



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**  
**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**  
**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**«Εφαρμογή Android για άτομα που χρήζουν βοήθειας»**

**Ντιάνα Όνινα**

**Εισηγητής: Ιωάννης Έλληνας, Καθηγητής**

**ΑΘΗΝΑ ΜΑΙΟΣ 2019**

**(Κενό Φύλλο)**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Εφαρμογή Android για άτομα που χρήζουν βοήθειας**

**Όνινα Ντιάνα**

**A.M. : 42879**

**Εισηγητής:**

**Ιωάννης Έλληνας, Καθηγητής**

**Εξεταστική Επιτροπή:**

**Ιωάννης Έλληνας**

**Ιωάννης Αμοργίνος**

**Αναστασία Βελώνη**

**Ημερομηνία εξέτασης: 16 / 05 / 2019**

**(Κενό Φύλλο)**

## **ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

Ο/Η κάτωθι υπογεγραμμένος/η Όνινα Ντιάνα, του Νικολάι, με αριθμό μητρώου 42879 φοιτητής/τρια του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω: «Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο. Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης. Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασης της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

## ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό της ανάπτυξης εφαρμογών με καινούργιες τεχνολογίες για λογισμικό Android. Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου, τον οποίο θα ήθελα να ευχαριστήσω.

## ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία έχει σαν θέμα την ανάπτυξη υβριδικής εφαρμογής για κινητά τηλέφωνα και ταμπλέτες για λειτουργικό σύστημα Android. Το framework που επιλέχθηκε για την υλοποίηση και τον σχεδιασμό της εφαρμογής είναι το Ionic framework, το οποίο συνδυάζει τις τεχνολογίες και γλώσσες προγραμματισμού όπως HTML, CSS, JAVASCRIPT, NODEJS, APACHE CORDOVA, ANGULARJS και JSON. Στην αρχή θα παρουσιαστεί συνοπτικά η ιστορική εξέλιξη των εφαρμογών, λειτουργικά συστήματα κινητών συσκευών, οι πλατφόρμες ανάπτυξης και στη συνέχεια θα αναπτυχθεί η εφαρμογή, οι τεχνολογίες που χρησιμοποιήθηκαν για την κατασκευή της καθώς και μια αναφορά στην υποστηρικτική ιστοσελίδα της εφαρμογής.

## ABSTRACT

The subject of this thesis is the development of a hybrid application for mobile phones and tablets for Android operating system. The framework chosen to development and design the application is the Ionic framework, which combines technologies and programming languages as HTML, CSS, JAVASCRIPT, NODEJS, APACHE CORDOVA, ANGULARJS and JSON. In the beginning, will be briefly present the historical evolution of applications, mobile operating systems, development platforms, and subsequently the application, the technologies used to build it, and a reference to the application's support website.

## Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1 <sup>ο</sup> : Εφαρμογές Κινητών Συσκευών .....	10
1.2 Λειτουργικά Συστήματα κινητών συσκευών .....	11
1.3 Πλατφόρμες Ανάπτυξης Εφαρμογών .....	11
1.4 Είδη Εφαρμογών .....	12
1.4.1 Εγγενείς Εφαρμογές .....	12
1.4.2 Προοδευτικές Εφαρμογές Ιστού .....	13
1.4.3 Υβριδικές Εφαρμογές .....	14
1.5 Γιατί επιλέχθηκε η πολυεπίπεδη πλατφόρμα για την ανάπτυξη της εφαρμογής.....	15
ΚΕΦΑΛΑΙΟ 2 <sup>ο</sup> : Η εφαρμογή και η ιστοσελίδα DoApp.....	16
Εισαγωγή.....	16
2.1 Σκοπός .....	16
2.2 Η εφαρμογή DoApp.....	17
2.2.1 Λειτουργία Επιτήρησης.....	17
2.2.2 Λειτουργία Τηλεφώνων Έκτακτης Ανάγκης.....	20
2.2.3 Λειτουργία Τηλεφωνικού Καταλόγου.....	20
2.2.4 Βοήθεια Εφαρμογής .....	21
2.3 Ανάπτυξη της εφαρμογής .....	23
2.3.1 Πλατφόρμα Ανάπτυξης .....	23
2.3.2 Γλώσσες Προγραμματισμού .....	25
2.3.3 Βιβλιοθήκες.....	28
2.3.4 Πρόσθετα .....	28
2.3.5 Εργαλεία.....	31
2.4 Η υποστηρικτική ιστοσελίδα DoApp.....	31
2.5 Ανάπτυξη της ιστοσελίδας .....	35
2.5.1 Γλώσσες Προγραμματισμού & Εργαλεία ανάπτυξης .....	35
2.5.2 Βιβλιοθήκες και Εργαλεία.....	35
ΚΕΦΑΛΑΙΟ 3 <sup>ο</sup> : Ανάπτυξη της εφαρμογής DoApp .....	37
3.1 Η Εγκατάσταση των απαραίτητων προγραμμάτων .....	37
3.2 Η δομή των αρχείων και του κώδικα της εφαρμογής .....	39
Συμπεράσματα και προτάσεις .....	60
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	61



**(Κενό φύλλο)**

## ΚΕΦΑΛΑΙΟ 1<sup>ο</sup> : Εφαρμογές Κινητών Συσκευών

### 1.1 Η «βιομηχανία» εφαρμογών για κινητές συσκευές

Η δυναμική της αγοράς των κινητών τηλεφώνων φάνηκε ήδη από τα μέσα της δεκαετίας του 1990 [1] όπου ξεκίνησαν να κυκλοφορούν ευρέως και να καθιερώνονται σαν ένα αναπόσπαστο «αξεσουάρ» για τον καθένα. Η μορφή, το μέγεθος, οι ενσωματωμένες μικροεφαρμογές (διευθυνσιογράφοι, email, chat, media players, gps) και φυσικά τα παιχνίδια κρατούσαν το καταναλωτικό κοινό διαρκώς σε εγρήγορση.

Ωστόσο, οι κατασκευάστριες εταιρείες γνώριζαν ότι λόγω του μικρού μεγέθους των συσκευών και της περιορισμένης μνήμης τους στην επεξεργασία δεδομένων, δεν μπορούσαν να ανταγωνιστούν τον βασικό αντίπαλό τους, τους φορητούς υπολογιστές, οι οποίοι εκμεταλλευόμενοι και τις υπηρεσίες του διαδικτύου, όχι μόνο εξελίσσονταν (και εξελίσσονται) αλλά αποσπούσαν και ένα σημαντικό κομμάτι από την δική τους αγορά.

Την λύση στα προβλήματα αυτά, έδωσε μια ηγέτιδα εταιρεία του χώρου, των ηλεκτρονικών υπολογιστών αλλά και της κινητής τηλεφωνίας, το 2007. Η Apple Inc. παρουσιάζει το iPhone, το πρώτο «έξυπνο» κινητό τηλέφωνο (smartphone), το οποίο έχει δικό του λειτουργικό σύστημα, το iOS. Με οθόνη αφής για γρήγορη πλοήγηση σε μια πληθώρα εφαρμογών. Συγχρόνως, δημιουργεί το Apple App Store, όπου οι χρήστες μπορούν να «κατεβάσουν» και άλλες εφαρμογές για το κινητό τους.

Η καινοτομία αυτή της Apple Inc., επέφερε πέραν της προαναφερθείσας εμφανούς εξέλιξης, δύο ακόμη επίσης σημαντικές εξελίξεις. Πρώτον, «υποχρέωσε» και τις άλλες εταιρείες να ακολουθήσουν το παράδειγμά της προκειμένου να παραμείνουν στην αγορά και είτε αυτόνομα είτε σε συνεργασία με εταιρείες πληροφορικής να δημιουργήσουν «έξυπνα» κινητά με δικό τους λειτουργικό σύστημα και δικές τους εφαρμογές, γεγονός που οδήγησε σε ένα νέο ρεύμα εξέλιξης, αυτό των «έξυπνων» κινητών συσκευών. Δεύτερον, δημιουργήθηκε μια νέα κοινότητα προγραμματιστών επαγγελματιών ή ακόμη και ερασιτεχνών που αναπτύσσουν εφαρμογές για κινητές συσκευές (mobile apps).

Πάνω από δέκα χρόνια μετά από το πρώτο smartphone και τα πρώτα mobile apps, σήμερα υπάρχουν περίπου πέντε εκατομμύρια εφαρμογές, οι οποίες ως επί τω πλείστων, έχουν αναπτυχθεί για τα δύο κυρίαρχα λειτουργικά συστήματα το iOS και το Android και ο αριθμός αυτός αυξάνεται με γοργούς ρυθμούς [2].

## 1.2 Λειτουργικά Συστήματα κινητών συσκευών

Όπως προαναφέρθηκε, η βασική διαφορά των smartphones με τα κινητά τηλέφωνα είναι η ενσωμάτωση του λειτουργικού συστήματος. Όπως και στους ηλεκτρονικούς υπολογιστές, το λειτουργικό σύστημα είναι αυτό που προκαθορίζει τις βασικές λειτουργίες της συσκευής καθώς και τις δυνατότητες που θα έχει ως προς τις εφαρμογές που πρόκειται να εγκατασταθούν.

Ωστόσο, αυτή η «αποκλειστικότητα» που διεκδικούν τα λειτουργικά συστήματα από τις εφαρμογές εξαιτίας των περιορισμών που θέτουν, δημιούργησε ένα νέο πρόβλημα στην κοινότητα των προγραμματιστών, γνωστό ως «κατακερματισμός» [3] γιατί έπρεπε να «ξαναγράφουν» την εφαρμογή προσαρμόζοντάς την στο εκάστοτε λειτουργικό σύστημα, κάτι που θα ανέβαζε το κόστος και το χρόνο ανάπτυξης και συντήρησής της.

Το πρόβλημα εν τω μεταξύ δε σταματούσε εκεί, καθώς και κινητές συσκευές με το ίδιο λειτουργικό δεν ανταποκρίνονταν το ίδιο ή και καθόλου στις διάφορες εφαρμογές εξαιτίας και των τεχνικών τους προδιαγραφών, όπως για παράδειγμα το μέγεθος της οθόνης ή την διεπαφή με τα κουμπιά (buttons), τα χρώματα κ.α.

Ενδεικτικά τα δημοφιλέστερα λειτουργικά συστήματα κινητών συσκευών είναι :

iOS της Apple

Android της Google

Windows Phone της Microsoft

Symbian της Nokia

Blackberry Os της RIM

Firefox OS του Mozilla



Εικόνα 1

## 1.3 Πλατφόρμες Ανάπτυξης Εφαρμογών

Την λύση στο πρόβλημα του κατακερματισμού, με μια πρώτη ματιά φαίνεται να δίνουν οι πολυεπίπεδες πλατφόρμες ανάπτυξης. Στις πλατφόρμες αυτές, δίνεται η δυνατότητα στους προγραμματιστές να αναπτύσσουν τις εφαρμογές τους μία φορά και αυτές να «αναγνωρίζονται» σε περισσότερα από ένα λειτουργικά συστήματα.

Σήμερα, υπάρχουν αρκετές τέτοιες πλατφόρμες που διευκολύνουν το έργο των προγραμματιστών, μειώνοντας το χρόνο και το κόστος ανάπτυξης μιας εφαρμογής για

κινητές συσκευές, αφού δεν χρειάζεται να μαθαίνουν νέες γλώσσες προγραμματισμού προκειμένου να προσαρμόζουν την εφαρμογή τους σε κάθε λειτουργικό σύστημα, ενώ με τον κατάλληλο σχεδιασμό (Responsive Web Design) οι web apps μπορούν και προσαρμόζονται στα διαφορετικά μεγέθη οθονών των συσκευών (κινητών τηλεφώνων και ταμπλετών).

Βέβαια πρέπει να σημειωθεί ότι οι πλατφόρμες αυτές, αναπτύχθηκαν με γλώσσες και τεχνολογίες διαδικτύου και κατ' επέκταση για εφαρμογές που ο βασικός πυλώνας ανάπτυξής τους είναι το διαδίκτυο και όχι το λειτουργικό σύστημα ή η εκάστοτε κινητή συσκευή, με όποια πλεονεκτήματα και μειονεκτήματα μπορεί να προκύπτουν από αυτό.

## **1.4 Είδη Εφαρμογών**

Ανάλογα με τον τρόπο με τον οποίο οι mobile apps εκτελούνται από τα smartphones, διακρίνονται σε: α) Εγγενείς Εφαρμογές (Native Applications), β) Προοδευτικές Εφαρμογές Ιστού (Progressive Web Applications) και γ) Υβριδικές Εφαρμογές (Hybrid Applications) [4].

### **1.4.1 Εγγενείς Εφαρμογές**

Πρόκειται για εκτελέσιμα αρχεία τα οποία βρίσκονται ήδη στην συσκευή ως τμήμα του λειτουργικού συστήματος ή που θα πρέπει ο χρήστης να τα κατεβάσει από το εκάστοτε διαδικτυακό κατάστημα εφαρμογών και να τα εγκαταστήσει στη συσκευή του. Εκτελώντας για πρώτη φορά την εκάστοτε εφαρμογή, αυτή συνδέεται με το λειτουργικό σύστημα και αποκτά πλήρη έλεγχο στην συσκευή.

Για την ανάπτυξη των native apps, οι προγραμματιστές πρέπει να χρησιμοποιούν το περιβάλλον ανάπτυξης εφαρμογών (SDK) που περιλαμβάνει εργαλεία, βιβλιοθήκες και επιπλέον πρόσθετα που τους παρέχει η εταιρεία δημιουργίας του εκάστοτε λειτουργικού συστήματος. Αν και ο τρόπος ανάπτυξης των εφαρμογών είναι σχεδόν ίδιος ανεξάρτητα από το λειτουργικό σύστημα, ωστόσο τα περιβάλλοντα ανάπτυξής τους διαφέρουν μεταξύ τους και σε συνδυασμό με το ότι οι προγραμματιστές πρέπει να γράφουν σε πηγαίο κώδικα, στην γλώσσα προγραμματισμού που υποστηρίζει το εκάστοτε λειτουργικό σύστημα, αυτό αποτελεί ένα σημαντικό μειονέκτημα για τις native apps – όπως αναφέρθηκε και στην προηγούμενη ενότητα.

Ωστόσο, η σημασία των native apps που τις καθιστούν ακόμη απαραίτητες, βασίζεται σε ένα άλλο χαρακτηριστικό των smartphones. Στη διεπαφή προγραμματισμού εφαρμογών (API's), την αλληλεπίδραση, δηλαδή, της εφαρμογής με το λειτουργικό σύστημα και τη

χρήση των διεπαφών που περιλαμβάνει. Οι API's μπορεί να αφορούν την αλληλεπίδραση μιας εφαρμογής με την οθόνη αφής, με την εισαγωγή ή την παραγωγή ήχων μέσω του μικροφώνου και των ηχείων αντίστοιχα ή να προσφέρουν υπηρεσίες, όπως οι τηλεφωνικές κλήσεις και τα γραπτά μηνύματα, η πλοήγηση στο διαδίκτυο κ.α..

Αν και οι υπηρεσίες αυτές είναι ήδη εγκατεστημένες στα λειτουργικά συστήματα μέσω των αντίστοιχων native apps, ο τρόπος και το εύρος των υπηρεσιών που αλληλεπιδρούν μπορεί να διαφέρει, αναπτύσσοντας έτσι μια νέα εφαρμογή που περιλαμβάνει αυτά τα χαρακτηριστικά.

Πέραν των API's, ένα ακόμη γνώρισμα των smartphones, που αποδεικνύει την αναγκαιότητα της «αποκλειστικότητας» που επιδιώκουν τα λειτουργικά συστήματα από τα native apps, είναι η γραφική διεπαφή (GI). Πρόκειται για ένα σύνολο εργαλείων, από γραφικά χαρακτηριστικά που παρέχει στις native apps, κάθε λειτουργικό σύστημα, όπως το μενού, οι μπάρες κύλισης, τα κουμπιά, οι ειδοποιήσεις κ.α.

Οι γραφικές διεπαφές, οι οποίες συχνά διαφέρουν και μεταξύ διαφορετικών εκδόσεων του ίδιου λειτουργικού συστήματος, εξοικειώνουν τον χρήστη με ένα συγκεκριμένο γραφικό περιβάλλον, το οποίο παρουσιάζεται μέσω των native apps.

#### **1.4.2 Προοδευτικές Εφαρμογές Ιστού**

Σε αντίθεση με τις native apps, οι progressive web apps δεν εκτελούνται στο λειτουργικό σύστημα της κινητής συσκευής αλλά στον περιηγητή του ιστού (browser). Αυτό σημαίνει ότι ο τελικός χρήστης για να εκτελέσει μια διαδικτυακή εφαρμογή δεν χρειάζεται να την εγκαταστήσει στη συσκευή του, αλλά αρκεί να επισκεφθεί την αντίστοιχη ηλεκτρονική διεύθυνση (URL) στον περιηγητή του.

Ωστόσο, ο ίδιος λόγος που κάνει ελκυστικές τις progressive web apps για τους χρήστες και τους developers –η αποφυγή του κατακερματισμού- είναι και αυτός που τους προσάπτει κάποια μειονεκτήματα. Καταρχάς, επειδή ακριβώς, δεν έχουν άμεση πρόσβαση στη συσκευή, αυτό αυξάνει τον χρόνο απόκρισής τους, αφού πρέπει να προηγηθεί το αίτημα και η απάντηση προς/από τον περιηγητή, ενώ παράλληλα η λειτουργία της εκάστοτε εφαρμογής δεν εξαρτάται μόνο από την συσκευή αλλά και από την ταχύτητα (επεξεργαστική ικανότητα) του περιηγητή.

Επίσης, το γεγονός ότι δεν χρειάζεται να επισκεφθεί ο χρήστης κάποιο αναγνωρισμένο “store” από την εταιρεία ανάπτυξης λογισμικού, για να εγκαταστήσει την εφαρμογή, άρα δεν υπάρχει έλεγχος ασφάλειας ή/και συμβατότητας με τη συσκευή, αφήνει κενά

ασφαλείας τόσο για την ορθή λειτουργία της συσκευής όσο και για τα προσωπικά δεδομένα του χρήστη.

Τέλος, αξίζει να σημειωθεί ότι είναι τόσο ραγδαία η εξέλιξη των γλωσσών – εργαλείων για την ανάπτυξη των web εφαρμογών, με αποτέλεσμα κάποιες φορές, οι περιηγητές ή ακόμη και τα λειτουργικά συστήματα των κινητών συσκευών να μην τις υποστηρίζουν πλήρως.

### 1.4.3 Υβριδικές Εφαρμογές

Οι υβριδικές εφαρμογές προέκυψαν από τον συνδυασμό των δύο παραπάνω τρόπων ανάπτυξης εφαρμογών, προκειμένου να καλύψουν κενά ή ατέλειες της λειτουργικότητάς τους. Στην ουσία αυτό που επιτυγχάνεται με τις υβριδικές εφαρμογές είναι η ανάπτυξη της εκάστοτε εφαρμογής να γίνεται με την χρήση web τεχνολογιών αλλά συγχρόνως να διατηρείται και η πρόσβαση στα χαρακτηριστικά της εκάστοτε κινητής συσκευής μέσω της διεπαφής του προγραμματισμού των εφαρμογών API's κάτι που είναι δυνατό να γίνει μόνο στις εγγενείς εφαρμογές.

Η ανάπτυξη των hybrid apps γίνεται σε πολυεπιπέδες πλατφόρμες (cross-platforms), όπου συντάσσονται τα τμήματα κώδικα που σχετίζονται με τις web τεχνολογίες, τα οποία είναι κοινά για όλες τις συσκευές και η μόνη διαφοροποίηση αφορά τα τμήματα που σχετίζονται με τα εκάστοτε λειτουργικά συστήματα.

Μεγάλη μερίδα της κοινότητας των apps developers, στράφηκε προς αυτήν κατεύθυνση για την ανάπτυξη εφαρμογών, καθώς τα συνδυαστικά οφέλη ήταν πολλά και οι εταιρείες. Μάλιστα, η αποδοχή είναι τόσο μεγάλη ώστε σήμερα να μιλάμε πλέον για ανταγωνισμό μόνο μεταξύ native και hybrid apps αφήνοντας σε «δεύτερη μοίρα» τις web apps.



Εικόνα 2

## 1.5 Γιατί επιλέχθηκε η πολυεπίπεδη πλατφόρμα για την ανάπτυξη της εφαρμογής

Είναι γεγονός πως δεν μπορούμε ποτέ τελικά να απαντήσουμε για το ποια είναι η βέλτιστη επιλογή και το αν κάποιος πρέπει να σχεδιάσει εγγενή ή υβριδική εφαρμογή για την επίτευξη των στόχων του. Οι τεχνολογίες που πρέπει να επιλέγουμε κάθε φορά για την ανάπτυξη και την σχεδίαση μιας εφαρμογής εξαρτώνται άμεσα από τους στόχους της εφαρμογής, τις λειτουργίες της, τις ανάγκες των χρηστών και τον χρόνο που έχουμε να διαθέσουμε για την δημιουργία της εφαρμογής.

Για την επίτευξη των στόχων της παρούσας εφαρμογής στα πλαίσια της πτυχιακής εργασίας δεν κρίθηκε απαραίτητη η χρήση και η διασύνδεση με κάποια εξειδικευμένη λειτουργία η οποία συνδέεται άμεσα με το λειτουργικό σύστημα της κάθε συσκευής. Ένα μεγάλο πλεονέκτημα των υβριδικών εφαρμογών είναι η γρήγορη εγκατάσταση μέσω `npm - ready start-up package` και πληθώρα `testing tools` που παρέχει το ίδιο `framework`, πολλές `third-party` βιβλιοθήκες, αλλά και η ίδια η φύση των υβριδικών εφαρμογών η ανάπτυξη και μέρος του `debugging` των οποίων μπορεί να γίνει απευθείας σε έναν `web browser`.

Η φύση των πολυεπίπεδων πλατφορμών και ο τρόπος σχεδίασης των υβριδικών εφαρμογών με την βοήθειά τους, μου φάνηκε αρκετά ενδιαφέρον αντικείμενο μελέτης. Στα πλαίσια της έρευνας και της αναζήτησης των κατάλληλων εργαλείων για την ανάπτυξη της εφαρμογής κατέληξα στην χρήση του `Ionic framework` για τους λόγους που αναλύονται περισσότερο στο κεφάλαιο 2.3.1.

## ΚΕΦΑΛΑΙΟ 2ο : Η εφαρμογή και η ιστοσελίδα DoApp

### Εισαγωγή

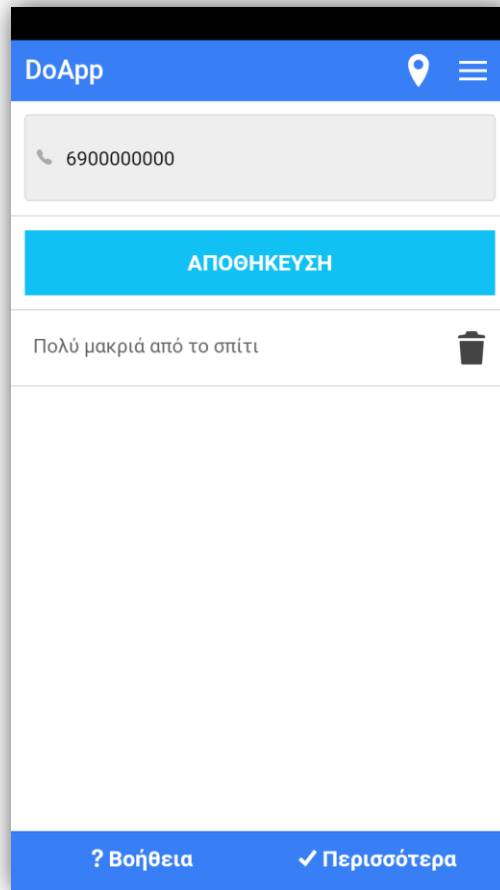
Σε αυτό το κεφάλαιο θα γίνει περιγραφή του σκοπού και της μεθοδολογίας ανάπτυξης της εφαρμογής η οποία πραγματοποιήθηκε στα πλαίσια της παρούσας πτυχιακής και η οποία θα αναφέρεται στο εξής ως “DoApp”. Ειδικότερα, θα αναφερθούμε στην πλατφόρμα ανάπτυξης Apache Cordova και στο Ionic framework, καθώς και στα υπόλοιπα «τεχνολογικά εργαλεία» που χρησιμοποιήθηκαν για την ανάπτυξή της. Επίσης, θα γίνει αναφορά, στην υποστηρικτική ιστοσελίδα DoApp, στα αντίστοιχα «εργαλεία» που χρησιμοποιήθηκαν καθώς και στο λόγο που θεωρήθηκε απαραίτητη για να πλαισιώνει την εφαρμογή.

### 2.1 Σκοπός

Βασικός σκοπός της εφαρμογής DoApp, είναι η εξ αποστάσεως και διακριτική επιτήρηση ατόμων που χρίζουν κάποιας «κηδεμονίας», είτε λόγω ηλικίας είτε για λόγους υγείας, περιφρουρώντας με τον τρόπο αυτό την ασφάλειά τους. Πιο συγκεκριμένα, η εφαρμογή με τη χρήση του συστήματος γεωεντοπισμού (GPS), μπορεί να αναγνωρίζει τη θέση που βρίσκεται ο χρήστης και μέσω ειδοποίησης (notification) να ειδοποιεί καταρχάς τον ίδιο και εν συνεχεία ένα δεύτερο άτομο μέσω μηνύματος (στο εξής «κηδεμόνα») για αυτήν.

Πρέπει να σημειωθεί, ότι δεν πρόκειται για ένα σύστημα παρακολούθησης των κινήσεων του χρήστη, αλλά για ένα σύστημα ειδοποίησης, το οποίο ενεργοποιείτε μόνο εφόσον ο κάτοχος της κινητής συσκευής ξεπεράσει την περίμετρο που έχει προκαθορισθεί με βάση ένα συγκεκριμένο σημείο εκκίνησης. **(εικόνα 3).**





Εικόνα 3

## 2.2 Η εφαρμογή DoApp

Όπως προαναφέρθηκε, η ομάδα- στόχος της εφαρμογής είναι άτομα τα οποία χρίζουν κάποιας κηδεμονίας, ενώ είναι πιθανό να είναι και τεχνολογικά ακατάρτιστα. Επομένως, πρώτο μέλημα για την ανάπτυξή της ήταν να είναι φιλική προς τον χρήστη. Με δεδομένη την ύπαρξη ενός τουλάχιστον «κηδεμόνα», που θα βοηθήσει στην εγκατάσταση της εφαρμογής και στην εισαγωγή των δεδομένων, ο τελικός χρήστης, το μόνο που έχει να κάνει είναι να διαβάσει τις πληροφορίες της οθόνης και να πατάει το αντίστοιχο κουμπί της επιλογής που επιθυμεί.

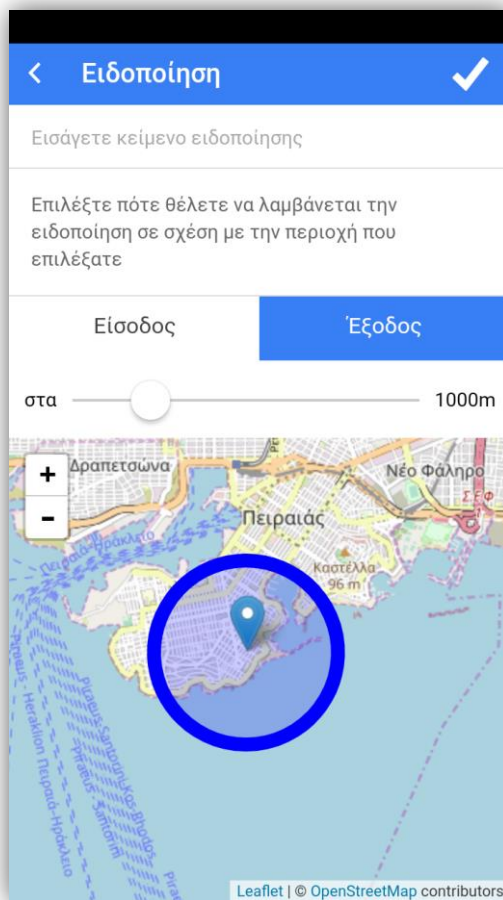
Παρακάτω βλέπουμε τις τρεις λειτουργίες, που περιλαμβάνονται στην εφαρμογή DoApp:

### 2.2.1 Λειτουργία Επιτήρησης

Στην εικόνα 4 βλέπουμε την οθόνη της συγκεκριμένης λειτουργίας.

Αφού οριστεί το τηλέφωνο ειδοποίησης του κηδεμόνα ο οποίος θα λαμβάνει το μήνυμα σε περίπτωση που ο χρήστης βγει εκτός ακτίνας στη συνέχεια δημιουργούμε την ειδοποίηση στην οποία ορίζονται: η αρχική τοποθεσία, όπου βρίσκεται ο χρήστης

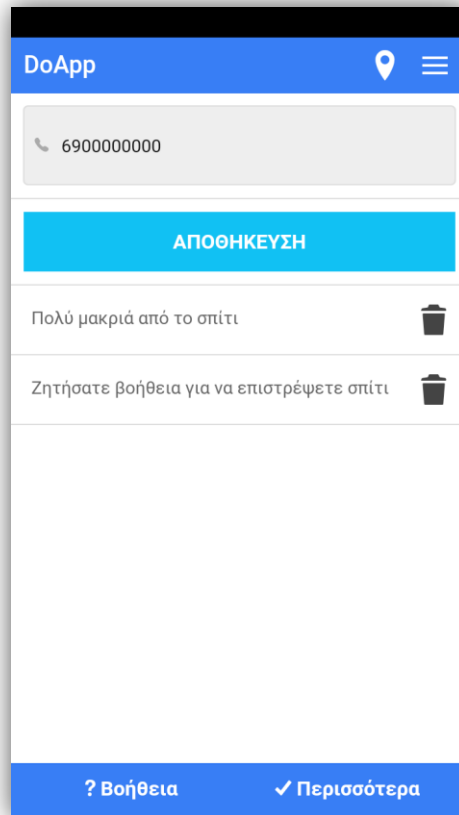
(σημείο αναφοράς), η περίμετρος ελέγχου (ακτίνα 50-5000 μέτρα) και το μήνυμα ειδοποίησης που θα του εμφανίζεται στην οθόνη όταν την υπερβεί.



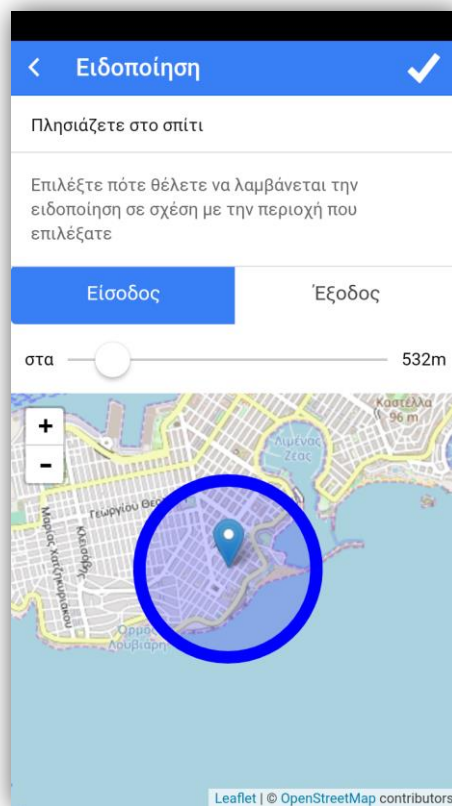
Εικόνα 4

Πρέπει να σημειωθεί ότι μπορούμε να :

- Αλλάξουμε όλα τα παραπάνω στοιχεία (αρχική τοποθεσία, ακτίνα περιμέτρου ελέγχου, μήνυμα, τηλέφωνο κηδεμόνα) (εικόνα 4)
- Ορίσουμε περισσότερες από μία περιμέτρους ελέγχου (εικόνα 5)
- Ορίσουμε να μας έρχεται μήνυμα κατά την έξοδο, κατά την είσοδο ή και κατά την είσοδο και κατά την έξοδο του χρήστη από την περίμετρο ελέγχου (εικόνα 6)



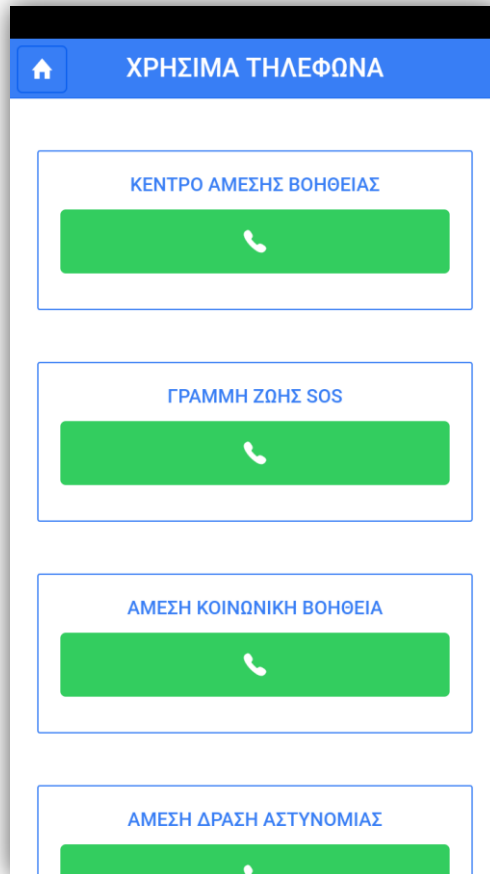
Εικόνα 5



Εικόνα 6

### 2.2.2 Λειτουργία Τηλεφώνων Έκτακτης Ανάγκης

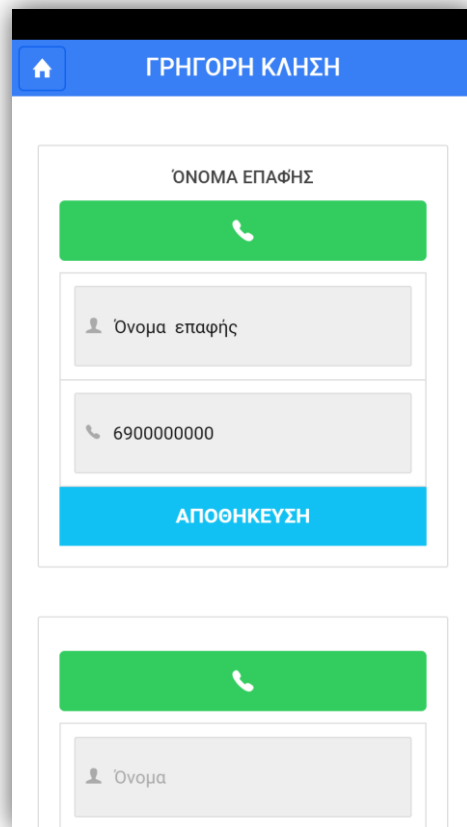
Ενισχύοντας τον σκοπό της εφαρμογής μας, θεωρήθηκε σκόπιμο να συμπεριληφθούν σε αυτή, μερικά τηλέφωνα άμεσης ανάγκης (ΕΚΑΒ, Άμεση Δράση, Πυροσβεστική, κ.λπ.) ώστε ο χρήστης να μπορεί να τα καλεί χωρίς να χρειάζεται να πληκτρολογεί τα νούμερα (π.χ σε περίπτωση που δεν τα θυμάται ή απλά δεν έχει τις γνώσεις ή/και την κατάλληλη εξοικείωση με την συσκευή του για να το κάνει) (εικόνα 6).



Εικόνα 7

### 2.2.3 Λειτουργία Τηλεφωνικού Καταλόγου

Όπως και με την λειτουργία των τηλεφώνων έκτακτης, έτσι και σε αυτήν την λειτουργία ορίζονται μερικά τηλέφωνα (φίλων, συγγενών, κ.λπ.), ώστε να έχει άμεση πρόσβαση ο χρήστης εικόνα 8, χωρίς να χρειάζεται να πληκτρολογεί τον αριθμό ή να μεταβεί στον τηλεφωνικό κατάλογο της συσκευής του που μπορεί να είναι μακροσκελής.

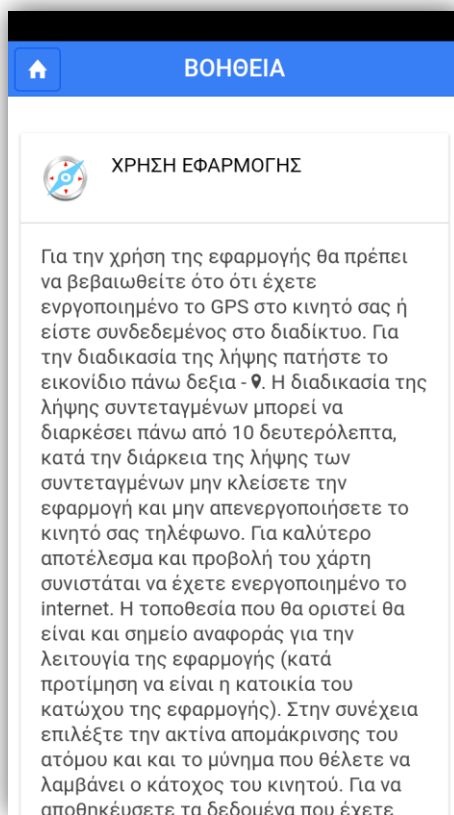


Εικόνα 8

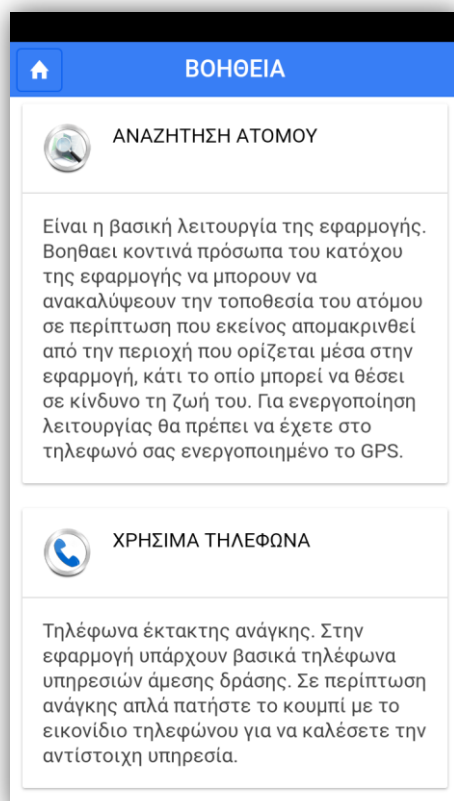
Γίνεται επομένως κατανοητό, ότι όλες οι λειτουργίες της εφαρμογής DoApp έχουν σαν στόχο να προστατεύσουν τον χρήστη αλλά ταυτόχρονα και να του διασφαλίσουν το δικαίωμα να κινείται ελεύθερα χωρίς άμεση επιτήρηση.

## 2.2.4 Βοήθεια Εφαρμογής

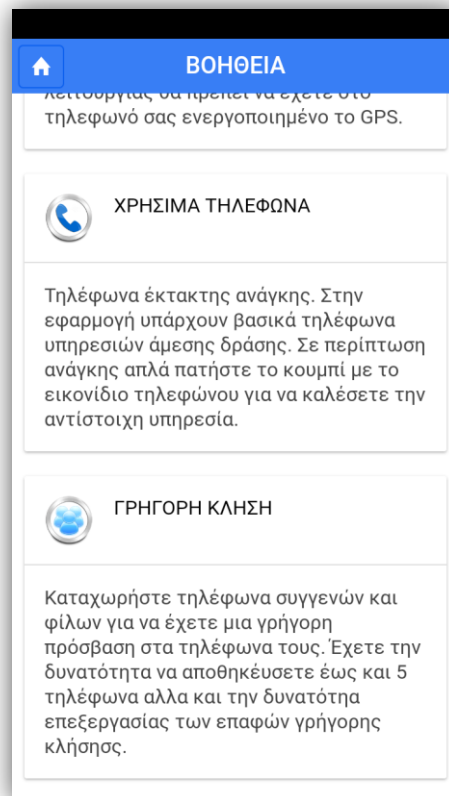
Για την κατανόηση της λειτουργίας της εφαρμογής και των επιμέρους στοιχείων της καθώς και για τις συμβουλές για την καλύτερη απόδοσή της δημιουργήθηκε ένα επιπλέον μενού «Βοήθεια». Στην συγκεκριμένη σελίδα σε κάθε καρτέλα ξεχωριστά υπάρχει μια μικρή περιγραφή και κάποιες οδηγίες για τις λειτουργίες της εφαρμογής. Στην καρτέλα «ΧΡΗΣΗ ΕΦΑΡΜΟΓΗΣ» (εικόνα 9) περιγράφεται η βασική λειτουργία της εφαρμογής καθώς και οι απαραίτητες προϋποθέσεις για τον ορισμό της τοποθεσίας (σημείο αναφοράς). Στην καρτέλα «ΑΝΑΖΗΤΗΣΗ ΑΤΟΜΟΥ» υπάρχει η περιγραφή του σκοπού της λειτουργίας εντοπισμού (εικόνα 10) και στην εικόνα 11 βλέπουμε τις καρτέλες «ΧΡΗΣΙΜΑ ΤΗΛΕΦΩΝΑ» όπου περιγράφεται ο σκοπός της συγκεκριμένης λειτουργίας και οι οδηγίες χρήσης της καθώς και η καρτέλα «ΓΡΗΓΟΡΗ ΚΛΗΣΗ» με τον σκοπό και τις δυνατότητες της συγκεκριμένης λειτουργίας.



Εικόνα 9



Εικόνα 10



Εικόνα 11

## 2.3 Ανάπτυξη της εφαρμογής

Η ανάπτυξη της εφαρμογής, στηρίζεται στην πλατφόρμα Apache Cordova και στο Ionic Framework, ενώ για την ολοκλήρωσή της χρησιμοποιήθηκαν και άλλες γλώσσες προγραμματισμού, εργαλεία (tools), βιβλιοθήκες (libraries) και πρόσθετα (plugins), όπως θα δούμε στις επόμενες παραγράφους του κεφαλαίου.

### 2.3.1 Πλατφόρμα Ανάπτυξης



Εικόνα 12

Το **Ionic Framework** είναι ένα από τα πιο δημοφιλή για την ανάπτυξη των υβριδικών εφαρμογών. Πρόκειται για ένα open source SDK (Software Development Kit - Kit Ανάπτυξης Λογισμικού) ανάπτυξης εφαρμογών για κινητά και ταμπλέτες του οποίου το UI (User Interface)

βασίζεται κατά κύριο λόγο σε markup γλώσσα HTML5, αλλά και σε native-styled στοιχεία (elements). Το μεγάλο πλεονέκτημά του είναι ότι το Ionic διαθέτει ένα πλούσιο σύνολο προηγμένων λειτουργιών όπως π.χ mobile components, icons κ.α, Plugins τα οποία μπορούν να χρησιμοποιηθούν για επιπλέον λειτουργίες ή/και για την διασύνδεση με native λειτουργίες και το OS της συσκευής [5]. Ένα άλλο μεγάλο πλεονέκτημα του Ionic

είναι η σχετικά εύκολη και γρήγορη διαδικασία testing & debugging. Οι διαδικασίες αυτές γίνονται αρκετά πιο απλές και γρήγορες χάρη στη φύση των υβριδικών εφαρμογών που ένα μεγάλο μέρος τους μπορεί να σχεδιαστεί, να δοκιμαστεί και να γίνει η αποσφαλμάτωση απευθείας σε έναν web browser ή σε μια κινητή συσκευή σε real time. Επίσης η ίδια η πλατφόρμα παρέχει κάποια εργαλεία testing ή/και την δυνατότητα προσθήκης έξτρα βιβλιοθηκών και εφαρμογών που βοηθάνε σε αυτή την διαδικασία [6]. Τέλος υπάρχει αναλυτικός οδηγός (documentation) στην επίσημη ιστοσελίδα που καλύπτει όλα τα στάδια της ανάπτυξης καθώς επίσης υπάρχει και μεγάλη ενεργή κοινότητα που παρέχει τεράστια γάμα παραδειγμάτων (tutorials).

Το **Apache Cordova** είναι μία από τις πιο δημοφιλείς πλατφόρμες ανάπτυξης υβριδικών εφαρμογών, ανοικτού κώδικα για κινητές συσκευές [7] καθώς επιτρέπει στους προγραμματιστές εφαρμογών να χρησιμοποιούν JavaScript, HTML5 και CSS3, αντί να βασίζονται στα APIs της εκάστοτε πλατφόρμας Android, iOS, Windows Phone κ.λπ.. (βλ. Κεφάλαιο 1).



Εικόνα 13

Ο πυρήνας των εφαρμογών Apache Cordova χρησιμοποιεί HTML5 και CSS3 για την «εμφάνιση» και JavaScript για τη «λογική» τους. Η HTML5 σε συνδυασμό με την JavaScript (ή κάποιου framework της) παρέχουν πλέον πρόσβαση στο υποκείμενο υλικό, όπως το επιταχυνσιόμετρο, τη φωτογραφική μηχανή, και το GPS. Ωστόσο, η υποστήριξη του προγράμματος περιήγησης για HTML5 στη συσκευή δεν είναι συνεπής σε όλους τους περιηγητές φορητών συσκευών, ιδιαίτερα στις παλαιότερες εκδόσεις του Android.

Για να ξεπεραστούν αυτοί οι περιορισμοί, το Apache Cordova ενσωματώνει κώδικα HTML5 μέσα σε ένα εγγενές WebView στη συσκευή, χρησιμοποιώντας μια διεπαφή ξένων λειτουργιών για να αποκτηθεί πρόσβαση στους εγγενείς πόρους της συσκευής. Η Apache Cordova μπορεί να επεκταθεί με ενσωματωμένη δυνατότητα χρήσης συνδέσεων (plugins) που επιτρέπουν στους προγραμματιστές να προσθέσουν λειτουργικότητα που μπορεί να κληθεί με χρήση της JavaScript, επιτρέποντας την άμεση επικοινωνία μεταξύ του φυσικού στρώματος και της σελίδας HTML5.

Η χρήση της Apache Cordova είναι απαραίτητη για την κατασκευή μιας υβριδικής εφαρμογής και χωρίς αυτή το Ionic (ή και άλλα mobile frameworks) δεν θα μπορούσε να λειτουργήσει και να έχει την λειτουργικότητα και την εμφάνιση μιας εφαρμογής.



### 2.3.2 Γλώσσες Προγραμματισμού

Στην ενότητα αυτή θα δούμε τις γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής.

Όπως αναφέρθηκε και στα προηγούμενα κεφάλαια το μεγαλύτερο πλεονέκτημα των υβριδικών εφαρμογών είναι ο γρήγορος σχεδιασμός τους για όλα τα λογισμικά ταυτόχρονα που επιτυγχάνεται με τη χρήση των γλωσσών προγραμματισμού και εργαλείων που χρησιμοποιούνται στο web development. Συγκεκριμένα το Cordova Apache χρησιμοποιεί HTML5, CSS3 & JAVASCRIPT και το Ionic Framework χρησιμοποιεί όλα τα παραπάνω με τη διαφορά ότι για την «λογική» της εφαρμογής χρησιμοποιείται ένα δομικό πλαίσιο (structural framework) ANGULARJS. Μεγάλο πλεονέκτημα του Ionic Framework επίσης είναι η ευελιξία του, έτσι για την επίτευξη των στόχων της κάθε εφαρμογής και για μεγαλύτερο δυναμισμό της, μπορούν εύκολα να χρησιμοποιηθούν και άλλες γλώσσες προγραμματισμού ανάλογα με τις δυνατότητες αι τις προτιμήσεις του κάθε προγραμματιστή.

Συγκεκριμένα για την δημιουργία της εφαρμογής DoApp χρησιμοποιήθηκαν οι παρακάτω γλώσσες:

#### Για την σήμανση και εμφάνιση:

- ✓ HTML5
- ✓ CSS3
- ✓ SASS
- ✓ JAVASCRIPT

#### Για τον προγραμματισμό και την λειτουργικότητα:

- ✓ ANGULARJS
- ✓ JSON
- ✓ JAVASCRIPT

Στη συνέχεια θα δούμε πιο αναλυτικά τι κάνει η κάθε γλώσσα και τα ιδιαίτερα χαρακτηριστικά τους.

## **HTML5:**

Η HTML5 (HyperText Markup Language - Γλώσσα Σήμανσης Υπερκειμένου) είναι γλώσσα σήμανσης, καθορίζει τις ιδιότητες και συμπεριφορές του περιεχομένου της ιστοσελίδας, εφαρμόζοντας σε αυτό ένα μοτίβο με βάση τη σήμανση. Είναι η πέμπτη και τελευταία μέχρι τώρα έκδοση της HTML. Η HTML5 περιλαμβάνει λεπτομερή μοντέλα για εμφάνιση μοντέρνων και σύνθετων διαδικτυακών εφαρμογών, επεκτείνει και βελτιώνει τη διαθέσιμη σήμανση για έγγραφα και εισάγει διεπαφές προγραμματισμού σήμανσης και εφαρμογών (API) για σύνθετες εφαρμογές ιστού. Για τους παραπάνω λόγους η HTML5 είναι επίσης δημοφιλή για την σήμανση κατά τον σχεδιασμό εφαρμογών για συσκευές κινητής τηλεφωνίας σε πολλαπλές πλατφόρμες [8].

## **CSS3:**

Η CSS (Cascading Style Sheets - Διαδοχικά Φύλλα Ύφους) ή (αλληλουχία φύλλων ύφους) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων ύφους που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στυλιστικά μια ιστοσελίδα δηλαδή να διαμορφώσει την εμφάνισή της (π.χ. χρώματα, γραμματοσειρές, στοίχιση κ.α.). Η CSS3 είναι η τελευταία εξέλιξη της γλώσσας Cascading Style Sheets και αποσκοπεί στην προσθήκη ακόμα περισσότερων δυνατοτήτων στην εμφάνιση εγγράφου σήμανσης. Φέρνει πολλά νέα χαρακτηριστικά και προσθήκες, όπως στρογγυλεμένες γωνίες, σκιές, κλίσεις, μεταβάσεις ή προσθήκη κίνησης (animations), καθώς και νέες διατάξεις όπως πολλαπλές στήλες, εύκαμπτες διατάξεις κουτιών ή πλέγματος [9].

## **SASS:**

Το Sass είναι γλώσσα που ανήκει στην κατηγορία των γλωσσών φύλλων ύφους που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που επεξεργάζεται ή μεταγλωττίζεται σε Διαδοχικά Φύλλα Ύφους (CSS). Το SassScript είναι scripting language και αποτελείται από δύο είδη σύνταξης. Η αρχική σύνταξη, που ονομάζεται "οδοντωτή σύνταξη", χρησιμοποιεί μια σύνταξη παρόμοια με την HamI. Χρησιμοποιεί τον διαχωρισμό σε ξεχωριστά μπλοκ κώδικα και χαρακτήρες νέας γραμμής για διαχωρισμό κανόνων. Η νεότερη σύνταξη, "SCSS" (Sassy CSS), χρησιμοποιεί μορφοποίηση μπλοκ όπως αυτή του CSS. Χρησιμοποιεί braces για να υποδηλώσει μπλοκ κώδικα και ερωτηματικά για να διαχωρίσουν γραμμές μέσα σε ένα μπλοκ. Η

συντομογραφία και τα αρχεία SCSS δίνουν παραδοσιακά τις επεκτάσεις .sass και .scss, αντίστοιχα [10].

### **JAVASCRIPT:**

Η JavaScript είναι υψηλού επιπέδου διερμηνευμένη γλώσσα προγραμματισμού για διαδικτυακές και μη εφαρμογές. Αρχικά δημιουργήθηκε για εκτέλεση κώδικα σε ιστοσελίδες των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται. Σήμερα η JavaScript χρησιμοποιείται σε διάφορες εφαρμογές και εκτός ιστοσελίδων, όπως mobile apps, εφαρμογές ηλεκτρονικών υπολογιστών (desktop widgets) και τα τελευταία χρόνια οι νεότερες εικονικές μηχανές (όπως το Node.js) στα πλαίσια της ανάπτυξης και της εξέλιξης, έχουν κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών στην πλευρά διακομιστή (server-side) [11].

### **ANGULARJS:**

Το AngularJS είναι ένα framework ανοικτού κώδικα για την ανάπτυξη δυναμικών διαδικτυακών εφαρμογών. Επεκτείνει τη σύνταξη του HTML με νέες HTML επικέτες (tags), που δημιουργούνται μέσω των «οδηγιών» (directives). Υποστηρίζει αρχιτεκτονική MVC, όπου όψη (view) είναι το HTML template και ελεγκτής (controller) είναι ο AngularJS controller. Ο τελευταίος περιλαμβάνει τη λογική που σχετίζεται με την όψη και συνδέεται μ'αυτήν μέσω του αντικειμένου \$scope και της αμφίδρομης διασύνδεσης. Τέλος υπάρχουν οι υπηρεσίες (services), που περιλαμβάνουν τη λογική της εφαρμογής, η οποία είναι ανεξάρτητη από τις όψεις. Υλοποιούνται ως μοναδιαία στιγμιότυπα και δημιουργούνται αυτόματα μέσω της «έγχυσης εξάρτησης» (dependency injection) την πρώτη φορά που απαιτούνται από ένα άλλο τμήμα του κώδικα (ελεγκτή, υπηρεσία κλπ.). Επιπλέον όλα τα παραπάνω δομούνται σε αυτόνομες λειτουργικές μονάδες (modules) [12].

### **JSON:**

Το JSON (JavaScript Object Notation) είναι ένα ανοιχτό πρότυπο ανταλλαγής δεδομένων που βασίζεται σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript και χρησιμοποιεί κείμενο που είναι αναγνώσιμο από τον άνθρωπο. Είναι το κύριο πρότυπο ανταλλαγής δεδομένων που χρησιμοποιείται για την ασύγχρονη επικοινωνία περιηγητή / server και είναι εύκολα αναγνώσιμο από πολλές γλώσσες προγραμματισμού.[13]

### 2.3.3 Βιβλιοθήκες

Στην παρακάτω ενότητα θα δούμε τις βιβλιοθήκες που χρησιμοποιήθηκαν για της επίτευξη των στόχων της εφαρμογής και της ορθής λειτουργίας της.

#### **Βιβλιοθήκες που χρησιμοποιήθηκαν:**

- ✓ Leaflet

#### **LEAFLET:**

Η leaflet είναι μια βιβλιοθήκη ανοικτού κώδικα με διαδραστικούς χάρτες φιλική προς κινητές συσκευές. Έχει πολύ απλό design και λειτουργεί αποτελεσματικά σε όλες τις μεγάλες πλατφόρμες επιτραπέζιων και φορητών υπολογιστών, μπορεί να επεκταθεί με πολλά πρόσθετα διαθέτει ένα πολύ αναλυτικό documentation που κάνει την χρήση της βιβλιοθήκης πού εύκολη. Ένα άλλο πολύ μεγάλο πλεονέκτημα της βιβλιοθήκης ότι το μέγεθός της είναι μόλις 38KB [14].

### 2.3.4 Πρόσθετα

Για την πλήρη λειτουργικότητα της εφαρμογής είναι σχεδόν απαραίτητη η χρήση κάποιον plugins. Ως plugin, ή plug-in, ορίζεται ένα σύστημα συστατικών κάποιου λογισμικού που προσθέτει ιδιαίτερες δυνατότητες σε ένα μεγαλύτερο λογισμικό. Συγκεκριμένα στις cross-platforms υπάρχει μεγάλη ποικιλία σε πρόσθετα που το καθένα από αυτά παρέχει μεγάλη γάμα λειτουργιών και έχει συγκεκριμένους σκοπούς. Αυτός είναι και ο λόγος που παρά το γεγονός ότι τα πρόσθετα συνδέονται άμεσα με την πλατφόρμα το framework και πολλές φορές είναι απαραίτητα για την διασύνδεση με το hardware ή το λειτουργικό σύστημα της συσκευής δεν συμπεριλαμβάνονται στο αρχικό πακέτο κατά την εγκατάσταση. Οι δυνατότητες που παρέχουν οι σύγχρονες πλατφόρμες είναι αμέτρητες επομένως είναι αδύνατον να συμπεριληφθεί στο αρχικό πακέτο κάθε δυνατή λειτουργία, επίσης ο όγκος της πλατφόρμας θα ήταν πολύ μεγάλος, όπως επίσης σε περίπτωση που δεν χρησιμοποιούσαμε κάποιες λειτουργίες αυτές θα εξακολουθούσαν να υπάρχουν στο σύστημα επιβαρύνοντας έτσι και την εφαρμογή, αλλά γεμίζοντάς την και με άχρηστα αρχεία.

#### **Plugins που χρησιμοποιήθηκαν:**

- ✓ Cordova Plugin Geolocation
- ✓ Cordova Plugin Geofence
- ✓ Cordova Plugin Device
- ✓ Cordova SMS Plugin

Αναλυτικά για το καθένα από τα παραπάνω:

### **Cordova Plugin Geolocation:**

Αυτό το πρόσθετο παρέχει πληροφορίες σχετικά με τη θέση της συσκευής, όπως γεωγραφικό πλάτος και μήκος. Το πρόσθετο αναλαμβάνει την σύνδεση μεταξύ της υβριδικής εφαρμογής και της λειτουργίας GPS (Global Positioning System) του λειτουργικού συστήματος της κινητής συσκευής μέσω API το οποίο βασίζεται στο W3C Geolocation API Specification. Η πληροφορίες της τοποθεσίας μπορεί να προέρχονται από λειτουργία GPS του κινητού, πληροφορίες δικτύου (Internet) ή μέσω GSM/CDMA. Το πρόσθετο για την λειτουργία του χρησιμοποιεί τις παρακάτω μεθόδους [15]:

- navigator.geolocation.getCurrentPosition
- navigator.geolocation.watchPosition
- navigator.geolocation.clearWatch

Και υποστηρίζεται από τα παρακάτω λειτουργικά συστήματα:

- Android
- iOS
- Windows

### **Cordova Plugin Geofence:**

Αρχικά ως geofence ονομάζεται μια εικονική περίμετρος για μια πραγματική γεωγραφική περιοχή. Η χρήση ενός geofence ονομάζεται geofencing (geo-περίφραξη) η οποία μπορεί να πραγματοποιηθεί με την βοήθεια της συσκευής η οποία λαμβάνει γνώση της θέσης με βάση την τοποθεσία της.

Το πρόσθετο αυτό παρέχει τις μεθόδους για την υλοποίηση της παραπάνω λειτουργίας. Με την χρήση του API αναλαμβάνει την επικοινωνία μεταξύ της εφαρμογής και του λειτουργικού συστήματος της κινητής συσκευής για την παρακολούθηση μεταβολής θέσης του χρήστη μέσω της λειτουργίας GPS.

Η χρήση του plugin γίνεται με την αρχικοποίηση του plugin στο window.geofence object.

Το πρόσθετο για την λειτουργία του χρησιμοποιεί τις παρακάτω μεθόδους:

- window.geofence.initialize
- window.geofence.addOrUpdate
- window.geofence.remove
- window.geofence.removeAll
- window.geofence.getWatched

## Constants

- `TransitionType.ENTER = 1`
- `TransitionType.EXIT = 2`
- `TransitionType.BOTH = 3`

Και υποστηρίζεται από τα παρακάτω λειτουργικά συστήματα:

- Android
- iOS  $\geq 7.0$
- Windows  $\geq 8.1$

Επειδή όμως το πρόσθετο επικοινωνεί με ένα σύνολο των native API των συσκευών έρχεται με κάποιους περιορισμούς σε κάθε λειτουργικό. Συγκεκριμένα όπως φαίνεται κα παραπάνω είναι η έκδοση της πλατφόρμας καθώς επίσης για το λειτουργικό Android μπορούμε να ορίσουμε έως και 100 διαφορετικά geofences, ενώ σε IOS και Windows 20 και απεριόριστα geofences αντίστοιχα [16].

## **Cordova Plugin Device:**

Αυτό το πρόσθετο ορίζει το global device object (γενικό αντικείμενο συσκευής), το οποίο παρέχει πληροφορίες σχετικά με το υλικό και το λογισμικό της. Η εφαρμογή σε περίπτωση που πρόκειται να επικοινωνήσει με άλλες εφαρμογές, με το λειτουργικό σύστημα ή με το hardware της συσκευής πρέπει να έχει αυτό το πρόσθετο και μπορεί να λειτουργήσει παρά μόνο μετά το deviceready event [17].

Υποστηρίζεται από τα παρακάτω λειτουργικά συστήματα:

- Android
- Browser
- iOS
- Windows
- OSX

## **Cordova SMS Plugin:**

Το πρόσθετο αυτό χρησιμοποιείται για την προσθήκη στην εφαρμογή δυνατότητας αποστολής sms (μηνυμάτων). Επίσης το plugin δίνει την δυνατότητα αποστολής αυτοματοποιημένων μηνυμάτων μέσω native εφαρμογής ή μέσω default εφαρμογής που έχει ορίσει ο χρήστης [18].

Υποστηρίζεται από τα παρακάτω λειτουργικά συστήματα:

- Android
- iOS
- Windows 8 , 8.1 , 10 Universal.

### 2.3.5 Εργαλεία

Για τον έλεγχο ορθής λειτουργίας της εφαρμογής χρησιμοποιήθηκε μια άλλη εφαρμογή η οποία διατίθεται δωρεάν στο Google Play Store και ονομάζεται fake GPS. Η συγκεκριμένη εφαρμογή παρεμβαίνει στην λειτουργία του GPS του κινητού δίνοντας ψεύτικες συντεταγμένες τις οποίες μπορεί να ορίζει ίδιος ο χρήστης.

Η χρήση αυτής της εφαρμογής κρίθηκε απαραίτητη στα πλαίσια του testing καθώς οι αποστάσεις που ορίζονται στην εφαρμογή DoApp είναι σχετικά μεγάλες (1000 – 5000 μ.), για να μπορεί να γίνει το testing σε real time κατά την διάρκεια του προγραμματισμού η θέση του κινητού θα έπρεπε να αλλάζει συνεχώς ανάλογα με την ακτίνα που έχει οριστεί στην εφαρμογή, κάτι το οποίο απαιτούσε χρήση κάποιου εργαλείου.

Για την σχεδίαση των εικόνων και του λογοτύπου της εφαρμογής, χρησιμοποιήθηκε το Adobe Photoshop και το Adobe Illustrator.

Ως text editors χρησιμοποιήθηκαν Sublime και Atom.

## 2.4 Η υποστηρικτική ιστοσελίδα DoApp

Όπως προαναφέρθηκε, η ομάδα-στόχος της εφαρμογής είναι άτομα τα οποία δεν έχουν ιδιαίτερα καλή σχέση με τις νέες τεχνολογίες και τις κινητές συσκευές, άρα σαν πρώτο μέλημα είχα την λειτουργικότητα της εφαρμογής.

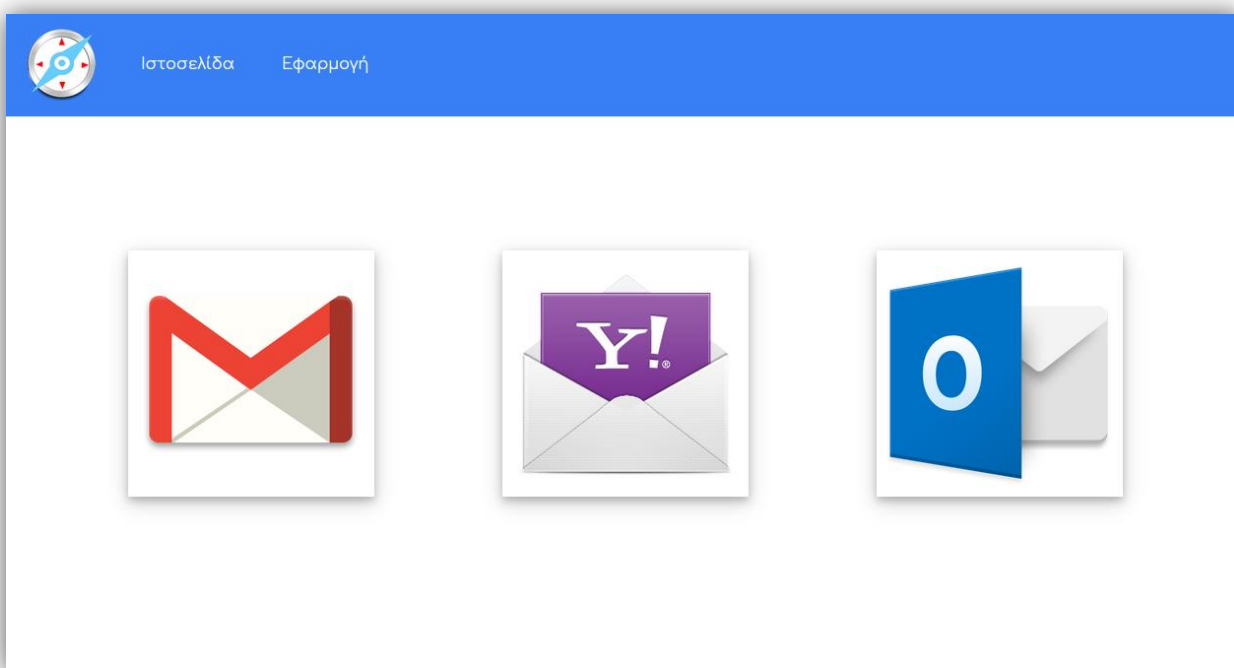
Με την ιστοσελίδα DoApp αυτό που επιτυγχάνεται είναι:

Ο χρήστης μπορεί να γνωρίσει την εφαρμογή και την χρησιμότητά της - μέσω των περιγραφών (descriptions) που δίνονται - σε μεγαλύτερες οθόνες (σταθερών ή φορητών υπολογιστών), όπου η ανάγνωσή τους θα γίνεται πιο εύκολα. Επίσης θα μπορεί να βλέπει τους αναλυτικούς οδηγούς χρήσης (tutorials) χωρίς να επιβαρύνεται η μνήμη της κινητής συσκευής του.

Ο κυριότερος όμως στόχος της δημιουργίας της ιστοσελίδας δεν ήταν μόνο η υποστήριξη της εφαρμογής αλλά η συγκέντρωση άλλων βοηθητικών λειτουργιών, ιστοσελίδων που αφενός μπορούν να βοηθήσουν τον χρήστη στην καθημερινότητά του αφετέρου χάρη σε απλό και φιλικό προς τον χρήστη σχεδιασμό να τον βοηθήσουν να εξοικειωθεί με το διαδίκτυο αξιοποιώντας μια μεγάλη γάμα ιστοσελίδων που μπορούν να χρησιμοποιηθούν ως επιμόρφωση, ενημέρωση, επικοινωνία ή διασκέδαση.

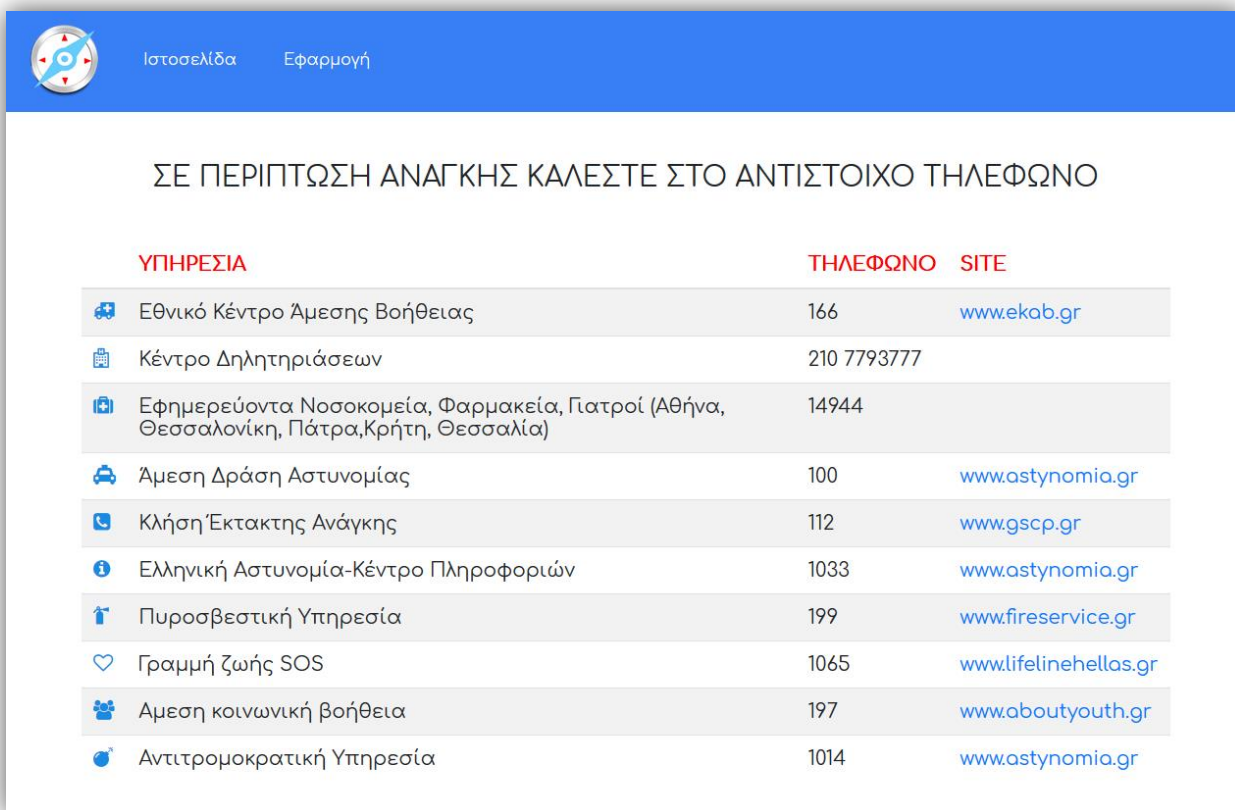
Περιλαμβάνει έξι (6) βασικές λειτουργίες, οι οποίες υποβοηθούν τους χρήστες να μεταβαίνουν άμεσα :

- ✓ στη σελίδα του ηλεκτρονικού τους ταχυδρομείου (εικόνα 14)
- ✓ σε ιστότοπους υπηρεσιών άμεσης ανάγκης (ΕΚΑΒ, Αστυνομία, Πυροσβεστική κ.λπ.) και να βλέπουν το αντίστοιχο αριθμό κλήσης (εικόνα 15)
- ✓ σε ιστότοπους διαδικτυακής ενημέρωσης (εικόνα 16)
- ✓ σε ιστότοπους διαδικτυακής ψυχαγωγίας (εικόνα 17)
- ✓ σε ημερολόγιο, αριθμομηχανή, στον ιστότοπο της Google Map και σε μετατροπέα μονάδων (εικόνα 18)
- ✓ σε διάφορους άλλους χρήσιμους ιστότοπους (εικόνα 19)

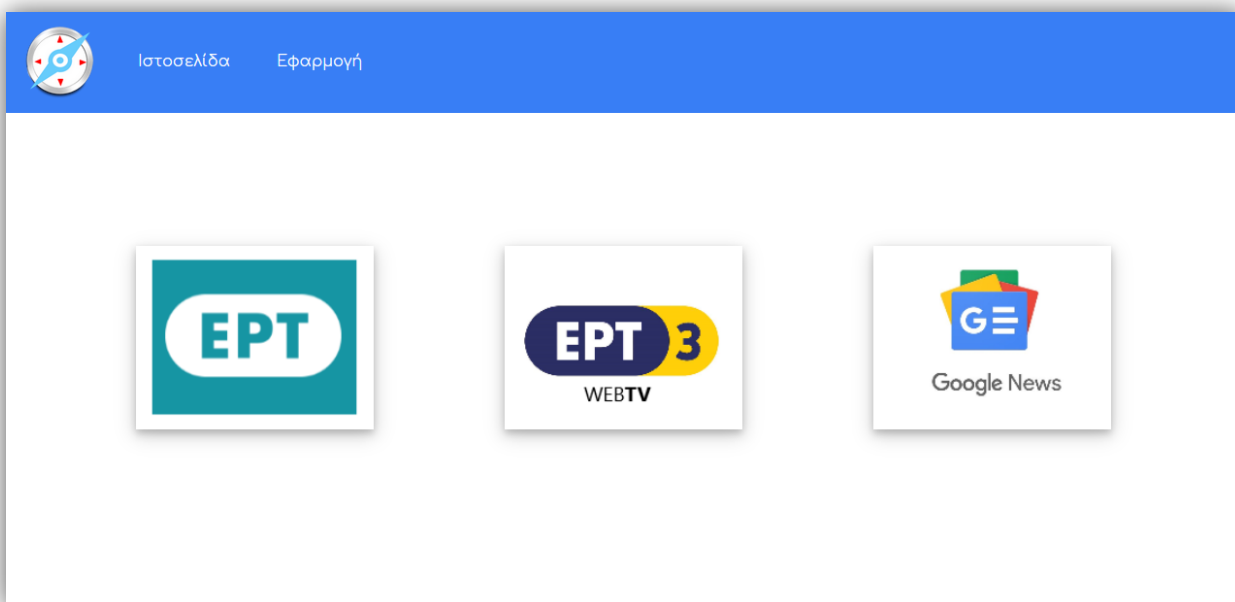


Εικόνα 14



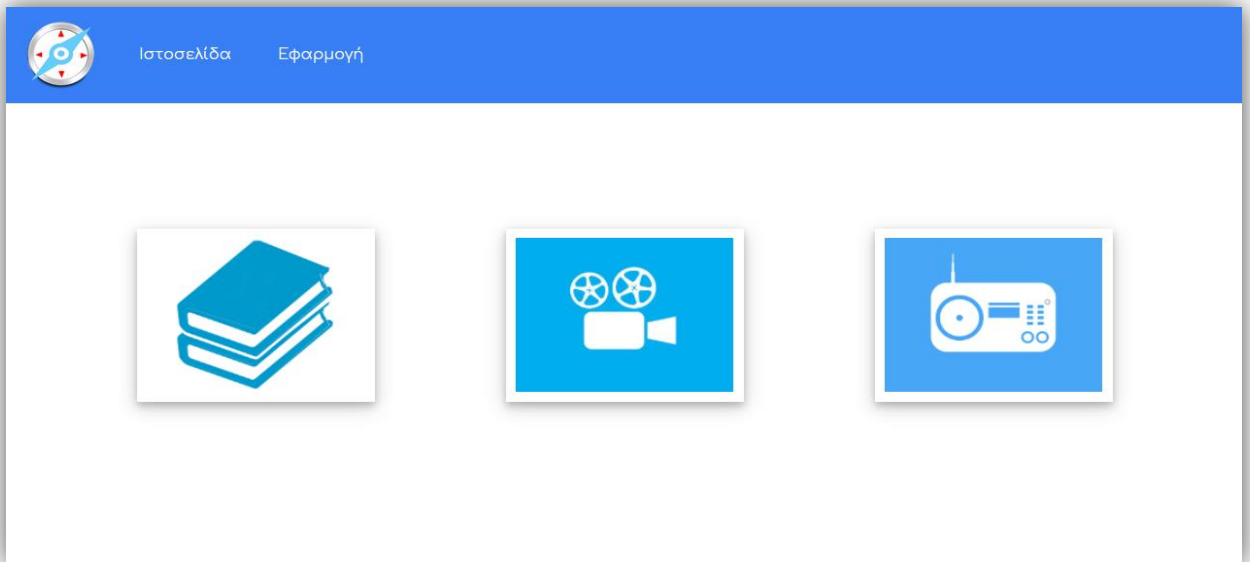


Εικόνα 15

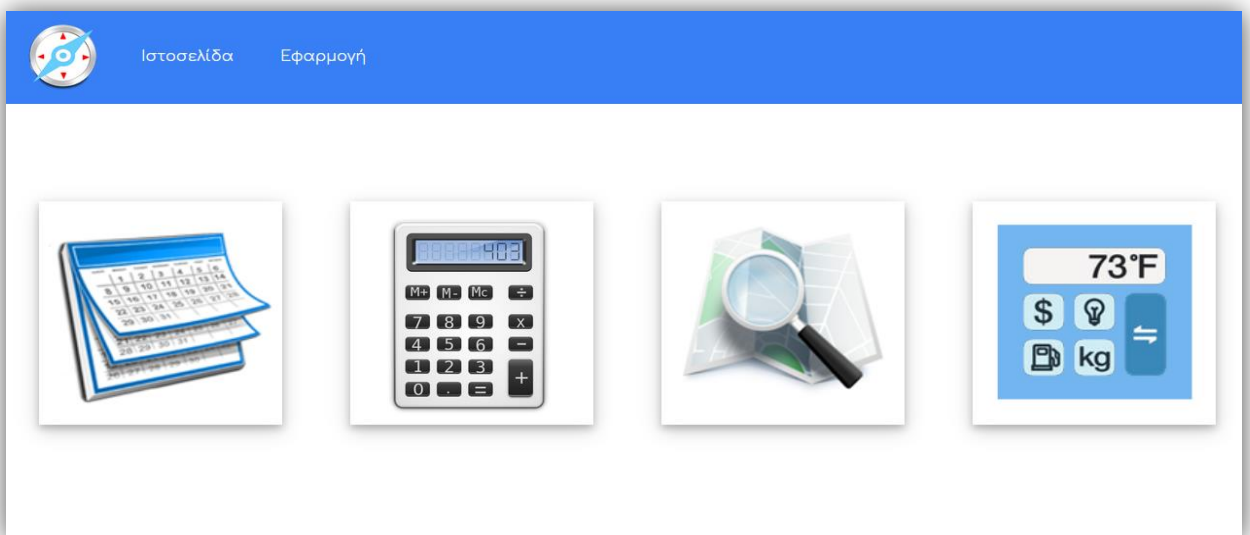


Εικόνα 16

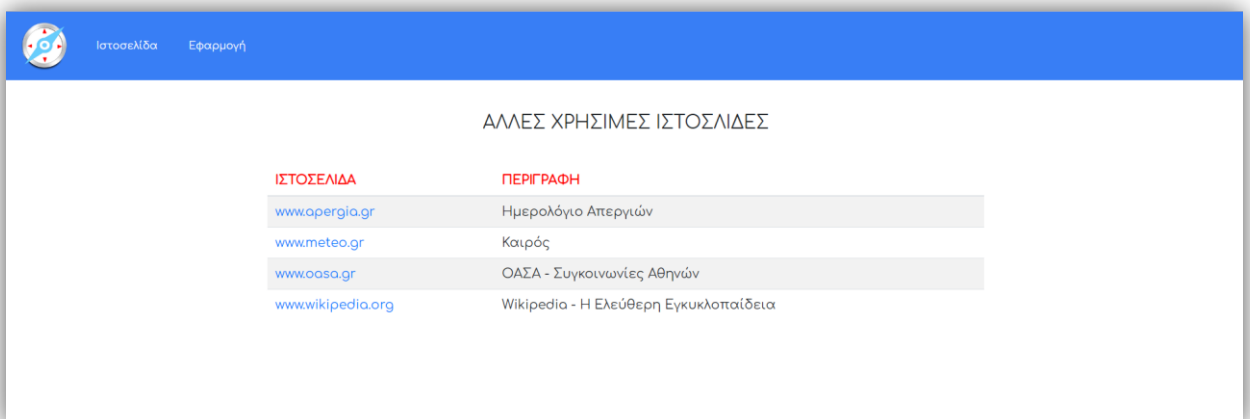
## Εφαρμογή Android για άτομα που χρήζουν βοήθειας



Εικόνα 17



Εικόνα 18



Εικόνα 19  
ΟΝΙΝΑ ΝΤΙΑΝΑ

Επίσης πέρα από τα παραπάνω μενού η ιστοσελίδα διαθέτει σελίδες όπως

- Περιγραφή ιστοσελίδας
- Βοήθεια ιστοσελίδας
- Περιγραφή εφαρμογής
- Παρουσίαση εφαρμογής
- Όροι χρήσης ιστοσελίδας και εφαρμογής αντίστοιχα

## 2.5 Ανάπτυξη της ιστοσελίδας

Η ανάπτυξη της ιστοσελίδας υλοποιήθηκε με τις γλώσσες σήμανσης και προγραμματισμού όπως HTML5, CSS3, JAVASCRIPT και PHP ενώ για την ολοκλήρωσή της χρησιμοποιήθηκαν και άλλα εργαλεία (tools) και βιβλιοθήκες (libraries) όπως θα δούμε στις επόμενες παραγράφους του κεφαλαίου.

### 2.5.1 Γλώσσες Προγραμματισμού & Εργαλεία ανάπτυξης

Για την ανάπτυξη της ιστοσελίδας χρησιμοποιήθηκαν οι παρακάτω γλώσσες:

**Για την σήμανση και εμφάνιση:**

- ✓ HTML5
- ✓ CSS3

**Για τον προγραμματισμό και την λειτουργικότητα:**

- ✓ JAVASCRIPT
- ✓ PHP

Η ιστοσελίδα δεν διαθέτει κάποιες εξειδικευμένες λειτουργίες όπως αναφέρθηκε και πιο πάνω ο βασικός σκοπός της ήταν η εξοικείωση του χρήστη με το διαδίκτυο και η υποστήριξη της εφαρμογής DoApp.

### 2.5.2 Βιβλιοθήκες και Εργαλεία

Για την επίτευξη των στόχων της εφαρμογής όπως η υλοποίηση βοηθητικών εφαρμογών,

Οι βιβλιοθήκες που χρησιμοποιήθηκαν για την κατασκευή της ιστοσελίδας:

- ✓ BOOTSTRAP
- ✓ JQUERY

**BOOTSTRAP:**

Το Bootstrap είναι ένα CSS Framework ανοιχτού κώδικα για την ανάπτυξη ιστοσελίδων και διαδικτυακών εφαρμογών με HTML, CSS και JS. Το Bootstrap περιέχει πρότυπα βασισμένα σε HTML και CSS για φόρμες, κουμπιά, πίνακες, πλοήγηση, εικόνες κ.α καθώς και κάποια plugins JavaScript. Επίσης ένα από τα βασικά πλεονεκτήματα του Bootstrap που το έκανε τόσο δημοφιλή είναι η δυνατότητα εύκολης και πιο γρήγορης ανάπτυξης ιστοσελίδων με βάση τα πρότυπα του Responsive Design [19].

Responsive design (Συμβατός σχεδιασμός ή αλλιώς Αναποκρίσημος σχεδιασμός) σχεδιασμός ιστοσελίδων αφορά στη δημιουργία ιστότοπων οι οποίοι προσαρμόζονται στις διαστάσεις τις οθόνης της κάθε συσκευής ώστε να έχουν τη σωστή προβολή (παρουσίαση περιεχομένου) σε όλα τα μεγέθη των οθονών, από μικρά τηλέφωνα έως μεγάλους επιτραπέζιους υπολογιστές.

### **JQUERY:**

jQuery είναι μια δωρεάν βιβλιοθήκη της JavaScript ανοιχτού κώδικα. Η jQuery λαμβάνει εργασίες που απαιτούν πολλές γραμμές κώδικα JavaScript για να υλοποιηθεί κάποια λειτουργία και τις αναδιπλώνει σε μεθόδους που σε πολλές περιπτώσεις καλούνται με μία μόνο γραμμή κώδικα [20].

Η jQuery περιέχει τις ακόλουθες λειτουργίες:

- HTML/DOM χειρισμός
- CSS χειρισμός
- HTML event μέθοδοι
- Effects and animations

Για την επεξεργασία και σχεδίαση των εικόνων που χρησιμοποιήθηκαν στην ιστοσελίδα, χρησιμοποιήθηκε το Adobe Photoshop και το Adobe Illustrator.

Ως text editors χρησιμοποιήθηκαν Sublime και Atom.

## ΚΕΦΑΛΑΙΟ 3<sup>ο</sup>: Ανάπτυξη της εφαρμογής DoApp

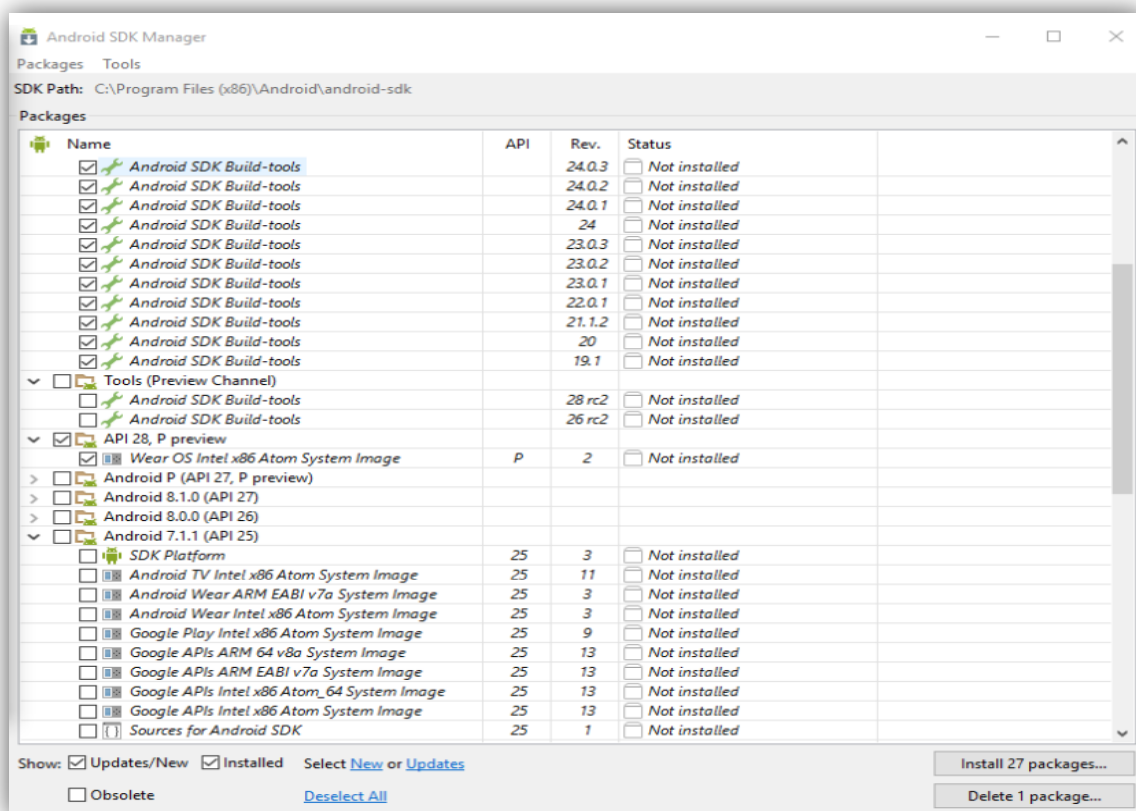
Σε αυτό το κεφάλαιο θα γίνει αναφορά για τον τρόπο ανάπτυξης της εφαρμογής και αναλυτική περιγραφή του πηγαίου κώδικα.

### 3.1 Η Εγκατάσταση των απαραίτητων προγραμμάτων

Στην παρακάτω ενότητα παρατίθενται όλα τα προγράμματα που είναι απαραίτητα για την δημιουργία της εφαρμογής καθώς και ο τρόπος της εγκατάστασης τους. Η εγκατάσταση προγραμμάτων γίνεται με την σειρά που αναγράφονται:

- ✓ Java SE
- ✓ Android SDK tools – σημείωση: μετά την εγκατάσταση είναι σημαντικό να προσθέσουμε το path που εγκαταστήσαμε τα εργαλεία στο PATH του συστήματος μας έτσι ώστε να μπορούμε να κάνουμε χρήση αυτών μέσω της γραμμής εντολών (εικόνα 20)
- ✓ Node js
- ✓ Ανοίγοντας την γραμμή εντολών βάζουμε την παρακάτω εντολή η οποία θα κάνει την εγκατάσταση του Ionic Framework – Apache Cordova

*npm install -g cordova ionic.*



Εικόνα 20

Για την δημιουργία της δομής της εφαρμογής χρησιμοποιείται η ακόλουθη διαδικασία:

1) Ανοίγουμε την γραμμή εντολών και κατευθυνόμαστε στην τοποθεσία όπου θέλουμε να αποθηκεύσουμε την εφαρμογή μας.

2) Στη συνέχεια μέσω ηrm δημιουργούμε την εφαρμογή με χρήση της παρακάτω παρακάτω εντολής - *ionic start DoApp blank* - η παραπάνω εντολή θα δημιουργήσει μια βασική εφαρμογή με το default πακέτο λειτουργιών στην τοποθεσία που βρισκόμαστε με όνομα DoApp. Το Ionic Framework παρέχει 3 βασικά pre-made εικαστικά θέματα: tabs starter, sidemenu starter και το blank starter που είναι το πιο «άδειο» και συνήθως χρησιμοποιείται για δημιουργία custom template. Στη παρούσα πτυχιακή χρησιμοποιήθηκε το template – blank starter όπως φαίνεται και στην παραπάνω εντολή. Στη συνέχεια θα πρέπει να επιλεγθεί και να εγκατασταθεί το λειτουργικό σύστημα (ή τα λειτουργικά συστήματα) με το οποίο/α θα είναι συμβατή η εφαρμογή. Αυτό γίνεται με την εντολή - *ionic platform add android* ή για το ios αντίστοιχα *ionic platform add ios*. Στη παρούσα πτυχιακή η εφαρμογή δημιουργήθηκε για android OS.

3) Για να εκτελέσουμε την εφαρμογή στο OS που επιλέξαμε, πληκτρολογούμε την εντολή - *ionic build android*, με την παραπάνω εντολή παράγεται το apk αρχείο το οποίο εγκαθίσταται στην συσκευή περνώντας το αρχείο μέσω ενός καλωδίου usb. Επίσης υπάρχει η δυνατότητα να «τρέξουμε» την εφαρμογή απευθείας στην κινητή συσκευή εφόσον είναι συνδεδεμένη με τον υπολογιστή και αφού έχει ενεργοποιηθεί το usb – debugging από τα developers tools της κινητής συσκευής, αυτό γίνεται πληκτρολογώντας την εντολή - *ionic run android*. Η εντολή αυτή εφόσον εκτελεστεί με επιτυχία θα εγκαταστήσει αυτόματα την εφαρμογή στην συνδεδεμένη συσκευή και στη συνέχεια θα την ανοίξει (εικόνα 21).



```
Γραμμή εντολών
:assembleDebug UP-TO-DATE
:cdvBuildDebug UP-TO-DATE
BUILD SUCCESSFUL
Total time: 4.778 secs
Built the following apk(s):
  C:/Users/user/DoApp/platforms/android/build/outputs/apk/android-debug.apk

ANDROID_HOME=D:\Android-SDK
JAVA_HOME=C:\Program Files\Java\jdk1.8.0_121
No target specified, deploying to device '908501167916'.

Skipping build...
Built the following apk(s):
  C:/Users/user/DoApp/platforms/android/build/outputs/apk/android-debug.apk
Using apk: C:/Users/user/DoApp/platforms/android/build/outputs/apk/android-debug.apk
Package name: com.ionicframework.doapp288849
LAUNCH SUCCESS
C:\Users\user\DoApp>
```

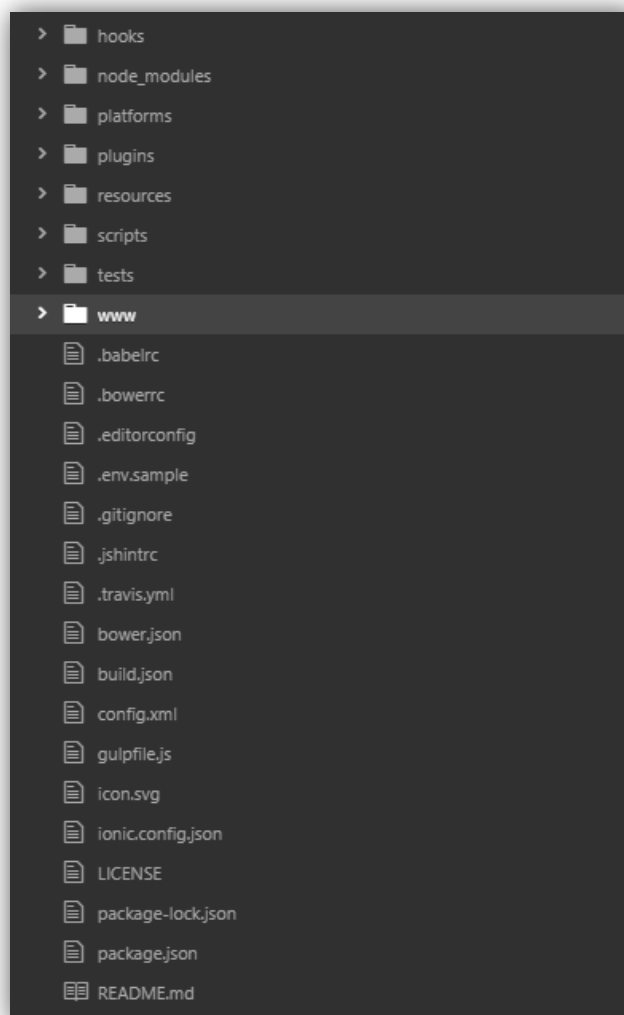
Εικόνα 21

Κατά την διάρκεια σχεδιασμού της συσκευής (κυρίως της όψης – view) υπάρχει και η δυνατότητα εκτέλεσης εφαρμογής στον browser, αυτό γίνεται με την παρακάτω εντολή – *ionic serve*. Η πιο πρόσφατες εκδόσεις του Ionic Framework επίσης έχουν και την δυνατότητα εκτέλεσης της εφαρμογής σε εικονικό περιβάλλον των τριών βασικών OS – Android, iOS και Windows με την εντολή - *ionic serve –lab*. Η εκτέλεση της εφαρμογής στο browser δεν συνιστάται στις περιπτώσεις που πρέπει να δοκιμαστούν οι λειτουργίες της εφαρμογής καθώς δεν γίνεται το emulating των λειτουργιών των λειτουργικών συστημάτων αλλά μόνο η εμφάνιση σε αυτά.

### 3.2 Η δομή των αρχείων και του κώδικα της εφαρμογής

Στην ενότητα αυτή θα αναλυθεί η δομή της εφαρμογής και θα γίνουν αναφορές στο πηγαίο κώδικά της.

Στην παρακάτω δομή φαίνεται η δομή της εφαρμογής (εικόνα 22):



Εικόνα 22

Η αναφορά θα γίνει κυρίως στους φακέλους:

- ✓ platforms
- ✓ plugins
- ✓ resources
- ✓ www

Περισσότερη ανάλυση θα γίνει στον φάκελο *www* – ο οποίος περιέχει αρχεία τα οποία είναι βασικά για την εμφάνιση και την λειτουργικότητα της εφαρμογής.

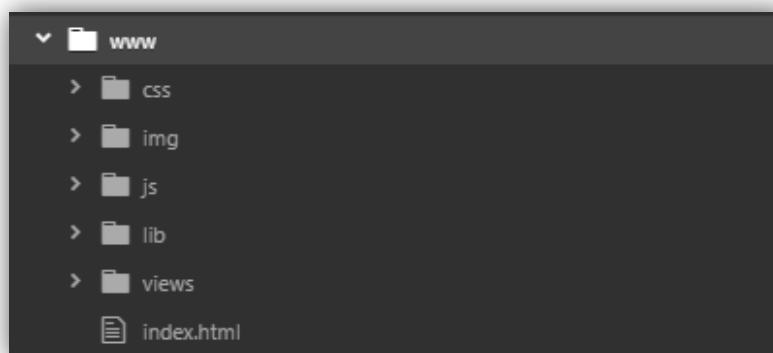
Στον φάκελο *platforms* προστίθενται τα λειτουργικά συστήματα τα οποία είναι συμβατά με την εφαρμογή. Σε κάθε φάκελο υπάρχουν όλα τα απαραίτητα αρχεία για την εγκατάσταση και την εκτέλεση της εφαρμογής στο εκάστοτε λειτουργικό σύστημα.

Στον φάκελο *plugins* υπάρχουν όλα τα plugins που χρησιμοποιούνται για την λειτουργία ή εμφάνιση της εφαρμογής.

Στον φάκελο *resources* υπάρχουν όλες οι εικόνες απαραίτητες για την εφαρμογή όπως π.χ τα app icons, splash screen κ.α.

Ο φάκελος *www* περιλαμβάνει και άλλους φακέλους που οι οποίοι περιέχουν ομαδοποιημένα τα αρχεία που συνθέτουν την λειτουργικότητα και την εμφάνιση της εφαρμογής.

Η δομή του φακέλου *www* (εικόνα 23):



Εικόνα 23

- ✓ css – αρχεία styles της εφαρμογής
- ✓ img – οι εικόνες που συμπεριλαμβάνονται μέσα στο view – όψη της εφαρμογής
- ✓ js – αρχεία javascript της εφαρμογής
- ✓ lib – βιβλιοθήκες που χρησιμοποιούνται στην εφαρμογή



- ✓ views – βρίσκονται τα html templates των διαφορετικών όψεων με τις οποίες θα έρθει σε επαφή ο χρήστης, καθώς επίσης και τα αντίστοιχα αρχεία js με τη λογική / λειτουργικότητα της κάθε όψης.
- ✓ Index.html – το αρχικό αρχείο της εφαρμογής μέσα στο οποίο καλούμε και ενσωματώνουμε όλες τις απαραίτητες βιβλιοθήκες και αρχεία. Επίσης στο αρχείο αυτό κάνουμε bind τον controller (AngularJS), ο οποίος είναι υπεύθυνος για τις λειτουργίες της εφαρμογής (ng-app="ionic-geofence")

### index.html:

```
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no, width=device-width">
  <meta http-equiv="Content-Security-Policy" content="default-src 'self' 'unsafe-eval' 'unsafe-inline' ws: gap://ready localhost:*; style-src 'self' 'unsafe-inline'; img-src 'self' data: https://*.mapbox.com https://*.openstreetmap.org;">
  <title>DoApp</title>
  <link rel="manifest" href="manifest.json">
  <link href="lib/ionic/css/ionic.css" rel="stylesheet">
  <link href="css/style.css" rel="stylesheet">
  <link rel="stylesheet" href="lib/leaflet/dist/leaflet.css">
  <!-- ionic/angularjs js -->
  <script src="lib/ionic/js/ionic.bundle.js"></script>
  <!-- cordova script (this will be a 404 during development) -->
  <script src="js/ng-cordova.min.js"></script>
  <script src="cordova.js"></script>
  <!-- app's js -->
  <script src="js/app.js"></script>
  <script src="js/controllers.js"></script>
  <script src="js/routes.js"></script>
  <script src="js/controllers/geofence.js"></script>
  <script src="js/controllers/geofences.js"></script>
  <script src="js/controllers/quickphones.js"></script>
  <script src="js/services/geofence-plugin-mock.js"></script>
  <script src="js/services/geofence.js"></script>
  <script src="js/services/geolocation.js"></script>
</head>
<body ng-app="ionic-geofence">
</body>
```

Εικόνα 24

Όπως βλέπουμε παραπάνω σε κανένα σημείο δεν καλείται κάποιο view αρχείο της εφαρμογής, αυτό δηλαδή με το οποίο έρχεται σε επαφή ο χρήστης, αλλά ούτε και στο ίδιο αρχείο υπάρχει κάποιο στοιχείο με το οποίο μπορεί να αλληλεπιδράσει ο χρήστης.

Όλη η δομή της εφαρμογής και τα αντίστοιχα templates που θα βλέπει ο χρήστης μαζί με τους υπεύθυνους controllers καλούνται στο αρχείο js/routes.js

**routes.js:**

```

1  angular.module("ionic-geofence").config(function ($stateProvider, $urlRouterProvider) {
2      $stateProvider
3          .state("geofences", {
4              url: "/geofences",
5              templateUrl: "views/geofences.html",
6              controller: "GeofencesCtrl"
7          })
8          .state("geofence-new", {
9              url: "/geofence/new/:longitude,:latitude",
10             templateUrl: "views/geofence.html",
11             controller: "GeofenceCtrl",
12
13             resolve: {
14                 geofence: function ($stateParams, Geofence) {
15                     return Geofence.create({
16                         longitude: parseFloat($stateParams.longitude),
17                         latitude: parseFloat($stateParams.latitude)
18                     });
19                 }
20             }
21         })
22         .state("extra", {
23             url: "/extra",
24             templateUrl: "views/extra.html",
25             controller: "GeofencesCtrl"
26         })
27         .state("phones", {
28             url: "/phones",
29             templateUrl: "views/phones.html",
30             controller: "GeofencesCtrl"
31         })
32         .state("quickphones", {
33             url: "/quickphones",
34             templateUrl: "views/quickphones.html",
35             controller: "QuickCall"
36         })
37         .state("descriptions", {
38             url: "/descriptions",
39             templateUrl: "views/descriptions.html",
40         })
41         .state("geofence-edit", {
42             url: "geofence/:geofenceId",
43             templateUrl: "views/geofence.html",
44             controller: "GeofenceCtrl",
45
46             resolve: {
47                 geofence: function ($stateParams, Geofence, $q) {
48                     var geofence = Geofence.findById($stateParams.geofenceId);
49
50                     if (geofence) {
51                         return $q.when(angular.copy(geofence));
52                     }
53
54                     return $q.reject("Δεν βρέθηκε η ειδοποίηση: " + $stateParams.geofenceId);
55                 }
56             }
57         });
58
59     $urlRouterProvider.otherwise("/geofences");
60 });

```

Εικόνα 25

Έτσι λοιπόν ο χρήστης έρχεται σε επαφή με το πρώτο template της εφαρμογής και στη συνέχεια ανάλογα με την πλοήγησή του στην εφαρμογή το αρχείο routes.js του σερβίρει τα αντίστοιχα αρχεία.

Το πρώτο template με το οποίο έρχεται σε επαφή ο χρήστης είναι το geofences.html

**geofences.html:**

```

1 <ion-view>
2 <ion-header-bar class="bar-positive" align-title="left">
3 <h1 class="title">DoApp</h1>
4 <div class="buttons pull-right">
5 <button class="button pull-right button-icon icon ion-location" ng-click="createNew()"></button>
6 <button class="button pull-right button-icon icon ion-navicon" ng-click="more()"></button>
7 </div>
8 </ion-header-bar>
9 <ion-content>
10 <div class="item item-input-inset">
11 <label class="item item-input-wrapper">
12 <i class="icon ion-ios-telephone placeholder-icon"></i>
13 <input type="text" ng-model="telephone.phone" placeholder="Τηλέφωνο">
14 </label>
15 </div>
16 <div class="item save_btn">
17 <button class="button bold button-calm button-full" ng-click="SavePhone()">
18 ΑΠΟΘΗΚΕΥΣΗ
19 </button>
20 </div>
21 <ion-list can-swipe="true">
22 <ion-item ng-click="editGeofence(geofence)" class="item item-icon-right" ng-repeat="geofence in geofences">
23 <p>{{geofence.notification.text}}</p>
24 <i class="icon ion-trash-b" ion-stop-event="click" ng-click="removeGeofence(geofence)"></i>
25 </ion-item>
26 </ion-list>
27 <div class="card" ng-if="geofences.length == 0">
28 <div class="item item-text-wrap">
29 Δεν έχετε καμία ειδοποίηση ακόμα.
30 </div>
31 </div>
32 </ion-content>
33 <ion-footer-bar align-title="left" class="bar-positive">
34 <div class="row">
35 <div class="col footer_col text-center"><a class="a_links" href="#/descriptions"><i class="icon ion-help"></i> Βοήθεια</a></div>
36 <div class="col footer_col text-center"><a class="a_links" href="#/extra"><i class="icon ion-checkmark"></i> Περισσότερα</a></div>
37 </div>
38 </ion-footer-bar>
39 </ion-view>
40

```

Εικόνα 26

Εδώ ο χρήστης καλείται να δηλώσει το κινητό τηλέφωνο του «κηδεμόνα», να δημιουργήσει ένα νέο σημείο αναφοράς, να διαγράψει όλα ή κάποια τα είδη υπάρχοντα σημεία αναφοράς ή να προηγηθεί στις επιπλέον λειτουργίες που διαθέτει η εφαρμογή καθώς επίσης μπορεί να ανατρέξει στην καρτέλα βοήθειας ή οποία παρέχει οδηγίες χρήσης εφαρμογής και περιγραφή λειτουργιών.

Ιδιαίτερο ενδιαφέρον έχει η σελίδα δημιουργίας του σημείου αναφοράς της οποίας η προβολή και οι λειτουργίες αξίζουν περεταίρω ανάλυση. Το template της δημιουργίας σημείου αναφοράς είναι το geofence.html

**geofence.html:**

```

1 <ion-view title="Geofence details">
2 <ion-header-bar class="bar-positive" align-title="left">
3 <div class="buttons pull-left">
4 <button class="button button-icon" ui-sref="geofences">
5 <i class="ion-chevron-left"></i>
6 </button>
7 </div>
8 <h1 class="title">Ειδοποίηση</h1>
9 <div class="buttons pull-right">
10 <button class="button button-icon icon ion-checkmark" ng-click="save()"></button>
11 </div>
12 </ion-header-bar>
13 <ion-content scroll="false">
14 <label class="item item-input">
15 <input type="text" placeholder="Εισάγετε κείμενο ειδοποίησης" ng-model="geofence.notification.text"
16 hide-keyboard-on-enter>
17 </label>
18 <label class="item">
19 <p class="wrapp_text">Επιλέξτε πότε θέλετε να λαμβάνεται την ειδοποίηση σε σχέση με την περιοχή
20 που επιλέξατε</p>
21 </label>
22 <div class="button-bar">
23 <a class="button btn_area" ng-class="isTransitionOfType(TransitionType.ENTER) ? 'button-positive'
24 : 'button-light'" ng-click="toggleWhenIgetCloser()"><span class="wrapp_text">Είσοδος</span></a>
25 <a class="button btn_area" ng-class="isTransitionOfType(TransitionType.EXIT) ? 'button-positive' :
26 'button-light'" ng-click="toggleWhenIamLeaving()"><span class="wrapp_text">Εξοδος</span></a>
27 </div>
28 <div class="range" data-tap-disabled="true">
29 στα
30 <input type="range" name="volume" min="50" max="5000" ng-model="paths.circle.radius">
31 {{ paths.circle.radius }}m
32 </div>
33 <div style="width: 100%; position: absolute; top: 240px; bottom: 0px;" data-tap-disabled="true">
34 <leaflet resizeable markers="markers" paths="paths" center="center" tiles="tiles" width="100%"
35 height="100%" event-broadcast="events"></leaflet>
36 </div>
37 </ion-content>
38 </ion-view>

```

Εικόνα 27

Σε αυτό το σημείο έχει ενδιαφέρον να δούμε και τους αντίστοιχους controllers οι οποίοι είναι υπεύθυνοι για τις λειτουργίες που παρέχει το κάθε template, έτσι όπως είδαμε στο αρχείο routes.js ο υπεύθυνος controller είναι ο “GeofenceCtrl”

```

8 .state("geofence-new", {
9   url: "/geofence/new/:longitude,:latitude",
10  templateUrl: "views/geofence.html",
11  controller: "GeofenceCtrl",
12

```

Εικόνα 28

Ο controller αυτός βρίσκεται στο path: www/js/controllers/geofence.js και το αρχείο αυτό είχε γίνει include στο index.html

Έτσι το αρχείο geofence.js έχει τον controller “GeofenceCtrl” ο οποίος είναι υπεύθυνος για όλες τις λειτουργίες στο συγκεκριμένο template.

**geofence.js:**

```
angular.module("ionic-geofence").controller("GeofenceCtrl", function (
    $scope,
    $ionicLoading,
    $state,
    geofence,
    Geofence
) {
    $scope.geofence = geofence;
    $scope.TransitionType = TransitionType;

    $scope.center = {
        lat: geofence.latitude,
        lng: geofence.longitude,
        zoom: 10
    };
    $scope.markers = {
        marker: {
            draggable: true,
            message: geofence.notification.text,
            lat: geofence.latitude,
            lng: geofence.longitude,
            icon: {}
        }
    };
    $scope.paths = {
        circle: {
            type: "circle",
            radius: geofence.radius,
            latlngs: $scope.markers.marker,
            clickable: false
        }
    };
});
```

Εικόνα 29

Εδώ βλέπουμε την εμφάνιση και την προβολή του χάρτη.

Στη συνέχεια βλέπουμε τον κώδικα που αφορά τα διαθέσιμα Transition Types (από Geofence Plugin η ανάλυση των οποίων θα γίνει παρακάτω) και την αποθήκευση των δεδομένων που εισάγει ο χρήστης. Επίσης σχεδόν σε κάθε λειτουργία της εφαρμογής υπάρχουν και οι αντίστοιχοι έλεγχοι που γίνονται είτε για την διευκόλυνση του χρήστη, είτε για την ορθή λειτουργία της εφαρμογής π.χ εδώ στην *function validate ()* φαίνεται το μήνυμα που εμφανίζει στον χρήστη σε περίπτωση που αποθηκεύει το σημείο αναφοράς χωρίς να προσθέσει το μήνυμα ειδοποίησης.

```

$scope.isWhenGettingCloser = function () {
    return $scope.geofence.transitionType === TransitionType.ENTER;
};

$scope.toggleWhenIgetCloser = function () {
    $scope.geofence.transitionType ^= TransitionType.ENTER;
};

$scope.toggleWhenIamLeaving = function () {
    $scope.geofence.transitionType ^= TransitionType.EXIT;
};

$scope.save = function () {
    if (validate()) {
        $scope.geofence.radius = parseInt($scope.paths.circle.radius);
        $scope.geofence.latitude = $scope.markers.marker.lat;
        $scope.geofence.longitude = $scope.markers.marker.lng;
        $scope.geofence.notification.data = angular.copy($scope.geofence);

        Geofence.addOrUpdate($scope.geofence).then(function () {
            $state.go("geofences");
        }, function (error) {
            $ionicLoading.show({
                template: "Αποτυχία προσθήκης τοποθεσίας, βεβαιωθείτε ότι η λειτουργία πρόσβασης στην
                τοποθεσία είναι ενεργοποιημένη",
                duration: 3000
            });
            console.log("Αποτυχία προσθήκης", error);
        });
    }
};

function validate () {
    if (!$scope.geofence.notification.text) {
        $ionicLoading.show({
            template: "Παρακαλώ εισάγετε κείμενο ειδοποίησης.",
            duration: 3000
        });
        return false;
    }
}

```

Εικόνα 30

Μετά την επιτυχή αποθήκευση του σημείου αναφοράς ο χρήστης επιστρέφει στην αρχική οθόνη της εφαρμογής το template της οποίας αναλύθηκε πιο πάνω.

Ο αντίστοιχος controller του συγκεκριμένου template βρίσκεται στο path: `www/js/controllers/geofences.js` και το αρχείο αυτό είχε γίνει include στο `index.html`

Έτσι το αρχείο `geofences.js` έχει τον controller “GeofencesCtrl” ο οποίος είναι υπεύθυνος για όλες τις λειτουργίες στο συγκεκριμένο template.

**geofences.js:**

```
angular.module("ionic-geofence").controller("GeofencesCtrl", function (
    $scope,
    $ionicActionSheet,
    $timeout,
    $log,
    $state,
    Geolocation,
    Geofence,
    $ionicLoading
) {
    /* TELEPHONE */
    $scope.telephone = {phone: ''};
    /* TELEPHONE */
    $scope.geofences = [];

    Geofence.getAll().then(function (geofences) {
        $ionicLoading.hide();
        $scope.geofences = geofences;
    }, function (reason) {
        $ionicLoading.hide();
        $log.error("Παρουσιάστηκε σφάλμα", reason);
    });
});
```

Εικόνα 31

Εδώ βλέπουμε την δήλωση του τηλεφώνου του κηδεμόνα και την προβολή όλων των σημείων αναφοράς που μπορεί να έχει δημιουργήσει ο χρήστης.

Στη συνέχεια έχουμε την function που είναι υπεύθυνη για την δημιουργία του καινούργιου σημείου αναφοράς, η οποία καλείται εφόσον ο χρήστης έχει πατήσει το αντίστοιχο κουμπί (βλέπουμε στο template η function να καλείτε με το ng-click="createNew ()")

```

$scope.createNew = function () {
    $log.log("Λήψη της τρέχουσας θέσης...");
    $ionicLoading.show({
        template: "Λήψη της τρέχουσας θέσης...",
        hideOnStateChange: true
    });
    Geolocation.getCurrentPosition()
        .then(function (position) {
            $log.info("Η τρέχουσα θέση βρέθηκε", position);

            $state.go("geofence-new", {
                latitude: position.coords.latitude,
                longitude: position.coords.longitude
            });
        }, function (reason) {
            $log.error("Δεν είναι δυνατή η λήψη της τρέχουσας τοποθεσίας", reason);
            $ionicLoading.show({
                template: "Δεν είναι δυνατή η λήψη της τρέχουσας τοποθεσίας",
                duration: 1500
            });
        });
    });
};

```

Εικόνα 32

Στη συνέχεια φαίνονται οι λειτουργίες επεξεργασίας σημείου αναφοράς και η διαγραφή του καθώς και η αποθήκευση του τηλεφώνου ειδοποίησης

```

$scope.editGeofence = function (geofence) {
    $state.go("geofence-edit", {
        geofenceId: geofence.id
    });
};

$scope.removeGeofence = function (geofence) {
    Geofence.remove(geofence);
};
/* SAVE NOTIFICATION PHONE NUMBER */

$scope.SavePhone = function() {
    if(typeof(Storage) != "undefined") {
        window.localStorage.setItem("phone", $scope.telephone.phone);
        alert("Το Τηλέφωνο Αποθηκεύτηκε!");
    } else {
        alert("Δεν υποστηρίζεται από την συσκευή σας!");
    }
};

```

Εικόνα 33



Τέλος το κουμπί των περισσότερων επιλογών

```

$scope.more = function () {
    $ionicActionSheet.show({
        titleText: "Περισσότερα",
        destructiveText: "<i class='icon ion-trash-b'></i> Διαγραφή όλων των ειδοποιήσεων",
        cancelText: "<i class='icon ion-android-cancel'></i> Ακύρωση",
        destructiveButtonClicked: function () {
            Geofence.removeAll();
            return true;
        }
    });
};
});

```

Εικόνα 34

Για τις λειτουργίες της εφαρμογής επίσης υπάρχουν και κάποια άλλα αρχεία στα οποία είναι τα αντίστοιχα services της εφαρμογής, Στο AngularJS, μια υπηρεσία είναι μια λειτουργία ή ένα αντικείμενο της εφαρμογής, στην ουσία είναι κάποιες μέθοδοι που καλούμε για μια συγκεκριμένη λειτουργία που θέλουμε να υλοποιήσουμε. Τα αρχεία αυτά βρίσκονται στο path: www/js/services και είναι geofence.js, geolocation.js και το geofence-plugin-mock.js όλα αυτά τα αρχεία έχουν γίνει include στο index.html

**geolocation.js:**

```

angular.module("ionic-geofence").factory("Geolocation", function ($q, $interval) {
    var currentPositionCache;

    return {
        getCurrentPosition: function () {
            if (!currentPositionCache) {
                var deferred = $q.defer();

                navigator.geolocation.getCurrentPosition(function (position) {
                    deferred.resolve(currentPositionCache = position);
                    $interval(function () {
                        currentPositionCache = undefined;
                    }, 10000, 1);
                }, function (error) {
                    deferred.reject(error);
                }, {timeout:10000, enableHighAccuracy: true});

                return deferred.promise;
            }
            return $q.when(currentPositionCache);
        }
    };
});

```

Εικόνα 35

Εδώ γίνεται η λήψη της τοποθεσίας για την δημιουργία του σημείου αναφοράς και η μέθοδος αυτή καλείται αντίστοιχα στο στο αρχείο `www/js/controllers/geofences.js` στη `function createNew ()`

### geofence.js:

Σημαντικά σημεία που αξίζει να δούμε εδώ είναι η δήλωση του σημείου αναφοράς και της ειδοποίησης που λαμβάνει στη συνέχεια ο χρήστης, η αποθήκευση των σημείων στη μνήμη της συσκευής αλλά και η φόρτωσή τους από την μνήμη της συσκευής, όπως φαίνεται παρακάτω:

```

create: function (attributes) {
    var defaultGeofence = {
        id: UUIDjs.create().toString(),
        latitude: 50,
        longitude: 50,
        radius: 1000,
        transitionType: TransitionType.ENTER,
        notification: {
            id: this.getNextNotificationId(),
            title: "Πατήστε για ειδοποίηση",
            text: "",
            icon: "res://ic_menu_mylocation",
            openAppOnClick: true
        }
    };

    return angular.extend(defaultGeofence, attributes);
},

loadFromLocalStorage: function () {
    var result = localStorage["geofences"];
    var geofences = [];

    if (result) {
        try {
            geofences = angular.fromJson(result);
        } catch (ex) {
        }
    }
    this._geofences = geofences;

    return $q.when(this._geofences);
},

saveToLocalStorage: function () {
    localStorage["geofences"] = angular.toJson(this._geofences);
},

```

```
getAll: function () {
    var self = this;

    if (!self._geofencesPromise) {
        self._geofencesPromise = $q.defer();
        self.loadFromDevice().then(function (geofences) {
            self._geofences = geofences;
            self._geofencesPromise.resolve(geofences);
        }, function (reason) {
            $log.error("Error fetching geofences", reason);
            self._geofencesPromise.reject(reason);
        });
    }

    return self._geofencesPromise.promise;
},

addOrUpdate: function (geofence) {
    var self = this;

    return $window.geofence.addOrUpdate(geofence).then(function () {
        var searched = self.findById(geofence.id);

        if (!searched) {
            self._geofences.push(geofence);
        } else {
            var index = self._geofences.indexOf(searched);

            self._geofences[index] = geofence;
        }

        self.saveToLocalStorage();
    });
},
```

Εικόνα 37

Επίσης καλούνται μέθοδοι για την διαγραφή των σημείων, όπως φαίνεται παρακάτω:

```
remove: function (geofence) {
    var self = this;

    $ionicLoading.show({
        template: "Διαγραφή ειδοποίησης",
        duration: 1000
    });
    $window.geofence.remove(geofence.id).then(function () {
        $ionicLoading.hide();
        self._geofences.splice(self._geofences.indexOf(geofence), 1);
        self.saveToLocalStorage();
    }, function (reason) {
        $log.error("Error while removing geofence", reason);
        $ionicLoading.show({
            template: "Error while removing geofence",
            duration: 1500
        });
    });
},

removeAll: function () {
    var self = this;

    $ionicLoading.show({
        template: "Διαγραφή όλων ειδοποιήσεων",
        duration: 1000
    });
    $window.geofence.removeAll().then(function () {
        $ionicLoading.hide();
        self._geofences.length = 0;
        self.saveToLocalStorage();
    }, function (reason) {
        $log.error("Error while removing all geofences", reason);
        $ionicLoading.show({
            template: "Error",
            duration: 1500
        });
    });
});
```

Εικόνα 38

### geofence-plugin-mock.js

Είναι το αρχείο που καλεί τις αντίστοιχες μεθόδους από το plugin του Geofence και το πιο σημαντικό που υπάρχει σε αυτό το αρχείο είναι το σημείο δήλωσης των Transition Types που φαίνεται παρακάτω:

```
receiveTransition: function () {},  
TransitionType: {  
  ENTER: 1,  
  EXIT: 2,  
  BOTH: 3  
}
```

Εικόνα 39

Που επιλέγει ο χρήστης στην δημιουργία σημείου αναφοράς για το πότε επιθυμεί να λάβει την ειδοποίηση (κατά την είσοδο ή κατά την έξοδο στην την περιοχή που έχει ορίσει). Οι μέθοδοι αυτές με την σειρά διεγείρουν κάποιες λειτουργίες στο plugin οι οποίες κατά κύριο λόγο είναι υπεύθυνες για την λειτουργία ορισμού ζώνης και ελέγχου των συντεταγμένων.

Μέχρι εδώ είδαμε πως λειτουργεί το κάθε template και τις λειτουργίες που χρειάζεται οι οποίες πραγματοποιούνται στα αντίστοιχα controllers που εκτελούν κάποιες functions, οι οποίες με τη σειρά τους μπορεί να χρησιμοποιούν κάποιες μεθόδους.

Όπως αναφέραμε και πιο πάνω όλα τα αρχεία αυτά γίνονται include στο index.html κάτω από το οποίο λειτουργεί όλη η εφαρμογή. Σε αυτό το σημείο θα πρέπει να γίνει και η αναφορά στο πιο σημαντικό κομμάτι της λογικής της AngularJS χωρίς την οποία τίποτα από τα παραπάνω δεν θα μπορούσε να εκτελεστεί. Στο αρχείο index.html στο tag body έχει δηλωθεί το ng-app="ionic-geofence". Το ng-app είναι ένα από τα directive της AngularJS που ορίζει την εφαρμογή (app). Με απλά λόγια το ng-app directive ορίζει στην AngularJS τον «ιδιοκτήτη» της εφαρμογής.

Κάτω από το ng-app ionic-geofence λειτουργεί όλη η εφαρμογή.

Στο ionic framework επίσης μέσα στο app θα πρέπει να δηλωθεί η function \$ionicPlatform.ready(function () ) η οποία εκτελεί λειτουργίες εφόσον η συσκευή / τα εγγενή plugins είναι έτοιμα.

Επίσης στη παρούσα εφαρμογή, μέσα στο app εκτελείτε η λειτουργία της ειδοποίησης και της αποστολής του μηνύματος στον κηδεμόνα.

app.js

```
angular.module("ionic-geofence", ["ionic", "leaflet-directive", "ngCordova"]).run(function ($window,
$document,
$ionicLoading,
$state,
$ionicPlatform,
$log,
$rootScope,
GeofencePluginMock,
$cordovaSms,
Geolocation
) {
    $ionicPlatform.ready(function () {
        $log.log("Ionic ready");
    });
});
```

Εικόνα 40

```
$window.geofence.onTransitionReceived = function (geofences) {
    $log.log(geofences);
    if (geofences) {
        $rootScope.$apply(function () {
            geofences.forEach(function (geo) {
                geo.notification = geo.notification || {
                    title: "Geofence transition",
                    text: "Without notification"
                };
                $ionicLoading.show({
                    template: geo.notification.title + ": " + geo.notification.text,
                    noBackdrop: true,
                    duration: 2000
                });
            });
        });
    }
};

$window.geofence.onNotificationClicked = function (notificationData) {
    $log.log(notificationData);
    if (notificationData) {
        $rootScope.$apply(function () {
            $ionicLoading.show({
                template: "Παρακαλώ περιμένετε! Γίνεται αποστολή μηνύματος σε κοντινό σας πρόσωπο σχετικά με την τοποθεσία στην οποία βρίσκεστε.",
                noBackdrop: true,
                duration: 5000
            });

            $state.go("geofence", {
                geofenceId: notificationData.id
            });
        });
        var lat;
        var long;
        var onSuccess = function (position) {
            lati = position.coords.latitude;
            longi = position.coords.longitude;
            lat = lati;
            long = longi;
            console.log(lat);
        }

        // onError Callback receives a PositionError object
        function onError(error) {
            alert('code: ' + error.code + '\n' +
                'message: ' + error.message + '\n');
        }
    }
};
```

Εικόνα 41

```
// onError Callback receives a PositionError object
function onError(error) {
    alert('code: ' + error.code + '\n' +
        'message: ' + error.message + '\n');
}

navigator.geolocation.getCurrentPosition(onSuccess, onError);
var options = {
    replaceLineBreaks: false, // true to replace \n by a new line, false by default
    android: {
        intent: '' // send SMS with the native android SMS messaging
        // intent: '' // send SMS without open any other app
        // intent: 'INTENT' // send SMS inside a default SMS app
    }
};
var number = window.localStorage.getItem("phone");
setTimeout(function() {
    var message = "DoApp! Χρήστης εκτός ακτίνας. Θέση " + lat + ", " + long;
    $cordovaSms
        .send(number, message, options)
        // navigator.geolocation.clearWatch(watchID);

        .then(function() {
            alert('Το μήνυμα εστάλη');
            // Success! SMS was sent
        }, function(error) {
            alert('Πρόβλημα αποστολής μηνύματος');
            // An error occurred
        });
}, 5000);
/*
ANOΙΓΕΙ APP ΓΙΑ ΝΑ ΚΑΛΕΣΕΙ
number = window.localStorage.getItem("phone");
window.location.href = "tel:"+number;
*/
}
};
```

Εικόνα 42

Εδώ φαίνεται η δημιουργία και η εμφάνιση της ειδοποίησης στον χρήστη καθώς και η αποστολή του μηνύματος στον κηδεμόνα με το αντίστοιχο μήνυμα που περιλαμβάνει και την τρέχουσα τοποθεσία του κατόχου της συσκευής. Για την αποστολή της τοποθεσία καλείτε ξανά η μέθοδος `getCurrentPosition` για τον ορισμό όσο το δυνατόν πιο ακριβή τοποθεσίας του χρήστη.

Άλλες λειτουργίες της εφαρμογής (βοηθητικές λειτουργίες).

## ΓΡΗΓΟΡΗ ΚΛΗΣΗ:

Template (view):

```
<div class="bar bar-header bar-positive">
  <h1 class="title">ΓΡΗΓΟΡΗ ΚΛΗΣΗ</h1>
  <a href="#/geofences"><button class="button icon ion-home"></button></a>
</div>
<ion-content class="padding has-header">
  <div class="list list-inset padding">
    <div class="item center"><h3 class="bold dark">{{phone1.name1 | uppercase}}</h3>
    <a href="{{tel:{{phone1.number1}}}" class="button button-block icon icon ion-ios-telephone button-balanced"></a>
    <div class="item item-input-inset">
      <label class="item item-input-wrapper">
        <i class="icon ion-person placeholder-icon"></i>
        <input type="text" ng-model="phone1.name1" placeholder="Όνομα">
      </label>
    </div>
    <div class="item item-input-inset">
      <label class="item item-input-wrapper">
        <i class="icon ion-ios-telephone placeholder-icon"></i>
        <input type="text" ng-model="phone1.number1" placeholder="Τηλέφωνο">
      </label>
    </div>
    <button class="button bold button-calm button-full" ng-click="SaveOne()">
      ΑΠΟΘΗΚΕΥΣΗ
    </button>
  </div>
</div>
<div class="list list-inset padding">
  <div class="item center"><h3 class="bold dark">{{phone2.name2 | uppercase}}</h3>
  <a href="{{tel:{{phone2.number2}}}" class="button button-block icon icon ion-ios-telephone button-balanced"></a>
  <div class="item item-input-inset">
    <label class="item item-input-wrapper">
      <i class="icon ion-person placeholder-icon"></i>
      <input type="text" ng-model="phone2.name2" placeholder="Όνομα">
    </label>
  </div>
  <div class="item item-input-inset">
    <label class="item item-input-wrapper">
      <i class="icon ion-ios-telephone placeholder-icon"></i>
      <input type="text" ng-model="phone2.number2" placeholder="Τηλέφωνο">
    </label>
  </div>
  <button class="button bold button-calm button-full" ng-click="SaveTwo()">
    ΑΠΟΘΗΚΕΥΣΗ
  </button>
</div>
</div>
<div class="list list-inset padding">
  <div class="item center"><h3 class="bold dark">{{phone3.name3 | uppercase}}</h3>
  <a href="{{tel:{{phone3.number3}}}" class="button button-block icon icon ion-ios-telephone button-balanced"></a>
  <div class="item item-input-inset">
    <label class="item item-input-wrapper">
      <i class="icon ion-person placeholder-icon"></i>
      <input type="text" ng-model="phone3.name3" placeholder="Όνομα">
    </label>
  </div>
  <div class="item item-input-inset">
    <label class="item item-input-wrapper">
      <i class="icon ion-ios-telephone placeholder-icon"></i>
      <input type="text" ng-model="phone3.number3" placeholder="Τηλέφωνο">
    </label>
  </div>
  <button class="button bold button-calm button-full" ng-click="SaveThree()">
    ΑΠΟΘΗΚΕΥΣΗ
  </button>
</div>
</div>
```

Εικόνα 43



```

</div>
<button class="button bold button-calm button-full" ng-click="SaveThree()">
  ΑΠΟΘΗΚΕΥΣΗ
</button>
</div>
</div>
<div class="list list-inset padding">
<div class="item center"><h3 class="bold dark">{{phone4.name4 | uppercase}}</h3>
<a href={{tel:phone4.number4}} class="button button-block icon icon ion-ios-telephone button-balanced"></a>
<div class="item item-input-inset">
<label class="item item-input-wrapper">
<i class="icon ion-person placeholder-icon"></i>
<input type="text" ng-model="phone4.name4" placeholder="Όνομα">
</label>
</div>
<div class="item item-input-inset">
<label class="item item-input-wrapper">
<i class="icon ion-ios-telephone placeholder-icon"></i>
<input type="text" ng-model="phone4.number4" placeholder="Τηλέφωνο">
</label>
</div>
<button class="button bold button-calm button-full" ng-click="SaveFour()">
  ΑΠΟΘΗΚΕΥΣΗ
</button>
</div>
</div>
<div class="list list-inset padding">
<div class="item center"><h3 class="bold dark">{{phone5.name5 | uppercase}}</h3>
<a href={{tel:phone5.number5}} class="button button-block icon icon ion-ios-telephone button-balanced"></a>
<div class="item item-input-inset">
<label class="item item-input-wrapper">
<i class="icon ion-person placeholder-icon"></i>
<input type="text" ng-model="phone5.name5" placeholder="Όνομα">
</label>
</div>
<div class="item item-input-inset">
<label class="item item-input-wrapper">
<i class="icon ion-ios-telephone placeholder-icon"></i>
<input type="text" ng-model="phone5.number5" placeholder="Τηλέφωνο">
</label>
</div>
<button class="button bold button-calm button-full" ng-click="SaveFive()">
  ΑΠΟΘΗΚΕΥΣΗ
</button>
</div>
</div>
</ion-content>
</ion-view>

```

Εικόνα 44

Controller (js):

```

angular.module("ionic-geofence").controller("QuickCall", function (
  $scope,
  $ionicLoading
) {
  /* --- QUICKPHONES PAGE --- */
  $scope.phone1 = {
    name1: '',
    number1 : ''
  };
  if(typeof(Storage) != "undefined") {
    $scope.phone1.name1 = window.localStorage.getItem("name1");
    $scope.phone1.number1 = window.localStorage.getItem("number1");
  } else {
    alert("Δεν υποστηρίζεται από την συσκευή σας!");
  }
  $scope.SaveOne = function() {
    if(typeof(Storage) != "undefined") {
      window.localStorage.setItem("name1", $scope.phone1.name1);
      window.localStorage.setItem("number1", $scope.phone1.number1);
      alert("Το Τηλέφωνο Αποθηκεύτηκε!");
    } else {
      alert("Δεν υποστηρίζεται από την συσκευή σας!");
    }
  };

  $scope.phone2 = {
    name2: '',
    number2 : ''
  };
  if(typeof(Storage) != "undefined") {
    $scope.phone2.name2 = window.localStorage.getItem("name2");
    $scope.phone2.number2 = window.localStorage.getItem("number2");
  } else {
    alert("Δεν υποστηρίζεται από την συσκευή σας!");
  }
  $scope.SaveTwo = function() {
    if(typeof(Storage) != "undefined") {
      window.localStorage.setItem("name2", $scope.phone2.name2);
      window.localStorage.setItem("number2", $scope.phone2.number2);
      alert("Το Τηλέφωνο Αποθηκεύτηκε!");
    } else {
      alert("Δεν υποστηρίζεται από την συσκευή σας!");
    }
  };
};

```

Εικόνα 45

```

$scope.phone3 = {
  name3: '',
  number3 : ''
};
if(typeof(Storage) != "undefined") {
  $scope.phone3.name3 = window.localStorage.getItem("name3");
  $scope.phone3.number3 = window.localStorage.getItem("number3");
} else {
  alert("Δεν υποστηρίζεται από την συσκευή σας!");
}
$scope.SaveThree = function() {
  if(typeof(Storage) != "undefined") {
    window.localStorage.setItem("name3", $scope.phone3.name3);
    window.localStorage.setItem("number3", $scope.phone3.number3);
    alert("Το Τηλέφωνο Αποθηκεύτηκε!");
  } else {
    alert("Δεν υποστηρίζεται από την συσκευή σας!");
  }
};

$scope.phone4 = {
  name4: '',
  number4 : ''
};
if(typeof(Storage) != "undefined") {
  $scope.phone4.name4 = window.localStorage.getItem("name4");
  $scope.phone4.number4 = window.localStorage.getItem("number4");
} else {
  alert("Δεν υποστηρίζεται από την συσκευή σας!");
}
$scope.SaveFour = function() {
  if(typeof(Storage) != "undefined") {
    window.localStorage.setItem("name4", $scope.phone4.name4);
    window.localStorage.setItem("number4", $scope.phone4.number4);
    alert("Το Τηλέφωνο Αποθηκεύτηκε!");
  } else {
    alert("Δεν υποστηρίζεται από την συσκευή σας!");
  }
};

$scope.phone5 = {
  name5: '',
  number5 : ''
};
if(typeof(Storage) != "undefined") {
  $scope.phone5.name5 = window.localStorage.getItem("name5");
  $scope.phone5.number5 = window.localStorage.getItem("number5");
} else {
  alert("Δεν υποστηρίζεται από την συσκευή σας!");
}
$scope.SaveFive = function() {
  if(typeof(Storage) != "undefined") {
    window.localStorage.setItem("name5", $scope.phone5.name5);
    window.localStorage.setItem("number5", $scope.phone5.number5);
    alert("Το Τηλέφωνο Αποθηκεύτηκε!");
  } else {
    alert("Δεν υποστηρίζεται από την συσκευή σας!");
  }
};

```

Εικόνα 46

## ΧΡΗΣΙΜΑ ΤΗΛΕΦΩΝΑ

Template (view):

```
<div class="bar bar-header bar-positive">
  <h1 class="title">ΧΡΗΣΙΜΑ ΤΗΛΕΦΩΝΑ</h1>
  <a href="#/home"><button class="button icon ion-home"></button></a>
</div>
<ion-content class="padding has-header">
  <div class="list list-inset padding">
    <div class="item center bdGr"><h3 class="bold positive">ΚΕΝΤΡΟ ΑΜΕΣΗΣ ΒΟΗΘΕΙΑΣ</h3>
    <a href="tel:166" class="button button-block icon icon icon ion-ios-telephone button button-balanced"></a>
    </div>
  </div>
  <div class="list list-inset padding">
    <div class="item center bdGr"><h3 class="bold positive">ΓΡΑΜΜΗ ΖΩΗΣ SOS</h3>
    <a href="tel:1065" class="button button-block icon icon icon ion-ios-telephone button button-balanced"></a>
    </div>
  </div>
  <div class="list list-inset padding">
    <div class="item center bdGr"><h3 class="bold positive">ΑΜΕΣΗ ΚΟΙΝΩΝΙΚΗ ΒΟΗΘΕΙΑ</h3>
    <a href="tel:197" class="button button-block icon icon icon ion-ios-telephone button button-balanced"></a>
    </div>
  </div>
  <div class="list list-inset padding">
    <div class="item center bdGr"><h3 class="bold positive">ΑΜΕΣΗ ΔΡΑΣΗ ΑΣΤΥΝΟΜΙΑΣ</h3>
    <a href="tel:100" class="button button-block icon icon icon ion-ios-telephone button button-balanced"></a>
    </div>
  </div>
  <div class="list list-inset padding">
    <div class="item center bdGr"><h3 class="bold positive">ΠΥΡΟΣΒΕΣΤΙΚΗ ΥΠΗΡΕΣΙΑ</h3>
    <a href="tel:199" class="button button-block icon icon icon ion-ios-telephone button button-balanced"></a>
    </div>
  </div>
</ion-content>
</ion-view>
```

Εικόνα 47

Η συγκεκριμένη λειτουργία χάρη στην απλότητά της δεν έχει τον αντίστοιχο controller καθώς δεν περιέχει κάποιες σύνθετες λειτουργίες.

## Συμπεράσματα και προτάσεις

Με την ολοκλήρωση της παρούσας πτυχιακής εργασίας, μπορούμε να θεωρήσουμε ότι επιτεύχθηκε ο αρχικός σκοπός, που ήταν η ανάπτυξη εφαρμογής για κινητά τηλέφωνα και ταμπλέτες για λειτουργικό σύστημα Android με τη χρήση του Ionic Framework. Για την υλοποίησή της χρησιμοποιήθηκαν γλώσσες προγραμματισμού, «εργαλεία», πρόσθετα και βιβλιοθήκες, τα οποία είναι λογισμικό ανοικτού κώδικα, σύγχρονα και αποτελούν μέχρι και σήμερα τους βασικούς πυλώνες ανάπτυξης υβριδικών εφαρμογών. Η εγκατάστασή τους «έκρυβε» δυσκολίες, κυρίως ως προς τις απαιτήσεις των δυνατοτήτων του ηλεκτρονικού υπολογιστή, αλλά και αργότερα με τις ενημερώσεις (updates), κρατώντας με πάντα σε εγρήγορση για την επικαιροποίηση της εφαρμογής. Λόγω του αντικειμένου της εφαρμογής, έπρεπε να διασφαλισθούν η ευκολία στη χρήση της από τον τελικό χρήστη, αλλά και η απλότητα των λειτουργιών, θεωρώ πως το τελικό αποτέλεσμα ανταποκρίνεται σε αυτούς τους στόχους. Έχοντας λειτουργήσει την εφαρμογή DoApp καθώς και την συνοδευτική ιστοσελίδα DoApp.gr, θεωρώ ότι πρόκειται για μια ολοκληρωμένη εφαρμογή, η οποία μπορεί να αποτελέσει ένα σημαντικό εργαλείο για άτομα που χρήζουν κηδεμονίας, χωρίς αυτό να σημαίνει ότι δεν μπορεί να τροποποιηθεί, εξελιχθεί ή και να προσαρμοστεί και για άλλες ομάδες χρηστών.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] <https://www.eleftherostypos.gr/istories/62000-i-istoria-tis-kinitis-tilefonias-apo-to-xt-hes-sto-simera-kai-epistrofi-sto-xthes/>
- [2] <https://smartphoneproject1.weebly.com/lambdaepsiloniotatauomicronupsilonrhogammaiotaakappaalpha-sigmaupsilonsigmatauetaumualphataualpha-kappaalphaiota-epsilonpsilonphialpharhomuomicrongammaepsilonpsilonsigma.html>
- [3] [https://www.academia.edu/29617264/Mobile\\_Web\\_Development?auto=download](https://www.academia.edu/29617264/Mobile_Web_Development?auto=download)
- [4] <https://www.mobiloud.com/blog/native-web-or-hybrid-apps/#2>
- [5] <https://ionicframework.com/docs/v1/overview/>
- [6] <https://ionicframework.com/docs/v1/concepts/structure.html>
- [7] <https://cordova.apache.org/docs/en/latest/guide/overview/index.html>
- [8] <https://w3c.github.io/html/>
- [9] <https://www.w3.org/TR/CSS/>
- [10] <https://sass-lang.com/community>
- [11] JavaScript: The Good Parts, Douglas Crockford, 2008
- [12] ng-book 2, Felipe Coury, Ari Lerner, Nate Murray, & Carlos Taborda, 2016
- [13] <https://www.json.org/>
- [14] <https://leafletjs.com/index.html>
- [15] <https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-geolocation/>
- [16] <https://www.npmjs.com/package/cordova-plugin-geofence>
- [17] <https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-device/>
- [18] <https://www.npmjs.com/package/cordova-plugin-sms>
- [19] <https://getbootstrap.com/docs/4.2/getting-started/introduction/>
- [20] jQuery For Dummies, by Lynn Beighley, Published by Wiley Publishing, Inc. 2010