

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
ΕΦΑΡΜΟΓΗ ANDROID (Globe)



ΑΝΑΣΤΑΣΙΑ ΙΩΑΝΝΑ ΜΕΞΑ
Π18101

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 |
| 6 | Αρχιτεκτονική συστήματος | 16 |
| 6.1 | Android Studio | 16 |
| 6.2 | Firebase..... | 16 |
| 6.3 | Permissions και dependencies..... | 20 |
| 6.4 | Υποστήριξη ελληνικής γλώσσας | 21 |
| 6.5 | Προφίλ χρήστη..... | 22 |
| 6.6 | Λειτουργία Authentication | 24 |
| 6.7 | Λειτουργία Currency Converter..... | 26 |
| 6.8 | Λειτουργία Landmark Detection..... | 29 |
| 7 | Συμπεράσματα και μελλοντικές επεκτάσεις | 34 |
| 8 | Παραρτήματα..... | 35 |
| 8.1 | Διάγραμμα κλάσεων σε UML | 35 |
| 8.2 | Ανάπτυξη ιστοσελίδας για την εμφάνιση στατιστικών | 36 |
| 9 | Βιβλιογραφικές Πηγές..... | 40 |



1 Περίληψη

Η παρούσα πτυχιακή εργασία αποσκοπεί την δημιουργία μίας εφαρμογής που απευθύνεται σε χρήστες κινητών Android, με στόχο την υποβοήθηση των τουριστών σε μια ξένη χώρα. Παρακάτω παρουσιάζονται αναλυτικά ο σκοπός, η χρήση και ο τρόπος λειτουργίας της εφαρμογής, δίνοντας ιδιαίτερη έμφαση στην επεξήγηση των τεχνολογιών που έχουν χρησιμοποιηθεί (Text Recognition, Language Identification, Translation features του ML KIT, Firebase, JSON requests). Επιπροσθέτως, διατίθεται το εγχειρίδιο χρήστη της εφαρμογής, καθώς και σκέψεις για μελλοντικές επεκτάσεις.

2 Abstract

This thesis aims to create an Android application for mobile users, with the purpose of assisting tourists in a foreign country. Below are presented in detail its use and functionality, emphasizing on the explanation of technologies that have been used (Text Recognition, Language Identification, Translation features of ML KIT, Firebase, JSON requests). In addition, its user manual is available, as well as considerations for future extensions.



3 Εισαγωγή

Ο σκοπός της εφαρμογής «Globe» είναι να βελτιώσει την ταξιδιωτική εμπειρία των χρηστών της, μέσω της παροχής βιοηθητικών λειτουργιών. Η μετάφραση κειμένου, η αντιστοίχιση συναλλάγματος και η αναγνώριση αξιοθέατων με την χρήση της κάμερας του κινητού, αποτελούν τις βασικές λειτουργίες που παρέχει η εφαρμογή στους χρήστες.

Η λειτουργία μετάφρασης υποστηρίζει μεταξύ άλλων αυτόματη αναγνώριση της γλώσσας, δυνατότητα ανάγνωσης κειμένου (text to speech) και μετατροπής ομιλίας σε κείμενο (speech to text), αναγνώριση και μετάφραση κειμένου από εικόνα ή από φωτογραφία, με την χρήση της κάμερας του κινητού (text recognition). Επομένως, ο χρήστης έχει στην διάθεσή του πολλαπλά εργαλεία για να μεταφράσει ένα κείμενο με τον πιο εύκολο τρόπο. Μερικά παραδείγματα χρήσης είναι η μετάφραση κειμένου από πινακίδες ή μενού εστιατορίων, αλλά και η καταγραφή ομιλίας και μετατροπής της σε κείμενο, ώστε να μην χρειαστεί ο χρήστης να πληκτρολογεί.

Η λειτουργία αντιστοίχισης συναλλάγματος διευκολύνει τον χρήστη στις μετατροπές, προσφέροντάς του πληροφορία για την πραγματική συναλλαγματική ισοτιμία σε 20 νομίσματα. Κατ' αυτόν τον τρόπο, μπορεί να μετατρέπει την συναλλαγματική αξία που επιθυμεί από και προς όποιο νόμισμα τον εξυπηρετεί και να ενημερωθεί κατάλληλα.

Η λειτουργία για την αναγνώριση αξιοθέατων με την χρήση της κάμερας του κινητού, προσφέρει στον χρήστη άμεσα και πανεύκολα με απλά ένα κλικ, πληροφορίες σχετικά με το αξιοθέατο που έχει «σκανάρει», στοιχεία επικοινωνίας (τηλέφωνο), την τοποθεσία του αξιοθέατου στο Google Maps, αλλά μπορεί να ανακατευθυνθεί και να επισκεφτεί την ιστοσελίδα του.

Ανταγωνιστικό στοιχείο της εφαρμογής, αφορά το γεγονός ότι όλες οι προαναφερθείσες λειτουργίες υπάρχουν μέσα στην ίδια εφαρμογή. Επομένως, ο χρήστης δεν χρειάζεται να περιηγείται στο διαδίκτυο σε αναζήτηση κάθε φορά της κατάλληλης εφαρμογής που θα εξυπηρετήσει τον σκοπό του.

Φυσικά, για να χρησιμοποιήσει κάποιος την εφαρμογή πρέπει πρώτα να πραγματοποιήσει την εγγραφή του και την σύνδεση στον λογαριασμό του. Η διαδικασία αυτή μπορεί να επιτευχθεί μέσα από την ίδια εφαρμογή συμπληρώνοντας τα στοιχεία που του ζητούνται. Μετά την επιτυχή αυθεντικοποίησή του, ο χρήστης μπορεί να δει τις πληροφορίες του και μερικά στατιστικά στοιχεία που τον αφορούν, μέσω του προφίλ του στην εφαρμογή.

Τέλος να σημειωθεί πως για τις ανάγκες της εργασίας, έχει δημιουργηθεί και μια ιστοσελίδα για την επίδειξη των πληροφοριών που καταγράφονται από την εφαρμογή.



4 Ανασκόπηση πεδίου

Πριν την έναρξη ανάπτυξης της εφαρμογής πραγματοποιήθηκε μια έρευνα εντοπισμού αντίστοιχων εφαρμογών. Μερικές από τις σχετικές εφαρμογές που βρέθηκαν, αναλύονται παρακάτω.

- [Google Translate](#)

Η παρούσα εφαρμογή αποτελεί μία δωρεάν υπηρεσία που παρέχεται από την Google. Μεταφράζει άμεσα λέξεις, προτάσεις, παραγράφους, ακόμα και ιστοσελίδες. Επιπλέον, είναι δυνατή η μετάφραση κειμένου από φωτογραφία ή ομιλία.

- [Landmark Recognition](#)

Η εφαρμογή αυτή μπορεί να εμφανίσει σχετικές πληροφορίες με βάση κάποιο αξιοθέατο που έχει ανιχνευτεί. Η φωτογραφία του αξιοθέατου μπορεί να τραβηγχτεί εκείνη την στιγμή, μέσω της κάμερας ή να επιλεχθεί μια από την συλλογή του κινητού.

- [All Currency Converter](#)

Αποτελεί έναν μετατροπέα νομισμάτων σε πραγματικό χρόνο με λειτουργία αριθμομηχανής. Παρέχει πληροφορίες συναλλαγματικών ισοτιμιών για όλα τα νομίσματα στον κόσμο, μαζί με διεθνείς τιμές χρυσού και αργύρου.

5 Παρουσίαση και χρήση εφαρμογής

Σε αυτή την ενότητα θα παρουσιαστούν όλα τα activities της εφαρμογής με τα οποία αλληλεπιδρά ο χρήστης. Επιπλέον, θα αναλυθεί η λειτουργία και η χρήση της εφαρμογής μέσω του εγχειρίδιου χρήστη (user manual) που επακολουθεί.

Main Activity:

Κατά την εκκίνηση της εφαρμογής, εμφανίζεται το animated λογότυπο μαζί με ένα σλόγκαν, που προτρέπει τον χρήστη να χρησιμοποιήσει την εφαρμογή. Το παρόν activity έχει τον ρόλο μιας loading οθόνης για την αρχικοποίηση της εφαρμογής. Ο χρήστης μετά από πέντε δευτερόλεπτα, θα μεταφερθεί αυτόματα στο επόμενο activity, με σκοπό να συνδεθεί στον λογαριασμό του.





Authentication:

Μετά το πέρας της αρχικής οθόνης, ο χρήστης ανακατευθύνεται στο Authentication activity με σκοπό να συνδεθεί στον λογαριασμό του. Υπάρχουν δύο πεδία στα οποία μπορεί να εισάγει το email του και τον κωδικό πρόσβασής του αντίστοιχα. Ύστερα και εφόσον έχει συμπληρώσει σωστά τα στοιχεία του, μπορεί να πατήσει το κουμπί “Go” για να συνδεθεί και να συνεχίσει την περιήγησή του στην εφαρμογή. Αν ο χρήστης δεν έχει λογαριασμό, μπορεί να δημιουργήσει έναν πατώντας πάνω στην επιλογή “New user? Sign up”. Έπειτα, θα μπορεί να συνδεθεί στην εφαρμογή με τα στοιχεία που εισήγαγε κατά την εγγραφή του.

- Email: Πρέπει να εισαχθεί ένας σωστός τύπος διεύθυνσης ηλεκτρονικού ταχυδρομείου.
- Password: Πρέπει ο κωδικός να είναι ο ίδιος με αυτόν που συμπλήρωσε κατά την εγγραφή του.

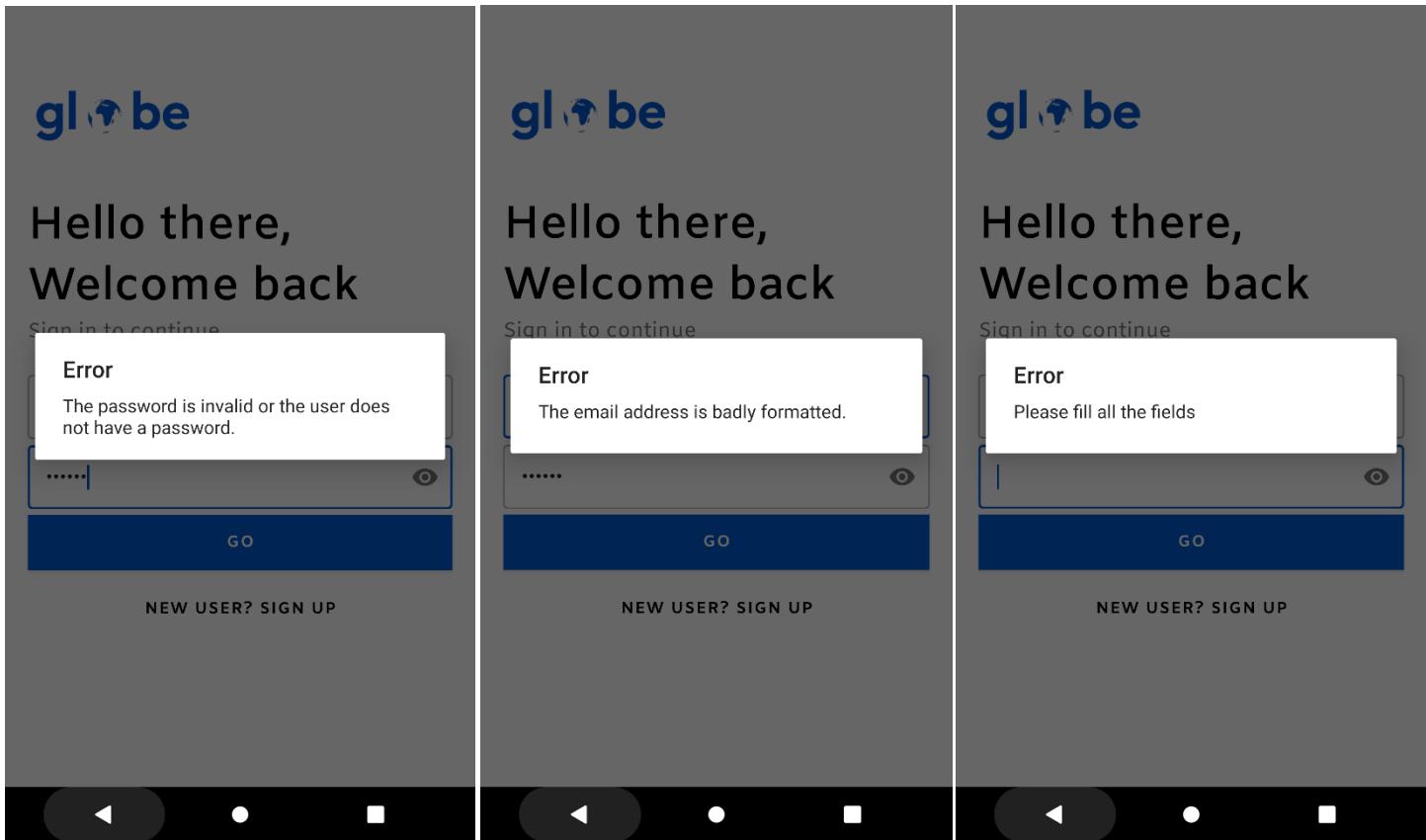
Και τα δύο πεδία είναι υποχρεωτικά.

Να σημειωθεί ότι στο πεδίο του κωδικού υπάρχει το κουμπί , με το οποίο ο χρήστης μπορεί να αποκρύψει και να κρύψει αντίστοιχα τους χαρακτήρες του κωδικού που έχει πληκτρολογήσει.

The figure consists of three side-by-side screenshots of a mobile application interface. All three screenshots feature a blue header bar with the word "globe" in white. Below the header, each screen displays the text "Hello there, Welcome back" in bold black font, followed by a smaller "Sign in to continue".

- Screenshot 1:** Shows two input fields: "Email" containing "Email" and "Password" containing "Password". Below the fields is a large blue button labeled "GO". At the bottom of the screen, there is a "NEW USER? SIGN UP" link and a navigation bar with three icons: a left arrow, a dot, and a right arrow.
- Screenshot 2:** Shows the same layout as the first screenshot, but the "Email" field now contains "test@test.gr" and the "Password" field contains "....". The "GO" button is blue and outlined in white.
- Screenshot 3:** Shows the same layout as the first two screenshots, but the "Email" field now contains "test@test.gr" and the "Password" field contains "test". The "GO" button is blue and outlined in white. Additionally, there is a small eye icon to the right of the password field.

Υπάρχουν έλεγχοι για το αν ο χρήστης εισάγει λάθος στοιχεία, δεν συμπληρώσει κανένα ή ένα από τα υποχρεωτικά πεδία, καθώς και αν το email που συμπλήρωσε είναι σε λάθος format. Ακολουθούν screenshots με τα αντίστοιχα μηνύματα λάθους.



Sign Up:

Στην περίπτωση που ο χρήστης θέλει να δημιουργήσει τον λογαριασμό του, τότε μεταφέρεται στο Sign Up activity στο οποίο του ζητούνται να εισάγει τα στοιχεία του. Συγκεκριμένα πρέπει να εισάγει το ονοματεπώνυμό του, το όνομα χρήστη της επιλογής του, την διεύθυνση του ηλεκτρονικού ταχυδρομείου του, τον κωδικό της επιλογής του, την ηλικία του και την χώρα στην οποία διαμένει. Ύστερα και εφόσον έχει συμπληρώσει σωστά τα στοιχεία του, μπορεί να πατήσει το κουμπί “Go” για να συνδεθεί και να συνεχίσει την περιήγησή του στην εφαρμογή. Αν ο χρήστης έχει ήδη λογαριασμό, μπορεί να συνδεθεί πατώντας πάνω στην επιλογή “Already have an account? Login”. Έπειτα, θα μπορεί να συνδεθεί στην εφαρμογή με τα στοιχεία που εισήγαγε κατά την εγγραφή του.

- Email: Πρέπει να εισαχθεί ένας σωστός τύπος διεύθυνσης ηλεκτρονικού ταχυδρομείου.
- Password: Πρέπει να αποτελείται από τουλάχιστον 4 λατινικούς χαρακτήρες, να περιέχει τουλάχιστον ένα κεφαλαίο και ένα πεζό γράμμα, τουλάχιστον έναν ειδικό χαρακτήρα, τουλάχιστον έναν αριθμό και να μην υπάρχουν καθόλου κενά.
- Full name: Πρέπει να εισαχθεί ένας σωστός τύπος ονόματος.
- Username: Πρέπει να αποτελείται από 4 μέχρι 20 λατινικούς χαρακτήρες και να μην υπάρχουν καθόλου κενά.
- Age: Δεκτές είναι μόνο ακέραιες τιμές από 1 μέχρι 99.
- Country: Ο χρήστης επιλέγει μια από τις διαθέσιμες χώρες που υποστηρίζονται.



Όλα τα πεδία είναι υποχρεωτικά.

Να σημειωθεί ότι στο πεδίο του κωδικού υπάρχει το κουμπί  , με το οποίο ο χρήστης μπορεί να αποκρύψει και να κρύψει αντίστοιχα τους χαρακτήρες του κωδικού που έχει πληκτρολογήσει.

The first screenshot shows the initial sign-up screen with empty input fields for Full Name, Username, Email, Password, and Age. It includes a dropdown for selecting the United States and a blue 'GO' button. The second screenshot shows a dropdown menu titled 'Select a country' with a search bar containing 'Search...'. A list of countries is shown with their flags and names: Albania (AL), Bulgaria (BG), China (CN), Croatia (HR), Czech Republic (CZ), Denmark (DK), United Kingdom (GB), United States (US), Australia (AU), France (FR), and Canada (CA). The third screenshot shows the same dropdown menu with the search bar containing 'gree'. The results have been narrowed down to show only Greece (GR).

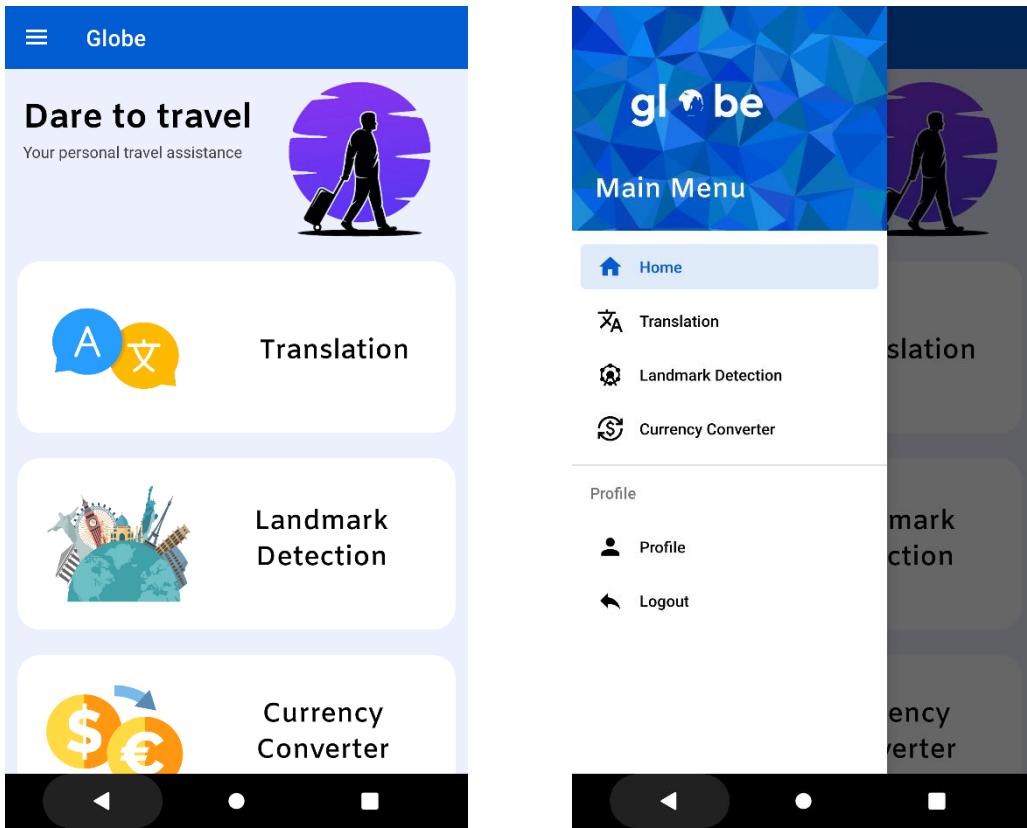
Φυσικά έχουν υλοποιηθεί όλοι οι απαραίτητοι έλεγχοι για την εγκυρότητα των στοιχείων που εισάγει ο χρήστης. Παρακάτω ακολουθούν screenshots που επιδεικνύουν την λανθασμένη συμπλήρωση πεδίων.

The left screenshot shows the sign-up screen with validation errors for all fields. The 'Full Name' field has a red exclamation mark and a tooltip 'Field cannot be empty'. The 'Username' field has a red exclamation mark and a tooltip 'Field cannot be empty'. The 'Email' field has a red exclamation mark and a tooltip 'Field cannot be empty'. The 'Password' field has a red exclamation mark and a tooltip 'Field cannot be empty'. The 'Age' field has a red exclamation mark and a tooltip 'Field cannot be empty'. The right screenshot shows validation errors for the 'Email' field ('Invalid email address') and the 'Age' field ('0').



Main Menu:

Μετά την επιτυχή σύνδεση στον λογαριασμό του, ο χρήστης μεταφέρεται στο κεντρικό μενού της εφαρμογής. Το activity αυτό αποτελεί μια οθόνη «διαμεσολαβητή», μέσω της οποίας ο χρήστης μπορεί να περιηγηθεί σε όλες τις λειτουργίες της εφαρμογής. Του δίνεται η δυνατότητα να επιλέξει μια από τις τρεις διαθέσιμες λειτουργίες, μετάφρασης, αναγνώρισης αξιοθέατων και μετατροπής συναλλάγματος. Επίσης, υπάρχει ένα πλαϊνό υπό-μενού μέσω του οποίου μπορεί να ανακατευθυνθεί σε μία από τις τρεις λειτουργίες, όπως αναφέρθηκε προηγουμένως, αλλά και να δει το προφίλ του και να αποσυνδεθεί από τον λογαριασμό του.



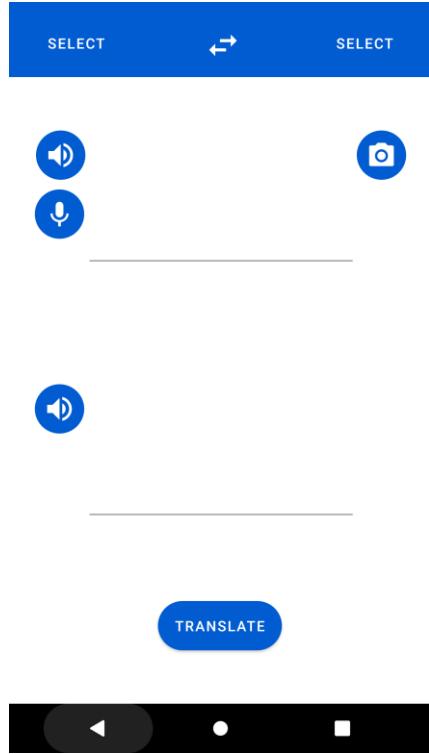
Translation:

Στο συγκεκριμένο activity είναι υλοποιημένες οι βασικότερες λειτουργίες για την μετάφραση. Ο χρήστης μπορεί να εισάγει ένα κείμενο το οποίο θα μεταφραστεί από και σε όποια γλώσσα το επιθυμεί. Επιπροσθέτως, υπάρχει η δυνατότητα για ανάγνωση και των δύο κειμένων με χρήση της τεχνολογίας Text To Speech (TTS) και εισαγωγής ενός κειμένου μέσω ομιλίας, με χρήση της τεχνολογίας Speech To Text (STT). Ακόμα, δίνεται η δυνατότητα αναγνώρισης κειμένου από φωτογραφία, είτε μέσω της συλλογής του κινητού, είτε μέσω της κάμερας.

Οι γλώσσες που υποστηρίζονται από την εφαρμογή για μετάφραση είναι:

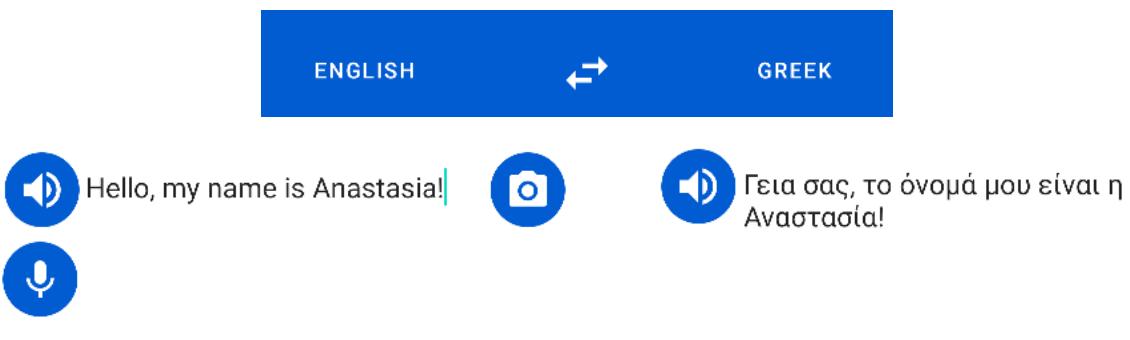
Αλβανικά, Βουλγάρικα, **Κινέζικα**, Κροατικά, Τσέχικα, Δανέζικα, **Αγγλικά**, **Γαλλικά**, **Γερμανικά**, Ελληνικά, Ινδικά, **Ιταλικά**, **Γιαπωνέζικα**, Πολωνικά, Πορτογαλικά, Ρουμάνικα, Ρωσικά, Ισπανικά, Σουηδικά, Ουκρανικά.

Οι γλώσσες που είναι τονισμένες με **μπλε** χρώμα, υποστηρίζονται επίσημα και στην λειτουργία ανάγνωσης κειμένου.



- Μέσω των κουμπιών ο χρήστης μπορεί να επιλέξει τις γλώσσες στις οποίες επιθυμεί να μεταφράσει το κείμενο που έχει εισάγει στο πρώτο πεδίο.
- Με το κουμπί γίνεται η εναλλαγή των γλωσσών που έχουν επιλεγεί (από – προς).
- Με τα κουμπιά ξεκινά η έναρξη της ανάγνωσης κειμένου του κατάλληλου πεδίου (TTS).
- Με το κουμπί ξεκινά η έναρξη μετατροπής ομιλίας σε κείμενο (STT).
- Με το κουμπί ο χρήστης μεταφέρεται σε ξεχωριστό activity για να επιλέξει μια εικόνα, με σκοπό να πραγματοποιηθεί η αναγνώριση κειμένου.
- Με το κουμπί γίνεται η μετάφραση του κειμένου που έχει εισάγει ο χρήστης στο πρώτο πεδίο.

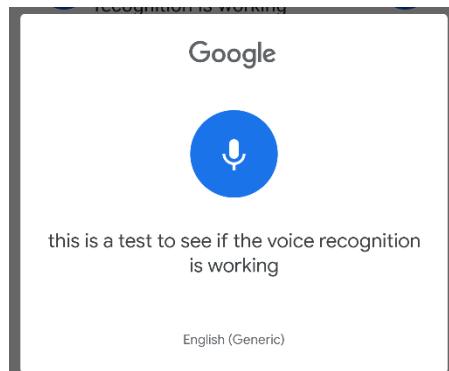
Ακολουθούν ενδεικτικά screenshots που δείχνουν την λειτουργικότητα της μετάφρασης.





FRENCH ENGLISH

Bonjour tu as quel âge? Hello you have how old?



Select Language:

Ο χρήστης μεταφέρεται στο συγκεκριμένο activity πατώντας σε ένα από τα κουμπιά “Select” με σκοπό να επιλέξει μια γλώσσα με την οποία θέλει να πραγματοποιηθεί η μετάφραση. Για την επιλογή αρχικής γλώσσας, του εμφανίζονται όλες οι υποστηριζόμενες γλώσσες, καθώς και η επιλογή “Auto”, δηλαδή αυτόματη αναγνώριση της γλώσσας που έχει πληκτρολογήσει (Language Identification). Ενώ όσον αφορά την επιλογή της τελικής γλώσσας, του εμφανίζονται μόνο οι διαθέσιμες γλώσσες που υποστηρίζει η εφαρμογή. Επιπλέον, και στις δύο περιπτώσεις, έχει την επιλογή να αναζητήσει την γλώσσα που επιθυμεί για μεγαλύτερη ευκολία.

Globe

| |
|-----------|
| Auto |
| Albanian |
| Bulgarian |
| Chinese |
| Croatian |
| Czech |
| Danish |
| English |

Globe

| |
|-----------|
| Albanian |
| Bulgarian |
| Chinese |
| Croatian |
| Czech |
| Danish |
| English |
| French |

← g ×

| |
|--------|
| German |
| Greek |



Select Image:

Ο χρήστης μεταφέρεται στο συγκεκριμένο activity, όταν θέλει να επιλέξει μία εικόνα για αναγνώριση κειμένου. Υπάρχει η δυνατότητα να τραβήξει εκείνη την στιγμή μια φωτογραφία το κείμενο που επιθυμεί ή να διαλέξει μια φωτογραφία από την συλλογή του. Μόλις διαλέξει την εικόνα που επιθυμεί, πατώντας το κουμπί “Results” μεταφέρεται πίσω στο Translation activity, για να πραγματοποιηθεί η μετάφραση.

Ακολουθούν screenshots με παραδείγματα και για τις δύο περιπτώσεις.

Από την συλλογή του κινητού:

The screenshot shows the mobile application's interface. At the top, there are two buttons: "CAMERA" and "GALLERY". Below them is a blue bar with "AUTO" on the left, a double-headed arrow icon in the center, and "FRENCH" on the right. On the left side of the screen is a camera viewfinder showing an airport terminal sign with "DEPARTURE" and "ARRIVALS" directions. In the center, there are three icons: a speaker icon labeled "DEPARTURE ARRIVALS", a camera icon, and a microphone icon. To the right, a blue circle contains a speaker icon with the text "Arrivées de départ". At the bottom left is a blue button labeled "RESULTS".

Από φωτογραφία μέσω της κάμερας:

The screenshot shows the mobile application's interface. At the top, there are two buttons: "CAMERA" and "GALLERY". Below them is a blue bar with "AUTO" on the left, a double-headed arrow icon in the center, and "GREEK" on the right. On the left side of the screen is a camera viewfinder showing a TGI Fridays sign with the text "FRIDAYS IN HERE IT'S ALWAYS FRIDAY". In the center, there are three icons: a speaker icon labeled "FRIDAYS IN HERE IT'S ALWAYS FRIDAY", a camera icon, and a microphone icon. To the right, a blue circle contains a speaker icon with the text "Παρασκευές εδώ είναι πάντα Παρασκευή". At the bottom left is a blue button labeled "RESULTS".



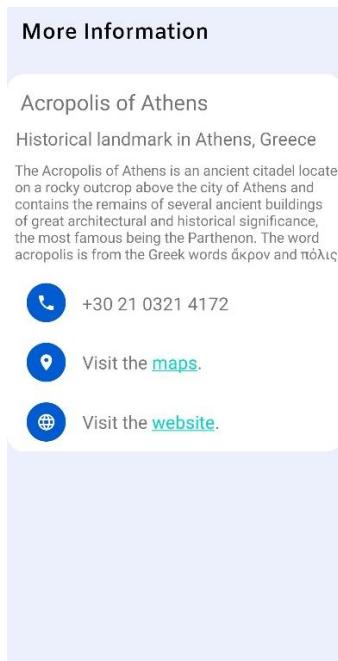
Landmarks:

Στο συγκεκριμένο activity, ο χρήστης ενημερώνεται ότι μπορεί να τραβήξει μια φωτογραφία του αξιοθέατου που επιθυμεί, ώστε να γίνει η αναγνώρισή του από την εφαρμογή. Στην συνέχεια, και μετά από επιτυχή αναγνώρισή του, ο χρήστης μπορεί να ενημερωθεί με περισσότερες πληροφορίες για το αξιοθέατο που «σκάναρε», πατώντας το κουμπί “More Info”.



Search Landmark:

Η εμφάνιση αυτού του activity γίνεται μόλις ο χρήστης πατήσει στην επιλογή “More Info” και εμφανίζει στον χρήστη πληροφορίες σχετικές με το αξιοθέατο που εντόπισε η εφαρμογή. Πιο συγκεκριμένα απεικονίζονται ο τίτλος του αξιοθέατου, μια μικρή περιγραφή για αυτό, ένα τηλέφωνο επικοινωνίας, ένας σύνδεσμος για την τοποθεσία του στο Google Maps και ένας σύνδεσμος προς την ιστοσελίδα του.





Currency Converter:

Στο αυτό το activity είναι υλοποιημένη η λειτουργία για την μετατροπή συναλλάγματος. Ο χρήστης μπορεί να εισάγει στο πρώτο πεδίο κειμένου το ποσό που θέλει να μετατρέψει. Επίσης, πρέπει να επιλέξει από τα dropdown menu τα νομίσματα για τα οποία θέλει να γίνει η μετατροπή. Μόλις πατήσει το κουμπί “Convert”, θα του εμφανιστεί η τιμή στην οποία αντιστοιχεί στο αρχικό ποσό που εισήγαγε, στο δεύτερο πεδίο κειμένου.

Τα νομίσματα τα οποία υποστηρίζει η εφαρμογή είναι:

- | | |
|--------------------------------|---------------------------------------|
| 1. ALL (Albania Lek) | 11. PLN (Poland Zloty) |
| 2. BGN (Bulgaria Lev) | 12. RON (Romania leu) |
| 3. CNY (China Yuan Renminbi) | 13. RUB (Russia Ruble) |
| 4. HRK (Croatia Kuna) | 14. SEK (Sweden Krona) |
| 5. CZK (Czech Republic Koruna) | 15. UAH (Ukraine Hryvnia) |
| 6. DKK (Denmark Krone) | 16. USD (United States Dollar) |
| 7. GBP (United Kingdom Pound) | 17. AUD (Australia Dollar) |
| 8. EUR (Euro Member Countries) | 18. CAD (Canada Dollar) |
| 9. INR (India Rupee) | 19. CHF (Switzerland Franc) |
| 10. JPY (Japan Yen) | 20. AED (United Arab Emirates Dirham) |

Ακολουθούν μερικά ενδεικτικά screenshots που δείχνουν την λειτουργικότητα του activity.

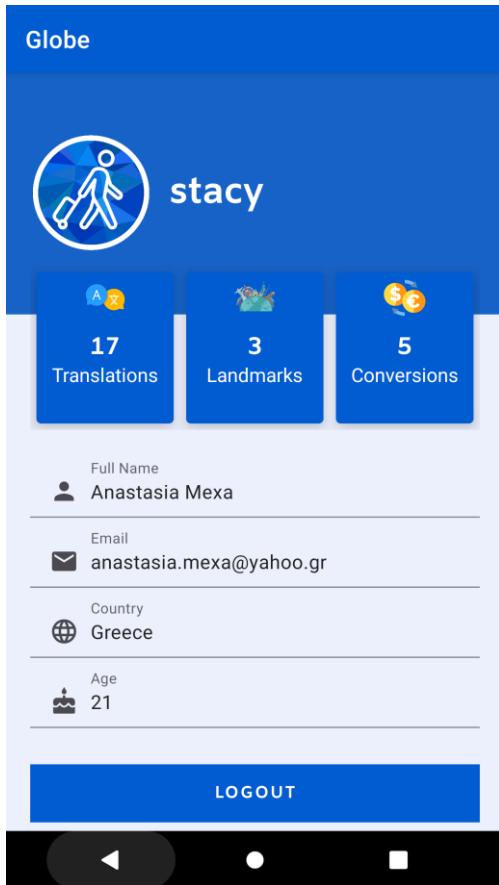
The screenshots illustrate the flow of the currency conversion process:

- Screenshot 1: Initial State**
The user has selected "EUR (Euro Member Countries, €)" as the source currency and entered "100" as the value. The target currency is set to "RON (Romania Leu, lei)". The converted amount is displayed as "481.42".
- Screenshot 2: Intermediate State**
The user has changed the source currency to "GBP (United Kingdom Pound, £)" and entered "14" as the value. The target currency is still set to "RON (Romania Leu, lei)". The converted amount is displayed as "16.11".
- Screenshot 3: Final State**
The user has changed the target currency to "AED (United Arab Emirates Dirham, ₦)" and entered "1" as the value. The source currency is now set to "USD (United States Dollar, \$)". The converted amount is displayed as "0.27".



User Profile:

Το προφίλ του χρήστη είναι υλοποιημένο στο User Profile μέσω του οποίου, ο χρήστης μπορεί να δει κάποιες πληροφορίες που τον αφορούν, τις οποίες είχε δώσει κατά την εγγραφή του και κάποια στατιστικά. Συγκεκριμένα, εμφανίζονται τα πλήθη των μεταφράσεων και των μετατροπών συναλλάγματος που έχει πραγματοποιήσει καθώς και των αξιοθέατων που έχουν αναγνωριστεί. Ακόμα εμφανίζονται το ονοματεπώνυμο, το email του, η χώρα που έχει δηλώσει και η ηλικία του. Τέλος, έχει την επιλογή να αποσυνδεθεί από τον λογαριασμό του, πατώντας στο κουμπί “Logout”.





6 Αρχιτεκτονική συστήματος

Σε αυτό το σημείο θα παρουσιαστούν τα εργαλεία και οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Συγκεκριμένα, θα γίνει αναφορά στην βάση δεδομένων, στην υποστήριξη των ελληνικών, στην ανάπτυξη του προφίλ χρήστη και των λειτουργιών Authentication, Currency Converter και Landmark Detection.

6.1 Android Studio

Φυσικά πρέπει να σημειωθεί ότι, για την ανάπτυξη της εφαρμογής και για την συγγραφή του Java κώδικα χρησιμοποιήθηκε το Android Studio (έκδοση Chipmunk | 2021.2.1 Patch 2), που αποτελεί το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για την Google.



6.2 Firebase

H Firebase είναι μία πλατφόρμα που αναπτύχθηκε από τη Google για τη δημιουργία εφαρμογών για κινητά και web. Έχει γίνει χρήση τριών υπηρεσιών που παρέχονται από την Firebase και αυτές είναι το Authentication, το Storage και η Realtime Database. Παρακάτω ακολουθεί ανάλυση και των τριών αυτών υπηρεσιών και πως συμβάλουν εν τέλη στην ανάπτυξη της εφαρμογής.



Authentication:

Έχει γίνει χρήση της λειτουργίας αυθεντικοποίησης των χρηστών μέσω email και password. Το Firebase Authentication SDK παρέχει μεθόδους για τη δημιουργία και τη διαχείριση χρηστών που χρησιμοποιούν τις διευθύνσεις ηλεκτρονικού ταχυδρομείου και τους κωδικούς πρόσβασης για να συνδεθούν. Ακολουθεί screenshot μέσα από την κονσόλα διαχείρισης της Firebase, που δείχνει δύο ενδεικτικούς χρήστες που έχουν εγγραφεί στην εφαρμογή.

| Identifier | Providers | Created ↑ | Signed In | User UID |
|--------------------------|-----------|-------------|--------------|------------------------------|
| nancygianina11@gmail.com | ✉ | Jun 8, 2022 | Aug 24, 2022 | 7iOkeQNjZ4PcRqOSqlY7GsYzCPu2 |
| anastasia.mexa@yahoo.gr | ✉ | Jun 8, 2022 | Sep 4, 2022 | 8aRCE7RcPvRW6qeK59eo8sQJy... |



Storage:

Το Storage της Firebase δίνει την δυνατότητα αποθήκευσης αρχείων όπως φωτογραφιών και βίντεο στο cloud. Στην περίπτωση της συγκεκριμένης εφαρμογής, έχει γίνει χρήση της υπηρεσίας Storage, για την αποθήκευση των εικονιδίων των σημαιών των χωρών, που χρησιμοποιούνται στο activity Select Language, όταν ο χρήστης διαλέγει γλώσσα για την μετάφραση. Ακολουθεί screenshot μέσα από την κονσόλα διαχείρισης της Firebase.

The screenshot shows the Firebase Storage interface. On the left is a list of files in the 'Flags' bucket:

| Name | Size | Type | Last modified |
|--------------------|----------|-----------|---------------|
| albania.png | 17.46 KB | image/png | Jun 8, 2022 |
| auto.png | 832 B | image/png | Jun 8, 2022 |
| bulgaria.png | 12.81 KB | image/png | Jun 8, 2022 |
| china.png | 22.14 KB | image/png | Jun 8, 2022 |
| croatia.png | 15.82 KB | image/png | Jun 8, 2022 |
| czech-republic.png | 15.85 KB | image/png | Jun 8, 2022 |
| denmark.png | 16.38 KB | image/png | Jun 8, 2022 |
| france.png | 16.01 KB | image/png | Jun 8, 2022 |
| germany.png | 11.36 KB | image/png | Jun 8, 2022 |
| greece.png | 14.39 KB | image/png | Jun 8, 2022 |

On the right, a detailed view of the 'greece.png' file is shown:

- Name: greece.png
- Size: 14,731 bytes
- Type: image/png
- Created: Jun 8, 2022, 3:04:20 PM
- Updated: Jun 8, 2022, 3:04:20 PM
- File location
- Other metadata

The image itself is the flag of Greece, featuring the blue and white horizontal stripes and the white cross in the center.

Realtime Database:

Η Realtime Database της Firebase είναι μια βάση δεδομένων που φιλοξενείται στο cloud. Τα δεδομένα αποθηκεύονται ως JSON και συγχρονίζονται σε πραγματικό χρόνο με κάθε συνδεδεμένο πελάτη. Δεν αποτελεί μια σχεσιακή βάση δεδομένων και ως εκ τούτου έχει διαφορετικές βελτιστοποιήσεις και λειτουργικότητα από μια συμβατική βάση δεδομένων.

Για τις ανάγκες τις εφαρμογής, η Realtime Database χρησιμοποιείται για την αποθήκευση πληροφοριών που αφορούν τις γλώσσες που υποστηρίζονται από την μετάφραση, πληροφορίες σχετικά με τους χρήστες που δίνονται κατά την εγγραφή τους και αποθηκεύονται επίσης μερικά στατιστικά. Τα στατιστικά απαρτίζονται από όλες τις μεταφράσεις, τις μετατροπές συναλλάγματος και τις αναγνωρίσεις αξιοθέατων που έχουν πραγματοποιηθεί από τους χρήστες της εφαρμογής. Τα στατιστικά αυτά μπορούν να προσπελαστούν και να εμφανιστούν πιο «όμορφα», μέσω της ιστοσελίδας που έχει δημιουργηθεί για τον σκοπό αυτό.

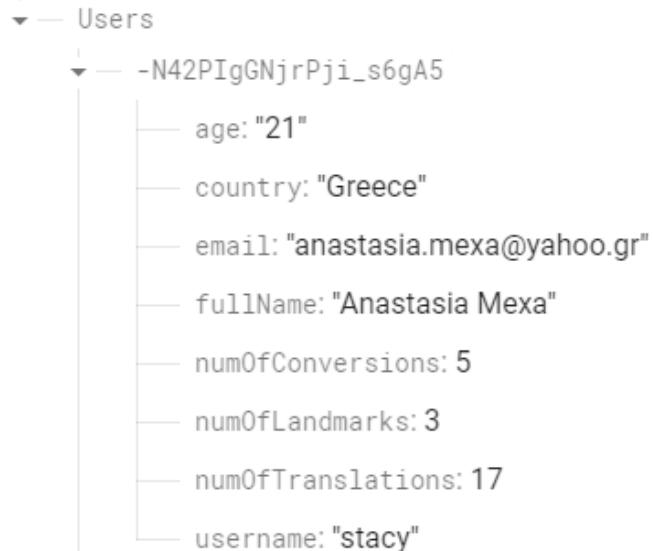
Για τις γλώσσες οι πληροφορίες που είναι αποθηκευμένες στην βάση δεδομένων αφορούν, το όνομα του αρχείου για το εικονίδιο που είναι αποθηκευμένο στο Storage, τον τίτλο της γλώσσας και τέλος οι κωδικοί των χωρών σε ISO 639-2 και ISO 639-1 μορφή. Ακολουθεί screenshot μέσα από την κονσόλα διαχείρισης της Firebase.

The screenshot shows the Firebase Realtime Database structure under the 'Languages' node:

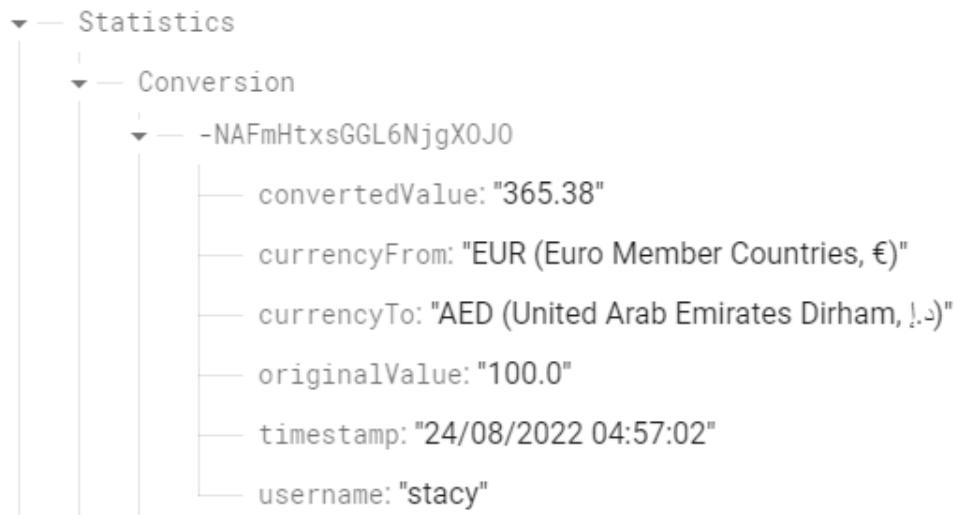
- Languages
 - Albanian
 - Icon: "albania.png"
 - Short: "ALB"
 - Tag: "sq"
 - Title: "Albanian"



Για τους χρήστες οι πληροφορίες που είναι αποθηκευμένες στην βάση δεδομένων αφορούν, την ηλικία, την χώρα, το email, το ονοματεπώνυμο, το πλήθος των μετατροπών συναλλάγματος, το πλήθος των αναγνωρισμένων αξιοθέατων, το πλήθος των μεταφράσεων και το όνομα χρήστη. Ακολουθεί screenshot μέσα από την κονσόλα διαχείρισης της Firebase.



Για τα στατιστικά των μετατροπών συναλλάγματος οι πληροφορίες που είναι αποθηκευμένες στην βάση δεδομένων αφορούν, την αρχική τιμή, την τελική τιμή, το αρχικό νόμισμα, το τελικό νόμισμα, το όνομα χρήστη που πραγματοποίησε την μετατροπή και ένα timestamp για το πότε έγινε η μετατροπή. Ακολουθεί screenshot μέσα από την κονσόλα διαχείρισης της Firebase.



Για τα στατιστικά των αναγνωρισμένων αξιοθέατων οι πληροφορίες που είναι αποθηκευμένες στην βάση δεδομένων αφορούν, το όνομα του αξιοθέατου που αναγνωρίστηκε, οι συντεταγμένες του, το όνομα χρήστη που πραγματοποίησε την αναγνώριση και ένα timestamp για το πότε έγινε αυτή. Ακολουθεί screenshot μέσα από την κονσόλα διαχείρισης της Firebase.



```

    - Landmark
      - -NAO_Ss1CVLsSL0_Lf30
        - coordinates
          - latitude: 37.9715323
          - longitude: 23.7257492
        - landmark: "Acropolis of Athens"
        - timestamp: "26/08/2022 12:57:36"
        - username: "stacy"
  
```

Για τα στατιστικά των μεταφράσεων οι πληροφορίες που είναι αποθηκευμένες στην βάση δεδομένων αφορούν, το αρχικό κείμενο, το μεταφρασμένο κείμενο, την αρχική γλώσσα, την τελική γλώσσα, το όνομα χρήστη που πραγματοποίησε την μετάφραση και ένα timestamp για το πότε έγινε η μετάφραση. Ακολουθεί screenshot μέσα από την κονσόλα διαχείρισης της Firebase.

```

    - Translation
      - -NAFm9LZjm1YdnFT1zdJ
        - languageFrom: "auto"
        - languageTo: "el"
        - originalText: "Hello"
        - timestamp: "24/08/2022 04:56:27"
        - translatedText: "Γεια σας"
        - username: "stacy"
  
```



6.3 Permissions και dependencies

Στο αρχείο AndroidManifest υπάρχουν τα uses-permissions, τα οποία ουσιαστικά ζητούν μερικά δικαιώματα από τον χρήστη κατά την λειτουργία της εφαρμογής. Συγκεκριμένα, τα δικαιώματα που ζητά η εφαρμογή από τον χρήστη αφορούν την δυνατότητα χρήσης του Internet, του μικροφώνου, της κάμερας και δυνατότητα ανάγνωσης του χώρου αποθήκευσης. Ο κώδικας που είναι υπεύθυνος για την διαχείριση των permissions απεικονίζεται παρακάτω.

```
<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" />
```

Όσον αφορά τα dependencies, βρίσκονται μέσα στο gradle αρχείο και αφορούν κάποια στοιχεία από εξωτερικές βιβλιοθήκες, τα οποία βοηθάνε στην ομαλή λειτουργία της εφαρμογής. Τα dependencies του προγράμματος απεικονίζονται στην εικόνα που ακολουθεί.

```
@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.firebaseio:firebase-database:20.0.4'
    implementation 'com.google.firebaseio:firebase-auth:21.0.1'
    implementation 'com.google.firebaseio: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:ccb: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'
}
```



6.4 Υποστήριξη ελληνικής γλώσσας

Για να είναι δυνατή η υποστήριξη των ελληνικών από την εφαρμογή, έχει γίνει χρήση του αρχείου strings.xml. Όλα τα κείμενα της εφαρμογής έχουν γραφτεί σε μορφή εντολών XML, δίνοντας ένα id και το αρχικό κείμενο. Επιπλέον, έχει γίνει προσθήκη της ελληνικής γλώσσας στο Translations Editor του Android Studio και κάθε κείμενο έχει μεταφραστεί στο αντίστοιχό του στα ελληνικά. Η γλώσσα της εφαρμογής αλλάζει ανάλογα με την γλώσσα που ο χρήστης έχει ορίσει ως προκαθορισμένη στο κινητό του. Εννοείται ότι μπορούν να προστεθούν και άλλες γλώσσες, με σκοπό την επέκταση της προσβασιμότητας της εφαρμογής.

```
<resources>
    <string name="app_name">Globe</string>
    <string name="navigation_drawer_open" translatable="false">Open navigation drawer</string>
    <string name="navigation_drawer_close" translatable="false">Close navigation drawer</string>
    <string name="welcome_back">Hello there, Welcome back</string>
    <string name="sign_in">Sign in to continue</string>
    <string name="email_hint" translatable="false">Email</string>
    <string name="password_hint">Password</string>
```

| Key | Resource Folder | Untranslatable | Default Value | Greek (el) in Greece (GR) |
|-------------------------|------------------|-------------------------------------|---------------------------|------------------------------|
| app_name | app/src/main/res | <input type="checkbox"/> | Globe | Globe |
| navigation_drawer_open | app/src/main/res | <input checked="" type="checkbox"/> | Open navigation drawer | |
| navigation_drawer_close | app/src/main/res | <input checked="" type="checkbox"/> | Close navigation drawer | |
| welcome_back | app/src/main/res | <input type="checkbox"/> | Hello there, Welcome back | Γειά σας, Καλωσήρθατε ξανά |
| sign_in | app/src/main/res | <input type="checkbox"/> | Sign in to continue | Συνδεθείτε για να συνεχίσετε |
| email_hint | app/src/main/res | <input checked="" type="checkbox"/> | Email | |
| password_hint | app/src/main/res | <input type="checkbox"/> | Password | Κωδικός |



Γειά σας,
Καλωσήρθατε ξανά

Συνδεθείτε για να συνεχίσετε

| | |
|----------------|-----------------------|
| Email | |
| Κωδικός | <input type="radio"/> |
| ΣΥΝΔΕΣΗ | |

ΝΕΟΣ ΧΡΗΣΤΗΣ; ΕΓΓΡΑΦΕΙΤΕ

globe

Καλωσήρθατε,

Εγγραφείτε για να ξεκινήσει το ταξίδι σας

| | |
|---------------|-----------------------|
| Όνοματεπώνυμο | |
| Username | |
| Email | |
| Κωδικός | <input type="radio"/> |
| Ηλικία | |

Ελληνική

Greece

ΣΥΝΔΕΣΗ

ΕΧΕΤΕ ΉΔΗ ΛΟΓΑΡΙΑΣΜΟ; ΣΥΝΔΕΘΕΙΤΕ

Τόλμησε να
ταξιδέψεις

Ο προσωπικός σου ταξιδιωτικός
βοηθός



Μετάφραση



Αναγνώριση
Αξιοθεάτων



Μετατροπή
Συναλλάγματος



6.5 Προφίλ χρήστη

Πριν γίνει αναφορά στην υλοποίηση του προφίλ χρήστη, πρέπει να διευκρινιστεί ο τρόπος με τον οποίο συλλέγονται τα στατιστικά χρήσης της εφαρμογής. Δηλαδή, πως καταγράφονται τα δεδομένα μεταφράσεων, μετατροπών συναλλάγματος και αναγνώρισης αξιοθέατων αντίστοιχα. Επειδή ο κώδικας αποθήκευσης των δεδομένων αυτών είναι παρόμοιος κάθε φορά, θα γίνει αναφορά μόνο στην περίπτωση των μετατροπών συναλλάγματος και εννοείται ότι αντίστοιχα ο κώδικας λειτουργεί και για τις άλλες δύο περιπτώσεις.

Κατά την επιτυχή μετατροπή ενός ποσού συναλλάγματος σε ένα άλλο, γίνεται ένα αίτημα προς την βάση δεδομένων, για την επιστροφή του πλήθους των μετατροπών που έχει πραγματοποιήσει ο χρήστης μέχρι εκείνη την στιγμή. Έπειτα, προστίθεται συνέπεια στο πλήθος αυτό και γίνεται η αποθήκευση των λεπτομερειών της μετατροπής στην βάση, καθώς και ενημέρωση του πεδίου που αφορά το συνολικό πλήθος των μετατροπών.

```
// Update the number of conversions in database
database.addValueEventListener(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())){
                // Incrementing the total number of conversions for the current user
                int sum = user.getNumOfConversions() + 1;
                String key = dataSnapshot.getKey();
                // Updating value
                database.child(key).child("numOfConversions").setValue(sum);

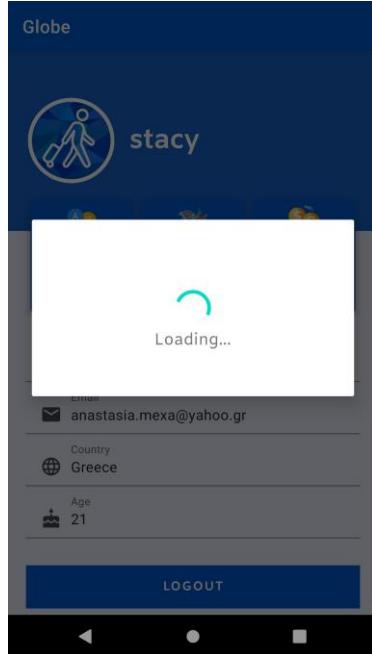
                // Archive every conversion in database
                FirebaseDatabase database1 = FirebaseDatabase.getInstance();
                DatabaseReference mRef = database1.getReference().child("Statistics").child("Conversion").push();
                mRef.child("currencyFrom").setValue(fromCurr);
                mRef.child("currencyTo").setValue(toCurr);
                mRef.child("originalValue").setValue(value.toString());
                mRef.child("convertedValue").setValue(textView.getText().toString());
                mRef.child("username").setValue(user.getUsername());
                mRef.child("timestamp").setValue(new SimpleDateFormat("dd/MM/yyyy hh:mm:ss").format(new Date()));
                break;
            }
        }
    }
})
```

Συνεπώς, με αυτόν τον τρόπο γίνεται η καταγραφή των στατιστικών στοιχείων από την εφαρμογή. Άρα τώρα θα γίνει αναφορά στην υλοποίηση του προφίλ χρήστη.



User Profile:

Στην αρχή της εκτέλεσης του activity εμφανίζεται ένα loading dialog για τρία δευτερόλεπτα, μέχρι να πραγματοποιηθεί η σωστή φόρτωση των στοιχείων του χρήστη από την βάση δεδομένων.



Πολύ απλά στην συνέχεια, γίνεται ένα αίτημα προς την βάση δεδομένων για την ανάκτηση των δεδομένων του χρήστη. Συγκεκριμένα, δημιουργείται ένα αντικείμενο τύπου User και μετά με την χρήση getters, ορίζεται η τιμή των κατάλληλων πεδίων.

```
// 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;
            }
        }
    }

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



6.6 Λειτουργία Authentication

Έχουν υλοποιηθεί δύο activity τα οποία συμβάλουν στην λειτουργία της αυθεντικοποίησης των χρηστών, το Authentication και το Sign Up, τα οποία έχουν αναλυθεί προηγουμένως. Σε αυτό το σημείο, θα γίνει μια παρουσίαση των βασικότερων κομματιών κώδικα που επιτελούν στην λειτουργία της αυθεντικοποίησης.

Sign Up:

Αρχικά, έχουν υλοποιηθεί μερικές μέθοδοι που είναι υπεύθυνες για τον έλεγχο ορθής συμπλήρωσης των στοιχείων του χρήστη. Συγκεκριμένα, αυτές αφορούν τον έλεγχο για το ονοματεπώνυμο, το όνομα χρήστη, την διεύθυνση του ηλεκτρονικού ταχυδρομείου, τον κωδικό και την ηλικία.

```
// ===== Methods for validating user input =====
private Boolean validateName() {...}

private Boolean validateUsername() {...}

private Boolean validateEmail() {...}

private Boolean validatePassword() {...}

private Boolean validateAge() {...}
// =====
```

Στην συνέχεια, όταν ο χρήστης πατήσει στο κουμπί “Go” για να συνδεθεί, καλείται η συνάρτηση `signup()`. Στην αρχή της εκτέλεσής της, καλούνται οι συναρτήσεις που αναφέρθηκαν προηγουμένως, με σκοπό τον έλεγχο της εισόδου από τον χρήστη. Αν έστω και μία από αυτές επιστρέψει ψευδή τιμή (`false`), τότε διακόπτεται η εκτέλεση της συνάρτησης. Σε διαφορετική περίπτωση, συνεχίζεται η εκτέλεση και γίνεται απόπειρα εισαγωγής του χρήστη στην βάση δεδομένων. Λεπτομερώς, δημιουργείται ένα αντικείμενο `User`, το οποίο περιέχει όλα τα δεδομένα του χρήστη και στην συνέχεια αποθηκεύεται στην βάση δεδομένων. Αν κάτι πάει λάθος, τότε ο χρήστης θα ενημερωθεί με κατάλληλο μήνυμα.

```
// Method for user sign up
public void signup(View view) {
    // Validate user input
    if(!validateName() | !validatePassword() | !validateEmail() | !validateUsername() | !validateAge()){
        return;
    }
    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, numOfTranslations: 0, numOfLandmarks: 0, numOfConversions: 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("Error", task.getException().getLocalizedMessage());
                }
            }
        });
}
```



Authentication:

Από την στιγμή που ο χρήστης διαθέτει ήδη λογαριασμό, για να χρησιμοποιήσει την εφαρμογή πρέπει να συνδεθεί σε αυτόν. Αυτόν τον σκοπό εξυπηρετεί η συνάρτηση *login()*. Αρχικά, ελέγχεται αν ο χρήστης έχει συμπληρώσει και τα δύο πεδία του email και του password και έπειτα, γίνεται κλήση της κατάλληλης συνάρτησης από την Firebase, για να διαπιστωθεί η εγκυρότητα των στοιχείων σύνδεσης του χρήστη. Σε περίπτωση επιτυχής ταυτοποίησης, ο χρήστης μεταφέρεται στο κεντρικό μενού της εφαρμογής (Main Menu activity), διαφορετικά ενημερώνεται με κατάλληλο μήνυμα λάθους.

```
// 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(getApplicationContext(), MainMenu.class);
                        startActivity(intent);
                    } else {
                        showMessage(getString(R.string.error), task.getException().getLocalizedMessage());
                    }
                }
            });
    } else {
        showMessage(getString(R.string.error), getString(R.string.fill_fields));
    }
}
```



6.7 Λειτουργία Currency Converter

Για την υλοποίηση της συγκεκριμένης λειτουργίας, έχει γίνει χρήση του [Exchange Rates Data API](#) όπου από εκεί λαμβάνεται η πληροφορία σχετικά με τις ισοτιμίες. Όλα τα δεδομένα από το API ανακτώνται μέσω JSON requests και responses. Πρώτα θα γίνει αναφορά στον τρόπο που λαμβάνονται τα δεδομένα και θα παρουσιαστούν παραδείγματα μέσα από την πλατφόρμα του Postman, με σκοπό την επίτευξη της καλύτερης δυνατής κατανόησης του τρόπου λειτουργίας των αιτημάτων.

Για να είναι δυνατή η χρήση του API έχει δημιουργηθεί ένας δωρεάν λογαριασμός σε αυτό, με περιορισμό τα 250 αιτήματα κάθε μήνα. Με την δημιουργία του λογαριασμού, παρέχεται ένα κλειδί αυθεντικοποίησης από το API, μέσω του οποίου γίνονται αποδεκτά τα αιτήματα. Το κλειδί αυτό τοποθετείται σε κάθε request στο header του URL, όπως απεικονίζεται στο παράδειγμα παρακάτω.

The screenshot shows the 'Authorization' tab in Postman. Under 'Type', 'API Key' is selected. In the 'Key' field, 'apikey' is entered. In the 'Value' field, a placeholder 'synkfr4QInqxDPR8FZfk9WgY7O6wdAdV' is shown. The 'Add to' dropdown is set to 'Query Params'.

Ένα παράδειγμα ενός αιτήματος είναι η ανάκτηση όλων των διαθέσιμων νομισμάτων που υποστηρίζονται από το API. Για να πραγματοποιηθεί το αίτημα, η δομή του URL πρέπει να έχει την εξής μορφή:

https://api.apilayer.com/exchangerates_data/symbols

Σημείωση: Το **πορτοκαλί** κομμάτι του URL θεωρείται ως βάση, η οποία παραμένει η ίδια για όλα τα αιτήματα.

Βεβαίως, μπορούν να οριστούν μερικές παράμετροι για την εξατομίκευση των αποτελεσμάτων. Για παράδειγμα, είναι δυνατή η επιστροφή αποτελεσμάτων που αφορούν συγκεκριμένα νομίσματα. Αυτό μπορεί να επιτευχθεί με την εισαγωγή των κωδικών των νομισμάτων στο URL, όπως φαίνεται παρακάτω.

The screenshot shows a POST request in Postman. The method is 'GET'. The URL is 'https://api.apilayer.com/exchangerates_data/latest?symbols=USD'. The 'Headers' tab shows 'Content-Type: application/json'. The 'Body' tab is selected, showing 'Pretty' view. The JSON response is:

```
1  "success": true,
2  "timestamp": 1662450243,
3  "base": "EUR",
4  "date": "2022-09-06",
5  "rates": {
6    "USD": 0.998328
7  }
```

Στο παραπάνω screenshot φαίνεται και η μορφή της απάντησης που λαμβάνεται από το API. Η βασικότερη πληροφορία στην απάντηση αυτή, είναι η ισοτιμία του δολαρίου σε σχέση με αυτή του ευρώ. Συγκεκριμένα, 1€ ισοδυναμεί σε περίπου 0.99\$.



Ως επόμενο παράδειγμα θα πραγματοποιηθεί στο API ένα αίτημα μετατροπής 100€ σε δολάρια.

```
GET https://api.apilayer.com/exchangerates_data/convert?to=USD&from=EUR&amount=100

Params ● Authorization ● Headers (5) Body Pre-request Script Tests Settings
Body Cookies Headers (22) Test Results
Pretty Raw Preview Visualize JSON ▾ 

```

1
2 "success": true,
3 "query": {
4 "from": "EUR",
5 "to": "USD",
6 "amount": 100
7 },
8 "info": {
9 "timestamp": 1662450663,
10 "rate": 0.99784
11 },
12 "date": "2022-09-06",
13 "result": 99.784
14

```


```

Δίνονται στο URL η πληροφορία από ποιο νόμισμα σε ποιο γίνεται η μετατροπή και το ποσό που θα μετατραπεί. Η απάντηση που στέλνει το API δείχνει ότι η αντιστοιχία σε δολάρια είναι 99.784\$.

Αφού λοιπόν έγινε αντιληπτός ο τρόπος λειτουργίας των JSON requests, θα αναλυθεί ο κώδικας της εφαρμογής.

Currency Converter:

Καταρχάς, στην μέθοδο `onCreate()` καλείται η συνάρτηση `loadConvTypes()` η οποία είναι υπεύθυνη για την φόρτωση των νομισμάτων που υποστηρίζει η εφαρμογή από το API. Χτίζεται το URL στην κατάλληλη μορφή και στην συνέχεια πραγματοποιείται το JSON request. Γίνεται ανάλυση της απάντησης και επιλέγονται οι πληροφορίες που χρειάζονται για να περαστούν σε μια λίστα. Μόλις ολοκληρωθεί η διαδικασία, τα περιεχόμενα της λίστας φορτώνονται στο spinner, ώστε τελικά να εμφανιστούν στον χρήστη σε μορφή dropdown list, από όπου μπορεί να επιλέξει το νόμισμα που επιθυμεί.



```

@Override
public void onResponse(Call call, Response response) throws IOException {
    final String mMessage = response.body().string();
    //System.out.println("!!!!!!!!!!!!!!!!!!!!!! " +mMessage);
    CurrencyConverter.this.runOnUiThread(new Runnable() {
        @Override
        public void run() {
            try {
                // Get the rates values from JSON
                JSONObject obj = new JSONObject(mMessage);
                JSONObject b = obj.getJSONObject("rates");

                Iterator keysToCopyIterator = b.keys();
                keysList = new ArrayList<>();

                int i = 0;
                while(keysToCopyIterator.hasNext()) {
                    // Add them to the list
                    String key = (String) keysToCopyIterator.next();
                    // List format: currency code (name, symbol)
                    keysList.add(key + " " + description.get(i));
                    i++;
                }

                // Set the contents of the spinners according to list contents
                ArrayAdapter<String> spinnerArrayAdapter = new ArrayAdapter<>(getApplicationContext(),
                        android.R.layout.simple_spinner_item, keysList );
                fromCurrency.setAdapter(spinnerArrayAdapter);
                toCurrency.setAdapter(spinnerArrayAdapter);
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    });
}

```

Μετά την επιτυχή φόρτωση των δεδομένων αυτών, ο χρήστης ορίζει κατάλληλα τις παραμέτρους για την μετατροπή που θέλει να πραγματοποιήσει. Όταν πατήσει το κουμπί “Convert”, τότε καλείται η συνάρτηση *convertCurrency()* η οποία είναι υπεύθυνη για την μετατροπή συναλλάγματος. Χτίζεται το URL στην κατάλληλη μορφή και στην συνέχεια πραγματοποιείται το JSON request. Γίνεται ανάλυση της απάντησης και εκτυπώνεται το αποτέλεσμα στον χρήστη. Τέλος, καταγράφονται τα στοιχεία της μετατροπής στην βάση δεδομένων, όπως επεξηγήθηκε προηγουμένως.

```

// Function that converts the currencies from API
public void convertCurrency(String fromCurr, String toCurr, Double value) throws IOException {
    // Convert the desired amount from one currency to another
    String url = "https://api.apilayer.com/exchangerates_data/convert?to=" +
                  toCurr.substring(0,3) + "&from=" + fromCurr.substring(0,3) + "&amount=" + value;

    // Make the request
    OkHttpClient client = new OkHttpClient();

    Request request = new Request.Builder()
        .url(url)
        .addHeader("apikey", "synkfr4QInqxDPR8FZfk9WgY706wdAdV")
        .build();
}

```

```

@Override
public void onResponse(Call call, Response response) throws IOException {
    final String mMessage = response.body().string();
    CurrencyConverter.this.runOnUiThread(new Runnable() {
        @Override
        public void run() {
            try {
                // Get the result value from JSON
                JSONObject obj = new JSONObject(mMessage);
                Double output = obj.getDouble("result");
                // Round result to two decimals
                textView.setText(String.format("%.2f", output));
            }
        }
    });
}

```



6.8 Λειτουργία Landmark Detection

Έχουν υλοποιηθεί δύο activity τα οποία συμβάλουν στην λειτουργία της ανίχνευσης των αξιοθέατων, το Landmarks και το SearchLandmark, τα οποία έχουν αναλυθεί προηγουμένως. Σε αυτό το κομμάτι, θα γίνει μια παρουσίαση των βασικότερων κομματιών κώδικα που επιτελούν στην λειτουργία αυτή.

Landmarks:

Εδώ έχει υλοποιηθεί όλη η λειτουργία αναγνώρισης του αξιοθέατου από την φωτογραφία που βγάζει ο χρήστης. Για να είναι δυνατή η υλοποίηση της συγκεκριμένης λειτουργίας, έχει γίνει χρήση του [Google Cloud Vision API](#) όπου από εκεί λαμβάνεται η σχετική πληροφορία. Όλα τα δεδομένα από το API ανακτώνται μέσω JSON requests και responses.

Για να είναι δυνατή η χρήση του API, έπρεπε να γίνει σύνδεση ενός Google λογαριασμού στο Google Cloud Console και να ενεργοποιηθεί ένας billing λογαριασμός. Τα αιτήματα που μπορούν να πραγματοποιηθούν χωρίς κάποια επιπλέον χρέωση είναι 1000 κάθε μήνα. Μετά την επιτυχή ολοκλήρωση των βημάτων αυτών, παρέχεται ένα κλειδί αυθεντικοποίησης από το API μέσω του οποίου γίνονται αποδεκτά τα αιτήματα. Το κλειδί αυτό είναι συμπληρωμένο μέσα στον κώδικα ως τιμή στην μεταβλητή CLOUD_VISION_API_KEY.

```
private static final String CLOUD_VISION_API_KEY = "AIzaSyAXx6hzG4aD3_gVp0xFz91Cv_JSiKrndo";
```

Ο κώδικας του συγκεκριμένου activity, βασίζεται πλήρως σε αυτόν που βρίσκεται στο documentation του Cloud Vision API, το οποίο μπορεί κανείς να επισκεφτεί μέσω του παρακάτω συνδέσμου:

https://cloud.google.com/vision/docs/detecting-landmarks#try_it

Για αυτόν τον λόγο, θα δοθεί έμφαση στην επεξήγηση του τρόπου λειτουργίας του API και στο πως ανακτάται η πληροφορία από το JSON response.

Η Google έχει δημιουργήσει μια ιστοσελίδα στα επίσημα docs του API, στην οποία μπορεί να γίνει η δοκιμή του, ώστε να επιδειχθεί ο τρόπος λειτουργίας του. Η ιστοσελίδα είναι προσβάσιμη μέσω του συνδέσμου <https://cloud.google.com/vision/docs/drag-and-drop>.

Ακολουθεί screenshot από την συγκεκριμένη ιστοσελίδα στην οποία έχει δοθεί ως είσοδος μια εικόνα της Ακρόπολης.

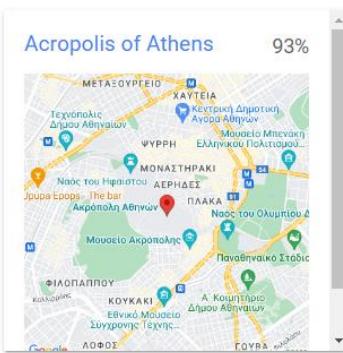
Try the API

[Landmarks](#) [Objects](#) [Labels](#) [Properties](#) [Safe Search](#)



Night-view-Parthenon-Athens.jpg

Show JSON ▾



Acropolis of Athens 93%

MΕΤΑΞΟΥΡΓΕΙΟ ΧΑΥΤΕΑ Κεντρικό Δημοτικό Ανορθό Νησάρια
Τερψιούλιο Δήμου Αθηναίων ΝΟΜΟΣ Αττικής Μουσείο Μπενάκη
ΨΥΡΗ ΕΛΛΗΝΙΚΟ ΠΑΤΡΙΟΣ
ΜΟΝΑΣΤΗΡΑΚΙ ΑΕΡΗΔΕΣ ΠΛΑΚΑ ΝΟΣ ΤΟΥ ΟΛΥΜΠΟΥ Δ
Αρρένες Μουσείο Ακρόπολης Παναθηναϊκό Στάδιος
ΦΙΛΟΠΑΠΠΟΥ ΚΟΥΚΑΚΙ Εθνικό Μουσείο Σύγχρονης Τέχνης
ΛΟΦΟΣ Α. ΚΟΥΠΤΩΤΟΣ Δήμου Αθηναίων ΓΟΥΡΑ

RESET NEW FILE



Γίνεται αντιληπτό ότι το API αναγνώρισε σωστά το αξιοθέατο και μάλιστα με σκορ 93%. Δίνεται επίσης η δυνατότητα από την ιστοσελίδα, παρουσίασης του JSON response. Παρακάτω ακολουθεί μια εικόνα ενός μέρους της απάντησης, το οποίο και αναλύεται και χρησιμοποιείται από την εφαρμογή.

```
"landmarkAnnotations": [
  {
    "boundingPoly": {
      "vertices": [
        {
          "x": 181,
          "y": 377
        },
        {
          "x": 1418,
          "y": 377
        },
        {
          "x": 1418,
          "y": 890
        },
        {
          "x": 181,
          "y": 890
        }
      ]
    },
    "description": "Acropolis of Athens",
    "locations": [
      {
        "latLng": {
          "latitude": 37.9715323,
          "longitude": 23.7257492
        }
      }
    ],
    "mid": "/m/0wjjj",
    "score": 0.9314561
  }
]
```

Η πληροφορία που είναι τονισμένη με κίτρινο χρώμα, είναι και αυτή που ανακτά η εφαρμογή. Αναλυτικά, το κείμενο του πεδίου “description” είναι και αυτό που εμφανίζεται τελικά στον χρήστη, ως το αξιοθέατο που αναγνωρίστηκε.

Ενώ οι συντεταγμένες μαζί με την περιγραφή του, αποθηκεύονται στη βάση δεδομένων στο πεδίο των στατιστικών για κάθε αξιοθέατο που αναγνωρίζεται.

```
message = entity.getDescription();
```

```
// Archive every landmark in database
FirebaseDatabase database1 = FirebaseDatabase.getInstance();
DatabaseReference mRef = database1.getReference().child("Statistics").child("Landmark").push();
mRef.child("landmark").setValue(entity.getDescription());
mRef.child("coordinates").setValue(entity.getLocations().get(0).getLatLng());
mRef.child("username").setValue(user.getUsername());
mRef.child("timestamp").setValue(new SimpleDateFormat("dd/MM/yyyy hh:mm:ss").format(new Date()));
```



Search Landmark:

Μετά την επιτυχή αναγνώριση ενός αξιοθέατου, εμφανίζεται στον χρήστη η επιλογή να δει περισσότερες πληροφορίες σχετικά με αυτό. Το Search Landmark υλοποιεί την συγκεκριμένη λειτουργία, κάνοντας χρήση ενός ακόμα API, αυτή την φορά του [Google Search API](#) από την SerpApi. Πάλι, όπως και σε όλα τα προηγούμενα API που έχουν αναλυθεί, η πληροφορία ανακτάται μέσω JSON requests και responses.

Για να είναι δυνατή η χρήση του API έχει δημιουργηθεί ένας δωρεάν λογαριασμός σε αυτό, με περιορισμό τα 100 αιτήματα κάθε μήνα. Με την δημιουργία του λογαριασμού, παρέχεται ένα κλειδί αυθεντικοποίησης από το API, μέσω του οποίου γίνονται αποδεκτά τα αιτήματα.

Το API αυτό, δίνει την δυνατότητα άντλησης της πληροφορίας που υπάρχει στο πεδίο knowledge graph ενός απλού google search. Το knowledge graph είναι ένα πλαίσιο πληροφοριών που εμφανίζονται στο Google όταν πραγματοποιείται μια αναζήτηση για μια οντότητα (άτομα, μέρη, οργανισμούς, πράγματα). Σκοπός του είναι να βοηθήσει τον χρήστη να λάβει ένα γρήγορο στιγμιότυπο πληροφοριών για αυτό που αναζητά. Περισσότερες πληροφορίες σχετικά με το knowledge graph υπάρχουν στον σύνδεσμο:

<https://support.google.com/knowledgepanel/answer/9787176?hl=en>

Παρακάτω απεικονίζεται το knowledge graph και το JSON response, σε ένα παράδειγμα αναζήτησης για το Άγαλμα της Ελευθερίας.



Τα παραδείγματα που ακολουθούν έχουν πραγματοποιηθεί ξανά μέσω της πλατφόρμας του Postman, με σκοπό να γίνει αντιληπτός ο τρόπος λειτουργίας των αιτημάτων.

Έστω ότι ο χρήστης αναζητεί σχετικές πληροφορίες για το Άγαλμα της Ελευθερίας. Για την εκτέλεση του αιτήματος πρέπει να εισαχθεί το κλειδί αυθεντικοποίησης του API στο header του URL, όπως απεικονίζεται παρακάτω.

Params • **Authorization** • Headers (5) Body Pre-request Script Tests Settings

| | |
|--|---------------------------------------|
| Type | API Key |
| The authorization header will be automatically generated when you send the request. Learn more about authorization ↗ | |
| Key | apikey |
| Value | 674b7e80b3392656c7525ae3d6b230cf4b... |
| Add to | Query Params |

Η δομή του URL πρέπει να έχει την εξής μορφή:

<https://serpapi.com/search.json?q=Statue+of+Liberty>

Σημείωση: Το πορτοκαλί κομμάτι του URL θεωρείται ως βάση, η οποία παραμένει η ίδια για όλα τα αιτήματα.

Ακολουθεί screenshot που απεικονίζει το JSON request και ένα μέρος από το response, με τονισμένες τις πληροφορίες που η εφαρμογή τελικά θα εμφανίσει στον χρήστη.

```
GET https://serpapi.com/search.json?q=Statue+of+Liberty

Params • Authorization • Headers (5) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (22) Test Results Status: 200 OK Time: 4.48 s Size: 115.44 KB Save Response ▾

Pretty Raw Preview Visualize JSON ▾ 🔍

53 "knowledge_graph": {
54     "title": "Statue of Liberty",
55     "type": "Sculpture by Frédéric Auguste Bartholdi",
56     "place_id": "ChiJPTacEpBQwokRKwIlDXelxkA",
57     "website": "https://www.nps.gov/stli/index.html",
58     "description": "The Statue of Liberty is a colossal neoclassical sculpture on Liberty Island in New York Harbor in New York City, in the United States.",
59     "local_map": {
60         "image": "https://www.google.com/maps/vt/
61             data=1rChVnp0PKIgW3x9hDEnW2nzL3FgjuLltUWAK01VCleh9pk10fzIxvarh3XCvRfG_ORE75kWrraDys7YVZbE_R7crKGUMmKE_XUmshVgiySqu_tmitA3ETwjV6YPQgIoLs2qHmnJDjokIgzzP_W5d1AuDGgGNs5inthaqSIX5-zDvoWT
62             GzHifNIAZSBZFECa_NmfEPjhkzv0zzIg",
63         "link": "https://www.google.com/maps/place/Statue+of+liberty/@40.6892494,-74.0445004,15z/data=!4m2!3m1!1s0x0:0x40c6a5770d25022b?sa=X",
64     },
65     "phone": "(212) 363-3200",
66     "address": "Ellis Island, New York, NY 10004, USA",
67     "lat": 40.6892494,
68     "lon": -74.0445004,
69     "nearby": [
70         "Ellis Island Immigration Museum"
71     ],
72     "nearby_desc": "Nearby landmarks include Ellis Island Immigration Museum, Statue of Liberty, and New York City skyline. The area is a popular tourist destination, featuring the Statue of Liberty and the New York City skyline in the background. The Ellis Island Immigration Museum is also located nearby, providing visitors with a comprehensive look at American immigration history."}
```

Στον κώδικα της εφαρμογής στην μέθοδο `onCreate()` χτίζεται το URL στην κατάλληλη μορφή και στην συνέχεια πραγματοποιείται το JSON request. Γίνεται ανάλυση της απάντησης και επιλέγονται οι πληροφορίες που χρειάζονται, που θα εκτυπωθούν στην οθόνη του κινητού ενημερώνοντας τον χρήστη.

```
// The query text is the name of the landmark detected in the previous activity
String q = getIntent().getStringExtra("name: "query");
// Convert the desired amount from one currency to another
String url = "https://serpapi.com/search.json?q="+q+"&api_key=674b7e80b3392656c7525ae3d6b230cf4b89a723e72dff2fc4f32784694d5b80";
// Make the request
OkHttpClient client = new OkHttpClient();
Request request = new Request.Builder()
    .url(url)
    .build();
```



```

client.newCall(request).enqueue(new Callback() {
    @Override
    public void onFailure(Call call, IOException e) {
        String mMessage = e.getMessage();
        //System.out.println("!!!!!!!!!!!!!!!!!!!!!! " + mMessage);
    }

    @Override
    public void onResponse(Call call, Response response) throws IOException {
        final String mMessage = response.body().string();
        SearchLandmark.this.runOnUiThread(new Runnable() {
            @Override
            public void run() {
                try {
                    // Get the result value from JSON
                    JSONObject obj = new JSONObject(mMessage);
                    JSONObject obj2 = obj.getJSONObject("knowledge_graph");
                    JSONObject obj3 = obj2.getJSONObject("local_map");
                    title.setText(obj2.getString( name: "title"));
                    type.setText(obj2.getString( name: "type"));
                    desc.setText(obj2.getString( name: "description"));
                    phone.setText(obj2.getString( name: "phone"));
                    String linkText = "Visit the <a href='"+obj2.getString( name: "website")+"'>website</a>.";
                    link.setText(Html.fromHtml(linkText));
                    link.setMovementMethod(LinkMovementMethod.getInstance());
                    String linkTextMaps = "Visit the <a href='"+obj3.getString( name: "link")+"'>maps</a>.";
                    maps.setText(Html.fromHtml(linkTextMaps));
                    maps.setMovementMethod(LinkMovementMethod.getInstance());
                    progressbar.setVisibility(View.INVISIBLE);
                    cardView.setVisibility(View.VISIBLE);
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        });
    }
}

```



7 Συμπεράσματα και μελλοντικές επεκτάσεις

Μετά το πέρας της πανδημίας του ιού Covid-19, έχει παρατηρηθεί μεγάλη αύξηση στον χώρο του τουρισμού τόσο στην χώρα μας, όσο και σε παγκόσμιο επίπεδο. Επομένως, ενδέχεται να υπάρξει αυξημένη ζήτηση, για μια εφαρμογή με στόχο την υποβοήθηση των τουριστών. Η εφαρμογή που αναπτύχθηκε στα πλαίσια αυτής της πτυχιακής εργασίας, είναι εύκολη και κατανοητή ως προς τη χρήση της, με ένα απλό και φιλικό προς το χρήστη γραφικό περιβάλλον. Σκοπός της είναι να εξυπηρετεί μερικές από τις βασικότερες ανάγκες ενός τουρίστα, ο οποίος βρίσκεται σε μια ξένη χώρα, όπως αυτές της μετάφρασης, της αναζήτησης πληροφορίας για αξιοθέατα και της μετατροπής συναλλάγματος.

Αρχικά, έγινε αναφορά σε παρόμοιες εφαρμογές που υπάρχουν ήδη στην αγορά, περιγράφοντας με λίγα λόγια την λειτουργία τους. Στη συνέχεια, παρουσιάστηκε η εκτέλεση της εφαρμογής και ο τρόπος λειτουργίας της, μέσω ενός User Manual. Τέλος, έγινε αναλυτική παρουσίαση στον τρόπο με τον οποίο υλοποιήθηκε, αναλύοντας όλες τις τεχνολογίες που χρησιμοποιήθηκαν.

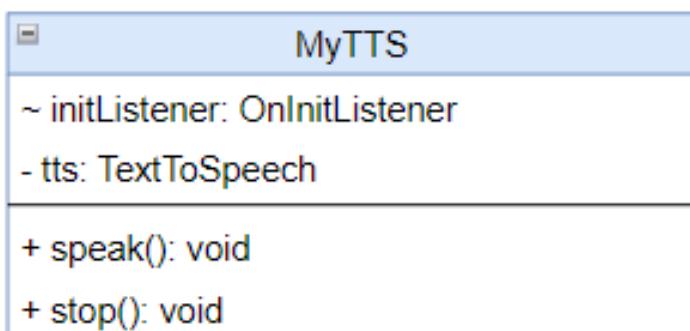
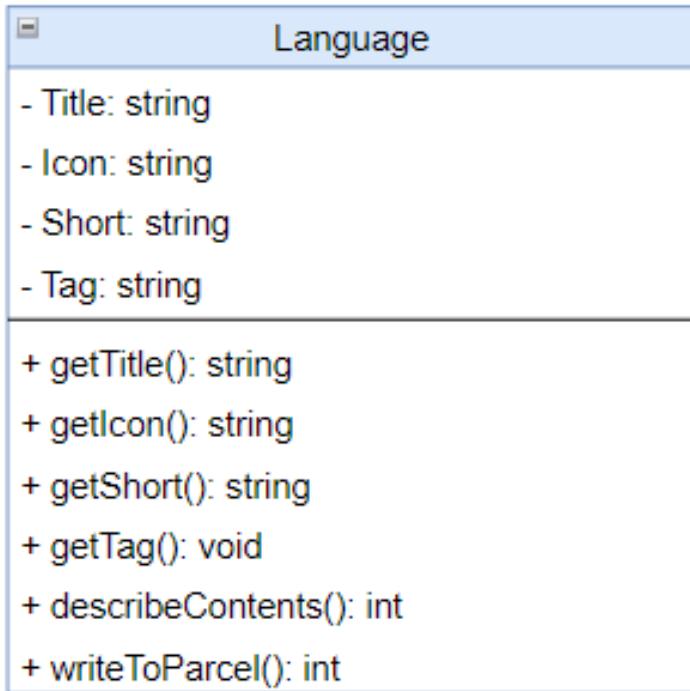
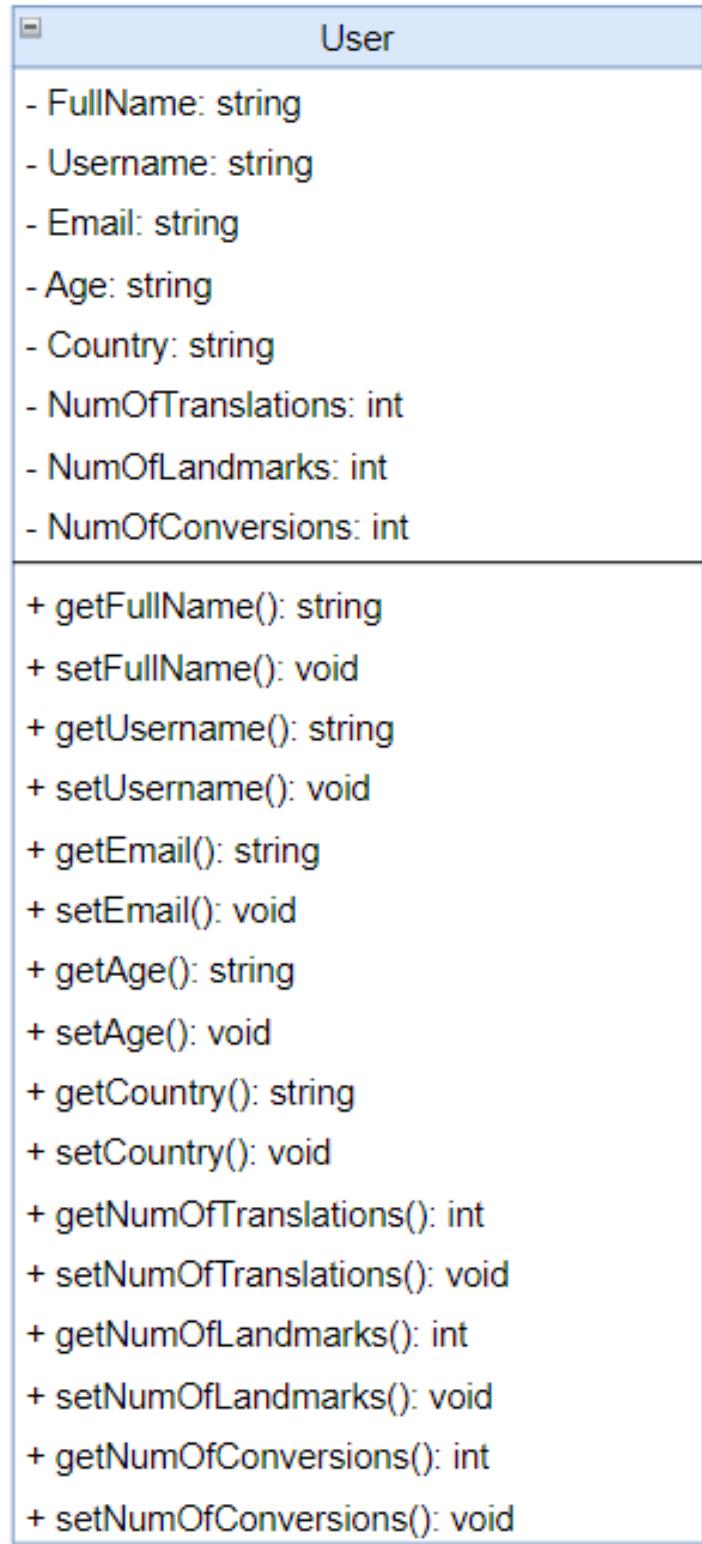
Σαφώς και υπάρχουν προοπτικές για μελλοντικές επεκτάσεις της εφαρμογής και εμπλουτισμό της με περισσότερες λειτουργίες και δυνατότητες. Για παράδειγμα, χρήσιμη θα ήταν η λειτουργία αποθήκευσης των πληροφοριών, αγαπημένων αξιοθέατων που έχει «σκανάρει» ο χρήστης. Έτσι, δεν θα χρειαζόταν να πραγματοποιήσει μια καινούργια αναζήτηση για να εντοπίσει ξανά τις πληροφορίες αυτές και θα μπορούσε να έχει πρόσβαση σε εκείνες μέσω του προφίλ του. Ακόμα, θα ήταν πολύ εξυπηρετική για έναν τουρίστα η λειτουργία πρότασης προσχεδιασμένων διαδρομών στις πόλεις (city tour), μέσω των οποίων θα μπορούσε να επισκεφτεί τα διασημότερα αξιοθέατά της. Συμπεραίνοντας, ως επέκταση της λειτουργίας μετατροπής συναλλάγματος, θα ήταν επιθυμητή και η εμφάνιση ενημερωτικών διαγραμμάτων στον χρήστη, σχετικά με την εξέλιξη των ισοτιμιών σε βάθος χρόνου.

Σε κάθε περίπτωση, όμως, μετά την επιτυχή ολοκλήρωση μελέτης του αντικειμένου και την κατασκευή της εφαρμογής, γίνεται αντιληπτή η χρησιμότητα και η αξία της ακόμα και στο βασικό επίπεδο που έχει υλοποιηθεί, που μπορεί φυσικά να εμπλουτιστεί με περαιτέρω λειτουργίες και βελτιώσεις.



8 Παραρτήματα

8.1 Διάγραμμα κλάσεων σε UML





8.2 Ανάπτυξη ιστοσελίδας για την εμφάνιση στατιστικών

Για την καλύτερη εμφάνιση των αποτελεσμάτων των στατιστικών που καταγράφονται από την εφαρμογή, δημιουργήθηκε μία απλή ιστοσελίδα. Αυτή συνδέεται μέσω JavaScript με την Firebase και φορτώνει όλα τα σχετικά δεδομένα, που είναι αποθηκευμένα στην βάση δεδομένων. Ο τρόπος με τον οποία πραγματοποιείται η σύνδεση φαίνεται στο παρακάτω screenshot.

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

Στην συνέχεια καλείται κατά την φόρτωση της σελίδας η συνάρτηση GetAllDataRealtime(), η οποία φορτώνει όλα τα δεδομένα από τα κατάλληλα reference στους αντίστοιχους πίνακες της ιστοσελίδας.

```
//----- 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);
    })

    const dbRef2 = ref(database, "Statistics/Translation");

    onValue(dbRef2,(snapshot)=>{
        var translations = [];

        snapshot.forEach(childSnapshot => {
            translations.push(childSnapshot.val());
        });

        AddAllItemsToTableTranslations(translations);
    })

    const dbRef3 = ref(database, "Statistics/Landmark");

    onValue(dbRef3,(snapshot)=>{
        var landmarks = [];

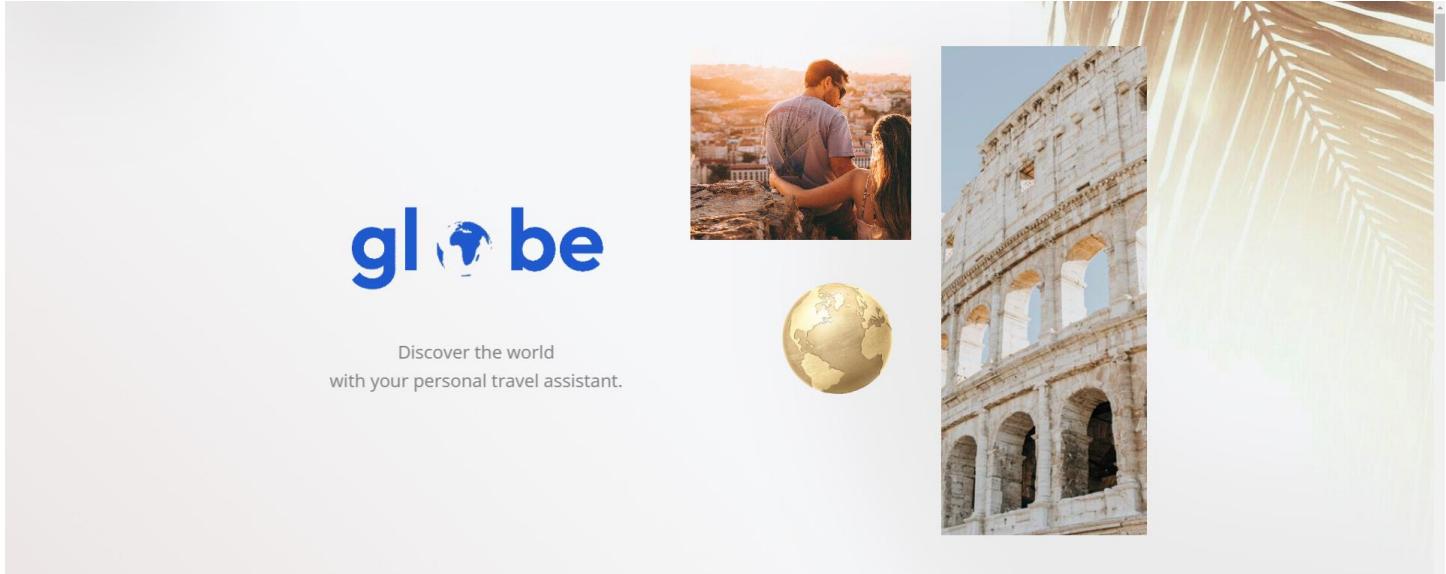
        snapshot.forEach(childSnapshot => {
            landmarks.push(childSnapshot.val());
        });

        AddAllItemsToTableLandmarks(landmarks);
    })
}

window.onload = GetAllDataRealtime;
```



Ακολουθούν μερικές εικόνες από την εκτέλεση της ιστοσελίδας, παρουσιάζοντας το user interface και τον τρόπο με τον οποίο απεικονίζονται οι πληροφορίες των στατιστικών.



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 | -0.1246260000000001 | 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, ₦) | 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, ₽) | 1.0 | 59.79 | 22/08/2022 07:27:00 |
| stacy | EUR (Euro Member Countries, €) | AED (United Arab Emirates Dirham, ₦) | 100.0 | 365.38 | 24/08/2022 04:57:02 |
| nancy | EUR (Euro Member Countries, €) | USD (United States Dollar, \$) | 10.0 | 9.96 | 24/08/2022 08:21:36 |
| andreas | EUR (Euro Member Countries, €) | AED (United Arab Emirates Dirham, ₦) | 1.0 | 3.66 | 24/08/2022 08:42:52 |
| stacy | EUR (Euro Member Countries, €) | USD (United States Dollar, \$) | 100.0 | 100.02 | 26/08/2022 12:58:24 |
| stacy | EUR (Euro Member Countries, €) | RON (Romania Leu, lei) | 100.0 | 481.42 | 04/09/2022 07:36:09 |





Translation Statistics

For every translation we collect specific data, such as the original and translated text, the initial and final language, current user's username and a timestamp.



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 | μαμά | μαμά | 24/08/2022 08:13:25 |
| nancy | el | ja | μαμά | μαμά | 24/08/2022 08:13:25 |
| nancy | el | ja | μαμά | μαμά | 24/08/2022 08:14:05 |
| nancy | el | zh | μαμά | μαμά | 24/08/2022 08:14:35 |
| nancy | el | zh | μαμά | μαμά | 24/08/2022 08:14:59 |
| nancy | el | en | μαμά | μαμά | 24/08/2022 08:15:11 |
| nancy | el | en | μαμά | μαμά | 24/08/2022 08:15:17 |





9 Βιβλιογραφικές Πηγές

1. <https://developer.android.com/develop/ui/views/animations/transitions/start-activity>
2. <https://firebase.google.com/docs/auth/android/google-signin>
3. <https://developers.google.com/identity/sign-in/android/people>
4. [https://developers.google.com/ml-kit/language/identification](https://developers.google.com/ml-kit/language/translation)
5. <https://developers.google.com/ml-kit/vision/text-recognition>
6. <https://firebase.google.com/docs/ml/recognize-text>
7. <https://firebase.google.com/docs/ml/recognize-landmarks>
8. <https://cloud.google.com/vision/docs/detecting-landmarks>
10. <https://firebase.google.com/docs/ml-kit>
11. <https://firebase.google.com/docs/ml>
12. <https://developers.google.com/ml-kit>
13. <https://developer.android.com/reference/android/speech/tts/TextToSpeech>
14. <https://developer.android.com/reference/android/speech/SpeechRecognizer>
15. <https://www.geeksforgeeks.org/how-to-convert-speech-to-text-in-android/>
16. <https://www.geeksforgeeks.org/how-to-convert-text-to-speech-in-android/>
17. <https://cloud.google.com/architecture/speech-translation-android-microservice>
18. [https://developers.google.com/android/reference/com/google/mlkit/vision/common/InputImage#fromFilePath\(android.content.Context,%20android.net.Uri\)](https://developers.google.com/android/reference/com/google/mlkit/vision/common/InputImage#fromFilePath(android.content.Context,%20android.net.Uri))
19. <https://developers.google.com/ml-kit/vision/text-recognition/v2/android>
20. <https://medium.com/@kanchanaj/how-to-create-a-simple-currency-converter-app-in-android-studio-with-the-use-of-api-38d07e8253fc>
21. <https://www.codespeedy.com/create-real-time-currency-converter-app-in-android-studio/>
22. <https://exchangeratesapi.io/documentation/>
23. https://apilayer.com/marketplace/exchangerates_data-api
24. <https://serpapi.com/>
25. <https://www.flaticon.com/packs/country-flags>
26. <https://www.eurochange.co.uk/travel-money/world-currency-abbreviations-symbols-and-codes-travel-money>
27. https://www.loc.gov/standards/iso639-2/php/English_list.php
28. <https://support.google.com/knowledgepanel/answer/9787176?hl=en>
29. <https://googleapis.github.io/google-http-java-client/json.html>
30. <https://stackoverflow.com/questions/2201917/how-can-i-open-a-url-in-androids-web-browser-from-my-application>
31. <https://stackoverflow.com/questions/21253303/exit-android-app-on-back-pressed>