



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Σπύρου Νικολάκη

Εφαρμογή Android για παρακολούθηση πυρκαγιών

Επιβλέπων:

Δ. Μιχαήλ

Μέλη Επιτροπής:

Γ. Δημητρακόπουλος, Κ. Τσερπές

Αθήνα 2015

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον κ. Μιχαήλ για τις συμβουλές, την καθοδήγηση και την υπομονή κατά την εκπόνηση της πτυχιακής μου εργασίας.

Ευχαριστώ τον κ. Τσερπέ και τον κ. Δημητρακόπουλο για τη στήριξη και τη βοήθεια.

Τέλος, ευχαριστώ την οικογένεια μου και τους φίλους μου που με υποστηρίζουν διαρκώς σε ό,τι κι αν επιχειρώ και βρίσκονται πάντοτε δίπλα μου για να με ενθαρρύνουν.

Πίνακας Περιεχομένων

ΠΕΡΙΛΗΨΗ.....	8
ABSTRACT.....	9
ΚΕΦΑΛΑΙΟ 1: Εισαγωγή.....	10
ΚΕΦΑΛΑΙΟ 2: Android Platform.....	12
2.1. Γενικά Στοιχεία.....	12
2.1.1. Συνοπτική Ιστορία.....	12
2.2. Τεχνικά Στοιχεία.....	13
2.2.1. Android Software Stack.....	13
2.2.2. Πυρήνας Linux.....	13
2.2.3. Σύστημα Καταγραφής Android.....	15
2.2.4. Βιβλιοθήκες.....	15
2.2.5. Android Runtime.....	16
2.2.6. Application Framework.....	17
2.2.7. Εφαρμογές.....	18
2.2.8. Κύκλος Ζωής Δραστηριότητας.....	20
2.2.9. Κύκλος Ζωής Αντικειμένου (Fragment).....	22
2.2.10. Βασικά στοιχεία λειτουργίας μια τυπικής εφαρμογής Android.....	24
ΚΕΦΑΛΑΙΟ 3: Fire Monitoring Service.....	26
3.1. Βασικά Στοιχεία.....	26
3.2. Τεχνικές Λεπτομέρειες.....	27
3.3. Βάση Δεδομένων.....	28
ΚΕΦΑΛΑΙΟ 4: Εφαρμογή Fire.....	29
4.1. Βασικά Στοιχεία.....	29
4.2. Κύρια Δραστηριότητα (MainActivity).....	30

4.3. Λίστα Πυρκαγιών (FiresList).....	32
4.4. Δραστηριότητα LocateFire.....	35
4.5. Χάρτης Πρόσφατων Πυρκαγιών (RecentFires).....	36
4.6. Δυνατότητα Συναγερμού (Alert).....	38
4.7. Μενού Επιλογών.....	40
4.7.1. Δραστηριότητα Αναζήτησης.....	41
4.7.2. Δραστηριότητα Ρυθμίσεων Χρήστη.....	45
4.8. Βοηθητικές Κλάσεις.....	48
4.8.1. Έλεγχος Σύνδεσης Διαδικτύου (CheckInternetConnection).....	48
4.8.2. Εύρεση Τοποθεσίας (LocationFinder).....	48
4.8.3. Σύνδεση με Server (ServerConnection).....	49
4.9. Υπηρεσία FiresCache.....	49
4.9.1. Γενική Λειτουργία.....	49
4.9.2. Μέθοδοι Μεταφοράς Δεδομένων από τη Λίστα Πυρκαγιών.....	51
4.9.3. Συγχρονισμός μεταξύ Ανάγνωσης και Γραφής της Λίστας.....	52
4.10. Απαιτούμενα Δικαιώματα Εφαρμογής και άλλα Χαρακτηριστικά του AndroidManifest.xml	53
4.11. Υλοποίηση Server.....	55
4.11.1. Αποστολή Αποτελεσμάτων στην Εφαρμογή.....	55
4.11.2. Αποθήκευση Φωτογραφίας.....	57
ΚΕΦΑΛΑΙΟ 5: Συμπεράσματα.....	58
5.1. Γενικά Συμπεράσματα.....	58
5.2. Σκέψεις για Μελλοντικές Εκδόσεις.....	58
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	60

Πίνακας Σχημάτων

Σχήμα 1: Η αρχιτεκτονική του Android (πηγή: wikipedia.org).....	13
Σχήμα 2: Ο κύκλος ζωής μιας δραστηριότητας.....	22
Σχήμα 3: Ο κύκλος ζωής ενός αντικειμένου (Fragment).....	24
Σχήμα 4: Τα βασικότερα μέρη της εφαρμογής Fire.....	30
Σχήμα 5: Η κύρια δραστηριότητα και τα άμεσα συνδεδεμένα με αυτή μέρη της εφαρμογής.....	31
Σχήμα 6: Η λίστα των πυρκαγιών και τα συνδεδεμένα με αυτή μέρη.....	33
Σχήμα 7: Η δραστηριότητα LocateFire και οι σχέσεις της με τα υπόλοιπα μέρη της εφαρμογής.....	35
Σχήμα 8: Το αντικείμενο του χάρτη πρόσφατων πυρκαγιών.....	37
Σχήμα 9: Το αντικείμενο του συναγερμού και οι σχέσεις του με τα υπόλοιπα μέρη της εφαρμογής.....	39
Σχήμα 10: Τι συμβαίνει όταν ο χρήστης πατά το κουμπί για να ορίσει το κάθε κριτήριο.....	44
Σχήμα 11: Η πλήρης λειτουργία της αναζήτησης.....	45
Σχήμα 12: Οι απαραίτητοι διάλογοι για τις ρυθμίσεις.....	50
Σχήμα 13: Από τι χρησιμοποιείται η κλάση "CheckInternetConnection"	52
Σχήμα 14: Ποια μέρη και πότε συνδέονται με την υπηρεσία.....	54
Σχήμα 15: Πώς και πότε επικοινωνεί η "FiresCache" με τα αρχεία και τη λίστα των πυρκαγιών της	55

Πίνακας Αποσπασμάτων Κώδικα

Απόσπασμα Κώδικα 1: Η μέθοδος onItemClick στο αντικείμενο της λίστας των πυρκαγιών.....	34
Απόσπασμα Κώδικα 2: Κλήση της δραστηριότητας "SetGeometry" με σκοπό την επιστροφή ενός αποτελέσματος.....	43
Απόσπασμα Κώδικα 3: Επιστροφή της τοποθεσίας που επέλεξε ο χρήστης.....	43
Απόσπασμα Κώδικα 4: Η υπογραφή της μεθόδου που επιστρέφει όλες τις πυρκαγιές της λίστας...	52
Απόσπασμα Κώδικα 5: Η υπογραφή της μεθόδου που επιστρέφει τις πυρκαγιές της τελευταίας ενημέρωσης.....	52
Απόσπασμα Κώδικα 6: Η υπογραφή της μεθόδου που επιστρέφει μια πυρκαγιά με βάση τη θέση της πυρκαγιάς στη λίστα της "FiresList".....	52
Απόσπασμα Κώδικα 7: Το κλείδωμα εγγραφής - ανάγνωσης.....	53
Απόσπασμα Κώδικα 8: Τα δικαιώματα για τον έλεγχο της σύνδεσης στο διαδίκτυο αλλά και για την επίτευξη της σύνδεσης αυτής.....	54
Απόσπασμα Κώδικα 9: Το δικαίωμα πρόσβασης στις υπηρεσίες της Google.....	54
Απόσπασμα Κώδικα 10: Το δικαίωμα για γραφή και ανάγνωση στην κάρτα SD.....	54
Απόσπασμα Κώδικα 11: Τα δικαιώματα που απαιτούνται για τον εντοπισμό της τοποθεσίας της συσκευής.....	55
Απόσπασμα Κώδικα 12: Η απαίτηση για την ύπαρξη του "OpenGL ES version 2".....	55
Απόσπασμα Κώδικα 13: Παράδειγμα μιας πυρκαγιάς σε κατανοητή για την εφαρμογή JSON μορφή	57

Ευρετήριο Εικόνων

Εικόνα 1: Η λίστα με τις πυρκαγιές.....	26
Εικόνα 2: Ο χάρτης με τις πυρκαγιές και κάποια στατιστικά στοιχεία.....	27
Εικόνα 3: Το αντικείμενο “FiresList” σε εκτέλεση.....	34
Εικόνα 4: Η “LocateFire” εν δράσει: μια πυρκαγιά στο δήμο Φαρσάλων.....	36
Εικόνα 5: Απεικόνιση πυρκαγιών ως πολύγωνα στο χάρτη της Google.....	38
Εικόνα 6: Το αντικείμενο "Alert"	40
Εικόνα 7: Η δραστηριότητα της αναζήτησης, όπου ο χρήστης επιλέγει τα κριτήρια αναζήτησης....	44
Εικόνα 8: Τα αποτελέσματα της αναζήτησης.....	45
Εικόνα 9: Η δραστηριότητα των ρυθμίσεων.....	47

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία, με τίτλο “Εφαρμογή Android για Παρακολούθηση Πυρκαγιών” στοχεύει στη δημιουργία μιας εφαρμογής για τη δημοφιλέστερη πλατφόρμα έξυπνων κινητών που θα ενημερώνει το χρήστη σχετικά με τις πυρκαγιές που είναι ενεργές στην Ελλάδα, ανεξάρτητα από το πού βρίσκεται. Η εφαρμογή αυτή έχει βασιστεί στη διαδικτυακή υπηρεσία “Fire Monitoring Service” του Εθνικού Αστεροσκοπείου Αθηνών.

Αρχικά, μελετώνται τα βασικότερα στοιχεία της πλατφόρμας Android και όλα τα απαραίτητα μέρη που χρειάζεται μια τυπική εφαρμογή προκειμένου να λειτουργήσει.

Στη συνέχεια, αναλύεται η λειτουργία της διαδικτυακής υπηρεσίας “Fire Monitoring Service” κι επεξηγούνται τα δεδομένα που υπάρχουν στη βάση δεδομένων της υπηρεσίας. Αυτό βοηθά στην καλύτερη κατανόηση των απαραίτητων για την ορθή λειτουργία της εφαρμογής δεδομένων.

Έπειτα, αναλύεται η λειτουργία της δημιουργημένης εφαρμογής, η δομή της και η σύνδεση μεταξύ των επιμέρους στοιχείων της. Μελετάται, επίσης, η επικοινωνία μεταξύ του server της υπηρεσίας και της εφαρμογής καθώς και οι δυνατότητες παραμετροποίησης που προσφέρονται. Επιπλέον, προβάλλεται η μορφή των δεδομένων που περνούν στην εφαρμογή κι εξετάζονται τα αρχεία δεδομένων που διατηρεί στην εξωτερική κάρτα μνήμης της συσκευής.

Αξίζει να σημειωθεί πως η εφαρμογή που δημιουργήθηκε, είναι σε θέση να παρέχει τις λειτουργίες που προσφέρει και η υπηρεσία, μαζί με κάποια επιπλέον στοιχεία παραμετροποίησης και τη δυνατότητα ειδοποίησης της υπηρεσίας για μια πυρκαγιά.

Στο τελευταίο κεφάλαιο, περιλαμβάνονται προτάσεις για μελλοντικές εκδόσεις της εφαρμογής, καθώς και το συμπέρασμα πως η εφαρμογή λειτουργεί ικανοποιητικά με βάση τα ζητήματα που αφορούν τη χρήση του διαδικτύου και της μνήμης της συσκευής.

ABSTRACT

The present thesis entitled "Fire Monitoring Android Application" aims at creating an application for the most popular smartphones platform that will inform the user about the fires that are active in Greece, regardless of the user's location. This application is based on "Fire Monitoring Service" of the National Observatory of Athens.

To begin with, the basic features of the Android platform and all the necessary parts needed for an application to work are studied.

Afterwards, the function of the web service "Fire Monitoring Service" is analyzed and the data stored in the database of the service are explained, which helps in the better understanding of the necessary data for the proper functioning of the application.

Then, the function of the created application, the structure and the connection between the individual parts are analyzed. Moreover, the communication between the server of the service and the application as well as the offered customization options are studied. Additionally, the structure of the data which pass from the server to the application is featured and the data files that are kept on the external memory card of the device are examined.

It is worth mentioning that this application can provide the functions that the service offers too, along with some additional parameterization features and the ability to alert the service about a fire.

In the last chapter recommendations for future versions of the application are included and it is concluded that the application works satisfyingly according to the issues related to the use of the internet and the memory of the device.

ΚΕΦΑΛΑΙΟ 1: Εισαγωγή

Τα τελευταία χρόνια, τα έξυπνα κινητά έχουν γνωρίσει μεγάλη εξάπλωση δημιουργώντας νέες ανάγκες στο καταναλωτικό κοινό. Σύμφωνα με το πρόγραμμα “Pew Research Center’s Internet & American Life Project”, το 56% των ενηλίκων στις Ηνωμένες Πολιτείες διαθέτει ένα έξυπνο κινητό (μετρήσεις Μαΐου 2013) σε αντίθεση με το αντίστοιχο ποσοστό του Μαΐου του 2011 που έφτανε το 35%. (A. Smith, 2013)

Ένα κοινό πρόβλημα που αντιμετώπιζαν τα κινητά των προηγούμενων γενιών (και ακόμη υφίσταται σε μερικές περιπτώσεις) ήταν η δυσκολία πρόσβασης σε διαδικτυακές υπηρεσίες ή ιστοσελίδες που ήταν δημιουργημένες για πρόσβαση μέσω ηλεκτρονικού υπολογιστή. Η ύπαρξη αυτού του προβλήματος οφειλόταν κατά βάση στη μικρή οθόνη των συσκευών και την αργή σύνδεση στο διαδίκτυο που μπορούσε να επιτύχει η συσκευή με τα μέσα που διέθετε.

Αυτά τα προβλήματα, με τη σειρά τους, δημιουργούσαν κάποια επιπλέον πρακτικά προβλήματα. Τέτοια ήταν, για παράδειγμα, η δυσκολία πλοήγησης του χρήστη στην ιστοσελίδα λόγω της μικρής οθόνης και της τεράστιας συγκριτικά ιστοσελίδας και ο μεγάλος χρόνος που απαιτούνταν για την προβολή μιας ιστοσελίδας.

Προκειμένου να επιλυθούν αυτά τα προβλήματα, πολλές ιστοσελίδες άρχισαν να διαθέτουν εκδόσεις για βελτιωμένη προβολή σε φορητές συσκευές. Το δίκτυο που δημιουργήθηκε από αυτές τις εκδόσεις ονομάστηκε “Mobile Web” (Ιστός Κινητής Τηλεφωνίας). Οι ιστοσελίδες που είχαν τη δυνατότητα δημιουργίας μιας έκδοσης για κινητά τηλέφωνα, φυσικά ήταν οι πιο δημοφιλείς, που είχαν μεγάλη επισκεψιμότητα και συνεπώς κάποια οικονομική ευχέρεια ώστε να μπορούν να “ανοιχτούν” προς καινούριες πλατφόρμες. Πολλές μικρότερες ιστοσελίδες μέχρι και σήμερα δε διαθέτουν άλλη έκδοση, καθιστώντας την πλοήγησή σε αυτές από κινητό τηλέφωνο αρκετά δύσκολη.

Καθώς όμως περνούσαν τα χρόνια και η τεχνολογία διαρκώς αναπτυσσόταν, προέκυψαν νέα ζητήματα για τα κινητά τηλέφωνα που πλέον είχαν αρχίσει να γίνονται “έξυπνα”. Στα smartphones, η πρόσβαση σε πολλές ιστοσελίδες μπορούσε να γίνει είτε μέσω ενός περιηγητή διαδικτύου, είτε μέσω της εφαρμογής που διέθετε η ιστοσελίδα για τις πλατφόρμες των έξυπνων κινητών (εφόσον φυσικά τη διέθετε). Κάποιες δημοφιλείς ιστοσελίδες, μάλιστα, προσέφεραν εφαρμογές όχι μόνο για έξυπνα κινητά αλλά και για τους προκατόχους τους, όπως για παράδειγμα έκανε το Facebook. (Bazilian, 2011)

Άλλα τέτοια παραδείγματα αποτελούν εφαρμογές εύκολης πρόσβασης σε κάποια άλλα κοινωνικά δίκτυα κι εφαρμογές ιστοσελίδων ενημέρωσης και ενημέρωσης του καιρού.

Τον πρώτο καιρό που εμφανίστηκαν τα έξυπνα τηλέφωνα, προορίζονταν κατά κύριο λόγο

για επαγγελματίες που χρειάζονταν διαρκή σύνδεση στο διαδίκτυο για επικοινωνία μέσω e-mail, fax κλπ.. Με τον καιρό, τα smartphones άρχισαν να γίνονται περισσότερο δημοφιλή με αποτέλεσμα να προσελκύσουν το ενδιαφέρον και άλλων ανθρώπων μέσω των δυνατοτήτων επικοινωνίας που προσέφεραν.

Στη συνέχεια, λόγω του γενικότερου πληθυσμού που χρησιμοποιούσε τα έξυπνα τηλέφωνα, βρήκαν χώρο στην αγορά και εταιρείες που μέχρι τότε απείχαν από τις τηλεπικοινωνίες. Πιο συγκεκριμένα, το 2007 κυκλοφόρησε το πρώτο iPhone και ένα χρόνο αργότερα το πρώτο κινητό τηλέφωνο με λειτουργικό Android (HTC Dream).

Πλέον, δε χρησιμοποιούν μόνο επαγγελματίες τα smartphones αλλά μια ευρύτερη ομάδα πληθυσμού χωρίς συγκεκριμένα χαρακτηριστικά (μπορεί η χρήση των έξυπνων κινητών να πραγματοποιείται κυρίως από νεότερους ηλικιακά ανθρώπους αλλά γενικότερα χρησιμοποιείται από ανθρώπους όλων των ηλικιών). (A. Smith, 2013) Οι πλατφόρμες των έξυπνων τηλεφώνων είναι ανοιχτές για τη δημιουργία εφαρμογών από τρίτους κατασκευαστές κι έτσι πολλά sites διάφορων σκοπών προσφέρουν εφαρμογές για τις πιο διαδεδομένες πλατφόρμες.

Οι εφαρμογές τέτοιου τύπου στοχεύουν στην καλύτερη εξυπηρέτηση του χρήστη στο site, στη μεγαλύτερη ευχρηστία και στην προβολή των λειτουργιών του site με τέτοιο τρόπο ώστε να είναι πιο εύκολο από το χρήστη να τις εντοπίσει σε μια μικρότερη οθόνη.

Στην παρούσα πτυχιακή εργασία, το αντικείμενο ήταν παρόμοιο. Η υπηρεσία “Fire Monitoring Service” φροντίζει για την ενημέρωση του πληθυσμού, όσον αφορά τις ενεργές πυρκαγιές. Επιπλέον προσφέρει κάποιες άλλες λειτουργίες, όπως αναζήτηση παλαιότερων πυρκαγιών, που έπρεπε να συμπεριληφθούν στην εφαρμογή.

Σκοπός ήταν η εφαρμογή να περιελάμβανε όλες τις σημαντικότερες λειτουργίες της υπηρεσίας και κάποιες επιπρόσθετες που θα ήταν δυνατές, μόνο σε μια φορητή συσκευή που θα έχει τη δυνατότητα σύνδεσης με το διαδίκτυο. Μια τέτοια επιπρόσθετη λειτουργία είναι η δυνατότητα συναγερμού, που δε θα είχε νόημα να υπήρχε στην υπηρεσία για κάποιον σταθερό υπολογιστή που θα χρησιμοποιούσε την υπηρεσία.

Επιπροσθέτως, λόγω της φύσης της επικοινωνίας της εφαρμογής με το διαδίκτυο μέσω του δικτύου κινητής τηλεφωνίας, θα πρέπει να μεταδίδονται όσο το δυνατόν λιγότερα και πιο απαραίτητα δεδομένα.

Με αυτούς τους βασικούς στόχους δημιουργήθηκε η εφαρμογή “Fire” που ως τελικό σκοπό έχει την ενημέρωση για πυρκαγιές στην ευρύτερη περιοχή της Ελλάδας, όπου και αν βρίσκονται, με την απλή προϋπόθεση να μπορεί να επιτευχθεί σύνδεση στο διαδίκτυο με το δίκτυο κινητής τηλεφωνίας ή με χρήση του Wi-Fi.

ΚΕΦΑΛΑΙΟ 2: Android Platform

2.1. Γενικά Στοιχεία

2.1.1. Συνοπτική Ιστορία

Το 2003, ιδρύθηκε η εταιρεία Android Inc. στην Καλιφόρνια από τους Andy Rubin και Rich Miner και μερικά ακόμη μεγάλα ονόματα από το πεδίο των τηλεπικοινωνιών. (Kovach, 2013; Perez, 2007) Στόχος τους, ήταν να δημιουργήσουν ένα λειτουργικό σύστημα για φωτογραφικές μηχανές που να προσφέρει εφαρμογές τρίτων, όπως είχε δηλώσει ο ίδιος ο Rubin. Στη συνέχεια όμως, έκριναν πως η αγορά των ψηφιακών φωτογραφικών μηχανών δεν ήταν πολύ μεγάλη και αποφάσισαν να επικεντρωθούν στα κινητά τηλέφωνα. (Welch, 2013)

Τον Αύγουστο του 2005, η Android Inc. εξαγοράστηκε από την Google, προκειμένου να ενεργοποιηθεί στην αγορά των τηλεπικοινωνιών. (Kovach, 2013) Ένα χρόνο αργότερα, κατασκευάστηκε ένα πρωτότυπο με QWERTY πληκτρολόγιο που έμοιαζε στα έξυπνα κινητά τηλέφωνα της BlackBerry προκειμένου να προσεγγίσει κατασκευαστές τηλεφώνων. Έπειτα, προσέθεσαν οθόνη αφής για να ανταγωνιστεί το iPhone της Apple που θα κυκλοφορούσε το 2007. (Ionescu, 2012)

Το Νοέμβριο του 2007, εμφανίστηκε μια συνεργασία πολλών μεγάλων εταιρειών των τηλεπικοινωνιών (και άλλων κοντινών πεδίων της πληροφορικής) μεταξύ των οποίων οι T-Mobile, Qualcomm, Motorola και φυσικά η Google. Αυτή η συνεργασία αποσκοπούσε στην ανάπτυξη ανοιχτών προδιαγραφών για κινητές συσκευές και ονομάστηκε Open Handset Alliance. (Open Handset Alliance, 2007)

Το πρώτο κινητό τηλέφωνο που βγήκε στην αγορά με λειτουργικό Android ήταν το HTC Dream (γνωστό επίσης ως T-Mobile G1 στις Ηνωμένες Πολιτείες και σε κάποιες χώρες της Ευρώπης) με την έκδοση 1.0, τον Οκτώβριο του 2008. (Wilson, 2008) Η εμπορική επιτυχία που σημείωσε ήταν σημαντική, καθώς υπολογίστηκε τον Μάρτιο του 2009 ότι καταλάμβανε το 6% της αγοράς στις Ηνωμένες Πολιτείες. (Krazit, 2009)

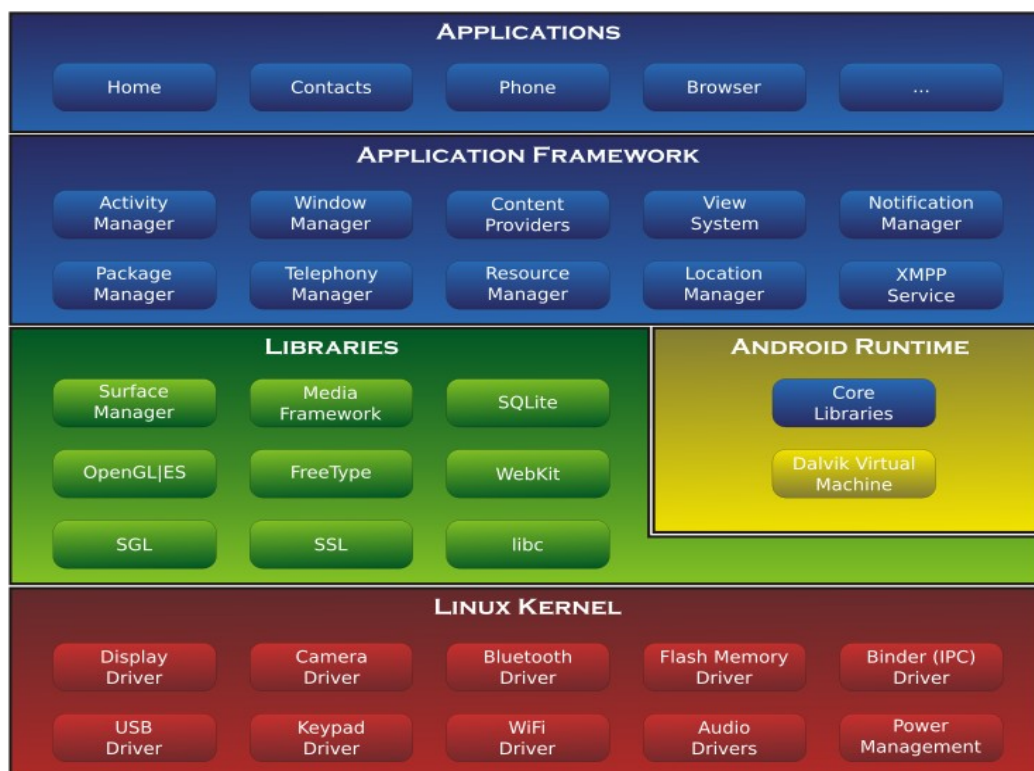
Το Android σταδιακά άρχισε να αυξάνει τις πωλήσεις του και το ποσοστό της αγοράς που καταλαμβάνει συρρικνώνοντας εκείνο των ανταγωνιστών του. Σήμερα, το Android καταλαμβάνει το 85% της παγκόσμιας αγοράς όσον αφορά τη μεταφορά φορτίων κινητών τηλεφώνων. (Tweedie, 2014)

2.2. Τεχνικά Στοιχεία

2.2.1. Android Software Stack

Πρακτικά, η πλατφόρμα του Android είναι μια στοίβα λογισμικού. Στο χαμηλότερο επίπεδο βρίσκεται ο πυρήνας Linux. Ακριβώς πάνω από αυτό το επίπεδο βρίσκονται οι βιβλιοθήκες (middleware) και ακόμη πιο πάνω βρίσκεται το επίπεδο των εφαρμογών.

Στο παρακάτω σχήμα, αναπαρίστανται τα επίπεδα αυτά με τα επιμέρους στοιχεία τους, τα οποία θα αναλυθούν στις επόμενες ενότητες.



Σχήμα 1: Η αρχιτεκτονική του Android (πηγή: wikipedia.org)

2.2.2. Πυρήνας Linux

Το Android χρησιμοποιεί τον πυρήνα του Linux με κάποιες διαφοροποιήσεις. Μάλιστα, μερικές από αυτές τις διαφοροποιήσεις έχουν προστεθεί και σε επόμενες εκδόσεις του πυρήνα, όπως λόγω χάρη τα wakelocks (τα οποία θα αναλυθούν παρακάτω) και κάποιοι οδηγοί. Πιο συγκεκριμένα, μέχρι τις αρχές του 2011 είχαν γίνει περίπου 100 commits από συγγραφείς του

Android, χωρίς να υπολογιστούν οι συνεισφορές στη διακλάδωση “staging” του πυρήνα (δηλ. στο σημείο όπου φυλάσσονται οι οδηγοί που δεν μπορούν να χρησιμοποιηθούν ακόμη από τον πυρήνα για τεχνικούς λόγους (Kroah-Hartman, 2009)). Τα περισσότερα από αυτά τα commits αφορούσαν την υποστήριξη για επεξεργαστές αρχιτεκτονικής ARM.

Τα patches αυτά αποσκοπούν στη διόρθωση κάποιων σφαλμάτων ή στην υποστήριξη νέου υλικού (π.χ. ARM, Bluetooth κλπ.), στη βελτίωση της διαχείρισης ενέργειας (wakelocks, alarm timer) και αναφοράς σφαλμάτων (logger, ram_console), καθώς και στη βελτίωση της απόδοσης (ashmem, Binder) και της ασφάλειας (Paranoid Network). (Stultz, 2011)

Ο Robert Love έγραψε σε άρθρο του κάποια επιγραμματικά στοιχεία για τις διαφοροποιήσεις του πυρήνα (Love, 2013) :

- Binder: Πρόκειται για ένα μηχανισμό επικοινωνίας μεταξύ διεργασιών.
- ashmem (Android shared memory): Είναι ένας κατανεμητής κοινής μνήμης.
- pmem (Process memory allocator): Ένας οδηγός για τη χαρτογράφηση μεγάλων κομματιών της φυσικής μνήμης στο χώρο του χρήστη.
- Logger: Ο Android Logger χρησιμοποιεί τρεις διαφορετικές ενδιάμεσες μνήμες (buffers) σε καθεμιά από τις οποίες αποθηκεύονται διαφορετικού τύπου γεγονότα. (Android Developers, n.d.-j)
- Wakelocks: Πρόκειται για ένα σύνολο patches στον πυρήνα Linux ώστε να επιτρέπεται στην καλούσα διεργασία να αποτρέψει το σύστημα από το να μπει σε μια κατάσταση χαμηλής κατανάλωσης ενέργειας. Αυτό που απαιτείται όμως είναι μεγάλη προσοχή στο να χρησιμοποιούνται τα wakelocks μόνο όταν είναι απολύτως απαραίτητο και για το λιγότερο δυνατό χρόνο, καθώς επηρεάζει τη διάρκεια ζωής της μπαταρίας. (Android Developers, n.d.-f)
- Low memory killer (Viking Killer): Ένας μηχανισμός που επιτρέπει το “θάνατο” διεργασιών όταν λιγοστεύει η διαθέσιμη μνήμη. Σκοτώνει τις διεργασίες που δεν έχουν χρησιμοποιηθεί για το μεγαλύτερο χρονικό διάστημα.
- paranoid network security: Περιορίζει την πρόσβαση ορισμένων διεργασιών σε κάποιες λειτουργίες δικτύωσης.

Επιπλέον, να σημειωθεί πως υποστηρίζεται και το σύστημα αρχείων Yaffs2, που είναι ένα σύστημα αρχείων τεχνολογίας flash.

2.2.3. Σύστημα Καταγραφής Android

Αξίζει να αφιερωθεί ξεχωριστή ενότητα για το σύστημα καταγραφής του Android, παρότι ανήκει στο επίπεδο του πυρήνα. Αρχικά, αποτελείται από τρεις ενδιάμεσες μνήμες πυρήνα (buffers) και το logcat που υποστηρίζει την ανάγνωση των μηνυμάτων καταγραφής. Η εντολή “logcat” μπορεί να εκτελεστεί μέσω του adb (Android Debug Bridge – γέφυρα εντοπισμού σφαλμάτων του Android) ή του DDMS (Dalvik Debug Monitor Server – διακομιστής παρακολούθησης αποσφαλμάτωσης Dalvik). (Android Developers, n.d.-j)

Όσον αφορά τις τρεις ενδιάμεσες μνήμες του πυρήνα, καθεμιά από αυτές περιέχει διαφορετικά γεγονότα ανάλογα με το περιεχόμενό τους και το μέρος του συστήματος όπου λαμβάνουν χώρα. Συγκεκριμένα, πρόκειται για τις εξής μνήμες:

- κύρια (main): η κύρια καταγραφή γεγονότων σχετικών με τις εφαρμογές
- γεγονότων (events): περιέχει πληροφορίες σχετικές με τα γεγονότα του συστήματος
- radio: πρόκειται για πληροφορίες του τηλεφώνου

Κάθε ένα μήνυμα του logger, περιλαμβάνει μια ετικέτα κι ένα βαθμό προτεραιότητας. Η ετικέτα (tag) υποδεικνύει το τμήμα του συστήματος από το οποίο προέρχεται το μήνυμα και η προτεραιότητα (priority) δείχνει το πόσο σημαντικό είναι το μήνυμα. Για παράδειγμα, κάποιες από τις βαθμίδες προτεραιότητας είναι οι βαθμίδες σφάλματος (error), ειδοποίησης (warning) ή απλώς φλυαρίας (verbose).

2.2.4. Βιβλιοθήκες

Προκειμένου να έχει τη δυνατότητα να χειριστεί κάποιους τύπους δεδομένων και να αντιληφθεί κάποια γεγονότα, το Android χρησιμοποιεί κάποιες βιβλιοθήκες και κάποιους εξυπηρετητές όπως είναι οι:

- Bionic: Μια πρότυπη βιβλιοθήκη C που είναι πολύ μικρή και πολύ γρήγορη. Αυτά τα χαρακτηριστικά είναι απαραίτητα διότι πρόκειται για φορητές συσκευές με χαμηλή επεξεργαστική ισχύς. Τέλος, αξίζει να σημειωθεί πως δεν είναι συμβατή με την glibc (GNU libc). (Brady, n.d.)

- Διαχειριστής Επιφάνειας (Surface Manager): Παρέχει τη δυνατότητα διαχείρισης οθόνης. Χρησιμοποιεί μια ενδιάμεση μνήμη, η οποία φροντίζει για τη σύνθεση τρισδιάστατων και δισδιάστατων γραφικών επιπέδων από διάφορες εφαρμογές.
- Διαχειριστής Ήχου (Audio Manager): Είναι υπεύθυνο για όλες τις συσκευές εξόδου ήχου.
- Media framework: Προσφέρει μια πληθώρα κωδικοποιήσεων για υποστήριξη πολυμέσων. Βασίζεται στο OpenCORE και περιλαμβάνει βιβλιοθήκες τόσο για ήχο και βίντεο όσο και για στατική εικόνα.
- SQLite: Πρόκειται για μια μηχανή σχεσιακής βάσης δεδομένων για την αποθήκευση δεδομένων σε Android. Προσφέρεται για χρήση στις εφαρμογές που εγκαθίστανται στη συσκευή ενώ τα σημαντικότερα χαρακτηριστικά της είναι πως πρόκειται για μια πολύ ελαφριά μηχανή βάσεων δεδομένων και πως λειτουργεί χωρίς διακομιστή. (SQLite, n.d.)
- WebKit: Το WebKit είναι μια μηχανή διάταξης για την απόδοση ιστοσελίδων σε περιηγητές ιστού. Πιο απλά, χρησιμοποιείται για την απόδοση HTML περιεχομένου και υποστηρίζει CSS, JavaScript κλπ.. Εκτός από το Android, μπορεί να βρεθεί στους περιηγητές Safari και Chrome της Apple και της Google αντίστοιχα. Στη δεύτερη περίπτωση, πρόκειται για κάποιο fork του project (ονομαζόμενο “Blink” (Bright, 2013)). Έχει γραφεί σε γλώσσα C++ και, πλέον, είναι ανοιχτού κώδικα. (WebKit, n.d.)
- OpenGL: Βιβλιοθήκη για τρισδιάστατη και δισδιάστατη επιτάχυνση λογισμικού. Προσφέρει το πιο διαδεδομένο API (application programming interface) γραφικών. Σε μια τυπική συσκευή Android χρησιμοποιείται τόσο από τους Χάρτες της Google, όσο και από τα παιχνίδια που μπορεί να είναι εγκατεστημένα. (OpenGL, n.d.)
- SSL (Secure Sockets Layer): Προσφέρει τη δυνατότητα εγκαθίδρυσης μιας κρυπτογραφημένης σύνδεσης μεταξύ ενός εξυπηρετητή κι ενός πελάτη ώστε οι συναλλαγές ανάμεσά τους να είναι ασφαλείς. Η τεχνολογία αυτή πλέον ονομάζεται Transport Layer Security (TLS) για τεχνικούς λόγους. (Android Developers, n.d.-1) Ύστερα από το σκάνδαλο με το σφάλμα Heartbleed η έκδοση 4.1.1 κρίθηκε ανασφαλής για οικονομικές συναλλαγές. (G. Smith, 2014)

2.2.5. Android Runtime

Ο χρόνος εκτέλεσης Android (όπως μπορεί να μεταφραστεί ελεύθερα το “Android Runtime”) είναι ένα υποεπίπεδο που εμπεριέχεται στο επίπεδο των βιβλιοθηκών. Η σημαντικότητα

αυτού του επιπέδου συνίσταται στο ότι με την έλλειψη αυτού το Android θα ήταν απλώς μια φορητή έκδοση του Linux. Εμπεριέχει δύο στοιχεία: την εικονική μηχανή ART (απόγονος της μηχανής Dalvik) και τις βιβλιοθήκες πυρήνα Java. (Android Developers, n.d.-c)

Η μηχανή ART είναι ο διαχειριστής του χρόνου εκτέλεσης στο Android. Οι κλάσεις καθώς και τα αρχεία .jar θα πρέπει να μετατραπούν σε αρχείο .dex προκειμένου να καταφέρουν να εκτελεστούν επιτυχώς από τη μηχανή. Τέλος, υποστηρίζεται η ύπαρξη πολλών εικονικών μηχανών στην ίδια συσκευή.

Όπως φαίνεται από τα παραπάνω η μηχανή ART δεν είναι μια εικονική μηχανή Java. Συνεπώς, απαιτείται μια βιβλιοθήκη όπου θα αναλαμβάνει να παρέχει τη λειτουργικότητα του πυρήνα της Java. Αυτό το ρόλο κατέχουν οι βιβλιοθήκες πυρήνα Java.

2.2.6. Application Framework

Αυτό το επίπεδο αποτελεί ένα σύνδεσμο μεταξύ των βιβλιοθηκών και των εφαρμογών. Επιπροσθέτως, διαχειρίζεται τη διεπιφάνεια του χρήστη και τους πόρους που χρησιμοποιούν οι εφαρμογές.

Μερικά από τα πιο σημαντικά κομμάτια του Application Framework είναι τα παρακάτω:

- Διαχειριστής Δραστηριοτήτων (Activity Manager): Διαχειρίζεται τον κύκλο ζωής δραστηριοτήτων των εφαρμογών. Περισσότερα για τον κύκλο ζωής μιας δραστηριότητας μπορούν να βρεθούν στην ενότητα με τίτλο “Κύκλος Ζωής Δραστηριότητας”.
- Πάροχοι Περιεχομένου (Content Providers): Συνδέει δεδομένα με μια διεργασία. Με αυτό τον τρόπο εξασφαλίζεται η διαχείριση των δεδομένων από την εφαρμογή στην οποία ανήκουν, η μη τροποποίηση των δεδομένων από κάποια άλλη εφαρμογή που δε θα έπρεπε να έχει πρόσβαση σε αυτά αλλά και ο διαμοιρασμός δεδομένων μεταξύ των εφαρμογών, όπου αυτό είναι απαραίτητο. Για την τελευταία παρεχόμενη επιλογή, απαιτείται ωστόσο η ανάπτυξη ενός παρόχου περιεχομένου (στις υπόλοιπες περιπτώσεις δε χρειάζεται κάτι τέτοιο). (Android Developers, n.d.-d)
- Διαχειριστής Τηλεφωνίας (Telephony Manager): Διαχειρίζεται και παρέχει πρόσβαση στις πληροφορίες που έχουν σχέση με τηλεφωνικές υπηρεσίες. Επιπλέον, έχει τη δυνατότητα να διακρίνει τις καταστάσεις της τηλεφωνικής υπηρεσίας της συσκευής. (Android Developers, n.d.-n)

- Διαχειριστής Τοποθεσίας (Location Manager): Παρέχει πρόσβαση στο σύστημα τοποθεσίας. (Android Developers, n.d.-g)
- Διαχειριστής Πόρων (Resource Manager): Ως πόροι νοούνται όλα τα στοιχεία μιας εφαρμογής που δεν είναι κώδικας. Για παράδειγμα, εικόνες, ήχοι, βίντεο κλπ.. (Android Developers, n.d.-b)
- Διαχειριστής Ειδοποιήσεων (Notification Manager): Αναλαμβάνει να ενημερώσει το χρήστη για οτιδήποτε μπορεί να συμβαίνει στη συσκευή. Μια ειδοποίηση μπορεί να είναι ένα τηλεφώνημα ή ένα μήνυμα, μια δυνατή ενημέρωση κάποιας εφαρμογής ή του ίδιου του λειτουργικού συστήματος, ή τέλος, κάτι που συνέβη μέσω κάποιας εφαρμογής ή ένα πρόβλημα του λειτουργικού συστήματος. Υπάρχουν τρεις μορφές ειδοποιήσεων: ένα εικονίδιο στην μπάρα καταστάσεων, εναλλαγή καταστάσεων των φωτεινών ενδείξεων της συσκευής ή με ειδοποίηση μέσω ήχου ή δόνησης. (Android Developers, n.d.-i)

2.2.7. Εφαρμογές

Στο ανώτατο επίπεδο της στοίβας λογισμικού του Android βρίσκονται οι εφαρμογές. Οι εφαρμογές μπορούν να χωριστούν σε δύο ομάδες με βάση το ποιος την έχει εγκαταστήσει στη συσκευή. Συνεπώς, οι ομάδες είναι οι εξής: προ εγκατεστημένες εφαρμογές (pre-installed) κι εφαρμογές επιλογής του χρήστη.

Αρχικά, οι pre-installed εφαρμογές καθιστούν τη συσκευή ένα λειτουργικό κινητό τηλέφωνο (εφόσον πρόκειται για κινητό τηλέφωνο) και προσφέρουν κάποιες εφαρμογές που ήταν διαθέσιμες και σε κινητά τηλέφωνα παλαιότερων γενεών, όπως π.χ. κάμερα (η εφαρμογή διαχείρισης της κάμερας), αριθμομηχανή, λογισμικό για αναπαραγωγή μουσικής κλπ.. Γενικά, τέτοιες εφαρμογές είναι οι:

- εφαρμογή για τηλεφωνικές κλήσεις
- εφαρμογή για λήψη και αποστολή γραπτών μηνυμάτων
- ένας τυπικός περιηγητής διαδικτύου
- ένα ημερολόγιο
- μια εφαρμογή ξυπνητηριών

Οι pre-installed εφαρμογές μπορούν να προσφέρουν επιπρόσθετες λειτουργίες στο χρήστη.

Αυτό φυσικά γίνεται παρέχοντας εφαρμογές για υπηρεσίες της Google, εφαρμογές της κατασκευάστριας εταιρίας (δηλαδή της εταιρίας που δημιούργησε τη συσκευή) κι εφαρμογές άλλων τρίτων συνεργαζόμενων με την κατασκευάστρια εταιριών. Τέτοιες υπηρεσίες είναι οι ακόλουθες:

- Youtube, για προβολή βίντεο στη δημοφιλή ιστοσελίδα
- Google Maps, για πρόσβαση στους χάρτες της εταιρείας με όλες τις διευκολύνσεις που παρέχουν, όπως π.χ. υπηρεσίες τοποθεσίας (υπάρχει η δυνατότητα να εντοπίσει την τοποθεσία του χρήστη και να την προβάλει στο χάρτη κτλ.)
- Google Search, για αναζήτηση στο διαδίκτυο
- Google Drive, για αποθήκευση αρχείων στην υπηρεσία cloud της Google
- εφαρμογές κοινωνικών δικτύων όπως Facebook, Twitter, Google+ κτλ.
- εφαρμογές γνωστών ειδησεογραφικών ιστοσελίδων

Όσον αφορά τους κατασκευαστές των συσκευών, σχεδόν όλοι (αν όχι όλοι), προσθέτουν στις συσκευές τους κάποιες δικές τους εφαρμογές. Με αυτές τις εφαρμογές, οι συσκευές διαφοροποιούνται και προσφέρουν κάποιες υπηρεσίες στους κατόχους των συσκευών που δε θα απολάμβαναν εφόσον κατείχαν κάποια άλλη συσκευή (μερικές φορές ακόμη και παλαιότερη συσκευή του ίδιου κατασκευαστή). Εφαρμογές τέτοιου τύπου μπορεί να είναι κάποιος περιηγητής ιστού ή πιο συχνά ένας launcher (πρόκειται για εργαλείο παραμετροποίησης του γραφικού περιβάλλοντος που μερικές φορές μπορεί να προσφέρει και κάποια εργαλεία, όπως π.χ. ενημέρωση για τον καιρό).

Τέλος, ένας χρήστης μπορεί να εγκαταστήσει οποιαδήποτε εφαρμογή επιθυμεί. Υπάρχει μια πληθώρα προσφερόμενων εφαρμογών. Ο κάθε χρήστης επιλέγει ανάλογα με τις ανάγκες και τα ενδιαφέροντά του τις εφαρμογές του. Οι πιο τυπικές εφαρμογές είναι οι ακόλουθες:

- παιχνίδια
- εφαρμογές κοινωνικών μέσων δικτύωσης
- σημειωματάρια
- εφαρμογές για υπηρεσίες cloud
- εφαρμογές για mail
- περιηγητές διαδικτύου
- εφαρμογές διαχείρισης κι επεξεργασίας πολυμέσων

2.2.8. Κύκλος Ζωής Δραστηριότητας

Αρχικά, μια δραστηριότητα είναι ένα στοιχείο μιας εφαρμογής με το οποίο ένας χρήστης μπορεί να αλληλεπιδράσει. Για παράδειγμα, μια δραστηριότητα μπορεί να προβάλει ένα χάρτη ή μια φόρμα εισαγωγής στοιχείων. Τυπικά, μια εφαρμογή αποτελείται από παραπάνω από μια δραστηριότητες και ο μηχανισμός για την εμφάνισή τους είναι αυτός της στοίβας.

Πιο αναλυτικά, η πρώτη δραστηριότητα (κύρια δραστηριότητα) που εμφανίζεται στο χρήστη θα πρέπει να δηλωθεί ως αρχική δραστηριότητα στο αρχείο “AndroidManifest.xml”. Από αυτή, είναι πολύ πιθανό να προκληθούν άλλες δραστηριότητες, οι οποίες θα μπουν στην κορυφή της στοίβας. Έστω, ότι μια νέα δραστηριότητα δημιουργείται και μπαίνει στην κορυφή της στοίβας, ο χρήστης μπορεί να εμφανίσει την προηγούμενη δραστηριότητα πατώντας απλώς το πλήκτρο “Επιστροφή” της συσκευής του.

Μια δραστηριότητα που έχει δημιουργηθεί (και μέχρι να καταστραφεί) μπορεί να βρίσκεται κάθε χρονική στιγμή σε μια από τις εξής τρεις καταστάσεις (Android Developers, n.d.-h):

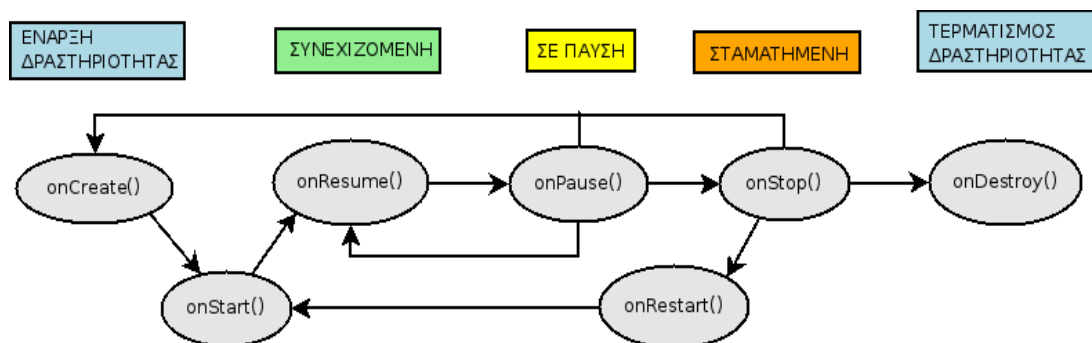
- Συνεχιζόμενη (Resumed): Μια δραστηριότητα βρίσκεται σε αυτή την κατάσταση όταν έχει την προσοχή του χρήστη.
- Σε Παύση (Paused): Μια άλλη δραστηριότητα έχει την προσοχή του χρήστη αλλά η δραστηριότητά μας είναι “ζωντανή” στη μνήμη και είναι ακόμη ορατή στο χρήστη. Το σύστημα μπορεί να θανατώσει μια δραστηριότητα που βρίσκεται σε παύση όταν η μνήμη της συσκευής λιγοστεύει σημαντικά, κάτι που συμβαίνει όμως σε έκτακτες συνθήκες.
- Σταματημένη (Stopped): Μια σταματημένη δραστηριότητα παραμένει “ζωντανή” στη μνήμη αλλά δεν είναι ορατή από το χρήστη. Εφόσον, το σύστημα χρειάζεται μνήμη θα τη θανατώσει.

Προκειμένου, να υλοποιηθούν αυτές οι καταστάσεις και να παρέχεται η δυνατότητα της μετάβασης μεταξύ αυτών για μια δραστηριότητα, η κλάση “Activity” παρέχει ένα σύνολο μεθόδων. Οι επτά μέθοδοι που είναι υπεύθυνες γι' αυτές τις μεταβάσεις είναι οι παρακάτω:

- onCreate(): Η μέθοδος αυτή καλείται όταν η δραστηριότητα δημιουργείται.
- onRestart(): Εφόσον, η δραστηριότητα έχει σταματήσει (βρίσκεται στην κατάσταση “Σταματημένη”), προτού κληθεί η onStart() προκειμένου να ξαναξεκινήσει θα πρέπει να κληθεί η onRestart().

- onStart(): Καλείται ακριβώς πριν η δραστηριότητα γίνει ορατή στο χρήστη.
- onResume(): Λίγο πριν ο χρήστης έχει τη δυνατότητα να αλληλεπιδράσει με τη δραστηριότητα, θα πρέπει να κληθεί αυτή η μέθοδος. Με αυτόν τον τρόπο η δραστηριότητα θα βρεθεί στην κορυφή της στοίβας δραστηριοτήτων.
- onPause(): Εκτελείται τη στιγμή που μια άλλη δραστηριότητα περνά στην κατάσταση “Συνεχιζόμενη”, δηλαδή όταν καλείται η μέθοδος onResume() μιας άλλης δραστηριότητας. Αυτή η μέθοδος θα πρέπει να φροντίζει ώστε η δραστηριότητα που περνά σε κατάσταση παύσης να μη χρησιμοποιεί τον επεξεργαστή.
- onStop(): Καθιστά τη δραστηριότητα αόρατη προς το χρήστη. Αυτό μπορεί να συμβαίνει για δυο λόγους, είτε επειδή πρόκειται να καταστραφεί είτε επειδή κάποια άλλη δραστηριότητα συνεχίστηκε (με την έννοια ότι πέρασε στην κατάσταση “Συνεχιζόμενη”) και την καλύπτει.
- onDestroy(): Καλείται όταν η δραστηριότητα τερματίζεται. Αυτό μπορεί να συμβεί είτε επειδή η δραστηριότητα εκτέλεσε τους σκοπούς της και πλέον είναι έτοιμη να τερματιστεί, είτε επειδή το σύστημα την τερματίζει για να ελευθερώσει χώρο στη μνήμη.

Με τις παραπάνω εφτά παρεχόμενες μεθόδους, δημιουργούνται τρεις βρόχοι στον κύκλο ζωής μιας δραστηριότητας. Οι βρόχοι αυτοί απεικονίζονται στο παρακάτω διάγραμμα.



Σχήμα 2: Ο κύκλος ζωής μιας δραστηριότητας

Πιο συγκεκριμένα, οι τρεις αυτοί βρόχοι είναι οι εξής:

- onStop() - onStart(): Μια σταματημένη δραστηριότητα που είναι αόρατη στο χρήστη, μπορεί να ξαναγίνει ορατή πολύ εύκολα, εφόσον βρίσκεται αποθηκευμένη στη μνήμη. Αρκεί μόνο η κλήση της συνάρτησης onRestart() η οποία πυροδοτεί την έναρξη της onStart(), κι έπειτα ο κύκλος συνεχίζεται κανονικά. Δηλαδή, ακολουθεί η onResume() που

φέρει τη δραστηριότητα στο προσκήνιο.

- onPause() - onResume(): Μια ορατή δραστηριότητα που όμως είναι σε παύση έρχεται στο προσκήνιο απλώς με την κλήση της onResume().
- onStop()/onPause() - onCreate(): Το σύστημα θανατώνει τη δραστηριότητα διότι δεν έχει αρκετό χώρο στη μνήμη και πρέπει να δημιουργήσει κάποια άλλη δραστηριότητα. Όταν ο χρήστης πλοηγηθεί ξανά στη “νεκρή” δραστηριότητα, τότε αυτή θα πρέπει να επαναδημιουργηθεί. Επομένως, εκτελείται ξανά η onCreate() και συνεχίζει κανονικά ο κύκλος ζωής της επαναδημιουργημένης δραστηριότητας.

2.2.9. Κύκλος Ζωής Αντικειμένου (Fragment)

Σε αυτό το σημείο, ενδιαφέρον έχει να αναφερθούμε στα Fragments και στον κύκλο ζωής τους. Ως Fragment μπορεί να θεωρηθεί ένα κομμάτι μιας δραστηριότητας που έχει το δικό του κύκλο ζωής, αλλά ταυτόχρονα επηρεάζεται από τον κύκλο ζωής της δραστηριότητας και σκοπός του είναι να επιτελέσει επιπρόσθετο έργο στη δραστηριότητα αυτή. Πιο απλά, πρόκειται για μια υποδραστηριότητα που εξαρτάται άμεσα από μια δραστηριότητα.

Η εξάρτηση ενός αντικειμένου από τη δραστηριότητά του φαίνεται από το εξής παράδειγμα: όταν η δραστηριότητα καταστραφεί, τότε καταστρέφονται κι όλα τα αντικείμενα που συνδέονται με αυτή. Αντίθετα, όταν η δραστηριότητα είναι ενεργή στη μνήμη, τότε η εφαρμογή μπορεί να χειριστεί όλα τα συνδεδεμένα αντικείμενα της ως ανεξάρτητες οντότητες.

Ο κύκλος ζωής ενός αντικειμένου μοιάζει πολύ με αυτό μιας δραστηριότητας. Υπάρχουν κι εδώ οι ακόλουθες τρεις καταστάσεις (Android Developers, n.d.-e):

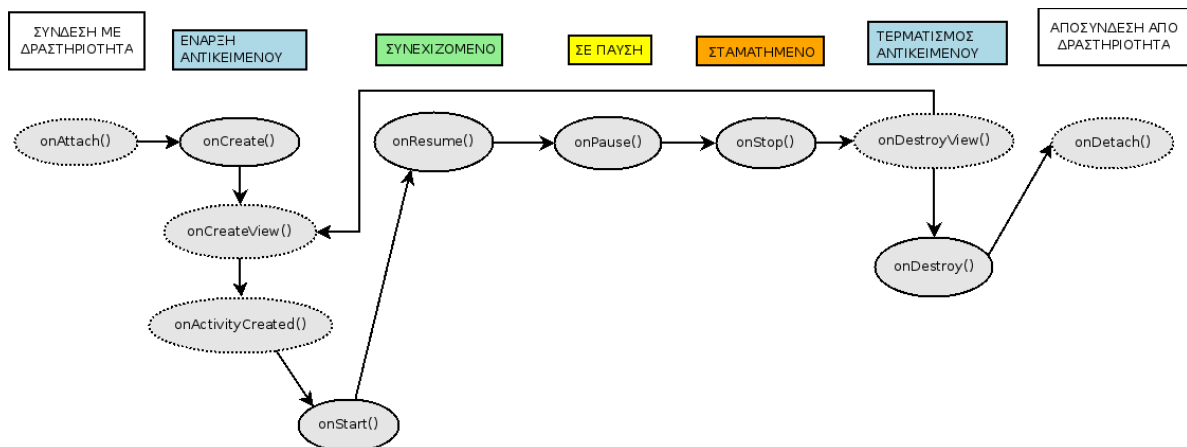
- Συνεχιζόμενη (Resumed): Το αντικείμενο είναι ορατό προς το χρήστη και φυσικά η συνδεδεμένη σε αυτό δραστηριότητα εκτελείται.
- Σε Παύση (Paused): Μια άλλη δραστηριότητα έχει την προσοχή του χρήστη αλλά η δραστηριότητά με την οποία συνδέεται το αντικείμενο είναι “ζωντανή” στη μνήμη .
- Σταματημένη (Stopped): Προκειμένου να περάσει ένα αντικείμενο σε αυτήν την κατάσταση θα πρέπει να ικανοποιείται μια από τις δύο αυτές συνθήκες: είτε η δραστηριότητα που το φιλοξενεί θεωρείται πλέον σταματημένη, είτε το αντικείμενο αυτό έχει αφαιρεθεί από τη δραστηριότητα αλλά υπάρχει ακόμη στη στοίβα της δραστηριότητας. Όπως έχει ήδη αναφερθεί παραπάνω, το αντικείμενο αυτό θα θανατωθεί, όταν θα τερματιστεί και η

δραστηριότητά του.

Η κλάση `Fragment` παρέχει έντεκα μεθόδους σχετικές με τον κύκλο ζωής της. Κάποιες από αυτές είναι ίδιες με τις μεθόδους μιας δραστηριότητας, ενώ πέντε από αυτές διαφέρουν λόγω των ιδιαιτεροτήτων των αντικειμένων έναντι των δραστηριοτήτων. Οι παρεχόμενες μέθοδοι `onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()` και `onDestroy()` δε θα αναλυθούν σε αυτή την ενότητα, μιας και η λειτουργία τους είναι παρόμοια με αυτήν που περιγράφηκε στην ενότητα “Κύκλος Ζωής Δραστηριότητας”. Αντίθετα, σε αυτό το σημείο θα επεξηγηθούν οι μέθοδοι των αντικειμένων που διαφέρουν από αυτές των δραστηριοτήτων.

- `onAttach()`: Καλείται την πρώτη φορά που το αντικείμενο θα συνδεθεί στη δραστηριότητά του.
- `onCreateView()`: Καλείται για να δημιουργήσει τη διεπιφάνεια χρήστη. Δεν είναι απαραίτητο να κληθεί, μιας και υπάρχουν αντικείμενα χωρίς γραφικό περιβάλλον.
- `onActivityCreated()`: Καλείται αφότου έχει ολοκληρωθεί η εργασία της μεθόδου `onCreate()` της συνδεδεμένης δραστηριότητας. Μπορεί να χρησιμοποιηθεί για να συμπληρώσει τα τελευταία κομμάτια της αρχικοποίησης του αντικειμένου.
- `onDestroyView()`: Καλείται όταν η διεπιφάνεια χρήστη που δημιουργήθηκε στη μέθοδο `onCreateView()` έχει αποσπαστεί από το αντικείμενο. Προκειμένου, να επαναχρησιμοποιηθεί αυτή η διεπιφάνεια θα πρέπει να επαναδημιουργηθεί.
- `onDetach()`: Καλείται όταν το αντικείμενο αποσπάται από τη συνδεδεμένη του δραστηριότητα.

Στο σχήμα που ακολουθεί, φαίνεται ο κύκλος ζωής ενός αντικειμένου με τις παραπάνω μεθόδους στη σωστή σειρά εκτέλεσης:



Σχήμα 3: Ο κύκλος ζωής ενός αντικειμένου (Fragment)

Είναι φανερό πως ένα αντικείμενο θα πρέπει πρώτα απ' όλα να συνδεθεί σε μια δραστηριότητα. Έπειτα, να αρχικοποιηθεί και να αρχικοποιήσει τη διεπιφάνεια χρήστη του. Στη συνέχεια, ο κύκλος εκτέλεσης μοιάζει πολύ με εκείνον μιας δραστηριότητας μέχρι και το σημείο όπου περνάει στη σταματημένη κατάσταση. Τέλος, η διεπιφάνεια χρήστη του αντικειμένου καταστρέφεται μαζί με το ίδιο το αντικείμενο που τελικά, αποσυνδέεται από τη δραστηριότητα.

2.2.10. Βασικά στοιχεία λειτουργίας μια τυπικής εφαρμογής Android

Μια εφαρμογή Android, μπορεί να χρησιμοποιεί διάφορα στοιχεία προκειμένου να επιτευχθεί η επιθυμητή λειτουργία της. Τέτοια στοιχεία παρουσιάστηκαν παραπάνω όπως οι δραστηριότητες και τα αντικείμενα. Το Android προσφέρει πολλά περισσότερα εργαλεία για την ανάπτυξη εφαρμογών, μερικά από τα πιο βασικά θα αναφερθούν σε αυτή την ενότητα.

Αρχικά, μια τυπική εφαρμογή θα χρειάζεται να είναι σε θέση να προσφέρει ένα γραφικό περιβάλλον για την αλληλεπίδραση με το χρήστη. Επομένως, θα χρειαστεί κάποια δραστηριότητα και πιθανόν κάποια επιπλέον αντικείμενα. Μιας κι έχουν αναλυθεί παραπάνω αυτά τα δυο στοιχεία, δεν κρίνεται σκόπιμο να επαναληφθούν.

Στη συνέχεια, το Android προσφέρει τη χρήση υπηρεσιών. Οι υπηρεσίες (services) επιφορτίζονται συνήθως με εργασίες που λαμβάνουν χώρα στο παρασκήνιο, και συνεπώς δεν προσφέρουν κάποιο γραφικό περιβάλλον. Παραδείγματα τέτοιων εργασιών αποτελούν οι συναλλαγές με το διαδίκτυο ή η ανάγνωση από αρχεία και γραφή σε αυτά. Μια υπηρεσία μπορεί να έχει μία από τις δύο παρακάτω μορφές (Android Developers, n.d.-m):

- Δημιουργημένη: Δημιουργείται όταν πυροδοτείται από κάποιο στοιχείο της εφαρμογής και ο κύκλος ζωής της είναι ανεξάρτητος από αυτόν του στοιχείου που την πυροδότησε (δηλαδή συνεχίζει τη λειτουργία της και μετά τη θανάτωση του στοιχείου που τη δημιούργησε).
- Δεσμευμένη: Δημιουργείται όταν κάποιο στοιχείο της εφαρμογής δεσμευτεί μαζί της. Διαφορετικά στοιχεία της εφαρμογής μπορούν να δεσμευτούν με μια υπηρεσία και να υποβάλλουν ερωτήματα ή/και να δέχονται αποτελέσματα από αυτή. Πρακτικά, μια δεσμευμένη υπηρεσία είναι ένας διακομιστής, ενώ ένα δεσμευμένο σε αυτή στοιχείο είναι ένας εξυπηρετούμενος.

Συχνή είναι και η περίπτωση όπου μια εφαρμογή θα πρέπει να αποθηκεύσει κάποια δεδομένα στη συσκευή ή κάποιες προτιμήσεις του χρήστη. Αυτό μπορεί να συμβεί είτε με την αποθήκευση σε κάποιο αρχείο στην ενσωματωμένη μνήμη της συσκευής, είτε στην εξωτερική μνήμη ή με τη βοήθεια της SQLite. (Android Developers, n.d.-k)

Προκειμένου ένας χρήστης, να μπορεί να περιηγηθεί σε μια εφαρμογή, υπάρχουν οι δυνατότητες χρήσης καρτελών ή λιστών. Οι δυνατότητες αυτές προσφέρονται από το “ActionBar” που έκανε την εμφάνισή του στο Android 3.0. Πέραν αυτής της ιδιότητας, το “ActionBar” έχει άλλη μια πολύ σημαντική λειτουργία: εμφανίζει το μενού με τις λειτουργίες των ρυθμίσεων, της αναζήτησης κλπ.. (Android Developers, n.d.-a)

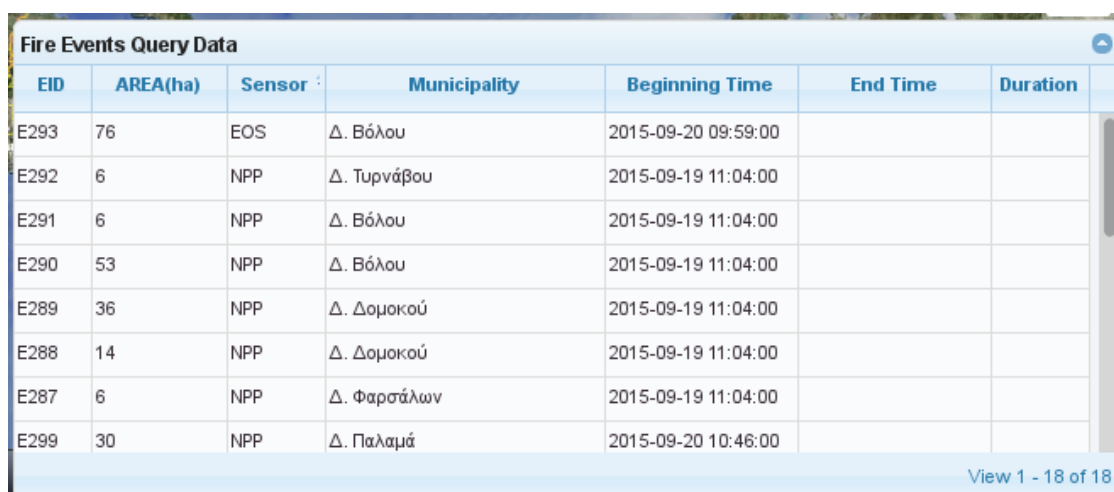
Τέλος, για κάθε εφαρμογή υπάρχει ένα αρχείο τύπου .xml που γνωστοποιεί στο Android όλα όσα χρειάζεται να γνωρίζει για την εφαρμογή. Το όνομα του αρχείου αυτού είναι AndroidManifest.xml και περιέχει πληροφορίες όπως το όνομα του πακέτου της εφαρμογής, τις δραστηριότητες και τις υπηρεσίες της εφαρμογής, τις βιβλιοθήκες με τις οποίες πρέπει να συνδεθεί καθώς και τα δικαιώματα που απαιτεί για να λειτουργήσει απρόσκοπτα (π.χ. πρόσβαση στο διαδίκτυο ή αναγνώριση της τοποθεσίας της συσκευής).

ΚΕΦΑΛΑΙΟ 3: Fire Monitoring Service

3.1. Βασικά Στοιχεία

Η υπηρεσία “Fire Monitoring Service” έχει ως στόχο την έγκαιρη και έγκυρη ενημέρωση για τυχόν σύγχρονες πυρκαγιές, αλλά περιλαμβάνει και λειτουργίες για παρακολούθηση παλαιότερων πυρκαγιών. Τα δεδομένα προέρχονται από εικόνες μέσω δορυφόρου.

Τα δεδομένα εμφανίζονται είτε ως μια λίστα με το σύνολο των πυρκαγιών μαζί με κάποια στοιχεία τους (π.χ. η διάρκεια της πυρκαγιάς καθώς και οι χρονικές στιγμές όπου εκδηλώθηκε κι όπου έσβησε), είτε ως ένα σύνολο γεωμετριών στο χάρτη της υπηρεσίας. Πρακτικά, τα δεδομένα είναι διαθέσιμα και στις δυο αυτές μορφές κατά τη διάρκεια της περιήγησης στην υπηρεσία, ωστόσο ο χρήστης ενδέχεται να επιλέξει την ελαχιστοποίηση της λίστας προκειμένου να έχει καλύτερη άποψη του χάρτη.



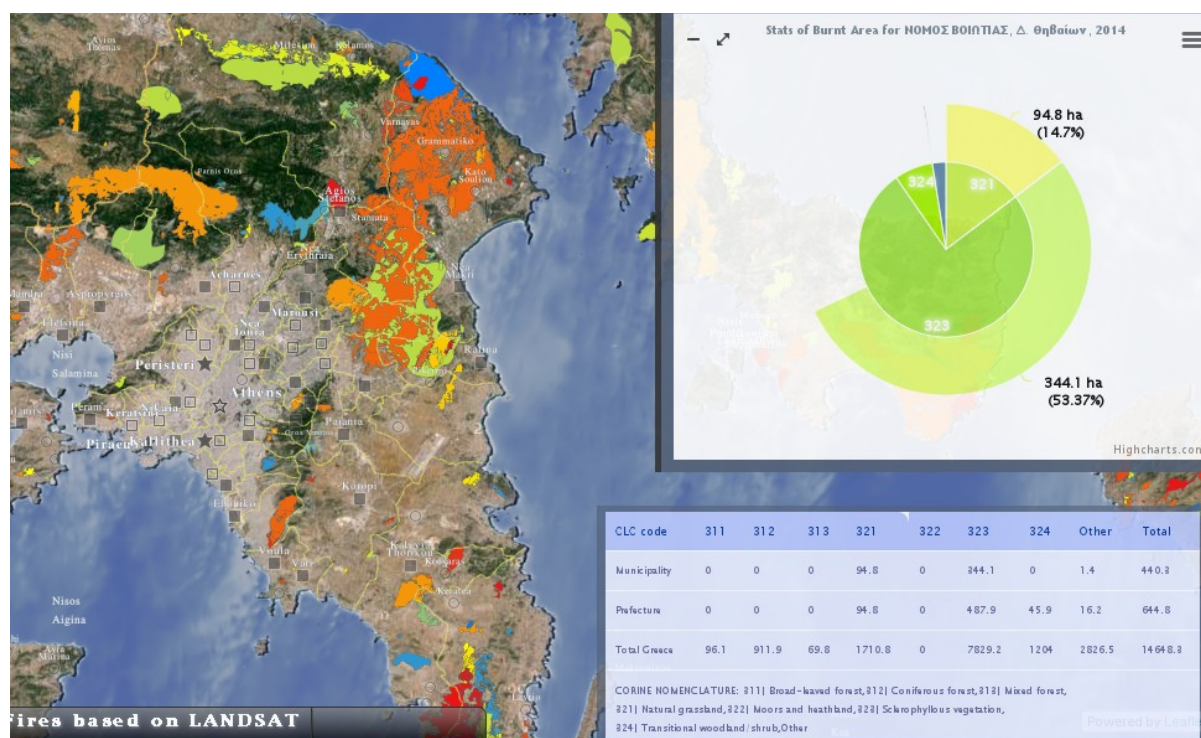
EID	AREA(ha)	Sensor	Municipality	Beginning Time	End Time	Duration
E293	76	EOS	Δ. Βόλου	2015-09-20 09:59:00		
E292	6	NPP	Δ. Τυρνάβου	2015-09-19 11:04:00		
E291	6	NPP	Δ. Βόλου	2015-09-19 11:04:00		
E290	53	NPP	Δ. Βόλου	2015-09-19 11:04:00		
E289	36	NPP	Δ. Δομοκού	2015-09-19 11:04:00		
E288	14	NPP	Δ. Δομοκού	2015-09-19 11:04:00		
E287	6	NPP	Δ. Φαρσάλων	2015-09-19 11:04:00		
E299	30	NPP	Δ. Παλαμά	2015-09-20 10:46:00		

View 1 - 18 of 18

Εικόνα 1: Η λίστα με τις πυρκαγιές

Όσον αφορά τους χάρτες, η υπηρεσία προσφέρει τρεις διαφορετικούς, ώστε η επιλογή της εμφάνισης των δεδομένων να επαφίεται στον ίδιο το χρήστη. Οι προσφερόμενοι χάρτες είναι οι ακόλουθοι:

- Google Earth, που αποτελεί και την αρχική επιλογή της υπηρεσίας
- οι ορθοφωτογραφίες του Κτηματολογίου Α.Ε., που αποτελούν το αποτέλεσμα φωτοληψιών από το 2007 έως και το 2009
- ο χάρτης Corine του Ευρωπαϊκού Οργανισμού Περιβάλλοντος της Ευρωπαϊκής Ένωσης



Εικόνα 2: Ο χάρτης με τις πυρκαγιές και κάποια στατιστικά στοιχεία

Όπως έχει ήδη αναφερθεί, η υπηρεσία διατηρεί ιστορικό πυρκαγιών από το 2007, το οποίο είναι διαθέσιμο για κάθε χρήστη. Αρκεί κάποιος να δοκιμάσει μια αναζήτηση για να βρει παλαιότερα αποτελέσματα. Για το πρόσφατο παρελθόν, υπάρχει ειδική επιλογή μέχρι και για τέσσερις ημέρες πριν.

Η εφαρμογή μπορεί να βρεθεί στο: http://ocean.space.noa.gr/seviri/fend_new/index.php.

3.2. Τεχνικές Λεπτομέρειες

Η ανανέωση της κατάστασης των πυρκαγιών λαμβάνει χώρα κάθε πέντε λεπτά.

Οι πυρκαγιές μπορούν να προβληθούν σε δυο λειτουργίες:

- Ακατέργαστη λειτουργία: ένα τετράγωνο στο χάρτη έχει εμβαδόν περίπου 35χλμ. x 35χλμ.
- Βελτιωμένη λειτουργία: ένα τετράγωνο έχει εμβαδόν 0,5χλμ. x 0,5χλμ.

Φυσικά, η βελτιωμένη λειτουργία είναι πιο ακριβής σχετικά με τα πολύγωνα που εμφανίζονται στο χάρτη και προσφέρει καλύτερη άποψη για το μέγεθος της φλεγόμενης έκτασης στο χρήστη.

3.3. Βάση Δεδομένων

Για την αποθήκευση των δεδομένων χρησιμοποιείται μόνο ένας πίνακας. Τα πεδία του πίνακα είναι επτά, και πιο αναλυτικά ο πίνακας αποτελείται από τα εξής πεδία:

- id (αναγνωριστικό της πυρκαγιάς): Αποτελεί το πρωτεύον κλειδί του πίνακα. Είναι ένας ακέραιος.
- conf (βαθμός εμπιστοσύνης): Πρόκειται για double με ακρίβεια. Ένα πεδίο τέτοιου τύπου καταλαμβάνει χώρο οκτώ bytes στη μνήμη.
- timestamp (χρονική σήμανση πυρκαγιάς): Δεν περιλαμβάνει ζώνη ώρας (without time zone). Στην πραγματικότητα, αυτό σημαίνει πως αποθηκεύεται σε UTC και χρησιμοποιεί τη ζώνη ώρας του συστήματος.
- municipality (ο δήμος στον οποίο ανήκει η περιοχή που βρίσκεται η πυρκαγιά): Είναι μια συμβολοσειρά, το πολύ μέχρι εκατό χαρακτήρων.
- sensor (ο αισθητήρας από τον οποίο λήφθηκαν τα δεδομένα): Πρόκειται για μια συμβολοσειρά το πολύ μέχρι είκοσι χαρακτήρων.
- usable (εάν μπορεί να χρησιμοποιηθεί η συγκεκριμένη εγγραφή): Είναι ένα πεδίο τύπου smallint που καταλαμβάνει δυο bytes στη μνήμη.
- the_geom (η γεωμετρία όπου αποθηκεύεται το πολύγωνο που αναπαριστά την πυρκαγιά): Είναι τύπου “MultiPolygon” που προσφέρεται από το PostGIS.

Μόνο ένας περιορισμός υπάρχει στον πίνακα κι αυτός είναι πως το πεδίο “id” είναι το πρωτεύον κλειδί του. Πιο συγκεκριμένα, το πρωτεύον κλειδί δηλώνει ότι αυτό το πεδίο θα πρέπει να είναι διαφορετικό για κάθε εγγραφή (UNIQUE) και να μην είναι κενό (NOT NULL).

ΚΕΦΑΛΑΙΟ 4: Εφαρμογή Fire

4.1. Βασικά Στοιχεία

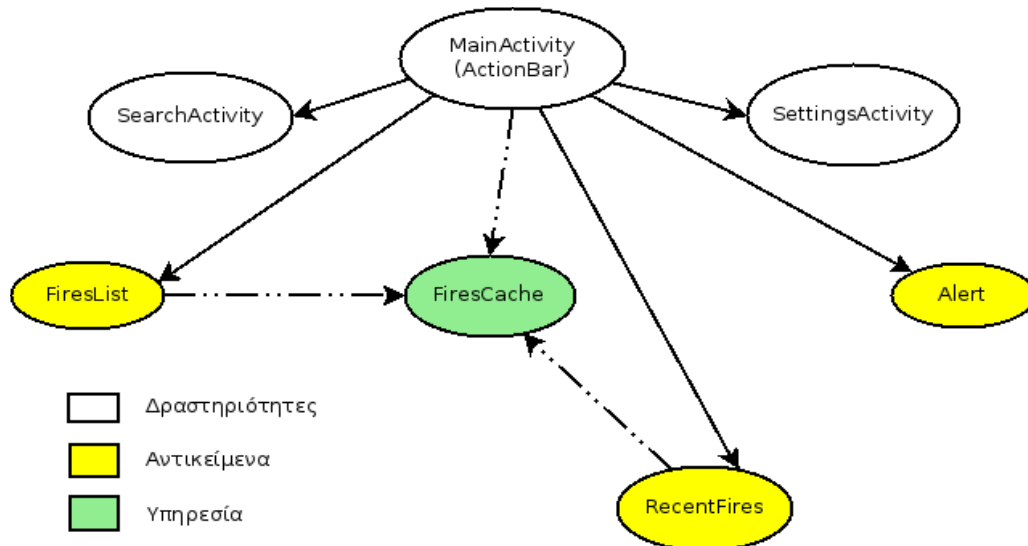
Η εφαρμογή Fire δημιουργήθηκε με σκοπό να φέρει τις δυνατότητες της υπηρεσίας “Fire Monitoring Service” στο Android. Ουσιαστικά, προσφέρει τις υπηρεσίες αναζήτησης και προβολής πυρκαγιών σε πραγματικό χρόνο (εφόσον συνδεθεί με τη βάση δεδομένων της υπηρεσίας) καθώς και την αναζήτηση παλαιότερων πυρκαγιών. Επιπλέον, προσφέρει μια επιπρόσθετη λειτουργία, που δίνει τη δυνατότητα σε ένα χρήστη να ανεβάσει στη βάση δεδομένων της υπηρεσίας μια φωτογραφία μιας πυρκαγιάς, μαζί με την τοποθεσία της συσκευής και το χρόνο λήψης της φωτογραφίας. Με αυτό τον τρόπο η υπηρεσία μπορεί να ενημερωθεί με μεγαλύτερη ακρίβεια για την τοποθεσία μιας πυρκαγιάς απ' ότι με τη βοήθεια του δορυφόρου που ενδέχεται να έχει αποκλίσεις μερικών χιλιομέτρων στους υπολογισμούς του.

Η εφαρμογή περιέχει τα εξής στοιχεία:

- δραστηριότητες και αντικείμενα που αναλαμβάνουν να φέρουν εις πέρας την επικοινωνία της εφαρμογής με το χρήστη
- μια υπηρεσία που αναλαμβάνει, μεταξύ άλλων, την επικοινωνία της εφαρμογής με το web service που λαμβάνει ερωτήματα και αποστέλλει αποτελέσματα πυρκαγιών που ικανοποιούν τα ερωτήματα αυτά
- το ActionBar για ομαλή μετάβαση μεταξύ των διαφορετικών στοιχείων της εφαρμογής, αλλά και για ταχύτερη κατανόηση της λειτουργίας της εφαρμογής από το χρήστη, μέσω ενός οικείου γραφικού περιβάλλοντος
- ένα αρχείο για αποθήκευση δεδομένων που είναι πιθανόν να χρησιμοποιηθούν στο μέλλον (cache file)
- ένα αρχείο ρυθμίσεων του χρήστη
- δυνατότητα εύρεσης της τοποθεσίας του χρήστη και προβολής των γύρω πυρκαγιών
- οπτικοποίηση των πυρκαγιών των τελευταίων τριών ημερών σε μορφή λίστας, αλλά και ως πολύγωνα στο χάρτη
- αναζήτηση παλαιότερων πυρκαγιών
- δυνατότητα εναλλαγής μεταξύ του χρόνου της συσκευής και κάποιου άλλου, ορισμένου από το χρήστη, χρόνου

- δυνατότητα επιλογής ρυθμού ενημερώσεων

Στο παρακάτω σχήμα φαίνονται οι βασικότερες δραστηριότητες, τα βασικότερα αντικείμενα και η σχέση μεταξύ αυτών, καθώς και η σχέση τους με τη μοναδική υπηρεσία της εφαρμογής “FiresCache”:



Σχήμα 4: Τα βασικότερα μέρη της εφαρμογής Fire

Τα βέλη στο σχήμα υποδεικνύουν τις σχέσεις μεταξύ των επιμέρους στοιχείων. Ως σχέση, στην προκειμένη περίπτωση, εννοείται η δυνατότητα πυροδότησης ενός μέρους από ένα άλλο, για παράδειγμα η δραστηριότητα “MainActivity” πυροδοτεί τις δραστηριότητες “SearchActivity” και “SettingsActivity”. Οι σχέσεις μεταξύ της υπηρεσίας “FiresCache” με τα υπόλοιπα μέρη, υποδηλώνει τη δέσμευση της υπηρεσίας από τα μέρη που συνδέονται μαζί της.

Η λειτουργία αυτών των μερών, καθώς και των υπολοίπων που δε φαίνονται σε αυτό το σχήμα θα αναλυθούν στις επόμενες υποενότητες.

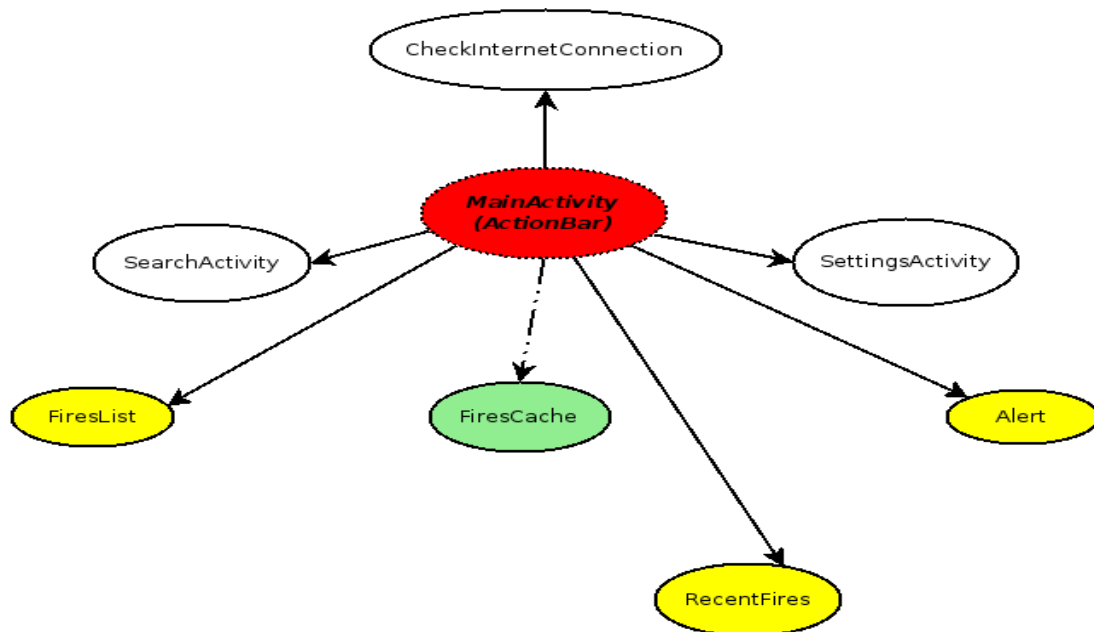
4.2. Κύρια Δραστηριότητα (MainActivity)

Η κύρια δραστηριότητα περιλαμβάνει την “ActionBar” και συνεπώς αποτελεί ένα πολύ σημαντικό κομμάτι της εφαρμογής καθώς ελέγχει όλα τα αντικείμενα και μερικές δραστηριότητες που βασίζονται στην “ActionBar” για την πυροδότηση τους.

Η συγκεκριμένη δραστηριότητα θέτει στην “ActionBar” ποια αντικείμενα θα ελεγχθούν από

αυτή, με ποια σειρά θα εμφανίζονται, τι ακριβώς θα συμβαίνει όταν επιλέγεται μια διαφορετική καρτέλα ή όταν αποεπιλέγεται μια καρτέλα. Επιπλέον, ελέγχει την ονοματοδοσία των καρτελών.

Η κύρια δραστηριότητα σε σχέση με τα συνδεδεμένα με αυτή μέρη, φαίνεται στο παρακάτω σχήμα.



Σχήμα 5: Η κύρια δραστηριότητα και τα άμεσα συνδεδεμένα με αυτή μέρη της εφαρμογής

Αρχικά, η δραστηριότητα αυτή, αξιοποιεί τη βοηθητική κλάση “CheckInternetConnection” προκειμένου να ελέγξει εάν η συσκευή είναι συνδεδεμένη στο διαδίκτυο. Στην περίπτωση όπου δεν είναι συνδεδεμένη, τερματίζει την εφαρμογή, ειδοποιώντας το χρήστη ότι για τη λειτουργία της απαιτείται σύνδεση μέσω Wi-Fi ή 3G/4G. Στην αντίθετη περίπτωση, αρχικοποιεί την “ActionBar”.

Η αρχικοποίηση της “ActionBar” περιλαμβάνει την αρχικοποίηση των αντικειμένων (“FiresList”, “RecentFires”, “Alert”) και την ονοματοδοσία των καρτελών. Επίσης, συνδέει την κάθε καρτέλα με ένα αντικείμενο μέσω της υποκλάσης “TabListener”. Αυτή η υποκλάση αναλαμβάνει την υλοποίηση της εναλλαγής των καρτελών με ό,τι αυτό συνεπάγεται, δηλαδή τι θα πρέπει να συμβεί όταν επιλεχθεί μια καρτέλα και τι όταν αποεπιλεχθεί.

Στη συνέχεια, δεσμεύει την υπηρεσία “FiresCache” και την αρχικοποιεί, με σκοπό να λάβει το αντικείμενο “FiresList” τα αποτελέσματα από την υπηρεσία, αμέσως ή σχεδόν αμέσως μόλις κάνει την εμφάνισή του. Με αυτόν τον τρόπο, είναι φανερό πως η υπηρεσία ξεκινά λίγη ώρα αργότερα από την εφαρμογή, σχεδόν όμως την ίδια στιγμή.

Τέλος, με τη βοήθεια του “ActionBar” δημιουργεί ένα μενού δύο επιλογών, οι οποίες είναι:

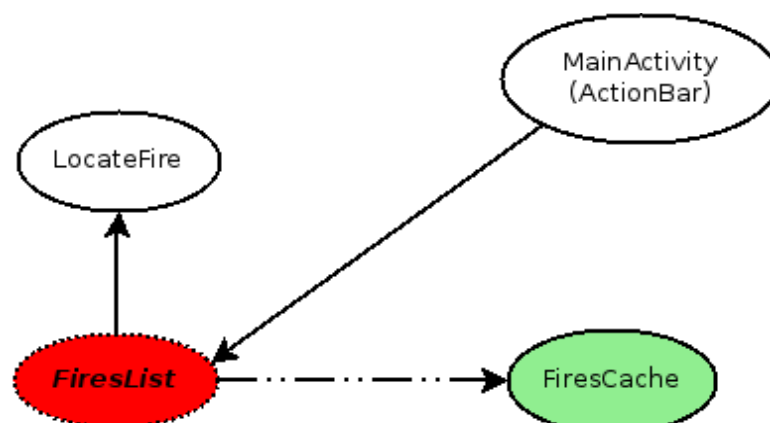
- αναζήτηση παλαιότερων πυρκαγιών
- βασικές ρυθμίσεις/επιλογές του χρήστη

Το μενού βρίσκεται στην κορυφή της οθόνης και αποτελείται από δυο εικονίδια που μπορεί εύκολα να αναγνωρίσει ο χρήστης τη λειτουργία του καθενός.

Σε αυτό το σημείο, αξίζει να σημειωθεί πως η κύρια δραστηριότητα συνδέεται και με τη βοηθητική κλάση “LocationFinder” προκειμένου να βεβαιωθεί πως είναι εγκατεστημένα τα GooglePlay Services χωρίς τα οποία δεν μπορεί να λειτουργήσει η εφαρμογή. Η σύνδεση αυτή δε φαίνεται στο σχήμα γιατί δε βοηθά στη λειτουργία της εφαρμογής, πέραν της αποτροπής κάποιων μοιραίων σφαλμάτων. Στην περίπτωση όπου δε βρεθούν οι εν λόγω υπηρεσίες, δε δημιουργούνται οι καρτέλες της εφαρμογής και ο χρήστης προτρέπεται να εγκαταστήσει την τελευταία έκδοση των υπηρεσιών του GooglePlay για να συνεχίσει. Στην αντίθετη περίπτωση, ισχύουν όλα τα παραπάνω, δηλαδή η ομαλή αρχικοποίηση της εφαρμογής.

4.3. Λίστα Πυρκαγιών (FiresList)

Η πρώτη οθόνη που εμφανίζεται μετά την έναρξη της εφαρμογής, είναι η λίστα με τις πυρκαγιές των τελευταίων τριών ημερών, που αποτελεί και την πρώτη καρτέλα του “ActionBar”.



Σχήμα 6: Η λίστα των πυρκαγιών και τα συνδεόμενα με αυτή μέρη

Το αντικείμενο της λίστας των πυρκαγιών, όπως και κάθε άλλο αντικείμενο, θα πρέπει να συνδέεται με μια δραστηριότητα, η οποία θα το δημιουργεί και θα το ελέγχει. Αυτή η

δραστηριότητα είναι η κύρια δραστηριότητα που περιέχει και το “ActionBar”.

Η λίστα είναι επιφορτισμένη με δυο λειτουργίες:

- την ενημέρωση των πυρκαγιών ανά τακτά χρονικά διαστήματα
- την οπτικοποίηση μιας πυρκαγιάς που θα επιλεγεί από τη λίστα, ως ένα πολύγωνο στο χάρτη

Για να καταφέρει την ενημέρωση των πυρκαγιών, θα πρέπει αρχικά να συνδεθεί με την υπηρεσία “FiresCache”. Στη συνέχεια, θα ζητήσει όλες τις πυρκαγιές που υπάρχουν στη λίστα της “FiresCache” και θα τις εμφανίσει. Έπειτα, θα επαναλαμβάνει τη διαδικασία ανά πέντε δευτερόλεπτα, ζητώντας όμως μόνο τις πυρκαγιές που ενημερώθηκαν στην “FiresCache” τα τελευταία πέντε δευτερόλεπτα. Η επιλογή των πέντε δευτερολέπτων είναι αρκετά σημαντική για δύο λόγους. Αρχικά, μόλις δημιουργηθεί η λίστα θα πρέπει να είναι εντελώς ενημερωμένη, δηλαδή όλες οι πυρκαγιές που υπάρχουν στη λίστα της υπηρεσίας, θα πρέπει να εμφανίζονται και στο χρήστη. Αυτό δε θα συνέβαινε στην περίπτωση όπου η υπηρεσία θα έψαχνε και στο cache file και στο διαδίκτυο για αποτελέσματα και ο χρόνος όπου έστελνε αίτημα για ενημέρωση ήταν σημαντικά μεγαλύτερος (περισσότερα για τη λειτουργία της υπηρεσίας “FiresCache” στην αντίστοιχη ενότητα που βρίσκεται παρακάτω). Ο δεύτερος λόγος είναι πως μία μελλοντική ενημέρωση της λίστας της υπηρεσίας, θα πρέπει να εμφανιστεί άμεσα στο χρήστη. Με τα αιτήματα που λαμβάνουν χώρα κάθε πέντε δευτερόλεπτα, είμαστε βέβαιοι πως οι πυρκαγιές θα εμφανιστούν στο χρήστη είτε σε λιγότερο από πέντε δευτερόλεπτα, είτε σε πέντε δευτερόλεπτα ακριβώς από τη στιγμή που η εφαρμογή είναι ενήμερη γι' αυτές.

Την οπτικοποίηση της πυρκαγιάς στο χάρτη, την πετυχαίνει με τη βοήθεια της δραστηριότητας “LocateFire”. Αυτή η δραστηριότητα περιέχει έναν ενημερωμένο χάρτη, όπου ο χρήστης μπορεί να περιηγηθεί ελεύθερα. Μια πυρκαγιά μπορεί να επιλεγεί από τη λίστα χάρη στην υλοποίηση του “OnItemClickListener”. Πιο συγκεκριμένα, για να υλοποιηθεί ο παραπάνω listener χρειάζεται να συμπεριληφθεί η μέθοδος “onItemClick”.

```
// when a list item is clicked it starts the LocateFire activity
@Override
public void onItemClick(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
    Intent intent = new Intent(getActivity(), LocateFire.class);
    intent.putExtra("position", arg2);
    intent.putExtra("lastRefreshed", lastRefreshed.getTimeInMillis());
    startActivity(intent);
}
```

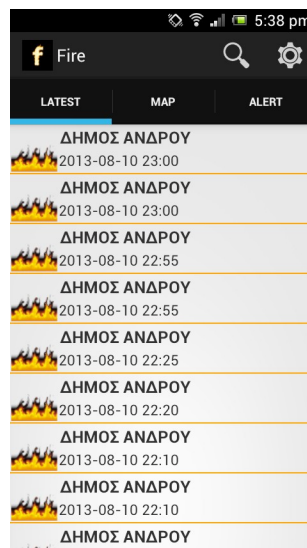
Απόσπασμα Κώδικα 1: Η μέθοδος onItemClick στο αντικείμενο της λίστας των πυρκαγιών

Η μέθοδος “onItemClick” φροντίζει για την πυροδότηση (ή συνέχιση εφόσον ήδη υπάρχει) της δραστηριότητας “LocateFire”. Προσθέτει τη θέση της επιλεγμένης πυρκαγιάς στη λίστα καθώς και το χρόνο της τελευταίας ενημέρωσης της λίστας στην αίτηση για αρχικοποίηση της νέας δραστηριότητας, κι έπειτα περνά στην οθόνη της “LocateFire”.

Συνοψίζοντας, το αντικείμενο “FiresList” ακολουθεί τα εξής βήματα:

1. δέσμευση υπηρεσίας “FiresCache”
2. λήψη όλων των πυρκαγιών της λίστας της υπηρεσίας και προβολή τους στη λίστα του αντικειμένου
3. (προαιρετικά) οπτικοποίηση πυρκαγιάς στο χάρτη με την επιλογή της
4. ενημέρωση ανά τακτά χρονικά διαστήματα

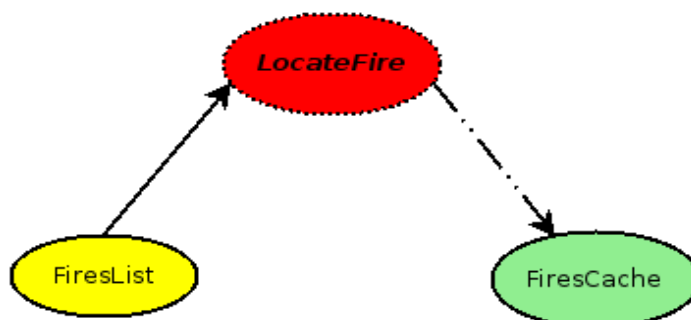
Ένα παράδειγμα της εκτέλεσης της “FiresList” φαίνεται παρακάτω:



Εικόνα 3: Το αντικείμενο “FiresList” σε εκτέλεση

4.4. Δραστηριότητα LocateFire

Η δραστηριότητα “LocateFire” ασχολείται με την οπτικοποίηση της πυρκαγιάς ως πολυγώνου καθώς και με το κεντράρισμα του πολυγώνου αυτού στο χάρτη. Μπορεί να πυροδοτηθεί μονάχα από τη λίστα με τις πυρκαγιές.

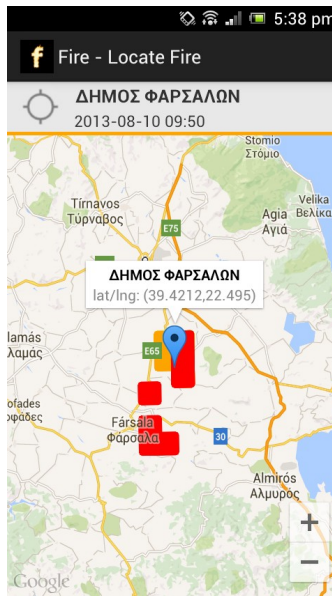


Σχήμα 7: Η δραστηριότητα LocateFire και οι σχέσεις της με τα υπόλοιπα μέρη της εφαρμογής

Μόλις αρχικοποιηθεί η δραστηριότητα, εμφανίζει το χάρτη. Στη συνέχεια, συνδέεται με την υπηρεσία “FiresCache”. Αυτό συμβαίνει για δύο λόγους: για αρχικοποίηση του χάρτη με τα πιο ενημερωμένα πολύγωνα και για την ακριβή θέση του προς αναζήτηση πολυγώνου, ώστε να κεντραριστεί στο χάρτη και να εμφανίσει τα στοιχεία του (σε ποιο δήμο ανήκει και πότε λαμβάνει χώρα η πυρκαγιά). Προκειμένου να επιτευχθεί ο δεύτερος σκοπός, η δραστηριότητα λαμβάνει τα στοιχεία που αποστέλλει το αντικείμενο “FiresList” (θέση στη λίστα και χρόνος τελευταίας ενημέρωσης της λίστας). Έπειτα, στέλνει αυτά τα στοιχεία με ένα ερώτημα στη “FiresCache”, η οποία βρίσκει όλα τα δεδομένα της συγκεκριμένης θέσης της λίστας και τα επιστρέφει στη “LocateFire”. Επομένως, σε αυτό το σημείο η δραστηριότητα αυτή βρίσκεται στη θέση να προβάλει την πυρκαγιά που ζήτησε ο χρήστης.

Γενικότερα, ο χάρτης που προσφέρεται από τη “LocateFire” είναι ελεύθερος προς περιήγηση από το χρήστη, ενώ ενημερώνεται κάθε φορά που η δραστηριότητα περνά στη συνεχιζόμενη κατάσταση ώστε να είναι πάντοτε ενημερωμένος. Τέλος, επιτρέπει την επαναφορά της κάμερας του χάρτη πάνω από την εξεταζόμενη πυρκαγιά με το πάτημα ενός κουμπιού, σε περίπτωση που η περιήγηση του χρήστη στο χάρτη την έχει απομακρύνει από την αρχική της θέση.

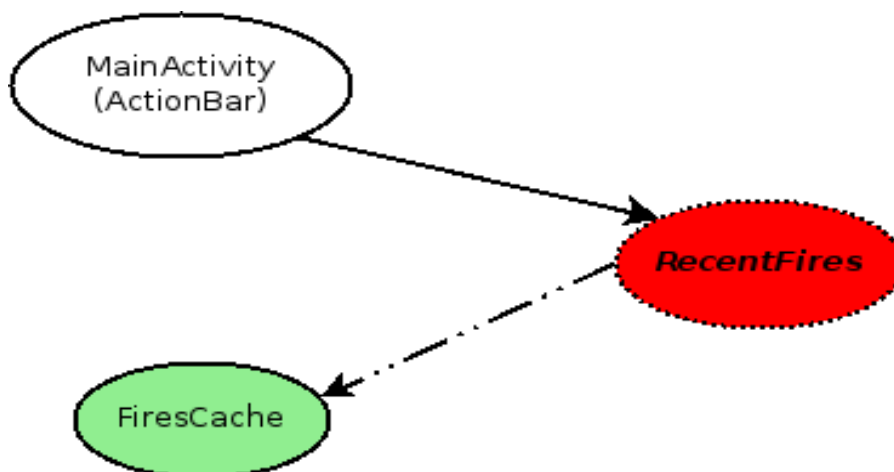
Παρακάτω, παρουσιάζεται μια απεικόνιση της δραστηριότητας “LocateFire”:



Εικόνα 4: Η “LocateFire” εν δράσει: μια πυρκαγιά στο δήμο Φαρσάλων

4.5. Χάρτης Πρόσφατων Πυρκαγιών (RecentFires)

Το αντικείμενο του χάρτη πρόσφατων πυρκαγιών εξαρτάται από την κύρια δραστηριότητα και το “ActionBar”. Προβάλλει τις πυρκαγιές των τριών τελευταίων ημερών και την τοποθεσία του χρήστη εφόσον είναι διαθέσιμη.



Σχήμα 8: Το αντικείμενο του χάρτη πρόσφατων πυρκαγιών

Μόλις επιλεγεί η καρτέλα το αντικείμενο αρχικοποιείται δημιουργώντας το χάρτη. Αμέσως δεσμεύει την υπηρεσία “FiresCache” για να λάβει τις πυρκαγιές που υπάρχουν στη λίστα της υπηρεσίας μέχρι εκείνη τη στιγμή. Κάθε πολύγωνο πυρκαγιάς έχει τρία χαρακτηριστικά:

- τα σημεία των γωνιών που θα σχηματίσουν το πολύγωνο
- ένα χρώμα ανάλογα με το πόσο παλιά είναι η πυρκαγιά
- ένα επίπεδο στο χάρτη (Ένας χάρτης GoogleMap μπορεί να περιέχει πολλά επίπεδα, όσο μικρότερο το επίπεδο, τόσο πιο κάτω θα βρίσκεται το πολύγωνο και πιθανόν να επικαλύπτεται από άλλα πολύγωνα που βρίσκονται σε πιο πάνω επίπεδα. Επομένως, και το επίπεδο στο χάρτη προσδιορίζεται ανάλογα με την παλαιότητα της πυρκαγιάς.)

Σύμφωνα με τα παραπάνω, γίνεται αντιληπτό πως το χρώμα και το επίπεδο ενός πολυγώνου είναι στενά συνδεδεμένα. Υπάρχουν τρεις ομάδες από επίπεδα και χρώματα:

- μηδενικό επίπεδο και κίτρινο χρώμα: Αποτελεί το πρώτο επίπεδο και συμβολίζει τις πυρκαγιές με την παλαιότερη χρονοσήμανση. Οι πυρκαγιές που ανήκουν σε αυτή την ομάδα έλαβαν χώρα πριν από δύο ημέρες από την ημερομηνία και ώρα που έχει οριστεί για την εφαρμογή, είτε πρόκειται για την πραγματική ημερομηνία της συσκευής, είτε για ημερομηνία επιλογής του χρήστη.
- πρώτο επίπεδο και πορτοκαλί χρώμα: Περιλαμβάνει όλες τις πυρκαγιές που έλαβαν χώρα την προηγούμενη ημέρα από την ημερομηνία που έχει οριστεί για την εφαρμογή.
- δεύτερο επίπεδο και κόκκινο χρώμα: Είναι το ανώτερο επίπεδο και έχει τη δυνατότητα να επικαλύπτει και τα δύο άλλα επίπεδα. Αποτελεί την ομάδα που περιέχει όλες τις πυρκαγιές που λαμβάνουν χώρα την ημέρα που έχει οριστεί για την εφαρμογή.

Επιπλέον, υπάρχει το γνωστό κουμπί των χαρτών της Google που κεντράρει το χάρτη στη θέση του χρήστη, εφόσον είναι διαθέσιμη η εύρεση της τοποθεσίας του.

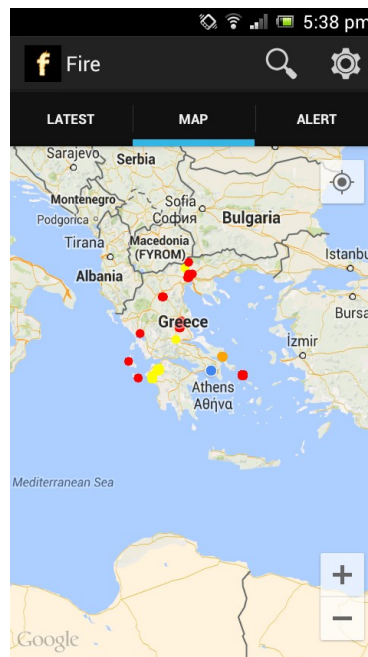
Συνοπτικά τα βήματα που ακολουθεί το αντικείμενο “RecentFires”, είναι τα εξής:

1. αρχικοποίηση του χάρτη
2. δέσμευση υπηρεσίας “FiresCache”
3. λήψη όλων των πυρκαγιών της λίστας
4. σχεδιασμός των πολυγώνων των πυρκαγιών με βάση την ομάδα στην οποία ανήκει το

καθένα από αυτά

5. εύρεση τοποθεσίας συσκευής (εφόσον είναι δυνατόν)
6. ενημέρωση πολυγώνων ανά τακτά χρονικά διαστήματα

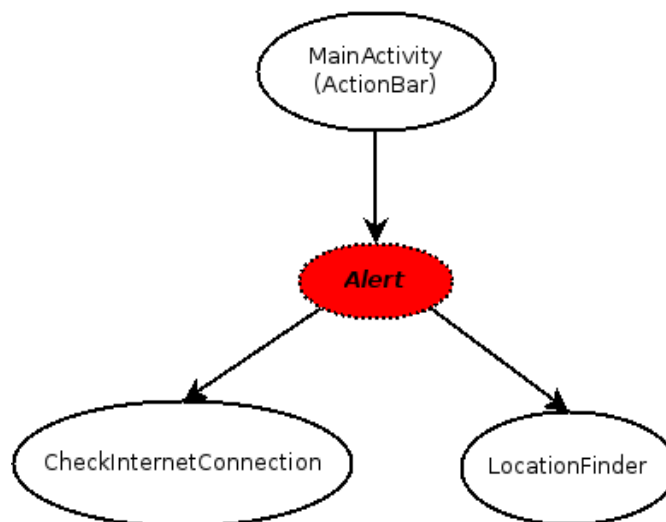
Τέλος, αξίζει να σημειωθεί πως η ενημέρωση των πολυγώνων του χάρτη γίνεται πάλι ανά πέντε δευτερόλεπτα για την ταχύτερη ενημέρωση του χάρτη από τη λίστα της υπηρεσίας “FiresCache”.



Εικόνα 5: Απεικόνιση πυρκαγιών ως πολύγωνα στο χάρτη της Google

4.6. Δυνατότητα Συναγερμού (Alert)

Το τρίτο και τελευταίο αντικείμενο της εφαρμογής, συνδέεται όπως και τα υπόλοιπα δύο με την κύρια δραστηριότητα και το “ActionBar”.

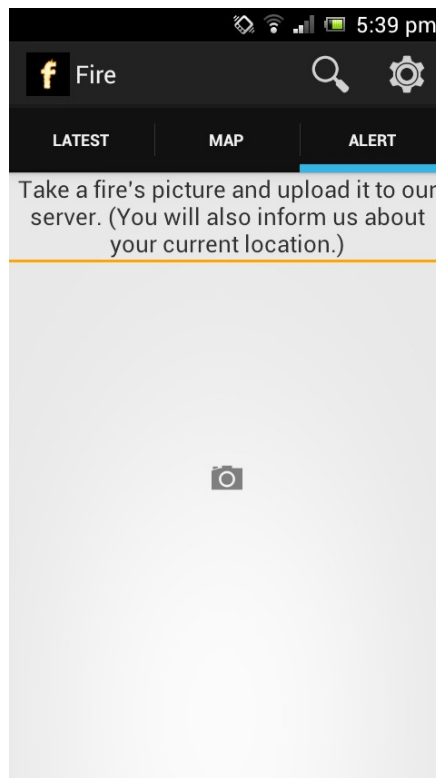


Σχήμα 9: Το αντικείμενο του συναγερμού και οι σχέσεις του με τα υπόλοιπα μέρη της εφαρμογής

Κάθε φορά που το αντικείμενο συναγερμού εισέρχεται στη συνεχιζόμενη κατάσταση, ελέγχει τη σύνδεση με το διαδίκτυο. Εφόσον δε λειτουργεί, ενημερώνει το χρήστη και δεν του επιτρέπει να κάνει οτιδήποτε άλλο σε αυτό το αντικείμενο. Στην αντίθετη περίπτωση, ενεργοποιεί τη διαδικασία αναζήτησης της τοποθεσίας του χρήστη μέσω της κλάσης “LocationFinder”. Όταν ενεργοποιείται η αναζήτηση της τοποθεσίας, ενεργοποιούνται και οι ενημερώσεις αυτής, με στόχο την επίτευξη του καλύτερου υπολογισμού της τοποθεσίας όταν πατηθεί το κουμπί για τη λήψη φωτογραφίας της φωτιάς.

Όταν πατηθεί το κουμπί για τη λήψη της φωτογραφίας, το αντικείμενο λαμβάνει την τοποθεσία του χρήστη. Ελέγχει αν υπάρχει ο φάκελος όπου αποθηκεύονται οι φωτογραφίες της εφαρμογής. Εάν δεν υπάρχει το δημιουργεί. Στη συνέχεια, ενεργοποιεί τη δραστηριότητα που προσφέρεται από τη συσκευή για λήψη φωτογραφιών και λαμβάνει τη φωτογραφία της πυρκαγιάς, την οποία αποθηκεύει στο φάκελο των φωτογραφιών της εφαρμογής στην κάρτα SD. Η ονομασία του αρχείου περιλαμβάνει το χρόνο με ακρίβεια χιλιοστού του δευτερολέπτου, καθώς και την τοποθεσία της συσκευής. Ύστερα, ο χρήστης μπορεί να επιλέξει αν θα στείλει τη φωτογραφία στο server της υπηρεσίας για πιθανή ανάλυση κι επεξεργασία της (ο server αυτή τη στιγμή πέρα από αποθήκευση και φύλαξη των φωτογραφιών δεν επιτελεί κάποιο άλλο έργο) ή αν θα ακυρώσει την αποστολή της.

Τέλος, ο χρήστης ενημερώνεται από το server για την επιτυχία ή αποτυχία της αποστολής της εικόνας.



Εικόνα 6: Το αντικείμενο "Alert"

Τα διαδοχικά βήματα του αντικειμένου συναγερμού στην περίπτωση όπου δε θα υπάρξουν προβλήματα, είναι τα παρακάτω:

1. έλεγχος σύνδεσης με διαδίκτυο
2. ενεργοποίηση ενημερώσεων τοποθεσίας
3. ονοματοδοσία της φωτογραφίας που θα ληφθεί
4. λήψη φωτογραφίας
5. αποστολή φωτογραφίας
6. ενημέρωση για επιτυχία ή αποτυχία αποστολής

4.7. Μενού Επιλογών

Οι δραστηριότητες που αποτελούν το μενού επιλογών συνδέονται με το “ActionBar” μιας κι αυτό ελέγχει το μενού. Το μενού περιλαμβάνει δύο δραστηριότητες (άμεσα τουλάχιστον):

- τη δραστηριότητα αναζήτησης και
- τη δραστηριότητα ρυθμίσεων του χρήστη

Κάθε μια από αυτές τις δραστηριότητες περιλαμβάνει τους δικούς της διαλόγους για την επίτευξη της ομαλής κι επιθυμητής λειτουργίας τους.

4.7.1. Δραστηριότητα Αναζήτησης

Η εφαρμογή Fire προσφέρει τη δυνατότητα αναζήτησης παλαιότερων πυρκαγιών, όπως ακριβώς και η υπηρεσία “Fire Monitoring Service”.

Η αναζήτηση μπορεί να έχει μία από τις εξής τέσσερις μορφές:

- αναζήτηση μιας μόνο χρονικής στιγμής
- αναζήτηση μιας χρονικής περιόδου μέχρι και έξι ημερών
- αναζήτηση μιας μόνο χρονικής στιγμής, όπου οι πυρκαγιές που θα επιστραφούν θα πρέπει να περιλαμβάνουν και το ζητούμενο σημείο του χάρτη
- αναζήτηση μιας χρονικής περιόδου μέχρι έξι ημερών, με ταυτόχρονη ικανοποίηση του ζητούμενου σημείου του χάρτη (δηλαδή θα πρέπει να περιλαμβάνεται το σημείο μέσα στα φλεγόμενα πολύγωνα που θα επιστραφούν)

Η δραστηριότητα αυτή περιλαμβάνει μια σειρά από διαλόγους και μια δραστηριότητα προκειμένου να επιλεγθούν τα κριτήρια της αναζήτησης. Χρειάζονται δύο πανομοιότυποι διάλογοι που φροντίζουν για τον ορισμό της ημερομηνίας και της ώρας, όπως επίσης και μια δραστηριότητα για τον ορισμό του σημείου.

Το κουμπί της αναζήτησης θα ενεργοποιηθεί μόλις τεθεί μια ημερομηνία και ώρα αναζήτησης, δηλαδή τη στιγμή όπου θα μπορεί να πραγματοποιηθεί αναζήτηση με κάποια από τις παραπάνω τέσσερις έγκυρες μορφές, πιο συγκεκριμένα με την πρώτη μορφή.

Ο διάλογος του ορισμού της ημερομηνίας κι ώρας, ουσιαστικά αποτελείται από δυο διαλόγους. Ο ένας αφορά τον ορισμό της ημερομηνίας και ο επόμενος τον ορισμό της ώρας. Ο διάλογος επιλογής ημερομηνίας προβάλλει ένα ημερολόγιο και δύο κουμπιά, ένα για ακύρωση της επιλογής και ένα για την πραγματοποίηση της επιλογής και εμφάνισης του διαλόγου ορισμού ώρας. Ο διάλογος επιλογής ώρας με τη σειρά του περιέχει δύο κουμπιά, ένα για την επιστροφή στο

διάλογο επιλογής ημερομηνίας και ένα για την ολοκλήρωση επιλογής ημερομηνίας και ώρας.

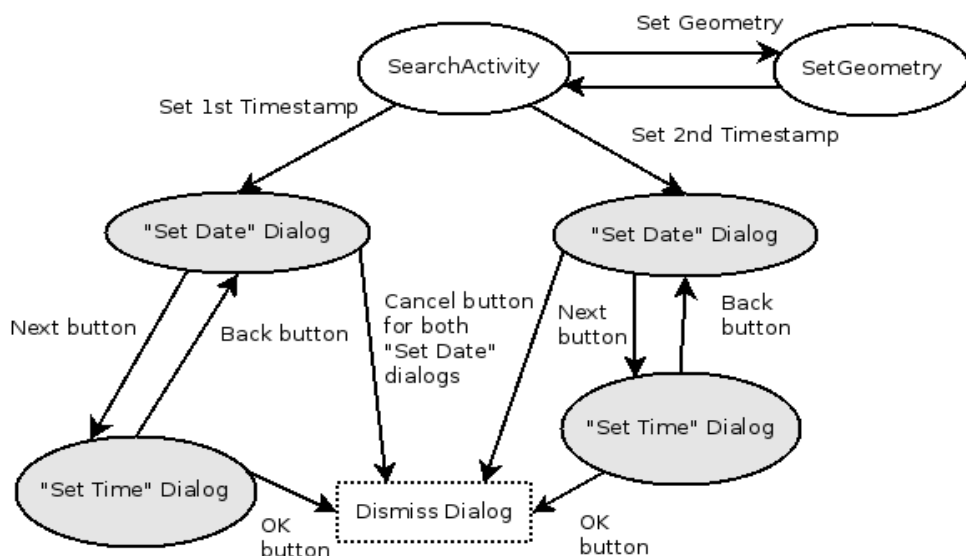
Το κουμπί επιλογής σημείου στο χάρτη πυροδοτεί μια νέα δραστηριότητα, τη δραστηριότητα “SetGeometry”. Η “SetGeometry” προβάλλει ένα χάρτη της Google. Ο χρήστης μπορεί να επιλέξει ένα σημείο κρατώντας πατημένο παρατεταμένα το δάχτυλό του στο σημείο αυτό. Έπειτα πατά το κουμπί οριστικής επιλογής και επιστρέφει στην οθόνη της αναζήτησης μαζί με το σημείο επιλογής του που θα συμπεριληφθεί στην αναζήτηση που θα πραγματοποιηθεί.

Τέλος, με την πίεση του κουμπιού αναζήτησης, η δραστηριότητα ελέγχει την ύπαρξη σύνδεσης με το διαδίκτυο. Στην περίπτωση, όπου είναι εφικτή η σύνδεση η δραστηριότητα “Results” αναλαμβάνει τη σύνδεση με το server της υπηρεσίας και την προβολή των αποτελεσμάτων που θα λάβει από αυτό. Στην αντίθετη περίπτωση, θα ενημερώσει το χρήστη ότι υπάρχει κάποιο πρόβλημα με τη σύνδεση.

Η δραστηριότητα “Results” που ακολουθεί τη δραστηριότητα της αναζήτησης, αναλαμβάνει να φέρει εις πέρας τρία έργα:

- τη λήψη των κριτηρίων αναζήτησης από τη δραστηριότητα της αναζήτησης
- την αποστολή του ερωτήματος στο server της υπηρεσίας
- τη λήψη των αποτελεσμάτων από το server και την προβολή τους ως πολύγωνα στο χάρτη

Ουσιαστικά, η “Results” πραγματοποιεί την αναζήτηση ενώ η “SearchActivity” καθορίζει τα κριτήρια του χρήστη. Το διάγραμμα με την επιλογή των κριτηρίων φαίνεται παρακάτω:



Σχήμα 10: Τι συμβαίνει όταν ο χρήστης πατά το κουμπί για να ορίσει το κάθε κριτήριο

Στο παραπάνω σχήμα διακρίνουμε πως από τη δραστηριότητα “SetGeometry” ο έλεγχος επιστρέφει στη “SearchActivity”. Ο λόγος που συμβαίνει αυτό έχει ήδη εξηγηθεί, όμως έχει νόημα να εξηγηθεί και ο τρόπος με τον οποίο λειτουργεί αυτό το σημείο.

Κατ' αρχάς, η δραστηριότητα της αναζήτησης καλεί τη δραστηριότητα “SetGeometry” με σκοπό να πάρει το αποτέλεσμα της επιλογής της τοποθεσίας του χρήστη. Η κλήση αυτή πραγματοποιείται με τις δύο παρακάτω γραμμές κώδικα:

```
Intent intent = new Intent(SearchActivity.this,
                           SetGeometry.class);
startActivityForResult(intent, 1);
```

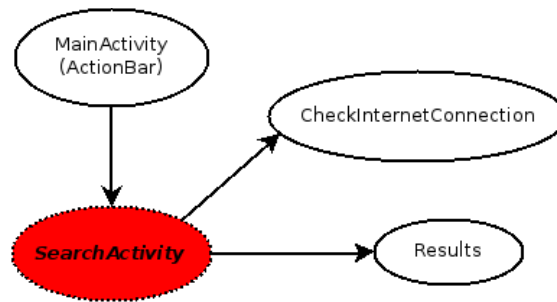
Απόσπασμα Κώδικα 2: Κλήση της δραστηριότητας "SetGeometry" με σκοπό την επιστροφή ενός αποτελέσματος

Με τη σειρά της η δραστηριότητα “SetGeometry” θα πρέπει να επιστρέψει το αποτέλεσμα της τοποθεσίας που επιθυμεί να θέσει ως κριτήριο ο χρήστης. Εφόσον, οι τιμές του γεωγραφικού πλάτους (latitude) και μήκους (longitude) του ορισμένου σημείου έχουν αποθηκευτεί στις μεταβλητές τύπου double, lat και lon αντίστοιχα, αρκούν οι παρακάτω γραμμές για την επιστροφή αυτών των τιμών σε μορφή πίνακα στην προηγούμενη δραστηριότητα:

```
double latlon[] = { lat, lon };
Intent returnIntent = new Intent();
returnIntent.putExtra("latlon", latlon);
setResult(RESULT_OK, returnIntent);
```

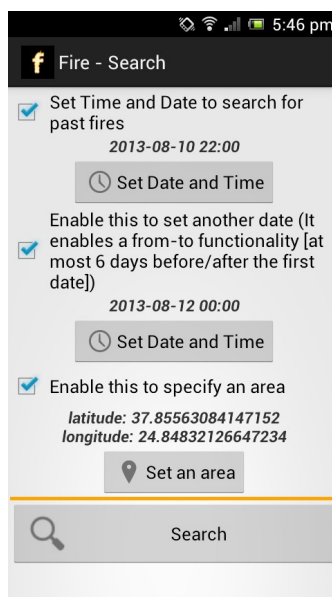
Απόσπασμα Κώδικα 3: Επιστροφή της τοποθεσίας που επέλεξε ο χρήστης

Η δραστηριότητα αναζήτησης είναι προσκολλημένη στο “ActionBar” με την έννοια ότι αυτό διαχειρίζεται το μενού και μέσω αυτού πυροδοτείται. Επιπλέον, σύμφωνα και με τα παραπάνω φαίνεται πως η “SearchActivity” πυροδοτεί τη “Results”. Η σχέση μεταξύ της “SearchActivity” και των υπολοίπων μερών της εφαρμογής (εξαιρούνται τα μέρη που σχετίζονται με τον ορισμό των κριτηρίων) φαίνεται παρακάτω:

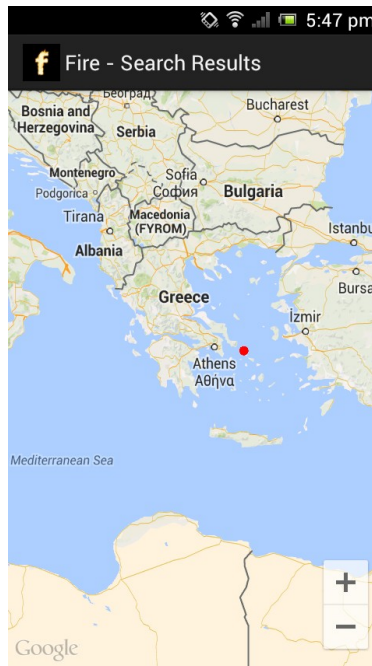


Σχήμα 11: Η πλήρης λειτουργία της αναζήτησης

Ένα παράδειγμα εκτέλεσης της αναζήτησης με όλα τα πεδία ενεργά φαίνεται παρακάτω:



Εικόνα 7: Η δραστηριότητα της αναζήτησης, όπου ο χρήστης επιλέγει τα κριτήρια αναζήτησης



Εικόνα 8: Τα αποτελέσματα της αναζήτησης

4.7.2. Δραστηριότητα Ρυθμίσεων Χρήστη

Η εφαρμογή προσφέρει τη δυνατότητα τροποποίησης κάποιων παραμέτρων ώστε να προσεγγίζει καλύτερα τις ανάγκες του χρήστη. Για παράδειγμα, ένας χρήστης μπορεί να επιθυμεί την παρακολούθηση σε πραγματικό χρόνο πυρκαγιών μιας συγκεκριμένης περιόδου ενώ κάποιος άλλος να χρειάζεται να είναι διαρκώς ενήμερος για τις πιο πρόσφατες πυρκαγιές. Η εφαρμογή Fire είναι σε θέση να ικανοποιήσει τις ανάγκες και των δύο παραπάνω χρηστών.

Οι επιλογές που προσφέρονται, συνοπτικά, είναι οι παρακάτω:

- αλλαγή ημερομηνίας και ώρας εφαρμογής σε κάποια παλαιότερη
- αλλαγή τροποποιημένης ημερομηνίας και ώρας στην ημερομηνία και ώρα συσκευής
- ρύθμιση συχνότητας ενημερώσεων εφαρμογής

Η τροποποίηση της ημερομηνίας και ώρας της εφαρμογής, δηλαδή η πρώτη επιλογή, πραγματοποιείται με τη χρήση διαλόγων ακριβώς όπως συνέβαινε και στη δραστηριότητα της αναζήτησης. Πιο αναλυτικά, μόλις ο χρήστης επιλέξει την αλλαγή της ημερομηνίας και ώρας, θα εμφανιστεί ένας διάλογος με ένα ημερολόγιο. Ο χρήστης μπορεί να περιηγηθεί στο ημερολόγιο

αυτό και να επιλέξει την ημερομηνία που επιθυμεί. Έπειτα, μπορεί να επιλέξει είτε να προχωρήσει στην επιλογή της ώρας, είτε να ακυρώσει την ημερομηνία που όρισε. Δεδομένου ότι θα επιλέξει να προχωρήσει στον ορισμό της ώρας, θα εμφανιστεί ένας καινούριος διάλογος για την επιλογή της ώρας. Όταν επιλέξει και ώρα, ο χρήστης μπορεί είτε να μεταβεί ξανά στον προηγούμενο διάλογο και να διαλέξει διαφορετική ημερομηνία ή να πατήσει το κουμπί “OK” και να καταχωρήσει την επιλογή του. Με αυτή την επιλογή, η δραστηριότητα συνδέεται με την υπηρεσία “FiresCache” ενημερώνοντας την πως θα πρέπει να διαγράψει το αρχείο με τις πυρκαγιές μόλις τερματιστεί η λειτουργία της. Αυτό συμβαίνει για να εξασφαλιστεί πως θα συμπεριληφθούν παλαιότερες πυρκαγιές από αυτές που υπάρχουν στο αρχείο σε περίπτωση που αλλάξει η ημερομηνία σε μια παλαιότερη που όμως εμπεριέχεται στο πλαίσιο των τριών ημερών που αναζητεί η “FiresCache” πυρκαγιές.

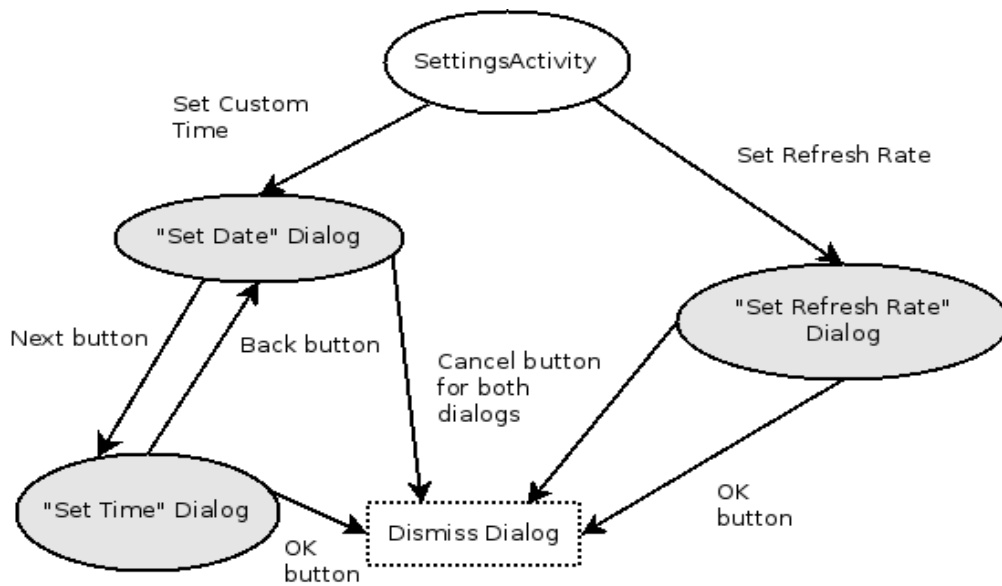
Η δεύτερη επιλογή αποτελεί πρακτικά την ακύρωση των αλλαγών της πρώτης. Δεν εμφανίζει κάτι στο χρήστη, αλλά την επόμενη φορά που θα ξεκινήσει η εφαρμογή θα πάρει την ημερομηνία και ώρα της συσκευής για τη λειτουργία της.

Η τελευταία επιλογή φροντίζει για τη συχνότητα της επικοινωνίας της εφαρμογής με το server για τη λήψη νεότερων αποτελεσμάτων. Για την επίτευξη αυτού του σκοπού προβάλλεται ένας διάλογος με τις δυνατές συχνότητες που μπορούν να επιλεγθούν. Αυτές είναι οι παρακάτω τρεις:

- ενημέρωση ανά πέντε λεπτά
- ενημέρωση ανά δέκα λεπτά
- ενημέρωση ανά δεκαπέντε λεπτά

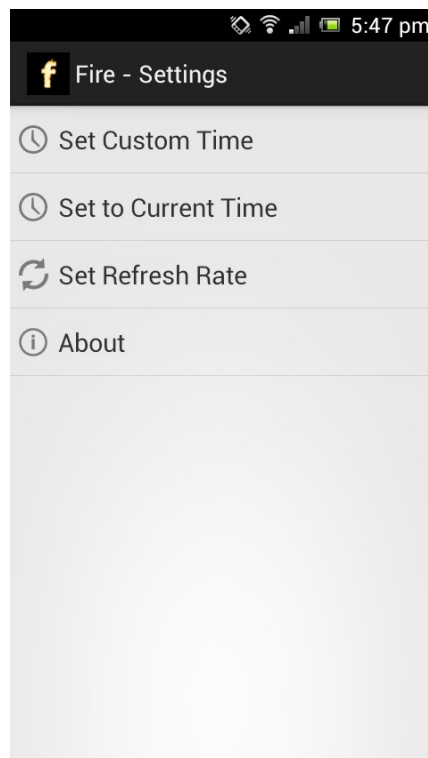
Μόλις λάβει χώρα κάποια αλλαγή, αμέσως αποθηκεύεται στο αρχείο επιλογών του χρήστη στην κάρτα SD. Αυτό συμβαίνει για την ενημέρωση της εφαρμογής σχετικά με τις επιλογές που πραγματοποίησε ο χρήστης και να συμμορφωθεί με αυτές.

Οι διάλογοι που απαιτούνται για τις ρυθμίσεις φαίνονται στο παρακάτω διάγραμμα:



Σχήμα 12: Οι απαραίτητοι διάλογοι για τις ρυθμίσεις

Η δραστηριότητα “SettingsActivity” συνδέεται μόνο με την κύρια δραστηριότητα λόγω του “ActionBar” και με τη μοναδική υπηρεσία της εφαρμογής. Επιπλέον, περιλαμβάνει τη σελίδα “About” της εφαρμογής.



Εικόνα 9: Η δραστηριότητα των ρυθμίσεων

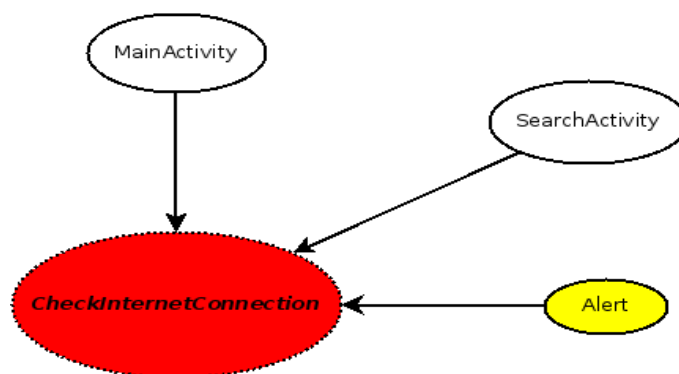
4.8. Βοηθητικές Κλάσεις

4.8.1. Έλεγχος Σύνδεσης Διαδικτύου (CheckInternetConnection)

Αυτή η βοηθητική κλάση πραγματοποιεί τον έλεγχο σύνδεσης με το διαδίκτυο όποτε αυτό είναι απαραίτητο. Η μέθοδος `isConnectingToInternet()` επιστρέφει μια μεταβλητή boolean, “true” αν υπάρχει σύνδεση, αλλιώς “false”.

Με αυτή τη βοηθητική κλάση συνδέονται η κύρια δραστηριότητα κατά την έναρξη της εφαρμογής, η δραστηριότητα της αναζήτησης μόλις πατηθεί το κουμπί της έναρξης της αναζήτησης και το αντικείμενο του συναγερμού μόλις αυτό εκκινείται.

Σχηματικά:



Σχήμα 13: Από τι χρησιμοποιείται η κλάση "CheckInternetConnection"

4.8.2. Εύρεση Τοποθεσίας (LocationFinder)

Χρησιμοποιεί στην εύρεση της τοποθεσίας της συσκευής. Αξιοποιεί τα GooglePlay Services προκειμένου να το επιτύχει αυτό απαιτώντας την ύπαρξη αυτών των υπηρεσιών στη συσκευή για να λειτουργήσει σωστά.

Χρησιμοποιείται μόνο από το αντικείμενο “Alert”, προκειμένου να προσθέσει την τοποθεσία στην ονομασία της φωτογραφίας που θα λάβει και θα αποστείλει στο server της υπηρεσίας.

Η κύρια δραστηριότητα χρησιμοποιεί αυτή τη βοηθητική κλάση για να βεβαιωθεί πως υπάρχουν τα GooglePlay Services εγκατεστημένα στη συσκευή.

4.8.3. Σύνδεση με Server (ServerConnection)

Προκειμένου να είμαστε βέβαιοι πως μόνο η εφαρμογή μας είναι σε θέση να συναλλάσσεται με το web service μας, θα πρέπει να ληφθούν ορισμένα σχετικά μέτρα. Ένα τέτοιο μέτρο προσφέρει η βοηθητική κλάση “ServerConnection”.

Στην κλάση αυτή, είναι αποθηκευμένος ένας κωδικός αναγνώρισης της εφαρμογής από το server. Ο ρόλος της κλάσης αυτής όμως, είναι σημαντικότερος.

Η συγκεκριμένη κλάση φροντίζει ώστε η επικοινωνία με το διακομιστή να είναι ασφαλής. Αυτό επιτυγχάνεται με τη χρήση του πρωτοκόλλου HTTPS, του SSL και του X.509 όπου καταφέρνουν την κρυπτογράφηση της επικοινωνίας και την αυθεντικοποίηση μεταξύ των συναλλασσομένων μερών.

4.9. Υπηρεσία FiresCache

4.9.1. Γενική Λειτουργία

Η υπηρεσία “FiresCache”, θα μπορούσε να χαρακτηριστεί ως η καρδιά της εφαρμογής. Αναλαμβάνει τα πιο περίπλοκα και (συχνά) χρονοβόρα έργα, συνδέεται με τις περισσότερες βασικές δραστηριότητες και αντικείμενα της εφαρμογής και απαιτείται η απρόσκοπτη λειτουργία της καθ' όλη τη διάρκεια της εκτέλεσης της εφαρμογής.

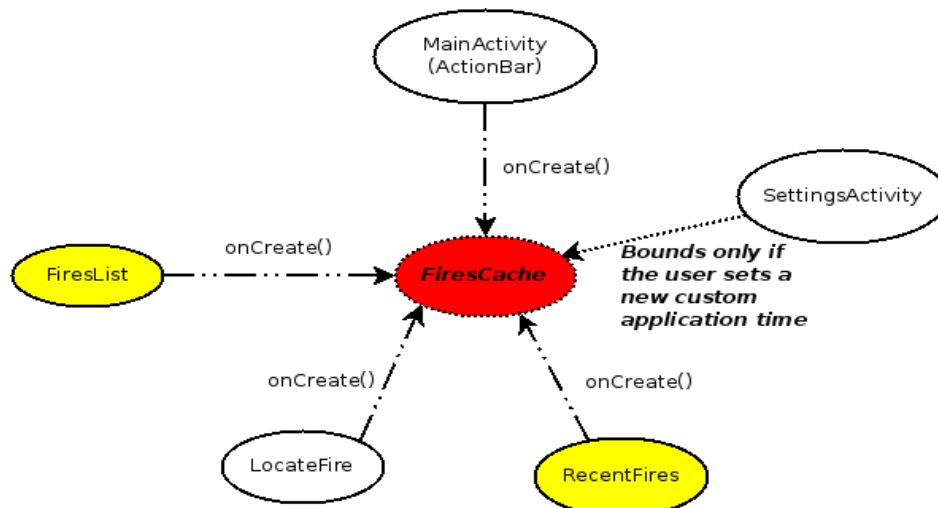
Δημιουργείται κατά τη δημιουργία της κύριας δραστηριότητας και ολοκληρώνει τον κύκλο ζωής της τη στιγμή που θα αποδεσμευτεί απ' όλα τα συνδεδεμένα με αυτή μέρη της εφαρμογής.

Τα έργα που επιτελεί είναι τα εξής:

- διαβάζει το αρχείο ρυθμίσεων του χρήστη για να αποφασίσει την ημερομηνία και την ώρα της εφαρμογής καθώς και το χρονικό διάστημα που θα πρέπει να στέλνει τακτικά ένα ερώτημα προκειμένου να ενημερωθεί η λίστα της
- διαβάζει το cache file με τις πυρκαγιές και διασταυρώνει με την ημερομηνία και ώρα της εφαρμογής ποιες από αυτές θα πρέπει να φορτώσει στη λίστα της
- ενημερώνει τη λίστα της ανά τακτά χρονικά διαστήματα
- αποθηκεύει τη λίστα στο cache file

- διαγράφει το cache file εάν ζητηθεί από τη δραστηριότητα των ρυθμίσεων
- διαθέτει διάφορες μεθόδους για την επιστροφή δεδομένων των πυρκαγιών της λίστας της

Τα συνδεδεμένα με αυτή μέρη φαίνονται παρακάτω:



Σχήμα 14: Ποια μέρη και πότε συνδέονται με την υπηρεσία

Όπως φαίνεται, όλα τα μέρη που συνδέονται μαζί της, πραγματοποιούν τη δέσμευση κατά τη δημιουργία τους. Μόνο η δραστηριότητα ρυθμίσεων δεν το κάνει κι αυτό εξηγείται από το γεγονός ότι ενδεχομένως να μη χρειάζεται.

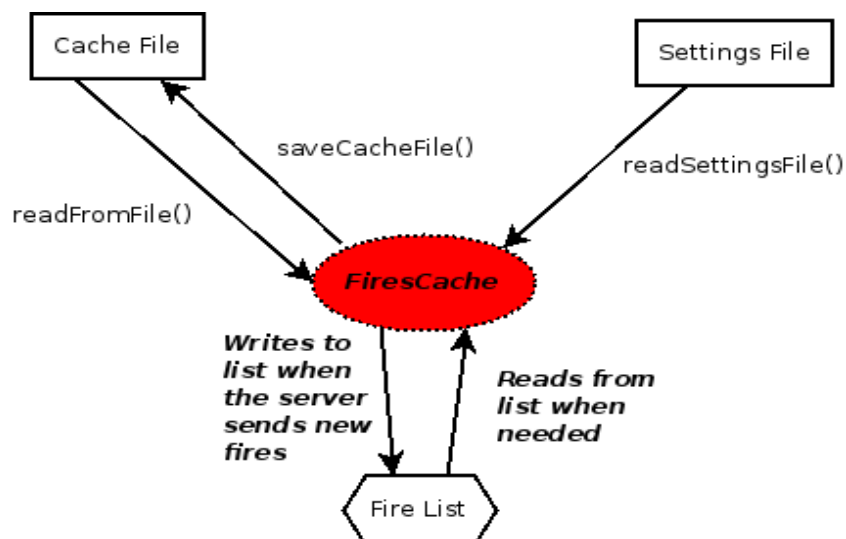
Όλα τα δεδομένα που δέχεται κι επεξεργάζεται είναι σε μορφή JSON. Οι πυρκαγιές που δέχεται από το server, οι πυρκαγιές που δέχεται από το cache file, οι πυρκαγιές που αποθηκεύει στο cache file, ακόμη και οι ρυθμίσεις από το αρχείο δεδομένων είναι μορφής JSON.

Οι λειτουργίες που εκτελεί με τη σειρά εκτέλεσης τους φαίνεται στην παρακάτω λίστα:

1. έναρξη υπηρεσίας
2. ανάγνωση του αρχείου ρυθμίσεων και αρχικοποίηση της λίστας των πυρκαγιών
3. πρώτη εκτέλεση για αναζήτηση των πυρκαγιών
4. αποστολή ερωτήματος στο server ανά τακτά χρονικά διαστήματα για ενημέρωση της λίστας (το χρονικό διάστημα καθορίζεται από το χρήστη μέσω της ανάγνωσης του αρχείου ρυθμίσεων κατά την έναρξη της υπηρεσίας)
5. προσθήκη των νέων πυρκαγιών στη λίστα και καταγραφή τους στο cache file μόλις ληφθούν από το server
6. “κλείσιμο” αρχείων που έχουν ανοιχτεί

7. τερματισμός υπηρεσίας

Τα βήματα τέσσερα και πέντε μπορεί να επαναληφθούν πάνω από μια φορά. Η σχέση μεταξύ της υπηρεσίας και των αρχείων και δομών δεδομένων που χρησιμοποιεί φαίνεται στο παρακάτω σχήμα:



Σχήμα 15: Πώς και πότε επικοινωνεί η "FiresCache" με τα αρχεία και τη λίστα των πυρκαγιών της

4.9.2. Μέθοδοι Μεταφοράς Δεδομένων από τη Λίστα Πυρκαγιών

Οι συνδεδεμένες δραστηριότητες και τα συνδεδεμένα με την υπηρεσία αντικείμενα, έχουν ως στόχο την απόσπαση πληροφοριών από τη λίστα πυρκαγιών της. Η μεταφορά δεδομένων από τη λίστα σε κάποιο άλλο μέρος της εφαρμογής, θα πραγματοποιηθεί με τη βοήθεια μιας μεθόδου.

Υπάρχουν τρεις τέτοιες μέθοδοι στην υπηρεσία, όπου κάθε μια χρησιμοποιείται για διαφορετικούς λόγους και για μεταφορά δεδομένων με βάση διαφορετικά κριτήρια.

Η πρώτη μέθοδος αφορά τη μεταφορά όλων των πυρκαγιών της λίστας. Συνήθως χρησιμοποιείται από κάποιο μέρος της εφαρμογής που αρχικοποιείται και χρειάζεται να γνωρίζει όλες τις πυρκαγιές που υπάρχουν στη λίστα μέχρι αυτή τη στιγμή. Δεν απαιτεί κάποια παράμετρο για την κλήση της κι επιστρέφει μια λίστα με το σύνολο των δεδομένων που έχουν αποθηκευτεί στη λίστα για μια πυρκαγιά. Η υπογραφή της μεθόδου είναι η εξής:

`getFirstTimeFires()`

Απόσπασμα Κώδικα 4: Η υπογραφή της μεθόδου που επιστρέφει όλες τις πυρκαγιές της λίστας

Η δεύτερη μέθοδος προσφέρει τη δυνατότητα της ενημέρωσης των υπολοίπων μερών της εφαρμογής με τις τελευταίες πυρκαγιές που προστέθηκαν στη λίστα. Ούτε αυτή δέχεται κάποια παράμετρο και επιστρέφει μια λίστα με τις τελευταίες πυρκαγιές. Η υπογραφή της φαίνεται παρακάτω:

`getFires()`

Απόσπασμα Κώδικα 5: Η υπογραφή της μεθόδου που επιστρέφει τις πυρκαγιές της τελευταίας ενημέρωσης

Η τελευταία μέθοδος φροντίζει για την επιστροφή μιας πυρκαγιάς. Αυτή η μέθοδος χρησιμοποιείται μοναχά από τη δραστηριότητα “LocateFire” και σε αντίθεση με τις άλλες δυο μεθόδους απαιτεί δύο παραμέτρους. Οι απαιτούμενες παράμετροι είναι η θέση της πυρκαγιάς στη λίστα που εμφανίζεται στο χρήστη μέσω του αντικειμένου “FiresList” καθώς και η ημερομηνία και χρόνος σε μορφή long που η λίστα της “FiresList” ενημερώθηκε για τελευταία φορά. Οι ζητούμενες παράμετροι είναι απαραίτητες για τον εντοπισμό της σωστής πυρκαγιάς. Θα αρκούσε μόνο η θέση στη λίστα της “FiresList” εάν δεν υπήρχε η περίπτωση να έχει ενημερωθεί η λίστα της υπηρεσίας σε μεταγενέστερο χρόνο από αυτόν της τελευταίας ενημέρωσης της “FiresList”. Επειδή όμως, η περίπτωση που περιγράφεται παραπάνω είναι πολύ πιθανή, το αποτέλεσμα που θα επέστρεφε αυτή η μέθοδος θα ήταν διαφορετικό από αυτό που θα ζητούσε ο χρήστης. Η υπογραφή της τελευταίας μεθόδου είναι η εξής:

`getSingleFire(int position, long lastRefreshed)`

Απόσπασμα Κώδικα 6: Η υπογραφή της μεθόδου που επιστρέφει μια πυρκαγιά με βάση τη θέση της πυρκαγιάς στη λίστα της “FiresList”

4.9.3. Συγχρονισμός μεταξύ Ανάγνωσης και Γραφής της Λίστας

Για να αποφευχθεί η σύγχυση μεταξύ των δεδομένων της λίστας της υπηρεσίας και των

δεδομένων που διατηρούν τα υπόλοιπα μέρη, θα πρέπει να υπάρχει ένας μηχανισμός συγχρονισμού.

Ο μηχανισμός αυτός, αποτελείται από ένα κλείδωμα εγγραφής – ανάγνωσης. Κάθε φορά που η υπηρεσία γράφει στη λίστα της, κλειδώνει το κλείδωμα της εγγραφής. Με αυτόν τον τρόπο, δεν επιτρέπει σε κανένα άλλο μέρος της εφαρμογής την ανάγνωση της λίστας και αν ζητηθεί η ανάγνωση, τότε αυτό το μέρος αναγκάζεται να περιμένει να ελευθερωθεί το κλείδωμα της εγγραφής. Εφόσον το κλείδωμα της εγγραφής είναι ξεκλειδωτό, μπορούν να πραγματοποιηθούν οι αναγνώσεις από τα υπόλοιπα μέρη. Να σημειωθεί πως το κλείδωμα ανάγνωσης δεν είναι αποκλειστικό όπως το κλείδωμα της εγγραφής, δηλαδή μπορούν παραπάνω από ένα μέρη να αναγνώσουν μαζί τη λίστα, όμως δεν μπορεί να λάβει χώρα ταυτόχρονα κάποια εγγραφή στη λίστα. Το κλείδωμα αυτό έχει δημιουργηθεί με τον εξής τρόπο:

```
// variable used for synchronization  
private ReadWriteLock lock = new ReentrantReadWriteLock();
```

Απόσπασμα Κώδικα 7: Το κλείδωμα εγγραφής - ανάγνωσης

4.10. Απαιτούμενα Δικαιώματα Εφαρμογής και άλλα Χαρακτηριστικά του AndroidManifest.xml

Κάθε εφαρμογή Android περιέχει ένα αρχείο με όνομα AndroidManifest.xml, το οποίο περιλαμβάνει τα βασικά στοιχεία της εφαρμογής και κάποια δικαιώματα που απαιτούνται για την ορθή εκτέλεση της.

Τα δικαιώματα έχουν να κάνουν με:

- τη σύνδεση στο διαδίκτυο
- το χάρτη της Google
- τη δυνατότητα εγγραφής στην κάρτα SD της συσκευής
- τη δυνατότητα εύρεσης της τοποθεσίας

Η εφαρμογή θα πρέπει να έχει τη δυνατότητα σύνδεσης με το διαδίκτυο για πολλούς λόγους. Είτε για την επικοινωνία της με το server της υπηρεσίας, είτε για το κατέβασμα των χαρτών από την Google. Επιπλέον, είναι απαραίτητο να μπορεί να αντιληφθεί την κατάσταση της σύνδεσης. Προκειμένου να βρίσκεται σε θέση η εφαρμογή να επιτύχει αυτά, χρειάζονται οι εξής δυο γραμμές στο AndroidManifest.xml:

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
```

Απόσπασμα Κώδικα 8: Τα δικαιώματα για τον έλεγχο της σύνδεσης στο διαδίκτυο αλλά και για την επίτευξη της σύνδεσης αυτής

Όσον αφορά το χάρτη της Google, απαιτείται η πρόσβαση στις υπηρεσίες της Google που θα επιτευχθεί με αυτή τη γραμμή:

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

Απόσπασμα Κώδικα 9: Το δικαίωμα πρόσβασης στις υπηρεσίες της Google

Η δυνατότητα εγγραφής στην κάρτα SD της συσκευής χρειάζεται για δύο λόγους:

- για την αποθήκευση και ανάγνωση των αρχείων με τις ρυθμίσεις και το cache file, όσο και για την αποθήκευση των φωτογραφιών που ενδεχομένως να ληφθούν μέσω του αντικειμένου “Alert”
- για το caching του χάρτη της Google

Πρέπει να σημειωθεί πως από το API 19 και ύστερα του Android, παραχωρείται το συγκεκριμένο δικαίωμα χωρίς τη ρητή αναφορά του στο AndroidManifest.xml. Επομένως, το δικαίωμα αυτό απαιτείται μέχρι και το API 18 όπως φαίνεται και παρακάτω:

```
<uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE"
    android:maxSdkVersion="18" />
```

Απόσπασμα Κώδικα 10: Το δικαίωμα για γραφή και ανάγνωση στην κάρτα SD

Τέλος, η εφαρμογή χρειάζεται να γνωρίζει την τοποθεσία της συσκευής σε ορισμένες περιπτώσεις. Θα πρέπει να είναι δυνατό να εντοπίζει την τοποθεσία είτε μέσω διαδικτύου (“ACCESS_COARSE_LOCATION”), είτε με τη βοήθεια του GPS (“ACCESS_FINE_LOCATION”). Τα δικαιώματα που απαιτούνται σχετικά από το AndroidManifest.xml της εφαρμογής είναι τα παρακάτω:

```

<!-- Required to show current location -->
<uses-permission android:name="android.permission.ACCESS_GPS" />
<uses-permission android:name="android.permission.ACCESS_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

```

Απόσπασμα Κώδικα 11: Τα δικαιώματα που απαιτούνται για τον εντοπισμό της τοποθεσίας της συσκευής

Ένα σημείο που έχει ενδιαφέρον να μελετηθεί στο AndroidManifest.xml, ακόμη κι αν δεν αποτελεί δικαίωμα είναι η απαίτηση της ύπαρξης του OpenGL ES version 2. Προκειμένου να προβληθεί ο χάρτης, θα πρέπει να υποστηρίζεται η δεύτερη έκδοση της OpenGL ES, σε αντίθετη περίπτωση η εφαρμογή δε θα μπορεί να εκτελεστεί στη συσκευή.

```

<!-- Required OpenGL ES 2.0. for Maps V2 -->
<uses-feature
    android:glEsVersion="0x00020000"
    android:required="true" />

```

Απόσπασμα Κώδικα 12: Η απαίτηση για την ύπαρξη του "OpenGL ES version 2"

4.11. Υλοποίηση Server

4.11.1. Αποστολή Αποτελεσμάτων στην Εφαρμογή

Όπως έχει ήδη αναφερθεί, η υπηρεσία “FiresCache” επικοινωνεί με το server κατά την εκκίνησή της ώστε να ενημερώσει τα ήδη υπάρχοντα στο cache file δεδομένα ή για να λάβει τα πρώτα αποτελέσματα. Επιπλέον, επικοινωνεί μαζί του ανά τακτά χρονικά διαστήματα για την ενημέρωση της λίστας της.

Πέραν της υπηρεσίας “FiresCache” και η δραστηριότητα “Results” που ακολουθεί τη δραστηριότητα “SearchActivity”, επικοινωνεί με το server για να προβάλει στο χρήστη τις πυρκαγιές που ζήτησε.

Γι' αυτό το λόγο έχει στηθεί ένας GlassFish Server κι έχει δημιουργηθεί ένα RESTful Java Web Service. Το web service αυτό, περιμένει μια κλήση από την εφαρμογή που θα περιέχει το πολύ μέχρι τέσσερις μεταβλητές. Οι μεταβλητές αυτές είναι:

- ένας αριθμός τύπου long που συμβολίζει μια χρονική στιγμή σε χιλιοστά του δευτερολέπτου
- ένας δεύτερος αριθμός τύπου long που συμβολίζει κι αυτός με τη σειρά του μια χρονική στιγμή σε χιλιοστά του δευτερολέπτου
- ένας δεκαδικός αριθμός τύπου double που εκφράζει το γεωγραφικό πλάτος
- ένας δεκαδικός αριθμός τύπου double που εκφράζει το γεωγραφικό μήκος

Η “FiresCache” πραγματοποιεί αιτήσεις μόνο με τις δύο πρώτες μεταβλητές ενώ η “Results” μπορεί να θέσει ό,τι κριτήρια επιθυμεί ο χρήστης, αρκεί να έχει θέσει μια τιμή για την πρώτη μεταβλητή. Αξίζει να σημειωθεί, πως οι δύο τελευταίες μεταβλητές αλληλοεξαρτώνται στο service με την έννοια πως εάν δεν έχει οριστεί μία εξ' αυτών, ακυρώνεται και η άλλη. Αυτό είναι λογικό, διότι οι δύο τελευταίες μεταβλητές σε συνδυασμό συμβολίζουν ένα σημείο στο χάρτη.

Το service θα πρέπει να μπορεί να ανταπεξέλθει στους παρακάτω συνδυασμούς:

- δέχεται μόνο την πρώτη μεταβλητή του χρόνου
- δέχεται και τις δύο μεταβλητές του χρόνου
- δέχεται μια μεταβλητή του χρόνου (την πρώτη) και τις δύο τελευταίες που συμβολίζουν ένα σημείο στο χάρτη
- δέχεται και τις τέσσερις μεταβλητές

Οποιοσδήποτε άλλος συνδυασμός θα επιστρέψει ένα μήνυμα σφάλματος και δε θα φέρει κάποια αποτελέσματα.

Στην περίπτωση όπου εντοπιστεί κάποια από τις παραπάνω τέσσερις αποδεκτές μορφές, το service θα πραγματοποιήσει μια επερώτηση προς τη βάση προκειμένου να λάβει τα αποτελέσματα των πυρκαγιών που ζήτησε η εφαρμογή. Στη συνέχεια, επεξεργάζεται τα αποτελέσματα αυτά μορφοποιώντας τα σε κατανοητή μορφή για την εφαρμογή. Η μορφή αυτή είναι JSON και για κάθε πυρκαγιά περιλαμβάνει τα εξής στοιχεία:

- το δήμο στον οποίο ανήκει η φλεγόμενη περιοχή (municipality)
- τη χρονική στιγμή που έλαβε χώρα η μέτρηση και διαπιστώθηκε πως η πυρκαγιά είναι ενεργή (timestamp)
- τη γεωμετρία της πυρκαγιάς σε μορφή GeoJSON (the_geom)

Παρακάτω φαίνεται ένα παράδειγμα μιας πυρκαγιάς σε κατανοητή από την εφαρμογή

μορφή:

```
{  "municipality": "ΔΗΜΟΣ ΑΝΔΡΟΥ",  
  "timestamp": "2013-08-10 21:35:00",  
  "the_geom": { "type": "MultiPolygon",  
    "coordinates": [  
      [[ [24.8599, 37.8525], [24.9031, 37.8522], [24.9027, 37.8179], [24.8595, 37.8183], [24.8599, 37.8525]] ]  
    ]  
  }  
}
```

Απόσπασμα Κώδικα 13: Παράδειγμα μιας πυρκαγιάς σε κατανοητή για την εφαρμογή JSON μορφή

Τέλος, το service συγκεντρώνει όλες τις πυρκαγιές που εντόπισε σε μια συμβολοσειρά και τις επιστρέφει στην εφαρμογή. Η εφαρμογή, με τη σειρά της, επεξεργάζεται αυτή τη συμβολοσειρά και προβάλλει ή αποθηκεύει τα δεδομένα.

4.11.2. Αποθήκευση Φωτογραφίας

Το αντικείμενο “Alert” προσφέρει τη δυνατότητα συναγερμού, δηλαδή ο χρήστης μπορεί να ειδοποιήσει την υπηρεσία για μια πυρκαγιά. Προκειμένου να συμβεί αυτό, θα πρέπει να μεταφορτώσει μια φωτογραφία στο server μαζί με την τοποθεσία της συσκευής την ώρα της λήψης της φωτογραφίας, καθώς και την ώρα της λήψης.

Για τη διαδικασία της μεταφόρτωσης και της αποθήκευσης της φωτογραφίας στο server χρησιμοποιείται ένα PHP script. Το script αυτό, αρχικά, προσπαθεί να συνδεθεί με τη βάση στην οποία θα αποθηκεύσει τα στοιχεία της φωτογραφίας, τα οποία είναι:

- το όνομα της φωτογραφίας
- η χρονική στιγμή της λήψης της
- το γεωγραφικό πλάτος και μήκος της συσκευής τη στιγμή της λήψης

Τα δεδομένα που απαιτούνται για αποθήκευση στη βάση, αποσπώνται από το όνομα της φωτογραφίας κατά τη λειτουργία του script. Τέλος, η φωτογραφία αποθηκεύεται σε έναν καθορισμένο από το script φάκελο στο server.

ΚΕΦΑΛΑΙΟ 5: Συμπεράσματα

5.1. Γενικά Συμπεράσματα

Η εφαρμογή που δημιουργήθηκε είναι πλήρως λειτουργική κι έρχεται σε συμφωνία με τους στόχους που είχαν τεθεί πριν την υλοποίησή της.

Πιο συγκεκριμένα, έχουν υλοποιηθεί οι εξής στόχοι:

- υλοποίηση βασικών λειτουργιών υπηρεσίας
- επιπρόσθετη λειτουργία εφαρμογής για ειδοποίηση της υπηρεσίας για μια πυρκαγιά
- λήψη λίγων απαραίτητων δεδομένων για την ορθή λειτουργία της εφαρμογής (περιορισμένες κλήσεις στην υπηρεσία και επιστροφή λιγότερων δεδομένων μέσω του ακριβού δικτύου κινητής τηλεφωνίας αλλά και για να καταλαμβάνουν λιγότερο χώρο στην περιορισμένη μνήμη μιας φορητής συσκευής)
- παροχή κάποιων δυνατοτήτων παραμετροποίησης, που φέρνει την εφαρμογή πιο κοντά στις ανάγκες του χρήστη
- οπτικοποίηση των πυρκαγιών με δύο τρόπους (όπως συμβαίνει και στην ίδια την υπηρεσία) ώστε να επιλέγει ο χρήστης ποιος τρόπος είναι ο καλύτερος για να προβληθούν οι πυρκαγιές κάθε φορά
- caching πυρκαγιών για να ζητούνται και να αποστέλλονται όσο το δυνατόν λιγότερα αποτελέσματα από το server, αλλά χωρίς αρνητικά αποτελέσματα στην εγκυρότητα των δεδομένων στην εφαρμογή

5.2. Σκέψεις για Μελλοντικές Εκδόσεις

Το πλέον σημαντικό ζήτημα για μια εφαρμογή κινητού τηλεφώνου είναι ο χώρος που χρειάζεται στη μνήμη κατά τη λειτουργία της που είναι αρκετά περιορισμένος. Ένα άλλο ζήτημα που καλούνται να αντιμετωπίσουν προγραμματιστές έξυπνων κινητών είναι η λιγότερη δυνατή κατανάλωση δεδομένων από το δίκτυο κινητής τηλεφωνίας, καθώς πρόκειται για μια ακριβή υπηρεσία.

Επομένως, στόχος κάποιας επόμενης έκδοσης θα πρέπει να είναι η ακόμη μεγαλύτερη

εξοικονόμηση χώρου στη μνήμη, όσο και των δεδομένων που ζητούνται μέσω του δικτύου. Θα πρέπει να υπάρχουν λιγότερες εγγραφές στη μνήμη, μία για κάθε πολύγωνο που φλέγεται που θα κρατά τη συνολική διάρκεια της πυρκαγιάς, το δήμο και την ακριβή τοποθεσία του πολυγώνου.

Τελικά, σε κάθε ανανέωση θα επιστρέφεται μόνο μια εγγραφή για κάθε πολύγωνο ώστε να ανανεώνεται η λίστα. Και αυτή τη στιγμή, κάθε ανανέωση φέρνει μια εγγραφή για ένα πολύγωνο (δηλαδή η κατανάλωση δεδομένων μέσω του δικτύου κινητής τηλεφωνίας θα παραμείνει η ίδια) που θα ανανεώσει τη λίστα χωρίς να προσθέσει κάποια εγγραφή σε αυτήν (εκτός και αν πρόκειται για καινούριο πολύγωνο ή μια πυρκαγιά που είχε σβήσει για κάποιο χρονικό διάστημα) και συνεπώς δε θα δεσμεύει περισσότερο χρόνο στην πολύτιμη μνήμη της συσκευής.

Ενδιαφέρον θα είχε ακόμη, η προβολή των στοιχείων που λαμβάνει η εφαρμογή από το server κατά την αναζήτηση του χρήστη, είτε στο χάρτη, είτε σε μια λίστα για ευκολότερη προβολή των αποτελεσμάτων και την παρουσίαση όλων των δεδομένων.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Android Developers. (n.d.-a). Action Bar. Διαθέσιμο στο:

<https://developer.android.com/guide/topics/ui/actionbar.html>

Android Developers. (n.d.-b). App Resources. Διαθέσιμο στο:

<http://developer.android.com/guide/topics/resources/index.html>

Android Developers. (n.d.-c). ART and Dalvik. Διαθέσιμο στο:

<https://source.android.com/devices/tech/dalvik/index.html>

Android Developers. (n.d.-d). Content Providers. Διαθέσιμο στο:

<https://developer.android.com/guide/topics/providers/content-providers.html>

Android Developers. (n.d.-e). Fragments. Διαθέσιμο στο:

<https://developer.android.com/guide/components/fragments.html>

Android Developers. (n.d.-f). Keeping the Device Awake. Διαθέσιμο στο:

<https://developer.android.com/training/scheduling/wakelock.html>

Android Developers. (n.d.-g). LocationManager. Διαθέσιμο στο:

<https://developer.android.com/reference/android/location/LocationManager.html>

Android Developers. (n.d.-h). Managing the Activity Lifecycle. Διαθέσιμο στο:

<https://developer.android.com/training/basics/activity-lifecycle/index.html>

Android Developers. (n.d.-i). NotificationManager. Διαθέσιμο στο:

<https://developer.android.com/reference/android/app/NotificationManager.html>

Android Developers. (n.d.-j). Reading and Writing Logs. Διαθέσιμο στο:

<https://developer.android.com/tools/debugging/debugging-log.html>

Android Developers. (n.d.-k). Saving Data. Διαθέσιμο στο:

<https://developer.android.com/training/basics/data-storage/index.html>

Android Developers. (n.d.-l). Security with HTTPS and SSL. Διαθέσιμο στο:

<https://developer.android.com/training/articles/security-ssl.html>

Android Developers. (n.d.-m). Services. Διαθέσιμο στο:

<https://developer.android.com/guide/components/services.html>

Android Developers. (n.d.-n). TelephonyManager. Διαθέσιμο στο:

<https://developer.android.com/reference/android/telephony/TelephonyManager.html>

Bazilian, E. (2011, July 13). Facebook Launches App for Non-Smartphones. Διαθέσιμο στο:

<http://www.adweek.com/news/technology/facebook-launches-app-non-smartphones-133369>

Brady, P. (n.d.). *Anatomy and Physiology of an Android*. Παρουσιάστηκε στο Google I/O 2008.

Διαθέσιμο στο: <https://androidteam.googlecode.com/files/Anatomy-Physiology-of-an-Android.pdf>

- Bright, P. (2013, April 4). Google going its own way, forking WebKit rendering engine. Διαθέσιμο στο: <http://arstechnica.com/information-technology/2013/04/google-going-its-own-way-forking-webkit-rendering-engine/>
- Ionescu, D. (2012, April 26). Original Android Prototype Revealed During Google, Oracle Trial. Διαθέσιμο στο: http://www.pcworld.com/article/254539/original_android_prototype_revealed_during_google_oracle_trial.html
- Kovach, S. (2013, August 13). How Android Grew To Be More Popular Than The iPhone. Διαθέσιμο στο: <http://www.businessinsider.com/history-of-android-2013-8?op=1>
- Krazit, T. (2009, April 23). T-Mobile has sold 1 million G1 Android phones. Διαθέσιμο στο: <http://www.cnet.com/news/t-mobile-has-sold-1-million-g1-android-phones/>
- Kroah-Hartman, G. (2009, March 18). The Linux Staging Tree, what it is and is not. Διαθέσιμο στο: <http://www.kroah.com/log/linux/linux-staging-update.html>
- Love, R. (2013, May 13). What Are The Major Changes That Android Made To The Linux Kernel? Διαθέσιμο στο: <http://www.forbes.com/sites/quora/2013/05/13/what-are-the-major-changes-that-android-made-to-the-linux-kernel>
- OpenGL. (n.d.). OpenGL Overview. Διαθέσιμο στο: <https://www.opengl.org/about/>
- Open Handset Alliance. (2007, November 5). Industry Leaders Announce Open Platform for Mobile Devices. Διαθέσιμο στο: http://www.openhandsetalliance.com/press_110507.html
- Perez, J. C. (2007, November 6). Q&A With Google Android Developer. Διαθέσιμο στο: <http://www.techhive.com/article/139331/article.html>
- Smith, A. (2013, June 5). Smartphone Ownership 2013. Διαθέσιμο στο: <http://www.pewinternet.org/2013/06/05/smartphone-ownership-2013/>
- Smith, G. (2014, April 15). Heartbleed Bug Puts Millions Of Android Devices At Risk. Διαθέσιμο στο: http://www.huffingtonpost.com/2014/04/15/heartbleed-bug-android_n_5153812.html
- SQLite. (n.d.). About SQLite. Διαθέσιμο στο: <https://sqlite.org/about.html>
- Stultz, J. (2011, April 13). Android OS for Servers? Διαθέσιμο στο: http://www.elinux.org/images/8/89/Elc2011_stultz.pdf
- Tweedie, S. (2014, July 31). Android Is Crushing The Smartphone Market, And It's Not Even Close Anymore. Διαθέσιμο στο: <http://www.businessinsider.com/android-market-share-record-breaking-85-2014-7>
- WebKit. (n.d.). Home Page. Διαθέσιμο στο: <http://www.webkit.org/>
- Welch, C. (2013, April 16). Before it took over smartphones, Android was originally destined for cameras. Διαθέσιμο στο: <http://www.theverge.com/2013/4/16/4230468/android-originally-cameras>

designed-for-cameras-before-smartphones

Wilson, M. (2008, September 23). T-Mobile G1: Full Details of the HTC Dream Android Phone.

Διαθέσιμο στο: <http://gizmodo.com/5053264/t-mobile-g1-full-details-of-the-htc-dream-android-phone>