



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**

**ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΩΝ**

**Ανάπτυξη εφαρμογής εύρεσης  
συνταξιδιωτών με Ionic Framework**

**TravelPal application with Ionic framework**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

του

**Χρήστου Λαγανά**

Αθήνα, Μάρτιος 2019

Η σελίδα αυτή είναι σκόπιμα λευκή.



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ  
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

## **Ανάπτυξη εφαρμογής εύρεσης συνταξιδιωτών με Ionic Framework**

### **TravelPal application with Ionic framework**

#### **ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

του

**Χρήστου Λαγανά**

**Επιβλέποντες :**

**Δασκαλοπούλου Ασπασία**  
Επίκουρος Καθηγήτρια Π.Θ.

**Σταμούλης Γεώργιος**  
Καθηγητής Π.Θ.

Εγκρίθηκε από την διμελή εξεταστική επιτροπή την ημερομηνία  
εξέτασης.

*(Υπογραφή)*

.....

**Δασκαλοπούλου Ασπασία**  
Επίκουρος Καθηγήτρια Π.Θ

*(Υπογραφή)*

.....

**Σταμούλης Γεώργιος**  
Καθηγητής Π.Θ.

Αθήνα, Μάρτιος 2019

*(Υπογραφή)*

.....

**Χρήστος Λαγανάς**

Διπλωματούχος Μηχανικός Ηλεκτρονικών Υπολογιστών,  
Τηλεπικοινωνιών και Δικτύων του Τμήματος Ηλεκτρολόγων  
Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστημίου Θεσσαλίας

© 2019 – All rights reserved

Η σελίδα αυτή είναι σκόπιμα λευκή.



## **Ευχαριστίες**

Θα ήθελα να εκφράσω την βαθιά μου ευγνωμοσύνη στην κα. Δασκαλοπούλου Ασπασία , την επιβλέπουσα καθηγήτριά μου για την καθοδήγηση και τον χρόνο που αφιέρωσε στην εκπόνηση της διπλωματικής αυτής εργασίας.

Τέλος , θέλω να ευχαριστήσω τους γονείς και τους φίλους μου για την αμέριστη υποστήριξη, την επιμονή και την ενθάρρυνση τους κατά τη διάρκεια των σπουδών μου.

## Περίληψη

Τα «έξυπνα» κινητά τηλέφωνα ή smartphones αποτελούν πλέον αναπόσπαστο κομμάτι της κοινωνίας μας. Δισεκατομμύρια άνθρωποι σε όλο τον πλανήτη τα χρησιμοποιούν όχι μόνο για επικοινωνία αλλά και για διασκέδαση, βελτίωση και διευκόλυνση του τρόπου ζωής και της καθημερινότητάς τους μέσω πολυάριθμων εφαρμογών που διατίθενται.

Η ύπαρξη πολλών διαφορετικών συσκευών με διαφορετικά λειτουργικά συστήματα δυσκολεύει την ανάπτυξη κάθε εφαρμογής ώστε να είναι συμβατή με όλα τα λειτουργικά.

Σκοπός της εργασίας είναι η ανάλυση των διαφορετικών τρόπων ανάπτυξης μιας εφαρμογής, τα πλεονεκτήματα και τα μειονεκτήματά τους καθώς και η ανάπτυξη μιας υβριδικής εφαρμογής εύρεσης συνταξιδιωτών χρησιμοποιώντας το Ionic Framework.

## **Abstract**

Nowadays, smartphones are now an integral part of our society. Billions of people across the globe are using them not only for communication but also for entertainment, improving and facilitating their daily lifestyle through numerous available applications.

The existence of many different devices with different operating systems makes it difficult to deploy any application to be compatible with all of them.

The aim of the thesis is to analyze the different ways of developing an application, their advantages and disadvantages as well as the development of a hybrid application for finding fellow travelers using the Ionic Framework.

## 1 Πίνακας περιεχομένων

1	Εισαγωγή .....	14
1.1	Ιστορικά στοιχεία.....	14
1.2	Δομή διπλωματικής εργασίας .....	15
2	Ανάλυση της Αγοράς έξυπνων φορητών συσκευών και στατιστικά στοιχεία.....	15
3	Ανάλυση λειτουργικών συστημάτων έξυπνων συσκευών .....	18
3.1	Android .....	18
3.2	i-OS.....	20
3.3	Μερίδιο στην Αγορά.....	22
4	Ανάλυση εφαρμογών για φορητές συσκευές .....	23
4.1	Εγγενείς εφαρμογές – Native applications .....	24
4.2	Διαδικτυακές εφαρμογές – Web applications.....	24
4.3	Υβριδικές εφαρμογές – Hybrid applications.....	25
4.4	Πλεονεκτήματα – Μειονεκτήματα .....	26
5	Τεχνολογίες ανάπτυξης υβριδικής εφαρμογής.....	28
5.1	Ionic Framework .....	28
5.2	Apache Cordova .....	29
5.3	Node.JS.....	30
5.4	Visual Studio Code .....	31
5.5	AngularJS.....	32
5.6	Google Firebase .....	34
5.7	Τεχνολογίες Παγκόσμιου Ιστού .....	35
5.7.1	HTML5 .....	35
5.7.2	Typescript.....	37
5.7.3	CSS3.....	38
5.8	Έρευνα προγραμματιστών Ionic 2017 .....	39

6	Παρουσίαση και Εργαλεία ανάπτυξης εφαρμογής.....	44
6.1	Στόχος της εφαρμογής.....	44
6.2	Απαιτήσεις συστήματος .....	44
6.3	Οδηγίες εγκατάστασης εργαλείων .....	44
6.4	Οδηγίες χρήσης εφαρμογής .....	45
6.5	Παρουσίαση λειτουργιών της εφαρμογής .....	45
6.6	Δοκιμές .....	51
7	Επίλογος.....	52
7.1	Συμπεράσματα.....	52
7.2	Μελλοντικές επεκτάσεις και αναβαθμίσεις .....	52
8	Βιβλιογραφία .....	54

# 1 Εισαγωγή

## 1.1 Ιστορικά στοιχεία

Ήταν ένα πρωινό της 3ης Απριλίου 1973 όπου ο Δρ. Martin Cooper της Motorola χρησιμοποιώντας μια πρωτότυπη συσκευή μήκους 25 εκατοστών και βάρους 900 γραμμαρίων, με την ονομασία Motorola DynaTAC 8000x έκανε την πρώτη κλήση κινητού τηλεφώνου.

Στις αρχές της δεκαετίας του '80 ξεκίνησε να λειτουργεί το πρώτο αυτοματοποιημένο δίκτυο κινητής τηλεφωνίας και άρχισε η παραγωγή κινητών συσκευών για το ευρύ αγοραστικό κοινό. Μέχρι τα τέλη της δεκαετίας τα κινητά τηλέφωνα ήταν ογκώδη για να μεταφέρονται στην τσέπη κι έτσι ήταν εγκατεστημένα κυρίως σε αυτοκίνητα.

Έπρεπε να φτάσουμε στις αρχές της δεκαετίας του 1990 για να ξεκινήσει η απογείωση των κινητών τηλεφώνων λόγω της ψηφιοποίησης των δικτύων και συσκευών. Πρώτη η Nokia ξεκίνησε την παραγωγή κινητών συσκευών μειώνοντας το μέγεθος κάνοντας το κινητό τηλέφωνο πιο φιλικό στο μέσο χρήστη. Ακολούθησε η IBM που προσπάθησε να προσθέσει στις συσκευές χαρακτηριστικά και λειτουργίες PDA όπως ημερολόγιο, αριθμομηχανή, σημειωματάριο και παιχνίδια. Ήταν το Simon, η πρώτη συσκευή που μπορεί να αναφέρεται ως smartphone έστω και αν ο συγκεκριμένος όρος δεν είχε ακόμη επινοηθεί.

Μέχρι το 2006 τα smartphones ήταν κυρίως επαγγελματική υπόθεση λόγω υψηλού κόστους σε σχέση με τα συμβατικά κινητά. Εκείνη τη χρονιά η Nokia πήρε την απόφαση να απευθυνθεί και σε μη επαγγελματίες χρήστες με τη σειρά N.

Το μεγαλύτερο όμως βήμα προς την εξάπλωση της αγοράς των smartphones έγινε το 2007 όταν ο πρόεδρος της Apple Inc. Steve Jobs παρουσίασε το πρώτο iPhone. Το iPhone συνδύαζε τις λειτουργίες και τις εφαρμογές των smartphones με τη δυνατότητα αγοράς νέων εφαρμογών μέσω του app Store. Έκτοτε όλες οι συσκευές iPhone χρησιμοποιούν το ίδιο λειτουργικό σύστημα iOS και οι εφαρμογές του γράφονται χρησιμοποιώντας την γλώσσα προγραμματισμού objective-C.

Λίγους μήνες αργότερα η Google Inc. παρουσιάζει την πρώτη της συσκευή με λειτουργικό σύστημα Android. Το Android είναι ένα λειτουργικό ανοικτού κώδικα το οποίο τρέχει τον πυρήνα του λειτουργικού Linux και οι εφαρμογές του γράφονται στη γλώσσα προγραμματισμού Java.

Λόγω της μεγάλης δημοτικότητας της συγκεκριμένης γλώσσας το Android γρήγορα θα γίνει το δημοφιλέστερο λειτουργικό σύστημα για κινητά και θα υιοθετηθεί από τις

περισσότερες και μεγαλύτερες εταιρίες κατασκευής κινητών συσκευών. Τα επόμενα χρόνια τα δυο λειτουργικά θα επικρατήσουν και σχεδόν θα εξαλείψουν τον όποιο ανταγωνισμό.

Η ύπαρξη δυο διαφορετικών γλωσσών προγραμματισμού εφαρμογών μας οδηγεί στην αναγκαστική ανάπτυξη της κάθε εφαρμογής από την αρχή για το κάθε ένα λειτουργικό σύστημα. Αυτό αυξάνει το χρόνο παραγωγής και έκδοσης των εφαρμογών άρα και του κόστους τους. Για την λύση του παραπάνω προβλήματος και τη δημιουργία υβριδικών εφαρμογών οι προγραμματιστές αναγκάστηκαν να καταφύγουν στη χρήση κάποιων νέων τεχνολογιών διαδικτύου.

## **1.2 Δομή διπλωματικής εργασίας**

Στο σημείο αυτό για την διευκόλυνση του αναγνώστη δίνεται μια διάρθρωση της δομής της διπλωματικής εργασίας . Στο 2<sup>ο</sup> κεφάλαιο παρουσιάζεται μια ανάλυση της αγοράς έξυπνων φορητών συσκευών. Το 3<sup>ο</sup> κεφάλαιο που ακολουθεί, αναλύει τα σημαντικότερα λειτουργικά συστήματα που χρησιμοποιούνται σήμερα. Το 4<sup>ο</sup> κεφάλαιο παρουσιάζει και αναλύει τους διαφορετικούς τρόπους ανάπτυξης εφαρμογών και τα πλεονεκτήματα και μειονεκτήματα κάθε κατηγορίας. Στη συνέχεια, στο 5<sup>ο</sup> κεφάλαιο γίνεται παρουσίαση των τεχνολογιών ανάπτυξης μιας υβριδικής εφαρμογής. Στο 6<sup>ο</sup> κεφάλαιο παρουσιάζεται η εφαρμογή, αναλύεται το λογισμικό και τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής και οδηγίες εγκατάστασής τους. Ακόμα, στο κεφάλαιο αυτό αναλύονται οι λειτουργίες και ο κώδικας της εφαρμογής. Τέλος, τα κεφάλαια 7 και 8 περιέχουν τον επίλογο και τις αναφορές σε βιβλιογραφία και περαιτέρω παραρτήματα

## **2 Ανάλυση της Αγοράς έξυπνων φορητών συσκευών και στατιστικά στοιχεία**

Από το 2007 και μετά οι πωλήσεις των smartphones αυξάνονται συνεχώς με αλματώδη βήματα. Σύμφωνα με έρευνες, στα τέλη του Νοεμβρίου του 2011, από το σύνολο των φωτογραφιών που δημιουργήθηκαν, το 27% τραβήχτηκε με smartphone εξοπλισμένο με κάμερα. Το Σεπτέμβριο του 2012 άλλη έρευνα έδειξε ότι το 80% των ιδιοκτητών smartphone χρησιμοποιεί την συσκευή του για online αγορές. Στις αρχές του 2013 οι πωλήσεις smartphones προσπέρασαν τις πωλήσεις συμβατικών κινητών τη-

λεφώνων και ξεπέρασαν το 1 δισεκατομμύριο κομμάτια, αυξημένες κατά 38% σε σχέση με τα 725 εκατομμύρια της προηγούμενης χρονιάς.

Σήμερα εκτιμάτε ότι κυκλοφορούν περισσότερα από 2.5 δισεκατομμύρια συσκευές δηλαδή περίπου το 30% του παγκόσμιου πληθυσμού έχει στην κατοχή του smartphone. Πρώτες στις πωλήσεις βρίσκονται η Samsung και η Apple Inc. κατέχοντας περίπου το 20.7% και 13.7% αντίστοιχα της παγκόσμιας αγοράς. Ακολουθούν άλλες μεγάλες εταιρίες όπως η Huawei με 9%, η Oppo με 8.1% και η Vivo με 6,8% (πίνακας 1).

**Worldwide Smartphone Sales to End Users by Vendor in 1Q17 (Thousands of Units)**

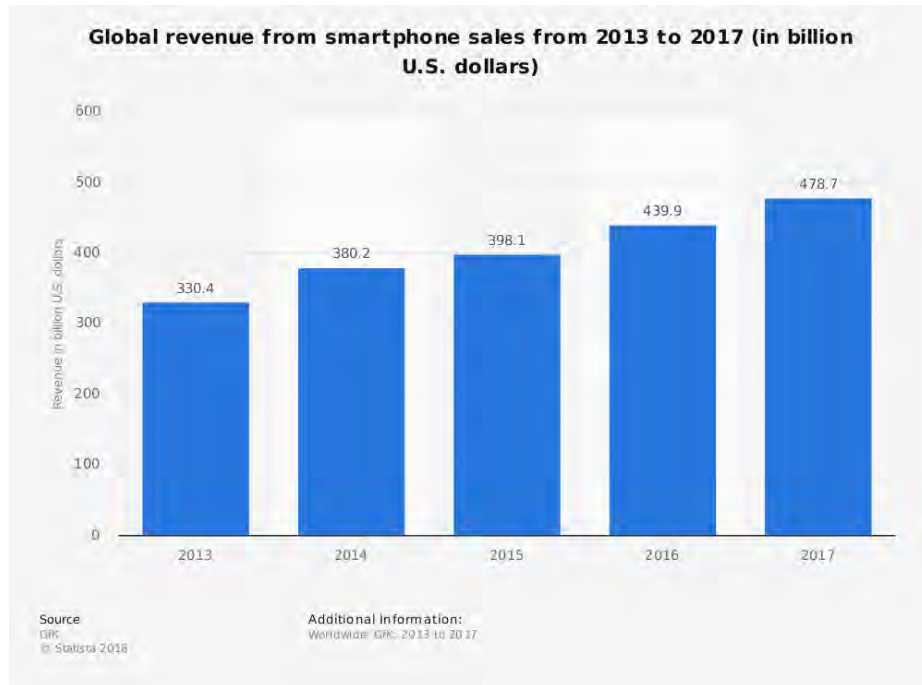
Vendor	1Q17 Units	1Q17 Market Share (%)	1Q16 Units	1Q16 Market Share (%)
Samsung	78,671.4	20.7	81,186.9	23.3
Apple	51,992.5	13.7	51,629.5	14.8
Huawei	34,181.2	9.0	28,861.0	8.3
Oppo	30,922.3	8.1	15,891.5	4.6
Vivo	25,842.2	6.8	14,001.0	4.0
Others	158,367.7	41.7	156,654.2	45.0
<b>Total</b>	<b>379,977.3</b>	<b>100.0</b>	<b>348,224.2</b>	<b>100.0</b>

Source: Gartner (May 2017)

**Πίνακας 1: Πωλήσεις έξυπνων κινητών το 1<sup>ο</sup> τρίμηνο του 2017**

Από μεριάς κερδών, οι κατασκευάστριες εταιρίες παρουσιάζουν συνεχώς αυξανόμενα έσοδα. Από τα 330 δισεκατομμύρια δολάρια του 2013 ανέβηκαν στα 478 δισεκατομμύρια δολάρια το 2017(εικόνα 1).





**Εικόνα 1: Συνολικά έσοδα από την πώληση έξυπνων κινητών**

Από αυτά το περίπου 37% αντιστοιχεί στην αγορά της Κίνας ενώ ακολουθεί η αγορά της Αμερικής με 8.5%. Σύμφωνα με στοιχεία του Forbes το μεγαλύτερο μερίδιο των κερδών αντιστοιχεί στην Apple με 44% και έπεται η Samsung με 33.6%.

### 3 Ανάλυση λειτουργικών συστημάτων έξυπνων συσκευών

Λειτουργικό σύστημα (αγγλικά: Operating System ή OS) ονομάζεται στην επιστήμη της πληροφορικής το λογισμικό που είναι υπεύθυνο για την διαχείριση και τον συντονισμό των εργασιών καθώς και τη διαχείριση των πόρων κάθε συσκευής. Το λειτουργικό παρέχει ένα διαμεσολαβητικό επίπεδο λογικής σύνδεσης μεταξύ υλικού και λογισμικού. Αποτελείται από ένα σύνολο μηχανισμών μέσω των οποίων επιτυγχάνεται αυτόματη διαχείριση των πόρων μιας συσκευής και ελεγχόμενη κατανομή τους στις εκτελούμενες εφαρμογές, έτσι ώστε οι τελευταίες να είναι σε θέση να προσπελάσουν εύκολα τους πόρους του συστήματος χωρίς να χρειάζεται να γνωρίζουν με ακρίβεια τη δομή του υλικού, αλλά και ώστε πολλαπλές εφαρμογές να μπορούν να εκτελούνται ταυτόχρονα χωρίς να έρχονται σε διένεξη μεταξύ τους ή με τη συσκευή. Τα πιο γνωστά και ευρέως διαδεδομένα λειτουργικά για smartphones είναι: το Android της Google, το i-OS της Apple Inc., το Windows Phone της Microsoft, το BlackBerry OS της RIM και το FireFox OS της Mozilla. Παρακάτω θα αναλύσουμε τα δυο δημοφιλέστερα από αυτά.

#### 3.1 Android



Android ονομάζεται το λειτουργικό που αναπτύχθηκε από την Google και αργότερα την Open Handset Alliance, μια κοινοπραξία 84 εταιριών τεχνολογίας με στόχο την ανάπτυξη ανοικτών προτύπων για κινητές συσκευές. Το Android είναι λογισμικό ανοικτού κώδικα βασισμένο στο λειτουργικό σύστημα Linux και οι εφαρμογές του συνθέτονται με χρήση της γλώσσας προγραμματισμού Java και διάφορων βιβλιοθηκών λογισμικού αναπτυγμένων από την Google. Παρόλο που ξεκίνησε σαν λειτουργικό σύστημα για κινητά τηλέφωνα και συσκευές με οθόνη αφής όπως tablet πλέον χρησιμοποιείται και σε μια πληθώρα ηλεκτρονικών συσκευών όπως έξυπνες τηλεοράσεις, ψηφιακές φωτογραφικές μηχανές, κονσόλες παιχνιδιών, έξυπνα ρολόγια και γυαλιά, συνηθισμένους ηλεκτρονικούς υπολογιστές μέχρι και σε αυτοκίνητα.

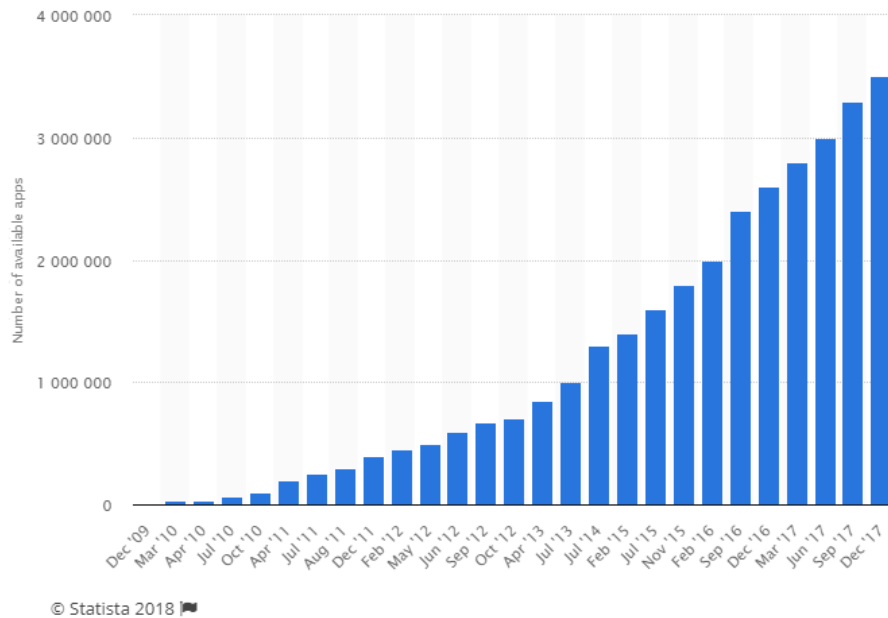
Το Νοέμβριο του 2007 ξεκίνησε η κυκλοφορία της έκδοσης Android Beta. Έπρεπε να περάσει σχεδόν ένας χρόνος για να κυκλοφορήσει η πρώτη εμπορική έκδοση Android 1.0. Από εκεί και έπειτα οι εκδόσεις του Android ανανεώνονται σε τακτά χρονικά διαστήματα και με νέα ονομασία που προέρχεται από τη ζαχαροπλαστική. (πίνακας εκδόσεων)

Κωδικό Όνομα	Νούμερο έκδοσης	Ημερομηνία κυκλοφορίας
N/A	1.0	23 Σεπτεμβρίου 2008
N/A	1.1	9 Φεβρουαρίου 2009
Cupcake	1.5	27 Απριλίου 2009
Donut	1.6	15 Σεπτεμβρίου 2009
Eclair	2.0 – 2.1	26 Οκτωβρίου 2009
Froyo	2.2 – 2.2.3	20 Μαΐου 2010
Gingerbread	2.3 – 2.3.7	6 Δεκεμβρίου 2010
Honeycomb	3.0 – 3.2.6	22 Φεβρουαρίου 2011
Ice Cream Sandwich	4.0 – 4.0.4	18 Οκτωβρίου 2011
Jelly Bean	4.1 – 4.3.1	9 Ιουλίου 2012
KitKat	4.4 – 4.4.4	31 Οκτωβρίου 2013
Lollipop	5.0 – 5.1.1	12 Νοεμβρίου 2014
Marshmallow	6.0 – 6.0.1	5 Οκτωβρίου 2015
Nougat	7.0 – 7.1.2	22 Αυγούστου 2016
Oreo	8.0 – 8.1	21 Αυγούστου 2017

**Πίνακας 2: Εκδόσεις λειτουργικού Android**

Η αγορά των εφαρμογών Android παρουσιάστηκε τον Αύγουστο του 2008 με την ονομασία Android market και εμπεριέχει όλες τις εφαρμογές για συσκευές με το συγκεκριμένο λειτουργικό. Από το Μάρτιο του 2015 συναντάται με τις ονομασίες Google Play και Play Store. Σε λιγότερο από μια δεκαετία ο αριθμός των εφαρμογών στο Play Store εκτοξεύτηκε από τις 16.000 του 2009 στα 3.5 εκατομμύρια του 2017. (εικόνα 2).

Number of available apps in the Google store from December 2009 to December 2017



Εικόνα 2: Πλήθος διαθέσιμων εφαρμογών στο Google Store

Οι χρήστες πλέον έχουν στη διάθεσή τους μια τεράστια γκάμα εφαρμογών οι οποίες διατίθενται είτε δωρεάν είτε έναντι μικρού αντιτίμου.

### 3.2 i-OS



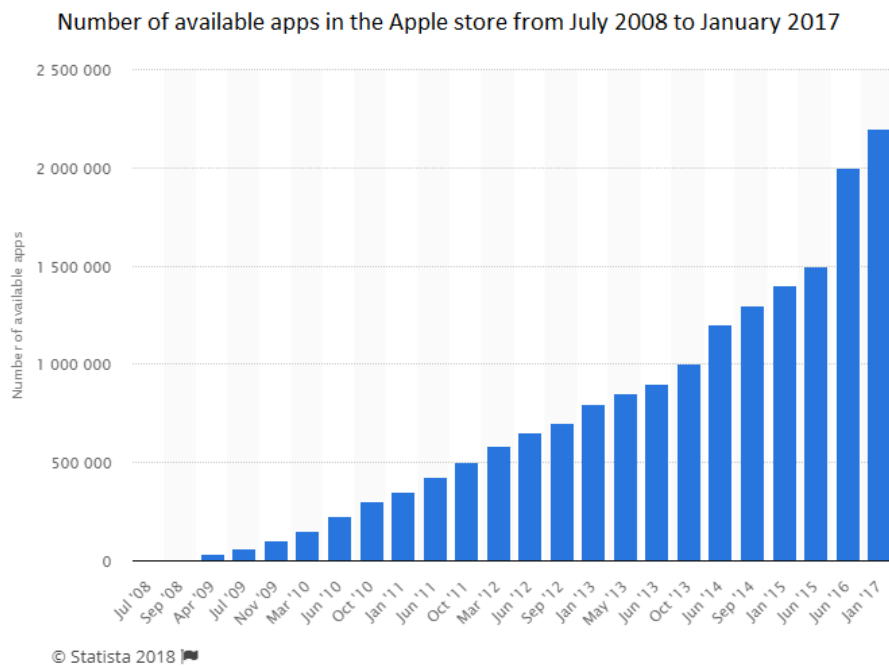
i-OS ονομάζεται το λειτουργικό που αναπτύχθηκε από την Apple Inc και παρουσιάστηκε το 2007. Ξεκίνησε ως λειτουργικό των κινητών τηλεφώνων i-Phone αλλά πλέον χρησιμοποιείται σε όλες τις φορητές συσκευές της Apple όπως τα i-Pad, i-Pod και i-Pod Touch. Αντίθετα με τις υπόλοιπες εταιρίες ανάπτυξης λογισμικού η Apple διαθέτει μια πολύ αυστηρή πολιτική και δεν επιτρέπει την εγκατάσταση του λειτουργικού i-OS σε άλλες

πέραν των δικών της συσκευών.

Οι εφαρμογές για i-OS συνθέτονται με τη χρήση της γλώσσας προγραμματισμού objective-C σε ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) της Apple που ονομάζεται X-Code. Τα δυο τελευταία χρόνια η Apple έχει αναπτύξει και προωθεί μια νέα γλώσσα προγραμματισμού, την Swift, η οποία βασίζεται πάνω στην objective-C αλλά είναι “ασφαφέστερη” καθώς είναι ευκολότερη στον εντοπισμό σφαλμάτων λο-

γισμικού. Στα τέλη του 2017 η Swift κατέλαβε την 11η θέση στη λίστα των πιο δημοφιλών γλωσσών προγραμματισμού.

Η αγορά των εφαρμογών για i-OS ονομάζεται app Store. Παρουσιάστηκε τον Ιούλιο του 2008 και εμπεριείχε μόλις 500 εφαρμογές. Μέσα σε ενάμιση χρόνο οι εφαρμογές είχαν φτάσει τις 150.000 το οποίο αντιστοιχεί περίπου στην έκδοση 275 εφαρμογών ανά ημέρα. Τον Ιούνιο του 2013 ξεπέρασαν για πρώτη φορά το 1 εκατομμύριο και στις αρχές του 2017 έφτασαν τα 2.2 εκατομμύρια. (πίνακας 4)



Εικόνα 3: Πλήθος διαθέσιμων εφαρμογών στο App Store

### 3.3 Μερίδιο στην Αγορά

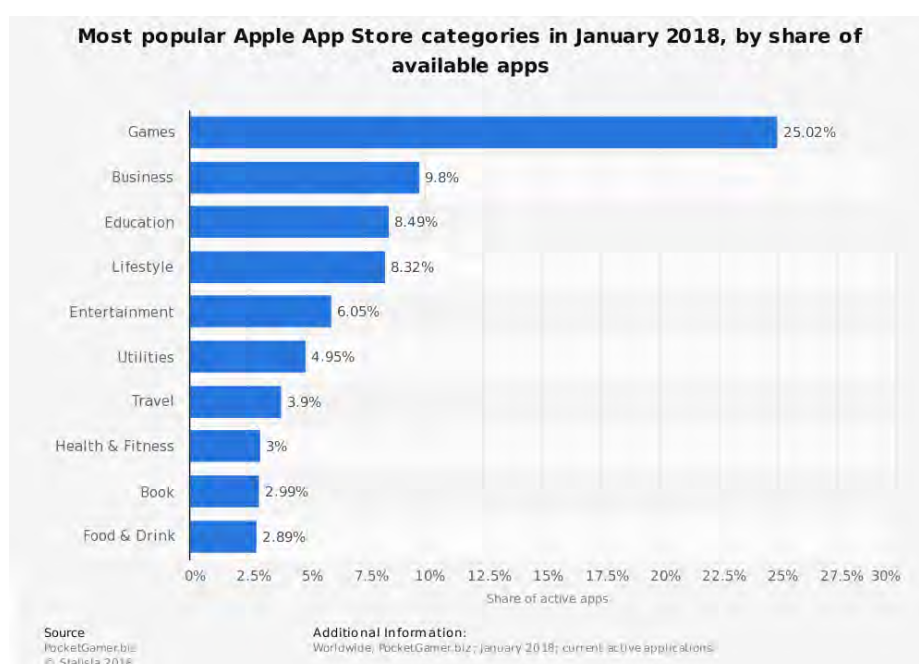
Σύμφωνα με πρόσφατες έρευνες το λειτουργικό Android καταλαμβάνει το μεγαλύτερο κομμάτι της αγοράς με ποσοστό 87,7% έναντι 12.1% του i-OS που είναι και ο βασικός ανταγωνιστής του. Τα δύο λειτουργικά αθροιστικά αγγίζουν το 99.9% της αγοράς οπότε καταλαβαίνουμε ότι μια εφαρμογή που “τρέχει” και στα δυο λειτουργικά απευθύνεται και στο μεγαλύτερο μέρος του αγοραστικού κοινού. (Εικόνα 4)



Εικόνα 4: Μερίδιο της αγοράς των λειτουργικών συστημάτων για smartphones

#### 4 Ανάλυση εφαρμογών για φορητές συσκευές

Με τον όρο εφαρμογή (Mobile app) εννοούμε ένα υπολογιστικό πρόγραμμα σχεδιασμένο για να τρέχει σε μια φορητή συσκευή όπως τηλέφωνο ή tablet. Οι εφαρμογές μπορεί να είναι προεγκατεστημένες στη συσκευή είτε να εγκατασταθούν από το χρήστη μέσω Google/App Store. Σύμφωνα με τελευταίες έρευνες οι δημοφιλέστερη κατηγορία εφαρμογών είναι ο τομέας των παιχνιδιών με 25.2% και ακολουθούν οι τομείς των επιχειρήσεων με 9,8% και της εκπαίδευσης με 8,49%. (πίνακας 6)



Πίνακας 6: Δημοφιλέστερες κατηγορίες εφαρμογών

Οι εφαρμογές παιχνιδιών συνήθως χρειάζονται για την ανάπτυξη τους κάποια πλατφόρμα γραφικών όπως η Unreal Engine και η Unity. Αυτό αυξάνει την πολυπλοκότητα του προγραμματισμού της άρα και του κόστους της. Στον αντίποδα κατηγορίες όπως της εκπαίδευσης ή των ταξιδιών που δεν έχουν την ανάγκη πολύπλοκων γραφικών μπορούν να προγραμματιστούν με τεχνολογίες web (HTML, Java Script, CSS) ως υβριδικές για εξοικονόμηση χρόνου και χρήματος.

Συνήθως όταν αναφερόμαστε σε εφαρμογές εννοούμε τις εγγενείς (native apps), ωστόσο παρακάτω θα αναλύσουμε τις διαφορές και τις ομοιότητες με τις άλλες δυο κατηγορίες : τις διαδικτυακές εφαρμογές (web apps) και τις υβριδικές εφαρμογές

(hybrid apps) καθώς και τα πλεονεκτήματα τα και μειονεκτήματα τις κάθε κατηγορίας.

#### 4.1 Εγγενείς εφαρμογές – Native applications



Όλες οι εφαρμογές που δημιουργούνται με τη χρήση μιας συγκεκριμένης γλώσσας προγραμματισμού για ένα συγκεκριμένο λειτουργικό σύστημα ονομάζεται εγγενής ή μητρική. Οι εφαρμογές αυτές εγκαθίστανται στην εκάστοτε συσκευή μέσω του διαδικτυακού καταστήματος της κάθε εταιρίας (πχ. App store για i-OS) είτε μέσω υπολογιστή (πχ. Σύνδεση με USB) και αλληλεπιδρούν απευθείας με το λειτουργικό χωρίς την χρήση ενδιάμεσου λογισμικού. Επειδή ακριβώς οι εγγενείς εφαρμογές είναι γραμμένες για το εκάστοτε λειτουργικό εκμεταλλεύονται και αξιοποιούν πλήρως τις επιδόσεις και τις λειτουργίες του και αυτό είναι που τις κάνει πιο “γρήγορες” στην απόκριση σε σχέση με τις υπόλοιπες. Επίσης έχουν πλήρη πρόσβαση στους υλικούς πόρους (hardware) της συσκευής όπως κάμερα, μικρόφωνο, πληκτρολόγιο, πωξίδα.

Για τον προγραμματισμό μιας native εφαρμογής χρειάζεται ο προγραμματιστής να μάθει την γλώσσα προγραμματισμού που αντιστοιχεί στο λειτουργικό για το οποίο επιθυμεί να τη διαθέσει (Java για Android ή ObjectiveC/Swift για i-OS). Επίσης χρειάζεται το κατάλληλο εργαλείο και περιβάλλον ανάπτυξης (SDK) για την εκάστοτε γλώσσα.

#### 4.2 Διαδικτυακές εφαρμογές – Web applications



Διαδικτυακές ονομάζουμε τις εφαρμογές που χρησιμοποιούν τον περιηγητή (browser) της συσκευής για την παρουσίασή τους. Δεν είναι πραγματικές εφαρμογές μιας και στην πραγματικότητα είναι ιστοσελίδες που φαίνονται και λειτουργούν σαν εγγενείς εφαρμογές. Ο χρήστης έχει πρόσβαση στην εκάστοτε εφαρμογή χωρίς να την έχει εγκατεστημένη στη συσκευή του καθώς η εφαρμογή είναι εγκατεστημένη σε κάποιο απομακρυσμένο διακομιστή (server) και παραδίδεται στο χρήστη με την επίσκεψη σε κάποια ηλεκτρονική διεύθυνση (url).



Για τον προγραμματισμό των διαδικτυακών χρειάζονται γνώσεις τεχνολογιών διαδικτύου όπως HTML5, JavaScript και CSS. Έτσι αποφεύγεται το εμπόδιο της εκμάθησης μιας νέας γλώσσας προγραμματισμού για κάθε ένα ξεχωριστό λειτουργικό σύστημα. Επίσης δεν χρειάζεται κάποιο συγκεκριμένο περιβάλλον ανάπτυξης για τον προγραμματισμό μιας διαδικτυακής εφαρμογής.

Οι εφαρμογές web λόγω του ότι “τρέχουν” μέσω μιας native εφαρμογής (του browser) μπορούν να εκμεταλλευτούν μόνο τις δυνατότητες που τους προσφέρει αυτός. Η απόδοση και η ταχύτητά τους εξαρτώνται όχι μόνο από την ισχύ του επεξεργαστή της συσκευής αλλά και από την ταχύτητα του browser αλλά και την ταχύτητα του δικτύου μιας και λειτουργούν στέλνοντας αιτήματα σε κάποιο απομακρυσμένο server και αναμένοντας τις απαντήσεις.

### 4.3 Υβριδικές εφαρμογές – Hybrid applications



Συνδυάζοντας στοιχεία από τις δυο προηγούμενες κατηγορίες εφαρμογών οι προγραμματιστές δημιούργησαν μια καινούργια, τις Υβριδικές. Χρησιμοποιώντας τις λειτουργίες web εφαρμογών και μέσω κάποιων διεπαφών προγραμματισμού εφαρμογών (APIs) οι υβριδικές εφαρμογές καταφέρνουν να οπτικοποιούνται και να έχουν πρόσβαση στις δυνατότητες του υλικού (hardware) της συσκευής (πχ. κάμερα, GPS, πυξίδα κλπ.) σαν εγγενείς εφαρμογές.

Για τη δημιουργία μιας εφαρμογής ο προγραμματιστής έχει δυο επιλογές: τη δημιουργία offline έκδοσης και online έκδοσης. Στην offline έκδοση όλα τα αρχεία που απαιτούνται αποθηκεύονται στην συσκευή κατά την εγκατάσταση. Το μειονέκτημα της συγκεκριμένης επιλογής έγκυται στη δυσκολία άμεσων ανανεώσεων της έκδοσης της εφαρμογής μιας και για κάθε ανανέωση απαιτείτε έλεγχο και έγκριση της εφαρμογής από το εκάστοτε App/Google store.

Στην online έκδοση ο προγραμματιστής ουσιαστικά φτιάχνει μια ιστοσελίδα. Αυτό του δίνει τη δυνατότητα να μπορεί να κάνει γρήγορες και εύκολες αλλαγές και ανανεώσεις της έκδοσης της εφαρμογής μιας και δεν χρειάζεται την προαναφερθείσα έγκριση.

Για τον προγραμματισμό υβριδικών εφαρμογών οι προγραμματιστές χρησιμοποιούν τεχνολογίες διαδικτύου (HTML5, CSS, JavaScript) και μέσω κάποιων έτοιμων

εργαλείων και δομών (frameworks) που θα αναλύσουμε αργότερα μετατρέπουν τον κώδικα HTML σε κώδικα που “τρέχει” σε διαφορετικά λειτουργικά (Android/i-OS).

#### 4.4 Πλεονεκτήματα – Μειονεκτήματα

	Εγγενείς Εφαρμογές (Native apps)	Υβριδικές εφαρμογές (Hybrid apps)	Διαδικτυακές εφαρμογές (Web Apps)
Διαθεσιμότητα	App Store (για i-OS)  Play Store (για Android)	App Store (για i-OS)  Play Store (για Android)	Web
Γλώσσα Προγραμματισμού	Objective-C/Swift (για i-OS)  Java (για Android)	HTML5, CSS, JS, Framework	HTML5, CSS, JS
Προσβασιμότητα Συσκευής	Ολική	Ολική	Μερική
Ταχύτητα ανάπτυξης	Αργή	Μέτρια	Γρήγορη
Κόστος ανάπτυξης	Μεγάλο	Μέτριο	Χαμηλό
Κόστος συντήρησης	Μεγάλο	Μέτριο	Χαμηλό
Περιβάλλον χρήστη	Άριστο	Υψηλό	Μέτριο
Απόδοση εφαρμογής	Υψηλή	Μέτρια	Μέτρια
Διαδικασία έγκρισης	Αναγκαστική	Υπό προϋποθέσεις	Καμία

**Πίνακας 3: Πλεονεκτήματα – Μειονεκτήματα τρόπων ανάπτυξης εφαρμογών**

1. Προσβασιμότητα υλικού της συσκευής: Εγγενείς και Υβριδικές εφαρμογές έχουν πλήρη πρόσβαση στο υλικό (hardware) της συσκευής όπως πχ. η κάμερα. Αντίθετα οι διαδικτυακές έχουν πρόσβαση μόνο στο υλικό που τους προσφέρει ο διακομιστής.

2. Περιβάλλον χρήστη: Προτεραιότητα κάθε εφαρμογής είναι να προσφέρει στο χρήστη μια καλή εμπειρία και ανάλογη του λειτουργικού συστήματος. Δηλαδή ο κάθε χρήστης έχει συνηθίσει το λειτουργικό της συσκευής του να έχει κάποια συγκεκριμένα κουμπιά, tabs, layout, themes. Οι εγγενείς εφαρμογές υιοθετούν πλήρως τα παραπάνω χαρακτηριστικά του λειτουργικού οπότε φέρνουν την εφαρμογή πιο κοντά στο γραφικό περιβάλλον που έχει συνηθίσει ο χρήστης. Αυτό δεν σημαίνει βέβαια ότι δεν γίνεται να παρέχετε μια καλή εμπειρία στο χρήστη με μια εφαρμογή ιστού ή μια υβριδική εφαρμογή - σημαίνει απλώς ότι τα γραφικά δεν θα

είναι ακριβώς τα ίδια με εκείνα με τα οποία οι χρήστης μπορεί να είναι ήδη εξοικειωμένος.

3. Ταχύτητα και Κόστος ανάπτυξης: Όπως είναι φυσικό όσο μεγαλώνει ο χρόνος ανάπτυξης μιας εφαρμογής (χρόνος εκμάθησης και εργατοώρες) τόσο αυξάνεται και το κόστος ανάπτυξης μιας και δυο αυτές έννοιες είναι αντιστρόφως ανάλογες. Για τη δημιουργία μια εγγενούς εφαρμογής για όλα τα λειτουργικά χρειάζεται περισσότερες γνώσεις και πολύ περισσότερος χρόνος. Για αυτό το λόγο και οι εγγενείς εφαρμογές είναι πολύ ακριβότερες. Ενδεικτικά να αναφέρουμε ότι το μέσο κόστος ανάπτυξης μιας εγγενούς εφαρμογής κυμαίνεται από 22 έως 27 χιλιάδες δολάρια για μια μικρή επιχείρηση ενώ ξεπερνάει τις 270 χιλιάδες δολάρια για μεγαλύτερες επιχειρήσεις.

4. Κόστος συντήρησης: Από τη στιγμή που βγαίνει στην αγορά κάθε εφαρμογή έχει ένα συνεχές κόστος συντήρησης μιας και πρέπει συνεχώς να δημιουργείτε νέο περιεχόμενο, να παρακολουθείται η απόδοσή της και να διορθώνονται σφάλματα ώστε να κρατάει ευχαριστημένους τους χρήστες της. Ανάλογα με την πολυπλοκότητα της εφαρμογής, μπορεί να υπάρχουν πρόσθετα έξοδα για διακομιστές, συντήρηση έκτακτης ανάγκης, ειδοποιήσεις push, επεξεργασία πληρωμών και άλλων. Σύμφωνα με έρευνες το ετήσιο κόστος συντήρησης αντιστοιχεί περίπου στο 20% του κόστους δημιουργίας της. Οπότε οι εγγενείς εφαρμογές εκτός από μεγαλύτερο κόστος ανάπτυξης έχουν και μεγαλύτερο κόστος συντήρησης.

5. Διαδικασία και κόστος έγκρισης: Για την δημοσιοποίηση της κάθε εγγενούς και υβριδικής εφαρμογής απαιτείται έλεγχος και έγκριση από το κάθε ένα διαφορετικό app store. Επίσης για το “ανέβασμα” της εφαρμογής στο App store της Apple απαιτείται η άδεια προγραμματιστή (developer License) που κοστίζει 99 δολάρια για ένα χρήστη ή 299 δολάρια για επιχείρηση. Αντίθετα, για διαδικτυακή εφαρμογή δεν χρειάζεται κάποια έγκριση.

Με βάση τα παραπάνω καταλήγουμε στο συμπέρασμα ότι οι εγγενείς εφαρμογές είναι η σωστή επιλογή όταν ο χρόνος και το κόστος δεν αποτελούν εμπόδιο. Μεταφράζονται απ' ευθείας σε γλώσσα μηχανής και έτσι αξιοποιούν πλήρως την αποδοτικότητα της εκάστοτε συσκευής.

Αντίθετα όταν υπάρχει θέμα χρόνου και κόστους οι υβριδικές εφαρμογές είναι προτιμότερες μιας και αναπτύσσονται γρηγορότερα και έχουν μικρότερο κόστος ανάπτυξης και συντήρησης.

## 5 Τεχνολογίες ανάπτυξης υβριδικής εφαρμογής

### 5.1 Ionic Framework



Για αρχή θα επεξηγήσουμε την έννοια του framework. Ως framework εννοούμε ένα πλαίσιο εργαλείων που διευκολύνουν όλη τη διαδικασία ανάπτυξης μιας εφαρμογής για κάποια συγκεκριμένη πλατφόρμα ή λειτουργικό. Αυτά τα πλαίσια συνήθως περιλαμβάνουν βιβλιοθήκες, προγράμματα υποστήριξης, μεταγλωττιστές αλλά και διεπαφές εφαρμογών (API) και έχουν ως στόχο την συγκέντρωση όλων των στοιχείων που χρειάζονται για την ολοκλήρωση ενός πρότζεκτ.

Ένα τέτοιο πλαίσιο εργαλείων είναι και αυτό που χρησιμοποιήσαμε στην συγκεκριμένη διπλωματική εργασία, το Ionic Framework.

Το Ionic είναι ένα HTML5 πλαίσιο ανάπτυξης εφαρμογών που παρέχει εργαλεία και υπηρεσίες για την κατασκευή υβριδικών εφαρμογών για κινητά χρησιμοποιώντας τεχνολογίες ιστού όπως HTML5, CSS, JS, SASS κλπ. Η αρχική του έκδοση κυκλοφόρησε το 2013 από μια startup εταιρία, την Drifty Co., και χτίστηκε με τη χρήση AngularJS και Apache Cordova τα οποία και θα επεξηγήσουμε αργότερα. Το Ionic χειρίζεται όλες τις αλληλεπιδράσεις, τη λειτουργικότητα και στοιχεία της αίσθησης και της διεπαφής χρήστη(UI) που χρειάζεται η εφαρμογή, όπως πλήκτρα, καρτέλες, λίστες κλπ για να δώσει στην εφαρμογή μια ωραία εμφάνιση.

Το Ionic είναι το δημοφιλέστερο framework για ανάπτυξη υβριδικών εφαρμογών με πάνω από 5 εκατομμύρια προγραμματιστές να το προτιμούν κυρίως λόγω του εξαιρετικού documentation που διαθέτει και της πολύ ενεργής κοινότητας και φόρουμ. Άλλα χαρακτηριστικά που το καθιστούν ελκυστικό είναι:

-Βοηθητικό πρόγραμμα γραμμής εντολών(CLI) που διευκολύνει την εκτέλεση εντολών εκκίνησης, κατασκευής, εκτέλεσης και προσομοίωσης εφαρμογών.

-Ionic Dev Application: μια δωρεάν εφαρμογή που διευκολύνει να τρέξουμε και να τεστάρουμε την εφαρμογή που προγραμματίζουμε απευθείας στη συσκευή iOS, Android ή Windows Phone αρκεί απλά να βρισκόμαστε στο ίδιο δίκτυο με τον υπολογιστή(μέσω wifi). Το Dev App έχει ενσωματωμένα πολλά Plugins οπότε δεν χρειάζονται ξεχωριστή εγκατάσταση.

-Δημιουργία εικόνων και splash screens για όλα τα μεγέθη συσκευών

-Live reload: όποιες αλλαγές κώδικα και αν γίνονται υπάρχει άμεση ενημέρωση της εφαρμογής, ακόμα και όταν εκτελείται απευθείας στη συσκευή.

-Ionic icons: Προσφορά πάνω από 800 έτοιμων εικονιδίων (MIT Licensed)

Δεδομένου ότι το Ionic είναι ένα πλαίσιο HTML5, χρειάζεται και ένα εγγενές περιτύλιγμα (native wrapper) όπως το Cordova ή PhoneGap για να τρέξει ως εγγενής εφαρμογή σε μια συσκευή.

## 5.2 Apache Cordova



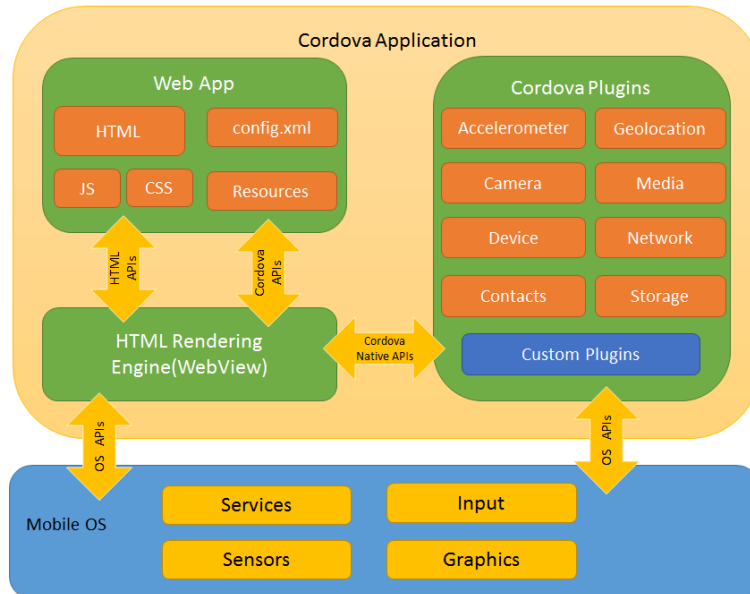
Το Apache Cordova είναι ένα εγγενές περιτύλιγμα (native wrapper) ανοιχτού κώδικα για την κατασκευή λογισμικού σε υπολογιστές και εφαρμογές σε κινητά τηλέφωνα το οποίο βασίζεται σε τεχνολογίες web όπως HTML, CSS, JavaScript.

Αρχικά δημιουργήθηκε το 2009 από την εταιρία Nitobi και είχε την ονομασία Phonegap. Το 2011 Adobe Systems εξαγόρασε την Nitobi και έτσι απέκτησε τα δικαιώματα του Phonegap και αργότερα κυκλοφόρησε μια έκδοση ανοιχτού κώδικα που ονόμασε Apache Cordova. Το Apache Cordova επιτρέπει στους προγραμματιστές λογισμικού να δημιουργούν υβριδικές εφαρμογές για κινητές συσκευές χρησιμοποιώντας τεχνολογίες web αντί να βασίζονται σε API συγκεκριμένης πλατφόρμας, όπως αυτά του Android, i-OS.

Το Apache Cordova ουσιαστικά παρέχει μια γέφυρα στον κώδικα JavaScript για να μπορεί να κάνει κλήση στις εγγενείς λειτουργίες της εφαρμογής (όπως κάμερα, ακουστικό, μικρόφωνο, μπαταρία κλπ.) μέσω μιας μεγάλης λίστας από πρόσθετα plug-ins που προσφέρει.

Το Cordova στηρίζει την ανάπτυξη εφαρμογών σε όλα σχεδόν τα λειτουργικά συστήματα φορητών συσκευών. Οι εφαρμογές που δημιουργούνται με τη χρήση του μπορεί να υποστηρίζονται από πολλές κινητές συσκευές, χωρίς να χρειάζονται διαφορετικές παραμετροποιήσεις για το κάθε ξεχωριστό λειτουργικό με αποτέλεσμα να εξοικονομείτε πολύτιμο χρόνο.

Οπότε καταλαβαίνουμε ότι το Apache Cordova σε συνδυασμό με ένα περιβάλλον UI όπως αυτό που προσφέρει το Ionic Framework είναι ένα πολύ ισχυρό εργαλείο για την ανάπτυξη υβριδικών εφαρμογών.



Εικόνα 5: Αρχιτεκτονική του Apache Cordova

### 5.3 Node.JS



Η επεκτασιμότητα και η καθυστέρηση είναι βασικοί δείκτες απόδοσης για τους διακομιστές ιστού (web servers). Η διατήρηση του χρόνου καθυστέρησης απάντησης μιας αίτησης του χρήστη σε λογικά επίπεδα είναι αρκετά δύσκολη. Το Node.js είναι ένα περιβάλλον εκτέλεσης JavaScript που επιτυγχάνει τη μείωση του χρόνου απόκρισης και απάντησης υιοθετώντας μια προσέγγιση "μη αποκλεισμού" για την εξυπηρέτηση αιτημάτων. Με άλλα λόγια, το Node.js δεν ξοδεύει χρόνο ή πόρους του συστήματος για την αναμονή των αιτήσεων εισόδου / εξόδου.

Παραδοσιακά για τη δημιουργία ενός διακομιστή ιστού (web server), για κάθε εισερχόμενο αίτημα ή σύνδεση ο διακομιστής δημιουργεί ένα νέο νήμα εκτέλεσης και δημιουργεί μια νέα διαδικασία για να χειριστεί το αίτημα και να στείλει πίσω μια απάντηση. Εννοιολογικά, αυτό έχει νόημα αλλά στην πράξη, σε περίπτωση μεγάλου αριθμού νημάτων, θα προκληθεί υπερφόρτωση και επιβάρυνση του συστήματος. Το Node.js επιτυγχάνει με τη χρήση συμβάντων (events) που προσφέρει η JavaScript να πετύχει μια ασύγχρονη επικοινωνία μεταξύ των υπολογιστικών πόρων. Για παράδειγμα όταν ένας περιηγητής ιστού (web browser) φορτώσει πλήρως ένα αρχείο, όταν ένας χρήστης πατήσει κάποιο κουμπί στην οθόνη της συσκευής του ή

ολοκληρώνεται ένα αίτημα AJAX, τα συμβάντα αυτά πυροδοτούν μια συγκεκριμένη επανάκληση (callback). Αυτό με τη σειρά του επιτρέπει τη ροή του κώδικα χωρίς να αφήνει ανενεργό τον επεξεργαστή προκειμένου να εκτελέσει μια λειτουργία όπως για παράδειγμα την ανάγνωση αρχείων από το δίσκο. Αυτή η προσέγγιση με λειτουργίες επανάκλησης (callbacks) απαιτεί λιγότερη μνήμη για να χειριστεί τις περισσότερες διασυνδέσεις από τις περισσότερες ανταγωνιστικές αρχιτεκτονικές που κλιμακώνονται με νήματα, συμπεριλαμβανομένου του Apache HTTP Server, των διαφόρων διακομιστών εφαρμογών Java, των IIS και ASP.NET και του Ruby on Rails.

#### 5.4 Visual Studio Code



Το Visual Studio Code είναι ένας επεξεργαστής πηγαίου κώδικα (source code editor) που αναπτύχθηκε από την Microsoft και είναι διαθέσιμος δωρεάν για τα περισσότερα λειτουργικά (Windows, Linux, OS X). Είναι ένας από τους δημοφιλέστερους editors μιας και προσφέρει μια πληθώρα λειτουργιών όπως υποστήριξη για εντοπισμό σφαλμάτων (debugging), ενσωματωμένο έλεγχο Git, επισήμανση συντακτικού (syntax highlighting) για πάρα πολλές γλώσσες προγραμματισμού, έξυπνο autocomplete κώδικα, και snippets. Είναι επίσης εξαιρετικά προσαρμόσιμο δηλαδή οι χρήστες έχουν τη δυνατότητα να αλλάζουν themes και να χρησιμοποιούν συντομεύσεις πληκτρολογίου. Στον παρακάτω πίνακα αναφέρουμε ενδεικτικά τις σημαντικότερες λειτουργίες του για μερικές από τις γλώσσες προγραμματισμού που υποστηρίζει.

Λειτουργίες	Γλώσσα προγραμματισμού
Επισήμανση συντακτικού	C, C#, C++, CSS, CoffeScript, HTML, JSON, JAVA, JavaScript, Objective-C, PHP, Pearl, Python, R, Ruby, SQL, Sass, Swift, TypeScript, Visual Basic, XML
Snippets	CSS, HTML, JavaScript, JSON, Less, Sass, Type-Script
Εντοπισμός σφαλμάτων	-JavaScript και TypeScript για πρότζεκτ Node.js -C, C#, C++ για Windows, Linux και OS X -Python με εγκατάσταση του Python Plug-in

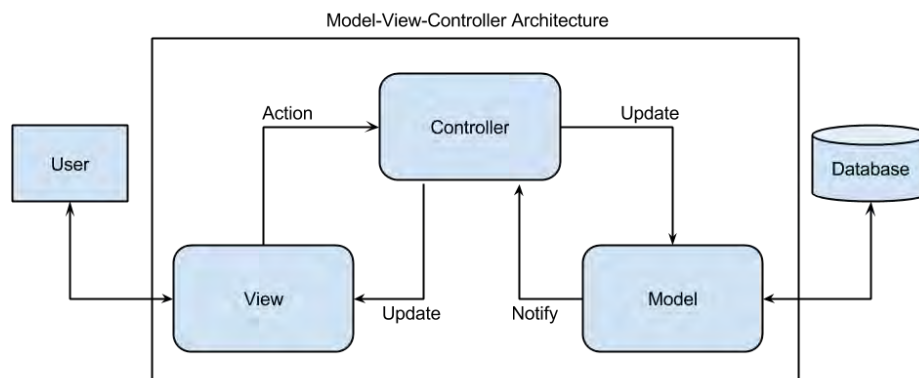
**Πίνακας 4: Λειτουργίες VS Code και γλώσσες προγραμματισμού όπου διατίθενται**

## 5.5 AngularJS

Το AngularJS είναι ένα πλαίσιο ανάπτυξης web ανοιχτού κώδικα βασισμένο σε JavaScript, το οποίο δημιουργήθηκε το 2009 και συντηρείται κυρίως από την Google και από μια μεγάλη κοινότητα ατόμων και εταιριών για την αντιμετώπιση πολλών από τις προκλήσεις που αντιμετωπίζονται κατά την ανάπτυξη εφαρμογών μιας σελίδας. Επιτυγχάνει την επέκταση της λειτουργικότητας του HTML χρησιμοποιώντας εκφράσεις και την εισαγωγή ιδεών όπως η δυναμική σύνδεση δεδομένων που υπάρχουν και σε άλλες γλώσσες όπως η Java, .NET και πλαίσια όπως η Spring. Όλα αυτά οδηγούν σε μειωμένο κώδικα, ενώ παράλληλα δημιουργούν πιο δυναμικές ιστοσελίδες.

Τα συστατικά στοιχεία της JavaScript συμπληρώνουν το Apache Cordova, το πλαίσιο που αναφέραμε ότι χρησιμοποιείται για την ανάπτυξη υβριδικών εφαρμογών για κινητά. Στόχος της είναι να απλοποιήσει τόσο την ανάπτυξη όσο και τη δοκιμή τέτοιων εφαρμογών, παρέχοντας ένα πλαίσιο με αρχιτεκτονική μοντέλου-προβολής-ελεγκτή (MVC).

Η συγκεκριμένη αρχιτεκτονική έχει πολύ διακριτές έννοιες, τις οποίες μπορούμε να διαχωρίσουμε και να καταλάβουμε καλύτερα επεξηγώντας τις παρακάτω.



Εικόνα 6: Αρχιτεκτονική MVC

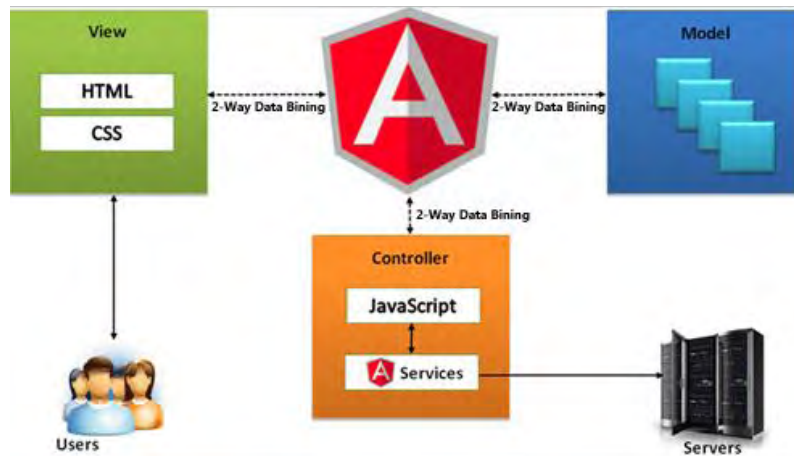
-Μοντέλο(Model): Στην AngularJS σαν μοντέλο αναφερόμαστε σε ένα αντικείμενο(Object) Javascript. Μπορεί να είναι απλά μια μεταβλητή όπως character, boolean, string ή ένας σύνθετος τύπος Object. Στην AngularJS το model δεν είναι ούτε κάποια κλάση ούτε κάποια μέθοδος getter ή setter. Είναι το χαμηλότερο επίπεδο του σχεδίου που είναι υπεύθυνο για τη διατήρηση των δεδομένων της εφαρμογής. Οι προγραμματιστές που είναι εξοικειωμένοι με το OOP (Object Oriented Programming) θα αισθανθούν εύκολη τη χρήση του μοντέλου.



-Προβολή(View): Σαν view ονομάζουμε οτιδήποτε βλέπει ο τελικός χρήστης στην οθόνη της συσκευής του. Με τη χρήση εκφράσεων της Angular που υποστηρίζουν την αμφίδρομη δέσμευση και εμφάνιση δεδομένων, η προβολή μπορεί και ενημερώνεται αυτόματα αν το μοντέλο αλλάξει από τον ελεγκτή.

-Ελεγκτής(Controller): Σαν ελεγκτή αναφερόμαστε στον κώδικα που αποκτά δυναμικά τα δεδομένα και δημιουργεί το περιεχόμενο της HTML. Ο ελεγκτής χειρίζεται όλες τις αλληλεπιδράσεις μεταξύ μοντέλου και προβολής και εκτελεί την λογική των εφαρμογών.

Με λίγα λόγια ο χρήστης μπορεί να αλληλεπιδρά με την προβολή του html (View), να αλλάξει τα δεδομένα (Model) και να καλεί τον ελεγκτή (Controller). Ο ελεγκτής τροποποιεί το μοντέλο, αλληλεπιδρά με το διακομιστή(server) και εκτελεί λειτουργίες CRUD(Create, Read, Update, Delete) και Lookup στα δεδομένα. Το AngularJS ανιχνεύει τυχόν αλλαγές στο μοντέλο και ενημερώνει την προβολή.



Εικόνα 7: Μοντέλο αρχιτεκτονικής AngularJS

## 5.6 Google Firebase



Η Firebase είναι μια πλατφόρμα ανάπτυξης εφαρμογών για κινητά και διαδικτυακές εφαρμογές που αναπτύχθηκε από την Firebase, Inc. το 2011, και στη συνέχεια εξαγοράστηκε από την Google το 2014. Η συγκεκριμένη πλατφόρμα προσφέρει μια πληθώρα υπηρεσιών και δυνατοτήτων με στόχο την απελευθέρωση των προγραμματιστών από τη διαχείριση

πολύπλοκων βάσεων δεδομένων ή συγγραφή ξεχωριστών API. Το Firebase είναι ο διακομιστής (server), το API και το datastore, όλα αυτά γραμμένα με τέτοιο γενικό τρόπο ώστε κάνουν εύκολη την τροποποίησή τους και κάλυψη των περισσότερων αναγκών που δημιουργούνται στη server-side δημιουργία της εφαρμογής. Θα περιγράψουμε μερικές από τις σημαντικότερες υπηρεσίες που προσφέρονται.

-Βάση δεδομένων σε πραγματικό χρόνο(Real Time Database): Τα δεδομένα σε πραγματικό χρόνο είναι ο δρόμος του μέλλοντος. Τίποτα δεν συγκρίνεται με αυτό. Οι περισσότερες βάσεις δεδομένων απαιτούν να πραγματοποιούμε κλήσεις HTTP για να λαμβάνουμε και να συγχρονίζουμε τα δεδομένα μας. Οι περισσότερες βάσεις δεδομένων μας δίνουν δεδομένα μόνο όταν το ζητάμε. Όταν συνδέουμε την εφαρμογή μας στο Firebase, δεν συνδεόμαστε μέσω κανονικής HTTP αλλά μέσω ενός WebSocket. Τα WebSockets είναι πολύ ταχύτερα από το HTTP. Όλα τα δεδομένα μας συγχρονίζονται αυτόματα μέσω αυτού του ενιαίου WebSocket τόσο γρήγορα όσο το δίκτυο του πελάτη που τα μεταφέρει. Με το που γίνει μια αλλαγή στα δεδομένα η Firebase στέλνει νέα ενημέρωση στους clients. Και όταν ένας client αποθηκεύσει μια αλλαγή στα δεδομένα, όλοι οι συνδεδεμένοι clients λαμβάνουν τα ενημερωμένα δεδομένα σχεδόν αμέσως.

- Αποθήκευση αρχείων(File Storage): Η Firebase παρέχει έναν απλό τρόπο αποθήκευσης δυαδικών αρχείων, τις περισσότερες φορές εικόνες, στο Google Cloud Storage του client. Επίσης το Firebase Storage διαθέτει το δικό του σύστημα κανόνων ασφαλείας για την προστασία του Google Cloud, παρέχοντας παράλληλα λεπτομερή δικαιώματα εγγραφής στους πιστοποιημένους clients.

-Διδικασία πιστοποίησης(Authentication): . Η δημιουργία ενός προσαρμοσμένου συστήματος ελέγχου ταυτότητας είναι πάρα πολύ δύσκολος. Το Firebase auth έχει ενσωματωμένο σύστημα ελέγχου ταυτότητας ηλεκτρονικού ταχυδρομείου / κωδικού πρόσβασης και υποστηρίζει επίσης το OAuth2 για Google, Facebook, Twitter και GitHub. Το Firebase Auth ενσωματώνεται απευθείας στη βάση δεδομένων Firebase, έτσι ώστε εύκολα να μπορούμε να το χρησιμοποιούμε για να ελέγχουμε την πρόσβαση στα δεδομένα μας.

-Firebase Hosting: Μια στατική και δυναμική υπηρεσία φιλοξενίας ιστοσελίδων, στατικών αρχείων όπως CSS, HTML, JavaScript και άλλων αρχείων, καθώς και δυναμική υποστήριξη Node.js. Η υπηρεσία παραδίδει αρχεία μέσω δικτύου παροχής περιεχομένου (CDN) μέσω κρυπτογράφησης HTTP Secure (HTTPS) και Secure Sockets Layer (SSL).

-Firebase Test Lab για Android: Μια υπηρεσία που παρέχει μια υποδομή βασισμένη σε cloud για τη δοκιμή εφαρμογών Android. Οι προγραμματιστές μπορούν να ξεκινήσουν τη δοκιμή των εφαρμογών τους σε μια μεγάλη ποικιλία συσκευών. Τα αποτελέσματα των δοκιμών - συμπεριλαμβανομένων των αρχείων καταγραφής, των βίντεο και των στιγμιότυπων οθόνης - διατίθενται στην κονσόλα του Firebase. Ακόμη και αν ένας προγραμματιστής δεν έχει γράψει κανένα κώδικα δοκιμής για την εφαρμογή του, το Test Lab μπορεί να τεστάρει αυτόματα την εφαρμογή.

-Firebase Analytics: Μια υπηρεσία μέτρησης που παρέχει πληροφορίες για τη χρήση της εφαρμογής και την αφοσίωση των χρηστών.

Οι παραπάνω υπηρεσίες καθώς και αρκετές ακόμα προσφέρονται μέσω της κονσόλας του Firebase και σε συνδυασμό με τους εύκολους οδηγούς, tutorials καθώς και του εύχρηστου documentation που παρέχει καθιστούν τη χρήση του μια εξαιρετική επιλογή για την εφαρμογή μας.

## 5.7 Τεχνολογίες Παγκόσμιου Ιστού

### 5.7.1 HTML5



Η HTML (HyperText Markup Language) είναι μια γλώσσα σήμανσης που χρησιμοποιείται για τη σύνταξη του περιεχομένου μιας σελίδας Παγκόσμιου Ιστού.

Η ομάδα Web Hypertext Application Technology Working Group (WHATWG) άρχισε να δουλεύει την συγκεκριμένη έκδοση τον Ιούνιο του 2004 με το όνομα Web Applications 1.0.

Η HTML5 προοριζόταν για αντικατάσταση της HTML 4.01, της XHTML 1.0, και της DOM Level 2 HTML. Στοχεύει στη μείωση της ανάγκης για ιδιόκτητα plug-in

και βαριές διαδικτυακές εφαρμογές όπως το Adobe Flash, το Microsoft Silverlight, το Apache Pivot, και η Sun JavaFX.

Το πρότυπο HTML5 υιοθετήθηκε ως αρχικό βήμα για τις εργασίες της νέας ομάδας εργασίας HTML του W3C το 2007. Αυτή η ομάδα δημοσίευσε το Πρώτο Δημόσιο Working Draft του προτύπου τον Ιανουάριο του 2008. Η τελευταία και πιο σταθερή έκδοση δημοσιεύθηκε τον Οκτώβριο του 2014 από την Κοινοπραξία World Wide Web (W3C) και αποσκοπεί στη βελτίωση της γλώσσας με την υποστήριξη των πιο πρόσφατων πολυμέσων, διατηρώντας ταυτόχρονα συσκευές και περιηγητές, εύκολα αναγνώσιμους από τον άνθρωπο και κατανοητά από υπολογιστές.

Το HTML5 περιλαμβάνει λεπτομερή μοντέλα επεξεργασίας για την ενθάρρυνση περισσότερων διαλειτουργικών υλοποιήσεων. Επεκτείνει, βελτιώνει και εξορθολογίζει τη διαθέσιμη σήμανση για έγγραφα και εισάγει διεπαφές προγραμματισμού και API για σύνθετες εφαρμογές ιστού. Για τους ίδιους λόγους, το HTML5 είναι επίσης πολύ χρήσιμο για υβριδικές εφαρμογές κινητών συσκευών, επειδή περιλαμβάνει χαρακτηριστικά γνωρίσματα σχεδιασμένα για συσκευές χαμηλής ισχύος.

Στο HTML χρησιμοποιούσαμε ένα σύνολο ετικετών για τη συγγραφή του κώδικα. Στο HTML5 προστέθηκαν νέα στοιχεία και ετικέτες για την καλύτερη μορφοποίηση κειμένου (όπως header, footer, article κ.α.) αλλά και ετικέτες για την παρουσίαση πολυμέσων (όπως audio, video, track). Επίσης παρέχει στους προγραμματιστές νέους μηχανισμούς όπως:

- Πρόσβαση στη γεωγραφική τοποθεσία του χρήστη (μέσω JS Geographical Location)
- Χρήση βάσεων δεδομένων web SQL, πρόσβαση στο χώρο αποθήκευσης της συσκευής και της προσωρινής μνήμης των εφαρμογών μέσω Javascript.
- Χρήση web socket δηλαδή πρωτοκόλλου επικοινωνίας που χρησιμοποιεί κανάλια αμφίδρομης επικοινωνίας μέσω σύνδεσης TCP και επιτρέπει πλήρη επικοινωνία μεταξύ πελατών και διακομιστών (client-server connection).
- Χρήση JS web worker API το οποίο επιτρέπει στις διεπαφές JavaScript και Browser να εκτελούνται σε ξεχωριστά νήματα(threads).

## 5.7.2 Typescript



Η TypeScript είναι μια γλώσσα προγραμματισμού ανοικτού κώδικα που αναπτύχθηκε και συντηρείται από τη Microsoft. Πρόκειται για ένα αυστηρό συντακτικό υπερσύνολο της JavaScript και προσθέτει προαιρετική στατική πληκτρολόγηση στη γλώσσα. Το TypeScript μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών JavaScript για εκτέλεση και από την πλευρά του πελάτη(client) και από την πλευρά του διακομιστή(server). Η TypeScript έχει σχεδιαστεί για την ανάπτυξη μεγάλων εφαρμογών και μεταγλωττίζεται σε JavaScript. Δεδομένου ότι η TypeScript είναι ένα υπερσύνολο της JavaScript, τα υπάρχοντα προγράμματα JavaScript είναι επίσης έγκυρα σαν προγράμματα TypeScript.

Η TypeScript υποστηρίζει αρχεία που μπορούν να περιέχουν πληροφορίες για υπάρχουσες βιβλιοθήκες JavaScript, όπως τα header αρχεία της C++ μπορούν να περιγράψουν τη δομή των υπάρχοντων αρχείων αντικειμένων. Αυτό επιτρέπει σε άλλα προγράμματα να χρησιμοποιούν τις τιμές που ορίζονται στα αρχεία Typescript σαν να ήταν στατικές. Αυτό το static typing σύστημα επιτρέπει την πρόσθεση τύπων μεταβλητών, λειτουργιών και ιδιοτήτων το οποίο βοηθάει τον μεταγλωττιστή να εμφανίζει προειδοποιήσεις σχετικά με πιθανά σφάλματα στον κώδικα, πριν από την εκτέλεση μιας εφαρμογής. Οι τύποι μεταβλητών βοηθούν επίσης κατά τη χρήση βιβλιοθηκών και πλαισίων, καθώς επιτρέπουν στους προγραμματιστές να γνωρίζουν ακριβώς ποιον τύπο API δεδομένων περιμένει. Το βασικό πράγμα που πρέπει να θυμάστε σχετικά με το παραπάνω σύστημα είναι ότι είναι προαιρετικό. Η TypeScript δεν αναγκάζει τους προγραμματιστές να προσθέσουν τύπους που δεν επιθυμούν να προσθέσουν. Ωστόσο, καθώς μια εφαρμογή γίνεται μεγαλύτερη και πιο περίπλοκη, οι τύποι μπορούν να αποτελέσουν μεγάλο πλεονέκτημα.

Επίσης ένα από τα μεγαλύτερα πλεονεκτήματα της TypeScript είναι η έξυπνη ολοκλήρωση κώδικα μέσω της λειτουργίας IntelliSense. Το Intellisense παρέχει ενεργές συμβουλές όπως ολοκλήρωση κώδικα, πληροφορίες παραμέτρων, γρήγορες πληροφορίες και λίστες για τον κώδικα που προστίθεται. Τα δημοφιλέστερα IDE που είναι διαθέσιμα σήμερα έχουν υποστήριξη για την ολοκλήρωση κώδικα, συμπεριλαμβανομένων των VS Code, Atom, WebStorm, Sublime αλλά ακόμη και επεξεργαστές γραμμής εντολών, όπως το Vim.

### 5.7.3 CSS3



Η CSS (Cascading Style Sheet) είναι μια γλώσσα που χρησιμοποιείται για την μορφοποίηση των ιστοσελίδων που είναι γραμμένη σε μια γλώσσα σήμανσης όπως HTML ή XHTML. Η CSS έχει ως στόχο το διαχωρισμό του οπτικού από το δομικό κομμάτι του κώδικα. Ο διαχωρισμός της HTML από το CSS διευκολύνει τη διατήρηση ιστότοπων, την κοινή χρήση φύλλων στυλ στις σελίδες και την προσαρμογή των σελίδων σε διαφορετικά περιβάλλοντα.

Με τη CSS μπορούμε να χειριστούμε την εμφάνιση και την αίσθηση που δίνει μια ιστοσελίδα. Χρησιμοποιώντας τη CSS, μπορούμε να ελέγξουμε το χρώμα του κειμένου, το στυλ και το μέγεθος γραμματοσειρών, την απόσταση μεταξύ των παραγράφων, τον τρόπο μεγέθυνσης και τοποθέτησης στηλών, τις εικόνες φόντου ή τα χρώματα που χρησιμοποιούνται, τα σχέδια σχεδίασης, καθώς και τις διαφορές στην εμφάνιση της σελίδας σε διαφορετικά μεγέθη οθόνης συσκευών. Η CSS μας εξοικονομεί πολύτιμο χρόνο καθώς προσφέρει:

- Επαναχρησιμοποίηση κώδικα για αρκετές σελίδες, καθώς αρκεί να ορίσουμε ένα στυλ για κάθε στοιχείο HTML και να το εφαρμόσουμε σε όσες ιστοσελίδες επιθυμούμε

- Πιο γρήγορη φόρτωση σελίδων καθώς δεν χρειάζεται να γράφετε ξεχωριστά HTML tag κάθε φορά. Απλά γράφουμε έναν κανόνα CSS μιας ετικέτας και τον εφαρμόζουμε σε όλες τις εμφανίσεις αυτής της ετικέτας. Έτσι λιγότερο κώδικας σημαίνει ταχύτερος χρόνος απόκρισης και ανανέωσης της σελίδας.

- Εύκολη συντήρηση του κώδικα μιας και για να γίνει μια καθολική αλλαγή στην εμφάνιση του κώδικα, απλά αλλάζουμε το στυλ και όλα τα στοιχεία σε όλες τις ιστοσελίδες θα ενημερωθούν αυτόματα.

Η κάθε μια από τις εκδόσεις του CSS (CSS1, CSS2, CSS3) προσθέτει νέες λειτουργίες και ανανεώνει τις ήδη υπάρχουσες της προηγούμενης έκδοσης.



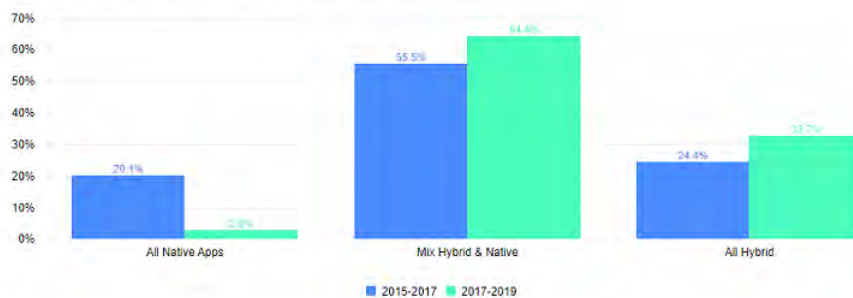
Εικόνα 8: CSS Modules

### 5.8 Έρευνα προγραμματιστών Ionic 2017

Τώρα που ολοκληρώσαμε την παρουσίαση των εργαλείων και τεχνολογιών που χρησιμοποιήσαμε εμείς θα ήταν χρήσιμη και η παρουσίαση μιας έρευνας, που έκανε η ομάδα του Ionic, σχετικά με τα εργαλεία και τις τεχνολογίες που χρησιμοποιεί η κοινότητα των προγραμματιστών Ionic για την ανάπτυξη των εφαρμογών του καθώς και για τους τύπους εφαρμογών που κατασκευάζουν. Περισσότεροι από 13.000 προγραμματιστές πήραν μέρος στην έρευνα. Τα αποτελέσματα αποκαλύπτουν μερικές πολύ ενδιαφέρουσες τάσεις.

Πριν από δύο χρόνια, το 20% των ερωτηθέντων προγραμματιστών δούλευαν αποκλειστικά με εγγενή εργαλεία αντί για υβριδικά. Σήμερα ο αριθμός αυτός έχει μειωθεί τρομακτικά και φτάνει μόλις το 2,9%. Ομοίως, ο αριθμός των υβριδικών εφαρμογών αυξάνεται ως μερίδιο της συνολικής ανάπτυξης εφαρμογών. Στα επόμενα δύο χρόνια, το 32,7% των ερωτηθέντων προγραμματιστών αναμένουν να εγκαταλείψουν εντελώς τη εγγενή ανάπτυξη εφαρμογών υπέρ των υβριδικών. Η ευρύτερη τάση είναι ότι η υβριδική ανάπτυξη κερδίζει συνεχώς μερίδιο στην αγορά, ενώ η εγγενή προσέγγιση μειώνεται. (Εικόνα 9)

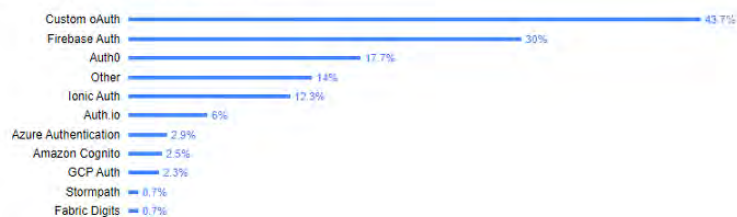
In the past two years, what percentage of your apps were hybrid (mix of web and native code)?  
 What percentage do you expect in the next two years?



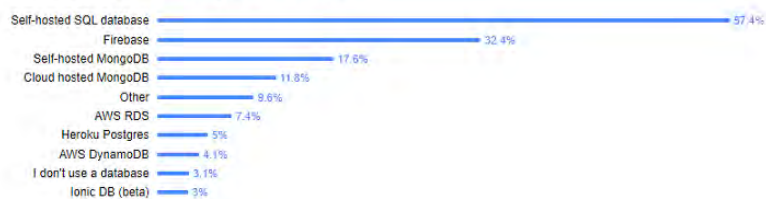
Εικόνα 9

Πέραν των self-hosted βάσεων δεδομένων SQL, σαν εξωτερική υπηρεσία hosting η Firebase της Google είναι ο ξεκάθαρος νικητής. Για ειδοποιήσεις push, το Firebase χρησιμοποιήθηκε από το 41,3% των προγραμματιστών ιονιc όπως επίσης πολύ ψηλά στις προτιμήσεις ήταν και άλλες υπηρεσίες του όπως το Firebase Auth. (Εικόνα 10)

How do you manage user authentication?



What database(s) do you use for your backend?

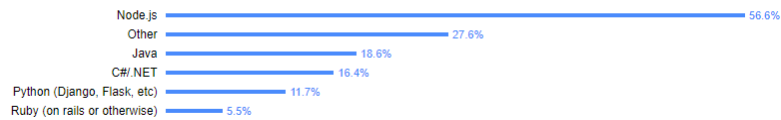


Εικόνα 10



Το Node.js είναι η κυρίαρχη επιλογή όταν πρόκειται για τεχνολογία backend με το 56,6% των προγραμματιστών να το προτιμούν. (Εικόνα 11)

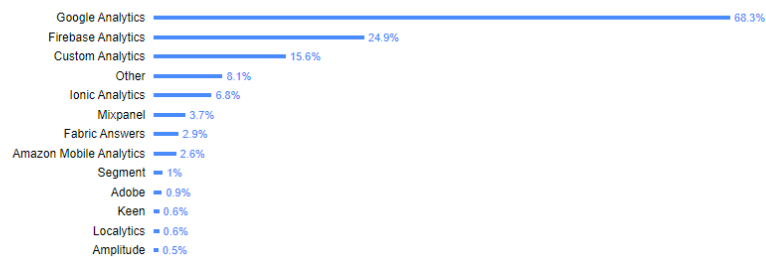
What backend technology does your app run on?



Εικόνα 11

Στον χώρο των αναλύσεων η Google κατέχει την πρωτιά, με το Google Analytics να καταλαμβάνει ποσοστό 68,3% ενώ το Firebase Analytics να ακολουθεί με 24,9%. (Εικόνα 12)

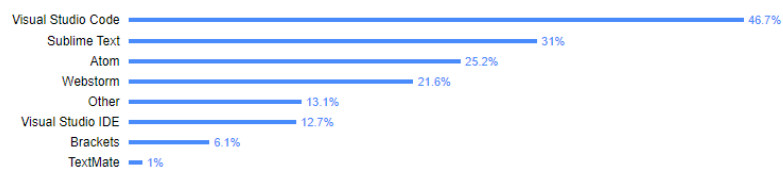
What service do you use for Analytics?



Εικόνα 12

Στα εργαλεία επεξεργασίας και ανάπτυξης κώδικα η Microsoft φαίνεται να έχει κερδίσει την εμπιστοσύνη του κοινού, με τα Visual Studio Code και Visual Studio IDE. Πάνω από τους μισούς προγραμματιστές που έλαβαν μέρος στην έρευνα χρησιμοποιεί ένα απ' τα δυο εργαλεία. (Εικόνα 13)

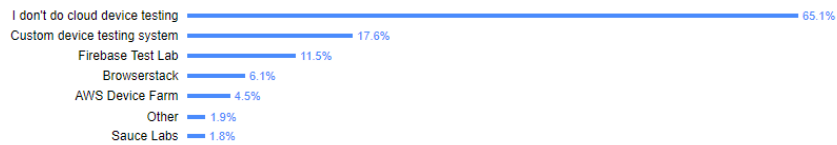
What Text Editor or IDE do you use?



Εικόνα 13

Η δοκιμή συσκευών είναι μια από τις πιο δύσκολες πτυχές της ανάπτυξης εφαρμογών για κινητά. Ακόμη και σε μεγαλύτερες ομάδες, κανείς δεν έχει αρκετές συσκευές για να καλύψει κάθε σενάριο. Μία λύση είναι η χρήση κάποιας πλατφόρμας που μεταφέρει τη διαδικασία του testing στο cloud ώστε να υπάρχει πρόσβαση σε περισσότερες συσκευές. Παρόλα αυτά η συντριπτική πλειοψηφία των προγραμματιστών, το 65,1%, δήλωσε ότι δεν χρησιμοποιεί καμία υπηρεσία ελέγχου συσκευών cloud. Ο χρόνος θα δείξει εάν αυτές οι υπηρεσίες θα κερδίσουν την εμπιστοσύνη τους. (Εικόνα 14)

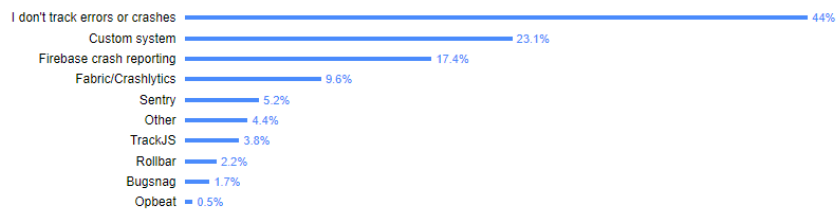
#### How do you test your app in the cloud?



Εικόνα 14

Επίσης ένα εντυπωσιακά μεγάλο ποσοστό των ερωτηθέντων, το 44%, δήλωσε ότι δεν χρησιμοποιούν κάποιο εργαλείο για να εντοπίζουν σφάλματα στις εφαρμογές που κατασκευάζει. Μια πιθανή εξήγηση είναι ότι τα εγγενή εργαλεία εντοπισμού σφαλμάτων είναι λιγότερο αποτελεσματικά όταν πρόκειται για την κατανόηση του web layer των εφαρμογών. (Εικόνα 15)

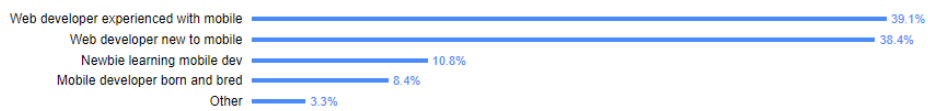
#### How do you track errors and crashes in your app?



Εικόνα 15

Σχεδόν το 80% προγραμματιστών Ionic που έλαβαν μέρος στην έρευνα θεωρούν τον εαυτό τους προγραμματιστές web ενώ ένα 10,8% δήλωσε ότι είναι αρχάριο στην ανάπτυξη εφαρμογών για κινητά. Αυτό δείχνει ότι το Ionic εκπληρώνει την αποστολή του δηλαδή να βοηθήσει τους προγραμματιστές web να συμμετάσχουν στην αναπτυσσόμενη και ευημερούσα αγορά των εφαρμογών για κινητά. (Εικόνα 16)

#### What is your background?



Εικόνα 16

## **6 Παρουσίαση και Εργαλεία ανάπτυξης εφαρμογής**

### **6.1 Στόχος της εφαρμογής**

Πρόκειται για μια υβριδική εφαρμογή εύρεσης “συνταξιδιωτών”. Σαν συνταξιδιώτη εννοείτε ο οποιοσδήποτε χρήστης της εφαρμογής έχει ήδη ταξιδέψει ή πρόκειται να ταξιδέψει στον ίδιο προορισμό(χώρα) με τον συνδεδεμένο στην εφαρμογή χρήστη. Στόχος της εφαρμογής είναι να δώσει τη δυνατότητα στον κάθε χρήστη να μπορεί να ανταλλάξει απόψεις και ιδέες, με βάση τη χώρα, με άλλους χρήστες μέσω ερωτήσεων/απαντήσεων αλλά και να συνομιλήσει με κάποιον χρήστη με τη χρήση προσωπικού chat.

### **6.2 Απαιτήσεις συστήματος**

Οι ελάχιστες απαιτήσεις συστήματος για την εγκατάσταση της εφαρμογής στο κινητό από τους χρήστες, είναι να διαθέτουν συσκευή με έκδοση είτε Android 5.0 είτε iOS 6.0 (ή μεταγενέστερες αυτών)και API 21.

### **6.3 Οδηγίες εγκατάστασης εργαλείων**

Καταρχάς είναι σημαντικό να αναφέρουμε τα εργαλεία και τη διαδικασία εγκατάστασης. Η εφαρμογή δημιουργήθηκε σε λειτουργικό σύστημα Windows 8.1.. Η διαδικασία εγκατάστασης είναι παρόμοια και για τα λειτουργικά iOS και Linux.

1. Εγκατάσταση Android Studio (μέσω της επίσημης σελίδας <https://developer.android.com/studio>) για το τρέξιμο και τον έλεγχο της εφαρμογής σε Virtual Devices που προσφέρει.
2. Εγκατάσταση Node.js (μέσω της επίσημης σελίδας <https://nodejs.org/en/download/>) επιλέγοντας το λειτουργικό που βρισκόμαστε καθώς και την LTS (stable) έκδοση.
3. Εγκατάσταση Visual Studio Code. (μέσω της επίσημης σελίδας <https://code.visualstudio.com/> ) επιλέγοντας πάλι το λειτουργικό που βρισκόμαστε.

Τώρα είμαστε έτοιμοι να ξεκινήσουμε τη διαδικασία εγκατάστασης του Ionic. Ανοίγοντας ένα παράθυρο κονσόλας και χρησιμοποιώντας την εντολή `npm` που

μας προσέφερε ο Node Package Manager μπορούμε να διαχειριστούμε την εγκατάσταση των πακέτων που θα χρειαστούμε.

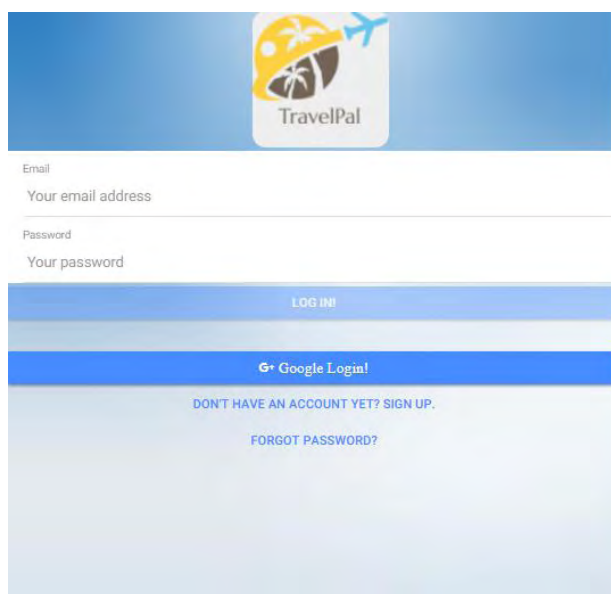
1. Npm install ionic cordova -g  
Εγκατάσταση των πακέτων του ionic και του cordova. Πλέον μπορούμε να χρησιμοποιήσουμε στην κονσόλα και τις εντολές που μας προσφέρουν τα δυο πακέτα.
2. Ionic start <project name> blank για τη δημιουργία ενός νέου project ionic.
3. Ionic cordova platform add android (ή add ios) για τη δημιουργία της εφαρμογής μας για την αντίστοιχη πλατφόρμα.
4. Ionic cordova build android (ή build ios) για τη δημιουργία του αντίστοιχου εκτελέσιμου αρχείου .apk
5. Το συγκεκριμένο αρχείο μπορούμε είτε να το εγκαταστήσουμε σε μια πραγματική συσκευή κινητού τηλεφώνου συνδέοντας το τηλέφωνο με τον υπολογιστή μέσω θύρας USB είτε να την τρέξουμε σε μια εικονική συσκευή (virtual device) στο Android Studio.

#### 6.4 Οδηγίες χρήσης εφαρμογής

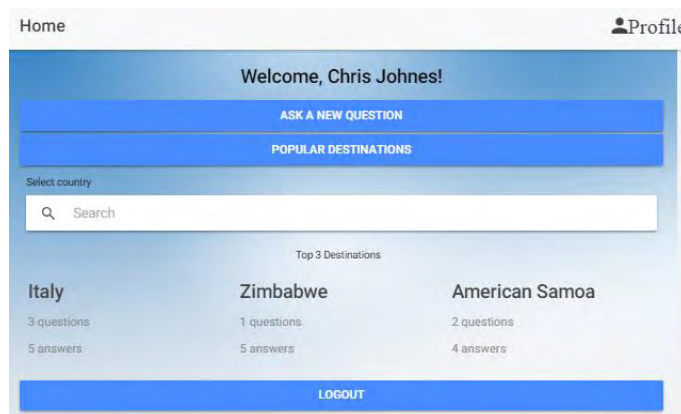
Η εκτέλεση της εφαρμογής είναι πολύ εύκολη. Αφού, εγκατασταθεί η εφαρμογή, ο χρήστης μπορεί να βρει μία συντόμευση της εφαρμογής στο κυρίως μενού της συσκευής του. Αγγίζοντας το εικονίδιο της συντόμευσης εκκινείτε αυτόματα η εφαρμογή.

#### 6.5 Παρουσίαση λειτουργιών της εφαρμογής

Κατά το άνοιγμα της εφαρμογής, ο χρήστης μπορεί να δημιουργήσει ένα λογαριασμό με βάση το email του (sign-up) ή να χρησιμοποιήσει τον ήδη υπάρχων Google+ λογαριασμό του.

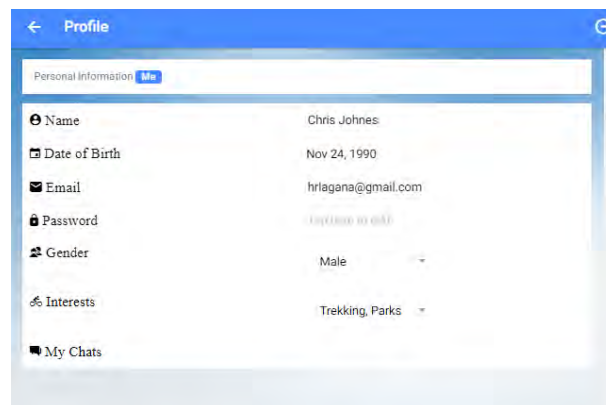


Έπειτα μπορεί με το email και τον προσωπικό κωδικό που έχει επιλέξει να κάνει είσοδο στο λογαριασμό του. Σε περίπτωση απώλειας του κωδικού του παρέχεται η αντίστοιχη επιλογή επαναφοράς του κωδικού. Μετά την είσοδο μεταφέρεται αυτόματα στην αρχική σελίδα όπου του δίνονται κάποιες επιλογές.

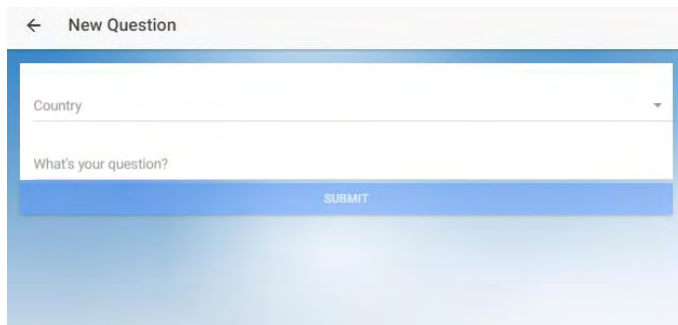


#### 1. Επιλογή Προφίλ Χρήστη

Εδώ ο χρήστης μπορεί να δει και να αλλάξει τις προσωπικές του πληροφορίες όπως: Ονομ/μο, Ημερομηνία Γέννησης, Email, Κωδικό εισόδου, Γένος, Ενδιαφέροντα. Επίσης έχει την δυνατότητα να επιλέξει να δει τις συνομιλίες του με άλλους χρήστες.



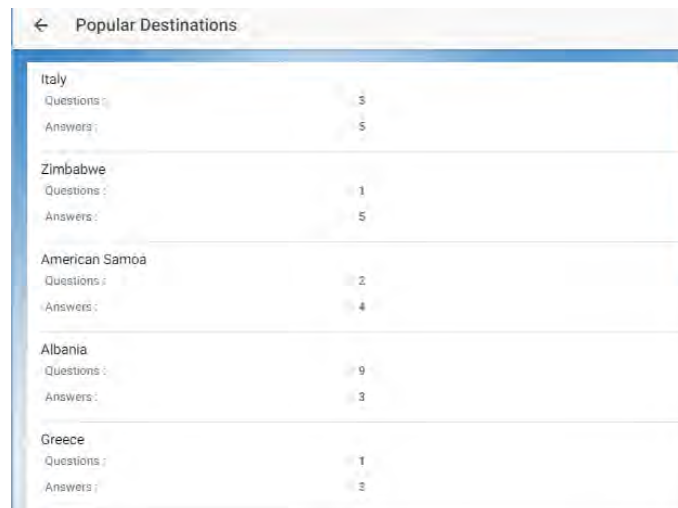
## 2. Δημιουργία νέας ερώτησης



The screenshot shows a mobile application interface for creating a new question. At the top, there is a back arrow and the text 'New Question'. Below this is a form with a 'Country' dropdown menu, a text input field with the placeholder 'What's your question?', and a blue 'SUBMIT' button at the bottom.

Εδώ ο χρήστης έχει τη δυνατότητα να κάνει μία ερώτηση που τον ενδιαφέρει επιλέγοντας και τον προορισμό/χώρα για τον οποίο απευθύνεται.

## 3. Παρουσίαση των Δημοφιλέστερων Προορισμών

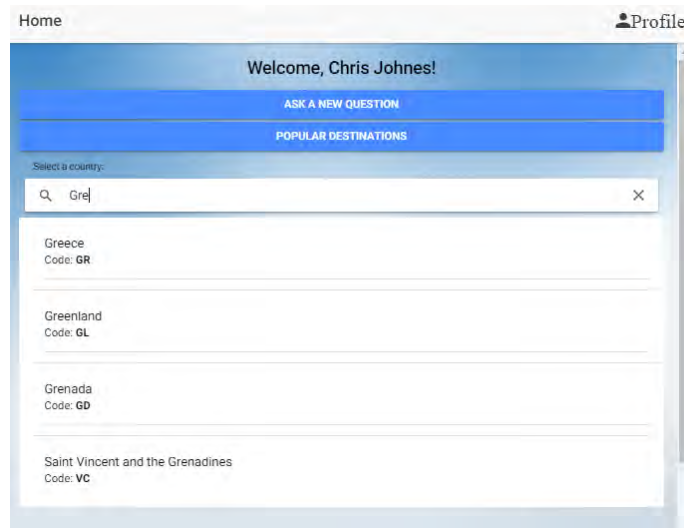


The screenshot shows a mobile application interface for 'Popular Destinations'. It features a list of countries with their respective question and answer counts. The data is as follows:

Country	Questions	Answers
Italy	3	5
Zimbabwe	1	5
American Samoa	2	4
Albania	9	3
Greece	1	2

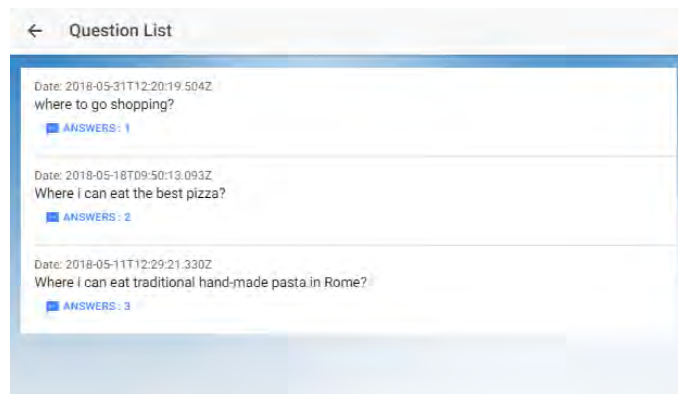
Εμφάνιση μιας λίστας με τους προορισμούς των οποίων οι ερωτήσεις έχουν απαντηθεί τις περισσότερες φορές. Οι προορισμοί μπορούν να επιλεγούν για την παρουσίαση των ερωτήσεων που περιέχουν.

#### 4. Επιλογή Προορισμού



Ο χρήστης έχει τη δυνατότητα να επιλέξει έναν προορισμό μέσω της μπάρας αναζήτησης. Καθώς ο χρήστης πληκτρολογεί του εμφανίζονται οι αντίστοιχες χώρες με την ακολουθία χαρακτήρων που έδωσε.

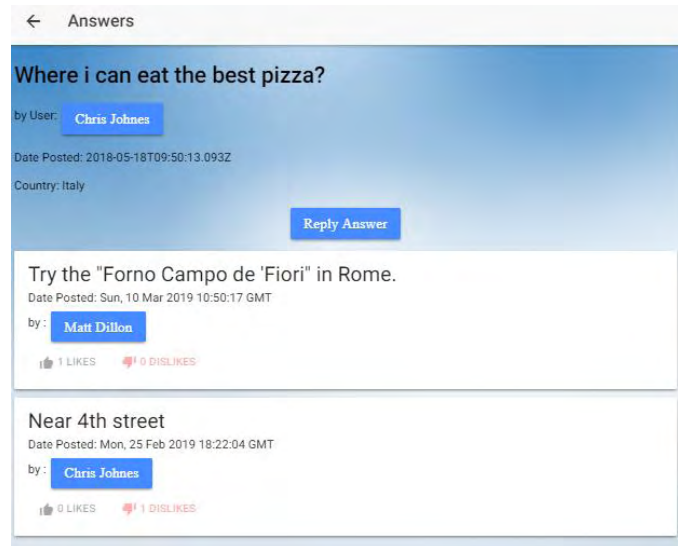
#### 5. Λίστα ερωτήσεων



Μετά την επιλογή προορισμού του εμφανίζεται μια λίστα των ερωτήσεων που απευθύνονται σ' αυτόν τον προορισμό καθώς και κάποιες πληροφορίες για την κάθε ερώτηση (ημερομηνία καταχώρησης και συνολικές απαντήσεις στις ερωτήσεις).

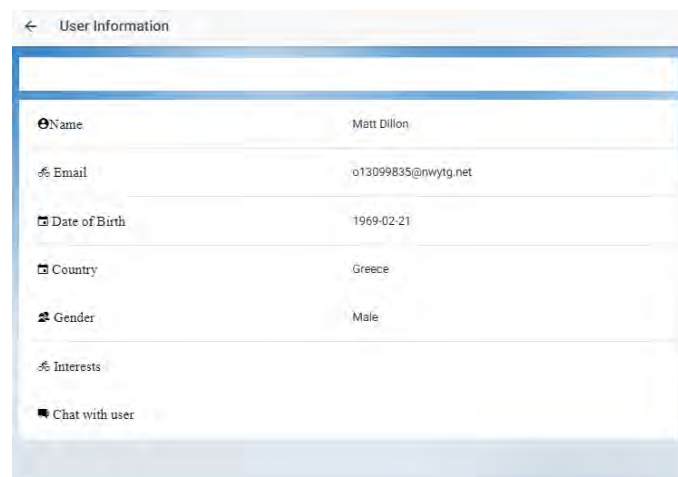


## 6. Επιλογή ερώτησης



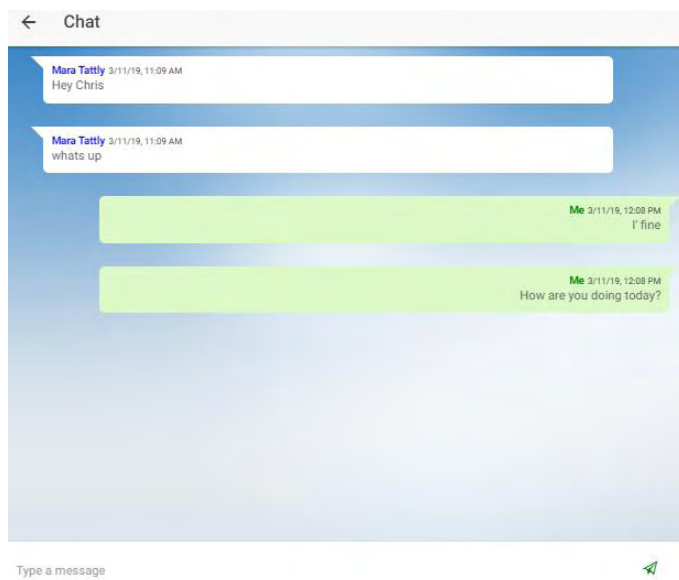
Με την επιλογή μιας ερώτησης εμφανίζονται στο χρήστη πληροφορίες για αυτήν(χρήστης που την καταχώρησε, ημερομηνία, προορισμός) καθώς και όλες οι απαντήσεις τις με τις αντίστοιχες πληροφορίες. Εδώ ο χρήστης μπορεί να κάνει like ή dislike κάποια απάντηση κατά το δοκούν.

## 7. Επιλογή χρήστη



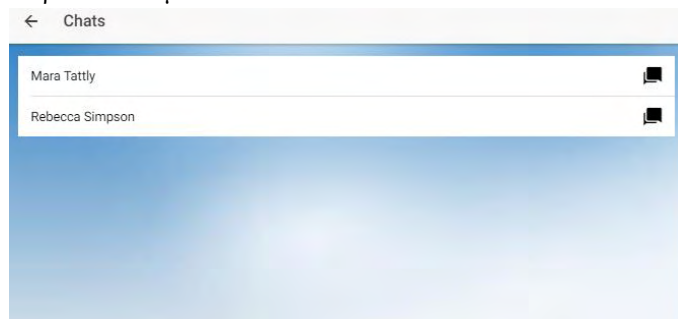
Με την επιλογή του ονόματος ενός άλλου χρήστη δίνονται οι πληροφορίες για τον συγκεκριμένο χρήστη καθώς και οι δυνατότητα συνομιλίας με αυτόν μέσω προσωπικού chat.

## 8. Προσωπικό Chat



Με την επιλογή του προσωπικού chat ανοίγει καινούργιο παράθυρο για συνομιλία με τον συγκεκριμένο χρήστη.

## 9. Καρτέλα συνομιλιών



Από το προφίλ του ο χρήστης έχει τη δυνατότητα να δει και να επιλέξει τις συνομιλίες που έχει ξεκινήσει με κάποιον άλλο χρήστη καθώς και αν έχει νέα μηνύματα σε κάποια συνομιλία.

## 6.6 Δοκιμές

Με την περάτωση της εφαρμογής πραγματοποιήθηκαν κάποιες δοκιμές σε πραγματικές συσκευές (Xiaomi 3S Pro και Samsung I9190 Galaxy S4 mini) και σε εικονική συσκευή Android (LG Nexus 5), για τον εντοπισμό πιθανών σφαλμάτων. Δυστυχώς η δοκιμή σε λειτουργικό iOS δεν ήταν εφικτή καθώς προϋποθέτει την κατοχή κινητής συσκευής iPhone και υπολογιστή Macbook των οποίων το κόστος ήταν πολύ υψηλό. Οι δοκιμές που έλαβαν χώρα είχαν τα επιθυμητά αποτελέσματα. Στον παρακάτω πίνακα δίνονται τα χαρακτηριστικά ορισμένων συσκευών στις οποίες έγιναν δοκιμές.

Συσκευή	Xiaomi 3S Pro	LG Nexus 5	Samsung I9190 Galaxy S4 mini
Έκδοση λογισμικού (Android OS version)	Lollipop v 5.1	Marshmallow v 6	Jelly Bean v 4.3
Μέγεθος/Ανάλυση οθόνης	5.0'' /720x1280 pixels	4.95'' /1080x1920 pixels	4.3'' /540x960 pixels
Επεξεργαστής CPU	Qualcomm MSM8939 Snapdragon 616, Octa-core (4x1.5 GHz Cortex-A53 & 4x1.2 GHz Cortex-A53)	Qualcomm MSM8974 Snapdragon 800, Quad-core 2.3 GHz Krait 400	Qualcomm MSM8930AB Snapdragon 400, Dual-core 1.7 GHz Krait 300
Μνήμη RAM	3072MB	2048MB	1536MB

Πίνακας 6.5: Χαρακτηριστικά συσκευών δοκιμών

## 7 Επίλογος

### 7.1 Συμπεράσματα

Με το τέλος της διπλωματικής εργασίας, ο αρχικός στόχος που είχε τεθεί, για τη δημιουργία μίας εφαρμογής που θα φέρνει σε επαφή ανθρώπους με κοινό ταξιδιωτικό προορισμό, έχει επιτευχθεί.

Η εφαρμογή είναι ιδιαίτερος φιλική προς το χρήστη με ένα πολύ απλό και λειτουργικό γραφικό περιβάλλον, που την καθιστά γρήγορη στην απόκριση και την απόδοση.

Ως πρώτη προσπάθεια για υλοποίηση εφαρμογής, μελετήθηκαν σε βάθος οι βασικές αρχές του Web Developing και του Ionic Framework και επιτεύχθηκε εξοκείωση με τα λογισμικά πακέτα υποστήριξης της ανάπτυξης υβριδικών εφαρμογών για έξυπνα κινητά τηλέφωνα.

### 7.2 Μελλοντικές επεκτάσεις και αναβαθμίσεις

Με την πάροδο των χρόνων οι ανάγκες των χρηστών διαρκώς αλλάζουν και επεκτείνονται, επομένως επιτακτική καθίσταται και η ανάγκη για συνεχή βελτίωση των εφαρμογών. Στην κατεύθυνση αυτή, θα μπορούσαν να υπάρξουν κάποιες περαιτέρω βελτιώσεις της εφαρμογής.

- Βελτίωση λειτουργίας chat: Εισαγωγή μηνυμάτων-ειδοποιήσεων (push notifications) για την ειδοποίηση, με ήχο/δόνηση και εικονιδίου, του χρήστη σε περίπτωση νέου μηνύματος σε κάποια από τις συνομιλίες του.
- Βελτίωση λειτουργίας Δημοφιλέστερων Προορισμών: Δημιουργία επιλογής φίλτρων για την αναζήτηση των Δημοφιλέστερων Προορισμών (περισσότερες ερωτήσεις, περισσότερες απαντήσεις, περισσότερα likes απαντήσεων κα.)
- Αντικατάσταση του Cordova με το Capacitor: Αρκετές φορές η υποστήριξης του Cordova στα τρέχοντα plugins του είναι αναξιόπιστη. Συνήθως πρέπει να βασίζομαστε σε τρίτες εταιρίες, ώστε να διατηρούν πάντοτε τα plugins τους ενημερωμένα. Εδώ έρχεται να βοηθήσει η ανάπτυξη του Capacitor το οποίο είναι ένα runtime για Cross-Platform εφαρμογές που διευκολύνει την κατασκευή web εφαρμογών που εκτελούνται εγγενώς σε iOS, Android, Electron και στον ιστό. Καλούμε αυτές τις εφαρμογές "Native Progressive Web Apps" και αντιπροσωπεύουν το επόμενο στάδιο των Υβριδικών εφαρμογών. Το Capacitor παρέχει ένα σύνολο API που επι-

τρέπει σε μια εφαρμογή να παραμείνει όσο το δυνατόν πιο κοντά στα πρότυπα του διαδικτύου, ενώ παράλληλα έχει πρόσβαση σε πλούσιες λειτουργίες εγγενούς συσκευών. Η προσθήκη εγγενών λειτουργιών είναι εύκολη με ένα απλό API Plugin για Swift σε iOS, Java για Android και JavaScript για τον ιστό. Τη δεδομένη χρονική στιγμή το Capacitor βρίσκεται σε Beta έκδοση.

## 8 Βιβλιογραφία

1. <https://ionicframework.com/docs>
2. [https://en.wikipedia.org/wiki/Mobile\\_phone](https://en.wikipedia.org/wiki/Mobile_phone)
3. [https://en.wikipedia.org/wiki/Android\\_version\\_history](https://en.wikipedia.org/wiki/Android_version_history)
4. <https://el.wikipedia.org/wiki/Android>
5. <https://en.wikipedia.org/wiki/IOS>
6. [https://en.wikipedia.org/wiki/Operating\\_system](https://en.wikipedia.org/wiki/Operating_system)
7. <https://docs.angularjs.org/tutorial>
8. <https://www.w3schools.com/>
9. <https://www.gartner.com/en>
10. <https://www.statista.com/>
11. <https://www.idc.com/promo/smartphone-market-share/os>
12. <https://www.nielsen.com/us/en/insights/news/2012/apps-dominate-mobile-time-spent-accessing-travel-in-u-s.html>
13. <https://www.forbes.com>
14. <https://www.telerik.com/blogs/everything-you-wanted-to-know-about-native-hybrid-and-web-apps-but-were-afraid-to-ask>
15. <https://www.app-press.com/blog/whats-the-cost-to-maintain-an-app>
16. <https://ionicframework.com/survey/2017>
17. <https://capacitor.ionicframework.com/docs/>
18. <https://devdactic.com/>
19. <https://school.hirez.io/>
20. <https://javebratt.com/>
21. <https://www.joshmorony.com>
22. <https://www.djamware.com/>
23. <https://stackoverflow.com>
24. <https://angular.io/docs>
25. <https://firebase.google.com/docs/>