



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**  
**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

*Εφαρμογές φορητών συσκευών για συνεργατική διαχείριση  
εκτάκτων περιστάσεων*

**ΙΩΑΝΝΙΔΗΣ ΠΕΤΡΟΣ , 71444277**

**ΕΠΙΒΛΕΠΟΥΣΑ ΚΑΘΗΓΗΤΡΙΑ : ΕΛΕΝΗ ΑΙΚΑΤΕΡΙΝΗ ΛΕΛΙΓΚΟΥ**

**ΑΙΓΑΛΕΩ**

**ΙΟΥΝΙΟΣ 2021**

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Ιωαννίδης Πέτρος του Βαγγέλη, με αριθμό μητρώου 71444277 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Βιομηχανικής Σχεδίασης και Παραγωγής, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών



ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Ε. Α. Λελίγκου

Π. Καρκαζής

Χ. Δρόσος

## *Ευχαριστίες*

Αρχικά, θα ήθελα να ευχαριστήσω την επιβλέπουσα καθηγήτρια κα. Ελένη Αικατερίνη Λελίγκου για την άψογη συνεργασία και επικοινωνία που υπήρξε, ώστε να μπορέσω να ανταπεξέλθω στην ολοκλήρωση της διπλωματικής μου εργασίας, πάνω στο αντικείμενο που αγαπώ, τον προγραμματισμό.

Επίσης θα ήθελα να ευχαριστήσω θερμά την οικογένειά μου και τους φίλους μου και ιδιαίτερα την μητέρα μου, για την υποστήριξη και το κουράγιο που μου έδωσαν κατά την περίοδο αυτή.

## Περίληψη

Οι ηλεκτρονικοί υπολογιστές και οι φορητές συσκευές τα τελευταία χρόνια αποτελούν αναπόσπαστο κομμάτι της ζωής του ανθρώπου και αποτελούν εργαλεία καθημερινής χρήσης. Τα κινητά τηλέφωνα ξεκίνησαν ως αντικείμενα επικοινωνίας και στην πορεία μετεξελίχθηκαν σε έξυπνα κινητά (smartphones) τα οποία πλέον προσφέρουν άπειρες λειτουργίες για τον χρήστη. Διαθέτουν ισχυρά λειτουργικά συστήματα τα οποία παρέχουν μια μεγάλη γκάμα εφαρμογών, από την οποία μπορεί να επιλέξει ο χρήστης, να αγοράσει ή να κατεβάσει στο κινητό του. Κάποιες απ' αυτές τις εφαρμογές, είναι ζωτικής σημασίας και προστατεύουν τον άνθρωπο από απροσδόκητα γεγονότα.

Εφόσον υπάρχει τεράστια ποικιλία εφαρμογών, οφείλει ο καθένας να επιλέξει και κάποιες αναγκαίες εφαρμογές που θα μπορούσαν να του σώσουν την ζωή, είτε από φυσικές καταστροφές που μπορεί να προκληθούν από έντονα καιρικά φαινόμενα, είτε από ατυχήματα στον δρόμο. Επίσης η συνεργασία και αλληλεπίδραση των χρηστών σε συνάρτηση με την συνεχή ενημέρωση μπορεί να βοηθήσει στην αποφυγή δυσάρεστων γεγονότων και να μετατρέψει το κινητό σε προστατευτικό εργαλείο ενημέρωσης και συνεργασίας. Όσο αυξάνεται η επικινδυνότητα εμφάνισης ακραίων καιρικών φαινομένων καθώς και η αμέλεια και αδιαφορία των ανθρώπων για το περιβάλλον και την κοινωνία, τόσο θα αυξάνεται και η ζήτηση και ανάγκη σχεδιασμού και ανάπτυξης τέτοιου είδους εφαρμογών.

Για τον λόγο αυτό καλούμαστε να σχεδιάσουμε μια τέτοια εφαρμογή η οποία θα μπορέσει να αντιμετωπίσει έκτακτες καταστάσεις με την πλήρη συνεργασία των χρηστών της.

## Abstract

Computers and mobile devices in recent years are an integral part of human life and are tools of everyday use. Mobile phones started as objects of communication and in the process evolved into smart phones (smartphones) which now offer infinite functions for the user. They have powerful operating systems that offer a wide range of applications, from which the user can choose to buy or download on his mobile. Some of these applications are vital and protect people from unexpected events.

Since there is a huge variety of applications, everyone should choose some necessary applications where they could save their life either from natural disasters that can be caused by severe weather or road accidents. Also, the cooperation and interaction of users with each other in connection with the constant information can help to avoid unpleasant events and turn the mobile phone into a protective tool of information and cooperation. As the risk of extreme weather events increases as well as people's negligence and indifference to the environment and society, so will the demand and need to design and develop such applications.

For this reason, we are called to design such an application that will be able to deal with emergencies with the full cooperation of its users.

## ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη.....	5
Abstract .....	6
<b>ΠΡΟΛΟΓΟΣ</b> .....	9
<b>ΚΕΦΑΛΑΙΟ 1</b> .....	10
1.1 Η Τεχνολογία σήμερα.....	10
1.2 Φορητές συσκευές και λειτουργικά συστήματα .....	11
1.3 Η Ανάπτυξη εφαρμογών με βάση την εξέλιξη της τεχνολογίας .....	12
1.4 Κατηγορίες εφαρμογών .....	13
<b>ΚΕΦΑΛΑΙΟ 2</b> .....	17
2.1 Συνεργατικές εφαρμογές .....	17
2.2 Εφαρμογές Risk Management.....	18
2.2.1 Συνεργατικές εφαρμογές στο πεδίο του Risk Management.....	18
2.2.2 Emergency Alert Apps για φυσικές καταστροφές από ακραία καιρικά φαινόμενα ...	19
<b>ΚΕΦΑΛΑΙΟ 3</b> .....	25
3.1 Η εφαρμογή.....	25
3.1.1 Περιγραφή Κεντρικής Ιδέας της εφαρμογής.....	25
3.2 Τεχνικά Χαρακτηριστικά.....	26
3.3 Ο τίτλος της εφαρμογής – “FastReport” .....	26
3.4 Που απευθύνεται .....	27
3.5 Στόχος της εφαρμογής .....	27
3.6 Διάγραμμα ροής.....	27
3.7 Περιγραφή οθονών εφαρμογής.....	28
3.8 Αρχιτεκτονική εφαρμογής και βάση δεδομένων.....	34
<b>ΚΕΦΑΛΑΙΟ 4</b> .....	40
4.1 Συμπεράσματα .....	40
4.2 Μελλοντικές προσθήκες και αλλαγές .....	40
<b>ΑΝΑΦΟΡΕΣ</b> .....	41
<b>ΠΑΡΑΡΤΗΜΑ Ι</b> .....	42

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1 - ΧΡΗΣΤΕΣ ΚΙΝΗΤΩΝ.....	9
Εικόνα 2 - Η ΕΞΕΛΙΞΗ ΤΩΝ ΚΙΝΗΤΩΝ.....	11
Εικόνα 3 - ΟΙΚΟΝΟΜΙΚΕΣ ΖΗΜΙΕΣ ΑΠΟ ΦΥΣΙΚΕΣ ΚΑΤΑΣΤΡΟΦΕΣ, 1900 – 2019 .....	19
Εικόνα 4 - ΠΑΓΚΟΣΜΙΕΣ ΑΝΑΦΟΡΕΣ ΦΥΣΙΚΩΝ ΚΑΤΑΣΤΡΟΦΩΝ ΜΕ ΒΑΣΗ ΤΟΝ ΤΥΠΟ ΤΟΥΣ, 1970 – 2019 .....	20
Εικόνα 5 - ΠΑΓΚΟΣΜΙΟΣ ΑΡΙΘΜΟΣ ΠΡΟΣΒΕΒΛΗΜΕΝΩΝ ΑΠΟ ΦΥΣΙΚΕΣ ΚΑΤΑΣΤΡΟΦΕΣ, 1900 – 2019.....	21
Εικόνα 6 - ΠΑΓΚΟΣΜΙΟΣ ΑΡΙΘΜΟΣ ΘΑΝΑΤΩΝ ΑΠΟ ΦΥΣΙΚΕΣ ΚΑΤΑΣΤΡΟΦΕΣ, 1900- 2019....	22
Εικόνα 7 - LOADING SCREEN .....	28
Εικόνα 8 - ΕΙΣΟΔΟΣ ΧΡΗΣΤΗ .....	28
Εικόνα 9 - ΕΓΓΡΑΦΗ ΧΡΗΣΤΗ .....	29
Εικόνα 10 - ΕΠΙΤΥΧΗΣ ΕΓΓΡΑΦΗ .....	29
Εικόνα 11 - ΑΡΧΙΚΗ ΟΘΟΝΗ .....	30
Εικόνα 12 - ΕΙΣΟΔΟΣ ΧΡΗΣΤΗ .....	30
Εικόνα 13 - ΦΟΡΜΑ ΑΝΑΦΟΡΑΣ.....	31
Εικόνα 14 - ΧΑΡΤΕΣ.....	31
Εικόνα 15 - NOTIFICATIONS.....	32
Εικόνα 16 - ΣΥΜΠΛΗΡΩΣΗ ΑΝΑΦΟΡΑΣ.....	32
Εικόνα 17 - REPORTS .....	33
Εικόνα 18 - REPORTS .....	33
Εικόνα 19 - ΡΥΘΜΙΣΕΙΣ ΚΑΙ ΚΟΥΜΠΙ ΕΞΟΔΟΥ .....	34
Εικόνα 20 - ΕΓΓΕΓΡΑΜΜΕΝΟΙ ΧΡΗΣΤΕΣ .....	38
Εικόνα 21 - ΑΝΑΦΟΡΕΣ ΧΡΗΣΤΩΝ.....	38
Εικόνα 22 - ΟΙ ΦΩΤΟΓΡΑΦΙΕΣ ΓΙΑ ΤΙΣ ΑΝΑΦΟΡΕΣ .....	39

## ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ

Διάγραμμα 1 - ΑΡΙΘΜΟΣ ΧΡΗΣΤΩΝ ΕΦΑΡΜΟΓΗΣ.....	14
Διάγραμμα 2 - ΛΗΨΕΙΣ ΠΑΙΧΝΙΔΙΩΝ .....	15
Διάγραμμα 3 – ΓΕΝΙΚΟ ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ .....	27

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

ΣΧΗΜΑ 1 – ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΦΑΡΜΟΓΗΣ.....	35
--	----



## ΠΡΟΛΟΓΟΣ

Τα κινητά τηλέφωνα μπορούν πλέον να χαρακτηριστούν και ως “προέκταση του χεριού του ανθρώπου”, λόγω του ότι τα χρησιμοποιούμε καθημερινά, σχεδόν για όλες τις ανάγκες μας. Σύμφωνα με μια έρευνα που έγινε το 2017, το 67% του παγκόσμιου πληθυσμού χρησιμοποιεί κινητά τηλέφωνα και tablets [1]. Οι φορητές αυτές συσκευές με την σειρά τους, προσφέρουν εφαρμογές από τις οποίες μπορεί να επιλέξει ο άνθρωπος να έχει στην κατοχή του. Η ζήτηση νέων, καινοτόμων και χρήσιμων εφαρμογών αυξάνεται ολοένα και περισσότερο και αποτελεί ζητούμενο της παρούσας διπλωματικής, την δημιουργία δηλαδή μιας χρήσιμης εφαρμογής που απαιτεί την αλληλεπίδραση μεταξύ των χρηστών της, για την άμεση ενημέρωση, διαχείριση και αντιμετώπιση εκτάκτων καταστάσεων.



ΕΙΚΟΝΑ 1 - ΧΡΗΣΤΕΣ ΚΙΝΗΤΩΝ

## ΚΕΦΑΛΑΙΟ 1

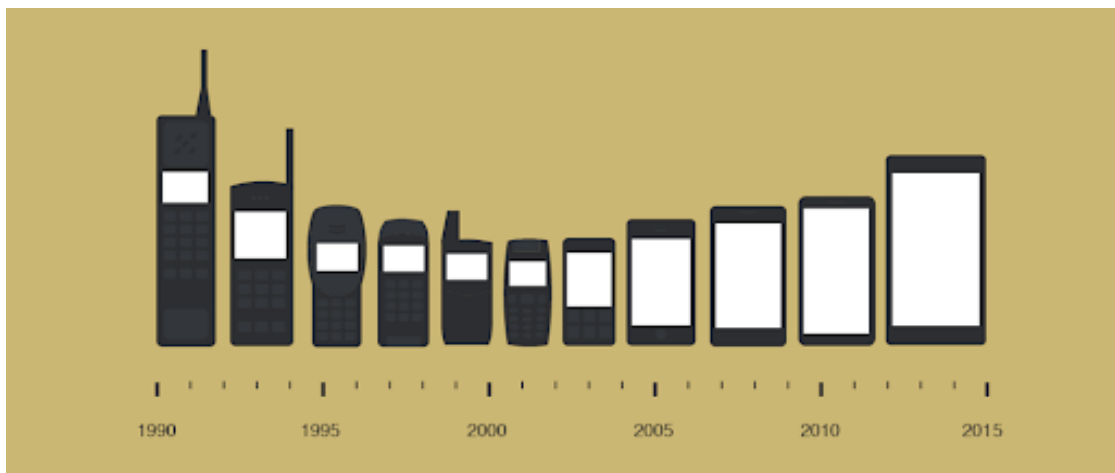
### 1.1 Η Τεχνολογία σήμερα

Τα τελευταία χρόνια η τεχνολογία σημειώνει ραγδαία εξέλιξη η οποία έχει επηρεάσει πολλούς τομείς, όπως είναι ο τομέας της ιατρικής, των ανανεώσιμων πηγών ενέργειας, της πληροφορικής κλπ. Η ιστορία της τεχνολογίας ξεκινά από τα αρχαία χρόνια με την επινόηση διάφορων εργαλείων και τεχνικών για την εξυπηρέτηση των ανθρώπων. Με την εμφάνιση του διαδικτύου το 1993 από το CERN στην Ελβετία, ήρθε και η επανάσταση της τεχνολογίας που μέχρι και σήμερα βιώνουμε [2]. Πλέον μπορούμε να έχουμε στα χέρια μας πολύ ισχυρές συσκευές με διάφορες λειτουργίες όπως είναι τα smartphones, tablets, smartwatches αλλά και υπολογιστές και laptops όπου μας κάνουν την ζωή πιο εύκολη, καλύτερη, γρηγορότερη και πιο διασκεδαστική. Ένα μεγάλο θετικό της εξέλιξης της τεχνολογίας είναι πως πλέον είναι προσιτή και πιο εύχρηστη για όλες τις ηλικίες, γεγονός το οποίο σημαίνει ότι τα παιδιά μπορούν να χρησιμοποιήσουν τους Η/Υ και τα κινητά τους τηλέφωνα για εκπαιδευτικούς σκοπούς(π.Χ. εύρεση πληροφοριών στο internet, εκπαιδευτικά παιχνίδια, online βιβλία και άρθρα) αλλά και τα άτομα μεγαλύτερης ηλικίας για θέματα υγείας, εύκολης επικοινωνίας με συγγενείς κλπ.

Για να γίνει περισσότερο αντιληπτή η επιρροή της τεχνολογίας σήμερα, αξίζει να σημειωθεί ότι ο παγκόσμιος αριθμός ενεργών χρηστών που χρησιμοποιούν το διαδίκτυο είναι περίπου 3,2 δισεκατομμύρια. Επίσης περισσότερα από 2 εκατομμύρια smartphones πωλούνται καθημερινά παγκοσμίως [3]. Τα στατιστικά αυτά δείχνουν το πόσο σημαντική είναι η επιρροή της τεχνολογίας σήμερα στην κοινωνία και το πως βαδίζουν οι άνθρωποι με την εξέλιξή της.

## 1.2 Φορητές συσκευές και λειτουργικά συστήματα

Με την έννοια των φορητών συσκευών το πρώτο πράγμα που μας έρχεται στο μυαλό είναι τα κινητά τηλέφωνα. Στις 3 Απριλίου του 1973 οι Αμερικανοί John F. Mitchell και Martin Cooper δημιούργησαν το πρώτο κινητό τηλέφωνο, το Motorola DynaTAC 8000x, 33 εκατοστών και ενός(1) κιλού το οποίο 10 χρόνια μετά κυκλοφόρησε στην αγορά για εμπορική χρήση [4].



ΕΙΚΟΝΑ 2 - Η ΕΞΕΛΙΞΗ ΤΩΝ ΚΙΝΗΤΩΝ

Στα πρώτα χρόνια κυκλοφορίας τους στην αγορά, τα κινητά τηλέφωνα διέθεταν τις απαραίτητες λειτουργίες, όπως είναι οι κλήσεις, τα μηνύματα, ημερομηνία και αριθμομηχανή. Μόλις σε 25 χρόνια όμως η εξέλιξή τους σημείωσε σταθερή ανοδική πορεία και οι απαιτήσεις των χρηστών γίνονταν ολοένα και περισσότερες με αποτέλεσμα σήμερα να μπορούμε να κάνουμε σχεδόν τα πάντα μέσω του κινητού μας. Πλέον πέρα από τις βασικές λειτουργίες, μπορούμε ακόμα να βγάλουμε φωτογραφίες και βίντεο σε απίστευτα υψηλή ευκρίνεια, να σερφάρουμε στο internet και με την βοήθεια των λειτουργικών συστημάτων που υποστηρίζουν, να κατεβάσουμε όλων των ειδών τις εφαρμογές.

Βέβαια στο πεδίο των φορητών συσκευών δεν υπάρχουν μόνο τα κινητά τηλέφωνα, αλλά συμπεριλαμβάνονται και τα tablets, mp3/mp4, smartwatch κλπ. Παρόλα αυτά εμείς θα επικεντρωθούμε στα κινητά τηλέφωνα και tablets αφού και η εφαρμογή της διπλωματικής εργασίας έχει αναπτυχθεί σε android περιβάλλον που υποστηρίζουν μόνο κινητά και tablets.

Ο πυρήνας λειτουργίας μιας φορητής συσκευής είναι το λειτουργικό σύστημα. Το λειτουργικό σύστημα είναι υπεύθυνο για την σωστή και αρμονική λειτουργία του υπολογιστή/κινητού και για την επικοινωνία μας με αυτό. Είναι το μέσον που μας

επιτρέπει να εκτελέσουμε προγράμματα και χωρίς αυτό η μηχανή δεν θα μπορούσε να καταλάβει εμάς ούτε εμείς αυτή. Στον τομέα των φορητών συσκευών δύο είναι τα κυρίαρχα εμπορικά λειτουργικά συστήματα, το Android και iOS.

Το Android είναι μια τροποποιημένη έκδοση του Linux και χρησιμοποιείται από εταιρίες κινητών όπως είναι τα Samsung, Huawei, Xiaomi. Το iOS είναι το λειτουργικό σύστημα της Apple και χρησιμοποιείται από τα iPhone. Ανεξαρτήτως κατασκευαστή, τα λειτουργικά συστήματα έχουν τον ίδιο σκοπό, κι αυτός είναι να επιτύχουν την ομαλή λειτουργία και επικοινωνία του χρήστη με την συσκευή καθώς και την προσφορά μιας τεράστιας ποικιλίας εφαρμογών για την διευκόλυνση και διασκέδαση του χρήστη.

### 1.3 Η Ανάπτυξη εφαρμογών με βάση την εξέλιξη της τεχνολογίας

Ας κάνουμε μια ιστορική αναδρομή στην εξέλιξη των εφαρμογών των φορητών συσκευών και πιο συγκεκριμένα πηγαίνοντας πίσω στο 1983 όπου ο Steve Jobs οραματίστηκε την δημιουργία μίας ψηφιακής πλατφόρμας εφαρμογών, όπου ο καθένας θα μπορούσε μέσω αυτής, να αγοράσει εφαρμογές από το κινητό του τηλέφωνο, γνωστή σήμερα ως App Store.

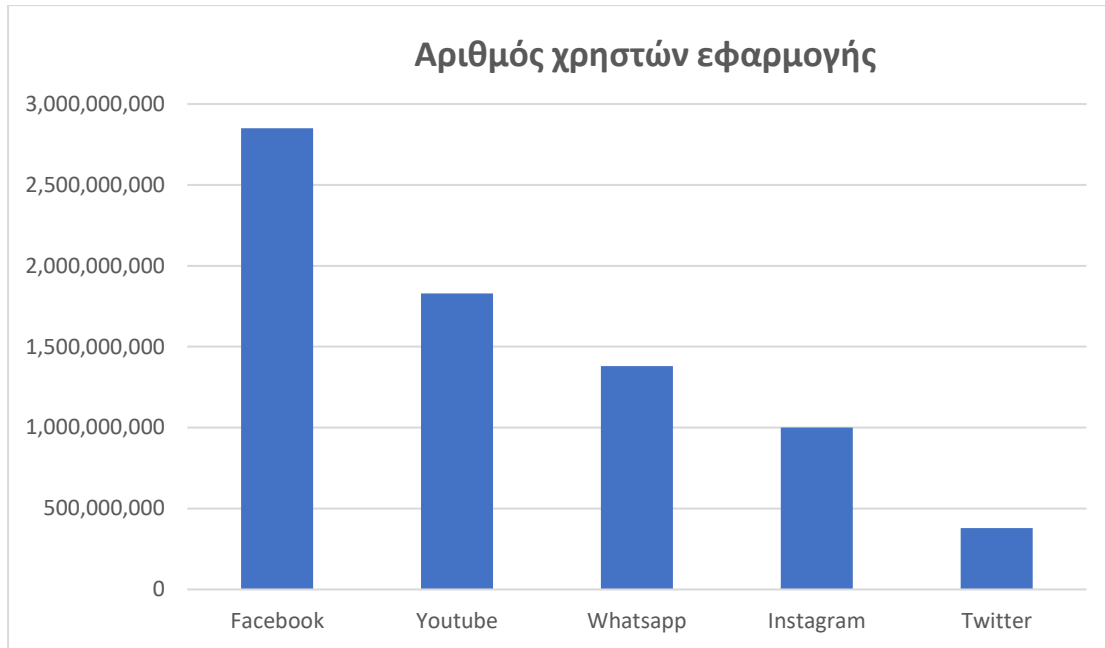
Στις 10 Ιουλίου του 2008, μαζί με την επανάσταση της τεχνολογίας έρχεται και η κυκλοφορία του Apple App Store προσφέροντας 500 εφαρμογές. Στην συνέχεια με την πρόοδο των smartphones και με την αναγνώριση της επιτυχίας του Apple App Store ακολούθησαν κι άλλες εταιρίες με διαφορετικά ψηφιακά καταστήματα όπως είναι το γνωστό σε όλους μας Google Play, Blackberry's App World και Amazon App Store με απώτερο σκοπό την δυνατότητα απόλαυσης των εφαρμογών από όλους τους χρήστες ανεξαρτήτως λειτουργικού συστήματος. Επίσης από το 2008 μέχρι και σήμερα υπάρχει τεράστια γκάμα εφαρμογών και σε όλα τα ψηφιακά καταστήματα υπάρχουν είτε δωρεάν εφαρμογές είτε εφαρμογές επί πληρωμή [5].

## 1.4 Κατηγορίες εφαρμογών

Οι φορητές συσκευές μαζί με τις εφαρμογές που προσφέρουν, αποτελούν πλέον αναπόσπαστο κομμάτι της ζωής του ανθρώπου. Ποιος θα φανταζόταν πριν 30 χρόνια ότι με ένα κινητό τηλέφωνο μεγέθους τσέπης, θα μπορεί ο άνθρωπος να επικοινωνήσει μέσω live chatting, θα αγοράζει προϊόντα, θα διαβάζει βιβλίο online, θα μπορεί να δει μια ταινία, θα μπορεί να πάει σε οποιονδήποτε προορισμό με GPS, ακόμα και να ελέγξει το ίδιο του το σπίτι από απόσταση. Κανείς βέβαια δεν θα μπορούσε να προβλέψει τέτοια εξέλιξη και αυτή πλέον μας καθιστά δυνατόν να χωρίσουμε όλες αυτές τις λειτουργίες και εφαρμογές σε κατηγορίες.

Θα ξεκινήσουμε λοιπόν από τα γνωστά σε όλους μας, Μέσα Κοινωνικής δικτύωσης ή αλλιώς Social Media. Όλος ο κόσμος γνωρίζει το Facebook, Twitter, Instagram κι αν μην είναι απαραίτητα χρήστες αυτών, παρόλα αυτά αξίζει να σημειωθεί ότι το Facebook απασχολεί 2.85 δισεκατομμύρια χρήστες, το Twitter 290.5 εκατομμύρια και το Instagram 1 δισεκατομμύριο χρήστες. Σκοπός αυτών των εφαρμογών είναι η δημιουργία ενός παγκόσμιου δικτύου επικοινωνίας των ανθρώπων μέσω ανάρτησης βίντεο και φωτογραφιών καθώς τους δίνεται και η δυνατότητα δημοσίευσης κειμένου και live chatting. Τέτοιου είδους εφαρμογές έχουν την μεγαλύτερη απήχηση έως και σήμερα και θεωρούνται πλέον ως “απαραίτητες” εφαρμογές που ο καθένας θα πρέπει να έχει στο κινητό του.

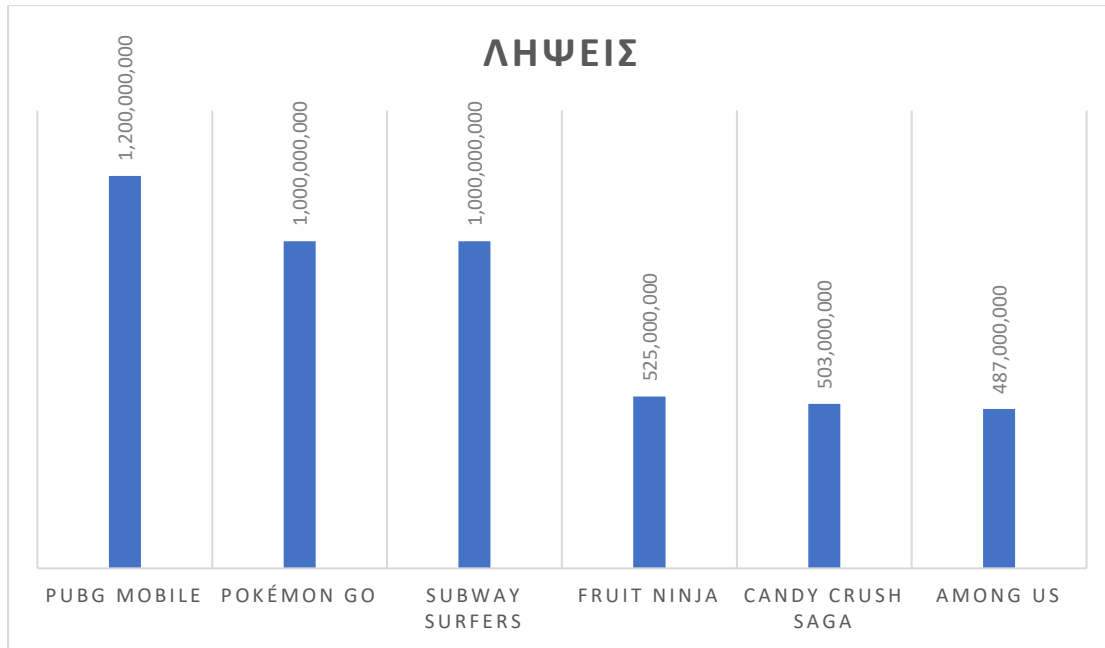
Συνεχίζουμε την κατηγοριοποίηση των εφαρμογών με τα Messaging Apps, όπως είναι το WhatsApp, το Messenger, το Viber, το Skype και άλλα. Αυτές οι εφαρμογές ήρθαν να αντικαταστήσουν τα παραδοσιακά SMS και κλήσεις των παλιών κινητών, με έναν πιο μοντέρνο και άμεσο σχεδιασμό προσφέροντας και περισσότερες λειτουργίες. Με τις προαναφερθείσες εφαρμογές, ο χρήστης, όχι μόνο μπορεί να στείλει ένα μήνυμα ή να κάνει μια κλήση αλλά μπορεί επίσης είτε να στείλει ένα φωνητικό μήνυμα είτε να κάνει μια βίντεο-κλήση είτε να στείλει μια φωτογραφία ή βίντεο, ακόμα και να κάνει πιο διασκεδαστική την συνομιλία με την προσθήκη emoticons σ’ αυτή. Βασική προϋπόθεση για να χρησιμοποιήσει ο χρήστης αυτές τις λειτουργίες, είναι να βρίσκεται συνδεδεμένος σε κάποιο δίκτυο. Αυτή η κατηγορία όπως και η προηγούμενη αποτελούν τις δημοφιλέστερες κατηγορίες εφαρμογών με τους περισσότερους χρήστες όπως φαίνεται και στο παρακάτω διάγραμμα.



**ΔΙΑΓΡΑΜΜΑ 1 - ΑΡΙΘΜΟΣ ΧΡΗΣΤΩΝ ΕΦΑΡΜΟΓΗΣ**

Ποιος θα περίμενε ότι θα φτάναμε στο σημείο να μπορούμε να πληρώσουμε έναν λογαριασμό ή να μεταφέρουμε λεφτά μέσω μιας εφαρμογής στο κινητό. Αυτό συμβαίνει πολύ συχνά σήμερα και αποτελεί την 3<sup>η</sup> κατηγορία εφαρμογών. Πλέον κάθε τράπεζα έχει την δική της εφαρμογή για κινητό, όπου δίνει την δυνατότητα στους χρήστες της να πληρώσουν λογαριασμούς, να μεταφέρουν χρήματα ,να δουν το υπόλοιπό τους απλά με ένα κλικ. Υπάρχουν επίσης και εφαρμογές όπως είναι η PayPal, Skrill, n26 οι οποίες σου επιτρέπουν να άμεσα και με ασφάλεια αγορές στο διαδίκτυο.

Από τις κατηγορίες εφαρμογών δεν θα μπορούσαν να λείπουν τα mobile games. Μπορεί να μην είναι το ίδιο χρήσιμα όπως οι τρεις προηγούμενες κατηγορίες αλλά σίγουρα είναι υπεύθυνα για την επιτυχία και την εξέλιξη των εφαρμογών. Αυτό βέβαια μπορεί να αποδειχτεί παρουσιάζοντας τα νούμερα λήψεων κάποιων εξαιρετικά δημοφιλών παιχνιδιών όπως είναι το PUBG Mobile, Pokémon Go, Subway Surfers, Fruit Ninja, Candy Crush Saga, Among Us.



ΔΙΑΓΡΑΜΜΑ 2 - ΛΗΨΕΙΣ ΠΑΙΧΝΙΔΙΩΝ

Από το παραπάνω διάγραμμα παρατηρούμε απίστευτα υψηλά νούμερα λήψεων των παιχνιδιών που αγγίζουν και τον αριθμό των δημοφιλέστερων social media και chatting εφαρμογών.

Συνεχίζουμε με τις εφαρμογές online αγορών αγαθών και φαγητού. Πολλά καταστήματα ρουχισμού, ηλεκτρικών και ηλεκτρονικών ειδών καθώς και διάφορων άλλων αγαθών διαθέτουν εφαρμογές για την πιο εύκολη και άμεση αγορά αλλά και για την αποφυγή συνωστισμού στα φυσικά τους καταστήματα. Επίσης εφαρμογές διανομής φαγητού όπως είναι το e-food, έχουν μεγάλη απήχηση στο κοινό. Σαφώς με αυτές τις εφαρμογές δεν επωφελούνται μόνο οι χρήστες, αλλά και τα καταστήματα που τις διαθέτουν, και για θέματα marketing αφού το μεγαλύτερο ποσοστό σήμερα επιλέγει τις online αγορές, αλλά και για θέματα οργάνωσης. Γι' αυτόν τον λόγο προωθούνται και διάφορες εκπτώσεις και προσφορές αν και εφόσον η αγορά γίνει μέσω της εκάστοτε εφαρμογής.

Μια κατηγορία όπου έχει εξελιχθεί και αναγνωριστεί τα τελευταία χρόνια είναι το IoT (internet of things) και απευθύνεται σε ασύρματες συσκευές που μπορούμε να τοποθετήσουμε στο σπίτι μας, οι οποίες διαθέτουν εφαρμογές που μας δίνουν την δυνατότητα να τις ελέγξουμε και διαχειριστούμε από μακριά μέσω του κινητού μας τηλεφώνου ή tablet. Μπορούμε για παράδειγμα να ρυθμίσουμε τον θερμοστάτη χωρίς να είμαστε σπίτι ή να ελέγξουμε τις κάμερες του σπιτιού, να δούμε μέσα από την εφαρμογή την ενέργεια που καταναλώνουμε ακόμα και να ανάψουμε κάποια συσκευή όπως είναι το κλιματιστικό πριν καν φτάσουμε σπίτι. Αυτή η κατηγορία είναι ίσως το επόμενο βήμα της τεχνολογίας και αποτελεί τρανή απόδειξη του πόσο ραγδαία είναι η εξέλιξή της.

Βέβαια υπάρχουν κι άλλες κατηγορίες εφαρμογών όπως είναι οι εφαρμογές προβολής βίντεο ή ταινιών και αναπαραγωγής μουσικής (YouTube, Netflix, Amazon prime, Spotify), εφαρμογές υγείας, διατροφής και γυμναστικής, εφαρμογές για μέσα μεταφοράς π.χ. Taxi beat ή και να βλέπεις τι ώρα περνάει το κάθε λεωφορείο από τις στάσεις ακόμα και Dating εφαρμογές. Όλες αυτές μπορεί να μην είναι το ίδιο χρήσιμες και “απαραίτητες” όσο οι προηγούμενες κατηγορίες που αναλύσαμε, αλλά κι αυτές έχουν συμβάλει στην ανάπτυξη της τεχνολογίας και της χρησιμότητας του κινητού τηλεφώνου. Αντιλαμβανόμαστε λοιπόν πόσα πράγματα μπορεί να κάνει ο άνθρωπος έχοντας στην κατοχή του, το “πολυεργαλείο” που ονομάζεται κινητό τηλέφωνο [6].



## ΚΕΦΑΛΑΙΟ 2

### 2.1 Συνεργατικές εφαρμογές

Με τον όρο Συνεργατικές εφαρμογές, εννοούμε τις εφαρμογές εκείνες που δίνουν την δυνατότητα στους χρήστες να συνεργαστούν μεταξύ τους για να πετύχουν έναν στόχο, όπως το ορίζει και το λεξιλόγιο της Οξφόρδης: *Work jointly*(δουλειά από κοινού). Αυτές οι εφαρμογές επισημαίνουν στον χρήστη ότι είναι μέλος ενός κοινωνικού δικτύου κατά το οποίο απαιτείται συνεργασία για την σωστή λειτουργία, σε αντίθεση με κάποιες κοινές εφαρμογές οι οποίες αποκρύπτουν τους χρήστες μεταξύ τους [7].

Βέβαια οι συνεργατικές εφαρμογές χωρίζονται κι αυτές με την σειρά τους σε κατηγορίες αφού δεν έχουν όλες κοινό σκοπό. Οι εφαρμογές συνεδριάσεων αποτελούν τις πιο γνωστές, οι οποίες δίνουν την δυνατότητα στους χρήστες να συνεδριάσουν μέσω βίντεο-κλήσης, chatting, να παρουσιάσουν διαφάνειες, να κανονίσουν ραντεβού και να αναρτήσουν αρχεία ώστε να είναι προσβάσιμα στο υπόλοιπο group. Τέτοιες εφαρμογές χρησιμοποιούνται σήμερα κατά κύριο λόγο από μαθητές, φοιτητές, εταιρίες και επιχειρήσεις για την πιο άμεση και απλούστερη επικοινωνία, αντιμετώπιση προβλημάτων και την επίτευξη στόχων. Μια τέτοια εφαρμογή είναι το MS teams (Microsoft teams), η οποία προσφέρει έναν συνεργατικό κόσμο στους χρήστες, δίνοντας την δυνατότητα κλήσεων και βίντεο-κλήσεων, συνεδριάσεων ακόμα και τον διαμοιρασμό οθόνη(Share screen).

Μια ακόμα καινοτόμα και χρήσιμη εφαρμογή είναι το SmartAPPCity, η οποία πρωτοκυκλοφόρησε στην Ισπανία αλλά είναι σχεδιασμένη για να λειτουργεί σε όλες τις χώρες και πόλεις του κόσμου. Πρόκειται ουσιαστικά για μια all-inclusive εφαρμογή η οποία παρέχει λύση σε καθημερινά ζητήματα των πολιτών με την απαιτούμενη συνεργασία τους [8]. Μια λειτουργία της είναι το PARKINGS AND BIKE SHARING, κατά την οποία μπορεί να δει ο χρήστης στο χάρτη τα πλησιέστερα ιδιωτικά Parking μαζί με την χωρητικότητα και διαθεσιμότητά τους, καθώς επίσης και να δει αν υπάρχει διαθέσιμο ποδήλατο στον εκάστοτε σταθμό ποδηλάτου. Βέβαια για να συμβεί αυτό απαιτείται πλήρη συνεργασία και συνεχή ενημέρωση μεταξύ των χρηστών.

Μια άλλη πολύ πρωτότυπη και ενδιαφέρουσα εφαρμογή είναι η HTC power to give. Με την εξέλιξη της επιστήμης και την ανάγκη να ερευνά και να δίνει απαντήσεις σε ολοένα και περισσότερα ζητήματα, ήρθε στην αγορά το HTC power to give, με την πολύ απλή του χρήση για να βοηθήσει στον τομέα αυτόν. Το μόνο που έχει να κάνει κάποιος είναι να κατεβάσει την εφαρμογή, να έχει ανοιχτό το Wi-Fi και να τοποθετήσει το κινητό για φόρτιση. Όσο το κινητό φορτίζει, δέχεται μικρά κομμάτια

από περιβαλλοντικά, ιατρικά, επιστημονικά ακόμα και εξωγήινα ερευνητικά projects τα οποία ολοκληρώνει και τα επιστρέφει στους δικαιούχους. Έτσι με την λήψη και τη χρησιμοποίηση της εφαρμογής γινόμαστε κομμάτι ενός παγκόσμιου δικτύου που έχει στόχο να δώσει απαντήσεις σε πολύ σημαντικές επιστημονικές ερωτήσεις που απασχολούν για χρόνια την ανθρωπότητα [9]

## 2.2 Εφαρμογές Risk Management

Με τον όρο Risk Management εννοούμε την διαχείριση κινδύνου και χρησιμοποιείται κυρίως από εταιρίες και επιχειρήσεις οι οποίες κάνουν μεθοδικά βήματα για να προσεγγίσουν τους κινδύνους και να πετύχουν το μέγιστο δυνατό κέρδος.

Στο πλαίσιο των εφαρμογών όμως, η έννοια του Risk Management είναι διαφορετική και απευθύνεται σε εφαρμογές οι οποίες διαχειρίζονται κινδύνους με το να ενημερώνουν, αποτρέπουν και να προστατεύουν τους χρήστες από φυσικές καταστροφές, ακραία καιρικά φαινόμενα, ατυχήματα, κινδύνους στις πόλεις, στους δρόμους κ.α.

Δυο είναι οι κύριες κατηγορίες όπου χωρίζονται οι εφαρμογές του Risk Management:

- Συνεργατικές εφαρμογές (Reports and Cooperation)
- Εφαρμογές εκτάκτων περιστάσεων (Emergency Alerts)

Και τις δυο θα τις αναλύσουμε, αλλά θα εστιάσουμε περισσότερο στις εφαρμογές αντιμετώπισης εκτάκτων καταστάσεων στο κομμάτι των φυσικών καταστροφών, αφού τα τελευταία χρόνια αποτελεί μείζον ζήτημα παγκοσμίως.

### 2.2.1 Συνεργατικές εφαρμογές στο πεδίο του Risk Management

Τι θα γινόταν αν οι χρήστες συνεργάζονταν μεταξύ τους για την διαχείριση κινδύνων; Η απάντηση δίνεται από κάποιες πολύ χρήσιμες εφαρμογές που βρίσκονται στην αγορά, όπως είναι το Nononville.

Πρόκειται για μια φιλική προς το χρήστη εφαρμογή που διαχειρίζεται από τοπικούς δήμους και επιτρέπει στους χρήστες να υποβάλλουν μια αναφορά εάν εντοπίσουν κάποιο πρόβλημα ή ζημιά στο δρόμο ή στη γειτονιά. Συγκεκριμένα, οι χρήστες ενημερώνουν τους δήμους για οποιοδήποτε πρόβλημα αντιμετωπίζουν, για παράδειγμα, πλημμύρες από σπασμένους σωλήνες ή μεγάλη λακκούβα στο δρόμο κ.λπ. προκειμένου να λάβουν άμεσα μέτρα. Κατά αυτόν τον τρόπο ενημερώνεται τόσο ο δήμος για την επίλυση της αναφοράς όσο και οι υπόλοιποι χρήστες, ώστε να αποφύγουν το επίκεντρο του προβλήματος. Στην Ελλάδα αυτή η εφαρμογή χρησιμοποιείται από πολλούς δήμους με μεγάλη επιτυχία [10].

Μια άλλη εφαρμογή είναι το LifeRing Assist, το οποίο επιτρέπει στους χρήστες να δημιουργήσουν άμεσα ένα ψηφιακό δίκτυο στο οποίο θα φαίνεται η τοποθεσία του κάθε χρήστη, έτσι ώστε σε περίπτωση ανάγκης να ειδοποιηθούν οι κοντινότεροι χρήστες και να σπεύσουν για βοήθεια. Η εφαρμογή παρέχει επίσης κάποια μέσα συνεργασίας όπως text messaging και Voice ώστε να υπάρχει μια πιο σωστή και έμπιστη συνεργασία μεταξύ των χρηστών της [11]

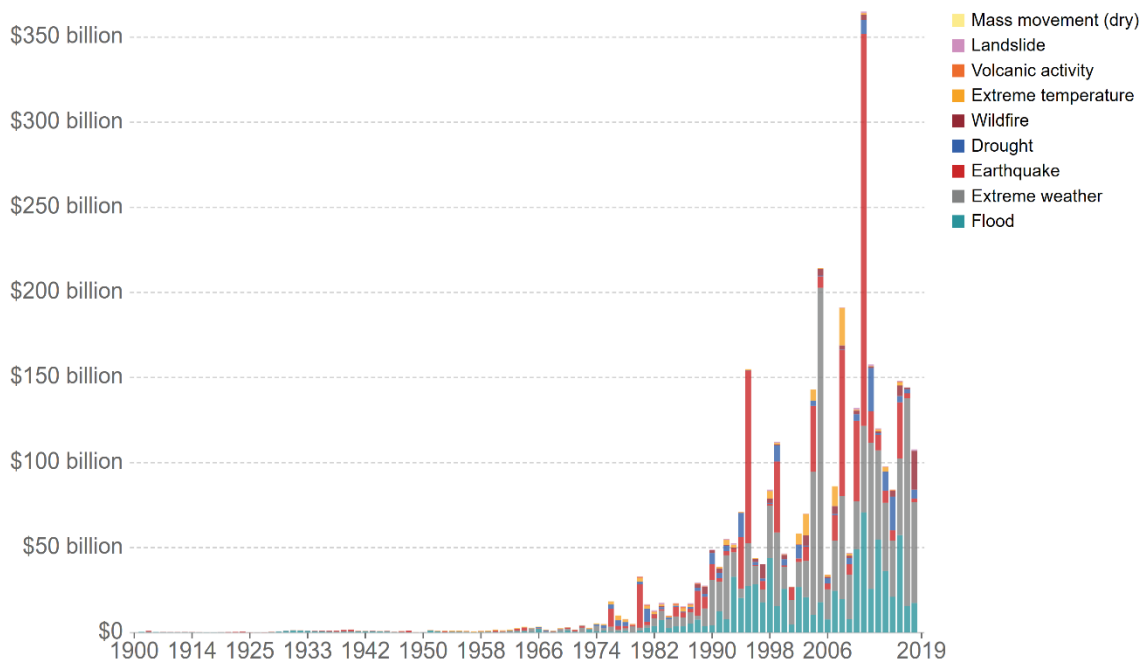
## 2.2.2 Emergency Alert Apps για φυσικές καταστροφές από ακραία καιρικά φαινόμενα

Οι φυσικές καταστροφές δεν έχουν επιπτώσεις μόνο στην σωματική ακεραιότητα του ανθρώπου αλλά και στο περιβάλλον και στην παγκόσμια οικονομία. Γι' αυτόν τον λόγο τα τελευταία χρόνια το επιστημονικό πεδίο έχει ρίξει το ενδιαφέρον του σε μελέτες που αφορούν τις φυσικές καταστροφές οι οποίες προκαλούνται από έντονα καιρικά φαινόμενα. Το μέγεθος αυτών των επιπτώσεων εξαρτάται από διάφορους παράγοντες, όπως είναι η ένταση και η συχνότητα εμφάνισης των καιρικών φαινομένων, η προετοιμασία και η ευαισθησία του κοινού που προσβάλλεται.

### Economic damage by natural disaster type, 1900 to 2019

Global economic damage from natural disasters, differentiated by disaster category and measured in US\$ per year.

Our World  
in Data



Source: EMDAT (2020): OFDA/CRED International Disaster Database, Université catholique de Louvain – Brussels – Belgium  
OurWorldInData.org/natural-disasters • CC BY

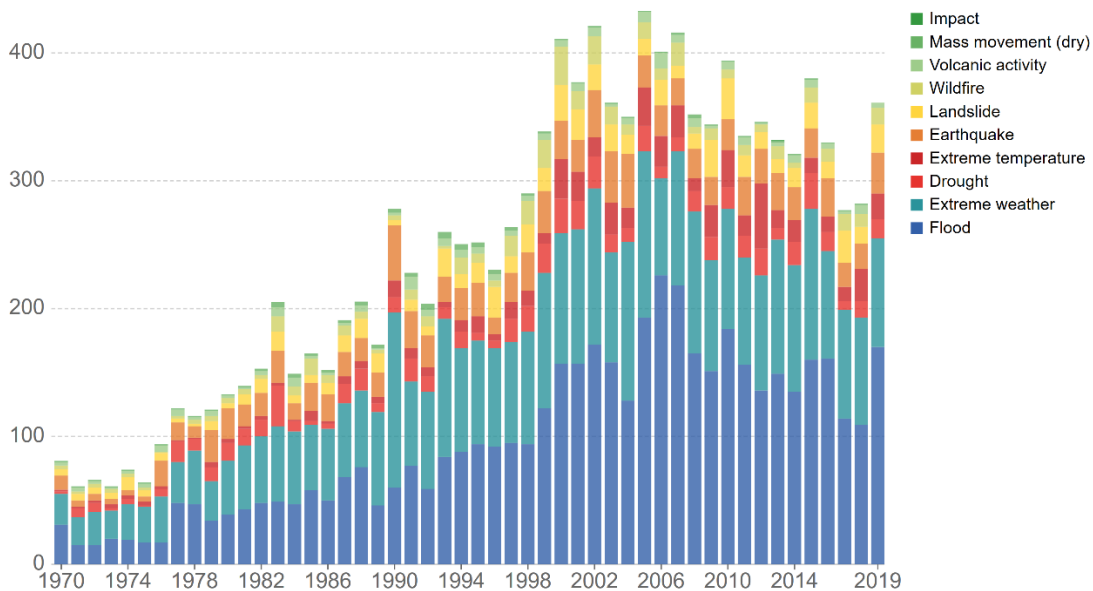
**ΕΙΚΟΝΑ 3 - ΟΙΚΟΝΟΜΙΚΕΣ ΖΗΜΙΕΣ ΑΠΟ ΦΥΣΙΚΕΣ ΚΑΤΑΣΤΡΟΦΕΣ, 1900 – 2019**

Παρατηρείται εξαιρετικά μεγάλη παγκόσμια οικονομική ζημία τα τελευταία 20 χρόνια από φυσικές καταστροφές και κυρίως από σεισμούς και έντονα καιρικά φαινόμενα [12].

Επίσης το διάγραμμα ένδειξης αριθμού των φυσικών καταστροφών που συμβαίνουν με την πάροδο του χρόνου, σημειώνει μια ανοδική πορεία. Ο πλανήτης δεν πλήττει κάθε 5ετία ή 10ετία από τις ίδιες φυσικές καταστροφές, για παράδειγμα το 1990 επιβεβαιώθηκαν μόλις 60 καταστροφές από πλημμύρα ενώ το 2019 επιβεβαιώθηκαν 170, δηλαδή σχεδόν τριπλασιάστηκε ο αριθμός. Βέβαια κάθε ήπειρος και χώρα συνήθως πλήττουν από διαφορετικά καιρικά φαινόμενα με βάση την γεωγραφική τους θέση [12].

### Global reported natural disasters by type, 1970 to 2019

The annual reported number of natural disasters, categorised by type. This includes both weather and non-weather related disasters.



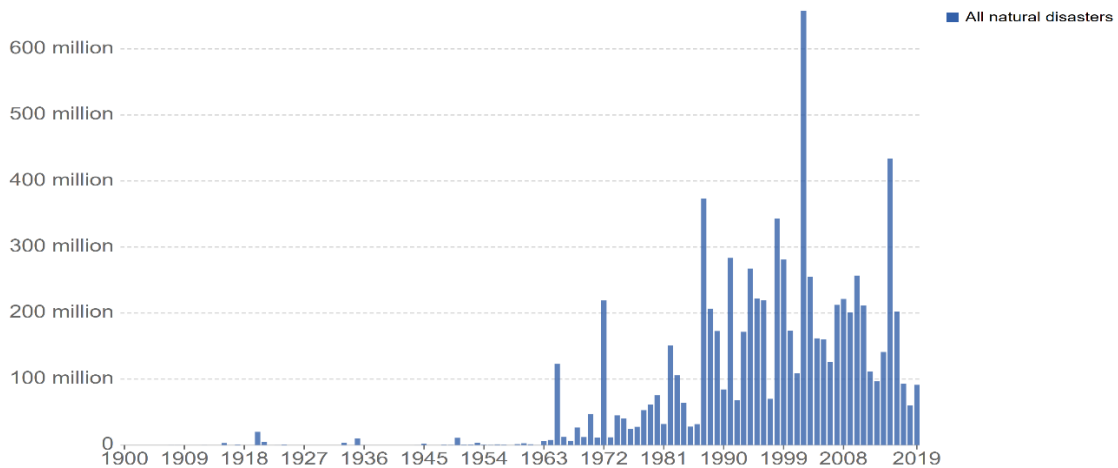
Source: EMDAT (2020): OFDA/CRED International Disaster Database, Université catholique de Louvain – Brussels – Belgium  
OurWorldInData.org/natural-disasters • CC BY

**ΕΙΚΟΝΑ 4 - ΠΑΓΚΟΣΜΙΕΣ ΑΝΑΦΟΡΕΣ ΦΥΣΙΚΩΝ ΚΑΤΑΣΤΡΟΦΩΝ ΜΕ ΒΑΣΗ ΤΟΝ ΤΥΠΟ ΤΟΥΣ, 1970 – 2019**

## Global number affected by natural disasters, All natural disasters, 1900 to 2019



Global total number of people affected by natural disasters. This is defined as the sum of the injured, affected and those left homeless after a disaster.



Source: EMDAT (2020): OFDA/CRED International Disaster Database, Université catholique de Louvain – Brussels – Belgium  
OurWorldInData.org/natural-disasters • CC BY

### ΕΙΚΟΝΑ 5 - ΠΑΓΚΟΣΜΙΟΣ ΑΡΙΘΜΟΣ ΠΡΟΣΒΕΒΛΗΜΕΝΩΝ ΑΠΟ ΦΥΣΙΚΕΣ ΚΑΤΑΣΤΡΟΦΕΣ, 1900 – 2019

Γεννιούνται λοιπόν πολλά ερωτήματα όπου καλούμαστε να απαντήσουμε όπως, γιατί οι εφαρμογές του Risk management είναι τόσο σημαντικές, γιατί είναι επίκαιρες και τι γινόταν μέχρι σήμερα όπου δεν υπήρχε τέτοια τεχνολογία ώστε να μας προετοιμάζει για την διαχείριση των φυσικών καταστροφών;

Για να απαντηθούν αυτά τα ερωτήματα πρέπει πρώτα να αναλογιστούμε και να διαχωρίσουμε τις φυσικές καταστροφές. Οι κύριες κατηγορίες είναι οι εξής [13]:

1. Οι υδρολογικές (Hydrological)
  2. Οι κλιματικές (Climatological)
  3. Οι γεωφυσικές (Geophysical)
  4. Οι μετεωρολογικές (Meteorological)
- *Υδρολογικές καταστροφές* μπορεί να είναι πλημμύρες που προκαλούνται από έντονες καταιγίδες, φουσκωμένους ποταμούς που ξεχειλίζουν και προκαλούν καταστροφές λόγω της ορμής ή ακόμα και από το απότομο λιώσιμο μεγάλης ποσότητας χιονιού.  
Επίσης το τσουνάμι και η έκχυση παγετώνων μπορούν να προκαλέσουν τέτοιου είδους καταστροφές.
  - *Κλιματικές καταστροφές* οι οποίες δημιουργούνται από έντονες καιρικές συνθήκες. Μια εξ' αυτών είναι η ξηρασία κατά την οποία δημιουργούνται πυρκαγιές λόγω της έλλειψης νερού στην γη και της υψηλής θερμοκρασίας

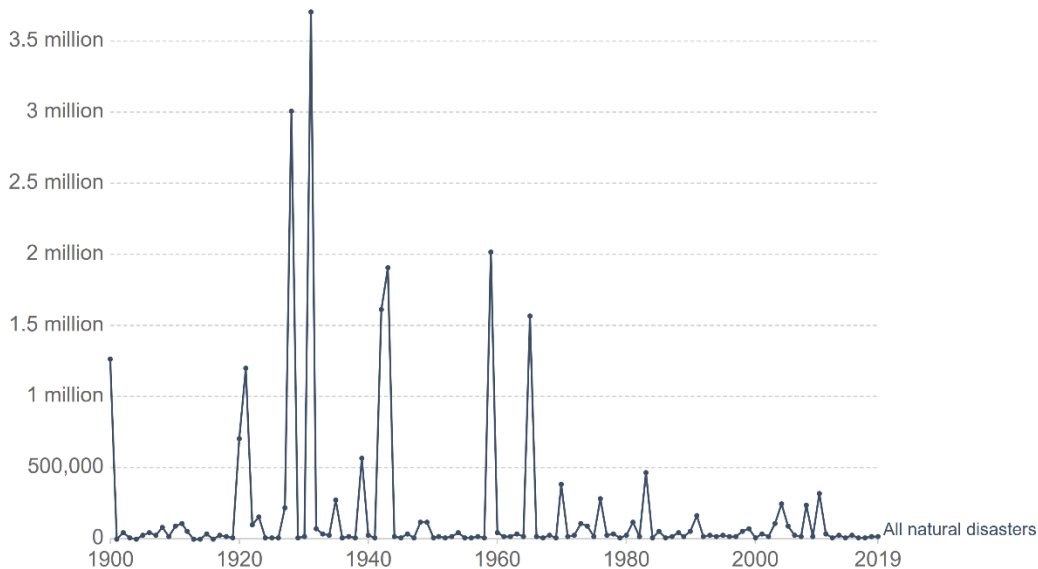
και έντονης ακτινοβολίας του ήλιου. Επίσης η χιονοθύελλα/παγοθύελλα και χαλαζόπτωση μπορεί να προκαλέσουν κλιματικές καταστροφές

- *Γεωφυσικές καταστροφές* όπως είναι οι σεισμοί, οι ηφαιστειακές εκρήξεις και οι κατολίσθηση, αποτελούν τις πιο επικίνδυνες και θανατηφόρες καταστροφές.
- *Μετεωρολογικές καταστροφές* που συνήθως δημιουργούνται από θύελλες, τυφώνες και τροπικούς κυκλώνες.

Επομένως μπορούμε να απαντήσουμε στο ερώτημα για την σπουδαιότητα των εφαρμογών Risk Management for natural disasters, με το παρακάτω διάγραμμα όπου μας δείχνει πως την τελευταία 10ετία έχουν μειωθεί αισθητά οι θάνατοι από φυσικές καταστροφές κι αυτό συμβαίνει λόγω της τωρινής τεχνολογίας όπου μας δίνει την δυνατότητα μέσω εφαρμογών να ενημερωνόμαστε άμεσα και να λαμβάνουμε εγκαίρως τα μέτρα για την προστασία μας. Επίσης τις καθιστά επίκαιρες, αφού όπως φαίνεται και στο Figure 4. τα τελευταία 20 χρόνια, πάνω από το 90% των καταστροφών προκλήθηκε από υδρολογικά, κλιματολογικά και μετεωρολογικά αίτια, δηλαδή από πλημμύρες, έντονα κύματα καύσωνα, ξηρασίες και γενικότερα λόγω της απότομης κλιματικής αλλαγής. Εφόσον λοιπόν βιώνουμε τέτοιες ακραίες καταστάσεις, είναι αναγκαίο να έχουμε χρήσιμες εφαρμογές οι οποίες θα μας προστατεύσουν από τον κίνδυνο.

### Global deaths from natural disasters, 1900 to 2019

Absolute number of global deaths per year as a result of natural disasters. "All natural disasters" includes those from drought, floods, extreme weather, extreme temperature, landslides, dry mass movements, wildfires, volcanic activity and earthquakes.



Source: EMDAT: OFDA/CRED International Disaster Database, Université catholique de Louvain – Brussels – Belgium  
OurWorldInData.org/natural-disasters/ • CC BY

**ΕΙΚΟΝΑ 6 - ΠΑΓΚΟΣΜΙΟΣ ΑΡΙΘΜΟΣ ΘΑΝΑΤΩΝ ΑΠΟ ΦΥΣΙΚΕΣ ΚΑΤΑΣΤΡΟΦΕΣ, 1900- 2019**

Μέχρι όμως να σημειώσει η τεχνολογία τέτοια πρόοδο και να σχεδιαστούν τέτοιου είδους εφαρμογές, δεν υπήρχε η ίδια αντιμετώπιση για παρόμοιες καταστάσεις. Οι άνθρωποι δεν ενημερωνόντουσαν για τυχόν επερχόμενες φυσικές καταστροφές με αποτέλεσμα να είναι εκτεθειμένοι σε αυτές. Αυτός είναι και ο λόγος όπου οι θάνατοι το 1980 άγγιξαν τους 500εκ. ενώ το 2019 δεν ξεπέρασαν καν τους 1.000.

Αυτοί είναι οι λόγοι που δημιούργησαν την αγορά των Emergency Alert Apps, η οποία είναι πλέον τεράστια και ο ανταγωνισμός αυτών μόνο θετικά μπορεί να αποφέρει. Οι εφαρμογές που σχεδιάζονται σ' αυτό το πεδίο δεν έχουν σκοπό να “κερδίσουν” σε λήψεις/πωλήσεις τις υπόλοιπες όμοιες εφαρμογές, αλλά στόχος τους είναι να προσφέρουν ολοένα και περισσότερες λειτουργίες και υπηρεσίες για να προστατεύσουν την ανθρωπότητα.

Είναι λοιπόν πολύ σημαντικό να πούμε ότι μπορούμε να μετατρέψουμε το κινητό μας τηλέφωνο σε εργαλείο επιβίωσης και αυτό το εργαλείο παρέχει πραγματικά χρήσιμες πληροφορίες όταν συμβεί το απροσδόκητο, είτε πρόκειται για φυσική καταστροφή, ιατρική κατάσταση έκτακτης ανάγκης, ζημιές στην πόλη ή ατυχήματα. Επομένως, ο κύριος σκοπός αυτών των εφαρμογών είναι να ενημερώνουν τους χρήστες, να τους ειδοποιούν και να τους προστατεύουν από δυσάρεστα γεγονότα που μπορεί να συμβούν ή έχουν ήδη συμβεί.

Θα αναφερθούμε σε Emergency Alert apps για φυσικές καταστροφές και ακραία καιρικά φαινόμενα. Τέτοιες εφαρμογές διαχειρίζονται από τις αρχές οι οποίες προβλέπουν ή εκτιμούν μια έκτακτη περίσταση και ειδοποιούν άμεσα τον χρήστη για να προστατευθεί.

Μια τέτοια εφαρμογή αποτελεί η Wildfire Alert (Android, iOS), η οποία έχει κατασκευαστεί για να ειδοποιεί τους χρήστες της, για τις περιοχές όπου υπάρχει πυρκαγιά, να τους ενημερώσει για τυχόν εξάπλωσή της ή ακόμα και να τους ενημερώσει για τα σημεία που έχουν αυξημένη επικινδυνότητα για να προκληθεί μια πυρκαγιά [14]. Σαφώς και είναι πολύ χρήσιμη εφαρμογή, αφού παρατηρούμε τα τελευταία χρόνια τόσο στην Ελλάδα όσο και σε άλλες χώρες μεγάλη αύξηση πυρκαγιών λόγω της κλιματικής αλλαγής και απροσεξίας των ανθρώπων.

Παρόμοιο σκοπό έχει και η εφαρμογή FEMA (Android, iOS), δηλαδή να ενημερώσει και να προετοιμάσει τους χρήστες της για τυχόν φυσικές καταστροφές. Σχεδιάστηκε αρχικά για την Αμερική η οποία κάθε χρόνο πλήττει από πολύ ακραία καιρικά φαινόμενα όπως είναι τυφώνες και πλημμύρες. Με λίγα λόγια προειδοποιεί πριν από κάθε καιρικό φαινόμενο, προσφέρει την λειτουργία κλήσης στην άμεση δράση(9-1-1) κατά την διάρκεια και το τέλος της καταστροφής και τέλος προτείνει λύσεις αντιμετώπισης [15].

Επίσης η εφαρμογή του Αμερικάνικου Ερυθρού Σταυρού Earthquake (Android, iOS), παρέχει αναλυτικές οδηγίες για το τι πρέπει να κάνει ο χρήστης πριν, κατά την διάρκεια και μετά από έναν σεισμό. Ο χρήστης λαμβάνει ειδοποίηση όταν συμβαίνει ένας σεισμός με τον αριθμό των ρίχτερ και τυχόν προβλέψεις για μετασεισμούς

καθώς του δίνεται και η δυνατότητα να ενημερώσει τους συγγενείς και φίλους ότι είναι ασφαλής απλά με ένα κλικ [16].

Το Weather Underground (Android, iOS) είναι μια crowdsourcing εφαρμογή όπου δίνει πληροφορίες και προβλέψεις για τον καιρό και για τυχόν επιδείνωσή του μέσω του χάρτη και φωτογραφιών.

Το Natural Disaster Monitor (Android, iOS) παρόμοια εφαρμογή με το Weather Underground αλλά επισημαίνει κυρίως τα σημεία του χάρτη όπου υπάρχει κίνδυνος για τυφώνες, σεισμούς, πλημύρες, τσουνάμι κλπ.



## ΚΕΦΑΛΑΙΟ 3

### 3.1 Η εφαρμογή

#### 3.1.1 Περιγραφή Κεντρικής Ιδέας της εφαρμογής

Εξίσου σημαντικό μέρος της παρούσας διπλωματικής εργασίας αποτέλεσε η σχεδίαση και ανάπτυξη της Android εφαρμογής **Fast Report**, η οποία έχει ως στόχο την άμεση αναφορά τυχόν προβλημάτων ή ζημιών στους δρόμους και στις γειτονιές (τα οποία μπορεί να έχουν προκληθεί είτε από ανθρώπινο λάθος είτε από έντονα καιρικά φαινόμενα), δημιουργώντας ένα δίκτυο στο οποίο οι χρήστες της εφαρμογής αλληλοβοηθούνται και αλληλοεπιδρούν μεταξύ τους με μοναδικό εργαλείο το κινητό τους τηλέφωνο, την χρήση δεδομένων κινητής τηλεφωνίας και τις υπηρεσίες του Google Maps.

Ο χρήστης είτε είναι σε μια γειτονιά είτε είναι σε έναν κεντρικό δρόμο, είτε είναι πεζός είτε με κάποιο μεταφορικό μέσον, μπορεί εύκολα με την χρήση της εφαρμογής να κάνει γρήγορα αναφορά σε περίπτωση που εντοπίσει κάποιο πρόβλημα π.Χ. μεγάλη λακκούβα στον δρόμο, πλημμύρα, χαλασμένες λάμπες, τρακάρισμα, κατολίσθηση κλπ. Κατ' αυτόν τον τρόπο ενημερώνει τους υπόλοιπους χρήστες της εφαρμογής για το πρόβλημα το οποίο έχει δημιουργηθεί καθώς και το μέρος στο οποίο έχει εντοπιστεί.

Έτσι ο χρήστης πέρα από την αναφορά του προβλήματος μέσω του Google Maps και φωτογραφιών, μπορεί να βοηθήσει περαιτέρω τους υπόλοιπους χρήστες σημειώνοντας στα σχόλια της αναφοράς εναλλακτικούς δρόμους ώστε να αποφύγουν το αναφερόμενο συμβάν.

Βέβαια απαραίτητη προϋπόθεση για κάθε χρήστη είναι να βρίσκεται συνδεδεμένος στο δίκτυο κινητής του τηλεφωνίας, είτε σε κάποιο Wi-Fi είτε με τα δεδομένα κινητής τηλεφωνίας του, ώστε να βλέπει άμεσα τις αναφορές των υπολοίπων, μέσω και των notifications.






Κατά αυτόν τον τρόπο δημιουργείται στην εφαρμογή ένα δίκτυο χρηστών με απώτερο σκοπό να βοηθήσει ο ένας τον άλλον αλλά και να ενημερωθεί ο δήμος ή η αρμόδια αρχή ώστε να αντιμετωπισθεί άμεσα το πρόβλημα.

### 3.2 Τεχνικά Χαρακτηριστικά

Η ανάπτυξη της εφαρμογής έγινε με την βοήθεια της γλώσσας προγραμματισμού Java στο περιβάλλον του Android Studio, έκδοση 4.1.3 με Android SDK Tools 31.0.0-rc2.



Χρησιμοποιήθηκε επίσης η υπηρεσία Firebase και συγκεκριμένα το Realtime Database για βάση δεδομένων, λόγω κυρίως της ανταλλαγής δεδομένων που απαιτεί η εφαρμογή σε πραγματικό χρόνο. Επίσης ο κώδικας των notifications έγινε στο περιβάλλον του node με την βοήθεια της JavaScript. Τέλος απαραίτητη ήταν και η χρήση των Google APIs και των παρακάτω services:

-  Maps SDK for Android
-  Places API
-  Directions API
-  Geocoding API
-  Geolocation API



### 3.3 Ο τίτλος της εφαρμογής – “FastReport”

Προτού ξεκινήσει ο σχεδιασμός της εφαρμογής, έπρεπε να βρεθεί ένα αντιπροσωπευτικό όνομα το οποίο θα έδειχνε περί τίνος πρόκειται η λειτουργία της εφαρμογής. Γι' αυτό τον λόγο δόθηκε ο τίτλος “FastReport” ο οποίος υποδηλώνει την άμεση και έγκαιρη αναφορά.

### 3.4 Που απευθύνεται

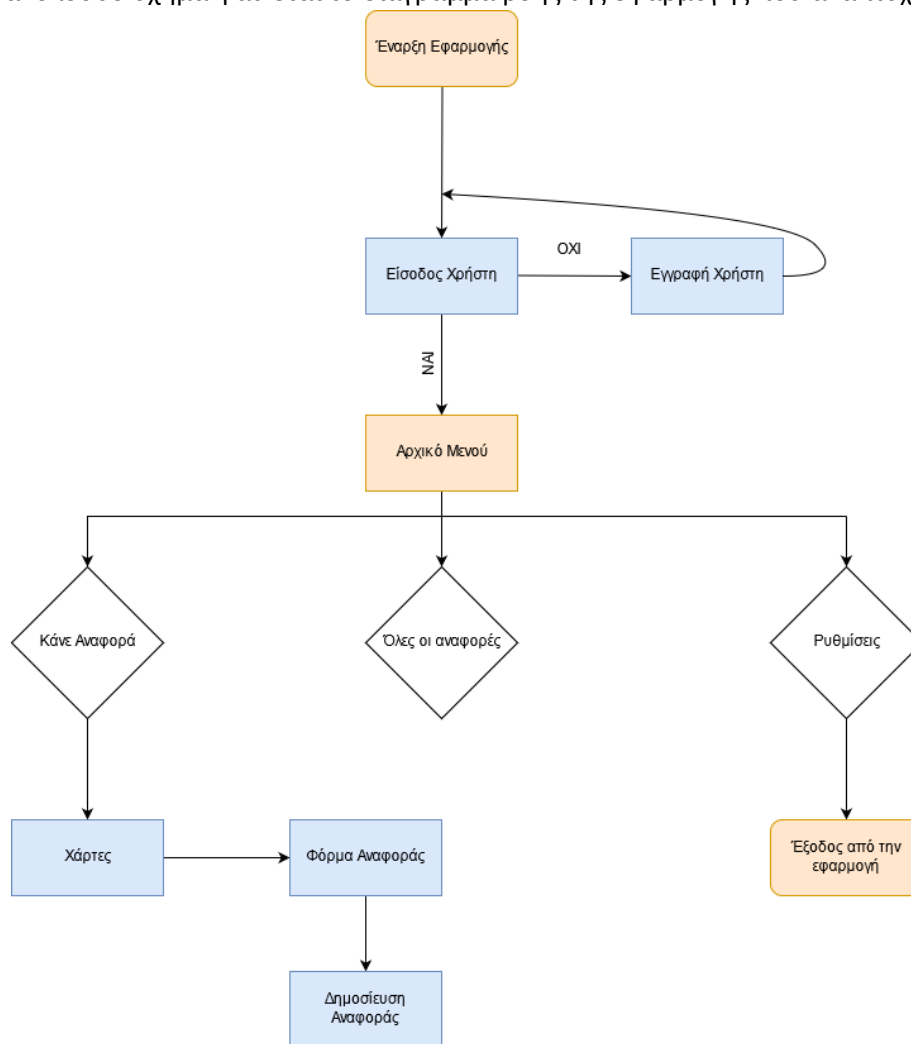
Η εφαρμογή απευθύνεται στο ευρύ κοινό, ανεξαρτήτου ηλικίας αρκεί να υπάρχει εξοικείωση με τα κινητά τηλέφωνα και την χρήση τους. Είναι πολύ φιλική προς τον χρήστη και χρησιμοποιεί την Αγγλική γλώσσα στην πιο απλή και κατανοητή της μορφή.

### 3.5 Στόχος της εφαρμογής

Η εφαρμογή στοχεύει στην προστασία του πολίτη από απροσδόκητα και δυσάρεστα γεγονότα, αποτρέπει την κυκλοφοριακή συμφόρηση στους δρόμους και στοχεύει στην άμεση επίλυση των προβλημάτων από τις αρμόδιες αρχές.

### 3.6 Διάγραμμα ροής

Στο ακόλουθο σχήμα φαίνεται το διάγραμμα ροής της εφαρμογής που αναπτύχθηκε.

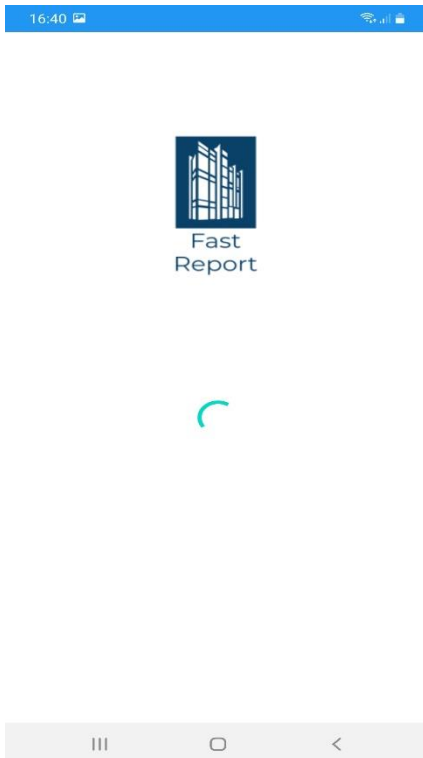


ΔΙΑΓΡΑΜΜΑ 3 – ΓΕΝΙΚΟ ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ

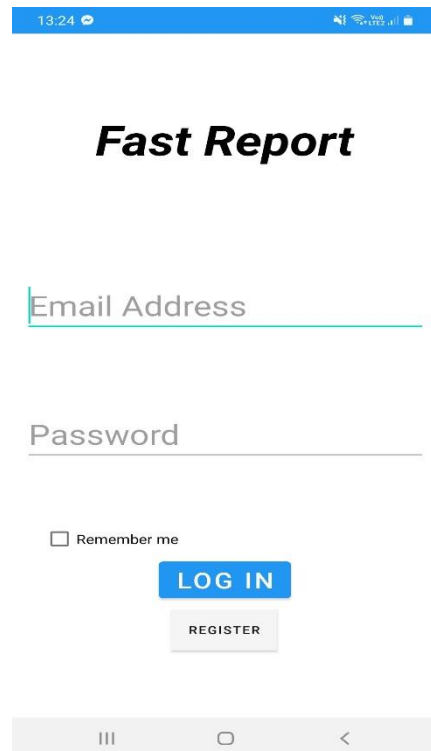
### 3.7 Περιγραφή οθονών εφαρμογής

Όταν πατήσουμε το εικονίδιο της εφαρμογής από την φορητή μας συσκευή, ανοίγει η εφαρμογή με ένα Loading Screen/καλωσόρισμα.

Ύστερα εμφανίζεται το πεδίο όπου ο χρήστης πρέπει να κάνει Log in για να εισέλθει στην εφαρμογή ή αν δεν είναι ήδη μέλος, μπορεί πολύ εύκολα να κάνει εγγραφή

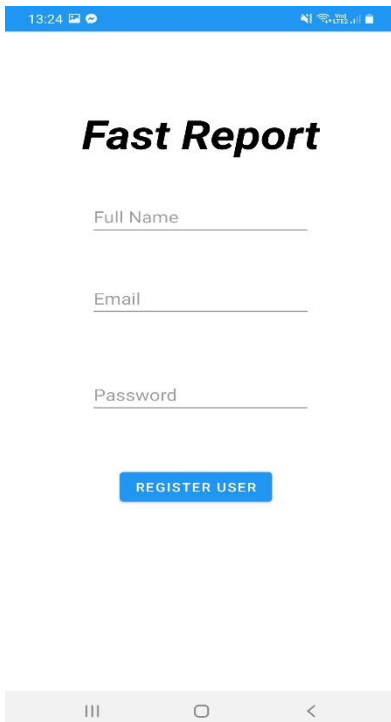


ΕΙΚΟΝΑ 7 - LOADING SCREEN

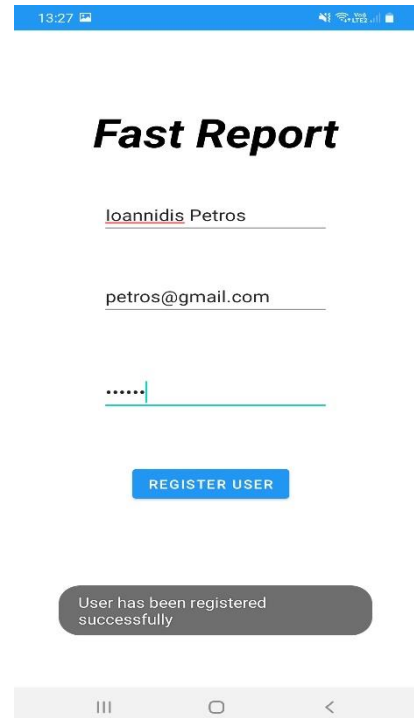


ΕΙΚΟΝΑ 8 - ΕΙΣΟΔΟΣ ΧΡΗΣΤΗ

Έστω ότι θέλει να πραγματοποιήσει κάποιος μια εγγραφή. Πρώτα θα πρέπει να συμπληρώσει τα απαραίτητα στοιχεία που του ζητά η εφαρμογή όπως φαίνεται στην εικόνα 9. Έπειτα αφού συμπληρώσει τα στοιχεία του και πατήσει το button 'Register user' όπως φαίνεται στην εικόνα 10, εμφανίζεται ένα μήνυμα επιβεβαίωσης της εγγραφής αν και εφόσον τα στοιχεία που έβαλε πληρούν τις απαραίτητες προϋποθέσεις.



ΕΙΚΟΝΑ 9 - ΕΓΓΡΑΦΗ ΧΡΗΣΤΗ



ΕΙΚΟΝΑ 10 - ΕΠΙΤΥΧΗΣ ΕΓΓΡΑΦΗ

Εφόσον έγινε η εγγραφή του χρήστη επιτυχώς, η εφαρμογή τον επιστρέφει αυτόματα στο πεδίο του Log in έτσι ώστε να εισέλθει στην εφαρμογή.

Εμφανίζεται το αρχικό μενού της εφαρμογής με ένα button 'MAKE A REPORT' όπου μπορεί να πατήσει ο χρήστης για να κάνει την αναφορά του και ένα navigation bar στο κάτω μέρος της οθόνης που παρέχει τρεις επιλογές, την αρχική οθόνη (Home), την συλλογή όλων των reports των χρηστών (Profile) και τις ρυθμίσεις (Settings).

13:27



## Fast Report

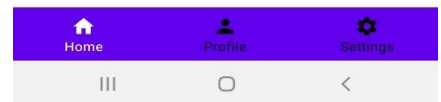
[MAKE A REPORT](#)


 Remember me

[LOG IN](#)
[REGISTER](#)


ΕΙΚΟΝΑ 12 - ΕΙΣΟΔΟΣ ΧΡΗΣΤΗ

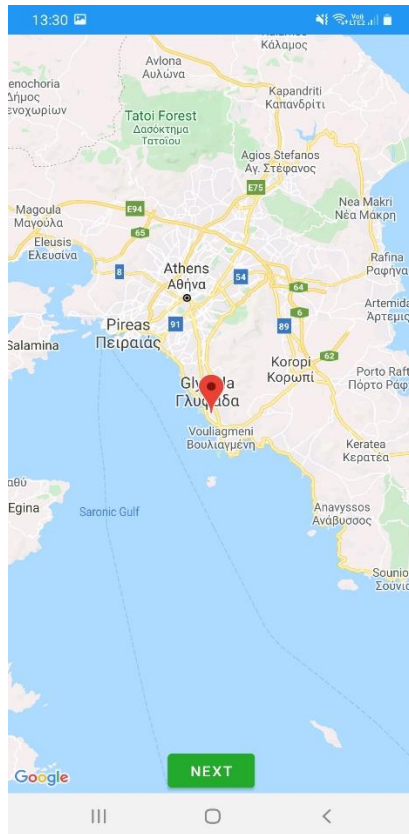
13:30



ΕΙΚΟΝΑ 11 - ΑΡΧΙΚΗ ΟΘΟΝΗ

Όταν πατήσει ο χρήστης το κουμπί για να κάνει την αναφορά του, εμφανίζεται αμέσως ο χάρτης και ένα marker το οποίο δείχνει την τρέχουσα τοποθεσία του. Υπάρχει βέβαια και η δυνατότητα zoom in και zoom out ώστε να επιβεβαιώσει και ο χρήστης την ακριβή τοποθεσία του.

Πατώντας το κουμπί 'NEXT' που φαίνεται στην εικόνα 13, ο χρήστης μεταφέρεται απευθείας στην φόρμα της αναφοράς που πρέπει να συμπληρώσει πριν την δημοσιεύσει. Το button 'CHOOSE FILE' απευθύνεται στην φωτογραφία του συμβάντος που θα πρέπει να έχει απαθανατίσει ο χρήστης και να την χρησιμοποιήσει ως αποδεικτικό. Δεξιά από το κουμπί υπάρχει και ένα πλαίσιο στο οποίο μπορεί να συμπληρώσει και να δώσει στοιχεία σχετικά με το συμβάν.



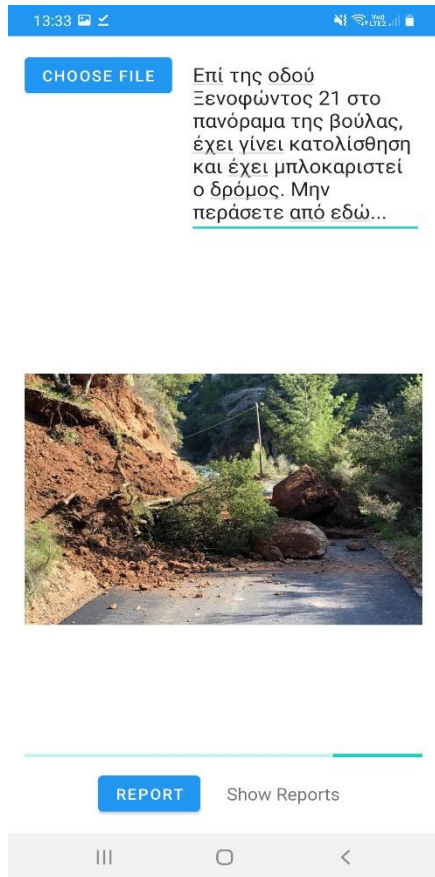
ΕΙΚΟΝΑ 14 - ΧΑΡΤΕΣ



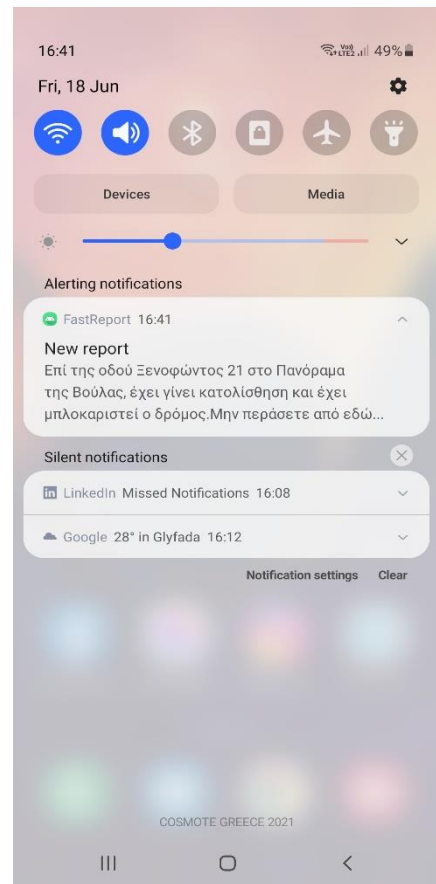
ΕΙΚΟΝΑ 13 - ΦΟΡΜΑ ΑΝΑΦΟΡΑΣ

Ο χρήστης επιλέγει την φωτογραφία που μόλις τράβηξε με το κινητό του και γράφει στα σχόλια όλες τις απαραίτητες πληροφορίες. Επίσης αν και εφόσον γνωρίζει, μπορεί να προτείνει δρόμους αποφυγής έτσι ώστε να αποφευχθεί η συμφόρηση αλλά και τυχόν ατύχημα. Στην συνέχεια πατώντας το κουμπί 'REPORT' εμφανίζεται μια μπάρα φόρτισης και όταν αυτή εξαφανισθεί σημαίνει ότι η αναφορά "ανέβηκε". Στα δεξιά υπάρχει και το κουμπί 'Show Reports' κατά το οποίο μπορεί να δει ο χρήστης και τις υπόλοιπες αναφορές που έχουν γίνει.

Επιπλέον όταν γίνει η αναφορά, στέλνεται αυτόματα ειδοποίηση στους υπόλοιπους χρήστες της εφαρμογής όπως φαίνεται στην εικόνα 16.



ΕΙΚΟΝΑ 16 - ΣΥΜΠΛΗΡΩΣΗ ΑΝΑΦΟΡΑΣ



ΕΙΚΟΝΑ 15 - NOTIFICATIONS

Πατώντας το εικονίδιο του Profile στο navigation bar, εμφανίζονται όλα τα reports από όλους τους χρήστες και με την βοήθεια του scroll down μπορεί να δει ο χρήστης όσες αναφορές τον ενδιαφέρουν.



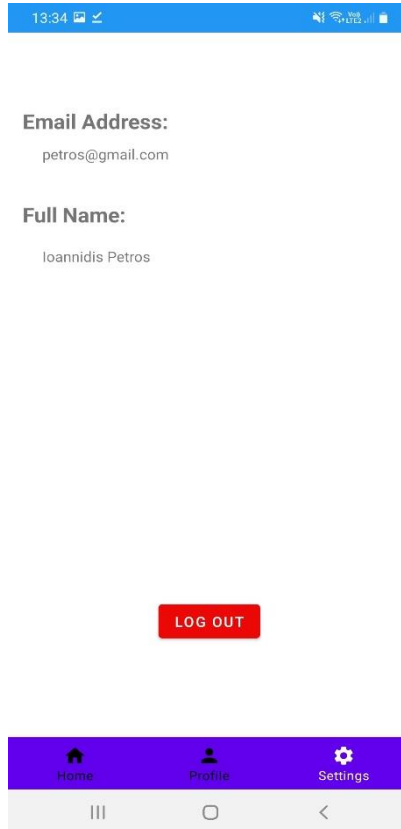


ΕΙΚΟΝΑ 18 - REPORTS



ΕΙΚΟΝΑ 17 - REPORTS

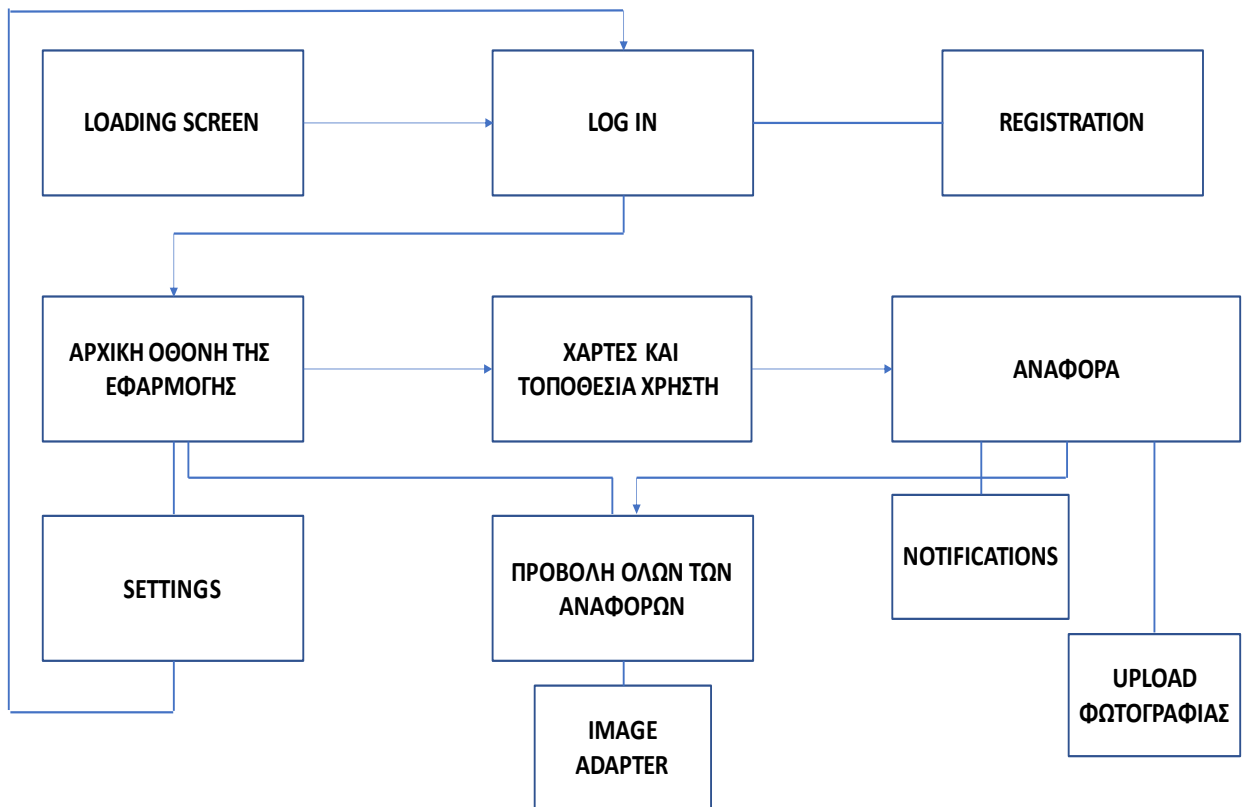
Τέλος, στις ρυθμίσεις εμφανίζονται οι πληροφορίες του χρήστη και εμφανίζεται ένα button 'LOG OUT' το οποίο μπορεί να πατήσει ο χρήστης για να βγει από την εφαρμογή.



**ΕΙΚΟΝΑ 19 - ΡΥΘΜΙΣΕΙΣ ΚΑΙ ΚΟΥΜΠΙ ΕΞΟΔΟΥ**

### 3.8 Αρχιτεκτονική εφαρμογής και βάση δεδομένων

Ο Κώδικας της εφαρμογής απαρτίζεται από 12 κομμάτια, εκ των οποίων κάποια απ' αυτά έχουν πλήρη συσχέτιση μεταξύ τους και επικοινωνούν για την σωστή λειτουργία της εφαρμογής, όπως φαίνεται στο παρακάτω σχήμα.



ΣΧΗΜΑ 1 – ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΦΑΡΜΟΓΗΣ

- Κώδικας LOADING SCREEN : Περιέχει ένα εικονίδιο της εφαρμογής και μια μπάρα φόρτωσης με delay τεσσάρων δευτερολέπτων, έτσι ώστε να εισέρχεται ομαλά ο χρήστης στην εφαρμογή.
- Κώδικας LOG IN : Περιέχει κώδικα που σου επιτρέπει να συμπληρώσεις τα στοιχεία σου ώστε να μπεις στην εφαρμογή αν και εφόσον είναι έγκυρα. Περιέχει ένα log in button `setOnClickListener` το οποίο όταν πατηθεί κάνει intent σε new activity δηλαδή στο activity του home screen και ένα register button το οποίο όταν πατηθεί κάνει intent στο activity της εγγραφής χρήστη.
- Κώδικας REGISTRATION : Περιέχει τον κώδικα για όλα τα πεδία που πρέπει να συμπληρώσει ο χρήστης για να εγγραφεί. Επίσης είναι το κομμάτι του κώδικα το οποίο επικοινωνεί με το Authentication του Firebase. Ορίζει κριτήρια για τα πεδία συμπλήρωσης όπως για το email να είναι της μορφής `@__.com` και ο κωδικός να ξεπερνά τους 6 αριθμούς ή χαρακτήρες. Επίσης διαθέτει

progress bar και όταν ολοκληρωθεί η εγγραφή του χρήστη επιστρέφει αυτόματα στο activity του LOG IN.

- Κώδικας ΑΡΧΙΚΗΣ ΟΘΟΝΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ : Διαθέτει ένα πράσινο κουμπί reportButton setOnClickListener το οποίο όταν πατηθεί κάνει intent στο GoogleMapsActivity. Διαθέτει επίσης και το navigation bar με τις επιλογές home, profile, settings και όποιο πατηθεί κάνει το αντίστοιχο intent
- Κώδικας ΓΙΑ ΧΑΡΤΕΣ ΚΑΙ ΤΟΠΟΘΕΣΙΑ ΧΡΗΣΤΗ : Σε αυτό το κομμάτι κάναμε implement τους χάρτες της Google και προσθέσαμε κάποια APIs όπως είναι το GeoFire ώστε να βρούμε τις συντεταγμένες latitude και longitude του χρήστη οι οποίες απεικονίζονται με ένα marker. Απαραίτητη ήταν και η προσθήκη των location Services καθώς τοποθετήθηκε και ένα button NEXT το οποίο οδηγεί στο Activity της αναφοράς.
- Κώδικας ΑΝΑΦΟΡΑΣ : Και σε αυτό το activity υπάρχει άμεση επικοινωνία με την βάση δεδομένων και συγκεκριμένα με το FirebaseStorage όπου αποτελεί τον χώρο που τοποθετούνται όλα τα reports. Επίσης έχουμε γράψει και τον κώδικα του URI για να μπορέσουμε να επιλέξουμε και να ανεβάσουμε φωτογραφία. Επίσης υπάρχουν 2 button, το 1<sup>ο</sup> είναι το button upload το οποίο ουσιαστικά κάνει upload το Report εμφανίζοντας μια progress bar, το 2<sup>ο</sup> button είναι το show Reports το οποίο οδηγεί τον χρήστη στο activity προβολής των αναφορών.
- Κώδικας NOTIFICATIONS : Για το συγκεκριμένο κομμάτι του κώδικα χρησιμοποιήθηκε το Node με την γλώσσα JavaScript κατά κύριο λόγο. Δημιουργήθηκε ουσιαστικά ένα function στο firebase το οποίο κάνει trigger από τον κώδικα των notifications και κάθε φορά που πατάει ο χρήστης το κουμπί Report τότε άμεσα στέλνεται από την βάση δεδομένων μια ειδοποίηση προς όλους τους χρήστες της εφαρμογής. Επίσης χρειάστηκαν και λίγες γραμμές κώδικα στο Android studio ώστε να επικοινωνήσει το υπόλοιπο project με τα notifications.
- Κώδικας UPLOAD ΦΩΤΟΓΡΑΦΙΑΣ : Πέρα από τον κώδικα της φωτογραφίας που γράφτηκε στο πεδίο της αναφοράς, χρειάστηκε και ξεχωριστός κώδικας για την φωτογραφία έτσι ώστε να της ορίσουμε τα σχόλια του χρήστη και το id του, για να υπάρχει μια σωστή οργάνωση στην βάση δεδομένων όπου αποθηκεύεται.
- Κώδικας ΠΡΟΒΟΛΗΣ ΟΛΩΝ ΤΩΝ ΑΝΑΦΟΡΩΝ : Αποτελείται από ένα RecyclerView το οποίο μας δίνει την δυνατότητα να συλλεχθούν και να εμφανιστούν όλες οι αναφορές που έχουν πραγματοποιηθεί. Επίσης διαθέτει το bottom navigation bar ώστε να μπορεί ο χρήστης να επιστρέψει σε οποιοδήποτε activity.
- Κώδικας IMAGE ADAPTER : Έχει άμεση συσχέτιση με τον προηγούμενο κώδικα που αναφέρθηκε, μπορεί να θεωρηθεί και το 2<sup>ο</sup> κομμάτι του, αφού ο κώδικας του image adapter είναι απαραίτητος ώστε να γίνει σωστά το upload και σε λίστα στην βάση δεδομένων, αλλά επίσης φτιάχνει και την θέση με crops και fits ώστε να προσαρμόζεται σωστά στην κάθε διαφορετική οθόνη που εκτελείται η εφαρμογή.
- Κώδικας SETTINGS : Αντλεί από το firebase και εμφανίζει στον χρήστη της πληροφορίες του που έχει δηλώσει κατά την εγγραφή του. Διαθέτει όπως και τα περισσότερα activities το navigation bar για εύκολη επιστροφή και τέλος περιέχει και

ένα κουμπί LOG OUT για την έξοδο του χρήστη από την εφαρμογή, το οποίο πρακτικά οδηγεί στο activity του LOG IN.

## Βάση δεδομένων Firebase

Χρησιμοποιήθηκε το Realtime Database του Firebase με την μορφή του JSON tree όπως βλέπουμε παρακάτω. Ο κάθε χρήστης διαθέτει ένα ξεχωριστό deviceToken το οποίο μας εξυπηρετεί για τα notifications(δηλαδή να έρχεται ειδοποίηση σε όλα τα διαφορετικά device Tokens της εφαρμογής εκτός απ' αυτόν που έκανε την αναφορά). Επίσης ο κάθε user διαθέτει το δικό του email και password.

Επιπλέον στην βάση δεδομένων μπαίνουν και οι αναφορές(uploads), οι οποίες διαθέτουν ένα imageUrl της εικόνας που ανέβασε ο χρήστης, το latitude και longitude(συντεταγμένες τοποθεσίας του χρήστη), τα σχόλια της αναφοράς(name) καθώς και το userId αφού θα πρέπει να γνωρίζουμε ποιος χρήστης έκανε την κάθε αναφορά.

- **Users**

- **1eCywAumyiYHse1qTLtpypsdRGY2**

- **deviceToken:**

- **email:**

- **fullName:**

- **uploads**

- **-MbfVTMVwTxgqGAFWQ0c**

- **imageUrl:**

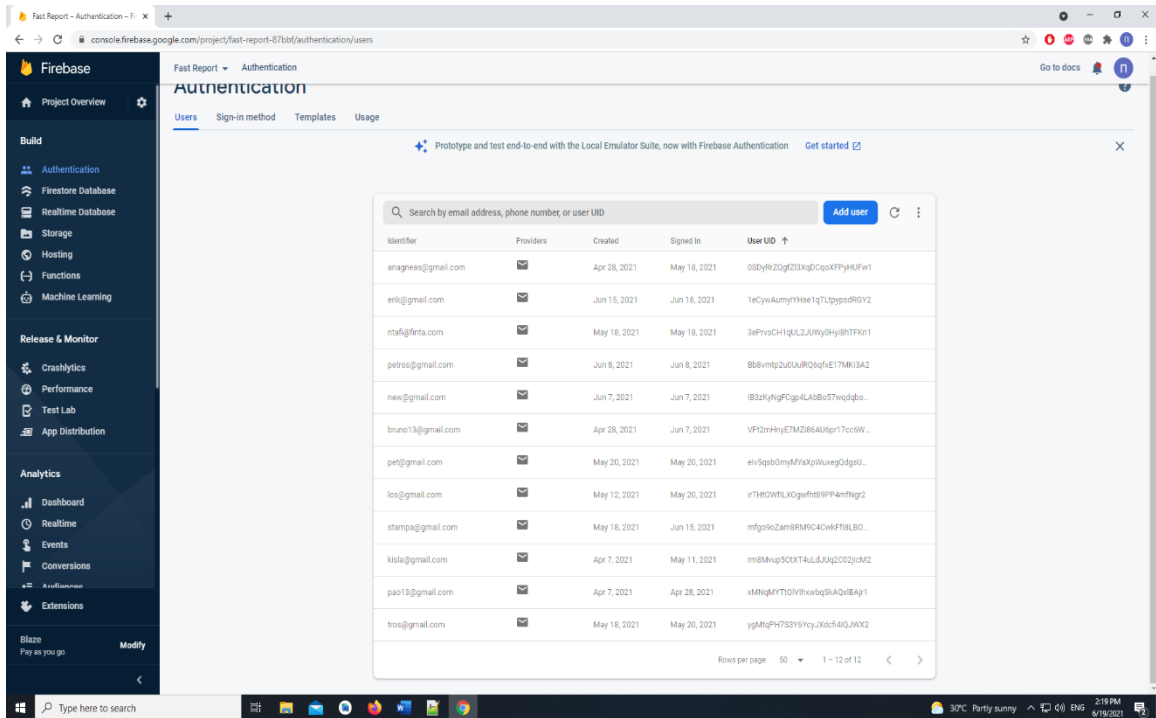
- **latitude:**

- **longitude:**

- **name:**

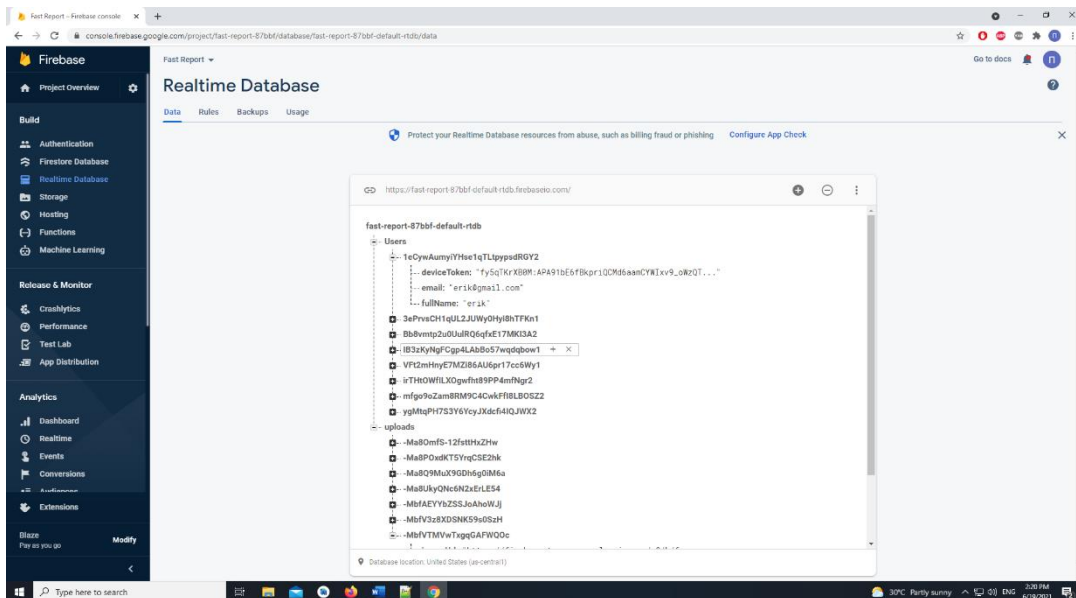
- **userId:**

Στην παρακάτω εικόνα απεικονίζεται η λίστα με τους εγγεγραμμένους χρήστες της εφαρμογής καθώς επίσης φαίνεται και το πότε έκανε την εγγραφή αλλά και το πότε συνδέθηκε για τελευταία φορά στην εφαρμογή.



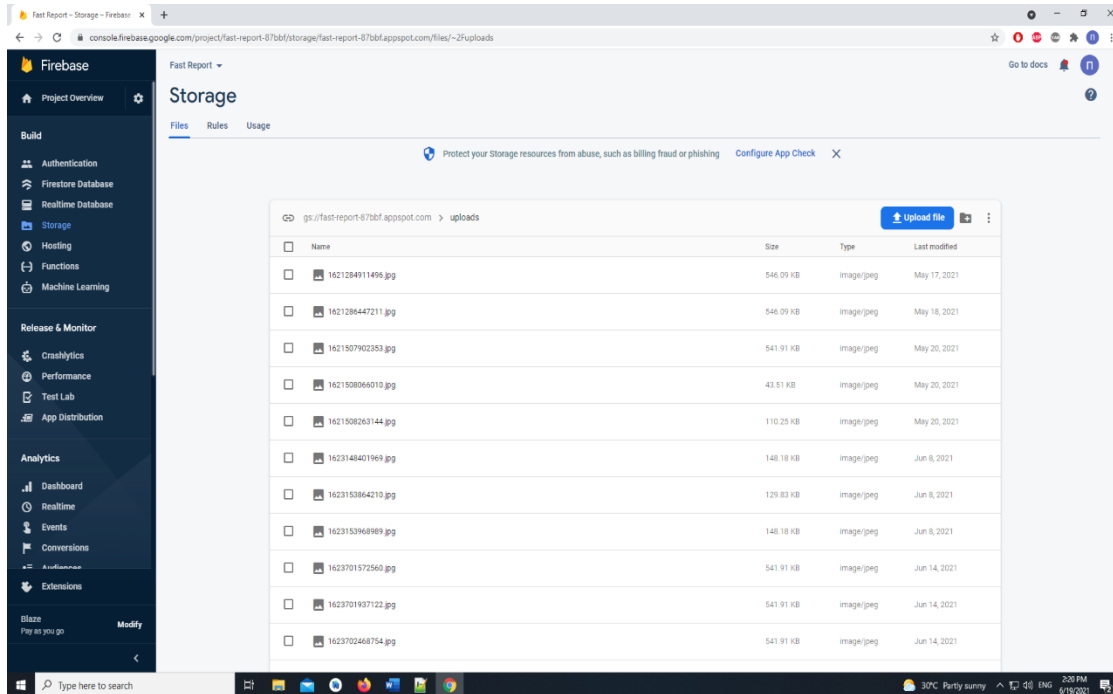
ΕΙΚΟΝΑ 20 - ΕΓΓΕΓΡΑΜΕΝΟΙ ΧΡΗΣΤΕΣ

Παρακάτω απεικονίζεται το JSON tree όπως αναφέραμε προηγουμένως με τα απαραίτητα στοιχεία του κάθε χρήστη.



ΕΙΚΟΝΑ 21 - ΑΝΑΦΟΡΕΣ ΧΡΗΣΤΩΝ

Τέλος, στο πεδίο του Storage έχουν αποθηκευτεί όλα τα uploads(αναφορές) του κάθε χρήστη και δίνονται πληροφορίες όπως για το μέγεθος της φωτογραφίας που τράβηξε, τα σχόλια που έκανε αλλά και το πότε έγινε η αναφορά.



ΕΙΚΟΝΑ 22 - ΟΙ ΦΩΤΟΓΡΑΦΙΕΣ ΓΙΑ ΤΙΣ ΑΝΑΦΟΡΕΣ

## ΚΕΦΑΛΑΙΟ 4

### 4.1 Συμπεράσματα

Η παρούσα διπλωματική εργασία παρουσιάζει λύσεις αντιμετώπισης και προστασίας του ανθρώπου από απρόσμενα ή έκτακτα περιστατικά, μέσω του σχεδιασμού και ανάπτυξης εφαρμογών για φορητές συσκευές. Τονίζει και αναλύει την επικινδυνότητα των φυσικών καταστροφών και των συχνών ατυχημάτων που συμβαίνουν στους δρόμους. Επίσης αναφέρει τρόπους και εφαρμογές που απαιτούν συνεργασία για την αντιμετώπιση τέτοιων καταστάσεων και αναλύει το Risk Management στο πεδίο των εφαρμογών. Τέλος σχεδιάστηκε μια εφαρμογή η οποία συνδυάζει την συνεργασία (cooperation) με την έκτακτη ειδοποίηση/ενημέρωση (emergency alert) προς όφελος της προστασίας του ανθρώπου.

### 4.2 Μελλοντικές προσθήκες και αλλαγές

Υπάρχουν περιθώρια βελτίωσης της εφαρμογής ώστε να γίνει πιο εύχρηστη και πιο αξιόπιστη. Κάποιες προσθήκες που σχεδιάζεται να πραγματοποιηθούν στο μέλλον είναι οι εξής:

- Να προστεθεί κάτω από την κάθε αναφορά η ημερομηνία/ώρα, το user id, και η περιοχή όπου συνέβη.
- Να προστεθεί φίλτρο αναζήτησης, έτσι ώστε ο κάθε χρήστης να μπορεί να επιλέξει μόνο τις περιοχές που τον ενδιαφέρουν.
- Να προστεθεί μπάρα αξιολόγησης της κάθε αναφοράς και κατ' επέκταση του κάθε χρήστη ώστε να υπάρχουν μόνο αξιόπιστες αναφορές.
- Να προστεθεί ένα home button στην φόρμα συμπλήρωσης της αναφοράς, καθώς δεν δίνεται η δυνατότητα επιστροφής λόγω απώλειας του nav bar.

Αλλαγές που θα χρησίμευαν στην οπτική και την λειτουργία της εφαρμογής είναι:

- Στο πεδίο όλων των Reports, να εμφανίζεται πρώτο, το τελευταίο Report που έγινε με βάση την ώρα που έγινε.
- Να χρησιμοποιηθεί το Firestore Database αντί του Realtime Database, για καλύτερη οργάνωση συλλογής δεδομένων.



## ΑΝΑΦΟΡΕΣ

- [1] Yordan, [https://www.gsmarena.com/more\\_than\\_5b\\_people\\_use\\_mobile\\_devices-news-25631.php](https://www.gsmarena.com/more_than_5b_people_use_mobile_devices-news-25631.php) , 15 June 2017
- [2] <http://old.uth.gr/main/help/help-desk/internet/internet3.html> , December 1997
- [3] <https://aginginplace.org/technology-in-our-life-today-and-how-it-has-changed>, Technology in our life today and how it has changed, May 2021
- [4] <https://www.m-stat.gr/el/history-of-mobile-phones/> , The History of mobile phones,10 July 2015
- [5] Matt Strain, <https://www.theguardian.com/media-network/2015/feb/13/history-mobile-apps-future-interactive-timeline> , History of mobile Apps, February 2013
- [6] L.Rachel, <https://www.engadget.com/2016-10-30-how-technology-has-changed-our-lives.html>, How technology has changed our lives, 30 October 2016
- [7] Prasun Dewan, <http://www.cs.unc.edu/~dewan/290/s97/notes/intro/node2.html>, Definition of Collaborative Application, January 2013
- [8] <https://www.smartcity.press/smart-city-apps/>, Smart city apps improving life in cities,26 December 2019
- [9] <https://www.htc.com/us/go/power-to-give/>, Plug in, be a part of the future
- [10] <https://www.novoville.com/el/>
- [11] Elan Frantz, <https://thelastmile.gotennapro.com/top-10-mobile-apps-for-emergency-response-teams/>, Top 10 apps for emergency response teams, 13 February 2020
- [12] Hannah Ritchie and Max Roser, <https://ourworldindata.org/natural-disasters>, Natural Disasters, 2014
- [13] Μετσόβιο κέντρο διεπιστημονικής έρευνας, <https://www.mirc.ntua.gr/natural-disasters-metsovo/natural-disasters>, Φυσικές καταστροφές
- [14] <https://wildfirealert.com/wfa-who-can-benefit>
- [15] Victoria Feng, <https://www.wired.com/story/best-emergency-apps/>, Emergency preparedness apps to keep on your phone, 4 May 2021
- [16] <https://www.redcross.org/get-help/how-to-prepare-for-emergencies/types-of-emergencies/earthquake.html>

## ΠΑΡΑΡΤΗΜΑ Ι

Κώδικας για το Loading screen

```
package com.example.fastreport;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;

public class LoadingScreen extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_loading_screen);

        // Handler handler = new Handler();
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                //Intent homeIntent = new Intent
                (LoadingScreen.this,MainActivity.class);
                startActivity(new Intent
                (LoadingScreen.this,MainActivity.class));
                finish();
            }
        },4000);
    }
}
```

## Κώδικας για το LOG IN

```

package com.example.fastreport;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.os.Handler;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

public class MainActivity extends AppCompatActivity implements
View.OnClickListener {

    private TextView register;
    private EditText editTextEmail, editTextPassword;
    private Button logIn;
    CheckBox remember;

    private FirebaseAuth mAuth;
    private ProgressBar progressBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        register = (TextView) findViewById(R.id.register);
        register.setOnClickListener(this);

        logIn = (Button) findViewById(R.id.logIn);
        logIn.setOnClickListener(this);

        editTextEmail = (EditText) findViewById(R.id.emailAddress);
        editTextPassword = (EditText) findViewById(R.id.pass);

        remember = findViewById(R.id.RememberMe);

        progressBar = (ProgressBar) findViewById(R.id.progressBar2);

        mAuth = FirebaseAuth.getInstance();

```

```
@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.register:
            startActivity(new Intent(this, Registration.class));
            break;

        case R.id.logIn:
            userLogIn();
            break;
    }
}
```

```

private void userLogIn() {

    String email = editTextEmail.getText().toString().trim();
    String password =
editTextPassword.getText().toString().trim();

    if(email.isEmpty()) {
        editTextEmail.setError("Email is required");
        editTextEmail.requestFocus();
        return;
    }

    if(!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
        editTextEmail.setError("Please provide valid email");
        editTextEmail.requestFocus();
        return;
    }

    if(password.isEmpty()) {
        editTextPassword.setError("Password is required");
        editTextPassword.requestFocus();
        return;
    }

    if(password.length() < 6) {
        editTextPassword.setError("Minimum password length is 6
characters");
        editTextPassword.requestFocus();
        return;
    }

    progressBar.setVisibility(View.VISIBLE);

    mAuth.signInWithEmailAndPassword(email,password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {

            if(task.isSuccessful()){
                //redirect to user profile
                startActivity(new
Intent(MainActivity.this,HomeActivity.class));
            }else {
                Toast.makeText(MainActivity.this, "Failed to
login! Please check your credentials", Toast.LENGTH_LONG).show();
            }
        }
    });
}
}

```

## Κώδικας για την εγγραφή του χρήστη

```

package com.example.fastreport;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCanceledListener;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.iid.FirebaseInstanceId;
import com.google.firebase.iid.InstanceIdResult;

import java.util.regex.Pattern;

public class Registration extends AppCompatActivity implements
View.OnClickListener {

    private TextView banner, registerUser;
    private EditText editTextFullName, editTextEmail, editTextPassword;
    private ProgressBar progressBar;

    private FirebaseAuth mAuth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registration);

        mAuth = FirebaseAuth.getInstance();

        banner = (TextView) findViewById(R.id.banner);
        banner.setOnClickListener(this);

        registerUser = (Button) findViewById(R.id.registerUser);
        registerUser.setOnClickListener(this);

        editTextFullName = (EditText) findViewById(R.id.fullName);
        editTextEmail = (EditText) findViewById(R.id.email);
        editTextPassword = (EditText) findViewById(R.id.password);

        progressBar = (ProgressBar) findViewById(R.id.progressBar);
    }
}

```

```

@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.banner:
            startActivity(new Intent(this,MainActivity.class));
            break;
        case R.id.registerUser:
            registererUser();
            break;
    }
}

private void registererUser() {

    String email = editTextEmail.getText().toString().trim();
    String fullName = editTextFullName.getText().toString().trim();
    String password = editTextPassword.getText().toString();

    if(fullName.isEmpty()) {
        editTextFullName.setError("Full name is required");
        editTextFullName.requestFocus();
        return;
    }

    if(email.isEmpty()) {
        editTextEmail.setError("Email is required");
        editTextEmail.requestFocus();
        return;
    }

    if(!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
        editTextEmail.setError("Please provide valid email");
        editTextEmail.requestFocus();
        return;
    }

    if(password.isEmpty()) {
        editTextPassword.setError("Password is required");
        editTextPassword.requestFocus();
        return;
    }

    if(password.length()<6){
        editTextPassword.setError("Minimum password length should be 6
characters.");
        editTextPassword.requestFocus();
        return;
    }
}

```





Κώδικας για την αρχική οθόνη με το button 'MAKE A REPORT'

```

package com.example.fastreport;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.app.Dialog;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GoogleApiAvailability;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.material.bottomnavigation.BottomNavigationView;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.iid.FirebaseInstanceId;
import com.google.firebase.iid.InstanceIdResult;

public class HomeActivity extends AppCompatActivity {

    private Button reportButton;

    private static final String TAG = "HomeActivity";

    private static final int ERROR_DIALOG_REQUEST = 9001;

    private FirebaseUser user;
    private DatabaseReference reference;
    private String userID;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);

        if(isServicesOK()){
            init();
        }
        setToken();
    }

```

```

BottomNavigationView bottomNavigationView =
findViewById(R.id.bottom_navigation);

    bottomNavigationView.setSelectedItemId(R.id.ic_home);

    bottomNavigationView.setOnNavigationItemSelectedListener(new
BottomNavigationView.OnNavigationItemSelectedListener() {
        @Override
        public boolean onNavigationItemSelected(@NonNull MenuItem
menuItem) {
            switch(menuItem.getItemId()){
                case R.id.ic_home:

                    return true;

                case R.id.ic_person:
                    startActivity(new
Intent(getApplicationContext(), ProfileActivity.class));
                    overridePendingTransition(0,0);
                    return true;

                case R.id.ic_settings:
                    startActivity(new Intent(getApplicationContext(),
SettingsActivity.class));
                    overridePendingTransition(0,0);
                    return true;
            }
            return false;
        }
    });
}

public boolean isServicesOK(){
    Log.d(TAG, "isServicesOK:checking google services version");

    int available =
GoogleApiAvailability.getInstance().isGooglePlayServicesAvailable(Home
Activity.this);

    if(available== ConnectionResult.SUCCESS){
        //everything is fine and the user can make map requests
        Log.d(TAG, "isServicesOK:Google Play Services is working");
        return true;
    }
    else
if(GoogleApiAvailability.getInstance().isUserResolvableError(available))
{
        //an error occurred but we can resolve it
        Log.d(TAG, "isServicesOK:an error occurred but we can fix
it");
        Dialog dialog =
GoogleApiAvailability.getInstance().getErrorDialog(HomeActivity.this,
available, ERROR_DIALOG_REQUEST);
        dialog.show();
    }else{
        Toast.makeText(this, "You can't make map requests",
Toast.LENGTH_SHORT).show();
    }
    return false;
}
}

```

```

private void setToken1() {
FirebaseInstanceId.getInstance().getInstanceId().addOnSuccessListener
(new OnSuccessListener<InstanceIdResult>() {
@Override
public void onSuccess(InstanceIdResult instanceIdResult)
{
reference =
FirebaseDatabase.getInstance().getReference("Users");
userID = user.getId();

reference.child(userID).addListenerForSingleValueEvent(new
ValueEventListener() {
@Override
public void onDataChange(@NonNull DataSnapshot
snapshot) {
Toast.makeText(HomeActivity.this, "1",
Toast.LENGTH_LONG).show();
User userProfile =
snapshot.getValue(User.class);
userProfile.deviceToken =
instanceIdResult.getToken();
}

@Override
public void onCancelled(@NonNull DatabaseError
error) {

}

});

});

}

private void setToken() {

}
}

```

## Κώδικας για τα Google maps και user location

```

package com.example.fastreport;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import com.firebase.geofire.GeoFire;
import com.firebase.geofire.GeoLocation;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

public class GoogleMapsActivity extends AppCompatActivity {
    SupportMapFragment supportMapFragment;
    FusedLocationProviderClient client;
    private Button button;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_google_maps);
        supportMapFragment = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.map);
        client = LocationServices.getFusedLocationProviderClient(this);

        if (ActivityCompat.checkSelfPermission(GoogleMapsActivity.this,
Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED) {
            getCurrentLocation();
        }else {
            ActivityCompat.requestPermissions(GoogleMapsActivity.this,new
String[]{Manifest.permission.ACCESS_FINE_LOCATION},44);
        }
        button = findViewById(R.id.ButtonToImage);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                openImageActivity();
            }
        });
    }
}

```

```

private void openImageActivity() {
    Intent intent = new
Intent (GoogleMapsActivity.this, ImageActivity.class);
    startActivity(intent);
}

private void getCurrentLocation() {

    if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
        // TODO: Consider calling
        //     ActivityCompat#requestPermissions
        // here to request the missing permissions, and then
        // overriding
        //     public void onRequestPermissionsResult(int requestCode,
String[] permissions,
        //                                     int[]
grantResults)
        // to handle the case where the user grants the permission.
        See the documentation
        // for ActivityCompat#requestPermissions for more details.
        return;
    }
    Task<Location> task = client.getLastLocation();
    task.addSuccessListener(new OnSuccessListener<Location>() {
        @Override
        public void onSuccess(Location location) {

            if (location != null) {
                supportMapFragment.getMapAsync(new
OnMapReadyCallback() {
                    @Override
                    public void onMapReady(GoogleMap googleMap) {
                        LatLng latLng = new
LatLng(location.getLatitude(), location.getLongitude());

                        MarkerOptions options = new
MarkerOptions().position(latLng).title("I am there");

                        googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, 10))
;

                        googleMap.addMarker(options);
                    }
                });
            }
        }
    });
}

```

```
String userID = FirebaseAuth.getInstance().getCurrentUser().getUid();
    DatabaseReference ref =
FirebaseDatabase.getInstance().getReference("UserLocation");

        GeoFire geoFire = new GeoFire(ref);

        geoFire.setLocation(userID, new
GeoLocation(location.getLatitude(), location.getLongitude()));
    }
});

}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
    if (requestCode == 44){
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {

            getLocation();
        }
    }
}

@Override
protected void onStop() {
    super.onStop();
    String userID =
FirebaseAuth.getInstance().getCurrentUser().getUid();
    DatabaseReference ref =
FirebaseDatabase.getInstance().getReference("UserLocation");

    GeoFire geoFire = new GeoFire(ref);

    geoFire.removeLocation(userID);
}
}
```

## Κώδικας για φόρμα αναφοράς

```

package com.example.fastreport;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import android.Manifest;
import android.content.ContentResolver;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.net.Uri;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.view.View;
import android.webkit.MimeTypeMap;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;
import com.bumptech.glide.util.LogTime;
import com.firebase.geofire.GeoLocation;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.tasks.Continuation;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.OnProgressListener;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.StorageTask;
import com.google.firebase.storage.UploadTask;
import com.google.gson.JsonObject;
import com.squareup.picasso.Picasso;

public class ImageActivity extends AppCompatActivity {

    private static final int PICK_IMAGE_REQUEST = 1;
    private Button mButtonChooseImage;
    private Button mButtonUpload;
    private TextView mTextViewShowUploads;
    private EditText mEditTextFileName;
    private ImageView mImageView;
    private ProgressBar mProgressBar;

```

```

private Uri mImageUri;

private StorageReference mStorageRef;
private DatabaseReference mDatabaseRef;

private FirebaseUser user;
private String userID;

private StorageTask mUploadTask;

private static final String TAG = ImageActivity.class.getName();

Boolean successLoc = false;
Location currentLocation;

private FusedLocationProviderClient fusedLocationProviderClient;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_image);

    mButtonChooseImage = findViewById(R.id.button_choose_image);
    mButtonUpload = findViewById(R.id.button_upload);
    mTextViewShowUploads = findViewById(R.id.text_view_show_uploads);
    mEditTextFileName = findViewById(R.id.edit_text_file_name);
    mImageView = findViewById(R.id.image_view);
    mProgressBar = findViewById(R.id.progress_bar);

    mStorageRef =
    FirebaseStorage.getInstance().getReference("uploads");
    mDatabaseRef =
    FirebaseDatabase.getInstance().getReference("uploads");

    user = FirebaseAuth.getInstance().getCurrentUser();
    userID = user.getId();

    mButtonChooseImage.setOnClickListener(new View.OnClickListener()
    {
        @Override
        public void onClick(View v) {
            openFileChooser();
        }
    });
}

```



```

mButtonUpload.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (mUploadTask != null && mUploadTask.isInProgress()) {
            Toast.makeText(ImageActivity.this, "Report in
Progress", Toast.LENGTH_SHORT).show();
        }else {
            uploadFile();
        }
    }
});

mTextViewShowUploads.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        openImagesActivity();
    }
});
}

private void openFileChooser() {
    Intent intent = new Intent();
    intent.setType("image/*");
    intent.setAction(Intent.ACTION_GET_CONTENT);
    startActivityForResult(intent,PICK_IMAGE_REQUEST);
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK
&& data != null && data.getData() != null) {
        mImageUri = data.getData();

        Picasso.get().load(mImageUri).into(mImageView);
    }
}

private String getFileExtension(Uri uri) {
    ContentResolver cR = getContentResolver();
    MimeTypeMap mime = MimeTypeMap.getSingleton();
    return mime.getExtensionFromMimeType(cR.getType(uri));
}

```

```

private void uploadFile() {

if (mImageUri != null) {
    mProgressBar.setVisibility(View.VISIBLE);
    final StorageReference fileReference =
mStorageRef.child(System.currentTimeMillis()
        + "." + getFileExtension(mImageUri));
    currentLocation = getDeviceLocation();
    fileReference.putFile(mImageUri).continueWithTask(
        new Continuation<UploadTask.TaskSnapshot,
Task<Uri>>() {
            @Override
            public Task<Uri> then(@NonNull
Task<UploadTask.TaskSnapshot> task) throws Exception {
                if (!task.isSuccessful()) {
                    throw task.getException();
                }
                return fileReference.getDownloadUrl();
            }
        })
        .addOnCompleteListener(new OnCompleteListener<Uri>()
{
            @Override
            public void onComplete(@NonNull Task<Uri> task) {
                if (task.isSuccessful()) {
                    Uri downloadUri = task.getResult();
                    Upload upload = new
Upload(mEditTextFileName.getText().toString().trim(),
downloadUri.toString(), userID,
currentLocation.getLatitude(), currentLocation.getLongitude());
                    mDatabaseRef.push().setValue(upload);

mProgressBar.setVisibility(View.INVISIBLE);
                    Toast.makeText(ImageActivity.this,
"Upload successful", Toast.LENGTH_LONG).show();
                }
                else { Toast.makeText(ImageActivity.this,
"upload failed: " + task.getException().getMessage(),
Toast.LENGTH_LONG).show();
                }
            }
        })
        .addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Toast.makeText(ImageActivity.this,
e.getMessage(), Toast.LENGTH_LONG).show();
            }
        });
    } else {
        Toast.makeText(this, "No file selected",
Toast.LENGTH_LONG).show();
    }
}
}

```

```

private void openImagesActivity() {
    Intent intent = new
Intent(ImageActivity.this, ProfileActivity.class);
    startActivity(intent);
}

private Location getDeviceLocation() {
    Log.d(TAG, "getDeviceLocation:getting the current devices
location");
    fusedLocationProviderClient =
LocationServices.getFusedLocationProviderClient(this);
    try {
        Task location =
fusedLocationProviderClient.getLastLocation();
        location.addOnCompleteListener(new
OnCompleteListener() {
            @Override
            public void onComplete(@NonNull Task task) {
                if (task.isSuccessful()) {
                    Log.d(TAG, "onComplete:found location!");
                    currentLocation = (Location)
task.getResult();
                    successLoc = true;
                } else {
                    Log.d(TAG, "onComplete:current location
is null");
                    Toast.makeText(ImageActivity.this,
"unable to get current location", Toast.LENGTH_SHORT).show();
                }
            }
        });
    } catch (SecurityException e) {
        Log.e(TAG, "getDeviceLocation: SecurityException" +
e.getMessage());
    }
    if(successLoc)
        return currentLocation;
    else
        return null;
}
}

```

## Κώδικας προβολής όλων των reports

```

package com.example.fastreport;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.material.bottomnavigation.BottomNavigationView;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;
import java.util.List;

public class ProfileActivity extends AppCompatActivity {

    private RecyclerView mRecyclerView;
    private ImageAdapter mAdapter;

    private DatabaseReference mDatabaseRef;
    private List<Upload> mUploads;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_profile);

        mRecyclerView = findViewById(R.id.recycler_view);
        mRecyclerView.setHasFixedSize(true);
        mRecyclerView.setLayoutManager(new LinearLayoutManager(this));

        mUploads = new ArrayList<>();
        mDatabaseRef =
        FirebaseDatabase.getInstance().getReference("uploads");
        mDatabaseRef.addValueEventListener(new ValueEventListener() {

```

```

@Override
public void onDataChange(DataSnapshot dataSnapshot) {
    for (DataSnapshot postSnapshot : dataSnapshot.getChildren())
    {
        Upload upload = postSnapshot.getValue(Upload.class);
        mUploads.add(upload);
    }
    mAdapter = new ImageAdapter(ProfileActivity.this, mUploads);
    mRecyclerView.setAdapter(mAdapter);
}
@Override
public void onCancelled(DatabaseError databaseError) {
    Toast.makeText(ProfileActivity.this,
databaseError.getMessage(), Toast.LENGTH_SHORT).show();
}
});
    BottomNavigationView bottomNavigationView =
findViewById(R.id.bottom_navigation);

    bottomNavigationView.setSelectedItemId(R.id.ic_person);

    bottomNavigationView.setOnNavigationItemSelectedListener(new
BottomNavigationView.OnNavigationItemSelectedListener() {
        @Override
        public boolean onNavigationItemSelected(@NonNull MenuItem
menuItem) {
            switch (menuItem.getItemId()) {
                case R.id.ic_home:
                    startActivity(new
Intent(getApplicationContext(), HomeActivity.class));
                    overridePendingTransition(0, 0);
                    return true;

                case R.id.ic_person:

                    return true;

                case R.id.ic_settings:
                    startActivity(new
Intent(getApplicationContext(), SettingsActivity.class));
                    overridePendingTransition(0, 0);
                    return true;
            }
            return false;
        }
    });
}
}

```

## Κώδικας για τα settings

```

package com.example.fastreport;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.material.bottomnavigation.BottomNavigationView;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class SettingsActivity extends AppCompatActivity {

    private Button logout;

    private FirebaseUser user;
    private DatabaseReference reference;
    private String userID;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_settings);

        BottomNavigationView bottomNavigationView =
        findViewById(R.id.bottom_navigation);

        bottomNavigationView.setSelectedItemId(R.id.ic_settings);

        logout = (Button) findViewById(R.id.logOutBtn);

        logout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                FirebaseAuth.getInstance().signOut();
                startActivity(new
                Intent(SettingsActivity.this, MainActivity.class));
            }
        });
    }
}

```

```

user = FirebaseAuth.getInstance().getCurrentUser();
reference = FirebaseDatabase.getInstance().getReference("uploads");
userID = user.getUid();

final TextView emailTextView = (TextView)
findViewById(R.id.emailAddressText);
final TextView fullNameTextView = (TextView)
findViewById(R.id.fullNameText);

reference.child(userID).addListenerForSingleValueEvent(new
 ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        User userProfile = snapshot.getValue(User.class);

        if (userProfile != null) {
            String fullName = userProfile.fullName;
            String email = userProfile.email;

            fullNameTextView.setText(fullName);
            emailTextView.setText(email);
        }
    }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {

        Toast.makeText(SettingsActivity.this, "Something wrong
happened!", Toast.LENGTH_LONG).show();
    }
});

bottomNavigationView.setOnNavigationItemSelectedListener(new
 BottomNavigationView.OnNavigationItemSelectedListener() {
    @Override
    public boolean onNavigationItemSelectedListener(@NonNull MenuItem
menuItem) {
        switch (menuItem.getItemId()) {
            case R.id.ic_home:
                startActivity(new
Intent(getApplicationContext(), HomeActivity.class));
                overridePendingTransition(0, 0);
                return true;

            case R.id.ic_person:
                startActivity(new
Intent(getApplicationContext(), ProfileActivity.class));
                overridePendingTransition(0, 0);
                return true;

            case R.id.ic_settings:
                return true;
        }
        return false;
    }
});
}
}
}

```

Κώδικας για τον κάθε χρήστη

```
package com.example.fastreport;  
  
public class User {  
  
    public String fullName,email,deviceToken;  
  
    public User() {  
  
    }  
  
    public User(String fullName,String email,String deviceToken) {  
        this.fullName=fullName;  
        this.email=email;  
        this.deviceToken=deviceToken;  
    }  
  
}
```



## Κώδικας για τα Uploads

```
package com.example.fastreport;

import com.firebase.geofire.GeoLocation;

public class Upload {
    public String name;
    public String userId;
    public String imageUrl;
    public Double longitude;
    public Double latitude;
    public Upload() {

    }

    public Upload(String name, String imageUrl, String userId, Double
longitude, Double latitude) {
        if (name.trim().equals("")) {
            name = "No name";
        }

        this.name = name;
        this.userId = userId;
        this.imageUrl = imageUrl;
        this.longitude=longitude;
        this.latitude=latitude;
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getImageUrl() {
        return this.imageUrl;
    }
    public void setImageUrl(String imageUrl) {
        this.imageUrl = imageUrl;
    }
}
```

## Κώδικας για τα notifications

```

package com.example.fastreport;

import android.util.Log;
import android.widget.Toast;

import androidx.annotation.NonNull;

import com.google.firebase.messaging.FirebaseMessagingService;
import com.google.firebase.messaging.RemoteMessage;

import org.jetbrains.annotations.NotNull;

import static android.content.ContentValues.TAG;

public class MyFirebaseMessagingService extends
FirebaseMessagingService{
    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {
        // ...
        // TODO(developer): Handle FCM messages here.
        // Not getting messages here? See why this may be:
https://goo.gl/39bRNJ
        Log.d(TAG, "From: " + remoteMessage.getFrom());

        // Check if message contains a data payload.
        if (remoteMessage.getData().size() > 0) {
            Log.d(TAG, "Message data payload: " +
remoteMessage.getData());

        }

        // Check if message contains a notification payload.
        if (remoteMessage.getNotification() != null) {
            Log.d(TAG, "Message Notification Body: " +
remoteMessage.getNotification().getBody());
        }

        // Also if you intend on generating your own notifications as
a result of a received FCM
        // message, here is where that should be initiated. See
sendNotification method below.
    }
}

```

```
// The Cloud Functions for Firebase SDK to create Cloud Functions and setup triggers.
```

```
const functions = require('firebase-functions');
```

```
// The Firebase Admin SDK to access Firestore.
```

```
const admin = require('firebase-admin');
```

```
admin.initializeApp();
```

```
exports.sendNotif = functions.database.ref('/uploads/{pushId}/name')
```

```
.onCreate((snapshot, context) => {
```

```
  const name = snapshot.val();
```

```
  console.log('Name:', name);
```

```
  //functions.logger.log('Db.ref:',
```

```
admin.database().ref(`/Users/{userId}/deviceToken`).once('value'));
```

```
  return admin.database().ref("Users")
```

```
.once("value")
```

```
.then((snapshot) => {
```

```
snapshot.forEach((userSnap) => {
```

```
  console.log('Current Token:', userSnap.child("deviceToken").val());
```

```
  var message = {
```

```
    notification: {
```

```
      title: "New report",
```

```
      body: name
```

```
    },
```

```
    data: {
```

```
      name: name
```

```
    },
```

```
    token:userSnap.child("deviceToken").val()
```

```
  };
```

```
admin.messaging().send(message)
    .then((response) => {
        // Response is a message ID string.
        console.log('Successfully sent message:', response);
        return null;
    })
    .catch((error) => {
        console.log('Error sending message:', error);
    });
});
});
});
});
```

## Κώδικας Image Adapter

```

package com.example.fastreport;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.squareup.picasso.Picasso;

import java.util.List;

public class ImageAdapter extends
RecyclerView.Adapter<ImageAdapter.ImageViewHolder>{

    private Context mContext;
    private List<Upload> mUploads;

    public ImageAdapter(Context context, List<Upload> uploads) {

        mContext = context;
        mUploads = uploads;
    }

    @Override
    public ImageViewHolder onCreateViewHolder( ViewGroup parent, int
viewType) {
        View v =
LayoutInflater.from(mContext).inflate(R.layout.image_item,parent,fals
e);
        return new ImageViewHolder(v);
    }

```

```

        @Override
        public void onBindViewHolder( ImageViewHolder holder, int
position) {
            Upload uploadCurrent = mUploads.get(position);
            holder.textViewName.setText(uploadCurrent.getName());

Picasso.get().load(uploadCurrent.getImageUrl()).fit().centerCrop().in
to(holder.imageView);

        }

        @Override

        public int getItemCount() {
            return mUploads.size();
        }

        public class ImageViewHolder extends RecyclerView.ViewHolder {

            public TextView textViewName;
            public ImageView imageView;

            public ImageViewHolder(@NonNull View itemView) {
                super(itemView);

                textViewName =
itemView.findViewById(R.id.text_view_name);
                imageView =
itemView.findViewById(R.id.image_view_upload);

                mContext = itemView.getContext();

                itemView.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View view) {

                        int itemPosition = getLayoutPosition();
                        Toast.makeText(mContext, "" + itemPosition,
Toast.LENGTH_SHORT).show();
                        Intent intent = new
Intent(mContext, FirstActivity.class);
                        intent.putExtra("name", ""+mUploads.get(itemPosition).getName());
                        intent.putExtra("img", ""+mUploads.get(itemPosition).getImageUrl());
                        mContext.startActivity(intent);

                    }
                });
            }
        }
    }
}

```