



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ  
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ  
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Πτυχιακή εργασία

**Ανάπτυξη εφαρμογής Android για τους φοιτητές του  
τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών  
Υπολογιστών του Πανεπιστημίου Πελοποννήσου.**

**Ηλιού Δημήτριος**  
2442

Επιβλέπων:  
ΧΡΙΣΤΟΔΟΥΛΟΥ ΣΩΤΗΡΙΟΣ  
< Επίκουρος Καθηγητής >

Συνεπιβλέπων:  
< ΤΖΗΜΑΣ ΙΩΑΝΝΗΣ >  
< Αναπληρωτής Καθηγητής >

< ΠΑΡΑΣΚΕΥΑΣ ΜΙΧΑΗΛ >  
< Αναπληρωτής Καθηγητής >

Πάτρα, < Μάιος, 2021 >

## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

<b>ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ .....</b>	<b>4</b>
<b>ΠΡΟΛΟΓΟΣ.....</b>	<b>5</b>
<b>Περίληψη στα Ελληνικά.....</b>	<b>6</b>
<b>English Abstract .....</b>	<b>8</b>
<b>ΕΥΧΑΡΙΣΤΙΕΣ.....</b>	<b>9</b>
<b>ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ.....</b>	<b>10</b>
<b>ΚΕΦΑΛΑΙΟ 1: Παρόμοιες και Αυτούσιες Εφαρμογές .....</b>	<b>11</b>
1.1 Αντίστοιχες Εφαρμογές Πανεπιστημίων .....	11
1.2 Αυτούσιες Εφαρμογές .....	12
<b>ΚΕΦΑΛΑΙΟ 2: Σχεδιασμός .....</b>	<b>13</b>
2.1. Σχεδιασμός της εφαρμογής .....	13
2.1.1. Ανακοινώσεις .....	15
2.1.2. Καθηγητές .....	17
2.1.3. Κοινότητα .....	18
2.1.4. Αντιστοιχίσεις Μαθημάτων .....	19
2.1.5. Χάρτης .....	20
2.1.6. Ημερολόγιο .....	21
2.1.7. Σχετικά με & Επιλογές.....	22
2.2. Σχεδιασμός της Βάσης Δεδομένων.....	22
<b>ΚΕΦΑΛΑΙΟ.3: Διάταξη και Γραφικά Εφαρμογής .....</b>	<b>25</b>
3.1. Πρώτη Εκτέλεση.....	25
3.2 Μενού Χρήστη & Ανακοινώσεις Τμήματος .....	26
3.3 Καθηγητές .....	27
3.4 Κοινότητα .....	28
3.5 Αντιστοιχίσεις .....	30
3.6 Χάρτης & Ακαδημαϊκό Ωρολόγιο.....	32
3.7 Σχετικά & Προτιμήσεις Χρήστη.....	34
<b>ΚΕΦΑΛΑΙΟ 4: Παρουσίαση Κώδικα .....</b>	<b>35</b>
4.1 Κλάσης.....	35
4.2 Activity .....	36
4.3 Fragments .....	38
4.4 RecyclerView & Adapters.....	39
4.5 Threads & AsyncTask .....	40
4.6 JobService.....	41
4.7 Σύνδεση με τον Ιστότοπο και Προσπέλαση XML .....	42
4.7.1 ArrayList & Hashmap.....	43
4.8 Χρήση Constructor .....	46

4.9 Δημιουργία τοπικής Βάσης Δεδομένων .....	47
4.9.1 Απόσπαση Δεδομένων .....	48
4.9.2 Τροποποίηση Δεδομένων .....	49
<b>ΚΕΦΑΛΑΙΟ 5: Μελλοντικές Επεκτάσεις .....</b>	<b>50</b>
5.1 Βαθμολογίες .....	50
5.2 Αιτήσεις και Δηλώσεις Μαθημάτων.....	50
5.3 Προσαρμοσμένο ωρολόγιο πρόγραμμα .....	50
5.4 Βιβλιοθήκη .....	51
<b>ΥΠΟΣΗΜΕΙΩΣΗ .....</b>	<b>52</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ .....</b>	<b>109</b>

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

- Τ.Ε = Τεχνολογικής Εκπαίδευσης
- CIED = Τμήμα Μηχανικών Πληροφορικής Τ.Ε. , ΤΕΙ Δυτικής Ελλάδας
- ELE = Τμήμα Ηλεκτρολόγων Μηχανικών Τ.Ε. , ΤΕΙ Δυτικής Ελλάδας
- ΗΜΜΥ = Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών πανεπιστήμιου Πελοποννήσου

## ΠΡΟΛΟΓΟΣ

Στόχος αυτής της πτυχιακής εργασίας ήταν η δημιουργία μιας εφαρμογής για κινητές συσκευές android με σκοπό την χρήση της από τους φοιτητές του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, αλλά και των πρώην τμημάτων των Μηχανικών Πληροφορικής Τ.Ε ΚΑΙ Ηλεκτρολόγων Μηχανικών Τ.Ε.

Ένα από τα σημαντικά προβλήματα το οποίο λύνεται με την βοήθεια της εφαρμογής έχει να κάνει με την τακτική ενημέρωση των φοιτητών σχετικά με νέες ανακοινώσεις του ιδρύματος. Οι φοιτητές πλέον, αφότου κατεβάσουν στην κινητή τους συσκευή την εφαρμογή, θα μπορούν να διαβάζουν ενημερώσεις αλλά και να λαμβάνουν ειδοποιήσεις μεσώ αυτής κάθε φορά που κατι καινούργιο δημοσιεύεται στην επίσημη ιστοσελίδα της σχολής. Έτσι οι σπουδαστές δεν θα χάνουν κρίσιμες ανακοινώσεις που έχουν να κάνουν κυρίως με δηλώσεις μαθημάτων, δηλώσεις συγγραμμάτων, ωρολόγιο πρόγραμμα κ.α.

## Περίληψη στα Ελληνικά

Η εφαρμογή υλοποιήθηκε με την χρήση του ολοκληρωμένου προγραμματιστικού περιβάλλοντος “Android studio” που είναι βασισμένο στην οικογένεια λογισμικού της «JetBrains IntelliJ IDEA» και γράφτηκε με την αντικειμενοστραφής γλώσσα προγραμματισμού «JAVA». Παράλληλα χρησιμοποιήθηκε το σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων “SQLite» και της γλώσσα διαχείρισης σχεσιακών βάσεων δεδομένων «SQL» για την αποθήκευση απαραίτητων δεδομένων.

Ο χρήστης στην πρώτη χρήση της εφαρμογής του ζητείται να επιλέξει το πρόγραμμα σπουδών που παρακολουθεί αναμεσα από τα εξής «HMMY» «CIED» «ELE». Η διαφορά μεταξύ αυτών είναι μια λειτουργία. Για τους φοιτητές που παρακολουθούν τα προγράμματα σπουδών Μηχανικών Πληροφορικής (CIED) και Ηλεκτρολόγων Μηχανικών (ELE) τους παρέχεται η δυνατότητα να αναζητήσουν τις αντιστοιχίες μαθημάτων τους με αυτών του καινούργιου προγράμματος σπουδών των «Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών». Οι υπόλοιπες λειτουργίες απευθύνονται σε όλους τους φοιτητές ανεξάρτητα τι πρόγραμμα παρακολουθούν.

Ένα από τα σημαντικά προβλήματα το οποίο λύνεται με την βοήθεια της εφαρμογής έχει να κάνει με την τακτική ενημέρωση των φοιτητών σχετικά με νέες ανακοινώσεις του ιδρύματος. Οι φοιτητές πλέον, αφότου κατεβάσουν στην κινητή τους συσκευή την εφαρμογή, θα μπορούν να διαβάζουν ενημερώσεις αλλά και να λαμβάνουν ειδοποιήσεις μέσω αυτής κάθε φορά που κάτι καινούργιο δημοσιεύεται στην επίσημη ιστοσελίδα της σχολής. Έτσι οι σπουδαστές δεν θα χάνουν κρίσιμες ανακοινώσεις που έχουν να κάνουν κυρίως με δηλώσεις μαθημάτων, δηλώσεις συγγραμμάτων, ωρολόγιο πρόγραμμα κ.α.

Κάποιες άλλες λειτουργίες είναι η αναζήτηση πληροφοριών σχετικά με τους διδάσκοντες, χάρτης της πανεπιστημιούπολης, πληροφορίες σχετικά με το ακαδημαϊκό ημερολόγιο και του ωρολογίου προγράμματος, πληροφορίες των σπουδαστικών συλλόγων και ομάδες αυτών στα κοινωνικά δίκτυα.



## English Abstract

The aim of this thesis is to develop an application for android mobile devices in order to be used by the students of the Department of Electrical and Computer Engineering, but also the former departments of Computer Engineering and Electrical Engineering.

The application was implemented using the integrated programming environment "Android studio" which is based on the software family of "JetBrains IntelliJ IDEA" and was written in the object-oriented programming language "JAVA". At the same time, the "SQLite" relational database management system and the "SQL" relational database management language were used to store all the necessary data.

When the user install and open for the first time the application, he is asked to choose the study direction he attends among the following, "HMMY", "CIED" and "ELE". The difference between them is one function. Students attending the Computer Engineering (CIED) and Electrical Engineering (ELE) curriculum are given the opportunity to look for their course equivalents with those of the new "Electrical & Computer Engineering" curriculum. The rest of the functions are addressed to all students regardless of what direction they choose to follow.

One of the important issues that is solved with the help of this application has to do with the regular information of the students about new announcements and news of the institution. Students after downloading the application on their mobile device, will be able to read updates and receive notifications through the app every time something new is published on the official website of the department. This way students will not miss critical announcements that have to do mainly with course statements, timetable etc.

Some other functions have to do with, the directory of professors, the map of the campus, information about the academic calendar and the schedule, information of the student associations and their groups on the social networks.



## ΕΥΧΑΡΙΣΤΙΕΣ

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κύριο Χριστοδούλου Σωτήριο, για την εμπιστοσύνη και το ενδιαφέρον του κατά την ανάθεση της εργασίας, την πολύτιμη βοήθεια και συνεργασία καθ' όλη τη διάρκεια της εκπόνησης της διπλωματικής εργασίας, δίνοντάς μου τη δυνατότητα να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου για την υποστήριξη καθ' όλη τη διάρκεια των ακαδημαϊκών μας σπουδών αλλά και την ηθική συμπαράσταση κατά τη διάρκεια της συγγραφής της πτυχιακής αυτής εργασίας.

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1. Έναρξη Λειτουργίας.....	13
Εικόνα 2. Λειτουργίες και Δυνατότητες της Εφαρμογής.....	15
Εικόνα 3. Ανάλυση Λειτουργίας Ανακοινώσεων .....	16
Εικόνα 4. Ανάλυση Λειτουργίας «Καθηγητές» .....	17
Εικόνα 5. Ανάλυση Λειτουργίας «Κοινότητα» .....	18
Εικόνα 6. Ανάλυση Λειτουργίας «Αντιστοιχήσεις Μαθημάτων» .....	19
Εικόνα 7. Ανάλυση Λειτουργίας «Χάρτης» .....	20
Εικόνα 8. Ανάλυση Λειτουργίας «Ακαδημαϊκό Ημερολόγιο» .....	21
Εικόνα 9. Ανάλυση Λειτουργίας «Σχετικά με» & «Επιλογές».....	22
Εικόνα 10. Πίνακες Βάσης Δεδομένων .....	23
Εικόνα 11. Αρχική Επιλογή .....	25
Εικόνα 12. Κατηγορία «Ανακοινώσεις» .....	26
Εικόνα 13. Κατηγορία «Καθηγητές» .....	28
Εικόνα 14. Κατηγορία «Κοινότητα» .....	29
Εικόνα 15. Κατηγορία Αντιστοιχίσεις CIED & ELE .....	30
Εικόνα 16. Παράδειγμα Μαθήματος Με και Χωρίς Αντιστοίχιση.....	31
Εικόνα 17. Κατηγορία «Χάρτης» .....	32
Εικόνα 18. Ακαδημαϊκό Ημερολόγιο .....	33
Εικόνα 19. Κατηγορίες «Σχετικά» & «Επιλογές» .....	34
Εικόνα 20. Έλεγχος Πρώτης Εκτέλεσης .....	38
Εικόνα 21. Παράδειγμα JobService.....	42
Εικόνα 22. Τρόπος Σύνδεσης με Ιστότοπο .....	43
Εικόνα 23. Παράδειγμα XML.....	44
Εικόνα 24. ArrayList<HashMap<String,String>>.....	45
Εικόνα 25. Προσπέλαση Ανακοινώσεων .....	45
Εικόνα 26. onPostExecute .....	46
Εικόνα 27. Παράδειγμα Constructor.....	47
Εικόνα 28. Παράδειγμα Προσπέλασης Δεδομένων του Πίνακα “CIED_TEI_COURSES_TABLE” ..	48
Εικόνα 29. fillCourse() .....	49
Εικόνα 30. Παράδειγμα εισαγωγής στοιχείων στο πίνακα “RSS_TABLE” .....	49

## ΚΕΦΑΛΑΙΟ 1: Παρόμοιες και Αυτούσιες Εφαρμογές

### 1.1 Αντίστοιχες Εφαρμογές Πανεπιστημίων

Πανεπιστήμια από όλο τον κόσμο έχουν αναπτύξει εφαρμογές με σκοπό την καλύτερη εξυπηρέτηση και ενημέρωση των φοιτητών τους. Πολλά από τα ξένα ιδρύματα παρέχουν αρκετές παρόμοιες δυνατότητες. Η πιο συνηθισμένη είναι η προβολή ανακοινώσεων τόσο του πανεπιστημίου όσο και του κάθε τμήματος που στεγάζει. Συνήθως η ροή είναι φιλική προς τον χρήστη και περιέχει όλα τα τελευταία νέα και εξελίξεις τα οποία μπορούν να φανούν χρήσιμα προς τον φοιτητή.

Μια άλλη κατηγορία είναι οι εκδηλώσεις όπου η εφαρμογή είναι συνδεδεμένη με το ημερολόγιο του Πανεπιστημίου και προβάλλει καθημερινές εκδηλώσεις και συμβάντα. Σε αυτή την κατηγορία επίσης ο χρήστης έχει την δυνατότητα να μπορεί να δει το προσωπικό ακαδημαϊκό του ημερολόγιο και τις αργίες που ακολουθούν.

Για τους σπουδαστές που επισκέπτονται συχνά την βιβλιοθήκη των πανεπιστημίων τους παρέχεται η δυνατότητα μέσα από το κινητό να ψάξουν συγγράμματα, να δουν την διαθεσιμότητα τους και να κάνουν κράτηση στο όνομα τους. Σε κάποιες από τις εφαρμογές δίνετε η δυνατότητα στους χρήστες να αναζητήσουν διαθέσιμους υπολογιστές όπου το πανεπιστήμιο παρέχει δωρεάν μέσα στη βιβλιοθήκη.

Για το εστιατόριο κάθε σχολής δίνεται η δυνατότητα στο χρήστη να διαβάσει το ημερήσιο ή/και το εβδομαδιαίο μενού. Μέσα από αυτή την δραστηριότητα συχνά παρουσιάζονται το ωράριο λειτουργίας, το μενού αλλά και σε κάποιες περιπτώσεις, ανάλυση συστατικών κάθε γεύματος με δυνατότητα από τον χρήστη την εφαρμογή φίλτρων ανάλογα αν την διατροφή που ακολουθεί.

Η ηλεκτρονική γραμματεία είναι η πιο συχνή υπηρεσία όπου παρατηρήθηκε σχεδόν σε όλες τις εφαρμογές που προ-μελετήθηκαν. Ο χρήστης μέσα από το κινητό του μπορεί να δει την συνολική του πρόοδο, να ενημερωθεί για νέες βαθμολογίες, το μέσο ορό του και ποσά μαθήματα του υπολείπονται. Επιπροσθέτως μέσω της ηλεκτρονικής γραμματείας ο σπουδαστής μπορεί να κάνει αίτηση για χορήγηση εγγραφών και δικαιολογητικών όπως π.χ. έκδοση μίας βεβαίωσης σπουδών.

Εκτός από τις παραπάνω λειτουργίες, υπάρχουν και κάποιες άλλες που δεν συμπεριλαμβάνονται σε όλες τις πανεπιστημιακές εφαρμογές. Κάποιες από αυτές είναι η προβολή του χάρτη της πανεπιστημιούπολης, όπου συχνά υπάρχει μια λίστα με όλα τα κτήρια, τις αίθουσες που στεγάζουν και έναν χάρτη όπου με μια πινέζα δείχνει την τοποθεσία των κτηρίων αυτών. Άλλες λειτουργίες είναι, να έχει ο φοιτητής πρόσβαση σε ένα κατάλογο με όλους τους καθηγητές και τους τρόπους επικοινωνίας με κάθε έναν από αυτούς και την δυνατότητα την κοινοποιήσεις, παρέχοντας στους φοιτητές συνδέσμους από όλα τα κοινωνικά δίκτυα για την εύκολη δικτύωση τους.

## 1.2 Αυτούσιες Εφαρμογές

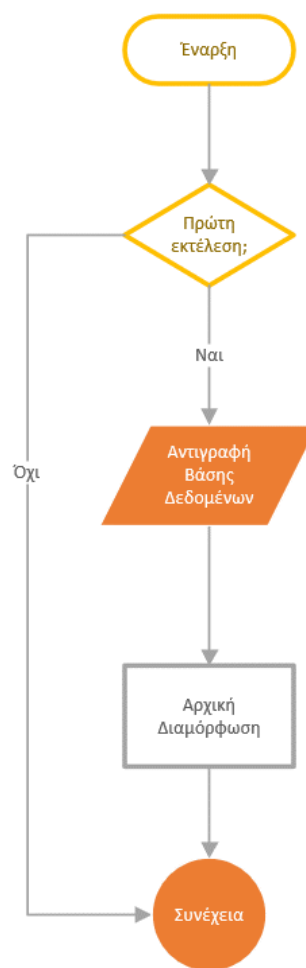
Η εφαρμογή που αναπτύχθηκε στα πλαίσια της πτυχιακής εργασίας απαιτεί κάποιες αυτούσιες εφαρμογές έτσι ώστε να εξασφαλισθεί η σωστή λειτουργία της. Οι εφαρμογές αυτές είναι:

- Πρόγραμμα πλοήγησης στο διαδίκτυο, όπου χρησιμοποιείται για το άνοιγμα των ιστοσελίδων που καλούνται μέσα από αυτή. Αν και συνήθως το προεπιλεγμένο πρόγραμμα στις κινητές συσκευές android είναι το Google Chrome, ο χρήστης μπορεί να αλλάξει και να διαλέξει ως κύριο οποιοδήποτε περιηγητή για το σύστημα του χωρίς να υπάρξει πρόβλημα.
- Πρόγραμμα αποστολής και λήψης Email, όπου χρησιμοποιείται για την αποστολή και προβολή email προς τους καθηγητές. Και εδώ ισχύει η ίδια περίπτωση με την προηγούμενη, δηλαδή αν και το Gmail έρχεται ως προεπιλεγμένο πρόγραμμα στο Android, ο χρήστης μπορεί να επιλέξει οποίο εξυπηρετητή θεωρεί πιο αξιόπιστο για αυτόν.
- Google Maps, όπου βοηθάει τους χρήστες στο να δουν τα κτήρια της πανεπιστημιούπολης στο χάρτη. Θα μελετήσουμε αυτή την λειτουργία αναλυτικότερα στα επόμενα κεφάλαια.
- Google Drive, όπου αξιοποιείτε για την αποθήκευση εγγράφων και φωτογραφιών.

## ΚΕΦΑΛΑΙΟ 2: Σχεδιασμός

Στο κεφάλαιο 2 θα αναλυθούν οι λειτουργίες της εφαρμογής βήμα προς βήμα με την χρήση διαγραμμάτων ροής. Για την δημιουργία των διαγραμμάτων χρησιμοποιήθηκε το Microsoft Visio Professional.

### 2.1. Σχεδιασμός της εφαρμογής



Εικόνα 1. Έναρξη Λειτουργίας

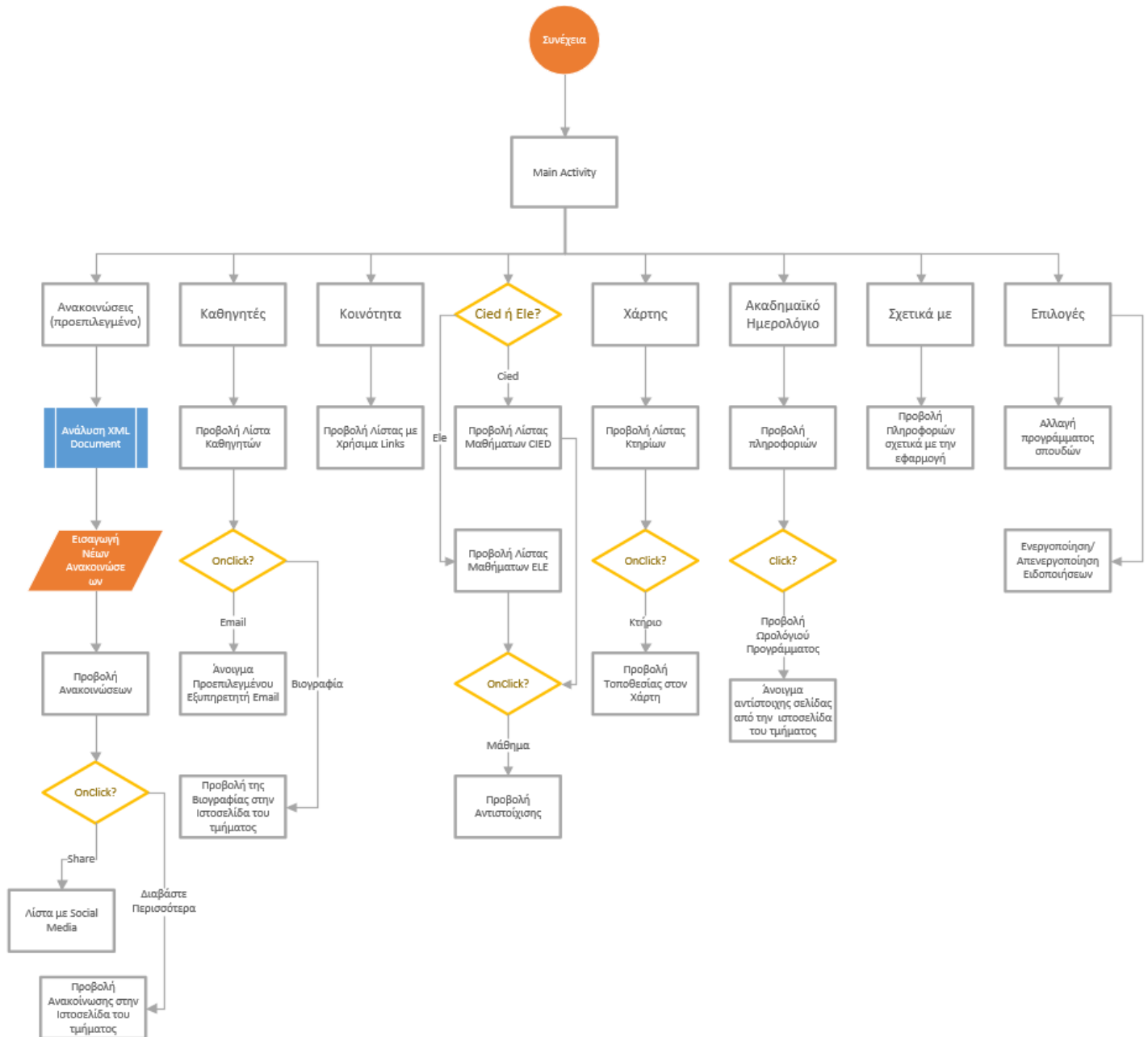
Η εφαρμογή κάθε φορά που εκτελείται από τον χρήστη εξετάζει το ενδεχόμενο του αν είναι η πρώτη φορά που τρέχει στην συσκευή ή αν έχει ξανά τρέξει στο παρελθόν. Στην πρώτη εκτέλεση, η εφαρμογή αντιγράφει στην τοπική μνήμη την βάση δεδομένων και έπειτα καλεί τον χρήστη να επιλέξει από μια σειρά επιλογών όπου θα αναλύσουμε στο επόμενο κεφάλαιο.

Από την δεύτερη εκτέλεση και έπειτα, τα παραπάνω βήματα παρακάμπτονται και συνεχίζει στα επόμενα.

Η εικόνα 2 δείχνει τις επιλογές του χρήστη μια προς μια που έχουν να κάνουν με όλους τους φοιτητές του τμήματος ανεξάρτητα τι πρόγραμμα σπουδών παρακολουθούν. Οι λειτουργίες είναι οι έξι:

- Ανακοινώσεις (προεπιλεγμένο)
- Καθηγητές
- Κοινότητα
- Χάρτης
- Ημερολόγιο
- Σχετικά με
- Ρυθμίσεις

Για τους φοιτητές των Μηχανικών Πληροφορικής Τ.Ε. και Ηλεκτρολόγων Μηχανικών Τ.Ε. τους δίνεται μια επιπλέον λειτουργία, την αναζήτηση και προβολή των αντίστοιχων μαθημάτων με αυτών του Ηλεκτρολόγων Μηχανικών και Μηχανικών Πληροφορικής. Η επιλογή αυτή δεν είναι διαθέσιμη για τους χρήστες του προγράμματος σπουδών «ΗΜΜΥ».



Εικόνα 2. Λειτουργίες και Δυνατότητες της Εφαρμογής

### 2.1.1. Ανακοινώσεις

Η κατηγορία «Ανακοινώσεις» είναι η πρώτη στο μενού επιλογών και η προεπιλεγμένη λειτουργία που εμφανίζεται κατά το άνοιγμα της εφαρμογής. Από την στιγμή της εκτέλεσης, η εφαρμογή τρέχει και κατεβάζει το XML αρχείο από τον επίσημο ιστότοπο

«<https://www.ece.uop.gr/announcement/feed/>», έπειτα περνάει στην βάση κάθε ανακοίνωση που δημοσιεύεται στην σελίδα αυτή κα τέλος, τα προβάλλει στην οθόνη.

Σε κάθε ανακοίνωση ο χρήστης έχει 2 επιλογές. Μπορεί να πατήσει το κουμπί “Διαβάστε Περισσότερα” και να ανακατευθυνθεί στην αντίστοιχη ανακοίνωση μέσα από την ιστοσελίδα του τμήματος ή να πατήσει το εικονίδιο δίπλα από την ημερομηνία για να μοιραστεί τον σύνδεσμο αυτής την ανακοινώσεις με κάποιον άλλον χρήστη.

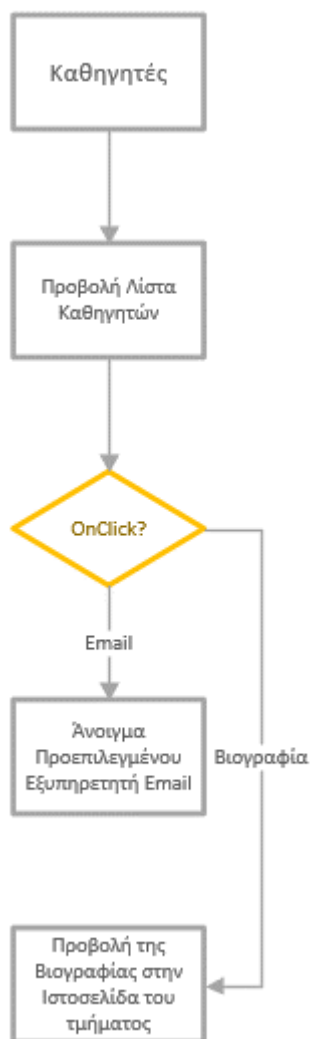


Εικόνα 3. Ανάλυση Λειτουργίας Ανακοινώσεων



## 2.1.2. Καθηγητές

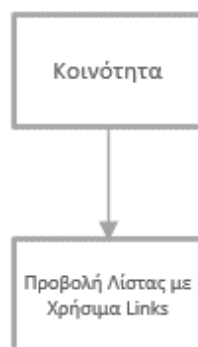
Στην κατηγορία καθηγητές εμφανίζεται στον φοιτητή η πλήρης λίστα με τους διδάσκοντες του τμήματος. Το κάθε ένα στοιχείο συνοδεύεται από δυο κουμπιά που το πρώτο, οδηγεί τον χρήστη στην αντίστοιχη σελίδα του τμήματος με το βιογραφικό του καθηγητή που επιλέχθηκε, ενώ το άλλο κουμπί δείχνει και ωθεί τον χρήστη να στείλει email στον διδάσκοντα.



Εικόνα 4. Ανάλυση Λειτουργίας «Καθηγητές»

### 2.1.3. Κοινότητα

Στην κατηγορία «Κοινότητα» προβάλλονται στον σπουδαστή διάφοροι σύνδεσμοι κοινωνικών δικτύων αλλά και άλλων υπηρεσιών με σκοπό την κοινωνικοποίηση, την βοήθεια να ενταχθεί και να προσαρμοστεί ευκολά στη νέα φοιτητή ζωή. Τα links θα αναλυθούν αναλυτικότερα στο επόμενο κεφάλαιο.



Εικόνα 5. Ανάλυση Λειτουργίας «Κοινότητα»

### 2.1.4. Αντιστοιχίες Μαθημάτων

Στην περίπτωση που ο φοιτητής στην πρώτη εκτέλεση της εφαρμογής επιλέξει ένα από τα παλιά προγράμματα σπουδών του τμήματος Μηχανικών Πληροφορικής Τ.Ε ή Ηλεκτρολόγων Μηχανικών Τ.Ε τότε θα του εμφανίζεται η κατηγορία «Αντιστοιχίσεις» με την λίστα μαθημάτων του αντίστοιχου προγράμματος σπουδών. Ο φοιτητής πατώντας σε οποιοδήποτε από τα μαθήματα στην λίστα έχει την δυνατότητα να δει την αντιστοιχία του μαθήματος δηλαδή το μάθημα που έχει αντικατασταθεί από το καινούργιο πρόγραμμα σπουδών με κάποιες πρόσθετες πληροφορίες που θα αναλυθούν στο επόμενο κεφάλαιο.



Εικόνα 6. Ανάλυση Λειτουργίας «Αντιστοιχίσεις Μαθημάτων»

### 2.1.5. Χάρτης

Στην κατηγορία «Χάρτης» εμφανίζεται μία λίστα με τα κτήρια του τμήματος και τις αίθουσες που το κάθε ένα από αυτά περιέχει. Με το πάτημα ενός κτηρίου, η εφαρμογή ανοίγει και δείχνει στο Google Maps με μια πινέζα την τοποθεσία του κτηρίου μέσα στην πανεπιστημιούπολη.



Εικόνα 7. Ανάλυση Λειτουργίας «Χάρτης»

### 2.1.6. Ημερολόγιο

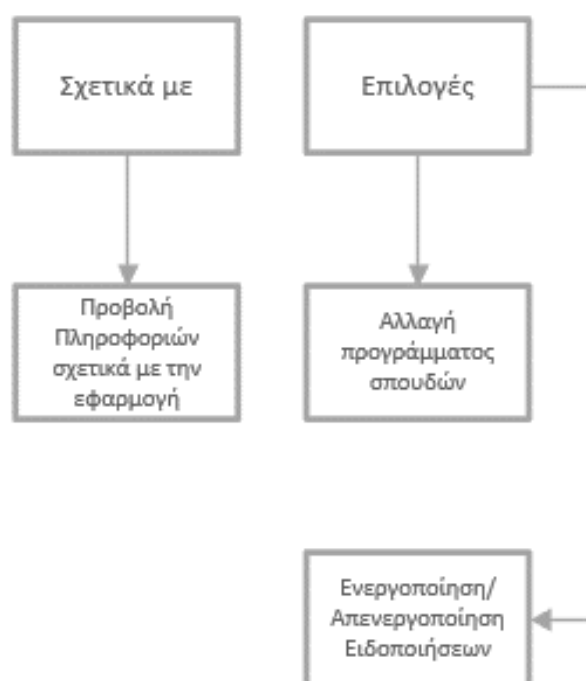
Στην κατηγορία «Ημερολόγιο» ο σπουδαστής μπορεί να ενημερωθεί σχετικά με το ακαδημαϊκό πρόγραμμα όλης της χρονιάς και να μεταβεί στην αντίστοιχη σελίδα του τμήματος για να ενημερωθεί σχετικά με το ωρολόγιο πρόγραμμα του εξάμηνου που παρακολουθεί.



Εικόνα 8. Ανάλυση Λειτουργίας «Ακαδημαϊκό Ημερολόγιο»

### 2.1.7. Σχετικά με & Επιλογές

Οι δυο τελευταίες κατηγορίες είναι οι « Σχετικά» και «Επιλογές». Στην κατηγορία «Σχετικά» ο σπουδαστής ενημερώνεται για τον σκοπό της εφαρμογής και πως μπορεί να την αξιοποιήσει. Στην κατηγορία «Επιλογές» ο χρήστης μπορεί να αλλάξει τις προτιμήσεις του σχετικά με το αν θέλει ή όχι να λαμβάνει ειδοποιήσεις όταν υπάρχουν νέες ανακοινώσεις του τμήματος ή/και να αλλάξει το πρόγραμμα σπουδών που είχε επιλέξει στην πρώτη εγκατάσταση της εφαρμογής.



Εικόνα 9. Ανάλυση Λειτουργίας «Σχετικά με» & «Επιλογές»

## 2.2. Σχεδιασμός της Βάσης Δεδομένων

Η Βάση δεδομένων, όπως ανέφερα και στην περίληψη, δημιουργήθηκε με την βοήθεια του ενσωματωμένου συστήματος διαχείρισης σχεσιακών βάσεων δεδομένων «SQLite». Η βάση ονομάζεται “hmtmy.db” και περιέχει συνολικά πέντε πίνακες, όπως εμφανίζονται στην παραπάνω εικόνα 10.

RSS_TABLE	
PK	ID_LINK
	RSS_TITLE
	RSS_CONTEXT
	RSS_DATE

FEL_TEI_COURSES_TABLE	
PK	FEL_ID
	COURSE_TITLE
	HMMY_ID

HMMY_COURSES_TABLE	
PK	ID
	COURSE_TITLE
	SEMESTER
	COURSE HOURS
	LAB_HOURS
	TUTORING_HOURS

CIED_TEI_COURSES_TABLE	
PK	TEI_ID
	COURSE_TITLE
	HMMY_ID

SETTINGS_TABLE	
PK	SETTINGS_ID
	SETTINGS_CHOICE

**Εικόνα 10. Πίνακες Βάσης Δεδομένων**

Ο πίνακας με το όνομα «RSS\_TABLE» χρησιμοποιείτε για την αποθήκευση των ανακοινώσεων που προβάλλονται στην κατηγορία «Ανακοινώσεις». Συγκεκριμένα,

- Στο πεδίο «ID\_LINK» αποθηκεύεται το URL της κάθε ανακοινώσεων και χρησιμοποιείται ως πρωτεύων κλειδί
- Στο πεδίο «RSS\_TITLE» αποθηκεύεται ο τίτλος της κάθε ανακοίνωσης
- Στο πεδίο «RSS\_CONTEXT» αποθηκεύεται το περιεχόμενο κάθε ανακοίνωσης
- Στο πεδίο «RSS\_DATE» αποθηκεύεται η ημερομηνία της κάθε ανακοίνωσης

Ο πίνακας «FEL\_TEI\_COURSES\_TABLE» και ο πίνακας «CIED\_TEI\_COURSES\_TABLE» είναι παρόμοιοι. Ο πρώτος περιέχει όλα τα μαθήματα του τμήματος Ηλεκτρολόγων Μηχανικών Τ.Ε και ο δεύτερος όλα τα μαθήματα του τμήματος Μηχανικών Πληροφορικής Τ.Ε. Και οι δυο αυτοί πίνακες έχουν τα τρία εξής πεδία:

- «FEL\_ID» & «CIED\_ID» που χρησιμοποιείται ως πρωτεύων κλειδί και περιέχει τους κωδικούς του κάθε μαθήματος.

- «COURSE\_TITLE» όπου είναι αποθηκευμένοι οι τίτλοι μαθημάτων
- «HMMY\_ID» στο οποίο είναι αποθηκευμένος ο κωδικός μαθήματος του αντίστοιχου μαθήματος από το καινούργιο πρόγραμμα σπουδών.

Ο πίνακας «HMMY\_COURSES\_TABLE» περιέχει όλα τα μαθήματα του καινούργιου προγράμματος σπουδών. Πιο αναλυτικά περιλαμβάνει,

- «HMMY\_ID» που χρησιμοποιείται ως πρωτεύων κλειδί και είναι αποθηκευμένος ο κωδικός του κάθε μαθήματος.
- «COURSE\_TITLE» στο οποίο πεδίο, είναι αποθηκευμένοι οι τίτλοι μαθημάτων
- «SEMESTER», αναφέρει τον αριθμό εξαμήνου διδασκαλίας του μαθήματος
- «COURSE\_HOURS», αναφέρει τον εβδομαδιαίο αριθμό ωρών διδασκαλίας του κάθε μαθήματος
- «LAB\_HOURS», αναφέρει πόσες ώρες την εβδομάδα διδάσκεται το κάθε εργαστήριο
- «TUTORING\_HOURS», αναφέρει τις εβδομαδιαίες ώρες του φροντιστηρίου του κάθε μαθήματος

Ο πίνακας « SETTINGS\_TABLE» χρησιμοποιείται για να αποθηκευτούν οι προτιμήσεις του χρήστη. Ο πίνακας περιέχει:

- «SETTINGS\_ID» που χρησιμοποιείται ως πρωτεύων κλειδί και είναι αποθηκευμένο το ID της ρύθμισης.
- «SETTINGS\_CHOICE» στο οποίο πεδίο, είναι αποθηκευμένοι οι προτιμήσεις χρήστη.

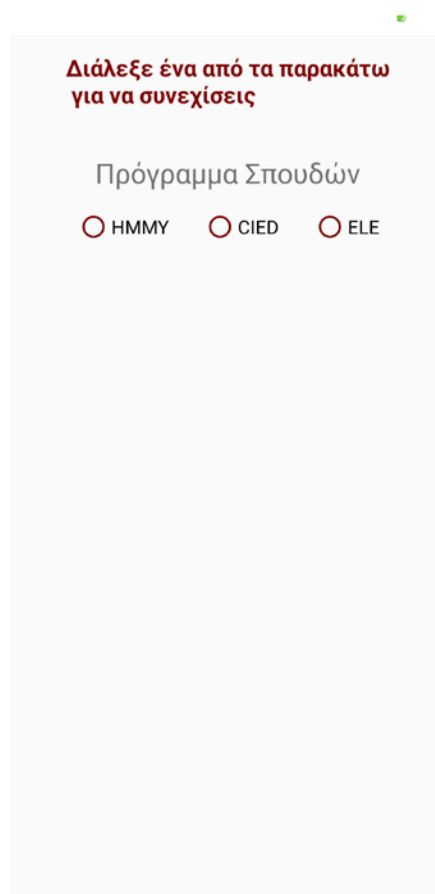
Σε αυτή την έκδοση υπάρχουν 3 στοιχεία μέσα στον πίνακα, “STUDIES”, “NEWS\_NOTIFICATION”, “NEWS\_COUNTER”. Στο στοιχείο με κλειδί το “STUDIES” αποθηκεύονται 3 διαφορετικές τιμές οι οποίες είναι “cied\_choice”, “ele\_choice” και “hmmmy\_choice”, ανάλογα το πρόγραμμα σπουδών του χρήστη. Στο στοιχείο με κλειδί το “NEWS\_NOTIFICATION” αποθηκεύονται 2 τιμές, “yes” αν ο χρήστης θέλει να ενημερώνεται για νέες ανακοινώσεις ή “no” αν ο χρήστης δεν επιθυμεί να λαμβάνει ειδοποιήσεις . Και τέλος, το στοιχείο με κλειδί “NEWS\_COUNTER”, στο οποίο αποθηκεύονται ακέραιοι αριθμοί, χρησιμοποιείται βοηθητικά για την εμφάνιση ειδοποιήσεων.



## ΚΕΦΑΛΑΙΟ.3: Διάταξη και Γραφικά Εφαρμογής

Στο κεφάλαιο 3 θα δούμε και θα αναλύσουμε την διάταξη και την εμφάνιση της εφαρμογής. Για την δημιουργία όλων των οθονών χρησιμοποιήθηκε το “ConstraintLayout” το οποίο ανήκει στην οικογένεια του “android.view.ViewGroup” όπου βοηθάει στην τοποθέτηση και στην μεγέθυνση γραφικών εργαλείων με έναν πολύ ευέλικτο τρόπο. Παράλληλα, εκτός από την ευελιξία, η συγκεκριμένη διάταξη μειώνει τις εμφωλεύσεις (Nesting) βελτιώνοντας έτσι την συνολική απόδοση της εφαρμογής.

### 3.1. Πρώτη Εκτέλεση



Εικόνα 11. . Αρχική Επιλογή

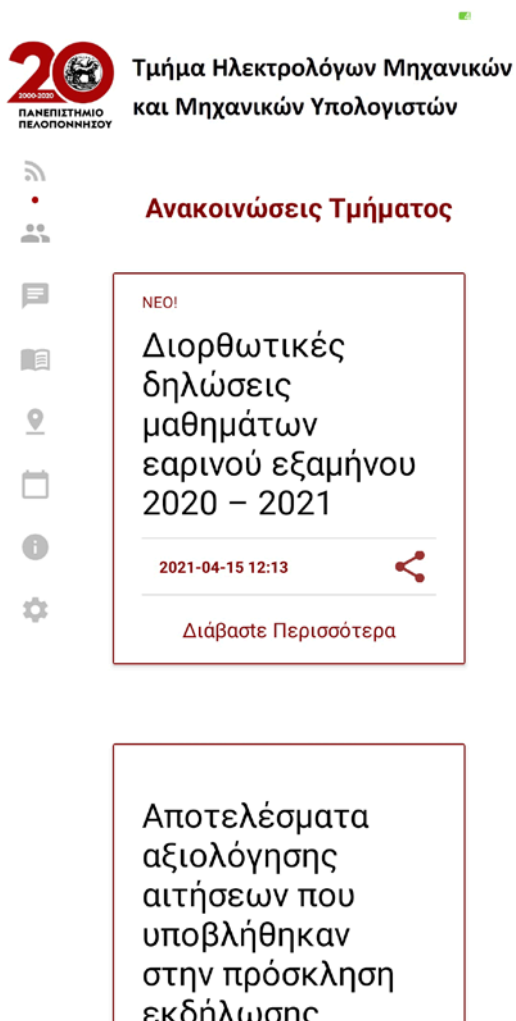
Με την πρώτη εκτέλεση της εφαρμογής, όπως αναφέραμε και στο προηγούμενο κεφάλαιο, ο χρήστης καλείται να διαλέξει ένα από τα τρία εξής προγράμματα σπουδών:

- HMMY, για το τμήμα Ηλεκτρολόγων Μηχανικών & μηχανικών Πληροφορικής,
- CIED, για το πρώην Τμήμα μηχανικών Πληροφορικής Τ.Ε,

- ELE, για το πρώην Τμήμα Ηλεκτρολόγων μηχανικών Τ.Ε.

Οι διάφορες μεταξύ των επιλογών έχουν να κάνουν με την λίστα μαθήματων που εμφανίζονται στην κατηγορία “Αντιστοιχίσεις” και με τα κοινωνικά δίκτυα που προβάλλονται στην κατηγορία «Κοινότητα».

### 3.2 Μενού Χρήστη & Ανακοινώσεις Τμήματος



Εικόνα 12. Κατηγορία «Ανακοινώσεις»

Στην εικόνα 12 προβάλλεται το μενού χρήστη, αλλά και η προεπιλεγμένη κατηγορία “Ανακοινώσεις Τμήματος”. Το μενού, το οποίο βρίσκεται πάντα στα αριστερά της οθόνης, περιέχει ένα εικονίδιο για κάθε κατηγορία. Ο χρήστης καταλαβαίνει σε ποια κατηγορία έχει

πατήσει, παρατηρώντας τη κόκκινη κουκίδα που εμφανίζεται στο κάτω μέρος από κάθε εικονίδιο.

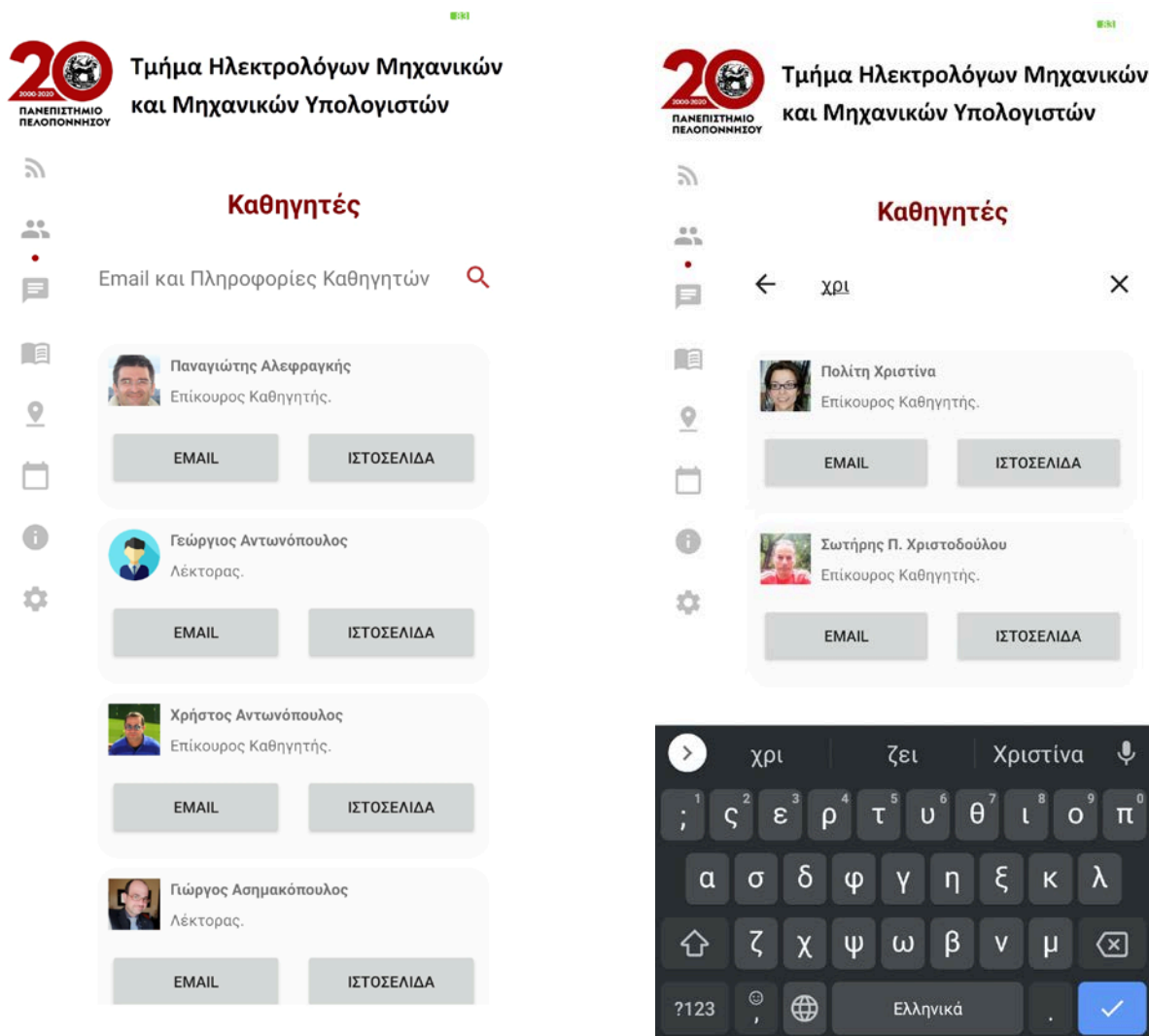
Στη κατηγορία «Ανακοινώσεις Τμήματος» ο φοιτητής ενημερώνεται σχετικά με τις νέες ανακοινώσεις. Κάθε άρθρο έχει τα ίδια χαρακτηριστικά, τα οποία είναι:

- Τίτλο
- Ημερομηνία δημοσίευσης
- Κουμπί κοινοποίησης
- Εμφάνιση «ΝΕΟ» αν το άρθρο δημοσιεύθηκε τις τελευταίες 96 ώρες
- Επιλογή «Διαβάστε περισσότερα»

### 3.3 Καθηγητές

Στο μενού «Καθηγητές» ο σπουδαστής μπορεί να αναζητήσει και να βρει τους καθηγητές και μέλη ΔΕΠ του τμήματος σύμφωνα με την ιστοσελίδα της σχολής<sup>(1)</sup>. Σε κάθε καθηγητή αναλογούν δυο κουμπιά, «EMAIL» και «ΙΣΤΟΣΕΛΙΔΑ».

- Το κουμπί Email παροτρύνει τον χρήστη στην προεπιλεγμένη εφαρμογή του κινητού του για αποστολή και λήψη ηλεκτρονικών μηνυμάτων (email), στην οποία δημιουργείτε αυτόματα ένα νέο μήνυμα με το email του διδάσκοντα να είναι συμπληρωμένο στην περιοχή του «Αποστολέα»
- Το κουμπί Ιστοσελίδα, παραπέμπει τον χρήστη στην σελίδα του διδάσκοντα στην επίσημη ιστοσελίδα του τμήματος.



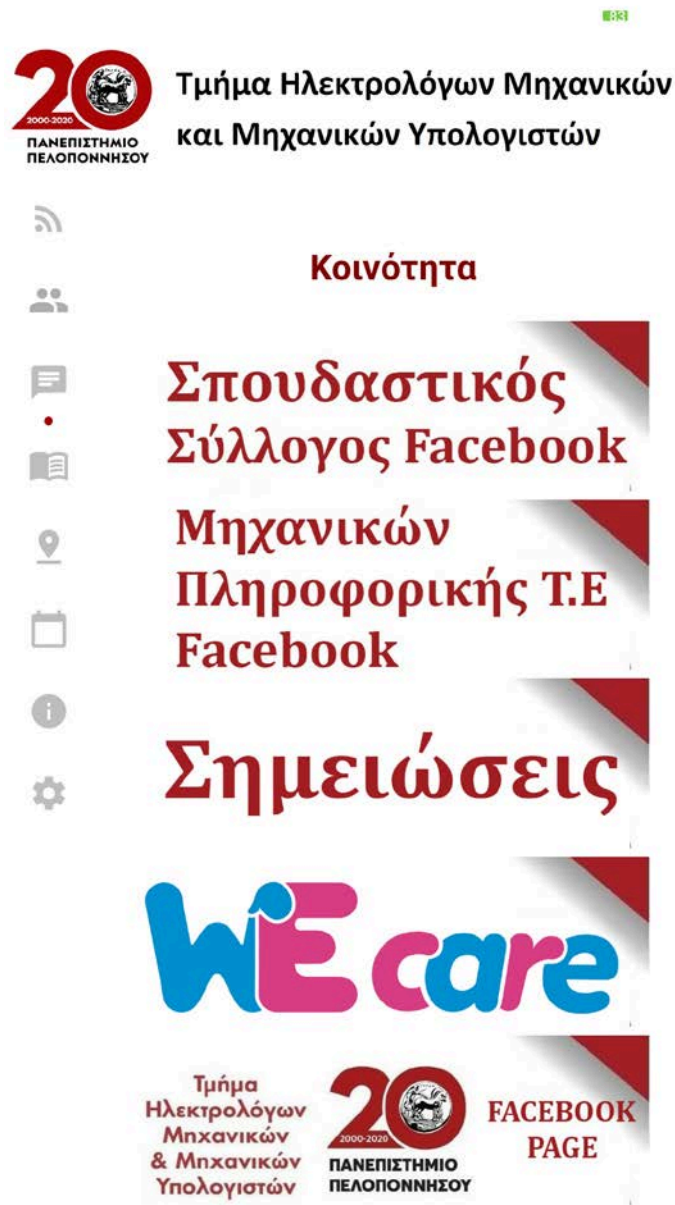
Εικόνα 13. Κατηγορία «Καθηγητές»

### 3.4 Κοινότητα

Στο μενού «Κοινότητα» ο φοιτητής μπορεί να, ανακαλύψει ομάδες φοιτητών στα κοινωνικά δίκτυα, αλλά και υπηρεσίες του τμήματος. Συγκεκριμένα μπορεί:

- Να γίνει μέλος στο Facebook στην ομάδα των Μηχανικών Πληροφορικής Τ.Ε (Αν έχει επιλεγεί το “CIED” ως πρόγραμμα σπουδών) <sup>(2)</sup>
- Να γίνει μέλος στο Facebook στην ομάδα των ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών<sup>(3)</sup>
- Να κάνει «Like» στην επίσημη σελίδα του τμήματος στο Facebook<sup>(4)</sup>
- Να ενημερωθεί σχετικά με τη «Δομή Συμβουλευτικής και Ψυχολογικής Υποστήριξης WeCare» του Πανεπιστημίου Πελοποννήσου<sup>(5)</sup>

- Να έχει πρόσβαση σε ένα κοινόχρηστο φάκελο μέσω την υπηρεσίας Google drive με τον οποίο θα μπορεί να μοιράζεται αλλά και να βρίσκει σημειώσεις από άλλους συμφοιτητές.

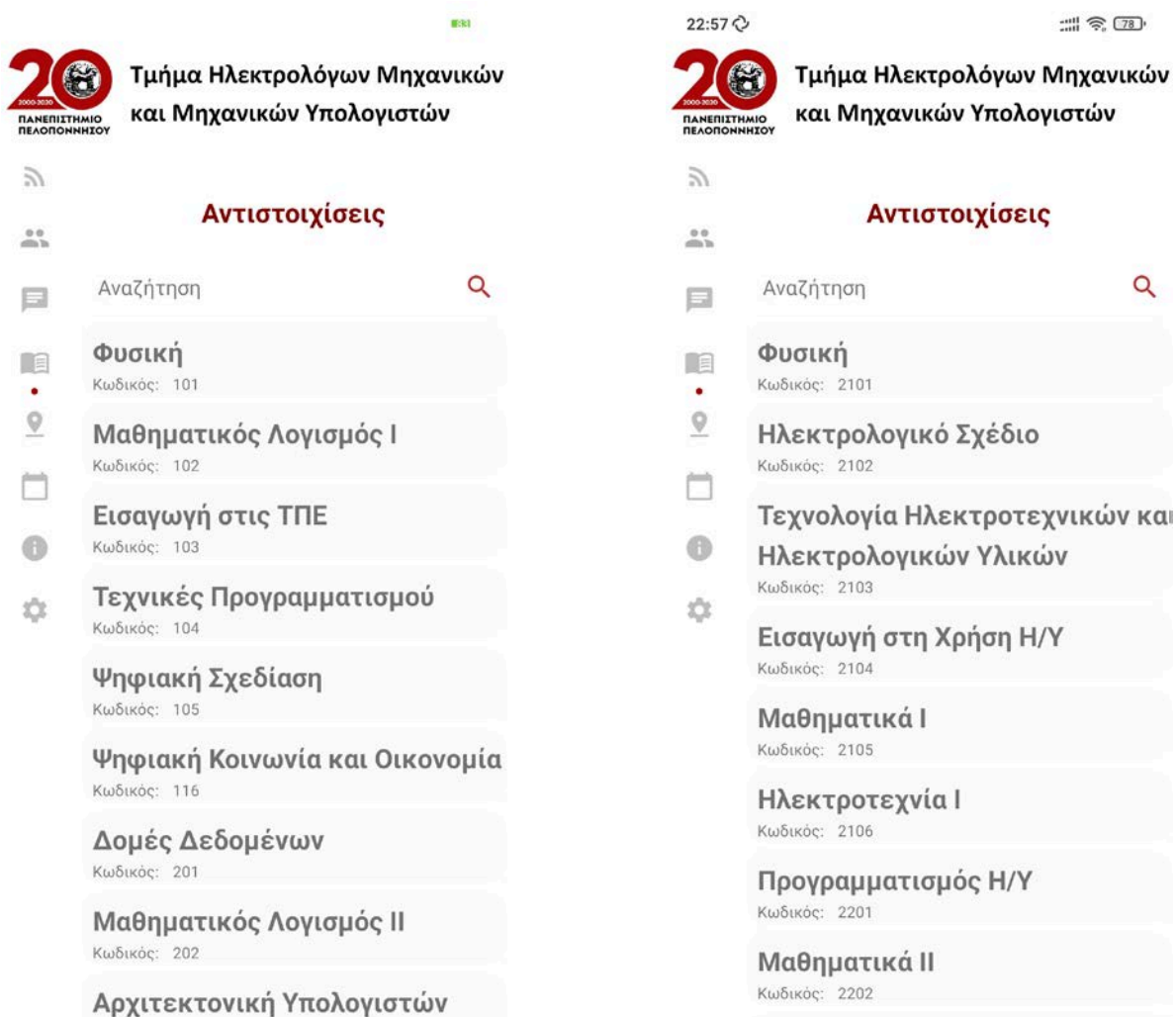


Εικόνα 14. Κατηγορία «Κοινότητα»

### 3.5 Αντιστοιχίσεις

Στο μενού «Αντιστοιχίσεις» ο φοιτητής μπορεί αναζήτηση μαθήματα από τα πρώην τμήματα του Τ.Ε.Ι Δυτικής Ελλάδας και να βρει με ποια αντιστοιχίζονται. Στην λίστα εμφανίζονται μόνο τα μαθήματα από το πρόγραμμα σπουδών που επέλεξε κατά την πρώτη εκτέλεση της εφαρμογής.

Για παράδειγμα αν ο φοιτητής επιλέξει “CIED” ως το πρόγραμμα σπουδών που παρακολουθεί τότε η λίστα θα δείχνει τα μαθήματα των Μηχανικών Πληροφορικής Τ.Ε<sup>(6)</sup>. Αν όμως επιλέξει “ELE” ως πρόγραμμα σπουδών τότε εμφανίζονται στην λίστα όλα τα μαθήματα των Ηλεκτρολόγων Μηχανικών Τ.Ε<sup>(7)</sup>. Τέλος αν ο σπουδαστής ανήκει στο καινούργιο τμήμα και έχει επιλέξει ως πρόγραμμα σπουδών το “HMMY” τότε η κατηγορία «Αντιστοιχίσεις» του είναι ανούσια και δεν εμφανίζεται στο μενού πλοήγησης.

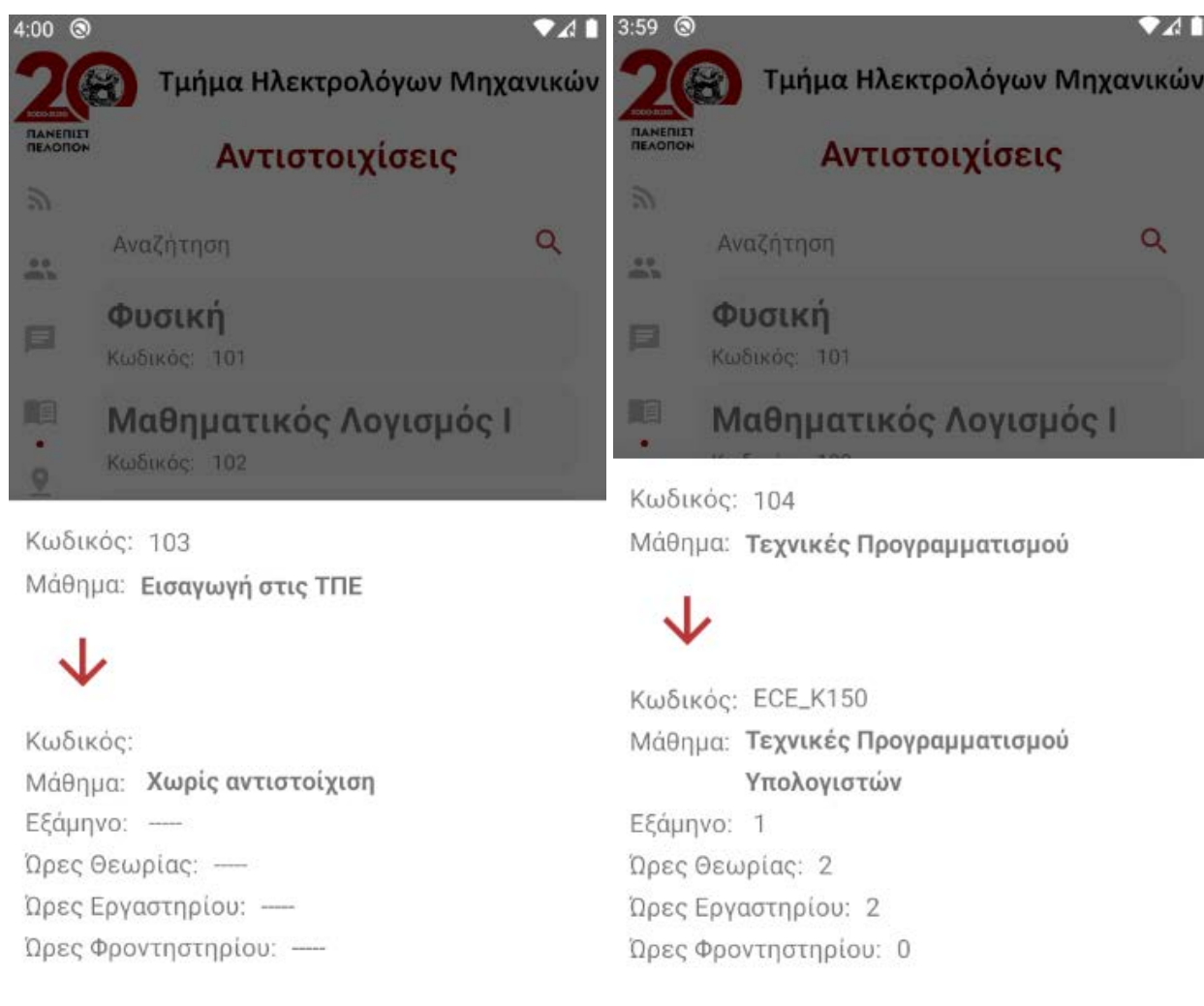


Εικόνα 15. Κατηγορία Αντιστοιχίσεις CIED & ELE

Αφού ο φοιτητής αναζητήσει και επιλέξει το μάθημα που τον ενδιαφέρει να μάθει με ποιο έχει αντικατασταθεί τότε εμφανίζεται ένα νέο παράθυρο που τον ενημερώνει σχετικά με τα παρακάτω στοιχεία<sup>(8)</sup>:

- Κωδικό μαθήματος
- Όνομα μαθήματος
- Εξάμηνο διδασκαλίας
- Ώρες θεωρίας
- Ώρες Εργαστηρίου
- Ώρες φροντιστηρίου

Αν το μάθημα δεν αντιστοιχείται με κανένα τότε ενημερώνει τον χρήστη ότι δεν υπάρχει αντιστοίχιση (εικόνα 16)

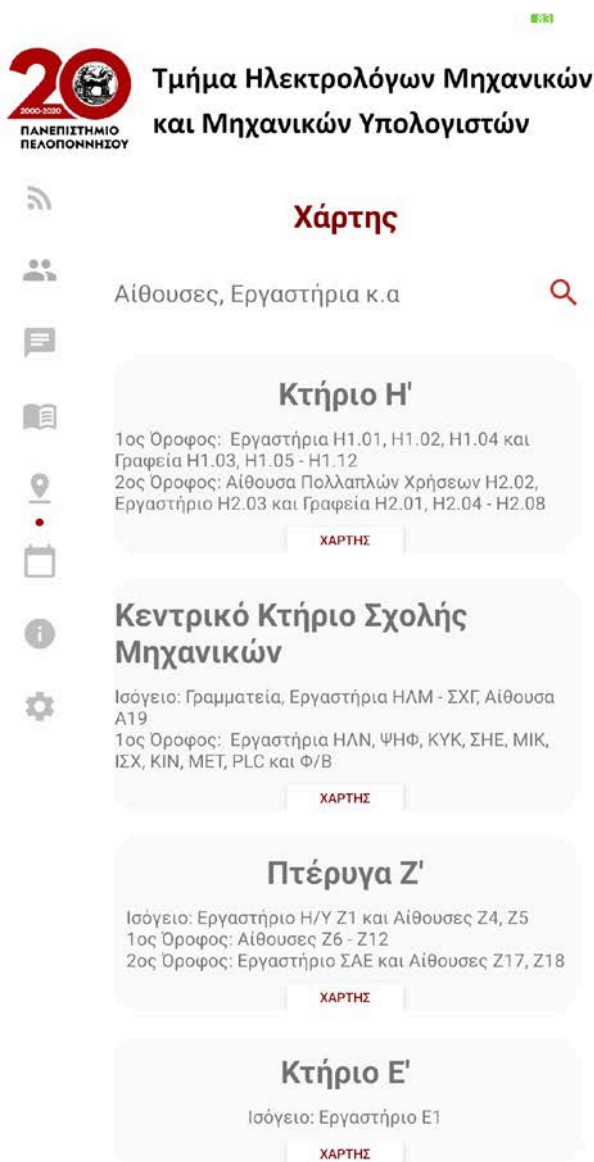


Εικόνα 16. Παράδειγμα Μαθήματος Με και Χωρίς Αντιστοίχιση

### 3.6 Χάρτης & Ακαδημαϊκό Ωρολόγιο

Στο μενού χάρτης ο φοιτητής μπορεί να αναζητήσει κτήρια, να μάθει τι περιλαμβάνει το καθένα και να δει στο χάρτη των Google Maps την ακριβή τοποθεσία τους στην πανεπιστημιούπολη<sup>(9)</sup>. Στην λίστα, κάθε κτήριο εμφανίζεται με:

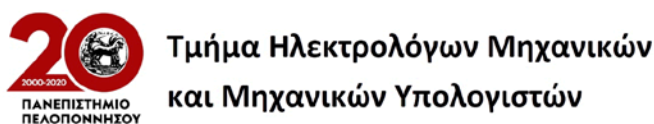
- Έναν τίτλο, που περιλαμβάνει το όνομα του κτηρίου
- Μια περιγραφή η οποία αναφέρει τις αίθουσες, τα γραφεία και τα εργαστήρια που αυτό περιλαμβάνει
- Ένα κουμπί το οποίο ανοίγει τον χάρτη του κινητού με μια πινέζα που δείχνει την ακριβή τοποθεσία του.



Εικόνα 17. Κατηγορία «Χάρτης»



Στο μενού «Ακαδημαϊκό Ημερολόγιο» ο χρήστης ενημερώνεται σχετικά με το ακαδημαϊκό ημερολόγιο της αντίστοιχης χρονιάς που διανύει σύμφωνα με την επίσημη ιστοσελίδα της σχολής<sup>(10)</sup>. Η κατηγορία επίσης περιέχει και ένα κουμπί με το όνομα «Δείτε εδώ το Εβδομαδιαίο Πρόγραμμα» το οποίο προωθεί τον χρήστη στην ιστοσελίδα της σχολής και συγκεκριμένα στην σελίδα με τα τρία ωρολόγια προγράμματα του εξαμήνου.



## Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών



### Ακαδημαϊκό Ημερολόγιο



Το ακαδημαϊκό ημερολόγιο του ακαδημαϊκού έτους 2020 – 2021, έχει ως εξής:



#### Χειμερινό Εξάμηνο



- Έναρξη μαθημάτων: 05.10.2020
- Ολοκλήρωση μαθημάτων και αναπληρώσεων: 29.01.2021



- Έναρξη εξεταστικής περιόδου: 01.02.2021
- Λήξη εξεταστικής περιόδου: 18.02.2021



#### Εαρινό Εξάμηνο



- Έναρξη μαθημάτων: 24.02.2021
- Ολοκλήρωση μαθημάτων και αναπληρώσεων: 18.06.2021



- Έναρξη εξεταστικής περιόδου: 22.06.2021
- Λήξη εξεταστικής περιόδου: 09.07.2021

#### Επαναληπτική Εξεταστική

- Έναρξη: 30.08.2021
- Λήξη: 24.09.2021

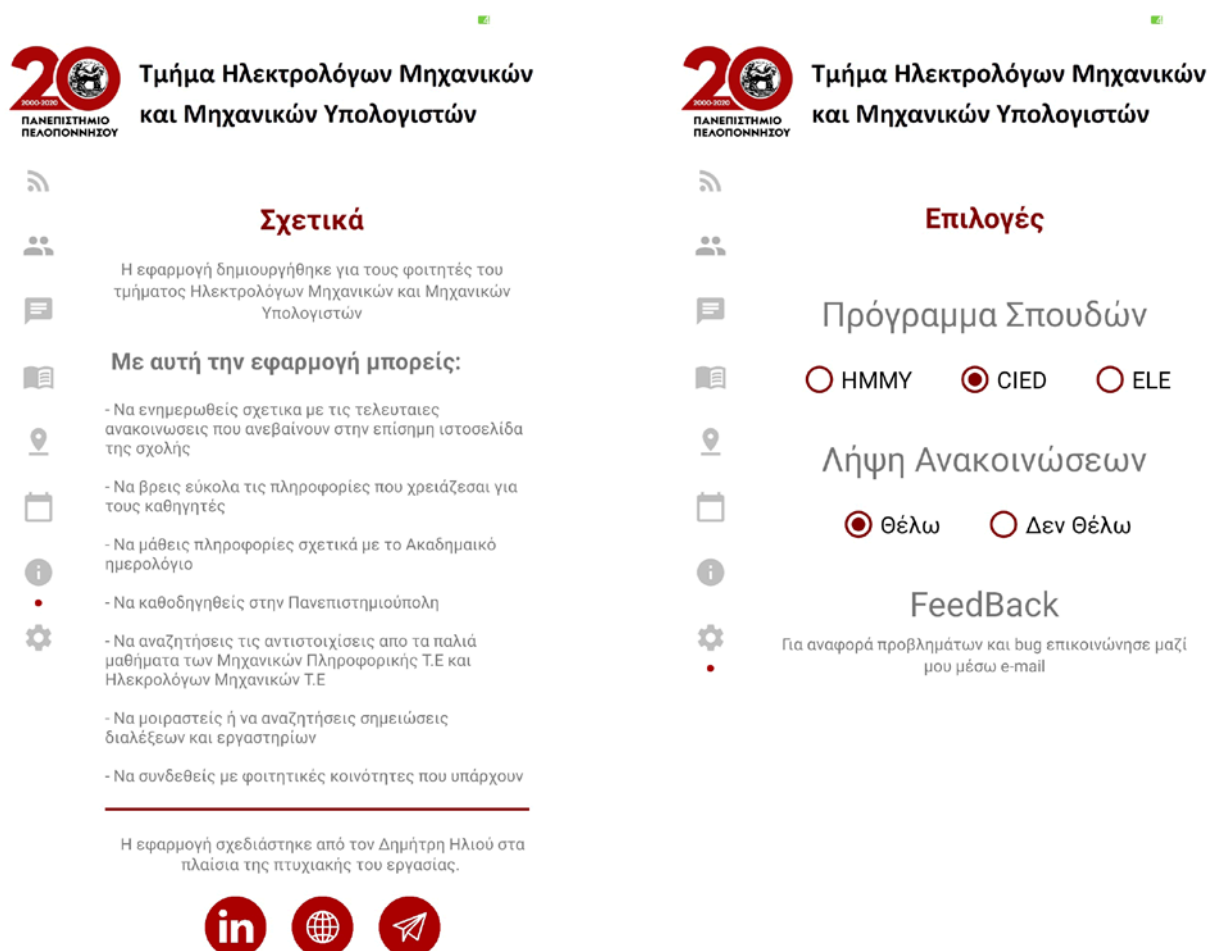
*[Δείτε Εδώ το Εβδομαδιαίο Πρόγραμμα](#)*

Εικόνα 18. Ακαδημαϊκό Ημερολόγιο

### 3.7 Σχετικά & Προτιμήσεις Χρήστη

Στο μενού «Σχετικά» ο σπουδαστής μπορεί να μάθει πως μπορεί να αξιοποιήσει την εφαρμογή δίνοντας του συμβουλές ενώ στο μενού «Επιλογές» μπορείς να αλλάξει τις προτιμήσεις. Πιο αναλυτικά έχει την δυνατότητα να αλλάξει:

- Το πρόγραμμα σπουδών του,
- Την λήψη ειδοποιήσεων κάθε φορά που μια νέα ανακοίνωση δημοσιεύεται στην ιστοσελίδα του τμήματος



Εικόνα 19. Κατηγορίες «Σχετικά» & «Επιλογές»

## ΚΕΦΑΛΑΙΟ 4: Παρουσίαση Κώδικα

Σε αυτό το κεφάλαιο θα ονομάσουμε στο σύνολο των κλάσεων και θα εξηγήσουμε κάποιες από τις λειτουργίες με κάποια κομμάτια κώδικα που αξίζουν να αναφερθούν.

### 4.1 Κλάσης

Η εφαρμογή παραλαμβάνει 25 κλάσης γραμμένες σε γλώσσα προγραμματισμού Java. Οι ονομασίες και η κατηγοριοποίηση τους είναι οι έξι:

- News
  1. NewsAdapter.java
  2. NewsFragment.java
  3. NewsModel.java
- Community
  1. CommunityAdapter.java
  2. CommunityFragment.java
  3. CommunityModel.java
  4. CommunityCallback.java
- Professors
  1. ProfessorsAdapter.java
  2. ProfessorsFragment.java
  3. ProfessorsModel.java
- Map
  1. MapAdapter.java
  2. MapFragment.java
  3. MapModel.java
- Courses
  1. CoursesFragmentDetails.java
  2. CoursesMatchingAdapter.java
  3. CoursesMatchingFragment.java
  4. CoursesHmmyMatchingModel.java
  5. CoursesTeiMatchingModel.java

- Database
  1. DatabaseHelper.java
- AboutFragment.java
- CalendarFragment.java
- MainActivity.java
- WelcomeActivity.java
- AppJobService.java
- SettingsFragment.java

## 4.2 Activity

Ένα activity είναι ένα συστατικό της εφαρμογής (και κατά συνέπεια το μέρος του κώδικα) το οποίο παρέχει τη δυνατότητα τουλάχιστον μιας αρχικής οθόνης, έτσι ώστε ο χρήστης να μπορεί να την χρησιμοποιήσει όπως για π.χ. να στείλει ένα email, να βγάλει μια φωτογραφία κτλ.

Κάθε activity προσφέρει ένα γραφικό περιβάλλον μέσα από το οποίο μας δίνονται οι επιλογές για το πρόγραμμα. Κάθε εφαρμογή που δημιουργούμε αποτελείται από ένα τουλάχιστον activity, ενώ επαγγελματικές εφαρμογές περιέχουν πολλά περισσότερα activities. Όσα activities και να έχουμε όμως, πάντα ένα θεωρείται το κύριο activity το οποίο παρουσιάζεται στον χρήστη όταν ξεκινάει η εφαρμογή. Από εκεί και πέρα, κάθε activity μπορεί να καλέσει ένα άλλο activity αν το απαιτεί η εφαρμογή μας.

Όταν σταματήσει ένα activity επειδή ένα άλλο καινούργιο activity ξεκινάει, το παλιό activity ενημερώνεται για την αλλαγή της κατάστασης του, μέσα από μια σειρά μεθόδων που ονομάζονται «lifecycle callback methods». Υπάρχουν αρκετές μέθοδοι από τις οποίες ένα activity θα περάσει από τη στιγμή που θα δημιουργηθεί μέχρι τη στιγμή που θα καταστραφεί. Αυτές οι μέθοδοι αντιπροσωπεύουν τα stages μέσα από τις οποίες περνάει το activity στη διάρκεια της λειτουργίας του.

Για να δημιουργήσουμε ένα activity χρειάζεται να δημιουργήσουμε μια δική μας κλάση η οποία κάνει extends από την κλάση Activity. Το activity που δημιουργήσαμε , που είναι το

μοναδικό εξ ορισμού, φορτώνει τα γραφικά του στοιχεία χρησιμοποιώντας το XML αρχείο που βρίσκεται μέσα στο φάκελο «res/ layout».

Κάθε activity κλάση ορίζει μια σειρά από events που καθορίζουν το lifecycle του activity. Πιο συγκεκριμένα η Activity κλάση ορίζει τα εξής events:

- onCreate( ) – καλείται όταν ξεκινάει το activity την πρώτη φορά
- onStart( ) – καλείται όταν το activity γίνεται ορατό στον χρήστη
- onResume( ) – καλείται όταν το activity χρησιμοποιείται από τον χρήστη
- onPause( ) – καλείται όταν το τρέχον activity έχει γίνει pause και ένα προηγούμενο activity που ήταν σε αδράνεια έχει ξεκινήσει πάλι
- onStop( ) – καλείται όταν το activity δεν είναι πια ορατό στον χρήστη
- onDestroy( ) – καλείται όταν το activity έχει διαγραφεί από την μνήμη του συστήματος και στην ουσία δεν υφίσταται πια για το πρόγραμμα
- onRestart( ) – καλείται όταν το activity έχει σταματήσει και ξεκινάει πάλι
- Εξ ορισμού, το activity που δημιουργείται περιέχει το onCreate( ) event μέσα στο οποίο έχουμε γράψει τον κώδικα που παρουσιάζει τα γραφικά στην οθόνη μας <sup>(11)</sup>

Στην εφαρμογή υπάρχουν συνολικά δύο Activities, το MainActivity και το WelcomeActivity. Στο WelcomeActivity ελέγχεται αν η εφαρμογή έχει εκτελεστεί ξανά στο παρελθόν. Με την βοήθεια του interface getSharedPreferences όπως βλέπουμε στην εικόνα 20 που ακολουθεί, δημιουργούμε μια προτίμηση (preference) με το όνομα “isFirstRun” και με Boolean τιμή true.

Στην πρώτη εκτέλεση τρέχει ο κώδικας στην γραμμή 44 και εμφανίζει την οθόνη που απεικονίζεται στην εικόνα 11. Από την δεύτερη εκτέλεση και έπειτα, χρησιμοποιώντας intent στην γραμμή 52 μεταφερόμαστε στην MainActivity.

```

40 new Handler().postDelayed(new Runnable() {
41     @Override
42     public void run() {
43         //set fragment for first use
44         Boolean isFirstRun = getSharedPreferences( name: "mystudies", MODE_PRIVATE)
45             .getBoolean( key: "isFirstRun", defValue: true);
46         if (isFirstRun) {
47             FirstScreen();
48             DataBaseHelper dataBaseHelper = new DataBaseHelper(getApplicationContext());
49             dataBaseHelper.changeSetting( id: "STUDIES", choice: "hmmmy_choice");
50         } else {
51             // set the default fragment when activity launch
52             Intent homeIntent = new Intent( packageContext: WelcomeActivity.this, MainActivity.class);
53             startActivity(homeIntent);
54             finish();
55         }
56         getSharedPreferences( name: "mystudies", MODE_PRIVATE).edit()
57             .putBoolean("isFirstRun", false).apply();
58     }
59 }
60 }, SPLASH_TIME_OUT);
61 }

```

Εικόνα 20. Έλεγχος Πρώτης Εκτέλεσης

### 4.3 Fragments

Ένα τεμάχιο (fragment) τυπικά παριστά ένα επαναχρησιμοποιήσιμο τμήμα της διεπαφής χρήστη μιας Activity, αλλά μπορεί επίσης να παριστά επαναχρησιμοποιήσιμη λογική προγράμματος. Γενικά, μπορούν να συνδυαστούν αρκετά τεμάχια, έτσι ώστε να δημιουργηθούν διεπαφές χρήστη, οι οποίες εκμεταλλεύονται τα μεγέθη οθόνης ταμπλετών. Επίσης, τα τεμάχια μπορούν εύκολα να εναλλαχθούν, ούτως ώστε οι Γραφικές Διεπαφές Χρήστη (ΓΔΧ) να γίνουν πιο δυναμικές. Το Fragment (πακέτο android.app) είναι η βασική κλάση για όλα τα τεμάχια.

Οι Fragment πρέπει να φιλοξενούνται από μια Activity, δε μπορούν να εκτελούνται ανεξάρτητα. Όταν η εφαρμογή εκτελείται σε οριζόντιο προσανατολισμό σε μια ταμπλέτα, η MainActivity φιλοξενεί όλες τις Fragment. Σε κατακόρυφο προσανατολισμό σε οποιαδήποτε συσκευή, η SettingsActivity φιλοξενεί την SettingsFragment και η MainActivity φιλοξενεί τις άλλες.

Ένα Fragment αποτελεί ένα κομμάτι από το UI, που περιέχει τον απαραίτητο κώδικα, για να διογκώσει ή να κατασκευάσει μια διάταξη (layout) καθώς και να διαχειριστεί την αλληλεπίδραση του χρήστη με αυτό.

Όπως και οι δραστηριότητες, έτσι και τα fragments έχουν κύκλο ζωής. Στη πραγματικότητα οι δραστηριότητες είναι στενά συνδεδεμένες με τα fragment και ο κύκλος ζωής της δραστηριότητας επηρεάζει το κύκλο ζωής του fragment, με το οποίο σχετίζεται. Πρώτον, τα fragment διατρέχουν αυτή τη σειρά των γεγονότων του κύκλου ζωής, με την εξής σειρά:

- onAttach (Activity) – Υποδεικνύει το fragment το οποίο είναι συνδεδεμένο με μια δραστηριότητα
- onCreate (Bundle) - Προετοιμάζει το fragment
- onCreateView (LayoutInflater, ViewGroup, Bundle) – Επιστρέφει το view (οθόνη) που σχετίζεται με το fragment.
- onActivityCreated(bundle) – Ενεργοποιείται για να συμπέσει με τη μέθοδο της δραστηριότητας onCreate()
- onStart() – Ενεργοποιείται για να συμπέσει με τη μέθοδο της δραστηριότητας onStart() και εμφανίζει το fragment
- onResume() – Ενεργοποιείται για να συμπέσει με τη μέθοδο της δραστηριότητας onResume() και υποδεικνύει το fragment που μπορεί να χειριστεί την αλληλεπίδραση <sup>(12)</sup>

Στην εφαρμογή υπάρχουν 8 διαφορετικά fragment που το κάθε ένα από αυτά καλείται μέσα από την MainActivity.

## 4.4 RecyclerView & Adapters

Η RecyclerView είναι μια βιβλιοθήκη που δημιουργήθηκε και χρησιμοποιείται με σκοπό την προβολή στοιχείων από λίστες που περιέχουν τεράστιο όγκο δεδομένων. Η RecyclerView ανακυκλώνει και επαναχρησιμοποιεί τα γραφικά στοιχεία για κάθε νέο στοιχείο της λίστας που χρειάζεται να προβληθεί. Αυτή η μέθοδος επαναχρησιμοποίησης βελτιώνει σημαντικά την απόδοση, αλλά και την ανταπόκριση της εφαρμογής, μειώνοντας την κατανάλωση ενέργειας<sup>(13)</sup>.

Η επικοινωνία μεταξύ μιας RecyclerView και των δεδομένων της λίστας γίνεται με την χρήση των Adapters. Ένας Adapter είναι μια γέφυρα μεταξύ ενός στοιχείου διεσπάρης χρήστη και της πηγής δεδομένων που μας βοηθά να προβάλλουμε δεδομένα στην οθόνη. Δηλαδή ο Adapter

είναι υπεύθυνος στο να δημιουργήσει ένα καινούργιο γραφικό στοιχείο για κάθε στοιχείου που υπάρχει στα δεδομένα της λίστας<sup>(14)</sup>.

## 4.5 Threads & AsyncTask

Κάθε φορά που εκτελούμε μια συσκευή στο κινητό μας, ένα καινούργιο Thread (Νήμα)<sup>(15)</sup> δημιουργείται στο οποίο εκτελείτε το μεγαλύτερο μέρος του κώδικα. Το νήμα ονομάζεται Main και όλα τα στοιχεία (Activities, Services κ.α.) τρέχουν σε αυτό. Συνήθως όταν εκτελείται μια ενέργεια, η διαδικασία δεν είναι χρονοβόρα. Υπάρχουν περιπτώσεις όμως, όπως π.χ. μια διαδικτυακή εργασία, που μπορεί να διαρκέσει αρκετά και στην χειρότερη περίπτωση να μπλοκάρει το Main Thread προκαλώντας τον εξαναγκασμένο τερματισμό της εφαρμογής. Σε αυτές τις περιπτώσεις λοιπόν υπάρχουν διάφορες λύσεις και μια από αυτές την προσφέρει η AsyncTask.

Η AsyncTask πρόκειται για ένα thread εντός μιας κλάσης η οποία επιτρέπει την επικοινωνία με το UI thread (το κεντρικό thread δηλαδή) της εφαρμογής με απλό και σίγουρο τρόπο, δίχως δηλαδή να μας απασχολούν ζητήματα συγχρονισμού κ.α, καθώς όλα αυτά τα αναλαμβάνει η AsyncTask class παρέχοντας σου μια σειρά από properties, συναρτήσεις και διαδικασίες πάνω στις οποίες μπορούμε να χτίσουμε (Extend) την δική μας κλάση.

Οι 4 ρουτίνες που προσφέρει η AsyncTask είναι:

- η onPreExecute η οποία εκτελείται εντός του UI Thread, συνήθως εδώ δημιουργούμε ένα ProgressDialog που ενημερώνει τον χρήστη ότι η εφαρμογή μας ξεκινά μια χρονοβόρα διαδικασία κλπ..
- doInBackground, εδώ τοποθετούμε τον κώδικα που θα εκτελεσθεί στο Thread της AsyncTask class. Η συνάρτηση είναι τύπου μεταβλητού αριθμού παραμέτρων, το είδος της επιστρεφόμενης τιμής καθορίζεται από εμάς κατά την δήλωση του AsyncTask μας.
- Η τρίτη είναι η onPostExecute, όπως και η onPreExecute εκτελείται εντός του UI Thread σου, συνήθως εδώ διαγράφουμε το ProgressDialog που δημιουργήσαμε κατά την onPreExecute. Επίσης εδώ μπορούμε να ενημερώσουμε το UI Thread για τα αποτελέσματα της AsyncTask.
- Η τέταρτη είναι η onProgressUpdate, και αυτή όπως η προηγούμενη διαδικασία τρέχει εντός του UI Thread μας και μας επιτρέπει να ενημερώνουμε το υπόλοιπο πρόγραμμα



για την εξέλιξη των εντολών που εκτελούνται εντός της `doInBackground` κάθε φορά που η τελευταία εκτελεί την συνάρτηση `publishProgress`.<sup>(16)</sup>

## 4.6 JobService

`JobService` ουσιαστικά είναι μια διεργασία που μπορεί να εκτελέσει μακροχρόνιες λειτουργίες στο παρασκήνιο χωρίς να απαιτείται η αλληλεπίδραση του χρήστη με την εφαρμογή. Ένα `JobService` περιλαμβάνει 3 PUBLIC μεθόδους:

- `jobFinished` η οποία καλείται για να ενημερώσει το `JobScheduler` ότι η διεργασία εκτελέστηκε με επιτυχία. (Γραμμή 64)
- `onStartJob` η οποία καλείται για να δηλώσει ότι μια διεργασία έχει ξεκινήσει να τρέχει στο παρασκήνιο. Επιστρέφοντας «true» δηλώνουμε θέλουν να επαναληφθεί στο μέλλον. (Γραμμή 69)
- `onStopJob` η οποία καλείται όταν θέλουμε να σταματήσουμε μια διεργασία από το να εκτελεστεί ξανά στο μέλλον επιστρέφοντας «false»<sup>(17)</sup>. (Γραμμή 83)
- 

Τα παραπάνω απεικονίζονται στην εικόνα 21.

```

67     private boolean jobCancelled = false;
68     @Override
69     public boolean onStartJob(JobParameters params) {
70         //turn on the notifications
71         DataBaseHelper check = new DataBaseHelper( context: this);
72         String notification = check.getSettings( id: "NEWS_NOTIFICATION");
73         if (notification.equals("yes")){
74             jobCancelled = false;
75         } else if (notification.equals("no")){
76             jobCancelled = true;
77         }
78         Log.d(TAG, msg: "Rss started");
79         createNotificationChannel();
80         doBackgroundWork(params);
81         return true;
82     }
83     private void doBackgroundWork(final JobParameters params) {
84         new Thread(new Runnable() {
85             @Override
86             public void run() {
87                 // do something here
88                 jobScheduler = (JobScheduler) getSystemService(JOB_SCHEDULER_SERVICE);
89                 checkForNewAnnouncements();
90                 if (jobCancelled) {
91                     return;
92                 }
93                 Log.d(TAG, msg: "Rss finished");
94                 jobFinished(params, wantsReschedule: false);
95             }
96         }).start();
97     }
98     @Override
99     public boolean onStopJob(JobParameters params) {
100         //turn off the notifications
101         Log.d(TAG, msg: "Rss cancelled");
102         jobCancelled = true;
103         return true;
104     }

```

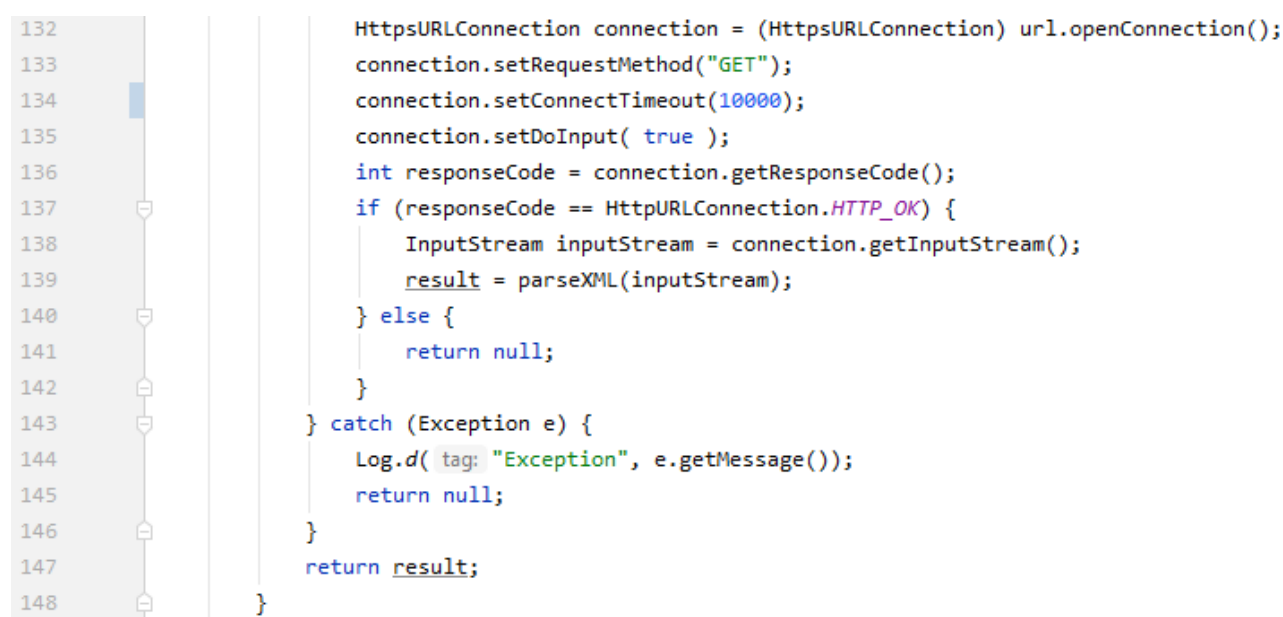
Εικόνα 21. Παράδειγμα JobService

## 4.7 Σύνδεση με τον Ιστότοπο και Προσπέλαση XML

Για την λήψη και την προβολή των ανακοινώσεων του τμήματος χρησιμοποιήθηκαν οι παρακάτω μέθοδοι.

Για την επικοινωνία της εφαρμογής με την ιστοσελίδα χρησιμοποιήθηκε η κλάση με το όνομα "HttpsURLConnection". Η χρήση της κλάσης εξασφαλίζει μια νέα σύνδεση καλώντας

“url.openConnection()” και έπειτα γίνεται η μετάδοση του αποτελέσματος στην μεταβλητή “connection”. Στη συνέχεια στέλνεται είναι GET Request όπου αναλύεται και αποθηκεύεται το XML στην μεταβλητή “result”. Αν για κάποιο λόγο η σύνδεση δεν είναι επιτυχής τότε τίποτα από τα παραπάνω δεν θα συμβεί και η εφαρμογή θα δείξει στην χρήστη το ανάλογο μήνυμα. Όλα τα παραπάνω προβάλλονται στην εικόνα22 που ακολουθεί.



Εικόνα 22. Τρόπος Σύνδεσης με Ιστότοπο

### 4.7.1 ArrayList & Hashmap

Όπως προ-αναφέρθηκε, τα δεδομένα αναλύονται από ένα XML αρχείο το οποίο προέρχεται από τον ιστότοπο <https://www.ece.uop.gr/announcement/feed/>. Το XML περιέχει τις ανακοινώσεις οι οποίες περικλείονται από τις ετικέτες <item> </item> (εικόνα 23).

Επιπλέον, αναμεσα από αυτές της ετικέτες χρησιμοποιούνται και κάποιες βοηθητικές από τις οποίες εμείς θα βασιστούμε της εξής τέσσερις:

- <title> : Περιεχει το τίτλο της ανακοίνωσης
- <link> : Περιεχει το σύνδεσμο της ανακοίνωσης
- <pubDate> : Περιεχει την ημερομηνία και ώρα δημοσίευσης της ανακοίνωσης
- <content:encoded> : Περιεχει το κείμενο της ανακοίνωσης

```

</item>
<title>Επαναπροκήρυξη πρόσκλησης εκδήλωσης ενδιαφέροντος της πράξης «Υποστήριξη των εκπαιδευτικών δραστηριοτήτων των ΑΕΙ κατά το ακαδημαϊκό έτος 2020-2021 με
την ενσωμάτωση ενισχυτικής διδασκαλίας» του Παν. Πελοποννήσου</title>
<link>https://www.ece.uop.gr/announcement/epanaprokiryksi-prosklisis-ekdilosis-endiiferontos-tis-praksis-ypostiriksi-ton-ekpaideftikon-drastiriotion-ton-aei-
kata-to-akadimaiko-etos-2020-2021-me-tin-ensomatosi-enischytikis-didaskalias-tou-pan/?utm_source=rss&#038;utm_medium=rss&#038;utm_campaign=epanaprokiryksi-prosklisis-
ekdilosis-endiiferontos-tis-praksis-ypostiriksi-ton-ekpaideftikon-drastiriotion-ton-aei-kata-to-akadimaiko-etos-2020-2021-me-tin-ensomatosi-enischytikis-didaskalias-tou-
pan</link>
<dc:creator><![CDATA[adminuser]]></dc:creator>
<pubDate>Wed, 28 Apr 2021 06:42:17 +0000</pubDate>
<guid isPermaLink="false">https://www.ece.uop.gr/?post_type=announcements&#038;p=14580</guid>
<description><![CDATA[<p>Δείτε την επαναπροκήρυξη της πρόσκλησης εκδήλωσης ενδιαφέροντος (7859/26.04.2021) για υποβολή &#230; </p>
<p>The post <a rel="nofollow" href="https://www.ece.uop.gr/announcement/epanaprokiryksi-prosklisis-ekdilosis-endiiferontos-tis-praksis-ypostiriksi-ton-ekpaideftikon-
drastiriotion-ton-aei-kata-to-akadimaiko-etos-2020-2021-me-tin-ensomatosi-enischytikis-didaskalias-tou-pan/">Επαναπροκήρυξη πρόσκλησης εκδήλωσης ενδιαφέροντος της πράξης
«Υποστήριξη των εκπαιδευτικών δραστηριοτήτων των ΑΕΙ κατά το ακαδημαϊκό έτος 2020-2021 με την ενσωμάτωση ενισχυτικής διδασκαλίας» του Παν. Πελοποννήσου</a> appeared first on
<a rel="nofollow" href="https://www.ece.uop.gr">Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών</a>.</p>
]]></description>
<content:encoded><![CDATA[<p>Δείτε την επαναπροκήρυξη της <a href="https://www.ece.uop.gr/wp-
content/uploads/2021/04/%CE%A0%CF%81%CF%8C%CF%83%CE%BA%CE%BB%CE%B7%CF%83%CE%B7_7859_2021_%CE%9F%CF%81%CE%B8%CE%AE-%CE%95%CF%80%CE%B1%CE%B0%CE%AC%CE%BB%CE%B7%CF%88%CE%B7.pdf">
<strong>πρόσκλησης εκδήλωσης ενδιαφέροντος</strong></a> (7859/26.04.2021) για υποβολή αιτήσεων υποψηφιότητας στο πλαίσιο υλοποίησης της πράξης <strong>«Υποστήριξη των
εκπαιδευτικών δραστηριοτήτων των ΑΕΙ κατά το ακαδημαϊκό έτος 2020-2021 με την ενσωμάτωση ενισχυτικής διδασκαλίας επιπρόσθετα των κύριων διαλέξεων</strong> με Κ.Α. 80573
(κωδικός ΜΙΣ: 5089190) του Πανεπιστημίου Πελοποννήσου (ΑΔΑ: 980Β469Β7Δ-1ΚΨ)</p>
<p>Αρχεία πρόσκλησης;&nbsp;&nbsp;&nbsp;<a href="https://www.ece.uop.gr/wp-content/uploads/2021/04/%CE%91%CE%AF%CF%84%CE%B7%CF%83%CE%B7.docx">Αίτηση</a>&nbsp;&nbsp;&nbsp;<a href="https://www.ece.uop.gr/wp-content/uploads/2021/04/%CE%A5%CF%80%CE%B5%CF%80%CE%B8%CF%85%CE%B0%CE%B7-%CE%94%CE%AE%CE%BB%CF%89%CF%83%CE%B7.docx">Υπεύθυνη δήλωση</a></p>
<p>Προθεσμία υποβολής αιτήσεων: <strong>07.05.2021</strong>&nbsp;&nbsp;&nbsp;<strong>ώρα <strong>14:00</strong>.</p>
<p>The post <a rel="nofollow" href="https://www.ece.uop.gr/announcement/epanaprokiryksi-prosklisis-ekdilosis-endiiferontos-tis-praksis-ypostiriksi-ton-ekpaideftikon-
drastiriotion-ton-aei-kata-to-akadimaiko-etos-2020-2021-me-tin-ensomatosi-enischytikis-didaskalias-tou-pan/">Επαναπροκήρυξη πρόσκλησης εκδήλωσης ενδιαφέροντος της πράξης
«Υποστήριξη των εκπαιδευτικών δραστηριοτήτων των ΑΕΙ κατά το ακαδημαϊκό έτος 2020-2021 με την ενσωμάτωση ενισχυτικής διδασκαλίας» του Παν. Πελοποννήσου</a> appeared first on
<a rel="nofollow" href="https://www.ece.uop.gr">Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών</a>.</p>
]]></content:encoded>
</item>

```

Εικόνα 23. Παράδειγμα XML

Η τεχνική που χρησιμοποιήθηκε για να αναλύσουμε (parse) της παραπάνω ετικέτες μπορούμε να την δούμε στις εικόνες 22 & 23 που ακολουθούν. Συγκεκριμένα δηλώθηκε ένα “ArrayList” που εσωτερικά, σε κάθε κελί του πίνακα, αποθηκεύεται ένα “HashMap”. ArrayList είναι μια επέκταση της απλής λίστας (List) ενώ ένα Hashmap είναι μια επέκταση του απλού χάρτη (Map).

Η διαφορά μεταξύ των δύο είναι πως στη λίστα μπορούμε να αποθηκεύσουμε μόνο μια τιμή σε κάθε κελί, ενώ στο χάρτη μπορούμε σε κάθε κελί να αποθηκεύσουμε την τιμή αλλά και ένα κλειδί που αποσκοπεί στη γρήγορη εύρεση του στοιχείου στο μέλλον. Κάθε στοιχείο του ArrayList περιέχει μια διαφορετική ανακοίνωση που χωρίζεται σε κομμάτια εσωτερικά ενός Hashmap, εισχωρώντας σε κάθε κελί τις παραπάνω ετικέτες ως κλειδιά (values) και το περιεχόμενο των ετικετών ως τιμές. (εικονα 25)

```

168
169 @
170 private ArrayList<HashMap<String, String>> parseXML(InputStream inputStream)
171     throws ParserConfigurationException, IOException, SAXException, ParseException {
172
173     DocumentBuilderFactory documentBuilderFactory = DocumentBuilderFactory.newInstance();
174     DocumentBuilder documentBuilder = documentBuilderFactory.newDocumentBuilder();
175     Document document = documentBuilder.parse(inputStream);
176     Element element = document.getDocumentElement();
177
178     NodeList itemList = element.getElementsByTagName("item");
179     NodeList itemChildren;
180
181     Node currentItem;
182     Node currentChild;
183
184     String link = null;
185     String title = null;
186     String pubDate = null;
187     String content = null;
188     String newDateFormat = null;
189
190     ArrayList<HashMap<String, String>> items = new ArrayList<>(); //the one that returns at the end of the execution
191     HashMap<String, String> currentMap; //supported hashmap
192

```

Εικόνα 24. ArrayList&lt;Hashmap&lt;String,String&gt;&gt;

```

193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
}

for (int i = 0; i < itemList.getLength(); i++) { //for each article
    currentItem = itemList.item(i);
    itemChildren = currentItem.getChildNodes();
    currentMap = new HashMap<>();

    for (int j = 0; j < itemChildren.getLength(); j++) { //for each tag inside the article
        currentChild = itemChildren.item(j);

        if (currentChild.getNodeName().equalsIgnoreCase( anotherString: "title")) {
            // Log.d("Title", String.valueOf(currentChild.getTextContent()));
            currentMap.put("title", currentChild.getTextContent());
            title = (String) currentChild.getTextContent();
        }
        if (currentChild.getNodeName().equalsIgnoreCase( anotherString: "content:encoded")) { //content for a future use
            // Log.d("description", String.valueOf(currentChild.getTextContent()));
            currentMap.put("content:encoded", currentChild.getTextContent());
            content = (String) currentChild.getTextContent();
        }
        if (currentChild.getNodeName().equalsIgnoreCase( anotherString: "pubDate")) {
            // Log.d("Title", String.valueOf(currentChild.getTextContent()));
            currentMap.put("pubDate", currentChild.getTextContent());
            pubDate = (String) currentChild.getTextContent();
            @SuppressWarnings("SimpleDateFormat")
            SimpleDateFormat inputPattern = new SimpleDateFormat( pattern: "EEE, d MMM yyyy HH:mm:ss Z");
            @SuppressWarnings("SimpleDateFormat")
            SimpleDateFormat outputPattern = new SimpleDateFormat( pattern: "yyyy-MM-dd HH:mm");
            Date date = inputPattern.parse(pubDate);
            newDateFormat = outputPattern.format(date);
        }
        if (currentChild.getNodeName().equalsIgnoreCase( anotherString: "link")) {
            currentMap.put("link", currentChild.getTextContent());
            link = (String) currentChild.getTextContent();
        }
    }
    if (!currentMap.isEmpty()) {
        items.add(currentMap);
        newsModel = new NewsModel(link, title, newDateFormat, content, category: "-1"); //category for a future use
        databaseHelper.addOne(newsModel); //adds each new item in the database
    }
}

return items;
}

```

Εικόνα 25. Προσπέλαση Ανακοινώσεων

Τέλος, στην ρουτίνα “onPostExecute”, χρησιμοποιούμε την κλάση “DatabaseHelper” για να τραβήξουμε από την βάση όλα τα παλιά και καινούργια στοιχεία, και με την βοήθεια του adapter να τα προβάλλουμε στη RecyclerView όπως είδαμε στο προηγούμενο κεφάλαιο.

```

151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
@Override
protected void onPostExecute(ArrayList<HashMap<String, String>> result) {
    super.onPostExecute(result);
    hideProgressDialog();
    if (result == null) {
        showToast(msg: "Please check your connection and try again.");
    }

    int before = resultItems.size();
    resultItems.clear();
    ArrayList<HashMap<String, String>> everyone = databaseHelper.getAllRss();
    databaseHelper.changeSetting(id: "NEWS_COUNTER", String.valueOf(everyone.size()));
    resultItems.addAll(everyone);
    mAdapter.notifyItemRangeInserted(before, everyone.size());
    mRecyclerView.invalidate();
}

```

Εικόνα 26. onPostExecute

## 4.8 Χρήση Constructor

Εκτός απ’ τις μεθόδους και τις μεταβλητές που ορίζουμε εμείς, κάθε κλάση περιλαμβάνει και μία ειδική μέθοδο που αφορά ένα σύνολο εντολών το οποίο εκτελείται κάθε φορά που ένα νέο αντικείμενο δημιουργείται. Η ειδική αυτή μέθοδος ονομάζεται constructor και ξεχωρίζει από τις υπόλοιπες μεθόδους της κλάσης, αφού η ονομασία που της δίνουμε είναι αυτή της ίδιας κλάσης

Η μέθοδος constructor στην εικόνα 27 που ακολουθεί, κάθε φορά που καλείται δημιουργεί ένα αντικείμενο κλάσης NewsModel όπου ουσιαστικά το χρησιμοποιούμε είτε για την εξαγωγή και την προβολή των ανακοινώσεων με την βοήθεια των Adapter ή για την εισαγωγή νέων ανακοινώσεων στην τοπική βάση δεδομένων. Στην εφαρμογή υπάρχουν συνολικά 5 constructors

```

3 public class NewsModel {
4     private String link_id;
5     private String title;
6     private String pubDate;
7     private String context;
8     private String category;
9
10    public NewsModel(String link_id, String title, String pubDate, String context, String category) {
11        this.link_id = link_id;
12        this.title = title;
13        this.pubDate = pubDate;
14        this.context = context;
15        this.category = category;
16    }
17
18    public String getLink_id() { return link_id; }
21
22    public void setLink_id(String link_id) { this.link_id = link_id; }
25
26    public String getTitle() { return title; }
29
30    public void setTitle(String title) { this.title = title; }
33
34    public String getPubDate() { return pubDate; }
37
38    public void setPubDate(String pubDate) { this.pubDate = pubDate; }
41
42    public String getContext() { return context; }
45
46    public void setContext(String context) { this.context = context; }
49
50    public String getCategory() { return category; }
53
54    public void setCategory(String category) { this.category = category; }
57

```

Εικόνα 27. Παράδειγμα Constructor

## 4.9 Δημιουργία τοπικής Βάσης Δεδομένων

Για την δημιουργία της τοπικής βάσης δεδομένων χρησιμοποιήθηκε η βοηθητική κλάση “SQLiteOpenHelper”. Με την κλάση αυτή γίνεται η αντιγραφή της ήδη υπάρχουσας βάσης με το όνομα “hmgm.db”, η οποία έχει τοποθετηθεί στο φάκελο “assets” του Android Studio. Μέσα στην βάση υπάρχουν οι πίνακες τις εικόνες 10.

### 4.9.1 Απόσπαση Δεδομένων

Ως παράδειγμα θα πάρουμε και θα αναλύσουμε το τρόπο όπου η εφαρμογή τραβάει και προβάλλει όλα τα μαθήματα του προγράμματος σπουδών του πρώην Μηχανικών Πληροφορικής Τ.Ε. Μέσα στην βοηθητική κλάση DatabaseHelper, δημιουργήσαμε μια PUBLIC κλάση “getAllCiedCourses” η οποία όταν καλείται επιστρέφει μια λίστα με όλα τα μαθήματα του προγράμματος σπουδών. Με την χρήση απλών SQL εντολών επιλέγουμε όλα τα στοιχεία από τον πίνακα “CIED\_TEI\_COURSES\_TABLE” και με την εντολή “getReadableDatabase” ανοίγουμε την βάση. Υστέρα δηλώνουμε έναν κέρσορα (Cursor) ο οποίος θα μας βοηθήσει στο να σαρώσουμε όλα τα στοιχεία.

Αφού σαρωθούν όλα τα στοιχεία, κλείνουμε αναγκάστηκα των κέρσορα και την βάση δεδομένων, και επιστρέφουμε (return) την λίστα. Τα βήματα αυτά παρουσιάζονται στην εικόνα 28.

```

179 public List<CourseTeiMatchingModel> getAllCiedCourses() {
180
181     List<CourseTeiMatchingModel> returnList = new ArrayList<>();
182     String queryString = "SELECT * FROM " + CIED_TEI_COURSES_TABLE;
183     SQLiteDatabase db =this.getReadableDatabase();
184     Cursor cursor = db.rawQuery(queryString, selectionArgs: null);
185
186     if(cursor.moveToFirst()) {
187         do {
188             String courseID = cursor.getString( columnIndex: 0);
189             String courseTitle = cursor.getString( columnIndex: 1);
190             String courseMatch = cursor.getString( columnIndex: 2);
191
192             CourseTeiMatchingModel newCourse = new CourseTeiMatchingModel(courseID,courseTitle,courseMatch);
193             returnList.add(newCourse);
194         }while(cursor.moveToNext());
195     }
196
197     cursor.close();
198     db.close();
199     return returnList;
200 }

```

Εικόνα 28. Παράδειγμα Προσπέλασης Δεδομένων του Πίνακα “CIED\_TEI\_COURSES\_TABLE”

Για την προβολή της λίστας που επιστρέψαμε, καλείτε η συνάρτηση “fillCourseList()” στην οποία, αφού ξεκαθαρίσει ποιο πρόγραμμα σπουδών έχει επιλέξει ο σπουδαστής στην πρώτη εκτέλεση της εφαρμογής, καλεί την συνάρτηση από την κλάση “DatabaseHelper.getAllCiedCourses” όπου επιστρέφει την λίστα με όλα τα μάθημα στην



βοηθητική λίστα `courseList`. Η ίδια διαδικασία ακολουθείται και για τα άλλα δυο προγράμματα σπουδών.

```

60
61 private void fillCourseList() {
62     DataBaseHelper dataBaseHelper = new DataBaseHelper(getActivity());
63     String old_studies = dataBaseHelper.getSettings( id: "STUDIES");
64
65     if (old_studies.equals("cied_choice")) {
66         courseList.addAll(dataBaseHelper.getAllCiedCourses());
67     } else if (old_studies.equals("ele_choice")) {
68         courseList.addAll(dataBaseHelper.getAllFelCourses());
69     }
70 }
71

```

Εικόνα 29. `fillCourse()`

## 4.9.2 Τροποποίηση Δεδομένων

Στην περίπτωση της τροποποίησης της βάσης, όταν για παράδειγμα η εφαρμογή αποθηκεύει νέες ανακοινώσεις, τότε ακολουθείτε η εξής τεχνική. Δημιουργούμε μια κλάση με είσοδο ένα γνώρισμα (στην περίπτωση του παραδείγματος είναι το `""Rssmodel rssmodel"`). Έπειτα ενεργοποιούμε την επεξεργασία της βάσης δεδομένων με την βοηθητική εντολή `this.getWritableDatabase` και δημιουργούμε ένα τοπικό `ContentValues` με όνομα `CV` στο οποίο αποθηκεύουμε κάθε χαρακτηριστικό του γνωρίσματος που με την εντολή `insert` το εισάγουμε στο πίνακα με τα στοιχεία.

```

275 @
276 public boolean addOne(NewsModel rssModel) {
277     SQLiteDatabase db = this.getWritableDatabase();
278     ContentValues cv = new ContentValues();
279     cv.put(RSS_ID_LINK, rssModel.getLink_id());
280     cv.put(RSS_TITLE, rssModel.getTitle());
281     cv.put(RSS_DATE, String.valueOf(rssModel.getPubDate()));
282     cv.put(RSS_CONTEXT, rssModel.getContext());
283     cv.put(RSS_CATEGORY, rssModel.getCategory());
284
285     db.insert(RSS_TABLE, nullColumnHack: null, cv);
286     db.close();
287     return true;
288 }

```

Εικόνα 30. Παράδειγμα εισαγωγής στοιχείων στο πίνακα `"RSS_TABLE"`

## ΚΕΦΑΛΑΙΟ 5: Μελλοντικές Επεκτάσεις

Σε αυτό το κεφάλαιο θα αναφέρω επεκτάσεις και λειτουργίες που θα μπορούσαν να προστεθούν σε μελλοντικές αναβαθμίσεις.

### 5.1 Βαθμολογίες

Υπηρεσία παρακολουθήσεις βαθμολογιών μέσω της εφαρμογής. Πιο αναλυτικά, η υπηρεσία αυτή:

- Θα ειδοποιεί τον σπουδαστή κάθε φορά που μια νέα βαθμολογία καταχωρείτε στο όνομα του
- Θα προβάλλει το ιστορικό βαθμολογιών του φοιτητή
- Θα προβάλλει στατιστικά όπως τον μέσο όρο, ποσοστό ολοκληρώσεις απαιτούμενων μαθήματων

### 5.2 Αιτήσεις και Δηλώσεις Μαθημάτων

Μέσω της εφαρμογής ο φοιτητής θα μπορεί να κάνει αίτηση χορήγησης εγγράφων όπως:

- Βεβαίωση σπουδών
- Αναλυτική βαθμολογία
- Βεβαίωση για στρατολογία
- Βεβαίωση προϋποθέσεων έναρξης Πρακτικής Άσκησης

Επιπλέον στην ίδια λειτουργία, σε κάθε αρχή νέου εξάμηνου ο φοιτητής θα έχει την δυνατότητα της δηλώσεις μαθημάτων και εργαστηρίων που επιθυμεί να παρακολουθήσει.

### 5.3 Προσαρμοσμένο ωρολόγιο πρόγραμμα

Στην τωρινή έκδοση το ωρολόγιο πρόγραμμα παρουσιάζεται σαν πλάνο και περιέχει όλα τα μαθήματα και όλες τις κατευθύνσεις ανεξάρτητος από το προφίλ του φοιτητή. Με την βοήθεια της δηλώσεις μαθημάτων που αναφέρθηκε προηγούμενος, ένα ακόμα ευχάριστο εργαλείο θα ήταν ο κάθε χρήστης να έχει στην κατηγορία «Ακαδημαϊκό ημερολόγιο» τα μαθήματα, τα μαθήματα που έχει δηλώσει από την κατεύθυνση που έχει επιλέξει με λεπτομέρειες όπως π.χ. μάθημα, όνομα καθηγητή, αίθουσα και ώρα διδασκαλίας.

## 5.4 Βιβλιοθήκη

Τέλος, θα μπορούσε να προστεθεί μια καινούργια κατηγορία με όνομα «Βιβλιοθήκη» η οποία θα είναι συνδεδεμένη με το σύστημα δανεισμού της βιβλιοθήκης του τμήματος, και ο καθένας σπουδαστής να μπορεί να ψάξει, να βρει και να κάνει κρατήσει στο όνομα του οποιοδήποτε σύγγραμμά.

## ΥΠΟΣΗΜΕΙΩΣΗ

Ακολουθούν οι κλάσης και τα αρχεία XML του πηγαίου κώδικα της εφαρμογής.

### AppJobService.java

```

package net.dimitrisiliou.myhmmy.BackgroundUpdate;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.job.JobParameters;
import android.app.job.JobScheduler;
import android.app.job.JobService;
import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Build;
import android.os.SystemClock;
import android.util.Log;

import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;
import androidx.core.app.TaskStackBuilder;

import net.dimitrisiliou.myhmmy.R;
import net.dimitrisiliou.myhmmy.database.DataBaseHelper;
import net.dimitrisiliou.myhmmy.news.NewsModel;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpURLConnection;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.X509TrustManager;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

public class AppJobService extends JobService {

    int notificationID = 0;
    TaskStackBuilder stackBuilder;

```

```

Intent notificationIntent = new Intent(Intent.ACTION_VIEW);
PendingIntent resultPendingIntent;
NotificationCompat.Builder builder;
NotificationManagerCompat notificationManager;
private ArrayList<HashMap<String, String>> resultItems = new
ArrayList<>();
private static String mFeedUrl =
"https://www.ece.uop.gr/announcement/feed/";
NewsModel newsModel;
JobScheduler jobScheduler;
private static final String TAG = "teeeeeeeeeeeeeeeeeesting";

private boolean jobCancelled = false;
@Override
public boolean onStartJob(JobParameters params) {
    //turn on the notifications
    DataBaseHelper check = new DataBaseHelper(this);
    String notification = check.getSettings("NEWS_NOTIFICATION");
    if (notification.equals("yes")){
        jobCancelled = false;
    } else if (notification.equals("no")){
        jobCancelled = true;
    }
    Log.d(TAG, "Rss started");
    createNotificationChannel();
    doBackgroundWork(params);
    return true;
}
private void doBackgroundWork(final JobParameters params) {
    new Thread(new Runnable() {
        @Override
        public void run() {
            // do something here
            jobScheduler =
(JobScheduler) getSystemService(JOB_SCHEDULER_SERVICE);
            checkForNewAnnouncements();
            if (jobCancelled) {
                return;
            }
            Log.d(TAG, "Rss finished");
            jobFinished(params, false);
        }
    }).start();
}
@Override
public boolean onStopJob(JobParameters params) {
    //turn off the notifications
    Log.d(TAG, "Rss cancelled");
    jobCancelled = true;
    return true;
}

private void checkForNewAnnouncements(){
    DataBaseHelper dbHelper = new
DataBaseHelper(getApplicationContext());
    getDataFromWeb(); // updates the articles but NOT the counter in the
database
    SystemClock.sleep(10000);
    ArrayList<HashMap<String, String>> newsArray = dbHelper.getAllRss();
//gets all the new articles from the database
    int newsCounter = newsArray.size(); // gets the size of that array
    int previousNewsCounter =

```

```

Integer.parseInt(dbHelper.getSettings("NEWS_COUNTER")); //gets the counter
from the last time the user launched the app
    Log.d("newsCounter", String.valueOf(newsCounter));
    Log.d("previousNewsCounter", String.valueOf(previousNewsCounter));

    if ( newsCounter > previousNewsCounter) {
        int newNumber = (newsCounter - previousNewsCounter) - 1 ;
        Log.d("newNumber", String.valueOf(newNumber));
        do{
            Log.d("inside DO-WHILE", String.valueOf(newNumber));

notificationIntent.setData(Uri.parse((newsArray.get(newNumber).get("link"))))
;
            stackBuilder = TaskStackBuilder.create(this);

stackBuilder.addNextIntentWithParentStack(notificationIntent);
            resultPendingIntent = stackBuilder.getPendingIntent(0,
PendingIntent.FLAG_UPDATE_CURRENT);
            builder = new NotificationCompat.Builder(this, "newNumber")
                .setSmallIcon(R.drawable.ic_rss)
                .setContentTitle("Νέα Ανακοίνωση")
                .setContentText(
newsArray.get(newNumber).get("title"))
                .setGroup("newNumber1")
                .setPriority(NotificationCompat.PRIORITY_DEFAULT);
            notificationManager = NotificationManagerCompat.from(this);
            notificationManager.notify(notificationID, builder.build());
            notificationID++;
            newNumber--;
        } while (newNumber >= 0);

    }
    dbHelper.changeSetting("NEWS_COUNTER",
String.valueOf(newsArray.size())); //UPDATES THE COUNTER AFTER NOTIFYING THE
USER
    Log.d(TAG, "Rss scanning");
}

private void createNotificationChannel() {
    // Create the NotificationChannel, but only on API 26+ because
    // the NotificationChannel class is new and not in the support
library
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        CharSequence name = "News Channel";
        String description = "Channel for student notification";
        int importance = NotificationManager.IMPORTANCE_DEFAULT;
        NotificationChannel channel = new
NotificationChannel("newNumber", name, importance);
        channel.setDescription(description);
        // Register the channel with the system; you can't change the
importance
        // or other notification behaviors after this
        NotificationManager notificationManager =
getSystemService(NotificationManager.class);
        notificationManager.createNotificationChannel(channel);
    }
}

class GetRssClass extends AsyncTask<Void, Void, ArrayList<HashMap<String,
String>>> {
    String mUrl;

    public GetRssClass(String url) {

```

```

        this.mUrl = url;
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }

    @Override
    protected ArrayList<HashMap<String, String>> doInBackground(Void...
voids) {
        ArrayList<HashMap<String, String>> result = new ArrayList<>();
        try {
            URL url = new URL(mUrl);

            ----- trust all websites with ssl CA-----
            -----
            try {
                HostnameVerifier(){
                    HostnameVerifier(){
                        public boolean verify(String hostname, SSLSession
session) {
                            return true;
                        }
                    }
                }
                SSLContext context = SSLContext.getInstance("TLS");
                context.init(null, new X509TrustManager[]{new
X509TrustManager(){
                    public void checkClientTrusted(X509Certificate[]
chain,
                    String authType)
                    throws CertificateException {}
                    public void checkServerTrusted(X509Certificate[]
chain,
                    String authType)
                    throws CertificateException {}
                    public X509Certificate[] getAcceptedIssuers() {
                        return new X509Certificate[0];
                    }
                }, new SecureRandom());
                HttpURLConnection.setDefaultSSLSocketFactory(
                    context.getSocketFactory());
            } catch (Exception e) { // should never happen
                e.printStackTrace();
            }

            -----

            HttpURLConnection connection = (HttpURLConnection)
url.openConnection();
            connection.setRequestMethod("GET");
            connection.setConnectTimeout(15000);
            connection.setDoInput( true );
            int responseCode = connection.getResponseCode();
            if (responseCode == HttpURLConnection.HTTP_OK) {
                InputStream inputStream = connection.getInputStream();
                result = parseXML(inputStream);
            } else {
                return null;
            }
        } catch (Exception e) {
            Log.d("Exception", e.getMessage());
            return null;
        }
        return result;
    }

```

```

    }

    @Override
    protected void onPostExecute(ArrayList<HashMap<String, String>>
result) {
        super.onPostExecute(result);
    }

    private ArrayList<HashMap<String, String>> parseXML(InputStream
inputStream)
        throws ParserConfigurationException, IOException,
SAXException, ParseException {

        DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
        DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();
        Document document = documentBuilder.parse(inputStream);
        Element element = document.getDocumentElement();

        NodeList itemlist = element.getElementsByTagName("item");
        NodeList itemChildren;

        Node currentItem;
        Node currentChild;

        ArrayList<HashMap<String, String>> items = new ArrayList<>();
        HashMap<String, String> currentMap;

        for (int i = 0; i < itemlist.getLength(); i++) {

            currentItem = itemlist.item(i);
            itemChildren = currentItem.getChildNodes();

            currentMap = new HashMap<>();

            for (int j = 0; j < itemChildren.getLength(); j++) {

                currentChild = itemChildren.item(j);

                if (currentChild.getNodeName().equalsIgnoreCase("title"))
{
                    // Log.d("Title",
String.valueOf(currentChild.getTextContent()));
                    currentMap.put("title",
currentChild.getTextContent());
                }

                if
(currentChild.getNodeName().equalsIgnoreCase("content:encoded")) {
                    // Log.d("description",
String.valueOf(currentChild.getTextContent()));
                    currentMap.put("content:encoded",
currentChild.getTextContent());
                }

                if
(currentChild.getNodeName().equalsIgnoreCase("pubDate")) {
                    // Log.d("Title",
String.valueOf(currentChild.getTextContent()));
                    currentMap.put("pubDate",
currentChild.getTextContent());
                }
            }
        }
    }

```



```

    }

    if (currentChild.getNodeName().equalsIgnoreCase("link"))
{
    currentMap.put("link",
currentChild.getTextContent());
    }

    }
    if (!currentMap.isEmpty()) {
        items.add(currentMap);
        String link = (String)currentMap.get("link");
        String title = (String)currentMap.get("title");
        String pubDate = (String)currentMap.get("pubDate");
        String content =
(String)currentMap.get("content:encoded");

        SimpleDateFormat inputPattern = new
SimpleDateFormat("EEE, d MMM yyyy HH:mm:ss Z");
        SimpleDateFormat outputPattern = new
SimpleDateFormat("yyyy-MM-dd HH:mm");
        Date date = inputPattern.parse(pubDate);
        String strDate = outputPattern.format(date);

        DataBaseHelper dataBaseHelper = new
DataBaseHelper(getApplicationContext());
        newsModel = new NewsModel(link, title, strDate, content,
"-1");

        dataBaseHelper.addOne(newsModel);
        dataBaseHelper.addOne(newsModel);

    }
    }
    return items;
}
}
public void getDataFromWeb() {
    GetRssClass fetchRss = new GetRssClass(mFeedUrl);
    fetchRss.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR);
}
}
}

```

### communityAdapter.java

```

package net.dimitrisiliou.myhmmmy.community;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import net.dimitrisiliou.myhmmmy.R;
import com.bumptech.glide.Glide;

import java.util.List;

public class communityAdapter extends

```

```

RecyclerView.Adapter<communityAdapter.CommunityViewHolder> {

    List<communityItem> mdata;
    communityCallback listener;

    public communityAdapter(List<communityItem> mdata, communityCallback
listener) {
        this.mdata = mdata;
        this.listener = listener;
    }

    @NonNull
    @Override
    public CommunityViewHolder onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {

        View view = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.item_community,parent,false) ;

        return new CommunityViewHolder(view);

    }

    @Override
    public void onBindViewHolder(@NonNull CommunityViewHolder holder, int
position) {

Glide.with(holder.itemView.getContext()).load(mdata.get(position).getImage())
    .into(holder.imgport);

    }

    @Override
    public int getItemCount() {
        return mdata.size();
    }

    public class CommunityViewHolder extends RecyclerView.ViewHolder {

        ImageView imgport;

        public CommunityViewHolder(@NonNull View itemView) {
            super(itemView);

            imgport = itemView.findViewById(R.id.item_community_img);
            itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    listener.onCommunityItemClick(getAdapterPosition());
                }
            });
        }
    }
}

```

## communityCallback.java

```

package net.dimitrisiliou.myhmmmy.community;

public interface communityCallback {

    void onCommunityItemClick(int pos);

}

```

## communityFragment.java

```

package net.dimitrisiliou.myhmmmy.community;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.net.Uri;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import net.dimitrisiliou.myhmmmy.R;
import net.dimitrisiliou.myhmmmy.database.DataBaseHelper;

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

public class communityFragment extends Fragment implements communityCallback
{

    List<communityItem> mdata;
    RecyclerView rv_community;
    communityAdapter communityAdapter;

    public communityFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_community, container,
false);
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle
savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        DataBaseHelper dbHelper = new DataBaseHelper(getContext());
        String old_studies = dbHelper.getSettings("STUDIES");
        rv_community = view.findViewById(R.id.rv_community);
        // create a list of items
        mdata = new ArrayList<>();

```

```

        if(old_studies.equals("hmmy_choice") ||
old_studies.equals("ele_choice")) {
            mdata.add(new communityItem(R.drawable.chaticon));
        }else {
            mdata.add(new communityItem(R.drawable.chaticon));
            mdata.add(new communityItem(R.drawable.chaticon_cied));
        }
        mdata.add(new communityItem(R.drawable.notesicon));
        mdata.add(new communityItem(R.drawable.wecare));
        mdata.add(new communityItem(R.drawable.hmmyfbbanner));
        communityAdapter = new communityAdapter(mdata,this);
        // setup grid recyclerview
        rv_community.setLayoutManager(new
GridLayoutManager(getActivity(),1));
        rv_community.setAdapter(communityAdapter);
    }

    @Override
    public void onCommunityItemClick(int pos) {
        DataBaseHelper dbHelper = new DataBaseHelper(getContext());
        String old_studies = dbHelper.getSettings("STUDIES");
        if(old_studies.equals("hmmy_choice") ||
old_studies.equals("ele_choice")) {
            if (pos == 0) {
                Uri uri =
Uri.parse("https://www.facebook.com/groups/elec.teipat/");
                Intent intent = new Intent(Intent.ACTION_VIEW, uri);
                startActivity(intent);
            } else if (pos == 1) {
                Uri uri = Uri.parse("https://drive.google.com/drive/folders/12-
fJAINrwViihtlYXPh5-dendcuGDmqX?usp=sharing");
                Intent intent = new Intent(Intent.ACTION_VIEW, uri);
                startActivity(intent);
            }else if (pos == 2) {
                Uri uri = Uri.parse("https://wecare.uop.gr/");
                Intent intent = new Intent(Intent.ACTION_VIEW, uri);
                startActivity(intent);
            }else if (pos == 3) {
                Uri uri =
Uri.parse("https://www.facebook.com/%CE%A4%CE%BC%CE%AE%CE%BC%CE%B1-
%CE%97%CE%BB%CE%B5%CE%BA%CF%84%CF%81%CE%BF%CE%BB%CF%8C%CE%B3%CF%89%CE%BD-
%CE%9C%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CE%BA%CF%8E%CE%BD-
%CE%9C%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CE%BA%CF%8E%CE%BD-
%CE%A5%CF%80%CE%BF%CE%BB%CE%BF%CE%B3%CE%B9%CF%83%CF%84%CF%8E%CE%BD-
%CE%A0%CE%B1%CE%BD-
%CE%A0%CE%B5%CE%BB%CE%BF%CF%80%CE%BF%CE%BD%CE%BD%CE%AE%CF%83%CE%BF%CF%85-
526093927472950/");
                Intent intent = new Intent(Intent.ACTION_VIEW, uri);
                startActivity(intent);
            }

            }else {
                if (pos == 0) {
                    Uri uri =
Uri.parse("https://www.facebook.com/groups/elec.teipat/");
                    Intent intent = new Intent(Intent.ACTION_VIEW, uri);
                    startActivity(intent);
                } else if (pos == 1) {
                    Uri uri = Uri.parse("https://www.facebook.com/groups/tesyd");
                    Intent intent = new Intent(Intent.ACTION_VIEW, uri);
                    startActivity(intent);
                }
            }
        }
    }
}

```

```

    }else if (pos == 2) {
        Uri uri =
Uri.parse("https://drive.google.com/drive/folders/12-fJAINrwViiht1YXPh5-
dendcuGDMqX?usp=sharing");
        Intent intent = new Intent(Intent.ACTION_VIEW, uri);
        startActivity(intent);
    }else if (pos == 3) {
        Uri uri = Uri.parse("https://wecare.uop.gr/");
        Intent intent = new Intent(Intent.ACTION_VIEW, uri);
        startActivity(intent);
    }
    else if (pos == 4) {
        Uri uri =
Uri.parse("https://www.facebook.com/%CE%A4%CE%BC%CE%AE%CE%BC%CE%B1-
%CE%97%CE%BB%CE%B5%CE%BA%CF%84%CF%81%CE%BF%CE%BB%CF%8C%CE%B3%CF%89%CE%BD-
%CE%9C%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CE%BA%CF%8E%CE%BD-
%CE%9C%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CE%BA%CF%8E%CE%BD-
%CE%A5%CF%80%CE%BF%CE%BB%CE%BF%CE%B3%CE%B9%CF%83%CF%84%CF%8E%CE%BD-
%CE%A0%CE%B1%CE%BD-
%CE%A0%CE%B5%CE%BB%CE%BF%CF%80%CE%BF%CE%BD%CE%BD%CE%AE%CF%83%CE%BF%CF%85-
526093927472950/");
        Intent intent = new Intent(Intent.ACTION_VIEW, uri);
        startActivity(intent);
    }
}
}}

```

### communityItem.java

```

package net.dimitrisiliou.myhmmmy.community;

import java.io.Serializable;

public class communityItem implements Serializable {

    private int image;
    private String title,description;

    public communityItem(){
    }

    public communityItem(int image) {
        this.image = image;
    }

    public communityItem(int image, String description) {
        this.image = image;
        this.title = description;
    }

    public communityItem(int image, String title, String description) {
        this.image = image;
        this.title = title;
        this.description = description;
    }

    public int getImage() {
        return image;
    }

    public void setImage(int image) {

```

```

        this.image = image;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }
}

```

### CourseFragmentDetails.java

```

package net.dimitrisiliou.myhmmmy.courses;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import com.google.android.material.bottomsheet.BottomSheetDialogFragment;

import net.dimitrisiliou.myhmmmy.R;
import net.dimitrisiliou.myhmmmy.database.DataBaseHelper;

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

public class CourseFragmentDetails extends BottomSheetDialogFragment {
    TextView teiTitle, hmmyid, semester, hours, lab, tutoring, teiID, hmmyTitle;
    String titleTEI, titleHMMY, idHMMY, idTEI, sem, tutor, hour, labs;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        Bundle bundle = getArguments();

        if (bundle != null) {
            titleTEI = bundle.getString("teiTitle");
            titleHMMY = bundle.getString("hmmyTitle");
            idHMMY = bundle.getString("hmmyid");
        }
    }
}

```

```

        idTEI = bundle.getString("teiId");
        sem = bundle.getString("semester");
        tutor = bundle.getString("tutoring");
        hour = bundle.getString("hours");
        labs = bundle.getString("lab");
    }

    return inflater.inflate(R.layout.fragment_courses_details, container,
false);
}
@Override
public void onViewCreated(@NonNull View view, @Nullable Bundle
savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);

    teiID = getView().findViewById(R.id.tei_idview);
    teiTitle =getView().findViewById(R.id.teicourseview);
    hmmyid =getView().findViewById(R.id.hmmy_idview);
    semester = getView().findViewById(R.id.hmmy_semesterview);
    hours = getView().findViewById(R.id.hmmy_hourview);
    lab = getView().findViewById(R.id.hmmy_labview);
    tutoring = getView().findViewById(R.id.hmmy_tutoringview);
    hmmyTitle = getView().findViewById(R.id.hmmytitleview);

    teiTitle.setText(titleTEI);
    teiID.setText(idTEI);
    teiTitle.setText(titleTEI);
    hmmyid.setText(idHMMY);
    semester.setText(sem);
    hours.setText(hour);
    lab.setText(labs);
    tutoring.setText(tutor);
    hmmyTitle.setText(titleHMMY);
}

public CourseFragmentDetails() {
    // Required empty public constructor
}
}

```

### CourseHmmyMatchingModel.java

```

package net.dimitrisiliou.myhmmy.courses;

public class CourseHmmyMatchingModel {
    private String id;
    private String title;
    private String semester;
    private String hours;
    private String lab;
    private String tutor;

    public CourseHmmyMatchingModel(String id, String title, String semester,
String hours, String lab, String tutor) {
        this.id = id;
        this.title = title;
        this.semester = semester;
        this.hours = hours;
        this.lab = lab;
        this.tutor = tutor;
    }
}

```

```

public String getId() {
    return id;
}

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

public String getTitle() {
    return title;
}

public void setTitle(String title) {
    this.title = title;
}

public String getSemester() {
    return semester;
}

public void setSemester(String semester) {
    this.semester = semester;
}

public String getHours() {
    return hours;
}

public void setHours(String hours) {
    this.hours = hours;
}

public String getLab() {
    return lab;
}

public void setLab(String lab) {
    this.lab = lab;
}

public String getTutor() {
    return tutor;
}

public void setTutor(String tutor) {
    this.tutor = tutor;
}
}

```

### CourseMatchingAdapter.java

```

package net.dimitrisiliou.myhmmmy.courses;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.TextView;

```



```

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import net.dimitrisiliou.myhmmmy.R;

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

public class CourseMatchingAdapter extends
RecyclerView.Adapter<CourseMatchingAdapter.CourseMatchingViewHolder>
implements Filterable {
    private List<CourseTeiMatchingModel> courseList;
    private List<CourseTeiMatchingModel> courseListFull;
    private OnNoteListener mOnNoteListener;

    public class CourseMatchingViewHolder extends RecyclerView.ViewHolder
implements View.OnClickListener {
        TextView textView1;
        TextView textView2;
        OnNoteListener onNoteListener;

        CourseMatchingViewHolder(View itemView, OnNoteListener
onNoteListener) {

            super(itemView);
            textView1 = itemView.findViewById(R.id.text_view1);
            textView2 = itemView.findViewById(R.id.text_view2);
            this.onNoteListener = onNoteListener;

            itemView.setOnClickListener(this);
        }

        @Override
        public void onClick(View v) {
            onNoteListener.onNoteClick(getAdapterPosition());
        }
    }

    public CourseMatchingAdapter(List<CourseTeiMatchingModel> courseList,
OnNoteListener onNoteListener) {
        this.courseList = courseList;
        courseListFull = new ArrayList<>(courseList);
        this.mOnNoteListener = onNoteListener;
    }

    @NonNull
    @Override
    public CourseMatchingViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
        View v =
LayoutInflater.from(parent.getContext()).inflate(R.layout.course_matching_ite
m,
                parent, false);
        return new CourseMatchingViewHolder(v,mOnNoteListener);
    }

    @Override
    public void onBindViewHolder(@NonNull CourseMatchingViewHolder holder,
int position) {
        CourseTeiMatchingModel currentItem = courseList.get(position);
        holder.textView1.setText(currentItem.getId());
        holder.textView2.setText(currentItem.getTitle());
    }
}

```

```

@Override
public int getItemCount() {
    return courseList.size();
}
@Override
public Filter getFilter() {
    return exampleFilter;
}
private Filter exampleFilter = new Filter() {
    @Override
    protected FilterResults performFiltering(CharSequence constraint) {
        List<CourseTeiMatchingModel> filteredList = new ArrayList<>();
        if (constraint == null || constraint.length() == 0) {
            filteredList.addAll(courseListFull);
        } else {
            String filterPattern =
constraint.toString().toLowerCase().trim();
            for (CourseTeiMatchingModel item : courseListFull) {
                if
(item.getTitle().toLowerCase().contains(filterPattern)) {
                    filteredList.add(item);
                }
            }
        }
        FilterResults results = new FilterResults();
        results.values = filteredList;
        return results;
    }
    @Override
    protected void publishResults(CharSequence constraint, FilterResults
results) {
        courseList.clear();
        courseList.addAll((List) results.values);
        notifyDataSetChanged();
    }
};
public interface OnNoteListener {
    void onNoteClick(int position);
}
}

```

### CourseMatchingFragment.java

```

package net.dimitrisiliou.myhmmmy.courses;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.view.inputmethod.EditorInfo;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;

```

```

import androidx.recyclerview.widget.RecyclerView;

import net.dimitrisiliou.myhmmmy.R;
import net.dimitrisiliou.myhmmmy.database.DataBaseHelper;

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

public class CourseMatchingFragment extends Fragment implements
CourseMatchingAdapter.OnNoteListener {
    private CourseMatchingAdapter adapter;
    private List<CourseTeiMatchingModel> courseList = new ArrayList<>();
    RecyclerView RvCourses;
    String hmmyTitle, semester, hours, lab, tutoring;

    public CourseMatchingFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        setHasOptionsMenu(true);
        View rootView = inflater.inflate(R.layout.fragment_courses,
container, false);
        Toolbar myToolbar = rootView.findViewById(R.id.toolbarCourses);
        ((AppCompatActivity) getActivity()).setSupportActionBar(myToolbar);
        ((AppCompatActivity)
getActivity()).getSupportActionBar().setTitle(null);
        ((AppCompatActivity)
getActivity()).getSupportActionBar().setSubtitle("Αναζήτηση");

        return rootView;
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle
savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        fillCourseList();
        setUpRecyclerView();
    }

    private void fillCourseList() {
        DataBaseHelper dataBaseHelper = new DataBaseHelper(getActivity());
        String old_studies = dataBaseHelper.getSettings("STUDIES");

        if (old_studies.equals("cied_choice")) {
            courseList.addAll(dataBaseHelper.getAllCiedCourses());
        } else if (old_studies.equals("ele_choice")) {
            courseList.addAll(dataBaseHelper.getAllFelCourses());
        }
    }

    private void setUpRecyclerView() {
        RvCourses = getView().findViewById(R.id.rv_courses);
        RvCourses.setHasFixedSize(true);
        RvCourses.setLayoutManager(new LinearLayoutManager(getContext()));
    }

```

```

        adapter = new CourseMatchingAdapter(courseList, this);
        RvCourses.setAdapter(adapter);
    }

    @Override
    public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
        super.onCreateOptionsMenu(menu, inflater);
        inflater.inflate(R.menu.course_matching_menu, menu);
        MenuItem searchItem = menu.findItem(R.id.action_search);
        final androidx.appcompat.widget.SearchView searchView =
            (androidx.appcompat.widget.SearchView) searchItem.getActionView();

        searchView.setQueryHint("Αναζήτηση");
        searchView.setImeOptions(EditorInfo.IME_ACTION_DONE);
        searchView.setOnQueryTextListener(new
            androidx.appcompat.widget.SearchView.OnQueryTextListener() {
                @Override
                public boolean onQueryTextSubmit(String query) {
                    return false;
                }

                @Override
                public boolean onQueryTextChange(String newText) {
                    adapter.getFilter().filter(newText);
                    return false;
                }
            });
    }

    @Override
    public void onNoteClick(int position) {

        CourseTeiMatchingModel mapTei = courseList.get(position);
        String hmmyid = mapTei.getCourseMatch();
        String teiId = mapTei.getId();
        String teiTitle = mapTei.getTitle();
        DataBaseHelper dataBaseHelper = new DataBaseHelper(getActivity());
        List<CourseHmmyMatchingModel> course =
            dataBaseHelper.getHmmyCourse(hmmyid);
        List<CourseHmmyMatchingModel> course2 = new ArrayList<>(course);
        course2.addAll(course);
        CourseHmmyMatchingModel map = course2.get(0);

        lab = String.valueOf(map.getLab());
        hours = String.valueOf(map.getHours());
        tutoring = String.valueOf(map.getTutor());
        semester = String.valueOf(map.getSemester());
        hmmyTitle = String.valueOf(map.getTitle());

        Bundle buddle = new Bundle();
        buddle.putString("lab", lab);
        buddle.putString("hours", hours);
        buddle.putString("tutoring", tutoring);
        buddle.putString("semester", semester);
        buddle.putString("hmmyTitle", hmmyTitle);
        buddle.putString("hmmyid", hmmyid);
        buddle.putString("teiId", teiId);
        buddle.putString("teiTitle", teiTitle);

        CourseFragmentDetails courseFragmentDetails = new
            CourseFragmentDetails();
    }

```

```

        courseFragmentDetails.setArguments(buddle);
        courseFragmentDetails.show(getActivity().getSupportFragmentManager(),
"tagname");
    }
}

```

### CourseTeiMatchingModel.java

```

package net.dimitrisiliou.myhmmmy.courses;

import android.os.Parcelable;

import androidx.annotation.NonNull;

import java.util.ArrayList;
import java.util.stream.Stream;

public class CourseTeiMatchingModel extends ArrayList<Parcelable> {
    private String id;
    private String title;
    private String courseMatch;

    public CourseTeiMatchingModel(String id, String title, String
courseMatch) {
        this.id = id;
        this.title = title;
        this.courseMatch = courseMatch;
    }

    public String getId() {
        return id;
    }

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

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getCourseMatch() {
        return courseMatch;
    }

    public void setCourseMatch(String courseMatch) {
        this.courseMatch = courseMatch;
    }

    @NonNull
    @Override
    public Stream<Parcelable> stream() {
        return null;
    }

    @NonNull
    @Override

```

```

public Stream<Parcelable> parallelStream() {
    return null;
}
}

```

### DataBaseHelper.java

```

package net.dimitrisiliou.myhmmmy.database;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

import net.dimitrisiliou.myhmmmy.courses.CourseHmmyMatchingModel;
import net.dimitrisiliou.myhmmmy.courses.CourseTeiMatchingModel;
import net.dimitrisiliou.myhmmmy.news.NewsModel;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class DataBaseHelper extends SQLiteOpenHelper {

    public static final String RSS_ID_LINK = "RSS_ID_LINK";
    public static final String RSS_TABLE = "RSS_TABLE";
    public static final String RSS_TITLE = "RSS_TITLE";
    public static final String RSS_CONTEXT = "RSS_CONTEXT";
    public static final String RSS_DATE = "RSS_DATE";
    public static final String RSS_CATEGORY = "RSS_CATEGORY";
    public static final String HMMY_COURSES_TABLE = "HMMY_COURSES_TABLE";
    public static final String TEI_COURSE_TITLE = "COURSE_TITLE";
    public static final String FEL_TEI_COURSES_TABLE =
"FEL_TEI_COURSES_TABLE";
    public static final String CIED_TEI_COURSES_TABLE =
"CIED_TEI_COURSES_TABLE";

    public static final String SETTINGS_TABLE = "SETTINGS_TABLE";
    public static final String SETTINGS_ID = "SETTINGS_ID";
    public static final String SETTINGS_CHOICE = "SETTINGS_CHOICE";

    private SQLiteDatabase myDataBase;
    private final Context myContext;
    private static final String DATABASE_NAME = "ymmh.db";
    public final static String DATABASE_PATH =
"/data/data/net.dimitrisiliou.myhmmmy/databases/";
    public static final int DATABASE_VERSION = 26;

    public DataBaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
}

```

```

        this.myContext = context;
    }
    //Create a empty database on the system
    public void createDatabase() throws IOException
    {
        boolean dbExist = checkDataBase();

        if(dbExist)
        {
            Log.v("DB Exists", "db exists");
            // By calling this method here onUpgrade will be called on a
            // writeable database, but only if the version number has been
            // bumped
            //onUpgrade(myDataBase, DATABASE_VERSION_old, DATABASE_VERSION);
        }
        else
        {
            this.getReadableDatabase();
            try
            {
                this.close();
                copyDataBase();
            }
            catch (IOException e)
            {
                throw new Error("Error copying database");
            }
        }
    }
    //Check database already exist or not
    private boolean checkDataBase()
    {
        boolean checkDB = false;
        try
        {
            String myPath = DATABASE_PATH + DATABASE_NAME;
            File dbfile = new File(myPath);
            checkDB = dbfile.exists();
        }
        catch(SQLiteException e)
        {
        }
        return checkDB;
    }
    //Copies the database from the local assets-folder to the just created
    empty database in the system folder
    private void copyDataBase() throws IOException
    {
        InputStream mInput = myContext.getAssets().open(DATABASE_NAME);
        String outFileName = DATABASE_PATH + DATABASE_NAME;
        OutputStream mOutput = new FileOutputStream(outFileName);
        byte[] mBuffer = new byte[2024];
        int mLength;
        while ((mLength = mInput.read(mBuffer)) > 0) {
            mOutput.write(mBuffer, 0, mLength);
        }
        mOutput.flush();
        mOutput.close();
        mInput.close();
    }

```

```

}
//delete database
public void db_delete()
{
    File file = new File(DATABASE_PATH + DATABASE_NAME);
    if(file.exists())
    {
        file.delete();
        System.out.println("delete database file.");
    }
}
//Open database
public void openDatabase() throws SQLException
{
    String myPath = DATABASE_PATH + DATABASE_NAME;
    myDataBase = SQLiteDatabase.openDatabase(myPath, null,
SQLiteDatabase.OPEN_READWRITE);
}

public synchronized void closeDataBase()throws SQLException
{
    if(myDataBase != null)
        myDataBase.close();
    super.close();
}

//this is called the first time a database is accessed.There should be
code in here to create a new database
@Override
public void onCreate(SQLiteDatabase db) {

    String createRssTable = " create table " + RSS_TABLE
        + " (" + RSS_ID_LINK + " text primary key, "
        + RSS_TITLE + " text not null, " + RSS_DATE
        + " date, " + RSS_CONTEXT + " text not null, "
        + RSS_CATEGORY + " text not null "
        + "));";

    db.execSQL(createRssTable);

}

// this is called if the database version number changes. It prevents
previous users apps from breaking when you change the database design.
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
{
    db.execSQL("DROP TABLE IF EXISTS "+ RSS_TABLE);
    onCreate(db);

}

public String getSettings(String id) {

    String queryString = "SELECT SETTINGS_CHOICE FROM SETTINGS_TABLE
WHERE " + SETTINGS_ID + " = '" + id + "' ";
    SQLiteDatabase db =this.getReadableDatabase();
    Cursor cursor = db.rawQuery(queryString, null);
    String settingChoice = null;

    if(cursor.moveToFirst()) {

```



```

        settingChoice = cursor.getString(0);
    }
    cursor.close();
    db.close();
    return settingChoice;
}

public List<CourseTeiMatchingModel> getAllCiedCourses() {

    List<CourseTeiMatchingModel> returnList = new ArrayList<>();
    String queryString = "SELECT * FROM " + CIED_TEI_COURSES_TABLE;
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.rawQuery(queryString, null);

    if(cursor.moveToFirst()) {
        do {
            String courseID = cursor.getString(0);
            String courseTitle = cursor.getString(1);
            String courseMatch = cursor.getString(2);

            CourseTeiMatchingModel newCourse = new
CourseTeiMatchingModel(courseID, courseTitle, courseMatch);
            returnList.add(newCourse);
        }while(cursor.moveToNext());
    }

    cursor.close();
    db.close();
    return returnList;
}

public List<CourseTeiMatchingModel> getAllFelCourses() {

    List<CourseTeiMatchingModel> returnList = new ArrayList<>();
    String queryString = "SELECT * FROM " + FEL_TEI_COURSES_TABLE;
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor1 = db.rawQuery(queryString, null);

    if(cursor1.moveToFirst()) {
        do {
            String courseID = cursor1.getString(0);
            String courseTitle = cursor1.getString(1);
            String courseMatch = cursor1.getString(2);
            CourseTeiMatchingModel newCourse = new
CourseTeiMatchingModel(courseID, courseTitle, courseMatch);
            returnList.add(newCourse);
        }while(cursor1.moveToNext());
    }

    cursor1.close();
    db.close();
    return returnList;
}

public List<CourseHmmyMatchingModel> getHmmyCourse(String hmmyid) {

    List<CourseHmmyMatchingModel> returnList = new ArrayList<>();

    if(hmmyid == null || hmmyid == " " || hmmyid == "" ||
hmmyid.isEmpty()) {

        String courseID = "-----";
        String courseTitle = " Χωρίς αντιστοίχιση";
    }
}

```

```

        String courseSemester = "-----";
        String courseHours = "-----";
        String courseLab = "-----";
        String courseTutor = "-----";

        CourseHmmyMatchingModel newCourse = new
CourseHmmyMatchingModel(courseID, courseTitle, courseSemester, courseHours, cours
eLab, courseTutor);
        returnList.add(newCourse);
    }
    else{
        String queryString = "SELECT * FROM " + HMMY_COURSES_TABLE + "
WHERE ID= " + "\"" + hmmyid+"\"";
        SQLiteDatabase db =this.getReadableDatabase();
        Cursor cursor = db.rawQuery(queryString, null);

        if(cursor.moveToFirst()) {
            do {
                String courseID = cursor.getString(0);
                String courseTitle = cursor.getString(1);
                String courseSemester = cursor.getString(2);
                String courseHours = cursor.getString(3);
                String courseLab = cursor.getString(4);
                String courseTutor = cursor.getString(5);

                CourseHmmyMatchingModel newCourse = new
CourseHmmyMatchingModel(courseID, courseTitle, courseSemester, courseHours, cours
eLab, courseTutor);
                returnList.add(newCourse);
            }while(cursor.moveToNext());
        }
        cursor.close();
        db.close();
        return returnList;
    }
}

public boolean changeSetting(String id, String choice) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv = new ContentValues();
    cv.put(SETTINGS_CHOICE, choice);
    db.update(SETTINGS_TABLE, cv, "SETTINGS_ID = '" + id + "'", null);
    db.close();
    return true;
}

public boolean addOne(NewsModel rssModel) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv = new ContentValues();
    cv.put(RSS_ID_LINK, rssModel.getLink_id());
    cv.put(RSS_TITLE, rssModel.getTitle());
    cv.put(RSS_DATE, String.valueOf(rssModel.getPubDate()));
    cv.put(RSS_CONTEXT, rssModel.getContext());
    cv.put(RSS_CATEGORY, rssModel.getCategory());

    db.insert(RSS_TABLE, null, cv);
    db.close();
    return true;
}

public ArrayList<HashMap<String, String>> getAllRss() {

```

```

        //get data from the database

        ArrayList<HashMap<String, String>> returnList = new ArrayList<>();

        String queryString = "SELECT * FROM " + RSS_TABLE + " ORDER BY
RSS_DATE DESC";
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery(queryString, null);

        if (cursor.moveToFirst()) {
            //loop through the cursor (result set) and create new rss objrcts
            and put them into the return list
            do {
                HashMap<String, String> currentMap = new HashMap<>();
                currentMap.put("title", cursor.getString(1));
                currentMap.put("link", cursor.getString(0));
                currentMap.put("pubDate", cursor.getString(2));
                currentMap.put("content:encoded", cursor.getString(3));
                currentMap.put("category", cursor.getString(4));

                returnList.add(currentMap);

            } while (cursor.moveToNext());

        }
        cursor.close();
        db.close();
        return returnList;
    }
}

```

### MapAdapter.java

```

package net.dimitrisiliou.myhmmmy.map;

import android.content.Intent;
import android.net.Uri;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import net.dimitrisiliou.myhmmmy.R;
import net.dimitrisiliou.myhmmmy.professors.ProfessorsItem;

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

public class MapAdapter extends
RecyclerView.Adapter<MapAdapter.MapViewHolder> implements Filterable {

```

```

List<MapItem> mapList;
MapFragment mContext;
private List<MapItem> mapListFull;

public MapAdapter(MapFragment context, List<MapItem> mdata) {
    this.mapList = mdata;
    mContext = context;
    mapListFull = new ArrayList<>(mapList);
}
private int lastPosition = -1;
@Override
public MapAdapter.MapViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {

    View view = LayoutInflater.from(parent.getContext())
        .inflate(R.layout.item_map,parent,false);

    return new MapAdapter.MapViewHolder(view);
}

@Override
public void onBindViewHolder(@NonNull MapAdapter.MapViewHolder holder,
int position) {
    lastPosition = position;

    holder.tv_name.setText(mapList.get(position).getName());
    holder.tv_desc.setText(mapList.get(position).getDesc());
    holder.tv_website.setOnClickListener(new
MapAdapter.onWebsiteClick(holder.getAdapterPosition()));
}

class onWebsiteClick implements View.OnClickListener {
    int position;

    public onWebsiteClick(int adapterPosition) {
        this.position = adapterPosition;
    }

    @Override
    public void onClick(View view) {
        MapItem map = mapList.get(position);
        Uri uri = Uri.parse(map.getMap());
        Intent intent = new Intent(Intent.ACTION_VIEW,uri);
        mContext.startActivity(intent);
        Log.d("CREATION", String.valueOf(uri));
    }
}

@Override
public int getItemCount() {
    return mapList.size();
}

public class MapViewHolder extends RecyclerView.ViewHolder {

    TextView tv_name,tv_desc;
    ImageView img;
    Button tv_email, tv_website;

    public MapViewHolder(@NonNull View itemView) {
        super(itemView);
}

```

```

        tv_name = itemView.findViewById(R.id.map_item_name) ;
        tv_desc = itemView.findViewById(R.id.map_item_desc);
        tv_website = itemView.findViewById(R.id.map_website);
    }
}

@Override
public Filter getFilter() {
    return exampleFilter;
}
private Filter exampleFilter = new Filter() {
    @Override
    protected FilterResults performFiltering(CharSequence constraint) {
        List<MapItem> filteredList = new ArrayList<>();
        if (constraint == null || constraint.length() == 0) {
            filteredList.addAll(mapListFull);
        } else {
            String filterPattern =
constraint.toString().toLowerCase().trim();
            for (MapItem item : mapListFull) {
                if (item.getName().toLowerCase().contains(filterPattern)
) {
                    filteredList.add(item);
                }
            }
        }
        FilterResults results = new FilterResults();
        results.values = filteredList;
        return results;
    }

    @Override
    protected void publishResults(CharSequence constraint, FilterResults
results) {
        mapList.clear();
        mapList.addAll((List) results.values);
        notifyDataSetChanged();
    }
};
}

```

### MapFragment.java

```

package net.dimitrisiliou.myhmmmy.map;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.view.inputmethod.EditorInfo;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

```

```

import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import net.dimitrisiliou.myhmmmy.R;

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

public class MapFragment extends Fragment {

    RecyclerView RvMap;
    MapAdapter adapter;
    List<MapItem> mdata;

    public MapFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        setHasOptionsMenu(true);
        View rootView = inflater.inflate(R.layout.fragment_map, container,
false);
        Toolbar myToolbar = rootView.findViewById(R.id.toolbarMap);
        ((AppCompatActivity) getActivity()).setSupportActionBar(myToolbar);
        ((AppCompatActivity)
getActivity()).getSupportActionBar().setTitle(null);
        ((AppCompatActivity)
getActivity()).getSupportActionBar().setSubtitle("Αίθουσες, Εργαστήρια κ.α");
        return rootView;
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle
savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        RvMap = view.findViewById(R.id.rv_map);

        MapItem item = new MapItem("Κτήριο Η'", "1ος Όροφος: Εργαστήρια
H1.01, H1.02, H1.04 και Γραφεία H1.03, H1.05 - H1.12 \n2ος Όροφος: Αίθουσα
Πολλαπλών Χρήσεων H2.02, Εργαστήριο H2.03 και Γραφεία H2.01, H2.04 -
H2.08", "https://goo.gl/maps/FVHdgLBoF2hEDTkW8");
        MapItem item1 = new MapItem("Κεντρικό Κτήριο Σχολής
Μηχανικών", "Ισόγειο: Γραμματεία, Εργαστήρια ΗΛΜ - ΣΧΓ, Αίθουσα Α19 \n1ος
Όροφος: Εργαστήρια ΗΛΝ, ΨΗΦ, ΚΥΚ, ΣΗΕ, ΜΙΚ, ΙΣΧ, ΚΙΝ, ΜΕΤ, PLC και
Φ/Β", "https://goo.gl/maps/9BpgXhKB7FHF7yFe7");
        MapItem item2 = new MapItem("Πτέρυγα Ζ'", "Ισόγειο: Εργαστήριο Η/Υ Ζ1
και Αίθουσες Ζ4, Ζ5 \n1ος Όροφος: Αίθουσες Ζ6 - Ζ12 \n2ος Όροφος: Εργαστήριο
ΣΑΕ και Αίθουσες Ζ17, Ζ18", "https://goo.gl/maps/C73Rp8Tp1BTXTbBR9");
        MapItem item3 = new MapItem("Κτήριο Ε'", "Ισόγειο: Εργαστήριο
Ε1", "https://goo.gl/maps/SB9rsL35JFc8oSS7A");
        MapItem item4 = new MapItem("Κτήριο Κ'", "Ισόγειο: Αίθουσες Κ0.01 -
Κ0.04 και Γραφεία Κ0.05 - Κ0.15 \n1ος Όροφος: Αίθουσες Κ1.07 και Κ1.12 \n2ος
Όροφος: Γραφεία και Εργαστήρια Κ2.01 -
Κ2.14", "https://goo.gl/maps/m12iPzwinbJksfmdA" );
        MapItem item5 = new MapItem("Μεγάλο
Αμφιθέατρο", "ΑΜΦ1", "https://goo.gl/maps/BVEBJS7gRy2iVgdB6");

```

```

        MapItem item6 = new MapItem("Μικρό Αμφιθέατρο
", "ΑΜΦ2", "https://goo.gl/maps/soGfyue3vh2CZQkj6");
        MapItem item7 = new
MapItem("Βιβλιοθήκη", "", "https://goo.gl/maps/A7zN65PJVXjxuzKG6");
        MapItem item8 = new MapItem("Φοιτητική
Εστία", null, "https://goo.gl/maps/iy2ZFxcPD8cUSgY57");
        MapItem item9 = new
MapItem("Εστιατόριο", "", "https://www.google.com/maps/place/38%C2%B013'10.1%22
N+21%C2%B044'49.2%22E/@38.2194729,21.74701,17z/data=!3m1!4b1!4m5!3m4!1s0x0:0x
0!8m2!3d38.2194729!4d21.74701");
        MapItem item10 = new
MapItem("Κυλικείο", "", "https://goo.gl/maps/nS9V3P83xWAVqHYy8");
        MapItem item11 = new MapItem("Συνεδριακό
Κέντρο", "", "https://goo.gl/maps/gAQeTEHmofcurh3SA");
        MapItem item12 = new MapItem("Ερευνητικό Εργαστήριο 'Εξυπνο
Σπίτι'", "", "https://goo.gl/maps/ayV9sHVYqgN1PHY9");

        mdata = new ArrayList<>();
        mdata.add(item);
        mdata.add(item1);
        mdata.add(item2);
        mdata.add(item3);
        mdata.add(item4);
        mdata.add(item5);
        mdata.add(item6);
        mdata.add(item7);
        mdata.add(item8);
        mdata.add(item9);
        mdata.add(item9);
        mdata.add(item10);
        mdata.add(item11);
        mdata.add(item12);

        // setup adapter and recyclerview
        RvMap.setLayoutManager(new LinearLayoutManager(getContext()));
        adapter = new MapAdapter(this, mdata);
        RvMap.setAdapter(adapter);

    }
    @Override
    public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
        super.onCreateOptionsMenu(menu, inflater);
        inflater.inflate(R.menu.course_matching_menu, menu);
        MenuItem searchItem = menu.findItem(R.id.action_search);
        final androidx.appcompat.widget.SearchView searchView =
(androidx.appcompat.widget.SearchView) searchItem.getActionView();

        searchView.setQueryHint("Αναζήτηση");
        searchView.setImeOptions(EditorInfo.IME_ACTION_DONE);
        searchView.setOnQueryTextListener(new
androidx.appcompat.widget.SearchView.OnQueryTextListener() {
            @Override
            public boolean onQueryTextSubmit(String query) {
                return false;
            }
            @Override
            public boolean onQueryTextChange(String newText) {
                adapter.getFilter().filter(newText);
                return false;
            }
        });
    }
}
}

```

## MapItem.java

```
package net.dimitrisiliou.myhmmmy.map;

public class MapItem {

    private String name, desc, map;

    public MapItem(String name, String desc, String website) {
        this.name = name;
        this.desc = desc;
        this.map = website;
    }

    @Override
    public String toString() {
        return "MapItem{" +
            "name='" + name + '\'' +
            ", desc='" + desc + '\'' +
            ", website='" + map + '\'' +
            '}';
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getDesc() {
        return desc;
    }

    public void setDesc(String desc) {
        this.desc = desc;
    }

    public String getMap() {
        return map;
    }

    public void setMap(String map) {
        this.map = map;
    }
}
```

## NewsAdapter.java

```
package net.dimitrisiliou.myhmmmy.news;

import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.text.Html;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
```



```

import androidx.cardview.widget.CardView;
import androidx.recyclerview.widget.RecyclerView;
import net.dimitrisiliou.myhmmmy.R;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import java.util.Locale;
import java.util.Objects;

public class NewsAdapter extends RecyclerView.Adapter<NewsAdapter.ViewHolder>
{
    private ArrayList<HashMap<String, String>> mDataset = new ArrayList<>();
    private Context mContext;
    private int lastPosition = -1;

    public static class ViewHolder extends RecyclerView.ViewHolder {
        public TextView tv_title, tv_date, tv_newannounce ;
        public ImageView iv_share;
        public CardView cardView;

        public ViewHolder(View v) {
            super(v);
            tv_title = (TextView) v.findViewById(R.id.title);
            tv_date = (TextView) v.findViewById(R.id.date);
            iv_share = (ImageView) v.findViewById(R.id.share);
            cardView = (CardView) v.findViewById(R.id.card_view);
            tv_newannounce = (TextView) v.findViewById(R.id.newannounce);
        }
    }

    public NewsAdapter(Context context, ArrayList<HashMap<String, String>>
dataset) {
        mContext = context;
        mDataset = dataset;
    }

    @Override
    public NewsAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        View v = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.rss_list_item, parent, false);

        ViewHolder vh = new ViewHolder(v);
        return vh;
    }

    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {
        // position should never be final
        lastPosition = position;

        HashMap<String, String> map = mDataset.get(position);
        holder.tv_title.setText(map.get("title"));
        String fromHtml =
Html.fromHtml(map.get("content:encoded")).toString();
        String lastStr = fromHtml.substring(2, fromHtml.length() - 1);
        holder.tv_date.setText(map.get("pubDate"));
        holder.cardView.setOnClickListener(new

```

```

onCardClick(holder.getAdapterPosition());
    holder.iv_share.setOnClickListener(new
onPostShare(holder.getAdapterPosition()));

//shows the "NEW" TAG
    String dateString = (String) holder.tv_date.getText();
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm",
Locale.UK);
    Date date;
    try {
        date = sdf.parse(dateString);
        assert date != null;
        int dayDiff = daysBetween(date, new Date()) - 1;
        if ( dayDiff <=4 ) { // show new tag for articles that were
published up to 4 days ago.
            holder.tv_newannounce.setText("NEO!");
        } else {
            holder.tv_newannounce.setText("");
        }
    } catch (ParseException e) {
        e.printStackTrace();
    }
}

private static int daysBetween(Date startDate, Date endDate) {
    int daysBetween = 0;
    while (startDate.before(endDate)) {
        startDate.setTime(startDate.getTime() + 86400000);
        daysBetween++;
    }
    return daysBetween;
}

class onCardClick implements View.OnClickListener {
    int position;

    public onCardClick(int adapterPosition) {
        this.position = adapterPosition;
    }

    @Override
    public void onClick(View view) {
        HashMap<String, String> map = mDataset.get(position);
        Log.d("Position", Objects.requireNonNull(map.get("link")));
        Uri uri = Uri.parse(map.get("link"));
        Intent intent = new Intent(Intent.ACTION_VIEW, uri);
        mContext.startActivity(intent);
    }
}

class onPostShare implements View.OnClickListener {
    int position;

    public onPostShare(int adapterPosition) {
        this.position = adapterPosition;
    }

    @Override
    public void onClick(View view) {
        HashMap<String, String> map = mDataset.get(position);
        Intent sendIntent = new Intent();
        sendIntent.setAction(Intent.ACTION_SEND);

```

```

        sendIntent.putExtra(Intent.EXTRA_TEXT,
            map.get("title") + " \n" + map.get("link"));
        sendIntent.setType("text/plain");
        mContext.startActivity(Intent.createChooser(sendIntent, "Share
this post"));
    }
}

@Override
public int getItemCount() {
    return mDataset.size();
}
}

```

### NewsFragment.java

```

package net.dimitrisiliou.myhmmmy.news;

import android.annotation.SuppressLint;
import android.app.ProgressDialog;
import android.app.job.JobInfo;
import android.app.job.JobScheduler;
import android.content.ComponentName;
import android.util.Log;import android.os.AsyncTask;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import net.dimitrisiliou.myhmmmy.BackroundUpdate.AppJobService;
import net.dimitrisiliou.myhmmmy.R;
import net.dimitrisiliou.myhmmmy.database.DataBaseHelper;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;
import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.security.SecureRandom;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import java.util.Objects;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpURLConnection;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.X509TrustManager;

```

```

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

import static android.content.Context.JOB_SCHEDULER_SERVICE;

public class NewsFragment extends Fragment {

    private RecyclerView mRecyclerView;
    private NewsAdapter mAdapter;
    private ArrayList<HashMap<String, String>> resultItems = new
ArrayList<>();
    private static String mFeedUrl =
"https://www.ece.uop.gr/announcement/feed/";
    NewsModel newsModel;
    DataBaseHelper dataBaseHelper;
    private ProgressDialog progressDialog;
    private boolean isLoading;

    public NewsFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        setHasOptionsMenu(true);
        View rootView = inflater.inflate(R.layout.activity_news, container,
false);

        getDataFromWeb();
        return rootView;
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle
savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        mRecyclerView = getView().findViewById(R.id.recycler_view);
        mRecyclerView.setHasFixedSize(true);
        mRecyclerView.setLayoutManager(new
LinearLayoutManager(getContext()));
        dataBaseHelper = new DataBaseHelper(getContext());
        mAdapter = new NewsAdapter(getContext(), resultItems);
        mRecyclerView.setAdapter(mAdapter);
    }

    public class GetRssClass extends AsyncTask<Void, Void,
ArrayList<HashMap<String, String>>> {
        String mUrl;

        public GetRssClass(String url) {
            this.mUrl = url;
        }

        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            showProgressDialog();
        }

        @Override
        protected ArrayList<HashMap<String, String>> doInBackground(Void...

```

```

voids) {
    ArrayList<HashMap<String, String>> result = new ArrayList<>();
    try {
        URL url = new URL(mUrl);

        //----- trust all websites with ssl CA-----
        -----
        try {
            HostnameVerifier(){
                public boolean verify(String hostname, SSLSession
                session) {
                    return true;
                }});
            SSLContext context = SSLContext.getInstance("TLS");
            context.init(null, new X509TrustManager[] {new
            X509TrustManager(){
                public void checkClientTrusted(X509Certificate[]
                chain,
                String authType)
                throws CertificateException {}
                public void checkServerTrusted(X509Certificate[]
                chain,
                String authType)
                throws CertificateException {}
                public X509Certificate[] getAcceptedIssuers() {
                    return new X509Certificate[0];
                }
            }}, new SecureRandom());
            HttpsURLConnection.setDefaultSSLConnectionFactory(
                context.getSocketFactory());
        } catch (Exception e) { // should never happen
            e.printStackTrace();
        }

        //-----
        HttpsURLConnection connection = (HttpsURLConnection)
        url.openConnection();
        connection.setRequestMethod("GET");
        connection.setConnectTimeout(10000);
        connection.setDoInput( true );
        int responseCode = connection.getResponseCode();
        if (responseCode == HttpURLConnection.HTTP_OK) {
            InputStream inputStream = connection.getInputStream();
            result = parseXML(inputStream);
        } else {
            return null;
        }
    } catch (Exception e) {
        Log.d("Exception", e.getMessage());
        return null;
    }
    return result;
}

@Override
protected void onPostExecute(ArrayList<HashMap<String, String>>
result) {
    super.onPostExecute(result);
    hideProgressDialog();
    if (result == null) {

```

```

        showToast("Please check your connection and try again.");
    }

    int before = resultItems.size();
    resultItems.clear();
    ArrayList<HashMap<String, String>> everyone =
dataBaseHelper.getAllRss();
    dataBaseHelper.changeSetting("NEWS_COUNTER",
String.valueOf(everyone.size())); //adds the complete number of items that
exists in the database
    resultItems.addAll(everyone);
    mAdapter.notifyItemRangeInserted(before, everyone.size());
    mRecyclerView.invalidate();
}

private ArrayList<HashMap<String, String>> parseXML(InputStream
inputStream)
    throws ParserConfigurationException, IOException,
SAXException, ParseException {

    DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
    DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();
    Document document = documentBuilder.parse(inputStream);
    Element element = document.getDocumentElement();

    NodeList itemlist = element.getElementsByTagName("item");
    NodeList itemChildren;

    Node currentItem;
    Node currentChild;

    String link = null;
    String title = null;
    String pubDate = null;
    String content = null;
    String newDateFormat = null;

    ArrayList<HashMap<String, String>> items = new ArrayList<>();
//the one that returns at the end of the execution
    HashMap<String, String> currentMap; //supported hashmap

    for (int i = 0; i < itemlist.getLength(); i++) { //for each
article

        currentItem = itemlist.item(i);
        itemChildren = currentItem.getChildNodes();
        currentMap = new HashMap<>();

        for (int j = 0; j < itemChildren.getLength(); j++) { //for
each tag inside the article
            currentChild = itemChildren.item(j);

            if (currentChild.getNodeName().equalsIgnoreCase("title"))
            {
                // Log.d("Title",
String.valueOf(currentChild.getTextContent()));
                currentMap.put("title",
currentChild.getTextContent());
                title = (String) currentChild.getTextContent();
            }
        }
    }
}

```

```

    }
    if
(currentChild.getNodeName().equalsIgnoreCase("content:encoded")) { //content
for a future use
        // Log.d("description",
String.valueOf(currentChild.getTextContent()));
        currentMap.put("content:encoded",
currentChild.getTextContent());
        content = (String) currentChild.getTextContent();
    }
    if
(currentChild.getNodeName().equalsIgnoreCase("pubDate")) {
        // Log.d("Title",
String.valueOf(currentChild.getTextContent()));
        currentMap.put("pubDate",
currentChild.getTextContent());
        pubDate = (String) currentChild.getTextContent();
        @SuppressWarnings("SimpleDateFormat")
SimpleDateFormat inputPattern = new
SimpleDateFormat("EEE, d MMM yyyy HH:mm:ss Z");
        @SuppressWarnings("SimpleDateFormat")
SimpleDateFormat outputPattern = new
SimpleDateFormat("yyyy-MM-dd HH:mm");
        Date date = inputPattern.parse(pubDate);
        newDateFormat = outputPattern.format(date);
    }
    if (currentChild.getNodeName().equalsIgnoreCase("link"))
{
        currentMap.put("link",
currentChild.getTextContent());
        link = (String) currentChild.getTextContent();
    }
    if (!currentMap.isEmpty()) {
        items.add(currentMap);
        newsModel = new NewsModel(link, title, newDateFormat,
content, "-1"); //category for a future use
        dbHelper.addOne(newsModel); //adds each new item in
the database
    }
}
return items;
}
}

private void showProgressDialog() {
    isLoading = true;
    progressDialog = new ProgressDialog(getContext());
    progressDialog.setCancelable(false);
    progressDialog.setMessage("Αναζήτηση");
    progressDialog.show();
}

private void hideProgressDialog() {
    isLoading = false;
    if (progressDialog != null) {
        progressDialog.cancel();
    }
}

public void getDataFromWeb() {
    GetRssClass fetchRss = new GetRssClass(mFeedUrl);
    fetchRss.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR);
}

```

```
}  
  
private void showToast(String msg) {  
    Toast.makeText(getActivity(), msg, Toast.LENGTH_SHORT).show();  
}  
}}
```

### NewsModel.java

```
package net.dimitrisiliou.myhmmmy.news;  
  
public class NewsModel {  
    private String link_id;  
    private String title;  
    private String pubDate;  
    private String context;  
    private String category;  
  
    public NewsModel(String link_id, String title, String pubDate, String  
context, String category) {  
        this.link_id = link_id;  
        this.title = title;  
        this.pubDate = pubDate;  
        this.context = context;  
        this.category = category;  
    }  
  
    public String getLink_id() {  
        return link_id;  
    }  
  
    public void setLink_id(String link_id) {  
        this.link_id = link_id;  
    }  
  
    public String getTitle() {  
        return title;  
    }  
  
    public void setTitle(String title) {  
        this.title = title;  
    }  
  
    public String getPubDate() {  
        return pubDate;  
    }  
  
    public void setPubDate(String pubDate) {  
        this.pubDate = pubDate;  
    }  
  
    public String getContext() {  
        return context;  
    }  
  
    public void setContext(String context) {  
        this.context = context;  
    }  
  
    public String getCategory() {  
        return category;  
    }  
}
```



```

public void setCategory(String category) {
    this.category = category;
}

@Override
public String toString() {
    return "RssModel{" +
        "link_id='" + link_id + '\'' +
        ", title='" + title + '\'' +
        ", pubDate='" + pubDate + '\'' +
        ", context='" + context + '\'' +
        '}';
}
}

```

### ProfessorsAdapter.java

```

package net.dimitrisiliou.myhmmmy.professors;

import android.content.Intent;
import android.net.Uri;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import net.dimitrisiliou.myhmmmy.R;

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

public class ProfessorsAdapter extends
RecyclerView.Adapter<ProfessorsAdapter.ProfessorsViewHolder> implements
Filterable {

    List<ProfessorsItem> profList;
    ProfessorsFragment mContext;
    private List<ProfessorsItem> profListFull;

    public ProfessorsAdapter(ProfessorsFragment context, List<ProfessorsItem>
mdata) {
        this.profList = mdata;
        mContext = context;
        profListFull = new ArrayList<>(profList);
    }
    private int lastPosition = -1;

    @Override
    public ProfessorsViewHolder onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {

        View view = LayoutInflater.from(parent.getContext())

```

```

        .inflate(R.layout.item_professors,parent,false);

        return new ProfessorsViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull ProfessorsViewHolder holder, int
position) {
        lastPosition = position;

        holder.tv_name.setText(profList.get(position).getName());
        holder.tv_desc.setText(profList.get(position).getDesc());
        holder.img.setImageResource(profList.get(position).getImg());
        holder.tv_email.setOnClickListener(new
onEmailClick(holder.getAdapterPosition()));
        holder.tv_website.setOnClickListener(new
onWebsiteClick(holder.getAdapterPosition()));
    }

    class onWebsiteClick implements View.OnClickListener {
        int position;

        public onWebsiteClick(int adapterPosition) {
            this.position = adapterPosition;
        }

        @Override
        public void onClick(View view) {
            ProfessorsItem map = profList.get(position);
            Uri uri = Uri.parse(map.getwebsite());
            Intent intent = new Intent(Intent.ACTION_VIEW,uri);
            mContext.startActivity(intent);
            Log.d("CREATION", String.valueOf(uri));
        }
    }

    class onEmailClick implements View.OnClickListener {
        int position;

        public onEmailClick(int adapterPosition) {
            this.position = adapterPosition;
        }

        @Override
        public void onClick(View view) {
            ProfessorsItem map = profList.get(position);
            Uri uri = Uri.parse(map.getEmail());
            Intent intent = new Intent(Intent.ACTION_VIEW,uri);
            mContext.startActivity(intent);
            Log.d("CREATION", String.valueOf(uri));
        }
    }

    public class ProfessorsViewHolder extends RecyclerView.ViewHolder {

        TextView tv_name,tv_desc;
        ImageView img;
        Button tv_email, tv_website;

        public ProfessorsViewHolder(@NonNull View itemView) {
            super(itemView);

```

```

        tv_name = itemView.findViewById(R.id.professors_item_name) ;
        tv_desc = itemView.findViewById(R.id.professors_item_desc);
        img = itemView.findViewById(R.id.professors_item_img);
        tv_email = itemView.findViewById(R.id.tv_email);
        tv_website = itemView.findViewById(R.id.tv_website);
    }
}
@Override
public int getItemCount() {
    return profList.size();
}
@Override
public Filter getFilter() {
    return exampleFilter;
}
private Filter exampleFilter = new Filter() {
    @Override
    protected FilterResults performFiltering(CharSequence constraint) {
        List<ProfessorsItem> filteredList = new ArrayList<>();
        if (constraint == null || constraint.length() == 0) {
            filteredList.addAll(profListFull);
        } else {
            String filterPattern =
constraint.toString().toLowerCase().trim();
            for (ProfessorsItem item : profListFull) {
                if (item.getName().toLowerCase().contains(filterPattern))
{
                    filteredList.add(item);
                }
            }
        }
        FilterResults results = new FilterResults();
        results.values = filteredList;
        return results;
    }
    @Override
    protected void publishResults(CharSequence constraint, FilterResults
results) {
        profList.clear();
        profList.addAll((List) results.values);
        notifyDataSetChanged();
    }
};
}
}

```

### ProfessorsFragment.java

```

package net.dimitrisiliou.myhmmmy.professors;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.view.inputmethod.EditorInfo;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;

```

```

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import net.dimitrisiliou.myhmmy.R;

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

/**
 * A simple {@link Fragment} subclass.
 */
public class ProfessorsFragment extends Fragment {

    RecyclerView RvProfessors;
    ProfessorsAdapter adapter;
    List<ProfessorsItem> mdata;

    public ProfessorsFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        setHasOptionsMenu(true);
        View rootView = inflater.inflate(R.layout.fragment_professors,
container, false);
        Toolbar myToolbar = rootView.findViewById(R.id.toolbarProf);
        ((AppCompatActivity) getActivity()).setSupportActionBar(myToolbar);
        ((AppCompatActivity)
getActivity()).getSupportActionBar().setTitle(null);
        ((AppCompatActivity)
getActivity()).getSupportActionBar().setSubtitle("Email και Πληροφορίες
Καθηγητών");
        return rootView;
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle
savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        RvProfessors = view.findViewById(R.id.rv_professors);

        // create list of professors items

        ProfessorsItem item = new ProfessorsItem("Παναγιώτης
Αλεφραγκής",getString(R.string.assistance_professor),R.drawable.alefrag,"mail
to:alefrag@uop.gr","https://www.ece.uop.gr/staff/alefragkis-panagiotis/");
        ProfessorsItem item2 = new ProfessorsItem("Γεώργιος
Αντωνόπουλος",getString(R.string.lecturer),R.drawable.user,"mailto:gantonopou
los@uop.gr","https://www.ece.uop.gr/staff/antonopoulos-georgios/");
        ProfessorsItem item3 = new ProfessorsItem("Χρήστος
Αντωνόπουλος",getString(R.string.assistance_professor),R.drawable.anton_ch,"m
ailto:gantonopoulos@uop.gr","https://www.ece.uop.gr/staff/christos-
antonopoulos/");
        ProfessorsItem item4 = new ProfessorsItem("Γιώργος

```

```

Ασημακόπουλος",getString(R.string.lecturer),R.drawable.asimakopoulo,"mailto:asim@uop.gr", "https://www.ece.uop.gr/staff/asimakopoulos-giorgos/");
    ProfessorsItem item5 = new ProfessorsItem("Πέτρος
Βλαχόπουλος",getString(R.string.professor),R.drawable.vlachopoulos,"mailto:petrosv@uop.gr", "https://www.ece.uop.gr/staff/vlachopoulos-petros/");
    ProfessorsItem item6 = new ProfessorsItem("Νικόλαος
Βώρος",getString(R.string.professor),R.drawable.voros,"mailto:voros@uop.gr", "https://www.ece.uop.gr/staff/voros-nikolaos/");
    ProfessorsItem item7 = new ProfessorsItem("Κωνσταντίνος
Γεωργάκας",getString(R.string.assistance_professor),R.drawable.georgakas,"mailto:kgeorgakas@uop.gr", "https://www.ece.uop.gr/staff/konstantinos-georgakas/");
    ProfessorsItem item8 = new ProfessorsItem("Αναστάσιος
Δροσόπουλος",getString(R.string.professor),R.drawable.drosopoulos,"mailto:drosop@uop.gr", "https://www.ece.uop.gr/staff/drosopoulos-anastasios/");
    ProfessorsItem item9 = new ProfessorsItem("Λάμπρος
Δρόσος",getString(R.string.professor),R.drawable.user,"mailto:ldrossos@uop.gr", "https://www.ece.uop.gr/staff/drosos-lampros/");
    ProfessorsItem item10 = new ProfessorsItem("Ιωάννης Δ.
Ζαχαράκης",getString(R.string.professor),R.drawable.zaharakis,"mailto:zaharakis@uop.gr", "https://www.ece.uop.gr/staff/zaharakis-ioannis/");
    ProfessorsItem item11 = new ProfessorsItem("Παναγιώτης
Ζέρβας",getString(R.string.assistance_professor),R.drawable.zervas,"mailto:pzervas@uop.gr", "https://www.ece.uop.gr/staff/panagiotis-zervas/");
    ProfessorsItem item12 = new ProfessorsItem("Δημήτρης
Καρέλης",getString(R.string.assistance_professor),R.drawable.karelis,"mailto:dkarelis@uop.gr", "https://www.ece.uop.gr/staff/karelis-dimitris/");
    ProfessorsItem item13 = new ProfessorsItem("Βασίλης
Καψάλης",getString(R.string.professor),R.drawable.kapsalis,"mailto:kapsalis@uop.gr", "https://www.ece.uop.gr/staff/kapsalis-vasilis/");
    ProfessorsItem item14 = new ProfessorsItem("Παρασκευάς
Κίτσος",getString(R.string.associate_professor),R.drawable.kitsos,"mailto:kitsos@uop.gr", "https://www.ece.uop.gr/staff/kitsos-paraskevas/");
    ProfessorsItem item15 = new ProfessorsItem("Ιωάννης
Κούγιας",getString(R.string.professor),R.drawable.kougias,"mailto:kougias@uop.gr", "https://www.ece.uop.gr/staff/kougias-ioannis/");
    ProfessorsItem item16 = new ProfessorsItem("Αθανάσιος
Κούτρας",getString(R.string.assistance_professor),R.drawable.koutras,"mailto:koutras@uop.gr", "https://www.ece.uop.gr/staff/koutras-athanasios/");
    ProfessorsItem item17 = new ProfessorsItem("Λάμπρος
Μπισδούνης",getString(R.string.professor),R.drawable.bisdounis,"mailto:bisdounis@uop.gr", "https://www.ece.uop.gr/staff/mpisdounis-lampros/");
    ProfessorsItem item18 = new ProfessorsItem("Μιχάλης
Παρασκευάς",getString(R.string.associate_professor),R.drawable.paraskevas,"mailto:mparask@uop.gr", "https://www.ece.uop.gr/staff/paraskevas-michalis/");
    ProfessorsItem item19 = new ProfessorsItem("Νίκος
Πετρέλλης",getString(R.string.associate_professor),R.drawable.petrellis,"mailto:nPetrellis@uop.gr", "https://www.ece.uop.gr/staff/petrellis-nikos/");
    ProfessorsItem item20 = new ProfessorsItem("Πολίτη
Χριστίνα",getString(R.string.assistance_professor),R.drawable.politi,"mailto:tpoliti@uop.gr", "https://www.ece.uop.gr/staff/christina-tania-politi/");
    ProfessorsItem item21 = new ProfessorsItem("Γιώργος
Σουλιώτης",getString(R.string.assistance_professor),R.drawable.souliotis,"mailto:gsoul@uop.gr", "https://www.ece.uop.gr/staff/souliotis-giorgos/");
    ProfessorsItem item22 = new ProfessorsItem("Ηλίας
Σταθάτος",getString(R.string.professor),R.drawable.stathatos,"mailto:estathatos@uop.gr", "https://www.ece.uop.gr/staff/stathatos-ilias/");
    ProfessorsItem item23 = new ProfessorsItem("Σπύρος
Συρμακέσης",getString(R.string.professor),R.drawable.sirmakessis,"mailto:syrm@a@uop.gr", "https://www.ece.uop.gr/staff/syrmakesis-spyros/");
    ProfessorsItem item24 = new ProfessorsItem("Νικόλαος
Σχοινιάς",getString(R.string.assistance_professor),R.drawable.user,"mailto:nsc_hinas@uop.gr", "https://www.ece.uop.gr/staff/schoinas-nikolaos/");

```

```

    ProfessorsItem item25 = new ProfessorsItem("Σωτηρόπουλος
Δημήτρης",getString(R.string.assistance_professor),R.drawable.sotiropoulos,"m
ailto:dg.sotiropoulos@uop.gr","https://www.ece.uop.gr/staff/sotiropoulos-
dimitris/");
    ProfessorsItem item26 = new ProfessorsItem("Βασίλειος
Ταμπακάς",getString(R.string.professor),R.drawable.tampakas,"mailto:tampakas@
uop.gr","https://www.ece.uop.gr/staff/vasileios-tampakas/");
    ProfessorsItem item27 = new ProfessorsItem("Ιωάννης
Τζήμας",getString(R.string.associate_professor),R.drawable.tzimas,"mailto:tzi
mas@uop.gr","https://www.ece.uop.gr/staff/tzimas-ioannis/");
    ProfessorsItem item28 = new ProfessorsItem("Βασίλειος
Τριανταφύλλου",getString(R.string.professor),R.drawable.triantafyllou,"mailto
:vtriantaf@uop.gr","https://www.ece.uop.gr/staff/triantafyllou-vasileios/");
    ProfessorsItem item29 = new ProfessorsItem("Ιωάννης
Τσακνάκης",getString(R.string.associate_professor),R.drawable.tsaknakis,"mail
to:jtsaknakis@uop.gr","https://www.ece.uop.gr/staff/ioannis-tsaknakis/");
    ProfessorsItem item30 = new ProfessorsItem("Λουκάς
Χαδέλλης",getString(R.string.professor),R.drawable.user,"mailto:loukas@uop.gr
","https://www.ece.uop.gr/staff/chadellis-loukas/");
    ProfessorsItem item31 = new ProfessorsItem("Βασίλειος
Χαραλαμπάκος",getString(R.string.assistance_professor),R.drawable.charalampak
os,"mailto:charalambakos@uop.gr","https://www.ece.uop.gr/staff/charalampakos-
vasileios/");
    ProfessorsItem item32 = new ProfessorsItem("Σωτήρης Π.
Χριστοδούλου",getString(R.string.assistance_professor),R.drawable.christodoul
ou,"mailto:sxristod@uop.gr","https://www.ece.uop.gr/staff/sotiris-p-
christodoulou/");

    mdata = new ArrayList<>();
    mdata.add(item);
    mdata.add(item2);
    mdata.add(item3);
    mdata.add(item4);
    mdata.add(item5);
    mdata.add(item6);
    mdata.add(item7);
    mdata.add(item8);
    mdata.add(item9);
    mdata.add(item10);
    mdata.add(item11);
    mdata.add(item12);
    mdata.add(item13);
    mdata.add(item14);
    mdata.add(item15);
    mdata.add(item16);
    mdata.add(item17);
    mdata.add(item18);
    mdata.add(item19);
    mdata.add(item20);
    mdata.add(item21);
    mdata.add(item22);
    mdata.add(item23);
    mdata.add(item24);
    mdata.add(item25);
    mdata.add(item26);
    mdata.add(item27);
    mdata.add(item28);
    mdata.add(item29);
    mdata.add(item30);
    mdata.add(item31);
    mdata.add(item32);

```

```

        // setup adapter and recyclerview

        RvProfessors.setLayoutManager(new LinearLayoutManager(getContext()));
        adapter = new ProfessorsAdapter(this, mdata);
        RvProfessors.setAdapter(adapter);

    }

    @Override
    public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
        super.onCreateOptionsMenu(menu, inflater);
        inflater.inflate(R.menu.course_matching_menu, menu);
        MenuItem searchItem = menu.findItem(R.id.action_search);
        final androidx.appcompat.widget.SearchView searchView =
            (androidx.appcompat.widget.SearchView) searchItem.getActionView();

        searchView.setQueryHint("Αναζήτηση");
        searchView.setImeOptions(EditorInfo.IME_ACTION_DONE);
        searchView.setOnQueryTextListener(new
            androidx.appcompat.widget.SearchView.OnQueryTextListener() {
                @Override
                public boolean onQueryTextSubmit(String query) {
                    return false;
                }
                @Override
                public boolean onQueryTextChange(String newText) {
                    adapter.getFilter().filter(newText);
                    return false;
                }
            });
    }
}

```

### ProfessorsItem.java

```

package net.dimitrisiliou.myhmy.professors;

public class ProfessorsItem {

    private String name, desc, email, website;
    private int img ;

    public ProfessorsItem(String name, String desc, int img, String email,
        String website) {
        this.name = name;
        this.desc = desc;
        this.img = img;
        this.email = email;
        this.website = website;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getwebsite() {

```

```

        return website;
    }

    public void setwebsite(String website) {
        this.website = website;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getDesc() {
        return desc;
    }

    public void setDesc(String desc) {
        this.desc = desc;
    }

    public int getImg() {
        return img;
    }

    public void setImg(int img) {
        this.img = img;
    }
}

```

### SettingsFragment.java

```

package net.dimitrisiliou.myhmmmy.settings;

import android.annotation.SuppressLint;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.job.JobInfo;
import android.app.job.JobScheduler;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.RadioButton;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;
import androidx.core.app.TaskStackBuilder;
import androidx.fragment.app.Fragment;

```



```

import net.dimitrisiliou.myhmmmy.R;
import net.dimitrisiliou.myhmmmy.WelcomeActivity;
import net.dimitrisiliou.myhmmmy.database.DataBaseHelper;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Objects;

import static android.content.ContentValues.TAG;
import static android.content.Context.JOB_SCHEDULER_SERVICE;

public class SettingsFragment extends Fragment implements
View.OnClickListener {

    RadioButton choice1, choice2, choice3, newsyes, newsno;

    public SettingsFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_settings, container,
false);
    }

    public void onViewCreated(@NonNull View view, @Nullable Bundle
savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        choice1 = (RadioButton) view.findViewById(R.id.hmmy_choice);
        choice2 = (RadioButton) view.findViewById(R.id.cied_choice);
        choice3 = (RadioButton) view.findViewById(R.id.ele_choice);
        newsyes = (RadioButton) view.findViewById(R.id.yes_news_choice);
        newsno = (RadioButton) view.findViewById(R.id.no_news_choice);

        choice1.setOnClickListener(this);
        choice2.setOnClickListener(this);
        choice3.setOnClickListener(this);
        newsyes.setOnClickListener(this);
        newsno.setOnClickListener(this);

        DataBaseHelper dbHelper = new DataBaseHelper(getContext());
        String notif = dbHelper.getSettings("NEWS_NOTIFICATION");
        String studies = dbHelper.getSettings("STUDIES");

        switch (studies) {
            case "hmmy_choice":
                choice1.setChecked(true);
                break;
            case "cied_choice":
                choice2.setChecked(true);
                break;
            case "ele_choice":
                choice3.setChecked(true);
                break;
        }
        switch (notif) {
            case "yes":
                newsyes.setChecked(true);

```

```

        break;
    case "no":
        newsno.setChecked(true);
        break;
    }
}

public void onClick(View view) {

    // Is the button now checked?
    boolean checked = ((RadioButton) view).isChecked();
    Intent restart = new Intent (getActivity(), WelcomeActivity.class);
    DataBaseHelper dbHelper = new DataBaseHelper(getContext());
    // Check which radio button was clicked

    switch(view.getId()) {
        case R.id.hmmy_choice:
            if (checked)
                dbHelper.changeSetting("STUDIES", "hmmy_choice");
            startActivity(restart);
            break;
        case R.id.cied_choice:
            if (checked)
                dbHelper.changeSetting("STUDIES", "cied_choice");
            startActivity(restart);
            break;
        case R.id.ele_choice:
            if (checked)
                dbHelper.changeSetting("STUDIES", "ele_choice");
            startActivity(restart);
            break;
        case R.id.yes_news_choice:
            if (checked)
                dbHelper.changeSetting("NEWS_NOTIFICATION", "yes");
            break;
        case R.id.no_news_choice:
            if (checked)
                dbHelper.changeSetting("NEWS_NOTIFICATION", "no");
            break;
    }
}
}

```

#### Callback.java

```

package net.dimitrisiliou.myhmmmy.sidemenu;

public interface Callback {

    void onSideMenuItemClick(int i);

}

```

#### MenuAdapter.java

```

package net.dimitrisiliou.myhmmmy.sidemenu;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;

```

```

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import net.dimitrisiliou.myhmmmy.R;

import java.util.List;

public class MenuAdapter extends
RecyclerView.Adapter<MenuAdapter.MenuViewHolder> {

    List<MenuItem> mData ;
    Callback listener;

    public MenuAdapter(List<MenuItem> mData, Callback listener) {
        this.mData = mData;
        this.listener = listener;
    }

    @NonNull
    @Override
    public MenuViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_menu,parent,fa
lse);
        return new MenuViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull MenuViewHolder holder, int
position) {

        holder.icon.setImageResource(mData.get(position).getIcon());
        if (mData.get(position).isSelected()) {
            holder.isSelected.setVisibility(View.VISIBLE);
        }
        else
            holder.isSelected.setVisibility(View.GONE);

    }

    @Override
    public int getItemCount() {
        return mData.size();
    }

    public class MenuViewHolder extends RecyclerView.ViewHolder {

        ImageView icon,isSelected;

        public MenuViewHolder(@NonNull View itemView) {
            super(itemView);

            icon = itemView.findViewById(R.id.item_menu_icon);
            isSelected = itemView.findViewById(R.id.item_menu_selected);

            // menu item click listener
            itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {

```

```

        listener.onSideMenuItemClick(getAdapterPosition());
    });
}
}
}

```

### MenuItem.java

```

package net.dimitrisiliou.myhmmmy.sidemenu;

public class MenuItem {

    private int icon,code;
    private boolean isSelected;

    public MenuItem(int icon, int code) {
        this.icon = icon;
        this.code = code;
    }

    public MenuItem(int icon, int code, boolean isSelected) {
        this.icon = icon;
        this.code = code;
        this.isSelected = isSelected;
    }

    public int getIcon() {
        return icon;
    }

    public void setIcon(int icon) {
        this.icon = icon;
    }

    public int getCode() {
        return code;
    }

    public void setCode(int code) {
        this.code = code;
    }

    public boolean isSelected() {
        return isSelected;
    }

    public void setSelected(boolean selected) {
        isSelected = selected;
    }
}

```

### MenuUtil.java

```

package net.dimitrisiliou.myhmmmy.sidemenu;

import net.dimitrisiliou.myhmmmy.R;
import net.dimitrisiliou.myhmmmy.database.DataBaseHelper;

import java.util.ArrayList;

```

```

import java.util.List;

public class MenuUtil {

    public static final int RSS_FRAGMENT_CODE = 0;
    public static final int BOOK_FRAGMENT_CODE = 1;
    public static final int COMMUNITY_FRAGMENT_CODE = 2;
    public static final int PROFESSORS_FRAGMENT_CODE = 3;
    public static final int MAP_FRAGMENT_CODE = 4;
    public static final int CALENDAR_FRAGMENT_CODE = 5;
    public static final int ABOUT_FRAGMENT_CODE = 6;
    public static final int SETTINGS_FRAGMENT_CODE = 7;

    public static List<MenuItem> getMenuList() {
        List<MenuItem> list = new ArrayList<>();
        // first menu item is selected
        list.add(new MenuItem(R.drawable.ic_rss, RSS_FRAGMENT_CODE,true));
        list.add(new
MenuItem(R.drawable.ic_professors,PROFESSORS_FRAGMENT_CODE,false));
        list.add(new MenuItem(R.drawable.ic_community,
COMMUNITY_FRAGMENT_CODE,false));
        list.add(new MenuItem(R.drawable.ic_book, BOOK_FRAGMENT_CODE,false));
        list.add(new MenuItem(R.drawable.ic_map, MAP_FRAGMENT_CODE,false));
        list.add(new MenuItem(R.drawable.ic_calendar,
CALENDAR_FRAGMENT_CODE,false));
        list.add(new MenuItem(R.drawable.ic_info,
ABOUT_FRAGMENT_CODE,false));
        list.add(new MenuItem(R.drawable.ic_settings,
SETTINGS_FRAGMENT_CODE,false));

        return list;
    }
}

```

### AboutFragment.java

```

package net.dimitrisiliou.myhmmmy;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class AboutFragment extends Fragment implements View.OnClickListener {

    ImageView linkedin,website,myemail;

    public AboutFragment() {
        // Required empty public constructor
    }

    @Override

```

```

public View onCreateView(LayoutInflater inflater, ViewGroup container,
                        Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment_about, container, false);
}

public void onViewCreated(@NonNull View view, @Nullable Bundle
savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);
    linkedin = view.findViewById(R.id.mylinkedin);
    website = view.findViewById(R.id.mywebsite);
    myemail = view.findViewById(R.id.myemail);
    linkedin.setOnClickListener(this);
    website.setOnClickListener(this);
    myemail.setOnClickListener(this);
}

@Override
public void onClick(View v) {
    if (linkedin.equals(v)) {
        Uri uri =
Uri.parse("https://www.linkedin.com/in/dimitrisiliou/");
        Intent intent = new Intent(Intent.ACTION_VIEW, uri);
        startActivity(intent);
    } else if (website.equals(v)) {
        Uri uri = Uri.parse("https://www.dimitrisiliou.net");
        Intent intent = new Intent(Intent.ACTION_VIEW, uri);
        startActivity(intent);
    } else if (myemail.equals(v)) {
        Uri uri = Uri.parse("mailto:dimitris.iliou@outlook.com");
        Intent intent = new Intent(Intent.ACTION_VIEW, uri);
        startActivity(intent);
    }
}
}
}

```

### CalendarFragment.java

```

package net.dimitrisiliou.myhmmmy;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.fragment.app.Fragment;

public class CalendarFragment extends Fragment implements
View.OnClickListener {
    TextView tv_curriculumButton;

    public CalendarFragment() {
        // Required empty public constructor
    }
}

```

```

    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_calendar, container,
false);
    }

    public void onViewCreated(@NonNull View view, @Nullable Bundle
savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        tv_curriculumButton= view.findViewById(R.id.curriculumButton);
        tv_curriculumButton.setOnClickListener(this);
    }

    @Override
    public void onClick(View view) {
        Uri uri = Uri.parse("https://www.ece.uop.gr/announcement/orologia-
programmata-earinou-eksaminou-2/");
        Intent intent = new Intent(Intent.ACTION_VIEW,uri);
        startActivity(intent);
        Log.d("CREATION", String.valueOf(uri));
    }
}

```

### MainActivity.java

```

package net.dimitrisiliou.myhmmy;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.annotation.SuppressLint;
import android.app.job.JobInfo;
import android.app.job.JobScheduler;
import android.content.ComponentName;
import android.os.Bundle;
import android.util.Log;

import net.dimitrisiliou.myhmmy.BackroundUpdate.AppJobService;
import net.dimitrisiliou.myhmmy.community.communityFragment;
import net.dimitrisiliou.myhmmy.courses.CourseMatchingFragment;
import net.dimitrisiliou.myhmmy.database.DataBaseHelper;
import net.dimitrisiliou.myhmmy.map.MapFragment;
import net.dimitrisiliou.myhmmy.news.NewsFragment;
import net.dimitrisiliou.myhmmy.settings.SettingsFragment;
import net.dimitrisiliou.myhmmy.sidemenu.Callback;
import net.dimitrisiliou.myhmmy.sidemenu.MenuAdapter;
import net.dimitrisiliou.myhmmy.sidemenu.MenuItem;
import net.dimitrisiliou.myhmmy.sidemenu.MenuUtil;
import net.dimitrisiliou.myhmmy.professors.ProfessorsFragment;

import java.util.List;

public class MainActivity extends AppCompatActivity implements Callback {
    private static final String TAG = "MainActivity";
}

```

```

RecyclerView menuRv;
List<MenuItem> menuItems;
MenuAdapter adapter;
int selectedMenuPos = 0 ;
JobScheduler jobScheduler;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    jobScheduler = (JobScheduler) getSystemService(JOB_SCHEDULER_SERVICE);
    DataBaseHelper dataBaseHelper = new DataBaseHelper(this);
    String notifications =
dataBaseHelper.getSettings("NEWS_NOTIFICATION");

    if (notifications.equals("yes")) {
        scheduleJob();
    } else if (notifications.equals("no")) {
        cancelJob();
    }
    // setup side menu
    setupSideMenu();
    // set the default fragment when activity launch
    setNewsFragment();
}

private void setupSideMenu() {
    DataBaseHelper dbHelper = new DataBaseHelper(this);
    String old_studies = dbHelper.getSettings("STUDIES") ;
    menuRv = findViewById(R.id.rv_side_menu);

    // get menu list item will get data from a static data class

    menuItems = MenuUtil.getMenuList();
    if(old_studies.equals("hmmmy_choice")) {
        int remove_courses_tab = 3;
        menuItems.remove(remove_courses_tab); //index = 3 is the course
(tab) item from MenuUtil
    }
    adapter = new MenuAdapter(menuItems, this);
    menuRv.setLayoutManager(new LinearLayoutManager(this));
    menuRv.setAdapter(adapter);
}

void setCommunityfragment() {

getSupportFragmentManager().beginTransaction().replace(R.id.container,new
communityFragment()).commit();

}

void setProfessorsFragment() {

getSupportFragmentManager().beginTransaction().replace(R.id.container,new
ProfessorsFragment()).commit();

}

```



```

void setNewsFragment() {

getSupportFragmentManager().beginTransaction().replace(R.id.container,new
NewsFragment()).commit();

}
void setCoursesFragment() {

getSupportFragmentManager().beginTransaction().replace(R.id.container, new
CourseMatchingFragment()).commit();

}
void setMapfragment() {

getSupportFragmentManager().beginTransaction().replace(R.id.container,new
MapFragment()).commit();

}
void setCalendarfragment() {

getSupportFragmentManager().beginTransaction().replace(R.id.container,new
CalendarFragment()).commit();

}
void setAboutfragment() {

getSupportFragmentManager().beginTransaction().replace(R.id.container,new
AboutFragment()).commit();

}
void setSettingsfragment() {

getSupportFragmentManager().beginTransaction().replace(R.id.container,new
SettingsFragment()).commit();

}

@Override
public void onSideMenuItemClick(int i) {

switch (menuItems.get(i).getCode()) {

case MenuUtil.RSS_FRAGMENT_CODE: setNewsFragment();
break;
case MenuUtil.PROFESSORS_FRAGMENT_CODE: setProfessorsFragment();
break;
case MenuUtil.BOOK_FRAGMENT_CODE: setCoursesFragment();
break;
case MenuUtil.COMMUNITY_FRAGMENT_CODE: setCommunityfragment();
break;
case MenuUtil.MAP_FRAGMENT_CODE: setMapfragment();
break;
case MenuUtil.CALENDAR_FRAGMENT_CODE: setCalendarfragment();
}
}

```

```

        break;
        case MenuUtil.ABOUT_FRAGMENT_CODE: setAboutfragment();
        break;
        case MenuUtil.SETTINGS_FRAGMENT_CODE: setSettingsfragment();
        break;
        default: setNewsFragment();
    } // highlight the selected menu item

    menuItems.get(selectedMenuPos).setSelected(false);
    menuItems.get(i).setSelected(true);
    selectedMenuPos = i;
    adapter.notifyDataSetChanged();

}

public void scheduleJob() {
    @SuppressWarnings("JobSchedulerService") ComponentName componentName =
new ComponentName(this, AppJobService.class);
    JobInfo info = new JobInfo.Builder(123, componentName)
        .setPeriodic(6*1000)
        .setRequiresCharging(false)
        .setPersisted(true)
        .setRequiresBatteryNotLow(true)
        .build();
    JobScheduler scheduler = (JobScheduler)
this.getSystemService(JOB_SCHEDULER_SERVICE);
    int resultCode = scheduler.schedule(info);
    if (resultCode == JobScheduler.RESULT_SUCCESS) {
        Log.d(TAG, "Job scheduled");
    } else {
        Log.d(TAG, "Job scheduling failed");
    }
}

public void cancelJob() {
    JobScheduler scheduler = (JobScheduler)
this.getSystemService(JOB_SCHEDULER_SERVICE);
    scheduler.cancel(123);
    Log.d(TAG, "Job cancelled");
}
}

```

### WelcomeActivity.java

```

package net.dimitrisiliou.myhmmmy;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Handler;
import android.text.BoringLayout;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;

import net.dimitrisiliou.myhmmmy.database.DataBaseHelper;
import net.dimitrisiliou.myhmmmy.news.NewsModel;

import java.io.IOException;

```

```

public class WelcomeActivity extends AppCompatActivity {
    private static int SPLASH_TIME_OUT = 1500;
    TextView welcome,hiden,feedback,feedbacktxt;
    RadioGroup radioGroup;
    Button button;
    SharedPreferences pref;
    String choice;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_welcome);
        DataBaseHelper dataBaseHelper = new
        DataBaseHelper(getApplicationContext());

        try {
            dataBaseHelper.createDatabase();
        } catch (IOException e) {
            e.printStackTrace();
        }
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                //set fragment for first use
                Boolean isFirstRun = getSharedPreferences("mystudies",
                MODE_PRIVATE)
                    .getBoolean("isFirstRun", true);
                if (isFirstRun) {
                    FirstScreen();
                    DataBaseHelper dataBaseHelper = new
                    DataBaseHelper(getApplicationContext());
                    dataBaseHelper.changeSetting("STUDIES", "hmmmy_choice");
                } else {
                    // set the default fragment when activity launch
                    Intent homeIntent = new
                    Intent(WelcomeActivity.this,MainActivity.class);
                    startActivity(homeIntent);
                    finish();
                }
                getSharedPreferences("mystudies", MODE_PRIVATE).edit()
                    .putBoolean("isFirstRun", false).apply();
            }
        },SPLASH_TIME_OUT);
    }

    @SuppressWarnings("NonConstantResourceId")
    public void onRadioButtonClicked(View view) {
        // Is the button now checked?
        DataBaseHelper dataBaseHelper = new
        DataBaseHelper(getApplicationContext());
        RadioButton button = (RadioButton) view;
        boolean checked = button.isChecked();
        Intent homeIntent = new
        Intent(WelcomeActivity.this,MainActivity.class);
        // Check which radio button was clicked
        switch(view.getId()) {
            case R.id.hmmmy_choice:
                if (checked)
                    dataBaseHelper.changeSetting("STUDIES", "hmmmy_choice");
                    startActivity(homeIntent);
                finish();
                break;

```

```
        case R.id.cied_choice:
            if (checked)
                DataBaseHelper.changeSetting("STUDIES", "cied_choice");
                startActivity(homeIntent);
                finish();
            break;
        case R.id.ele_choice:
            if (checked)
                DataBaseHelper.changeSetting("STUDIES", "ele_choice");
                startActivity(homeIntent);
                finish();
            break;
    }
}

public void FirstScreen(){
    setContentView(R.layout.fragment_settings);

    welcome = (TextView) findViewById(R.id.welcome_settings);
    hidden = (TextView) findViewById(R.id.textView15);
    feedback = (TextView) findViewById(R.id.textView16);
    feedbacktxt = (TextView) findViewById(R.id.textView7);
    radioGroup = (RadioGroup) findViewById(R.id.radiobutton2);
    hidden.setVisibility(View.INVISIBLE);
    radioGroup.setVisibility(View.INVISIBLE);
    feedback.setVisibility(View.INVISIBLE);
    feedbacktxt.setVisibility(View.INVISIBLE);
    welcome.setText("Διάλεξε ένα από τα παρακάτω\n για να συνεχίσεις");
}
}
```

## ΒΙΒΛΙΟΓΡΑΦΙΑ

1. **Πανεπιστήμιο Πελοποννήσου.** *Διδακτικό Ερευνητικό Προσωπικό του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών.* [Ηλεκτρονικό] [Παραπομπή: 15 Μαΐου 2021.] [https://www.ece.uop.gr/our\\_team\\_category/didaktiko-erevnitiko-dep/](https://www.ece.uop.gr/our_team_category/didaktiko-erevnitiko-dep/).
  
2. **Facebook Groups.** *Ομάδα Facebook του Σπουδαστικού Συλλόγου Τμήματος Μηχανικών Πληροφορικής ΑΤΕΙ Δυτικής Ελλάδας.* [Ηλεκτρονικό] [Παραπομπή: 15 Μαΐου 2021.] <https://www.facebook.com/groups/tesyd>.
  
3. **Facebook Pages.** *Επίσημη σελίδα Facebook του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών.* [Ηλεκτρονικό] [Παραπομπή: 15 Μαΐου 2021.] <https://www.facebook.com/%CE%A4%CE%BC%CE%AE%CE%BC%CE%B1-%CE%97%CE%BB%CE%B5%CE%BA%CF%84%CF%81%CE%BF%CE%BB%CF%8C%CE%B3%CF%89%CE%BD-%CE%9C%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CE%BA%CF%8E%CE%BD-%CE%9C%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CE%BA%CF%8E%CE%BD-%CE%A5%CF%80%CE%BF%CE%BB%CE%BF%CE%B3%CE%B9%CF%83%CF%84%CF%8E%CE%BD-%CE%A0%CE%B1%CE%BD-%CE%A0%CE%B5%CE%BB%CE%BF%CF%80%CE%BF%CE%BD%CE%BD%CE%AE%CF%83%CE%BF%CF%85-5260939274729504>
  
4. **Facebook Groups.** *Ομάδα Facebook του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών.* [Ηλεκτρονικό] [Παραπομπή: 15 Μαΐου 2021.] <https://www.facebook.com/groups/elec.teipat>.
  
5. **WeCare.** *Δομή Συμβουλευτικής και Ψυχολογικής Υποστήριξης – WeCare του Πανεπιστημίου Πελοποννήσου.* [Ηλεκτρονικό] [Παραπομπή: 15 Μαΐου 2021.] <https://wecare.uop.gr/>.
  
6. **Τ.Ε.Ι Δυτικής Ελλάδας.** *Πρόγραμμα Σπουδών Μηχανικών Πληροφορικής Τ.Ε.* [Ηλεκτρονικό] [Παραπομπή: 15 Μαΐου 2021.] [http://www.cied.teiwest.gr/images/cied/studies/curriculum\\_20\\_06\\_2017.pdf](http://www.cied.teiwest.gr/images/cied/studies/curriculum_20_06_2017.pdf).

7. **Τ.Ε.Ι Δυτικής Ελλάδας**. *Πρόγραμμα Σπουδών Ηλεκτρολόγων Μηχανικών Τ.Ε.* [Ηλεκτρονικό] [Παραπομπή: 15 Μαΐου 2021.]  
<http://www.ele.teiwest.gr/index.php/education/undergraduate/courses>.
8. **Πανεπιστήμιο Πελοποννήσου**. *Οδηγός Αντιστοιχίσεων Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Πληροφορικής.* [Ηλεκτρονικό] [Παραπομπή: 15 Μαΐου 2021.]  
<https://www.ece.uop.gr/pinakes-antistoichiseon/>.
9. **Πανεπιστήμιο Πελοποννήσου**. *Χώροι Εκπαίδευσης Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Πληροφορικής.* [Ηλεκτρονικό] [Παραπομπή: 15 Μαΐου 2021.]  
<https://www.ece.uop.gr/choroi-ekpaidefsis/>.
10. **Πανεπιστήμιο Πελοποννήσου**. *Ακαδημαϊκό Ημερολόγιο του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών.* [Ηλεκτρονικό] [Παραπομπή: 15 Μαΐου 2021.]  
<https://www.ece.uop.gr/akadimaiko-imerologio/>.
11. **Android Developers**. *Create dynamic lists with RecyclerView. The official site for Android app developers.* [Ηλεκτρονικό] [Παραπομπή: 15 Μάιος 2021.]  
<https://developer.android.com/guide/topics/ui/layout/recyclerview>.
12. **Android Developers**. *Documentation for JobService.* [Ηλεκτρονικό] [Παραπομπή: 15 Μαΐου 2021.] <https://developer.android.com/reference/android/app/job/JobService>.
13. **Android Developers**. *The official site for Android app developers.* [Ηλεκτρονικό] [Παραπομπή: 15 Μάη 2021.] <https://developer.android.com/reference/java/lang/Thread>.
14. **Android Developers**. *Documentation for Activiy.* [Ηλεκτρονικό]  
<https://developer.android.com/reference/android/app/Activity>.
15. **Android Developers**. *Documentation for Fragments.* [Ηλεκτρονικό] [Παραπομπή: 15 Μάιος 2021.] <https://developer.android.com/guide/fragments>.
16. **Android Developers**. *Documemtation for Adapter.* [Ηλεκτρονικό] [Παραπομπή: 15 Μαΐου 2021.] <https://developer.android.com/reference/android/widget/Adapter>.

17. **Android Developers.** *Documentation for AsyncTask.* [Ηλεκτρονικό] [Παραπομπή: 15 Μαΐου 2021.] <https://developer.android.com/reference/android/os/AsyncTask>.