

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΚΠΑΙΔΕΥΤΙΚΟ ΛΟΓΙΣΜΙΚΟ

ΕΓΧΕΙΡΙΔΙΟ ΑΝΑΛΎΣΗΣ ΚΑΙ ΣΧΕΔΙΑΣΜΟΥ ΕΦΑΡΜΟΓΗΣ

ΟΜΑΔΑ ΑΝΑΠΤΥΞΗΣ:

- ΝΙΚΟΛΑΣ ΠΑΤΕΡΑΣ Π17172
- ΑΝΔΡΕΑΣ ΘΕΟΔΩΡΙΔΗΣ Π17164
- ΒΑΣΙΛΕΙΟΣ ΖΑΡΤΗΛΑΣ ΠΑΠΑΧΑΡΑΛΑΜΠΟΥΣ Π17168

Πειραιάς, Ιούλιος 2021

ПЕРІЕХОМЕNA

ПЕ	PIEXOME	ENA	2	
1.	ΕΙΣΑΓ	ΓΩΓΗ	6	
	1.1. ΘEMA			
	1.2. ΣΤΟΧΟΣ		6	
	1.2.1	1 MULTIPLICATION GAME	6	
	1.2.2	2 ΕΦΑΡΜΟΓΗ ΔΙΑΧΕΙΡΙΣΤΗ	7	
2.	ΓENΙΚ	ΓΕΝΙΚΟΤΕΡΑ ΓΙΑ ΤΙΣ ΕΦΑΡΜΟΓΕΣ		
	2.1.1	1 ΙΔΙΑΙΤΕΡΟΤΗΤΑ ΤΩΝ USER MANUALS	7	
	2.1.2	2 ΙΔΙΑΙΤΕΡΟΤΗΤΑ ΕΚΜΑΘΗΣΗΣ	7	
	2.1.3	3 ΙΔΙΑΙΤΕΡΟΤΗΤΑ ΑΞΙΟΛΟΓΗΣΗΣ ΚΑΙ TEST	7	
	2.1.4	4 ΜΕΘΟΔΟΛΟΓΙΑ ΚΑΤΑΓΡΑΦΗΣ ΣΤΑΤΙΣΤΙΚΩΝ	8	
3.				
	3.1. E	EΦΑΡΜΟΓΗΣ MULTIPLICATION GAME	9	
	3.2. E	ΕΦΑΡΜΟΓΗΣ ΔΙΑΧΕΙΡΙΣΤΗ	14	
4.	ΒΑΣΗ	ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ		
	4.1. E	Εγγραφή χρηστών	16	
	4.2.	Δεδομένα Χρηστών	16	
	4.2.1	1 Πίνακας «tables»	16	
	4.2.2	2 Πίνακας «user_logs»	16	
	4.2.3	3 Πίνακας «users»	17	
5.	ЕПЕΞ	ΉΓΗΣΗ ΚΩΔΙΚΑ	18	
	5.1. N	MULTIPLICATION GAME	18	
	5.1.1	1 \activities	18	
	5.1	1.1.1 BaseActivity.kt	18	

5.1.1.2	ForgotPasswordActivity.kt	18
5.1.1.3	LearnActivity.kt	19
5.1.1.4	MainMenuActivity.kt	19
5.1.1.5	ProfileDetailsActivity.kt	19
5.1.1.6	SignInActivity.kt	19
5.1.1.7	SignUpActivity.kt	20
5.1.1.8	SplashActivity.kt	20
5.1.1.9	TableListActivity.kt	20
5.1.1.10	TableResultActivity.kt	20
5.1.1.11	EditProfileDetailsActivity.kt	21
5.1.1.12	UserLogsListActivity.kt	21
5.1.1.13	TestActivity.kt	22
5.1.2 \d	database	23
5.1.2.1	FirestoreHelper.kt	23
5.1.3 \f	ragments	24
5.1.3.1	LearnSectionFirstFragment.java	24
5.1.3.2	LearnSectionFourthFragment.java	24
5.1.3.3	LearnSectionSecondFragment.java	24
5.1.3.4	LearnSectionThirdFragment.java	25
5.1.4 \n	models	26
5.1.4.1	UserLog.kt	26
5.1.4.2	User.kt	26
5.1.4.3	MultiplicationTable.kt	26
5.1.4.4	Warning.kt	26
5.1.5 \u	ıtils	27
5.1.5.1	Constants.kt	27
5.1.5.2	CustomDialog.kt	27

5.1.5.3	GlideLoader.kt	27
5.1.5.4	SnackBarErrorClass.kt	27
5.1.5.5	SnackBarErrorView.kt	27
5.1.5.6	SnackBarSuccessClass.kt	28
5.1.5.7	SnackBarSuccessView.kt	28
5.1.5.8	Utils.kt	28
5.1.6 \a	ndapters	29
5.1.6.1	TableListAdapter.kt	29
5.1.6.2	ViewPageLearnAdapter.kt	29
5.1.6.3	UserLogListAdapter.kt	29
5.1.7 \e	extensions	30
5.1.7.1	ViewExtensions.kt	30
5.2. ЕФАР	ΜΟΓΗ ΔΙΑΧΕΙΡΙΣΤΗ	31
5.2.1 \a	activities	31
5.2.1.1	BaseActivity.kt	31
5.2.1.2	ForgotPasswordActivity.kt	31
5.2.1.3	MainMenuActivity.kt	31
5.2.1.4	SignInActivity.kt	32
5.2.1.5	SplashActivity.kt	32
5.2.1.6	UserDetailsActivity.kt	32
5.2.1.7	UserListActivity.kt	32
5.2.1.8	UserLogsListActivity.kt	33
5.2.1.9	ProfileDetailsActivity.kt	33
5.2.2 \a	ndapters	34
5.2.2.1	UsersListAdapter.kt	34
5.2.2.2	UsersLogsListAdapter.kt	34
5.2.3 \c	database	34

	5.2.3.1	FirestoreHelper.kt	34
	5.2.4 \	extension	35
	5.2.4.1	ViewExtensions.kt	35
	5.2.5 \r	nodels	35
	5.2.5.1	Log.kt	35
	5.2.5.2	User.kt	35
	5.2.6 \u	utils	36
	5.2.6.1	CustomDialog.kt	36
	5.2.6.2	SnackBarErrorClass.kt	36
	5.2.6.3	SnackBarErrorView.kt	36
	5.2.6.4	SnackBarSuccessClass.kt	36
	5.2.6.5	SnackBarSuccessView.kt	37
	5.2.6.6	Utils.kt	37
	5.2.6.7	Constants.kt	37
6.	ВІВЛІОГРАФІА		38
7	ΒΙΒΛΙΟΘΗΚΕΣ & ΕΡΓΑΛΕΙΑ		42

1. ΕΙΣΑΓΩΓΗ

1.1. **QEMA**

Αρχικά, έχουμε επιλέξει το <u>θέμα 2</u> («Εκπαιδευτικό παιχνίδι για την εκμάθηση της προπαίδειας των Μαθηματικών») για την υλοποίηση της τελικής προγραμματιστικής εργασίας του μαθήματος.

Επιπρόσθετα, έχουμε δημιουργήσει δύο διαφορετικά projects σε Android που είναι προγραμματισμένα σε kotlin, μια εφαρμογή του χρήστη και μία για τον διαχειριστή στην οποία μπορεί να επεξεργαστεί τα δεδομένα της εφαρμογής ανάλογα.

Η βάση δεδομένων που χρησιμοποιήθηκε είναι η FireStore Cloud (NoSQL).

To **source code** του project είναι διαθέσιμο στο **GitHub** αφού χρησιμοποιήσαμε version control.

- https://github.com/p17172p17168p17164/Multiplication-Game-Android
- https://github.com/p17172p17168p17164/Multiplication-Game-Admin-Android-App

1.2. ΣΤΟΧΟΣ

1.2.1 MULTIPLICATION GAME

Ο στόχος μας κατά την σχεδίαση ήταν η δημιουργία ενός φιλικού UI το οποίο περιέχει ήχους, μουσική, εφέ έτσι ώστε ο χρήστης να νιώθει πιο πολύ ενδιαφέρον όταν χρησιμοποιεί την εφαρμογή (αφού θεωρείται εκπαιδευτικό παιχνίδι). Επίσης, να γίνεται καλή διάγνωση λαθών για να μπορούμε να βελτιώσουμε τον χρήστη ανάλογα. Με λίγα λόγια, στόχος μας είναι η υλοποίηση μιας εφαρμογής που περιλαμβάνει τρόπους παρουσίασης διδακτικού υλικού για να γίνεται το θέμα κατανοητό και να μπορεί να απομνημονευθεί από τους μαθητές και να εμπεδώσουμε την ύλη μέσω ασκήσεων.

 $EI\Sigma A \Gamma \Omega \Gamma H$ 6

1.2.2 ΕΦΑΡΜΟΓΗ ΔΙΑΧΕΙΡΙΣΤΗ

Στόχος της εφαρμογής του διαχειριστή είναι η επιτήρηση των προόδων του κάθε μαθητή αλλά και η επεξεργασία των μαθητών του.

2. ΓΕΝΙΚΟΤΕΡΑ ΓΙΑ ΤΙΣ ΕΦΑΡΜΟΓΕΣ

2.1.1 I Δ IAITEPOTHTA T Ω N USER MANUALS

Έχουμε χρησιμοποιήσει μία διαφορετική προσέγγιση στο εγχειρίδιο χρήσης του μαθητή έτσι ώστε να είναι ευκολότερο και πιο κατανοητό για το μαθητή. Για παράδειγμα έχουμε τοποθετήσει τόξα στις εικόνες αναγράφοντας κάθε φορά και το αντίστοιχο βήμα όπου είχαμε περιγράψει με λόγια σε κάθε πιθανή ερώτηση που θα έκανε ο χρήστης! Επίσης προσθέσαμε και μερικές συμβουλές στο εγχειρίδιο του μαθητή έτσι ώστε να τον βοηθήσουμε με την επαφή που θα έχει με την εφαρμογή! Για παράδειγμα έχουμε προσθέσει συμβουλές για τον κωδικό πρόσβασης.

2.1.2 ΙΔΙΑΙΤΕΡΟΤΗΤΑ ΕΚΜΑΘΗΣΗΣ

Παρέχουμε στον μαθητή μέσω των 4 sections που έχουμε στο αρχείο LearnActivity.kt τρόπους, «κόλπα» και tips για να μπορεί να θυμάται πως να λύνει πολλαπλασιασμούς κι όχι να τους θυμάται απέξω.

2.1.3 ΙΔΙΑΙΤΕΡΟΤΗΤΑ ΑΞΙΟΛΟΓΗΣΗΣ ΚΑΙ ΤΕST

Ο χρήστης μπορεί να επιλέξει σε ποιον πίνακα θέλει να αξιολογηθεί, έτσι μπορεί να κάνει μεμονωμένη εξάσκηση για καλύτερη εμπέδωση του κάθε πίνακα. Σε κάθε εξίσωση που λύνει σωστά του επικροτούμε με ένα μήνυμα έτσι ώστε να τον ενθαρρύνουμε να συνεχίσει ακάθεκτος. Αν όμως λύσει λάθος την εξίσωση δεν τον τιμωρούμε αλλά του εμφανίζουμε ένα αισιόδοξο μήνυμα ότι την επόμενη φορά θα τα καταφέρει, έτσι δεν τον απογοητεύουμε κι ούτε τον αποθαρρύνουμε και περνάμε το μήνυμα ότι μπορεί να μάθει μέσα από τα λάθη του αρκεί να προσπαθεί και να εξασκείται.

 $EI\Sigma A \Gamma \Omega \Gamma H$ 7

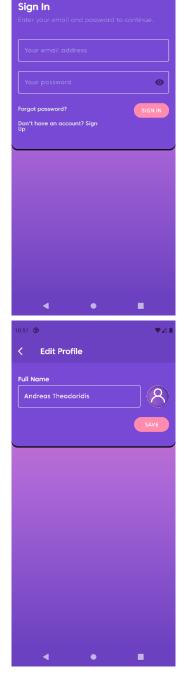
2.1.4 ΜΕΘΟΔΟΛΟΓΙΑ ΚΑΤΑΓΡΑΦΗΣ ΣΤΑΤΙΣΤΙΚΩΝ

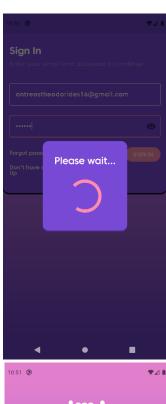
Όταν ο χρήστης λύνει την κάθε εξίσωση σημειώνουμε στην βάση δεδομένων μας αν την έλυσε σωστά, λανθασμένα ή την έχει αγνοήσει. Αποθηκεύουμε όλες τις εξισώσεις για κάθε χρήστη. Όταν ο χρήστης στην αρχική οθόνη της εφαρμογής μας πατήσει την επιλογή «START TEST» τότε ξεκινάει να λύνει ξανά εξισώσεις πολλαπλασιασμού. Η ιδιαιτερότητα που έχουν τα test αυτά είναι ότι εμφανίζουμε στον χρήστη τις εξισώσεις που έχει απαντήσει λανθασμένα είτε έχει αγνοήσει (skip), εάν απαντήσει σωστά αυτή την φορά τότε αποθηκεύουμε την εξίσωση ως σωστά απαντημένη και δεν θα την εμφανίζει πλέον. Να σημειώσουμε ότι δεν παίρνουμε μόνο τις λανθασμένες εξισώσεις αλλά έχουμε δημιουργήσει ένα αλγόριθμό όπου κάθε φορά βάση πιθανοτήτων θα επιλέξει αν θα πάρει λανθασμένες απαντήσεις εξισώσεων του χρήστη ή αν θε δημιουργήσει μια εξίσωση random [πάντα στα πλαίσια της προπαίδειας] για να λύσει ο χρήστης.

 $EI\Sigma A \Gamma \Omega \Gamma H$ 8

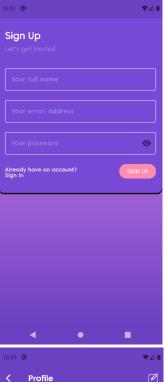
3. ΦΩΤΟΓΡΑΦΙΕΣ

3.1. ΕΦΑΡΜΟΓΗΣ MULTIPLICATION GAME

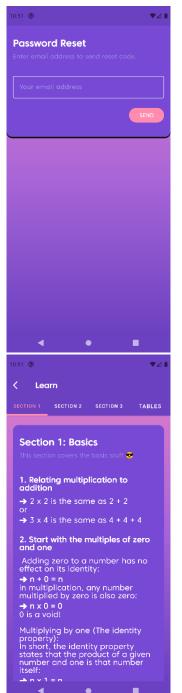


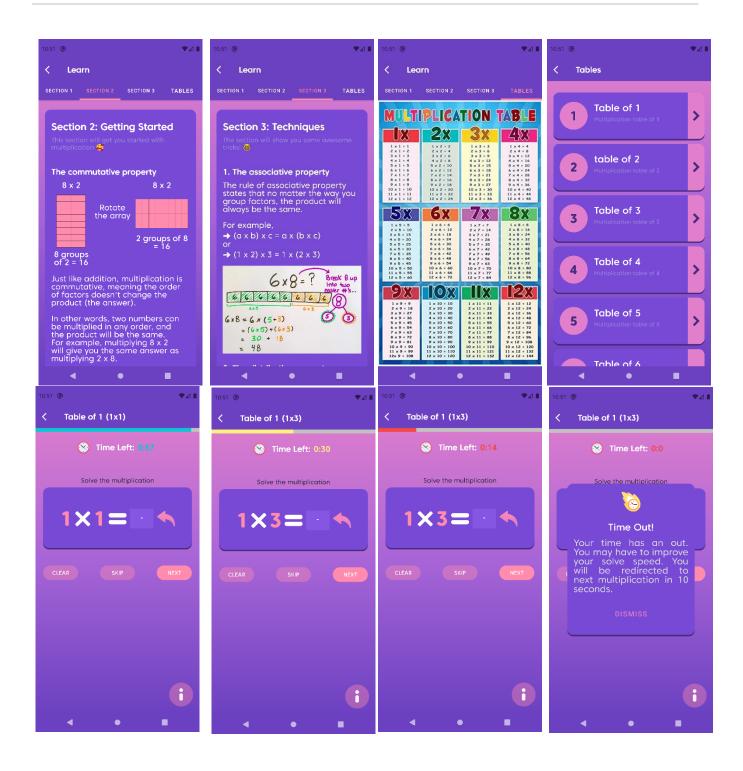


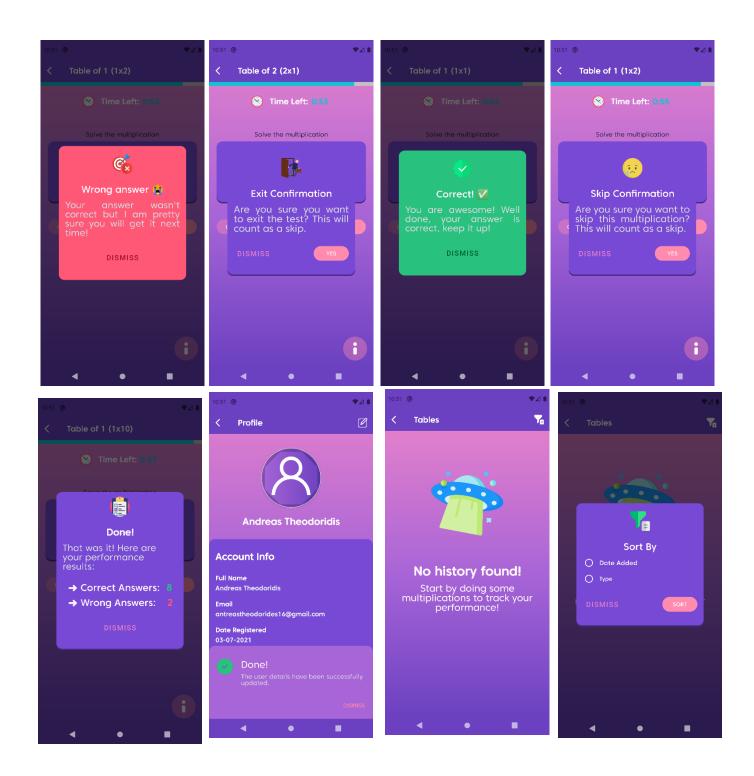












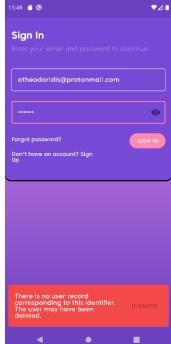






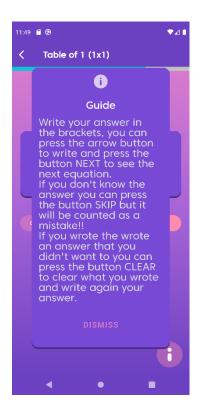




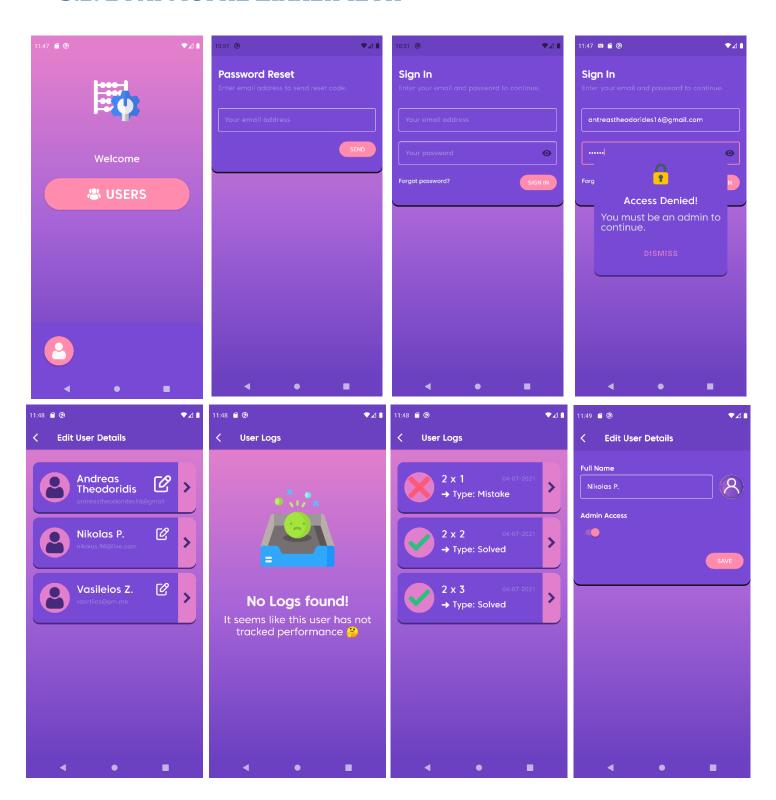


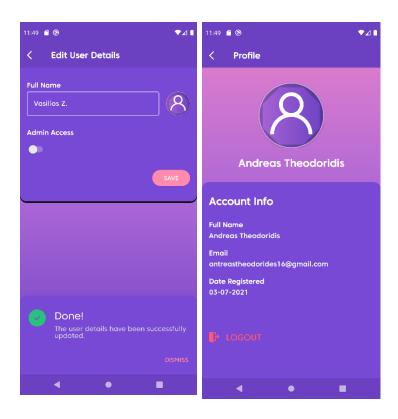






3.2. ΕΦΑΡΜΟΓΗΣ ΔΙΑΧΕΙΡΙΣΤΗ





4. ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

4.1. Εγγραφή χρηστών

Οι χρήστες κάνουν εγγραφή με το email τους και δίνοντας τον δικό τους κωδικό πρόσβασης και τους καταγράφουμε στο Authentication της Firebase. Να σημειώσουμε ότι ο διαχειριστής δεν μπορεί να κάνει εγγραφή αλλά καθορίζεται απευθείας από εμάς.

4.2. Δεδομένα Χρηστών

Χρησιμοποιήσαμε Firestore Database της. Μέσα σε στην βάση έχουμε τους εξής πίνακες: tables, user_logs, users.

4.2.1 Πίνακας «tables»

Στον πίνακα tables έχουμε απλά αποθηκεύσει τους πίνακες πολλαπλασιασμού.

4.2.2 Πίνακας «user_logs»

Στον πίνακα αυτό αποθηκεύουμε την κάθε εξίσωση όπου είτε έχει λυθεί σωστά ή λανθασμένα, είτε έχει αγνοηθεί. Κάθε εξίσωση (documents) έχει τα εξής πεδία:

- *dateAdded* για να ξέρουμε πότε λύθηκε η εξίσωση.
- numFirst ο πρώτος αριθμός του πολλαπλασιασμού.
- numSecond ο δεύτερος αριθμός του πολλαπλασιασμού.
- random πεδίο που μας βοηθά για τον αλγόριθμο που υλοποιήσαμε για την εμφάνιση δυναμικών test.
- **type** για να αποθηκεύουμε αν έκανε skip ή αν απάντησε σωστά ή λανθασμένα ο χρήστης.
- **userId** για να αποθηκεύουμε ποιος χρήστης έλυσε την εξίσωση.

 $BA\Sigma H \Delta E\Delta OMEN\Omega N$ 16

4.2.3 Πίνακας «users»

Στον πίνακα αυτό αποθηκεύουμε τις πληροφορίες για κάθε χρήστη μας.

- adminfalse για να αποθηκεύουμε αν είναι admin ο χρήστης.
- dateRegistered για να αποθηκεύουμε πότε έκανε εγγραφή ο χρήστης.
- email για να αποθηκεύουμε το email του χρήστη.
- **fullName** για να αποθηκεύουμε το όνομα του χρήστη.
- userId για να αποθηκεύουμε το id στο οποίο αντιστοιχούν οι πληροφορίες.

 $BA\Sigma H \Delta E\Delta OMEN\Omega N$ 17

5. ΕΠΕΞΗΓΗΣΗ ΚΩΔΙΚΑ

5.1. MULTIPLICATION GAME

■ 5.1.1 \activities

Στο package Activities βρίσκονται όλα τα activities της εφαρμογής.

5.1.1.1 BaseActivity.kt

Αυτό το αρχείο χρησιμοποιείται για τον καθορισμό των λειτουργιών της εφαρμογής. Υπάρχουν συναρτήσεις με τις οποίες εμφανίζουμε και κρύβουμε Progress Dialogs [βλέπε fun showProgressDialog()]. Επίσης υπάρχουν συναρτήσεις για να μεταφερόμαστε στα activities που υπάρχουν στην εφαρμογή μας [βλέπε fun goToMainActivity()]. Να σημειώσουμε ότι το αρχείο κληρονομείται από πολλά αρχεία στην εφαρμογή μας έτσι έστω να μπορούμε να χρησιμοποιούμε τις λειτουργείες του.

5.1.1.2 ForgotPasswordActivity.kt

Αυτό το αρχείο χρησιμοποιείται όταν ο χρήστης ξεχάσει τον κωδικό του. Μέσω της συνάρτησης fun resetPassword() ο χρήστης μπορεί να αλλάξει τον κωδικό του. Επίσης γίνονται οι ανάλογοι έλεγχοι [βλέπε fun validateFields()] έτσι ώστε ο χρήστης να συμπληρώσει ορθά τα πεδία που του ζητούνται και παράλληλα βοηθώντας τον εφόσον εμφανίζονται και τα ανάλογα μηνύματα όταν εντοπίσουμε ότι έχει συμπληρώσει κάτι λανθασμένα ή κάτι μη αποδεκτό από το σύστημα μας.



5.1.1.3 LearnActivity.kt

Αυτό το αρχείο χρησιμοποιείται για την εκμάθηση της προπαίδειας στο χρήστη. Υπάρχει ένα action bar [βλέπε fun setupActionBar()] όπου τοποθετήσαμε τέσσερα [4] sections όπου θα υπάρχει η ύλη της προπαίδειας. Έχουμε δώσει στον χρήστη διάφορες συμβουλές για να θυμάται τους πίνακες πολλαπλασιασμού.



5.1.1.4 MainMenuActivity.kt

Αυτό το αρχείο χρησιμοποιείται για την πλοήγηση του χρήστη στην εφαρμογή. Στο activity αυτό ο χρήστης βλέπει κουμπιά όπου ανάλογα με το πιο θα αλληλεπιδράσει θα μεταφερθεί στο αντίστοιχο activity [βλέπε fun setupUI()]. Η μεταφορά του χρήστη γίνεται με την βοήθεια [εφόσον το κληρονομεί το παρών αρχείο] του αρχείο BaseActivity.kt όπου προαναφέραμε τις λειτουργίες του πιο πάνω.

ProfileDetailsActivity.kt **5.1.1.5**

αρχείο χρησιμοποιείται για να εμφανίζουμε [βλέπε fun successProfileDetailsFromFirestore(), fun loadProfileDetails()] στον χρήστη της πληροφορίες του λογαριασμού του [ονοματεπώνυμο, email, ημερομηνία εγγραφής]. Μέσω αυτού του activity μπορεί να μεταβεί στο activity editProfileDetailsActivity.kt για να επεξεργαστή τα στοιχεία του [μπορεί να επεξεργαστεί μόνο το ονοματεπώνυμο του]. Επίσης όταν βρεθεί στο activity αυτό μπορεί να κάνει και αποσύνδεση [βλέπε fun setupUI()].



5.1.1.6 SignInActivity.kt

Αυτό το αρχείο χρησιμοποιείται για την σύνδεση του χρήστη. Κατά την προσπάθεια του χρήστη να συνδεθεί μέσω της fun signInUser() γίνονται οι ανάλογοι έλεγχοι [βλέπε fun validateFields()] για να επιβεβαιώσουμε τα στοιχεία του χρήστη. Αν επιβεβαιωθούν τα στοιχεία του τότε τον μεταφέρουμε στο κεντρικό activity [βλέπε fun userLoggedInSuccess()].



5.1.1.7 SignUpActivity.kt

Αυτό το αρχείο χρησιμοποιείται για την εγγραφή του χρήστη. Μέσω της fun validateRegisterDetails() ελέγχουμε αν συμπλήρωσε ο χρήστης τα απαραίτητα πεδία ορθά. Μετέπειτα εφόσον είναι ορθά τα στοιχεία μέσω της fun registerUser() αποθηκεύουμε τον χρήστη στην βάση και αμέσως περνάει στο επόμενο activity [βλέπε userRegistrationSuccess()].



5.1.1.8 SplashActivity.kt

Αυτό το αρχείο χρησιμοποιείται κατά την είσοδο του χρήστη στην εφαρμογή. Έχουμε δημιουργήσει ένα load screen όπου κατά την διάρκεια φόρτωσης την εφαρμογής [βλέπε hideSystemUI()] ελέγχουμε αν ο χρήστης έχει συνδεθεί τουλάχιστον μία φορά στην εφαρμογή και τον μεταφέρουμε στο κεντρικό activity αλλιώς αν δεν έχει συνδεθεί (ή έχει αποσυνδεθεί) τότε τον μεταφέρουμε στο activity SignInActivity.kt για να συνδεθεί [βλέπε moveToNextActivity()].

5.1.1.9 TableListActivity.kt

Αυτό το αρχείο χρησιμοποιείται για ελέγχουμε ότι φορτώθηκαν όλα τα δεδομένα μας επιτυχώς από την βάση μας [βλέπε fun loadTables()]. Εφόσον η διαδικασία έγινε επιτυχώς τότε εμφανίζουμε τους πίνακες στον χρήστη [βλέπε fun successTablesList()].



5.1.1.10 TableResultActivity.kt

Αυτό το αρχείο χρησιμοποιείται για να διεκπεραιώνεται η διαδικασία αξιολόγησης του χρήστη. Στον χρήστη εμφανίζουμε κάθε φορά μία εξίσωση πολλαπλασιασμού [οι εξισώσεις δημιουργούνται μέσω της βάσεις μας, βλέπε setEquationVariables()] και καλείται να την λύσει. Στον χρήστη δίνουμε 3 επιλογές [clear, skip, next] στο μέσω του activity.

Κουμπί clear: Πατώντας το κουμπί clear καθαρίζουμε το πεδίο όπου τοποθετείτε η απάντηση.

Κουμπί skip: Πατώντας το κουμπί skip ο χρήστης προχωρά στην επόμενη πράξη χωρίς να χρειαστεί να απαντήσει. Φυσικά πατώντας το κουμπί skip περνούμε στα στατιστικά μας την αδυναμία του χρήστη στο να απαντήσει.

Κουμπί next: Πατώντας το κουμπί next ελέγχουμε αν έχει δοθεί απάντηση [βλέπε fun validateFields()] αν έχει δοθεί ενημερώνουμε τον χρήστη αν απάντησε ορθά ή όχι και προχωράμε στην εμφάνιση της επόμενης εξίσωσης [βλέπε fun goToNextEquation()].

Χρόνος: Ο χρήστης έχει στην διάθεσή του ένα [1] λεπτό για να λύσει κάθε εξίσωση [βλέπε fun startTimer(), fun pauseTimer()].

Progress Bar: Στον χρήστη εμφανίζουμε και ένα progress bar [βλέπε fun onTick()] το οποίο όσο μειώνεται ο χρόνος αλλάζει και το χρώμα του. Ξεκινάει με πράσινο, στην συνέχεια γίνεται κίτρινο και όταν λιγοστέψει ο χρόνος αρκετά γίνεται κόκκινο.

5.1.1.11 EditProfileDetailsActivity.kt

Αυτό αρχείο χρησιμοποιείται επεξεργάζεται το νια να ο χρήστης το ονοματεπώνυμό του [βλέπε fun updateUserProfileDetails(), fun userProfileUpdateSuccess()]. Φυσικά γίνονται ξανά έλεγχοι για να ελέγξουμε ότι ο χρήστης έδωσε σωστά το ονοματεπώνυμο του [βλέπε validateFields()], εφόσον όλα γίνουν όπως πρέπει του εμφανίζουμε το αντίστοιχο μήνυμα.



5.1.1.12 UserLogsListActivity.kt

Αυτό το αρχείο χρησιμοποιείται για να εμφανίζουμε την πρόοδο του χρήστη. Κάθε item στον recycler είναι και μία εξίσωση που έχει λύσει, για κάθε εξίσωση εμφανίζεται η πληροφορία αν το έχει λύσει ορθά, λανθασμένα ή αν έχει αγνοήσει την εξίσωση.



5.1.1.13 TestActivity.kt

Αυτό το αρχείο χρησιμοποιείται για να εμφανίζουμε test στον χρήστη ανάλογα με τις αδυναμίες του. Στην πραγματικότητα δημιουργούμε δυναμικά test κάθε φορά με βάση των λανθασμένων εξισώσεων που είχε απαντήσει στους πίνακες πολλαπλασιασμού [βλέπε else της fun initEquationNumbers(), fun setEquation()]. Αν απαντήσει σωστά αυτή την φορά του επικυρώνουμε ως σωστή την εξίσωση και τότε δεν θα εμφανίζεται στα επόμενα test που θα δημιουργήσουμε. Επίσης μεριμνήσαμε και εμφανίζουμε τυχαίες εξισώσεις αν δεν υπάρχουν αρκετές για το test μας μέσω ενός αλγορίθμου που δημιουργήσαμε [βλέπε fun initEquationNumbers() , fun loadLogs()]. Επίσης όταν ο χρήστης θέλει να προχωρήσει στην επόμενη εξίσωση ελέγχουμε την απάντηση του και αποθηκεύουμε την απάντηση του στην βάση [βλέπε fun goToNextEquation() , fun checkAnswer()]. Να σημειώσουμε ότι ο χρόνος του χρήστη που έχει για να λύσει τις εξισώσεις διαφοροποιείται. Δηλαδή δεν έχει 1 λεπτό για κάθε εξίσωση του εκάστοτε πίνακα αλλά έχει 1 λεπτό να λύσει όσο περισσότερες εξισώσεις μπορεί [βλέπε override fun onTick()].

■ 5.1.2 \database

Σε αυτό τον φάκελο έχουμε το αρχείο που είναι υπεύθυνο για την επικοινωνία της εφαρμογής με την βάση δεδομένων μας.

5.1.2.1 FirestoreHelper.kt

Το αρχείο αυτό χρησιμοποιείται για την επικοινωνία της εφαρμογής μας με την βάση δεδομένων. Με την fun getCurrentUserID παίρνουμε το User ID του χρήστη που είναι συνδεδεμένος στην εφαρμογή μας. Με την fun getUserDetails παίρνουμε τα στοιχεία του χρήστη που είναι συνδεμένος στην εφαρμογή από την βάση δεδομένων. Με την fun getUserList παίρνουμε μια λίστα με όλους του χρήστες που συνδέθηκαν ή δημιούργησαν λογαριασμό στην εφαρμογή μας, παίρνουμε τα στοιχεία τους και τα εμφανίζουμε στην εφαρμογή. Με την fun UpdateUserProfileData ενημερώνουμε στην βάση δεδομένων τα νέα στοιχεία του χρήστη.

■ 5.1.3 \fragments

Το Fragment αντιπροσωπεύει ένα επαναχρησιμοποιήσιμο τμήμα του UI της εφαρμογής. Ένα fragment καθορίζει και διαχειρίζεται τη δική του διάταξη, έχει τον δικό του κύκλο ζωής και μπορεί να χειριστεί τα δικά του συμβάντα εισόδου. Επίσης τα fragment πρέπει να φιλοξενούνται από μια δραστηριότητα ή άλλο κομμάτι γι΄ αυτό και το κάθε fragment καλείτε μέσω των αρχείων που βρίσκονται στον φάκελο activities.

5.1.3.1 LearnSectionFirstFragment.java

Το αρχείο αυτό χρησιμοποιείται για να εμφανίσει στον χρήστη την πρώτη ενότητα διδασκαλίας για την μαθηματική προπαίδεια. Στην πρώτη ενότητα παρουσιάζουμε στον χρήστη πρώτα παραδείγματα και συμβουλές για να λύσει μια εξίσωση πολλαπλασιασμού με την σκεπτική της πρόσθεσης και μετά του γράφουμε κάποιες συμβουλές όσον αφορά τον πίνακα του μηδέν [0] και του ένα [1].

5.1.3.2 LearnSectionFourthFragment.java

Το αρχείο αυτό χρησιμοποιείται για να εμφανίσει στον χρήστη την τέταρτη ενότητα διδασκαλίας για την μαθηματική προπαίδεια. Είναι η τελευταία ενότητα διδασκαλίας και του παρουσιάζουμε όλους τους πίνακες της προπαίδειας και τις λύσεις τους.

5.1.3.3 LearnSectionSecondFragment.java

Το αρχείο αυτό χρησιμοποιείται για να εμφανίσει στον χρήστη την δεύτερη ενότητα διδασκαλίας για την μαθηματική προπαίδεια. Στην δεύτερη ενότητα διδασκαλίας μαθαίνουμε στον χρήστη την αντιμεταθετική ιδιότητα και παραδείγματα της.



5.1.3.4 LearnSectionThirdFragment.java

Το αρχείο αυτό χρησιμοποιείται για να εμφανίσει στον χρήστη την τρίτη ενότητα διδασκαλίας για την μαθηματική προπαίδεια. Στην δεύτερη ενότητα διδασκαλίας εξηγούμε στον χρήστη πως δουλεύει η προσαιτεριστική ιδιότητα και παραδείγματα της και μετά του εξηγούμε την λειτουργία της επιμεριστικής ιδιότητας και παραδείγματα της.

■ 5.1.4 \models

Αυτό το package χρησιμοποιείται για να μπορούμε να διατηρούμε τα δεδομένα μας «ενεργά». Έτσι ανά πάσα στιγμή έχουμε άμεση πρόσβαση στα δεδομένα χωρίς να χρειάζεται να τα ξανά φορτώνουμε από την βάση δεδομένων. Αρά σε τυπικές λειτουργίες και λειτουργείες που χρησιμοποιούμε συχνά έχουμε τα δεδομένα μας άμεσα και γρήγορα στην διάθεση μας.

Στο data model αυτό αποθηκεύουμε τα στατιστικά των επιδόσεων του εκάστοτε χρήστη έτσι ώστε να έχουμε άμεση πρόσβαση σε αυτά σε όλη την εφαρμογή.

■ 5.1.4.2 User.kt

Στο data model αυτό αποθηκεύουμε τα προσωπικά δεδομένα του εκάστοτε χρήστη έτσι ώστε να έχουμε άμεση πρόσβαση σε αυτά σε όλη την εφαρμογή.

i 5.1.4.3 MultiplicationTable.kt

Στο data model αυτό αποθηκεύουμε τους πίνακες πολλαπλασιασμού έτσι ώστε να έχουμε άμεση πρόσβαση σε αυτά σε όλη την εφαρμογή.

5.1.4.4 Warning.kt

Στο data model αυτό αποθηκεύουμε τις προειδοποιήσεις έτσι ώστε να έχουμε άμεση πρόσβαση σε αυτά σε όλη την εφαρμογή.

■ 5.1.5 \utils

Όλα τα αρχεία που βρίσκονται στο package αυτό είναι βοηθητικά προγράμματα αρχεία της εφαρμογής.

☐ 5.1.5.1 Constants.kt

Αυτό το αρχείο χρησιμοποιείται για να αποθηκεύουμε όλες τις σταθερές που χρησιμοποιούμε στην εφαρμογή.

5.1.5.2 CustomDialog.kt

Στο αρχείο αυτό έχουμε δημιουργήσει όλα τα dialog που θα χρησιμοποιήσουμε στην εφαρμογή.

☐ 5.1.5.3 GlideLoader.kt

Στο αρχείο αυτό το χρησιμοποιούμε για να φορτώσουμε ευκολότερα αρχεία από το τηλέφωνο του χρήστη μέσω της βοήθειας της αντίστοιχης εφαρμογής.

3.1.5.4 SnackBarErrorClass.kt

Στο αρχείο αυτό έχουμε δημιουργήσει το snack bar error που θα το χρησιμοποιήσουμε στην εφαρμογή όταν θέλουμε να ενημερώσουμε τον χρήστη για κάτι το οποίο απέτυχε.

Στο αρχείο αυτό έχουμε δημιουργήσει το view του error snack bar, δηλαδή τον τρόπο εμφάνισης του στην εφαρμογή.

5.1.5.6 SnackBarSuccessClass.kt

Στο αρχείο αυτό έχουμε δημιουργήσει το snack bar success που θα το χρησιμοποιήσουμε στην εφαρμογή όταν θέλουμε να ενημερώσουμε τον χρήστη για κάτι το οποίο έχει γίνει με επιτυχία.

5.1.5.7 SnackBarSuccessView.kt

Στο αρχείο αυτό έχουμε δημιουργήσει το view του success snack bar, δηλαδή τον τρόπο εμφάνισης του στην εφαρμογή.

■ 5.1.5.8 Utils.kt

Στο αρχείο αυτό έχουμε δημιουργήσει γενικότερες συναρτήσεις στις οποίες χρησιμοποιούμε στην εφαρμογής μας ανάλογα με το τι χρειαζόμαστε.

■5.1.6 \adapters

Ένας Adapter λειτουργεί ως γέφυρα μεταξύ ενός AdapterView και την σειρά από κάποια δεδομένων που θέλουμε να εμφανίσουμε. Ο adapter μας παρέχει πρόσβαση στα δεδομένα στην εκάστοτε περίπτωση και είναι επίσης υπεύθυνος για τη δημιουργεί μία προβολή για κάθε στοιχείο που θέλουμε να εμφανίζεται στον χρήστη.

5.1.6.1 TableListAdapter.kt

Με τον adapter αυτόν εμφανίζουμε στον χρήστη μία σειρά από επιλογές στο TableListActivity.kt. Η κάθε επιλογή είναι ένα test πολλαπλασιασμού για την προπαίδεια .

5.1.6.2 ViewPageLearnAdapter.kt

Με τον adapter αυτόν εμφανίζουμε στον χρήστη μία σειρά από sections με την χρήση των fragments στο LearnActivity.kt.

5.1.6.3 UserLogListAdapter.kt

Με τον adapter αυτόν εμφανίζουμε στον χρήστη την πρόοδο του. Κάθε item είναι μία εξίσωση.

■ 5.1.7 \extensions

Στο package αυτό έχουμε αρχεία όπου είναι βοηθητικά για άλλα αρχεία.

5.1.7.1 ViewExtensions.kt

Αυτό το αρχείο χρησιμοποιείται για την εμφάνιση των snack bars.

5.2. ΕΦΑΡΜΟΓΗ ΔΙΑΧΕΙΡΙΣΤΗ

■ 5.2.1 \activities

Στο package Activities βρίσκονται όλα τα activities της εφαρμογής.

5.2.1.1 BaseActivity.kt

Αυτό το αρχείο χρησιμοποιείται για τον καθορισμό των λειτουργιών της εφαρμογής. Υπάρχουν συναρτήσεις με τις οποίες εμφανίζουμε και κρύβουμε Progress Dialogs [βλέπε fun showProgressDialog()]. Επίσης υπάρχουν συναρτήσεις για να μεταφερόμαστε στα activities που υπάρχουν στην εφαρμογή μας [βλέπε fun goToMainActivity()]. Να σημειώσουμε ότι το αρχείο κληρονομείται από πολλά αρχεία στην εφαρμογή μας έτσι έστω να μπορούμε να χρησιμοποιούμε τις λειτουργείες του.

5.2.1.2 ForgotPasswordActivity.kt

Αυτό το αρχείο χρησιμοποιείται όταν ο χρήστης ξεχάσει τον κωδικό του. Μέσω της συνάρτησης fun resetPassword() ο χρήστης μπορεί να αλλάξει τον κωδικό του. Επίσης γίνονται οι ανάλογοι έλεγχοι [βλέπε fun validateFields()] έτσι ώστε ο χρήστης να συμπληρώσει ορθά τα πεδία που του ζητούνται και παράλληλα βοηθώντας τον εφόσον εμφανίζονται και τα ανάλογα μηνύματα όταν εντοπίσουμε ότι έχει συμπληρώσει κάτι λανθασμένα ή κάτι μη αποδεκτό από το σύστημα μας.

5.2.1.3 MainMenuActivity.kt

Αυτό το αρχείο χρησιμοποιείται για την πλοήγηση του χρήστη στην εφαρμογή. Στο activity αυτό ο χρήστης βλέπει κουμπιά όπου ανάλογα με το πιο θα αλληλεπιδράσει θα μεταφερθεί στο αντίστοιχο activity [βλέπε fun setupUI()]. Η μεταφορά του χρήστη γίνεται με την βοήθεια [εφόσον το κληρονομεί το παρών αρχείο] του αρχείο BaseActivity.kt όπου προαναφέραμε τις λειτουργίες του πιο πάνω.

5.2.1.4 SignInActivity.kt

Αυτό το αρχείο χρησιμοποιείται για την σύνδεση του χρήστη. Κατά την προσπάθεια του χρήστη να συνδεθεί μέσω της fun signInUser() γίνονται οι ανάλογοι έλεγχοι [βλέπε fun validateFields()] για να επιβεβαιώσουμε τα στοιχεία του χρήστη. Αν επιβεβαιωθούν τα στοιχεία του τότε τον μεταφέρουμε στο κεντρικό activity [βλέπε fun userLoggedInSuccess()].

5.2.1.5 SplashActivity.kt

Αυτό το αρχείο χρησιμοποιείται κατά την είσοδο του χρήστη στην εφαρμογή. Έχουμε δημιουργήσει ένα load screen όπου κατά την διάρκεια φόρτωσης την εφαρμογής [βλέπε hideSystemUI()] ελέγχουμε αν ο χρήστης έχει συνδεθεί τουλάχιστον μία φορά στην εφαρμογή και τον μεταφέρουμε στο κεντρικό activity αλλιώς αν δεν έχει συνδεθεί (ή έχει αποσυνδεθεί) τότε τον μεταφέρουμε στο activity SignInActivity.kt για να συνδεθεί [βλέπε moveToNextActivity()].

5.2.1.6 UserDetailsActivity.kt

Αυτό το αρχείο χρησιμοποιείται όταν επιλέξει κάποιο χρήστη ο διαχειριστής να του εμφανίζει ένα activity στο οποίος θα μπορείς να επεξεργαστεί τα δεδομένα του χρήστη [ονοματεπώνυμο, παραχώρηση δικαιωμάτων διαχειριστή].

5.2.1.7 UserListActivity.kt

Αυτό το αρχείο χρησιμοποιείται για να εμφανίζουμε [βλέπε fun successUsersListFromFirestore()] στον διαχειριστή μία λίστα από τους μαθητές του. Πατώντας πάνω σε ένα από τα item θα βλέπει περισσότερες πληροφορίες.

5.2.1.8 UserLogsListActivity.kt

Αυτό το αρχείο χρησιμοποιείται για να εμφανίζουμε στο διαχειριστή όλες τις απαντήσεις των εξισώσεων. Κάθε item στο recycler view εμφανίζει αν έχει απαντήσει σωστά, λανθασμένα ή έχει αγνοήσει την εξίσωση [βλέπε fun successUserLogsFromFireStore(), fun loadUserLogs()].

5.2.1.9 ProfileDetailsActivity.kt

Αυτό το αρχείο χρησιμοποιείται για να εμφανίζουμε [βλέπε fun successProfileDetailsFromFirestore(), fun loadProfileDetails()] στον χρήστη της πληροφορίες του λογαριασμού του [ονοματεπώνυμο, email, ημερομηνία εγγραφής].

5.2.2 \adapters

Ένας Adapter λειτουργεί ως γέφυρα μεταξύ ενός AdapterView και την σειρά από κάποια δεδομένων που θέλουμε να εμφανίσουμε. Ο adapter μας παρέχει πρόσβαση στα δεδομένα στην εκάστοτε περίπτωση και είναι επίσης υπεύθυνος για τη δημιουργεί μία προβολή για κάθε στοιχείο που θέλουμε να εμφανίζεται στον χρήστη.

5.2.2.1 UsersListAdapter.kt

Με τον adapter αυτόν εμφανίζουμε στον διαχειριστή όλους τους χρήστες που χρησιμοποιούν την εφαρμογή.

5.2.2.2 UsersLogsListAdapter.kt

Με τον adapter αυτόν εμφανίζουμε στον διαχειριστή όλες τι απαντήσεις από τις εξισώσεις του εκάστοτε χρήστη που έχει επιλέξει.

■ 5.2.3 \database

Σε αυτό τον φάκελο έχουμε το αρχείο που είναι υπεύθυνο για την επικοινωνία της εφαρμογής με την βάση δεδομένων μας.

i 5.2.3.1 FirestoreHelper.kt

Το αρχείο αυτό χρησιμοποιείται για την επικοινωνία της εφαρμογής μας με την βάση δεδομένων μας. Με την μέθοδο getCurrentUserID παίρνουμε το User ID του χρήστη που είναι συνδεδεμένος στην εφαρμογή μας. Με την μέθοδο getUserDetails παίρνουμε τα στοιχεία του χρήστη που είναι συνδεμένος στην εφαρμογή από την βάση δεδομένων. Με την μέθοδο getUserList παίρνουμε μια λίστα με όλους του χρήστες συνδέθηκαν ή δημιούργησαν λογαριασμό στην εφαρμογή μας, παίρνουμε τα στοιχεία τους και τα εμφανίζουμε στην εφαρμογή. Με την μέθοδο UpdateUserProfileData ενημερώνουμε στην βάση δεδομένων τα νέα στοιχεία του χρήστη.

■ 5.2.4 \extension

Στο package αυτό έχουμε αρχεία όπου είναι βοηθητικά για άλλα αρχεία.

5.2.4.1 ViewExtensions.kt

Αυτό το αρχείο χρησιμοποιείται για την εμφάνιση των snack bars.

■ 5.2.5 \models

Αυτό το package χρησιμοποιείται για να μπορούμε να διατηρούμε τα δεδομένα μας «ενεργά». Έτσι ανά πάσα στιγμή έχουμε άμεση πρόσβαση στα δεδομένα χωρίς να χρειάζεται να τα ξανά φορτώνουμε από την βάση δεδομένων. Αρά σε τυπικές λειτουργίες και λειτουργείες που χρησιμοποιούμε συχνά έχουμε τα δεδομένα μας άμεσα και γρήγορα στην διάθεση μας.

a 5.2.5.1 Log.kt

Στο data model αυτό αποθηκεύουμε τα στατιστικά των επιδόσεων του εκάστοτε χρήστη έτσι ώστε να έχουμε άμεση πρόσβαση σε αυτά σε όλη την εφαρμογή.

☐ 5.2.5.2 User.kt

Στο data model αυτό αποθηκεύουμε τα προσωπικά δεδομένα του εκάστοτε χρήστη έτσι ώστε να έχουμε άμεση πρόσβαση σε αυτά σε όλη την εφαρμογή.

■ 5.2.6 \utils

Όλα τα αρχεία που βρίσκονται στον φάκελο αυτό είναι υπεύθυνα για τα βοηθητικά προγράμματα της εφαρμογής.

i 5.2.6.1 CustomDialog.kt

Στο αρχείο αυτό έχουμε δημιουργήσει όλα τα dialog που θα χρησιμοποιήσουμε στην εφαρμογή.

1 5.2.6.2 SnackBarErrorClass.kt

Στο αρχείο αυτό έχουμε δημιουργήσει το snack bar error που θα το χρησιμοποιήσουμε στην εφαρμογή όταν θέλουμε να ενημερώσουμε τον χρήστη για κάτι το οποίο απέτυχε.

Στο αρχείο αυτό έχουμε δημιουργήσει το view του error snack bar, δηλαδή τον τρόπο εμφάνισης του στην εφαρμογή.

Στο αρχείο αυτό έχουμε δημιουργήσει το snack bar success που θα το χρησιμοποιήσουμε στην εφαρμογή όταν θέλουμε να ενημερώσουμε τον χρήστη για κάτι το οποίο έχει γίνει με επιτυχία.

5.2.6.5 SnackBarSuccessView.kt

Στο αρχείο αυτό έχουμε δημιουργήσει το view του success snack bar, δηλαδή τον τρόπο εμφάνισης του στην εφαρμογή.

☐ 5.2.6.6 Utils.kt

Στο αρχείο αυτό έχουμε δημιουργήσει γενικότερες συναρτήσεις στις οποίες χρησιμοποιούμε στην εφαρμογής μας ανάλογα με το τι χρειαζόμαστε.

1 5.2.6.7 Constants.kt

Αυτό το αρχείο χρησιμοποιείται για να αποθηκεύουμε όλες τις σταθερές που χρησιμοποιούμε στην εφαρμογή.

6. ΒΙΒΛΙΟΓΡΑΦΙΑ

- https://www.flaticon.com/
- https://developer.android.com/docs
- https://www.prodigygame.com/main-en/blog/how-to-teach-multiplication/
- https://stackoverflow.com/questions/14437745/how-to-override-action-bar-backbutton-in-android
- https://stackoverflow.com/questions/51908817/alert-dialog-does-not-dismiss
- https://stackoverflow.com/questions/7928803/background-music-android
- https://stackoverflow.com/questions/14433482/how-to-resume-mediaplayer-inandroid
- https://stackoverflow.com/questions/58063609/how-to-play-background-musicthrough-all-activities-using-kotlin
- https://stackoverflow.com/search?q=pause+and+resume+mediaplayer
- https://stackoverflow.com/questions/16157362/android-onresume-onpause-and-mediaplayer
- https://stackoverflow.com/questions/17522185/how-to-resume-my-audio-inandroid
- https://stackoverflow.com/questions/20880962/background-music-plays-when-app-is-closed/20881244#20881244
- https://stackoverflow.com/questions/44069499/service-is-creating-new-instanceof-mediaplayer
- https://thecodeprogram.com/android-mediaplayer-explanation-with-service
- https://developer.android.com/training/transitions/start-activity

ΒΙΒΛΙΟΓΡΑΦΙΑ 38

- https://stackoverflow.com/questions/4250149/requestfeature-must-be-calledbefore-adding-content
- https://stackoverflow.com/questions/2257963/how-to-show-a-dialog-to-confirmthat-the-user-wishes-to-exit-an-android-activity
- https://stackoverflow.com/questions/8384067/how-to-dismiss-the-dialog-withclick-on-outside-of-the-dialog
- https://stackoverflow.com/questions/24032956/action-bar-back-button-notworking
- https://stackoverflow.com/questions/8035682/animate-progressbar-update-inandroid
- https://stackoverflow.com/questions/4817014/animate-a-custom-dialog
- https://stackoverflow.com/questions/51365021/update-progressbar-with-acountdowntimer
- https://stackoverflow.com/questions/52569581/countdown-timer-with-aprogressbar
- https://dev.to/collinskesuibai/show-hide-android-soft-keyboard-android-31-seconds-of-code-1297
- https://stackoverflow.com/questions/46038476/how-could-i-split-a-string-into-an-array-in-kotlin
- https://stackoverflow.com/questions/4312319/how-to-capture-the-virtualkeyboard-show-hide-event-in-android
- https://stackoverflow.com/questions/61195561/kotlin-trying-to-add-the-followingfunction-to-a-global-shared-file-to-be-used
- https://stackoverflow.com/questions/61195561/kotlin-trying-to-add-the-followingfunction-to-a-global-shared-file-to-be-used
- https://stackoverflow.com/questions/47199242/error-unresolved-reference-inkotlin-android

ΒΙΒΛΙΟΓΡΑΦΙΑ 39

- https://stackoverflow.com/questions/57717448/how-to-fix-unresolved-referencewhen-passing-context-with-thismapsactivity
- https://stackoverflow.com/questions/46038476/how-could-i-split-a-string-into-anarray-in-kotlin
- https://dev.to/collinskesuibai/show-hide-android-soft-keyboard-android-31-seconds-of-code-1297
- https://stackoverflow.com/questions/29743058/material-design-how-to-settransparency-for-android-floating-action-button
- https://stackoverflow.com/questions/42922181/error-icon-is-not-showing-in-all-edittext-fields
- https://stackoverflow.com/questions/5273436/how-to-get-activitys-content-view
- https://stackoverflow.com/questions/5265913/how-to-use-putextra-and-getextrafor-string-data
- https://stackoverflow.com/questions/58944590/how-to-resume-countdowntimer
- https://stackoverflow.com/questions/8306374/android-how-to-pause-and-resumea-count-down-timer
- https://stackoverflow.com/questions/58771978/how-to-resume-a-pausedcountdowntimer-in-kotlin
- https://www.youtube.com/watch?v=VnRxq4Fpl64&list=RDCMUCSlvrn68cUk8CS8M btBmBkA&start_radio=1
- https://stackoverflow.com/questions/43977565/is-it-possible-to-increase-the-height-of-the-line-inside-the-progress-bar-androi/43977753#43977753
- https://stackoverflow.com/questions/52569581/countdown-timer-with-aprogressbar
- https://stackoverflow.com/questions/11285961/how-to-make-a-background-20transparent-on-android
- https://stackoverflow.com/questions/29743058/material-design-how-to-settransparency-for-android-floating-action-button

 $BIB\Lambda IO\Gamma PA\Phi IA$ 40

- https://stackoverflow.com/questions/37499207/android-floating-action-buttonsemi-transparent-background-color
- https://stackoverflow.com/questions/50817383/change-the-textinputlayoutoutline-color
- https://stackoverflow.com/questions/58038118/how-to-place-two-views-center-horizontal-in-constraintlayout
- https://stackoverflow.com/questions/8255618/check-if-extras-are-set-or-not
- https://stackoverflow.com/questions/4233873/how-do-i-get-extra-data-from-intent-on-android
- https://stackoverflow.com/questions/44301301/android-how-to-achievesetonclicklistener-in-kotlin
- https://stackoverflow.com/questions/46128759/java-android-countdown-timerand-progressbar-read-first/46128971
- https://stackoverflow.com/questions/15293608/disable-button-click-sound-inandroid

 $BIB\Lambda IO\Gamma PA\Phi IA$ 41

7. ΒΙΒΛΙΟΘΗΚΕΣ & ΕΡΓΑΛΕΙΑ

- Android Studio
- Firebase Cloud (Βάση Δεδομένων)
- Github (*Version Control*)
- Adobe XD (Σχεδίαση UI)
- Kotlin

ΒΙΒΛΙΟΘΗΚΕΣ

Όνομα	Έκδοση
Kotlin	1.5.20
com.google.firebase:firebase-database	-
com.google.firebase:firebase-bom	27.1.0
com.google.firebase:firebase-firestore	23.0.1
com.firebaseui:firebase-ui-firestore	7.1.1
com.firebaseui:firebase-ui-auth	7.1.1
com.google.firebase:firebase-auth-ktx	-
de.hdodenhof:circleimageview	3.1.0
com.github.bumptech.glide:glide	4.12.0
com.github.bumptech.glide:compiler	4.11.0