



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Help4All - Mobile Εφαρμογή Αλληλοβοήθειας

**Χρήστος Γ. Χατζηπέτρος
Επιβλέποντες: Αλέξης Δελής, Καθηγητής**

**ΑΘΗΝΑ
Αύγουστος 2021**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Heip4All - Mobile Εφαρμογή Αλληλοβοήθειας

Χρήστος Γ. Χατζηπέτρος

A.M: 1115201700190

ΕΠΙΒΛΕΠΟΝΤΕΣ: Αλέξης Δελής, Καθηγητής

ΠΕΡΙΛΗΨΗ

Τη σημερινή εποχή στην Ελλάδα, οι πόροι που διαθέτει το κράτος για άτομα με δυσκολίες αυτοεξυπηρέτησης είναι ελάχιστοι. Ως εκ τούτου, βασικές ανάγκες όπως η πρόσβαση σε ιατροφαρμακευτικές εγκαταστάσεις ή και σε καταστήματα αγοράς αγαθών πρώτης ανάγκης πραγματοποιούνται δύσκολα χωρίς την βοήθεια από κάποιο άλλο άτομο. Αυτή η κατάσταση επιδεινώθηκε ακόμα περισσότερο εξαιτίας της πανδημίας COVID-19 και των περιοριστικών μέτρων που λήφθηκαν για τον έλεγχο της διασποράς της. Ως λύση αντιμετώπισης του αναφερθέντος προβλήματος, προτείνεται σε αυτή την πτυχιακή εργασία η χρήση της εφαρμογής "Help4All" . Πρόκειται για μια mobile εφαρμογή, που επιτρέπει στους χρήστες της τόσο να αναζητήσουν βοήθεια για την εξυπηρέτησή τους, όσο και να την προσφέρουν ως εθελοντές. Είναι ουσιαστικά, ένας τρόπος διασύνδεσης των χρηστών ώστε να προαχθούν ο εθελοντισμός και η αλληλοβοήθεια σαν πηγές αντιμετώπισης των σύγχρονων κοινωνικών αναγκών. Η εφαρμογή χτίστηκε στο περιβάλλον του Android Studio χρησιμοποιώντας την γλώσσα προγραμματισμού Kotlin που αποτελεί βελτιστοποιημένη έκδοση της Java για κινητά τηλέφωνα. Για την αποθήκευση λογαριασμών χρηστών, δεδομένων για αιτήματα βοήθειας αλλά και συνομιλιών μεταξύ χρηστών χρησιμοποιείται ένας συνδυασμός από: php server,mysql server και Firebase.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Ανάπτυξη Εφαρμογών για Κινητά Τηλέφωνα

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: android,php,mysql,kotlin,API,client,server

ABSTRACT

In Greece, the available resources to individuals requiring medical home assistance remain insufficient. Therefore, taking care of vital needs such as having access to medical care and basic necessities can seem almost impossible without other people's aid. This situation has worsened due to the COVID-19 pandemic and the strict measures that have been implemented to prevent its spread. In this thesis, "Help4All" is proposed as an effective way of helping us face the above-mentioned challenge. It is a mobile application that allows its users not only to seek help for their various needs but to also provide their support to people in need. Basically, "Help4All" acts as a medium of connecting users to promote volunteerism and mutual aid as means of solving contemporary social issues.

Help4All was built on Android Studio using the Kotlin programming language. Kotlin is an optimized version of Java, aimed at mobile devices. In order to keep track of user accounts, help requests, and chatting messages, a combination of php,mysql, and Firebase servers was used.

SUBJECT AREA: Mobile Application Development

KEYWORDS: android,php,mysql,kotlin,API,client,server

*Η παρούσα πτυχιακή είναι αφιερωμένη στην οικογένειά μου,
στους γονείς μου Γιώργο και Παρασκευή.*

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον καθ. Αλέξη Δελή για την υπομονή, τη συνεργασία και την πολύτιμη καθοδήγησή του καθ' όλη την πορεία δημιουργίας της παρούσας Πτυχιακής Εργασίας.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ	7
1 ΠΡΟΛΟΓΟΣ	8
2 ΔΟΜΗ ΕΦΑΡΜΟΓΗΣ	9
2.1 Μοντέλο Λειτουργίας	9
2.2 Backend Υποδομές	10
2.3 Frontend Υποδομές	10
2.4 Τεχνικές Προδιαγραφές	13
2.4.1 Αρχεία JSON [1]	13
2.4.2 Αιτήματα HTTP(S)	13
2.4.2.1 Το παραδοσιακό HTTP	14
2.4.2.2 Η καινοτομία του HTTPS	16
2.4.2.3 HTTP και Android	16
2.4.2.4 Αυθυπόγραφα Πιστοποιητικά SSL	18
2.5 Παραδείγματα Χρήσης	22
3 ΚΥΚΛΟΣ ΛΕΙΤΟΥΡΓΙΑΣ	24
3.1 Είσοδος στο πρόγραμμα	24
3.2 Παροχή βοήθειας	24
3.3 Αίτηση βοήθειας	27
4 ΤΕΧΝΙΚΕΣ ΑΝΑΠΤΥΞΗΣ	28
4.1 Η γλώσσα προγραμματισμού Kotlin	28
4.1.1 Χειρισμός Null Pointer Exceptions	28
4.1.2 Εύκολη δημιουργία κλάσεων	29
4.2 Android Studio	30
4.3 Υποστήριξη μεταφράσεων	31
4.4 Πρότυπα λειτουργίας Android	35
4.5 Μηχανισμοί ασφαλείας	35
4.5.1 Είσοδος με Firebase	35
4.5.2 Token authentication	35
4.5.3 Προστασία από SQL Injections	36
4.5.3.1 Ο μηχανισμός	36
4.5.3.2 Η υλοποίηση	37
4.5.4 Δικαιώματα χρήστη MySQL	37
5 ΕΠΙΛΟΓΟΣ	39
5.1 Συμπεράσματα	39
5.2 Μελλοντικές ιδέες	39
ΣΥΝΤΜΗΣΕΙΣ - ΑΡΚΤΙΚΟΛΕΞΑ - ΑΚΡΩΝΥΜΙΑ	40
ΑΝΑΦΟΡΕΣ	41

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

1	Επικοινωνία μεταξύ συστημάτων Help4all	9
2	Παράδειγμα δραστηριότητας της εφαρμογής για σύνδεση/εγγραφή χρήστη .	10
3	Παράδειγμα δραστηριότητας της εφαρμογής για συνομιλία μεταξύ χρηστών .	11
4	Αρχείο LoginActivity.kt	11
5	Αρχείο activitylogin.xml	12
6	Μορφή JSON αρχείου	13
7	Δημιουργία πιστοποιητικού	18
8	Περιεχόμενο .crt αρχείου	19
9	Περιεχόμενο .key αρχείου	20
10	Εισαγωγή Πιστοποιητικού	21
11	Αρχική σελίδα για συνδεδεμένο χρήστη	22
12	Συνομιλία με το χρήστη που αποδέχθηκε το αίτημα	23
13	Κώδικας εισόδου/εγγραφής χρήστη	24
14	Κεντρική Οθόνη	25
15	Λίστα αιτημάτων εν ενεργεία	25
16	Λίστα αιτημάτων στην εφαρμογή	26
17	Δείγμα καταλόγων συνομιλίας	26
18	Βασική Φόρμα Αίτησης Βοήθειας	27
19	JSON αρχείο Αίτησης Βοήθειας	27
20	Android Studio	30
21	Πρώτη βασική οθόνη στα Ελληνικά	31
22	Δεύτερη βασική οθόνη στα Ελληνικά	32
23	Translations Editor	32
24	strings.xml για την Αγγλική Γλώσσα	33
25	strings.xml για την Ελληνική Γλώσσα	33
26	Ανάθεση πολύγλωσσου κειμένου	34
27	PHP server - Μέθοδος εισαγωγής Αιτήματος Βοήθειας	37
28	phrmyadmin - Πίνακας χρηστών ΒΔ	38

1 ΠΡΟΛΟΓΟΣ

Στη σημερινή εποχή, το διαδίκτυο και τα έξυπνα κινητά τηλέφωνα χρησιμοποιούνται από ένα μεγάλο μέρος του ενήλικου πληθυσμού, καθώς έχουν γίνει αναπόσπαστο κομμάτι της καθημερινότητάς μας. Αυτό σημαίνει πως από εδώ και στο εξής, δημιουργούνται γενιές ανθρώπων που, σε αντίθεση με παλαιότερα, θα μπορούν να χρησιμοποιήσουν mobile εφαρμογές ανεξαρτήτως ηλικίας.

Με βάση την παραπάνω διαπίστωση, γίνεται αντιληπτό πως υπάρχει πλέον, και θα υπάρχει για τα επόμενα χρόνια, μια μερίδα ανθρώπων που χρειάζονται εφαρμογές οι οποίες να προσαρμόζονται στις δικές τους, ξεχωριστές ανάγκες. Το Help4All σκοπεύει να γίνει μια απ' αυτές, προσφέροντας σε κάθε πολίτη που μπορεί να αντιμετωπίσει δυσκολίες αυτοεξυπηρέτησης την βοήθεια που χρειάζεται μέσω του εθελοντισμού.

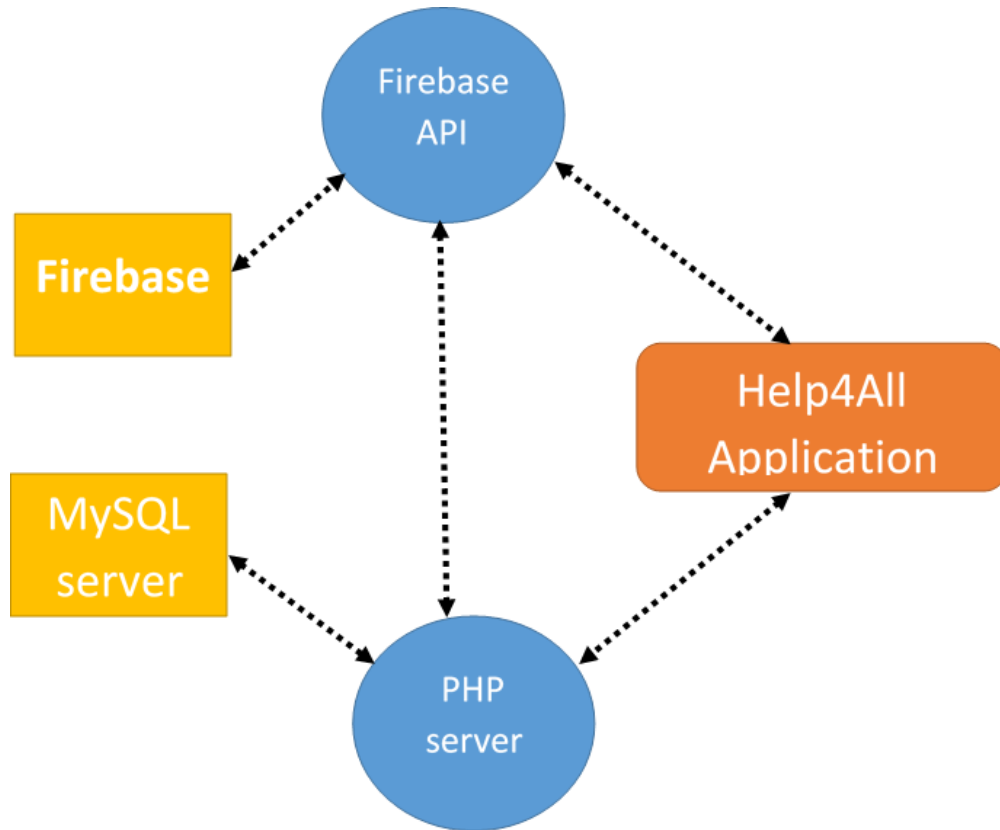
Πιο συγκεκριμένα, στοχεύοντας σε δύο κατηγορίες πολιτών, αυτούς που μπορούν να προσφέρουν τη βοήθειά τους σε τρίτους (εθελοντές-αρωγοί) και κατ' επέκταση αυτούς που την έχουν ανάγκη, η εφαρμογή που παρουσιάζεται διαθέτει στις δύο αυτές ομάδες χρηστών ένα ασφαλές περιβάλλον αλληλεπίδρασης και επικοινωνίας. Επιτρέπει ουσιαστικά σε κάθε χρήστη σε ανάγκη να καταχωρεί αιτήματα αναζήτησης βοήθειας τα οποία αποθηκεύονται και μπορούν να διαβαστούν από τους εκάστοτε εθελοντές. Σε κάθε εθελοντή που μπορεί να συνεισφέρει στην υλοποίηση κάποιου αιτήματος δίνεται η δυνατότητα αποδοχής του και ηλεκτρονικής συνομιλίας με τον αρχικό συντάκτη του, κάνοντας έτσι την βοήθεια που παρέχεται άμεση και αποτελεσματική.

Αξίζει να τονιστεί πως όσοι χρειαστούν τις προσφερόμενες υπηρεσίες, δεν περιορίζονται σε άτομα τρίτης ηλικίας, αν κι αυτά ίσως αποτελέσουν την πλειονότητα των χρηστών του Help4All. Από άτομα με ειδικές ανάγκες, μέχρι ανθρώπους που λόγω μιας προσωρινής κατάστασης (όπως η πανδημία COVID-19 των τελευταίων ετών), δεν είναι σε θέση να αντεπεξέλθουν σε κάποια υποχρέωσή τους, μπορούμε μέσω του Help4All να προσφέρουμε σε όλους.

Τέλος, μιας που πρωταρχικός στόχος της εφαρμογής είναι η προσφορά αρωγής σε όσο το δυνατόν περισσότερους ανθρώπους, επιλέχθηκε η πλατφόρμα του Android για την κατασκευή της, διότι αποτελεί το κυρίαρχο στην αγορά λειτουργικό σύστημα για κινητές συσκευές.

2 ΔΟΜΗ ΕΦΑΡΜΟΓΗΣ

Το Help4All αποτελεί μια εφαρμογή που βασίζεται στην αλληλεπίδραση ποικίλων μηχανισμών για την απρόσκοπτη λειτουργία της. Με βάση τις αρχές που διέπουν την σύγχρονη Επιστήμη Υπολογιστών, ακολουθεί μια ανάλυση των διάφορων υπολειτουργιών της, οι οποίες μπορούν να περιγραφούν κι από το παρακάτω σχήμα:



Σχήμα 1: Επικοινωνία μεταξύ συστημάτων Help4all

2.1 Μοντέλο Λειτουργίας

Ξεκινώντας από ένα υψηλό επίπεδο αφαίρεσης, το βασικό μοντέλο λειτουργίας περιλαμβάνει:

- Backend υλοποιήσεις
- Frontend υλοποιήσεις
- Ενδιάμεσα API

Πρακτικά, το πρόγραμμα επικοινωνεί με την Backend υποδομή του μέσω κατάλληλων αιτημάτων και λαμβάνει απαντήσεις που μοιράζεται με τον χρήστη. Αυτές οι πληροφορίες κατά βάση αφορούν:

- Αιτήματα Σύνδεσης
- Κείμενα Συνομιλίας
- Αιτήματα Αλληλοβοήθειας και περιεχόμενες πληροφορίες

Για να αναλυθεί περαιτέρω η αναφερθείσα λειτουργικότητα, θα πρέπει να παρουσιαστεί κάθε ενότητα του αντίστοιχου σχήματος ξεχωριστά.

2.2 Backend Υποδομές

Για την αποτελεσματική υλοποίηση του ((πίσω μέρους)) μιας εφαρμογής συνήθως χρησιμοποιείται μια υλοποίηση Βάσης Δεδομένων (ΒΔ). Η ιδιαιτερότητα του Help4All είναι η παράλληλη χρήση δύο μορφών ΒΔ. Συγκεκριμένα χρησιμοποιούνται:

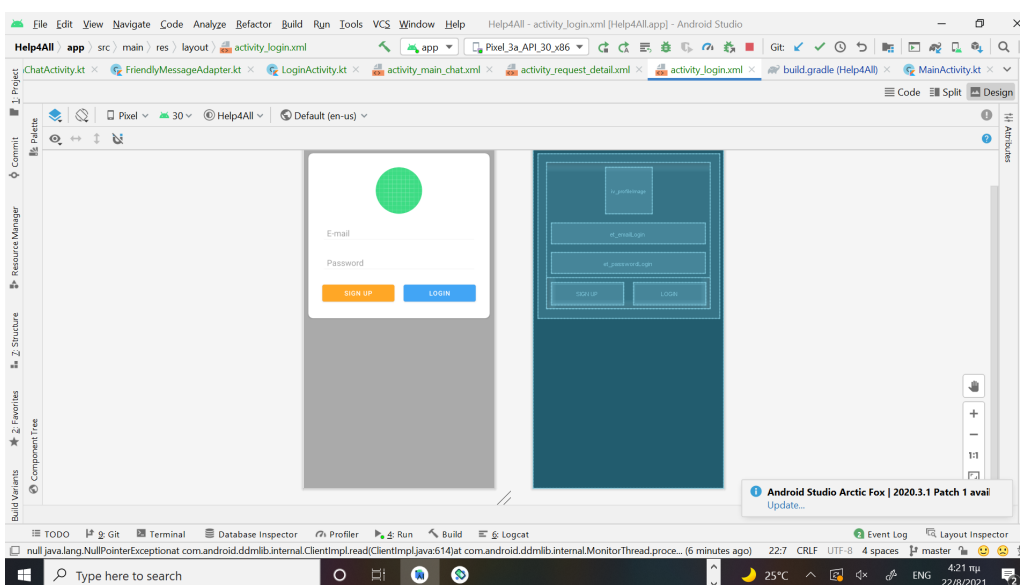
- Firebase
- MySQL server

Η χρήση αυτή παρέχει πλεονεκτήματα αλλά και δυσκολίες. Συνοπτικά, η χρήση του Firebase επιτρέπει την προσπέλαση λειτουργιών της Google για εφαρμογές κινητών τηλεφώνων, ενώ ένας MySQL server παρέχει την κλασική γρήγορη και αξιόπιστη απόδοση ενός συστήματος SQL ΒΔ.

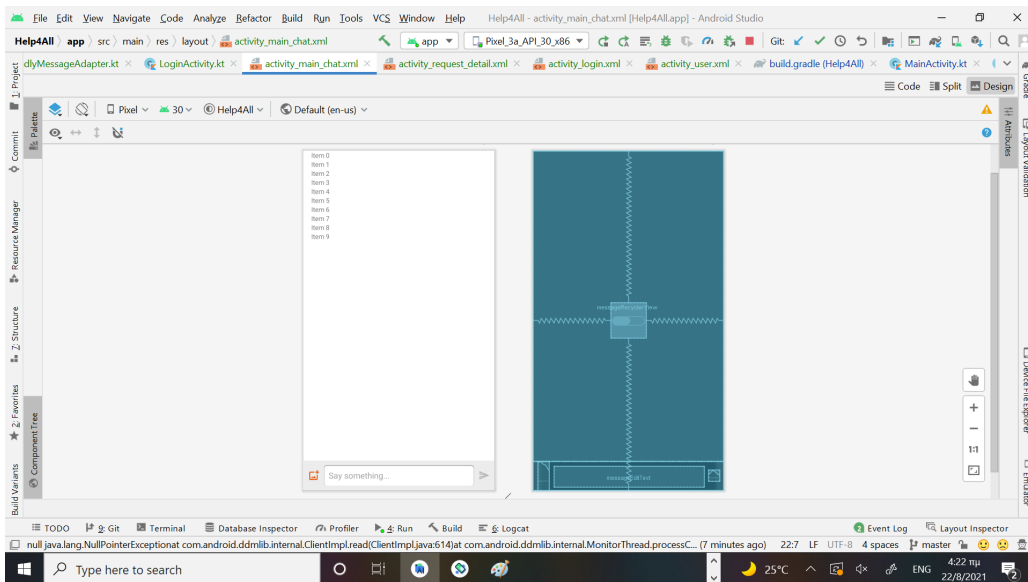
Αξίζει να σημειωθεί πως η εφαρμογή **δεν** επικοινωνεί απευθείας με τα συστήματα ΒΔ, καθώς αυτό θα ερχόταν σε αντίθεση με τις σύγχρονες πρακτικές σχεδιασμού εφαρμογών που αλληλεπιδρούν διαδικτυακά. Μεσολαβεί επομένως ένας PHP server που αναλαμβάνει την γεφύρωση των οντοτήτων εφαρμογής και ΒΔ.

2.3 Frontend Υποδομές

Η πλευρά του προγράμματος με την οποία έρχεται σε επαφή ο χρήστης, βασίζεται στην λογική υλοποίησης του **Activity** στα συστήματα Android. Πρόκειται ουσιαστικά για μια **κλάση** που στοχεύει να συνδέσει τις οθόνες που παρουσιάζονται στο χρήστη, κατά το όνομα της, με αντίστοιχες δραστηριότητες. [2] Ουσιαστικά, κάθε αντικείμενό της, παρουσιάζει ένα παράθυρο στο κινητό τηλέφωνο του χρήστη με σκοπό να τον καθοδηγήσει προς μια συγκεκριμένη ενέργεια. Η χρήση της για την ανάπτυξη μιας εφαρμογής Android εγγυάται την σωστή λειτουργία της εφαρμογής σε κάθε είδους κινητό τηλέφωνο/ταμπλέτα ή άλλη Android συσκευή. Κάτι τέτοιο είναι ιδιαίτερα σημαντικό για το Help4All καθώς οι ομάδες χρηστών στις οποίες στοχεύει θα διαθέτουν μια μεγάλη ποικιλομορφία κινητών συσκευών, με διαφορετικά χαρακτηριστικά. Ενδεικτικά ακολουθούν κάποιες περιπτώσεις παραθύρων που αξιοποιούνται στην πορεία αλληλεπίδρασης με την εφαρμογή:



Σχήμα 2: Παράδειγμα δραστηριότητας της εφαρμογής για σύνδεση/εγγραφή χρήστη



Σχήμα 3: Παράδειγμα δραστηριότητας της εφαρμογής για συνομιλία μεταξύ χρηστών

Κάθε τέτοια σελίδα αποτελείται από ένα `.xml` κι ένα `.kt` αρχείο που περιέχουν τον κώδικα για την μορφοποίηση αλλά και τη λειτουργία της αντίστοιχα. Για παράδειγμα, για την δραστηριότητα **εισόδου** στην εφαρμογή, τα αρχεία είναι τα εξής:

```

1 package com.example.help4all
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5
6 class LoginActivity : AppCompatActivity() {
7     private lateinit var mAuth: FirebaseAuth
8     private var isFirstImage = true
9     private var imageUrl: Uri? = null
10
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         setContentView(R.layout.activity_login)
14         val loginbutton: Button = findViewById<View>(R.id.btn_login) as Button
15         val registerbutton: Button = findViewById<View>(R.id.btn_register) as Button
16         val profileimage: CircleImageView = findViewById<View>(R.id.iv_profileImage) as CircleImageView
17         mAuth = FirebaseAuth.getInstance()
18         monitorClicks(loginbutton, registerbutton, profileimage)
19     }
20
21     private fun monitorClicks(
22         loginbutton: Button,
23         registerbutton: Button,
24         profileimage: CircleImageView
25     ) {
26         //set listeners for the two buttons
27         loginbutton.setOnClickListener {
28             isFirstImage = false
29             CreateOrLogin()
30         }
31         registerbutton.setOnClickListener {
32             isFirstImage = true
33             CreateOrLogin()
34         }
35         //now set a listener for the profile image |
36         profileimage.setOnClickListener {
37             pickImage()
38         }
39     }
40
41     private fun CreateOrLogin() {
42         //get the email and password from the fields
43     }
44 }

```

Σχήμα 4: Αρχείο LoginActivity.kt

Στο `.kt` αρχείο φαίνεται ο κώδικας που εκτελεί τις λειτουργίες της οθόνης LoginActivity. Ουσιαστικά, κατά τη δημιουργία του παραθύρου, στη μέθοδο **onCreate** ανατίθενται σε μεταβλητές τα κουμπιά σύνδεσης, εγγραφής και το πλαίσιο της εικόνας προφίλ, ώστε στη συνέχεια να προσδιοριστούν οι ενέργειες που θα γίνουν κατά το πάτημά τους. Πρακτικά, τα δύο κουμπιά χρησιμοποιούν τη συνάρτηση `CreateOrLogin()` ενώ το πλαίσιο εικόνας την `pickImage()`.

```

1 |<?xml version="1.0" encoding="utf-8"?>
2 |<-RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3 |   xmlns:app="http://schemas.android.com/apk/res-auto"
4 |   xmlns:tools="http://schemas.android.com/tools"
5 |   android:layout_width="match_parent"
6 |   android:layout_height="match_parent"
7 |   tools:context=".MainActivity"
8 |   android:background="@android:color/darker_gray">
9 |
10 |
11 |
12 |
13 |
14 |
15 |
16 |
17 |
18 |
19 |
20 |
21 |
22 |
23 |
24 |
25 |
26 |
27 |
28 |
29 |
30 |
31 |
32 |
33 |
34 |
35 |
36 |
37 |
38 |
39 |
40 |
41 |
42 |
43 |
44 |
45 |
46 |
47 |
48 |
49 |
50 |
51 |
52 |
53 |
54 |
55 |
56 |
57 |
58 |
59 |
60 |
61 |
62 |
63 |
64 |
65 |
66 |
67 |
68 |
69 |
70 |
71 |
72 |
73 |
74 |
75 |
76 |
77 |
78 |
79 |
80 |
81 |
82 |
83 |
84 |
85 |
86 |
87 |
88 |
89 |
90 |
91 |
92 |
93 |
94 |
95 |
96 |
97 |
98 |
99 |
100|
101|
102|
103|
104|
105|
106|
107|
108|
109|
110|
111|
112|
113|
114|
115|
116|
117|
118|
119|
120|
121|
122|
123|
124|
125|
126|
127|
128|
129|
130|
131|
132|
133|
134|
135|
136|
137|
138|
139|
140|
141|
142|
143|
144|
145|
146|
147|
148|
149|
150|
151|
152|
153|
154|
155|
156|
157|
158|
159|
160|
161|
162|
163|
164|
165|
166|
167|
168|
169|
170|
171|
172|
173|
174|
175|
176|
177|
178|
179|
180|
181|
182|
183|
184|
185|
186|
187|
188|
189|
190|
191|
192|
193|
194|
195|
196|
197|
198|
199|
200|
201|
202|
203|
204|
205|
206|
207|
208|
209|
210|
211|
212|
213|
214|
215|
216|
217|
218|
219|
220|
221|
222|
223|
224|
225|
226|
227|
228|
229|
230|
231|
232|
233|
234|
235|
236|
237|
238|
239|
240|
241|
242|
243|
244|
245|
246|
247|
248|
249|
250|
251|
252|
253|
254|
255|
256|
257|
258|
259|
260|
261|
262|
263|
264|
265|
266|
267|
268|
269|
270|
271|
272|
273|
274|
275|
276|
277|
278|
279|
280|
281|
282|
283|
284|
285|
286|
287|
288|
289|
290|
291|
292|
293|
294|
295|
296|
297|
298|
299|
300|
301|
302|
303|
304|
305|
306|
307|
308|
309|
310|
311|
312|
313|
314|
315|
316|
317|
318|
319|
320|
321|
322|
323|
324|
325|
326|
327|
328|
329|
330|
331|
332|
333|
334|
335|
336|
337|
338|
339|
340|
341|
342|
343|
344|
345|
346|
347|
348|
349|
350|
351|
352|
353|
354|
355|
356|
357|
358|
359|
360|
361|
362|
363|
364|
365|
366|
367|
368|
369|
370|
371|
372|
373|
374|
375|
376|
377|
378|
379|
380|
381|
382|
383|
384|
385|
386|
387|
388|
389|
390|
391|
392|
393|
394|
395|
396|
397|
398|
399|
400|
401|
402|
403|
404|
405|
406|
407|
408|
409|
410|
411|
412|
413|
414|
415|
416|
417|
418|
419|
420|
421|
422|
423|
424|
425|
426|
427|
428|
429|
430|
431|
432|
433|
434|
435|
436|
437|
438|
439|
440|
441|
442|
443|
444|
445|
446|
447|
448|
449|
450|
451|
452|
453|
454|
455|
456|
457|
458|
459|
460|
461|
462|
463|
464|
465|
466|
467|
468|
469|
470|
471|
472|
473|
474|
475|
476|
477|
478|
479|
480|
481|
482|
483|
484|
485|
486|
487|
488|
489|
490|
491|
492|
493|
494|
495|
496|
497|
498|
499|
500|
501|
502|
503|
504|
505|
506|
507|
508|
509|
510|
511|
512|
513|
514|
515|
516|
517|
518|
519|
520|
521|
522|
523|
524|
525|
526|
527|
528|
529|
530|
531|
532|
533|
534|
535|
536|
537|
538|
539|
540|
541|
542|
543|
544|
545|
546|
547|
548|
549|
550|
551|
552|
553|
554|
555|
556|
557|
558|
559|
560|
561|
562|
563|
564|
565|
566|
567|
568|
569|
570|
571|
572|
573|
574|
575|
576|
577|
578|
579|
580|
581|
582|
583|
584|
585|
586|
587|
588|
589|
590|
591|
592|
593|
594|
595|
596|
597|
598|
599|
600|
601|
602|
603|
604|
605|
606|
607|
608|
609|
610|
611|
612|
613|
614|
615|
616|
617|
618|
619|
620|
621|
622|
623|
624|
625|
626|
627|
628|
629|
630|
631|
632|
633|
634|
635|
636|
637|
638|
639|
640|
641|
642|
643|
644|
645|
646|
647|
648|
649|
650|
651|
652|
653|
654|
655|
656|
657|
658|
659|
660|
661|
662|
663|
664|
665|
666|
667|
668|
669|
670|
671|
672|
673|
674|
675|
676|
677|
678|
679|
680|
681|
682|
683|
684|
685|
686|
687|
688|
689|
690|
691|
692|
693|
694|
695|
696|
697|
698|
699|
700|
701|
702|
703|
704|
705|
706|
707|
708|
709|
710|
711|
712|
713|
714|
715|
716|
717|
718|
719|
720|
721|
722|
723|
724|
725|
726|
727|
728|
729|
730|
731|
732|
733|
734|
735|
736|
737|
738|
739|
740|
741|
742|
743|
744|
745|
746|
747|
748|
749|
750|
751|
752|
753|
754|
755|
756|
757|
758|
759|
760|
761|
762|
763|
764|
765|
766|
767|
768|
769|
770|
771|
772|
773|
774|
775|
776|
777|
778|
779|
780|
781|
782|
783|
784|
785|
786|
787|
788|
789|
790|
791|
792|
793|
794|
795|
796|
797|
798|
799|
800|
801|
802|
803|
804|
805|
806|
807|
808|
809|
810|
811|
812|
813|
814|
815|
816|
817|
818|
819|
820|
821|
822|
823|
824|
825|
826|
827|
828|
829|
830|
831|
832|
833|
834|
835|
836|
837|
838|
839|
840|
841|
842|
843|
844|
845|
846|
847|
848|
849|
850|
851|
852|
853|
854|
855|
856|
857|
858|
859|
860|
861|
862|
863|
864|
865|
866|
867|
868|
869|
870|
871|
872|
873|
874|
875|
876|
877|
878|
879|
880|
881|
882|
883|
884|
885|
886|
887|
888|
889|
890|
891|
892|
893|
894|
895|
896|
897|
898|
899|
900|
901|
902|
903|
904|
905|
906|
907|
908|
909|
910|
911|
912|
913|
914|
915|
916|
917|
918|
919|
920|
921|
922|
923|
924|
925|
926|
927|
928|
929|
930|
931|
932|
933|
934|
935|
936|
937|
938|
939|
940|
941|
942|
943|
944|
945|
946|
947|
948|
949|
950|
951|
952|
953|
954|
955|
956|
957|
958|
959|
960|
961|
962|
963|
964|
965|
966|
967|
968|
969|
970|
971|
972|
973|
974|
975|
976|
977|
978|
979|
980|
981|
982|
983|
984|
985|
986|
987|
988|
989|
990|
991|
992|
993|
994|
995|
996|
997|
998|
999|
1000|

```

Σχήμα 5: Αρχείο activitylogin.xml

Στο αντίστοιχο .xml αρχείο, βρίσκεται η εμφανισιακή δομή του ίδιου παραθύρου. Κάθε στοιχείο που βρίσκεται μέσα στις αντίστοιχες xml ετικέτες συνεισφέρει στην τελική εμφάνιση της εφαρμογής. Συγκεκριμένα, με την δομή CardView δημιουργείται μια περιοχή σχήματος κάρτας στην εφαρμογή η οποία περιέχει κουμπιά (Button), κυκλικές εικόνες (CircleImageView) και πεδία κειμένου (EditText). Όλα αυτά, ευθυγραμμίζονται στην οθόνη μιας κινητής συσκευής χάρη στην οντότητα LinearLayout.

2.4 Τεχνικές Προδιαγραφές

Έχοντας υπόψη τον σκελετό επικοινωνίας του Σχήματος 1, μπορεί πλέον κανείς να κατανοήσει τις τεχνικές λεπτομέρειες που σχετίζονται με την αλληλεπίδραση των προαναφερθέντων οντοτήτων. Για να γίνει αυτό εύκολο, θα παρουσιαστεί όλη η πορεία της επικοινωνίας τους, κατά τον κύκλο ζωής της εφαρμογής. Ωστόσο, προκειμένου αυτό να είναι εφικτό, θα αναλυθούν πρώτα, κάποιες βασικές μέθοδοι αλληλεπίδρασης προγραμμάτων.

2.4.1 Αρχεία JSON [1]

Τα αρχεία JSON αποτελούν χαρακτηριστικό μέσο ανταλλαγής μηνυμάτων διαδικτυακών εφαρμογών. Η χρήση τους ξεκίνησε από μια απ' τις πιο διαδεδομένες γλώσσες για Web-based εφαρμογές, τη Javascript κι επεκτάθηκε σχεδόν στο σύνολο του παγκόσμιου ιστού. Η ευκολία χρήσης τους έγκειται στην δομή τους, που εγκιβωτίζει ζεύγη **πεδίου-τιμής** ή/και πίνακες. Ένα παράδειγμα ενός τέτοιου αρχείου για ένα προϊόν ενός ηλεκτρονικού καταστήματος θα μπορούσε να είναι το ακόλουθο:

```
{
  "Όνομα Προϊόντος": "Παντελόνι XXL Αντρικό",
  "Τιμή": 30,
  "Μέγεθος": "XXL",
  "Διαθέσιμα Χρώματα": [
    "Μαύρο",
    "Κόκκινο",
    "Μπλε"
  ],
  "Διαθέσιμα Κομμάτια": 12,
  "Επιστροφές": "Δεκτές"
}
```

Σχήμα 6: Μορφή JSON αρχείου

Εδώ καθίσταται φανερό πώς όχι μόνο υποστηρίζονται **πίνακες** από τη συγκεκριμένη μορφή αρχείου αλλά και ζευγάρια της μορφής:

<Field >: <Value >

με **μικτούς** Τύπους Δεδομένων. Έτσι, ο προγραμματιστής μπορεί με ευκολία να ενσωματώσει όλες τις πληροφορίες ενός μηνύματος σε ένα μόνο **.json** αρχείο, χωρίς να αναλωθεί σε διάφορες χρονοβόρες μετατροπές.

2.4.2 Αιτήματα HTTP(S)

Κάθε χρήστης του διαδικτύου, αρχάριος ή μη, έχει αλληλεπιδράσει με το **HyperText Transfer Protocol (HTTP)** ακόμα κι από τον φυλλομετρητή του. Πρόκειται για το πιο ευρέως γνωστό πρωτόκολλο διαδικτυακής επικοινωνίας και παραμένει σε χρήση από το 1991 μέχρι σήμερα. Όπως είναι αναμενόμενο, έχουν υπάρξει ποικίλες αναβαθμίσεις και παραλλαγές του,

οι οποίες θα αναλυθούν επιγραμματικά προκειμένου να αναδειχθεί τόσο η χρησιμότητα αλλά και η συσχέτιση τους με την παρούσα ανάλυση.

2.4.2.1 Το παραδοσιακό HTTP

Ο βασικός σκελετός του κλασικού HTTP [3] στηρίζεται στην απλή αλληλουχία:

1. Σύνδεση στον εξυπηρετητή
2. Αίτημα προς τον εξυπηρετητή
3. Απάντηση από τον εξυπηρετητή

Το κρίσιμο σημείο των παραπάνω διαδικασιών είναι το **αίτημα** στον εξυπηρετητή. Ουσιαστικά εκεί εκδηλώνεται η **ανάγκη** που έχει το πρόγραμμα (και κατά συνέπεια ο χρήστης) και με βάση αυτή καθορίζεται και η απάντηση που θα ληφθεί.

Το αίτημα αυτό θα παρουσιαστεί στον απομακρυσμένο υπολογιστή με μία από τις **μεθόδους** του πρωτοκόλλου. Για λόγους λακωνικότητας, θα αναλυθούν μόνο αυτές που χρησιμοποιούνται στην τρέχουσα εφαρμογή, οι οποίες τυγχάνει να είναι ίσως οι πιο βασικές και διαδεδομένες στο διαδίκτυο.

- Μέθοδος GET

Χρησιμοποιείται κυρίως όταν ο υπολογιστής θέλει να διαβάσει το περιεχόμενο ενός πόρου του εξυπηρετητή. Για παράδειγμα, όταν ένας φυλλομετρητής ζητά το περιεχόμενο της σελίδας `http://www.di.uoa.gr` το αίτημα που στέλνεται έχει την εξής μορφή:

```
GET / HTTP/1.1
Host: www.di.uoa.gr
User-Agent: Mozilla/5.0 \
(Windows NT 10.0; Win64; x64; rv:91.0) \
Gecko/20100101 Firefox/91.0
Accept: text/html, application/xhtml+xml, \
application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Pragma: no-cache
Cache-Control: no-cache
```

Από την πρώτη γραμμή, που αποτελεί την **κεφαλίδα** δηλώνεται τόσο η ζητούμενη μέθοδος όσο και η έκδοση του HTTP που χρησιμοποιείται. Το υπόλοιπο αίτημα περιέχει ένα σύνολο πληροφοριών για τον browser και για τον τρόπο χειρισμού της μεταδιδόμενης πληροφορίας.

Αξίζει να σημειωθεί πως, τα αιτήματα GET επιδέχονται παραμετροποίησης. Μπορεί να ανατεθεί ένα οποιοδήποτε σύνολο τιμών σε μεταβλητές και να μεταφερθούν στον εξυπηρετητή. Συγκεκριμένα, η προσπέλαση τού: `http://www.di.uoa.gr/?SomeVariable=Content` δίνει το εξής αίτημα:

```

GET /?SomeVariable=Content HTTP/1.1
Host: www.di.uoa.gr
User-Agent: Mozilla/5.0 \
(Windows NT 10.0; Win64; x64; rv:91.0) \
Gecko/20100101 Firefox/91.0
Accept: text/html, application/xhtml+xml, \
application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Pragma: no-cache
Cache-Control: no-cache

```

κι έτσι είναι εύκολο για τον εξυπηρετητή να καθορίσει το περιεχόμενο της παραμέτρου *SomeVariable* με βάση κάθε ερώτημα που διαβάζει.

- Μέθοδος POST

Σε αντιδιαστολή με την GET η POST χρησιμοποιείται κατά βάση για την **αποστολή δεδομένων**. Ως εκ τούτου, ο χειρισμός μεταβλητών **ούτε είναι ορατός**, ούτε επηρεάζεται (υπό σωστή χρήση) από το εκάστοτε url σύνδεσης. Γεννάται ωστόσο το ερώτημα που αφορά στη μορφή αυτών των δεδομένων. Ας ξεκινήσει η διαδικασία απάντησής του, από ένα απλό παράδειγμα POST:

```

POST /test HTTP/1.1
Host: foo.example
Content-Type: application/x-www-form-urlencoded
Content-Length: 27

```

```
field1=value1&field2=value2
```

Εδώ οι μεταβλητές *field1*, *field2* παίρνουν τις τιμές *value1*, *value2* αντίστοιχα. Ο τρόπος αυτός μεταφοράς των δεδομένων σίγουρα είναι εύκολα κατανοητός. Είναι όμως εύκολα συντηρήσιμος για μεγάλες εφαρμογές?

Είναι προφανές πως υπάρχει ανάγκη για κάτι πιο αποδοτικό. Σε αυτό το σημείο αξιοποιείται η JSON μορφή αρχείων, καθώς **υποστηρίζεται από το ίδιο το πρωτόκολλο**.

Πιο συγκεκριμένα, αιτήματα της μορφής:

```

POST /test HTTP/1.1
Host: foo.example
Content-Type: application/json
Accept: application/json

```

```

{
  "field1": value1,
  "field2": value2
}

```

είναι απολύτως έγκυρα και διευκολύνουν την προγραμματιστική προσαρμογή των δεδομένων υπό αποστολή στο HTTP.

2.4.2.2 Η καινοτομία του HTTPS

Η μέχρι τώρα διατυπωμένη στρατηγική ανταλλαγής πληροφοριών με το HTTP είναι εύκολη στην υλοποίηση. Παραμένει ωστόσο διάτρητη, έχοντας μεγάλα κενά ασφαλείας. Για την αποφυγή τους, δημιουργήθηκε μια παραλλαγή του HTTP, γνωστή ως HTTPS. Παρέχει κρυπτογράφηση των μεταφερόμενων πληροφοριών μεταξύ πελάτη και εξυπηρετητή, κάνοντας την επικοινωνία ασφαλέστερη ακόμη κι αν κάπως υποκλαπεί το περιεχόμενο της.

2.4.2.3 HTTP και Android

Το λειτουργικό σύστημα Android, σαν κάθε σύγχρονο ΛΣ υποστηρίζει τα TCP/IP πρωτόκολλα στα οποία συγκαταλέγεται και το HTTP(S). Με την βοήθεια βιβλιοθηκών ανοιχτού κώδικα, ένας υποψήφιος προγραμματιστής μπορεί να αξιοποιήσει κάθε μέθοδο HTTP που προαναφέρθηκε με σχετική ευκολία. Ο χειρισμός της σύνταξης για μια από τις βιβλιοθήκες αυτές, την Fuel είναι ο παρακάτω:

```
val params: ArrayList<Pair<DT, DT>> = ArrayList()
params.add(Pair(<some_field>, <some_value>))
val (req, resp, result) = Fuel.post(<domain>, params).response()
```

όπου DT είναι ένας οποιοσδήποτε τύπος δεδομένων της Java (π.χ String). Η κλήση της συνάρτησης response() επιστρέφει μέσα στο πεδίο **result** ένα αντικείμενο κλάσης Result που περιέχει την απάντηση του εξυπηρετητή στο POST και ένα πεδίο σηματοδότησης κάποιου πιθανού σφάλματος.

Η ανωτέρω λιτή σύνταξη, φαίνεται αναμφίβολα ελκυστική για έναν μελλοντικό Android προγραμματιστή. Ωστόσο, υπάρχουν κάποιες ((κρυφές)) παγίδες, που ξεκινούν από την έκδοση Android 9 Pie. Πιο συγκεκριμένα, για λόγους ασφαλείας, κάθε νεότερη έκδοση του Android, ξεκινώντας από αυτή, απαγορεύει την δημιουργία αιτημάτων HTTP από οποιαδήποτε εφαρμογή. Παραμένουν ως έγκυρα, μόνο τα HTTPS αιτήματα. Αυτό μπορεί να είναι ιδιαίτερα προβληματικό σε ένα περιβάλλον ανάπτυξης εφαρμογών, που οι διάφοροι εξυπηρετητές είναι απλά υπολογιστές του τοπικού δικτύου. Προκειμένου να αντιμετωπιστεί αυτή η δυσκολία, παρουσιάζονται οι εξής λύσεις [4]:

- Για περιβάλλοντα ανάπτυξης: Εξαναγκασμός της εφαρμογής για τη χρήση HTTP
Για να παρακαμφθεί ο περιορισμός του HTTPS, θα πρέπει στο αρχείο AndroidManifest.xml το οποίο ουσιαστικά μεταδίδει μια περιγραφή της εφαρμογής στο ΛΣ, να προστεθεί η επιλογή:

```
<application android:usesCleartextTraffic="true">
</application >
```

- Για περιβάλλοντα μικρής διαθεσιμότητας του προγράμματος: Έκδοση ενός αυθυπόγραφου πιστοποιητικού
Προκειμένου το HTTPS να διασφαλίσει την αυθεντικότητα τού υπολογιστή με τον οποίο προσπαθεί ο πελάτης να επικοινωνήσει, χρησιμοποιεί **SSL πιστοποιητικά**. Αυτά συνήθως προέρχονται από **έμπιστες αρχές έκδοσης πιστοποιητικών**, ωστόσο είναι εύκολη και η δημιουργία ενός πιστοποιητικού από τον οποιονδήποτε χρησιμοποιώντας το δωρεάν εργαλείο **openssl**. Τα πιστοποιητικά που εκδίδονται και υπογράφονται από ιδιώτες, δεν είναι εφάμιλλης αξιοπιστίας μ' αυτά που υπογράφονται από εκδοτικές αρχές, καθώς δεν υπάρχουν παρόμοιες δυνατότητες ελέγχου κι ανάκλησής τους.
- Για περιβάλλοντα παραγωγικής λειτουργίας: Κατοχύρωση ενός **domain** κι έκδοση πιστοποιητικού από έμπιστη αρχή.
Μια εφαρμογή που θα είναι διαθέσιμη στο ευρύ κοινό, δεν μπορεί παρά να στηρίζεται

σε ένα δικό της domain και να έχει ένα κατάλληλο πιστοποιητικό SSL. Σε αντίθετη περίπτωση, οι ενδεχόμενοι κίνδυνοι ασφαλείας μπορούν να εκθέσουν ανεπανόρθωτα τόσο την εφαρμογή και τους προγραμματιστές της, όσο και τους χρήστες της.

2.4.2.4 Αυθυπόγραφα Πιστοποιητικά SSL

Για τον μέσο προγραμματιστή, η καλύτερη επιλογή από τις παραπάνω για το μεγαλύτερο διάστημα ανάπτυξης/δοκιμών, θα είναι η έκδοση ενός αυθυπόγραφου πιστοποιητικού SSL. Η διαδικασία αυτή μπορεί να γίνει με το πακέτο openssl σε Linux διανομές με τη χρήση του ως εξής [5]:

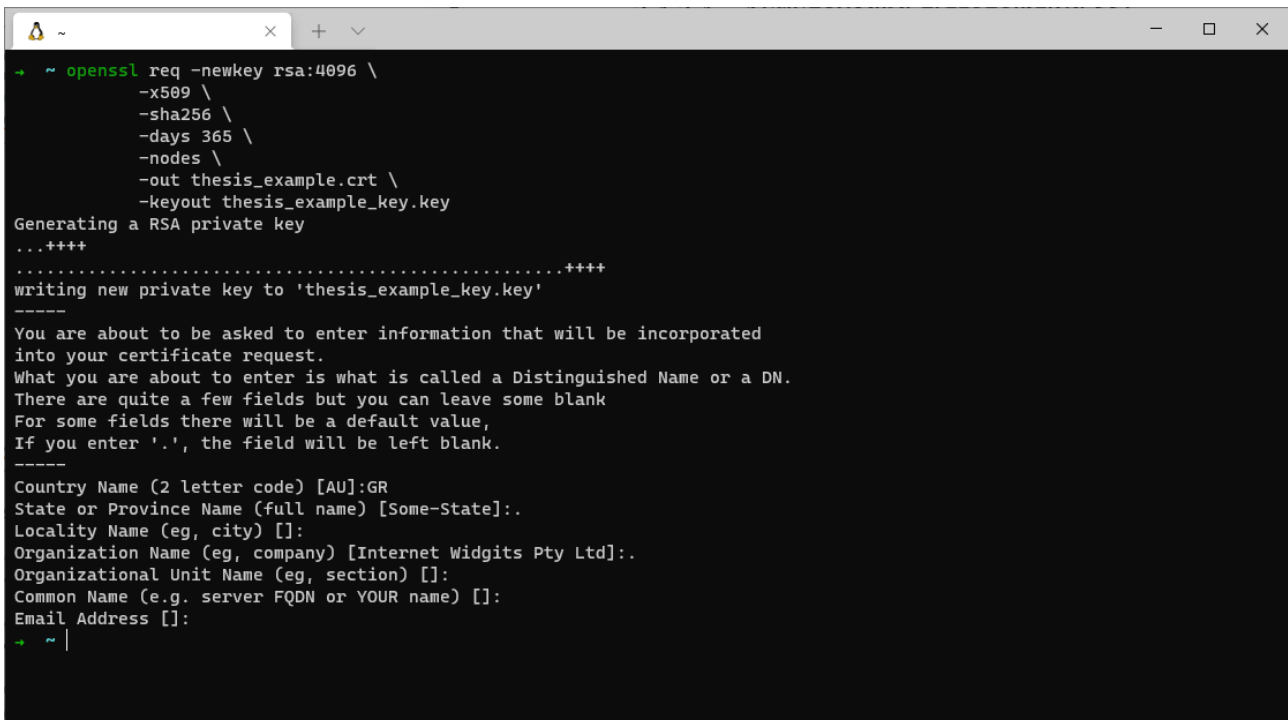
```
openssl req -newkey rsa:4096 \
-x509 \
-sha256 \
-days 365 \
-nodes \
-out <filename>.crt \
-keyout <filename>.key
```

όπου:

- -newkey rsa:4096 σημαίνει να χρησιμοποιηθεί ένα RSA κλειδί των 4096 bits.
- -x509 σημαίνει να δημιουργηθεί ένα πιστοποιητικό που θα υπακούει στη μορφή X.509
- Η παράμετρος days αναφέρεται στο χρονικό διάστημα ισχύος του πιστοποιητικού
- Οι παράμετροι out,keyout σηματοδοτούν τα ονόματα των παραγόμενων αρχείων

Αρκεί μια απλή εκτέλεση του παραπάνω και η απάντηση σε μερικές ερωτήσεις, προκειμένου να συμπληρωθούν οι κατάλληλες πληροφορίες του πιστοποιητικού.

Πράγματι, το αποτέλεσμα μιας τέτοιας εντολής σε μια διανομή Ubuntu 20.04 είναι το ακόλουθο:



```
~ openssl req -newkey rsa:4096 \
-x509 \
-sha256 \
-days 365 \
-nodes \
-out thesis_example.crt \
-keyout thesis_example_key.key
Generating a RSA private key
...++++
.....++++
writing new private key to 'thesis_example_key.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:GR
State or Province Name (full name) [Some-State]:.
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:.
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
~ |
```

Σχήμα 7: Δημιουργία πιστοποιητικού

Επιβεβαιώνοντας το περιεχόμενο των δύο αρχείων, παίρνουμε τις παρακάτω εικόνες:

```

File: thesis_example.crt
-----BEGIN CERTIFICATE-----
1  MIIE+zCCAuOgAwIBAgIUe64wkLIphQSI1WCD+nm0mwaKGR8wDQYJKoZIhvcNAQEL
2  BQAwDTELMAkGA1UEBhMCFR1IWhhcNMjEwODI1MDcyMTH3WhcNMjEwODI1MDcyMTH3
3  WjANMQswCQYDVQQGEwJHUjCCAIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIB
4  AKmIAoEM5ctkH2V3x899VAKLJjJUqOLpRbMUmTccj+qNM0q5S6ieH5oCmpE/Uafs
5  kJW07NuaUTn8m+A5UhFz/q0tNaum8u6Xj8sKaBuQ1bsGhY7KHCCXvoKFRmzBaAVu
6  quEu97X7UxPo6kTLFp8I9PFbe9c5r0xFPcKPvWmeezr6XD1sbfdjrbyjqVj76BFS
7  6YhptNnOP7S2hKmpN+f42ys/VKjKDZdqZrrzfsz3W4+wFVQgzgjnJfZea27Be5LY
8  P+qCMPN2FHwor33Y1sLF72Ei7pgLV14wwwvCtmo9WcjsrFA8r8mAirNYmPVjFTeqe
9  8+Dyv4tyf9+doixSv2YvBh5JHBSFKbI7nFswG8aMi00802HzIZumnkIh7mV+Nz+K
10 UpWdxBJCAHAcYHVWuNmM4Z1yObAUEV1umnzIBfyHFkHAZ+cPw6BFsXAp7FG6RPTG
11 htD6PmWptw0bwwccBLcM7Df70bYgJQB5AW1YavScNYQ72rMzCEAYRW5y3UK0aSqN
12 uMRi8jqQS8H94sDmg+fTiXZf3JucgA2kKT1vS85Y1VbyHNEbx267bfY+VCYZ4wga
13 2L3lxRDdY/TrNxCLzAH0vi72B5Ck9j8DuRkF8Tkr6jgJ6ruMH4sXt+tDDpKLYSHB
14 Dpechmh1T4Adw+BgzQdiIjfhpeLD6e4ymrM0q0q2BwidAgMBAAGjUzBRMB0GA1Ud
15 DgQWBBOm8QaoIKMmExfRwJNFtgr/h/x02jAfBgNVHSMEGDAWgBQm8QaoIKMmExfR
16 wJNFtgr/h/x02jAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUAA4ICAQBI
17 rhjSNW8LPwsbLzXv13oKFSKAOpKLnz49svfkYwpUiZtw5DK/grLSgJN0j6V3ot2e
18 bd1m6wr40fLcd4bhrgJRp/PN+f1AsPicsuDs4/06gdWrvNvjhhrVQ3kdfML78L+o
19 G3c16fh+cEH1cupXAKMAegAPYxqTgMgpQKw+TeFp8BLciWnuTm4w+3fe0j53LcyD
20 OuoTVK9rj10zYK2sCy/XeNtvuqPtwiwZqa5cmkInsQDnJ+LKLXkxkguB28idui0m
21 s6gkDQds8pbv9npMumjQNBwc0nOANGwqEXD0zv6Tff2NiBnk7y2HYIvE4A3aiEGE
22 /BBEGPOXFRhcI8t34qVryEsurF2/QfwIRFStY1LR0gPTr4Ru4YwE148fqyHgZMfG
23 6BaghIRofxpnLUekfuvFCmtPLH4ymYP2UmY6zPDazsXao1VAHCIh0Vb4Vuhk0pWT
24 LPLoKE0UNfo3QvosB+f3+82D01JLJjYgNLXbEe34L1iMLPcjUb9QYrs9Rqhy+Tv
25 /8fDkIX+Qgy0PC1jKxp3iNzshPRRr9m0r0svGpmsBhb/U1pokucYt2q0ByUhyS4e
26 nU9+5Nv/m7+0Fu5EQzyUPyfVewaGSQGN7ida3DbTSVj2j/wZDn48jUbubqNuTs0M
27 qk6cP9Dch3NplEw08uliqBEUiJJhRaIOlpnDpxJ8Pg==
28 -----END CERTIFICATE-----
29
~
[END]

```

Σχήμα 8: Περιεχόμενο .crt αρχείου

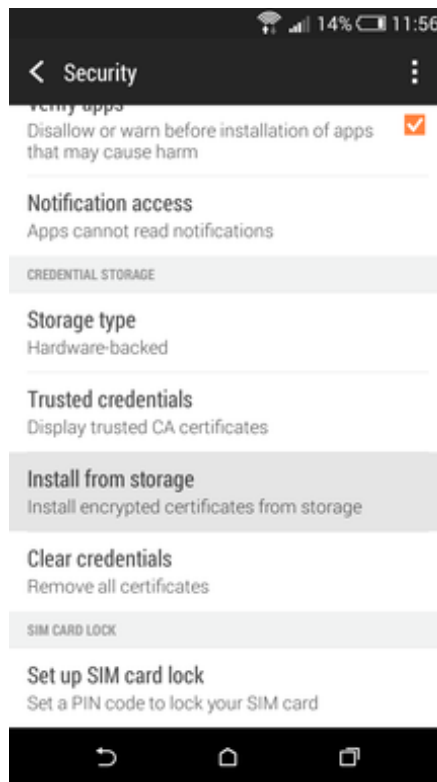
```

1  -----BEGIN PRIVATE KEY-----
2  MIIJQwIBADANBgkqhkiG9w0BAQEFAASCCS0wgggkPAgEAAoICAQCpiAKBDOXLZB9L
3  d8FPVQcPSyVVKji6UwzFJK3HI/qjTKKuUooh+aAppRP1Gn7JCvtOzblE5/Jvg
4  0V2RC/6jrZ2rpuLuL4/LCmgbkNw7BoWQyhwL76Cn9ZswWgFbqrhLve1+1MT60PE
5  5RaFCPTxw3tH0azsRT3Cj71pnns6+Lw9bG33Y628o6LY+gRUUmIabTzj+0toSp
6  qTfn+NsrP1Soyg2Xama6837M91uP1hVUIIM4I5432XmtuwXuS2D/qgjdZdhr8KK99
7  2NbJRe9hIu6Yc1deMMLwrZqPvN7KxQPK/JgIqzWJj1YxU3qnvPg8r+Lcn/fnaIs
8  Ur9mLwYeSRwUhsmy05xbFoPGjItKPNH8yGbpp5Cie5Lfjc/iLkVncQS0gBWHMh1
9  cLzZ0GdcjmwFBFdbpp84gX8hxShwGfnKcOgRbMQKexRgUaUxobQ+j5lqbVjm8MH
10 HAZXD0w3+9G2ICUAeQFtWGr0nDWE09qzMwhAGEVuct1CtGkqjbJEyV16kEvB/eLA
11 5hvn04L2X9ybnIANpCk9b0v0WNWg8hzRG8duu232PLQmGeMIGtpd5cUQ3WP06zcQ
12 pcwB9L4u9geQpPY/A7kZBF5k+o4Ceq7jB+LF7frQw6SpWEhwQ6XnIZoDU+AHCPg
13 YGUYHvI34qS3enuMpqzNKjqtgVonQIDAQBAoICAQCBAwF4iRSvPSLcZQt7DuGLH
14 CP0+AjusR9uXn0LQu+S0IPkw4Xa9nCAVsk56QlsNyeozGmE6BlbqqaLnrdX0k5v
15 GZBwVFWUNJ3SSam6h1R9qxUnXxq9+0XNFxHcHpiTPT7DXi4qsys3jsLWZ3uzxpgo
16 7dLhfwRUXFMS7eenRYHgt6e3WCPwDagwZG264LD0esc//rIVbqgEbkYinf2o
17 kPULpnFWLIRBPgGrgBoswDKcy+rFN+wYRTW01BLVJeFta61peNNe34jJKd/h9kur
18 LaAs8Gky5UeCcdJcQe4PaP4Wso1b9M4RrPzA52mmC3gej8v4P28NR8lqY66qh40G
19 +x2fjXN5qGQvNINC628gwZafeGomCNkMPf6nA3b5+Mjagdkik/JiKdxPMCvN/Q
20 aZALwot5TnJQo/DMrAJ6R1sDcFUWau6YJdEBi0zvtV+sNjy5AeJMgyP61xN3I2Ta
21 gKULYEyoxyRML4r8IXLW/6cP59q44BaR0BvdBocn/ioXJENwKfMK4/7urP9n0hP9
22 XVAJmimKhnKxIrpCzfdLIQh0Tg3kfi3XLaI/Wqsjqz0HWpUV10JaUgCLxqB10r8
23 zgN9V+8sBXiv4MFBeP2acXqQ6Y6XrKdcwqsq8f5B9o1tzQU9pqh4b5nR5+300L7U
24 g0U7ik7abR17xKpqIBm5QKCAQEANcs0r22RETvDxybAA0FHWsYqT8xkTwnIKB
25 2E+7qanIbmrgHULMyErB15fE6Zn05fY4wTOhvxASBEPrCIrEPWFbinyk0Y/nDSJq
26 qIrs+aUIwsjgvKpBjzZQtR0J6s4gqskPeCY7zsEiU0djuKb33wrFHQ5vNk+PBu
27 3AgTYRDWlpvKhsx/x9C7PF0h2M07ns2osQwco/KODoEQqkwDdTFLDi72GvXA2084
28 RMiPI3mPDTZ7L+aL0TEp/vFu093KITBh+vNtY92kb+JZtvJqLwhQA+ZSjPwGSXW6
29 w0xOUFStlqzVif3HcdZcyVwdALDsUu9jDNTFMvuPpZUCPTsOWKCAQEAWQaXkGJX
30 tuzpKFwVT0lik9oL7xscKglDVo2MLTmcJEEAmM0G6Tfi4ArCzGAS+654C63twBE
31 qeSK+ttNjDdieogkD6t4v8usr4A4z5fCwXnztpgdNfrU1kmo+zd73TVFXfTB1Y
32 Ao07Zkw5TJ6tWAgLcyIJXkMz2nWfi0GywoXMa42iYwnDnDGeUp962aaa+RJKt0E
33 aHETBu2H10kAYXMFNBtuXf6Sbbop263EeD8ajtGUCyL6f/vokspZ5daBIYJH103
34 kFLIRVSSHerrRJDk6x25WTUpMmmr0Mz6fYEKgy3fNodx3H+oafMte+3E+QU2rBr3
35 +DqyDyowLACpBwKCAQEArB7MysMr9tMjH/iqN1dohw38HRvkjfuAt0uc9LkDdnyM
36 /Uor5NoAd4wS6305wIKISq7U/8IA3nz1PYA5K6xrv3LGBtFCDNM2s21XbTL8jhRE
37 8wq4UHWdbuTLD64IRZ3jotrVQEWZuzSTG8jzgLdOQM+OgfgOS20KKEwW8cYoxxfw
38 znDw9RrhYua6jGzncwElyuvKOT85tz2NtBC67tJILvb8z2sFTNwsV/WMYGYjba
39 dydmImNVKpUDQDQkHxaqBwALDxRz50LIEONHmoX1QPvQyCcxsgM/g2QR4LTVVl
40 ZaSHozrxCBDLvyokJqc/G+isCyS02dq00tn5rnpQKCAQBWkD8cmvFjNcCnK0
41 5sbqWhT08YJJJf/9bhC48jdSAsrW6Y281Dwgu8JwVZS2iOrtiRNzA+I+ypYIFrB
42 2HeD7VRnFhFYLFRZ0dm8cnbTMA4dT/KQLMSuKcQ2Y3MolionJ3W81tsP6RbnSnaS
43 A0nHkD3Wg8FF7P/xi0BrTUVVdL3lcoPkdjYo7jJihZpP8s7s+MYnrqWRDjTPdE34
44 WF8zZzefKEuLQcn/NzaC1FYg0EBpXpAthxCwiIFrXP7B2tC3KLWHySG7zG7LjX2
45 s19gnWkZdpvVsV0/LEXD5nXjScPmbyS2PA5kDpp4xf7zPwTBEmlr7Ac8jOLJbdN
46 eJdJAoIBACgcX7epzOPSW1+v50fg3MzXdrbgCNHh0iK80bm89HeqbnLIW03awurZ
47 rMLiRynwvINTdRe7equ4sxdenS/DLPf3Mr0po5B1lhpHf/9SF53paj2XL0ByxFuJE
48 S3lz0TL4iEuiL8IFbkKas6azvTfd1av/So6wRNqetyoWY+iS3593JuYA+8eUZXX
49 SuMU/cUmb2bsN1WUR/A3b0aJcVJJtmZD9fIjG4mqjdk96674dw0JspLcTaLWq0jF
50 eVJlRtVq6fvkIrKc0WzngyVeNcVXuVzMYZz3y0JjA7J4Z2LFSM+JmU1A/Zrf2o1S

```

Σχήμα 9: Περιεχόμενο .key αρχείου

οι οποίες πρακτικά αντιπροσωπεύουν και την έξοδο του προγράμματος openssl. **Αξίζει να σημειωθεί**, πως τα πιστοποιητικά αυτά, είναι απλά αρχεία. Για να αξιοποιηθούν από την εκάστοτε κινητή συσκευή, πρέπει να εισαχθούν σε αυτή, στο ειδικό μενού των κινητών Android που φαίνεται στην παρακάτω φωτογραφία:



Σχήμα 10: Εισαγωγή Πιστοποιητικού

Παράλληλα, ανάλογα με την έκδοση του ΛΣ Android ενδεχομένως να χρειαστεί μια μετατροπή της μορφής του πιστοποιητικού, που γίνεται με την εξής εντολή:

```
openssl x509 -inform PEM -outform DER \
-in <name>.crt -out <new_name>.crt
```

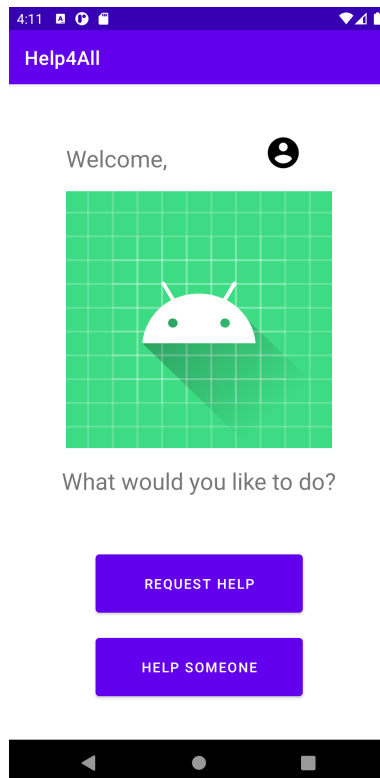
η οποία σε περίπτωση επιτυχίας δεν παράγει κάποια έξοδο στην οθόνη, δημιουργώντας μόνο το αντίστοιχο αρχείο.

2.5 Παραδείγματα Χρήσης

Όπως είναι φανερό, κάθε σενάριο χρήσης του εν λόγω προγράμματος περιστρέφεται γύρω από δύο βασικές ομάδες χρηστών:

- Τους εθελοντές-αρωγούς
- Τα άτομα υπό ανάγκη βοήθειας

Έτσι, προκειμένου να είναι ξεκάθαρη η λειτουργικότητα του Help4All ο κάθε χρήστης καλείται από τ' αρχικά στάδια της εφαρμογής ν' αποφασίσει ποια από τις δύο κατευθύνσεις θα ακολουθήσει. Αυτό γίνεται στην οθόνη που ακολουθεί:



Σχήμα 11: Αρχική σελίδα για συνδεδεμένο χρήστη

Ύστερα υπάρχουν για τις δύο ομάδες χρηστών οι αντίστοιχες σειρές ενεργειών:

Εθελοντές:

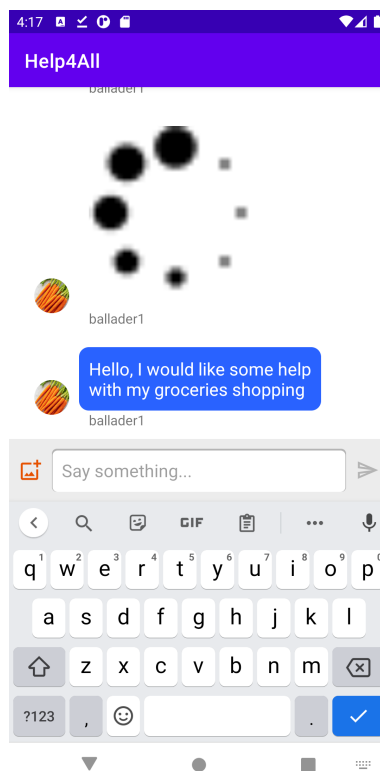
1. Έλεγχος λίστας ενεργών αιτημάτων
2. Επιλογή αιτήματος προς ικανοποίηση και προβολή λεπτομερών πληροφοριών για αυτό
3. Συνομιλία με τον χρήστη που κατέθεσε το αίτημα
4. Από κοινού αποδοχή αιτήματος

Αιτούντες Βοήθεια:

1. Καταχώρηση Αιτήματος
2. Αναμονή μέχρι τη λήψη ειδοποίησης για την αποδοχή του αιτήματος από κάποιον εθελοντή

3. Συνομιλία με τον χρήστη που αποδέχθηκε το αίτημα
4. Από κοινού αποδοχή αιτήματος

Ενδεικτικά, μια οθόνη κοινή και στα δύο σενάρια φαίνεται παρακάτω:



Σχήμα 12: Συνομιλία με το χρήστη που αποδέχθηκε το αίτημα

Αξίζει να σημειωθεί πως τα παραπάνω αποτελούν τα **Κύρια Σενάρια Επιτυχίας**. Ως δευτερεύουσες περιπτώσεις υπάρχουν και υλοποιημένες δυνατότητες για μη αποδοχή αιτημάτων, ή/και διαχείριση εσφαλμένης αποδοχής.

3 ΚΥΚΛΟΣ ΛΕΙΤΟΥΡΓΙΑΣ

Έχοντας αναλύσει σε βάθος τις τεχνικές που χρησιμοποιούνται κατά τη λειτουργία της εφαρμογής, θα προβούμε στην παρουσίαση των βασικών ροών επικοινωνίας κι αλληλεπίδρασης με το χρήστη

3.1 Είσοδος στο πρόγραμμα

Για να έχει πρόσβαση ο χρήστης στο πρόγραμμα κάνει είτε **εγγραφή** είτε **είσοδο** χρησιμοποιώντας την ηλεκτρονική του διεύθυνση και έναν κωδικό πρόσβασης. Η επικοινωνία με τον εξυπηρετητή Firebase που εξηγήθηκε, γίνεται με τον ακόλουθο κώδικα:

```

if(email.isNotEmpty() && pass.isNotEmpty()) {
    GlobalScope.launch(Dispatchers.IO) { this: CoroutineScope
        try {
            //make sure to distinguish between firsttimer and existing user
            if(firsttimeUser) {
                //await is critical on the following
                // we can't update the pic if the user hasn't been created
                auth.createUserWithEmailAndPassword(email, pass).await()
                //handle the profile pic
                auth.currentUser?.let { it: FirebaseUser?
                    val update = UserProfileChangeRequest.Builder().setPhotoUri(imgUri).build()
                    it?.updateProfile(update)
                }?.await()
            }else {
                auth.signInWithEmailAndPassword(email, pass).await()
            }
            loginFinished()
        } catch (ex: Exception) {
            /*
             * make sure to transfer firebase error messages so the user knows when
             * a mistake has been made. E.g: too short password
             */
            loginError(ex)
        }
    }
}

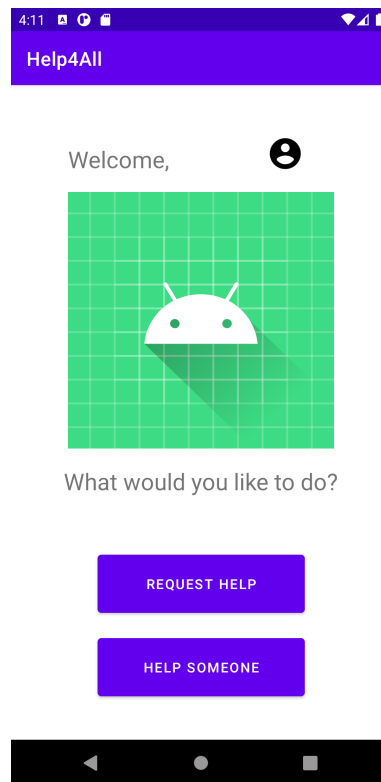
```

Σχήμα 13: Κώδικας εισόδου/εγγραφής χρήστη

ο οποίος εξασφαλίζει τόσο την είσοδο στην εφαρμογή, όσο και τη δημιουργία μιας συνεδρίας (session).

3.2 Παροχή βοήθειας

Οι εθελοντές που παρέχουν τις διάφορες υπηρεσίες τους, μπορούν να δουν μια λίστα με τα ενεργά αιτήματα πατώντας το πλήκτρο "Request Help" της παρακάτω οθόνης:



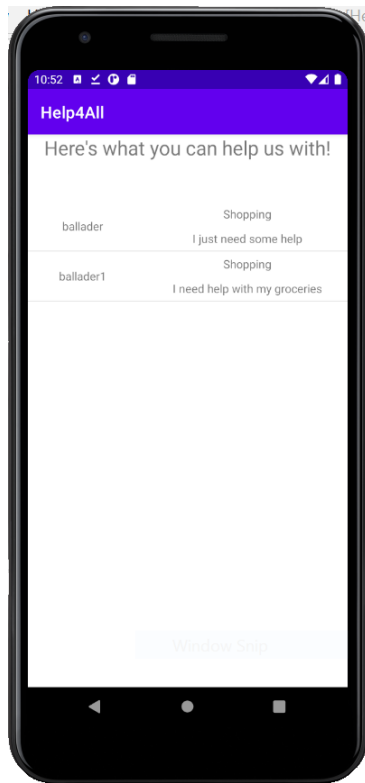
Σχήμα 14: Κεντρική Οθόνη

Από τον server λαμβάνεται ένα αρχείο JSON που είναι της μορφής:

```
{
  "result0":{
    "user":"ballader",
    "type":"Shopping",
    "comments":"I just need some help",
    "lat":"",
    "long":""
  },
  "result1":{
    "user":"ballader1",
    "type":"Shopping",
    "comments":"I need help with my groceries",
    "lat":"37.97945",
    "long":"23.71622"
  }
}
```

Σχήμα 15: Λίστα αιτημάτων εν ενεργεία

κι αποτυπώνεται στην οθόνη μέσω μιας *scrollable* λίστας:



Σχήμα 16: Λίστα αιτημάτων στην εφαρμογή

Από εκεί ο χρήστης μπορεί να επιλέξει το αίτημα που τον ενδιαφέρει και αφού δει περαιτέρω πληροφορίες για αυτό, να μεταφερθεί σε μια ιδιωτική συνομιλία με το άτομο που ζητά τη βοήθειά του. Τα αρχεία της συνομιλίας φυλάσσονται σε προσωποποιημένους καταλόγους στη ΒΔ του Firebase που φαίνονται παρακάτω:



Σχήμα 17: Δείγμα καταλόγων συνομιλίας

Ανά πάσα στιγμή, οποιοδήποτε από τα δύο μέλη μπορεί να απορρίψει την συνεργασία τους καθιστώντας το αίτημα ξανά ενεργό. Μόνο όμως ο δημιουργός του μπορεί να το καταστήσει ικανοποιημένο. Η επαλήθευση αυτή της ταυτότητας γίνεται μέσω του μηχανισμού των tokens που αναλύθηκε παραπάνω.

3.3 Αίτηση βοήθειας

Ως μια βασική ομάδα χρηστών, οι χρήστες που αιτούνται βοήθεια καλούνται να συμπληρώσουν την ακόλουθη φόρμα:

Σχήμα 18: Βασική Φόρμα Αίτησης Βοήθειας

καθώς και να αποφασίσουν αν θα μοιραστούν την τρέχουσα τοποθεσία τους, στην οποία ο προγραμματιστής έχει **άμεσα** πρόσβαση με βάση υπάρχουσες τεχνικές του Android. Στον εξυπηρετητή οι πληροφορίες φτάνουν ξανά μέσω JSON της μορφής:

```
{
  "username": "ballader1",
  "type": "Shopping",
  "comments": "I need help with my groceries",
  "latitude": "37.97945",
  "longitude": "23.71622"
}
```

Σχήμα 19: JSON αρχείο Αίτησης Βοήθειας

και μόλις σταλθούν επιτυχώς, εμφανίζεται αντίστοιχη ειδοποίηση στο χρήστη. Ο PHP server σε αυτό το διάστημα, ενημερώνει την ΒΔ MySQL η οποία αποκτά αυτή την εικόνα αιτημάτων:

4 ΤΕΧΝΙΚΕΣ ΑΝΑΠΤΥΞΗΣ

4.1 Η γλώσσα προγραμματισμού Kotlin

Η Kotlin είναι μια μοντέρνα αντικειμενοστραφής γλώσσα προγραμματισμού **ανοιχτού κώδικα**. Σχεδιασμένη ειδικά για την ανάπτυξη εφαρμογών για κινητά τηλέφωνα, παρέχει κατάλληλες προγραμματιστικές δομές ώστε να χρησιμοποιούνται στο μέγιστο οι δυνατότητες των Android τηλεφώνων ενώ παράλληλα να μειώνονται οι πιθανότητες λάθους χρήσης τού κώδικα. [6] Ως εναλλακτική γλώσσα ανάπτυξης χρησιμοποιείται η Java. Αυτό είναι φυσικό, μιας που η Kotlin παράγει κώδικα για το JVM και έχει μια 1-1 αντιστοιχία με τη Java. Είναι μάλιστα εντυπωσιακό, πως στα συνηθισμένα εργαλεία ανάπτυξης εφαρμογών Android παρέχονται εργαλεία μετατροπών από τη μία γλώσσα στην άλλη! Έτσι, μια εφαρμογή μπορεί να περιέχει μια οποιαδήποτε αναλογία κώδικα Java/Kotlin.. Για το Help4All ιδιαίτερα βοηθητικός υπήρξε ο ευφύεστατος σχεδιασμός της Kotlin που στοχεύει στην ελαχιστοποίηση του απαιτούμενου κώδικα για μια εφαρμογή και κατά συνέπεια των πιθανών λαθών σε αυτόν. Συνοπτικά, κάποια από τα πλεονεκτήματά της για προγραμματισμό κινητών τηλεφώνων είναι τα εξής:

1. Χειρισμός **Null Pointer Exceptions**
2. Εύκολη δημιουργία κλάσεων με αυτόματη δημιουργία μεθόδων get()/set() και toString().
3. Αυτοματοποιημένες κλασικές προγραμματιστικές διαδικασίες όπως μετατροπή μεταξύ τύπων (casting).
4. Βελτιστοποιημένη μεταγλώττιση που παράγει μικρότερα προγράμματα.

Για την παραγωγικότητα ενός προγραμματιστή, τα σημεία (1),(2) και (3) της παραπάνω λίστας είναι τα πιο σημαντικά, τα οποία θα αναλυθούν παρακάτω.

4.1.1 Χειρισμός Null Pointer Exceptions

Τα **Null Pointer Exceptions** αποτελούν ένα από τα συχνότερα σφάλματα σε προγράμματα. Δυστυχώς, το αν θα προκύψουν ή όχι εξαρτάται συνήθως από τη ροή εκτέλεσης μιας εφαρμογής. Έτσι, μπορεί σε μια περίπτωση οριακού (**Edge Case**) σφάλματος, ο εντοπισμός τους να γίνει καθυστερημένα. Η Kotlin επιχειρεί να οργανώσει τον κώδικα της με τέτοιο τρόπο ώστε να προβλέπεται εξαρχής στην λογική του η πιθανότητα ενός NULL pointer. Αναλυτικά, η σύνταξη:

```
var <varName>:<varType>
<varName> = null
```

θα αποτραπεί κατά τη διάρκεια μεταγλώττισης του προγράμματος, καθώς η μεταβλητή που παρουσιάζεται δεν έχει καθοριστεί ως μια που επιτρέπεται να πάρει τη συμβολική τιμή **NULL**. Με παρόμοιες τεχνικές, λάθη που θα εμφανίζονταν κατά την εκτέλεση, ακόμα και σπάνια ίσως, λόγω ενός σφάλματος που μπορεί να προέκυπτε υπό πολύ συγκεκριμένες συνθήκες, αποτρέπονται εξ' ολοκλήρου. Για την πληρότητα του εγγράφου, η σωστή σύνταξη για να επιτραπούν οι NULL τιμές θα ήταν η:

```
var <varName>:<varType>?
<varName> = null
```

4.1.2 Εύκολη δημιουργία κλάσεων

Στην κλασική Java μια κλάση με πεδία προκαθορισμένων τύπων της γλώσσας έχει την παρακάτω μορφή:

```

public class Person {
    private String name;
    private String surname;
    private String id;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getSurname() {
        return surname;
    }
    public void setSurname(String surname) {
        this.surname = surname;
    }
    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    @Override public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return
            false;
        Person person = (Person) o;
        if (name != null ? !name.equals(person.name) : person.
            name != null) {
            return false;
        }
        if (surname != null ? !surname.equals(person.surname) :
            person.surname != null) {
            return false;
        }
        return id != null ? id.equals(person.id) : person.id ==
            null;
    }
    @Override public String toString() {
        return "Person{" +
            "name='" + name + "' +
            "surname='" + surname + "' +
            "id='" + id + "' +
            '}'
    }
}

```

Ο κώδικας των μεθόδων που παρουσιάζονται πρακτικά έχει μια γενική μορφή που μπορεί να προσαρμοστεί σε οποιαδήποτε άλλη κλάση με αντίστοιχα πεδία. Η Kotlin εκμεταλλεύεται αυτό το γεγονός μετατρέποντας τον αντίστοιχο απαιτούμενο κώδικα σε:

```
data class Person(var name: String , var surname: String , var id
: String )
```

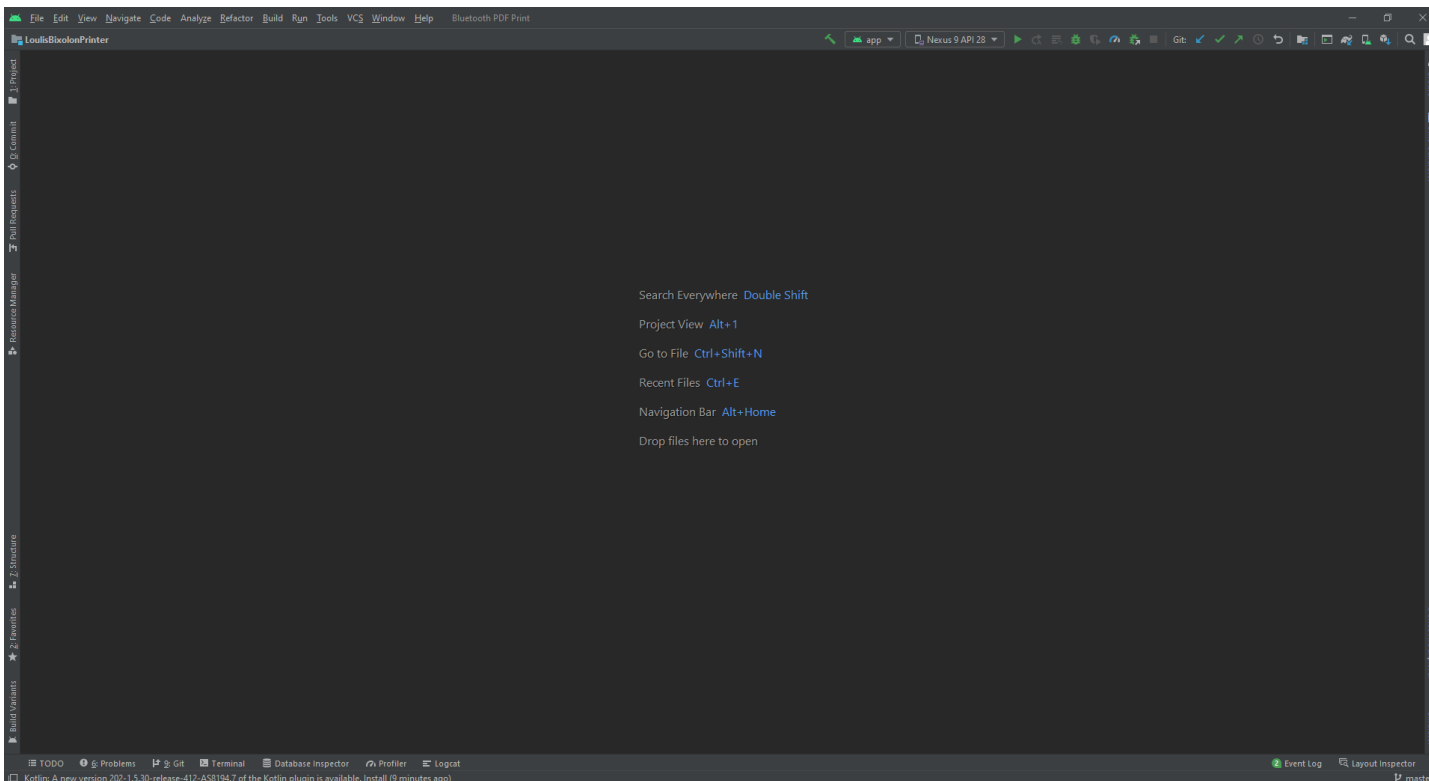
Η κλάση Person της μιας αυτής γραμμής κώδικα ενσωματώνει έτοιμες τις πιο πάνω λειτουργίες. [7]

4.2 Android Studio

Κυρίαρχο IDE για την ανάπτυξη μιας τέτοιας εφαρμογής, ανεξάρτητα απ' τη γλώσσα επιλογής, αποτελεί το Android Studio. Πέρα από την προφανή συμβατότητα με τις προαναφερθείσες γλώσσες, προσφέρει [8]:

- Version Control με πλήρη ενσωμάτωση με το Git
- Άμεση γραφική απεικόνιση του σχεδιασμού των παραγόμενων οθονών
- Δωρεάν προσομοιωτές Android για την άμεση εκτέλεση και δοκιμή των εφαρμογών από υπολογιστή
- Εργαλεία αποσφαλμάτωσης
- Κατασκευή των εφαρμογών με το Gradle για την αυτόματη λήψη κι ενσωμάτωση βιβλιοθηκών.

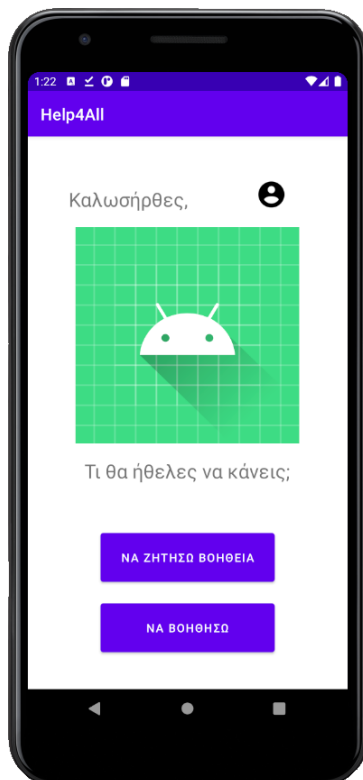
Ένα ενδεικτικό στιγμιότυπο του Android Studio φαίνεται εδώ:



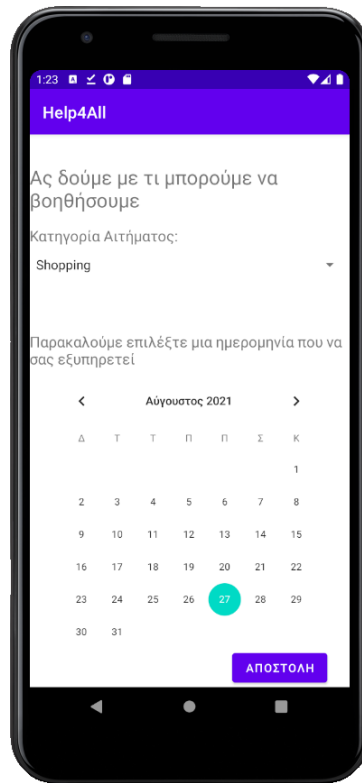
Σχήμα 20: Android Studio

4.3 Υποστήριξη μεταφράσεων

Τα εργαλεία ανάπτυξης για Android απευθύνονται σε προγραμματιστές εφαρμογών που μπορούν να λειτουργήσουν σε παγκόσμιο επίπεδο. Επομένως, είναι επιτακτική η ανάγκη δημιουργίας ενός μηχανισμού υποστήριξης πολλαπλών γλωσσών. Αυτή η δυνατότητα αποτελεί κομμάτι του Help4All όπως φαίνεται στις παρακάτω εκδοχές δύο κρίσιμων οθονών:

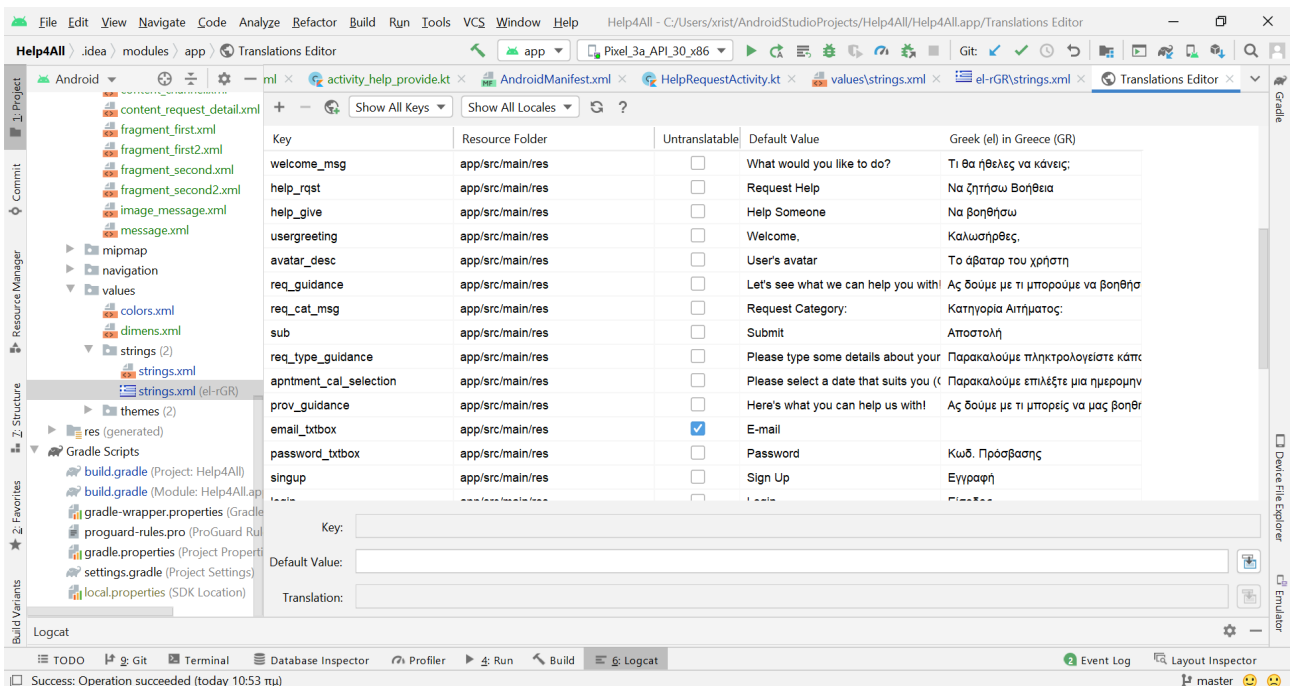


Σχήμα 21: Πρώτη βασική οθόνη στα Ελληνικά



Σχήμα 22: Δεύτερη βασική οθόνη στα Ελληνικά

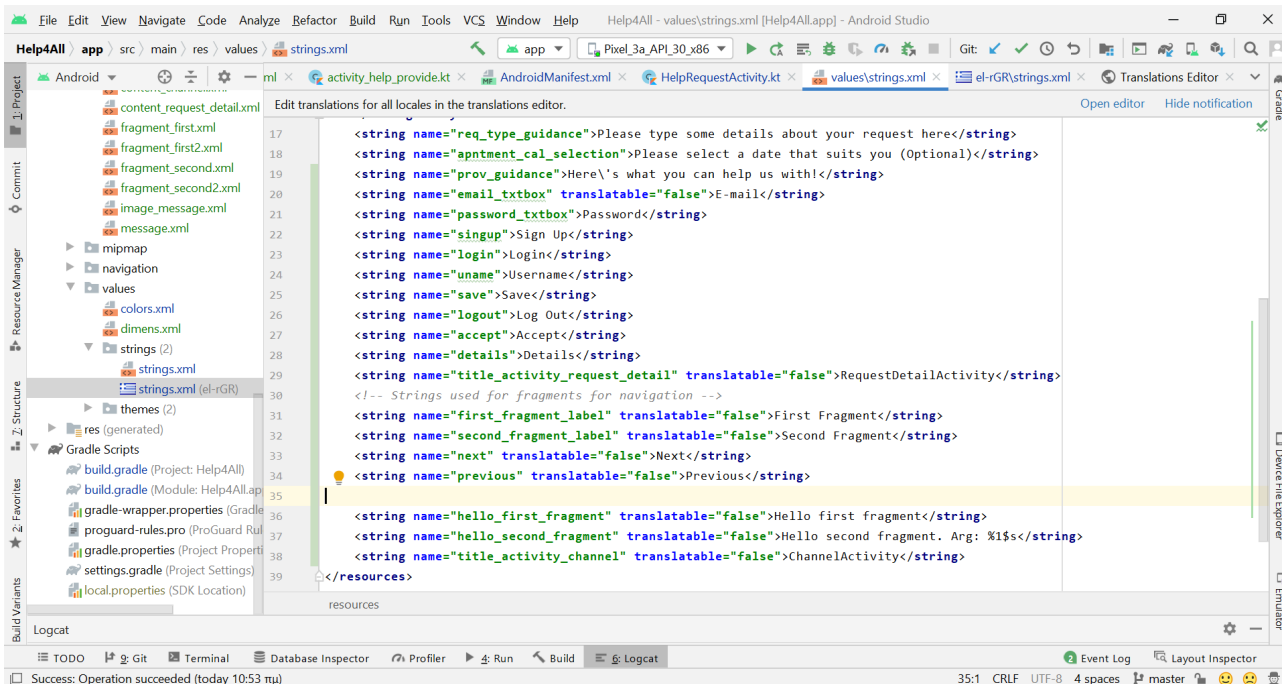
Η δυνατότητα προσαρμογής σε ένα σύνολο γλωσσών, γίνεται εύκολη καθώς για κάθε οπτικό αντικείμενο (όπως για παράδειγμα τα κουμπιά των παραπάνω οθονών) είναι εφικτή η ανάθεση μεταβλητού κειμένου σε αυτό. Παρέχεται από το Android Studio ο Translation Editor που δημιουργεί τέτοιες μεταβλητές εκφράσεις κειμένου κι έχει την εξής μορφή:



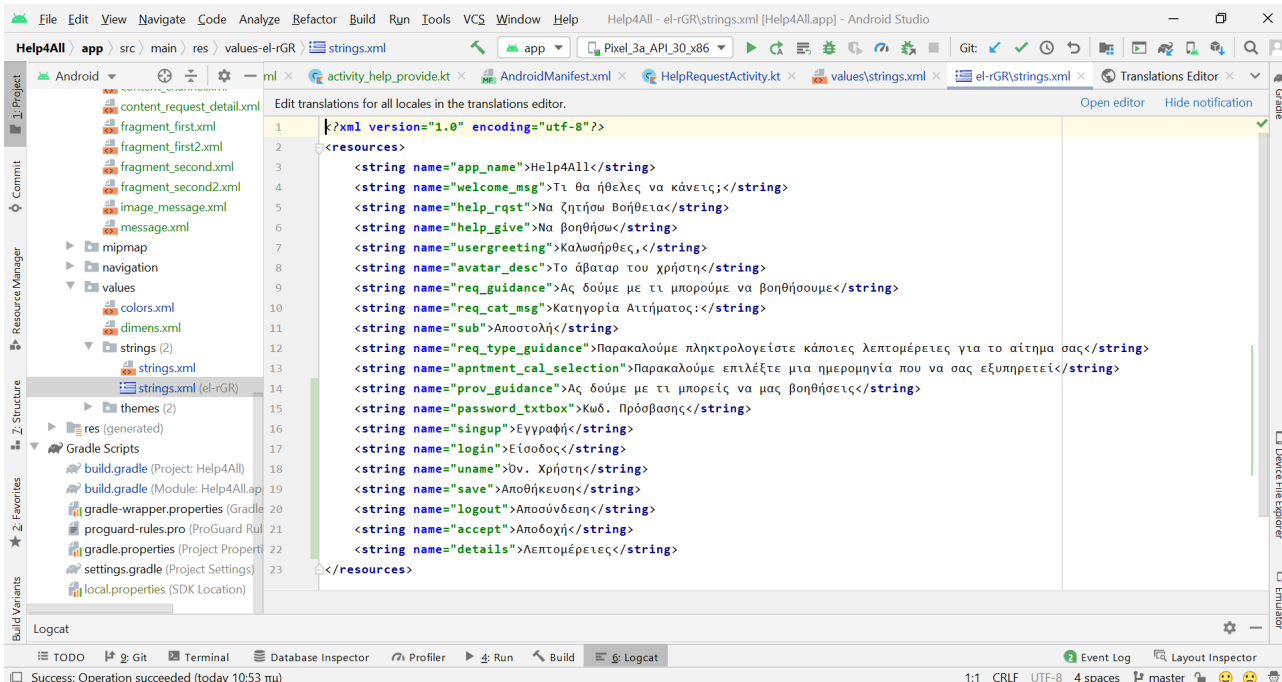
Σχήμα 23: Translations Editor

Παρατηρεί εύκολα κανείς, πως για κάθε κείμενο υπάρχει ένα **Όνομα-Κλειδί**, η προεπιλεγμένη (Αγγλική) τιμή του, και η μετάφρασή του στα Ελληνικά. Για πεδία που δεν επιδέχονται

μετάφρασης, υπάρχει το checkbox **Untranslatable**. Κατά τη διάρκεια της μεταγλώττισης, με βάση τον πίνακα αυτό, δημιουργούνται για κάθε γλώσσα αρχεία της μορφής *strings.xml* που κωδικοποιούν όλες αυτές τις πληροφορίες. Για το Help4All τα δύο αυτά αρχεία είναι τα εξής:



Σχήμα 24: strings.xml για την Αγγλική Γλώσσα

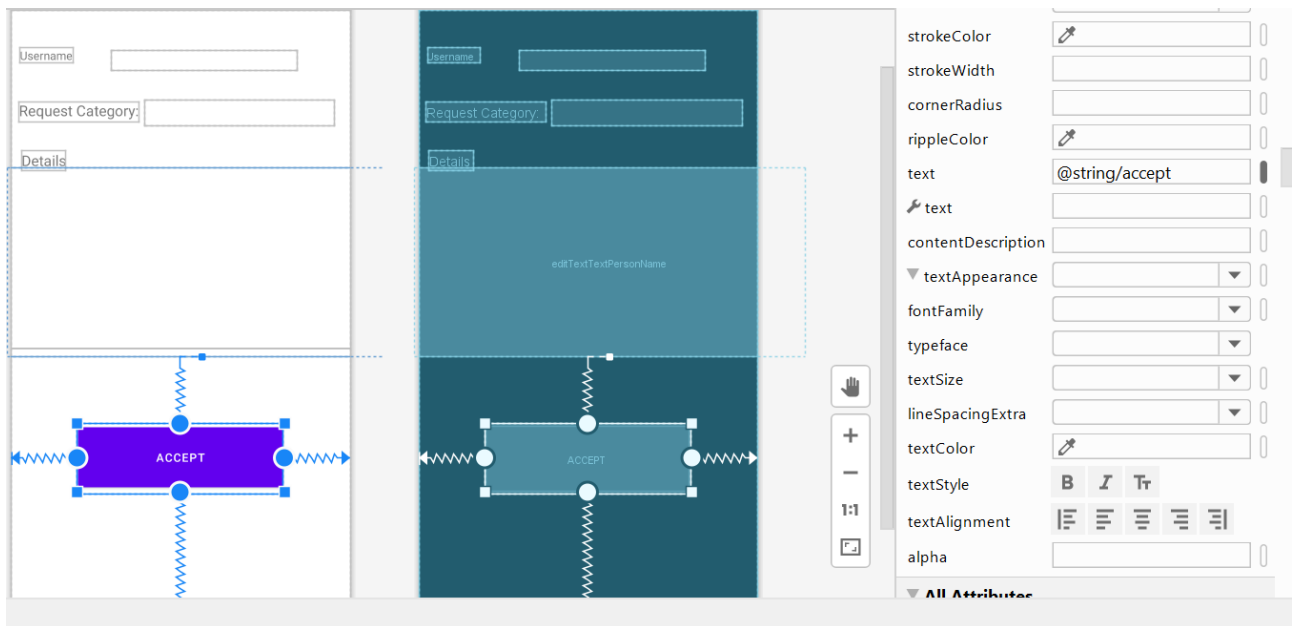


Σχήμα 25: strings.xml για την Ελληνική Γλώσσα

Για να δοθεί σε ένα οπτικό αντικείμενο το κείμενο μιας μεταβλητής από το string.xml τίθεται η ιδιότητα του κειμένου της στη μορφή:

`@string/<key>`

όπως φαίνεται και στην ακόλουθη εικόνα της οθόνης αποδοχής ενός αιτήματος βοήθειας:



Σχήμα 26: Ανάθεση πολύγλωσσου κειμένου

4.4 Πρότυπα λειτουργίας Android

Προκειμένου να λειτουργήσει όχι μόνο σωστά αλλά κι αποδοτικά μια εφαρμογή σε Android θα πρέπει να τηρεί πιστά ένα σύνολο κανόνων που προσδιορίζεται τόσο από την Google όσο κι από την έρευνα των επιστημόνων πληροφορικής και την ανάπτυξη της επιστήμης. [9] Το Help4All ακολουθεί πιστά τα πρότυπα αυτά, κι ένα υποσύνολο των κανόνων είναι το εξής:

- Η εφαρμογή πρέπει να διαμοιράζει τη λειτουργία της σε πολλαπλά νήματα.
- Το κύριο νήμα, θα έχει όσο το δυνατόν λιγότερο φόρτο εργασίας, κυρίως σχετιζόμενο με την τροποποίηση και χρήση του UI.
- Οι εργασίες που τρέχουν στο κύριο νήμα θα είναι ασύγχρονες και πιθανό σφάλμα σε αυτές δεν θα πρέπει να επηρεάζει την εφαρμογή.
- Αναγκαία είναι η προσαρμογή της εφαρμογής σε κάθε μεγέθους οθόνη κινητού ή ταμπλέτας.
- Η πλοήγηση της εφαρμογής είναι συμβατή με τα προεπιλεγμένα κουμπιά που παρέχει το λειτουργικό σύστημα
- Κάθε εφαρμογή υποστηρίζει την τελευταία έκδοση του λειτουργικού συστήματος

4.5 Μηχανισμοί ασφαλείας

Ακόμα κι αν το κείμενο αυτό γράφεται σε μια εποχή κυριαρχίας αυστηρών και αποτελεσματικών πρωτοκόλλων ασφαλείας για το διαδίκτυο (π.χ HTTPS) είναι αναγκαία η δημιουργία δικλιδών ασφαλείας για την προστασία της εφαρμογής από επιθέσεις που μπορούν να οδηγήσουν σε υποκλοπή δεδομένων ή/και καταστροφή της λειτουργίας της. Οι κύριοι μηχανισμοί προστασίας που θα αναλυθούν, επιλέχθηκαν καθώς αποτελούν παραδείγματα εργασίας που πρέπει οπωσδήποτε να πραγματοποιηθεί από τον προγραμματιστή μιας οποιασδήποτε τέτοιας εφαρμογής. Μάλιστα, η αποφυγή υλοποίησης τους μπορεί να εκθέσει ανεπανόρθωτα το πρόγραμμα και τον δημιουργό του, ακόμα κι αν αυτή βρίσκεται σε αρκετά προστατευμένο δικτυακό περιβάλλον.

4.5.1 Είσοδος με Firebase

Η είσοδος στην εφαρμογή γίνεται μέσω των μηχανισμών που παρέχει η Google για το Firebase. Απόρροια αυτού, είναι όχι μόνο η ασφαλής επικοινωνία με τους διακομιστές που φιλοξενούν τα δεδομένα των χρηστών αλλά και η ασφαλής φύλαξη των ευαίσθητων πληροφοριών για κάθε χρήστη. Παραδείγματος χάριν οι κωδικοί πρόσβασης φυλάσσονται ως hashes μη-αναστρέψιμων κρυπτογραφικών συναρτήσεων. Έτσι, ακόμη κι αν διαρρεύσουν στον Παγκόσμιο Ιστό, περιορίζεται σημαντικά η πιθανή ζημία που μπορεί να γίνει στην εφαρμογή.

4.5.2 Token authentication

Ακόμα, για τις περισσότερες πράξεις που γίνονται από τον εκάστοτε χρήστη της εφαρμογής, γίνονται αιτήματα σε κάποιον από τους εξυπηρετητές που επικοινωνούν με τις ΒΔ. Αυτά τα αιτήματα πρέπει να συνοδεύονται από κατάλληλες πληροφορίες που να πιστοποιούν την ταυτότητα του χρήστη, ώστε να γίνονται αποδεκτά. Σε αντίθετη περίπτωση, ένας επιτήδειος αντίπαλος θα μπορούσε να αναγκάσει την εφαρμογή να εκτελέσει διάφορες συναλλαγές δεδομένων χωρίς να έχει την απαραίτητη εξουσιοδότηση.

Το Firebase δημιουργεί ένα προσωρινό token που αντιστοιχεί στον χρήστη ο οποίος έχει συνδεθεί στην εφαρμογή. Με το ίδιο token εφοδιάζεται και ο εξυπηρετητής MySQL ο οποίος

αναλαμβάνει να πιστοποιήσει πριν από κάθε του ενέργεια την ταυτότητα του χρήστη που την αιτείται.

4.5.3 Προστασία από SQL Injections

4.5.3.1 Ο μηχανισμός

Μια κλασική επίθεση που κυριαρχεί δυστυχώς ακόμη στο διαδίκτυο είναι αυτή του SQL Injection [10]. Πρόκειται ουσιαστικά για την διοχέτευση κακόβουλου κώδικα SQL σε ένα κατά τα άλλα ασφαλές ερώτημα στη ΒΔ. Πιο συγκεκριμένα ας εξεταστεί το εξής υποθετικό query:

```
SELECT *
FROM credentials
WHERE username = ' <user input> '
```

Υπό κανονικές συνθήκες, θα έπρεπε να επιστραφούν τα στοιχεία εισόδου για κάθε δοθέν όνομα χρήστη. Ωστόσο, δίνοντας στο χρήστη πρόσβαση στον κώδικα του ερωτήματος, του επιτρέπεται η σύνταξη εκτελέσιμου κώδικα. Έστω, ότι ο χρήστης **Bob** δεν υπάρχει στη ΒΔ. Έτσι το εξής:

```
SELECT *
FROM credentials
WHERE username = 'Bob'
```

δεν θα επιστρέψει κάποια στοιχεία. Ωστόσο, αν ο επιτιθέμενος προσπαθήσει να κάνει είσοδο με το όνομα χρήστη:

Bob OR '1'='1'

θα σχηματιστεί ο κώδικας:

```
SELECT *
FROM credentials
WHERE username = 'Bob' OR '1'='1'
```

ο οποίος θα επιστρέψει **όλους τους χρήστες που βρίσκονται στην ΒΔ**.

Η οριστική λύση σε αυτή την πιθανή επίθεση δίνεται από τα **prepared statements** [11]. Πρόκειται για μια διαδικασία που χωρίζει κάθε ερώτημα στη ΒΔ σε δύο βήματα ως εξής:

1. Προετοιμασία εκτέλεσης του ερωτήματος
2. Σύγκριση των μεταβλητών κι εκτέλεσή του

Στο παραπάνω παράδειγμα η ΒΔ θα δεχόταν το ερώτημα ως εξής:

```
SELECT *
FROM credentials
WHERE username = ?
```

και θα προετοιμαζόταν εσωτερικά ώστε να κάνει απλά την σύγκριση της στήλης username με την είσοδο του χρήστη. Έτσι, ο χρήστης δεν βρίσκεται ποτέ σε πεδίο που μπορεί να παράξει εκτελέσιμο κώδικα κι έτσι υπάρχει πλήρης κάλυψη από αυτή την επίθεση. Ο PHP εξυπηρετητής του Help4All επικοινωνεί με τον MySQL server χρησιμοποιώντας αποκλειστικά αυτή τη μέθοδο.

4.5.3.2 Η υλοποίηση

Στις τελευταίες εκδόσεις της PHP οι διαδικασίες για την χρήση των prepared statements σε MySQL παρέχονται στον προγραμματιστή [12]. Η εκτέλεση κάθε επερωτήματος στη ΒΔ είναι μια διαδικασία τριών σταδίων:

1. Δημιουργία του query και προετοιμασία της ΒΔ με την κλήση της *mysqli_prepare()*
2. Αντιστοίχιση μεταβλητών με τις ελεύθερες εκφράσεις του query μέσω της *mysqli_stmt_bind_param()*
3. Εκτέλεση με την *mysqli_stmt_execute()*

Τα παραπάνω φαίνονται και στο εξής κομμάτι κώδικα, που βρίσκεται στον PHP εξυπηρετητή.

```
//run insert query using prepared statements to prevent sql injection
$sql = "INSERT INTO Request (req_username, req_type, req_comm, req_date, req_latitude, req_longitude)
VALUES (?, ?, ?, NOW(), ?, ?)";
if($stmt = mysqli_prepare($link, $sql)){
    // Bind variables to the prepared statement as parameters
    mysqli_stmt_bind_param($stmt, "sssd", $req_username,$req_type,$req_com,$req_lat,$req_long);
    // Attempt to execute the prepared statement
    if(mysqli_stmt_execute($stmt)){
        echo "Success!";
    }
    else {
        die("ERROR:".mysqli_error($link));
    }
}
```

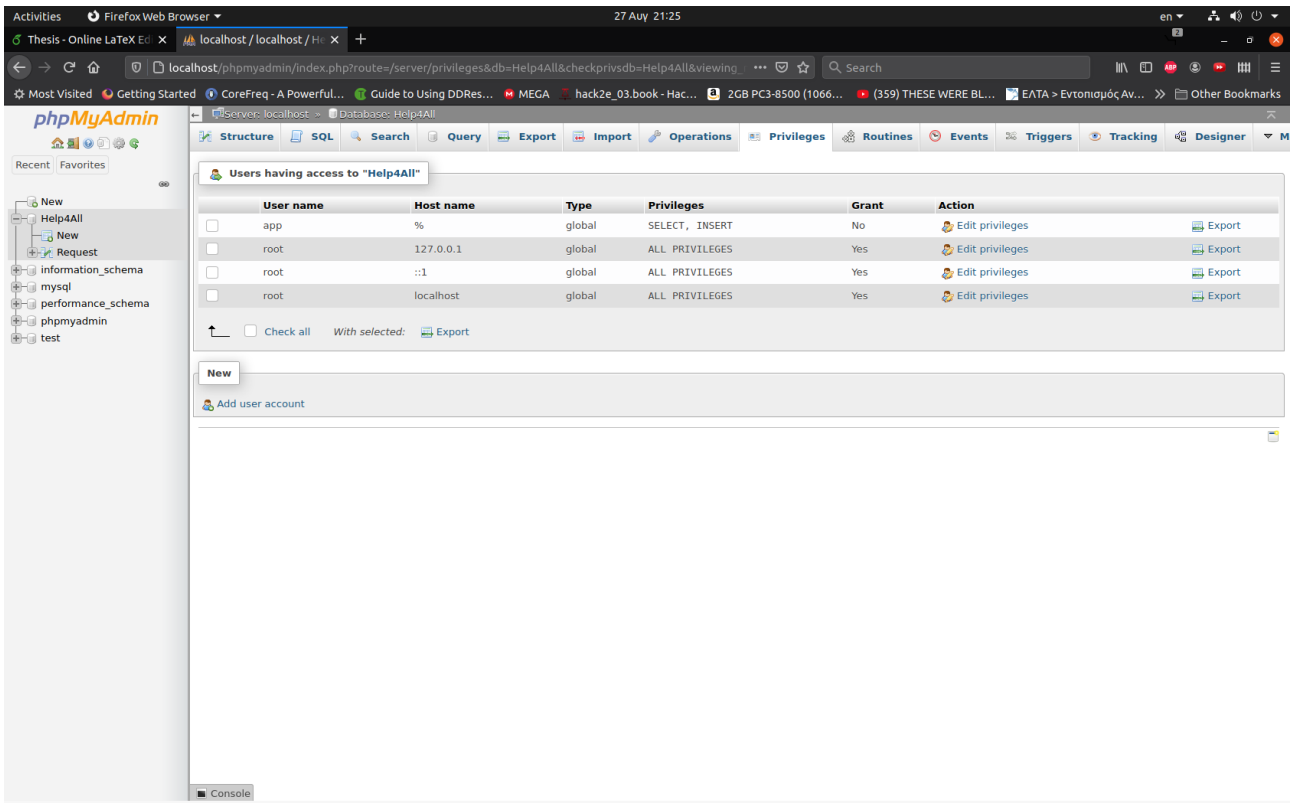
Σχήμα 27: PHP server - Μέθοδος εισαγωγής Αιτήματος Βοήθειας

4.5.4 Δικαιώματα χρήστη MySQL

Τέλος, για να διασφαλιστεί η ελαχιστοποίηση κάθε πιθανής ζημιάς σε περίπτωση εισόδου ενός κακόβουλου αντιπάλου στο σύστημα, κάθε επικοινωνία με τις ΒΔ, γίνεται από λογαριασμούς χρηστών με περιορισμένα δικαιώματα. Έτσι ακόμη κι αν με κάποιο τρόπο φτάσει στη ΒΔ μια εντολή της μορφής:

```
DROP TABLE <table name>
```

ή κάποια αντίστοιχη που τροποποιεί δομικά στοιχεία της ΒΔ, θα απορριφθεί αυτόματα. Οι χρήστες και τα αντίστοιχα δικαιώματά τους δημιουργούνται εύκολα με την παρακάτω οθόνη του rhrmyadmin



Σχήμα 28: phpmyadmin - Πίνακας χρηστών ΒΔ

5 ΕΠΙΛΟΓΟΣ

5.1 Συμπεράσματα

Επιλογικά, φανερώνονται κάποια βασικά συμπεράσματα μέσα από την εκπόνηση της παρούσας πτυχιακής εργασίας. Γίνεται αντιληπτό τόσο το μέγεθος της δυσκολίας της ανάπτυξης μιας πολυδιάστατης εφαρμογής, όσο και της συντήρησης/προστασίας της από τους πιθανούς κινδύνους της έκθεσης της στο ευρύ κοινό. Επιβαρυντικός παράγοντας στην περιοχή της συντήρησης είναι και η διαρκής ανάπτυξη του λειτουργικού συστήματος Android, κάνοντας τις ήδη αναφερθείσες απαιτήσεις που επιβάλλεται να πληροί η εφαρμογή, ένα σχεδόν ανυπέρβλητο σε ατομικό επίπεδο εμπόδιο για ένα πρόγραμμα που θα χαρακτηρίζεται εμπορικά αντάξιο εφάμιλλων άλλων. Ωστόσο ο προγραμματισμός μιας εφαρμογής που μπορεί εν δυνάμει να προσφέρει μια μικρή βοήθεια σ' ένα σύνολο ανθρώπων, δημιουργεί ίσως τα πιο όμορφα συναισθήματα. Άλλωστε, για κάθε προγραμματιστή, το όριο τού τί μπορεί να κατασκευάσει το ορίζει μόνο η φαντασία του!

5.2 Μελλοντικές ιδέες

Σίγουρα το Help4All αποτελεί μια εφαρμογή που μπορεί να αποκτήσει πολλές ακόμη λειτουργίες και να αναπτυχθεί περαιτέρω. Το επόμενο βήμα για την ανάπτυξή της θα είναι ένα παράλληλο Web περιβάλλον, ώστε οι χρήστες να μην περιορίζονται μόνο σε mobile εφαρμογές. Ύστερα μπορούν να ακολουθήσουν επιπλέον δυνατότητες όπως:

- Ηχητικά μηνύματα στις συνομιλίες
- Υποστήριξη φωνητικής κλήσης και βιντεοκλήσης
- Διαμόρφωση του προφίλ χρήστη ώστε να προτιμούνται συγκεκριμένα αιτήματα
- Σύστημα επιβράβευσης χρηστών με πόντους, προκειμένου να γίνεται ελκυστικό για κάποιον να προσφέρει
- Δυνατότητα δικτύωσης με άτομα που αλληλοεξυπηρετούνται συχνά μέσω του Help4All
- Ενσωμάτωση λογαριασμών κοινωνικών δικτύων (π.χ Είσοδος στην εφαρμογή μέσω Facebook)

ΣΥΝΤΜΗΣΕΙΣ - ΑΡΚΤΙΚΟΛΕΞΑ - ΑΚΡΩΝΥΜΙΑ

API	Application Programming Interface
DT	Data Type
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
JSON	Javascript Object Notation
PHP	PHP: Hypertext Preprocessor
SQL	Structured Query Language
SSL	Secure Sockets Layer
TLS	Transport Layer Security
XML	Extensible Markup Language
ΒΔ	Βάση Δεδομένων
ΕΚΠΑ	Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
ΛΣ	Λειτουργικό Σύστημα

ΑΝΑΦΟΡΕΣ

- [1] ECMA International. The json data interchange standard.
- [2] Google. Activity : Android developer.
- [3] R. Fielding et al. Hypertext transfer protocol (http/1.1): Message syntax and routing (RFC7230).
- [4] Google. Network security configuration : Android developer.
- [5] OpenSSL Software Foundation. Openssl manpage.
- [6] Google. Kotlin : Android developer.
- [7] Antonio Leiva. Data classes in kotlin: save a good bunch of lines of code.
- [8] Google. Android studio : Android developer.
- [9] Google. Material design guidelines : Android developer.
- [10] OWASP Foundation. Sql injection.
- [11] Oracle Corporation. Mysql reference manual : Prepared statements.
- [12] The PHP Group. Php manual : Mysql prepared statements.