ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΗΠΕΙΡΟΥ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Θέμα Πτυχιακής Εργασίας:

«Η υλοποίηση της εφαρμογής Nextktel σε iOS.»

Επιβλέπων καθηγητής: Χρυσόστομος Στύλιος

Ονοματεπώνυμο Φοιτήτριας: Γεωργία Μπελεσιώτη

Αρ. Μητρώου: I42

Ιούνιος 2014

ΠΕΡΙΕΧΟΜΕΝΑ

1. Εισαγωγή

Η παρούσα Πτυχιακή Εργασία σκοπό έχει τη σχεδίαση και την υλοποίηση της εφαρμογής Nextktel. Η εν λόγω εφαρμογή θα παρέχει πληροφορίες για την ώρα αναχώρησης, τους ενδιάμεσους σταθμούς και τις αλλαγές των λεωφορείων που πρέπει να πραγματοποιήσει ο χρήστης για να φτάσει εγκαίρως στον προορισμό του. Η εφαρμογή υλοποιείται σε λειτουργικό iOS 7 και τρέχει σε συσκευές iPhone, iPod και iPad.

Όσον αφορά στην δομή της παρούσας εργασίας, αρχικά αποδίδονται οι αρετές μιας εφαρμογής. Στη συνέχεια, εξετάζονται οι διακειμενικές σχέσεις ανάμεσα στο iPhone και το Android. Στο επόμενο κεφάλαιο, εξετάζονται η ιστορία του λογότυπου της Apple και η ιστορία του Iphone. Ακόμη, γίνεται νύξη στο iOS λειτουργικό, συμπεριλαμβανομένης της νέας έκδοσης iOS 8. Επίσης, γίνεται σύντομη αναφορά στη λειτουργία των Navigation controllers και αποδίδεται ο όρος Delegation. Επιπροσθέτως, αναλύεται η Αρχιτεκτονική Model-View-Controller (MVC) και παρατίθενται πληροφορίες για τη Γλώσσα Προγραμματισμού Objective-C. Έπειτα, εξετάζεται το σύστημα Client-Server Computing. Επιπλέον, αναλύεται η λειτουργία του αλγόριθμου αναζήτησης κατά βάθος (depth- first-search ή dfs) και παρουσιάζονται κάποια παραδείγματα για την καλύτερη κατανόηση του εν λόγω αλγόριθμου. Στη συνέχεια, σκιαγραφούνται η φιλοσοφία και η χρησιμότητα της εφαρμογής Nextktel. Παράλληλα, εξετάζεται η υλοποίηση του αλγόριθμου αναζήτησης κατά βάθος στην εν λόγω εφαρμογή και αναλύονται ακροθιγώς κάποια καίρια σημεία του κώδικα.

2. Οι αρετές μιας εφαρμογής

Σύμφωνα με τον οδηγό «HIG (Human Interface Guide)» της Apple, τα τρία είδη εφαρμογών που υλοποιούνται ως επί το πλείστον σε iOS κατηγοριοποιούνται ως εξής:

- **«Εφαρμογές Εξομοίωσης ή Εικονικής πραγματικότητας (Immersive)»**

Πρόκειται για εφαρμογές εικονικής πραγματικότητας, οι οποίες όπως υποδηλώνει η ονομασία τους «εξομοιώνουν πραγματικές ή φανταστικές συνθήκες». Ενδεικτικά, αναφέρονται τα παιχνίδια και τα εικονικά εργαλεία.

Εικόνα 1 - Εικονική πυξίδα, Πηγή Φωτογραφίας: Αναστάσιος, Πιοτόπουλος, «Δημιουργία mobile application σε iOS (iPhone/iPad) για την οργάνωση οικιακών εξόδων, ακολουθώντας πρακτικά πρότυπα προγραμματισμού», Πτυχιακή Εργασία, Τμήμα Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών, Λάρισα 2011, σ. 29 παρατίθεται στο engadget.

- **«Εφαρμογές Παραγωγικότητας (Productivity)»**

Οι εφαρμογές αυτές έχουν δημιουργηθεί για να εξυπηρετούν κάποιο ιδιαίτερο σκοπό και συνήθως χρησιμοποιούνται για την διαχείριση πληροφοριών. Οι προσφερόμενες πληροφορίες διατάσσονται με ιεραρχία και προβάλλονται σε διαφορετικά views με την παραγωγική μέθοδο (από το γενικό στο ειδικό). Κατόπιν τούτου, «όσο περισσότερο εισχωρούμε στα επίπεδα τόσο πιο συγκεκριμένες πληροφορίες παρουσιάζονται στον χρήστη». Η χρήση των γραφικών γίνεται χωρίς υπερβολές.

Εικόνα 2 - "Settings", Εφαρμογή Παραγωγικότητας. Πηγή Φωτογραφίας: Αναστάσιος, Πιοτόπουλος, «Δημιουργία mobile application σε iOS (iPhone/iPad) για την οργάνωση οικιακών εξόδων, ακολουθώντας πρακτικά πρότυπα προγραμματισμού», Πτυχιακή Εργασία, Τμήμα Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών, Λάρισα 2011, σ. 30.

- **«Εφαρμογές Χρησιμότητας (Utility)»**

Πρόκειται για εφαρμογές με συγκεκριμένες λειτουργίες, οι οποίες διαχειρίζονται μικρό όγκο δεδομένων. Η περίοδος χρήσης τους διαρκεί για δευτερόλεπτα και είναι ο πιο συνηθισμένος τύπος εφαρμογών. Για παράδειγμα σ' αυτή την κατηγορία εμπίπτει η εφαρμογή «μετατροπέας μονάδων μέτρησης».[1]

Εικόνα 3 - Παράδειγμα Εφαρμογής Χρησιμότητας μετατροπέας μονάδων μέτρησης. Πηγή Φωτογραφίας: Βλ. Αναστάσιος Πιοτόπουλος, ό.π., σ. 31 παρατίθεται στο solsie.com

[1]Αναστάσιος, Πιοτόπουλος, «Δημιουργία mobile application σε iOS (iPhone/iPad) για την οργάνωση οικιακών εξόδων, ακολουθώντας πρακτικά πρότυπα προγραμματισμού», Πτυχιακή Εργασία, Τμήμα Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών, Λάρισα 2011, σ. 29-31.

Στο σημείο αυτό, αναφέρονται ορισμένα ιδιαίτερα χαρακτηριστικά που πρέπει να συγκεντρώνει μια iOS εφαρμογή για να θεωρείται «πετυχημένη»:

- «**Η σχεδίαση (design)»**

Θα πρέπει να σχεδιάσουμε μια λιτή και «καλά οργανωμένη διεπαφή χρήστη», ώστε η εφαρμογή να είναι «αποδοτική και ευχάριστη».

- «**Η διαχείριση μνήμης (memory management)»** [2]

Κάθε πρόγραμμα απαιτεί ορισμένο χρόνο επεξεργασίας και μνήμη για να επεξεργασθεί τις εντολές και τα δεδομένα. Δεδομένου ότι η μνήμη αποτελεί έναν από τους πιο πολύτιμους πόρους του υπολογιστή, ένα πρόγραμμα πρέπει να δημιουργηθεί κατά τέτοιο τρόπο ώστε να χρησιμοποιεί τη μικρότερη ποσότητα μνήμης και το λιγότερο χρόνο επεξεργαστικής ισχύος.[3]

Επιπροσθέτως, όταν δημιουργούμε μια εφαρμογή πρέπει να προσέχουμε να μην υπάρχουν διαρροές μνήμης, να απελευθερώνονται όσο νωρίτερα γίνεται οι πόροι και να τίθενται όρια στους πόρους και τις προβολές. Παραδείγματος χάρη, ορίζουμε «στο mail client να φορτώνονται μόνο 2 γραμμές από κάθε μήνυμα (lazy loading) και μόνο τα 10 πιο πρόσφατα μηνύματα (όρια)».

Επίσης, πρέπει να ελέγχουμε τις προειδοποιήσεις που στέλνει το iOS προς την εφαρμογή όταν η μνήμη της συσκευής είναι σε οριακό σημείο. Όταν γίνουν δεκτές τέτοιες ειδοποιήσεις πρέπει να προβούμε σε κάποια από τις παρακάτω ενέργειες:

---

[2]Αναστάσιος, Πιοτόπουλος, «Δημιουργία mobile application σε iOS (iPhone/iPad) για την οργάνωση οικιακών εξόδων, ακολουθώντας πρακτικά πρότυπα προγραμματισμού», Πτυχιακή Εργασία, Τμήμα Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών, Λάρισα 2011, σ. 28, 31.

[3]ITL Education Solutions Limited (ITL ESL), *Introduction to Computer Science,* Pearson Education, second ed., India 2011. Βλ. http://my.safaribooksonline.com/book/electrical-engineering/computer-engineering/9788131760307/computer-program/section_8.10

α) «Στο UIApplication delegate, υλοποιώντας τη μέθοδο "applicationDidReceiveMemoryWarning"».

β) «Στους view controllers, υλοποιώντας τη μέθοδο "didReceiveMemoryWarning"».

γ) «Κάνοντας εγγραφή στην κοινοποίηση "UIApplicationDidReceivwMemoryWarning"».

- «**Ανταπόκριση (responsiveness)» ,[4] «σταθερότητα και αξιοπιστία»**

Μια εφαρμογή θα πρέπει να δοκιμαστεί στις πιο ακραίες συνθήκες, ειδάλλως ενδέχεται να παρουσιάσει προβλήματα στους χρήστες. Για παράδειγμα η εφαρμογή θα πρέπει να έχει δοκιμαστεί σε περιπτώσεις απενεργοποίησης WiFi, το τηλέφωνο να ρυθμιστεί σε λειτουργία αεροπλάνου κλπ.[5]

Επίσης, η εφαρμογή θα πρέπει να ξεκινά γρήγορα και να ανταποκρίνεται άμεσα στις εισόδους του χρήστη. Παράλληλα, η εφαρμογή θα πρέπει να αποθηκεύει τα δεδομένα σε οποιοδήποτε στάδιο κι αν βρίσκεται. Παραδείγματος χάρη, όταν ο χρήστης λαμβάνει κάποια κλήση τη στιγμή που παίζει ένα παιχνίδι, θα πρέπει να διακόπτεται αυτόματα το παιχνίδι και θα μπορεί να επανέλθει στο ίδιο στάδιο του παιχνιδιού μετά την ολοκλήρωση της κλήσης.

- «**Κατανάλωση ενέργειας (power consumption)»**

Η ανταλλαγή δεδομένων στο δίκτυο είναι μια διαδικασία που απαιτεί μεγάλη κατανάλωση ενέργειας. Όταν δεν υπάρχει διαρκής επικοινωνία της συσκευής με το δίκτυο εξοικονομείται ενέργεια. Επιπλέον, η συνετή χρήση των Location Services συμβάλλει στην εξοικονόμηση ενέργειας.

---

[4]Αναστάσιος, Πιοτόπουλος, «Δημιουργία mobile application σε iOS (iPhone/iPad) για την οργάνωση οικιακών εξόδων, ακολουθώντας πρακτικά πρότυπα προγραμματισμού», Πτυχιακή Εργασία, Τμήμα Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών, Λάρισα 2011, σ. 32.

[5]"What makes an app a good app - 7 pointers", (14/4/2014).

Βλ.http://dotnet.dzone.com/articles/what-makes-app-good-app-10

- **Ασφάλεια**

Η εκάστοτε εφαρμογή iOS διαθέτει ξεχωριστό sandbox για να είναι ασφαλής η συσκευή και το δίκτυο κινητής τηλεφωνίας. Για την προστασία του κώδικα της εφαρμογής από τους επίδοξους εισβολείς χρησιμοποιούνται οι παρακάτω συνήθεις πρακτικές: «χρήση keychain, πιστοποιητικών ασφαλείας, υπηρεσιών εμπιστευτικότητας (trust), υπηρεσιών σύνταξης κρυπτογραφικών μηνυμάτων (cryptographic message syntax services)» κλπ.

Συνοψίζοντας, για να δημιουργήσουμε μια εφαρμογή στο iOS πρέπει να λάβουμε υπόψιν τα εξής:

- ελκυστική σχεδίαση
- σωστή διαχείριση μνήμης
- ανταπόκριση (responsiveness), σταθερότητα και αξιοπιστία
- εξοικονόμηση ενέργειας
- ασφάλεια.[6]

[6]Αναστάσιος, Πιοτόπουλος, «Δημιουργία mobile application σε iOS (iPhone/iPad) για την οργάνωση οικιακών εξόδων, ακολουθώντας πρακτικά πρότυπα προγραμματισμού», Πτυχιακή Εργασία, Τμήμα Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών, Λάρισα 2011, σ. 32-34.

3. iPhone εναντίον Android

Εικόνα 4- Iphone εναντίον Android.
Πηγή http://www.techgear.gr/android-vs-ios-software-support-81383/

Στη σύγχρονη τεχνοκρατική εποχή σημειώνεται έντονος ανταγωνισμός στην αγορά των smartphones ανάμεσα στις δύο κορυφαίες εταιρείες κινητής τηλεφωνίας, iPhone και Android.

Ένα βασικό πλεονεκτήματα του Android είναι πως «η ανάπτυξη εφαρμογών γίνεται σχεδόν αποκλειστικά στη Java, ίσως την πιο δημοφιλή, πιο ολοκληρωμένη και καλύτερα δομημένη γλώσσα που υπάρχει σήμερα».

Επιπροσθέτως, η δημιουργία Android εφαρμογών δεν απαιτεί πολλά έξοδα, αφού ο προγραμματιστής χρησιμοποιεί τον προσωπικό του υπολογιστή με οποιοδήποτε λειτουργικό σύστημα, ένα κινητό Android για να ελέγξει τις εφαρμογές και έναν λογαριασμό developer στο Android Market, για τον οποίο χρειάζεται η εφάπαξ καταβολή ενός ισχνού αντίτιμου (περίπου 25$).

Όσον αφορά στο iPhone, η ανάπτυξη των εφαρμογών γίνεται σε Objective C, η οποία αποτελεί υπερσύνολο της C. Αναντίρρητα, το κόστος εξοπλισμού για την δημιουργία μιας iPhone εφαρμογής είναι σαφώς μεγαλύτερο σε σύγκριση με το κόστος μιας Android εφαρμογής, αφού ο προγραμματιστής θα πρέπει να είναι κάτοχος Mac υπολογιστή, iPhone κινητού για τον έλεγχο των εφαρμογών του και να έχει δημιουργήσει έναν λογαριασμό developer, για τον οποίο θα πληρώνει 100$ κάθε χρόνο. Επ' αυτού, πρέπει να σημειωθεί ότι μια συσκευή iPhone είναι πολύ ακριβότερη σε σχέση με μια συσκευή Android.

Εντούτοις, έχει δημιουργηθεί η εντύπωση ότι οι χρήστες iPhone είναι περισσότερο πρόθυμοι να ξοδέψουν χρήματα για την αγορά μιας iOS εφαρμογής σε σχέση με τους χρήστες Android, επειδή το iPhone αποτελεί ένα «status symbol», μια συσκευή που αγοράζουν πολλοί με απώτερο σκοπό τον «εντυπωσιασμό». Επομένως, η στάση αυτή οφελεί τους iOS developers. Από την άλλη πλευρά, το Android βασίζεται στο Linux και εμφανίζει έναν καθαρά «open source χαρακτήρα». Μεγάλη μερίδα των χρηστών του Android διέπεται από αυτήν την φιλοσοφία, ως εκ τούτου θα ξόδευε χρήματα για μια εφαρμογή «μόνο αν είναι εγγυημένα χρήσιμη και προσεγμένη».

Εκτός αυτού, οι iOS developers έχουν το μεγάλο πλεονέκτημα ότι το iPhone ακολουθεί ένα συγκεκριμένο πρότυπο αρχιτεκτονικής και εργαλείων σε όλα τα μοντέλα του, αφού έχει «μία οθόνη συγκεκριμένων διαστάσεων, ένα δεδομένο interface διάδρασης με τον χρήστη και συγκεκριμένο σετ από αισθητήρες». Αφετέρου, το Android συγκεντρώνει πλήθος ετερόκλητων χαρακτηριστικών,

«όπως οθόνες διαφορετικών διαστάσεων και αναλύσεων, διαφορετικά σετ αισθητήρων, διαφορετικά interfaces διάδρασης κ.α.». Εύλογα, είναι απαραίτητο οι Android developers να γνωρίζουν τα παραπάνω στοιχεία προκειμένου να αναπτύξουν εφαρμογές, οι οποίες δεν θα παρουσιάζουν προβλήματα και θα τρέχουν με τον ίδιο τρόπο σε όλες τις συσκευές Android.[7]

4. Το λογότυπο της Apple

Το πρώτο λογότυπο της Apple σχεδιάστηκε το 1976 από τον Ron Wayne. Ο Ron Wayne δημιούργησε ένα βικτοριανό λογότυπο που απεικόνιζε τον Newton να βρίσκεται κάτω από ένα δέντρο. Το εν λόγω λογότυπο συνοδεύεται από την παρακάτω φράση του Wordsworth: «“A mind forever voyaging through strange seas of thought, alone”» (= «“Ενας νους που ταξιδεύει αιώνια στις παράξενες θάλασσες της σκέψης, μόνος”»).

Εικόνα 5- Το πρώτο λογότυπο της Apple. Πηγή: http://freddesign.co.uk/so-why-an-apple-the-history-of-the-apple-logo/

Αξιοσημείωτο είναι ότι το 1977 κυκλοφόρησε ο Apple II με το νέο πολύχρωμο λογότυπο. Ο art director Rob Janoff επιμελήθηκε τη σχεδίαση του νέου λογότυπου της Apple. Ο Steve Jobs τον είχε παροτρύνει να μην το κάνει χαριτωμένο. Ο Janoff δημιούργησε ένα απλό σχήμα μήλου σε δύο εκδοχές: ένα μήλο ολόκληρο και ένα μήλο με μια δαγκωματιά. Ο Jobs επέλεξε το σχήμα με την δαγκωματιά.

Εικόνα 6- Το δεύτερο λογότυπο της Apple. Πηγή:http://creativebits.org/interview/interview_rob_janoff_designer_apple_logo

[7]«Εισαγωγή στην Ανάπτυξη Android Εφαρμογών» του Εργαστηρίου Τεχνολογίας Πολυμέσων του Ε.Μ.Π., Digital Academy, σ. 5, 6. Βλ.http://www.dga.gr/web/publications/files/android.pdf

Εικόνα 7- Apple II,
Πηγή Φωτογραφίας: http://creativebits.org/interview/interview_rob_janoff_designer_apple_logo

Επίσης, ο Jobs επέλεξε το μήλο να διαθέτει «ραβδώσεις από έξι διαφορετικά χρώματα, με ψυχεδελικές αποχρώσεις κλεισμένες ανάμεσα σε μια λωρίδα πράσινου της φύσης και μια λωρίδα γαλάζιο του ουρανού», παρόλο που η εκτύπωση του εν λόγω λογότυπου ήταν πολύ ακριβή. Στο πάνω μέρος της μπροσούρας ο McKenna σημείωσε το παρακάτω ρητό που συχνά αποδίδεται στον Leonardo da Vinci και επηρέασε την κοσμοθεωρία του Jobs: «"Η απλότητα είναι η υπέρτατη επιτήδευση"».[8]

Στη συνέχεια, το πολύχρωμο μήλο της Apple αντικαταστάθηκε από το αντίστοιχο μονόχρωμο σύμβολο αποκτώντας, έτσι, βιομηχανικό και μινιμαλιστικό ύφος.[9]

Εικόνα 8- Το πολύχρωμο σύμβολο της Apple αντικαταστάθηκε από το μονόχρωμο σύμβολο.
Πηγή Φωτογραφίας:http://freddesign.co.uk/so-why-an-apple-the-history-of-the-apple-logo/

[8]Walter, Isaacson, *Steve Jobs του Walter Issacson, Η επίσημη βιογραφία*, Εκδ. Ψυχογιός, 15η εκδ., Αθήνα 2013, σ. 79, 89, 90.

Επίσης, βλ. http://creativebits.org/interview/interview_rob_janoff_designer_apple_logo

Ακόμη, βλ.http://freddesign.co.uk/so-why-an-apple-the-history-of-the-apple-logo/

[9]Βλ.http://freddesign.co.uk/so-why-an-apple-the-history-of-the-apple-logo/

5. Η ιστορία του iPhone

Εικόνα 9- Μοντέλα iPhone 2G, 3G, 3GS, 4, 4S.
Πηγή Φωτογραφίας: http://www.iphoner.gr/άρθρο/

Τον Ιανουάριο του 2007 κάνει την εμφάνισή του το πρώτο iPhone κινητό τηλέφωνο από την εταιρεία Apple Inc. Το νεωτερικό, ιδιαίτερο χαρακτηριστικό που έκανε το iPhone δημοφιλές στο ευρύ κοινό ήταν η τεχνολογία αφής.[10] Η πρώτη γενιά του iPhone είχε βάρος 136 γραμμάρια και παρουσίαζε τα παρακάτω ιδιαίτερα χαρακτηριστικά: «επεξεργαστή 400MHz, 128 MB μνήμης, κάμερα 2 megapixel, οθόνη 3,5 ιντσών με ανάλυση 320 x 480» και τεχνολογία EDGE.[11]

Στη συνέχεια, το iPhone 3G, δεύτερης γενιάς παρείχε την δυνατότητα σύνδεσης σε 3G δίκτυα (2008). Αξίζει να αναφερθεί ότι το 2008 δημιουργήθηκε, επίσης, το AppStore[12] που σήμερα αριθμεί περισσότερες από 900.000 εφαρμογές για κινητά τηλέφωνα.[13]

---

[10]Βλ. http://www.fortunegreece.com/photo-gallery/istoria-tou-iphone/#2

Επίσης, βλ. Αλέξανδρος, Κουτσονάσιος, «Ανάπτυξη και ανάκληση διαδραστικών ξεναγήσεων με φορητές συσκευές», Διπλωματική Εργασία, Τμήμα Ηλεκτρολόγων και Μηχανικών Υπολογιστών, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Θεσσαλονίκη 2010, σ. 12.

[11]Βλ.http://www.fortunegreece.com/photo-gallery/istoria-tou-iphone/#3

[12]Αλέξανδρος, Κουτσονάσιος, «Ανάπτυξη και ανάκληση διαδραστικών ξεναγήσεων με φορητές συσκευές», Διπλωματική Εργασία, Τμήμα Ηλεκτρολόγων και Μηχανικών Υπολογιστών, Αριστοτέλειο Πανεπιστήμιο Θεσσαλλονίκης, Θεσσαλονίκη 2010, σ. 12.

Επίσης, βλ.http://www.fortunegreece.com/photo-gallery/istoria-tou-iphone/#4

[13]Βλ. http://www.fortunegreece.com/photo-gallery/istoria-tou-iphone/#4

Επίσης, βλ. http://www.apple.com/gr/iphone-5s/features/

Το 2009 κυκλοφόρησε το iPhone 3GS με βελτιωμένες δυνατότητες και πρόσθετα χαρακηριστικά, όπως πυξίδα, ταχύτερο επεξεργαστή, διπλή μνήμη και κάμερα 3.2 megapixel.[14]

Εικόνα 10 - iPhone 4.
Πηγή Φωτογραφίας: http://www.fortunegreece.com/photo-gallery/istoria-tou-iphone/#6

Το 2010 κυκλοφόρησε το iPhone 4 με «οθόνη Retina ανάλυσης 960x640, ταχύτερο επεξεργαστή 800 MHz A4, κάμερα 5 megapixel και διπλάσια μνήμη». «"Αν αυτό που θέλετε είναι μέγεθος και σχήμα, όμορφο σχεδιασμό και μεγαλύτερη διάρκεια ζωής για τη μπαταρία, ποιότητα κατασκευής και ευχαρίστηση, τότε το iPhone 4 είναι για σας"», αναφέρει χαρακτηριστικά ο αρθρογράφος των *New York Times* Ντέιβιντ Πογκ.[15]

Το 2011 κυκλοφόρησε το iPhone 4S «με διπλοπύρηνο επεξεργαστή A5, κάμερα 8 megapixel και δυνατότητα εγγραφής βίντεο 1080p (δηλαδή HD)».[16]

Το 2012 παρουσιάστηκε το iPhone 5 με οθόνη 4 ιντσών το πλαίσιο της οποίας ήταν κατασκευασμένο από αλουμίνιο και γυαλί. Επιπλέον, το iPhone 5 διέθετε «νέο επεξερεγαστή A6, 1GB RAM, μια βελτιωμένη εμπρόσθια κάμερα 1,3 megapixel και μια τροποποιημένη κάμερα 8 megapixel».[17]

---

[14]Αλέξανδρος, Κουτσονάσιος, «Ανάπτυξη και ανάκληση διαδραστικών ξεναγήσεων με φορητές συσκευές», Διπλωματική Εργασία, Τμήμα Ηλεκτρολόγων και Μηχανικών Υπολογιστών, Αριστοτέλειο Πανεπιστήμιο Θεσσαλλονίκης, Θεσσαλονίκη 2010, σ. 12.

Επίσης, βλ. http://www.fortunegreece.com/photo-gallery/istoria-tou-iphone/#5

[15]Βλ. http://www.fortunegreece.com/photo-gallery/istoria-tou-iphone/#7

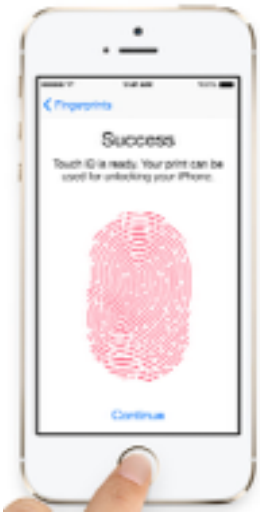
[16]Βλ. http://www.fortunegreece.com/photo-gallery/istoria-tou-iphone/#8

[17]Βλ. http://www.fortunegreece.com/photo-gallery/istoria-tou-iphone/#9

Τεχνικές προδιαγραφές iPhone 5s

Το iPhone 5S διαθέτει επεξεργαστή A7 αρχιτεκτονικής 64 bit, ο οποίος προσδίδει την διπλάσια ταχύτητα στη CPU σε σχέση με τον επεξεργαστή A6. Αξίζει να αναφερθεί ότι το iPhone 5s είναι «το πρώτο smartphone με αρχιτεκτονική 64 bit στον κόσμο». Επιπλέον, διαθέτει συνεπεξεργαστή M7, οποίος υπολογίζει τα δεδομένα κίνησης και καθιστά αποδοτικότερη τη λειτουργία της συσκευής. Ακόμη, το iPhone 5S προσφέρει εντυπωσιακά γραφικά τα οποία χαρακτηρίζονται από μεγάλη ευκρίνεια και παρέχει την δυνατότητα βιντεοκλήσεων FaceTime.

Εικόνα 11- Ρεαλιστικά, εντυπωσιακά εφέ του iPhone 5S. Πηγή Φωτογραφίας: http://www.apple.com/gr/iphone-5s/features/



Επιπλέον, το iPhone 5S διαθέτει λειτουργικό σύστημα iOS 7 και αισθητήρα αναγνώρισης δακτυλικών αποτυπωμάτων. Εντύπωση, επίσης, προκαλούν το κουμπί αρχικής Οθόνης από κρύσταλλο ζαφειριού και το Touch ID. Ο νέος αισθητήρας αναγνώρισης δακτυλικών αποτυπωμάτων, Touch ID προσφέρει στον χρήστη την δυνατότητα να ξεκλειδώνει με το δακτυλικό του αποτύπωμα το κινητό.[18]

Εικόνα 12- Απεικονίζεται το κουμπί αρχικής Οθόνης από κρύσταλλο ζαφειριού και το Touch ID του iPhone 5S.
Πηγή Φωτογραφίας: http://www.apple.com/gr/iphone-5s/features/

[18]Βλ. http://www.apple.com/gr/iphone-5s/features/

Τεχνικές προδιαγραφές iPhone 5c

Εικόνα 13: iPhone 5c. Πηγή http://www.apple.com/gr/iphone-5c/features/

Το iPhone 5c διαθέτει επεξεργαστή A6 chip, ο οποίος προσφέρει μεγαλύτερη εξοικονόμηση ενέργειας. Επίσης, η κάμερα iSight 8MP προφέρει την δυνατότητα λήψης φωτογραφιών υψηλής ανάλυσης.

Ακόμη, η λήψη βίντεο HD σε ανάλυση 1080p και η νέα λειτουργία zoom 3x δίνει την δυνατότητα στον χρήστη να έρθει πιο κοντά στο θέμα λήψης. Επιπροσθέτως, το iPhone 5c προσφέρει την δυνατότητα λήψης φωτογραφιών κατά την εγγραφή βίντεο και την δυνατότητα ανίχνευσης μέχρι και 10 προσώπων. Παράλληλα, το iPhone 5c έχει κατασκευαστεί με νέα κάμερα FaceTime HD, η οποία προσφέρει μεγαλύτερα pixel, καλύτερο αισθητήρα οπίσθιου φωτισμού και ποιοτικότερες κλήσεις FaceTime.[19]

Εικόνα 14 - iOS, Πηγή: http://en.wikipedia.org/wiki/IPhone_OS#iPhone_SDK

6. iOS λειτουργικό

Το iOS είναι ένα λειτουργικό σύστημα που αναπτύχθηκε από την Apple Inc. Το iOS έκανε την εμφάνισή του το 2007 υποστηρίζοντας αρχικά μόνο το iPhone, ωστόσο πλέον έχει αναβαθμιστεί και υποστηρίζει και άλλες συσκευές της Apple, όπως το iPod Touch (Σεπτέμβριος 2007), iPad (Ιανουάριος 2010), το iPad Mini (Νοέμβριος 2012) και την δεύτερης γενιάς Apple TV (Σεπτέμβριος 2010).[20]

[19]Βλ.http://www.apple.com/gr/iphone-5c/features/

[20]Βλ.http://en.wikipedia.org/wiki/IPhone_OS#iPhone_SDK

Εικόνα 15 - Τα επίπεδα του iOS SDK. Πηγή: Αναστάσιος, Πιοτόπουλος, ό.π., σ. 10 παρατίθεται στην Πηγή:Apple

Το SDK (Software Development Kit) του iOS αποτελείται από τέσσερα επίπεδα - στρώματα: Cocoa Touch, Media, Core Services, Core OS. Τα χαμηλά επίπεδα του iOS προσφέρουν τις βασικές υπηρεσίες στις οποίες στηρίζονται όλες οι εφαρμογές, «ενώ τα υψηλότερα στρώματα περιέχουν υπηρεσίες και τεχνολογίες σχετικές με εξελιγμένα γραφικά και διεπαφές χρήστη».[21]

Εικόνα 16- WWDC 2014, Πηγή Φωτογραφίας: http://www.e-pcmag.gr/news/neo-mac-os-x-yosemite-kai-oi-upiresies-icloud-drive-maildrop-kai-airdrop

6.1 Το νέο iOS 8

Το WWDC συνέδριο διοργανώνεται κάθε χρόνο από την Apple με σκοπό να γνωστοποιούνται οι νέες εκδόσεις λειτουργικού και οι νέες τεχνολογίες της Apple.

Στο φετινό WWDC συνέδριο της Apple που πραγματοποιήθηκε τον Ιούνιο του 2014 παρουσιάστηκε το νέο iOS 8 για iPhone, iPad και iPod Touch και ποικίλες νέες υπηρεσίες. Επίσης, η Apple ανακοίνωσε τη νέα έκδοση του λειτουργικού για desktop υπολογιστές, Mac OS X 10.10 Yosemite και τη νέα γλώσσα προγραμματισμού Swift. Αξιοσημείωτο είναι ότι το λειτουργικό Mac OS X 10.10 Yosemite θα επιτρέπει στους χρήστες να πραγματοποιούν τηλεφωνικές κλήσεις και να δέχονται sms μηνύματα.

Εικόνα 17- Mac OS X 10.10 Yosemite με μοντέρνα εικονίδια και flat σχεδίαση. Πηγή http://www.e-pcmag.gr/news/neo-mac-os-x-yosemite-kai-oi-upiresies-icloud-drive-maildrop-kai-airdrop

[21]Αναστάσιος, Πιοτόπουλος, «Δημιουργία mobile application σε iOS (iPhone/iPad) για την οργάνωση οικιακών εξόδων, ακολουθώντας πρακτικά πρότυπα προγραμματισμού», Πτυχιακή Εργασία, Τμήμα Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών, Λάρισα 2011, σ.10.

Στο σημείο αυτό, παρατίθενται ορισμένα νέα χαρακτηριστικά του iOS 8:

Αναβαθμισμένη έκδοση του Safari

Επίσης, περιλαμβάνει μία αναβαθμισμένη έκδοση του Safari, η οποία όπως υποστηρίζει η Apple τρέχει ταχύτερα από τους browser που κυκλοφορούν στην αγορά.

MailDrop

Επιπροσθέτως, η υπηρεσία MailDrop προσφέρει την δυνατότητα επισύναψης αρχείων μεγάλου μεγέθους και την αποστολής τους με e-mail.

HealthKit

Η εν λόγω εφαρμογή σχετίζεται με την παρακολούθηση της υγείας και της φυσικής κατάστασης του χρήστη.

QuickType

Το νέο πληκτρολόγιο με τη λειτουργία QuickType θα προβλέπει τη λέξη που επιθυμεί να γράψει ο χρήστης.

Hey Siri

Το Hey Siri είναι παρόμοιο του Google Now και χρησιμεύει «για φωνητική ενεργοποίηση της Siri». Επιπλέον, δίνοντας φωνητικές εντολές ο χρήστης θα μπορεί να επιλέξει τραγούδια από το iTunes κλπ.

Family Sharing

Το νέο iOS προσφέρει την δυνατότητα σε οικογένειες μέχρι 6 ατόμων να ανταλλάσσουν δεδομένα, όπως φωτογραφίες, υπενθυμίσεις, καταχωρίσεις στην ατζέντα και εφαρμογές για αγορές χωρίς να απαιτείται το κάθε μέλος της οικογένειας να έχει δημιουργήσει ξεχωριστό λογαριασμό.

Spotlight

«Το Spotlight είναι εξελιγμένη αναζήτηση, όπου μέσω άλλων εφαρμογών και υπηρεσιών, θα προτείνει στο χρήστη κινήσεις όπως να παρακολουθήσει μία ταινία, να ακούσει ένα τραγούδι από το iTunes».

iMessage - Group Messaging

Το iMessage χρησιμοποιείται ως επί το πλείστον από τους περισσότερους χρήστες του iPhone. Με τη λειτουργία Tap to Talk, ο χρήστης έχει την δυνατότητα να στείλει ηχητικά ή video μηνύματα.[22]

HomeKit

Όσον αφορά στα νέα χαρακτηριστικά του iOS 8, το HomeKit είναι ένα λογισμικό που θα προσφέρει στον χρήστη την δυνατότητα να ελέγχει τις «"έξυπνες"» οικιακές συσκευές, παραδείγματος χάρη την κλειδαριά, την πόρτα του γκαράζ, τον φωτισμό, τις κάμερες ασφαλείας κλπ.[23]

«Το iOS 8 θα μπορεί να εγκατασταθεί στις παρακάτω συσκευές: iPhone 4s, iPhone 5, iPhone 5c, iPhone 5s, iPod touch 5ης γενιάς, iPad 2, iPad με Retina display, iPad Air, iPad mini και iPad mini με Retina display».[24]

---

[22]http://www.e-pcmag.gr/news/neo-mac-os-x-yosemite-kai-oi-upiresies-icloud-drive-maildrop-kai-airdrop

Επίσης, βλ.http://www.e-pcmag.gr/news/i-apple-parousiase-neo-ios-8-gia-iphone-ipad-kai-ipod-touch

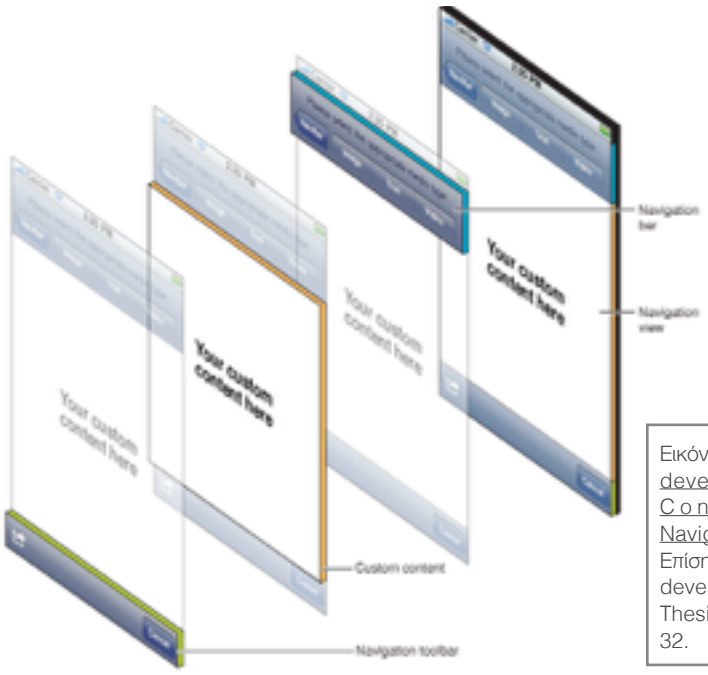
[23]Κώστας, Δεληγιάννης, «Το "έξυπνο" σπίτι λανσάρισε η Apple», *Καθημερινή* (03/06/2014). Βλ. http://www.kathimerini.gr/770094/article/texnologia/thlefwnia/to-e3ypno-spiti-lansarise-h-apple

[24]http://www.e-pcmag.gr/news/i-apple-parousiase-neo-ios-8-gia-iphone-ipad-kai-ipod-touch

7. Navigation Controllers

Η κύρια λειτουργία των Navigation Controllers (ελεγκτών πλοήγησης) είναι να διαχειρίζονται την παρουσίαση των δεδομένων στις οθόνες μας. Οι Navigation Controllers «κρατούν» ένα Array, μια στοίβα από View Controllers. Σκοπός των Navigation Controllers είναι να ελέγχουν τη στοίβα και να παρουσιάζουν με ιεραρχία τα views των View Controllers. Επίσης, περιλαμβάνουν μια μπάρα πλοήγησης, η οποία έχει ένα κουμπί για να μπορεί ο χρήστης να πηγαίνει προς τα πίσω Views.[25]

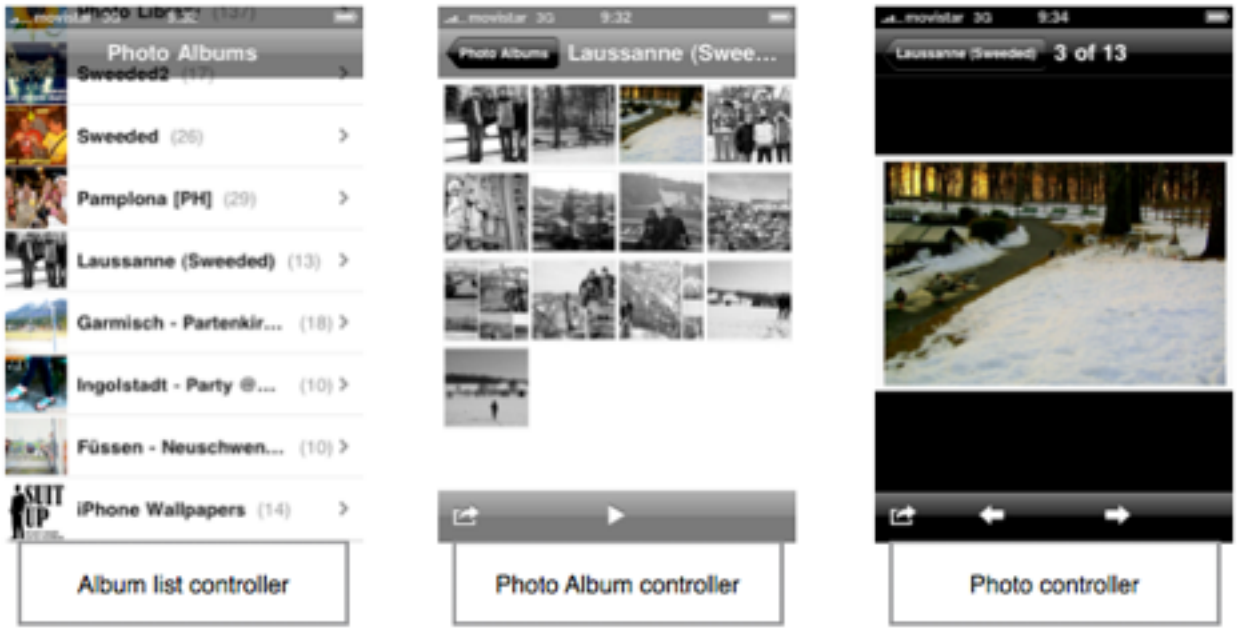
Επισημαίνεται ότι κάθε φορά που μια iOS εφαρμογή εμφανίζει ένα user interface (περιβάλλον εργασίας χρήστη), το περιεχόμενο που εμφανίζεται διοικείται από έναν View Controller ή από μια ομάδα View Controllers σε συντονισμό μεταξύ τους.[26]



Εικόνα 18 - «Τα views του navigation interface» Πηγή:https://developer.apple.com/library/ios/documentation/WindowsViews/Conceptual/ViewControllerCatalog/Chapters/NavigationControllers.html
Επίσης, βλ. Anuar, Lezama, "Introduction to the mobile application development with an example of a PhraseBook app", Master Thesis, Universitat Politècnica de Catalunya, Barcelona 2010, p. 32.

[25]Οι πληροφορίες για το Navigation Controller αντλήθηκαν από το διαδίκτυο και κυρίως από τις ακόλουθες πηγές: Anuar Lezama, "Introduction to the mobile application development with an example of a PhraseBook app", Master Thesis, Universitat Politècnica de Catalunya, Barcelona 2010, p. 32, 33.
Επίσης, βλ. https://developer.apple.com/library/ios/documentation/WindowsViews/Conceptual/ViewControllerCatalog/Chapters/NavigationControllers.html

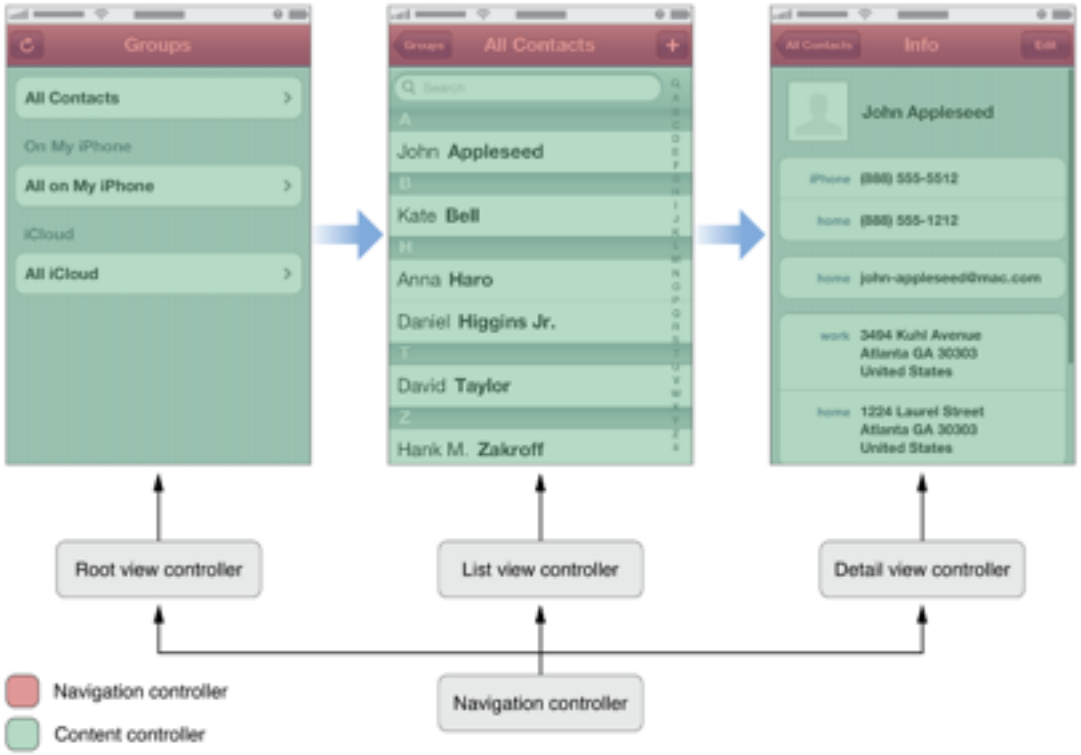
[26]https://developer.apple.com/library/ios/featuredarticles/ViewControllerPGforiPhoneOS/Introduction/Introduction.html



Εικόνα 19: Παράδειγμα λειτουργίας View Controllers
Πηγή: Anuar Lezama, "Introduction to the mobile application development with an example of a PhraseBook app", Master Thesis, Universitat Politècnica de Catalunya, Barcelona 2010, σ. 34, 35.

Η βασική ιδέα για το πως γίνεται η διαχείριση των πληροφοριών σε μια εφαρμογή συνοψίζεται στα εξής: κατ' αρχάς, θα πρέπει να υπάρχει μια controller λίστα με τη main, η οποία θα περιλαμβάνει τις πρώτες πληροφορίες, για παράδειγμα τη λίστα με τα διαθέσιμα άλμπουμ. Στη συνέχεια, εισάγεται μια άλλη επιμέρους controller λίστα με πιο συγκεκριμένες πληροφορίες, για παράδειγμα τις φωτογραφίες μέσα στο άλμπουμ. Τέλος, τη σκυτάλη παίρνει ο Detail Controller, ο οποίος πρόκειται να παρουσιάσει την τελική πληροφορία, για παράδειγμα την φωτογραφία που επιλέχθηκε από το άλμπουμ. Επιπλέον, η μπάρα πλοήγησης μας επιτρέπει να μεταβούμε στο προηγούμενο στάδιο της εφαρμογής.[27]

[27]Anuar, Lezama, "Introduction to the mobile application development with an example of a PhraseBook app", Master Thesis, Universitat Politècnica de Catalunya, Barcelona 2010, p. 34, 35.



Εικόνα 20: Διάταξη των Controllers - Παράδειγμα παρουσίασης μιας εφαρμογής με επαφές, Πηγή:https://developer.apple.com/library/ios/featuredarticles/ViewControllerPGforiPhoneOS/Introduction/Introduction.html

8. Delegation

Είναι γεγονός ότι ξοδεύουμε πολύ χρόνο για την αποστολή μηνυμάτων στα αντικείμενα. Μερικές φορές, όμως, θέλουμε τα αντικείμενα να στέλνουν μηνύματα σε εμάς – «callback». Στο Cocoa Touch, τα callbacks υλοποιούνται με μια τεχνική η οποία είναι γνωστή ως αντιπροσωπεία. Το callback είναι μια λειτουργία η οποία ενεργοποιείται όταν εκδηλώνεται ένα συμβάν, συνήθως όταν ο χρήστης βάζει μια είσοδο. Δεν ξέρουμε ακριβώς πότε θα εκδηλωθεί το συμβάν, ωστόσο έχουμε δημιουργήσει το callback, ώστε όταν συμβεί κάτι, να κληθεί ο κώδικάς μας.[28] Με άλλα λόγια, δημιουργούμε ένα αντικείμενο αλλά δεν ξέρουμε πότε θα συμβεί κάτι σε αυτό. Όταν συμβεί κάποιο γεγονός, με τη λειτουργία callback το εν λόγω αντικείμενο καλεί μια συνάρτηση στο αντικείμενο Delegate με σκοπό να το ενημερώσει ότι κάτι του έχει συμβεί. Το αντικείμενο Delegate

[28]Joe, Conway, Aaron Hillegass, *iPhone Programming: The Big Nerd Ranch Guide,* Big Nerd Ranch, Inc., Atlanta 2010, p. 73, 74.

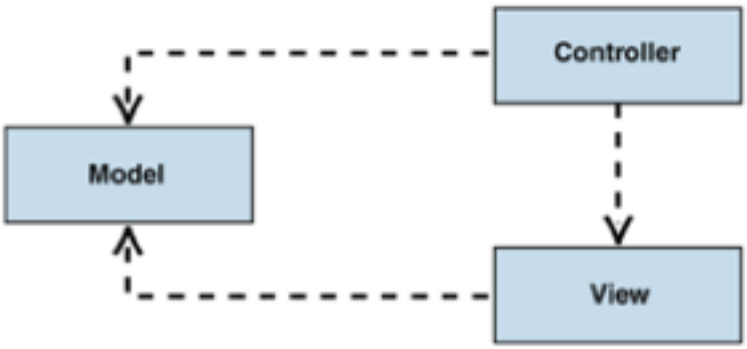
με τη σειρά του υλοποιεί αυτή τη μέθοδο και στη συνέχεια πληροφορεί το αρχικό αντικείμενο ότι η συνάρτηση υλοποιήθηκε.

9. Αρχιτεκτονική Model-View-Controller (MVC)

Η αρχιτεκτονική Model-View-Controller (MVC) χρησιμοποιείται για την διάκριση των λειτουργιών της εφαρμογής και την αποθήκευση των δεδομένων σε τρία ξεχωριστά τμήματα / αντικείμενα.

Το αντικείμενο Controller αναγνωρίζει την είσοδο που δέχεται από τον χρήστη (συνήθως πάτημα ποντικιού ή πληκτρολόγηση) και ειδοποιεί το model ή και το αντικείμενο View για να πραγματοποιήσουν τις απαραίτητες αλλαγές. Συγκεκριμένα, «το View διαχειρίζεται την απεικόνιση των δεδομένων, ενώ το Model "κρατά" τη δομή των δεδομένων, απαντά στα "αιτήματα" για ενημέρωση των δεδομένων που δέχεται από το Controller, ενώ δέχεται και "αιτήματα" για ανάκτηση δεδομένων, συνήθως από το View». Επίσης, επισημαίνεται ότι τα αντικείμενα View και Controller είναι άρρηκτα εξαρτημένα από το Model, ενώ το Model δεν εξαρτάται από τα υπόλοιπα αντικείμενα της MVC αρχιτεκτονικής.[29]

[29]Τζανέτος, Πομόνης, «Προς το Web 3.0: Διαδικασία Ανάπτυξης και Αρχιτεκτονική Υποστήριξης εφαρμογών Παγκόσμιου Ιστού που συνδυάζουν τεχνολογίες Web 2.0 και Semantic Web.», Διδακτορική Διατριβή, Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής, Πανεπιστήμιο Πατρών, Πάτρα 2010, σ. 25, 26.



Εικόνα 21: Αρχιτεκτονική Model-View-Controller (MVC)
Πηγή Φωτογραφίας:Τζανέτος, Πομόνης, «Προς το Web 3.0: Διαδικασία Ανάπτυξης και Αρχιτεκτονική Υποστήριξης εφαρμογών Παγκόσμιου Ιστού που συνδυάζουν τεχνολογίες Web 2.0 και Semantic Web.», Διδακτορική Διατριβή, Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής, Πανεπιστήμιο Πατρών, Πάτρα 2010, σ. 26.

Το παραπάνω σχηματικό απεικονίζει τη ροή ελέγχου μεταξύ των αντικειμένων όταν δοθεί ένα ερέθισμα από τον χρήστη, παραδείγματος χάρη το πάτημα ενός κουμπιού. Είναι έκδηλο ότι τα μοντέλα και τα Views δεν επικοινωνούν μεταξύ τους άμεσα. Ο Controller (ελεγκτής) βρίσκεται ανάμεσα στα δύο αντικείμενα, λαμβάνοντας μηνύματα από κάποια αντικείμενα και στέλνοντας οδηγίες σε άλλα.[30]

Τα αντικείμενα μπορούν να κατανεμηθούν στις εξής κατηγορίες:

Model (μοντέλο)

- Το σύνολο των κλάσεων που κρατούν τα δεδομένα της εφαρμογής.[31]
- Είναι υπεύθυνο για να αποθηκεύει δεδομένα και να τα κάνει διαθέσιμα στα άλλα αντικείμενα.

[30]Aaron, Hillegass, *Objective-C Programming: The Big Nerd Ranch Guide,* Big Nerd Ranch, Inc., Atlanta 2011, p. 182.

[31]Dave Mark, Jack Nutting, Jeff LaMarche, *Beginning iPhone 4 Developement, Exploring the iOS SDK*, Apress, United States of America 2011, p. 34.

Τα NSString, NSDate και NSArray αποτελούν παραδοσιακά μοντέλα αντικειμένων.[32]

View (προβολή)

• «Παρουσιάζει το μοντέλο στο χρήστη σε κατάλληλο περιβάλλον».[33] Με άλλα λόγια, πρόκειται για τα ορατά στοιχεία μιας εφαρμογής.

Σημειώνεται ότι η κλάση UIView και διάφορες άλλες υποκλάσεις της, συμπεριλαμβανομένης της UIWindow, είναι κοινά παραδείγματα View στιγμιοτύπων.[34]

Controller (ελεγκτής)

- Συνδέει το μοντέλο και το View και αποτελεί τη λογική της εφαρμογής, αφού αποφασίζει για το πως θα πρέπει να διαχειριστεί τις εισόδους του χρήστη.[35] Ο Controller είναι χρήσιμος για να συνδέσουμε και να καθοδηγήσουμε τα διάφορα μέρη της εφαρμογής μας.[36]
- Ειδοποιεί το View όταν μεταβάλλονται τα δεδομένα στο μοντέλο. Παράλληλα, ειδοποιεί το μοντέλο όταν ο χρήστης τροποποιεί τα δεδομένα στο View. [37]

---

[32]Aaron, Hillegass, *Objective-C Programming: The Big Nerd Ranch Guide,* Big Nerd Ranch, Inc., Atlanta 2011, p. 182.

[33]Αλέξανδρος, Κουτσονάσιος, «Ανάπτυξη και ανάκληση διαδραστικών ξεναγήσεων με φορητές συσκευές», Διπλωματική Εργασία, Τμήμα Ηλεκτρολόγων και Μηχανικών Υπολογιστών, Αριστοτέλειο Πανεπιστήμιο Θεσσαλλονίκης, Θεσσαλονίκη 2010, σ. 22.

[34]Aaron, Hillegass, *Objective-C Programming: The Big Nerd Ranch Guide,* Big Nerd Ranch, Inc., Atlanta 2011, p. 182.

[35]Dave Mark, Jack Nutting, Jeff LaMarche, *Beginning iPhone 4 Developement, Exploring the iOS SDK*, Apress, United States of America 2011, p. 34.

[36]Aaron, Hillegass, *Objective-C Programming: The Big Nerd Ranch Guide,* Big Nerd Ranch, Inc., Atlanta 2011, p. 182.

[37]Αλέξανδρος, Κουτσονάσιος, «Ανάπτυξη και ανάκληση διαδραστικών ξεναγήσεων με φορητές συσκευές», Διπλωματική Εργασία, Τμήμα Ηλεκτρολόγων και Μηχανικών Υπολογιστών, Αριστοτέλειο Πανεπιστήμιο Θεσσαλλονίκης, Θεσσαλονίκη 2010, σ. 22.

9.1 Πλεονεκτήματα αρχιτεκτονικής MVC

Στο σημείο αυτό αναφέρονται κάποια από τα πλεονεκτήματα και τα μειονεκτήματα της αρχιτεκτονικής MVC:

1) «Σαφήνεια σχεδιασμού»: Ο σαφής διαχωρισμός των αντικειμένων και η κατανομή των εργασιών συμβάλλει στην ευκολότερη υλοποίηση και συντήρηση του προγράμματος.

2) «Αποδοτικός σχεδιασμός» και διευκόλυνση στην διαδικασία της αποσφαλμάτωσης: Οι αλλαγές που συντελούνται σε ένα μέρος του προγράμματος δεν επηρεάζουν τα υπόλοιπα μέρη, καθιστώντας ευκολότερη την αποσφαλμάτωση.

3) «Πολλαπλές όψεις: Η εφαρμογή μπορεί να εμφανίσει την κατάσταση ενός μοντέλου με πολλούς διαφορετικούς τρόπους. Όλα τα Views χρησιμοποιούν τα ίδια ακριβώς δεδομένα αλλά τα παρουσιάζουν με τελείως διαφορετικό τρόπο.»[38]

Παραδείγματος χάρη, πολλές ιστοσελίδες ενδέχεται να φιλοξενούν τα ίδια δεδομένα, χρησιμοποιώντας τα ίδια Model. Παράλληλα, η συσκευή απεικόνισης μιας ιστοσελίδας (π.χ. κινητή συσκευή, υπολογιστής, PDA) ή η επιλογή της μορφής μιας ιστοσελίδας εξαρτώνται από τις προτιμήσεις και τις ανάγκες του εκάστοτε χρήστη. Η αρχιτεκτονική MVC εξυπηρετεί τέτοιου είδους αλλαγές, αφού τα View αλλάζουν χωρίς να επηρεάζεται το Model.[39]

---

[38]Απόστολος, Ανδριώτης, «Σύστημα Ανάγνωσης - Διαχείρισης Ηλεκτρονικών Βιβλίων», Πτυχιακή Εργασία, Αλεξάνδρειο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Θεσσαλονίκης, Σχολή Τεχνολογικών Εφαρμογών Τμήμα Πληροφορικής, Θεσσαλονίκη 2007, σ. 14.

[39]Τζανέτος, Πομόνης, «Προς το Web 3.0: Διαδικασία Ανάπτυξης και Αρχιτεκτονική Υποστήριξης εφαρμογών Παγκόσμιου Ιστού που συνδυάζουν τεχνολογίες Web 2.0 και Semantic Web.», Διδακτορική Διατριβή, Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής, Πανεπιστήμιο Πατρών, Πάτρα 2010, σ. 30.

9.2 Μειονεκτήματα αρχιτεκτονικής MVC

1) Οι προγραμματιστές οφείλουν να γνωρίζουν την διαδικασία ανάπτυξης και λειτουργίας των Views, παρόλο που το View δεν εξαρτάται από το Model. Σε περίπτωση που στο Model πραγματοποιηθούν αλλεπάλληλες αλλαγές, ενδέχεται το View να βομβαρδιστεί από ενημερωτικά αιτήματα. Επειδή ορισμένα View, όπως οι γραφικές απεικονίσεις, είθισται να χρειάζονται αρκετό χρόνο για να φορτωθούν, υπάρχει πιθανότητα να ληφθούν με καθυστέρηση ορισμένα ενημερωτικά αιτήματα από το Model.[40]

10. Γλώσσα προγραμματισμού Objective-C

Η Objective-C είναι ένας συνδυασμός δύο γλωσσών, της C και της Smalltalk.[41] Η γλώσσα προγραμματισμού Objective-C είναι η κύρια γλώσσα προγραμματισμού που χρησιμοποιείται από την Apple για τα λειτουργικά συστήματα OS X και iOS και τα application programming interfaces, Cocoa και Cocoa Touch. Η Objective-C αναπτύχθηκε στις αρχές του 1980 από τους Brad Cox και Tom Love στην εταιρεία Stepstone.[42]

Το 1985 ο Brad Cox πούλησε τη γλώσσα Objective-C στη NeXT Computer, Inc. Η εταιρεία NeXT ιδρύθηκε από τον Steve Jobs, ο οποίος είχε απολυθεί από την Apple Computer το ίδιο έτος. Η εταιρεία NeXT χρησιμοποίησε τη γλώσσα Objective-C για να χτίσει το λειτουργικό σύστημα

---

[40]Τζανέτος, Πομόνης, «Προς το Web 3.0: Διαδικασία Ανάπτυξης και Αρχιτεκτονική Υποστήριξης εφαρμογών Παγκόσμιου Ιστού που συνδυάζουν τεχνολογίες Web 2.0 και Semantic Web.», Διδακτορική Διατριβή, Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής, Πανεπιστήμιο Πατρών, Πάτρα 2010, σ. 30, 31.

[41]Gary Bennett, Mitch Fisher, Bred Lees, *Objective-C for Absolute Beginners iPhone, iPad, and Mac Programming Made Easy*, Apress, sec. ed., New York 2011, p. 103.

[42]Βλ. http://en.wikipedia.org/wiki/Objective-C παρατίθεται στο Garling, Caleb, "iPhone Coding Language Now World's Third Most Popular", (07/09/2012). Βλ.http://www.wired.com/2012/07/apple-objective-c/

NextStep και τα εργαλεία προγραμματισμού της. Όσον αφορά στο software, στην πραγματικότητα η γλώσσα Objective-C έδωσε στη NeXT ένα ανταγωνιστικό πλεονέκτημα. Προγραμματιστές που χρησιμοποιούσαν το λειτουργικό σύστημα NextStep και Objective-C θα μπορούσαν να γράψουν τα προγράμματα γρηγορότερα σε σχέση με όσους χρησιμοποιούσαν την παραδοσιακή γλώσσα C. Η εταιρεία NeXT αγοράστηκε από την Apple Computer στα τέλη του 1996 και η Objective C εξακολουθεί να αποτελεί την κύρια γλώσσα προγραμματισμού που χρησιμοποιείται από την Apple.[43]

Όσον αφορά στη σύνταξη της Objective-C, αξίζει να αναφερθεί ότι το interface και το implementation γράφονται σε διαφορετικά αρχεία. Συγκεκριμένα, τα αρχεία που αντιπροσωπεύουν το interface στην Objective-C έχουν κατάληξη .h και αποτελούν αρχεία κεφαλίδες (header files).[44] Για παράδειγμα, αν η κλάση έχει ονοματιστεί Ball, το interface αρχείο θα ονομαστεί Ball.h και θα περιλαμβάνει την δήλωση των συναρτήσεων της εφαρμογής. Αντίθετα, το implementation έχει επέκταση αρχείου .m (method file)[45] και αντιπροσωπεύει την υλοποίηση των συναρτήσεων της εφαρμογής.

---

[43]Gary Bennett, Mitch Fisher, Bred Lees, *Objective-C for Absolute Beginners iPhone, iPad, and Mac Programming Made Easy*, Apress, second ed., New York 2011, p. 103, 104.

[44]Αλέξανδρος, Κουτσονάσιος, «Ανάπτυξη και ανάκληση διαδραστικών ξεναγήσεων με φορητές συσκευές», Τμήμα Ηλεκτρολόγων και Μηχανικών Υπολογιστών, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Θεσσαλονίκη 2010, σ. 9, 10.

Βλ. http://en.wikipedia.org/wiki/Objective-C

[45]Βλ. http://en.wikipedia.org/wiki/Objective-C

10.1 Παράδειγματα interface και implementation file[46]

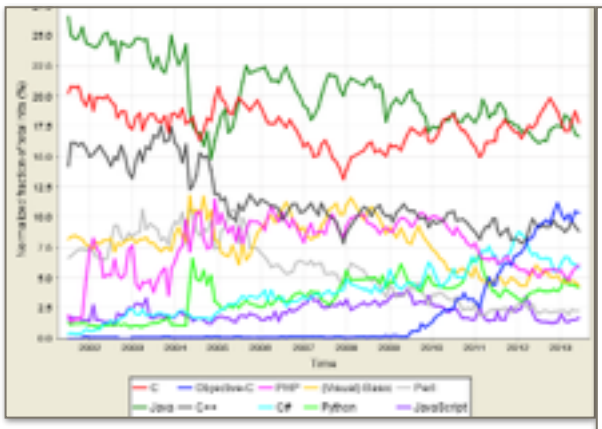
Ακολουθεί το interface file (HelloWorld.h):

```
#import <Foundation/Foundation.h>

@interface HelloWorld : NSObject
{
}

- (void)printGreeting;

@end
```

Ιδού, το implementation file (HelloWorld):

```
#import "HelloWold.h"

@implementation HelloWorld

-(void)printGreeting
{
   NSLog(@"Hello World!");
}

@end
```

[46]Τα παραδείγματα interface και implementation file (HelloWorld) αντλήθηκαν από Gary Bennett, Mitch Fisher, Bred Lees, *Objective-C for Absolute Beginners iPhone, iPad, and Mac Programming Made Easy*, Apress, second ed., New York 2011, p. 107.



Jun 2014	Jun 2013	Change	Programming Language	Ratings	Change
1	1		C	16.191%	-1.62%
2	2		Java	16.113%	-0.54%
3	3		Objective-C	10.934%	+0.58%
4	4		C++	6.420%	-2.39%
5	6	^	C#	3.044%	-1.84%
6	7	^	(Visual) Basic	3.736%	-0.61%
7	5	v	PHP	2.848%	-3.14%
8	8		Python	2.710%	-1.47%
9	10	^	JavaScript	2.000%	+0.36%
10	12	^	Visual Basic .NET	1.914%	+0.80%
11	13	^	Transact-SQL	1.898%	+0.99%
12	9	v	Perl	1.480%	-0.79%
13	11	v	Ruby	1.392%	-0.09%
14	42	^^	ActionScript	1.327%	+1.13%
15	43	^^	F#	1.006%	+0.89%
16	14	v	Lisp	0.843%	-0.04%
17	18	^	Delphi/Object Pascal	0.833%	+0.23%
18	15	v	Pascal	0.808%	+0.03%
19	20	^	MATLAB	0.774%	+0.21%
20	23	^	Assembly	0.771%	+0.27%

Εικόνα 22, 23- Στατιστικά στοιχεία της«TIOBE Index» (2002-2014) σχετικά με την δημοτικότητα της εκάστοτε γλώσσας προγραμματισμού διεθνώς. Πηγή:http://en.wikipedia.org/wiki/TIOBE_index#mediaviewer/File:Tiobe_index.png Επίσης, βλ. http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

Σύμφωνα με την «TIOBE Index»,[47] οι αναζητήσεις για την Objective-C στις μηχανές αναζήτησης το 2012 άγγιξαν το 9.3 % του λογισμικού παγκοσμίως, ενώ για την C++ υπολογίζονται στο 9.1 %. Παράλληλα, μια δεύτερη έρευνα της «Transparent Language Index» τοποθετεί την Objective-C σε 9,2 % και την C + + σε 7,9 %. Ωστόσο, παρόλη την δημοτικότητα των προϊόντων της Apple, το Android αποτελεί το πιο ευρέως χρησιμοποιούμενο λειτουργικό σύστημα κινητής τηλεφωνίας - τουλάχιστον στις ΗΠΑ.[48]

---

[47]Κατά την «TIOBE Programming Community Index», ο αριθμός των επισκέψεων στις μηχανές αναζήτησης καθορίζει τις αξιολογήσεις μιας προγραμματιστικής γλώσσας. Βλ.http://www.tiobe.com/index.php/content/paperinfo/tpci/tpci_definition.htm

[48]Garling, Caleb, "iPhone Coding Language Now World's Third Most Popular", (07/09/2012). Βλ.http://www.wired.com/2012/07/apple-objective-c/

Το 2013, όπως φαίνεται από το παραπάνω σχηματικό, ο δείκτης δημοτικότητας της Objective-C έφτασε σε ποσοστό 11 %. Όσον αφορά στο ποσοστό δημοτικότητας της Java, παρουσίασε σταδιακή ύφεση, σε αντίθεση με την C η οποία εμφάνισε ανοδική τάση.[49]

Το 2014 η Objective-C παραμένει η τρίτη δημοφιλέστερη γλώσσα προγραμματισμού. Ωστόσο, η χρονιά αυτή είναι σημείο - σταθμός για την πορεία της Objective-C, αφού τον Ιούνιο ανακοινώθηκε από την Apple η αντικατάσταση της Objective-C από την γλώσσα Swift. Κατόπιν τούτου, αναμένουμε να δούμε πόσο προσφιλής θα είναι η νέα γλώσσα προγραμματισμού στο ευρύ προγραμματιστικό κοινό.[50]

11. Σύστημα Client-Server Computing

Ορισμός Client-Server Computing

«Το σύστημα client-server είναι ένα σύστημα στο οποίο το δίκτυο ενώνει διάφορους υπολογιστικούς πόρους, ώστε οι clients να μπορούν να ζητούν υπηρεσίες από έναν server, ο οποίος προσφέρει πληροφορίες ή επιπρόσθετη υπολογιστική ισχύ».[51] «Βασικές δικτυακές λειτουργίες όπως η ανταλλαγή δεδομένων, η πρόσβαση στο Web και η πρόσβαση σε βάσεις δεδομένων» ή η

[49]Βλ. http://en.wikipedia.org/wiki/TIOBE_index#mediaviewer/File:Tiobe_index.png

[50]Βλ.http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

[51]Σημειώσεις για το Εργαστήριο «Παράλληλης Κατανεμημένης Επεξεργασίας», Τμήμα Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας. Βλ. http://www.it.uom.gr/project/client_server/theoria1.htm#%C7

λειτουργία των βασικών πρωτοκόλλων εφαρμογών στο διαδίκτυο, για παράδειγμα το HTTP βασίζονται στο εν λόγω σύστημα.[52]

Η λειτουργία του συστήματος client-server συνοψίζεται στα εξής:

1. «Ο χρήστης δημιουργεί μια αίτηση ή ένα ερώτημα».
2. «Ο client μορφοποιεί το ερώτημα και το στέλνει στο server».
3. «Ο server ελέγχει την δυνατότητα πρόσβασης του χρήστη».
4. «Ο server επεξεργάζεται το ερώτημα και επιστρέφει τα αποτελέσματα».
5. «Ο client λαμβάνει την ανταπόκριση και τη μορφοποιεί για τον χρήστη».
6. «Ο χρήστης βλέπει και χειρίζεται την πληροφορία».

Εικόνα 24- Το σύστημα client-server.
Πηγή: http://en.wikipedia.org/wiki/Client–server_model

Στην ουσία, ο client είναι ο υπολογιστής, ο οποίος θα αρχίσει την επικοινωνία με τον Server μέσω του διαδικτύου. Ο client θα κάνει μια αίτηση για υπηρεσίες, ο server θα επιστρέψει μια απάντηση στον client και στη συνέχεια ο client θα προβάλλει την απάντηση στον χρήστη.

[52]Τζανέτος, Πομόνης, «Προς το Web 3.0: Διαδικασία Ανάπτυξης και Αρχιτεκτονική Υποστήριξης εφαρμογών Παγκόσμιου Ιστού που συνδυάζουν τεχνολογίες Web 2.0 και Semantic Web.», Διδακτορική Διατριβή, Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής, Πανεπιστήμιο Πατρών, Πάτρα 2010, σ. 22.

Επισημαίνεται ότι όταν ο server λαμβάνει μια αίτηση από τον client αντιδρά με δύο τρόπους: είτε ανταποκρίνεται αμέσως στην αίτηση του client ή την βάζει σε αναμονή μαζί με άλλες αιτήσεις από clients μέχρι να μπορέσει να ανταποκριθεί στο κάλεσμά τους.

Παράδειγμα άμεσης ενεργοποίησης για την αίτηση

Χαρακτηριστικό παράδειγμα άμεσης ενεργοποίησης είναι ότι ο server καλείται με αίτηση να υπολογίσει έναν αριθμό και τον επιστρέφει άμεσα στον client.

Παράδειγμα αίτησης σε αναμονή

Αντιπροσωπευτικό παράδειγμα αίτησης σε αναμονή είναι η εκτύπωση κειμένου σε έναν εκτυπωτή δικτύου. «Ο server τοποθετεί την αίτηση σε μια ουρά μαζί με αιτήσεις εκτυπώσεων και από άλλους clients. Μετά επεξεργάζεται την αίτηση με βάση την σειρά προτεραιότητας, η οποία, σε αυτή την περίπτωση, καθορίζεται από τη σειρά με την οποία ο server παρέλαβε την απαίτηση.»

Είδη των Servers

Οι servers διακρίνονται στις παρακάτω κατηγορίες:

- «Server Εφαρμογών (Application servers).
- Server Πληροφοριών (Data servers).
- Server Υπολογισμών (Compute servers).
- Server Βάσεων Δεδομένων (Database servers).
- Server Πόρων ή Επικοινωνιών (Resource or Communications servers)».

Οι προαναφερθέντες τύποι Server μπορούν να λειτουργήσουν συνδυαστικά. Είθισται να χρησιμοποιούνται περισσότερο οι servers εφαρμογών, βάσεων δεδομένων και συναλλαγών.

Εύλογα, το είδος του server που θα χρησιμοποιήσουμε εξαρτάται από το είδος της εργασίας που θέλουμε να πραγματοποιήσουμε.[53]

11. 1 Πλεονεκτήματα client-server computing

- Η συντήρηση των συστημάτων client-server είναι εύκολη διαδικασία, αφού ο server μπορεί να αντικατασταθεί, να αναβαθμιστεί ή να εγκατασταθεί κάπου αλλού, χωρίς να γνωστοποιηθεί η οποιαδήποτε αλλαγή στον client.

- Τα δεδομένα αποθηκεύονται με ασφάλεια στους servers, οι οποίοι παρέχουν πρόσβαση στους πόρους του διαδικτύου μόνο σε αξιόπιστους clients.

- Όταν ένας client εισάγει καινούργια δεδομένα ενημερώνονται σχεδόν ταυτόχρονα και οι υπόλοιποι clients του δικτύου.

- Υπάρχουν πολλές client-server τεχνολογίες οι οποίες εξασφαλίζουν «την ασφάλεια της εφαρμογής, τη φιλικότητα της διεπαφής χρήστη (user interface) και την ευκολία χρήσης». Οι πιο γνωστές τεχνολογίες για την δημιουργία server είναι οι Hypertext Preprocessor (PHP), Java Server Pages (JSP), Active Server Pages (ASP), Server Side Includes (SSI).[54]

11.2 Μειονεκτήματα client-server computing

---

[53]Σημειώσεις για το Εργαστήριο Παράλληλης Κατανεμημένης Επεξεργασίας, Τμήμα Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας. Βλ. http://www.it.uom.gr/project/client_server/theoria1.htm#%C7

[54]Τζανέτος, Πομόνης, «Προς το Web 3.0: Διαδικασία Ανάπτυξης και Αρχιτεκτονική Υποστήριξης εφαρμογών Παγκόσμιου Ιστού που συνδυάζουν τεχνολογίες Web 2.0 και Semantic Web.», Διδακτορική Διατριβή, Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής, Πανεπιστήμιο Πατρών, Πάτρα 2010, σ. 23, 24.

- Όταν ο server λαμβάνει πολλές ταυτόχρονες client αιτήσεις, ενδέχεται να υπερφορτωθεί.

- Αν πέσει ο server (fail), οι αιτήσεις των clients δε θα απαντηθούν.[55]

12. Αλγόριθμος αναζήτησης κατά Βάθος (depth- first-search ή dfs)

Ορισμός Αλγόριθμου

«Ένας αλγόριθμος είναι "μία καλά σχεδιασμένη" υπολογιστική διαδικασία η οποία δέχεται ένα σύνολο τιμών ως είσοδο και παράγει ένα σύνολο τιμών ως έξοδο».

Αναζήτηση κατά Βάθος (Depth-First Search ή DFS)

- Η DFS διασχίζει το γράφημα όσο «"πιο βαθιά"» γίνεται.
- «Εξερευνούνται οι ακμές που ξεκινούν από τον πιο πρόσφατα ανακαλυφθέντα κόμβο u, ο οποίος έχει ακόμη ανεξερεύνητες ακμές».[56]
- Ο αλγόριθμος επισκέπτεται «όλους τους κόμβους ενός γράφου μετακινούμενος πάντα προς κάποιο γειτονικό κόμβο, εάν τέτοιος υπάρχει και δεν τον έχει ήδη ξαναεπισκεφθεί. Εάν δεν υπάρχει τέτοιος κόμβος ο αλγόριθμος επιστρέφει προς τα πίσω και εκτελείται αναδρομικά».[57]

---

[55]Τζανέτος, Πομόνης, «Προς το Web 3.0: Διαδικασία Ανάπτυξης και Αρχιτεκτονική Υποστήριξης εφαρμογών Παγκόσμιου Ιστού που συνδυάζουν τεχνολογίες Web 2.0 και Semantic Web.», Διδακτορική Διατριβή, Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής, Πανεπιστήμιο Πατρών, Πάτρα 2010, σ. 24, 25.

[56]Αντώνιος, Συμβώνης, Κατερίνα, Ποτίκα, Διαλέξεις 1 και 4 του μαθήματος «Αλγόριθμοι και Πολυπλοκότητα». Βλ. https://semfe.gr/files/users/477/algorithmoi_dialekseis_1-8.pdf

[57]Ηλίας, Κ. Σάββας, Σημειώσεις για το μάθημα «Αλγόριθμοι και Πολυπλοκότητα», Τμήμα Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών, Λάρισα Ιανουάριος 2005, σ. 94. Βλ. http://www.teilar.gr/dbData/ProfAnn/profann-2a82b2bb.pdf

- Συνοψίζοντας, η φιλοσοφία του αλγόριθμου κατά βάθος είναι ότι ψάχνουμε «τον πιο απομακρυσμένο ‘‘απόγονο’’» πριν επισκεφθούμε τον επόμενο γειτονικό κόμβο,[58] δηλαδή ερευνούμε και εξαντλούμε πρώτα όλους τους πιθανούς γειτονικούς κόμβους που υπάρχουν στο γράφημα και έπειτα γυρίζουμε προς τα πίσω.

Στο σημείο αυτό, θα παραλληλίσουμε το γράφημα με έναν λαβύρινθο και θα σκιαγραφήσουμε τον τρόπο που θα λειτουργούσαμε αν θα έπρεπε να τον διασχίσουμε.

- Αρχίζουμε από τον s και επιλέγουμε την πρώτη ακμή έξω από αυτόν, στον κόμβο v.
- Επαναλαμβάνουμε την ίδια διαδικασία μέχρι να βρεθούμε σε «‘‘αδιέξοδο’’», δηλαδή σε έναν κόμβο του οποίου θα έχουμε εξετάσει όλους τους γείτονες.
- Έπειτα, κατευθυνόμαστε προς τα πίσω μέχρι να βρούμε κάποιον ανεξερεύνητο γειτονικό κόμβο και συνεχίζουμε την πορεία μας από εκεί.

DFS(u)

«Σημείωσε ότι το u "εξερευνήθηκε" και πρόσθεσε το u στην συνεκτική

συνιστώσα R

for κάθε ακμή (u, v) do

    if ο κόμβος v δεν έχει "εξερευνηθεί" then

        κάλεσε αναδρομικά την DFS(v)

    end

end» [59]

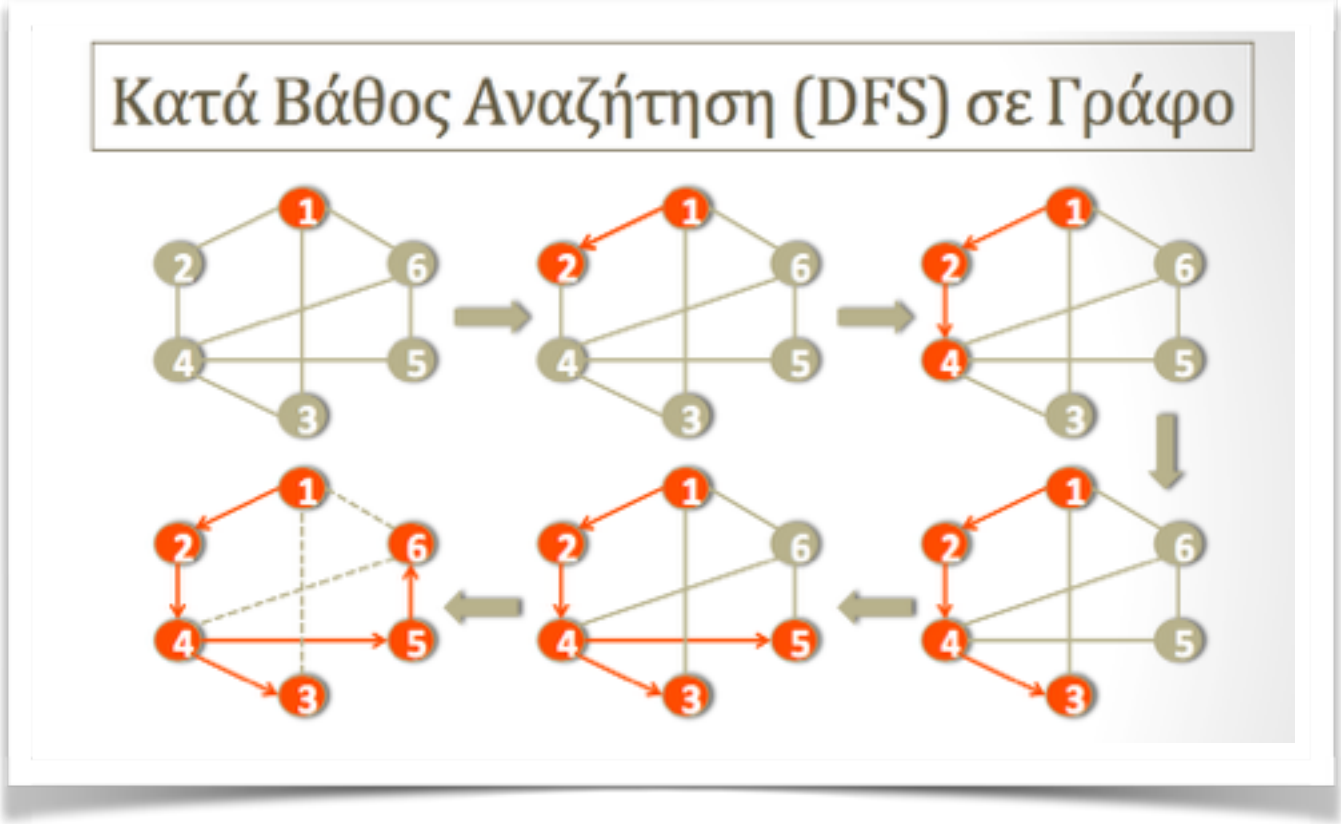
---

[58]Νίκος, Νικολάου, Σημειώσεις για το μάθημα «Κατανεμημένοι Αλγόριθμοι», Διάλεξη 3: «Αλόριθμοι σε Γράφους ΙΙ». Βλ. http://www.cs.ucy.ac.cy/~nicolasn/epl432/Lectures/03-graph-algorithms-II.pdf

[59]Δημήτρης, Μιχαήλ, Σημειώσεις για το μάθημα «Αλγόριθμοι και Πολυπλοκότητα», Τμήμα Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο, σ. 26, 27. Βλ. http://dm.hua.gr/michail/teaching/algs/slides/030_Graphs.pdf

12. 1 Παραδείγματα για την κατανόηση του αλγόριθμος αναζήτησης κατά Βάθος (depth- first-search ή dfs)[60]

Πρώτο παράδειγμα:



Εικόνα 25- Παράδειγμα Κατά Βάθους Αναζήτησης, Πηγή Φωτογραφίας: Νίκος Νικολάου, Σημειώσεις για το μάθημα «Κατανεμημένοι Αλγόριθμοι», Διάλεξη 3: «Αλόριθμοι σε Γράφους ΙΙ». Βλ.http://www.cs.ucy.ac.cy/~nicolasn/epl432/Lectures/03-graph-algorithms-II.pdf

[60]Τα παραδείγματα αντλήθηκαν από τις παρακάτω πηγές:

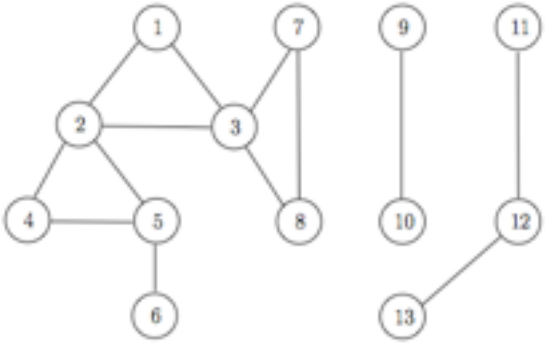
Δημήτρης, Μιχαήλ, Σημειώσεις για το μάθημα «Αλγόριθμοι και Πολυπλοκότητα», Τμήμα Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο. Βλ. http://dm.hua.gr/michail/teaching/algs/slides/030_Graphs.pdf

Επίσης, βλ. Νίκος, Νικολάου, Σημειώσεις για το μάθημα «Κατανεμημένοι Αλγόριθμοι», Διάλεξη 3: «Αλόριθμοι σε Γράφους ΙΙ». Βλ. http://www.cs.ucy.ac.cy/~nicolasn/epl432/Lectures/03-graph-algorithms-II.pdf

Δεύτερο Παράδειγμα:

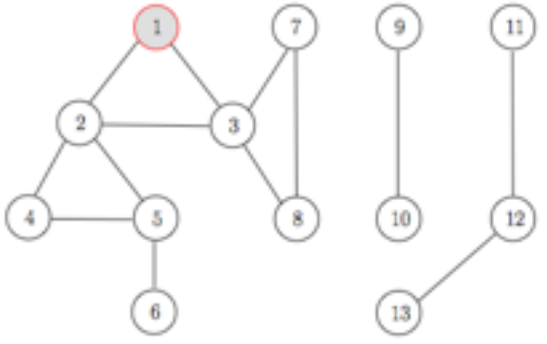
Αναζήτηση Πρώτα Κατά Βάθος

Παράδειγμα DFS από κόμβο v_1



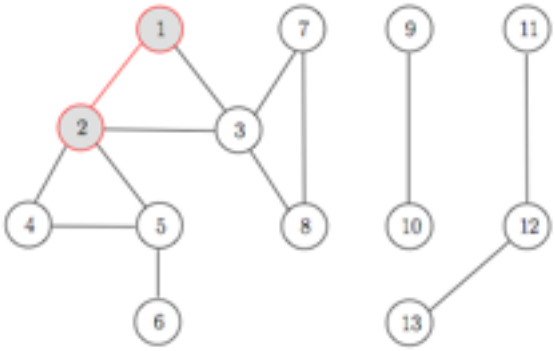
Αναζήτηση Πρώτα Κατά Βάθος

Παράδειγμα DFS από κόμβο v_1



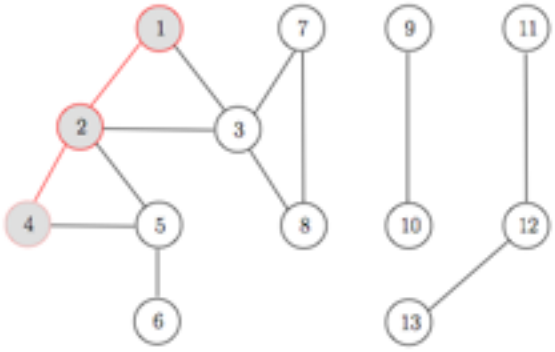
Αναζήτηση Πρώτα Κατά Βάθος

Παράδειγμα DFS από κόμβο v_1



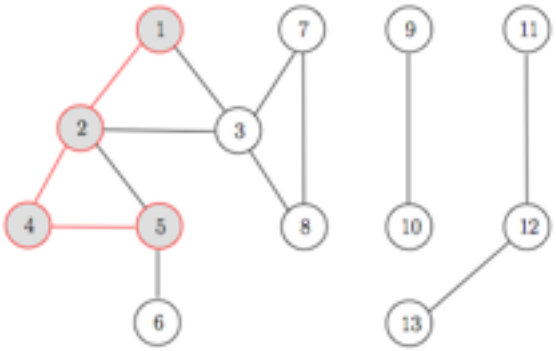
Αναζήτηση Πρώτα Κατά Βάθος

Παράδειγμα DFS από κόμβο v_1



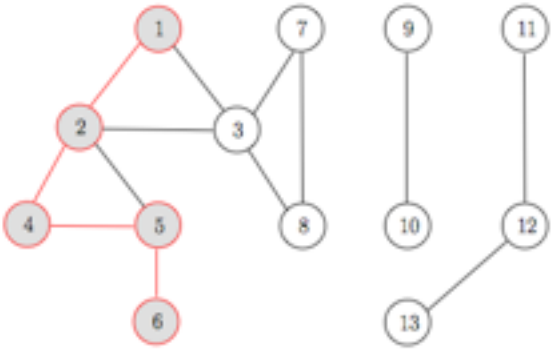
Αναζήτηση Πρώτα Κατά Βάθος

Παράδειγμα DFS από κόμβο v_1



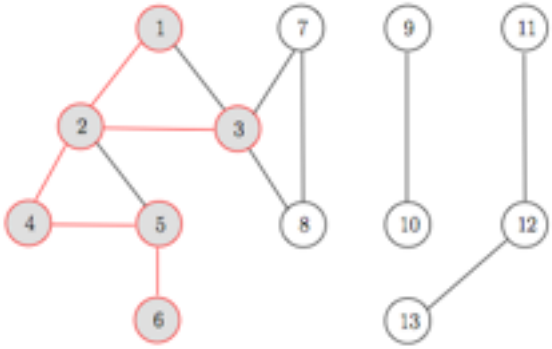
Αναζήτηση Πρώτα Κατά Βάθος

Παράδειγμα DFS από κόμβο v_1



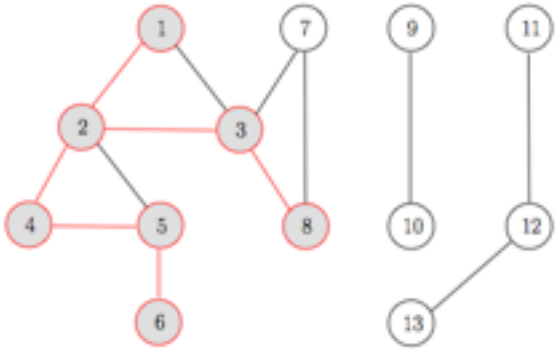
Αναζήτηση Πρώτα Κατά Βάθος

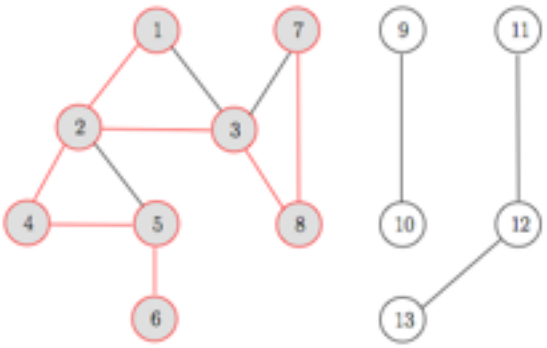
Παράδειγμα DFS από κόμβο v_1



Αναζήτηση Πρώτα Κατά Βάθος

Παράδειγμα DFS από κόμβο v_1





Εικόνα 26- Παράδειγμα αλγόριθμου DFS.
Πηγή: Δημήτρης, Μιχαήλ, Σημειώσεις για το μάθημα «Αλγόριθμοι και Πολυπλοκότητα», Τμήμα Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο. Βλ. http://dm.hua.gr/michail/teaching/algs/slides/030_Graphs.pdf

13. Αφορμή για τη γέννεση της εφαρμογής Nextktel

Εικόνα 27- Εικονίδιο της Εφαρμογής Nextktel, Πηγή: http://www.clipartbest.com/bus-stop-logo

Στην παρούσα εργασία σχεδιάστηκε και υλοποιήθηκε η εφαρμογή Nextktel. Το πρώτο συνθετικό αφενός παραπέμπει στην εταιρεία NeXT του Steve Jobs, αφετέρου δηλώνει τις ενδιάμεσες στάσεις και τις αλλαγές των λεωφορείων που χρειάζεται να πραγματοποιήσει ο χρήστης για να φτάσει στον προορισμό του.

Η εφαρμογή Nextktel σχετίζεται άμεσα με θέματα που αφορούν την καθημερινή μετακίνηση και έχει απώτερο σκοπό να διευκολύνει τους χρήστες που πραγματοποιούν διαδρομές ΚΤΕΛ. Εύλογα, είναι κουραστικό

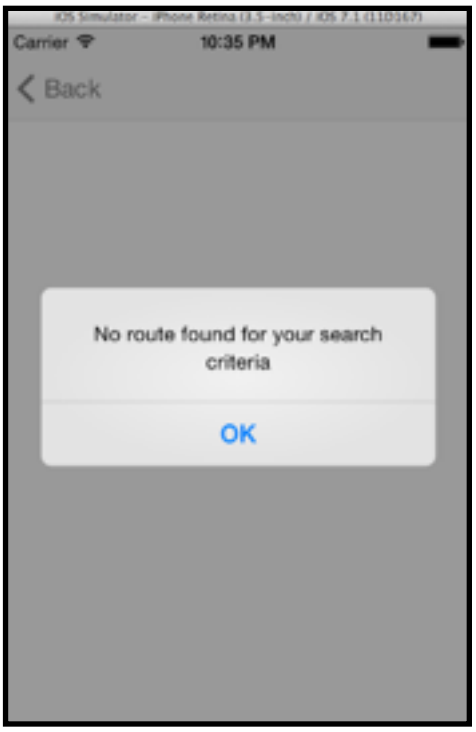
για όσους ταξιδεύουν συχνά με ΚΤΕΛ να ψάχνουν στις ιστοσελίδες ή να τηλεφωνούν στους αρμόδιους των ΚΤΕΛ για να πληροφορηθούν την ώρα αναχώρησης και τις αλλαγές των λεωφορείων που πρέπει να πραγματοποιήσουν για να φτάσει εγκαίρως στον προορισμό του.

Ωστόστο, το εναρκτήριο λάκτισμα για την δημιουργία της εν λόγω εφαρμογής δόθηκε όταν διαπίστωσα ότι για την διαδρομή Λευκάδα - Άρτα δεν υπήρχε απευθείας δρομολόγιο. Κατόπιν τούτου, για να φτάσω εγκαίρως στην περιοχή όπου στεγάζεται το ΤΕΙ της Άρτας έπρεπε να ακολουθήσω την διαδρομή Λευκάδα - Πρέβεζα - Φιλιππιάδα - Άρτα - Κωστακίους αλλάζοντας τέσσερις συγκοινωνίες, αφού το Σάββατο δεν υπάρχει δρομολόγιο για την διαδρομή Πρέβεζα - Άρτα.

14. Περιγραφή της εφαρμογής Nextktel

Εικόνα 28- XCode, Πηγή: https://developer.apple.com/xcode/

Για τη σχεδίαση και την υλοποίηση της εφαρμογής Nextktel χρησιμοποιήθηκε η πλατμφόρμα XCode 5 και το λειτουργικό iOS 7. Όσον αφορά στον τρόπο λειτουργίας της εφαρμογής, ο χρήστης χρειάζεται να πληκτρολογήσει μόνο την αφετηρία, τον προορισμό, την ώρα, και την ημερομηνία που επιθυμεί να ταξιδεύσει. Η εφαρμογή θα εμφανίσει το δρομολόγιο που ξεκινά την πλησιέστερη ώρα από την ώρα αναχώρησης που έχει επιλέξει ο χρήστης.



Εικόνα 29: Οι όψεις της εφαρμογής Nextktel.

Για την δημιουργία της εν λόγω εφαρμογής εξετάστηκαν τα δρομολόγια ΚΤΕΛ Ναύπλιο - Ισθμός - Λευκάδα, Λευκάδα - Πρέβεζα - Άρτα, Λευκάδα - Πρέβεζα - Φιλιππιάδα - Άρτα. Σύμφωνα με τις υποδείξεις των αρμόδιων υπαλλήλων του ΚΤΕΛ, η διαδρομή Ναύπλιο - Ισθμός με λεωφορείο αντιστοιχεί χρονικά σε μια ώρα, η διαδρομή Ισθμός - Λευκάδα διαρκεί τέσσερις ώρες, η διαδρομή Λευκάδα - Πρέβεζα είναι μισή ώρα, Πρέβεζα - Άρτα μία ώρα, Πρέβεζα - Φιλιππιάδα σαράντα λεπτά, Φιλιππιάδα - Άρτα μισή ώρα.

Τα δρομολόγια τοποθετήθηκαν στο Json αρχείο και χρησιμοποιήθηκε ο αλγόριθμος κατά βάθος για να βρεθεί η διαδρομή με τον μικρότερο χρόνο άφιξης στον προορισμό του χρήστη. Ο χρήστης μπορεί είτε να κατεβάσει το αρχείο Json από τον Server μέσω του διαδικτύου, είτε να το πάρει απευθείας από τη μνήμη του κινητού του σε περίπτωση που δεν είναι συνδεδεμένος με το διαδίκτυο.

Για την δημιουργία του Server χρησιμοποιήθηκε η έκδοση Azure.[61] Συγκεκριμένα, δημιούργησα μια web σελίδα και ανέβασα το Json αρχείο στον Server χρησιμοποιώντας το παρακάτω URL[62]: http://webapplication37356.azurewebsites.net/vertices.html.

---

[61] http://azure.microsoft.com/en-us/

Για την δημιουργία του Server αντλήθηκαν πληροφορίες από το παρακάτω tutorial:

http://azure.microsoft.com/en-us/documentation/articles/web-sites-dotnet-get-started/

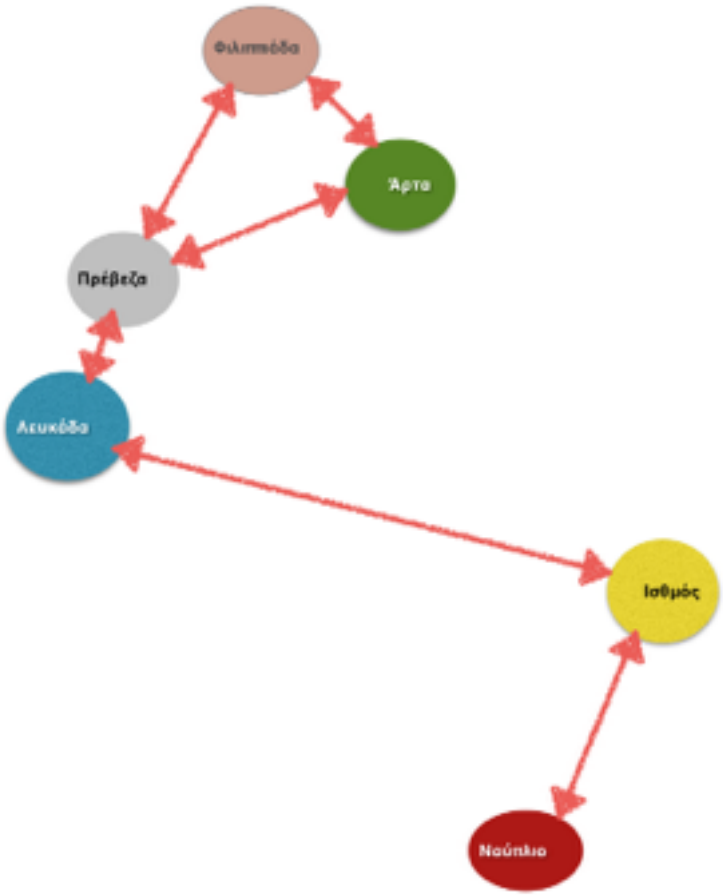
[62] Το URL (Uniform Resource Locator = «Παγκόσμιος Εντοπιστής Πόρων») είναι μια διεύθυνση, η οποία εμπεριέχει πληροφορίες για «το που βρίσκεται ένα αρχείο και το τι θα πρέπει να κάνει ο φυλλομετρητής (browser) με αυτό». Βλ. Elizabeth Castro, Bruce Hyslop, *HTML5 και CSS3*, μτφ. Φώτης Σκουλαρίκης, Κλειδάριθμος, έβδομη αμερικανική έκδοση, 2013, σ. 46.

15. Υλοποίηση του Αλγόριθμου κατά βάθος στην εφαρμογή Nextktel

Ο αλγόριθμος κατά βάθος στην εφαρμογή Nextktel υλοποιείται μέσω της παρακάτω συνάρτησης:

```
-(void)findWeightedGraphForStartingPoint:(NSString *)startPoint
                              toEndPoint:(NSString *)endPoint
                           atStartingTime:(NSDate *)startingTime
                          andVertecesHistory:(NSMutableArray *)vertecesHistory
```

Όταν καλείται για πρώτη φορά η προαναφερθείσα συνάρτηση στο startPoint τοποθετείται η αφετηρία, το endPoint (= ο προορισμός), το startingTime (= η ώρα αναχώρησης) που έχει επιλέξει ο χρήστης και το vertecesHistory, όπου αρχικά υπάρχει ένα κενό Array.



Εικόνα 30-
Ο αλγόριθμος κατά Βάθος στην εφαρμογή Nextktel.

Αξιοσημείωτο είναι ότι η εν λόγω συνάρτηση φιλτράρει όλα τα διανύσματα που ανήκουν στο vertecesArray με βάση τα παρακάτω κριτήρια:

1ο κριτήριο του χώρου : startPoint

Ως startPoint θεωρείται ο εκάστοτε κόμβος που εξετάζεται σε κάθε αναζήτηση από τη συνάρτηση findWeightedGraphForStartingPoint. Μετά το πρώτο φιλτράρισμα το Array ονομάζεται startvertices και περιλαμβάνει μόνο όσα διανύσματα ξεκινούν από τον κόμβο που εξετάζεται εκείνη τη στιγμή.

Για παράδειγμα, αν ο χρήστης έχει επιλέξει την διαδρομή Λευκάδα - Άρτα, ως πρώτο StartPoint θεωρείται η αφετηρία που έχει επιλέξει ο χρήστης. Στη συνέχεια, καλείται αναδρομικά η συνάρτηση findWeightedGraphForStartingPoint και ως StartPoint θεωρείται αυτή τη φορά το διάνυσμα του αμέσως επόμενου γειτονικού κόμβου, δηλαδή η Πρέβεζα και ούτω καθεξής.

2ο κριτήριο του χώρου και του χρόνου: StartVertecesWithOkTime.

Από τον πίνακα startvertices δημιουργούμε το νέο πίνακα StartVertecesWithOkTime. Στον πίνακα StartVertecesWithOkTime αποθηκεύονται μόνο τα διανύσματα που ξεκινούν από τον κόμβο που εξετάζει η συνάρτηση την δεδομένη στιγμή, για παράδειγμα για την διαδρομή Λευκάδα - Άρτα εξετάζουμε την Πρέβεζα και σε χρόνο μεταγενέστερο από τη στιγμή που φτάσαμε στην Πρέβεζα. Παραδείγματος χάρη, αν φτάσαμε στη 1.00 μ.μ. στην Πρέβεζα θα αποθηκευτούν στον πίνακα StartVertecesWithOkTime όσα δρομολόγια ξεκινούν από την Πρέβεζα στη 1.00 μ.μ. και μετά.

Έπειτα, ο αλγόριθμος συγκρίνει τις ημερομηνίες vertex.dateStart και την ώρα αναχώρησης (startingTime) και ελέγχει ποια ημερομηνία είναι μεγαλύτερη από την άλλη. Για παράδειγμα, παραβάλλουμε τον χρόνο που ο αλγόριθμος έφτασε στην Πρέβεζα με το dateStart των διανυσμάτων που ξεκινούν από την Πρέβεζα. Αν ο αλγόριθμος έφτασε στη 1.00 μ.μ. στην Πρέβεζα θα κρατήσει όσα διανύσματα ξεκινούν από τη 1.00 μ.μ. και μετά.

3ο κριτήριο του χώρου, του χρόνου και να μην έχουμε ξαναπεράσει από την ίδια πόλη: StartVertecesWithOkTimeThaDontLookBack

Από τα διανύσματα που βρίσκονται στον πίνακα startVertecesWithOkTime επιλέγονται όσα δεν έχουν ως προορισμό κάποια από τις περιοχές που έχουμε ήδη επισκεφθεί και αποθηκεύονται στο νέο πίνακα StartVertecesWithOkTimeThaDontLookBack. Επομένως, ο πίνακας startVertecesWithOkTimeThaDontLookBack εμπεριέχει όσα διανύσματα πληρούν τα εξής κριτήρια: του χώρου, του χρόνου και δεν έχουν ως προορισμό κάποια περιοχή που έχουμε ήδη επισκεφθεί.

Αφού εξεταστούν τα παραπάνω τρία κριτήρια καταλήγουμε σε ένα Array που περιλαμβάνει τις πιθανές διαδρομές.

Στη συνέχεια, ελέγχουμε αν τα διανύσματα που βρίσκονται στο Array StartVertecesWithOkTimeThaDontLookBack ικανοποιούν την παρακάτω συνθήκη: αν ο προορισμός τους, δηλαδή το vertex.to ταυτίζεται με τον τελικό προορισμό που έχει επιλέξει ο χρήστης. Αν όντως καταλήγει στον τελικό προορισμό αποθηκεύουμε το ιστορικό, που συμπεριλαμβάνει και το τελικό διάνυσμα, στο possibleRouts και τελειώνει ο αλγόριθμος.

Εντούτοις, αν το vertex.to δεν είναι ο τελικός προορισμός ξανακαλούμε τη συνάρτηση findWeightedGraphForStartingPoint με διαφορετικά ορίσματα. Αυτή την φορά θέτουμε ως StartPoint το vertex.to, δηλαδή τον προορισμό του διανύσματος και ως startingTime θεωρούμε τον χρόνο που φτάσαμε στον προορισμό «to». Για παράδειγμα, αν η διαδρομή που έχει επιλέξει ο χρήστης είναι «Λευκάδα - Άρτα» και το διάνυσμα έφτασε μέχρι την Πρέβεζα, ξανακαλούμε το εν λόγω διάνυσμα με StartPoint αυτήν την φορά την Πρέβεζα και startingTime τον χρόνο που έφτασε στην Πρέβεζα. Το γεγονός ότι καλούμε την ίδια συνάρτηση με διαφορετικά ορίσματα λέγεται αναδρομή. Η αναδρομή τελειώνει όταν το vertex.to συνάδει με τον τελικό προορισμό που έχει επιλέξει ο χρήστης.

Όταν βρούμε το τελικό αποτέλεσμα περνάμε το VertHistory[63] στο possibleRouts.[64] Κατόπιν τούτου, το possibleRouts περιλαμβάνει όλες τις επιτυχείς ακολουθίες διανυσμάτων, οι οποίες συνάδουν με την αφετηρία και τον προορισμό που έχει επιλέξει ο χρήστης. Έπειτα, ταξινομούμε στο possibleRouts όσα array από vertices (= σύνολο διανυσμάτων)[65] φτάνουν στον τελικό προορισμό. Η ταξινόμηση γίνεται με βάση τον χρόνο άφιξης στον προορισμό, δηλαδή καλούμε τη συνάρτηση findWeightedGraphForStartingPoint και διατάσσουμε στο possibleRouts ένα - ένα τα διανύσματα με βάση τον χρόνο άφιξής τους στον τελικό προορισμό. Τέλος, διαλέγουμε το Array διανυσμάτων που έχει τον μικρότερο χρόνο άφιξης στον τελικό προορισμό, οπότε τελειώνει οριστικά ο αλγόριθμος.

Εικόνα 31- Απεικόνιση του Αλγόριθμου κατά Βάθος σε χάρτη με τους κόμβους που χρησιμοποιήθηκαν στην εφαρμογή Nextktel. Πηγή: apple maps.

[63]Επισημαίνεται ότι το VertHistory περιλαμβάνει όλα τα διανύσματα που έχουμε εξετάσει μέχρι στιγμής κατά την αναδρομή, δηλαδή τα διανύσματα από τα οποία έχουμε περάσει για να φτάσουμε στο σημείο που βρισκόμαστε την δεδομένη στιγμή. Για παράδειγμα, αν εξετάσουμε την διαδρομή Λευκάδα - Άρτα το VertHistory θα εμπεριέχει όλα τα διανύσματα από την αφετηρία μέχρι τον προορισμό.

[64]Το possibleRouts περιλαμβάνει όλες τις επιτυχείς διαδρομές οι οποίες ξεκινούν από την αφετηρία και καταλήγουν στον προορισμό.

[65]Η κάθε διαδρομή αποτελείται από ένα array και το κάθε array αποτελείται από ένα σύνολο διανυσμάτων.

Στο σημείο αυτό, παρατίθεται ο κώδικας της εν λόγω εφαρμογής και σχολιάζονται ακροθιγώς ορισμένα καίρια σημεία του κώδικα. Επισημαίνεται ότι για τη συγγραφή του κώδικα αντλήθηκαν στοιχεία κυρίως από πηγές του διαδικτύου.

Όσον αφορά στον κώδικα της εφαρμογής, αξίζει να αναφερθεί ότι η κλάση UIViewController δημιουργεί συνολικά δύο Controllers, τον Homeviewcontroller και τον Secondviewcontroler, οι οποίοι αντιστοιχούν σε διαφορετικά views που προβάλλονται στην οθόνη. Επομένως, θα δημιουργηθεί ένας View Controller για κάθε full-screen view της εφαρμογής.

16. Κώδικας της εφαρμογής NextKtel

Κώδικας αρχείου AppDelegate.h[66]

```
//
// AppDelegate.h
// NextKtel
//
// Created by Georgia on 11/9/13.
// Copyright (c) 2013 Georgia. All rights reserved.
//

#import <UIKit/UIKit.h>

@interface AppDelegate : UIResponder<UIApplicationDelegate>
```

[66]Για τη συγγραφή του κώδικα στο αρχείο AppDelegate.h και .m αντλήθηκαν στοιχεία κυρίως από την παρακάτω διευθύνση:

www.stackoverflow.com/questions/19774738/coredata-managedobjectcontext-not-being-created

@property (strong, nonatomic) UIWindow *window;

@property (strong, nonatomic) UINavigationController *navController;

@property (strong, nonatomic) NSMutableArray *verticesArray;

@property (nonatomic, retain, readonly) NSManagedObjectModel *managedObjectModel;

@property (nonatomic, retain, readonly) NSManagedObjectContext *managedObjectContext;

@property (nonatomic, retain, readonly) NSPersistentStoreCoordinator *persistentStoreCoordinator;

@end

[Στο αρχείο AppDelegate.h γίνεται η δήλωση των properties της κλάσης AppDelegate.]

Κώδικας αρχείου AppDelegate.m[67]

---

[67]Για την δημιουργία του κώδικα στο αρχείο AppDelegate.m αντλήθηκαν στοιχεία κυρίως από τις ακόλουθες πηγές:

Ankit, Aggarwal, “iOS Core Data Tutorial With Example”, (30/7/2013).

Βλ. http://www.codigator.com/tutorials/ios-core-data-tutorial-with-example/

Ray, Wenderlich, “Core Data on iOS 5 Tutorial: How To Preload and Import Existing Data”, (23/10/2012).

Βλ. http://www.raywenderlich.com/forums/viewtopic.php?f=20&t=3241&start=40

Ray, Wenderlich, “Working with JSON in iOS 5 Tutorial” (28/10/2011). Βλ. http://www.raywenderlich.com/5492/working-with-json-in-ios-5

Βλ.http://stackoverflow.com/questions/21303892/loading-a-property-list-with-a-universal-file-path

Βλ.http://stackoverflow.com/questions/19774738/coredata-managedobjectcontext-not-being-created

```
//
//  AppDelegate.m
//  NextKtel
//
//  Created by Georgia on 11/9/13.
//  Copyright (c) 2013 Georgia. All rights reserved.
//
#import "AppDelegate.h"
#import "HomeViewController.h"

@implementation AppDelegate
```

[Στο αρχείο AppDelegate.m γίνεται η υλοποίηση του κώδικα.]

```
@synthesize managedObjectContext = _managedObjectContext;
@synthesize managedObjectModel = _managedObjectModel;
@synthesize persistentStoreCoordinator = _persistentStoreCoordinator;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:
  (NSDictionary *)launchOptions

  self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
  // Override point for customization after application launch.

  HomeViewController *homeVc = [[HomeViewController alloc] init];
self.navController = [[UINavigationController alloc] initWithRootViewController:homeVc];
```

[Ο NavigationController κρατάει και διαχειρίζεται τη στοίβα των ViewControllers. Ο NavigationController αποθηκεύεται στο property navController του αρχείου AppDelegate.h και παίρνει ως είσοδο το αντικείμενο homeVc. Ορίζουμε το αντικείμενο homeVc να αντιπροσωπεύει το RootViewController.]

[self.window setRootViewController:self.navController];

[Θέτουμε το navController ως RootViewController στο window.]

```
    self.window.backgroundColor = [UIColor whiteColor];
    [self.window makeKeyAndVisible];
    return YES;
}
```

[Η μέθοδος didFinishLaunchingWithOptions επιστρέφει «yes» για να ενημερώσει το στοιχείο UIApplication ότι η εφαρμογή είναι έτοιμη να ξεκινήσει.][68]

```
- (void)applicationWillResignActive:(UIApplication *)application
{
    // Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.
    // Use this method to pause ongoing tasks, disable timers, and throttle down OpenGL ES frame rates. Games should use this method to pause the game.
```

[68]Bart, Jacobs, “First Steps with UIKit”, (17/12/2012). Βλ. http://code.tutsplus.com/tutorials/first-steps-with-uikit--mobile-14013

```
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    // Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its current state in case it is terminated later.
    // If your application supports background execution, this method is called instead of applicationWillTerminate: when the user quits.
}

- (void)applicationWillEnterForeground:(UIApplication *)application
{
    // Called as part of the transition from the background to the inactive state; here you can undo many of the changes made on entering the background.
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
    // Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, optionally refresh the user interface.
}

- (void)applicationWillTerminate:(UIApplication *)application
{
```

```
    // Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:.
}

#pragma mark - Core Data stack

// Returns the managed object context for the application.
// If the context doesn't already exist, it is created and bound to the persistent store coordinator for the application.
- (NSManagedObjectContext *)managedObjectContext
{
    if (_managedObjectContext != nil) {
        return _managedObjectContext;
    }

    NSPersistentStoreCoordinator *coordinator = [self persistentStoreCoordinator];
    if (coordinator != nil) {
        _managedObjectContext = [[NSManagedObjectContext alloc] init];
        [_managedObjectContext setPersistentStoreCoordinator:coordinator];
    }
    return _managedObjectContext;
}
```

[Η μέθοδος θα επιστρέψει το «managedObjectContext» αντικείμενο που συνδέεται με τη βάση δεδομένων από το αρχείο Sqlite.[69]]

```
// Returns the managed object model for the application.
// If the model doesn't already exist, it is created from the application's model.
- (NSManagedObjectModel *)managedObjectModel
{
   if (_managedObjectModel != nil)
   {
      return _managedObjectModel;
   }
      NSURL *modelURL = [[NSBundle mainBundle] URLForResource:@"KtelDataModel" withExtension:@"momd"];
            _managedObjectModel = [[NSManagedObjectModel alloc] initWithContentsOfURL:modelURL];
   return _managedObjectModel;
}
```

[Η μέθοδος θα επιστρέψει το «managedObjectModel» αντικείμενο που συνδέεται με τη βάση δεδομένων από το αρχείο Sqlite.][70]

---

[69]Ο κώδικας αντλήθηκε από την ακόλουθη πηγή:

Ankit, Aggarwal, “iOS Core Data Tutorial With Example”, (30/7/2013).

Βλ. http://www.codigator.com/tutorials/ios-core-data-tutorial-with-example/

[70]Ο κώδικας αντλήθηκε από την ακόλουθη πηγή:

Βλ. Ankit, Aggarwal, “iOS Core Data Tutorial With Example”, (30/7/2013).

http://www.codigator.com/tutorials/ios-core-data-tutorial-with-example/

```
// Returns the persistent store coordinator for the application.
// If the coordinator doesn't already exist, it is created and the application's store added to it.
- (NSPersistentStoreCoordinator *)persistentStoreCoordinator
{
    if (_persistentStoreCoordinator != nil)
    {
        return _persistentStoreCoordinator;
    }
            NSURL *storeURL = [[self applicationDocumentsDirectory]
URLByAppendingPathComponent:@"NextKtel.sqlite"];

    NSError *error = nil;
    _persistentStoreCoordinator = [[NSPersistentStoreCoordinator alloc]
                                   initWithManagedObjectModel:[self managedObjectModel]];

  if (![_persistentStoreCoordinator addPersistentStoreWithType:NSSQLiteStoreType
                                                 configuration:nil
                                                           URL:storeURL
                                                       options:nil
                                                         error:&error])
    {
        /*
         Replace this implementation with code to handle the error appropriately.
```

abort() causes the application to generate a crash log and terminate. You should not use this function in a shipping application, although it may be useful during development.

Typical reasons for an error here include:

*** The persistent store is not accessible;**

*** The schema for the persistent store is incompatible with current managed object model.**

Check the error message to determine what the actual problem was.

If the persistent store is not accessible, there is typically something wrong with the file path. Often, a file URL is pointing into the application's resources directory instead of a writeable directory.

If you encounter schema incompatibility errors during development, you can reduce their frequency by:

*** Simply deleting the existing store:**

[[NSFileManager defaultManager] removeItemAtURL:storeURL error:nil]

*** Performing automatic lightweight migration by passing the following dictionary as the options parameter:**

@{NSMigratePersistentStoresAutomaticallyOption:@YES, NSInferMappingModelAutomaticallyOption:@YES}

Lightweight migration will only work for a limited set of schema changes; consult "Core Data Model Versioning and Data Migration Programming Guide" for details.

```
         */
        NSLog(@"Unresolved error %@, %@", error, [error userInfo]);
        abort();
    }

    return _persistentStoreCoordinator;
}
```

[Η μέθοδος θα επιστρέψει το «persistentStoreCoordinator» αντικείμενο που συνδέεται με τη βάση δεδομένων από το αρχείο Sqlite.][71]

```
#pragma mark - Application's Documents directory

// Returns the URL to the application's Documents directory.
- (NSURL *)applicationDocumentsDirectory
{
    return [[[NSFileManager defaultManager]
            URLsForDirectory:NSDocumentDirectory
            inDomains:NSUserDomainMask] lastObject];
}
```

[71] Ο κώδικας αντλήθηκε από την ακόλουθη πηγή:

Βλ. Ankit, Aggarwal, “iOS Core Data Tutorial With Example”, (30/7/2013).

http://www.codigator.com/tutorials/ios-core-data-tutorial-with-example/

@end[72]

Κώδικας αρχείου HomeViewController.h

```
//
//  HomeViewController.h
//  NextKtel
//
//  Created by Georgia on 11/10/13.
//  Copyright (c) 2013 Georgia. All rights reserved.
//

#import <UIKit/UIKit.h>

@interface HomeViewController : UIViewController <UIActionSheetDelegate>

@property (strong, nonatomic) IBOutlet UILabel *selectStartingPointLabel;
@property (strong, nonatomic) IBOutlet UIButton *selectStartingPointButton;
@property (strong, nonatomic) IBOutlet UILabel *selectDestinationLabel;
@property (strong, nonatomic) IBOutlet UIButton *selectDestinationButton;
```

[72] Ο κώδικας αντλήθηκε από την παρακάτω πηγή: Βλ.http://stackoverflow.com/questions/21303892/loading-a-property-list-with-a-universal-file-path

@property (strong, nonatomic) IBOutlet UILabel *timeLabel;

@property (strong, nonatomic) IBOutlet UIDatePicker *datePicker;

[Στο interface του αρχείου HomeViewController.h ορίζονται τα properties των κλάσεων UILabel, UIButton, UIDatePicker και NSMutableArray. Με το IBOutlet συνδέεται το αρχείο HomeViewController.h με το αρχείο HomeViewController.xib.]

@property(strong, nonatomic) NSMutableArray *locationNamesArray;

@property(strong, nonatomic) NSMutableArray *ASButtonTitles;

@property(strong, nonatomic) NSMutableArray *verticesArray;

[Όσον αφορά στο αντικείμενο verticesArray, αποτελεί έναν πίνακα ο οποίος εμπεριέχει τα στοιχεία που βρίσκονται στο Json αρχείο και αφορούν την αφετηρία, τον προορισμό και τον χρόνο.]

@property(strong, nonatomic) NSMutableArray *visitedAreas;

@property(strong, nonatomic) NSMutableArray *savedVerteces;

@property(strong, nonatomic) NSMutableArray *possibleRouts;

@property(strong, nonatomic) NSMutableArray *vertecesHistory;

- **(IBAction)selectStartingPointBtnPressed:(id)sender;**

[Το πάτημα του κουμπιού θα πυροδοτήσει την εκτέλεση της επιλογής του σημείου αφετηρίας και θα έχει ως είσοδο το γενικού τύπου αντικείμενο sender.]

- **(IBAction)selectDestinationButtonPressed:(id)sender;**

[Το πάτημα του κουμπιού θα πυροδοτήσει την εκτέλεση της επιλογής του προορισμού και θα έχει ως είσοδο το γενικού τύπου αντικείμενο sender.]

- (IBAction)nextBtnPressed:(id)sender;

[Το πάτημα του κουμπιού θα πυροδοτήσει την εκτέλεση της επιλογής της επόμενης σελίδας και θα έχει ως είσοδο το γενικού τύπου αντικείμενο sender.]

@end

Κώδικας αρχείου HomeViewController.m

```
//
//  HomeViewController.m
//  NextKtel
//
//  Created by Georgia on 11/10/13.
//  Copyright (c) 2013 Georgia. All rights reserved.
//
#import "HomeViewController.h"
#import "SecondViewController.h"
#import "JsonParser.h"
#import "AppDelegate.h"
```

```
#import "Vertex.h"


@interface HomeViewController ()


@end

@implementation HomeViewController

{

    int kDestinationActionSheetTag;

    int kStartingPointActionSheetTag;
```

[Στις ακέραιες μεταβλητές θα μπουν τα Strings με τα ονόματα των πόλεων.]

```
}

- (void)viewDidLoad

{

    self.savedVerteces = [[NSMutableArray alloc] init];

    self.possibleRouts = [[NSMutableArray alloc] init];

    [super viewDidLoad];

    // Do any additional setup after loading the view from its nib.

    kDestinationActionSheetTag = 111;

    kStartingPointActionSheetTag = 112;
```

[Στο πρόγραμμα υπάρχουν συνολικά δύο actionsheet, το kDestinationActionSheetTag και το kStartingPointActionSheetTag.]

```
        self.selectStartingPointButton.titleLabel.text = @"Start";
        self.selectDestinationButton.titleLabel.text = @"Destination";
        self.locationNamesArray = [[NSMutableArray alloc] initWithObjects:@"Preveza",
                                   @"Lefkada", @"Filippiada", @"Arta", @"Is8mos", @"Nafplio", nil];
```

[Δημιουργούμε ένα Array που θα περιλαμβάνει όλα τα ονόματα των πόλεων.]

```
    [self.datePicker setTimeZone:[NSTimeZone localTimeZone]];
}
-(void)viewWillAppear:(BOOL)animated
{
    [self getDataFromServerAndMakeVerticesArray];
}
-(void)getDataFromServerAndMakeVerticesArray
{
    JsonParser *jsonParser = [[JsonParser alloc] init];
    NSData *jsonData = [jsonParser downlaodDataFromServer];
    if (jsonData)
    {
        ((AppDelegate *)[[UIApplication sharedApplication] delegate]).verticesArray =
        [jsonParser createVertesesFromJsonData:jsonData];
    }
```

}

-(NSArray *)calculateBestPath

[Δημιουργούμε έναν αλγόριθμο ο οποίος θα υπολογίζει τη βέλτιστη διαδρομή ως προς τον χρόνο.]

{

self.verticesArray = ((AppDelegate *)[[UIApplication sharedApplication] delegate]).verticesArray;

self.visitedAreas = [[NSMutableArray alloc] init];

[Με τον όρο visitedAreas εννοούμε τις περιοχές που έχουμε ήδη επισκεφθεί για να μην τις ξαναεπισκεφθούμε. Με αυτόν τον τρόπο θα αποφύγουμε να κάνουμε κύκλους.]

self.vertecesHistory = [[NSMutableArray alloc] init];

[Με τον όρο vertecesHistory νοείται το ιστορικό των διανυσμάτων, δηλαδή από ποια διανύσματα έχουμε περάσει για να φτάσουμε στο σημείο προορισμού.]

[self.visitedAreas removeAllObjects];

[self.possibleRouts removeAllObjects];

[self.vertecesHistory removeAllObjects];

[Κάθε φορά που πατάμε το εκάστοτε κουμπί θα διαγράφεται το ιστορικό που έχει μείνει από κάποια προηγούμενη αναζήτηση.]

[self findWeightedGraphForStartingPoint:self.selectStartingPointLabel.text

toEndPoint:self.selectDestinationLabel.text

```
                atStartingTime:self.datePicker.date

              andVertecesHistory:nil];
```

[Θα δημιουργήσω μια συνάρτηση findWeightedGraphForStartingPoint, η οποία καλείται αναδρομικά (= η ίδια καλεί τον εαυτό της). Η εν λόγω συνάρτηση θα αποτελείται από τέσσερα ορίσματα: α) το σημείο αφετηρίας β) το σημείο προορισμού γ) την ώρα αναχώρησης δ) το ιστορικό, δηλαδή αν έχει ξαναπεράσει από την ίδια πόλη.]

```
if ([self.possibleRouts count] == 0)
    {
        UIAlertView *alert =
        [[UIAlertView alloc] initWithTitle:nil
                                   message:@"No route found for your search criteria"
                                  delegate:self
                         cancelButtonTitle:@"OK"
                         otherButtonTitles:nil];
        [alert show];
        return nil;
    }
    //find fastest path
    NSMutableArray *destinationVerteces = [[NSMutableArray alloc] init];
    for (NSMutableArray *root in self.possibleRouts)
    {
```

```
        Vertex *destinationVertex = [root lastObject];
        [destinationVerteces addObject:destinationVertex];
    }

    NSSortDescriptor *datEndSD = [NSSortDescriptor
sortDescriptorWithKey:@"dateEnd" ascending:YES];
    [destinationVerteces sortUsingDescriptors:@[datEndSD]];
```

[Ο αλγόριθμος ταξινομεί αυτόματα όλες τις διαδρομές με βάση το dateEnd (= με βάση τον χρόνο που θα φτάσει στον προορισμό) και με ascending:YES (= ταξινόμηση διανυσμάτων από το μικρότερο στο μεγαλύτερο). Επισημαίνεται ότι ως βέλτιστη διαδρομή θα θεωρηθεί αυτή που θα έχει τον μικρότερο χρόνο άφιξης στον προορισμό.]

```
Vertex *bestLastVertex = [destinationVerteces objectAtIndex:0];
```

[Το objectAtIndex:0 είναι η πρώτη θέση στο destinationVerteces. Αυτή την θέση καταλαμβάνει το διάνυσμα με τον μικρότερο χρόνο άφιξης.]

```
    NSArray *bestPath;
    for (NSMutableArray *root in self.possibleRouts)
    {
        Vertex *lastObj = [root lastObject];
        if ([lastObj.dateEnd isEqualToDate:bestLastVertex.dateEnd])
        {
```

```
                bestPath = root;
                break;
            }
        }
        return bestPath;
}

-(void)findWeightedGraphForStartingPoint:(NSString *)startPoint
                              toEndPoint:(NSString *)endPoint
                           atStartingTime:(NSDate *)startingTime
                         andVertecesHistory:(NSMutableArray *)vertecesHistory
```

[Ο χρήστης έχει ορίσει ένα σημείο αφετηρίας, ένα σημείο προορισμού, την ώρα αναχώρησης και το ιστορικό των διανυσμάτων. Καλώ τη συνάρτηση findWeightedGraphForStartingPoint ανανεώνοντας κάθε φορά το startingTime με βάση το χρόνο άφιξης από το προηγούμενο διάνυσμα. Βρίσκω πάλι τα έγκυρα διανύσματα που δεν κοιτάζουν προς τα πίσω, δηλαδή τα διανύσματα των οποίων ο προορισμός δεν ανήκει στα visitedAreas.]

```
{
    [self.visitedAreas addObject:startPoint];
    NSMutableArray *startVerteces = [[NSMutableArray alloc] init];

   for (Vertex *vertex in self.verticesArray)
    {
```

[Το verticesArray εμπεριέχει όλα τα διανύσματα. Διαβάζω ένα - ένα τα διανύσματα από το verticesArray και για κάθε Vertex που ανήκει στο verticesArray ελέγχω αν πληροί τα κριτήρια του if.]

```
        if ([vertex.from isEqualToString:startPoint])
        {
            [startVerteces addObject:vertex];
```

[1ο κριτήριο: startPoint.

Αν το vertex.from είναι ίδιο με το startPoint, τότε προστίθεται στο Array startVerteces. Γεμίζω το Array με όσα αντικείμενα διανύσματα ξεκινούν από το startPoint.]

```
        }
    }
    //find verteces that start after the starting time
    NSMutableArray *startVertecesWithOkTime = [[NSMutableArray alloc] init];
```

[2ο κριτήριο: startVertecesWithOkTime.

Στον πίνακα startVertecesWithOkTime μπαίνουν όσα διανύσματα ξεκινούν μετά από τον χρόνο που φτάσαμε στο σημείο στο οποίο βρισκόμαστε.]

```
    for (Vertex *vertex in startVerteces)
    {
        if ([vertex.dateStart compare:startingTime] == NSOrderedDescending)
        {
```

```
            //date1 is later than date2

                [startVertecesWithOkTime addObject:vertex] ;

            }

            else if ([vertex.dateStart compare:startingTime] == NSOrderedAscending)

            {

            //date1 is earlier than date2

             }

            else

            {

                  //dates are the same //

                  [startVertecesWithOkTime addObject:vertex];
```

[Δια της ατόπου, αφού το startingTime δεν είναι ούτε μεγαλύτερο ούτε μικρότερο θα είναι ίσο.[73] Ελέγχω όλα τα Verteces και τοποθετώ σε ένα διάνυσμα όσα από αυτά πληρούν την προαναφερθείσα συνθήκη.]

```
            }

      }

      NSMutableArray *startVertecesWithOkTimeThaDontLookBack = [[NSMutableArray
alloc] init];
```

[3ο κριτήριο: το ζητούμενο είναι ο αλγόριθμος να μην έχει ξαναπεράσει από την ίδια πόλη.

Από τα διανύσματα που βρίσκονται στο Array startVertecesWithOkTime επιλέγουμε τα διανύσματα που δεν έχουν ως προορισμό κάποια από τις περιοχές που έχουμε ήδη επισκεφθεί και τα

[73] Βλ.http://stackoverflow.com/questions/5965044/how-to-compare-two-nsdates-which-is-more-recent

αποθηκεύουμε στον πίνακα startVertecesWithOkTimeThaDontLookBack.Το ζητούμενο είναι να μην ερευνήσουμε πάλι τις περιοχές που έχουμε ήδη επισκεφθεί.]

```
for (Vertex *vertex in startVertecesWithOkTime)
    {
        if ([self.visitedAreas containsObject:vertex.to])
        {
        }
        else
        {
            [startVertecesWithOkTimeThaDontLookBack addObject:vertex];
        }
    }
    for (Vertex *vertex in startVertecesWithOkTimeThaDontLookBack)

    {

NSMutableArray *vertHistory = [[NSMutableArray alloc] initWithArray:vertecesHistory];
        [vertHistory addObject:vertex];
        if ([vertex.to isEqualToString:endPoint])
        {
```

```
                //you find a path
                [self.possibleRouts addObject:vertHistory];
            }
            else
            {

                [self findWeightedGraphForStartingPoint:vertex.to
                                        toEndPoint:endPoint
                                    atStartingTime:vertex.dateEnd
                                    andVertecesHistory:vertHistory];
            }
        }
}
- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

- (IBAction)timeBtnPressed:(id)sender {
}
```

```
- (IBAction)selectStartingPointBtnPressed:(id)sender

{

    self.ASButtonTitles = [[NSMutableArray alloc] initWithArray:self.locationNamesArray];

    [self.ASButtonTitles addObject:@“Cancel”];

    UIActionSheet *actionSheet = [[UIActionSheet alloc] initWithTitle:nil

                                                             delegate:self

                                                    cancelButtonTitle:nil

                                               destructiveButtonTitle:nil

                                                    otherButtonTitles:nil];
```

[Δημιουργείται η κλάση UIActionSheet με delegate το self. Το ActionSheet περιλαμβάνει ένα Array με ονόματα πόλεων. Συνολικά, υπάρχουν δύο ActionSheet, το ένα περιλαμβάνει τα ονόματα των πόλεων της αφετηρίας και το άλλο τα ονόματα των πόλεων του προορισμού. Επισημαίνεται ότι και τα δύο ActionSheet έχουν τον ίδιο delegate, το self που αντιπροσωπεύει τον HomeViewControler.]

```
    actionSheet.tag = kStartingPointActionSheetTag;
```

[Στην προκειμένη περίπτωση καλείται το actionSheet της αφετηρίας.]

```
    for (NSString *title in self.ASButtonTitles)

    {

        [actionSheet addButtonWithTitle:title];
```

[Κάθε κουμπί έχει στο Array έναν τίτλο (0, 1, 2, 3, 4, 5, 6, 7) που αντιστοιχεί σε κάποια πόλη. Γίνεται σαφές ποια πόλη έχει επιλέξει ο χρήστης με βάση το νούμερο που αυτή καταλαμβάνει στο Array (0, 1, 2, 3, 4, 5, 6, 7).]

}

[actionSheet setCancelButtonIndex: [self.ASButtonTitles count] -1];

[Στην τελευταία θέση του πίνακα υπάρχει το κουμπί Cancel.]

[actionSheet showInView:self.view];

}

- (IBAction)selectDestinationButtonPressed:(id)sender

[Στο σημείο αυτό, καλείται το actionSheet για τον προορισμό.]

{

self.ASButtonTitles = [[NSMutableArray alloc] initWithArray:self.locationNamesArray];

[self.ASButtonTitles addObject:@“Cancel"];

UIActionSheet *actionSheet = [[UIActionSheet alloc] initWithTitle:nil

delegate:self

cancelButtonTitle:nil

destructiveButtonTitle:nil

otherButtonTitles:nil];

```
    actionSheet.tag = kDestinationActionSheetTag;
    for (NSString *title in self.ASButtonTitles)
    {
        [actionSheet addButtonWithTitle:title];
    }
    [actionSheet setCancelButtonIndex: [self.ASButtonTitles count] -1];
    [actionSheet showInView:self.view];
}

- (IBAction)nextBtnPressed:(id)sender
{
    NSArray *path;
    if ([self.selectStartingPointLabel.text length] > 0 &&
       [self.selectDestinationLabel.text length] > 0)
```

[Ελέγχεται αν ο χρήστης έχει επιλέξει αφετηρία και προορισμό.]

```
    {
        path = [self calculateBestPath];
    }
    else
    {
```

UIAlertView *alert = [[UIAlertView alloc] initWithTitle:nil message:@"Please compleate all field first" delegate:self cancelButtonTitle:@"Ok" otherButtonTitles:nil];

[Αν ο χρήστης δεν έχει συμπληρώσει την αφετηρία και τον προορισμό θα εμφανιστεί στην οθόνη μια προειδοποίηση.]

```
        [alert show];
        return;
    }
    SecondViewController *secondVC = [[SecondViewController alloc] init];
    secondVC.path = path;
    [self.navigationController pushViewController:secondVC animated:YES];
}
-(void)actionSheet:(UIActionSheet *)actionSheet clickedButtonAtIndex:
(NSInteger)buttonIndex
{
    if (actionSheet.tag == kStartingPointActionSheetTag)
    {
        if (buttonIndex == actionSheet.cancelButtonIndex)
        {
        }
        else
        {
```

```
            NSString *pressedTitle = [self.ASButtonTitles objectAtIndex:buttonIndex];
            self.selectStartingPointLabel.text = pressedTitle;
```

[Το buttonIndex θα δείξει ποιο κουμπί έχει πατηθεί. Εξετάζουμε ποιος τίτλος είναι επιλεγμένος λαμβάνοντας υπόψη τα εξής: α) Αν το actionSheet αφορά την αφετηρία β) αν η επιλογή είναι το cancel γ) ειδάλλως υποθέτουμε ότι η επιλογή του χρήστη θα είναι ετικέτα. Αν το ButtonIndex είναι το ίδιο με το cancelButtonIndex δεν γίνεται τίποτα, ειδάλλως τοποθετείται το όνομα της πόλης στο Label. Όταν το selectStartingPointLabel.text από το Actionsheet ισούται με το pressedTitle καταλαβαίνουμε ποια πόλη έχει επιλέξει ο χρήστης ως σημείο αφετηρίας.]

```
        }
    }
    else if (actionSheet.tag == kDestinationActionSheetTag)
    {
        if (buttonIndex == actionSheet.cancelButtonIndex)
        {
        }
        else
        {
            NSString *pressedTitle = [self.ASButtonTitles objectAtIndex:buttonIndex];
            self.selectDestinationLabel.text = pressedTitle;
```

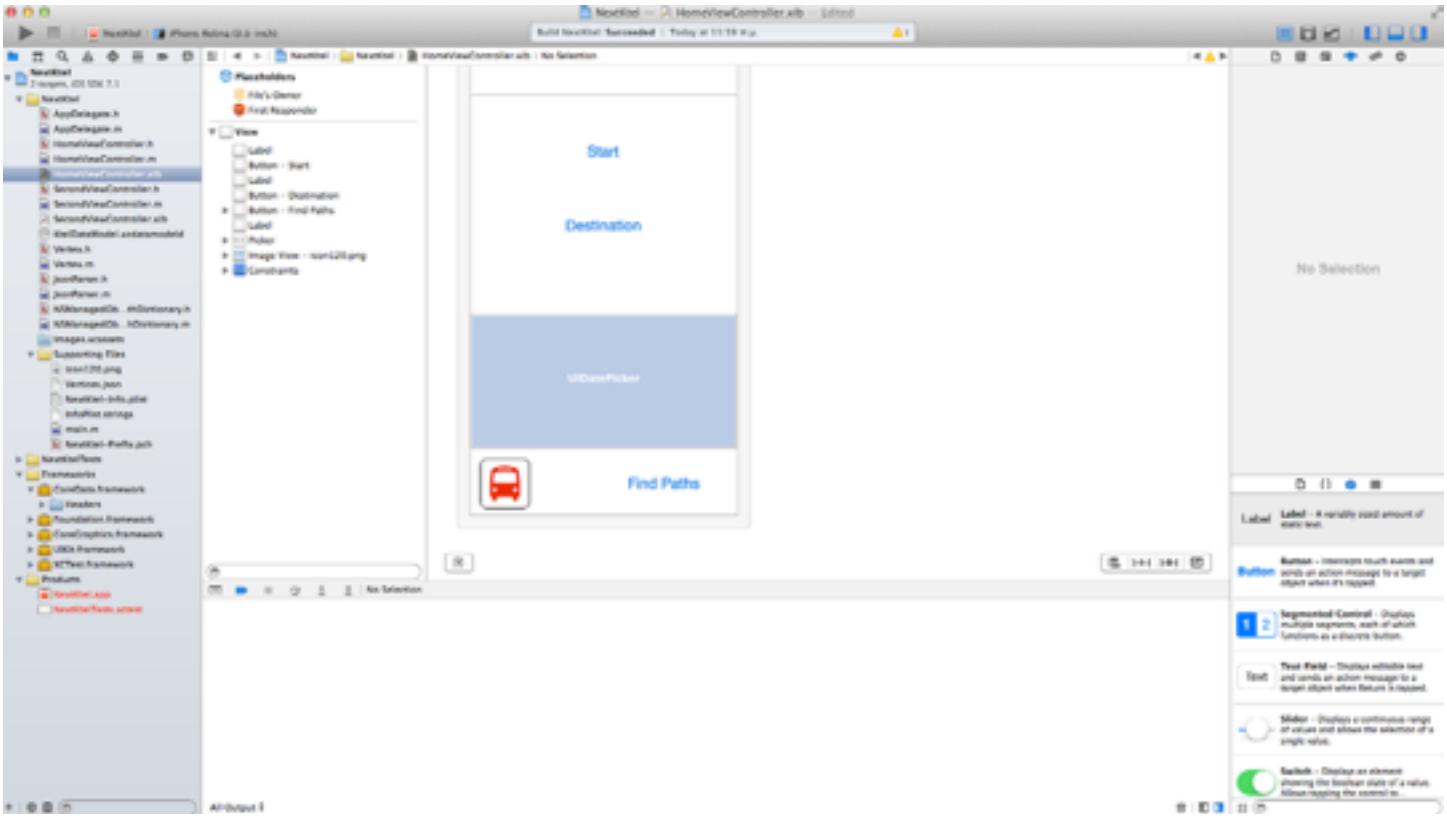
[Όταν το selectDestinationLabel ισούται με το pressedTitle γίνεται αντιληπτό ποια πόλη έχει επιλέξει ο χρήστης ως σημείο προορισμού.]

```
    }
```

```
    }

}

@end
```

Αρχείο HomeViewController.xib

Στο HomeViewController.xib αρχείο δημιουργείται το πρώτο view της εφαρμογής Nextktel.



Κώδικας αρχείου SecondViewController.h

```
//

//  SecondViewController.h

//  NextKtel

//

//  Created by Georgia on 17/11/13.

//  Copyright (c) 2013 Georgia. All rights reserved.

//


#import <UIKit/UIKit.h>


@interface SecondViewController : UIViewController
```

[Με τον SecondViewController μεταφερόμαστε στο δεύτερο full-screen view της εφαρμογής.]

```
@property NSArray *path;
```

[Δηλώνεται το μονοπάτι, δηλαδή η διαδρομή που έχει επιλέξει ο χρήστης από την αφετηρία μέχρι τον προορισμό.]

```
@property (weak, nonatomic) IBOutlet UITextView *textView;


@end
```

Κώδικας αρχείου SecondViewController.m

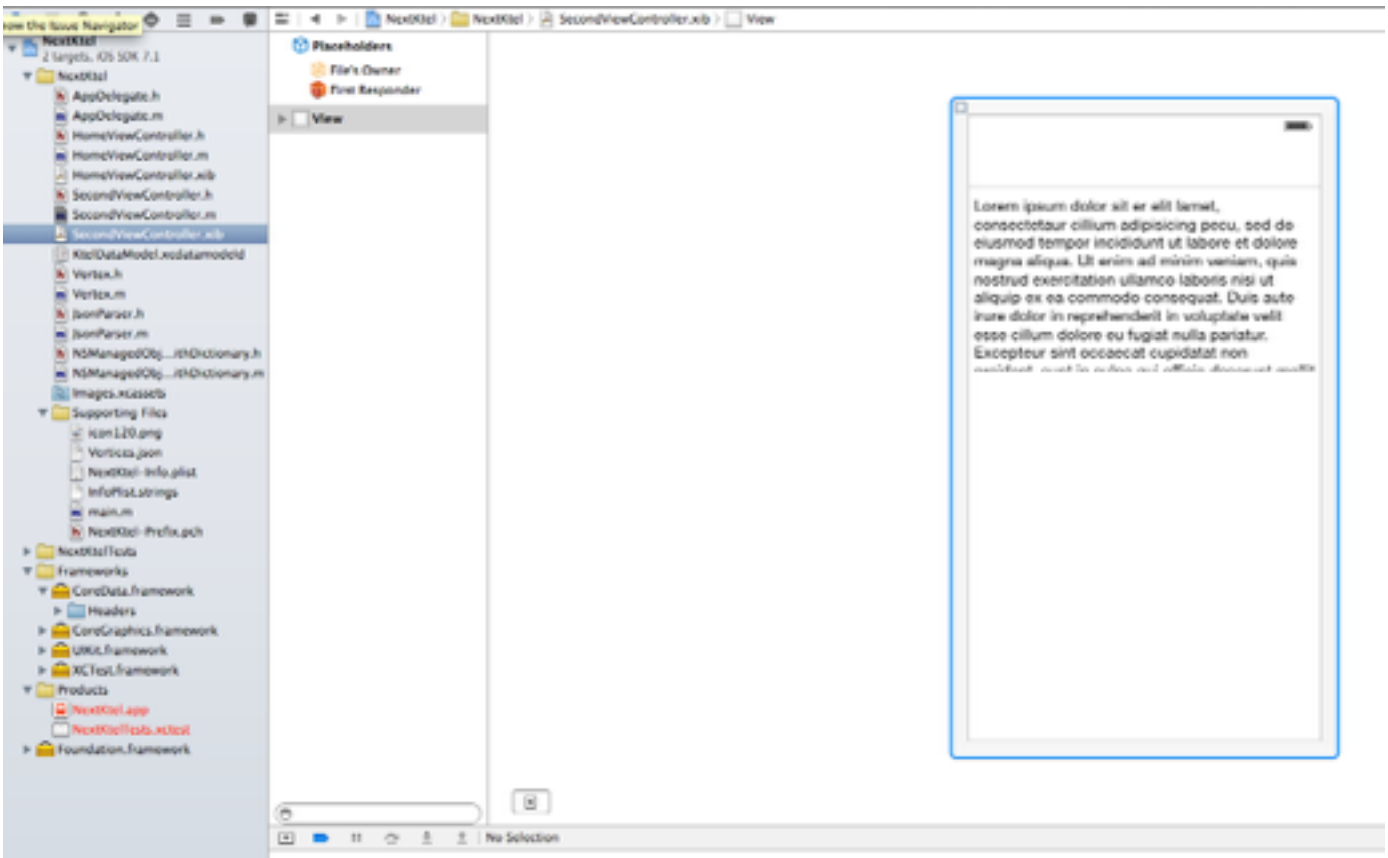
```
//

//  SecondViewController.m

//  NextKtel

//

//  Created by Georgia on 20/11/13.

//  Copyright (c) 2013 Georgia. All rights reserved.

//

#import "SecondViewController.h"

#import “Vertex.h"



@interface SecondViewController ()

@end



@implementation SecondViewController

- (void)viewDidLoad

{

    [super viewDidLoad];

    // Do any additional setup after loading the view from its nib.

        NSDateFormatter *dateFormatter = [[NSDateFormatter alloc] init];

         [dateFormatter setDateFormat:@"dd/MM/yyyy HH:mm"];
```

```
    [dateFormatter setTimeZone:[NSTimeZone localTimeZone]];


    NSMutableString *finalPathString = [[NSMutableString alloc] init];
    for (Vertex *vert in self.path)
    {
        [finalPathString appendString:vert.from];
       [finalPathString appendString:@"->"];
        [finalPathString appendString:vert.to];
        [finalPathString appendString:@" Start:"];
        [finalPathString appendString:[dateFormatter stringFromDate:vert.dateStart]];
        [finalPathString appendString:@"  Arrival:"];
        [finalPathString appendString:[dateFormatter stringFromDate:vert.dateEnd]];
        if ([self.path lastObject] != vert)
        {
            [finalPathString appendString:@"---->"];
        }
    }
    self.textView.text = finalPathString;
}
- (void)didReceiveMemoryWarning
{
```

```
    [super didReceiveMemoryWarning];

    // Dispose of any resources that can be recreated.

}


@end
```

SecondViewController.xib

Στο SecondViewController.xib αρχείο δημιουργείται το δεύτερο view της εφαρμογής Nextktel.



Κώδικας αρχείου Vertex.h

```
//

//  Vertex.h

//  NextKtel

//

//  Created by Georgia on 26/1/14.

//  Copyright (c) 2014 Georgia. All rights reserved.

//

#import <Foundation/Foundation.h>

#import <CoreData/CoreData.h>



@interface Vertex : NSManagedObject

@property (nonatomic, retain) NSString * to;

@property (nonatomic, retain) NSString * from;

@property (nonatomic, retain) NSDate * dateEnd;

@property (nonatomic, retain) NSDate * dateStart;



@end
```

[Δηλώνουμε την κλάση Vertex, την οποία θα γεμίσουμε με τα αντικείμενα - διανύσματα του Json αρχείου.]

Κώδικας αρχείου Vertex.m

```
//

//  Vertex.m

//  NextKtel

//

//  Created by Georgia on 26/1/14.

//  Copyright (c) 2014 Georgia. All rights reserved.

//

#import "Vertex.h"

@implementation Vertex

@dynamic to;

@dynamic from;

@dynamic dateEnd;

@dynamic dateStart;



@end
```

Κώδικας αρχείου JsonParser.h

```
//

//  JsonParser.h

//  NextKtel
```

```
//
//  Created by Georgia on 1/3/14.
//  Copyright (c) 2014 Georgia. All rights reserved.
//
#import <Foundation/Foundation.h>
@interface JsonParser : NSObject
-(NSMutableArray *)createVertesesFromJsonData:(NSData *)jsonData;
-(NSData *)downlaodDataFromServer;


@end
```

Κώδικας αρχείου JsonParser.m[74]

```
//
//  JsonParser.m
//  NextKtel
//
//  Created by Georgia on 1/3/14.
```

---

[74] Για την δημιουργία του αρχείου JsonParser τα δεδομένα αντλήθηκαν από τις παρακάτω πηγές:

Chris, Ching, “How To USE iOS NSURLConnection By Example”, (3/1/2013). Βλ. http://codewithchris.com/tutorial-how-to-use-ios-nsurlconnection-by-example/

Επίσης, βλ. http://stackoverflow.com/questions/18549639/nsdateformatter-gives-me-back-greenwich-time

http://stackoverflow.com/questions/18923675/nsurlrequest-cache-issue-ios-7

```
//  Copyright (c) 2014 Georgia. All rights reserved.
//
#import "JsonParser.h"
#import "Vertex.h"
#import "AppDelegate.h"
#import "NSManagedObject+safeSetValuesKeysWithDictionary.h"
@implementation JsonParser
-(NSData *)downlaodDataFromServer
```

[Δημιουργούμε την παραπάνω συνάρτηση για να κατεβαίνουν τα δεδομένα απ' τον Server.]

```
{
    NSString *str = @"http://webapplication37356.azurewebsites.net/vertices.html";
```

[Το String είναι το URL, δηλαδή η διεύθυνση που βρίσκεται το αρχείο στον Server.]

```
    NSURL *url = [NSURL URLWithString:str];
     NSURLRequest *request = [NSURLRequest requestWithURL:url
                                        cachePolicy:NSURLCacheStorageNotAllowed
timeoutInterval:60];[75]
    NSURLResponse * response = nil;
     NSError * error = nil;
```

[75]http://stackoverflow.com/questions/18923675/nsurlrequest-cache-issue-ios-7

NSData * data = [NSURLConnection sendSynchronousRequest:request

returningResponse:&response

error:&error];

[Η αίτηση (Request) πραγματοποιείται σύγχρονα, γιατί το πρόγραμμα παγώνει όταν καλείται η εν λόγω συνάρτηση και περιμένει να επιστρέψει το αποτέλεσμα.[76]]

if (error!= nil)

[Αν υπάρχει error, το πρόγραμμα παίρνει τα δεδομένα τοπικά από το αρχείο json, δηλαδή από τη μνήμη του κινητού.]

{

NSString *verticesJsonPath = [[NSBundle mainBundle] pathForResource:@"Vertices" ofType:@"json"];

NSData *jsonData = [NSData dataWithContentsOfFile:verticesJsonPath];

data = jsonData;

}

return data;

}

-(NSMutableArray *)createVertesesFromJsonData:(NSData *)jsonData

[76]Βλ.Chris, Ching, “How To USE iOS NSURLConnection By Example” (3/1/2013). Βλ. http://codewithchris.com/tutorial-how-to-use-ios-nsurlconnection-by-example/

{

NSError *error;

NSArray *jsonArray = [NSJSONSerialization JSONObjectWithData:jsonData

options:kNilOptions error:&error];

[Μετατρέπουμε τα jsonData σε jsonArray.]

NSLog(@"JsonError:%@", error);

NSDateFormatter *dateFormatter = [[NSDateFormatter alloc] init];

[dateFormatter setDateFormat:@"dd/MM/yyyy HH:mm"];

[dateFormatter setTimeZone:[NSTimeZone localTimeZone]];

NSMutableArray *vertecesArray = [[NSMutableArray alloc] init];

for (NSDictionary *jsonDic in jsonArray)

{

NSManagedObjectContext *nsManagedObjectContext =

((AppDelegate *)[UIApplication sharedApplication].delegate).managedObjectContext;

Vertex *vertex = [[Vertex alloc] initWithEntity:[NSEntityDescription

entityForName:@“Vertex"

inManagedObjectContext:nsManagedObjectContext]

insertIntoManagedObjectContext:nsManagedObjectContext];

[Δημιουργούμε ένα αντικείμενο vertex το οποίο είναι υποκλάση του NSManagedObject.]

[vertex safeSetValuesForKeysWithDictionary:jsonDic dateFormatter:dateFormatter];

```
        [vertecesArray addObject:vertex];


    }

    return vertecesArray;

}


@end
```

Κώδικας αρχείου NSManagedObject+safeSetValuesKeysWithDictionary.h

```
//

//  NSManagedObject+safeSetValuesKeysWithDictionary.h

//  NextKtel

//

//  Created by Georgia on 1/3/14.

//  Copyright (c) 2014 Georgia. All rights reserved.

//

#import <CoreData/CoreData.h>

@interface NSManagedObject (safeSetValuesKeysWithDictionary)

- (void)safeSetValuesForKeysWithDictionary:(NSDictionary *)keyedValues dateFormatter:

(NSDateFormatter *)dateFormatter;

@end
```

Κώδικας αρχείου NSManagedObject +safeSetValuesKeysWithDictionary.m[77]

```
//
//  NSManagedObject+safeSetValuesKeysWithDictionary.m
//  NextKtel
//
//  Created by Georgia on 1/3/14.
//  Copyright (c) 2014 Georgia. All rights reserved.
//


#import "NSManagedObject+safeSetValuesKeysWithDictionary.h"


@implementation NSManagedObject (safeSetValuesKeysWithDictionary)

- (void)safeSetValuesForKeysWithDictionary:(NSDictionary *)keyedValues dateFormatter:
(NSDateFormatter *)dateFormatter
```

---

[77] Ο κώδικας του αρχείου NSManagedObject+safeSetValuesKeysWithDictionary.m αντλήθηκε από τις παρακάτω πηγές:
https://github.com/vdaubry/JSON-to-CoreData/blob/master/NSManagedObject%2BsafeSetValuesForKeysWithDictionary.m
Επίσης, βλ.Tom, Harrington, “Saving Json to Core Data”, (2/6/2011) .
http://www.cimgf.com/2011/06/02/saving-json-to-core-data/

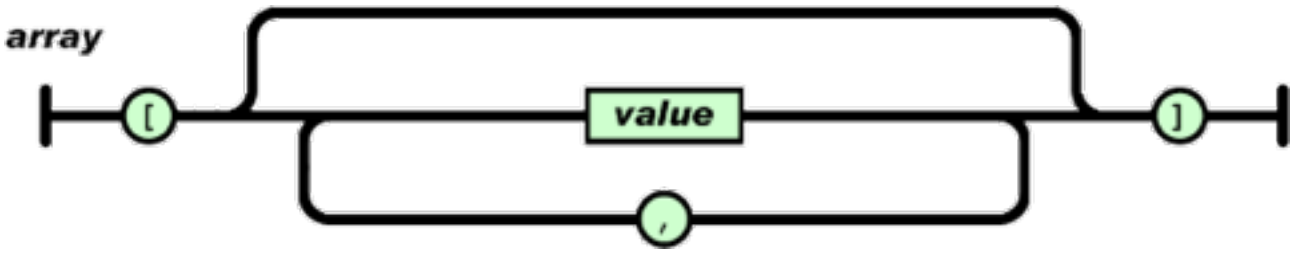
```
{
    NSDictionary *attributes = [[self entity] attributesByName];
    for (NSString *attribute in attributes)
    {
        id value = [keyedValues objectForKey:attribute];
        if (value == nil)
        {
            continue;
        }
        NSAttributeType attributeType = [[attributes objectForKey:attribute] attributeType];
        if ((attributeType == NSStringAttributeType) && ([value isKindOfClass:[NSNumber
class]]))
        {
            value = [value stringValue];
        }
        else if (((attributeType == NSInteger16AttributeType) ||
                  (attributeType == NSInteger32AttributeType) ||
                  (attributeType == NSInteger64AttributeType) ||
                  (attributeType == NSBooleanAttributeType)) && ([value isKindOfClass:[NSString
class]]))
        {
```

```
        value = [NSNumber numberWithInteger:[value integerValue]];
    }
    else if ((attributeType == NSFloatAttributeType) &&  ([value isKindOfClass:[NSString
class]]))
    {
        value = [NSNumber numberWithDouble:[value doubleValue]];
    }
    else if ((attributeType == NSDateAttributeType) && ([value isKindOfClass:[NSString
class]]) && (dateFormatter != nil))
    {
        value = [dateFormatter dateFromString:value];
    }
    [self setValue:value forKey:attribute];
  }
}

@end
```

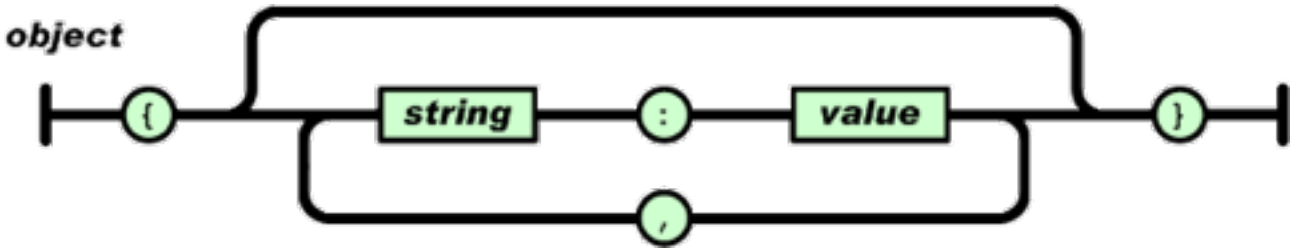
Json αρχείο

Το Json αρχείο περιλαμβάνει όλα τα δρομολόγια με τη μορφή [{“string” : “value”}].

Πιο αναλυτικά αναφέρονται τα εξής:



Σύμφωνα με το παραπάνω σχηματικό, το value αντιπροσωπεύει ένα object που αποτελείται από επαναλαμβανόμενες ακολουθίες strings και values.[78]



Εικόνες 32, 33- Μορφή του Json αρχείου στην εφαρμογή Nextktel. Βλ.http://json.org

[78] Βλ. http://json.org

Ενδεικτικά, παρατίθενται ορισμένα vertices από το Json αρχείο της εφαρμογής Nextktel:


```
[
{
"from": "Nafplio",
"to": "Is8mos",
"dateStart": "01/01/2014 05:30 GMT",
"dateEnd": "01/01/2014 06:30 GMT"
},
{
"from": "Nafplio",
"to": "Is8mos",
"dateStart": "01/01/2014 07:00 GMT",
"dateEnd": "01/01/2014 08:00 GMT"
},
{
"from": "Nafplio",
"to": "Is8mos",
"dateStart": "01/01/2014 08:00 GMT",
"dateEnd": "01/01/2014 09:00 GMT"
}]
```

Server

Μόλις εισάγουμε σε ένα πρόγραμμα περιήγησης διαδικτύου (browser) το URL: http://webapplication37356.azurewebsites.net/vertices.html θα εμφανιστούν όλα τα περιεχόμενα του Json αρχείου.

Ενδεικτικά, παραθέτουμε κάποια από τα περιεχόμενα του Json αρχείου που εμφανίζονται μόλις εισάγουμε το URL:

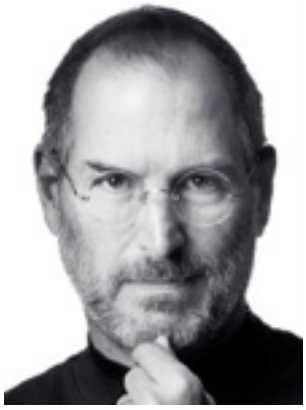

```
[
  .
    {"dateEnd":
    "01/01/2014 06:30 GMT",

    "dateStart":
    "01/01/2014 05:30 GMT",

    "from":
    "Nafplio",

    "to":
    "Is8mos"


    },
  .
    {"dateEnd":
    "01/01/2014 08:00 GMT",

    "dateStart":
    "01/01/2014 07:00 GMT",

    "from":
    "Nafplio",

    "to":
    "Is8mos"


    },
  .
    {"dateEnd":
    "01/01/2014 09:00 GMT",

    "dateStart":
```

```
        "01/01/2014 08:00 GMT",

        "from":
        "Nafplio",

        "to":
        "Is8mos"


        },
    .
        {"dateEnd":
        "01/01/2014 11:00 GMT",

        "dateStart":
        "01/01/2014 10:00 GMT",

        "from":
        "Nafplio",

        "to":
        "Is8mos"


        },
    .
        {"dateEnd":
        "01/01/2014 12:00 GMT",

        "dateStart":
        "01/01/2014 11:00 GMT",

        "from":
        "Nafplio",

        "to":
        "Is8mos"


        }.
]
```

17. Συμπεράσματα

Η χρησιμότητα της εν λόγω εφαρμογής έγκειται στο να διευκολύνει τις μετακινήσεις των επιβατών του ΚΤΕΛ, παρέχοντάς τους πληροφορίες για τα δρομολόγια. Η σχεδίαση και η υλοποίησή της με βοήθησε να διευρύνω τις γνώσεις μου και να εξελίξω τις δυνατότητές μου στη συγγραφή iOS κώδικα και στην δημιουργία Server. Παράλληλα, με βοήθησε να κατανοήσω τον τρόπο λειτουργίας του Αλγόριθμου Αναζήτησης κατά Βάθος και γενικότερα να εντρυφήσω στην Objective C με την οποία θα ήθελα να ασχοληθώ επαγγελματικά στο μέλλον.

18. Επίλογος



Εικόνα 34- Steve Jobs (1955 -2011) Πηγή:http://www.amazon.com/Steve-Jobs-Walter-Isaacson-ebook/dp/B004W2UBYW

Η διαφήμιση της Apple «Think Different» (1997) προέβαλλε το παρακάτω μότο: «Εκείνοι που είναι τόσο τρελοί ώστε να πιστεύουν ότι μπορούν να αλλάξουν τον κόσμο είναι εκείνοι που τον αλλάζουν».[79] Στο ίδιο θεματικό μοτίβο κινήθηκε ο Steve Jobs στην τελετή αποφοίτησης των φοιτητών του Πανεπιστήμιου του Stanford εκφράζοντας την παραίνεση «Stay hungry. Stay foolish».[80]

Αναντίρρητα, ο Προγραμματισμός αποτελεί πρόσφορο έδαφος για να ακονίσουμε το μυαλό μας και να πραγματοποιήσουμε τα πιο τρελά όνειρά μας, αρκεί να τολμήσουμε, να πιστέψουμε στις δυνατότητές μας και να παλεύσουμε για να υλοποιήσουμε τους στόχους μας. Είναι γεγονός ότι η

[79] Walter Isaacson, *Steve Jobs του Walter Issacson, Η επίσημη βιογραφία*, Εκδ. Ψυχογιός, 2013, 15η εκδ., σ. οπισθόφυλλο.

[80] Βλ.http://news.stanford.edu/news/2005/june15/jobs-061505.html

Apple με την τεχνολογία της άλλαξε τον κόσμο, καιρός να αλλάξουμε και εμείς τον κόσμο και την καθημερινότητά μας μέσα από καινοτόμες εφαρμογές.

Επιλογή Βιβλιογραφίας

- http://azure.microsoft.com/en-us/

- http://azure.microsoft.com/en-us/documentation/articles/web-sites-dotnet-get-started/

- Ankit, Aggarwal, “iOS Core Data Tutorial With Example”, (30/7/2013).

Βλ. http://www.codigator.com/tutorials/ios-core-data-tutorial-with-example/

- Απόστολος, Ανδριώτης, «Σύστημα Ανάγνωσης - Διαχείρισης Ηλεκτρονικών Βιβλίων», Πτυχιακή Εργασία, Αλεξάνδρειο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Θεσσαλονίκης, Σχολή Τεχνολογικών Εφαρμογών Τμήμα Πληροφορικής, Θεσσαλονίκη 2007.

- Gary, Bennett, Mitch Fisher, Bred Lees, *Objective-C for Absolute Beginners iPhone, iPad, and Mac Programming Made Easy*, Apress, second ed., New York 2011.

- Garling, Caleb, "iPhone Coding Language Now World’s Third Most Popular", (07/09/2012).

Βλ.http://www.wired.com/2012/07/apple-objective-c/

- Elizabeth Castro, Bruce Hyslop, *HTML5 και CSS3*, μτφ. Φώτης Σκουλαρίκης, Κλειδάριθμος, έβδομη αμερικανική έκδοση, 2013.

- Chris, Ching, “How To USE iOS NSURLConnection By Example” (3/1/2013). Βλ. http://codewithchris.com/tutorial-how-to-use-ios-nsurlconnection-by-example/

- Joe, Conway, Aaron Hillegass, *iPhone Programming: The Big Nerd Ranch Guide,* Big Nerd Ranch, Inc., Atlanta 2010.

- http://creativebits.org/interview/interview_rob_janoff_designer_apple_logo
- Κώστας, Δεληγιάννης, «Το “έξυπνο” σπίτι λανσάρισε η Apple», *Καθημερινή* (03/06/2014). Βλ. http://www.kathimerini.gr/770094/article/texnologia/thlefwnia/to-e3ypno-spiti-lansarise-h-apple
- https://developer.apple.com/library/ios/documentation/WindowsViews/Conceptual/ViewControllerCatalog/Chapters/NavigationControllers.html

- https://developer.apple.com/library/ios/featuredarticles/ViewControllerPGforiPhoneOS/Introduction/Introduction.html

-“What makes an app a good app - 7 pointers”, (14/4/2014).

Βλ.http://dotnet.dzone.com/articles/what-makes-app-good-app-10

- http://en.wikipedia.org/wiki/IPhone_OS#iPhone_SDK

- http://en.wikipedia.org/wiki/Objective-C

- http://freddesign.co.uk/so-why-an-apple-the-history-of-the-apple-logo/

- https://github.com/vdaubry/JSON-to-CoreData/blob/master/NSManagedObject%2BsafeSetValuesForKeysWithDictionary.m

- Tom, Harrington, “Saving Json to Core Data”, (2/6/2011).

Βλ. http://www.cimgf.com/2011/06/02/saving-json-to-core-data/

-Aaron, Hillegass, *Objective-C Programming: The Big Nerd Ranch Guide,* Big Nerd Ranch, Inc., Atlanta 2011.

-Walter Isaacson, *Steve Jobs του Walter Issacson, Η επίσημη βιογραφία*, Εκδ. Ψυχογιός, 15η εκδ., Αθήνα 2013.

-ITL Education Solutions Limited (ITL ESL), *Introduction to Computer Science,* Pearson Education, second ed., India 2011. Βλ. http://my.safaribooksonline.com/book/electrical-engineering/computer-engineering/9788131760307/computer-program/section_8.10

-Bart, Jacobs, “First Steps with UIKit”, (17/12/2012). Βλ. http://code.tutsplus.com/tutorials/first-steps-with-uikit--mobile-14013

- http://json.org

- Αλέξανδρος, Κουτσονάσιος, «Ανάπτυξη και ανάκληση διαδραστικών ξεναγήσεων με φορητές συσκευές», Διπλωματική Εργασία, Τμήμα Ηλεκτρολόγων και Μηχανικών Υπολογιστών, Αριστοτέλειο Πανεπιστήμιο Θεσσαλλονίκης, Θεσσαλονίκη 2010.

- Anuar Lezama, “Introduction to the mobile application development with an example of a PhraseBook app”, Master Thesis, Universitat Politècnica de Catalunya, Barcelona 2010.

- Dave Mark, Jack Nutting, Jeff LaMarche, *Beginning iPhone 4 Developement, Exploring the iOS SDK*, Apress, United States of America 2011.

- Δημήτρης, Μιχαήλ, Σημειώσεις για το μάθημα «Αλγόριθμοι και Πολυπλοκότητα», Τμήμα Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο, σ.26. Βλ. http://dm.hua.gr/michail/teaching/algs/slides/030_Graphs.pdf

- http://news.stanford.edu/news/2005/june15/jobs-061505.html

- Νίκος, Νικολάου, Σημειώσεις για το μάθημα «Κατανεμημένοι Αλγόριθμοι», Διάλεξη 3: «Αλόριθμοι σε Γράφους ΙΙ». Βλ. http://www.cs.ucy.ac.cy/~nicolasn/epl432/Lectures/03-graph-algorithms-II.pdf

- Αναστάσιος, Πιοτόπουλος, «Δημιουργία mobile application σε iOS (iPhone/iPad) για την οργάνωση οικιακών εξόδων, ακολουθώντας πρακτικά πρότυπα προγραμματισμού», Πτυχιακή Εργασία, Τμήμα Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών, Λάρισα 2011.

- Τζανέτος, Πομόνης, «Προς το Web 3.0: Διαδικασία Ανάπτυξης και Αρχιτεκτονική Υποστήριξης εφαρμογών Παγκόσμιου Ιστού που συνδυάζουν τεχνολογίες Web 2.0 και Semantic Web.», Διδακτορική Διατριβή, Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής, Πανεπιστήμιο Πατρών, Πάτρα 2010.

- Ηλίας, Κ. Σάββας, Σημειώσεις για το μάθημα «Αλγόριθμοι και Πολυπλοκότητα», Τμήμα Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών, Λάρισα Ιανουάριος 2005, σ. 94. Βλ.http://www.teilar.gr/dbData/ProfAnn/profann-2a82b2bb.pdf

- http://stackoverflow.com/questions/11971945/concatenate-string-in-string-objective-c

- http://stackoverflow.com/questions/18923675/nsurlrequest-cache-issue-ios-7

- http://stackoverflow.com/questions/5965044/how-to-compare-two-nsdates-which-is-more-recent

- http://stackoverflow.com/questions/21303892/loading-a-property-list-with-a-universal-file-path

- Αντώνιος Συμβώνης, Κατερίνα Ποτίκα, Διαλέξεις 1 και 4 του μαθήματος «Αλγόριθμοι και Πολυπλοκότητα». Βλ. https://semfe.gr/files/users/477/algorithmoi_dialekseis_1-8.pdf

- Ray, Wenderlich, "Core Data on iOS 5 Tutorial: How To Preload and Import Existing Data", (23/10/2012). Βλ. http://www.raywenderlich.com/forums/viewtopic.php?f=20&t=3241&start=40

- Ray, Wenderlich, "Working with JSON in iOS 5 Tutorial", (28/10/2011) Βλ. http://www.raywenderlich.com/5492/working-with-json-in-ios-5

- http://www.apple.com/gr/iphone-5c/features/

- http://www.apple.com/gr/iphone-5s/features/

- «Εισαγωγή στην Ανάπτυξη Android Εφαρμογών» του Εργαστηρίου Τεχνολογίας Πολυμέσων του Ε.Μ.Π., Digital Academy, σ. 5, 6. Βλ.http://www.dga.gr/web/publications/files/android.pdf

- http://www.e-pcmag.gr/news/i-apple-parousiase-neo-ios-8-gia-iphone-ipad-kai-ipod-touch

- http://www.e-pcmag.gr/news/neo-mac-os-x-yosemite-kai-oi-upiresies-icloud-drive-maildrop-kai-airdrop

- http://www.fortunegreece.com/photo-gallery/istoria-tou-iphone/#2

- http://www.fortunegreece.com/photo-gallery/istoria-tou-iphone/#3

- http://www.fortunegreece.com/photo-gallery/istoria-tou-iphone/#4

- www.fortunegreece.com/photo-gallery/istoria-tou-iphone/#5

- http://www.fortunegreece.com/photo-gallery/istoria-tou-iphone/#7

- http://www.fortunegreece.com/photo-gallery/istoria-tou-iphone/#8

- www.fortunegreece.com/photo-gallery/istoria-tou-iphone/#9

- Σημειώσεις για το Εργαστήριο «Παράλληλης Κατανεμημένης Επεξεργασίας», Τμήμα Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας. Βλ. http://www.it.uom.gr/project/client_server/theoria1.htm#%C7

- http://stackoverflow.com/questions/18549639/nsdateformatter-gives-me-back-greenwich-time

- www.stackoverflow.com/questions/19774738/coredata-managedobjectcontext-not-being-created

- http://www.tiobe.com/index.php/content/paperinfo/tpci/tpci_definition.htm

- http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

Επιλογή Βιβλιογραφίας Φωτογραφικού Υλικού

- Αναστάσιος, Πιοτόπουλος, «Δημιουργία mobile application σε iOS (iPhone/iPad) για την οργάνωση οικιακών εξόδων, ακολουθώντας πρακτικά πρότυπα προγραμματισμού», Πτυχιακή Εργασία, Τμήμα Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών, Λάρισα 2011.

- Anuar, Lezama, "Introduction to the mobile application development with an example of a PhraseBook app", Master Thesis,Universitat Politècnica de Catalunya, Barcelona 2010.

Apple maps

- http://creativebits.org/interview/interview_rob_janoff_designer_apple_logo

- https://developer.apple.com/xcode/

- google.com (images (Apple))

- http://en.wikipedia.org/wiki/TIOBE_index#mediaviewer/File:Tiobe_index.png

- http://freddesign.co.uk/so-why-an-apple-the-history-of-the-apple-logo/

- http://json.org

- http://poincare.matf.bg.ac.rs/~andjelkaz/pzv/cas4/mvc.pdf

- http://www.apple.com/gr/iphone-5c/features/

- http://www.apple.com/gr/iphone-5s/features/

- http://www.apple.com/gr/mac-mini/

- http://www.amazon.com/Steve-Jobs-Walter-Isaacson-ebook/dp/B004W2UBYW

- http://www.brandchannel.com/home/post/steve-jobs-evolution-apple-logo.aspx

- http://www.clipartbest.com/bus-stop-logo (το εικονίδιο της εφαρμογής)

- http://www.e-pcmag.gr/news/neo-mac-os-x-yosemite-kai-oi-upiresies-icloud-drive-maildrop-kai-airdrop

- http://www.iphoner.gr/άρθρο/

- http://www.techgear.gr/android-vs-ios-software-support-81383/

- http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html