

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
ΕΦΑΡΜΟΓΗ ANDROID (Globe)



ΑΘΑΝΑΣΙΑ ΚΟΜΜΑΤΙΔΟΥ
Π18078

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΕΥΘΥΜΙΟΣ ΑΛΕΠΗΣ

Πειραιάς, Σεπτέμβριος 2022



ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	Περίληψη	3
2	Abstract	3
3	Εισαγωγή	4
4	Ανασκόπηση πεδίου	5
5	Παρουσίαση και χρήση εφαρμογής	5
5.1	Εγγραφή και σύνδεση στην εφαρμογή	6
5.2	Κεντρικό μενού εφαρμογής	8
5.3	Λειτουργία Μετάφρασης	9
5.4	Λειτουργία Αναγνώρισης Αξιοθέατων	12
5.5	Λειτουργία Μετατροπής συναλλάγματος	14
5.6	Προφίλ χρήστη	15
6	Αρχιτεκτονική συστήματος	16
6.1	Εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής	16
6.1.1	Android studio	16
6.1.2	Firebase	16
6.2	Άδειες χρήσης	16
6.3	Dependencies	17
6.4	Υποστήριξη ελληνικής γλώσσας	18
6.5	Σύνδεση και εγγραφή	19
6.6	Text to Speech	21
6.7	Λειτουργία Μετάφρασης	22
6.8	Προφίλ χρήστη	28
6.9	Μορφή Βάσης	29
6.9.1	Firebase Authentication	29
6.9.2	Firebase Realtime Database	29
6.9.3	Firebase Storage	30
7	Συμπεράσματα και μελλοντικές επεκτάσεις	31
8	Παραρτήματα	32
8.1	Διάγραμμα κλάσεων σε UML	32
8.2	Ανάπτυξη ιστοσελίδας για την εμφάνιση στατιστικών	33
9	Ηλεκτρονική Βιβλιογραφία	37



1 Περίληψη

Η παρούσα πτυχιακή εργασία αναπτύχθηκε στα πλαίσια του προπτυχιακού κύκλου σπουδών του Πανεπιστημίου Πειραιά, στο τμήμα Πληροφορικής, και αποσκοπεί στην ανάπτυξη μιας Android εφαρμογής η οποία αποτελεί χρήσιμο εργαλείο στα χέρια των τουριστών όταν αυτοί έχουν επισκεφτεί μία ξένη χώρα και παρέχει δυνατότητες μετάφρασης κειμένου, αναγνώρισης μέσω κάμερας και εμφάνισης λεπτομερειών ενός αξιοθέατου και μετατροπής του συναλλάγματος. Παρακάτω θα αναλυθούν περαιτέρω ο σκοπός και η λειτουργικότητα της εφαρμογής, καθώς και οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξή της. Τέλος, θα υπάρχει αναλυτική περιγραφή του εγχειριδίου χρήστη και θα γίνει αναφορά στις δυνατότητες για μελλοντική επέκταση της εφαρμογής.

2 Abstract

This thesis was developed as part of the undergraduate course of the University of Piraeus, in the Department of Informatics, and aims to develop an Android application which is a useful tool for tourists when they have visited a foreign country and provides translation, landmark recognition through camera and a currency converter. In this paper, there will be a further analysis of the purpose and functionality of the application, as well as the technologies used for its development. Finally, there will be a detailed description of the user manual and a mention will be made of the possibilities for future expansion of the application.

3 Εισαγωγή

Η ραγδαία ανάπτυξη που γνώρισε ο σύγχρονος κόσμος σε όλους τους τεχνολογικούς τομείς, συνέβαλε στην άνθιση της βιομηχανίας του τουρισμού. Ο τουρισμός αποτελεί σημαντικό κοινωνικοοικονομικό φαινόμενο με θεαματική δυναμική, καθώς ζωτικής σημασίας είναι η συνεισφορά του τόσο στην οικονομία μιας χώρας, όσο και στην ανάδειξη του πολιτισμού της.

Αντικείμενο της παρούσας εργασίας είναι η ανάπτυξη μιας android εφαρμογής, με την ονομασία «Globe», η οποία αποτελεί ένα ψηφιακό εργαλείο υποβοήθησης τουριστών στις καθημερινές τους επαφές και συναλλαγές στην χώρα που έχουν ταξιδέψει.

Συγκεκριμένα, θα δίνει στον χρήστη τρεις βασικές δυνατότητες:

1. Αρχικά, με τη χρήση Language Identification και Translation features του ML KIT που προσφέρεται από την Firebase παρέχεται η δυνατότητα γρήγορης μετάφρασης κειμένου. Μάλιστα υποστηρίζονται 20 διαφορετικές γλώσσες καθώς και η επιλογή αυτόματης αναγνώρισης της γλώσσας του κειμένου που έχει επιλεγεί για μετάφραση. Επιπροσθέτως υποστηρίζονται λειτουργίες Text-to-Speech, ώστε ο χρήστης να υπαγορεύει το κείμενο το οποίο επιθυμεί να μεταφραστεί, και Speech-to-Text, για την εκφώνηση τόσο του αυθεντικού όσο και του μεταφρασμένου κειμένου. Τέλος, υποστηρίζεται και η αναγνώριση κειμένου μέσα από εικόνα. Η εικόνα μπορεί να είναι ήδη αποθηκευμένη στο κινητό του χρήστη ή να ληφθεί εκείνη τη στιγμή με τη χρήση της κάμερας. Έτσι, η λειτουργία αυτή καθίσταται ιδιαίτερα χρήσιμη στην ανάγνωση κάποιων πινακίδων, μενού εστιατορίων, κλπ.
2. Έπειτα, με τη χρήση των Google Cloud Vision API και Google Knowledge Graph SerpApi, ο χρήστης μπορεί να προβεί στην αναγνώριση με τη χρήση της κάμερας των αξιοθέατων και εμφάνιση πληροφοριών σχετικά με αυτά. Έτσι, το κινητό του τηλέφωνο μετατρέπεται σε έναν τουριστικό οδηγό που του προσφέρει γρήγορα και εύκολα έγκυρες πληροφορίες για το αξιοθέατο που βρίσκεται μπροστά του.
3. Τέλος, με τη χρήση του Exchange Rates Data API, υποστηρίζεται η δυνατότητα μετατροπής του συναλλάγματος με βάση την τρέχουσα συναλλαγματική ισοτιμία. Η εφαρμογή επιτρέπει στον χρήστη να διαλέξει ανάμεσα σε 20 διαφορετικά νομίσματα, παρέχοντάς του εύκολα την σχετική πληροφορία που αναζητεί.

Επιπρόσθετα, η χρήση της εφαρμογής προϋποθέτει την είσοδο του χρήστη, συμπληρώνοντας τα προσωπικά του στοιχεία. Επομένως έχουν υλοποιηθεί οι λειτουργίες εγγραφής (sign in) και σύνδεσής του (login), με τη χρήση του Firebase Authentication.

Τέλος, έχει υλοποιηθεί και μία ιστοσελίδα για την εμφάνιση των στατιστικών στοιχείων που καταγράφονται κατά την χρήση της εφαρμογής.



Εικόνα 1. Λογότυπο εφαρμογής

4 Ανασκόπηση πεδίου

Αυτό που διαφοροποιεί την εφαρμογή «Globe» από τις υπόλοιπες του είδους της, είναι το γεγονός ότι συνδυάζει πολλές λειτουργίες σε μια μόνο εφαρμογή. Αν κάποιος χρήστης περιηγηθεί σε ένα ηλεκτρονικό κατάστημα εφαρμογών (πχ. Play Store) θα ανακαλύψει παρόμοιες εφαρμογές που επιτελούν μια μόνο λειτουργία κάθε φορά.

Συγκεκριμένα, μετά από έρευνα βρέθηκαν οι παρακάτω συναφείς εφαρμογές:

1. [Google Translate](#)

Η εφαρμογή αυτή αποτελεί την πιο δημοφιλή εφαρμογή μετάφρασης στον ψηφιακό κόσμο. Η Google προσφέρει δωρεάν αυτή την υπηρεσία και με την οποία λέξεις, φράσεις και ιστοσελίδες μεταφράζονται σε περισσότερες από 100 γλώσσες και διαλέκτους.

2. [Google Lens](#)

Η Google έχει αναπτύξει ακόμα μία ευρέως γνωστή εφαρμογή, η οποία επιτρέπει στους χρήστες της να αναγνωρίσουν αξιοθέατα ανά τον κόσμο, μέσω της οπτικής ανάλυσής τους, και στην συνέχεια να αναζητήσουν περεταίρω πληροφορίες για αυτά.

3. [Currency Converter](#)

Η συγκεκριμένη εφαρμογή αποτελεί ένα παράδειγμα εφαρμογής υπολογισμού ισοτιμιών συναλλάγματος από μία πληθώρα εφαρμογών που κυκλοφορούν. Δίνει την δυνατότητα στους χρήστες της να μετατρέψουν το συνάλλαγμά τους και να ενημερωθούν σχετικά με την αξία του κάθε νομίσματος μέσω διαγραμμάτων.

5 Παρουσίαση και χρήση εφαρμογής

Κατά την εκκίνηση της εφαρμογής, εμφανίζεται στην οθόνη του χρήστη το animated λογότυπό της, όπως και ένα σλόγκαν που καλωσορίζει τον χρήστη.



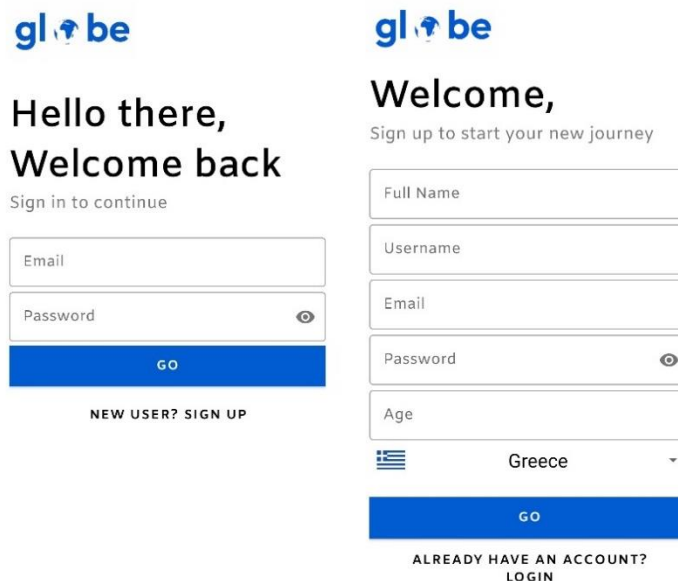
Εικόνα 2. Εκκίνηση εφαρμογής

5.1 Εγγραφή και σύνδεση στην εφαρμογή

Στη συνέχεια, εμφανίζεται η φόρμα σύνδεσης, στην οποία ζητείται η συμπλήρωση των στοιχείων του χρήστη, εφόσον έχει ήδη προηγηθεί η εγγραφή του στην εφαρμογή. Συγκεκριμένα, απαιτείται η συμπλήρωση του email και του κωδικού πρόσβασης, ώστε να συνδεθεί στο λογαριασμό του. Αν δεν έχει προηγηθεί δημιουργία λογαριασμού, μπορεί να πατήσει το κουμπί «New user? Sign up» και να μεταφερθεί στην φόρμα του sign up για να πραγματοποιήσει την εγγραφή του.

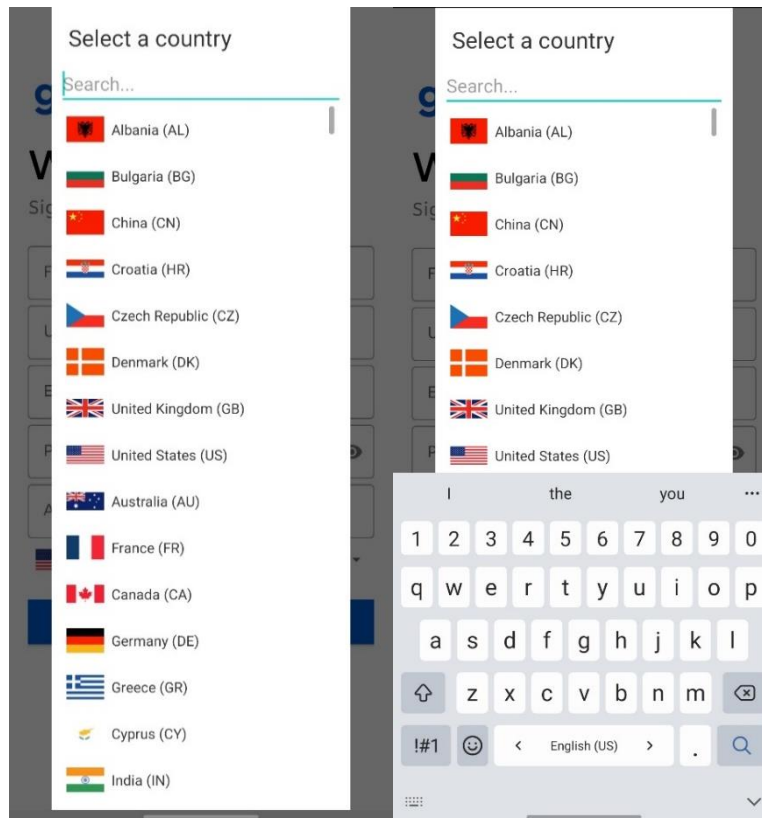
Για την εγγραφή του χρήστη απαιτείται η ορθή συμπλήρωση των στοιχείων του. Αναλυτικότερα, ζητούνται το ονοματεπώνυμό, το όνομα χρήστη, το email, ο επιθυμητός κωδικός πρόσβασης, η ηλικία και η χώρα διαμονής του εκάστοτε χρήστη. Η επιλογή της χώρας γίνεται μέσω αναζήτησής της στο pop up παράθυρο, είτε κάνοντας scroll το παράθυρο προς τα κάτω και επιλέγοντας την επιθυμητή χώρα, είτε αναζητώντας την στην φόρμα αναζήτησης στο πάνω μέρος της οθόνης. Αξίζει να σημειωθεί πως υπάρχει αυτόματος έλεγχος της ορθότητας κάθε στοιχείου, καθώς η συμπλήρωση όλων των πεδίων είναι υποχρεωτική.

Ύστερα από την καταγραφή των στοιχείων του και την δημιουργία του λογαριασμού του με το πάτημα του κουμπιού «Go», ο χρήστης ανακατευθύνεται στο κεντρικό μενού της εφαρμογής.

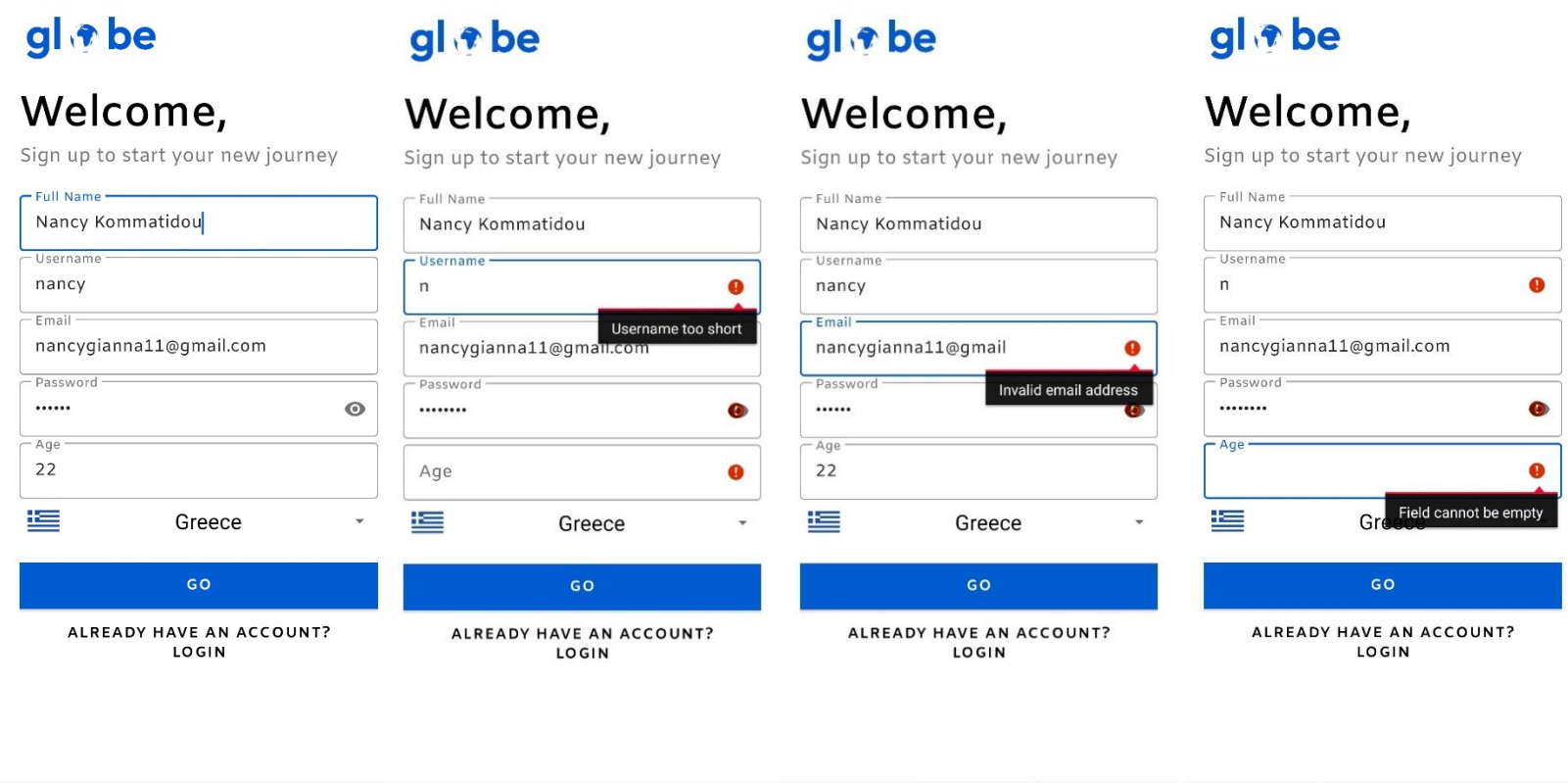


The image displays two side-by-side screenshots of the 'glabe' application interface. The left screenshot shows the login screen with the text 'Hello there, Welcome back' and a 'Sign in to continue' prompt. It features input fields for 'Email' and 'Password', a 'GO' button, and a link for 'NEW USER? SIGN UP'. The right screenshot shows the registration screen with the text 'Welcome, Sign up to start your new journey'. It includes input fields for 'Full Name', 'Username', 'Email', 'Password', and 'Age', a dropdown menu for selecting a country (currently set to 'Greece'), a 'GO' button, and a link for 'ALREADY HAVE AN ACCOUNT? LOGIN'.

Εικόνες 3&4. Σύνδεση και εγγραφή χρήστη



Εικόνες 5&6. Αναζήτηση χώρας διαμονής



Εικόνες 7,8,9,10. Έλεγχοι συμπλήρωσης πεδίων φόρμας εγγραφής

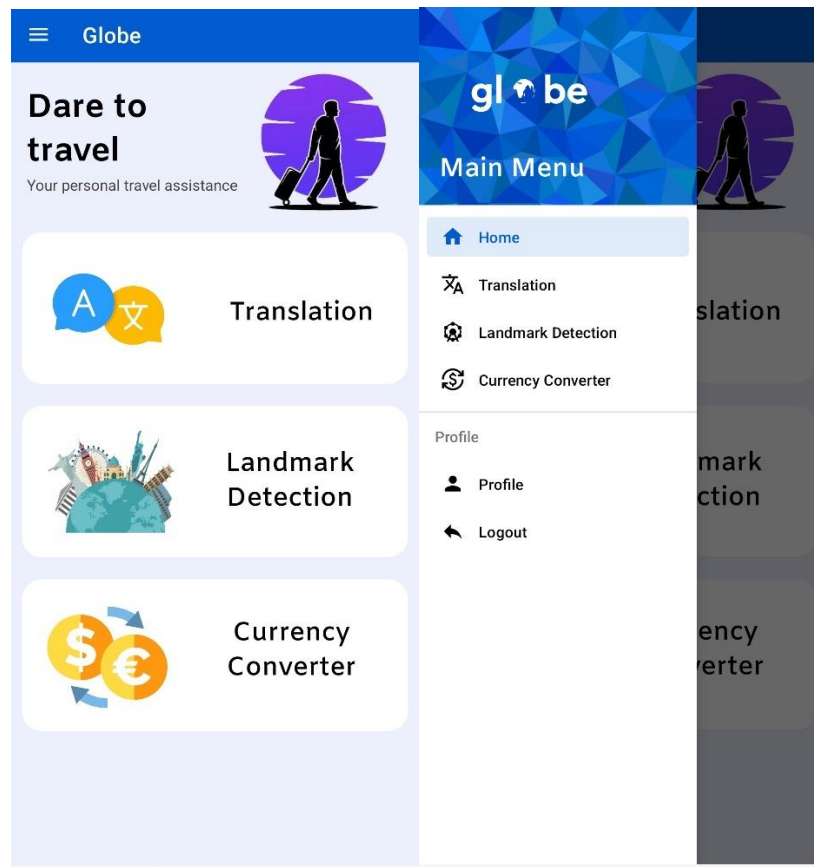
5.2 Κεντρικό μενού εφαρμογής

Ο χρήστης μπορεί να πλοηγηθεί στο κεντρικό μενού της εφαρμογής το οποίο αποτελείται από τρεις επιλογές:

1. Μετάφραση (Translation)
2. Αναγνώριση αξιοθέατων (Landmark Detection)
3. Μετατροπέας συναλλάγματος (Currency Converter)

Επίσης, για ευκολότερη περιήγηση υπάρχει και ένα πλαϊνό μενού, στο οποίο εμφανίζονται δύο ακόμη επιλογές:

1. Προβολή προφίλ (Profile)
2. Αποσύνδεση (Logout)

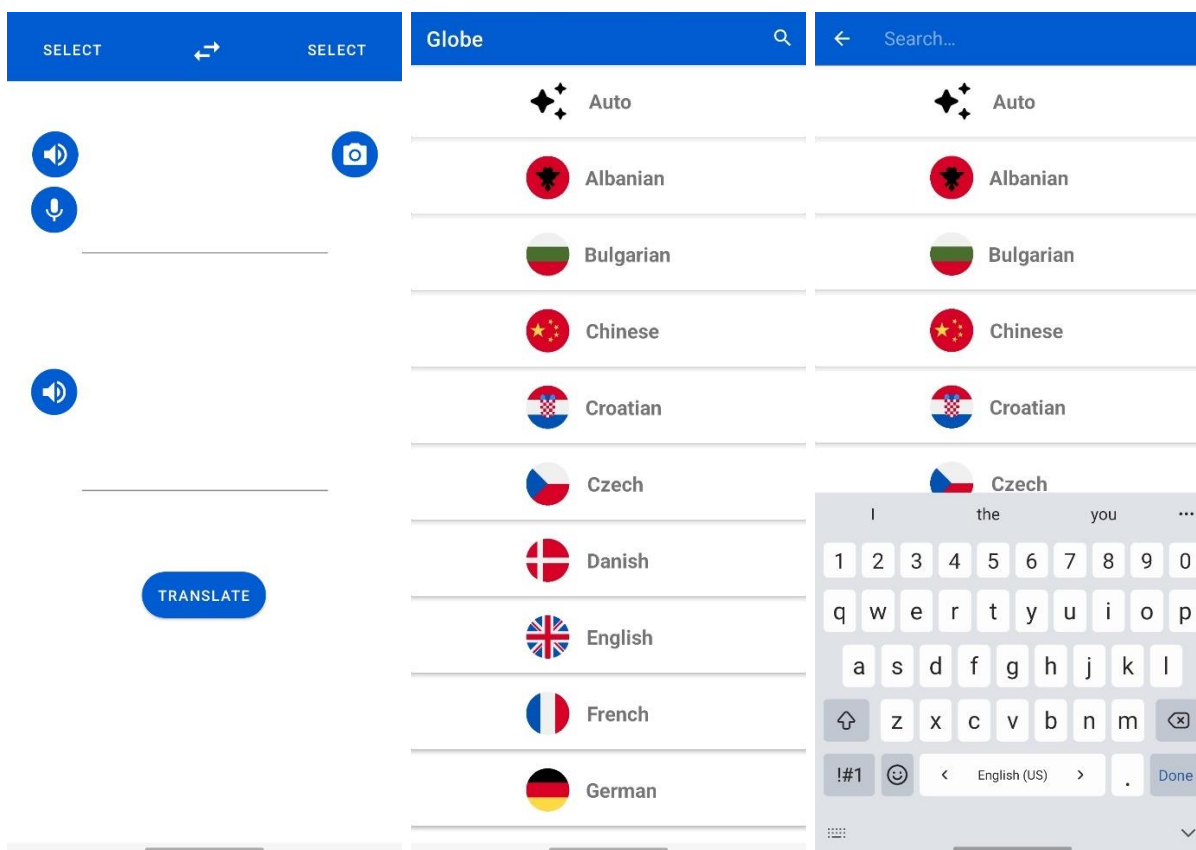


Εικόνες 11&12. Κεντρικό μενού, μενού πλοήγησης

5.3 Λειτουργία Μετάφρασης

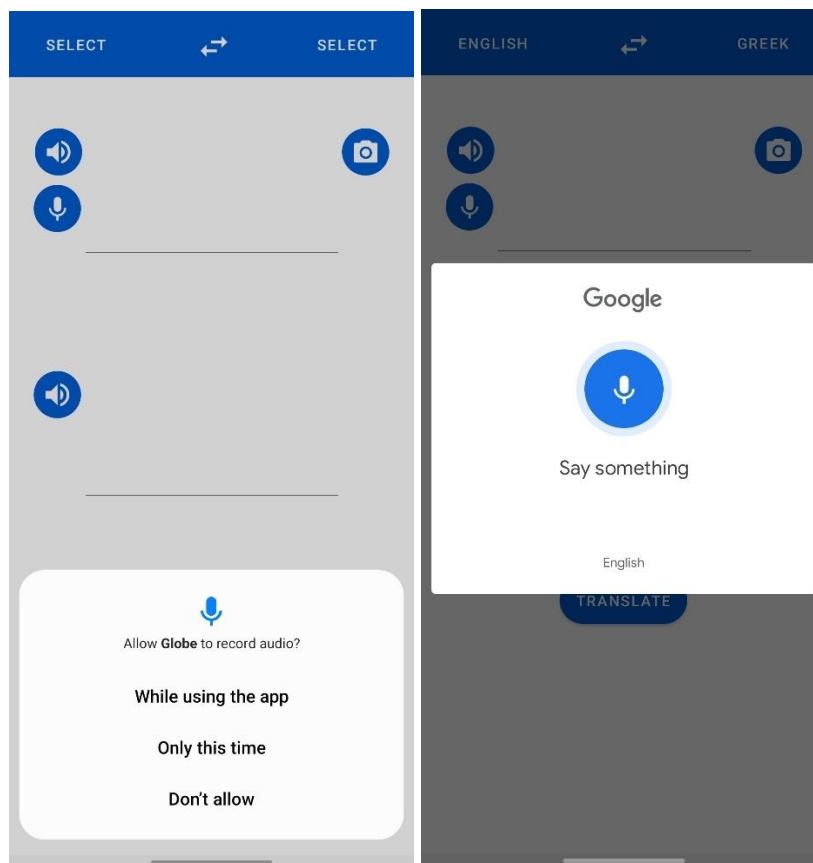
Στο πάνω μέρος της οθόνης ο χρήστης μπορεί να επιλέξει την γλώσσα στην οποία βρίσκεται το κείμενο που θέλει να μεταφράσει, επιλέγοντας το «Select» στα αριστερά, καθώς και την γλώσσα στην οποία επιθυμεί να μεταφραστεί το κείμενο αυτό, επιλέγοντας το «Select» στα δεξιά. Το κουμπί που απεικονίζει τα δύο ανάποδα βελάκια και βρίσκεται ανάμεσα στα δύο Select, δίνει στον χρήστη την δυνατότητα της αντικατάστασης της αριστερής γλώσσας μετάφρασης με την δεξιά και αντίθετα.

Η εφαρμογή επιτρέπει μετάφραση ανάμεσα σε 20 διαθέσιμες γλώσσες, και υποστήριξη αυτόματης αναγνώρισης γλώσσας, σε περίπτωση που ο χρήστης δεν γνωρίζει την γλώσσα στην οποία είναι γραμμένο το κείμενο που θέλει να μεταφράσει. Επίσης, στην φόρμα επιλογής γλώσσας υπάρχει πεδίο αναζήτησης για ευκολότερη εύρεση της επιθυμητής γλώσσας.



Εικόνες 13&14. Λειτουργία μετάφρασης και επιλογή γλώσσας

Ο χρήστης μπορεί είτε να πληκτρολογήσει το κείμενο προς μετάφραση στο πάνω πλαίσιο κειμένου, είτε να το εκφωνήσει μετά το πάτημα του κουμπιού με την ένδειξη του μικροφώνου στα αριστερά, ενεργοποιώντας έτσι μία λειτουργία speech to text, για την χρήση της οποίας απαιτείται η αποδοχή της άδειας χρήσης του μικροφώνου από την εφαρμογή.



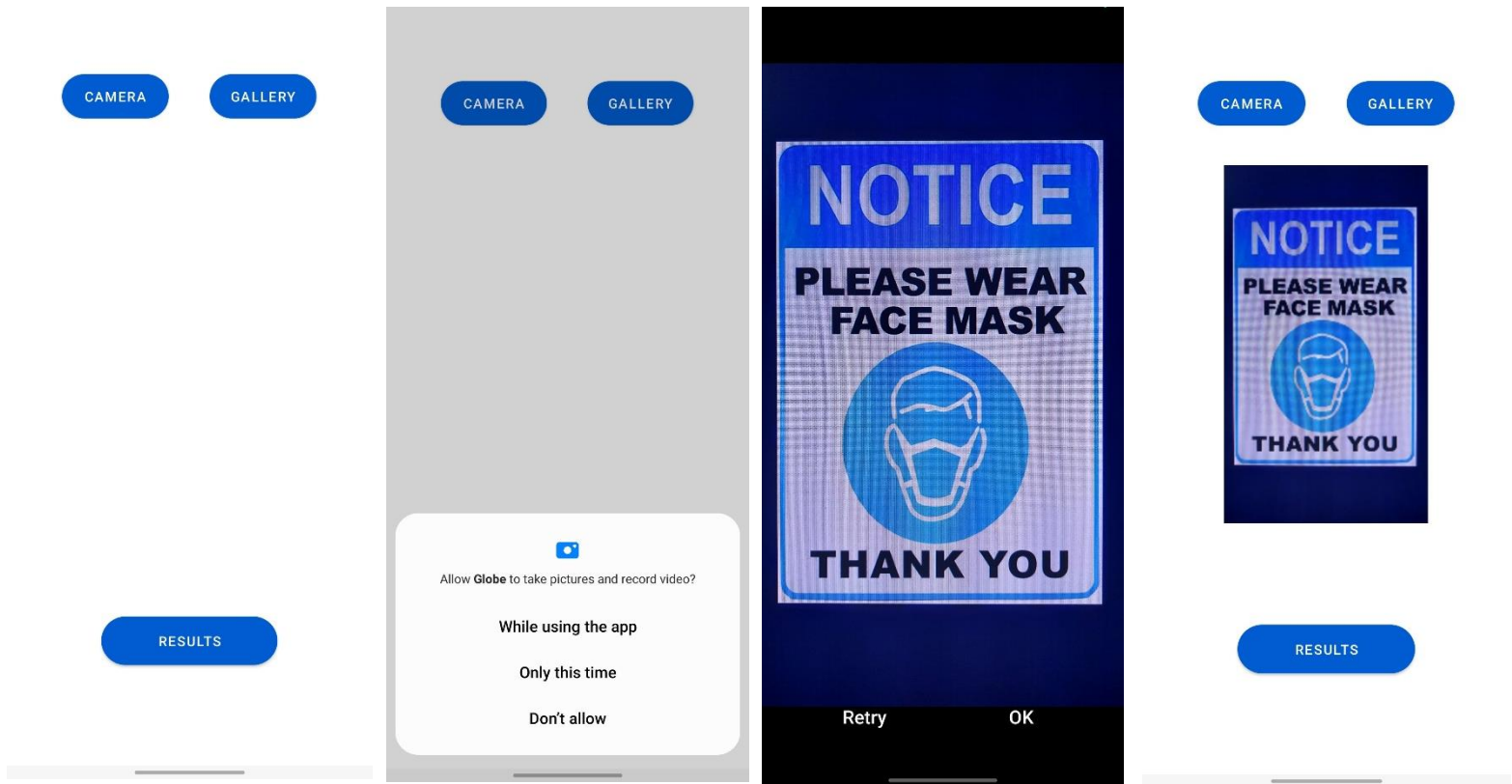
Εικόνες 15&16. Άδεια και λειτουργία εκφώνησης κειμένου

Επίσης, προσφέρεται η δυνατότητα αναγνώρισης κειμένου μέσα από εικόνα, με το πάτημα του κουμπιού με την ένδειξη της κάμερας. Ο χρήστης έπειτα καλείται να επιλέξει αν επιθυμεί να εξάγει το κείμενο από μία εικόνα που θα τραβήξει εκείνη την στιγμή, πατώντας το κουμπί «Camera», ή από μία που έχει ήδη αποθηκευμένη στη συλλογή του, πατώντας το κουμπί «Gallery».

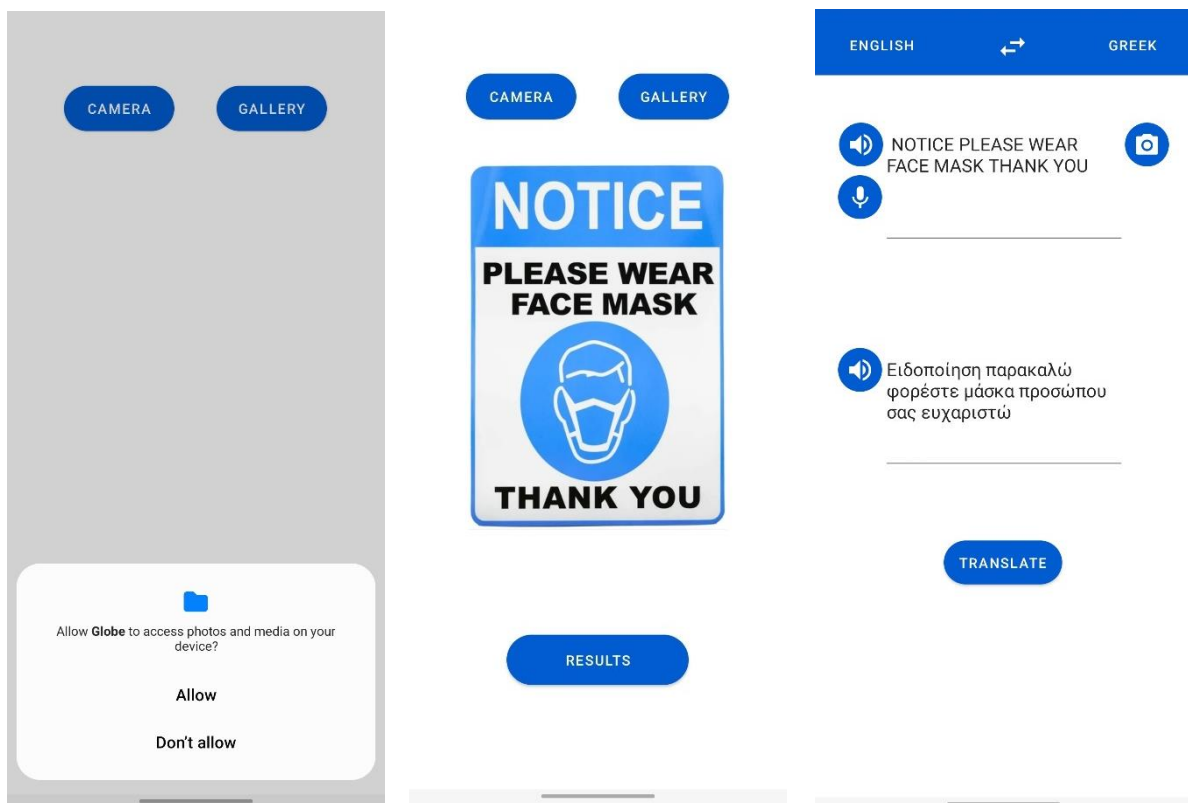
Αφού τραβήξει ή επιλέξει μία εικόνα, καλείται να πατήσει το κουμπί «Results» για να εξάγει το κείμενό της και έπειτα να το εμφανίσει στο πλαίσιο κειμένου προς μετάφραση.

Για την χρήση της κάμερας και την πρόσβαση της εφαρμογής στα αρχεία του χρήστη απαιτείται κάθε φορά η αντίστοιχη αποδοχή της άδειας χρήσης.

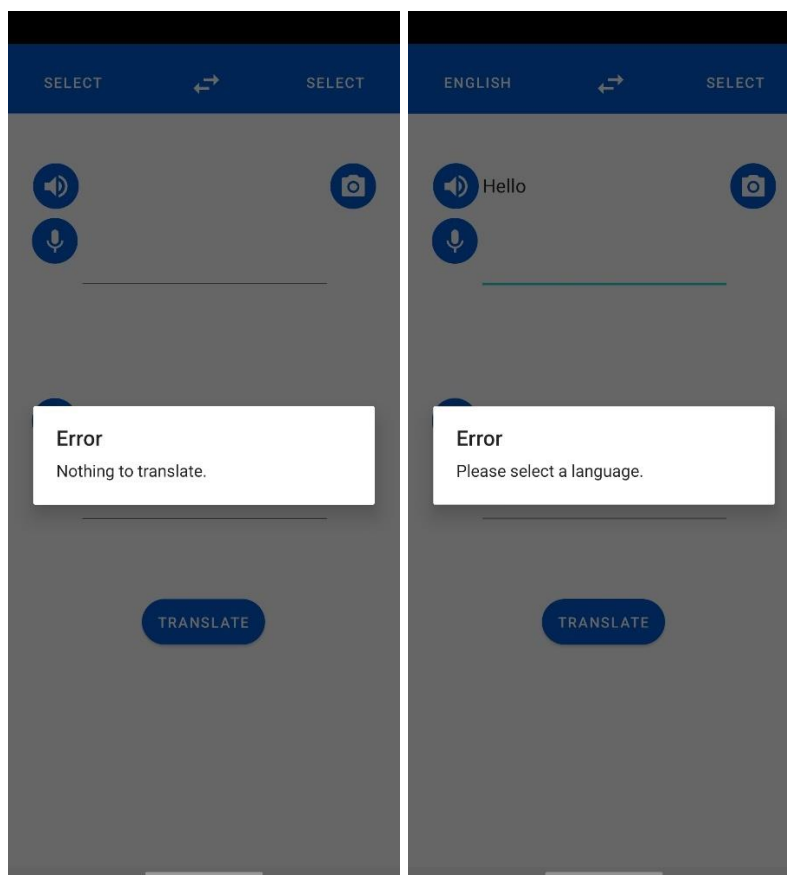
Τέλος, η μετάφραση πραγματοποιείται με το πάτημα του κουμπιού «Translate», και έχουν υλοποιηθεί οι κατάλληλοι έλεγχοι συμπλήρωσης κειμένου και επιλογής γλώσσας, καθώς η ύπαρξή τους είναι αναγκαία για να γίνει η μετάφραση.



Εικόνες 17,18,19,20. Άδεια και λειτουργία λήψης εικόνας



Εικόνες 21,22,23. Άδεια και λειτουργία εισαγωγής εικόνας



Εικόνες 24&25. Έλεγχοι συμπλήρωσης πεδίων

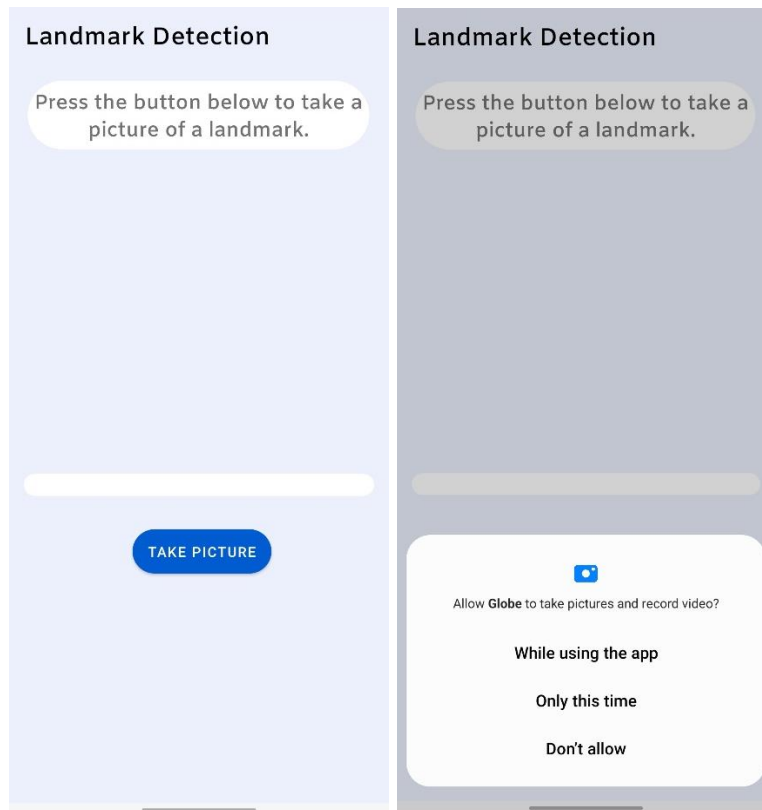
5.4 Λειτουργία Αναγνώρισης Αξιοθέατων

Στο κάτω μέρος της οθόνης ο χρήστης μπορεί να επιλέξει την λήψη μίας εικόνας από ένα αξιοθέατο το οποίο θέλει να αναγνωρίσει ή να μάθει περισσότερες πληροφορίες, επιλέγοντας το κουμπί «Take picture».

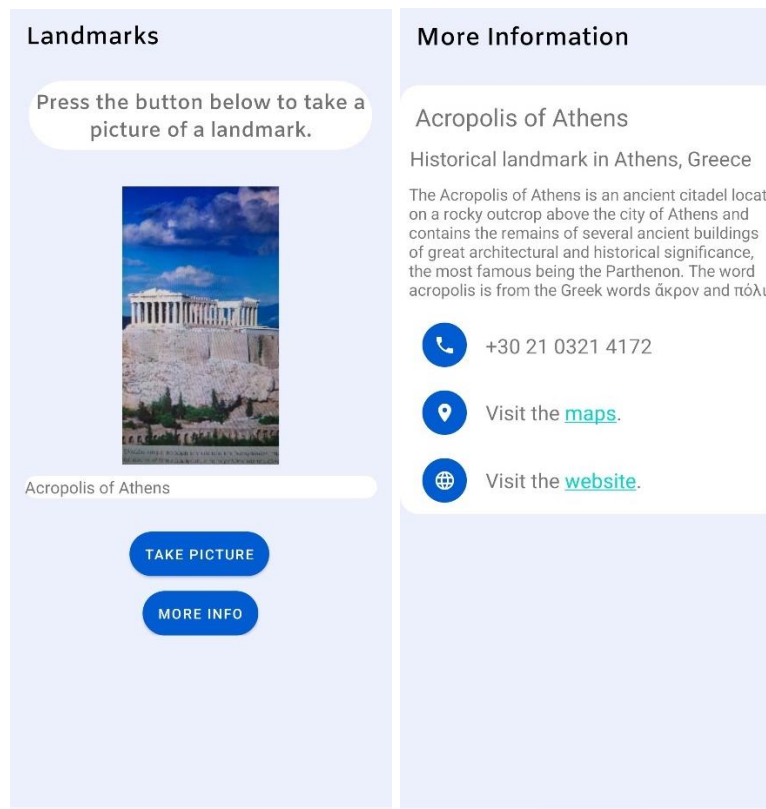
Για την χρήση της κάμερας απαιτείται η αντίστοιχη αποδοχή της άδειας χρήσης της.

Αυτόματα με την λήψη της φωτογραφίας, εμφανίζεται η εικόνα που τραβήχτηκε στο κέντρο της οθόνης και στο κάτω πλαίσιο το όνομα του αξιοθέατου που αναγνωρίστηκε. Αν δεν αναγνωρίστηκε κάποιο, τότε εμφανίζεται το κατάλληλο μήνυμα.

Αν ο χρήστης επιθυμεί την εύρεση περισσότερων σχετικών πληροφοριών, μπορεί να πατήσει το κουμπί «More info» και να ανακατευθυνθεί σε μία φόρμα όπου θα αναγράφονται πληροφορίες για το αξιοθέατο, καθώς και ένα τηλέφωνο επικοινωνίας και δύο σύνδεσμοι. Ο πρώτος σύνδεσμος εμφανίζει την τοποθεσία του αξιοθέατου στον χάρτη, και ο δεύτερος ανοίγει την ιστοσελίδα που του αντιστοιχεί.



Εικόνες 26,27. Αναγνώριση αξιοθέατων και άδεια κάμερας



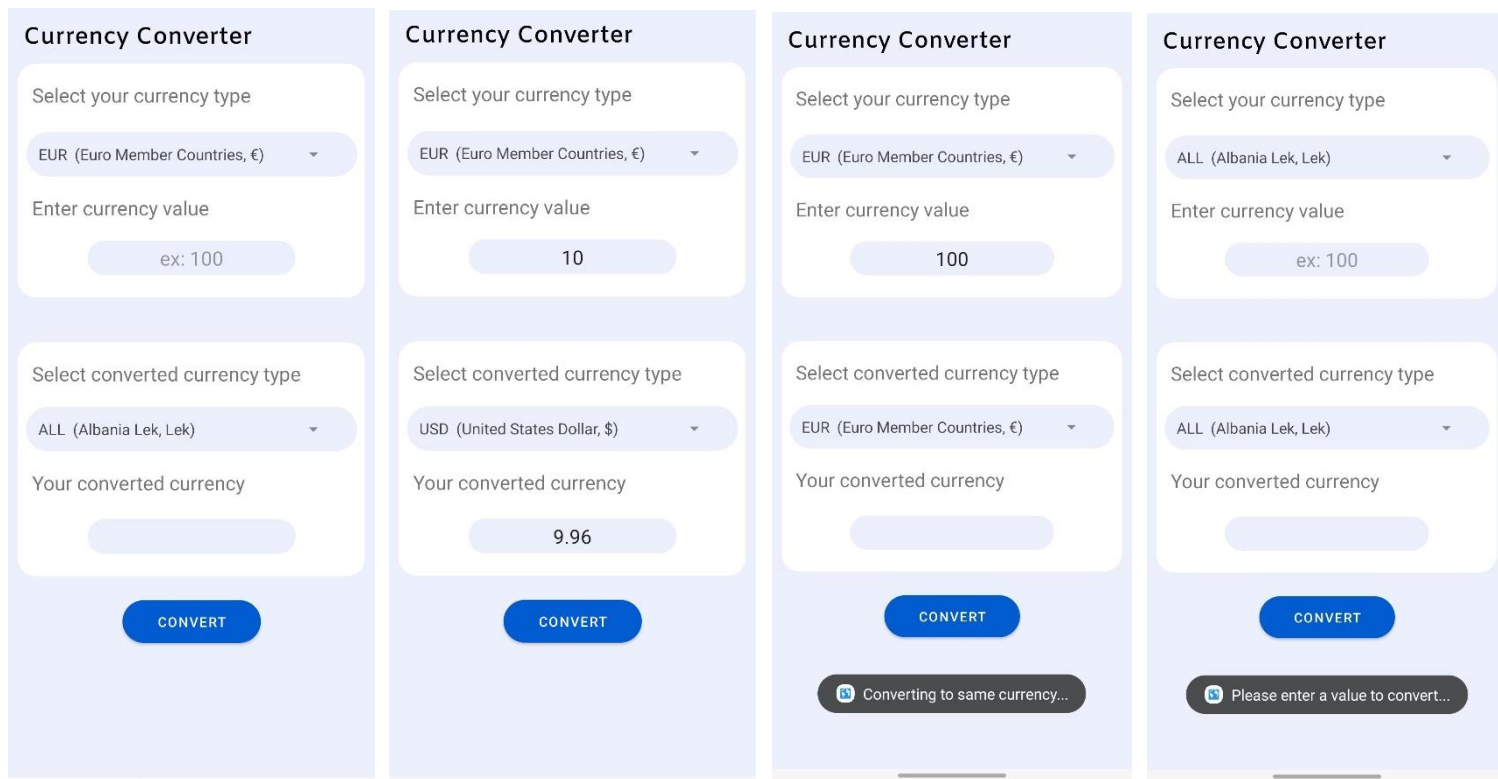
Εικόνες 28,29. Αποτελέσματα αναγνώρισης αξιοθέατου

5.5 Λειτουργία Μετατροπής συναλλάγματος

Στο πάνω μέρος του μετατροπέα συναλλάγματος, ο χρήστης καλείται να επιλέξει τον τύπο του νομίσματος που επιθυμεί να τροποποιήσει, όπως και το χρηματικό ποσό. Μπορεί να επιλέξει ανάμεσα σε 20 διαφορετικά συναλλάγματα, πατώντας το dropdown μενού.

Έπειτα, απαιτείται η επιλογή του επιθυμητού συναλλάγματος, και με το πάτημα του κουμπιού «Convert» υπολογίζεται η αντιστοίχιση του ποσού που ζητήθηκε.

Τέλος, υπάρχουν οι κατάλληλοι έλεγχοι συμπλήρωσης όλων των πεδίων, ενώ δεν επιτρέπεται η επιλογή του ίδιου τύπου συναλλάγματος.



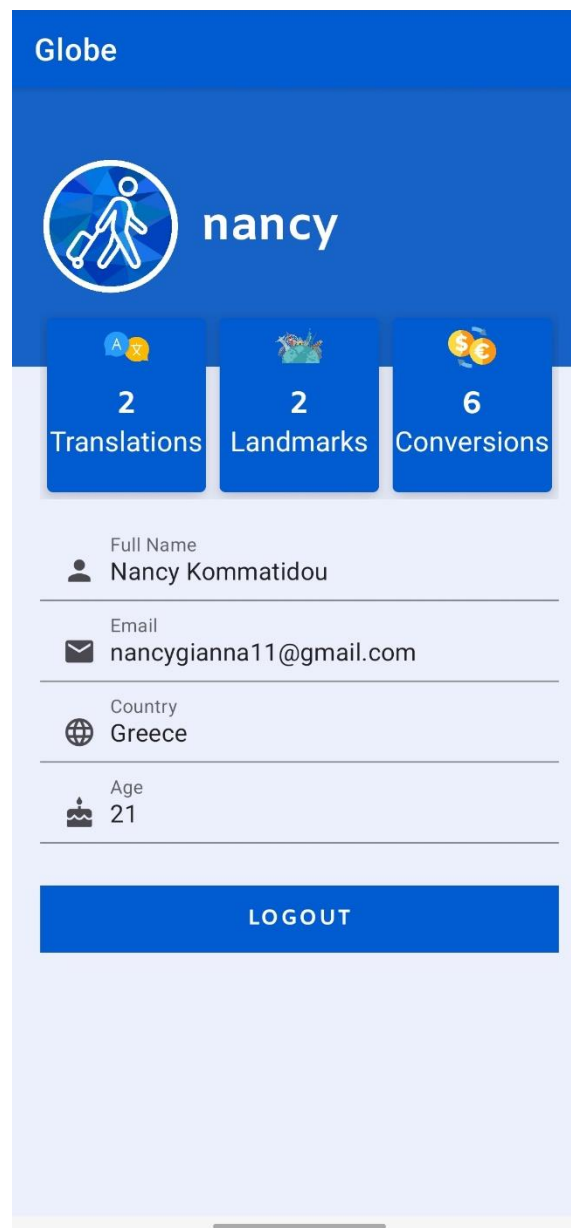
The figure displays four sequential screenshots of a mobile application titled "Currency Converter". Each screen shows the following elements: a "Select your currency type" dropdown menu, an "Enter currency value" input field, a "Select converted currency type" dropdown menu, and a "Your converted currency" output field. A blue "CONVERT" button is located at the bottom of each screen.

- Screen 1:** The first dropdown is set to "EUR (Euro Member Countries, €)" and the input field contains "ex: 100". The second dropdown is set to "ALL (Albania Lek, Lek)". The output field is empty.
- Screen 2:** The first dropdown is set to "EUR (Euro Member Countries, €)" and the input field contains "10". The second dropdown is set to "USD (United States Dollar, \$)". The output field displays "9.96".
- Screen 3:** The first dropdown is set to "EUR (Euro Member Countries, €)" and the input field contains "100". The second dropdown is also set to "EUR (Euro Member Countries, €)". Below the "CONVERT" button, a grey message box states "Converting to same currency...".
- Screen 4:** The first dropdown is set to "ALL (Albania Lek, Lek)" and the input field contains "ex: 100". The second dropdown is set to "ALL (Albania Lek, Lek)". Below the "CONVERT" button, a grey message box states "Please enter a value to convert...".

Εικόνες 30,31,32,33. Μετατροπέας συναλλάγματος και έλεγχοι

5.6 Προφίλ χρήστη

Διαλέγοντας την επιλογή «Profile» από το μενού πλοήγησης, ο χρήστης έχει την δυνατότητα να επισκεφτεί το προφίλ του. Εδώ μπορεί να ενημερωθεί σχετικά με το πλήθος των ενεργειών που έχει προβεί, όσον αφορά τις δυνατότητες της εφαρμογής. Συγκεκριμένα, στο πάνω μέρος της οθόνης εμφανίζεται το όνομα χρήστη, και ακολουθούν το πλήθος των μεταφράσεων, της αναγνώρισης αξιοθέατων και της μετατροπής συναλλάγματος, καθώς και τα προσωπικά στοιχεία του. Τέλος, στο κάτω μέρος της οθόνης υπάρχει και η επιλογή «Logout» με την οποία ο χρήστης μπορεί να αποσυνδεθεί από τον λογαριασμό του και να οδηγηθεί στην αρχική σελίδα σύνδεσης στην εφαρμογή.



Εικόνα 34. Προφίλ χρήστη

6 Αρχιτεκτονική συστήματος

6.1 Εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής

6.1.1 Android studio

Η εφαρμογή αναπτύχθηκε εξ ολοκλήρου σε περιβάλλον Android Studio (έκδοση Chipmunk | 2021.2.1 Patch 2), ένα πρόγραμμα σχεδιασμένο για την ανάπτυξη εφαρμογών στην πλατφόρμα Android. Το πρόγραμμα υποστηρίζει την χρήση 2 γλωσσών προγραμματισμού, της Java και της Kotlin. Ωστόσο, η παρούσα πτυχιακή είναι γραμμένη σε Java.



Εικόνα 35. Android studio

6.1.2 Firebase

Η Firebase είναι μια πλατφόρμα που αναπτύχθηκε από την Google για τη δημιουργία εφαρμογών για κινητά και εφαρμογών ιστού. Παρέχει μία πληθώρα δυνατοτήτων, με κυριότερες την λειτουργία του Authentication, την Realtime Database και το Storage.

Αναλυτικότερα,

1. Το Firebase Authentication παρέχει υπηρεσίες υποστήριξης, εύχρηστα SDK και έτοιμες βιβλιοθήκες διεπαφής χρήστη για τον έλεγχο ταυτότητας των χρηστών της εφαρμογής. Παρέχει μεθόδους δημιουργίας και διαχείρισης χρηστών που χρησιμοποιούν τις διευθύνσεις email και τους κωδικούς πρόσβασης για να συνδεθούν, καθώς και αριθμούς τηλεφώνου, δημοφιλείς παρόχους ταυτοποίησης όπως το Google, το Facebook και το Twitter και πολλά άλλα.
2. Η Firebase Realtime Database είναι μια cloud βάση δεδομένων. Τα δεδομένα αποθηκεύονται σε μορφή JSON και συγχρονίζονται σε πραγματικό χρόνο με κάθε συνδεδεμένο πελάτη.
3. Το Cloud Storage της Firebase είναι μια ασφαλής υπηρεσία για την αποθήκευση εικόνων, ήχου, βίντεο ή άλλου περιεχομένου που δημιουργείται από χρήστες.



Εικόνα 36. Firebase

6.2 Άδειες χρήσης

Η εφαρμογή «Globe» προσφέρει πολλές λειτουργίες οι οποίες προϋποθέτουν την παροχή αντίστοιχης άδειας χρήσης από τον εκάστοτε χρήστη. Τέτοιου είδους εξουσιοδότηση απαιτείται:

1. Για την χρήση της κάμερας, όταν ο χρήστης επιθυμεί να προβεί στην εξαγωγή κειμένου από μία εικόνα ή να αναγνωρίσει ένα αξιοθέατο
2. Για την χρήση του μικροφώνου, όταν ο χρήστης θέλει να εκφωνήσει το κείμενο το οποίο πρόκειται να μεταφραστεί

3. Για την χρήση της εσωτερικής μνήμης του κινητού τηλεφώνου και την ανάγνωση των αρχείων που αυτή περιέχει, όταν γίνεται επιλογή μίας ήδη αποθηκευμένης εικόνας, το κείμενο της οποίας θέλει να μεταφραστεί
4. Για την χρήση του διαδικτύου, η οποία είναι βασική προϋπόθεση για την χρήση της εφαρμογής καθώς απαιτείται για κάθε βασική λειτουργία της (σύνδεση/εγγραφή, μετάφραση, αναγνώριση αξιοθέατου, μετατροπή συναλλάγματος, κλπ)

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.CAMERA" />
<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" />
```

Εικόνα 37. Permissions

6.3 Dependencies

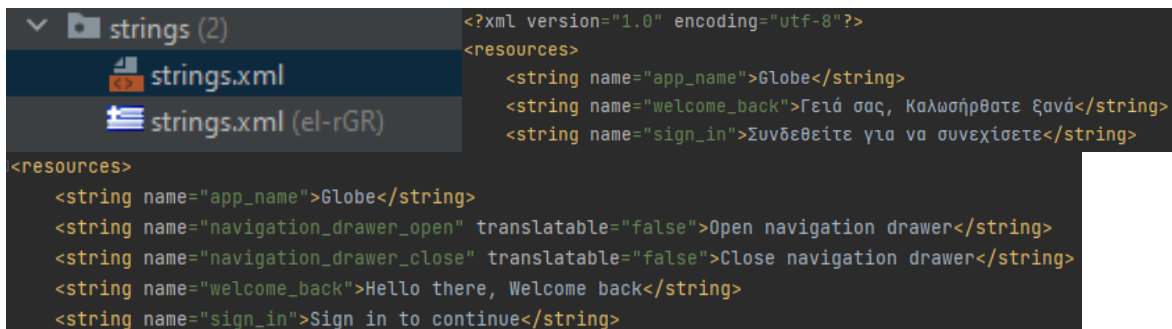
Μέσα στο *build.gradle* αρχείο της εφαρμογής, γίνεται υλοποίηση κάποιων dependencies που σχετίζονται με εξωτερικές βιβλιοθήκες, απαραίτητες για την λειτουργία της εφαρμογής, με κυριότερες αυτές που αναφέρονται σε στοιχεία της firebase και του ml kit, καθώς αντιστοιχούν στις βασικότερες λειτουργίες της εφαρμογής.

```
dependencies {
    implementation 'androidx.appcompat:appcompat:1.3.0'
    implementation 'com.google.android.material:material:1.4.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation 'com.google.firebase:firebase-database:20.0.4'
    implementation 'com.google.firebase:firebase-auth:21.0.1'
    implementation 'com.google.firebase:firebase-storage:20.0.1'
    implementation 'com.google.mlkit:translate:17.0.0'
    implementation 'com.google.mlkit:language-id:17.0.3'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
    implementation 'pl.droidsonroids.gif:android-gif-drawable:1.2.19'
    implementation 'com.google.android.material:material:1.6.1'
    implementation 'com.hbb20:ccp:2.6.0'
    implementation 'com.github.bumptech.glide:glide:4.12.0'
    implementation 'com.google.api-client:google-api-client-android:1.20.0' exclude module: 'httpclient'
    implementation 'com.google.http-client:google-http-client-gson:1.20.0' exclude module: 'httpclient'
    implementation 'com.google.apis:google-api-services-vision:v1-rev2-1.21.0' exclude module: 'httpclient'
    implementation 'com.google.guava:listenablefuture:9999.0-empty-to-avoid-conflict-with-guava'
    implementation 'com.jakewharton:butterknife:10.0.0'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.12.0'
    annotationProcessor 'com.jakewharton:butterknife-compiler:10.0.0'
    // Text features
    implementation 'com.google.android.gms:play-services-mlkit-text-recognition:18.0.0'
}
```

Εικόνα 38. Dependencies

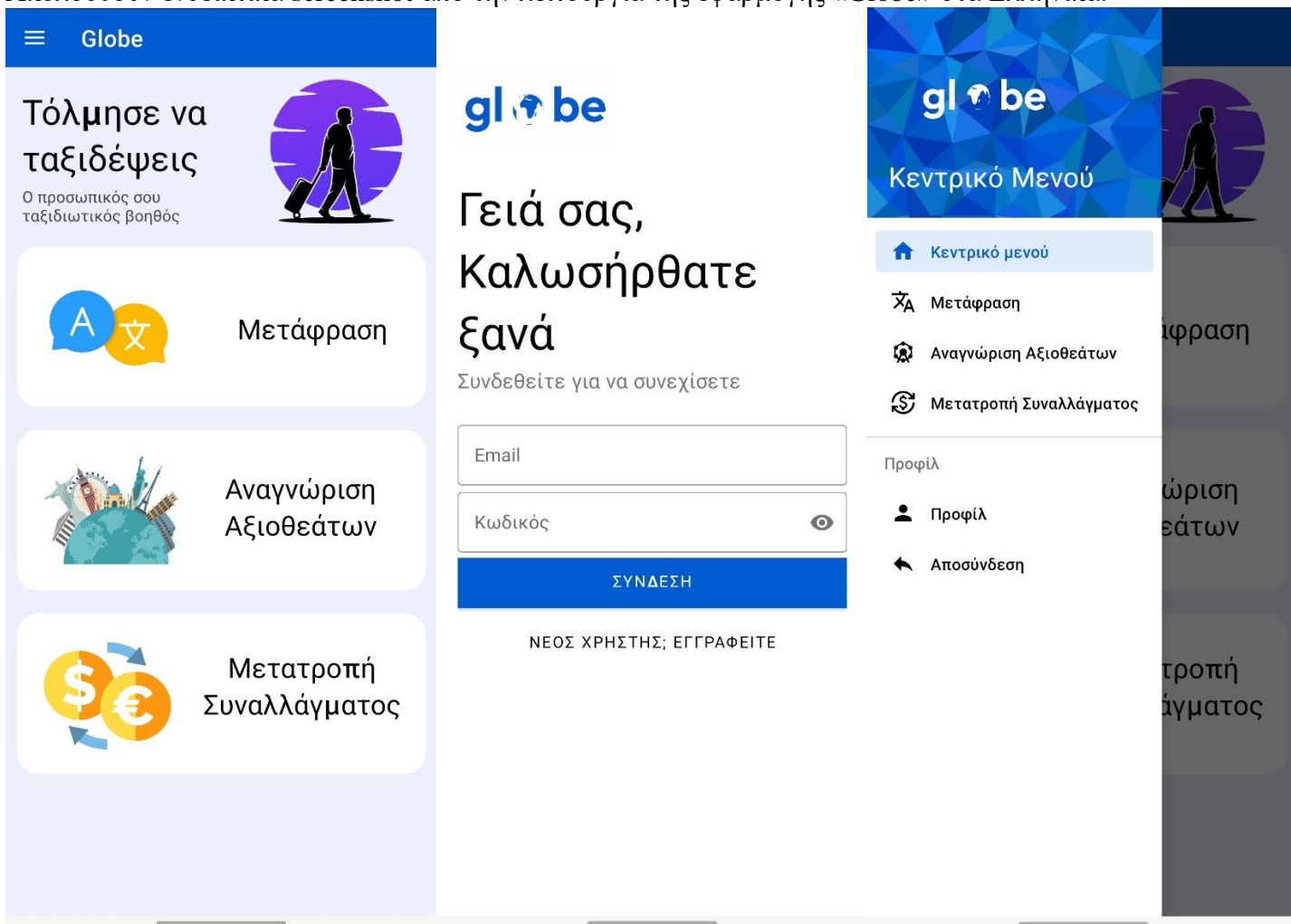
6.4 Υποστήριξη ελληνικής γλώσσας

Κάθε συμβολοσειρά που χρησιμοποιείται στην εφαρμογή προέρχεται από το *string.xml* αρχείο του project. Με τη χρήση του Translations Editor που προσφέρει το Android Studio, επιλέγουμε σαν δεύτερη γλώσσα της εφαρμογής τα Ελληνικά και δημιουργείται ένα δεύτερο αρχείο *string.xml* το οποίο περιέχει τις μεταφράσεις των κειμένων που είναι αποθηκευμένες στο αρχικό *string.xml*, όπως φαίνεται παρακάτω.



Εικόνες 39,40,41. string.xml (Ελληνικά και Αγγλικά)

Ακολουθούν ενδεικτικά screenshot από την λειτουργία της εφαρμογής «Globe» στα Ελληνικά.



Εικόνες 42,43,44. Λειτουργία εφαρμογής στα Ελληνικά

6.5 Σύνδεση και εγγραφή

Η χρήση της εφαρμογής απαιτεί την σύνδεση του χρήστη στον προσωπικό του λογαριασμό, είτε την εγγραφή του συμπληρώνοντας τα στοιχεία που του ζητούνται.

Το activity *Authentication.java* είναι υπεύθυνο για την σύνδεση του χρήστη στον ήδη υπάρχοντα λογαριασμό του. Αρχικά, δημιουργείται και αρχικοποιείται ένα αντικείμενο τύπου FirebaseAuth *mAuth*.

Η μέθοδος *login* αρχικά ελέγχει ότι έχουν συμπληρωθεί τα δύο υποχρεωτικά πεδία για την σύνδεση του χρήστη (email, κωδικός πρόσβασης), και έπειτα καλείται η συνάρτηση *signInWithEmailAndPassword()* της Firebase η οποία ελέγχει την ορθότητα και την εγκυρότητα των στοιχείων. Αν τα στοιχεία αντιστοιχούν σε έναν εγγεγραμμένο χρήστη, τότε συνδέεται επιτυχώς στον λογαριασμό του και ανακατευθύνεται στο κεντρικό μενού της εφαρμογής. Σε αντίθετη περίπτωση εμφανίζεται το κατάλληλο μήνυμα λάθους.

```
private FirebaseAuth mAuth; mAuth = FirebaseAuth.getInstance();

// Method for user login
public void login(View view){
    // If fields are complete
    if (!email.getText().toString().isEmpty() && !password.getText().toString().isEmpty()) {
        mAuth.signInWithEmailAndPassword(email.getText().toString(), password.getText().toString())
            .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful()){
                        // Go to next activity
                        Intent intent = new Intent( packageContext Authentication.this, MainMenu.class);
                        startActivity(intent);
                    } else {
                        showMessage("Error", task.getException().getLocalizedMessage());
                    }
                }
            });
    } else {
        showMessage("Error", "Please fill all the fields");
    }
}
```

Εικόνα 45. Firebase Authentication – Login

Το activity *SignUp.java* είναι υπεύθυνο για την εγγραφή του χρήστη στην εφαρμογή. Αρχικά, δημιουργείται και αρχικοποιείται ένα αντικείμενο τύπου FirebaseAuth *mAuth*, καθώς και μία αναφορά στην βάση δεδομένων.

Η μέθοδος *signup()* αρχικά ελέγχει ότι έχουν συμπληρωθεί τα πέντε υποχρεωτικά πεδία για την εγγραφή του χρήστη (ονοματεπώνυμο, όνομα χρήστη, email, κωδικός πρόσβασης, ηλικία), και έπειτα καλείται η συνάρτηση *createUserWithEmailAndPassword()* της Firebase, η οποία δημιουργεί έναν νέο χρήστη, αν αυτός δεν υπάρχει ήδη στη βάση, όπου και εμφανίζεται το κατάλληλο μήνυμα λάθους. Έπειτα, εγγράφονται στην Realtime database της Firebase τα στοιχεία του χρήστη κάτω από το κλαδί «Users». Αν η εγγραφή ήταν επιτυχής, τότε ο χρήστης συνδέεται στο λογαριασμό που δημιούργησε και ανακατευθύνεται στο κεντρικό μενού της εφαρμογής.

```
private FirebaseAuth mAuth; private FirebaseAuth mAuth;
DatabaseReference database;
database = FirebaseDatabase.getInstance().getReference( path: "Users");
// Method for user sign up
public void signup(View view) {
    // Validate user input
    if(!validateName() | !validatePassword() | !validateEmail() | !validateUsername() | !validateAge()){
        return;
    }
    // If fields are complete
    mAuth.createUserWithEmailAndPassword(email.getText().toString(), password.getText().toString())
        .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()){
                    // Get the country of user
                    String country = ccp.getSelectedCountryEnglishName();
                    // Create a User object with the data provided by the user
                    User user = new User(fullname.getText().toString(), username.getText().toString(), email.getText().toString(),
                        age.getText().toString(), country, numTranslations: 0, numLandmarks: 0, numConversions: 0);
                    // Insert the new user to database
                    database.push().setValue(user);
                    // Go to next activity
                    Intent intent = new Intent( packageContext: SignUp.this, MainMenu.class);
                    startActivity(intent);
                } else {
                    showMessage(getString(R.string.error), task.getException().getLocalizedMessage());
                }
            }
        });
}
```

Εικόνα 46. Firebase Authentication – Sign up

Επίσης, έχουν υλοποιηθεί όλοι οι απαραίτητοι έλεγχοι για την ορθή και έγκυρη συμπλήρωση των στοιχείων εγγραφής του χρήστη. Συγκεκριμένα, το ονοματεπώνυμο μπορεί να αποτελείται από γράμματα (πεζά και κεφαλαία), το όνομα χρήστη από 3-15 γράμματα της αγγλικής αλφαβήτου, το email πρέπει να αποτελεί μία έγκυρη μορφή διεύθυνσης ηλεκτρονικού ταχυδρομείου, ο κωδικός πρόσβασης θα πρέπει να έχει τουλάχιστον ένα πεζό και κεφαλαίο γράμμα, τουλάχιστον έναν αριθμό και έναν ειδικό χαρακτήρα και συνολικά να είναι πάνω από 4 χαρακτήρες (τα κενά απαγορεύονται), και τέλος η ηλικία να είναι ένας αριθμός από το 0-100.

```
// ===== Methods for validating user input =====
private Boolean validateName() {
    String val = fullname.getText().toString();
    String namePattern = "[a-zA-Z ]+$";

    if (val.isEmpty()) {
        fullname.setError(getString(R.string.empty_field));
        return false;
    } else if (!val.matches(namePattern)) {
        fullname.setError(getString(R.string.only_letters));
        return false;
    } else {
        fullname.setError(null);
        return true;
    }
}

private Boolean validateUsername() {
    String val = username.getText().toString();
    String noWhiteSpace = "\\A\\w{4,20}\\z";

    if (val.isEmpty()) {
        username.setError(getString(R.string.empty_field));
        return false;
    } else if (val.length() >= 15) {
        username.setError(getString(R.string.long_username));
        return false;
    } else if (val.length() <= 3) {
        username.setError(getString(R.string.short_username));
        return false;
    } else if (!val.matches(noWhiteSpace)) {
        username.setError(getString(R.string.white_space));
        return false;
    } else {
        username.setError(null);
        return true;
    }
}

private Boolean validateEmail() {
    String val = email.getText().toString();
    String emailPattern = "[a-zA-Z0-9._-]+@[a-z]+\\.+[a-z]+";

    if (val.isEmpty()) {
        email.setError(getString(R.string.empty_field));
        return false;
    } else if (!val.matches(emailPattern)) {
        email.setError(getString(R.string.invalid_email));
        return false;
    } else {
        email.setError(null);
        return true;
    }
}
```

```
private Boolean validatePassword() {
    String val = password.getText().toString();
    String passwordVal = "[a-zA-Z0-9!@#$%^&*]{4,}" +
        "(?=.*[0-9])" + //at least 1 digit
        "(?=.*[a-z])" + //at least 1 lower case letter
        "(?=.*[A-Z])" + //at least 1 upper case letter
        "(?=.*[a-zA-Z])" + //any letter
        "(?=.*[@#$%^&*])" + //at least 1 special character
        "(?=\\S+$)" + //no white spaces
        ".{4,}" + //at least 4 characters
        "$";

    if (val.isEmpty()) {
        password.setError(getString(R.string.empty_field));
        return false;
    } else if (!val.matches(passwordVal)) {
        password.setError(getString(R.string.weak_password));
        return false;
    } else {
        password.setError(null);
        return true;
    }
}

private Boolean validateAge() {
    String val = age.getText().toString();

    if (val.isEmpty()) {
        age.setError(getString(R.string.empty_field));
        return false;
    } else if (Integer.parseInt(val) <= 0 || Integer.parseInt(val) >= 100) {
        age.setError(getString(R.string.invalid_age));
        return false;
    } else {
        age.setError(null);
        return true;
    }
}
```

Εικόνες 47,48. Έλεγχοι συμπλήρωσης πεδίων εγγραφής χρήστη

6.6 Text to Speech

Η παρούσα εφαρμογή, στο κομμάτι της μετάφρασης υποστηρίζει την λειτουργία ανάγνωσης κειμένου (Text-To-Speech). Η λειτουργία αυτή παρέχεται από την κλάση *MyTTS.java*, όπου ορίζονται σαν προεπιλεγμένη γλώσσα τα Αγγλικά και μέσω της μεθόδου *speak()* γίνεται έλεγχος για την γλώσσα στην οποία είναι γραμμένο το κείμενο. Αν το κείμενο είναι γραμμένο σε μία από τις παρακάτω γλώσσες: Γαλλικά, Γερμανικά, Ιταλικά, Κινέζικα, Ιαπωνικά, τότε υποστηρίζεται η λειτουργία ανάγνωσης κειμένου με την αντίστοιχη προφορά.

```
public void speak(String message, String lan){
    switch(lan) {
        case "fr":
            tts.setLanguage(Locale.FRENCH);
            break;
        case "de":
            tts.setLanguage(Locale.GERMAN);
            break;
        case "it":
            tts.setLanguage(Locale.ITALIAN);
            break;
        case "zh":
            tts.setLanguage(Locale.CHINESE);
            break;
        case "ja":
            tts.setLanguage(Locale.JAPANESE);
            break;
        default:
            break;
    }
    tts.speak(message, TextToSpeech.QUEUE_ADD, params: null, utteranceId: null);
}
```

Εικόνα 49. Μέθοδος ανάγνωσης κειμένου speak()

6.7 Λειτουργία Μετάφρασης

Η λειτουργία αυτή παρέχεται από τα translation features του ML Kit για συσκευές Android. Το activity υπεύθυνο για αυτή την λειτουργία είναι το *Translation.java* μέσα στο οποίο εκτελείται η μετάφραση.

Αρχικά, ο χρήστης καλείται να επιλέξει τις γλώσσες ανάμεσα στις οποίες θα γίνει η μετάφραση. Αυτό πραγματοποιείται με το πάτημα του κουμπιού «Select» και καλείται η μέθοδος *language()* η οποία οδηγεί τον χρήστη στο επόμενο activity *SelectLanguage.java* με την χρήση ενός Intent που έχει ως παράμετρο το id του κουμπιού που πατήθηκε.

```
// Method called when the language buttons are pressed
public void language(View view) {
    // Start new activity to select language
    Intent intent = new Intent( packageContext: this, SelectLanguage.class);
    intent.putExtra( name: "btnPressed", view.getResources().getResourceEntryName(view.getId()));
    startActivityForResult(intent, REQUEST_CODE);
}
```

Εικόνα 50. Μέθοδος επιλογής γλώσσας language()

Έχει δημιουργηθεί μία κλάση *Language.java* ώστε να περιγράφει κάθε γλώσσα με τα εξής χαρακτηριστικά: όνομα (title), εικονίδιο (icon, και συγκεκριμένα την σημαία της χώρας), συντομογραφία ονόματος χώρας (short), και το tag της, όπως το αναγνωρίζει το ML Kit.

Το activity *SelectLanguage.java* αποτελείται από ένα recycler view, το οποίο κατά την έναρξη του activity διαβάζει την Realtime database που είναι συνδεδεμένη με το project και αρχικοποιώντας μία μεταβλητή τύπου *DatabaseReference*, ορίζει σαν σημείο αναφοράς στην βάση το κλαδί «*Languages*».

Έπειτα, διαβάζει διαδοχικά όλα τα στοιχεία που βρίσκονται σε εκείνο το σημείο της βάσης και δημιουργεί ένα αντικείμενο τύπου Language, κάθε φορά με τα χαρακτηριστικά της γλώσσας που έχει πάρει από την βάση, περνώντας τα στην συνέχεια σε μία λίστα τύπου Language. Έτσι, η λίστα αυτή περιέχει όλες τις διαθέσιμες

γλώσσες από και προς μετάφραση. Τέλος, η λίστα αυτή αποτελεί το περιεχόμενο ενός adapter (της κλάσης *MyAdapter.java*).

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
    setContentView(R.layout.activity_select_language);
    // Initialize the recyclerView and get the database reference
    recyclerView = findViewById(R.id.languageList);
    database = FirebaseDatabase.getInstance().getReference("Languages");
    recyclerView.setHasFixedSize(true);
    recyclerView.setLayoutManager(new LinearLayoutManager(this));

    matches = new ArrayList<>();
    list = new ArrayList<>(); // This list contains all the data from database

    // Set the adapter
    myAdapter = new MyAdapter(this, list);
    recyclerView.setAdapter(myAdapter);

    // Show auto detection language option only when choosing language to translate from
    String btn = getIntent().getStringExtra("btnPressed");
    if (btn.equals("LanFrom")){
        Language auto = new Language(title: "Auto", icon: "auto.png", aShort: "AUTO", tag: "auto");
        list.add(auto);
    }

    // Read data from database
    database.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            // For every language (child)
            for (DataSnapshot dataSnapshot : snapshot.getChildren()){
                // Create a Language object and set it's attributes, according to database values
                Language lan = dataSnapshot.getValue(Language.class);
                list.add(lan);
            }
            myAdapter.notifyDataSetChanged();
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {
            showMessage("Error", error.getMessage());
        }
    });
}
```



Εικόνες 51,52. Ανάγνωση διαθέσιμων γλωσσών από τη βάση και βάση (Languages)

Οι γλώσσες εμφανίζονται στον χρήστη μέσω ενός recycler view και κάθε μία από αυτές αποτελεί ένα κομμάτι Card view. Η αντιστοίχιση του card view με τις γλώσσες που περιέχονται στην λίστα του adapter γίνεται με την μέθοδο *onBindViewHolder()*. Επίσης, εντός της μεθόδου αυτής δημιουργείται ένα *StorageReference* ώστε να ανακτηθεί η εικόνα (σημαία) που αντιστοιχεί σε κάθε χώρα και είναι αποθηκευμένη εντός του *Storage* της *Firebase*. Η μέθοδος *getIcon()* διαβάζει το όνομα της εικόνας που είναι αποθηκευμένη στο *Storage*, το οποίο ταυτίζεται με την αντίστοιχη πληροφορία που υπάρχει στη βάση.

```
// Get language cover images from Firebase Storage
StorageReference imageRef2 = holder.storageReference.child(lan.getIcon());
imageRef2.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
    // Use Glide to get the image through url
    @Override
    public void onSuccess(Uri uri) { Glide.with(context).load(uri).into(holder.cover); }
}).addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) { e.getLocalizedMessage(); }
});
```

Εικόνα 53. Ανάγνωση εικόνας από το *Storage* και προβολή της

Επιπλέον, υπάρχει η δυνατότητα γρήγορης αναζήτησης της επιθυμητής από τον χρήστη γλώσσας με τη χρήση της μεθόδου *getFilter()* της κλάσης *MyAdapter.java*. Κύριος σκοπός αυτής της μεθόδου είναι η εύρεση γλωσσών των οποίων ο τίτλος ξεκινάει με την ονομασία της γλώσσας που αναζητεί ο χρήστης εκείνη τη στιγμή, και να επιστρέψει το αποτέλεσμα σε μία φιλτραρισμένη λίστα η οποία έπειτα θα προβληθεί στον χρήστη.

```
@Override
public Filter getFilter() { return listFilter; }

// Return the filter results
private Filter listFilter = new Filter() {
    @Override
    protected FilterResults performFiltering(CharSequence constraint) {
        ArrayList<Object> filteredList = new ArrayList<>();
        // constraint variable is the input of the search bar
        // If constraint is null, return all the results
        if (constraint == null || constraint.length() == 0) {
            filteredList.addAll(listFull);
        } else { // Create a string with the filter pattern (case insensitive)
            String filterPattern = constraint.toString().toLowerCase().trim();
            // Iterate all languages from list and find those that match the filter pattern
            for (Object item : listFull) {
                if (((Language) item).getTitle().toLowerCase().startsWith(filterPattern)){
                    filteredList.add(item); // add it to the filtered list
                }
            }
        }
        FilterResults results = new FilterResults();
        results.values = filteredList;
        // Returns the filtered list to the publishResults method
        return results;
    }
}
```

Εικόνα 54. Αναζήτηση γλώσσας

Ακόμα, η λειτουργία της μετάφρασης υποστηρίζει την αναγνώριση κειμένου από φωνητική εντολή, με την χρήση της μεθόδου *talk()* της κλάσης *Translation.java*. Για την χρήση αυτής της λειτουργίας απαιτείται αποδοχή της άδειας χρήσης του μικροφώνου από την εφαρμογή. Τέλος, ανάλογα με τη γλώσσα που έχει επιλέξει ο χρήστης, σε εκείνη θα πρέπει να εκφωνηθεί το κείμενο που επιθυμεί.

```
// Voice recognition method
public void talk(View view) {
    // Request for microphone permission
    if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.RECORD_AUDIO) != PackageManager.PERMISSION_GRANTED){
        ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.RECORD_AUDIO}, REQUEST_CODE_MICROPHONE);
    } else {
        Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, from); // Choose the language
        intent.putExtra(RecognizerIntent.EXTRA_PROMPT, value: "Say something");
        startActivityForResult(intent, REQUEST_CODE_MICROPHONE);
    }
}
```

Εικόνα 55. Λειτουργία Speech to Text

Επιπρόσθετα, υποστηρίζεται ο αυτόματος εντοπισμός της γλώσσας στην οποία είναι γραμμένο το κείμενο που πρόκειται να μεταφραστεί. Αυτό γίνεται με τη χρήση της μεθόδου *identify()* της κλάσης *Translation.java*, όπου χρησιμοποιείται ένας *Language Identifier* του *ML Kit*, ο οποίος αναγνωρίζει κείμενο σε περισσότερες από 100 διαφορετικές γλώσσες. Για τον προσδιορισμό της γλώσσας μιας συμβολοσειράς, καλείται η μέθοδος *LanguageIdentification.getClient()* για να αρχικοποιηθεί ένα αντικείμενο τύπου *LanguageIdentifier* και, στη συνέχεια, η συμβολοσειρά δίνεται σαν όρισμα στη μέθοδο *identifyLanguage()* του *LanguageIdentifier*. Αν η αναγνώριση της γλώσσας έγινε επιτυχώς, τότε ο κωδικός της γλώσσας αυτής αποτελεί την γλώσσα από την οποία θα μεταφραστεί το κείμενο και καλείται η *makeTranslation()* προκειμένου να γίνει η μετάφραση.

```
// Method for language identification
public void identify() {
    LanguageIdentifier languageIdentifier = LanguageIdentification.getClient();
    // Text to be identified
    languageIdentifier.identifyLanguage(raw.getText().toString())
        .addOnSuccessListener(new OnSuccessListener<String>() {
            @Override
            public void onSuccess(@NonNull String languageCode) {
                if (languageCode.equals("und")) { // If language can't be identified
                    Toast.makeText(getApplicationContext(), "Can't identify language.", Toast.LENGTH_LONG).show();
                } else {
                    Toast.makeText(getApplicationContext(), "Language: " + languageCode, Toast.LENGTH_LONG).show();
                    // Set the language to be translated from
                    from = languageCode;
                    makeTranslation(); // Translate
                    from = "auto";
                }
            }
        })
        .addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                // Model could not be loaded or other internal error
                Toast.makeText(getApplicationContext(), "Failed to identify language", Toast.LENGTH_LONG).show();
            }
        });
}
```

Εικόνα 56. Λειτουργία *Language Identification*

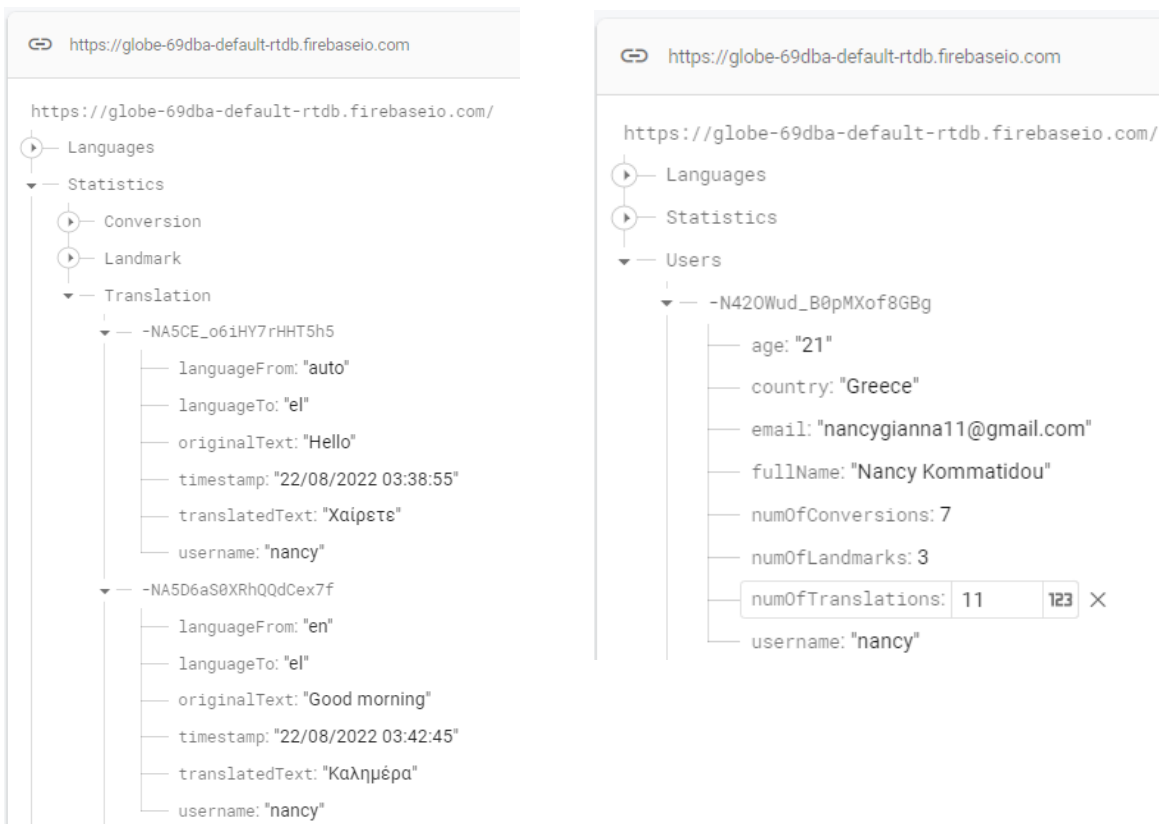
Η *makeTranslation()* εκτελεί την μετάφραση με τη δημιουργία ενός αντικειμένου τύπου *Translator*, στο οποίο ορίζονται οι γλώσσες που συμμετέχουν στην μετάφραση (*SourceLanguage* και *TargetLanguage*). Έπειτα, απαιτείται η λήψη των απαιτούμενων *translation models*, αν αυτά δεν υπάρχουν ήδη κατεβασμένα στην συσκευή. Αν η λήψη είναι επιτυχής, καλείται η μέθοδος *translate()* στον *translator* που έχει δημιουργηθεί με όρισμα το κείμενο προς μετάφραση, αλλιώς εμφανίζεται το κατάλληλο μήνυμα.

Αν και η μετάφραση είναι επιτυχής, τότε εμφανίζεται στο κάτω πλαίσιο κειμένου το μεταφρασμένο κείμενο και με την χρήση ενός *listener* γράφουμε σημαντικά στοιχεία που αφορούν την μετάφραση στην βάση.

Συγκεκριμένα, στο κλαδί «*Statistics*» αποθηκεύονται:

1. Η γλώσσα που είναι γραμμένο το κείμενο προς μετάφραση
2. Η γλώσσα στην οποία θα μεταφραστεί το κείμενο
3. Το αρχικό κείμενο προς μετάφραση
4. Το μεταφρασμένο κείμενο
5. Το όνομα χρήστη του τρέχοντος συνδεδεμένου χρήστη που κάνει την μετάφραση
6. Ένα *timestamp* με την ημερομηνία και την ώρα που έγινε η μετάφραση

Επίσης, κάθε χρήστης στη βάση έχει ένα πεδίο με όνομα *numOfTranslations*, το οποίο αντιστοιχεί στο πλήθος των μεταφράσεων που έχει κάνει. Με κάθε μετάφραση, ο αριθμός αυτός αυξάνεται κατά μία μονάδα.



Εικόνα 57. Αποθήκευση στατιστικών στοιχείων μετάφρασης στη βάση

```
TranslatorOptions options =  
    new TranslatorOptions.Builder()  
        // Language identification  
        .setSourceLanguage(TranslateLanguage.fromLanguageTag(from))  
        .setTargetLanguage(TranslateLanguage.fromLanguageTag(to))  
        .build();  
final Translator translator = com.google.mlkit.nl.translate.Translation.getClient(options);
```

Εικόνα 58. Διαμόρφωση αντικειμένου Translator

Τέλος, προσφέρεται η δυνατότητα εξαγωγής κειμένου προς μετάφραση μέσα από μία εικόνα. Αυτό γίνεται με την χρήση της βιβλιοθήκης `com.google.android.gms:play-services-mlkit-text-recognition` του ML Kit η οποία προσφέρει αναγνώριση λατινογενούς κειμένου μέσα από εικόνα ή βίντεο.

Για την υλοποίηση της λειτουργίας αυτής, αρχικά ορίζουμε ένα αντικείμενο τύπου `TextRecognizer` και μέσω της μεθόδου `process()` του δίνουμε ως όρισμα μία εικόνα να επεξεργαστεί και να εντοπίσει σε μορφή blocks. Κάθε block περιέχει ένα τμήμα του κειμένου που αναγνωρίστηκε, και μέσω της μεθόδου `processTextRecognitionResult()` της κλάσης `SelectImage.java`, ενώνουμε κάθε μπλοκ σε ένα μεγαλύτερο κομμάτι κειμένου το οποίο τελικά προορίζεται για μετάφραση με το πάτημα του κουμπιού «Results».

Αξίζει να σημειωθεί, ότι η εικόνα μπορεί να προέρχεται είτε από τα αποθηκευμένα αρχεία της συσκευής του χρήστη, είτε να γίνει η λήψη της εκείνη τη στιγμή με την χρήση της κάμερας. Και οι δύο αυτές ενέργειες προϋποθέτουν την αποδοχή της αντίστοιχης άδειας από τον χρήστη.

```
public void runTextRecognition(View view) throws IOException {
    InputImage image;
    if (from_camera){
        image = InputImage.fromBitmap(photo, rotationDegrees: 0);
    } else {
        image = InputImage.fromFilePath(context: this, selectedImageUri);
    }
    TextRecognizer recognizer = TextRecognition.getClient(TextRecognizerOptions.DEFAULT_OPTIONS);
    recognizer.process(image)
        .addOnSuccessListener(
            new OnSuccessListener<Text>() {
                @Override
                public void onSuccess(Text texts) {
                    processTextRecognitionResult(texts);
                }
            })
        .addOnFailureListener(
            new OnFailureListener() {...});
}
```

Εικόνα 59. Αναγνώριση κειμένου μέσω ενός TextRecognizer

```
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == REQUEST_CODE_STORAGE && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
        if (ActivityCompat.checkSelfPermission(context: this, Manifest.permission.READ_EXTERNAL_STORAGE) != PackageManager.PERMISSION_GRANTED) {
            return;
        }
        imageChooser();
    }
    if (requestCode == REQUEST_CODE_CAMERA && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
        if (ActivityCompat.checkSelfPermission(context: this, Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
            return;
        }
        // Create the camera_intent ACTION_IMAGE_CAPTURE
        // it will open the camera for capture the image
        Intent camera_intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        camera_intent.putExtra(MediaStore.EXTRA_OUTPUT, photo);
        // Start the activity with camera_intent,
        // and request pic id
        startActivityForResult(camera_intent, pic_id);
    }
}

// this function is triggered when
// the Select Image Button is clicked
void imageChooser() {
    // create an instance of the
    // intent of the type image
    Intent i = new Intent();
    i.setType("image/*");
    i.setAction(Intent.ACTION_GET_CONTENT);

    // pass the constant to compare it
    // with the returned requestCode
    startActivityForResult(Intent.createChooser(i, title: "Select Picture"), gallery_pic_id);
}
```

Εικόνα 60. Εισαγωγή ή λήψη εικόνας

6.8 Προφίλ χρήστη

Μέσω του activity *UserProfile.java* δίνεται στον χρήστη η δυνατότητα να εξερευνήσει το προφίλ του. Η κλάση αυτή είναι υπεύθυνη για την ανάγνωση πληροφοριών από την Realtime database και συγκεκριμένα από το κλαδί «Users», και την εμφάνιση αυτών στον χρήστη.

Αρχικά, κατά την φόρτωση του activity εμφανίζεται ένα LoadingDialog για τρία δευτερόλεπτα έως ότου διαβαστούν από την βάση οι κατάλληλες πληροφορίες. Έπειτα, εμφανίζονται με την σειρά: το όνομα χρήστη, ο αριθμός των μεταφράσεων, της αναγνώρισης αξιοθέατων και της μετατροπής νομισμάτων που έχουν γίνει, το πλήρες όνομα, η διεύθυνση ηλεκτρονικού ταχυδρομείου, η χώρα διαμονής και η ηλικία του χρήστη. Όλα αυτά είναι στοιχεία που ο χρήστης έχει δώσει κατά την εγγραφή του στην εφαρμογή.

```
// Create and show a custom loading dialog
LoadingDialog loadingDialog = new LoadingDialog( myActivity: UserProfile.this);
loadingDialog.startLoadingDialog();
// After 3 seconds, dismiss the loading dialog
Handler handler = new Handler();
handler.postDelayed(new Runnable() {
    @Override
    public void run() { loadingDialog.dismissDialog(); }
}, delayMillis: 3000);



// Load user's info from database
database.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        // For every user (child)
        for (DataSnapshot dataSnapshot : snapshot.getChildren()){
            User user = dataSnapshot.getValue(User.class);
            // For currently connected user
            if (user.getEmail().equals(FirebaseAuth.getInstance().getCurrentUser().getEmail())){
                translations.setText(String.valueOf(user.getNumOfTranslations()));
                landmarks.setText(String.valueOf(user.getNumOfLandmarks()));
                conversions.setText(String.valueOf(user.getNumOfConversions()));
                username.setText(user.getUsername());
                fullName.getEditText().setText(user.getFullName());
                email.getEditText().setText(user.getEmail());
                country.getEditText().setText(user.getCountry());
                age.getEditText().setText(user.getAge());
                break;
            }
        }
    }
}
```

Εικόνα 61. Προφίλ χρήστη

6.9 Μορφή Βάσης

Παρακάτω παρουσιάζονται κάποια ενδεικτικά screenshots για την απεικόνιση του σχήματος της βάσης

6.9.1 Firebase Authentication

Identifier	Providers	Created ↑	Signed In	User UID
nancygianna11@gmail.com		Jun 8, 2022	Aug 24, 2022	7iOkeQNjZ4PcRqOSqly7GsYzCPu2
anastasia.mexa@yahoo.gr		Jun 8, 2022	Sep 4, 2022	8aRCE7RcPvRW6qek59eo8sQJyO...

Εικόνα 62. Firebase Authentication

6.9.2 Firebase Realtime Database

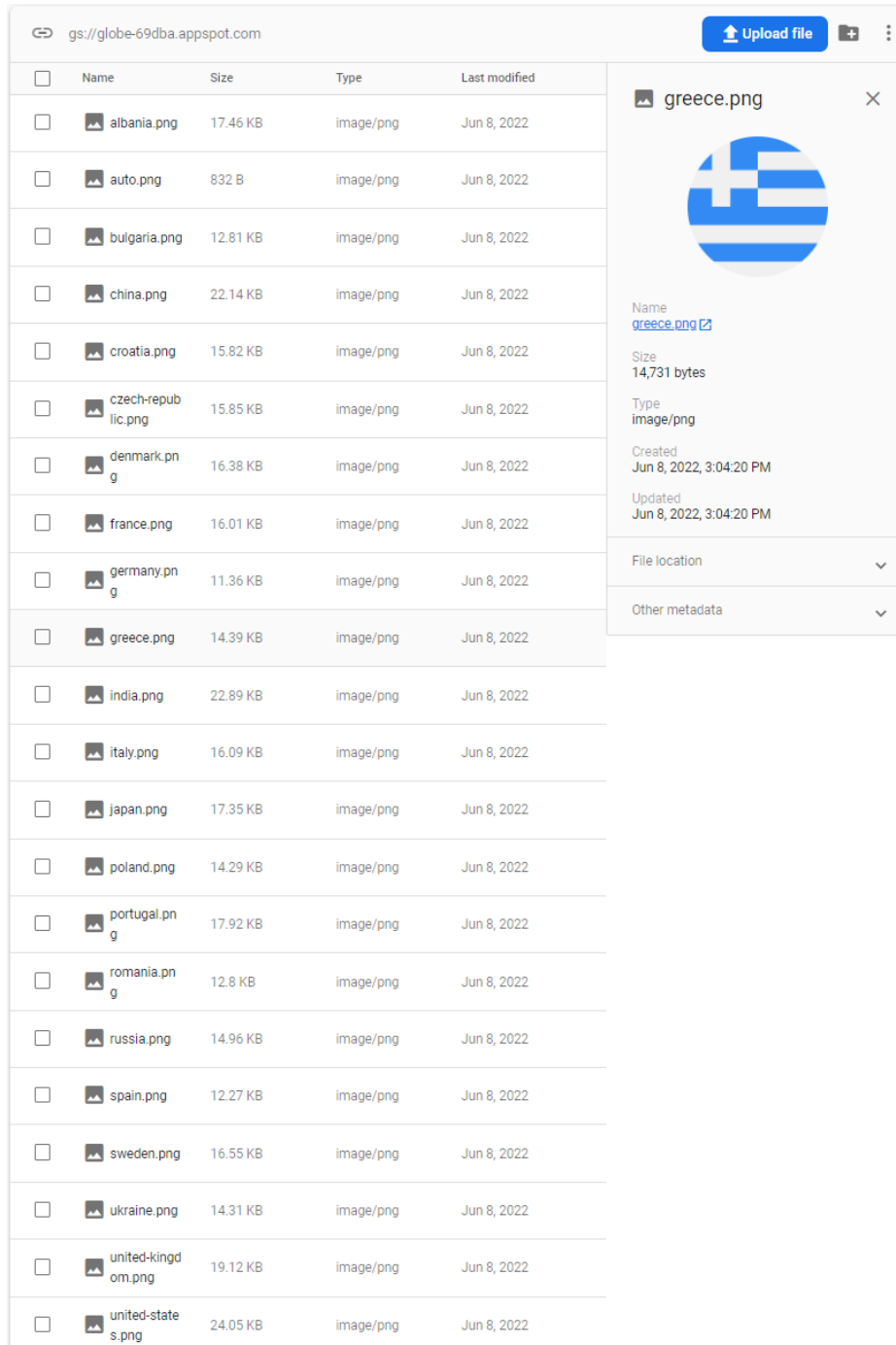
Εδώ αποθηκεύονται οι γλώσσες που υποστηρίζει η εφαρμογή, τα στατιστικά των χρηστών για το πλήθος των μεταφράσεων, της αναγνώρισης αξιοθέατων και της μετατροπής συναλλάγματος, καθώς και τα στοιχεία κάθε εγγεγραμμένου χρήστη.



Εικόνα 63. Realtime Database

6.9.3 Firebase Storage

Εδώ αποθηκεύονται οι εικόνες που χρησιμοποιούνται στην εφαρμογή της μετάφρασης, και συγκεκριμένα οι σημαίες κάθε χώρας της οποίας η γλώσσα είναι διαθέσιμη για μετάφραση.



Εικόνα 64. Storage

7 Συμπεράσματα και μελλοντικές επεκτάσεις

Η εφαρμογή «Globe» αποτελεί ένα σημαντικό τουριστικό εργαλείο καθώς προσφέρει άμεσα και γρήγορα λύση σε βασικές ανάγκες που προκύπτουν σε ένα ταξίδι σε μία ξένη χώρα. Με την χρήση της εφαρμογής αυτής, ο χρήστης κάνει πιο εύκολη την επικοινωνία του αλλά και την κατανόηση του περιβάλλοντός του, εισάγεται σε μία διαδικασία αναζήτησης περισσότερων πληροφοριών σχετικά με ένα αξιοθέατο, που πολλές φορές συνεπάγεται με την απόκτηση γνώσεων όσον αφορά την ιστορία της χώρας που έχει επισκεφτεί, αλλά και διευκολύνει τις χρηματικές συναλλαγές του. Τόσο η ανάπτυξη, όσο και η προώθηση του τουρισμού έχει καίριο ρόλο στην παγκόσμια οικονομία. Επομένως, η χρήση μίας τέτοιας εφαρμογής μπορεί να συμβάλει στην άνθιση του τουριστικού τομέα.

Στο παρόν έγγραφο, γίνεται αναλυτική παρουσίαση της χρήσης της εφαρμογής, καθώς και λεπτομερής περιγραφή των τεχνικών χαρακτηριστικών της, εστιάζοντας σε κάθε λειτουργία της και σε κάθε εργαλείο που χρησιμοποιήθηκε για την υλοποίησή της. Επίσης, γίνεται αναφορά της θέσης της εφαρμογής στον παγκόσμιο ιστό, έπειτα από μελέτη σχετικά με άλλες συναφείς διαθέσιμες εφαρμογές.

Η εφαρμογή αυτή αποτελεί μία πολύ καλή βάση, η οποία μπορεί να εμπλουτιστεί με επιπρόσθετες λειτουργίες και δυνατότητες. Μερικές ενδεικτικές βελτιώσεις και προσθήκες είναι:

- Η ύπαρξη προτεινόμενων λέξεων και φράσεων, με σκοπό την εύκολη και γρήγορη μετάφρασή τους. Παραδείγματα τέτοιων λέξεων και φράσεων είναι: «Καλημέρα», «Πόσο κοστίζει;», «Τί ώρα είναι;», «Θα ήθελα να κάνω μία κράτηση», κλπ.
- Η αποθήκευση αγαπημένων μετατροπών συναλλάγματος, για την πιο άμεση πραγματοποίηση συχνών μετατροπών.
- Η αξιολόγηση ενός αξιοθέατου που έχει επισκεφτεί ο χρήστης, καθώς και η προβολή προηγούμενων κριτικών άλλων χρηστών.

Συμπεραίνοντας, η εφαρμογή απευθύνεται σε ταξιδιώτες ανεξαρτήτου ηλικίας και εθνικότητας, παρέχοντάς τους βασικές δυνατότητες, όταν αυτοί έχουν επισκεφτεί μία ξένη χώρα. Φυσικά και είναι δυνατή η ενσωμάτωση νέων λειτουργιών, οι οποίες αποσκοπούν στην μελλοντική επέκταση της λειτουργικότητας της εφαρμογής. Ωστόσο, η παρούσα υλοποίηση αποτελεί ένα ολοκληρωμένο εργαλείο υποβοήθησης τουριστών.

8 Παραρτήματα

8.1 Διάγραμμα κλάσεων σε UML



Εικόνα 65. Class Diagram

8.2 Ανάπτυξη ιστοσελίδας για την εμφάνιση στατιστικών

Για την καλύτερη εμφάνιση των κυριότερων στατιστικών δεδομένων της εφαρμογής, δημιουργήθηκε μία ιστοσελίδα η οποία εμφανίζει σε μορφή πινάκων τα στοιχεία που έχουν αποθηκευτεί στην βάση για κάθε ενέργεια του χρήστη. Συγκεκριμένα, από το κλαδί «*Statistics*» εμφανίζονται:

- Για την λειτουργία της μετάφρασης:
 1. Η γλώσσα που είναι γραμμένο το κείμενο προς μετάφραση
 2. Η γλώσσα στην οποία θα μεταφραστεί το κείμενο
 3. Το αρχικό κείμενο προς μετάφραση
 4. Το μεταφρασμένο κείμενο
 5. Το όνομα χρήστη του τρέχοντος συνδεδεμένου χρήστη που κάνει την μετάφραση
 6. Ένα timestamp με την ημερομηνία και την ώρα που έγινε η μετάφραση
- Για την λειτουργία της αναγνώρισης αξιοθέατων:
 1. Οι συντεταγμένες του αξιοθέατου που αναγνωρίστηκε
 2. Το όνομα αυτού
 3. Το όνομα χρήστη του τρέχοντος συνδεδεμένου χρήστη
 4. Ένα timestamp με την ημερομηνία και την ώρα που έγινε η αναγνώριση
- Για την λειτουργία της μετατροπής συναλλάγματος:
 1. Ο τύπος του νομίσματος που επιθυμεί ο χρήστης να τροποποιήσει
 2. Ο επιθυμητός τύπος συναλλάγματος
 3. Το αρχικό ποσό προς μετατροπή
 4. Το τελικό τροποποιημένο ποσό
 5. Το όνομα χρήστη του τρέχοντος συνδεδεμένου χρήστη που κάνει την μετατροπή
 6. Ένα timestamp με την ημερομηνία και την ώρα που έγινε η μετατροπή

Η επικοινωνία της βάσης (Firebase) με την σελίδα γίνεται μέσω JavaScript όπως φαίνεται παρακάτω.

```
//import { initializeApp } from "firebase/app";
import { initializeApp } from "https://www.gstatic.com/firebasejs/9.4.0/firebase-app.js";
import { getDatabase, ref, child, onValue, get } from "https://www.gstatic.com/firebasejs/9.4.0/firebase-database.js";

// TODO: Replace the following with your app's Firebase project configuration
// See: https://firebase.google.com/docs/web/learn-more#config-object
const firebaseConfig = {
  // ...
  // The value of `databaseURL` depends on the location of the database
  databaseURL: "https://globe-69dba-default-rtdb.firebaseio.com/",
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);

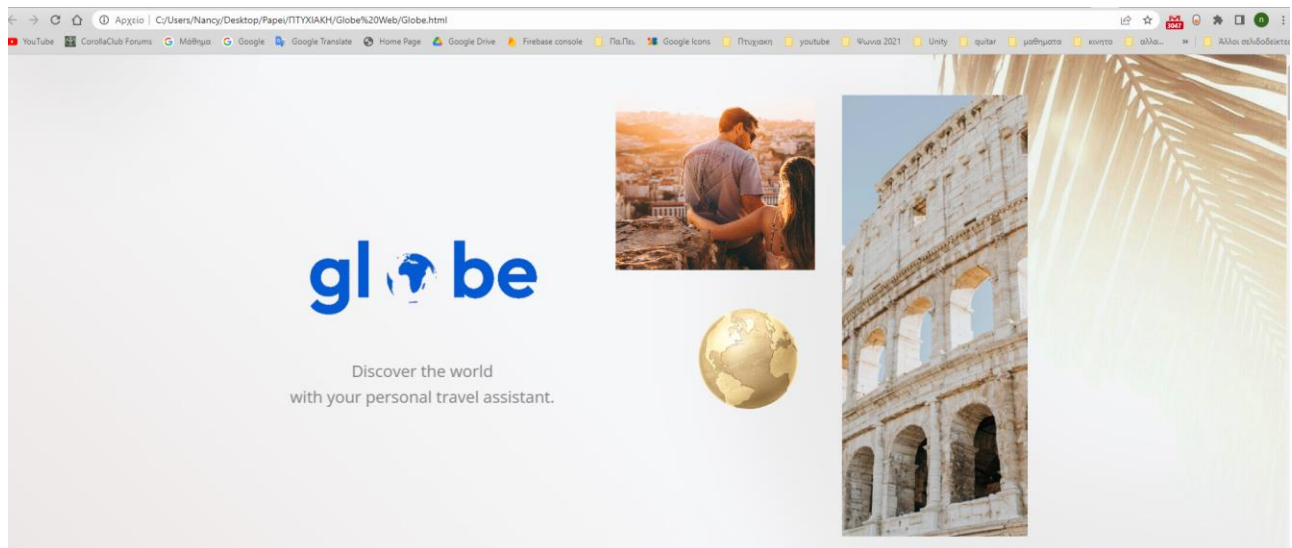
// Initialize Realtime Database and get a reference to the service
const database = getDatabase(app);
```

Εικόνα 66. JavaScript – Firebase connection

```
//----- GETTING ALL THE DATA -----  
  
function GetAllDataRealtime() {  
  const dbRef1 = ref(database, "Statistics/Conversion");  
  
  onValue(dbRef1, (snapshot) => {  
    var conversions = [];  
  
    snapshot.forEach(childSnapshot => {  
      conversions.push(childSnapshot.val());  
    });  
  
    AddAllItemsToTableConversions(conversions);  
  })  
  window.onload = GetAllDataRealtime;  
}
```

Εικόνα 67. Realtime ανάγνωση των δεδομένων της βάσης

Παρακάτω ακολουθούν ενδεικτικά screenshot από την ιστοσελίδα.



Landmark Statistics

For every landmark detection we collect specific data, such as the coordinates of the landmark, current user's username and a timestamp.



Landmark Detections

Username	Landmark	Latitude	Longitude	Timestamp
nancy	Acropolis of Athens	37.9715323	23.7257492	22/08/2022 07:46:36
nancy	Big Ben	51.500782	-0.12462600000000001	22/08/2022 07:50:03
nancy	Acropolis of Athens	37.9715323	23.7257492	24/08/2022 08:19:52
andreas	Statue of Liberty National Monument	40.6892494	-74.04450039999999	24/08/2022 08:55:11
stacy	Liberty Island	40.689824	-74.045047	26/08/2022 12:56:42
stacy	Acropolis of Athens	37.9715323	23.7257492	26/08/2022 12:57:36
stacy	Acropolis of Athens	37.9715323	23.7257492	04/09/2022 07:39:17

Currency Conversion Statistics

For every currency conversion we collect specific data, such as original and converted amount of currency, the names of the currencies, current user's username and a timestamp.



Conversions

Username	Currency From	Currency To	Original Value	Converted Value	Timestamp
nancy	EUR (Euro Member Countries, €)	ALL (Albania Lek, Lek)	1	117.12	22/08/2022 03:56:06
nancy	EUR (Euro Member Countries, €)	AED (United Arab Emirates Dirham, د.إ)	1	3.65	22/08/2022 04:00:45
nancy	EUR (Euro Member Countries, €)	RON (Romania Leu, lei)	1	4.88	22/08/2022 04:08:38
nancy	EUR (Euro Member Countries, €)	USD (United States Dollar, \$)	1.0	0.99	22/08/2022 04:11:22
nancy	EUR (Euro Member Countries, €)	RUB (Russia Ruble, Р)	1.0	59.79	22/08/2022 07:27:00



Translations

Username	Language From	Language To	Original Text	Translated Text	Timestamp
nancy	auto	el	Hello	Χαίρετε	22/08/2022 03:38:55
nancy	en	el	Good morning	Καλημέρα	22/08/2022 03:42:45
stacy	auto	el	Hello	Γεια σας	24/08/2022 04:56:27
nancy	el	ja	મામા	મામા	24/08/2022 08:13:25



9 Ηλεκτρονική Βιβλιογραφία

1. <https://firebase.google.com/>
2. <https://firebase.google.com/products/realtime-database>
3. <https://firebase.google.com/products/storage>
4. <https://firebase.google.com/products/auth>
5. <https://firebase.google.com/docs/database/android/read-and-write>
6. <https://developers.google.com/ml-kit>
7. <https://developers.google.com/ml-kit/language/translation/android>
8. <https://developers.google.com/ml-kit/language/identification/android>
9. <https://developers.google.com/ml-kit/vision/text-recognition/android>
10. <https://developers.google.com/ml-kit/vision/text-recognition/v2/android>
11. <https://developer.android.com/reference/android/speech/tts/TextToSpeech>
12. <https://www.geeksforgeeks.org/how-to-convert-speech-to-text-in-android/>
13. <https://exchangeratesapi.io/>
14. https://apilayer.com/marketplace/exchangerates_data-api
15. <https://serpapi.com/knowledge-graph>
16. <https://firebase.google.com/docs/ml/recognize-landmarks>
17. <https://firebase.google.com/docs/ml/android/recognize-landmarks>