



ΤΕΙ ΗΠΕΙΡΟΥ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
Τ.Ε.

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**"ΑΝΑΠΤΥΞΗ ΕΙΚΟΝΟΣΤΡΑΦΩΝ ΚΙΝΗΤΩΝ ΕΦΑΡΜΟΓΩΝ ΓΙΑ
ΔΙΑΔΙΚΤΥΑΚΕΣ ΠΛΑΤΦΟΡΜΕΣ-WEB ΑΝΕΞΑΡΤΗΤΩΣ ΛΕΙΤΟΥΡΓΙΚΟΥ
ΣΥΣΤΗΜΑΤΟΣ"**

Γαβριήλ Βασιλάκης

Επιβλέπων Καθηγητής: Κωνσταντίνος Αγγέλης

ΆΡΤΑ 2015

ΕΥΧΑΡΙΣΤΙΕΣ

Για την εκπόνηση της παρούσας Πτυχιακής Εργασίας, θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή κ. Κωνσταντίνο Αγγέλη και τον κο Ιάκωβο Σ. Βενιέρη καθηγητή ΕΜΠ.

Επίσης θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον κο Σταύρο Ι. Παπαπαναγιώτου, Διδάκτορα Ηλεκτρολόγο Μηχανικό και Μηχανικό Υπολογιστών ΕΜΠ, Γενικό Διευθυντή της εταιρίας «ΔΙΑΔΙΚΤΥΑΚΕΣ ΕΠΙΧΕΙΡΗΣΕΙΣ ΕΛΛΑΔΟΣ» καθώς και τον κο Λάζαρο Τουμανίδη, Πτυχιούχο Ηλεκτρονικό Μηχανικό ΤΕ.

Χωρίς την πολύτιμη συνεισφορά τους, δεν θα είχα φτάσει στο επιθυμητό αποτέλεσμα, την εκπόνηση της πτυχιακής.

Άρτα 2015,
Γαβριήλ Βασιλάκης

Περιεχόμενα

1. ΕΙΣΑΓΩΓΗ.....	7
1.1 Αντικείμενο της πτυχιακής εργασίας	7
2. Η ΒΙΟΜΗΧΑΝΙΑ ΤΗΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ ΓΙΑ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ.....	8
2.1 Υπάρχουσα Κατάσταση	8
2.2 Μερίδιο Αγοράς	9
2.3 Εγγενείς εφαρμογές (Native applications)	11
2.4 Διαδικτυακές Εφαρμογές (Web applications).....	13
2.5 Υβριδικές Εφαρμογές (Hybrid applications).....	15
2.6 Πολυεπίπεδες πλατφόρμες Ανάπτυξης (Cross-Platforms development)	18
2.6.1 Πολυεπίπεδη πλατφόρμα ανάπτυξης (cross-platform).....	18
2.6.2 Χρησιμοποίηση πολυεπίπεδης πλατφόρμας ανάπτυξης (cross-platform).....	18
2.6.3 Πλεονεκτήματα χρησιμοποίησης πολυεπίπεδης πλατφόρμας ανάπτυξης (cross-platform).....	18
2.6.4 Μειονεκτήματα χρησιμοποίησης πολυεπίπεδης πλατφόρμας ανάπτυξης (cross-platform).....	19
2.6.5 Προγραμματιστικά εργαλεία και τεχνολογίες για πολυεπίπεδες πλατφόρμες ανάπτυξης (cross-platform)	19
3. Ανάπτυξη Εφαρμογής για την ιστοπύλη Capital.gr	28
3.1 Περιγραφή Έργου.....	28
3.2 Χαρακτηριστικά υλοποίησης	29
3.3 Τεχνολογίες-Πλαίσια εφαρμογής που χρησιμοποιήθηκαν για την υλοποίηση του έργου Capital.gr	33
3.3.1 Πολυεπίπεδη πλατφόρμα ανάπτυξης εφαρμογής PhoneGap	33
3.4 Πλαίσιο ανάπτυξης Angular.js	40
3.5 Πλαίσιο ανάπτυξης ionic HTML5.....	43
3.6 Εργαλεία που χρησιμοποιήθηκαν.....	44
3.6.1 HTML.....	44
3.6.2 Χρησιμοποίηση HTML5	44
3.6.3 CSS.....	45
3.6.4 Χρησιμοποίηση CSS3	45
3.6.5 JavaScript	46
3.6.6 Server.....	46
3.6.7 PHP.....	46
3.6.8 jQuery	47
3.6.9 jQuery Mobile	47

3.6.10 Προσθήκη του jQuery mobile σε ιστοσελίδα.....	48
4. Υπηρεσία γνωστοποιήσεων (notification service)	50
4.1 Ωθήσεις ειδοποιήσεων (Push Notifications)	50
4.1.1 Χρησιμοποίηση των ωθήσεων ειδοποιήσεων (push notifications)	50
4.2 Ενεργοποίηση των ωθήσεων ειδοποιήσεων (push notifications) σε συσκευή Android	50
4.2.1 Δημιουργία έργου API Google.....	51
4.2.2 Ενεργοποίηση υπηρεσίας GCM και η απόκτηση κλειδιού API.....	51
4.2.3 Αποστολή push notifications της εφαρμογής Capital.gr σε συσκευή Android	55
4.3 Ενεργοποίηση των push notifications σε συσκευή iOS	57
4.3.1 Παράμετροι για push notifications	59
4.3.2 Provisioning profile (Διαθέσιμο περίγραμμα) και πιστοποιητικά SSL.....	61
4.3.3 Κατασκευή App ID (Ταυτότητα App) και SSL Certificate	62
4.3.4 Δημιουργία Provisioning profile	67
4.3.5 Αποστολή push notifications της εφαρμογής Capital.gr σε συσκευή iOS	71
5. Συμπεράσματα – Μελλοντικές επεκτάσεις	73
Αναφορές.....	75

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1: Στατιστικά στοιχεία για τους χρήστες που χρησιμοποιούν κινητές συσκευές για περιήγηση στο διαδίκτυο σε σχέση με εκείνους που χρησιμοποιούν τον προσωπικό τους υπολογιστή.	8
Σχήμα 2: Κατάλογος όλων των κατηγοριών και ο αριθμός των εφαρμογών (apps) για την Apple, το Android και το Windows Phone.	9
Σχήμα 3: Οι παγκόσμιες πωλήσεις κινητών συσκευών σε τελικούς χρήστες ανά λειτουργικό σύστημα το 2011, το 2012 και το 1ο τρίμηνο του 2013.....	9
Σχήμα 4: Οι παγκόσμιες πωλήσεις κινητών συσκευών σε τελικούς χρήστες από προμηθευτές το 2011, το 2012 και το 1ο τρίμηνο του 2013.....	10
Σχήμα 5: Εκτιμήσεις του Canaccord Genuity για τις πωλήσεις των Smartphones από την OEM.	10
Σχήμα 6: Σύγκριση χαρακτηριστικών Native, Υβριδικών και Web εφαρμογών.	17
Σχήμα 7: Στατιστικά στοιχεία μεταξύ χρηστών desktop, mobile και multi-platform.	27

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Η αρχιτεκτονική native εφαρμογής για κινητή συσκευή.	11
Εικόνα 2: Η αρχιτεκτονική διαδικτυακής εφαρμογής για κινητή συσκευή.	13
Εικόνα 3: Η αρχιτεκτονική υβριδικής εφαρμογής για κινητή συσκευή.	15
Εικόνα 4: Οθόνη έναρξης του Titanium.....	20
Εικόνα 5: Τρόπος λειτουργίας πολυεπίπεδης πλατφόρμας (cross-platform) Titanium.....	21
Εικόνα 6: Λογότυπο του Corona SDK.....	21
Εικόνα 7: Λογότυπο του Intel XDK.....	22
Εικόνα 8: Intel XDK Build Tab.	22
Εικόνα 9: Λογότυπο του Sencha Touch.....	23
Εικόνα 10: Λογότυπο του Xamarin.....	24
Εικόνα 11: Δημιουργία εφαρμογών στην πολυεπίπεδη πλατφόρμα ανάπτυξης Xamarin για iOS, Android με χρήση της γλώσσας προγραμματισμού C#.	24
Εικόνα 12: Λογότυπο του Unity 3D.....	25
Εικόνα 13: Αρχιτεκτονικό διάγραμμα για πολυεπίπεδη πλατφόρμα (PhoneGap).....	33
Εικόνα 14: Λογότυπο του node.js.....	36
Εικόνα 15: Αρχική σελίδα Android simulator.	39
Εικόνα 16: Αρχική σελίδα Android simulator με την χρήση πλατφόρμας PhoneGap.....	39
Εικόνα 17: Λογότυπο του AngularJS.....	40
Εικόνα 18: Μοτίβο αρχιτεκτονικής MVC.....	41
Εικόνα 19: Κώδικας HTML με ενσωμάτωση Angular.js.....	42
Εικόνα 20: Αποτέλεσμα κώδικα εφαρμογής με χρήση του Angular.js.....	42
Εικόνα 21: HTML DOM δέντρο αντικειμένων.....	42
Εικόνα 22: Λογότυπο του ionic.....	43
Εικόνα 23: Συνύπαρξη του ionic και του AngularJS.	43
Εικόνα 24: Προσθήκη του jQuery mobile σε ένα CDN.....	48
Εικόνα 25: Προσθήκη του jQuery mobile στον υπολογιστή.....	48
Εικόνα 26: Λογότυπα βασικών γλωσσών προγραμματισμού.	49
Εικόνα 27: Αρχική σελίδα του προγράμματος Putty.	55
Εικόνα 28: Εκτέλεση εντολών στην κονσόλα, για αποστολή push notification σε συσκευή Android.....	56
Εικόνα 29: Αποτελέσματα αποστολής του notification σε συσκευή Android.	57
Εικόνα 30: Διαδικασία ενεργοποίησης της λειτουργίας των push notifications.	58
Εικόνα 31: Αρχική σελίδα του Apple Developer.....	62
Εικόνα 32: Λίστα με τις υπηρεσίες της εφαρμογής.....	63
Εικόνα 33: Ανάπτυξη και παραγωγή του πιστοποιητικού.	64
Εικόνα 34: Διαδικασία δημιουργίας του CSR.....	65
Εικόνα 35: Δημιουργία του provisioning profile.	67
Εικόνα 36: Διαδικασία δημιουργίας του provisioning profile.	68
Εικόνα 37: Επιλογή του App ID.....	68
Εικόνα 38: Επιλογή πιστοποιητικών.....	69
Εικόνα 39: Επιλογή των συσκευών για να δημιουργηθεί το provisioning profile.....	69
Εικόνα 40: Διαδικασία ονομασίας προφίλ.	70
Εικόνα 41: Εκτέλεση εντολών στην κονσόλα, για αποστολή push notification σε συσκευή iOS.....	71
Εικόνα 42: Αποτελέσματα αποστολής του notification στη συσκευή iOS.	72

1. ΕΙΣΑΓΩΓΗ

1.1 Αντικείμενο της πτυχιακής εργασίας

Η παρούσα πτυχιακή εργασία αναφέρεται στην ανάπτυξη εικονοστραφών κινητών εφαρμογών για διαδικτυακές πλατφόρμες με χρήση εργαλείων προγραμματισμού – ανάπτυξης ανεξαρτήτου λειτουργικού συστήματος.

Είναι κατανοητό, ότι η ραγδαία ανάπτυξη των κινητών συσκευών, καθώς επίσης και η τάχιστα εξάπλωση των νέων διαδικτυακών ψηφιακών εφαρμογών και υπηρεσιών, όπως η διαδικτυακή τηλεόραση δημιούργησε την ανάγκη για την ανάπτυξη εφαρμογών, οι οποίες θα μπορεί να είναι προσβάσιμες από όλους τους τύπους κινητών συσκευών (βάση λειτουργικού συστήματος).

Αυτό ακριβώς αναπτύσσεται στην συγκεκριμένη πτυχιακή, όπου εξετάζεται η ανάπτυξη υβριδικών εφαρμογών με την χρήση της πολυεπίπεδης πλατφόρμας *phonegap*. Επίσης αναφέρονται οι διάφοροι τύπου πολυεπίπεδες πλατφόρμες και τα πλαίσια ανάπτυξης εφαρμογών που λειτουργούν καθώς και συγκεκριμένες λειτουργίες που υποστηρίζονται από αυτές.

Λόγω της πρακτικής μου άσκησης στην εταιρεία ΔΙΑΔΙΚΤΥΑΚΕΣ ΕΠΙΧΕΙΡΗΣΕΙΣ ΕΛΛΑΔΟΣ ΕΠΕ, μου δώθηκε η ευκαιρία να συμμετάσχω στην ομάδα ανάπτυξης των εφαρμογών για κινητές συσκευές για τον πελάτη ΑΤΤΙΚΕΣ ΕΚΔΟΣΕΙΣ – *capital.gr*. Η εμπλοκή μου αυτή με οδήγησε να μπορέσω να αποκτήσω μία μικρή εμπειρία σχετικά με την ανάπτυξη εφαρμογών σε πλατφόρμα *PhoneGap* χρησιμοποιώντας πλαίσια ανάπτυξης όπως *Angular.js* *MVVM* και *Ionic HTML5*.

Οι συγκεκριμένες τεχνολογίες είναι *state- of- the art* και αποτελούν την αιχμή του δόρατος εταιρειών ανάπτυξης εφαρμογών σε παγκόσμιο επίπεδο.

Η πτυχιακή εργασία εξετάζει και την ανάπτυξη λειτουργιών *Ωθήσεων Ειδοποιήσεων (push notification)*, οι οποίες αναπτύχθηκαν στο πλαίσιο της υλοποίησης του έργου *capital.gr* από την εταιρεία ΔΙΑΔΙΚΤΥΑΚΕΣ ΕΠΙΧΕΙΡΗΣΕΙΣ ΕΛΛΑΔΟΣ ΕΠΕ.

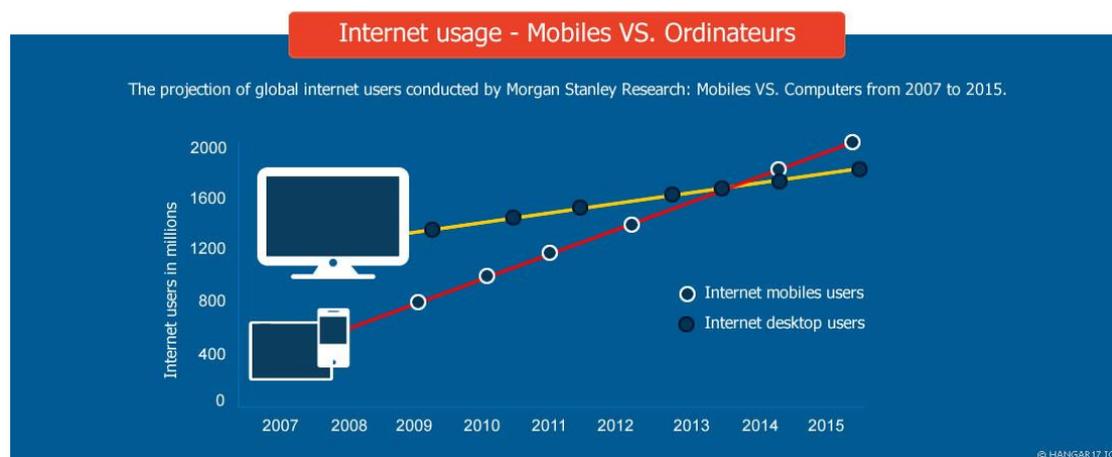
2. Η ΒΙΟΜΗΧΑΝΙΑ ΤΗΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ ΓΙΑ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ

2.1 Υπάρχουσα Κατάσταση

Τα «έξυπνα» κινητά τηλέφωνα ή smartphones αποτελούν πλέον αναπόσπαστο κομμάτι της κοινωνίας και της καθημερινότητας μας. Εκατομμύρια άνθρωποι σε όλο τον κόσμο απολαμβάνουν τις αυξημένες δυνατότητες που προσφέρουν, δυνατότητες που πριν από μερικά χρόνια είχαν μόνο οι σταθεροί ηλεκτρονικοί υπολογιστές.

Ο όρος «smartphone» εμφανίστηκε για πρώτη φορά μετά το 1997 όταν η Ericsson περιέγραψε το GS 88 ως «smartphone», παρότι οι συσκευές που συνδυάζουν τηλεφωνία και πληροφορική είχαν επινοηθεί από το 1973, όπου ο Θεόδωρος Παρασκευάκος κατοχύρωσε τις έννοιες του συνδυασμού νοημοσύνης, της επεξεργασίας δεδομένων και των οθονών οπτικής απεικόνισης με τα τηλέφωνα.

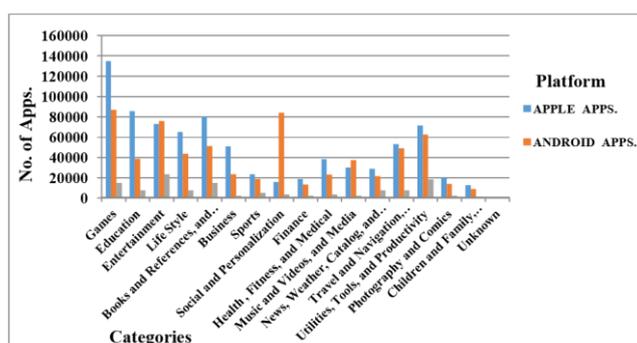
Σταδιακά μετά την εμφάνιση των ηλεκτρονικών υπολογιστών (desktop) άρχισαν να εμφανίζονται διάφορες μορφές φορητών ηλεκτρονικών υπολογιστών (notebooks-laptops), οι οποίοι έδιναν τη δυνατότητα στο χρήστη να μπορεί να έχει πρόσβαση από διάφορα σημεία, μεταφέροντας το μαζί του. Τα τελευταία χρόνια στη θέση των netbooks, εμφανίστηκε μια καινούρια κατηγορία κινητών συσκευών τύπου «ταμπλέτας» τα λεγόμενα tablets, τα οποία αποτελούν μια ενδιάμεση κατάσταση των έξυπνων κινητών τηλεφώνων και των ηλεκτρονικών υπολογιστών, καθώς τα περισσότερα εξ' αυτών χρησιμοποιούν όμοια λειτουργικά συστήματα με τα έξυπνα κινητά τηλέφωνα (Android, IOS, Windows Phone, κλπ.) ενώ παράλληλα λόγω της μεγαλύτερης οθόνης τους είναι χρηστικά και σαν notebooks. Στατιστικά δείχνουν ότι όσο περνάνε τα χρόνια οι κινητές συσκευές έχουν μια αυξανόμενη τάση να αντικαταστήσουν τους ηλεκτρονικούς υπολογιστές.



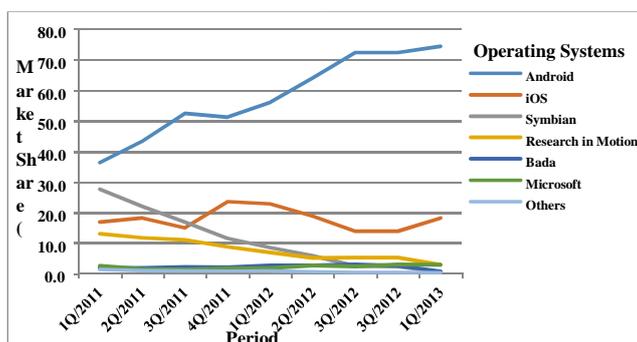
Σχήμα 1: Στατιστικά στοιχεία για τους χρήστες που χρησιμοποιούν κινητές συσκευές για περιήγηση στο διαδίκτυο σε σχέση με εκείνους που χρησιμοποιούν τον προσωπικό τους υπολογιστή.

2.2 Μερίδιο Αγοράς

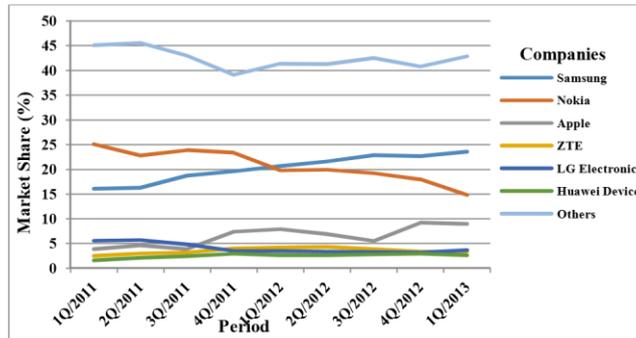
Στις 27 Φεβρουαρίου του 2013, ο συνολικός αριθμός των αιτήσεων στο Android market ήταν περίπου 652.000 εφαρμογές. Στις 28 Φεβρουαρίου του 2013, ο συνολικός αριθμός των εφαρμογών στο App Store της Apple ήταν 803.000 εφαρμογές. Επιπλέον η Microsoft δημοσίευσε 127.000 εφαρμογές για Windows Phone την ίδια χρονική περίοδο. Τέλος, τα Blackberry είχαν 10 αιτήσεις έφθασαν περίπου τις 70.000 αιτήσεις. Στο σχήμα 2 παρακάτω και στον πίνακα 3 βλέπουμε μια λίστα στην οποία κατηγοριοποιούνται οι αιτήσεις για τις τρεις πιο δημοφιλείς πλατφόρμες: το App Store της Apple, το Android Store και το Windows Phone Store. Στη συνέχεια, το σχήμα 3 απεικονίζει τις παγκόσμιες πωλήσεις Smartphones στους τελικούς χρήστες ανά λειτουργικό σύστημα το 2011, το 2012 και το πρώτο τρίμηνο του 2013. Στο σχήμα 4 απεικονίζονται οι παγκόσμιες πωλήσεις κινητών συσκευών σε τελικούς χρήστες από τον προμηθευτή το 2011, το 2012 και το πρώτο τρίμηνο του 2013. Τέλος, το σχήμα 5 απεικονίζει τις εκτιμήσεις του Canaccord Genuity για τις μονάδες πωλήσεων των Smartphones από την OEM.



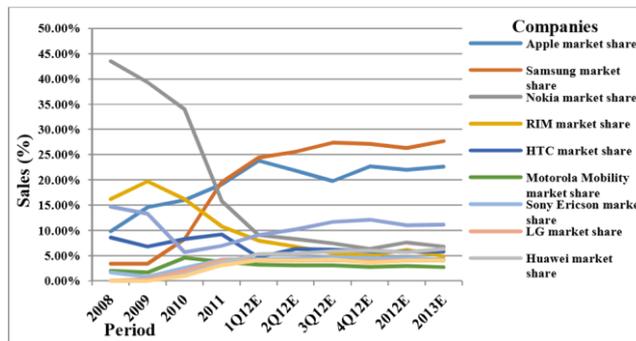
Σχήμα 2: Κατάλογος όλων των κατηγοριών και ο αριθμός των εφαρμογών (apps) για την Apple, το Android και το Windows Phone.



Σχήμα 3: Οι παγκόσμιες πωλήσεις κινητών συσκευών σε τελικούς χρήστες ανά λειτουργικό σύστημα το 2011, το 2012 και το 1ο τρίμηνο του 2013.



Σχήμα 4: Οι παγκόσμιες πωλήσεις κινητών συσκευών σε τελικούς χρήστες από προμηθευτές το 2011, το 2012 και το 1ο τρίμηνο του 2013.



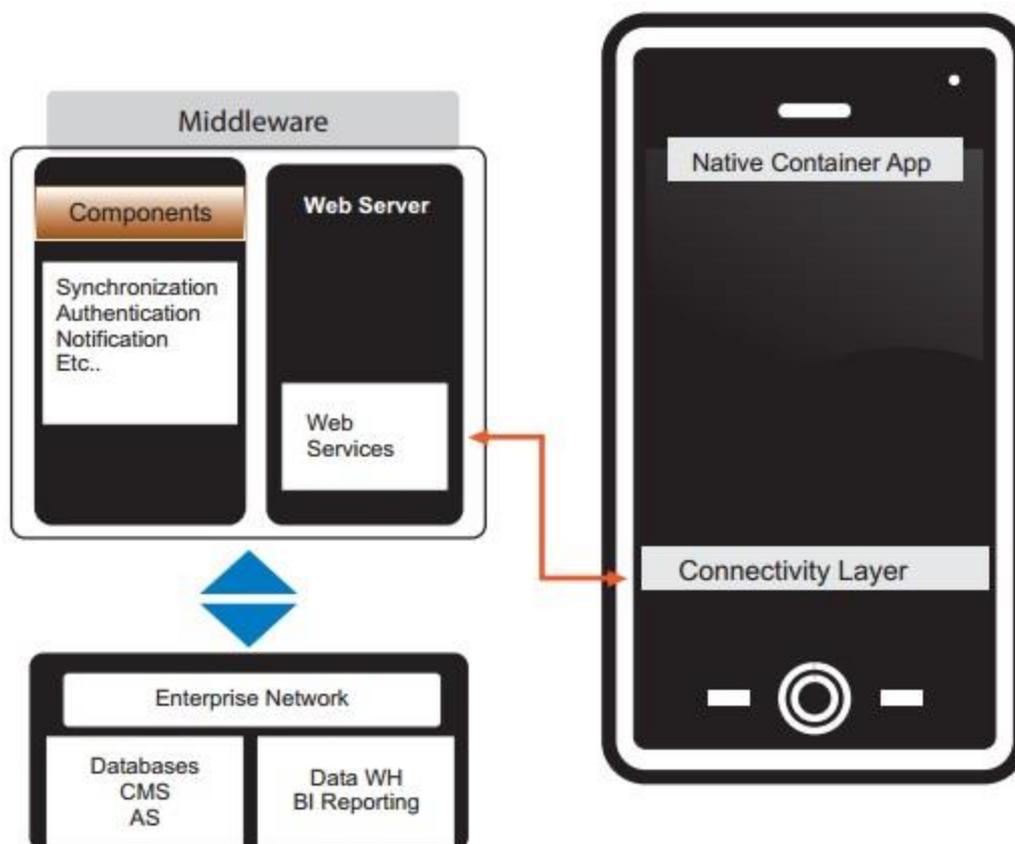
Σχήμα 5: Εκτιμήσεις του Canaccord Genuity για τις πωλήσεις των Smartphones από την OEM.

Με βάση τα ανωτέρω είναι κατανοητό ότι οι κινητές συσκευές αποτελούν το μέσο προβολής και χρήσης στοιχείων. Μέσα σε αυτό το πλαίσιο αναπτύχθηκε με γοργούς ρυθμούς η βιομηχανία ανάπτυξης εφαρμογών σε κινητές συσκευές. Εταιρείες όπως η Google, Microsoft κ.α. επενδύουν δισεκατομμύρια δολάρια για την υλοποίηση πλαισίων ανάπτυξης (Frameworks) καθώς και πολυεπίπεδων πλατφορμών ανάπτυξης υβριδικών εφαρμογών[22].

Στις επόμενες παραγράφους αναλύονται οι τύποι εφαρμογών κινητών συσκευών που κυκλοφορούν στην αγορά.

2.3 Εγγενείς εφαρμογές (Native applications)

Οι εγγενείς εφαρμογές είναι εφαρμογές οι οποίες αποτελούν την πλειοψηφία των εφαρμογών κινητών συσκευών που βρίσκονται στα ηλεκτρονικά καταστήματα της Google, Apple, Microsoft.



Εικόνα 1: Η αρχιτεκτονική native εφαρμογής για κινητή συσκευή.

Η διαδικασία της εγκατάστασης των native εφαρμογών μπορεί να γίνει είτε από το χρήστη είτε από την ίδια την εφαρμογή που περιέχεται ως μέρος του λειτουργικού συστήματος. Ο πιο διαδεδομένος τρόπος εγκατάστασης μιας native εφαρμογής είναι μέσω του εκάστοτε διαδικτυακού καταστήματος εφαρμογών της αντίστοιχης εταιρείας που δημιούργησε το λειτουργικό σύστημα π.χ. Windows store για το λειτουργικό σύστημα των Windows phone. Ένα πρώτο σημαντικό πλεονέκτημα που έχουν οι native εφαρμογές είναι πως δεν χρειάζεται να είμαστε συνδεδεμένοι στο ίντερνετ για να χρησιμοποιήσουμε την εφαρμογή, εκτός βέβαια και αν είναι απαίτηση της ίδιας της εφαρμογής. Ακόμα, η διεπαφή χρήστη (user interface) και η απόκριση είναι συνήθως πολύ πιο γρήγορα και εντυπωσιακά από τα αντίστοιχα μιας web εφαρμογής, ενώ παράλληλα έχουμε τη δυνατότητα να χρησιμοποιήσουμε και όλα τα προηγμένα χαρακτηριστικά της εκάστοτε εφαρμογής όπως είναι η κάμερα, το GPS κ.ά. Μετά την εγκατάσταση, ο χρήστης μπορεί να εκκινήσει την εφαρμογή όπως κάθε άλλη εφαρμογή που είναι προ-εγκατεστημένη στη συσκευή. Συνήθως δημιουργείται ένα εικονίδιο στην

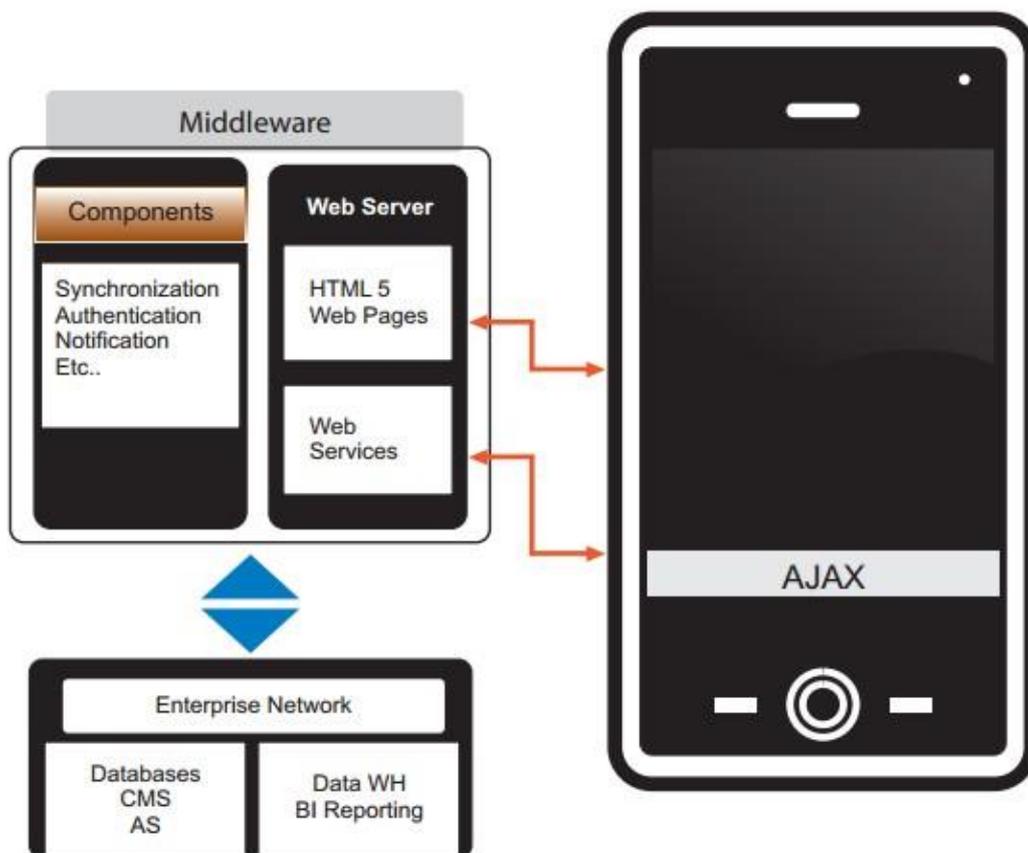
επιφάνεια εργασίας της συσκευής, το οποίο επιλέγει ο χρήστης κάθε φορά που επιθυμεί να εκτελέσει την εφαρμογή. Κατά την πρώτη εκτέλεση η εφαρμογή θα συνδεθεί άμεσα με το λειτουργικό σύστημα και με αυτόν τον τρόπο μπορεί να έχει πρόσβαση στις λειτουργίες των εφαρμογών που συνοδεύουν το λειτουργικό σύστημα, αποκτώντας πλήρη έλεγχο της συσκευής. Ας πάρουμε για παράδειγμα τη κάμερα της συσκευής, την οποία ελέγχουν πολλές εφαρμογές όπως για παράδειγμα το Instagram και το Skype.

Για την δημιουργία μιας native εφαρμογής ο προγραμματιστής θα πρέπει να γράψει τον πηγαίο κώδικα, στην γλώσσα προγραμματισμού που υποστηρίζει το λειτουργικό σύστημα της συσκευής και να συμπεριλάβει οτιδήποτε άλλο είναι απαραίτητο για την λειτουργία της εφαρμογής, όπως για παράδειγμα εικόνες, βίντεο κ.ά. Χρησιμοποιώντας εργαλεία που παρέχονται από την εταιρεία που δημιούργησε το λειτουργικό σύστημα, τα παραπάνω αρχεία μεταγλωττίζονται και παράγεται ένα αρχείο το οποίο αποτελεί την εφαρμογή που θα αποθηκευτεί στη συσκευή. Τα εργαλεία αυτά σε συνδυασμό με επιπρόσθετες ευκολίες αποτελούν το περιβάλλον ανάπτυξης εφαρμογών (SDK) που προορίζονται για το εκάστοτε λειτουργικό σύστημα. Η διαδικασία ανάπτυξης μιας native εφαρμογής παρουσιάζει ομοιότητες μεταξύ των διαφορετικών λειτουργικών συστημάτων, όμως τα περιβάλλοντα ανάπτυξης (SDK's) διαφέρουν μεταξύ τους. Οι διαφορές που παρουσιάζονται έχουν ως αποτέλεσμα ένα από τα πιο σημαντικά μειονεκτήματα της ανάπτυξης native εφαρμογών, ο κώδικας που γράφτηκε για ένα συγκεκριμένο λειτουργικό σύστημα δεν μπορεί να χρησιμοποιηθεί σε κάποιο άλλο. Επίσης ένα μικρό μειονέκτημα μπορεί να θεωρηθεί και το γεγονός πως για να γίνει update μια εφαρμογή, πρέπει να ξανανέβει στο εκάστοτε store, να την ξανακατεβάσει και να την ξανά-εγκαταστήσει ο εκάστοτε χρήστης ενώ στην περίπτωση των web app το update είναι πολύ πιο άμεσο. Η εφαρμογή αφού εγκατασταθεί στη κινητή συσκευή και εκτελεστεί από το χρήστη, αλληλοεπιδρά με το λειτουργικό σύστημα της συσκευής κάνοντας χρήση των διαθέσιμων διεπαφών. Οι διεπαφές προγραμματισμού μπορούν να χωριστούν σε δύο βασικές κατηγορίες:

1. Διεπαφές προγραμματισμού χαμηλού επιπέδου (low-level APIs).
2. Διεπαφές προγραμματισμού υψηλού επιπέδου (high-level APIs).

Οι διεπαφές προγραμματισμού χαμηλού επιπέδου παρέχουν τη δυνατότητα σε μια native εφαρμογή να αλληλοεπιδρά με το πληκτρολόγιο, την οθόνη αφής, να ελέγχει την κάμερα για λήψη φωτογραφιών ή βίντεο και να συνδέεται σε δίκτυα. Αυτές είναι μερικές από τις δυνατότητες που παρέχουν τα παραπάνω εργαλεία, αφού στο σύνολο τους παρέχουν πρόσβαση σε οποιοδήποτε χαρακτηριστικό της συσκευής. Οι διεπαφές προγραμματισμού υψηλού επιπέδου προσφέρουν μια σειρά από υπηρεσίες όπως είναι η πρόσβαση στο ημερολόγιο, τις επαφές, η δυνατότητα τηλεφωνικών κλήσεων, αποστολής και λήψης γραπτών μηνυμάτων μέσω της συσκευής και η πλοήγηση στο διαδίκτυο.

2.4 Διαδικτυακές Εφαρμογές (Web applications)



Εικόνα 2: Η αρχιτεκτονική διαδικτυακής εφαρμογής για κινητή συσκευή.

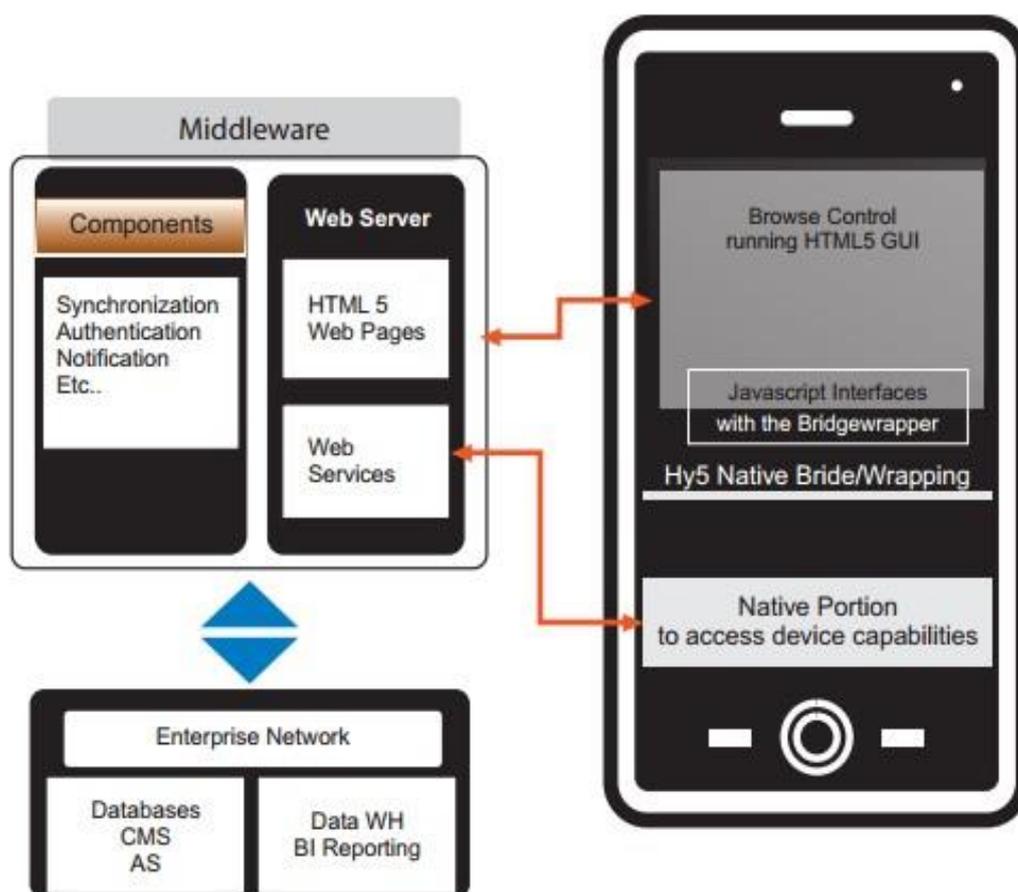
Στο πεδίο της ανάπτυξης των native εφαρμογών για κινητές συσκευές έγινε αισθητό το μέγεθος του κατακερματισμού και έτσι έρχεται να δώσει λύση σε αυτό το πρόβλημα η συγκεκριμένη κατηγορία. Επιλέγοντας τις διαδικτυακές εφαρμογές, εννοούμε πως θα χρησιμοποιούμε τον περιηγητή (browser) της συσκευής μας για να δούμε το site μας. Ένα από τα πιο σημαντικά παραδείγματα είναι οι εφαρμογές ηλεκτρονικού ταχυδρομείου (email), όπως το Gmail της google ή το outlook της Microsoft, που χρησιμοποιούν τον περιηγητή για την απεικόνισή τους και έχουν υλοποιηθεί με τεχνολογίες διαδικτύου. Οι διαδικτυακές εφαρμογές έχουν τη δυνατότητα να εκτελούνται σε διαφορετικά λειτουργικά συστήματα και συσκευές χρησιμοποιώντας τεχνολογίες διαδικτύου (HTML, CSS, JavaScript). Με τον τρόπο αυτό οι προγραμματιστές αποφεύγουν το εμπόδιο της εκμάθησης νέων γλωσσών προγραμματισμού για κάθε διαφορετικό λειτουργικό σύστημα. Ένα σημαντικό πλεονέκτημα των διαδικτυακών εφαρμογών είναι πως δεν χρειαζόμαστε κάποιο συγκεκριμένο SDK όπως (π.χ. το Android SDK) για να δημιουργήσουμε μια web εφαρμογή. Επίσης οι εφαρμογές αυτές θα φιλοξενοούνται στον εξυπηρέτη (client) και

θα μπορούν να εκτελεστούν από τον χρήστη με την επίσκεψή του στην αντίστοιχη ηλεκτρονική διεύθυνση (URL) μέσω του περιηγητή, αποφεύγοντας με αυτόν τον τρόπο τα διάφορα ηλεκτρονικά καταστήματα (marketplaces) των παρόχων των λειτουργικών συστημάτων και τις διαφορετικές απαιτήσεις που θα πρέπει να πληρεί μια εφαρμογή για να γίνει διαθέσιμη μέσω αυτών. Σε αυτό το σημείο αξίζει να σημειωθεί πως αν μια web εφαρμογή είναι καλοσχεδιασμένη δηλαδή έχει το κατάλληλο (Responsive Web Design), μπορεί να καλύψει την πλειοψηφία των συσκευών και των browser που υπάρχουν. Ακόμα ένα πλεονέκτημα μπορεί να θεωρηθεί και η ανεξάρτητη διανομή, μιας και στην ουσία η εφαρμογή ζει στο web και όχι σε κάποιο store, αφού ο χρήστης απλά συνδέεται με την εφαρμογή χρησιμοποιώντας τον περιηγητή ιστού. Ο περιηγητής ιστού είναι και αυτός με τη σειρά του μια native εφαρμογή, άρα έχει άμεση πρόσβαση στα APIs του λειτουργικού συστήματος, όμως μόνο μερικά από αυτά είναι διαθέσιμα στις εφαρμογές που εκτελούνται σε αυτόν. Ενώ οι native εφαρμογές έχουν πλήρη πρόσβαση στη συσκευή στην οποία εκτελούνται, οι web εφαρμογές έχουν περιορισμένη πρόσβαση σε πολλά χαρακτηριστικά της συσκευής και σε κάποιες περιπτώσεις η πρόσβαση απουσιάζει τελείως. Η απουσία πρόσβασης οφείλεται σε δύο κύριους λόγους[23]:

- Στην ασφάλεια της συσκευής, διότι όπως αναφέραμε και προηγουμένως οι native εφαρμογές είναι διαθέσιμες μέσω διάφορων marketplaces τα οποία πριν διαθέσουν μια εφαρμογή στους χρήστες, ελέγχουν εάν διαθέτει τα κατάλληλα χαρακτηριστικά που θα μπορούσαν να είναι επιβλαβή για την συσκευή ή για τα προσωπικά δεδομένα των χρηστών. Ενώ στις web εφαρμογές αυτός ο έλεγχος απουσιάζει αφού συνήθως είναι λίγο πιο αργές σε ανταπόκριση, μιας και στέλνουν σε έναν οποιοδήποτε εξυπηρετητή (server) ένα αίτημα (request) και περιμένουμε απάντηση.
- Στην απουσία υλοποίησης διάφορων προγραμματιστικών διεπαφών, οι διαδικτυακές εφαρμογές εξαρτώνται από την πορεία της HTML5 και κυρίως των JavaScript APIs που υποστηρίζει. Πολλά από αυτά δεν έχουν υλοποιηθεί ακόμα ή δεν υποστηρίζονται πλήρως από τους περιηγητές.

Τέλος, θα πρέπει να δοθεί ιδιαίτερη σημασία στην απόδοση των διαδικτυακών εφαρμογών, οι οποίες χρησιμοποιούν τον περιηγητή ιστού ως περιβάλλον εκτέλεσης και μπορούν να κάνουν μόνο ότι μπορεί να καταλάβει ο browser της εκάστοτε συσκευής. Η απόδοση δεν εξαρτάται μόνο από την επεξεργαστική ισχύ της συσκευής στην οποία εκτελείται, αλλά και από τον ίδιο τον περιηγητή, ο οποίος είναι αυτός που της ορίζει την απόδοση που θα έχει κάθε φορά.

2.5 Υβριδικές Εφαρμογές (Hybrid applications)

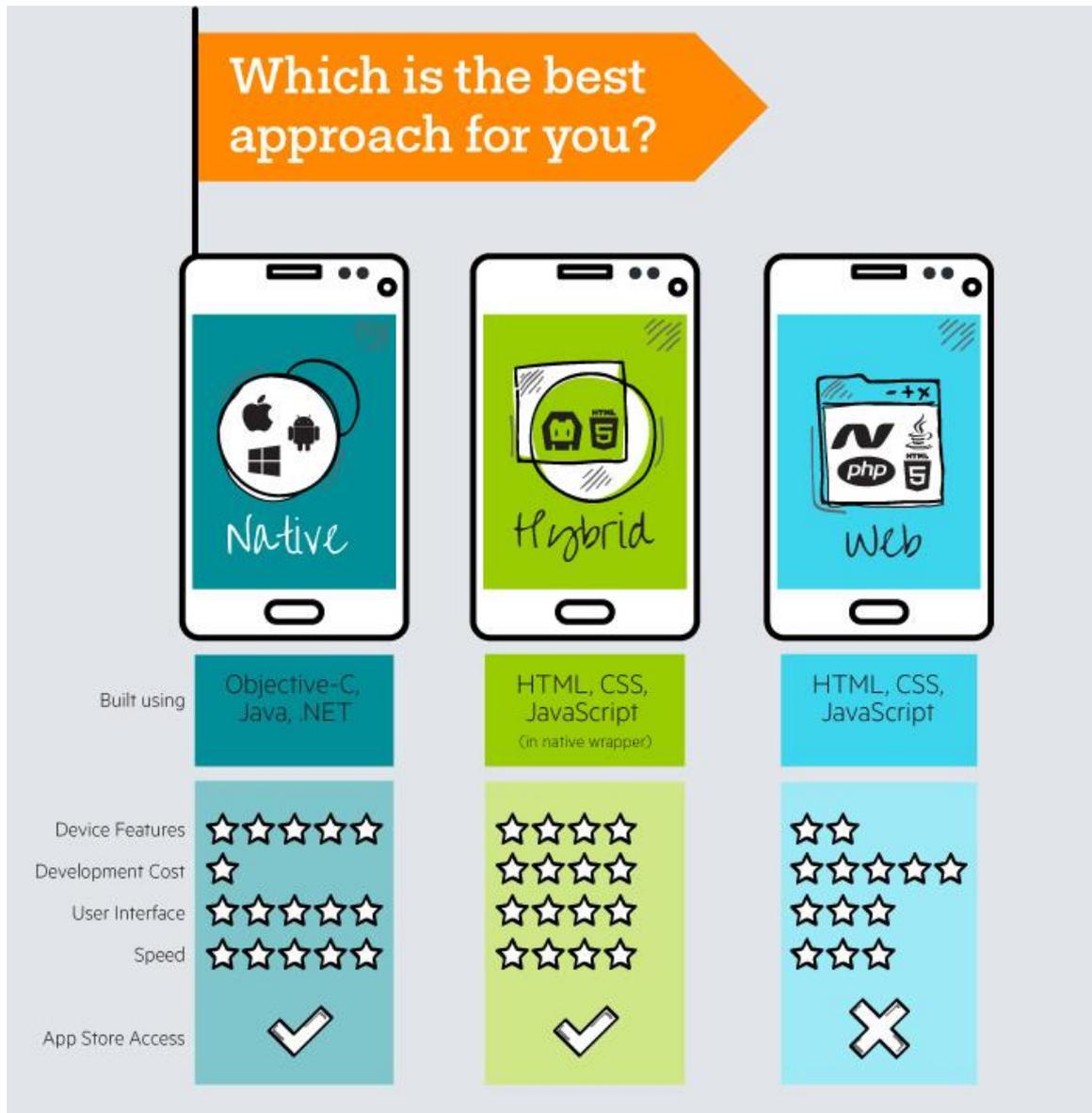


Εικόνα 3: Η αρχιτεκτονική υβριδικής εφαρμογής για κινητή συσκευή.

Οι υβριδικές εφαρμογές συνδυάζουν χαρακτηριστικά και από τις δύο παραπάνω κατηγορίες, αποτελούν μέρος των native εφαρμογών και τμήμα των web εφαρμογών. Ακολουθώντας αυτή τη προσέγγιση οι προγραμματιστές έχουν την δυνατότητα να υλοποιήσουν τις κύριες λειτουργίες της εφαρμογής χρησιμοποιώντας Web τεχνολογίες και παράλληλα να διατηρήσουν την πρόσβαση σε χαρακτηριστικά της συσκευής που είναι διαθέσιμα μόνο στις εγγενείς εφαρμογές, μέσω των APIs που λειτουργούν ως γέφυρα ανάμεσα στις κύριες λειτουργίες της εφαρμογής και τα χαρακτηριστικά της συσκευής. Οι υβριδικές εφαρμογές είναι αρκετά δημοφιλής επειδή επιτρέπουν την ανάπτυξη σε cross-platform και έτσι μειώνεται σημαντικά το κόστος ανάπτυξης. Δηλαδή επιτυγχάνεται η επαναχρησιμοποίηση του μέρους της εφαρμογής που υλοποιήθηκε σε web τεχνολογίες, ενώ το μόνο που χρειάζεται να τροποποιηθεί για να λειτουργήσει η εφαρμογή και σε άλλα λειτουργικά είναι η αλλαγή των native APIs που χρησιμοποιούνται για τις λειτουργίες της συσκευής. Οι προγραμματιστές της εφαρμογής έχουν την δυνατότητα να χρησιμοποιήσουν έτοιμα εργαλεία, όπως το Phonegap και το Sencha Touch που τους επιτρέπουν να γράφουν τον ίδιο κώδικα σε όλες τις πλατφόρμες, χρησιμοποιώντας τη δύναμη της HTML. Το μέρος της εφαρμογής

που υλοποιήθηκε με web τεχνολογίες μπορεί να είναι είτε μια σελίδα που εντοπίζεται στον εξυπηρετητή με τον οποίο επικοινωνεί η εφαρμογή, είτε ένα σύνολο από αρχεία που περιέχουν τις γλώσσες διαδικτύου, τα οποία συγκεντρώνονται στην εφαρμογή και αποθηκεύονται τοπικά στη συσκευή. Κάθε προσέγγιση έχει θετικά και αρνητικά σημεία που πρέπει να ληφθούν υπόψιν. Η πρώτη προσέγγιση επιτρέπει την παροχή ανανεώσεων της εφαρμογής αποφεύγοντας τη διαδικασία αξιολόγησης και αποδοχής από οποιοδήποτε store στο οποίο διατίθεται η εφαρμογή. Το μειονέκτημα σε αυτό εμφανίζεται στην απουσία της offline λειτουργίας της εφαρμογής, αφού το περιεχόμενο βρίσκεται στον εξυπηρετητή. Η δεύτερη προσέγγιση παρέχει τη δυνατότητα της offline λειτουργικότητας, αφού τα αρχεία που απαιτούνται συμπεριλαμβάνονται στην ίδια την εφαρμογή που εγκαθίσταται στη συσκευή. Όμως, αυτή η επιλογή αφαιρεί τη δυνατότητα άμεσων ανανεώσεων της εφαρμογής. Η λύση που μπορεί να βρεθεί είναι συνδυαστική, δηλαδή σε συνδυασμό και των δύο προσεγγίσεων. Τα αρχεία που μπορούν να φιλοξενηθούν στον εξυπηρετητή εγκαθίστανται σε αυτόν και όταν εκτελεστεί για πρώτη φορά η εφαρμογή θα τα αποθηκεύσει τοπικά στη συσκευή (app-cache). Τέλος, σε περίπτωση αλλαγής κάποιου αρχείου η συσκευή θα ενημερωθεί για την αλλαγή και θα επιχειρήσει να τα ανακτήσει εκ νέου από τον εξυπηρετητή[23].

Στο σχήμα 6 που ακολουθεί μπορούμε να δούμε την σύγκριση των χαρακτηριστικών για εγγενείς, web και υβριδικές εφαρμογές.



Σχήμα 6: Σύγκριση χαρακτηριστικών Native, Υβριδικών και Web εφαρμογών.

Πηγή: <http://www.telerik.com/blogs/everything-you-wanted-to-know-about-native-hybrid-and-web-apps-but-were-afraid-to-ask> [2].

2.6 Πολυεπίπεδες πλατφόρμες Ανάπτυξης (Cross-Platforms development)

2.6.1 Πολυεπίπεδη πλατφόρμα ανάπτυξης (cross-platform)

Η ορολογία ανεξάρτητο πλατφόρμας ή πολυ-πλατφόρμα δέχεται ως ελεύθερη μετάφραση cross-platform ή multi-platform είναι ένα χαρακτηριστικό το οποίο αναφέρεται στο λογισμικό που εφαρμόζεται και διαλειτουργεί σε πολλαπλά λειτουργικά συστήματα ή πλατφόρμες υλικού. Στη πράξη επιτρέπει στους προγραμματιστές με μια συγκεκριμένη γλώσσα προγραμματισμού να αναπτύξουν λογισμικό γράφοντας το πρόγραμμα μια μόνο φορά, δέχοντας μικρή έως καθόλου τροποποίηση, τρέχοντας σε όλα τα συστήματα.

2.6.2 Χρησιμοποίηση πολυεπίπεδης πλατφόρμας ανάπτυξης (cross-platform)

Ο κινητήριος παράγοντας πίσω από τις τεχνολογίες cross-platform είναι ότι θα είμαστε σε θέση να χρησιμοποιούμε αυτές τις τεχνολογίες για να εντοπίζουν περισσότερες πλατφόρμες, μειώνοντας την ανάπτυξη του πηγαίου κώδικα. Ένα από τα πλεονεκτήματα που έχουμε με τη χρησιμοποίηση cross-platform εφαρμογών είναι ότι μας παρέχουν πιο εύκολη υλοποίησή με τις γλώσσες HTML και JavaScript παρά με την ανάπτυξη με Java. Αυτό συμβαίνει λόγω της ευκολίας της χρήσης των εργαλείων ανάπτυξης και την εξοικείωση των γλωσσών, μειώνουν τα τεχνικά εμπόδια που μπορεί να εμποδίζουν την έγκριση της φυσικής ανάπτυξης. Παρόλα αυτά απαιτούν μεγάλη εξειδίκευση (και σε διαφορετικές τεχνολογίες) για την χρήση τους.

2.6.3 Πλεονεκτήματα χρησιμοποίησης πολυεπίπεδης πλατφόρμας ανάπτυξης (cross-platform)

- ✓ Το πλαίσιο cross-platform είναι το κατάλληλο εργαλείο για έναν προγραμματιστή εφαρμογών για κινητές συσκευές, το οποίο μπορεί να τον οδηγήσει στην επιτυχία, διότι του επιτρέπει σε μια πολύ ανταγωνιστική αγορά που θα πρέπει να είναι όσο το δυνατόν πιο αποτελεσματικοί οι μέθοδοι παραγωγής, να προγραμματίσει μια εφαρμογή η οποία θα είναι δυνατό να πραγματοποιηθεί το ίδιο σε όλες τις πλατφόρμες, εξοικονομώντας χρόνο και πόρους. Χωρίς το cross-platform ο χρόνος που απαιτούνταν ήταν αρκετός, αφού έπρεπε να προγραμματιστεί μια εφαρμογή για μια μόνο πλατφόρμα.
- ✓ Μια cross-platform στρατηγική μπορεί να εφαρμοστεί μόνο με ένα cross-platform πλαίσιο, όπου κάθε εφαρμογή χρειάζεται να προγραμματιστεί μια μόνο φορά και στη συνέχεια μπορεί να διανεμηθεί σε όλες τις πλατφόρμες, με αποτέλεσμα να πραγματοποιείται πλήρη πρόσβαση στην αγορά κινητών συσκευών.

- ✓ Η ανάπτυξη με cross-platform σε κινητές εφαρμογές επιτρέπει την εύκολη ενσωμάτωση με το περιβάλλον cloud. Η ολοκλήρωση της ανταλλαγής για παράδειγμα, κωδικοποιείται σε cross-platform περιβάλλον και θα εκτελεστεί σε πολλαπλές πλατφόρμες κινητών.[21]

2.6.4 Μειονεκτήματα χρησιμοποίησης πολυεπίπεδης πλατφόρμας ανάπτυξης (cross-platform)

- ✓ Κάθε πλατφόρμα έχει το δικό της ξεχωριστό πλαίσιο λειτουργίας και η καθεμία ορισμένη ευελιξία σε σχέση με τις υπόλοιπες. Επίσης καθίσταται υποχρεωτική η ανάγκη για τους προγραμματιστές να χρησιμοποιούν περιοριστικά εργαλεία και τεχνολογίες στην εκάστοτε εφαρμογή. Αυτό θέτει σε κίνδυνο τις προτιμήσεις των χρηστών και σε μειονεκτική θέση τους προγραμματιστές γιατί μπορεί να μην είναι σε θέση να αξιοποιήσουν τη μοναδική λειτουργικότητα και τα εργαλεία αντίστοιχης native πλατφόρμας κατά την ανάπτυξη μιας εφαρμογής για πολλαπλές πλατφόρμες.

Τα περισσότερα πλαίσια ανάπτυξης χρησιμοποιούν ένα υποσύνολο της JavaScript. Σε περίπτωση που δεσμευτούν σε ένα πλαίσιο και χρειαστεί να μετακομίσουν σε ένα διαφορετικό πλαίσιο, θα μπορούσε ενδεχομένως να αποδειχθεί δύσκολο να κάνει αυτή τη μετάβαση.

2.6.5 Προγραμματιστικά εργαλεία και τεχνολογίες για πολυεπίπεδες πλατφόρμες ανάπτυξης (cross-platform)

Υπάρχουν πολλά είδη ανάπτυξης εφαρμογών για κινητές συσκευές, τα τρία πιο βασικά εξ' αυτών είναι οι Native (εγγενείς) εφαρμογές, οι εφαρμογές Web και οι υβριδικές εφαρμογές. Οι Native εφαρμογές εκτελούνται εξ ολοκλήρου σε κινητή συσκευή και ασχολούνται άμεσα και αποτελεσματικά με το υλικό της συσκευής. Οι Web εφαρμογές με τη σειρά τους τρέχουν σε Web browsers και οι περισσότεροι από αυτούς εκτελούνται σε έναν απομακρυσμένο διακομιστή, βέβαια τρέχουν σε πολλές πλατφόρμες, αλλά με μικρότερη απόδοση. Οι υβριδικές εφαρμογές είναι ένα μείγμα της εγγενούς και της Web εφαρμογής, δεδομένου ότι οι προγραμματιστές είναι δύσκολο να γράψουν το κώδικα για κάθε πλατφόρμα και είναι αρκετά χρονοβόρο, στρέφονται στα πλαίσια cross-platform που διευκολύνουν το θέμα αυτό με την επίτευξη της λύσης «γράψω-μια-φορά αναπτύσσω-παντού» («write-once deploy-everywhere»)[21].

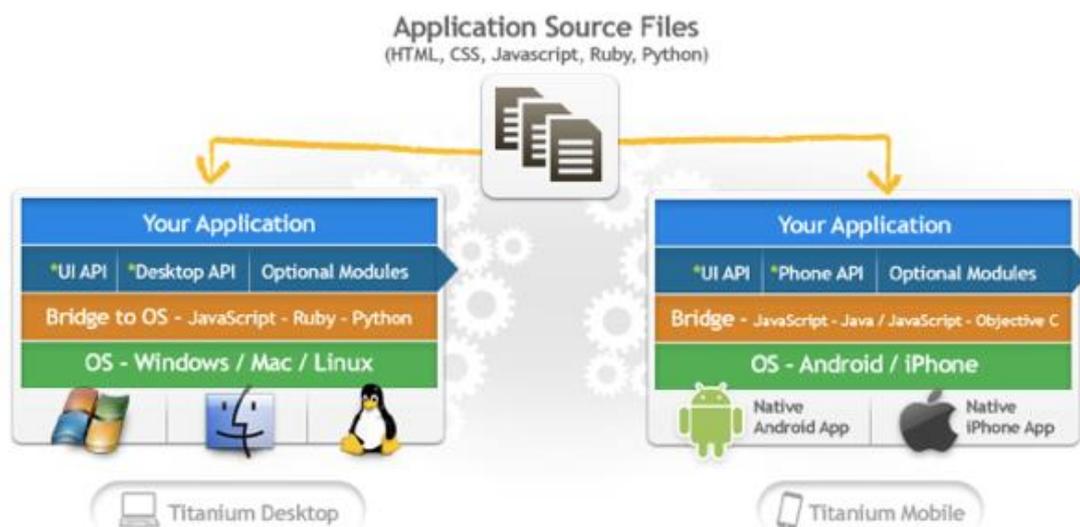
- **Appcelerator Titanium**

Η πλατφόρμα ανάπτυξης Appcelerator είναι μια ιδιωτική εταιρεία ανάπτυξης εφαρμογών για κινητές συσκευές με έδρα το Mountain View, της Καλιφόρνια η οποία εισάγαγε την πλατφόρμα Titanium το Δεκέμβριο του 2008.



Εικόνα 4: Οθόνη έναρξης του Titanium.

Το Appcelerator ιδρύθηκε από τους Jeff Haynie και Nolan Wright το 2006 με την ονομασία Hakano, με κύρια δραστηριότητα τις web εφαρμογές. Μετέπειτα τον Οκτώβριο του 2007 μετονομάστηκε σε Appcelerator με ανθρώπινο δυναμικό 160 εργαζομένους. Το Titanium είναι ένα open-source πλαίσιο που επιτρέπει τη δημιουργία εγγενών εφαρμογών (native apps) σε σταθερούς υπολογιστές, tablet, κινητά τηλέφωνα μέσω γλωσσών προγραμματισμού για το διαδίκτυο όπως HTML, PHP, JavaScript, Python.



Εικόνα 5: Τρόπος λειτουργίας πολυεπίπεδης πλατφόρμας (cross-platform) Titanium.

Το Titanium αποτελείται από ένα SDK δηλαδή μια εργαλειοθήκη ανάπτυξης λογισμικού που παρέχει τα κατάλληλα εργαλεία, μεταγλωττιστές και API για τη δημιουργία εφαρμογών. Το Titanium είναι διαθέσιμο για Mac, Linux και Windows. Για την ανάπτυξη των εφαρμογών για iPhone θα πρέπει να εκτελείτε σε λειτουργικό Mac με το iPhone SDK. Η ανάπτυξη για το Android απαιτεί το Android SDK. Τέλος, μας δίνεται και η δυνατότητα αναβάθμισης στην Titanium έκδοση Enterprise η οποία μας προσφέρει καλύτερη υποστήριξη και επιπλέον υπηρεσίες[3].

- **Corona SDK**

Η πλατφόρμα ανάπτυξης Corona SDK δημιουργήθηκε από τον Walter Luh, ο ιδρυτής της Corona Labs, ο οποίος μετά την αναχώρηση του από την Adobe το 2007, ήταν ο κύριος αρχιτέκτονας του Flash Lite. Τον Ιούνιο του 2009 κυκλοφορεί το πρώτο Corona SDK beta δωρεάν.



Εικόνα 6: Λογότυπο του Corona SDK.

Λίγο καιρό αργότερα τον Απρίλιο του 2010 κυκλοφορεί για πρώτη φορά το Corona SDK 2.0 beta έχοντας ένα νέο χαρακτηριστικό, την υποστήριξη cross-platform σε iPhone, iPad, Android συσκευές με αποτέλεσμα να επιτρέπει στους προγραμματιστές λογισμικού να χτίσουν κινητές εφαρμογές[4].

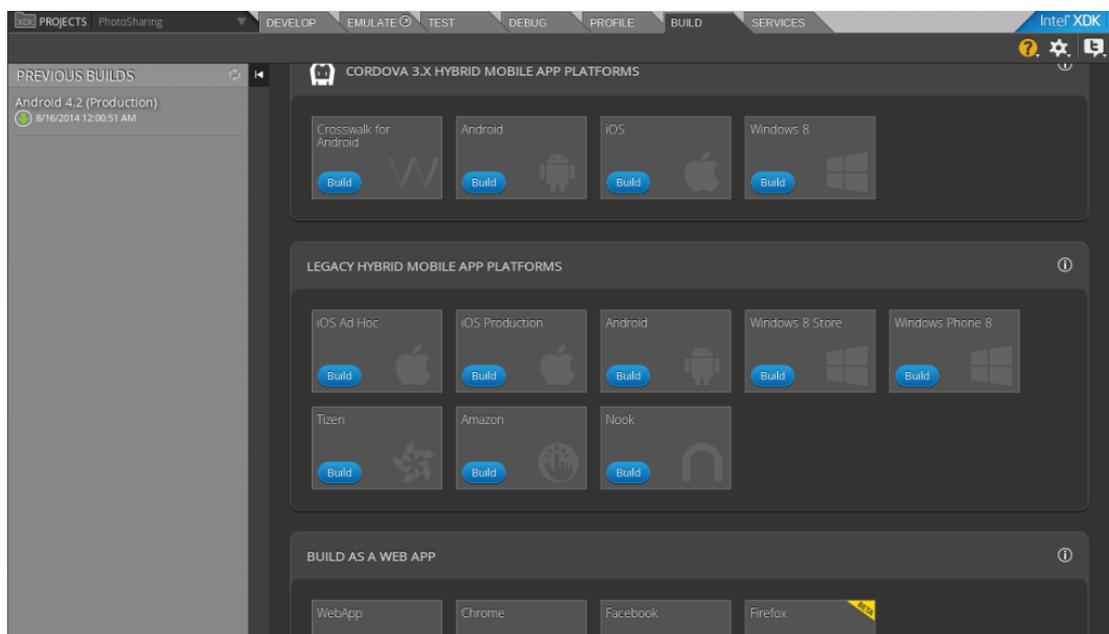
- **Intel XDK**

Μια νέα λύση για την ανάπτυξη της HTML5, CSS, JavaScript είναι το Intel XDK. Είναι ένα cross-platform εργαλείο ανάπτυξης που παρέχει μια απλοποιημένη ροή εργασίας και επιτρέπει στους προγραμματιστές εύκολα και γρήγορα να σχεδιάσουν, να χτίσουν και να διορθώσουν web και υβριδικές εφαρμογές σε διάφορα καταστήματα app.



Εικόνα 7: Λογότυπο του Intel XDK.

Μερικά από τα κύρια χαρακτηριστικά του XDK είναι ότι σου δίνει τη δυνατότητα να κάνεις δωρεάν download και το πρώτο πράγμα που παρατηρεί κανείς, είναι το πόσο εύκολα μπορείς να κάνεις εγκατάσταση σε σχέση με τις υπόλοιπες πλατφόρμες. Το XDK περιλαμβάνει επίσης ένα γραφικό επεξεργαστή, ο οποίος λείπει σε μεγάλο βαθμό από όλες τις κινητές πλατφόρμες HTML5.



Εικόνα 8: Intel XDK Build Tab.

Το Intel XDK έχει τη δυνατότητα να αναπτυχθεί σε οποιαδήποτε πλατφόρμα, επειδή η μεταγλώττιση γίνεται στο cloud. Άλλες πλατφόρμες προσφέρουν μια παρόμοια υπηρεσία για την πολλαπλή ανάπτυξη της πλατφόρμας, αλλά είναι περιορισμένη. Επίσης υποστηρίζει πλαίσια όπως το Bootstrap και jQuery Mobile. Αυτά

τα στοιχεία διεπαφής χρήστη μας επιτρέπουν να οικοδομήσουμε εύκολα και γρήγορα το περιβάλλον εργασίας της εφαρμογής μας. Ένα μειονέκτημα που έχει το XDK είναι ότι δεν μπορούμε να ενημερώσουμε το πρόγραμμα επεξεργασίας της βάσης που είναι ενσωματωμένη με το XDK. Μόλις δημιουργήσουμε ένα app πρέπει να βρούμε ένα τρόπο για να δοκιμάσουμε τη λειτουργικότητά του, αυτό θα πραγματοποιηθεί με τη βοήθεια ενός εξομοιωτή για κινητές συσκευές και έτσι θα έχουμε τη δυνατότητα να δοκιμάσουμε σε πραγματικές συσκευές και τις επιδόσεις του προφίλ χωρίς να χρειάζεται κάποια περαιτέρω εγκατάσταση[5].

- **Sencha Touch 2**

Το Sencha Touch 2 είναι ένα υψηλής απόδοσης πλαίσιο ανοιχτού κώδικα για κινητές εφαρμογές cross-platform. Η πρώτη έκδοση του Sencha Touch διατέθηκε στις 17 Ιουλίου 2010, το Sencha Touch 2 είναι η δεύτερη έκδοση του Sencha Touch που αναπτύχθηκε από Sencha.



Εικόνα 9: Λογότυπο του Sencha Touch.

Το αποτέλεσμα της εφαρμογής είναι υβριδικό και είναι γραμμένο σε γλώσσες Web: HTML5, CSS3 και JavaScript. Στόχος του πλαισίου είναι να επιτρέπει στους προγραμματιστές να δημιουργούν εύκολα και γρήγορα επίγειες κινητές εφαρμογές με HTML5 σε υποστηριζόμενες πλατφόρμες που λειτουργούν με Android, Windows, iOS, BlackBerry, Kindle και Bada. Για την ανάπτυξη εφαρμογών απαιτείται ένας web server ο οποίος θα τρέχει σε τοπικό επίπεδο. Επίσης το συγκεκριμένο πλαίσιο εκμεταλλεύεται την επιτάχυνση υλικού ώστε να παραδώσει μια απίστευτη εμπειρία σε app[6].

- **Xamarin**

Το Xamarin επιτρέπει την ανάπτυξη cross-platform εφαρμογών, το οποίο είναι εξ' ολοκλήρου γραμμένο σε C#. Οι εφαρμογές μοιράζονται τον ίδιο κώδικα για iOS, Android, Windows, Mac κ.ά. και παράγει μια εγγενή εφαρμογή ή ολοκληρωμένη εφαρμογή σε .NET που παρέχει το IDE (MonoDevelop IDE και Xamarin plug-in για το Visual Studio).



Εικόνα 10: Λογότυπο του Xamarin.

Οτιδήποτε μπορούμε να κάνουμε σε Objective-C ή Java, μπορούμε να το κάνουμε σε C# για όλες τις διαθέσιμες πλατφόρμες, παράδειγμα δεν χρειάζεται να εγκαταστήσουμε το Java SDK για την ανάπτυξη σε Android, διότι θα χρησιμοποιήσουμε το εργαλείο Visual Studio που μας παρέχει όλες τις απαραίτητες λειτουργίες για την ανάπτυξη εφαρμογών. Το Xamarin έχει πάνω από 500.000 προγραμματιστές σε περισσότερες από 120 χώρες σε όλο τον κόσμο[7].



Εικόνα 11: Δημιουργία εφαρμογών στην πολυεπίπεδη πλατφόρμα ανάπτυξης Xamarin για iOS, Android με χρήση της γλώσσας προγραμματισμού C#.

- **Unity3D**

Το Unity 3D είναι μια cross-platform 3D μηχανή δημιουργίας παιχνιδιού που αναπτύχθηκε από την Unity Technologies, συμπεριλαμβανομένου και ολοκληρωμένο περιβάλλον ανάπτυξης (IDE). Μπορούμε να το χρησιμοποιήσουμε για την ανάπτυξη παιχνιδιών σε desktop, ιστοσελίδες, κονσόλες και κινητές συσκευές. Η πρώτη έκδοση που ανακοινώθηκε το 2005 ήταν μόνο για Mac OS, στο παγκόσμιο συνέδριο για developers της Apple, μετέπειτα επεκτάθηκε και είναι διαθέσιμο σε περισσότερες από δέκα πλατφόρμες.



Εικόνα 12: Λογότυπο του Unity 3D.

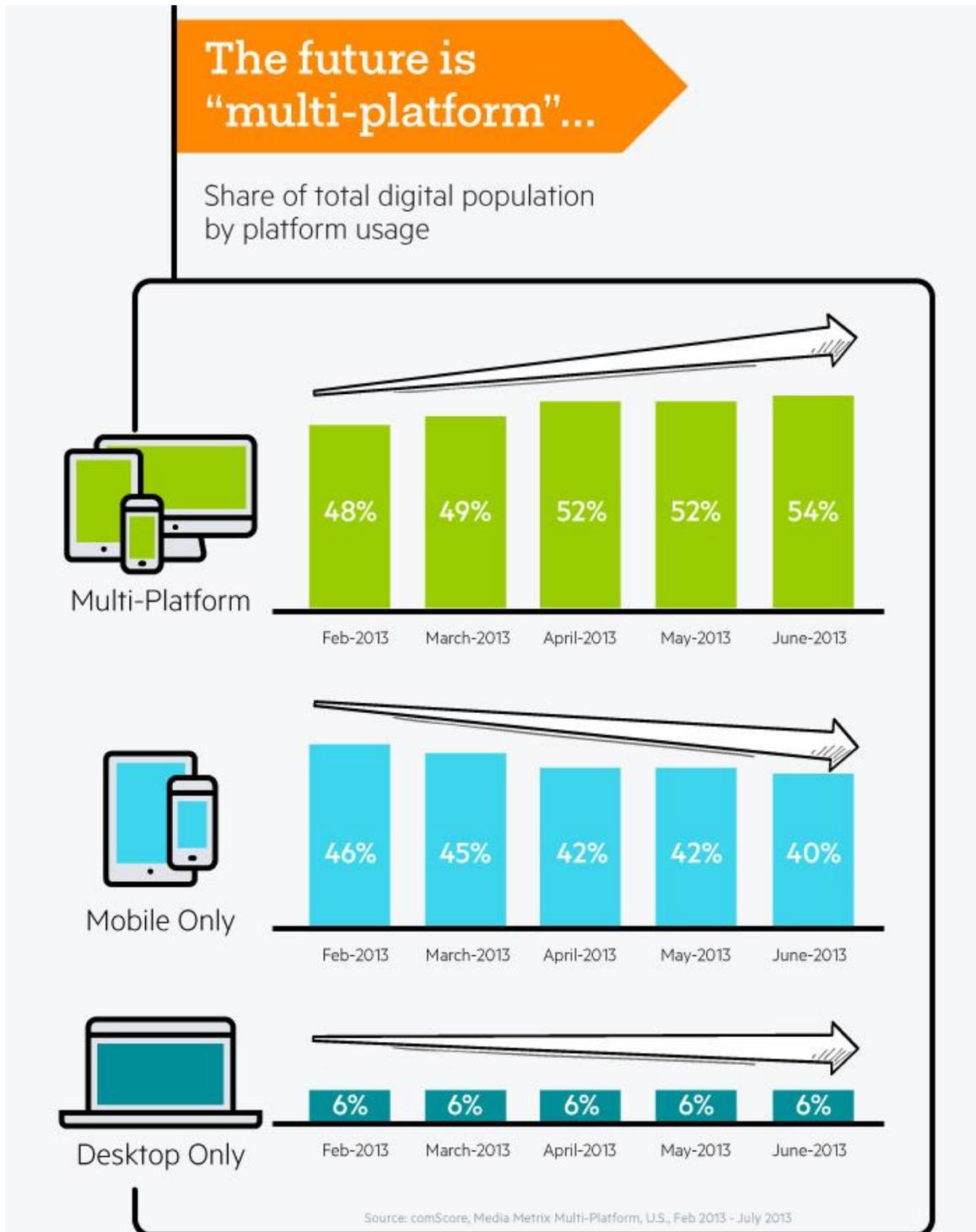
Η πλήρης επαγγελματική σουίτα, το Unity Pro είναι διαθέσιμο από \$1500 το χρόνο ή \$75 το μήνα σε άτομα για προσωπική χρήση ή εταιρείες για επαγγελματική χρήση. Στις 3 Μαρτίου του 2015 με την απελευθέρωση του Unity 5.0, η Unity Technologies έκανε την πλήρη λειτουργία να διατίθεται δωρεάν, συμπεριλαμβανομένων όλων των χαρακτηριστικών και την υποστήριξη. Τέλος, το Unity 3D εργαλείο παρέχει ένα οπτικό βοήθημα που μειώνει την ποσότητα του χρόνου για την ολοκλήρωση του κώδικα και έτσι οι προγραμματιστές μπορούν απλά να σύρουν τους κώδικες από το ένα αντικείμενο στο άλλο. Υποστηρίζει τις περισσότερες πλατφόρμες με μόνη εξαίρεση τους χρήστες των iPhone που τους περιορίζει να το κατεβάσουν για μια δοκιμαστική περίοδο των 30 ημερών[8].

Ακολουθεί ένας συνοπτικός συγκριτικός πίνακας για τις πολυεπίπεδες πλατφόρμες ανάπτυξης (cross platforms)[21].

Εργαλεία Cross-Platform	Υποστηριζόμενες πλατφόρμες	Γλώσσα Προγραμματισμού	Περιβάλλον Ανάπτυξης	Άδεια
Rhodes	Android, BlackBerry, iOS, Windows Phone and Symbian	HTML and Ruby.	RhoStudio IDE - Eclipse, Visual Studio, Netbeans, IntelliJ, and Textemate.	Open-Source
PhoneGap	Android, iOS, webOS, Windows Phone, Symbian, Blackberry and Bada	HTML5, CSS3 and JavaScript.	Eclipse, and XCode.	Open-Source
DragonRad	Android, Windows Mobile, and Blackberry.	D And D	DragonRad Designer	Paid
MoSync	iOS, Windows Phone, Android, JavaME, BlackBerry and Symbian.	C, C++, and Java	Eclipse-based	Open-Source
Appcelerator Titanium	iOS, Android and BlackBerry.	HTML, CSS and JavaScript.	Eclipse-based Titanium Studio.	Open-Source
Sencha Touch 2	iOS, Android, BlackBerry, Kindle and Bada.	HTML5, CSS3 and JavaScript.	Eclipse, and Netbeans	Free license for development, and a paid license for OEM.
jQuery Mobile	Android, BlackBerry, Windows Phone, Bada, palm WebOS, Symbian and MeeGo.	HTML5, CSS.	jReply	Free license.
Xamarin	iOS, Android, and Windows Phone.	C#	MonoDevelop IDE	30 days trial.
Unity 3D	iOS, Android, and Windows Phone	C#, JavaScript and Boo.	Maya, 3DS Max.	30 days trial.
Corona SDK	iOS, Android, Kindle Fire and NOOK.	HTML5.	Lua scripting language.	Free unlimited trial, paid to publish.
Flash Professional	iOS, Android, and BlackBerry	JavaScript Flash Language (JSFL) and ActionScript	AIR 3.2 SDK	30 days trial.

Πίνακας 1: Σύγκριση cross-platform εργαλείων ανάπτυξης κινητών συσκευών.

Στο σχήμα 7 που ακολουθεί παρουσιάζονται τα στατιστικά στοιχεία για τους χρήστες desktop, mobile και multiplatform.



Σχήμα 7: Στατιστικά στοιχεία μεταξύ χρηστών desktop, mobile και multi-platform.

Πηγή: <http://www.telerik.com/blogs/everything-you-wanted-to-know-about-native-hybrid-and-web-apps-but-were-afraid-to-ask> [2].

3. Ανάπτυξη Εφαρμογής για την ιστοπύλη Capital.gr

3.1 Περιγραφή Έργου

Κατά την διάρκεια της πρακτικής μου άσκησης, στην εταιρεία ΔΙΑΔΙΚΤΥΑΚΕΣ ΕΠΙΧΕΙΡΗΣΕΙΣ ΕΛΛΑΔΟΣ ΕΠΕ, μου δώθηκε η ευκαιρία συμμετοχής στην ομάδα ανάπτυξης της διαδικτυακής εφαρμογής για την επιχείρηση ΑΤΤΙΚΕΣ ΕΚΔΟΣΕΙΣ Α.Ε. και συγκεκριμένα για την ιστοσελίδα capital.gr.

Το capital.gr είναι από τις γνωστότερες ιστοσελίδες οικονομικού και ενημερωτικού περιεχομένου στην Ελληνική Αγορά.

Το έργο είχε αντικείμενο την ανάπτυξη διαδραστικής διαδικτυακής εφαρμογής προβολής περιεχομένου και τιμών δεικτών και μετοχών σε πραγματικό χρόνο για κινητές συσκευές.

Η εταιρεία ΔΙΑΔΙΚΤΥΑΚΕΣ ΕΠΙΧΕΙΡΗΣΕΙΣ ΕΛΛΑΔΟΣ ΕΠΕ, χρησιμοποίησε την πλατφόρμα Phone Gap με χρήση Angular.js και Ionic HTML5 πλαίσιο εφαρμογών για την ανάπτυξη της εφαρμογής.

Ακολουθεί η περιγραφή – ανάλυση του έργου :

Phone Gap mobile development framework

Η εφαρμογή υλοποιήθηκε με την χρήση της πλατφόρμας ανάπτυξης native διαδικτυακών εφαρμογών Phone Gap. Η πλατφόρμα επιτρέπει την ανάπτυξη cross platform εφαρμογών αποφεύγοντας την χρησιμοποίηση της αντίστοιχης γλώσσας ανάπτυξης του εκάστοτε λειτουργικού συστήματος κάνοντας χρήση web τεχνολογιών όπως HTML5, CSS3 και Javascript.

Οι εφαρμογές εκτελούνται μέσα από wrappers που «απευθύνονται» σε κάθε πλατφόρμα- λειτουργικό σύστημα. Σαν αποτέλεσμα θα έχουμε μια υλοποίηση η οποία θα «μεταγλωττίζεται» στα αντίστοιχα πρότυπα ώστε να είναι διαθέσιμη σε κάθε λειτουργικό σύστημα.

Για να καλύψουμε τις ανάγκες της κάθε πλατφόρμας ανάπτυξης ενσωματώθηκαν στην υλοποίηση τρία (3) διαφορετικά themes ένα για το κάθε λειτουργικό σύστημα. Ουσιαστικά πραγματοποιήθηκε ένα back end development και τρία (3) διαφορετικά front end, ώστε να επιτευχθεί το καλύτερο δυνατό αποτέλεσμα περιήγησης για τον χρήστη του κάθε λειτουργικού συστήματος (ios, android, windows 8).

3.2 Χαρακτηριστικά υλοποίησης

Ακολουθεί ανάλυση των υποεφαρμογών με βάση τις απαιτήσεις του capital.gr.

Menu

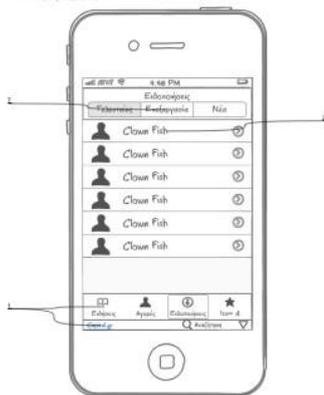
Ορισμός συγκεκριμένου μενού από κύριες κατηγορίες (tags) ειδήσεων το οποίο κάνει οριζόντιο scroll left - right. Στο κάτω μέρος υπάρχουν tabs που θα έχουν κάποιες σημαντικές κατηγορίες ειδήσεων και navigation points για σελίδες Αγορών.

Δημοφιλέστερα άρθρα - Προτεινόμενα κλπ.

Υπάρχουν ειδικές θέσεις (tabbed views) όπου μπορεί ο χρήστης να βλέπει λίστες με τα δημοφιλέστερα άρθρα ή τα άρθρα με τα περισσότερα σχόλια κλπ.

Ειδοποιήσεις

6 - Ειδοποιήσεις άρθρων



Υπάρχουν οθόνες που θα μπορεί ο χρήστης να προσθαφαιρεί σύμβολα μετοχών για τα οποία θα λαμβάνει push notifications. Στην οθόνη καταχώρησης ειδοποίησης ο χρήστης θα μπορεί :

- επιλέγει ένα ή περισσότερα σύμβολα μετοχής
- επιλέγει τις ημέρες που θέλει να ισχύει η ειδοποίηση
- επιλέγει ώρες που δε θα ισχύει η ειδοποίηση

Εκτός από σύμβολα μετοχών μελλοντικά εκτός από σύμβολα μετοχών θα μπορεί να βάζει keywords δικά του, οπότε αυτό πρέπει εξ αρχής έχει προβλεφθεί κατά την υλοποίηση της εφαρμογής . Υπάρχει στο κάτω μέρος tab το οποίο θα αφορά τις σελίδες ειδοποιήσεων.

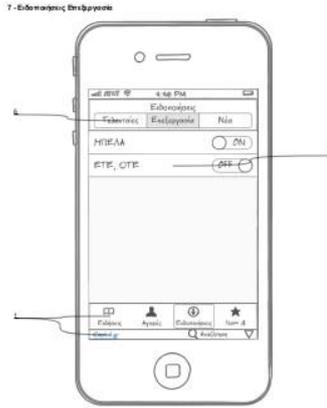
Όταν "σκάει" ειδοποίηση εμφανίζεται κόκκινο νουμεράκι στο tab + κάποιου είδους "φούσκα" στην εφαρμογή. Ο χρήστης πατώντας το tab εμφανίζεται η λίστα με τις ειδοποιήσεις που έχουν τρέξει.

1 - Αρχική Οθόνη



8 - Ειδοποίηση άρθρου

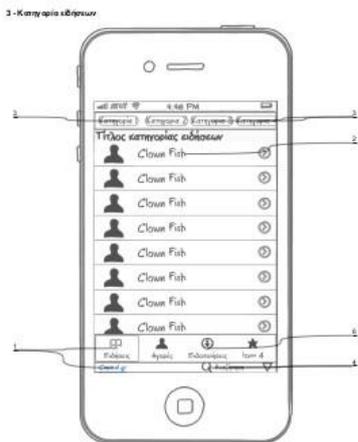




Λίστα ειδήσεων / κατηγορία (tag)

Η οθόνη που εμφανίζει λίστες έχει δυνατότητα να μπορεί να προβάλει rows ή tiles με:

- Τίτλο + (ημ/νία) + (αρ.σχολίων)
- Τίτλος + Περίληψη + (ημ/νία) + (αρ.σχολίων)
- Τίτλος + Περίληψη + Εικόνα + (ημ/νία) + (αρ.σχολίων)



Υπάρχει η δυνατότητα από τη λίστα ειδήσεων σε κάποιες "ειδήσεις" ο χρήστης να τις πατάει και αντί να ανοίγει η οθόνη εμφάνισης είδησης να ανοίγει ένα WebView με συγκεκριμένο url. Οπότε το service που τραβάει τα στοιχεία του news lists για κάθε news item έχει κάποιο field που ορίζει αν κάνει navigate σε σελίδα news ή σε webview. Σε περίπτωση που είναι WebView το navigation γίνεται ακριβώς με τον ίδιο τρόπο που θα γίνεται και για την ανάγνωση μιας είδησης. Δηλαδή Scroll από δεξιά προς τα αριστερά. Όταν ο χρήστης κάνει scroll μια λίστα και φτάνει στο τέλος εμφανίζεται ένδειξη στο κάτω μέρος της λίστας που τον ενημερώνει ότι φορτώνει την επόμενη σελίδα κ.ο.κ. Όταν

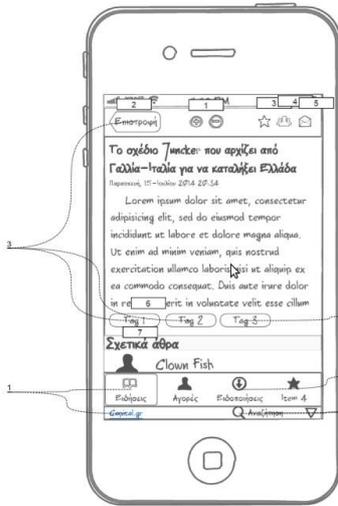
φορτώνεται μία σελίδα μετά από 3 sec και εφόσον ο χρήστης δεν έχει αλλάξει οθόνη φορτώνεται στο background η επόμενη σελίδα. Οι εικόνες που εμφανίζονται στις λίστες είναι lazy loaded και θα κασάρονται στο local storage της συσκευής για 24 ώρες.

Κάνοντας κλικ πάνω στην εικόνα ανοίγει modal window με την εικόνα σε μεγέθυνση. (Αυτό υλοποιείτε μόνο σε συγκεκριμένες εικόνες, που επιθυμεί ο πελάτης. Η πληροφορία υπάρχει σε κάποιο field του object που επιστρέφει το service των ειδήσεων).

Κάνοντας swipe down στη λίστα μιας κατηγορίας ειδήσεων πρέπει να κάνει reload τη λίστα Navigation. Το navigation είναι της μορφής drill down δηλαδή από λίστα -> detail οθόνη και αντίστροφα. Δε υπάρχουν άσχετα popups κλπ., έτσι ο χρήστης δεν μπερδεύεται. Επίσης σε όλες τις οθόνες, όταν ο χρήστης πατάει το λόγγο του Capital ή κάποιου είδους icon παρακάμπτεται όλο το navigation stack και επιστρέφει στην αρχική οθόνη.

Οθόνη άρθρου

2 - Άρθρο



- Υπάρχει η δυνατότητα να κάνει resize τα fonts (μεγαλύτερα - μικρότερα)
- Υπάρχει η δυνατότητα να κάνει share σε FB + Twitter + Google+
- Υπάρχει η δυνατότητα αξιολόγησης (Αστεράκια από 1 - 5)
- Υπάρχει η δυνατότητα αποστολής email.
- Υπάρχει η δυνατότητα αποθήκευσης σε reading list.

Στο κάτω μέρος του άρθρου εμφανίζονται σχετικά Tags - Μετοχές που ο χρήστης μπορεί να πατά και να πηγαίνει στις αντίστοιχες οθόνες εμφάνισης περισσότερων ειδήσεων που αφορούν το tag ή τη μετοχή που επιλέγει. Στο κάτω μέρος επίσης υπάρχει λίστα με σχετικά άρθρα με αυτό. Αν

το άρθρο έχει ενσωματωμένα URL's όταν ο χρήστης κάνει κλικ επάνω σε αυτά τον οδηγεί σε web view. Κατ' αυτό τον τρόπο ο χρήστης δεν εγκαταλείπει την εφαρμογή.

Αρχική Οθόνη

Στην αρχική υπάρχουν 2 ή 3 διαφορετικά group από ειδήσεις οι οποίες θα εμφανίζονται σε λίστα ή tiles. Επίσης στην αρχική οθόνη εμφανίζονται οι ειδοποιήσεις που αφορούν το χρήστη.

Οθόνη αναζήτησης

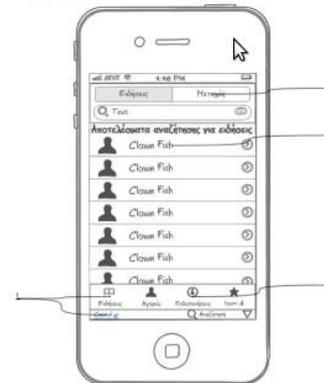
Η αναζήτηση είναι προσβάσιμη από οποιαδήποτε οθόνη της εφαρμογής.

Χωρίζεται σε δύο sections:

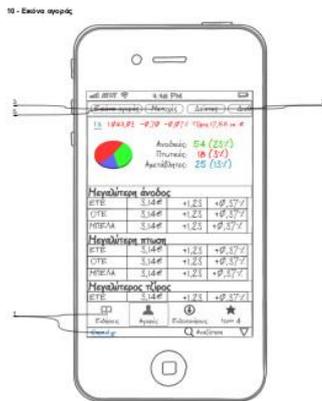
- Αναζήτηση είδησης
- Αναζήτηση μετοχής

Η αναζήτηση είδησης εμφανίζει αποτελέσματα σε μορφή list σαν την οθόνη με τις λίστες ειδήσεων. Η αναζήτηση μετοχής εμφανίζει τις μετοχές + όνομα εταιρίας σε λίστα όπου όταν ο χρήστης πατάει επάνω σε αυτή πηγαίνει σε ειδική οθόνη που εμφανίζει αναλυτικά στοιχεία για τη μετοχή.

4 - Αναζήτηση Ειδήσεων



Οθόνη επισκόπησης αγοράς



Εδώ εμφανίζεται η τιμή του ΓΔ και πίνακες με στοιχεία της αγοράς όπως:

- Ανοδικές - Πτωτικές - Σταθερές μετοχές.
- Μεγαλύτερη άνοδος
- Μεγαλύτερη πτώση
- Μεγαλύτερος τζίρος κ.ο.κ.(ουσιαστικά πρέπει να φτιαχτεί ένα view που έχει δυνατότητα να εμφανίζει πίνακες με στοιχεία μετοχών.
- Στο πάνω μέρος της οθόνης υπάρχει οριζόντιο scrolling menu με επιλογές που στέλνουν σε οθόνες που αφορούν τις αγορές. πχ. δείκτες, Forex, Commodities και ότι άλλο χρειαστεί.

Οθόνη Ισοτιμιών

Θα εμφανίζονται σε tiles (όχι λίστα) οι τιμές των ισοτιμιών.

Οθόνη μετοχής



Εμφανίζει αναλυτικά στοιχεία για μία μετοχή.

Τα στοιχεία αυτά είναι:

1. Τρέχουσες τιμές - δείκτες που αφορούν τη μετοχή.
2. Γράφημα intraday + ιστορικό (κάνοντας κλικ πάνω στο γράφημα γίνεται full screen). Τα γραφήματα θέλουμε να είναι dynamic και όχι στατικές εικόνες.
3. Σχετικές ειδήσεις για τη μετοχή.
4. Πλήκτρο για προσθήκη στη λίστα παρακολούθησης.

Οθόνη λίστας παρακολούθησης μετοχών

Εμφανίζεται η λίστα με τις τρέχουσες τιμές όλων των μετοχών που έχω επιλέξει ο χρήστης. Υπάρχει η δυνατότητα για sorting ανάλογα με τιμή, άνοδο - πτώση, τζίρο κλπ. Όταν κάνει swipe down ο χρήστης ενημερώνεται η λίστα.

3.3 Τεχνολογίες-Πλαίσια εφαρμογής που χρησιμοποιήθηκαν για την υλοποίηση του έργου Capital.gr

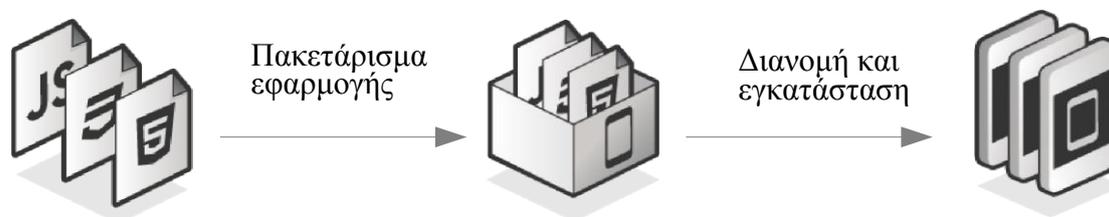
3.3.1 Πολυεπίπεδη πλατφόρμα ανάπτυξης εφαρμογής PhoneGap

Ιστορικά στοιχεία της πολυεπίπεδης πλατφόρμας phonegap

Το Phonegap δημιουργήθηκε το 2009 από μια εταιρεία που ονομάζεται Nitobi, αναπτύχθηκε για πρώτη φορά σε μια iPhoneDevCamp εκδήλωση στο Σαν Φρανσίσκο. Στις 4 Οκτωβρίου του 2011 η Adobe ανακοίνωσε την απόκτηση του Nitobi Software και μαζί με αυτό τα δικαιώματα για την μάρκα phonegap και η ανοιχτή βασική πηγή δωρήθηκε στο Apache Software Foundation υπό την επωνυμία Cordova. Έπειτα τον Σεπτέμβριο του 2012 η Phonegap Build υπηρεσία επιτρέπει σε έναν προγραμματιστή να ανεβάσει έναν πηγαίο κώδικα σε μια cloud compiler που δημιουργεί εφαρμογές για κάθε πλατφόρμα.

Πολυεπίπεδη πλατφόρμα phonegap;

Το Phonegap είναι ένα πλαίσιο (framework) ανάπτυξης ανοιχτού κώδικα για την κατασκευή λογισμικού σε υπολογιστές και εφαρμογές σε κινητά τηλέφωνα το οποίο βασίζεται σε τεχνολογίες web όπως HTML, CSS, JavaScript.



Εικόνα 13: Αρχιτεκτονικό διάγραμμα για πολυεπίπεδη πλατφόρμα (PhoneGap).

Κόστος πλατφόρμας

Το κόστος το οποίο χρειάζεται να ξοδέψει κάποιος είναι μηδενικό γιατί η πλατφόρμα προσφέρεται δωρεάν. Αυτό σημαίνει ότι οι εταιρείες και οι προγραμματιστές μπορούν να χρησιμοποιήσουν το Phonegap για τις κινητές εφαρμογές που είναι δωρεάν και ανοιχτού κώδικα. Παράλληλα του επιτρέπει να δημιουργεί εφαρμογές που τρέχουν σε μια ποικιλία φορητών συσκευών.

Στόχος πλατφόρμας phonegap

Ο κύριος στόχος του είναι να επιτυγχάνει στο μεγαλύτερο βαθμό όσο αυτό είναι εφικτό τη δυνατότητα διεπαφής του χρήστη και της εφαρμογής με τις τεχνολογίες web που αναφέραμε παραπάνω.

Υποστηριζόμενες πλατφόρμες του phonegap

- Android
- iOS
- Windows 8
- Windows Phone 7 & 8
- Blackberry 5.x +
- Symbian
- WebOS
- Tizen

Native χαρακτηριστικά

Το phonegap έρχεται να δώσει μια ακόμα λύση, κλείνοντας το φάσμα το οποίο υπήρχε με τα native χαρακτηριστικά των κινητών συσκευών. Μια web εφαρμογή δεν υποστηρίζει πλήρως πολλές λειτουργίες της συσκευής, όπως είναι η πρόσβαση στα αρχεία ή στη κάμερα. Το phonegap χρησιμοποιεί ένα μηχανισμό γεφύρωσης που επιτρέπει στην JavaScript την προβολή web σε μια εγγενή εφαρμογή. Το phonegap περιέχει πλήρης plugins για να υποστηρίξουν τις δυνατότητες της κάθε συσκευής, όπως φωτογραφική μηχανή, αναπαραγωγή πολυμέσων και εγγραφή, επαφές, σύστημα αρχείων, διαθεσιμότητα του δικτύου, Bluetooth, σύνδεση Facebook, ημερολόγιο, σάρωση barcode.

Πλεονεκτήματα

1. Επιτρέπει στους προγραμματιστές να δημιουργήσουν εφαρμογές που χρησιμοποιούν τις τεχνολογίες web μέσω των γλωσσών HTML5 και CSS3 για τη διασύνδεση και τη Javascript για λογική αντί ξεχωριστή γλώσσα για κάθε συσκευή.
2. Το Phonegap στηρίζει την ανάπτυξη εφαρμογών σε όλα σχεδόν τα λειτουργικά συστήματα.
3. Οι εφαρμογές που δημιουργούνται με τη χρήση Phonegap μπορεί να υποστηρίζονται από πολλές κινητές συσκευές, χωρίς να προκύπτουν

παραμετροποιήσεις. Με αποτέλεσμα να γίνεται εξοικονόμηση σε χρόνο και σε χρήμα χωρίς να κάνει την ίδια διαδικασία πολλές φορές.

Μειονεκτήματα

1. Η χρήση της τεχνολογίας web έχει οδηγήσει σε πολλές εφαρμογές Phonegap να τρέχουν πιο αργά σε σύγκριση με εγγενείς εφαρμογές με παρόμοια λειτουργικότητα.
2. Το phonegap δεν περιέχει κάποια έτοιμη βάση διεπαφής χρήστη(UI) περιεχομένου με αποτέλεσμα πολλές φορές η υλοποίηση να κρατάει περισσότερο.

Προϋποθέσεις

Η εγκατάσταση του PhoneGap απαιτεί την προηγούμενη εγκατάσταση των καταλλήλων SDK της εκάστοτε πλατφόρμας που θέλουμε να δουλέψουμε. Για τη δημιουργία ενός έργου σε οποιαδήποτε πλατφόρμα θα πρέπει πρώτα να εκτελεστεί το περιβάλλον της γραμμής εντολών από το ίδιο το μηχάνημα που υποστηρίζει το SDK της πλατφόρμας. Το CLI(Command Line Interface) υποστηρίζει τους ακόλουθους συνδυασμούς:

- IOS (Mac)
- Amazon Fire OS (Mac, Linux, Windows)
- Android (Mac, Linux, Windows)
- BlackBerry 10 (Mac, Linux, Windows)
- Windows Phone 8 (Windows)
- Windows (Windows)
- Firefox OS (Mac, Linux, Windows)

Στο Mac, η γραμμή εντολών είναι διαθέσιμη μέσω του τερματικού (Terminal).

Οδηγός εγκατάστασης πλατφόρμας Phonegap

Αρχικά, για την εγκατάσταση του Phonegap χρειάζεται το Node.js, το οποίο τρέχει σε cross-platform περιβάλλον και είναι open source για εφαρμογές δικτύωσης. Οι εφαρμογές που είναι σε Node.js είναι γραμμένες σε JavaScript και μπορούν να εκτελεστούν σε Microsoft Windows, OS X, Linux. Μετά την εγκατάσταση θα πρέπει να είμαστε σε θέση να επικαλεστούμε NPM και code στη γραμμή εντολών, αν θέλουμε μπορούμε προαιρετικά να χρησιμοποιήσουμε ένα εργαλείο όπως το NVM ή nave για να διαχειριστεί την εγκατάσταση του Node.js.

Ιστορία

Το Node.js δημιουργήθηκε και δημοσιεύτηκε για πρώτη φορά σε Linux το 2009, αναπτύχθηκε και συντηρήθηκε από τον Ryan Dahl, ο οποίος εργαζόταν στην εταιρεία Joyent η οποία τον χρηματοδότησε[9].



Εικόνα 14: Λογότυπο του node.js

Δεύτερο βήμα, λήψη και εγκατάσταση ενός GIT client, δεν είναι υποχρεωτικό μας βοηθάει αν θέλουμε να ενσωματώσουμε συγκεκριμένα plugins.

Τρίτο βήμα, τοποθετούμε το Apache Cordova χρησιμοποιώντας NPM.

Διευκρίνιση: Είτε Cordova είτε Phonegap είναι το ίδιο πράγμα, απλά το Phonegap ανήκει πλέον στην Adobe.

- σε OS X και Linux:

```
$ sudo npm install -g cordova
```

Στο OS X και Linux χρησιμοποιώντας την NPM εντολή με SUDO μέσω του Terminal γίνεται εγκατάσταση ως Super User.

Ανοίγοντας τη γραμμή εντολών(Command Line) από τον υπολογιστή μας, το γνωστό (DOS) των Windows και εκτελούμε την ακόλουθη εντολή.

- στα Windows:

```
C:\>npm install -g cordova
```

Μόλις ολοκληρωθεί η εγκατάσταση μπορείτε να επικαλεστείτε στη γραμμή εντολών τη λέξη phonegap για περαιτέρω βοήθεια.

Η σημαία `-g` παραπάνω αναφέρεται στο ότι πρόκειται να εγκαταστήσουμε το Cordova παγκόσμια (global), διαφορετικά θα εγκατασταθεί στον υποκατάλογο του τρέχοντος καταλόγου εργασίας.

Οδηγός χρήσης του Phonegap

Δημιουργούμε ένα app και το ονομάζουμε όπως θέλουμε, εδώ συγκεκριμένα το ονομάσαμε «hello», μόλις δημιουργηθεί, πάμε μέσα στον υπολογιστή και στον προορισμό που έχουμε επιλέξει θα έχει δημιουργηθεί ένας φάκελος με διάφορα αρχεία. Ο www περιέχει την αρχική σελίδα της εφαρμογής μας, μαζί με διάφορους πόρους CSS, JS, IMG. Το config.xml αρχείο περιέχει σημαντικά δεδομένα που απαιτούνται για την παραγωγή και τη διανομή της εφαρμογής.

```
$ cordova create hello
```

Με την επόμενη εντολή του λέμε να μπει στο φάκελο της εφαρμογής.

```
$ cd hello
```

Για να μπορέσουμε να κατασκευάσουμε το έργο θα πρέπει να καθορίσουμε μια σειρά από πλατφόρμες. Για να εκτελεστούν οι ακόλουθες εντολές πρέπει το μηχάνημα μας να υποστηρίζει και να έχει ήδη εγκαταστήσει το κάθε SDK.

Οι παρακάτω εντολές είναι δυνατόν να εκτελεστούν από έναν υπολογιστή Mac:

```
$ cordova platform add ios
$ cordova platform add amazon-fireos
$ cordova platform add android
$ cordova platform add blackberry10
$ cordova platform add firefoxos
```

Εκτελούμε οποιαδήποτε πλατφόρμα μέσα από έναν υπολογιστή Windows, όπου WP παραπέμπει σε διαφορετικές εκδόσεις των Windows Phone.

```
$ cordova platform add wp8
$ cordova platform add windows
$ cordova platform add amazon-fireos
$ cordova platform add android
$ cordova platform add blackberry10
$ cordova platform add firefoxos
```

Εκτελώντας την παρακάτω εντολή, ελέγχουμε την τρέχουσα σειρά της πλατφόρμας μας.

```
$ cordova platforms ls
```

Διευκρίνιση: Οι εντολές platforms και platform είναι συνώνυμες.

Ακολουθώντας τις παρακάτω εντολές αφαιρούμε μια πλατφόρμα:

```
$ cordova platform remove blackberry10
$ cordova platform rm amazon-fireos
$ cordova platform rm android
```

Με την ακόλουθη προαιρετική εντολή χτίζουμε το έργο:

```
$ cordova build
```

Κάνει build στο πεδίο εφαρμογής της κάθε κατασκευής σε συγκεκριμένες πλατφόρμες:

```
$ cordova build ios
```

Δοκιμή της εφαρμογής σε εξομοιωτή ή συσκευή

Τα SDK για κινητές πλατφόρμες έρχονται πακέτο με εξομοιωτές(emulators) που εκτελούν μια εικόνα της συσκευής, για να δούμε πως αλληλοεπιδρά με πολλά χαρακτηριστικά της πλατφόρμας. Εκτελώντας μια εντολή όπως τη παρακάτω, βλέπουμε μέσα σε εξομοιωτή μια συγκεκριμένη πλατφόρμα:

```
$ cordova emulate android
```

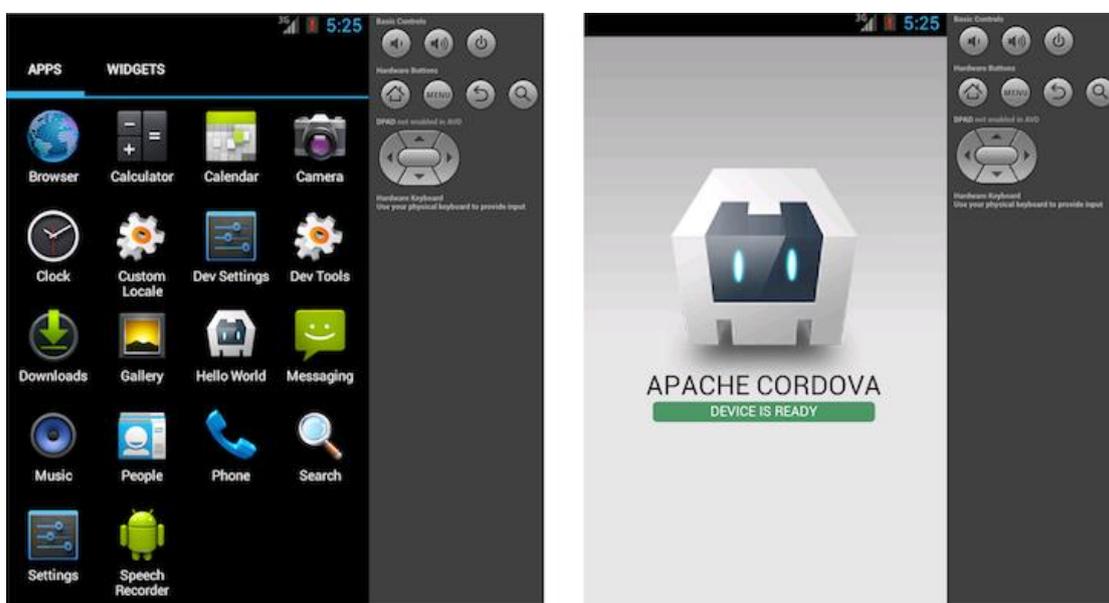
Διευκρίνιση: Η υποστήριξη σε εξομοιωτή δεν είναι προς το παρόν διαθέσιμη για το Amazon Fire OS.

Ξεκινάμε το Android SDK και εκτελούμε μια συγκεκριμένη εικόνα της συσκευής:



Εικόνα 15: Αρχική σελίδα Android simulator.

Στη συνέχεια με την εντολή Cordova emulate ανανεώνει την εικόνα του εξομοιωτή για να εμφανίσει την πιο πρόσφατη εφαρμογή, η οποία είναι διαθέσιμη για έναρξη από την αρχική οθόνη:



Εικόνα 16: Αρχική σελίδα Android simulator με την χρήση πλατφόρμας PhoneGap.

Εναλλακτικά, μπορούμε να συνδέσουμε την εφαρμογή στον υπολογιστή μας και ελέγχουμε την εφαρμογή απευθείας[10]:

```
⌘ cordova run android
```

3.4 Πλαίσιο ανάπτυξης Angular.js

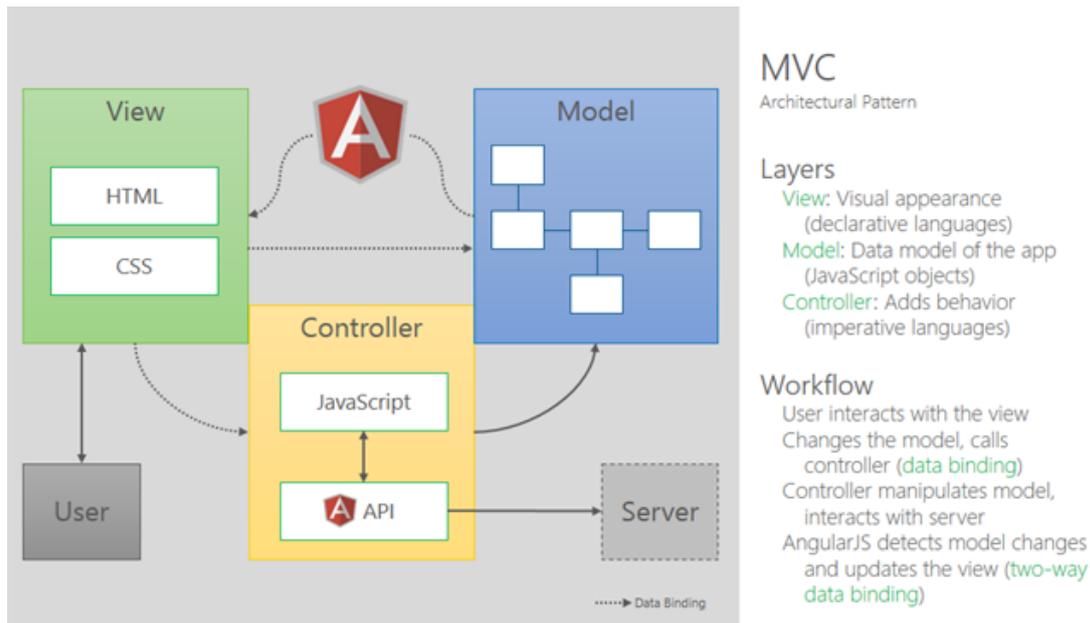
Ιστορία

Η πρώτη έκδοση σε AngularJS κυκλοφόρησε το 2012. Το AngularJS αναπτύχθηκε αρχικά το 2009 από έναν υπάλληλο της Google, τον Misko Hevery οποίος άρχισε να εργάζεται πάνω σε αυτό. Η ιδέα του τελικώς αποδείχθηκε πολύ καλή και το έργο του πλέον υποστηρίζεται από την Google.



Εικόνα 17: Λογότυπο του AngularJS.

Το AngularJS είναι ένα open-source πλαίσιο ανάπτυξης web, το οποίο συντηρείται από την Google και είναι γραμμένο σε JavaScript. Το AngularJS είναι χτισμένο γύρω από την πεποίθηση ότι ο δηλωτικός προγραμματισμός θα πρέπει να χρησιμοποιείται για την κατασκευή της διεπαφής χρήστη και τη σύνδεση στοιχείων λογισμικού. Λέγοντας δηλωτικός προγραμματισμός (declarative programming) εννοούμε τις επιθυμητές ιδιότητες ενός ζητούμενου αποτελέσματος, ένα παράδειγμα είναι η γλώσσα HTML και η CSS κ.ά. Στόχος του είναι να απλοποιεί την ανάπτυξη και τη δοκιμή των εφαρμογών με την παροχή ενός πλαισίου για client-side, model-view-controller (MVC) αρχιτεκτονική. Όπως φαίνεται και στην παρακάτω εικόνα το MVC αποτελείται από τρία βασικά μέρη: το οπτικό μέρος, το μοντέλο δεδομένων που περιέχει αντικείμενα σε JavaScript της εκάστοτε εφαρμογής και έναν ελεγκτή που προσθέτει συμπεριφορά σε προστακτικές γλώσσες (imperative languages). Εάν ο χρήστης θέλει να αλληλοεπιδράσει με το εικαστικό μέρος, καλεί τον ελεγκτή (controller) και χειρίζεται το μοντέλο το οποίο αλληλοεπιδρά με τον διακομιστή (server). Το AngularJS ανιχνεύει τις αλλαγές του μοντέλου και ενημερώνει το view.



Εικόνα 18: Μοτίβο αρχιτεκτονικής MVC.

Παράδειγμα σε AngularJS

Το AngularJS διανέμεται ως αρχείο JavaScript και μπορεί να προστεθεί σε μια ιστοσελίδα με μια ετικέτα script:

```
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>
```

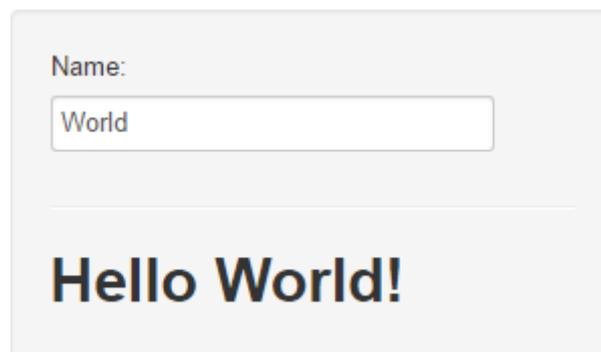
Το παρακάτω κομμάτι κώδικα είναι ένα HTML5 αρχείο το οποίο περιέχει ορισμένες εντολές με μπλε background σε Angular. Η `ng-app` λέει στο AngularJS να είναι ενεργό σε αυτό το τμήμα της σελίδας, στην συγκεκριμένη περίπτωση στο σύνολο του εγγράφου. Η επόμενη εντολή `angular.min.js` φορτώνει ένα αρχείο JavaScript. Στη συνέχεια η `ng-model` συνδέει τη μορφή και το μοντέλο, αυτό πρακτικά σημαίνει ότι οι όποιες αλλαγές πραγματοποιηθούν στον έλεγχο, ενημερώνονται τα δεδομένα στο μοντέλο μας και όταν αλλάξει το μοντέλο ενημερώνει τον έλεγχο. Η τελευταία εντολή η οποία εμπεριέχεται μέσα σε `{ {} }` είναι ένας δηλωτικός τρόπος καθορισμού σύνδεσης δεδομένων θέσεις στην HTML. Το AngularJS θα ενημερώσει αυτόματα το κείμενο αυτό κάθε φορά που θα μεταβάλλεται το value του input και θα αλλάζει αναλόγως, όπως φαίνεται και στην αμέσως επόμενη εικόνα[11].

```

1. <!doctype html>
2. <html ng-app>
3.   <head>
4.     <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.15/
angular.min.js"></script>
5.   </head>
6.   <body>
7.     <div>
8.       <label>Name:</label>
9.       <input type="text" ng-model="yourName" placeholder="Enter a name here">
10.      <hr>
11.      <h1>Hello {{yourName}}!</h1>
12.    </div>
13.  </body>
14. </html>

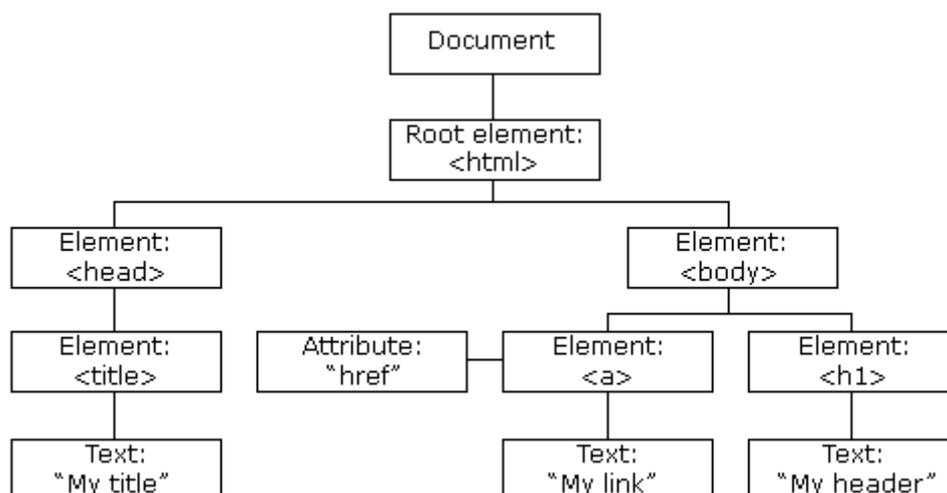
```

Εικόνα 19: Κώδικας HTML με ενσωμάτωση Angular.js



Εικόνα 20: Αποτέλεσμα κώδικα εφαρμογής με χρήση του Angular.js

Το Angular χρησιμοποιεί ένα πρότυπο W3C (World Wide Web Consortium) για τη σωστή δομή ενός HTML αρχείου με την ονομασία DOM (Document Object Model) το οποίο κατασκευάζει ένα δέντρο από αντικείμενα (objects):



Εικόνα 21: HTML DOM δέντρο αντικειμένων.

Το πρότυπο του W3C DOM χωρίζεται σε τρία διαφορετικά τμήματα:

1. Core (Πυρήνας) DOM // πρότυπο μοντέλο για όλους τους τύπους εγγράφων
 2. XML DOM // πρότυπο μοντέλο για έγγραφα XML
- HTML DOM // πρότυπο μοντέλο για έγγραφα HTML

3.5 Πλαίσιο ανάπτυξης ionic HTML5

Το ionic δημιουργήθηκε από τους προγραμματιστές Max Lynch, Ben Sperry και Adam Bradley του Drifty ΣΙΑ το 2013. Πριν δημιουργηθεί το ionic το Drifty περιελάμβανε drag-and-drop εργαλεία διεπαφής με τη χρήση jQuery Mobile και Bootstrap, λαμβάνοντας όμως υπόψη τους πελάτες τους, οι οποίοι προσπαθούσαν να χτίσουν κινητές εφαρμογές, η ομάδα αποφάσισε να δημιουργήσει ένα δικό της πλαίσιο που θα επικεντρώνεται στις προδιαγραφές κατασκευής σύγχρονων ιστοσελίδων. Έπειτα τον Μάρτιο του 2014 κυκλοφόρησε η πρώτη έκδοση σε beta και πλέον χρησιμοποιείται από όλους τους προγραμματιστές σε όλο τον κόσμο.



Εικόνα 22: Λογότυπο του ionic.

Το ionic είναι ένα front-end πλαίσιο για τη δημιουργία υβριδικών κινητών εφαρμογών που βασίζονται σε γλώσσες web όπως η HTML5, CSS και JavaScript. Το ionic εστιάζεται κυρίως στην αλληλεπίδραση του UI (User Interface), στην εμφάνιση και στην αίσθηση της εκάστοτε εφαρμογής. Το ionic αξιοποιεί το HTML και το CSS και συνυπάρχει με το AngularJS αντί για jQuery το οποίο αξιοποιεί την JavaScript ούτως ώστε να λειτουργήσει στο μέγιστο των δυνατοτήτων του, διότι βασίζεται σε φυσική επιτάχυνση υλικού και όχι σε εκτεταμένη DOM χειραγώγηση[12].



Εικόνα 23: Συνύπαρξη του ionic και του AngularJS.

3.6 Εργαλεία που χρησιμοποιήθηκαν

3.6.1 HTML

Η HTML είναι το ακρωνύμιο των λέξεων (Hyper Text Markup Language) η οποία είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες. Η HTML δεν είναι γλώσσα προγραμματισμού, αλλά δημιουργεί ένα σύνολο από εικόνες, κείμενα, βίντεο κ.ά. στον browser. Η συγκεκριμένη γλώσσα χρησιμοποιεί ένα σύνολο από ετικέτες τα λεγόμενα tags τα οποία είναι λέξεις κλειδιά που κλείνονται μέσα σε σύμβολα (μεγαλύτερο από και μικρότερο από) < >. Χρησιμεύουν για την εκάστοτε μορφοποίηση κείμενου, την δημιουργία εικόνων, βίντεο και την δημιουργία συνδέσμων (links) που ενδεχομένως θα περιέχει το site μας. Το όνομα του αρχείου έχει επέκταση .html .

Μερικά παραδείγματα των tags:

<html> </html>

<title> </title>

<p> </p>

<h1> </h1>

<button> </button>

<form> </form>

<video> </video>

<audio> </audio>

3.6.2 Χρησιμοποίηση HTML5

Η HTML5 είναι η πιο πρόσφατη έκδοση γλώσσα σήμανσης για τον Παγκόσμιο Ιστό(www) και πλέον έχει αντικαταστήσει την HTML 4.01. Ο όρος HTML5 αναφέρεται σε ένα σύνολο τεχνολογιών και όχι απλά σε μια νέα έκδοση της γλώσσας HTML. Ο τρόπος έναρξης ενός αρχείου HTML5 είναι: <!DOCTYPE html> έτσι ξεκινάμε για να δηλώσουμε ότι πρόκειται να ακολουθήσει κώδικας HTML5. Ένας από τους λόγους που χρησιμοποιήσα την συγκεκριμένη γλώσσα είναι:

1. Η μεγαλύτερη ευκολία στον προγραμματιστή
2. Περισσότερες δυνατότητες με λιγότερες γραμμές κώδικα.

Επίσης αξίζει να σημειωθεί ότι επιτρέπει στους προγραμματιστές να δημιουργήσουν mobile web sites τα οποία σου δίνουν ταυτόχρονα τις ιδιότητες της εφαρμογής και του Ίντερνετ. Οι συγκεκριμένες εφαρμογές τα τελευταία χρόνια είναι που έχουν τη μεγαλύτερη ζήτηση, με σημαντικά οικονομικά οφέλη σε Ελλάδα και εξωτερικό.

Η κύρια δομή για να ξεκινήσουμε μια ιστοσελίδα σε HTML5:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title> </title>
  </head>
  <body>
    <div>
      <header>
        <h1> </h1>
      </header>

      <footer> </footer>
    </div>
  </body>
</html>
```

3.6.3 CSS

Περνάμε σε μια γλώσσα που ουσιαστικά είναι υπεύθυνη για την μορφή και την εμφάνιση ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης (HTML) και γενικώς για το design της σελίδας. Η CSS (Cascading Style Sheets) είναι προορισμένη να διαμορφώνει περισσότερα χρώματα και χαρακτηριστικά με αποτέλεσμα να σου δημιουργεί στιλιστικά μια όμορφη και ευχάριστη ιστοσελίδα. Με το συγκεκριμένο εργαλείο που χρησιμοποιούμε για την κατασκευή ενός website έχει ως βασικό σκοπό την βελτίωση της ταχύτητας φόρτωσης δημιουργώντας ένα καινούργιο αρχείο με τη μορφή .css και με αυτόν τον τρόπο επιτυγχάνουμε την μείωση του χώρου που μπορεί να καταλαμβάνει ένα site σε έναν server.

3.6.4 Χρησιμοποίηση CSS3

Το CSS3 είναι το τελευταίο πρότυπο για CSS το οποίο είναι πλήρως συμβατό με τις προηγούμενες εκδόσεις του.

Παράδειγμα κώδικα css:

```
body {background-color: red;}
h1 {color: #d0e4fe; font-size: 30px;}
hr {color: #b0c4de;}
p {font-size: 8px; margin-left: 20px;}
a: link {color: blue}
```

a: visited {color: red}

a: hover {color: green}

a: active {color: yellow}

3.6.5 JavaScript

Η JavaScript είναι μια scripting γλώσσα η οποία έχει ως περιβάλλον εκτέλεσης τον περιηγητή και έχει σχεδιαστεί για να χρησιμοποιεί την έννοια της διαδραστικότητας στην HTML σελίδα μας. Ο σκοπός της δημιουργία της είναι να μπορούν ο πελάτης με το χρήστη να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου. Πολλοί την μπερδεύουν με την java αλλά στην πραγματικότητα είναι εντελώς δύο διαφορετικές γλώσσες και το αρχείο που δημιουργείται είναι σε μορφή .js .

Παράδειγμα σε JavaScript:

```
<script>
function displayDate() {
    document.getElementById ("demo").innerHTML = Date ();
}
</script>
```

Για να δηλώσουμε ότι αρχίζουμε να γράφουμε JavaScript χρησιμοποιούμε το tag <script> και μέσα εδώ γράφουμε εντολές που στο συγκεκριμένο παράδειγμα πατώντας ένα κουμπί το οποίο έχουμε δημιουργήσει με την HTML το καλούμε με μια συνάρτηση και θέλουμε να μας εμφανίζει την ώρα, μόλις τελειώσουμε κλείνουμε με </script>.

3.6.6 Server

Ο Server ή εξυπηρετητής στην ουσία είναι ένας υπολογιστής ο οποίος δεν κλείνει ποτέ, είναι υπεύθυνος για τον έλεγχο ενός δικτύου και αν κάτι πάει στραβά δεν θα μπορεί ο απλός χρήστης να έχει πρόσβαση στην εκάστοτε ιστοσελίδα. Έτσι για λόγους ασφαλείας η πρόσβαση επιτρέπεται μόνο σε εξειδικευμένα άτομα-χρήστες. Μια κατηγορία server είναι αυτοί που σου προσφέρουν τις ιστοσελίδες (web pages) και ονομάζονται web servers. Οι πιο γνωστοί είναι ο IIS και ο Apache τον οποίο χρησιμοποιήσαμε και για την ιστοσελίδα.

3.6.7 PHP

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία δυναμικών σελίδων στο διαδίκτυο. Η PHP έχει άμεση επαφή με τον διακομιστή (π.χ. Apache) ο οποίος την επεξεργάζεται και θα εμφανίσει τα αποτελέσματα σε πραγματικό χρόνο. Ένα αρχείο

με κώδικα PHP έχει κατάληξη σε .php , αλλά για να τρέξει το πρόγραμμα πρέπει και ο server να είναι ρυθμισμένος να το μεταγλωττίσει σε HTML. Πλέον τα περισσότερα λειτουργικά συστήματα υποστηρίζουν την εκτέλεση του κώδικα PHP.

Παράδειγμα κώδικα σε PHP:

```
<html>
<body>
<?php echo "Hello world"?>
</body>
</html>
```

Ο παραπάνω κώδικας θα μας εμφανίσει το Hello world.

3.6.8 jQuery

Η jQuery είναι η πιο δημοφιλής βιβλιοθήκη της JavaScript που έχει χρησιμοποιηθεί όσο καμία άλλη και έχει σχεδιαστεί για να απλοποιήσει το client-side scripting του HTML και να υλοποιεί οποιαδήποτε διαδικτυακή εφαρμογή, είτε αυτή είναι απλή είτε όχι. Η πρώτη έκδοση του jQuery ήταν 26 Αυγούστου 2006 και η τελευταία έκδοση 2.1 στις 24 Ιανουαρίου 2014, υπό την άδεια του MIT μέχρι και σήμερα. Η jQuery απλοποιεί τον προγραμματισμό καθώς με την βοήθεια της μπορούν υλοποιηθούν εργασίες πολύ πιο απλά και εύκολα σε σχέση με τη συμβατική JavaScript που θα ήταν ιδιαίτερα πολύπλοκες. Στα πλεονεκτήματα της συγκαταλέγεται ο μεγάλος αριθμός δυνατοτήτων και λειτουργιών που διαθέτει, καθώς και τη μεγάλη open-source κοινότητα που την υποστηρίζει[14].

3.6.9 jQuery Mobile

Το jQuery mobile είναι ένα ανοιχτό πλαίσιο για την δημιουργία κινητών εφαρμογών web και πιο συγκεκριμένα μια βιβλιοθήκη της JavaScript, που αναπτύσσεται από την ομάδα του jQuery και έτσι το καθιστά πολύ πιο εύκολο να το μάθει κανείς αν το γνωρίζει ήδη. Η πρώτη του έκδοση ξεκίνησε στις 16 Οκτωβρίου του 2010 και με τελευταία σταθερή έκδοση σε 1.4.5 στις 31 Οκτωβρίου του 2014. Μερικά από τα κύρια χαρακτηριστικά του είναι ότι χρησιμοποιεί HTML5 και CSS3 για την μορφοποίηση των σελίδων με ελάχιστο scripting, είναι συμβατό με όλους τους σημαντικούς φυλλομετρητές και λειτουργεί σε όλα τα δημοφιλή smartphones, tablets και desktop. Επίσης το jQuery mobile είναι συμβατό και με πολλά πλαίσια εφαρμογής για κινητά και πλατφόρμες όπως είναι το phonegap και περισσότερα[13].

3.6.10 Προσθήκη του jQuery mobile σε ιστοσελίδα

Υπάρχουν οι εξής δύο τρόποι για να προσθέσουμε το jQuery mobile σε ένα οποιοδήποτε site:

- Το link της βιβλιοθήκης του jQuery mobile που αποθηκεύετε σε ένα CDN.
- Το link της βιβλιοθήκης του jQuery mobile που αποθηκεύετε στον υπολογιστή μας.

Το CDN (Content Delivery Network) χρησιμοποιείται για τη διανομή συχνών χρησιμοποιούμενων αρχείων, γεγονός που καθιστά την ταχύτητα λήψης πολύ πιο γρήγορη για το χρήστη.

Στις παρακάτω εικόνες που ακολουθούν παρατίθενται οι κώδικες σχετικά με τους αντίστοιχους τρόπους προσθήκης του jQuery mobile σε ένα HTML αρχείο περιλαμβάνοντας το ακόλουθο στυλ (.css), javascript βιβλιοθήκες (.js) για να τα πάρει το jQuery mobile και να πραγματοποιήσει τις κατάλληλες ενέργειες μέσα στο <head>, όπου το tag <meta> καθορίζει το πώς το πρόγραμμα περιήγησης θα πρέπει να ελέγχει το επίπεδο ζουμ της σελίδας και τις ανάλογες διαστάσεις. Επίσης καθορίζει το πλάτος της σελίδας και της εκάστοτε συσκευής, το οποίο ποικίλει. Ακόμα, η αρχική κλίμακα ισούται με 1 και θέτει το αρχικό επίπεδο ζουμ, όταν η σελίδα φορτώνεται πρώτα από το πρόγραμμα περιήγησης. Τέλος, το tag <script> δεν εμπεριέχει το type= "text / javascript" και αυτό γιατί η JavaScript είναι η προεπιλεγμένη γλώσσα scripting σε HTML5 και σε όλα τα σύγχρονα προγράμματα περιήγησης.

jQuery Mobile CDN:

```
<head>
<!-- Include meta tag to ensure proper rendering and touch zooming -->
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- Include jQuery Mobile stylesheets -->
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
<!-- Include the jQuery library -->
<script src="http://code.jquery.com/jquery-1.11.2.min.js"></script>
<!-- Include the jQuery Mobile library -->
<script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
</head>
```

[Δοκιμάστε το μόνοι σας »](#)

Εικόνα 24: Προσθήκη του jQuery mobile σε ένα CDN.

```
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="jquery.mobile-1.4.5.css">
<script src="jquery.js"></script>
<script src="jquery.mobile-1.4.5.js"></script>
</head>
```

Εικόνα 25: Προσθήκη του jQuery mobile στον υπολογιστή.



Εικόνα 26: Λογότυπα βασικών γλωσσών προγραμματισμού.

4. Υπηρεσία γνωστοποιήσεων (notification service)

4.1 Ωθήσεις ειδοποιήσεων (Push Notifications)

Τα push notifications (ώθηση ειδοποιήσεων) είναι ένα άμεσο κανάλι για τους χρήστες των εκάστοτε εφαρμογών, τους οποίους τους κρατάει ενήμερους κάνοντας τις κατάλληλες ενημερώσεις και αναβαθμίσεις στην εφαρμογή. Ένα σημαντικό πλεονέκτημα των push notifications είναι ότι δεν απαιτείται από μια φορητή συσκευή να είναι ανοικτή (background), ώστε να μπορεί ο χρήστης να λάβει την ειδοποίηση. Αυτό επιτρέπει σε ένα smartphone(κινητή συσκευή) να λαμβάνει και να εμφανίζει ειδοποιήσεις ακόμα και όταν η οθόνη της συσκευής είναι κλειδωμένη. Ειδοποιήσεις στέλνονται ανά πάσα στιγμή ακόμα και όταν τρέχουμε την συγκεκριμένη εφαρμογή (foreground), είναι στο χέρι του προγραμματιστή τι θα επιλέξει να κάνει, υπάρχει ακόμα και η δυνατότητα να προγραμματίσει τα push notifications να έρχονται σε συγκεκριμένη ώρα και μέρα ανάλογα με τις ανάγκες και τις απαιτήσεις του πελάτη.

4.1.1 Χρησιμοποίηση των ωθήσεων ειδοποιήσεων (push notifications)

Αν για παράδειγμα ένας χρήστης λάβει ένα tweet, σχετικό με κάποια εφαρμογή που έχει ήδη εγκατεστημένη στη κινητή του συσκευή, δεδομένου ότι η εφαρμογή δεν εκτελείται αυτή τη στιγμή, δεν μπορεί να ελέγξει για τα γεγονότα αυτά. Έτσι όταν συμβαίνει αυτό το γεγονός που μας ενδιαφέρει, η πλευρά του εξυπηρετητή (server-side) μπορεί να στείλει στη συγκεκριμένη εφαρμογή push notifications. Υπάρχουν τρία πράγματα που μπορούμε να κάνουμε με push notifications, τα οποία αν τα συνδυάσουμε θα έχουμε ένα καλύτερο αποτέλεσμα:

- Προβολή ενός σύντομου μηνύματος κειμένου
- Αναπαραγωγή ενός σύντομου ήχου
- Ορισμός ενός αριθμού στο εικονίδιο της εφαρμογής

4.2 Ενεργοποίηση των ωθήσεων ειδοποιήσεων (push notifications) σε συσκευή Android

Πρώτα απ' όλα πριν ξεκινήσουμε, πρέπει να φροντίσουμε να ρυθμίσουμε τις υπηρεσίες του Google Play SDK, για να δοκιμάσουμε την εφαρμογή μας θα πρέπει να χρησιμοποιήσουμε είτε:

- Μια συμβατή συσκευή Android με τη νεότερη έκδοση που περιλαμβάνει το Google Play Store.

- Το Android emulator με AVD (Android Virtual Device), που τρέχει την πλατφόρμα του Google APIs που βασίζονται στη νεότερη έκδοση του Android.

Το AVD είναι μια διαμόρφωση εξομοιωτή που μας επιτρέπει να μοντελοποιήσουμε μια πραγματική συσκευή με τον καθορισμό του υλικού και του λογισμικού.

Αυτό το SDK το χρειαζόμαστε για να χρησιμοποιήσουμε τα API του GCM (Google Cloud Messaging), το οποίο είναι μια δωρεάν υπηρεσία που επιτρέπει στους προγραμματιστές να στέλνουν μηνύματα στην εφαρμογή. Αυτό για παράδειγμα θα μπορούσε να είναι ένα απλό μήνυμα (νέο e-mail) λέγοντας στην εφαρμογή ότι υπάρχουν νέα δεδομένα που πρέπει να τα πάρει από τον διακομιστή. Η υπηρεσία GCM χειρίζεται όλες τις πτυχές αναμονής των μηνυμάτων και την παροχή προς και από την εφαρμογή.

4.2.1 Δημιουργία έργου API Google

Για να δημιουργήσουμε ένα έργο API της Google πρέπει να ακολουθήσουμε τα παρακάτω βήματα:

1. Ανοίγουμε το Google Developers Console.
2. Αν δεν έχουμε δημιουργήσει ένα έργο API ακόμα, κάνουμε κλικ στην επιλογή **Δημιουργία έργου**.
3. Δίνουμε ένα όνομα έργου και κάνουμε κλικ στο κουμπί **Δημιουργία**. Στη συνέχεια μόλις δημιουργηθεί το έργο, εμφανίζεται μια σελίδα που εμφανίζει το ID του έργου μας και τον αριθμό του σχεδίου. Για παράδειγμα, **Αριθμός Έργου: 682250056998**.
4. Τέλος, αντιγράφουμε το αριθμό του έργου γιατί θα το χρησιμοποιήσουμε αργότερα ως αναγνωριστικό του αποστολέα GCM.

4.2.2 Ενεργοποίηση υπηρεσίας GCM και η απόκτηση κλειδιού API

Ανεξάρτητα από την πλατφόρμα στην οποία θα αναπτυχθεί μια εφαρμογή, το πρώτο βήμα που πρέπει να κάνουμε είναι να εγγραφούμε με GCM. Για να ενεργοποιήσουμε την συγκεκριμένη υπηρεσία πρέπει:

- ✓ Στην πλαϊνή αριστερή μπάρα να επιλέξουμε **APIs & auth**.
- ✓ Στην εμφανιζόμενη λίστα των APIs, γυρίζουμε το **Google Cloud Messaging για το Android** εναλλαγή στο On.

Για να αποκτήσουμε ένα κλειδί API:

1. Στην αριστερή πλαϊνή μπάρα επιλέγουμε **API & auth** → **credentials**.
2. Σύμφωνα με την **δημόσια πρόσβαση του API** κάνουμε κλικ στην επιλογή **Δημιουργία νέου κλειδιού**.

3. Στο παράθυρο **Δημιουργήστε ένα νέο κλειδί**, κάνουμε κλικ στο **πλήκτρο διακομιστή**.
4. Το παράθυρο διαλόγου των ρυθμίσεων με αποτέλεσμα, μας παρέχεται η διεύθυνση IP του διακομιστή μας, για σκοπούς ελέγχου μπορούμε να χρησιμοποιήσουμε 0.0.0.0/0.
5. Στη συνέχεια κάνουμε κλικ στην επιλογή **Δημιουργία**.
6. Στην ανανεωμένη σελίδα αντιγράφουμε το κλειδί API, διότι θα το χρειαστούμε αργότερα για να εκτελέσει τον έλεγχο ταυτότητας στο διακομιστή της εφαρμογής.

Εάν θέλουμε να δημιουργήσουμε ένα νέο κλειδί επειδή νομίζουμε ότι το υπάρχον κλειδί έχει παραβιαστεί και επιθυμούμε να προχωρήσουμε σε άμεση διαγραφή, τότε κάνουμε κλικ στην επιλογή Διαγραφή. Τέλος, μόλις τελειώσουμε με τα παραπάνω βήματα που αναφέραμε, είμαστε έτοιμοι να ξεκινήσουμε την εφαρμογή της GCM.

Παρακάτω ακολουθεί ο κώδικας (για την εφαρμογή του Capital.gr), μέσω του οποίου δέχεται push notifications μια συσκευή Android και λαμβάνουμε το ID που είχαμε δημιουργήσει παραπάνω μέσω του Google Developer:

```
if (platform == 'android') {
    if (localStorage.getItem('notificationRegId') === null) {
        var androidConfig = {
            "senderID": "682250056998"
        };
        //if (development)
        $cordovaPush.unregister(androidConfig).then(function(result) {
            // // Success!
            //localStorage.setItem('notificationRegId',null);
        }, function(err) {
            // //development = false;
            // // Error
        });
    }
}
```

Η παρακάτω μέθοδος register επιτρέπει στην εφαρμογή μας να λαμβάνει push notifications και το αντικείμενο Config ρυθμίζει τους παραμέτρους για τον καθορισμό των προκαθορισμένων τιμών.

```
$cordovaPush.register(androidConfig).then(function (result) {
    //alert(result);
}, function (err) {
    alert (error);
});
```

```

    });
}
$scope.$on('$cordovaPush:notificationReceived', function (event,
notification) {
    //alert(notification);
    switch (notification.event) {
        case 'registered':
            if (notification.regid.length > 0) {
                var url =
                "http://capital.ibhellas.gr/notifications/register.php?name=bob&email=someone@eve
                rybody.com&regId="+notification.regid;
                localStorage.setItem('notificationRegId',
                JSON.stringify(notification.regid));
                $.ajax({ type: "GET",
                url: url,
                async: false,
                success : function(text)
                {
                    //alert(text);
                }
                });
            }
            break;

        case 'message':
            if (notification.message !== "") {
                if (notification.foreground) {
                    try{
                        window.plugins.toast.showShortTop(notification.payload.stockName, null, function
                        (b) {
                            alert('toast error: ' + b)
                        })
                    }
                }
            }
        }
    }
}

```

```

        });}
    catch(e){alert(e)}
    // //navigator.notification.beep();
    navigator.vibrate(1000);

    navigator.notification.alert(notification.payload.message,
function () {
    try {
        $state.go('eventmenu.notification', { details:
notification.payload.articleId});

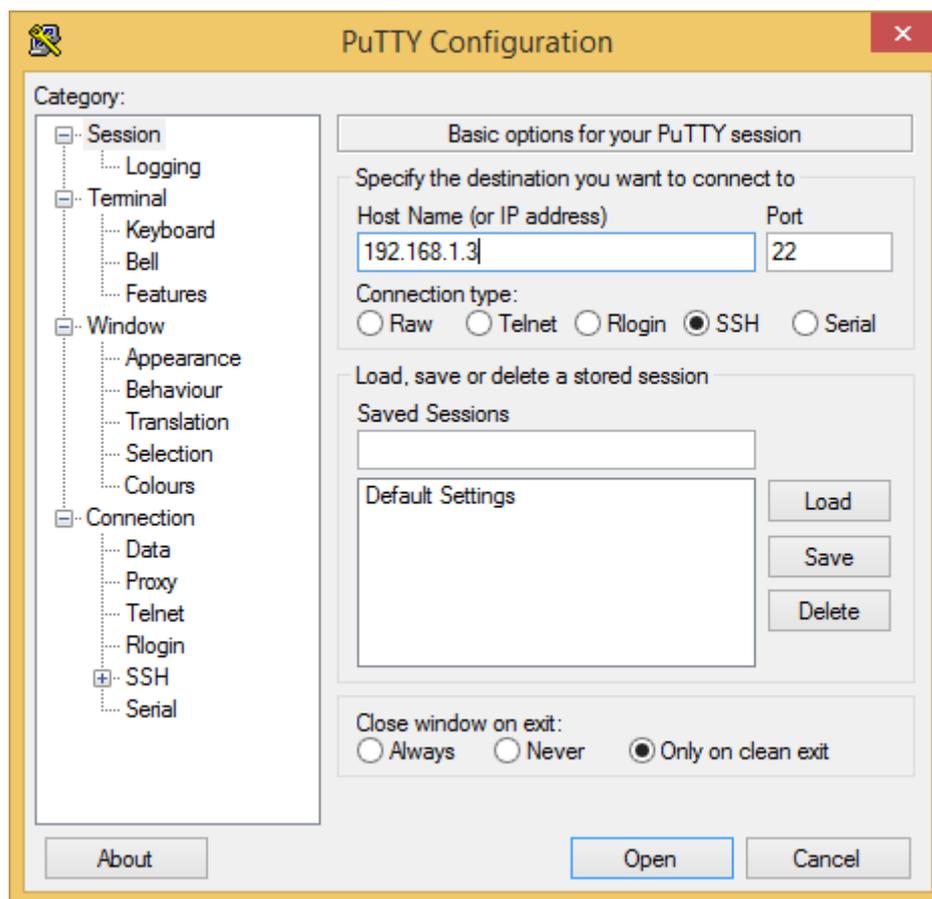
        } catch(e){alert(e)}
        }, notification.payload.stockName, "Περισσότερα..");
    } else {
        $state.go('eventmenu.notification', { details:
notification.payload.articleId});
    }
    break;
case 'error':
    alert('GCM error = ' + notification.msg);
    break;

default:
    alert('An unknown GCM event has occurred');
    break;
}
});
}

```

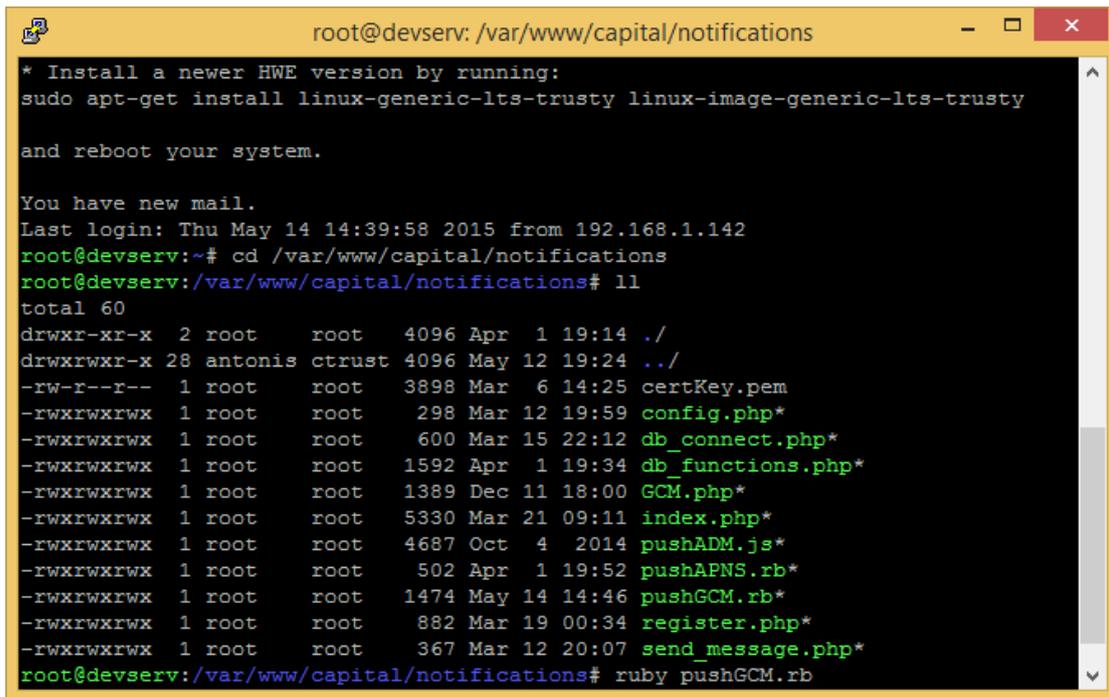
4.2.3 Αποστολή push notifications της εφαρμογής Capital.gr σε συσκευή Android

Αρχικά, βρίσκουμε ένα προγραμματάκι όπως το putty που χρησιμοποιούμε στο συγκεκριμένο παράδειγμα και όπως φαίνεται και στην παρακάτω εικόνα μας ζητάει να πληκτρολογήσουμε την IP του δικτύου μας και την ανάλογη πόρτα και με επιλεγμένη την επιλογή **SSH**, πατάμε **Open**.



Εικόνα 27: Αρχική σελίδα του προγράμματος Putty.

Μας εμφανίζει μια κονσόλα στην οποία βάζουμε τα στοιχεία του server μας, ένα όνομα και τον αντίστοιχο κωδικό. Στη συνέχεια με **cd** πάμε στον προορισμό όπου βρίσκεται στον server μας, ένας φάκελος που τον έχουμε ονομάσει notifications και ψάχνουμε μέσα σ' αυτόν χρησιμοποιώντας το **ll** για να μας εμφανίσει όλα τα αρχεία που περιέχει ο φάκελος. Τέλος, πληκτρολογούμε **ruby pushGCM.rb**, όπου είναι και το όνομα του αρχείου που ψάχνουμε στο οποίο βρίσκεται ο κώδικας που έχουμε προγραμματίσει για να στέλνει τα notifications σε γλώσσα Ruby και πατώντας Enter στέλνει το notification στην συσκευή μας. Όπως βλέπουμε χρησιμοποιούμε το κλειδί που πήραμε από το GCM για τα push notifications, στην συνέχεια είναι το registration ID και τέλος του προσδιορίζουμε το όνομα του μηνύματος, το περιεχόμενο, και τον ήχο ειδοποίησης που θα λαμβάνουμε κάθε φορά στην συσκευή μας.



```
root@devserv: /var/www/capital/notifications
* Install a newer HWE version by running:
sudo apt-get install linux-generic-lts-trusty linux-image-generic-lts-trusty
and reboot your system.

You have new mail.
Last login: Thu May 14 14:39:58 2015 from 192.168.1.142
root@devserv:~# cd /var/www/capital/notifications
root@devserv:/var/www/capital/notifications# ll
total 60
drwxr-xr-x  2 root   root   4096 Apr  1 19:14 ./
drwxrwxr-x 28 antonis ctrust 4096 May 12 19:24 ../
-rw-r--r--  1 root   root   3898 Mar  6 14:25 certKey.pem
-rwxrwxrwx  1 root   root    298 Mar 12 19:59 config.php*
-rwxrwxrwx  1 root   root    600 Mar 15 22:12 db_connect.php*
-rwxrwxrwx  1 root   root   1592 Apr  1 19:34 db_functions.php*
-rwxrwxrwx  1 root   root   1389 Dec 11 18:00 GCM.php*
-rwxrwxrwx  1 root   root   5330 Mar 21 09:11 index.php*
-rwxrwxrwx  1 root   root   4687 Oct  4 2014 pushADM.js*
-rwxrwxrwx  1 root   root    502 Apr  1 19:52 pushAPNS.rb*
-rwxrwxrwx  1 root   root   1474 May 14 14:46 pushGCM.rb*
-rwxrwxrwx  1 root   root    882 Mar 19 00:34 register.php*
-rwxrwxrwx  1 root   root    367 Mar 12 20:07 send_message.php*
root@devserv:/var/www/capital/notifications# ruby pushGCM.rb
```

Εικόνα 28: Εκτέλεση εντολών στην κονσόλα, για αποστολή push notification σε συσκευή Android.

```
require 'rubygems'
require 'pushmeup'

GCM.host = 'https://android.googleapis.com/gcm/send'

GCM.format =: json

GCM.key = "AIzaSyBI08KqYJL9IfXkxcYayhcSox41NHCx3d8"

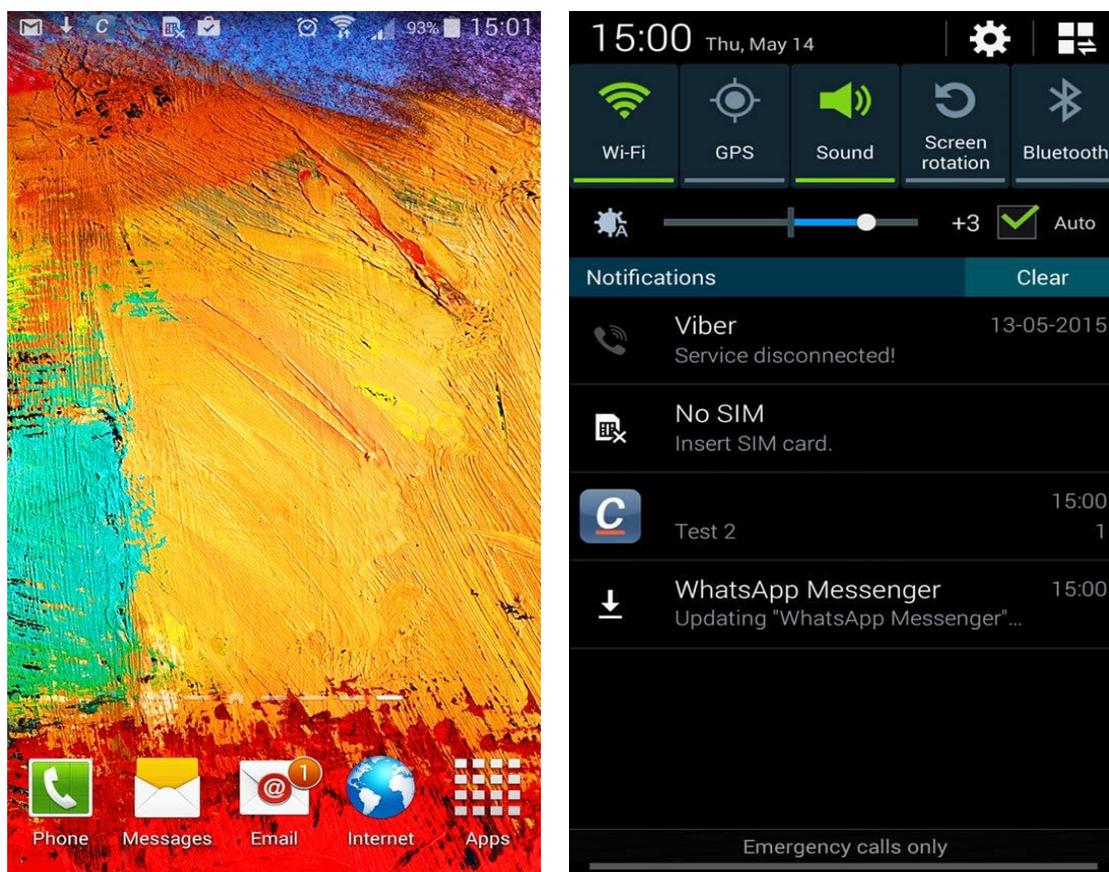
destination =
["APA91bEDYAuuqXWqy58RZtllTIoWC1alwZst1aeIdU68Ia6q4K4tmtXixApklyQD
yWiuQhIZzT8oOpsBkF157y4AhBKOZUF6M92_bZ0ClAsCLC2vEZJUp9OegLZks0
m7UkZizLkL57_s1tzzfjcSnOsJW59hwXvsdPQ"];

data = {:articleId => "2136701",:stockName => "BYTE", :message => "Test 2",
:msgcnt => "1", :soundname => "beep.wav"}

GCM.send_notification (destination, data)
```

Μόλις τελειώσουμε με όλα τα βήματα, μπορούμε να δούμε το αποτέλεσμα στις ακόλουθες εικόνες. Όπου στην πρώτη μας εμφανίζει, πως βλέπουμε την συγκεκριμένη ειδοποίηση όταν βρισκόμαστε στην αρχική σελίδα της κινητή μας συσκευής, όπου πάνω δεξιά φαίνεται το εικονίδιο της, πράγμα που σημαίνει ότι μας έχει έρθει κάποιο μήνυμα και πρέπει να το διαβάσουμε. Η δεύτερη εικόνα μας εμφανίζει το εικονίδιο της εφαρμογής μας, με το όνομα που της έχουμε προσδιορίσει

για το συγκεκριμένο notification, την ώρα που στάλθηκε και πατώντας το συγκεκριμένο πλαίσιο μας μεταφέρει στην εφαρμογή.

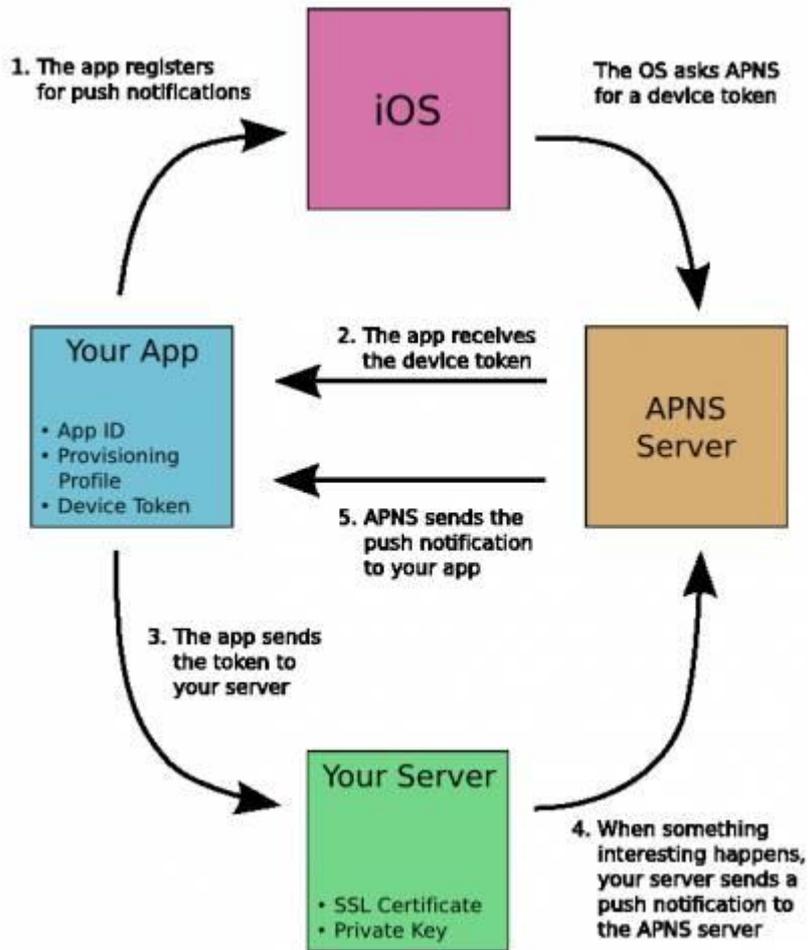


Εικόνα 29: Αποτελέσματα αποστολής του notification σε συσκευή Android.

4.3 Ενεργοποίηση των push notifications σε συσκευή iOS

Η Apple αρχικά εισήγαγε τα push notifications, για να επιτρέπουν στις εφαρμογές να ανταποκρίνονται στα γεγονότα, αν η εφαρμογή δεν εκτελείται στο προσκήνιο (foreground). Ωστόσο, το λειτουργικό σύστημα και οι συσκευές iOS έχουν αλλάξει σημαντικά τα τελευταία χρόνια και οι εφαρμογές δεν χρειάζεται να βασίζονται αποκλειστικά σε push notifications, για την εκτέλεση των καθηκόντων στο προσκήνιο, αυτό βέβαια δεν σημαίνει ότι δεν μας είναι πλέον χρήσιμα.

Στην παραπάνω εικόνα απεικονίζεται η διαδικασία με την οποία ενεργοποιούμε τα push notifications σε μια συσκευή iOS, πάμε να την αναλύσουμε βήμα – βήμα.



Εικόνα 30: Διαδικασία ενεργοποίησης της λειτουργίας των push notifications.

Μια εφαρμογή δίνει τη δυνατότητα στα push notifications, όπου ο χρήστης πρέπει να επιβεβαιώσει ότι επιθυμεί να λαμβάνει αυτές τις ειδοποιήσεις. Η εφαρμογή λαμβάνει μια διεύθυνση που τα push notifications θα τη στείλουν με τη σειρά τους στον διακομιστή μας. Όταν συμβαίνει κάτι ενδιαφέρον για την εφαρμογή μας, τότε ο διακομιστής στέλνει μια ειδοποίηση ώθηση για το Apple Push Notification Service, ή APNs για συντομία. Στη συνέχεια το APNs στέλνει την κοινοποίηση ώθηση στη συσκευή του χρήστη. Όταν η συσκευή του χρήστη λάβει την κοινοποίηση ώθηση, θα έχει τη δυνατότητα να δείχνει μια ειδοποίηση, να αναπαράγει έναν ήχο και να ενημερώνει το εικονίδιο της εφαρμογής.

Το APNs είναι μια υπηρεσία που δημιουργήθηκε από την Apple, που ξεκίνησε μαζί με το iOS 3.0 τον Ιούνιο του 2009. Το APNs επιτρέπει κοινοποιήσεις και αποστολή μηνυμάτων σε τρίτους, για οποιαδήποτε συσκευή της Apple, όπως το iPhone, iPad κ.τ.λ.

4.3.1 Παράμετροι για push notifications

Για να προσθέσουμε τα push notifications σε μια εφαρμογή με λειτουργικό σύστημα iOS χρειαζόμαστε:

- ✓ **Ένα iPhone ή ένα iPad:** push notifications που δεν λειτουργούν σε προσομοιωτή, θα πρέπει να τα δοκιμάσουμε στη συσκευή μας.
- ✓ **Ένα iOS Developer Program μελών:** θα πρέπει να φτιάξουμε ένα νέο App ID και ένα προφίλ για κάθε εφαρμογή που χρησιμοποιεί push, καθώς και ένα πιστοποιητικό SSL για το διακομιστή, αυτό μπορείτε να το κάνετε στο iOS Provisioning Portal.
- ✓ **Έναν server που είναι συνδεδεμένος στο διαδίκτυο:** push notifications στέλνονται πάντα από ένα διακομιστή. Μόνο για την ανάπτυξη μπορούμε να χρησιμοποιήσουμε το Mac με το διακομιστή, αλλά παραγωγική χρήση θα πρέπει να έχετε τουλάχιστον κάτι σαν ένα VPS (Virtual Private Server). Ένας κοινόχρηστος λογαριασμός φιλοξενίας δεν είναι αρκετά καλή λύση. Θα πρέπει να είμαστε σε θέση να εκτελέσουμε μια διαδικασία στο διακομιστή, να εγκαταστήσουμε ένα πιστοποιητικό SSL και να έχουμε τη δυνατότητα να κάνουμε εξερχόμενες συνδέσεις TLS σε ορισμένα ports. Βέβαια οι περισσότεροι shared hosting πάροχοι δεν μας επιτρέπουν να κάνουμε κάτι τέτοιο αν και θα μπορούσαν.

Παρακάτω ακολουθεί ο κώδικας (για την εφαρμογή του Capital.gr), μέσω του οποίου δέχεται push notifications μια συσκευή iOS και λαμβάνουμε το ID:

```
if (platform == 'ios' && onDevice) {  
    if (localStorage.getItem('notificationRegId') === null) {  
        var iosConfig = {  
            "badge": true,  
            "sound": true,  
            "alert": true  
        };  
        $cordovaPush.register(iosConfig).then(function(result) {  
            // Success -- send deviceToken to server, and store for future use  
            //$http.post("http://server.co/", {user: "Bob", tokenID:  
result.deviceToken})  
            // TODO: prepare server for posting data
```

```

        Var url =
"http://capital.ibhellas.gr/notifications/register.php?name=blahgfd&email=someone@
everybody.com&regId="+result;

        localStorage.setItem('notificationRegId', JSON.stringify(result));
$.ajax({ type: "GET",
        url: url,
        async: false,
        success: function(text)
        {
            //alert(text);

        }
});
//http://capital.ibhellas.gr/notifications/register.php?
//localStorage.setItem('notificationRegId', result.deviceToken);
//alert(JSON.stringify(result));
}, function(err) {
    alert("Registration error: " + err)
});
}
$scope.$on('$cordovaPush:notificationReceived', function(event,
notification) {
    if (notification.alert) {
        if (notification.foreground){
            navigator.notification.alert(notification.alert,function(){

$scope.go('eventmenu.notification',{details:notification.articleId});
            },notification.stockName,'Περισσότερα...');
        } else {
            $scope.go('eventmenu.notification',{details:notification.articleId});
        }
    }
}
}

```

```

if (notification.sound) {
    $cordovaDialogs.beep(2);
}

if (notification.badge) {

```

Στην παρακάτω μέθοδος ο ακέραιος αριθμός ρυθμίζει την καταμέτρηση του ορατού σήματος όταν η εφαρμογή δεν λειτουργεί.

```

$cordovaPush.setBadgeNumber(notification.badge).then(function(result) {
    //alert(result);
    // Success!
}, function(err) {
    // An error occurred. Show a message to the user
});
}
});
}
// TODO Windows phone else case
}

```

4.3.2 Provisioning profile (Διαθέσιμο περίγραμμα) και πιστοποιητικά SSL

Για να ενεργοποιήσουμε τα push notifications στην εφαρμογή μας, θα πρέπει να υπογραφεί ένα προφίλ προβλέψεων που έχει ρυθμιστεί για τη λειτουργία push. Επιπλέον, ο διακομιστής μας θα πρέπει να υπογράψει τις ανακοινώσεις της στην APNs με ένα πιστοποιητικό SSL.

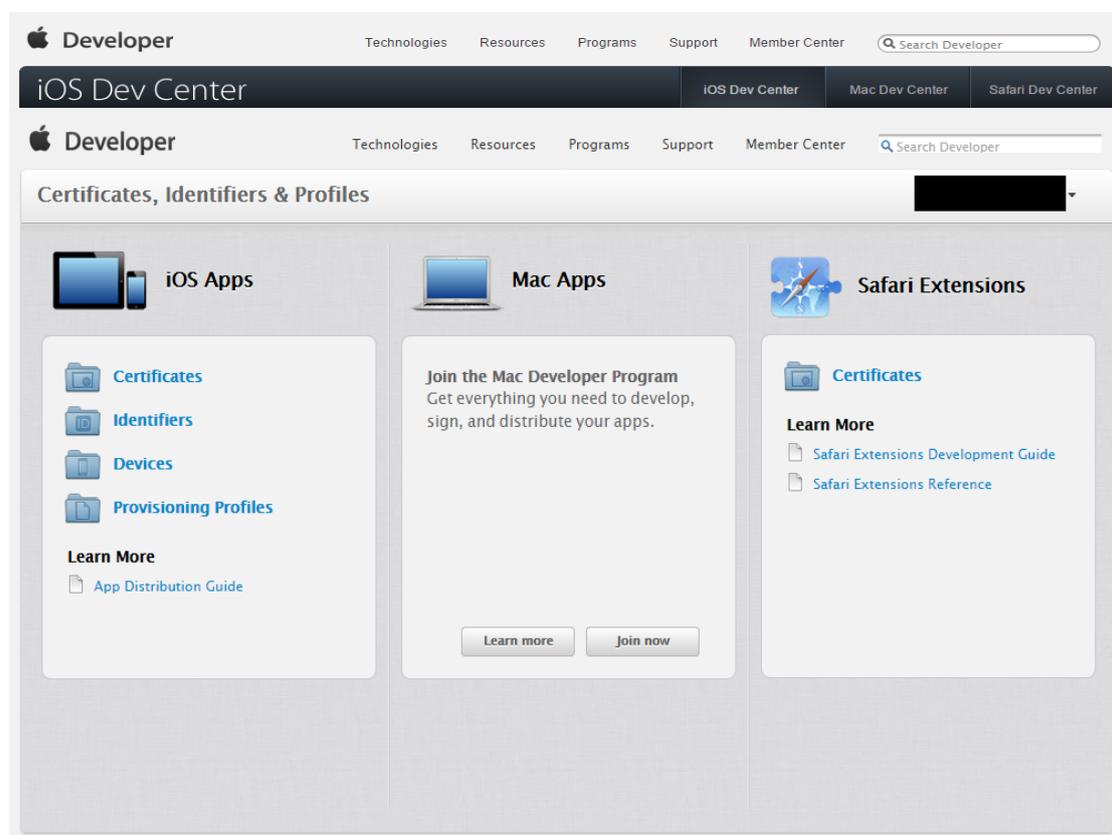
Το προφίλ προβλέψεων και το πιστοποιητικό SSL είναι στενά δεμένα μεταξύ τους και ισχύουν μόνο για ένα αναγνωριστικό app. Αυτή είναι μια προστασία που εξασφαλίζει μόνο ο διακομιστής μας, ο οποίος μπορεί να στείλει push notifications και κανένα άλλος. Όπως μας είναι γνωστό, οι εφαρμογές χρησιμοποιούν διαφορετικά προφίλ προβλέψεων για την ανάπτυξη και τη διανομή. Υπάρχουν επίσης δύο τύποι πιστοποιήσεων push διακομιστή:

- **Ανάπτυξη:** Εάν η εφαρμογή μας βρίσκεται σε λειτουργία εντοπισμού σφαλμάτων και έχει υπογραφεί με το αναπτυξιακό προφίλ προβλέψεων (Υπογραφή κώδικα ταυτότητας είναι το «iPhone Developer»), τότε ο server μας πρέπει να χρησιμοποιεί το πιστοποιητικό ανάπτυξης.

- **Παραγωγή:** Οι εφαρμογές που διανέμονται ως ad hoc (σχεδιασμένη λύση για ένα συγκεκριμένο πρόβλημα) ή από το App Store (όταν Υπογραφή κώδικα ταυτότητας είναι "iPhone Distribution"), πρέπει να επικοινωνήσουμε με έναν διακομιστή που χρησιμοποιεί το πιστοποιητικό παραγωγής. Σε περίπτωση που δεν υπάρχει αντιστοιχία μεταξύ αυτών, τα push notifications δεν μπορούν να σταλούν στην εφαρμογή μας.

4.3.3 Κατασκευή App ID (Ταυτότητα App) και SSL Certificate

Αρχικά, συνδεόμαστε στο [iOS Dev Center](#) και επιλέγουμε από τον δεξιό πίνακα: "**Certificates, Identifiers & Profiles**".

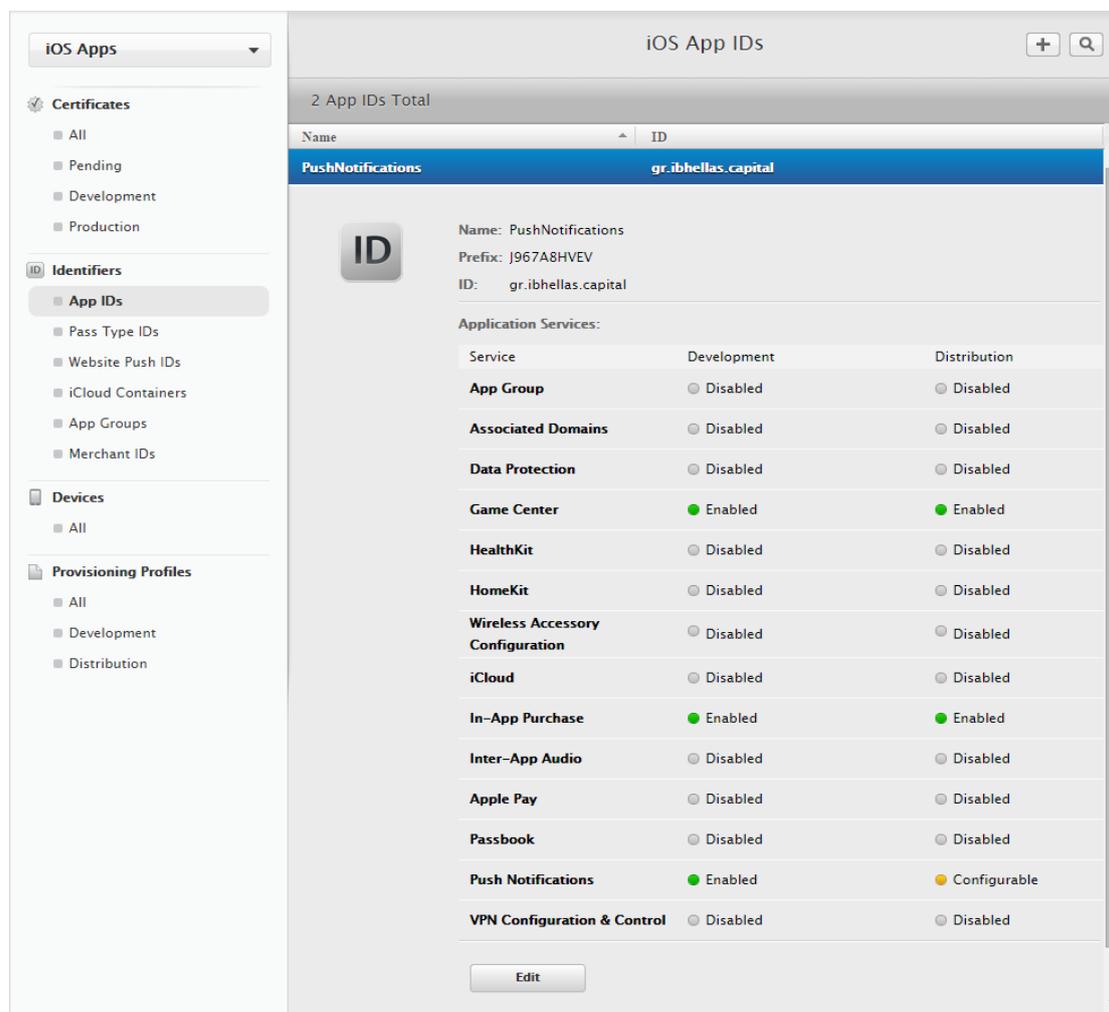


Εικόνα 31: Αρχική σελίδα του Apple Developer.

Στη συνέχεια θα μας εμφανίσει το παρακάτω παράθυρο και δεδομένου ότι κάνουμε μια iOS εφαρμογή, επιλέγουμε **Identifiers** από την ενότητα **iOS Apps**.

Τώρα θα δημιουργήσουμε μια νέα ταυτότητα App. Κάθε push app χρειάζεται το δικό του μοναδικό ID, διότι push notifications αποστέλλονται σε μια συγκεκριμένη εφαρμογή. Πηγαίνουμε στο **App IDs** στη sidebar και κάνουμε κλικ στο κουμπί + που βρίσκεται πάνω δεξιά και συμπληρώνουμε ένα όνομα της επιλογής μας στο **App ID Description**, στη συνέχεια κάνουμε κλικ την επιλογή **Explicit App ID** και στο παρακάτω πλαίσιο του Bundle ID πληκτρολογούμε το ID σε μορφή (com.domainname.appname), όπου θα το χρησιμοποιήσουμε ακριβώς το ίδιο και στο

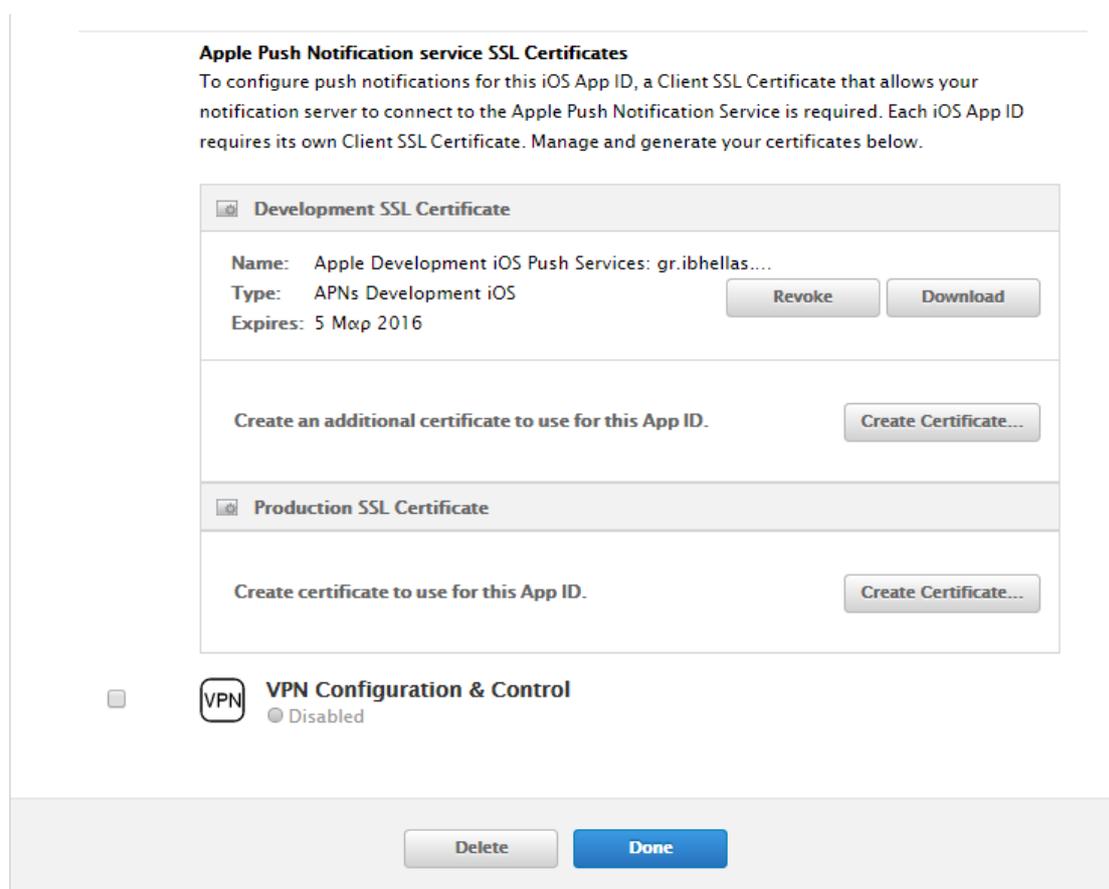
έργο Xcode μας. Τέλος, στην κατηγορία **App Services** κάνουμε κλικ στην επιλογή Push Notifications. Αφού τελειώσουμε με όλα τα βήματα πατάμε το κουμπί **Continue** και θα μας ζητηθεί να επιβεβαιώσουμε τα στοιχεία της ταυτότητας App, αν όλα είναι εντάξει κάνουμε κλικ στην **Υποβολή** και έτσι έχουμε εγγραφεί επιτυχώς. Στην εικόνα που ακολουθεί έχουμε ήδη δημιουργήσει το App ID για την εφαρμογή μας, με όνομα PushNotifications και ID: **gr.ibhellas.capital** και ακριβώς παρακάτω ακολουθούν τα κατάλληλα services που έχουμε ορίσει εμείς στην εφαρμογή μας.



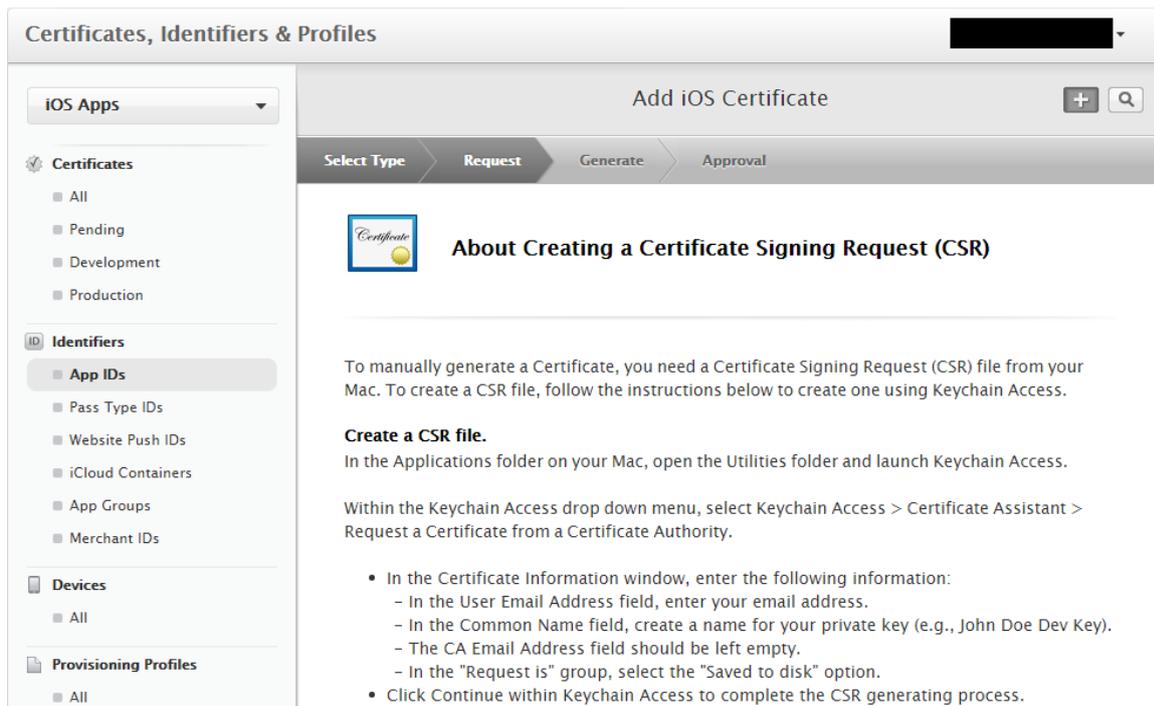
Εικόνα 32: Λίστα με τις υπηρεσίες της εφαρμογής.

Κάνοντας Edit, όπως φαίνεται στη παρακάτω εικόνα και μετακινηθούμε προς τα κάτω στην ενότητα Push Notifications επιλέγουμε **Δημιουργία Πιστοποιητικού** στην ενότητα **Ανάπτυξη Πιστοποιητικού SSL**. Το πρώτο πράγμα που θα μας ζητήσει είναι να δημιουργήσουμε ένα Αίτημα Υπογραφής Πιστοποιητικού, αυτό θα πρέπει να έχει γίνει από πριν μέσω ενός προγράμματος του **Keychain Access** και επιλέγουμε **Βοηθός πιστοποιητικών** και στη συνέχεια **Αίτημα πιστοποιητικού από αρχή έκδοσης πιστοποιητικών**. Έπειτα στο παράθυρο που θα μας ανοίξει, εισάγουμε ένα e-mail μας, μια ενδεικτική ονομασία π.χ. Capital, το οποίο θα μας βοηθήσει να βρούμε εύκολα το ιδιωτικό κλειδί αργότερα. Κάνουμε κλικ στην επιλογή **αποθήκευση στο δίσκο** και πατάμε **Συνέχεια**. Αν πάμε στην ενότητα κλειδιά του Keychain Access, θα δούμε ότι

έχει δημιουργηθεί ένα δημόσιο και ένα ιδιωτικό κλειδί, κάνουμε δεξί κλικ και Εξαγωγή στο δεύτερο και αποθηκεύουμε το ιδιωτικό κλειδί όπως Capital.p12 και πληκτρολογούμε μια φράση πρόσβασης. Αφού τελειώσουμε με αυτά τα σύντομα βήματα θα είμαστε σε θέση να ανεβάσουμε το Certificate Signing Request (CSR) που δημιουργήσαμε προηγουμένως. Σε λίγα δευτερόλεπτα θα έχει δημιουργηθεί το πιστοποιητικό SSL και έτσι θα είμαστε σε θέση ώστε να το κατεβάσουμε, πατώντας λήψη με όνομα «aps-development.cer». Πλέον έχουμε ένα έγκυρο πιστοποιητικό που είναι διαθέσιμο για την ανάπτυξη, το οποίο είναι έγκυρο μόνο για 3 μήνες, σε αντίθεση με το πιστοποιητικό παραγωγής που εξακολουθεί να ισχύει για ένα χρόνο. Αν θέλουμε να ανεβάσουμε την εφαρμογή μας στο store, επαναλαμβάνουμε αυτή τη διαδικασία για το πιστοποιητικό παραγωγής.



Εικόνα 33: Ανάπτυξη και παραγωγή του πιστοποιητικού.



Εικόνα 34: Διαδικασία δημιουργίας του CSR.

Έτσι τώρα έχουμε 3 αρχεία:

- το CSR
- το ιδιωτικό κλειδί ως αρχείο p12 (Capital.p12)
- το πιστοποιητικό SSL, aps_development.cer

Αφού αποθηκεύσουμε και τα τρία αρχεία, μπορούμε να κρατήσουμε και το CSR, σε περίπτωση που λήξει το πιστοποιητικό να μπορούμε να χρησιμοποιήσουμε το ίδιο με μόνη διαφορά το αρχείο .cer που θα αλλάξει.

Το push μέρος των servers μας θα είναι γραμμένο σε PHP, έτσι θα πρέπει να βρούμε ένα τρόπο ώστε να μετατρέψουμε το πιστοποιητικό και το ιδιωτικό κλειδί σε μια πιο εύχρηστη μορφή, γι' αυτό θα συνδυάσουμε το πιστοποιητικό και το ιδιωτικό κλειδί σε ένα ενιαίο αρχείο με τη μορφή PEM. Ανοίγουμε το τερματικό και πηγαίνουμε στο φάκελο όπου έχουμε κατεβάσει τα αρχεία, στην συγκεκριμένη περίπτωση στα έγγραφα:

```
$ cd ~ /Documents/
```

Μετατρέπουμε το αρχείο .cer σε .pem:

```
$openssl x509 -in aps_development.cer -inform der -out CapitalCert.pem
```

Μετατρέπουμε το αρχείο .p12 του ιδιωτικού κλειδιού σε ένα αρχείο .pem:

```
$openssl pkcs12 -nocerts -out CapitalKey.pem -in Capital.p12
```

Enter Import Password:

Mac verified OK

Enter PEM pass phrase:

Verifying – Enter PEM pass phrase:

Θα πρέπει πρώτα να εισάγουμε την συνθηματική φράση για το αρχείο .p12 έτσι ώστε το openssl να μπορεί να το διαβάσει. Στη συνέχεια θα πρέπει να εισέλθει σε μια νέα φράση πρόσβασης που θα χρησιμοποιηθεί για την κρυπτογράφηση του αρχείου PEM. Τέλος, συνδυάζουμε το πιστοποιητικό και το ιδιωτικό κλειδί σε ένα ενιαίο αρχείο .pem:

\$ cat CapitalCert.pem CapitalKey.pem > capital.pem

Σε αυτό το σημείο μπορούμε να κάνουμε ένα έλεγχο του πιστοποιητικού, εκτελώντας την ακόλουθη εντολή:

\$telnet gateway.sandbox.push.apple.com 2195

Trying 17.172.232.226...

Connected to gateway.sandbox.push-apple.com.akadns.net

Escape character is '^['.

Πραγματοποιείται σύνδεση με το διακομιστή APNs, χωρίς κρυπτογράφηση. Αν δούμε την παραπάνω απάντηση, τότε το Mac μας μπορεί να φτάσει APNs, πατάμε Ctrl + C για να κλείσει η σύνδεση.

Ας προσπαθήσουμε αυτή τη φορά να συνδεθούμε ξανά, χρησιμοποιώντας το πιστοποιητικό και το ιδιωτικό κλειδί για να δημιουργήσουμε μια ασφαλή σύνδεση:

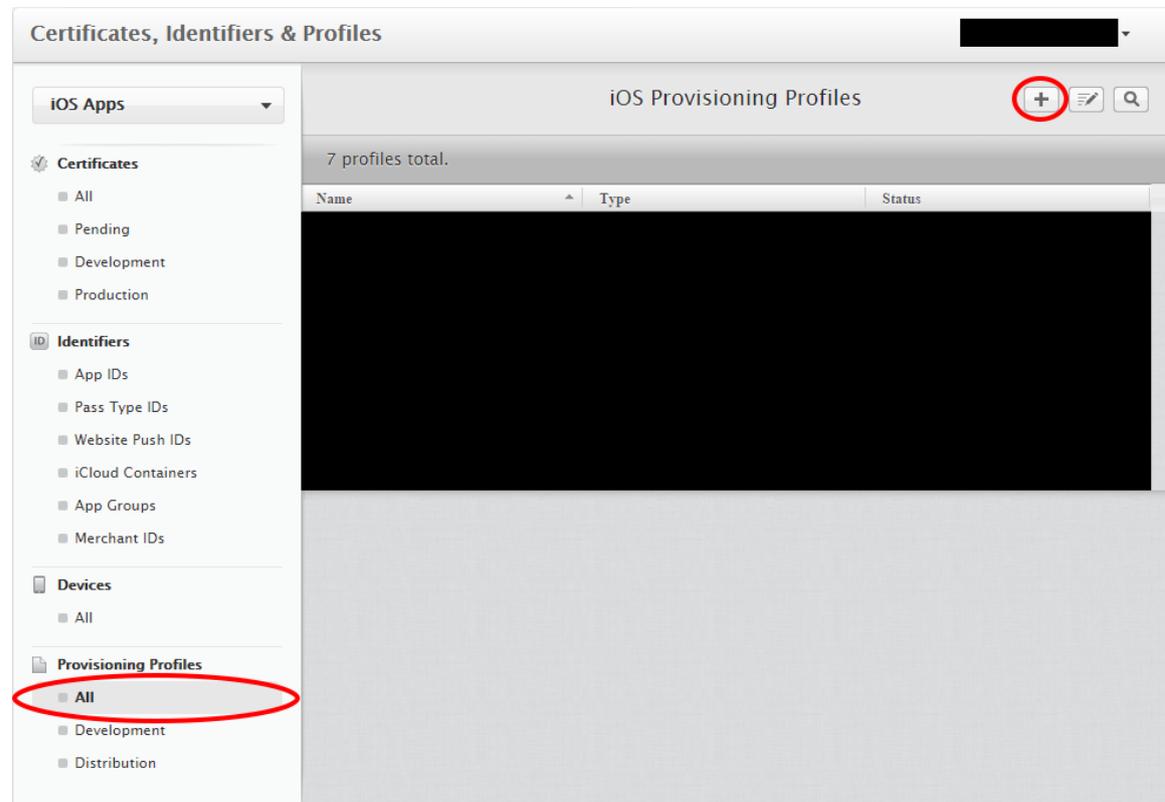
\$openssl s_client -connect gateway.sandbox.push.apple.com:2195 -cert CapitalCert.pem -key CapitalKey.pem

Enter pass phrase for CapitalKey.pem:

Εάν η σύνδεση είναι επιτυχής, θα πρέπει να μπορούμε να πληκτρολογούμε μερικούς χαρακτήρες και όταν πατήσουμε enter ο διακομιστής πρέπει να αποσυνδεθεί. Όσον αφορά την παραγωγική χρήση ακολουθούμε την ίδια διαδικασία, αλλά δεν χρησιμοποιούμε το «sandbox» server.

4.3.4 Δημιουργία Provisioning profile

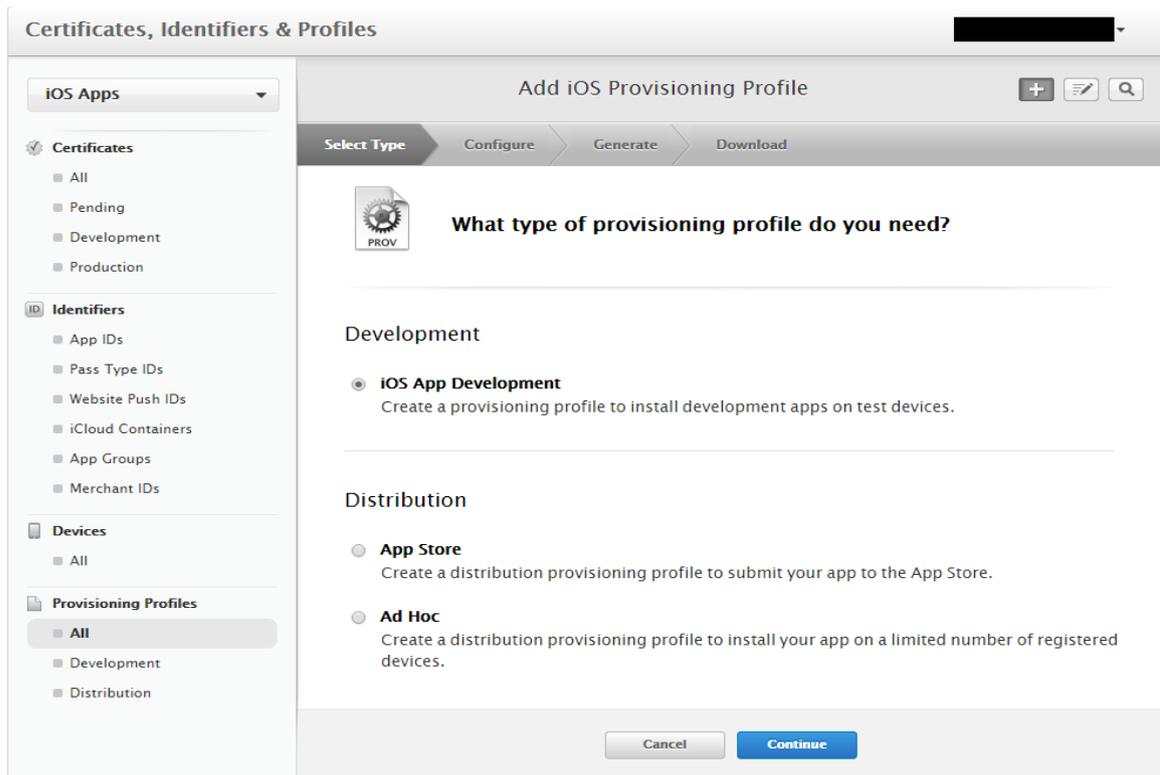
Πηγαίνουμε στην ενότητα Provisioning Profiles που υπάρχει στην πλαϊνή γραμμή και κάνουμε κλικ στο + κουμπί.



Εικόνα 35: Δημιουργία του provisioning profile.

Βήμα 1ο: Επιλέγουμε κατηγορία

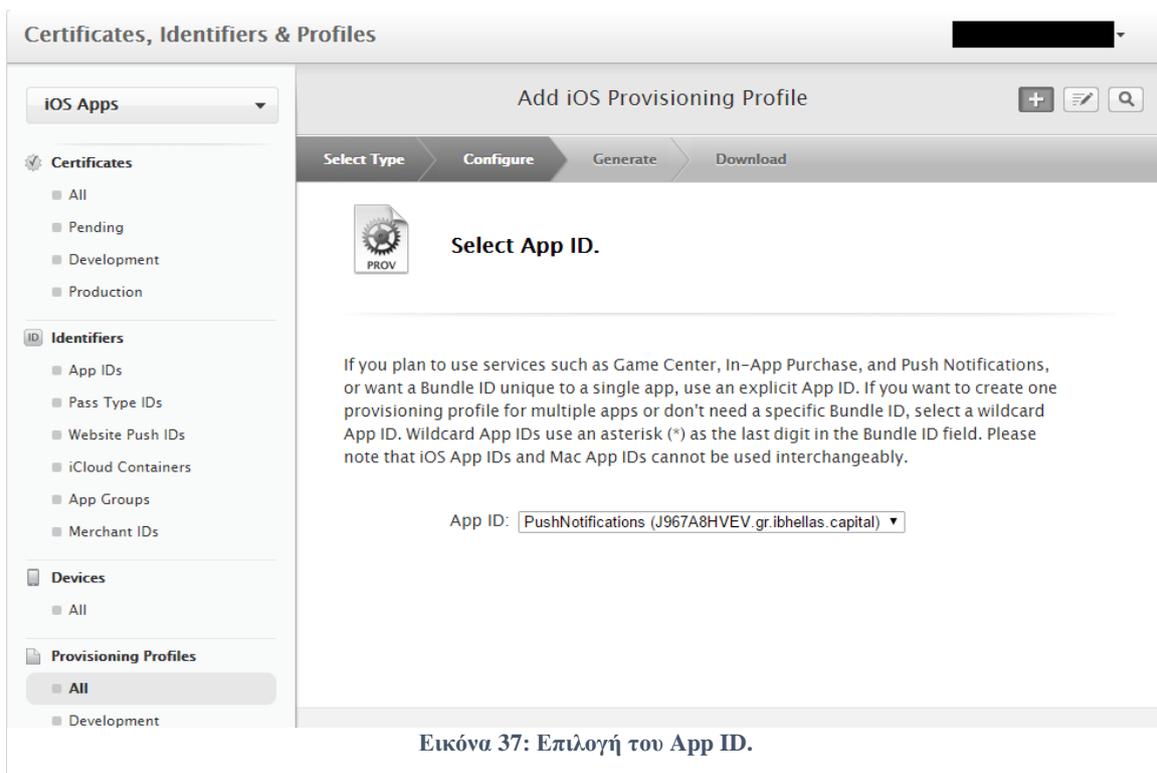
Επιλέγουμε το κουμπί επιλογής "iOS App Development" στο πρώτο βήμα του οδηγού, όπως φαίνεται στην παρακάτω εικόνα και πατάμε **Συνέχεια**.



Εικόνα 36: Διαδικασία δημιουργίας του provisioning profile.

Βήμα 2ο: Διαμόρφωση

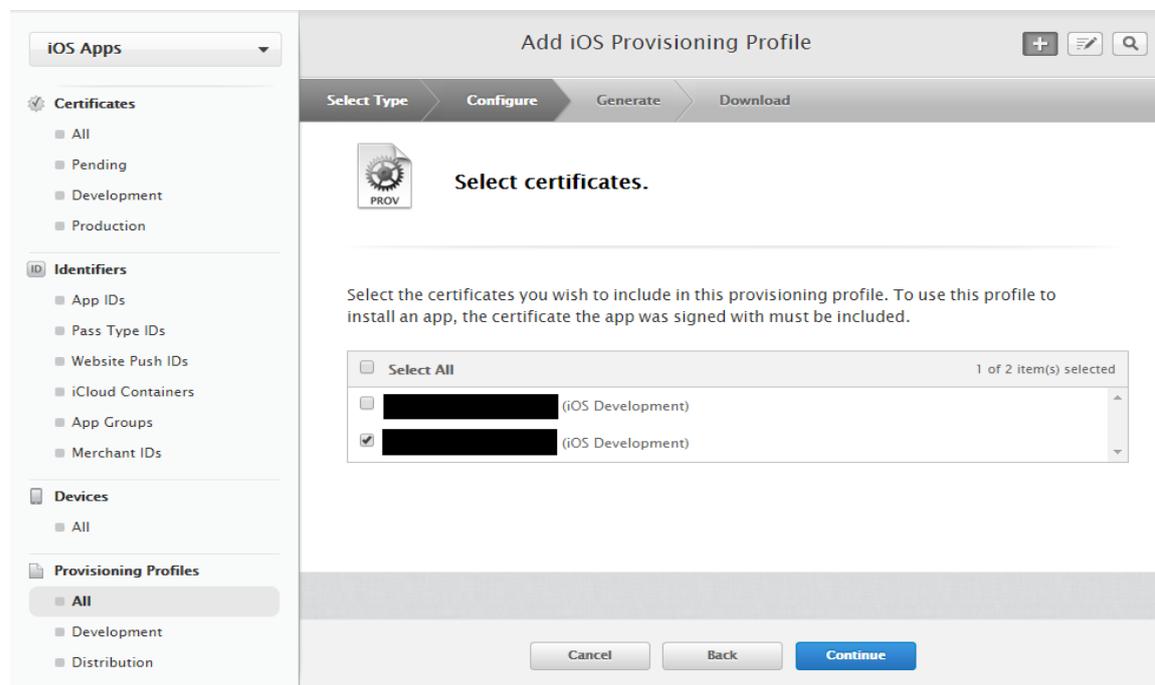
Επιλέγουμε το αναγνωριστικό Capital app που έχουμε δημιουργήσει, με αυτό τον τρόπο θα διασφαλίσουμε ότι αυτό το προφίλ είναι ρητά συνδεδεμένο με το Capital app.



Εικόνα 37: Επιλογή του App ID.

Βήμα 3ο: Δημιουργία

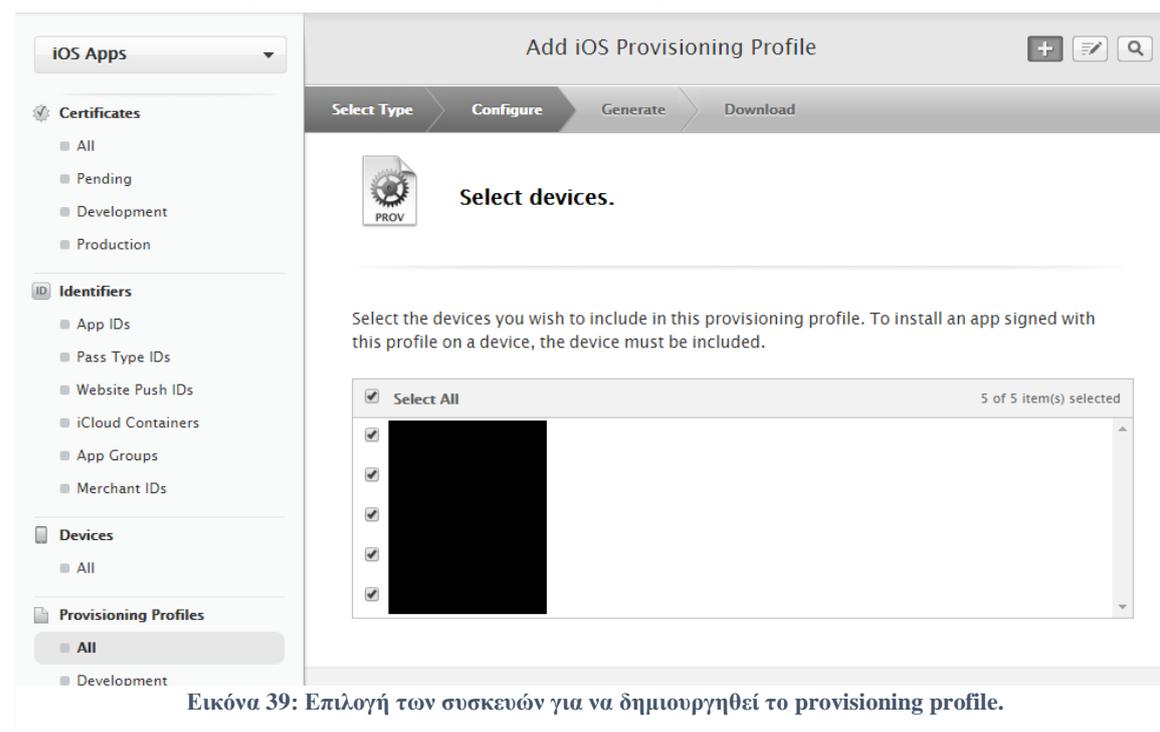
Σε αυτό το βήμα μπορούμε να επιλέξουμε τα πιστοποιητικά που θέλουμε να συμπεριλάβουμε σε αυτό το προφίλ.



Εικόνα 38: Επιλογή πιστοποιητικών.

Βήμα 4ο: Επιλέγουμε συσκευή

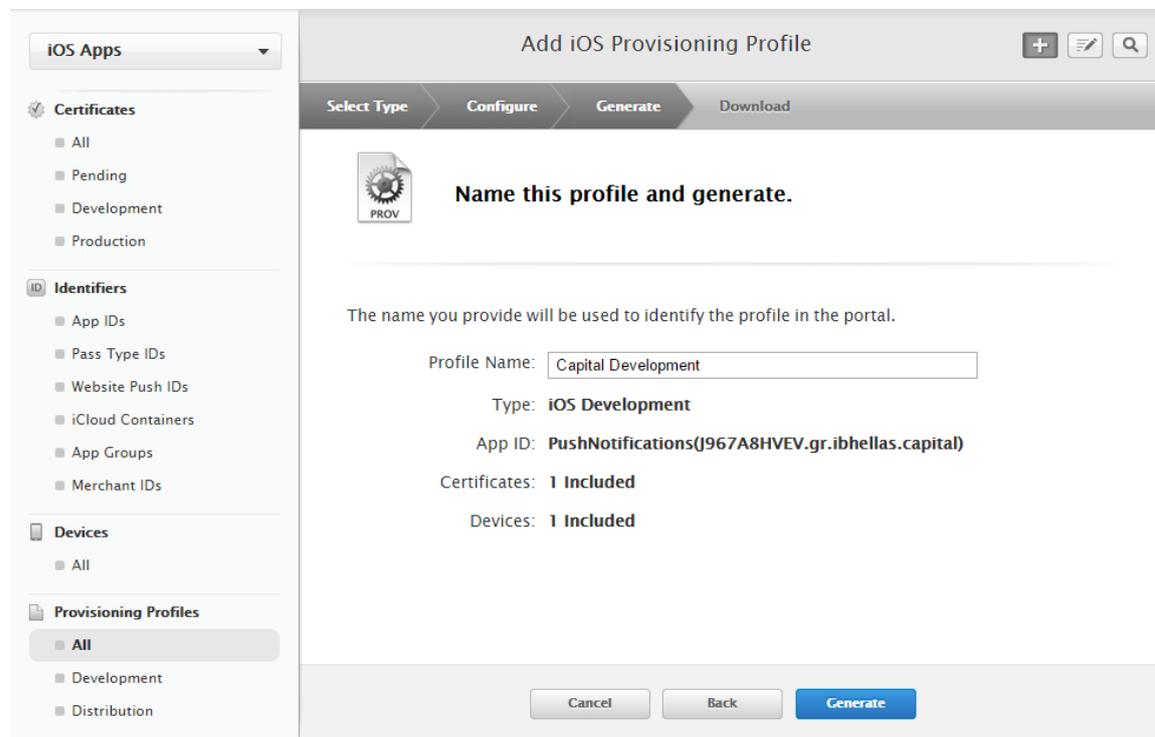
Επιλέγουμε τις συσκευές που θέλουμε να συμπεριλάβουμε στο προφίλ και από τη στιγμή που θα δημιουργήσουμε το προφίλ ανάπτυξης θα πρέπει να επιλέξουμε τυπικά τις συσκευές που χρησιμοποιούμε για την ανάπτυξη.



Εικόνα 39: Επιλογή των συσκευών για να δημιουργηθεί το provisioning profile.

Βήμα 5ο: Ονομάζουμε αυτό το προφίλ

Ρυθμίζουμε το όνομα του προφίλ μας ως "Capital Development" στη συγκεκριμένη περίπτωση και πατάμε Δημιουργία, όπως φαίνεται στην παρακάτω εικόνα.



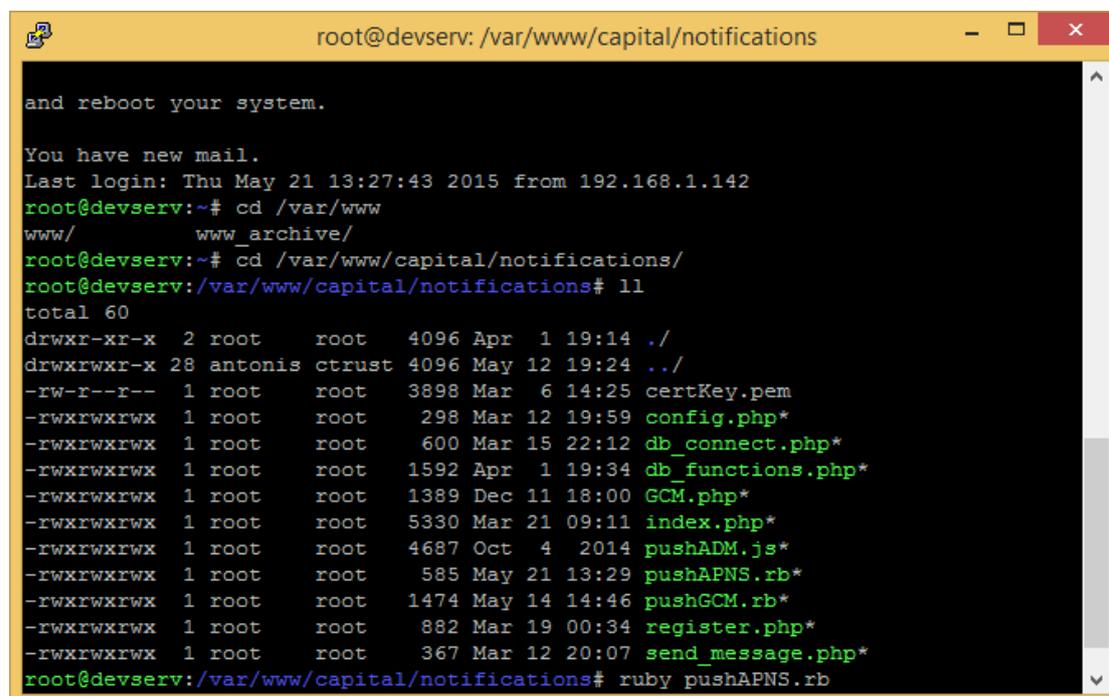
Εικόνα 40: Διαδικασία ονομασίας προφίλ.

Τέλος, πατάμε στο κουμπί Λήψη, ώστε να κατεβάσουμε το νέο προφίλ που μόλις δημιουργήσαμε.

4.3.5 Αποστολή push notifications της εφαρμογής Capital.gr σε συσκευή iOS

Η διαδικασία που ακολουθούμε για να στείλουμε push notification σε μια συσκευή iOS δεν διαφοροποιείται πολύ σε σχέση με τη διαδικασία που ακολουθήσαμε για το Android, εκτός από ορισμένα σημεία τα οποία θα δούμε ευθύς αμέσως.

Χρησιμοποιούμε και πάλι το πρόγραμμα **putty** ή οποιοδήποτε άλλο πρόγραμμα της αρεσκείας μας και ακολουθούμε τα βήματα κανονικά, αλλά αυτή τη φορά επιλέγουμε το αρχείο **pushAPNS.rb** και το εκτελούμε στην κονσόλα όπως φαίνεται και στην παρακάτω εικόνα, ώστε να σταλεί το notification στην κινητή μας συσκευή. Στην συνέχεια ακολουθεί ο κώδικας που περιέχει το συγκεκριμένο αρχείο, όπου στον προορισμό που του έχουμε ορίσει υπάρχει και ένα αρχείο **certKey.pem** στο οποίο εμπεριέχεται το πιστοποιητικό και ιδιωτικό κλειδί (private key) που έχουμε δημιουργήσει. Στην συνέχεια βρίσκεται το ανάλογο registration ID για iOS (device token) και τέλος ορίζουμε τις ανάλογες παραμέτρους που θα έχει το notification κάθε φορά που θα στέλνουμε στη συσκευή μας.



```
root@devserv: /var/www/capital/notifications
and reboot your system.
You have new mail.
Last login: Thu May 21 13:27:43 2015 from 192.168.1.142
root@devserv:~# cd /var/www
www/          www_archive/
root@devserv:~# cd /var/www/capital/notifications/
root@devserv:/var/www/capital/notifications# ll
total 60
drwxr-xr-x  2 root   root   4096 Apr  1 19:14 ./
drwxrwxr-x 28 antonis ctrust 4096 May 12 19:24 ../
-rw-r--r--  1 root   root   3898 Mar  6 14:25 certKey.pem
-rwxrwxrwx  1 root   root    298 Mar 12 19:59 config.php*
-rwxrwxrwx  1 root   root    600 Mar 15 22:12 db_connect.php*
-rwxrwxrwx  1 root   root   1592 Apr  1 19:34 db_functions.php*
-rwxrwxrwx  1 root   root   1389 Dec 11 18:00 GCM.php*
-rwxrwxrwx  1 root   root   5330 Mar 21 09:11 index.php*
-rwxrwxrwx  1 root   root   4687 Oct  4 2014 pushADM.js*
-rwxrwxrwx  1 root   root    585 May 21 13:29 pushAPNS.rb*
-rwxrwxrwx  1 root   root   1474 May 14 14:46 pushGCM.rb*
-rwxrwxrwx  1 root   root    882 Mar 19 00:34 register.php*
-rwxrwxrwx  1 root   root    367 Mar 12 20:07 send_message.php*
root@devserv:/var/www/capital/notifications# ruby pushAPNS.rb
```

Εικόνα 41: Εκτέλεση εντολών στην κονσόλα, για αποστολή push notification σε συσκευή iOS.

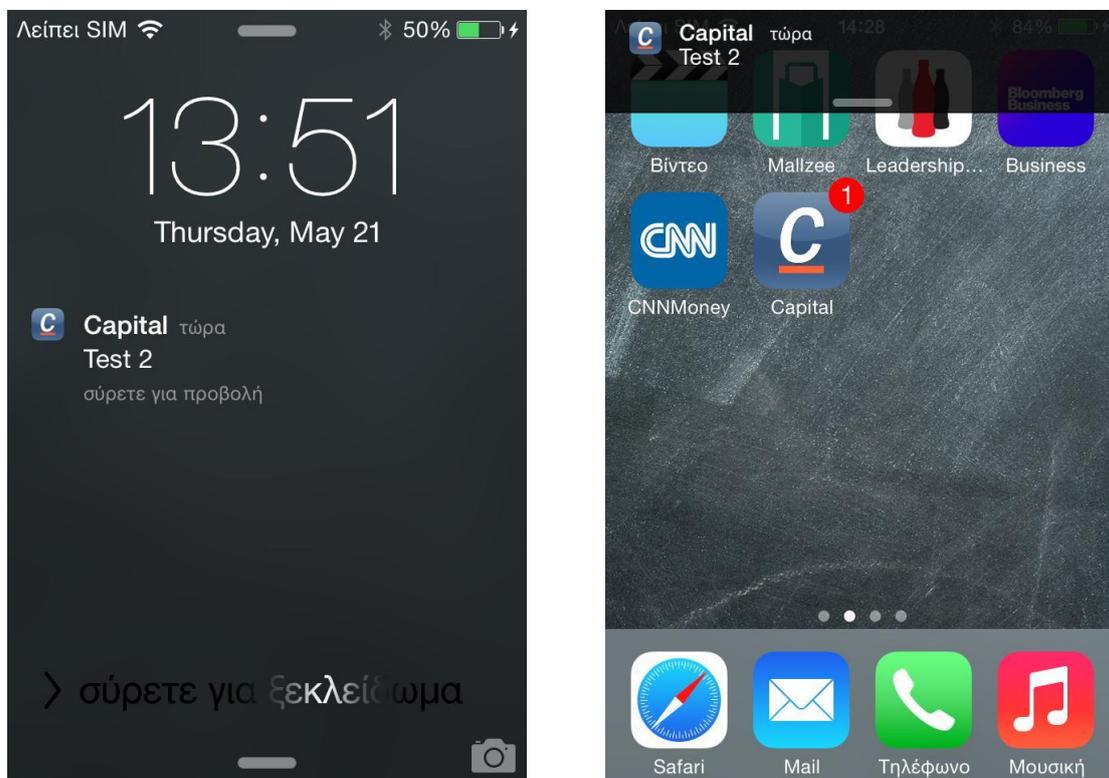
```
require 'rubygems'
require 'pushmeup'
APNS.host = 'gateway.sandbox.push.apple.com'
APNS.port = 2195
APNS.pem = '/var/www/capital/notifications/certKey.pem'
APNS.pass = '1bH3Il@$$P@$$'
```

device_token

```
= '779dc1895cd80caa618e94348a1931205cc86189b7dd02444f158bebaa8aa9c1'
```

```
APNS.send_notification(device_token, :alert => 'Test 2',:other => {:articleId => "2136701",:stockName => "BYTE"} , :badge => 1, :sound => 'beep.wav')
```

Στις παρακάτω δύο εικόνες βλέπουμε το αποτέλεσμα που θα μας εμφανίσει η iOS συσκευή μας με το που θα λάβουμε την ειδοποίηση. Στην πρώτη εικόνα έχουμε λάβει την ειδοποίηση, παρόλο που η συσκευή μας είναι κλειστή και μας εμφανίζει τις πληροφορίες που έχουμε ορίσει, ενώ στη δεύτερη βρισκόμαστε στο background της συσκευής και μας εμφανίζεται το νούμερο 1 στο εικονίδιο της εφαρμογής μαζί με το μαύρο πλαίσιο και το όνομα της που φαίνεται στο πάνω μέρος της οθόνης. Έτσι μας ενημερώνει ότι έχουμε λάβει κάποια ενημέρωση στην οποία πρέπει να προβούμε στην προβολή της.



Εικόνα 42: Αποτελέσματα αποστολής του notification στη συσκευή iOS.

5. Συμπεράσματα – Μελλοντικές επεκτάσεις

Η συγκεκριμένη πτυχιακή εργασία αναφέρθηκε στις πλατφόρμες ανάπτυξης υβριδικών εφαρμογών για κινητές συσκευές κάνοντας μία αναδρομή της εξέλιξης ανάπτυξης των εφαρμογών αυτών.

Επίσης παρουσιάστηκε η πλατφόρμα PhoneGap, με την οποία υλοποιήθηκε συγκεκριμένο έργο ανάπτυξης εφαρμογών για επιχειρηματικό όμιλο στα ΜΜΕ της χώρας.

Τα πλεονεκτήματα ανάπτυξης εφαρμογών με την χρήση cross platforms είναι πολλά. Κατά την προσωπική μου άποψη θεωρώ ότι η ανάπτυξη τέτοιου είδους εφαρμογών, θα μονοπωλήσουν την Βιομηχανία ανάπτυξης τα επόμενα χρόνια.

Συνοπτικά παραθέτω τους λόγους :

Με την ίδια υλοποίηση συμβατότητα σε διαφορετικά λειτουργικά συστήματα

- ✓ Χρήση τεχνολογιών “open source” και εργαλείων σε σχέση με την υλοποίηση
- ✓ Δυναμικά πλαίσια ανάπτυξης όπως ionic html5 που παρουσιάζουν πολλαπλά πλεονεκτήματα
- ✓ Χρήση javascript και πλαισίων ανάπτυξης εφαρμογών, όπως το Angular.js, της εταιρείας Google Corporation.
- ✓ Χαμηλότερο λειτουργικό κόστος για τις εταιρείες ανάπτυξης , λόγω του ότι με μία υλοποίηση καλύπτονται διάφορα λειτουργικά συστήματα. Κατ’αυτό τον τρόπο υπάρχει οικονομία κλίμακος στις επιχειρήσεις ανάπτυξης λογισμικού εφαρμογών, κάτι το οποίο λειτουργεί πολλαπλασιαστικά και για την τελική τιμή διάθεσης των εφαρμογών / υπηρεσιών στην αγορά αλλά και στην ανάπτυξη υγιών επιχειρηματικών μοντέλων με έμφαση στην εξειδίκευση και την καινοτομία.
- ✓ Οι εταιρείες πάροχοι λειτουργικών συστημάτων (Apple, Google, MicroSoft) έχουν δηλώσει την πρόθεσή τους για την ανάπτυξη του συγκεκριμένου μοντέλου και επενδύουν εκατομμύρια δολάρια για την επίτευξη του συγκεκριμένου σκοπού. Δεν είναι τυχαίο ότι η Google αναπτύσσει το πλαίσιο ανάπτυξης εφαρμογών Angular.js και δηλώνει ότι αποτελεί στρατηγική και τον πυλώνα ανάπτυξης της εταιρείας τα επόμενα χρόνια.
- ✓ Το κόστος των κινητών συσκευών έχει αρχίσει και φθίνει, με αποτέλεσμα την μεγαλύτερη διείσδυση στην αγορά αλλά και την αντικατάσταση των φορητών υπολογιστών από κινητές συσκευές (μεγάλες οθόνες με υψηλή ανάλυση, μεγάλη διάρκεια ζωής μπαταρίας, εξελιγμένοι επεξεργαστές, κ.α.)

Είναι κατανοητό ότι οι υβριδικές εφαρμογές αποτελούν το μοντέλο ανάπτυξης των εφαρμογών σε κινητές συσκευές. Αυτό θα είναι και το σκηνικό τα επόμενα χρόνια εισάγοντας την χρήση του διαδικτύου πραγμάτων (internet of things) το οποίο απαιτεί εκτός των άλλων διαχείριση μεγάλων ποσοτήτων δεδομένων (big data) αλλά και εξειδικευμένες καινοτόμες εφαρμογές κινητών συσκευών κάτι το οποίο επιτυγχάνεται με την χρήση cross platforms.

Περαιτέρω έρευνα μπορεί να επικεντρωθεί στην διαχείριση μεγάλων ποσοτήτων δεδομένων (big data) από υβριδικές εφαρμογές, καθώς και στην αρχιτεκτονική Ωθήσεων ειδοποιήσεων (push notifications).

Αναφορές

- [1] Google Corporation, διαθέσιμο σε: <https://www.google.com/about/company/>
- [2] Telerik.com, διαθέσιμο σε: <http://www.telerik.com/>.
- [3] Appcelerator Titanium, διαθέσιμο σε: <http://www.appcelerator.com/>.
- [4] Corona SDK, διαθέσιμο σε: <https://coronalabs.com/>.
- [5] Intel XDK, διαθέσιμο σε: <https://software.intel.com/en-us/intel-xdk>.
- [6] Sencha Touch 2.0, διαθέσιμο σε: <http://www.sencha.com/blog/announcing-sencha-touch-2/>.
- [7] Xamarin, διαθέσιμο σε: <http://xamarin.com/>.
- [8] unity3d.com, διαθέσιμο σε: <https://unity3d.com/>.
- [9] nodejs.org, διαθέσιμο σε: <https://nodejs.org/>.
- [10] Adobe Corporation. PhoneGap, διαθέσιμο σε: <http://phonegap.com/>.
- [11] Angular.js, διαθέσιμο σε: <https://angularjs.org/>.
- [12] ionic html5, διαθέσιμο σε: <http://ionicframework.com/>.
- [13] jquerymobile, διαθέσιμο σε: <https://jquerymobile.com/>.
- [14] jquery.com, διαθέσιμο σε: <https://jquery.com/>.
- [15] NGCORDOVA, διαθέσιμο σε: <http://ngcordova.com/docs/plugins/pushNotifications/>.
- [16] GCM (Google Cloud Messaging), διαθέσιμο σε: <https://developers.google.com/cloud-messaging/>.
- [17] Apple local and Push Notifications, διαθέσιμο σε: <https://developer.apple.com/notifications/>.
- [18] Apple push notifications services, διαθέσιμο σε: <http://www.raywenderlich.com/32960/apple-push-notification-services-in-ios-6-tutorial-part-1>.
- [19] Administration & creation Apple certificates, διαθέσιμο σε: <https://developer.apple.com/notifications/>.
- [20] clickmedia S.A, διαθέσιμο σε: <http://www.clickmedia.gr/>.
- [21] Sansour R., Kafri N., Sabha M., "A survey on mobile multimedia application development frameworks", Arab American University Computer Science Department, 31-1-2014.

[22] Flora H., Wang X., Chande S., "An investigation on the characteristics of Mobile Applications: A survey study", IJ Information Technology and Computer Science, 2014, 11, 21-27.

[23] Singh, I., Palmieri, M., (2011), "COMPARISON OF CROSS-PLATFORM MOBILE DEVELOPMENT TOOLS", Mälardalen University-Innovation, Development and Technology, 2011.

[24] Charland, A. & Leroux, B., (2011), "Mobile Application Development: Web vs. Native ", Communications of the ACM 54, 2011, pp. 49-53.

[25] w3schools.com, διαθέσιμο σε: <http://www.w3schools.com/>