



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

Σχολή Τεχνολογιών Πληροφορικής και Επικοινωνιών
Τμήμα Ψηφιακών Συστημάτων
Π.Μ.Σ Ψηφιακά Συστήματα και Υπηρεσίες
Κατεύθυνση Δικτυοκεντρικών Πληροφοριακών Συστημάτων

Ανάπτυξη υβριδικής εφαρμογής για κινητές συσκευές



Μεταπτυχιακή εργασία
Μαρκουλιδάκης Παναγιώτης AM 13049
Υπεύθυνος καθηγητής: Δουλκερίδης Χρήστος
Πειραιάς Φεβρουάριος 2016

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον Πατέρα μου Κώστα για την υποστήριξη του σε όλα τα εγχειρήματα της ζωής μου και που πάντα βρίσκεται δίπλα μου. Την μητέρα μου Ιλιάνα για την αμέριστη συμπαράσταση και την υπομονή της. Την Θεία μου Άλεξ για την υποστήριξή της ακόμα και σε άβατα νερά. Τον Φίλο μου Δημήτρη για την βοήθεια που μου πρόσφερε με τις εργασίες καθώς και την ψυχολογική υποστήριξη. Την φίλη μου Αννέζα για την βοήθειά της στην βελτίωση της εφαρμογής του σχεδιασμό του λογότυπου και την επιμονή της να τελειώσω.

Περίληψη

Οι κινητές συσκευές έχουν γίνει αναπόσπαστο κόμματα της καθημερινότητας. Σύμφωνα με μελέτη της ComScore οι χρήστες κινητών συσκευών έχουν ξεπεράσει σε πλήθος τους χρήστες ηλεκτρονικών υπολογιστών ενώ για το 2017 ο κύκλος εργασιών από την ανάπτυξη εφαρμογών για κινητές συσκευές θα ανέλθει σε 77 δις δολάρια. Στα πλαίσια της παρούσας θα εξετάσουμε θέματα ανάπτυξης εφαρμογών κινητών συσκευών τις διαθέσιμες τεχνολογίες καθώς και το παράδειγμα χρήσης της εφαρμογής OneApp, εφαρμογή για την παρακολούθηση ασφαλιστικού χαρτοφυλακίου.

Abstract

The use of mobile devices is quite common nowadays. According to ComScore Networks, an American internet analysis company, the number of mobile users exceeded the computer users since 2014. By 2017, its expected that over 268 billion downloads of mobile apps will generate \$77 billion worth of revenue. Native *mobile applications* are targeted at specific mobile platforms. This phenomenon imposes severe constraints, such as the use of different development environments, technologies, and APIs for each mobile platform, leading inevitably to a waste of development time and effort, and an increased maintenance cost. This master thesis focuses on cross platform mobile application development.

Περιεχόμενα

Περιεχόμενα	9
1. Εισαγωγή.....	13
1.1. Παρακίνηση.....	13
1.2. Σκοπός	13
1.3. Η ιστορία των κινητών τηλεφώνων.....	14
2. Τεχνολογικό υπόβαθρο	15
2.1. Mobile applications	15
2.1.1. Web mobile applications.....	15
2.1.2. Native Mobile applications	15
2.1.3. Hybrid Mobile Apps.....	16
2.2. Χρησιμοποιούμενες Τεχνολογίες	17
2.2.1. HTML 5.....	17
2.2.1.1. HTML5 APIs	19
2.2.1.2. Beta APIs	21
2.2.2. CSS.....	24
2.2.3. Javascript.....	25
2.2.3.1. JavaScript libraries/frameworks.....	25
2.2.4. XML	27
2.2.5. JSON	28
2.2.6. JWT	29
2.2.6.1. Χρήση των JWT.....	29
2.2.6.2. Παράδειγμα JSON Web Token.....	30
2.2.7. Docker	31
3. Ανάπτυξη Mobile Εφαρμογών.....	33
3.1. Εισαγωγή	33
3.2. Το Πρόβλημα του Κατακερματισμού	33

3.3.	Υβριδικές Εφαρμογές.....	36
3.3.1.	Παρούσα κατάσταση.....	36
3.3.2.	Ανάπτυξη υβριδικών εφαρμογών.....	37
3.3.2.1.	Web App.....	38
3.3.2.2.	WebView.....	38
3.3.2.3.	Plugins.....	38
3.3.3.	Υβριδικές εφαρμογές και κατακερματισμός.....	39
3.3.4.	Hybrid development frameworks.....	40
3.3.4.1.	Apache Cordova.....	40
3.3.4.2.	Crosswalk Project.....	40
4.	Ανάπτυξη εφαρμογής: OneApp.....	41
4.1.	Αρχιτεκτονική.....	42
4.2.	Ανάπτυξη εφαρμογής.....	44
4.2.1.	Δημιουργία νέου Project.....	45
4.2.2.	App designer.....	46
4.3.	Application Emulation.....	47
4.4.	Application Debugging.....	48
4.5.	Profiler.....	49
4.6.	Build.....	50
5.	Εγχειρίδιο χρήστη.....	53
6.	Συμπεράσματα.....	59
7.	Βιβλιογραφία.....	61

Πίνακας Εικόνων

Εικόνα 1 Παράδειγμα HTML 5 ιστοσελίδας	18
Εικόνα 2 Παράδειγμα κανόνα CSS	24
Εικόνα 3 Παράδειγμα επαλήθευσης φόρμας με χρήση Angular JS	27
Εικόνα 4 Παράδειγμα αντικειμένου JSON	28
Εικόνα 5 Διάγραμμα ακολουθίας JWT	30
Εικόνα 6 Παράδειγμα JWT. Αριστερά Base64 encoded. Δεξιά JSON format	30
Εικόνα 7: Android screen size fragmentation	34
Εικόνα 8 Χρησιμοποιούμενες εκδόσεις Android	35
Εικόνα 9 Salesforce: Hybrid vs Native mobile apps	36
Εικόνα 10 Αρχιτεκτονική Apache Cordova	37
Εικόνα 11 Apache Cordova	40
Εικόνα 12 Το λογότυπο της εφαρμογής, Σχεδιασμός Καπούτοσου Αννέζα	41
Εικόνα 13 Αναπαράσταση της αρχιτεκτονικής OneApp	42
Εικόνα 14 Project uno: αρχική οθόνη εισαγωγής	44
Εικόνα 15 Intel xdk: δημιουργία νέου project	45
Εικόνα 16 Το περιβάλλον εξομοίωσης του Intel XDK	47
Εικόνα 17 Intel XDK Debugging Session	49
Εικόνα 18 Intel XDK profiler	49
Εικόνα 19 Παράδειγμα build εφαρμογής	50
Εικόνα 20 Jarsigner	51
Εικόνα 21 Εικονίδιο εφαρμογής	53
Εικόνα 22 Οθόνη πιστοποίησης χρήστη	54
Εικόνα 23 αρχική οθόνη εφαρμογής	55
Εικόνα 24 Παράδειγμα καρτέλας πελάτη	56
Εικόνα 25 Αριστερά: Λίστα συμβολαίων πελάτη. Δεξιά: Εικόνα συμβολαίου	57
Εικόνα 26 Επισυναπτόμενα έγγραφα πελάτη	58

1. Εισαγωγή

1.1. Παρακίνηση

Με την ανάπτυξη της τεχνολογίας και την μετεξέλιξη του διαδικτύου σε Internet of Things (IoT) ο όρος κινητή συσκευή εκτός από τα κινητά τηλέφωνα επεκτείνεται σε έξυπνες συσκευές όπως Smartwatches Activity Trackers ακόμα και τηλεοράσεις. Οι εφαρμογές αυτές μπορεί να είναι προ-εγκατεστημένες στην συσκευή ή, να διανέμονται στους χρήστες μέσω ενός application store, ή να είναι προσβάσιμες μέσω ενός web-browser. Σε αντίθεση με την παραδοσιακή ανάπτυξη εφαρμογών(εφαρμογές για computers), στο mobile development ο προγραμματιστής πρέπει να λάβει υπόψη του την πληθώρα των κινητών συσκευών, των χαρακτηριστικών(ανάλυση οθόνης, hardware specs), των διαφορετικών εκδόσεων των λειτουργικών συστημάτων που χρησιμοποιούν οι συσκευές αυτές ώστε να αποφασίσει για τον τρόπο που θα αναπτύξει την εφαρμογή. Σύμφωνα με μελέτη της OpenSignal¹, τον Αύγουστο του 2015 υπήρχαν 24.093 μοναδικές συσκευές Android ενώ αντίστοιχα το 2014 υπήρχαν 18.796. Ο χώρος του Mobile Development ανεβαίνει συνεχώς και υπολογίζεται² ότι η αγορά για το 2016 θα ξεπεράσει τα 143δισ δολάρια ενώ τα κέρδη το 2017 θα είναι περί τα 77δισ δολάρια.

1.2. Σκοπός

Μέσα από την εργασία αυτή προσπάθησα να αποκτήσω όλη την απαραίτητη γνώση σε θεωρητικό και την τεχνική εμπειρία που χρειάζεται για την ανάπτυξη μιας εμπορικής - OneApp - εφαρμογής για χρήση σε κινητές συσκευές με λειτουργικό Android και iPhone. Το OneApp είναι μια εφαρμογή για την υποστήριξη ασφαλιστικών συμβούλων και την πρόσβαση τους σε πραγματικό χρόνο στο χαρτοφυλάκιο τους από παντού. Ένας από τους

¹ OpenSignal Android Fragmentation Visualized Αύγουστος 2015

² Developer Economics: State of the Developer Nation Q3.2015

βασικούς στόχους του εγχειρήματος ήταν λειτουργία της εφαρμογής ανεξαρτήτως λειτουργικού στις τερματικές συσκευές.

1.3. Η ιστορία των κινητών τηλεφώνων

Η ιδέα για την δημιουργία κινητών τηλεφώνων επικοινωνίας ξεκίνησε λίγο μετά την λήξη του Β' Παγκοσμίου πολέμου με την Σουηδία, την Αμερική και την Φιλανδία να ηγούνται. Τριάντα περίπου χρόνια αργότερα την 3^η Απριλίου του 1973 Ο Dr. Martin Cooper πατέρας των κινητών τηλεφώνων και τότε Senior Developer στην Motorola πραγματοποίησε την πρώτη κλήση³ μέσω κινητού τηλεφώνου προς τον Joel Engel υπεύθυνο για το πρόγραμμα κινητών συσκευών των Bell Labs (AT&T σήμερα). Σταθμός στην εξέλιξη των δικτύων κινητής τηλεφωνίας στις αρχές του 1990 ήταν το GSM το οποίο σήμανε την ψηφιοποίηση των δικτύων κινητής τηλεφωνίας και την περίοδο 1G. Ραγδαίες εξελίξεις στην τεχνολογία όπως το CDMA, HSDPA, WCDMA, και ολοκλήρωση των κυκλωμάτων σε μικρότερες κλίμακες καθώς και άλλες τεχνολογικές εξελίξεις είχαν σαν αποτέλεσμα την συρρίκνωση των συσκευών, την ενσωμάτωση πρόσθετων χαρακτηριστικών (κάμερα, αναγνώστη δακτυλικών αποτυπωμάτων, αισθητήρες μέτρησης των συνθηκών του περιβάλλοντος) αλλαγές που αύξησαν ραγδαία το ποσοστό των χρηστών. Σήμερα τα δίκτυα 4^{ης} γενιάς είναι Packet-switched χρησιμοποιούν το Internet Protocol και η ταχύτητα τους ορίζεται στα 100Mb/s⁴ για High mobility επικοινωνίες – χρήστες που κινούνται με τρένο ή αυτοκίνητο - και 1Gb/s για low mobility επικοινωνίες, χρήστες που βρίσκονται σε σταθερό σημείο ή περπατάνε.

³ The Economist: Father of the Cell Phone, Ιούνιος 2009

⁴ Wikipedia 4G

2. Τεχνολογικό υπόβαθρο

2.1. Mobile applications

Οι εφαρμογές κινητών σχεδιάζονται και αναπτύσσονται για συγκεκριμένες πλατφόρμες και φορητές συσκευές όπως κινητά τηλέφωνα, tablets, PDA. Οι εφαρμογές αυτές όπως αναφέραμε ήδη είναι διαθέσιμες μέσω κάποιου application store ή μέσω κάποιας ιστοσελίδας. Οι εφαρμογές για κινητές συσκευές χωρίζονται σε τρεις κατηγορίες ανάλογα με την τεχνολογία που χρησιμοποιούν.

- Web mobile applications
- Native mobile applications
- Hybrid Mobile applications

2.1.1. Web mobile applications

Τα web mobile apps δεν είναι κάτι παραπάνω από mobile friendly websites. Είναι γραμμένα με τέτοιο τρόπο ώστε να προσαρμόζονται στην οθόνη της κινητής συσκευής και είναι βελτιστοποιημένες (είναι;) ώστε να χρησιμοποιούν ελάχιστο όγκο δεδομένων. Είναι προσβάσιμες μέσα από τον Web browser της κινητής συσκευής και έχουν περιορισμένη πρόσβαση στους πόρους τους συστήματος.

2.1.2. Native Mobile applications

Οι Native εφαρμογές λειτουργούν στις κινητές συσκευές και έχουν αναπτυχθεί για συγκεκριμένο τύπο συσκευής με συγκεκριμένα χαρακτηριστικά με συγκεκριμένη πλατφόρμα και συγκεκριμένο framework το οποίο συνήθως υποδεικνύεται από τον κατασκευαστή. Παραδείγματος χάρη Objective C και Xcode για το Apple IOS, Java και Android Studio ή Eclipse για το Google Android και C# και Visual Studio για το Microsoft Windows Phone. Οι

εφαρμογές αυτές μπορεί να είναι προ εγκατεστημένες μαζί με το λειτουργικό σύστημα της συσκευής ή να διανέμονται μέσω από το αντίστοιχο application store του λειτουργικού.

2.1.3. Hybrid Mobile Apps

Υβριδικές εφαρμογές είναι εκείνες οι εφαρμογές που έχουν αναπτυχθεί με εργαλεία τα οποία δεν επιβάλλουν την χρήση τους σε κάποιο συγκεκριμένο περιβάλλον (Android,IOS,Firefox OS Ubuntu OS κ.ο.κ) αλλά μπορούν να γίνουν compiled σε οποιοδήποτε περιβάλλον επιθυμεί ο developer. Ο κύκλος ανάπτυξης παραμένει ίδιος με αυτών που παραδοσιακά χρησιμοποιείται, η διάθεση των εφαρμογών γίνεται μέσα από το κανάλι διανομής της πλατφόρμας(PlayStore, AppStore) και το αντικείμενό τους μπορεί να είναι παιχνίδι όπως το Halo της Microsoft ή μια εφαρμογή κοινωνικής δικτύωσης Facebook. Οι υβριδικές εφαρμογές μοιάζουν αρκετά με τις ιστοσελίδες, κάνουν χρήση τεχνολογιών web: HTML, CSS. JavaScript με την διαφορά ότι δεν εκτελούνται μέσα από τον Browser της συσκευής αλλά μέσα σε ένα WebView. Το WebView είναι ένας μηχανισμός για την προβολή σελίδων HTML το οποίο όμως επιτρέπει να γίνει χρήση του υλικού του συστήματος(Accelerometer,αναγνώστης δακτυλικών αποτυπωμάτων κοκ).

Το οπτικό αποτέλεσμα μιας υβριδικής εφαρμογής είναι ίδιο με αυτό που θα είχαμε αν την ίδια την εφαρμογή την αναπτύσσαμε για συγκεκριμένη πλατφόρμα (λχ IOS). Για τον τελικό χρήστη είναι δύσκολο να καταλάβει αν αυτό που βλέπει είναι ένα hybrid mobile app. Μια καλογραμμένη υβριδική εφαρμογή δεν διαφέρει οπτικά και μπορεί να κάνει χρήση χαρακτηριστικών που είναι διαθέσιμα σε συγκεκριμένη πλατφόρμα.

2.2. Χρησιμοποιούμενες Τεχνολογίες

2.2.1. HTML 5

Η HTML5 (Hyper Text Markup Language) είναι μια γλώσσα σήμανσης (Markup language) αποτελεί τον θεμέλιο λίθο του Web και είναι ο αντικαταστάτης της HTML 4.01. Υπεύθυνη ομάδα για την ανάπτυξη της HTML 5 είναι το World Wide Web Consortium(W3C) και το Web Hypertext Application Technology Working Group (WHATWG). Το πρώτο draft δημοσιεύτηκε το Ιανουάριο του 2007. Η HTML5 ορίζει μοντέλα διεργασιών με σκοπό την ανάπτυξη της διαλειτουργικότητας. Επεκτείνει και εξορθολογίζει την σημειολογία και μέσω προγραμματιστικών διεπαφών (API) επιτρέπει την ανάπτυξη σύνθετων web εφαρμογών. Για του ίδιους λόγους η HTML 5 είναι η προτεινόμενη επιλογή για την ανάπτυξη υβριδικών εφαρμογών ενώ έχει δοθεί μεγάλη προσοχή στις φορητές συσκευές και στις συσκευές με χαμηλότερες δυνατότητες.

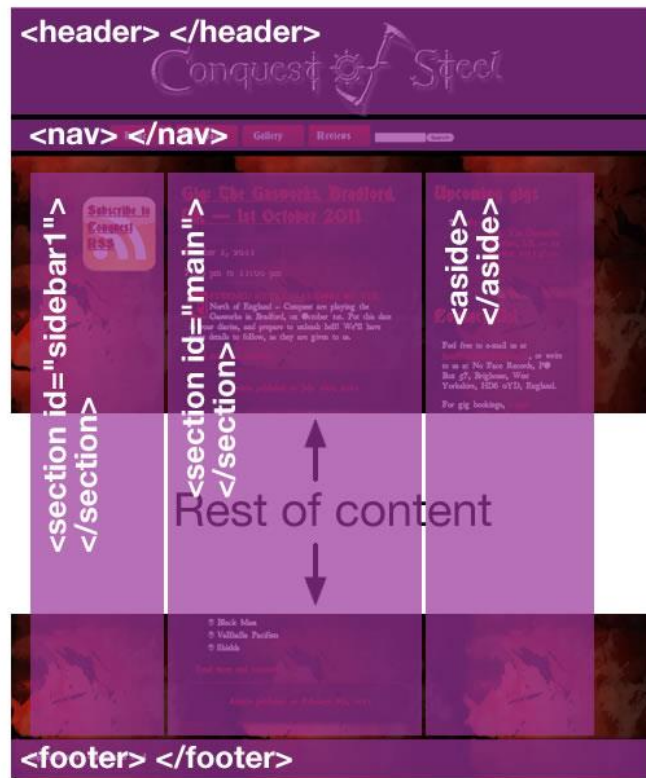
Πολλές από τις ετικέτες που χρησιμοποιεί η HTML έχουν τροποποιηθεί ώστε ο προγραμματιστής να έχει καλύτερο έλεγχο πάνω στα δεδομένα διευκολύνοντας έτσι το “form validation”(Web Forms 2) μια διαδικασία που παραδοσιακά γινόταν με χρήση JavaScript. Έναν από τους στόχους που έχει η HTML5 είναι να ομογενοποιήσει τις τεχνολογίες του Web. Εκτός από τα ανοικτά πρότυπα που αντικατέστησε η HTML 5 όπως την HTML 4, XHTML 1 και την DOM 12 HTML η HTML5 αντικαθιστά και εμπορικές λύσεις όπως το Adobe Flash.⁵ Οι νέες ετικέτες multimedia (<video>, <canvas>, <audio>) εξυπηρετούν αυτόν τον σκοπό και έρχονται να επιλύσουν προβλήματα ασυμβατότητας που υπήρχαν στο παρελθόν με διάφορες πλατφόρμες(Linux Android IOS) και προβλήματα ασφαλείας. Εκτός από τις ετικέτες των πολυμέσων η HTML 5 φέρνει ριζικές αλλαγές στον σχεδιασμό των ιστοσελίδων. Παραδοσιακά οι developers χρησιμοποιούσαν την ετικέτα div – ένα block element⁶- για να δημιουργήσουν τον σκελετό της ιστοσελίδας. Στην HTML 5 έχουν προστεθεί έξι⁷ δομικές ετικέτες απλοποιώντας έτσι τον σχεδιασμό μιας ιστοσελίδας και δίνει την δυνατότητα στην

⁵ Adobe: Flash HTML5 and open WEB Standards

⁶ Block Elements: Αντικείμενα που ξεκινάνε σε νέα γραμμή του εγγράφου και καταλαμβάνουν το 100% του width

⁷ W3.org: HTML structural elements

ίδια την συσκευή έναν μηχανισμό για να αντιλαμβάνεται το είδος του περιεχομένου. Η δυνατότητα αυτή είναι αρκετά χρήσιμη για την ανάπτυξη ιστοσελίδων για ανθρώπους με προβλήματα όρασης. Ένα παράδειγμα δομής μιας ιστοσελίδας σχεδιασμένη σε HTML 5 φαίνεται στην εικόνα 4.



Εικόνα 1 Παράδειγμα HTML 5 ιστοσελίδας

- **<header>**: Καθορίζει την επικεφαλίδα της ιστοσελίδας. Συνήθως εδώ τοποθετείται το λογότυπο.
- **<nav>**: Φιλοξενεί το <menu>⁸ της ιστοσελίδας.
- **<section>**: Χρησιμοποιείται για την ομαδοποίηση περιεχομένου διαφορετικού χαρακτήρα από το κυρίως αντικείμενο της ιστοσελίδας (Στην πράξη χρησιμοποιείται για την προβολή διαφημίσεων.)
- **<article>**: Περιέχει την κεντρική πληροφορία της ιστοσελίδας

⁸ Το <menu> υπήρχε στην HTML 4 ως υπό κατάργηση αντικείμενο στην HTML 5.1. Mozilla Developer Network: <menu>

- **<aside>**: Καθορίζει μια περιοχή περιεχομένου συγγενή με το κυρίως περιεχόμενο. Παραδείγματος χάρη σε μια ειδησεογραφική ιστοσελίδα τα “related articles” θα εμφανιζόντουσαν μέσα από <aside>.
- **<footer>**: Ορίζει την ποδίδα⁹ της ιστοσελίδας.

2.2.1.1. HTML5 APIs

Αναφέραμε νωρίτερα ότι ή HTML 5 είναι κατάλληλη για την δημιουργία υβριδικών εφαρμογών καθώς παρέχει διάφορα APIs για την αλληλεπίδραση με τον χρήστη και την λήψη εξωτερικών δεδομένων από την συσκευή.

WebGL

Επιτρέπει στο περιεχόμενο μιας ιστοσελίδας να κάνει 3D rendering με χρήση του OpenGL ES 2.0 χωρίς την χρήση εξωτερικών βιβλιοθηκών. Η λειτουργία αυτή βασίζεται πάνω στο <canvas>. Οι εφαρμογές σε WebGL αποτελούνται από δύο μέρη, α) τον κώδικα που ελέγχει την εφαρμογή γραμμένη σε Java και β) ειδικό κώδικα που καθορίζει τα εφέ. Το rendering των γραφικών γίνεται με χρήση της GPU τα WebGL αντικείμενα μπορούν να αναμειχθούν με τα υπόλοιπα στοιχεία της ιστοσελίδας και να συνθέσουν μια συνολική εμπειρία για τον χρήστη.(UX user experience).

Subsource Integrity

Το Subsource Integrity(SRI) είναι ένας μηχανισμός ασφαλείας που επιτρέπει στον browser να πιστοποιήσει ότι τα αρχεία που έχει κατεβάσει από εξωτερικούς συνδέσμους(CDN¹⁰) δεν έχουν αλλοιωθεί. Ο browser θα κατεβάσει το αρχείο θα υπολογίσει το HASH βάσει του αλγορίθμου που έχει υπαγορεύσει ο προγραμματιστής και αν αυτά ταιριάζουν τότε θα χρησιμοποιήσει το έγγραφο.

⁹ Βάση Τηλεπικοινωνιακών όρων: Footer= ποδίδα

¹⁰ Content Distribution Network

Ambient Light

Παρέχει πληροφορίες στην εφαρμογή σχετικά με τις συνθήκες σχετικού φωτός. Τα δεδομένα αυτά συλλέγονται από αισθητήρες φωτός ή άλλους παρεμφερείς αισθητήρες. Το Ambient Light προς το παρόν υποστηρίζεται μόνο από την τελευταία έκδοση(45¹¹) του Firefox για κινητές συσκευές.

Beacon API

Παρέχει την δυνατότητα αποστολής δεδομένων ασύγχρονα κατά το κλείσιμο της ιστοσελίδας. Η συγκεκριμένη λειτουργία δίνει την δυνατότητα στις εφαρμογές να μεταδώσουν μικρό¹² όγκο δεδομένων όπως δεδομένα στατιστικού χαρακτήρα ή δεδομένα debugging.

File API

Το File API επιτρέπει στον developer μέσω της JavaScript να αποκτήσει πρόσβαση σε ένα αρχείο που έχει ανεβάσει ο χρήστης μέσω φόρμας ή μέσω “Drag n drop”. Μέσω του File API ο προγραμματιστής μπορεί να φτιάξει μηχανισμό προεπισκόπησης αρχείων, δυνατότητα που για να την αποκτήσει παλαιότερα έπρεπε να κάνει χρήση του Adobe Flash ή κάποιου άλλου Plug-in.

High Resolution Time

Επιστρέφει την τρέχουσα ώρα με ακρίβεια εκατοστού του χιλιοστού του δευτερολέπτου. Το HRT παρέχει την «performanceTiming»(PT) η οποία επιστρέφει τα milliseconds από την

¹¹ Can I use? Ambient Light: <http://caniuse.com/ambient-light>

¹² Το Beacon Api δεν έχει γίνει ακόμα standardized και η υλοποίηση του είναι μερικώς ολοκληρωμένη.

στιγμή που η σελίδα ξεκίνησε να φορτώνει. Σε αντίθεση με την “Date” το PT δεν βασίζεται στο ρολόι του συστήματος κάνοντάς την έτσι ιδανική για την παρακολούθηση των κινήσεων του χρήστη. Η δυνατότητα αυτή είναι αρκετά χρήσιμη για Gaming development, Animation, benchmarking.

WebWorkers

Για αρκετά μεγάλο χρονικό διάστημα υπολογιστές και κινητές συσκευές χρησιμοποιούσαν μονοπύρηνους επεξεργαστές. Τα τελευταία 15 χρόνια οι επεξεργαστές έχουν γίνει πολυπύρηντοι ενώ τα τελευταία χρόνια έχουν αρχίσει να υπάρχουν και πολυπύρηνες φορητές συσκευές. Το WebWorkes API παρέχουν με εύκολο τρόπο έναν μηχανισμό στην εφαρμογής μας για την εκτέλεση κώδικα στο background. Ο κώδικας αυτός τρέχει σε ξεχωριστό νήμα και ενημερώνει την εφαρμογή για την ολοκλήρωσή του μέσω ενός event handler. Με τον τρόπο αυτό μπορούμε να τρέξουμε απαιτητικά κομμάτια κώδικα (Data processing, Games) χωρίς να σταματάμε το Main thread και άρα να επηρεάζουμε την εμπειρία του χρήστη.

2.2.1.2. Beta APIs

Η HTML 5 προτείνει αρκετά API προκειμένου να επεκτείνει τις δυνατότητες της JavaScript δίνοντας έτσι στους προγραμματιστές τα κατάλληλα εργαλεία για να φτιάξουν σύνθετες εφαρμογές. Υπεύθυνοι για την υλοποίηση των standard και την ενσωμάτωση νέων χαρακτηριστικών στα web engines είναι η εκάστοτε κοινότητα. Για τον Gecko (web engine του Mozilla Firefox) είναι το Mozilla Foundation και το Mozilla Developer Network. Είναι κατανοητό λοιπόν ότι δεν υλοποιούνται όλα τα προτεινόμενα χαρακτηριστικά στον ίδιο χρόνο και στον ίδιο βαθμό. Ανάλογα με τους στόχους της κοινότητας ή και την αναμενόμενη ζήτηση είναι και το ενδιαφέρον που δείχνουν οι προγραμματιστές. Τα επόμενα APIs που αναφέρονται βρίσκονται σε δοκιμαστικό στάδιο είτε επειδή το standard είναι υπό διαβούλευση (draft) ή επειδή δεν έχει ακόμα υιοθετηθεί από τα web engines. Τα ποσοστά ενσωμάτωσης που αναγράφονται έχουν ληφθεί από το caniuse.com ιστοσελίδα που παρακολουθεί την υλοποίηση των χαρακτηριστικών ανάμεσα στους δημοφιλείς Browser. Για

την ανάπτυξη υβριδικών εφαρμογών σε Android & IOS μας ενδιαφέρουν τα στατιστικά του Chrome(WebKit) ενώ για Windows 10 ο Edge.

Bluetooth (0%)

Παρέχει πρόσβαση σε κοντινές συσκευές μέσω Bluetooth. Λόγω των δυνατοτήτων της τεχνολογίας Bluetooth και του είδους των συσκευών που την χρησιμοποιούν(Activity/Health trackers, έξυπνοι λαμπτήρες) στον Google Chrome(έκδοση 50)η συγκεκριμένη διεπαφή είναι διαθέσιμη μόνο σε ασφαλές περιβάλλον, απαιτεί δηλαδή η εφαρμογή να χρησιμοποιεί TLS.

Speech Synthesis (58%)

Παρέχει την δυνατότητα μετατροπής κειμένου σε ομιλία. Μέσω της συγκεκριμένης διεπαφής ο προγραμματιστής μπορεί να αναπτύξει εφαρμογές που επιτρέπουν φωνητικές οδηγίες, φωνητική πλοήγηση ή να βελτιώσουν την εμπειρία χρήσης της εφαρμογής από άτομα με ειδικές ανάγκες.

Permissions API (47,58%)

Οι συσκευές ολοένα γίνονται και πιο έξυπνες. Νέοι αισθητήρες προστίθενται σε αυτές αυξάνοντας τις πληροφορίες που μπορεί να συλλέξει η συσκευή τόσο για τον χρήστη της όσο και στο περιβάλλον στο οποίο βρίσκεται. Το μεγάλο πρόβλημα που τίθεται είναι τι συμβαίνει με την ασφάλεια του χρήστη. Μπορεί μια εφαρμογή να έχει πρόσβαση σε όσα δεδομένα θέλει; Η απάντηση είναι όχι. Η εφαρμογή θα πρέπει να ζητήσει πρόσβαση στα δεδομένα που χρειάζεται και αν ο χρήστης το επιθυμεί τότε η εφαρμογή θα αποκτήσει πρόσβαση στους απαιτούμενους πόρους. Το Permissions API έρχεται να καλύψει την ανάγκη αυτή. Σκοπός του είναι να ενημερώσει τον χρήστη για τας δικαιώματα που ζητάει η εφαρμογή π.χ πρόσβαση στην τοποθεσία ή στα τοπικά αρχεία και να δώσει ή να αρνηθεί την πρόσβαση στην εφαρμογή ανάλογα με την επιλογή του χρήστη. Από την πλευρά του προγραμματιστή το Permissions API κάθεται πάνω από τα υπόλοιπα API προσφέροντας έτσι μια ενιαία διεπαφή για την διαχείριση των δικαιωμάτων.

Network Information API (18.96%)

Ο Responsive σχεδιασμός των εφαρμογών έχει φέρει επανάσταση στις web εφαρμογές. Η εφαρμογή μπορεί να προσαρμόζεται εύκολα στις διαφορετικές αναλύσεις με την χρήση μερικών “media queries” να αλλάζει δομή να αλλάζει γραμματοσειρές ή οποιαδήποτε άλλη αλλαγή χρειάζεται. Το να κρίνουμε τις δυνατότητες του browser από την ανάλυση της οθόνης είναι σαν να προσπαθούμε να πούμε πόσο γρήγορο είναι ένα αμάξι κοιτάζοντας τα χαλάκια του. Το Network API δίνει την απαραίτητη πληροφόρηση που χρειάζεται μια Web εφαρμογή για την κατάσταση του δικτύου. Μπορεί να ενημερώσει την εφαρμογή για την ταχύτητα της ενεργής σύνδεσης ένα πρόκειται για σύνδεση με περιορισμό δεδομένων(metered connection) και αν υπάρχει ενεργή σύνδεση.

WebRTC Peer-to-peer API (57.9%)

Επιτρέπει στις web εφαρμογές την δημιουργία επικοινωνίας σε πραγματικό χρόνο με κανάλια ήχου εικόνας ή δεδομένων ανάμεσα σε χρήστες ή με Server που υλοποιεί το WebRTC πρωτόκολλο. Το συγκεκριμένο API απαιτεί πρόσβαση στις συσκευές πολυμέσων (Μικρόφωνο, κάμερα).

Web Notifications (45,61%)

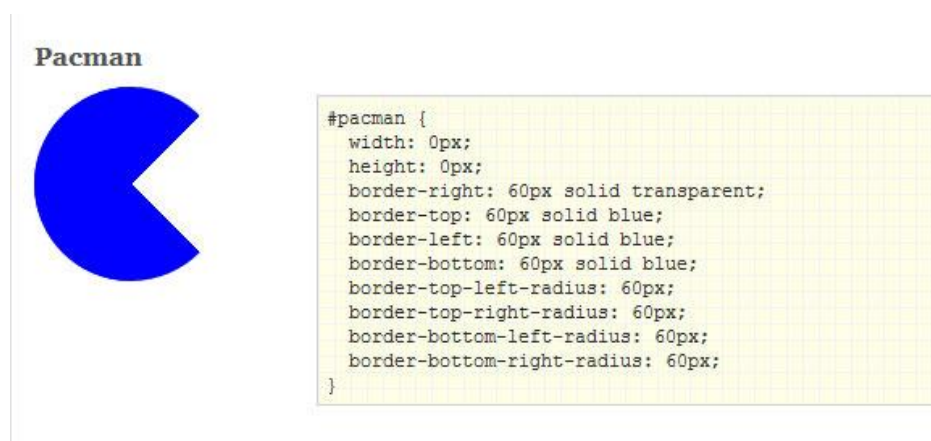
Μια από τις πιο διαδεδομένες μεθόδους αλληλεπίδρασης των εφαρμογών με τον χρήστη είναι η δημιουργία «ειδοποιήσεων». Τα Notifications είναι μια διεπαφή για την ειδοποίηση του χρήστη έξω από το περιβάλλον της εφαρμογής χωρίς να χρειάζεται να αλληλοεπιδράσει εκείνος. Παράδειγμα τέτοιας ειδοποίησης είναι το μήνυμα που εμφανίζεται κατά την λήψη ενός email.

2.2.2. CSS

Τα Cascading Style Sheets (Φύλλα μορφοποίησης αλληλουχίας) είναι μια γλώσσα εφαρμογής στυλ για έγγραφα γραμμένα σε γλώσσα σήμανσης (HTML XHTML κ.ο.κ.). Χρησιμοποιείται δηλαδή για να προσαρμόσει την εμφάνιση μιας ιστοσελίδας και είναι το βασικό εργαλείο για την δημιουργία του UX (user experience). Με τα CSS μπορούμε να ορίσουμε βασικά στοιχεία της μορφής μια ιστοσελίδας όπως το μέγεθος της γραμματοσειράς και το χρώμα φόντου ή και να ορίσουμε πιο σύνθετους κανόνες όπως την απόκρυψη κάποιων στοιχείων του εγγράφου για συγκεκριμένες συνθήκες προβολής.

Τα CSS είναι εφαρμόζουν τους κανόνες ιεραρχικά ανάλογα με την θέση ενός κανόνα μέσα στο αρχείο (line number) και το πόσο συγκεκριμένος είναι ένας κανόνας. Οι κανόνες ορίζονται για κάποιον ή κάποιους selector(s). Έτσι μπορούμε να εφαρμόσουμε κανόνες για διαφορετικές ετικέτες. Οι βασικοί selectors είναι:

- **ID:** Αποτελεί το μοναδικό αναγνωριστικό μιας ετικέτας και αποτελεί την πιο δυνατή μορφή selector
- **CLASS:** Ομαδοποιεί ετικέτες με κοινά χαρακτηριστικά. Είναι η πιο διαδεδομένη μορφή selector.
- **ELEMENT:** Είναι η πιο βασική μορφή selector και μορφοποιεί την οικογένεια μιας ετικέτας (Π.χ <p>).



Εικόνα 2 Παράδειγμα κανόνα CSS

2.2.3. Javascript

Η JavaScript είναι μια γλώσσα σεναρίων, με ασθενείς τύπους, συναρτήσεις και υποστηρίζει τον προγραμματισμό γεγονότων. Η σύνταξή της είναι επηρεασμένη από τη C ενώ πολλά ονόματα και συμβάσεις ονοματοδοσίας προέρχονται από την Java. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Το πρότυπο της γλώσσας ονομάζεται ECMA script και τώρα βρισκόμαστε στην έκτη έκδοσή της. Σύμφωνα με το IEEE Spectrum¹³ η JavaScript είναι όγδοη ανάμεσα στο Top 10 των γλωσσών προγραμματισμού ενώ σύμφωνα με το GitHub είναι η πρώτη σε χρήση γλώσσα στο GitHub. Η JavaScript χρησιμοποιείται για την αλληλεπίδραση της ιστοσελίδας με τον χρήστη και την δημιουργία δυναμικού περιεχομένου. Η τρέχουσα έκδοση είναι η ECMA 6 η οποία προσθέτει νέα χαρακτηριστικά όπως τα “Arrow Functions” (αντίστοιχα των Lambda expressions στην Java) απλοποίηση των κλάσεων κ.α.

2.2.3.1. JavaScript libraries/frameworks

Όπως με κάθε γλώσσα προγραμματισμού έτσι και στην Javascript υπάρχουν διαφορετικά frameworks τα οποία έχουν αναπτυχθεί προκειμένου να απλοποιήσουν την διαδικασία ανάπτυξης εφαρμογών και την συντήρησή τους «κωδικοποιώντας» συχνά χρησιμοποιούμενες λειτουργίες σε συναρτήσεις. Σε αυτή την ενότητα θα αναφέρουμε τα πιο γνώστα JS libraries και frameworks.

JQuery

Το JQuery είναι μια cross-platform JavaScript βιβλιοθήκη με σκοπό την απλοποίηση της διαδικασίας scripting στην πλευρά του τελικού χρήστη. Το JQuery είναι η πιο διαδεδομένη JS βιβλιοθήκη η οποία χρησιμοποιείται σε πάνω από το 69%¹⁴ των 10 εκατομμυρίων πιο συχνά

¹³ IEEE Spectrum: The 2015 top ten programming languages

¹⁴ Built With: JQuery Statistics

επισκέψιμων site στο διαδίκτυο. Το JQuery είναι μια λογισμικό ανοικτού κώδικά παρέχεται υπό τους όρους του MIT License¹⁵ και η ανάπτυξη του ξεκίνησε τον Αύγουστο του 2006 από τον John Resig. Η σύνταξη του JQuery έχει γίνει με τέτοιον τρόπο ώστε να είναι ευκολότερη η πλοήγηση μέσα σε ένα έγγραφο και η επιλογή στοιχείων του DOM. Μερικά από τα βασικά χαρακτηριστικά που προσφέρει το JQuery είναι:

- Εύκολη επιλογή των συστατικών του DOM, ανεξαρτήτου browser
- Τροποποίηση των ιδιοτήτων CSS των συστατικών βάσει κριτηρίου(Ετικέτα, κλάση όνομα)
- Γεγονότα, επιτρέπει την δημιουργία Event και Event Listener
- AJAX
- JSON parser
- Plugins

AngularJS

Το Angular JS είναι ένα cross-platform JavaScript framework ανοικτού κώδικα το οποίο αναπτύχθηκε από τον Misko Hevery υπάλληλο της Google και σήμερα συντηρείται από την Google και άλλες εταιρείες που συνεισφέρουν στην κοινότητα. Σκοπός του Angular ήταν να αντιμετωπιστούν οι προκλήσεις που αντιμετωπίζει ένας προγραμματιστής κατά την ανάπτυξη “Single Page” Εφαρμογών. Πρόκειται για ένα model-view-controller(MVC) framework το οποίο απλοποιεί την διαδικασία ανάπτυξης αλλά και testing των εφαρμογών. Το Angular αντιμετωπίζει τον HTML κώδικα ως στατικό κομμάτι και δουλεύει ως εξής: Το framework διαβάζει ολόκληρο το έγγραφο, στο έγγραφο μέσα υπάρχουν custom elements που χρησιμοποιεί το Angular. Στην συνέχεια μεταγλωττίζει τα elements αυτά σε JS εντολές για συνδέσει τις εισόδους και τις εξόδους της εφαρμογής σε συγκεκριμένα Μοντέλα τα οποία αντιπροσωπεύονται από JS μεταβλητές. Η ανάθεση των τιμών στις μεταβλητές μπορεί να γίνει απ ευθείας (var clientName = “John”;) ή μέσω ενός JSON είτε στατικά είτε δυναμικά.

¹⁵ OpenSource: MIT License

```

<!doctype html>
<html lang="en" ng-app="app" ng-controller="FormController as formCtrl">
<head>
  <meta charset="utf-8" />
  <title>Example of Angular Form Validation</title>
</head>
<body>

<form>
Enter your email:
<input type="email" required>
<button type="submit">Submit</button>
</form>

<script src=
"https://ajax.googleapis.com/ajax/libs/angularjs/1.3.15/angular.min.js">
</script>
<script>
angular.module('app', [])
.controller('FormController', function() {
});
</script>
</body>
</html>

```

Εικόνα 3 Παράδειγμα επαλήθευσης φόρμας με χρήση Angular JS

2.2.4. XML

Η XML (Extensible Markup Language) είναι μία γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Ορίζεται, κυρίως, στην προδιαγραφή XML 1.0 (XML 1.0 Specification), που δημιούργησε ο διεθνής οργανισμός προτύπων W3C. Η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο Διαδίκτυο. Είναι μία μορφοποίηση δεδομένων κειμένου, με ισχυρή υποστήριξη και η σχεδίαση της εστιάζει στα κείμενα, χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων, που προκύπτουν από WebServices. Υπάρχει μία ποικιλία διεπαφών προγραμματισμού εφαρμογών, που μπορούν να χρησιμοποιούν οι προγραμματιστές, για να προσπελαίνουν δεδομένα XML, αλλά και διάφορα συστήματα σχημάτων XML, τα οποία είναι σχεδιασμένα για να βοηθούν στον ορισμό γλωσσών, που προκύπτουν από την XML. Έως το 2009, έχουν αναπτυχθεί εκατοντάδες γλώσσες που βασίζονται στην XML όπως το RSS, το SOAP και της XHTML.

2.2.5. JSON

Το JSON (JavaScript Object Notation) είναι ένα ελαφρύ πρότυπο ανταλλαγής δεδομένων. Είναι εύκολο να διαβαστεί και να γραφεί. Είναι εύκολο για τα υπολογιστικά συστήματα να το αναλύσουν (parse) και να το παράγουν (generate). Βασίστηκε σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript, Standard ECMA-262 Έκδοση 3η - Δεκέμβριος 1999. Το JSON είναι ένα πρότυπο κειμένου το οποίο είναι τελείως ανεξάρτητο από γλώσσες προγραμματισμού αλλά χρησιμοποιεί πρακτικές (conventions) οι οποίες είναι γνωστές στους προγραμματιστές της οικογένειας προγραμματισμού C, συμπεριλαμβανομένων των C, C++, C#, Java, JavaScript, Perl, Python, και πολλών άλλων. Αυτές οι ιδιότητες κάνουν το JSON μια ιδανική γλώσσα προγραμματισμού ανταλλαγής δεδομένων. Το JSON μπορεί να υλοποιηθεί με δύο διαφορετικούς τρόπους. Ως μια συλλογή από ζευγάρια με ονομάτων και τιμών. Σε διάφορες γλώσσες προγραμματισμού, αυτό δηλώνεται ως ένα object με συγκεκριμένη δομή λεξικό και (hash table) ή λίστα κλειδιών ή associative πίνακα. Η μπορεί να υλοποιηθεί σαν ταξινομημένη λίστα τιμών. Στις περισσότερες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένας πίνακας (array), διάνυσμα, λίστα, ή ακολουθία. Λόγω της ευελιξίας των αντικειμένων JSON είναι ο προτεινόμενος τρόπος για την ανταλλαγή δεδομένων ανάμεσα σε Server-Client.



```
1  {
2    "data":
3    [
4      {
5        "id": 1,
6        "name": "Sequel Pro 0.8",
7        "version_string": "0.8",
8        "appcast_url": "http://www.sequelpro.com/appcast/app-releases.xml",
9        "build_no": 19,
10       "release_notes": "",
11       "download_link": "http://sequel-pro.googlecode.com/files/sequel-pro-0.8.dmg",
12       "release_type": "Stable",
13       "created": null,
14       "updated": 1296545735,
15       "release_date": 1207958400,
16       "archive": 0
17     },
18     {
19       "id": 2,
20       "name": "Sequel Pro 0.9",
21       "version_string": "0.9",
22       "appcast_url": "http://www.sequelpro.com/appcast/release_0.9.html",
23       "build_no": 30,
```

Εικόνα 4 Παράδειγμα αντικειμένου JSON

2.2.6. JWT

Το JWT (JSON Web Token) είναι ένα ανοιχτό πρότυπο (RFC 7519¹⁶) που ορίζει έναν ασφαλή τρόπο μετάδοσης πληροφοριών μέσω του διαδικτύου. Τα δεδομένα μπορούν να επαληθευθούν καθώς είναι ψηφιακά υπογεγραμμένα. Τα JWT μπορούν να υπογραφούν με την χρήση κωδικού HMAC ή με την χρήση προσωπικών/δημόσιων κλειδιών (RSA). Λόγω του μικρού τους μεγέθους μπορούν να σταλούν μέσω POST request ή να ενσωματωθεί σε HTTP header.

2.2.6.1. Χρήση των JWT

Τα JSON Web Tokens χρησιμοποιούνται κυρίως για δύο σκοπούς:

A) Authentication

Το βασικό πεδίο χρήσης των JWT είναι η επαλήθευση των χρηστών. Ο χρήστης πιστοποιείται σε ένα authentication σύστημα και του δίνεται ένα JWT. Κάθε ένα από τα επόμενα Request του χρήστη θα πρέπει να έχουν το Token. Η συγκεκριμένη πρακτική είναι αρκετά διαδεδομένη και επιτρέπει την δημιουργία Single Sign On υποδομών κυρίως λόγω του μικρού overhead που προσθέτουν τα Tokens αλλά και της ευκολίας χρήσης τους από συστήματα σε διαφορετικά domains.

B) Ανταλλαγή δεδομένων

Το δεύτερο πεδίο χρήσης των JWT είναι η ανταλλαγή πληροφοριών. Με την χρήση δημοσίων/ιδιωτικών κλειδιών τα συναλλασόμενα μέρη μπορούν να πιστοποιήσουν ότι ο αποστολέας είναι αυτός που ισχυρίζεται. Επιπλέον η υπογραφή υπολογίζεται χρησιμοποιώντας το header και το Payload διασφαλίζοντας έτσι ότι το περιεχόμενο του μηνύματος ότι δεν έχει αλλοιωθεί.

¹⁶ IETF: RFC 7519

2.2.7. Docker

Το Docker είναι ένα λογισμικό ανοιχτού κώδικα για την αυτοματοποίηση του deployment των εφαρμογών με χρήση software containers παρέχοντας έναν επίπεδο virtualization σε επίπεδο λειτουργικού συστήματος. Το Docker χρησιμοποιεί χαρακτηριστικά απομόνωσης πόρων (resource isolation) του Linux Kernel όπως τα cgroups και τα kernel namespaces. Τα namespace του Linux Kernel απομονώνει την εφαρμογή ώστε αυτή να έχει περιορισμένη πρόσβαση στο δέντρο διεργασιών, το δίκτυο, τα user ids αλλά και το filesystem. Μέσω των cgroups επιτυγχάνεται ο περιορισμός στους πόρους του συστήματος όπως την CPU, το μέγεθος της RAM, αλλά και τα I/O όπως δίσκους και δίκτυο. Σε αντίθεση με μια εικονική μηχανή δεν χρειάζεται ένα δικό του λειτουργικό σύστημα με αποτέλεσμα να είναι ελαφρύ και να καταναλώνει λιγότερους πόρους.

3. Ανάπτυξη Mobile Εφαρμογών

3.1. Εισαγωγή

Η ανάπτυξη εφαρμογών για κινητές συσκευές όπως και η ανάπτυξη εφαρμογών για ηλεκτρονικούς υπολογιστές σημαίνει την ανάπτυξη κώδικα σε συγκεκριμένη γλώσσα προγραμματισμού για συγκεκριμένο υπολογιστικό σύστημα. Στους ηλεκτρονικούς υπολογιστές το πρόβλημα δεν είναι τόσο έντονο, με τα Microsoft Windows να έχουν σχεδόν το 90%¹⁷ του μεριδίου αγοράς η ανάπτυξη εφαρμογών γίνεται πιο εύκολη. Στις κινητές συσκευές η κατάσταση δεν είναι ίδια. Ενώ το Android έχει τα ινία της αγοράς με ποσοστό 82% το app store του IOS έχει μεγαλύτερο τζίρο¹⁸ κάνοντάς το ιδιαίτερα ελκυστικό. Η δημιουργία λοιπόν μια εφαρμογής για κινητή συσκευή η οποία να λειτουργεί τουλάχιστον στα δύο βασικά λειτουργικά συστήματα απαιτεί την δημιουργία δυο διαφορετικών Native Apps. Αυτό μεταφράζεται ότι ο σχεδιασμός ή ανάπτυξη ο έλεγχος γίνεται δυο φορές. Το Android project θα πρέπει να αναπτυχθεί σε Java ενώ το iOS σε Swift (Objective C). Ο προγραμματιστής θα πρέπει να αποκτήσει την κατάλληλη γνώση πάνω στις δύο πλατφόρμες προκειμένου να δημιουργήσει ποιοτικές εφαρμογές. Ένα ακόμη μεγάλο πρόβλημα που υπάρχει στις κινητές συσκευές είναι το πρόβλημα του κατακερματισμού.

3.2. Το Πρόβλημα του Κατακερματισμού

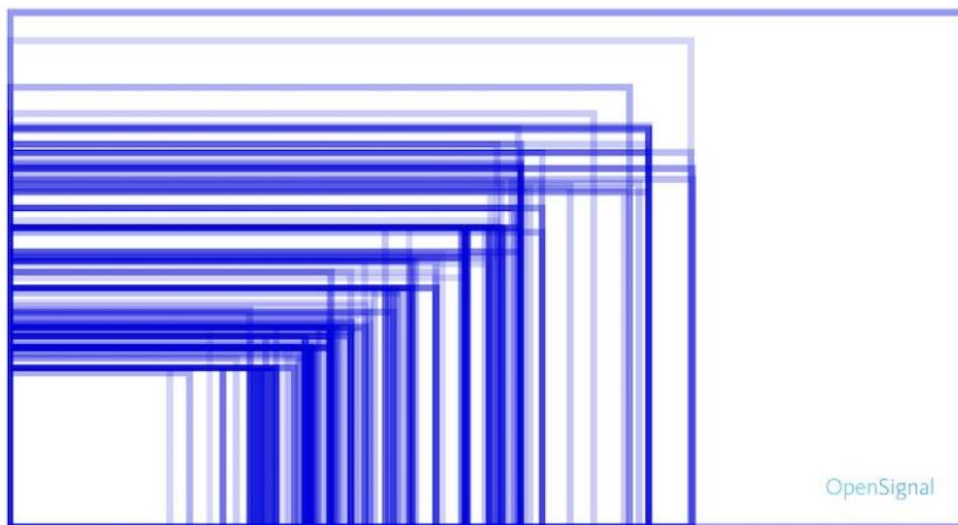
Οι γλώσσες προγραμματισμού είναι γλώσσες. Σαν την φυσική γλώσσα έχουν συντακτικούς και γραμματικούς κανόνες και το λεξιλόγιο συνεχώς εμπλουτίζεται. Νέα χαρακτηριστικά προστίθενται και άλλα θεωρούνται παρωχημένα (deprecated) και ο προγραμματιστής πρέπει πάντα να φροντίζει η εφαρμογή του να λειτουργεί παντού. Μπορεί στις εφαρμογές που γράφονται για ηλεκτρονικούς υπολογιστές τα πράγματα να είναι πιο απλά και η λειτουργικότητα της εφαρμογής να διασφαλίζεται με την εγκατάσταση των αντίστοιχων βιβλιοθηκών στις κινητές συσκευές δεν συμβαίνει το ίδιο. Εδώ θα πρέπει να

¹⁷ MarketShare: Desktop Operating System Market Share

¹⁸ 9to5Mac iOS App Store brings in 75% more revenue than Playstore

προσέξουμε την έκδοση του λειτουργικού συστήματος τα τεχνικά χαρακτηριστικά της συσκευής και πιο ειδικά την ανάλυση της οθόνης. Σύμφωνα με το statcounter.com(ιστοσελίδα παροχής στατιστικών) οι επικρατούσες αναλύσεις οθόνης για την περίοδο Μάρτιος 2015 – Φεβρουάριος 2016 είναι έξι. Στην εικόνα 5 φαίνονται οι διαφορετικές αναλύσεις οθόνης που χρησιμοποιούνται στα 24.093 διαφορετικά μοντέλα κινητών με λογισμικό Android τον Αύγουστο του 2015¹⁹.

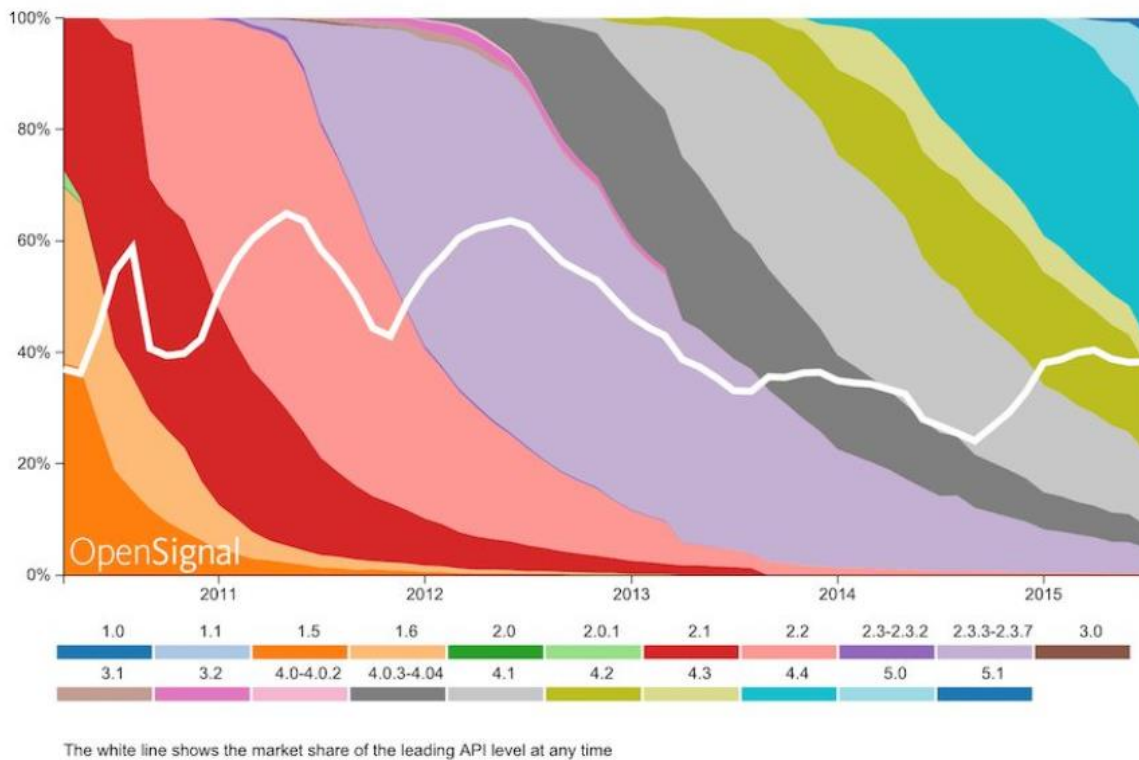
SCREEN SIZE FRAGMENTATION



Εικόνα 7:Android screen size fragmentation

¹⁹ OpenSignal: Android Fragmentation Visualized 08.2015

ANDROID OPERATING SYSTEM FRAGMENTATION



Εικόνα 8 Χρησιμοποιούμενες εκδόσεις Android

Αντίστοιχο κατακερματισμό έχουμε και στην έκδοση του Android που χρησιμοποιούν οι συσκευές.

Το πρόβλημα του κατακερματισμού δεν είναι αποκλειστικό φαινόμενο του Android, αλλά υπάρχει στο IOS²⁰ και στα Windows Phone (πλέον Windows)²¹. Γίνεται εύκολα αντιληπτό ότι δημιουργείται μια υποχρέωση στον προγραμματιστή για εκτενέστερο έλεγχο συμβατότητας της εφαρμογής και προσεκτικότερο σχεδιασμό.

²⁰ Zdnet: IOS 7 fragmentation issues 02.2015

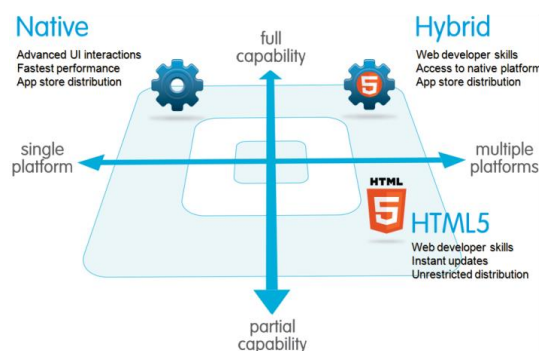
²¹ PC Magazine Microsoft After Smartphone Fragmentation

3.3. Υβριδικές Εφαρμογές

Λύση στο προηγούμενο πρόβλημα έρχεται να δώσει η ανάπτυξη υβριδικών εφαρμογών. Όπως αναφέραμε στο προηγούμενο κεφάλαιο οι υβριδικές εφαρμογές κάνουν χρήση διάφορων τεχνολογιών web όπως (HTML 5, CSS, JavaScript) σε συνδυασμό με κομμάτια native κώδικα ώστε να δημιουργήσουν μια cross platform εφαρμογή.

3.3.1. Παρούσα κατάσταση

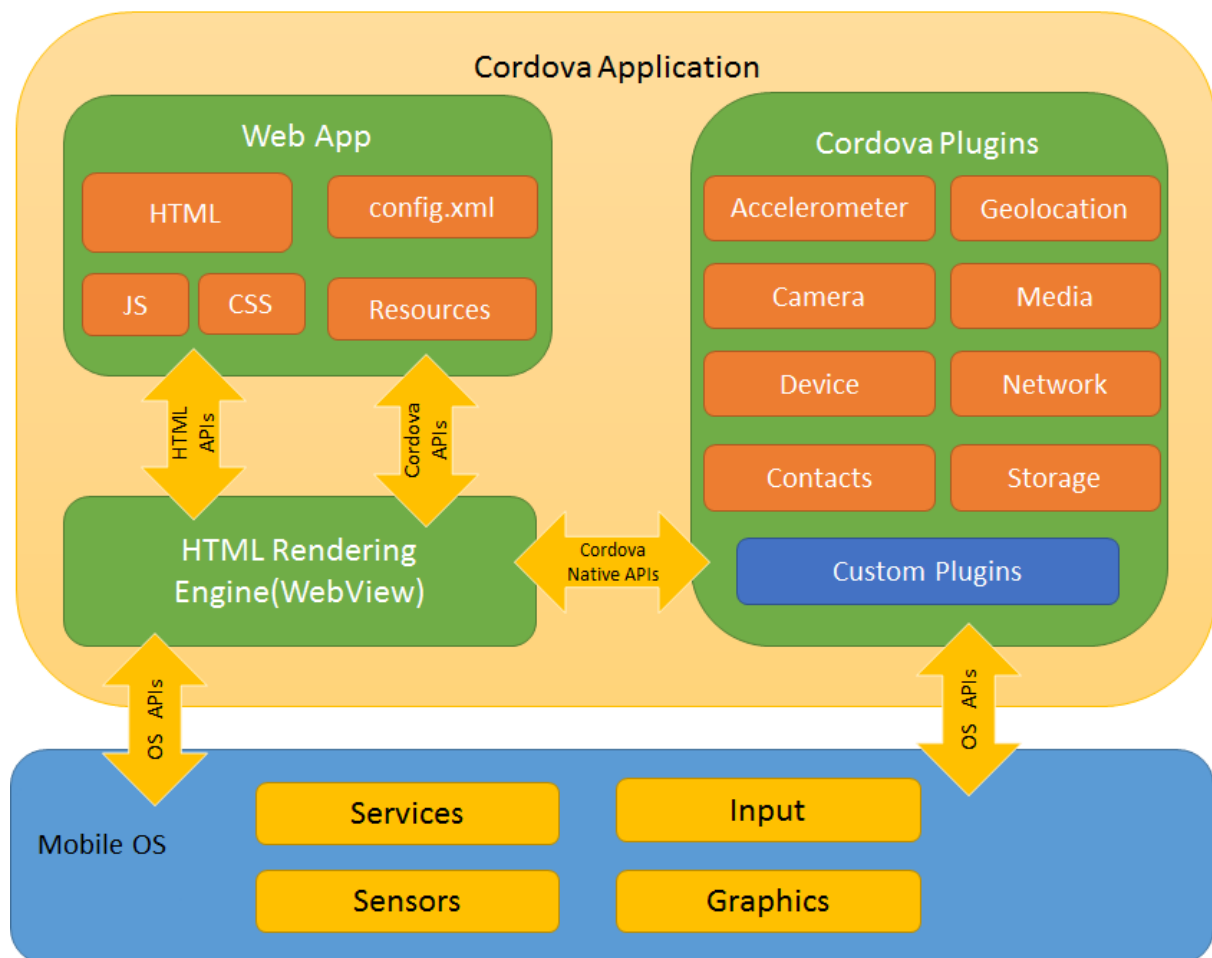
Ολοένα και περισσότερες εταιρείες αναθεωρούν την τακτικής τους ως προς το Mobile development και πολλές από τις native εφαρμογές που έχουν αναπτύξει τις μετατρέπουν σε web-applications ή σε υβριδικές εφαρμογές. Σκοπός της αλλαγής είναι η εξοικονόμηση χρόνου από την ανάπτυξη των εφαρμογών και σαφώς για την προσέλκυση περισσότερων χρηστών από νέες πλατφόρμες. Παράδειγμα τέτοιων εταιρειών που εγκατέλειψαν τις native εφαρμογές και τις αντικατέστησαν είναι το: Facebook, το Twitter, το Evernote, το Gmail, και το AppStore!



Εικόνα 9SalesForce: Hybrid vs Native mobile apps

3.3.2. Ανάπτυξη υβριδικών εφαρμογών

Όπως ήδη αναφέραμε οι υβριδικές εφαρμογές κάνουν χρήση τεχνολογιών web όπως η HTML 5 και η JavaScript, πως όμως λειτουργεί μια υβριδική εφαρμογή; Κάθε ένα από τα frameworks που υπάρχουν για την ανάπτυξη υβριδικών εφαρμογών χρησιμοποιεί δικά του εργαλεία ό κυρίως τρόπος λειτουργίας όμως είναι παντού ο ίδιος. Στο επόμενο σχήμα φαίνεται ο τρόπος που χρησιμοποιεί το Apache Cordova για την λειτουργία μιας Hybrid εφαρμογής.



Εικόνα 10 Αρχιτεκτονική Apache Cordova

Η εφαρμογή “Cordova Application” εγκαθίστατε στην συσκευή και αλληλοεπιδρά με το λειτουργικό σύστημα μέσω των παρεχόμενων APIs. Η εφαρμογή αποτελείται από τρία συστατικά μέρη, την κύρια εφαρμογή, τον μηχανισμό απεικόνισης HTML και τα plugins.

3.3.2.1. Web App

Είναι το σημείο όπου εφαρμόζεται το Business Logic και ζει ο κώδικας της εφαρμογής. Η εφαρμογή αναπτύσσεται σαν ιστοσελίδα και τρέχει από το index.html. Το αρχείο αυτό περιέχει τις JavaScript βιβλιοθήκες που απαιτούνται καθώς και τα αντίστοιχα CSS που καθορίζουν το “Look and Feel” της εφαρμογής. Μέσα στο container της εφαρμογής βρίσκεται και το αρχείο παραμετροποίησης “config.xml”. Πρόκειται για το αρχείο ρυθμίσεων της εφαρμογής, αυτό καθορίζει πως συμπεριφέρεται η εφαρμογή(πως αντιδρά στην αλλαγή του προσανατολισμού της οθόνης). Η εφαρμογή εκτελείται από το container(apk στο Android) μέσα σε ένα WebView.

3.3.2.2. WebView

Το WebView ή αλλιώς HTML Rendering Engine είναι το συστατικό του λειτουργικού συστήματος της συσκευής το οποίο επιτρέπει στις εφαρμογές να προβάλουν περιεχόμενο ιστού. Το στοιχείο αυτό βρίσκεται προ εγκατεστημένο στην συσκευή. Στο IOS και στο Android²² το rendering engine που χρησιμοποιείται είναι το KDE WebKit²³. Στο Apache Cordova ο προγραμματιστής έχει τη δυνατότητα να δημιουργήσει δικό του WebView χρησιμοποιώντας εγγενή χαρακτηριστικά του λειτουργικού. Το WebView παίζει καθοριστικό ρόλο στον τρόπο με τον οποίο απεικονίζεται η εφαρμογή μας στην οθόνη του χρήστη.

3.3.2.3. Plugins

Τα plugins αποτελούν δομικά στοιχεία του οικοσυστήματος. Παρέχουν διεπαφή ανάμεσα στο Cordova και τα native συστήματα του λειτουργικού ενώ παρέχει μεθόδους για την χρήση των APIs της συσκευής. Το Cordova παρέχει 21 βασικά plugins (Core Plugins) τα οποία παρέχουν πρόσβαση στο hardware της συσκευής(δίκτυο, κάμερα, τοποθεσία, δόνηση), βασικές λειτουργίες(αποθηκευτικός χώρος, επαφές). Εκτός των Core Plugins το Cordova

²² Google Developers: WebView

²³ Wikipedia: WebKit

επιτρέπει την χρήση user plugins τα οποία μπορούν να επεκτείνουν την χρήση της εφαρμογής και να προσθέσουν νέα χαρακτηριστικά όπως την σύνδεση με υπηρεσίες file sharing, υπηρεσίες διαφήμισης, ή ακόμη και κάποια νέα μέθοδο για την αλληλεπίδραση με τον χρήστη. Τα Plugins δεν πρέπει να τα μπερδεύουμε με τα γραφικά στοιχεία του περιβάλλοντος χρήστη(UI) ή με στοιχεία MVC. Οι λειτουργίες αυτές παρέχονται μπορούν να προστεθούν χρησιμοποιώντας τα κατάλληλα Frameworks όπως το Ionic το Bootstrap το AppFramework κ.ο.κ. Η χρήση UI Framework αποτελεί επιλογή του developer.

3.3.3. Υβριδικές εφαρμογές και κατακερματισμός

Όπως έχουμε ήδη αναφέρει, βάση των υβριδικών εφαρμογών είναι η HTML 5 όπου σε συνδυασμό με την CSS 3 έχουμε την ευελιξία να ελαχιστοποιήσουμε - αν όχι να εξαλείψουμε - τα προβλήματα που μας δημιουργεί ο κατακερματισμός. Με χρήση Media queries μπορούμε να ορίσουμε διαφορετικούς κανόνες μορφοποίησης ανάλογα με τα χαρακτηριστικά της οθόνης μας(ανάλυση, πυκνότητα pixel κ.ο.κ.) Τα Media queries είναι αρκετά ευέλικτα και μπορούν να χρησιμοποιηθούν για τη μορφοποίηση μόνο των διαστάσεων ελαχιστοποιώντας έτσι τα κομμάτια που πρέπει να γραφούν ξανά. Ένας άλλος τρόπος για την επίλυση του προβλήματος είναι να χρησιμοποιήσουμε κάποιο frontend framework όπως το bootstrap. Τα framework αυτά παρέχουν «Fluid-Design» προσαρμόζονται δηλαδή στην οθόνη απαλλάσσοντάς μας έτσι από την υποχρέωση να γράφουμε περιττό κώδικα. Τα πιο διαδεδομένα framework για την ανάπτυξη υβριδικών εφαρμογών είναι τα: Ionic, App Framework, JQuery Mobile, Ratchet, Bootstrap.

3.3.4. Hybrid development frameworks

3.3.4.1. Apache Cordova

Το Cordova είναι από τα πιο διαδεδομένα Frameworks ανάπτυξης υβριδικών εφαρμογών. Αναπτύχθηκε από την Nitobi η οποία εξαγοράστηκε από την Adobe Systems το 2011 φτιάχνοντας το PhoneGap και δωρίζοντας²⁴ ταυτόχρονα τον πηγαίο κώδικα στο Apache Software Foundation υπό την ονομασία Apache Cordova. Το Cordova χρησιμοποιεί HTML 5 CSS3 και επεκτείνει τις δυνατότητες της γλώσσας μέσω των plugins όπως περιγράψαμε πιο πάνω. Το Cordova αποτελεί την βάση και για άλλα διαδεδομένα frameworks όπως το Adobe PhoneGap, το Intel XDK ή το Monaca, Το Cordova παράγει εφαρμογές που λειτουργούν σε Android, BlackBerry, Firefox OS, Ubuntu touch, webOS, Windows Phone, Windows 8



Εικόνα 11 Apache Cordova

3.3.4.2. Crosswalk Project

Το Crosswalk Project είναι μια πλατφόρμα ανοιχτού κώδικα το οποίο αναπτύχθηκε από την Intel. Χρησιμοποιεί ως βάση του τον Chromium. Σε αντίθεση με το Cordova το Crosswalk χρησιμοποιεί δοκιμαστικές τεχνολογίες για την δημιουργία μοναδικών χαρακτηριστικών. Χαρακτηριστικό παράδειγμα είναι η αξιοποίηση της Intel RealSense μια τεχνολογίας που χρησιμοποιεί τρεις κάμερες με σκοπό την αντίληψη του «βάθους πεδίου» από την συσκευή για την αλληλεπίδραση με τον χρήστη. Υποστηρίζει επίσης το WebGL μια υλοποίηση του OpenCL σε JavaScript για την ετερογενή παράλληλη επεξεργασία δεδομένων μέσα από το WebView.

²⁴ Adobe PhoneGap: PhoneGap, Cordova, and what's in a name?

4. Ανάπτυξη εφαρμογής: OneApp

Στο πλαίσιο της μεταπτυχιακής μου εργασίας αποφάσισα να φτιάξω μια εφαρμογή για κινητές συσκευές. Λόγω αντικειμένου εργασίας ανέπτυξα μια εφαρμογή για την παρακολούθηση ασφαλιστικού χαρτοφυλακίου για τους ασφαλιστικούς συμβούλους της εταιρείας (Smart Insurance Agency). Σκοπός μου με την εφαρμογή ήταν να δώσω τα κατάλληλα εργαλεία στους συνεργάτες ώστε να έχουν πρόσβαση στα δεδομένα που είναι κρίσιμα για την δουλειά χωρίς να είναι απαραίτητη η παρουσία τους στα γραφεία της εταιρείας. Όλες οι πληροφορίες σχετικά με το πελατολόγιο τα συμβόλαια τις οικονομικές οφειλές είναι διαθέσιμα στους συνεργάτες μέσω web εφαρμογής η οποία όμως δεν είναι mobile friendly υποχρεώνοντας τον χρήστη να κάνει χρήση ηλεκτρονικού υπολογιστή. Τα δεδομένα αποθηκεύονται στην βάση δεδομένων του ERP η οποία ήταν και το μοναδικό κομμάτι τρίτου κατασκευαστή το οποίο χρησιμοποίησα και το αντιμετώπισα ως “Black Box”. Η εφαρμογή αυτή αρχικά αναπτύχθηκε ως Native application για Android λόγω ύπαρξης όμως αρκετών χρηστών iPhone και κάποιων λιγότερων με Windows Phone δημιούργησε την υποχρέωση ανάπτυξης αντίστοιχων native εφαρμογών για καθένα από τα λειτουργικά συστήματα. Δεδομένου της μέτριας εμπειρίας μου στον προγραμματισμό Windows Phone και την μηδενική σε IOS κατέφυγα στην λύση μιας cross-platform λύσης. Εκτός από την Mobile εφαρμογή χρειάστηκε να φτιάξω μερικά Web Services για την άντληση δεδομένων από την βάση δεδομένων του πληροφοριακού συστήματος.

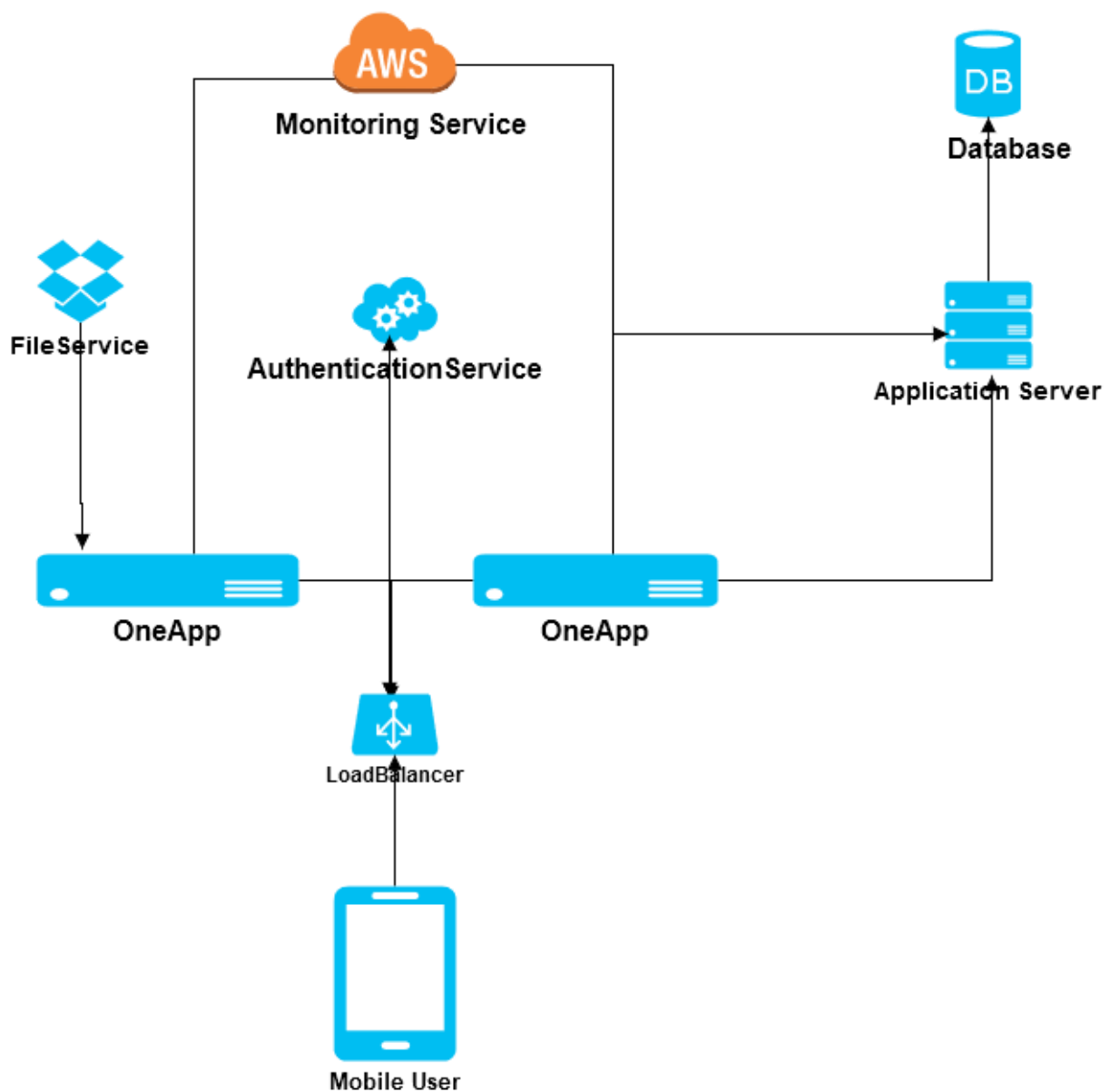


Εικόνα 12 Το λογότυπο της εφαρμογής²⁵, Σχεδιασμός Καπούτοσου Αννέζα

²⁵ Το λογότυπο της εφαρμογής διανέμεται υπό τους Όρους της creative commons [Αναφορά Δημιουργού-Μη Εμπορική Χρήση-Όχι Παράγωγα Έργα 4.0 Διεθνές](#)

4.1. Αρχιτεκτονική

Στην παρούσα ενότητα θα αναλύσω τον τρόπο λειτουργίας της εφαρμογής τα συστατικά της μέρη καθώς και τις τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της. Η εφαρμογή (OneApp) είναι μέρος ενός μεγαλύτερου πληροφοριακού συστήματος και αποτελεί το σημείο αλληλεπίδρασης τους χρήστη με το Π.Σ. Στο επόμενο διάγραμμα παρουσιάζεται η αρχιτεκτονική που χρησιμοποιήθηκε.



Εικόνα 13 Αναπαράσταση της αρχιτεκτονικής OneApp

Αρχή για την δόμηση του οικοσυστήματος της εφαρμογής ήταν η βάση δεδομένων (DB) η οποία όπως ανέφερα ήταν τρίτου κατασκευαστή. Η βάση δεδομένων αποτέλεσε το μεγαλύτερο πρόβλημα από πλευράς performance της εφαρμογής καθώς το μηχανογραφικό σύστημα χρησιμοποιεί μια αρκετά παλιά έκδοση της βάσης δεδομένων SYSDATABASE Advantage SQL. Λόγω αυτού έχουν γίνει μια σειρά από παραμετροποιήσεις προκειμένου να βελτιωθεί η απόδοση των WebService. Μπροστά από τους application servers της εφαρμογής βρίσκεται ένας Nginx Load Balancer ο οποίος αναθέτει τα Requests βάση του μοντέλου “Least Connections”²⁶ (Ο Server με τα λιγότερα connections). Οι Applications servers είναι cloud hosted VPS(Virtual Private Servers) χρησιμοποιούν NGINX με το php-fpm module για την χρήση της PHP 7. Τα webservices έχουν αναπτυχθεί σαν docker images. Η λειτουργία του συστήματος παρακολουθείται από ξεχωριστό server (EC2) ενώ η διαχείριση και το deployment γίνεται με την βοήθεια του Ansible. Η ανταλλαγή δεδομένων ανάμεσα στην εφαρμογή του χρήστη και τους servers γίνεται με χρήση JSON. Η ταυτοποίηση των χρηστών γίνεται μέσω του αντίστοιχου Authentication web service ενώ μετά την ταυτοποίηση του χρήστη η πιστοποίηση των στοιχείων κατά την κλήση των web services γίνεται με χρήση JWT. Μέσω της εφαρμογής ο χρήστης μπορεί να κάνει λήψη ψηφιακά έγγραφα του πελάτη (π.χ αιτήσεις) τα οποία ανακτώνται μέσω των Dropbox Web Service.

²⁶ NGINX: What is load balancing

4.2. Ανάπτυξη εφαρμογής

Όπως ήδη αναφέραμε στην εισαγωγή του κεφαλαίου αναπτύχθηκε αρχικά ως Native εφαρμογή για Android με την ονομασία “project uno”. Λόγω όμως του αυξημένου αριθμού των χρηστών iOS κρίθηκε απαραίτητη η δημιουργία μιας cross platform λύσης. Η ανάπτυξη της εφαρμογής έγινε με χρήση του Apache Cordova και ως εργαλείο ανάπτυξης το Intel XDK.



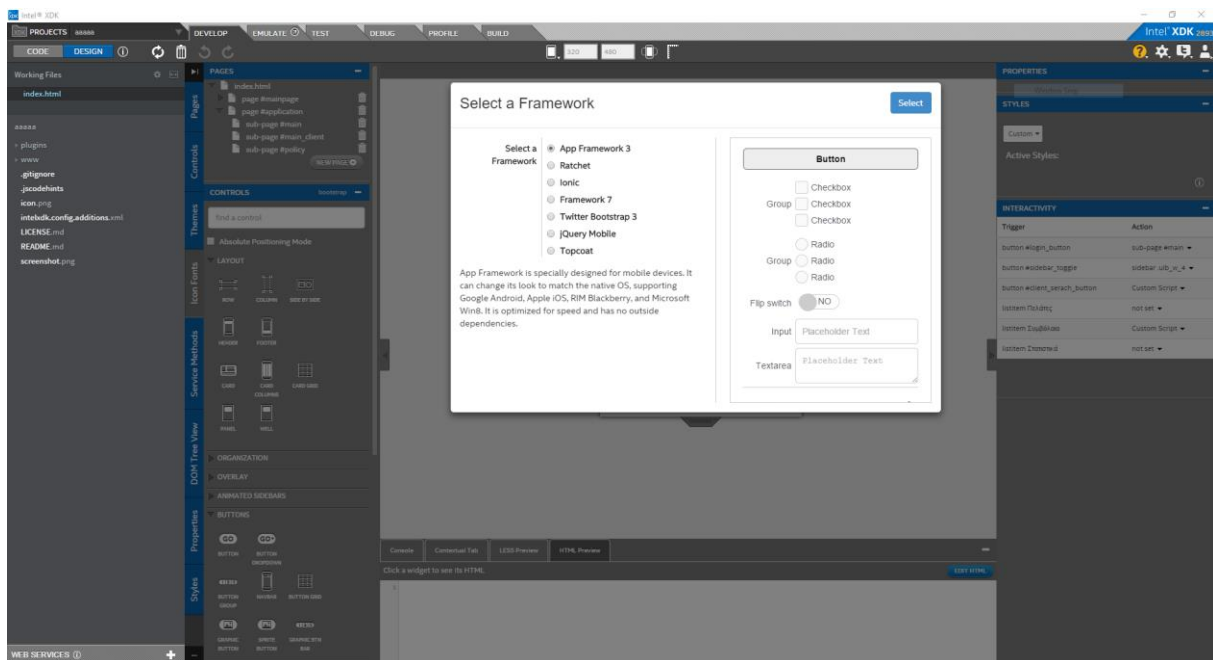
Εικόνα 14 Project uno: αρχική οθόνη εισαγωγής.

Η διαδικασία λήψης του Intel XDK²⁷ είναι αρκετά απλή και ο οδηγός εγκατάστασης της εφαρμογής κατευθύνει τον χρήστη καθ' όλη την διάρκεια. Ο λόγος που προτίμησα το Intel XDK έναντι άλλων προγραμμάτων (Eclipse, Visual Studio) ήταν το μικρό του μέγεθος η πληθώρα των προσφερόμενων χαρακτηριστικών και το φιλικό περιβάλλον χρήσης.

²⁷ Intel Developer zone: software.intel.com/intel-xdk

4.2.1. Δημιουργία νέου Project

Κατά την πρώτη εκκίνηση του Intel XDK θα κληθούμε να δημιουργήσουμε ένα νέο project. Μπορούμε να το αναπτύξουμε ως Standard HTML project -pure HTML- ή να δημιουργήσουμε ένα Apache Cordova project. Στην δεύτερη περίπτωση το XDK θα φορτώσει τις απαραίτητες βιβλιοθήκες για εμάς. Εκτός από κενό Project μπορούμε να κάνουμε χρήση κάποιου δημοφιλούς template γλιτώνοντας έτσι χρόνο από την σχεδίαση του Layout. Ένα από τα χαρακτηριστικά XDK είναι το “app designer”. Πρόκειται για ένα drag n drop interface που μας επιτρέπει να δημιουργήσουμε το Layout της εφαρμογής με απλά κλικ. Το app designer είναι καλά σχεδιασμένο και αποφεύγει την εισαγωγή περιττού κώδικα(όσο είναι δυνατόν). Βέβαια όπως με κάθε code-generator tool πάντα θα υπάρχει περιττός κώδικας και καλό θα ήταν να τον αποφύγουμε εφόσον μπορούμε.



Εικόνα 15 Intel xdk: δημιουργία νέου project

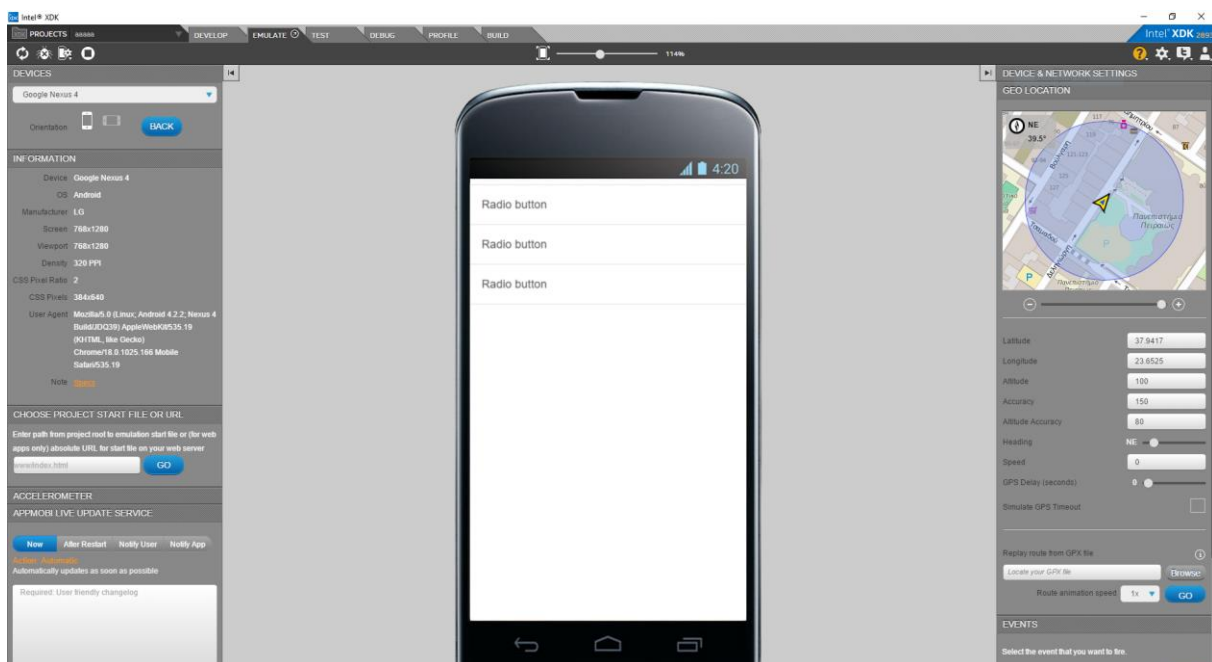
Μετά την επιλογή του μοντέλου το XDK θα μας ζητήσει να επιλέξουμε το Framework με το οποίο θα σχεδιάσουμε την εφαρμογή μας. Η intel έχει προνοήσει και στο IDE της υποστηρίζει επτά από τα πιο δημοφιλή fronted frameworks. Μέσα από το “Contextual tab” (Εικόνα 15 κάτω από την μέση) του app designer μπορούμε να διαβάσουμε το documentation για το συγκεκριμένο element στο επιλεγμένο framework.

4.2.2. App designer

Το «App designer» είναι το GUI builder του XDK και βρίσκεται στο “Develop” tab. Η επιφάνεια εργασίας χωρίζεται σε τέσσερις ενότητες. Η πρώτη είναι ο file explorer, από εκεί μπορούμε να δούμε τα αρχεία της εφαρμογής μας ενώ σε ξεχωριστό σημείο μπορούμε να δούμε τα Working Files. Πρόκειται για τα αρχεία που έχουμε ανοίξει διευκολύνοντας μας στην επεξεργασία τους. Στην πρώτη στήλη βρίσκονται και τα κουμπιά εναλλαγής από τον AppDesigner στο Brackets. Το Brackets είναι ο επεξεργαστής κειμένου με τον οποίο γράφουμε τον κώδικα της εφαρμογής μας. Παρέχει μια πληθώρα χαρακτηριστικών όπως auto-indent code-completion (κ.α.) ενώ μπορούμε να επεκτείνουμε τις δυνατότητές του μέσω PlugIns και να προσθέσουμε χαρακτηριστικά όπως git-client και code validator(JS-Lint). Η δεύτερη στήλη του AppDesigner μας δίνει πρόσβαση στα δομικά στοιχεία του κώδικα. Από εκεί μπορούμε να προσθέσουμε html entities όπως φόρμες, κουμπιά, sidebar (κ.τ.λ) αλλά και νέα pages που θα χρησιμοποιεί η εφαρμογή μας. Κατά την ανάπτυξη υβριδικών εφαρμογών θα πρέπει πάντα να θυμόμαστε ότι η εφαρμογή μας είναι “Single Page” web apps που σημαίνει ότι η δομή της θα πρέπει να έχει μια λογική συνέχεια ώστε να απλοποιηθεί όσο το δυνατόν η συντήρησή της. Η τρίτη στήλη αποτελεί το κυρίως μέρος του App Designer εκεί προσθέτουμε τα συστατικά μέρη της εφαρμογής. Τέλος στην τελευταία ενότητα καθορίζουμε του CSS κανόνες για το τρέχον στοιχείο και τα events της εφαρμογής. Αξίζει να σημειωθεί ότι θα πρέπει να αποφεύγουμε τις χειροκίνητες αλλαγές στα CSS αρχεία που παράγει το Intel XDK, καθώς στο πρώτο Compile θα χαθούν. Αντί αυτού μπορούμε να δημιουργήσουμε ένα νέο CSS αρχείο και να προσθέσουμε τους δικούς μας κανόνες ή να πρόκειται για αλλαγή στο ίδιο το framework μπορούμε να τροποποιήσουμε τα less αρχεία.

4.3. Application Emulation

Ένα από τα πιο σημαντικά στάδια στην ανάπτυξη εφαρμογών είναι η προσομοίωση. Ο προσομοιωτής αναλαμβάνει να «παίξει» τον ρόλο μια συσκευής ώστε να μπορέσει ο προγραμματιστής να δοκιμάσει την εφαρμογή υπό διαφορετικές συνθήκες. Στο Intel XDK η λειτουργία αυτή είναι διαθέσιμη μέσω του «Emulate» tab. Ο emulator του XDK μας δίνει την δυνατότητα να εξομοιώσουμε την εφαρμογή μας στο περιβάλλον δεκαεννέα συσκευών. Οι συσκευές που είναι διαθέσιμες κάνουν χρήση διαφορετικών εκδόσεων του Android, iOS, Windows Phone ενώ είναι επιλεγμένες έτσι ώστε να υπάρχει και πληθώρα διαφορετικών χαρακτηριστικών. Στην αριστερή στήλη του Emulator μπορούμε να βρούμε βασικές πληροφορίες για την συσκευή όπως: την έκδοση του λειτουργικού, την διαθέσιμη RAM, την ανάλυση της οθόνης την έκδοση του user agent κ.α.



Εικόνα 16 Το περιβάλλον εξομοίωσης του Intel XDK

Κατά την εξομοίωση της εφαρμογής μπορούμε να καθορίσουμε διάφορες παραμέτρους όπως την τρέχουσα θέση της εφαρμογής, το σφάλμα του GPS, δεδομένα για τον αισθητήρα επιτάχυνσης, την κατάσταση του δικτύου, τον τύπο σύνδεσης, και άλλα δεδομένα όπως την στάθμη της μπαταρίας κτλ. Μέσα στον emulator ενσωματώνονται και λειτουργίες debugger. Ο debugger που χρησιμοποιεί το Intel XDK είναι ίδιος με αυτόν του Google Chrome. Μέσα από εκεί μπορούμε να κάνουμε αποσφαλμάτωση της εφαρμογής μας, να ορίσουμε brake

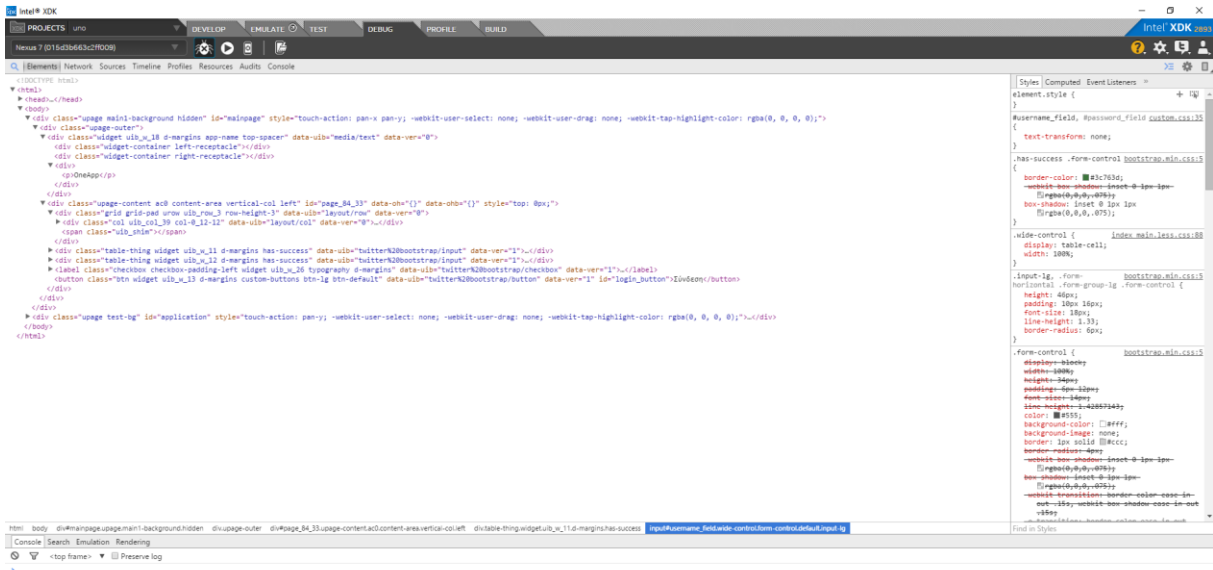
points στον εκτέλεση του κώδικα, να δούμε τα περιεχόμενα των μεταβλητών μας, να βρούμε πληροφορίες για την χρήση δεδομένων κ.ο.κ

4.4. Application Debugging

Ένα από τα πιο κρίσιμα στάδια στην ανάπτυξης εφαρμογών είναι η διαδικασία αποσφαλμάτωσης του κώδικα. Αναφερθήκαμε νωρίτερα στην ενσωματωμένο debugger του Emulator. Η χρήση μόνο του emulator δεν είναι αρκετή καθώς ο εξομοιωτής χρησιμοποιεί ένα αρκετά ελεύθερο περιβάλλον παρουσιάζοντας μας μια ουτοπική κατάσταση της εφαρμογής μας. Στο περιβάλλον του προσομοιωτή οι δικλίδες ασφαλείας όπως οι λίστες που καθορίζουν σε ποιους διαδικτυακούς πόρους έχει πρόσβαση η εφαρμογή μας ή οι μηχανισμοί δικαιωμάτων είναι απενεργοποιημένη. Με τον τρόπο αυτό μπορεί να έχουμε εσφαλμένη εντύπωση για την λειτουργία της εφαρμογής μας και να μην με τον τρόπο τον οποίο την σχεδιάσαμε. Για τους λόγους αυτούς πριν από την διάθεσή της εφαρμογής μας θα πρέπει να χρησιμοποιούμε τον debugger. Στο Intel XDK ο Debugger είναι διαθέσιμος μέσω του αντίστοιχου tab (Debug) ενώ η χρήση του απαιτεί την σύνδεση²⁸ τουλάχιστον μιας συσκευής (Android²⁹ ή iOS) για να λειτουργήσει. Την πρώτη φορά που θα χρησιμοποιήσουμε τον debugger καθώς και κάθε φορά που προσθέτουμε modules μέσω του project manager το Intel XDK δημιουργεί ένα debug module και το εγκαθιστά στην συσκευή. Κατόπιν η εφαρμογή εκτελείται στην συσκευή και μέσα από τον debugger έχουμε πλήρη πρόσβαση στα δεδομένα της συσκευής.

²⁸ Λόγω ενός bug στο Intel XDK (υπάρχει και στην έκδοση 2893) η σύνδεση συσκευών σε θύρες USB 3 ενδέχεται να δημιουργήσει προβλήματα στην λειτουργία του Debugger

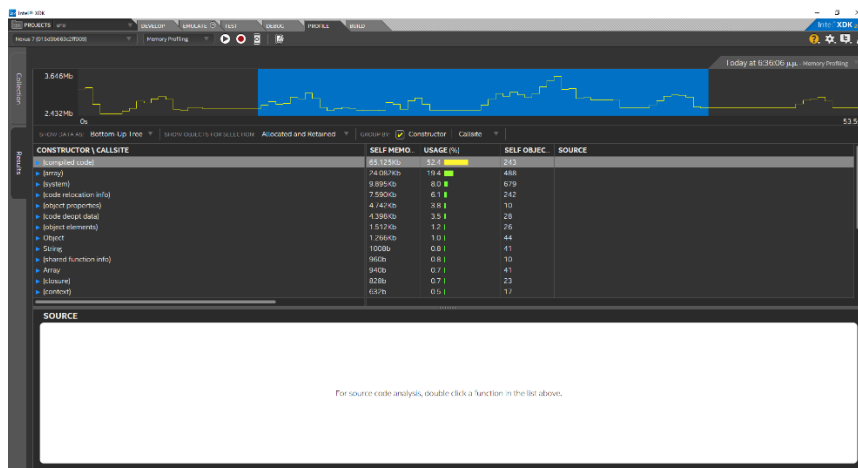
²⁹ Οδηγίες για την ενεργοποίηση του debugger σε android <http://developer.android.com/tools/device.html>



Εικόνα 17 Intel XDK Debugging Session

4.5. Profiler

Ο Profiler είναι το βασικό εργαλείο στην ανάλυση της συμπεριφοράς της εφαρμογής μας. Μας βοηθάει να βρούμε πιθανά προβλήματα στον κώδικα και μας δημιουργεί ένα προφίλ χρήσης των πόρων του συστήματος. Στην εικόνα 18 απεικονίζεται παράδειγμα χρήσης του Profiler μέσα από το Intel XDK.

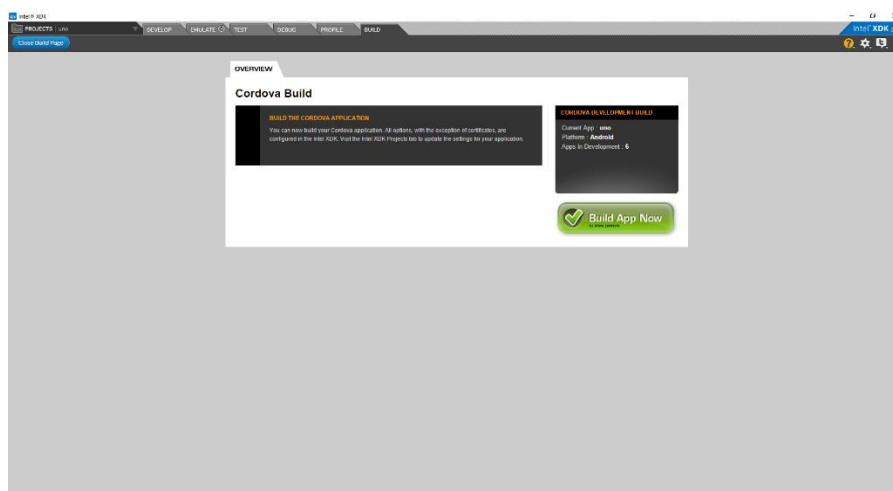


Εικόνα 18 Intel XDK profiler

Η λειτουργία του profiler είναι ιδιαίτερα απλή. Η εφαρμογή εκτελείται στην κινητή συσκευή και εμείς προσομοιώνουμε τον χρήστη. Ταυτόχρονα ο profiler συλλέγει δεδομένα από την συσκευή μας και μετά την ολοκλήρωση της συνόδου δημιουργεί ένα report. Ο Profiler στο XDK μπορεί να συλλέξει δεδομένα που αφορούν γενικότερους πόρους του συστήματος, δεδομένα χρήσης του επεξεργαστή και της RAM. Στην εικόνα 18 φαίνονται τα αποτελέσματα του profiling με δεδομένα χρήσης για την μνήμη RAM. Στο παράδειγμα ή μέγιστη χρήση μνήμης έφτασε τα 3,64MB ενώ στην μεσαία γραμμή βλέπουμε ομαδοποιημένους τους πόρους και το ποσοστό μνήμης που καταλαμβάνουν. Επεκτείνοντας τις ομάδες μπορούμε δούμε στοιχεία για συγκεκριμένο σημείο του κώδικα ώστε να μπορέσουμε να βελτιστοποιήσουμε την εφαρμογή μας.

4.6. Build

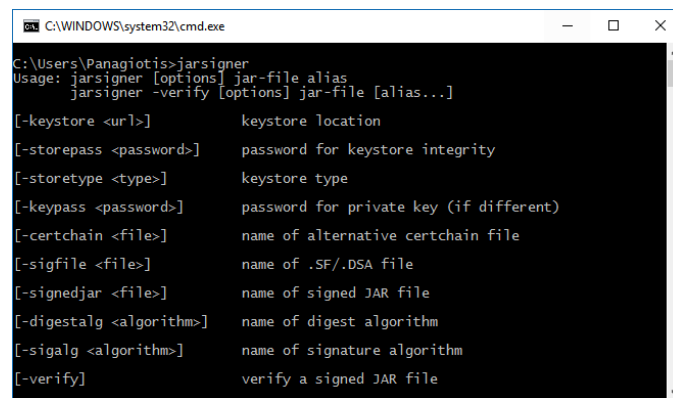
Η ανάπτυξη της εφαρμογής ολοκληρώνεται με το build. Σε αυτό το στάδιο επιλέγουμε την πλατφόρμα για την οποία θέλουμε να κάνουμε compile την εφαρμογή μας. Οι παράμετροι για την κάθε πλατφόρμα ορίζονται μέσα από το “Project Management” του Intel XDK. Αφού επιλέξουμε την πλατφόρμα το Intel XDK ανεβάζει τον κώδικα της εφαρμογής στους Server της Intel προκειμένου να κάνει το compile. Μετά το πέρας της διαδικασίας μας δίνεται έναν secure link ώστε να κατεβάσουμε το εκτελέσιμο αρχείο.



Εικόνα 19 Παράδειγμα build εφαρμογής

Για να μπορέσουμε να διανεύουμε την εφαρμογή μας στο PlayStore θα πρέπει πρώτα να κάνουμε sign το εκτελέσιμο αρχείο (apk) και στην συνέχεια να χρησιμοποιήσουμε το zipalign.

Η υπογραφή των apk είναι μια απαραίτητη διαδικασία στο Android που αποσκοπεί στην επαλήθευση του author της εφαρμογής. Δεν απαιτείται το πιστοποιητικό να έχει εκδοθεί από κάποιο Certificate Authority. Η διαδικασία είναι εύκολη και γίνεται με χρήση του jarsigner, ένα εργαλείο που διανέμεται μαζί με το Java Developer Kit. Στην εικόνα 19 απεικονίζεται το cli help του jarsigner.



```
C:\WINDOWS\system32\cmd.exe
C:\Users\Panagiotis>jarsigner
Usage: jarsigner [options] jar-file alias
       jarsigner -verify [options] jar-file [alias...]

[-keystore <url>]      keystore location
[-storepass <password>] password for keystore integrity
[-storetype <type>]    keystore type
[-keypass <password>] password for private key (if different)
[-certchain <file>]   name of alternative certchain file
[-sigfile <file>]     name of .SF/.DSA file
[-signedjar <file>]   name of signed JAR file
[-digestalg <algorithm>] name of digest algorithm
[-sigalg <algorithm>] name of signature algorithm
[-verify]             verify a signed JAR file
```

Εικόνα 20 Jarsigner

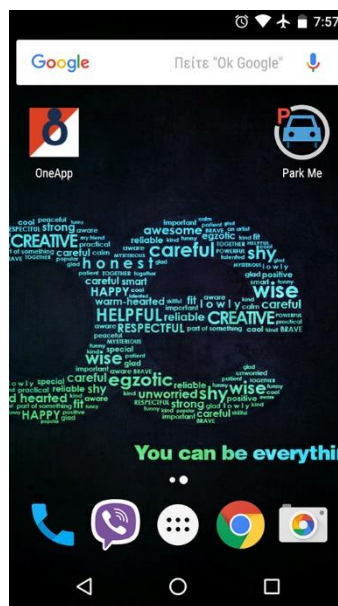
Το υπογεγραμμένο αρχείο που παράχθηκε θα πρέπει να το περάσουμε από διαδικασία zipalign προκειμένου να μπορέσουμε να το διαθέσουμε στους χρήστες μας μέσω του Play Store. Το zipalign είναι ένα εργαλείο ευθυγράμμισης αρχείων που βελτιστοποιεί τις εφαρμογές για το Android. Σκοπός του είναι να διασφαλίσει ότι όλα τα ασυμπίεστα δεδομένα ευθυγραμμίζονται σε σχέση με την αρχή του αρχείου. Πιο συγκεκριμένα όλα τα αρχεία εντός του apk θα πρέπει να ευθυγραμμιστούν σε ένα όριο των 4byte. Αυτό επιτρέπει σε όλα τα μέρη να έχουν πρόσβαση στην mmap()³⁰ ακόμα και δυαδικά δεδομένα. Το όφελος που προκύπτει από την διαδικασία αυτή είναι μικρότερη χρήση μνήμης κατά την εκτέλεση της εφαρμογής

³⁰ mmap(): κάνει mapping αρχείων ή συσκευών στην μνήμη.

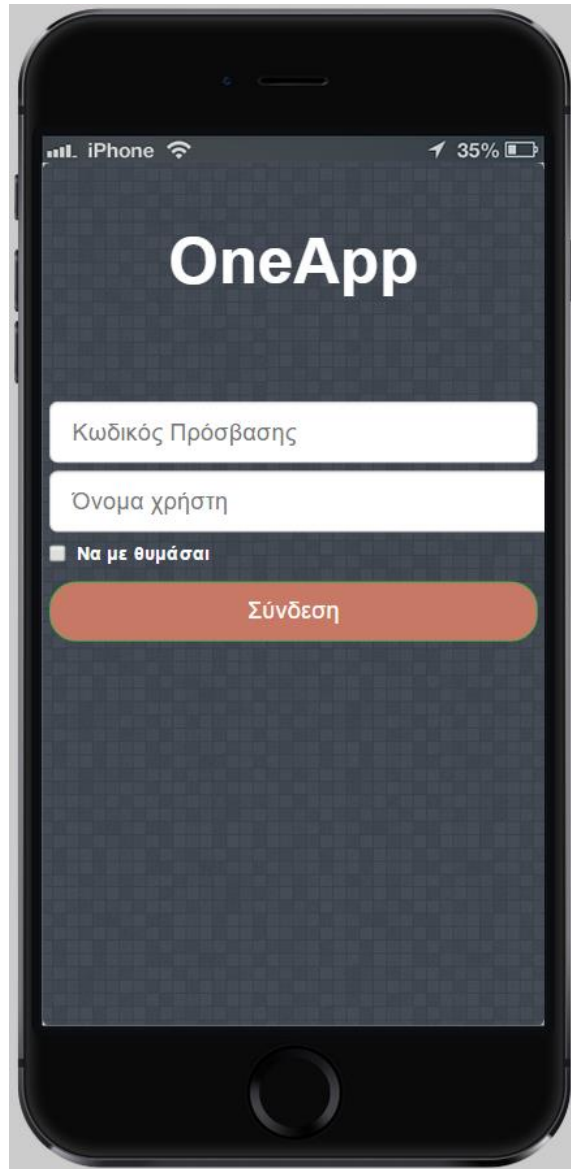
5. Εγχειρίδιο χρήστη

Στο τρέχων κεφάλαιο παρουσιάζονται εικόνες χρήσης από την εφαρμογή που ανέπτυξα κατά την μεταπτυχιακή μου εργασία. Λόγω του αντικειμένου της εφαρμογής και των προσωπικών δεδομένων που εμφανίζονται σκοπίμως κάποιες από τις οθόνες της εφαρμογής εμφανίζονται σε αρχικό στάδιο της ανάπτυξης ή με καλυμμένα κάποια σημεία.

Η εφαρμογή είναι προσβάσιμη μέσω του αντίστοιχου εικονιδίου.

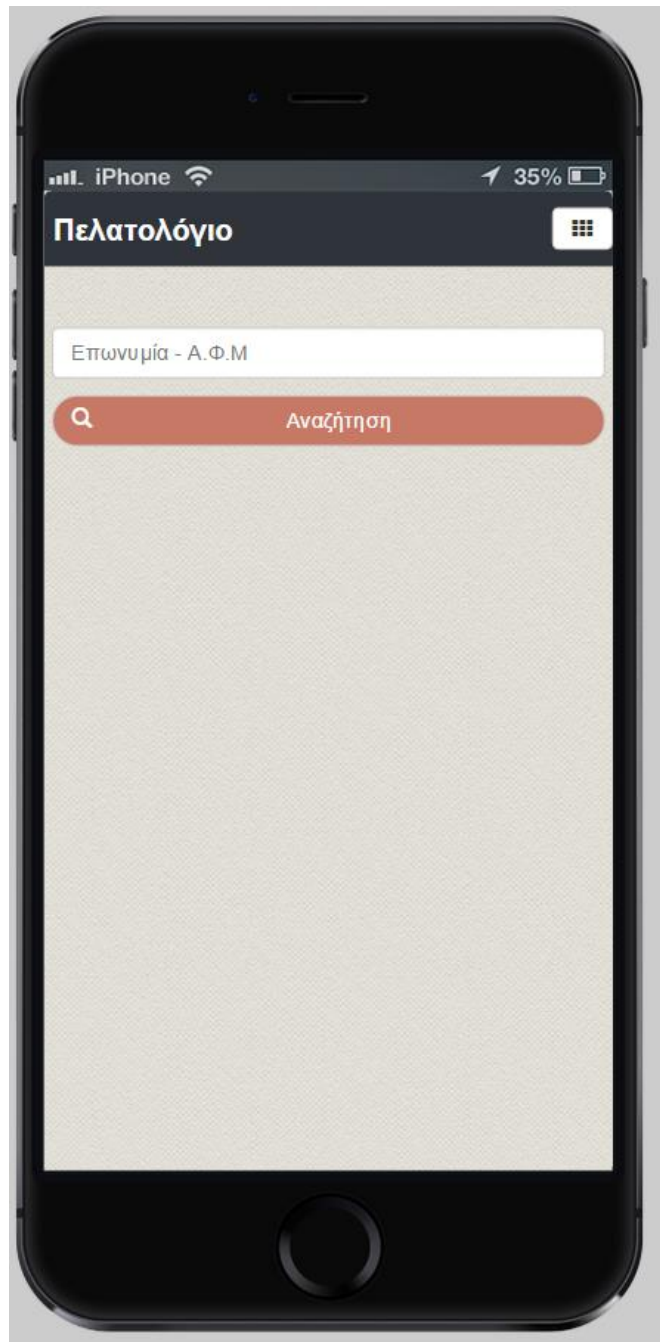


Εικόνα 21 Εικονίδιο εφαρμογής



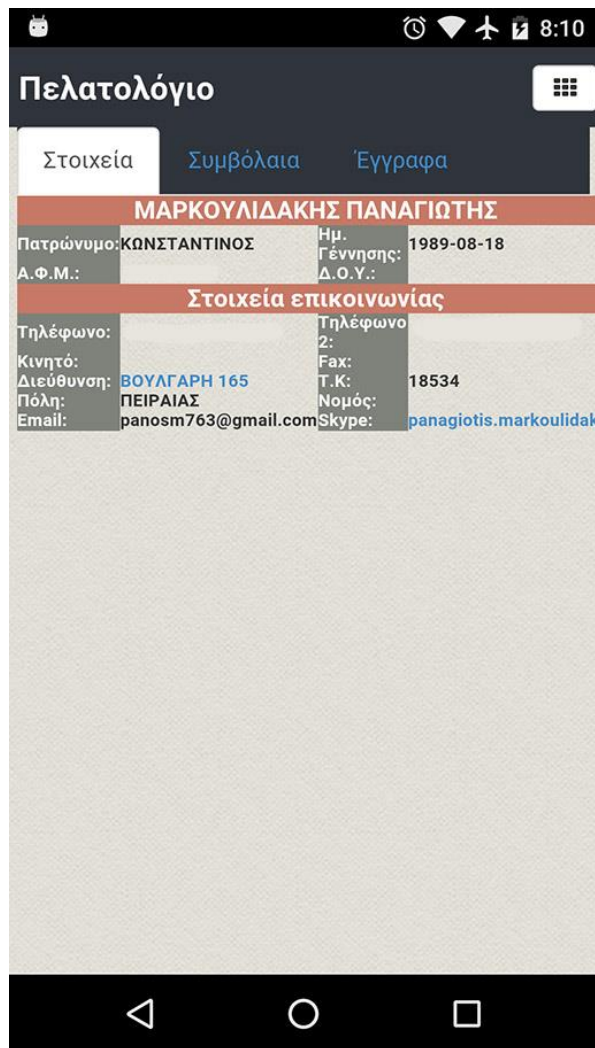
Εικόνα 22 Οθόνη πιστοποίησης χρήστη

Για την πρόσβαση στην εφαρμογή ο συνεργάτης πρέπει πρώτα να πιστοποιηθεί από την εφαρμογή. Ο χρήστης μπορεί να γίνει απομνημόνευση των στοιχείων του, σε αυτή την περίπτωση μετά την ταυτοποίηση του χρήστη το token που θα λάβει από το authentication service θα αποθηκευτεί για στο private storage της εφαρμογής.



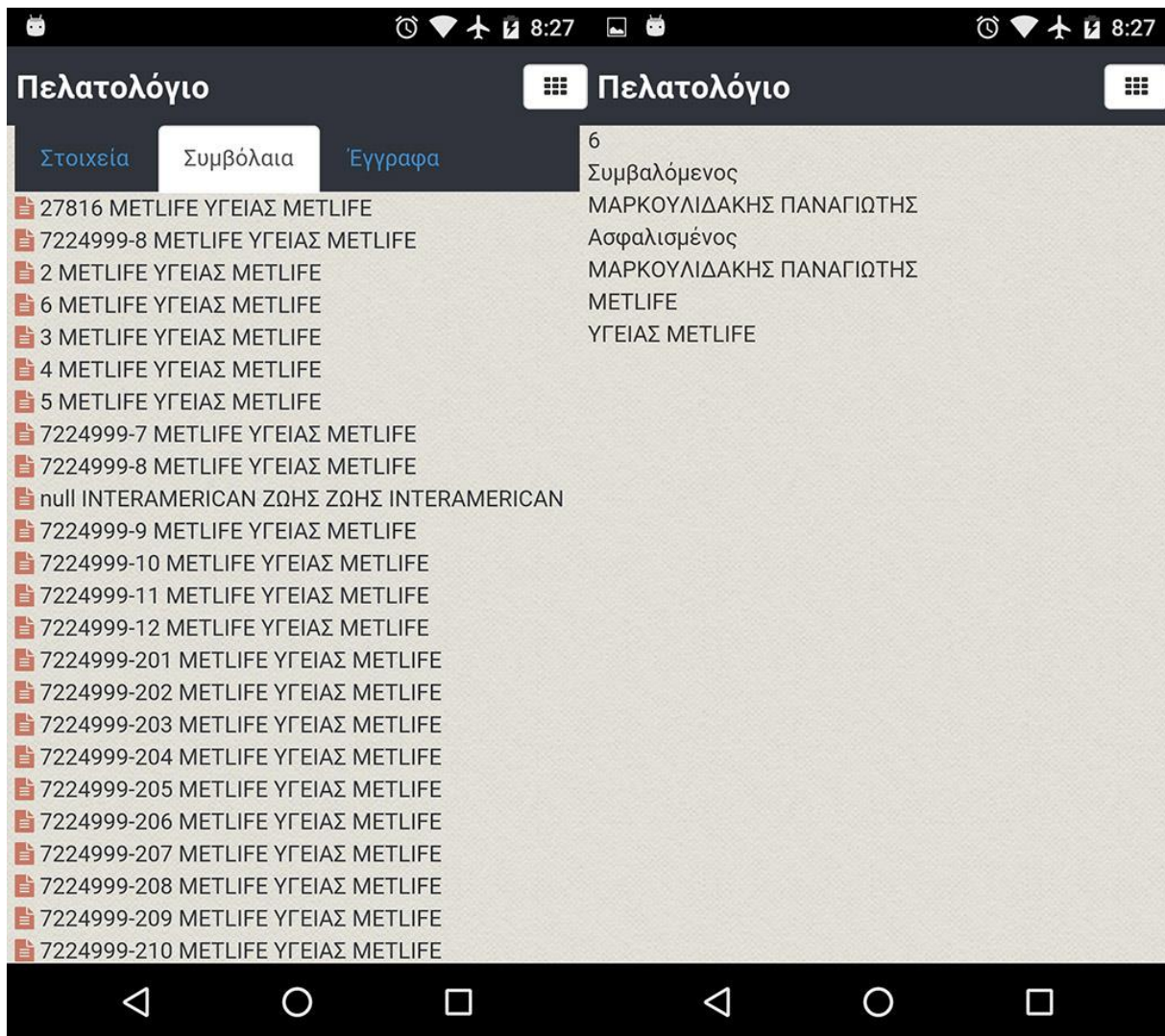
Εικόνα 23 αρχική οθόνη εφαρμογής

Μετά την πιστοποίηση του χρήστη η εφαρμογή μεταβαίνει στην κεντρική οθόνη στην οποία μπορεί να πραγματοποιήσει αναζήτηση πελατών βάσει «Επωνυμίας» ή ΑΦΜ. Κάθε συνεργάτης μπορεί να δει στοιχεία που ανήκουν στο δικό τους χαρτοφυλάκιο.



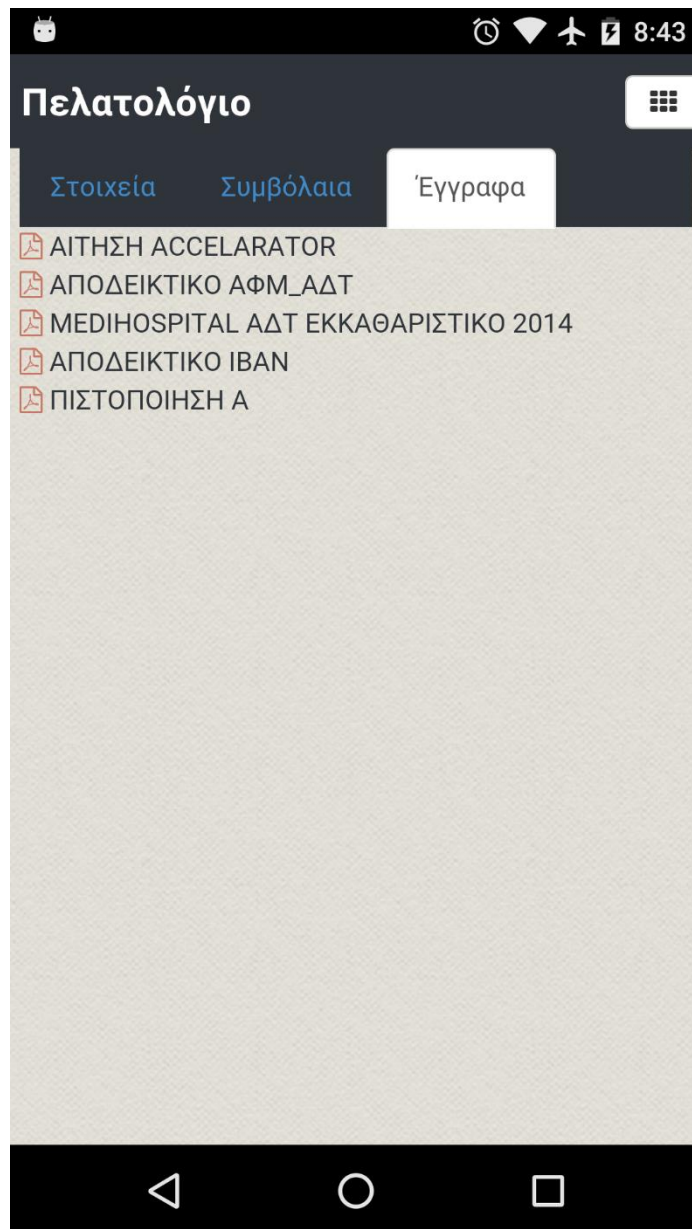
Εικόνα 24 Παράδειγμα καρτέλας πελάτη

Μετά την αναζήτηση ο χρήστης μπορεί να δει την καρτέλα του πελάτη. Στην καρτέλα υπάρχουν τα τυπικά στοιχεία του πελάτη και τα στοιχεία επικοινωνίας του. Για την διευκόλυνση του χρήστη η διεύθυνση είναι ενεργός σύνδεσμος που ενεργοποιεί την αντίστοιχη εφαρμογή πλοήγησης για την μετάβαση στην τοποθεσία. Αντίστοιχα τα τηλέφωνα ενεργοποιούν την εφαρμογή κλήσεων ενώ το skype username ενεργοποιεί την εφαρμογή skype εάν είναι εγκατεστημένη. Μέσα από το μενού ο χρήστης μπορεί να δει λίστα με τα συμβόλαια του πελάτη και τα έγγραφα που υπάρχουν στο αρχείο.



Εικόνα 25 Αριστερά: Λίστα συμβολαίων πελάτη. Δεξιά: Εικόνα συμβολαίου

Στην αριστερή εικόνα εμφανίζεται η λίστα των συμβολαίων του πελάτη. Η ταξινόμηση τους γίνεται βάση της ημερομηνία έναρξής τους και τα στοιχεία που εμφανίζονται είναι ο αριθμός συμβολαίου η εταιρεία και ο κλάδος. Ανοίγοντας ένα συμβόλαιο βλέπουμε τα τυπικά του στοιχεία. Στην εικόνα 25 απεικονίζεται ο αρχικός σχεδιασμός της καρτέλας ενός συμβολαίου. Μέσα στην καρτέλα του συμβολαίου ο συνεργάτης μπορεί να βρει μια πληθώρα στοιχείων ανάλογα των κλάδο που ανήκει το συμβόλαιο. Για τα συμβόλαια ζωής πέρα από τα τυπικά στοιχεία του συμβολαίου ο συνεργάτης μπορεί να βρει οικονομικά στοιχεία του συμβολαίου(μικτά και καθαρά ασφάλιστρα), τον κωδικό ηλεκτρονικής πληρωμής και πιο ειδικά στοιχεία όπως οι δικαιούχοι του συμβολαίου ή τα προστατευόμενα μέλη.



Εικόνα 26 Επισυναπτόμενα έγγραφα πελάτη

Τέλος από το tab έγγραφα ο συνεργάτης μπορεί να δει τα διαθέσιμα έγγραφα του πελάτη του και να τα κατεβάσει. Για την διευκόλυνση του χρήστη το εικονίδιο αλλάζει ανάλογα με τον τύπο του αρχείου. Οι διαθέσιμες κατηγορίες είναι: PDF, έγγραφα κειμένου, υπολογιστικά φύλλα, εικόνες, ενώ υπάρχει και ένα γενικό εικονίδιο για τα άγνωστα αρχεία.

6. Συμπεράσματα

Η διαρκής αύξηση των χρηστών των κινητών συσκευών ωθεί τις εταιρείες και τους προγραμματιστές για την ανάπτυξη εφαρμογών για κινητές συσκευές. Τα εκτιμώμενα κέρδη από τις εφαρμογές για κινητές συσκευές υπολογίζονται περί τα 77δισ δολάρια. Η δημιουργία native apps για τις διαφορετικές πλατφόρμες κινητών εφαρμογών έχει ως αποτέλεσμα την σπατάλη προγραμματιστικού χρόνου και είναι δαπανηρή. Η εξέλιξη των τεχνολογιών Web έρχονται να δώσουν τα απαραίτητα εργαλεία για την δημιουργία Cross-Platform εφαρμογών.

Μέσα από την παρούσα διπλωματική εργασία απόκτησα το θεωρητικό υπόβαθρο για την ανάπτυξη mobile εφαρμογών ενώ μέσα από την ανάπτυξη της εφαρμογής (One App) χρησιμοποίησα τεχνολογίες αιχμής. Κατά τον σχεδιασμό της πλατφόρμας και των Web Services χρειάστηκε να αντιμετωπίσω θέματα performance τα οποία και έλυσα με την αξιοποίηση τεχνολογιών όπως το HTTP/2 την PHP 7.0 και το Anycast DNS. Λόγω της φύσης των δεδομένων που χρησιμοποιεί η εφαρμογή η ασφάλεια αποτέλεσε σημαντική προτεραιότητα. Η χρήση των JSON Web Tokens για την πιστοποίηση των χρηστών αλλά και η ενσωμάτωση TLS στα Web Services ήταν ένα βήμα προς τα εκεί. Τέλος ο σχεδιασμός της εφαρμογής με τέτοιο τρόπο ώστε να είναι συμβατή με περισσότερες πλατφόρμες κινητών συσκευών με βοήθησε ώστε να δουλέψω με γνωστές σε εμένα τεχνολογίες αλλά με τρόπο τον οποίο μέχρι και σήμερα δεν είχα αξιοποιήσει.

7. Βιβλιογραφία

- <menu>*. (n.d.). Ανάκτηση από Mozilla Developer Network: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/menu>
- 4G*. (n.d.). Ανάκτηση από Wikipedia: <https://en.wikipedia.org/wiki/4G>
- Android Fragmentation visualized*. (2015, Αύγουστος). Ανάκτηση από OpenSignal: https://opensignal.com/assets/pdf/reports/2015_08_fragmentation_report.pdf
- Android Fragmentation Visualized*. (2015, Αύγουστος). Ανάκτηση από OpenSignal: <http://opensignal.com/reports/2015/08/android-fragmentation/>
- Cass, S. (2015, Ιούλιος 20). *The 2015 Top Ten Programming Languages*. Ανάκτηση από IEEE Spectrum: <http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages>
- Cordova Overview*. (2016). Ανάκτηση από Apache foundation: <https://cordova.apache.org/docs/en/6.x/guide/overview/index.html>
- Desktop Operating System Market Share*. (2016, 02). Ανάκτηση από NETMARKETSHARE: <https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0>
- Developer Economics Q3 2014: State of the Developer Nation*. (2014). Ανάκτηση από Developer Economics: <https://www.developereconomics.com/reports/developer-economics-q3-2014/>
- Flash, HTML5 and Open Web Standards*. (2015, Νοέμβριος 30). Ανάκτηση από Adobe Systems: <https://blogs.adobe.com/conversations/2015/11/flash-html5-and-open-web-standards.html>
- Getting started with WebGL*. (n.d.). Ανάκτηση από Mozilla Developer Network: https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/Tutorial/Getting_started_with_WebGL
- Growth of Time Spent on Mobile Devices Slows*. (2015, Οκτώβριος 7). Ανάκτηση από eMarketer: <http://www.emarketer.com/Article/Growth-of-Time-Spent-on-Mobile-Devices-Slows/1013072>
- High Resolution Time Level 2*. (n.d.). Ανάκτηση από W3C: <http://w3c.github.io/hr-time/>

- HTML structural elements.* (n.d.). Ανάκτηση από w3c wiki:
https://www.w3.org/wiki/HTML_structural_elements
- Irish, P. (n.d.). *When milliseconds are not enough: performance.now.* Ανάκτηση από Google Developers: <https://developers.google.com/web/updates/2012/08/When-milliseconds-are-not-enough-performance-now>
- jQuery Usage Statistics.* (2016, 02). Ανάκτηση από Built With:
<http://trends.builtwith.com/javascript/jQuery>
- JSON Web Token.* (2015, Μάιος). Ανάκτηση από Internet Engineering Task Force:
<https://tools.ietf.org/html/rfc7519>
- Miller, C. (2016, Ιανουάριος 20). *iOS App Store brings in 75% more revenue than Play Store despite difference in downloads.* Ανάκτηση από 9to5Mac:
<http://9to5mac.com/2016/01/20/app-store-ios-downloads-vs-android-revenue/>
- MIT License.* (n.d.). Ανάκτηση από Open Source Initiative:
<https://opensource.org/licenses/MIT>
- mmap.* (n.d.). Ανάκτηση από Wikipedia: <https://en.wikipedia.org/wiki/Mmap>
- Mobile application development.* (n.d.). Ανάκτηση από Wikipedia:
https://en.wikipedia.org/wiki/Mobile_application_development
- Murphy, D. (2015, Μάιος). *Microsoft Going After Smartphone Fragmentation in Windows 10 Mobile.* Ανάκτηση από PC Magazine:
<http://www.pcmag.com/article2/0,2817,2484312,00.asp>
- P, V. (2014, Ιούλιος 27). *What is Cordova and how does it work.* Ανάκτηση από SAP:
<http://scn.sap.com/community/developer-center/mobility-platform/blog/2014/07/27/what-is-cordova-and-how-does-it-work>
- PhoneGap, Cordova, and what's in a name?* (2012, Μάρτιος 19). Ανάκτηση από Adobe PhoneGap:
<http://phonegap.com/2012/03/19/phonegap-cordova-and-what%E2%80%99s-in-a-name/>
- Scan, B. (2009, Ιούνιος 04). *Father of the Cell Phone.* Ανάκτηση από Economist:
[http://www.economist.com/node/13725793?story_id=13725793,](http://www.economist.com/node/13725793?story_id=13725793)

Subresource Integrity. (n.d.). Ανάκτηση από Mozilla Developer Network:
https://developer.mozilla.org/en-US/docs/Web/Security/Subresource_Integrity

Using Hardware Devices. (n.d.). Ανάκτηση από Android developers:
<http://developer.android.com/tools/device.html>

WebKit. (2016, Μάρτης 3). Ανάκτηση από Wikipedia: <https://en.wikipedia.org/wiki/WebKit>

WebView. (n.d.). Ανάκτηση από Google developers:
<http://developer.android.com/reference/android/webkit/WebView.html>

What is load balancing. (2016). Ανάκτηση από Nginx:
<https://www.nginx.com/resources/glossary/load-balancing/>

Whittaker, Z. (2015, Φεβρουάριος). <http://www.zdnet.com/article/ios-7-fragmentation-issues/>.
Ανάκτηση από IOS 7 fragmentation issues: <http://www.zdnet.com/article/ios-7-fragmentation-issues/>

