

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη Android εφαρμογής για την υποστήριξη και οργάνωση των φοιτητών του τμήματος
Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων (ΔΙ.ΠΑ.Ε.)



Του φοιτητή
Καρανικόλα Γεώργιου
Αρ. Μητρώου: 175058

Επιβλέπων
Κεραμόπουλος Ευκλείδης
Βαθμίδα: Αναπληρωτής Καθηγητής

Ημερομηνία 29/09/2021

Τίτλος Δ.Ε. Ανάπτυξη Android εφαρμογής για την υποστήριξη και οργάνωση των φοιτητών του τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων (ΔΙ.ΠΑ.Ε.)

Κωδικός Δ.Ε. 21162

Όνοματεπώνυμο φοιτητή/των Καρανικόλας Γεώργιος

Όνοματεπώνυμο εισηγητή Κεραμόπουλος Ευκλείδης

Ημερομηνία ανάληψης Δ.Ε. 02-03-2021

Ημερομηνία περάτωσης Δ.Ε. 29/09/2021

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Καρανικόλα Γεώργιου που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιοδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητα και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Αφιέρωση»

Πρόλογος

Η παρούσα πτυχιακή επιλέχθηκε διότι, οι εφαρμογές κινητών κυριαρχούν στην καθημερινότητα της ζωής μας. Το θέμα που επιλέχθηκε ήταν για να γνωρίσω καλύτερα την ανάπτυξη Android εφαρμογών και την γλώσσα προγραμματισμού Kotlin. Επίσης, με την δημιουργία του έργου, μου πρόσφερε πολλές επιπλέον γνώσεις.

Περίληψη

Το θέμα της πτυχιακής είναι να οργανώσει και να βοηθήσει τους φοιτητές από την στιγμή που θα μούνε στο τμήμα έως και την στιγμή που θα τελειώσουν. Η εφαρμογή παρέχει εύκολη και γρήγορη πρόσβαση σε ότι μπορεί να χρειαστούν καθόλη την διάρκεια της φοιτητικής περιόδου. Η εφαρμογή σχεδιάστηκε με όσο το δυνατό πιο ποιοτικό τρόπο ώστε να είναι συντηρήσιμη και επεκτάσιμη μιας και μπορεί να μπαίνουν διάφορες λειτουργίες για την καλύτερη και ευχάριστη εμπειρία του φοιτητή. Για αυτό το σκοπό χρησιμοποιήθηκαν διάφορες αρχιτεκτονικές και κανόνες καθώς και μια πληθώρα τεχνολογιών και βιβλιοθηκών.

« Development of an Android application for the support and organization of the students of the Department of Information and Electronic Engineering (I.H.U.) »

«Karanikolas Georgios»

Abstract

The purpose of this thesis is to organize and assist the students from the moment they start their curriculum until the time of its completion. This application provides an easy and quick access in everything that they might need during the duration of their lives as students. It was designed to be easily maintainable and expandable, since it can include different functions for a better and more pleasant experience for the student. For this, it uses a variety of architectures and rules, as well as an abundance of technologies and libraries.

Ευχαριστίες

Ως το ελάχιστο που μπορώ να κάνω, με την παρούσα παράγραφο οφείλω να ευχαριστήσω όλους όσους συνέλαβαν στην εκπόνηση της πτυχιακής.

Τον επιβλέποντα καθηγητή μου, κος Κεραμόπουλο Ευκλείδη, για την πολύτιμη υποστήριξη του, με τις παραγωγικές υποδείξεις του και το ευχάριστο κλίμα συνεργασίας που είχαμε.

Επίσης στους φοιτητές του τμήματος που με βοήθησαν με την συμπλήρωση του ερωτηματολογίου και με τις χρήσιμες προτάσεις τους για την εφαρμογή.

Τέλος τους δικούς μου ανθρώπους, για την όλη υποστήριξη όλο αυτό το καιρό που μου παρείχαν ότι χρειαζόμουν ώστε να μείνω καθηλωμένος στον στόχο μου.

Περιεχόμενα

Πρόλογος.....	iv
Περίληψη.....	v
Abstract	vi
Ευχαριστίες	vii
Περιεχόμενα	viii
Κατάλογος Σχημάτων	xi
Κατάλογος Πινάκων.....	xii
Συντομογραφίες.....	xiii
Κεφάλαιο 1ο: Εισαγωγή.....	1
Κεφάλαιο 2ο: Android, Android Studio και Kotlin.....	3
2.1 Εισαγωγή.....	3
2.2 Android.....	3
2.2.1 Εισαγωγή.....	3
2.2.2 Η εξέλιξη του Android	3
2.2.3 Η αρχιτεκτονική των Android εφαρμογών.....	4
2.2.4 Γιατί να επιλέξω το Android	6
2.3 Android Studio	7
2.3.1 Εισαγωγή.....	7
2.3.2 Γιατί να επιλέξω Android Studio	8
2.4 Kotlin.....	8
2.4.1 Εισαγωγή.....	8
2.4.2 Η ιστορία της Kotlin.....	9
2.4.3 Η εξέλιξη της Kotlin.....	9
2.4.4 Γιατί να επιλέξω την Kotlin	11
2.5 Επίλογος.....	12
Κεφάλαιο 3ο: Ανάπτυξη της εφαρμογής Student IEE-IHU	13
3.1 Εισαγωγή.....	13
3.2 Διαδικασία ανάπτυξης.....	13
3.2.1 Οργάνωση.....	13
3.2.2 Εργαλεία.....	13

3.3	Firebase	17
3.3.1	Εισαγωγή	17
3.3.2	Authentication	17
3.3.3	Realtime Database	18
3.3.4	Storage	18
3.4	Απομακρυσμένη βάση δεδομένων SQL	19
3.4.1	Δομή της βάσης	19
3.4.2	Δημιουργία και εισαγωγές της βάσης	20
3.4.3	Περιγραφή του PHP API για την διαχείριση της βάσης	21
3.5	Οι βιβλιοθήκες που χρησιμοποιήθηκαν	23
3.5.1	Glide	23
3.5.2	Glide Transformations	23
3.5.3	MotionToast	24
3.5.4	Navigation Components	24
3.5.5	Firebase UI	24
3.5.6	Google Maps	24
3.5.7	EasyPermissions-ktx	25
3.5.8	MaterialStyledDialogs	25
3.5.9	Material Dialog Input	25
3.5.10	MMKV	25
3.5.11	Coroutines	26
3.5.12	Room Components	26
3.5.13	Lifecycle Components	27
3.5.14	Espresso Test	27
3.5.15	Retrofit2	27
3.5.16	ExpandableTextView	28
3.5.17	ViewPager2	28
3.5.18	RecyclerView	28
3.5.19	CardView	28
3.6	Το ελληνοαγγλικό σύστημα	29
3.7	Η δομή της εφαρμογής	30
3.7.1	Το αρχείο AndroidManifest.xml	30
3.7.2	Η δομή των αρχείων Kotlin	30
3.7.3	Η δομή των αρχείων Res	35
3.7.4	Η διαμόρφωση του Gradle	38

3.8	Λειτουργίες.....	40
3.8.1	Μενού πλοήγησης.....	40
3.8.2	Αυθεντικοποίηση.....	41
3.8.3	Προφίλ (Profile).....	44
3.8.4	Ρυθμίσεις (Settings).....	45
3.8.5	Κοινότητες (Communities).....	47
3.8.6	Βρείτε (Find).....	48
3.8.7	Κατάλογος Επικοινωνίας (Contact).....	51
3.8.8	Τμήμα (Department).....	52
3.8.9	Οδηγοί (Guides).....	53
3.8.10	Σελίδα Βοήθειας (Help page).....	55
3.8.11	Κεντρική Σελίδα (Home).....	56
3.8.12	Έλεγχος Δικτύου.....	56
3.8.13	IEE Apps.....	57
3.9	Δοκιμές (Testing).....	58
3.10	Οδηγίες χρήσης.....	58
3.11	Επίλογος.....	59
Κεφάλαιο 4ο:	Συμπεράσματα ή/και προτάσεις βελτίωσης.....	61
	BIBΛΙΟΓΡΑΦΙΑ.....	62
	ΠΑΡΑΡΤΗΜΑ Α : Ερωτηματολόγιο.....	65

Κατάλογος Σχημάτων

Σχήμα 2.1: Τα επίπεδα της αρχιτεκτονικής του Android. [2][5-7]	5
Σχήμα 3.1: Πίνακας περιεχομένων στο Notion.....	14
Σχήμα 3.2: Η μέθοδος Kanban στο Notion.....	14
Σχήμα 3.3: Το αποθετήριο GitHub του έργου.....	15
Σχήμα 3.4: Το εργαλείο GitKraken του έργου.....	15
Σχήμα 3.5: Τα πρόσθετα του Android Studio.....	16
Σχήμα 3.6: Το εργαλείο Postman.....	17
Σχήμα 3.7: Η δομή της Realtime Database στην Firebase.....	18
Σχήμα 3.8: Η δομή του Storage στην Firebase.....	19
Σχήμα 3.9: Η δομή της απομακρυσμένης βάσης δεδομένων SQL.....	19
Σχήμα 3.10: Το αρχείο για την δημιουργία της απομακρυσμένης βάσης δεδομένων SQL.....	20
Σχήμα 3.11: Το αρχείο για τις εισαγωγές της απομακρυσμένης βάσης δεδομένων SQL	21
Σχήμα 3.12: Το αρχείο PHP API για την άντληση πληροφοριών από την απομακρυσμένη βάση δεδομένων SQL.....	22
Σχήμα 3.13: Η συνέχεια από το αρχείο PHP API για την άντληση πληροφοριών από την απομακρυσμένη βάση δεδομένων SQL.....	23
Σχήμα 3.14: Το Translations Editor για την επεξεργασία των γλωσσών της εφαρμογής.....	29
Σχήμα 3.15: Τα permissions από το αρχείο AndroidManifest.xml.....	30
Σχήμα 3.16: Τα meta-data από το αρχείο AndroidManifest.xml	30
Σχήμα 3.17: Η αρχιτεκτονική MVVM.....	31
Σχήμα 3.18: Η δομή των φακέλων / αρχείων από το Data.....	33
Σχήμα 3.19: Η δομή των φακέλων / αρχείων από το UI.....	34
Σχήμα 3.20: Η δομή των αρχείων από το Utils.....	35
Σχήμα 3.21: Η δομή των φακέλων / αρχείων από το Res.....	37
Σχήμα 3.22: Η ρυθμίσεις στο Gradle για τα layouts.....	38
Σχήμα 3.23: Το αρχείο Gradle του project.....	39
Σχήμα 3.24: Το αρχείο Gradle του module.....	39
Σχήμα 3.25: Η συνέχεια από το αρχείο Gradle του module.....	40
Σχήμα 3.26: Το μενού πλοήγησης στην εφαρμογή.....	41
Σχήμα 3.27: Η σελίδα σύνδεσης.....	43
Σχήμα 3.28: Η σελίδα εγγραφής.....	44
Σχήμα 3.29: Η σελίδα του προφίλ.....	45
Σχήμα 3.30: Η σελίδα των ρυθμίσεων.....	46
Σχήμα 3.31: Η σελίδα των κοινοτήτων.....	48
Σχήμα 3.32: Η σελίδα βρες.....	51
Σχήμα 3.33: Η σελίδα του καταλόγου.....	52
Σχήμα 3.34: Η σελίδα του τμήματος.....	53
Σχήμα 3.35: Η σελίδα των οδηγών.....	54
Σχήμα 3.36: Η σελίδα βοήθειας.....	55
Σχήμα 3.37: Η κεντρική σελίδα.....	56
Σχήμα 3.38: Η σελίδα εκτός δικτύου.....	57
Σχήμα 3.39: Μια δοκιμή (test) για την είσοδο στην εφαρμογή.....	58

Κατάλογος Πινάκων

Πίνακας 1: Εξέλιξη των εκδόσεων του Android[3]	4
Πίνακας 2: Οι βασικές εκδόσεις της Kotlin.[17].....	9
Πίνακας 3: Οι σημαντικές διαφορές της Kotlin με την Java[19][20][21].....	11

Συντομογραφίες

IHU	International Hellenic University
IEE	Information and Electronic Engineering
GIF	Graphics Interchange Format
API	Application Programming Interface
URL	Uniform Resource Locator
UI	User Interface
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
DAO	Data Access Objects
IDE	Integrated Development Environment

Κεφάλαιο 1ο: Εισαγωγή

Στόχος της παρούσας πτυχιακής είναι η ανάπτυξη Android εφαρμογής η οποία θα βοηθάει τους φοιτητές του τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων (ΙΕΕ-ΙΗΥ) από την στιγμή που θα μπουνε στο τμήμα έως και την στιγμή που θα το τελειώσουν.

Αρχικά με την παροχή οργανωμένων πληροφοριών και εύρεση διάφορων πληροφοριών όπως μαθήματα, καθηγητές, τηλέφωνα, υπηρεσίες, προσφορές κλπ. που πιθανόν να χρειαστούν. Έτσι ώστε να τους προσφέρει όσο το δυνατόν ποιο ποιοτικό τρόπο ζωής σχετικά με το τμήμα από άποψη βελτίωσης ταχύτητας, εύρεσης ότι πληροφοριών χρειάζονται στο τμήμα. Τέλος, οι πληροφορίες που θα έχει η εφαρμογή θα παρουσιάζονται με μια ομορφότερη εμφάνιση και θα έχουν μια μεγαλύτερη εγκυρότητα, μιας και θα έχουν ελεγχθεί σε κάποιο βαθμό ώστε να μπουν στην εφαρμογή.

Για το σκοπό αυτό χρησιμοποιήθηκαν, το λογισμικό ανάπτυξης Android ως λογισμικό για την λειτουργία της εφαρμογής, το Android Studio ως IDE για την ανάπτυξη της εφαρμογής και η γλώσσα προγραμματισμού Kotlin ως προεπιλεγμένη γλώσσα προγραμματισμού για την ανάπτυξη της εφαρμογής.

Επιπλέον χρησιμοποιήθηκε η PHP για γλώσσα προγραμματισμού για την ανάπτυξη των APIs. Τέλος, χρησιμοποιήθηκαν πολλές επιπρόσθετες βιβλιοθήκες και οργανωτικά προγράμματα ώστε να διευκολύνουν την διαδικασία ανάπτυξης της εφαρμογής.

Στο δεύτερο κεφάλαιο αναλύονται σε τρία υποκεφάλαια, τα εξής: το λειτουργικό σύστημα Android, το Android Studio και η γλώσσα προγραμματισμού Kotlin.

Στην πρώτο υποκεφάλαιο, αναλύεται το λειτουργικό σύστημα Android. Η εξέλιξη του καθώς και η αρχιτεκτονική των Android εφαρμογών. Τέλος για ποιο λόγο να επιλέξει κάποιος το Android σαν λειτουργικό για να δημιουργήσει το έργο του.

Στο δεύτερο υποκεφάλαιο αναλύεται το Android Studio. Τα δομικά στοιχεία του καθώς και γιατί να επιλέξει κάποιος το Android Studio για την ανάπτυξη μιας Android εφαρμογής.

Στο τρίτο υποκεφάλαιο αναλύεται η γλώσσα προγραμματισμού Kotlin. Η ιστορία της καθώς και η εξέλιξη της. Τέλος, για ποιο λόγο να επιλέξει κάποιος την Kotlin για γλώσσα προγραμματισμού στην δημιουργία του έργου του.

Στο τρίτο κεφάλαιο παρουσιάζετε η εφαρμογή Student ΙΕΕ-ΙΗΥ. Είναι το μεγαλύτερο και το σημαντικότερο κεφάλαιο της πτυχιακής, διότι μέσα από αυτό παρουσιάζετε η εφαρμογή και η διαδικασία ανάπτυξης της. Αρχίζοντας αναλύεται η οργάνωση της εφαρμογής καθώς και τα βοηθητικά εργαλεία που χρησιμοποιήθηκαν για την δημιουργία της. Έπειτα αναλύονται οι λειτουργίες της Firebase όπου χρησιμοποιήθηκαν. Στη συνέχεια γίνεται ανάπτυξη στην απομακρυσμένη βάση δεδομένων SQL, στη δομή της βάσης, τα δύο αρχεία για την δημιουργία - συμπλήρωση της βάσης και η περιγραφή του PHP API για την διαχείριση της εφαρμογής. Ακολούθως αναφέρονται οι επιπρόσθετες βιβλιοθήκες όπου χρησιμοποιήθηκαν στην εφαρμογή αναλυτικά και το ελληνοαγγλικό σύστημα της εφαρμογής. Συνεχίζοντας, υπάρχει η δομή της εφαρμογής, το αρχείο AndroidManifest.xml, η δομή των αρχείων Kotlin, των αρχείων Res και το Gradle. Έπειτα αναλύονται όλες οι λειτουργίες της εφαρμογής όπως αυθεντικοποίηση,

κοιότητες κλπ., οι δοκιμές (Testing) της εφαρμογής και ακολουθούν οι οδηγίες χρήσης για την εφαρμογή.

Τέλος, στο τέταρτο κεφάλαιο θα παρουσιαστούν τα γενικά συμπεράσματα τα οποία προέκυψαν από την ανάπτυξη της εφαρμογής καθώς και μελλοντικές προτάσεις για την εφαρμογή.

Κεφάλαιο 2ο: Android, Android Studio και Kotlin

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα αναλυθεί, το λειτουργικό σύστημα, το περιβάλλον ανάπτυξης (IDE) και η γλώσσα προγραμματισμού όπου χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής.

Στο πρώτο υποκεφάλαιο, θα αναπτυχθεί το λειτουργικό σύστημα όπου επιλέχθηκε για την υλοποίηση της εφαρμογής, το Android. Στο δεύτερο υποκεφάλαιο, θα αναπτυχθεί το περιβάλλον ανάπτυξης (IDE) όπου επιλέχθηκε για την υλοποίηση της εφαρμογής, το Android Studio. Τέλος, στο τρίτο υποκεφάλαιο, θα αναπτυχθεί η γλώσσα προγραμματισμού που επιλέχθηκε για την υλοποίηση της εφαρμογής, η Kotlin.

2.2 Android

2.2.1 Εισαγωγή

Το Android είναι ένα λειτουργικό σύστημα (Operating System) που αναπτύχθηκε από την Google με βάση τον πυρήνα Linux. Το λειτουργικό σύστημα Android δημιουργήθηκε κυρίως για συσκευές κινητών τηλεφώνων (smartphone), παρόλα αυτά υποστηρίζει πολλές συσκευές όπως έξυπνα ρολόγια (Android Wear), τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto), ταμπλέτες (Tablets) κλπ.[1].

Το Android είναι το πιο δημοφιλές λειτουργικό σύστημα για κινητά. Με κύριο ανταγωνιστή το iOS για συσκευές της Apple. Μια έρευνα που πραγματοποιήθηκε το έτος 2017, όπου διαπιστώθηκε ότι το 64.8% των προγραμματιστών για κινητά χρησιμοποιούν το Android ως προτιμώμενη πλατφόρμα για δημιουργία εφαρμογών[1].

Η Google κυκλοφορεί τον πηγαίο κώδικα Android υπό τις άδειες ανοιχτού κώδικα. Επειδή είναι ανοιχτού κώδικα το έργο, πολλοί δημιούργησαν δικές τους τροποποιημένες εκδόσεις του Android μέσω του Android Open Source Project (ASOP)[1].

2.2.2 Η εξέλιξη του Android

Η ιστορία του Android ξεκινάει από τον Οκτώβριο του 2003, πριν τα smartphones γίνουν ευρέως γνωστά ή η Apple να ανακοινώσει το πρώτο iPhone. Η Android Inc ιδρύθηκε στο Palo Alto της Καλιφόρνια από τους τέσσερις ιδρυτές της, τον Rich Miner, Nick Sears, Chris White και Andy Rubin[2].

Το 2005 η Google εξαγόρασε την Android Inc. Ο Rubin και άλλα ιδρυτικά μέλη συνέχισαν παρόλα αυτά την ανάπτυξη του λειτουργικού συστήματος υπό τους νέους ιδιοκτήτες της εταιρίας. Τότε αποφασίστηκε ότι θα χρησιμοποιηθεί ως βάση το Linux και έτσι θα προσφέρετε δωρεάν σε τρίτους κατασκευαστές κινητών τηλεφώνων[2].

Στις 5 Νοεμβρίου του 2007 κυκλοφόρησε η πρώτη δημόσια beta έκδοση του Android 1.0 για προγραμματιστές και στις 23 Σεπτεμβρίου του 2008 κυκλοφόρησε επίσημα η έκδοση του Android 1.0[2].

Το λειτουργικό σύστημα Android από την πρώτη κυκλοφορία έως και σήμερα έχει αλλάξει πάρα πολύ οπτικά και λειτουργικά αρκετές φορές. Έχει μια ραγδαία εξέλιξη με τον χρόνο. Στο Πίνακα 1, φαίνονται αναλυτικά οι εκδόσεις, καθώς το επίπεδο API και η ημερομηνία κυκλοφορίας τους.

Πίνακας 1: Εξέλιξη των εκδόσεων του Android[3]

Όνομα έκδοσης	Αριθμός έκδοσης	Ημερομηνία κυκλοφορίας	Επίπεδο API
Snow Cone	12.0	2021 Q3	31
Red Velvet Cake	11.0	19/02/2020	30
Queen Cake	10.0	03/09/2019	29
Pie	9.0	06/08/2018	28
Oreo	8.0-8.1	21/08/2017	26-27
Nougat	7.0 – 7.1.2	22/08/2016	24-25
Marshmallow	6.0-6.0.1	05/10/2015	23
Lollipop	5.0-5.1.1	12/11/2014	21-22
KitKat	4.4-4.4.4	31/10/2013	19-20
Jelly Bean	4.1-4.3.1	09/07/2012	16-18
Ice Cream Sandwich	4.0-4.0.4	18/10/2011	14-15
Honeycomb	3.0-3.2.6	22/02/2011	11-13
Gingerbread	2.3-2.3.7	06/12/2010	9-10
Froyo	2.2-2.2.3	20/05/2010	8
Éclair	2.0-2.1	26/10/2009	5-7
Donut	1.6	15/08/2009	4
Cupcake	1.5	27/04/2009	3
Banana bread	1.1	09/02/2009	2
Apple pie	1.0	23/09/2008	1

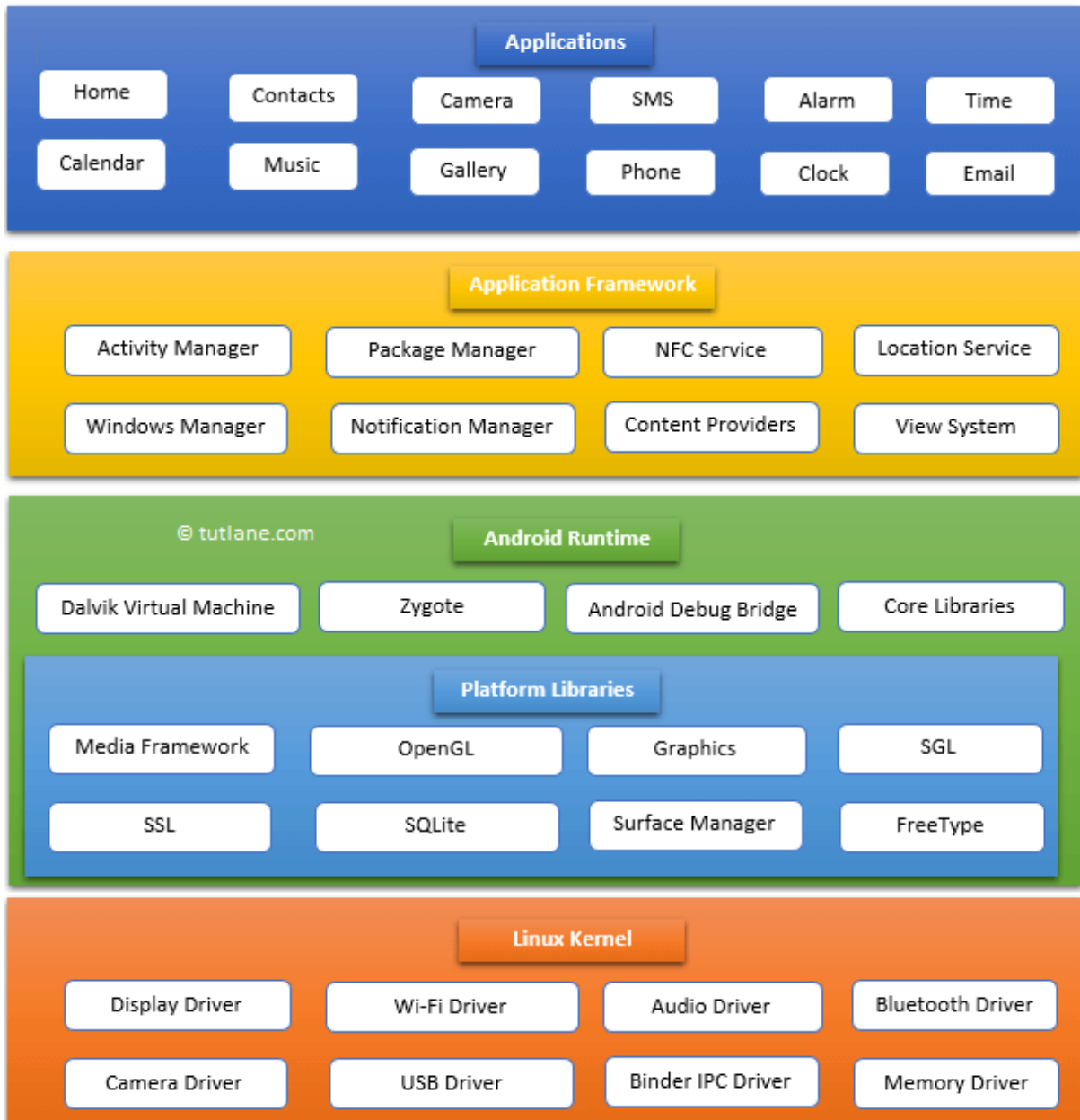
2.2.3 Η αρχιτεκτονική των Android εφαρμογών

Η αρχιτεκτονική του Android είναι μια στοίβα δομημένων λογισμικών στοιχείων που υποστηρίζουν ανάγκες κινητής συσκευής όπως φαίνεται και στο Σχήμα 2.1.

Αυτή η στοίβα περιέχει:[4-6]

- Τον πυρήνα Linux.
- Τις βιβλιοθήκες πλατφόρμας.
- Το Android Runtime.
- Το Android Framework.
- Τις εφαρμογές.

Σε αυτά τα στοιχεία, το κύριο συστατικό είναι ο πυρήνας του Linux για να παρέχει τις λειτουργίες του λειτουργικού συστήματος σε κινητά και το Dalvik Virtual Machine (DVM) το οποίο είναι υπεύθυνο για την εκτέλεση μιας εφαρμογής στα κινητά.



Σχήμα 2.1: Τα επίπεδα της αρχιτεκτονικής του Android. [2][5-7]

2.2.3.1 Ο πυρήνας Linux

Ο πυρήνας Linux είναι το κατώτερο επίπεδο και το κύριο στοιχείο της αρχιτεκτονικής του Android. Διαχειρίζεται όλες τις βασικές λειτουργίες του συστήματος, όπως διαχείριση διαδικασιών, οθόνης, μνήμης, ηλεκτρολογίου, Bluetooth κλπ.[4-6].

2.2.3.2 Οι βιβλιοθήκες πλατφόρμας

Οι βιβλιοθήκες πλατφόρμας περιλαμβάνουν διάφορες βιβλιοθήκες C / C++ και βιβλιοθήκες που βασίζονται σε Java, όπως SSL, Graphics, SQLite, Media κλπ. για την παροχή υποστήριξης της ανάπτυξης του Android[4-6].

2.2.3.3 Το Android Runtime

Το Android Runtime περιβάλλον είναι ένα σημαντικό και κύριο μέρος του Android. Περιέχει στοιχεία όπως βασικές βιβλιοθήκες και την εικονική μηχανή Dalvik. Το Android Runtime είναι αυτό που τροφοδοτεί τις εφαρμογές, μαζί με τις βιβλιοθήκες και αποτελεί βάση για το πλαίσιο εφαρμογής.

Η εικονική μηχανή Dalvik (DVM) είναι μια εικονική μηχανή που βασίζεται σε μητρώο όπως η εικονική μηχανή Java (JVM). Είναι ειδικά σχεδιασμένη για Android, για να διασφαλίσει με αυτό τον τρόπο την εκτέλεση πολλών περιπτώσεων αποτελεσματικά και γρήγορα και τέλος βασίζεται στον πυρήνα Linux για νήματα και διαχείριση μνήμης χαμηλού επιπέδου.

Οι βασικές βιβλιοθήκες στο χρόνο εκτέλεσης του Android επιτρέπουν να εφαρμόσουμε εφαρμογές Android χρησιμοποιώντας την τυπική γλώσσα προγραμματισμού JAVA[4-6].

2.2.3.4 Το Application Framework

Το Application Framework παρέχει τις κλάσεις που χρησιμοποιούνται για την δημιουργία Android εφαρμογών. Επίσης παρέχει μια γενική αφαίρεση για πρόσβαση υλικού και διαχειρίζεται τη διεπαφή χρήστη και τους πόρους της εφαρμογής. Το Application Framework περιλαμβάνει υπηρεσίες όπως υπηρεσίες τοποθεσίας, ειδοποιήσεων, NFC κλπ. [4-6].

2.2.3.5 Οι εφαρμογές

Το ανώτερο επίπεδο της αρχιτεκτονικής του Android είναι οι εφαρμογές. Οι εγγενείς εφαρμογές και οι εφαρμογές τρίτων όπως επαφές, email, ρολόι κλπ. Ότι και να κατασκευάσουμε, θα εγκατασταθούν μόνο σε αυτό το επίπεδο.

Στο επίπεδο εφαρμογής εκτελείται εντός του Android Runtime χρησιμοποιώντας τις τάξεις και τις υπηρεσίες που διατίθενται από το Application Framework[4-6].

2.2.4 Γιατί να επιλέξω το Android

Το λειτουργικό σύστημα Android είναι τόσο δημοφιλές και διαθέσιμο που πλέον έχει γίνει ένα κομμάτι της ζωής των ανθρώπων, μιας και υπάρχουν Android παντού, smartphones, ρολόγια, τηλεοράσεις, αυτοκίνητα κλπ..

Τα βασικά πλεονεκτήματα που συναντούμε στο λειτουργικό σύστημα Android είναι:[7]

- Ανοιχτού κώδικα
- Εύκολη ανάπτυξη εφαρμογών
- Ποικιλία επιλογής συσκευών
- Μεγάλη παραμετροποίηση

2.2.4.1 Ανοιχτού κώδικα

Το Android είναι ένα λειτουργικό ανοιχτού κώδικα, αυτό είναι ένα από τα μεγαλύτερα του θετικά, είναι και ο λόγος που αναφέρετε και πρώτο. Το χαρακτηριστικό αυτό, δίνει την δυνατότητα σε

οποιοδήποτε προγραμματιστή ή ομάδα προγραμματιστών να έχει πρόσβαση στον πηγαίο κώδικα και να κάνει οποιαδήποτε αλλαγή θέλει ώστε να φτιάξει κάποια δικιά του τροποποιημένη έκδοση για προσωπική ή εμπορική χρήση. Κάτω από κάποιους όρους ή να διορθώσει διάφορα πιθανά σφάλματα ή να λάβει προτάσεις και σχόλια για να βελτιώσει η Google[7].

2.2.4.2 Εύκολη ανάπτυξη εφαρμογών

Η ανάπτυξη εφαρμογών στο Android είναι εξαιρετικά εύκολη και προσιτή σε αντίθεση με της Apple. Υπάρχει μια πολύ μεγάλη ποικιλία οδηγών από το “πως να φτιάξεις μια εφαρμογή από το μηδέν”, μέχρι στο “πως να την ανεβάσεις εύκολα στο Play Store”.

Ένα ακόμα θετικό στην ανάπτυξη εφαρμογών για Android είναι ότι δεν χρειάζεται κάποιο ακριβό εξοπλισμό, μιας και υπάρχουν φθηνά Android για αγορά. Όπως και πολλές εικονικές συσκευές Android για χρήση και ανάπτυξη εφαρμογών.

Συνοπτικά με όλα αυτά τα πλεονεκτήματα που παρέχει το Android όπου αναφέρθηκαν παραπάνω, είναι πολύ εύκολο για έναν νέο στην ανάπτυξη εφαρμογών αλλά και για έναν πιο έμπειρο, στο να δημιουργήσει εφαρμογές για Android. Χωρίς κάποιο ιδιαίτερο κόστος, με μεγάλη καθοδήγηση από το πλήρη οδηγό με τεκμηριώσεις (documentations) και όλους τους επιπλέον οδηγούς που υπάρχουν[7].

2.2.4.3 Ποικιλία επιλογής συσκευών

Το Android σε αντίθεση με τα iPhones που τα έχει μόνο η Apple, τα έχουν πάρα πολλοί κατασκευαστές με αποτέλεσμα να μπορεί κάποιος να διαλέξει την αγαπημένη του συσκευή, ανάμεσα σε μια τεράστια ποικιλία κινητών όπου θα είναι στα μέτρα του και συνήθως σε πολύ καλύτερες τιμές μιας και δεν υπάρχει μονοπώλιο[7].

2.2.4.4 Μεγάλη παραμετροποίηση

Τα Android έχουν μεγάλη παραμετροποίηση. Μπορεί κάποιος να φέρει το Android στα μέτρα του. Από το να διαλέξει εξαρχής την εταιρία και τα συμπληρωματικά λογισμικά που έχει η εκάστοτε εταιρία, το να βάλει κάποια ελεύθερη Rom έως μέσω διάφορων εφαρμογών και ρυθμίσεων. Γενικά το Android δίνει μεγάλη ελευθερία σε αυτό το κομμάτι σε αντίθεση με τα iPhones της Apple[7].

2.3 Android Studio

2.3.1 Εισαγωγή

Το Android Studio είναι το επίσημο IDE για την Google, για ανάπτυξη Android εφαρμογών με βάση το IntelliJ IDEA της JetBrains. Είναι διαθέσιμο για Windows, macOS και Linux λειτουργικά συστήματα. Γενικά θεωρείται ένα βαρύ IDE για αυτό έχει και κάποιες απαιτήσεις από το υλικό του υπολογιστή. Παρόλα αυτά στις νεότερες εκδόσεις έχει βελτιωθεί αρκετά αυτό το θέμα[8-10][13].

Το Android Studio ανακοινώθηκε στις 15 Μαΐου 2013 στο Google I/O συνέδριο όπου ήταν σε πρώιμο στάδιο έκδοσης 0.1[9]. Τον Ιούνιο του 2014 κυκλοφόρησε από την έκδοση 0.8 σαν beta και

τον Δεκέμβριο του 2014 βγήκε η πρώτη σταθερή έκδοση 1.0[10]. Μέχρι την έκδοση του Μαΐου 2021 όπου ήταν η 4.2 είχε ένα αναφορικό όνομα εκδόσεως από την μέρα όπου βγήκε. Τον Ιούλιο 2021 άλλαξε και η νέα έκδοση είναι η Arctic Fox (2020.3.1)[11].

Στις 7 Μαΐου 2019, η Kotlin αντικατέστησε την Java ως προτιμώμενη γλώσσα της Google για την ανάπτυξη Android εφαρμογών, η Java παρόλα αυτά εξακολουθεί να υποστηρίζεται[12].

2.3.2 Γιατί να επιλέξω Android Studio

Το Android Studio διαθέτει πολλές συναρπαστικές δυνατότητες που μπορούν να βελτιώσουν την παραγωγικότητα και γενικότερα την ανάπτυξη Android εφαρμογών όπως:[8][13][14]

- Υποστήριξη κατασκευής που βασίζεται στο Gradle.
- Δυνατότητες ProGuard ενσωμάτωσης και υπογραφής εφαρμογών.
- Γρήγορο και πλούσιο σε χαρακτηριστικά σύστημα εξομοιωτών (emulators).
- Σκοτεινό θέμα για καλύτερη ξεκούραση των ματιών.
- Διάφορους οδηγούς βασισμένους σε πρότυπα για την δημιουργία κοινών σχεδίων και εξαρτημάτων στο Android.
- Πάρα πολλά πρόσθετα για επιπλέον δυνατότητες του Android Studio.
- Προσομοιωτή για προβολή και βασική επεξεργασία της τοπικής βάσης.
- Ολοκληρωμένο σύστημα ανάγνωσης σφαλμάτων και μηνύματα ελέγχου.
- Ολοκληρωμένο σύστημα ελέγχου και διαχείρισης πολυμέσων αρχείων.
- Σύστημα αναπαράστασης σε πραγματικό χρόνο διεπαφών.
- Σύστημα drag and drop για την διεπαφή.
- Ενοποιημένο περιβάλλον για ανάπτυξη εφαρμογών για όλες τις συσκευές Android.
- Ενσωματωμένη υποστήριξη για το Google Cloud Platform.
- Ενσωματωμένη υποστήριξη για την Firebase.
- Ενσωματωμένη υποστήριξη για το GitHub
- Διάφορα εργαλεία ελέγχου και διαχείρισης της απόδοσης, χρηστικότητας και συμβατότητας έκδοσης και άλλων προβλημάτων.

Τα παραπάνω χαρακτηριστικά είναι πολλά από αυτά που έχει και υποστηρίζει το Android Studio. Υπάρχει μια ραγδαία ανάπτυξη και υποστήριξη του Android Studio συνεχώς μέχρι και σήμερα.

Ένας από τους ανταγωνιστές το πρώτο διάστημα ήταν το Eclipse. Πολύ γρηγορά ήρθε σε δεύτερη μοίρα. Το Android Studio όπως έχει αναφερθεί έχει πολλές δυνατότητες που δεν υποστηρίζει το Eclipse όπως το drag and drop σύστημα, την σταθερότητα του συστήματος, το πολύ δυνατό Gradle, πολλές ενσωματώσεις που έχει το Android Studio και πολλά ακόμα[13][14]. Αυτός ο συνδυασμός ευκολίας και προσαρμοστικότητας μαζί με όλες αυτές τις δυνατότητες το κάνουν ένα ολοκληρωμένο και δυνατό IDE για ανάπτυξη Android εφαρμογών.

2.4 Kotlin

2.4.1 Εισαγωγή

Η Kotlin είναι μια μοντέρνα γλώσσα προγραμματισμού που αναπτύχθηκε από την εταιρία JetBrains. Είναι συνοπτική, ασφαλής και δια λειτουργική με την Java και άλλες γλώσσες.

Χρησιμοποιείται ευρέως για δημιουργία εφαρμογών από την πλευρά του διακομιστή, για υπολογιστές και για κινητά. Η Google την υποστηρίζει επίσημα για την πλατφόρμα Android με αποτέλεσμα οι εφαρμογές Android να μπορούν να γραφτούν σε Kotlin και μάλιστα σαν προτεινόμενη γλώσσα. Η Kotlin διαθέτει υποστήριξη πολλαπλών πλατφορμών. Αυτό σημαίνει ότι ο κώδικας της μπορεί να επαναχρησιμοποιηθεί σε Android, iOS ή Web και αυτό είναι πολύ θετικό [15][16].

2.4.2 Η ιστορία της Kotlin

Η ιστορία της Kotlin αρχίζει τον Ιούλιο του 2011, η JetBrains αποκάλυψε το Project Kotlin, μια νέα γλώσσα για την JVM όπου την δούλευαν ήδη ένα χρόνο. Ο επικεφαλής του project ήταν ο Dmitry Jemerov. Το όνομα προέρχεται από το νησί Kotlin όπου είναι κοντά στην Αγία Πετρούπολη[15][16].

Ο λόγος δημιουργίας της ήταν επειδή οι περισσότερες γλώσσες δεν είχαν τα χαρακτηριστικά που ήθελαν με μονή εξαίρεση την Scala, αλλά και αυτή είχε αργό χρόνο σύνταξης με αποτέλεσμα να είναι και αυτή ανεπαρκής. Βασικός στόχος της Kotlin ήταν να είναι στην σύνταξη τόσο γρήγορη όσο η Java[15] [16].

Τον Φεβρουάριο του 2012, η JetBrains άνοιξε την βάση του έργου υπό την άδεια Apache 2. Στις 15 Φεβρουάριου του 2016 κυκλοφόρησε η πρώτη σταθερή έκδοση 1.0 και η JetBrains δεσμεύτηκε για μακροπρόθεσμη συμβατότητα προς τα πίσω ξεκινώντας από αυτή την έκδοση[15][16]. Το 2017 στο Google I/O η Google ανακοίνωσε την υποστήριξη της Kotlin για την δημιουργία εφαρμογών στο Android [15] [16]. Τέλος, στις 7 Μαΐου του 2019 ανακοίνωσε την Kotlin σαν προεπιλεγμένη γλώσσα για ανάπτυξη εφαρμογών στο Android[12].

2.4.3 Η εξέλιξη της Kotlin

Η Kotlin από την μέρα που κυκλοφόρησε έως και σήμερα έχει μια ραγδαία ανάπτυξη και συνεχών αναβαθμίσεων της με νέες δυνατότητες και βελτίωση των επιδόσεων. Στο Πίνακα 2 φαίνονται οι βασικές εκδόσεις της Kotlin και οι ημερομηνίες κυκλοφορίας του.

Πίνακας 2: Οι βασικές εκδόσεις της Kotlin.[17]

Αριθμός έκδοσης	Ημερομηνία κυκλοφορίας
1.1	15/02/2016
1.2	28/11/2017
1.3	29/10/2018
1.4	17/08/2020
1.5	05/05/2021

Στην συνέχεια αναφέρονται σημαντικές αλλαγές / προσθήκες ανά έτος στην Kotlin[17][18].

2011:

- Κυκλοφόρησε η Kotlin.

- Κυκλοφόρησε το πρώτο πρωτότυπο σύνταξης.
- 2012:
- Κυκλοφόρησε το πρώτο Kotlin web demo.
 - Η Kotlin έγινε ανοιχτού κώδικα.
 - Η Kotlin ξεκίνησε να λειτουργεί στο Android.
- 2013:
- Το πρώτο κομμάτι της υποστήριξης μετατροπής SAM εμφανίστηκε στην Kotlin.
 - Προστέθηκαν νέες δυνατότητες της γλώσσας: delegated properties, callable references, static constants και static fields.
 - Κυκλοφόρησε το πρόσθετο Gradle.
- 2014:
- Το Kotlinlang.org ξεκίνησε να λειτουργεί.
 - Προστέθηκαν νέες δυνατότητες για την βελτίωση της δια λειτουργικότητας JavaScript.
- 2015:
- Προστέθηκαν νέες δυνατότητες για την βελτίωση της εμπειρίας του προγραμματιστή σε διάφορες πλατφόρμες: multiple constructors, companion objects, sealed classes και lateinit properties.
- 2016:
- Κυκλοφόρησε η Kotlin 1.0: Γλώσσα για το JVM και το Android.
 - Το Gradle εισάχθηκε στην Kotlin.
 - Το χαρακτηριστικό Coroutines εμφανίστηκε για πρώτη φορά.
- 2017:
- Η Kotlin είναι επίσημο πλέον για το Android σαν γλώσσα.
 - Κυκλοφόρησε η Kotlin 1.1: με υποστήριξη JavaScript, Coroutines και άλλα.
 - Κυκλοφόρησε το πρώτο βιβλίο για την Kotlin.
 - Εμφανίστηκε η πρώτη τεχνολογική προεπισκόπηση της Kotlin/Native.
 - Κυκλοφόρησε η Kotlin 1.2: η οποία βελτίωσε την κοινή χρήση κώδικα μεταξύ πλατφορμών και βελτίωσε τον χρόνο σύνταξης.
- 2018:
- Κυκλοφόρησε το Ktor.
 - Οι Inline classes επέτρεψαν την αναδίπλωση μιας τιμής κάποιου τύπου χωρίς την δημιουργία πραγματικού αντικειμένου περιτύλιξης.
 - Κυκλοφόρησε η Kotlin 1.3: με Coroutines, Kotlin/Native beta και άλλα.
 - Η Kotlin ενσωμάτωσε εκτελέσιμα και επεξεργάσιμα αποσπάσματα Kotlin σε αναρτήσεις ιστολογικού και άλλα επιπλέον πράγματα.
- 2019:
- Η ομάδα έκανε τα πρώτα βήματα στην εξερεύνηση της Kotlin για την επιστήμη των δεδομένων.
 - Στην ανάπτυξη εφαρμογών Android έγινε προεπιλογή η Kotlin.
 - Υλοποιήθηκαν οι μετατροπές SAM για την Kotlin class.
 - Υλοποιήθηκε ο πρώτος διαγωνισμός Kotlin Heroes στο Code forces.
- 2020:
- Το Kotlin Multiplatform μεταφέρθηκε στην έκδοση άλφα.
 - Κυκλοφόρησε η Kotlin 1.4: με έμφαση στην ποιότητα και την απόδοση.
 - Κυκλοφόρησε το Kotlinx.serialization 1.0.
 - Ανακοινώθηκε ένας νέος ρυθμός κυκλοφορίας για την Kotlin και το πρόσθετο Kotlin.

- Κυκλοφόρησε το βιβλίο “Atomic Kotlin”.
- 2021:

- Κυκλοφόρησε η Kotlin 1.5.

2.4.4 Γιατί να επιλέξω την Kotlin

Η Kotlin και ο ανταγωνιστής της η Java, έχουν και οι δύο θετικά και αρνητικά. Παρόλα αυτά, η Kotlin προσφέρει έναν ποιο ωραίο και οργανωμένο κώδικα με αποτέλεσμα να είναι ποιο γρήγορη η παραγωγή κώδικα και ποιο εύκολη η συντήρησή του. Έχει πολλές ενδιαφέρουσες προσθήκες όπως τα coroutines, data class, null safety κλπ.

Παρακάτω στον Πίνακα 3, φαίνονται οι σημαντικές διαφορές μεταξύ Kotlin και Java.

Πίνακας 3: Οι σημαντικές διαφορές της Kotlin με την Java[19][20][21].

Χαρακτηριστικά	Kotlin	Java
Checked Exceptions	Μη διαθέσιμο	Διαθέσιμο
Code Conciseness	Συνοπτικός κώδικας	Μη συνοπτικός κώδικας
Coroutines	Διαθέσιμο	Μη διαθέσιμο
Data Classes	Απαιτείται μόνο η προσθήκη της λέξης - κλειδιού data στον ορισμό κλάσης.	Απαιτείται η σύνταξη πολλή κώδικα boilerplate. (πολλές γραμμές κώδικα με ελάχιστες έως καθόλου αλλαγές.)
Extension Functions	Διαθέσιμο	Μη διαθέσιμο
Higher-Order Functions and Lambdas	Είναι ένα προκαθορισμένο χαρακτηριστικό της.	Εφαρμόζονται χρησιμοποιώντας Callables και οι εκφράσεις Lambdas εισάγονται στην Java 8.
Implicit Widening Conversions	Μη διαθέσιμο	Διαθέσιμο
Inline Functions	Διαθέσιμο	Μη διαθέσιμο
Native Support for Delegation	Διαθέσιμο	Μη διαθέσιμο
Non-private Fields	Μη διαθέσιμο	Διαθέσιμο
NullPointerExceptions	Μη διαθέσιμο	Διαθέσιμο
Primitive Types	Είναι αντικείμενα	Δεν είναι αντικείμενα
Smart Casts	Διαθέσιμο	Μη διαθέσιμο
Static Members	Μη διαθέσιμο	Διαθέσιμο
Support for Constructors	Μπορεί να έχει έναν ή περισσότερους δευτερεύων κατασκευαστές.	Δεν είναι δυνατή η δημιουργία δευτερεύων κατασκευαστών.
Ternary Operator	Μη διαθέσιμο	Διαθέσιμο
Wildcard Types	Έχει εναλλακτική δυνατότητα declaration-	Διαθέσιμο

	site και types ως εναλλακτική λύση.	
--	--	--

Όπως φαίνονται και παραπάνω, οι δύο γλώσσες μεταξύ τους έχουν αρκετά κοινά και διάφορες. Στο θέμα του Android η Kotlin έχει το πάνω χέρι με πολύ ποιο αισθητό ωραίο, σύντομο κώδικα και πολύ ποιο καλή υποστήριξη. Αυτό την κάνει την ιδανική επιλογή!

2.5 Επίλογος

Στο κεφάλαιο αυτό παρουσιάστηκαν, το λειτουργικό σύστημα, το περιβάλλον ανάπτυξης (IDE) και η γλώσσα προγραμματισμού. Ποιο συγκεκριμένα, στο πρώτο υποκεφάλαιο αναφέρθηκε το τι είναι το Android, η εξέλιξη και η αρχιτεκτονική του. Τέλος, γιατί να επιλέξω το Android ως λειτουργικό σύστημα. Στο δεύτερο υποκεφάλαιο αναφέρθηκε το τι είναι το Android Studio και γιατί να επιλέξω το Android Studio για περιβάλλον ανάπτυξης (IDE). Τέλος, στο τρίτο υποκεφάλαιο αναφέρθηκε, στο τι είναι η Kotlin, η ιστορία και η εξέλιξη της και τέλος γιατί να επιλέξω την Kotlin ως γλώσσα προγραμματισμού.

Κεφάλαιο 3ο: Ανάπτυξη της εφαρμογής Student IEE-IHU

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα αναλυθεί πλήρως η ανάπτυξη της εφαρμογής Student IEE-IHU. Ξεκινώντας από τις μεθόδους σχεδίασης και τα βοηθητικά εργαλεία που χρησιμοποιήθηκαν. Στην συνέχεια θα αναλυθούν οι τεχνολογίες και οι βιβλιοθήκες που χρησιμοποιήθηκαν. Έπειτα θα αναλυθεί η δομή όλης της εφαρμογής και τέλος τις λειτουργίες και κάποιες οδηγίες χρήσης για την εφαρμογή.

3.2 Διαδικασία ανάπτυξης

3.2.1 Οργάνωση

Η εφαρμογή δεν άρχισε να σχεδιάζεται τυχαία. Πριν ξεκινήσει η διαδικασία ανάπτυξης έγινε μια έρευνα για μεθόδους σχεδίασης και για το περιεχόμενο της.

Αρχικά φτιάχτηκε μια λίστα με τα features που θα ήταν καλό να έχει η εφαρμογή. Μετά έγινε ένας έλεγχος για τα ποια από αυτά είναι ποιο υλοποιήσιμα στην παρούσα φάση του έργου σύμφωνα με τον χρόνο και τις γνώσεις που υπάρχουν. Έτσι μπήκανε σε μια σειρά με το ποια θα υλοποιηθούν πρώτα και ποια θα φτιαχτούν στο τέλος ή θα γίνουν μελλοντικά features σε μελλοντικές αναβαθμίσεις που θα υπάρξουν μαζί με νέες βελτιώσεις.

Για την διαχείριση των εργασιών του έργου, χρησιμοποιήθηκε η μέθοδος Kanban. Είναι ένας πίνακας με κάρτες έτσι ώστε να οργανωθούν καλύτερα όλες οι εργασίες του έργου σε τι φάση βρίσκονται, πόσο σημαντικές είναι και τι πρέπει να κάνουν. Συνήθως χρησιμοποιούνται οι κατηγορίες ToDo-Doing-Done[22]. Χρησιμοποιείται πολύ συχνά σε Agile μεθόδους ανάπτυξης. Το εργαλείο όπου χρησιμοποιήθηκε για αυτή την δουλειά είναι το Notion, αφού το παρέχει δωρεάν και με αρκετά καλή διαχείριση αυτής της μεθόδου.

Για την διαχείριση και τον συγχρονισμό του έργου χρησιμοποιήθηκε το GitHub και το GitKraken ώστε να μπορεί από διάφορες συσκευές να έχει το ίδιο κώδικα και να ενημερώνετε χωρίς να υπάρχει φόβος να χαθεί η πρόοδος του έργου.

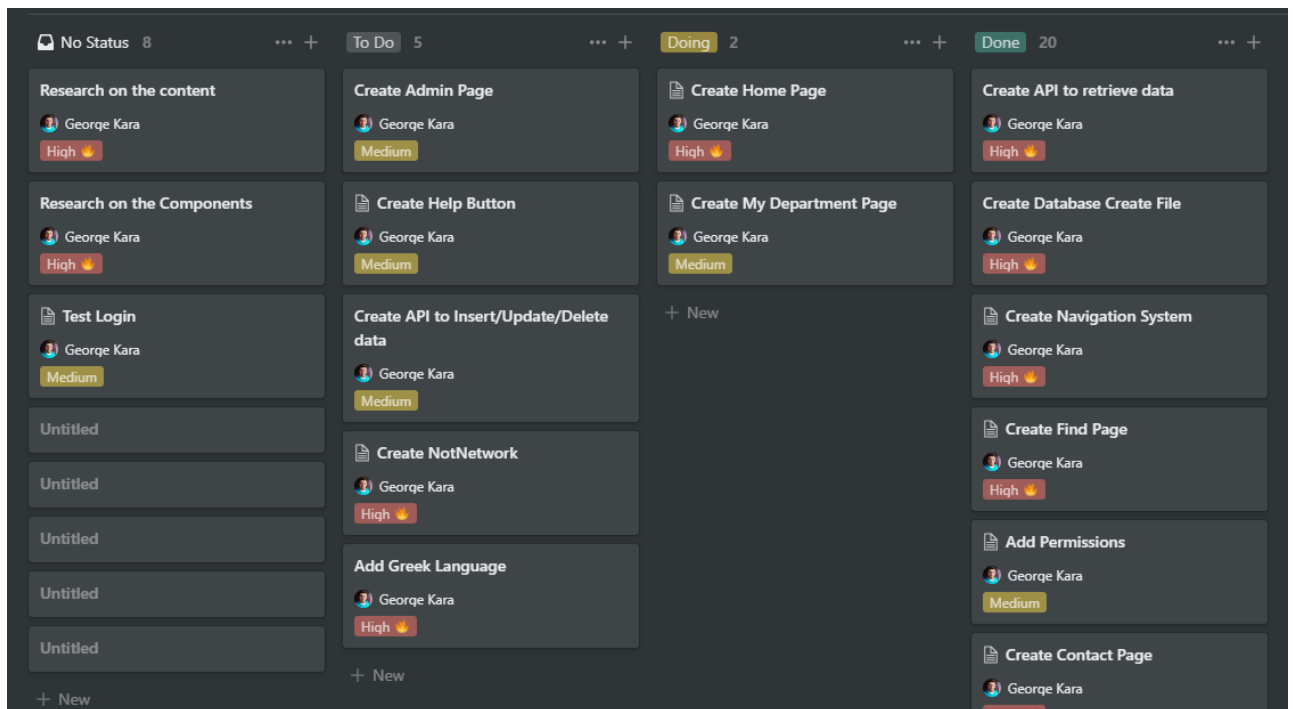
Επίσης όσο αφορά το περιεχόμενο, πραγματοποιήθηκε μέσω Google Forms, ένα ερωτηματολόγιο ώστε να παρθούν απαντήσεις από τους μελλοντικούς χρήστες της εφαρμογής. Έτσι ώστε να διαμορφωθεί το περιεχόμενο όσο το δυνατόν καλύτερα, σύμφωνα και με τις απαντήσεις. Το ερωτηματολόγιο βρίσκεται στο Παράρτημα Α.

3.2.2 Εργαλεία

Το πρώτο εργαλείο που χρησιμοποιήθηκε είναι το Notion για την καταγραφή σημειώσεων και συλλογή πληροφοριών για την εφαρμογή όπως φαίνεται στο Σχήμα 3.1. Επίσης χρησιμοποιήθηκε για την μέθοδο Kanban όπως φαίνεται στο Σχήμα 3.2 όπου αναφέρθηκε στο προηγούμενο υποκεφάλαιο[23].

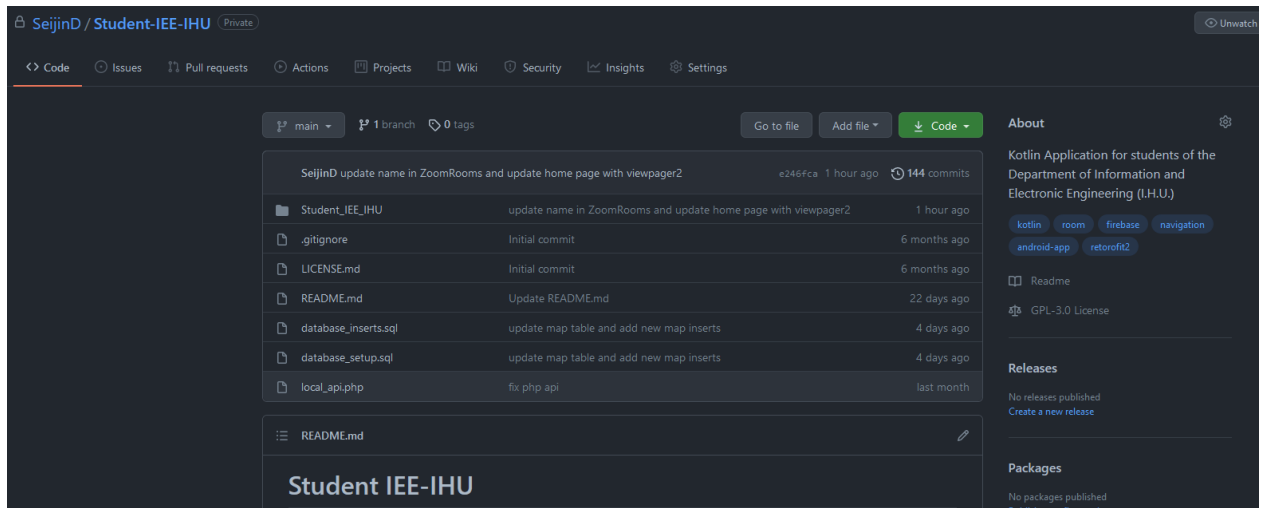
The image shows a dark-themed Notion page titled "Student IEE-IHU". Below the title is a section labeled "Table of Content" which lists several menu items, each with a corresponding icon: "Subject" (globe icon), "To-Do" (heart icon), "Contents" (book icon), "Components" (gear icon), "Research" (clock icon), and "Libraries" (book icon).

Σχήμα 3.1: Πίνακας περιεχομένων στο Notion.

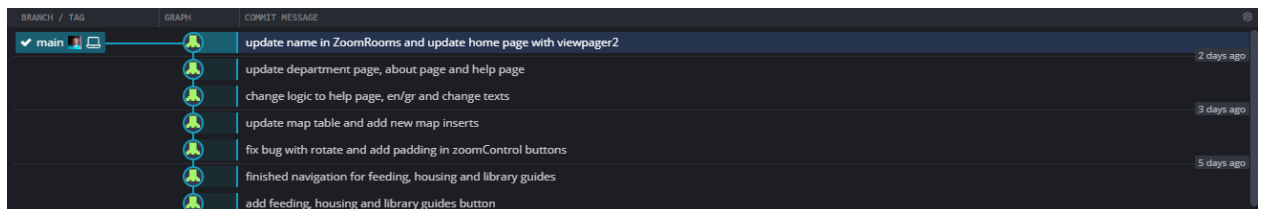


Σχήμα 3.2: Η μέθοδος Kanban στο Notion.

Έπειτα χρησιμοποιήθηκε το GitHub και το GitKraken, για Version Control System. Το GitHub για διαδικτυακό αποθετήριο όπως φαίνεται στο Σχήμα 3.3 και το GitKraken για εύκολη τοπική διαχείριση, για συνεργασία με το GitHub όπως φαίνεται στο Σχήμα 3.4.



Σχήμα 3.3: Το αποθετήριο GitHub του έργου.



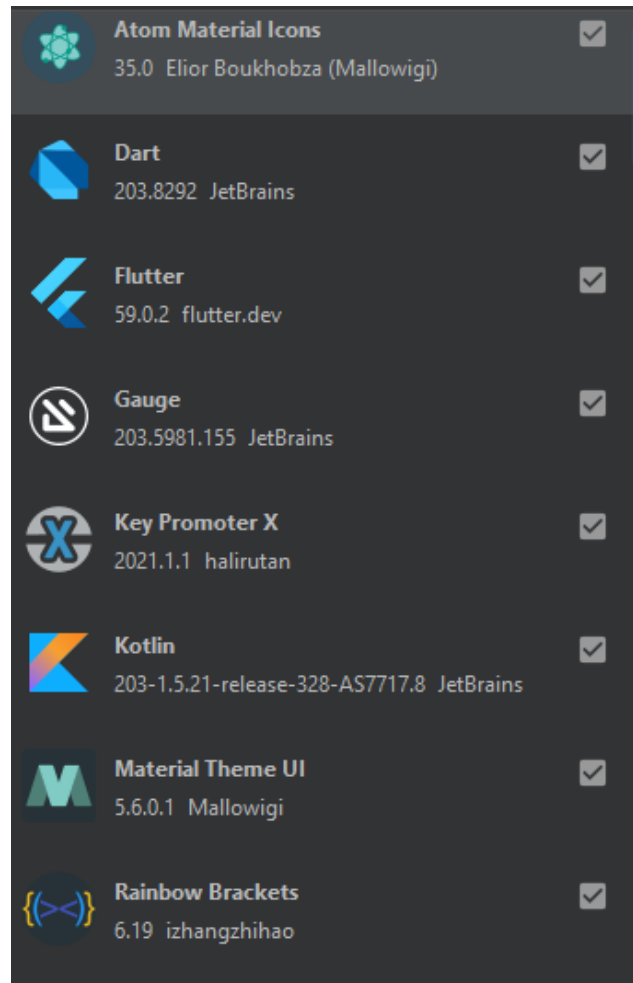
Σχήμα 3.4: Το εργαλείο GitKraken του έργου.

Επόμενο και κύριο εργαλείο το Android Studio. Χρησιμοποιήθηκε για την ανάπτυξη της Android εφαρμογής μαζί με αρκετές εικονικές συσκευές (Emulators), για δοκιμές και ελέγχους σε διαφορετικές εκδόσεις Android όπως και ανάλυση οθονών. Στο Android Studio έγιναν και κάποιες παραμετροποιήσεις, όπως ανέβηκε η ram που χρησιμοποιεί για πιο μεγάλη ταχύτητα. Αλλιώς πηγαίνει πιο αργά λόγω μεγάλου έργου γιατί γεμίζει η μνήμη του. Επίσης προστέθηκαν κάποια προσθετά, για ευκολότερη προσωπική διαδικασία ανάπτυξης της εφαρμογής.

Αυτά τα προσθετά είναι:

- Το Material Theme UI για να έχει καλύτερο χρωματισμό για το μάτι ώστε να μην κουράζονται τα μάτια γρήγορα.
- Το Rainbow Brackets για ευκολότερη ανάγνωση του κώδικα με τις αγκύλες μιας και τις διαφοροποιεί με διαφορετικά χρώματα και είναι δυσκολότερο να μπερδευτούν που ανοίγει και κλείνει κάποια αγκύλη.
- Τέλος το Key Promoter X, ώστε κάθε φορά που κάποια διαδικασία γίνεται με κάποια συντόμευση, αλλά δεν πραγματοποιήθηκε με αυτή, να ενημερώνει ότι δεν χρησιμοποιήθηκε. Να υπενθυμίζει ποια είναι η κάθε συντόμευση, με αυτό το τρόπο γίνεται πολύ γρηγορότερη η διαδικασία ανάπτυξης της εφαρμογής.

Και μπορούμε να τα δούμε πως φαίνονται και στο Σχήμα 3.5.



Σχήμα 3.5: Τα πρόσθετα του Android Studio.

Προτελευταίο εργαλείο είναι το Google Forms. Χρησιμοποιήθηκε για την δημιουργία ερωτηματολογίου ώστε να δοθούν κάποια στατιστικά για προβλήματα και ιδέες για την εφαρμογή από τους μελλοντικούς χρήστες της όπου απευθύνετε.

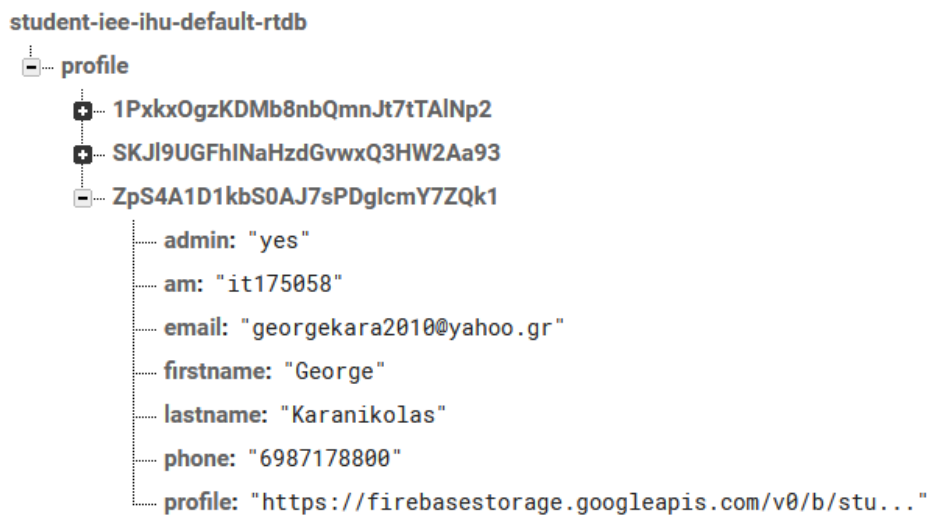
Τελευταίο εργαλείο που χρησιμοποιήθηκε είναι το Postman. Όπως φαίνεται στο Σχήμα 3.6, για ευκολότερη διαχείριση και έλεγχο των αιτημάτων από το API που φτιάχτηκε για την απομακρυσμένη SQL βάση, ώστε να είμαστε σίγουροι ότι δουλεύουν τα αιτήματα και δεν είναι κάποιο πρόβλημα της εφαρμογής.

3.3.3 Realtime Database

Η Realtime Database είναι μια βάση δεδομένων NoSQL που φιλοξενείται στο σύννεφο (cloud). Τα δεδομένα αποθηκεύονται σε μορφή JSON. Τα δεδομένα συγχρονίζονται σε όλους τους πελάτες, σε πραγματικό χρόνο και παραμένουν διαθέσιμα και αν η εφαρμογή βγει εκτός σύνδεσης[26].

Σε αυτή τη μορφή της εφαρμογής χρησιμοποιήθηκε μόνο ένας πίνακας όπως φαίνεται στο Σχήμα 3.7 με το όνομα “profile”. Στην ουσία είναι ένα παιδί για κάθε χρήστη και αποθηκεύει τα παρακάτω στοιχεία: AM, Όνομα, Επίθετο, Τηλέφωνο, Σύνδεσμο για την εικόνα προφίλ που είναι στο Storage και ένα πεδίο για το αν είναι ή όχι διαχειριστής (admin).

Είναι πολύ εύκολο το να προστεθούν ή να αφαιρεθούν πεδία στο πίνακα όπως και στο να δημιουργηθεί κάποιος άλλος νέος πίνακας.

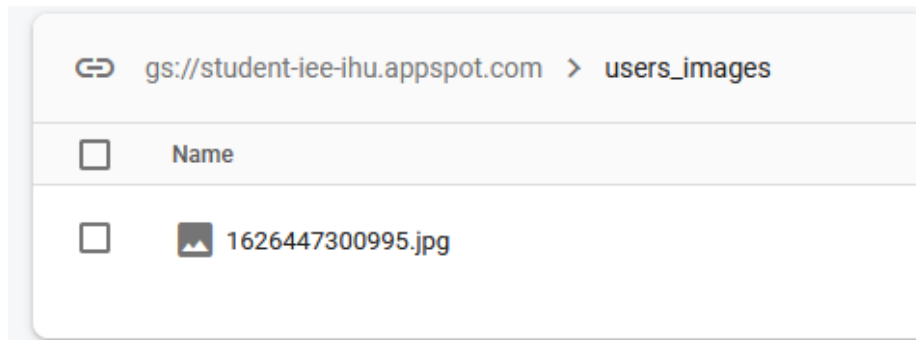


Σχήμα 3.7: Η δομή της Realtime Database στην Firebase.

3.3.4 Storage

Το Storage είναι μια απλή οικονομική υπηρεσία αποθήκευσης αντικείμενων για κάποια εφαρμογή, όπως για φωτογραφίες ή βίντεο ενός χρήστη που φιλοξενείται στο σύννεφο (cloud).[27].

Για την εφαρμογή σε αυτή την μορφή της χρησιμοποιήθηκε μόνο ένας φάκελος ο “users_images” που φαίνεται στο Σχήμα 3.8. Αποθηκεύει τις εικόνες των χρηστών και τα φορτώνει στην εφαρμογή μέσω ενός συνδέσμου που φτιάχνετε αυτόματα από την Firebase.



Σχήμα 3.8: Η δομή του Storage στην Firebase.

3.4 Απομακρυσμένη βάση δεδομένων SQL

3.4.1 Δομή της βάσης

Η απομακρυσμένη βάση φτιάχτηκε για να φορτώνονται οι 8 πίνακες που ανανεώνονται συχνά, ώστε να μην χρειάζεται συνέχεια αναβάθμιση η εφαρμογή.

Εύκολα μπορούν να προστεθούν ή να αφαιρεθούν πίνακες όπως να αλλαχτούν οι υπάρχον και να προσαρμοστούν στο API που φτιάχτηκε έτσι ώστε να είναι αρκετά συντηρήσιμο. Σαφώς θα πρέπει να υπάρχει και αναβάθμιση της εφαρμογής.

Οι πίνακες που υπάρχουν, είναι οι εξής:

- Κοινότητες (Communities)
- Μαθήματα (Courses)
- Καθηγητές (Teachers)
- Επίσημες Υπηρεσίες (Official Services)
- Ανεπίσημες Υπηρεσίες (Unofficial Services)
- Χρήσιμες Ιστοσελίδες (Useful Websites)
- Προσφορές (Offers)
- Χάρτες (Maps)

Και μπορούμε να τα δούμε και στο Σχήμα 3.9.

student_iee_ihu	128.0 KiB
communities	16.0 KiB
courses	16.0 KiB
maps	16.0 KiB
offers	16.0 KiB
official_services	16.0 KiB
teachers	16.0 KiB
unofficial_services	16.0 KiB
useful_websites	16.0 KiB

Σχήμα 3.9: Η δομή της απομακρυσμένης βάσης δεδομένων SQL.

3.4.2 Δημιουργία και εισαγωγές της βάσης

Δημιουργήθηκαν δύο αρχεία για την απομακρυσμένη βάση δεδομένων. Το πρώτο αρχείο είναι για την δημιουργία όλων των πινάκων με τα πεδία και ότι άλλο χρειάζεται όπως φαίνεται στο Σχήμα 3.10. Έπειτα το δεύτερο αρχείο για τις εισαγωγές στους πίνακες όπως φαίνεται στο Σχήμα 3.11, με αυτόν τον τρόπο είναι πολύ εύκολο σε περίπτωση βλάβης ή πιθανότητα μετακίνησης να ξανά χτιστεί η βάση χωρίς να χρειαστεί να γίνει export από την βάση μιας και υπάρχουν πάντα στα δυο αρχεία.

```

1  -- Courses
2
3  CREATE TABLE `courses` (
4      `id` int(11) NOT NULL,
5      `title` VARCHAR(100) COLLATE utf8_unicode_ci DEFAULT NULL,
6      `semester` VARCHAR(20) COLLATE utf8_unicode_ci DEFAULT NULL,
7      `teachers` varchar(200) COLLATE utf8_unicode_ci DEFAULT NULL,
8      `link` varchar(500) COLLATE utf8_unicode_ci DEFAULT NULL
9  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=UTF8_UNICODE_CI;
10
11 ALTER TABLE `courses`
12     ADD PRIMARY KEY (`id`);
13
14 ALTER TABLE `courses`
15     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
16
17 -- Teachers
18
19 CREATE TABLE `teachers` (
20     `id` int(11) NOT NULL,
21     `name` varchar(50) COLLATE utf8_unicode_ci DEFAULT NULL,
22     `email` varchar(50) COLLATE utf8_unicode_ci DEFAULT NULL,
23     `personal_site` varchar(200) COLLATE utf8_unicode_ci DEFAULT NULL,
24     `category` varchar(20) COLLATE utf8_unicode_ci DEFAULT NULL,
25     `link` varchar(500) COLLATE utf8_unicode_ci DEFAULT NULL
26 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
27
28 ALTER TABLE `teachers`
29     ADD PRIMARY KEY (`id`);
30
31 ALTER TABLE `teachers`
32     MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
33

```

Σχήμα 3.10: Το αρχείο για την δημιουργία της απομακρυσμένης βάσης δεδομένων SQL.

```

1  INSERT INTO `courses` (`title`, `semester`, `teachers`, `link`) VALUES
2  ('Μαθηματικά Ι', '1ο', 'Αντωνίου Ευστάθιος, Τζέκης Παναγιώτης', 'https://www
3  ('Δομημένος Προγραμματισμός', '1ο', 'Γουλιάνας Κωνσταντίνος, Ουγιάρου Στ
4  ('Εισαγωγή στην Επιστήμη των Υπολογιστών', '1ο', 'Ηλιούδης Χρήστος', 'htt
5
6  INSERT INTO `teachers` (`name`, `email`, `personal_site`, `category`, `li
7  ('Αδαμίδης Παναγιώτης', 'adamidis@it.teithe.gr', 'https://people.iee.ihu.
8  ('Βίτσας Βασίλης', 'vitsas@it.teithe.gr', 'https://www.it.teithe.gr/~vits
9
10 INSERT INTO `communities` (`title`, `category`, `link`) VALUES
11 ('Τμήμα Μηχανικών Πληροφορικής Και Ηλεκτρονικών Συστημάτων ΔΙΠΑΕ Θεσ/νίκτ
12 ('Ενημέρωση Φοιτητών ΔΙΠΑΕ Σίνδου', 'facebook_pages', 'https://www.facebo
13 ('ΔΙΠΑΕ Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων', 'discord_ser
14 ('ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ Α.Τ.Ε.Ι.ΤΗ.', 'facebook_groups', 'https://www.facebo
15 ('ΔΙ.ΠΑ.Ε. - Τμήμα Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών
16 ('Περιβαλλοντική Ομάδα ΔΙΠΑΕ Θεσσαλονίκης', 'facebook_groups', 'https://w
17 ('ΔΙΠΑΕ(Σίνδος)', 'facebook_groups', 'https://www.facebook.com/groups/154
18

```

Σχήμα 3.11: Το αρχείο για τις εισαγωγές της απομακρυσμένης βάσης δεδομένων SQL .

3.4.3 Περιγραφή του PHP API για την διαχείριση της βάσης

Στα Σχήματα 3.12, 3.13 βλέπουμε το API που φτιάχτηκε με PHP. Από εκεί παίρνουμε τα δεδομένα της απομακρυσμένης βάσης όταν χρειάζεται για ανανέωση των πληροφοριών στην εφαρμογή στον ανάλογο πίνακα κάθε φορά. Φτιάχτηκε έτσι ώστε να κάνει απλά την δουλειά του και να είναι εύκολο να προστεθεί, να αφαιρεθεί ή να αλλαχτεί κάποιος υπάρχων πίνακας.

Αναλυτική περιγραφή του PHP API:

Γραμμές 4-7: Δηλώνονται τα στοιχεία που θα χρειαστεί για την σύνδεση.

Γραμμή 10: Γίνεται το connection της βάσης με τα στοιχεία που δηλώθηκαν και αποθηκεύονται σε ένα αντικείμενο.

Γραμμές 13-16: Πραγματοποιείται έλεγχος αν γίνει κάποιο λάθος ώστε να το εμφανίσει.

Γραμμή 18: Δηλώνονται ότι θα είναι σε μορφή 'utf8'.

Γραμμές 20-22: Παίρνει την παράμετρο "table" αν υπάρχει και την βάζει σε μια μεταβλητή.

Γραμμή 24: Από εδώ και έπειτα υπάρχει ένα switch όπου έχει cases. Κάθε case αντιστοιχίζετε με έναν πίνακα. Η διαδικασία είναι ίδια για κάθε case, με την διαφορά στον πίνακα που αναφέρετε. Στις παρακάτω γραμμές θα γίνει περιγραφή μόνο για το πρώτο case, αφού δεν αλλάζει κάτι ουσιαστικά στα άλλα case.

Γραμμή 27: Δημιουργεί ένα query.

Γραμμή 30: Το εκτελεί.

Γραμμή 33: Παίρνει τα αποτελέσματα και τα βάζει σε μεταβλητές.

Γραμμή 35: Δημιουργεί ένα πίνακα "result".

Γραμμές 36-38: Τρέχει μια while, βάζει τις μεταβλητές με τα αποτελέσματα σε έναν πίνακα “temp”. Μόλις τα πάρει όλα τα βάζει στον πίνακα “results”.

Τέλος από οποιοδήποτε case του switch βγει, πηγαίνει στην γραμμή 187 όπου θα γυρίσει σε JSON μορφή τον πίνακα results όπου έχει τα αποτελέσματα από το ανάλογο πίνακα που θέλουμε που βγήκε από το ανάλογο case.

```

1  <?php
2
3  //database constants
4  define('DB_HOST', 'localhost');
5  define('DB_USER', 'root');
6  define('DB_PASS', '');
7  define('DB_NAME', 'student_iee_ihu');
8
9  //connecting to database and getting the connection object
10 $conn = new mysqli(DB_HOST, DB_USER, DB_PASS, DB_NAME);
11
12 //Checking if any error occurred while connecting
13 if (mysqli_connect_errno()) {
14     echo "Failed to connect to MySQL: " . mysqli_connect_error();
15     die();
16 }
17
18 mysqli_set_charset($conn, "utf8");
19
20 if(isset($_GET['table'])) {
21     $table = $_GET['table'];
22 }
23
24 switch($table) {
25     case 'teachers':
26         //creating a query
27         $stmt = $conn->prepare("SELECT `id`, `name`, `email`, `personal_site`, `category`, `link` FROM `teachers`");
28
29         //executing the query
30         $stmt->execute();
31
32         //binding results to the query
33         $stmt->bind_result($id, $name, $email, $personal_site, $category, $link);
34
35         $results = array();
36
37         //traversing through all the result
38         while($stmt->fetch()){
39             $temp = array();
40             $temp['id'] = $id;
41             $temp['name'] = $name;
42             $temp['email'] = $email;
43             $temp['personal_site'] = $personal_site;
44             $temp['category'] = $category;
45             $temp['link'] = $link;
46             array_push($results, $temp);
47         }
48         break;
49     case 'courses':

```

Σχήμα 3.12: Το αρχείο PHP API για την άντληση πληροφοριών από την απομακρυσμένη βάση δεδομένων SQL.

```

163     case 'useful_websites':
164         $stmt = $conn->prepare("SELECT `id`, `title`, `description`, `category`, `link` FROM `useful_websites`");
165
166         $stmt->execute();
167
168         $stmt->bind_result($id, $title, $description, $category, $link);
169
170         $results = array();
171
172         while($stmt->fetch()){
173             $temp = array();
174             $temp['id'] = $id;
175             $temp['title'] = $title;
176             $temp['description'] = $description;
177             $temp['category'] = $category;
178             $temp['link'] = $link;
179             array_push($results, $temp);
180         }
181         break;
182     default:
183         // Default
184 }
185
186 //displaying the result in json format
187 echo json_encode($results);

```

Σχήμα 3.13: Η συνέχεια από το αρχείο PHP API για την άντληση πληροφοριών από την απομακρυσμένη βάση δεδομένων SQL.

3.5 Οι βιβλιοθήκες που χρησιμοποιήθηκαν

3.5.1 Glide

Η Glide είναι μια γρήγορη και καλή βιβλιοθήκη όπου μπορεί κάποιος να φορτώσει εικόνες στο Android, χωρίς κάποια δυσκολία. Το Glide προσφέρει ένα πολύ εύχρηστο API για χρήση.

Υποστηρίζει λήψη, αποκωδικοποίηση, προβολή φωτογραφιών, βίντεο και GIF[28].

Ένα απλό παράδειγμα:

```

Glide.with(fragment)
    .load(url)
    .into(imageView);

```

Στο παράδειγμα δείχνει ότι στο συγκεκριμένο Fragment είτε Activity θα φορτώσει το URL όπου μπορεί να είναι είτε κάποιος διαδικτυακός ιστότοπος είτε κάποιο αρχείο μέσα από την εφαρμογή. Το βάζει σε κάποιο συγκεκριμένο σημείο, πχ ένα imageView.

Η συγκεκριμένη βιβλιοθήκη χρησιμοποιήθηκε οπουδήποτε υπάρχουν εικόνες μέσα στην εφαρμογή. Είναι μια από τις βασικότερες βιβλιοθήκες της εφαρμογής, όπως και στις περισσότερες που φτιάχνονται συνήθως.

3.5.2 Glide Transformations

Η Glide Transformations είναι μια απλή βιβλιοθήκη όπου παρέχει μια ποικιλία μετατροπών για την βιβλιοθήκη Glide ώστε να υπάρχουν διάφορες δυνατότητες για επεξεργασία των πολυμέσων[29].

3.5.3 MotionToast

Η MotionToast είναι μια χρήσιμη βιβλιοθήκη όσο αφορά την εύκολη και γρήγορη επεξεργασία της εμφάνισης των toast που εμφανίζονται μέσα στην εφαρμογή.

Υποστηρίζει φωτεινό και σκοτεινό θέμα αυτόματα. Έχει πρότυπα για μηνύματα επιτυχίας, προβλήματος, προειδοποίησης ή πληροφόρησης του χρήστη. Μπορούμε και εδώ εύκολα να βάλουμε πόσο θα κρατήσει, που θα εμφανιστεί ή τι γραμματοσειρά θα έχει[30].

Ένα απλό παράδειγμα:

```
MotionToast.createColorToast(this, "Upload Completed!",
    MotionToast.TOAST_SUCCESS,
    MotionToast.GRAVITY_BOTTOM,
    MotionToast.LONG_DURATION,
    ResourcesCompat.getFont(this, R.font.helvetica_regular))
```

Στο παράδειγμα δείχνει ότι εμφανίζεται το συγκεκριμένο κείμενο “Upload Completed!”, θα είναι τύπου Success μήνυμα, θα εμφανιστεί χαμηλά, θα κρατήσει περισσότερο χρόνο μιας και είναι Long και όχι Short. Τέλος, θα έχει την συγκεκριμένη γραμματοσειρά.

3.5.4 Navigation Components

Η Navigation Components είναι μια από τις σημαντικότερες βιβλιοθήκες για μια μεγάλη ποικιλία εφαρμογών. Περιέχει αρκετές λειτουργίες και δυνατότητες προς χρήση.

Μέσω αυτής της βιβλιοθήκης χτίστηκε όλη η λογική και η πλοήγηση μεταξύ των Fragments. Καθώς τα πλάγια μενού όπου συνδέονται με το graph του Navigation Component.

Επίσης μια επιπρόσθετη λειτουργία είναι τα Safe Args όπου επιτρέπει την μεταφορά δεδομένων μέσω παραμέτρων μεταξύ των Safe Args. Αρκεί να δηλωθούν τα dependencies και το πρόσθετο στο Gradle[31].

3.5.5 Firebase UI

Η Firebase UI είναι χρήσιμη βιβλιοθήκη όταν η εφαρμογή υλοποιείτε με την Firebase. Για τον λόγο ότι η βιβλιοθήκη δίνει μια σειρά από μεθόδους κλπ. για ευκολότερη σύνδεση των layouts elements με το Firebase API[32].

3.5.6 Google Maps

Η Google Maps είναι η πρώτη βιβλιοθήκη που χρησιμοποιείται όσο αφορά τους χάρτες. Παρέχει μια μεγάλη ποικιλία από λειτουργίες και δυνατότητες σε ότι έχει σχέση με τους χάρτες. Όπως προβολή διάφορων τύπου χάρτη, προσθήκη και προβολή σημείων, κουμπιά βοηθητικής πλοήγησης κλπ.[33].

3.5.7 EasyPermissions-ktx

Η EasyPermissions-ktx είναι βιβλιοθήκη που αφορά την εύκολη διαχείριση των δικαιωμάτων μιας εφαρμογής. Παρέχει μεθόδους για τον ευκολότερο έλεγχο στο αν ο χρήστης έχει δεχτεί ή όχι κάποιο δικαίωμα όπου χρειάζεται η εφαρμογή την ώρα που λειτουργεί. Έτσι κρατείτε ο κώδικας καθαρός και ασφαλής από σφάλματα[34].

3.5.8 MaterialStyledDialogs

Η MaterialStyledDialogs είναι μια βιβλιοθήκη για εύκολη και ομορφότερη προβολή αναδυόμενων παραθύρων. Έτσι κρατείτε ο κώδικας καθαρός και ασφαλής από σφάλματα.

Ένα απλό παράδειγμα:

```
val dialog = MaterialStyledDialog.Builder(this)
    .setTitle("Awesome!")
    .setDescription("Your feedback is always welcome.")
    .setIcon(R.drawable.ic_launcher)
    .build()
dialog.show()
```

Στο παράδειγμα δείχνει ότι θα εμφανιστεί το συγκεκριμένο αναδυόμενο παράθυρο με τίτλο “Awesome!” και περιγραφή “Your feedback is always welcome”. Τέλος να έχει την συγκεκριμένη εικόνα και να εμφανιστεί[35].

3.5.9 Material Dialog Input

Η Material Dialog Input είναι μέρος μιας γενικότερης βιβλιοθήκης της Material Dialogs για εύκολη και ομορφότερη προβολή αναδυόμενων παραθύρων όπως και η προηγούμενη αλλά η συγκεκριμένη έχει ευκολότερη δυνατότητα για αναδυόμενα παράθυρα που πρέπει να πάρει δεδομένα από τον χρήστη[36].

3.5.10 MMKV

Η MMKV είναι αποτελεσματική, μικρή βιβλιοθήκη. Είναι εύκολη στην χρήση να αποθηκεύσει key-value στην εφαρμογή, όπως είναι και τα SharedPreferences, με την διαφορά ότι είναι αρκετά πιο βολική. Επίσης πιο γρήγορη στην εγγραφή[37].

Ένα απλό παράδειγμα:

```
val kv = MMKV.mmkvWithID("languageMode")

kv?.encode("string", lang)
```

Στο παράδειγμα δείχνει ότι φτιάχνει μια μεταβλητή με id “languageMode”, σε αυτή την τιμή θα μπει ένα αλφαριθμητικό δεδομένο με όνομα “lang”.

Ένα δεύτερο παράδειγμα:

```
val kv = MMKV.mmkvWithID("languageMode")
kv?.decodeString("string")
```

Στο παράδειγμα δείχνει τον τρόπο που παίρνει την τιμή από το id “languageMode”.

3.5.11 Coroutines

Οι Coroutines είναι η προτεινόμενη λύση για ασύγχρονο προγραμματισμό σε Android. Υπάρχει μια συλλογή από βιβλιοθήκες που βοηθάνε στην υλοποίηση των Coroutines.

Βασικά θετικά των Coroutines είναι:

- Είναι ελαφριές και μπορούν να τρέξουν πολλές ταυτόχρονα.
- Υπάρχουν λιγότερες διαρροές μνήμης.

Με αυτό τον τρόπο βελτιώνετε κατά πολύ η παραγωγικότητα και η αποτελεσματικότητα της εφαρμογής[38].

3.5.12 Room Components

Η Room Components είναι μια ολοκληρωμένη βιβλιοθήκη με πληθώρα λειτουργιών και δυνατοτήτων για τοπική αποθήκευση στην εφαρμογή. Χρησιμοποιεί SQLite βάση διότι είναι μια ποιο ελαφριά έκδοση της SQL. Επιτρέπει ευκολά και απλά να αποθηκεύουμε τα δεδομένα μας τοπικά.

Μια απλή υλοποίηση της είναι να δημιουργηθούν τουλάχιστον μια κλάση για Room Database, μια τουλάχιστον για DAO και τέλος τουλάχιστον να έχει μια κλάση για το Entity πχ. Car.

Μέσα σε ένα αρχείο Room Database δηλώνονται όλα τα Entities και τα Dao. Μπορεί να δημιουργηθούν διάφορες βοηθητικές μέθοδοι, π.χ. για να παίρνει ή να καταστρέφει η εφαρμογή την Room Database.

Μέσα σε ένα αρχείο Dao δηλώνονται όλα τα ερωτήματα (Query, Inserts, Deletes κλπ.). Γράφοντας κάποια tags και δημιουργώντας μεθόδους.

Μέσα σε ένα αρχείο Entity είναι το αντικείμενο, πχ. Car, προθέτονται κάποια tags όπου χρειάζεται η Room ώστε να τα διαβάσει.

Ένας καλός τρόπος υλοποίησης είναι να δημιουργείτε για κάθε διαφορετικό Entity και ένα διαφορετικό Dao ώστε να υπάρχει καλύτερη ομαδοποίηση, έλεγχος του κώδικα για καλύτερη επεκτασιμότητα και ευκολία οργάνωσης[39].

3.5.13 Lifecycle Components

Οι Lifecycle Components είναι μια συλλογή από βιβλιοθήκες όπου βοηθάνε πολύ στις δραστηριότητες και τα τμήματα που αποτελούν βασικά στοιχεία για την δημιουργία του UI. Καθώς ο χρήστης κάνει περιήγηση στην εφαρμογή. Τα στοιχεία περνούν από διαφορετικές καταστάσεις στον κύκλο ζωής του Android, άρα η κατανόηση του κύκλου ζωής του Android και η καλύτερη εκμετάλλευση βοηθάει πολύ την ανάπτυξη της εφαρμογής.

Αυτή η συλλογή βιβλιοθηκών βοηθάνε πολύ στην καλύτερη διαχείριση και εκμετάλλευση του κύκλου ζωής του Android, μιας και προσφέρουν διάφορες μεθόδους και λειτουργίες ώστε να χρησιμοποιείται καλύτερα ο κύκλος ζωής στα στοιχεία. Με αποτέλεσμα την καλύτερη εκμετάλλευση των πόρων και λιγότερα σφάλματα[40].

3.5.14 Espresso Test

Η Espresso Test είναι μια ολοκληρωμένη βιβλιοθήκη που ασχολείται με τις δοκιμές αλληλεπιδράσεων των χρηστών σε μια εφαρμογή. Απλοποιεί αρκετά τις διαδικασίες των δοκιμών. Έχει μια ποικιλία λειτουργιών ώστε να μπορεί να ελεγχθούν όσο καλύτερα γίνεται οι εφαρμογές που καλούνται να μην παραλείπονται.

Επίσης υπάρχουν πολλές επεκτάσιμες βιβλιοθήκες που συνεργάζονται ή καλύπτουν την Espresso Test από διάφορους δημιουργούς.

Ένα μικρό παράδειγμα:

```
onView(withId(R.id.login_email)).perform(replaceText("georgekara2010@yahoo.gr"))
onView(withId(R.id.login_password)).perform(replaceText("password"))
onView(withId(R.id.login_button)).perform(click())
```

Στο συγκεκριμένο παράδειγμα, έχει υλοποιηθεί στην εφαρμογή, παίρνει από τα view τα τρία id. Στα δυο πρώτα τοποθετεί το κείμενο που έχει μέσα στις παρενθέσεις. Στο τρίτο id εκτελεί την εντολή “click”, με αυτόν τον τρόπο γίνεται έλεγχος σύνδεσης στην εφαρμογή.

Αν πετύχει το τεστ, θα βγάλει πράσινη έγκριση αλλιώς θα βγάλει κάποιο σφάλμα ή κόκκινη έγκριση. Επίσης δείχνει λεπτομερώς την κάθε εντολή που εκτελεί ώστε να υπάρχει καλύτερος έλεγχος στην εκτέλεση[41].

3.5.15 Retrofit2

Η Retrofit2 είναι από τις καλύτερες βιβλιοθήκες για HTTP επικοινωνία. Έχει πάρα πολλές δυνατότητες και κάνει πολύ εύκολη την διαχείριση τέτοιων ερωτημάτων και διαχείριση μορφών JSON. Κάθε μέθοδος πρέπει να έχει ένα HTTP σχολιασμό που παρέχει τη μέθοδο αιτήματος και την σχετική διεύθυνση URL. Υπάρχουν 8 ενσωματωμένοι σχολιασμοί: HTTP, GET, POST, PUT, PATCH, DELETE, OPTIONS και HEAD[42].

Μερικά παραδείγματα:

```
@GET("/local_api.php?table=teachers")
suspend fun getTeachers(): Response<List<Teacher>>
```

```
@GET("/users")
suspend fun getUsers(): Response<List<Users>>
```

Στο πρώτο παράδειγμα δείχνει πως παίρνει μέσω μιας παράμετρου την “table” τους καθηγητές με την βοήθεια μιας λίστας σε μια απάντηση. Στο δεύτερο παράδειγμα δείχνει πως παίρνει τους χρήστες χωρίς την βοήθεια κάποιου παραμέτρου απευθείας από το URL.

3.5.16 ExpandableTextView

Η ExpandableTextView είναι μια απλή βιβλιοθήκη όταν στην εφαρμογή χρειάζονται κείμενα σε μορφή επέκταση / σύμπτυξη. Είναι πολύ απλή και ευκολά χρησιμοποιούμενη[43].

3.5.17 ViewPager2

Η ViewPager2 είναι μια βιβλιοθήκη όπου βοηθάει να φτιαχτούν ωραία UI όταν η εφαρμογή θα έχει κάποια σελίδα που θα έχει μέσα της διαδοχικές σελίδες σαν βιβλίο είτε για την ανάλλαξη μεταξύ tabs μέσα στην σελίδα[44].

3.5.18 RecyclerView

Η RecyclerView είναι βιβλιοθήκη για να διευκολύνει την καλύτερη προβολή μεγάλων συνόλων δεδομένων. Δημιουργεί δυναμικά τα στοιχεία όταν χρειάζονται. Υποδηλώνει το όνομα της, ανακυκλώνει αυτά τα στοιχεία όταν μετακινείται η οθόνη προς κάποια κατεύθυνσή χωρίς να τα καταστρέφει.

Γενικά, έχει πάρα πολλές δυνατότητες προς χρήση. Πλέον είναι μια από τις default βιβλιοθήκες που χρησιμοποιούνται σε συνεργασία με την CardView πολλές φορές[45].

3.5.19 CardView

Η CardView είναι απλή βιβλιοθήκη που φτιάχνει και εμφανίζει πληροφορίες σε μορφή καρτών, έχει αρκετές ρυθμίσεις ώστε να το προσαρμόσει ο χρήστης όπως θέλει.

Συνήθως χρησιμοποιείται με την βιβλιοθήκη RecyclerView, Αν και χρησιμοποιείται και αρκετά μονή της για στατικά αντικείμενα στο UI. Επίσης είναι ανάλογα με την δημιουργικότητα του καθενός τι θα σκεφτεί να κάνει[46].

3.6 Το ελληνοαγγλικό σύστημα

Η εφαρμογή παρέχει δυο γλώσσες για διάδραση, τα Ελληνικά και τα Αγγλικά που είναι η κύρια παγκόσμια γλώσσα. Για να μπορεί να χρησιμοποιηθεί και από άτομα που έρχονται από το εξωτερικό είτε από άτομα που την έχουν σε μεγαλύτερη προτίμηση.

Για να γίνει πιο εύκολη αυτή η λειτουργία, πρέπει όλα τα Strings που δημιουργούνται στα layouts και στον κώδικα να μην είναι hardcoded. Πρέπει να γίνονται μεταβλητές στο αρχείο strings.xml. Για να γίνει αυτό, είναι πολύ εύκολο μιας και το Android Studio δίνει αυτή την δυνατότητα. Την ώρα που γράφετε κάποιο κείμενο, βγάζει επιλογή για μετατροπή σε μεταβλητή, βάζοντας και ένα όνομα. Έτσι, χρησιμοποιείται αυτή στο σημείο όπου θα έμπαινε το κείμενο.. Με αυτό τον τρόπο μπορεί εύκολα να προστεθούν όσες γλώσσες χρειαστεί απλά πατώντας δεξί κλικ στο αρχείο strings.xml, στο Open Translations Editor όπως φαίνεται στο Σχήμα 3.14 και να προσθέσει την γλώσσα που χρειάζεται. Μετά το μόνο που πρέπει να γίνει είναι να μεταφραστούν όλα τα κείμενα στην ανάλογη γλώσσα. Αυτό δεν αρκεί για να είναι η εφαρμογή πλήρως μεταφρασμένη σε μια άλλη γλώσσα. Πρέπει να μεταφραστούν όλα τα αρχεία πχ PDFs που ανοίγουν. Ώστε όταν ανοίγει το PDF, να ανοίγει με την γλώσσα προτίμησης. Τέλος το ίδιο πρέπει να γίνει και με τις ιστοσελίδες που ανοίγουν, στην σελίδα του τμήματος. Καλό ήταν ότι είναι ήδη μεταφρασμένη στις δυο γλώσσες που χρειαζόταν η συγκεκριμένη εφαρμογή, όπως σε μερικά PDF. Έτσι με αυτό τον τρόπο η εφαρμογή ανάλογα με την προτίμηση που έχει διαλέξει ο χρήστης θα του παρέχει και το περιεχόμενο στην ανάλογη γλώσσα.

Key	Resource Folder	Untranslatable	Default Value	Greek (el) in Greece (GR)
old_email	app/src/main/res	<input type="checkbox"/>	Old Email	Παλιό Ηλ. Διεύθυνσης
update_email	app/src/main/res	<input type="checkbox"/>	Update Email	Ενημέρωση Ηλ. Διεύθυνσης
send_verification_email	app/src/main/res	<input type="checkbox"/>	Send Verification Email	Αποστολή Ηλ. Διεύθυνσης Επαλήθευ
help_you	app/src/main/res	<input type="checkbox"/>	Click to help you!	Κάντε κλικ για να σας βοηθήσουμε!
help_title	app/src/main/res	<input type="checkbox"/>	Help Page	Σελίδα Βοήθειας
first_day	app/src/main/res	<input type="checkbox"/>	First Day	Πρώτη Μέρα
zoom_rooms	app/src/main/res	<input type="checkbox"/>	Zoom Rooms	Zoom Δωμάτια
r1_r8	app/src/main/res	<input checked="" type="checkbox"/>	R1-R8	
e1_e8	app/src/main/res	<input checked="" type="checkbox"/>	E1-E8	
e1	app/src/main/res	<input checked="" type="checkbox"/>	E1	
e2	app/src/main/res	<input checked="" type="checkbox"/>	E2	

Σχήμα 3.14: Το Translations Editor για την επεξεργασία των γλωσσών της εφαρμογής.

3.7 Η δομή της εφαρμογής

3.7.1 Το αρχείο AndroidManifest.xml

Στο αρχείο AndroidManifest.xml όπως φαίνεται στο Σχήμα 3.15, 3.16, σε πρώτη φάση προστεθήκανε όλα τα δικαιώματα που θα χρειαστεί η εφαρμογή ώστε να λειτουργεί. Μετά προστέθηκε η κλάση CoreApplication σαν κύριο Application Class του Module της εφαρμογής. Επιπλέον δηλώθηκαν όλα τα Activities της εφαρμογής που υπάρχουν επειδή πρέπει να δηλώνονται.

Τέλος προστέθηκαν δυο meta-data, το πρώτο είναι το Google API KEY για το Google Maps που χρειάζεται για να λειτουργήσει και το δεύτερο είναι για το αρχείο με τα Fonts που έχουν κατεβεί από την Google μέσω του Android Studio.

```

<!-- Permissions -->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" tools:ignore="ScopedStorage" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

<uses-permission android:name="android.permission.CALL_PHONE" />

```

Σχήμα 3.15: Τα permissions από το αρχείο AndroidManifest.xml

```

<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSyDEwx8DH82xyMdIaE4hS5Svr822nz-VEz8" />
<meta-data
    android:name="preloaded_fonts"
    android:resource="@array/preloaded_fonts" />

```

Σχήμα 3.16: Τα meta-data από το αρχείο AndroidManifest.xml

3.7.2 Η δομή των αρχείων Kotlin

3.7.2.1 Data

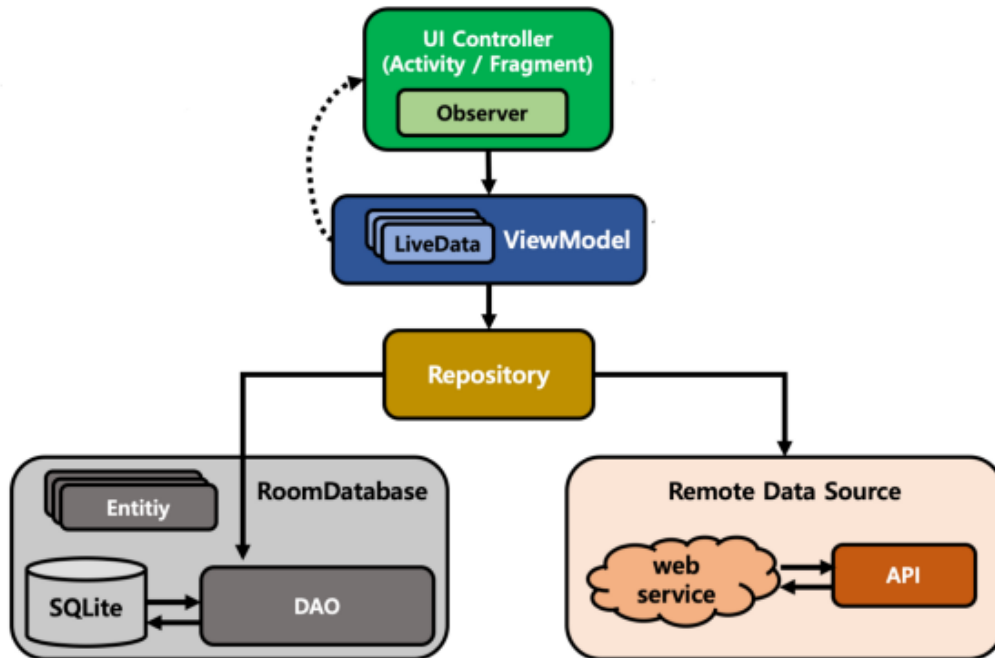
Ο φάκελος Data διαχειρίζεται τα δεδομένα της εφαρμογής, τα τοπικά και τα απομακρυσμένα. Γενικά ακολουθείτε κατά κύριο λόγο η αρχιτεκτονική MVVM όπως φαίνεται και στο Σχήμα 3.17.

Η αρχιτεκτονική MVVM είναι μια αρχιτεκτονική Model-View-ViewModel που καταργεί την στενή σύζευξη μεταξύ κάθε στοιχείου στην εφαρμογή. Το θετικό σε αυτή την αρχιτεκτονική είναι ότι τα παιδιά δεν έχουν άμεση αναφορά στον γονέα τους, αλλά μόνο αναφορά από παρατηρήσιμα[47][48].

Model: Το model αντιπροσωπεύει τα δεδομένα και την επιχειρηματική λογική της εφαρμογής. Αποτελείται από την επιχειρηματική λογική (τοπική, απομακρυσμένη ή και τα δυο), την πηγή δεδομένων, τα μοντέλα της εφαρμογής και τα αποθετήρια[47][48].

View: Το view αποτελείται από τον κώδικα UI (Activities, Fragments, κλπ.) και γενικά τα XML. Στέλνει την ενέργεια που θα κάνει ο χρήστης στο ViewModel. [47][48].

ViewModel: Το viewModel υλοποιεί την λογική του μοντέλου (Model) ώστε να μπορεί να χρησιμοποιηθεί από τις όψεις (View). Πρόκειται για την γέφυρα μεταξύ των όψεων και των μοντέλων. Δεν γνωρίζει ποιο view πρέπει να χρησιμοποιήσει καθώς δεν έχει κάποια άμεση αναφορά στο view. Το viewModel δεν θα πρέπει να γνωρίζει με ποιον αλληλοεπιδρά γενικά[47][48].



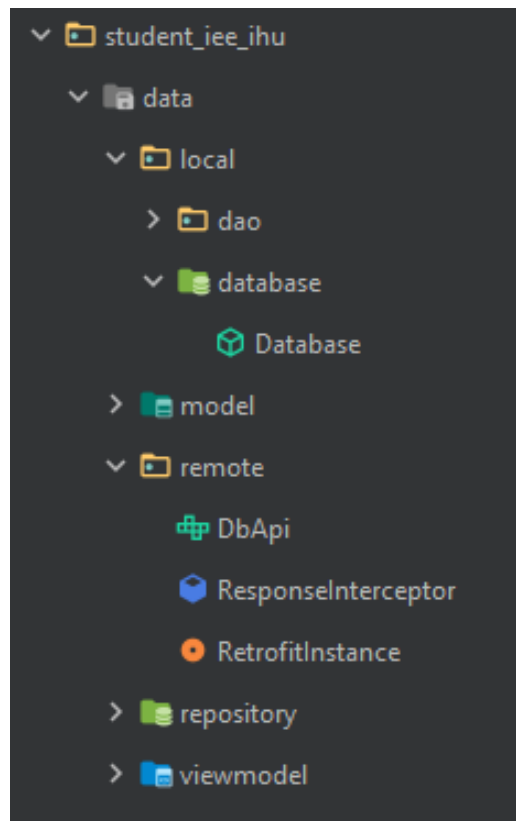
Σχήμα 3.17: Η αρχιτεκτονική MVVM.

Ο φάκελος χωρίζεται στους εξής 5 φακέλους, τον local, τον remote, τον model, τον repository και τον viewModel όπου φαίνονται και στο Σχήμα 3.18.

- Ο φάκελος local περιέχει την τοπική βάση, στην συγκεκριμένη περίπτωση την Room. Χωρίζετε σε δυο φακέλους, τον dao και στον database.
 - Ο φάκελος database περιέχει μόνο ένα αρχείο, το Database όπου είναι η τοπική βάση, όπου δηλώνονται όλες οι κλάσεις και τα dao που υπάρχουν. Όποια μέθοδος χρειαστεί σχετικά με την βάση ή ρύθμιση. Πχ η getDatabase όπου έχει υλοποιηθεί και στην εφαρμογή σε περίπτωση ανάγκης.
 - Ο φάκελος dao περιέχει ένα αρχείο dao για κάθε πίνακα της βάσης έτσι ώστε να διαχωρίζονται, να είναι εύκολα συντηρήσιμα και να μην επηρεάζει το ένα το άλλο. Το

κάθε αρχείο dao έχει όλα τα ερωτήματα που σχετίζονται με την βάση, όπως πχ readData, searchDataBase, insertData κλπ.

- Ο φάκελος remote περιέχει 3 αρχεία, το RetrofitInstance, το ResponseInterceptor και το DbApi. Το αρχείο RetrofitInstance αποτελείται από 3 αντικείμενα. Το client όπου εδώ χρησιμεύει και το δεύτερο αρχείο (ResponseInterceptor) σαν custom αντικείμενο για ένα από τα option του client, επιπλέον εδώ έχει τους χρόνους που θα κάνει το ανάλογο Timeout για connect, read κλπ. Το retrofit όπου παίρνει το client αντικείμενο, το baseUrl και το GsonConverterFactory. Τελευταίο κομμάτι που είναι αυτό που καλείται κιόλας στις επόμενες κλάσεις, ένα αντικείμενο από την κλάση dbApi.
- Ο φάκελος repository περιέχει μια κλάση για κάθε πίνακα όπως και πριν στο local με τα dao. Η κάθε κλάση παίρνει τα τοπικά και τα απομακρυσμένα δεδομένα, συγκεκριμένα καλεί τις μεθόδους από το ανάλογο dao και από το dbApi την ανάλογη μέθοδο για να πάρει τα δεδομένα. Ωστε μετά να τα στείλουμε στα viewModels.
- Ο φάκελος viewModel περιέχει μια κλάση για κάθε πίνακα όπως και πριν στο local με τα dao και τα repository. Η κλάση ViewModel είναι αυτή που μεσολαβεί με τα δεδομένα από τα repositories και τα views.
- Ο φάκελος model περιέχει αρχεία με όλα τα μοντέλα που χρησιμοποιούνται πχ course, teacher κλπ.
- Εκτός φακέλου Data υπάρχει και ένα ακόμα αρχείο, το CoreApplication που είναι το Main Application αρχείο του module, όπου τρέχει όλη την ώρα. Εκεί υπάρχει μια μεταβλητή database και μια για κάθε repository. Και χάρη στο "lazy", θα τρέξουν μόνο όταν η εφαρμογή το ζητήσει. Μια πολύ χρήσιμη τεχνική για καλύτερες επίδοσης της εφαρμογής.

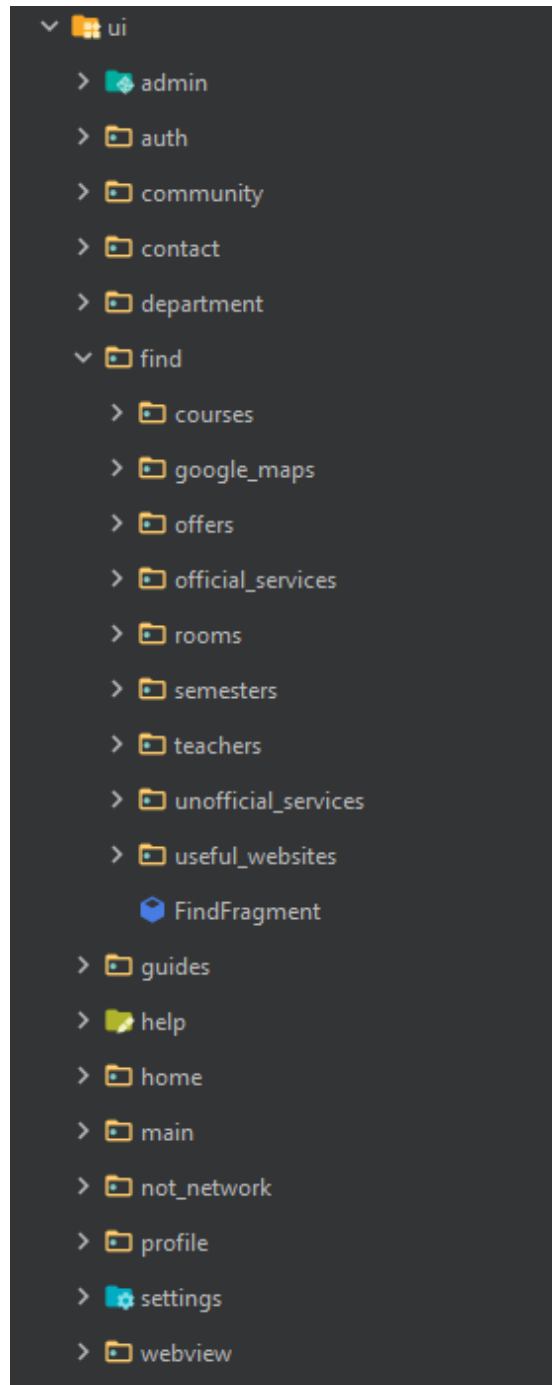


Σχήμα 3.18: Η δομή των φακέλων / αρχείων από το Data.

3.7.2.2 UI

Ο φάκελος UI περιέχει όλα τα Activities, Fragments, Adapters κλπ. Είναι χωρισμένοι σε φακέλους ανά λειτουργία κατά κύριο λόγο. Υπάρχουν συνολικά 13-14 φάκελοι όπως φαίνεται και στο Σχήμα 3.19. Οι admin, auth, community, contact, department, find, guides, help, home, not_network, profile, settings, main, webView. Στην ουσία μόνο το main δεν έχει κάποιες λειτουργίες αλλά τα δυο Activity (Main και AdminMain). Ο φάκελος find, ξανά κατηγοριοποιείται σε φακέλους ανάλογα την κάθε λειτουργία αναζήτησης μέσα στο find.

Υπάρχουν οι εξής 9 φάκελοι στο find, οι courses, google_maps, offers, official_services, unofficial_services, rooms, semesters, teachers, useful_websites. Σε παρακάτω υποκεφάλαιο με τις λειτουργίες θα αναλυθούν περισσότερο τα αρχεία ανάλογα την λειτουργία που έχει το καθένα.



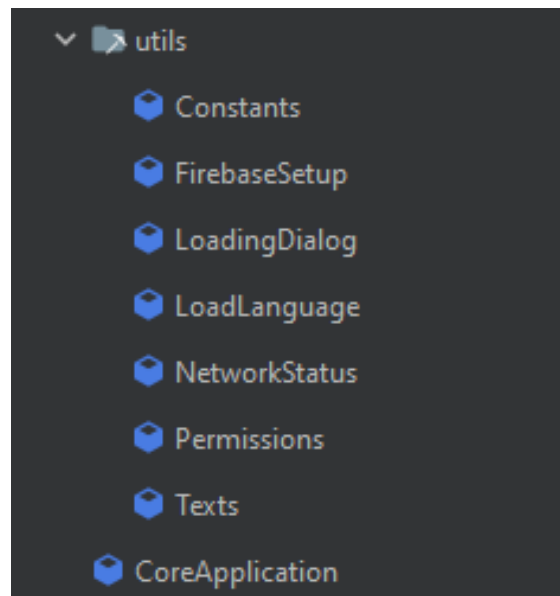
Σχήμα 3.19: Η δομή των φακέλων / αρχείων από το UI.

3.7.2.3 Utils

Ο φάκελος Utils περιέχει 7 αρχεία όπου αφορούν την εφαρμογή όπως φαίνεται και στο Σχήμα 3.20, που θα αναφερθούν και σε τι χρησιμεύουν.

Αυτές οι κλάσεις φτιάχνονται με κύριο λόγο να μην επαναλαμβάνεται κάποιο κομμάτι κώδικα ή για να βρίσκονται σε ένα σημείο συγκεντρωμένα, αποκομμένα για εύκολη συντήρηση και επανάκτηση του κώδικα στην εφαρμογή.

- Το πρώτο αρχείο είναι το Constants, ένα πολύ συνηθισμένο και απαραίτητο αρχείο σε κάθε εφαρμογή. Εκεί αποθηκεύονται διάφορες σταθερές μεταβλητές που μπορούν να χρησιμοποιηθούν οπουδήποτε στην εφαρμογή με την δήλωση της από την κλάση.
- Το δεύτερο αρχείο είναι το FirebaseSetup, κάνει setup την Firebase, έτσι ώστε να χρησιμοποιείται όπου χρειάζεται μέσα στην εφαρμογή οι μεταβλητές του πχ user, database κλπ. και να μην δηλώνονται νέες μεταβλητές συνέχεια σε κάθε κλάση που θα χρησιμοποιηθεί η Firebase.
- Το τρίτο αρχείο είναι το Texts, είναι αυτό που έχει όλα τα κείμενα από Help Page, Department και About.
- Το τέταρτο αρχείο είναι το LoadingDialog, είναι ένα custom loading dialog για να χρησιμοποιείται όταν θέλει η εφαρμογή κάποια αναμονή για φόρτωση.
- Το πέμπτο αρχείο είναι το LoadLanguage, είναι μια μέθοδος που διαβάζει και δίνει ποια γλώσσα είναι επιλεγμένη από τον χρήστη.
- Το έκτο αρχείο είναι το NetworkStatus. Μία μέθοδος που βλέπει αν έχει δίκτυο ή όχι για να πράξει ανάλογα.
- Το έβδομο αρχείο είναι το Permissions, όπου είναι μέθοδοι για αίτημα απόκτησης ή για να δει αν υπάρχει κάποιο δικαίωμα ώστε να χρησιμοποιήσει όταν θέλει να ζητήσει τα δικαιώματα.



Σχήμα 3.20: Η δομή των αρχείων από το Utils.

3.7.3 Η δομή των αρχείων Res

3.7.3.1 Η δομή τους στους φακέλους

Αρχικά όπως είναι σε όλα τα έργα, όλες οι εικόνες αποθηκεύονται στο drawable, κάνοντας έλεγχο αν κάποια εικόνα έχει πολύ μεγάλο μέγεθος, επειδή συνήθως υπάρχουν θέματα με μεγάλες εικόνες σε παλιότερες συσκευές.

Προστέθηκε ο φάκελος font, όπου προστεθήκαν 4 fonts από την Google, ώστε να γίνουν δόκιμες για μελλοντικές αλλαγές στην γραμματοσειρά αν χρειαστεί.

Έπειτα προστέθηκε ο φάκελος menu, όπου έχει τα δυο menu, του διαχειριστή και του απλού χρήστη. Αυτά χρησιμοποιούνται από το draw layout menu. Μπήκαν τα κατάλληλα ονόματα, ετικέτες και εικονίδια.

Αργότερα προστέθηκε ο φάκελος navigation όπου έχει το έγγραφο με όλα τα Fragments και τις μετακινήσεις μεταξύ τους όπως και αν έχει κάποιο argument για μεταφορά. Μπήκαν τα κατάλληλα ονόματα και ετικέτες αντίστοιχα και εδώ.

Ο φάκελος mipmap όπου υπάρχει εξαρχής, αλλάχτηκε το εικονίδιο χρήσης για το icon της εφαρμογής σε όλα τα μεγέθη ώστε να μπει το λογότυπο του τμήματος.

Στον φάκελο values, υπάρχει το αρχείο colors όπου επεξεργάστηκε για να έχει τα χρώματα όπου επιλέχθηκαν για την εφαρμογή.

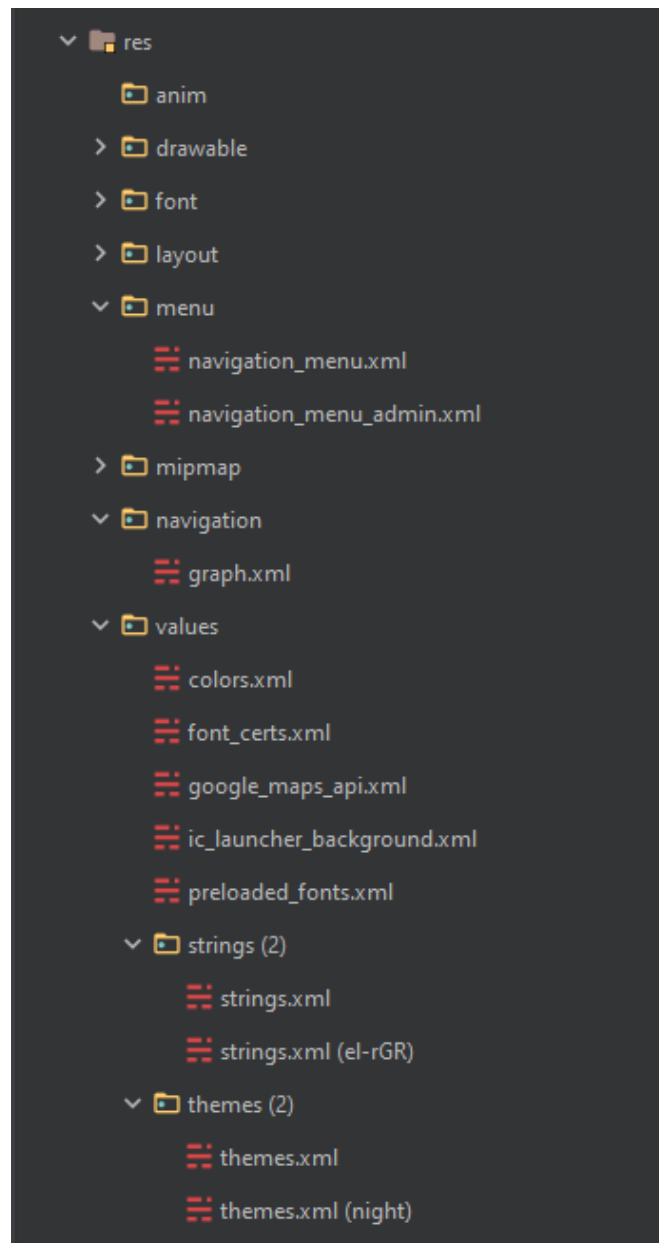
- Το αρχείο font_certs όπου έχει τα κλειδιά για τα fonts.
- Το google_maps_api όπου έχει τον κωδικό για το Google Maps API.
- Ένα αρχείο όπου χρησιμοποιείται για background (απλό λευκό χρώμα) στο icon της εφαρμογής.
- Ένα αρχείο preloaded_fonts έχει μια λίστα με τα fonts όπου υπάρχουν κατεβασμένα στην εφαρμογή.

Στο φάκελο values βρίσκονται δυο υποφάκελοι. Ο φάκελος themes όπου υπάρχουν δυο θέματα, το white και το night theme. Το night theme επεξεργάστηκε με τα κατάλληλα χρώματα και κάποια ρύθμιση περί δικαιωμάτων για το night.

Επίσης ο φάκελος strings έχει ένα αρχείο για κάθε γλώσσα. Στη συγκεκριμένη περίπτωση έχει 2 αρχεία, ένα για τα ελληνικά και ένα για τα αγγλικά. Όσο αφορά αυτά τα αρχεία αναφερόμαστε στο προηγούμενο υποκεφάλαιο για το ελληνοαγγλικό σύστημα.

Τέλος έμεινε ο φάκελος με τα layouts. Είναι όλα τα views για την εφαρμογή. Διότι τα αρχεία είναι πάρα πολλά, είναι δύσκολη η αναζήτηση και η οργάνωση τους. Έχει γίνει ταξινόμηση σε φακέλους ανά λειτουργία και τα υπόλοιπα σε ένα φάκελο όλα μαζί. Έτσι είναι καλύτερα ταξινομημένα και κυρίως ποιο εύκολα επεξεργάσιμα. Στον φάκελο μέσα από το Android Studio δεν φαίνεται η διαφορά αλλά αν θέλει να μετακινηθεί ή να δημιουργηθεί κάποιο view δείχνει που να αποθηκευτεί.

Όλα αυτά φαίνονται και στο Σχήμα 3.21.



Σχήμα 3.21: Η δομή των φακέλων / αρχείων από το Res.

3.7.3.2 Η οργάνωση των Layouts στο Gradle

Για να πραγματοποιηθεί αυτή η ταξινόμηση, θέλει να γίνει ρύθμιση στο build.gradle του module, όπως φαίνεται στο Σχήμα 3.22. Στην ουσία θα πάρει την θέση src/main/res και μετά θα διαβάσει όλους τους υποφακέλους που έχει το layouts, όχι μόνο το layouts όπως είναι στο default. Πχ src/main/res/layouts/layout_auth, src/main/res/layouts/layout_guides κλπ.

Με αυτό το τρόπο διαβάσει όλους τους φακέλους και τα αρχεία και είναι εύκολο να χρησιμοποιηθούν σε αυτή την μορφή.

```

sourceSets {
    main {
        res.srcDirs =
            [
                'src/main/res',
                'src/main/res/layouts/layout_auth',
                'src/main/res/layouts/layout_community',
                'src/main/res/layouts/layout_guides',
                'src/main/res/layouts/layout_profile',
                'src/main/res/layouts/layout_settings',
                'src/main/res/layouts/layout_find',
                'src/main/res/layouts/layout_help',
                'src/main/res/layouts/layout_home',
                'src/main/res/layouts/layout_main',
                'src/main/res/layouts/layout_not_network',
                'src/main/res/layouts/layout_webview',
                'src/main/res/layouts/layout_other',
                'src/main/res/layouts/layout_contact',
                'src/main/res/layouts/layout_admin',
                'src/main/res/layouts/layout_department'
            ]
    }
}

```

Σχήμα 3.22: Η ρυθμίσεις στο Gradle για τα layouts.

3.7.4 Η διαμόρφωση του Gradle

Αρχίζοντας από το γενικό build.gradle του project όπου φαίνεται στο Σχήμα 3.23. Στο buildscript ακολουθήθηκε η αρχιτεκτονική με το να μπαίνουν οι εκδόσεις βιβλιοθηκών σε αυτό το σημείο για ευκολότερη διαχείριση μεταξύ modules σε περίπτωση μελλοντικού διαχωρισμού. Έτσι δεν χρειάζεται να επαναλαμβάνονται οι εκδόσεις, κάνοντας ευκολότερη διαχείριση όσο αφορά την οργάνωση των βιβλιοθηκών. Στα dependencies προστέθηκαν δυο classpath, ένα για τα Google Services και ένα για τα Safe Args όπου χρειάστηκαν.

Όσον αφορά το δεύτερο αρχείο, είναι το build.gradle του module όπου φαίνεται στα Σχήματα 3.24, 3.25 Έχουν προστεθεί τα κατάλληλα πρόσθετα που χρειάστηκαν. Έπειτα ενεργοποιήθηκε το viewBinding. Επιπλέον όπως αναφέρθηκε πριν, προστέθηκε η λογική για τους φακέλους των layouts.

Τέλος προστεθήκαν οι βιβλιοθήκες που χρειάστηκαν, μαζί με ένα σχόλιο το οποίο είναι είτε ατομικό είτε ομαδικό.

```

1 // Top-level build file where you can add configuration options common to all sub-projects/
2 buildscript {
3     ext.kotlin_version = "1.5.21"
4     ext.nav_version = "2.3.5"
5     ext.room_version = "2.3.0"
6     ext.coroutines = "1.5.1-native-mt"
7     ext.glide = "4.12.0"
8     ext.retrofit = "2.9.0"
9     ext.okhttp3 = "5.0.0-alpha.2"
10    repositories {
11        google()
12        mavenCentral()
13    }
14    dependencies {
15        classpath 'com.android.tools.build:gradle:7.0.0'
16        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
17
18        // Google Services
19        classpath "com.google.gms:google-services:4.3.8"
20
21        // Safe Args
22        classpath "androidx.navigation:navigation-safe-args-gradle-plugin:$nav_version"
23
24        // NOTE: Do not place your application dependencies here; they belong
25        // in the individual module build.gradle files
26    }
27 }

```

Σχήμα 3.23: Το αρχείο Gradle του project.

```

1 plugins {
2     id 'com.android.application'
3     id 'com.google.gms.google-services'
4     id 'kotlin-android'
5     id 'kotlin-kapt'
6     id 'kotlin-android-extensions'
7     id 'androidx.navigation.safeargs.kotlin'
8 }
9
10
11 android {
12     compileSdkVersion 31
13     buildToolsVersion "30.0.3"
14
15     defaultConfig {
16         applicationId "eu.seijindemon.student_iee_ihu"
17         minSdkVersion 23
18         targetSdkVersion 31
19         versionCode 1
20         versionName "1.0"
21
22         testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
23     }
24     buildFeatures {
25         viewBinding true
26     }
27     buildTypes {
28         release {
29             minifyEnabled false
30             proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
31         }
32     }
33     compileOptions {
34         sourceCompatibility JavaVersion.VERSION_1_8
35         targetCompatibility JavaVersion.VERSION_1_8
36     }
37     kotlinOptions {
38         jvmTarget = '1.8'
39     }
40     sourceSets {
41         main {
42             dependencies {

```

Σχήμα 3.24: Το αρχείο Gradle του module.

```

118 // Glide
119 implementation "com.github.bumptech.glide:glide:$glide"
120 kapt "com.github.bumptech.glide:compiler:$glide"
121 // Glide Transformations
122 implementation 'jp.wasabeef.glide-transformations:4.3.0'
123 // If you want to use the GPU Filters
124 implementation 'jp.co.cyberagent.android:gpuimage:2.1.0'
125
126 // Google Maps
127 implementation 'com.google.android.gms:play-services-maps:17.0.1'
128
129 // MaterialStyledDialogs
130 implementation 'com.github.javiersantos:MaterialStyledDialogs:3.0.1'
131
132 // Material Dialog Input
133 implementation 'com.afollestad.material-dialogs:input:3.3.0'
134
135 // MMKV
136 implementation 'com.tencent:mmkv-static:1.2.10'
137
138 // Coroutines
139 implementation "org.jetbrains.kotlinx:kotlinx-coroutines-core:$coroutines"
140 implementation "org.jetbrains.kotlinx:kotlinx-coroutines-android:$coroutines"
141 implementation "org.jetbrains.kotlinx:kotlinx-coroutines-play-services:$coroutines"
142
143 // Room Components
144 implementation "androidx.room:room-runtime:$room_version"
145 kapt "androidx.room:room-compiler:$room_version"
146 implementation "androidx.room:room-ktx:$room_version"
147 androidTestImplementation "androidx.room:room-testing:$room_version"
148

```

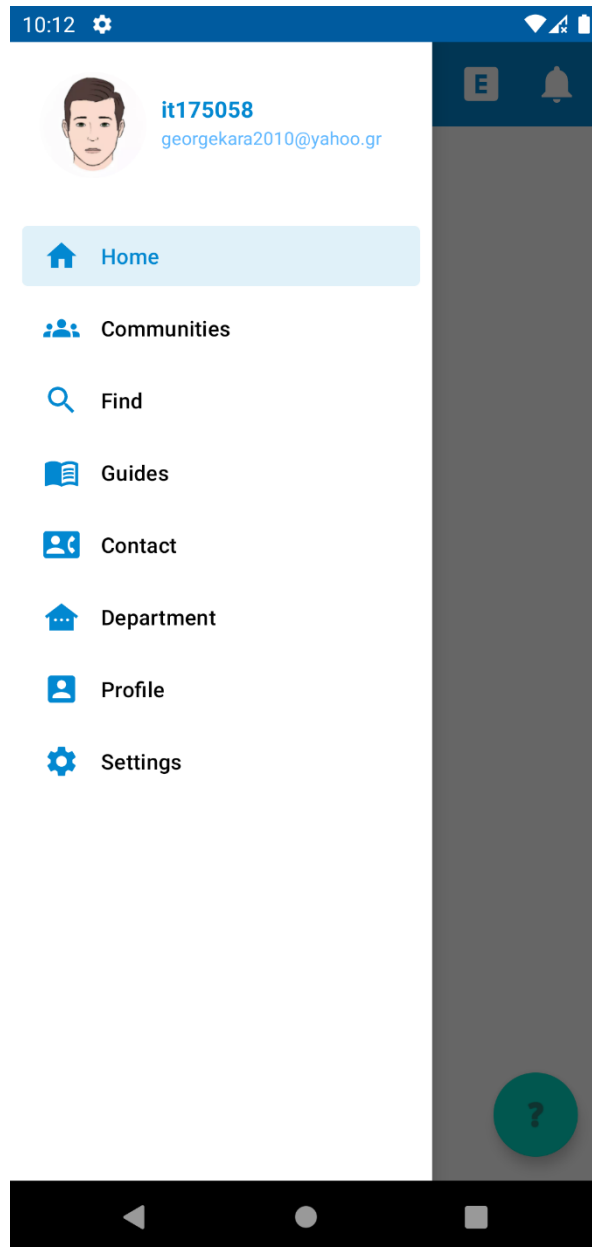
Σχήμα 3.25: Η συνέχεια από το αρχείο Gradle του module.

3.8 Λειτουργίες

3.8.1 Μενού πλοήγησης

Η λειτουργία μενού πλοήγησης που φαίνεται στο Σχήμα 3.26, είναι μια από τις βασικές λειτουργίες. Μέσα από αυτή ο χρήστης πλοηγείται στην εφαρμογή. Η υλοποίηση της έγινε με το Navigation Component, με την βοήθεια δυο navigation menu (ένα για τους απλούς χρήστες και ένα για τους διαχειριστές) και με την βοήθεια ενός εγγράφου. Όλη η πλοήγηση γίνεται μέσω Fragments όπου είναι όλα καταγεγραμμένα στο έγγραφο και οι επιλογές στο μενού πλοήγησης είναι μέσα από τα navigation menu.

Το μενού της πλοήγησης ανοίγει είτε πατώντας πάνω αριστερά το ειδικό κουμπί που έχει, είτε τραβώντας την οθόνη από αριστερά προς δεξιά. Το μενού της πλοήγησης κλείνει είτε πατώντας πίσω είτε τραβώντας την οθόνη από δεξιά προς αριστερά.



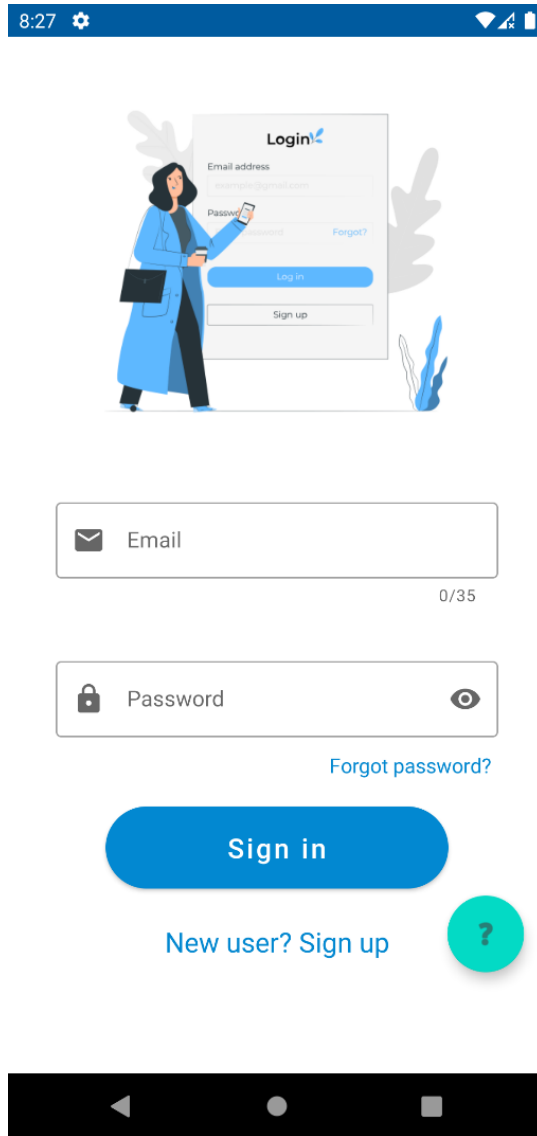
Σχήμα 3.26: Το μενού πλοήγησης στην εφαρμογή.

3.8.2 Αυθεντικοποίηση

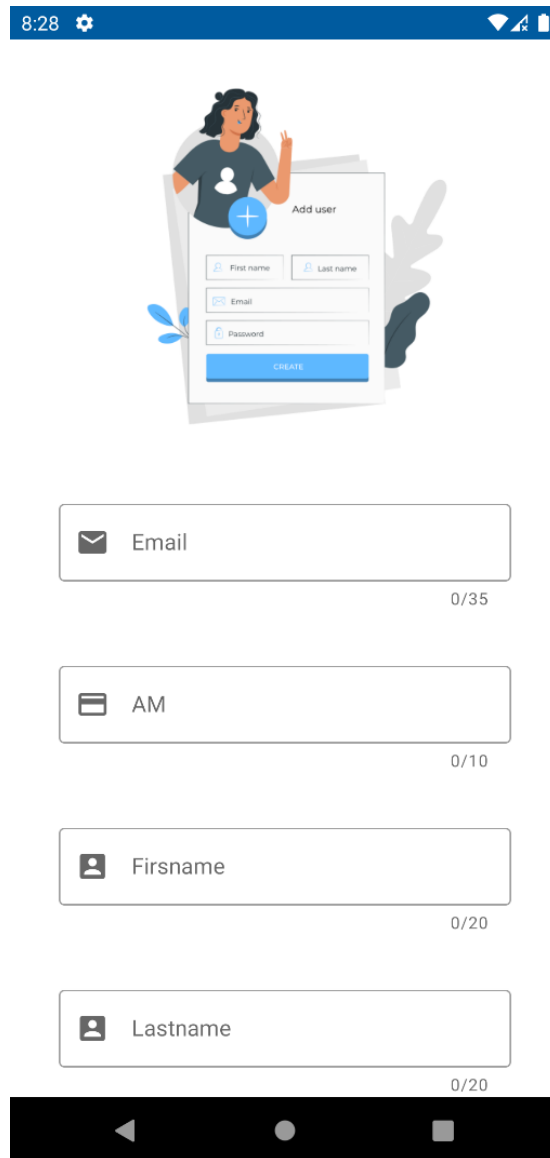
Η αυθεντικοποίηση της εφαρμογής έχει αρκετά τμήματα, τα οποία θα αναφερθούν παρακάτω. Όλα τα θέματα αυθεντικοποίησης γίνονται με την Firebase όπως έχει αναφερθεί και σε προηγούμενο υποκεφάλαιο. Σε όλες τις λειτουργίες της αυθεντικοποίησης υπάρχουν τα κατάλληλα μηνύματα ορθής ή λάθος λειτουργίας.

- Πρώτη λειτουργία της αυθεντικοποίησης είναι η εγγραφή (Sign up) όπου φαίνεται στο Σχήμα 3.27. Η εγγραφή στην εφαρμογή γίνεται μέσω επιβεβαίωσης email. Η διαδικασία εγγραφής δίνεται από την επιλογή του Activity σύνδεσης, όπου υπάρχει η επιλογή να ανοίξει το Activity για την εγγραφή. Στην φόρμα της εγγραφής υπάρχουν όλα τα κατάλληλα μηνύματα ελέγχου, βοηθητικά εικονίδια κλπ. Αν γίνει επιτυχής η εγγραφή, τότε θα σταλθεί μήνυμα επιβεβαίωσης στο email του χρήστη, το οποίο θα πρέπει να δεχτεί.

- Δεύτερη λειτουργία της αυθεντικοποίησης είναι η είσοδος (Sign in) όπου φαίνεται στο Σχήμα 3.28. Όταν ξεκινάει η εφαρμογή, ανοίγει σε αυτή την σελίδα. Όταν ανοίγει ο χρήστης την εφαρμογή αναζητά αν υπάρχει χρήστης ήδη συνδεδεμένος από προηγούμενη σύνδεση. Αν υπάρχει ελέγχει αν είναι επιβεβαιωμένος χρήστης. Αν είναι, ελέγχει αν είναι απλός χρήστης ή διαχειριστής και ανοίγει το ανάλογο Activity με το ανάλογο navigation menu. Αν δεν είναι επιβεβαιωμένος χρήστης, τον ρωτάει να του στείλει ξανά email επιβεβαίωσης σε περίπτωση που δεν στάλθηκε ή χάθηκε το email. Όσπου να επιβεβαιωθεί το email του χρήστη δεν μπορεί να πραγματοποιήσει σύνδεση στην εφαρμογή. Αν δεν υπάρχει χρήστης ήδη όταν κάνει τον έλεγχο, τότε συμπληρώνει ο χρήστης τα στοιχεία του και αν είναι σωστά θα κάνει σύνδεση με τους ίδιους ελέγχους που κάνει και αυτόματα στην αρχή. Επίσης όπως και στην εγγραφή υπάρχουν όλα τα κατάλληλα μηνύματα ελέγχου και βοηθητικά εικονίδια. Όταν γίνει σύνδεση, βγάζει και το κατάλληλο μήνυμα ότι συνδέθηκε.
- Τρίτη λειτουργία της αυθεντικοποίησης είναι η έξοδος (Logout). Η οποία είναι μια απλή λειτουργία που υπάρχει στην σελίδα των ρυθμίσεων (πάνω πάνω πρώτη). Αν γίνει χρήση του, κάνει αποσύνδεση και πηγαίνει στην σελίδα της εισόδου χρήστη.
- Τετάρτη λειτουργία της αυθεντικοποίησης είναι η αλλαγή email (Change email). Βρίσκετε στην σελίδα των ρυθμίσεων. Αν ο χρήστης θελήσει να το κάνει, θα πρέπει να βάλει ξανά τον κωδικό του μιας και πρέπει να ξανά κάνει επαλήθευση ότι είναι ο ίδιος χρήστης. Έπειτα θα συμπληρώσει το νέο του email. Θα χρειαστεί επίσης μια επιβεβαίωση όταν θα ξανά πάει να συνδεθεί και είναι έτοιμη η αλλαγή.
- Πέμπτη λειτουργία της αυθεντικοποίησης είναι αλλαγή password (Change password). Σε περίπτωση που έχει ξεχαστεί ο κωδικός ακόμα και αν απλά είναι επιθυμία του χρήστη, για την αλλαγή του. Στην πρώτη περίπτωση στην σελίδα εισόδου υπάρχει η επιλογή για να σταλθεί email για να αλλαχθεί ο κωδικός και απλά ο χρήστης συμπληρώνει το email ώστε να σταλθεί το κατάλληλο email επιβεβαίωσης. Στην δεύτερη περίπτωση υπάρχει επιλογή στην σελίδα ρυθμίσεων και απλά θα ζητήσει επιβεβαίωση ότι όντως θέλει να σταλθεί email για αλλαγή κωδικού στο email όπου είναι εκείνη την στιγμή συνδεδεμένο αυτόματα.
- Έκτη λειτουργία της αυθεντικοποίησης είναι η διαγραφή λογαριασμού (Delete Account). Στην σελίδα των ρυθμίσεων υπάρχει η επιλογή “διαγραφή χρήστη”. Θα ζητηθεί μια επιβεβαίωση ότι όντως θέλει διαγραφή λογαριασμού. Αν ο χρήστης το επιβεβαιώσει, θα μεταφερθεί στην σελίδα εισόδου και θα έχει διαγραφεί ο λογαριασμός του χρήστη.



Σχήμα 3.27: Η σελίδα σύνδεσης.



8:28 [Settings] [Signal] [Battery]

Add user

First name Last name

Email

Password

CREATE

Email 0/35

AM 0/10

Firsname 0/20

Lastname 0/20

[Navigation Bar]

Σχήμα 3.28: Η σελίδα εγγραφής.

3.8.3 Προφίλ (Profile)

Στη λειτουργία προφίλ (Profile) όπως φαίνεται στο Σχήμα 3.29, ο χρήστης μπορεί να δει τα στοιχεία του στην εφαρμογή και δίνεται η δυνατότητα αλλαγής τους. Υπάρχουν κατάλληλοι έλεγχοι και τα βοηθητικά εικονίδια για αυτή τη λειτουργία. Δίνεται η επιπλέον επιλογή προσθήκης τηλεφώνου, να προστεθεί ή να αλλάξει την φωτογραφία προφίλ. Πατώντας το κουμπί για αποθήκευση στοιχείων, θα εμφανίσει κατάλληλο μήνυμα ότι έγινε η αλλαγή. Επίσης οι καινούργιες αλλαγές γίνονται απευθείας.

Σχήμα 3.29: Η σελίδα του προφίλ.

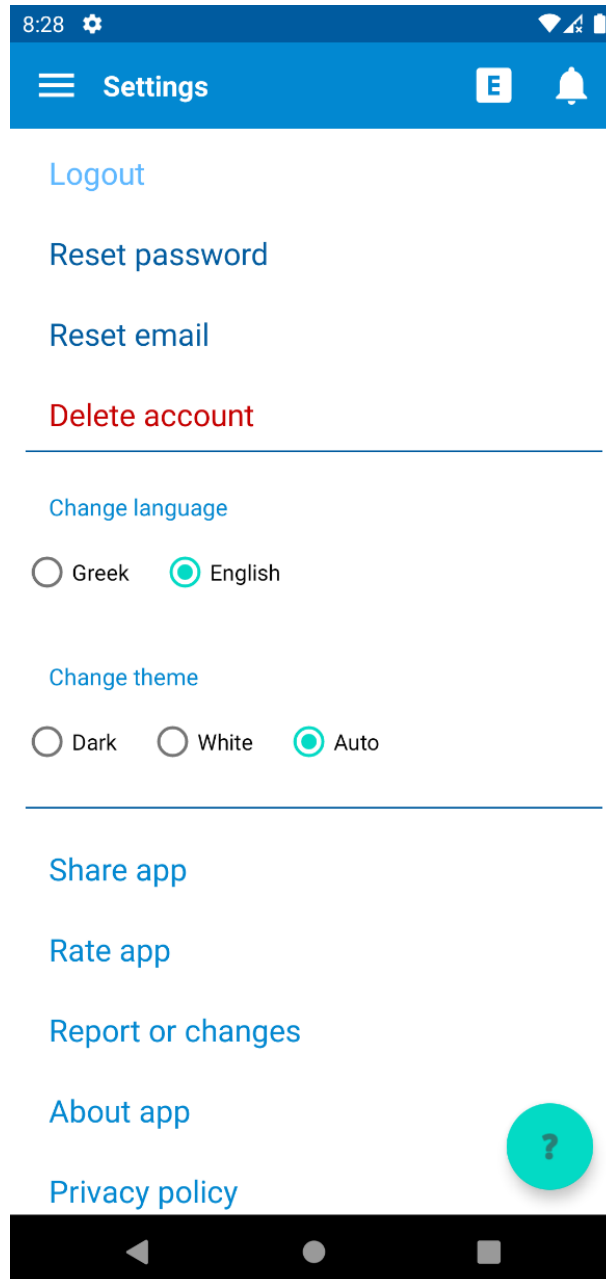
3.8.4 Ρυθμίσεις (Settings)

Στη λειτουργία ρυθμίσεις (Settings) όπου φαίνεται στο Σχήμα 3.30, ο χρήστης έχει μια πληθώρα επιλογών, οι οποίες χωρίζονται σε τρεις ενότητες. Δεν θα γίνει αναφορά στην πρώτη ενότητα ρυθμίσεων μιας και αναφέρεται ήδη στο υποκεφάλαιο με την αυθεντικοποίηση..

Στην δεύτερη ενότητα υπάρχουν δυο επιλογές. Πρώτη επιλογή είναι η αλλαγή γλώσσας μεταξύ ελληνικών και αγγλικών μέσω radio buttons. Όταν επιλεγθεί γίνεται ανανέωση της σελίδας και πάει στο home με την νέα γλώσσα. Κάθε φορά όταν φορτώνει την σελίδα των ρυθμίσεων, έχει επιλεγμένη τη γλώσσα που είχε επιλεγεί τελευταία φορά. Η δεύτερη επιλογή είναι η αλλαγή θέματος μεταξύ φωτεινού, σκοτεινού και αυτόματου από το σύστημα, πάλι μέσω radio buttons. Με την διαφορά ότι όταν επιλέγει κάποιο θέμα ο χρήστης, κάνει ανανέωση σελίδας αλλά χωρίς να πάει κάπου αλλού. Επίσης και εδώ κάθε φορά έχει φορτωμένη την τελευταία επιλογή.

Στην τρίτη ενότητα, έχει τις εξής επιλογές:

- Κοινοποίηση εφαρμογής.
- Βαθμολόγηση της στο Google Play.
- Να γίνει κάποια αναφορά ή πρόταση για την εφαρμογή. Ανοίγοντας ένα μενού και συμπληρώνετε τίτλο και περιεχόμενο και μόλις πατηθεί το κουμπί για να στείλει πάει στο email με όλα συμπληρωμένα.
- Προβολή πληροφοριών για την εφαρμογή.
- Προβολή του πολιτικού απορρήτου της εφαρμογής.

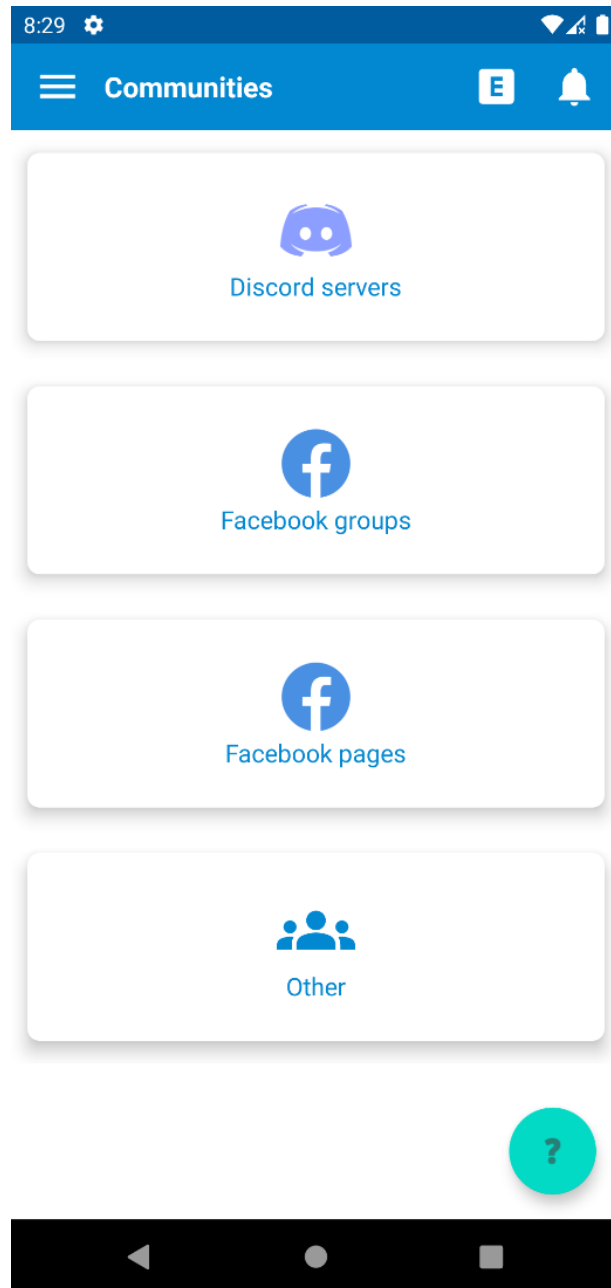


Σχήμα 3.30: Η σελίδα των ρυθμίσεων.

3.8.5 Κοινότητες (Communities)

Στη λειτουργία κοινοτήτες (Communities) όπου φαίνεται στο Σχήμα 3.31, ο χρήστης μπορεί να δει τις κοινότητες για το τμήμα και τους φοιτητές. Υπάρχουν τέσσερις επιλογές. Η πρώτη είναι για τους Discord Servers, η δεύτερη είναι για Facebook Groups, η τρίτη είναι για Facebook Pages και η τέταρτη είναι η Others για όλα τα υπόλοιπα κοινωνικά δίκτυα όπου μπορεί να υπάρχουν ομάδες για το τμήμα. Όταν μπει μέσα σε ένα από αυτά ο χρήστης θα δει ένα recyclerView με cards και πληροφορίες για το καθένα. Αν πατήσει σε κάποιο card πάνω θα τον στείλει μέσω συνδέσμου στο ανάλογο ιστότοπο.

Οι κοινότητες, είναι ένας πίνακας στην απομακρυσμένη βάση SQL όπως και ένας πίνακας στην τοπική βάση Room. Συγχρονίζονται κάθε φορά που θα μπει ο χρήστης μέσα στις κοινότητες, μέσα από το μενού. Κάνει ένα αίτημα για να πάρει ότι έχει ο πίνακας και αν έχει αλλαγές τις προσθέτει. Σε περίπτωση που υπάρχει κάποιο θέμα με την απομακρυσμένη βάση ή απλά καθυστερεί το αίτημα, δεν γίνεται κάτι και απλά παραμένει ο χρήστης με ότι έχει τοπικά η βάση της Room. Κάθε επιλογή στο μενού, πχ Facebook groups, Facebook pages κλπ., είναι ένα διαφορετικό query στην τοπική βάση της Room. Έτσι δεν χρειάζεται να κάνει άλλα αιτήματα στην απομακρυσμένη βάση αλλά μόνο μια φορά για αν υπάρχει κάτι νέο και έτσι γίνονται πιο γρήγορα τα queries.



Σχήμα 3.31: Η σελίδα των κοινοτήτων.

3.8.6 Βρείτε (Find)

Στη λειτουργία βρείτε (Find) όπως φαίνεται στο Σχήμα 3.32, είναι μια από τις μεγαλύτερες λειτουργίες της εφαρμογής μιας και έχει αρκετές επιλογές για εύρεση πληροφοριών. Υπάρχουν συγκεκριμένα 9 επιλογές.

- Η πρώτη που είναι για τα Εξάμηνα (Semesters). Σε αυτή την επιλογή δίνονται τα 10 εξάμηνα που μπορεί να δει ο χρήστης πατώντας στο καθένα. Μέσα από το καθένα μπορεί να δει τα αντίστοιχα μαθήματα που έχει. Αν πατηθεί κάποιο μάθημα, θα ανοίξει τη σελίδα του τμήματος στο συγκεκριμένο μάθημα και με την προσθήκη JavaScript, θα εμφανίσει συγκεκριμένα μόνο το κομμάτι του μαθήματος χωρίς όλα τα περιττά. Επιπλέον, ανάλογα με

την προεπιλεγμένη γλώσσα που είναι η εφαρμογή από τον χρήστη, φορτώνει τον σύνδεσμο του μαθήματος στην ιστοσελίδα στην συγκεκριμένη γλώσσα προτίμησης του χρήστη.

- Η δεύτερη που είναι οι Αίθουσες (Rooms). Σε αυτή την επιλογή όταν πατηθεί δίνονται δυο επιλογές τα Zoom Rooms και τα Rooms. Τα Zoom Rooms έχουν υλοποιηθεί με την βιβλιοθήκη ViewPager2, όπου περιέχει δυο tabs το “R1-R8” και το “E1-E8” και μπορεί ο χρήστης, είτε πατώντας τα είτε με swar αριστερά ή δεξιά να πλοηγηθεί και έχει recyclerView με cards, που δείχνει τις αίθουσες και αν πατήσει πάνω θα πάει απευθείας στο zoom μέσω του συνδέσμου. Όσο αφορά τα Rooms, είναι οι φυσικές αίθουσες της σχολής και υλοποιήθηκαν με ακριβώς τον ίδιο τρόπο, με την διαφορά ότι δεν έχουν δυνατότητα να πατηθούν για να πάνε τον χρήστη κάπου. Οι δυο κατηγορίες όπου χωρίζονται είναι το Κτήριο Π. και το Κτήριο Η..
- Υπάρχουν άλλες 7 επιλογές που βρίσκετε στην ίδια λειτουργική λογική.
 - Τα Μαθήματα (Courses),
 - Επίσημες Υπηρεσίες (Official Services),
 - Ανεπίσημες Υπηρεσίες (Unofficial Services),
 - Χρήσιμες Ιστοσελίδες (Useful Websites),
 - Πρόσφορες (Offers),
 - Καθηγητές (Teachers),
 - Χάρτες (Maps).

Όλες οι κατηγορίες έχουν σύστημα search που φτιάχνει query στην τοπική βάση για να μπορεί να ψάχνει σύμφωνα με ότι θέλει. Από κάτω έχει ένα recyclerView και cards για τα αποτελέσματα. Στους Χάρτες (Maps) είναι η μόνη διαφορά που αντί για recyclerView με cards έχει έναν Google Map με marks. Επιπλέον για να μπορεί ο χρήστης να χρησιμοποιήσει την υπηρεσία αυτή, θα πρέπει να δεχτεί τα δικαιώματα που θα ζητήσει η εφαρμογή, όπως έκανε και στην σύνδεση την πρώτη φορά. Αν δεν τα δεχτεί δεν μπορεί να χρησιμοποιήσει την υπηρεσία εμφανίζοντας επανειλημμένα μήνυμα ρωτώντας συνεχώς μέχρι να δεχτεί τα δικαιώματα ώστε να χρησιμοποιήσει την υπηρεσία. Θα αναφερθεί τι μπορεί να δει ο χρήστης στην κάθε μια επιλογή ξεχωριστά από τις 7, ώστε να φανεί τι κερδίζει από αυτές.

- Μαθήματα:
 - Ποια μαθήματα υπάρχουν.
 - Σε ποιο εξάμηνο είναι το μάθημα.
 - Τι μάθημα έχει ένα εξάμηνο.
 - Ποιοι καθηγητές έχουν το κάθε μάθημα.
 - Ποια μαθήματα διδάσκει ένας καθηγητής.
- Επίσημες Υπηρεσίες:
 - Ποιες επίσημες υπηρεσίες υπάρχουν.
 - Ποιος είναι ο σύνδεσμος της υπηρεσίας.
 - Αν υπάρχει κάποια επίσημη υπηρεσία για κάτι που θα αναζητηθεί.
- Ανεπίσημες Υπηρεσίες:
 - Ποιες ανεπίσημες υπηρεσίες υπάρχουν.
 - Ποιος είναι ο σύνδεσμος της εκάστοτε υπηρεσίας.
 - Ποιος την δημιούργησε κάποια υπηρεσία.

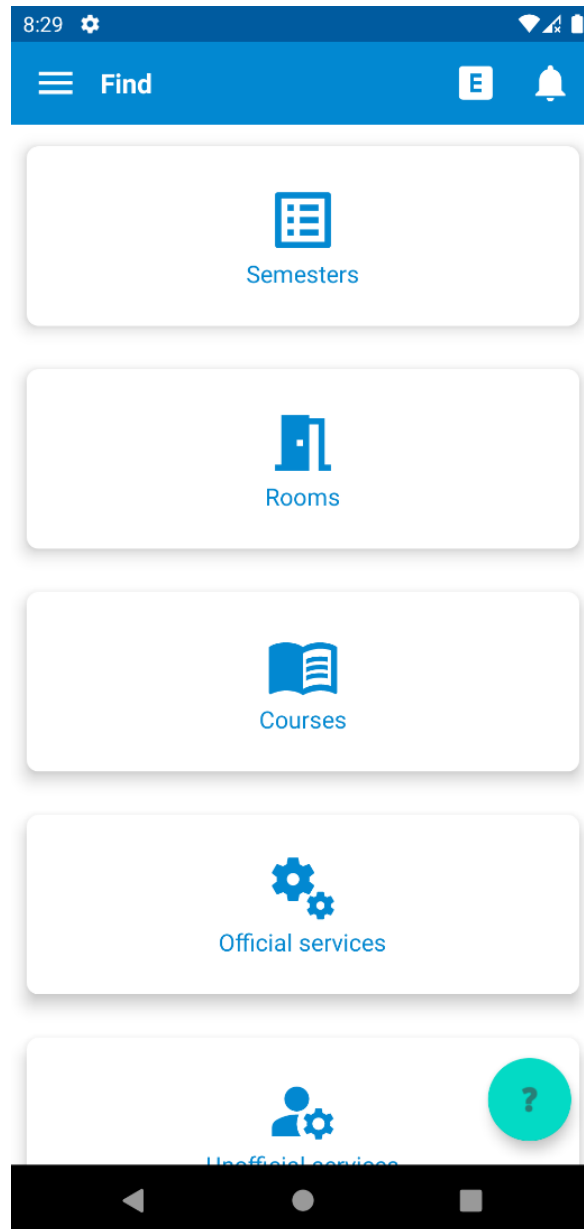
- Ποιες εφαρμογές έχει δημιουργήσει κάποιος.
- Αν υπάρχει κάποια ανεπίσημη υπηρεσία για κάτι που θα αναζητηθεί.

- Χρήσιμες Ιστοσελίδες:
 - Ποιες χρήσιμες ιστοσελίδες υπάρχουν.
 - Ποιος είναι ο σύνδεσμος της ιστοσελίδας.
 - Αν υπάρχει κάποια χρήσιμη ιστοσελίδα για κάτι που θα αναζητηθεί.
 - Ποιες χρήσιμες ιστοσελίδες υπάρχουν για την συγκεκριμένη κατηγορία.

- Προσφορές:
 - Ποιες προσφορές υπάρχουν.
 - Ποιος είναι ο σύνδεσμος της προσφοράς.
 - Αν υπάρχει κάποια προσφορά για κάτι που θα αναζητηθεί.
 - Ποιες πρόσφορες υπάρχουν για την συγκεκριμένη κατηγορία.

- Καθηγητές:
 - Ποιοι καθηγητές υπάρχουν.
 - Ποιο είναι το email του καθηγητή.
 - Ποια είναι η προσωπική του ιστοσελίδα.
 - Τι είδους καθηγητής είναι.
 - Ποιοι καθηγητές υπάρχουν σε αυτή την κατηγορία καθηγητή.

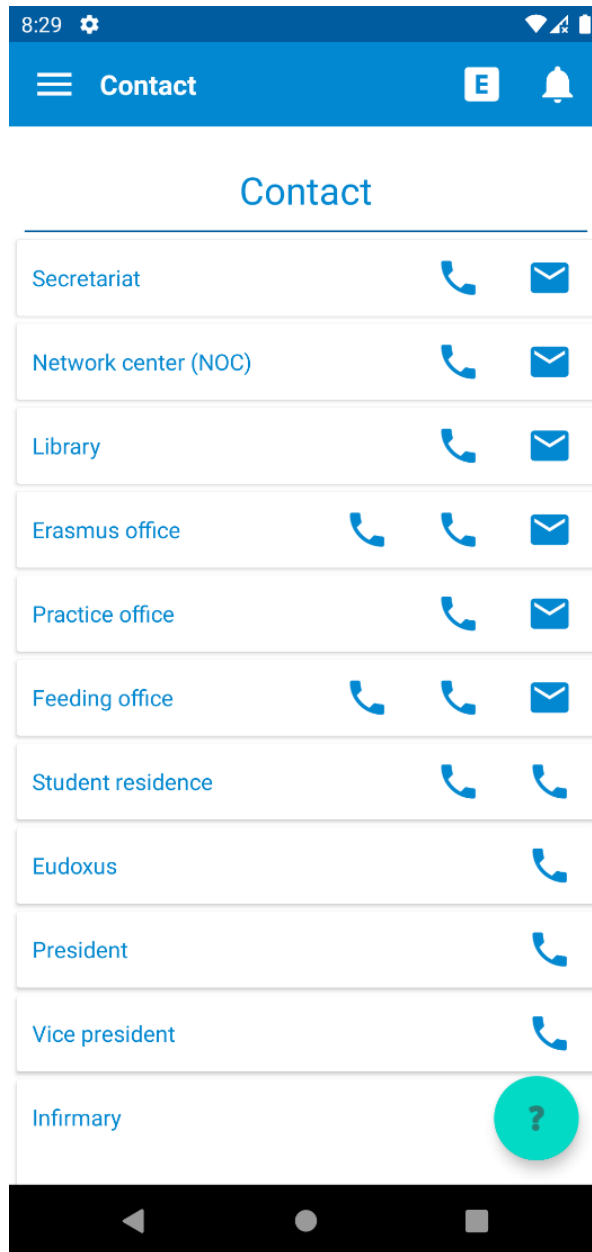
- Χάρτες:
 - Ποια μέρη υπάρχουν.
 - Που βρίσκετε ένα μέρος.
 - Στοιχεία για αυτό το μέρος.



Σχήμα 3.32: Η σελίδα βρες.

3.8.7 Κατάλογος Επικοινωνίας (Contact)

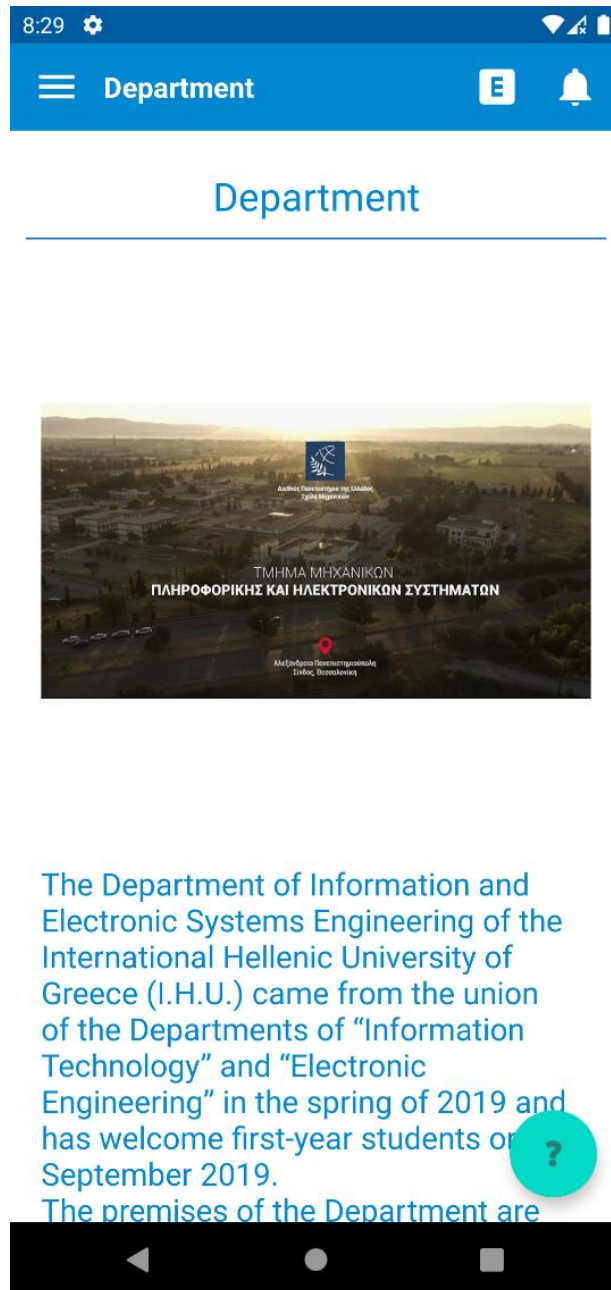
Στη λειτουργία κατάλογος επικοινωνίας (Contact) όπου φαίνεται στο Σχήμα 3.33, ο χρήστης μπορεί να δει μαζεμένα όλα τα τηλέφωνα και τα email όπου θα χρειαστεί στην σχολή όπως π.χ. της γραμματείας. Έτσι ώστε να είναι πολύ ευκολότερο να μαζέψει τα στοιχεία που αναζητά, χωρίς να χρειαστεί να το ψάξει κάπου, π.χ. ένα τηλέφωνο ή ένα email. Για να μπορέσει να χρησιμοποιεί αυτή την λειτουργία θα πρέπει πρώτα να αποδεχτεί τα δικαιώματα όπως αναφέρθηκε και προηγουμένως.



Σχήμα 3.33: Η σελίδα του καταλόγου.

3.8.8 Τμήμα (Department)

Στη λειτουργία τμήμα (Department), ο χρήστης μπορεί να δει κάποιες πληροφορίες για το τμήμα. Είναι μια από τις πιο απλές λειτουργίες της εφαρμογής μας και δεν βοηθάει σε κάτι περισσότερο από το να δει και ίσως να μάθει κάποια πράγματα για το τμήμα ο χρήστης.



Σχήμα 3.34: Η σελίδα του τμήματος.

3.8.9 Οδηγοί (Guides)

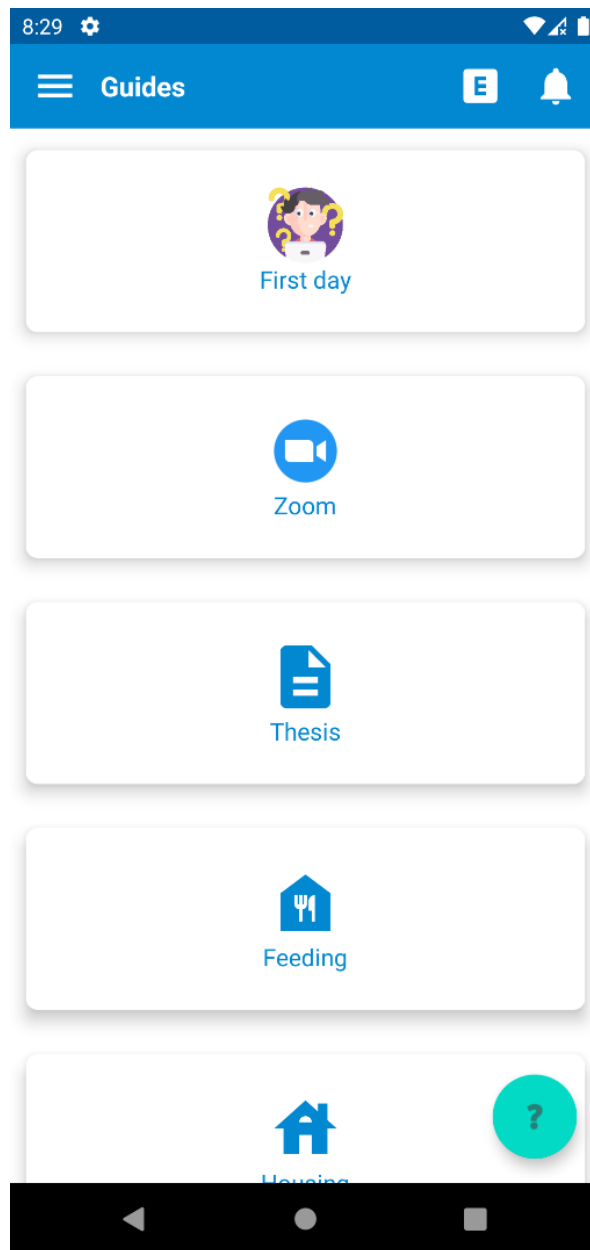
Στην λειτουργία οδηγοί (Guides) όπου φαίνεται στο Σχήμα 3.35, ο χρήστης μπορεί να δει διάφορους πολύ χρήσιμους οδηγούς για την σχολή.

Οι πρώτοι οδηγοί που δημιουργήθηκαν, αν και είναι υπό επεξεργασία ακόμα για την τελειοποίηση τους, είναι οι εξής:

- Πρώτη Ημέρα (First Day)
- Zoom
- Πτυχιακή/Διπλωματική (Thesis)

- Σίτιση (Feeding)
- Στέγαση (Housing)
- Βιβλιοθήκη(Library)
- Λογισμικά (Softwares)

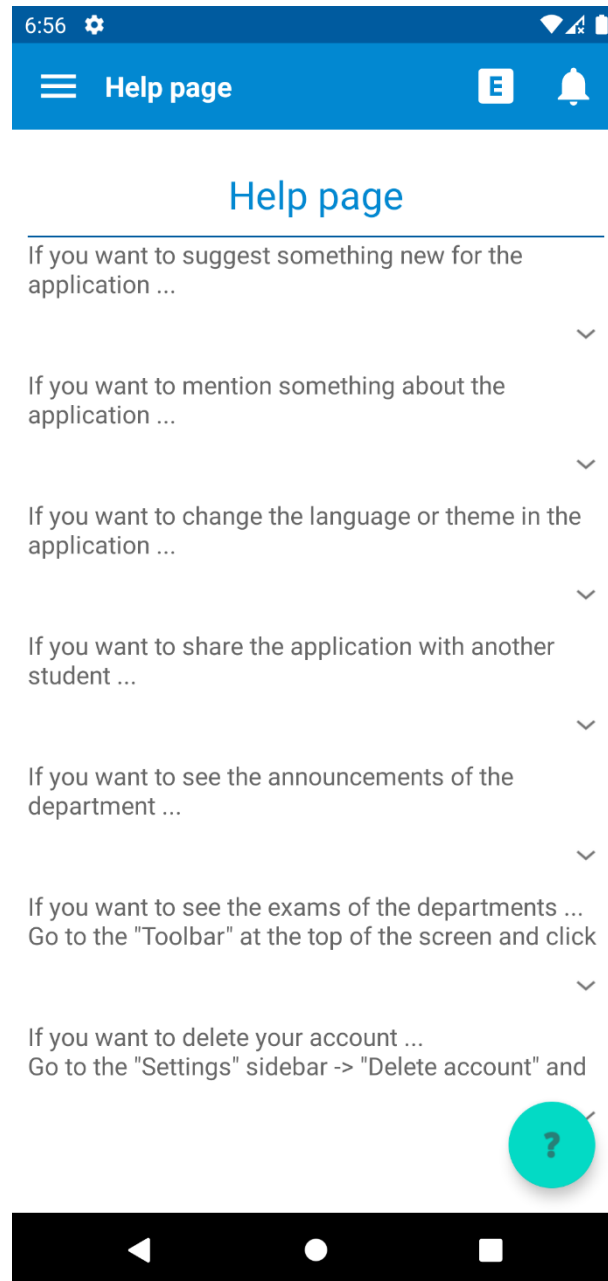
Μια χρήσιμη λειτουργία, που μπορεί να επεκταθεί αρκετά ανάλογα με το τι χρειάζεται κάθε φορά ο χρήστης για το τμήμα, π.χ. να προστεθούν νέοι οδηγοί για νέα προγράμματα.



Σχήμα 3.35: Η σελίδα των οδηγών.

3.8.10 Σελίδα Βοήθειας (Help page)

Στη λειτουργία σελίδα βοήθειας (Help page) όπου φαίνεται στο Σχήμα 3.36, ο χρήστης έχει πρόσβαση μέσω ενός floating button (κάτω δεξιά) καθόλη τη διάρκεια της περιήγησης του, στην εφαρμογή. Πατώντας το κουμπί του ανοίγει μια σελίδα με διάφορες ερωτήσεις και απαντήσεις κάτι σαν βοηθός περιήγησης. Για την υλοποίηση του χρησιμοποιήθηκε μια βιβλιοθήκη η ExpandableTextView. Σκοπός αυτής της σελίδας είναι η βοήθεια των χρηστών σε διάφορα ερωτήματα σχετικά με την εφαρμογή.



Σχήμα 3.36: Η σελίδα βοήθειας.

3.8.11 Κεντρική Σελίδα (Home)

Στη λειτουργία κεντρική σελίδα (Home) όπου φαίνεται στο Σχήμα 3.37, υπάρχει ένα σαν carousel με διάφορα κείμενα, που αναφέρουν διάφορες σημαντικές πληροφορίες για την σχολή και την εφαρμογή. Η υλοποίηση έγινε με την βοήθεια του ViewPager2 και του TabLayout.



Student IEE-IHU

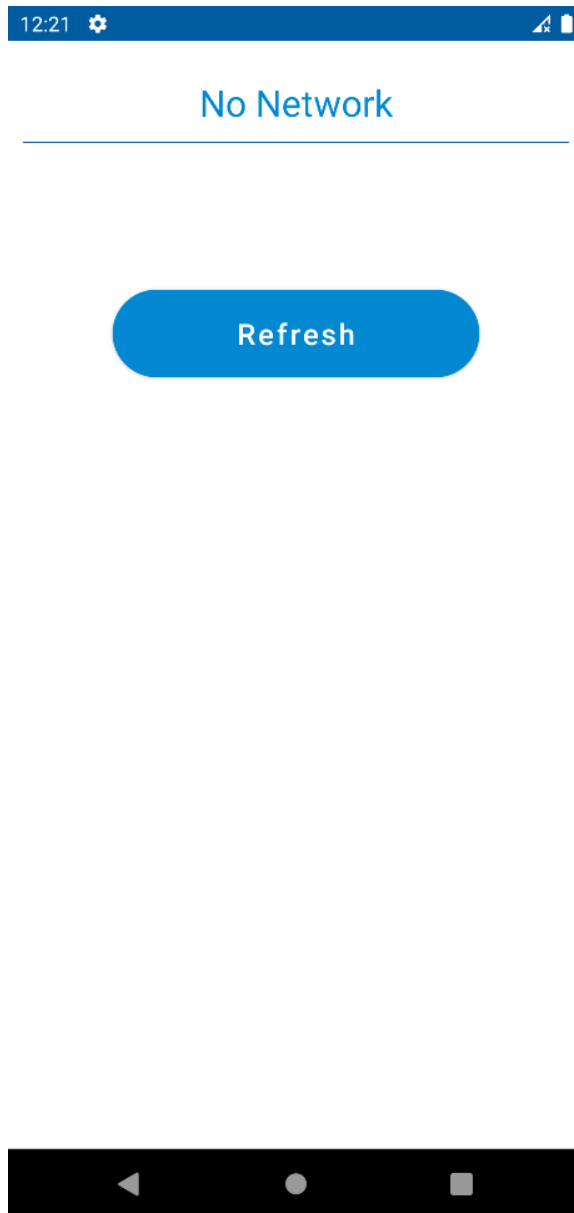


Σχήμα 3.37: Η κεντρική σελίδα.

3.8.12 Έλεγχος Δικτύου

Στη λειτουργία ελέγχου δικτύου όπου φαίνεται στο Σχήμα 3.38, η εφαρμογή έχει σε κάθε Activity ένα έλεγχο. Σε περίπτωση που δεν υπάρχει internet, να στείλει τον χρήστη σε μια σελίδα με ένα κουμπί για refresh, μέχρι να ξανά επανέλθει. Άρα αν κατεβεί το παράθυρο ή γυρίσει η οθόνη ή αλλάξει Activity κλπ., θα στείλει τον χρήστη στην σελίδα του μη έχοντας δίκτυο για να κάνει την

ανανέωση της σελίδας. Τώρα όσο αφορά το πότε και πως κάνει τον έλεγχο, ενδεικτικά μπορεί να αλλάξει ανάλογα με τις απαιτήσεις της εφαρμογής.



Σχήμα 3.38: Η σελίδα εκτός δικτύου.

3.8.13 IEE Apps

Στην λειτουργία IEE Apps, υπάρχει ένα κουμπί (πάνω δεξιά στην μπάρα), εικονίδιο (καμπανάκι). Αν ο χρήστης έχει εγκατεστημένο στην συσκευή του το IEE Apps της σχολής τότε τον στέλνει σε αυτήν αλλιώς το ανοίγει μέσω του φυλλομετρητή. Αυτή η λειτουργία είναι για γρήγορη πρόσβαση και συνεργασία μέσω της παρούσας εφαρμογής.

3.9 Δοκιμές (Testing)

Η εφαρμογή σχεδιάστηκε με την λογική ότι θα έχει διάφορες δοκιμές (testing) αυτοματοποιημένα, πέρα από τις διάφορες δοκιμές που γίνονται σε κανονική χρήση. Αυτός είναι ο λόγος που χρησιμοποιήθηκε και η βιβλιοθήκη Espresso όπως αναφέρθηκε και στο υποκεφάλαιο με τις βιβλιοθήκες. Στο Σχήμα 3.39 φαίνεται μια υλοποιημένη δοκιμή για την είσοδο στην εφαρμογή.

```
@Test
fun login() {
    Thread.sleep( millis: 3000 );
    onView(withId(R.id.login_email)).perform(replaceText( stringToBeSet: "georgekara2010@yahoo.gr" ))
    onView(withId(R.id.login_password)).perform(replaceText( stringToBeSet: "password" ))
    onView(withId(R.id.login_button)).perform(click())
}
```

Σχήμα 3.39: Μια δοκιμή (test) για την είσοδο στην εφαρμογή.

Στο συγκεκριμένο παράδειγμα, όπου είναι για δοκιμή στην είσοδο της εφαρμογής, κάνει τα εξής βήματα:

- Περιμένει 3 δευτερόλεπτά.
- Πηγαίνει και βάζει στο id “login_email” το συγκεκριμένο κείμενο.
- Πηγαίνει και βάζει στο id “login_password” το συγκεκριμένο κείμενο.
- Και τέλος στο id “login_button” κάνει κλικ.

3.10 Οδηγίες χρήσης

Σε αυτό το υποκεφάλαιο θα αναφερθούν οι οδηγίες χρήσης για έναν νέο χρήστη στην εφαρμογή.

- 1) Κατεβάζει την εφαρμογή από το Google Play ή σε όποιο άλλο αποθετήριο είναι αποθηκευμένη και διαθέσιμη η εφαρμογή.
- 2) Όταν η εφαρμογή ανοίξει στην σελίδα σύνδεσης, πατάει για δημιουργία λογαριασμού.
- 3) Συμπληρώνει όλα τα στοιχεία που του ζητάει και πατάει το κουμπί δημιουργίας.
- 4) Πηγαίνει στο email όπου καταχώρησε και πατάει το σύνδεσμο για επαλήθευση του email του.
- 5) Ξανά μπαίνει στην εφαρμογή, στην σελίδα σύνδεσης και συμπληρώνει τα στοιχεία του και πατάει σύνδεση.
- 6) Δίνει τα δικαιώματα και ξανά πατάει σύνδεση, (Σε περίπτωση που κολλήσει και δεν συνδεθεί, απλά κλείνει και ξανά ανοίγει την εφαρμογή.)
- 7) Έτσι θα έχει συνδεθεί στην εφαρμογή και θα τον εμφανίσει στην κεντρική σελίδα.
- 8) Αν πατήσει στην σελίδα προφίλ μπορεί να αλλάξει τα στοιχεία του ή να προσθέσει προαιρετικά φωτογραφία ή τον αριθμό τηλεφώνου.
- 9) Αν πατήσει στην σελίδα ρυθμίσεων μπορεί να αλλάξει το θέμα της εφαρμογής ή την γλώσσα της εφαρμογής.
- 10) Μπορεί να δει ότι περιέχει η εφαρμογή και σε περίπτωση αποριών μπορεί να πατήσει το κουμπί βοήθειας που υπάρχει πάντα δεξιά κάτω στην οθόνη.
- 11) Σε περίπτωση όπου θέλει να αλλάξει email ή password μπορεί να πάει στην σελίδα των ρυθμίσεων και να κάνει την διαδικασία που θέλει.

- 12) Σε περίπτωση κάποιου προβλήματος ή προτάσεις για αναβάθμιση μπορεί μέσω της σελίδας ρυθμίσεων να στείλει κάποιο email μέσω της ειδικής φόρμας που έχει φτιαχτεί για αυτό το λόγο.

3.11 Επίλογος

Στο κεφάλαιο αυτό παρουσιάστηκε η εφαρμογή Student IEE-IHU, το οποίο αναπτύχθηκε για τους σκοπούς της παρούσας πτυχιακής. Αρχικά αναπτύχθηκε η οργάνωση και τα εργαλεία για την ανάπτυξη της εφαρμογής. Μετά αναφέρθηκε στη Firebase, την απομακρυσμένη βάση δεδομένων SQL, στο τι είναι και πως χρησιμοποιήθηκαν. Αργότερα αναφέρθηκαν όλες οι επιπρόσθετες βιβλιοθήκες που χρησιμοποιήθηκαν και αναφέρθηκαν κάποια πράγματα για αυτές. Στη συνέχεια, αναφέρθηκε για το Ελληνοαγγλικό σύστημα της εφαρμογής. Μετέπειτα αναλύθηκε η δομή της εφαρμογής. Ύστερα οι λειτουργίες της εφαρμογής αναλυτικότερα και οι δοκιμές της εφαρμογής. Τέλος, κάποιες οδηγίες χρήσης που μπορεί να ακολουθήσει κάποιος που θέλει να χρησιμοποιήσει την εφαρμογή για πρώτη φορά. Όλο το κεφάλαιο συνοδεύτηκε με τις κατάλληλες εικόνες για να μπορεί να γίνουν ποιο κατανοητά και όσο το δυνατόν καλύτερη επίδειξη της εφαρμογής.

Κεφάλαιο 4ο: Συμπεράσματα ή/και προτάσεις βελτίωσης

Στην παρούσα πτυχιακή εργασία είδαμε τον τρόπο ανάπτυξης της εφαρμογής Student IEE-IHU. Δόθηκε ιδιαίτερη έμφαση στις τεχνολογίες και βιβλιοθήκες που χρησιμοποιήθηκαν καθώς και τις λειτουργίες που περιλαμβάνει η εφαρμογή. Επιπλέον δόθηκε σημασία και στην ορθότητα της εφαρμογής για όσο το δυνατόν καλύτερη συντήρηση και μελλοντική επέκταση της.

Επόμενες προσθήκες στην εφαρμογή που θα προγραμματιστούν είναι, αρχικά να μεταφερθεί σχεδόν όλο το περιεχόμενο της εφαρμογής, πχ τα κείμενα από την σελίδα βοήθειας, του τμήματος, περιγραφή της εφαρμογής, αίθουσες κλπ., στην απομακρυσμένη βάση ώστε να γίνεται το ίδιο σε όλη την εφαρμογή όπως με τους πίνακες που γίνονται ακολουθώντας πάλι την αρχιτεκτονική MVVM. Έτσι ώστε να είναι όλη η εφαρμογή ποιο εύκολα ανανεώσιμη χωρίς να χρειάζεται ιδιαίτερες αναβαθμίσεις για αλλαγές κειμένων κλπ. Επόμενη προσθήκη όπου θα γίνει, είναι η σελίδα του διαχειριστή, ώστε να κάνει προσθήκη, επεξεργασία ή διαγραφή μέσα από την εφαρμογή, έχει χτιστεί εξαρχής η εφαρμογή πάνω σε αυτή την λογική, δυο χρηστών (απλός και διαχειριστής) ώστε να μην χρειάζεται ολόκληρη ανακατασκευή. Τέλος, σημαντική αναβάθμιση που θα μπορούσε να γίνει είναι κάποια λειτουργία για να προγραμματίζει ο χρήστης το πρόγραμμα της σχολής και μέσω της εφαρμογής.

Εναλλακτικά, η εφαρμογή θα μπορούσε να χρησιμοποιηθεί με αλλαγές στο περιεχόμενο και με κάποιες μικρό προσαρμογές, ώστε να χρησιμοποιηθεί από οποιαδήποτε σχολή, είτε μιλάμε για Λύκειο, ΙΕΚ, Πανεπιστήμιο κλπ. Είναι μια εφαρμογή με φιλοσοφία να βοηθήσει τους χρήστες της να κερδίσουν χρόνο, να έχουν καλύτερη μεταβίβαση στη νέα τους εκπαιδευτική βαθμίδα και κατά επέκταση στην σταδιοδρομία τους.

Συνοψίζοντας, τέτοιου είδους εφαρμογές είναι πολύ χρήσιμες να υπάρχουν, για ευκολότερη μεταβίβαση και σταδιοδρομία στη σχολή και κατά επέκταση ίσως και σε άλλους τομείς. Η εφαρμογή Student IEE-IHU ήρθε σαν ιδέα σε αυτή την φιλοσοφία της επιπρόσθετης οργανωμένης βοήθειας και να προσφέρει ότι περισσότερο μπορεί σε αυτή. Η εφαρμογή δέχεται συνεχώς αναβαθμίσεις, ανάλογα με τις απαιτήσεις της κάθε χρονικής στιγμής και τις νέες φρέσκες ιδέες.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Nimesh Kasun Ekanayake, "Android Operating System", General Sir John Kotelawala Defence University, 2018
- [2] Steve Brachmann, "A Brief History of Google's Android Operating System", 09/2021, Available: <https://www.ipwatchdog.com/2014/11/26/a-brief-history-of-googles-android-operating-system/id=52285>
- [3] Alexis Fruhinsholz, "Android versions comparison", SocialCompare, 09/2021, Available: <https://socialcompare.com/en/comparison/android-versions-comparison>
- [4] Ahamed Shibly, "Android Operating System: Architecture, Security Challenges and Solutions", South Eastern University of Sri Lanka, 2016
- [5] Tutlane, "Android Architecture", Tutlane, 09/2021, Available: <https://www.tutlane.com/tutorial/android/android-architecture>
- [6] Umer Farooq, "Android Operating System Architecture", Virtual University of Pakistan, 2018
- [7] Chris Yackulic, "Prime Advantages of Android Operating System", Android Headlines, 09/2021, Available: <https://www.androidheadlines.com/2018/10/prime-advantages-of-android-operating-system.html>
- [8] Google Developers, "Meet Android Studio", Android for Developers, 09/2021, Available: <https://developer.android.com/studio/intro>
- [9] Xavier Ducrohet, Tor Norbye, Katherine Chou, "Android Studio: An IDE built for Android ", Android Developers Blog, 09/2021, Available: <https://android-developers.googleblog.com/2013/05/android-studio-ide-built-for-android.html>
- [10] Jamal Eason, "Android Studio 1.0", Android Developers Blog, 09/2021, Available: <https://android-developers.googleblog.com/2014/12/android-studio-10.html>
- [11] Amanda Alexander, "Android Studio Arctic Fox (2020.3.1) Stable", Android Developers Blog, 09/2021, Available: <https://android-developers.googleblog.com/2021/07/android-studio-arctic-fox-202031-stable.html>
- [12] Chet Haase, "Google I/O 2019: Empowering developers to build the best experiences on Android + Play ", Android Developers Blog, 09/2021, Available: <https://android-developers.googleblog.com/2019/05/google-io-2019-empowering-developers-to-build-experiences-on-Android-Play.html>
- [13] Mehul Rajput, "Why Android Studio Is Better for Android Developers Instead of Eclipse", DZone, 09/2021, Available: <https://dzone.com/articles/why-android-studio-better>
- [14] Hussain Fakhruddin, "Eclipse vs Android Studio: Which IDE Is Better for Android Developers?", Teks mobile, 09/2021, Available: <https://teks.co.in/site/blog/eclipse-vs-android-studio-ide-better-android-developers/>
- [15] DEVOPEDIA, "Kotlin (Language)", DEVOPEDIA, 09/2021, Available: <https://devopedia.org/kotlin-language>
- [16] Swati Saxena, Kotlin at a Glance, BPB Publications, 2019

- [17] Kotlinlang, "What's new", JetBrains, 09/2021, Available: <https://kotlinlang.org/docs/whatsnew11.html>
- [18] Amit Raja Naik, "Ten Years of Kotlin Programming Language", Analytics India Magazine, 09/2021, Available: <https://analyticsindiamag.com/ten-years-of-kotlin-programming-language/>
- [19] Daniela Gotseva, Yavor Tomov, Petko Danov, "Comparative study Java vs Kotlin", Technical University of Sofia, 2019
- [20] Vijay Singh, "Kotlin vs Java: Important Differences That You Must Know", hackr.io, 09/2021, Available: <https://hackr.io/blog/kotlin-vs-java>
- [21] Mariana Berga, Rute Figueiredo, "Kotlin vs Java: the 12 differences you should know", IMAGINARY CLOUD, 09/2021, Available: <https://www.imaginarycloud.com/blog/kotlin-vs-java/>
- [22] Rachaelle Lynn, "What Is a Kanban Board?", planview, 08/2021, Available: <https://www.planview.com/resources/guide/introduction-to-kanban/what-is-kanban-board/>
- [23] Devang Jagdale, "Firebase – Introduction", GeeksforGeeks, 08/2021, Available: <https://www.geeksforgeeks.org/firebase-introduction/>
- [24] Firebase, "Firebase Pricing", Google, 08/2021, Available: <https://firebase.google.com/pricing?authuser=0>
- [25] Firebase, "Firebase Authentication", Google, 08/2021, Available: <https://firebase.google.com/docs/auth?authuser=0>
- [26] Firebase, "Firebase Realtime Database", Google, 08/2021, Available: <https://firebase.google.com/docs/database?authuser=0>
- [27] Firebase, "Cloud Storage for Firebase", Google, 8/2021, Available: <https://firebase.google.com/docs/storage?authuser=0>
- [28] Glide, "About Glide", GitHub, 08/2021, Available: <https://bumptech.github.io/glide/>
- [29] Glide Transformations, "Glide Transformations", GitHub, 08/2021, Available: <https://github.com/wasabeef/glide-transformations>
- [30] MotionToast, "Motion Toast - A Beautiful Toast Library for Android Kotlin", GitHub, 08/2021, Available: <https://github.com/Spikeysanju/MotionToast>
- [31] Google Developers, "Navigation", Android for Developers, 08/2021, Available: <https://developer.android.com/guide/navigation>
- [32] FirebaseUI, "FirebaseUI for Android — UI Bindings for Firebase", GitHub, 08/2021, Available: <https://github.com/firebase/FirebaseUI-Android>
- [33] Google Developers, "Maps SDK for Android overview", Android for Developers, 08/2021, Available: <https://developers.google.com/maps/documentation/android-sdk/overview>
- [34] EasyPermissions-ktx, "EasyPermissions-ktx", GitHub, 08/2021, Available: <https://github.com/VMadalin/easypermissions-ktx>
- [35] MaterialStyledDialogs, "MaterialStyledDialogs", GitHub, 08/2021, Available: <https://github.com/javiersantos/MaterialStyledDialogs>
- [36] Material Dialogs, "Material Dialogs Input", GitHub, 08/2021, Available: <https://github.com/afollestad/material-dialogs>

- [37] MMKV, "MMKV for Android", GitHub, 08/2021, Available: <https://github.com/Tencent/MMKV>
- [38] Google Developers, "Kotlin coroutines on Android", Android for Developers, 08/2021, Available: <https://developer.android.com/kotlin/coroutines>
- [39] Google Developers, "Save data in a local database using Room", Android for Developers, 08/2021, Available: <https://developer.android.com/training/data-storage/room>
- [40] Denis Buketa, "Android Lifecycle", raywenderlich.com, 08/2021, Available: <https://www.raywenderlich.com/21382977-android-lifecycle>
- [41] Google Developers, "Test UI for a single app", Android for Developers, 08/2021, Available: <https://developer.android.com/training/testing/ui-testing/espresso-testing>
- [42] Retrofit, "Retrofit", GitHub, 08/2021, Available: <https://square.github.io/retrofit/>
- [43] ExpandableTextView, "ExpandableTextView", GitHub, 08/2021, Available: <https://github.com/Manabu-GT/ExpandableTextView>
- [44] Rajdeep Singh, "ViewPager2 in Android: Getting Started", raywenderlich.com, 08/2021, Available: <https://www.raywenderlich.com/8192680-viewpager2-in-android-getting-started>
- [45] Google Developers, "Create dynamic lists with RecyclerView", Android for Developers, 08/2021, Available: <https://developer.android.com/guide/topics/ui/layout/recyclerview>
- [46] Google Developers, "Create a Card-Based Layout", Android for Developers, 08/2021, Available: <https://developer.android.com/guide/topics/ui/layout/cardview>
- [47] Microsoft Doc, "The MVVM Pattern", Microsoft, 08/2021, Available: [https://docs.microsoft.com/en-us/previous-versions/msp-n-p/hh848246\(v=pandp.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/hh848246(v=pandp.10)?redirectedfrom=MSDN)
- [48] Amit Shekhar, "MVVM Architecture - Android Tutorial for Beginners - Step by Step Guide", MindOrks, 08/2021, Available: <https://blog.mindorks.com/mvvm-architecture-android-tutorial-for-beginners-step-by-step-guide>

ΠΑΡΑΡΤΗΜΑ Α : Ερωτηματολόγιο

Το ερωτηματολόγιο που πραγματοποιήθηκε στα πλαίσια της πτυχιακής μέσω Google Forms, είχε τις εξής ερωτήσεις, σε τέσσερις κατηγορίες:

- Γενικά στοιχεία
 - Φύλο (Ανδρας, Γυναίκα, Άλλο).
 - Τόπος φοίτησης (Θεσσαλονίκη, Εκτός Θεσσαλονίκης).
 - Ηλικία (18-24, 25+).
 - Έτος σπουδών (1^ο, 2^ο, 3^ο, 4^ο, 5^ο).
 - Εισαχθείς τμήματος (Πληροφορικής, Ηλεκτρονικής, Νέου Τμήματος).
- Χρήση
 - Πρόσβαση στην εφαρμογή από τη συσκευή.
- Παρατηρήσεις
 - Πληροφορίες που βρίσκετε δύσκολα στην ιστοσελίδα, είτε στο IEE Apps του τμήματος.
 - Τι δυσκολίες συναντάτε στο τμήμα και πως πιστεύετε, ότι θα μπορούσαν να επιλυθούν μέσω μιας εφαρμογής: (π.χ. δυσκολία να βρω τις zoom αίθουσες - λύση: σελίδα με όλες τις αίθουσες και τα link).
 - Γενικές απορίες που έχετε σχετικά με το τμήμα και δεν έχετε βρει την λύση: (π.χ. πρόγραμμα σπουδών, πτυχιακή κλπ.).
- Λειτουργίες
 - Προτείνετε αναφορές ιστοσελίδων, που θα ήταν χρήσιμες για τους φοιτητές: (π.χ. foititikanea.gr, eudoxus.gr).
 - Προτείνετε βοηθητικές λειτουργίες, που θα θέλατε να υπάρχουν στην εφαρμογή: (π.χ. Εμφάνιση σημείων στο Google Map, δυνατότητα απευθείας κλήσης τηλεφώνου γραμματείας, κλπ. μέσω της εφαρμογής).

Υπήρχαν στο σύνολο 56 εθελοντές που απάντησαν στο συγκεκριμένο ερωτηματολόγιο.

Πρώτα γενικά συμπεράσματα που έβγαλα από τις απαντήσεις είναι:

1^ο Η πλειοψηφία των εθελοντών είναι άνδρες, βέβαια αυτό εξηγείτε κιόλας από την διαφορά αριθμού ανδρών/γυναικών στο τμήμα.

2^ο Η πλειοψηφία των εθελοντών είναι άτομα που μένουν εντός Θεσσαλονίκης, άρα θα μπορούσαν να προστεθούν διάφοροι προορισμοί και προσφορές για την Θεσσαλονίκη στην εφαρμογή.

3^ο Η πλειοψηφία των εθελοντών είναι άτομα ηλικίας 18-24 με ένα ποσοστό 10% να είναι 25+.

4^ο Το 65% των εθελοντών βρίσκονται εντός κανονικών αριθμών εξάμηνων, ενώ ένα ποσοστό 35% βρίσκετε στο 5^ο+ εξάμηνο, όπου σημαίνει καθυστέρηση σπουδών για τον οποιοδήποτε λόγο.

5^ο Το 50% των εθελοντών ήταν του τμήματος Πληροφορικής ενώ 25% του τμήματος της Ηλεκτρονικής και 25% του Νέου Τμήματος. Αυτό μπορεί να συμπεράνει και το περισσότερο ενδιαφέρον ή εξοικείωση από άτομα της Πληροφορικής προς τέτοιες υπηρεσίες σαν την εφαρμογή Student IEE-IHU.

6^ο Οι 54 στους 56 εθελοντές έχουν κινητό έκδοσης Android 7+, άρα αν η εφαρμογή καλύβει αυτό το API, έχει καλύψει σχεδόν το μεγαλύτερο ποσοστό χρηστών και αν συμπεριληφθεί σαν παράμετρος ότι η εφαρμογή θα καθυστερήσει κάνα χρόνο για να βγει. Θα έχει ανεβεί και άλλο το ποσοστό εάν δεν γίνει 100% μιας και οι παλιές εκδόσεις σιγά σιγά χάνονται.

7^{ov} Από εκεί και πέρα όσο αφορά όλες τις ερωτήσεις από εδώ και κάτω, μελετήθηκαν προσεχτικά, έγινε διαχωρισμός ποιες μπορούν να υλοποιηθούν στην εφαρμογή και ποιες όχι, όπως και όλα τα παράπονα όπου υπήρχαν προς τις ήδη υπάρχοντες υπηρεσίες. Τέλος έγινε προσπάθεια υλοποίησης των περισσότερων αιτημάτων και με αυτό τον γνώμονα σχεδιάστηκε μεγάλο μέρος της εφαρμογής.