

Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Εφαρμογή Εύρεσης Πλησιέστερων Προσφορών Με Χρήση Γεωγραφικής Θέσης Nearest Offers Application Using Geofencing
Όνοματεπώνυμο Φοιτητή	Μιχαήλ Σκόρδος
Πατρώνυμο	Ιωάννης
Αριθμός Μητρώου	ΜΠΠΛ15065
Επιβλέπων	Ευθύμιος Αλέπης, Επίκουρος Καθηγητής

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Αλέπης Ε.

Επίκουρος Καθηγητής

Βίβρου Μ.

Καθηγήτρια

Πατσάκης Κ.

Επίκουρος Καθηγητής

Περίληψη

Σκοπός της εφαρμογής είναι να μπορούν οι χρήστες να ενημερώνονται την στιγμή που κάνουν τις αγορές τους για τυχόν προσφορές με βάση την τοποθεσία τους. Αυτό έχει ως αποτέλεσμα αρκετά πλεονεκτήματα και για τον αγοραστή-χρήστη αλλά και για τον πωλητή τα οποία θα αναλύσουμε παρακάτω. Επιπλέον δίνεται η δυνατότητα στον πωλητή-χρήστη της firebase να στέλνει ομαδικά μηνύματα σε όλους τους χρήστες της εφαρμογής μέσω email για να ενημερώνει για διάφορα θέματα που θα μπορούσαν να ενδιαφέρουν τον χρήστη όπως προσφορές, διαφήμιση κάποιου προϊόντος και τυχόν εκδηλώσεις.

Επίσης η εφαρμογή μας κρατά στατιστικά στοιχεία σχετικά με τον αριθμό των φορών που οι χρήστες τις εφαρμογής εισήλθαν σε μια περιοχή όπου αυτά μπορούν να τα βλέπουν και οι ίδιοι οι χρήστες. Έτσι μπορούμε να δούμε την κινητικότητα μιας περιοχής και να εξάγουμε χρήσιμα συμπεράσματα.

Βασικό στοιχείο της εφαρμογής μας προγραμματιστικά που διαφοροποιείτε είναι ότι προσφέρει τη δυνατότητα η γεωγραφική περιοχή όπου ενεργοποιείτε η αυτόματη αποστολή μηνύματος-notification στον χρήστη να βρίσκεται μέσα σε πολύγωνο και όχι ακτίνα κύκλου όπως συνηθίζεται. Αυτό έχει το πλεονέκτημα να έχουμε μεγαλύτερη ακρίβεια αφού μπορούμε να προσδιορίσουμε ακριβώς το πολύγωνο όπου όταν ο χρήστης εισέλθει σε αυτό να του αποσταλεί μήνυμα από το κατάστημα που είναι εκεί κοντά.

Όσον αφορά την εγγραφή του χρήστη για να μπορεί να χειρίζεται την εφαρμογή έχουμε ένα registration form όπου μόλις ο νέος χρήστης βάλει τα στοιχεία του θα του έρθει αυτοματοποιημένο email όπου θα πατήσει σε ένα link για να γίνει verified. Εφόσον πραγματοποιηθεί αυτό μόνο και μόνο τότε μπορεί να κάνει login μέσω του login form.

Η εφαρμογή μας έχει και άλλες λειτουργίες όπως χάρτη google maps όπου ο χρήστης μπορεί να βλέπει σημεία ψυχαγωγίας κοντά σε αυτόν με βάση τη τοποθεσία του και οδηγίες κατεύθυνσης για το πως να πάει στο σημείο που επιθυμεί.

Θα αναφερθούμε στο χειρισμό της εφαρμογής από πλευράς χρήστη αλλά και από πλευράς του διαχειριστή της firebase και τις δυνατότητες που έχει ο καθένας. Τέλος θα εξηγήσουμε κάποια σημεία κώδικα που παρουσιάζουν ενδιαφέρον και που δίνουν στην εφαρμογή μας ξεχωριστό χαρακτήρα.

Summary

The purpose of the app is to let users know when they make purchases for any offers based on their location. This has several advantages for the buyer-user as well as for the seller, which we will discuss below. In addition, the firebase salesperson-user can send grouped messages to all application users via email to inform about topics that might interest the user such as promotions, product advertising, and events.

Our app also keeps statistics on the number of times users have accessed an application in an area where users can view them. So we can see the mobility of a region and draw useful conclusions.

A key element of our programmatic differentiation is that it offers the possibility that the geographic area where you enable the automatic sending of a message-notification to the user is within a polygon and not a circle as usual. This has the advantage of having more precision since we can precisely define the polygon where when the user enters it, a message is sent from the store that is nearby.

Regarding the registration of the user to be able to handle the application we have a registration form where once the new user puts his / her details he will receive an automated email where he will click on a link to be verified. Once this is done, it can login via the login form.

Our app has other features such as google maps where the user can see entertainment spots near him based on his location and directions on how to go to where he wants. We will refer to the user's handling of the application as well as the firebase manager and the capabilities of each.

Finally, we will explain some code points that are of interest and that give our application a distinct character.

Περιεχόμενα:

Περίληψη.....	4
Summary.....	5
Εισαγωγή	7
Ανασκόπηση Πεδίου	8
Αντίστοιχες Εφαρμογές.....	10
Foursquare & Swarm.....	10
Pokemon Go	10
NPS National Mall.....	11
Gas Buddy	12
Trigger.....	13
Sephora.....	13
Spyzie.....	14
Control-Apple Maps	15
Kaspersky kids safe	16
Εγχειρίδιο Χρήστη	18
Login Form – Registration Form	18
Φόρμα Σύνδεσης:.....	18
Φόρμα Εγγραφής:	19
Κεντρική Σελίδα.....	20
Κουμπί Ενεργοποίησης GPS	21
Χάρτης-Shop Map.....	24
Ρυθμίσεις Εφαρμογής	27
Στατιστικά δεδομένα εφαρμογής	28
Χειρισμός από πλευράς διαχειριστή(firebase)	29
Ειδοποίηση με μήνυμα προς όλους τους χρήστες.....	29
Authentication.....	32
Αρχιτεκτονική Συστήματος.....	33
Εργαλεία που χρησιμοποιήθηκαν.....	33
Σχολιασμός κώδικα	33
Polygon geofencing	33
Google maps implementation.....	37
Search button	37

Search nearest places.....	38
Email Verification	41
Συμπεράσματα και μελλοντικές επεκτάσεις.....	44
Βιβλιογραφία – πηγές	46

Εισαγωγή

Η εφαρμογή αρχικά ξεκίνησε σαν βασική λειτουργία να χρησιμοποιεί τη γεωγραφική θέση του χρήστη και να ειδοποιεί για κοντινά σημεία ενδιαφέροντος. Συνήθως τέτοιου είδους εφαρμογές για να ειδοποιηθεί ο χρήστης όταν μπαίνει σε ένα σημείο ενδιαφέροντος χρησιμοποιούν ακτίνα κύκλου από το σημείο το οποίο πολλές φορές μπορεί να μη βολεύει και να οδηγεί σε ανακριβή αποτελέσματα. Έτσι στη συγκεκριμένη εφαρμογή υλοποιήθηκε κώδικας όπου δεν λαμβάνει την ακτίνα κύκλου από ένα σημείο ,δημιουργώντας ένα κύκλο που όταν εισέλθει ο χρήστης τότε ειδοποιείτε, αλλά δημιουργήσαμε πολύγωνα που τότε να ειδοποιείτε. Αυτό έχει σαν θετικό ότι έχουμε μεγαλύτερη ακρίβεια όπως παραδείγματος χάρη θέλουμε ο χρήστης να ειδοποιηθεί όταν μπει σε ένα μεγάλο εμπορικό κέντρο, με την ακτίνα κύκλου θα έπρεπε να βάλουμε σαν κέντρο το εμπορικό και μια μεγάλη ακτίνα που θα έπιανε περιοχές και έξω από το εμπορικό ενώ με το να θέτουμε πολύγωνο μπορούμε να βάλουμε ακριβώς την περιοχή που είναι μέσα στο εμπορικό.

Επιπλέον πρόσθεσα τη δυνατότητα στο χρήστη να μπορεί να βλέπει την τοποθεσία του και πατώντας κάθε φορά το αντίστοιχο κουμπί να βλέπει γύρω του κάποια σημεία ενδιαφέροντος (καφετέριες, καταστήματα ρούχων κλπ.). Και μπορεί να δει και οδηγίες από τον χάρτη για το πως να πάει σε κάποιο σημείο. Τα δεδομένα για τα σημεία ενδιαφέροντος τα παίρνω μέσω του API της google.

Εκτός από τα παραπάνω πρόσθεσα και μια επιπλέον λειτουργία η οποία μπορεί να δίνει πληροφορίες και στον χρήστη αλλά και στον διαχειριστή της εφαρμογής. Αυτή η λειτουργία αφορά κάποια σημεία ενδιαφέροντος τα οποία η εφαρμογή μας κάνει μετρήσεις για το πόσες φορές έχει εισέλθει ένας χρήστης. Αυτό είναι πολύ χρήσιμο καθότι μπορούμε να δούμε τις ώρες αιχμής για κάποιο σημείο(π.χ. λιμάνι, εμπορικό κέντρο).

Θα ήθελα να αναφέρω ακόμα ότι η εφαρμογή μπορεί να χρησιμοποιηθεί πολύ εύκολα και από χρήστες που δεν είναι τόσο εξοικειωμένοι με την τεχνολογία. Το μόνο που έχουν να κάνουν είναι να φτιάξουν ένα λογαριασμό πολύ εύκολα και αμέσως μετά εφόσον βέβαια γίνει ταυτοποίηση του ηλεκτρονικού τους ταχυδρομείου να μπορέσουν να εισέλθουν και να χρησιμοποιήσουν την εφαρμογή από την αρχική οθόνη.

Παρέχεται σε τρεις γλώσσες Ελληνικά, Αγγλικά και Γερμανικά οι οποίες αλλάζουν μέσα από τις ρυθμίσεις που από εκεί αν θέλει ο χρήστης μπορεί να απενεργοποιήσει και τις ειδοποιήσεις της εφαρμογής.

Ανασκόπηση Πεδίου

Εδώ θα ήθελα να κάνω αναφορά σε εφαρμογές που η κύρια λειτουργία τους είναι το Geofencing και που έχουν χιλιάδες downloads στο Google app.

Τι είναι όμως το Geofencing ?

Το Geofencing είναι ένα χαρακτηριστικό ενός προγράμματος λογισμικού που χρησιμοποιεί το παγκόσμιο σύστημα εντοπισμού θέσης (GPS) ή την αναγνώριση ραδιοσυχνοτήτων (RFID) για τον καθορισμό γεωγραφικών ορίων. Το Geofencing επιτρέπει σε ένα διαχειριστή να ρυθμίζει ενεργοποιητές έτσι ώστε όταν μια συσκευή εισέρχεται (ή εξέρχεται) από τα όρια που ορίζει ο διαχειριστής, εκδίδεται μια ειδοποίηση. Πολλές εφαρμογές geofencing ενσωματώνουν το Google Earth, επιτρέποντας στους διαχειριστές να καθορίζουν τα όρια πάνω από μια δορυφορική προβολή μιας συγκεκριμένης γεωγραφικής περιοχής. Άλλες εφαρμογές ορίζουν όρια βάσει γεωγραφικού μήκους και γεωγραφικού πλάτους ή μέσω χαρτών που δημιουργούνται από το χρήστη και βασίζονται στο Web.

Συνήθως είναι οριοθετημένο στον κώδικα μιας εφαρμογής για κινητά, ειδικά επειδή οι χρήστες πρέπει να συμμετέχουν στις υπηρεσίες εντοπισμού θέσης για να λειτουργήσει το geofencing. Εάν μεταβείτε σε ένα συμβάν(συναυλία π.χ), μπορεί να έχουν μια εφαρμογή που μπορείτε να κάνετε λήψη, η οποία θα παρέχει πληροφορίες σχετικά με το συμβάν. Ή, ένας έμπορος λιανικής πώλησης μπορεί να σχεδιάσει ένα geofence γύρω από τα καταστήματά του για να ενεργοποιήσει ειδοποιήσεις για κινητά για πελάτες που έχουν κατεβάσει την εφαρμογή για κινητά του μεταπωλητή. Σε αυτές τις περιπτώσεις προγραμματίζεται στην εφαρμογή η γεωγραφική θέση που διαχειρίζεται ο λιανοπωλητής και οι χρήστες μπορούν να επιλέξουν να απορρίψουν την πρόσβαση στην τοποθεσία για την εφαρμογή.

Μια εφαρμογή geofence μπορεί επίσης να ρυθμιστεί από τους τελικούς χρήστες που χρησιμοποιούν τις δυνατότητες geofencing στις εφαρμογές τους για κινητά. Αυτές οι εφαρμογές, όπως οι Υπενθυμίσεις iOS, σας επιτρέπουν να επιλέξετε μια διεύθυνση ή μια τοποθεσία όπου θέλετε να ενεργοποιήσετε μια συγκεκριμένη ειδοποίηση ή μια ειδοποίηση ώθησης. Αυτό ονομάζεται εντολή "αν αυτή, τότε αυτή", όπου μια εφαρμογή προγραμματίζεται για να ενεργοποιήσει μια ενέργεια βασισμένη σε άλλη ενέργεια. Για παράδειγμα, "Αν είμαι πέντε μέτρα από την μπροστινή πόρτα μου, ενεργοποιήστε τα φώτα μου". Ή μπορείτε να ζητήσετε μια εφαρμογή υπενθύμισης να σας στείλει μια ειδοποίηση μόλις φτάσετε σε μια συγκεκριμένη τοποθεσία.

Το Geofencing δεν είναι μόνο για κινητές εφαρμογές - χρησιμοποιείται για τον έλεγχο και την παρακολούθηση των οχημάτων στη ναυτιλιακή βιομηχανία, των βοοειδών στη γεωργία και - θα δείτε αυτό το θέμα να εμφανίζεται σε συζητήσεις με drone. Σχεδόν κάθε drone είναι προγραμματισμένο να λειτουργεί με geofences, τα οποία συνήθως δημιουργούνται γύρω από αεροδρόμια, υπαίθριους χώρους και ακόμη και τον Λευκό Οίκο. Η FAA μπορεί να προγραμματίσει αυτά τα drone-resistant geofences κατόπιν αιτήματος τα οποία μπορούν να σταματήσουν ένα drone στον αέρα ενώ άλλα θα στείλουν προειδοποιητικό μήνυμα στον χρήστη. Ορισμένες γεωγραφικές περιοχές θα ζητήσουν την εξουσιοδότηση των χρηστών - μια διαδικασία που συνδέει την ταυτότητα του χρήστη με το drone τους - έτσι ώστε η επιβολή του νόμου να μπορεί να παρακολουθεί τα μη επανδρωμένα αεροσκάφη.

Ας δούμε κάποιες από αυτές τις χρήσεις geofencing και επιπλέον μερικές στο

Χρησιμότητα	Παράδειγμα
Διαχείριση Drone	Ένα αθλητικό γεγονός μπορεί να χρησιμοποιήσει geofencing για να δημιουργήσει μια προσωρινή ζώνη χωρίς πτήσεις, η οποία εμποδίζει τα drone να διασχίσουν μια καθορισμένη περίμετρο.
Διαχείριση Στόλου	Το Geofencing μπορεί να ειδοποιήσει έναν αποστολέα όταν ένας οδηγός φορτηγού αλλάξει διαδρομή.
Διαχείριση Ανθρώπινου Δυναμικού	Η έξυπνη κάρτα ενός υπαλλήλου θα στείλει μια ειδοποίηση για την ασφάλεια εάν ο εργαζόμενος επιχειρήσει να εισέλθει σε μια μη εξουσιοδοτημένη γεωγραφική περιοχή.
Μάρκετινγκ	Μια μικρή επιχείρηση μπορεί να δημιουργήσει κείμενο σε έναν πελάτη συμμετοχής σε έναν κωδικό κουπονιού όταν το smartphone του πελάτη εισέλθει σε μια καθορισμένη γεωγραφική περιοχή.
Διαχείριση περιουσιακών στοιχείων	Ο διαχειριστής δικτύου μπορεί να ρυθμίσει τις ειδοποιήσεις έτσι ώστε όταν ένα νοσοκομείο που ανήκει στο iPad εγκαταλείπει το χώρο του νοσοκομείου, ο διαχειριστής μπορεί να παρακολουθήσει τη θέση της συσκευής και να την κλειδώσει για να αποφευχθεί η χρήση της.
Επιβολή νόμου	Ένα βραχιόλι αστραγάλου μπορεί να προειδοποιήσει τις αρχές εάν ένα άτομο που βρίσκεται κάτω από κατ'οίκον περιορισμό εγκαταλείψει τις εγκαταστάσεις.
Αυτοματισμός στο σπίτι	Όταν το smartphone του ιδιοκτήτη του σπιτιού εγκαταλείπει τη γεωμετρική περίμετρο του σπιτιού, ο θερμοστάτης χαμηλώνει σε προκαθορισμένη θερμοκρασία.
Διαχείριση συμμόρφωσης	Τα αρχεία καταγραφής δικτύου μπορούν να καταγράψουν διασταυρώσεις geo-fence για να τεκμηριωθούν η σωστή χρήση των συσκευών και η συμμόρφωσή τους με τις καθιερωμένες πολιτικές.

παρακάτω πίνακα.

Αντίστοιχες Εφαρμογές

Ας δούμε τώρα κάποιες γνωστές εφαρμογές που η κύρια δυνατότητα τους είναι το Geofencing.

Foursquare & Swarm

Το Foursquare ήταν πρωτοπόρος στις εφαρμογές που έκαναν χρήση της τοποθεσίας. Έχει διαχωρίσει την εφαρμογή της σε δύο μέρη: το Foursquare συνεχίστηκε ως έξυπνο εργαλείο αναζήτησης που επικεντρώνεται στην ανακάλυψη κοντινών τοποθεσιών, εκδηλώσεων, εστιατορίων και καταστημάτων, ενώ το Swarm δημιουργήθηκε για αυτούς που κάνουν κοινή χρήση της τοποθεσίας τους μαζί με φίλους. Οι χρήστες εισάγουν τα αγαπημένα τους φαγητά και δραστηριότητες το Foursquare τους βοηθά να βρίσκουν κοντινά μέρη που ταιριάζουν στο λογαριασμό.

Screenshot:



Pokemon Go

Είναι από τα πιο δημοφιλή παιχνίδια στο google play με εκατομμύρια downloads. Οι παίκτες εξερευνούν τη γειτονιά τους με τα πόδια, χρησιμοποιώντας το smartphone τους ως χάρτη για να ανακαλύψουν Pokemon, καθώς και για να συλλέξουν αντικείμενα. Το παιχνίδι χρησιμοποιεί το geofencing για να στέλνει συνεχώς ειδοποιήσεις στον χρήστη που

έχει ανοικτό το gps όταν αυτός εισέρχεται σε περιοχή που έχει κάποιο αντικείμενο για να συλλέξει. Μετά την επιτυχία του συγκεκριμένου παιχνιδιού βγήκαν και άλλα παρόμοια.

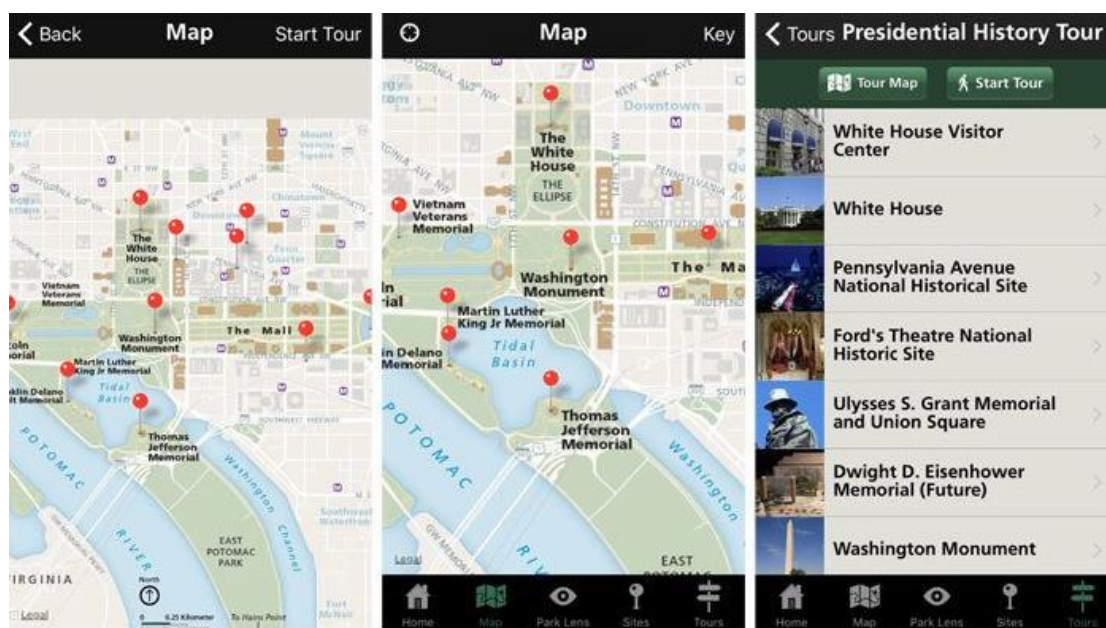
Screenshot:



NPS National Mall

Η τεχνολογία τοποθεσίας αποτελεί επίσης μια μεγάλη βοήθεια για τους τουρίστες. Ένα όμορφο παράδειγμα είναι η εφαρμογή National Mall η οποία λειτουργεί ως μια εξαιρετική εφαρμογή για την εξερεύνηση ιστορικών κτηρίων και μνημείων. Η εφαρμογή έρχεται με έναν λεπτομερή χάρτη που παρακολουθεί σημεία ενδιαφέροντος γύρω από τον χρήστη με το πλήκτρο “Locate Me”.

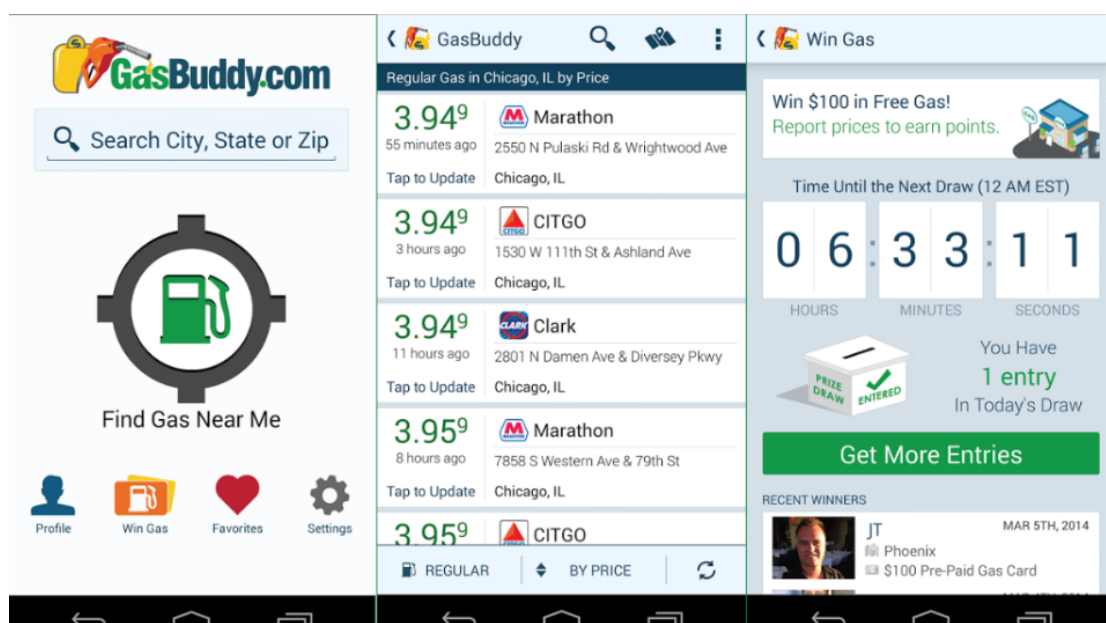
Screenshot:



Gas Buddy

Είναι μια εφαρμογή κινητού που βοηθάει τον χρήστη να βρίσκει το πιο φθινό πρατήριο βενζίνης με βάση την τοποθεσία του. Πολλές από τις τιμές βασίζονται σε αναφορές των χρηστών οπότε η εφαρμογή δίνει ανταμοιβές(πόντους) σε όσους το κάνουν. Όταν μαζέψουν αρκετούς πόντους μπορούν να πάρουν δωρεάν βενζίνη.

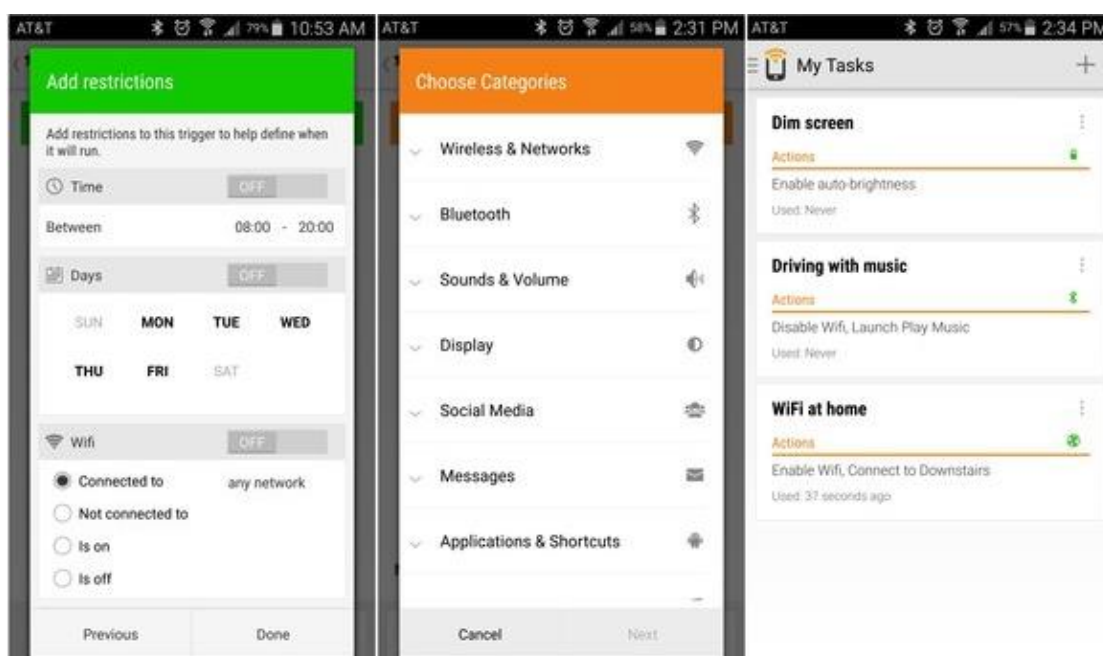
Screenshot:



Trigger

Το trigger είναι μια φανταστικά ευέλικτη εφαρμογή που μπορεί να ενεργοποιήσει αυτόματα μια ποικιλία ενεργειών τηλεφώνου με βάση ετικέτες NFC, συνδέσεις δικτύου Wi-Fi, συνδέσεις Bluetooth και πολλά άλλα. Με μια αναβάθμιση εντός εφαρμογής, οι χρήστες μπορούν να ορίσουν geofences ως task triggers, επιτρέποντάς τους να ενεργοποιούν ή να απενεργοποιούν αυτόματα το Wi-Fi όταν φεύγουν από το χώρο εργασίας τους ή να στέλνουν ένα μήνυμα SMS "Είμαι σπίτι" στους αγαπημένους τους όταν γυρνάνε σπίτι. Οι δυνατότητες περιορίζονται μόνο από την ευστροφία σας.

Screenshot:

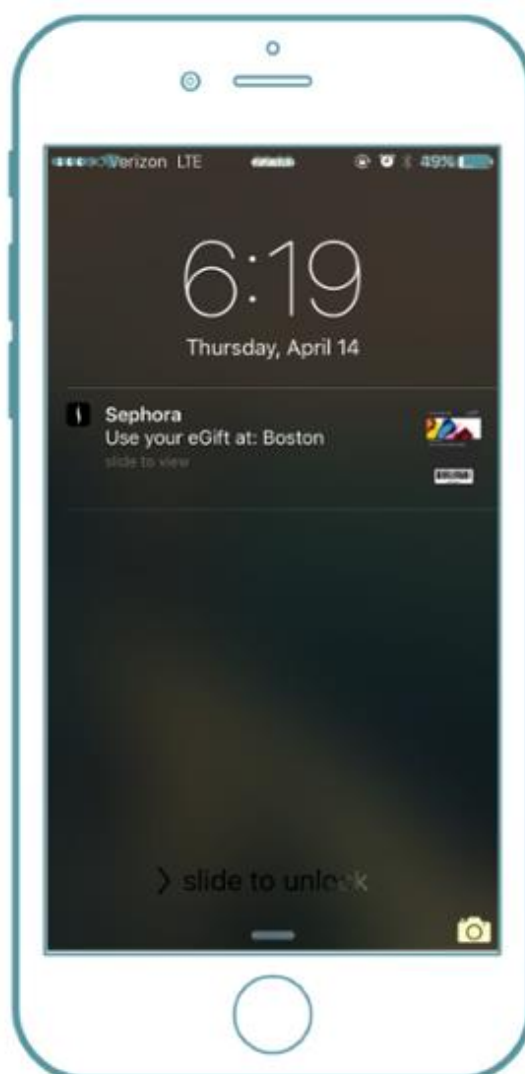


Sephora

Η Sephora είναι παγκόσμιος ηγέτης στο λιανικό εμπόριο παγκόσμιου κύρους. Είναι ένα από τα καλύτερα κέντρα ομορφιάς στο Διαδίκτυο. Στον κόσμο υπάρχουν περίπου 2300 καταστήματα σε 33 διαφορετικές χώρες. Το Sephora σχεδιάστηκε επίσης για συσκευές που λειτουργούν με λειτουργικό σύστημα Android και iOS. Η εφαρμογή του Sephora είναι εξοπλισμένη με διάφορα είδη χαρακτηριστικών που καθιστούν την εμπειρία σας στην αγορά πιο εύκολη και βολική. Ένα από τα καλύτερα χαρακτηριστικά του Sephora είναι ότι υποστηρίζει το Geofencing που σημαίνει ότι όταν περνάτε από οποιαδήποτε από τις εφαρμογές του

καταστήματος θα σας υπενθυμίσω με ειδοποίηση push. Αυτό είναι το τέλειο παράδειγμα της Geofenced push notification.

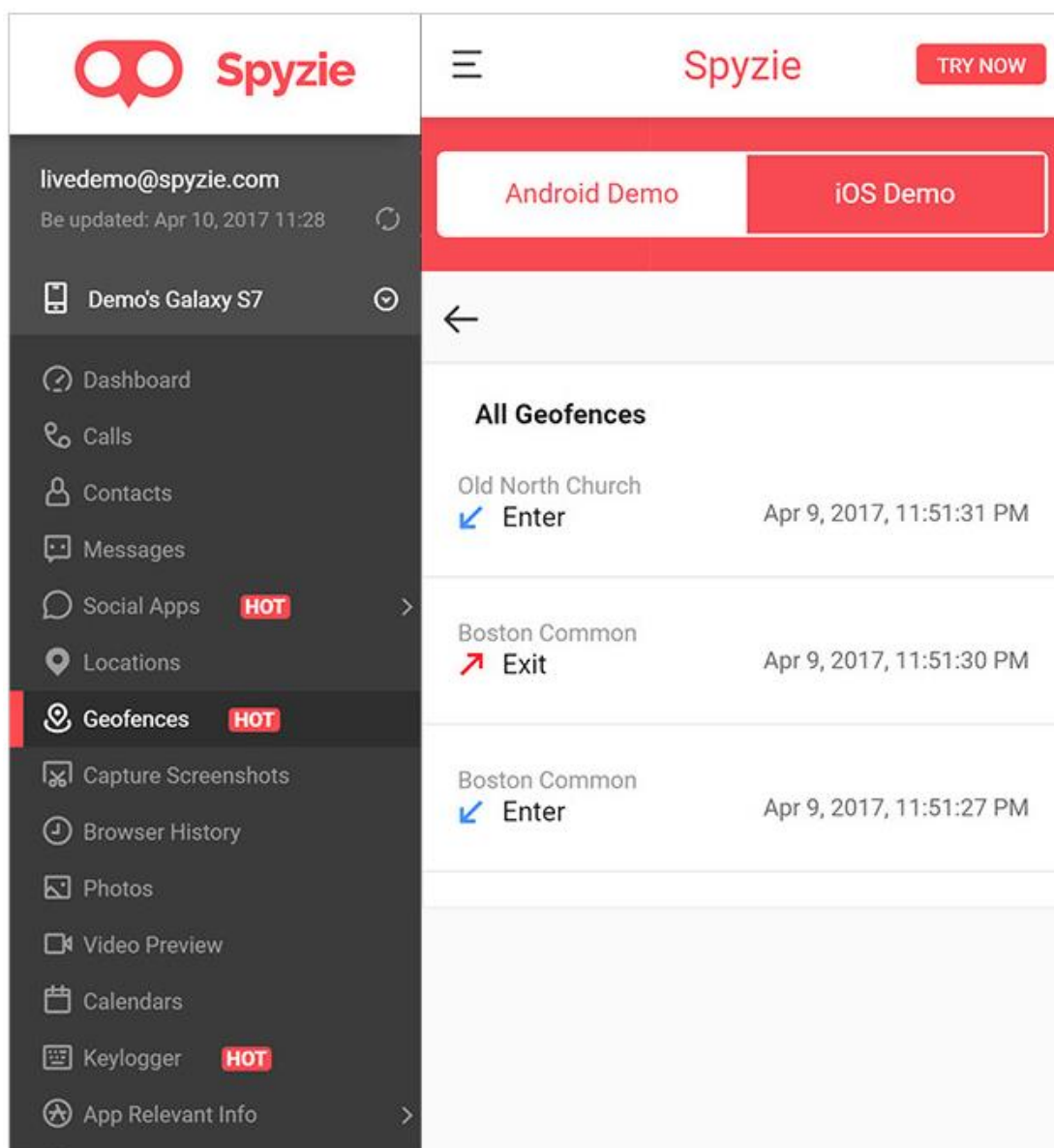
Screenshot:



Spyzie

Το Spyzie είναι μια διαδικτυακή λύση παρακολούθησης κινητού τηλεφώνου που είναι εξοπλισμένη με διάφορα είδη λειτουργιών που σας βοηθούν στην καθημερινή σας ζωή. Ο Spyzie επιτρέπει στους γονείς να προστατεύουν τα παιδιά τους από μια επιβλαβή απειλή στον ιστό, όπως το περιεχόμενο για ενήλικες, τον ηλεκτρονικό εκφοβισμό, τα αρπακτικά ζώα και πολλά άλλα. Είναι βασικά σχεδιασμένο για συσκευές που λειτουργούν σε λειτουργικό σύστημα Android και iOS. Με το Spyzie μπορείτε να έχετε πρόσβαση σε όλες τις επαφές, τα μηνύματα, τα αρχεία καταγραφής κλήσεων (εισερχόμενες, εξερχόμενες και αναπάντητες κλήσεις), την τοποθεσία GPS, το Keylogger, το Geofencing και πολλά άλλα στοιχεία της συσκευής προορισμού. Απλά πρέπει να εγκαταστήσετε το Spyzie plugin στο τηλέφωνο προορισμού Android. Στην περίπτωση της συσκευής iOS, δεν χρειάζεται να εγκαταστήσετε κανένα εργαλείο κατασκοπείας.

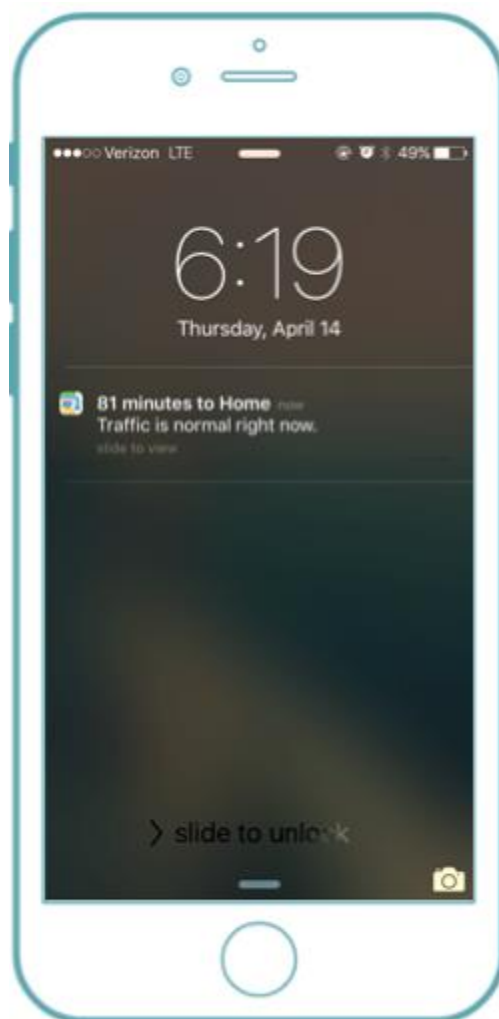
Screenshot:



Control-Apple Maps

Η Apple Maps είναι μια υπηρεσία Map που αναπτύχθηκε από την Apple Inc. Από προεπιλογή, είναι εγκατεστημένη σε κάθε συσκευή που τροφοδοτείται από το λειτουργικό σύστημα iOS. Όπως οι Χάρτες Google, διαθέτει επίσης Geofencing και άλλα ισχυρά χαρακτηριστικά που σας επιτρέπουν να βλέπετε τους χάρτες της Apple πριν φύγετε. Μερικές φορές έχουν υπάρξει στιγμές όταν βγαίνετε από το σπίτι σας, οι χάρτες Apple θα ανιχνεύσουν αυτόματα την τοποθεσία σας και θα σας δώσουν ειδοποίηση push με πληροφορίες σχετικά με την οδήγηση στο σπίτι. Οι χάρτες της Apple σχεδιάζονται κυρίως για να κάνουν την οδηγική εμπειρία πιο εύκολη και βολική.

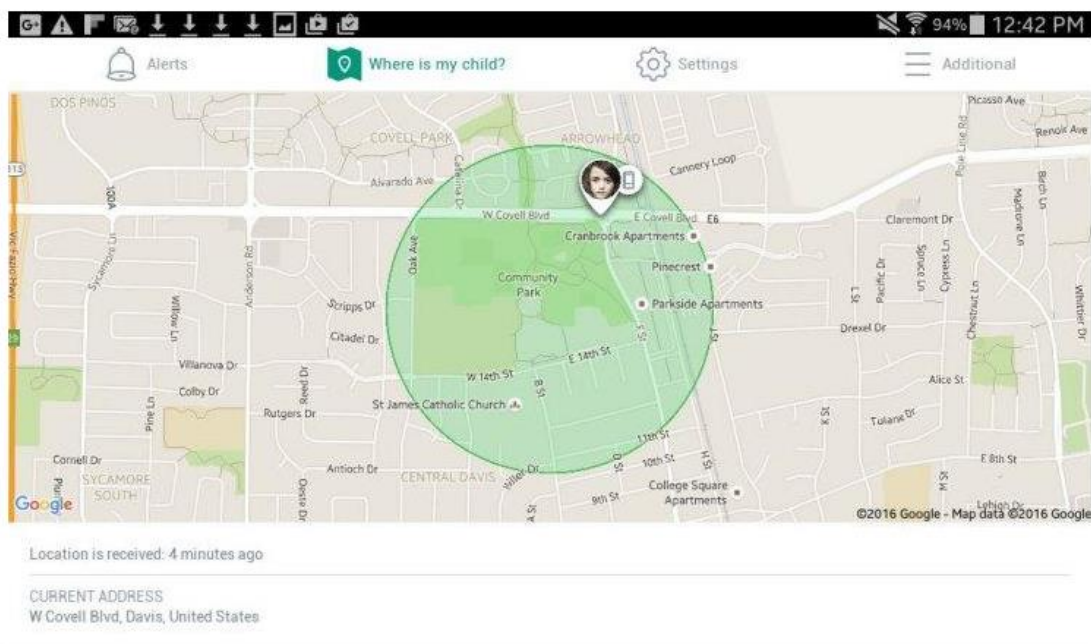
Screenshot:



Kaspersky kids safe

Το Kaspersky Safe Kids είναι ένα λογισμικό που σχεδιάστηκε από την Kaspersky Labs. Ο κύριος λόγος για την ανάπτυξη αυτής της εφαρμογής είναι η προστασία των παιδιών από επιβλαβείς απειλές στον ιστό. Στο πακέτο ασφαλών παιδιών Kaspersky, υπάρχουν πολλά χαρακτηριστικά που είναι διαθέσιμα, όπως το Geofencing για την παρακολούθηση της θέσης του παιδιού σας.

Screenshot:



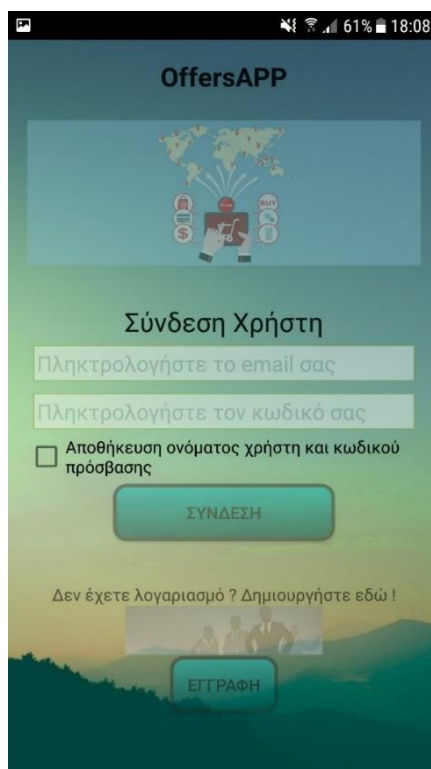
Εγχειρίδιο Χρήστη

Login Form – Registration Form

Αφού γίνει εγκατάσταση της εφαρμογής μας μόλις ο χρήστης ανοίξει για πρώτη φορά την εφαρμογή θα εμφανιστεί η φόρμα Login όπου ο χρήστης αν είναι ήδη εγγεγραμμένος θα πληκτρολογήσει το email του και τον κωδικό του και πατώντας το κουμπί «ΣΥΝΔΕΣΗ» θα εισέλθει στην εφαρμογή μας.

Φόρμα Σύνδεσης:

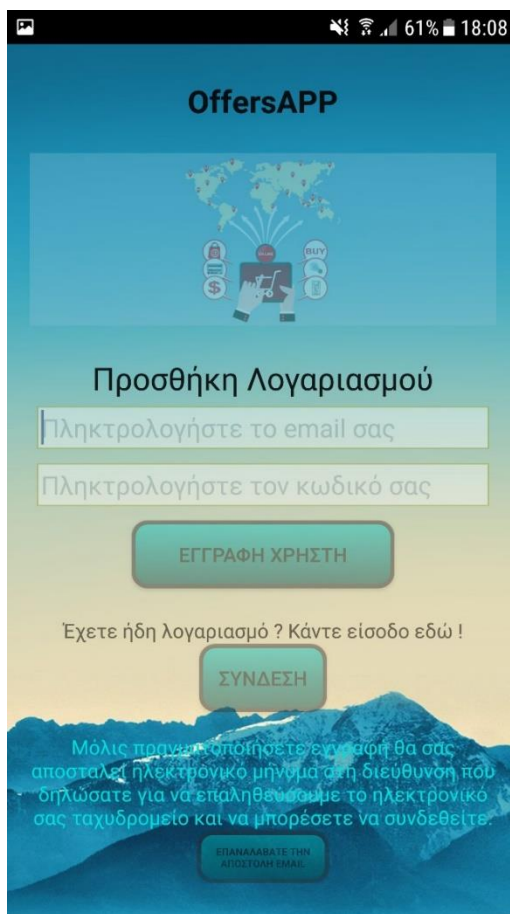
Screenshot:



Έχει την δυνατότητα επίσης να πατήσει στο checkbox που ενεργοποιώντας το θα αποθηκεύσει τα στοιχεία του και δεν θα χρειαστεί να τα πληκτρολογήσει ξανά την επόμενη φορά που θα ανοίξει την εφαρμογή. Εφόσον όμως ο χρήστης μας ανοίγει για πρώτη φορά την εφαρμογή θα χρειαστεί να κάνει εγγραφή οπότε θα πατήσει στο κουμπί «ΕΓΓΡΑΦΗ» και θα ανοίξει η Φόρμα Εγγραφής.

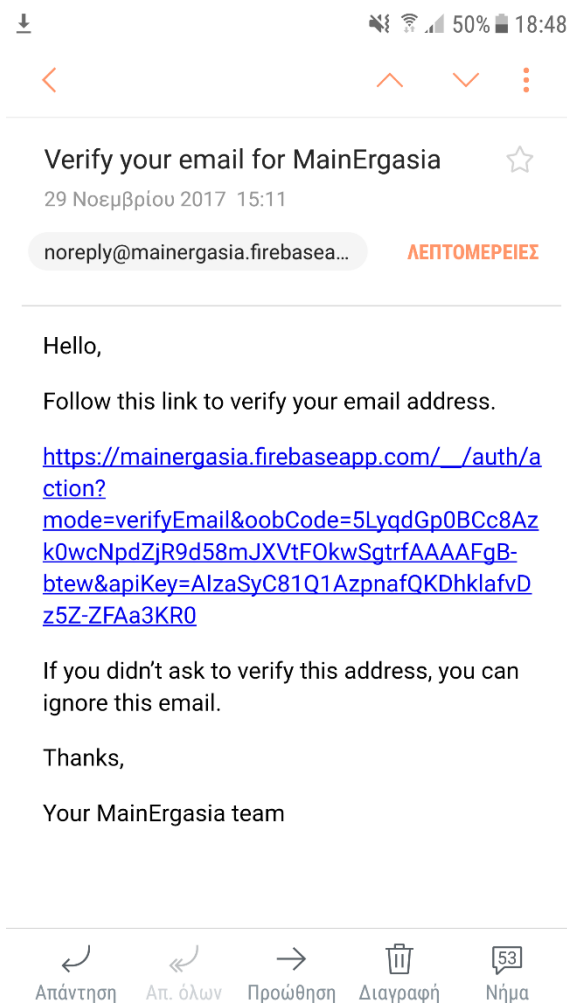
Φόρμα Εγγραφής:

Screenshot:



Στη φόρμα εγγραφής θα χρειαστεί ο νέος μας χρήστης να πληκτρολογήσει το επιθυμητό email και κωδικό που θα χρησιμοποιεί για να μπορεί να συνδεθεί, εφόσον τα πληκτρολογήσει στα edittext και πατήσει στο κουμπί «ΕΓΓΡΑΦΗ ΧΡΗΣΤΗ» θα του σταλεί ηλεκτρονικό μήνυμα στη διεύθυνση που δήλωσε. Στο μήνυμα θα υπάρχουν οδηγίες που θα παραπέμπουν τον χρήστη να πατήσει σε συγκεκριμένο σύνδεσμο, με αυτό το τρόπο κάνουμε ταυτοποίηση του ηλεκτρονικού του ταχυδρομείου.

Screenshot:



Επιπλέον, στην περίπτωση που δεν λάβει κάποιο ηλεκτρονικό μήνυμα ο χρήστης μπορεί να πατήσει στο κουμπί «ΕΠΑΝΑΛΑΒΑΤΕ ΤΗΝ ΑΠΟΣΤΟΛΗ EMAIL» στο κάτω μέρος της φόρμας. Εάν προσπαθήσουμε να συνδεθούμε στην εφαρμογή χωρίς να έχουμε κάνει ταυτοποίηση δεν θα μας αφήσει και θα μας εμφανίσει μήνυμα να ταυτοποιήσουμε το ηλεκτρονικό μας ταχυδρομείο.

Κεντρική Σελίδα

Έχοντας πραγματοποιήσει τις απαραίτητες ενέργειες για την εγγραφή στο Login Form και αφού βάλουμε τα στοιχεία μας πατάμε το κουμπί «ΣΥΝΔΕΣΗ» και εμφανίζεται η κεντρική σελίδα όπου εκεί έχουμε διάφορες επιλογές που μπορούμε να κάνουμε και τις οποίες θα αναλύσουμε παρακάτω.

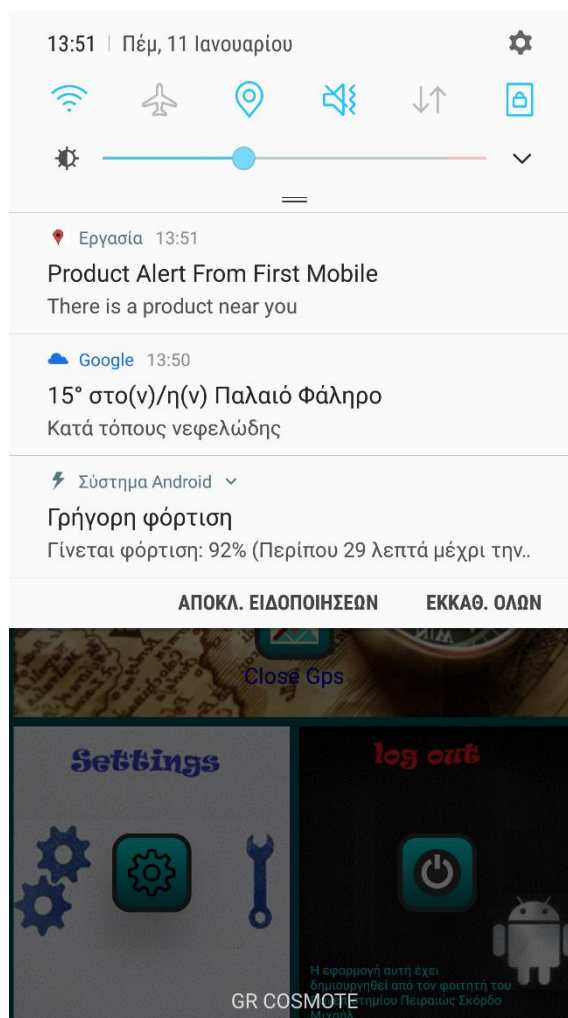
Screenshot-Κεντρική σελίδα:

**Κουμπί Ενεργοποίησης GPS**

Στο κέντρο της σελίδας μας υπάρχει ένα κουμπί όπου με το πάτημα ενεργοποιείτε το GPS, όταν το ενεργοποιήσουμε και εφόσον βρισκόμαστε σε γεωγραφική θέση όπου υπάρχει κατάστημα-σημείο ενδιαφέροντος τότε θα μας αποσταλεί ειδοποίηση από το κατάστημα σε μορφή notification όπως επίσης και μήνυμα στη κεντρική σελίδα πάνω αριστερά. Οπότε ο χρήστης ενημερώνεται και σε μορφή notification αλλά και σε μήνυμα μέσα στην εφαρμογή.

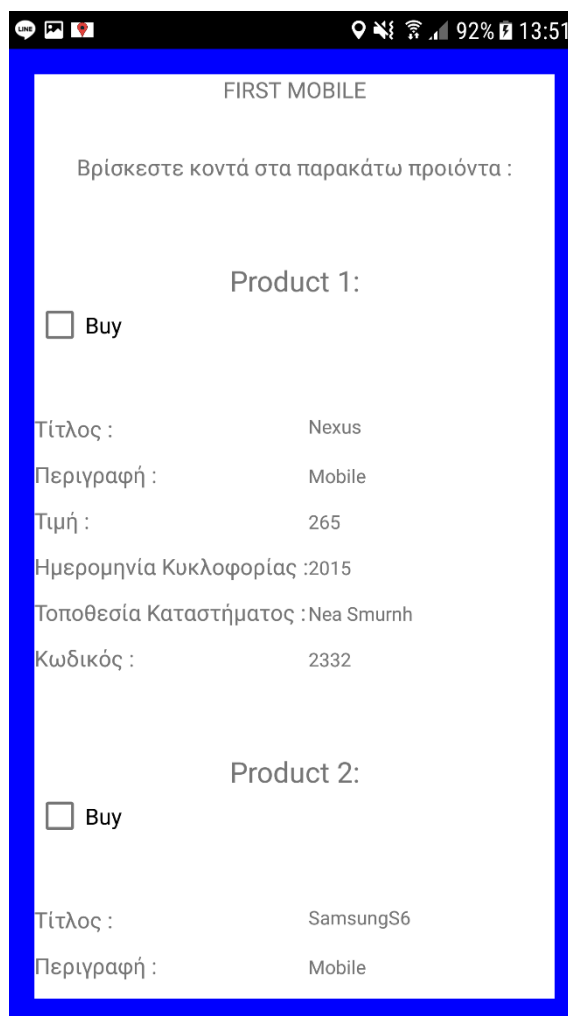
Screenshot Ανοικτού Gps – Ειδοποίηση από Κατάστημα:



Screenshot Notification:

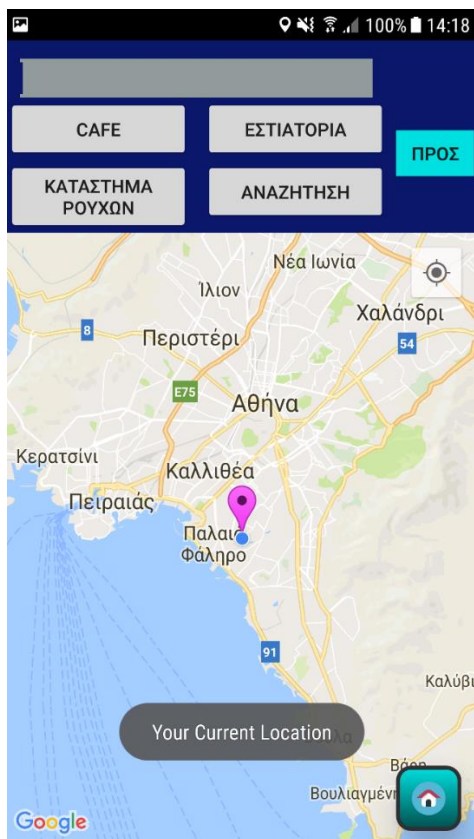
Στη περίπτωση που ο χρήστης δεν περνάει από κάποια γεωγραφική θέση - πολύγωνο που έχει οριστεί τότε δεν θα του έρθει ειδοποίηση αλλά μπορεί να αφήσει ανοιχτό το gps της εφαρμογής μας για να ειδοποιηθεί εφόσον περάσει μέσα από κάποια γεωγραφική θέση - πολύγωνο. Επίσης πατώντας ξανά το ίδιο κουμπί μπορούμε να κλείσουμε το gps , παρατηρούμε ότι αλλάζει και το εικονίδιο του κουμπιού κάθε φορά που το πατάμε για να μας διευκολύνει στο να γνωρίζουμε πότε είναι ανοιχτό και πότε κλειστό.

Στο πάτημα του notification έχουμε δημιουργήσει ένα εικονικό κατάστημα που παίρνει τα προϊόντα από μια βάση στη firebase μας όπου ο χρήστης μπορεί να κάνει αγορές και αυτό θα μπορούμε να το δούμε στη firebase.

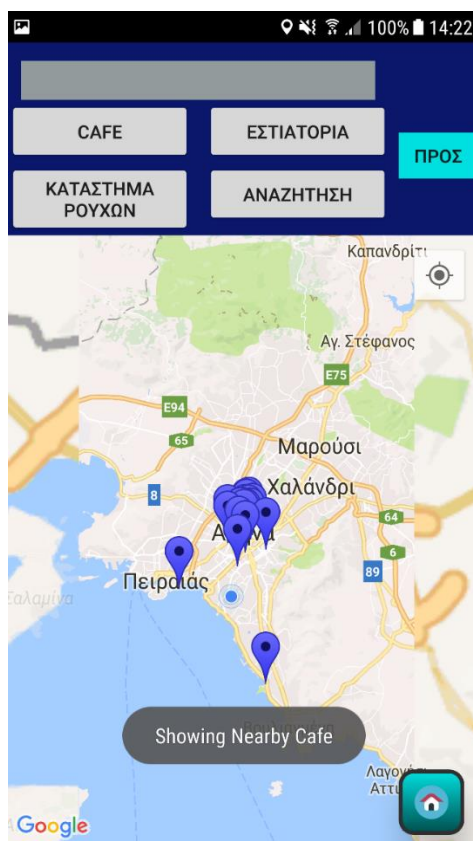
Screenshot – Άνοιγμα του Notification :**Χάρτης-Shop Map**

Μια επιπλέον λειτουργία που έχει η εφαρμογή μας είναι η σύνδεση με το google maps αρι και συγκεκριμένα πατώντας στο κουμπί κάτω από το Shop Map μας εμφανίζεται μια σελίδα με χάρτη που μας εμφανίζει τη θέση μας. Εκεί μπορούμε να κάνουμε αναζήτηση κάποια διεύθυνση ενός καταστήματος αν θέλουμε ή να επιλέξουμε να μας εμφανίσει τα πλησιέστερα σημεία ενδιαφέροντος πατώντας στο ανάλογο κουμπί.

Screenshot 1:

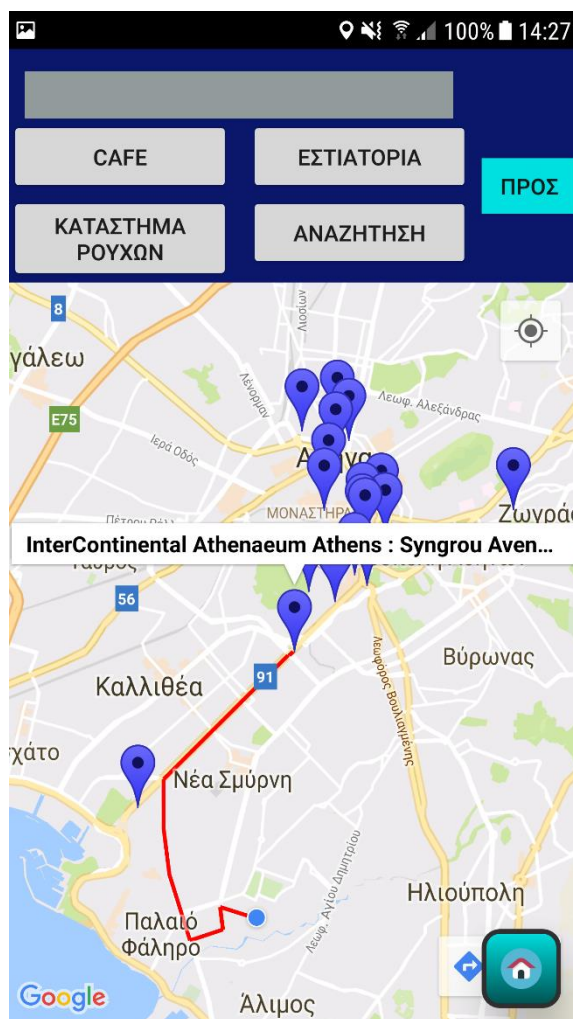


Screenshot 2:



Επιπλέον έχουμε τη δυνατότητα να μας εμφανίσει τη διαδρομή από το σημείο που βρισκόμαστε στο σημείο που θέλουμε να πάμε. Για να το κάνουμε αυτό πατάμε πάνω στη πινέζα-σημείο ενδιαφέροντος και εν συνεχεία το κουμπί «ΠΡΟΣ» και μας εμφανίζει με κόκκινο τη διαδρομή.

Screenshot:



Ρυθμίσεις Εφαρμογής

Ο χρήστης έχει την δυνατότητα στις ρυθμίσεις εφαρμογής να βάλει την επιθυμητή γλώσσα και αν θέλει να αποκλείσει τις ειδοποιήσεις – notifications της εφαρμογής. Με τον αποκλεισμό των ειδοποιήσεων θα εμφανίζεται μόνο το μήνυμα του καταστήματος μέσα από την εφαρμογή πάνω αριστερά στη Κεντρική Σελίδα που αναφέρεται σαν «Shop Message».

Screenshot Ρυθμίσεις Εφαρμογής:



Ο χρήστης εφόσον επιλέξει τις ρυθμίσεις που χρειάζεται αυτές αποθηκεύονται σαν shared preferences οπότε δεν αλλάζουν με το κλείσιμο της εφαρμογής. Πατώντας στο κουμπί κάτω αριστερά μεταβαίνουμε πάλι στη Κεντρική Σελίδα.

Στατιστικά δεδομένα εφαρμογής

Εδώ μπορεί ο χρήστης της εφαρμογής να παρακολουθήσει κάποια δεδομένα για την κίνηση κάποιων συγκεκριμένων σημείων ενδιαφέροντος που έχουν φτιαχτεί από τον administrator της εφαρμογής. Αυτά τα δεδομένα καταγράφουν την “κίνηση” των σημείων αυτών και μπορούν να μας βοηθήσουν να βγάλουμε διάφορα συμπεράσματα όπως το κατά πόσο χρησιμοποιείτε η εφαρμογή μας από τους χρήστες και για το εάν αυτό μεταφράζεται και σε αγορές. Μελλοντικά θα μπορούσαμε να προσθέσουμε και ίσως κάποιο διάγραμμα που να δείχνει τις ώρες για να βλέπουμε πότε έχει περισσότερη κίνηση.

Screenshot Στατιστικά δεδομένα:



Χειρισμός από πλευράς διαχειριστή(firebase)

Ειδοποίηση με μήνυμα προς όλους τους χρήστες

Ο διαχειριστής της Firebase έχει τη δυνατότητα να στείλει εάν το επιθυμεί μήνυμα-ειδοποίηση σε μορφή notification σε όλους τους χρήστες της εφαρμογής. Το notification θα σταλεί στους χρήστες απλά και μόνο αν έχουν εγκατεστημένη την εφαρμογή χωρίς να είναι αναγκαίο να τρέχει η εφαρμογή. Με αυτό τον τρόπο τα καταστήματα μπορούν να ειδοποιήσουν απευθείας, μέσω του διαχειριστή της εφαρμογής, τους χρήστες για προσφορές, παραλαβή

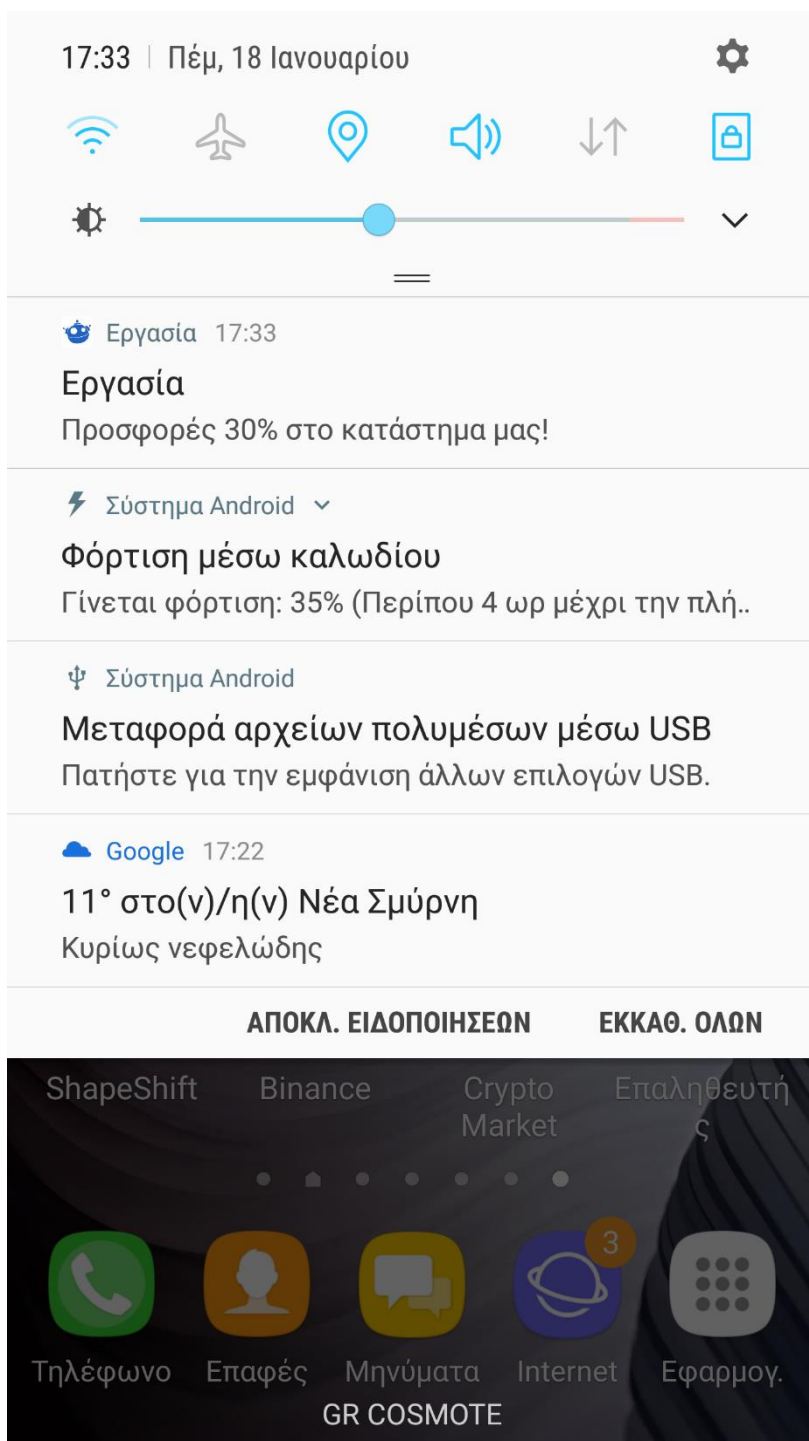
κάποιου νέου προϊόντος, άνοιγμα νέου καταστήματος που συνεργάζεται με την εφαρμογή και ότι άλλο θα θέλαμε να τους πληροφορήσουμε.

Για να μπορέσουμε να στείλουμε το μήνυμα θα πρέπει να μπούμε στο project της Firebase για την εφαρμογή μας και να πατήσουμε στην επιλογή GROW. Θα μας ανοίξει μια λίστα με άλλες επιλογές και θα πατήσουμε στην επιλογή Notifications. Έπειτα πατάμε NEW MESSAGE και πληκτρολογούμε το μήνυμα που θέλουμε να στείλουμε.

Screenshot Από Firebase:

The screenshot displays the 'Compose message' interface in the Firebase console. The top bar is blue with the text 'Notifications' and a back arrow. Below the bar, the main content area is white. It contains several sections: 'Message text' with a text input field containing 'Προσφορές 30% στο κατάστημα μας!'; 'Message label (optional)' with a text input field containing 'Προσφορές'; 'Delivery date' with a dropdown menu set to 'Send Now'; 'Target' section with radio buttons for 'User segment' (selected), 'Topic', and 'Single device'; 'Target user if...' section with a dropdown menu set to 'App' and a package name 'com.example.user.mainergasia' with an 'AND' button; a note 'Cannot add additional statements. All apps have been selected.'; 'Conversion events' section with a dropdown arrow; and 'Advanced options' section with a dropdown arrow. At the bottom right, there are two buttons: 'SAVE AS DRAFT' and 'SEND MESSAGE'.

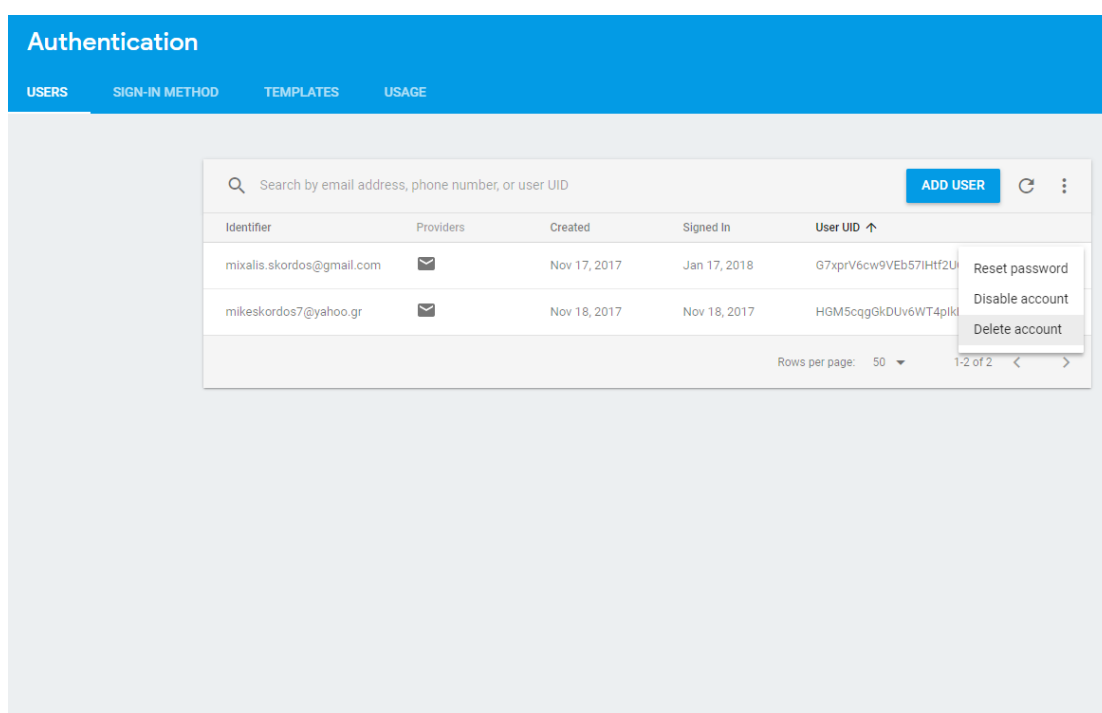
Screenshot Από το Κινητό:



Authentication

Χρησιμοποιούμε το authentication της Firebase για να πιστοποιούμε τους χρήστες και για να βλέπουμε πόσοι χρήστες έχουν κάνει εγγραφή στην εφαρμογή μας. Επίσης από εκεί έχουμε τρεις επιλογές για το λογαριασμό του χρήστη. Αυτές είναι Reset Password, Disable Account και Delete Account. Έτσι με αυτό τον τρόπο ο διαχειριστής έχει τον έλεγχο της εφαρμογής αλλά χωρίς να μπορεί να δει τον κωδικό των χρηστών αλλά μόνο το username τους.

Screenshot Από Firebase:



The screenshot shows the Firebase Authentication console interface. At the top, there is a blue header with the word "Authentication" and four tabs: "USERS", "SIGN-IN METHOD", "TEMPLATES", and "USAGE". Below the header is a search bar with the text "Search by email address, phone number, or user UID" and an "ADD USER" button. A table lists two users with columns for Identifier, Providers, Created, Signed In, and User UID. A context menu is open over the second user, showing options for "Reset password", "Disable account", and "Delete account".

Identifier	Providers	Created	Signed In	User UID ↑
mixalis.skordos@gmail.com	✉	Nov 17, 2017	Jan 17, 2018	G7xprV6cw9VEb57IHf2U
mikeskordos7@yahoo.gr	✉	Nov 18, 2017	Nov 18, 2017	HGM5cqqGkDUv6WT4plk

Αρχιτεκτονική Συστήματος

Εργαλεία που χρησιμοποιήθηκαν

Η εφαρμογή φτιάχτηκε στο Android Studio με χρήση της Java. Σαν βάση χρησιμοποιήσαμε την Firebase της Google. Επίσης σαν emulator χρησιμοποιήσαμε ένα Samsung Galaxy S6 ενώ έγινε test και σε άλλα smartphones για να διαπιστωθεί ότι δουλεύει κανονικά.

Επίσης χρειάστηκε να φτιάξουμε λογαριασμό στη Google για να μπορέσουμε να χρησιμοποιήσουμε κάποια API της google όπως :

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

Παρακάτω θα σχολιάσω κάποια κομμάτια κώδικα που παίζουν σημαντικό ρόλο στην εφαρμογή.

Σχολιασμός κώδικα

Polygon geofencing

Πριν προχωρήσω στο πρώτο κομμάτι κώδικα που αξίζει σχολιασμού θα ήθελα να αναφερθώ στον όρο Geofencing.

Προέρχεται από τις λέξεις geo (γεωγραφικό) και fencing (οριοθετώντας μια περίμετρο). Για εμάς προγραμματιστικά σημαίνει η διαδικασία για να ελέγξουμε αν ένα αντικείμενο εισήλθε ή εξήλθε από μια γεωγραφική περιοχή. Αυτή η περιοχή μπορεί να είναι προκαθορισμένη (οι υπολογισμοί γίνονται σε πραγματικό χρόνο στη συσκευή ή σε ένα server) ή να επεξεργάζεται μετά (τα δεδομένα είναι αποθηκευμένα και επεξεργάζονται σε δεύτερο χρόνο σε ένα server).

Κάποιες εφαρμογές που χρησιμοποιούν geofencing είναι :

- Σύστημα παρακολούθησης κρατουμένων
- Ειδοποίηση γονέα εάν το παιδί τους εγκαταλείπει μια καθορισμένη περιοχή
- Αποστολή ειδοποίησης όταν ένα όχημα εγκαταλείπει προκαθορισμένα σύνορα (κλοπή)
- Παρακολούθηση ζώων στην άγρια φύση εάν εγκαταλείψουν το περιβάλλον τους
- Για οδηγούς ταξί να ειδοποιούνται για τυχόν πελάτες στην περιοχή που βρίσκονται
- Λογισμικά πλοήγησης για να γνωρίζουν αν ο οδηγός έφτασε στο προορισμό του

- Παρακολούθηση περιουσιακού στοιχείου εάν εγκαταλείψει την περιοχή που θα έπρεπε να βρίσκεται (κλοπή).

Παρακάτω θα σχολιάσουμε τον κώδικα της εφαρμογής μας για τη λειτουργία του geofencing. Η διαδικασία θα ήταν πολύ πιο απλή αν χειριζόμασταν την γεωγραφική περιοχή με βάση ένα σημείο και την απόσταση από αυτό το σημείο, δηλαδή δημιουργώντας ένα κύκλο και να ελέγχουμε αν εισέρχεται ή εξέρχεται το αντικείμενο από το κύκλο. Στην περίπτωση μας όμως η γεωγραφική περιοχή θα είναι πολύγωνο το οποίο μπορούμε να χειριστούμε καλύτερα και με μεγαλύτερη ακρίβεια. Επιπλέον λόγω του ότι η σύνθεση των πόλεων είναι σε τετράγωνα και πολύγωνα βοηθάει περισσότερο στην σωστή και όχι περιττή ειδοποίηση.

Θα χρειαστεί να χρησιμοποιήσουμε την λογική ενός αλγορίθμου που ονομάζεται Ray Casting Algorithm για να εξετάσουμε αν το τρέχον γεωγραφικό πλάτος/μήκος είναι μέσα στο πολύγωνο.

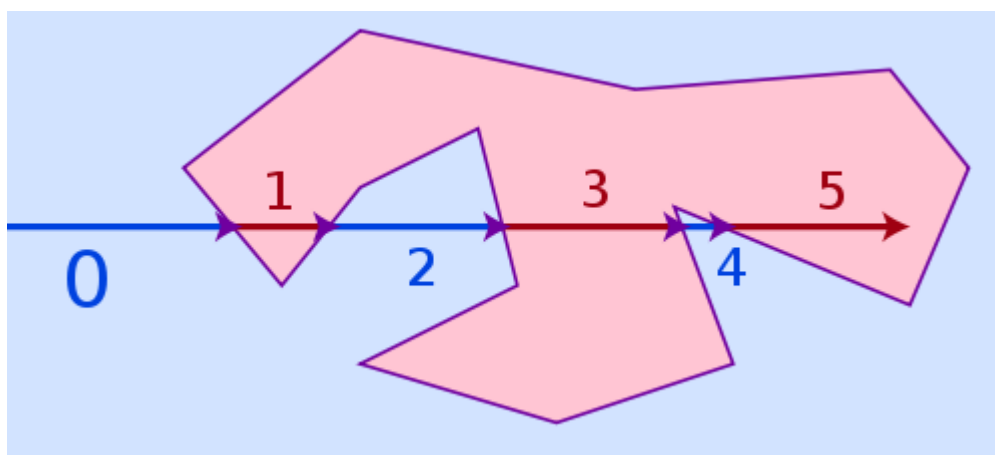
Βήματα Αλγορίθμου:

- Παίρνουμε τις γραμμές του πολυγώνου και διατηρούμε μόνο τις γραμμές στις οποίες το γεωγραφικό πλάτος (Y) τέμνει το τρέχον γεωγραφικό πλάτος

$$Y1 < Y < Y2$$
- Για κάθε μια από τις γραμμές, υπολογίζουμε την ακριβή θέση γεωγραφικού μήκους (X), στην οποία κάθε γραμμή τέμνει το τρέχον γεωγραφικό πλάτος (Y). Για να γίνει αυτό πρώτα υπολογίζουμε την κλίση. Για κάθε γραμμή:

$$m = (Y2 - Y1) / (X2 - X1)$$
 Στη συνέχεια, υπολογίζουμε το γεωγραφικό μήκος (X) για κάθε γραμμή στο τρέχον γεωγραφικό πλάτος (Y).

$$X = X1 + (Y - Y1) / m$$
- Ταξινομούμε τα μήκη
- Περνάμε από όλα τα μήκη. Το πρώτο μήκος που περνάμε είμαστε μέσα στο πολύγωνο. Το δεύτερο είμαστε έξω. Στο τρίτο είμαστε μέσα ξανά, ...



- Συγκρίνουμε το τρέχον γεωγραφικό μήκος με τα παραπάνω μήκη για να προσδιορίσουμε εάν βρισκόμαστε μέσα ή έξω από το πολύγωνο.

Τώρα θα παρουσιάσουμε την λογική του παραπάνω αλγορίθμου μέσα από τον κώδικα java που δημιουργήσαμε για την εφαρμογή μας. Πρώτα φτιάξαμε κάποια αντικείμενα-objects για να εκπροσωπήσουμε τα σημεία, τις γραμμές και τα πολύγωνα.

```

01 class Point {
02     private double x;
03     private double y;
04
05     public Point(double x, double y) {
06         this.x = x;
07         this.y = y;
08     }
09
10     public double getX() {
11         return x;
12     }
13
14     public double getY() {
15         return y;
16     }
17 }
18
19 }
20
21 class Line {
22     private Point from;
23     private Point to;
24
25     public Line(Point from, Point to) {
26         this.from = from;
27         this.to = to;
28     }
29
30     public Point getFrom() {
31         return from;
32     }
33
34     public Point getTo() {
35         return to;
36     }
37 }
38
39 }
40
41 class Polygon {
42     private Point[] points;
43
44     public Polygon(Point[] points) {
45         this.points = points;
46     }
47
48     public Point[] getPoints() {
49         return points;
50     }
51 }
52
53 }

```

Ένα σημείο-point είναι μια θέση με συντεταγμένες x και y. Για να το εφαρμόσουμε στο geofencing, μπορούμε απλώς να σκεφτούμε τις συντεταγμένες x ως γεωγραφικά μήκη και τις συντεταγμένες y ως γεωγραφικά πλάτη.

Μια γραμμή-line είναι μια ευθεία γραμμή με κατεύθυνση. Έχει ένα σημείο από (from-point) και ένα σημείο προς (to-point). Θέλουμε να τη χρησιμοποιήσουμε για να αντιπροσωπεύσουμε τις άκρες του πολυγώνου μας.

Ένα πολύγωνο-polygon, προφανώς, είναι ένα πολυστρωματικό σχήμα που περιέχει έναν αριθμό σημείων.

Ο σκοπός μας είναι να δημιουργήσουμε μια μέθοδο που να υπολογίζει αν είμαστε μέσα ή έξω από το πολύγωνο. Αυτή η μέθοδος θα πρέπει:

- Να υπολογίζει τις γραμμές του πολυγώνου
- Να φιλτράρει τις γραμμές που τέμνουν τον άξονα y
- Να υπολογίζει τα σημεία που οι γραμμές τέμνονται με τον άξονα y.
- Ταξινομεί τα σημεία ως προς τον άξονα x.
- Να χρησιμοποιεί τον αλγόριθμο Ray Casting για να ελέγχει αν είμαστε μέσα ή έξω από το πολύγωνο.

Ας πάρουμε κάθε σημείο ξεχωριστά:

- Υπολογισμός γραμμών πολυγώνου

Απλά παίρνουμε τα σημεία του πολυγώνου και τα ενώνουμε μεταξύ τους. Έπειτα κλείνουμε το πολύγωνο ενώνοντας το τελευταίο σημείο με το πρώτο.

```

01 List<Line> calculateLines(Polygon polygon) {
02     List<Line> results = new LinkedList<line>();
03
04     // get the polygon points
05     Point[] points = polygon.getPoints();
06
07     // form lines by connecting the points
08     Point lastPoint = null;
09     for (Point point : points) {
10         if (lastPoint != null) {
11             results.add(new Line(lastPoint, point));
12         }
13         lastPoint = point;
14     }
15
16     // close the polygon by connecting the last point
17     // to the first point
18     results.add(new Line(lastPoint, points[0]));
19
20     return results;
21 }

```

- Φιλτράρισμα γραμμών που τέμνουν τον άξονα-y

```

01 List<Line> filterIntersectingLines(List<Line> lines, double y) {
02     List<Line> results = new LinkedList<Line>();
03     for (Line line : lines) {
04         if (islineIntersectingAtY(line, y)) {
05             results.add(line);
06         }
07     }
08     return results;
09 }
10
11 boolean islineIntersectingAtY(Line line, double y) {
12     double minY = Math.min(
13         line.getFrom().getY(), line.getTo().getY()
14     );
15     double maxY = Math.max(
16         line.getFrom().getY(), line.getTo().getY()
17     );
18     return y > minY && y <= maxY;
19 }

```

- Υπολογισμός σημείων που οι γραμμές από τον άξονα-x τέμνουν τον άξονα-y
Χρησιμοποιούμε τη παρακάτω μέθοδο:

```

01 List<Point> calculateIntersectionPoints(
02     List<Line> lines, double y) {
03     List<Point> results = new LinkedList<Point>();
04     for (Line line : lines) {
05         double x = calculateLineXAtY(line, y);
06         results.add(new Point(x, y));
07     }
08     return results;
09 }
10
11 double calculateLineXAtY(Line line, double y) {
12     Point from = line.getFrom();
13     double slope = calculateSlope(line);
14     return from.getX() + (y - from.getY()) / slope;
15 }
16
17 double calculateSlope(Line line) {
18     Point from = line.getFrom();
19     Point to = line.getTo();
20     return (to.getY() - from.getY()) / (to.getX() - from.getX());
21 }

```

- Ταξινόμηση σημείων ως προς τον άξονα-x

```

1 void sortPointsByX(List<Point> points) {
2     Collections.sort(points, new Comparator<Point>() {
3         public int compare(Point p1, Point p2) {
4             return Double.compare(p1.getX(), p2.getX());
5         }
6     });
7 }

```

- Έλεγχος αν είμαστε μέσα ή έξω από το πολύγωνο χρησιμοποιώντας τον αλγόριθμο Ray Casting

```

01 boolean calculateInside(List<Point> sortedPoints, double x) {
02     boolean inside = false;
03     for (Point point : sortedPoints) {
04         if (x < point.getX()) {
05             break;
06         }
07         inside = !inside;
08     }
09     return inside;
10 }

```

Αρχικά είμαστε έξω από το πολύγωνο. Σε κάθε σημείο αλλάζουμε τη κατάσταση (inside – outside – inside – outside - ...). Μέχρι να φτάσουμε στη θέση-x. Τότε ξέρουμε αν είμαστε μέσα ή έξω από το πολύγωνο.

Η τελική μέθοδος που θα χρησιμοποιήσει όλα τα παραπάνω θα είναι :

```

1 boolean checkInside(Polygon polygon, double x, double y) {
2     List<Line> lines = calculateLines(polygon);
3     List<Line> intersectionLines = filterIntersectingLines(lines, y);
4     List<Point> intersectionPoints = calculateIntersectionPoints(intersectionLines, y);
5     sortPointsByX(intersectionPoints);
6     return calculateInside(intersectionPoints, x);
7 }

```

Google maps implementation

Search button

Αρχικά φτιάξαμε ένα Google Maps Activity και ακολουθήσαμε τις οδηγίες που υπάρχουν στα σχόλια μέσα σε αυτό για να πάρουμε ένα API Key το οποίο θα χρησιμοποιήσουμε στην εφαρμογή μας. Θα σχολιάσουμε το κώδικα που δημιουργήσαμε για να μπορούμε να έχουμε την επιλογή της αναζήτησης στο Maps Activity. Με την επιλογή αυτή έχουμε τη δυνατότητα στους χάρτες μας πληκτρολογώντας μια περιοχή να μας την εμφανίζει σε marker(πινέζα) την περιοχή ή περιοχές που έχουν το όνομα αυτό.

Screenshot Από τον Κώδικα:

```

case R.id.E_search: {
    EditText tf_location = (EditText) findViewById(R.id.TF_location);
    String location = tf_location.getText().toString();
    List<Address> addressList = null;
    MarkerOptions markerOptions = new MarkerOptions();
    Log.d("location = ", location);

    if (!location.equals("")) {
        Geocoder geocoder = new Geocoder(this);
        try {
            addressList = geocoder.getFromLocationName(location, 5);
        } catch (IOException e) {
            e.printStackTrace();
        }

        if (addressList != null) {
            for (int i = 0; i < addressList.size(); i++) {
                Address myAddress = addressList.get(i);
                LatLng latLng = new LatLng(myAddress.getLatitude(), myAddress.getLongitude());
                markerOptions.position(latLng);
                mMap.addMarker(markerOptions);
                mMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));
            }
        }
    }
}
break;

```

Μόλις ο χρήστης πατήσει το κουμπί «SEARCH» θα πάρουμε οτιδήποτε έχει γράψει στο text-field και έπειτα θα αναζητήσουμε αυτό το μέρος. Καταρχήν κάνουμε έναν έλεγχο ότι ο χρήστης έχει πληκτρολογήσει κάτι στο text-field και δεν είναι empty string και αν όντως αυτό ισχύει τότε μπορούμε να προχωρήσουμε παρακάτω όπου χρησιμοποιούμε την κλάση Geocoder και συγκεκριμένα τη μέθοδο getFromLocationName. Η συγκεκριμένη μέθοδος παίρνει δύο παραμέτρους και επιστρέφει μια λίστα από διευθύνσεις. Στη συνέχεια, με μια επανάληψη for βάζουμε markers στη λίστα των διευθύνσεων για να οπτικοποιήσουμε τις διευθύνσεις στο χάρτη.

Search nearest places

Στην εφαρμογή μας έχουμε προσθέσει την επιλογή να μπορεί ο χρήστης να κάνει αναζήτηση για σημεία που τον ενδιαφέρουν κοντά σε αυτόν όπως νοσοκομεία, σχολεία, καφετέριες κλπ. Επειδή η εφαρμογή μας έχει να κάνει με προσφορές έχουμε βάλει αναζήτηση για καφετέριες, μαγαζιά με ρούχα και εστιατόρια.

Θα χρειαστεί να δημιουργήσουμε 3 κλάσεις Java τις οποίες θα ονομάσουμε GetNearbyPlacesData, DownloadUrl και DataParser. Έπειτα φτιάχνουμε 3 κουμπιά στο αρχείο layout για κάθε ένα από τον τύπο σημείο ενδιαφέροντος που αναζητάμε.

Screenshot Μεθόδου readUrl():

```
public String readUrl(String myUrl) throws IOException //Για να διαβάσει το Url
{
    String data = "";
    InputStream inputStream = null;
    HttpURLConnection urlConnection = null;

    try {
        URL url = new URL(myUrl);
        urlConnection=(HttpURLConnection) url.openConnection(); // Initialize here
        urlConnection.connect();

        inputStream = urlConnection.getInputStream();
        BufferedReader br = new BufferedReader(new InputStreamReader(inputStream));
        StringBuffer sb = new StringBuffer();

        String line = ""; //at start is empty
        while((line = br.readLine()) != null) // read each line one by one
        {
            sb.append(line);
        }

        data = sb.toString();
        br.close();
    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    finally {
        inputStream.close();
        urlConnection.disconnect();
    }
    Log.d("DownloadURL", "Returning data= "+data);

    return data;
}
```

Στη κλάση DownloadUrl φτιάξαμε τη μέθοδο readUrl() που διαβάζει Url χρησιμοποιώντας το αντικείμενο HttpURLConnection και μεθόδους διαχείρισης αρχείων όπως είναι η Input stream και Buffered Reader. Στη συνέχεια αφού διαβάσουμε το αρχείο Url πρέπει να το αναλύσουμε. Έτσι φτιάξαμε τη κλάση DataParser όπου περιέχει 3 μεθόδους, τις getPlace(), getPlaces() και parse().

Screenshot Μεθόδου getPlaces():

```

private List<HashMap<String, String>>getPlaces(JSONArray jsonArray)
{
    int count = jsonArray.length();
    List<HashMap<String, String>> placelist = new ArrayList<>();
    HashMap<String, String> placeMap = null;

    for(int i = 0; i<count;i++)
    {
        try {
            placeMap = getPlace((JSONObject) jsonArray.get(i));
            placelist.add(placeMap);
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
    return placelist;
}

```

Screenshot Μεθόδου parse():

```

public List<HashMap<String, String>> parse(String jsonData)
{
    JSONArray jsonArray = null;
    JSONObject jsonObject;

    Log.d("json data", jsonData);

    try {
        jsonObject = new JSONObject(jsonData);
        jsonArray = jsonObject.getJSONArray("results");
    } catch (JSONException e) {
        e.printStackTrace();
    }
    return getPlaces(jsonArray);
}

```

Η μέθοδος parse() θα αναλύσει τα δεδομένα και θα τα περνάει στη μέθοδο getPlaces() η οποία θα τα αποθηκεύει σε μορφή λίστας από HashMap. Η getPlaces() με μια επανάληψη for θα περνάει κάθε μέρος ένα προς ένα και θα καλεί τη μέθοδο getPlace() για κάθε μέρος. Όσο για την getPlace() αυτό που κάνει είναι να μετατρέπει την jsonArray σε HashMap και να τη προσθέτει στη λίστα των HashMap.

Στη συνέχεια πάμε στο MapsActivity όπου το πρώτο που πρέπει να κάνουμε είναι να καθαρίσουμε τον χάρτη από markers οπότε γράφουμε mMap.clear();. Έπειτα θα φτιάξουμε ένα url και ένα αντικείμενο τύπου λίστας που θα αποθηκεύει 2 αντικείμενα. Το πρώτο αντικείμενο

θα είναι το mMap και το δεύτερο θα αποθηκεύει το url. Τέλος, φτιάχνουμε ένα αντικείμενο τύπου GetNearbyPlacesData() και το τρέχουμε με τον κώδικα GetNearbyPlacesData.execute(dataTransfer);.

Screenshot :

```
case R.id.B_restaurants:
    mMap.clear();
    dataTransfer = new Object[2];
    String restaurant = "restaurant";
    url = getUrl(latitude, longitude, restaurant);
    getNearbyPlacesData = new GetNearbyPlacesData();
    dataTransfer[0] = mMap;
    dataTransfer[1] = url;

    getNearbyPlacesData.execute(dataTransfer);
    Toast.makeText(MapsActivity.this, "Showing Nearby Restaurants", Toast.LENGTH_LONG).show();
    break;
```

Email Verification

Στην εφαρμογή μας όπως είπαμε και στο εγχειρίδιο χρήστη ο χρήστης μόλις κάνει εγγραφή θα πρέπει να ταυτοποιήσει το ηλεκτρονικό του ταχυδρομείο και αφού γίνει αυτό το βήμα τότε και μόνο τότε μπορεί να κάνει log in. Στον κώδικά μας έχουμε φτιάξει δύο μεθόδους που η μια κάνει αποστολή ενός μηνύματος ηλεκτρονικού ταχυδρομείου με σύνδεσμο όπου ο χρήστης πατώντας τον κάνει ταυτοποίηση του ηλεκτρονικού ταχυδρομείου του και η άλλη ελέγχει αν ο χρήστης έχει κάνει ταυτοποίηση. Η πρώτη μέθοδος καλείται μόλις πατήσει ο χρήστης στο κουμπί για εγγραφή και η δεύτερη μόλις προσπαθήσει να κάνει σύνδεση. Και στις δύο περιπτώσεις αν η εγγραφή ή η σύνδεση είναι επιτυχής καλείται η αντίστοιχη μέθοδος.

Screenshot Μεθόδου sendVerificationEmail():

```
public void sendVerificationEmail()
{
    if(firebaseAuth.getCurrentUser() ==null){
        Toast.makeText(MainActivity.this,
            "Failed to send verification email.",
            Toast.LENGTH_SHORT).show();
    }
    else {

        final FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();

        user.sendEmailVerification()
            .addOnCompleteListener((task) -> {
                if (task.isSuccessful()) {
                    // email sent
                    Toast.makeText(MainActivity.this, "Verification email
                    // after email is sent just logout the user and finish

                } else {
                    // email not sent, so display message and restart the

                    Toast.makeText(MainActivity.this,
                        "Failed to send verification email.",
                        Toast.LENGTH_SHORT).show();
                }
            });
    }
}
```

Στη `sendVerificationEmail()` κάνουμε έναν έλεγχο αν έχουμε συνδεδεμένο κάποιον χρήστη και αν ναι τότε συνεχίζουμε με την αποστολή του μηνύματος ηλεκτρονικού ταχυδρομείου μέσω της μεθόδου της Firebase `sendEmailVerification()`. Για να σταλεί στον χρήστη που είναι συνδεδεμένος πατάμε `user.sendEmailVerification()`. Έπειτα στέλνουμε και το αντίστοιχο μήνυμα αν πραγματοποιηθεί ή όχι η αποστολή.

Screenshot Μεθόδου checkEmailVerified():

```
private void checkIfEmailVerified()
{
    firebaseAuth=FirebaseAuth.getInstance();

    if (firebaseAuth.getCurrentUser().isEmailVerified())
    {
        // user is verified, so you can finish this activity or send user to activity which you want.
        finish();
        Toast.makeText(LoginActivity.this, "Successfully logged in", Toast.LENGTH_SHORT).show();
        startActivity(new Intent(this, HomePage.class));
    }
    else
    {
        // email is not verified, so just prompt the message to the user and restart this activity.
        // NOTE: don't forget to log out the user.
        Toast.makeText(LoginActivity.this, "In order to login verify your email", Toast.LENGTH_LONG).show();
        //restart this activity
    }
}
```

Στη μέθοδο `checkIfEmailVerified()` ελέγχουμε με μια `if` αν το ηλεκτρονικό ταχυδρομείο του χρήστη έχει ταυτοποιηθεί μέσω της μεθόδου `isEmailVerified` της Firebase. Εάν ναι τότε ανοίγει η εφαρμογή μας, δηλαδή η κεντρική σελίδα, και εάν όχι στέλνει σε μορφή μηνύματος `toast` ότι χρειάζεται ταυτοποίηση.

Συμπεράσματα και μελλοντικές επεκτάσεις

Υπάρχουν πολλές δυνατότητες χρησιμότητας του Geofencing, εμείς σταθήκαμε στο κομμάτι του marketing. Δηλαδή φτιάξαμε μια εφαρμογή που μπορεί να βοηθήσει από μικρούς ιδιοκτήτες καταστημάτων μέχρι μεγάλα εμπορικά κέντρα. Το push notification που χρησιμοποιεί η εφαρμογή μας θα μπορούσε μελλοντικά να παραπέμπει στο site του καταστήματος για παρακολούθηση των προσφορών ή ακόμα και οδηγίες πλοήγησης στο κατάστημα.

Επίσης θα μπορούσε να υπάρχει και λίστα από καταστήματα που είναι κοντά στον χρήστη και που έχουν προσφορές εκτός από ενός που βγάζει τώρα. Έτσι θα μπορούσε και ο χρήστης να έχει περισσότερες επιλογές. Ευελπιστώ σύντομα στο μέλλον να καταφέρω να προσθέσω και αυτές τις επιπλέον επιλογές σε μια νεότερη έκδοση της εφαρμογής.

Μελλοντικά εφόσον υπήρχαν καταστήματα που είναι εγγεγραμμένα στην εφαρμογή θα μπορούσε να υπήρχε και σελίδα όπου ο ίδιος ο ιδιοκτήτης με δικό του λογαριασμό να μπορεί να αλλάζει τις ειδοποιήσεις που στέλνει η εφαρμογή όταν κάποιος πλησιάζει στο δικό του κατάστημα. Έτσι θα μπορούσε να ενημερώνει αυτός όποτε θέλει για τις προσφορές που έχει ή οτιδήποτε άλλο σχετικό που θα τον βοηθούσε την επιχείρησή του. Δηλαδή να μπορεί ο ίδιος να αλλάζει το μήνυμα που εμφανίζεται και μέσα στην αρχική σελίδα και στο push notification. Αυτό θα βοηθούσε και τον administrator της εφαρμογής αλλά και τον ιδιοκτήτη του καταστήματος ο οποίος θα είχε τον έλεγχο στα χέρια του. Δεν μπορούμε να παραλείψουμε και το γεγονός ότι θα μπορούσε ο ιδιοκτήτης του καταστήματος και στατιστικά στοιχεία για την κίνηση γύρω από το κατάστημά του και με τη ποσοστό τα άτομα που βλέπουν μια προσφορά μπαίνουν και στο κατάστημα.

Οι δυνατότητες είναι πολλές, επιπλέον μια θα μπορούσε να είναι να προστεθούν σαν επέκταση της εφαρμογής τα πρατήρια βενζίνης που υπάρχουν κοντά στον χρήστη με τις τιμές των καυσίμων εκείνη την χρονική στιγμή. Έτσι όταν θα ήθελε να βάλει καύσιμα να ανοίγει το gps και να βλέπει που υπάρχουν γύρω του βενζινάδικα με τις τιμές.

Το geofencing είναι ένα εργαλείο που πραγματικά μπορεί να μας βοηθήσει στη καθημερινότητα μας αλλά θα πρέπει να χρησιμοποιούμε με προσοχή για να μην παραβιάζονται προσωπικά δεδομένα, ο χρήστης πρέπει να ενημερώνεται ρητά για το που δίνει τη συγκατάθεση του και που χρησιμοποιούνται τα δεδομένα που καταγράφει η εφαρμογή.

Ένας άλλος κίνδυνος αφορά την δυνατότητα παράκαμψης της ικανότητας τους για γεωγραφικό εντοπισμό ή και η αποστολή ψευδών πληροφοριών θέσης στους χρήστες. Αυτό σημαίνει στην πραγματικότητα ότι αν οι χρήστες χρησιμοποιούν τέτοιες εφαρμογές για να παρακολουθούν τους πόρους ή τους ανθρώπους, οι επιτιθέμενοι θα μπορούσαν να φανούν σαν αυτούς τους ανθρώπους ή ότι οι πόροι δεν είναι εκεί που πρέπει να είναι. Είδαμε κάποιες εφαρμογές για παρακολούθηση των παιδιών από τους γονείς τους, αυτό γίνεται για να είναι πιο ασφαλείς τα παιδιά. Αν η εφαρμογή δεν είναι σωστά φτιαγμένη ως φανταστούμε πόσο επικίνδυνο θα ήταν κάποιος να μπορέσει να πάρει αυτή τη πληροφορία.

Έτσι το να φτιάξουμε μια τέτοια εφαρμογή που να είναι διαθέσιμη στον κόσμο θα πρέπει να παρέχει όλα τα συστήματα ασφαλείας και να είναι έτσι φτιαγμένη ώστε να μην είναι ευάλωτη σε επιθέσεις. Η τεχνολογία μπαίνει στη ζωή μας για να μας την κάνει καλύτερη και πλέον αποτελεί βασικό κομμάτι. Όμως δεν πρέπει να ξεχνάμε και τους κινδύνους που μπορεί να προκύψουν και επιβάλλεται πλέον εάν χρησιμοποιούμε τέτοιου είδους εφαρμογές να προσπαθούμε να χρησιμοποιούμε αξιόπιστες και να προσέχουμε τις άδειες που δίνουμε στην εφαρμογή. Αυτό από την πλευρά των χρηστών αλλά και οι προγραμματιστές από τη μεριά μας πρέπει να δίνουμε ιδιαίτερη σημασία στο security κομμάτι της εφαρμογή και να μην επιτρέπουμε σε τρίτους να μπορούν να πάρουν δεδομένα από τους χρήστες που χρησιμοποιούν την εφαρμογή μας.

Βιβλιογραφία – πηγές

<https://developer.android.com/training/location/geofencing>

<https://developers.google.com/maps/documentation/android-sdk/polygon-tutorial>

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

<https://www.spyzie.com/geofence/best-geofence-apps.html>

<https://whatis.techtarget.com/definition/geofencing>

<https://www.codementor.io/flame3/send-push-notifications-to-android-with-firebase-du10860kb>

<https://firebase.googleblog.com/2017/02/email-verification-in-firebase-auth.html>

<https://www.w3schools.com/>