

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

**ΤΜΗΜΑ  
ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ  
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**



**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Ανάπτυξη αλγορίθμου καταμέτρησης παραμέτρων άθλησης και  
υλοποίησή του σε αρχιτεκτονική iOS**

Κεφάλαια Αικατερίνη

Επιβλέπων καθηγητής: Αγγελίδης Παντελής

Κοζάνη, Οκτώβριος 2015



Copyright (C) Κεφάλαια Αικατερίνη, 2015, Κοζάνη

#### Δήλωση Πνευματικών Δικαιωμάτων

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα εργασία αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.



## Ευχαριστίες

Πρωτίστως θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Αγγελίδη Παντελή για την ευκαιρία που μου έδωσε, για την ανάθεση της εργασίας, την εμπιστοσύνη και τη βοήθεια που μου προσέφερε για την υλοποίησή της.

Επίσης θα ήθελα να ευχαριστήσω την οικογένειά μου για την αμέριστη υποστήριξη και συμπαράσταση κατά τη διάρκεια των σπουδών μου και ιδιαίτερος την αδερφή μου.

Τέλος, θα ήθελα να ευχαριστήσω τους φίλους μου και τους συμφοιτητές μου για τη συμπαράσταση κατά τη διάρκεια όλων αυτών των χρόνων.

# Περίληψη

Τα έξυπνα τηλέφωνα, γνωστά ως smartphones έχουν γίνει πλέον κομμάτι της ζωής μας και με την πάροδο του χρόνου όλο και περισσότεροι άνθρωποι τα χρησιμοποιούν για επαγγελματική και φυσικά για προσωπική χρήση. Οι κατηγορίες των έξυπνων τηλεφώνων είναι πολλές και οι εφαρμογές για τα τηλέφωνα αυτά ακόμα περισσότερες.

Για την υλοποίηση της παρούσας διπλωματικής απαιτήθηκε ιδιαίτερη μελέτη και προσπάθεια, διάβασμα πολλών βιβλίων αλλά και διαδικτυακών πηγών, καθώς πρώτη φορά κλήθηκα να ασχοληθώ με μια εφαρμογή στο περιβάλλον Mac OS X, χρησιμοποιώντας το πρόγραμμα Xcode και τη νέα γλώσσα προγραμματισμού Swift. Επομένως ο χρόνος που χρειάστηκε να ασχοληθώ με τη διπλωματική ήταν αυξημένος.

Η mobile εφαρμογή που υλοποιήθηκε στα πλαίσια της διπλωματικής εργασίας αφορά το λειτουργικό σύστημα iOS, ανήκει στην κατηγορία Health and Fitness και έχει δυο βασικούς άξονες. Ο πρώτος αφορά κυρίως την καταμέτρηση των βημάτων του χρήστη και το είδος των βημάτων (περπάτημα, τρέξιμο) και ο δεύτερος αφορά την καταχώρηση του εβδομαδιαίου προγράμματος γυμναστηρίου με τις αντίστοιχες λεπτομέρειες.

Σκοπός της εργασίας είναι να γίνει μια σωστή προσέγγιση στον τρόπο που σχεδιάζονται mobile εφαρμογές.

**Λέξεις κλειδιά:** σχεδιασμός εφαρμογής, fitness, steps, θερμίδες, Swift, πρόγραμμα, iOS, iPhone, Xcode, Application, Smartphone.



# Abstract

Smartphones have become part of our lives and over time more and more people use them for professional and of course for personal use. The categories of smart phones are many and the applications for these phones even more.

For the impelentation of this thesis required special study and effort, reading of many books and online resources as well it was the first time that i was asked to deal with an application enviroment in Mac OS X, using the tool Xcode and the new programming language Swift. Therefore, the time taken to deal with the thesis was increased.

The mobile application that was implemented under the thesis concerns the iOS operating system, belongs to the category Health and Fitness and has two main axes. The first mainly concerns the counting of steps of the user and the time of steps (walking, running) and the second relates to the weekly fitness program with corresponding details.

Aim of this study is to make a correct approach to the way that mobile applications are designed.

**Keywords:** application design, fitness, steps, calories, Swift, program, iOS, iPhone, Xcode, application, Smartphone.





# Πίνακας Περιεχομένων

<b>Κεφάλαιο 1</b> .....	<b>1</b>
<b>Εισαγωγή στις κινητές πλατφόρμες</b> .....	<b>1</b>
1.1 Εισαγωγή .....	1
1.2 Αρχιτεκτονική εφαρμογής .....	1
1.3 Κινητή πλατφόρμα.....	2
1.4 Είδη κινητής πλατφόρμας.....	2
1.4.1 Έξυπνα τηλέφωνα (smartphones).....	2
1.4.2 Ταμπλέτες (Tablets) .....	4
1.4.3 Προσωπικός ψηφιακός βοηθός (PDA).....	5
1.5 Λειτουργικά συστήματα για τις κινητές πλατφόρμες .....	7
1.6 Τα πρώτα λειτουργικά συστήματα για τις κινητές πλατφόρμες .....	7
1.6.1 Palm OS.....	7
1.6.2 Windows Mobile.....	8
1.7 Τωρινά λειτουργικά συστήματα για κινητές πλατφόρμες .....	9
1.7.1 Android .....	10
1.7.2 Windows Phone .....	11
1.7.3 Blackberry OS .....	12
1.7.4 iOS .....	14
1.7.5 Άλλα λειτουργικά συστήματα.....	16
1.8 Λειτουργικά συστήματα για κινητές πλατφόρμες που δε χρησιμοποιούνται πια.....	17
1.9 Εφαρμογές για κινητές πλατφόρμες.....	17
<b>Κεφάλαιο 2</b> .....	<b>19</b>
<b>Υλικό και λογισμικό</b> .....	<b>19</b>
2.1 Υλικό .....	19
2.1.1 Ηλεκτρονικός υπολογιστής .....	19
2.1.2 Συσκευές iPhone, iPod Touch, iPad.....	20
2.2 Το λογισμικό.....	21
2.2.1 Mac OS X.....	21
2.2.2 iOS SDK.....	23
Cocoa Touch .....	23
Media.....	24
Core Services .....	25
Core OS.....	25
2.2.3 Objective C .....	26
2.3.4 Swift .....	27
2.3.5 Xcode .....	28
2.3.6 SQLite.....	29
2.3.7 Core Data.....	30

<b>Κεφάλαιο 3.....</b>	<b>33</b>
<b>Ανάλυση της εφαρμογής .....</b>	<b>33</b>
<b>3.1 Απαιτήσεις σχεδίασης .....</b>	<b>33</b>
<b>3.2 Χρησιμοποιώντας το Xcode.....</b>	<b>34</b>
3.2.1 Πρώτη καρτέλα εφαρμογής .....	40
3.2.2 Δεύτερη καρτέλα εφαρμογής .....	47
3.2.3 Τρίτη καρτέλα εφαρμογής .....	50
<b>Κεφάλαιο 4.....</b>	<b>54</b>
<b>Συμπεράσματα και μελλοντικοί στόχοι.....</b>	<b>54</b>
<b>4.1 Συμπεράσματα.....</b>	<b>54</b>
<b>4.2 Μελλοντικοί στόχοι .....</b>	<b>55</b>
<b>Βιβλιογραφικές αναφορές .....</b>	<b>57</b>

## Κατάλογος εικόνων

Εικόνα 1.1: HTC One Smartphone .....	4
Εικόνα 1.2: Microsoft Surface 2 RT Tablet .....	5
Εικόνα 1.3: Dell Axim X50V PDA.....	6
Εικόνα 1.4: Palm OS .....	8
Εικόνα 1.5: Windows Mobile 6.5.....	9
Εικόνα 1.6: Android 5.0.....	11
Εικόνα 1.7: Windows Phone 8.1 .....	12
Εικόνα 1.8: Blackberry 10 OS .....	13
Εικόνα 1.9: iOS 8.....	16
Εικόνα 2.10: Mac OS 9 .....	22
Εικόνα 2.11: Γλώσσα προγραμματισμού Objective-C .....	26
Εικόνα 2.12: Γλώσσα προγραμματισμού Swift .....	28
Εικόνα 2.13: Εργαλείο προγραμματισμού Xcode.....	29
Εικόνα 2.14: SQLite .....	30
Εικόνα 2.15: Core Data .....	31
Εικόνα 3.16: Δημιουργώντας καινούριο project στο Xcode .....	35
Εικόνα 3.17: :Ονομάζοντας το project και επιλέγοντας τα βασικά χαρακτηριστικά .....	36
Εικόνα 3.18: Workspace window .....	37
Εικόνα 3.19: Προσομοιωτής .....	38
Εικόνα 3.20: Images .....	39
Εικόνα 3.21: Inspector Pane .....	39
Εικόνα 3.22: Library Pane.....	40
Εικόνα 3.23: Καρτέλα για το εβδομαδιαίο πρόγραμμα άθλησης .....	40
Εικόνα 3.24: Καρτέλα με όλες τις ημέρες της εβδομάδας .....	41
Εικόνα 3.25: Εισαγωγή λεπτομερειών προγράμματος για μια μέρα .....	43
Εικόνα 3.26: Δήλωση μεταβλητών για τα κελιά των ημερών.....	44
Εικόνα 3.27: Ορίζοντας το όνομα και την αρίθμηση.....	44
Εικόνα 3.28: Δίνοντας όνομα στο Identifier για το Table View Cell.....	45
Εικόνα 3.29: Χρησιμοποιώντας το Identifier DayTableViewCell.....	45
Εικόνα 3.30: Σύνδεση και δήλωση στοιχείων για λεπτομέρειες του προγράμματος.....	45
Εικόνα 3.31: Χρήση NSUserDefaults για την αποθήκευση μεταβλητών.....	46
Εικόνα 3.32: Αποστολή της ημέρας που επιλέχθηκε από το χρήστη στη δεύτερη οθόνη .....	46
Εικόνα 3.33: Καρτέλα για τις παραμέτρους μέτρησης σε πραγματικό χρόνο.....	47
Εικόνα 3.34: Καρτέλα που δείχνει σε πραγματικό χρόνο τις παραμέτρους των κινήσεων του χρήστη.....	48
Εικόνα 3.35: Εισαγωγή του Framework Core Motion .....	49
Εικόνα 3.36: Δήλωση των CMPedometer και CMAltimeter.....	49
Εικόνα 3.37: Υπολογισμός και αναπαραγωγή στοιχείων μετακίνησης.....	50
Εικόνα 38: Καρτέλα για τα συνολικά στοιχεία μέτρησης της ημέρας .....	50
Εικόνα 3.39: Σχεδίαση της δομής των κελιών στο Main.storyboard .....	51
Εικόνα 3.40: Χρήση της κλάσης CMMotionActivityManager .....	51



## **Διάρθρωση κειμένου**

Στο πρώτο κεφάλαιο γίνεται μια εισαγωγή στο θέμα της διπλωματικής, περιγράφοντας συνοπτικά το ποια είναι η λειτουργία της εφαρμογής, κάποιες γενικές πληροφορίες σχετικά με τις κινητές πλατφόρμες και ποιες οι τεχνολογίες που ήταν απαραίτητες για να υλοποιηθεί.

Στο δεύτερο κεφάλαιο περιγράφονται το υλικό και το λογισμικό που χρησιμοποιούνται για την ανάπτυξη της εφαρμογής.

Στο τρίτο κεφάλαιο παρουσιάζεται η πορεία υλοποίησης της εφαρμογής.

Στο τέταρτο κεφάλαιο αναφέρεται ο επίλογος της εργασίας και αναφέρονται τα συμπεράσματα και οι μελλοντικές επεκτάσεις της εφαρμογής.



# Κεφάλαιο 1

## Εισαγωγή στις κινητές πλατφόρμες

Το κεφάλαιο αυτό αποτελεί την εισαγωγή στο σύστημα της εφαρμογής, καθώς και μια βασική γνωριμία με τις έξυπνες συσκευές, τις πλατφόρμες σε κινητά τηλέφωνα και τα λειτουργικά συστήματα που υπάρχουν γι' αυτές.

### 1.1 Εισαγωγή

Η εφαρμογή αυτή απευθύνεται κυρίως σε ανθρώπους οι οποίοι έχουν την άσκηση ως βασικό μέρος της ζωής τους και η καθημερινότητά τους την περιλαμβάνει. Ο χρήστης μπορεί έχει οργανωμένα στην εφαρμογή αυτή το εβδομαδιαίο του πρόγραμμα και συγχρόνως να ξέρει ανά πάσα στιγμή πόσα βήματα έχει κάνει και με ποιο τρόπο.

Η εφαρμογή αναπτύσσεται στο λειτουργικό σύστημα MAC OS X για τις συσκευές iPhone, οι οποίες χρησιμοποιούν το λειτουργικό σύστημα iOS.

### 1.2 Αρχιτεκτονική εφαρμογής

Η εφαρμογή ακολουθεί το πρότυπο tab bar και περιλαμβάνει 2 βασικές λειτουργίες. Στην πρώτη ο χρήστης αποθηκεύει το εβδομαδιαίο του πρόγραμμα άθλησης και στη δεύτερη γίνεται καταμέτρηση σε πραγματικό χρόνο των βημάτων που εκτελεί, τον τρόπο με τον οποίο τα εκτελεί (εάν περπατάει, εάν τρέχει) και με βάση αυτό το στοιχείο γίνονται διάφορες παραμετροποιήσεις για επιπλέον στοιχεία και καταγραφές.



## 1.3 Κινητή πλατφόρμα

Όταν αναφέρουμε τον όρο κινητή πλατφόρμα αναφερόμαστε σε έναν υπολογιστή μεγέθους τσέπης που αποτελείται κυρίως από μια οθόνη αφής ή μη και ένα πληκτρολόγιο. Κάποιοι άλλοι χαρακτηρισμοί είναι υπολογιστής τσέπης, συσκευή χειρός, κινητή συσκευή ή κινητό.

Τα έξυπνα κινητά είναι ιδιαίτερος διαδεδομένα στις μέρες μας και μας διευκολύνει σε καθημερινές μας λειτουργίες και ανάγκες παρέχοντας μας κάποιες βασικές δυνατότητες που έχει και ένας υπολογιστής.

Ουσιαστικά αναφερόμαστε σε κινητά τηλέφωνα και κινητές συσκευές τις οποίες ο χρήστης μπορεί να έχει μαζί του όπου και αν βρίσκεται και να έχει πρόσβαση σε διάφορα δεδομένα τα οποία είναι απαραίτητα για την καθημερινότητά του.

## 1.4 Είδη κινητής πλατφόρμας

### 1.4.1 Έξυπνα τηλέφωνα (smartphones)

Έξυπνο τηλέφωνο [1] είναι είναι ένα κινητό τηλέφωνο το οποίο βασίζεται σε ένα λειτουργικό σύστημα κινητής τηλεφωνίας με προηγμένη υπολογιστική ικανότητα και συνδεσιμότητα σε σχέση με ένα απλό συμβατικό κινητό τηλέφωνο.

Τα πρώτα έξυπνα τηλέφωνα συνδυάζαν βασικές λειτουργίες ενός κινητού τηλεφώνου και ενός προσωπικού ψηφιακού βοηθού (PDA). Τα επόμενα χρόνια στα προηγμένα μοντέλα προστέθηκαν διάφορες λειτουργίες άλλων συσκευών όπως των φορητών media-players, των ψηφιακών φωτογραφικών μηχανών, των βιντεοκαμερών τσέπης και των μονάδων πλοήγησης GPS, που είχαν ως αποτέλεσμα τη διαμόρφωση μιας συσκευής με πολλές χρήσεις.

Με την πάροδο των χρόνων προστέθηκαν και άλλα σημαντικά χαρακτηριστικά, όπως οι οθόνες αφής υψηλής ανάλυσης. Με τη χρήση Wifi και κινητών ευρυζωνικών

υπηρεσιών έγινε δυνατή η πρόσβαση σε δεδομένα υψηλής ταχύτητας. Η ανάπτυξη στην αγορά των κινητών ήταν ταχεία και η υιοθέτηση των έξυπνων τηλεφώνων γίνεται όσο περνάει ο καιρός όλο και πιο ευρεία.

Τα σύγχρονα έξυπνα τηλέφωνα χρησιμοποιούν λειτουργικά συστήματα τα οποία είναι κυρίως το Android της Google, το iOS της Apple, τα Windows phone της Microsoft, το Symbian της Nokia, το BlackBerry OS της RIM, καθώς και κάποιες ενσωματωμένες διανομές Linux, όπως το Maemo και το MeeGo.

Ο όρος έξυπνο τηλέφωνο (smartphone) εμφανίστηκε το 1997 όταν η Ericsson περιέγραψε το GS 88 “Penelope” ως smartphone. Μια από τις σημαντικότερες διαφορές ανάμεσα σε ένα smart phone και ένα απλό κινητό τηλέφωνο είναι ότι οι προηγμένες διεπαφές προγραμματισμού εφαρμογών (APIs) στα έξυπνα τηλέφωνα σχετικά με τη λειτουργία τρίτων εφαρμογών μπορούν να επιτρέψουν σε αυτές τις εφαρμογές να έχουν καλύτερη ενσωμάτωση στο λειτουργικό σύστημα και στο hardware του τηλεφώνου απ’ότι συμβαίνει συνήθως στα απλά κινητά τηλέφωνα.

Η ζήτηση των έξυπνων τηλεφώνων ξεπέρασε το υπόλοιπο της αγοράς κινητής τηλεφωνίας για πολλά χρόνια. Σύμφωνα με μια έρευνα που έγινε το στις ΗΠΑ το 2012, περίπου το ήμισυ των καταναλωτών κινητών τηλεφώνων έχουν στην κατοχή τους smartphones. Η ιδιοκτησία smartphones στην γκάμα ηλικίας 25-34 ανέρχεται στο 62%. Παγκοσμίως σε μερίδιο επί των κερδών, τα smartphones υπερβαίνουν κατά πολύ το μερίδιο των απλών κινητών τηλεφώνων.

Σύμφωνα με μια έρευνα του Canaccord Genuity το Νοέμβριο του 2011, η Apple κατέχει το 52% των συνολικών κερδών της βιομηχανίας της κινητής τηλεφωνίας, ενώ κατέχει μόνο το 4,2% της παγκόσμιας αγοράς συσκευών. Η Samsung κατέχει τη δεύτερη θέση με ποσοστό 29% και κατασκευάζει τόσο smartphones όσο και απλά κινητά τηλέφωνα, όμως δεν αναφέρει την κατανομή διαχωρισμού των κερδών της μεταξύ των δυο ειδών συσκευών.



*Εικόνα 1.1: HTC One Smartphone*

## 1.4.2 Ταμπλέτες (Tablets)

Ταμπλέτα (tablet) [2] είναι ένας φορητός υπολογιστής ή προσωπικός ψηφιακός βοηθός, το μέγεθός του είναι μεγαλύτερο από ένα κινητό τηλέφωνο, είναι ενσωματωμένο σε μια επίπεδη οθόνη αφής και κυρίως λειτουργεί αγγίζοντας την οθόνη αντί να χρησιμοποιεί ένα φυσικό πληκτρολόγιο. Χρησιμοποιεί την οθόνη σαν εικονικό πληκτρολόγιο, μια παθητική γραφίδα ή μια ψηφιακή πένα.

Οι ταμπλέτες διαθέτουν μια μεγάλη ποικιλία μεγεθών οθόνης, που ξεκινάει από τις 7 ίντσες και φτάνει τις 22. Τα συνηθέστερα μεγέθη βέβαια κυμαίνονται από 7 μέχρι 10.

Οι κύριες κατηγορίες αρχιτεκτονικής είναι το δισκίο x86 και η αρχιτεκτονική ARM. Η x86 είναι δημοφιλής στο Tablet PC, λόγω της χρήσης σε φορητούς υπολογιστές που μπορούν να μοιράζονται κοινό λογισμικό και υλικό στο οποίο μπορεί να τρέξει μια έκδοση των Windows. Η ARM έχει περισσότερη δύναμη και είναι πιο αποδοτική από πλευράς κόστους για φορητούς υπολογιστές και κερδίζει σε δημοτικότητα άλλους κατασκευαστές.

Κάποια βασικά χαρακτηριστικά των tablets είναι το επιταχυνσιόμετρο, οι αισθητήρες φωτισμού και εγγύτητας, τα μέσα αποθήκευσης, οι ασύρματες συνδέσεις και την τεχνολογία 3D.



*Εικόνα 1.2: Microsoft Surface 2 RT Tablet*

### 1.4.3 Προσωπικός ψηφιακός βοηθός (PDA)

Ο προσωπικός ψηφιακός οδηγός [3] είναι μια μικρή και εύχρηστη συσκευή. Οι συσκευές αυτές χρησιμοποιούνται με ένα ειδικό στυλό στη θέση ενός πληκτρολογίου και μπορούν να χρησιμοποιηθούν για την αποθήκευση και ανάκτηση πληροφοριών. Όπως οι περισσότερες συσκευές ηλεκτρονικών υπολογιστών, έτσι και πολλές από τις συσκευές αυτές έχουν τη δυνατότητα να συνδεθούν στο διαδίκτυο. Τα σύγχρονα μοντέλα έχουν επιπρόσθετα χαρακτηριστικά όπως μικρόφωνο, ακουστικό, αλλά και ηχείο και έτσι μπορούν να χρησιμοποιηθούν κατά περιπτώσεις ως κινητά τηλέφωνα και φορητά ηχοσυστήματα.

Ο όρος προσωπικός ψηφιακός βοηθός χρησιμοποιήθηκε πρώτη φορά το 1922 από τον John Sculley για τη συσκευή Apple Newton. Η συσκευή αυτή ήταν βασισμένη σε επεξεργαστή ARM και υποστήριζε αναγνώριση γραφής με μια ειδική γραφίδα. Το πρώτο

κινητό τηλέφωνο με όλες τις λειτουργίες ενός PDA που κατασκευάστηκε το 1966 από την εταιρεία Nokia, το 9000 Communicator, έγινε το πρώτο σε πωλήσεις παγκοσμίως και ήταν εκείνο που επί της ουσίας καθιέρωσε τον όρο PDA.

Τα βασικά λειτουργικά συστήματα για το PDA είναι τα Palm OS, Microsoft Windows Mobile (Pocket PC) με Windows CE kernel και WebOS. Κάποια άλλα που χρησιμοποιούνται πιο σπάνια είναι τα EPOC, Linux (VR3, iPAQ, Opie etc.), Newton και QNX.



*Εικόνα 1.3: Dell Axim X50V PDA*

## **1.5 Λειτουργικά συστήματα για τις κινητές πλατφόρμες**

Τα λειτουργικά συστήματα που χρησιμοποιούνται για τις κινητές πλατφόρμες είναι γνωστά ως Mobile Operating Systems [4]. Τα χαρακτηριστικά τους είναι ίδια με εκείνα των ηλεκτρονικών υπολογιστών, όπως δηλαδή τα Windows, Linux και Mac OS X.

Τα mobile operating systems είναι κατασκευασμένα και προσαρμοσμένα έτσι ώστε να είναι άμεσα συσχετισμένα με ασύρματες επικοινωνίες, τοπικά δεδομένα και να χρησιμοποιούν όσο το δυνατό γίνεται λιγότερους πόρους από έναν ηλεκτρονικό υπολογιστή.

## **1.6 Τα πρώτα λειτουργικά συστήματα για τις κινητές πλατφόρμες**

### **1.6.1 Palm OS**

Το Palm OS [5] είναι από τα πρώτα λειτουργικά συστήματα για smartphones και για τους υπολογιστές τσέπης Palm. Αρχικά οι συσκευές κατασκευάζονταν μόνο από την εταιρεία Palm και στη συνέχεια και άλλες εταιρείες χρησιμοποίησαν το λειτουργικό αυτό όπως οι Sony και Handspring. Οι εκδόσεις ξεκίνησαν από Palm OS 1.0 και έφτασαν έως Palm OS 4.0.

Το λειτουργικό αυτό ξεκίνησε το 1996 και χρησιμοποιήθηκε σε κινητές συσκευές, smartphones, GPS, barcode readers κλπ. Το 2009 η Palm επικεντρώθηκε στο Palm webOS και στις κινητές συσκευές Windows.



*Εικόνα 1.4: Palm OS*

## 1.6.2 Windows Mobile

Η εταιρεία Microsoft κατασκεύασε το λειτουργικό σύστημα για κινητά τηλέφωνα και η έκδοση αυτή είναι παρόμοια με αυτή για ηλεκτρονικούς υπολογιστές. Παρουσιάστηκε πρώτη φορά το 2000 ως PocketPC 2000 και μετονομάστηκε το 2003 σε Windows Mobile [6]. Έως το 2007 ήταν το πιο δημοφιλές λειτουργικό στην Αμερική και στην πορεία είχε καθοδική πορεία. Από το 2010 ο ανταγωνισμός με τα λειτουργικά αυτά των Android και iOS ήταν μεγάλος. Έπειτα η Microsoft σταμάτησε την εξέλιξη των Windows Mobile και επικεντρώθηκε σε αυτή των Windows Phone.

Υπάρχουν τρεις βασικές εκδόσεις των Windows Mobile. Η πρώτη είναι η Windows Mobile Professional και τρέχει σε smartphones με οθόνη αφής. Η δεύτερη είναι η Windows Mobile Standard και τρέχει σε κινητά τηλέφωνα χωρίς οθόνες αφής. Η τρίτη είναι η Windows Mobile Classic και τρέχει στον προσωπικό ψηφιακό βοηθό ή σε Pocket PCs.



Η τελευταία έκδοση εκδόθηκε το 2010 και ήταν η 6.5.5 όπου έπειτα από αυτήν η Microsoft ανακοίνωσε το νέο λειτουργικό σύστημα Windows Phone και σταμάτησε αυτό του Windows Mobile.



Εικόνα 1. 5: Windows Mobile 6.5

## 1.7 Τωρινά λειτουργικά συστήματα για κινητές πλατφόρμες

Έπειτα από τα λειτουργικά συστήματα που αναφέρθηκαν προηγουμένως αναπτύχθηκαν νέα, βελτιωμένα, με καινούρια χαρακτηριστικά λειτουργικά συστήματα που χρησιμοποιούνται μέχρι και σήμερα.

Τα λειτουργικά συστήματα για κινητές συσκευές συνδυάζουν χαρακτηριστικά των ηλεκτρονικών υπολογιστών με άλλα χαρακτηριστικά που είναι κατάλληλα για κινητή χρήση. Τα περισσότερα από αυτά είναι οθόνη αφής, Bluetooth, Wi-Fi, GPS κινητή πλοήγηση, κάμερα, καταγραφέας φωνής, αναπαραγωγή μουσικής κλπ.



Οι κινητές συσκευές με δυνατότητες επικοινωνίας περιλαμβάνουν δυο λειτουργικά συστήματα. Η κύρια πλατφόρμα συμπληρώνεται από μια δεύτερη χαμηλότερου επιπέδου που λειτουργεί το ραδιόφωνο και άλλα υλικά. Έρευνες έχουν δείξει ότι αυτά τα συστήματα χαμηλού επιπέδου μπορεί να περιέχουν κάποιες ευπάθειες σε θέματα ασφάλειας που επιτρέπουν σε κακόβουλους σταθμούς βάσης να παίρνουν τον έλεγχο σε υψηλά επίπεδα της κινητής συσκευής.

## 1.7.1 Android

Το Android [7] είναι λειτουργικό σύστημα το οποίο τρέχει τον πυρήνα του λειτουργικού Linux και αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Οι κατασκευαστές λογισμικού γράφουν κώδικα στη γλώσσα προγραμματισμού Java και χρησιμοποιούν διάφορες βιβλιοθήκες λογισμικού αναπτυγμένων από την Google με σκοπό να ελέγξουν τη συσκευή.

Είναι κυρίως ανεπτυγμένο για συσκευές με οθόνες αφής, δηλαδή για κινητά τηλέφωνα, ταμπλέτες, τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto) και ρολόγια χειρός (Android Wear). Εκτός όμως από αυτά έχει χρησιμοποιηθεί και σε ψηφιακές φωτογραφικές μηχανές, σε κονσόλες παιχνιδιών, σε ηλεκτρονικούς υπολογιστές και σε άλλες ηλεκτρονικές συσκευές.

Οι συσκευές με Android έχουν τις περισσότερες πωλήσεις και είναι το πιο ευρέως διαδεδομένο λογισμικό στον κόσμο. Η πρώτη του παρουσίαση έλαβε χώρα το 2007. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache Licence, μιας ελεύθερης άδειας λογισμικού. Το λογότυπο του Android σχεδιάστηκε από την Ιρίνα Μπλοκ.

Γίνεται χρήση βάσης δεδομένων SQLite για τις ανάγκες αποθήκευσης. Υποστηρίζει διάφορες τεχνολογίες συνδεσιμότητας όπως 3G, 4G, Bluetooth και Wi-fi. Διαθέτει φυλλομετρητή βασισμένο στην τεχνολογία WebKit και είναι δυνατό να συνεργαστεί με οθόνες αφής, αισθητήρες επιτάχυνσης, GPS κλπ.

Επίσης περιλαμβάνει προσομοιωτή συσκευής, μνήμη, εργαλεία για διόρθωση σφαλμάτων και για ανάλυση της απόδοσης του εκτελέσιμου λογισμικού. Το Google Play είναι ένας κατάλογος εφαρμογών που μπορούν να φορτωθούν και να εγκατασταθούν στις συσκευές άμεσα μέσω ασύρματων καναλιών, χωρίς τη χρήση υπολογιστή. Στην αρχή ήταν διαθέσιμες μόνο δωρεάν εφαρμογές, όμως το 2009 έγιναν διαθέσιμες και εφαρμογές επί πληρωμή.



*Εικόνα 1.6: Android 5.0*

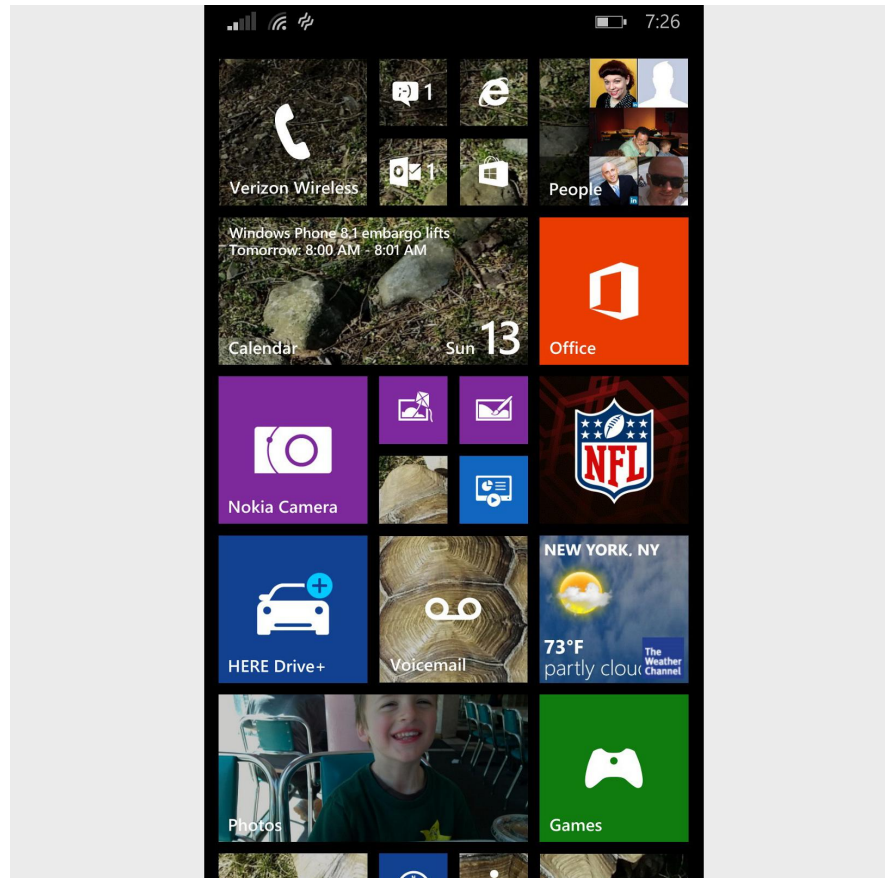
## 1.7.2 Windows Phone

Είναι η έκδοση του λειτουργικού συστήματος Windows της Microsoft για smartphones. Αναπτύχθηκε στις αρχές του 2000 και στις αρχικές του εκδόσεις το όνομά του ήταν Windows Mobile. Από το 2010 έως και τώρα οι πωλήσεις των Windows Phone [8] έχουν αυξηθεί κατά πολύ και έφτασαν στο νούμερο δύο των πωλήσεων στην κατηγορία των smartphones, ενώ υπολογίζεται από ειδικούς πως μέχρι το 2017 οι πωλήσεις θα αυξηθούν κι άλλο.

Το Windows Phone χαρακτηρίζεται κυρίως από τη Nokia, εφόσον η σειρά μοντέλων Lumia λειτουργεί μόνο με Windows Phone. Επιπλέον, η Lumia μεσολάβησε ώστε η Microsoft να αγοράσει τη Nokia. Η Microsoft κυκλοφόρησε την πρώτη δική της συσκευή Windows Phone με την ονομασία Microsoft Lumia 535.

Τα Windows Phone προσφέρουν στους χρήστες μια εύκολη χρήση και κάνουν τη ζωή τους πιο απλή καθώς έχουν προστεθεί πολλές λειτουργίες από αυτές των ηλεκτρονικών υπολογιστών. Οι χρήστες μπορούν να πλοηγηθούν με τον Internet Explorer και επίσης έχει προστεθεί το Microsoft Office για τις εφαρμογές γραφείου. Διάφορες εφαρμογές προσφέρονται στο Windows app store και δωρεάν και επί πληρωμή και με την προσθήκη του Xbox οι χρήστες μπορούν να παίζουν τα παιχνίδια που κατεβάζουν.

Η αναβάθμιση γίνεται χωρίς τη χρήση υπολογιστή, μόνο από το τηλέφωνο του χρήστη. Η τελευταία έκδοση είναι η Windows 8.1.



*Εικόνα 1.7: Windows Phone 8.1*

### 1.7.3 Blackberry OS

Το Blackberry OS [9] αναπτύχθηκε από την εταιρεία Blackberry Ltd για τη σειρά συσκευών Blackberry. Παρέχει πολλές λειτουργίες όπως τη δυνατότητα χρήσης πολλών εφαρμογών ταυτόχρονα.

Ο αρχικός στόχος της ανάπτυξης αυτού του λειτουργικού ήταν να υποστηρίξει εταιρικές εφαρμογές και κυρίως email και γι' αυτό χρησιμοποιείται κυρίως για εταιρικούς λόγους.

Οι εφαρμογές για το BlackBerry OS γίνονται προσβάσιμες από το BlackBerry App world, το οποίο ονομάστηκε BlackBerry World μετά από την ανακοίνωση της έκδοσης BlackBerry10.

Οι εφαρμογές αναπτύσσονται σε στο ολοκληρωμένο περιβάλλον ανάπτυξης QNX Momentics IDE και οι γλώσσες οι οποίες συντελούν στη δημιουργία τους είναι οι Java, C/C++, HTML5, JAVASCRIPT και CSS.



*Εικόνα 1.8: BlackBerry 10 OS*

## 1.7.4 iOS

Το iOS [10] είναι το λογισμικό για κινητά τηλέφωνα το οποίο αναπτύχθηκε από την Apple Inc. Παρουσιάστηκε το 2007 για το τηλέφωνο iPhone αρχικά και έπειτα αναπτύχθηκε και για άλλες συσκευές, όπως το iPod touch, το iPad και το Apple TV. Η Apple δε δίνει την άδεια για εγκατάσταση του λογισμικού της σε άλλες συσκευές που δεν ανήκουν σε αυτή, σε αντίθεση με το Android της Google και το Windows Phone της Microsoft.

Οι εφαρμογές διατίθενται από το App Store, το οποίο τον Ιανουάριο του 2015 είχε πάνω από 1,4 εκατομμύρια εφαρμογές. Οι εφαρμογές αυτές έχουν μεταφορτωθεί πάνω από 100 δισεκατομμύρια φορές.

Το iOS βασίζεται στην έννοια του άμεσου χειρισμού με τη χρήση πολλών χειρισμών ταυτόχρονα. Χρησιμοποιούνται διάφορα στοιχεία για το χειρισμό του λογισμικού, όπως ολισθητές, διακόπτες και κουμπιά. Η αλληλεπίδραση του χρήστη με το λειτουργικό σύστημα περιλαμβάνει επίσης χειρονομίες, όπως χτύπημα στην οθόνη. Ορισμένες εφαρμογές χρησιμοποιούν εσωτερικά επιταχυνσιόμετρα. Κάποιες άλλες βασικές λειτουργίες είναι η μεγέθυνση που μπορεί να επιτευχθεί με το άνοιγμα των δυο δαχτύλων ή η εναλλαγή φωτογραφιών σέρνοντας το δάχτυλο δεξιά ή αριστερά.

Το iOS μοιράζεται κάποια Frameworks με το OS X, όπως το Core Foundation και το Foundation Kit. Όμως η εργαλειοθήκη του είναι η Cocoa Touch και όχι η Cocoa όπως στο OS X. Αυτό συμβαίνει για να παρέχει το UIKit framework και όχι το AppKit framework. Ως εκ τούτου δεν είναι συμβατό με εφαρμογές OS X.

Νέες εκδόσεις του λογισμικού ανακοινώνονται και δημοσιεύονται κάθε χρόνο. Η τρέχουσα έκδοση είναι η iOS 9.0 και δημοσιεύτηκε 16 Σεπτεμβρίου το 2015. Απαιτεί περίπου 1,3 GB της μνήμης flash της συσκευής. Τρέχει για το iPhone 4S και τις νεότερες εκδόσεις των κινητών τηλεφώνων της, από το iPad 2 και έπειτα, το iPad Pro, όλα τα μοντέλα του iPad Mini και τέλος για όλη την πέμπτη γενιά iPod Touch και μετά. Υπάρχουν τέσσερα επίπεδα άντλησης, το Core OS, το Core Services, το Media και το Cocoa Touch. Το Φεβρουάριο του 2015 η StatCounter ανέφερε ότι το iOS χρησιμοποιείται στο 23,18% των συσκευών και στο 66,25% των ταμπλετ παγκοσμίως.

Η Apple παρέχει τις κύριες αναβαθμίσεις του λογισμικού στο iTunes. Πριν την έκδοση iOS 4 το 2010, οι χρήστες iPod Touch έπρεπε να πληρώσουν για τις αναβαθμίσεις του λογισμικού.

Οι εφαρμογές αναπτύσσονται με το εργαλείο Xcode και την έκδοση 5 ή μεταγενέστερη και υλοποιούνται για το λογισμικό iOS 5 ή μεταγενέστερο. Τρέχουν στο iOS και την αρχιτεκτονική 64-bit ARM ή την προηγούμενη 32-bit. Για την πλοήγηση χρησιμοποιείται ο browser Safari ο οποίος υποστηρίζει και τις web εφαρμογές.

Για να αναπτυχθεί μια εφαρμογή χρησιμοποιείται το λογισμικό ανάπτυξης εφαρμογών iOS SDK το οποίο δημιούργησε η Apple και δόθηκε στους προγραμματιστές το 2008. Οι προγραμματιστές μπορούν να δημιουργήσουν εφαρμογές με το εργαλείο Xcode και να τις δοκιμάσουν στον εξομοιωτή που παρέχει το Xcode, τον iOS Simulator. Για να μπορέσει η εφαρμογή να ανέβει στο App Store και να διατεθεί στο κοινό, θα πρέπει ο ιδρυτής της να καταβάλει το ποσό των 99 ευρώ το χρόνο. Ο ιδρυτής της εφαρμογής μπορεί να ορίσει την τιμή για την πώληση της εφαρμογής του, η οποία θα πρέπει να έχει το ελάχιστο ποσό που ορίζει η Apple. Από τα κέρδη της εφαρμογής θα λαμβάνει το 70% και το υπόλοιπο 30% ανήκει στην Apple. Υπάρχει και η εναλλακτική της δωρεάν διάθεσης όπου δεν υπάρχουν έξοδα διανομής, μόνο αυτά των εξόδων εγγραφής.

Οι προγραμματιστές υλοποιούν τις εφαρμογές τους στις γλώσσες προγραμματισμού Objective-C και Swift. Οι απαιτήσεις του συστήματος είναι ένας υπολογιστής Mac με λειτουργικό σύστημα Mac OS X Leopard ή νεότερο. Οι προηγούμενες εκδόσεις δεν υποστηρίζονται, ούτε και άλλα λειτουργικά συστήματα όπως τα Windows.

Το iOS είναι το δεύτερο δημοφιλέστερο λογισμικό για κινητές συσκευές μετά από το Android. Στις ταμπλέτες είναι το πιο δημοφιλές. Μέχρι τα μέσα του 2012 ενεργοποιήθηκαν 410 εκατομμύρια συσκευές, ενώ το 2014 ο Tim Cook είπε ότι πωλήθηκαν 800 εκατομμύρια συσκευές έως τον Ιούνιο του ίδιου έτους. Τον Ιανουάριο του 2015 η Apple ανακοίνωσε ότι πούλησε 1 δισεκατομμύριο συσκευές από το 2007.



*Εικόνα 1.9: iOS 8*

### 1.7.5 Άλλα λειτουργικά συστήματα

Κάποια άλλα λειτουργικά συστήματα για κινητές πλατφόρμες που χρησιμοποιούνται λιγότερο σήμερα [4] είναι τα Firefox OS, Sailfish OS, Tizen και Ubuntu Touch OS.

## **1.8 Λειτουργικά συστήματα για κινητές πλατφόρμες που δε χρησιμοποιούνται πια**

Υπάρχουν κάποια λειτουργικά συστήματα που χρησιμοποιούνταν στο παρελθόν [4] και όχι πια όπως τα Bada, Symbian, Palm OS, webOS, Maemo, MeeGo, Limo.

## **1.9 Εφαρμογές για κινητές πλατφόρμες**

Οι εφαρμογές κινητών πλατφορμών [11] σχεδιάστηκαν για να τρέχουν σε κινητές συσκευές, όπως τα κινητά τηλέφωνα και οι ταμπλέτες. Οι περισσότερες συσκευές πωλούνται με προεγκατεστημένες κάποιες εφαρμογές, όπως είναι το email, οι χάρτες, το ημερολόγιο και η εφαρμογή για αγορές άλλων εφαρμογών.

Η λέξη εφαρμογή έγινε πολύ δημοφιλής και το 2010 η American Dialect Society την κατέταξε σε “Word of the Year”.

Οι εφαρμογές που διατίθενται ανήκουν σε διάφορες κατηγορίες για παραγωγή και ανάκτηση δεδομένων, όπως email, ημερολόγιο, επαφές και πληροφορίες καιρού. Στην πορεία της ανάπτυξης των εφαρμογών μεγάλωσαν οι απαιτήσεις των χρηστών και όλο και περισσότερες εφαρμογές άρχισαν να δημιουργούνται καλύπτοντας ένα ευρύ φάσμα. Σε αυτό φυσικά έπαιξε ρόλο και η μεγάλη αύξηση του αριθμού των χρηστών που χρησιμοποιούν έξυπνες συσκευές. Τα κέρδη που αποφέρουν οι εφαρμογές κινητών συσκευών είναι πολλά και πάνω από 529.000 θέσεις εργασίας δημιουργήθηκαν λόγω της μεγάλης ανάπτυξης της αγοράς εφαρμογών.

Οι εφαρμογές που υπάρχουν σήμερα βοηθούν τους χρήστες κινητών συσκευών να κάνουν τη ζωή τους πιο εύκολη και να έχουν οποιαδήποτε στιγμή πρόσβαση σε δεδομένα που χρειάζονται. Σε πολλές από τις εφαρμογές φυσικά είναι απαραίτητη η χρήση δεδομένων για πρόσβαση στο διαδίκτυο, όπως διάφορες εφαρμογές παιχνιδιών και μηνυμάτων.





# Κεφάλαιο 2

## Υλικό και λογισμικό

Στο κεφάλαιο αυτό θα περιγραφεί το υλικό και λογισμικό που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής.

### 2.1 Υλικό

#### 2.1.1 Ηλεκτρονικός υπολογιστής

Για να μπορέσει να δημιουργηθεί μια εφαρμογή σε iOS απαραίτητη προϋπόθεση αποτελεί ένας ηλεκτρονικός υπολογιστής με το λειτουργικό σύστημα που παρέχει η Apple.

Η Apple παράγει τους εξής ηλεκτρονικούς υπολογιστές [12]:

- iMac, ο οποίος αποτελείται από μια οθόνη και όλο του το υλικό βρίσκεται εκεί
- Macbook, το οποίο είναι είδος notebook, ιδιαίτερα λεπτό
- Macbook Pro, ο οποίος είναι φορητός υπολογιστής
- Macbook Air, ο οποίος είναι επίσης φορητός υπολογιστής, πιο λεπτός και ελαφρύς από τον Macbook Pro
- Mac Mini, το οποίο είναι ένα ολοκληρωμένο υπολογιστικό σύστημα και οι διαστάσεις του μοιάζουν με αυτές του σκληρού εξωτερικού δίσκου
- Mac Pro, ο οποίος είναι σταθερός υπολογιστής

## 2.1.2 Συσκευές iPhone, iPod Touch, iPad

Η Apple παρέχει τις συσκευές iPhone, iPod Touch και iPad και παρέχουν στο χρήστη μια μεγάλη γκάμα από λειτουργίες και εφαρμογές. [12]. Στην αγορά ξεχωρίζουν οι συσκευές αυτές κυρίως για το απλό λειτουργικό τους σύστημα που είναι ιδιαίτερος φιλικό προς το χρήστη και του παρέχονται δυνατότητες για φωτογραφία, μουσική, διαδίκτυο κλπ.

Το λειτουργικό είναι τόσο απλά σχεδιασμένο ώστε με τη χρήση και μόνο δυο δαχτύλων να γίνεται εισαγωγή και επεξεργασία δεδομένων. Το πληκτρολόγιο που χρησιμοποιείται για την εισαγωγή των δεδομένων είναι πάνω στην οθόνη και είναι εικονικό. Φυσικά πάνω στη συσκευή υπάρχει ένα κεντρικό κουμπί το οποίο ανά πάσα στιγμή φέρνει το χρήστη στην αρχική του θέση στη συσκευή, απομακρύνοντάς τον από την εφαρμογή που πιθανώς έχει ανοιχτή εκείνη τη στιγμή.

Στις τρεις αυτές συσκευές υπάρχουν τρεις βασικοί αισθητήρες. Ο πρώτος χρησιμοποιείται για να αντιλαμβάνεται την αλλαγή στη γωνία κλίσης της συσκευής, εάν είναι κάθετη η οριζόντια, ο δεύτερος για να απενεργοποιεί την οθόνη όταν ο χρήστης έχει τη συσκευή στο αυτί του και μιλάει στο τηλέφωνο και ο τρίτος για να ρυθμίζει τη φωτεινότητα της οθόνης και να την αυξομειώνει ώστε να μην ξοδεύεται άσκοπα η μπαταρία ενώ δεν είναι απαραίτητο.

Το 2007 παρουσιάστηκε για πρώτη φορά το iPhone και ξεκινούσε από την τιμή των 499 δολαρίων με συμβόλαιο. Η επόμενη έκδοση ήρθε το 2008 και ήταν το iPhone 3G. Το 2009 παρουσιάστηκε το iPhone 3GS με καλύτερα χαρακτηριστικά από το προηγούμενο μοντέλο και το κύριο είναι η προσθήκη δυνατότητα χρήσης δικτύου κινητής τηλεφωνίας 3G. Ακολούθησε το iPhone 4 το 2010 με προσθήκη μπροστινής κάμερας, καλύτερη οθόνη και καλύτερο επεξεργαστή, το iPhone 4s το 2011, το iPhone 5 το 2012, το iPhone 5S το 2013 με τη σημαντική προσθήκη του touch ID, τα iPhone 6 και iPhone 6 Plus το 2014 και τα iPhone 6S και 6S Plus το 2015.

Το 2001 έκανε την εμφάνισή του το iPod Touch, το οποίο αυτή τη στιγμή μοιάζει με το iPhone αλλά χωρίς τη δυνατότητα ενός κινητού τηλεφώνου. Χρησιμοποιεί και αυτό το λειτουργικό σύστημα iOS και δεν υποστηρίζει εφαρμογές για τις οποίες είναι απαραίτητη η σύνδεση σε δίκτυο κινητής τηλεφωνίας. Η μπαταρία του κρατάει περισσότερο από το iPhone καθώς καταναλώνει λιγότερη ενέργεια και κοστίζει λιγότερο.

Το 2010 παρουσιάστηκε το iPad. Χρησιμοποιεί επίσης το λειτουργικό σύστημα iOS και είναι μεγαλύτερο από το iPhone. Μια από τις βασικότερες λειτουργίες του είναι η ανάγνωση ηλεκτρονικών βιβλίων. Φυσικά όμως υποστηρίζει διάφορες εφαρμογές και παιχνίδια τα οποία έχουν επανασχεδιαστεί για να λειτουργούν και σε αυτό.

## 2.2 Το λογισμικό

### 2.2.1 Mac OS X

Το Mac OS X είναι μια σειρά γραφικών λειτουργικών συστημάτων που αναπτύσσεται και πωλείται από την Apple Inc. και περιέχεται στους υπολογιστές Mac ως το λειτουργικό τους σύστημα. Το Mac OS X [13] είναι η εξέλιξη του OS X το οποίο ήταν το αρχικό λειτουργικό σύστημα της Apple από το 1984 έως το 1999.

Το Mac OS X είναι ένα UNIX λειτουργικό σύστημα το οποίο άρχισε να αναπτύσσεται στην εταιρεία NeXT από τα τέλη του 1980 έως και το 1997 που ήρθε η εξαγορά από την Apple.

Η πρώτη έκδοση κυκλοφόρησε το 1999 ως Mac OS X Server 1.0, ενώ η πρώτη έκδοσή του Mac OS X v10.0 "Cheetah για φορητούς και επιτραπέζιους υπολογιστές ήρθε το 2001.

Το Mac OS X έχει τις εξής αναβαθμίσεις:

- Mac OS X v10.1 "Puma"
- Mac OS X v10.2 "Jaguar"
- Mac OS X v10.3 "Panther"
- Mac OS X v10.4 "Tiger"
- Mac OS X v10.5 "Leopard"
- Mac OS X v10.6 "Snow Leopard"
- Mac OS X v10.7 "Lion"
- Mac OS X v10.8 "Mountain Lion"
- Mac OS X v10.9 "Mavericks"
- Mac OS X v10.10 "Yosemite"
- Mac OS X v10.11 "El Capitan"

Το Mac OS X μπορεί να αναβαθμιστεί από το Mac App Store.

Όταν η Apple προχώρησε στην αλλαγή του OS X σε MAC OS X, η πιο ορατή διαφορά ήταν το γραφικό περιβάλλον Aqua. Η χρήση κουμπιών σαν χρωματιστές σταγόνες, οι διαφάνειες και τα φωτορεαλιστικά εικονίδια έφεραν υφή και χρώμα σε σχέση με τα προηγούμενα λειτουργικά. Πολλοί χρήστες εξέφρασαν την αρνητική άποψη ότι ήταν πολύ "χαριτωμένο" χωρίς επαγγελματικό ερέθισμα. Άλλοι πίστεψαν ότι το Aqua ήταν ένα γενναίο και πρωτοποριακό βήμα σε μια εποχή που τα γραφικά περιβάλλοντα ήταν απλά βαρετά. Παρόλο το διχασμό, η εμφάνισή του ήταν άμεσα αναγνωρίσιμη, ακόμα και πριν

την πρώτη έκδοσή του Mac OS X, άλλοι προγραμματιστές άρχισαν να προσπαθούν να αντιγράψουν την εμφάνιση του Aqua.

Κατά τα τέλη της δεκαετίας του '90, για τη διευκόλυνση της μετατροπής των υφισταμένων εφαρμογών από το Mac OS 9 στο Mac OS X, ενσωματώθηκε στο Mac OS X το Carbon (άνθρακας) API (Application Programming Interface - περιβάλλον προγραμματισμού εφαρμογών). Τα πιο ισχυρά API του Mac OS X που προήλθαν από το Openstep δεν ήταν συμβατά με τις εκδόσεις που προηγήθηκαν του Mac OS X. Αυτά τα API αναφέρονται ως Cocoa. Αυτή η κληρονομιά είναι πολύ εμφανής στους προγραμματιστές με Cocoa, αφού τα περισσότερα Cocoa class ονόματα ξεκινούν με το συνθετικό "NS" από τη λέξη Nextstep.

Το Mac OS X διατηρεί συμβατότητα με εφαρμογές γραμμένες για παλιότερες εκδόσεις του Mac OS παρέχοντας ένα περιβάλλον προσομοίωσης που λέγεται Classic, που επιτρέπει στους χρήστες να τρέχουν το MAC OS 9 ως κομμάτι του Mac OS X.



Εικόνα 2.10: Mac OS 9

## 2.2.2 iOS SDK

Το iOS SDK (Software Development Kit) [14] δημιουργήθηκε από την Apple Inc. και δημοσιεύτηκε το 2008 με σκοπό να αναπτυχθούν εφαρμογές για το iOS.

Τα περιεχόμενά του περιλαμβάνουν τις τέσσερις βασικές κατηγορίες:

- Cocoa Touch
- Media
- Core Services
- Core OS

### Cocoa Touch

Το Cocoa Touch είναι UI framework που ουσιαστικά χρησιμοποιείται για να τρέξουν οι εφαρμογές στο iOS. Βασίζεται στο Mac OS X Cocoa API και είναι κυρίως γραμμένο σε Objective-C. Επιτρέπει στο χρήστη να χρησιμοποιήσει υλικά και χαρακτηριστικά που δεν υπάρχουν στο Mac OS X και είναι μοναδικά στο iOS. Ακολουθεί την αρχιτεκτονική λογισμικού Model-View-Controller.

Κάποια από τα κύρια χαρακτηριστικά του και τις τεχνολογίες του είναι:

- App επέκταση
- Handoff
- Document Picker
- AirDrop
- TextKit
- UIKit δυναμική
- Πολυδιεργασίες
- Αυτόματη διάταξη
- Storyboards
- UI κατάσταση διατήρησης
- Apple υπηρεσία έγκαιρης ειδοποίησης
- Τοπικές ειδοποιήσεις
- Αναγνώριση κίνησης

- Πρότυπο σύστημα View Controllers

Τα κύρια frameworks για ανάπτυξη εφαρμογών σε συσκευές iOS είναι:

- Foundation Kit Framework
- UIKit Framework
- GameKit Framework
- iAd Framework
- MapKit Framework
- Address Book UI Framework
- EventKit UI Framework
- Message UI Framework
- Notification Center Framework
- PushKit Framework
- Twitter Framework

## Media

Σ' αυτή την κατηγορία περιλαμβάνονται κατηγορίες γραφικών και πολυμέσων. Χρησιμοποιούνται frameworks που επιτρέπουν animations και γραφικά.

Χρησιμοποιούνται τεχνολογίες όπως είναι:

- OpenAL (Open Audio Library)
- audio mixing
- audio recording
- video playback
- Quartz
- Core animation
- OpenGL ES

Η τεχνολογία OpenGL παρέχει τη δυνατότητα για animation με γραφικά επιπέδου 3D.

## Core Services

Το Core Services ουσιαστικά περιέχει frameworks και βιβλιοθήκες για την ανάπτυξη εφαρμογών.

Περιλαμβάνει:

- Σύνδεση δικτύων
- Ενσωματωμένη βιβλιοθήκη SQLite
- Core Location
- Threads
- Core Motion

Το σημαντικότερο framework είναι το Core Foundation το οποίο αποτελείται από διεπαφές για διαχείριση δεδομένων και υπηρεσιών, όπως η διαχείριση προτιμήσεων του χρήστη. Τα frameworks αυτά είναι γραμμένα στη γλώσσα C.

Η ενσωματωμένη βιβλιοθήκη SQLite δίνει τη δυνατότητα για αποθήκευση δεδομένων.

Το Core Location ανακοινώθηκε το 2008 και είναι ένα framework το οποίο χρησιμοποιείται κυρίως από εφαρμογές iOS για τον εντοπισμό της συσκευής . Τρέχει σε iPhone 3.0 και μετέπειτα.

## Core OS

Το Core OS περιλαμβάνει drivers, αρχεία συστήματος, τον πυρήνα και είναι υπεύθυνο για τη διαχείριση των threads, της μνήμης και τη δικτύωση.



## 2.2.3 Objective C

Η Objective-C [15] είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που βασίζεται στη C. Είναι η βασική γλώσσα προγραμματισμού που χρησιμοποιείται για την ανάπτυξη εφαρμογών στα iOS και OS X λειτουργικά συστήματα.

Αναπτύχθηκε στις αρχές του 1980 και ήταν η κύρια γλώσσα προγραμματισμού που χρησιμοποιήθηκε από τη NeXT για το λειτουργικό σύστημα NextSTEP.

Τα αρχεία implementation έχουν κατάληξη .m και τα αρχεία header/interface έχουν κατάληξη .h.

Εφόσον είναι αντικειμενοστραφής γλώσσα προγραμματισμού χρησιμοποιεί αντικείμενα, κλάσεις, κληρονομικότητα και άλλα χαρακτηριστικά των αντικειμενοστραφών γλωσσών. Ακόμη υπάρχει η δυνατότητα για χρήση βιβλιοθηκών σε C καθώς είναι ένα υπερσύνολο της C.



# Objective-C

*Εικόνα 2.11: Γλώσσα προγραμματισμού Objective-C*

## 2.3.4 Swift

Η Swift [16] είναι η διάδοχος γλώσσα προγραμματισμού της Objective-C που δημιούργησε η Apple με σκοπό τον προγραμματισμό εφαρμογών σε iOS, OS X και Watch OS και παρουσιάστηκε το 2014. Η Apple προσπάθησε να πάρει πιο απλά στοιχεία προγραμματισμού όπως αυτά της Python ώστε να δημιουργήσει μια γλώσσα πιο εύκολη ως προς τη χρήση της. Σχεδιάστηκε για να λειτουργεί με τα frameworks Cocoa και Cocoa Touch καθώς επίσης και με κώδικα γραμμένο σε Objective-C. Η Apple υποστηρίζει πως η Swift είναι πιο ασφαλής γλώσσα προγραμματισμού, καθώς επίσης και πιο περιεκτική. Είναι φτιαγμένη με το framework LLVM compiler και επιτρέπει τη χρήση Objective-C, C, C++ και Swift στο ίδιο πρόγραμμα.

Η Swift δεν εκθέτει pointers και accessors, σε αντίθεση με την Objective-C η οποία χρησιμοποιεί pointers διάχυτα. Χρησιμοποιεί τον ίδιο χρόνο εκτέλεσης, αλλά απαιτεί iOS 7, OS X 10.9 και μεταγενέστερα. Χρησιμοποιεί αυτόματη αναφορά καταμέτρησης (Automatic Reference Counting) για να κάνει διαχείριση μνήμης.

Κάποιες διαφορές με την Objective-C είναι ότι στις δηλώσεις δε χρειάζεται να χρησιμοποιείται το σύμβολο ; , δε χρειάζονται header αρχεία, δε χρειάζεται δήλωση break στην περίπτωση switch κλπ.

Η νέα αυτή γλώσσα προγραμματισμού υιοθετείται από προγραμματιστές όλο και περισσότερο καθώς είναι πιο εύκολη στη χρήση της και με πολλές δυνατότητες.



*Εικόνα 2.12: Γλώσσα προγραμματισμού Swift*

### **2.3.5 Xcode**

Το Xcode [17] είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού, το οποίο αποτελείται από μια σουίτα εργαλείων για την ανάπτυξη εφαρμογών για τα λειτουργικά συστήματα iOS και MAC OS. Δημιουργήθηκε από την Apple το 2003. Η τελευταία του σταθερή έκδοση είναι η 7.0.1 και είναι διαθέσιμη δωρεάν στο Mac App Store για τους χρήστες με OS X Yosemite και OS X El Capitan.

Το εργαλείο αυτό είναι περίπλοκο και συνδυαστικό και χρησιμοποιείται σχεδόν εξ ολοκλήρου για να κατασκευαστεί μια εφαρμογή.

Κάποια από τα βασικά χαρακτηριστικά του είναι τα εξής:

- Source Editor
- Assistant Editor
- Interface Builder
- iOS Simulator
- Compilers
- Asset Catalog



*Εικόνα 2.13: Εργαλείο προγραμματισμού Xcode*

### 2.3.6 SQLite

Το SQLite [18] είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που περιέχεται σε μια προγραμματιστική βιβλιοθήκη C. Σε αντίθεση με άλλα συστήματα διαχείρισης βάσης δεδομένων, το SQLite δεν είναι μια ξεχωριστή διεργασία που προσπελάζεται από μια εφαρμογή πελάτη, αλλά ένα ενσωματωμένο μέρος της.

Είναι συμβατό με ACID και υλοποιεί το μεγαλύτερο μέρος του προτύπου SQL, χρησιμοποιώντας μια δυναμική και εβδομαδιαία τυπωμένη SQL σύνταξη που δεν εγγυάται την ακεραιότητα του τομέα. Είναι μια δημοφιλής επιλογή ως ενσωματωμένη βάση δεδομένων για τοπική αποθήκευση ή αποθήκευση πελάτη σε λογισμικό εφαρμογής, όπως ως φυλλομετρητής. Είναι η πιο πλατιά αναπτυσσόμενη μηχανή βάσης δεδομένων και χρησιμοποιείται σήμερα από πολλούς φυλλομετρητές ευρείας χρήσης, από λειτουργικά συστήματα και από ενσωματωμένα συστήματα. Το SQLite συνδέεται με πολλές γλώσσες προγραμματισμού.

Το SQLite αποθηκεύει τη συνολική βάση δεδομένων (ορισμούς, πίνακες, δείκτες και τα ίδια τα δεδομένα) ως ένα μοναδικό διαλειτουργικό αρχείο στη μηχανή ενός οικοδεσπότη. Υλοποιεί αυτό τον απλό σχεδιασμό με κλείδωμα όλου του αρχείου της βάσης δεδομένων κατά τη διάρκεια της εγγραφής. Το SQLite διαβάζει λειτουργίες που μπορεί να είναι πολυλειτουργικές, αν και οι εγγραφές μπορούν να γίνουν μόνο με τη σειρά.

Πολλές διεργασίες υπολογιστή ή threads μπορεί να προσπελάσουν την ίδια βάση δεδομένων ταυτόχρονα. Πολλές προσπελάσεις ανάγνωσης μπορούν να ικανοποιηθούν παράλληλα. Μια πρόσβαση εγγραφής μπορεί να ικανοποιηθεί μόνο αν καμιά άλλη πρόσβαση δεν εξυπηρετείται την ίδια στιγμή. Αλλιώς, η πρόσβαση εγγραφής επιτυγχάνει με ένα κωδικό σφάλματος (ή μπορεί να ξαναγίνει αυτόματη προσπάθεια μέχρι τη λήξη του χρόνου). Αυτή η ταυτόχρονη κατάσταση πρόσβασης μπορεί να αλλάξει όταν αντιμετωπίζονται προσωρινοί πίνακες. Αυτός ο περιορισμός χαλάρωσε στην έκδοση 3.7 όταν η write-ahead logging (WAL) ενεργοποιήθηκε, ενεργοποιώντας ταυτόχρονες αναγνώσεις και εγγραφές.



*Εικόνα 2.14: SQLite*

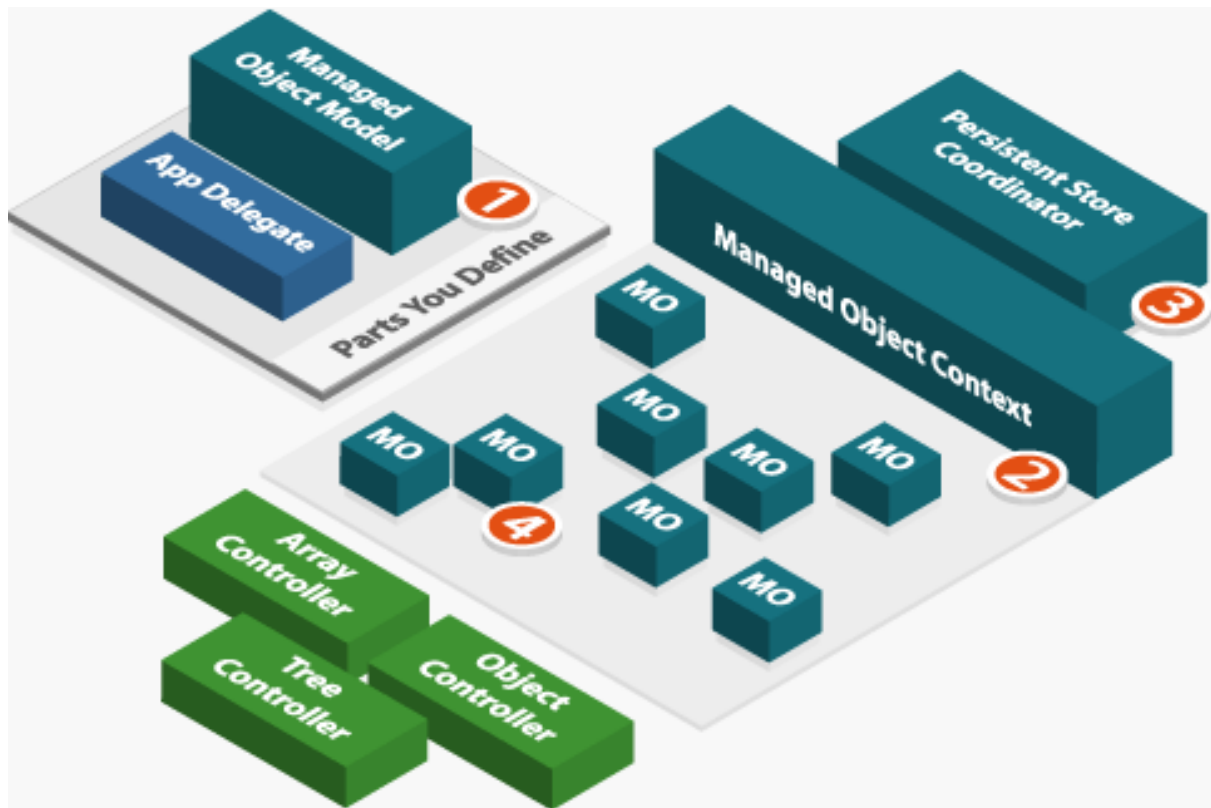
### **2.3.7 Core Data**

Core Data [19] είναι ένα framework που χρησιμοποιείται για τη διαχείριση model layer αντικειμένων σε μια εφαρμογή. Παρέχει γενικευμένες και αυτοματοποιημένες λύσεις σε κοινές εργασίες που σχετίζονται με τον κύκλο ζωής του αντικειμένου και τη διαχείριση γραφήματος αντικειμένου. Παρέχεται από την Apple για τα λειτουργικά συστήματα OS X και iOS.

Το Core Data τυπικά μειώνει κατά 50 έως 70 τα εκατό την ποσότητα του κώδικα που χρειάζεται να γραφεί για να υποστηριχτεί το model layer.

Τα δεδομένα μπορούν να επεξεργαστούν χρησιμοποιώντας τα υψηλότερα επίπεδα αντικειμένων που εκπροσωπούν κάποιες οντότητες και τις μεταξύ τους σχέσεις. Το Core

Data διασυνδέεται άμεσα με την SQLite. Μπορεί να σειριοποιήσει αντικείμενα σε XML, Binary ή SQLite για αποθήκευση.



Εικόνα 2.15: Core Data



# Κεφάλαιο 3

## Ανάλυση της εφαρμογής

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν υλικά και λογισμικό που αναφέρθηκαν προηγουμένως και θα αναλυθούν περισσότερο. Το είδος της εφαρμογής που αναπτύχθηκε ανήκει στην κατηγορία εφαρμογών Health and Fitness. Αναφέρεται κυρίως σε ανθρώπους οι οποίοι έχουν στην καθημερινότητά τους και ως τρόπο ζωής τη γυμναστική και με τη βοήθεια της εφαρμογής μπορούν να έχουν πρόσβαση εύκολα στα δεδομένα που τους αφορούν και συνοπτικά να βλέπουν ανά πάσα στιγμή λεπτομέρειες που τους ενδιαφέρουν.

Όπως είναι φυσικό, δε γίνεται να παρουσιαστεί και να αναλυθεί όλος ο κώδικας και η σχεδιάσή του. Θα αναφερθούν όμως κάποια κομμάτια σημαντικά που θα μας κάνουν να καταλάβουμε τον τρόπο σκέψης, να πάρουμε μια ιδέα για το πως λειτουργεί το συνδυαστικό αυτό εργαλείο Xcode, τις δυνατότητες που μας παρέχει και πως μας βοήθησε ώστε να υλοποιήσουμε την εφαρμογή αυτή. Ακόμη θα γίνει μια εισαγωγή για τη γλώσσα προγραμματισμού Swift που χρησιμοποιήθηκε και θα δούμε πως συντάσσεται.

### 3.1 Απαιτήσεις σχεδίασης

Η σχεδίαση της εφαρμογής είναι σημαντική καθώς θα πρέπει με πολύ απλό τρόπο να οργανώνονται όλα τα στοιχεία που θέλουμε να έχουμε στην οθόνη μας χωρίς να γίνεται περίπλοκο για το χρήστη. Θα πρέπει να είναι ευχάριστη η χρήση της εφαρμογής και να φαίνεται οργανωμένη η οθόνη της.

Ανάλογα με το είδος και τις ανάγκες της εφαρμογής καταναλώνεται περισσότερη ή λιγότερη ενέργεια. Παραδείγματος χάριν, οι εφαρμογές που κάνουν χρήση σύνδεσης με το δίκτυο, καταναλώνουν περισσότερη ενέργεια και κατά συνέπεια συμβάλουν στην πιο γρήγορη εξάντληση της μπαταρίας.

Η εφαρμογή θα πρέπει να είναι γρήγορη και να μην αναγκάζει το χρήστη να περιμένει. Δίνεται προτεραιότητα σε στοιχεία που θέλουμε να εμφανίζονται πρώτα και



αφήνουμε για μετέπειτα τα υπόλοιπα. Οποιαδήποτε στιγμή διακοπεί η χρήση της εφαρμογής θα πρέπει να μη χαθούν τα μέχρι στιγμής δεδομένα και επίσης να μην καταλαμβάνουν πολύ χώρο κατά την αποθήκευσή τους.

Όταν στη συσκευή εξαντλείται η διαθέσιμη μνήμη, η λειτουργία της συσκευής κάνει επανεκκίνηση. Για να αποφευχθεί αυτό,, παρέχονται σημαντικές λειτουργίες από το Xcode οι οποίες μειώνουν σημαντικά τη χρήση της φυσικής μνήμης. Οι πόροι που δε χρησιμοποιούνται θα πρέπει να απελευθερώνονται άμεσα και εκείνοι που χρησιμοποιούνται θα πρέπει να απελευθερώνονται όταν ολοκληρώνεται η χρήση τους.

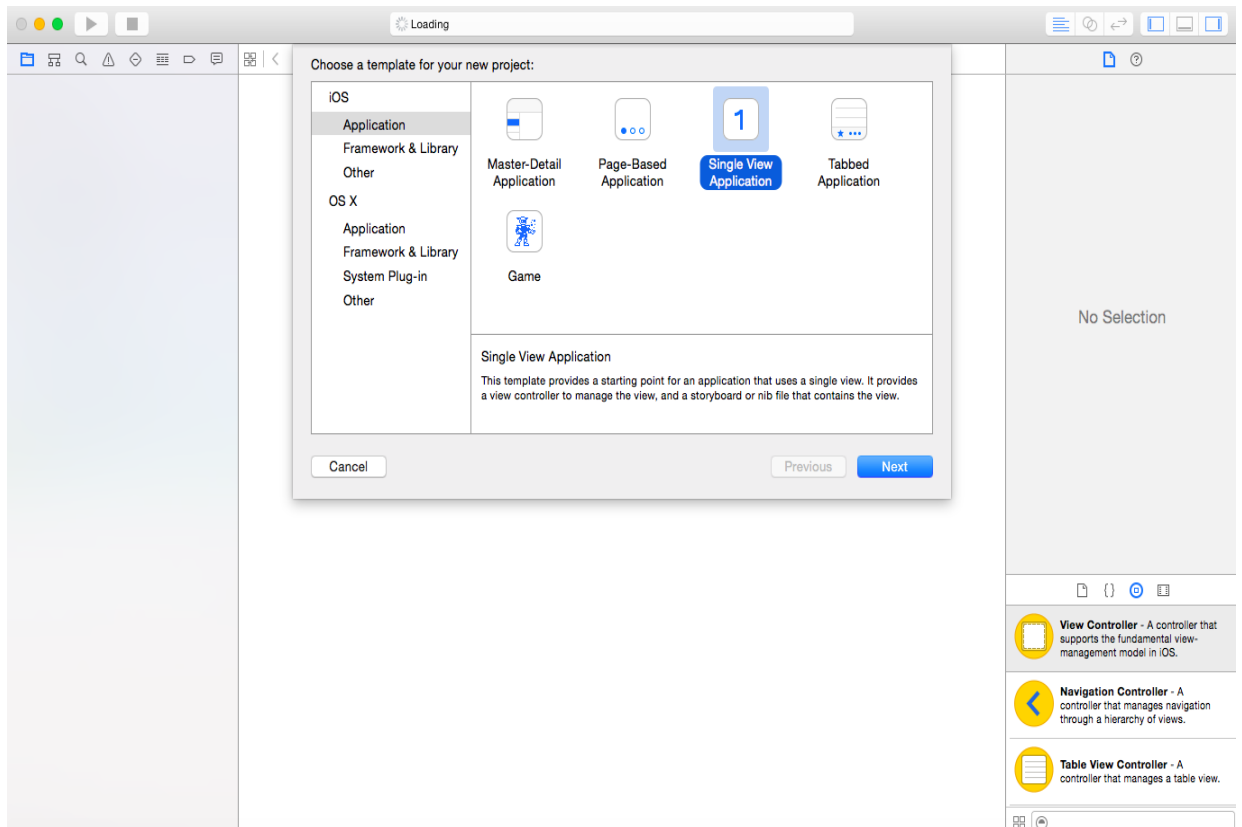
Η εφαρμογή πρέπει να έχει όσο το δυνατό πιο ασφαλή κώδικα. Με τη χρήση της νέας γλώσσας προγραμματισμού Swift, ο κώδικας είναι πιο ασφαλής σε σχέση με την Objective-C. Επίσης η ίδια η Apple στο iOS έχει πολλά επίπεδα ασφαλείας και προσπαθεί έτσι ώστε τα δεδομένα των εφαρμογών να μην αλληλοεπιδρούν.

## 3.2 Χρησιμοποιώντας το Xcode

Ξεκινώντας με το Xcode δημιουργούμε ένα καινούριο project. Η εφαρμογή κατασκευάστηκε για iOS . Έχουμε την επιλογή για:

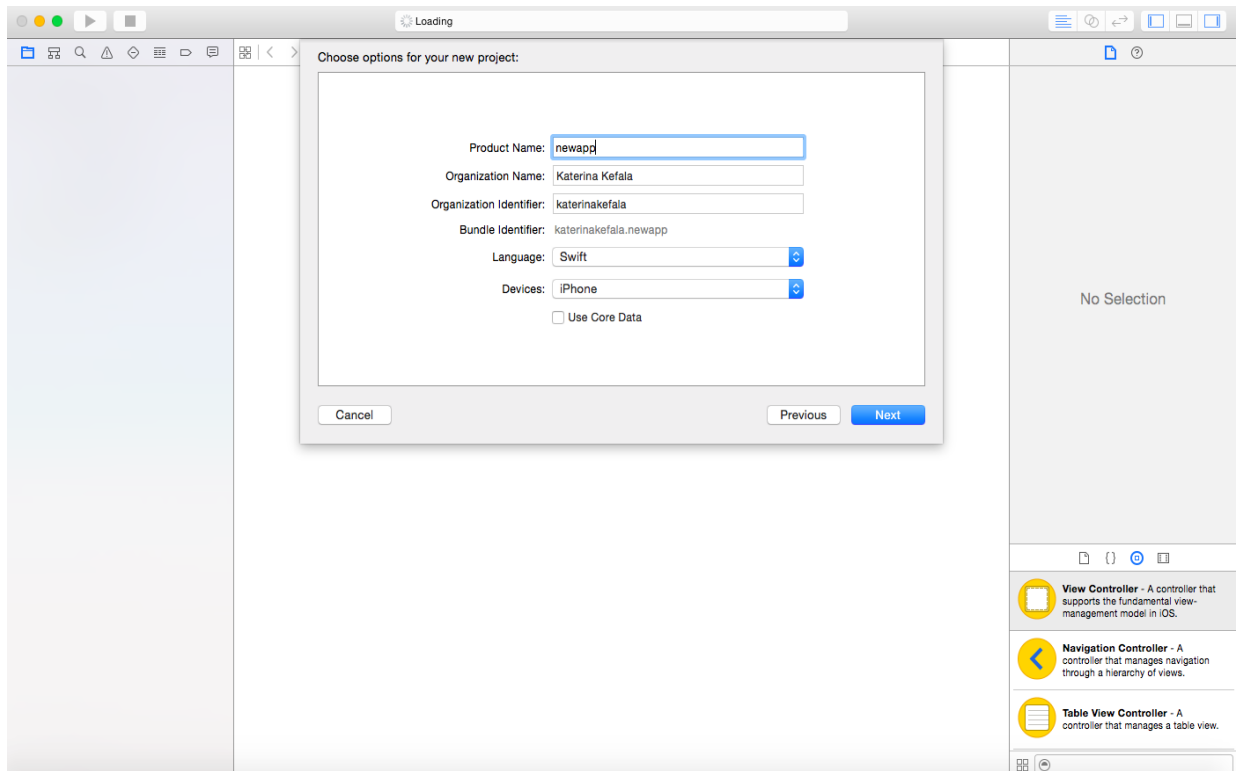
- Master-Detail Application
- Page-Based Application
- Single View Application
- Tabbed Application
- Game

Στην προκειμένη περίπτωση επέλεξα Single View Application για την εφαρμογή.



*Εικόνα 3.16: Δημιουργώντας καινούριο project στο Xcode*

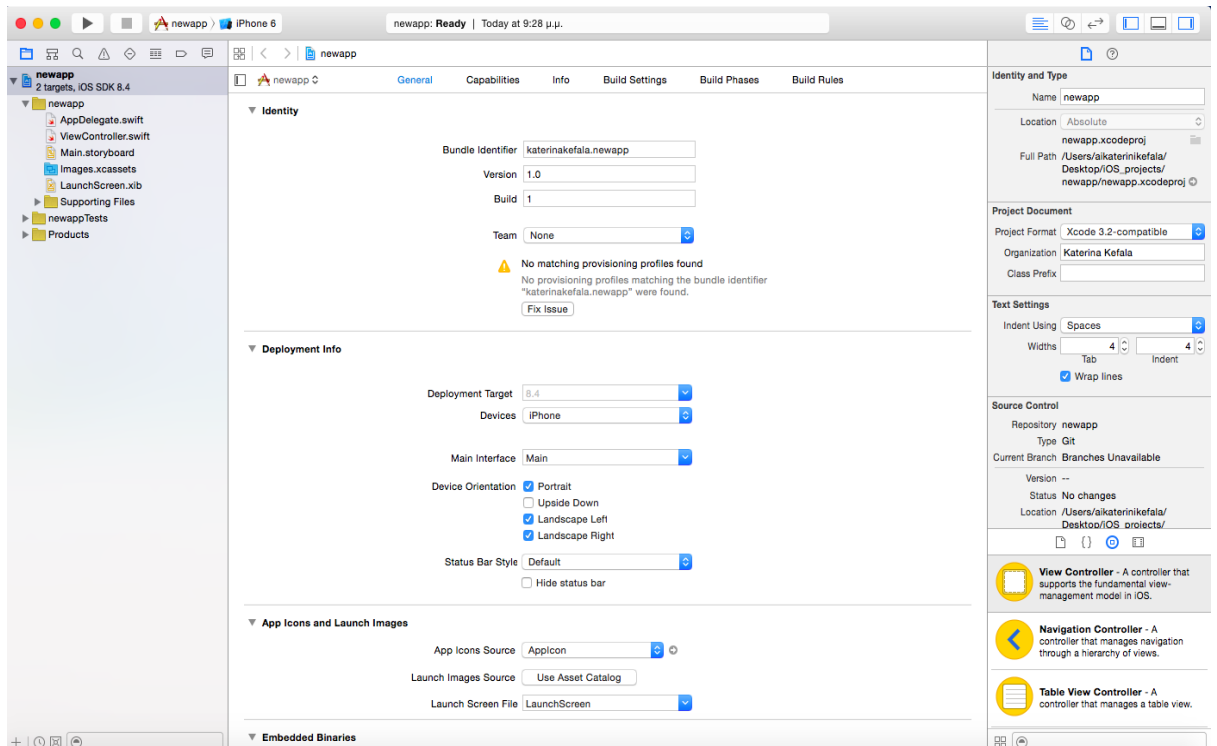
Στη συνέχεια δίνουμε ένα όνομα για το συγκεκριμένο project, έστω newapp, επιλέγουμε γλώσσα προγραμματισμού Swift και όσο για τις συσκευές επιλέγουμε iPhone.



*Εικόνα 3.17: :Ονομάζοντας το project και επιλέγοντας τα βασικά χαρακτηριστικά*

Στη συνέχεια επιλέγουμε που θέλουμε να αποθηκεύσουμε το project και επιλέγουμε δημιουργία “Create”.

Μετά από αυτό βλέπουμε την αρχική οθόνη της εφαρμογής μας με τα βασικά της χαρακτηριστικά.

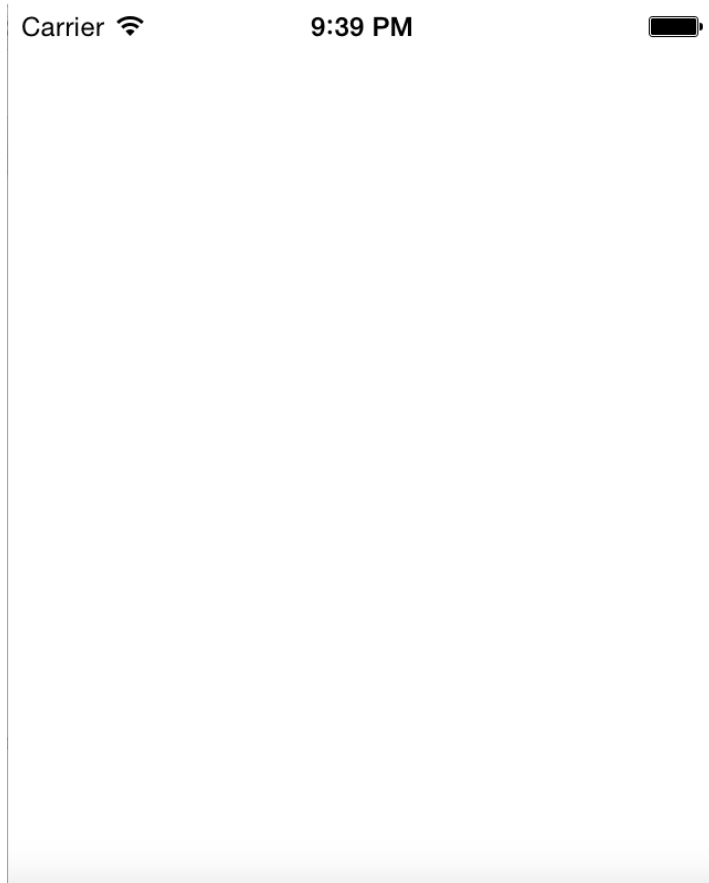


*Εικόνα 3.18: Workspace window*

Η οθόνη η οποία βλέπουμε ουσιαστικά χωρίζεται σε τέσσερα μέρη:

- Το πάνω μέρος το οποίο ονομάζεται toolbar
- Η αριστερή στήλη είναι η navigator area
- Η δεξιά στήλη είναι η utility area
- Το κεντρικό μέρος ονομάζεται editor area

Στο toolbar πάνω δεξιά βλέπουμε την επιλογή για την επιλογή του προσομοιωτή, το κουμπί run για να τρέξει το project, εναλλακτικά μπορούμε να επιλέξουμε product και run, και την περιοχή activity viewer.

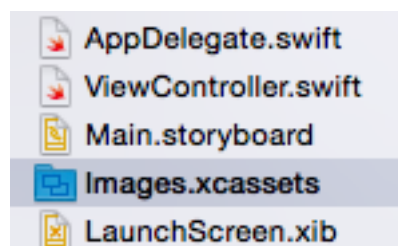


*Εικόνα 3.19: Προσομοιωτής*

Στην περιοχή project navigator φαίνονται τα αρχεία που προγράμματός μας. Για τη συγκεκριμένη εφαρμογή χρησιμοποιήθηκαν 11 αρχεία. Για την εφαρμογή επιλέχθηκε η κατηγορία tab bar. Η εφαρμογή περιέχει τρία tabs με διαφορετικές λειτουργίες.

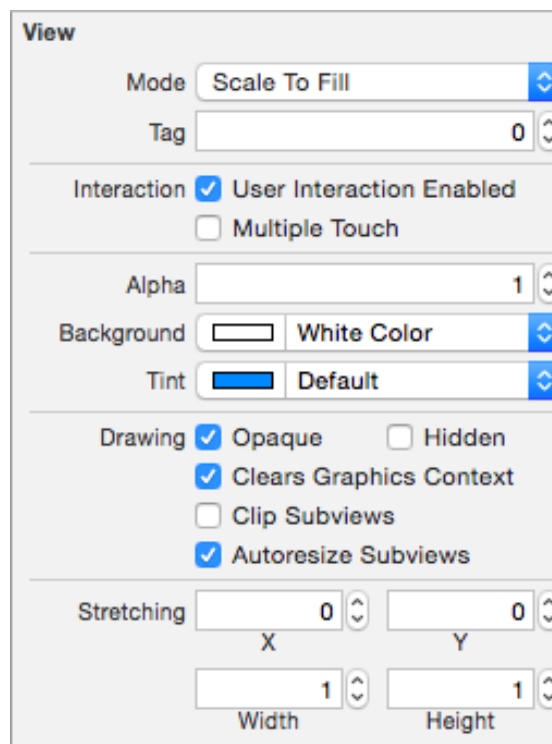
Στο Main.storyboard μπορεί να σχεδιαστεί από την αρχή η οθόνη ή οι οθόνες της εφαρμογής μας.

Επιλέγοντας στην αριστερή στήλη images μπορούμε να εισάγουμε εικόνες τις οποίες θα χρειαστούμε για την κατασκευή της εφαρμογής μας.



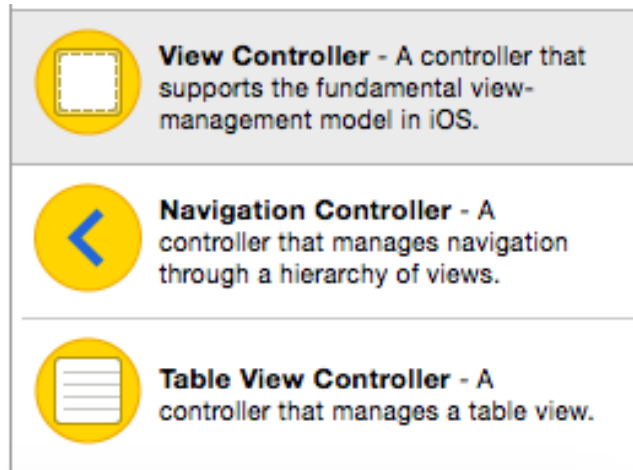
*Εικόνα 3.20: Images*

Στη δεξιά στήλη πάνω βλέπουμε το Inspector Pane απ' όπου μπορούμε να επεξεργαστούμε τα στοιχεία που έχουμε εισάγει στο main.storyboard αλλάζοντας τους για παράδειγμα γραμματοσειρά, χρώματα κλπ.



*Εικόνα 3.21: Inspector Pane*

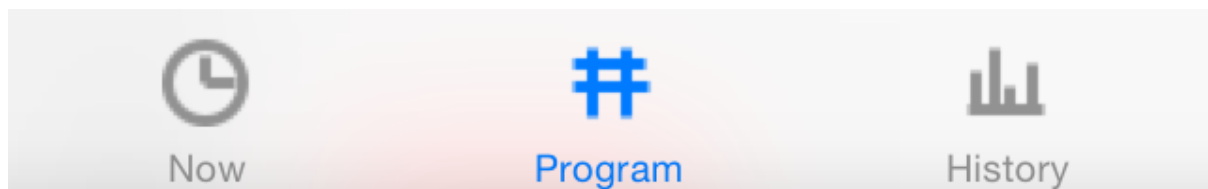
Στη δεξιά στήλη κάτω υπάρχει το Library Pane από το οποίο παίρνουμε ότι σχεδιαστικό στοιχείο θέλουμε να προσθέσουμε στην εφαρμογή μας. Από μια οθόνη, ένα view controller, έως ταμπέλες, κουμπιά και διακόπτες.



Εικόνα 3.22: Library Pane

Η εφαρμογή περιέχει 3 καρτέλες που θα περιγραφούν παρακάτω.

### 3.2.1 Πρώτη καρτέλα εφαρμογής



Εικόνα 3.23: Καρτέλα για το εβδομαδιαίο πρόγραμμα άθλησης

Η μία από αυτές τις καρτέλες αφορά το καθημερινό πρόγραμμα γυμναστικής. Πληθώρα ανθρώπων γυμνάζεται καθημερινά ή σχεδόν καθημερινά και δείχνει μια ιδιαίτερη προτίμηση στην εκγύμναση αυτή του γυμναστηρίου. Σχεδόν όλα τα γυμναστήρια παρέχουν προγράμματα τα οποία ο κόσμος δείχνει να προτιμά και κυρίως οι γυναίκες. Κάθε πρόγραμμα έχει άλλη διάρκεια και γίνεται διαφορετική καύση θερμίδων ανάλογα με το πρόγραμμα και το εκάστοτε γυμναστήριο.

Δίνεται λοιπόν η δυνατότητα στο χρήστη να οργανώσει και να αποθηκεύσει λεπτομέρειες για το πρόγραμμά του.

Ξεκινάει βλέποντας στην οθόνη του όλες τις ημέρες της εβδομάδας από τη Δευτέρα έως και την Κυριακή. Για τη σχεδίασή του χρησιμοποιήσαμε table view cells τα οποία έχουν ακριβώς τις ίδιες διαστάσεις και την ίδια αισθητική.

Daily Gym Program		
<b>1</b>	<b>Monday</b>	>
<b>2</b>	<b>Tuesday</b>	>
<b>3</b>	<b>Wednesday</b>	>
<b>4</b>	<b>Thursday</b>	>

*Εικόνα 3.24: Καρτέλα με όλες τις ημέρες της εβδομάδας*



Ο χρήστης μετακινώντας το δάχτυλό του προς τα κάτω βλέπει έως και την Κυριακή.

Πατώντας πάνω στη Δευτέρα ή σε οποιαδήποτε άλλη ημέρα ο χρήστης μετακινείται σε μια άλλη οθόνη στην οποία μπορεί να αποθηκεύσει τις λεπτομέρειες για το πρόγραμμα που παρακολουθεί εκείνη την ημέρα.

Σε κάθε γυμναστήριο μπορεί να υπάρχουν διαφορετικά προγράμματα αλλά ακόμη και τα ίδια προγράμματα να υφίστανται με άλλο όνομα ή να γίνονται διαφορετικά. Τα περισσότερα οργανωμένα γυμναστήρια παρέχουν όλες τις πληροφορίες για το εκάστοτε πρόγραμμα και φυσικά δεν μπορούν να παραλείψουν την περιγραφή και να αναφέρουν στους πελάτες πόσες θερμίδες καίνε ανάλογα με το είδος του προγράμματος, με τη δυσκολία του και με τη διάρκειά του. Για όλους αυτούς τους λόγους θεωρήθηκε πιο χρηστικό να εισάγει ο χρήστης τα βασικά στοιχεία για τα προγράμματα τα οποία παρακολουθεί.

← Back **Daily Program Details** Save

**Name:**

**Calories:**

**Start time:** 9:30 PM

6	27	
7	28	
8	29	AM
<b>9</b>	<b>30</b>	<b>PM</b>
10	31	
11	32	
12	33	

**End time:** 10:15 PM

*Εικόνα 3.25: Εισαγωγή λεπτομερειών προγράμματος για μια μέρα*

Όταν ο χρήστης πατάει στο κενό για να εισάγει το όνομα του προγράμματός του εμφανίζεται στη οθόνη ένα πληκτρολόγιο με γράμματα και αριθμούς.

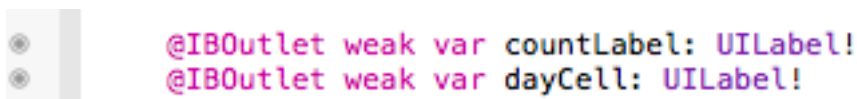
Όταν ο χρήστης πατάει πάνω στο κενό για να εισάγει τις θερμίδες που καταναλώνει εμφανίζεται ένα πληκτρολόγιο μόνο με αριθμούς.

Στη συνέχεια υπάρχουν δυο ακόμα επιλογές για να αποθηκεύσει, η ώρα που ξεκινάει το πρόγραμμα και η ώρα που τελειώνει. Η ταμπέλα δίπλα στο Start time και στο End time δείχνει μια προεπιλεγμένη ώρα η οποία αλλάζει τη στιγμή που σέρνει ο χρήστης το δάχτυλό του στην επιλογή ώρας. Κάτω από το End time υπάρχει ο ίδιος ακριβώς τρόπος με το Start time να επιλέξει ο χρήστης την ώρα. Όταν συμπληρώσει τα στοιχεία μπορεί να πατήσει Save για να αποθηκεύσει τις αλλαγές που πραγματοποίησε ή Back για να γυρίσει στην προηγούμενη οθόνη και να μην πραγματοποιήσει καμία αλλαγή. Το ίδιο μπορεί να κάνει και για τις υπόλοιπες μέρες της εβδομάδας ή όχι για όλες.

Με αυτό τον τρόπο έχει συλλέξει σε ένα σημείο το καθημερινό πρόγραμμα του γυμναστηρίου του, να ξέρει πόσες θερμίδες πρόκειται να καταναλώσει και να ανατρέχει οποιαδήποτε στιγμή θελήσει σε αυτό.

Φυσικά για όλες αυτές τις λεπτομέρειες, για τη μοναδική αποθήκευση των στοιχείων που εισάγει ο χρήστης κλπ χρειάστηκε να υλοποιηθεί κώδικας.

Στην αρχική οθόνη για παράδειγμα χρειάστηκε να δηλωθούν σε ένα αρχείο τα στοιχεία τα οποία χρησιμοποιήσαμε.



```
@IBOutlet weak var countLabel: UILabel!  
@IBOutlet weak var dayCell: UILabel!
```

*Εικόνα 3.26: Δήλωση μεταβλητών για τα κελιά των ημερών*

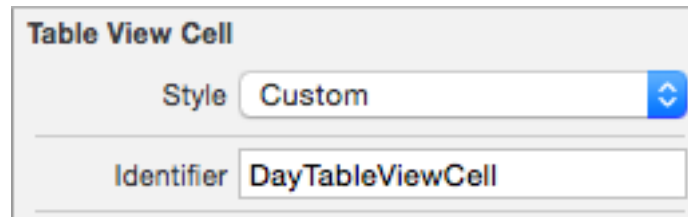
Σε άλλο αρχείο χρειάστηκε να δουλέψουμε με αυτά τα στοιχεία. Για παράδειγμα να τους δώσουμε όνομα και αρίθμηση ανάλογα με τον αριθμό της ημέρας.

```
let day1 = Day(name: "Monday", count:"1")!
```

*Εικόνα 3.27: Ορίζοντας το όνομα και την αρίθμηση*

Όταν χρησιμοποιούμε κελιά πρέπει αναγκαστικά να δώσουμε κάποιο αναγνωριστικό για τη χρήση του.

Πηγαίνοντας λοιπόν πίσω στο Main.storyboard επιλέγουμε το κελί και στη δεξιά στήλη επιλέγουμε Attributes Inspector και δίνουμε ένα όνομα στο Identifier. Στην προκειμένη περίπτωση δώσαμε το όνομα DayTableViewCell.



Εικόνα 3.28: Δίνοντας όνομα στο Identifier για το Table View Cell

Με αυτόν τον τρόπο αναγνωρίζει το κελί για το οποίο θέλουμε να κάνουμε κάποια ενέργεια όπως εδώ:

```
let cellIdentifier = "DayTableViewCell"  
  
let cell = tableView.dequeueReusableCellWithIdentifier(cellIdentifier, forIndexPath: indexPath) as! DayTableViewCell
```

Εικόνα 3.29: Χρησιμοποιώντας το Identifier DayTableViewCell

Όσον αφορά τη δεύτερη οθόνη επίσης πρέπει να συνδέσουμε τα στοιχεία τα οποία θέλουμε να χρησιμοποιήσουμε με κάποιο αρχείο και να τα δηλώσουμε.

```
@IBOutlet weak var programTextInput: UITextField!  
@IBOutlet weak var caloriesTextInput: UITextField!  
  
@IBOutlet weak var showstarttimeLabel: UILabel!  
@IBOutlet weak var showendtimeLabel: UILabel!  
  
@IBOutlet weak var startTimePicker: UIDatePicker!  
@IBOutlet weak var endTimePicker: UIDatePicker!
```

Εικόνα 3.30: Σύνδεση και δήλωση στοιχείων για λεπτομέρειες του προγράμματος

Για την αποθήκευση των μεταβλητών χρησιμοποιούμε UserDefaults.

```
let defaults: UserDefaults = UserDefaults.standardUserDefaults()
```

*Εικόνα 3.31: Χρήση UserDefaults για την αποθήκευση μεταβλητών*

Μια δυσκολία η οποία συναντήθηκε ήταν ο τρόπος με τον οποίο θα περάσουν τα δεδομένα από τη μια οθόνη στην άλλη. Πως δηλαδή η δεύτερη οθόνη θα γνωρίζει ποια μέρα επέλεξε ο χρήστης και πως η πρώτη οθόνη θα γνωρίζει σε ποια από όλες τις ημέρες εισήγαγε και αποθήκευσε ο χρήστης τις αλλαγές. Για να μπορέσει όμως αυτό να τεθεί σε λειτουργία καθώς πρόκειται για ξεχωριστά αρχεία στις δυο οθόνες, χρησιμοποιήθηκε ο εξής κώδικας στην πρώτη οθόνη:

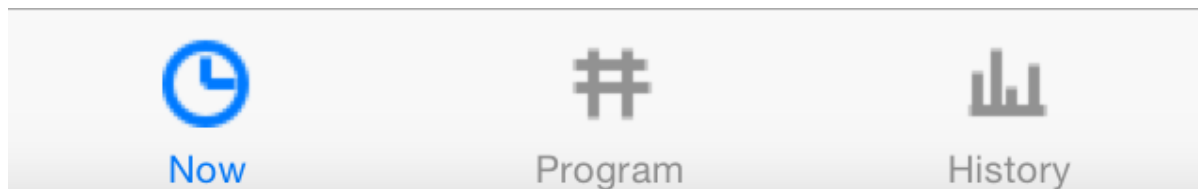
```
override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {  
    // Get the new view controller using [segue destinationViewController].  
    // Pass the selected object to the new view controller.  
    if segue.identifier == "showdetail"{  
        let dayDetailViewController = segue.destinationViewController as! DetailViewController  
        if let selectedDayCell = sender as? DayTableViewCell {  
            let indexPath = tableView.indexPathForCell(selectedDayCell)!  
            let selectedDay = days[indexPath.row]  
            dayDetailViewController.day = selectedDay.name  
        }  
    }  
}
```

*Εικόνα 3.32: Αποστολή της ημέρας που επιλέχθηκε από το χρήστη στη δεύτερη οθόνη*

Με αυτόν τον τρόπο περνάει το στοιχείο του ποια μέρα επέλεξε ο χρήστης για να μεταφερθεί στην επόμενη οθόνη και να συμπληρώσει τα στοιχεία του.

Έτσι το αρχείο στη δεύτερη οθόνη, μέσω του day καταλαβαίνει το στοιχείο που του έχει δώσει του άλλο αρχείο και με μια δήλωση if αποθηκεύει τα στοιχεία στη σωστή μέρα.

### 3.2.2 Δεύτερη καρτέλα εφαρμογής

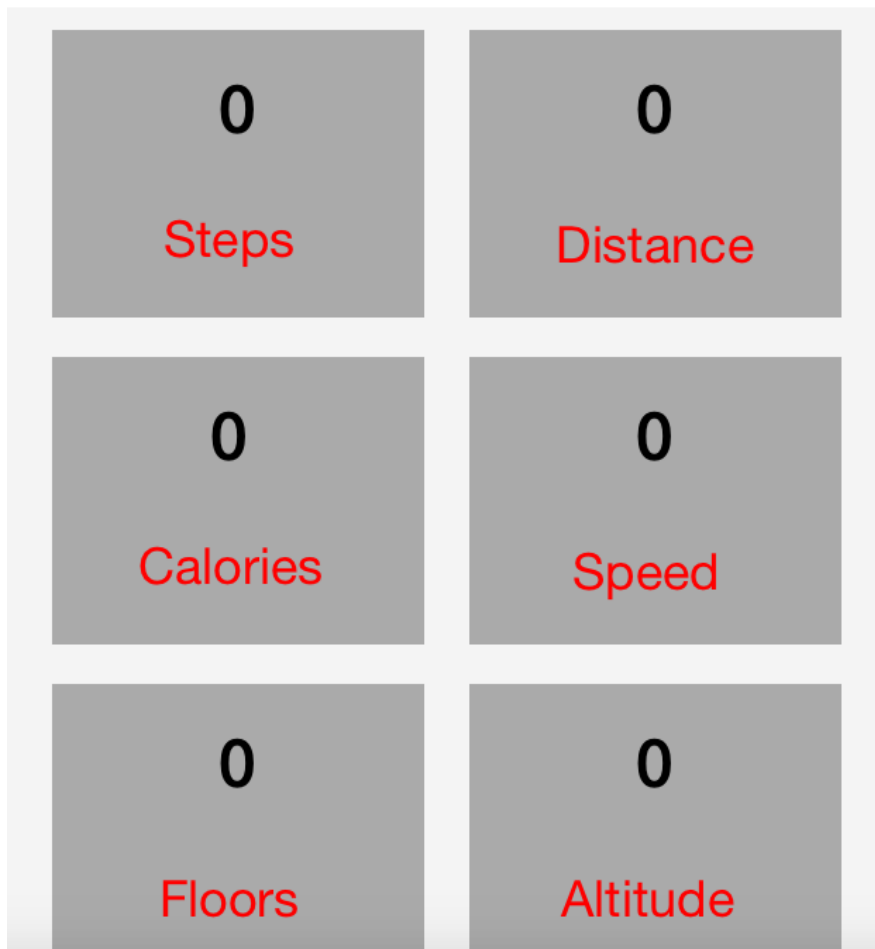


*Εικόνα 3.33: Καρτέλα για τις παραμέτρους μέτρησης σε πραγματικό χρόνο*

Στη δεύτερη καρτέλα που υπάρχει στην εφαρμογή φαίνονται στοιχεία στο χρήστη σε πραγματικό χρόνο. Τα στοιχεία αυτά αφορούν τα βήματα που κάνει ο χρήστης, την απόσταση που διανύει, με τι ταχύτητα, πόσες θερμίδες καίει, τον αριθμό των ορόφων που ανεβαίνει και κατεβαίνει και μια προσεγγιστική τιμή για τη μεταξύ τους απόσταση.

Πάνω πάνω στη θέση της ταμπέλας Exercise αναγράφεται το είδος της άσκησης που κάνει εκείνη τη στιγμή. Δηλαδή εάν περπατάει αναγράφεται η λέξη Walking, εάν τρέχει αναγράφεται η λέξη Running και εάν κάνει ποδήλατο αναγράφεται η λέξη Cycling.

# Exercise



*Εικόνα 3.34: Καρτέλα που δείχνει σε πραγματικό χρόνο τις παραμέτρους των κινήσεων του χρήστη*

Ακολουθήθηκε ο ίδιος τρόπος με πριν για να δηλώσουμε τις ταμπέλες.

Για να μπορέσουμε να μετρήσουμε πρώτα και κύρια τα βήματα του χρήστη ήταν απαραίτητο να χρησιμοποιήσουμε το framework Core Motion. Με αυτό το framework δίνεται η δυνατότητα στην εφαρμογή να λάβει δεδομένα για την κίνηση από το υλικό (hardware) της συσκευής και να τα επεξεργαστεί.

Με το framework [20] αυτό υποστηρίζεται η πρόσβαση στα δεδομένα του επιταχυνσιόμετρου χρησιμοποιώντας διασυνδέσεις block-based. Επίσης μπορούν να χρησιμοποιηθούν δεδομένα από το γυροσκόπιο για τις συσκευές που το έχουν ενσωματωμένο. Τα δεδομένα από το επιταχυνσιόμετρο και το γυροσκόπιο χρησιμοποιούνται ιδιαίτερος από εφαρμογές παιχνιδιών και από εφαρμογές που χρειάζονται τα δεδομένα αυτά ως είσοδο για να μπορούν να χρησιμοποιηθούν με επιτυχία.

Για να γίνει χρήση φυσικά του Core Motion είναι απαραίτητο να το δηλώσουμε.

```
import CoreMotion
```

*Εικόνα 3.35: Εισαγωγή του Framework Core Motion*

Απαραίτητα για τη μέτρηση των βημάτων και της απόστασης είναι η χρήση των CMPedometer και CMAltimeter. Με το CMPedometer [21] αντικείμενο ανακτώνται δεδομένα που σχετίζονται με πεζούς, δηλαδή τα βήματα που κάνει ο χρήστης, την απόσταση που διανύει και τον αριθμό των ορόφων που ανεβαίνει και κατεβαίνει. Καταφέρει να διαχειριστεί ένα χώρο προσωρινής αποθήκευσης των ιστορικών δεδομένων. Το αντικείμενο CMAltimeter [22] σχετίζεται με τα δεδομένα σε μια εφαρμογή που έχουν να κάνουν με το υψόμετρο. Αντικατοπτρίζει την αλλαγή στο τρέχον υψόμετρο και όχι στο απόλυτο υψόμετρο. Μπορεί να χρησιμοποιηθεί π.χ. από μια εφαρμογή σχετική με πεζοπορία για να ανιχνευθεί η ανύψωση κατά τη διάρκειά της.

Το core motion μεταφέρει γεγονότα από το υψόμετρο μόνο ενώ η εφαρμογή τρέχει, είτε στο προσκίνητο είτε στο παρασκήνιο. Αν η εφαρμογή ανασταλεί, η μεταφορά των γεγονότων σταματά προσωρινά και συνεχίζει όταν η εφαρμογή επιστρέψει στο προσκίνητο.

```
let pedometer = CMPedometer()  
let altimeter = CMAltimeter()
```

*Εικόνα 3.36: Δήλωση των CMPedometer και CMAltimeter*

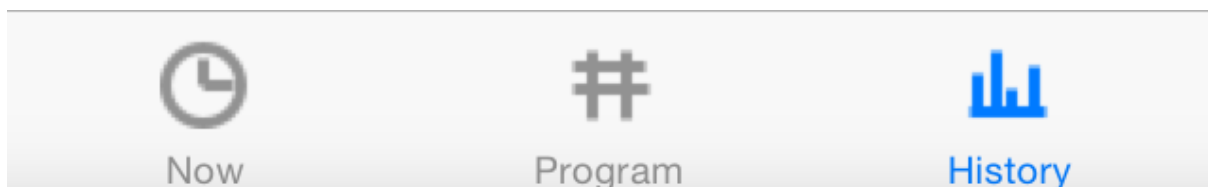


Η κύρια μέτρηση που γίνεται αφορά στα βήματα . Έπειτα γίνεται υπολογισμός για την απόσταση που διανύεται και με τι ταχύτητα. Εδώ φαίνεται ο τρόπος με τον οποίο αναπαράγονται τα στοιχεία αυτά στην καρτέλα:

```
self.stepsLabel.text = "\\(data!.numberOfSteps)"
let distance = data!.distance as! Double
self.distanceLabel.text = "\\(self.lengthFormatter.stringFromMeters(data!.distance as! Double))"
let time = data!.endDate.timeIntervalSinceDate(data!.startDate)
let speed = distance/time
self.speedLabel.text = "\\(self.lengthFormatter.stringFromMeters(speed))/s"
```

Εικόνα 3.37: Υπολογισμός και αναπαράγωγή στοιχείων μετακίνησης

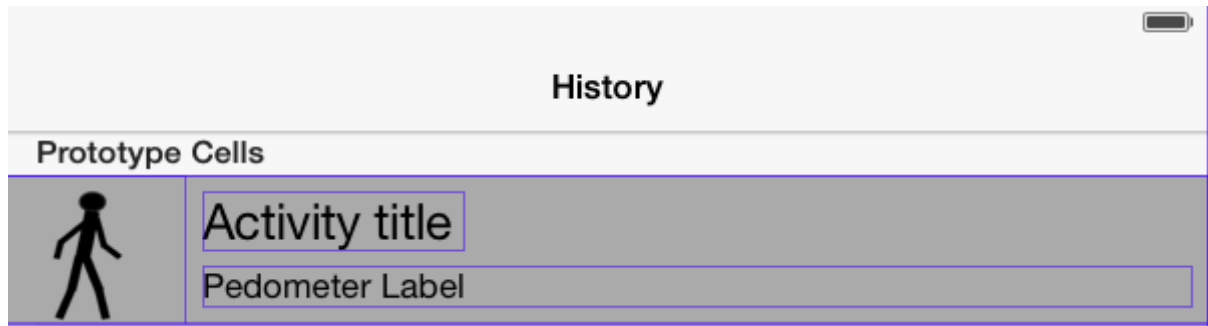
### 3.2.3 Τρίτη καρτέλα εφαρμογής



Εικόνα 38: Καρτέλα για τα συνολικά στοιχεία μέτρησης της ημέρας

Καρτέλα για τα συνολικά στοιχεία μέτρησης της ημέρας

Στην Τρίτη καρτέλα της εφαρμογής εμφανίζονται τα συνολικά βήματα που εκτέλεσε ο χρήστης εκείνη την ημέρα και με ποιο τρόπο.



*Εικόνα 3.39: Σχεδίαση της δομής των κελιών στο Main.storyboard*

Στη θέση του activity title μπαίνει το κείμενο Walking, Running και Cycling, στη θέση του Pedometer Label μπαίνουν τα βήματα τα οποία έχει κάνει και στη θέση της φωτογραφίας που βλέπουμε την εικόνα που αντιστοιχεί στη δραστηριότητα Walking, μπαίνουν οι αντίστοιχες φωτογραφίες των Running και Cycling αντίστοιχα.

Φυσικά είναι απαραίτητη και σ' αυτό το αρχείο η εισαγωγή του CoreMotion.

Η κλάση CMMotionActivityManager [23] παρέχει πρόσβαση στα δεδομένα κίνησης που αποθηκεύονται σε μια συσκευή. Τα δεδομένα κίνησης αντανακλούν τις κινήσεις του χρήστη, δηλαδή εάν περπατάει, εάν τρέχει, εάν είναι στάσιμος κλπ. Με τη χρήση αυτής της κλάσης μπορούν να ζητηθούν γνωστοποιήσεις όταν αλλάζει το είδος κίνησης ή και να συγκεντρωθούν δεδομένα από αλλαγές του παρελθόντος.

```
let motionActivityManager = CMMotionActivityManager()
```

*Εικόνα 3.40: Χρήση της κλάσης CMMotionActivityManager*

Μία ακόμη κλάση που χρησιμοποιήθηκε είναι η NSOperationQueue [24] η οποία ρυθμίζει την εκτέλεση ενός συνόλου NSOperation αντικειμένων. Μια λειτουργία, αφού προστεθεί στην ουρά, παραμένει εκεί μέχρι να ακυρωθεί ή μέχρι να ολοκληρώσει την αποστολή του. Οι διεργασίες εντός της ουράς, οργανώνονται ανάλογα με τα επίπεδα προτεραιότητας και εκτελούνται αναλόγως.

Μια εφαρμογή μπορεί να δημιουργήσει ουρές πολλαπλών λειτουργιών και να υποβάλει διεργασίες σε κάποιο από αυτά.

Ένα αντικείμενο δεν είναι έτοιμο να εκτελέσει τη διεργασία του, εκτός αν όλες οι εξαρτημένες διεργασίες του έχουν τελειώσει με την εκτέλεση.

Αφού προστεθεί μια διεργασία στην ουρά, δε γίνεται να αφαιρεθεί απ' ευθείας. Θα πρέπει να περιμένει μέχρι να ολοκληρωθεί η ακυρωθεί το έργο του. Επί του παρόντος αυτό σημαίνει ότι ο κώδικας εργασίας θα πρέπει να ελέγξει την κατάσταση ακύρωσης, να σταματήσει ότι κάνει εκείνη τη στιγμή και να θεωρηθεί ότι έχει τελειώσει.

Οι ουρές λειτουργίας συνήθως παρέχουν threads για να εκτελέσουν τις διεργασίες τους. Χρησιμοποιούν τη βιβλιοθήκη libdispatch για να αρχίσει η εκτέλεση των διεργασιών τους. Έτσι οι διεργασίες πάντα εκτελούνται σε ξεχωριστά thread, ανεξάρτητα από το εάν έχουν οριστεί σύγχρονες ή ασύγχρονες λειτουργίες.



# Κεφάλαιο 4

## Συμπεράσματα και μελλοντικοί στόχοι

Στο κεφάλαιο αυτό αναφέρονται τα συμπεράσματα τα οποία βγήκαν από την ανάπτυξη της διπλωματικής και κατ' επέκταση της εφαρμογής καθώς επίσης και οι μελλοντικοί στόχοι που υπάρχουν για τη βελτίωση της εφαρμογής.

### 4.1 Συμπεράσματα

Στην παρούσα διπλωματική εργασία υλοποίησα μια εφαρμογή καταμέτρησης διάφορων παραμέτρων άθλησης στο λειτουργικό σύστημα iOS. Για την υλοποίηση της εφαρμογής αυτής χρειάστηκε να δουλέψω με το περίπλοκο αυτό εργαλείο, το Xcode, να μελετήσω, να έρθω σε επαφή και να χρησιμοποιήσω μια νέα γλώσσα προγραμματισμού που δημιουργήθηκε μόλις την προηγούμενη χρονιά, τη Swift. Η γλώσσα αυτή βελτιώνεται όσο περνάει ο καιρός και εμφανίζονται καινούριες εκδόσεις της. Όσο περνάει ο καιρός η Swift καθιερώνεται όλο και περισσότερο και η Objective-C έχει αρχίσει να έχει καθοδική πορεία.

Πολύ βασικό πλεονέκτημα θεωρώ ότι μου δόθηκε η δυνατότητα να γνωρίσω τον τρόπο με τον οποίο λειτουργεί μια εφαρμογή για κινητή πλατφόρμα στο λειτουργικό σύστημα iOS. Κατά τη διάρκεια της εκπόνησης της εργασίας αυτής αντιμετώπισα πολλά προβλήματα και με τη γλώσσα προγραμματισμού αλλά και με το Xcode. Όμως μέσα από αυτή την εμπειρία κατάφερα να βρω λύσεις για τα περισσότερα από αυτά τα προβλήματα που παρουσιάστηκαν, να μάθω από αυτά και να είμαι περισσότερο έτοιμη για την πιθανή μελλοντική μου ενασχόληση με το αντικείμενο αυτό.

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε ένας φορητός υπολογιστής MacBook Air με OS X Yosemite και η εφαρμογή τοποθετήθηκε σε κινητή συσκευή iPhone 5S και δοκιμάστηκε επιτυχώς.

## 4.2 Μελλοντικοί στόχοι

Η εφαρμογή αυτή μπορεί να δεχτεί βελτιώσεις και προσθήκες για να γίνει ακόμα καλύτερη και να έχει κάποια επιπρόσθετα χαρακτηριστικά.

Κάποιες ιδέες για τη βελτίωση της εφαρμογής είναι:

- η προσθήκη δυνατότητας κοινοποίησης στα μέσα κοινωνικής δικτύωσης
- η εύρεση άλλων ατόμων που χρησιμοποιούν την εφαρμογή για μεγαλύτερο κίνητρο κάλυψης αποστάσεων και εκγύμνασης γενικότερα
- η προβολή μηνυμάτων επιβράβευσης για μεγάλες αποστάσεις
- η δυνατότητα ειδοποιήσεων λίγη ώρα πριν από την έναρξη του προγράμματος
- η δυνατότητα προσθήκης δεύτερου προγράμματος



## Βιβλιογραφικές αναφορές

1. <https://en.wikipedia.org/wiki/Smartphone>
2. [https://en.wikipedia.org/wiki/Tablet\\_computer](https://en.wikipedia.org/wiki/Tablet_computer)
3. [https://en.wikipedia.org/wiki/Personal\\_digital\\_assistant](https://en.wikipedia.org/wiki/Personal_digital_assistant)
4. [https://en.wikipedia.org/wiki/Mobile\\_operating\\_system](https://en.wikipedia.org/wiki/Mobile_operating_system)
5. [https://en.wikipedia.org/wiki/Palm\\_OS](https://en.wikipedia.org/wiki/Palm_OS)
6. [https://en.wikipedia.org/wiki/Windows\\_Mobile](https://en.wikipedia.org/wiki/Windows_Mobile)
7. <https://el.wikipedia.org/wiki/Android>
8. [https://el.wikipedia.org/wiki/Windows\\_Phone](https://el.wikipedia.org/wiki/Windows_Phone)
9. [https://en.wikipedia.org/wiki/BlackBerry\\_OS](https://en.wikipedia.org/wiki/BlackBerry_OS)
10. <https://en.wikipedia.org/wiki/IOS>
11. [https://en.wikipedia.org/wiki/Mobile\\_app](https://en.wikipedia.org/wiki/Mobile_app)
12. [https://en.wikipedia.org/wiki/Apple\\_Inc](https://en.wikipedia.org/wiki/Apple_Inc).
13. [https://el.wikipedia.org/wiki/Mac\\_OS\\_X](https://el.wikipedia.org/wiki/Mac_OS_X)
14. [https://en.wikipedia.org/wiki/IOS\\_SDK](https://en.wikipedia.org/wiki/IOS_SDK)
15. <https://en.wikipedia.org/wiki/Objective-C>
16. [https://en.wikipedia.org/wiki/Swift\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Swift_(programming_language))
17. <https://en.wikipedia.org/wiki/Xcode>
18. <https://en.wikipedia.org/wiki/SQLite#Development>
19. [https://en.wikipedia.org/wiki/Core\\_Data](https://en.wikipedia.org/wiki/Core_Data)
20. [https://developer.apple.com/library/ios/documentation/CoreMotion/Reference/CoreMotion\\_Reference/](https://developer.apple.com/library/ios/documentation/CoreMotion/Reference/CoreMotion_Reference/)
21. [https://developer.apple.com/library/watchos/documentation/CoreMotion/Reference/CMPedometer\\_class/index.html](https://developer.apple.com/library/watchos/documentation/CoreMotion/Reference/CMPedometer_class/index.html)
22. [https://developer.apple.com/library/watchos/documentation/CoreMotion/Reference/CMAltimeter\\_class/index.html](https://developer.apple.com/library/watchos/documentation/CoreMotion/Reference/CMAltimeter_class/index.html)
23. [https://developer.apple.com/library/ios/documentation/CoreMotion/Reference/CMMotionActivityManager\\_class/](https://developer.apple.com/library/ios/documentation/CoreMotion/Reference/CMMotionActivityManager_class/)
24. [https://developer.apple.com/library/mac/documentation/Cocoa/Reference/NSOperationQueue\\_class/](https://developer.apple.com/library/mac/documentation/Cocoa/Reference/NSOperationQueue_class/)