

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη Android Εφαρμογής σε περιβάλλον Android  
Studio»



**StudyBuddy**

**Της φοιτήτριας  
Μιχαηλίδου Παρθένα  
Αρ. Μητρώου: 113752**

**Επιβλέπων  
Ονοματεπώνυμο Ρήγας  
Κωτσάκης**

**Ημερομηνία 31/05/2025**

Τίτλος Δ.Ε. Δημιουργία Android εφαρμογής για την διαχείριση εκπαιδευτικού υλικού ιδιαίτερων μαθημάτων

Κωδικός Δ.Ε. **24315**

Ονοματεπώνυμο φοιτήτριας Παρθένα Μιχαηλίδου.

Ονοματεπώνυμο εισηγητή Ρήγας Κωτσάκης

Ημερομηνία ανάληψης Δ.Ε. 07/11/24

Ημερομηνία περάτωσης Δ.Ε. 31/05/25

*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 της Μιχαηλίδου Παρθένας που την εκπόνησε. 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 οποιονδήποτε 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 απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή του τμήματος Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων ΔΙΠΑΕ κύριο Ρήγα Κωτσάκη για την ευκαιρία που μου έδωσε να υλοποιήσω μια δική μου ιδέα ως θέμα της Π.Ε..

Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου , η οποία αποτελεί το θεμέλιο της πορείας μου. Η αμέριστη στήριξη τους και η πίστη τους ήταν για μένα πολύτιμα εφόδια, ιδίως στις δύσκολες στιγμές που απαιτούν υπομονή , επιμονή και ψυχραιμία.

Τέλος θα ήθελα να ευχαριστήσω τους φίλους που απέκτησα κατά τη διάρκεια των σπουδών μου στο τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων , για τις όμορφες στιγμές και την άψογη συνεργασία που είχαμε όλα αυτά τα χρόνια.

*«Αφιερώνω την παρούσα εργασία σε όλους όσους στάθηκαν δίπλα μου καθόλη τη διάρκεια των σπουδών μου, προσφέροντας μου έμπνευση , ενθάρρυνση και ψυχική δύναμη»*

## Πρόλογος

Η συνεχή εξέλιξη των τεχνολογιών πληροφορικής προσφέρει νέες δυνατότητες στο χώρο της εκπαίδευσης , με τρανταχτό παράδειγμα την περίοδο του Covid που τα μαθήματα γινόντουσαν με τεχνολογικά μέσα . Αν και ήταν μια δύσκολη περίοδος η εκπαίδευση μπόρεσε και εκμεταλλεύτηκε την τεχνολογία για να συνεχίσει το έργο της. Όπως όλοι μας , έτσι και οι καθηγητές έχουν στην διάθεση τους πολλούς τρόπους και εργαλεία για την ψηφιοποίηση , διαχείριση αλλά και παροχή εκπαιδευτικού υλικού διευκολύνοντας την αλληλεπίδραση με τους μαθητές του .

Ένας από τους λόγους που ήθελα να ασχοληθώ με android εφαρμογές είναι κυρίως ότι και εγώ με την σειρά μου στηρίζομαι αρκετά στη κινητή συσκευή μου μέσα στην καθημερινότητα μου . Έτσι πάντα αυτός ο κόσμος μου κέντριζε το ενδιαφέρον. Ύστερα από την ανάγκη που υπήρξε από φιλικό πρόσωπο το οποίο είναι καθηγήτρια Φιλολογικών μαθημάτων σκέφτηκα να δημιουργήσω ένα εργαλείο όπου ο καθηγητής μπορεί να συγκεντρώνει το υλικό και να το διαθέτει στους μαθητές τους.

Ο σκοπός της εργασίας είναι η δημιουργία μιας λειτουργικής , επεκτάσιμης , ασφαλούς αλλά και εύκολη στη χρήση εφαρμογής που θα μπορούσε να ανήκει στον κόσμο της σύγχρονης εκπαίδευσης . Επίσης αποσκοπεί στο να αναδείξει τη σημασία της χρήσης σύγχρονων τεχνολογιών, όπως οι τεχνολογίες κινητής τηλεφωνίας για την ανάπτυξη καινοτόμων εφαρμογών .

## Περίληψη

Η παρούσα πτυχιακή εργασία έχει ως βασικό σκοπό την παρουσίαση αλλά και την ανάπτυξη μιας εφαρμογής Android, η οποία επιτρέπει σε ένα καθηγητή να δημιουργήσει και να διαχειριστεί το εκπαιδευτικό υλικό των μαθημάτων του . Δίνοντας πρόσβαση στους μαθητές του στα αντίστοιχα μαθήματα μπορεί να εκτελέσει το λειτούργημα του με μεγαλύτερη ευκολία και με λιγότερο κόστος καθώς το μόνο που χρειάζεται είναι η χρήση των ήδη υπάρχων ηλεκτρονικών συσκευών που έχει στην κατοχή του . Ακόμα δίνεται η δυνατότητα επικοινωνίας με τους μαθητές του μέσω προσωπικών μηνυμάτων. Επίσης έχει προστεθεί και πίνακας leaderboard για να δίνει ώθηση στους μαθητες για μεγαλύτερη προσπάθεια.

Η εφαρμογή υλοποιήθηκε με προγραμματιστική γλώσσα Kotlin και αξιοποιεί τις υπηρεσίες του Firebase για authentication αλλά και αποθήκευση των απαραίτητων αρχείων . Προσφέρει στους χρήστες της ένα φιλικό περιβάλλον . Υποστηρίζει επεξεργασία υλικού μέσω Fragments και View Binding αλλά και την χρήση activities , πχ για την δημιουργία λογαριασμού.

Η συγκεκριμένη εργασία αναλύει τους βασικούς άξονες της εφαρμογής όπως την η ανάλυση του λογισμικού που χρησιμοποιήθηκε, την ανάλυση των συγκεκριμένων τεχνολογιών που εφαρμόστηκαν στην υλοποίηση της εφαρμογής αλλά και την παρουσίαση του κώδικα και της εφαρμογής.

# «Creation of an Android application for managing educational material for special courses»

«Parthena Michailidou»

## **Abstract**

The main purpose of this thesis is to present and develop an Android application, which allows a teacher to create and manage the educational material of his courses. By giving his students access to the respective courses, he can perform his function with greater ease and at less cost, as all he needs is the use of the existing electronic devices in his possession. He is also given the opportunity to communicate with his students through personal messages. A leaderboard has also been added to motivate students to make greater efforts.

The application was implemented with the Kotlin programming language and utilizes Firebase services for authentication and storage of the necessary files. It offers its users a friendly environment. It supports editing material through Fragments and View Binding as well as the use of activities, e.g. for creating an account.

This specific work analyzes the main axes of the application, such as the analysis of the software used, the analysis of the specific technologies applied in the implementation of the application, and the presentation of the code and the application.

## Thanks

I would like to thank the professor of the Department of Computer Engineering and Electronic Systems, DIPAE, Mr. Rigas Kotsakis, for the opportunity he gave me to implement an idea of my own as a subject for the P.E..

I would also like to thank my family, which is the foundation of my path. Their unwavering support and their faith were valuable resources for me, especially in difficult times that require patience, perseverance and composure.

Finally, I would like to thank the friends I made during my studies at the Department of Computer Engineering and Electronic Systems, for the beautiful moments and the excellent cooperation we had all these years.

*«I dedicate this work to all those who stood by me throughout my studies, offering me inspiration, encouragement and mental strength.»*



2.4Επίλογος.....	21
Κεφάλαιο 3ο:Εργαλεία και ιστοσελίδες που χρησιμοποιήθηκαν	
3.1Εισαγωγή.....	22
3.2Android Studio.....	22
3.3Canva.....	23
3.3.1 Background images.....	23
3.4. Iconik Ai.....	24
3.5 Επίλογος.....	24
Κεφάλαιο 4ο:Αρχιτεκτονική Συστήματος	
4.1Εισαγωγή.....	25
4.2Frontend (Android app).....	25
4.2.1 Kotlin.....	25
4.2.2 Activities, Fragments & Dialogs.....	25
4.2.2.1 Activities.....	26
4.2.2.2 Fragments.....	26
4.2.2.3 Dialogs.....	27
4.2.3 View Binding.....	27
4.3Backend (Firebase).....	28
4.3.1 Firebase Authentication.....	28
4.3.2 Firestore Database.....	29
4.3.3 Firebase Storage.....	29
4.4Δομή Δεδομένων.....	29
4.4.1 Συλλογή: COURSES.....	30
4.4.2 Συλλογή: MATERIALS.....	30
4.4.3 Συλλογή: RECEIVE MESSAGES.....	31
4.4.4 Συλλογή: SENT MESSAGES.....	31
4.4.5 Συλλογή: STUDENTS.....	32
4.4.6 Συλλογή: TEACHERS.....	33
4.4.7 Συλλογή: USERS.....	34
4.4.8 Συλλογή:EXERCISES.....	34
4.5Επίλογος.....	34
Κεφάλαιο 5ο:Δοκιμές	
5.1 Εισαγωγή.....	35
5.2 Σεμινάρια Ελέγχου.....	35

5.3	Αποτελεσματικότητα Δοκιμών.....	35
5.4	Παρατηρήσεις.....	35
5.6	Επίλογος.....	35
Κεφάλαιο 6ο:Παρουσίαση της εφαρμογής StudyBuddy		
6.1	Επίλογος.....	36
6.2	Δημιουργία Εφαρμογής.....	36
6.3	AndroidManifest.....	37
6.4	Βιβλιοθήκες.....	38
6.5	Κλασεις με ui.....	38
6.5.1	Activities.....	39
6.5.1.1	SplashActivity.....	39
6.5.1.2	LogInActivity.....	40
6.5.1.3	SignUpActivity.....	41
6.5.1.4	CreateCourseActivity.....	42
6.5.1.5	Main Activity.....	43
6.5	Fragments.....	45
6.5.2.1	About.....	45
6.5.2.2	CourseConten.....	46
6.5.2.3	LeaderBoard.....	47
6.5.2.4	MyCourses.....	48
6.5.2.5	MyGrades.....	49
6.5.2.6	MyMessages.....	50
6.5.2.7	MyStudents.....	51
6.5.2.8	MyTeachers.....	52
6.5.2.9	Profile.....	53
6.5.2.10	Students.....	54
6.5.2.11	TeacherGradeStudent.....	55
6.6	Models package.....	57
6.7	Firebase package.....	58
6.8	Επίλογος.....	58
Κεφάλαιο 7ο:Συμπεράσματα και Μελλοντική Εργασία.....		
		65
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		
		66
ΠΑΡΑΡΤΗΜΑ Α.....		
		67
ΠΑΡΑΡΤΗΜΑ Β : Αποτελέσματα Ερευνας.....		
		68

## Κατάλογος Σχημάτων

Σχήμα 1.1: 1ο Λογότυπο Android.....	15
Σχήμα 1.2: 2ο Λογότυπο Android.....	15
Σχήμα 1.3: 3ο Λογότυπο Android.....	16
Σχήμα 1.4: 4ο Λογότυπο Android.....	16
Σχήμα 1.5: 5ο Λογότυπο Android.....	16
Σχήμα 1.6: Έκδοση Froyo.....	18
Σχήμα 1.7: Έκδοση Lollipop.....	18
Σχήμα 1.8: Έκδοση Nougat.....	18
Σχήμα 1.9: Έκδοση Oreo.....	18
Σχήμα 1.10: Έκδοση Pie.....	19
Σχήμα 3.1: Λογότυπο Android Studio.....	22
Σχήμα 3.2: Λογότυπο Canva.....	23
Σχήμα 3.3: log/Sign background.....	23
Σχήμα 3.4: general background.....	23
Σχήμα 3.5: Λογότυπο SB.....	24
Σχήμα 4.1: Λογότυπο Kotlin.....	25
Σχήμα 4.2: Activities.....	26
Σχήμα 4.3: Fragments.....	27
Σχήμα 4.4: Dialogs.....	27
Σχήμα 4.5: activity view binding.....	28
Σχήμα 4.6: fragment view binding.....	28
Σχήμα 4.7: firebase logo.....	29
Σχήμα 4.8: Firebase Authentication.....	29
Σχήμα 4.9: Firebase Collections.....	30
Σχήμα 4.10: Course Collection.....	31
Σχήμα 4.11: Material Collection.....	31
Σχήμα 4.12: Receive Message Collection.....	32
Σχήμα 4.13: Receive Message Detail Sub Collection.....	32
Σχήμα 4.14: Sent Message Collection.....	32
Σχήμα 4.15: Sent Message Detail Sub Collection.....	33
Σχήμα 4.16: Student Collection.....	33
Σχήμα 4.17: Course Grade Sub Collection.....	34
Σχήμα 4.18: Teacher Collection.....	34
Σχήμα 4.19: User Collection.....	35
Σχήμα 4.20: Exercise Collection.....	35
Σχήμα 6.1: Λειτουργίες συστήματος.....	37
Σχήμα 6.2: Άδειες εφαρμογής.....	37
Σχήμα 6.3: Plugins.....	39
Σχήμα 6.4: Ενεργοποίηση view binding.....	39
Σχήμα 6.5: Dependencies.....	40
Σχήμα 6.6: Packages.....	40
Σχήμα 6.7: SplashScreen.....	41
Σχήμα 6.8: LogIn.....	42
Σχήμα 6.9: SignUp.....	43

Σχήμα 6.10: CreateCourse.....	44
Σχήμα 6.11:TeacherMenu.....	45
Σχήμα 6.12: StudentMenu.....	45
Σχήμα 6.13: About.....	46
Σχήμα 6.14: CourseContentTeacher.....	47
Σχήμα 6.15: CourseContentStudent.....	47
Σχήμα 6.16: editCourseDialog.....	48
Σχήμα 6.17: Leaderboard.....	48
Σχήμα 6.18:My coursesTeacher.....	49
Σχήμα 6.19:My coursesStudent.....	49
Σχήμα 6.20:MyGrades.....	50
Σχήμα 6.21:ReceiveMessage.....	51
Σχήμα 6.22:SentMessages.....	51
Σχήμα 6.23:Mystudents.....	52
Σχήμα 6.24:MyTeachers.....	53
Σχήμα 6.25:SentMessage.....	53
Σχήμα 6.26:Profil.....	54
Σχήμα 6.27:ChangeRole.....	54
Σχήμα 6.28:Students.....	55
Σχήμα 6.29:AddStudent.....	55
Σχήμα 6.30:StudentGrade.....	56
Σχήμα 6.31:changeGrade.....	56
Σχήμα 6.32:Data classes.....	57
Σχήμα 6.33:User Methods.....	58
Σχήμα 6.34:Course Methods.....	59
Σχήμα 6.36:Material Methods.....	60
Σχήμα 6.36:Exercise Methods.....	61
Σχήμα 6.37:Teacher Methods.....	62
Σχήμα 6.38:Student Methods.....	63
Σχήμα 6.39:Sent Message Methods.....	64
Σχήμα 6.40:Receive Message Methods.....	64
Σχήμα A.1: getCurrentUserId().....	67
Σχήμα A.2: loadUserData().....	67

## Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
SB	Study Buddy
UI	Διεπαφής Χρήστη

## Κεφάλαιο 1ο: Ιστορικό Υπόβαθρο

### 1.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα μιλήσουμε για τη γλώσσα προγραμματισμού Android . Θα αναφερθεί η πορεία της καθώς και η εξέλιξη της.

### 1.2 Τι είναι το Android

Η λέξη "Android" αναφέρεται συγκεκριμένα σε ένα λειτουργικό σύστημα για κινητά τηλέφωνα και όχι μόνο (βασισμένο στο Linux) το οποίο αναπτύσσεται από την Google. Είναι λογισμικό ανοιχτού κώδικα, το οποίο σημαίνει ότι οποιοσδήποτε μπορεί να πάρει τον πηγαίο κώδικα και να τον χρησιμοποιήσει και να τον παραμετροποιήσει. [2] Καθώς είναι ελεύθερα διαθέσιμο, κατασκευαστές hardware όπως η HTC, η Motorola και η Samsung παίρνουν τον κώδικα και το χρησιμοποιούν ως βάση για να χτίσουν πάνω του τις δικές τους υλοποιήσεις που τρέχουν στα κινητά τους τηλέφωνα. Εκτός από τα smartphone μπορεί να χρησιμοποιηθεί σε tablet , σε συσκευές που υποστηρίζονται Linux από κάμερες, κονσόλες παιχνιδιών και τηλεοράσεις μέχρι συστήματα αυτοκινήτων, ρολόγια και γυαλιά.

### 1.3 Η εξέλιξη του Android

Η Android Inc. ιδρύθηκε στο Palo Alto της Καλιφόρνια τον Οκτώβριο του 2003, πέντε χρόνια πριν κυκλοφορήσει το πρώτο smartphone Android. Ιδρύθηκε από τους Andy Rubin, Rich Miner, Nick Sears και Chris White. Ο Andy Rubin πριν ιδρύσει την Android Inc. και γίνει μέλος της Google, εργάστηκε στην Apple από το 1989 έως το 1992. όπου και του δόθηκε το παρατσούκλι “Android” λόγω της αγάπης του για τα ρομπότ. Στην πραγματικότητα, το Android.com ήταν ο προσωπικός ιστότοπος του Rubin μέχρι το 2008. Αρχικά, το Android προοριζόταν να είναι ένα λειτουργικό σύστημα για ψηφιακές φωτογραφικές μηχανές. Η εταιρεία αργότερα αποφάσισε τότε ότι η αγορά των φωτογραφικών μηχανών δεν ήταν αρκετά μεγάλη για τους στόχους της, και πέντε μήνες αργότερα έστρεψε τις προσπάθειές της και ανέβαζε το Android ως λειτουργικό σύστημα χειρός που θα ανταγωνιζόταν τα Symbian και τα Microsoft Windows Mobile. [3]

Τον Ιούλιο του 2005, η Google απέκτησε το Android Inc. για τουλάχιστον 50 εκατομμύρια δολάρια. Η πρώτη επίσημη έκδοση του Android παρουσιάστηκε το 2008 με το HTC Dream (γνωστό και ως T-Mobile G1), το πρώτο smartphone που βασίστηκε στο νέο λειτουργικό σύστημα.

Η πρώτη έκδοση του Android ονομάστηκε “Cupcake” και έπειτα συνέχισε με ονομασίες γλυκών όπως “Donut” και “Eclair” . Κατά την ανακοίνωση του Android “KitKat” το 2013, η Google εξήγησε ότι “Δεδομένου ότι αυτές οι συσκευές κάνουν τη ζωή μας τόσο γλυκιά, κάθε έκδοση Android ονομάζεται με βάση ένα επιδόρπιο”.

Το 2014 κυκλοφόρησε το “Lollipop”, και ακολούθησαν “Marshmallow”, “Nougat”, “Oreo”, “Pie”. Το 2019 η ονομασία άλλαξε και έτσι βγήκαν πλέον τα “11” και “12” που τρέχει μέχρι και σήμερα.[4]

### 1.4 Η εξέλιξη του λογοτύπου

Το λογότυπο του Android έχει αλλάξει 5 φορές απο το 2007. Το πρώτο λογότυπο φαίνεται στην εικόνα παρακάτω χρησιμοποιήθηκε από το 2007 μέχρι και το 2014.



Σχήμα 1.1: 1ο Λογότυπο Android

Το δεύτερο λογότυπο φαίνεται στην εικόνα παρακάτω χρησιμοποιήθηκε για ένα χρόνο από το 2014 μέχρι και το 2015.



Σχήμα 1.2: 2ο Λογότυπο Android

Το τρίτο λογότυπο φαίνεται στην εικόνα παρακάτω χρησιμοποιήθηκε για τέσσερα χρόνια από το 2015 μέχρι και το 2019.



Σχήμα 1.3: 3ο Λογότυπο Android

Το τέταρτο λογότυπο φαίνεται στην εικόνα παρακάτω χρησιμοποιήθηκε για τέσσερα χρόνια από το 2019 μέχρι και το 2023.



Σχήμα 1.4: 4ο Λογότυπο Android

Το πέμπτο λογότυπο φαίνεται στην εικόνα παρακάτω που χρησιμοποιείται μέχρι και σήμερα από το 2023.



Σχήμα 1.5: 5ο Λογότυπο Android

## 1.5 Χαρακτηριστικά και λειτουργίες

- Λειτουργίες οθόνης: Η πλατφόρμα είναι προσαρμόσιμη σε πολλές αναλύσεις οθόνης (από VGA μέχρι 4K), δισδιάστατες ψηφιακές γραφικές βιβλιοθήκες, 17 τρισδιάστατα γραφικά βασισμένα στην OpenGL ES 3.0+ έκδοση χαρακτηριστικών, καθώς και παραδοσιακές απεικονίσεις οθόνης "έξυπνων" συσκευών κινητής τηλεφωνίας.
- Αποθήκευση Δεδομένων: Χρήση βάσης δεδομένων SQLite για τις ανάγκες αποθήκευσης.
- Συνδεσιμότητα: Το Android υποστηρίζει τεχνολογίες συνδεσιμότητας συμπεριλαμβανομένου GSM/LTE,3G,4G,CDMA,Bluetooth,NFC και Wi-Fi.
- Αποστολή μηνυμάτων: SMS και MMS είναι οι διαθέσιμοι τρόποι ανταλλαγής μηνυμάτων.
- Περιήγηση στο διαδίκτυο: Για την περιήγηση στον ιστό το Android διαθέτει φυλλομετρητή το Chrome. Και άλλοι φυλλομετρητές είναι διαθέσιμοι από το Google play.
- Υποστήριξη Java: Λογισμικό γραμμένο στην Java είναι δυνατόν να μεταγλωττιστεί και να εκτελεστεί στην εικονική μηχανή Dalvik, η οποία αποτελεί εξειδικευμένη υλοποίηση εικονικής μηχανής, σχεδιασμένης για χρήση σε φορητές συσκευές.
- Υποστήριξη υλικού: Μπορεί να συνεργαστεί με κάμερες στατικής ή κινούμενης εικόνας, οθόνες αφής, GPS, αισθητήρες επιτάχυνσης, μαγνητόμετρα, δισδιάστατους καθώς και τρισδιάστατους επιταχυντές γραφικών.
- Περιβάλλον ανάπτυξης λογισμικού: Περιλαμβάνει ένας προσομοιωτή συσκευής, εργαλεία για διόρθωση σφαλμάτων, μνήμη και εργαλεία ανάλυσης της απόδοσης του εκτελέσιμου λογισμικού καθώς και ένα επιπρόσθετο για το Android Studio.
- Αγορά και εγκατάσταση εφαρμογών: Το Play Store είναι ένας κατάλογος εφαρμογών που μπορούν να μεταφορτωθούν και εγκατασταθούν στην συσκευή άμεσα μέσω ασύρματων καναλιών, χωρίς την χρήση υπολογιστή.
- Οθόνη αφής πολλαπλών σημείων: Το λειτουργικό Android υποστηρίζει οθόνες αφής πολλαπλών σημείων.

## 1.6 Εκδόσεις Android

Παρακάτω κατατίθενται οι βασικές εκδόσεις του Android Studio με αύξουσα χρονολογική σειρά .

- Έκδοση: 1.6 , Ονομασία: Donut , Ημερομηνία : 15/09/2009
- Έκδοση: 2.0 , Ονομασία: Eclair , Ημερομηνία : 26/10/2009
- Έκδοση: 2.2 , Ονομασία: Froyo , Ημερομηνία : 20/05/2010
- Έκδοση: 2.3 , Ονομασία: Gingerbread, Ημερομηνία : 9/02/2011
- Έκδοση: 3.2. ,Ονομασία: Honeycomb, Ημερομηνία : 15/07/2011
- Έκδοση: 4.0. ,Ονομασία: Ice Cream Sandwich, Ημερομηνία : 16/12/2011
- Έκδοση: 4.1. ,Ονομασία: Jelly Bean, Ημερομηνία : 9/07/12
- Έκδοση: 4.4. ,Ονομασία: KitKat, Ημερομηνία : 31/10/2023
- Έκδοση: 5.0. ,Ονομασία: Lollipop, Ημερομηνία : 3/11/2014
- Έκδοση: 6.0. ,Ονομασία: Marshmallow, Ημερομηνία : 5/10/2015
- Έκδοση: 7.0. ,Ονομασία: Nougat, Ημερομηνία : 22/08/2016
- Έκδοση: 8.0. ,Ονομασία: Oreo, Ημερομηνία : 21/08/2016
- Έκδοση: 9. ,Ονομασία: Pie, Ημερομηνία : 6/08/2018
- Έκδοση: 10. ,Ονομασία: 10, Ημερομηνία : 3/09/2019

- Έκδοση: 11. ,Ονομασία: 11, Ημερομηνία : 8/09/2020
- Έκδοση: 12. ,Ονομασία: 12, Ημερομηνία : 4/10/2021
- Έκδοση: 13. ,Ονομασία: 13, Ημερομηνία : 15/08/2022
- Έκδοση: 14. ,Ονομασία: 14, Ημερομηνία : 4/10/2023

Ενδεικτικά το logo κάποιων εκδόσεων



Σχήμα 1.6: Έκδοση Froyo



Σχήμα 1.7: Έκδοση Lollipop



Σχήμα 1.8: Έκδοση Nougat



Σχήμα 1.9: Έκδοση Oreo



Σχήμα 1.10: Έκδοση Pie

## 1.7 Επίλογος

Σε αυτό το κεφάλαιο είδαμε τις βασικές πληροφορίες γύρω από το Android αλλά και την χρονική της εξέλιξη. Είδαμε την εξέλιξη του λογότυπου, τα χαρακτηριστικά και τις λειτουργίες αλλά και τις βασικές εκδόσεις.

## Κεφάλαιο 2ο: Ανάλυση Απαιτήσεων και Προδιαγραφών

### 2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα αναλύσουμε τις απαιτήσεις και τις προδιαγραφές που χρειάζονται για να μπορεί να χρησιμοποιεί κάποιος την εφαρμογή Sb. Θα δούμε τις λειτουργίες που έχει ο κάθε χρήστης ανάλογο του ρόλου του.

### 2.2 Χρήστες Συστήματος

Οι χρήστες χωρίζονται σε 2 κατηγορίες : “Καθηγητής” και ”Μαθητής” , με την κάθε κατηγορία να έχει διαφορετικές λειτουργίες και επιλογές . Και οι 2 ρόλοι μπορούν να:

- Δημιουργήσουν,τροποποιήσουν και διαγράψουν τον λογαριασμό τους.
- Σύνδεση και αποσύνδεση από την εφαρμογή
- Προβολή όλων των μαθημάτων τους
- Προβολη leaderboard των μαθητών τους ανα μάθημα
- Προβολη about όπου αναγράφονται πληροφορίες της εφαρμογής.
- Προβολή υλικού μαθήματος

#### 2.2.1 Χρήστης “Καθηγητης”

- Δημιουργία ,τροποποίηση και διαγραφή μαθημάτων
- Ανέβασμα τροποποίηση και διαγραφή ύλης ή υλικού σε μορφή pdf για κάθε μάθημα
- Προβολή , πρόσθεση και αφαίρεση μαθητή από κάθε μαθημα
- Αποστολή μηνύματος σε μαθητή του και προβολή των μηνυμάτων που είτε έστειλε ή έλαβε
- Προβολή όλων των μαθητών και δυνατότητα βαθμολόγησης

#### 2.2.2 Χρήστης “Μαθητής”

- Αποστολή μηνύματος σε καθηγητή του και προβολή των μηνυμάτων που είτε έστειλε ή έλαβε
- Προβολή όλων των καθηγητών
- Προβολή βαθμολογιών του ανα μάθημα

### 2.3 Λειτουργικές Απαιτήσεις

Για να μπορεί κάποιος χρήστης να χρησιμοποιεί την SB υπάρχουν κάποιες απαιτήσεις οι οποίες αναφέρονται παρακάτω

#### 2.3.1 Συσκευή με λειτουργικό Android

Η εφαρμογή την δεδομένη στιγμή υποστηρίζει μόνο λειτουργικό android, οπότε απαιτείται ο χρήστης να διαθέτει συσκευή με λειτουργικό Android.

### **2.3.2 Λογαριασμό στην SB**

Για να μπορεί κάποιος να έχει πρόσβαση στην εφαρμογή και να έχει πρόσβαση στις λειτουργίες της είναι απαραίτητο η δημιουργία λογαριασμού σε αυτήν για να γίνεται και η ταυτοποίηση του.

### **2.3.3 Σύνδεση στο διαδίκτυο**

Η εφαρμογή δεν υποστηρίζει χρήση χωρίς σύνδεση στο δίκτυο. Είναι πολύ σημαντικό να υπάρχει συνεχής ενημέρωση των πληροφοριών ,για αυτό τον λόγο είναι και αναγκαίο ο χρήστης να διαθέτει σύνδεση στο δίκτυο.

### **2.3.4 Επιλογή ρόλου “Καθηγητή”**

Αν κάποιος χρήστης επιθυμεί να αλλάξει ρόλο και να έχει αυτόν του καθηγητή ,απαιτείται να γνωρίζει τον μυστικό κωδικό καθώς θα του ζητηθεί για να μπορέσει να γίνει η αλλαγή. Σε πραγματικό χρόνο αυτόν τον κωδικό θα τον λαμβάνει απο τον κατασκευαστή της εφαρμογής. Στην SB ο κωδικός είναι “teacher”

## **2.4 Επίλογος**

Σε αυτό το κεφάλαιο είδαμε και αναλύστε τις προϋποθέσεις που χρειάζεται να πληρεί ο χρήστης για να λειτουργήσει την εφαρμογή και τις διαχωρίσαμε και στους 2 ρόλους που είναι διαθέσιμοι : καθηγητής και μαθητής.

## Κεφάλαιο 3ο: Εργαλεία και ιστοσελίδες που χρησιμοποιήθηκαν

### 3.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα δούμε τι εργαλεία και ποιές ιστοσελίδες χρησιμοποιήθηκαν για την υλοποίηση της SB.

### 3.2 Android Studio

Το Android Studio είναι το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για το λειτουργικό σύστημα Android της Google, βασισμένο στο λογισμικό IntelliJ IDEA της JetBrains και είναι σχεδιασμένο ειδικά για ανάπτυξη Android.[9]. Είναι διαθέσιμο για λειτουργικά συστήματα Windows, macOS και Linux. Το Android Studio διατίθεται με άδεια χρήσης Apache, αλλά συνοδεύεται επίσης από ορισμένες ενημερώσεις SDK που υπόκεινται σε μη ελεύθερη άδεια χρήσης, γεγονός που το καθιστά μη λογισμικό ανοιχτού κώδικα.

Το Android Studio ανακοινώθηκε στις 16 Μαΐου 2013, στο συνέδριο Google I/O. Βρισκόταν σε στάδιο προεπισκόπησης πρώιμης πρόσβασης ξεκινώντας από την έκδοση 0.1 τον Μάιο του 2013, και στη συνέχεια εισήλθε σε στάδιο beta ξεκινώντας από την έκδοση 0.8 η οποία κυκλοφόρησε τον Ιούνιο του 2014.

Μέχρι τις 7 Μαΐου 2019, η Java ήταν η προτιμώμενη γλώσσα της Google για την ανάπτυξη εφαρμογών Android, αλλά εγκαταστάθηκε από την Kotlin. Η Java εξακολουθεί να υποστηρίζεται, όπως και η C++.[9]



Σχήμα 3.1: Λογότυπο Android Studio

### 3.3 Canva

Η Canva είναι μια αυστραλιανή πολυεθνική εταιρεία ιδιόκτητου λογισμικού που παρέχει μια πλατφόρμα γραφιστικής με εργαλεία για τη δημιουργία γραφικών μέσων κοινωνικής δικτύωσης, παρουσιάσεων, καρτ ποστάλ, διαφημιστικών προϊόντων και ιστοσελίδων. Η υπηρεσία, η οποία ξεκίνησε στην Αυστραλία το 2013, προσφέρει εργαλεία σχεδιασμού για ιδιώτες και εταιρείες. Οι προσφορές της περιλαμβάνουν πρότυπα για παρουσιάσεις, αφίσες και περιεχόμενο μέσων κοινωνικής δικτύωσης, καθώς και λειτουργίες για επεξεργασία φωτογραφιών και βίντεο.

Η πλατφόρμα χρησιμοποιεί μια διεπαφή drag-and-drop που αποσκοπεί στην απλοποίηση της διαδικασίας σχεδιασμού. Η Canva λειτουργεί με βάση ένα μοντέλο freemium και έχει επεκτείνει τις υπηρεσίες της με την πάροδο των ετών ώστε να περιλαμβάνει μια πλατφόρμα προϊόντων εκτύπωσης και ένα εργαλείο επεξεργασίας βίντεο.[10]

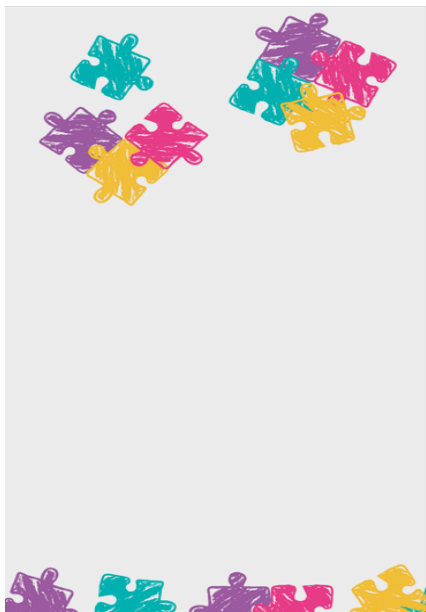
Ιστοσελίδα: <https://www.canva.com/>



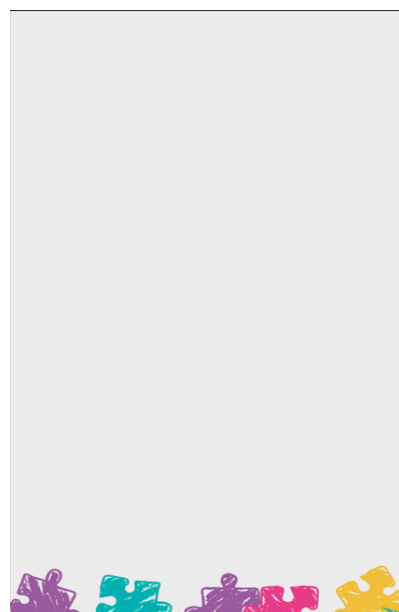
Σχήμα 3.2: Λογότυπο Canva

#### 3.3.1 Background images

Στην πλατφόρμα το canva δημιουργήθηκαν τα background των activities/fragments της SB. Παρακάτω στις εικόνες στα αριστερά βλέπουμε το background των log/sign up activity και στα δεξιά το background όλων των υπόλοιπων οθονών.



Σχήμα 3.3: log/Sign background



Σχήμα 3.4: general background

### 3.4 Iconik Ai

Το Iconik Ai είναι μια πλατφόρμα στην οποία μπορεί κανείς να δημιουργήσει εκπληκτικά εικονίδια εφαρμογών για Android, iOS και εφαρμογές web. Με τη γεννήτρια εικονιδίων που βασίζεται σε τεχνητή νοημοσύνη, μπορεί κάποιος να απελευθερώσει τη δημιουργικότητά του και να σχεδιάσει προσαρμοσμένα εικονίδια που αντιπροσωπεύουν τέλεια την ταυτότητα της εφαρμογής του. Στην πλατφόρμα αυτή δημιουργήθηκε το icon launcher της εφαρμογής.

Ιστοσελίδα: <https://www.iconikai.com/>



Σχήμα 3.5: Λογότυπο SB

### 3.5 Επίλογος

Σε αυτό το κεφάλαιο είδαμε και αναλύσαμε εργαλεία και ιστοσελίδες που χρειάστηκαν για την υλοποίηση της SB αλλά και κάποιες βασικές πληροφορίες για αυτά .

## Κεφάλαιο 4ο: Αρχιτεκτονική Συστήματος

### 4.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα αναλύσουμε την αρχιτεκτονική της εφαρμογής . Χωρίζεται σε 3 βασικούς άξονες οι οποίοι αναφέρονται παρακατω.

### 4.2 Frontend (Android app)

#### 4.2.1 Kotlin

Η SB είναι γραμμένη σε γλώσσα Kotlin. Η Kotlin είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που σχεδιάστηκε από την ομάδα JetBrains. Είναι μια γλώσσα προγραμματισμού ανοιχτού κώδικα (OP), γενικής χρήσης που συνδυάζει τόσο αντικειμενοστραφή όσο και συναρτησιακά χαρακτηριστικά. Ο ενθουσιασμός που έχει προκύψει γύρω από την Kotlin δημιουργείται από το γεγονός ότι η Google την χαρακτήρισε επίσημα ως την γλώσσα για την ανάπτυξη εφαρμογών Android όπως έχει αναφερθεί παραπάνω [9]. Η Kotlin είναι επίσης συνεργάσιμη με τις υπάρχουσες γλώσσες Android.



Σχήμα 4.1: Λογότυπο Kotlin

#### 4.2.2 Activities , Fragments , Dialogs

Το ui της SB αποτελείται από Activities , Fragments αλλα και Dialogs με τα components του καθενός.

### 4.2.2.1 Activities

Μια activity είναι ένα κρίσιμο στοιχείο του πλαισίου εφαρμογών Android στο Kotlin (ειδικά κατά την ανάπτυξη εφαρμογών Android χρησιμοποιώντας το Kotlin). Είναι μια ενιαία οθόνη με ένα περιβάλλον χρήστη (UI), όπως ένα παράθυρο ή μια σελίδα σε μια εφαρμογή ή ιστότοπο για υπολογιστές.

Η SB έχει συνολικά 5 activities όπως φαίνεται και παρακάτω . Περισσότερες πληροφορίες θα δοθούν παρακάτω στην ανάλυση του κώδικα

```
</> activity_create_course.xml
</> activity_login.xml
</> activity_main.xml
</> activity_signup.xml
</> activity_splash.xml
```

Σχήμα 4.2: Activities

### 4.2.2.2 Fragments

Στο Android, το τμήμα fragment είναι το μέρος της Activity που αντιπροσωπεύει ένα τμήμα της UI στην οθόνη.

Η SB έχει συνολικά 11 fragments όπως φαίνεται και παρακάτω . Περισσότερες πληροφορίες θα δοθούν παρακάτω στην ανάλυση του κώδικα

```
</> fragment_about.xml
</> fragment_course_content.xml
</> fragment_leaderboard.xml
</> fragment_my_courses.xml
</> fragment_my_grades.xml
</> fragment_my_messages.xml
</> fragment_my_students.xml
</> fragment_my_teachers.xml
</> fragment_profile.xml
</> fragment_students.xml
</> fragment_teacher_grade.xml
```

Σχήμα 4.3: Fragments

#### 4.2.2.3 Dialogs

Ένα dialog είναι ένα μικρό παράθυρο που ζητά από τον χρήστη να λάβει μια απόφαση ή να εισαγάγει πρόσθετες πληροφορίες. Ένα παράθυρο διαλόγου δεν γεμίζει την οθόνη και χρησιμοποιείται συνήθως για συμβάντα που απαιτούν από τους χρήστες να προβούν σε μια ενέργεια πριν προχωρήσουν.

Η SB έχει συνολικά 4 custom dialogs όπως φαίνεται και παρακάτω . Περισσότερες πληροφορίες θα δοθούν παρακάτω στην ανάλυση του κώδικα

```
</> add_content_dialog.xml
</> course_dialog.xml
</> dialog_add_student.xml
</> dialog_edit_grade.xml
```

Σχήμα 4.4: Dialogs

#### 4.2.3 View Binding

Το view binding είναι μια λειτουργία που εισήχθη στο Android Studio 3.6 και εξάλειψε την ανάγκη επανειλημμένης κλήσης της συνάρτησης findViewById(). Αντίθετα, δημιουργεί μια κλάση σύνδεσης

για κάθε ένα από τα αρχεία διάταξης XML, παρέχοντας άμεσες αναφορές στις προβολές εντός αυτής της διάταξης.

Στην παρακάτω εικόνα φαίνεται στιγμιότυπο για το πως γίνεται η σύνδεση με αρχείο xml είτε από activity(1η εικόνα) , είτε από fragment(2η εικόνα)

```
class LoginActivity : AppCompatActivity() {  
  
    private lateinit var binding: ActivityLoginBinding // Declare the binding variable  
    private lateinit var auth: FirebaseAuth  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
  
        binding = ActivityLoginBinding.inflate(layoutInflater)  
        setContentView(binding.root)  
    }  
}
```

Σχήμα 4.5: activity view binding

```
class AboutFragment : Fragment(){  
  
    private var _binding: FragmentAboutBinding? = null  
    private val binding get() = _binding!!  
  
    override fun onCreateView(  
        inflater: LayoutInflater, container: ViewGroup?,  
        savedInstanceState: Bundle?  
    ): View {  
        _binding = FragmentAboutBinding.inflate(inflater, container, attachToParent: false)  
        return binding.root  
    }  
}
```

Σχήμα 4.6: fragment view binding

### 4.3 Backend (Firebase)

Η SB αποθηκεύει όλα τα δεδομένα του κάθε χρήστη στο Firebase. Το Firebase είναι μια πλατφόρμα ανάπτυξης εφαρμογών Backend-as-a-Service (BaaS) που παρέχει φιλοξενούμενες υπηρεσίες backend, όπως βάση δεδομένων σε πραγματικό χρόνο, αποθήκευση στο cloud, έλεγχο ταυτότητας, αναφορά σφαλμάτων, μηχανική μάθηση, απομακρυσμένη διαμόρφωση και φιλοξενία για τα στατικά σας αρχεία.[12]. Παρακάτω αναφέρονται οι 3 υπηρεσίες που εντάχθηκαν στην εφαρμογή.



Σχήμα 4.7: firebase logo

### 4.3.1 Firebase Authentication

Το Firebase Authentication δίνει τη δυνατότητα να ενσωματωθεί έλεγχος ταυτότητας Firebase με ένα προσαρμοσμένο σύστημα ελέγχου ταυτότητας το οποίο καθορίζεται από τον προγραμματιστή. Η SB το επιτυγχάνει με συνδυασμό email/password. Παρακάτω δίνεται στιγμιότυπο με παράδειγμα από την κονσόλα του firebase.[13]

teststudent@gmail.com	✉	May 21, 2025	May 29, 2025	HfAE2W95QqMR7RitD945Rkc...
test@gmail.com	✉	May 4, 2025	May 29, 2025	U92QHdsYyaey5wBdEQeiVJD...

Σχήμα 4.8: Firebase Authentication

### 4.3.2 Firestore Database

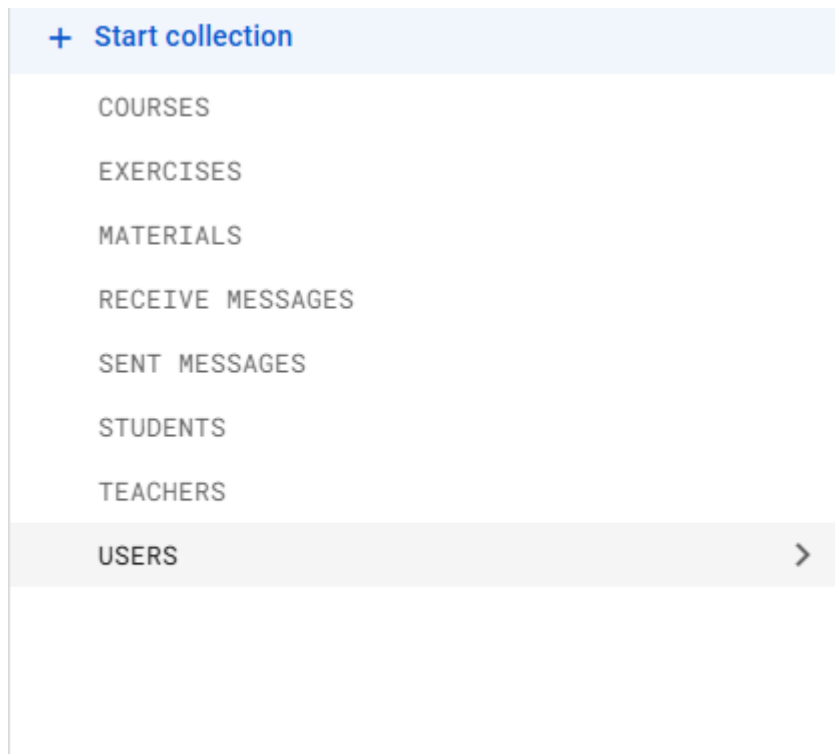
Το Cloud Firestore είναι μια βάση δεδομένων NoSQL που φιλοξενείται στο cloud, στην οποία οι εφαρμογές σας Apple, Android και web μπορούν να έχουν άμεση πρόσβαση μέσω εγγενών SDK. Το Cloud Firestore είναι επίσης διαθέσιμο σε εγγενή SDK Node.js, Java, Python, Unity, C++ και Go, εκτός από τα API REST και RPC.[14]

### 4.3.3 Firebase Storage

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our Apple platforms, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

### 4.4 Δομή δεδομένων

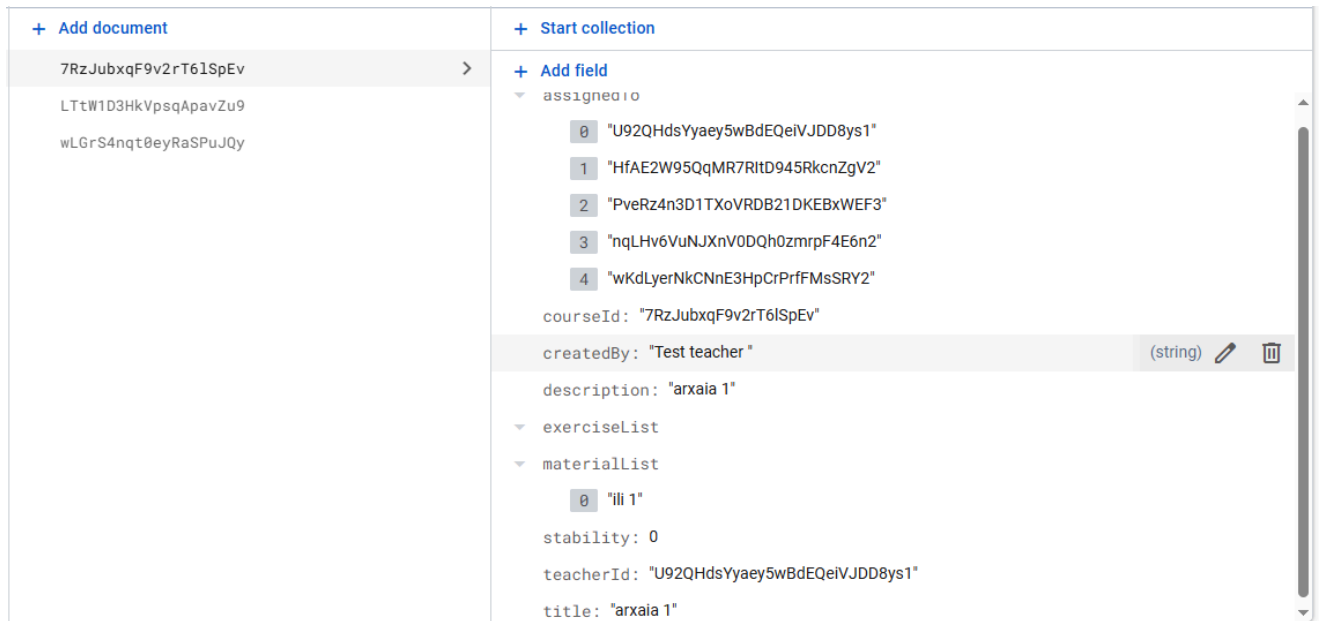
Τα δεδομένα στην firebase αποθηκεύονται σε collection όπως φαίνεται και στην εικόνα παρακάτω όπου βλέπουμε τα collections που έχει η SB. Το κάθε collection είναι η αποτύπωση μιας data class που έχουμε στον κώδικα



Σχήμα 4.9: Firebase Collections

#### 4.4.1 Συλλογή: COURSES

Εδώ αποθηκεύονται τα στοιχεία ενός μαθήματος όταν αυτό δημιουργείται. Συνδέεται με την data class Course.



Σχήμα 4.10: Course Collection

#### 4.4.2 Συλλογή: MATERIALS

Εδώ αποθηκεύονται τα στοιχεία της ύλης ενός μαθήματος όταν αυτό δημιουργείται. Συνδέεται με την data class Material και αντιπροσωπεύει αρχείο pdf που αποθηκεύεται στο storage.



Σχήμα 4.11: Material Collection

#### 4.4.3 Συλλογή: RECEIVE MESSAGES

Εδώ αποθηκεύονται τα στοιχεία των μηνυμάτων που λαμβάνει ο κάθε χρήστης όταν αυτό δημιουργείται. Περιέχει το subcollection RECEIVE MESSAGE DETAIL. Συνδέεται με την data class Receive Message και Receive Message Detail.

<a href="#">+ Add document</a>	<a href="#">+ Start collection</a>	<a href="#">+ Add document</a>
HFAE2W95QqMR7RItd945RkcnZgV2 >	RECEIVE MESSAGE DETAIL >	ZYGPqD40P3NEwCHYA90u
PveRz4n3D1TXoVRDB21DKEBxWEF3		

Σχήμα 4.12: Receive Message Collection

<a href="#">+ Start collection</a>	<a href="#">+ Add document</a>	<a href="#">+ Start collection</a>
RECEIVE MESSAGE DETAIL >	ZYGPqD40P3NEwCHYA90u >	<a href="#">+ Add field</a>
<a href="#">+ Add field</a>		dateAdded: 1748499796031 from: "U92QHdsYyaey5wBdEQeiVJDD8ys1" message: "delete test" stability: 0 to: "HfAE2W95QqMR7RItd945RkcnZgV2"

Σχήμα 4.13: Receive Message Detail Sub Collection

#### 4.4.4 Συλλογή: SENT MESSAGES

Εδώ αποθηκεύονται τα στοιχεία των μηνυμάτων που στέλνει ο κάθε χρήστης όταν αυτό δημιουργείται. Περιέχει το subcollection SENT MESSAGE DETAIL. Συνδέεται με την data class Sent Message και Sent Message Detail..

SENT MESSAGES	U92QHdsYyaey5wBdEQeiVJDD8ys1	SENT MESSAGE DETAIL
<a href="#">+ Add document</a>	<a href="#">+ Start collection</a>	<a href="#">+ Add document</a>
U92QHdsYyaey5wBdEQeiVJDD8ys1 >	SENT MESSAGE DETAIL >	7yAfEzbbyyAd60sXXD7Y kX94A61Y8XQHEJp80E5x
<a href="#">+ Add field</a>		

Σχήμα 4.14: Sent Message Collection

+ Start collection	+ Add document	+ Start collection
SENT MESSAGE DETAIL >	7yAfEzbbyyAd60sXXD7Y >	+ Add field
+ Add field	kX94A61Y8XQHEJp80E5x	dateAdded: 1748499796025
		from: "U92QHdsYyaey5wBdEQeiVJDD8ys1"
		message: "delete test"
		stability: 0
		to: "HfAE2W95QqMR7RitD945RkcnZgV2"

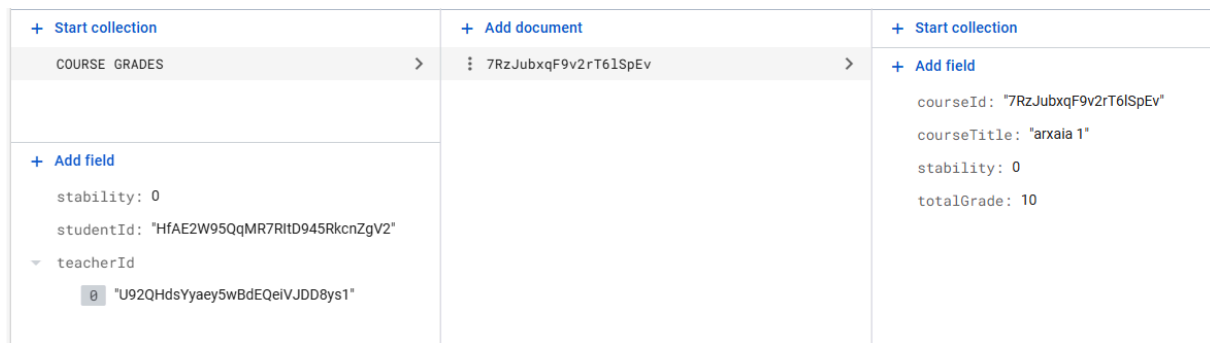
Σχήμα 4.15: Sent Message Detail Sub Collection

#### 4.4.5 Συλλογή: STUDENTS

Εδώ αποθηκεύονται οι μαθητες με τους καθηγητές τους και τους βαθμούς που παίρνουν σε κάθε μάθημα. Περιέχει την subcollection COURSE GRADES Συνδέεται με την data class Student και Course Grade.

+ Add document	+ Start collection	+ Add document
HfAE2W95QqMR7RitD945RkcnZgV2 >	COURSE GRADES >	7RzJubxqF9v2rT61SpEv
PveRz4n3D1TXoVRDB21DKEBxWEF3	+ Add field	
nqLHv6VuNJXnV8DQh0zmrpF4E6n2	stability: 0	
wKdLyerNkCnNE3HpCrPrFFMsSRy2	studentId: "HfAE2W95QqMR7RitD945RkcnZgV2"	
	teacherId	
	<input type="checkbox"/> "U92QHdsYyaey5wBdEQeiVJDD8ys1"	

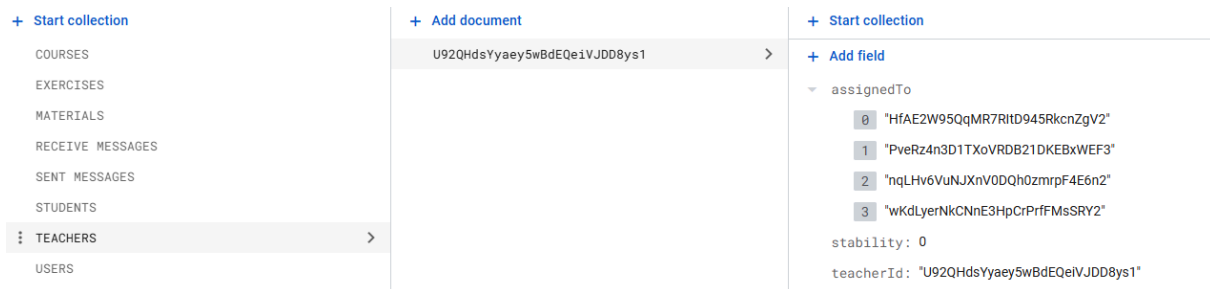
Σχήμα 4.16: Student Collection



Σχήμα 4.17: Course Grade Sub Collection

#### 4.4.6 Συλλογή: TEACHERS

Εδώ αποθηκεύονται οι καθηγητές και οι μαθητές που έχει ο καθένας. Συνδέεται με την data class Teacher.



Σχήμα 4.18: Teacher Collection

#### 4.4.7 Συλλογή: USERS

Εδώ αποθηκεύονται τα στοιχεία των χρηστών κατα την εγγραφή τους στην εφαρμογή. Συνδέεται με την data class User.



Σχήμα 4.19: User Collection

#### 4.4.8 Συλλογή: EXERCISES

Εδώ αποθηκεύονται τα στοιχεία των ασκήσεων που περιέχει ένα μάθημα. Συνδέεται με την data class Material και αντιπροσωπεύει αρχείο pdf που αποθηκεύεται στο storage.



Σχήμα 4.20: Exercise Collection

#### 4.5 Επίλογος

Σε αυτό το κεφάλαιο είδαμε αναλυτικά την δομή που έχει η βάση της εφαρμογής στο firebase και αναλύσαμε τα βασικά της στοιχεία. Επίσης είδαμε και τα collections που έχει η βάση.

## **Κεφάλαιο 5ο: Δοκιμές εφαρμογής**

### **5.1 Εισαγωγή**

Σε αυτό το κεφάλαιο θα δούμε τον τρόπο με τον οποίο πραγματοποιήθηκαν οι δοκιμές της εφαρμογής.

### **5.2 Σεμινάρια Ελέγχου**

Ο τρόπος με τον οποίο έγιναν οι δοκιμές σε πρώτο στάδιο ήταν να τρέχει η εφαρμογή σε διάφορους emulator μέσα στο android studio με διαφορετική έκδοση ο καθένας. Αυτό έγινε για να μπορέσω να δω αν η εφαρμογή είναι λειτουργική σε διάφορες εκδόσεις του android. Κάθε φορά που άλλαζε ο κώδικας παράλληλα γινόταν και το τεστ στους emulators.

### **5.3 Αποτελεσματικότητα Δοκιμών**

Απο τις δοκιμές που έγιναν διαπιστώθηκε πως η εφαρμογή είναι λειτουργική σε διάφορες συσκευές, γίνεται σωστή και ορθή αποθήκευση αλλά και ανάκτηση των δεδομένων. Επίσης από αξιολογήσεις άλλων χρηστών σημειώθηκε πως η εφαρμογή προσφέρει ένα φιλικό αλλά και εύχρηστο UI.

### **5.4 Παρατηρήσεις**

Κατα την περίοδο των δοκιμών παρατηρήθηκε πως υπάρχουν μικρές καθυστερήσεις που οφείλονται στην κακή σύνδεση στο δίκτυο. Είναι εμφανές και στο χρήστη πως χρειάζονται 1 με 2 δευτερόλεπτα για να δούμε τα δεδομένα.

### **5.6 Επίλογος**

Σε αυτο το κεφάλαιο αναλύσαμε την δοκιμή της εφαρμογής αλλα έγινε και μια μικρή αξιολόγηση σχετικά με το αποτέλεσμα της.

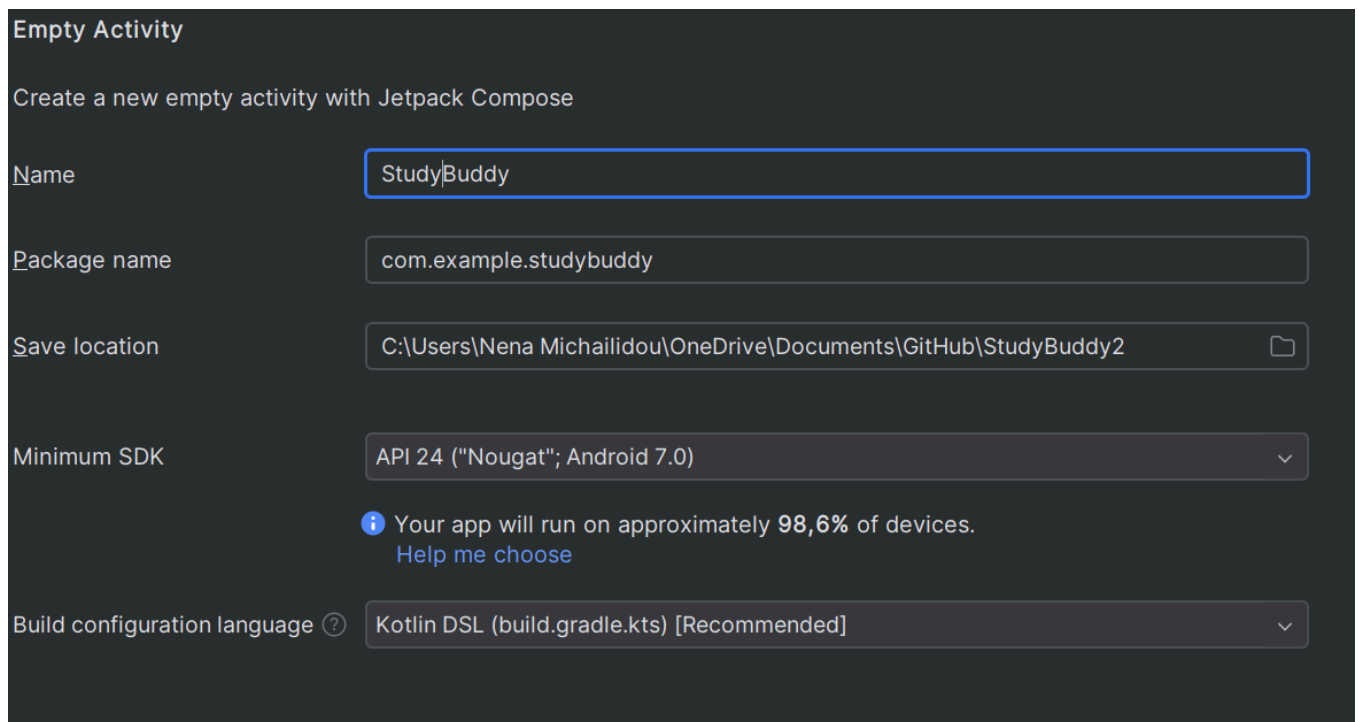
## Κεφάλαιο 6ο: Παρουσίαση της εφαρμογής StudyBuddy

### 6.1 Επίλογος

Σε αυτό το κεφάλαιο θα αναλύσουμε τον κώδικα της εφαρμογής.

### 6.2 Δημιουργία Εφαρμογής

Το πρώτο στάδιο είναι να δημιουργήσουμε το project και να επιλέξουμε την κατώτερη έκδοση στην οποία θα λειτουργεί η εφαρμογή. Επιλέγουμε το όνομα της εφαρμογής και την γλώσσα προγραμματισμού στην οποία θα προγραμματίσουμε. Η εφαρμογή είναι υλοποιημένη με η Kotlin όπως έχει αναφερθεί και παραπάνω, με έκδοση λειτουργικού συστήματος το Nougat. Και το όνομα της εφαρμογής είναι το StudyBuddy. Διάλεξα ένα όνομα που να αντιπροσωπεύει την λειτουργία της εφαρμογής.



Σχήμα 6.1: Λειτουργίες συστήματος

### 6.3 AndroidManifest

Το AndroidManifest περιέχει πληροφορίες σχετικά με μια εφαρμογή Android. Κάθε εφαρμογή έχει ένα μόνο αρχείο XML AndroidManifest στη ρίζα του συνόλου πηγαίου κώδικα που ονομάζεται AndroidManifest.xml. Το λειτουργικό σύστημα Android και οι υπηρεσίες ψηφιακής διανομής (για παράδειγμα, το Google Play) το χρησιμοποιούν για να βρουν πληροφορίες, όπως το όνομα της εφαρμογής, το σημείο εισόδου της εφαρμογής, την υποστήριξη έκδοσης Android, την υποστήριξη λειτουργιών υλικού και τα δικαιώματα εφαρμογής.[15]

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission
    android:name="android.permission.READ_EXTERNAL_STORAGE"
    android:maxSdkVersion="32" />
<uses-permission android:name="android.permission.READ_MEDIA_IMAGES" />
<uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
```

Σχήμα 6.2: Άδειες εφαρμογής

- Android.permission.INTERNET ζητάει δικαίωμα χρήσης internet είτε από δεδομένα κινητής τηλεφωνίας είτε από χρήση του Wi-Fi.
- Android.permission.READ\_EXTERNAL\_STORAGE ζητάει πρόσβαση στο storage όπου αποθηκεύονται τα pdf των μαθημάτων
- Android.permission.READ\_MEDIA\_IMAGES ζητάει πρόσβαση για την αποθήκευση φωτογραφιών.
- Android.permission.POST\_NOTIFICATIONS ζητάει άδεια για να στέλνει ειδοποιήσεις.

## 6.4 Βιβλιοθήκες

Το android όλες τις απαραίτητες βιβλιοθήκες που χρειάζεται η εφαρμογή της αποθηκεύει στο αρχείο `build.gradle(Module.app)` . Εκεί μπορεί κάποιος να βάλει καινούργιες βιβλιοθήκες για να τις χρησιμοποιήσει.

```
plugins {  
    alias(libs.plugins.android.application)  
    alias(libs.plugins.kotlin.android)  
    alias(libs.plugins.kotlin.compose)  
    alias(libs.plugins.google.gms.google.services)  
    id("kotlin-parcelize")  
}
```

Σχήμα 6.3: Plugins

Έχει προστεθεί το plugin για πρόσβαση στη firebase και το kotlin-parcelize για την εύκολη πρόσβαση στα data classes της εφαρμογής.

```
buildFeatures{  
    viewBinding = true  
    compose = true  
}
```

Σχήμα 6.4: Ενεργοποίηση view binding

Στην παραπάνω εικόνα βλέπουμε τον τρόπο με τον οποίο ενεργοποιούμε το view binding που αναφέρθηκε προηγουμένως.

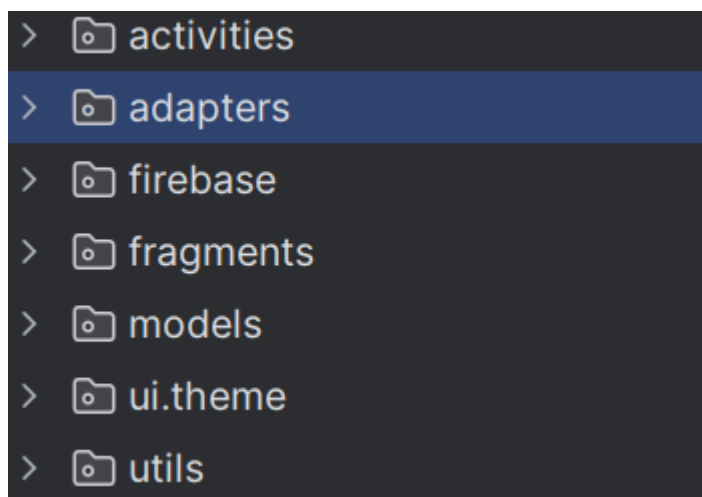
```
implementation(libs.firebase.storage)
implementation(libs.firebase.firestore)
implementation(libs.firebase.auth.ktx)
implementation(libs.hdodenhof.circleimageview)
implementation(libs.glide)
implementation(libs.androidx.navigation.fragment.ktx)
implementation(libs.androidx.navigation.ui.ktx)
implementation(libs.firebase.messaging)
```

Σχήμα 6.5: Dependencies

Στην παραπάνω εικόνα βλέπουμε τα Dependencies που έχουν προστεθεί στην εφαρμογή

### 6.5 Κλάσεις με ui

Για την καλύτερη οργάνωση των αρχείων έχουν ομαδοποιηθεί σε packages ανάλογα με την κατηγορία τους

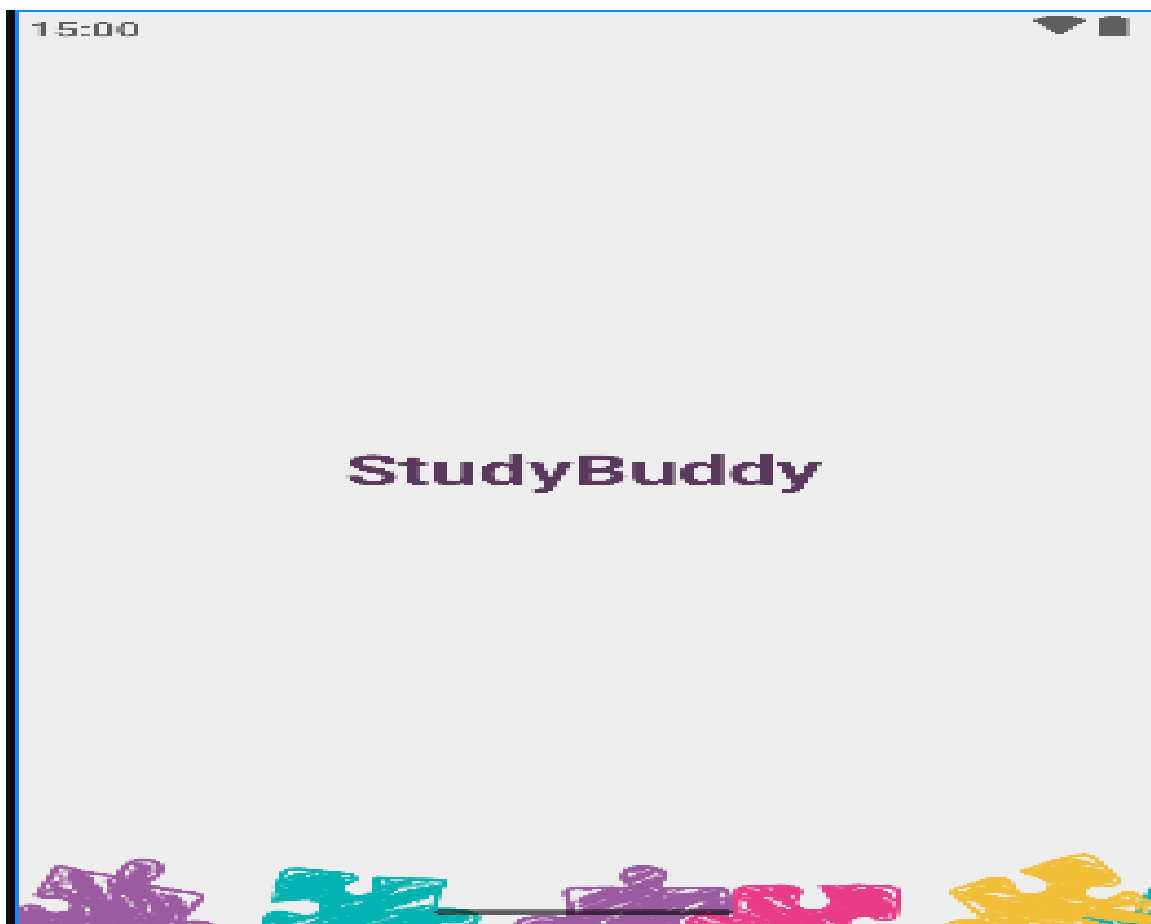


Σχήμα 6.6: Packages

## 6.5.1 Activities

### 6.5.1.1 SplashActivity

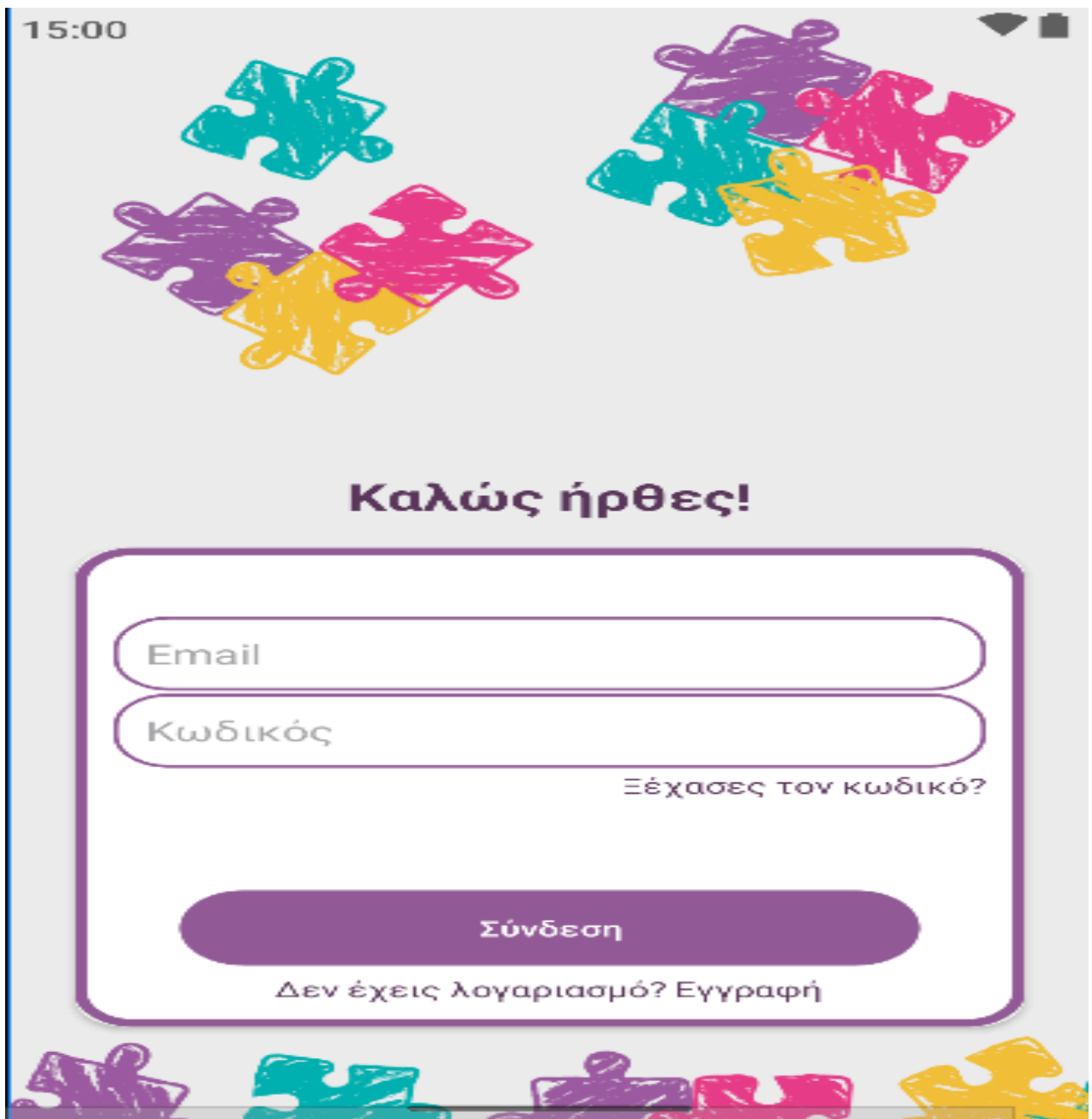
Ανοίγοντας την εφαρμογή βρισκόμαστε στη activity Splash Screen . Το layout περιέχει background image και textview με το όνομα της εφαρμογής. Με κατάλληλο κώδικα έχει γίνει full screen activity . Στο Handler ελέγχεται με την κατάλληλη μέθοδο της αν υπάρχει τρέχον χρήστης . Αν υπάρχει μας μεταφέρει στην Main μετα απο 2,5 δευτερόλεπτα , αλλιως μας πηγαίνει στην Login activity .



Σχήμα 6.7: SplashScreen

### 6.5.1.2 LoginActivity

Το layout περιέχει background image , textView για καλός όρισμα , για την επιλογή του ξέχασα τον κωδικό και την μεταφορά στην sign up activity στην περίπτωση που ο χρήστης δεν έχει λογαριασμό. Περιέχει επίσης ένα card view μέσα στο οποίο είναι τα απαραίτητα edittexts για την εισαγωγή email password . Τέλος υπάρχει το κουμπί που πραγματοποιεί την σύνδεση με την βάση. Μεσα στην LoginActivity.kt δίνεται λειτουργία με setOnClickListener στα 2 textView και μετα στο listener του button πρώτα γίνεται έλεγχος αν ο χρήστης έχει συμπληρώσει τα πεδία με κατάλληλη μέθοδο και εμφανίζονται τα απαραίτητα μηνύματα και στην συνέχεια δίνονται τα στοιχεία που έχει πληκτρολογήσει για να γίνει η σύνδεση . Με την auth.signInWithEmailAndPassword() ελέγχεται αν υπάρχει χρήστης με αυτά τα στοιχεία . Αν ναι , γίνονται load τα στοιχεία του και μεταφερόμαστε στην main με την μέθοδο load User Data() . Αν όχι,παίρνουμε το κατάλληλο μήνυμα.

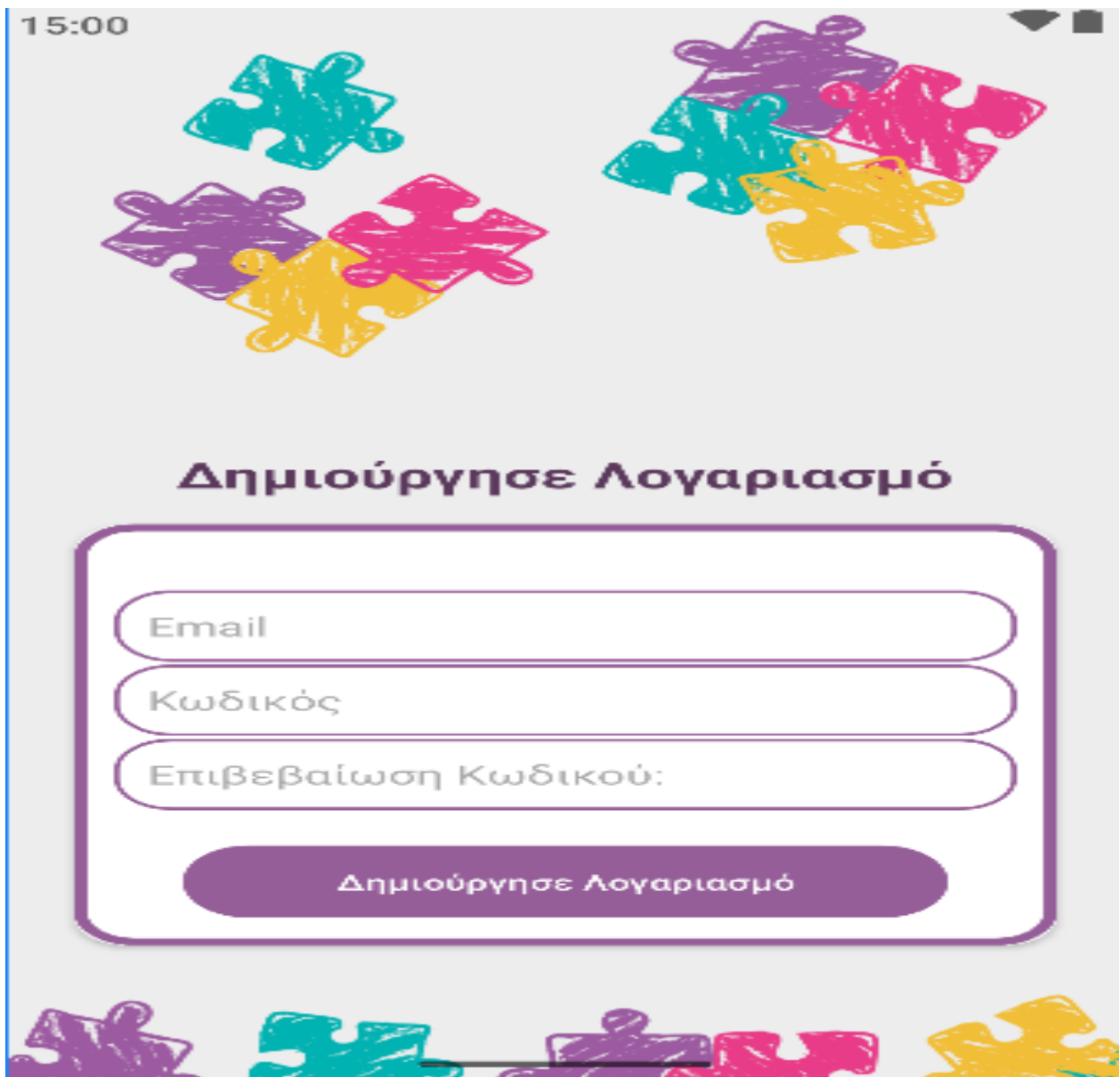


Σχήμα 6.8: Login

### 6.5.1.3 SignUpActivity

Το layout περιέχει background image και ένα card view μέσα στο οποίο είναι τα απαραίτητα edittexts για την εισαγωγή email , password και confirm password. Τέλος υπάρχει το κουμπί που πραγματοποιεί την εγγραφή του χρήστη με την βάση.

Μεσα στην signUpActivity.kt δίνεται λειτουργία με setOnClickListener στο button. Πρώτα γίνεται έλεγχος αν ο χρήστης έχει συμπληρώσει τα πεδία με κατάλληλη μέθοδο και εμφανίζονται τα απαραίτητα μηνύματα και στην συνέχεια δίνονται τα στοιχεία που έχει πληκτρολογήσει για να γίνει η εγγραφή . Με την createUserWithEmailAndPassword() προστίθεται στο authentication ο χρήστης και μετά καλείται η μέθοδος registerUser όπου και προστίθεται ο χρήστης στο collection User . Αν είναι επιτυχής , μεταφερόμαστε στο profil για συμπλήρωση . Αν όχι,παίρνουμε το κατάλληλο μήνυμα.

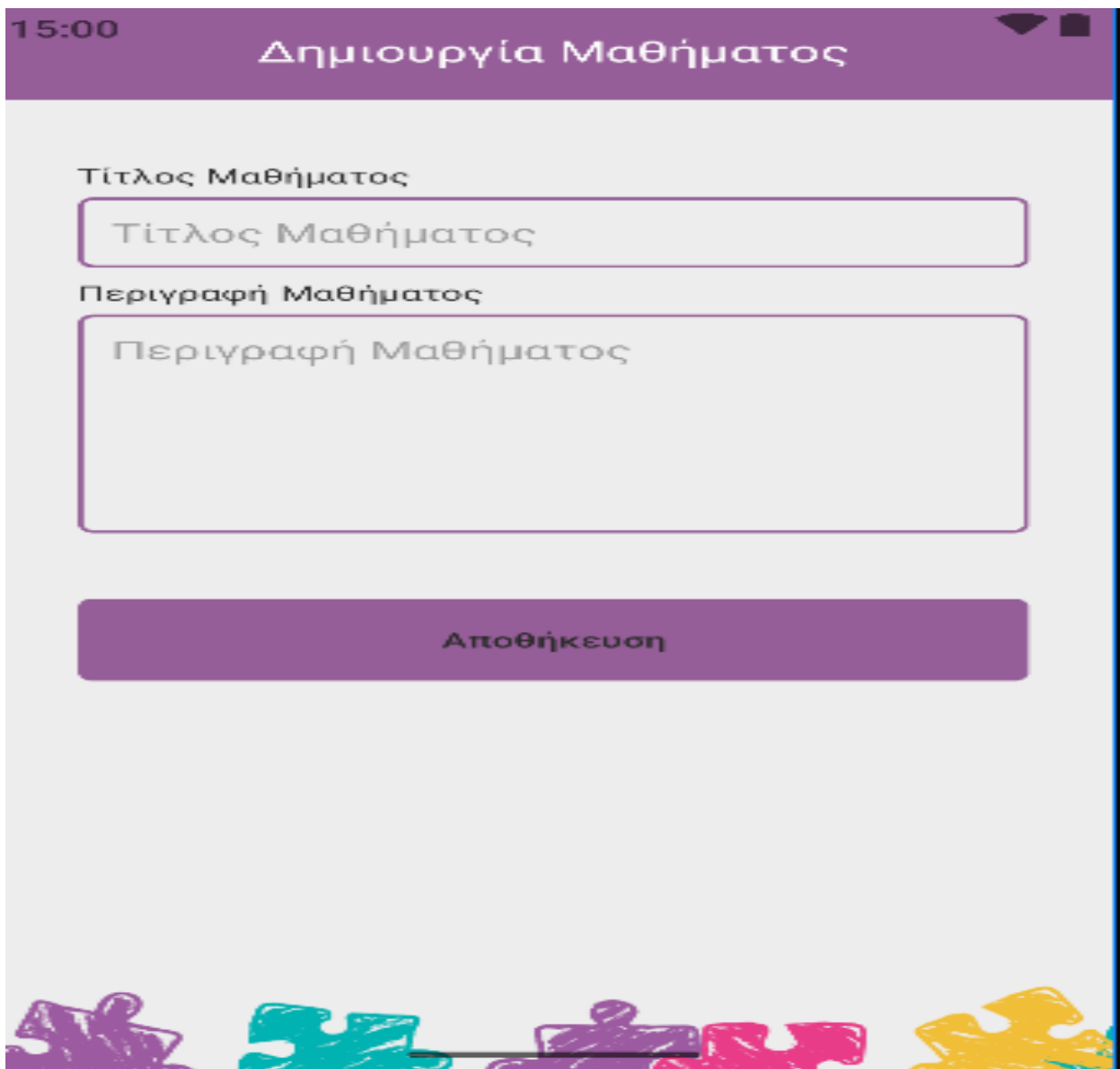


Σχήμα 6.9: SignUp

### 6.5.1.4 CreateCourseActivity

Το layout περιέχει background image , textview και edittext για την συμπλήρωση των στοιχείων του μαθήματος( τίτλο και περιγραφή) . Τέλος υπάρχει το κουμπί που πραγματοποιεί την δημιουργία του μαθήματος στη βάση.

Στο activity μεταφερόμαστε από το MyCoursesFragment όπου και επιστρέφουμε μόλις δημιουργηθεί καινούργιο μάθημα.



Σχήμα 6.10: CreateCourse

### 6.5.1.5 Main Activity

Το layout περιέχει background image, drawer menu για την διευκόλυνση του χρήστη. Το κύριο μενού( δεξιά εικόνα ) περιέχει ένα card view με ένα recycler στο οποίο φαίνονται σε όσα μαθήματα είναι εγγεγραμμένος ο χρήστης. Μέσα στο recycler υπάρχει ένα editText, το οποίο είναι ορατό μόνο αν δεν υπάρχουν μαθήματα του χρήστη. Ακόμα υπάρχει ένα floating button για την δημιουργία νέου μαθήματος το οποίο μας πηγαίνει στην CreateCourseActivity..

Το item menu της ταινίας μας αναφέρει το όνομα της ταινίας , το όνομα του χρήστη που την δημιούργησε και την υπόθεση της. Όταν πατήσει κάποιος σε κάποια ταινία μεταφερόμαστε στο περιεχόμενο της.

Μέσα στην mainActivity.kt καλείται η loadUserData() της fb . Δίνονται στοιχεία στο recycler και ελέγχεται για το αν υπάρχουν ταινίες ή όχι. Αν να εμφανίζουμε τις ταινίες στο recycler, αν όχι γίνεται φανερό το textView. Επίσης δίνεται και η λειτουργία στο drawer menu με onBackPressed() για να κλείνουμε/ ανοίγουμε το drawer και onItemClick() για να μεταφερομαστε στα αντίστοιχα fragments.

Στην onCreate γίνεται έλεγχος σχετικά με το από πού ερχόμαστε. Αν ήρθαμε από το profileFragment ανοίγουμε το αντίστοιχο fragment αλλιώς μεταφερόμαστε στο mycoursesFragment. Ακόμη ορίζεται και για το πια menu θα είναι visible ανάλογα με τον ρόλο του χρήστη.



- Προφίλ
- ← Αποσύνδεση

---

- ✉ Τα μηνύματα μου
- 📖 Τα μαθήματα μου
- 🏠 Οι μαθητές μου
- 🏆 LeaderBoard

About

Σχήμα 6.11: TeacherMenu



- Προφίλ
- ← Αποσύνδεση

---

- ✉ Τα μηνύματα μου
- 📖 Τα μαθήματα μου
- 👨‍🏫 Οι καθηγητές μου
- 📊 Οι βαθμολογίες μου
- 🏆 LeaderBoard

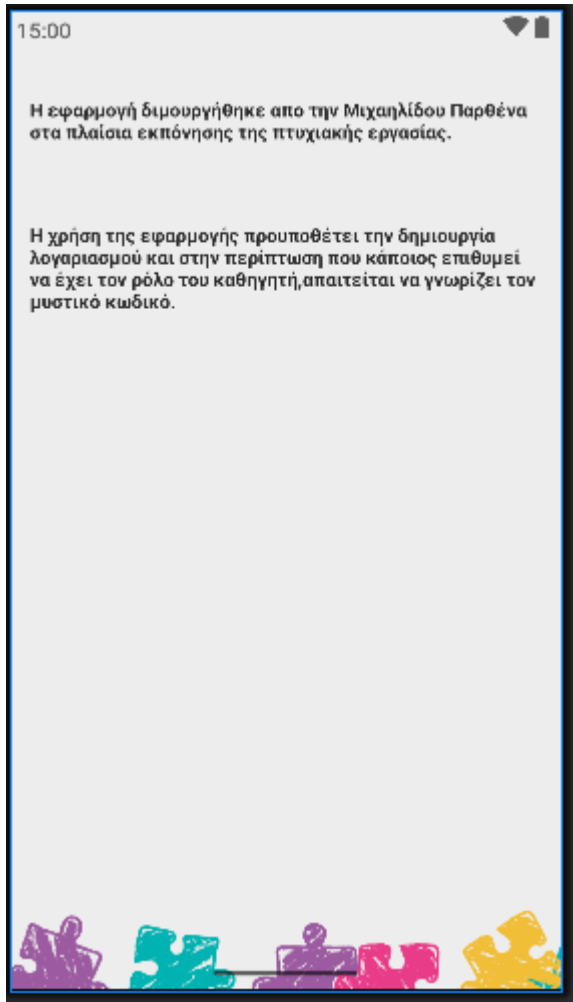
About

Σχήμα 6.12: StudentMenu

## 6.5.2 Fragments

### 6.5.2.1 About

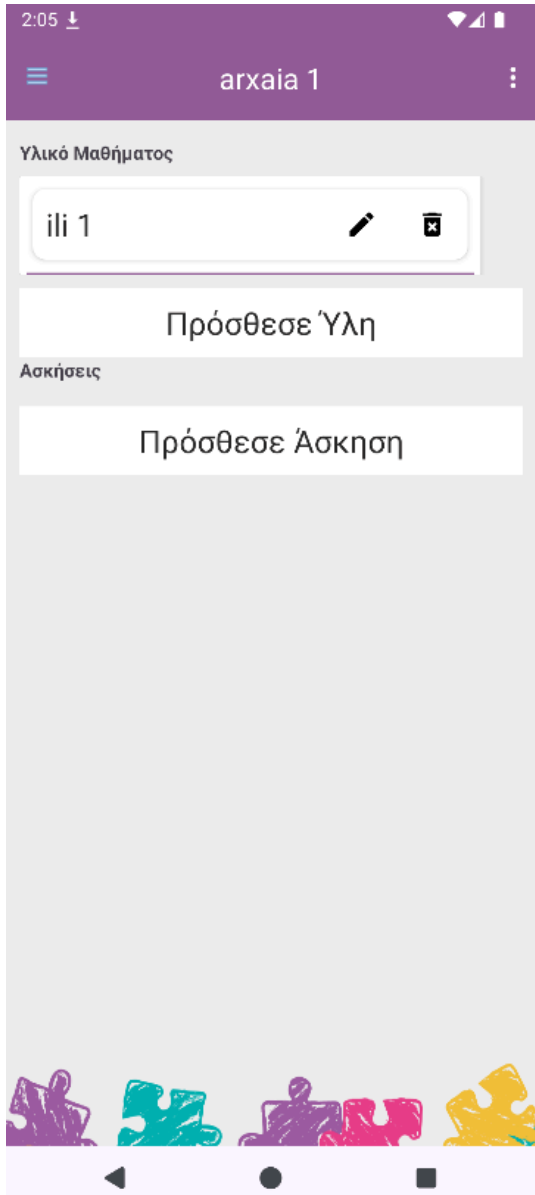
Το layout περιέχει background image και πληροφορίες για την εφαρμογή.



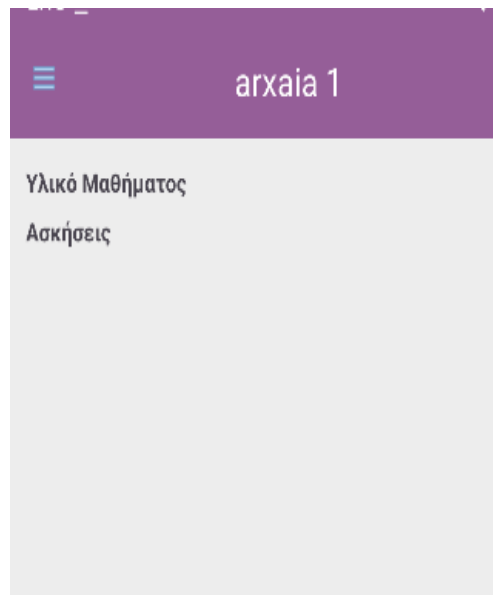
Σχήμα 6.13: About

### 6.5.2.2 CourseContent

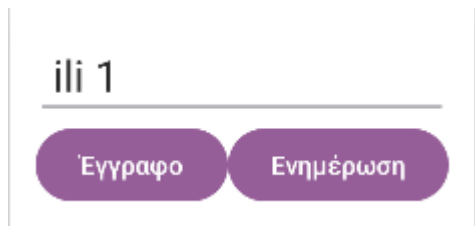
Το layout περιέχει background image ,2 recycler και textview για την προσθήκη ύλης και ασκήσεων . Αν δεν υπάρχουν ήδη δεδομένα τα recycler είναι κρυμμένα. Με το πάτημα σε textview ανοίγει το κατάλληλο dialog για την εισαγωγή αρχείου pdf. Αν ο χρήστης δεν είναι αυτός που έχει δημιουργήσει το μάθημα δεν βλέπει τα textview. Όπως επίσης δεν είναι φανερό και το edit/delete. Με click μπορεί ο χρήστης να κατεβάσει το αρχείο στο κινητό του και αντιστοίχα να αλλάξει ή να διαγράψει το *item* με το κατάλληλο dialog που εμφανίζεται. Ακόμη υπάρχει ένα το drawer menu αριστερά και δεξιά το μενού που μας στέλνει στα μέλη του μαθήματος. Και αυτό είναι κρυμμένο αν δεν είναι ο χρήστης που το έχει δημιουργήσει.



Σχήμα 6.14: CourseContentTeacher



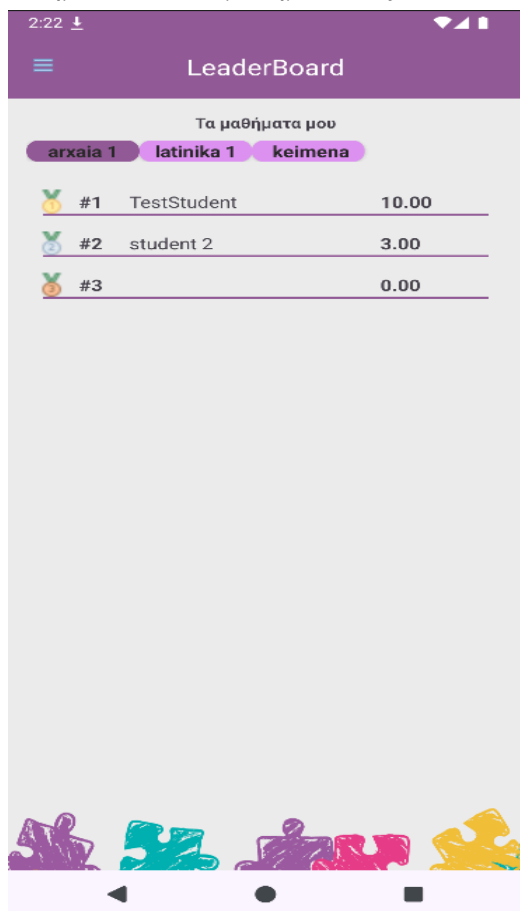
Σχήμα 6.15: CourseContentStudent



Σχήμα 6.16: editCourseDialog

### 6.5.2.3 LeaderBoard

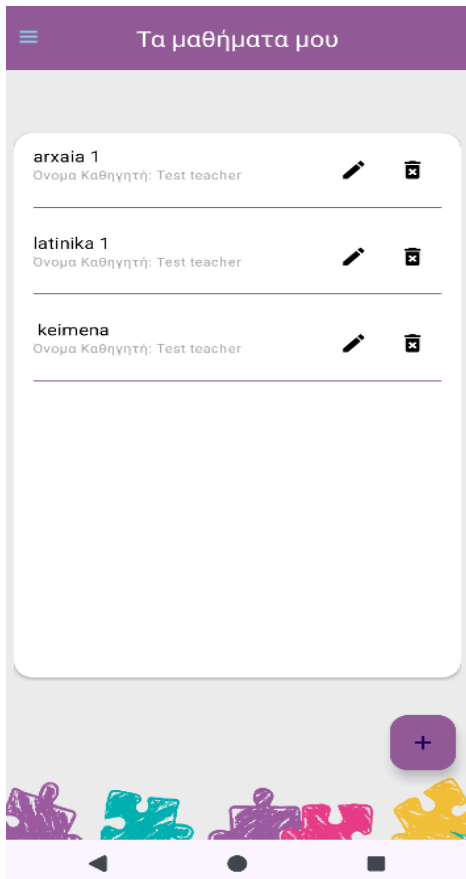
Το layout περιέχει background image και 2 recycler Στο πρώτο φαίνονται όσα μαθήματα είναι εγγεγραμμένος ο χρήστης και στο δεύτερο φαίνονται οι μαθητές με την βαθμολογία τους ξεκινώντας από αυτόν με την μεγαλύτερη. Αν δεν υπάρχουν ήδη δεδομένα τα recycler είναι κρυμμένα. Με το πάτημα σε κάποιο μάθημα αλλάζει το κάτω recycler με το αντίστοιχο leaderboard.



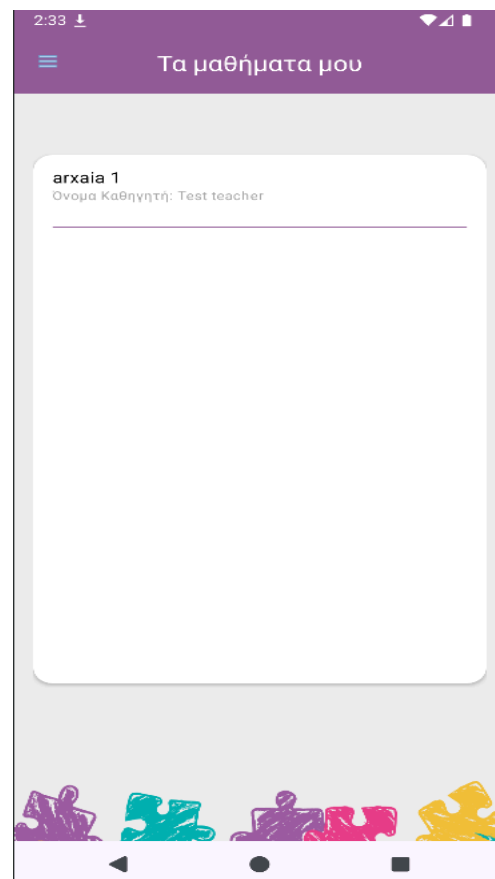
Σχήμα 6.17: Leaderboard

### 6.5.2.4 MyCourses

Το layout περιέχει background image και 1 recycler όπου και φαίνονται όλα τα μαθήματα που είναι εγγεγραμμένος ο χρήστης. Γίνεται έλεγχος για το αν ο χρήστης είναι ο δημιουργός . Αν ναι τότε είναι visible η επιλογή του edit/delete. Ακόμη είναι visible και η επιλογή του να προσθέσεις και καινούργιο μάθημα. Με το πάτημα σε κάποιο μάθημα μεταφερόμαστε στο περιεχόμενο του μαθήματος.



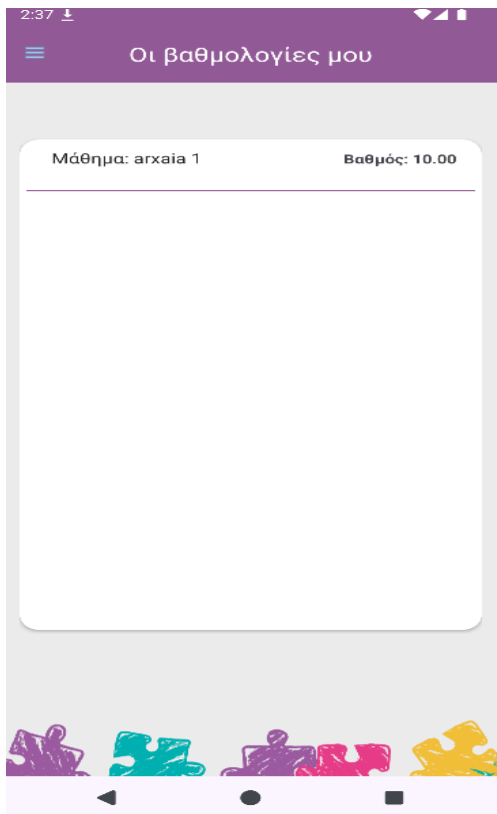
Σχήμα 6.18: My courses Teacher



Σχήμα 6.19: My courses Student

### 6.5.2.5 MyGrades

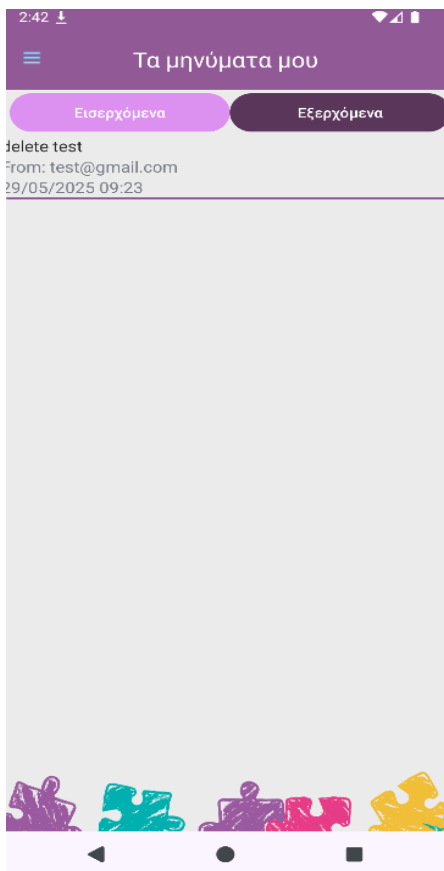
Το layout περιέχει background image και 1 recycler όπου και φαίνονται όλα τα μαθήματα που είναι εγγεγραμμένος ο χρήστης αλλά αυτή τη φορά με τη βαθμολογία . Αυτό το fragment είναι visible στους μαθητές .



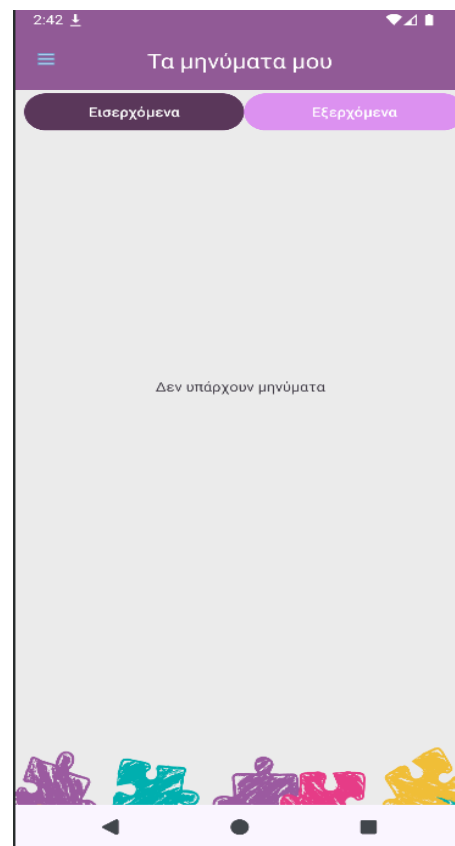
Σχήμα 6.20: MyGrades

### 6.5.2.6 MyMessages

Το layout περιέχει background image ,2 buttons και 1 recycler όπου και φαίνονται τα μηνύματα του χρήστη ανάλογα με το αν επιλέξει να δει τα εισερχόμενα ή τα εξερχόμενα μηνύματα του. Ανάλογα με το ποιο κουμπί είναι επιλεγμένο αλλάζει και το χρώμα του κουμπιού . Αν δεν υπάρχουν μηνύματα εμφανίζεται το κατάλληλο μήνυμα.



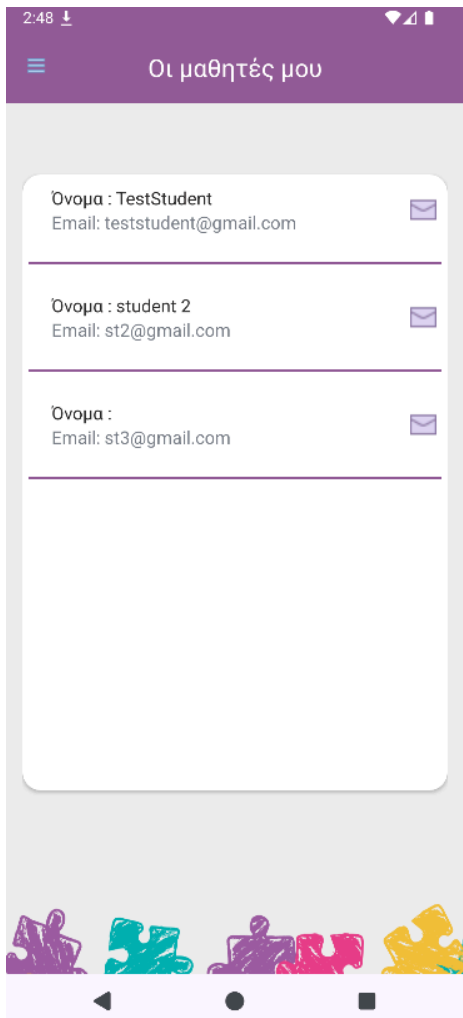
Σχήμα 6.21:ReceiveMessage



Σχήμα 6.22:SentMessages

### 6.5.2.7MyStudents

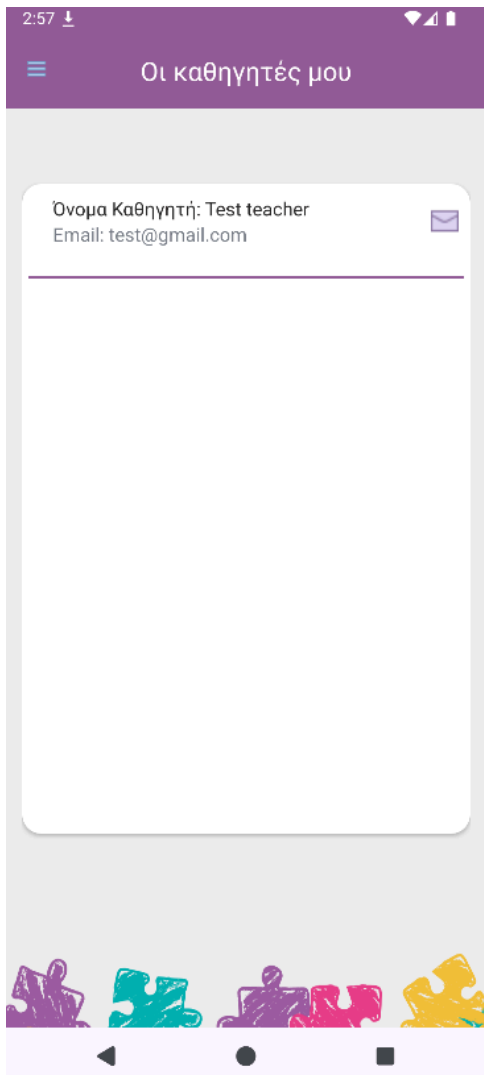
Το layout περιέχει background image και 1 recycler όπου και φαίνονται όλοι οι μαθητές του χρήστη. Αυτό το layout είναι εμφανές μόνο στους καθηγητές. Αν δεν υπάρχουν ακόμα μαθητες εμφανίζεται το κατάλληλο μήνυμα. Αν πατήσει ο χρήστης πάνω στο εικονίδιο του μηνύματος στέλνει μήνυμα στην συγκεκριμένο μαθητη και αν πατήσει πάνω στον μαθητη θα μεταφερθεί στο TeacherGradeStudent για βαθμολόγηση.



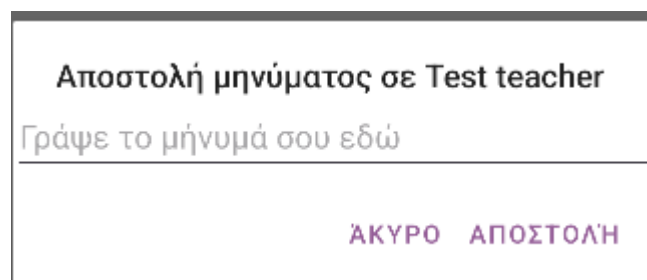
Σχήμα 6.23:MyStudents

### 6.5.2.8 MyTeachers

Το layout περιέχει background image και 1 recycler όπου και φαίνονται όλοι οι καθηγητές του χρήστη. Αυτό το layout είναι εμφανές μόνο στους μαθητές. Αν δεν υπάρχουν ακόμα καθηγητές εμφανίζεται το κατάλληλο μήνυμα. Αν πατήσει ο χρήστης πάνω στο εικονίδιο του μηνύματος στέλνει μήνυμα στην συγκεκριμένο καθηγητή



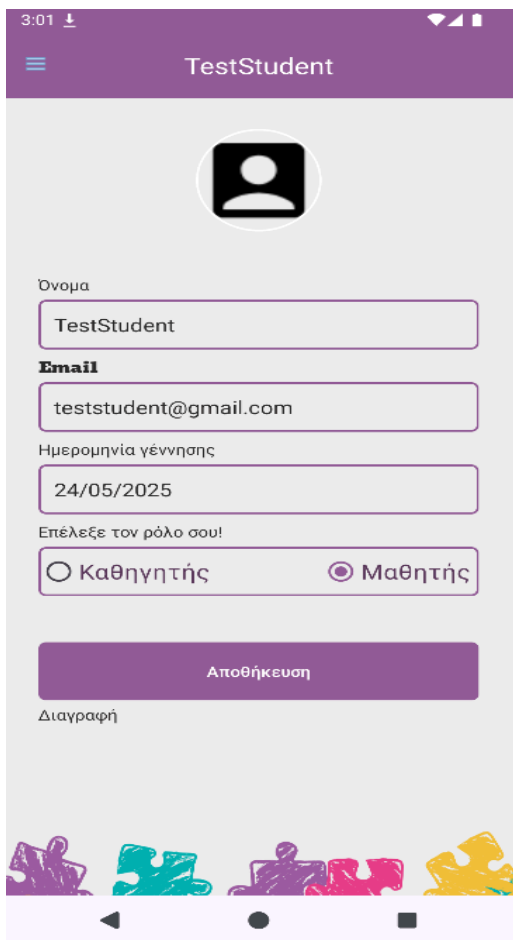
Σχήμα 6.24: MyTeachers



Σχήμα 6.25: SendMessage

## 6.5.2.9 Profile

To layout περιέχει background image και 2 edittexts για την εισαγωγή name και birthdate. Το Edittext του Email έχει ήδη συμπληρωθεί με την κατάλληλη μέθοδο και δεν μπορείς να το αλλάξεις. Για την ημερομηνία ανοίγει ένα dialog για διευκόλυνση του χρήστη. Με την δημιουργία του προφίλ ο χρήστης παίρνει το ρόλο του μαθητή και εδώ μπορεί να τον αλλάξει. Ο χρήστης από δω αν ερχόμαστε απο την main αποθηκεύει για πρώτη φορά το προφίλ του ή αλλιώς αν θέλει αλλάζει τα πεδία που μπορεί.



Σχήμα 6.26: Profil

### Επιβεβαίωση Καθηγητή

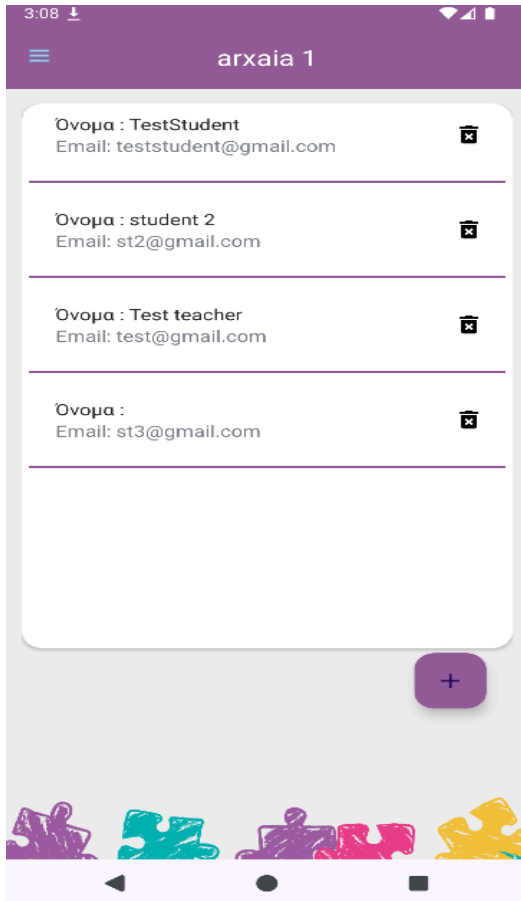
Εισάγετε τον μυστικό κωδικό

Άκυρο OK

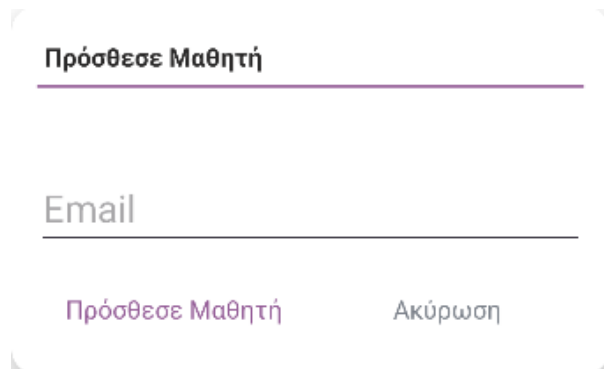
Σχήμα 6.27: ChangeRole

### 6.5.2.10 Students

Το layout περιέχει background image και 1 recycler στο οποίο φαινονται όσοι μαθητες είναι μέλη σε αυτο το μάθημα και ενα button για την εισαγωγή νέου μαθητη. Ακόμη αν πατήσει ο χρήστης πάνω στον μαθητή μπορεί να το αφαιρέσει από το μάθημα. Πρόσβαση έχει μόνο ο χρήστης που δημιουργήσει το μάθημα. Έχει εξηγηθεί στο courseContentFragmnet.



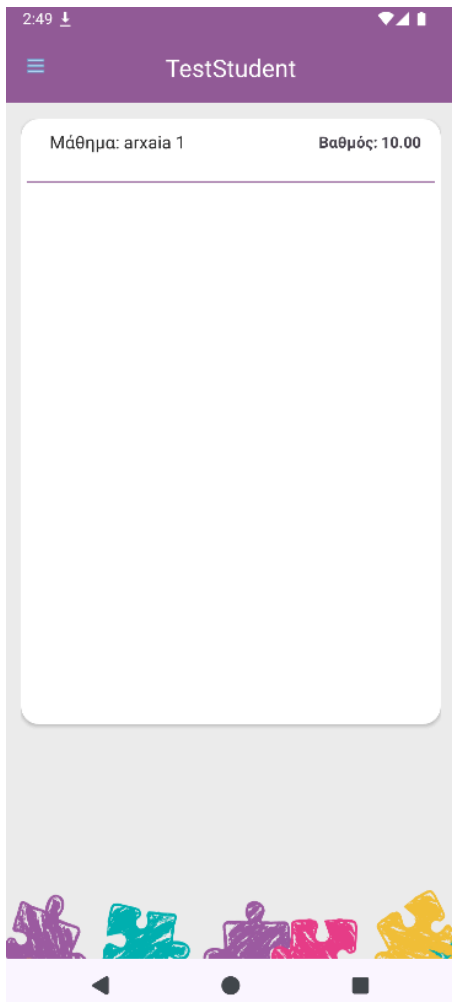
Σχήμα 6.28: Students



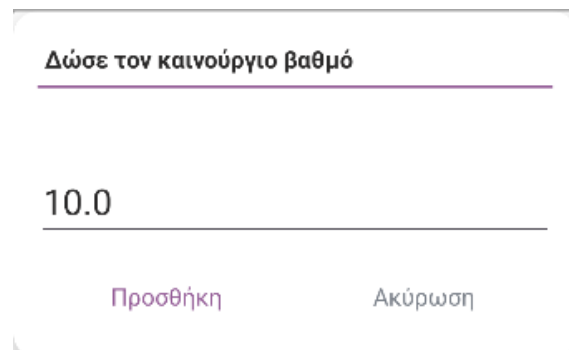
Σχήμα 6.29: AddStudent

### 6.5.2.11 TeacherGradeStudent

Το layout περιέχει background image και 1 recycler στο οποίο φαινονται όλα τα μαθήματα που έχει ο χρήστης με τον εκάστοτε μαθητή και οι βαθμολογίες αντίστοιχα. Αν πατήσει ο χρήστης πάνω στο μάθημα μπορεί να του αλλάξει τον βαθμό.



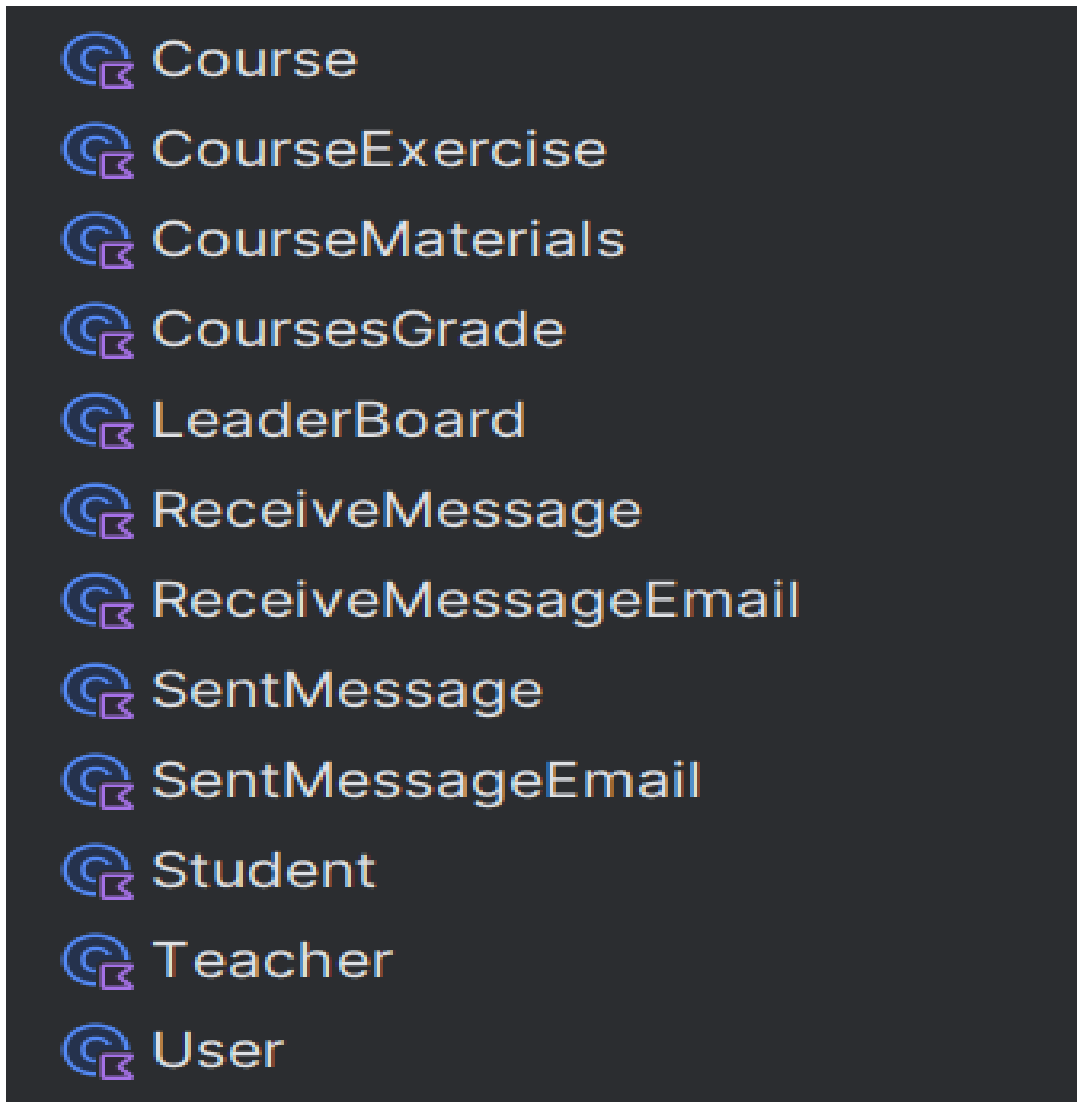
Σχήμα 6.30:StudentGrade



Σχήμα 6.31:changeGrade

## 6.6 Models package

Μέσα στο Models package περιέχονται όλες οι απαραίτητες data classes που δημιουργήθηκαν για την αποθήκευση και επεξεργασία δεδομένων στη βάση. Στην παρακάτω φωτογραφία φαίνονται όλες οι data classes.



Σχήμα 6.32:Data classes

## 6.7 Firebase package

Μεσα στο Firebase package υπάρχει η κλάση FirestoreClass() μέσα την οποία υπάρχει όλος ο κώδικας που αλληλεπιδρά με την βάση, οι μέθοδοι δηλαδή για αποθήκευση , διαγραφή , αλλαγή και φόρτωση όλων των collection.

- Users collection

```
/**
 * A function to add a new user
 */
fun registerUser(activity: SignUpActivity,userInfo : User){...}

/**
 * A function to update the users Data
 */
fun updateUserProfileData(fragment: ProfileFragment, userHashMap: HashMap<String, Any>) {...}

/**
 * A function to get the user details and act according the activity or fragment
 */
fun loadUserData(activity: Activity? = null, fragment: Fragment? = null){...}

/**
 * A function to load the details of a specific assignedTo user with the email
 */
fun loadAssignedToUserDetails(fragment: StudentsFragment, email: String){...}

/**
 * A function to delete the user's profile
 */
fun deleteUser(fragment: ProfileFragment) {...}
```

Σχήμα 6.33:User Methods

- Course Collection

```
fun createCourse(activity: CreateCourseActivity, course: Course) {...}
/**
 * A function to update Course details
 */
fun updateCourse(course: Course, onComplete: () -> Unit) {...}
/**
 * A function to update Course assignedTo
 */
fun addAssignedStudentToCourse(activity: StudentsFragment, course: Course, user: User){...}
/**
 * A function to load the list with the assignedTo users
 */
fun loadAssignedStudentsListDetail(activity: StudentsFragment, assignedTo: ArrayList<String>)
/**
 * A function to load courses in which the currentUser is assignedTo
 */
fun loadCoursesList(fragment: Fragment){...}
/**
 * A function to load the details of a specific course
 */
fun loadCourseDetails(fragment: CourseContentFragment, courseId: String){...}
/**
 * A function to delete a course
 */
fun deleteCourse(courseId: String, onComplete: () -> Unit) {...}
```

Σχήμα 6.34:Course Methods

- Material Collection

```
/**
 * A function to add a new material
 */
fun addMaterial(material: CourseMaterials, onComplete: () -> Unit){...}

/**
 * A function to add a new material to course
 */
fun addMaterialToCourse(courseId: String ,materialTitle: String){...}

/**
 * A function to load the materials
 */
fun loadMaterials(courseId: String, onComplete: (ArrayList<CourseMaterials>) -> Unit){...}

/**
 * A function to update the materials
 */
fun updateMaterial(material: CourseMaterials, onComplete: () -> Unit) {...}

/**
 * A function to delete the materials
 */
fun deleteMaterial(materialId: String, onComplete: () -> Unit) {...}
```

Σχήμα 6.35:Material Methods

- Exercise Collection

```
/**
 * A function to add a new exercise
 */
fun addExercise(exercise: CourseExercise, onComplete: () -> Unit){...}

/**
 * A function to add a new material to course
 */
fun addExerciseToCourse(courseId: String ,exerciseTitle: String){...}

/**
 * A function to load the exercises
 */
fun loadExercise(courseId: String, onComplete: (ArrayList<CourseExercise>) -> Unit){...}

/**
 * A function to update the exercises
 */
fun updateExercise(exercise: CourseExercise, onComplete: () -> Unit) {...}

/**
 * A function to delete the exercises
 */
fun deleteExercise(exerciseId: String, onComplete: () -> Unit) {...}
```

Σχήμα 6.36:Exercise Methods

- Teacher Collection

```
/**
 * A function to add a new student to Teachers
 */
fun addStudentToTeacher(activity: StudentsFragment, teacher: Teacher, studentId: String) {...}

/**
 * A function to load the list of assignedStudents of a specific teacher
 */
fun loadAssignedStudents(teacherId: String, onComplete: (List<User>) -> Unit) {...}

/**
 * A function to delete a student from assignedTo in course
 */
fun deleteStudentFromCourse(studentId: String , courseId: String, onComplete: () -> Unit) {...}

/**
 * A function to delete a course from student
 */
private fun deleteCourseFromStudent(studentId: String , courseId: String){...}

/**
 * A function to add a new student to Students
 */
fun addStudentToStudents(studentId: String, courseId: String) {...}
```

Σχήμα 6.37:Teacher Methods

- Student Collection

```
/**
 * A function to add a new student to Students
 */
fun addStudent(student: Student) {...}

/**
 * A function to add a new courseGrade to Students
 */
fun addGradeToStudent(courseGrade: CoursesGrade , studentId: String) {...}

/**
 * A function to load the list of a specific student's teachers
 */
fun loadStudentsTeacher(studentId: String, onComplete: (List<User>) -> Unit) {...}

/**
 * A function to load a student's grades
 */
fun loadStudentGrades(studentId: String, fragment: Fragment){...}

/**
 * A function to update a course's grade pf a specific student
 */
fun updateCourseGrade(studentId: String ,newGrade: Double, courseId: String , onComplete: ()
```

Σχήμα 6.38: Student Methods

- Sent Message Collection

```
/**
 * A function to add current user's sent messages
 */
fun addSendMessage(currentUser: String, messageInfo: SendMessage) {...}

/**
 * A function to load current user's sent messages
 */
fun loadSentMessages(currentUserId: String, onComplete: (List<SendMessageEmail>) -> Unit) {.
```

Σχήμα 6.39:Sent Message Methods

- Receive Message Collection

```
/**
 * A function to add current user's message to the owner of the message
 */
fun addReceiveMessage(ownerId: String, messageInfo: ReceiveMessage) {...}

/**
 * A function to load current user's receive messages
 */
fun loadReceiveMessages(currentUserId: String, onComplete: (ArrayList<ReceiveMessageEmail>
) {...}
```

Σχήμα 6.40:Receive Message Methods

## 6.8 Επίλογος

Σε αυτό το κεφάλαιο έγινε παρουσίαση της εφαρμογής. Είδαμε τις οθόνες που υπάρχουν και την λειτουργία τους. Επίσης είδαμε και σημαντικά κομμάτια του κώδικα καθώς και μεθόδους.

## **Κεφάλαιο 7ο: Συμπεράσματα και Μελλοντική Επέκταση**

### **7.1 Επίλογος**

Στο κεφάλαιο αυτό, παρουσιάζουμε τα συμπεράσματα της υλοποίησης της εργασίας που πραγματοποιήθηκε. Τέλος γίνεται λόγος για τυχόν μελλοντικές επεκτάσεις, ώστε η εφαρμογή να γίνει αποδοτικότερη ως προς τον χρήστη.

### **7.2 Συμπέρασμα**

Στόχος της παρούσας πτυχιακής ήταν, η δημιουργία ενός συστήματος για την διαχείριση εκπαιδευτικού υλικού ιδιαίτερων μαθημάτων . Ο στόχος επιτεύχθηκε, καθώς δημιουργήσαμε μία έξυπνη εφαρμογή, δίνει στον χρήστη τη δυνατότητα δημιουργία και επεξεργασία μαθήματος, πρόσθεση και αφαίρεση μαθητών , βαθμολόγηση αλλά και συνομιλία με τους μαθητές. Επίσης μέσα από μενού μπορεί να πλοηγείται με ευκολία στις λειτουργίες της εφαρμογής.

### **7.3 Μελλοντική Επέκταση**

Η εφαρμογή StudyBuddy που δημιουργήθηκε στα πλαίσια της εκπόνησης της πτυχιακής εργασίας, σχεδιάστηκε με σκοπό να βοηθήσει τον χρήστη και να κάνει την ζωή του πιο εύκολη. Παρά το γεγονός ότι η συγκεκριμένη εφαρμογή έφερε εις πέρας τους στόχους που είχαν τεθεί, σίγουρα μπορούν να γίνουν αρκετές επεκτάσεις ώστε να βελτιωθεί ακόμη περισσότερο η εφαρμογή.

Αρχικά το επόμενο στάδιο είναι μέσω της τεχνητής νοημοσύνης να δίνεται η δυνατότητα στον καθηγητή να δημιουργούνται αυτόματα τεστ είτε σε μορφή κουιζ αλλά και συμπλήρωση κενών. Ίσως να δίνεται και η ευκαιρία να διαλέγει ο κάθε χρήστης τι είδους τεστ προτιμάει.

Έπειτα θα ήταν χρήσιμο να υπάρχει και υποστήριξη offline mode για κάποια σημεία της εφαρμογής.

Και τέλος αλλά και απαραίτητο να στέλνεται notification σε συγκεκριμένο χρήστη.

# ΒΙΒΛΙΟΓΡΑΦΙΑ

## Βιβλία

[1] Denis Panjuta , Loveth Nwokike , *Tiny Android Projects Using Kotlin*, February 2, 2024

## Internet Site

[2] <https://www.doctorandroid.gr/p/iphone.html>

[3] <https://www.lovefortechology.net/2024/11/The%20-Birth-%20of%20-Android-%20Historical%20-Review.html>

[4] <https://iguru.gr/istoria-tou-android/>

[5] <https://www.pcedugr.com/post/%CF%84%CE%B9-%CE%B1%CE%BA%CF%81%CE%B9%CE%B2%CF%8E%CF%82-%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9-%CF%84%CE%BF-%CE%BB%CE%B5%CE%B9%CF%84%CE%BF%CF%85%CF%81%CE%B3%CE%B9%CE%BA%CF%8C-%CF%83%CF%8D%CF%83%CF%84%CE%B7%CE%BC%CE%B1-android>

[6] <https://ti-einai.gr/android/>

[7] <https://www.lovefortechology.net/2024/11/The%20-Birth-%20of%20-Android-%20Historical%20-Review.html>

[8] <https://el.wikipedia.org/wiki/Android>

[9] [https://en.wikipedia.org/wiki/Android\\_Studio](https://en.wikipedia.org/wiki/Android_Studio)

[10] <https://en.wikipedia.org/wiki/Canva>

[11] <https://www.iconikai.com/>

[12] <https://firebase.google.com/>

[13] <https://firebase.google.com/docs/auth/android/custom-auth>

[14] <https://firebase.google.com/docs/database>

[15] <https://docs.unity3d.com/6000.1/Documentation/Manual/android-manifest.html>

## Παράρτημα Α : Firebase Methods

Δύο πολύ σημαντικές μέθοδοι.

- `getCurrentUserId()` . Παιρνούμε το `id` του χρήστη που είναι συνδεδεμένος

```
/**
 * A function for getting the user id of current logged user.
 */
fun getCurrentUserId(): String{
    //it's ether empty
    val currentUser = FirebaseAuth.getInstance().currentUser
    var currentUserId = ""
    if(currentUser != null) {
        currentUserId = currentUser.uid
    }
    return currentUserId
}
```

Σχήμα Α.1: `getCurrentUserId()`

- `loadUserData()` . Παιρνούμε τα στοιχεία του χρήστη που είναι συνδεδεμένος και ανάλογα απο που καλείται αντιδρά διαφορετικά

```
fun loadUserData(activity: Activity? = null, fragment: Fragment? = null){
    db.collection( collectionPath: "USERS")
        .document(getCurrentUserId())
        .get()
        .addOnSuccessListener{ document ->
            val loggedInUser = document.toObject(User::class.java)
            if (loggedInUser != null) {
                when {
                    activity is LoginActivity -> {
                        activity.signInSuccess(loggedInUser)
                    }
                    activity is MainActivity -> {
                        activity.updateNavigationUserDetails(loggedInUser)
                    }
                    fragment is ProfileFragment -> {
                        fragment.setUserDetails(loggedInUser)
                    }
                    fragment is MyCoursesFragment -> {
                        fragment.setUpUserDetails(loggedInUser)
                        fragment.updateNavigationUserDetails(loggedInUser)
                    }
                    fragment is CourseContentFragment -> {
                        fragment.setUserDetails(loggedInUser)
                    }
                }
            }
        }
}
```

Σχήμα Α.2: `loadUserData()`

## Παράρτημα Β: Ερωτηματολόγιο

Στα πλαίσια της αξιολόγησης δημιουργήθηκε ερωτηματολόγιο το οποίο αξιολογεί την εμπειρία του χρήστη χρήστη με το ui. Σκοπός ήταν να δούμε πως αλληλεπιδρά η εφαρμογή σε διάφορες ηλικίες και φύλλα.

Μορφή ερωτηματολογίου

**Ποια είναι η ηλικία σας;**

- Κάτω των 18
- 18–24
- 24 - 35
- 35+

**Ποιο είναι το φύλο σας;**

- Άνδρας
- Γυναίκα
- Άλλο / Δεν επιθυμώ να απαντήσω

**Πόσο εύχρηστη θεωρείτε τη διεπαφή της εφαρμογής;**

- Πολύ εύχρηστη
- Εύχρηστη
- Ουδέτερη
- Δύσκολη
- Πολύ δύσκολη

**Πόσο ικανοποιημένοι είστε από τον σχεδιασμό της εφαρμογής (UI);**

- Πολύ ικανοποιημένος/η
- Ικανοποιημένος/η
- Ουδέτερος/η
- Μη ικανοποιημένος/η
- Καθόλου ικανοποιημένος/η

**Τα χρώματα ήταν ευχάριστα;**

- Ναι, απόλυτα
- Ναι, σε γενικές γραμμές
- Όχι ιδιαίτερα
- Καθόλου

**Βρήκατε εύκολη την πλοήγηση μεταξύ των οθονών/λειτουργιών;**

- Ναι
- Όχι
- Σε κάποιο βαθμό

**Πόσο πιθανό είναι να χρησιμοποιήσετε την εφαρμογή ξανά;**

- Πολύ πιθανό
- Πιθανό
- Απίθανο

- Καθόλου πιθανό

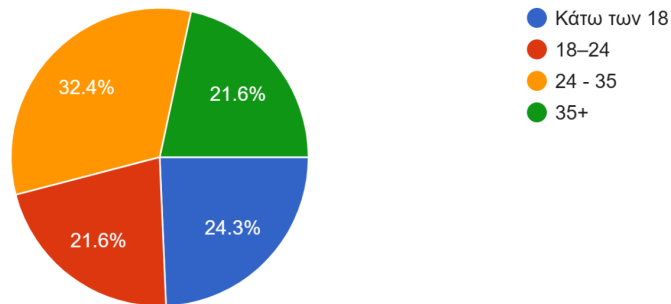
**Θα προτείνατε την εφαρμογή σε άλλους;**

- Ναι
- Όχι
- Ίσως

## Αποτελέσματα

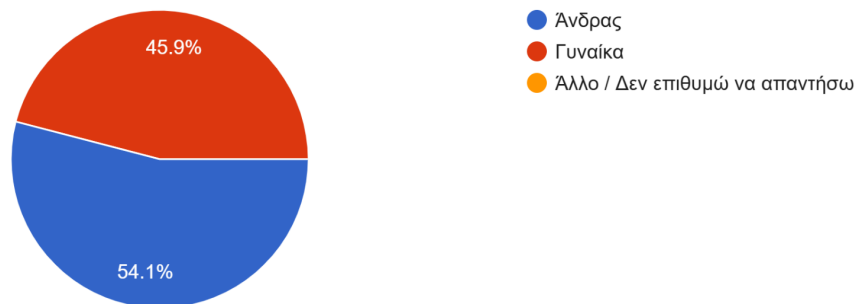
Ποια είναι η ηλικία σας;

37 responses



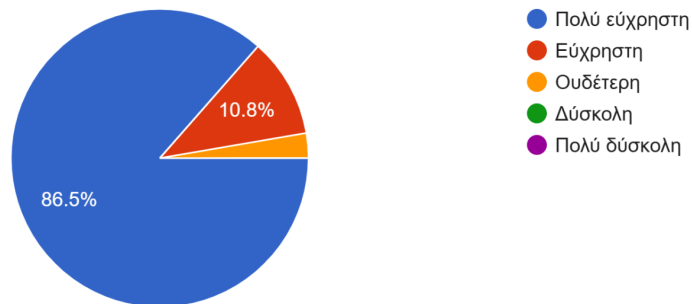
Ποιο είναι το φύλο σας;

37 responses



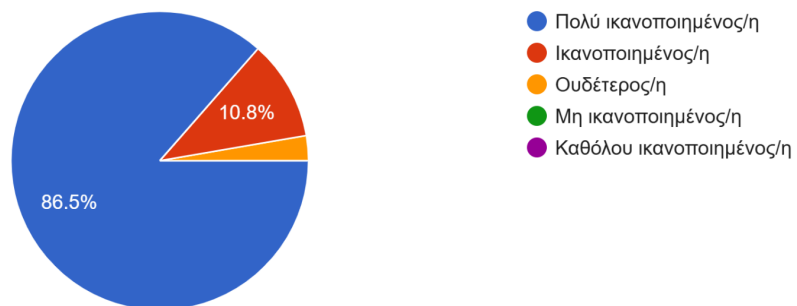
Πόσο εύχρηστη θεωρείτε τη διεπαφή της εφαρμογής;

37 responses



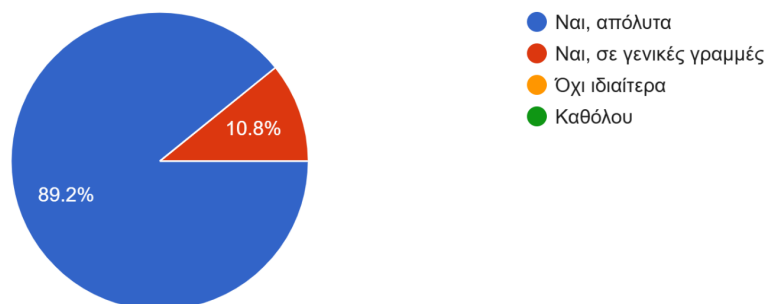
Πόσο ικανοποιημένοι είστε από τον σχεδιασμό της εφαρμογής (UI);

37 responses

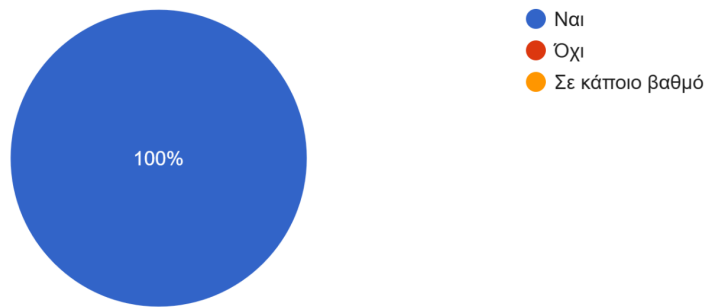


Τα χρώματα ήταν ευχάριστα;

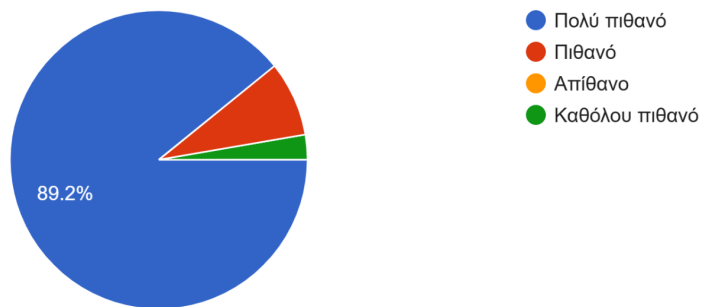
37 responses



Βρήκατε εύκολη την πλοήγηση μεταξύ των οθονών/λειτουργιών;  
37 responses



Πόσο πιθανό είναι να χρησιμοποιήσετε την εφαρμογή ξανά;  
37 responses



Θα προτείνετε την εφαρμογή σε άλλους;  
37 responses

