



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή Εργασία

Τίτλος Πτυχιακής Εργασίας	«Παιχνίδι Γνώσεων Πραγματικού Χρόνου Πολλαπλών Παικτών σε Συσκευές Android» «Real-Time Multiplayer Quiz Game in Android Devices»
Ονοματεπώνυμο Φοιτητή	Γκουτζηγιάνης Χρήστος
Πατρώνυμο	Δημήτριος
Αριθμός Μητρώου	Π18030
Επιβλέπων	Αλέπης Ευθύμιος, Καθηγητής

Ημερομηνία Παράδοσης

Φεβρουάριος 2025

Copyright ©

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πειραιώς.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

Περίληψη

Η παρούσα πτυχιακή εργασία παρουσιάζει την ανάπτυξη μια εφαρμογής παιχνιδιού γνώσεων σε πραγματικό χρόνο, η οποία στοχεύει αποκλειστικά σε συσκευές Android. Η εφαρμογή χρησιμοποιείται ταυτόχρονα ως εργαλείο ψυχαγωγίας και διασκέδασης, αλλά και ως έναν αποτελεσματικό τρόπο για τον χρήστη να διευρύνει τις εγκυκλοπαιδικές του γνώσεις.

Συγκεκριμένα η εφαρμογή εστιάζει στον γρήγορο ρυθμό παιχνιδιού και απόκρισης, ως βασικό στοιχείο της διαδικασίας. Ο ανταγωνισμός μεταξύ χρηστών, σε συνδυασμό με τον γρήγορο ρυθμό, προσφέρουν μια ευχάριστη εμπειρία παιχνιδιού. Εξίσου σημαντική είναι η πληθώρα και ποικιλία ερωτήσεων που περιέχει, καθώς έτσι η εφαρμογή παραμένει ενδιαφέρουσα μετά από αρκετές ώρες παιχνιδιού.

Η εργασία αυτή αποτελεί ένα εξαιρετικό παράδειγμα ποιοτικής ενασχόλησης με τις κινητές μας συσκευές, ειδικά σε μια εποχή όπου παίζουν καίριο ρόλο στην καθημερινότητά μας.

Λέξεις Κλειδιά: εφαρμογή για κινητά, πραγματικού χρόνου, πολλών παικτών, παιχνίδι γνώσεων, ερωτήσεις, απαντήσεις

Abstract

This thesis presents the development of a real-time quiz game application, which is exclusively aimed at Android devices. The application serves both as a tool for entertainment and fun, as well as an effective way for users to broaden their encyclopedic knowledge.

Specifically, the application focuses on a fast-paced gameplay and response system, which are key elements of the process. The competition between users, combined with the rapid pace, offers an enjoyable gaming experience. Equally important is the wide variety and diversity of questions, which help keep the application engaging even after many hours of gameplay.

This work is an excellent example of quality engagement with our mobile devices, especially at a time when they play a crucial role in our daily lives.

Keywords: mobile application, real-time, multiplayer, quiz game, questions, answers.

Περιεχόμενα

Περίληψη.....	3
Abstract.....	3
Κατάλογος Εικόνων	6
Εισαγωγή.....	7
Ευχαριστίες.....	8
Στόχοι.....	9
Αναφορά σε Παρόμοιες Εφαρμογές	10
Quizdom.....	10
Trivia Crack: Fun Quiz Games	10
Outsmarted	11
Εγχειρίδιο Χρήστη	12
Σημειώσεις Εκκίνησης.....	12
Εκκίνηση Πρώτη Φορά	12
Εκκίνηση Παιχνιδιού.....	13
Κατά τη Διάρκεια του Παιχνιδιού.....	13
Τέλος Παιχνιδιού	13
Σχεδιασμός Εφαρμογής.....	14
Διάγραμμα Ροής Χρήστη.....	14
Screenshots Εφαρμογής	16
Login Screen	16
Sign Up Screen	16
Main Screen	17
Help Screen.....	17
Game Screen	18
Game Over Screen (Win)	19
Game Over Screen (Loss).....	19
Τεχνική Τεκμηρίωση.....	20
Περιγραφή Λύσης και Λογισμικού	20
Λειτουργικό Σύστημα.....	20
Περιβάλλον Ανάπτυξης	20
Firebase	20
OpenTriviaDB API.....	21
Οργάνωση Πληροφορίας.....	22
Αναλυτική περιγραφή της διαδικασίας παιχνιδιού.....	23
Αναλυτικός Κατάλογος Κλάσεων και Αρχείων	28

LoginActivity.java.....	28
SignUpActivity.java.....	28
MainActivity.java.....	28
HelpDialog.java	29
PreGameDialog.java	29
GameActivity.java.....	29
GameOverActivity.java	29
CustomAdapter.java	29
QuestionAnswer.java.....	29
firebase.json	29
functions (Φάκελος).....	29
Συμπεράσματα	30
Σύνοψη.....	30
Περιορισμοί και προβλήματα.....	30
Μελλοντικές επεκτάσεις.....	30
Βιβλιογραφία	31
Android.....	31
Firebase	31
Παράρτημα Α	32

Κατάλογος Εικόνων

Figure 1: Quizdom	10
Figure 2: Outsmarted	11
Figure 3: Διάγραμμα Ποής Χρήστη	14
Figure 4: Login Screen.....	16
Figure 5: Sign Up Screen.....	16
Figure 6: Main Screen.....	17
Figure 7: Help Screen.....	17
Figure 8: Game Screen.....	18
Figure 9: Game Over Screen (Win)	19
Figure 10: Game Over Screen (Loss).....	19
Figure 11: OpenTriviaDB API Response.....	21
Figure 12: Παράδειγμα Collections και Documents	22
Figure 13: Δομή Δεδομένων Firestore	23
Figure 14: Οι clients γράφουν το userID στη Firesore	24
Figure 15: Έλεγχος για παίκτες	25
Figure 16: onCreate Cloud Function.....	26
Figure 17: Ανάγνωση ερωτήσεων από Android clients.....	27
Figure 18: Τέλος παιχνιδιού	28

Εισαγωγή

Αυτό το εγχείρημα έχει ως βάση του το συνεχώς αναπτυσσόμενο τοπίο ψυχαγωγίας και διασκέδασης μέσα από τις κινητές μας συσκευές. Η αυξανόμενη υπολογιστική δύναμη που παρουσιάζουν τα κινητά, η καθολικότητα τους ως βασικό εργαλείο και η μεγάλη ζήτηση ψυχαγωγίας είναι κάποιοι από τους παράγοντες που επηρεάζουν το προαναφερθείσα τοπίο. Μέσα από κείμενο, εικόνες, βίντεο και διαδραστικές εφαρμογές οι χρήστες αναζητούν συνεχώς την επόμενη εφαρμογή με την οποία θα περάσουν ποιοτικά τον ελεύθερο τους χρόνο.

Μια σταθερά στην ιστορία των κινητών συσκευών αποτελούν τα παιχνίδια γνώσεων. Ακόμα και πριν οι κινητές συσκευές γίνουν τόσο καθολικές όσο είναι σήμερα, τα παιχνίδια γνώσεων ήταν εξαιρετικά δημοφιλή. Μέσα από επιτραπέζια και κονσόλες, όλοι έχουμε παίξει έστω μια φορά ένα παιχνίδι γνώσης. Μεταγενέστερα βλέπουμε κυριολεκτικά εκατοντάδες εφαρμογές παιχνιδιών γνώσεων να κατακτούν το ενδιαφέρον των χρηστών. Η δημοφιλία τους είναι ένας από τους παράγοντες που ενέπνευσαν την συγκεκριμένη πτυχιακή εργασία.

Επίσης παρουσιάζουν μεγάλο τεχνικό ενδιαφέρον. Η πολυπλοκότητα τους αποτελεί μια ενδιαφέρουσα εκπαιδευτική εμπειρία, καθώς αποτελούνται από πολλά και διαφορετικά κομμάτια τα οποία πρέπει να ενωθούν για να αποδώσουν ένα αρμονικό σύνολο. Συγκεκριμένα ο συγχρονισμός, με μέσο το διαδίκτυο, διαφορετικών συσκευών Android, το στοιχείο του πραγματικού χρόνου και η εύληπτη και χρηστική διεπαφή της εφαρμογής αποτελούν κάποια από τα παραπάνω κομμάτια.

Συμπερασματικά, το παρόν εγχείρημα όχι μόνο αξιοποιεί τη διαρκώς εξελισσόμενη τεχνολογία των κινητών συσκευών, αλλά αναδεικνύει και τις δυνατότητες που προσφέρει ο συνδυασμός ψυχαγωγίας και εκπαίδευσης. Η ανάπτυξη μιας εφαρμογής παιχνιδιού γνώσεων σε πραγματικό χρόνο, πέρα από την τεχνική πρόκληση, αποσκοπεί να προσφέρει μια ποιοτική εμπειρία στον χρήστη, ενώ ταυτόχρονα συμβάλλει στην καλλιέργεια της γνώσης με διασκεδαστικό τρόπο.

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να εκφράσω τις βαθύτατες ευχαριστίες μου στον καθηγητή Ευθύμιο Αλέπη, που μου έδωσε την ευκαιρία να υλοποιήσω την πτυχιακή μου εργασία με θέμα και τεχνολογίες που έχουν άμεση σχέση και ανταπόκριση στον αγορά εργασίας. Επίσης τα μαθήματα του αποτελούν την γνωστική βάση της παρούσας εργασίας, αλλά και το έναυσμα να ασχοληθώ με πιο προχωρημένα θέματα γύρω από το περιβάλλον Android.

Ακόμη θα ήθελα να ευχαριστήσω τους συμφοιτητές μου Χρήστο Λέσκο και Κατερίνα Δεδεμάδη για τον πολύτιμο σχολιασμό και feedback στα αρχικά στάδια ανάπτυξης της εφαρμογής, καθώς αποτέλεσαν τους πρώτους χρήστες της

Σεπτέμβριος 2024

Γκουτζηγιάννης Χρήστος

Στόχοι

Οι βασικοί στόχοι της διπλωματικής εργασίας είναι οι εξής:

- Ανάπτυξη μιας φιλικής προς το χρήστη εφαρμογής Android. Η εφαρμογή απευθύνεται σε οποιοδήποτε χρήστη κινητής συσκευής Android. Αυτό συνεπάγεται ότι θα πρέπει ανεξαρτήτου του επιπέδου εξοικείωσης του χρήστη με την τεχνολογία να είναι εύκολη και ξεκάθαρη στη χρήση της. Αυτό επιτυγχάνεται με έναν απλό σχεδιασμό χρησιμοποιώντας όσο τον δυνατό
- Ενσωμάτωση μιας μεγάλης ποικιλίας ερωτήσεων. Οι ερωτήσεις αποτελούν το κύριο περιεχόμενο του παιχνιδιού. Χωρίς διασκεδαστικές και ενδιαφέρουσες ερωτήσεις το παιχνίδι δεν μπορεί να είναι ελκυστικό. Επομένως κύριο μέλημα είναι η πληθώρα ερωτήσεων, έτσι ώστε μακροπρόθεσμα να παραμένει ενδιαφέρον το παιχνίδι.
- Δημιουργία ενός αποδοτικού συστήματος συγχρονισμού πολλαπλών χρηστών. Το σύστημα συγχρονισμού πρέπει να είναι αξιόπιστο και γρήγορο για να μην εμποδίζει στη διάρκεια του παιχνιδιού. Οι χρήστες πρέπει να είναι ενημερωμένοι για το τι συμβαίνει ανά πάσα στιγμή, από την εύρεση αντιπάλου μέχρι τη λήξη του παιχνιδιού.
- Συνδυασμός της εκπαιδευτικής και της ψυχαγωγικής διάστασης της εφαρμογής. Σημαντικό επίσης είναι, οι χρήστες να γνωρίζουν τα λάθη που έκαναν οι ίδιοι στις ερωτήσεις, καθώς έτσι μόνο μπορούν σταδιακά να διευρύνουν τις γνώσεις τους.

Αναφορά σε Παρόμοιες Εφαρμογές

Κατά την ανάπτυξη της εφαρμογής κρίθηκε απαραίτητο να γίνει μια σύντομη ανασκόπηση σε παρόμοιες εφαρμογές. Έτσι ενημερώθηκε για τις τάσεις που κυριαρχούν αυτή την εποχή γύρω από τα παιχνίδια γνώσεων, τα δυνατά σημεία των εφαρμογών, αλλά και διάφορα σημεία που χρίζουν βελτίωσης. Παρακάτω γίνεται μια αναφορά στις πιο αξιόλογες εφαρμογές που βρήκα κατά την έρευνα μου, οι οποίες είναι: Quizdom, Trivia Crack: Fun Quiz Games, Outsmarted.

Quizdom



Figure 1: Quizdom

Το Quizdom είναι το πιο δημοφιλές παιχνίδι στην Ελληνική αγορά, ξεκινώντας το 2011. Συγκεκριμένα έχει παραπάνω από 2 εκατομμύρια χρήστες αυτή τη στιγμή, ενώ παλαιότερα ήταν εγκατεστημένο σε πάνω από τα μισά smartphones στην Ελλάδα. Το 2017 έκανε παγκόσμια έναρξη, προσεγγίζοντας έτσι αγορές εκτός της ελληνικής, ενώ το 2016 εξαγοράστηκε από τον γερμανικό όμιλο εταιριών friends4media Group.

Το βασικό στοιχείο της εφαρμογής που παρουσιάζει ενδιαφέρον είναι η ασύγχρονη φύση του. Συγκεκριμένα οι χρήστες δεν παίζουν μαζί σε πραγματικό χρόνο, αλλά σε διαφορετικές χρονικές στιγμές. Το θετικό της προσέγγισης αυτής, είναι ότι ο χρήστης δεν χρειάζεται να δεσμεύσει ένα συγκεκριμένο χρονικό διάστημα για να παίξει ένα παιχνίδι. Έτσι παίζει την σειρά του γρήγορα, διευκολύνοντας την διεξαγωγή των παιχνιδιών. Αντιθέτως, το αρνητικό είναι ότι πολλές φορές τα παιχνίδια διαρκούν μέρες και εγκαταλείπονται εντελώς.

Trivia Crack: Fun Quiz Games

Το Trivia Crack: Fun Quiz Games, από την εταιρία etemax, είναι ένα από τα πιο δημοφιλή παιχνίδια γνώσεων παγκοσμίως. Τη στιγμή της σύνταξης αυτού του κειμένου έχει πάνω από 100 εκατομμύρια downloads στο Google Play Store.

Υπάρχουν τρία σημεία ενδιαφέροντος στη συγκεκριμένη εφαρμογή: ο οπτικός σχεδιασμός, η ποσότητα των ερωτήσεων και η εκπαιδευτική πλευρά στο κομμάτι της διαφήμισης. Αρχικά η εφαρμογή περιέχει πάνω από ένα εκατομμύριο ερωτήσεις, ενώ προθέτονται ακόμη και θεματικές ενότητες ερωτήσεων με βάση, για παράδειγμα, το Halloween. Εξίσου σημαντικός είναι ο έντονος σχεδιασμός με μια έντονη παλέτα χρωμάτων και γραφιστικών. Τέλος η εφαρμογή βασίζεται αρκετά στην εκπαιδευτική λειτουργία της. Αυτό το βλέπουμε από το slogan της εφαρμογής στο Google Play Store: “Learn Something New, Every Day”.

Outsmarted



Figure 2: Outsmarted

Το Outsmarted, της εταιρίας QPlay, ξεχωρίζει καθώς συνδυάζει τον ψηφιακό και αναλογικό κόσμο. Έχει πάνω από 100 χιλιάδες downloads στο Google Play Store, ενώ αξιοσημείωτο είναι και το site του παιχνιδιού όπου στεγάζεται αναλυτικό documentation.

Συγκεκριμένα το παιχνίδι χρησιμοποιεί ταυτόχρονα ένα επιτραπέζιο παιχνίδι και μια Android εφαρμογή, δίνοντας έτσι μια πολύ ιδιαίτερη εμπειρία στους χρήστες της. Έχει πάνω από 7.000 ερωτήσεις και δίνει την επιλογή για συγκεκριμένες ερωτήσεις με βάση την χώρα των παικτών.

Εγχειρίδιο Χρήστη

Σημειώσεις Εκκίνησης

Για να ξεκινήσετε να χρησιμοποιείτε την εφαρμογή Quizlet, ακολουθείστε τα παρακάτω βήματα.

Αρχικά βεβαιωθείτε ότι διαθέτετε μια κινητή συσκευή smartphone. Επίσης βεβαιωθείτε ότι το τηλέφωνο σας είναι συνδεδεμένο σε Wi-Fi ή τα δεδομένα κινητής τηλεφωνίας είναι ενεργοποιημένα. Η σύνδεση στο διαδίκτυο είναι απαραίτητη τόσο για την εγκατάσταση όσο και για την λειτουργία της εφαρμογής.

Βήματα εγκατάστασης:

1. Ελέγξτε ότι έχετε μια λειτουργική σύνδεση στο διαδίκτυο. Για παράδειγμα μπορείτε να μπειτε στον φυλλομετρητή του κινητού σας και να μπειτε στην ιστοσελίδα google.com. Εάν φορτώσει με επιτυχία τότε προχωρήστε στα επόμενα βήματα.
2. Ανοίξτε την ψηφιακή εφαρμογή διανομής της συσκευής σας. Ανάλογα με την συσκευή σας αυτή μπορεί να είναι το Google Play Store ή Microsoft Store.
3. Στην γραμμή αναζήτησης, πληκτρολογήστε “Quizlet” και επιλέξτε τη σωστή εφαρμογή στα αποτελέσματα αναζήτησης.
4. Κάντε κλικ στο κουμπί Εγκατάσταση/Install για να κατεβάσετε και να εγκαταστήσετε την εφαρμογή.
5. Μόλις εγκατασταθεί, πατήστε το εικονίδιο Quizlet, στην λίστα με τις εφαρμογές σας για να ξεκινήσει η εφαρμογή.

Εκκίνηση Πρώτη Φορά

Όταν εκκινήσετε την εφαρμογή για την πρώτη φορά, θα πρέπει να δημιουργήσετε ένα λογαριασμό στην εφαρμογή. Βεβαιωθείτε ότι έχετε μια έγκυρη διεύθυνση email, όπως nick@gmail.com. Επίσης θα χρειαστείτε έναν κωδικό που μόνο εσείς θα γνωρίζετε και ένα όνομα χρήστη το οποίο θα βλέπουν οι άλλοι χρήστες.

Όταν ανοίξετε την εφαρμογή θα δείτε στο πάνω μέρος της οθόνης έναν τίτλο που λέει “Log In”. Εδώ θα συμπληρώσετε το email και τον κωδικό σας, αφού δημιουργήσετε τον λογαριασμό σας. Για να δημιουργήσετε έναν λογαριασμό κάντε κλικ πάνω στο μήνυμα που λέει “Not yet registered? Sign Up”.

Σε αυτή την οθόνη πρέπει να συμπληρώσετε τα 4 κουτάκια με τις εξής πληροφορίες:

1. Display Name: Το όνομα το οποίο θα βλέπουν οι άλλοι χρήστες. Προσοχή δεν χρησιμοποιείτε αυτό για να συνδεθείτε στην εφαρμογή.
2. Email: Η διεύθυνση email, που θα χρησιμοποιήσετε για να συνδεθείτε στην εφαρμογή.
3. Password: Ο κωδικός σας. Βεβαιωθείτε ότι ο κωδικός σας είναι ισχυρός, δηλαδή περιέχει νούμερα, σημεία στίξης, μικρά και μεγάλα γράμματα.
4. Confirm password: Εδώ εισάγετε πάλι τον κωδικό σας, για να επιβεβαιωθεί ότι τον γράψατε σωστά.

Αν εισάγετε σωστά τα παραπάνω στοιχεία θα δείτε ένα σύντομο μήνυμα επιτυχίας και θα μεταφερθείτε στην προηγούμενη οθόνη. Εισάγετε το email και τον κωδικό σας για να συνδεθείτε στην εφαρμογή.

Εκκίνηση Παιχνιδιού

Αφού συνδεθείτε, βρίσκετε στην κεντρική οθόνη του παιχνιδιού. Εκεί βλέπετε το όνομα της εφαρμογής, Quizlet, το όνομα χρήστη που θέσατε κατά την εγγραφή σας και δυο κουμπιά. Το δεύτερο κουμπί δείχνει ένα σύντομο μήνυμα βοήθειας με τις βασικές πληροφορίες του παιχνιδιού. Πατώντας το πρώτο κουμπί, ξεκινάει η αναζήτηση παιχνιδιού.

Κατά την αναζήτηση του παιχνιδιού, η εφαρμογή ψάχνει έναν αντίπαλο έτσι ώστε να ξεκινήσει ένα παιχνίδι. Βλέπετε επίσης το σχετικό μήνυμα, “Finding a game for you...”. Αν περιμένετε πάνω από 5 λεπτά, σημαίνει ότι δεν υπάρχουν συνδεδεμένοι παίκτες αυτή τη στιγμή και μπορείτε να δοκιμάσετε να παίξετε αργότερα.

Κατά τη Διάρκεια του Παιχνιδιού

Αφού η εφαρμογή βρει έναν διαθέσιμο παίκτη, ξεκινάει το παιχνίδι.

Έχετε 1 λεπτό και 30 δευτερόλεπτα να απαντήσετε 5 ερωτήσεις γνώσεως. Μπορείτε να βλέπετε τον διαθέσιμο χρόνο, πάνω δεξιά στην οθόνη. Προσοχή εάν ολοκληρωθεί ο χρόνος και δεν έχετε απαντήσει όλες τις ερωτήσεις, το παιχνίδι θα μετρήσει μόνο όσες έχετε απαντήσει. Οι αναπάντητες ερωτήσεις δεν μετριοούνται ως λάθος.

Κάθε φορά που θέλετε να απαντήσετε μια ερώτηση, επιλέξτε την απάντηση και πατήστε το κουμπί στο κάτω μέρος της οθόνης. Αφού απαντήσετε μια ερώτηση δεν μπορείτε να επιστρέψετε να αλλάξετε την απάντησή σας.

Επίσης πάνω αριστερά στην οθόνη μπορείτε να δείτε ποια ερώτηση από τις πέντε βλέπετε. Σημαντικό επίσης είναι ότι κάθε ερώτηση έχει μόνο μια σωστή απάντηση και όλες οι ερωτήσεις είναι πολλαπλής επιλογής.

Η θεματολογία των ερωτήσεων είναι ευρεία, με πάνω από 4.000 ερωτήσεις, και περιέχει θέματα από αθλητικά μέχρι ιστορία και τέχνη.

Τέλος Παιχνιδιού

Κατά την ολοκλήρωση του παιχνιδιού θα δείτε μια νέα οθόνη. Εκεί η εφαρμογή περιμένει όλοι οι παίκτες να ολοκληρώσουν τις απαντήσεις τους. Αφού γίνει αυτό, μετράει τις σωστές απαντήσεις και βγάζει στο πάνω μέρος της οθόνης αν κερδίσατε ή όχι. Αν τελειώσατε γρήγορα, θα χρειαστεί να περιμένετε λίγο, έτσι ώστε να ολοκληρώσει το παιχνίδι και ο αντίπαλος σας.

Μπορείτε επίσης να δείτε μια λίστα με τις απαντήσεις σας. Συγκεκριμένα αν απαντήσατε σωστά μια ερώτηση θα δείτε την ερώτηση και την απάντηση σε πράσινο φόντο. Αν απαντήσατε λάθος, θα δείτε την ερώτηση, την απάντησή σας και την σωστή απάντηση σε κόκκινο φόντο. Μπορείτε να κάνετε scroll για να δείτε όλες τις ερωτήσεις που απαντήσατε.

Τέλος μπορείτε να πατήσετε το κουμπί που λέει “Go back to main screen”, για να επιστρέψετε στην αρχική οθόνη και να ξεκινήσετε νέο παιχνίδι.

Σχεδιασμός Εφαρμογής

Διάγραμμα Ροής Χρήστη

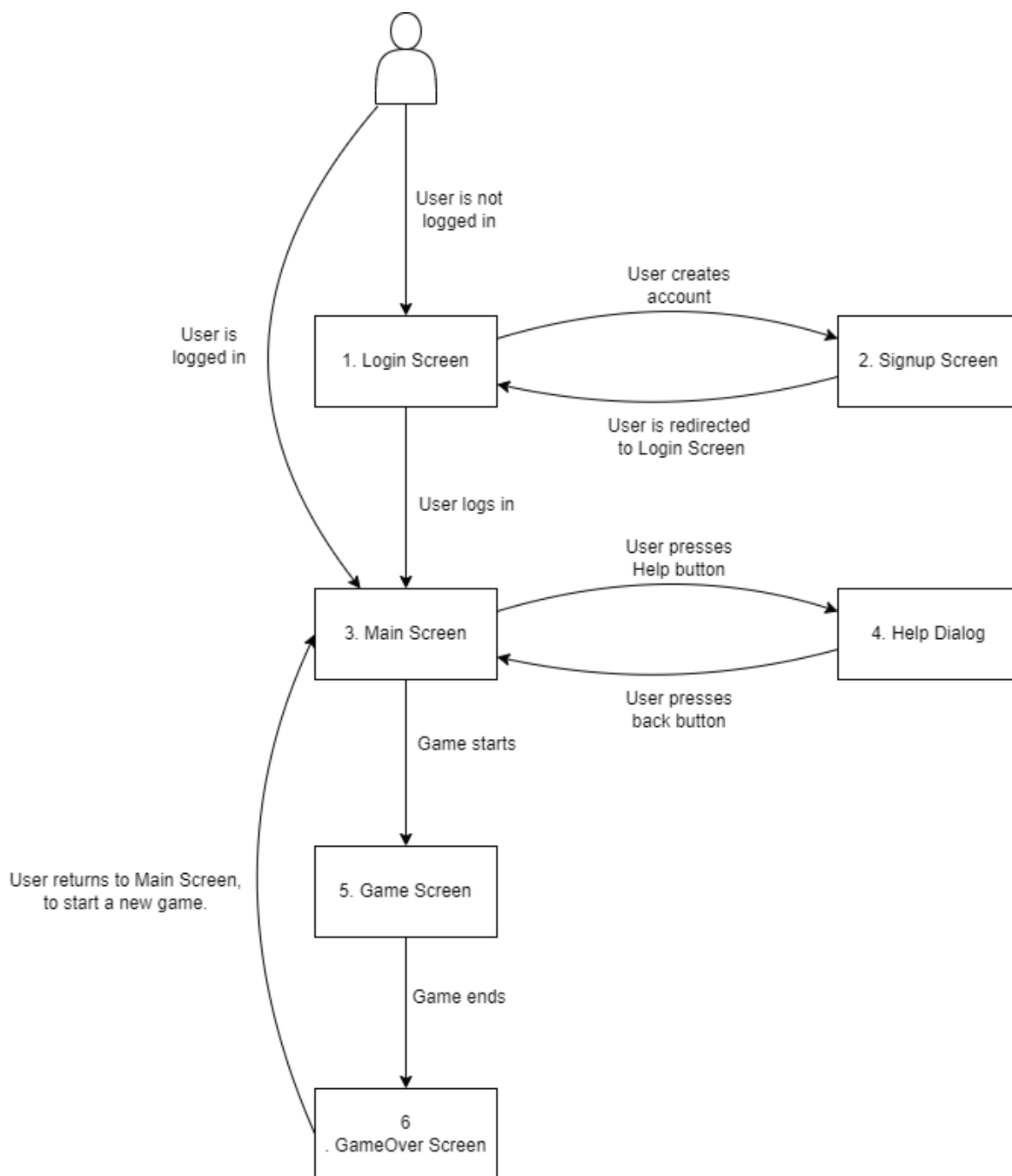


Figure 3: Διάγραμμα Ροής Χρήστη

Το διάγραμμα που προηγήθηκε αποτελεί την διαδρομή που ακολουθεί ένας χρήστης, όσο χρησιμοποιεί την εφαρμογή. Επίσης αναφέρονται και οι ενέργειες που απαιτούνται για να μεταβεί ο χρήστης από την μια οθόνη στην άλλη.

Αναλυτικά οι οθόνες είναι οι εξής:

1. Login Screen.

Η συγκεκριμένη οθόνη είναι η πρώτη οθόνη που βλέπει ο χρήστης. Εάν ο χρήστης έχει κάνει πρόσφατα Log In στην εφαρμογή αυτή η οθόνη παρακάμπτεται εντελώς και ο χρήστης κατευθύνεται αμέσως στη Main Screen. Αν δεν έχει κάνει login πρόσφατα, ο χρήστης καλείται να συμπληρώσει το username και το password του.

2. Signup Screen.

Η Signup Screen είναι υπεύθυνη για την δημιουργία του λογαριασμού του κάθε χρήστη. Ο χρήστης καλείται να συμπληρώσει το email του, το display name (δηλαδή το username το οποίο θα φαίνεται σε άλλους χρήστες), το password και ξανά το ίδιο password για επικύρωση. Αφού δημιουργήσει λογαριασμό κατευθύνεται πίσω στο Login Screen.

3. Main Screen.

Σε αυτή την οθόνη ο χρήστης μπορεί να επιλέξει να ξεκινήσει ένα καινούργιο παιχνίδι ή να δει κάποιες σύντομες οδηγίες.

4. Help Dialog.

Μια σύντομη περιγραφή του παιχνιδιού, έτσι ώστε να γνωρίζει ο χρήστης πόσες ερωτήσεις θα πρέπει να απαντήσει και πόσο χρόνο έχει να το κάνει.

5. Game Screen.

Η Game Screen είναι η βασική οθόνη του παιχνιδιού. Εδώ ο χρήστης μπορεί να βλέπει τις ερωτήσεις και τις επιλογές. Επίσης βλέπει και πόσο χρόνο του απομένει για να απαντήσει όλες τις ερωτήσεις.

6. GameOver Screen.

Δείχνει το τελικό αποτέλεσμα του παιχνιδιού και δείχνει ποιες ερωτήσεις πέτυχε ο χρήστης σωστά και ποιες λάθος.

Στη συνέχεια θα δούμε αναλυτικά screenshots από την εφαρμογή και κάποια σχόλια πάνω στον οπτικό σχεδιασμό της.

Screenshots Εφαρμογής

Login Screen

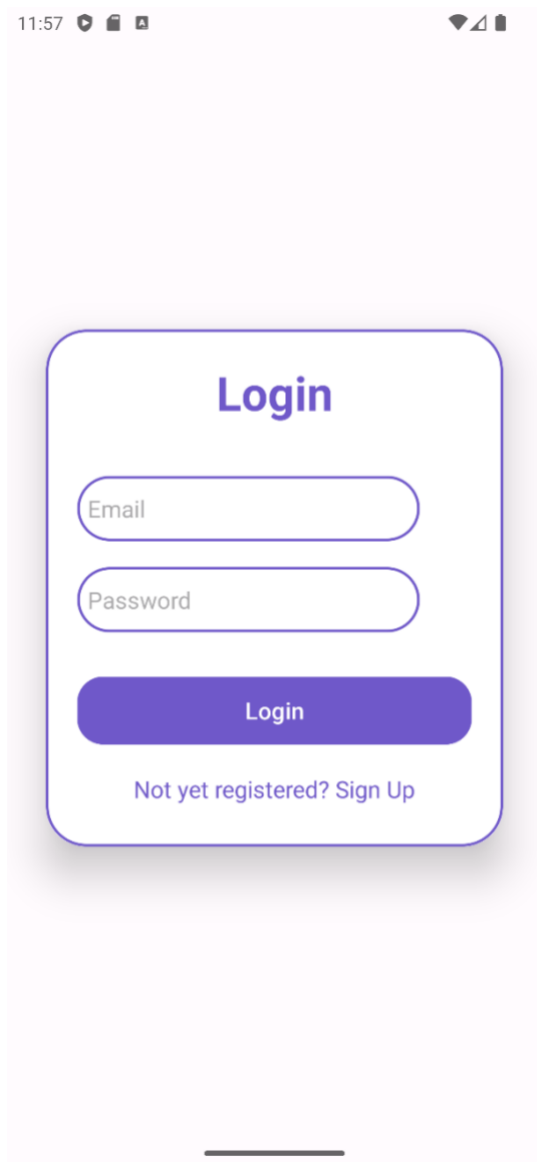


Figure 4: Login Screen

Sign Up Screen

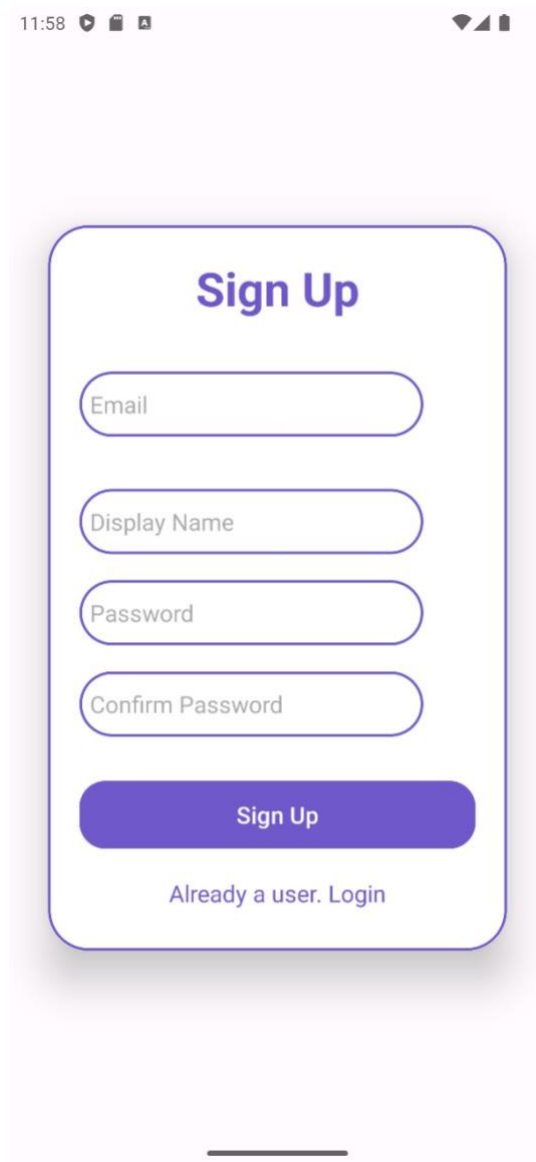


Figure 5: Sign Up Screen

Main Screen

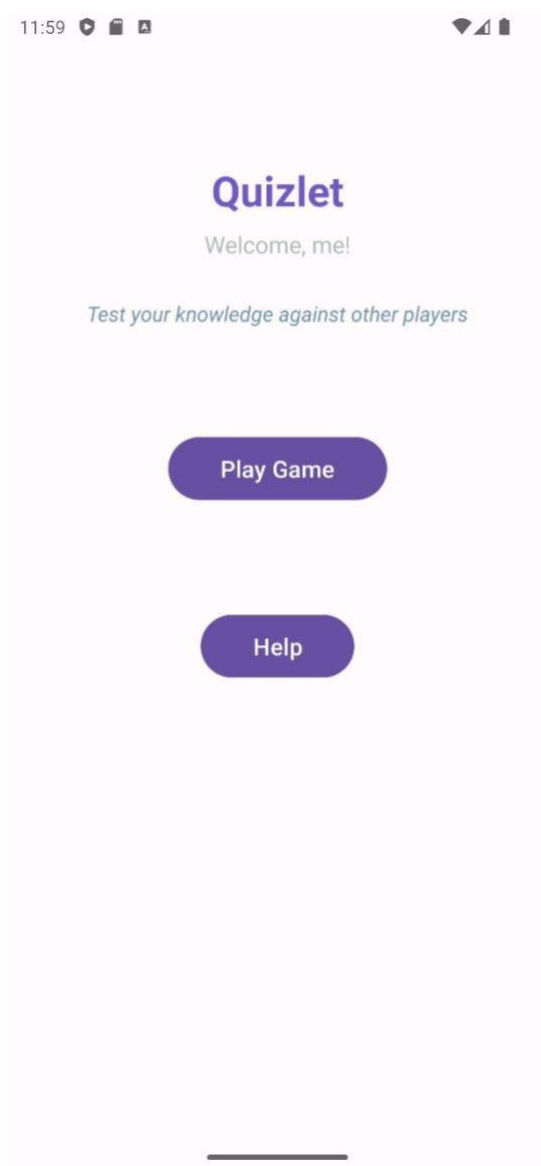


Figure 6: Main Screen

Help Screen

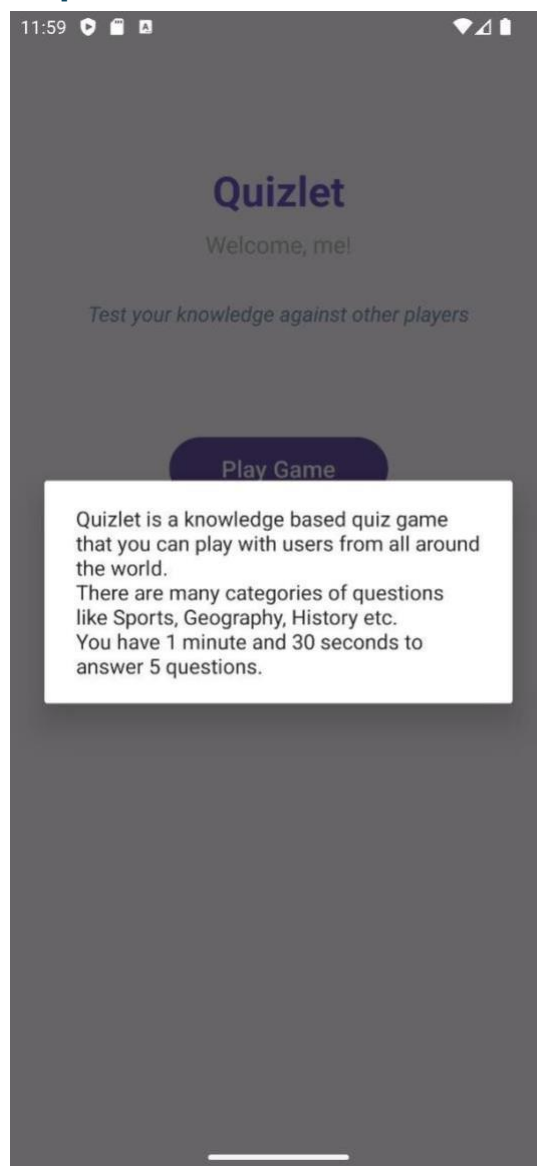
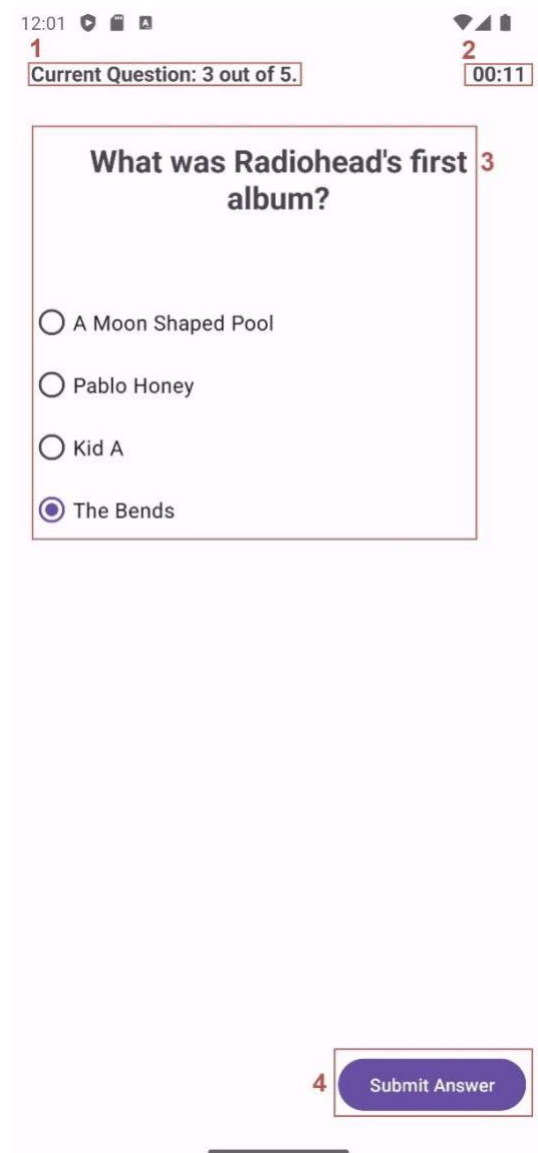


Figure 7: Help Screen

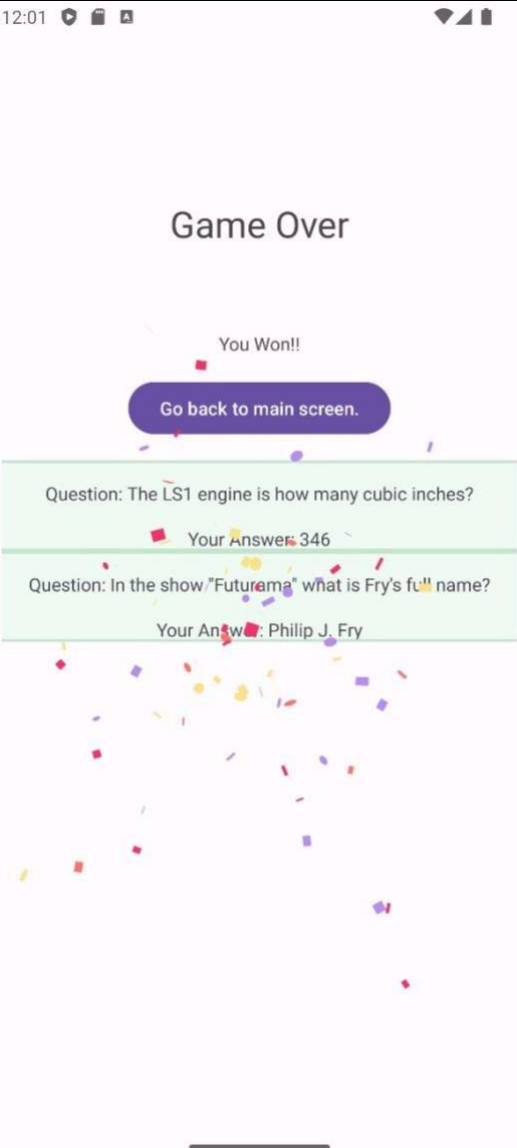
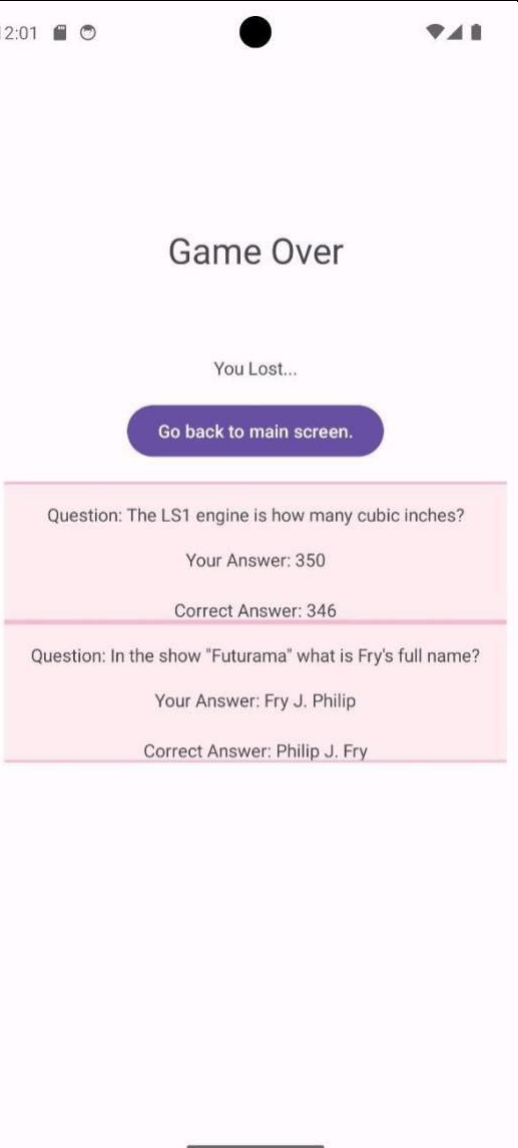
Game Screen



Στη Game Screen βλέπουμε 4 βασικά στοιχεία:

1. Το νούμερο της ερώτησης που βλέπει ο χρήστης.
2. Ο χρόνος που το απομένει να απαντήσει όλες τις ερωτήσεις.
3. Την ερώτηση μαζί με την απάντηση.
4. Το κουμπί το οποίο αποθηκεύει την απάντηση και προχωράει στην επόμενη ερώτηση.

Figure 8: Game Screen

Game Over Screen (Win)		Game Over Screen (Loss)
		
Figure 9: Game Over Screen (Win)		Figure 10: Game Over Screen (Loss)

Τεχνική Τεκμηρίωση

Περιγραφή Λύσης και Λογισμικού

Λειτουργικό Σύστημα

Για την υλοποίηση της εφαρμογής επιλέχθηκε η πλατφόρμα Android, συγκεκριμένα η ελάχιστη έκδοση API 24. Με αυτό τον τρόπο έχουμε μια καλή ισορροπία μεταξύ συμβατότητας με πολλές συσκευές και νέες δυνατότητες που προσφέρονται από πρόσφατες εκδόσεις. Σημαντικό επίσης ήταν να ελαχιστοποιηθούν τα εξωτερικά dependencies, έτσι όπου ήταν δυνατό χρησιμοποιήθηκαν built-in λύσεις του Android.

Περιβάλλον Ανάπτυξης

Η ανάπτυξη της εφαρμογής έγινε με το περιβάλλον Android Studio, έκδοση Jellyfish σε γλώσσα προγραμματισμού Java 21. Εξαιρετικά χρήσιμα κατά την ανάπτυξη της εφαρμογής ήταν το Integrated Terminal και το Debugger που προσφέρει το Android Studio. Επίσης έγινε χρήση του Git ως VCS (Version Control System) και Github.

Firebase

Η Firebase αποτελεί τον πυρήνα της backend αρχιτεκτονικής της εφαρμογής. Οι ανεπτυγμένες δυνατότητες της και το εξαιρετικό developer documentation μου επέτρεψαν να υλοποιήσω την εφαρμογή γρήγορα και με ασφάλεια. Τα επιμέρους κομμάτια της Firebase χρησιμοποιήθηκαν είναι τα εξής:

- Authentication.

Η υπηρεσία Firebase Authentication χρησιμοποιήθηκε για την αναγνώριση των χρηστών. Συγκεκριμένα επιλέχθηκε το Password Authentication, έναντι του Google ή Facebook Authentication, για την απλότητα του. Πολύ χρήσιμο ήταν ότι η Firebase αποθηκεύει κάποια παραπάνω στοιχεία για κάθε χρήστη όπως το display name. Αυτό είναι ένα username που χρησιμοποιήθηκε για να βλέπουν οι χρήστες με ποιον αντίπαλο παίζουν, χωρίς να ξέρουν το email του.

- Firestore.

Η Firebase Firestore είναι μια NoSQL βάση δεδομένων και είναι η βάση που προτείνεται από την Google για νέα προϊόντα, έναντι της Realtime Database. Το πιο σημαντικό feature της Firestore είναι οι προχωρημένες δυνατότητες αναζήτησης και φιλτραρίσματος των δεδομένων. Θα εμβαθύνω στην δομή των δεδομένων και στην Firestore στην επόμενη παράγραφο.

- Cloud Functions.

Οι Cloud Functions για την Firebase είναι ένα serverless framework που σου επιτρέπει να εκτελείς αυτόματα κώδικα στο backend σε απόκριση σε συμβάντα που ενεργοποιούνται από triggers στην Firestore, triggers στην Realtime Database, HTTP requests και πολλά άλλα διαφορετικά events.

Οι γλώσσες προγραμματισμού που μπορούν να χρησιμοποιηθούν για την υλοποίηση των Functions είναι JavaScript και Python. Στη συγκεκριμένη εφαρμογή επιλέχθηκε η JavaScript, κυρίως επειδή τον SDK της είναι παλαιότερο και πιο σταθερό. Επίσης είναι η πιο συχνή επιλογή από την κοινότητα χρηστών της Firebase.

Ο λόγος που επιλέχθηκε η υπηρεσία Cloud Functions, είναι ότι έτσι απομονώθηκε ένα μέρος της λειτουργικότητας στο backend και μειώθηκε το φόρτο εργασίας στον client.

Αξίζει να αναφερθεί ότι για να χρησιμοποιήσεις Cloud Functions, πρέπει να ενταχθείς στο pay-as-you-go πλάνο της Firebase και όχι στο δωρεάν. Αυτό δεν ήταν πρόβλημα διότι τα δωρεάν όρια ήταν πολύ γενναιόδωρα και δεν χρειάστηκε να τα ξεπεράσω.

- Emulator tools.

Κατά τη διάρκεια της ανάπτυξης της εφαρμογής χρησιμοποιήθηκαν τα Firebase Emulator Tools για να προσομοιωθεί το production περιβάλλον τοπικά στον υπολογιστή. Αυτό ήταν εξαιρετικά βοηθητικό καθώς οι αλλαγές στη Firestore και στα Cloud Functions ήταν άμεσες και γρήγορες. Χωρίς τα Emulator Tools, για κάθε αλλαγή σε ένα Cloud Function θα έπρεπε να περιμένω περίπου 2-4 λεπτά έτσι ώστε η Firebase να τελειώσει το deployment των ανανεωμένων Functions. Παρότι κάποια κομμάτια των Emulator Tools βρίσκονται ακόμα σε beta έκδοση, δεν αντιμετώπισα κανένα πρόβλημα.

OpenTriviaDB API

Οι ερωτήσεις προέρχονται από το OpenTriviaDB API, το οποίο διατηρείται από την εταιρία PixelTail Games, και είναι δωρεάν στη χρήση και λογισμικό ανοιχτού κώδικα. Συγκεκριμένα το OpenTriviaDB είναι μια βάση δεδομένων ερωτήσεων, οι οποίες προέρχονται από χρήστες της υπηρεσίας και ελέγχονται έτσι ώστε να ενσωματωθούν στη βάση. Η βάση περιέχει πάνω από 4.000 ερωτήσεις με τρία διαφορετικά επίπεδα δυσκολιών και πολλές κατηγορίες όπως Αθλητικά, Ταινίες, Επιστήμες, Μυθολογία, Επιστήμη κ.α.

Επίσης το API επιστρέφει τα δεδομένα σε JSON και στο HTTP request θέτουμε τον αριθμό των ερωτήσεων που θέλουμε. Συγκεκριμένα το API στη σελίδα https://opentdb.com/api_config.php, κάνει generate ένα URL με βάση τον αριθμό ερωτήσεων και τα θέματα που θέλει ο χρήστης στις ερωτήσεις του. Η εφαρμογή χρησιμοποιεί το URL <https://opentdb.com/api.php?amount=5&type=multiple>, δηλαδή 5 ερωτήσεις πολλαπλής επιλογής. Ένα παράδειγμα ενός API response είναι:

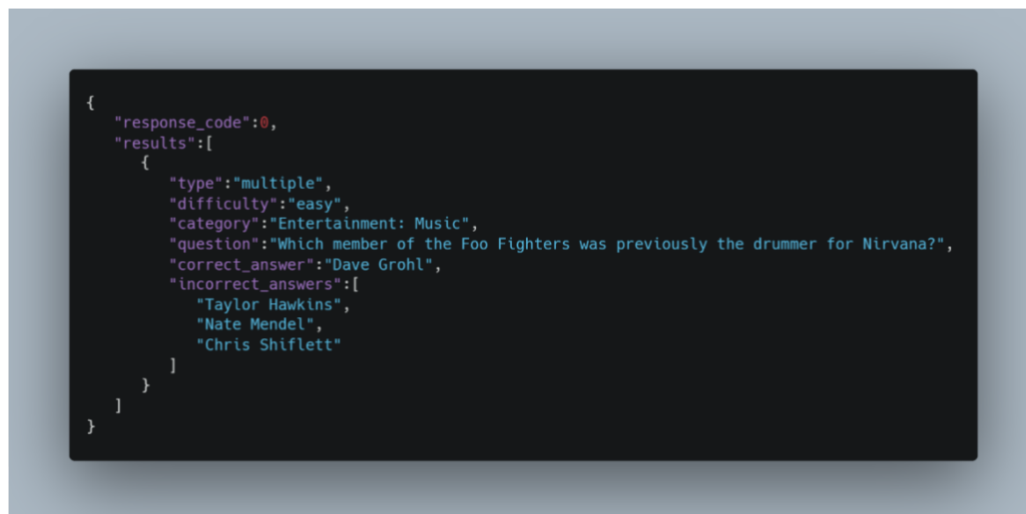


Figure 11: OpenTriviaDB API Response

Το API επιστρέφει έναν κωδικό επιτυχίας και μια λίστα με ερωτήσεις. Για κάθε ερώτηση δίνει τις εξής πληροφορίες:

- Τύπος ερώτησης, μπορεί να είναι πολλαπλής επιλογής ή σωστό/λάθος.
- Δυσκολία, με τρεις διαβαθμίσεις.
- Κατηγορία.
- Ερώτηση.
- Σωστή απάντηση.
- Λάθος απαντήσεις.

Οργάνωση Πληροφορίας

Αρχικά, η Firabase Firestore είναι μια NoSQL βάση δεδομένων που βασίζεται σε documents. Σε αντίθεση με μια SQL βάση δεδομένων που χρησιμοποιεί πίνακες και σειρές, η Firestore χρησιμοποιεί documents που ανήκουν σε collections. Συγκεκριμένα ένα document αποτελείται από δεδομένα οργανωμένα σε ζευγάρια κλειδιού-τιμής και πρέπει πάντα να ανήκει σε ένα collection. Τα documents υποστηρίζουν κλασσικά δεδομένα όπως ακέραιους, Boolean, strings, αλλά και subcollections. Ένα παράδειγμα των collections και documents από το documentation της Firestore είναι:



Figure 12: Παράδειγμα Collections και Documents

Στο συγκεκριμένο παράδειγμα βλέπουμε ένα collection users και δύο documents, aloveace και aturing. Η ιεραρχική δομή της Firestore μας δίνει πολύ ελευθερία στο πως θέλουμε να αναπτύξουμε την δομή της πληροφορίας. Ταυτόχρονα έχει πολύ χαμηλό latency στον συγχρονισμό δεδομένων μεταξύ clients, κάνοντας την εφαρμογή γρήγορη και φιλική προς τον χρήστη.

Παρακάτω ακολουθεί ένα διάγραμμα που δείχνει την δομή των δεδομένων σε μορφή δένδρου. Συνοπτικά έχουμε τα εξής:

- Η collection users κρατάει το game_id, δηλαδή το ID του εκάστοτε παιχνιδιού στο οποίο συμμετέχει ο χρήστης.
- Η collection queue κρατάει μια ουρά με τους παίκτες που θα ήθελαν να παίξουν. Το timestamp χρησιμοποιείται για να διακρίνουμε ποιος παίκτης μπήκε πρώτος στην ουρά

- Η collection games κρατάει όλη την πληροφορία σχετικά με το κάθε παιχνίδι, όπως τις ερωτήσεις, τους πόντους του κάθε παίκτη και αν έχει ολοκληρωθεί το παιχνίδι ή όχι.

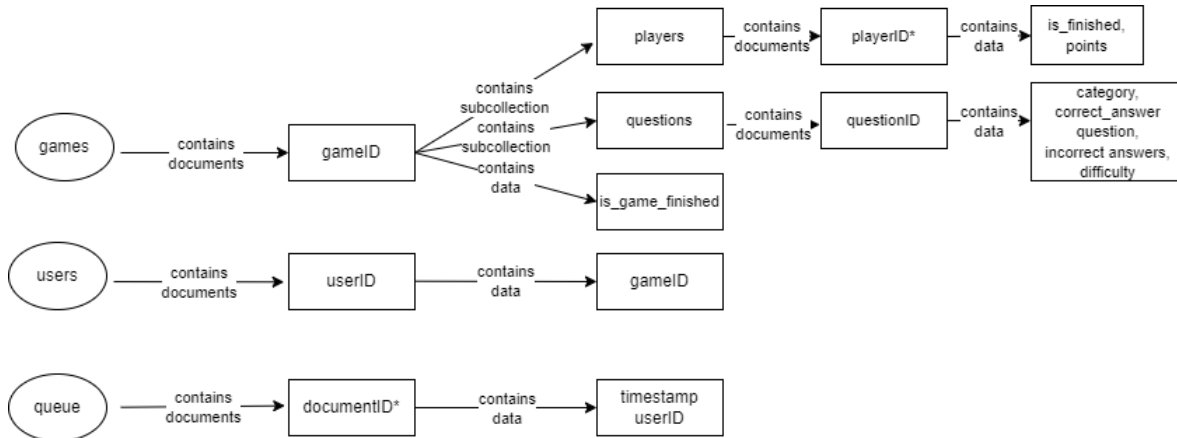


Figure 13: Δομή Δεδομένων Firestore

Τα σημειωμένα με αστεράκι documents (documentID, playerID), αποτελούν documents όπου το ID τους είναι auto-generated από την Firebase και δεν χρησιμοποιείται κάπου. Είναι υποχρεωτικό για κάθε document να έχει ένα ID.

Μια σημαντική λεπτομέρεια για την Firestore είναι ότι το κάθε read είναι ρηχό/shallow. Δηλαδή αν διαβάσουμε ένα document στο games collection, η Firestore δεν θα διαβάσει αυτόματα τα subcollections players και questions. Αυτό επηρεάζει την απόδοση, αλλά κυρίως το κόστος. Αν επιχειρούσαμε να διαβάσουμε την ίδια δομή στη Realtime Database, θα είχαμε το τριπλάσιο κόστος. Για αυτό στο documentation της Firebase για τη Realtime Database λέει ρητά ότι πρέπει τα δεδομένα να είναι οργανωμένα όσο πιο flat γίνεται. Αυτό όμως δυσκολεύει όταν έχεις περίπλοκες ιεραρχίες δεδομένων.

Αναλυτική περιγραφή της διαδικασίας παιχνιδιού

Σε αυτή την παράγραφο θα δείξω αναλυτικά πως οι χρήστες ξεκινούν ένα παιχνίδι, παίζουν και πως τελειώνουν ένα παιχνίδι, με έμφαση στην υλοποίηση του Android client και την αλληλεπίδραση του με το backend.

Αρχικά κάθε χρήστης όταν κάνει signup έχει ένα μοναδικό userID, το οποίο το θέτει η Firebase Authentication, έτσι ώστε να μπορεί να ξεχωρίζει διαφορετικούς χρήστες. Έτσι όταν ένας χρήστης κάνει log in για πρώτη φορά στην εφαρμογή γράφει στη Firestore, στο collection users, ένα document με τίτλο το userID του. Αυτό το document περιέχει ένα gameId το οποίο είναι το παιχνίδι στο οποίο συμμετέχει εκείνη την στιγμή. Η αρχική τιμή του είναι ένα κενό string.

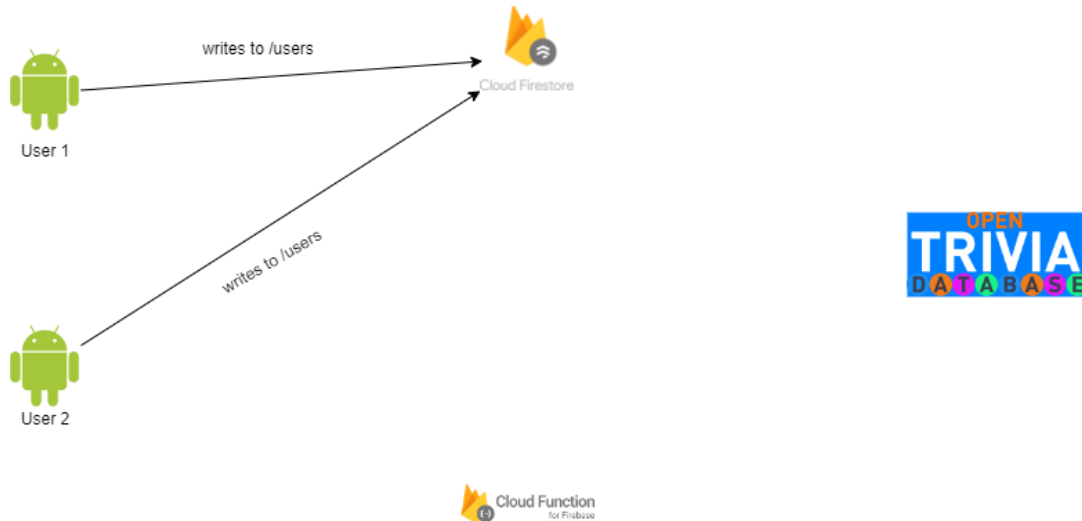


Figure 14: Οι clients γράφουν το userID στη Firesore

Στη συνέχεια κάθε χρήστης πατάει το κουμπί έτσι ώστε να ξεκινήσει το παιχνίδι. Όταν γίνεται αυτό, ο κάθε client γράφει στο collection queue ένα document το οποίο περιέχει το userID και ένα timestamp με βάση την ώρα της Firebase. Έτσι γνωρίζουμε ποιοι παίκτες θέλουν να παίξουν και ποιος παίκτης μπήκε πρώτος στην ουρά. Επίσης εφόσον οι clients απλά προσθέτουν documents στο ίδιο collection δεν υπάρχει κάποιος κίνδυνος να γράψουν στο ίδιο document. Κάθε φορά που ένας client γράφει στο collection queue, κάνει trigger ένα Cloud Function, με όνομα "onCreate".

Η onCreate function ελέγχει πόσα documents έχουμε στο queue collection, δηλαδή πόσοι παίκτες περιμένουν να ξεκινήσουν ένα παιχνίδι. Εάν έχουμε λιγότερους από 2 παίκτες, δεν κάνει τίποτα. Εάν έχουμε πάνω από 2 παίκτες, ξεκινάει την διαδικασία έναρξης του παιχνιδιού. Ιδιαίτερα σημαντικό εδώ είναι ότι η onCreate function, ταξινομεί τους παίκτες με το timestamp που έχουν, έτσι διαλέγει τους παίκτες που περιμένουν την περισσότερη ώρα. Ταυτόχρονα ο κάθε client θέτει έναν listener στο collection users στο document με τίτλο το userID του.

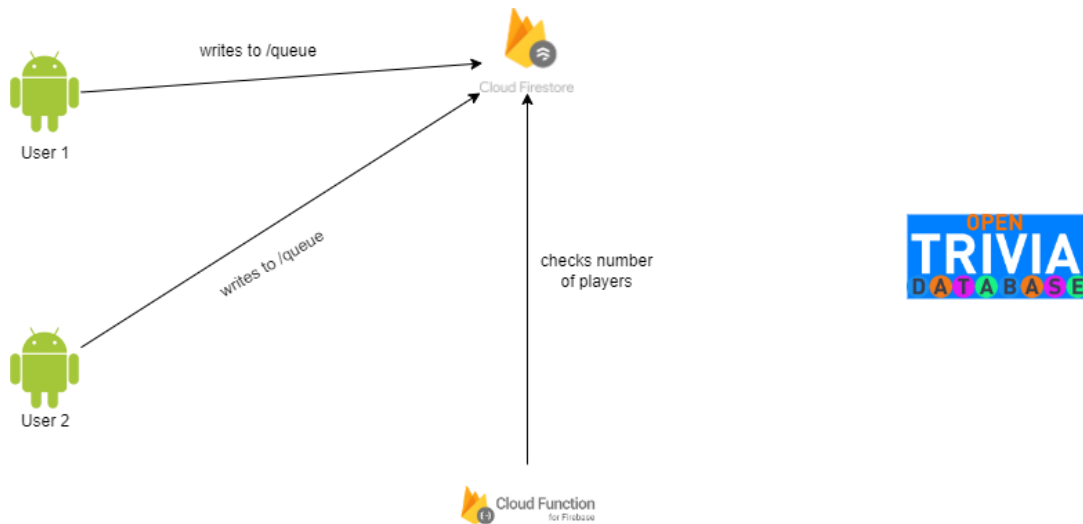


Figure 15: Έλεγχος για παίκτες

Η onCreate function κάνει τα εξής για να προετοιμάσει την έναρξη ενός παιχνιδιού

1. Διαβάζει τα userID των παικτών που θα συμμετέχουν στο παιχνίδι.

Η onCreate function χρειάζεται τα userID των παικτών έτσι ώστε να ενημερώσει τους παίκτες ότι το παιχνίδι θα ξεκινήσει στο Βήμα 4.

2. Στέλνει request στο Open Trivia DB API για να πάρει τις ερωτήσεις.

Συγκεκριμένα χρησιμοποίησα το fetch API για να στείλω ένα GET HTTP request, στο URL <https://opentdb.com/api.php?amount=5&type=multiple>. Έτσι παίρνουμε 5 ερωτήσεις και οι ερωτήσεις είναι τύπου πολλαπλής επιλογής. Το API επίσης προσφέρει ερωτήσεις τύπου σωστού ή λάθους, αλλά δεν τις επέλεξα για το παιχνίδι.

3. Αφού πάρει τις ερωτήσεις, τις αποκωδικοποιεί.

Αφού το request ολοκληρωθεί με επιτυχία, η onCreate function μετατρέπει τις ερωτήσεις από JSON σε JavaScript αντικείμενα σε μια λίστα. Επίσης χρησιμοποιήθηκε η βιβλιοθήκη [he](https://github.com/mathiasbynens/he) (<https://github.com/mathiasbynens/he>), καθώς κάποιος χαρακτήρες, όπως τα μονά αποσιωπητικά, είναι κωδικοποιημένοι έτσι ώστε να εμφανίζονται σωστά ως JSON δεδομένα.

4. Δημιουργεί ένα καινούργιο document στη συλλογή games και ανανεώνει τα documents στη users collection.

Για την δημιουργία ενός νέου παιχνιδιού, η onCreate δημιουργεί ένα καινούργιο document στη games collection. Συγκεκριμένα περιέχει μια μεταβλητή is_game_finished, και δυο subcollections questions και players. Το subcollection players και η is_game_finished χρησιμοποιούνται κατά την λήξη του παιχνιδιού και θα αναλυθούν αργότερα. Η subcollection questions αποθηκεύει τις ερωτήσεις που θα χρησιμοποιηθούν στο παιχνίδι. Αφού όλα τα παραπάνω ολοκληρωθούν, γράφεται το ID του καινούργιου παιχνιδιού στο document του κάθε παίκτη στη collection users. Έτσι κάνει trigger ο listener που οι clients είχαν θέσει νωρίτερα και ενημερώνονται για την εκκίνηση του παιχνιδιού.

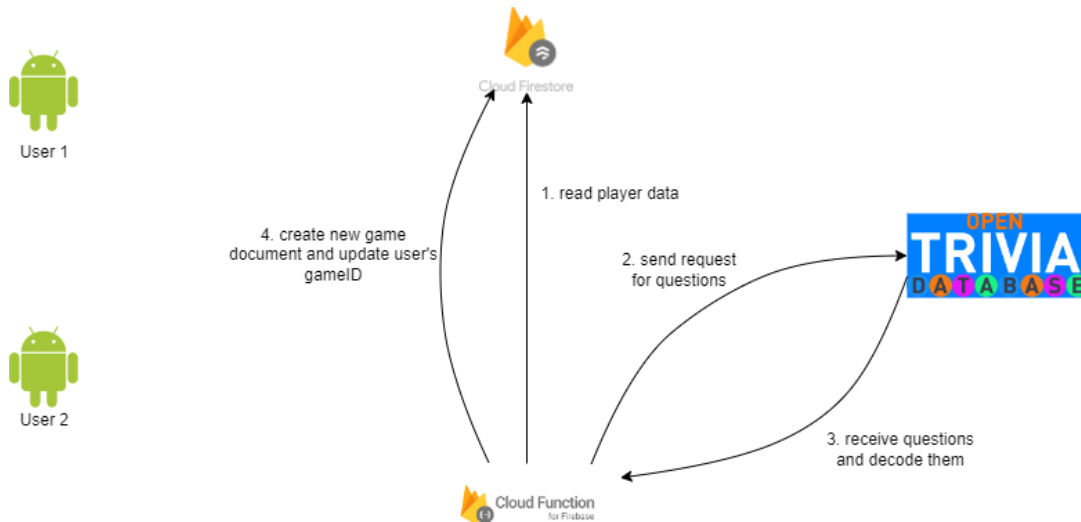


Figure 16: onCreate Cloud Function

Κάποιες ενδιαφέρουσες λεπτομέρειες για την υλοποίηση της onCreate function είναι οι εξής:

- Χρησιμοποιεί το onDocumentCreated trigger, το οποίο παίρνει ως παράμετρο το path “queue/{userID}”. Με αυτό τον τρόπο συνδέεται στην Firestore, παρακολουθεί όλα τα documents που δημιουργούνται και παίρνει το userID τους.
- Ορίζεται τοπικά στον υπολογιστή ως ένα έγκυρο NodeJS project. Αυτό μας επιτρέπει να ορίσουμε τα dependencies τοπικά, και η Firebase τα κατεβάζει κατά την διάρκεια του deployment.
- Πολύ χρήσιμες φάνηκαν οι δυνατότητες της Firestore να φιλτράρει και ταξινομήσει δεδομένα, καθώς εύκολα βλέπουμε ποιος χρήστης περιμένει την περισσότερη ώρα για να βρει ένα παιχνίδι.

Αφού γίνεται trigger ο listener, ο κάθε client διαβάζει το gameId από το users collection και κατεβάζει τις ερωτήσεις. Επίσης αφαιρεί το document που είχε γράψει στο queue collection, καθώς δεν βρίσκεται πια σε αναζήτηση παιχνιδιού. Μεταφέρεται στο GameActivity, όπου ξεκινάει το χρονόμετρο και θα μπορεί να βλέπει τις ερωτήσεις. Για την προβολή των ερωτήσεων χρησιμοποιήθηκε ένα RadioGroup, καθώς εξασφαλίζει ότι μόνο μια απάντηση θα επιλεγεί για κάθε ερώτηση και επειδή είναι ένα built in Android widget, είναι γνώριμο και φιλικό προς όλους τους χρήστες. Τέλος κάθε φορά που ο χρήστης «κλείδωνε» μια απάντηση δημιουργείται ένα QuestionAnswer αντικείμενο και αυξάνεται το counter points (αν η απάντηση ήταν σωστή). Το QuestionAnswer αντικείμενο αποτελεί την απάντηση του χρήστη σε μια ερώτηση, δηλαδή απηθηκεύει την ερώτηση και την απάντηση. Χρησιμοποιείται στο τέλος του παιχνιδιού έτσι ώστε ο χρήστης να ξέρει ποιες ερωτήσεις απάντησε σωστά.

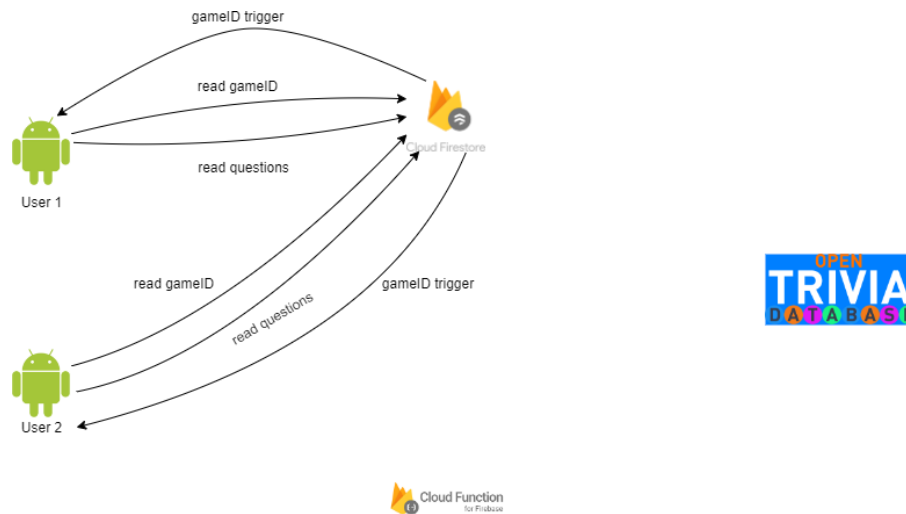


Figure 17: Ανάγνωση ερωτήσεων από Android clients

Αφού ολοκληρωθεί το παιχνίδι ο χρήστης μεταφέρεται στη GameOverActivity. Αρχικά προβάλλονται όλες οι απαντήσεις που έδωσε, καθώς καταγράφηκαν ως QuestionAnswer αντικείμενα. Συγκεκριμένα προβάλλονται ως λίστα σε ένα RecyclerView, για να μπορεί ο χρήστης να κάνει scroll και να δει όλες τις απαντήσεις του. Στη συνέχεια κάθε client, ξεκινάει ένα transaction και αρχικά ελέγχει το subcollection του αντιπάλου για να δει αν τελείωσε πρώτος ή δεύτερος. Ο λόγος που χρησιμοποιούνται transactions σε αυτή την περίπτωση είναι ότι και οι δυο clients θέλουν να γράψουν στο ίδιο document. Η Firebase ορίζει ότι αν ένα transaction δεν μπορεί να ολοκληρωθεί, δοκιμάζει ξανά μετά από λίγο. Έτσι εγγυάται ότι ποτέ δεν θα ολοκληρωθούν τα μισά writes ή reads μέσα σε ένα transaction. Με βάση το παρακάτω διάγραμμα ο User 1 τελειώνει πρώτος και ο User 2 τελειώνει δεύτερος. Για την λήξη του παιχνιδιού έχουμε τα εξής βήματα:

1. Ο User 1 γράφει στο document του στο subcollections players (το path είναι `games/{gameID}/players/userID`) τους πόντους του και την μεταβλητή `is_finished` σε true.
2. Θέτει επίσης έναν listener στην μεταβλητή `is_game_finished` (το path είναι `games/{gameID}/`), έτσι ώστε να ενημερωθεί όταν θα τελειώσει ο άλλος παίκτης.
3. Ο User 2 γράφει τους δικούς του πόντους, γυρίζει την μεταβλητή `is_finished` και την μεταβλητή `is_game_finished`.
4. Έτσι κάνει trigger τον listener από το βήμα 2 και ο User 1 ξέρει ότι το παιχνίδι έχει τελειώσει.

Εφόσον και οι δύο παίκτες έχουν τελειώσει, διαβάζουν τους πόντους του αντιπάλου και εμφανίζεται το αντίστοιχο μήνυμα στην οθόνη.

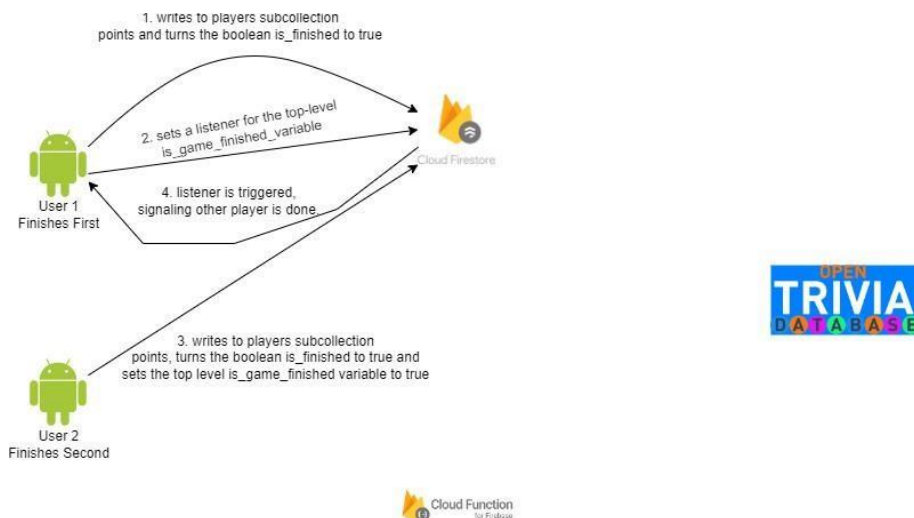


Figure 18: Τέλος παιχνιδιού

Τέλος ο κάθε client θέτει το gameId του σε ένα κενό string έτσι ώστε να είναι έτοιμος να ξεκινήσει ένα καινούργιο παιχνίδι. Επίσης χρησιμοποιήθηκε η βιβλιοθήκη <https://github.com/DanielMartinus/Konfetti>, για να δείξει κομφετί στον νικητή.

Αναλυτικός Κατάλογος Κλάσεων και Αρχείων

LoginActivity.java

Το Activity στο οποίο υλοποιήθηκε το login. Συγκεκριμένα χρησιμοποιήθηκε η μέθοδος signInWithEmailAndPassword(). Επίσης ελέγγω αν ο χρήστης έχει ήδη συνδεθεί, έτσι ώστε να παρακάμψει εντελώς αυτή την οθόνη. Τέλος κατά τη διάρκεια του development χρησιμοποιήθηκε η useEmulator() μέθοδος, έτσι ώστε να συνδεθεί η εφαρμογή με τα Firebase Emulator Tools.

SignUpActivity.java

Το Activity στο οποίο υλοποιήθηκε το signup. Συγκεκριμένα χρησιμοποιήθηκε η μέθοδος createUserWithEmailAndPassword(). Επίσης υλοποιήθηκε validation έτσι ώστε οι δύο κωδικοί να είναι ίδιοι και τα πεδία να μην είναι άδεια.

MainActivity.java

Το βασικό Activity όπου χρήστης ξεκινάει το παιχνίδι. Από εδώ επίσης μπορεί να δει την σύντομη βοήθεια. Ενδιαφέρον σε αυτή τη κλάση παρουσιάζει το πως χρησιμοποιήθηκε το Tasks API έτσι ώστε να συγχρονιστούν τα reads στην Firestore. Συγκεκριμένα επειδή τα reads στην Firestore είναι ασύγχρονα και γίνονται στο background, επιλέχθηκε η μέθοδος Tasks.whenAll(), η οποία παίρνει ως παραμέτρους διάφορα Tasks και ολοκληρώνεται μόνο όταν όλα τα επιμέρους Tasks ολοκληρωθούν.

HelpDialog.java

Μια κλάση που κάνει inherit από την built-in Android κλάση DialogFragment και δείχνει ένα μικρό μήνυμα βοήθειας. Καλείται από την MainActivity, χρησιμοποιώντας έναν FragmentManager.

PreGameDialog.java

Ομοίως κάνει inherit από την built-in Android κλάση DialogFragment και δείχνει στον χρήστη ότι αναζητά ένα παιχνίδι. Καλείται από την MainActivity, χρησιμοποιώντας έναν FragmentManager.

GameActivity.java

Σε αυτό το Activity προβάλλονται οι ερωτήσεις. Πατώντας ο χρήστης το κουμπί «Submit Answer», αποθηκεύεται η απάντηση του και το UI αλλάζει για να δείξει την επόμενη ερώτηση. Το Timer υλοποιήθηκε χρησιμοποιώντας τη builtin CountdownTimer του Android.

GameOverActivity.java

Εδώ προβάλλονται τα τελικά αποτελέσματα το παιχνιδιού. Χρησιμοποιήθηκε ένα RecyclerView για την εύκολη προβολή των αποτελεσμάτων. Τα δεδομένα για το RecyclerView είναι αποθηκευμένα σε μια λίστα QuestionAnswer αντικειμένων. Επίσης εδώ χρησιμοποιήθηκε το Tasks API για να γίνει η ανάγνωση των βαθμών του αντιπάλου.

CustomAdapter.java

Ο Adapter που χρησιμοποιήθηκε για το RecyclerView.

QuestionAnswer.java

Η κλάση QuestionAnswer αποθηκεύει τις απαντήσεις του χρήστη. Έχει μια Boolean μεταβλητή. Αν είναι true σημαίνει ότι ο χρήστης απάντησε σωστά και αποθηκεύει μόνο την ερώτηση και τη σωστή απάντηση. Αν είναι false σημαίνει ότι απάντησε λάθος και αποθηκεύει επίσης την λάθος απάντηση.

firebase.json

Αυτό είναι ένα configuration αρχείο για τα Firebase Emulators Tools, όπου μπορείς να θέσεις πχ τις πόρτες που τρέχουν τα τοπικά εργαλεία κτλ.

functions (Φάκελος)

Εδώ βρίσκονται τα Cloud Functions. Συγκεκριμένα έχουμε τα αρχεία package.json, package-lock.json τα οποία διαχειρίζονται τα dependencies και το αρχείο index.js στο οποίο ορίζουμε τα functions.

Συμπεράσματα

Σύνοψη

Η ανάπτυξη της συγκεκριμένης εφαρμογής αποτελεί μια φιλική προς τον χρήστη και ευχάριστη εμπειρία παιχνιδιού. Ο γρήγορος συγχρονισμός, το στοιχείο του ανταγωνισμού και η πληθώρα ερωτήσεων την κάνουν εύκολη στη χρήση προς όλους. Ταυτόχρονα η αξιοποίηση ψηφιακών εργαλείων, όπως η Firebase ενισχύουν την απόδοση της, ενώ παραμένει προσιτή στη πλειοψηφία των Android συσκευών.

Περιορισμοί και προβλήματα

Ο βασικός περιορισμός κατά την ανάπτυξη της εφαρμογής ήταν η ύπαρξη δύο διαφορετικών συσκευών. Συγκεκριμένα αυτό έκανε το debugging πολύ δύσκολο, καθώς υπήρχαν δύο διαφορετικά logs, δύο διαφορετικές συσκευές που διάβαζαν και έγραφαν από στη βάση και δύο οθόνες που έπρεπε να λειτουργούν παράλληλα για να δοκιμαστεί η εφαρμογή. Αυτό ξεπεράστηκε ως ένα βαθμό με την ένταξη των Firebase Emulator Tools, καθώς έδωσαν πιο εύκολη πρόσβαση στη Firestore και γρηγορότερη απόκριση σε αλλαγές στις Cloud Functions.

Επίσης ένας προβληματισμός αποτελεί το κόστος της εφαρμογής αν έτρεχε στο production. Η Firebase δίνει πολύ γενναιόδωρα όρια για read/writes στη βάση, αλλά και στις Cloud Functions. Αφού όμως ξεπεραστούν τα δωρεάν όρια κάθε ανάγνωση στη βάση και κάθε κάλεσμα σε ένα Cloud Function κοστίζει. Σίγουρα κοστίζει περισσότερο από το να έστηνε κάποιος ένα δικό του backend API και να το έκανε deploy σε έναν server. Δεν θα ήταν όμως εφικτό να πάρουμε την ίδια λειτουργικότητα και αξιοπιστία αν φτιάχναμε το δικό μας backend, ειδικά σε ένα τόσο μικρό χρονικό διάστημα.

Τέλος, παρότι το Open Trivia DB API είναι δωρεάν στη χρήση, έχει συγκεκριμένα rate limits, όπου επιτρέπει έναν συγκεκριμένο αριθμό requests την ώρα. Αυτό σημαίνει ότι αν ξεπεραστεί αυτό το όριο η εφαρμογή δεν θα έχει πρόσβαση σε ερωτήσεις.

Μελλοντικές επεκτάσεις

Υπάρχουν διάφορες ιδέες για την επέκταση και βελτίωση της εφαρμογής. Η πρώτη είναι ένα παραπάνω ξεχωριστό service το οποίο θα έκανε caching στο API. Έτσι θα ελαχιστοποιούσαμε το κίνδυνο να ξεπεράσουμε τα όρια του και θα μπορούσαμε να είχαμε ένα backup ερωτήσεων σε περίπτωση που το API σταματήσει να λειτουργεί.

Επίσης θα είχε ενδιαφέρον να μπορούν οι χρήστες να γίνουν «φίλοι» και να μπορούν να διαλέγουν με ποιον θα παίξουν. Παρότι αυτό θα είχε μεγαλύτερο υπολογιστικό κόστος, θα βελτιώνει εξαιρετικά την εμπειρία των χρηστών, προσθέτοντας μια νέα διάσταση στην εφαρμογή.

Βιβλιογραφία

Ως βιβλιογραφία καταθέτω διάφορα links από documentation τα οποία ήταν καίρια κατά την ανάπτυξη της εφαρμογής.

Android

<https://developer.android.com/reference/android/os/CountDownTimer> - CountDownTimer Reference.

<https://developer.android.com/reference/android/app/DialogFragment> - DialogFragment Reference.

<https://developer.android.com/guide/fragments/dialogs> - Display dialogs with DialogFragment.

<https://developer.android.com/reference/androidx/recyclerview/widget/RecyclerView> - RecyclerView Reference.

<https://developer.android.com/develop/ui/views/layout/recyclerview> - Create dynamic lists with RecyclerView.

<https://developer.android.com/reference/android/widget/RadioGroup> - RadioGroup Reference.

<https://developer.android.com/develop/ui/views/components/radiobutton> - Add radio buttons to your app.

Firebase

<https://firebase.blog/posts/2016/09/become-a-firebase-taskmaster-part-1/> - Firebase Blog, Become a Firebase Taskmaster Part 1

<https://firebase.blog/posts/2016/09/become-a-firebase-taskmaster-part-2> - Firebase Blog, Become a Firebase Taskmaster Part 2

<https://firebase.blog/posts/2016/09/become-a-firebase-taskmaster-part-3> 29 - Firebase Blog, Become a Firebase Taskmaster Part 3

<https://firebase.blog/posts/2016/10/become-a-firebase-taskmaster-part-4> - Firebase Blog, Become a Firebase Taskmaster Part 4

<https://firebase.google.com/docs/firestore/data-model> - Cloud Firestore Data Model

<https://firebase.google.com/docs/firestore/manage-data/structure-data> - Firestore Structure Data

<https://firebase.google.com/docs/firestore/manage-data/transactions> - Firestore Transactions

<https://firebase.google.com/docs/firestore/extend-with-functions-2nd-gen> - Extend Firestore with Cloud Functions

https://firebase.google.com/docs/emulator-suite/connect_functions - Firebase Emulator Tools, Connect your app to the Cloud Functions Emulator

<https://firebase.google.com/docs/auth/android/password-auth> - Firebase Authentication, Password Authentication

Παράρτημα Α

Εξωτερικές πηγές λογισμικού:

1. <https://github.com/mathiasbynens/he> - A robust HTML entity encoder/decoder written in JavaScript.
2. <https://github.com/DanielMartinus/Konfetti> - Easily celebrate little and big moments in your app with this lightweight confetti library!

Οι πηγές λογισμικού βρίσκονται σε δημόσια αποθετήρια στο GitHub και διανέμονται με ελεύθερες άδειες λογισμικού.