



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Ανάπτυξη εφαρμογής περιήγησης σε μονοπάτια «SANTORINITRAIL»
για Android**

Καραμολέγκου Μαργαρίτα

Εισηγητής: Δρ. Ιωάννης Έλληνας, Καθηγητής

Ανάπτυξη εφαρμογής περιήγησης σε μονοπάτια «SANTORINI TRAIL» για Android

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Ανάπτυξη εφαρμογής περιήγησης σε μονοπάτια «SANTORINI TRAIL» για
Android**

**Μαργαρίτα Καραμολέγκου
Α.Μ. 37233**

Εισηγητής:

Ιωάννης Έλληνας, Καθηγητής

Εξεταστική Επιτροπή:

Ημερομηνία εξέτασης

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/Η κάτωθι υπογεγραμμένος/η
του, με αριθμό μητρώου
φοιτητής/τρια του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών του
Πανεπιστημίου Δυτικής Αττικής πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας
μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του
συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και
πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή
μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί
προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του
άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος
φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του
έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η
Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου,
του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα
καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον
ενός ημερολογιακού δμήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά
εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού
Κανονισμού.»

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω θερμά:

Τον καθηγητή μου, κ. Ιωάννη Έλληνα, που μου έδωσε την ιδέα για το θέμα της πτυχιακής, ένα θέμα που κατάφερε να συνδυάσει τα προσωπικά μου ενδιαφέροντα με τα επαγγελματικά μου. Με παρακίνησε να μάθω πολλά για την ανάπτυξη εφαρμογών με το Android και να δημιουργήσω μια χρήσιμη κατά την γνώμη μου εφαρμογή, για την οποία είμαι περήφανη. Οι συμβουλές του, σε όλη τη διάρκεια της πτυχιακής μου εργασίας, ήταν πολύτιμες.

ΠΕΡΙΛΗΨΗ

Στην παρούσα πτυχιακή εργασία μελετάται η κατασκευή μιας εφαρμογής Android, που ονομάζεται SantoriniTrail και ενημερώνει τον χρήστη για τα μονοπάτια που υπάρχουν στην Σαντορίνη. Η εφαρμογή δημιουργήθηκε με την χρήση του λογισμικού Android Studio και προορίζεται για χρήση σε κινητά και tablet με λειτουργικό Android. Η επιλογή του θέματος έγινε λόγω της κυριαρχίας του λειτουργικού συστήματος του Android σε κινητές συσκευές σε παγκόσμια κλίμακα. Επίσης μέχρι στιγμής δεν κυκλοφορεί στην αγορά κάποια εφαρμογή που να ασχολείται με τα μονοπάτια της Σαντορίνης. Με την βοήθεια των χαρτών της Google διευκολύνονται οι περιπατητές στην πλοήγηση και μπορούν να πάρουν οδηγίες προς τα μονοπάτια, να βρουν το στίγμα τους στον χάρτη και να σημειώσουν σε ποια μονοπάτια πήγαν και τα σχόλιά τους για αυτά και με αυτόν τον τρόπο να καταγράφουν και τις αναμνήσεις τους. Αυτή η εφαρμογή αναπτύχθηκε για να προσφέρει στους χρήστες της έναν οδηγό στα μονοπάτια της Σαντορίνης, δίνοντας τους μια πλούσια εμπειρία, όπου όλες οι πληροφορίες βρίσκονται σε ένα μέρος και είναι εύκολη η πρόσβαση σε αυτές.

ABSTRACT

The following thesis explores the creation of SantoriniTrail, an Android application that informs it's users about the trails existing in Santorini. The app was made using Android Studio and is focused on being used on Android smart phones and tablets. The choice of subject was due to Android's dominance as an operating system in mobile devices worldwide. With the help of Google maps it is easy for users to navigate and get directions to the trails, save trails completed and to obtain their geo-location and tag it with their comments. This application was developed to provide users with a guide on Santorini paths where all the information is in one place and easily accessible.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Ανάπτυξη Εφαρμογών για κινητές συσκευές Android

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Android,εφαρμογή, Google maps , Σαντορίνη, οδηγός.

ΠΕΡΙΕΧΟΜΕΝΑ

1 ΕΙΣΑΓΩΓΗ.....	13
1.1 Σύλληψη της ιδέας.....	13
1.2 Περιγραφή του αντικειμένου της εφαρμογής SantoriniTrail	14
1.3 Τι είναι το Android;.....	15
1.4 Τι είναι ένα Mobile App;	17
1.5 Γιατί έγινε το SantoriniTrail σε Android app;	18
1.6 Βασικές αρχές εφαρμογών Android.....	20
1.7 Στοιχεία Εφαρμογής Android.....	21
1.8 Το αρχείο Manifest	24
1.9 Πόροι (Resources)	25
1.10 Βασικές Αρχές Δημιουργίας μιας Android εφαρμογής	26
1.11 Υποστήριξη διαφορετικών εκδόσεων του λειτουργικού συστήματος	27
1.12 Υποστήριξη διαφορετικών οθονών.....	29
1.13. Τοπικοποίηση εφαρμογής	34
1.14. Ροή για την καλύτερη εμπειρία χρήστη.....	35
1.15 Παγκόσμιο Σύστημα Στιγματοθέτησης και οι Υπηρεσίες Βάσει Τοποθεσίας	36
2 ΑΝΑΛΥΣΗ	39
2.1 Λειτουργικές και μη-λειτουργικές απαιτήσεις εφαρμογής.....	39
2.2 Περιπτώσεις Χρήσης.....	41
3 ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ	47
3.1 Εγκατάσταση Android Studio	47
3.2 Αρχική Οθόνη Εφαρμογής SantoriniTrail.....	48
3.3 Κεντρική Οθόνη Εφαρμογής SantoriniTrail.....	51
3.4 Μενού Εφαρμογής SantoriniTrail.....	52
3.5 Επιλογές του Μενού της Εφαρμογής SantoriniTrail.....	54
3.6 Οθόνη Επιλογής Μονοπατιού Εφαρμογής SantoriniTrail	56
3.7 Οθόνη περιγραφής μονοπατιού εφαρμογής SantoriniTrail	58
4 ΟΔΗΓΙΕΣ ΧΡΗΣΗΣ.....	61
4.1 Εγκατάσταση της εφαρμογής σε συσκευή Android	61
4.2 Χρησιμοποιώντας την εφαρμογή στο κινητό ή στην ταμπλέτα	64
5 ΔΟΚΙΜΗ ΛΟΓΙΣΜΙΚΟΥ	75
5.1 Διαδικασίες δοκιμής.....	75

5.2 Δοκιμή μονάδας.....	76
5.3 Δοκιμή Ευχρηστίας.....	77
5.4 Αποτελέσματα	78
6 ΣΥΜΠΕΡΑΣΜΑΤΑ.....	81
6.1 Μελλοντικές Βελτιώσεις	81
7 ΠΑΡΑΡΤΗΜΑ Α'	83
8 ΒΙΒΛΙΟΓΡΑΦΙΑ.....	109

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1.1 Μερίδιο αγοράς των φορητών λειτουργικών συστημάτων παγκοσμίως.....	20
Σχήμα 1.2: Διάγραμμα πίτα που απεικονίζει τα δεδομένα του πίνακα 1.1	29
Σχήμα 2.1: Διάγραμμα περιπτώσεων χρήσης.....	42

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1.1: Ποσοστά συσκευών ανά έκδοση Android. Δεδομένα που συλλέχθηκαν κατά την διάρκεια μιας περιόδου 7 ημερών που τελείωσε στις 7 Μαΐου του 2019. Οι εκδόσεις με διανομή κάτω από 0,1% δεν απεικονίζονται στον πίνακα.	28
Πίνακας 1.2: Πίνακας και διαγράμματα πίτας που απεικονίζουν την διανομή συσκευών με διάφορες διατάξεις, σύμφωνα με δεδομένα που συλλέχθηκαν σε μία περίοδο 7 ημερών που τελείωσε στις 7 Μαΐου του 2019. Οι εκδόσεις με διανομή κάτω από 0,1% δεν απεικονίζονται.	31
Πίνακας 2.1: Λειτουργικές απαιτήσεις εφαρμογής.....	40
Πίνακας 2.2: Μη Λειτουργικές απαιτήσεις εφαρμογής.....	41
Πίνακας 2.3: Περίπτωση χρήσης 1.....	43
Πίνακας 2.4: Περίπτωση χρήσης 2.....	43
Πίνακας 2.5: Περίπτωση χρήσης 3.....	43
Πίνακας 2.6: Περίπτωση χρήσης 4.....	44
Πίνακας 2.7: Περίπτωση χρήσης 5.....	45
Πίνακας 2.8: Περίπτωση χρήσης 6.....	45
Πίνακας 2.9: Περίπτωση χρήσης 7.....	46
Πίνακας 2.10: Περίπτωση χρήσης 8.....	46
Πίνακας 2.11: Περίπτωση χρήσης 9.....	46
Πίνακας 2.12: Περίπτωση χρήσης 10.....	46

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

API	Application Programming Interface
APK	Android Package
APP	Application
ART	Android Runtime
CPU	Central Processing Unit
DP/DPI	Density Independent Pixels
GPS	Global Positioning System
IDE	Development Environment
JVM	Java Virtual Machine
LBS	Location-based service
MDPI	Medium Density Independent Pixels
OHA	Open Handset Alliance
OS	Operating System
PX	Pixel
SD	Storage Device
SDK	Software Development Kit
SMS	Short Message Service
VM	Virtual Machine
USB	Universal Serial Bus
XML	Extensible Markup Language

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο αναλύεται το αντικείμενο της δημιουργίας μιας εφαρμογής, από την σύλληψη της ιδέας έως και τα διάφορα στάδια οργάνωσης και σχεδιασμού του τρόπου ανάπτυξης της εφαρμογής. Γίνεται μια σύντομη αναφορά στο Παγκόσμιο Σύστημα Στιγματοθέτησης (Global Positioning System - GPS). Αναλύεται το λειτουργικό σύστημα Android, με σκοπό να γίνει κατανοητό το γιατί επιλέχθηκε για την συγκεκριμένη εφαρμογή.

Για να δημιουργηθεί μια εφαρμογή πρέπει να εξεταστεί τόσο το πρόβλημα του σχεδιασμού της εμπειρίας του χρήστη, όσο και το πρόβλημα της κατανόησης της πραγματικής κατανομής των συστατικών που αποτελούν την εφαρμογή και τις αλληλεπιδράσεις τους.

Έτσι, σε αυτό το κεφάλαιο θα αναλυθούν κάποιες κατευθυντήριες γραμμές και επαναλαμβανόμενα μοτίβα που «διευκολύνουν» τον σχεδιασμό της εφαρμογής, για να βοηθήσει στην απόκτηση ποιοτικών λύσεων. Θα μελετηθεί η ιδιαιτερότητα της ανάπτυξης εφαρμογών για το λειτουργικό σύστημα Android, ανάλογα με τις διατάξεις, την γλώσσα και την οικογένεια των συσκευών που το χρησιμοποιούν.

1.1 Σύλληψη της ιδέας

Το ποίημα του Κωνσταντίνου Καβάφη «Ιθάκη» ξεκινά με το γνωστό σε όλους: Σα βγεις στον πηγαιμό για την Ιθάκη, να εύχεται να νάναι μακρύς ο δρόμος, γεμάτος περιπέτειες, γεμάτος γνώσεις. Το ταξίδι είναι η αγαπημένη απασχόληση των περισσότερων ανθρώπων και σαν τον Καβάφη εύχονται να διαρκέσει πολύ. Η χαλάρωση δεν είναι το μόνο που αναζητούν οι ταξιδιώτες. Αρκετοί ταξιδιώτες είναι και εξερευνητές. Αυτοί θέλουν το ταξίδι τους να είναι γεμάτο με περιπέτειες, νέες εμπειρίες και γνώσεις. Φυσικά, θέλουν και να αποτυπώνονται οι αναμνήσεις τους. Κάποτε, το μόνο “αξεσουάρ” που είχε ένας ταξιδιώτης ήταν η πένα του – αργότερα είχε την φωτογραφική του μηχανή. Σήμερα τα πράγματα έχουν αλλάξει – δραστικά. Με την ανάπτυξη της τεχνολογίας των “έξυπνων” τηλεφώνων μπορεί ο ταξιδιώτης να έχει μια διαδραστική εμπειρία, όπου μαθαίνει, μοιράζεται και καταγράφει τις εμπειρίες του.

Από την Ιθάκη, μεταφερόμαστε στην Σαντορίνη, που είναι ένας προορισμός πρώτος στις προτιμήσεις ανθρώπων από όλα τα μήκη της γης. Μέχρι την στιγμή ανάθεσης του θέματος της πτυχιακής εργασίας δεν κυκλοφορούσε στην αγορά κάποια εφαρμογή που να ασχολείται με τα μονοπάτια της Σαντορίνης. Από αυτήν την «έλλειψη» γεννήθηκε η ιδέα της παρούσας πτυχιακής όπου μελετήθηκε η κατασκευή μιας εφαρμογής Android, που ονομάζεται SantoriniTrail και ενημερώνει τον χρήστη για τα μονοπάτια που υπάρχουν στην Σαντορίνη. Η εφαρμογή δημιουργήθηκε με την χρήση του λογισμικού Android Studio και προορίζεται για χρήση σε κινητά και tablet με λειτουργικό Android. Η επιλογή του θέματος έγινε λόγω της κυριαρχίας του λειτουργικού συστήματος του Android σε κινητές συσκευές σε παγκόσμια κλίμακα.

Η ομορφιά της Σαντορίνης, είναι αυτό που την κάνει μοναδική και με την εφαρμογή SantoriniTrail μπορούν οι ταξιδιώτες να απολαύσουν αυτή την ομορφιά, να ζήσουν μια περιπέτεια και να αποκτήσουν γνώσεις καθώς περπατούν στα μονοπάτια της. Με την βοήθεια των χαρτών της Google διευκολύνονται στην πλοήγηση και μπορούν να πάρουν οδηγίες προς τα μονοπάτια, να βρουν το στίγμα τους στον χάρτη και να σημειώσουν σε ποια μονοπάτια πήγαν και τα σχόλια τους για αυτά και με αυτόν τον τρόπο να καταγράψουν και τις αναμνήσεις τους.

Αυτή η εφαρμογή αναπτύχθηκε για να προσφέρει στους χρήστες της έναν οδηγό στα μονοπάτια της Σαντορίνης, δίνοντάς τους μια πλούσια εμπειρία, όπου όλες οι πληροφορίες βρίσκονται σε ένα μέρος και είναι εύκολα προσβάσιμες.

1.2 Περιγραφή του αντικειμένου της εφαρμογής SantoriniTrail

Αν και η πεζοπορία παρέχει μια ευκαιρία για να έρθει κάποιος «σε αρμονία με τη φύση», είναι επίσης μια δραστηριότητα που μπορεί να ωφεληθεί από την κινητή τεχνολογία.

Μια συσκευή Android μπορεί να βελτιώσει την οργάνωση μιας πεζοπορίας, παρουσιάζοντας νέα μονοπάτια προς εξερεύνηση, νέες προκλήσεις, να παρέχει πληροφορίες που μπορούν να σώσουν κάποιον σε περίπτωση που χαθεί η τραυματιστεί.

Το SantoriniTrail είναι ένας οδηγός πεζοπορίας και εξερεύνησης του τοπίου, της φύσης, της αρχιτεκτονικής και της ιστορίας της Σαντορίνης. Η εφαρμογή περιλαμβάνει περιγραφές για πεζοπορικές διαδρομές στο νησί. Κάθε μία από τις διαδρομές έχει

υψομετρικό προφίλ, βαθμό δυσκολίας, εκτιμώμενο χρόνο περαίωσης και προτείνει σημεία ενδιαφέροντος. Η εφαρμογή εντοπίζει το στίγμα του χρήστη και εκείνος μπορεί να επιλέξει την πλησιέστερη διαδρομή.

Τα βήματα που ακολουθήθηκαν είναι:

- Εξαγωγή προδιαγραφών λογισμικού, όπου ορίζονται οι λειτουργικές δυνατότητες του λογισμικού και οι περιορισμοί που διέπουν την λειτουργία του.
- Σχεδιασμός και υλοποίηση του λογισμικού. Παράγεται το λογισμικό που ορίστηκε στις προδιαγραφές.
- Επικύρωση του λογισμικού. Το λογισμικό πρέπει να επικυρωθεί προκειμένου να εξασφαλιστεί ότι θα κάνει αυτά που ζητούνται.
- Εξέλιξη και συντήρηση του λογισμικού ώστε να ανταποκρίνεται στις μεταβαλλόμενες απαιτήσεις.

1.3 Τι είναι το Android;

Πολλοί όταν αναφέρονται στο Android έχουν την λανθασμένη εντύπωση ότι το Android είναι μια συσκευή τηλεφώνου ή μια εφαρμογή, ενώ στην πραγματικότητα το Android είναι ένα λειτουργικό σύστημα. Σχεδιασμένο με βάση τον πυρήνα του Linux, το Android κυκλοφόρησε ως ένα λειτουργικό σύστημα ανοιχτού κώδικα. Στον πιο απλοϊκό ορισμό του, το Linux είναι ένα λειτουργικό σύστημα που συναντάται συχνότερα σε διακομιστές και επιτραπέζιους υπολογιστές. Το Android δεν είναι απλώς μια έκδοση του Linux, λόγω των πολλών αλλαγών που έχει υποστεί, αλλά σχετίζεται. Έτσι το Android είναι ένα λειτουργικό σύστημα σχεδιασμένο με γνώμονα την φορητότητα, όπου βρίσκονται οι λειτουργίες και οι εφαρμογές των κινητών τηλεφώνων. Όλα όσα βλέπει ο χρήστης στην οθόνη της συσκευής του αποτελούν μέρος του λειτουργικού συστήματος. Όταν λαμβάνει μια κλήση, ένα μήνυμα κειμένου ή ένα μήνυμα ηλεκτρονικού ταχυδρομείου, το λειτουργικό σύστημα επεξεργάζεται αυτές τις πληροφορίες και τις μετατρέπει σε ευανάγνωστη μορφή. Ως πλατφόρμα ανοιχτού κώδικα επιτρέπει σε προγραμματιστές από όλο το κόσμο να αναπτύξουν εφαρμογές για συσκευές Android [1,2].

Το Android αναπτύχθηκε για πρώτη φορά από μια εταιρεία της Silicon Valley με την επωνυμία Android Inc. Αυτό όμως που έδωσε στο Android ένα πλεονέκτημα στην παροχή ενός ολοκληρωμένου λογισμικού, ήταν η συνεργασία με την Google το 2007

Μαργαρίτα Καραμολέγκου

μέσω του Open Handset Alliance (OHA), μιας κοινοπραξίας 84 εταιρειών για την ανάπτυξη ανοικτών προτύπων για κινητές συσκευές. Πέραν της Google κάποιες από τις επιχειρήσεις που είναι μέλη της κοινοπραξίας, ως κατασκευαστές, πάροχοι ή προγραμματιστές του Android είναι οι εταιρείες HTC, Sony, Dell, Intel, Motorola, Qualcomm, Texas Instruments, Google, Samsung Electronics, LG Electronics, T-Mobile, Sprint Corporation.

Η Java ήταν η πρώτη επίσημη και η πιο υποστηριζόμενη από την Google γλώσσα προγραμματισμού για το Android, που σημαίνει ότι με αυτήν έχουν κατασκευαστεί οι περισσότερες εφαρμογές στο Play Store. Το 2017 η Google πρόσθεσε και την γλώσσα Kotlin, μια γλώσσα προγραμματισμού που σχεδιάστηκε για να συνεργάζεται πλήρως με την Java [3]. Αν και το Android βασίζεται στην Java, δεν υπάρχει εικονική μηχανή Java (Java Virtual Machine - JVM) στην πλατφόρμα. Αντί να επιτρέπει στα προγράμματα Java να τρέχουν μέσω του JVM, η Google ανέπτυξε ειδικά για το Android, μια εικονική μηχανή το Dalvik. Το Dalvik εκτελεί μεταγλωττισμένο κώδικα Java και το διαβάζει ως Dalvik bytecode και σχεδιάστηκε για να βελτιστοποιεί την ισχύ της μπαταρίας και να διατηρεί την λειτουργικότητα σε περιβάλλοντα με περιορισμένη μνήμη και ισχύ CPU, όπως αυτή των κινητών τηλεφώνων και των tablet. Το 2014 στην έκδοση Lollipop του Android το Dalvik αντικαταστάθηκε από το Android Runtime (ART) προσφέροντας ακόμη γρηγορότερη εκτέλεση των εφαρμογών, μεγαλύτερη αυτονομία της μπαταρίας και άλλες βελτιώσεις [4].

Το λειτουργικό σύστημα Android χωρίζεται σε διάφορους αριθμούς έκδοσης, οι οποίοι έχουν κωδικό όνομα, βασισμένο σε ονόματα γλυκών που ακολουθούν αλφαβητική σειρά όπως Android Lollipop, Marshmallow, Nougat, Oreo ή Pie. Η κάθε νέα έκδοση του Android υποδηλώνει σημαντικά άλματα στα χαρακτηριστικά, τη λειτουργία και τη σταθερότητα.

Τα περισσότερα μοντέρνα smartphones και tablet μεσαίου κόστους διαθέτουν Android 8.x Oreo ή Android 9 Pie, ενώ σε οικονομικότερα μοντέλα, είναι ακόμα δυνατό να βρεθεί το Android 5.x ή παλαιότερο, αλλά αυτές οι συσκευές είναι όλο και λιγότερο συνηθισμένες.

Οι περισσότεροι κατασκευαστές συσκευών Android, όπως η Samsung, HTC, Motorola, Sony, κάνουν προσαρμογές στο λειτουργικό σύστημα που τροποποιούν την διεπαφή χρήστη. Προσθέτουν επιπλέον λειτουργίες στο κινητό τηλέφωνο, διαφορετικά

εικονίδια και άλλες μικροαλλαγές που έχουν σχεδιαστεί για να παρέχουν μια εμπειρία μοναδική σε σχέση με τον συγκεκριμένο κατασκευαστή τηλεφώνων.

Ο βασικός σχεδιασμός του Android εστιάζει κυρίως στην χρήση οθονών αφής φορητών συσκευών (όπως smartphones και tablets). Όμως η χρήση του δεν σταματάει εκεί. Το Android σήμερα καλύπτει μια τεράστια γκάμα επιλογών και διαφορετικών συσκευών σε πολλούς τομείς της τεχνολογίας. Ήδη, χρησιμοποιείται και σε τηλεοράσεις, με ένα εξειδικευμένο σχεδιασμό για το σκοπό αυτό (Android TV), χρησιμοποιείται σε ρολόγια χειρός (Android Wear), σε αυτοκίνητα (Android Auto) και σε άλλες συσκευές.

1.4 Τι είναι ένα Mobile App;

Μια εφαρμογή για κινητά (mobile application), το γνωστό σε όλους app, είναι ένα είδος λογισμικού που έχει σχεδιαστεί για να λειτουργεί σε μια φορητή συσκευή. Οι εφαρμογές είναι μικρές, μεμονωμένες μονάδες λογισμικού με περιορισμένη λειτουργία. Με την άνοδο των εφαρμογών κινητής τηλεφωνίας, οι άνθρωποι σήμερα είναι σε αναζήτηση πληροφοριών εν κινήσει. Αυτός ο τομέας της βελτίωσης της τεχνολογίας των κινητών τηλεφώνων επιτρέπει στους προγραμματιστές να προσφέρουν στους χρήστες ακριβώς αυτό που αναζητούν στην προτιμώμενη περιοχή ενδιαφέροντός τους [5,6].

Αρχικά οι εφαρμογές προοριζόνταν για την μεταφορά υπηρεσιών παρόμοιων με αυτών ενός υπολογιστή σε κινητές συσκευές, όπως το ηλεκτρονικό ταχυδρομείο, το ημερολόγιο και βάση δεδομένων επαφών. Ωστόσο, σύντομα υπήρξε ταχεία επέκταση σε άλλους τομείς, όπως κινητά παιχνίδια, αυτοματισμοί εργοστασίων, GPS και υπηρεσίες εντοπισμού θέσης με αποτέλεσμα σήμερα να υπάρχουν διαθέσιμες εκατομμύρια εφαρμογές. Λόγω των περιορισμένων πόρων του υλικού των πρώτων κινητών, η ταυτόχρονη εκτέλεση πολλών λειτουργιών στις εφαρμογές ήταν συνήθως κάτι προς αποφυγή. Αντ' αυτού, κάθε εφαρμογή παρείχε περιορισμένες και μεμονωμένες λειτουργίες, όπως παιχνίδι, αριθμομηχανή ή περιήγηση στο διαδίκτυο.

Καθώς οι εφαρμογές για κινητά γίνονται πιο ισχυρές, σήμερα ακολουθούν μια κατεύθυνση διαφορετική από αυτή στα ενσωματωμένα συστήματα λογισμικού που βρίσκονται σε υπολογιστές. Υπάρχει μια πιο εξελιγμένη προσέγγιση που περιλαμβάνει την ανάπτυξη ειδικά για το κινητό περιβάλλον, εκμεταλλευόμενοι τόσο τους περιορισμούς όσο και τα πλεονεκτήματα του. Για παράδειγμα, οι εφαρμογές που χρησιμοποιούν λειτουργίες βάσει τοποθεσίας σχεδιάζονται εξ αρχής λαμβάνοντας

υπόψη ότι ο χρήστης δεν έχει την ίδια αντίληψη της τοποθεσίας με τον χρήστη του υπολογιστή.

Έτσι η εξειδίκευση στις λειτουργίες των εφαρμογών κάνουν τις σημερινές συσκευές πιο δημοφιλείς, επειδή επιτρέπουν στους καταναλωτές να επιλέγουν ποιες εφαρμογές να κατεβάσουν, εξατομικεύοντας έτσι τις συσκευές τους.

1.5 Γιατί έγινε το SantoriniTrail σε Android app;

Μετά την εποχή των επιτραπέζιων και των φορητών υπολογιστών, προέκυψε νέο ενδιαφέρον για κινητές συσκευές και tablet. Το 2008 η σημαντικότερη επικεφαλίδα στα μέσα μαζικής ενημέρωσης σε ότι αφορά την τεχνολογία πληροφορικής ήταν «Η κινητή θα ξεπεράσει τη σταθερή πρόσβαση στο Internet μέχρι το 2014» μια τολμηρή πρόβλεψη από την Mary Meeker, αναλύτρια της Kleiner Perkins Caufield Byers, η οποία αναθεωρεί τις τάσεις της τεχνολογίας ετησίως τον Μάιο. Σήμερα, σύμφωνα με την τελευταία έκθεση Digital Future in Focus από την comScore, έχει ξεπεράσει εδώ και πολύ καιρό αυτό το σημείο ανατροπής, ενώ σε ορισμένες χώρες, όπως την Ινδία, το Μεξικό και την Ινδονησία έχουν 4 φορές υψηλότερο επίπεδο χρηστών που προτιμούν την πρόσβαση στο διαδίκτυο μέσω κινητών έναντι των επιτραπέζιων υπολογιστών [7].

Σύμφωνα με τις πληροφορίες της στατιστικής πύλης comScore για το 2018, το ποσό των χρηστών κινητής τηλεφωνίας είναι περίπου στο ίδιο επίπεδο με τους χρήστες υπολογιστών γραφείου, ενώ ο χρόνος πρόσβασης στο διαδίκτυο από κινητές συσκευές έχει ξεπεράσει τον χρόνο σε υπολογιστές. Επίσης η έρευνα έδειξε ότι πάνω από το 80% του χρόνου που περνά ο χρήστης στην συσκευή είναι σε εφαρμογές (app) αντί για προγράμματα περιήγησης.

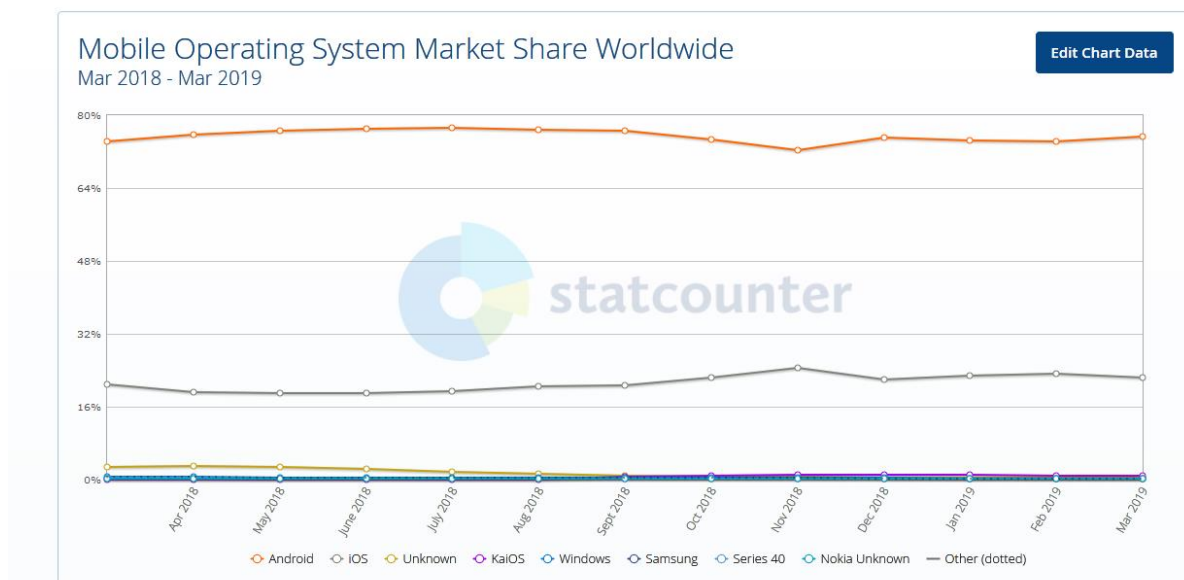
Οι εταιρείες συνειδητοποίησαν αυτή τη νέα τάση και άρχισαν να επικεντρώνονται περισσότερο στη βιομηχανία κινητής τηλεφωνίας. Σήμερα, είναι μια ταχέως αναπτυσσόμενη βιομηχανία που προσφέρει πολλές ευκαιρίες. Υπάρχουν πολλές συσκευές στην αγορά με διαφορετικό λογισμικό και υλικό, καθιστώντας την εργασία των μηχανικών λογισμικού πιο περίπλοκη, επειδή τα προϊόντα τους πρέπει να ικανοποιούν ένα τεράστιο αριθμό πελατών. Πρέπει να αναζητήσουν την καλύτερη λύση για το πώς να ικανοποιήσουν την αγορά με τον πιο βολικό τρόπο.

Σήμερα, το Android είναι το πιο διαδεδομένο λειτουργικό στον κόσμο. Αρκεί να αναφέρουμε ότι οι συσκευές Android έχουν περισσότερες πωλήσεις από όσες έχουν οι συσκευές Windows, Windows Phone, iOS και Mac OS X μαζί.

Σε ότι αφορά τα «αντίπαλα στρατόπεδα» του Android, κυριότερος ανταγωνιστής είναι το iPhone της Apple. Στην ερώτηση αν είναι το Android καλύτερο από τα iPhone, δεν υπάρχει σωστή απάντηση γιατί εξαρτάται από το τι προτιμά ο χρήστης. Τα iPhones είναι αρκετά εύχρηστα και έχουν αίσθηση ποιοτικού προϊόντος, ενώ το Android είναι πιο προσαρμόσιμο και υπάρχουν κυριολεκτικά εκατοντάδες διαφορετικά τηλέφωνα που δεν είναι πάντα ποιοτικά. Αυτό συμβαίνει επειδή οποιοδήποτε τηλέφωνο σε οποιαδήποτε τιμή μπορεί να τρέξει Android, αν και τα πιο ακριβά μοντέλα θα το τρέχουν πιο γρήγορα από τα οικονομικότερα.

Η Google επιμένει στην ανοιχτή διάθεση του Android και αυτό έχει σαν αποτέλεσμα ότι παρά τη δημοτικότητα των iPhones στη Δύση, το Android είναι στην πραγματικότητα το λειτουργικό σύστημα για τα έξυπνα κινητά με τις μεγαλύτερες πωλήσεις παγκοσμίως από το 2011.

Κατά την εξέταση των παγκόσμιων στατιστικών για τις πωλήσεις, το Android έχει ξεκάθαρο προβάδισμα έναντι του iOS όσον αφορά το μερίδιο αγοράς. Η στατιστική πύλη Statcounter αναφέρει ότι κατά την περίοδο Μαρτίου 2018 έως Μαρτίου 2019 το Android διατηρεί ένα τεράστιο ποσοστό 75,39% του τομέα, και το iOS να ακολουθεί με το 22,35% [8].



Σχήμα 1.1 Μεριδίο αγοράς των φορητών λειτουργικών συστημάτων παγκοσμίως

1.6 Βασικές αρχές εφαρμογών Android

Τα εργαλεία της εργαλειοθήκης ανάπτυξης λογισμικού (Android SDK) μεταγλωττίζουν τον κώδικα της εφαρμογής μαζί με οποιαδήποτε δεδομένα και αρχεία πόρων σε ένα πακέτο Android (APK), το οποίο είναι αρχείο με κατάληξη .apk. Ένα αρχείο APK περιέχει όλο το περιεχόμενο μιας εφαρμογής Android και είναι το αρχείο που χρησιμοποιούν οι συσκευές με Android για την εγκατάσταση της εφαρμογής.

Κάθε εφαρμογή Android ζει στο δικό του απομονωμένο περιβάλλον (sandbox). Αυτός ο ασφαλής εικονικός χώρος προστατεύεται από τα εξής χαρακτηριστικά ασφάλειας Android.

- Το λειτουργικό σύστημα είναι "πολλαπλών χρηστών" (multi-user) όπου κάθε εφαρμογή Android θεωρείται ένας διαφορετικός χρήστης.
- Από προεπιλογή, το σύστημα εκχωρεί σε κάθε εφαρμογή ένα μοναδικό αναγνωριστικό χρήστη Linux (user ID) που χρησιμοποιείται μόνο από το

σύστημα και είναι άγνωστο στην εφαρμογή. Το σύστημα ορίζει δικαιώματα για όλα τα αρχεία μιας εφαρμογής, έτσι ώστε μόνο το user ID που αντιστοιχεί σε αυτήν την εφαρμογή να έχει πρόσβαση σε αυτά.

- Κάθε διεργασία έχει τη δική της εικονική μηχανή (VM), οπότε ο κώδικας μιας εφαρμογής εκτελείται μεμονωμένα από άλλες εφαρμογές.
- Από προεπιλογή, κάθε εφαρμογή τρέχει στη δική της διεργασία Linux. Το σύστημα Android ξεκινά τη διεργασία όταν χρειάζεται να εκτελεστεί κάποιο από τα στοιχεία της εφαρμογής και στη συνέχεια τερματίζει τη διεργασία όταν δεν είναι πλέον απαραίτητο ή όταν το σύστημα πρέπει να ανακτήσει τη μνήμη για άλλες εφαρμογές.

Στο σύστημα Android κάθε εφαρμογή, από προεπιλογή, έχει πρόσβαση μόνο στα στοιχεία που απαιτεί για να κάνει τη δουλειά της και τίποτα περισσότερο. Αυτό δημιουργεί ένα πολύ ασφαλές περιβάλλον στο οποίο μια εφαρμογή δεν μπορεί να έχει πρόσβαση σε τμήματα του συστήματος για τα οποία δεν του έχει δοθεί άδεια. Ωστόσο, υπάρχουν τρόποι για μια εφαρμογή να μοιράζεται δεδομένα με άλλες εφαρμογές και να έχει πρόσβαση σε υπηρεσίες συστήματος:

- Είναι δυνατό δύο εφαρμογές να μοιράζονται το ίδιο αναγνωριστικό χρήστη Linux, οπότε μπορεί η μία εφαρμογή να έχει πρόσβαση στα αρχεία της άλλης. Για τη διατήρηση των πόρων του συστήματος, οι εφαρμογές με το ίδιο αναγνωριστικό χρήστη μπορούν επίσης να εκτελούνται στην ίδια διαδικασία Linux και να μοιράζονται την ίδια εικονική μηχανή (VM). Οι εφαρμογές θα πρέπει να έχουν υπογραφεί με το ίδιο πιστοποιητικό.
- Μια εφαρμογή μπορεί να ζητήσει άδεια πρόσβασης στα δεδομένα της συσκευής, όπως για παράδειγμα στις επαφές του χρήστη, τα μηνύματα SMS, το αποθηκευτικό χώρο (κάρτα SD), την κάμερα και στο Bluetooth. Ο χρήστης πρέπει να παραχωρήσει ρητά αυτά τα δικαιώματα [9].

1.7 Στοιχεία Εφαρμογής Android

Ένα ιδιαίτερο χαρακτηριστικό των εφαρμογών Android είναι ότι παρέχουν πολλαπλά σημεία εισόδου. Οι εφαρμογές Android κατασκευάζονται ως ένας συνδυασμός στοιχείων που μπορούν να χρησιμοποιηθούν ξεχωριστά. Για παράδειγμα, μια δραστηριότητα (activity) είναι ένας τύπος συστατικού της εφαρμογής που παρέχει μια

διεπαφή χρήστη. Η "κύρια" δραστηριότητα ξεκινά όταν ο χρήστης κάνει κλικ στο εικονίδιο της εφαρμογής, που βρίσκεται στο μενού της συσκευής Android όπου έχει εγκατασταθεί η εφαρμογή. Ωστόσο ακόμα και μια ειδοποίηση στο συρτάρι ειδοποιήσεων (notification drawer) της συσκευής ή και μια διαφορετική εφαρμογή μπορεί να εκκινήσει την εφαρμογή ή να μεταφέρει τον χρήστη απευθείας σε μια διαφορετική δραστηριότητα της εφαρμογής. Άλλα στοιχεία, όπως οι δέκτες εκπομπής (broadcast receivers) επιτρέπουν στην εφαρμογή Android να εκτελεί εργασίες φόντου χωρίς διεπαφή χρήστη.

Το καθένα από τα βασικά δομικά στοιχεία μιας εφαρμογής Android, εξυπηρετεί συγκεκριμένο σκοπό και έχει συγκεκριμένο κύκλο ζωής που ορίζει πότε το στοιχείο θα δημιουργηθεί και πότε θα καταστραφεί. Οι 4 βασικοί τύποι στοιχείων της εφαρμογής είναι:

1) Δραστηριότητα (Activity):

Η διεπαφή χρήστη που χειρίζεται την αλληλεπίδραση του χρήστη με την οθόνη της συσκευής του. Οι δραστηριότητες μπορούν να έχουν πλήρη οθόνη ή πλωτά παράθυρα. Μία από τις δραστηριότητες της εφαρμογής θα είναι η κύρια δραστηριότητα. Αυτή συνήθως είναι η δραστηριότητα (οθόνη) που παρουσιάζεται στο χρήστη όταν η εφαρμογή ξεκινά για πρώτη φορά. Μια εφαρμογή συνήθως έχει πολλές δραστηριότητες, κάθε οθόνη μιας εφαρμογής είναι μια διαφορετική δραστηριότητα. Παρόλο που οι δραστηριότητες συνεργάζονται για να σχηματίσουν μια ολοκληρωτική εμπειρία χρήστη στην εφαρμογή, κάθε μία είναι ανεξάρτητη από τις άλλες. Έτσι μπορεί και μια ξεχωριστή εφαρμογή στην συσκευή να ξεκινήσει οποιαδήποτε από αυτές τις δραστηριότητες, αν το επιτρέπει η εφαρμογή. Όλες οι δραστηριότητες μιας εφαρμογής είναι χαλαρά διασυνδεδεμένες: Μια δραστηριότητα σχεδιάζεται γύρω από μια συγκεκριμένη ενέργεια του χρήστη ενώ αλληλεπιδρά με μια άλλη δραστηριότητα. Επιπλέον, από μια δραστηριότητα μπορούν να ενεργοποιηθούν πολλές διαφορετικές δραστηριότητες ανάλογα με τη δράση του χρήστη. Μια δραστηριότητα διευκολύνει τις ακόλουθες αλληλεπιδράσεις μεταξύ συστήματος και εφαρμογής:

- Παρακολουθεί τι ακριβώς ενδιαφέρει τον χρήστη (τι είναι στην οθόνη) για να διασφαλιστεί ότι το σύστημα συνεχίζει να τρέχει την διεργασία που φιλοξενεί τη δραστηριότητα.
- Διατήρηση των διεργασιών που έχουν προηγηθεί γιατί μπορεί ο χρήστης να θελήσει να επιστρέψει σε αυτές τις σταματημένες δραστηριότητες.

- Βοηθά στον τερματισμό (kill) της τρέχουσας διεργασίας ώστε να μπορέσει ο χρήστης να επιστρέψει σε δραστηριότητες με αποκαταστημένη την προηγούμενη κατάστασή τους.
- Παροχή ενός τρόπου υλοποίησης της ροής μεταξύ εφαρμογών και συντονισμού της ροής από το σύστημα, όπως συμβαίνει για παράδειγμα στην κοινοποίηση (share).

2) Υπηρεσίες (Services)

Μια υπηρεσία είναι ένα στοιχείο που τρέχει στο παρασκήνιο για να εκτελεί λειτουργίες χωρίς να παρέχει διεπαφή χρήστη και μπορεί να λειτουργεί ακόμα και αν ο χρήστης τερματίσει την εφαρμογή. Είναι χρήσιμες όταν οι λειτουργίες διαρκούν πολύ ή πρέπει να πραγματοποιούνται τακτικά (όπως ο έλεγχος ενός διακομιστή για νέα αλληλογραφία).

3) Δέκτες εκπομπής (Broadcast receivers)

Διαχειρίζονται την επικοινωνία μεταξύ του Android OS και των εφαρμογών. Σε γενικές γραμμές, οι εκπομπές μπορούν να χρησιμοποιηθούν ως σύστημα ανταλλαγής μηνυμάτων μεταξύ των εφαρμογών. Το σύστημα στέλνει αυτόματα εκπομπές για διάφορα γεγονότα συστήματος, όπως για παράδειγμα όταν το σύστημα μπαίνει ή βγαίνει από την λειτουργία “Χρήση σε πτήση”. Οι εκπομπές του συστήματος αποστέλλονται σε όλες τις εφαρμογές που έχουν εγγραφεί για να λάβουν το συμβάν. Παρόλο που δεν έχουν διεπαφή χρήστη, μπορούν να ενημερώνουν τον χρήστη για το συμβάν στην μπάρα ειδοποιήσεων της συσκευής.

4) Προμηθευτές περιεχομένου (Content providers)

Ένα στοιχείο προμηθευτή περιεχομένου παρέχει δεδομένα από μία εφαρμογή σε άλλες κατόπιν αιτήματος. Ένας προμηθευτής περιεχομένου μπορεί να χρησιμοποιήσει διαφορετικούς τρόπους για να αποθηκεύσει τα δεδομένα και τα δεδομένα μπορούν να αποθηκευτούν σε μια βάση δεδομένων, σε αρχεία ή ακόμα και σε ένα δίκτυο.

Οι προμηθευτές περιεχομένου επιτρέπουν την συγκέντρωση του περιεχομένου σε ένα μέρος και την πρόσβαση από πολλές διαφορετικές εφαρμογές ανάλογα με τις εκάστοτε ανάγκες. Ένας προμηθευτής περιεχομένου συμπεριφέρεται πάρα πολύ σαν μια βάση δεδομένων όπου μπορεί μια άλλη εφαρμογή να κάνει ερώτημα, να επεξεργαστεί το περιεχόμενό του, καθώς και να προσθέσει ή να διαγράψει περιεχόμενο χρησιμοποιώντας τις μεθόδους insert(), update(), delete() και query(). Οι προμηθευτές περιεχομένου, χρησιμεύουν και στην ανάγνωση και την εισαγωγή δεδομένων σε μια

εφαρμογή που δεν διαμοιράζεται τα δεδομένα της, όπως στην περίπτωση αποθήκευσης σημειώσεων στο NotePad. Στις περισσότερες περιπτώσεις τα δεδομένα αποθηκεύονται σε μια βάση δεδομένων SQLite, στο διαδίκτυο, ή κάποιο άλλο μέσο αποθήκευσης στο οποίο έχει πρόσβαση η εφαρμογή.

1.8 Το αρχείο Manifest

Κάθε έργο στο Android περιλαμβάνει ένα αρχείο δηλώσεων, το AndroidManifest.xml, αποθηκευμένο στον ριζικό κατάλογο του έργου της εφαρμογής. Το αρχείο δήλωσης είναι ένα σημαντικό μέρος της εφαρμογής, επειδή ορίζει τη δομή και τα μεταδεδομένα της εφαρμογής, τα στοιχεία της και τις απαιτήσεις της. Το αρχείο δήλωσης καθορίζει επίσης τα μεταδεδομένα της εφαρμογής, τα οποία περιλαμβάνουν το εικονίδιο, τον αριθμό έκδοσης, κλπ. Δηλώνει το ελάχιστο επίπεδο API που απαιτεί η εφαρμογή. Καθορίζει τις απαιτούμενες άδειες, όπως για την πρόσβαση στις επαφές του χρήστη ή στο διαδίκτυο. Δηλώνει τις απαιτήσεις υλικού, οθόνης ή πλατφόρμας, όπως φωτογραφικής μηχανής, επιταχυνσιόμετρου, οθόνη επαφής, τις API βιβλιοθήκες με τις οποίες πρέπει να συνδεθεί η εφαρμογή, όπως την βιβλιοθήκη χαρτών της Google. Μπορεί επίσης να καθοριστεί αν η εφαρμογή πρέπει να εγκατασταθεί σε μια κάρτα SD της εσωτερικής μνήμης. Το σύστημα Android πρέπει να έχει αυτές τις πληροφορίες ώστε να εκτελέσει σωστά τον κώδικα της εφαρμογής.

Όταν τα βασικά στοιχεία της εφαρμογής όπως δραστηριότητες, υπηρεσίες, δέκτες εκπομπής ή προμηθευτές περιεχομένου δηλώνονται στο αρχείο manifest μπορούν να χρησιμοποιήσουν φίλτρα προθέσεων (Intent Filters) και άδειες (Permissions), για να καθορίζουν τον τρόπο με τον οποίο συντονίζονται μεταξύ τους και με άλλες εφαρμογές. Οι δραστηριότητες, οι υπηρεσίες και οι προμηθευτές περιεχομένου που συμπεριλαμβάνονται μεν στον κώδικα της εφαρμογής, αλλά δεν δηλώνονται στο αρχείο manifest δεν είναι ορατοί στο σύστημα και συνεπώς δεν μπορούν ποτέ να εκτελεστούν. Ωστόσο, οι δέκτες εκπομπής μπορούν είτε να δημιουργηθούν δυναμικά στον κώδικα ως αντικείμενα BroadcastReceiver είτε να δηλωθούν στο manifest.

Το σύστημα εντοπίζει τα στοιχεία που μπορούν να ανταποκριθούν σε μια πρόθεση συγκρίνοντας την πρόθεση που λαμβάνεται με τα φίλτρα προθέσεων που παρέχονται στο αρχείο manifest άλλων εφαρμογών στη συσκευή.

Όταν δηλώνεται μια δραστηριότητα στο αρχείο manifest της εφαρμογής, μπορούν προαιρετικά να συμπεριληφθούν φίλτρα προθέσεων που δηλώνουν τις δυνατότητες της

Μαργαρίτα Καραμολέγκου

δραστηριότητας, ώστε να μπορούν να ανταποκρίνονται στις προθέσεις άλλων εφαρμογών.

Οι συσκευές Android δεν διαθέτουν όλες τις ίδιες δυνατότητες και χαρακτηριστικά. Για την αποτροπή της εγκατάστασης μιας εφαρμογής σε συσκευές που δεν διαθέτουν τα χαρακτηριστικά που απαιτούνται από την εφαρμογή, είναι σημαντικό να καθοριστεί σαφώς ένα προφίλ για τους τύπους συσκευών που υποστηρίζει η εφαρμογή, δηλώνοντας τις απαιτήσεις συσκευής και λογισμικού στο αρχείο manifest. Οι περισσότερες από αυτές τις δηλώσεις είναι μόνο πληροφοριακές και το σύστημα δεν τις διαβάζει, αλλά τις διαβάζουν εξωτερικές υπηρεσίες, όπως το Google Play, για να παρέχουν φιλτράρισμα στους χρήστες όταν αναζητούν εφαρμογές από τη συσκευή τους.

1.9 Πόροι (Resources)

Μια καλογραμμένη εφαρμογή Android είναι κάτι παραπάνω από απλός κώδικας. Απαιτεί πόρους, όπως εικόνες, αρχεία ήχου και οτιδήποτε σχετίζεται με την οπτική παρουσίαση της εφαρμογής, ξεχωριστούς από τον πηγαίο κώδικα, τους οποίους προσπελαίνει με προγραμματιστικό τρόπο. Η αποθήκευση των πόρων σ'ένα μέρος είναι μια πιο οργανωμένη προσέγγιση για την ανάπτυξη και βοηθά στην πιο εύκολη ανάγνωση και συντήρηση του κώδικα. Για παράδειγμα, μπορεί να ορίζονται κινούμενα σχέδια, μενού, στυλ, χρώματα και η διάταξη των διεπαφών χρήστη της δραστηριότητας με αρχεία XML. Η χρήση των πόρων εφαρμογών καθιστά εύκολη την ενημέρωση διαφόρων χαρακτηριστικών της εφαρμογής χωρίς την τροποποίηση του κώδικα. Παρέχοντας σύνολα εναλλακτικών πόρων, μπορεί να βελτιστοποιηθεί η εφαρμογή για μια ποικιλία διαμορφώσεων συσκευών, όπως διαφορετικές γλώσσες και μεγέθη οθόνης.

Για καθέναν από τους πόρους που συμπεριλαμβάνεται σε ένα έργο Android, τα εργαλεία δημιουργίας του SDK ορίζουν ένα μοναδικό αναγνωριστικό ID. Για παράδειγμα, αν η εφαρμογή περιέχει ένα αρχείο εικόνας που ονομάζεται logo.png (αποθηκευμένο στον κατάλογο res / drawable /), τότε τα εργαλεία SDK δημιουργούν ένα αναγνωριστικό πόρου με όνομα R.drawable.logo. Το αναγνωριστικό μπορεί να χρησιμοποιηθεί για αναφορά στην εικόνα και την τοποθέτησή της στη διεπαφή χρήστη.

Η διαδικασία ξεχωρισμού των πόρων από τον πηγαίο κώδικα είναι σημαντική γιατί προσφέρει την δυνατότητα παροχής εναλλακτικών πόρων για διαφορετικές διαμορφώσεις συσκευών. Για παράδειγμα, όταν ορίζονται οι συμβολοσειρές των Μαργαρίτα Καραμολέγκου

διεπαφών χρήστη σε XML, μπορούν να μεταφραστούν οι συμβολοσειρές σε άλλες γλώσσες και να αποθηκευτούν σε ξεχωριστά αρχεία. Στη συνέχεια, το Android εφαρμόζει τις κατάλληλες συμβολοσειρές γλώσσας στη διεπαφή χρήστη με βάση ένα προσδιοριστικό γλώσσας (language qualifier) που προστίθεται στο όνομα του καταλόγου των πόρων (όπως res / values-fr / για τιμές γαλλικών συμβολοσειρών) και η γλώσσα επιλέγεται από τον χρήστη στις ρυθμίσεις γλώσσας της συσκευής χρήστη.

Το Android υποστηρίζει πολλά διαφορετικά προσδιοριστικά για τους εναλλακτικούς πόρους της εφαρμογής. Ο προσδιοριστής είναι μια σύντομη συμβολοσειρά που συμπεριλαμβάνεται στο όνομα των καταλόγων πόρων, προκειμένου να οριστεί η διαμόρφωση της συσκευής για την οποία πρέπει να χρησιμοποιηθούν οι πόροι αυτοί. Για παράδειγμα, θα πρέπει να υπάρχουν διαφορετικές διατάξεις για τις δραστηριότητες της εφαρμογής, ανάλογα με τον προσανατολισμό και το μέγεθος της οθόνης της συσκευής. Όταν η οθόνη της συσκευής είναι σε κατακόρυφο προσανατολισμό, ίσως να είναι αναγκαίο μια διάταξη με κουμπιά να είναι κάθετη, αλλά όταν η οθόνη είναι σε οριζόντιο προσανατολισμό, τα κουμπιά να μπορούσαν να ευθυγραμμιστούν οριζόντια. Για την αλλαγή της διάταξης ανάλογα με τον προσανατολισμό, ορίζονται δύο διαφορετικές διατάξεις και εφαρμόζεται το κατάλληλο προσδιοριστικό στο όνομα καταλόγου της κάθε διάταξης. Το σύστημα αυτόματα εφαρμόζει την κατάλληλη διάταξη σύμφωνα με τον τρέχοντα προσανατολισμό της συσκευής.

1.10 Βασικές Αρχές Δημιουργίας μιας Android εφαρμογής

Η σύλληψη μιας πρωτοποριακής, χρήσιμης και εμπορικής ιδέας για μια εφαρμογή Android είναι πάντα μια καλή αρχή. Όμως η πραγματική επιτυχία της εφαρμογής κρύβεται στον σχεδιασμό της. Το Android είναι η μεγαλύτερη εγκατεστημένη βάση από οποιαδήποτε πλατφόρμα κινητών συσκευών και αυξάνεται ταχέως – κάθε μέρα άλλο ένα εκατομμύριο χρήστες ενεργοποιούν την Android συσκευή τους για πρώτη φορά και αρχίζουν να ψάχνουν για εφαρμογές, παιχνίδια και άλλο ψηφιακό περιεχόμενο. Το επόμενο βήμα λοιπόν είναι η εφαρμογή να είναι προσβάσιμη στους περισσότερους από τους χρήστες που φορτώνουν το Android στις εκατοντάδες εκατομμύρια συσκευές τους σε πάνω από 190 χώρες ανά τον κόσμο. Οι συσκευές αυτές διαφέρουν σε έκδοση Android, γλώσσα και μέγεθος οθόνης, παράγοντες που πρέπει να ληφθούν υπόψιν κατά τον σχεδιασμό της εφαρμογής. Η μελέτη της προσαρμοστικότητας της εφαρμογής

στις ποικίλες παραμέτρους από συσκευή σε συσκευή, είναι αυτό που συνιστάται κατά τον σχεδιασμό μιας εφαρμογής.

1.11 Υποστήριξη διαφορετικών εκδόσεων του λειτουργικού συστήματος

Κάθε νέα έκδοση του Android εισάγει νέα χαρακτηριστικά, οπότε θα πρέπει να επιλεχθεί ο σχεδιασμός της εφαρμογής σύμφωνα με την δημοφιλέστερη έκδοση που κυκλοφορεί, ικανοποιώντας ταυτόχρονα τις ανάγκες της εφαρμογής.

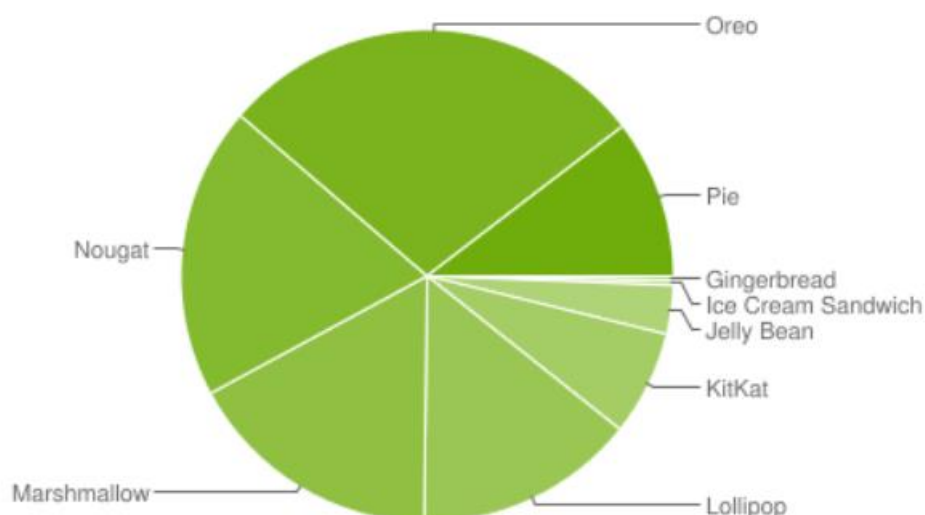
Οι τελευταίες εκδόσεις του Android συχνά παρέχουν τις καλύτερες διεπαφές των προγραμματιστικών διαδικασιών (API) για μια νέα εφαρμογή, ωστόσο η εφαρμογή θα πρέπει να συνεχίσει να υποστηρίζει παλαιότερες εκδόσεις του Android μέχρι να ενημερωθούν περισσότερες συσκευές.

Στην ιστοσελίδα της Google για τους προγραμματιστές Android εφαρμογών υπάρχει ταμπλό (dashboard) με τις εκδόσεις της πλατφόρμας Android, που ενημερώνεται τακτικά και δείχνει την κατανομή των ενεργών συσκευών που τρέχουν για κάθε έκδοση του Android, με βάση τον αριθμό των συσκευών που επισκέπτονται το Google Play Store. Γενικά, είναι μια καλή πρακτική η εφαρμογή να σχεδιάζεται με βάση την υποστήριξη περίπου 90% των ενεργών συσκευών, με παράλληλη στόχευση της εφαρμογής στην τελευταία έκδοση [10].

Λαμβάνοντας υπόψιν το ποσοστό των συσκευών που τρέχουν μια συγκεκριμένη έκδοση του Android (πίνακας 1.1 και το σχήμα 1.1), η εφαρμογή SantoriniTrail έχει αναπτυχθεί για να μπορεί να υποστηρίξει την έκδοση 4.1.x (Jelly Bean), τουλάχιστον, επειδή η έκδοση Jelly Bean εξακολουθεί να χρησιμοποιείται από το 1.2% των συσκευών [11].

Πίνακας 1.1: Ποσοστά συσκευών ανά έκδοση Android. Δεδομένα που συλλέχθηκαν κατά την διάρκεια μιας περιόδου 7 ημερών που τελείωσε στις 7 Μαΐου του 2019. Οι εκδόσεις με διανομή κάτω από 0,1% δεν απεικονίζονται στον πίνακα.

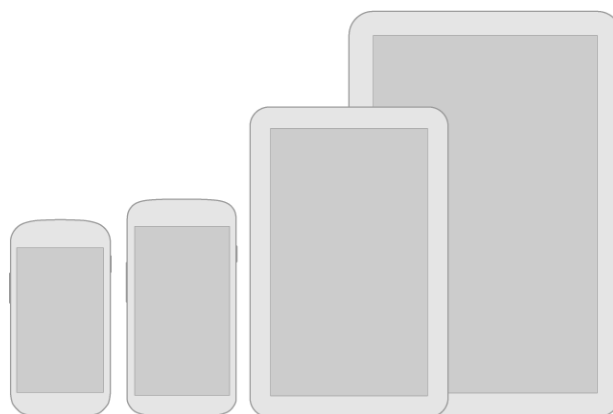
Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.3%
4.1.x	Jelly Bean	16	1.2%
4.2.x		17	1.5%
4.3		18	0.5%
4.4	KitKat	19	6.9%
5.0	Lollipop	21	3.0%
5.1		22	11.5%
6.0	Marshmallow	23	16.9%
7.0	Nougat	24	11.4%
7.1		25	7.8%
8.0	Oreo	26	12.9%
8.1		27	15.4%
9	Pie	28	10.4%



Σχήμα 1.2: Διάγραμμα πίτα που απεικονίζει τα δεδομένα του πίνακα 1.1

1.12 Υποστήριξη διαφορετικών οθονών

Μια εφαρμογή Android μπορεί να εγκατασταθεί σε συσκευές με οθόνες που κυμαίνονται στο μέγεθος και στο σχήμα τους. Ως εκ τούτου, θα πρέπει να περιλαμβάνει ορισμένα εναλλακτικά μέσα που βελτιστοποιούν την εμφάνισή της για διαφορετικά μεγέθη οθόνης και αναλογίες απεικόνισης. Το μέγεθος της οθόνης είναι ο ορατός χώρος που παρέχεται για την διεπαφή χρήστη μιας εφαρμογής. Το μέγεθος της οθόνης, δεν είναι το πραγματικό μέγεθος της οθόνης της συσκευής - λαμβάνει υπόψη τον προσανατολισμό της οθόνης, τα διακοσμητικά του συστήματος (όπως τη γραμμή πλοήγησης) και τις αλλαγές στην διαμόρφωση των παραθύρων (όπως όταν ο χρήστης ενεργοποιεί την λειτουργία πολλαπλών παραθύρων). Κάθε μέγεθος οθόνης προσφέρει διαφορετικές δυνατότητες και προκλήσεις για την αλληλεπίδραση του χρήστη. Για να ικανοποιηθεί και να εντυπωσιαστεί πραγματικά ο χρήστης, η εφαρμογή θα πρέπει να υπερβαίνει την απλή υποστήριξη διαφορετικών οθονών: Πρέπει να βελτιστοποιεί την εμπειρία του χρήστη για κάθε διαμόρφωση της οθόνης [12, 13].



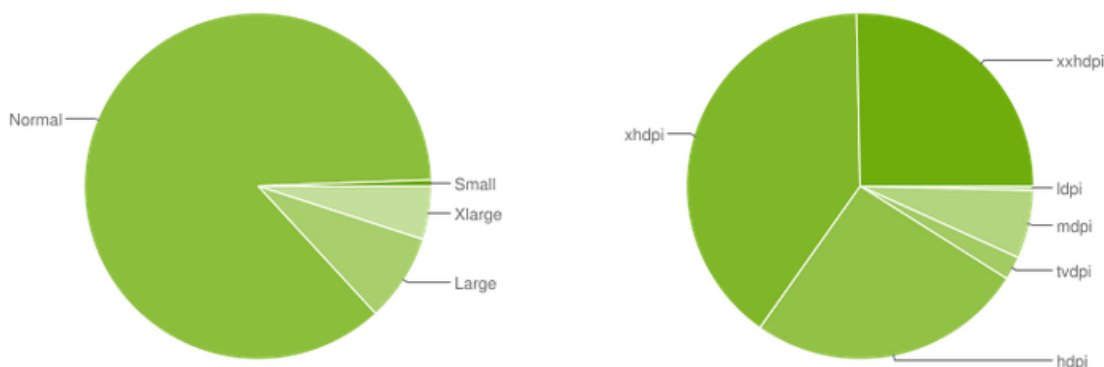
Εικόνα 1.1: Το μέγεθος «μετράει» κατά τον σχεδιασμό μιας εφαρμογής Android

Επίσης ο προσανατολισμός της οθόνης (οριζόντια ή κατακόρυφα) θεωρείται μια παραλλαγή του μεγέθους της οθόνης, οπότε θα πρέπει να μελετηθεί και η διάταξη για τη βελτιστοποίηση της εμπειρίας του χρήστη σε κάθε προσανατολισμό. Αξιοποιώντας τα ευέλικτα συστήματα των διατάξεων του Android μπορούν να σχεδιαστούν εφαρμογές που κλιμακώνουν στρωτά από την μια συσκευή στην άλλη, δημιουργώντας στον χρήστη την αίσθηση ότι η εφαρμογή έχει σχεδιαστεί συγκεκριμένα για την δική του συσκευή.

Στον πίνακα 1.2 εμφανίζονται δεδομένα σχετικά με το ποσοστό των ενεργών συσκευών, σύμφωνα με συγκεκριμένη διαμόρφωση οθόνης, που ορίζεται από ένα συνδυασμό του μεγέθους της οθόνης και την πυκνότητά της. Για να απλοποιηθεί ο τρόπος που γίνεται ο σχεδιασμός διεπαφών χρήστη για διαφορετικές διαμορφώσεις οθόνης, το Android χωρίζει το φάσμα των πραγματικών μεγεθών οθόνης και πυκνοτήτων, όπως φαίνονται στον παρακάτω πίνακα.

Πίνακας 1.2: Πίνακας και διαγράμματα πίτας που απεικονίζουν την διανομή συσκευών με διάφορες διατάξεις, σύμφωνα με δεδομένα που συλλέχθηκαν σε μία περίοδο 7 ημερών που τελείωσε στις 7 Μαΐου του 2019. Οι εκδόσεις με διανομή κάτω από 0,1% δεν απεικονίζονται.

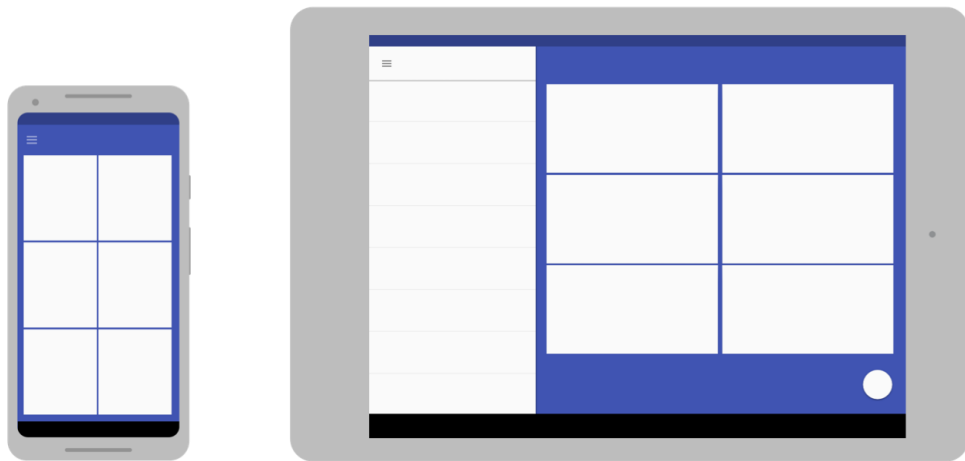
	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Total
Small	0.4%				0.1%	0.1%	0.6%
Normal		0.9%	0.3%	24.0%	37.7%	23.6%	86.5%
Large		2.4%	1.9%	0.6%	1.6%	1.7%	8.2%
Xlarge		3.1%		1.3%	0.6%		5.0%
Total	0.4%	6.4%	2.2%	25.9%	40.0%	25.4%	



Κάποιες πρακτικές που βοηθούν στην βελτιστοποίηση του σχεδιασμού για διαφορετικές διατάξεις είναι:

- Να διασφαλιστεί ότι η διάταξη αλλάζει σωστά μέγεθος για ακόμη και μικρές παραλλαγές στο μέγεθος της οθόνης. Για να γίνει αυτό πρέπει να ληφθεί υπόψη η ευελιξία κατά τον σχεδιασμό της διάταξης. Από προεπιλογή, το Android αλλάζει το μέγεθος της διάταξης της εφαρμογής ώστε να ταιριάζει στην τρέχουσα οθόνη. Η βασική αρχή που πρέπει να ακολουθηθεί είναι η αποφυγή της ενσωμάτωσης της θέσης και του μεγέθους των στοιχείων της διεπαφής χρήστη στον κώδικα της εφαρμογής. Αντ' αυτού, τα μεγέθη προβολών πρέπει να μπορούν να επεκτείνονται και να καθορίζονται οι θέσεις προβολής σε σχέση με το parent view ή άλλες σχετικές προβολές, έτσι ώστε η επιθυμητή σειρά και τα σχετικά μεγέθη να παραμείνουν ίδια καθώς αναπτύσσεται η διάταξη.

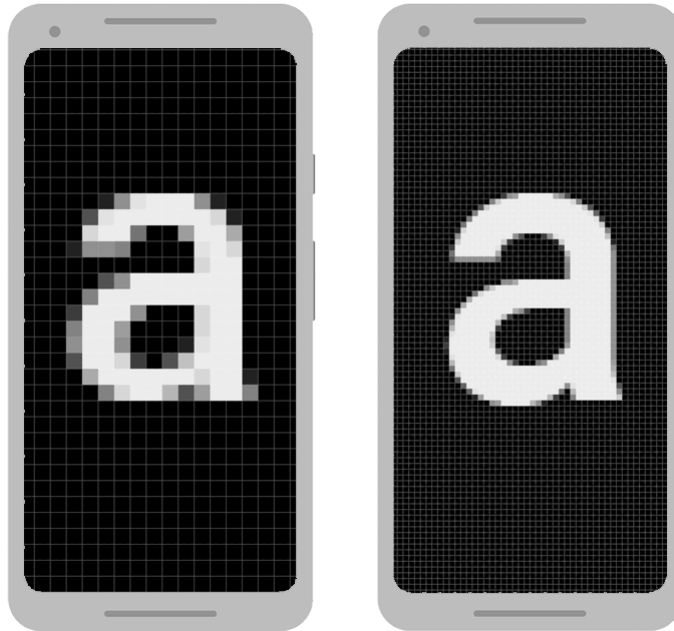
- Η δημιουργία μοναδικών αρχείων XML για κάθε διάταξη (layout) και αποθήκευσή τους στον αντίστοιχο φάκελο πόρων (resource file). Μια ευέλικτη διάταξη είναι πολύ σημαντική, αλλά και ο σχεδιασμός διαφορετικών διατάξεων βελτιστοποιεί την εμπειρία του χρήστη για το διαθέσιμο χώρο σε διαφορετικές συσκευές όπως τα τηλέφωνα και τα tablet. Έτσι, το Android επιτρέπει να παρέχονται εναλλακτικά αρχεία διαμόρφωσης που εφαρμόζει το σύστημα κατά το χρόνο εκτέλεσης με βάση το μέγεθος της οθόνης της τρέχουσας συσκευής.



Εικόνα 1.3. Η ίδια εφαρμογή χρησιμοποιεί διαφορετική διάταξη για διαφορετικά μεγέθη οθόνης

- Επειδή η διάταξη πρέπει να επεκταθεί ώστε να ταιριάζει με την τρέχουσα οθόνη, το ίδιο θα πρέπει να κάνουν και οι εικόνες bitmap που συνδέονται με οποιαδήποτε από τις προβολές διάταξης. Ωστόσο, η επέκταση ενός συνηθισμένου bitmap σε αυθαίρετες κατευθύνσεις μπορεί να οδηγήσει σε παράξενη κλιμάκωση και σε παραμορφωμένες εικόνες. Για να λυθεί αυτό, το Android υποστηρίζει τα nine-patch bitmaps, στα οποία καθορίζονται μικρές περιοχές εικονοστοιχείων που είναι επεκτάσιμα - η υπόλοιπη εικόνα παραμένει σταθερή.
- Η πυκνότητα των εικονοστοιχείων (pixel density) είναι ο αριθμός των εικονοστοιχείων μέσα σε μια φυσική περιοχή της οθόνης και αναφέρεται ως dpi (κουκίδες ανά ίντσα). Αυτό διαφέρει από την ανάλυση (resolution), που είναι ο συνολικός αριθμός εικονοστοιχείων σε μια οθόνη. Η εφαρμογή επιτυγχάνει την "ανεξαρτησία πυκνότητας" όταν διατηρεί το φυσικό μέγεθος (από την άποψη του χρήστη) στον σχεδιασμό της διεπαφής χρήστη όταν εμφανίζεται σε οθόνες με

διαφορετικές πυκνότητες εικονοστοιχείων (όπως φαίνεται στην εικόνα 1.4). Η διατήρηση της ανεξαρτησίας πυκνότητας είναι σημαντική επειδή, χωρίς αυτήν, ένα στοιχείο της διεπαφής χρήστη (όπως ένα κουμπί) μπορεί να φαίνεται μεγαλύτερο σε μια οθόνη χαμηλής πυκνότητας και μικρότερο σε μια οθόνη υψηλής πυκνότητας. Το σύστημα Android βοηθά την επίτευξη της ανεξαρτησίας πυκνότητας παρέχοντας τα *density-independent pixels* (dp or dip) ως μονάδα μέτρησης που πρέπει να χρησιμοποιείται αντί των pixel (px).



Εικόνα 1.4. Μια υπερβολική απεικόνιση δύο συσκευών που έχουν το ίδιο μέγεθος αλλά διαφορετικές πυκνότητες εικονοστοιχείων

- Για να εμφανίζονται οι εικόνες όσο το δυνατόν καλύτερα σε όλες τις οθόνες, πρέπει να παρέχονται εναλλακτικές εικόνες bitmap που ταιριάζουν με κάθε πυκνότητα οθόνης. Για παράδειγμα, εάν η εφαρμογή παρέχει bitmaps μόνο για οθόνες μέσης πυκνότητας (mdpi), το Android τις κλιμακώνει όταν είναι σε οθόνη υψηλής πυκνότητας, έτσι ώστε η εικόνα να καταλαμβάνει τον ίδιο φυσικό χώρο στην οθόνη. Αυτή η κλιμάκωση μπορεί να είναι ορατή στα bitmaps και για αυτό συνίσταται η εφαρμογή να περιλαμβάνει εναλλακτικές εικόνες bitmap σε υψηλότερη ανάλυση.
- Με την χρήση των διανυσματικών γραφικών σε απλά εικονίδια, μπορεί να αποφευχθεί η δημιουργία ξεχωριστών εικόνων για κάθε πυκνότητα. Επειδή τα διανυσματικά γραφικά ορίζουν την απεικόνιση με γεωμετρικά μονοπάτια γραμμής

αντί για εικονοστοιχεία, μπορούν να σχεδιαστούν σε οποιοδήποτε μέγεθος χωρίς να κλιμακώνεται το αντικείμενο.

- Τα εργαλεία και το πλαίσιο λειτουργίας του Android παρέχουν όλα όσα χρειάζονται για την δημιουργία μιας εφαρμογής που θα είναι διαθέσιμη σε όλες τις διαμορφώσεις οθόνης. Ωστόσο, ο προγραμματιστής έχει την επιλογή να σχεδιάσει μια εφαρμογή για ορισμένες διαμορφώσεις οθόνης λόγω κάποιου είδους ασυμβατότητας. Σε αυτή την περίπτωση, πρέπει να δηλωθεί ότι η εφαρμογή υποστηρίζει μόνο συγκεκριμένες οθόνες.

1.13. Τοπικοποίηση εφαρμογής

Σήμερα, άνθρωποι από όλα τα μήκη της γης μοιράζονται την εμπειρία της χρήσης των έξυπνων συσκευών τους, ενώ μιλούν διαφορετικές γλώσσες, γράφουν από αριστερά προς τα δεξιά ή από δεξιά προς τα αριστερά, έχουν διαφορετικά νομίσματα και διαφορετικό τρόπο απεικόνισης της ημερομηνίας.

Για την προσέγγιση περισσότερων χρηστών, μια εφαρμογή Android πρέπει να χειρίζεται κείμενα, αρχεία ήχου, αριθμούς, νομίσματα και γραφικά με τρόπους κατάλληλους για τις τοποθεσίες όπου χρησιμοποιείται η εφαρμογή.

Με την τοπικοποίηση (localization) μιας εφαρμογής μεγιστοποιείται και το δυναμικό διανομής της. Ένα από τα βασικά βήματα που πρέπει να σχεδιαστούν κατά τον κύκλο ανάπτυξης της εφαρμογής για να προσδίδουν σε επιτυχημένες τοπικές εκκινήσεις από το Google Play είναι η διαφοροποίηση του κώδικα ώστε να εξυπηρετεί τις τοπικές αγορές. Είναι καλή πρακτική να χρησιμοποιούνται οι τυποποιημένες μορφές που παρέχονται από το σύστημα για ημερομηνίες, ώρες, αριθμούς και νομίσματα και όλοι οι πόροι που σχετίζονται με την κουλτούρα των χρηστών να διαχωρίζονται από το υπόλοιπο της εφαρμογής [14]. Για την προσαρμογή στην κουλτούρα των ατόμων που χρησιμοποιούν μια εφαρμογή, το Android προσφέρει την δυνατότητα να επιλέγονται συγκεκριμένοι πόροι από τον κατάλογο πόρων της εφαρμογής σύμφωνα με τις ρυθμίσεις του τοπικού συστήματος που επιλέγει ο χρήστης στην συσκευή Android.

Κατά την ανάπτυξη της εφαρμογής πρέπει να ληφθούν υπόψη όλες οι συμβολοσειρές που χρειάζονται μετάφραση και να μετακινηθούν σε αρχείο `string.xml`. Επίσης στον κώδικα πρέπει να υπάρχουν επαρκείς πληροφορίες που να επεξηγούν στον

μεταφραστή τι πρέπει να μεταφραστεί και τι όχι. Επίσης πρέπει να χρησιμοποιηθούν ευέλικτες διατάξεις ανάλογες με τις τοπικές ρυθμίσεις του χρήστη.

Για παράδειγμα, οι χρήστες που χρησιμοποιούν την γραφή από δεξιά προς τα αριστερά προτιμούν μια διεπαφή χρήστη από τα δεξιά προς τα αριστερά, η οποία περιλαμβάνει μενού ευθυγραμμισμένο στο δεξιό μέρος, κείμενο ευθυγραμμισμένο με το κείμενο στα δεξιά και βέλη που δείχνουν προς τα αριστερά. Μέσω ξεχωριστών φακέλων για τους πόρους της εφαρμογής μπορούν να είναι διαθέσιμες οι εναλλακτικές διατάξεις στην οθόνη.

Για την μέγιστη απόδοση της εφαρμογής πρέπει να γίνει έρευνα της αγοράς και να προσδιοριστούν οι γλώσσες και οι χώρες όπου μπορεί να υπάρχουν ευκαιρίες για την εφαρμογή. Επίσης πρέπει να ερευνηθεί αν υπάρχει η δυνατότητα παροχής της κατάλληλης υποστήριξης χρηστών σε αυτές τις αγορές [15,16].

1.14. Ροή για την καλύτερη εμπειρία χρήστη

Κατά την σχεδίαση μιας εφαρμογής πρέπει να δημιουργείται μια διεπαφή χρήστη που να είναι οικεία στους χρήστες, χωρίς να παραλείπονται στοιχεία που τους είναι γνώριμα από τις δημοφιλείς εφαρμογές που χρησιμοποιούν καθημερινά. Κάποια από τα κριτήρια που διασφαλίζουν ότι μια εφαρμογή παρέχει τα τυποποιημένα πρότυπα σχεδίασης και αλληλεπίδρασης Android, που χρειάζονται για μια συνεπή εμπειρία χρήστη είναι [17]:

- Η εφαρμογή δεν επαναπροσδιορίζει την αναμενόμενη λειτουργία ενός εικονιδίου συστήματος (όπως το κουμπί Επιστροφή).
- Η εφαρμογή δεν αντικαθιστά ένα εικονίδιο συστήματος με ένα εντελώς διαφορετικό εικονίδιο.
- Εάν η εφαρμογή παρέχει μια προσαρμοσμένη έκδοση ενός τυποποιημένου εικονιδίου συστήματος, το εικονίδιο μοιάζει πολύ με το εικονίδιο του συστήματος και ενεργοποιεί την τυπική συμπεριφορά του συστήματος.
- Η εφαρμογή δεν επαναπροσδιορίζει και δεν κάνει κακή χρήση των προτύπων της διεπαφής χρήστη του Android, έτσι ώστε τα εικονίδια ή οι συμπεριφορές να είναι παραπλανητικές ή να προκαλούν σύγχυση στους χρήστες.
- Η εφαρμογή υποστηρίζει την τυποποιημένη πλοήγηση του συστήματος για την επιστροφή σε προηγούμενη οθόνη.

- Η απόκρυψη των παραθύρων διαλόγου γίνεται χρησιμοποιώντας το κουμπί Επιστροφή.
- Οποτεδήποτε πατηθεί το πλήκτρο Home γίνεται μετάβαση στην αρχική οθόνη της συσκευής
- Οι ειδοποιήσεις ακολουθούν τις οδηγίες σχεδίασης Android.
- Πολλαπλές ειδοποιήσεις στοιβάζονται σε ένα αντικείμενο ειδοποίησης, όπου αυτό είναι δυνατόν.
- Οι ειδοποιήσεις παραμένουν στην οθόνη μόνο αν σχετίζονται με τρέχοντα συμβάντα (όπως αναπαραγωγή μουσικής ή τηλεφωνική κλήση).
- Οι ειδοποιήσεις δεν περιέχουν διαφημίσεις ή περιεχόμενο που δεν σχετίζεται με τη βασική λειτουργία της εφαρμογής, εκτός εάν ο χρήστης το έχει επιτρέψει.
- Η εφαρμογή χρησιμοποιεί ειδοποιήσεις μόνο για να υποδείξει μια αλλαγή στο πλαίσιο που σχετίζεται με τον χρήστη προσωπικά (όπως ένα εισερχόμενο μήνυμα), ή για να εκθέσει πληροφορίες / χειριστήρια σχετικά με ένα συνεχιζόμενο συμβάν (όπως αναπαραγωγή μουσικής ή τηλεφωνική κλήση).

1.15 Παγκόσμιο Σύστημα Στιγματοθέτησης και οι Υπηρεσίες Βάσει Τοποθεσίας

Το Παγκόσμιο Σύστημα Στιγματοθέτησης, το κοινώς γνωστό ως GPS, είναι ένα είδος τεχνολογίας πλοήγησης που αποτελείται από ένα δίκτυο 24 δορυφόρων και από επίγειους σταθμούς κατανεμημένους σε όλο τον κόσμο. Παρέχει λεπτομέρειες για την ακριβή θέση και τον χρόνο ανεξάρτητα από το πού βρίσκεται ο χρήστης του. Δεν επηρεάζεται από κακές καιρικές συνθήκες ή οποιοδήποτε άλλο εμπόδιο στη διαδικασία πλοήγησης. Το GPS χρησιμοποιείται ευρέως σε πολλούς τομείς όπως στον στρατό, στην επιστήμη, την τεχνολογία, το εμπόριο, τη γεωλογία, τις τηλεπικοινωνίες κλπ. [18,19]. Η τεχνολογία του GPS ανήκει στην Αμερικανική κυβέρνηση και εποπτεύεται από την Πολεμική Αεροπορία της χώρας, αλλά είναι δωρεάν για όλους να την χρησιμοποιούν [20]. Το GPS ξεκίνησε έχοντας αποκλειστικά στρατιωτικούς σκοπούς, και χρησιμοποιήθηκε για να καθοδηγεί αεροσκάφη, στρατιώτες και ακόμη και βόμβες. Στις περισσότερες περιπτώσεις, οι δέκτες τοποθετήθηκαν σε ανοικτές περιοχές που είχαν πρόσβαση σε δορυφόρους. Ωστόσο από τότε που το GPS ανοίχθηκε για εμπορική χρήση, νέες εφαρμογές εισήγαγαν μεγαλύτερες απαιτήσεις στο σύστημα.

Αυτές οι νέες εφαρμογές απαιτούσαν σήματα GPS για να φτάσουν σε σημεία που αποκλείονταν από κάποιο είδος εμποδίων, όπως δέντρα ή στέγες. Έτσι, δημιουργήθηκε η έννοια του υποβοηθούμενου GPS το A-GPS [21]. Το AGPS είναι η καλύτερη λύση τις περισσότερες φορές. Οι χρήστες θέλουν το τηλέφωνό τους να γνωρίζει την τοποθεσία τους σε πραγματικό χρόνο, να μην χρησιμοποιείται πολύ ενέργεια από την μπαταρία και να μπορεί να γίνει ανανέωση όποτε το απαιτεί το λογισμικό, χωρίς να περιμένει για ένα καλό σήμα GPS. Η τοποθεσία AGPS δεν είναι τόσο ακριβής όσο θα είναι μια πραγματική θέση GPS, αλλά είναι αρκετά καλή για σχεδόν κάθε περίπτωση χρήσης και οι μικρορυθμίσεις που μπορούν να γίνουν με πραγματικά δεδομένα GPS όταν ανανεώνονται, αντισταθμίζουν τις περισσότερες αποκλίσεις.

Το υποβοηθούμενο GPS (A-GPS) είναι ένα σύστημα που επιτρέπει στους δέκτες του συστήματος παγκόσμιας στιγματοθέτησης (GPS) να λαμβάνουν πληροφορίες από πόρους δικτύου για να βοηθήσουν στη δορυφορική θέση. Ένα σύστημα A-GPS είναι ιδιαίτερα χρήσιμο όταν ο δέκτης βρίσκεται σε μια θέση όπου είναι δύσκολο να δεισδύσουν τα δορυφορικά σήματα. Εκτός από την καλύτερη κάλυψη που προσφέρει, το A-GPS βελτιώνει επίσης τον χρόνο εκκίνησης, ο οποίος είναι ο χρόνος που απαιτείται από τους δορυφόρους και τους δέκτες για τη δημιουργία αξιόπιστης σύνδεσης. Αυτό αρχικά χρειαζόταν ένα λεπτό περίπου. Για ακόμα καλύτερη κάλυψη, ορισμένα κινητά τηλέφωνα κάνουν χρήση ενός συνδυασμού A-GPS και άλλων τεχνολογιών που βασίζονται σε τοποθεσίες, όπως το σύστημα εντοπισμού θέσης Wi-Fi και το τρίγωνο θέσης κυψέλης.

Ένα παλιό ρητό λέει ότι το κλειδί για να τα βγάλεις πέρα στη ζωή είναι να γνωρίζεις το πού βρίσκεσαι πού ήσουν και το πού πηγαίνεις. Στον σημερινό ψηφιακό κόσμο, υπάρχουν app για αυτό.

Οι υπηρεσίες βάσει τοποθεσίας (Location-based services / LBS) είναι από τις πιο δημοφιλείς υπηρεσίες που βασίζονται σε διαφορετικές τεχνολογίες πλοήγησης που παρέχονται από το δίκτυο κινητής επικοινωνίας [22,23]. Οι εφαρμογές που βασίζονται στην γνώση της τοποθεσίας απέκτησαν μεγάλη δυναμική τα τελευταία χρόνια πολλαπλασιάζοντας τον αριθμό των χρηστών. Οι υπηρεσίες βάσει τοποθεσίας λαμβάνουν τις συντεταγμένες θέσης από τον κινητό σταθμό εδάφους και τις στέλνουν στον χρήστη του κινητού τηλεφώνου και στο κέντρο επικοινωνίας ώστε να μπορούν να χρησιμοποιηθούν για διάφορες υπηρεσίες βάσει τοποθεσίας. Έτσι, με λίγα λόγια, το LBS είναι μια ασύρματη υπηρεσία που καθορίζει τις πληροφορίες για την θέση του

χρήστη της συσκευής πληροφοριών (κινητό) χρησιμοποιώντας το σύστημα γεωγραφικών πληροφοριών (GIS) και άλλες δορυφορικές πλατφόρμες πλοήγησης.

Οι εφαρμογές με υπηρεσίες βάσει τοποθεσίας μπορούν να βοηθήσουν τους χρήστες τους να αφήνουν άλλους να παρακολουθήσουν τις κινήσεις τους, να βρουν ή να φτάσουν σε κάποιο μέρος, ή απλά να γνωρίζουν την τρέχουσα τοποθεσία τους και τι είναι γύρω τους.

Από τα τέλη της δεκαετίας του 1990, οι εφαρμογές LBS προέκυψαν από τη σύγκλιση τριών διαφορετικών τεχνολογιών: πρόσβαση στο κινητό διαδίκτυο, παγκόσμια στιγματοθέτηση και προηγμένες γραφικές διεπαφές. Σήμερα διαπιστώνεται ότι ο αριθμός των χρηστών LBS κινητής τηλεφωνίας έχει αυξηθεί σταθερά τα τελευταία χρόνια.

Μια πρόσφατη έρευνα, επιβεβαιώνει ότι η πλοήγηση με LBS έχει καταστεί βασικό στοιχείο των χρηστών κινητής τηλεφωνίας, με τους Χάρτες Google να είναι η συντριπτική επιλογή για τους χρήστες κινητών τηλεφώνων [24]. Η προτίμηση στους χάρτες της Google οφείλεται, στα εκτενή δεδομένα και στην προσοχή στη λεπτομέρεια που προσφέρει η Google.

ΚΕΦΑΛΑΙΟ 2

ΑΝΑΛΥΣΗ

Σε αυτό το κεφάλαιο αναλύονται οι λειτουργικές και οι μη-λειτουργικές απαιτήσεις της εφαρμογής, εξετάζονται οι περιπτώσεις χρήσης της εφαρμογής, όπου αναδεικνύεται και η αλληλεπίδραση μεταξύ του χρήστη και της εφαρμογής. Ο σωστός σχεδιασμός του έργου είναι μια σημαντική δραστηριότητα. Μέσω της δομής του έργου, γίνεται η παρακολούθηση της προόδου του έργου, εκτιμάται ο προϋπολογισμός και ο χρόνος ολοκλήρωσής του. Κατά τον σχεδιασμό του έργου, πρέπει επίσης να εκτιμούνται οι κίνδυνοι, που μπορεί να επηρεάσουν την υλοποίηση του έργου. Η προετοιμασία για τους κινδύνους μειώνει τις επιπτώσεις τους.

2.1 Λειτουργικές και μη-λειτουργικές απαιτήσεις εφαρμογής

Ένα από τα πρώτα βήματα στην σχεδίαση ενός συστήματος λογισμικού είναι η καταγραφή των υπηρεσιών που παρέχονται από το σύστημα και οι λειτουργικοί περιορισμοί του, δηλαδή οι απαιτήσεις του συστήματος. Αυτές κατατάσσονται σε λειτουργικές και μη λειτουργικές απαιτήσεις.

Λειτουργική απαίτηση είναι κάθε απαίτηση που καθορίζει τι πρέπει να κάνει το σύστημα. Με άλλα λόγια, μια λειτουργική απαίτηση θα περιγράφει μια συγκεκριμένη συμπεριφορά λειτουργίας του συστήματος όταν πληρούνται ορισμένες προϋποθέσεις, όπως για παράδειγμα: "Αποστολή μηνύματος ηλεκτρονικού ταχυδρομείου όταν ο χρήστης πατήσει το κουμπί «επικοινωνία» ή «Εμφάνιση τοποθεσίας του χρήστη στον χάρτη».

Πίνακας 2.1: Λειτουργικές απαιτήσεις εφαρμογής

A/A	Λειτουργικές απαιτήσεις	Περιγραφή
1	Αναζήτηση συντεταγμένων (τοποθεσίας) του χρήστη	Βρίσκει τις συντεταγμένες του χρήστη την στιγμή που χρησιμοποιεί την εφαρμογή
2	Επιλογή μονοπατιού	Ο χρήστης μπορεί να επιλέξει ένα από τα 15 μονοπάτια
3	Ανάγνωση περιγραφής μονοπατιού	Εμφανίζεται ο χάρτης και σημάδια για την αρχή και το τέλος του μονοπατιού το υψόμετρο, το μήκος του και ο εκτιμώμενος χρόνος ολοκλήρωσης.
4	Πάτημα (Click) όταν έχει πραγματοποιηθεί το μονοπάτι.	Ο χρήστης επιλέγει ημερομηνία και ώρα πραγματοποίησης του μονοπατιού.
5	Εμφάνιση σχολίων	Ο χρήστης μπορεί να εισάγει σχόλια σχετικά με τη διαδρομή και να διαβάσει αποθηκευμένα σχόλια αν υπάρχουν.
6	Επιλογή επιστροφής στο χάρτη	Από το μενού ο χρήστης μπορεί να επιλέξει το κουμπί HOME και να επιστρέψει στο χάρτη.
7	Κοινοποίηση	Ο χρήστης μπορεί να επιλέξει SHARE στο μενού και να κοινοποιήσει την εφαρμογή
8	Αποστολή e-mail	Ο χρήστης μπορεί να στείλει μήνυμα ηλεκτρικού ταχυδρομείου στον δημιουργό της εφαρμογής
9	Έξοδος	Ο χρήστης μπορεί να εξέλθει της εφαρμογής πατώντας το κουμπί QUIT
10	Ορισμός γλώσσας εφαρμογής	Ο χρήστης μπορεί να χρησιμοποιήσει την εφαρμογή στα Ελληνικά ή στα Αγγλικά σύμφωνα με τις ρυθμίσεις του κινητού του

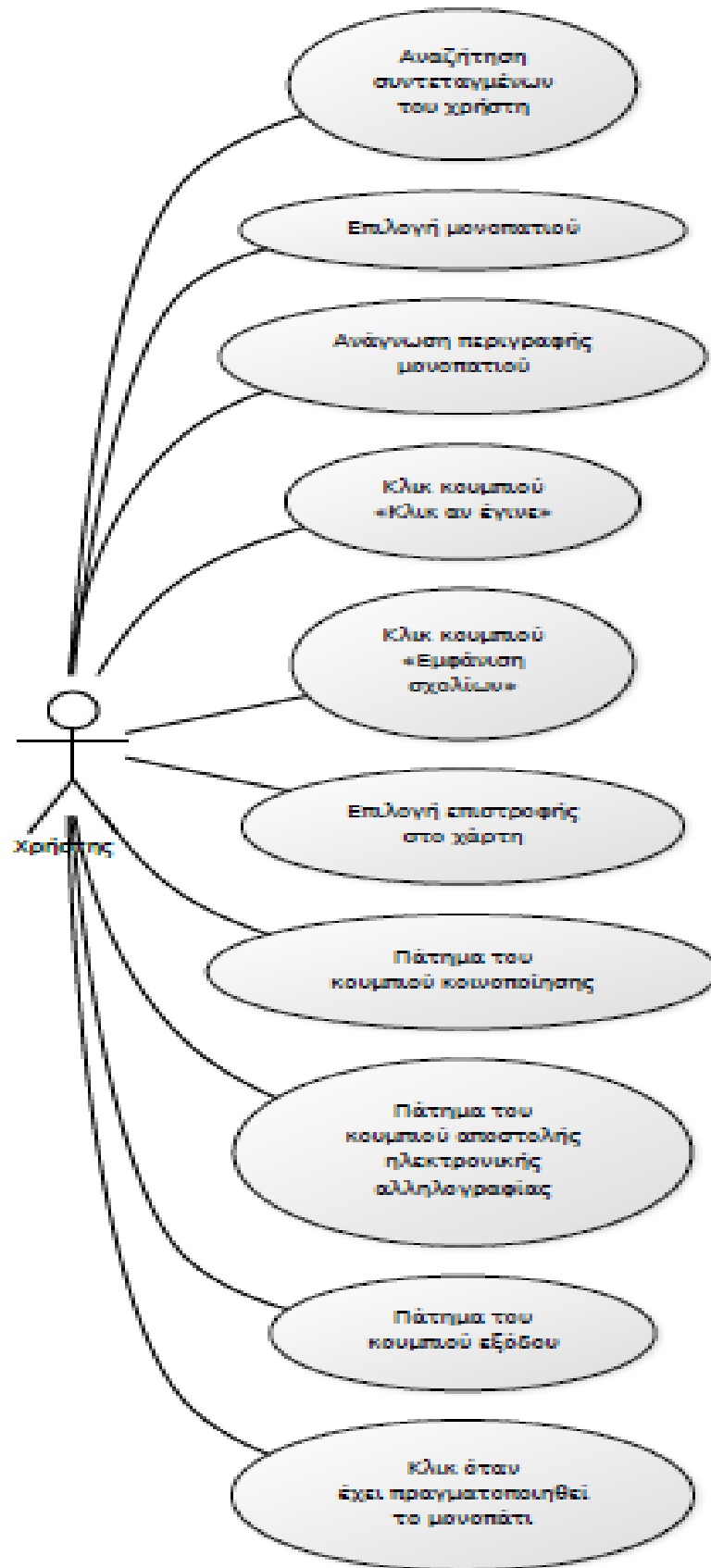
Οι μη λειτουργικές απαιτήσεις καθορίζουν τον τρόπο με τον οποίο το σύστημα εκτελεί μια συγκεκριμένη λειτουργία. Με άλλα λόγια, μια μη λειτουργική απαίτηση θα περιγράψει πώς πρέπει να συμπεριφέρεται ένα σύστημα και ποια όρια υπάρχουν στη λειτουργικότητά του. Οι μη λειτουργικές απαιτήσεις καθορίζουν γενικά χαρακτηριστικά ή τα χαρακτηριστικά ποιότητας του συστήματος, για παράδειγμα: η απόδοση, ο χρόνος απόκρισης, η ευελιξία, το εύχρηστο περιβάλλον.

Πίνακας 2.2: Μη Λειτουργικές απαιτήσεις εφαρμογής

A/A	Μη Λειτουργική απαίτηση	Περιγραφή
1	Σχεδιασμός	Η εφαρμογή έχει απλή διεπαφή χρήσης
2	Πλατφόρμα	Η εφαρμογή είναι διαθέσιμη για κινητά και τάμπλετ που έχουν λειτουργικό σύστημα Android
3	Επικοινωνία	Η εφαρμογή πρέπει να εντοπίζει έγκυρο σύστημα GPS, αλλά δεν χρειάζεται να υπάρχει πρόσβαση στο διαδίκτυο.

2.2 Περιπτώσεις Χρήσης

Οι περιπτώσεις χρήσης είναι ένα σύνολο σεναρίων που απεικονίζουν το πώς η εφαρμογή χρησιμοποιείται από τους χρήστες. Έχουν στόχο να καθορίσουν και να περιγράψουν τις λειτουργικές απαιτήσεις του συστήματος, να δώσουν μια σαφή και συνεπή περιγραφή για το τι θα πρέπει να κάνει το σύστημα, και τέλος να παρέχουν την κατάλληλη βάση για να γίνονται έλεγχοι για την επαλήθευση του συστήματος. Στο Σχήμα 2.1 απεικονίζονται οι περιπτώσεις χρήσης που προσφέρονται στον χρήστη. Στους παρακάτω πίνακες των περιπτώσεων χρήσης απεικονίζεται ξεχωριστά η κάθε αλληλεπίδραση μεταξύ του χρήστη και της εφαρμογής.



Σχήμα 2.1: Διάγραμμα περιπτώσεων χρήσης

Πίνακας 2.3: Περίπτωση χρήσης 1

Κωδικός Περίπτωσης:	Use Case -1
Ονομασία	Αναζήτηση συντεταγμένων του χρήστη
Εμπλεκόμενοι ρόλοι	Χρήστης
Φυσιολογική ροή	Ο χρήστης ανοίγει την εφαρμογή και εμφανίζεται ο χάρτης της Σαντορίνης και μια μπλε βούλα που αντιπροσωπεύει το στίγμα του
Εναλλακτική ροή	Ο χρήστης δεν βρίσκεται στη Σαντορίνη. Στην οθόνη δεν εμφανίζεται το στίγμα του.
Εξαιρέσεις	Ο χρήστης έχει κλειστό το GPS. Δεν εμφανίζεται το στίγμα του.

Πίνακας 2.4: Περίπτωση χρήσης 2

Κωδικός Περίπτωσης:	Use Case -2
Ονομασία	Επιλογή μονοπατιού
Εμπλεκόμενοι ρόλοι	Χρήστης
Φυσιολογική ροή	<ol style="list-style-type: none"> 1) Ο χρήστης πατάει στην επάνω δεξιά γωνία της εφαρμογής όπου βρίσκεται η συντόμευση με τις 3 τελείες 2) Εμφανίζεται η επιλογή «Επέλεξε μονοπάτι» 3) Ο χρήστης πατάει στην επιλογή «Επέλεξε μονοπάτι» 4) Εμφανίζεται μια λίστα με τα 15 διαθέσιμα μονοπάτια στην Σαντορίνη 5) Ο χρήστης επιλέγει μονοπάτι πατώντας σε αυτό που τον ενδιαφέρει
Εναλλακτική ροή	<ol style="list-style-type: none"> 1) Ο χρήστης πατάει στην αριστερή επάνω γωνία με τις 3 οριζόντιες γραμμές και εμφανίζεται το μενού της εφαρμογής. 2) Εμφανίζονται διάφορες επιλογές, μεταξύ των οποίων και η επιλογή «Επέλεξε μονοπάτι»

Πίνακας 2.5: Περίπτωση χρήσης 3

Κωδικός Περίπτωσης:	Use Case -3
Ονομασία	Ανάγνωση περιγραφής μονοπατιού
Εμπλεκόμενοι ρόλοι	Χρήστης
Προϋποθέσεις	Έχει επιλεγθεί μονοπάτι
Φυσιολογική ροή	<ol style="list-style-type: none"> 1) Εμφανίζεται ο χάρτης με την αρχή και το τέλος του μονοπατιού και πληροφορίες για την ανύψωση, το μήκος της διαδρομής, ο εκτιμώμενος χρόνος ολοκλήρωσης της διαδρομής, και μια περιγραφή της διαδρομής. Στην μπάρα ενεργειών εμφανίζεται το όνομα της διαδρομής και από κάτω χαρακτηρισμός για το βαθμό δυσκολίας της. 2) Ο χρήστης διαβάζει όσα παρουσιάζονται.

Πίνακας 2.6: Περίπτωση χρήσης 4

Κωδικός Περίπτωσης:	Use Case -4
Ονομασία	Πάτημα (Click) κουμπιού «Κλικ αν έγινε»
Εμπλεκόμενοι ρόλοι	Χρήστης
Προϋποθέσεις	Έχει επιλεγθεί μονοπάτι
Φυσιολογική ροή	<ol style="list-style-type: none"> 1) Εμφανίζεται στον χρήστη ημερολόγιο για την επιλογή της ημερομηνίας πραγματοποίησης του μονοπατιού. 2) Επιλογή του κουμπιού OK, αφού έχουν εισαχθεί τα δεδομένα. 3) Επιστροφή στην προηγούμενη οθόνη, όπου έχει αλλάξει η λεζάντα του κουμπιού που πατήθηκε και δηλώνει ότι πραγματοποιήθηκε το μονοπάτι.
Εναλλακτική ροή	<p>Επιλογή του κουμπιού ακύρωσης, αν ο χρήστης δεν θέλει να εισαχθούν τα δεδομένα.</p> <p>Επιστροφή στην προηγούμενη οθόνη. Η λεζάντα του κουμπιού που πατήθηκε παραμένει το ίδιο.</p>

Πίνακας 2.7: Περίπτωση χρήσης 5

Κωδικός Περίπτωσης:	Use Case -5
Ονομασία	Πάτημα (Click) κουμπιού «Εμφάνιση σχολίων»
Εμπλεκόμενοι ρόλοι	Χρήστης
Προϋποθέσεις	Έχει επιλεγθεί μονοπάτι
Φυσιολογική ροή	<p>1) Εμφανίζεται μια οθόνη με πεδίο εισαγωγής σχολίων και τον κέρσορα έτοιμο στο σημείο γραφής. Επίσης εμφανίζεται το κουμπί «Καταχώρησε» και η μπάρα ενεργειών με το όνομα του μονοπατιού, και από κάτω αναφέρει ο χρήστης βρίσκεται στην οθόνη για τα σχόλια.</p> <p>2) Ο χρήστης εισάγει τα σχόλια του για την διαδρομή.</p> <p>3) Πατάει το κουμπί «Καταχώρησε»</p> <p>4) Εμφανίζονται στην οθόνη τα σχόλιά του, η ημερομηνία και η ώρα εισαγωγής των σχολίων.</p> <p>5) Με παρατεταμένο πάτημα στο σχόλιο, εμφανίζεται αναδυόμενο παράθυρο, με την ερώτηση διαγραφής του σχολίου.</p> <p>6) Ο χρήστης μπορεί να επιλέξει «OK»</p> <p>7) Κλείνει το αναδυόμενο παράθυρο και εμφανίζεται η οθόνη, με το σχόλιο διαγραμμένο.</p>
Εναλλακτική ροή 1	<p>1) Έχουν ήδη εισαχθεί σχόλια στην εφαρμογή. Εμφανίζεται μια οθόνη με πεδίο εισαγωγής σχολίων και τον κέρσορα έτοιμο στο σημείο γραφής. Επιπλέον εμφανίζονται τα προηγούμενα σχόλια, με την ημερομηνία και την ώρα καταχώρησής τους. Επίσης εμφανίζεται το κουμπί «Καταχώρησε» και η μπάρα ενεργειών με το όνομα του μονοπατιού, και την επιγραφή που αναφέρει ο χρήστης βρίσκεται στην οθόνη για τα σχόλια.</p> <p>3) Ο χρήστης δεν εισάγει κάποιο σχόλιο.</p> <p>4) Εμφανίζονται στην οθόνη μια κενή γραμμή και η ημερομηνία και η ώρα που πατήθηκε το κουμπί «Καταχώρησε» .</p> <p>6) Ο χρήστης μπορεί να επιλέξει «Cancel»</p> <p>7) Κλείνει το αναδυόμενο παράθυρο και εμφανίζεται η προηγούμενη οθόνη.</p>

Πίνακας 2.8: Περίπτωση χρήσης 6

Κωδικός Περίπτωσης:	Use Case -6
Ονομασία	Επιλογή επιστροφής στο χάρτη
Εμπλεκόμενοι ρόλοι	Χρήστης
Προϋποθέσεις	Έχει επιλεγθεί το μενού στα αριστερά
Φυσιολογική ροή	<p>1) Έχει επιλεγθεί από το μενού στα αριστερά το κουμπί Home</p> <p>2) Εμφανίζεται ο χάρτης</p>

Πίνακας 2.9: Περίπτωση χρήσης 7

Κωδικός Περίπτωσης:	Use Case -7
Όνομασία	Πάτημα του κουμπιού κοινοποίησης
Εμπλεκόμενοι ρόλοι	Χρήστης
Προϋποθέσεις	Έχει επιλεγθεί το μενού στα αριστερά
Φυσιολογική ροή	1) Έχει επιλεγθεί από το μενού στα αριστερά το κουμπί κοινοποίησης 2) Εμφανίζεται αναδυόμενο παράθυρο με τις διαθέσιμες επιλογές της συσκευής για την κοινοποίηση της εφαρμογής.

Πίνακας 2.10: Περίπτωση χρήσης 8

Κωδικός Περίπτωσης:	Use Case -8
Όνομασία	Πάτημα του κουμπιού αποστολής ηλεκτρονικής αλληλογραφίας
Εμπλεκόμενοι ρόλοι	Χρήστης
Προϋποθέσεις	Έχει επιλεγθεί το μενού στα αριστερά
Φυσιολογική ροή	1) Έχει επιλεγθεί από το μενού στα αριστερά το κουμπί αποστολής ηλεκτρονικής αλληλογραφίας 2) Εμφανίζεται αναδυόμενο παράθυρο με τις διαθέσιμες επιλογές της συσκευής για την αποστολή ηλεκτρονικής αλληλογραφίας.

Πίνακας 2.11: Περίπτωση χρήσης 9

Κωδικός Περίπτωσης:	Use Case -9
Όνομασία	Πάτημα του κουμπιού εξόδου
Εμπλεκόμενοι ρόλοι	Χρήστης
Προϋποθέσεις	Έχει επιλεγθεί το μενού στα αριστερά
Φυσιολογική ροή	1) Έχει επιλεγθεί από το μενού στα αριστερά το κουμπί εξόδου 2) Γίνεται έξοδος από την εφαρμογή.

Πίνακας 2.12: Περίπτωση χρήσης 10

Κωδικός Περίπτωσης:	Use Case -10
Όνομασία	Πάτημα (Click) όταν έχει πραγματοποιηθεί το μονοπάτι
Εμπλεκόμενοι ρόλοι	Χρήστης
Φυσιολογική ροή	1) Ο χρήστης επιλέγει τα Ελληνικά στις ρυθμίσεις της συσκευής 2) Η εφαρμογή εμφανίζεται στα Ελληνικά
Εναλλακτική ροή	1) Ο χρήστης επιλέγει τα Αγγλικά στις ρυθμίσεις της συσκευής 2) Η εφαρμογή εμφανίζεται στα Αγγλικά

ΚΕΦΑΛΑΙΟ 3

ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ

Αυτό το κεφάλαιο επικεντρώνεται στην υλοποίηση της εφαρμογής και παρουσιάζονται τα αποτελέσματα του σχεδιασμού της εφαρμογής όπως αυτά αναλύθηκαν στα προηγούμενα κεφάλαια. Γίνεται αναφορά στο περιβάλλον ανάπτυξης της εφαρμογής

3.1 Εγκατάσταση Android Studio

Η εφαρμογή SantoriniTrail αναπτύχθηκε στο περιβάλλον εργασίας προγραμματισμού Android Studio. Το Android Studio είναι το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για την ανάπτυξη εφαρμογών Android, βασισμένο στο IntelliJ IDEA. Πέρα από τον ισχυρό επεξεργαστή κώδικα και τα εργαλεία προγραμματισμού, το Android Studio προσφέρει πλήθος λειτουργιών που βελτιώνουν την παραγωγικότητα των προγραμματιστών. Παραδείγματα τέτοιων λειτουργιών είναι ένας γρήγορος και πλούσιος σε χαρακτηριστικά εξομοιωτής, ένα ενιαίο περιβάλλον για ανάπτυξη εφαρμογών για όλους τους τύπους συσκευών Android, εκτεταμένα εργαλεία και πλαίσια δοκιμών, πρότυπα κώδικα και ενσωμάτωση του GitHub που βοηθά τους προγραμματιστές να εισάγουν δείγματα κώδικα για βασικές λειτουργίες εφαρμογών, κ.α..

Η εγκατάσταση του Android Studio γίνεται κατεβάζοντας και τρέχοντας το αρχείο τύπου .exe της εφαρμογής από την ιστοσελίδα <https://developer.android.com/studio> και ακολουθώντας τα βήματα του Setup Wizard.

Αφού έχει ολοκληρωθεί η εγκατάσταση και έχει ανοίξει το Android Studio, το επόμενο βήμα είναι η εγκατάσταση της εργαλειοθήκης ανάπτυξης λογισμικού (SDK) επιλέγοντας από το μενού του Android Studio : File → Settings → System Settings → Android SDK ή πατώντας το εικονίδιο του SDK Manager στην μπάρα εργαλείων. Εμφανίζονται όλα τα διαθέσιμα πακέτα της πλατφόρμας SDK σύμφωνα με τις εκδόσεις του Android και του επιπέδου της διεπαφής προγραμματισμού (API Level) , και δίνει πληροφορίες για την κατάστασή τους, δηλαδή αν κάποιο είναι εγκατεστημένο και αναβαθμισμένο ή όχι. Επιλέγοντας το πακέτο και πατώντας το κουμπί OK γίνεται η εγκατάσταση ή η αναβάθμιση του πακέτου στο Android Studio.

3.2 Αρχική Οθόνη Εφαρμογής SantoriniTrail

Η οθόνη εκκίνησης (launch/splash screen) είναι η πρώτη εμπειρία του χρήστη με την εφαρμογή [25]. Οι οθόνες εκκίνησης μπορούν να εμφανιστούν κατά την εκκίνηση μιας εφαρμογής στην αρχική οθόνη όταν φορτώνεται μια εφαρμογή, αντί να εμφανιστεί μια κενή οθόνη. Η εμφάνιση μιας οθόνης εκκίνησης μπορεί να μειώσει την αίσθηση ενός μεγάλου χρόνου φόρτωσης και έχει τη δυνατότητα να προσθέσει απόλαυση στην εμπειρία του χρήστη. Η σωστή χρήση των οθονών εκκίνησης προβλέπει ότι δεν πρέπει να εμφανίζονται εάν εκτελείται ήδη μια εφαρμογή.

Υπάρχουν δύο τύποι οθονών εκκίνησης:

- 1) Οι οθόνες εκκίνησης της διεπαφής χρήστη με αντικείμενα που “κρατούν την θέση” (placeholders) για περιεχόμενο που χρειάζεται χρόνο για να φορτωθεί. Η διεπαφή χρήστη θα εμφανίζει μια μη διαδραστική προεπισκόπηση της πραγματικής διεπαφής χρήστη της εφαρμογής. Σε αυτήν τα βασικά δομικά στοιχεία - όπως η γραμμή κατάστασης και η γραμμή εφαρμογών - εμφανίζονται χωρίς περιεχόμενο μέχρι να φορτωθεί η εφαρμογή. Τα αντικείμενα placeholder εμφανίζονται ως εικονίδια με μια διακριτική κίνηση που υποδηλώνει την φόρτωση του περιεχομένου. Αυτή η οθόνη εκκίνησης είναι κατάλληλη τόσο για εκκινήσεις εφαρμογών όσο και για μεταβάσεις δραστηριοτήτων σε μια εφαρμογή.
- 2) Οι οθόνες εκκίνησης που παρέχουν στιγμιαία έκθεση σε ένα εμπορικό σήμα. Εκφράζουν την επωνυμία ενώ φορτώνουν μια εφαρμογή, εμφανίζοντας ένα λογότυπο ή άλλα στοιχεία που βελτιώνουν την αναγνώριση της μάρκας. Πρέπει να αποφεύγεται η χρήση κειμένου σε αυτήν την οθόνη, είναι αποδεκτό μόνο ότι υπάρχει στο λογότυπό και, κατά περίπτωση, κάποιο σλόγκαν.

Για την εφαρμογή SantoriniTrail έχει δημιουργηθεί μια δραστηριότητα που ονομάζεται `SplashscreenActivity`. Εμφανίζει μια εικόνα για την οθόνη εκκίνησης όπου φαίνεται το λογότυπο της εφαρμογής (Εικόνα 3.1).



Εικόνα 3.1: Αρχική οθόνη της εφαρμογής SantoriniTrail.

Στην δραστηριότητα `SplashscreenActivity` οποιοδήποτε άγγιγμα στην οθόνη υποδηλώνει την θέληση του χρήστη να διακόψει την οθόνη εκκίνησης και να συνεχίσει στην επόμενη δραστηριότητα της εφαρμογής. Στον κώδικα της δραστηριότητας βλέπουμε ότι όταν δημιουργείται η δραστηριότητα ξεκινά ένα νέο νήμα (thread) (Εικόνα 3.2). Το νήμα «κοιμάται» για όσα `millisecond` (5000) ορίζει το `splashtime` και περιοδικά «ξυπνά» για να ελέγξει την ενεργή σημαία (`active flag`) για να δει αν πρέπει να τερματιστεί. Συγχρόνως το αρχικό νήμα της εφαρμογής συνεχίζει να τρέχει. Αυτό το κύριο νήμα ενεργοποιεί την δραστηριότητα και περιμένει σε ένα βρόγχο συμβάντος για τον χρήστη να αλληλεπιδράσει με την οθόνη εκκίνησης. Αν ο χρήστης ακουμπήσει την οθόνη πριν τελειώσει το χρονόμετρο παράγεται ένα συμβάν `onTouch`. Αυτό το συμβάν παγιδεύεται από την μέθοδο `onTouchEvent(MotionEvent)` στην κλάση `Activity` που με την σειρά της θέτει το ενεργό (`active`) `splashscreen` σε `false`. Αυτό τερματίζει τον χρονομετρητή ώστε να τερματίσει η δραστηριότητα της οθόνης εκκίνησης και να ξεκινήσει την επόμενη δραστηριότητα, την `MainMenu` που χειρίζεται την κεντρική οθόνη της εφαρμογής `SantoriniTrail`.

```
public class SplashscreenActivity extends Activity {
    protected boolean _active = true;
    protected int _splashTime = 5000;

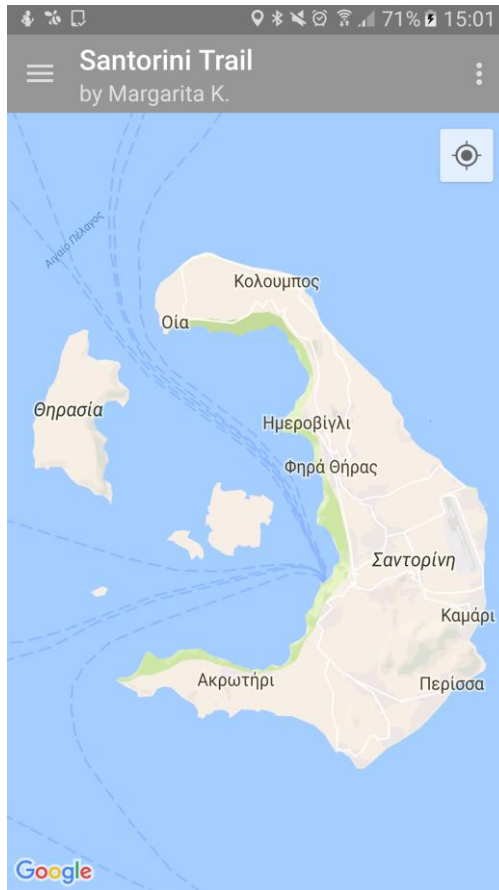
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        this.requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.activity_splashscreen);
        // thread for displaying the SplashScreen
        Thread splashTread = new Thread() {
            @Override
            public void run() {
                try {
                    int waited = 0;
                    while (_active && (waited < _splashTime)) {
                        sleep(100);
                        if (_active) {
                            waited += 100;
                        }
                    }
                } catch (InterruptedException e) {
                    // do nothing
                } finally {
                    finish();
                    startActivity(new Intent(SplashscreenActivity.this,
MainMenu.class));
                }
            }
        };
        splashTread.start();
    }

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
            _active = false;
        }
        return true;
    }
}
```

Εικόνα 3.2: Η κλάση SplashscreenActivity, της εφαρμογής SantoriniTrail.

3.3 Κεντρική Οθόνη Εφαρμογής SantoriniTrail

Η εφαρμογή έχει σχεδιαστεί για να χρησιμοποιηθεί από χρήστες που βρίσκονται στην Σαντορίνη και για αυτό το λόγο η κεντρική οθόνη της εφαρμογής είναι ο χάρτης των Google maps εστιασμένος στο νησί της Σαντορίνης. Όταν είναι διαθέσιμο το σήμα GPS στην οθόνη εμφανίζεται και το στίγμα του χρήστη, εφόσον βρίσκεται στην Σαντορίνη.



Εικόνα 3.3: Κεντρική οθόνη της εφαρμογής SantoriniTrail.

```

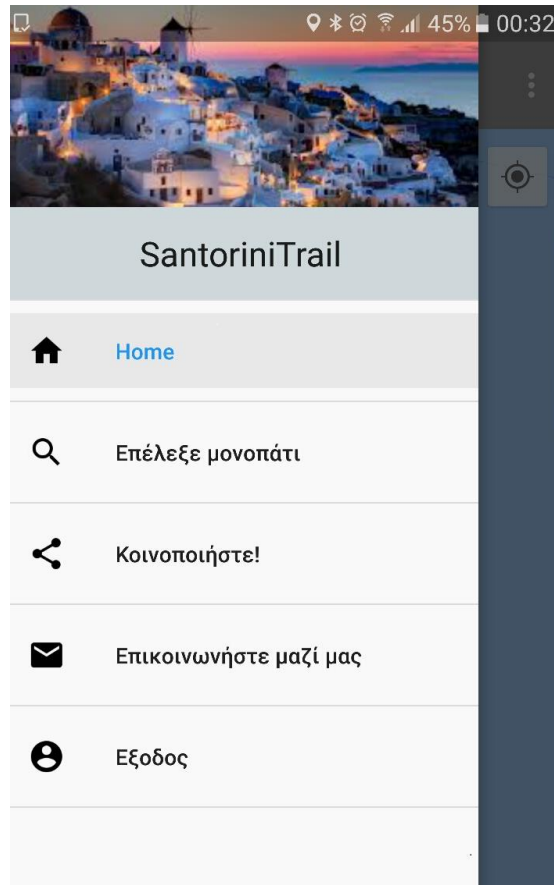
18
19 public class MainMenu extends MyBaseActivity
20 implements OnMapReadyCallback {
21
22     public dbhandler db = new dbhandler(this);
23
24     @Override
25     protected void onCreate(Bundle savedInstanceState) {
26         super.onCreate(savedInstanceState);
27         db = new dbhandler(this);
28         setContentView(R.layout.mainmenu2);
29         createMaterial(DRAWER_HOME, "Santorini Trail", "by Margarita K.");
30
31         MapFragment mapFragment = (MapFragment) getFragmentManager()
32             .findFragmentById(R.id.fragment2);
33         mapFragment.getMapAsync(this);
34     }
35
36     @Override
37     public void onMapReady(GoogleMap map) {
38         LatLng santorini = new LatLng(36.405136, 25.445741);
39
40         map.setMyLocationEnabled(true);
41         map.moveCamera(CameraUpdateFactory.newLatLngZoom(santorini, 11));
42
43
44     }
45
46
47 }
48

```

Εικόνα 3.4: Η κλάση MainMenu, της εφαρμογής SantoriniTrail.

3.4 Μενού Εφαρμογής SantoriniTrail

Ένα από τα πιο σημαντικά μέρη της διεπαφής χρήστη μιας εφαρμογής, είναι το μενού όπου γίνεται εύκολα η εναλλαγή μεταξύ των διαφόρων λειτουργιών της εφαρμογής. Στις οδηγίες του «Material design», που είναι ένα σύνολο σχεδιαστικών κανόνων (design language) που δημιουργήθηκε από τη Google, επισημαίνεται ότι όταν χρησιμοποιούνται τα σωστά εικονίδια, υπάρχει μια ροή στις έννοιες που καταλαβαίνουν διεθνώς οι χρήστες των εφαρμογών Android, μια σταθερή εμπειρία χρήστη προσθέτοντας έτσι και στην λειτουργικότητα της εφαρμογής αλλά και στην αισθητική της. Στην εφαρμογή Santorini Trail χρησιμοποιήθηκε κυλιόμενο μενού πλοήγησης (Navigation drawer), το οποίο είναι ένας πίνακας με τις κύριες επιλογές πλοήγησης της εφαρμογής να εμφανίζονται στο αριστερό άκρο της οθόνης. Ο πίνακας είναι κρυμμένος τις περισσότερες φορές, αλλά αποκαλύπτεται όταν ο χρήστης κάνει σάρωση της οθόνης από τα αριστερά προς τα δεξιά, ή όταν ο χρήστης αγγίζει το εικονίδιο με τις τρεις γραμμές στα αριστερά, στη μπάρα ενεργειών.



Εικόνα 3.5: Κυλιόμενο μενού πλοήγησης, της εφαρμογής SantoriniTrail.

Οι καρτέλες του μενού πλοήγησης είναι οι εξής:

- Αρχική οθόνη – Home
- Επέλεξε μονοπάτι – Select trail
- Κοινοποιήστε – Share
- Επικοινωνήστε μαζί μας – Contact us
- Έξοδος – Quit

Η δημιουργία του κυλιόμενου μενού πλοήγησης βασίστηκε στη βιβλιοθήκη MaterialDrawer [26] και προστέθηκαν οι λειτουργίες της κάθε καρτέλας, στην κλάση MyBaseActivity.java.

```

66 drawer = new DrawerBuilder().withActivity(this).withToolbar(toolbar).withHeader(R.layout.drawerheader).addDrawerItems(
67     new PrimaryDrawerItem().withName("Home").withIcon(R.drawable.ic_home_black_18dp).withIdentifier(DRAWER_HOME).withCheckable(false),
68     new DividerDrawerItem(),
69     new PrimaryDrawerItem().withName("Find a trail").withIcon(R.drawable.ic_search_black_18dp).withIdentifier(DRAWER_TRAILS).withCheckable(false),
70     new DividerDrawerItem(),
71     new PrimaryDrawerItem().withName("Share").withIcon(R.drawable.ic_share_black_18dp).withIdentifier(DRAWER_SHARE).withCheckable(false),
72     new DividerDrawerItem(),
73     new PrimaryDrawerItem().withName("Contact us").withIcon(R.drawable.ic_email_black_18dp).withIdentifier(DRAWER_CONTACTUS).withCheckable(false),
74     new DividerDrawerItem(),
75     new PrimaryDrawerItem().withName("Quit").withIcon(R.drawable.ic_account_circle_black_18dp).withIdentifier(DRAWER_USERPROFILE).withCheckable(false),
76     new DividerDrawerItem()
77 ).withOnDrawerItemClickListener((parent, view, position, id, drawerItem) -> {
80     if (drawerItem != null) {
81         switch (drawerItem.getIdentifier()) {
82             case 1:
83                 openhome();
84                 break;
85             case 2:
86                 openfind();
87                 break;
88             case 3:
89                 share();
90                 break;
91             case 4:
92                 opencontactus();
93                 break;
94             case 5:
95                 Exit();
96                 break;
97         }
98     }
99     return false;

```

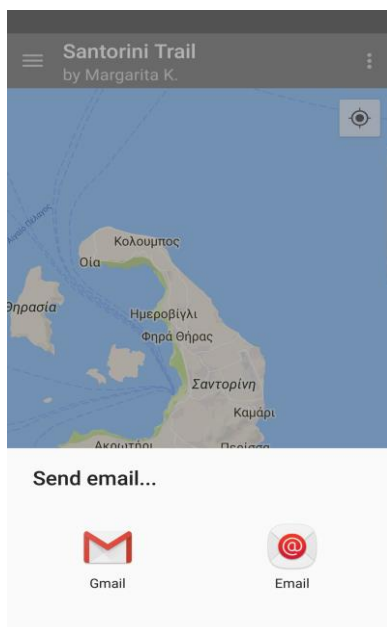
Εικόνα 3.6: Τμήμα του κώδικα που διαχειρίζεται τις λειτουργίες των καρτελών του κυλιόμενο μενού πλοήγησης ,της εφαρμογής SantoriniTrail.

3.5 Επιλογές του Μενού της Εφαρμογής SantoriniTrail

Σε αυτό το σημείο έχει ενδιαφέρον να γίνει μια αναφορά στα αντικείμενα των προθέσεων (Intents). Ένα από τα σπουδαία χαρακτηριστικά των εξαρτημάτων των εφαρμογών Android είναι η δυνατότητά τους να συνδέονται με άλλες εφαρμογές Android. Αυτή η διασύνδεση βασίζεται στις ενέργειες των αντικειμένων πρόθεσης, οι οποίες στέλνουν ασύγχρονα μηνύματα και ανακοινώνουν την πρόθεσή τους να κάνουν μια ενέργεια. Το κατάλληλο εξάρτημα εφαρμογής Android που έχει την δυνατότητα να διεκπεραιώσει την ενέργεια το αναλαμβάνει. Οι προθέσεις διακρίνονται σε υπονοούμενες (Implicit) και σαφείς (Explicit). Στις υπονοούμενες προθέσεις δεν διευκρινίζεται το εξάρτημα που διεκπεραιώνει την ενέργεια και πρέπει το μήνυμα που αποστέλλεται να περιλαμβάνει επαρκείς πληροφορίες για το σύστημα ώστε αυτό να καθορίσει ποιο από τα διαθέσιμα εξαρτήματα είναι καλύτερο να τρέξει για αυτό το σκοπό. Έτσι για παράδειγμα στην εφαρμογή Santorini Trail, δεν έχει ενσωματωθεί κάποια κλάση για τη αποστολή της ηλεκτρονικής αλληλογραφίας αλλά η εφαρμογή

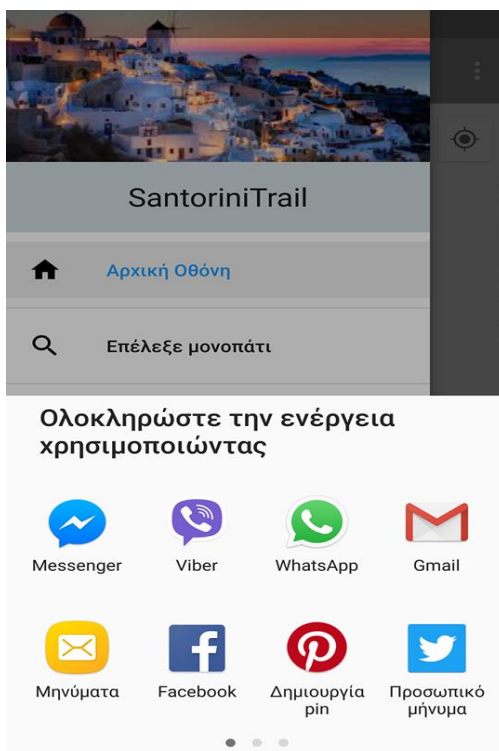
Ανάπτυξη εφαρμογής περιήγησης σε μονοπάτια «SANTORINI TRAIL» για Android

χρησιμοποιεί αντικείμενα προθέσεων. Ανακοινώνει την πρόθεση για αποστολή ηλεκτρονικής αλληλογραφίας και τα εξαρτήματα που είναι εγκατεστημένα στην κινητή συσκευή και μπορούν να το χειριστούν το αναλαμβάνουν. Ο χρήστης στο παράδειγμα της εικόνας (Εικόνα 3.7) μπορεί να επιλέξει ανάμεσα στις δύο εφαρμογές Gmail και Email.



Εικόνα 3.7: Οθόνη αποστολής αλληλογραφίας της εφαρμογής SantoriniTrail.

Παρομοίως γίνεται για την κοινοποίηση (Share) (Εικόνα 3.8).



Εικόνα 3.8: Οθόνη κοινοποίησης (Share) της εφαρμογής SantoriniTrail.

Αντιθέτως στις σαφείς προθέσεις υπάρχει κάποιο συγκεκριμένο εξάρτημα που πρέπει να διεκπεραιώσει την ζητούμενη ενέργεια και διατυπώνεται η ακριβής κλάση που τρέχει. Οι προθέσεις αυτές συνήθως δεν περιέχουν άλλες πληροφορίες, απλά είναι ένας τρόπος ώστε η εφαρμογή να ξεκινήσει κάποιες εσωτερικές εργασίες καθώς ο χρήστης αλληλεπιδρά με την εφαρμογή. Παρατηρώντας τον σχετικό κώδικα της εφαρμογής Santorini Trail (Εικόνα 3.9) μπορεί να κατανοηθεί αυτή η διαφορά. Οι σαφείς προθέσεις στην εφαρμογή Santorini Trail χρησιμοποιούνται για να τρέξουν κλάσεις της εφαρμογής: Για παράδειγμα η κλάση MainMenu χρησιμοποιείται όταν ο χρήστης θέλει να επιστρέψει στην αρχική οθόνη και η κλάση ListTrailsActivity όταν ο χρήστης θέλει να δει τα διαθέσιμα μονοπάτια.

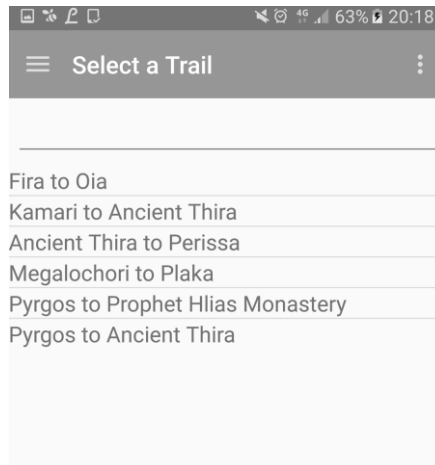


```
23 public class MyBaseActivity extends ActionBarActivity {
103 }
104 protected void openfind() {
105     if (selectItem != DRAWER_TRAILS) {
106         Intent intent = new Intent(this, ListTrailsActivity.class);
107         startActivity(intent);
108     }
109 }
110 protected void openhome() {
111     if (selectItem != DRAWER_HOME) {
112         Intent intent = new Intent(this, MainMenu.class);
113         startActivity(intent);
114     }
115 }
116 protected void share() {
117     Intent sendIntent = new Intent();
118     sendIntent.setAction(Intent.ACTION_SEND);
119     sendIntent.putExtra(Intent.EXTRA_TEXT, "Check out this app: www.santorinitrail.simplesite.com.");
120     sendIntent.setType("text/plain");
121     startActivity(sendIntent);
122 }
123 protected void openprofile() {
124 }
125 protected void opencontactus() {
126     Intent emailIntent = new Intent(Intent.ACTION_SENDTO, Uri.fromParts(
127         "mailto", "santorinitrail@gmail.com", null));
128     emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Subject");
129     startActivity(Intent.createChooser(emailIntent, "Send email..."));
130 }
131 }
132 }
```

Εικόνα 3.9: Τμήμα του κώδικα που διαχειρίζεται τις προθέσεις της εφαρμογής SantoriniTrail.

3.6 Οθόνη Επιλογής Μονοπατιού Εφαρμογής SantoriniTrail

Η παρουσίαση των διαθέσιμων μονοπατιών στην εφαρμογή γίνεται σε μια οθόνη όπου τα μονοπάτια βρίσκονται σε λίστα. Ο χρήστης μπορεί να επιλέξει το μονοπάτι, σύμφωνα με το όνομα που περιγράφει την διαδρομή.



Εικόνα 3.10: Οθόνη όπου παρουσιάζονται σε λίστα τα διαθέσιμα μονοπάτια.

Η κλάση ListTrailsActivity (Εικονες 3.11 και 3.12) διαχειρίζεται δεδομένα από την βάση δεδομένων, τα εμφανίζει στην λίστα και διαχειρίζεται τι συμβαίνει όταν ο χρήστης επιλέγει ένα από τα μονοπάτια.

```
ListTrailsActivity.java x
ListTrailsActivity
1 package com.teipir.margarita.santorinitrail;
2
3 import android.content.Intent;
4 import android.os.Bundle;
5 import android.text.Editable;
6 import android.text.TextWatcher;
7 import android.view.View;
8 import android.widget.AdapterView;
9 import android.widget.EditText;
10 import android.widget.ListView;
11
12 import com.teipir.margarita.santorinitrail.adapters.TrailAdapter;
13 import com.teipir.margarita.santorinitrail.db.dbhandler;
14 import com.teipir.margarita.santorinitrail.db.trail;
15
16 import java.util.List;
17
18 public class ListTrailsActivity extends MyBaseActivity {
19     public dbhandler db = new dbhandler(this);
20     List<trail> trailList;
21     TrailAdapter trailAdapter;
22     ListView listView;
23
24     @Override
25     protected void onCreate(Bundle savedInstanceState) {
26         super.onCreate(savedInstanceState);
27         setContentView(R.layout.activity_list_trails);
28         loadtrails();
29         createMaterial(DRAWER_TRAILS, "Select a Trail", "");
30         EditText filterText = (EditText) findViewById(R.id.filter);
31         filterText.addTextChangedListener(filtertextwatcher);
32     }
}
```

Εικόνα 3.11: Τμήμα του κώδικα που διαχειρίζεται την εμφάνιση των μονοπατιών σε λίστα.

```

ListTrailsActivity.java x
33
34 private void loadtrails() {
35     trailList = db.getAlltrails("");
36     trailAdapter = new TrailAdapter(this, R.layout.adapter_itemtrail, trailList);
37     listView = (ListView) findViewById(R.id.ListTrailView);
38     listView.setAdapter(trailAdapter);
39     listView.setClickable(true);
40     listView.setOnItemClickListener(
41         (parent, view, position, id) -> {
42             trail t = (trail) listView.getItemAtPosition(position);
43             Intent intent = new Intent(ListTrailsActivity.this, PathDetailsActivity.class);
44             intent.putExtra("id", t.get_id());
45             startActivity(intent);
46         });
47
48
49
50
51
52 }
53
54 private TextWatcher filtertextwatcher = new TextWatcher() {
55     public void afterTextChanged(Editable s) {
56     }
57
58     public void beforeTextChanged(CharSequence s, int start, int count,
59         int after) {
60     }
61
62     public void onTextChanged(CharSequence s, int start, int before,
63         int count) {
64         trailAdapter.getFilter().filter(s);
65     }
66 }
67 };
68 }
69

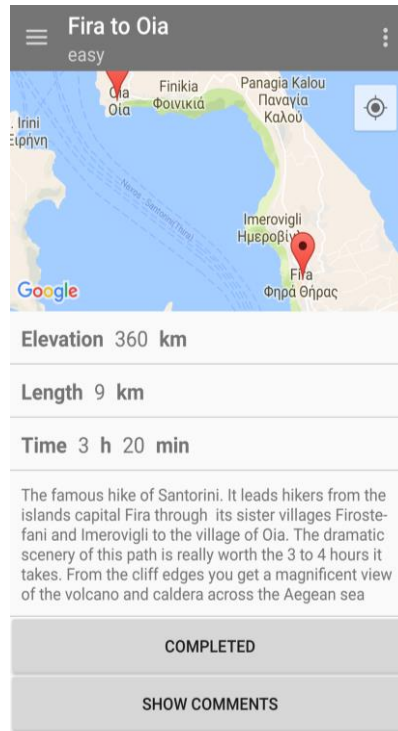
```

Εικόνα 3.12: Συνέχεια του κώδικα που διαχειρίζεται την εμφάνιση των μονοπατιών σε λίστα.

3.7 Οθόνη περιγραφής μονοπατιού εφαρμογής SantoriniTrail

Η παρουσίαση της περιγραφής ενός επιλεγμένου μονοπατιού γίνεται σε μια οθόνη όπου εστιάζεται στο τμήμα του χάρτη της Σαντορίνης όπου βρίσκεται το μονοπάτι (Εικόνα 3.13). Υπάρχουν δυο κόκκινοι δείκτες που σηματοδοτούν το σημείο εκκίνησης και το σημείο όπου τελειώνει η διαδρομή. Κάτω από τον χάρτη δίνονται πληροφορίες για το μονοπάτι όπως το μήκος της διαδρομής και ο εκτιμώμενος χρόνος διεκπεραίωσης. Επίσης υπάρχει η περιγραφή του μονοπατιού που μπορεί να διαβάσει ο χρήστης αγγίζοντας την οθόνη και κάνοντας κύλιση προς τα πάνω. Από κάτω υπάρχουν δυο κουμπιά τα οποία μπορεί να επιλέξει ο χρήστης για να εισάγει και να διαβάσει της σημειώσεις του σχετικά με το μονοπάτι.

Ανάπτυξη εφαρμογής περιήγησης σε μονοπάτια «SANTORINI TRAIL» για Android



Εικόνα 3.13: Οθόνη όπου παρουσιάζεται η περιγραφή του μονοπατιού.

Κεφάλαιο 4

Οδηγίες Χρήσης

Στο κεφάλαιο αυτό γίνεται ανάλυση του τρόπου χρήσης της εφαρμογής σε κάθε διαθέσιμο στιγμιότυπό της (οθόνη). Περιγράφεται και εξηγείται η λειτουργία της και αναλύεται σε βήματα η κάθε κίνηση που μπορεί να κάνει ο χρήστης. Επίσης γίνεται επεξήγηση του τρόπου εγκατάστασης της εφαρμογής σε κινητό ή ταμπλέτα Android.

4.1 Εγκατάσταση της εφαρμογής σε συσκευή Android

Το ζητούμενο είναι να μεταφερθούν τα αρχεία της εφαρμογής SantoriniTrail με την επέκταση .apk από το Android Studio στον υπολογιστή στο κινητό. Τα αρχεία με την επέκταση .apk αντιπροσωπεύουν το πακέτο Android Package (APK) που χρησιμοποιείται από το λειτουργικό σύστημα Android για τη διανομή και εγκατάσταση εφαρμογών για κινητά.

Η εγκατάσταση μιας εμπορικής εφαρμογής είναι μια σχετικά εύκολη διαδικασία. Ο κύριος τρόπος που εγκαθίσταται μια εφαρμογή στο Android είναι μέσω της εφαρμογής Play Store που βρίσκεται προ-εγκατεστημένη στα περισσότερα κινητά και ταμπλέτες Android. Με μια απλή αναζήτηση γίνεται η εύρεση της ζητούμενης εφαρμογής και κατόπιν εμφανίζεται μια ενημερωτική οθόνη σχετικά με την εφαρμογή και το κουμπί της εγκατάστασης που αναλαμβάνει αυτόματα την εγκατάσταση της εφαρμογής.



Εικόνα 4.1 Εγκατάσταση εμπορικής εφαρμογής μέσω του Play Store της Google σε συσκευή Android

Επειδή η αναζήτηση για εφαρμογές σε συσκευές που χρησιμοποιούν πληκτρολόγιο αφής δεν είναι ο πιο βολικός τρόπος διαθέσιμος, υπάρχει και εναλλακτική λύση όπου η εγκατάσταση της εφαρμογής γίνεται από τον υπολογιστή, μέσω του ιστότοπου του Google Play στο play.google.com. Εφόσον γίνεται σύνδεση με τον ίδιο λογαριασμό Google που χρησιμοποιείται για το τηλέφωνο ή ταμπλέτα Android, με ένα κλικ στο κουμπί εγκατάστασης στον ιστότοπο εγκαθίσταται αυτόματα η εφαρμογή στη συσκευή. Αν είναι συνδεδεμένες πολλές συσκευές Android, στον ίδιο λογαριασμό της Google γίνεται επιλογή της συσκευής στην οποία θα εγκατασταθεί η εφαρμογή. Αφού ενημερωθεί το Google Play για την εγκατάσταση της εφαρμογής, το τηλέφωνο ή η ταμπλέτα θα ξεκινήσει τη λήψη της εφαρμογής και θα την εγκαταστήσει αυτόματα.

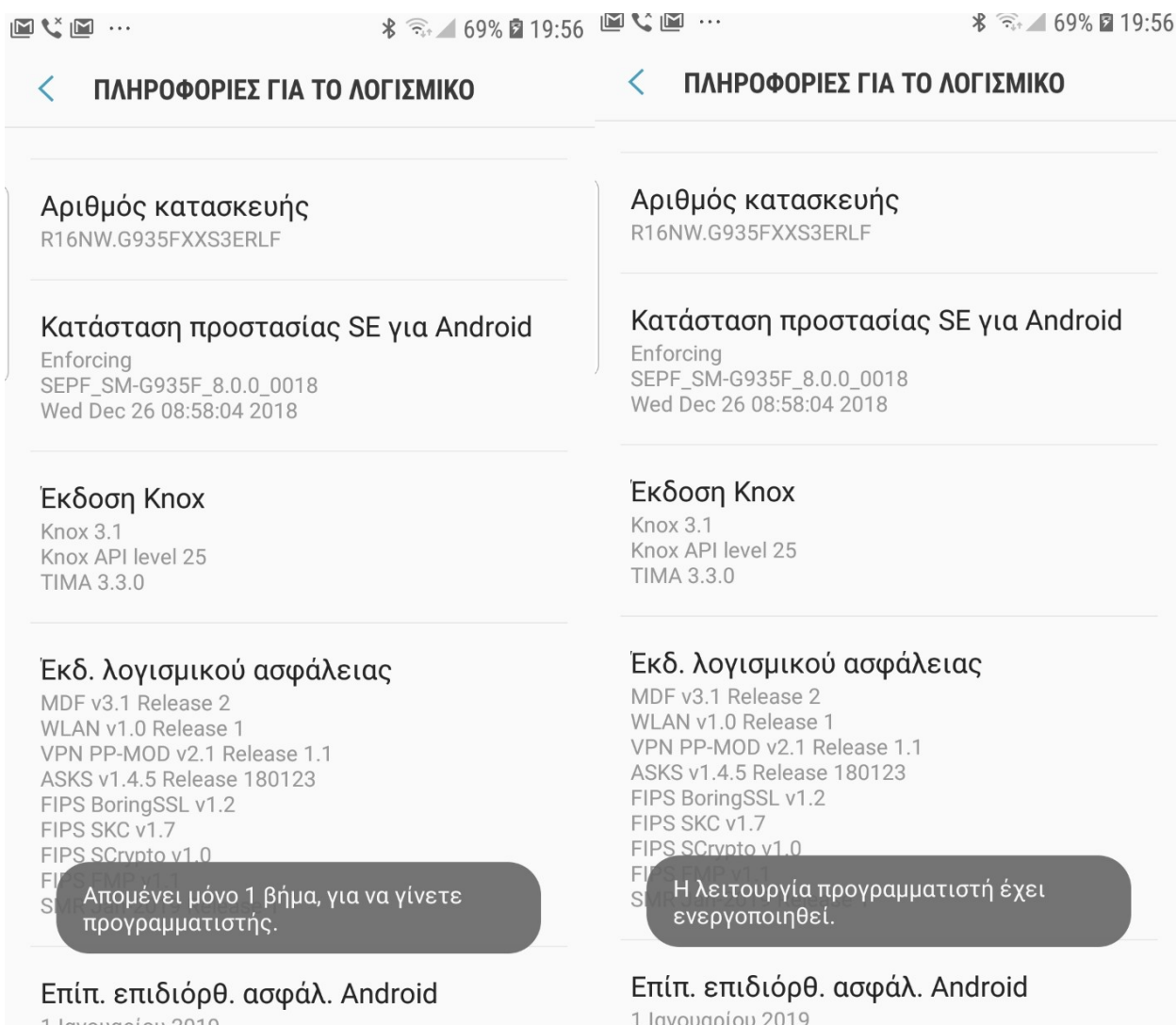
Όμως σε κάποιες περιπτώσεις δεν μπορεί να γίνει η χρήση του Google Play, όπως για ορισμένες- ιδιαίτερα οικονομικές - συσκευές Android που προέρχονται απευθείας από κατασκευαστές στην Κίνα, που δεν πιστοποιούνται από την Google και δεν περιλαμβάνουν το Google Play. Αυτό συμβαίνει και στην περίπτωση εφαρμογών που δεν έχουν σύμβαση με το Google Play. Το Android υποστηρίζει τη λειτουργία του sideloading, η οποία επιτρέπει την πλάγια εγκατάσταση εφαρμογών εκτός του Google Play (Εικόνα 4.2). Ωστόσο, αυτό είναι απενεργοποιημένο από προεπιλογή στις συσκευές Android για λόγους ασφαλείας. Η ενεργοποίηση της λειτουργίας, γίνεται από την εφαρμογή Ρυθμίσεις στο Android, επιλέγοντας την κατηγορία Οθόνη Κλειδώματος και Ασφάλεια και ενεργοποιώντας το πλαίσιο εγκατάσταση αγνώστων εφαρμογών, για τον φάκελο που διαχειρίζεται τα αρχεία της συσκευής.



Εικόνα 4.2 Στάδια ενεργοποίησης, πλάγιας εγκατάστασης εφαρμογών.

Μετά την ενεργοποίηση αυτής της ρύθμισης, μπορεί να γίνει η λήψη μιας εφαρμογής Android σε μορφή .apk και να εγκατασταθεί στη συσκευή. Για παράδειγμα, μπορεί να κατέβει το αρχείο .apk στο πρόγραμμα περιήγησης του Android και να ανοιχτεί από την εφαρμογή Λήψεις. Επίσης μπορεί να κατέβει το αρχείο στον υπολογιστή, να αντιγραφεί στο σύστημα αρχείων του Android με ένα καλώδιο USB, να χρησιμοποιηθεί μια εφαρμογή διαχείρισης αρχείων για την εύρεση της τοποθεσίας του αρχείου στην συσκευή και να πατηθεί το αρχείο .apk για να ξεκινήσει η εγκατάσταση.

Συγκεκριμένα για τους προγραμματιστές εφαρμογών Android , για τους οποίους είναι σημαντικό να δοκιμάζουν πάντα την εφαρμογή σε μια πραγματική συσκευή πριν την απελευθερώσουν στους χρήστες, μπορούν να εγκαταστήσουν και να κάνουν δοκιμές στην εφαρμογή μέσω του Android Studio. Πρώτα χρειάζεται να ρυθμιστούν οι επιλογές προγραμματιστή στο κινητό ή στην ταμπλέτα που θα χρησιμοποιηθεί. Από τις Ρυθμίσεις της συσκευής, και για συσκευές Android 8.0 ή νεότερες επιλέγεται το «Πληροφορίες για το τηλέφωνο», και μετά το «πληροφορίες λογισμικού». Πραγματοποιείται κύλιση προς τα κάτω και στο πεδίο που αναγράφεται το «αριθμός κατασκευής», ο χρήστης κάνει κλικ επτά διαδοχικές φορές. Η συσκευή αμέσως μετά ειδοποιεί ότι έχουν ενεργοποιηθεί τα δικαιώματα προγραμματιστή.

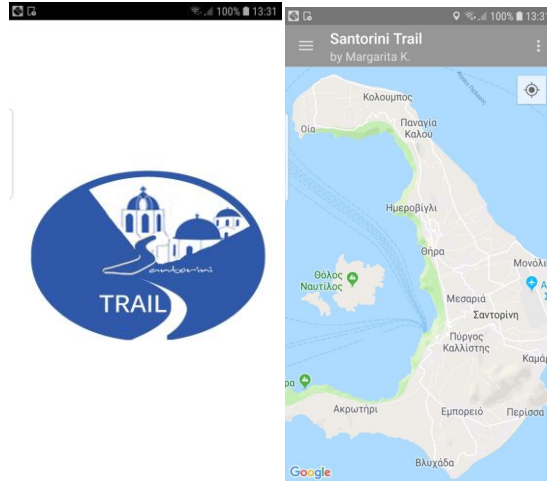


Εικόνα 4.3 Βήματα ενεργοποίησης λειτουργίας προγραμματιστή.

Γίνεται επιστροφή στην προηγούμενη οθόνη για να βρεθούν οι επιλογές προγραμματιστή στο κάτω μέρος της οθόνης. Στη συνέχεια, πρέπει να γίνει κύλιση της οθόνης λίγο προς τα κάτω και να ενεργοποιηθεί ο εντοπισμός σφαλμάτων USB. Αυτό επιτρέπει στο Android Studio και σε άλλα εργαλεία SDK να αναγνωρίζουν τη συσκευή που χρησιμοποιείται όταν συνδέεται μέσω USB, ώστε να μπορούν να χρησιμοποιηθούν το εργαλείο εντοπισμού σφαλμάτων και άλλα εργαλεία.

4.2 Χρησιμοποιώντας την εφαρμογή στο κινητό ή στην ταμπλέτα

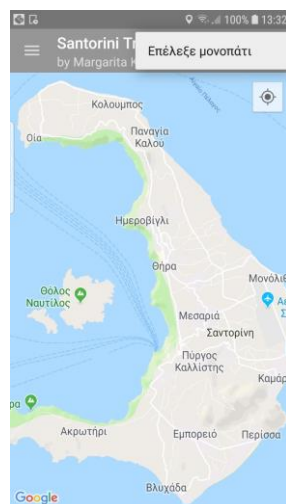
Ξεκινώντας ο χρήστης κάνει επιλογή του εικονιδίου της εφαρμογής SantoriniTrail και εμφανίζεται η αρχική οθόνη με το λογότυπο της εφαρμογής. Μετά εμφανίζεται ο χάρτης της Σαντορίνης. Αν ο χρήστης βρίσκεται στη Σαντορίνη και έχει ανοιχτό το GPS τότε εμφανίζεται μια μπλέ κουκίδα με το στίγμα του πάνω στον χάρτη.



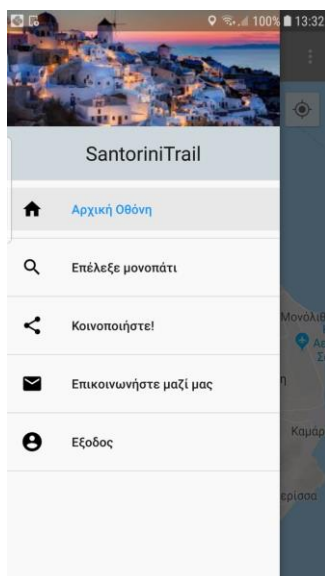
Εικόνα 4.4 Εμφάνιση του λογότυπου και του χάρτη της Σαντορίνης σε επόμενη οθόνη.

Η επιλογή του μονοπατιού μπορεί να γίνει με 2 τρόπους:

- Ο χρήστης πατάει στην επάνω δεξιά γωνία της εφαρμογής όπου βρίσκεται η συντόμευση με τις 3 τελείες και εμφανίζεται η επιλογή «Επέλεξε μονοπάτι» (Εικόνα 4.5).
- Ο χρήστης πατάει στην επάνω αριστερή γωνία με τις 3 οριζόντιες γραμμές και εμφανίζεται το μενού της εφαρμογής. Εμφανίζονται διάφορες επιλογές μεταξύ των οποίων και η επιλογή «Επέλεξε μονοπάτι» (Εικόνα 4.6).



Εικόνα 4.5 Η πρώτη μέθοδος επιλογής του μονοπατιού



Εικόνα 4.6 Η εναλλακτική μέθοδος επιλογής του μονοπατιού

Εμφανίζεται μια λίστα με τα μονοπάτια της Σαντορίνης και ο χρήστης μπορεί να επιλέξει μονοπάτι κάνοντας κλικ σε αυτό που τον ενδιαφέρει.



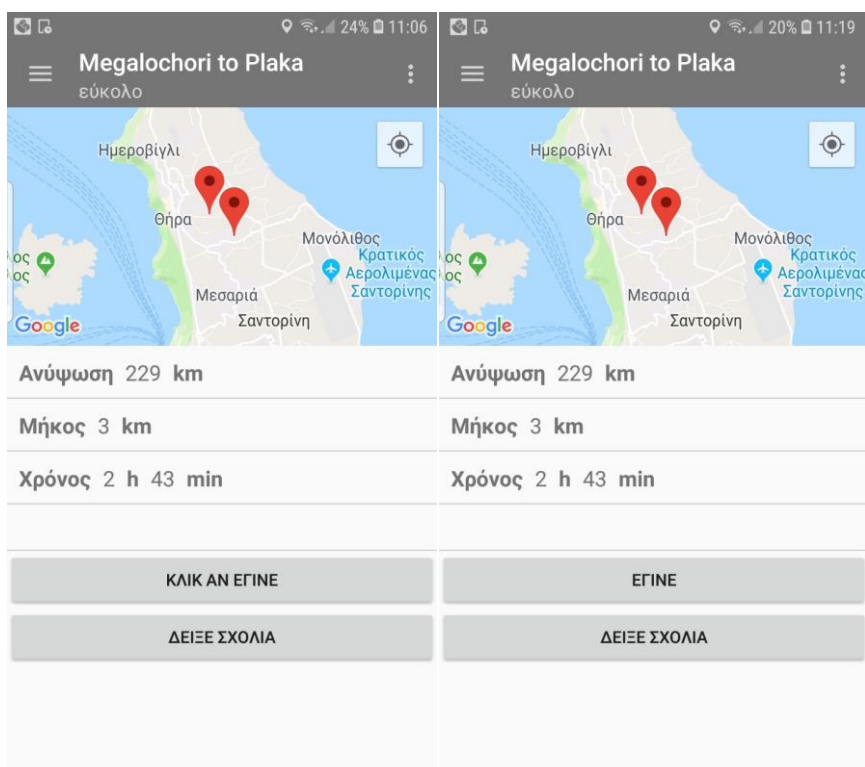
Εικόνα 4.7 Βήμα 2^ο: Εμφάνιση των διαθέσιμων μονοπατιών σε λίστα

Μόλις επιλεγθεί το μονοπάτι γίνεται η προβολή νέου τμήματος του χάρτη εστιασμένο με μεγέθυνση στην περιοχή της Σαντορίνης όπου βρίσκεται το μονοπάτι και σηματοδοτείται η αρχή και το τέλος της διαδρομής. Επίσης εμφανίζονται πληροφορίες για το συνολικό θετικό υψομετρικό της διαδρομής, το μήκος της διαδρομής, ο εκτιμώμενος χρόνος ολοκλήρωσης της διαδρομής, και μια περιγραφή της διαδρομής. Στην μπάρα εργαλείων εμφανίζεται το όνομα της διαδρομής και από κάτω χαρακτηρισμός για τον βαθμό δυσκολίας της (εύκολο, μέτριο, δύσκολο)



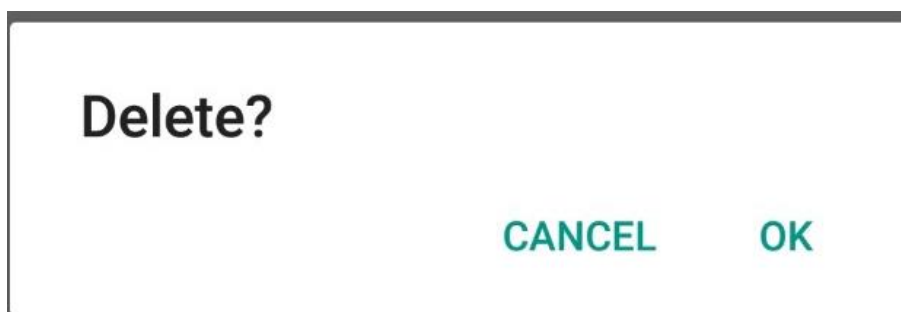
Εικόνα 4.8 Βήμα 3^ο: Περιγραφή του επιλεγμένου μονοπατιού

Στο κάτω μέρος της οθόνης, κάτω από την περιγραφή του μονοπατιού, υπάρχει η επιλογή «κλικ αν έγινε». Όταν πατηθεί εμφανίζεται στον χρήστη ένα ημερολόγιο για την επιλογή της ημερομηνίας πραγματοποίησης του μονοπατιού και πεδίο όπου μπορεί να συμπληρώσει την ώρα πραγματοποίησής του. Αφού έχουν εισαχθεί τα δεδομένα και με την επιλογή του κουμπιού OK επιστρέφει στην προηγούμενη οθόνη όπου έχει αλλάξει η λεζάντα του κουμπιού που πατήθηκε και δηλώνει ότι πραγματοποιήθηκε το μονοπάτι. Επίσης, υπάρχει η εναλλακτική επιλογή του κουμπιού ακύρωσης, αν ο χρήστης δεν θέλει να εισαχθούν τα δεδομένα και γίνεται επιστροφή στην προηγούμενη οθόνη όπου η λεζάντα έχει παραμείνει η ίδια.



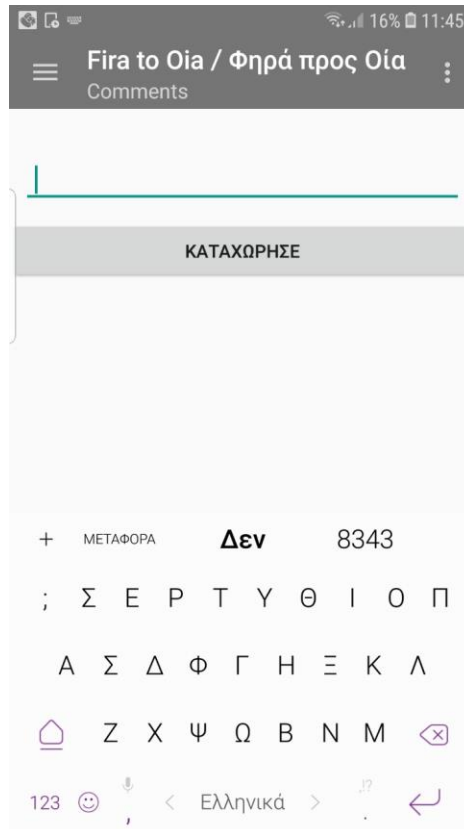
Εικόνα 4.9 Το κουμπί «κλικ αν έγινε» πριν και μετά την επιλογή του

Αν ο χρήστης θέλει να διαγράψει τα δεδομένα που έχει εισάγει μπορεί να κάνει κλικ πάνω στο κουμπί «έγινε» και θα εμφανιστεί ένα πλαίσιο διαλόγου που γράφει «Διαγραφή;» και μπορεί να επιλέξει ΟΚ ή ακύρωση.



Εικόνα 4.10 Πλαίσιο διαλόγου διαγραφής

Το επόμενο βήμα είναι να πατηθεί το κουμπί «Εμφάνιση Σχολίων». Εμφανίζεται μια οθόνη με πεδίο εισαγωγής σχολίων και τον κέρσορα έτοιμο στο σημείο γραφής (Εικόνα 4.11). Επίσης εμφανίζεται το κουμπί «Καταχώρησε» και η μπάρα ενεργειών με το όνομα του μονοπατιού και από κάτω γράφει «Σχόλια». Ο χρήστης εισάγει τα σχόλιά του για την διαδρομή, πατάει το κουμπί «Καταχώρησε» και εμφανίζονται στην οθόνη τα σχόλια του η ημερομηνία και η ώρα εισαγωγής των σχολίων.

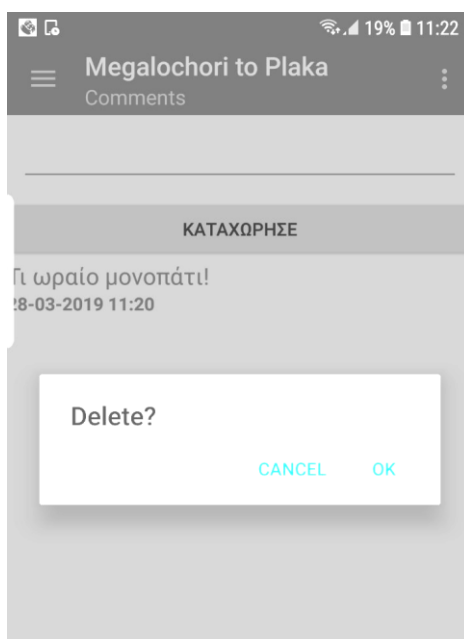


Εικόνα 4.11 Ο κέρσορας σε αναμονή για τα σχόλια του χρήστη



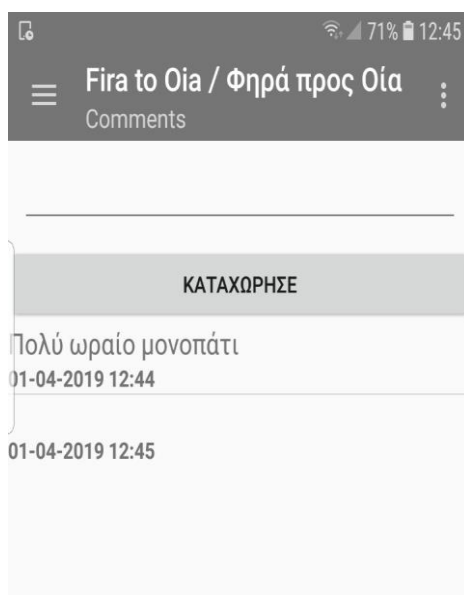
Εικόνα 4.12 Οθόνη με σχόλια για το μονοπάτι

Με παρατεταμένο πάτημα στο σχόλιο, εμφανίζεται αναδυόμενο παράθυρο, με την ερώτηση διαγραφής του σχολίου. Ο χρήστης μπορεί να επιλέξει «ΟΚ» η Ακύρωση. Με την ακύρωση γίνεται επιστροφή στην προηγούμενη οθόνη ενώ με το πάτημα του ΟΚ κλείνει το αναδυόμενο παράθυρο και εμφανίζεται η οθόνη, με το σχόλιο διαγραμμένο.



Εικόνα 4.13 Διαγραφή Σχολίου

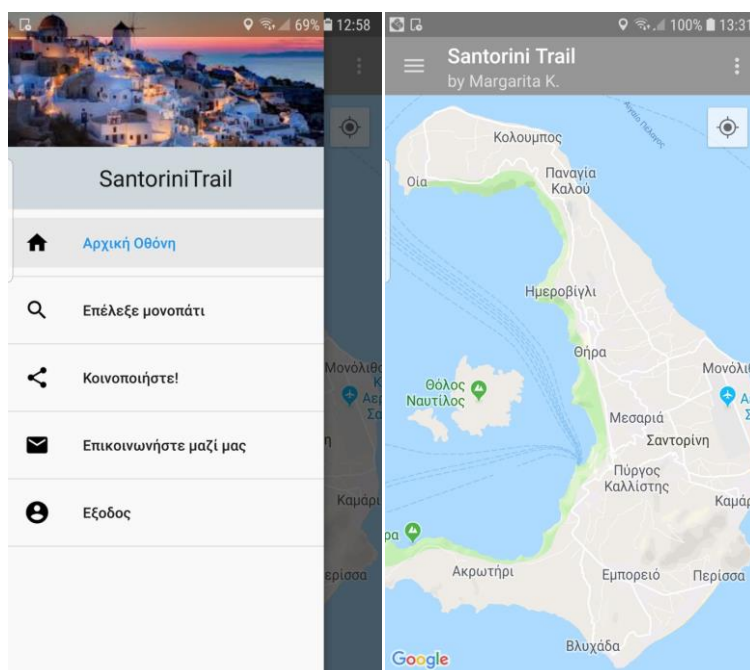
Εναλλακτικά αν έχουν ήδη εισαχθεί σχόλια στην εφαρμογή, εμφανίζεται μια οθόνη με πεδίο εισαγωγής σχολίων και τον κέρσορα έτοιμο στο σημείο γραφής. Επιπλέον εμφανίζονται τα προηγούμενα σχόλια, με την ημερομηνία και την ώρα καταχώρησης τους. Επίσης εμφανίζεται το κουμπί «Καταχώρησε» και η μπάρα ενεργειών με το όνομα του μονοπατιού, και την επιγραφή που αναφέρει ο χρήστης βρίσκεται στην οθόνη για τα σχόλια. Αν ο χρήστης δεν εισάγει κάποιο σχόλιο εμφανίζονται στην οθόνη μια κενή γραμμή και η ημερομηνία και η ώρα που πατήθηκε το κουμπί «Καταχώρησε» .



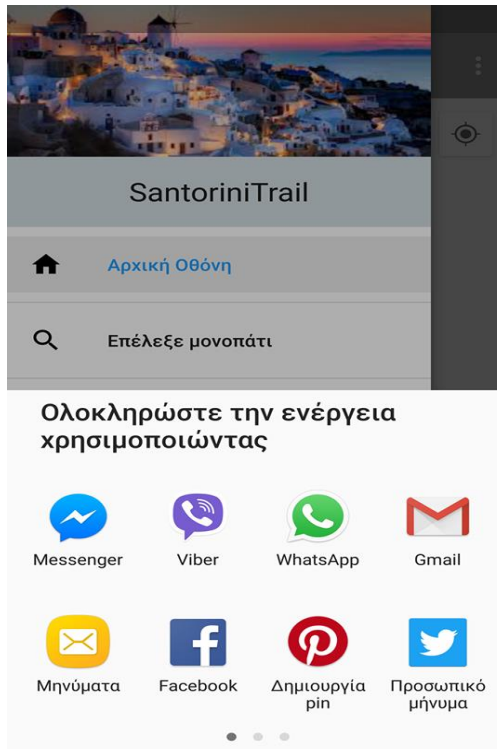
Εικόνα 4.14. Εισαγωγή επιπλέον σχολίων και κενό σχόλιο

Στο κυλιόμενο μενού είναι διαθέσιμες και οι εξής λειτουργίες (Εικόνες 4.15, 4.16, 4.17):

- 1) Η επιλογή επιστροφής στον χάρτη κάνοντας κλικ στο πλαίσιο Αρχική Οθόνη.
- 2) Κοινοποίηση της εφαρμογής, με κλικ στο πλαίσιο που γράφει «Κοινοποιήστε!» εμφανίζεται αναδυόμενο παράθυρο με τις διαθέσιμες επιλογές της συσκευής για την κοινοποίηση της εφαρμογής.
- 3) Αποστολή αλληλογραφίας στην δημιουργό της εφαρμογής. Έχει επιλεγθεί το κουμπί αποστολής ηλεκτρονικής αλληλογραφίας και εμφανίζεται αναδυόμενο παράθυρο με τις διαθέσιμες επιλογές της συσκευής για την αποστολής ηλεκτρονικής αλληλογραφίας.
- 4) Έξοδος από την εφαρμογή. Με πάτημα στο πλαίσιο που γράφει έξοδος μπορεί να γίνει έξοδος από την εφαρμογή

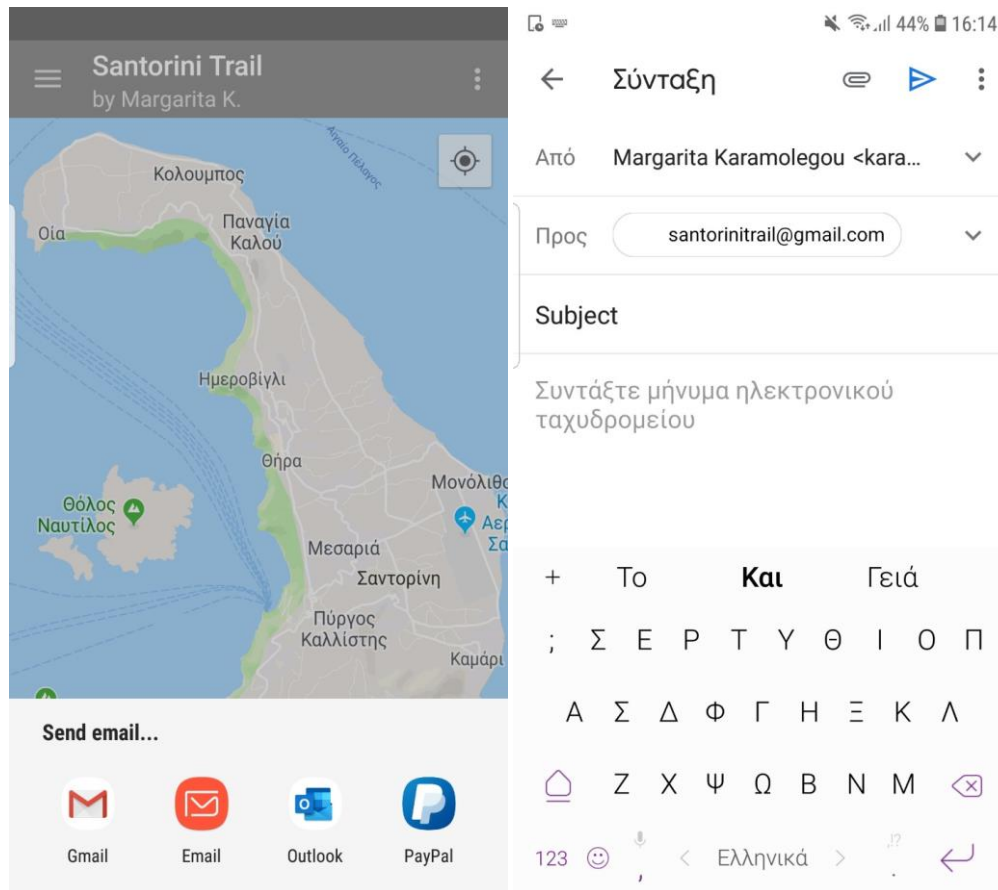


Εικόνα 4.15 Μενού και διαθέσιμες επιλογές. Μετά από πάτημα στο πλαίσιο «Αρχική Οθόνη» εμφανίζεται ο χάρτης



Εικόνα 4.16 Μετά από την επιλογή του κουμπιού κοινοποίησης εμφανίζονται οι διαθέσιμοι τρόποι κοινοποίησης

Ανάπτυξη εφαρμογής περιήγησης σε μονοπάτια «SANTORINI TRAIL» για Android



Εικόνα 4.17 Μετά από την επιλογή του πλαισίου «Επικοινωνήστε μαζί μας» εμφανίζονται οι διαθέσιμοι τρόποι αποστολής αλληλογραφίας. Η διεύθυνση ταχυδρομείου για την εφαρμογή Santorinitrail έχει συμπληρωθεί αυτόματα μετά από το «Προς»

ΚΕΦΑΛΑΙΟ 5

Δοκιμή Λογισμικού

Αυτό το κεφάλαιο επικεντρώνεται στη δοκιμή της εφαρμογής. Πρώτον, γίνεται αναφορά σε τυποποιημένα βήματα δοκιμής μιας εφαρμογής. Επίσης γίνεται αναφορά στις δοκιμές μονάδας (unit tests), οι οποίες δοκιμάζουν τα μικρότερα μέρη του κώδικα για να βεβαιωθεί ότι συμπεριφέρονται όπως αναμένεται. Στη συνέχεια, επισημαίνονται οι δοκιμές ευχρηστίας (usability tests), οι οποίες συχνά αποκαλύπτουν προβλήματα στην εφαρμογή καθώς παρακολουθούν την αλληλεπίδραση μεταξύ του χρήστη και της εφαρμογής.

5.1 Διαδικασίες δοκιμής

Οι διαδικασίες δοκιμής βοηθούν τους σχεδιαστές εφαρμογών να ανακαλύψουν διάφορους τύπους προβλημάτων ποιότητας στην εφαρμογή τους. Παρακάτω παρατίθενται τα βήματα της Google για την δοκιμή εφαρμογών [27].

- Περιήγηση σε όλα τα μέρη της εφαρμογής - όλες τις οθόνες, τα παράθυρα διαλόγου, τις ρυθμίσεις και όλες τις ροές χρηστών. Αν η εφαρμογή επιτρέπει επεξεργασία ή δημιουργία περιεχομένου, αναπαραγωγή παιχνιδιού ή αναπαραγωγή πολυμέσων, εισάγονται αυτές οι ροές για την δημιουργία και την τροποποίηση του περιεχομένου. Κατά την δοκιμή της εφαρμογής, εισάγονται μεταβατικές αλλαγές στη συνδεσιμότητα δικτύου, τη λειτουργία μπαταρίας, το διαθέσιμο GPS ή τη θέση, το φορτίο του συστήματος κ.ο.κ.
- Πάτημα του πλήκτρου Home από κάθε οθόνη της εφαρμογής, και, στη συνέχεια, επανεκκίνηση της εφαρμογής από το μενού της συσκευής όπου βρίσκονται όλες οι εφαρμογές.
- Από κάθε οθόνη εφαρμογής, μετάβαση σε άλλη εφαρμογή που εκτελείται και, στη συνέχεια, επιστροφή στην εφαρμογή υπό δοκιμή χρησιμοποιώντας το κουμπί της συσκευής που ανοίγει τις πρόσφατες εφαρμογές.
- Πάτημα του πλήκτρου Πίσω από κάθε οθόνη εφαρμογής (και παράθυρο διαλόγου)
- Περιστροφή της συσκευή ανάμεσα σε οριζόντιο και κατακόρυφο προσανατολισμό τουλάχιστον τρεις φορές, από κάθε οθόνη της εφαρμογής.

- Μετάβαση σε άλλη εφαρμογή για να σταλεί η εφαρμογή δοκιμής στο παρασκήνιο. Μετάβαση στις Ρυθμίσεις και έλεγχος για το εάν η εφαρμογή δοκιμής έχει οποιοσδήποτε υπηρεσίες που εκτελούνται ενώ βρίσκεται στο παρασκήνιο. Για το Android 4.0 και νεότερες εκδόσεις, μετάβαση στην οθόνη εφαρμογών και εύρεση της εφαρμογής στην καρτέλα με τις εφαρμογές που τρέχουν στη συσκευή.
- Πάτημα στο κουμπί λειτουργίας για να μπει η συσκευή σε αναστολή λειτουργίας και, στη συνέχεια, πάτημα ξανά στο κουμπί λειτουργίας για να «ξυπνήσει» η οθόνη.
- Ρύθμιση της συσκευής να κλειδώνει όταν πιέζεται το κουμπί τροφοδοσίας. Πάτημα στο κουμπί τροφοδοσίας για να μπει η συσκευή σε αναστολή λειτουργίας και, στη συνέχεια, πάτημα ξανά στο κουμπί λειτουργίας για να ενεργοποιηθεί η οθόνη και στη συνέχεια ξεκλείδωμα της συσκευής.
- Για συσκευές που έχουν πληκτρολόγιο με ολίσθηση, να συρθεί τουλάχιστον μία φορά το πληκτρολόγιο προς τα μέσα και προς τα έξω. Για συσκευές που διαθέτουν αποσπώμενα πληκτρολόγια, σύνδεση της συσκευής στην βάση του πληκτρολογίου.
- Για συσκευές με εξωτερική θύρα οθόνης, σύνδεση της εξωτερικής οθόνης.
- Ενεργοποίηση και παρατήρηση όλων των τύπων ειδοποιήσεων που μπορεί να εμφανίσει η εφαρμογή στο συρτάρι ειδοποιήσεων. Επέκταση των ειδοποιήσεων όπου ισχύει (Android 4.1 και νεότερες εκδόσεις) και πάτημα όλων των προσφερόμενων ενεργειών.

5.2 Δοκιμή μονάδας

Έγιναν δοκιμές στην εφαρμογή για τον έλεγχο της ορθής λειτουργικότητας των εφαρμοζόμενων μεθόδων. Οι δοκιμές πραγματοποιήθηκαν στην εφαρμογή Android Studio, η οποία χρησιμοποιήθηκε επίσης ως περιβάλλον για την ανάπτυξη της εφαρμογής.

Για την επιβεβαίωση ότι όλες οι εφαρμοζόμενες λειτουργίες τρέχουν σωστά, κατά τη διάρκεια της ανάπτυξης δοκιμάστηκε κάθε νέα δυνατότητα με όλους τους δυνατούς τρόπους που μπορεί να χρησιμοποιήσει ο χρήστης. Όταν εμφανιζόταν κάποια αποτυχία (δυσλειτουργία), το θέμα επιλύονταν αμέσως πριν την ανάπτυξη οποιουδήποτε νέου χαρακτηριστικού. Αυτές οι δοκιμές χρειάστηκαν πολύ χρόνο, αλλά πρόσφεραν πολύτιμες πληροφορίες για τα χαρακτηριστικά που αναπτύχθηκαν στην εφαρμογή.

5.3 Δοκιμή Ευχρηστίας

Η καλύτερη ποιότητα της εφαρμογής επιτυγχάνεται με μια δοκιμή ευχρηστίας. Η δοκιμή ευχρηστίας γίνεται με χρήστες, οι οποίοι λαμβάνουν ένα σύνολο εργασιών και πρέπει να τις ολοκληρώσουν. Η δραστηριότητα των χρηστών και οι αλληλεπιδράσεις παρακολουθούνται και προσφέρουν τη δυνατότητα εμφάνισης σφαλμάτων εφαρμογής, κατά κύριο λόγο σε επίπεδο διεπαφής χρήστη. Αυτές οι δοκιμές δίνουν άμεση πληροφόρηση σχετικά με τον τρόπο με τον οποίο οι πραγματικοί χρήστες χρησιμοποιούν το σύστημα και αποκαλύπτονται προβλήματα που δεν εμφανίζονται με την στενή προσέγγιση στην δοκιμή ευχρηστίας από έμπειρους προγραμματιστές που προσπαθούν να αξιολογήσουν την διεπαφή χρήστη.

Η δοκιμή ευχρηστίας δεν απαιτεί περισσότερους από 5 χρήστες, επειδή τα κύρια προβλήματα θα ανακαλυφθούν από τον πρώτο ή τον δεύτερο χρήστη κατά την δοκιμή της εφαρμογής. Οι επόμενοι χρήστες συνήθως αποκαλύπτουν τα ίδια προβλήματα με τους προηγούμενους δοκιμαστές και δεν ανακαλύπτουν νέα [28], [29].

Ζητήθηκε από τους χρήστες να πραγματοποιήσουν τα εξής βήματα:

1. Εύρεση τριών μονοπατιών
2. Αποθήκευση σχολίων για κάθε μονοπάτι
3. Διαγραφή ενός από τα σχόλια
4. Αποστολή e-mail στον δημιουργό της εφαρμογής
5. Κοινοποίηση της εφαρμογής
6. Έξοδος από την εφαρμογή

Οποιοσδήποτε μπορεί να χρησιμοποιήσει την εφαρμογή SantoriniTrail. Δεν αναζητήθηκαν χρήστες με συγκεκριμένα χαρακτηριστικά (ηλικία, φύλο ή εμπειρία κινητού Android). Υπήρχαν πέντε χρήστες για τη δοκιμή χρηστικότητας: τέσσερις χρήστες κινητού τηλεφώνου Android και ένας χρήστης κινητής iOS. Οι χρήστες κινητού Android έλαβαν το κινητό τους με εγκατεστημένη την εφαρμογή SantoriniTrail και το 5ο άτομο έλαβε μια ταμπλέτα με εγκατεστημένη την εφαρμογή. Δόθηκαν οδηγίες και τα βήματα που έπρεπε να πραγματοποιηθούν χωρίς να τους υποδειχτεί η εφαρμογή στην συσκευή.

5.4 Αποτελέσματα

Έπειτα από τις δοκιμές διαπιστώθηκαν κάποια σφάλματα:

1. Εύρεση του μονοπατιού

Ο χρήστης δυσκολευόταν να βρει τα μονοπάτια.

Συχνότητα Περιστατικού: Υψηλή

Περιγραφή: Έκαναν κλικ πάνω στον χάρτη, αντί να βρουν τις τρεις παράλληλες γραμμές που ανοίγουν το κυλιόμενο μενού πλοήγησης ή τις τρεις βούλες ή την συντόμευση με τις 3 τελείες στην δεξιά πλευρά της οθόνης, όπου υπάρχει η επιλογή εύρεση του μονοπατιού.

Επίλυση: Δεν έχει επιλυθεί. Ακολουθεί της οδηγίες σχεδιασμού της εφαρμογής ως έχουν. Αν προχωρήσει προς εμπορική χρήση θα διορθωθεί μελλοντικά.

2. Εγγραφή σχολίων

Ο χρήστης δυσκολευόταν να βρει που εισάγονται τα σχόλια

Συχνότητα Περιστατικού: Υψηλή

Περιγραφή: Επειδή το κουμπί έγραφε «Εμφάνιση Σχολίων» δεν μπορούσαν να κάνουν την σύνδεση ότι αυτός είναι ο χώρος για την εγγραφή των σχολίων.

Επίλυση: Διορθώθηκε. Τώρα το κουμπί αλλάζει ανάμεσα σε 2 καταστάσεις. Η πρώτη είναι το «Γράψε Σχόλια» και η δεύτερη το «Εμφάνιση Σχολίων»

3. Αποθήκευση σχολίων

Ο χρήστης δυσκολευόταν να βρει πως αποθηκεύονται τα σχόλια

Συχνότητα Περιστατικού: Χαμηλή

Περιγραφή: Ο χρήστης δεν βλέπει το κουμπί «Καταχώρησε. Αυτό συνέβη σε ενήλικο χρήστη άνω των 65.

Επίλυση: Δεν διορθώθηκε. Προς το παρόν η εφαρμογή δεν στοχεύει συγκεκριμένα σε αυτήν την ηλικιακή ομάδα. Αν προχωρήσει προς εμπορική χρήση θα διορθωθεί ο σχεδιασμός και η εμφάνιση του κουμπιού.

4. Διαγραφή σχολίων

Ο χρήστης δυσκολευόταν να βρει πώς διαγράφονται τα σχόλια

Συχνότητα Περιστατικού: Μέτρια

Περιγραφή: Επειδή η διαγραφή γίνεται με παρατεταμένο πάτημα πάνω στο σχόλιο οι μεγαλύτεροι σε ηλικία ενήλικες χρήστες δεν ήταν συνηθισμένοι σε αυτόν τον τρόπο διαγραφής.

Επίλυση: Δεν διορθώθηκε. Αν προχωρήσει προς εμπορική χρήση θα προστεθεί και κάποιο κουμπί για την διαγραφή.

5. Ένα άχρηστο βήμα

Περιγραφή: Το κουμπί «Κλικ αν έγινε» για την αποθήκευση της ημερομηνίας δεν χρησιμεύει κάπου. Όταν γίνεται εγγραφή των σχολίων αποθηκεύεται αυτόματα η ημερομηνία και η ώρα της εγγραφής.

Συχνότητα Περιστατικού: Υψηλή

Αιτία: Λανθασμένη σύλληψη της χρησιμότητας του κουμπιού από τον προγραμματιστή.

Επίλυση: Δεν έχει επιλυθεί, θα επιλυθεί στην επόμενη έκδοση της εφαρμογής.

ΚΕΦΑΛΑΙΟ 6

Συμπεράσματα

Ο σκοπός αυτής της εργασίας είναι η ανάπτυξη μιας πλήρους λειτουργικής εφαρμογής Android για πλοήγηση στα μονοπάτια της Σαντορίνης. Ο στόχος αυτός επιτεύχθηκε. Χρησιμοποιήθηκε το πιο διαδεδομένο λειτουργικό σύστημα για κινητά και δημιουργήθηκε η εφαρμογή SantoriniTrail που στοχεύει στο πιο πολυσύχναστο νησί της Ελλάδος, για να διευκολύνει τους περιπατητές στην εύρεση μονοπατιών και την γνωριμία με την ιστορία του νησιού. Προβλήματα και δυσκολίες εμφανίστηκαν κατά τη διάρκεια της ανάπτυξης. Επίσης στις δοκιμές που έγιναν, έγιναν αντιληπτά τα σημεία που χρειάζονται βελτίωση και γεννήθηκαν νέες ιδέες.

Η εφαρμογή εφαρμόστηκε σε συσκευές κινητών και tablet και είχε τη δυνατότητα να ρυθμίζεται το μέγεθος και ο σχεδιασμός κειμένου για κάθε συσκευή. Η εφαρμογή παρέχεται σε δύο γλώσσες: Ελληνική και Αγγλική. Ένα εγχειρίδιο χρήσης γράφτηκε για την εφαρμογή. Η εφαρμογή δοκιμάστηκε για να διασφαλιστεί ότι λειτουργεί σωστά. Κατά τη γνώμη μου, είναι έτοιμη να χρησιμοποιηθεί σε λειτουργία δοκιμής.

6.1 Μελλοντικές Βελτιώσεις

Η ανάπτυξη της εφαρμογής ήταν ένα μεγάλο βήμα προς την απόκτηση εμπειρίας και την βελτίωση δεξιοτήτων επίλυσης προβλημάτων. Η περαιτέρω ανάπτυξη μπορεί να στοχεύσει σε μια πλουσιότερη εμπειρία χρήστη, όπως για παράδειγμα ο χρήστης να μπορεί να κοινοποιεί πληροφορίες για το μονοπάτι που ολοκλήρωσε όπως τον χρόνο που χρειάστηκε, την απόσταση, τα σχόλιά του.

Μια άλλη ιδέα είναι οι ενημερώσεις της τοποθεσίας να καταγράφονται και να μοιράζονται με τρίτους σε πραγματικό χρόνο, για την διατήρηση της ανάμνησης ή για λόγους ασφαλείας.

Θα μπορούσαν επίσης να αναπτυχθούν οι λειτουργίες πολυμέσων. Η εφαρμογή θα μπορούσε να κάνει λήψη φωτογραφιών και να μοιράζονται και αυτές μέσω ηλεκτρονικού ταχυδρομείου ή σε μέσα κοινωνικής δικτύωσης. Επίσης και η δυνατότητα μεταφόρτωσης ενός βίντεο μπορεί να προσθέσει μια πλουσιότερη εμπειρία χρήστη. Η παροχή ζωντανής πλοήγησης μπορεί επίσης να συμπεριληφθεί ώστε να είναι ευκολότερο να επισκεφτεί κανείς το μονοπάτι πάλι.

ΠΑΡΑΡΤΗΜΑ Α΄

Στο παράρτημα αυτό παρατίθεται ο κώδικας ανάπτυξης της παρούσας εφαρμογής.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.teipir.margarita.santorinitrail">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"
/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
/>

    <permission
        android:name="com.example.androidmapsv2.permission.MAPS_RECEIVE"
        android:protectionLevel="signature" />

    <uses-permission
android:name="com.example.androidmapsv2.permission.MAPS_RECEIVE" />
    <uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainMenu"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.VIEW" />

                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
        <activity
            android:name=".ListTrailsActivity"
            android:label="@string/title_activity_list_trails"></activity>
        <activity
            android:name=".PathDetailsActivity"
            android:label="@string/title_activity_path_details"></activity>
        <activity
            android:name=".SplashscreenActivity"
            android:configChanges="orientation|keyboardHidden|screenSize"
            android:label="@string/app_name"
            android:theme="@style/FullscreenTheme">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <meta-data
```

Ανάπτυξη εφαρμογής περιήγησης σε μονοπάτια «SANTORINI TRAIL» για Android

```
        android:name="com.google.android.geo.API_KEY"
        android:value="AIzaSyDR7g4KAex6SBIU2jJiwIxfv7ZttvuD29M" />

    <activity
        android:name=".CommentActivity"
        android:label="@string/title_activity_comment"></activity>
</application>

</manifest>

public class SplashscreenActivity extends Activity {
    protected boolean _active = true;
    protected int _splashTime = 5000;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        this.requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.activity_splashscreen);
        // thread for displaying the SplashScreen
        Thread splashTread = new Thread() {
            @Override
            public void run() {
                try {
                    int waited = 0;
                    while (_active && (waited < _splashTime)) {
                        sleep(100);
                        if (_active) {
                            waited += 100;
                        }
                    }
                } catch (InterruptedException e) {
                    // do nothing
                } finally {
                    finish();
                    startActivity(new Intent(SplashscreenActivity.this,
MainMenu.class));
                }
            }
        };
        splashTread.start();

    }

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
            _active = false;
        }
        return true;
    }
}

public class MainMenu extends MyBaseActivity
implements OnMapReadyCallback {

    public dbhandler db = new dbhandler(this);

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        db = new dbhandler(this);
        setContentView(R.layout.mainmenu2);
    }
}
```

Ανάπτυξη εφαρμογής περιήγησης σε μονοπάτια «SANTORINI TRAIL» για Android

```
createMaterial(DRAWER_HOME, "Santorini Trail", "by Margarita K.");

MapFragment mapFragment = (MapFragment) getFragmentManager()
    .findFragmentById(R.id.fragment2);
mapFragment.getMapAsync(this);
}

@Override
public void onMapReady(GoogleMap map) {
    LatLng santorini = new LatLng(36.407330, 25.426498);

    map.setMyLocationEnabled(true);
    map.moveCamera(CameraUpdateFactory.newLatLngZoom(santorini, 12));
}

}

public class MyBaseActivity extends ActionBarActivity {
    public Toolbar toolbar;
    public Drawer drawer;
    public int selectItem = -1;
    public static final int DRAWER_HOME = 1;
    public static final int DRAWER_TRAILS = 2;
    public static final int DRAWER_SHARE = 3;
    public static final int DRAWER_CONTACTUS = 4;
    public static final int DRAWER_USERPROFILE = 5;

    public void Exit() {
        System.runFinalizersOnExit(true);
        System.exit(0);
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Override
    protected void onSaveInstanceState(Bundle outState) {
        super.onSaveInstanceState(outState);
    }

    protected void createMaterial(int selectdrawer, String ToolbarTitle,
String ToolbarSubtitle) {
        toolbar = (Toolbar) findViewById(R.id.my_toolbar);
        toolbar.setTitle(ToolbarTitle);
        toolbar.setSubtitle(ToolbarSubtitle);
        selectItem = selectdrawer;
        toolbar.inflateMenu(R.menu.my_menu);
        toolbar.setOnMenuItemClickListener(new
Toolbar.OnMenuItemClickListener() {
            @Override
            public boolean onMenuItemClick(MenuItem item) {
                switch (item.getItemId()) {
                    case R.id.action_search:
                        openfind();
                        return true;
                }
            }
        });
    }
}
```

```

        default:
            return true;
    }
}
});
drawer = new
DrawerBuilder().withActivity(this).withToolbar(toolbar).withHeader(R.layout.d
rawerheader).addDrawerItems(
    new
PrimaryDrawerItem().withName(R.string.Home).withIcon(R.drawable.ic_home_black
_18dp).withIdentifier(DRAWER_HOME).withCheckable(false),
    new DividerDrawerItem(),
    new
PrimaryDrawerItem().withName(R.string.findtrail).withIcon(R.drawable.ic_searc
h_black_18dp).withIdentifier(DRAWER_TRAILS).withCheckable(false),
    new DividerDrawerItem(),
    new
PrimaryDrawerItem().withName(R.string.share).withIcon(R.drawable.ic_share_bla
ck_18dp).withIdentifier(DRAWER_SHARE).withCheckable(false),
    new DividerDrawerItem(),
    new
PrimaryDrawerItem().withName(R.string.contactus).withIcon(R.drawable.ic_email
_black_18dp).withIdentifier(DRAWER_CONTTACTUS).withCheckable(false),
    new DividerDrawerItem(),
    new
PrimaryDrawerItem().withName(R.string.Exit).withIcon(R.drawable.ic_account_ci
rcle_black_18dp).withIdentifier(DRAWER_USERPROFILE).withCheckable(false),
    new DividerDrawerItem()
).withOnDrawerItemClickListener(new
Drawer.OnDrawerItemClickListener() {
    @Override
    public boolean onItemClick(AdapterView<?> parent, View view, int
position, long id, IDrawerItem drawerItem) {
        if (drawerItem != null) {
            switch (drawerItem.getIdentifier()) {
                case 1:
                    openhome();
                    break;
                case 2:
                    openfind();
                    break;
                case 3:
                    share();
                    break;
                case 4:
                    opencontactus();
                    break;
                case 5:
                    Exit();
                    break;
            }
        }
        return false;
    }
}).build();
}
protected void openfind() {
    if (selectItem != DRAWER_TRAILS) {
        Intent intent = new Intent(this, ListTrailsActivity.class);
        startActivity(intent);
    }
}
protected void openhome() {

```


Ανάπτυξη εφαρμογής περιήγησης σε μονοπάτια «SANTORINI TRAIL» για Android

```
if (selectItem != DRAWER_HOME) {
    Intent intent = new Intent(this, MainMenu.class);
    startActivity(intent);
}
}
protected void share() {
    Intent sendIntent = new Intent();
    sendIntent.setAction(Intent.ACTION_SEND);
    sendIntent.putExtra(Intent.EXTRA_TEXT, "Check out this app:
www.santorinitrail.simplesite.com.");
    sendIntent.setType("text/plain");
    startActivity(sendIntent);
}
protected void openprofile() {
}
protected void opencontactus() {
    Intent emailIntent = new Intent(Intent.ACTION_SENDTO, Uri.fromParts(
        "mailto", "santorinitrail@gmail.com", null));
    emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Subject");
    startActivity(Intent.createChooser(emailIntent, "Send email..."));
}
}

public class ListTrailsActivity extends MyBaseActivity {
    public dbhandler db = new dbhandler(this);
    List<trail> trailList;
    TrailAdapter trailAdapter;
    ListView listView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_list_trails);
        loadtrails();
        createMaterial(DRAWER_TRAILS, "Select a Trail", "");
        EditText filterText = (EditText) findViewById(R.id.filter);
        filterText.addTextChangedListener(filtertextwatcher);
    }

    private void loadtrails() {
        trailList = db.getAlltrails("");
        trailAdapter = new TrailAdapter(this, R.layout.adapter_itemtrail,
trailList);
        listView = (ListView) findViewById(R.id.listTrailView);
        listView.setAdapter(trailAdapter);
        listView.setClickable(true);
        listView.setOnItemClickListener(
            new AdapterView.OnItemClickListener() {
                public void onItemClick(
                    @SuppressWarnings("rawtypes") AdapterView parent,
                    View view, int position, long id) {
                    trail t = (trail)
listView.getItemAtPosition(position);
                    Intent intent = new Intent(ListTrailsActivity.this,
PathDetailsActivity.class);
                    intent.putExtra("id", t.get_id());
                    startActivity(intent);
                }
            });
    }
}
```

```

private TextWatcher filtertextwatcher = new TextWatcher() {
    public void afterTextChanged(Editable s) {
    }

    public void beforeTextChanged(CharSequence s, int start, int count,
        int after) {
    }

    public void onTextChanged(CharSequence s, int start, int before,
        int count) {
        trailAdapter.getFilter().filter(s);
    }
};
}

public class PathDetailsActivity extends MyBaseActivity
    implements OnMapReadyCallback, BaseDialogFragment.BaseDialogInterface
{
    public dbhandler db=new dbhandler(this);
    private completedTr cTr;

    private trail tr;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Intent intent =getIntent();
        int id = intent.getIntExtra("id",0);
        setContentView(R.layout.activity_path_details);

        tr = (trail)db.get_trail(id);

        TextView showDescription =
        (TextView) findViewById(R.id.showDescription);
        showDescription.setText(tr.get_description());

        TextView showHeight = (TextView) findViewById(R.id.showHeight);
        showHeight.setText(Integer.toString(tr.get_height()));

        TextView showLength = (TextView) findViewById(R.id.showLength);
        showLength.setText(Integer.toString(tr.get_length()));
        int min=tr.get_minutes();
        int hours=min/60;
        min=min%60;
        TextView showHours = (TextView) findViewById(R.id.showHours);
        TextView showMin = (TextView) findViewById(R.id.showMin);
        TextView labelHours = (TextView) findViewById(R.id.labelHours);
        TextView labelMinutes = (TextView) findViewById(R.id.labelMinutes);
        if (hours >0) {
            showHours.setText(Integer.toString(hours));
            showHours.setVisibility(TextView.VISIBLE);
            labelHours.setVisibility(TextView.VISIBLE);
        }
        else
        { showHours.setVisibility(TextView.GONE);
            labelHours.setVisibility(TextView.GONE);
        }

        if (min >0) {
            showMin.setText(Integer.toString(min));
            showMin.setVisibility(TextView.VISIBLE);
            labelMinutes.setVisibility(TextView.VISIBLE);
        }
    }
}

```

```

    }
    else {
        showMin.setVisibility(View.GONE);
        labelMinutes.setVisibility(View.GONE);
    }
    checkCompleted();
    MapFragment mapFragment = (MapFragment) getSupportFragmentManager()
        .findFragmentById(R.id.map2);
    mapFragment.getMapAsync(this);

    createMaterial(-
1, tr.getName(), tr.getDifficultyText(this.getResources()));

}

@Override
public void onMapReady(GoogleMap map) {
    if (tr != null) {
        LatLng start = new LatLng (tr.getStartLatitude(),
tr.getStartLongitude());
        LatLng end = new LatLng (tr.getEndLatitude(),
tr.getEndLongitude());
        LatLng middle = new LatLng(Math.min(tr.getStartLatitude(),
tr.getEndLatitude())+
            Math.abs(tr.getStartLatitude()-
tr.getEndLatitude())/2, (Math.min(tr.getStartLongitude(),
tr.getEndLongitude())+
            Math.abs(tr.getStartLongitude()-
tr.getEndLongitude())/2);
        map.setMyLocationEnabled(true);
        map.moveCamera(CameraUpdateFactory.newLatLngZoom(middle, 12));
        map.addMarker(new MarkerOptions()
            .position(start)
            .title("Start"));
        map.addMarker(new MarkerOptions()
            .position(end)
            .title("End"));
    }

}

public void setCompleted (View view){

    if (cTr==null) {
        DialogFragment completedialog = new completetraildialog();
        completedialog.show(getSupportFragmentManager(), "Confirm");
    }
    else{
        DialogFragment confirm = new confirmdialog();
        confirm.show(getSupportFragmentManager(), "Confirm");
    }

}

@Override
public void share() {
    if(cTr!=null) {
        Intent sendIntent = new Intent();
        sendIntent.setAction(Intent.ACTION_SEND);
        sendIntent.putExtra(Intent.EXTRA_TEXT, " I completed the trail
"+tr.getName()+" in Santorini! Check it out! "+tr.getDescription());
        sendIntent.setType("text/plain");
    }
}

```

```

        startActivity(sendIntent);
    }
    else{
        Intent sendIntent = new Intent();
        sendIntent.setAction(Intent.ACTION_SEND);
        sendIntent.putExtra(Intent.EXTRA_TEXT, " This trail "+tr.get_name()+" in
Santorini is great! Check it out! "+tr.get_description());
        sendIntent.setType("text/plain");
        startActivity(sendIntent);
    }
}

    public void checkCompleted (){
        List<completedTr> tempList = db.getAllCompletedTr(" where
trail_id="+tr.get_id());
        Button sb = (Button)findViewById(R.id.showCompleted);
        if(tempList.size()>0){
            cTr = tempList.get(0);
            sb.setText(R.string.showCompleted);
        }

        else{
            cTr = null;
            sb.setText(R.string.showCompleted1);
        }

    }

    public void showcomments(View view){
        Intent intent = new Intent(this,CommentActivity.class);
        intent.putExtra("id", tr.get_id());
        startActivity(intent);
    }

    @Override
    public void onPositiveClick(DialogFragment dialog) {
        if(cTr==null){
            completetraildialog ctd = (completetraildialog)dialog;
            cTr = new completedTr();
            cTr.set_trail_id(tr.get_id());
            cTr.set_id(db.getCompletedTrNextId());
            cTr.set_timecompleted(ctd.selecteddate);
            cTr.set_minutes(ctd.selectedtime);
            db.add_completedTr(cTr);
        }
        else{
            db.deletecompletedTr(cTr.get_id());
        }
        checkCompleted();
    }

    @Override
    public void onNegativeClick(DialogFragment dialog) {
    }
}

public class CommentActivity extends MyBaseActivity {

```

```

public dbhandler db=new dbhandler(this);
List<comment> commentList;
CommentAdapter commentAdapter;
ListView listView;
int clickposition;

private trail tr;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    Intent intent =getIntent();
    int id = intent.getIntExtra("id",0);
    setContentView(R.layout.activity_comment);

    tr = (trail)db.get_trail(id);

    createMaterial(-1,tr.get_name(),"Comments");
loadcomments();

}

private void loadcomments(){
    commentList = db.getAllcomments(" where trail_id = "+tr.get_id());
    commentAdapter = new
CommentAdapter(this,R.layout.comment_data,commentList);
    listView = (ListView) findViewById(R.id.comment_list);
    listView.setAdapter(commentAdapter);
    listView.setLongClickable(true);
    listView.setOnItemLongClickListener(
        new AdapterView.OnItemClickListener() {

            public boolean onItemClick(
                @SuppressWarnings("rawtypes") AdapterView parent,
                View view, int position, long id) {
clickposition = position;
                AlertDialog.Builder myDialog
                    = new
AlertDialog.Builder(CommentActivity.this);

                myDialog.setTitle("Delete?");

                myDialog.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                    // do something when the button is clicked
                    public void onClick(DialogInterface arg0, int
arg1) {
                        comment c =
(comment)listView.getItemAtPosition(clickposition);
                        if (c!=null)
                            db.deletecomment(c.get_id());
                        loadcomments();
                    }
                });

                myDialog.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
                    // do something when the button is clicked
                    public void onClick(DialogInterface arg0, int
arg1) {

```

```

        });
        myDialog.show();

        comment c =
(comment) listView.getItemAtPosition(position);
        if (c!=null)
            db.deletecomment(c.get_id());

        return true;
    });
}

public void createcomment(View view){
    EditText et = (EditText)findViewById(R.id.comment);
    String description = et.getText().toString();

    comment c = new comment(db.getCommentNextId(), tr.get_id(), new Date(),
description);
    db.add_comment(c);
    loadcomments();

}

}

public class CommentAdapter extends ArrayAdapter<comment> {
    private List<comment> items;

    public CommentAdapter(Context context, int resource, List<comment>
objects) {
        super(context, resource, objects);
        items = objects;
    }

    public int getCount() {
        return items.size();
    }

    public comment getItem(int position) {
        return items.get(position);
    }

    public int getPosition(int id) {
        int countcomments = getCount();
        for (int i = 0; i < countcomments; ++i) {
            if (items.get(i).get_id() == id) {
                return i;
            }
        }
        return -1;
    }

    @Override

```

Ανάπτυξη εφαρμογής περιήγησης σε μονοπάτια «SANTORINI TRAIL» για Android

```
public View getView(int position, View convertView, ViewGroup parent) {
    View v = convertView;
    if (v == null) {
        LayoutInflater vi = (LayoutInflater)
getContext().getSystemService(
        Context.LAYOUT_INFLATER_SERVICE);
        v = vi.inflate(R.layout.comment_data, null);
    }
    comment c = getItem(position);
    if (c != null) {
        TextView tv = (TextView) v.findViewById(R.id.user_comment);
        if (tv != null) {
            tv.setText(c.get_description());
        }
        TextView cv = (TextView) v.findViewById(R.id.date);
        if (cv != null) {
            DateFormat iso8601Format = new SimpleDateFormat("dd-MM-yyyy
HH:mm", Locale.getDefault());
            cv.setText(iso8601Format.format(c.get_createdtime()));
        }
    }
    return v;
}

public class TrailAdapter extends ArrayAdapter<trail> {
    private List<trail> items;

    private List<trail> originalitems;

    public TrailAdapter(Context context, int resource, List<trail> objects) {
        super(context, resource, objects);
        items = objects;
    }

    public int getCount() {
        return items.size();
    }

    public trail getItem(int position) {
        return items.get(position);
    }

    public int getPosition(int id) {
        int counttrails = getCount();
        for (int i = 0; i < counttrails; ++i) {
            if (items.get(i).get_id() == id) {
                return i;
            }
        }
        return -1;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent)
    {
        View v = convertView;
        if (v == null) {
            LayoutInflater vi = (LayoutInflater)
```

```

getContext().getSystemService(
    Context.LAYOUT_INFLATER_SERVICE);
    v = vi.inflate(R.layout.adapter_itemtrail, null);
}
trail t = getItem(position);
if (t!=null) {
    TextView tv = (TextView) v.findViewById(R.id.trailname);
    if (tv != null) {
        tv.setText(t.get_name());
    }
}
return v;
}

@Override
public Filter getFilter(){return new Filter(){
    @Override
    protected android.widget.Filter.FilterResults
performFiltering(CharSequence constraint) {
        final android.widget.Filter.FilterResults oReturn = new
android.widget.Filter.FilterResults();
        final List<trail> results =new ArrayList<trail>();
        if (originalitems == null) {originalitems=items;}
        if (constraint!=null)
        {
            if (constraint.length()==0) {
                for (final trail g : originalitems) {
                    results.add(g);
                }
            }
            else {for (final trail g : originalitems) {
                if (g.get_description().contains(constraint)) {
                    results.add(g);
                }
            }
        }
        oReturn.values=results;
    }

    return oReturn;
}

@SuppressWarnings("unchecked")
@Override
protected void publishResults(CharSequence constraint,
    FilterResults results) {
    items = (ArrayList<trail>) results.values;
    notifyDataSetChanged();
}

};}

}

public class comment {

int __id;

    public int get_id() {
        return this.__id;
    }
}

```



```

public void set_id(int _id) {
    this.__id = _id;
}

int _trail_id;

public int get_trail_id() {
    return this._trail_id;
}

public void set_trail_id(int trail_id ) {
    this._trail_id = trail_id;
}

Date _createdtime;

public Date get_createdtime() {
    return this._createdtime;
}

public void set_createdtime(Date createdtime ) {
    this._createdtime = createdtime;
}

public void set_createdtime_forSQL(String createdtime ) {
    DateFormat iso8601Format = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
    try {
        _createdtime = iso8601Format.parse(createdtime);
    } catch (ParseException e) {
    }
}

public String get_createdtime_forSQL() {
    DateFormat iso8601Format = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss", Locale.getDefault());

    return iso8601Format.format(_createdtime);
}

String _description;

public String get_description() {
    return this._description;
}

public void set_description(String description) {
    this._description = description;
}

public comment(int _id, int trail_id, Date createdtime,String description){
    set_id(_id);
    set_trail_id(trail_id);
    set_createdtime(createdtime);
    set_description(description);
}

public comment () {
    set_id(0);
    set_trail_id(0);
    set_createdtime(new Date());
}

```

```

        set_description("");
    }
}
public class completedTr {

    int __id;

    public int get_id() {
        return this.__id;
    }

    public void set_id(int _id) {
        this.__id = _id;
    }

    int _trail_id;

    public int get_trail_id() {
        return this._trail_id;
    }

    public void set_trail_id(int trail_id ) {
        this._trail_id = trail_id;
    }

    Date _timecompleted;

    public Date get_timecompleted() {
        return this._timecompleted;
    }

    public void set_timecompleted(Date timecompleted ) {
        this._timecompleted = timecompleted;
    }

    public void set_timecompleted_forSQL(String timecompleted ) {
        DateFormat iso8601Format = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss");
        try {
            _timecompleted = iso8601Format.parse(timecompleted);
        } catch (ParseException e) {
        }
    }

    public String get_timecompleted_forSQL() {
        DateFormat iso8601Format = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss", Locale.getDefault());

        return iso8601Format.format(_timecompleted);
    }

    int _minutes;

    public int get_minutes() {
        return this._minutes;
    }

    public void set_minutes(int minutes) {
        this._minutes = minutes;
    }

    public completedTr(int _id, int trail_id, Date timecompleted, int

```

```

minutes){
    set_id(_id);
    set_trail_id(trail_id);
    set_timecompleted(timecompleted);
    set_minutes(minutes);

}

public completedTr () {
    set_id(0);
    set_trail_id(0);
    set_timecompleted(new Date());
    set_minutes(0);

}
}

public class dbhandler extends SQLiteOpenHelper {

    public static final int DATABASE_VERSION = 15;
    public static final String DATABASE_NAME = "SantoriniTrail";

    public dbhandler(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {

        createCommentTable(db);
        createCompletedTrailsTable(db);
        populateTrails(db);

    }

    public void createCommentTable(SQLiteDatabase db) {
        db.execSQL("drop table if exists comment;");
        db.execSQL("create table comment (_id integer primary key, trail_id
integer,description text, createdtime text);");
    }

    public void createCompletedTrailsTable(SQLiteDatabase db) {
        db.execSQL("drop table if exists completedTr;");
        db.execSQL("create table completedTr (_id integer primary key,
trail_id integer,minutes integer,timecompleted text );");
    }

    public void populateTrails(SQLiteDatabase db) {
        db.execSQL("drop table if exists trail;");
        db.execSQL("create table trail (_id integer primary key, name
text,length integer," +
            " height integer, minutes integer, difficulty integer,
description text, startlatitude real," +
            " startlongitude real, endlatitude real, endlongitude real
);");

        db.execSQL("INSERT INTO trail
(_id,name,length,height,minutes,difficulty, startlatitude, startlongitude ,
endlatitude , endlongitude,description) " +

```

```
"VALUES ('9','Fira to Oia / Φηρά προς Οία  
, '9', '360', '200', '1', '36.420368', '25.431981', '36.461546', '25.374632', 'The  
hike that is on most of Santorini's visitors to-do list, leads hikers from  
the island's capital Fira, through the villages of Firostefani and Imerovigli  
and then takes them along the rim of the Caldera until they reach the village  
of Oia. From the cliff edges you get magnificent views of the volcano and the  
Aegean Sea that change as you tread across the northern part of the caldera.  
Depending on your physical condition and how many photo stops you will make  
(count on making many), the trail can take from 2.5 hours up to 5 hours. The  
trail can also be done from Oia towards Fira, with the same level of  
difficulty as altitude differences are insignificant. If planned correctly  
the upside of beginning in Fira is ending up in Oia around sunset time and  
relaxing just before the entrance to the village at the church of Panagia to  
enjoy the view in a less crowded spot. It is advised to bring water along and  
to wear closed shoes as part of the trail is full of small volcanic pebbles,  
that always find their way between sandals and the soles of feet. Also keep  
in mind that taking a bus or finding a taxi right after the sunset in Oia can  
be a challenge as the spot gets overcrowded.\n" +
```

```
"\n" +
```

```
"Start the trail a few meters away from the bus and taxi  
stations in Fira, on the street across the busy Theotokopoulos square, where  
the bookstore, bank and cafes are situated. Take the cobbled street between  
the optical store and the travel agency Pelican Travel, head upwards and take  
a right after Hondos Center. Continue ascending on this road past the  
nightclubs and shops until you reach the Archaeological museum, where you  
should turn left. Pass the entrance to the cable car and turn right at the  
corner just after it, where the restaurant Zaphora is located. Continue  
ascending, along the rim of the caldera, taking a left turn when you see  
Irimi's café and pass in front of the reddish old manor, a neoclassic  
building, home to the Nomikos Conference Centre. Continue following the paved  
steps into Firostefani, the village that "crowns" Fira. At Agios Gerasimos  
Church continue following the path as it takes you away from the cliffs edge  
and into a busier street in the middle of the village. Keep walking on this  
path as it takes you out of the village into a short dirt road and then  
becomes a paved path again that brings you to the monastery of Agios  
Nikólaos. Continue on the path and enter Imerovigli. Keep on the path and  
turn left on the first path you see after Chromata hotel. Try to stay on the  
alleys on the left to pass catch a glimpse of some luxurious hotels and  
villas, and the view of the caldera. The courtyard of Anastasi Church, is a  
nice stop for some photos and move on towards Blue Note café, from this point  
you can make a detour for Skaros Rock, one of the most impressive spots of  
Santorini. Just turn left and start descending via a steep path, past the  
little chapel of Agios Geórgios and continue on the rim of the caldera, to  
get to Skaros rock. The main path will eventually take you to the outskirts  
of the village and to a cobblestone path on the edge of the caldera, passing  
by some more luxurious hotels. After the hotel Pantheon you will see a fork  
in the road. If you have time you can take a detour by taking the path on the  
left that leads to the church of Agios Antonios Atragoudistos, a nice spot  
for a stop. The church has been carved into the rock and the walls of the  
volcano inside the church has frescoes dating back to 1750. The church was  
built around 1050, and according to tradition an icon of Saint Anthony from  
the church of Agios Markos would constantly disappear from inside the church  
and appear at the spot where Agios Antonios is now. The locals took it as a  
sign to build the church. Keeping on the path we will see another church  
ahead on your left, called Agios Markos and after a few meters you will come  
to a fork in the road with the church of Profitis Ilias in front of you. On  
the right you will get a easier, wider, paved path without much of a view, on  
the left you will come to a narrow, dirt path with a magnificent view. Both  
paths lead to the same spot in the end, in front of the hotels Aenaon Villas  
and San Antonio. \n" +
```

```
"Now you will have to walk on the left side of the asphalt  
road for a bit, at an area called Vounia, try to walk a bit quickly on this
```

part and you will arrive at a canteen (food truck). The path that begins on the left is covered with pumice and lava stones and later turns to grayish ash justifying the area's name of Black Mountain - Mavro Vouno. You will pass the churches of Panagia and Stavros and the dirt road will be black or red from the volcanic pebbles. Then you will arrive at another church that is also called Profitis Ilias. Stay on the path to the left - the right path leads to the small village of Finikia. Soon, you will arrive at the desalination company of Ia. Here you can follow the paved street besides the asphalt road, then the road itself, and you finally arrive in the centre of Oia. \n" +

"\n" +

"\n" +

"Από τα Φηρά μέχρι το Φηροστεφάνι, το Ημεροβίγλι και την

Οία\n" +

"\n" +

"Η πεζοπορία που διαλέγουν να κάνουν οι περισσότεροι από τους επισκέπτες της Σαντορίνης, οδηγεί τους πεζοπόρους από την πρωτεύουσα του νησιού τα Φηρά στην Οία κατά μήκος της χείλους της Καλντέρας, περνώντας από τα χωριά Φηροστεφάνι και Ημεροβίγλι. Η διαδρομή προσφέρει υπέροχη θέα στο ηφαίστειο και στο Αιγαίο που αλλάζει καθώς περνάτε το βόρειο τμήμα της καλντέρας. Ανάλογα με τη φυσική σας κατάσταση και πόσες στάσεις θα κάνετε για φωτογραφίες, το μονοπάτι μπορεί να διαρκέσει από 2,5 ώρες έως 5 ώρες. Το μονοπάτι μπορεί επίσης να ξεκινήσει από την Οία και να κατευθυνθεί προς τα Φηρά, με το ίδιο επίπεδο δυσκολίας αφού οι υψομετρικές διαφορές είναι ασήμαντες. Με σωστό προγραμματισμό μπορείτε να βρεθείτε κοντά στην Οία την ώρα του ηλιοβασιλέματος όπου μπορείτε να ξεκουραστείτε και να απολαύσετε την θέα σε ένα λιγότερο συνωστισμένο σημείο λίγο πριν την είσοδο στο χωριό, στην εκκλησία της Παναγίας. Συνιστάται να φέρετε νερό μαζί και να φορέσετε κλειστά παπούτσια, καθώς μέρος της διαδρομής είναι γεμάτο από μικρά ηφαιστειακά βότσαλα, που πάντα βρίσκουν τρόπο να μπουν ανάμεσα από σανδάλια και πατούσες. Έχετε επίσης κατά νου ότι η χρήση λεωφορείου ή η εύρεση ενός ταξί αμέσως μετά το ηλιοβασίλεμα στην Οία μπορεί να αποτελέσει πρόκληση καθώς στο σημείο

γίνεται συνωστισμός.\n" +

"\n" +

"Ξεκινήστε το μονοπάτι λίγα μέτρα παραπέρα από τους σταθμούς λεωφορείων και ταξί στα Φηρά, απέναντι από την πολυσύχναστη πλατεία Θεοτοκόπουλου, όπου βρίσκονται το βιβλιοπωλείο, η τράπεζα και κάποιες καφετέριες. Πάρτε τον πλακόστρωτο δρόμο μεταξύ του καταστήματος με τα οπτικά και του ταξιδιωτικού πρακτορείου Pelican Travel, κατευθυνθείτε προς τα πάνω και στρίψτε δεξιά μετά το Hondos Center. Συνεχίστε να ανεβαίνετε στον δρόμο αυτό προσπερνώντας τα νυχτερινά κέντρα και τα καταστήματα μέχρι να φτάσετε στο Αρχαιολογικό Μουσείο, όπου θα πρέπει να στρίψετε αριστερά. Θα περάσετε μπροστά από την είσοδο στο τελεφερίκ και στη γωνία, εκεί όπου βρίσκεται το εστιατόριο Zaphora θα πρέπει να στρίψτε δεξιά. Συνεχίστε να ανεβαίνετε, κατά μήκος του χείλους της καλντέρας, στρίβοντας αριστερά μετά το καφενείο της Ειρήνης και περνώντας μπροστά από το κοκκινωπό παλιό αρχοντικό, ένα νεοκλασικό κτίριο που στεγάζει το Συνεδριακό Κέντρο Νομικού. Συνεχίστε να ακολουθείτε τα πλακόστρωτα σκαλοπάτια που οδηγούν στο Φηροστεφάνι. Στην εκκλησία του Αγίου Γερασίμου συνεχίστε ακολουθώντας το μονοπάτι που σας μεταφέρει από την άκρη της καλντέρας σε έναν πιο πολυσύχναστο δρόμο στο κέντρο του χωριού. Περπατήστε σε αυτό το μονοπάτι καθώς σας βγάζει από το χωριό σε ένα μικρό χωματόδρομο και στη συνέχεια γίνεται πάλι πλακόστρωτο μονοπάτι που σας φέρνει στο μοναστήρι του Αγίου Νικολάου. Συνεχίστε στο μονοπάτι και μπειτε στο Ημεροβίγλι. Στρίψτε αριστερά στο πρώτο μονοπάτι που θα δείτε μετά το ξενοδοχείο Chromata. Προσπαθήστε να μείνετε στα σοκάκια στα αριστερά και θα βλέπετε διάφορα πολυτελή ξενοδοχεία και βίλες και την θέα της καλντέρας. Η αυλή της εκκλησίας της Αναστάσεως είναι μια ωραία στάση για κάποιες φωτογραφίες και έπειτα μεταβείτε στην καφετέρια Blue Note, από όπου μπορείτε να κάνετε μια παράκαμψη για να περπατήσετε στον βράχο του Σκάρου, ένα από τα πιο εντυπωσιακά σημεία της Σαντορίνης. Απλά στρίβετε αριστερά και αρχίζετε να κατεβαίνετε σε ένα απότομο μονοπάτι, περνώντας από το μικρό παρεκκλήσι του Αγίου Γεωργίου, για να φτάσετε στο βράχο του Σκάρου. Το κύριο μονοπάτι θα σας οδηγήσει τελικά στα περίχωρα του χωριού σε ένα λιθόστρωτο

μονοπάτι στην άκρη της καλντέρας, περνώντας από κάποια ακόμα πιο πολυτελή ξενοδοχεία. Μετά το ξενοδοχείο Πάνθεον θα δείτε μια διακλάδωση στο δρόμο. Αν έχετε χρόνο μπορείτε να κάνετε μια παράκαμψη στο μονοπάτι στα αριστερά που οδηγεί στην εκκλησία του Αγίου Αντωνίου του Ατραγούδιστου, ένα ωραίο σημείο για μια στάση. Η εκκλησία είναι σκαλισμένη στο βράχο και τα τοιχώματα του ηφαιστείου μέσα στην εκκλησία έχουν τοιχογραφίες που χρονολογούνται από το 1750. Η εκκλησία χτίστηκε γύρω στο 1050 και σύμφωνα με την παράδοση μια εικόνα του Αγίου Αντωνίου από την εκκλησία του Αγίου Μάρκου εξαφανιζόταν συνεχώς από μέσα στην εκκλησία και εμφανίζονται στο σημείο που βρίσκεται σήμερα ο Άγιος Αντώνιος. Οι ντόπιοι το πήραν ως σημάδι για να χτίσουν την εκκλησία. Παραμένοντας στο μονοπάτι θα δείτε την εκκλησία του Αγίου Μάρκου στα αριστερά και μετά από μερικά μέτρα θα έρθετε σε μια διακλάδωση στο δρόμο με την εκκλησία του Προφήτη Ηλία μπροστά σας. Μπορείτε να επιλέξετε ανάμεσα στα 2 μονοπάτια, στα δεξιά βρίσκετε ένα ευκολότερο, ευρύτερο, πλακόστρωτο μονοπάτι χωρίς ιδιαίτερη θέα, στα αριστερά είναι ένα στενός χωματόδρομος με υπέροχη θέα. Και τα δύο μονοπάτια οδηγούν στο ίδιο σημείο στο τέλος, μπροστά από τα ξενοδοχεία Aenaon Villas και San Antonio. Τώρα θα πρέπει να περπατήσετε στην αριστερή πλευρά του κεντρικού δρόμου για λίγο, σε μια περιοχή που ονομάζεται Βουνιά, προσπαθήστε να περπατήσετε λίγο γρήγορα σε αυτό το σημείο και σύντομα θα φτάσετε σε μια καντίνα. Το μονοπάτι που αρχίζει στα αριστερά καλύπτεται με πέτρες από ελαφρόπετρα και λάβα και αργότερα μετατρέπεται σε γκριζωπή τέφρα, έτσι δεν είναι περίεργο ότι η περιοχή εδώ ονομάζεται «Μαύρο Βούνιο». Θα περάσετε τις εκκλησίες της Παναγίας και του Σταυρού και ο χωματόδρομος θα είναι μαύρος ή κόκκινος από τα ηφαιστειακά βότσαλα. Στη συνέχεια θα φτάσετε σε μια ακόμη εκκλησία του Προφήτη Ηλία. Μείνετε στο μονοπάτι προς τα αριστερά - το μονοπάτι στα δεξιά οδηγεί στο μικρό χωριό της Φοινικιάς. Σύντομα, θα φτάσετε στην εταιρεία αφαλάτωσης της Οίας. Εδώ μπορείτε να ακολουθήσετε τον πλακόστρωτο δρόμο δίπλα στον κεντρικό δρόμο, μετά θα περπατήσετε για λίγο στον κεντρικό δρόμο και φτάνετε τελικά στο κέντρο της Οίας.

```

\n" +
"\n')," +
"('2','Kamari to Ancient
Thira','2.7','284','48','2','36.367549','25.474222','37.2776','25.479200','Th
e trail begins in Kamari, just below the cliff. When looking at the mountain
you will see the long winding paved road that takes you to Ancient Thira by
car. Our route starts on a path, a bit further on the right from the paved
road. There are signs that mention the route and that it is also the way to
get to a rock climbing area. The path quickly reaches the church called
Zodoxos Pigi - a name connected to Santorinis own... life source-a spring in a
cave with holy water. This is a nice stop, as you can rest at the stone table
and cool down in the shade under trees or in the cave. Inside the cave you
can drink water, admire the stalactites and the stone basins the water has
created. Also check out ancient... graffiti tags - personal names from the
Archaic period that have been carved on the stones surrounding the spring.
The water of the cave has been used since ancient times and ancient carvings
in the stones indicate that work was done so the water could reach the front
of the cave. Research in the early 1900s revealed the existence of an ancient
mine that was exploited by the inhabitants of Ancient Thira in a second cave
some meters away. Also it is stated that during the period of religious
persecutions, this space was used as a hiding place by the Christians of that
time. Locals celebrate the church once a year and light tea-light candles
inside the cave, adding to its ambience. The natural spring in the cave like
opening of the limestone bedrock of Profitis Hlias was the main water source
for the city of ancient Thera as well as for the entire island. The cave
which measures 26 m long is known as Zodoxos Pigi (Life giving spring) after
the chapel of the same name situated at its entrance. In antiquity, the
spring was connected to Ancient Thira via the main road leaving the city
towards the north. Parts of this ancient way survive today, slightly lower
than the modern path which passes in front of the spring. The ancient road
continued further along the eastern slope of Profitis Hlias, to an extensive
cemetery with stone cut tombs. It is likely that a second branch of the
ancient road split off near the spring and traveled down to the northern
harbor of the city, known as ancient Oia at the foot of Mesa Vouno. The water

```

```

of the spring drips from the ceiling of the cave into a natural basin created
by the spring water itself. Traces of ancient cuttings in the rock of the
cave indicate that water management works were carried out. Among these a
channel, the lower part of which survives today of 0.70-0.80 m takes the
water from the spring to the front of the cave. It is indeed likely that
there was also some kind of construction underneath the modern chapel.
Although the quantity of water supplied by the spring was most likely not
sufficient to cover the needs of the inhabitants of the city, it was in
continuous use during antiquity. It appears that the spring was also used, in
ancient times as well as modern, as a place for travelers on foot to stop and
rest. Numerous inscriptions of the Archaic period, bearing personal names,
were carved into the smooth surface of stones surrounding the spring.')

```

```

public trail get_trail(int id) {
    try {
        SQLiteDatabase db = this.getReadableDatabase();

        Cursor cursor = db
            .query("trail", new String[]{"_id", "name", "length",
"height", "minutes", "difficulty",
"startlongitude", "startlatitude",
"endlongitude", "endlatitude", "description"},
                "_id" + "=?", new String[]{String.valueOf(id)},
                null, null, null, null);
        if (cursor != null)
            cursor.moveToFirst();

        trail itrail = new trail();
        itrail.set_id(cursor.getInt(cursor.getColumnIndex("_id")));
        itrail.set_name(cursor.getString(cursor
            .getColumnIndex("name")));
        itrail.set_length(cursor.getInt(cursor
            .getColumnIndex("length")));
        itrail.set_height(cursor.getInt(cursor
            .getColumnIndex("height")));
        itrail.set_minutes(cursor.getInt(cursor
            .getColumnIndex("minutes")));
        itrail.set_difficulty(cursor.getInt(cursor
            .getColumnIndex("difficulty")));
        itrail.set_startlatitude(cursor.getDouble(cursor
            .getColumnIndex("startlatitude")));
        itrail.set_startlongitude(cursor.getDouble(cursor
            .getColumnIndex("startlongitude")));
        itrail.set_endlatitude(cursor.getDouble(cursor
            .getColumnIndex("endlatitude")));
        itrail.set_endlongitude(cursor.getDouble(cursor
            .getColumnIndex("endlongitude")));
        itrail.set_description(cursor.getString(cursor
            .getColumnIndex("description")));

        cursor.close();
        db.close();
        Log.d("My.DB", "Fetched trail :" + itrail.get_id());
        return itrail;
    } catch (Exception e) {
        return null;
    }
}

public List<trail> getAlltrails(String where) {
    List<trail> itrailList = new ArrayList<trail>();

    String selectQuery = "SELECT * FROM " + "trail" + where;

    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(selectQuery, null);

    if (cursor.moveToFirst()) {
        do {
            trail itrail = new trail();
            itrail.set_id(cursor.getInt(cursor.getColumnIndex("_id")));
            itrail.set_name(cursor.getString(cursor
                .getColumnIndex("name")));

```



```

        itrail.set_length(cursor.getInt(cursor
            .getColumnIndex("length")));
        itrail.set_height(cursor.getInt(cursor
            .getColumnIndex("height")));
        itrail.set_minutes(cursor.getInt(cursor
            .getColumnIndex("minutes")));
        itrail.set_difficulty(cursor.getInt(cursor
            .getColumnIndex("difficulty")));
        itrail.set_startlatitude(cursor.getDouble(cursor
            .getColumnIndex("startlatitude")));
        itrail.set_startlongitude(cursor.getDouble(cursor
            .getColumnIndex("startlongitude")));
        itrail.set_endlatitude(cursor.getDouble(cursor
            .getColumnIndex("endlatitude")));
        itrail.set_endlongitude(cursor.getDouble(cursor
            .getColumnIndex("endlongitude")));
        itrail.set_description(cursor.getString(cursor
            .getColumnIndex("description")));
        itrailList.add(itrail);
    } while (cursor.moveToNext());
}

Log.d("My.DB", "Fetched " + cursor.getCount() + " trails");
cursor.close();
db.close();

return itrailList;
}

public void deletetrail(int id) {
    SQLiteDatabase db = this.getWritableDatabase();
    db.delete("trail", "_id" + " = ?",
        new String[]{String.valueOf(id)});
    db.close();
    Log.d("My.DB", "Deleted trail :" + id);
}

public int updatetrail(trail itrail) {
    SQLiteDatabase db = this.getWritableDatabase();

    ContentValues values = new ContentValues();
    values.put("name", itrail.get_name());

    values.put("length", itrail.get_length());

    values.put("height", itrail.get_height());

    values.put("minutes", itrail.get_minutes());

    values.put("difficulty", itrail.get_difficulty());
    values.put("startlongitude", itrail.get_startlongitude());
    values.put("startlatitude", itrail.get_startlatitude());
    values.put("endlongitude", itrail.get_endlongitude());
    values.put("endlatitude", itrail.get_endlatitude());
    values.put("description", itrail.get_description());
    // updating row
    int res = db.update("trail", values, "_id" + " = ?",
        new String[]{String.valueOf(itrail.get_id())});

    db.close();
}

```

```

Log.d("My.DB", "Updated trail :" + itrail.get_id());
return res;
}

public void add_comment(comment icomment) {

    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put("_id", icomment.get_id());
    values.put("createdtime", icomment.get_createdtime_forSQL());
    values.put("trail_id", icomment.get_trail_id());
    values.put("description", icomment.get_description());

    // Inserting Row
    db.insert("comment", null, values);
    db.close(); // Closing database connection
    Log.d("My.DB", "Added comment " + icomment.get_id());
}

public comment get_comment(int id) {
    try {
        SQLiteDatabase db = this.getReadableDatabase();

        Cursor cursor = db
            .query("comment", new String[]{"_id", "trail_id",
"createdtime", "description"},
                "_id" + "=?", new String[]{String.valueOf(id)},
                null, null, null, null);
        if (cursor != null)
            cursor.moveToFirst();

        comment icomment = new comment();
        icomment.set_id(cursor.getInt(cursor.getColumnIndex("_id")));
        icomment.set_trail_id(cursor.getInt(cursor
            .getColumnIndex("trail_id")));
        icomment.set_createdtime_forSQL(cursor.getString(cursor
            .getColumnIndex("createdtime")));
        icomment.set_description(cursor.getString(cursor
            .getColumnIndex("description")));

        cursor.close();
        db.close();
        Log.d("My.DB", "Fetched comment :" + icomment.get_id());
        return icomment;
    } catch (Exception e) {
        return null;
    }
}

public List<comment> getAllcomments(String where) {
    List<comment> icommentList = new ArrayList<comment>();

    String selectQuery = "SELECT * FROM " + "comment" + where;

    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(selectQuery, null);

    if (cursor.moveToFirst()) {
        do {
            comment icomment = new comment();
            icomment.set_id(cursor.getInt(cursor.getColumnIndex("_id")));
            icomment.set_trail_id(cursor.getInt(cursor
                .getColumnIndex("trail_id")));

```

Ανάπτυξη εφαρμογής περιήγησης σε μονοπάτια «SANTORINI TRAIL» για Android

```
        icomment.set_createdtime_forSQL(cursor.getString(cursor
            .getColumnIndex("createdtime")));
        icomment.set_description(cursor.getString(cursor
            .getColumnIndex("description")));

        icommentList.add(icomment);

    } while (cursor.moveToNext());
}

Log.d("My.DB", "Fetched " + cursor.getCount() + " comment");
cursor.close();
db.close();
return icommentList;
}

public int getCommentNextId() {
    try {

        String selectQuery = "SELECT ifnull(max(_id)+1,1) maxid FROM
comment";
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor cursor = db.rawQuery(selectQuery, null);

        if (cursor != null)
            cursor.moveToFirst();
        int maxid = (cursor.getInt(cursor.getColumnIndex("maxid")));

        cursor.close();
        db.close();

        return maxid;
    } catch (Exception e) {
        return 1;
    }
}

public void deletecomment(int id) {
    SQLiteDatabase db = this.getWritableDatabase();
    db.delete("comment", "_id" + " = ?",
        new String[]{String.valueOf(id)});
    db.close();
    Log.d("My.DB", "Deleted comment :" + id);
}

public void add_completedTr(completedTr icompletedTr) {

    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put("_id", icompletedTr.get_id());
    values.put("timecompleted", icompletedTr.get_timecompleted_forSQL());
    values.put("trail_id", icompletedTr.get_trail_id());
    values.put("minutes", icompletedTr.get_minutes());

    // Inserting Row
    db.insert("completedTr", null, values);
}
```

Ανάπτυξη εφαρμογής περιήγησης σε μονοπάτια «SANTORINI TRAIL» για Android

```
db.close(); // Closing database connection
Log.d("My.DB", "Added completedTr " + icompletedTr.get_id());
}

public completedTr get_completedTr(int id) {
    try {
        SQLiteDatabase db = this.getReadableDatabase();

        Cursor cursor = db
            .query("completedTr", new String[]{"_id", "trail_id",
"timecompleted", "minutes"},
                "_id" + "=?", new String[]{String.valueOf(id)},
                null, null, null, null);
        if (cursor != null)
            cursor.moveToFirst();

        completedTr icompletedTr = new completedTr();
        icompletedTr.set_id(cursor.getInt(cursor.getColumnIndex("_id")));
        icompletedTr.set_trail_id(cursor.getInt(cursor
            .getColumnIndex("trail_id")));
        icompletedTr.set_timecompleted_forSQL(cursor.getString(cursor
            .getColumnIndex("timecompleted")));
        icompletedTr.set_minutes(cursor.getInt(cursor
            .getColumnIndex("minutes")));

        cursor.close();
        db.close();
        Log.d("My.DB", "Fetched completedTr :" + icompletedTr.get_id());
        return icompletedTr;
    } catch (Exception e) {
        return null;
    }
}

public List<completedTr> getAllcompletedTr(String where) {
    List<completedTr> icompletedTrList = new ArrayList<completedTr>();

    String selectQuery = "SELECT * FROM " + "completedTr" + where;

    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(selectQuery, null);

    if (cursor.moveToFirst()) {
        do {
            completedTr icompletedTr = new completedTr();

            icompletedTr.set_id(cursor.getInt(cursor.getColumnIndex("_id")));
            icompletedTr.set_trail_id(cursor.getInt(cursor
                .getColumnIndex("trail_id")));
            icompletedTr.set_timecompleted_forSQL(cursor.getString(cursor
                .getColumnIndex("timecompleted")));
            icompletedTr.set_minutes(cursor.getInt(cursor
                .getColumnIndex("minutes")));

            icompletedTrList.add(icompletedTr);

        } while (cursor.moveToNext());
    }

    Log.d("My.DB", "Fetched " + cursor.getCount() + " completedTr");
    cursor.close();
    db.close();
}
```

```

        return icompletedTrList;
    }

    public int getCompletedTrNextId() {
        try {

            String selectQuery = "SELECT ifnull(max(_id)+1,1) maxid FROM
completedTr";
            SQLiteDatabase db = this.getWritableDatabase();
            Cursor cursor = db.rawQuery(selectQuery, null);

            if (cursor != null)
                cursor.moveToFirst();
            int maxid = (cursor.getInt(cursor.getColumnIndex("maxid")));

            cursor.close();
            db.close();

            return maxid;
        } catch (Exception e) {
            return 1;
        }
    }

    public void deletecompletedTr(int id) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete("completedTr", "_id" + " = ?",
            new String[]{String.valueOf(id)});
        db.close();
        Log.d("My.DB", "Deleted completedTr :" + id);
    }

}

public class BaseDialogFragment extends DialogFragment {
    public interface BaseDialogInterface{
        public void onPositiveClick (DialogFragment dialog);
        public void onNegativeClick (DialogFragment dialog);
    }
    BaseDialogInterface myInterface;

    @Override
    public void onAttach(Activity activity){
        super.onAttach(activity);
        try{
            myInterface=(BaseDialogInterface) activity;
        }
        catch(Exception e){}
    }

}

public class completetraildialog extends BaseDialogFragment {
    public Date selecteddate ;
    public int selectedtime;
    private View view;

```

```

@Override
public Dialog onCreateDialog(Bundle savedInstanceState) {
    LayoutInflater inflater = getActivity().getLayoutInflater();
    view = (inflater.inflate(R.layout.complete_dialog, null));
    return new AlertDialog.Builder(getActivity())
        // setTitle(R.string.library_materialdrawer_licenseId)
        .setView(view)
        .setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
myInterface.onNegativeClick(completetraildialog.this);
            }
        })
        .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                DatePicker dp =
(DatePicker) view.findViewById(R.id.select_date);
                selecteddate = new
Date(dp.getYear(), dp.getMonth(), dp.getDayOfMonth());
                // EditText edtxt =
(EditText) view.findViewById(R.id.CompletedTime);
                // selectedtime =
Integer.valueOf(edtxt.getText().toString());
myInterface.onPositiveClick(completetraildialog.this);
            }
        })
        .create();
}
}

public class confirmdialog extends BaseDialogFragment {
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        return new AlertDialog.Builder(getActivity())
            //
            .setTitle(R.string.library_materialdrawer_licenseId)
            // setMessage(R.string.library_AndroidIconics_licenseId)
            .setTitle("")
            .setMessage("")
            .setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int which) {
myInterface.onNegativeClick(confirmdialog.this);
                }
            })
            .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int which) {
myInterface.onPositiveClick(confirmdialog.this);
                }
            })
            .create();
    }
}
}

```

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] <https://www.androidauthority.com/what-is-android-328076/>
- [2] <https://www.techopedia.com/definition/5415/android>
- [3] https://en.wikipedia.org/wiki/Android_Runtime
- [4] <https://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>
- [5] https://en.wikipedia.org/wiki/Mobile_app
- [6] <https://www.techopedia.com/definition/2953/mobile-application-mobile-app>
- [7] <https://medium.com/true-enterprise-mobility/the-fourth-industrial-revolution-is-mobile-24bfb7b842b1>
- [8] <http://gs.statcounter.com/os-market-share/mobile/worldwide>
- [9] <https://developer.android.com/guide/components/fundamentals>
- [10] <https://developer.android.com/training/basics/supporting-devices/platforms>
- [11] <https://developer.android.com/about/dashboards/>
- [12] <https://developer.android.com/training/multiscreen/screensizes>
- [13] https://developer.android.com/guide/practices/screens_support
- [14] <https://developer.android.com/training/basics/supporting-devices/languages.html>
- [15] <https://developer.android.com/guide/topics/resources/localization.html>
- [16] <https://developer.android.com/distribute/best-practices/launch/localization-checklist>
- [17] <https://developer.android.com/docs/quality-guidelines/core-app-quality>
- [18] https://en.wikipedia.org/wiki/Global_Positioning_System
- [19] <https://www.landmark.com.gr/βασικές-αρχές-gps/βασική-θεωρία>
- [20] <https://www.androidcentral.com/how-does-gps-work-my-phone>
- [21] <https://www.techopedia.com/definition/24218/assisted-gps-a-gps>

- [22] <https://themanifest.com/app-development/7-examples-location-based-services-apps>
- [23] <https://geoawesomeness.com/expected-trends-technological-advancements-location-based-services/>
- [24] <https://themanifest.com/app-development/location-tracking-mobile-apps>
- [25] <https://material.io/design/communication/launch-screen.html#branded-launch>
- [26] <https://mikepenz.github.io/MaterialDrawer/>
- [27] <https://developer.android.com/docs/quality-guidelines/core-app-quality#tests>
- [28] <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>
- [29] https://en.wikipedia.org/wiki/Usability_testing