



Πανεπιστήμιο Πελοποννήσου  
Σχολή Οικονομίας, Διοίκησης και Πληροφορικής  
Τμήμα Πληροφορικής και Τηλεπικοινωνιών  
Π.Μ.Σ. στην Επιστήμη και Τεχνολογία Υπολογιστών

## Μεταπτυχιακή Εργασία

«Υβριδική Εφαρμογή Τουριστικής Πλοήγησης για  
όλες τις πλατφόρμες κινητών Λειτουργικών Συστημάτων»

“Hybrid Tourist Navigation App for all mobile O.S. platforms”

Καραγιάννης Θεόδωρος

ΑΜ: 2022201402004

Βασιλάκης Κωνσταντίνος  
Τσελίκας Νικόλαος, Αντωνίου Αγγελική

Τρίπολη, Απρίλιος 2017

# Πίνακας Περιεχομένων / Table of Contents

Περίληψη.....	3
Λέξεις Κλειδιά.....	3
Abstract.....	4
Keywords.....	4
Κεφάλαιο 1 / Chapter 1.....	5
1.1 Εισαγωγή / Introduction.....	5
1.2 Παράδειγμα Εφαρμογής / Case Study.....	8
Κεφάλαιο 2 / Chapter 2.....	10
2.1 Σχεδιασμός Απαιτήσεων / Requirements Design.....	10
2.2 Προδιαγραφές Μικρο-Εφαρμογής / App Specs.....	16
Κεφάλαιο 3 / Chapter 3.....	20
3.1 Τεχνολογίες ανάπτυξης / Development Technologies.....	20
3.1.1 Native vs Web vs Hybrid Mobile Apps.....	20
3.1.2 Διαδικτυακές Τεχνολογίες / Web Technologies.....	23
3.2 Εργαλεία και πλαίσια ανάπτυξης / Development Tools and Frameworks.....	32
Κεφάλαιο 4 / Chapter 4.....	37
Ανάπτυξη και Υλοποίηση μικρο-εφαρμογής / App Development and Implementation.....	37
Κεφάλαιο 5 / Chapter 5.....	54
Παραδείγματα χρήσης της μικρο-εφαρμογής / App Use Cases.....	54
Κεφάλαιο 6 / Chapter 6.....	58
Συμπεράσματα - Μελλοντικές κατευθύνσεις / Conclusions - Future Directions....	58
Βιβλιογραφία / References & Ιστοσελίδες / Websites.....	61
Παραρτήματα / Appendixes.....	63
Πίνακας Εικόνων / Table of Figures.....	63

## Περίληψη

Η παρούσα εργασία αποτελεί προσπάθεια δημιουργίας μιας Υβριδικής Εφαρμογής Τουριστικής Πλοήγησης για όλες τις βασικές πλατφόρμες των σύγχρονων Mobile Λειτουργικών Συστημάτων, δηλαδή του iOS της εταιρείας Apple, του Android της εταιρείας Google και της σύμπραξης O.H.A., καθώς και του Windows Phone της εταιρείας Microsoft.

Αφού γίνει μια επισκόπηση του επιστημονικού / ερευνητικού και τεχνολογικού χώρου στον οποίο εντάσσεται η συγκεκριμένη προσπάθεια, θα καταδειχθεί η ανάγκη βάσει της οποίας προέκυψε ο σχεδιασμός και η ανάπτυξη αυτής της εφαρμογής, καθώς και το τι αποτελεί σκοπό της.

Στη συνέχεια, θα παρουσιαστεί ως μελέτη παραδείγματος (Case Study) η εφαρμογή προσαρμοσμένη στις απαιτήσεις ενός συγκεκριμένου μουσειακού/τουριστικού χώρου, θα εξηγηθεί επαρκώς ποια είναι η μεθοδολογία σχεδιασμού και ανάπτυξης της εφαρμογής, πάνω σε ποιες τεχνολογίες βασίστηκε, καθώς και με ποια εργαλεία υλοποιήθηκε.

Ακολούθως, θα δοθούν - με οπτικό κυρίως τρόπο - κάποια τυπικά σενάρια χρήσης της εφαρμογής, ενώ τέλος, θα αναφερθούν τα συμπεράσματα σχετικά με την εργασία και οι μελλοντικές κατευθύνσεις που μπορεί να πάρει αυτή.

## Λέξεις Κλειδιά

Πολιτιστική Κληρονομιά - Τουριστικοί Οδηγοί - Ηλεκτρονικοί/Φορητοί Τουριστικοί Οδηγοί - Έξυπνα Κινητά - Υβριδικές Κινητές Μικρο-Εφαρμογές

## Abstract

This paper is an attempt to create a Hybrid Application for Touristic Navigation for all key platforms of modern Mobile Operating Systems, namely iOS by Apple Inc., Android by the Google Inc. and the O.H.A. partnership, and the Windows Phone O.S. by Microsoft Corporation.

After an initial overview of the scientific / research and technological field, which this specific application belongs to, the need for the design and the development of this application will be pointed out, along with its purpose.

Following that, the application will be presented as a Case Study, adapted to the requirements of a particular museum / touristic area. The methodology of design and development of the application, will be adequately explained, together with the implementation tools and the technologies on which it was based.

Finally, typical application usage scenarios will be given – mainly by visual means – leading to a conclusions report regarding this paper, and the future directions that this application might have.

## Keywords

Cultural Heritage – Tourist Guides – Electronic/Mobile Tourist Guides – SmartPhones – Ubiquitous Computing – Hybrid Mobile Apps – GPS – Geolocation API – Google Maps API – WebStorage API – Apache Cordova – HTML5/CSS3/Javascript – Android, iOS, WindowsPhone – Intel® XDK

# Κεφάλαιο 1 / Chapter 1

## 1.1 Εισαγωγή / Introduction

Η δεύτερη δεκαετία του 21<sup>ου</sup> αιώνα χαρακτηρίζεται έντονα από την «εισβολή» στη ζωή μας των έξυπνων κινητών τηλεφώνων ή όπως είθισται να αποκαλούνται “smartphones”.

Αυτές οι μικρές και φορητές ηλεκτρονικές συσκευές – θαύματα της σύγχρονης τεχνολογίας – έχουν πρωταγωνιστικό ρόλο σχεδόν σε κάθε έκφανση της καθημερινότητάς μας. Αποτελούν πλέον, τον πιστό σύντροφο σε κάθε μας δραστηριότητα, σε προσωπικό, κοινωνικό και επαγγελματικό επίπεδο.

Πέρα από τα επιμέρους εντυπωσιακά χαρακτηριστικά και δυνατότητες που διακρίνουν τις συγκεκριμένες συσκευές, αυτό που τους προσδίδει το μέγιστο πλεονέκτημά τους (αυτό που τις χαρακτηρίζει ως «έξυπνες» δηλαδή) είναι η δυνατότητά τους να δέχονται και να «εκτελούν» μικρο-προγράμματα ή μικρο-εφαρμογές, τα περίφημα “applications”, εν συντομία “apps”.

Apps σχεδιασμένα και υλοποιημένα από εκατομμύρια αυτόνομους, ανεξάρτητους προγραμματιστές, χιλιάδες εταιρείες (startups, μικρού/μεσαίου μεγέθους ή ακόμα και πολυεθνικές) και οργανισμούς ή τμήματα του δημόσιου τομέα όλων σχεδόν των χωρών του κόσμου.

Όλα αυτά τα apps ανέρχονται πλέον σε κάποια εκατομμύρια – αν και δύσκολο να υπολογιστεί το πραγματικό τους πλήθος – και η επίσημη διανομή τους είναι μέσα από τα «καταστήματα μικρο-εφαρμογών» των μεγάλων εταιρειών του χώρου, όπως Apple, Google, Microsoft, Amazon, κλπ. Τον Ιούνιο του 2016 υπολογίζεται [1] πως το Google Play Store διέθετε προς αγορά/μεταφόρτωση (download) περί τα 2.2 εκατομμύρια apps, ενώ αντίστοιχα το Apple App Store κάτι λιγότερο, περίπου 2 εκατομμύρια apps!

Αυτή η υπερπροσφορά των apps έχει να κάνει με το γεγονός πως βρίσκουν πεδίο εφαρμογής σε ό,τι μπορεί να σκεπτεί ο ανθρώπινος νους: apps για την ενημέρωσή μας, apps για τη διασκέδασή μας, apps για την υγεία μας, apps για την επικοινωνία μας, apps για το επάγγελμά μας, apps για τον ελεύθερο χρόνο μας, apps για τις διαπροσωπικές και κοινωνικές σχέσεις μας, apps για τα ταξίδια μας, apps ακόμα και για την κουζίνα μας! Κυριολεκτικά ανεξάντλητα τα πεδία εφαρμογής τους!

Αυτή ακριβώς η τεράστια ποικιλία τους (και βέβαια η χρησιμότητα πολλών εξ αυτών) είναι το μεγάλο τους όπλο, ο λόγος για τον οποίο έχουν γίνει ιδιαίτερα δημοφιλή και αγαπητά, σε σημείο μάλιστα που για ορισμένα από αυτά τα apps, ο χρήστης τους δεν διανοείται πλέον την καθημερινότητά του χωρίς να έχει πρόσβαση σε αυτά!

Ένας από τους χώρους στους οποίους έχουν διεισδύσει οι μικρο-εφαρμογές για έξυπνα κινητά τηλέφωνα, είναι και ο χώρος του τουρισμού και ειδικότερα ο χώρος της τουριστικής πληροφόρησης.

Με μια απλή διαδικτυακή αναζήτηση κανείς μπορεί να διαπιστώσει πως υπάρχουν εκατοντάδες (ίσως και χιλιάδες) apps που προσφέρουν τις υπηρεσίες τους στον τουρίστα, τον επισκέπτη, τον ταξιδιώτη: apps που αναλαμβάνουν την εύρεση και κράτηση (φθηνών) εισιτηρίων (αεροπορικών/σιδηροδρομικών/ακτοπλοϊκών κλπ) καθώς και ξενοδοχείων ή χώρων διαμονής, apps για την κράτηση ενοικιαζόμενων αυτοκινήτων, apps για τον καιρό/την ώρα/τη γλώσσα/την κουζίνα/τις δημόσιες συγκοινωνίες/τα αξιοθέατα/την ιστορία της χώρας ή της πόλης προς επίσκεψη, και πραγματικά η λίστα όλων αυτών των εξειδικευμένων μικρο-εφαρμογών συνεχίζεται και συνεχίζεται.

Στην ίδια ακριβώς διαπίστωση θα φτάσει κανείς αν επιχειρήσει μια περιήγηση στα μεγάλα «καταστήματα μικρο-εφαρμογών»: αναρίθμητα apps (δωρεάν ή επί πληρωμή) για όλες τις παραπάνω χρήσεις. Αρκετά από αυτά τα apps είναι προσανατολισμένα στην τουριστική ξενάγηση μέσω χρήσης κάποιας υπηρεσίας χαρτών.

Με τα προηγούμενα εννοείται η δυνατότητα του χρήστη της μικρο-εφαρμογής να περιηγείται - περπατώντας και όχι βρισκόμενος σε κάποιο όχημα - ανάμεσα στα εκθέματα ενός μουσείου, ενός αρχαιολογικού χώρου, ενός τοπικού φεστιβάλ, ενός λατρευτικού τόπου συγκέντρωσης, μιας έκθεσης προϊόντων ή υπηρεσιών ή εν πάση περιπτώσει, σε αξιοθέατα γύρω από μια γειτονιά ή περιοχή της πόλης. Η μικρο-εφαρμογή του παρέχει ενημέρωση και πληροφόρηση σε σχέση με αυτά που υπάρχουν γύρω του, προτείνει διαδρομές προς εξερεύνηση, παρουσιάζει συναφείς με τα εκθέματα πληροφορίες, εντοπίζει το σημείο που αυτός βρίσκεται στον χώρο, του επιτρέπει να μοιράζεται στα κοινωνικά δίκτυα (social media) αυτά που εκείνη τη στιγμή αυτός βιώνει, κ.α. πολλές δυνατότητες.

Πραγματικά υπάρχουν αρκετές ιδιαίτερα αξιόλογες μικρο-εφαρμογές που προσφέρουν τουριστική ξενάγηση, κυρίως apps για τα κορυφαία μουσεία του πλανήτη, τις πρωτεύουσες και τις δημοφιλείς

τουριστικές πόλεις πολλών χωρών ή τις διάσημες τουριστικές ατραξιόν ανά τον κόσμο. Οι περισσότερες από αυτές χρησιμοποιούν με εντυπωσιακό τρόπο τη σύγχρονη τεχνολογία όπως πολυμέσα, τις κάμερες των smartphones, τους αισθητήρες τους, τους ασύρματους δέκτες τους π.χ Bluetooth, WiFi, το σύστημα γεωγραφικού εντοπισμού (GPS), την επαυξημένη πραγματικότητα (augmented reality), την τρισδιάστατη απεικόνιση κλπ.

Πόσα όμως από αυτά τα apps παρέχουν μια πραγματικά απλή, λιτή και εύχρηστη περιήγηση στον επισκέπτη ενός αρχαιολογικού/μουσειακού/εκθεσιακού τόπου που βρίσκεται σε υπαίθριο ή εν γένει εξωτερικό χώρο; Πραγματικά ελάχιστα, και ιδιαίτερα όσον αφορά τον πλούσιο σε αντίστοιχα σημεία ελλαδικό χώρο, θα έλεγε κανείς μηδαμινά!

Αρκεί μια απλή διαδικτυακή αναζήτηση στα βασικά τουριστικά σημεία ενδιαφέροντος ανά την Ελλάδα (για παράδειγμα Νέο Μουσείο Ακρόπολης, Αρχαιολογικό Μουσείο Αθηνών, Αρχαιολ.Χώρος Ακρόπολης, Αρχαιολ.Χώρος Ολυμπίας, Αρχαιολ.Χώρος Δελφών, Μυκηνών, Κνωσού κλπ) για να διαπιστώσει κανείς την παντελή έλλειψη τέτοιων προσφερόμενων λύσεων!

Ακόμα και σε εκτεταμένη βιβλιογραφική αναζήτηση [2] στον ευρύτερα σχετιζόμενο επιστημονικό χώρο, προκύπτει ένδεια επιστημονικών δημοσιεύσεων και αναφορών! (ενδεικτικά: [3], [4], [5]).

Με όλα τα προαναφερθέντα καταδεικνύεται συνεπώς, το κενό και η έλλειψη στην προσφορά μικρο-εφαρμογών για αυτή ακριβώς την ομάδα χρηστών και ακριβώς αυτού του τύπου χρήσης. Δηλαδή απλές στη χρήση και στη διεπαφή μικρο-εφαρμογές, για τον επισκέπτη υπαίθριων αρχαιολογικών ή μουσειακών χώρων στην Ελληνική επικράτεια. Εξ ου λοιπόν και η ανάγκη για τον σχεδιασμό και την ανάπτυξη τέτοιων μικρο-εφαρμογών.

Σκοπός συνεπώς, της παρούσας εργασίας, είναι ακριβώς αυτός: το να σχεδιαστεί και να υλοποιηθεί μια μικρο-εφαρμογή για έξυπνα κινητά τηλέφωνα “smartphones” (και κατ’επέκταση ίσως, Η/Υ τύπου «πλακιδίου» δηλ. tablets) που να παίξει τον ρόλο ενός ψηφιακού ξεναγού για εκθέματα και σημεία ενδιαφέροντος που βρίσκονται σε εξωτερικούς αρχαιολογικούς ή/και μουσειακούς χώρους της Ελλάδας (ενδεχομένως και εκτός της ελληνικής επικράτειας).

Ως παράδειγμα εφαρμογής (case study) για την παρούσα εργασία, θα χρησιμοποιηθεί το Βιομηχανικό Μουσείο Φωταερίου που βρίσκεται

στην Τεχνόπολι, στο Γκάζι της Αθήνας (βλ. [www.technopolis-athens.com/web/guest/museum](http://www.technopolis-athens.com/web/guest/museum)), όπως θα αναλυθεί παρακάτω.

## 1.2 Παράδειγμα Εφαρμογής / Case Study

Το Βιομηχανικό Μουσείο Φωταερίου, που αποτελεί το πρώτο βιομηχανικό μουσείο της Αθήνας, έχει ως στόχο την ανάδειξη και προστασία του παλαιού εργοστασίου φωταερίου της Αθήνας, ενός μνημείου βιομηχανικής κληρονομιάς που παρείχε ενέργεια και φωτισμό στην πρωτεύουσα για περίπου 130 χρόνια [6].



Εικόνα 1: Το λογότυπο του Μουσείου

Η επίσκεψη στο Μουσείο περιλαμβάνει ένα μουσειακό περίπατο με 13 στάσεις σε συγκεκριμένα σημεία του εργοστασίου. Μέσα από αυτή τη βόλτα, ο επισκέπτης έχει τη δυνατότητα να ακολουθήσει τη γραμμή παραγωγής του φωταερίου, να θαυμάσει το μηχανολογικό εξοπλισμό και τα παλαιά μηχανήματα, να ανακαλύψει το ρόλο των επιχειρηματιών και των εργαζομένων του εργοστασίου, να γνωρίσει τις συνθήκες εργασίας και να εξοικειωθεί με έννοιες όπως η βιομηχανική κληρονομιά και αρχαιολογία [6].



Εικόνα 2: Μιά κάτοψη των εξωτερικών χώρων του Μουσείου



Το συγκεκριμένο μουσείο συνδυάζει εκθέματα τόσο σε εσωτερικό όσο και σε εξωτερικό επίπεδο, βρίσκεται εντός περιορισμένου σχετικά χώρου (δεν αποτελεί μεγάλη έκταση δηλαδή) καθιστώντας έτσι την – εξωτερική τουλάχιστον – περιήγηση στους χώρους του, μια εύκολη σχετικά υπόθεση για τον επισκέπτη του, πόσω μάλλον όταν η τοποθεσία του ίδιου του μουσείου βρίσκεται στο κέντρο της Αθήνας. Αποτελεί συνεπώς, έναν κατάλληλο χώρο, μια ενδεδειγμένη τοποθεσία, για τον σχεδιασμό και την ανάπτυξη μιας μικρο-εφαρμογής που θα ξεναγεί με λιτό και εύχρηστο τρόπο τον χρήστη της στους εξωτερικούς χώρους του.

## Κεφάλαιο 2 / Chapter 2

### 2.1 Σχεδιασμός Απαιτήσεων / Requirements Design

Στο παρόν σημείο της εργασίας πρέπει να οριοθετηθούν και να ταυτοποιηθούν ποιές θα είναι οι ελάχιστες απαιτήσεις που θα πρέπει να καλύπτει, και ποιοι οι αναμενόμενοι στόχοι που θα πρέπει να επιτύχει, η προς ανάπτυξη μικρο-εφαρμογή.

Η μικρο-εφαρμογή απευθύνεται στον επισκέπτη του Βιομηχανικού Μουσείου Φωταερίου που θέλει να ξεναγηθεί στους εξωτερικούς χώρους του μουσείου, κάνοντας χρήση του έξυπνου κινητού του τηλεφώνου. Ο επισκέπτης μπορεί να είναι οποιασδήποτε ηλικιακής κατηγορίας και μορφωτικού ή τεχνολογικού επιπέδου, με μόνη προϋπόθεση την κατοχή και βασική - τουλάχιστον - ικανότητα χρήσης του “smartphone” του (ή έστω tablet του).

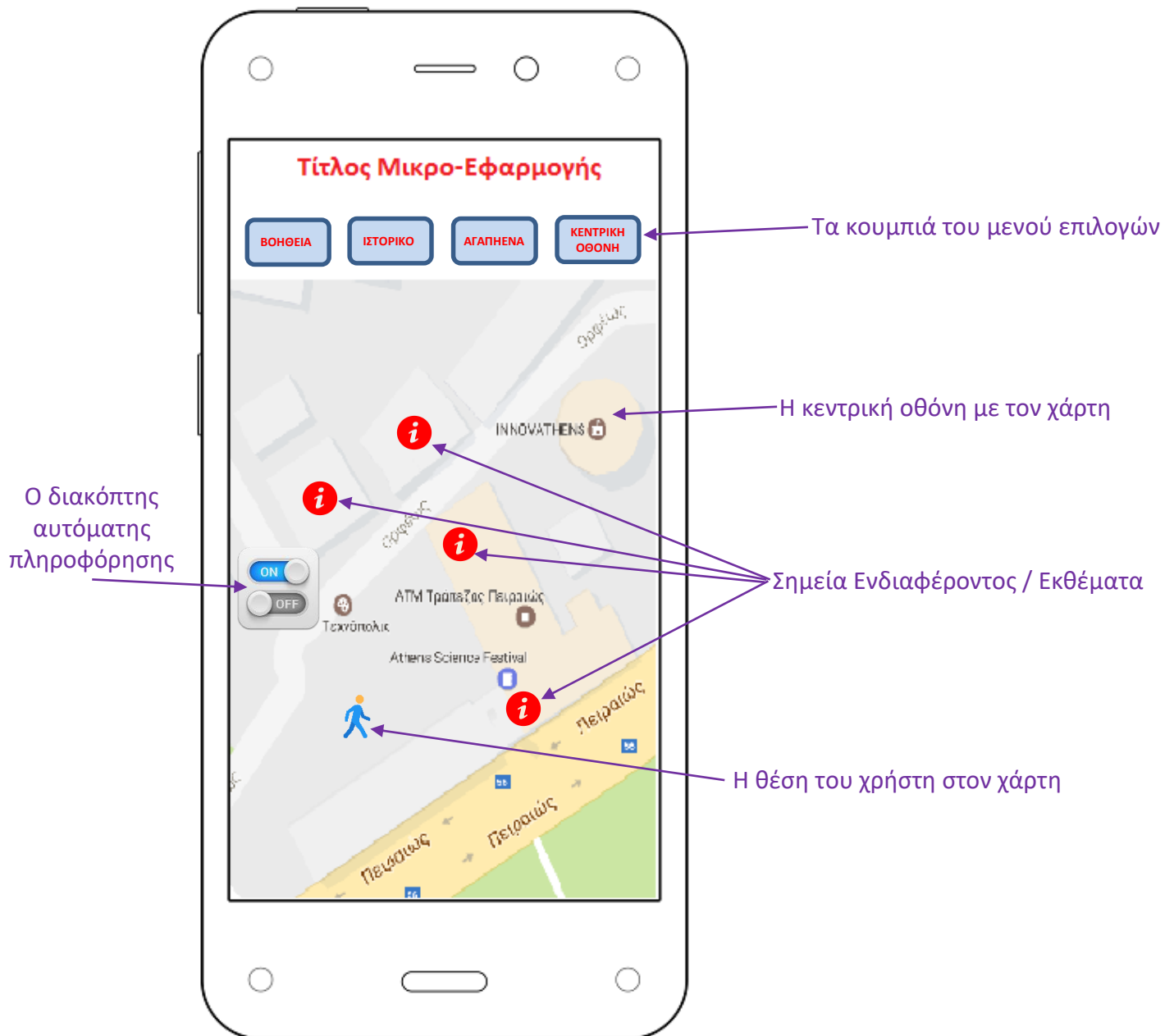
Συνεπώς, η μικρο-εφαρμογή θα πρέπει να έχει μια απλή και κατανοητή εμφάνιση, τόσο σε επίπεδο χρωμάτων όσο και σε επίπεδο στοιχείων πάνω στην οθόνη, μία εύχρηστη/εύληπτη διεπαφή χρήστη (user interface) και να παρέχει μια άμεσα αντιληπτή εμπειρία χρήσης.

Οι βασικές δυνατότητες και τα κύρια χαρακτηριστικά ως προς τις υπηρεσίες που θα προσφέρει η μικρο-εφαρμογή στον χρήστη, μπορούν να συνοψιστούν στα εξής:

- Η μικρο-εφαρμογή θα απεικονίζει χάρτη (κάτοψη) του Μουσείου με τα εκθέματα (ή σημεία ενδιαφέροντος) που αυτό περιέχει.
- Τα σημεία ενδιαφέροντος θα πρέπει να απεικονίζονται με έντονο και ευδιάκριτο τρόπο.
- Ο χρήστης θα μπορεί να επιλέξει (με χειρονομία τύπου “tap”, δηλ. αγγίγματος/χτυπήματος στην οθόνη) να δει πληροφορίες για σημείο ή σημεία που τον ενδιαφέρουν.
- Ο χρήστης θα μπορεί να εντοπίζει τον εαυτό του (τόσο σε στάση, όσο και εν κινήσει) σε σχέση με τα σημεία ενδιαφέροντος.
- Κατά την προσέγγιση του χρήστη σε κάποιο σημείο ενδιαφέροντος, θα υπάρχει η δυνατότητα της αυτόματης προβολής και θέασης πληροφοριών για το συγκεκριμένο σημείο.
- Η προηγούμενη δυνατότητα θα μπορεί να ενεργοποιείται και να απενεργοποιείται καθολικά με έναν «διακόπτη».

- Η προβολή των πληροφοριών για κάποιο σημείο ενδιαφέροντος θα γίνεται μέσω «επιπλέοντος» (floating) παραθύρου πάνω στον χάρτη του app.
- Οι πληροφορίες προς απεικόνιση θα μπορούν να είναι «πλούσια» μορφοποιημένες (με χρώματα, έντονους ή πλάγιους χαρακτήρες ενδεχομένως διαφορετικού μεγέθους ή γραμματοσειράς, ακόμα και με εισαγωγή πολυμέσων, δηλαδή εικόνων, βίντεο, ήχων) και όχι «απλού» κειμένου. Επίσης θα παρέχεται η δυνατότητα παραπομπής σε ιστοσελίδες για περισσότερο περιεχόμενο.
- Για τις επισκέψεις του χρήστη σε διάφορα σημεία ενδιαφέροντος (δηλαδή για τα σημεία που ο χρήστης προσέγγισε τοπικά) θα τηρείται λίστα ιστορικού επισκέψεων σε ξεχωριστή οθόνη της μικρο-εφαρμογής, την οποία και θα μπορεί να θεωρεί και διαχειρίζεται.
- Τα σημεία ενδιαφέροντος που ο χρήστης εκτιμά ως σημαντικά ή αξιωμακρόνευτα, θα μπορεί (με χειρονομία τύπου διπλού “tap” και σχετική ερώτηση που θα του τίθεται) να τα αποθηκεύει σε λίστα αγαπημένων σημείων, που θα τηρείται σε ξεχωριστή οθόνη της μικρο-εφαρμογής, την οποία και θα μπορεί να θεωρεί και διαχειρίζεται.
- Θα παρέχεται οθόνη βοήθειας, με όλες τις απαραίτητες οδηγίες χρήσης της μικρο-εφαρμογής και τυχόν διευκρινήσεις.
- Για όλες τις παραπάνω ξεχωριστές οθόνες (Κεντρική οθόνη ή οθόνη του χάρτη - οθόνη Βοήθειας - οθόνη Ιστορικού - οθόνη Αγαπημένων) θα υπάρχει μενού επιλογής με κουμπιά, τοποθετημένο μόνιμα στο άνω μέρος της επιφάνειας της μικρο-εφαρμογής, κάτω από τον τίτλο αυτής.

Για να γίνουν πιο άμεσα αντιληπτά τα περισσότερα από τα προαναφερθέντα, στο σημείο αυτό θα παρατεθούν κάποια εικονικά (dummy) πρωτότυπα (mockups) των οθονών της μικρο-εφαρμογής, όπως σχεδιάστηκαν κατά τη διάρκεια της διαδικασίας της ανάλυσης απαιτήσεων:



Εικόνα 3: Η κεντρική οθόνη της μικρο-εφαρμογής

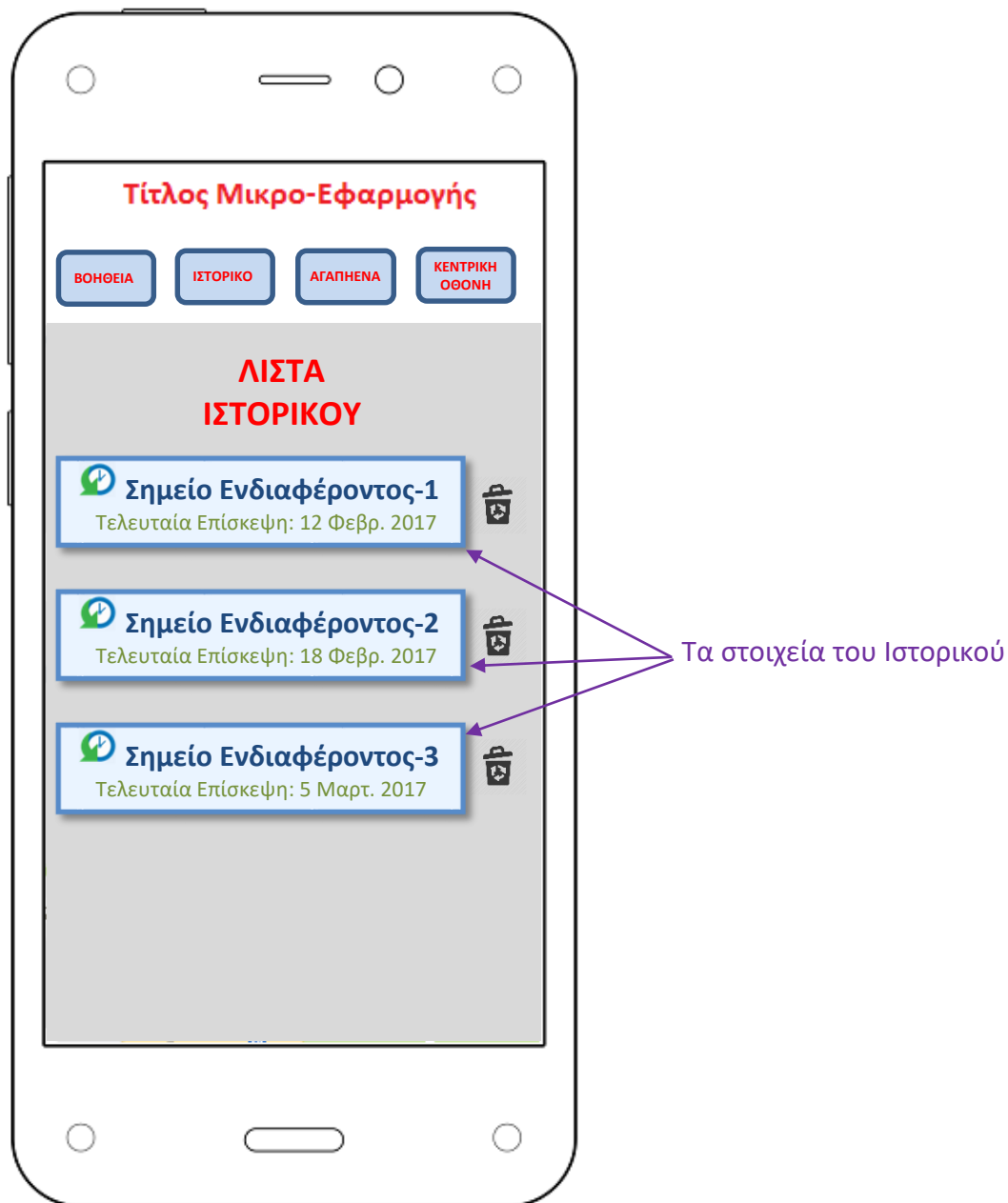
Στο συγκεκριμένο mockup παρουσιάζεται η κεντρική οθόνη και το βασικό user interface, με τον τίτλο του app και το μενού επιλογών που διαθέτει, υπό τη μορφή κουμπιών. Στην ευρύτερη περιοχή της οθόνης, απεικονίζεται ο χάρτης του χώρου στον οποίο βρίσκεται ο χρήστης, με τον ίδιο να απεικονίζεται με κατάλληλο εικονίδιο. Επίσης εμφανίζονται με κατάλληλο εικονίδιο και όλα τα σημεία ενδιαφέροντος που βρίσκονται στον χώρο προς περιήγηση. Τέλος, υπάρχει στο αριστερό μέσο της οθόνης, διακόπτης για την ενεργοποίηση / απενεργοποίηση της αυτόματης πληροφόρησης.



Το παράθυρο προβολής πληροφοριών για το σημείο ενδιαφέροντος που επισκέπτεται ο χρήστης

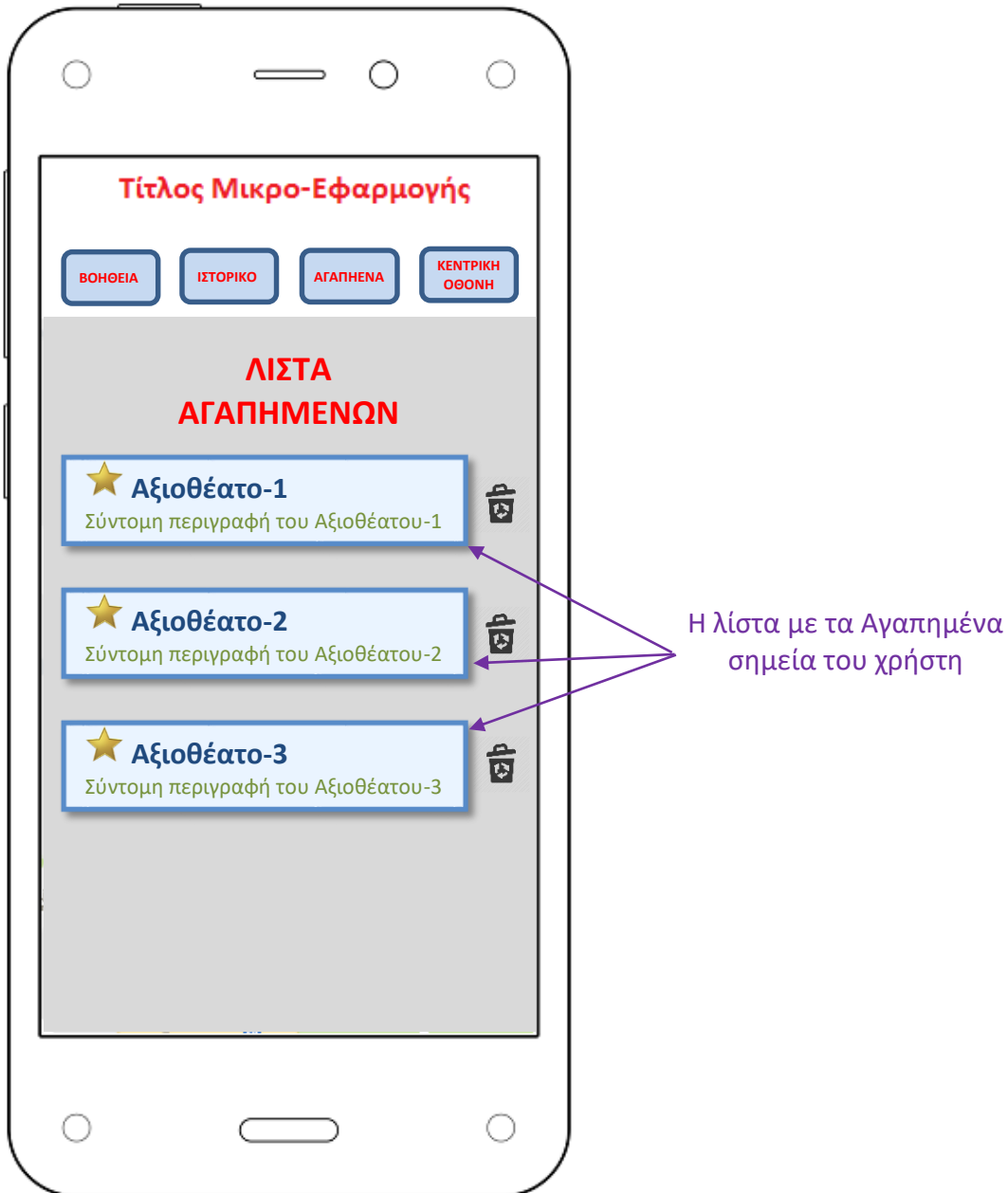
Εικόνα 4: Η κεντρική οθόνη και το παράθυρο πληροφόρησης

Σε αυτό το mockup παρουσιάζεται η κεντρική οθόνη και πάλι, όπου όμως τώρα φαίνεται η προσέγγιση του χρήστη σε κάποιο σημείο ενδιαφέροντος. Όταν ο χρήστης πλησιάσει το σημείο, θα εμφανιστεί αυτόματα (εφόσον είναι ενεργοποιημένος ο αντίστοιχος διακόπτης) πάνω στον χάρτη, παράθυρο που προβάλλει όλες τις απαραίτητες πληροφορίες για την ενημέρωση του επισκέπτη. Υπόψη πως αυτό το παράθυρο θα μπορεί να εμφανιστεί οποτεδήποτε το θελήσει ο χρήστης της μικρο-εφαρμογής, αρκεί να κάνει μονό “tap” πάνω στο εικονίδιο κάποιου σημείου ενδιαφέροντος.



Εικόνα 5: Η οθόνη με τη λίστα του Ιστορικού

Το mockup αυτό παρουσιάζει την οθόνη του Ιστορικού, όπου εμφανίζονται υπό μορφή λίστας τα σημεία ενδιαφέροντος που ο χρήστης της μικρο-εφαρμογής έχει επισκεφτεί κάποια στιγμή κατά το παρελθόν, μαζί με την ακριβή ημερομηνία επίσκεψης. Η λίστα του Ιστορικού θα πρέπει να δημιουργείται και να ανανεώνεται αυτόματα με την επίσκεψη του χρήστη σε κάποιο από αυτά τα σημεία. Μπορεί να παρατηρήσει κανείς το εικονίδιο του «Κάδου Ανακύκλωσης» στα δεξιά κάθε στοιχείου της λίστας, με το οποίο ο χρήστης έχει τη δυνατότητα να διαγράψει το στοιχείο που επιθυμεί.



Εικόνα 6: Η οθόνη με τη λίστα των Αγαπημένων

Το τελευταίο mockup απεικονίζει την οθόνη των Αγαπημένων, όπου εμφανίζονται υπό μορφή λίστας, σημεία ενδιαφέροντος ιδιαίτερης σπουδαιότητας, που ο χρήστης της μικρο-εφαρμογής έχει επιλέξει να αποθηκεύσει. Μαζί με τον τίτλο του αξιοθέατου, εμφανίζεται και μια σύντομη περιγραφή αυτού. Και σε αυτή τη λίστα υπάρχει η δυνατότητα διαγραφής στοιχείων.

## 2.2 Προδιαγραφές Μικρο-Εφαρμογής / App Specs

Μετά την ανάλυση απαιτήσεων της εργασίας, είναι απαραίτητο να προσδιοριστούν με όσο το δυνατόν μεγαλύτερη ακρίβεια, οι ελάχιστες προδιαγραφές τις οποίες θα πρέπει να ικανοποιεί η μικρο-εφαρμογή προκειμένου να ανταπεξέλθει στο έργο της.

Οι προδιαγραφές που εν τέλει θα τεθούν, πρακτικά λίγο έχουν να κάνουν με τη λειτουργικότητα ή τις διαδικασίες που θα διεκπεραιώνονται από τη μικρο-εφαρμογή, καθώς όπως έχει ήδη αναφερθεί, πρόκειται για - εξ ορισμού - απλό και εύκολο στη χρήση app.

Οι προδιαγραφές κυρίως θα τεθούν σε τεχνικό επίπεδο, καθώς πρόκειται ίσως για το πιο ακανθώδες ζήτημα σχετικά με την υλοποίηση της μικρο-εφαρμογής.

Ας δούμε αναλυτικά όμως, όλα αυτά τα θέματα:

Όσον αφορά τα σενάρια και τις διαδικασίες χρήσης του app, τα πράγματα είναι απλά. Ο χρήστης-επισκέπτης ενός αρχαιολογικού/μουσειακού χώρου ευρισκόμενου σε εξωτερική τοποθεσία, ενεργοποιεί το app αφού πρώτα έχει ενεργοποιήσει τη σύνδεση δεδομένων ή WiFi σύνδεση της συσκευής (έτσι ώστε να έχει η συσκευή πρόσβαση στο Internet, για να μπορέσει το app να «κατεβάσει» τον χάρτη) και τον ασύρματο δέκτη GPS (προκειμένου να εντοπίζεται η γεωγραφική θέση του χρήστη και η κίνησή του στον χώρο). Σε περίπτωση που κατά την εκκίνησή του το app διαπιστώσει πως δεν υπάρχει GPS δυνατότητα και (κυρίως) σύνδεση στο Internet από τη συσκευή, τότε θα παρουσιάζει ενημερωτικό μήνυμα στον χρήστη που θα τον προτρέπει να ενεργοποιήσει τις σχετικές δυνατότητες της συσκευής και κατόπιν θα τερματίζει το app.

Η μικρο-εφαρμογή εντοπίζει σε σύντομο χρονικό διάστημα (τυπικά εντός ολίγων δευτερολέπτων) τη γεωγραφική θέση που βρίσκεται ο χρήστης-επισκέπτης και την εμφανίζει με συγκεκριμένο ευδιάκριτο εικονίδιο πάνω στον χάρτη της τοποθεσίας. Στη συνέχεια παρακολουθεί την κίνηση του χρήστη (απαραίτητη προϋπόθεση η κίνηση αυτή να προκύπτει μέσω φυσιολογικού βαδίσματος ή περιπάτου) και αν διαπιστωθεί η προσέγγισή του σε κάποιο από τα σημεία ενδιαφέροντος της τοποθεσίας τότε θα αναδυθεί με αυτόματο τρόπο παράθυρο πληροφόρησης προς τον χρήστη της μικρο-εφαρμογής, σχετικό με το σημείο ενδιαφέροντος που αυτός έχει τώρα προσεγγίσει. Στο συγκεκριμένο παράθυρο περιγράφεται με συνοπτικό



τρόπο το σημείο ενδιαφέροντος (ιστορικά, πολιτιστικά, στατιστικά στοιχεία που το αφορούν κλπ). Το κείμενο του παραθύρου – εκτός του ότι πρέπει να είναι μορφοποιημένο κατάλληλα για να είναι ευανάγνωστο – μπορεί να συνοδεύεται από πολυμεσικά στοιχεία όπως εικόνες, φωτογραφίες, ήχους, βίντεο, κινούμενα γραφικά (animation). Τέλος, και για τον χρήστη που θα το επιθυμήσει, στο παράθυρο θα υπάρχει παραπομπή μέσω διαδικτυακού συνδέσμου, σε ιστοσελίδα (ή ιστοσελίδες) που θα παρέχει αναλυτικότερες πληροφορίες σχετικές με το συγκεκριμένο σημείο ενδιαφέροντος.

Με την απομάκρυνση του χρήστη από το τρέχον σημείο ενδιαφέροντος, θα διακόπτεται η αυτόματη πληροφόρησή του, θα αποσύρεται δηλαδή, το παράθυρο πληροφόρησης και η διαδικασία θα επαναλαμβάνεται με τον ίδιο ακριβώς τρόπο στο επόμενο σημείο ενδιαφέροντος που θα προσεγγίσει ο χρήστης κ.ο.κ.

Σε αυτό το σημείο πρέπει να τονιστεί πως υπάρχει πάντα η δυνατότητα για τον χρήστη της μικρο-εφαρμογής, να κάνει επισκόπηση του παραθύρου πληροφόρησης για οποιοδήποτε σημείο ενδιαφέροντος που θα επιθυμήσει, αρκεί να κάνει μονό “tap” πάνω σε αυτό, ακόμα και εάν δεν βρίσκεται κοντά του.

Τα παράθυρα πληροφόρησης έχουν τη δυνατότητα «χειροκίνητης» απόκρυψης από τον χρήστη, αν αυτός επιλέξει το X στην πάνω δεξιά γωνία τους. Βέβαια, αν ο διακόπτης της αυτόματης πληροφόρησης είναι ενεργοποιημένος και ο χρήστης ευρίσκεται πλησίον του σημείου ενδιαφέροντος, τότε το παράθυρο πληροφόρησης θα εμφανιστεί ξανά άμεσα. Η λύση εδώ είναι να απενεργοποιήσει τον διακόπτη της αυτόματης πληροφόρησης.

Κάθε σημείο που ο χρήστης προσεγγίζει (και ανεξάρτητα αν θα εμφανιστεί ή όχι παράθυρο αυτόματης πληροφόρησης) προστίθεται αυτόματα – μαζί με την ημερομηνία προσέγγισης – σε μια λίστα Ιστορικού που αποθηκεύεται εντός της μικρο-εφαρμογής.

Ο χρήστης του app έχει ανά πάσα στιγμή πρόσβαση – μέσω αντίστοιχου κουμπιού στο μενού επιλογών του app – στην αποθηκευμένη λίστα του Ιστορικού, στην οποία μπορεί να επιλέξει να διαγράψει μεμονωμένα ή όλα τα στοιχεία της λίστας.

Ανά πάσα στιγμή έχει επίσης πρόσβαση στη λίστα των Αγαπημένων. Αυτή η λίστα δημιουργείται (και αποθηκεύεται επίσης, εντός της μικρο-εφαρμογής) όταν ο χρήστης κάνει διπλό “tap” πάνω σε οποιοδήποτε σημείο ενδιαφέροντος που θα επιθυμήσει. Τότε θα προκύψει παράθυρο διαλόγου, στο οποίο θα ερωτάται ο χρήστης εάν

επιθυμεί να αποθηκεύσει το σημείο ως Αγαπημένο. Στην καταφατική απάντηση, το σημείο θα προστίθεται στη λίστα των Αγαπημένων. Σε αυτή τη λίστα λοιπόν, ο χρήστης θα μπορεί να δει τα σημεία ενδιαφέροντος που τον εντυπωσίασαν ιδιαίτερα ή θεωρεί σημαντικά, μαζί με μια λιτή περιγραφή του σημείου και θα έχει τη δυνατότητα και εδώ, να διαγράψει μεμονωμένα ή όλα τα στοιχεία της λίστας.

Τέλος, ο χρήστης του app θα μπορεί σε όποια οθόνη και να βρίσκεται, να μεταβεί - πάλι, μέσω αντιστοιχων κουμπιών στο μενού επιλογών - στην οθόνη της Βοήθειας και βέβαια να επιστρέψει στην κεντρική, στη βασική οθόνη της μικρο-εφαρμογής (με τον χάρτη, τα σημεία ενδιαφέροντος κλπ) για να συνεχίσει την ξενάγησή του και την περιήγησή του στον αρχαιολογικό/μουσειακό χώρο.

---

---

Όσον αφορά την περιγραφή των προδιαγραφών της μικρο-εφαρμογής σε τεχνικό επίπεδο, αναπόφευκτα τα πράγματα πρέπει να γίνουν πιο συγκεκριμένα:

Το συγκεκριμένο app θα υλοποιηθεί για όλες τις βασικές πλατφόρμες έξυπνων κινητών τηλεφώνων. Δηλαδή για την πλατφόρμα των iPhones (με λειτουργικό σύστημα iOS της εταιρείας Apple), για τα Android smartphones (με λειτουργικό σύστημα Android της εταιρείας Google και της σύμπραξης O.H.A.) και για τα τηλέφωνα με λειτουργικό σύστημα Windows Phone της εταιρείας Microsoft. Με λίγα λόγια, η μικρο-εφαρμογή θα πρέπει να «τρέχει» στα Android κινητά, στα κινητά Windows και στα iPhones.

Οι συγκεκριμένες τρεις πλατφόρμες λειτουργικών συστημάτων για έξυπνα κινητά, επιλέχθηκαν καθώς αποτελούν αθροιστικά σχεδόν το 99% της σχετικής αγοράς [7]. Συνεπώς, συζήτηση για οποιαδήποτε άλλη πλατφόρμα τίθεται αυτομάτως εκτός ρεαλιστικής βάσης.

Όσον αφορά τις συσκευές στις οποίες θα πρέπει να «τρέχει» το app, αυτές θα καλύπτουν κατ' ελάχιστο τις εξής γενικές προδιαγραφές όσον αφορά το υλικό (hardware) τους: συσκευές πολυπύρηνες, με τουλάχιστον 1GB κεντρικής μνήμης RAM, δυνατότητα δικτύωσης στο Internet μέσω WiFi ή γρήγορης σύνδεσης δεδομένων (data connection) μέσω τεχνολογίας 3G/4G (για το «κατέβασμα» του χάρτη του app), ασύρματος δέκτης GPS (για την εύρεση και εντοπισμό γεωγραφικού στίγματος) και τέλος, μια μεγάλη - τουλάχιστον 4.5"-5" ιντσών - οθόνη αφής για την ικανοποιητική προβολή και θέαση των πληροφοριών που παρέχει η μικρο-εφαρμογή.

Όπως γίνεται φανερό λοιπόν, αναφερόμαστε σε συσκευές κατασκευής - προσεγγιστικά - του έτους 2014/2015 και νεότερες. Αυτό πρακτικά συμπαράσχει και τις εκδόσεις των προαναφερθέντων λειτουργικών συστημάτων. Σε γενικές γραμμές, η μικρο-εφαρμογή θα «τρέχει» σε περιβάλλον Android 4.4 KitKat και ανώτερο, iOS έκδοση 9 και 10, Windows Phone έκδοση 10 (και με επιφύλαξη στην έκδοση 8.1)

#### Σημείωση:

Το app θα υλοποιηθεί ως υβριδική κινητή εφαρμογή, και όχι ως εγγενής (native) εφαρμογή για κάθε μια από τις τρεις πλατφόρμες, επιλογή που θα εξηγηθεί επαρκώς στο αμέσως επόμενο κεφάλαιο. Ως υβριδική εφαρμογή θα στηριχθεί για την ανάπτυξη και την υλοποίησή της στις υπάρχουσες τεχνολογίες του Διαδικτύου, και συγκεκριμένα στο τρίπτυχο HTML5/CSS3/Javascript. Αυτές οι τεχνολογίες υποστηρίζουν εύρεση και εντοπισμό γεωγραφικής θέσης μέσω του Geolocation API.

Το συγκεκριμένο A.P.I. (Application Programming Interface = Διεπαφή Προγραμματισμού Εφαρμογής) επιτρέπει τον προσδιορισμό στίγματος μέσω πολλών διαφορετικών τεχνικών [8]:

- είτε μέσω του ασύρματου δέκτη GPS
- είτε μέσω τριγωνισμού από τις κυψέλες κινητής τηλεφωνίας
- είτε μέσω του ασύρματου πομποδέκτη WiFi
- είτε μέσω της IP διεύθυνσης της συσκευής
- είτε ενδεχομένως με κάποια άλλη τεχνική π.χ. μέσω του ασύρματου πομποδέκτη Bluetooth

Η τεχνική που παρέχει τον ακριβέστερο εντοπισμό στίγματος (και αυτό είναι το ζητούμενο για το προς ανάπτυξη app) είναι η πρώτη, γι' αυτόν ακριβώς τον λόγο μπαίνει ο δέκτης GPS στις ελάχιστες προδιαγραφές υλικού για την υποστήριξη της μικρο-εφαρμογής.

Σύμφωνα και με τα παραπάνω όμως, δεν μπορεί να αποκλειστεί η δυνατότητα μιας συσκευής να παρέχει ικανοποιητικό εντοπισμό γεωγραφικού στίγματος με κάποια από τις υπόλοιπες τεχνικές.

## Κεφάλαιο 3 / Chapter 3

### 3.1 Τεχνολογίες ανάπτυξης / Development Technologies

Περνώντας στη φάση της ανάπτυξης της μικρο-εφαρμογής, θα πρέπει να απαντηθούν κάποια σημαντικά ερωτήματα για την ίδια τη διαδικασία της ανάπτυξης:

- α) Ποιος είναι ο συντομότερος και απλούστερος δρόμος για το development (ανάπτυξη) του app;
- β) Ποιος ο περισσότερο fail-safe (ασφαλής) τρόπος να αναπτυχθεί το app;
- γ) Ποιες είναι οι τεχνολογίες πάνω στις οποίες θα στηριχθεί η όλη προσπάθεια της ανάπτυξης της μικρο-εφαρμογής;

Και στα τρία ερωτήματα η απάντηση είναι η ίδια: οι τεχνολογίες του διαδικτύου είναι η λύση, οι μοντέρνες διαδικτυακές (web) τεχνολογίες της HTML5, των CSS3 και της Javascript θα αποτελέσουν τις βάσεις πάνω στις οποίες θα χτιστεί το app.

Γιατί όμως οι συγκεκριμένες τεχνολογίες; Πριν καταλήξουμε στη συλλογιστική της απάντησης, θα πρέπει πρώτα να εμβάθινουμε λίγο στον κόσμο των μικρο-εφαρμογών για έξυπνα κινητά τηλέφωνα:

#### 3.1.1 Native vs Web vs Hybrid Mobile Apps

Τα apps διακρίνονται, ανάλογα με τη μέθοδο υλοποίησής τους, σε τρεις κατηγορίες [9]:

1. **Native (εγγενή) Apps:** στην πρώτη κατηγορία, έχουμε τις μικρο-εφαρμογές που αναπτύσσονται με τις τεχνικές, τις μεθόδους, τα εργαλεία και κυρίως τη γλώσσα προγραμματισμού που «επιβάλλει» η κάθε πλατφόρμα, π.χ. Java για Android native apps, Objective-C για Apple native apps, C# ή VB ή C++ που περιέχονται στο Visual Studio της Microsoft, για Windows Phone native apps.
2. **Web (διαδικτυακά) Apps:** εδώ αναφερόμαστε πρακτικά σε responsive web sites (προσαρμοζόμενους ιστοτόπους) που έχουν αναπτυχθεί με τη χρήση HTML+CSS+Javascript και προορίζονται να «αναπαραχθούν» από τον φυλλομετρητή (browser) του smartphone.

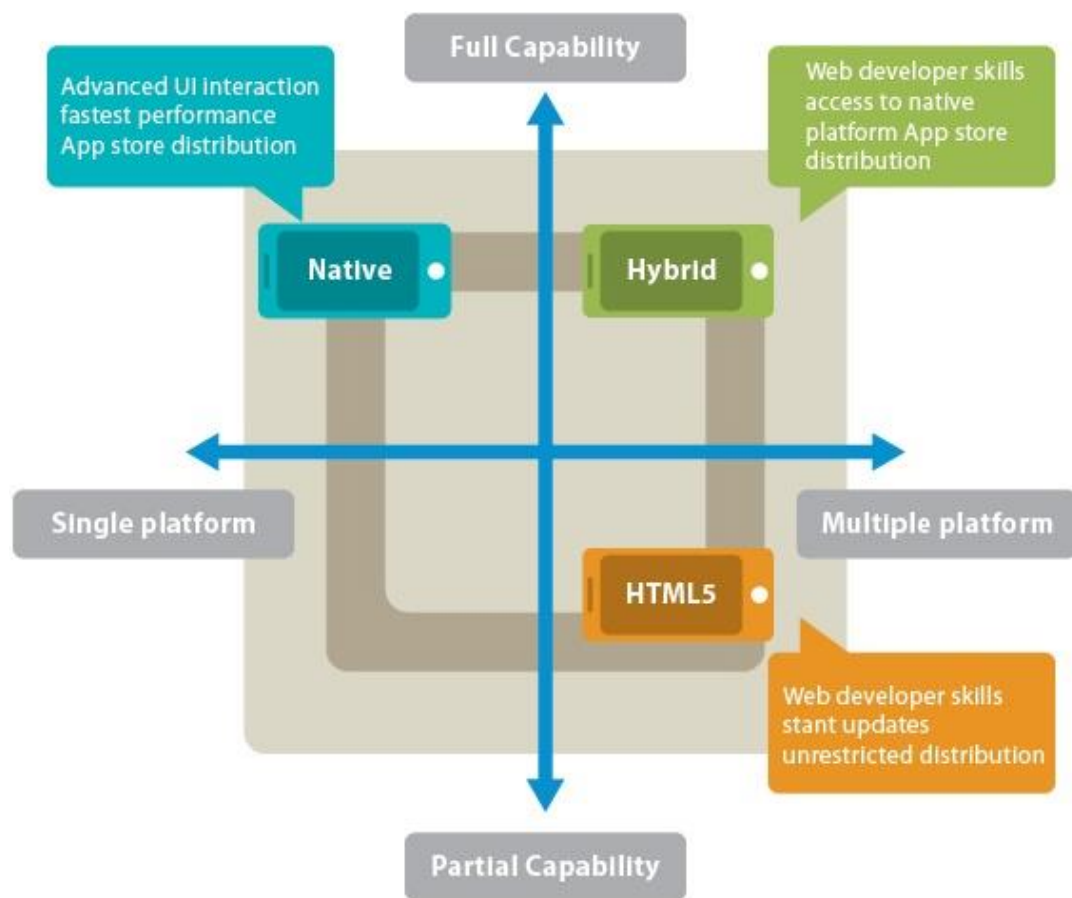
3. **Hybrid (υβριδικά) Apps:** η τρίτη κατηγορία - που ουσιαστικά είναι μια μίξη των δύο πρώτων - αναφέρεται στις μικρο-εφαρμογές που είναι υλοποιημένες και αυτές με HTML+CSS+Javascript, όμως χρησιμοποιούν και κομμάτια κώδικα (μικρότερα ή μεγαλύτερα, δεν έχει σημασία) γραμμένα στη γλώσσα προγραμματισμού που απαιτεί η κάθε πλατφόρμα για την οποία προορίζεται το app, π.χ. Objective-C ή Java.

Για κάθε έναν από τους παραπάνω τρόπους υλοποίησης μικρο-εφαρμογών, υπάρχουν υπέρμαχοι και πολέμιοι, και βέβαια εννοείται, συγκεκριμένα πλεονεκτήματα και μειονεκτήματα:

- Για τα μεν native apps, το μεγάλο τους πλεονέκτημα είναι η «εγγύτητα» με το UI της πλατφόρμας και η απόλυτη εκμετάλλευση των δυνατοτήτων της συσκευής σε συνδυασμό με τον άμεσο τρόπο προγραμματισμού. Από την άλλη όμως, έχουν σαν μεγάλο μειονέκτημα το γεγονός πως την ίδια ακριβώς μικρο-εφαρμογή πρέπει να την αναπτύξει κανείς με διαφορετική γλώσσα προγραμματισμού και διαφορετικά εργαλεία, για διαφορετική πλατφόρμα, εκτοξεύοντας έτσι τον χρόνο και το κόστος υλοποίησης σε υψηλά επίπεδα.
- Τα web apps έχουν ως πλεονέκτημά τους την εύκολη και γρήγορη ανάπτυξή τους (αφού βασίζονται σε κοινές και διαδεδομένες τεχνολογίες) και κυρίως το γεγονός πως για πολλαπλές πλατφόρμες χρειάζεται μια μόνο υλοποίηση, αλλά δυστυχώς ούτε την «εμφάνιση», ούτε τη λειτουργικότητα, ούτε τις επιδόσεις των native apps μπορούν να προσεγγίσουν, ενώ παράλληλα χρειάζονται μόνιμη σύνδεση της συσκευής με το Internet, προκειμένου να τις «φορτώσει» και «εκτελέσει» από κάποιον απομακρυσμένο server.
- Οι hybrid apps μάλλον αποτελούν την «χρυσή τομή» ανάμεσα στις δύο προηγούμενες φιλοσοφίες ανάπτυξης, καθώς στηριζόμενες στο τρίπτυχο HTML/CSS/JS, προσφέρουν ιδιαίτερα μικρούς χρόνους ανάπτυξης, σε συνδυασμό με μεγάλη ευκολία, και βέβαια το συγκριτικό πλεονέκτημα έναντι των native apps, της «μιας-και-έξω» υλοποίησης για όλες τις πλατφόρμες. Μπορεί να μην είναι απόλυτα συγκρίσιμες με τις native apps ως προς το UI και τις επιδόσεις τους, όμως πραγματικά δεν απέχουν και πολύ, καθώς είναι «ενδεδυμένες» με κομμάτια κώδικα και APIs γραμμένα σε native μορφή, στοιχείο που επίσης τους επιτρέπει να

εκμεταλλεύονται τις δυνατότητες της συσκευής σε πολύ μεγάλο βαθμό.

Συνοψίζοντας λοιπόν κανείς, θα κατέληγε αβίαστα στο συμπέρασμα, πως η ενδεδειγμένη λύση για τον απλό, εύκολο και απροβλημάτιστο τρόπο ανάπτυξης της μικρο-εφαρμογής της παρούσας εργασίας, είναι η λύση του υβριδικού app και της χρήσης διαδεδομένων και προσφιλών τεχνολογιών του διαδικτύου.



**Developers Skillsets Required**  
Native and Hybrid Mobile App Architectures

Native 12 SkillSets	iOS	Objective C	Storyboards	XCode	iOS SDK
	Android	Java	Android XML	Android Studio	Android SDK
	Windows	C# or JS	XAML or HTML5	Visual Studio	WinRT API
Hybrid 4 SkillSets	All Devices	JS	HTML5	Any	Common Aps

*Εικόνα 7: Native vs Web vs Hybrid apps*

### 3.1.2 Διαδικτυακές Τεχνολογίες / Web Technologies

Ας ρίξουμε τώρα, μια πιο προσεκτική ματιά στις περίφημες τεχνολογίες του διαδικτύου, στην HTML5, στα CSS3 και στην Javascript, προκειμένου να τις γνωρίσουμε καλύτερα και κυρίως, να αντιληφθούμε το γιατί θεωρούνται τόσο σημαντικές και γιατί επιλέχθηκαν για την ανάπτυξη του συγκεκριμένου app.

Πριν όμως ξεκινήσουμε την ανάλυση των χαρακτηριστικών για κάθε μια επιμέρους τεχνολογία, καλό είναι να διευκρινιστούν τα ακόλουθα:

- Η γλώσσα σήμανσης (markup) HTML ασχολείται με το περιεχόμενο μιας ιστοσελίδας ή μιας web εφαρμογής
- Η γλώσσα CSS (πιο πρόσφατη έκδοση: CSS3) ασχολείται με την εμφάνιση της ιστοσελίδας ή της web εφαρμογής
- Η γλώσσα σεναρίων JavaScript δίνει «λογική» και διαδραστικότητα στην ιστοσελίδα ή web εφαρμογή

Η αρχή λοιπόν θα γίνει με την HTML5 [10]. HTML=HyperText Markup Language=Γλώσσα Σήμανσης ΥπερΚειμένου. Η HTML5 αποτελεί τη σύγχρονη, βελτιωμένη σε πάμπολλα σημεία και πιο πλούσια σε χαρακτηριστικά έκδοσή της HTML. Πρόκειται ουσιαστικά για τη συνέχεια, επέκταση και μετεξέλιξη της HTML 4.01, XHTML 1.0 & 1.1 και του DOM Level 2 HTML

Δεν αποτελεί νέα έκδοση της HTML και δεν είναι αυτόνομη τεχνολογία καθώς είναι στενά συνυφασμένη με τεχνολογίες όπως CSS3, Javascript οι οποίες συνδυαστικά με τεχνολογίες όπως XML, JSON, AJAX, DOM HTML και διάφορα APIs αποτελούν την πλατφόρμα ανάπτυξης web εφαρμογών HTML5.

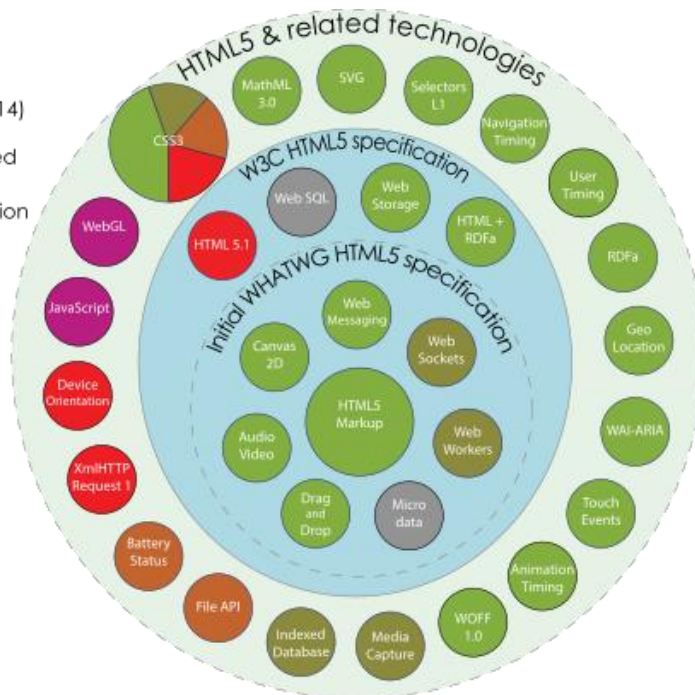
Η HTML5 αποτελεί συνεργασία του World Wide Web Consortium (W3C) και του Web Hypertext Application Technology Working Group (WHATWG). Το WHATWG πρακτικά ασχολείται με αυτό που ονομάζει “living standard” (ζώντα πρότυπο) της HTML5, ένα διαρκώς αναπτυσσόμενο υπερσύνολο web τεχνολογιών.

Το W3C την καθιέρωσε ως πρότυπο (standard) σχετικά πρόσφατα (Οκτώβριος του 2014). Ακολουθεί η έκδοση 5.1 στο τέλος του 2016 και σε εύλογο χρονικό διάστημα πρόκειται να καθιερωθεί και η έκδοση 5.2 [11]

# HTML5

Taxonomy & Status (October 2014)

- Recommendation/Proposed
- Candidate Recommendation
- Last Call
- Working Draft
- Non-W3C Specifications
- Deprecated or inactive



Εικόνα 8: Το τρέχον status της HTML5 και των επιμέρους τεχνολογιών της

Κάνοντας μια σύντομη ιστορική αναδρομή, διαπιστώνει κανείς πως το web έχει αλλάξει δραματικά σε σχέση με το 1999 όταν και έγινε διαθέσιμη στο κοινό η προηγούμενη αναθεώρηση του προτύπου της HTML, η HTML 4.01.

Έκτοτε δημιουργήθηκαν νέες συνήθειες και τάσεις των χρηστών του web, όπως η όλο και ταχύτερη πρόσβαση στο Internet, η έκρηξη στην ανάλυση (δια)δικτυακού περιεχομένου από φορητές συσκευές τύπου smartphones/tablets, η ραγδαία αύξηση της χρήσης multimedia (πολυμέσων), η δημοσίευση κειμένων από εξωτερικά blogs (ιστολόγια) και φόρουμ (διαμερισμός περιεχομένου – content sharing), τα social media, η καλύτερη εκμετάλλευση των μηχανών αναζήτησης κ.α.

Οι νέες αυτές συνήθειες και τάσεις των χρηστών του web στηρίχθηκαν κατά βάση σε τεχνολογικές λύσεις και προϊόντα τρίτων κατασκευαστών όπως π.χ. το Adobe Flash. Επίσης χρησιμοποιήθηκε πάρα πολύ η JavaScript για να προσδώσει διαδραστικότητα στο web περιεχόμενο.

Σε αυτά τα ζητήματα κλήθηκε να δώσει απάντηση το W3C μέσα από την εξέλιξη της HTML. Η σύγχρονη HTML έπρεπε να ικανοποιεί τις τρέχουσες και μελλοντικές τεχνολογικές απαιτήσεις, χωρίς την ανάγκη χρήσης browser plugins (επιπρόσθετα φυλλομετρητών) τρίτων κατασκευαστών ή εκτενούς και πολύπλοκου κώδικα σε JavaScript. Έτσι λοιπόν η HTML5 αναπτύχθηκε έχοντας ως αφετηρία τους παρακάτω στόχους:



- νέα χαρακτηριστικά έπρεπε να προστεθούν στη δομή των HTML, CSS και JavaScript.
- τη μείωση των περιπτώσεων που χρειάζεται η εγκατάσταση plugins στον browser για συγκεκριμένους τύπους στοιχείων (όπως video και audio).
- την καλύτερη διαχείριση σφαλμάτων.
- την προσθήκη περισσότερων ετικετών (tags) οι οποίες θα αντικαταστήσουν κομμάτια κώδικα JavaScript που χρησιμοποιούσαν συχνά οι web designers (προγραμματιστές του διαδικτύου).
- το νέο πρότυπο θα έπρεπε να είναι αυτόνομο χωρίς να χρειάζεται να καλεί κομμάτια κώδικα από άλλα πρότυπα.
- τα βήματα του σχεδιασμού και της ανάπτυξης του νέου προτύπου θα έπρεπε να είναι ορατά στο κοινό.

Ποιοι είναι όμως οι πρακτικοί λόγοι για να επιλέξουν οι web developers την HTML5 έναντι της HTML 4.01; Τα παρακάτω δίνουν την απάντηση στο ερώτημα:

- η HTML5 είναι ήδη γύρω μας και χρησιμοποιείται εκτενέστατα.
- Οι σύγχρονοι browsers υποστηρίζουν στον μέγιστο βαθμό τα standards της HTML5.
- η HTML5 αποδίδει πλέον σημασιολογική (semantic) υπόσταση στο περιεχόμενο του web.
- η HTML5 παρέχει Client-Side Storage (αποθήκευση στον πελάτη, δηλ. στον Η/Υ του χρήστη), ένα χρόνιο πρόβλημα βρήκε λύση!
- η HTML5 επιτρέπει την offline (χωρίς διαδικτυακή σύνδεση δηλαδή) εκτέλεση των web εφαρμογών.
- Διευκολύνεται η εισαγωγή δεδομένων μέσω της HTML5.
- η HTML5 υποστηρίζει ποικιλία APIs που κυριολεκτικά λύνουν τα χέρια των developers (προγραμματιστών), κάνοντας το έργο τους πιο απλό.
- η HTML5 διαθέτει εγγενείς Multimedia δυνατότητες και δεν εξαρτάται από Plugins όπως π.χ. Adobe Flash.
- με την HTML5 γίνεται εύκολη η δημιουργία Cross-Platform (για όλες τις πλατφόρμες υλικού και λογισμικού δηλαδή) παιχνιδιών.
- με το responsive design (προσαρμοστική σχεδίαση) της HTML5, η ίδια ιστοσελίδα ή web εφαρμογή, υλοποιείται μια φορά για ποικιλία συσκευών π.χ. PCs, tablets, smartphones κ.α.
- ο κώδικας γίνεται απλούστερος, πιο περιεκτικός και δυνατός.

Τα βασικά χαρακτηριστικά του νέου προτύπου και οι καινοτομίες που αυτό εισάγει, είναι εν συντομία τα εξής:

- Απλή, συμπαγής και «καθαρή» σύνταξη των ετικετών.
- Νέες ετικέτες με παράλληλη «απόσυρση» παλαιότερων.
- Ελαχιστοποίηση των Plugins.
- Καλύτερη λειτουργία εντοπισμού των λαθών.
- Πλουσιότερη εισαγωγή δεδομένων – Web Forms 2.0.
- Βελτιωμένο και πιο ευέλικτο Validation (επικύρωση) στις Web Forms.
- Σημασιολογικές Ετικέτες και Microformats / Microdata.
- Drag & Drop λειτουργίες (σύρε και άφησε).
- Νέα Δυναμικά Στοιχεία (Elements) και Τεχνική Polyfills.
- Templates (πρότυπα).
- Data Attributes (ιδιότητες στα δεδομένα).
- Αντικείμενα (objects) Audio και Video.
- Δημιουργία 2D/3D Γραφικών με το αντικείμενο Canvas.
- Client Side Storage (Web Storage, Web SQL, Indexed DB).
- Offline εκτέλεση (Application Cache).
- Multi-Threaded Programming (Web Workers).
- Επικοινωνία μεταξύ web εφαρμογών (web sockets).
- Ενσωμάτωση του AJAX μοντέλου.
- File System / File Reader API (διαχείριση του συστήματος αρχείων).
- Promises (ασύγχρονοι διαχειριστές γεγονότων).
- Responsive Web Design.
- Server-Sent Events (ενημέρωση ιστοσελίδας από τον διακομιστή).
- MathML και SVG αντικείμενα
- Νέα και ευέλικτα APIs μέσω JavaScript π.χ. GeoLocation.

Και τέλος, είναι σημαντικό να τονιστεί πως η HTML5 είναι «συμβατή προς τα πίσω», ενώ και για πρώτη φορά καθορίζει κανόνες ανάλυσης (parsing) του DOM tree (δέντρου).

---

Προχωρώντας στην «γλώσσα» CSS3 (Cascading Style Sheets = Διαδοχικά Φύλλα Στυλ ή αλληλουχία φύλλων στυλ) οφείλουμε πάλι να σταθούμε στη διαφοροποίησή τους από την HTML5, καθώς ενώ η τελευταία ασχολείται με το περιεχόμενο των ιστοσελίδων ή των εφαρμογών του διαδικτύου, τα CSS3 ασχολούνται με την εμφάνιση και την «αισθητική» απεικόνιση του περιεχομένου.

Εδώ ακριβώς βρίσκεται και ο λόγος που τα CSS3 είναι τόσο σημαντικά [12]: επειδή πετυχαίνουν τον διαχωρισμό του περιεχομένου

από την «παρουσίαση» του! Πριν την εμφάνιση των CSS, ο web developer έπρεπε να ορίσει με απόλυτο τρόπο τη μορφή του περιεχομένου στα έγγραφα του διαδικτύου μέσω της HTML. Ας δούμε όμως στο σημείο αυτό, πιο συγκεκριμένα τα χαρακτηριστικά και τις ιδιότητές τους.

Τα CSS επιπέδου 3 δεν είναι γλώσσα προγραμματισμού. Είναι ένας μηχανισμός μορφοποίησης ή απόδοσης στυλ (π.χ. χρώμα, μέγεθος, γραμματοσειρά) του περιεχομένου των web εγγράφων. Ως προδιαγραφές συντηρούνται, αναθεωρούνται και τροποποιούνται από το W3C [13].

Δημιουργήθηκαν για πρώτη φορά το 1995/96 στην κοιτίδα του διαδικτύου, το CERN. Στο επίπεδο 2 πέρασαν το 1998, με δύο αναθεωρήσεις στη συνέχεια, η πρώτη το 2011 και η δεύτερη το 2016. Το επίπεδο 3 των CSS είναι οργανωμένο σε (δομικές μονάδες) modules, σε διαφορετικά στάδια προτυποποίησης το καθένα, και γι'αυτό θεωρείται ακόμα και σήμερα «έργο εν εξελίξει». Εντούτοις, η υιοθέτησή τους από όλους τους μοντέρνους browsers και τα σύγχρονα λειτουργικά συστήματα που τρέχουν σε πάσης φύσης συσκευές, είναι σε εξαιρετικά μεγάλο βαθμό.

Τα CSS3 έχουν απλή σύνταξη και χρησιμοποιούν απλές λέξεις της Αγγλικής για να καθορίσουν ιδιότητες. Ένα φύλλο στυλ απαρτίζεται από *κανόνες*. Κάθε κανόνας ή σύνολο κανόνων αποτελείται από έναν ή περισσότερους *επιλογείς* και ένα *τμήμα δηλώσεων*.

Πρακτικά μπορούμε να πούμε πως ο επιλογέας (μέσω μιας τεχνικής «ταιριάζματος» με συγκεκριμένες κλάσεις, αναγνωριστικά και ετικέτες) ταυτοποιεί συγκεκριμένα στοιχεία (HTML elements) του web εγγράφου, πάνω στα οποία εφαρμόζονται οι ιδιότητες μορφοποίησης ή στυλ, που ορίζει το τμήμα δηλώσεων.

Στο παρακάτω τμήμα κώδικα σε CSS3 για παράδειγμα,

```
h1 {  
    color: blue;  
    text-align: center;  
}
```

ορίζεται πως όλες οι επικεφαλίδες “h1” (επιλογέας), θα έχουν χρώμα στη γραμματοσειρά τους Κόκκινο και η Στοίχισή τους θα είναι στο κέντρο του εγγράφου (τμήμα δηλώσεων).

Εάν για κάποιο ή κάποια στοιχεία (elements) της HTML, έχουν οριστεί πάνω από ένας κανόνας στυλ (διαδοχή ή αλληλουχία στυλ), τότε εφαρμόζεται ένας αλγόριθμος προτεραιοτήτων, προκειμένου να

καθοριστεί ποιος κανόνας στυλ θα εφαρμοστεί πρώτος στο/στα στοιχεία, ποιος δεύτερος κ.ο.κ

Η επίδραση που είχαν τα CSS στην «εμφάνιση» και την «παρουσίαση» του διαδικτυακού περιεχομένου σε έγγραφα και εφαρμογές, ήταν (και ακόμα είναι) κολοσσιαία! Χρώματα, σχήματα, εφέ, διαφάνειες, σκιές, κινήσεις και άλλα πολλά, προχώρησαν την αισθητική του web περιεχομένου σε άλλα επίπεδα!

Εκτός όμως από τον πλούτο της εμφάνισης, τα CSS παρουσιάζουν και μια σειρά άλλων πλεονεκτημάτων, τα σημαντικότερα εκ των οποίων είναι:

- Διαχωρισμός του περιεχομένου από την παρουσίασή του, επιτυγχάνοντας έτσι ευελιξία και έλεγχο στα μορφολογικά χαρακτηριστικά και ταυτόχρονα μείωση στην πολυπλοκότητα και την επαναληψιμότητα.
- Εμβέλεια επίδρασης. Μέσω CSS κανόνων μπορεί κανείς να αλλάξει την εμφάνιση ενός ολόκληρου site ή εφαρμογής. Και με ελάχιστες αλλαγές στους κανόνες, να το τροποποιήσει εξ ολοκλήρου.
- Εύρος επίδρασης. Με επιλογή συγκεκριμένων HTML στοιχείων, αποφεύγονται οι κουραστικές και άσκοπες επαναλήψεις κώδικα.
- Αναμορφοποίηση περιεχομένου. Με ελάχιστες αλλαγές ένα διαφορετικό στυλ μπορεί να εφαρμοστεί στην ίδια σελίδα ή συσκευή.
- Προσβασιμότητα. Οι CSS κανόνες βοηθούν την κατανόηση του περιεχομένου και από άτομα με δυσκολία όρασης.

Χωρίς να αγνοούνται μεμονωμένοι περιορισμοί που τα χαρακτηρίζουν και που προς το παρόν δεν έχουν ξεπεραστεί, τα CSS είναι πραγματικά το ιδανικό συμπλήρωμα της HTML5!



Εικόνα 9: HTML5+CSS3+Javascript=η σύγχρονη "γλώσσα" του Διαδικτύου

Ολοκληρώνοντας τη γνωριμία μας με τις διαδικτυακές τεχνολογίες πάνω στις οποίες θα «χτιστεί» η μικρο-εφαρμογή της παρούσας εργασίας, θα αναφερθούμε συνοπτικά στη γλώσσα σεναρίων Javascript.

Η Javascript (με επίσημα προτυποποιημένο όνομα το ECMAScript) είναι μια γλώσσα σεναρίων υψηλού επιπέδου, διερμηνευόμενη, άτυπη (untyped), δυναμική, βασίζεται στα πρωτότυπα (prototype-based) και διαθέτει συναρτήσεις ως αντικείμενα πρώτης τάξης. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού. Έχει επηρεαστεί πολύ από την C όσον αφορά τη σύνταξη της (αλλά και από άλλες γλώσσες στην εξέλιξη της), ενώ αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, χωρίς όμως να σχετίζεται με αυτήν, καθώς έχουν πολύ διαφορετική σημασιολογία [14]. Διαθέτει API για επεξεργασία κειμένου, arrays, ημερομηνιών και κανονικών εκφράσεων (regular expressions) αλλά δεν περιλαμβάνει καθόλου I/O π.χ. για δικτύωση, αποθήκευση ή γραφικά, βασιζόμενη για αυτά στο host περιβάλλον στο οποίο ενσωματώνεται.

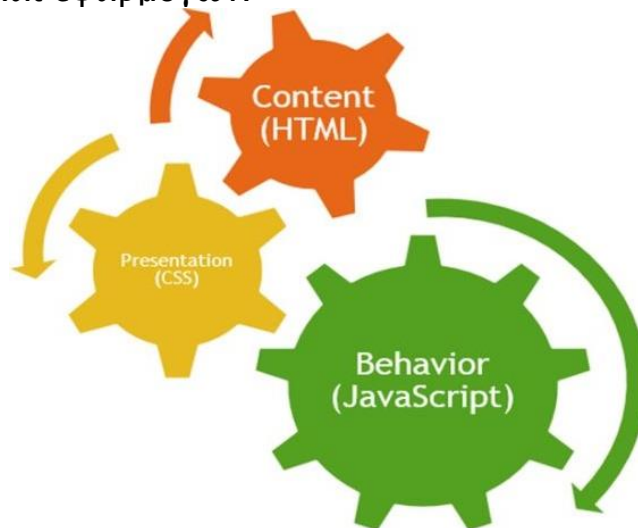
```
if (kairos == "Broxh") {  
    msg = "Θα χρειαστείτε ομπρέλα!";  
}  
else {  
    msg = "Δεν πρόκειται να βρέξει";  
}  
document.getElementById("message").innerHTML = msg;
```

Στο παραπάνω παράδειγμα κώδικα γραμμένου σε Javascript, φαίνεται ο έλεγχος στη μεταβλητή kairos. Ανάλογα του τι περιέχει αυτή, δίνεται αντίστοιχη τιμή στη μεταβλητή msg, που με τη σειρά της δίνει περιεχόμενο στο HTML στοιχείο (element) με ID "message".

Η Javascript είναι παιδί της εταιρείας Netscape Communications. Δημιουργήθηκε το 1995 (με πρώτο όνομα το Mocha και έπειτα το Livescript), εμπνευσμένη από τις γλώσσες προγραμματισμού Self και Scheme και με επιρροές από την C. Το πρώτο συνθετικό "Java" μπερδεύει, αλλά πρόκειται μάλλον για τρυκ μάρκετινγκ από την Netscape Communications, προκειμένου η Javascript να «κλέψει» στα πρώτα της βήματα, λίγο από την αύρα της δημοφιλούς για εκείνη την εποχή γλώσσας Java. Ως πρότυπο καθιερώθηκε το 1996 με το standard ECMA-262.

Πρόκειται για μια εξαιρετικά δημοφιλή γλώσσα σεναρίων, ιδιαίτερα στην κοινότητα των web developers, αρχικά χρησιμοποιούμενη για την εκτέλεση σεναρίων στην πλευρά του πελάτη (client side) και στη συνέχεια σε ευρεία γκάμα εφαρμογών, από επεξεργασία PDF εγγράφων, μέχρι και βιντεο-παιχνίδια. Πλέον χρησιμοποιείται και για την ανάπτυξη εφαρμογών στην πλευρά του εξυπηρετητή (server side, βλ. Node.js)

Καθώς έχει ευρύτατη υποστήριξη από όλους τους μοντέρνους browsers και όλες τις μεγάλες εταιρείες λογισμικού, μπορεί κανείς να ισχυριστεί πως μαζί με την HTML5 και τα CSS3, είναι ο πυρήνας των τεχνολογιών που χρησιμοποιούνται για την ανάπτυξη web περιεχομένου και εφαρμογών.



Εικόνα 10: HTML5+CSS3+JavaScript=τα γρανάζια που κινούν το Διαδίκτυο

Η Javascript διαθέτει μια εντυπωσιακά ευρεία λίστα (βλ. <http://jster.net/catalog>) από πλαίσια ανάπτυξης (Frameworks) και βιβλιοθήκες (Libraries), που συμπληρώνουν, επεκτείνουν και εμπλουτίζουν τη γλώσσα με ιδιαίτερα χαρακτηριστικά και δυνατότητες.

Ίσως η πιο διαδεδομένη βιβλιοθήκη στην παρούσα φάση, η βιβλιοθήκη με την ευρύτερη χρήση και εφαρμογή, είναι η jQuery [15]. Καθώς η μικρο-εφαρμογή της εργασίας θα υιοθετήσει κάποιες από τις τεχνικές της jQuery, θα ήταν χρήσιμη μια σύντομη αναφορά σε αυτήν. Δημιουργήθηκε το 2006 ως μια δωρεάν και open source (ανοικτού κώδικα) βιβλιοθήκη της Javascript, κατάλληλη για όλες τις πλατφόρμες των σύγχρονων λειτουργικών συστημάτων και όλους τους browsers.

Σχεδιάστηκε για να απλοποιήσει το client-side scripting (συγγραφή προγραμματιστικών σεναρίων για την υπολογιστική

συσκευή του χρήστη) της HTML, πρακτικά να κάνει πιο εύκολη και άμεση τη διάσχιση και την επιλογή ή χειρισμό στοιχείων ενός DOM δέντρου. Ουσιαστικά στοχεύει **α)** στον περαιτέρω διαχωρισμό της Javascript και της HTML, **β)** στη σύντμηση και σαφήνεια του κώδικα, **γ)** στον περιορισμό των ασυμβατοτήτων μεταξύ των browsers και **δ)** στην επεκτασιμότητα (κυρίως μέσω των plugins)

Για να γίνει αντιληπτό κυρίως το β), παρατίθεται το ακόλουθο παράδειγμα, σε «καθαρό» κώδικα Javascript συγκρινόμενος με τον αντίστοιχο κώδικα σε jQuery μορφή:

#### JavaScript

```
Function changeBackground(color) {  
    Document.body.style.background = color;  
}  
Onload="changeBackground('red');"
```

#### jQuery

```
$( 'body' ).css ( 'background' , '#ccc' );
```

Ο παραπάνω κώδικας απλώς αλλάζει το χρώμα φόντου του HTML στοιχείου με ετικέτα "body". Η διαφορά μιλάει από μόνη της...

## 3.2 Εργαλεία και πλαίσια ανάπτυξης / Development Tools and Frameworks

Τα εργαλεία που χρησιμοποιήθηκαν στη φάση της ανάπτυξης και της υλοποίησης της μικρο-εφαρμογής είναι το εξής ένα: το ολοκληρωμένο περιβάλλον Intel® XDK (μαζί με το Crosswalk Project web RT), ενώ το πλαίσιο πάνω στο οποίο στηρίχθηκε η ανάπτυξη του app, είναι το Apache Cordova framework. Ξεχωριστή μνεία τέλος, οφείλει να γίνει στο API των χαρτών της εταιρείας Google, το Google Maps Javascript API, μέσω του οποίου κατέστη δυνατή η απεικόνιση του γεωγραφικού εντοπισμού και της πλοήγησης του χρήστη της μικρο-εφαρμογής.

Πριν περάσουμε στην ανάλυση του κώδικα καθαυτού, θα χρειαστεί να δούμε περιληπτικά, τις λειτουργίες και τα χαρακτηριστικά αυτών:

Αρχικά όσον αφορά το ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών (Integrated Development Environment) Intel® XDK, πρέπει να αναφερθεί πως πρόκειται για ένα δωρεάν I.D.E. που παρέχει η εταιρεία Intel, σε προγραμματιστές που ενδιαφέρονται να αναπτύξουν υβριδικές mobile εφαρμογές βασισμένες σε HTML5, CSS3 και Javascript, για συσκευές κάθε τύπου (smartphones, tablets, laptops, Η/Υ), όλων των λειτουργικών συστημάτων (cross platform) και όλων των μοντέρνων browsers. Υποστηρίζει ακόμα (κυρίως ίσως, βλ. σημείωση παρακάτω) την ανάπτυξη τέτοιων εφαρμογών για τον κόσμο του IoT (Internet of Things).

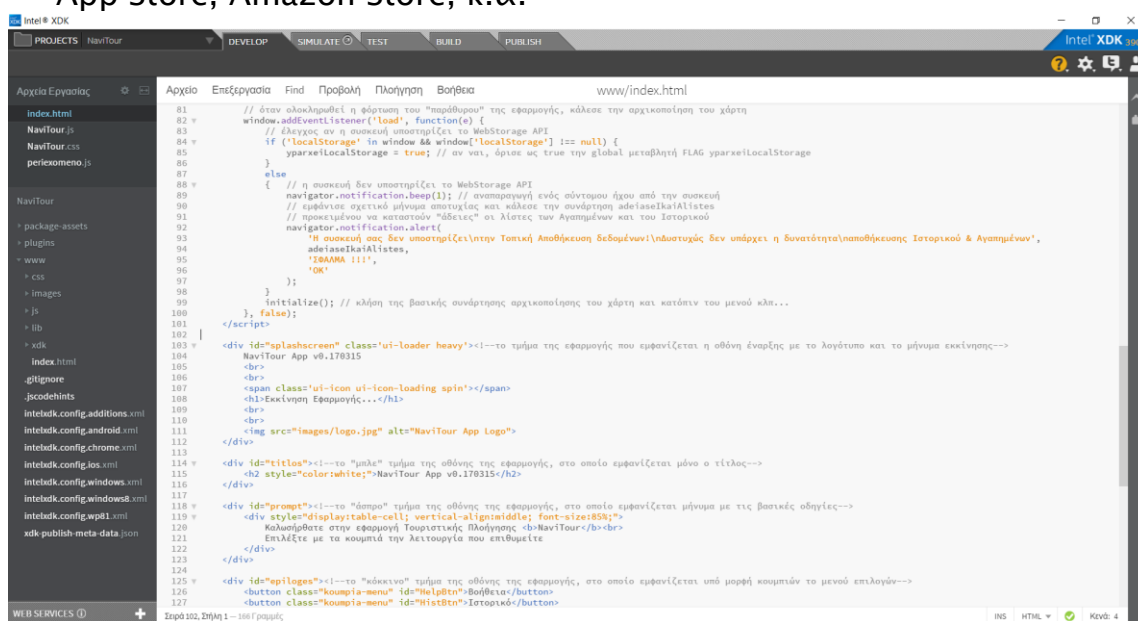
Το Intel® XDK I.D.E. έχει ένα πολύ εύκολο και απλό στο χειρισμό περιβάλλον εργασίας, στο οποίο αφού πρώτα συνδεθείς (μέσω account/δικτυακού λογαριασμού που δημιουργεί ο developer στον ιστότοπο της εταιρείας Intel) δημιουργείς νέο ή ανοίγεις υπάρχον έργο (project). Η δημιουργία ενός νέου project γίνεται με διάφορους τρόπους, π.χ. μέσω έτοιμων προτύπων (templates), μέσω εισαγωγής προ υπάρχοντος κώδικα, μέσω επιλογής τύπου εφαρμογής και χτίσιμό της από «λευκή σελίδα», με τη βοήθεια εξειδικευμένου υποσυστήματος (component) του IDE, του App Designer, όπου με drag'n'drop των διάφορων HTML στοιχείων, μπορεί κανείς πρώτα να «σχεδιάσει» την εφαρμογή πάνω στην οθόνη εικονικής συσκευής και έπειτα να καταπιαστεί με τον κώδικα.

Ουσιαστικά το περιβάλλον εργασίας του Intel® XDK I.D.E. βασιζέται στη λογική των καρτελών, όπου κάθε καρτέλα αντιστοιχεί και σε



διαφορετικό στάδιο ανάπτυξης της εφαρμογής. Οι καρτέλες (tabs) που διαθέτει το I.D.E. είναι οι εξής:

- **Projects:** εδώ γίνεται η δημιουργία, διαγραφή και διαχείριση των έργων του developer
- **Develop (Code/Design):** το βασικό tab εργασίας και ανάπτυξης εφαρμογών, είτε ως κώδικα (Code), είτε ως γραφικά στοιχεία (Design)
- **Simulate:** tab στο οποίο δοκιμάζεται η εφαρμογή σε εικονικές συσκευές διαφορετικών πλατφορμών (android, apple, windows)
- **Test:** εδώ (και με διάφορους τρόπους) γίνεται η δοκιμή της υπό ανάπτυξης εφαρμογής σε πραγματική συσκευή του developer
- **Build:** το συγκεκριμένο tab «παράγει» την τελική «εκτελέσιμη» μορφή της εφαρμογής (π.χ. αρχείο apk για android συσκευές)
- **Publish:** στο τελευταίο tab γίνεται η «έκδοση» της εφαρμογής σε διάφορα online «καταστήματα» εφαρμογών π.χ. Google Play, Apple App Store, Amazon Store, κ.α.



Εικόνα 11: Το Intel XDK IDE, με ανοιχτό το tab DEVELOP για συγγραφή κώδικα

Για τη συγγραφή κώδικα στην καρτέλα DEVELOP (και σε λειτουργία code) ενσωματώνεται ο open source επεξεργαστής (editor) Brackets (βλ. <http://brackets.io>) ένας μοντέρνος και φιλικός στη χρήση editor, με πραγματικά πολλές δυνατότητες και υποστήριξη debugging (εκσφαλμάτωσης) μέσω του Google Chrome (λειτουργία Live Preview όπως την ονομάζει).

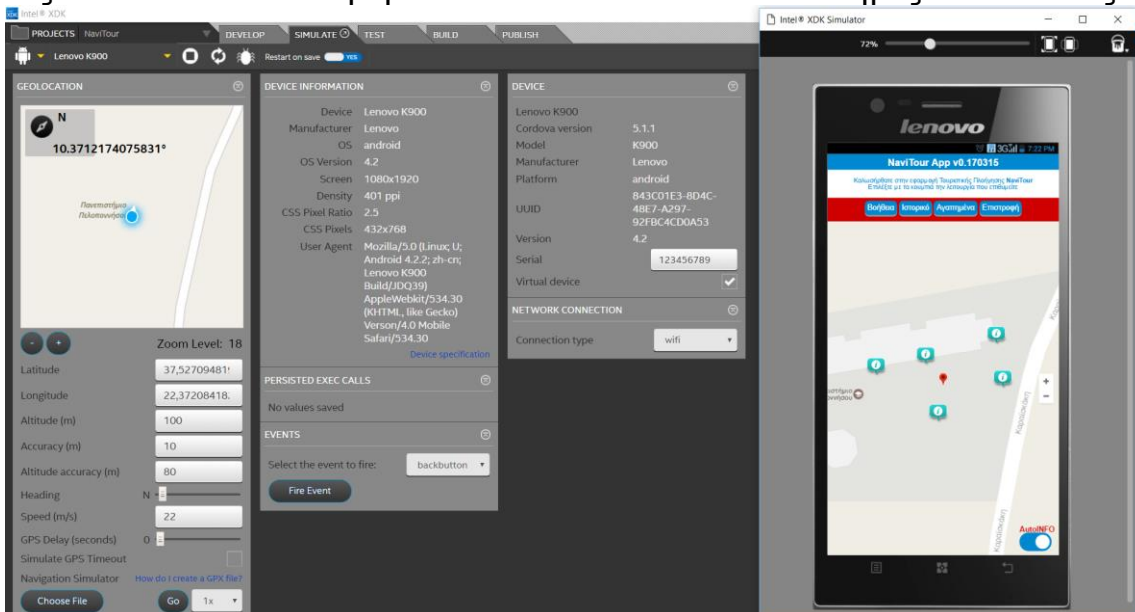
Το Intel® XDK I.D.E. ενσωματώνει επίσης την τελευταία έκδοση του web runtime (βιβλιοθήκη χρόνου εκτέλεσης) Crosswalk Project [16], το οποίο ουσιαστικά εξαλείφει τις ασυμβατότητες και δυσλειτουργίες

που παρουσιάζουν τα web views τόσων διαφορετικών συσκευών (π.χ. έξυπνα κινητά και tablets, με μεγάλη διαφοροποίηση στην παλαιότητα κατασκευής τους, στην αρχιτεκτονική -32/64bit- τους, στο υλικό τους, στις διαστάσεις τους...) και τόσων διαφορετικών λειτουργικών συστημάτων, που και αυτά με τη σειρά τους παρουσιάζουν άλλα προβλήματα (π.χ. ο κατακερματισμός των εκδόσεων του Android).

Με τον όρο Web View εδώ, εννοείται η «μηχανή απεικόνισης» της HTML (rendering engine) του εκάστοτε λειτουργικού συστήματος, δηλαδή ο μηχανισμός με τον οποίο μπορεί ο κώδικας που είναι γραμμένος σε HTML5, CSS3 και Javascript, να «μεταφραστεί» σε περιεχόμενο έτοιμο για προβολή στην οθόνη της εκάστοτε συσκευής.

Η web runtime του Crosswalk Project, ουσιαστικά παρέχει το δικό της “up to date” (ενημερωμένο) και πλήρως δυνατοτήτων Web View, εξασφαλίζοντας έτσι την απρόσκοπτη και απροβλημάτιστη εκτέλεση των εφαρμογών που αναπτύσσει ο developer στο Intel® XDK I.D.E., σε πάσης φύσεως συσκευές και λειτουργικά συστήματα. Να σημειωθεί εδώ πάντως, πως η χρήση του Crosswalk Project είναι προαιρετική και πως, αν τελικά χρησιμοποιηθεί, πρακτικά ενσωματώνεται στο τελικό εκτελέσιμο της εφαρμογής που γίνεται build, αυξάνοντας κατά περίπου 25MB το μέγεθός της.

Επίσης, πρέπει να αναφερθεί πως το build των εφαρμογών που αναπτύσσει κανείς στο Intel® XDK I.D.E., γίνεται όχι σε τοπικό επίπεδο, δηλαδή στον Η/Υ του developer, αλλά σε cloud servers (διακομιστές στο «σύννεφο») της εταιρείας Intel, όπου και οι εφαρμογές συνδέονται και φορτώνονται με όλα τα απαραίτητα plugins και libraries, με τα οποία τις «ντύνουν» τα διάφορα frameworks στα οποία στηρίζονται αυτές.



Εικόνα 12: Το tab SIMULATE ενεργό στο Intel XDK IDE

### Σημείωση:

Δυστυχώς το Intel® XDK I.D.E. ήδη από το δεύτερο μισό του 2016 [17], έχει κάνει γνωστές τις προθέσεις του να επικεντρωθεί σχεδόν αποκλειστικά στον κόσμο του IoT και στην ανάπτυξη μικρο-εφαρμογών για αυτού του είδους τις συσκευές, και σταδιακά να αποχωρήσει από τις υβριδικές εφαρμογές για φορητές συσκευές, καθώς εκτιμά πως οι τεχνολογίες για ανάπτυξη μικρο-εφαρμογών για αυτού του είδους των συσκευών, έχουν πλέον ωριμάσει και τα σχετικά εργαλεία είναι πλέον πολλά και αξιόλογα. Στην τελευταία του έκδοση (3900, Μάρτιος 2017) μάλιστα, και προς αυτή την κατεύθυνση, έθεσε την καρτέλα “Publish” σε κατάσταση «απόσυρσης» με καταληκτική ημερομηνία τον Ιούνιο του 2017, όπου από τότε επίσης θα σταματήσει το build των εφαρμογών μέσω των cloud servers της εταιρείας Intel και πλέον ο developer θα χρειάζεται να το κάνει «με το χέρι» και στον Η/Υ του, χρησιμοποιώντας πλείστα όσα επιπλέον εργαλεία, βιβλιοθήκες και πλαίσια ανάπτυξης για κάθε διαφορετική πλατφόρμα που τον ενδιαφέρει.

---

---

Πριν ασχοληθούμε με το Apache Cordova, ας αναφερθούμε επιγραμματικά και στο Google Maps Javascript API [18], καθώς είναι αυτό το οποίο θα παρέχει υπηρεσίες απεικόνισης γεωγραφικού εντοπισμού και πλοήγησης για τον χρήστη της μικρο-εφαρμογής.

Σαν υπηρεσία χαρτών και πλοήγησης, οι χάρτες της εταιρείας Google είναι ευρέως διαδεδομένοι και καθημερινά χρησιμοποιούνται από εκατομμύρια απλούς χρήστες και επαγγελματίες σε κάθε είδους εφαρμογές και σε πληθώρα ετερογενών συσκευών. Το API τους διατίθεται για τις mobile πλατφόρμες Android και iOS και για όλους τους σύγχρονους browsers.

Μέσω του Google Maps Javascript API δίνεται η δυνατότητα σε web (app) developers ανά τον κόσμο, να ενσωματώσουν τους χάρτες και τις σχετικές υπηρεσίες τους ή / και να ανακτήσουν δεδομένα από αυτούς, σε ιστοσελίδες και διαδικτυακές εφαρμογές, με εύκολο και άμεσο τρόπο, παρέχοντας μεγάλο εύρος επιλογών, από απλή χρήση έως και εκτενή παραμετροποίηση.

---

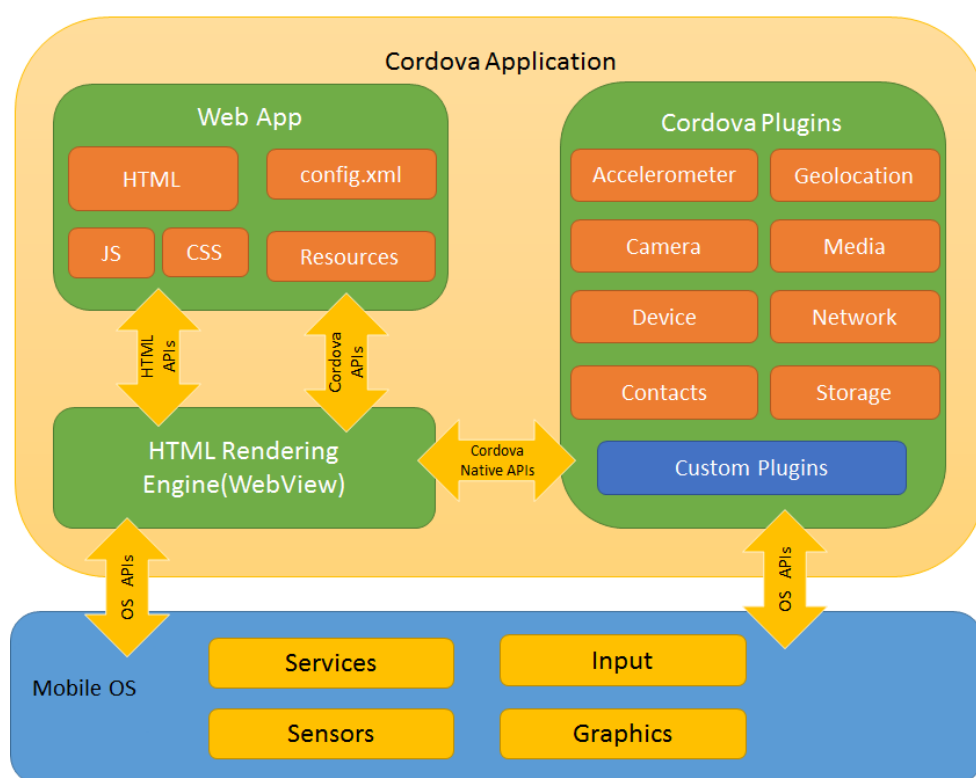
---

Το πλαίσιο ανάπτυξης (framework) Apache Cordova [19], είναι το framework πάνω στο οποίο βασίστηκε η ανάπτυξη της μικρο-εφαρμογής της παρούσας εργασίας – μέσω του Intel® XDK I.D.E. που το ενσωματώνει – προκειμένου να μπορέσει η μικρο-εφαρμογή να

είναι cross platform και συμβατή με τον μεγαλύτερο δυνατό αριθμό συσκευών.

Πρόκειται για το πιο δημοφιλές και ευρέως χρησιμοποιούμενο πλαίσιο ανάπτυξης mobile εφαρμογών, που αρχικά δημιουργήθηκε το 2009 ως PhoneGap από την εταιρεία Nitobi, το 2011 εξαγοράστηκε από την εταιρεία Adobe Systems, που μετονόμασε το προϊόν σε Adobe PhoneGap και στη συνέχεια το έδωσε ως open source framework με τη σημερινή του ονομασία. Ανάμεσα στους βασικούς υποστηρικτές του Apache Cordova project, βρίσκονται μεγάλες εταιρείες του χώρου της Πληροφορικής, όπως οι Adobe, BlackBerry, Google, IBM, Intel, Microsoft, Mozilla, κ.α.

Βασίζεται στις γνωστές τεχνολογίες ανάπτυξης web/mobile περιεχομένου και εφαρμογών HTML5, CSS3 και Javascript, που μαζί με την παράλληλη υλοποίηση πολλών Plugins στη γλώσσα προγραμματισμού κάθε ξεχωριστής πλατφόρμας, «ντύνουν» τον κώδικα κάθε υβριδικής εφαρμογής, δίνοντάς του τη δυνατότητα να έχει πρόσβαση σε native λειτουργίες της κάθε συσκευής, όπως π.χ. επιταχυνσιόμετρο, κάμερα, σύστημα αρχείων, μικρόφωνο, γεωγραφικός εντοπισμός κ.λ.π., αν και οι επιδόσεις στις native λειτουργίες της υβριδικής εφαρμογής, δεν μπορούν να φτάσουν πάντα αυτές των καθαρά native εφαρμογών.



Εικόνα 13: Η αρχιτεκτονική μιας εφαρμογής βασισμένης στο Cordova framework

## Κεφάλαιο 4 / Chapter 4

### Ανάπτυξη και Υλοποίηση μικρο-εφαρμογής / App Development and Implementation

Μετά την αναφορά στα διάφορα εργαλεία, βιβλιοθήκες, APIs και frameworks που απαιτήθηκαν στη φάση της ανάπτυξης και υλοποίησης της μικρο-εφαρμογής, μπορούμε να περάσουμε στην ανάλυση και επεξήγηση του τρόπου με τον οποίο κατασκευάστηκε ο κώδικας του app της τρέχουσας εργασίας.

Συνοψίζοντας τα αμέσως προηγούμενα, επαναλαμβάνουμε πως η μικρο-εφαρμογή κατασκευάστηκε ως μια υβριδική HTML5, CSS3 και Javascript εφαρμογή για τις mobile πλατφόρμες Android, iOS και Windows Phone, κάνοντας χρήση του Intel® XDK I.D.E., διάφορων APIs όπως Geolocation, Google Maps, Web Storage και στηρίζοντας την ανάπτυξή της σε πλαίσια ανάπτυξης και βιβλιοθήκες όπως jQuery, Crosswalk, Apache Cordova.

Η μικρο-εφαρμογή που αναπτύχθηκε ονομάστηκε “NaviTour”, θεωρώντας πως ο τίτλος αντιπροσωπεύει και τη λειτουργία της. Τα χρώματα που επιλέχθηκαν για το U.I. είναι τρία, κόκκινο-άσπρο-μπλε και η συγκεκριμένη χρωματική παλέτα ακολουθεί το U.I. από άκρη σε άκρη, προκειμένου να υπάρχει ομοιογένεια. Η κύρια δομή της μικρο-εφαρμογής έχει ως εξής: Ο φάκελος “www” περιέχει τον κώδικα όλου του app, οργανωμένο με τον ακόλουθο τρόπο:

- Αρχείο “index.html”: το βασικό αρχείο εκκίνησης του app
- Υποφάκελος “css”: περιέχει το αρχείο “NaviTour.css” με όλους τους βασικούς CSS3 κανόνες μορφοποίησης των HTML στοιχείων του app
- Υποφάκελος “images”: περιέχει τα απαραίτητα αρχεία εικόνων
- Υποφάκελος “js”: περιέχει τα Javascript αρχεία που αποτυπώνουν όλη τη λογική και λειτουργικότητα του app
- Υποφάκελος “lib”: περιέχει όλες τις αναγκαίες βιβλιοθήκες για την υποστήριξη του app
- Υποφάκελος “xdk”: περιέχει το αρχείο “project-info.json” που παράγει αυτόματα το I.D.E. με διάφορες ιδιότητες του project

Έξω από τον κύριο φάκελο “www” της μικρο-εφαρμογής, θα βρούμε αρχεία που δημιουργεί αυτόματα το I.D.E. και περιέχουν πληροφορίες και ιδιότητες του project καθ'αυτού, σημαντικά αρχεία xml τύπου

“config” που περιέχουν όλες τις βασικές και απαραίτητες πληροφορίες για την προσαρμογή και εκτέλεση του app σε κάθε διαφορετική πλατφόρμα και επίσης τους φακέλους “.tern-defs” και “plugins” με όλα τα αναγκαία Cordova plugins και τέλος τον φάκελο “package-assets” που περιέχει τα αρχεία εικόνων που θα χρησιμοποιηθούν ως splashscreens και icons (οθόνες εκκίνησης και εικονίδια) της μικρο-εφαρμογής.

#### Σημείωση:

Στον φάκελο “lib”, εκτός από το library jQuery, υπάρχει το library AppFramework [20], μια βιβλιοθήκη κατασκευής Intel, που παρέχει μέσω Javascript, ένα UI framework προσαρμοσμένο για χρήση από HTML5 browsers κάθε σύγχρονης συσκευής, καθώς και η βιβλιοθήκη FastClick [21] που ελαχιστοποιεί το μεσοδιάστημα ανάμεσα στο φυσικό “tap” που κάνει ο χρήστης πάνω σε στοιχείο της οθόνης του app και στην ενεργοποίηση του γεγονότος “click” στον HTML5 browser.

Η μικρο-εφαρμογή, αρχικά δημιουργήθηκε βασιζόμενη στο HTML5+Cordova template του Intel® XDK I.D.E. με τίτλο “single view app”, στην πορεία όμως αφαιρέθηκαν κατά 99% όλα τα προκατασκευασμένα κομμάτια κώδικα του template, σε σημείο που πλέον μπορεί να ισχυριστεί κανείς, πως είναι ένα app «φτιαγμένο από το μηδέν»

Ας δούμε τώρα αναλυτικά και βήμα-βήμα, τη λογική και τον τρόπο κατασκευής του app, εστιάζοντας σε κάθε HTML5, CSS3 και Javascript αρχείο...

● Ξεκινώντας, το αρχείο “index.html” αποτελεί το βασικό αρχείο «εκκίνησης» και «φόρτωσης» του app, είναι το συστατικό μέρος του app που συνδέει το έργο (project) με τις υπόλοιπες βιβλιοθήκες, APIs και Frameworks που είναι απαραίτητα για τις λειτουργίες του, εκτελεί τους αρχικούς διαγνωστικούς ελέγχους, δημιουργεί και οριοθετεί τους βασικούς χώρους και τα πλαίσια λειτουργικότητας πάνω στην οθόνη της συσκευής και καλεί προς ενεργοποίηση τη συνάρτηση αρχικοποίησης initialize(), παραδίδοντας τον έλεγχο της λειτουργίας της μικρο-εφαρμογής στον Javascript κώδικα των αρχείων “NaviTour.js” και “perioxomeno.js” που βρίσκονται στον υποφάκελο “js”.

Στο τμήμα HEAD δηλώνονται τα βασικά στοιχεία και ιδιότητες της μικρο-εφαρμογής, γίνονται οι συνδέσεις/φορτώσεις με τα αρχεία εξωτερικής μορφοποίησης CSS, συνδέονται/φορτώνονται τα απαραίτητα αρχεία JavaScript καθώς και τα απαραίτητα JS Frameworks και APIs. Τέλος, με ενσωματωμένο script ελέγχεται, κατά τη στιγμή που η συσκευή καθίσταται έτοιμη (event "device ready"), αν υπάρχει συνδεσιμότητα με το διαδίκτυο, και τέλος, αποκρύπτει την αρχική οθόνη "splashscreen".

```

<head>
  (.....) <!--ΟΠΟΥ ΥΠΑΡΧΕΙ ΑΥΤΟ, ΠΑΡΑΛΕΙΠΕΤΑΙ (ΜΗ ΣΗΜΑΝΤΙΚΟΣ) ΚΩΔΙΚΑΣ-->
  <!--αρχείο μορφοποίησης CSS των εικονιδίων του AppFramework-->
  <link rel="stylesheet" type="text/css" href="lib/appframework/icons.css" />
  <!--αρχείο μορφοποίησης CSS του U.I. του AppFramework-->
  <link rel="stylesheet" type="text/css" href="lib/appframework/af.ui.css" />
  <!--το βασικό αρχείο μορφοποίησης CSS της εφαρμογής-->
  <link rel="stylesheet" type="text/css" href="css/NaviTour.css" />

  <!--κλήση και φόρτωση της JS βιβλιοθήκης JQuery-->
  <script type="text/javascript" charset="utf-8" src="lib/jquery.min.js"></script>
  <!--κλήση και φόρτωση της JS βιβλιοθήκης FastClick-->
  <script type="text/javascript" charset="utf-8" src="lib/fastclick.min.js"></script>
  <!--κλήση και φόρτωση της JS βιβλιοθήκης του AppFramework-->
  <script (.....) src="lib/appframework/appframework.ui.min.js"></script>
  <!--κλήση και φόρτωση του JS κώδικα με το περιεχόμενο της μικρο-εφαρμογής-->
  <script type="text/javascript" charset="utf-8" src="js/periexomeno.js"></script>
  <!--κλήση και φόρτωση του βασικού "λειτουργικού" JS κώδικα της μικρο-εφαρμογής-->
  <script type="text/javascript" charset="utf-8" src="js/NaviTour.js"></script>

  <!--κλήση και φόρτωση του Google Maps JS API, με συγκεκριμένο προσωπικό κλειδί, καθώς
  και της βιβλιοθήκης γεωμετρικών υπολογισμών-->
  <script async defer
    src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDLFxZk8dP6ZWW1PiSDWVntf5GL
    XLl6Qrw&libraries=geometry">
  </script>

  <!--κλήση και φόρτωση της "ghost" JS βιβλιοθήκης του Cordova Framework-->
  <script type="text/javascript" src="cordova.js"></script>

  <script type="text/javascript">
    // συνάρτηση που απλώς προκαλεί το κλείσιμο της εφαρμογής, με βοήθεια από εδώ
    function killApp() {navigator.app.exitApp();}

    // αναμονή για τη φόρτωση όλων των βιβλιοθηκών API της συσκευής
    document.addEventListener("deviceready", onDeviceReady, false);

    //συνάρτηση που καλείται όταν οι βιβλιοθήκες API της συσκευής είναι πλέον έτοιμες
    function onDeviceReady() {
      if (navigator.onLine) { // έλεγχος αν η συσκευή έχει διαδίκτυακή σύνδεση
        // κρύψε την εναρκτήρια οθόνη μετά από δύο (2) δευτερόλεπτα
        setTimeout(function() {$('#splashscreen').hide();}, 2000);
      }
      else {
        navigator.notification.beep(1); // αναπαραγωγή ήχου από τη συσκευή
        // εμφάνισε σχετικό μήνυμα αποτυχίας και κάλεσε τη συνάρτηση killApp
        // για να κλείσει η εφαρμογή
        navigator.notification.alert(
          'Αποτυχία Σύνδεσης στο Internet!\n
          Ενεργοποιήστε την WiFi/3G/4G σύνδεση\n
          καθώς και τον GPS δέκτη σας\n
          και επανεκκινήστε την εφαρμογή!',
          killApp,
          'ΚΡΙΣΙΜΟ ΣΦΑΛΜΑ !!!',
          'ΕΞΟΔΟΣ'
        );
      }
    }
  </script>
</head>

```

Στο τμήμα BODY τώρα, αρχικά καλείται ενσωματωμένο script, που μετά τη φόρτωση και εμφάνιση του παράθυρου της μικρο-εφαρμογής, ελέγχει το αν η συσκευή υποστηρίζει το WebStorage API και καλεί τη συνάρτηση αρχικοποίησης του app (initialize). Στη συνέχεια ορίζει ένα προς ένα τα τμήματα (div) της οθόνης της μικρο-εφαρμογής.

```

<body>

  <script>
    // όταν ολοκληρωθεί η φόρτωση του "παράθυρου" της εφαρμογής, κάλεσε την
    αρχικοποίηση του χάρτη
    window.addEventListener('load', function(e) {
      // έλεγχος αν η συσκευή υποστηρίζει το WebStorage API, με βοήθεια από εδώ
      if ('localStorage' in window && window['localStorage'] !== null) {
        yparxeiLocalStorage = true;
        // αν ναι, όρισε ως true την global μεταβλητή FLAG yparxeiLocalStorage
      }
      else {
        // η συσκευή δεν υποστηρίζει το WebStorage API
        navigator.notification.beep(1); // αναπαραγωγή ήχου από τη συσκευή
        // εμφάνισε σχετικό μήνυμα αποτυχίας και κάλεσε τη σχετική συνάρτηση
        // adeiaseIkaiAlistes (βλ. perieoxomeno.js) προκειμένου να καταστούν
        // "άδειες" οι λίστες των Αγαπημένων και του Ιστορικού
        navigator.notification.alert(
          'Η συσκευή σας δεν υποστηρίζει\ν
          την Τοπική Αποθήκευση δεδομένων!\n
          Δυστυχώς δεν υπάρχει η δυνατότητα\ν
          αποθήκευσης Ιστορικού & Αγαπημένων',
          adeiaseIkaiAlistes,
          'ΣΦΑΛΜΑ !!!',
          'OK'
        );
      }
      // κλήση της βασικής συνάρτησης αρχικοποίησης του χάρτη, του μενού κλπ...
      initialize();
    }, false);
  </script>

  <!--εδώ θα εμφανίζεται η οθόνη έναρξης με το λογότυπο και το μήνυμα εκκίνησης-->
  <div id="splashscreen" class='ui-loader heavy'>
    NaviTour App v0.170315
    <br>
    <br>
    <span class='ui-icon ui-icon-loading spin'></span>
    <h1>Εκκίνηση Εφαρμογής...</h1>
    <br>
    <br>
    
  </div>

  <!--το "μπλε" τμήμα της οθόνης του app, στο οποίο εμφανίζεται μόνο ο τίτλος-->
  <div id="titlos">
    <h2 style="color:white;">NaviTour App v0.170315</h2>
  </div>

  <!--το "άσπρο" τμήμα της οθόνης που εμφανίζει μήνυμα με τις βασικές οδηγίες-->
  <div id="prompt">
    <div style="display:table-cell; vertical-align:middle; font-size:85%;>
      Καλωσήρθατε στην εφαρμογή Τουριστικής Πλοήγησης <b>NaviTour</b><br>
      Επιλέξτε με τα κουμπιά τη λειτουργία που επιθυμείτε
    </div>
  </div>

  <!--το "κόκκινο" τμήμα της οθόνης, εμφανίζει υπό μορφή κουμπιών το μενού επιλογών-->
  <div id="epilogos">
    <button class="koumpia-menu" id="HelpBtn">Βοήθεια</button>
    <button class="koumpia-menu" id="HistBtn">Ιστορικό</button>
    <button class="koumpia-menu" id="FavBtn">Αγαπημένα</button>
    <button class="koumpia-menu" id="ReturnBtn">Επιστροφή</button>
  </div>

  <!--το κεντρικό τμήμα της οθόνης του app, στο οποίο θα εμφανίζεται ο χάρτης μαζί με
  όλες τις πληροφορίες και ενδείξεις που τυχόν μπορεί να περιέχει, καθώς και
  σχετικό μήνυμα αναμονής σε περίπτωση αποτυχίας φόρτωσης ή εμφάνισης του χάρτη-->

```



```

<div id="perioxi_xarth">
  <section>
    <h1>Εύρεση Τοποθεσίας</h1>
    <br>
    Παρακαλώ περιμένετε...
  </section>
</div>

<div id="Boitheia">
<!--PLACEHOLDER: το κεντρικό τμήμα της οθόνης της μικρο-εφαρμογής, στο οποίο θα
  εμφανίζονται όλες οι οδηγίες χρήσης της εφαρμογής-->
</div>

<div id="Istoriko">
<!--PLACEHOLDER: το κεντρικό τμήμα της οθόνης της μικρο-εφαρμογής, στο οποίο θα
  εμφανίζεται λίστα με το ιστορικό των επισκέψεων του χρήστη-->
</div>

<div id="Agapimena">
<!--PLACEHOLDER: το κεντρικό τμήμα της οθόνης της μικρο-εφαρμογής, στο οποίο θα
  εμφανίζεται λίστα με τα αγαπημένα αξιοθέατα του χρήστη-->
</div>

<div id="floating-panel">
  <!--το τμήμα κάτω δεξιά στην οθόνη του app, που θα εμφανίζεται ο διακόπτης
  ενεργοποίησης/απενεργοποίησης της αυτόματης εμφάνισης πληροφοριών κατά την
  διάρκεια της τουριστικής πλοήγησης του χρήστη-->
  <b style="color:#ce0000;">AutoINFO</b>
  <br>
  <label class="diakopths">
    <!--σε κάθε αλλαγή του διακόπτη κάλεσε τη σχετική συνάρτηση (βλ.NaviTour.js)-->
    <input type="checkbox" checked id="diakopthsInfo" onchange="allagh()">
    <div class="slider"></div>
  </label>
</div>
</body>

```

● Περνώντας στο αρχείο “NaviTour.css” που βρίσκεται στον υποφάκελο “css”, αυτό που πρέπει να αναφερθεί, είναι πως πρόκειται για το αρχείο που καθορίζει την «μορφοποίηση» του περιεχομένου της μικρο-εφαρμογής, δηλαδή το αρχείο που περιλαμβάνει όλους τους CSS3 κανόνες που επηρεάζουν την παρουσίαση και την «αισθητική» του περιεχομένου του app και τη μορφή με την οποία αυτό εμφανίζεται. Είναι χωρισμένο σε τμήματα κανόνων, σχετικούς με κάθε δομικό στοιχείο της εφαρμογής.

Είναι προφανές, πως στο σημείο αυτό, δεν χρειάζεται να παρατεθούν όλοι οι CSS3 κανόνες που περιλαμβάνονται στο αρχείο, παρά μόνο κάποιοι ενδεικτικοί και πιο σημαντικοί για τον καθορισμό της εμφάνισης της μικρο-εφαρμογής:

```

/* τοποθέτηση των στοιχείων σε κάθε λίστα της μικρο-εφαρμογής ένα επίπεδο "πιο μέσα" */
ul { list-style-position: inside; }
/*****ΜΟΡΦΟΠΟΙΗΣΗ ΚΑΘΕΤΗΣ (PORTRAIT) ΕΜΦΑΝΙΣΗΣ*****/
/* με παρόμοιο κανόνα μορφοποιείται και η ΟΡΙΖΟΝΤΙΑ (LANDSCAPE) ΕΜΦΑΝΙΣΗ */
@media (orientation: portrait) {
  #titlos { width:100vw; height:4vh; text-align:center; background-color:#0088d1; }
  #prompt { width:100vw; height:7vh; display:table; text-align:center; background-color:white; color:#0088d1; }
  #epiloges { width:100vw; height:6vh; padding:5px; text-align:center; background-color:#ce0000; }
  #perioch_xarth, #Boitheia, #Istoriko, #Agapimena { width:100vw; height:83vh; display:block; background-color: white; }
}
/*****ΜΟΡΦΟΠΟΙΗΣΗ ΕΜΦΑΝΙΣΗΣ ΤΗΣ ΠΕΡΙΟΧΗΣ ΕΠΙΛΟΓΩΝ*****(με βοήθεια από εδώ)*
.koumpia-menu { /* μορφοποίηση των κουμπιών επιλογής της μικρο-εφαρμογής */
  display: inline-block; /* τοποθέτηση των κουμπιών το ένα δίπλα στο άλλο */
  padding: 5px 5px;
  font-size: 14px;
  cursor: pointer; /* εμφάνιση δείκτη πάνω από τα κουμπιά */
  text-align: center;
  text-decoration: none; /* το κείμενο πάνω στα κουμπιά είναι φυσιολογικό, "σκέτο" */
  outline: none;
  color: #fff;
  background-color: #0088d1;
  border: none;
  border-radius: 7px; /* καμπυλότητα των γωνιών των κουμπιών */
  box-shadow: 0 3px #999; /* εμφάνιση σκιάς γύρω από τα κουμπιά, για επίτευξη 3D εφέ */
}
/* μορφοποίηση των κουμπιών όταν το δάχτυλο (ή το mouse) βρίσκεται από πάνω τους */
.koumpia-menu:hover {
  background-color: #0047b3
}
/* μορφοποίηση των κουμπιών όταν γίνουν ενεργά (δηλαδή πατηθούν) */
.koumpia-menu:active {
  background-color: #0047b3;
  box-shadow: 0 5px #666;
  transform: translateY(4px); /* μετακίνηση των κουμπιών προς τα κάτω */
}
/*****ΜΟΡΦΟΠΟΙΗΣΗ ΕΜΦΑΝΙΣΗΣ ΤΗΣ ΟΘΟΝΗΣ ΒΟΗΘΕΙΑΣ*****/
/* με ακριβώς τον ίδιο τρόπο μορφοποιούνται και οι οθόνες ΙΣΤΟΡΙΚΟ & ΑΓΑΠΗΜΕΝΑ */
#Boitheia {
  padding: 20px;
  color: #0088d1;
  font-size: 1.2em;
  line-height: 150%;
  overflow-y: scroll; /* μόνιμη εμφάνιση της κάθετης scrollbar */
}
/*****ΜΟΡΦΟΠΟΙΗΣΗ ΕΜΦΑΝΙΣΗΣ ΤΟΥ ΣΤΟΙΧΕΙΟΥ FLOATING PANEL*****/
#floating-panel {
  position: absolute;
  bottom: 1%;
  right: 1%;
}

```

```

z-index: 5; /* τοποθέτηση του panel πάνω από οτιδήποτε άλλο */
background-color: transparent;
text-align: center;
line-height: 12px;
}
/*****ΜΟΡΦΟΠΟΙΗΣΗ ΕΜΦΑΝΙΣΗΣ ΤΟΥ ΣΤΟΙΧΕΙΟΥ TOGGLE SWITCH*****(με βοήθεια από εδώ)/
.diakopths { /* τοποθέτηση και μέγεθος του διακόπτη */
    position: relative;
    width: 60px;
    height: 34px;
}

.diakopths input { /* κρύψει τον default διακόπτη της HTML5 */
    display:none;
}

.slider { /* εδώ μορφοποιείται ο συρόμενος διακόπτης */
    position: absolute;
    cursor: pointer;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    background-color: #999;
    border-radius: 34px;
    -webkit-transition: .4s; /* χρόνος μετάβασης του συρόμενου διακόπτη (Apple) */
    transition: .4s; /* χρόνος μετάβασης ("αλλαγής") του συρόμενου διακόπτη */
}

.slider:before { /* εδώ μορφοποιείται το "γέμισμα" του διακόπτη, το "στρογγυλό" δηλαδή */
    position: absolute;
    content: "";
    height: 26px;
    width: 26px;
    left: 4px;
    bottom: 4px;
    background-color: white;
    border-radius: 50%;
    -webkit-transition: .4s; /* χρόνος μετάβασης του συρόμενου διακόπτη (Apple) */
    transition: .4s; /* χρόνος μετάβασης ("αλλαγής") του συρόμενου διακόπτη */
}

input:checked + .slider { /* μορφοποίηση του input όταν επιλεγεί, μαζί με τον διακόπτη */
    background-color: #0088d1;
}

input:focus + .slider { /* μορφ/ση του input όταν λάβει το focus, μαζί με τον διακόπτη */
    box-shadow: 0 0 1px #0088d1;
}

input:checked + .slider:before {
    -webkit-transform: translateX(26px); /* οριζόντια μετακίνηση του input όταν επιλεγεί
(μαζί με τον συρόμενο διακόπτη) για Apple */
    -ms-transform: translateX(26px); /* οριζόντια μετακίνηση του input όταν επιλεγεί
(μαζί με τον συρόμενο διακόπτη) για Microsoft */
    transform: translateX(26px); /* οριζόντια μετακίνηση του input όταν επιλεγεί (μαζί με
τον συρόμενο διακόπτη) */
}
/*****ΜΟΡΦΟΠΟΙΗΣΗ ΕΜΦΑΝΙΣΗΣ ΛΙΣΤΩΝ ΙΣΤΟΡΙΚΟΥ & ΑΓΑΠΗΜΕΝΩΝ*****/
/* εδώ έχουν προστεθεί κομμάτια κώδικα από το W3.CSS, ένα δωρεάν και ελεύθερο προς χρήση
css framework (περισσότερα στο www.w3schools.com/w3css/default.asp) */
.w3-container{padding:0.01em 16px}
.w3-container:after{content:"";display:table;clear:both}
.w3-ul.w3-hoverable li:hover{background-color:#ccc}
ul.w3-ul{list-style-type:none;padding:0;margin:0}ul.w3-ul li{padding:6px 2px 6px
16px;border-bottom:5px solid white}ul.w3-ul li:last-child{border-bottom:none}
.w3-card-4{box-shadow:0 4px 8px 0 rgba(0,0,0,0.2),0 6px 20px 0 rgba(0,0,0,0.19)}
.w3-padding{padding:8px 16px}
.w3-padding-12{padding-top:12px;padding-bottom:12px!important}
.w3-closebtn{-webkit-transition:background-color .25s,color .15s,box-shadow .25s,opacity
0.25s,filter 0.25s,border 0.15s;transition:background-color .25s,color .15s,box-shadow
.15s,opacity .25s,filter .25s,border .15s}
.w3-closebtn{text-decoration:none;float:right;font-size:24px;font-
weight:bold;color:inherit}
.w3-closebtn:hover,.w3-closebtn:focus{color:#ce0000;text-decoration:none;cursor:pointer}
.w3-left{float:left}
.w3-margin-right{margin-right:16px}
.w3-xlarge{font-size:24px}
.w3-xxlarge{font-size:48px}

```

● Συνεχίζοντας με το αρχείο "NaviTour.js" που βρίσκεται στον υποφάκελο "js", θα γίνει απόλυτα κατανοητή η προσέγγιση που ακολουθήθηκε στην «λογική» του σχεδιασμού της μικρο-εφαρμογής.

Το αρχείο "NaviTour.js" είναι το βασικό αρχείο JavaScript της μικρο-εφαρμογής, αυτό στο οποίο εμπεριέχεται όλη η κυρίως λειτουργικότητα και διαδραστικότητα για τον χρήστη. Σε γενικές γραμμές αρχικοποιεί και εμφανίζει τα αντικείμενα και τα διάφορα επιμέρους στοιχεία του app (όπως π.χ. ο χάρτης, οι Markers – «πινέζες», τα InfoWindows–«παράθυρα πληροφόρησης»), εμφανίζει τη βασική διεπαφή προς τον χρήστη (π.χ. το μενού επιλογών), αντιδρά στις επιλογές του, τον ενημερώνει για τις ενέργειές του και πρακτικά συντελεί ουσιαστικά στην επίτευξη των προκαθορισμένων στόχων του προγράμματος. Αναλυτικά:

Αρχικά, η συνάρτηση "initialize" καλείται από το ενσωματωμένο script στο body του αρχείου "index.html" (αφού πρώτα «φορτωθεί» το παράθυρο του app στο web view της συσκευής). Η συνάρτηση είναι υπεύθυνη για την αρχικοποίηση και την εκκίνηση της λειτουργίας της μικρο-εφαρμογής. Πρακτικά μιλώντας, αποδίδει αρχικές τιμές σε βασικά αντικείμενα και δομές δεδομένων όπως π.χ. ο χάρτης ή οι Markers, ελέγχει τη λειτουργία του μενού επιλογών για τον χρήστη (δηλαδή τα κουμπιά), «γεμίζει» τη βασική οθόνη με τον χάρτη και όλα τα σχετικά αντικείμενα πάνω σε αυτόν όπως Markers, InfoWindows, ZoomControl κλπ, διαχειρίζεται τα κλικ του χρήστη πάνω στους Markers και τα InfoWindows του χάρτη και τέλος, με έμμεσο τρόπο ανακτά, παρακολουθεί, ενημερώνει και διαχειρίζεται τη γεωγραφική θέση του χρήστη:

```
function initialize() {
    //πρόσθεσε στο prototype του InfoWindow object, την ιδιότητα "ανοιχτοAuto" (με
    //default τιμή False), προκειμένου να μπορεί να ελεγχθεί ανά πάσα στιγμή, αν υπάρχει
    //στον χάρτη, InfoWindow ανοικτό αυτόματα από τον διακόπτη AutoInfo ή όχι
    google.maps.InfoWindow.prototype.anoixtoAuto = false; //με βοήθεια από εδώ
    //πρόσθεσε στο prototype του InfoWindow object, την ιδιότητα "ανοιχτοTap" (με default
    //τιμή False), προκειμένου να μπορεί να ελεγχθεί ανά πάσα στιγμή, αν υπάρχει στον
    //χάρτη, InfoWindow ανοικτό από τον χρήστη (με κλικ/ταπ δηλαδή) ή όχι
    google.maps.InfoWindow.prototype.anoixtoTap = false; //με βοήθεια από εδώ

    //κάλεσε τη συν/ση "koumpia" για τη διαχείριση και συντονισμό του μενού επιλογών
    koumpia();

    //εδώ ορίζονται όλες οι επιλογές για τον τρόπο εμφάνισης και λειτουργίας του χάρτη
    epilogesXarth = {
        // προεπιλεγμένο σημείο εκκίνησης για τον χάρτη (συντεταγμένες στο U.O.P.)
        center: {lat: 37.527098, lng: 22.372205},
        zoom: 18, //το επίπεδο zoom για τον χάρτη, 18 είναι το μεγαλύτερο zoom
        mapTypeId: 'roadmap', //ο χάρτης να είναι οδικού τύπου
        mapTypeControl: false, //ο χάρτης δεν θα έχει μενού επιλογών
        zoomControl: true, //ο χάρτης θα έχει κουμπιά ρύθμισης του zoom...
        zoomControlOptions: { //με τις ακόλουθες επιμέρους επιλογές...
            style: google.maps.ZoomControlStyle.SMALL, //μικρό μέγεθος
            position: google.maps.ControlPosition.RIGHT_CENTER //στοιχισμένα κεντρ.δεξιά
        }
    }
}
```

```

},
//ο χάρτης δεν θα έχει κουμπί μετάβασης/επαναφοράς σε πλήρη οθόνη
fullscreenControl: false,
scaleControl: false, //ο χάρτης δεν θα απεικονίζει κλίμακα
//ο χάρτης δεν θα έχει επιλογή για μετάβαση σε λειτουργία Street View
streetViewControl: false,
//ο χάρτης με swipe στην οθόνη από τον χρήστη, θα μετακινείται
//(θα κάνει pan δηλαδή και όχι scroll)
gestureHandling: 'greedy'    };

//δημιουργία του χάρτη με τις παραπάνω επιλογές
xarths = new google.maps.Map(doc(.....)Id('perioxh_xarth'), epilogesXarth);

//δημιουργία του InfoWindow για την αρχική θέση του χρήστη πάνω στον χάρτη
infoStart = new google.maps.InfoWindow();

//δημιουργία του InfoWindow που θα παρουσιάζει τις πληροφορίες των σημείων ενδ/ντος
infoParathyro = new google.maps.InfoWindow();

//για όλα τα σημεία ενδιαφέροντος της εφαρμογής, εκτέλεσε τα ακόλουθα:
for (var i=0; i<shmeia.length; i++) { //τα παρακάτω με βοήθεια από εδώ
var shmeio = shmeia[i];
//δημιούργησε Marker πάνω στον χάρτη με τις εξής επιλογές...
var infoMarker = new google.maps.Marker({
    position: {lat: shmeio[1], lng: shmeio[2]}, //συντεταγμένες του Marker
//πάνω σε ποιά αντικείμενο χάρτη θα τοποθετηθεί (ενδέχεται να υπάρχουν και άλλοι χάρτες)
    map: xarths,
    //το εικονίδιο του Marker
    icon: {url: "images/Info.png", scaledSize: new google.maps.Size(40, 40)},
    title: shmeio[0], //ο τίτλος του Marker (φαίνεται με hover)
    //το animation τη στιγμή της τοποθέτησής του πάνω στον χάρτη
    animation: google.maps.Animation.DROP
});

//πρόσθεσε τον Marker στο Array που θα περιέχει όλους τους Markers του χάρτη
infoMarkers.push(infoMarker);

google.maps.event.addListener(infoMarker, 'dblclick', (function (infoMarker,i) {
//ενεργοποιείται όταν ο χρήστης κάνει διπλό κλικ ή ταπ πάνω στον Marker
return function() {//αν υπάρχει ανοικτό I.W. είτε αυτόματα, είτε με ταπ
if ((infoParathyro.anoixtoAuto)|| (infoParathyro.anoixtoTap)) {//τότε
infoParathyro.close(); //κλείσει το ανοικτό I.W.
infoParathyro.anoixtoAuto = false;
infoParathyro.anoixtoTap = false;
}
function onConfirm(buttonIndex) { //αν ο χρήστης επέλεξε το ΝΑΙ
if ((buttonIndex==1) && (yparxeiLocalStorage)) {//και υπάρχει L.S.
var kleidi = 'A' + i; //φτιάξε κλειδί με το σημείο π.χ. A3
var perigraphh = Plhrofories[i][1]; //το value του κλειδιού
localStorage.setItem(kleidi,perigraphh); //add στην L.S.
}
}
//εμφάνισε μήνυμα για αποθήκευση του σημείου στα Αγαπημένα
navigator.notification.confirm(
'Θέλετε να αποθηκεύσετε\n'+
'tο συγκεκριμένο αξιοθέατο\n'+
'('+shmeia[i][0]+')\n'+ //τίτλος του σημείου
'στη λίστα των Αγαπημένων;',
//callback συνάρτηση με το Index του κουμπιού που πατήθηκε
onConfirm,
'ΑΠΟΘΗΚΕΥΣΗ ΣΗΜΕΙΟΥ', //τίτλος παράθυρου διαλόγου
['ΝΑΙ','ΟΧΙ'] //ετικέτες κουμπιών παράθυρου διαλόγου
);
}
})(infoMarker, i) //αφορά τον Marker και το σημείο που είναι τώρα "ενεργό"!
);

google.maps.event.addListener(infoMarker, 'click', (function (infoMarker,i) {
//ενεργοποιείται όταν ο χρήστης κάνει μονό κλικ ή ταπ πάνω στον Marker
return function() {
//βάλει τις πληροφορίες του σημείου στο I.W. και εμφάνισέ το
infoParathyro.setContent(Plhrofories[i][0]); //πρόσθεσε περιεχόμενο
infoParathyro.open(xarths, infoMarker); //τοποθέτησέ το στον Marker
infoParathyro.anoixtoAuto = false;
infoParathyro.anoixtoTap = true;
})(infoMarker, i) //αφορά τον Marker και το σημείο που είναι τώρα "ενεργό"!
);
}
}
//event listener που ενεργοποιείται όταν γίνεται κλικ/ταπ για να κλείσει το ανοικτό I.W.

```

```

google.maps.event.addListener(infoParathyro, 'closeclick', function () {
    //δήλωσε πως το I.W. δεν είναι ανοιχτό με αυτόματο τρόπο
    infoParathyro.anoixtoAuto = false;
    //δήλωσε πως το I.W. δεν είναι ανοιχτό από τον χρήστη (μέσω κλικ/ταπ)
    infoParathyro.anoixtoTap = false;
});

//κάλεσε τη συνάρτηση που θα διαχειριστεί τη θέση του χρήστη πάνω στον χάρτη
EmfaniseThesi();
}

```

Η συνάρτηση "EmfaniseThesi" είναι υπεύθυνη αρχικά να ελέγξει αν μπορεί η συσκευή να υποστηρίξει το Geolocation API. Εάν όχι, τότε θα καλέσει τη συνάρτηση "handleLocationError" για να διαχειριστεί τα όποια λάθη. Εάν ναι, θα λάβει την αρχική θέση του χρήστη πάνω στον χάρτη και θα την εμφανίσει με τον κατάλληλο Marker και το αντίστοιχο InfoWindow. Στη συνέχεια, και σε κάθε ακόλουθη κίνηση του χρήστη, θα μετακινεί τον Marker που τον απεικονίζει στη νέα θέση και θα «μεταβιβάζει» αυτή τη θέση στη συνάρτηση "Proseggise":

```

function EmfaniseThesi() {
    if (navigator.geolocation) { //έλεγχος αν η συσκευή υποστηρίζει το Geolocation API
        //κλήση της μεθόδου "watchPosition" του Geolocation API για να "παρακολουθείται"
        //κάθε αλλαγή στη γεωγραφική θέση του χρήστη
        navigator.geolocation.watchPosition(function(position) {
            //απόδοση στην global μεταβλητή των συντεταγμένων της τρέχουσας θέσης του χρήστη
            pos = {
                lat: position.coords.latitude, //Γεωγρ.Πλάτος
                lng: position.coords.longitude //Γεωγρ.Μήκος
            };
            //αν είναι η πρώτη φορά που λαμβάνεται η γεωγραφική θέση μέσω της μεθόδου "watchPosition"
            if (enarxh==0) {
                //δώσε το timestamp της θέσης στη μεταβλητή enarxh,
                //για να περάσουμε στη φάση "κίνησης" του χρήστη
                enarxh=position.timestamp;
                //όρισε ως κέντρο του χάρτη την τρέχουσα θέση του χρήστη
                xarths.setCenter(pos);
                //μετακίνησε οπτικά τον χάρτη στην τρέχουσα θέση
                xarths.panTo(new google.maps.LatLng(pos));
                //δημιούργησε τον Marker για την απεικόνιση της τρέχ.θέσης του χρήστη πάνω στον χάρτη
                trexMarker = new google.maps.Marker({
                    position: pos, //τοποθέτησε τον Marker στη συγκεκριμένη τρέχουσα θέση
                    map: xarths, //τοποθέτησε τον Marker στον συγκεκριμένο χάρτη
                    //το εικονίδιο του Marker
                    icon: {url: "images/pos.png", scaledSize: new google.maps.Size(32, 32)},
                    title: "Τρέχουσα Θέση", //ο τίτλος του Marker (φαίνεται με hover)
                    //το animation τη στιγμή της τοποθέτησής του πάνω στον χάρτη
                    animation: google.maps.Animation.BOUNCE
                });

                //μετά από 3" σταμάτησε το BOUNCE animation του Marker
                setTimeout(function() {trexMarker.setAnimation(null);}, 3000);

                //εμφάνισε το I.W. που θα ενημερώσει τον χρήστη πως βρέθηκε η αρχική θέση του
                //πρόσθεσε περιεχόμενο στο I.W.
                infoStart.setContent(' (.....)Η τοποθεσία σας εντοπίστηκε!</b>');
                infoStart.open(xarths, trexMarker); //τοποθέτησε το στον κατάλληλο Marker
                setTimeout(function() {infoStart.close();}, 5000); //μετά από 5" κλείσε το I.W.
            }
            //εάν όμως δεν είναι η πρώτη φορά που εντοπίζεται η θέση,
            //αλλά κάποια από τις επόμενες θέσεις του χρήστη
            else {
                trexMarker.setPosition(pos); //μετακίνησε οπτικά τον Marker στην τρέχ.θέση
                //κάλεσε τη συνάρτηση που θα διαχειριστεί την κίνηση του χρήστη
                //πάνω στον χάρτη και μεταβίβασέ της την τρέχουσα θέση
                Proseggise(position);
            }
        }, function() { //κάτι δεν πάει καλά με τον εντοπισμό της γεωγραφικής θέσης!
            //κάλεσε τη συνάρτηση χειρισμού σφαλμάτων και μεταβίβασέ της
            //το I.W. και τη θέση του κέντρου του χάρτη

```

```

        handleLocationError(true, infoStart, xarths.getCenter());
    },
    //παράμετροι στην κλήση της μεθόδου "watchPosition" (υψηλή ακρίβεια εντοπισμού)
    {enableHighAccuracy:true});
} else {//η συσκευή δεν υποστηρίζει το Geolocation API!
    //κάλυψε τη συνάρτηση χειρισμού σφαλμάτων και μεταβίβασέ της
    //το I.W. και τη θέση του κέντρου του χάρτη
    handleLocationError(false, infoStart, xarths.getCenter());
}
}

```

Η συνάρτηση "handleLocationError" καλείται από τη συνάρτηση "EmfaniseThesi", μετά από έλεγχο που έχει κάνει η δεύτερη και έχει διαπιστώσει δυσλειτουργία είτε στον εντοπισμό της γεωγραφικής θέσης του χρήστη, είτε αδυναμία στην υποστήριξη από τη συσκευή του Geolocation API. Και στις δύο περιπτώσεις, εμφανίζει ενημερωτικά μηνύματα προς τον χρήστη, μέσω κατάλληλου InfoWindow τοποθετημένου στο κέντρο του χάρτη. Δέχεται ως παραμέτρους μια boolean μεταβλητή (true για την πρώτη περίπτωση σφάλματος & false για τη δεύτερη), το InfoWindow που θα εμφανίσει το μήνυμα, και το κέντρο του χάρτη:

```

function handleLocationError(browserHasGeolocation, infoStart, pos) {
    infoStart.setPosition(pos); //τοποθέτησε το I.W. στο κέντρο του χάρτη
    infoStart.setContent(browserHasGeolocation ? //πρόσθεσε περιεχόμενο στο I.W.
        //πρώτη περίπτωση σφάλματος (true)
        '<h4 style="color:#ce0000">ΣΦΑΛΜΑ: Αποτυχία Υπηρεσίας Geolocation!</h4>'+
        '<b>Ενεργοποιήστε τον GPS δέκτη σας!</b>':
        //δεύτερη περίπτωση σφάλματος (false)
        '<h4 style="color:#ce0000">ΣΦΑΛΜΑ: Η Συσκευή σας/Το Συστημά σας δεν
        υποστηρίζει την Υπηρεσία Geolocation!</h4>');
}

```

Η συνάρτηση "Proseggise" καλείται από τη συνάρτηση "EmfaniseThesi", η οποία της μεταβιβάζει την τρέχουσα γεωγραφική θέση του χρήστη πάνω στον χάρτη. Ελέγχει κατά πόσον ο χρήστης έχει προσεγγίσει κάποιο από όλα τα σημεία ενδιαφέροντος (ή αντίστοιχα έχει απομακρυνθεί από αυτό). Στην περίπτωση προσέγγισης, ο Marker του σημείου αλλάζει χρώμα (δηλ. το εικονίδιο γίνεται πορτοκαλί) και μετά από διάφορους ελέγχους εμφανίζει InfoWindow με όλες τις τουριστικές πληροφορίες που αφορούν το συγκεκριμένο σημείο ενώ ταυτόχρονα αποθηκεύει το σημείο και την ημερομηνία και ώρα επίσκεψης στην Local Storage. Στην περίπτωση που ο χρήστης βρίσκεται μακριά από κάποιο σημείο ενδιαφέροντος (και πάλι μετά από κατάλληλους ελέγχους) φροντίζει να κλείνει το αυτόματα ανοιχτό InfoWindow (που ενδεχομένως έχει ανοίξει μετά από μια προσέγγιση):

```

function Proseggise(position) {
    var trexonShmeio = new google.maps.LatLng((.....).latitude, (.....).longitude);
    //η μεταβλητή παίρνει την τιμή -1 για να δείξει ότι ο χρήστης
    //βρίσκεται μακριά από κάποιο σημείο ενδιαφέροντος
    var shmeioProseggishs = -1;
    //για όλα τα σημεία ενδιαφέροντος του χάρτη, εκτέλεσε τα κάτωθι:
    for (var i=0; i<shmeia.length; i++) {
        var shmeio = shmeia[i];
    }
}

```



```

var shmeioEndiaferontos= new google.maps.LatLng(shmeio[1], shmeio[2]);
//η μέθοδος "computeDistanceBetween" -από τη βιβλιοθήκη geometry του Google Maps
//API- ελέγχει την απόσταση μεταξύ δύο γεωγρ.σημείων (εν προκειμένου του τρέχοντος
//σημείου που βρίσκεται ο χρήστης και εκάστου σημείου ενδιαφέροντος)
var apostash = (.....).computeDistanceBetween(trexonShmeio, shmeioEndiaferontos);
//εάν ο χρήστης έχει προσεγγίσει το σημείο σε απόσταση μικρότερη των 4m
if (apostash < 4) {
    //όρισε ως τιμή τον α.α. του σημείου ενδιαφέροντος, δήλωσε δηλαδή πως ο
    //χρήστης έχει προσεγγίσει το συγκεκριμένο σημείο
    shmeioProseggishs = i;
    //άλλαξε το εικονίδιο του i Marker (από το Array των Markers)
    infoMarkers[i].setIcon({
        url: "images/Infov.png",
        scaledSize: new google.maps.Size(40, 40)
    });
    //εάν η αυτόματη πληροφόρηση είναι ενεργή ΚΑΙ δεν υπάρχει αυτόματα ανοιχτό
    //I.W. ΚΑΙ δεν υπάρχει ανοιχτό I.W. από τον χρήστη (μέσω κλικ/ταπ)
    if ((automaticInfo) && (!infoParathyro.anoixtoAuto) && (!infoParathyro.anoixtoTap)) {
        //πρόσθεσε το περιεχόμενο που αφορά το συγκεκριμένο σημείο στο I.W.
        infoParathyro.setContent(Plhrofories[i][0]);
        //τοποθέτησε το I.W. στον κατάλληλο Marker
        infoParathyro.open(xarths, infoMarkers[i]);
        //δήλωσε πως το I.W. είναι ανοιχτό με αυτόματο τρόπο
        infoParathyro.anoixtoAuto = true;
        //δήλωσε πως το I.W. δεν είναι ανοιχτό από τον χρήστη (μέσω κλικ/ταπ)
        infoParathyro.anoixtoTap = false;
    }
    if (yparxeiLocalStorage) { //εάν η συσκευή υποστηρίζει το WebStorage API
        var shmera = new Date(); //βάλε στη μεταβλητή την τρέχ.ημ/νία και ώρα
        //μετέτρεψέ τη σε συμβολοσειρά (αυτό είναι πλέον το value του κλειδιού)
        shmera = shmera.toLocaleString();
        var kleidi = 'I' + i; //φτιάξε κλειδί με το σημείο π.χ. I2
        //πρόσθεσέ τα στην localStorage ως στοιχείο της λίστας του Ιστορικού
        localStorage.setItem(kleidi,shmera);
    }
}
}
//εάν ο χρήστης βρίσκεται μακριά (ή έχει απομακρυνθεί μετά από προσέγγιση)
//από κάποιο σημείο ενδιαφέροντος ΚΑΙ υπάρχει αυτόματα ανοιχτό I.W.
if ((shmeioProseggishs == -1) && (infoParathyro.anoixtoAuto)) {
    infoParathyro.close(); //κλείσε το ανοιχτό I.W.
    //δήλωσε πως το I.W. δεν είναι ανοιχτό με αυτόματο τρόπο
    infoParathyro.anoixtoAuto = false;
    //δήλωσε πως το I.W. δεν είναι ανοιχτό από τον χρήστη (μέσω κλικ/ταπ)
    infoParathyro.anoixtoTap = false;
}
}
}

```

Η συνάρτηση "allagh" καλείται κάθε φορά που ο χρήστης απενεργοποιεί ή ενεργοποιεί (κλείνει δηλαδή ή ανοίγει) τον διακόπτη (ToggleSwitch) "AutoInfo" που βρίσκεται στο floating panel της βασικής οθόνης. Αν ο χρήστης ανοίξει τον διακόπτη, τότε ορίζει μια global μεταβλητή ως true. Αν τον κλείσει, τότε φροντίζει να θέσει ως false την παραπάνω global μεταβλητή και μετά από κατάλληλους ελέγχους κλείνει το ενδεχομένως αυτόματα ανοιχτό InfoWindow:

```

function allagh() {
    //όρισε ως τιμή της μεταβλητής το true ή false που θα πάρεις από το ToggleSwitch
    var autoInfo = document.getElementById("diakopthsInfo").checked;
    //εάν πάρεις True (δηλαδή ο χρήστης επέλεξε να ενεργοποιήσει τον διακόπτη)
    if (autoInfo) {
        automaticInfo=true;
    }
    //αλλιώς εάν πάρεις False (δηλαδή ο χρήστης επέλεξε να απενεργοποιήσει τον διακόπτη)
    else{
        automaticInfo=false;
        //εάν υπάρχει αυτόματα ανοιχτό I.W. ΚΑΙ δεν υπάρχει ανοιχτό I.W. από τον χρήστη
        if ((infoParathyro.anoixtoAuto) && (!infoParathyro.anoixtoTap)) {
            infoParathyro.close(); //κλείσε το ανοιχτό I.W.
            //δήλωσε πως το I.W. δεν είναι ανοιχτό με αυτόματο τρόπο
            infoParathyro.anoixtoAuto = false;
            //δήλωσε πως το I.W. δεν είναι ανοιχτό από τον χρήστη (μέσω κλικ/ταπ)
        }
    }
}

```



```

        infoParathyro.anoixtoTap = false;
    }
}

```

Η συνάρτηση "koumpria" είναι μονίμως ενεργή (μέσα από την κλήση της από τη συνάρτηση "initialize") και «περιμένει» το κλικ ή το ταπ του χρήστη σε κάποιο από τα κουμπιά του μενού επιλογών της μικρο-εφαρμογής. Εάν αυτό συμβεί, τότε φροντίζει να αποκρύψει τις μη επιθυμητές από τον χρήστη οθόνες και να εμφανίσει την επιλεγμένη - από το κλικ ή το ταπ του χρήστη - οθόνη, με ταυτόχρονη «φόρτωση» του κατάλληλου περιεχομένου σε αυτήν:

```

function koumpria() {
    //ο χρήστης επέλεξε την οθόνη "ΒΟΗΘΕΙΑ"
    document.getElementById('HelpBtn').onclick = function() {
        $('#perioxh_xarth').hide(); //απέκρυψε τη βασική οθόνη (δηλ.το αντίστοιχο div)
        $('#Istoriko').hide(); //απέκρυψε την οθόνη (το αντίστοιχο div δηλαδή) "ΙΣΤΟΡΙΚΟ"
        $('#Agapimena').hide(); //απέκρυψε την οθόνη (το αντίστοιχο div δηλ.) "ΑΓΑΠΗΜΕΝΑ"
        $('#Boitheia').show(); //εμφάνισε την οθόνη (το αντίστοιχο div δηλαδή) "ΒΟΗΘΕΙΑ"
        //κάλεσε τη συνάρτηση (βλ. περιεχομενο.js) που θα "γεμίσει"
        //με περιεχόμενο την οθόνη "ΒΟΗΘΕΙΑ"
        loadBoitheia();
        //κρύψε τον διακόπτη (ToggleSwitch) "AutoInfo"
        document.getElementById('floating-panel').style.display='none';
    };
    //ο χρήστης επέλεξε την οθόνη "ΙΣΤΟΡΙΚΟ"
    document.getElementById('HistBtn').onclick = function() {
        $('#perioxh_xarth').hide(); //απέκρυψε τη βασική οθόνη (δηλ.το αντίστοιχο div)
        $('#Boitheia').hide(); //απέκρυψε την οθόνη (το αντίστοιχο div δηλαδή) "ΒΟΗΘΕΙΑ"
        $('#Agapimena').hide(); //απέκρυψε την οθόνη (το αντίστοιχο div δηλ.) "ΑΓΑΠΗΜΕΝΑ"
        $('#Istoriko').show(); //εμφάνισε την οθόνη (το αντίστοιχο div δηλαδή) "ΙΣΤΟΡΙΚΟ"
        //κάλεσε τη συνάρτηση (βλ. περιεχομενο.js) που θα "γεμίσει"
        //με περιεχόμενο την οθόνη "ΙΣΤΟΡΙΚΟ"
        loadIstoriko();
        //κρύψε τον διακόπτη (ToggleSwitch) "AutoInfo"
        document.getElementById('floating-panel').style.display='none';
    };
    //ο χρήστης επέλεξε την οθόνη "ΑΓΑΠΗΜΕΝΑ"
    document.getElementById('FavBtn').onclick = function() {
        $('#perioxh_xarth').hide(); //απέκρυψε τη βασική οθόνη (δηλ.το αντίστοιχο div)
        $('#Boitheia').hide(); //απέκρυψε την οθόνη (το αντίστοιχο div δηλαδή) "ΒΟΗΘΕΙΑ"
        $('#Istoriko').hide(); //απέκρυψε την οθόνη (το αντίστοιχο div δηλαδή) "ΙΣΤΟΡΙΚΟ"
        $('#Agapimena').show(); //εμφάνισε την οθόνη (το αντίστοιχο div δηλ.) "ΑΓΑΠΗΜΕΝΑ"
        //κάλεσε τη συνάρτηση (βλ. περιεχομενο.js) που θα "γεμίσει"
        //με περιεχόμενο την οθόνη "ΑΓΑΠΗΜΕΝΑ"
        loadAgapimena();
        //κρύψε τον διακόπτη (ToggleSwitch) "AutoInfo"
        document.getElementById('floating-panel').style.display='none';
    };
    //ο χρήστης επέλεξε την επιστροφή στη βασική οθόνη
    document.getElementById('ReturnBtn').onclick = function() {
        $('#Boitheia').hide(); //απέκρυψε την οθόνη (το αντίστοιχο div δηλαδή) "ΒΟΗΘΕΙΑ"
        $('#Istoriko').hide(); //απέκρυψε την οθόνη (το αντίστοιχο div δηλαδή) "ΙΣΤΟΡΙΚΟ"
        $('#Agapimena').hide(); //απέκρυψε την οθόνη (το αντίστοιχο div δηλ.) "ΑΓΑΠΗΜΕΝΑ"
        $('#perioxh_xarth').show(); //εμφάνισε τη βασική οθόνη (δηλ.την οθόνη του χάρτη)
        //εμφάνισε τον διακόπτη (ToggleSwitch) "AutoInfo"
        document.getElementById('floating-panel').style.display='block';
        //μετακίνησε οπτικά τον χάρτη, με κέντρο του την τρέχουσα θέση του χρήστη
        xarths.panTo(new google.maps.LatLng(pos));
    };
}

```

● Η ανάλυση του κώδικα ολοκληρώνεται με το αρχείο “perioxomeno.js” που βρίσκεται και αυτό στον υποφάκελο “js”. Η ονομασία του αρχείου προδίδει και τη λειτουργία του, καθώς εκεί εμπεριέχονται εν πολλοίς οι πληροφορίες που δίνουν αξία και έννοια στο περιεχόμενο της μικρο-εφαρμογής.

Ειδικότερα, το αρχείο "perioxomeno.js" είναι το αρχείο JavaScript της μικρο-εφαρμογής, που περιέχει όλα τα δεδομένα και τις δομές δεδομένων (μεταβλητές, arrays, objects κλπ) σε καθολικό (global) επίπεδο εμβέλειας, προκειμένου να είναι προσβάσιμα από οποιοδήποτε σημείο της μικρο-εφαρμογής. Επίσης περιέχει τις τουριστικές πληροφορίες και τα γεωστατικά στοιχεία για κάθε σημείο ενδιαφέροντος ή αξιοθέατο του app. Τέλος, περιέχει συναρτήσεις για την εμφάνιση και τη διαχείριση του σχετικού περιεχομένου στην οθόνη «Βοήθεια» και στις λίστες-οθόνες «Ιστορικό» και «Αγαπημένα».

Τμήματα και σημεία του κώδικα στο συγκεκριμένο αρχείο, που επαναλαμβάνονται (έστω με μικρές παραλλαγές) ή «δεν προσθέτουν γνώση» όσον αφορά τις λειτουργίες και τη μεθοδολογία ανάπτυξης της μικρο-εφαρμογής, έχουν σκοπίμως παραληφθεί...

```
//global μεταβλητή τύπου FLAG για την ύπαρξη ή όχι WebStorage
var yparxeiLocalStorage = false;
//global μεταβλητή που δηλώνει τον αρχικό εντοπισμό της τοποθεσίας
//του χρήστη (βλ. function EmfaniseThesi @ NaviTour.js)
var enarxh = 0;
//global μεταβλητή που δηλώνει αν η αυτόματη πληροφόρηση για τον χρήστη είναι
//ενεργοποιημένη ή όχι (βλ. functions Proseggise/allagh @ NaviTour.js)
var automaticInfo = true;
//global αντικείμενο για να δέχεται αντικείμενο τύπου Google Maps position
//(βλ. function EmfaniseThesi @ NaviTour.js)
var pos = null;
//global μεταβλητή που θα "δεχτεί" όλες τις επιλογές για τον χάρτη
//της μικρο-εφαρμογής (βλ. function initialize @ NaviTour.js)
var epilogesXarth = null;
//global μεταβλητή που θα "δεχτεί" τον χάρτη της εφαρμογής
//(βλ. functions initialize/EmfaniseThesi @ NaviTour.js)
var xarths = null;
//global μεταβλητή που θα "δεχτεί" τον Marker για την απεικόνιση της
//τρέχουσας θέσης του χρήστη (βλ. function EmfaniseThesi @ NaviTour.js)
var trexMarker = null;
//global μεταβλητή που θα "δεχτεί" το InfoWindow της αρχικής θέσης του χρήστη
//(ή των σφαλμάτων) (βλ. functions initialize/EmfaniseThesi @ NaviTour.js)
var infoStart = null;
//global μεταβλητή που θα "δέχεται" τα InfoWindows των σημείων ενδιαφέροντος
//της μικρο-εφαρμογής (βλ. functions initialize/Proseggise @ NaviTour.js)
var infoParathyro = null;
//global μεταβλητή τύπου ARRAY που θα "δεχτεί" τις πληροφορίες και
//τις περιγραφές όλων των σημείων ενδιαφέροντος της μικρο-εφαρμογής
var Plhrofories = [];
//global μεταβλητή τύπου ARRAY που θα "δεχτεί" τους τίτλους και
//τις συντεταγμένες όλων των σημείων ενδιαφέροντος της μικρο-εφαρμογής
var shmeia = [];
//global μεταβλητή τύπου ARRAY που θα "δεχτεί" τους Markers για όλα τα σημεία
//ενδιαφέροντος (βλ. functions initialize/Proseggise @ NaviTour.js)
var infoMarkers = [];
```

Η συνάρτηση “loadBoitheia” αναλαμβάνει την «φόρτωση» περιεχομένου στο HTML στοιχείο (element) με ID “Boitheia”, δηλαδή

πρακτικά «γεμίζει» την οθόνη της Βοήθειας με όλα αυτά που πρέπει να εμφανίσει:

```
function loadBoitheia() {
    document.getElementById('Boitheia').innerHTML=
    '<h2 style="color:#ce0000;text-align:center;">Οδηγίες Χρήσης της εφαρμογής</h2>'+
    '<hr><p style="font-size: 1.1em; line-height: 140%;">Με την εφαρμογή NaviTour μπορείτε να
    πλοηγηθείτε και να δείτε ιστορικά στοιχεία, '+
    'φωτογραφίες και πληροφορίες σχετικά με τα διάφορα εκθέματα και κτήρια του Βιομηχανικού
    Μουσείου Φωταερίου. '+
    'Για να λειτουργήσει σωστά η εφαρμογή χρειάζεται απαραίτητα η σύνδεση της συσκευής σας
    στο Internet μέσω Wifi/3G/4G '+
    '(για τη λήψη και εμφάνιση του χάρτη) καθώς και του GPS δέκτη, προκειμένου να εντοπίζει
    το σημείο που βρίσκεστε ή κινείστε.</p>'+
    '<hr><ul><b>Επεξηγήσεις Συμβόλων και Ενεργειών:</b>'+
    (.....);
}
```

Η συνάρτηση “diagrafhIstoixeiou” δέχεται ως όρισμα τον α.α. του αποθηκευμένου στοιχείου της λίστας του Ιστορικού, που έχει επιλέξει ο χρήστης προς διαγραφή και ακολούθως το διαγράφει, αφαιρώντας το από την Local Storage:

```
function diagrafhIstoixeiou(lsID) {
    var iListItemID = 'I'+lsID; //όρισε το στοιχείο προς διαγραφή ως "I"+α.α., π.χ. I4
    localStorage.removeItem(iListItemID); //αφαίρεσέ το από την Local Storage
    loadIstoriko(); //“φόρτωσε” ξανά τη λίστα του Ιστορικού... (πλέον μειωμένη κατά ένα)
}
```

Η συνάρτηση “diagrafhIlistas” καλείται τη στιγμή που ο χρήστης της εφαρμογής πατήσει το κουμπί για να διαγράψει όλη τη λίστα του Ιστορικού. Η όλη διαδικασία εκτελείται επαναληπτικά και με αντίστροφο τρόπο, διότι αλλιώς επηρεάζονται οι α.α. των στοιχείων της λίστας και προκύπτει “σύγχυση”:

```
function diagrafhIlistas(){
    //με αντίστροφη σειρά π.χ. από το (8-1):7 μέχρι το 0
    for (var i = localStorage.length-1; i >= 0; i--) {
        var kleidi = localStorage.key(i);
        //απομονώνουμε τον πρώτο χαρακτήρα του κλειδιού για να δούμε αν είναι "I" ή "A"
        var xar1 = kleidi.charAt(0);
        if (xar1 == 'I') { //αν το κλειδί αφορά στοιχείο της λίστας του Ιστορικού
            localStorage.removeItem(kleidi); //αφαίρεσε αυτό το στοιχείο από την L.S.
        }
    }
    //“φόρτωσε” ξανά τη λίστα του Ιστορικού... (πλέον, δεν περιέχει στοιχεία)
    loadIstoriko();
}
```

Η συνάρτηση “loadKenoIstoriko” αναλαμβάνει την εμφάνιση μιας «κενής» οθόνης Ιστορικού, πρακτικά αυτό που κάνει, είναι να «φορτώνει» στο HTML στοιχείο (element) με ID “Istoriko” ένα section element που απλά περιέχει ενημερωτικό μήνυμα για τον χρήστη:

```
function loadKenoIstoriko() {
    var kenoIstoriko = '<section>'+
        '<h1>Κενό Ιστορικό!!!</h1>'+
        '<br>'+
        'Δεν υπάρχουν εγγραφές στο Ιστορικό...'+
        '</section>';
    document.getElementById('Istoriko').innerHTML = kenoIstoriko;
}
```

Η συνάρτηση "loadIstoriko" είναι υπεύθυνη για την «φόρτωση», εμφάνιση και διαχείριση περιεχομένου στην οθόνη του Ιστορικού. Ουσιαστικά, αφού πρώτα ελέγξει την υποστήριξη από τη συσκευή του WebStorage API, δημιουργεί τη λίστα με όλα τα στοιχεία Ιστορικού επαναληπτικά – αφού πρώτα «εντοπίσει» τέτοια στοιχεία –, αλλιώς καλεί τη συνάρτηση "loadKenoIstoriko". Επίσης εκτελεί τις αναγκαίες ενέργειες σχετικές με τη διαγραφή στοιχείων της λίστας:

```
function loadIstoriko() {
    var istorFound = false; //εξ ορισμού, ορίζεται πως δεν υπάρχει αποθηκευμένο Ιστορικό
    if (yparxeiLocalStorage){
        var iList = '<div class="w3-container">'+
            '<h2 style="color:#ce0000;">Ιστορικό Αξιοθέατων</h2>'+
            '<p style="font-size:1.1em; line-height:140%;">'+
            'Παρακάτω εμφανίζονται (.....) </p>'+
            //προσοχή στο πως καλείται η συνάρτηση diagrafhIlistas στο πάτημα του κουμπιού!
            '<p style=(.....)"><button class="koumpia-menu" id="DelHistBtn"
            onclick="diagrafhIlistas()">Διαγραφή όλων</button></p>'+
            '<ul class="w3-ul w3-card-4" style="background-color:#f5f0f0;">';

        for (var i = 0; i < localStorage.length; i++){ //διάσχιση όλης της Local Storage

            var kleidi = localStorage.key(i);
            //απομονώνουμε τον πρώτο χαρακτήρα του κλειδιού για να δούμε αν είναι "I"ή"A"
            var xar1 = kleidi.charAt(0);
            //απομονώνουμε τον δεύτερο χαρακτήρα του κλειδιού για να δούμε τον α.α. του
            var xar2 = kleidi.charAt(1);

            if (xar1 == 'I') { //αν το κλειδί αφορά στοιχείο της λίστας του Ιστορικού
                istorFound = true; //ορίζουμε πως υπάρχει αποθηκευμένο Ιστορικό!
                iList += '<li class="w3-padding-12">'+
                //προσοχή στο πως καλείται η συνάρτηση diagrafhIstoixeiou στο πάτημα του X
                '<span class="w3-closebtn w3-padding w3-margin-right w3-xxlarge"
                onclick="diagrafhIstoixeiou('+xar2+')">&times;</span>'+
                ''+
                //εμφάνιση του τίτλου για το αποθηκευμένο Ιστορικό στοιχείο
                '<span class="w3-xlarge" style="color:#ce0000;">'+shmeia[xar2][0]+'</span><br>'+
                //εμφάνιση της ημ/νίας και ώρας τελευτ.επίσκεψης για το αποθηκευμένο Ιστορικό στοιχείο
                '<span>'+localStorage.getItem(kleidi)+'</span>'+
                '</li>';
            } // η μεταβλητή iList επιπροσθέτει περιεχόμενο επαναληπτικά!
        }
        iList += '</ul>'+
            '</div>';

        //εάν τελικά υπάρχει αποθηκευμένο Ιστορικό, τότε "φόρτωσε" στο HTML στοιχείο
        //(element) με ID "Istoriko" το περιεχόμενο της μεταβλητής iList
        if (istorFound) {
            document.getElementById('Istoriko').innerHTML = iList;
        }
        else { //αλλιώς κάλεσε τη συνάρτηση "loadKenoIstoriko"
            loadKenoIstoriko();
        }
    }
}
```

Οι επόμενες συναρτήσεις "diagrafhAstoixeiou", "diagrafhAlistas", "loadKenoAgapimena" και "loadAgapimena" ασχολούνται με το περιεχόμενο και τη διαχείριση των στοιχείων της λίστας των Αγαπημένων, με τον αντίστοιχο ακριβώς τρόπο που λειτουργούν και οι συναρτήσεις της λίστας του Ιστορικού που προηγήθηκαν, γι' αυτό και θα παραληφθούν από την αναφορά...

Η συνάρτηση "adeiaselkaiAlistes" καλείται από το ενσωματωμένο script στο body του αρχείου "index.html", στην περίπτωση που η συσκευή δεν υποστηρίζει το WebStorage API. Με τη σειρά της, καλεί τις συναρτήσεις "loadKenoIstoriko" και "loadKenoAgapimena", προκειμένου να εμφανίσουν κενά περιεχόμενα στις αντίστοιχες οθόνες «Ιστορικού» και «Αγαπημένων»:

```
function adeiaselkaiAlistes(){
    loadKenoIstoriko();
    // απέκρυψε την οθόνη του "Ιστορικού" καθώς παίρνει άμεσα το focus
    $('#Istoriko').hide();
    loadKenoAgapimena();
    // απέκρυψε την οθόνη των "Αγαπημένων" καθώς παίρνει άμεσα το focus
    $('#Agapimena').hide();
}
```

```
var Plhroforia01 =
    '<h1 style="color:#ce0000;text-align:center;">Εργοστάσιο Φωταερίου Αθηνών</h1>'+
    '<p style="text-align:center;"></p>'+
    '<p style="color:#0088d1; overflow-y:auto;"><b>Η ιστορία του εργοστασίου:</b> Το 1857 ο γάλλος Φραγκίσκος Φεράλδι (Francois Feraldi) '+
    '(.....)</p>'+
    '<div style="text-align:center; color:black;">Πηγή: <a href="https://goo.gl/d1Vrb4" target="_blank">Τεχνόπολις</a>'+
    ' <sub><small>(τελευταία πρόσβαση: 6 Φεβρουαρίου 2017)</small></sub></div>';

var Perigrafh01 = "Στο συγκεκριμένο αξιοθέατο μπορείτε να δείτε ιστορικά στοιχεία ανά περίοδο, σχετικά με το Εργοστάσιο Φωταερίου Αθηνών.";

var Plhroforia02 = (.....)
var Perigrafh02 = (.....)

var Plhroforia03 = (.....)
var Perigrafh03 = (.....)

var Plhroforia04 = (.....)
var Perigrafh04 = (.....)

var Plhroforia05 = (.....)
var Perigrafh05 = (.....)

//Array που περιέχει τις τουριστικές πληροφορίες και τις περιγραφές για όλα τα σημεία
//ενδιαφέροντος που εμφανίζονται στη μικρο-εφαρμογή
Plhrofories = [
    [Plhroforia01,Perigrafh01],
    [Plhroforia02,Perigrafh02],
    [Plhroforia03,Perigrafh03],
    [Plhroforia04,Perigrafh04],
    [Plhroforia05,Perigrafh05]
];

//Array που περιέχει τους τίτλους και τις γεωγραφικές συντεταγμένες όλων των σημείων
//ενδιαφέροντος της μικρο-εφαρμογής
shmeia = [
    ['Εργοστάσιο Φωταερίου Αθηνών', 37.527136, 22.371748],
    ['Βιομηχανικό Μουσείο Φωταερίου', 37.527179, 22.371996],
    ['Μηχανουργείο', 37.527260, 22.372351],
    ['Αεριοφυλάκιο 1', 37.526953, 22.372058],
    ['Νέο Υδατάεριο', 37.527087, 22.372382]
];
```

## Κεφάλαιο 5 / Chapter 5

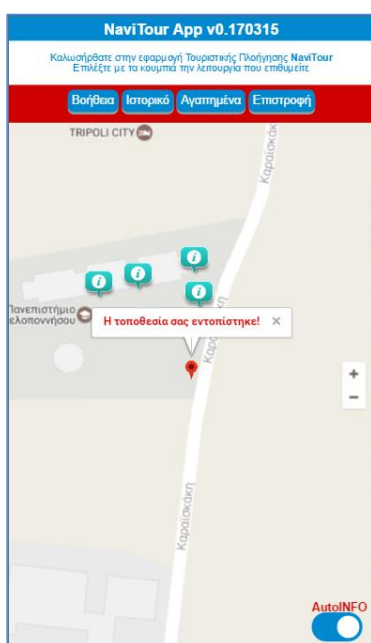
### Παραδείγματα χρήσης της μικρο-εφαρμογής / App Use Cases

Για το app της τρέχουσας εργασίας, χρησιμοποιήθηκαν μόλις πέντε (05) γεωγραφικά σημεία ενδιαφέροντος (αξιοθέατα), είναι όμως αρκετά να αναδείξουν τις λειτουργίες και της δυνατότητες της μικρο-εφαρμογής. Στο παρόν κεφάλαιο της εργασίας, θα παρουσιαστούν με τη χρήση στιγμιοτύπων οθόνης (screenshots), τυπικά σενάρια χρήσης του app σε πραγματικές συνθήκες.



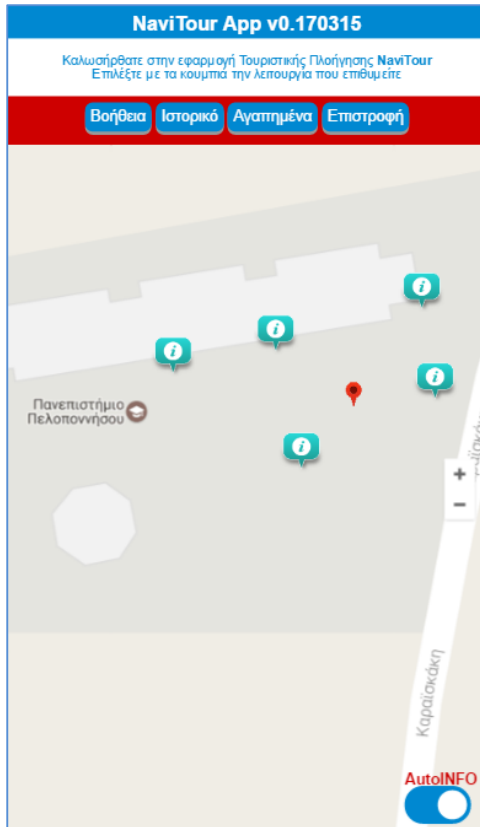
Στο πρώτο screenshot εμφανίζεται το Splashscreen (η οθόνη εκκίνησης δηλαδή) που αντικρύζει ο χρήστης αμέσως μετά το άνοιγμα της μικρο-εφαρμογής

Εικόνα 14: Η οθόνη εκκίνησης του app



Στο δεύτερο screenshot εμφανίζεται το ενημερωτικό μήνυμα (σε μορφή InfoWindow) για την εύρεση του γεωγραφικού σημείου στο οποίο βρίσκεται ο χρήστης, ο οποίος απεικονίζεται με την κόκκινη «πινέζα»

Εικόνα 15: Εύρεση τοποθεσίας του χρήστη



Εικόνα 16: Μετακίνηση του χρήστη

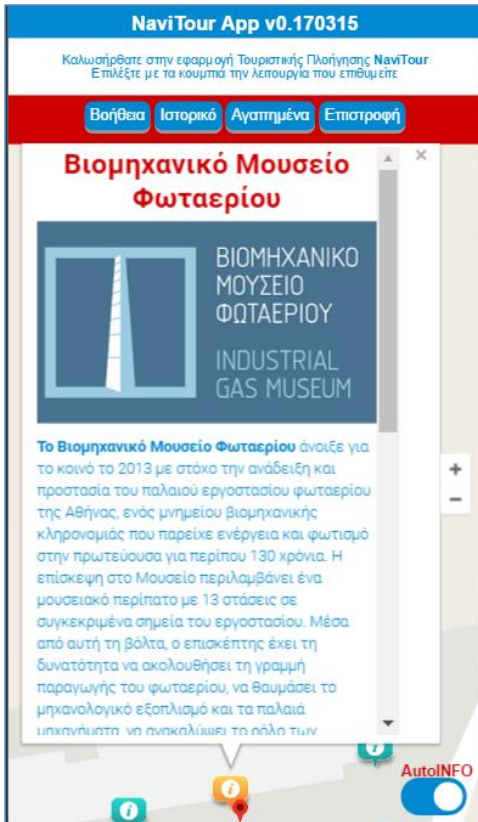
Το συγκεκριμένο screenshot απεικονίζει τον χρήστη, καθώς αυτός κινείται ανάμεσα στα αξιοθέατα προκειμένου να τα προσεγγίσει. Παρατηρούμε επίσης τα εικονίδια για τα αξιοθέατα (Markers), τα χειριστήρια για το zoom, τον διακόπτη AutoInfo, το μενού επιλογών (κουμπιά) & τον Τίτλο/Υπότιτλο του app



Εικόνα 17: Προσέγγιση του χρήστη σε σημείο ενδιαφέροντος

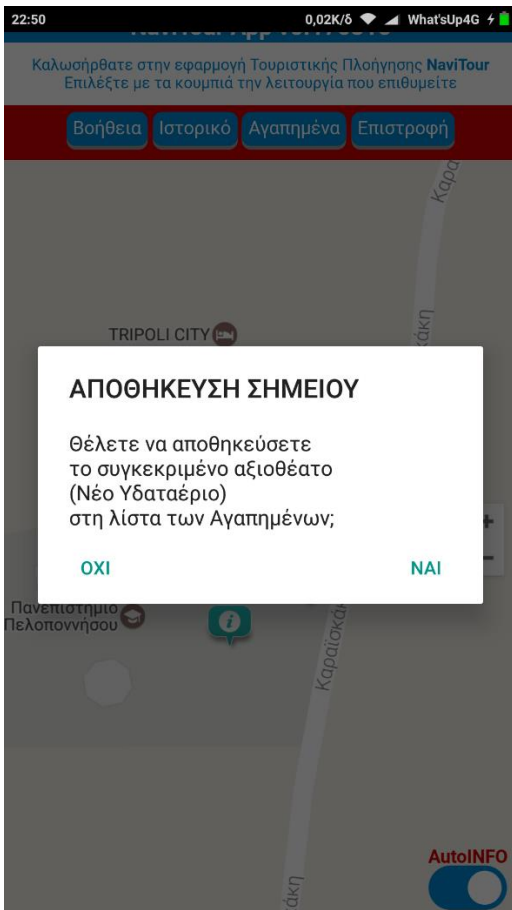
Εδώ βλέπουμε τον χρήστη καθώς έχει προσεγγίσει ένα αξιοθέατο και αυτόματα εμφανίζεται παράθυρο InfoWindow με όλες τις χρήσιμες τουριστικές πληροφορίες για το συγκεκριμένο σημείο. Παρατηρούμε επίσης, πως αλλάζουν χρώμα οι Markers που έχει επισκεφτεί ο χρήστης!





Εικόνα 18: Παράθυρο InfoWindow

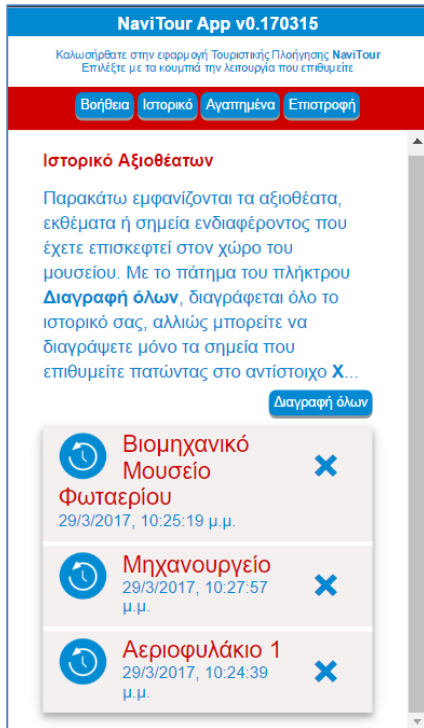
Στιγμιότυπο οθόνης που παρουσιάζει ανοικτό παράθυρο InfoWindow – με πλήθος πληροφοριών για σημείο ενδιαφέροντος – και πως προσαρμόζει το περιεχόμενό του κάνοντας χρήση scrollbar...



Εικόνα 19: Ερώτηση για αποθήκευση στα Αγαπημένα

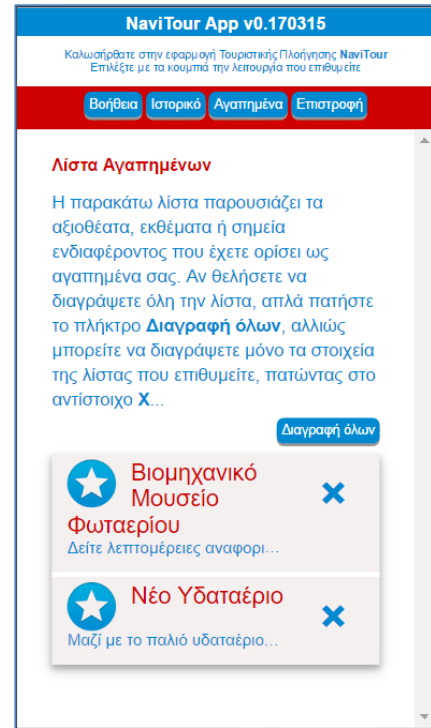
Το συγκεκριμένο screenshot εμφανίζει το παράθυρο διαλόγου με το οποίο η μικρο-εφαρμογή ρωτάει τον χρήστη αν επιθυμεί τη δημιουργία Αγαπημένου σημείου ενδιαφέροντος. Αμέσως προηγουμένως έχει προηγηθεί το διπλό «ταπ» του χρήστη πάνω στο συγκεκριμένο σημείο.





Εικόνα 20: Η οθόνη με τη λίστα του Ιστορικού

Οι οθόνες  
Ιστορικού  
και  
Αγαπημένων



Εικόνα 21: Η οθόνη των Αγαπημένων



Εικόνα 22: Η οθόνη της Βοήθειας

Screenshot που απεικονίζει  
την οθόνη με τις οδηγίες  
χρήσης της μικρο-εφαρμογής

## Κεφάλαιο 6 / Chapter 6

### Συμπεράσματα – Μελλοντικές κατευθύνσεις / Conclusions – Future Directions

Τα συμπεράσματα στα οποία καταλήγει η εργασία με το πέρας της διαδικασίας σχεδιασμού, ανάπτυξης και υλοποίησης της μικρο-εφαρμογής, είναι αρκετά και ιδιαίτέρως ενδιαφέροντα.

Η μικρο-εφαρμογή επιτέλεσε τον σκοπό της. Τα συγκεκριμένα σημεία στα οποία στόχευσε, όπως αναπτύχθηκαν εκτενώς στο κεφάλαιο 2.1 υπό μορφή απαιτήσεων, ικανοποιήθηκαν πλήρως.

Το app δόθηκε για τεστ σε περιορισμένο πλήθος χρηστών (4 ατόμων), οι οποίοι – αφού πρώτα διευκρινίστηκε ο τρόπος χρήσης του – το δοκίμασαν σε πραγματικές συνθήκες, δηλώνοντας στη συνέχεια απόλυτα ικανοποιημένοι από τις λειτουργίες της μικρο-εφαρμογής και το περιβάλλον διεπαφής της, και γενικότερα ευχαριστημένοι από την εμπειρία χρήσης της.

Όσον αφορά τις προδιαγραφές που ορίστηκαν στο κεφάλαιο 2.2 και ιδίως το τεχνικό τους μέρος, το app τελικά μπόρεσε να τις υλοποιήσει σε μεγάλο βαθμό, αλλά όχι πλήρως. Δυστυχώς, το app παρόλο που εκτελείται άριστα στις πλατφόρμες του Android και των Windows 10 (τόσο σε επίπεδο προσομοίωσης, όσο και σε πραγματικές συσκευές), δεν μπόρεσε να γίνει build για την πλατφόρμα της Apple (iOS). Η αποτυχία οφείλεται στο ότι το Intel® XDK I.D.E. προκειμένου να επιτρέψει στον developer μιας μικρο-εφαρμογής να την κάνει build για τη συγκεκριμένη πλατφόρμα, απαιτεί την καταχώρηση γνήσιου πιστοποιητικού (developer certificate) από την εταιρεία Apple. Όμως για την έκδοση αυτού του πιστοποιητικού, η εταιρεία ζητάει την ενεργοποίηση συνδρομής με ελάχιστος κόστος τα 99\$, γεγονός που υπερβαίνει το νόημα της τρέχουσας εργασίας. Η απροβλεπτή εκτέλεση του app σε περιβάλλον προσομοίωσης για iOS συσκευές – μέσα από το Intel® XDK I.D.E. – εντούτοις, προΐδεάζει για την εύκολη, απλή και άμεση υλοποίησή του στην πλατφόρμα της Apple, εάν και όποτε κριθεί αυτό αναγκαίο.

Η μικρο-εφαρμογή εγκαταστάθηκε και λειτούργησε με επιτυχία σε τρία Android smartphones (εκδόσεων 4.4 – 5 – 6) και ένα Android tablet έκδοσης 4.1, καθώς και σε ένα έξυπνο κινητό Windows Phone έκδοσης 10. Με μόνη αστοχία την αδυναμία εύρεσης γεωγραφικού στίγματος

σε τακτά χρονικά διαστήματα, αλλά πιο αραιά (π.χ. κάθε 30–40”) στη συσκευή με Windows 10, και με μόνη πιθανή εξήγηση την προβληματική και μειωμένη απόδοση στη λειτουργία του ενσωματωμένου GPS δέκτη της συσκευής, μπορεί να ισχυριστεί κανείς, πως το υλικό/λογισμικό των συσκευών συνεργάστηκε αρμονικά με τη μικρο-εφαρμογή.

Σχετικά με τη διαδικασία της ανάπτυξης του app, το ίδιο το περιβάλλον του Intel® XDK I.D.E., είναι ιδιαίτερα φιλικό και χρήσιμο για τον προγραμματιστή τέτοιου είδους εφαρμογών. Αποτελεί πράγματι κρίμα η «μεταβολή» στην κατεύθυνσή του, που επιχειρείται το τελευταίο διάστημα από την ιδιοκτήτρια εταιρεία, καθώς θα «φτωχύνει» σε βασικά και αναγκαία χαρακτηριστικά, με ορατό κίνδυνο ποσοστό των developers που το χρησιμοποιούν, να μεταναστεύσει πιθανώς σε «ανταγωνιστικά» προϊόντα.

Ερχόμενοι στις τεχνολογίες που χρησιμοποιήθηκαν στην τρέχουσα εργασία για την ανάπτυξη του app, μπορούμε να ισχυριστούμε πως πλέον βρίσκονται σε υψηλότατο επίπεδο ολοκλήρωσης και ωριμότητας, επιτρέποντας στον developer να ασχοληθεί σχεδόν αποκλειστικά με τη λογική και τον αλγόριθμο της εφαρμογής που αναπτύσσει, χωρίς να ξοδεύει χρόνο και προσπάθεια στην κατανόηση των «τι» και των «πως» για κάθε επιμέρους συστατικό της εφαρμογής. Με άλλα λόγια, το τρίπτυχο HTML5/CSS3/Javascript μαζί με τα συνοδευτικά τους APIs, libraries και frameworks, «λύνει» κυριολεκτικά τα χέρια των απανταχού web developers και πείθει πως είναι απόλυτα ικανό για εύκολη, απλή και γρήγορη ανάπτυξη πάσης φύσεως εφαρμογών για όλες τις διαφορετικές συσκευές, πλατφόρμες υλικού και λογισμικού και λειτουργικά συστήματα που απαρτίζουν τον τεχνολογικό κόσμο του σήμερα. Συνεπώς, μόνο τυχαία δεν είναι η ευρύτατη αποδοχή που γνωρίζουν αυτές οι τεχνολογίες από την παγκόσμια κοινότητα των προγραμματιστών.

---

Όσον αφορά τις μελλοντικές κατευθύνσεις που θα μπορούσε να πάρει η τρέχουσα εργασία σχετικά με την περαιτέρω ανάπτυξη του app, τη βελτίωση και επέκταση των δυνατοτήτων του, καθώς και την προσθήκη επιπλέον χαρακτηριστικών σε αυτό, τα πεδία εστίασης είναι τα εξής:

- η μικρο-εφαρμογή μπορεί να απευθυνθεί σε δυνητικά ευρύτερο κοινό, με την (απλή σχετικά, καθώς το επιτρέπει το globalization API του Apache Cordova) μετατροπή της σε πολύγλωσσο app, και όχι

μόνο στα ελληνικά (π.χ. με την προσθήκη βασικών ευρωπαϊκών γλωσσών)

- εκτός από την οπτική ενημέρωση και πληροφόρηση του χρήστη (μέσω των InfoWindows), θα πρέπει να ενσωματωθεί και η ακουστική, όπως για παράδειγμα η αναγγελία των σημείων ενδιαφέροντος κατά την προσέγγιση του χρήστη σε αυτά, η μετατροπή του κειμένου σε ομιλία (π.χ. μέσω της λειτουργίας T.T.S.) για AMEA κ.α.
- πολύ χρήσιμο νέο χαρακτηριστικό θα είναι η δυνατότητα του χρήστη να επιλέγει επιθυμητό αξιοθέατο και η μικρο-εφαρμογή να τον «οδηγεί» προς αυτό, με βήμα-προς-βήμα οπτικές και φωνητικές οδηγίες
- στις σύγχρονες τάσεις των μικρο-εφαρμογών για φορητές συσκευές, είναι και η δυνατότητά τους να «διαμοιράζονται» (share) το περιεχόμενό τους μέσω των social media, χαρακτηριστικό που θα βελτιώσει την εμπειρία χρήσης του τρέχοντος app, ιδίως ανάμεσα σε χρήστες νέους σε ηλικία (π.χ. share στοιχείων του Ιστορικού ή των Αγαπημένων μέσω Facebook ή Twitter κλπ)
- ως επέκταση ιδιαίτερα σημαντική αλλά και αυξημένης πολυπλοκότητας ως προς την υλοποίησή της, πρέπει να θεωρηθεί η δυνατότητα της μικρο-εφαρμογής να κάνει διαχείριση του περιεχομένου της με εξωτερικό UI (και όχι μέσω περιεχομένου που «προσθέτει με το χέρι» ο app developer εσωτερικά στον κώδικα, όπως συμβαίνει επί του παρόντος). Αυτό από τη μεριά του απλού χρήστη δεν τυγχάνει χρησιμότητας, όμως για οργανισμό/επιχείρηση που έχει στη διάθεσή του/της τη μικρο-εφαρμογή και θέλει να τροποποιεί, αφαιρεί, επεκτείνει, ενημερώνει ανά τακτά χρονικά διαστήματα το περιεχόμενό της, αποτελεί κρίσιμο χαρακτηριστικό!

## Βιβλιογραφία / References & Ιστοσελίδες / Websites

- [1] [www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/](http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/), τελευταία πρόσβαση στις 17/03/2017
- [2] [http://dl.acm.org/results.cfm?query=acmdlTitle%3A%28%252Bmobile%20%252Bapp%20cultural%20heritage%20museum%20guide%20our%20navigation%20application%29&start=0&filtered=&within=owners%2Eowner%3DHOSTED&dte=&bfr=&srt=\\_score](http://dl.acm.org/results.cfm?query=acmdlTitle%3A%28%252Bmobile%20%252Bapp%20cultural%20heritage%20museum%20guide%20our%20navigation%20application%29&start=0&filtered=&within=owners%2Eowner%3DHOSTED&dte=&bfr=&srt=_score), τελευταία πρόσβαση στις 17/03/2017
- [3] Cushing, A., Cowan, B., (2016) *Walk1916: Exploring how a mobile walking tour app can provide value for LAMs*, ASIST 2016, Copenhagen, Denmark
- [4] Bouwer, A., Nack, F., El Ali, A., (2012) *Lost in Navigation: Evaluating a Mobile Map App for a Fair*, ICMI 2012, Santa Monica, California, USA
- [5] Alvermann, J., (2016) *Mobile Media in the Museum Space: the Example of the Neanderthal Museum's App "Neanderthal+"*, UBIComp/ISWC 2016, Heidelberg, Germany
- [6] [www.technopolis-athens.com/web/guest/museum/home](http://www.technopolis-athens.com/web/guest/museum/home), τελευταία πρόσβαση στις 17/03/2017
- [7] [www.gartner.com/newsroom/id/3415117](http://www.gartner.com/newsroom/id/3415117), τελευταία πρόσβαση στις 19/03/2017
- [8] [apress.jensimmons.com/v5/pro-html5-programming/ch5.html](http://apress.jensimmons.com/v5/pro-html5-programming/ch5.html), τελευταία πρόσβαση στις 19/03/2017
- [9] [www.mobiloud.com/blog/native-web-or-hybrid-apps/](http://www.mobiloud.com/blog/native-web-or-hybrid-apps/), τελευταία πρόσβαση στις 20/03/2017
- [10] [www.w3.org/TR/2014/REC-html5-20141028/](http://www.w3.org/TR/2014/REC-html5-20141028/), τελευταία πρόσβαση στις 20/03/2017
- [11] [www.w3.org/standards/techs/html#w3c\\_all](http://www.w3.org/standards/techs/html#w3c_all), τελευταία πρόσβαση στις 20/03/2017
- [12] [wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://wikipedia.org/wiki/Cascading_Style_Sheets), τελευταία πρόσβαση στις 21/03/2017

- [13] [www.w3.org/Style/CSS/](http://www.w3.org/Style/CSS/), τελευταία πρόσβαση στις 21/03/2017
- [14] [wikipedia.org/wiki/JavaScript](http://wikipedia.org/wiki/JavaScript), τελευταία πρόσβαση στις 22/03/2017
- [15] [wikipedia.org/wiki/JQuery](http://wikipedia.org/wiki/JQuery), τελευταία πρόσβαση στις 22/03/2017
- [16] [wikipedia.org/wiki/Crosswalk\\_Project](http://wikipedia.org/wiki/Crosswalk_Project), τελευταία πρόσβαση στις 23/03/2017
- [17] <https://software.intel.com/en-us/xdk/docs/release-notes-information-intel-xdk>, τελευταία πρόσβαση στις 23/03/2017
- [18] <https://developers.google.com/maps/documentation/javascript/>, τελευταία πρόσβαση στις 23/03/2017
- [19] <https://cordova.apache.org/>, τελευταία πρόσβαση στις 23/03/2017
- [20] <https://github.com/01org/appframework>, τελευταία πρόσβαση στις 24/03/2017
- [21] <https://github.com/ftlabs/fastclick>, τελευταία πρόσβαση στις 24/03/2017
- [22] Cameron, D., (2015) HTML5, JavaScript® and jQuery®, 24-Hour Trainer, John Wiley & Sons, Inc., Indianapolis, Indiana, USA
- [23] Goldstein, A., Lazaris, L., Weyl, E., (2015) HTML5 & CSS3 For The Real World, SitePoint Pty.Ltd., USA
- [24] Brown, T., (2015) CSS Master, SitePoint Pty.Ltd., USA
- [25] Jones, D., (2014) JavaScript: Novice to Ninja, SitePoint Pty.Ltd., USA
- [26] Svennerberg, G., (2010) Beginning Google Maps API 3, Apress®, USA

## Παραρτήματα / Appendixes

### Πίνακας Εικόνων / Table of Figures

Εικόνα 1: Το λογότυπο του Μουσείου.....	8
Εικόνα 2: Μιά κάτοψη των εξωτερικών χώρων του Μουσείου .....	8
Εικόνα 3: Η κεντρική οθόνη της μικρο-εφαρμογής .....	12
Εικόνα 4: Η κεντρική οθόνη και το παράθυρο πληροφόρησης .....	13
Εικόνα 5: Η οθόνη με τη λίστα του Ιστορικού.....	14
Εικόνα 6: Η οθόνη με τη λίστα των Αγαπημένων.....	15
Εικόνα 7: Native vs Web vs Hybrid apps .....	22
Εικόνα 8: Το τρέχον status της HTML5 και των επιμέρους τεχνολογιών της.....	24
Εικόνα 9: HTML5+CSS3+Javascript=η σύγχρονη "γλώσσα" του Διαδικτύου .....	28
Εικόνα 10: HTML5+CSS3+Javascript=τα γρανάζια που κινούν το Διαδίκτυο .....	30
Εικόνα 11: Το Intel XDK IDE, με ανοιχτό το tab DEVELOP για συγγραφή κώδικα.....	33
Εικόνα 12: Το tab SIMULATE ενεργό στο Intel XDK IDE .....	34
Εικόνα 13: Η αρχιτεκτονική μιας εφαρμογής βασισμένης στο Cordova framework..	36
Εικόνα 14: Η οθόνη εκκίνησης του app .....	54
Εικόνα 15: Εύρεση τοποθεσίας του χρήστη.....	54
Εικόνα 16: Μετακίνηση του χρήστη .....	55
Εικόνα 17: Προσέγγιση του χρήστη σε σημείο ενδιαφέροντος .....	55
Εικόνα 18: Παράθυρο InfoWindow.....	56
Εικόνα 19: Ερώτηση για αποθήκευση στα Αγαπημένα .....	56
Εικόνα 20: Η οθόνη με τη λίστα του Ιστορικού.....	57
Εικόνα 21: Η οθόνη των Αγαπημένων .....	57
Εικόνα 22: Η οθόνη της Βοήθειας .....	57