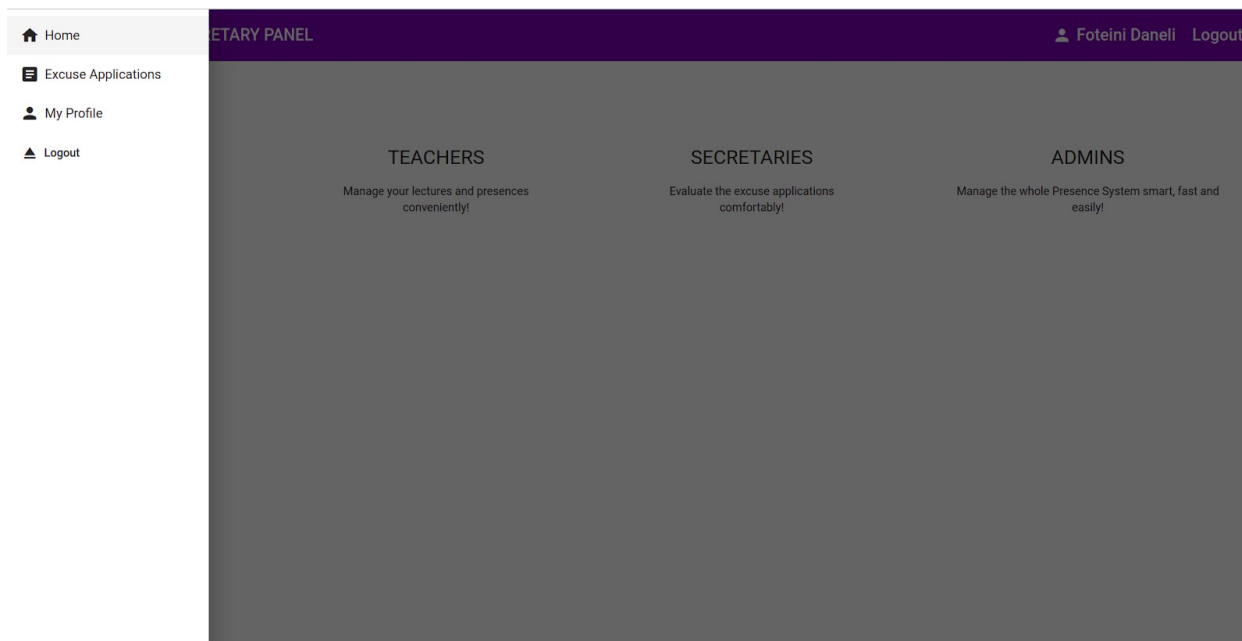
5.6. Λειτουργίες Γραμματέα (Secretary)

Στην ενότητα αυτή παρουσιάζονται τα σενάρια χρήσης για τις λειτουργίες που επιτελεί ο ρόλος του γραμματέα. Επειδή στο σύστημα διαχείρισης παρουσιών οι λειτουργίες αυτές επιτελούνται απο τον γραμματέα αλλά και απο τον διαχειριστή του συστήματος, δεν συμπεριλήφθηκαν στην ενότητα 5.3. Ετσι για πρακτικούς λόγους παρουσιάζονται σε αυτήν.

Η βασική διαφορά σε σχέση με τον διαχειριστή συστήματος είναι ότι η δράση του γραμματέα περιορίζεται στο τμήμα στο οποίο ανήκει.

Πλοήγηση στο μενού επιλογών

Ο γραμματέας αφού συνδεθεί στο σύστημα, μπορεί να δει το μενού επιλογών του και να πλοηγηθεί σε αυτό, πατώντας το κουμπί που αναπαρίσταται με τρεις οριζόντιες παύλες πάνω αριστερά στην οθόνη.



Εικ.113. Μενού Γραμματέα

Προβολή στοιχείων αίτησης δικαιολόγησης απουσίας

Excuse Applications

Lecture Type
-- All --
Choose Lecture Type

Status
Pending
Choose Status

Selected: -- None --
Choose Course Schedule

Student	Course	Lecture
it21780	Introduction to C	lab_8
it21526	Introduction to C	lab_8
it21788	Distributed Systems	lab_5

Items per page: 3 1 - 3 of 3

Application Details

Student: it21788
Student's group: group_1
Course: Distributed Systems
Lecture: lab_5
Lecture's session date & time: 01/26/2022 12:00 - 14:00
Status: Pending
Date & time of application: 01/26/2022 14:37:58

Absence Reason
I have had covid-19 and I had to stay at home.

Evaluation

Cancel

Εικ.114. Προβολή στοιχείων αίτησης δικαιολόγησης απουσίας

Είναι σημαντικό να αναφερθεί ότι ο γραμματέας μπορεί να έχει πρόσβαση στις αιτήσεις δικαιολόγησης παρουσιών που αφορούν το τμήμα του. Το σύστημα παρέχει στον γραμματέα τα ίδια ακριβώς φίλτρα αναζήτησης αιτήσεων που παρέχει και στον φοιτητή. Ο πίνακας επίσης έχει τα ίδια φίλτρα αναζήτησης αιτήσεων που αναφέρθηκαν στην ενότητα 5.5. Έτσι η πλοήγηση του γραμματέα είναι εύκολη, γρήγορη και αποτελεσματική.

Ο γραμματέας πλοηγούμενος στον πίνακα των αιτήσεων, μπορεί να επιλέξει μια αίτηση πατώντας την γραμμή του πίνακα στην οποία αντιστοιχεί. Έτσι έχει την δυνατότητα να δει περισσότερες λεπτομέρειες σχετικά με την αίτηση, στην καρτέλα που εμφανίζεται δεξιά του πίνακα, ώστε να την αξιολογήσει κατάλληλα.

Αξιολόγηση αίτησης δικαιολόγησης απουσίας (έγκριση ή απόρριψη)

The screenshot displays the 'Excuse Applications' interface. On the left, there are filters for 'Lecture Type' (set to '-- All --') and 'Status' (set to 'Pending'). Below these is a 'Selected: -- None --' dropdown. A search bar labeled 'Search applications' is positioned above a table of applications. The table has three columns: 'Student', 'Course', and 'Lecture'. It lists three applications, with the third one (Student: it21788, Course: Distributed Systems, Lecture: lab_5) highlighted. At the bottom of the table, it shows 'Items per page: 3' and '1 - 3 of 3'. On the right, an 'Evaluate Application' modal is open, featuring radio buttons for 'Approve' (selected) and 'Reject', and 'Confirm' and 'Cancel' buttons.

Student	Course	Lecture
it21780	Introduction to C	lab_8
it21526	Introduction to C	lab_8
it21788	Distributed Systems	lab_5

Εικ.115. Αξιολόγηση αίτησης δικαιολόγησης απουσίας (έγκριση ή απόρριψη)

Στην καρτέλα προβολής των στοιχείων της αίτησης ο γραμματέας μπορεί να επιλέξει το κουμπί “Evaluation” και να αξιολογήσει την αίτηση εγκρίνοντας ή απορρίπτοντάς την, στην αντίστοιχη φόρμα που εμφανίζεται απο το σύστημα. Αν ο γραμματέας εγκρίνει την αίτηση, τότε η απουσία καταχωρείται πλέον ως δικαιολογημένη, ενώ αν την απορρίψει τότε η απουσία παραμένει αδικαιολόγητη.

Ο γραμματέας μπορεί να αξιολογήσει την αίτηση του φοιτητή εντός 30 ημερών απο την υποβολή της. Διαφορετικά η αίτηση καταχωρείται ως απορριμμένη και η απουσία που σχετίζεται με εκείνη παραμένει αδικαιολόγητη.

Στην περίπτωση που ο γραμματέας επιχειρήσει την αξιολόγηση μιας αίτησης μετα το πέρας του χρονικού ορίου των 30 ημερών κατάθεσης της αίτησης, το σύστημα τον ενημερώνει με κατάλληλο μήνυμα, όπως φαίνεται στην παρακάτω εικόνα.

Excuse Applications

Lecture Type
-- All --
Choose Lecture Type

Status
-- All --
Choose Status

Selected: -- None --
Choose Course Schedule

Student	Course	Lecture	Status
it21526	Java I	lab_1	Rejected
it21569	Introduction to C	lab_8	Approved
it21780	Introduction to C	lab_8	Pending
it21526	Introduction to C	lab_8	Pending
it21788	Distributed Systems	lab_5	Pending

Items per page: 5 1 - 5 of 5

Evaluate Application

☒ Approve ☐ Reject

Confirm

Cancel

You can evaluate every excuse application within 30 days of the statement

Close

Εικ.116. Αδυναμία αξιολόγησης αίτησης μετα απο 30 μέρες

Τέλος πρέπει να αναφερθεί το γεγονός ότι ο καθηγητής εντός των 30 ημερών απο την κατάθεση της αίτησης, μπορεί να αλλάξει την αξιολόγησή του ως προς την αίτηση, δηλαδή μπορεί να εγκρίνει μια αίτηση που πριν είχε απορρίψει και το αντίθετο.

Κεφάλαιο 6: Σύνοψη

Στο παρόν κεφάλαιο παρατίθενται συνοπτικά τα βασικά συμπεράσματα, καθώς και κάποιες προοπτικές εξέλιξης για το σύστημα διαχείρισης παρουσιών.

6.1. Βασικά Συμπεράσματα

Η εφαρμογή διαχείρισης παρουσιών αναπτύχθηκε με κύριο σκοπό την διευκόλυνση των ακαδημαϊκών μελών του Χαροκοπέιου πανεπιστημίου, περιορίζοντας το φαινόμενο της γραφειοκρατίας, στο πλαίσιο της καταγραφής και δικαιολόγησης των παρουσιών των φοιτητών.

Το σύστημα διαχείρισης παρουσιών θέτει ως προτεραιότητα τον ανθρώπινο παράγοντα και τις ιδιαιτερότητες του. Ως εκ τούτου για κάθε χρήστη του συστήματος παρέχονται δυνατότητες προσαρμοσμένες στο ποιόν του ρόλου του, καθώς και ευελιξίες οι οποίες ίσως θα ήταν αδύνατες σε γραφειοκρατικό επίπεδο.

Για παράδειγμα, είναι στην βούληση του καθηγητή αν θα βάλει απουσία σε έναν δικαίως αργοπορημένο φοιτητή ή όχι, καθώς του δίνεται η δυνατότητα να μεταβάλλει την απόφασή του εύκολα και γρήγορα, ενώ ο φοιτητής σε περίπτωση που λάβει απουσία μπορεί να κάνει αίτηση δικαιολόγησης άμεσα. Απο την άλλη πλευρά ο γραμματέας μπορεί να αξιολογήσει την αίτηση, αλλά και να αλλάξει την αξιολόγησή του ως προς αυτήν, εντός 30 ημερών απο την υποβολή της.

Παράλληλα δίνεται η δυνατότητα καταγραφής των παρουσιών και για θεωρητικές και για εργαστηριακές διαλέξεις ενός μαθήματος, παρέχοντας έτσι χρήσιμα δεδομένα που μπορούν για παράδειγμα να αξιοποιηθούν για στατιστικά συμπεράσματα σχετικά με την προσέλευση των φοιτητών στις διαλέξεις. Επίσης ακόμα κι αν μια διάλεξη πρέπει να επαναληφθεί

περισσότερες απο μια φορές για ξεχωριστές ομάδες φοιτητών, το σύστημα παρέχει και πάλι την δυνατότητα καταγραφής των παρουσιών τους.

Οι ευελιξίες αυτές συμβάλλουν στην έγκαιρη, συνεπή και διαφανή απόδοση παρουσιών ή απουσιών για τους φοιτητές και καθιστούν πιο άμεση και εύκολη την δικαιολόγησή τους σε σχέση με το ισχύον παραδοσιακό μη ψηφιοποιημένο σύστημα.

Σε συνολικό επίπεδο το σύστημα διαχείρισης παρουσιών, με την παροχή μιας συνεπούς και άμεσης καταγραφής των παρουσιών των φοιτητών, τους ενισχύει το πνεύμα της αφοσίωσης και της πειθαρχίας και τους βοηθά να συνειδητοποιήσουν πόσο καταλυτικό ρόλο έχουν αυτές οι αρετές σε όλη την διαδικασία της περάτωσης των σπουδών τους.

Οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίησή του και συγκεκριμένα το Angular Framework για το client side και το Spring Boot για το server side, βοήθησαν στην δημιουργία μιας συνολικά εύχρηστης εφαρμογής, με τρόπο που να παρέχει ευχάριστη εμπειρία στους χρήστες.

6.2. Προοπτικές εξέλιξης του Συστήματος

Το σύστημα διαχείρισης παρουσιών, παρέχει πολλά οφέλη και διευκολύνσεις σε όλα τα εμπλεκόμενα μέλη που σχετίζονται με αυτό. Επίσης έχει αναπτυχθεί με τέτοιο τρόπο ώστε να είναι εύκολα και γρήγορα επεκτάσιμο, καθώς κάθε λογισμικό πάντα επιδέχεται βελτιώσεις και εξελίξεις. Με γνώμονα αυτό το γεγονός, παρακάτω παρουσιάζονται κάποιες πιθανές προοπτικές εξέλιξης του συστήματος διαχείρισης παρουσιών.

Εκ πρώτης όψεως, θα ήταν αρκετά σημαντική η σύνδεση της εφαρμογής με τον κατάλογο χρηστών LDAP, καθώς και το ανέβασμά της στον server του Χαροκοπέιου Πανεπιστημίου. Η σύνδεση με τον κατάλογο χρηστών LDAP σε συνδυασμό με το ανέβασμα του συστήματος διαχείρισης παρουσιών στον server της σχολής, καθιστά το σύστημα εφαρμόσιμο στην πράξη για το πανεπιστήμιο.

Επιπλέον στο γενικό πλαίσιο της εφαρμογής θα ήταν καλή ιδέα η προσθήκη ενός επιπλέον ρόλου ως υπεύθυνου τμήματος (Department Manager) για κάθε τμήμα, κάθε σχολής του πανεπιστημίου. Ο υπεύθυνος τμήματος θα είναι επιφορτισμένος με τα καθήκοντα του διαχειριστή του συστήματος (Admin) με την διαφορά ότι η δράση του θα περιορίζεται στο τμήμα στο οποίο ανήκει. Έτσι ο φόρτος εργασίας του διαχειριστή του συστήματος, θα μοιράζεται και μεταξύ των υπευθύνων των τμημάτων.

Παράλληλα όσον αφορά την λειτουργία δικαιολόγησης απουσίας, μια απαραίτητη προοπτική εξέλιξης θα ήταν το ανέβασμα ενός δικαιολογητικού αρχείου pdf απο τον φοιτητή. Το αρχείο αυτό θα πιστοποιεί τον λόγο απουσίας του φοιτητή και θα είναι χρήσιμο για την αξιολόγηση της αίτησής του απο την γραμματεία.

Επιπροσθέτως στο κομμάτι που έχει να κάνει με την αξιολόγηση της αίτησης του φοιτητή, το σύστημα θα μπορούσε να τον ενημερώνει αυτόματα για την έκβασή της με email.

Επίσης η ασφάλεια και η διαχείριση χρηστών του συστήματος μπορούν να αναβαθμιστούν περεταίρω στην εφαρμογή με την χρήση της τεχνολογίας Keycloak σε συνδυασμό με το OAuth2 και το Jwt. Το Keycloak απλοποιεί την διαδικασία της ασφάλειας και της αυθεντικοποίησης των χρηστών παρέχοντας πληθώρα διευκολύνσεων σε επίπεδο ανάπτυξης του λογισμικού.

Τέλος θα ήταν χρήσιμο να προστεθούν test cases σε όλη την έκταση της εφαρμογής, για την καλύτερη διαχείριση των σφαλμάτων και την αποτελεσματικότερη και ταχύτερη ανάπτυξη των επεκτάσεων. Εξ' άλλου όσο ένα σύστημα γίνεται μεγαλύτερο, τόσο μεγαλώνουν και οι ανάγκες για μια καλύτερη και ποιοτικότερη διαδικασία ελέγχου.

Κεφάλαιο 7: Επίλογος

Η παρούσα πτυχιακή εργασία αφορά την ανάλυση, την σχεδίαση και την ανάπτυξη ενός συστήματος διαχείρισης παρουσιών για τους φοιτητές. Κύριος σκοπός του συστήματος είναι ο περιορισμός της γραφειοκρατίας και η διευκόλυνση των εμπλεκόμενων χρηστών, μέσα από την ψηφιοποίηση της όλης διαδικασίας με τρόπο αποδοτικό και φιλικό προς αυτούς.

Το σύστημα διαχείρισης παρουσιών βασίζεται στο μοντέλο client-server και χωρίζεται σε δύο ανεξάρτητες μεταξύ τους εφαρμογές. Η μια είναι ανεπτυγμένη για το frontend με το Angular Framework και η άλλη είναι ανεπτυγμένη για το backend με το Spring Boot του Spring Framework. Ωστόσο αυτές οι τεχνολογίες συνδιάστηκαν αρμονικά με διάφορες βιβλιοθήκες προκειμένου να καλύψουν τον συνολικό σκοπό του συστήματος.

Η περάτωση αυτής της πτυχιακής εργασίας, μου προσέδωσε πολύτιμη εμπειρία γύρω από αρκετά θέματα που αφορούν την ανάλυση, την σχεδίαση και την υλοποίηση μιας web εφαρμογής στο σύνολό της.

Επίσης μέσα από αυτή την διαδικασία έμαθα να διαχειρίζομαι τον χρόνο και την ενέργειά μου κατάλληλα, ώστε να επιφέρω το μέγιστο δυνατό αποτέλεσμα σε κάθε νέα πρόκληση. Παράλληλα, κατάλαβα πόσο σημαντική είναι η διαδικασία ιεράρχησης των προτεραιοτήτων συναρτήσει των παραγόντων του χρόνου και της ποιότητας, στην ανάπτυξη μιας εφαρμογής.

Το σημαντικότερο όμως για εμένα είναι ότι μέσα από την προσπάθεια που κατέβαλα για την ολοκλήρωση της παρούσας πτυχιακής εργασίας, συνηδαιτοποίησα, ακόμα περισσότερο, πόσο καθοριστικές είναι οι αρετές της υπομονής και της πειθαρχίας καθώς και η επιμονή και προσήλωση που χρειάζεται για την εκπλήρωση κάθε μεγάλου στόχου.

BIBΛΙΟΓΡΑΦΙΑ

- [1] Why Spring? <https://spring.io/why-spring>
- [2] OVERVIEW. Spring Framework. <https://spring.io/projects/spring-framework>
- [3] OVERVIEW. Spring Boot. <https://spring.io/projects/spring-boot>
- [4] Reducing boilerplate code with project lombok.
<https://objectcomputing.com/resources/publications/sett/january-2010-reducing-boilerplate-code-with-project-lombok>
- [5] Project Information. Project Summary. <https://commons.apache.org/proper/commons-csv/summary.html>
- [6] Using Apache Commons Csv. <https://commons.apache.org/proper/commons-csv/>
- [7] Class CSVFormat.
<https://commons.apache.org/proper/commons-csv/apidocs/org/apache/commons/csv/CSVFormat.html>
- [8] Overview. POI-HSSF and POI-XSSF/SXSSF - Java API To Access Microsoft Excel Format Files.
<https://poi.apache.org/components/spreadsheet/index.html>
- [9] How to use the HSSF API. General Use. User API (HSSF and XSSF).
https://poi.apache.org/components/spreadsheet/how-to.html#user_api
- [10] Java Bean Validation Basics. <https://www.baeldung.com/javax-validation>
- [11] Matthew Tyson (2019). “What is Jpa?” Introduction to the Java Persistence API.
<https://www.infoworld.com/article/3379043/what-is-jpa-introduction-to-the-java-persistence-api.html>
- [12] OVERVIEW. Spring Data JPA. <https://spring.io/projects/spring-data-jpa>
- [13] About JPQL Queries. Understanding EclipseLink.
<https://www.eclipse.org/eclipselink/documentation/2.5/concepts/queries002.htm>
- [14] Thorben Janssen (χ.χ). JPQL. How to Define Queries in JPA and Hibernate .
<https://thorben-janssen.com/jpql/>
- [15] Named Queries. JPA-JPQL. https://www.tutorialspoint.com/jpa/jpa_jpql.htm
- [16] Spring Data JPA @Query. <https://www.baeldung.com/spring-data-jpa-query#1-jpql>
- [17] Eager and Lazy Loading. JPA-JPQL. https://www.tutorialspoint.com/jpa/jpa_jpql.htm
- [18] Hibernate. <https://www.javatpoint.com/hibernate-tutorial>

- [19] What Does Object-Relational Mapping (ORM) Mean? Object-Relational Mapping (ORM). <https://www.techopedia.com/definition/24200/object-relational-mapping--orm>
- [20] Hibernate ORM. <https://hibernate.org/orm/>
- [21] Hibernate Architecture. Hibernate. <https://www.javatpoint.com/hibernate-architecture>
- [22] Spring Boot Architecture. Spring Boot. <https://www.javatpoint.com/spring-boot-architecture>
- [23] Ramesh Fadatare (χ.χ). JPA Architecture. <https://www.javaguides.net/2018/12/jpa-architecture.html>
- [24] MySQL Tutorial. MySQL. <https://www.javatpoint.com/mysql-tutorial>
- [25] OVERVIEW. Spring Security. <https://spring.io/projects/spring-security>
- [26] Introduction. Spring Security. <https://www.javatpoint.com/spring-security-introduction>
- [27] Hibernate Validator. HIBERNATE. <https://hibernate.org/validator/>
- [28] Introduction to JSON Web Tokens. <https://jwt.io/introduction>
- [29] Chinmayee Deshpande (2022). What is Angular?: Architecture, Features, and Advantages. <https://www.simplilearn.com/tutorials/angular-tutorial/what-is-angular>
- [30] What is Angular Module? <https://angular-training-guide.rangle.io/modules/introduction>
- [31] Understanding Angular. <https://angular.io/guide/understanding-angular-overview>
- [32] Angular templates and views. <https://howtodoinjava.com/angular/angular-templates-and-views/>
- [33] Anastasiia Shybeko (χ.χ). 8 ADVANTAGES OF ANGULAR FOR BUSINESSES AND DEVELOPERS. <https://light-it.net/blog/8-advantages-of-angular-for-businesses-and-developers/>
- [34] What is IVY? All About Angular Engine Ivy In 5 Mins. <https://www.angularminds.com/blog/article/what-is-angular-ivy.html>
- [35] Angular CLI. <https://angular-training-guide.rangle.io/cli>
- [36] The RxJS library. <https://angular.io/guide/rx-library>
- [37] OVERVIEW. RxJS. <https://rxjs.dev/guide/overview#purity>
- [38] Observer. Behavioral Patterns. <https://refactoring.guru/design-patterns/observer>
- [39] Introduction. Functional Programming Paradigm. <https://www.geeksforgeeks.org/functional-programming-paradigm/>
- [40] What Does Collection Mean? Collection. <https://www.techopedia.com/definition/25317/collection>
- [41] Advantages and Disadvantages of RxJS. RxJS. <https://www.javatpoint.com/advantages-and-disadvantages-of-rxjs>

- [42] RxJS: Reactive Extensions for JavaScript. <https://www.npmjs.com/package/rxjs>
- [43] Angular Material Tutorial. <https://www.javatpoint.com/angular-material>
- [44] Angular Material. <https://material.angular.io/>
- [45] ngx-mat-timepicker. <https://www.npmjs.com/package/ngx-mat-timepicker>
- [46] CSS Flexible Box Layout.
https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout
- [47] What Does Media Query Mean? Media Query.
<https://www.techopedia.com/definition/30449/media-query-responsive-web-design>
- [48] angular / flex-layout. <https://github.com/angular/flex-layout>
- [49] Getting Started with Angular Flex-Layout (2020).
<https://www.excellerate.com/blogs/getting-started-with-angular-flex-layout/>
- [50] Advantages of Angular flex layout. Angular Wiki.
<https://www.angularjswiki.com/flexlayout/>
- [51] Duncan Faulkner (2021). Angular Flex-Layout: The Alternative Layout Library for Flex-box and CSS Grid. <https://medium.com/ngconf/angular-flex-layout-ddf1c8fad37e>
- [52] Working with Moment.js Date Libraries (2020). <https://www.section.io/engineering-education/nodejs-date-and-time-objects-with-moment/>
- [53] Kirtesh Shah (2020). Introduction To Json Web Token (JWT). <https://www.c-sharpcorner.com/article/introduction-of-json-web-token-jwt/>
- [54] Requirements Analysis - Understanding the Process & Techniques (2018).
<https://reqtest.com/requirements-blog/requirements-analysis/>
- [55] What is Unified Modeling Language (UML)?
<https://www.visual-paradigm.com/guide/uml-unifiedmodeling-language/what-is-uml/>

ΑΠΟΘΕΤΗΡΙΑ ΚΩΔΙΚΑ

Server Side Repository: <https://github.com/john-makris/pms-backend>

Client Side Repository: <https://github.com/john-makris/pms-frontend>