



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΩΝ & ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

---

«ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ  
ΓΙΑ ΚΙΝΗΤΑ ΤΗΛΕΦΩΝΑ ΑΞΙΟΠΟΙΩΝΤΑΣ  
ΤΙΣ ΠΛΗΡΟΦΟΡΙΕΣ ΤΩΝ ΔΙΣΘΗΤΗΡΩΝ  
ΤΟΥΣ»

---

**ΣΤΑΣΙΝΟΣ ΝΙΚΟΛΑΟΣ**

**A.M. 4020**

**ΥΠΕΥΘΥΝΟΣ ΚΑΘΗΓΗΤΗΣ**

**ΧΡΗΣΤΟΣ Ι. ΜΠΟΥΡΑΣ, ΚΑΘΗΓΗΤΗΣ**

**ΕΠΙΒΛΕΠΟΝΤΕΣ**

**ΑΝΔΡΕΑΣ ΠΑΠΑΖΩΗΣ**

**ΒΑΣΙΛΕΙΟΣ ΚΟΚΚΙΝΟΣ**

**ΠΑΤΡΑ 2013**



# ΠΡΟΛΟΓΟΣ

---

---

Η παρούσα διπλωματική εργασία αποτελεί τον επίλογο των σπουδών μου ως προπτυχιακός φοιτητής του τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής (ΤΜΗΥΠ) του Πανεπιστημίου Πατρών. Ο τίτλος της διπλωματικής εργασίας είναι «Σχεδίαση και Υλοποίηση Εφαρμογής για Κινητά Τηλέφωνα Αξιοποιώντας τις Πληροφορίες των Αισθητήρων τους».

Πριν την παρουσίαση και ανάλυση των αποτελεσμάτων της διπλωματικής εργασίας μου θα ήθελα να ευχαριστήσω θερμά όσους με βοήθησαν και με συμβούλευσαν όλο αυτό το διάστημα.

Θα ήθελα να ευχαριστήσω τον κ. Χρήστο Μπούρα, Καθηγητή του ΤΜΗΥΠ και υπεύθυνο καθηγητή της διπλωματικής μου, για την καθοδήγηση του αλλά και για την ευκαιρία που μου έδωσε να ασχοληθώ με το συγκεκριμένο θέμα και να εμπλουτίσω τις γνώσεις μου.

Επίσης θα ήθελα να ευχαριστήσω θερμά τον Ανδρέα Παπαζώη, επιβλέποντα της διπλωματικής μου εργασίας, για όλη την υποστήριξη, καθοδήγηση και συμβουλές που μου πρόσφερε σε όλη την διάρκεια εκπόνησής της καθώς επίσης και τον Βασίλειο Κόκκινο για την υποστήριξη του.

Τέλος ευχαριστώ τους φίλους μου για την στήριξη τους σε αυτή την προσπάθεια και φυσικά τους γονείς μου και την αδελφή μου για όλα όσα έχουν κάνει για μένα και για την υποστήριξη τους σε όλη την διάρκεια των σπουδών μου.

Νικόλαος Στασινός



# ΠΕΡΙΕΧΟΜΕΝΑ

---

---

<b>ΛΙΣΤΑ ΕΙΚΟΝΩΝ .....</b>	<b>9</b>
<b>ΛΙΣΤΑ ΚΩΔΙΚΩΝ.....</b>	<b>11</b>
<b>ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ ΚΑΙ ΑΚΡΩΝΥΜΙΑ .....</b>	<b>13</b>
<b>1 ΕΙΣΑΓΩΓΗ .....</b>	<b>15</b>
1.1 ΔΟΜΗ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ .....	17
<b>2 HTML5.....</b>	<b>19</b>
2.1 SEMANTIC MARKUP .....	20
2.2 MEDIA .....	21
2.3 CANVAS 2D CONTEXT .....	21
2.4 FORM ENHANCEMENTS.....	22
2.5 CASCADING STYLE SHEETS .....	22
2.5.1 <i>Selectors</i> .....	23
2.5.2 <i>Transforms</i> .....	23
2.5.3 <i>Transitions</i> .....	23
2.5.4 <i>Animations</i> .....	23
2.5.5 <i>Media Queries</i> .....	24
2.5.6 <i>Web Fonts</i> .....	24
2.6 PAGE VISIBILITY ΚΑΙ TIMER CONTROL.....	24
2.7 SESSION HISTORY .....	25
2.8 DRAG AND DROP .....	25
2.9 SVG AND MATHML.....	25
2.10 WebGL AND TYPED ARRAYS .....	26
2.11 TOUCH EVENTS.....	26
2.12 OFFLINE WEB APPLICATIONS .....	27
2.12.1 <i>Application Cache</i> .....	27
2.12.2 <i>Data Storage</i> .....	28
2.12.3 <i>Κατάσταση Συνδεσιμότητας</i> .....	30
2.13 FILES.....	30
2.14 DEVICE APIS .....	31
2.14.1 <i>Geolocation</i> .....	31
2.14.2 <i>Device Orientation</i> .....	32
2.14.3 <i>User Media</i> .....	33
2.15 WEB REAL-TIME COMMUNICATION .....	33

2.16	WEB SOCKETS .....	34
2.17	SERVER-SENT EVENTS .....	34
2.18	WEB WORKERS .....	34
2.19	ΆΛΛΑ ΠΡΟΤΥΠΑ .....	35
<b>3</b>	<b>HTML5 FRAMEWORKS .....</b>	<b>37</b>
3.1	JQUERY MOBILE .....	37
3.2	JQT .....	38
3.3	SENGHA TOUCH .....	38
3.4	ADOBE PHONEGAP .....	39
3.5	5ML .....	41
3.6	PARSE .....	42
3.7	TITANIUM APPCELERATOR .....	44
<b>4</b>	<b>ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΡΧΙΤΕΚΤΟΝΙΚΗ .....</b>	<b>47</b>
4.1	MOBILE ΣΤΡΑΤΗΓΙΚΗ .....	47
4.1.1	<i>Είδη Web Εφαρμογών .....</i>	<i>47</i>
4.2	CONTEXT .....	49
4.3	PROGRESSIVE ENHANCEMENT .....	50
4.4	REGRESSIVE ENHANCEMENT .....	51
4.5	RESPONSIVE DESIGN .....	51
4.6	ΓΕΝΙΚΕΣ ΙΔΕΕΣ .....	51
4.7	ΣΧΕΔΙΑΣΜΟΣ ΕΦΑΡΜΟΓΗΣ .....	53
4.7.1	<i>Κύρια Σελίδα .....</i>	<i>54</i>
4.7.2	<i>Σελίδα Gallery .....</i>	<i>55</i>
4.7.3	<i>Σελίδα Map .....</i>	<i>56</i>
4.7.4	<i>Επεκτασιμότητα Εφαρμογής .....</i>	<i>56</i>
4.7.5	<i>Αρχιτεκτονική .....</i>	<i>57</i>
<b>5</b>	<b>ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ .....</b>	<b>61</b>
5.1	ΑΝΙΧΝΕΥΣΗ HTML5 ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ .....	61
5.2	OFFLINE APPLICATION .....	61
5.2.1	<i>Application Cache .....</i>	<i>61</i>
5.2.2	<i>Ανίχνευση Συνδεσιμότητας .....</i>	<i>62</i>
5.3	ΔΙΑΧΕΙΡΙΣΗ ΑΡΧΕΙΩΝ .....	63
5.3.1	<i>Media Capture .....</i>	<i>63</i>
5.3.2	<i>File Api .....</i>	<i>63</i>
5.3.3	<i>FileSystem Api .....</i>	<i>64</i>
5.3.4	<i>Local Storage .....</i>	<i>64</i>
5.3.5	<i>WebSQL Database .....</i>	<i>64</i>
5.3.6	<i>Geolocation Api .....</i>	<i>66</i>
5.4	ΔΙΑΧΕΙΡΙΣΗ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΧΡΗΣΤΗ .....	66
5.4.1	<i>Swipe Events .....</i>	<i>66</i>
5.4.2	<i>Orientation Change Event .....</i>	<i>68</i>
5.4.3	<i>CSS3 2D Transform .....</i>	<i>69</i>

5.4.4	<i>Προβολή Αρχείων</i> .....	69
5.4.5	<i>Επικοινωνία με το Χρήστη</i> .....	69
5.4.6	<i>Viewport</i> .....	71
5.5	ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΑΡΧΕΙΟΥ.....	72
5.5.1	<i>Λήψη Αρχείου</i> .....	72
5.5.2	<i>Μεταφόρτωση Αρχείου</i> .....	72
5.5.3	<i>Facebook</i> .....	73
<b>6</b>	<b>ΛΕΙΤΟΥΡΓΙΕΣ</b> .....	<b>77</b>
6.1	ΕΝΤΟΠΙΣΜΟΣ FEATURES.....	77
6.2	MAIN PAGE .....	78
6.3	GALLERY.....	80
6.3.1	<i>Αναπαραγωγή αρχείων</i> .....	81
6.3.2	<i>Location</i> .....	83
6.3.3	<i>Uploading Αρχείων</i> .....	84
6.3.4	<i>Facebook</i> .....	84
6.3.5	<i>Λήψη Αρχείου</i> .....	86
6.4	ΛΕΙΤΟΥΡΓΙΑ FULL SCREEN .....	86
6.5	VIDEO-DEMO ΛΕΙΤΟΥΡΓΙΩΝ .....	88
<b>7</b>	<b>ΣΥΜΠΕΡΑΣΜΑΤΑ</b> .....	<b>89</b>
<b>8</b>	<b>ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ</b> .....	<b>91</b>
	<b>Β Ι Β Λ Ι Ο Γ Ρ Α Φ Ι Α</b> .....	<b>93</b>





# ΛΙΣΤΑ ΕΙΚΟΝΩΝ

---

---

Εικόνα 1 - Mobile OS and Programming Languages .....	16
Εικόνα 2 - jQuery Mobile user interface components .....	37
Εικόνα 3 - jQT User Interface Components.....	38
Εικόνα 4 - Sencha Touch User Interface Components.....	39
Εικόνα 5 - Adobe PhoneGap Build Interface.....	40
Εικόνα 6 - Διαδικασία ανάπτυξης εφαρμογής με PhoneGap .....	41
Εικόνα 7 - 5ml Development Interface .....	42
Εικόνα 8 - Data Browser.....	43
Εικόνα 9 - Cloud Code Editor.....	44
Εικόνα 10 - Titanium Appcelerator .....	44
Εικόνα 11 - Web Application .....	49
Εικόνα 12 - Android Native Web App.....	49
Εικόνα 13 - Σχέδιο Κύριας Οθόνης.....	53
Εικόνα 14 - Οθόνη Gallery.....	53
Εικόνα 15 - Κυρια Οθονη Εφαρμογης.....	54
Εικόνα 16 - Gallery .....	54
Εικόνα 17 - Αρχιτεκτονική Εφαρμογής .....	57
Εικόνα 18 - Facebook App Settings .....	73
Εικόνα 19 - Features Detection 1 .....	77
Εικόνα 20 - Features Detection 2 .....	77
Εικόνα 21 - Κύριες λειτουργίες .....	78
Εικόνα 22 - Φωτογράφιση .....	78
Εικόνα 23 - Βιντεοσκόπηση.....	79
Εικόνα 24 - Ηχογράφιση .....	79
Εικόνα 25 - Toast Notification .....	79

Εικόνα 26 - Mini Preview.....	79
Εικόνα 27- Geolocation Off.....	80
Εικόνα 28 - Geolocation On.....	80
Εικόνα 29 - Gallery .....	81
Εικόνα 30 - Gallery Menu.....	81
Εικόνα 31 - Κάθετη απεικόνιση εικόνας.....	82
Εικόνα 32 - Οριζόντια απεικόνιση εικόνας.....	82
Εικόνα 33 - Video Preview.....	82
Εικόνα 34 - Audio Preview .....	82
Εικόνα 35 - Landscape Video.....	83
Εικόνα 36 - Location on Map.....	83
Εικόνα 37 - Upload Confirmation.....	84
Εικόνα 38 - Uploading process indicator.....	84
Εικόνα 39 - Upload completed notification .....	84
Εικόνα 40 - Confirm Facebook Share .....	85
Εικόνα 41 - Facebook Login Screen.....	85
Εικόνα 42 - Photo on Facebook.....	85
Εικόνα 43 - Video on Facebook.....	85
Εικόνα 44 - Download File.....	86
Εικόνα 45 - Download Complete.....	86
Εικόνα 46 - Preview with native app.....	86
Εικόνα 47 - Full Screen Mode.....	87

# ΛΙΣΤΑ ΚΩΔΙΚΩΝ

---

---

Κώδικας 1 - Video Element .....	21
Κώδικας 2 - Canvas.....	21
Κώδικας 3 - Touch Events.....	27
Κώδικας 4 - AppCache Manifest .....	28
Κώδικας 5 - LocalStorage .....	28
Κώδικας 6 - LocalStorage Object.....	29
Κώδικας 7 - LocalStorage Delete & Length .....	29
Κώδικας 8 - File Input Element.....	31
Κώδικας 9 - Geolocation .....	32
Κώδικας 10 - OrientationChange Event .....	33
Κώδικας 11 - Camcorder Input File Element.....	33
Κώδικας 12 - AppCache Manifest Example.....	62
Κώδικας 13 - Connectivity Check .....	62
Κώδικας 14 - “Device” Input Elements.....	63
Κώδικας 15 - Onclick Event for Input Elements .....	63
Κώδικας 16 - File System Save File.....	64
Κώδικας 17 - Geolocation .....	66
Κώδικας 18 - Swipe Handle .....	67
Κώδικας 19 - Orientation Change Handle .....	68
Κώδικας 20 - CSS3 2D Scale Transform .....	69
Κώδικας 21 - Preview Files .....	69
Κώδικας 22 - Notifications & Dialog Box.....	70
Κώδικας 23 - Viewport, FullScreen & Homescreen Icon.....	71
Κώδικας 24 - Download Link Element.....	72

Κώδικας 25 - Ajax File Upload .....	72
Κώδικας 26 - Parse & Facebook Init .....	74
Κώδικας 27 - Facebook Upload Photo .....	74
Κώδικας 28 - Facebook Login Status .....	75

# ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ ΚΑΙ ΑΚΡΩΝΥΜΙΑ

---

---

3G	3rd Generation Mobile Telecommunications
API	Application Programming Interface
App	Application
Ajax	Asynchronous JavaScript and XML
AppCache	Application Cache
Blob	Binary Large Object
CPU	Central Processing Unit
CSS	Cascading Style Sheets
CSS3	Cascading Style Sheets level 3
DOM	Document Object Model
GPS	Global Positioning System
HTML	HyperText Markup Language
HTML5	HyperText Markup Language version 5
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDE	Integrated Development Environment
iOS	iPhone Operating System (Apple)
IP	Internet Protocol
MathML	Mathematical Markup Language
OS	Operating System
OpenGL ES	Open Graphics Library for Embedded Systems
RTMP	Real Time Messaging Protocol
SDK	Software Development Kit
SQL	Structured Query Language
SVG	Scalable Vector Graphics
TCP	Transmission Control Protocol
UI/U.I.	User Interface
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WHATWG	Web Hypertext Application Technology Working Group
WWW	World Wide Web
WebGL	Web Graphics Library
WiFi	Wireless Local Area Network (WLAN)
XHTML	eXtensible HyperText Markup Language
XML	Extensible Markup Language



# 1 Εισαγωγή

Τα τελευταία χρόνια ο τομέας των εφαρμογών κινητών συσκευών έχει σημειώσει μεγάλη επέκταση, με εταιρίες και ανεξάρτητους developers να αφιερώνουν όλο και περισσότερο χρόνο και χρήμα για ανάπτυξη καινοτόμων και εύχρηστων εφαρμογών. Φυσικά αυτό το γεγονός οφείλεται στον όλο και μεγαλύτερο αριθμό χρηστών έξυπνων κινητών συσκευών, όπως smartphones και tablets. Με την έλευση του πρώτου iPhone και ύστερα με την διάδοση του Android η άποψη μας για το τι είναι κινητό τηλέφωνο, και γενικότερα κινητή συσκευή, έχει αλλάξει ριζικά. Έχει εξελιχθεί πλέον σε ένα «πολύ-εργαλείο» και φυσικά αναπόσπαστο κομμάτι της καθημερινής μας ζωής.

Σήμερα υπάρχουν αρκετές και διαφορετικές πλατφόρμες για ανάπτυξη εφαρμογών. Οι δυο κυρίαρχες αυτή την στιγμή είναι το iOS και το Android που κατέχουν το μεγαλύτερο μερίδιο αγοράς. Πέρα από αυτά τα δυο λειτουργικά συστήματα για κινητές συσκευές υπάρχουν επίσης και : RIM Blackberry , Windows Phone , HP Palm webOS , Samsung Bada , Symbian, MeeGo. Να αναφέρουμε επίσης και τα Tizen (Samsung + Intel) , Firefox OS (Mozilla) και Ubuntu Touch, λειτουργικά που εμφανιστήκαν σχετικά πρόσφατα.

Το καθένα από αυτά τα λειτουργικά λοιπόν έχουν τα δικά τους software developments kits, απαιτούν ειδικά εργαλεία για ανάπτυξη εφαρμογών που είναι διαφορετικά σε κάθε πλατφόρμα. Επίσης απαιτούν γνώσεις σε διαφορετικές γλώσσες προγραμματισμού [2] όπως μπορούμε να δούμε στην Εικόνα 1 και φυσικά έχουν και διαφορετικό τρόπο για πρόσβαση στις λειτουργίες των διαφορετικών συσκευών και των υπηρεσιών του κάθε λειτουργικού. Όλα αυτά έχουν ως αποτέλεσμα οι developers να πρέπει να έχουν εξιδανικευμένες γνώσεις για κάθε πλατφόρμα χωριστά πράγματα το οποίο φυσικά απαιτεί πολύ χρόνο και κόπο από μέρος του. Αυτό βεβαίως έχει αρνητική επιρροή και για την ίδια την εταιρεία που έχει αναλάβει την ανάπτυξη κάποιας εφαρμογής καθώς όλα τα παραπάνω απαιτούν περισσότερο χρόνο για τους εξής λόγους [2]:

- καθυστέρηση στην ολοκλήρωση ανάπτυξης της εφαρμογής,
- πιθανά προβλήματα ασυμβατότητας μεταξύ των διαφόρων συσκευών ,
- πολύ μεγαλύτερος χρόνος ελέγχου ποιότητας του τελικού προϊόντος αφού πρέπει να ελεγχθούν πολλές και αρκετά διαφορετικές συσκευές για κάθε πλατφόρμα.

Mobile OS Type	Skill Set Require
Apple iOS	C, Objective C
Google Android	Java (Harmony flavored, Dalvik VM)
RIM BlackBerry	Java (J2ME flavored)
Symbian	C, C++, Python, HTML/CSS/JS
Windows Mobile	.NET
Window 7 Phone	.NET
HP Palm webOS	HTML/CSS/JS
MeeGo	C, C++, HTML/CSS/JS
Samsung bada	C++

Εικόνα 1 - Mobile OS and Programming Languages

Το κύριο πρόβλημα της ανάπτυξης εφαρμογών για κάθε πλατφόρμα χωριστά είναι η χρήση διαφορετικών SDK και οι γνώσεις σε διαφορετικές και συχνά πιο πολύπλοκες γλώσσες προγραμματισμού. Με την εμφάνιση της HTML5 τα παραπάνω προβλήματα άρχισαν να αντιμετωπίζονται σε σημαντικό βαθμό. Πλέον η ανάπτυξη εφαρμογών δεν χρειάζεται να γίνεται στοχευόμενα σε κάθε πλατφόρμα καθώς με την HTML5 είναι δυνατή η δημιουργία cross-platform εφαρμογών που μπορούν να λειτουργήσουν ακριβώς με τον ίδιο τρόπο σε όλες τις πλατφόρμες. Η εφαρμογή σε αυτή την περίπτωση τρέχει μέσω του browser χωρίς αυτό βεβαίως να σημαίνει ότι πρέπει να βρίσκεται μόνιμα συνδεδεμένος στο Internet ο χρήστης για να μπορεί να την χρησιμοποιήσει αφού πλέον τα αρχεία που είναι απαραίτητα για να τρέξει η εφαρμογή μπορούν να αποθηκευτούν τοπικά στον browser. Φυσικά οι λειτουργίες που απαιτούν σύνδεση στο διαδίκτυο, όπως το ανέβασμα ενός αρχείου δεν θα δουλεύουν αλλά θα εμφανίζουν ένα κατάλληλο μήνυμα καθώς μπορούμε πλέον να καταλαβαίνουμε πότε ο χρήστης είναι online ή offline. Επιπρόσθετα παρέχεται και πρόσβαση σε λειτουργίες της ίδιας της συσκευής και του λειτουργικού. Επίσης ο ίδιος κώδικας με ελάχιστες αλλαγές μπορεί και να χρησιμοποιηθεί για δημιουργία υβριδικών native εφαρμογών με την χρήση εργαλείων όπως το Cordova. Και φυσικά όλα αυτά απλά αναπτύσσοντας την εφαρμογή με τεχνολογίες διαδικτύου (html, javascript, css, κτλ.) οι οποίες φυσικά είναι πολύ πιο διαδεδομένες και εύχρηστες. Παρακάτω αναφέρονται μερικά βασικά χαρακτηριστικά που παρουσιάζονται για πρώτη φορά στην HTML5 [7]:

- Video/audio element με δυνατότητα streaming και με support σε συγκεκριμένα formats
- Forms με ειδικά field types για text, search, email, date, color, file, κτλ
- Canvas element για 2d περιεχόμενο
- Drag and drop για desktop browsers
- session history



- application cache για χρησιμοποίηση της ιστοσελίδας/εφαρμογής χωρίς να είναι αναγκαία η σύνδεση στο Internet
- session storage, local storage
- indexedDB , web SQL database
- Geolocation
- device orientation
- WebGL για 3D περιεχόμενο
- server sent events όπως xmlhttprequest
- websocket
- file api
- filesystem api
- webworkers για multi-threaded web
- access webcam
- web notifications

Αξίζει τέλος να σημειώσουμε ότι πολλές καινούργιες πλατφόρμες, όπως το Tizen, Firefox OS, Ubuntu Touch κ.α. πλέον στρέφονται στην HTML5 για την ανάπτυξη των εφαρμογών τους.

Με την εφαρμογή που θα αναλύσουμε παρακάτω ο χρήστης μπορεί να αποθηκεύσει τοπικά και άμεσα οποιαδήποτε φωτογραφία-βίντεο-ηχογράφηση ακόμα και αν δημιουργήθηκε εκείνη την στιγμή μιας και η εφαρμογή προσφέρει άμεση πρόσβαση στην camera, videocamera, και microphone. Μπορεί γρήγορα και εύκολα να αναπαράγει τα αρχεία που τράβηξε και να δει και την τοποθεσία όπου δημιουργήθηκαν. Να σημειώσουμε επίσης ότι προσφέρεται και δυνατότητα ανεβάσματος σε server κάτι το οποίο μπορεί να εξελιχτεί με πολλαπλούς τρόπους και να προσφέρει επιπλέον λειτουργικότητα και υπηρεσίες.

## 1.1 Δομή Διπλωματικής Εργασίας

Στα πλαίσια αυτής της διπλωματικής εργασίας θα μελετήσουμε την γλώσσα σήμανσης HTML5, η οποία και επιλέχτηκε για την υλοποίηση της εφαρμογής. Θα διερευνήσουμε πως μπορούμε να αξιοποιήσουμε τις νέες τεχνολογίες και χαρακτηριστικά που προσφέρει και πως μπορούμε να τα ενσωματώσουμε σε μια εφαρμογή για φορητές συσκευές.

Στον 1<sup>ο</sup> Κεφάλαιο θα αναφερθούμε γενικά για την HTML5 και τον τομέα του Mobile Application Development. Θα δούμε συνοπτικά τι άλλες τεχνολογίες υπάρχουν και ποια προβλήματα προκύπτουν από τα διαφορετικά λογισμικά αλλά και συσκευές. Επίσης γίνεται αναφορά στις τεχνολογίες που εισάγει η HTML5 αλλά και τα πλεονεκτήματα αυτών.

Στο 2<sup>ο</sup> Κεφάλαιο γίνεται αναλυτική παρουσίαση των νέων προτύπων που εισάγονται με την HTML5, των βασικών λειτουργιών που προσφέρουν αυτά καθώς και πως μπορούμε να τα χρησιμοποιήσουμε.

Στο 3<sup>ο</sup> Κεφάλαιο γίνεται λόγος για frameworks που στηρίζονται στην HTML5 και χρησιμοποιούνται συχνά στην ανάπτυξη εφαρμογών για κινητά τηλέφωνα και άλλες συσκευές. Παρουσιάζεται ο σκοπός τους κάθε framework αλλά και οι βασικές διευκολύνσεις και λειτουργικότητα που προσφέρουν.

Στο 4<sup>ο</sup> Κεφάλαιο παρουσιάζονται κάποιες βασικές μέθοδοι σχεδιασμού εφαρμογών για φορητές συσκευές και γίνεται αναφορά στην αρχιτεκτονική της εφαρμογής που αναπτύξαμε και οι μέθοδοι και σχεδιαστικές ιδέες που χρησιμοποιήθηκαν.

Στο 5<sup>ο</sup> Κεφάλαιο αναφερόμαστε σε τεχνικά θέματα υλοποίησης της εφαρμογής. Γίνεται εκτενής αναφορά στο πως μπορούμε να υλοποιήσουμε τα χαρακτηριστικά και τις υπηρεσίες που ενσωματώθηκαν στην εφαρμογή.

Στο 6<sup>ο</sup> Κεφάλαιο παρουσιάζουμε αναλυτικά την λειτουργικότητα της εφαρμογής και της εμπειρίας χρήσης του χρήστη. Η ανάλυση γίνεται και με εικόνες από την εφαρμογή για καλύτερη κατανόηση επεξήγηση.

Τέλος στο 7<sup>ο</sup> Κεφάλαιο αναφέρουμε τα συμπεράσματα στα οποία καταλήξαμε ύστερα από την όλη έρευνα και ανάλυση που κάναμε.

## 2 HTML5

Η HTML5 είναι μια cross-platform mark-up γλώσσα προγραμματισμού για ορισμό δομημένων εγγράφων. Δεν περιορίζεται από την συσκευή στην οποία τρέχει επόμενος είναι ανεξάρτητη από αυτή. Υπάρχει προς τα πίσω υποστήριξη με παλαιότερα HTML standards ενώ παράλληλα φέρνει νέα λειτουργικότητα αφαιρώντας χαρακτηριστικά που δεν χρειάζονται ή δεν χρησιμοποιούνται πλέον. Επίσης με την HTML5 πλέον έχουν και επισήμως καταγράψει κάποια standards που ήδη υποστηρίζονταν από διάφορους web browsers.[11]

Στις αρχές του 2000 το w3c (world wide web Consortium) ανέπτυξε τα πρότυπα XHTML (eXtensible HyperText Markup Language) και XForms ώστε να είναι το μέλλον του Web. Πολλά όμως χαρακτηριστικά αυτών των προτύπων δεν είχαν προς τα πίσω υποστήριξη ενώ είχαν και πολύ αυστηρά και χωρίς λάθη πρότυπα γραφής. Αυτό είχε ως αποτέλεσμα τα παραπάνω πρότυπα να δημιουργήσουν δυσαρέσκεια και να χαρακτηριστούν ως μη πρακτικά, οδηγώντας μια ομάδα web browser vendors και άλλους ενδιαφερομένους να εξελίξουν την υπάρχουσα html4 προσθέτοντάς της νέα χαρακτηριστικά διατηρώντας έτσι την προς τα πίσω υποστήριξη. Το w3c όμως δεν συμφώνησε με αυτή την ιδέα και ως αποτέλεσμα δημιουργήθηκε το WHATWG (Web Hypertext Application Technology Working Group) το οποίο πρόκειται για ένα χαλαρό, μη επίσημο και ανοικτό πλαίσιο συνεργασίας των κατασκευαστών web browsers και τρίτων ενδιαφερομένων, όπως ειπώθηκε από το ίδιο το group.[11]

Μετά από αρκετά χρόνια όμως, με την υποστήριξη των browser vendors και της κοινότητας web development η δουλειά του WHATWG έγινε τελικά δεκτή από το w3c και έτσι ξεκίνησε μια προσπάθεια ενοποίησης των διαφόρων χαρακτηριστικών ώστε να τυποποιηθεί η HTML5. Υπάρχουν βέβαια ακόμα διαφορές μεταξύ των χαρακτηριστικών των w3c και WHATWG που προσθέτουν στο κύριο πρότυπο αλλά ο κύριος στόχος τους είναι να αναπτύξουν το πρότυπο μαζί με τους browser vendors και να παίρνουν feedback χρήσης παράλληλα με την δημιουργία του προτύπου. Αυτό έχει ως αποτέλεσμα πολλά χαρακτηριστικά να υπάρχουν στις νέες εκδόσεις των web browser αλλά να μην έχουν ακόμα συμπεριληφθεί στο επίσημο πρότυπο της HTML5. Αυτό βέβαια έχει και ένα μειονέκτημα καθώς για το ίδιο χαρακτηριστικό μπορεί να υπάρχουν διαφορετικές υλοποιήσεις σε διαφορετικούς web browsers.[11] [7]

Σε αυτή την διπλωματική εργασία θα ασχοληθούμε με την HTML5 πέρα από το πρότυπο όπως αυτό ορίζεται από την w3c και θα δούμε χαρακτηριστικά όπως αυτά έχουν υλοποιηθεί στους web browsers που ακόμα μπορεί να μην έχουν συμπεριληφθεί επίσημα στο πρότυπο. Πρόκειται για μια πρακτική ματιά στην ανάπτυξη εφαρμογών και σελίδων σε HTML5 με έμφαση στην υλοποίηση για κινητές συσκευές.

## 2.1 Semantic Markup

Η HTML5 ορίζει διάφορα νέα elements και attributes. Το θέμα είναι να γίνει η markup πιο σημασιολογική για τους προγραμματιστές, τους διαχειριστές περιεχομένου όπως μηχανές αναζήτησης και screen readers. Το πρότυπο στοχεύει σε μια πιο σημασιολογική δομή της HTML απορρίπτοντας πολλά χαρακτηριστικά σχετικά με την παρουσίαση. Η λογική πίσω από αυτό εξηγείται παρακάτω:

- **Media-independent markup** ώστε να δουλεύει για περισσότερους χρήστες και να υπάρχει καλύτερη προσβασιμότητα
- **Style-independent markup** ώστε να υπάρχει διαχωρισμός μεταξύ της δομής του εγγράφου και του layout του και είναι πιο εύκολη η συντήρησή του.
- **Διαχωρισμός του σχεδιαστικού αποτελέσματος** σε μικρότερου μεγέθους έγγραφα.

Κάθε στοιχείο (element) στην HTML5 μπορεί να ανήκει σε καμία ή περισσότερες κατηγορίες στοιχείων που έχουν παρόμοια χαρακτηριστικά [26]:

- **Metadata content:** περιεχόμενο που καθορίζει την συμπεριφορά του εγγράφου, ορίζει την σχέση του εγγράφου με άλλα έγγραφα ή περιέχει άλλες πληροφορίες του εγγράφου. Για παράδειγμα: link, meta, script, title.
- **Flow content:** κυρίως περιεχόμενο που χρησιμοποιείται στο body ενός εγγράφου. Όπως: a, article, audio, div, header, p κτλ.
- **Sectioning content:** περιεχόμενο που ορίζει την «σκοπιά» (scope) των headings και των footers. Για παράδειγμα: article, aside, nav, section
- **Heading content:** περιεχόμενο που ορίζει το header μια επιλογής. Για παράδειγμα: h1, h2, hgroup
- **Phrasing content:** περιεχόμενο που περιέχει μέσα του κείμενο του εγγράφου. Όπως: abbr, audio, canvas, img, em
- **Embedded content:** περιεχόμενο που εξάγει στο έγγραφο άλλα αντικείμενα όπως: audio, embed, iframe, img
- **Interactive content:** περιεχόμενο που προορίζεται για interaction με τον χρήστη όπως: a, button, menu, select

## 2.2 Media

Η υποστήριξη πολυμέσων είναι πολύ σημαντική για τις σημερινές εφαρμογές. Η HTML5 ορίζει elements και APIs για audio, video, subtitles και embedded περιεχόμενο.

Παλαιότερα για να μπορέσει κάποιος προγραμματιστής να χρησιμοποιήσει rich media περιεχόμενο έπρεπε να βασιστεί σε plugins τρίτων και σε extensions των browsers. Το να μην χρειάζεται να στηριζόμαστε σε plugins και extensions είναι ένας από τους κύριους στόχους της HTML5 ώστε να βελτιωθεί η «ανοικτότητα» και η προσβασιμότητα στο web περιεχόμενο.

### Κώδικας 1 - Video Element

```
<video src="vid.mp4"> </video>
```

## 2.3 Canvas 2D Context

Η HTML5 ορίζει το canvas element. Πρόκειται για ένα bitmap canvas που εξαρτάται από την ανάλυση που έχει και χρησιμοποιείται για την δυναμική σχεδίαση γραφικών. Μπορεί να χρησιμοποιηθεί δηλαδή για διαγράμματα, παιχνίδια και άλλα γραφικά.

Το πρότυπο του Canvas 2D Context ορίζει ένα JavaScript API για την προγραμματιστική σχεδίαση πάνω στη 2D canvas επιφάνεια. Το API ορίζει μεθόδους για τον σχεδιασμό σχημάτων, κειμένου, διαβαθμίσεων και εικόνων πάνω στον καμβά και άλλες μεθόδους για την διαχείριση των δεδομένων του bitmap [27].

Παρακάτω βλέπουμε ένα παράδειγμα εισόδου περιεχομένου στο canvas element με JavaScript.

### Κώδικας 2 - Canvas

```
<canvas id="new_canvas"></canvas>

<script>
var s='new_canvas';
var canv=document.getElementById(`s`);
var ctx=canv.getContext(`2d`);
var img=new Image();
img.src="html5.gif"
img.onload=function () {
ctx.drawImage(img,0,0);
}</script>
```

## 2.4 Form Enhancements

Οι φόρμες είναι πολύ σημαντικό κομμάτι για την κατασκευή διαδραστικών HTML εγγράφων. Ωστόσο εξαιτίας της απλότητας του και της περιορισμένης προσβασιμότητας και χρηστικότητας τους οι προγραμματιστές έπρεπε να καταφεύγουν σε προγραμματιστικές λύσεις σε JavaScript για να εμπλουτίσουν τις φόρμες και να βελτιώσουν αυτά τα προβλήματα.

Η HTML5 φέρνει πολλές βελτιώσεις-πρόσθετα στις φόρμες. Νέα είδη εισόδων για αριθμούς, ημερομηνίες, διευθύνσεις email κτλ κατάργησαν την ανάγκη για προγραμματιστικές λύσεις μιας και πλέον παρέχονται native εργαλεία [26]. Νέα χαρακτηριστικά για φόρμες όπως το placeholder και autofocus φέρνουν βελτιώσεις στην προσβασιμότητα και στην χρηστικότητα και μειώνουν την ανάγκη για προγραμματισμό.

Τέτοιες βελτιώσεις και προσθέσεις δουλεύουν πολύ καλά στο πλαίσιο των κινητών συσκευών όπου η είσοδος από τον χρήστη συνηθίζεται να είναι αργή και προβληματική. Για παράδειγμα έχοντας ειδικό input field για νούμερα επιτρέπει στο λειτουργικό σύστημα της συσκευής να ανοίξει κατευθείαν το πληκτρολόγιο των αριθμών με αποτέλεσμα ο χρήστης να μπορεί να παρέχει την είσοδο πολύ πιο γρήγορα και αποτελεσματικά βελτιώνοντας έτσι την χρηστικότητα. Επίσης ο έλεγχος για λάθη στην φόρμα από την μεριά του client (του χρήστη δηλαδή) μειώνει την ανάγκη για ανανέωση της σελίδας μιας και ο browser μπορεί να δείχνει μηνύματα λάθους σε φόρμες κατευθείαν χωρίς να χρειαζόμαστε JavaScript κώδικα.

## 2.5 Cascading Style Sheets

Με το CSS 3 εισάγονται πολλά νέα χαρακτηριστικά για σχεδιασμό της εμφάνισης του web application. Τα καλά διαχωρισμένα επίπεδα εξωτερικής εμφάνισης διατηρούν την δομή του εγγράφου καθαρή και οργανωμένη ενώ οι επιλογές για πλούσιο σχεδιασμό της εμφάνισης και οι παρεχόμενες δυνατότητες για εφέ μειώνουν τις ανάγκες για επιπλέον κώδικα και παρέχουν και fallback λειτουργία για τους παλαιότερους browsers. Έτσι με το να διαχειρίζεται, για παράδειγμα, ο browser την αναπαράσταση των animations του περιβάλλοντος χρήστη ο προγραμματιστής μπορεί να εστιάσει στην βελτιστοποίηση της αποκρισιμότητας και της απόδοσης της εφαρμογής μιας και ο browser μπορεί να χρησιμοποιήσει τις βέλτιστες τεχνικές διαχείρισης [15]. Παρακάτω θα παρουσιάσουμε κάποια βασικά στοιχεία.

## 2.5.1 Selectors

Τα CSS3 selectors είναι πρότυπα που μας δίνουν την δυνατότητα να επιλέξουμε τα στοιχεία που θέλουμε από το DOM δέντρο. Έτσι μπορούμε να εφαρμόσουμε σε αυτά συγκεκριμένο style χωρίς να επηρεάζονται τα υπόλοιπα elements ενώ είναι δυνατή η χρησιμοποίησή τους και από JavaScript για επιλογή στοιχείων. Με τα Selectors μπορούμε να δημιουργήσουμε πλούσιες αλληλεπιδραστικές εφαρμογές και να μειώσουμε τον κώδικα που χρειαζόμαστε για την επιλογή elements. Επίσης είναι πολύ σημαντικά και για την βελτιστοποίηση την απόδοσης. [51]

## 2.5.2 Transforms

Με τα 2D και 3D Transforms μπορούμε να «μεταφέρουμε» τα elements στον δισδιάστατο και τρισδιάστατο χώρο. Τα στοιχεία μπορούν να μετακινηθούν στον άξονα x, y να περιστραφούν αλλά και να μεγεθυνθούν ή σμικρυνθούν. Το πρότυπο ορίζει πολλές 2D και 3D λειτουργίες μεταμόρφωσης που μπορούν να χρησιμοποιηθούν.

Τα Transforms αποτελούν πολύ σημαντικό κομμάτι για την δημιουργία του user interface καθώς μπορούμε να δημιουργήσουμε διάφορα effects και προηγμένα διαδραστικά γραφικά. Συνδυάζοντας τα με animations (καθορισμένα στον χρόνο) ή transitions μπορούμε να δημιουργήσουμε πλούσια περιβάλλοντα χρήστη με CSS κανόνες.[23]

## 2.5.3 Transitions

Με τα CSS Transitions μπορούμε να αλλάξουμε το style των elements ορίζοντας κάποιο χρονικό διάστημα στο οποίο θα γίνει η αλλαγή. Έτσι μπορούν να χρησιμοποιηθούν για απλά animations. Συνήθως όταν μια ιδιότητα CSS αλλάξει η αλλαγή φαίνεται στιγμιαία, με τα transitions όμως μπορούμε αυτή η αλλαγή να γίνει προοδευτικά στο χρόνο ενώ παράλληλα να κάνουμε και κάποιες ρυθμίσεις σε αυτή. Κάτι τέτοιο είναι ιδιαίτερα χρήσιμο για effects παρουσίασης.[32]

## 2.5.4 Animations

Για πιο πολύπλοκα animations χρησιμοποιούμε τα CSS Animations Αυτά ορίζουν τα “keyframes” τα οποία χρησιμοποιούνται ώστε να ορίσουμε την πρόοδο του animation μεταξύ της έναρξης του και του τέλους του. Μπορούμε να ορίσουμε κάποια animations να επαναλαμβάνονται όσες φορές τα ορίσουμε, να εναλλάσσονται μεταξύ έναρξης και τέλους, να διαχειριζόμαστε την κατάσταση εκτέλεσης και προσωρινής διακοπής τους, να καθυστερήσουμε την έναρξη τους κ.α. [33]

## 2.5.5 Media Queries

Παλαιότερες εκδόσεις της HTML και του CSS ήδη υποστήριζαν την οργάνωση των stylesheets και κανόνων σε συγκεκριμένους media τύπους όπως 'screen', 'print' ή 'mobile'. Τα Media Queries μπορούν να χρησιμοποιηθούν για την ανίχνευση του μεγέθους και των διαστάσεων της συσκευής, το orientation, το aspect ratio, το color depth, κ.α. Η ανίχνευση αυτών των χαρακτηριστικών είναι ιδιαίτερα χρήσιμη όταν χρησιμοποιούμε την ίδια HTML markup για διαφορετικούς τύπους συσκευών. Για παράδειγμα το layout μιας web εφαρμογής μπορεί να χρησιμοποιεί περισσότερο οριζόντιο χώρο σε κάποιο desktop browser ενώ σε mobile συσκευές θα πρέπει να έχει κατακόρυφη οριοθέτηση και να μην επιτρέπεται το οριζόντιο scrolling. Επίσης και οι background εικόνες θα μπορούν να αλλαχθούν σε μικρότερες ώστε να ταιριάζουν καλύτερα σε μικρότερες οθόνες. [50]

## 2.5.6 Web Fonts

Στο CSS3 υποστηρίζεται νέα μέθοδος για δυναμική «φόρτωση» custom fonts και ορισμό των ιδιοτήτων τους. Πέρα από τα ήδη υποστηριζόμενα web fonts. [22]

## 2.6 Page Visibility και Timer Control

Το να μπορούμε να αποφύγουμε κάτι το οποίο δεν είναι απαραίτητο είναι μια σημαντική ιδέα σχετικά με την βελτιστοποίηση της απόδοσης. Οι περισσότεροι μοντέρνοι browsers έχουν ένα tabbed interface με , πιθανότατα, πολλές σελίδες ανοιγμένες ταυτόχρονα. Επιπλέον, πολλές λειτουργίες του interface χρησιμοποιούν animations και effects για να βελτιώσουν την εμπειρία χρήστη. Αυτό όμως έχει ως αποτέλεσμα, αν αυτές οι σελίδες δεν χρειάζεται να είναι ορατές για κάποια χρονικά διαστήματα, να μην έχουν ιδιαίτερη αξία την συγκεκριμένη στιγμή και να σπαταλούν πόρους του συστήματος χωρίς σημαντικό λόγο.

Το πρότυπο Page Visibility ορίζει ένα JavaScript API ώστε να μπορούμε να γνωρίζουμε πότε το έγγραφο είναι ορατό και πότε αυτό αλλάζει την κατάσταση ορατότητας του μέσω του «visibilitychange» [37]. Επίσης με το Timing control πρότυπο για animations που βασίζονται σε scripts [44] και με το πρότυπο Efficient Script Yielding έχουμε την δυνατότητα να διαχειριστούμε την ουρά των events ώστε να έχουμε αποδοτικότερα animations καθώς και διαχείριση των interface events [39]. Ένα παράδειγμα είναι η «requestAnimationFrame» με την οποία ζητάμε από τον browser να εκτελεστεί κάποιο animation με την διαφορά ότι θα εκτελεστεί πιο αποδοτικά καθώς λαμβάνονται υπόψη παράμετροι όπως αν η σελίδα είναι ενεργή, αν το συγκεκριμένο animation είναι ορατό εκείνη την στιγμή κ.α. Έτσι εξοικονομούμε



και υπολογιστική δύναμη αλλά και μπαταρία σε περίπτωση φορητών συσκευών και γενικότερη βελτίωση της απόδοσης.

## 2.7 Session History

Η HTML αρχικά σχεδιάστηκε ώστε να βασίζεται σε έγγραφα και hyperlinks που δείχνουν σε άλλα έγγραφα με καθένα από αυτά να έχει το δικό του μοναδικό URL (Uniform Resource Locator). Αυτή όμως η δομή δεν είναι βολική για web εφαρμογές που περιέχουν δυναμικό περιεχόμενο και που το περιβάλλον χρήστη αλλάζει ανάλογα την διαπεραστικότητα.

Δυο βασικές λειτουργίες που συνηθίζουν οι χρήστες να κάνουν είναι να αποθηκεύουν σελιδοδείκτες και να πηγαίνουν μπρος και πίσω στο ιστορικό της σελίδας που βρίσκονται (session history). Τέτοιες λειτουργίες συνήθως υλοποιούνταν σε δυναμικές Ajax εφαρμογές και συνήθως χρειάζεται περισσότερη δουλειά για να ολοκληρωθούν.

Η HTML5 λύνει αυτά τα προβλήματα επιτρέποντας έτσι τους προγραμματιστές να διαχειρίζονται δυναμικά το session history. Το ιστορικό μπορεί πλέον να το αλλάξουμε και να το χρησιμοποιήσουμε για πλοήγηση ενώ μπορούμε να αλλάξουμε ακόμα και την διεύθυνση στην address bar χωρίς να χρειαστεί ανανέωση της σελίδας.

## 2.8 Drag and Drop

Το Drag and Drop είναι μια κοινή τεχνική διαδραστικότητας όπου τα elements μπορούν να μετακινηθούν μέσα στο User Interface από το ένα μέρος στο άλλο. Παλαιότεροι browsers είχαν υλοποιήσει δικές τους λύσεις για την συγκεκριμένη λειτουργία αλλά με την HTML5 προτυποποιείται το API.

Το πρότυπο ορίζει χαρακτηριστικά των elements και DOM (Document Object Model) events για εύκολη ενεργοποίηση και διαχείριση των μετακινήσιμων elements και των drop targets [5]. Αυτή η λειτουργία υπήρχε και παλαιότερα βασισμένη σε cross-browser JavaScript λύσεις αλλά με το HTML5 API πλέον χρειαζόμαστε πολύ λιγότερο κώδικα JavaScript. Ο browser διαχειρίζεται την διαδραστικότητα και την δυναμική σχεδίαση, μειώνοντας έτσι την καθυστέρηση του interface και την ανάγκη για περισσότερη επεξεργασία [8].

## 2.9 SVG and MathML

Με την HTML5 είναι δυνατή η ενσωμάτωση SVG (Scalable Vector Graphics) και η MathML (Mathematical Markup Language) markup στην HTML.

Η SVG είναι μια markup γλώσσα με την οποία μπορούμε να περιγράψουμε δυο διαστάσεων διανύσματα γραφικών σε XML (Extensible Markup Language). Η γλώσσα σήμανσης μπορεί να χρησιμοποιηθεί με το DOM API για την δημιουργία δυναμικών και διαδραστικών λειτουργιών σε εφαρμογές [16]. Στην HTML5 μπορούμε να ενσωματώσουμε την SVG σήμανση μέσα στο `svg` στοιχείο.

Η MathML είναι μια XML γλώσσα σήμανσης για την περιγραφή της δομής και του περιεχομένου μαθηματικών τύπων. Μπορεί να ενσωματωθεί σε ένα HTML5 έγγραφο με το `math` στοιχείο [21].

Και οι δυο αυτές γλώσσες βοηθούν στο να μειωθεί η ανάγκη για δημιουργία ειδικά προσαρμοσμένων εικόνων κάνοντας έτσι το περιεχόμενο πιο προσβάσιμο, δυναμικό και κάνοντας δυνατή την αλληλεπίδραση με αυτό. Επίσης τα διανυσματικά γραφικά μπορούν να προσαρμοστούν στον διαθέσιμο χώρο ανεξάρτητα από το μέγεθος της οθόνης πράγμα που βοηθά στην cross-platform χρήση για διαφορετικές συσκευές.

## 2.10 WebGL and Typed Arrays

Το WebGL (Web Graphics Library) είναι ένα low-level 3D rendering API το οποίο προέρχεται από το OpenGL ES 2.0 (Open Graphics Library for Embedded Systems) και σχεδιάστηκε ως ένα γενικό πλαίσιο σχεδιασμού γραφικών για το canvas element της HTML5. Ένα διαδραστικό API για 3D γραφικά είναι πολύ σημαντικό για την ανάπτυξη παιχνιδιών και προσφέρει έτσι και μια πλατφόρμα για cross-platform παιχνίδια [40]. Το WebGL αναπτύχθηκε από το Kronos Group και στην συνέχεια συμμετείχαν browser vendors και προγραμματιστές 3D εφαρμογών.

Βασιζόμενο λοιπόν στο OpenGL ES API το WebGL μπορεί να τρέξει σε πολλές διαφορετικές συσκευές όπως επιτραπέζιους υπολογιστές, κινητά τηλεφώνά, tablets και τηλεοράσεις. Επίσης η χρησιμοποίησή του δεν περιορίζεται στα παιχνίδια μόνο αλλά μπορεί να χρησιμοποιηθεί και σε 3D μοντελοποίηση, σχεδιαστικά εργαλεία, εξομοιώσεις, αστικοποίηση δεδομένων, διαδραστική τέχνη κ.α.

Προερχόμενα από το WebGL τα typed arrays έχουν αποκτήσει το δικό τους πρότυπο [25]. Στην κλασική JavaScript δεν υπάρχουν τύποι πινάκων ενώ στο WebGL API χρειαζόμαστε typed arrays καθώς είναι πιο αποδοτικά σαν δομές δεδομένων. Φυσικά μπορούν να χρησιμοποιούνται και σε άλλο πλαίσιο περά των γραφικών, όπου χρειάζεται η επεξεργασία μεγάλου μεγέθους binary δεδομένων.

## 2.11 Touch Events

Παραδοσιακά τα events στο περιβάλλον χρήστη μιας web εφαρμογής προερχόταν από κάποια συσκευή εισόδου όπως το ποντίκι ή το πληκτρολόγιο. Σήμερα όμως με τα μοντέρνα κινητά τηλέφωνα και τα tablets συνήθως χρησιμοποιούμε οθόνες αφής. Στα

touch events αρχικά αντιστοιχήθηκαν κάποια pointer και keyboard events αλλά χωρίς εξειδικευμένα touch events δεν ήταν δυνατό να αξιοποιηθούν όλες οι δυνατότητες της αλληλεπίδρασης με την είσοδο αφής.

Το Touch Events πρότυπο ορίζει ένα σετ από events για ένα ή περισσότερα σημεία αφής στην επιφάνεια αφής. Το πρότυπο ορίζει τα touchstart, touchmove, touchend, touchcancel events και διάφορα νέα χαρακτηριστικά για το κάθε event [19]. Όλα αυτά δίνουν την δυνατότητα στους προγραμματιστές να προσφέρουν επιπλέον λειτουργικότητα και κάνοντας πιο πλούσια την αλληλεπίδραση με την εφαρμογή χρησιμοποιώντας swipes και multi-touch gestures.

Παρακάτω βλέπουμε ένα παράδειγμα ορισμού των events handler και τις πληροφορίες που μας παρέχουν.

### Κώδικας 3 - Touch Events

```
" element ". addEventListener ( " touchstart ", startHandler , false );  
" element ". addEventListener ( " touchmove ", moveHandler , false );  
" element ". addEventListener ( " touchend ", endHandler , false );  
  
e. touches [0]. pageX  
e. touches [0]. pageY
```

## 2.12 Offline Web Applications

Οι ιστοσελίδες λόγω του σχεδιασμού τους χρειάζονται μια ενεργή σύνδεση στο διαδίκτυο. Οι εφαρμογές όμως θα πρέπει να μπορούν να δουλεύουν και εκτός σύνδεσης ή σε συνδέσεις που δεν είναι σταθερές. Ειδικά όταν έχουμε να κάνουμε με κινητά δίκτυα που δεν μπορούμε να έχουμε μια σταθερή σύνδεση συνεχώς φαίνεται ακόμα περισσότερο η ανάγκη για υποστήριξη εκτός σύνδεσης λειτουργία στην HTML5.

Υπάρχουν διάφοροι τρόποι ώστε να ενεργοποιήσουμε την υποστήριξη για εκτός σύνδεσης λειτουργία σε HTML5 εφαρμογές. Παρακάτω θα παρουσιάσουμε τα χαρακτηριστικά που μπορούμε να χρησιμοποιήσουμε.

### 2.12.1 Application Cache

Η AppCache (Application Cache) είναι ένας σχετικά εύκολος τρόπος ώστε να ορίσουμε όλα τα αρχεία που χρειάζεται μια εφαρμογή ώστε να μπορεί να δουλεύει και εκτός σύνδεσης [5]. Ένα manifest αρχείο ορίζεται στο HTML έγγραφο και μέσα σε αυτό το manifest αρχείο υπάρχουν εγγραφές με τα περιεχόμενα/αρχεία που είτε πρέπει πάντα να αποθηκεύονται τοπικά ή να μην αποθηκεύονται ποτέ, υπάρχουν εγγραφές για fallback URLs για αρχεία που δεν έχουν αποθηκευτεί τοπικά αλλά η ανάκτηση τους έχει αποτύχει. Επίσης πέρα από αυτό το manifest αρχείο υπάρχουν και

JavaScript events που μας επιτρέπουν να διαχειριστούμε τις τυχόν αλλαγές που μπορεί να γίνουν στην cache.

Ένα παράδειγμα του manifest αρχείου φαίνεται παρακάτω:

#### Κώδικας 4 - AppCache Manifest

```
CACHE MANIFEST
# 2010-06-18:v2

# Explicitly cached 'master entries'.
CACHE:
/favicon.ico
index.html
stylesheet.css
images/logo.png
scripts/main.js

# Resources that require the user to be online.
NETWORK:
login.php
/myapi
http://api.twitter.com

# static.html will be served if main.py is inaccessible
# offline.jpg will be served in place of all images in images/large/
# offline.html will be served in place of all other .html files
FALLBACK:
/main.py /static.html
images/large/ images/offline.jpg
*.html /offline.html
```

## 2.12.2 Data Storage

Συνήθως όταν θέλαμε να αποθηκεύσουμε κάποια δεδομένα στην μεριά του client χρησιμοποιούσαμε cookies. Αυτό όμως με την HTML5 έχει βελτιωθεί καθώς ορίζονται νέες επιλογές για την διατήρηση δεδομένων.

Ορίζονται λοιπόν δυο νέες μέθοδοι αποθήκευσης της μορφής κλειδιού-τιμής (key/value) [5]. Αυτά είναι το localStorage και το sessionStorage [30]. Το API είναι ίδιο και για τα δυο απλά στο sessionStorage τα δεδομένα είναι διαθέσιμα μόνο για συγκεκριμένο browser session. Αυτά τα δυο χαρακτηριστικά είναι πολύ εύκολα στην χρήση αλλά έχουν τον περιορισμό ότι μπορούμε να αποθηκεύσουμε μόνο δεδομένα κειμένου.

Οι τιμές αποθηκεύονται σαν αλφαριθμητικά και με την μέθοδο setItem, που παίρνει για παραμέτρους το κλειδί και την τιμή του. Η ανάκληση γίνεται με την μέθοδο getItem που παίρνει το κλειδί[15].

#### Κώδικας 5 - localStorage

```
var str = "Some data";
```

«Σχεδίαση και Υλοποίηση Εφαρμογής για Κινητά Τηλέφωνα Αξιοποιώντας τις Πληροφορίες των Αισθητήρων τους»

```
localStorage.setItem("key1", str);  
// ...  
str = localStorage.getItem("key1");
```

Όπως και όλα τα αντικείμενα της javascript είναι δυνατή η χρήση του αντικειμένου localStorage σαν να είναι πίνακας.

#### Κώδικας 6 - localStorage Object

```
var str = "Some data";  
localStorage["key1"] = str;  
// ...  
str = localStorage["key1"];
```

Επιπλέον παρέχονται οι συναρτήσεις removeItem για την διαγραφή ενός αντικειμένου, clear για το άδειασμα όλων των δεδομένων και key για επιστροφή ενός συγκεκριμένου κλειδιού. Το πλήθος των αποθηκευμένων αντικειμένων δίνεται από την ιδιότητα length.

#### Κώδικας 7 - localStorage Delete & Length

```
localStorage.removeItem("key1") // delete key1  
localStorage.clear() // delete all data  
FirstKey = localStorage.key(0) // get the 1st key  
var size = localStorage.length
```

Στην πρόταση του W3C για το Web Storage προτείνονται τα 5MB σαν όριο για το μέγεθος των δεδομένων που μπορεί να αποθηκεύσει κάθε σελίδα και αυτό ακολουθείται από όλους τους φυλλομετρητές. Σε περίπτωση που το όριο ξεπεραστεί τότε η τρέχουσα λειτουργία σταματάει και εμφανίζεται ένα λάθος τύπου QUOTA\_EXCEEDED\_ERR.

Παρόλο που το όριο των 5MB είναι μία πρόταση και δεν είναι δεσμευτικό, αυτή τη στιγμή δεν παρέχεται η δυνατότητα ερώτησης του χρήστη ώστε μία σελίδα να χρησιμοποιήσει περισσότερο χώρο στο δίσκο.

Επίσης έχουν οριστεί και δυο ακόμη storage APIs: η client side SQL database και η Indexed Database [30]. Για την webSQL υπάρχει ένα ασύγχρονο και transactional JavaScript API για SQL βάσεις δεδομένων. Παρόλο που είναι πολύ εκφραστική μιας και βασίζεται σε key/value storage δεν είναι ακόμα σίγουρο αν τελικά θα προτιμηθεί από τους browser vendors και τους προγραμματιστές μιας και φαίνεται μια στροφή προς την indexedDB. Κάθε βάση θα μπορεί να έχει έναν και μοναδικό αριθμό έκδοσης μία δεδομένη στιγμή και έτσι δεν θα μπορούν να υπάρχουν ταυτόχρονα διαφορετικές εκδόσεις της ίδιας βάσης. Η δυνατότητα για αρίθμηση της έκδοσης μίας βάσης δεδομένων που χρησιμοποιείται από μία εφαρμογή, στοχεύει στο να δίνει τη δυνατότητα στον σχεδιαστή της εφαρμογής να διαχειρίζεται με ευκολία τις αλλαγές στο σχήμα της βηματικά και μη καταστροφικά. Έτσι μπορεί να αποφευχθεί με

μεγαλύτερη ευκολία η πιθανότητα να εκτελείται κώδικας προηγούμενης έκδοσης και λόγω του περιορισμού σχετικά με την ταυτόχρονη ύπαρξη διαφορετικών εκδόσεων, αποκλείεται το ενδεχόμενο δύο διαφορετικά παράθυρα για παράδειγμα να χρησιμοποιούν διαφορετικές βάσεις και διαφορετικό κώδικα.

Για την indexed βάση δεδομένων έχουμε στην διάθεση μας ένα σύγχρονο και ένα ασύγχρονο API για την αποθήκευση και ανάκτηση μεγάλων ποσοτήτων δομημένων δεδομένων [41]. Επίσης με την indexedDB μπορούμε να ικανοποιήσουμε πιο πολύπλοκες ανάγκες για αποθήκευση και αναζήτηση δεδομένων σε σχέση με τους τρόπους αποθήκευσης κλειδιού-τιμής της webSQL.

### 2.12.3 Κατάσταση Συνδεσιμότητας

Το να γνωρίζουμε αν ο χρήστης είναι online η offline είναι πολύ χρήσιμο καθώς μπορούμε να καθορίσουμε πως μεταβάλλεται το User Interface ή πως απαντά η εφαρμογή σε κάποιες ενέργειες του χρήστη που σχετίζονται με την συνδεσιμότητα. Η HTML5 παρέχει λειτουργικότητα για να ανιχνεύει την κατάσταση συνδεσιμότητας καθώς και events που ενεργοποιούνται κατά την αλλαγή κατάστασης του δικτύου.

Παρόλο που αυτό το στοιχείο παρέχει σημαντικές πληροφορίες για την κατάσταση του δικτύου, αυτές οι πληροφορίες δεν είναι πάντα αξιόπιστες. Λόγω της φύσης των κινητών δικτύων η εφαρμογή δεν είναι δυνατό να γνωρίζει πάντα την ακριβή κατάσταση της συνδεσιμότητας. Η μόνη επιλογή για την ακριβή ανίχνευση είναι να κάνουμε requests και να περιμένουμε για την επιτυχή απάντηση ή την αποτυχία απάντησης.

Γι αυτό οι εφαρμογές πρέπει να κατασκευάζονται ώστε να λειτουργούν online αλλά να μπορούν και να ανταποκριθούν ανάλογα όταν η σύνδεση χαθεί ή δεν είναι αναξιόπιστη.

## 2.13 Files

Η πρόσβαση σε τοπικό σύστημα αρχείων είναι πολύ σημαντική για τις εφαρμογές. Τα JavaScript APIs για την αποθήκευση δεδομένων και τις βάσεις δεδομένων μπορούμε να τα χρησιμοποιήσουμε για δομημένα δεδομένα και για caching αλλά είναι προβληματικά για μεγάλα binary δεδομένα. Το File API ορίζει διεπαφές για δημιουργία, διάβασμα, γράψιμο και διαχείριση και επεξεργασία τοπικών αρχείων και φακέλων[47][46][43]. Επίσης στο πρότυπο ορίζονται και η διαχείριση λαθών και η ασφάλεια μέσω του sandboxed συστήματος.

Υπάρχουν δυο εκδόσεις διαχείρισης αρχείων στο API. Το synchronous (σύγχρονο) και το asynchronous (ασύγχρονο). Στο ασύγχρονο οι ενέργειες γίνονται το κύριο νήμα της εφαρμογής ενώ το σύγχρονο προορίζεται για χρήση σε Web Workers . Μπορούμε να διαχειριστούμε binary αρχεία και κείμενο στην μνήμη ή σαν Blob

«Σχεδίαση και Υλοποίηση Εφαρμογής για Κινητά Τηλέφωνα Αξιοποιώντας τις Πληροφορίες των Αισθητήρων τους»

(Binary Large Object) URLs με το DOM API. Με αυτές τις δυνατότητες μπορούμε να οργανώσουμε καλύτερα τις δικτυακές μεταφορές και την υποστήριξη χρήσης εκτός σύνδεσης για μεγάλα αρχεία. Συνδυάζοντας το File API με το Drag and Drop API και τους Web Workers, HTML5 φόρμες μπορούν να εμπλουτιστούν ακόμα περισσότερο και να έχουμε και καλύτερη απόδοση.

Παρακάτω βλέπουμε ένα απλό παράδειγμα του element και πως μπορούμε να διαχειριστούμε την είσοδό του με JavaScript

#### Κώδικας 8 - File Input Element

```
<input
  type="file" id="files"
  name="files[]" multiple/>

<script>
  function handleFileSelect(evt) {
    // FileList object
    var files = evt.target.files;

    // Loop through the FileList.
    for (var i = 0, f; f = files[i]; i++)
    {
      var reader = new FileReader();

      // Closure to capture
      // the file information.
      reader.onload = (function(theFile) {
        return function(e) {
          //e.target.result
          //theFile
        };
      })(f);
      // Read in the file as a data URL.
      reader.readAsDataURL(f);
    }
  }
</script>
```

## 2.14 Device APIs

Η κύρια διαφορά μεταξύ των web εφαρμογών και των native είναι η δυνατότητα χρησιμοποίησης των αισθητήρων που έχουν οι συσκευές. Στις επόμενες παραγράφους θα παρουσιάσουμε κάποια πρότυπα και τα APIs τους τα οποία και παρέχουν την πρόσβαση στις λειτουργίες της συσκευής από την JavaScript.

### 2.14.1 Geolocation

Το πλαίσιο (context) μέσα στο οποίο ανήκει μια προσωποποιημένη εφαρμογή είναι σημαντική έννοια και καθορίζεται κυρίως από την τοποθεσία του χρήστη.

Ωστόσο οι υλοποιήσεις για ανίχνευση της θέσεις που υπήρχαν για web sites είναι πολύ ανακριβής και κυρίως βασίζονταν στην IP.

Το Geolocation API ορίζει ένα JavaScript API για web εφαρμογές που δίνει την δυνατότητα να αναζητηθεί η τοποθεσία του χρήστη. Το API δεν παρέχει πληροφορίες για την πηγή αυτής της πληροφορίας αν και αυτή συνήθως προέρχεται είτε από το GPS (Global Positioning System) που υπάρχει στην συσκευή είτε από κάποιο WiFi (Wireless Local Area Network, WLAN) network του οποίου η τοποθεσία είναι γνωστή ή ακόμα και από τοποθεσία της κεραία κινητής τηλεφωνίας [42]. Το API παρέχει τις συντεταγμένες και την ακρίβεια αυτών.

Οι υλοποιήσεις στους browser συνήθως παρέχουν κάποιου είδους επιβεβαίωσης από τον χρήστη πρώτου επιτραπέι στην εφαρμογή να βρει την φυσική θέση του χρήστη [3].

Παρακάτω βλέπουμε απλό κώδικα με τον οποίο παίρνουμε τις συντεταγμένες της θέσης του χρήστη/συσκευής.

#### Κώδικας 9 - Geolocation

```
var navgeo=navigator.geolocation;
navgeo.getCurrentPosition(showPosition,onError);
//also monitor position as it changes
navgeo.watchPosition(showPosition);

function onError(error){error.code}

function showPosition(position) {
  var lat = position.coords.latitude;
  var lng = position.coords.longitude;
  ... }
```

## 2.14.2 Device Orientation

Η γνώση του φυσικού προσανατολισμού (orientation) της συσκευής και η αλλαγή αυτού δίνει την δυνατότητα για ανάπτυξη προηγμένων εφαρμογών πως για παράδειγμα διαδραστικά παιχνίδια με προηγμένους μηχανισμούς εισόδων πληροφορίας. Τυπικά η πηγή αυτής της πληροφορίας προέρχεται συνήθως από αισθητήρες όπως το γυροσκόπιο (gyroscope), την πυξίδα και το επιταχυνσιόμετρο (accelerometer).

Το πρότυπο Device Orientation ορίζει APIs και events για πρόσβαση στις πληροφορίες αυτών των αισθητήρων [18]. Οι πληροφορίες αυτές μπορούν να χρησιμοποιηθούν για πλούσια αλληλεπίδραση με τον χρήστη αλλά και για την δημιουργία gestures σε διάφορων ειδών εφαρμογές.

Παρακάτω βλέπουμε σε κώδικα ένα απλό παράδειγμα ανίχνευσης και πιθανής διαχείρισης της αλλαγής του προσανατολισμού της συσκευής.



### Κώδικας 10 - OrientationChange Event

```
window.addEventListener("orientationchange",  
    function() {  
        var h=window.innerHeight  
        var w=window.innerWidth  
        if (h>w) {  
            ...  
        }  
        else{  
            ...  
        }  
    }, false);
```

## 2.14.3 User Media

Το πρότυπο `getUserMedia` ορίζει τρόπους για να έχουμε πρόσβαση στα multimedia streams της συσκευής στην οποία τρέχει κάποια εφαρμογή. Τα streams μπορεί να είναι ήχος, βίντεο, εικόνα και η συσκευή από την οποία προέρχεται μπορεί να είναι για παράδειγμα μια web camera που είναι συνδεδεμένη σε έναν desktop υπολογιστή ή το μικρόφωνο σε ένα μια κινητή συσκευή [20].

Επίσης υπάρχουν και ειδικές επεκτάσεις του input form element για είσοδο από εφαρμογές εικόνας, βίντεο και ήχου από το λειτουργικό σύστημα. Ένα παράδειγμα του ειδικού input element φαίνεται παρακάτω.

### Κώδικας 11 - Camcorder Input File Element

```
<input type="file" accept="video/*" capture="camcorder"/>
```

## 2.15 Web Real-time Communication

Η υποστήριξη multimedia όπως αναπαραγωγή ήχου και εικόνας είναι πολύ σημαντική για διακρατικές πλούσιες εφαρμογές διαδικτύου. Ωστόσο η αναπαραγωγή και μόνο αρχείων δεν είναι αρκετή.

Το Web RT C API φέρνει την δυνατότητα για native media streaming μέσα σε HTML έγγραφα [17]. Έτσι πλέον δεν χρειάζεται να βασιζόμαστε σε 3<sup>rd</sup> party software όπως το Adobe Flash and RTMP (Real Time Messaging Protocol). Επίσης έχει οριστεί και ένα κανάλι επικοινωνίας για peer-to-peer streaming μεταξύ δυο user agents.

Σε συνδυασμό με το `getUserMedia` API μπορούμε να δούμε ή να εγγράψουμε κάποιο stream από τοπική πηγή όπως για παράδειγμα web camera [20]. Αυτά τα πρότυπα θέτουν και κάποια θέματα ασφαλείας αλλά υπόσχονται πολλές δυνατότητες σχετικές με multimedia για web εφαρμογές.

«Σχεδίαση και Υλοποίηση Εφαρμογής για Κινητά Τηλέφωνα Αξιοποιώντας τις Πληροφορίες των Αισθητήρων τους»

## 2.16 Web Sockets

Το Web Sockets API ορίζει ένα πρωτόκολλο επικοινωνίας δυο κατευθύνσεων, για εφαρμογές πραγματικού χρόνου μεταξύ ενός client όπως για παράδειγμα, ένας web browser, και ενός remote server [29]. Επειδή λόγω της φύσης του HTTP (Hypertext Transfer Protocol) η χρήση του σε πολύ διαδραστικές εφαρμογές προκαλούσε προβλήματα καθυστέρησης απάντησης και απόκρισης συνήθως η υλοποίηση τέτοιων εφαρμογών βασιζόταν σε πολύπλοκες λύσεις.

Η σύνδεση μέσω Web Socket μπορεί να είναι είτε «ανοικτή» είτε προστατευμένη όπως το HTTP και HTTPS (Hypertext Transfer Protocol Secure). Το πρωτόκολλο χρησιμοποιεί μια μοναδική TCP (Transmission Control Protocol) σύνδεση η οποία παραμένει ενεργή και επιτρέπει κίνηση και προς τις δυο κατευθύνσεις.

Το πρωτόκολλο επικοινωνίας ορίζει ένα επίπεδο πάνω από το TCP το οποίο και ορίζει το handshaking της HTTP σύνδεσης, το μοντέλο ασφαλείας που βασίζεται στο «origin», τον μηχανισμό διευθυνσιοδότησης και ονοματοδοσίας για πολλαπλές υπηρεσίες στο ίδιο port και πολλαπλά host names στην ίδια μοναδική IP (Internet Protocol), τον μηχανισμό για την αντιμετώπιση των περιορισμών στο μέγεθος των TCP πακέτων κ.α. Ο προορισμός αυτού των Web Sockets είναι να παρέχουν ένα απλό πρωτόκολλο που συνεργάζεται σωστά με το HTTP και την υπάρχουσα HTTP δομή και είναι όσο το δυνατόν πιο κοντά στο TCP ενώ παράλληλα λαμβάνονται υπόψη και τα θέματα ασφαλείας που προκύπτουν.

## 2.17 Server-Sent Events

Τα Server-Sent Events αποτελούν ένα πρότυπο το οποίο ορίζει ένα είδος data stream το text/event-stream το οποίο και μπορεί να χρησιμοποιηθεί για να συνδέσει έναν event listener από τη μεριά του client ώστε να ακούει σε events που στέλνονται από τον server [28]. Αυτά τα streams για παράδειγμα μπορούν να χρησιμοποιηθούν για να αποστέλλονται πραγματικού χρόνου push notifications για ενημερώσεις σχετικές με ανανέωση/ενημέρωση δεδομένων. Το API είναι πολύ απλό και την διαδικασία λήψης των μηνυμάτων την αναλαμβάνει ο browser απαλλάσσοντας έτσι τον προγραμματιστή από επιπλέον δουλειά.

## 2.18 Web Workers

Τα τελευταία χρόνια υπήρξε μια μεγάλη στροφή στον παράλληλο προγραμματισμό/υπολογισμό. Επεξεργαστές πολλών πυρήνων έχουν εμφανιστεί μέχρι και στις κινητές συσκευές όπως κινητά τηλέφωνα και tablets ενώ ο αριθμός των πυρήνων στις μοντέρνες CPUs (Central Processing Unit) όλο και αυξάνεται. Αυτός ο

παραλληλισμός της επεξεργασίας φέρνει νέες ευκαιρίες και δυνατότητες και στον κόσμο των εφαρμογών για κινητές συσκευές.

Έτσι λοιπόν ακολουθώντας τις τάσεις της βιομηχανίας υπολογιστών και σε συνδυασμό με την διάδοση των web τεχνολογιών οι εφαρμογές διαδικτύου έχουν αρχίσει να γίνονται όλο και πιο ικανές και παράλληλα και πιο απαιτητικές για επεξεργαστική δύναμη. Ωστόσο παρουσιάζοντας ένα κλασικό μοντέλο νημάτων (threading) και προσαρμόζοντας κατάλληλα τα browser APIs μπορεί να είναι υπερβολική λύση και δεν ταιριάζει και με την απλότητα αλλά και την προς τα πίσω συμβατότητα του Web.

Η JavaScript από τον σχεδιασμό της είναι single-threaded μαζί με ένα μοντέλο ασύγχρονων events. Επίσης και το user interface τρέχει στο ίδιο thread και γι αυτό αν κάποιο JavaScript script τρέχει για πολύ ώρα μπορεί να παγώσει ολόκληρο το UI κατά την επεξεργασία. Η τεχνική που χρησιμοποιείται είναι να χωρίζεται ο κώδικας σε μικρότερα κομμάτια και ο browser να διαχειρίζεται τον χρονοπρογραμματισμό των user interface events και της επεξεργασίας της JavaScript. Αυτό όμως το μοντέλο προγραμματισμού δεν είναι τόσο αποδοτικό και το πάγωμα του UI είναι δύσκολο να αποφευχθεί σε απαιτητικές στιγμές [14].

Τα Web Workers προτάθηκαν ως λύση για την παράλληλη επεξεργασία στην JavaScript. Παρουσιάστηκε μια απλή διεπαφή για sandboxed στοιχεία με περιορισμένη προσβασιμότητα στα APIs του browser και ένα κανάλι ασύγχρονης επικοινωνίας μεταξύ του worker και της κύριας εφαρμογής [31].

Τα Web Workers είναι εξωτερικά JavaScript αρχεία που εκτελούνται από ένα web site αλλά ο worker τρέχει σε ξεχωριστό ανεξάρτητο, από την κύρια εφαρμογή και UI, thread. Η μόνη σύνδεση με την κύρια εφαρμογή είναι το κανάλι επικοινωνίας μεταξύ τους.

Το API είναι απλό και εύκολο να δουλέψει κάποιος με αυτό και ενεργοποιείται πλέον και η δυνατότητα για παράλληλη επεξεργασία/προγραμματισμό στις web εφαρμογές.

## 2.19 Άλλα Πρότυπα

Υπάρχουν ακόμα πολλά πρότυπα και προτάσεις για επιπλέον χαρακτηριστικά συσκευών και πρόσβαση σε δεδομένα αισθητήρων. Παρακάτω υπάρχει μια σύντομη λίστα με μερικά από αυτά:

- **Web Notifications** [24] για την εμφάνιση απλών ειδοποιήσεων και ενημερώσεων στον χρήστη
- **Fullscreen** [48] για ενεργοποίηση και διαχείριση της δυνατότητας εμφάνισης των εφαρμογών σε πλήρη οθόνη

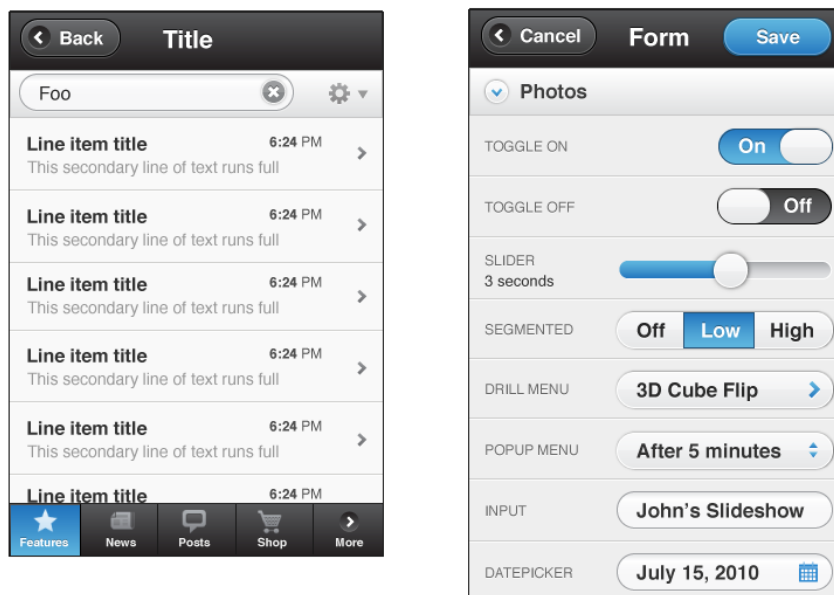
- **Pointer Lock** [45]για διαχείριση των εισόδων pointers, όπως για παράδειγμα σε χρήση σε παιχνίδια first-person shooter
- **Clipboard** για διαχείριση των ενεργειών copy, cut και paste
- **Battery Status** [35]για πρόσβαση σε πληροφορίες σχετικές με την κατάσταση της μπαταρίας της συσκευής
- **Vibration API** [34]για πρόσβαση στον μηχανισμό δόνησης της συσκευής
- **Network Information** [36]για πρόσβαση στις πληροφορίες σύνδεσης δικτύου της συσκευής
- **Contacts** για πρόσβαση στις επαφές που έχει αποθηκευμένες ο χρήστης στην συσκευή του
- **Navigation Timing** [49][38]για παροχή analytics των events που συμβαίνουν από την στιγμή που ο χρήστης περιηγείται στην εφαρμογή

## 3 HTML5 Frameworks

Στο παρόν κεφάλαιο θα παρουσιάσουμε κάποιες τεχνολογίες και frameworks τα οποία χρησιμοποιούνται πολύ συχνά στην ανάπτυξη εφαρμογών για κινητά τηλέφωνα (και όχι μόνο) και στηρίζονται σε νέα χαρακτηριστικά της HTML5. Τα εργαλεία αυτά βοηθούν στην υλοποίηση όμορφων και πιο δομημένων εφαρμογών ενώ κάνουν την αξιοποίηση χαρακτηριστικών τόσο της HTML5 αλλά και τις ιδίας της συσκευής πιο εύκολη μειώνοντας έτσι και τον χρόνο ανάπτυξης [7].

### 3.1 jQuery Mobile

Το jQuery Mobile<sup>1</sup> πρόκειται για ένα client-side framework για κινητές συσκευές και οθόνες αφής. Πρόκειται για μια βιβλιοθήκη περιβάλλοντος χρήστη που παρέχει τα κατάλληλα εργαλεία για την εύκολη και γρήγορη ανάπτυξη του περιβάλλοντος της εφαρμογής. Στηρίζεται στην HTML5 και στο jQuery JavaScript framework<sup>2</sup>. Ο στόχος αυτού του project είναι να παρέχει υποστήριξη σε όσο το δυνατόν περισσότερες συσκευές. Ένα απλό παράδειγμα του UI που μπορεί κάποιος να αξιοποιήσει φαίνεται στην Εικόνα 2 - jQuery Mobile user interface components.



Εικόνα 2 - jQuery Mobile user interface components

<sup>1</sup> <http://jquerymobile.com/>

<sup>2</sup> <http://jquery.com/>

Το jQuery Mobile είναι ανοικτού κώδικα και υποστηρίζεται από πολλές και μεγάλες εταιρείες. Στο user interface μπορούν να χρησιμοποιηθούν και θέματα ενώ υπάρχουν και πολλά extensions και widgets από τρίτους. Είναι αρκετά διαδεδομένο μιας και στηρίζεται στο πολύ γνωστό jQuery και έχει πολύ καλό documentation και community support.

## 3.2 jQT

Το jQT<sup>3</sup> (ή jQTouch) είναι ένα ελαφρύ plug-in που βασίζεται στο jQuery και χρησιμοποιείται για mobile web development σε συσκευές με iOS, Android, Blackberry και WebOS. Υποστηρίζει μόνο το WebKit browser engine όπως επίσης και παραμετροποιήσιμα θέματα εμφάνισης, διάφορα χαρακτηριστικά του user interface καθώς και τρόπους διαχείρισης της εισόδου αφής. Παρακάτω στην Εικόνα 3 βλέπουμε ένα παράδειγμα των περιεχομένων που μπορούν να υποστηριχτούν.



Εικόνα 3 - jQT User Interface Components

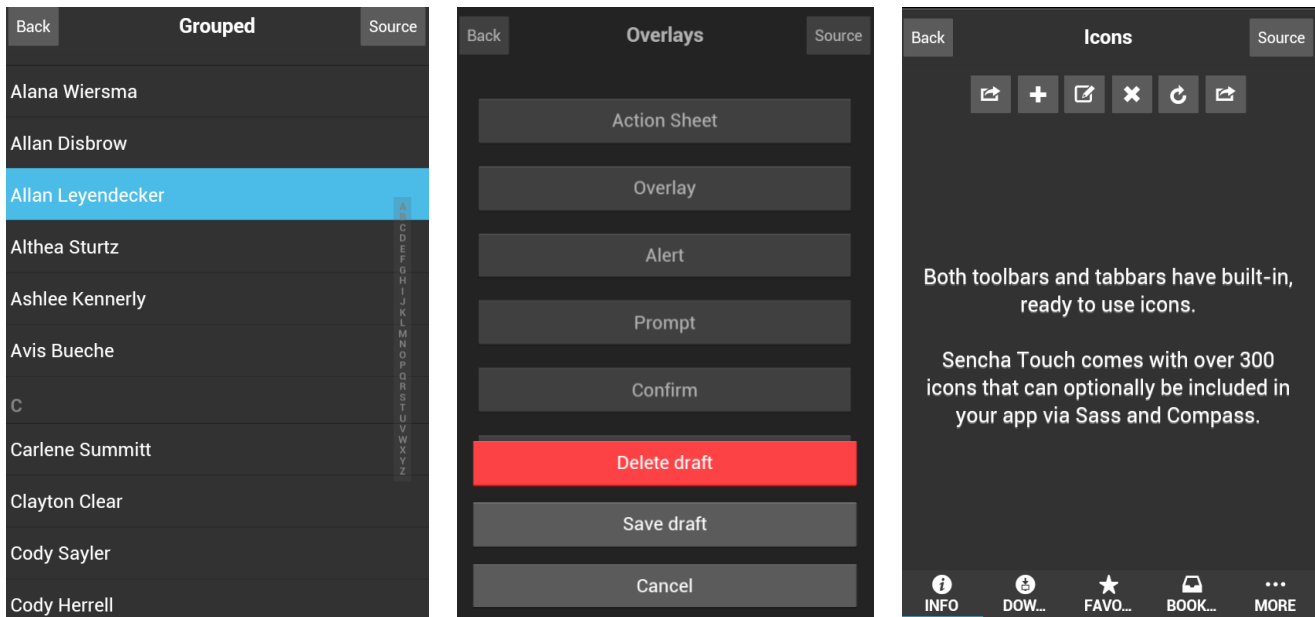
## 3.3 Sencha Touch

Το Sencha Touch<sup>4</sup> είναι ένα υψηλής απόδοσης framework ανοικτού κώδικα για ανάπτυξη HTML5 web εφαρμογών για Android, iPhone, Blackberry, Windows Phone και άλλες συσκευές. Παρέχει διάφορες δυνατότητες στην κατασκευή του user interface με διάφορα plug-ins, themes και άλλα χαρακτηριστικά. Επιπρόσθετα πλέον

<sup>3</sup> <http://jqts.com/>

<sup>4</sup> <http://www.sencha.com/products/touch/>

υπάρχει πλήρης υποστήριξη για δημιουργία hybrid native εφαρμογής με την χρήση του Apache Cordova ή Adobe PhoneGap. Στην παρακάτω Εικόνα 4 βλέπουμε ένα παράδειγμα από κάποιες επιλογές που υπάρχουν για το user interface. [12]



Εικόνα 4 - Sencha Touch User Interface Components

### 3.4 Adobe PhoneGap

Το Adobe PhoneGap<sup>5</sup> πρόκειται για ένα εργαλείο που επιτρέπει την μετατροπή μιας web εφαρμογής σε native [7]. Στηρίζεται στο Apache Cordova<sup>6</sup> το οποίο και αρχικά αναπτύχθηκε από την Adobe και πρόκειται για project ανοικτού κώδικα. Εκτός από το «πακετάρισμα» της web εφαρμογής σε native, πράγμα το οποίο προσφέρει δυνατότητα ανεβάσματος στο store εφαρμογών της αντίστοιχης πλατφόρμας, το PhoneGap προσφέρει και διεπαφές σε JavaScript πλήθος native δυνατοτήτων που μπορεί και να μην καλύπτονται από την HTML5 [8]. Για παράδειγμα μπορούμε να έχουμε πρόσβαση στον τοπικό αποθηκευτικό χώρο της συσκευής όπως στην SD card, στο native σύστημα ειδοποιήσεων κ.α. Οι πλατφόρμες τις οποίες υποστηρίζει είναι Android, BlackBerry, iOS, Windows Phone, Windows 8 και Tizen. Επίσης υπάρχουν διαθέσιμα πολλά plug-ins που προσφέρουν επιπρόσθετες υπηρεσίες ενώ η Adobe έχει ειδική υπηρεσία που μετατρέπει στο Cloud το HTML5 web application σε native σε

<sup>5</sup> <http://phonegap.com/>

<sup>6</sup> <http://cordova.apache.org/>

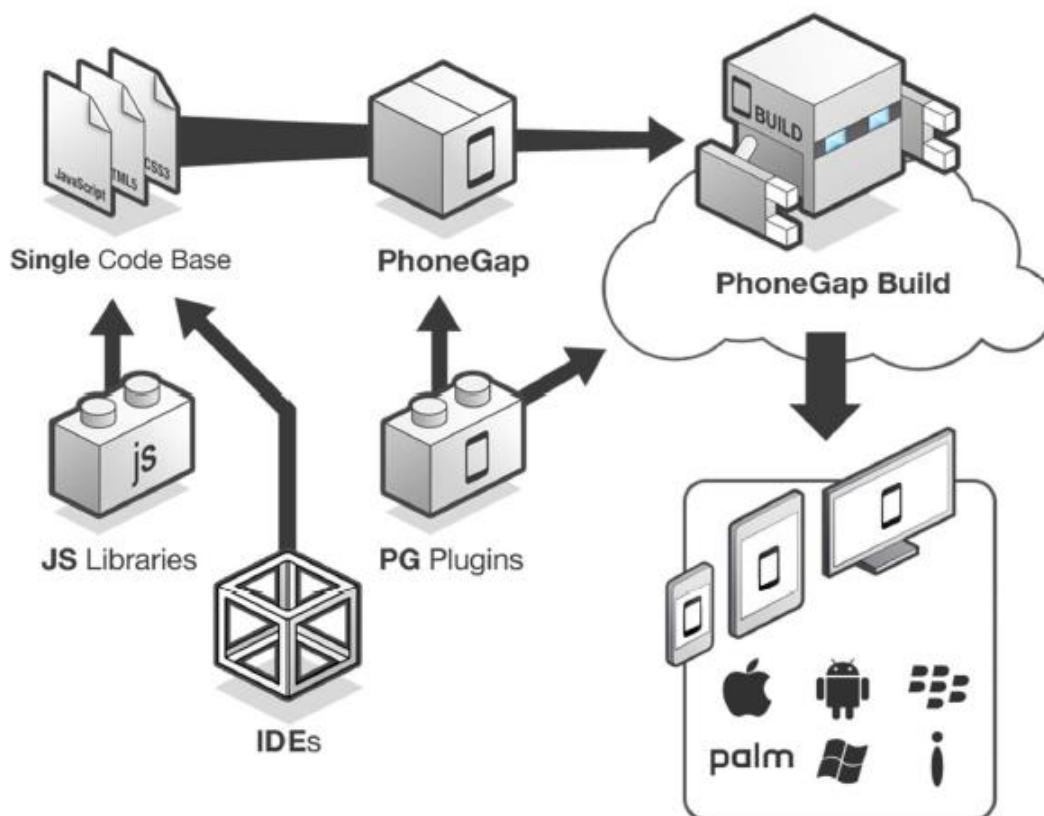
όλες τις υποστηριζόμενες πλατφόρμες με το Adobe PhoneGap Build<sup>7</sup>. Στην Εικόνα 5 βλέπουμε το περιβάλλον του εργαλείου αυτού.

The image shows two screenshots of the Adobe PhoneGap Build web interface. The top screenshot displays the 'Your apps' section for the 'PG Build App'. It includes a form to enter an app description, a table with app details (App ID: 673411, Version: n\_a, PhoneGap: 2.9.0), and buttons for 'Enable debugging', 'Enable hydration', 'delete', and 'Ready to build'. A banner for Adobe Creative Cloud is also visible. The bottom screenshot shows the 'PG Build App' page with a QR code, an 'Install' button, and a 'Builds' section with tabs for 'Builds', 'Plugins', 'Collaborators', and 'Settings'. Below this, there is a table of build configurations for different platforms (iOS, Windows, HP) with 'Rebuild' buttons and status indicators.

Εικόνα 5 - Adobe PhoneGap Build Interface

<sup>7</sup> <https://build.phonegap.com/>





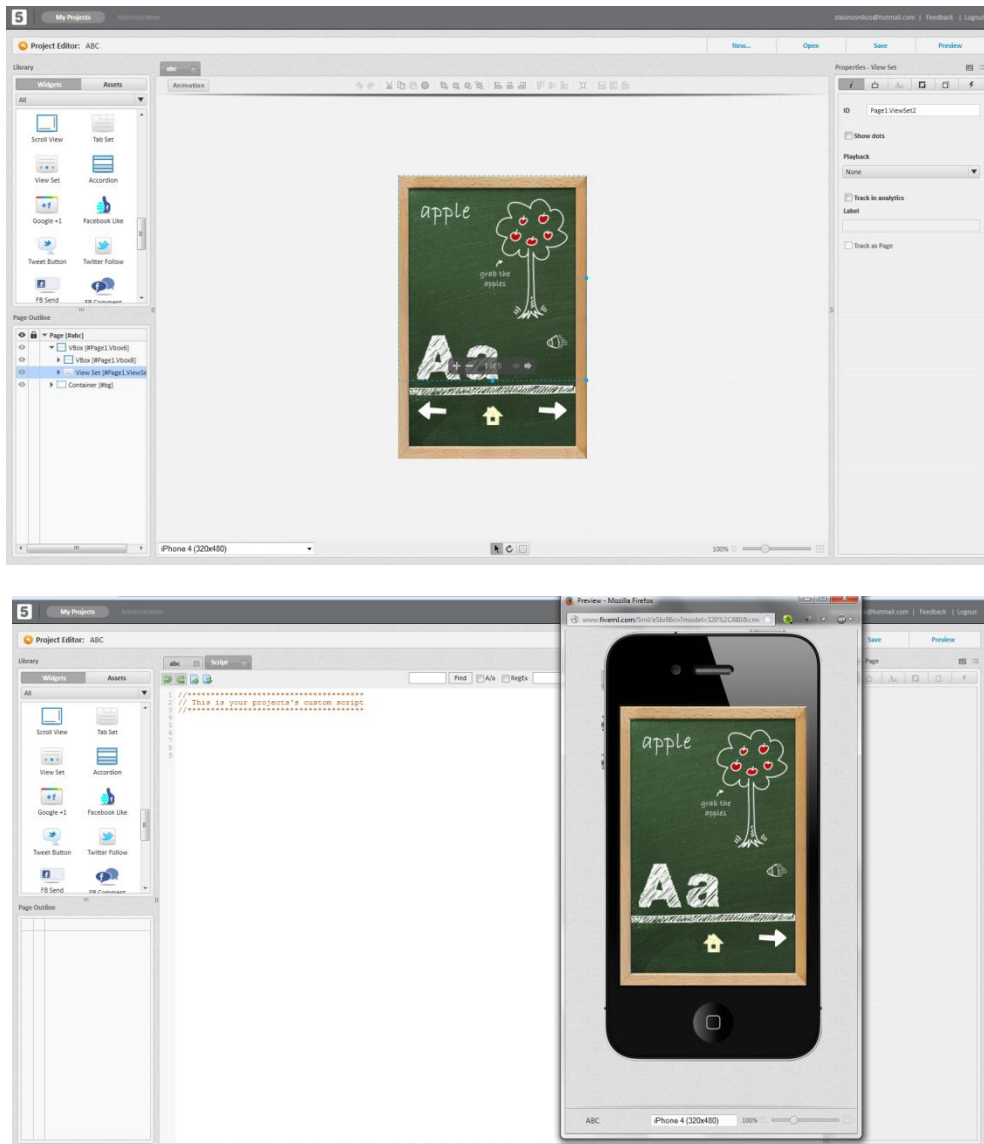
Εικόνα 6 - Διαδικασία ανάπτυξης εφαρμογής με PhoneGap

### 3.5 5ml

Το 5ml<sup>8</sup> (Fiveml) είναι ένα εργαλείο για ανάπτυξη εφαρμογών σε HTML5 για κινητές συσκευές. Αναπτύσσεται από την Velti και παρέχεται σε δωρεάν και επί πληρωμή έκδοση. Παρέχει ένα ιδιαίτερο εύχρηστο περιβάλλον για τον σχεδιασμό του UI της εφαρμογής με πολλές έτοιμες επιλογές όπως χωρισμός της οθόνης σε τμήματα, 3D Carousel, Accordion, ViewSet κ.α. Όλα βρίσκονται σε μορφή drag and drop ενώ για κάθε επιλογή (widget) παρέχονται επιπλέον σχετικές ρυθμίσεις. Επίσης έχει ενσωματωμένο ένα πολύ καλό εργαλείο για σχεδίαση animation και υποστήριξη social interaction. Πέρα από το βασικό περιβάλλον σχεδίασης του UI υπάρχει και ένας editor όπου ο προγραμματιστής γράφει τον JavaScript κώδικα της εφαρμογής του. Επίσης σε κάθε κατάλληλο component υπάρχουν και επιλογές για διαχείριση events όπως on-click κ.α. Επιπλέον υπάρχει και ενσωματωμένο εργαλείο προεπισκόπησης της εφαρμογής σε κινητό τηλέφωνο (εικονικό) ενώ μελλοντικά θα υπάρξει και υποστήριξη για μετατροπή του web application σε υβριδικό native (μέσω Apache Cordova). Στην παρακάτω Εικόνα 2 βλέπουμε το βασικό περιβάλλον του 5ml.

<sup>8</sup> <http://www.fiveml.com/>

«Σχεδίαση και Υλοποίηση Εφαρμογής για Κινητά Τηλέφωνα Αξιοποιώντας τις Πληροφορίες των Αισθητήρων τους»



Εικόνα 7 - 5ml Development Interface

### 3.6 Parse

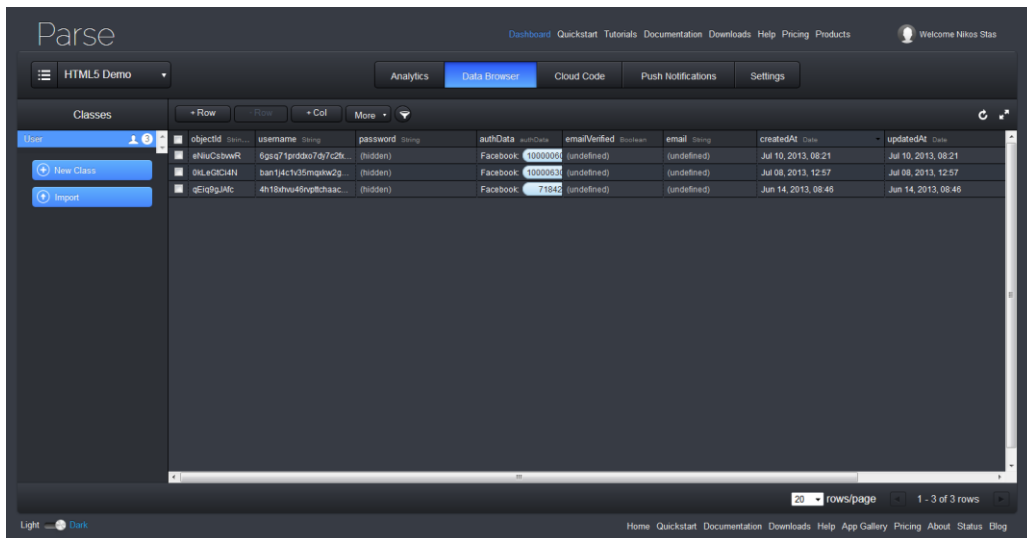
Το Parse<sup>9</sup> είναι ένα framework το οποίο παρέχει κυρίως backend services σε web applications και επικεντρώνεται κυρίως σε κινητές συσκευές. Παρέχει APIs για πολλές πλατφόρμες όπως iOS, OSX, Android, Windows Phone, Windows 8, JavaScript κ.α. Το ιδιαίτερο χαρακτηριστικό του Parse είναι ότι είναι cloud based δηλαδή οι υπηρεσίες που παρέχει βρίσκονται στο Cloud. Μερικές από τις κύριες υπηρεσίες του είναι:

<sup>9</sup> <https://www.parse.com/>

«Σχεδίαση και Υλοποίηση Εφαρμογής για Κινητά Τηλέφωνα Αξιοποιώντας τις Πληροφορίες των Αισθητήρων τους»

- Αποθήκευση και ανάγνωση δεδομένων στο Cloud (Parse Data),
- Σύστημα εγγραφής/σύνδεσης χρηστών,
- Σύστημα ασφαλείας για τα δεδομένα για το ποιος μπορεί να έχει πρόσβαση σε αυτά και ποιος όχι
- Σύστημα analytics για ανίχνευση της επισκεψιμότητας και των προτιμήσεων των χρηστών της εφαρμογής (Parse Analytics)
- Σύστημα push notifications (Parse Push) για αποστολή ειδοποιήσεων από server/cloud side
- Υποστήριξη social interaction όπως πχ Facebook login κ.α.
- Εκτέλεση κώδικα εφαρμογής από το Cloud με το Cloud Code για αποφυγή συστήματος server
- Hosting των εφαρμογών

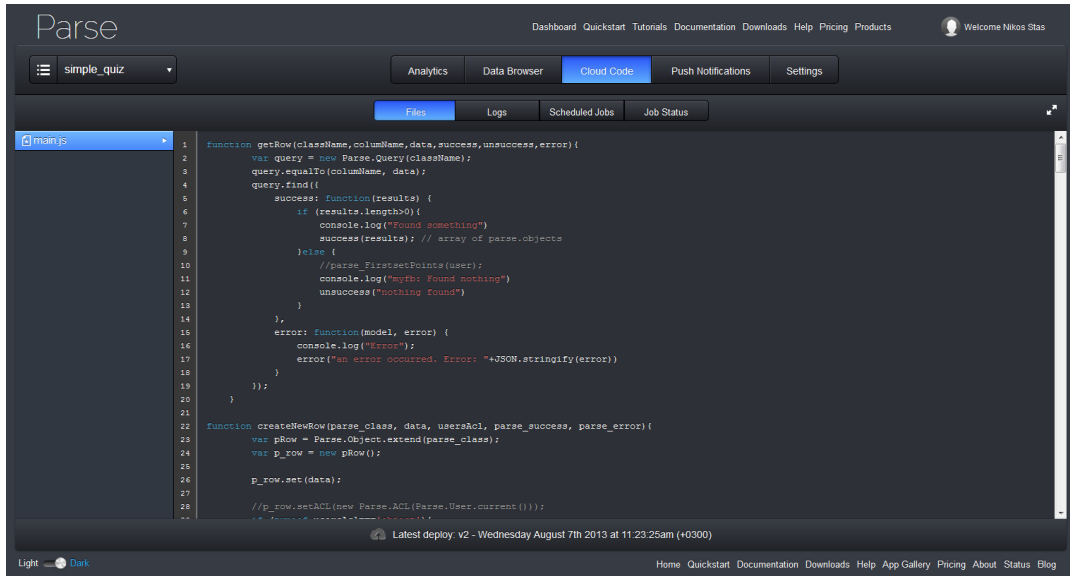
Οι βασικές υπηρεσίες του παρέχονται δωρεάν με κάποιους περιορισμούς όπως για παράδειγμα ποσά requests μπορούν να γίνουν σε μια εφαρμογή κτλ ενώ υπάρχει και επί-πληρωμή υπηρεσία. Παρακάτω βλέπουμε το περιβάλλον διαχείρισης του Cloud Data και του Cloud Code.



The screenshot shows the Parse Data Browser interface. The top navigation bar includes 'Dashboard', 'Quickstart', 'Tutorials', 'Documentation', 'Downloads', 'Help', 'Pricing', and 'Products'. The user is logged in as 'Welcome Nikos Stas'. The main content area displays a table of user data for the 'User' class. The table has columns for 'objectId', 'username', 'password', 'authData', 'emailVerified', 'email', 'createdAt', and 'updatedAt'. There are three rows of data shown.

objectId	username	password	authData	emailVerified	email	createdAt	updatedAt
8NuCsbwR	6ziq71prkso7q7c2b...	(hidden)	Facebook: 1000000	(undefined)	(undefined)	Jul 10, 2013, 08:21	Jul 10, 2013, 08:21
8KLG0C4N	ban14c1v35mpaix2g...	(hidden)	Facebook: 1000000	(undefined)	(undefined)	Jul 08, 2013, 12:57	Jul 08, 2013, 12:57
4Eiq9gJMc	4h18thv45v8lpaac...	(hidden)	Facebook: 71842	(undefined)	(undefined)	Jun 14, 2013, 08:45	Jun 14, 2013, 08:45

Εικόνα 8 - Data Browser



Εικόνα 9 - Cloud Code Editor

Πρόκειται για μια πολύ χρήσιμη υπηρεσία καθώς μπορεί κάποιος να τρέχει κώδικα server-side σε JavaScript άμεσα και εύκολα χωρίς να χρειάζεται να ασχολείται με στήσιμο server κτλ.

### 3.7 Titanium Appcelerator<sup>10</sup>



EXPLORE THE SDK

Create rich native iOS, Android, hybrid, and mobile web apps from a single JavaScript-based SDK.



MORE ABOUT STUDIO

An Eclipse-based integrated development environment (IDE) that simplifies the development process.



WRITE LESS CODE

Efficient code modules to reduce time from zero-to-app. Write less code. Create scalable apps.

Εικόνα 10 - Titanium Appcelerator

Πρόκειται για ένα ακόμα framework υλοποίησης εφαρμογών για κινητές συσκευές με δυνατότητα δημιουργίας πακέτου εγκατάστασης σε διάφορες πλατφόρμες. Η κύρια διαφορά του με άλλα είναι ότι ενώ ο προγραμματιστής γράφει την εφαρμογή σε

<sup>10</sup> <http://www.appcelerator.com/titanium/>

«Σχεδίαση και Υλοποίηση Εφαρμογής για Κινητά Τηλέφωνα Αξιοποιώντας τις Πληροφορίες των Αισθητήρων τους»

HTML5, το framework παράγει native κώδικα πράγμα το οποίο βελτιώνει πάρα πολύ την απόδοση της. Υποστηρίζει τις πλατφόρμες: iOS, Android, Windows, Blackberry και παρέχει ένα JavaScript-based SDK με πολλά APIs για όλες τις υποστηριζόμενες πλατφόρμες. Επίσης υπάρχει η δυνατότητα χρήσης ειδικού IDE και Cloud Services. Το μειονέκτημα του είναι ότι σε κάποιες περιπτώσεις ίσως χρειάζονται και κάποιες γνώσεις πάνω στην native πλατφόρμα.



## 4 Σχεδιασμός και Αρχιτεκτονική

Στο κεφάλαιο αυτό θα αναφερθούμε σε σημαντικά θέματα σχεδιασμού τόσο της λειτουργικότητας όσο και του περιβάλλοντος χρήστη και της τελικής εμπειρίας χρήστη των σύγχρονων mobile web εφαρμογών και ιστοσελίδων. Θα δούμε πως διάφορες σχεδιαστικές ιδέες επηρεάζουν την εμπειρία χρήστη και την ίδια την εφαρμογή και τι πλεονεκτήματα και μειονεκτήματα μπορεί αυτές να έχουν. Τέλος θα αναφερθούμε στον σχεδιασμό που ακολουθήθηκε στην εφαρμογή που αναπτύξαμε με ποιο σκεπτικό έγινε ο συγκεκριμένος σχεδιασμός και τι σκοπό εξυπηρετεί.

### 4.1 Mobile Στρατηγική

Όταν θέλουμε να δημιουργήσουμε μια mobile web εφαρμογή πρέπει να λάβουμε υπόψη μας ότι έχουμε να επιλέξουμε ανάμεσα στις εξής επιλογές: browser-based applications, full-screen web application ή native web applications. Πολλοί κανόνες αρχιτεκτονικής είναι κοινοί στους προηγούμενους τύπους εφαρμογών αλλά υπάρχουν επίσης και μερικές πρακτικές που είναι χρήσιμες μόνο σε έναν συγκεκριμένο τύπο.

Δεν πρέπει να αποκλείουμε χρήστες επειδή δεν έχουν τον mobile browser ή συσκευή που κρίνουμε ως πιο κατάλληλο για την εφαρμογή μας. Υπάρχουν πολλά παραδείγματα ακόμα και από πολύ μεγάλες εταιρείες που περιορίζουν την προσβασιμότητα στις ιστοσελίδες τους ή τις εφαρμογές τους σε συγκεκριμένες συσκευές ή περιηγητές ιστού λόγω των τεχνολογιών που υποστηρίζουν. Μπορεί ακόμα και να μην είναι καν βελτιστοποιημένα για touch συσκευές. Πρέπει λοιπόν να λαμβάνουμε υπόψη μας την διαφορετικότητα των χαρακτηριστικών τόσο των συσκευών όσο και τον περιηγητών ιστού χωρίς να αποκλείουμε χρήστες και να ρυθμίζουμε ανάλογα τον σχεδιασμό και ανάπτυξη της εφαρμογής. Κάτι τέτοιο βέβαια έχει κόστος στην ανάπτυξη της εφαρμογής και πιθανότητα να περιορίζει και την λειτουργικότητα την οποία επιδιώκουμε. Έτσι είναι στην ευχέρεια του κάθε προγραμματιστή τι θα αποφασίσει και πως θα σχεδιάσει την εφαρμογή του ώστε να καλύπτει τις ανάγκες και τις προδιαγραφές της εφαρμογής.

#### 4.1.1 Είδη Web Εφαρμογών

Πέρα από την λειτουργικότητα και τις τεχνολογίες πρέπει να αποφασίσουμε και σε ποιες πλατφόρμες θα αναπτύξουμε την mobile web εφαρμογή μας. Μπορούμε να επιλέξουμε ανάμεσα σε browser-based web application, full screen web εφαρμογή (browser based ή hybrid native) ή/και μια πλήρως native εφαρμογή [15]. Η επιλογή

μας εξαρτάται από την κάθε περίπτωση και τις προδιαγραφές που θέλουμε να καλύψουμε.

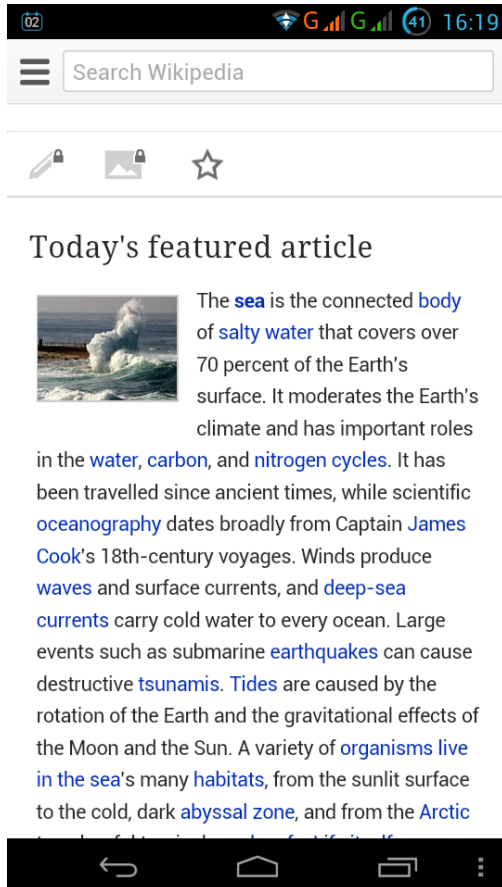
Με τις full-screen web applications κερδίζουμε περισσότερο χώρο στην παρουσίαση του περιβάλλοντος χρήστη αλλά και ο ίδιος ο χρήστης κερδίζει χώρο στην εσωτερική μνήμη της συσκευής του καθώς δεν αποθηκεύεται μόνιμα η εφαρμογή. Επίσης υπάρχει και εικονίδιο της εφαρμογής στο «Home-Screen» της συσκευής πράγμα το οποίο διευκολύνει τον χρήστη να θυμάται την εφαρμογή για μελλοντική χρήση [15]. Επίσης υπάρχει και η δυνατότητα για δημιουργία πακέτου εγκατάστασης ως native εφαρμογή ώστε να έχουμε πρόσβαση και στο application store των λειτουργικών συστημάτων (iOS, Android κτλ).

Το ίδιο πλεονέκτημα της καλύτερης ορατότητας-παρουσίασης του περιβάλλοντος χρήστη της εφαρμογής υπάρχει και στις native εφαρμογές. Εδώ όμως υπάρχει και η δυνατότητα να τρέχουμε native κώδικα με αποτέλεσμα να έχουμε πολύ περισσότερες δυνατότητες σε χαρακτηριστικά και τεχνολογίες απ ό τι μπορεί να προσφέρει ένας web browser, παρόλο που αυτό το κενό μικραίνει όλο και περισσότερο. Τέλος ο χρήστης μπορεί να βρει την εφαρμογή ακόμα πιο εύκολα αφού πλέον είναι διαθέσιμη και μέσα από το αντίστοιχο κατάστημα εφαρμογών της πλατφόρμας.

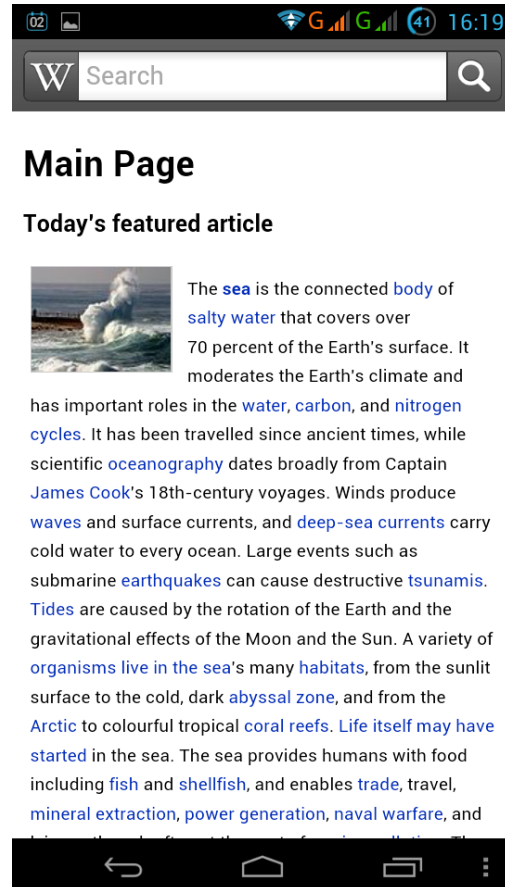
Συνήθως η λύση που θα ακλουθήσαμε εξαρτάται από το ίδιο το project και τις απαιτήσεις του. Μια καλή τεχνική είναι να προσφέρουμε ένα browser-based website με ενδεχομένως λιγότερες δυνατότητες και μια δεύτερη έκδοση σε full-screen ή native υλοποίηση η οποία και υποστηρίζει περισσότερα χαρακτηριστικά [4]. Πρέπει όμως σε κάθε περίπτωση να δίνουμε στον χρήστη την επιλογή για το τι θα προτιμήσει και να μην παρουσιάζουμε για παράδειγμα σε κάθε επίσκεψη του στην ιστοσελίδα μας ότι μπορεί να εγκαταστήσει και την native/full-screen έκδοση. Θα πρέπει το μήνυμα να εμφανίζεται μερικές φορές και μετά να εξαφανίζεται ώστε να μην πιέζουμε τον χρήστη και τον προδιαθέτουμε αρνητικά.

Ένα παράδειγμα υλοποίησης των παραπάνω είναι το Wikipedia. Εκτός από την mobile ιστοσελίδα για όλες τις πλατφόρμες υπάρχει και το native web application, το οποίο και υλοποιήθηκε με το Apache Cordova. Αυτή η εφαρμογή διατίθεται στο iOS από το App Store, στο BlackBerry από το App World και στο Android από το Google Play Store. Παρακάτω βλέπουμε την μορφή που έχει τόσο η ιστοσελίδα όσο και η native εφαρμογή.





Εικόνα 11 - Web Application



Εικόνα 12 - Android Native Web App

## 4.2 Context

Θα πρέπει να τονίσουμε ότι ένας mobile χρήστης κινείται σε τελείως διαφορετικό πλαίσιο απ' ό,τι ένας desktop χρήστης. Οι καταστάσεις δηλαδή στις οποίες χρησιμοποιούν τις εφαρμογές είναι διαφορετικές και θα πρέπει να ληφθούν σοβαρά υπόψη αυτές οι διαφορές. Έτσι λοιπόν θα πρέπει να σκεφτούμε και να ορίσουμε όλες τις πιθανές καταστάσεις/πλαίσια μέσα στα οποία ο χρήστης καλείται να χρησιμοποιήσει κάποια mobile web εφαρμογή [6]:

- Που βρίσκεται ο χρήστης;
- Τον λόγο που ο χρήστης χρησιμοποιεί την εφαρμογή.
- Τι ακριβώς αναζητεί ο χρήστης;
- Τι μπορούμε να προσφέρουμε στον χρήστη από μια “mobile” σκοπιά ώστε να λύσει η εφαρμογή το πρόβλημα του
- Που θα είναι ο χρήστης την στιγμή που χρησιμοποιεί την εφαρμογή; Θα περιπατάει στον δρόμο, θα χρησιμοποιεί κάποιο μέσο μεταφοράς, στο γραφείο του κ.α.

Αυτά τα «πλαίσια» θα μας βοηθήσουν πολύ στο να σχεδιάσουμε σωστά την πλοήγηση στην mobile web εφαρμογή, την χρηστικότητα αυτής καθώς και τα σενάρια χρήσης της [15].

### 4.3 Progressive Enhancement

Οι συσκευές που υπάρχουν σήμερα και χρησιμοποιούνται για την πρόσβαση τόσο σε μοντέρνες ιστοσελίδες όσο και σε web εφαρμογές παρουσιάζουν μεγάλη ποικιλία στα χαρακτηριστικά τους. Έτσι η υποστήριξη όλων των πιθανών browsers και των χαρακτηριστικών της κάθε συσκευής είναι μια επίπονη διεργασία για τον προγραμματιστή. Τα καινούργια χαρακτηριστικά που παρουσιάζονται στις νέες εκδόσεις των περιηγητών είναι πολύ χρήσιμα και εντυπωσιακά αλλά πρέπει να λάβουμε υπόψη μας και την ύπαρξη παλαιότερων περιηγητών στην ανάπτυξη και τον σχεδιασμό κάποιας εφαρμογής.

Με το progressive enhancement σχεδιάζουμε και αναπτύσσουμε ιστοσελίδες και εφαρμογές ιστού παρέχοντας πρόσβαση σε αυτές ανεξάρτητα από τις δυνατότητες του περιηγητή ιστού στον οποίο τρέχουν [10]. Τα κύρια σημεία της έννοιας του progressive enhancement είναι:

- Το βασικό περιεχόμενο είναι προσβάσιμο απ όλους τους περιηγητές ιστού.
- Η βασική λειτουργία είναι προσβάσιμη από όλους τους περιηγητές ιστού.
- Η semantic markup περιέχει όλα τα περιεχόμενα.
- Το εμπλουτισμένο layout παρέχεται από «εξωτερικό» CSS.
- Η εμπλουτισμένη συμπεριφορά/λειτουργία παρέχεται από εξωτερική JavaScript
- Οι ρυθμίσεις του περιηγητή ιστού του τελικού χρήστη πρέπει να μένουν ανέπαφες

Ο στόχος είναι να έχουμε ένα set κώδικα ο οποίος θα είναι συμβατός με όλες τις συσκευές. Έτσι θα μπορούμε να παρέχουμε την ίδια εμπειρία χρήστη σε κάθε συσκευή. Δεν πρέπει όμως να δημιουργούμε πολύ απλές ιστοσελίδες ή εφαρμογές ώστε να λειτουργούν σε όλες τις συσκευές αλλά και ούτε να δημιουργούμε ιστοσελίδες και εφαρμογές με τελευταίες τεχνολογίες οι οποίες θα λειτουργούν μόνο σε high-end συσκευές.

Στο mobile web, σύμφωνα με την progressive enhancement προσέγγιση, γίνεται κάποια server-side ανίχνευση και προσαρμογή πράγμα που είναι απαραίτητο για κάποια συγκεκριμένα mobile χαρακτηριστικά.

## 4.4 Regressive Enhancement

Το Regressive Enhancement είναι το αντίθετο του Progressive Enhancement. Δηλαδή ξεκινάμε με την υποστήριξη πιο τεχνολογικά εξελιγμένων πλατφορμών και μετά προσθέτουμε polyfills δηλαδή βιβλιοθήκες που καλύπτουν τα κενά και τις τεχνολογίες που δεν υποστηρίζονται [15]. Έτσι υπάρχει υποστήριξη και σε παλαιότερες πλατφόρμες με λιγότερες δυνατότητες. Αυτή όμως η μέθοδος σε mobile πλατφόρμες συνήθως δεν είναι ικανοποιητική καθώς προσθέτει μια επιπλέον επιβάρυνση έχοντας έτσι αντίκτυπο στην απόδοση της πλατφόρμας.

## 4.5 Responsive Design

Η «Αποκρίσιμη Σχεδίαση» είναι ένας τρόπος να σχεδιάσουμε μια ιστοσελίδα ή μια web εφαρμογή ώστε να προσαρμόζεται στα διάφορα μεγέθη οθονών και συσκευών. Ο παραδοσιακός τρόπος σχεδίασης μιας web ιστοσελίδας/εφαρμογής είναι να βασιζόμαστε σε ένα συγκεκριμένο πλάτος το οποίο προκύπτει από κάποια πρόβλεψη μας για τα αναμενόμενα μεγέθη οθονών του κοινού στο οποίο απευθύνεται η εφαρμογή και να προσαρμόζουμε τα στοιχεία στο πλάτος αυτό. [15]

Χρησιμοποιώντας τα Media Queries [2.5.5] μπορούμε να παρέχουμε προσαρμοζόμενα layouts για διαφορετικά μεγέθη οθονών. Για παράδειγμα μπορούμε να μικρύνουμε κάποιες φωτογραφίες ή ακόμα και να την κρύψουμε τελείως προκειμένου να έχουμε ένα πιο καθαρό και ευδιάκριτο layout σε μικρά μεγέθη οθονών. Έτσι με αυτόν τον τρόπο μπορούμε να χρησιμοποιούμε τον ίδιο κώδικα ως βάση και να καλύπτουμε παράλληλα όλα τις συσκευές και οθόνες.

## 4.6 Γενικές Ιδέες

Ένα mobile website ή εφαρμογή συνήθως αποτελείται από τις παρακάτω ζώνες:

- Header
- Κύρια Πλοήγηση
- Στοιχεία περιεχομένου
- Δευτερεύουσα πλοήγηση
- Footer

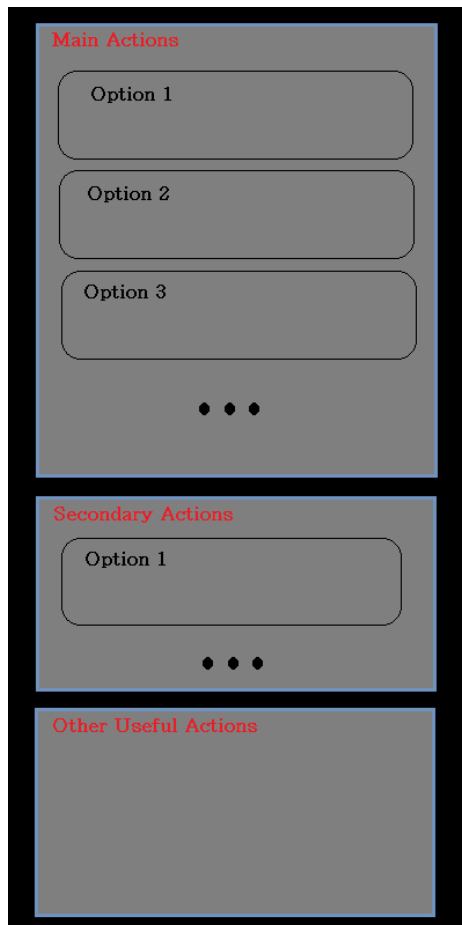
Τα παραπάνω στοιχεία τα συναντάμε σε μια κάθετη οριοθέτηση σε κινητά μαζί με την δυνατότητα για κάθετο scrolling. Η οριζόντια τοποθέτηση πρέπει να αποφεύγεται

και να προτιμάται μόνο για landscape τοποθέτηση. Ακολουθούν κάποιες γενικές κατευθύνσεις και προτάσεις για το User Interface και Experience [15]:

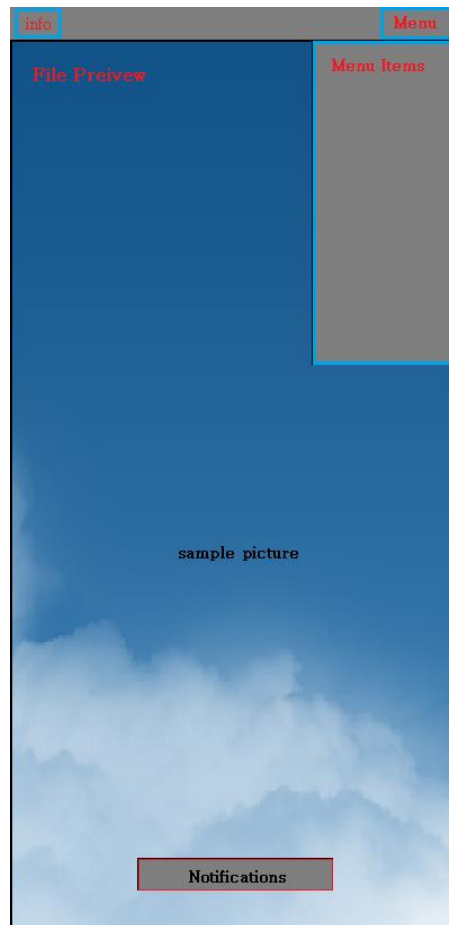
- Πρέπει να αποφεύγεται το οριζόντιο scrolling
- Πρέπει να χρησιμοποιούμε Responsive Design ώστε να προσφέρουμε τον καλύτερο σχεδιασμό για τις διάφορες συνθήκες που μπορεί να προκύψουν, όπως για παράδειγμα αλλαγής orientation
- Πρέπει να υπάρχει κάποια γραφική «ενότητα» με το αντίστοιχο desktop website, αν υπάρχει.
- Μειωμένη χρήση κειμένου
- Διαχωρισμός των τμημάτων με διαφορετικό background image/color
- Κρατάμε την κύρια πλοήγηση σε 4 ή 5 links
- Ύπαρξη «Πίσω» και «Go Top» κουμπιών
- Τα πιο χρησιμοποιούμενα χαρακτηριστικά να βρίσκονται στο πάνω μέρος
- Πρέπει να υπάρχει αρκετός κενός χώρος μεταξύ των touchable elements
- Για τα κουμπιά/elements που χρησιμοποιούνται περισσότερο πρέπει να έχουν μεγάλο μέγεθος
- Χρησιμοποίηση gestures όπου υποστηρίζονται
- Χρησιμοποίηση «άπειρης» λίστας αντί για οργάνωση σε σελίδες
- Χρησιμοποίηση της λειτουργίας «παρατεταμένου πατήματος» (long-press)
- Προτίμηση προκαθορισμένης-μόνιμης μπάρας πλοήγησης στο πάνω ή κάτω μέρος
- Απόκρυψη της μπάρας διευθύνσεων
- Ανάλυση των native touch applications για ιδέες χρηστικότητας
- Προσαρμογή του περιεχομένου στο πλάτος της οθόνης της συσκευής
- Χρησιμοποίηση κατάλληλου μεγέθους γραμματοσειράς
- Αποθήκευση του ιστορικού και των ρυθμίσεων του χρήστη για μελλοντική χρήση
- Υποστήριξη όσο το δυνατόν περισσότερων συσκευών

## 4.7 Σχεδιασμός Εφαρμογής

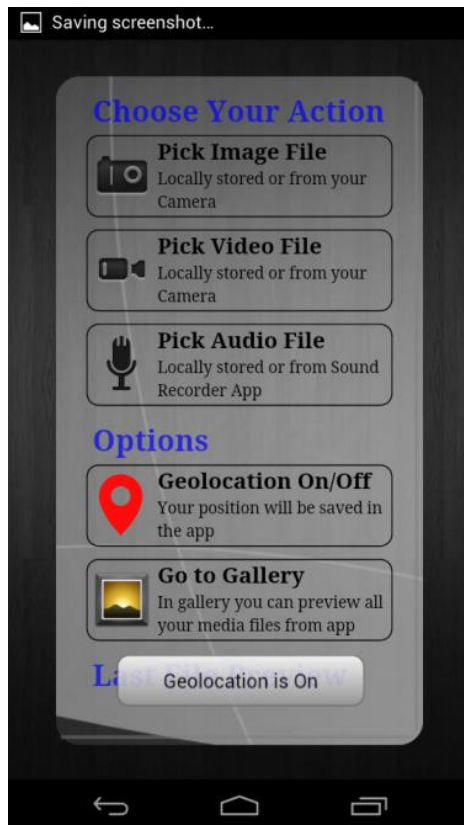
Για τον σχεδιασμό της εφαρμογής προσπαθήσαμε να ακολουθήσουμε τα περισσότερα από τα χαρακτηριστικά και τις ιδέες που περιγράψαμε παραπάνω. Ο κύριος στόχος ήταν να συνδυάσουμε όσο το δυνατόν περισσότερα νέα χαρακτηριστικά της HTML5 και να δημιουργήσουμε μια εφαρμογή χρήσιμη για τον τελικό χρήστη. Η ιδέα λοιπόν ήταν να δημιουργήσουμε μια ολοκληρωμένη λύση φωτογραφιών και βίντεο με πλούσια διαχείριση αρχείων και φυσικά social interaction που είναι συνηθισμένο πλέον σε σχεδόν κάθε εφαρμογή. Το αρχικό σχέδιο της εφαρμογής που θα φιλοξενήσει αυτά τα χαρακτηριστικά φαίνεται στις παρακάτω εικόνες:



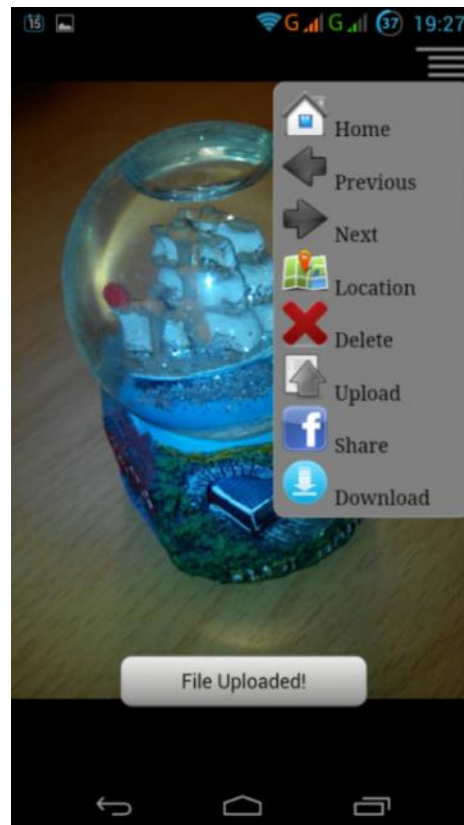
Εικόνα 13 - Σχέδιο Κύριας Οθόνης



Εικόνα 14 - Οθόνη Gallery



Εικόνα 15 - Κύρια Οθονη Εφαρμογης



Εικόνα 16 - Gallery

Η εφαρμογή προσαρμόζεται στο πλάτος της οθόνης της συσκευής ενώ υπάρχει και η δυνατότητα απόκρυψης της μπάρας διευθύνσεων, σε όσους browsers το υποστηρίζουν, επιτρέποντας καλύτερη αξιοποίηση της διαθέσιμης περιοχής οθόνης [4].

### 4.7.1 Κύρια Σελίδα

Όπως παρατηρούμε στην Εικόνα 13 χρησιμοποιούμε μια κατακόρυφη οριοθέτηση των στοιχείων όπου θα φιλοξενηθούν τόσο οι κύριες όσο και οι δευτερεύουσες λειτουργίες. Τα στοιχεία προσαρμόζονται στο πλάτος της οθόνης χωρίς να υπάρχει η δυνατότητα ολίσθησης κατά πλάτος [4]. Ολίσθηση υπάρχει μόνο κατακόρυφη σε περίπτωση που τα αντικείμενα υπερβαίνουν το ύψος της οθόνης. Κάτι τέτοιο είναι επιθυμητό ώστε ο χρήστης να γνωρίζει εκ των προτέρων προς ποια κατεύθυνση θα πρέπει να ψάξει για επιπλέον λειτουργίες που πιθανόν να μην χωράνε στην οθόνη της συσκευής του. Αυτό έχει ως αποτέλεσμα την γρηγορότερη πλοήγηση του χρήστη το οποίο είναι πολύ χρήσιμο για μια mobile εφαρμογή. Οι διάφορες λειτουργίες έχουν χωριστεί σε ομάδες προτεραιότητας. Έτσι στο πάνω μέρος που εμφανίζεται και πρώτο έχουμε τις κύριες λειτουργίες ώστε ο χρήστης να έχει άμεση πρόσβαση σε αυτές μιας και αυτές είναι το πιο πιθανό να χρειαστεί να χρησιμοποιήσει. Στο πρώτο

λοιπόν πλαίσιο αποφασίσαμε να τοποθετήσουμε τις τρεις κύριες λειτουργίες της εφαρμογής δηλαδή την φωτογράφιση, βιντεοσκόπηση και ηχογράφιση. Στο δεύτερο πλαίσιο έχουμε τοποθέτηση την μετάβαση στην σελίδα/οθόνη Gallery για την προεπισκόπηση των αρχείων και το κουμπί για την ενεργοποίηση/απενεργοποίηση. Λειτουργίες δηλαδή όχι τόσο άμεσης προτεραιότητας. Επίσης στο δεύτερο πλαίσιο τοποθετήσαμε ένα mini-preview του τελευταίου αρχείου, μια λειτουργία που είναι ορίσαμε ως δευτερεύουσας σημασίας καθώς δεν υπάρχει ανάγκη για άμεση πρόσβαση του χρήστη σε αυτή με το που ανοίγει την εφαρμογή. Τέλος οι επιλογές αυτές είναι ευδιάκριτες και ξεχωρίζουν μεταξύ τους. Έχουν χρησιμοποιηθεί ειδικά εικονίδια για την καλύτερη αναγνώριση τους από τον χρήστη και έχει αφιερωθεί αρκετός χώρος στο κουμπί τους για ευκολότερη/αμεσότερη επιλογή. Έχει επιλεγεί και ειδικό background τόσο για καλύτερο αισθητικό αποτέλεσμα αλλά και για να τονίζονται οι διαθέσιμες επιλογές.

## 4.7.2 Σελίδα Gallery

Στην δεύτερη Εικόνα 14 βλέπουμε το σχέδιο της δεύτερης σελίδας. Στο πάνω μέρος της οθόνης υπάρχει ένα header. Εκεί έχουμε ενσωματώσει δυο λειτουργίες. Η πρώτη παρέχει ενημέρωση στον χρήστη ότι κάποια ενέργεια είναι σε εξέλιξη. Στην περίπτωση της εφαρμογής έχουμε επιλέξει την διαδικασία ανεβάσματος κάποιου αρχείου. Έτσι ο χρήστης θα μπορεί να γνωρίζει ανά πάσα στιγμή ότι κάποια διαδικασία ανεβάσματος κάποιου αρχείου που έχει επιλέξει εξελίσσεται. Η δεύτερη αποτελεί ένα κουμπί το οποίο και ενεργοποιεί μια λίστα από επιπλέον επιλογές σχετικές με το αρχείο το οποίο βλέπει εκείνη την στιγμή. Η τοποθέτηση του κουμπιού αυτού στο πάνω δεξιά μέρος της οθόνης έγινε με το σκεπτικό ότι ο χρήστης είναι πιο εύκολο να προσεγγίσει με το δάκτυλο του το συγκεκριμένο σημείο της οθόνης του, κάνοντας έτσι πιο εύκολη και ακριβή την επιλογή. Το μενού με την λίστα επιλογών εμφανίζεται σε κατακόρυφη διάταξη και οι επιλογές συνοδεύονται από εικονίδια για πιο εύκολη επιλογή και κατανόηση των επιλογών από τον χρήστη. Το μενού εμφανίζεται πάνω από την εικόνα ενώ αν ο χρήστης πατήσει πάλι το κουμπί κλείνει η λίστα. Το μέγεθος του κουμπιού όσο και των επιλογών είναι κατάλληλο ώστε να διευκολύνεται η επιλογή τους σε κινητές συσκευές. Η εικόνα ή το βίντεο/ηχογράφιση καταλαμβάνουν όπως είναι φυσικό το μεγαλύτερο μέρος της οθόνης ώστε να έχουν όσο το δυνατό πιο ευκρινή αναπαράσταση τους. Το μέγεθος τους προσαρμόζεται ανάλογα το πώς κρατάει ο χρήστης την συσκευή. Αν είναι σε οριζόντια θέση το αρχείο προσαρμόζεται στο ύψος της οθόνης ενώ αν είναι σε κάθετη θέση προσαρμόζεται στο πλάτος της οθόνης. Τέλος υπάρχουν και ειδικές σύντομες ειδοποιήσεις οι οποίες εμφανίζονται για λίγα δευτερόλεπτα στην οθόνη και μετά εξαφανίζονται όπως βλέπουμε στις Εικόνα 14, Εικόνα 16. Οι ειδοποιήσεις αυτές προορίζονται για όχι τόσο σημαντικές ενέργειες και κυρίως περιγράφουν ολοκλήρωση ενεργειών. Τέτοιες ειδοποιήσεις υπάρχουν και στην αρχική σελίδα/οθόνη της εφαρμογής όπως μπορούμε να δούμε και στην Εικόνα 15. Επίσης σε

κάποιες επιλογές της λίστας που εμφανίζεται από τον κουμπί πάνω δεξιά ενεργοποιούν πρώτα έναν διάλογο επιβεβαίωσης. Ο διάλογος αυτός αφορά κυρίως ενέργειες ανεβάσματος αρχείου σε τρίτους και ο σκοπός του είναι να αποφευχθούν λανθασμένα πατήματα και πιθανές άσκοπες χρεώσεις από την υπηρεσία δεδομένων του παρόχου κινητής τηλεφωνίας.

### 4.7.3 Σελίδα Map

Τον ίδιο σχεδιασμό με την σελίδα/οθόνη του Gallery έχουμε ακολουθήσει και στην σελίδα προβολής της γεωγραφικής θέσης του αρχείου. Υπάρχει δηλαδή το header με μόνο ένα κουμπί που οδηγεί τον χρήστη στην προηγούμενη σελίδα, το Gallery, και η αναπαράσταση του χάρτη όπως αναπαριστούμε και την εικόνα/αρχείο.

### 4.7.4 Επεκτασιμότητα Εφαρμογής

Στην εφαρμογή έχουν χρησιμοποιηθεί λειτουργίες από το Facebook, το Parse καθώς και μια custom υλοποίηση ανεβάσματος αρχείου σε server με PHP.

Με τις λειτουργίες που παρέχει το Facebook μπορούμε να υλοποιήσουμε επιπλέον δυνατότητες, πέρα του ανεβάσματος στον λογαριασμό του χρήστη. Μπορούμε να στείλουμε ένα αρχείο στο profile κάποιου φίλου μας, να στείλουμε με προσωπικό μήνυμα κάποιο αρχείο. Επίσης μπορούμε να συνδυάσουμε και τις πληροφορίες της τοποθεσίας του αρχείου και της ώρας/ημερομηνίας μαζί με κάποιο τυποποιημένο ή προσωπικό μήνυμα/σχόλιο στο profile κάποιου χρήστη. Απαραίτητη βεβαία είναι η ύπαρξη σύνδεσης δεδομένων.

Με το Parse μπορούμε να έχουμε μια βάση δεδομένων με τους χρήστες της εφαρμογής μας. Μπορούμε έτσι να υλοποιήσουμε επιπλέον υπηρεσίες βασισμένες σε σύστημα εγγραφής/σύνδεσης χρηστών προσφέροντας έτσι νέα βελτιωμένη λειτουργικότητα. Ένα παράδειγμα θα μπορούσε να είναι ένα σύστημα μηνυμάτων μεταξύ των χρηστών της εφαρμογής, ένας πίνακας με τον αριθμό των φωτογραφιών του κάθε χρήστη, την τοποθεσία με τις περισσότερες φωτογραφίες κ.α. Όλα αυτά βέβαια προϋποθέτουν την ύπαρξη σύνδεσης στο διαδίκτυο.

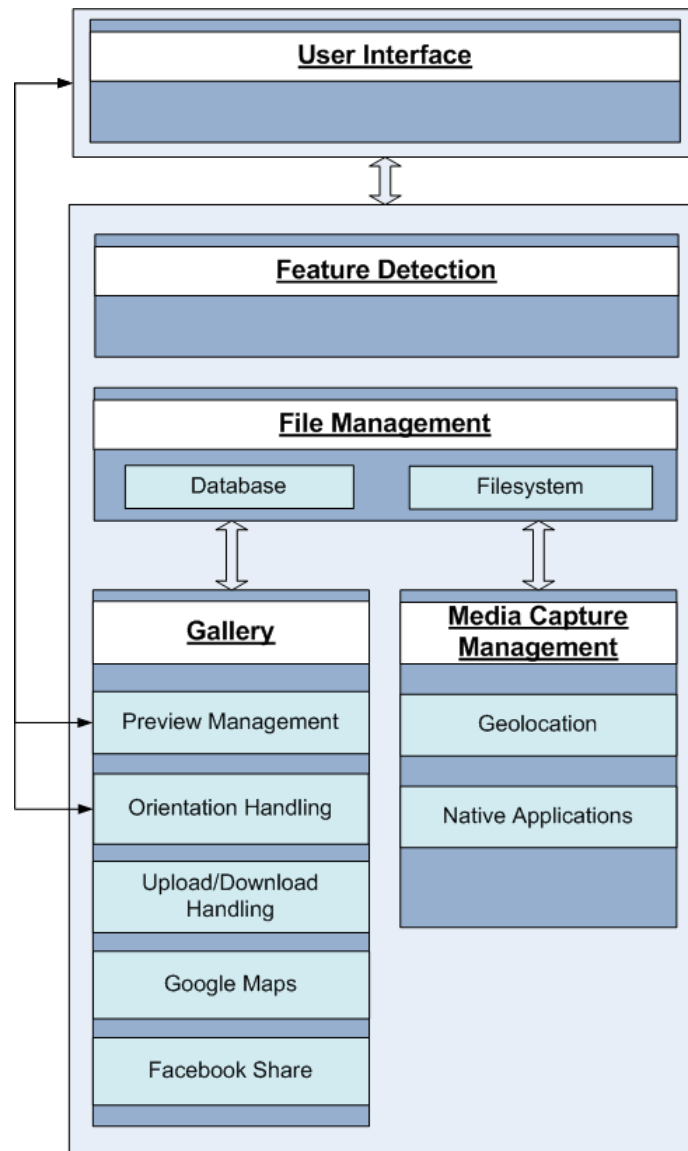
Το ανέβασμα σε συγκεκριμένο server θα μπορούσε να προσφέρει πρόσθετες λειτουργίες οργάνωσης και διαχείρισης των αρχείων. Όπως ύπαρξη και διαχείριση φακέλων για αρχεία, δημόσιοι και προσωπικοί φάκελοι για διαμοιρασμό με άλλους χρήστες, επιπλέον social λειτουργίες από την μεριά του server κ.α.

Να σημειωθεί ότι με την χρήση του File System API της HTML5 δυνατότητα για οργάνωση και διαχείριση αρχείων με φάκελους κ.τ.λ. υπάρχει και όντως της ίδιας της εφαρμογής.



## 4.7.5 Αρχιτεκτονική

Ο σχεδιασμός της εφαρμογής ακολουθεί μια τμηματική προσέγγιση με σαφείς διασυνδέσεις μεταξύ των ξεχωριστών στοιχείων της αρχιτεκτονικής. Παρακάτω βλέπουμε σχηματικά το διάγραμμα της αρχιτεκτονικής όπου διακρίνουμε τα διάφορα τμήματα της και τα επιμέρους στοιχεία.



Εικόνα 17 - Αρχιτεκτονική Εφαρμογής

Πιο αναλυτικά παρακάτω αναλύουμε τα δομικά στοιχεία της αρχιτεκτονικής:

**User Interface:** Αυτό το στοιχείο αναφέρεται στον σχεδιασμό και την διαχείριση του περιβάλλοντος χρήστη (User Interface) της εφαρμογής και συμπεριλαμβάνει και την

«Σχεδίαση και Υλοποίηση Εφαρμογής για Κινητά Τηλέφωνα Αξιοποιώντας τις Πληροφορίες των Αισθητήρων τους»

εμπειρία χρήστη (User Experience). Αλληλεπιδρά με τα υπόλοιπα στοιχεία της αρχιτεκτονικής παρέχοντας έτσι μια διασύνδεση του χρήστη με αυτά. Παρέχει την δυνατότητα αλληλεπίδρασης μεταξύ χρήστη και εφαρμογής δίνοντας το κατάλληλο περιβάλλον επικοινωνίας.

**Features Detection Management:** Στο στοιχείο αυτό γίνεται η ανίχνευση των χαρακτηριστικών που υποστηρίζονται από τον browser στον οποίο τρέχει η εφαρμογή και ανάλογα μπορούμε να μεταβούμε ή όχι στο επόμενο «επίπεδο» (ιστοσελίδα). Αυτό γίνεται παρουσιάζοντας και έναν πίνακα με τα υποστηριζόμενα χαρακτηριστικά του περιηγητή ιστού.

**File Management:** Το συγκεκριμένο στοιχείο αφορά στη διαχείριση της λειτουργίας αποθήκευσης, διαγραφής, η τροποποίησης κάποιου αρχείου που δέχθηκε η εφαρμογή. Αποτελείται από δυο υπο-στοιχεία: το Filesystem και το Database.

**Filesystem:** Το στοιχείο αυτό αναφέρεται σε ένα sandboxed filesystem στο οποίο έχει δυνατότητα πρόσβασης μόνο η εφαρμογή. Εδώ αποθηκεύονται τα αρχεία που λαμβάνονται από τις native εφαρμογές. Να σημειώσουμε ότι στον συγκεκριμένο χώρο έχει πρόσβαση μόνο η εφαρμογή και δεν υπάρχει δυνατότητα πρόσβασης στο filesystem του ίδιου του λειτουργικού και των αρχείων του χρήστη. Προφανώς για λόγους ασφαλείας.

**Database:** Αποτελεί μια βάση δεδομένων για την αποθήκευση των ονομάτων των αρχείων που έχουμε αποθηκεύσει στο filesystem της εφαρμογής για γρηγορότερη εύρεση του αρχείου όταν το χρειαστούμε. Επίσης υπάρχει και η δυνατότητα να αποθηκεύσουμε και το path. Προορίζεται για την καλύτερη οργάνωση και βελτιστοποίηση της εύρεσης των αρχείων.

**Media capture:** Εδώ βρίσκονται μερικές από τις πιο βασικές λειτουργίες της εφαρμογής. Έχουμε την εισαγωγή του media αρχείου από τρίτες εφαρμογές, το Geolocation και την αποθήκευση του αρχείου από το στοιχείο File Management.

**Native applications :** Από το στοιχείο αυτό καλείται η εφαρμογή του λειτουργικού της συσκευής η οποία και παρέχει δυνατότητες λήψης φωτογραφιών, βίντεο και ηχογραφήσεων με όλες τις δυνατότητες ρυθμίσεων που παρέχει αυτή. Σε περίπτωση που υπάρχουν περισσότερες εφαρμογές να ολοκληρώσουν κάποια λειτουργία ο χρήστης μπορεί να επιλέξει μονός του την επιθυμητή εφαρμογή συστήματος. Αν δεν υπάρχει συγκεκριμένη εφαρμογή θα μπορεί να επιλεγεί ο περιηγητής αρχείων.

Ύστερα από επιβεβαίωση του χρήστη το web app αναλαμβάνει να «δεχθεί» την φωτογραφία από τη σχετική εφαρμογή.

**Geolocation:** Το συγκεκριμένο στοιχείο λαμβάνει κατευθείαν από την συσκευή/λειτουργικό τη τοποθεσία του χρήστη και αποθηκεύεται για μελλοντική χρήση. Στην εφαρμογή μας χρησιμοποιείται για την αποθήκευση geo-tags στα αρχεία. Για να μπορεί η εφαρμογή να λάβει τις συντεταγμένες από το σύστημα θα πρέπει πρώτα ο χρήστης να το έχει επιτρέψει. Εμφανίζεται σχετικό μήνυμα από τον περιηγητή ιστού για επιβεβαίωση.

**Gallery:** Το συγκεκριμένο στοιχείο αφορά στην προβολή των αρχείων media που υπάρχουν στην εφαρμογή και γενικά την διαχείριση τους. Περιλαμβάνει τα sub-modules : Preview Management, Orientation Handling, Upload & Download Handling, Google Maps and Facebook Share. Όπως βλέπουμε και στην εικόνα της αρχιτεκτονικής υπάρχει σύνδεση με το File Management καθώς από αυτό λαμβάνουμε τα αρχεία.

**Orientation Change Handle:** Προσαρμόζει το περιβάλλον χρήστη και τις εικόνες, βίντεο ανάλογα αν η συσκευή βρίσκεται σε οριζόντια ή κάθετη θέση. Η πληροφορία της θέσης της συσκευής εξάγεται «κατευθείαν» από τον orientation sensor της συσκευής μέσω του κατάλληλου Api της HTML5.

**Preview Management:** Το συγκεκριμένο στοιχείο αφορά στη διαδικασία εμφάνισης και αναπαραγωγής των media αρχείων της εφαρμογής δηλαδή των εικόνων, των βίντεο και των ηχογραφήσεων. Αυτά τα αρχεία τα λαμβάνουμε από το File Management στοιχείο.

**Google Maps:** Εμφανίζει τη γεωγραφική τοποθεσία στην οποία δημιουργήθηκε το αντίστοιχο αρχείο του οποίου τις συντεταγμένες (geo-tag) έχουμε αποθηκευμένες. Για την αναπαράσταση της τοποθεσίας χρησιμοποιούμε την πλατφόρμα Google Maps.

**Upload/Downalod Handling:** Διαχειρίζεται την όλη διαδικασία του ανεβάσματος του αρχείου που επέλεξε ο χρήστης στον server που χρησιμοποιείται από την εφαρμογή ως χώρος αποθήκευσης αρχείων. Επίσης περιλαμβάνει και την διαδικασία λήψης/αποθήκευσης κάποιου αρχείου από την εφαρμογή στην συσκευή του χρήστη.

**Facebook Share Management:** Διαχειρίζεται την διαδικασία σύνδεσης του χρήστη με τον λογαριασμό του στο facebook και την διαδικασία ανεβάσματος του αρχείου στο album της εφαρμογής, το οποίο δημιουργείται αυτόματα στον χρήστη.

#### 4.7.5.1 Διασύνδεση Στοιχείων

Ακλουθήσαμε μια λογική διασύνδεση μεταξύ των στοιχείων της αρχιτεκτονικής. Το στοιχείο User Interface περιλαμβάνει την διαχείριση του περιβάλλοντος χρήστη και την επικοινωνία εφαρμογής χρήστη. Περιλαμβάνει των CSS3 κώδικα και τον JavaScript κώδικα που δημιουργεί και διαχειρίζεται το γραφικό περιβάλλον. Διαχειρίζεται την αναπαραγωγή και εμφάνιση των αρχείων της εφαρμογής ενώ αναλαμβάνει και την τροποποίηση του περιβάλλοντος ανάλογα το orientation της συσκευής. Έτσι βλέπουμε και την στενή σύνδεση που υπάρχει με το στοιχείο Gallery και συγκεκριμένα το Preview Management και το Orientation Handling Περιέχει τους διάλογους επιλογής που προορίζονται κυρίως για επιβεβαίωση ή ακύρωση ενεργειών όπως upload αρχείων κτλ.

Η δεύτερη διασύνδεση είναι αυτή με το στοιχείο “file management”. Με αυτή παρέχουμε την αποθήκευση, ανάγνωση και διαχείριση των αρχείων της εφαρμογής.

Παρέχει την κατάλληλη διεπαφή στα υπόλοιπα στοιχεία ώστε να λάβουν και να στείλουν τα κατάλληλα δεδομένα αρχείων βίντεο, φωτογραφιών και ηχογραφήσεων. Στο στοιχείο “Gallery” παρέχεται η δυνατότητα ανάγνωσης και περιήγησης των διαθέσιμων αρχείων μέσω του “File System” και “Database” ενώ αντίστοιχα στο “Media Capture Management” παρέχεται η δυνατότητα της αποθήκευσης των αρχείων που λαμβάνονται από το input element μέσω των native εφαρμογών.

## 5 Υλοποίηση Εφαρμογής

Η βασική τεχνολογία που χρησιμοποιείται στο web app είναι η HTML5, η οποία αποτελείται από νέα HTML elements, JavaScript APIs και CSS3 ιδιότητες. Παρακάτω θα παρουσιάσουμε πιο αναλυτικά τις επιμέρους τεχνολογίες της HTML5 που χρησιμοποιήθηκαν και πως χρησιμοποιήθηκαν. Επίσης θα παρουσιάσουμε και κάποια κομμάτια κώδικα της εφαρμογής για να γίνει περισσότερο κατανοητή η προγραμματιστική χρησιμοποίηση αυτών των τεχνολογιών. Πρέπει να αναφέρουμε ότι λόγω της μη υποστήριξης όλων των HTML5 χαρακτηριστικών που χρησιμοποιήθηκαν από όλους τους browsers η εφαρμογή υποστηρίζεται προς το παρόν από τον Chrome για Android και τον Opera Mobile για κινητά τηλέφωνα και Chrome, Opera για desktop.

### 5.1 Ανίχνευση HTML5 χαρακτηριστικών

Για την αναγνώριση των χαρακτηριστικών της HTML5 που υποστηρίζει ο browser που τρέχει το web app, χρησιμοποιήθηκε το Modernizr.js . Πρόκειται για μια βιβλιοθήκη JavaScript που κάνει αυτόματα τους απαραίτητους ελέγχους και μας δίνει τα κατάλληλα objects για τα features που θέλουμε να ελέγξουμε.

Για παράδειγμα: το object Modernizr.applicationcache είναι true ή false ανάλογα αν υπάρχει υποστήριξη ή όχι για Application Cache στον browser.

Τα αποτελέσματα της υποστήριξης ή μη εμφανίζονται σε κατάλληλο πίνακα. Ο συγκεκριμένος πίνακας/σελίδα είναι χρήσιμος κυρίως για προγραμματιστές ή για απασφαλμάτωση κάποιας δυσλειτουργίας της εφαρμογής. Υπάρχει κυρίως για να παρέχει κάποιες λεπτομέρειες ως προς την υποστήριξη του συστήματος/περιηγητή.

### 5.2 Offline Application

#### 5.2.1 Application Cache

Κάναμε χρήση της Application Cache προκειμένου η εφαρμογή να έχει την δυνατότητα να λειτουργήσει απρόσκοπτα και εκτός σύνδεσης με το διαδίκτυο. Για τον λόγο αυτό δημιουργήσαμε ένα manifest αρχείο. Μέσα σε αυτό ορίσαμε τα αρχεία της εφαρμογής που θα πρέπει να αποθηκευτούν τοπικά στην συσκευή του χρήστη και ποια αρχεία της εφαρμογής χρειάζονται σύνδεση στο διαδίκτυο. Τα αρχεία αυτά θα αποθηκευτούν τοπικά στον περιηγητή ιστού με το πρώτο φόρτωμα της εφαρμογής

από τον ιστότοπο. Παρακάτω βλέπουμε μέρος του manifest αρχείου της εφαρμογής μας και τα ονόματα των αντίστοιχων αρχείων.

#### Κώδικας 12 - AppCache Manifest Example

```
CACHE MANIFEST
#1
html5demo.html
html5demo.js
html5demo.css
css/gallery.css
gpson.png
gps.png
mic.png
```

Όπως βλέπουμε παραπάνω για να δουλέψει και εκτός σύνδεσης η πρώτη σελίδα της εφαρμογής έχουμε συμπεριλάβει τόσο το αρχείο .html όσο και τα .css, .js καθώς και τις αντίστοιχες εικόνες/assets που χρειαζόμαστε ώστε να έχει την σωστή εμφάνιση και λειτουργικότητα η σελίδα. Παρόμοια έχουμε συμπεριλάβει και τα υπόλοιπα αρχεία που χρειάζεται η εφαρμογή.

### 5.2.2 Ανίχνευση Συνδεσιμότητας

Για να ανιχνεύσουμε αν ο περιηγητής και κατά συνέπεια και η εφαρμογή μας είναι συνδεδεμένη με το διαδίκτυο κάποια στιγμή χρησιμοποιούμε το `object navigator.onLine` το οποίο και επιστρέφει `true` ή `false` ανάλογα με την συνδεσιμότητα. Η παραπάνω λειτουργία χρησιμοποιείται σε διάφορα σημεία της εφαρμογής όπου χρειάζεται ο χρήστης να επικοινωνήσει με κάποιον εξυπηρετητή στο διαδίκτυο ώστε να ανεβάσει/διαμοιράσει κάποιο από τα αρχεία της εφαρμογής.

Ένα απλό παράδειγμα των του παραπάνω, το οποίο και χρησιμοποιείται στην εφαρμογή, είναι το παρακάτω:

#### Κώδικας 13 - Connectivity Check

```
if (navigator.onLine) {
    // Do something
} else {
    drawToast(' Not online :( ');
}
```

## 5.3 Διαχείριση Αρχείων

### 5.3.1 Media Capture

Χρησιμοποιήσαμε το συγκεκριμένο Device Api ώστε να έχουμε πρόσβαση στην camera και το μικρόφωνο της συσκευής. Η λήψη του αρχείου γίνεται μέσω ενός ειδικού τύπου input element με την ιδιότητα type να είναι ορισμένη ως file και προσθέτοντας την ειδική επιλογή capture. Η αντίστοιχη native εφαρμογή του συστήματος είναι αυτή που δημιουργεί και στέλνει το αρχείο στην εφαρμογή μας. Παρακάτω βλέπουμε μέρος του κώδικα της εφαρμογής που κάνει χρήση του παραπάνω API:

#### Κώδικας 14 - "Device" Input Elements

```
<input id='camphoto' type="file" accept="image/*" capture='camera' style="visibility:hidden;">

<input id='video' type="file" accept="video/*" capture='camcorder' style="visibility:hidden;">

<input id='audio' type="file" accept="audio/*" capture='microphone' style="visibility:hidden;">
```

Όπως βλέπουμε στον παραπάνω κώδικα χρησιμοποιήθηκε και η επιλογή accept ώστε να βάλουμε κάποιους περιορισμούς ως προς το τι αρχεία δέχεται ως είσοδο το κάθε input element. Έχουμε δηλαδή μόνο αρχεία εικόνας για την λειτουργία της φωτογράφισης, μόνο αρχεία βίντεο για την βιντεοσκόπηση και αρχεία ήχου για την ηχογράφιση. Επίσης χρησιμοποιήθηκε και η επιλογή style visibility ώστε να κρύψουμε τις συγκεκριμένες φόρμες. Αυτό έγινε για να σχεδιάσουμε το δικό μας κουμπί χωρίς τους περιορισμούς του input element. Πατώντας το δικό μας κουμπί ενεργοποιούμε το input element όπως φαίνεται και στον παρακάτω κώδικα:

#### Κώδικας 15 - Onclick Event for Input Elements

```
onclick="document.getElementById('video').click()
```

### 5.3.2 File Api

Με το File Api διαβάζουμε το αρχείο - file object - που δέχεται η εφαρμογή με το input element. Έτσι μπορούμε πλέον να χρησιμοποιήσουμε πληροφορίες σχετικές με το αρχείο όπως το όνομα, το μέγεθος του αρχείου καθώς και τα ίδια τα δεδομένα του αρχείου. Επίσης μπορούμε και να γράψουμε ένα αρχείο στο filesystem για παράδειγμα.

«Σχεδίαση και Υλοποίηση Εφαρμογής για Κινητά Τηλέφωνα Αξιοποιώντας τις Πληροφορίες των Αισθητήρων τους»

### 5.3.3 FileSystem Api

Για την αποθήκευση του αρχείου χρησιμοποιήσαμε το FileSystem Api. Έτσι δημιουργήσαμε ένα sandboxed file system μεγέθους 1GB, το οποίο έχει την ιδιότητα TEMPORARY δηλαδή το σύστημα μπορεί να το διαγράψει αν δεν υπάρχει πλέον άλλος ελεύθερος χώρος για να χρησιμοποιήσει. Τα αρχεία αποθηκεύονται στον root κατάλογο.

Εδώ θα πρέπει να σημειώσουμε ότι την στιγμή την εκπόνησης της παρούσας διπλωματικής, λίγοι περιηγητές ιστού υποστηρίζουν το παραπάνω χαρακτηριστικό. Ο Google Chrome και ο Opera, τόσο οι εκδόσεις για σταθερούς υπολογιστές όσο και οι εκδόσεις για κινητές συσκευές (κυρίως με λειτουργικό σύστημα Android), υποστηρίζουν το παραπάνω χαρακτηριστικό.

### 5.3.4 Local Storage

Στο Local Storage αποθηκεύουμε τον αριθμό των αρχείων που έχουμε αποθηκεύσει στην εφαρμογή μας συνολικά και τον αριθμό των αρχείων που υπάρχουν αποθηκευμένα την παρούσα στιγμή στην εφαρμογή.

### 5.3.5 WebSQL Database

Στην WebSQL βάση δεδομένων έχουμε επιλέξει να αποθηκεύσουμε το path του κάθε αρχείου ώστε να μπορούμε εύκολα να το ανακτήσουμε για ανασκόπηση αργότερα.

Παρακάτω βλέπουμε ένα κομμάτι κώδικα της εφαρμογής όπου μπορούμε να δούμε συνολικά όσα περιγράψαμε παραπάνω. Βλέπουμε την προγραμματιστική διαδικασία αποθήκευσης ένα αρχείου εικόνας στην εφαρμογή.

#### Κώδικας 16 - File System Save File

```
function onInitFs(fs) {
  fsVar = fs;
  console.log('Opened file system: ' + fs.name);
}

window.requestFileSystem = window.requestFileSystem ||
window.webkitRequestFileSystem;

window.requestFileSystem(window.TEMPORARY, 1 * 1024 * 1024 * 1024, onInitFs,
errorHandler);

document.getElementById('camphoto').addEventListener('change',
handleFileSelect, false);

function handleFileSelect(evt) {
  var files = evt.target.files; // FileList object
```

«Σχεδίαση και Υλοποίηση Εφαρμογής για Κινητά Τηλέφωνα Αξιοποιώντας τις Πληροφορίες των Αισθητήρων τους»



```

// Loop through the FileList and render image files as thumbnails.
for (var i = 0, file; file = files[i]; i++) {
  console.log('file type: ' + file.type);
  // Only process image files.
  if (file.size < 5000000) {
    if (file.type.match('image/jpeg')) {
      console.log("The input is image file");
      //continue;
      localStorage.setItem('counter', ++localStorage.counter);
      key_c = localStorage.counter;

      localStorage.setItem('fcounter', ++fkey_c);
      fkey_c = localStorage.fcounter;
      //
      // save file to file-system
      //
      (function (f) {
        fsVar.root.getFile('picture_' + fkey_c + '.jpg', {
          create : true,
          exclusive : true
        }, function (fileEntry) {
          fileEntry.createWriter(function (fileWriter) {
            fileWriter.write(f); // Note: write() can take a
File or Blob object.
          }, errorHandler);
        }, errorHandler);
      })(file);

      var reader = new FileReader();

      // Closure to capture the file information.
      reader.onloadend = (function (theFile) {
        return function (e) {
          console.log('mphka');
          preview.innerHTML = '';
          var imgObj = new Image();
          preview.appendChild(imgObj);
          imgObj.id = 'picturepre';
          imgObj.src = this.result;
          imgObj.setAttribute('onclick', 'open_win()');

          //
          // save to webSQLdb
          //

          db.transaction(function (tx) {
            tx.executeSql("INSERT INTO img(path) VALUES
(?)",
            ['picture_' + fkey_c + '.jpg']);
            sql = '';
            console.log('sql1');
            drawToast("Image saved!");
            //listImages();
          });

          var testvar,
          len;

          console.log('sql results all :\n' + testvar +
'len:\n' + len);

          // call function:
          // get Geolocation of Pic
          //
          if (localStorage.checkPosition === 'true') {
            getPosition();

```

```
        drawToast("Got Position");
    }
    };
    }) (file);
    reader.readAsDataURL(file);
}
}
}
```

### 5.3.6 Geolocation Api

Μαζί με το αρχείο αποθηκεύουμε και την τοποθεσία του. Το Geolocation Api μας παρέχει τις συντεταγμένες της συσκευής του χρήστη μόλις αποθηκευτεί το αρχείο στην εφαρμογή. Θα πρέπει βέβαια να είναι ενεργοποιημένη και η αντίστοιχη επιλογή.

#### Κώδικας 17 - Geolocation

```
function getPosition() {
    if (navigator.geolocation) {
        var timeoutVal = 10 * 1000 * 1000;

        navigator.geolocation.getCurrentPosition(
            savePosition,
            //displayPosition,
            displayError, {
                enableHighAccuracy : true,
                timeout : timeoutVal,
                maximumAge : 0
            });
    } else {
        alert("Geolocation is not supported by this browser");
    }
}

// save geolocation
function savePosition(position) {
    localStorage.setItem('lat_' + parseInt(localStorage.counter),
    position.coords.latitude);
    localStorage.setItem('long_' + parseInt(localStorage.counter),
    position.coords.longitude);
    localStorage.setItem('accuracy_' + parseInt(localStorage.counter),
    position.coords.accuracy);
    localStorage.setItem('timestamp_' + parseInt(localStorage.counter),
    position.timestamp);}
```

## 5.4 Διαχείριση Περιβάλλοντος Χρήστη

### 5.4.1 Swipe Events

Στην σελίδα Gallery δημιουργήσαμε swipe events ώστε ο χρήστης να μπορεί να περιηγηθεί στα αρχεία του πιο εύκολα. Πρόκειται στην ουσία για συνεχόμενα touch

«Σχεδίαση και Υλοποίηση Εφαρμογής για Κινητά Τηλέφωνα Αξιοποιώντας τις Πληροφορίες των Αισθητήρων τους»

events δεξιά ή αριστερά. Συγκεκριμένα με το που ακουμπήσει ο χρήστης το δάκτυλο του στην οθόνη της συσκευής ενεργοποιείται το touchstart event και έτσι έχουμε το αρχικό σημείο επαφής. Με το σύρσιμο του δακτύλου ενεργοποιείται το touchmove event στο οποίο και υπολογίζουμε την απόσταση που έχει διανύσει στον οριζόντιο άξονα, μιας και αυτός μας ενδιαφέρει για το swipe. Σε σε περίπτωση που αυτή η απόσταση έχει ξεπεράσει ένα ελάχιστο που έχουμε ορίσει ενεργοποιούμε και τις αντίστοιχες λειτουργίες που θέλουμε, στην περίπτωση μας το CSS3 2D Transform. Τέλος μόλις τελειώσει το άγγιγμα ενεργοποιείται το touchend event στο οποίο ελέγχουμε ξανά την απόσταση ώστε να μεταβούμε στο επόμενο ή προηγούμενο αρχείο ανάλογα.

Παρακάτω βλέπουμε ένα παράδειγμα από τον κώδικα της εφαρμογής:

### Κώδικας 18 - Swipe Handle

```
function initEvents() {
    //registering touch events to the wrapper
    wrapper.addEventListener("touchstart", startHandler, false);
    wrapper.addEventListener("touchmove", moveHandler, false);
    wrapper.addEventListener("touchend", endHandler, false);
};

//funciton called when touch start event is fired i.e finger is pressed on
the screen
function startHandler(e) {
    //stores the starting X co-ordinate when finger touches the device
screen
    e.preventDefault();
    startX = e.touches[0].pageX; //.changedTouches[0]
    //timer is set on
    timer = setInterval(function () {
        timerCounter++;
    }, 10);
    isTouchStart = true;
    //e.preventDefault(); //prevents the window from scrolling.
};

//funciton called when touch move event is fired i.e finger is dragged over
the screen
function moveHandler(e) {
    if (isTouchStart) {
        distanceX = e.touches[0].pageX - startX;
        if (distanceX > 40 || distanceX < -40) {
            wrapper.style.webkitTransitionDuration = 1 + "s";
            wrapper.style.webkitTransform = "scale(0.5)";
        }
        //move the slide container along with the movement of the finger
        // wrapper.style.webkitTransform = "translate3d(" + (distanceX +
currentDistance) + "px, 0,0)";
    }
};

//funciton called when touch end event is fired i.e finger is released from
screen
// here's what we ll do after the touch events!!!!!!!
function endHandler(e) {
    clearInterval(timer); //timer is stopped
    if (distanceX > 40) {
        direction = "right";
        console.log('right');
    }
};
```

```
        previous();
    }
    if (distanceX < -40) {
        direction = "left";
        console.log('left');
        next();
    }
}

timerCounter = 0; //reset timerCounter
isTouchStart = false; //reset the boolean var
distanceX = 0; //reset the distance moved for next iteration
};
```

Επίσης πρέπει να αναφέρουμε ότι χρησιμοποιούμε και το `e.preventDefault()`; ώστε να αποτρέψουμε πιθανές παρεμβολές από άλλους διαχειριστές αγγίγματος από τον περιηγητή ιστού. Πολλοί περιηγητές ιστού αναθέτουν ειδικές λειτουργίες σε κάποιες συγκεκριμένες ακολουθίες αγγιγμάτων με αποτέλεσμα όταν συμπίπτουν με κάποια που χρησιμοποιείται σε εφαρμογή να υπάρχει πρόβλημα στο ποια θα ενεργοποιηθεί πρώτη.

## 5.4.2 Orientation Change Event

Η εφαρμογή αυτή προορίζεται για φορητές συσκευές οι οποίες μπορούν να είναι είτε σε οριζόντια είτε σε κάθετη θέση. Έτσι θα πρέπει η εφαρμογή μας να αντιμετωπίζει κατάλληλα αυτές τις αλλαγές την θέση της συσκευής και να αναπροσαρμόζει το περιεχόμενο της. Αυτό γίνεται χρησιμοποιώντας το `orientationchange` event το οποίο ενεργοποιείται σε κάθε αλλαγή της θέσης της συσκευής. Σε κάθε ενεργοποίηση του event ελέγχουμε αν το νέο πλάτος ή ύψος είναι μεγαλύτερο. Έτσι μπορούμε να διαπιστώσουμε την οριζόντια η κάθετη θέση της συσκευής και να αναπροσαρμόσουμε το `div` στο οποίο αναπαριστούμε τις εικόνες και βίντεο της εφαρμογής ώστε να ταιριάζουν καλύτερα στις νέες διαστάσεις πλάτους-ύψους. Παρακάτω βλέπουμε τον κώδικα της εφαρμογής που διαχειρίζεται την αλλαγή αυτή:

### Κώδικας 19 - Orientation Change Handle

```
window.addEventListener("orientationchange", function () {
    // Announce the new orientation number
    console.log("orientation change listener");
    document.getElementById("container").style.height =
(window.innerHeight - 38) + "px";
    if (window.innerHeight > window.innerWidth) {
        document.getElementById("slideshow").style.width = "100%";
        document.getElementById("slideshow").style.height = "auto";
    } else {
        document.getElementById("slideshow").style.height = "100%";
        document.getElementById("slideshow").style.width = "auto";
    }
});
}, false);
```

### 5.4.3 CSS3 2D Transform

Για να εμπλουτίσουμε την εμπειρία χρήστη προσθέσαμε ένα απλό εφέ κατά την μετάβαση από το ένα αρχείο στο άλλο. Πρόκειται για ένα scale CSS3 2D Transform, του οποίου ορίσαμε την διάρκεια ώστε να ξεκινάει με το swipe event, που περιγράψαμε και παραπάνω, κατά το touchmove event. Επίσης προσθέσαμε και ένα επιπλέον scale CSS3 2D Transform ακριβώς μετά την αλλαγή του αρχείου ώστε να επανέλθει η προβολή του αρχείου στην αρχική κανονική κατάσταση. Τα δυο scale CSS3 2D Transform που χρησιμοποιήθηκαν στην εφαρμογή μας φαίνονται στον παρακάτω κώδικα σε JavaScript:

#### Κώδικας 20 - CSS3 2D Scale Transform

```
wrapper.style.webkitTransitionDuration = 1 + "s";  
wrapper.style.webkitTransform = "scale(0.5)";
```

```
wrapper.style.webkitTransitionDuration = 0.5 + "s";  
wrapper.style.webkitTransform = "scale(1)";
```

### 5.4.4 Προβολή Αρχείων

Για την προβολή των αρχείων του χρήστη, δηλαδή τις εικόνες, βίντεο και ηχογραφήσεις χρησιμοποιήθηκαν 2 HTML5 elements. Το Video Element για τα βίντεο και τις ηχογραφήσεις και το Canvas Element για τις εικόνες. Επίσης ανάλογα με το τι είδος είναι το αρχείο, κρύβουμε και εμφανίζουμε το κατάλληλο element. Παρακάτω βλέπουμε τον HTML κώδικα καθώς και τον JavaScript κώδικα για την εμφάνιση και κρύψιμο των αντίστοιχων elements:

#### Κώδικας 21 - Preview Files

```
<canvas id="slideshow" ></canvas>
```

```
<video style="display:none" id="videopreview" type='video/mp4'  
controls ontouchend='this.play();'>Video Not Supported</video>
```

```
document.getElementById("slideshow").style.display = "block";  
document.getElementById("videopreview").style.display = "none";
```

### 5.4.5 Επικοινωνία με το Χρήστη

Έχουμε δημιουργήσει δυο τρόπους επικοινωνίας της εφαρμογής με τον χρήστη και αντίστροφα. Αναπτύξαμε τα notifications, σύντομα μικρά μηνύματα που εμφανίζονται στο κάτω μέρος της εφαρμογής και εξαφανίζονται μετά από μερικά

δευτερόλεπτα και τα dialog boxes που είναι διάλογοι επιβεβαιώσεις από τον χρήστη για την ολοκλήρωση ή ακύρωση κάποιας ενέργειας.

## Κώδικας 22 - Notifications & Dialog Box

### Notifications

```
function hideToast () {
    window.clearInterval (intervalCounterG);
    var alert = document.getElementById ("toast");
    alert.style.opacity = 0;
}

function drawToast (message) {
    window.clearInterval (intervalCounterG);
    var alert = document.getElementById ("toast");
    //alert.innerHTML = '';
    if (alert == null) {
        var toastHTML = '<div id="toast">' + message + '</div>';
        document.body.insertAdjacentHTML ('beforeEnd', toastHTML);
    } else {
        alert.innerHTML = message;
        //var toastHTML = '<div id="toast">' + message + '</div>';
        //document.body.insertAdjacentHTML ('beforeEnd', toastHTML);
        alert.style.opacity = .9;
    }
    intervalCounterG = window.setInterval (hideToast, 2000);
}
```

### dialogboxMsg.js

```
function postMsg (msg) {
    document.getElementById ('msg').innerHTML = msg;
    document.getElementById ("dialogLink").click ();
}

function simpleMsg (msg, opt) {
    disableCancel ();
    document.getElementById ('msg').innerHTML = msg;
    document.getElementById ("dialogLink").click ();
    okbtn.onclick = function (e) {
        if (opt ===
'nofiles') {closeDialog ();window.location='html5demo.html';}
        else {
            closeDialog ();
        }
    };
}

function closeDialog () {
    document.getElementById ("close").click ();
}

function enableCancel () {
    document.getElementById ('cancelbtn').style.display='inline';
    //document.getElementById ('cancelbtn').style.left='50px';
    //document.getElementById ('okbtn').style.right='-50px';
    document.getElementById ('okbtn').style.right= '-'+((250)-
200)/2+'px';
}
```

«Σχεδίαση και Υλοποίηση Εφαρμογής για Κινητά Τηλέφωνα Αξιοποιώντας τις Πληροφορίες των Αισθητήρων τους»

```

document.getElementById('cancelbtn').style.left=
((250/*window.innerWidth*0.8*/)-200)/2+'px';
}

function disableCancel(){
document.getElementById('cancelbtn').style.display='none';
//document.getElementById('cancelbtn').style.left='50px';
document.getElementById('okbtn').style.right= '-'+((250)-
100)/2+'px';
//document.getElementById('okbtn').style.right='-100px';
}

```

## 5.4.6 Viewport

Χρησιμοποιώντας το meta tag viewport ο περιηγητής ιστού σχεδιάζει κατάλληλα το περιβάλλον της εφαρμογής ώστε να ταιριάζει στην ανάλυση της οθόνης της συσκευής στην οποία τρέχει. Χρησιμοποιήσαμε τις παρακάτω viewport ιδιότητες: width, height, initial-scale, minimum-scale και maximum-scale. Έχουμε θέσει τις τιμές των scale ιδιοτήτων στο 0.95 γιατί, από δοκιμές, με αυτή την τιμή και με το επιπλέον meta που βλέπουμε παρακάτω και αφορά την βελτιστοποίηση της σελίδας ως mobile web application, είναι δυνατή η απόκρυψη της address bar σέρνοντας προς τα κάτω την σελίδα. Έχουμε χρησιμοποιήσει και ένα ειδικό εικονίδιο που εμφανίζεται δίπλα στην διεύθυνση της σελίδα στην μπάρα διεύθυνσης αλλά και στον σελιδοδείκτη. Επίσης βάλουμε και ειδικό icon να φαίνεται όταν ο χρήστης επιλέξει να αποθηκεύσει μια συντόμευση της εφαρμογή στην home-screen του λειτουργικού του και ενεργοποιήσαμε και την λειτουργία full-screen όταν ανοίγεται από εκεί (λειτουργία που δουλεύει σε Chrome Android Beta v32 και iOS προς το παρόν).

### Κώδικας 23 - Viewport, FullScreen & Homescreen Icon

```

<meta name="apple-mobile-web-app-capable" content="yes" />
<meta name="mobile-web-app-capable" content="yes">

<link rel="shortcut icon" href="/html5demo/favicon.ico"
type="image/x-icon">
<link rel="icon" href="/html5demo/favicon.png" />

<link rel="apple-touch-icon" href="/html5demo/favicon.png"/>
<link rel="apple-touch-icon-precomposed"
href="/html5demo/favicon.png"/>

<meta name="viewport" content="width=device-width,height=device-
height,initial-scale=1, minimum-scale=1, maximum-scale=1" />

```

## 5.5 Υπηρεσία Διαχείρισης Αρχείου

### 5.5.1 Λήψη Αρχείου

Για να μπορεί ο χρήστης να κατεβάσει το αρχείο που έχει αποθηκευτεί στην εφαρμογή στην συσκευή του χρησιμοποιήσαμε το link element με την επιλογή download ενεργοποιημένη. Έτσι με την παραπάνω επιλογή, πατώντας το link του αρχείου που έχουμε περάσει στο element θα ενεργοποιηθεί την διαδικασία αποθήκευσης αρχείου του περιηγητή ιστού. Επίσης μπορεί να εμφανιστεί και ένας διάλογος αποθήκευσης, ανάλογα τον περιηγητή.

#### Κώδικας 24 - Download Link Element

```
<a download="" href="" id="dlink" ></a>
```

### 5.5.2 Μεταφόρτωση Αρχείου

Για να ανεβάσουμε ένα αρχείο στον server μας στην διεύθυνση <http://62.217.125.126/html5demo/upload.php> χρησιμοποιήσαμε το jQuery Ajax POST με τις παραμέτρους που φαίνονται παρακάτω:

#### Κώδικας 25 - Ajax File Upload

```
try {
    $.ajax({
url : "/html5demo/upload.php",
    type : "POST",
    data : 'file=' + file222 + '&ext=' + ext,
    processData : false,
    contentType : 'application/x-www-form-urlencoded',
    cache : false,
    success : function (data) {
        console.log("success ");
drawToast('File Uploaded!');
document.getElementById("upload_loading").style.display = 'none';
    },
    error : function (shr, status, data) {
        console.log("error " + data + " Status " + shr.status);
drawToast(" An error occured :( ");
document.getElementById("upload_loading").style.display =
'none';
    },
    complete : function (shr, status, data) {
        console.log("complete " + data + " Status " + shr.status);
    }
    });
} catch (e) {
    console.log(e);
}
```



### 5.5.3 Facebook

Για να ανεβάσουμε ένα αρχείο στον λογαριασμό ενός χρήστη στο Facebook πρέπει να ακολουθηθούν κάποια βήματα:

Το αρχικό βήμα είναι να δημιουργήσουμε έναν λογαριασμό στο Facebook ως developer. Μπορεί να ενεργοποιηθεί σε οποιονδήποτε λογαριασμό. Στην συνέχεια στον λογαριασμό αυτό πρέπει να δημιουργήσουμε μια νέα εφαρμογή. Οι επιλογές της δικής μας εφαρμογής φαίνονται στην Εικόνα 18.

Με την δημιουργία της εφαρμογής έχουμε στην διάθεση μας το App ID και το οποίο και θα χρησιμοποιήσουμε στον κώδικα της εφαρμογής μας ώστε να γίνεται η επιβεβαίωση και επικοινωνία με το Facebook και να μπορούμε να χρησιμοποιήσουμε τις δυνατότητες που μας δίνει, όπως να κάνει log in ο χρήστης και να ανεβάσει κάποιο αρχείο μέσα από την εφαρμογή μας στον λογαριασμό του. Παρακάτω βλέπουμε την αρχικοποίηση που γίνεται στην εφαρμογή μας για την επικοινωνία με το Facebook:

The screenshot displays the Facebook App Settings interface for an application named 'HTML5 Demo'. The settings are organized into several sections:

- Basic info:** Includes fields for Display Name (HTML5 Demo), Namespace, Contact Email (stasinonikos@hotmail.com), App Domains, and Sandbox Mode (set to Disabled).
- Select how your app integrates with Facebook:**
  - Website with Facebook Login:** Site URL is http://62.217.125.126/html5demo/.
  - App on Facebook:** Canvas URL is http://62.217.125.126/html5demo/ and Secure Canvas URL is https://62.217.125.126/html5demo/. Canvas Width is set to Fixed (760px) and Canvas Height is set to Fluid. Unity Integration Enabled is Disabled.
  - Mobile Web:** Mobile Site URL is http://62.217.125.126/html5demo/ and Accept Mobile Web Payments is Disabled.
- Native iOS App:** Publish from my iOS app to Facebook.
- Native Android App:** Publish from my Android app to Facebook.
- Page Tab:** Build a custom tab for Facebook Pages.

A 'Save Changes' button is located at the bottom of the settings panel.

Εικόνα 18 - Facebook App Settings

### Κώδικας 26 - Parse & Facebook Init

```
Parse.initialize("GWldgmeWVv8GNytBjvZga83slW8lOfrblwbBd8rW",  
"5GHxRN0ln80MC12Mb8yo8KGZMIZwf6KU9dI6v16Q");  
  
(function (d, s, id) {  
  var js,  
      fjs = d.getElementsByTagName(s)[0];  
  if (d.getElementById(id)) {  
    return;  
  }  
  js = d.createElement(s);  
  js.id = id;  
  js.src = "//connect.facebook.net/en_US/all.js";  
  fjs.parentNode.insertBefore(js, fjs);  
})(document, 'script', 'facebook-jssdk');  
  
window.fbAsyncInit = function () {  
  Parse.FacebookUtils.init({  
    appId : '475031902586526', // Facebook App ID  
    //channelUrl : '//WWW.YOUR_DOMAIN.COM/channel.html', // Channel File  
    //status : true, // check login status  
    cookie : true, // enable cookies to allow Parse to access the session  
    xfbml : true, // parse XFBML  
    frictionlessRequests : true  
  });  
}
```

Στον παραπάνω κώδικα βλέπουμε και την αρχικοποίηση του Parse για το οποίο μιλήσαμε σε προηγούμενο κεφάλαιο. Στα πλαίσια της διπλωματικής απλά κάνουμε εγγραφή του χρήστη μέσω Facebook και αποθηκεύουμε τα στοιχεία του στην βάση δεδομένων του Parse. Έχουμε έτσι στην διάθεση μας ένα σύστημα login-register για μελλοντική χρήση. Επίσης στην αρχή του κώδικα γίνεται και αρχικοποίηση του Facebook JavaScript SDK το οποίο και μας δίνει τις λειτουργίες σύνδεσης του χρήστη στον λογαριασμό του στο Facebook. Να σημειώσουμε ότι στο πεδίο appId βάζουμε το App ID που πήραμε κατά την δημιουργία της εφαρμογής στον λογαριασμό Facebook του developer της εφαρμογής.

Παρακάτω βλέπουμε τον κώδικα που μεταφορτώνει μια φωτογραφία στον λογαριασμό Facebook κάποιου χρήστη:

### Κώδικας 27 - Facebook Upload Photo

```
function postPhoto(imageData, mimeType, userId) {  
  document.getElementById("upload_loading").style.display = 'inline';  
  blob = imageData;  
  var fd = new FormData();  
  fd.append("source", blob);  
  fd.append("access_token", acctkn);  
  try {  
    $.ajax({  
      url : "https://graph.facebook.com/me/photos",  
      type : "POST",  
      data : fd,  

```

«Σχεδίαση και Υλοποίηση Εφαρμογής για Κινητά Τηλέφωνα Αξιοποιώντας τις Πληροφορίες των Αισθητήρων τους»

```

processData : false,
contentType : false,
cache : false,
success : function (data) {
    console.log("success " + data);
    drawToast('File Uploaded on Facebook!');
    document.getElementById("upload_loading").style.display =
'none';
},
error : function (shr, status, data) {
    console.log("error " + data + " Status " + shr.status);
    drawToast(" An error occured :( ");
    document.getElementById("upload_loading").style.display =
'none';
},
complete : function (shr, status, data) {
    console.log("complete " + data + " Status " + shr.status);
}
});
} catch (e) {
    console.log(e);
}
}

```

Το access token που πρέπει να στείλουμε όπως φαίνεται παραπάνω πρόκειται για ένα μοναδικό token του χρήστη που είναι συνδεδεμένος και μπορούμε να το ανακτήσουμε όπως φαίνεται στον παρακάτω κώδικα:

#### Κώδικας 28 - Facebook Login Status

```

FB.getLoginStatus(function (response) {
    console.log("response : " + response.status);
    if (response.status === 'connected') {
        // the user is logged in and has authenticated your
        // app, and response.authResponse supplies
        // the user's ID, a valid access token, a signed
        // request, and the time the access token
        // and signed request each expire
        var uid = response.authResponse.userID;
        var accessToken = response.authResponse.accessToken;
        acctkn = accessToken;
    }
});

```

Με το FB.getLoginStatus μπορούμε να ελέγξουμε την κατάσταση σύνδεσης του χρήστη με το Facebook και να έχουμε στην διάθεση μας κάποιες πληροφορίες όπως το userID και το accessToken.

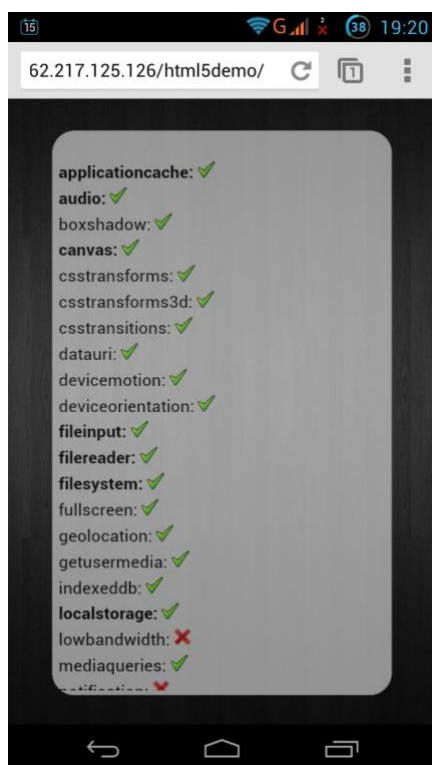


## 6 Λειτουργίες

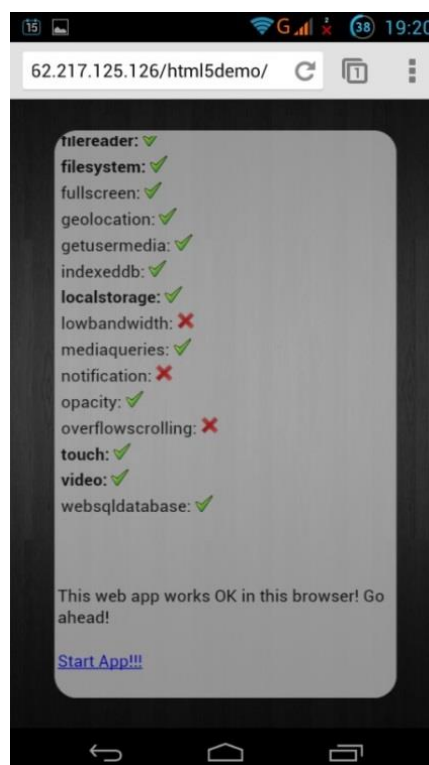
Σε αυτή την ενότητα παρουσιάζεται η λειτουργικότητα της εφαρμογής, οι βασικές σελίδες και η αλληλεπίδραση μεταξύ χρήστη και εφαρμογής. Θα δούμε πως παρουσιάζονται οι διαθέσιμες λειτουργίες στον χρήστη και πως αυτός μπορεί να τις χρησιμοποιήσει. Θα δούμε πως η εφαρμογή επικοινωνεί και ενημερώνει τον χρήστη και πως γίνεται η μετάβαση από την μια λειτουργία στην άλλη.

### 6.1 Εντοπισμός Features

Η αρχική σελίδα ενημερώνει τον χρήστη για το ποια features υποστηρίζονται και για το αν μπορεί η εφαρμογή για τρέξει σωστά στον browser του. Η υποστήριξη του κάθε χαρακτηριστικού απεικονίζεται με κατάλληλα εικονίδια όπως φαίνονται στις παρακάτω εικόνες. Αν καλύπτονται οι απαιτήσεις τότε εμφανίζεται το κατάλληλο μήνυμα και ο χρήστης μπορεί να προχωρήσει στην επομένη σελίδα. Αλλιώς ενημερώνεται ότι δεν είναι δυνατό να λειτουργήσει η εφαρμογή και προτείνεται ο Google Chrome ως υποστηριζόμενος περιηγητής ιστού.



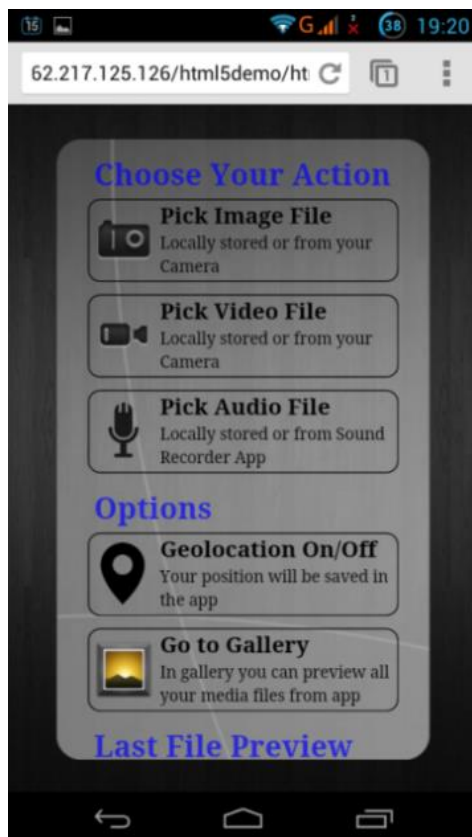
Εικόνα 19 - Features Detection 1



Εικόνα 20 - Features Detection 2

## 6.2 Main Page

Στην επόμενη σελίδα ο χρήστης μπορεί να επιλέξει είτε να βγάλει μια καινούργια φωτογραφία, είτε να τραβήξει ένα νέο βίντεο είτε μια νέα ηχογράφιση. Βλέπουμε στην Εικόνα 21 πως εμφανίζονται οι επιλογές ως κουμπιά που με ένα απλό άγγιγμα πάνω τους ενεργοποιεί τις αντίστοιχες εφαρμογές. Στις Εικόνα 22, Εικόνα 23, Εικόνα 24 βλέπουμε τις εφαρμογές που ανοίγουν στο λειτουργικό σύστημα Android αντίστοιχα για τις φωτογραφίες, βίντεο και ηχογραφήσεις. Επίσης στις Εικόνα 25, Εικόνα 26 βλέπουμε το σύντομο μήνυμα “toast” που εμφανίζεται για επιβεβαίωση της αποθήκευσης του αρχείου καθώς και την σύντομη προεπισκόπηση του αρχείου με κρυμμένη την μπάρα διευθύνσεων.



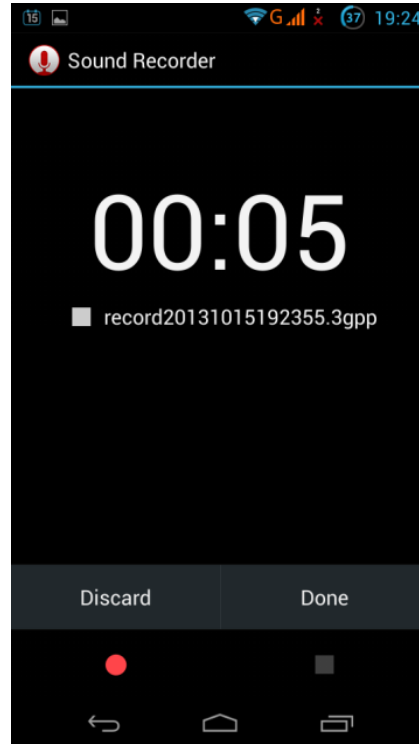
Εικόνα 21 - Κύριες λειτουργίες



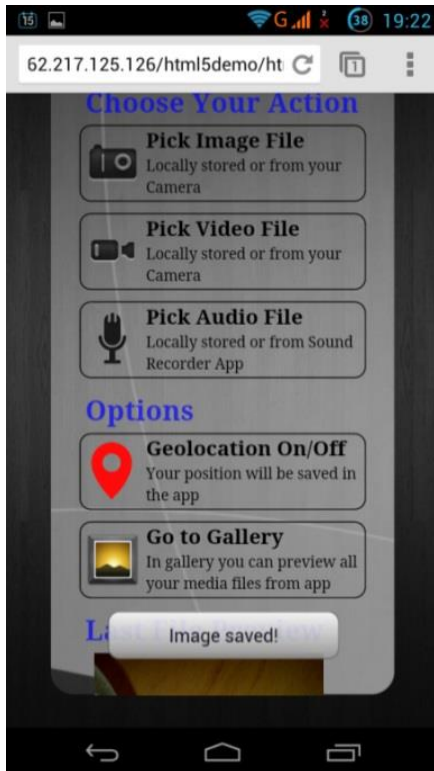
Εικόνα 22 - Φωτογράφιση



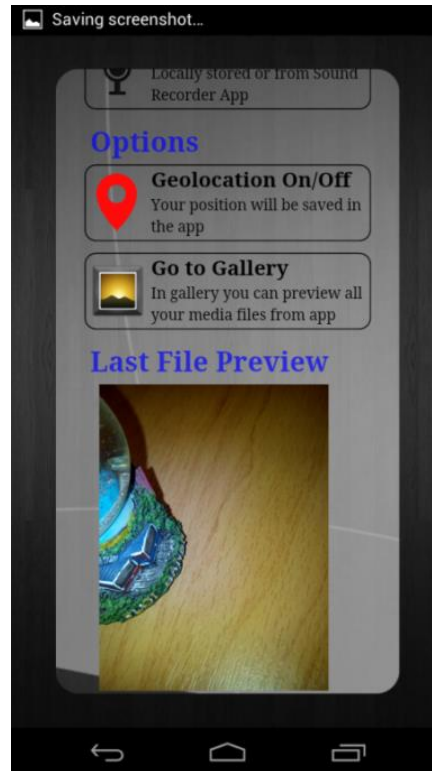
Εικόνα 23 - Βιντεοσκόπηση



Εικόνα 24 - Ηχογράφηση



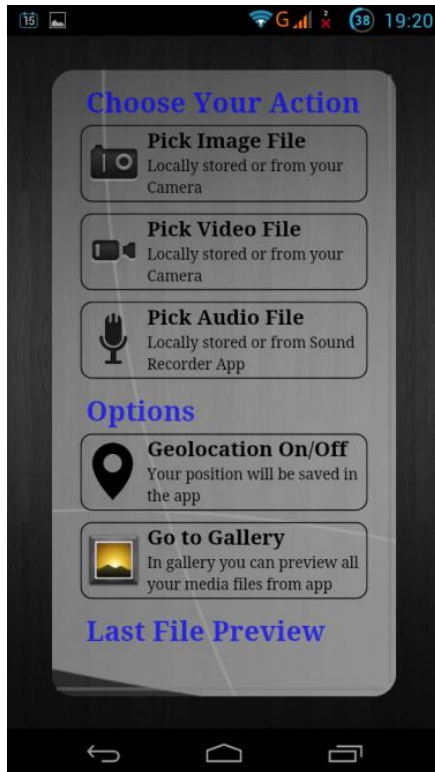
Εικόνα 25 - Toast Notification



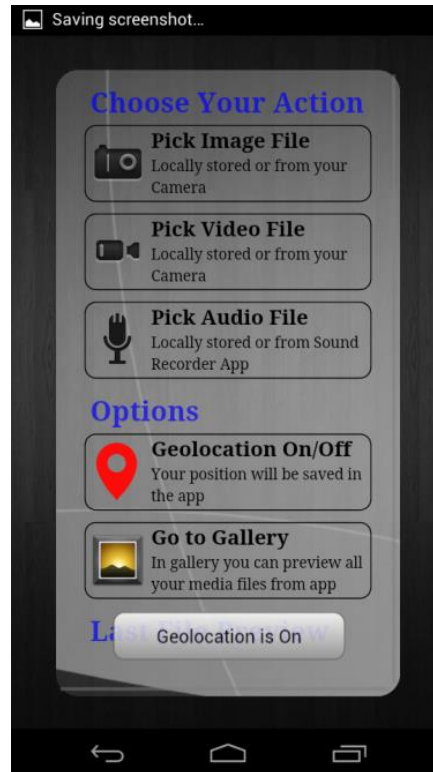
Εικόνα 26 - Mini Preview

Επίσης όπως βλέπουμε στις Εικόνα 27 και Εικόνα 28 ο χρήστης μπορεί να ενεργοποιήσει ή να απενεργοποιήσει το Geolocation πατώντας στην περιοχή «Geolocation On/Off». Όπως βλέπουμε και στις εικόνες το εικονίδιο γίνεται κόκκινο ως ένδειξη ότι έχουμε ενεργοποιημένη την λειτουργία και παραμένει έτσι μέχρι την «Σχεδίαση και Υλοποίηση Εφαρμογής για Κινητά Τηλέφωνα Αξιοποιώντας τις Πληροφορίες των Αισθητήρων τους»

απενεργοποίηση της. Επίσης εμφανίζεται και ένα σύντομο μήνυμα ως επιβεβαίωση της ενεργοποίησης/απενεργοποίησης.



Εικόνα 27- Geolocation Off



Εικόνα 28 - Geolocation On

Τέλος πατώντας πάνω στο «Go to Gallery» ο χρήστης μεταβαίνει σε επομένη σελίδα της εφαρμογής όπου και μπορεί να δει και να διαχειριστεί τα αρχεία που έχει δημιουργήσει από την αρχική σελίδα.

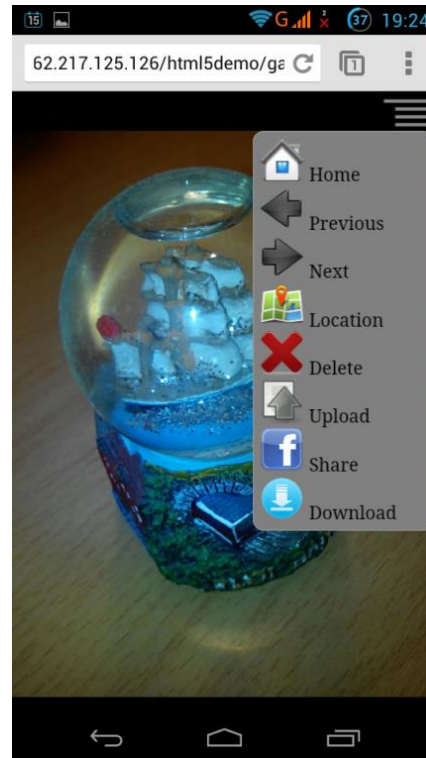
## 6.3 Gallery

Παρακάτω βλέπουμε την σελίδα του Gallery. Εδώ Ο χρήστης μπορεί να περιηγηθεί στα αρχεία του είτε με swipe left/right είτε μέσω των κουμπιών που υπάρχουν στο μενού που ανοίγει πατώντας στο ειδικό κουμπί πάνω δεξιά όπως φαίνεται και στην Εικόνα 30. Επίσης όπως μπορούμε να δούμε και στην Εικόνα 29 μόλις φορτώσει κάποιο αρχείο εμφανίζεται και ένα σύντομο μήνυμα με το όνομα του αρχείου που βλέπει αυτή την στιγμή ο χρήστης.





Εικόνα 29 - Gallery



Εικόνα 30 - Gallery Menu

Παρακάτω περιγράφουμε συνοπτικά την λειτουργία των επιλογών του μενού:

**Home:** επιστροφή στην αρχική σελίδα.

**Next:** Επόμενο αρχείο.

**Previous:** Προηγούμενο αρχείο.

**Location:** Εμφάνιση τοποθεσίας αρχείου. Αν δεν υπάρχει εμφανίζεται κατάλληλο μήνυμα.

**Delete:** Διαγραφή αρχείου.

**Upload:** Ανέβασμα αρχείου στον server.

**Facebook Share:** Ανέβασμα αρχείου στον Facebook λογαριασμό του χρήστη.

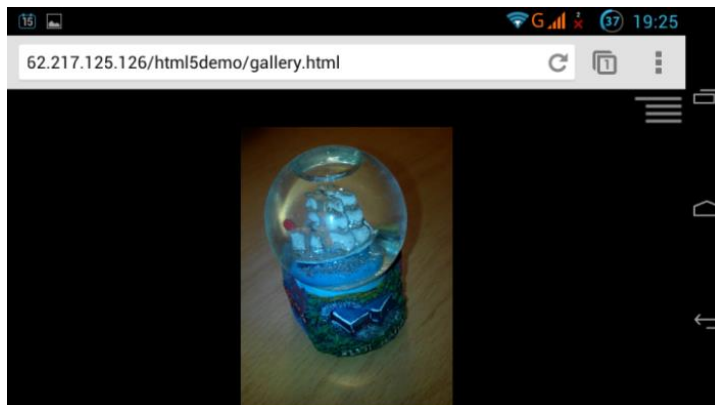
**Download:** Κατέβασμα αρχείου στον αποθηκευτικό χώρο του χρήστη.

### 6.3.1 Αναπαραγωγή αρχείων

Στις παρακάτω εικόνες βλέπουμε πως απεικονίζονται οι φωτογραφίες στην εφαρμογή τόσο σε οριζόντια όσο και σε κάθετη τοποθέτηση της συσκευής. Παρατηρούμε ότι σε οριζόντια θέση η εικόνα προσαρμόζεται στο νέο ύψος ώστε να φαίνεται ολόκληρη. Επίσης υπάρχει και η δυνατότητα απόκρυψης της μπάρας διευθύνσεων με σύρσιμο του δακτύλου προς τα πάνω σε περιοχή εκτός της εικόνας.



Εικόνα 31 - Κάθετη απεικόνιση εικόνας

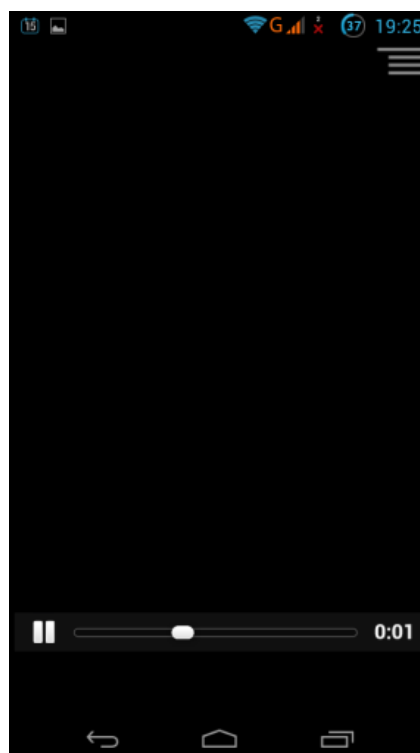


Εικόνα 32 - Οριζόντια απεικόνιση εικόνας

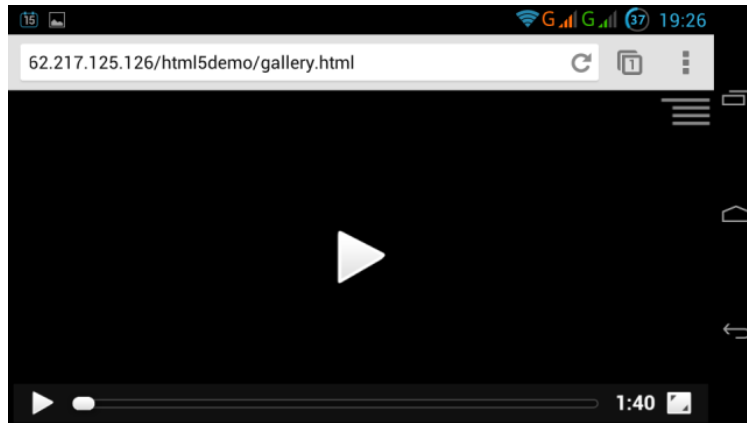
Παρόμοια με τις φωτογραφίες εμφανίζονται και τα βίντεο και οι ηχογραφήσεις όπως μπορούμε να δούμε και από τις παρακάτω εικόνες.



Εικόνα 33 - Video Preview



Εικόνα 34 - Audio Preview



Εικόνα 35 - Landscape Video

### 6.3.2 Location

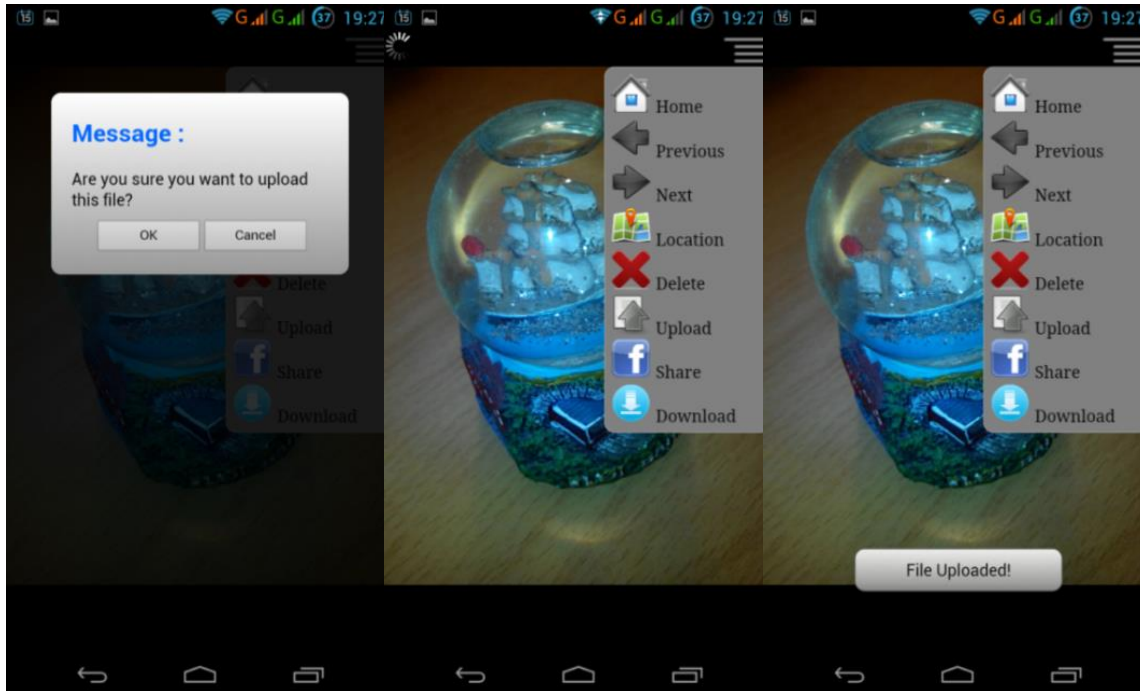
Επιλέγοντας την επιλογή Location ο χρήστης μεταφέρεται σε νέα σελίδα όπου και εμφανίζεται ένας χάρτης με μια κόκκινη κουκίδα η οποία και αντιπροσωπεύει την ακριβή τοποθεσία που είναι συνδεδεμένη με το αρχείο. Πατώντας πάνω στην κουκίδα εμφανίζονται και περισσότερα στοιχεία όπως ώρα, ημερομηνία, συντεταγμένες. Βλέπουμε ότι πάνω δεξιά υπάρχει και ένα εικονίδιο το οποίο πατώντας το οδηγούμαστε στην προηγούμενη σελίδα Gallery. Παρακάτω βλέπουμε την σχετική Εικόνα 36.



Εικόνα 36 - Location on Map

### 6.3.3 Uploading Αρχείων

Επιλέγοντας την επιλογή Upload ο χρήστης ερωτείται αν είναι βέβαιος ότι θελε να ανεβάσει το συγκεκριμένο αρχείο και σε περίπτωση επιβεβαίωσης αρχίζει η διαδικασία μεταφόρτωσης και εμφανίζεται πάνω αριστερά ένα κινούμενο εικονίδιο το οποίο μας δείχνει ότι υπάρχει μια διαδικασία μεταφόρτωσης σε εξέλιξη. Με την ολοκλήρωση εμφανίζεται αντίστοιχο σύντομο μήνυμα. Τα παραπάνω μπορούμε να τα δούμε και από τις Εικόνα 37, Εικόνα 38, Εικόνα 39.



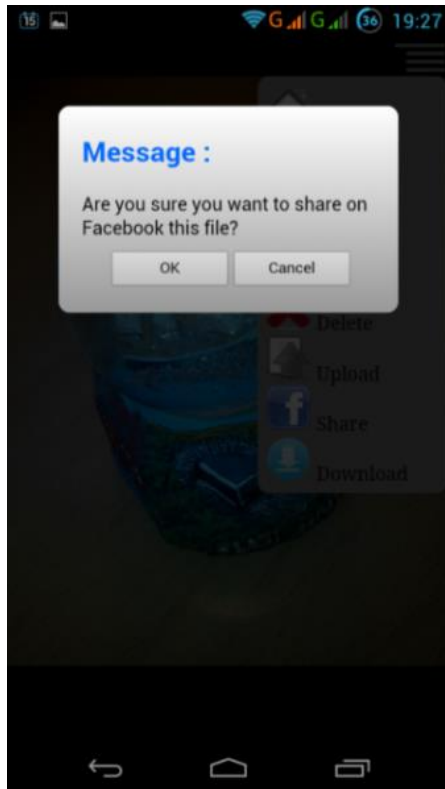
Εικόνα 37 - Upload Confirmation

Εικόνα 38 - Uploading process indicator

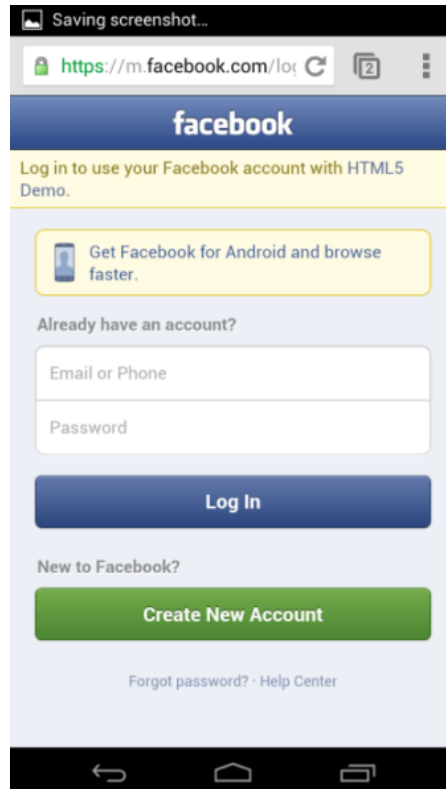
Εικόνα 39 - Upload completed notification

### 6.3.4 Facebook

Παρόμοια μηνύματα και διαδικασίες υπάρχουν και για το ανέβασμα αρχείου στον λογαριασμό Facebook του χρήστη. Στην περίπτωση αυτή αν ο χρήστη δεν είναι ήδη συνδεδεμένος στον λογαριασμό του θα οδηγηθεί σε κατάλληλη σελίδα του Facebook όπως φαίνεται στην Εικόνα 41 για να εισάγει τα στοιχεία του και να κάνει login. Επίσης στις Εικόνα 42 , Εικόνα 43 βλέπουμε πως φαίνονται τα αρχεία στον λογαριασμό Facebook κάποιου χρήστη. Παρατηρούμε ότι εμφανίζεται και το όνομα της εφαρμογής με την οποία έχουν ανέβει, HTML5 Demo στην περίπτωση μας.



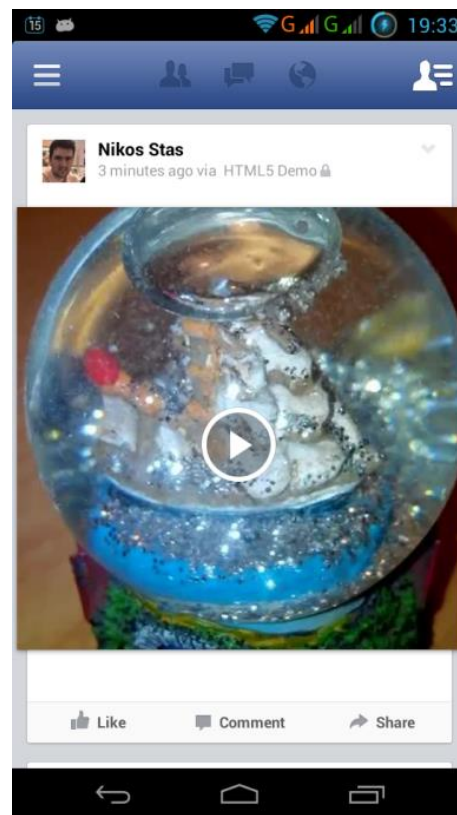
Εικόνα 40 - Confirm Facebook Share



Εικόνα 41 - Facebook Login Screen



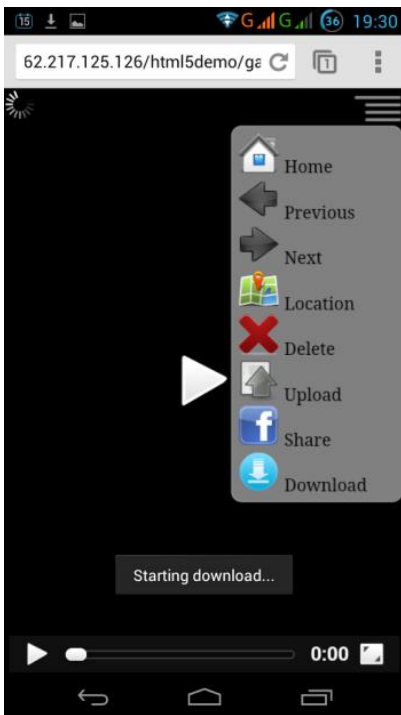
Εικόνα 42 - Photo on Facebook



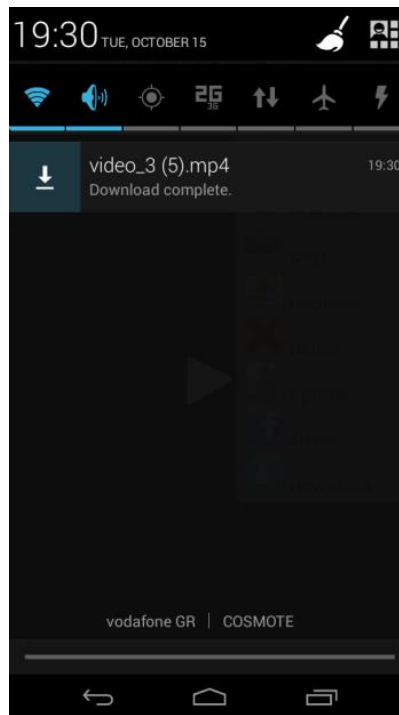
Εικόνα 43 - Video on Facebook

### 6.3.5 Λήψη Αρχείου

Τέλος όπως βλέπουμε και στις παρακάτω εικόνες ο χρήστης έχει την δυνατότητα να κατεβάσει τα αρχεία του στον αποθηκευτικό χώρο της συσκευής του πατώντας στην επιλογή “Download” και να χρησιμοποιεί πλέον και μέσα από το λειτουργικό σύστημα της συσκευής όπως βλέπουμε στην Εικόνα 46. Εδώ να σημειώσουμε και την δυνατότητα που υπάρχει για παράλληλες λειτουργίες όπως για παράδειγμα upload ενός αρχείου και download ενός άλλου όπως φαίνεται και στην Εικόνα 44.



Εικόνα 44 - Download File



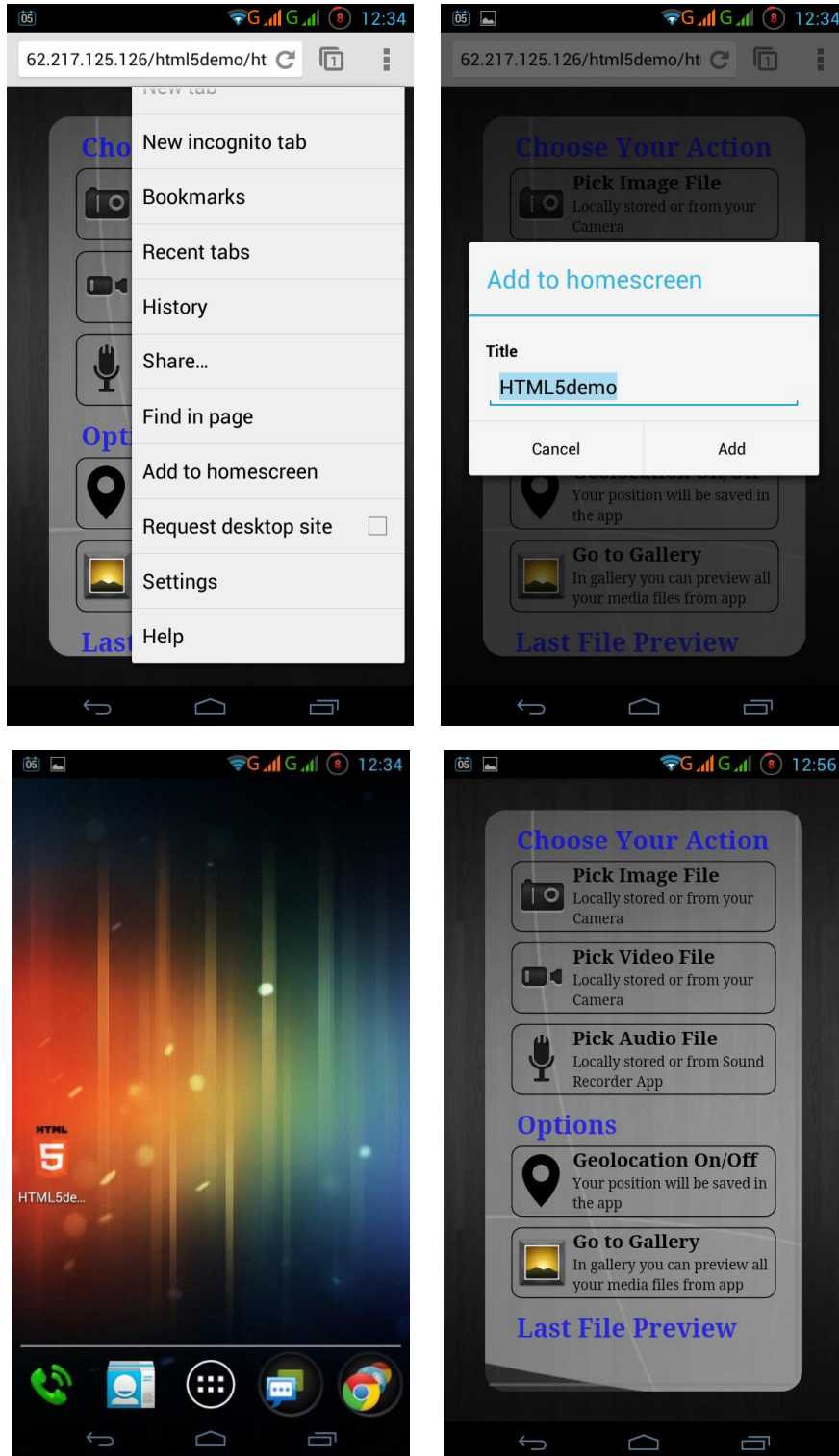
Εικόνα 45 - Download Complete



Εικόνα 46 - Preview with native app

## 6.4 Λειτουργία Full Screen

Ο χρήστης έχει την δυνατότητα σε Android Chrome Beta v32 ή σε νεότερες εκδόσεις άλλων browsers όταν υποστηριχτεί το HTML5 feature να επιλέξει να αποθηκεύσει ένα shortcut icon στην home-screen του λειτουργικού του ώστε να τρέχει την εφαρμογή ως εφαρμογή συστήματος (εικονικά). Αυτό ενεργοποιεί την Full Screen ιδιότητα. Παρακάτω βλέπουμε πως μπορεί αυτό να γίνει στον Android Chrome Beta v32 και πως φαίνεται η εφαρμογή σε full screen mode.



Εικόνα 47 - Full Screen Mode

## 6.5 Video-Demo Λειτουργιών

Στα πλαίσια της παρουσίασης των λειτουργιών που υλοποιήσαμε, τις οποίες περιγράψαμε παραπάνω και είναι διαθέσιμες στον τελικό χρήστη δημιουργήσαμε και μια video-παρουσίαση αυτών των δυνατοτήτων. Το βίντεο δείχνει λεπτομερέστατα και οργανωμένα σε κατηγορίες την διαδικασία που πρέπει να ακολουθήσει ένας χρήστης ώστε να ολοκληρώσει κάποια λειτουργία όπως λήψη φωτογραφίας/βίντεο/ηχογράφησης, διαχείριση αρχείων, upload σε dedicated server και Facebook κ.α. Το βίντεο δημιουργήθηκε με την βοήθεια του AirDroid<sup>11</sup> με το οποίο και έχουμε πρόσβαση στην οθόνη του κινητού τηλεφώνου μέσω ειδικού web site από το οποίο και καταγράψαμε την όλη διαδικασία. Στο βίντεο βλέπουμε πως συμπεριφέρεται η εφαρμογή σε πραγματικό περιβάλλον, δηλαδή σε λειτουργικό σύστημα Android 4.2 και με την χρήση του Google Chrome for Android.

Το βίντεο είναι διαθέσιμο στην παρακάτω διεύθυνση:

[http://ru6.cti.gr/mobile/uploads/demo\\_html5app.mp4](http://ru6.cti.gr/mobile/uploads/demo_html5app.mp4)

Η εφαρμογή είναι διαθέσιμη στην διεύθυνση:

<http://62.217.125.126/html5demo/>

---

<sup>11</sup> <http://www.airdroid.com/>



## 7 Συμπεράσματα

Στην παρούσα διπλωματική εργασία μελετήθηκε η HTML5 ως προγραμματιστικό εργαλείο για την ανάπτυξη εφαρμογών σε φορητές συσκευές και πιο συγκεκριμένα κινητά τηλέφωνα αλλά και tablets. Επίσης παρουσιάσαμε μεθόδους σχεδιασμού mobile web εφαρμογών αλλά και άλλες πολύ χρήσιμες και πολύ-χρησιμοποιημένες τεχνολογίες που βασίζονται στην HTML5.

Αρχικά, αναλύθηκαν τα πιο βασικά νέα χαρακτηριστικά που εισάγει η HTML5. Ερευνήσαμε τα νέα APIs σε JavaScript και τις λειτουργίες που αυτά προσφέρουν ενεργοποιώντας δυνατότητες της συσκευής, μέσω του web, που παλαιότερα μόνο σε μια native υλοποίηση θα μπορούσαμε να δούμε. Επίσης ασχοληθήκαμε με τα νέα HTML elements καθώς και τα νέα χαρακτηριστικά του CSS3 με τα νέα animations και τις νέες ιδιότητες για εμπλουτισμό του user interface μιας εφαρμογής.

Στην συνέχεια αναφερθήκαμε σε frameworks που στηρίζονται ή συνεργάζονται με την HTML5 για την ανάπτυξη εφαρμογών σε κινητές συσκευές. Εξετάσαμε εργαλεία όπως το jQuery Mobile, jQT, Sencha Touch και 5ml που βοηθούν στην γρηγορότερη σχεδίαση του UI βοηθώντας παράλληλα στην δημιουργία ενός πιο οργανωμένου και πλούσιου περιβάλλοντος. Επίσης εξετάσαμε και εργαλεία τα οποία είτε προσφέρουν επιπρόσθετες υπηρεσίες όπως το Parse ( cloud υπηρεσίες ) ή το Adobe PhoneGap με το οποίο και μπορούμε να μετατρέψουμε μια εφαρμογή διαδικτύου σε native αξιοποιώντας έτσι κάποια επιπλέον χαρακτηριστικά και αποκτώντας πρόσβαση στα καταστήματα εφαρμογών των λειτουργικών συστημάτων, όπως το Android, iOS, Windows Phone κτλ.

Επιπλέον αναφερθήκαμε σε γενικά σχεδιαστικά θέματα εφαρμογών για κινητά τηλέφωνα. Αναφερθήκαμε στα είδη εφαρμογών που μπορούμε να δημιουργήσουμε ανάλογα την πλατφόρμα που θα επιλέξουμε. Έτσι έχουμε πλήρως web εφαρμογή, full screen web application, hybrid web app σε μορφή native package και πλήρως native app. Επίσης αναφερθήκαμε σε τρόπους σχεδιασμού ώστε κάποια εφαρμογή να είναι προσβάσιμη από όσο το δυνατόν περισσότερες συσκευές με διαφορετικά χαρακτηριστικά, όπως το Progressive Enhancement. Τέλος αναλύσαμε τον σχεδιασμό που ακλουθήσαμε στην εφαρμογή μας.

Επίσης αναφερθήκαμε αναλυτικά στην υλοποίηση της εφαρμογής παρουσιάζοντας κομμάτια κώδικα και αναλύοντας τα. Εξηγήσαμε ποιες λειτουργίες αυτά εξυπηρετούν και πως πετύχαμε την υλοποίηση αυτών. Ο σκοπός ήταν να δείξουμε πως μπορούν πολλά από τα νέα χαρακτηριστικά και τεχνολογίες να συνδυαστούν μεταξύ τους αλλά και με τρίτες υπηρεσίες όπως το Facebook και το Parse, ώστε να δημιουργηθεί μια ολοκληρωμένη εφαρμογή. Κλείνοντας την ανάλυση παρουσιάσαμε τις λειτουργίες που έχει στην διάθεση του ο χρήστης στην εφαρμογή. Η παρουσίαση αυτή έγινε με

εικόνες μέσα από την εφαρμογή ώστε να υπάρχει καλύτερη εικόνα για την εμπειρία χρήστη.

Όπως διαπιστώσαμε η ευκολία ανάπτυξης σε HTML5 είναι μεγάλη για κάποιον που έχει μια βασική εξοικείωση με τεχνολογίες διαδικτύου. Αυτό έχει ως αποτέλεσμα να χρειάζεται πολύ λιγότερος χρόνος για την δημιουργία κάποιας εφαρμογής και μπορούμε έτσι να επικεντρωνόμαστε περισσότερο στις ιδέες και χαρακτηριστικά αυτής. Επιπλέον μπορούμε να επικεντρωθούμε περισσότερο στην εμπειρία του χρήστη αλλά και την απόδοση της εφαρμογής. Επίσης το γεγονός ότι μπορεί κάποιος να αποκτήσει πρόσβαση σε ένα πολύ μεγάλο πλήθος συσκευών διαφορετικών λειτουργικών συστημάτων αλλά και χαρακτηριστικών είναι πολύ δελεαστικό για τον απλό λόγο ότι έχουμε τεραστία αύξηση της βάσης των χρηστών της εφαρμογής. Αυτό είναι και ένα από τα μεγαλύτερα πλεονεκτήματα έναντι της ανάπτυξης εφαρμογών σε native platforms μαζί με την δυσκολία προγραμματισμού που εισάγουν.

Μαζί με τα πλεονεκτήματα όμως υπάρχουν και κάποια μειονεκτήματα. Το γεγονός ότι αναφερόμαστε σε τόσες διαφορετικές συσκευές σημαίνει ότι πρέπει να λάβουμε υπόψη μας κυρίως τις διαφορές στις διαστάσεις της οθόνης αλλά και τα χαρακτηριστικά που πιθανώς να μην υποστηρίζονται. Βεβαία αυτά είναι ένα αναγκαίο κακό που δεν μπορούμε να αποφύγουμε λόγω της υποστήριξης τόσων platforms. Επίσης και ίσως το σημαντικότερο είναι ότι μια native εφαρμογής ακόμα υπερτερεί σε απόδοση σε ότι έχει να κάνει με το smoothness του user interface αν και σε αυτόν τον τομέα έχουν γίνει σημαντικά βήματα και ίσως στο μέλλον το κενό μικρύνει μεταξύ τους [1]. Τέλος ακόμα υπάρχουν κάποια χαρακτηριστικά που δεν υποστηρίζονται στην HTML5 ενώ υπάρχουν στις native platforms.

Καταλήγοντας, το μέλλον στην ανάπτυξη εφαρμογών φαίνεται να είναι η HTML5 με πολλές εταιρείες αλλά και ανεξάρτητοι developers να στρέφονται σε αυτή λόγω της ευκολίας αλλά και της cross-platform υποστήριξης. Πολλά νέα λειτουργικά συστήματα άλλωστε όπως Tizen, Ubuntu και Firefox OS έχουν στραφεί ήδη στην HTML5 για τις εφαρμογές τους

## 8 Μελλοντική Εργασία

Καταλήγοντας, θα παρουσιάσουμε κάποια πιθανά θέματα για μελλοντική έρευνα πάνω στο mobile application development και στην περαιτέρω βελτίωση της εφαρμογής που υλοποιήσαμε.

Ένα θέμα που θα μπορούσε να ερευνηθεί είναι η υποστήριξη όσο ο δυνατόν περισσότερων browsers γίνεται. Το κυριότερο θέμα με την εφαρμογή μας είναι η μη υποστήριξη του FileSystem API από όλους τους browsers. Προς το παρόν μόνο Chrome και ο Opera Mobile παρέχουν το συγκεκριμένο API. Μια λύση θα μπορούσε να είναι ένα polyfill που στηρίζεται σε κάποιο άλλο υποστηριζόμενο χαρακτηριστικό όπως το Indexed Database API.

Επίσης αντί του Media Capture API θα μπορούσε να χρησιμοποιηθεί το getUsermedia (Media Capture and Streams) το οποίο και θα έδινε την δυνατότητα για υλοποίηση επιπλέον υπηρεσιών όπως live video streaming ακόμα και από χρήστη σε χρήστη. Φυσικά είναι δυνατή και η εγγραφή ήχου/βίντεο/εικόνας (με την βοήθεια του RecordRTC για παράδειγμα) ίσως όμως τίθενται θέματα ως προς το πόσο μπορούν να αξιοποιηθούν όλες οι δυνατότητες της συσκευής. Για παράδειγμα αν μπορούμε να έχουμε φωτογραφίες στην πλήρη υποστηριζόμενη ποιότητα/ανάλυση της συσκευής. Κι εδώ η υποστήριξη είναι περιορισμένη στον Chrome προς το παρόν.

Επιπρόσθετα θα μπορούσε να υλοποιηθεί και μια υβριδική native εφαρμογή χρησιμοποιώντας κάποια από τα εργαλεία/frameworks που περιγράψαμε παραπάνω. Κάτι τέτοιο θα ήταν ιδιαίτερα εύκολο μιας και τα περισσότερα εργαλεία στηρίζονται στην HTML5 οπότε ο επιπρόσθετος κώδικας που θα χρειαστούμε θα είναι ελάχιστος. Επίσης θα μπορούσε να ερευνηθεί η καλύτερη, πιο στοχευμένη υποστήριξη περισσότερων διαφορετικών αναλύσεων οθονών με την χρήση των Media Queries.

Τέλος, θα μπορούσε να μελετηθεί ένα Debug API για να βοηθήσει στην ανάπτυξη εργαλείων εντοπισμού σφαλμάτων και αξιολόγησης της απόδοσης. Κάτι τέτοιο θα φάνει ιδιαίτερα χρήσιμο σε περιπτώσεις debugging εφαρμογών που είναι «πακεταρισμένες» σε native εφαρμογές αλλά και σε γενικότερη διευκόλυνση αποδεδεσμένων τους προγραμματιστές από τα εργαλεία του κάθε browser. Ολοκληρώνοντας, άλλο ένα επίσης σημαντικό θέμα είναι η δημιουργία ενός επίσημου γενικευμένου τρόπου δημιουργίας πακέτου εγκατάστασης των web applications στις πλατφόρμες που υποστηρίζονται, ώστε να μην στηρίζομαστε σε τρίτες υλοποιήσεις.



## ΒΙΒΛΙΟΓΡΑΦΙΑ

---

---

### **Βιβλία:**

- [1] Appcelerator Inc., Native vs HTML5 Mobile App Development Which option is best? , 2012
- [2] Charland A. and Leroux B. Mobile Application Development: Web vs. Native, April 2011
- [3] Fling, B. Mobile Design and Development: Practical Techniques for Creating Mobile Sites and Web Apps. O'Reilly Media Inc., 2009.
- [4] Hooker S. and Berkman E., Designing Mobile Interfaces. O'Reilly Media, 2011
- [5] Gokul Seenivasan, Aspire Systems, HTML5 - the new standard for Interactive Web
- [6] Josh Dehlinger and Jeremy Dixon, Department of Computer and Information Sciences, Mobile Application Software Engineering: Challenges and Research Directions, 2011
- [7] Jukka Paasonen, HTML5 as Common User Interface Layer in Mobile Device Platforms, 2012
- [8] Kimmo Puputti, Master's Thesis: Mobile HTML5:Implementing a Responsive Cross-Platform Application, 2012
- [9] Mark Power, Mobile Web Apps, JISC CETIS 2012
- [10] Parker T., Toland P., Jehl S. and Wachs M. C. Designing with Progressive Enhancement: Building the Web That Works for Everyone. New Riders Publishing, 2010.
- [11] Pilgrim M., HTML5: Up And Running. O'Reilly Media / Google Press, 2010.
- [12] Pooja Prasad, Technical Lead, RapidValue Solutions, Choosing The Right HTML5 Framework To Build Your Mobile Web Application, July 2013
- [13] Shwetank Dixit, The Mobile Web: Key Industry and Technological Trends
- [14] Souders S., Even Faster Web Sites. O'Reilly Media, 2009.

- [15] Maximiliano Firtman, Programming the Mobile Web, Second Edition. O'Reilly Media 2013

### **Ιστόχωροι:**

- [16] Andersson O., Scalable Vector Graphics (SVG) Tiny 1.2 Specification, W3C.  
<http://www.w3.org/TR/SVGTiny12/>
- [17] Bergkvist A., Burnett D. C., Jennings C., Narayanan A., WebRTC 1.0: Real time Communication Between Browsers , W3C.  
<http://dev.w3.org/2011/webrtc/editor/webrtc.html>
- [18] Block S. and Popescu A., DeviceOrientation Event Specification , W3C.  
<http://dev.w3.org/geo/api/spec-source-orientation.html>
- [19] Brubeck M., Moon S., and Schepers D. Touch Events, W3C.  
<http://www.w3.org/TR/touch-events/>
- [20] Burnett D. C., Narayanan A., getusermedia: Getting access to local devices that can generate multimedia streams , W3C.  
<http://dev.w3.org/2011/webrtc/editor/getusermedia.html>
- [21] Carlisle D., Ion P., Miner R., Mathematical Markup Language (MathML), W3C.  
<http://www.w3.org/TR/MathML/>
- [22] Daggett J., CSS Fonts Module Level 3 , W3C. <http://dev.w3.org/csswg/css3-fonts/>
- [23] Fraser S., Jackson D., Hyatt D., Marrin C., O'Connor E. and Schulze D., CSS Transforms , W3C. <http://dev.w3.org/csswg/css3-transforms/>
- [24] Gregg J., Web Notifications , W3C.  
<http://dev.w3.org/2006/webapi/WebNotifications/publish/Notifications.html>
- [25] Herman D. and Russell K., Typed Array Specification. Editor's Draft.  
<https://www.khronos.org/registry/typedarray/specs/latest/>
- [26] Hickson I., HTML 5 , W3C. <http://dev.w3.org/html5/spec/Overview.html>
- [27] Hickson I., HTML Canvas 2D Context , W3C.  
<http://dev.w3.org/html5/2dcontext/>
- [28] Hickson I., Server-Sent Events , W3C. <http://dev.w3.org/html5/eventsource/>
- [29] Hickson I., The WebSocket API , W3C. <http://dev.w3.org/html5/websockets/>
- [30] Hickson I., Web Storage , W3C. <http://dev.w3.org/html5/webstorage/>

- [31] Hickson I., Web Workers , W3C. <http://dev.w3.org/html5/workers/>
- [32] Jackson D., Hyatt D., Marrin C. and Baron L. D., CSS Transitions , W3C. <http://dev.w3.org/csswg/css3-transitions/>
- [33] Jackson D., Hyatt D., Marrin C., Galineau S. and Baron L. D., CSS Animations , W3C. <http://dev.w3.org/csswg/css3-animations/>
- [34] Kostiainen A., Vibration API, W3C. <http://dev.w3.org/2009/dap/vibration/>
- [35] Kostiainen A., and Lamouri, M. Battery Status API, W3C. <http://dvcs.w3.org/hg/dap/raw-file/tip/battery/Overview.html>
- [36] Lamouri M., The Network Information API, W3C. <http://www.w3.org/TR/netinfo-api/>
- [37] Mann, J. and Jain A., Page Visibility, W3C. <http://dvcs.w3.org/hg/webperf/raw-file/tip/specs/PageVisibility/Overview.html>
- [38] Mann J. and Wang Z., Performance Timeline, W3C. <http://dvcs.w3.org/hg/webperf/raw-file/tip/specs/PerformanceTimeline/Overview.html>
- [39] Mann J. and Weber J., Efficient Script Yielding, W3C. <http://dvcs.w3.org/hg/webperf/raw-file/tip/specs/setImmediate/Overview.html>
- [40] Marrin C., WebGL Specification. <https://www.khronos.org/registry/webgl/specs/latest/>
- [41] Mehta N., Sicking J., Graff E., Popescu A. and Or-low J., Indexed Database API, W3C. <http://dvcs.w3.org/hg/IndexedDB/raw-file/tip/Overview.html>
- [42] Popescu A., Geolocation API Specifcation, W3C. <http://www.w3.org/TR/geolocation-API/>
- [43] Ranganathan A. and Sicking J., File API, W3C. <http://dev.w3.org/2006/webapi/FileAPI/>
- [44] Robinson J. and McCormack C., Timing control for script-based animations, W3C. <http://dvcs.w3.org/hg/webperf/raw-file/tip/specs/RequestAnimationFrame/Overview.html>
- [45] Scheib V., Pointer Lock, W3C. <http://dvcs.w3.org/hg/pointerlock/raw-file/default/index.html>
- [46] Uhrhane E., File API: Directories and System, W3C. <http://dev.w3.org/2009/dap/file-system/file-dir-sys.html>

- [47] Uhrhane E., File API: Writer, W3C. <http://dev.w3.org/2009/dap/file-system/file-writer.html>
- [48] Van Kesteren A. and Celik T., Fullscreen, W3C. <http://dvcs.w3.org/hg/fullscreen/raw-file/tip/Overview.html>
- [49] Wang Z., Navigation Timing, W3C. <http://www.w3c-test.org/webperf/specs/NavigationTiming/>
- [50] Wium Lie H., Celik T., Glazman D. and Van Kesteren A., Media Queries, W3C. <http://www.w3.org/TR/css3-mediaqueries/>
- [51] Celik T., Etemad E. J., Glazman D., Hickson I., Linss P. and Williams J. Selectors Level 3, W3C. <http://www.w3.org/TR/selectors/>