

Σύνθεση, Διαχείριση και Αναπαραγωγή Μουσικών Λιστών σε Φορητές Συσκευές από δεδομένα Κοινωνικών Δικτύων

Generation, Management and Playback of Musical
Playlists on Mobile Devices using Social Network data

Όνοματεπώνυμο
Αριθμός Μητρώου

Επιβλέπων Καθηγητής

Μπουγιάκας Γιώργος
ΜΕ 09066

Σγούρος Νικήτας

1 Σύνοψη

Η εργασία αυτή έχει ως αντικείμενο την υλοποίηση εφαρμογής που θα εκτελείται σε λειτουργικό σύστημα φορητών συσκευών (τηλεφώνων, ταμπλετών, κλπ.) η οποία θα αναλύει τοπικά αποθηκευμένη μουσική (στη συσκευή) θα αντλεί πληροφορία από κοινωνικά δίκτυα για τη συγκεκριμένη μουσική και θα μπορεί να δημιουργεί λίστες με τραγούδια βασισμένη στο στυλ καλλιτέχνη που θα επιλέγει ο χρήστης. Ταυτόχρονα θα παρουσιάζει πληροφορίες σχετικές με τις μουσικές επιλογές του χρήστη.

2 Περιεχόμενα

1	Σύνοψη	2
2	Περιεχόμενα	3
3	Εισαγωγή	4
4	Συστήματα Δημιουργίας Μουσικών Προτάσεων	5
4.1	Βασικά Συστατικά Μοντέλων Μουσικών Προτάσεων	9
4.1.1	User Modelling (Μοντελοποίηση Χρηστών)	9
4.1.2	Item Profiling (Μοντελοποίηση Μουσικής)	13
4.1.3	Διάφοροι Τύποι Ερωτημάτων (Queries)	14
4.2	Σύγχρονες Τεχνικές για Αυτοματοποιημένες Μουσικές Προτάσεις	15
4.2.1	Εξόρυξη Μεταπληροφορίας (Δημογραφικό Μοντέλο)	16
4.2.2	Φιλτράρισμα μέσω Συνεργασίας	16
4.2.3	Ανάκτηση Πληροφορίας μέσω του Περιεχομένου/Ήχου/Σήματος	20
4.2.4	Ανάκτηση Πληροφορίας με βάση Μοντέλα Συναισθημάτων	21
4.2.5	Ανάκτηση Πληροφορίας από Ευρύτερα Γενικά Πλαίσια (Context-Based)	24
4.2.6	Υβριδικά Μοντέλα	25
5	Last.FM	26
6	Πλατφόρμα Φορητών Συσκευών, Λειτουργικό Σύστημα	30
6.1	iOS	31
6.2	XCode & Objective-C	34
7	Frameworks & Libraries	36
7.1	Core Data & SQLite	36
7.2	Grand Central Dispatch	38
7.3	JSON & ASIHTTPRequest	40
8	Αρχιτεκτονική MVC	42
8.1	Ρόλοι & Σχέσεις των Αντικειμένων MVC	43
8.1.1	Τα Αντικείμενα Μοντέλου περιλαμβάνουν Δεδομένα & Βασικές Συμπεριφορές	44
8.1.2	Τα Αντικείμενα Προβολής παρουσιάζουν στον χρήστη Πληροφορία	45
8.1.3	Τα Αντικείμενα Ελέγχου "δένουν" το Μοντέλο με την Προβολή	47
9	Η Εφαρμογή	49
9.1	Ανάλυση Μουσικής	49
9.2	Παρουσίαση Καλλιτεχνών & των Προφίλ τους	54
9.3	Δημιουργία Μουσικών Λιστών	57
9.3.1	Αλγόριθμος Υπολογισμού Σχετικότητας	58
10	Προβλήματα κατά την Ανάπτυξη και Επίλυση τους	62
10.1	Ανομοιόμορφες εικόνες από Last.FM	62
10.2	Εγγραφή στη βάση δεδομένων από ξεχωριστά νήματα ταυτόχρονα	69
10.3	Κλήση Ασύγχρονης μεθόδου από Σύγχρονο νήμα	70
11	Μελλοντικές Βελτιώσεις	71
12	Πηγές & Βιβλιογραφία	72

3 Εισαγωγή

Πλέον στην εποχή μας οι φορητές "έξυπνες" συσκευές έχουν γίνει αναπόσπαστο εργαλείο της καθημερινότητας μας. Σχεδόν όλοι χρησιμοποιούμε τέτοιες συσκευές και το πιθανότερο απολαμβάνουμε τη μουσική της προτίμησής μας μέσω των συσκευών αυτών.

Η αποθήκευση, διανομή και αγορά ψηφιακής μουσικής είναι πλέον κάτι πολύ εύκολο και προσβάσιμο στο ευρύ κοινό. Συνεπώς ο περισσότερος κόσμος έχει εκατοντάδες ίσως και χιλιάδες τραγούδια μαζί του αποθηκευμένα σε κάποια έξυπνη συσκευή.

Έτσι προκύπτει η ανάγκη για την επιλογή των τραγουδιών που θα ακούσει ο χρήστης ανάλογα με την περίσταση ή την διάθεση του. Είναι σχεδόν βέβαιο πως η επιλογή θα πρέπει να γίνει χειροκίνητα από τον ίδιο, διαλέγοντας ένα ένα τα τραγούδια που θέλει να προσθέσει στην εκάστοτε λίστα.

Για να αυτοματοποιηθεί η διαδικασία αυτή και να προκύψουν θετικά αποτελέσματα χρειάζονται εκτενή στοιχεία για την μουσική (για κάθε τραγούδι). Καμία υπηρεσία δε θα μπορούσε να το παρέχει αυτό γιατί είναι εκατοντάδες χιλιάδες τα υπάρχοντα τραγούδια στη διεθνή μουσική σκηνή και είναι εκατοντάδες νέα που προστίθενται καθημερινά.

4 Συστήματα Δημιουργίας Μουσικών Προτάσεων

Με την ραγδαία εξάπλωση των ψηφιακών μουσικών φορμά, η διαχείριση και αναζήτηση τραγουδιών απέκτησε μεγάλη σημασία. Οι τεχνικές ανεύρεσης πληροφορίας σχετικά με τη μουσική (Music Information Retrieval - MIR) έχουν τελειοποιηθεί την τελευταία δεκαετία αλλά τα συστήματα δημιουργίας μουσικών προτάσεων είναι ακόμα σε πολύ πρώιμο στάδιο.

Σε αυτή την ενότητα θα εξετάσουμε το γενικό πλαίσιο και τις προσεγγίσεις αιχμής στις αυτοματοποιημένες μουσικές προτάσεις. Οι δύο πιο δημοφιλείς μέθοδοι που δοκιμάστηκαν στο πρόσφατο παρελθόν είναι οι:

- Collaborative Filtering (CF)
- Content-Based Model (CBM)

Ο πρώτος (CF) βασίζεται στη λογική πως πλήθος ακροατών θα φιλτράρει εκατοντάδες ή ακόμα και χιλιάδες τραγούδια και θα επιλέξει κάποια από αυτά για να τα προτείνει στους υπόλοιπους. Αυτή η μέθοδος δεν είχε ιδιαίτερα καλά αποτελέσματα γιατί η μουσική δημιουργεί έντονα πολύ διαφορετικά συναισθήματα σε κάθε ακροατή.

Έτσι η προσοχή στράφηκε στο μοντέλο CBM σε συνδυασμό με το μοντέλο EBM (Emotional-Based Model).

Τα βασικά συστατικά των μοντέλων αυτών είναι τα:

- User Modelling
- Item Profiling
- Match Algorithms

Με την ανάπτυξη των δικτύων τις τελευταίες δεκαετίες, το internet παίζει τον βασικότερο ρόλο στην ανάκτηση πολυμεσικής πληροφορίας όπως βίντεο, βιβλία, μουσική, κλπ. Οι άνθρωποι θεωρούν τη μουσική ένα σημαντικό κομμάτι της ζωής τους και το να ακούνε μουσική είναι μία από τις πιο συχνές δραστηριότητες τους. Προηγούμενες μελέτες έχουν καταλήξει στο συμπέρασμα πως οι άνθρωποι προτιμούν να ακούνε μουσική περισσότερο από οποιαδήποτε άλλη δραστηριότητα (όπως πχ το να βλέπουν τηλεόραση, να διαβάζουν βιβλία ή να βλέπουν ταινίες).

Η μουσική είναι ένα πανίσχυρο μέσο επικοινωνίας και προσωπικής έκφρασης γι' αυτό το λόγο η έρευνα γύρω από αυτή είναι μεγάλη πρόκληση.

Το βασικό πρόβλημα που αντιμετωπίζεται είναι το πως θα οργανωθούν και θα διαχειριστούν οι εκατομμύρια μουσική τίτλοι που παράγονται ακατάπαυστα. Οι διάφορες τεχνικές MIR (Music Information Retrieval) σχεδιάστηκαν για να λύσουν προβλήματα όπως η κατηγοριοποίηση της μουσικής σε είδη (genres), αναγνώριση καλλιτεχνών και ανίχνευση μουσικών οργάνων. Από το 2005 και μετά οργανώνεται το ετήσιο συνέδριο MIREX (Music Information Retrieval Evaluation eXchange) για να διευκολύνει την ανάπτυξη των τεχνικών MIR.

Ο στόχος ενός συστήματος προτάσεων μουσικής είναι να βοηθήσει τους ακροατές να φιλτράρουν και να ανακαλύψουν τραγούδια με βάση τις μουσικές τους προτιμήσεις. Ένα καλό σύστημα θα πρέπει να αναγνωρίσει αυτόματα το μουσικό γούστο του ακροατή και να δημιουργήσει μουσικές λίστες αυτόματα.

Πλέον με βάση τη μουσική συμπεριφορά των χρηστών, όλες τις αξιολογήσεις που έχουν κάνει σε μουσικά κομμάτια και τη βοήθεια CF και CBM αλγορίθμων έχουν επιτευχθεί αρκετά ενθαρρυντικά αποτελέσματα. Τα συστήματα μπορούν πλέον να παράγουν μουσικές λίστες για τους χρήστες βασισμένες σε low-level χαρακτηριστικά όπως ο ρυθμός ή το κλειδί στη φωνή ή high-level χαρακτηριστικά όπως το είδος μουσικής, τα μουσικά όργανα, κλπ.

Μερικές ιστοσελίδες συστημάτων μουσικών προτάσεων όπως το Last.FM, το Allmusic, το Pandora και το Shazam έχουν επιτυχώς υλοποιήσει τις παραπάνω μεθόδους με εντυπωσιακά αποτελέσματα.

Η μουσική είναι κάτι παγκόσμιο και υποκειμενικό. Μπορεί να μεταφέρει συναισθήματα αλλά και να τροποποιήσει τη διάθεση των ακροατών. Το μουσικό γούστο διαφέρει από ακροατή σε ακροατή συνεπώς οι παραπάνω προτάσεις δε μπορούν να καλύψουν πάντα τις ανάγκες των ακροατών.

Έτσι έχει προταθεί μια νέα υβριδική προσέγγιση που μαζί με τις παραδοσιακές μεθόδους συμπεριλαμβάνει ένα μοντέλο (Emotional-Based Model) που επιχειρεί να ανιχνεύσει του συναισθηματικούς παράγοντες του ακροατή και ένα μοντέλο (Context-Based Model) που επιχειρεί να ανιχνεύσει

πληροφορία περιεχομένου όπως σχόλια, κριτικές ή λέξεις κλειδιά από σελίδες κοινωνικής δικτύωσης (tags). Η ανάπτυξη τέτοιων υβριδικών μοντέλων είναι πολλά υποσχόμενη αλλά είναι ακόμα σε πολύ πρώιμο στάδιο. Με τις πιο πρόσφατες έρευνες στην ψυχολογία, στην επεξεργασία σήματος, στην εξόρυξη γνώσης από υπολογιστές και τη μουσικολογία υπάρχει μεγάλο περιθώριο για ανάπτυξη των τεχνικών στο κοντινό μέλλον.

Πανεπιστήμιο Πειραιώς

4.1 Βασικά Συστατικά Μοντέλων Μουσικών Προτάσεων

4.1.1 User Modelling (Μοντελοποίηση Χρηστών)

Ένα επιτυχημένο σύστημα πρέπει να είναι ικανό να καλύψει τις διαφορετικές απαιτήσεις των χρηστών του. Δυστυχώς, το να αποκτηθεί πληροφορία για τον εκάστοτε χρήστη είναι ιδιαίτερα δαπανηρό και σε υλικούς (επεξεργαστική δύναμη, χώρος αποθήκευσης, κλπ.) αλλά και σε ανθρώπινους πόρους.

Η μοντελοποίηση χρηστών είναι ένα από τα πιο βασικά συστατικά. Μοντελοποιεί τη διαφορετικότητα του προφίλ του κάθε χρήστη/ακροατή. Για παράδειγμα οι χρήστες μπορεί να διαφέρουν στις προτιμήσεις τους ανάλογα με τη γεωγραφική περιοχή ή την ηλικία τους. Άλλοι παράγοντες που επηρεάζουν τις μουσικές προτιμήσεις είναι το φύλο, ο τρόπος ζωής του κάθε ακροατή ακόμα και τα υπόλοιπα ενδιαφέροντα του.

Πρόσφατες μελέτες έδειξαν πως η ευφυΐα, η προσωπικότητα και η μουσική προτίμηση είναι έννοιες συνδεδεμένες. Σύμφωνα με τον Rentfrow & Gosling που μελέτησαν τη σχέση μεταξύ των μουσικών προτιμήσεων και του Big-Five Inventory (BFI: δεκτικότητα στην εμπειρία, ευσυνειδησία, εξωστρέφεια, συγκαταβατικότητα και νευρωτισμό), οι έρευνες τους έδειξαν πως ένα εξωστρεφές άτομο θα τείνει να επιλέξει μουσική που είναι πιο ενεργητική και ρυθμική από ένα εσωστρεφές άτομο.

Συνεπώς η μοντελοποίηση των χρηστών / ακροατών είναι απαραίτητη για την σωστή πρόβλεψη των μουσικών τους προτιμήσεων. Έχει χωριστεί σε δύο βασικά μέρη, τη μοντελοποίηση του προφίλ των χρηστών και τη μοντελοποίηση των εμπειριών των χρηστών.

4.1.1.1 Βήμα 1: Μοντελοποίηση του Προφίλ των Χρηστών

Ο O. Celma πρότεινε την κατηγοριοποίηση των προφίλ των χρηστών σε 3 τομείς: τον δημογραφικό, τον γεωγραφικό και τον ψυχογραφικό.

Τομέας	Παράδειγμα
Δημογραφικός	Ηλικία, Φύλο, Οικογενειακή Κατάσταση
Γεωγραφικός	Τοποθεσία, πόλη, χώρα, κλπ.
Ψυχογραφικός	Στατικές: ενδιαφέροντα, τρόπος ζωής, προσωπικότητα, κλπ. Μεταβλητές: διάθεση, συμπεριφορά, γνώμη, κλπ.

Με βάση τη σταθερότητα ο ψυχογραφικός τομέας έχει στατικά χαρακτηριστικά που βοηθούν στις μακροχρόνιες προβλέψεις και τα μεταβλητά χαρακτηριστικά που βοηθούν σε προβλέψεις που μπορούν να αλλάξουν ώρα με την ώρα.

4.1.1.2 Βήμα 2: Μοντελοποίηση της Εμπειρίας των Χρηστών

Ανάλογα με την εμπειρία του κάθε χρήστη, οι προσδοκίες τους στη μουσική διαφέρουν. Ο Jennings ανέλυσε τους διαφορετικούς τύπους ακροατών σε ηλικία από 16-45 ετών και τους ενέταξε σε τέσσερις κατηγορίες: εξειδικευμένοι, ενθουσιώδεις, ανέμελοι και αδιάφοροι.

Κατηγορία	Ποσοστό	Περιγραφή
Εξειδικευμένοι	7	Τα πάντα στη ζωή τους είναι συνδεδεμένα με τη μουσική. Οι γνώσεις τους στη μουσική είναι διευρυμένες.
Ενθουσιώδεις	21	Η μουσική παίζει σημαντικό ρόλο στη ζωή τους αλλά είναι εξισορροπημένη και με άλλα ενδιαφέροντα.
Ανέμελοι	32	Η μουσική είναι καλοδεχούμενη, αλλά άλλα πράγματα είναι πολύ πιο σημαντικά.
Αδιάφοροι	40	Ακόμα και αν μέσα σε μια νύχτα η μουσική αποτελούσε παρελθόν και έπαψε να ισχύει δε θα τους απασχολούσε ιδιαίτερα. Το μεγαλύτερο ποσοστό των ανθρώπων ανήκει σε αυτή την κατηγορία.

Η παραπάνω πληροφορία εξηγεί γιατί η εμπειρία των χρηστών πρέπει να ληφθεί σοβαρά υπ' όψη κατά τον σχεδιασμό συστημάτων αυτοματοποιημένων μουσικών προτάσεων. Για παράδειγμα με βάση τις μουσικές προσδοκίες, πρέπει να αναλογιστούμε το πόσο εκτενής και βαθιά θα είναι η ανάλυση ή οποία όσο πιο βαθιά φτάσει μπορεί να εξορύξει μουσική ιδιαίτερα ενδιαφέρουσα αλλά ελάχιστα δημοφιλή. Κάτι τέτοια θα ήταν καλοδεχούμενο από την 1η κατηγορία χρηστών αλλά πιθανώς να απογοήτευε τις υπόλοιπες.

Η ανάλυση της εμπειρίας των χρηστών μπορεί να επιτευχθεί είτε από μία αρχική έρευνα ή από τη συνεχή παρακολούθηση της μουσικής συμπεριφοράς και συνηθειών των χρηστών.

4.1.2 Item Profiling (Μοντελοποίηση Μουσικής)

Το δεύτερο σημαντικό στοιχείο ενός συστήματος μουσικών προτάσεων είναι η μοντελοποίηση του κάθε κομματιού μουσικής. Το 2005 ο Patches κατηγοριοποίησε τα μέτα-στοιχεία (metadata) για τη μουσική σε 3 κατηγορίες: Editorial metadata, cultural metadata & acoustic metadata.

Κατηγορία	Περιγραφή
Editorial Metadata	Στοιχεία που αποκτήθηκαν από κάποιον ειδικό ή από ομάδα ειδικών. Αυτά είναι τα στοιχεία που διατίθενται από τον δημιουργό, στοιχεία όπως ο συνθέτης, ο τίτλος, το είδος κλπ.
Cultural Metadata	Στοιχεία που αποκτήθηκαν αναλύοντας μεγάλο όγκο πληροφορίας κειμένου συνήθως από το Internet. Αυτά τα στοιχεία περιέχουν συσχετίσεις μεταξύ τραγουδιών ή καλλιτεχνών, επαναλαμβανόμενα φαινόμενα, κατηγορίες κλπ.
Acoustic Metadata	Στοιχεία που αποκτούνται από την ανάλυση του μουσικού σήματος (της ηχητικής κυματομορφής). Στοιχεία που δεν προκύπτουν από καμιά πηγή κειμένου όπως ρυθμός, μουσικά όργανα, κλπ.

Η πληθώρα των μέχρι τώρα συστημάτων χρησιμοποιούν την 3η κατηγορία για την ανακάλυψη νέας μουσικής και την πρόταση της στους χρήστες.

4.1.3 Διάφοροι Τύποι Ερωτημάτων (Queries)

Αν υποθέσουμε πως οι χρήστες ήδη έχουν την πληροφορία για τη μουσική τους, ο πιο γρήγορος τρόπος να αναζητήσουν μουσική είναι με το να εισάγουν στο σύστημα βασικά editorial metadata όπως πχ. το όνομα του καλλιτέχνη, στίχους, κλπ. Παρ' όλ' αυτά αυτή υπάρχει και η περίπτωση η παραπάνω πληροφορία να μην είναι γνωστή.

Τα τελευταία δέκα χρόνια έχουν αναπτυχθεί τεχνολογίες από τις οποίες δημιουργήθηκαν πολύπλοκα συστήματα αναζήτησης μουσικής που ο χρήστης αντί να εισάγει πληροφορίες γι' αυτό που ψάχνει μπορεί να τραγουδήσει ή να αναπαράγει με τη φωνή του τη μελωδία του τραγουδιού και το σύστημα να το εντοπίσει. Αυτά τα συστήματα ονομάζονται "Query by humming / singing system (QBSH)". Ακόμα και έτσι, αυτό απαιτεί αρκετή προσπάθεια από τον χρήστη και δεν έχει πάντα σωστά / επιτυχή αποτελέσματα.

4.2 Σύγχρονες Τεχνικές για Αυτοματοποιημένες Μουσικές Προτάσεις

Ένα ιδανικό σύστημα μουσικών προτάσεων θα έπρεπε να είναι ικανό να προσφέρει στο χρήστη αυτόματα μουσικές προτάσεις φτιαγμένες αποκλειστικά σε σχέση με τις προσωπικές μουσικές του προτιμήσεις. Σε αντίθεση με τα βιβλία ή τις ταινίες, ένα μουσικό τραγούδι είναι πολύ μικρότερο σε διάρκεια και ένας χρήστης θα το ακούσει παραπάνω από μία φορές.

Τα ήδη υπάρχοντα συστήματα προτάσεων (όχι για μουσική) όπως αυτό του Amazon ή του EBay έχουν γνωρίσει μεγάλη επιτυχία. Μπορούν να προτείνουν συμπληρωματικά προϊόντα, ο αγοραστής μπορεί να κάνει σύγκριση μεταξύ των αγαθών ακόμα και να επικοινωνήσει με τον πωλητή. Ένα σύστημα μουσικών προτάσεων διαφέρει κατά πολύ γιατί δεν προτείνει προϊόντα σε ανταγωνιστικές τιμές (πχ μια ηλεκτρονική συσκευή) αλλά τραγούδια και λίστες άρρηκτα συνδεδεμένα με τις προσωπικές προτιμήσεις του χρήστη.

Μέχρι τώρα πληθώρα από ιστοσελίδες "μουσικών αναζητήσεων/ανακαλύψεων" όπως το Last.FM, το Allmusic, το Pandora, το Audiobaba, το Mog, το Musicoverly, το Spotify, το Apple Genius έχουν αναλύσει εκατομμύρια χρήστες και ο όγκος των δεδομένων που συλλέγονται ανά πάσα στιγμή είναι γιγαντώδης.

Παρακάτω θα αναλυθούν διάφορες μέθοδοι εξόρυξης πληροφορίας που χρησιμοποιούνται από τα πιο δημοφιλή συστήματα.

4.2.1 Εξόρυξη Μεταπληροφορίας (Δημογραφικό Μοντέλο)

Η βασικότερη και πιο απλοϊκή μέθοδος. Χρησιμοποιούνται τα στοιχεία που έχουν δοθεί από τον παραγωγό / δημιουργό όπως ο τίτλος του τραγουδιού, το όνομα του καλλιτέχνη, οι στίχοι κλπ. ώστε να βρεθούν τα ζητούμενα τραγούδια.

Περιορισμοί: Παρ' όλο που είναι γρήγορη και ακριβής μέθοδος, τα μειονεκτήματα είναι προφανή. Πρώτα απ' όλα ο χρήστης πρέπει να γνωρίζει την "εκδοτική πληροφορία" του τραγουδιού που αναζητά. Κατά δεύτερον είναι πολύ χρονοβόρο για κάποιον χρήστη να συντηρεί όλη αυτή την πληροφορία για την υπάρχουσα και τη νέα μουσική που αποκτάει. Τέλος πέρα από την αναζήτηση ενός συγκεκριμένου τραγουδιού δε μπορεί να προσφέρει αληθινές προτάσεις γιατί η γνώση είναι περιορισμένη στην "εκδοτική πληροφορία" ενώ οι μουσικές προτιμήσεις και το προφίλ του χρήστη δε λαμβάνονται καθόλου υπ' όψη.

4.2.2 Φιλτράρισμα μέσω Συνεργασίας

Το φιλτράρισμα μέσω συνεργασίας αναπτύχθηκε για να καταστεί δυνατή η πρόταση μουσικής σε ένα χρήστη με βάση τις προτάσεις άλλων χρηστών με κοινό μουσικό γούστο. Από τους πιο γνωστούς αλγόριθμους που έχουν χρησιμοποιηθεί υποθέτει πως οι μουσικές προτιμήσεις του χρήστη X με τον χρήστη Ψ είναι συναφείς όσο πιο πολλά κοινά τραγούδια και καλλιτέχνες έχουν στη συλλογή τους.

Αντί να υπολογίζει τη συνάφεια μεταξύ τραγουδιών δημιουργεί ένα οικοσύστημα μουσικών "γειτόνων" ανάλογα με το τι κοινά στοιχεία έχουν οι βιβλιοθήκες των χρηστών. Το σκορ για τα τραγούδια που δεν έχει ανακαλύψει ο Χ χρήστης και πρόκειται να του προταθούν προκύπτει από έναν συνδυασμό του σκορ που τους έχουν αποδώσει οι κοντινότεροι "γείτονες".

Το φιλτράρισμα μέσω συνεργασίας έχει υποκατηγοριοποιηθεί στις παρακάτω τρεις κατηγορίες.

4.2.2.1 Φιλτράρισμα μέσω Συνεργασίας βασισμένο στη Μνήμη

Το φιλτράρισμα μέσω συνεργασίας βασισμένο στη μνήμη υπολογίζει το τραγούδι που θα προταθεί λαμβάνοντας υπ' όψη **ολόκληρη** τη μουσική συλλογή από προηγούμενες αξιολογήσεις που έχει δώσει ο εκάστοτε χρήστης. Κάθε χρήστης κατηγοριοποιείται με χρήστες παρόμοιου γούστου και το νέο τραγούδι (προς πρόταση) προκύπτει βρίσκοντας τον κοντινότερο "γείτονα" χρησιμοποιώντας έναν τεράστιο αριθμό από ψήφους χρηστών.

4.2.2.2 Φιλτράρισμα μέσω Συνεργασίας βασισμένο σε Μοντέλα

Σε αντίθεση με το φιλτράρισμα βασισμένο στη Μνήμη, το φιλτράρισμά που βασίζεται σε μοντέλα χρησιμοποιεί αλγόριθμους εξόρυξης δεδομένων και εκπαιδεύει το σύστημα μοντελοποιώντας τις μουσικές προτιμήσεις και συμπεριφορές του χρήστη. Έπειτα χρησιμοποιώντας το μοντέλο του κάθε

χρήστη δημιουργεί προτάσεις και προβλέψεις. Το μοντέλο εξελίσσεται και ενημερώνεται συνεχώς.

4.2.2.3 Υβριδικό Φιλτράρισμα μέσω Συνεργασίας

Στη συγκεκριμένη περίπτωση συνδυάζονται παραπάνω από μία μέθοδοι και η διαφορά στα αποτελέσματα είναι μεγάλη. Τα αποτελέσματα και οι προτάσεις που προκύπτουν είναι πιο επιτυχείς και ακριβείς

Περιορισμοί του Φιλτραρίσματος μέσω Συνεργασίας:

Εξαιτίας τις υποκειμενικότητας που διέπει τη μουσική, η θεωρία πως οι χρήστες με παρόμοιες μουσικές συμπεριφορές θα έχουν και παρόμοιο μουσικό γούστο και προτιμήσεις δεν έχει δοκιμαστεί διεξοδικά. Παρόλο που έως τώρα τα αποτελέσματα είναι ενθαρρυντικά προκύπτουν διάφορα προβλήματα.

- **Μεροληψία λόγω δημοτικότητας:** Συνήθως ή δημοφιλέστερη μουσική θα λάβει περισσότερες αξιολογήσεις. Η πιο σπάνια μουσική μπορεί να μη λάβει και καμία αξιολόγηση. Σαν αποτέλεσμα το φιλτράρισμα μέσω συνεργασίας συνήθως προτείνει μόνο την δημοφιλή μουσική στους χρήστες του. Παρόλο που οι προτάσεις ταιριάζουν ο χρήστης σπάνια θα εκπλαγεί θετικά ανακαλύπτοντας κάτι νέο.

- **Αργή Αρχή:** Το πρόβλημα είναι επίσης γνωστό ως αρχική απουσία δεδομένων. Όταν ένας χρήστης ξεκινά να χρησιμοποιεί το σύστημα τα αποτελέσματα θα είναι πολύ περιορισμένα έως και μηδενικά.
- **Ανθρώπινη Προσπάθεια:** Ένα τέλειο σύστημα προτάσεων δε θα έπρεπε να απαιτεί ανθρώπινη προσπάθεια μιας και οι χρήστες δεν διατίθενται πάντα να αξιολογήσουν ή να ψηφίσουν τις μουσικές τους επιλογές. Επίσης οι αξιολογήσεις μπορεί αμερόληπτα να ενισχυθούν από το μουσικό γούστο μόνο αυτών που ψηφίζουν αλλά αυτό δεν είναι απαραίτητα αντιπροσωπευτικό. Λόγω της έλλειψης ισόποσης κατανομής αξιολογήσεων στους χρήστες οι μέθοδοι αυτοί μπορεί να προβλέψουν και να προτείνουν λανθασμένα αποτελέσματα (false-positives, false-negatives).

4.2.3 Ανάκτηση Πληροφορίας μέσω του Περιεχομένου/Ήχου/Σήματος

Η συγκεκριμένη προσέγγιση είναι θεμελιακά διαφορετική από τις προηγούμενες. Η πρόβλεψη γίνεται με την ανάλυση του ίδιου του ήχου, της αναλογικής κυματομορφής του εκάστοτε τραγουδιού και όχι από μεταπληροφορίες (είτε από τον δημιουργό είτε από τους ακροατές). Η ιδέα είναι να βρεθεί τραγούδι(α) που ηχητικά θα μοιάζουν με κάτι που έχει ακούσει ο χρήστης στο παρελθόν και όχι με κάτι που έχει χαρακτηρίσει ο χρήστης πως του "άρεσε". Έχει ήδη γίνει εκτεταμένη έρευνα για το ποια χαρακτηριστικά μιας κυματομορφής μπορούν να εξορυχθούν και να είναι συγκρίσιμα (πχ ρυθμός, τέμπο). Με βάση αυτά τα χαρακτηριστικά υπολογίζεται η "απόσταση" που έχει ένα τραγούδι από ένα άλλο.

Περιορισμοί:

Σε κάποιο ποσοστό οι συγκεκριμένες μέθοδοι λύνουν το πρόβλημα που προκύπτει από τις μεθόδους φιλτραρίσματος μέσω συνεργασίας. Για παράδειγμα με το να συγκρίνονται ακουστικά χαρακτηριστικά και να υπολογίζεται η "απόσταση" μεταξύ των τραγουδιών δε χρειάζεται καμία παρέμβαση ή πληροφορία από τους χρήστες.

Τα βασικά προβλήματα που προκύπτουν είναι δύο. Πρώτον το ότι δυο τραγούδια ταιριάζουν μεταξύ τους σε ακουστικά χαρακτηριστικά δε σημαίνει απαραίτητα πως συμπίπτουν και τα δύο με τις μουσικές προτιμήσεις του ακροατή.

Δεύτερον υπάρχει μεγάλο περιθώριο για έρευνα στο **ποια** είναι τα ακουστικά χαρακτηριστικά που μπορούμε να ανιχνεύσουμε και να συγκρίνουμε

σε ένα τραγούδι και κατά πόσο αυτά αρκούν στο να υπολογιστεί η ακριβής "απόσταση".

4.2.4 Ανάκτηση Πληροφορίας με βάση Μοντέλα Συναισθημάτων

Η μουσική ήταν πάντα ένα εργαλείο προσωπικής έκφρασης. Πλούσια σε περιεχόμενο και συναισθηματικές εκφράσεις την καθιστούν δύσκολο τομέα για να ανακτηθεί πληροφορία με συμβατικές μεθόδους. Η έρευνα πάνω στο συναίσθημα που προκαλεί η μουσική είναι εκτενής και πλέον αποτελεί τον δημοφιλέστερο και πιο μοντέρνο τρόπο για τα συστήματα μουσικών αναζητήσεων και προτάσεων. Μια εμπορική υπηρεσία διαδικτύου (Musicoverly) χρησιμοποιεί το θεμελιώδες μοντέλο συναισθημάτων (2D valence-arousal) που έχει καθιερωθεί στην ψυχολογία. Επιτρέπει στους χρήστες να εντοπίσουν τη συναισθηματική τους κατάσταση σε ένα διδιάστατο περιβάλλον με τον έναν άξονα να κινείται στο θετικό ή αρνητικό και τον άλλον στο ενθουσιώδες ή ήρεμο.

Παρόμοιο με το μοντέλο ανάλυσης ήχου έτσι και το μοντέλο ανάλυσης συναισθημάτων εντοπίζει συναισθηματικά χαρακτηριστικά όπως ενέργεια, ρυθμός, τέμπο, αρμονία κλπ. για να υπολογίσει "αποστάσεις" μεταξύ των τραγουδιών.

Περιορισμοί:

Η Συλλογή Δεδομένων. Για να λειτουργήσει το συγκεκριμένο μοντέλο χρειάζεται να έχει αναλυθεί ένας τεράστιος όγκος δεδομένων. Για να βρεθεί μια αξιόπιστη βάση απαιτείται μεγάλη ανθρώπινη προσπάθεια και πρέπει να

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ - ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΔΙΚΤΥΟΚΕΝΤΡΙΚΑ ΣΥΣΤΗΜΑΤΑ

εξεταστεί από ανθρώπους το κάθε προτεινόμενο αποτέλεσμα για να γίνουν οι απαραίτητες τροποποιήσεις ώστε να είναι αποτελεσματικό.

Σχετικότητα και Ορισμοί. Το συναίσθημα είναι δύσκολο να οριστεί και να περιγραφθεί. Δυο διαφορετικοί άνθρωποι μπορούν να εκφράσουν το ίδιο συναίσθημα με διαφορετικούς όρους και δεν υπάρχει τέλεια ορισμένη σχέση μεταξύ δυο όρων που περιγράφουν κάποιο συναίσθημα. Υπάρχουν βέβαια έρευνες που επιχειρούν να ταξινομήσουν τους όρους που εκφράζουν συναίσθημα σε ομάδες και κατηγορίες αλλά είναι δύσκολο να καταγράψουν τον πλούτο της ανθρώπινης αντίληψης και έκφρασης. Το πιο επιτυχημένο μοντέλο έχει αναπτυχθεί από τον Russell και εκφράζει τις βασικές ανθρώπινες συναισθηματικές καταστάσεις σε έναν κύκλο. Ευχαρίστηση (0 μοίρες), Ενθουσιασμός (45 μοίρες), Ερεθισμός (90 μοίρες), Άγχος (135 μοίρες), Δυσaréστηση (180 μοίρες), Κατάθλιψη (225 μοίρες), Υπνηλία (270 μοίρες) και Ανάπαυση (315 μοίρες).

Άλλη μια διαδεδομένη κατηγοριοποίηση συναισθημάτων είναι αυτή του MIREX ή οποία ομαδοποιεί συναισθήματα σε 5 μεγάλες κατηγορίες.

Ομάδα 1	passionate, rousing, confident, boisterous, rowdy
Ομάδα 2	rollicking, cheerful, fun, sweet, amiable/good natured
Ομάδα 3	literate, poignant, wistful, bittersweet, autumnal, brooding
Ομάδα 4	humorous, silly, campy, quirky, whimsical, witty, wry
Ομάδα 5	aggressive, fiery, tense/anxious, intense, volatile, visceral

Πανεπιστήμιο Πειραιώς

4.2.5 Ανάκτηση Πληροφορίας από Ευρύτερα Γενικά Πλαίσια (Context-Based)

Η συγκεκριμένη μέθοδος δεν χρησιμοποιεί ούτε ακουστικά χαρακτηριστικά ούτε βαθμολογίες και χαρακτηρισμούς αποδομένους από τους χρήστες. Αντίθετα κάνει χρήση της "κοινής γνώμης" ψάχνοντας σε έναν τεράστιο όγκο πληροφορίας που παρέχουν οι υπηρεσίες κοινωνικής δικτύωσης (facebook, twitter, etc.) και το διαδίκτυο γενικότερα (blogs, web sites) όπου συλλέγεται η συμφραζόμενη πληροφορία όταν αναφέρεται κάπου ο καλλιτέχνης ή το άλμπουμ ή το τραγούδι. Η συμφραζόμενη πληροφορία μπορεί να είναι κριτικές, σχόλια χρηστών, λέξεις κλειδιά, κλπ.

Γίνεται εκτενής χρήση τεχνικών εξόρυξης πληροφορίας από το web ή από έγγραφα. Πολλοί ερευνητές ισχυρίζονται πως η εν λόγω μέθοδος έχει καλύτερα αποτελέσματα πλέον από ότι η μέθοδος που βασίζεται στην ηχητική ανάλυση διότι ο όγκος πληροφορίας που υπάρχει στο διαδίκτυο είναι εξαιρετικά μεγάλος για να καλυφθούν οι ανάγκες τις εξόρυξης.

Δυστυχώς όμως, και αυτή η μέθοδος επηρεάζεται από το γεγονός πως πάντα κάποιοι καλλιτέχνες/τραγούδια/κατηγορίες μουσικής είναι πιο δημοφιλή από άλλα και συνεπώς υπάρχει εκτενώς μεγαλύτερη πληροφορία προς εξόρυξη για αυτά που επισκιάζει τα υπόλοιπα.

4.2.6 Υβριδικά Μοντέλα

Τα υβριδικά μοντέλα έχουν ως σκοπό να συνδυάσουν δύο ή και περισσότερα από τα παραπάνω μοντέλα για να ενισχύσουν όσο το δυνατόν περισσότερο την αξιοπιστία / σχετικότητα των αποτελεσμάτων / προτάσεων. Ο Burke έχει προτείνει τρόπους με τους οποίους μπορούν να συνδυαστούν δύο μοντέλα, με ζύγιση/μέτρηση βάρους αποτελεσμάτων, μίξη, συνδυασμό χαρακτηριστικών, κλπ. Δεν υπάρχει αμφιβολία πως ένα υβριδικό μοντέλο θα έχει καλύτερα αποτελέσματα από ότι ένα απλό μοντέλο καθώς συνδυάζει τα πλεονεκτήματα όλων των μοντέλων που χρησιμοποιεί ενώ ταυτόχρονα περιορίζει έως και εκμηδενίζει τα μειονεκτήματα τους.

Η υπηρεσία Last.FM χρησιμοποιεί και τα 3 μοντέλα που αναλύθηκαν παραπάνω και καταλήγει στον χαρακτηρισμό καλλιτεχνών, δίσκων και τραγουδιών με λέξεις κλειδιά (tags) που προκύπτουν από συλλογικό φιλτράρισμα, μουσικούς "γείτονες", ανάλυση ηχητικής κυματομορφής, ανάλυση συναισθημάτων, κλπ. Η εφαρμογή που γράφτηκε για την εργασία αυτή χρησιμοποιεί τα δεδομένα αυτά για την κατασκευή μουσικών λιστών.

5 Last.FM

Το Last.FM είναι ένα κοινωνικό δίκτυο που ασχολείται αποκλειστικά με τη μουσική.

Κάθε χρήστης εγγράφεται στην υπηρεσία και εγκαθιστά στον υπολογιστή/συσκευή του ένα επιπρόσθετο πρόγραμμα (plug-in) ανάλογα με το πρόγραμμα αναπαραγωγής μουσικής που χρησιμοποιεί.

Το πρόγραμμα παρακολουθεί και καταγράφει τι μουσική ακούει ο χρήστης (καλλιτέχνης και τραγούδι) και πόσες φορές ακούει το κάθε τι. Επίσης λαμβάνει υπ' όψη την ώρα της μέρας ακόμα και την εποχή.



Άθως Αρούρης, 31, Male, Greece
www.twitter.com/snolly

[Edit »](#)

158513 plays since 19 Jan 2005

[1,676 Loved Tracks](#) | [4 Posts](#) | [2 Playlists](#) | [14 shouts](#)

Recently Listened Tracks

[Settings](#) | [Tweet](#) | [Share](#)



The Beta Band - Dry the Rain



11 May 2:33pm



Crazy P - Love On The Line

6 May 6:59pm



Tracy Chapman & Luciano Pavarotti - Baby, can I hold you tonight

6 May 6:53pm



Imany - You Will Never Know

6 May 6:49pm



Madleen Kane - Cherchez Pas



6 May 6:45pm

Έτσι μετά από κάποιο εύλογο χρονικό διάστημα καταγραφής μπορεί να εντοπίσει ποιοι χρήστες της υπηρεσίας έχουν κοινό μουσικό γούστο και κατά πόσο. Αυτούς τους ονομάζει "μουσικούς γείτονες" και έπειτα προτείνει σε κάποιον χρήστη να ακούσει μουσική που δεν έχει ακούσει ως τώρα αλλά ακούνε οι μουσικοί του γείτονες.

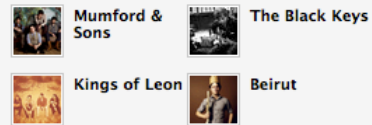
People on Last.fm with similar music taste to you.

1 2 Next ▾



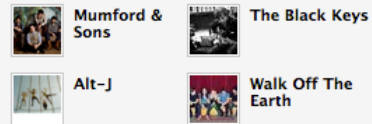
LindseyHugs
Lindsey, 25, United States
lindsey11387.tumblr.com
indie, folk, rock, electronic and british.
Last track: Father John Misty - Nancy From Now On
[Visit LindseyHugs's profile](#)

Artists you share



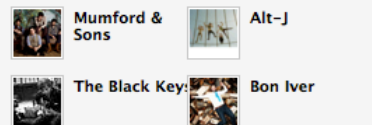
AndyArnay
Female, Portugal
folk, indie, rock, british and acoustic.
Last track: Mumford & Sons - I Will Wait
[Visit AndyArnay's profile](#)

Artists you share



bveerman
24, Male, Canada
indie, folk, alternative, electronic and rock.
Last track: The Avett Brothers - Tear Down the House
[Visit bveerman's profile](#)

Artists you share



Η υπηρεσία έγινε σύντομα δημοφιλής, έχει εγγεγραμμένους εκατομμύρια χρήστες και καταγράφει εκατοντάδες εκατομμύρια μουσικές προτιμήσεις.

Επιπλέον ο κάθε χρήστης έχει τη δυνατότητα να χαρακτηρίσει οποιοδήποτε τραγούδι ή καλλιτέχνη θέλει με λέξεις κλειδιά (tags) που θεωρεί πως είναι σχετικά. Αυτά τα tags δεν είναι απαραίτητο να χαρακτηρίζουν μόνο το

είδος μουσικής αλλά και περιοχές σύνθεσης, εποχές ακρόασης, συναισθήματα που προκύπτουν κατά την ακρόαση, ταινίες που ενδεχομένως έχει ακουστεί κάποιο τραγούδι, κοκ. Η υπηρεσία δεν περιορίζει πουθενά τον χρήστη στο πως θα χαρακτηρίσει τον καλλιτέχνη/τραγούδι.

Έχοντας συλλέξει εκατομμύρια tags για εκατοντάδες χιλιάδες τραγούδια (σχεδόν όλη την παγκόσμια μουσική δισκογραφία) από εκατομμύρια χρήστες (αμεροληψία στον χαρακτηρισμό λόγω πλήθους) η υπηρεσία παρέχει αυτή την πληροφορία πίσω στους χρήστες και στους προγραμματιστές μέσα από ένα API (application protocol interface) με web services.

API Methods

Artist

[Artist.addTags](#)
[Artist.getCorrection](#)
[Artist.getEvents](#)
[Artist.getInfo](#)
[Artist.getPastEvents](#)
[Artist.getPodcast](#)
[Artist.getShouts](#)
[Artist.getSimilar](#)
[Artist.getTags](#)
[Artist.getTopAlbums](#)
[Artist.getTopFans](#)
[Artist.getTopTags](#)
[Artist.getTopTracks](#)

Track

[Track.addTags](#)
[Track.ban](#)
[Track.getBuylinks](#)
[Track.getCorrection](#)
[Track.getFingerprintMetadata](#)
[Track.getInfo](#)
[Track.getShouts](#)
[Track.getSimilar](#)
[Track.getTags](#)
[Track.getTopFans](#)
[Track.getTopTags](#)
[Track.love](#)
[Track.removeTag](#)

Κάνοντας χρήση αυτών των web services η εφαρμογή που υλοποιήθηκε προσπαθεί να αναλύσει τη μουσική του χρήστη και να του προτείνει μουσικές λίστες έτοιμες προς ακρόαση.

Πανεπιστήμιο Πειραιώς

6 Πλατφόρμα Φορητών Συσκευών, Λειτουργικό Σύστημα

Τη στιγμή που γράφεται η εργασία αυτή υπάρχουν 3 δημοφιλή λειτουργικά συστήματα φορητών συσκευών. Αυτά είναι:

1. Apple iOS
2. Google Android
3. Microsoft Windows Mobile

Η Apple εισήγαγε το 2007 το πρώτο της "έξυπνο" τηλέφωνο και ένα χρόνο μετά έδωσε στους προγραμματιστές εργαλεία για να μπορούν να αναπτύξουν εφαρμογές για την πλατφόρμα αυτή και να τις διαθέτουν στο κοινό μέσα από ένα παγκόσμιο κατάστημα (είτε επί πληρωμή, είτε δωρεάν).

Αργότερα ακολούθησε στο ίδιο ακριβώς μοντέλο και η Google η οποία όμως διαθέτει το λειτουργικό της σε 3ους κατασκευαστές συσκευών για να το παραμετροποιήσουν όπως θέλουν.

Τέλος πρόσφατα μπήκε δυναμικά στο χώρο και η Microsoft προσπαθώντας να διεκδίκησει ένα μερίδιο της αγοράς και εκείνη.

Για την εργασία αυτή επιλέχθηκε το iOS της Apple ως πλατφόρμα ανάπτυξης μιας και είναι η πιο δημοφιλής πλατφόρμα στη χώρα μας, στην Ευρώπη και στην Αμερική.

6.1 iOS

Το iOS βρίσκεται στην έκδοση 6.1 (Μάιος 2013) και επιτρέπει την ανάπτυξη εφαρμογών από την έκδοση 2. Προσφέρει πληθώρα δυνατοτήτων και κάνει χρήση τεχνολογιών όπως:

- οθόνη αφής που αντιλαμβάνεται πολλαπλές χειρονομίες (ζουμ με 2 δάχτυλα, κλπ.)
- μικρόφωνο
- ηχείο
- ψηφιακή πυξίδα
- GPS (Global Positioning System)
- όργανο μέτρησης επιτάχυνσης - accelerometer (όργανο που καταλαβαίνει τη φορά της συσκευής, την περιστροφή της, κλπ.)
- συνδεσιμότητα σε δίκτυα Wi-Fi αλλά και κινητής τηλεφωνίας (3G, LTE)
- web browser με πλήρη υποστήριξη HTML-5
- αναγνώριση ομιλίας
- περιήγηση σε χάρτη
- αναπαραγωγή εικόνας και ήχου
- κάμερα
- ασύρματη μετάδοση εικόνας και ήχου

Το περιβάλλον χρήσης του iOS είναι βασισμένο στην ιδέα του άμεσου χειρισμού μέσω χειρονομιών αφής. Τα στοιχεία του γραφικού περιβάλλοντος

περιέχουν συρόμενα κουμπιά (sliders), διακόπτες (switches), απλά κουμπιά (buttons), πεδία κειμένου (text fields), κλπ. Οι χειρονομίες είναι τραβήγματα (swipe), τσιμπήματα (pinching) που μεταφράζονται σε διαγραφές, ζουμ σε περιοχές και λοιπές λειτουργίες.



Αρχιτεκτονικά το iOS αποτελείται από 4 επιμέρους στρώματα:

1. Core OS layer (αναλαμβάνει τις βασικές λειτουργίες της συσκευής)
2. Core Services layer (αναλαμβάνει όλες τις υπηρεσίες που παρέχει η συσκευή στους προγραμματιστές)
3. Media layer (αναλαμβάνει όλες τις λειτουργίες πολυμέσων της συσκευής)
4. Cocoa Touch layer (αναλαμβάνει την απεικόνιση του γραφικού περιβάλλοντος στο χρήστη και όλες τις ενέργειες του χρήστη).

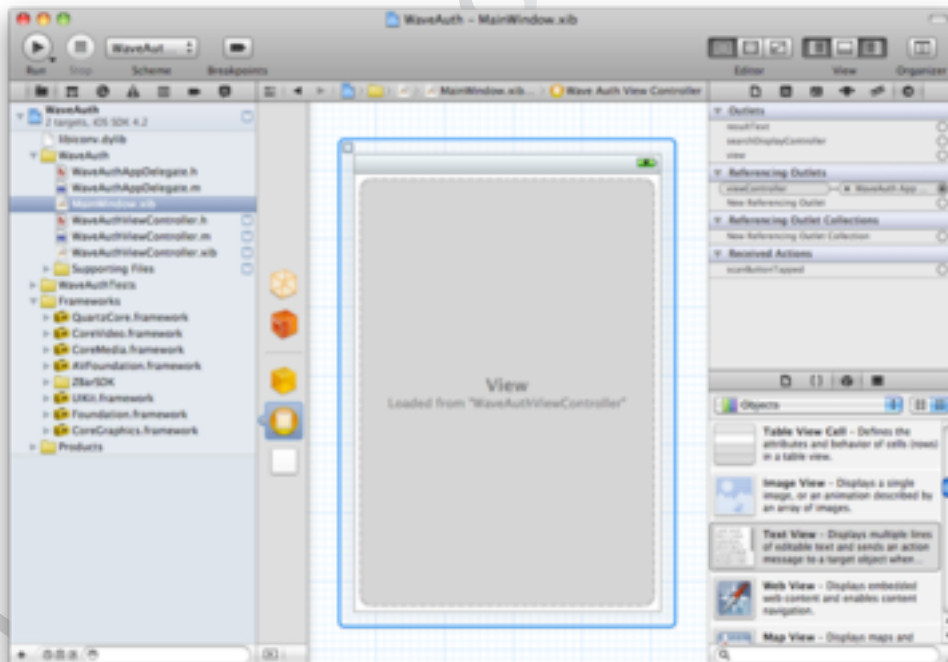


6.2 XCode & Objective-C

Η Objective-C είναι μια υψηλού επιπέδου, αντικειμενοστραφής γλώσσα προγραμματισμού γενικής χρήσης η οποία προσθέτει δυνατότητα χρήσης μηνυμάτων τύπου Smalltalk στη γλώσσα προγραμματισμού C.

Είναι η βασική γλώσσα προγραμματισμού που χρησιμοποιείται από την Apple στο Mac OS X και στο iOS.

Οι εφαρμογές που αναπτύσσονται για το iOS γράφονται αποκλειστικά σε Objective-C με τη χρήση του εργαλείου XCode.



Το XCode είναι ένα ενοποιημένο περιβάλλον ανάπτυξης - integrated development environment (IDE) που περιέχει μια σουίτα από εργαλεία ανάπτυξης λογισμικού. Η πρώτη του έκδοση κατασκευάστηκε το 2003 και τώρα βρίσκεται στην έκδοση 4.6.

Στο XCode μπορεί να σχεδιαστεί το γραφικό περιβάλλον της εφαρμογής, οι βάσεις δεδομένων που ενδέχεται να χρησιμοποιηθούν (Core Data) καθώς και να γραφτεί ο κώδικας που αναλαμβάνει όλη τη λειτουργικότητα. Ουσιαστικά είναι το μοναδικό εργαλείο που χρειάζεται για την ανάπτυξη μιας εφαρμογής για iOS.

7 Frameworks & Libraries

Παρακάτω παρουσιάζονται όλες οι βιβλιοθήκες (libraries) και τα πλαίσια (frameworks) που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής.

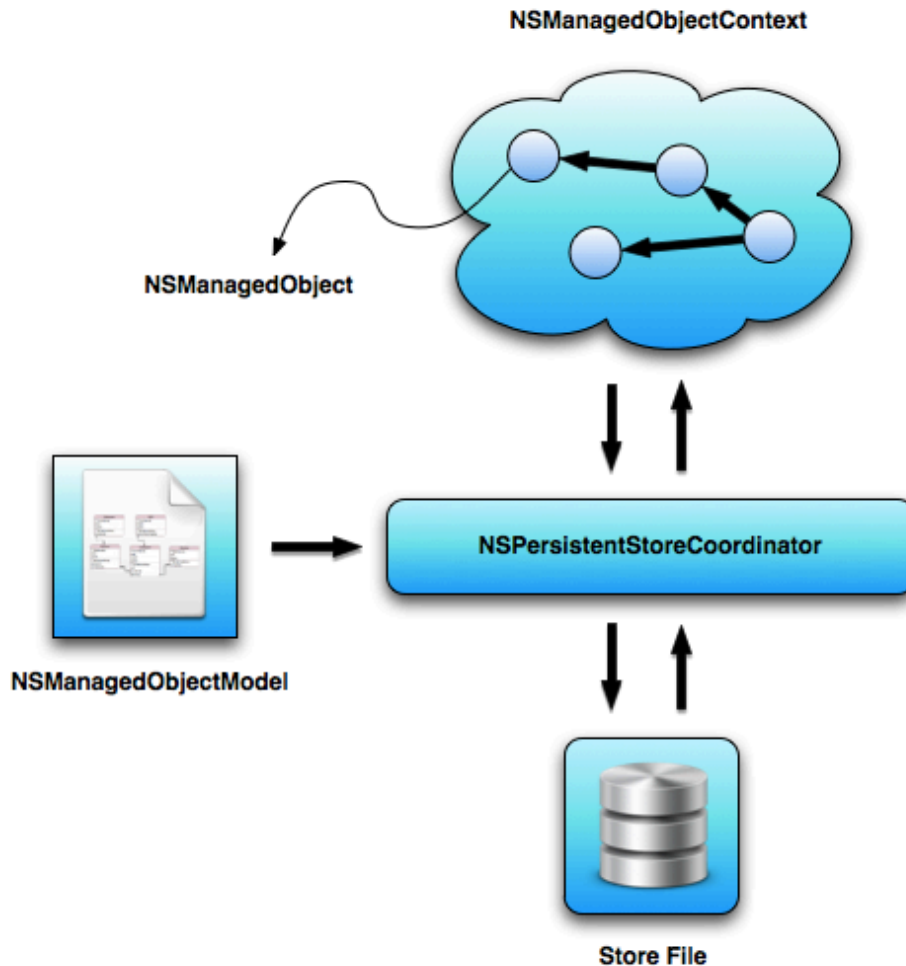
7.1 Core Data & SQLite



Το Core Data είναι ένα πλαίσιο διαχείρισης δεδομένων σε σχεσιακό μοντέλο οντοτήτων-ιδιοτήτων.

Αφού δομηθούν οι πίνακες και οριστούν οι σχέσεις μεταξύ των ιδιοτήτων, το Core Data αναλαμβάνει να δημιουργήσει Objective-C αντικείμενα που περιγράφουν τις οντότητες και τις ιδιότητες αυτές. Ο προγραμματιστής διαχειρίζεται τα δεδομένα αυτά χρησιμοποιώντας τα αντικείμενα χωρίς να χρειάζεται να συντάξει εντολές SQL.

Τα δεδομένα μπορούν να οργανωθούν σε XML, binary ή σε SQLite βάσεις και είναι serialized (μπορούν να μεταδοθούν μέσω δικτύων) ανά πάσα στιγμή.



Τα βήματα είναι τα εξής:

1. Σχεδιάζεται το μοντέλο στο XCode που ορίζει τις οντότητες (entities), τις ιδιότητες τους (attributes) και τις σχέσεις μεταξύ αυτών (relationships).
2. Παράγονται από το (XCode) οι κλάσεις που περιγράφουν τα Objective-C αντικείμενα (NSManagedObject).
3. Γίνεται ή όποια ενέργεια, ερώτηση, τροποποίηση, δημιουργία ή διαγραφή μέσω των αντικειμένων.
4. Ο NSPersistentStoreCoordinator αναλαμβάνει να αποθηκεύσει τα δεδομένα από τη μνήμη (NSManagedObjects) στο αρχείο της βάσης (store file).

7.2 Grand Central Dispatch



Το Grand Central Dispatch είναι ένα framework της Objective-C που κάνει δυνατή τη δημιουργία νημάτων (threads) σε μια εφαρμογή και τη διαχείριση τους.

Παρέχει τις εξής "ουρές" αλλά και τη δυνατότητα κατασκευής νέων.

- DISPATCH_QUEUE_PRIORITY_DEFAULT
- DISPATCH_QUEUE_PRIORITY_HIGH
- DISPATCH_QUEUE_PRIORITY_LOW
- DISPATCH_QUEUE_PRIORITY_BACKGROUND

Όταν υλοποιείται μια εφαρμογή αρχικά έχει μόνο ένα νήμα το οποίο ονομάζουμε MAIN_THREAD. Ένα νήμα σημαίνει πως οποιαδήποτε πράξη, λειτουργία, βήμα αλγορίθμου, κλπ εκτελείται σειριακά μέσα σε αυτό το νήμα. Το MAIN_THREAD είναι και το νήμα που εξ' ορισμού ελέγχει και το γραφικό περιβάλλον της εφαρμογής (User Interface), δηλ το πάτημα ενός κουμπιού, την απεικόνιση πληροφορίας στην οθόνη, κοκ.

Όταν μια εφαρμογή είναι αρκετά πολύπλοκη και έχει λειτουργίες οι οποίες για να εκτελεστούν χρειάζεται χρόνος ή υπάρχουν παραπάνω από μία

λειτουργίες που πρέπει να εκτελούνται **ταυτόχρονα** τότε γίνεται αναγκαία η χρήση πολλαπλών νημάτων.

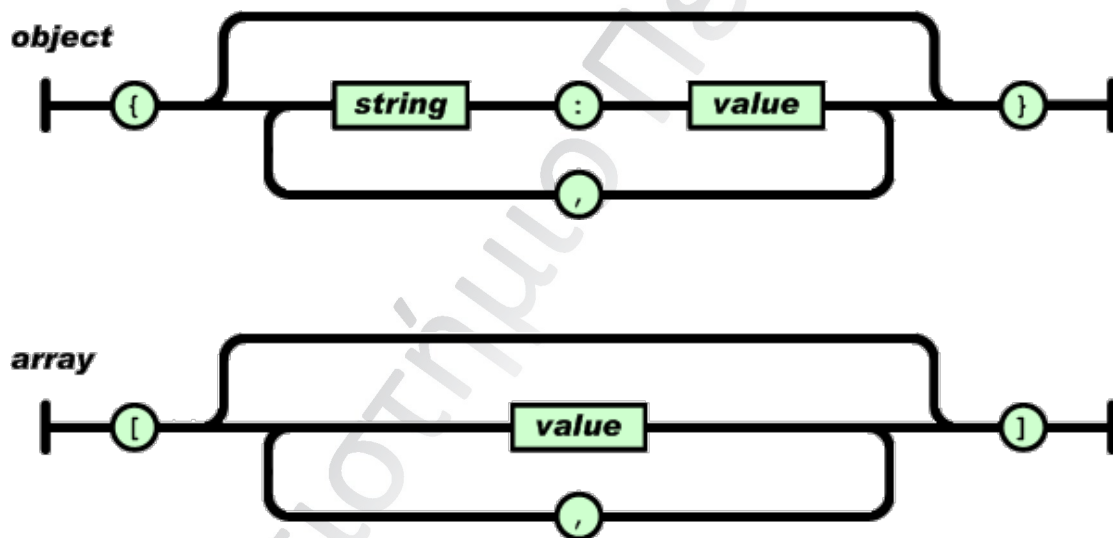
Στη συγκεκριμένη περίπτωση η λειτουργία ανάλυσης της μουσικής του χρήστη διαρκεί περίπου 1 δευτερόλεπτο ανά τραγούδι πράγμα που σημαίνει πως για να εξεταστεί μια μουσική βιβλιοθήκη 1500 τραγουδιών απαιτείται περίπου μισή ώρα. Μισή ώρα που αν χρησιμοποιηθεί ένα μόνο νήμα το γραφικό περιβάλλον της εφαρμογής δε θα αποκρίνεται.

Είναι αναγκαίο και απαραίτητο να τρέξει η διεργασία αυτή σε άλλο νήμα για να ανταποκρίνεται ταυτόχρονα το γραφικό περιβάλλον και να μπορεί να εκτελεστεί κι άλλη διεργασία.

Κάθε νήμα μπαίνει σε μια από τις παραπάνω ουρές (ή σε άλλη νέα) και κάθε νήμα στην ίδια ουρά εκτελείται σειριακά. Οι προεπιλεγμένες ουρές στην Objective-C έχουν ξεχωριστές προτεραιότητες (σε επεξεργαστικούς πόρους) και η ουρά BACKGROUND μπορεί να "παγώνει" όσο η εφαρμογή δεν είναι ενεργή και να συνεχίσει με το που ενεργοποιηθεί η εφαρμογή.

7.3 JSON & ASIHTTPRequest

Το JSON (JavaScript Object Notation) είναι ένα φορμά ανταλλαγής δεδομένων το οποίο είναι ελαφρύ (ως προς τη χρήση πόρων) και είναι εύκολα δημιουργήσιμο και προσπελάσιμο από υπολογιστές. Είναι βασισμένο σε ένα μέρος της JavaScript, αποτελείται αποκλειστικά από χαρακτήρες κειμένου αλλά χρησιμοποιεί σύμβολα που είναι ευρέως γνωστά στους προγραμματιστές που γνωρίζουν γλώσσες C, C++, C#, Java, JavaScript, Perl και Python.



Το Last.FM διαθέτει τα δεδομένα του σε μορφή JSON και το παρακάτω είναι ένα παράδειγμα του αιτήματος για τα top tags ενός τραγουδιού και η απάντηση της υπηρεσίας σε μορφή JSON.

Αίτημα:

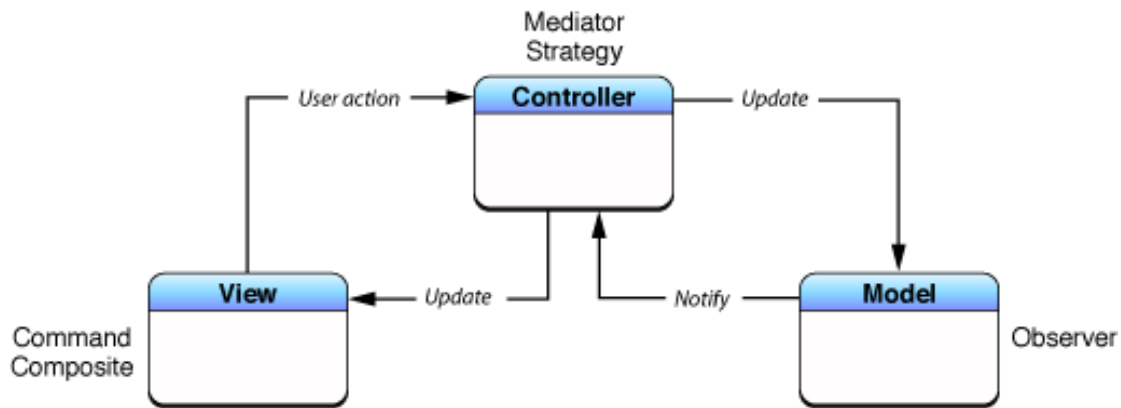
[http://ws.audioscrobbler.com/2.0/?method=track.gettoptags&artist=radiohead
&track=paranoid+android&api_key=a4906f397ed277f149a02d67034f2b60&for
mat=json](http://ws.audioscrobbler.com/2.0/?method=track.gettoptags&artist=radiohead&track=paranoid+android&api_key=a4906f397ed277f149a02d67034f2b60&format=json)

Απάντηση:



Το ASIHTTPRequest είναι μια βιβλιοθήκη της Objective-C με την οποία είναι δυνατή η κατασκευή αιτημάτων JSON η αποστολή τους στο web service που τα παρέχει η μεταφορά της απάντησης και η ανάγνωση του μηνύματος JSON.

8 Αρχιτεκτονική MVC



Οι εφαρμογές που υλοποιούνται για iOS ακολουθούν κατεξοχήν την αρχιτεκτονική MVC : Model View Controller.

Η αρχιτεκτονική MVC κατηγοριοποιεί τα αντικείμενα (objects) της εφαρμογής ανάλογα με τον γενικό ρόλο τους μέσα στην εφαρμογή. Θεωρείται ένα σύνθετο πρότυπο (pattern) δεδομένου ότι περιλαμβάνει αρκετά περισσότερα απλούστερα πρότυπα.

Τα αντικειμενοστραφή προγράμματα επωφελούνται με πολλούς τρόπους όταν υιοθετήσουν την αρχιτεκτονική MVC. Εξασφαλίζεται πως πολλά αντικείμενα θα μπορούν να επαναχρησιμοποιηθούν (reusability) και τα interfaces των αντικειμένων είναι καλύτερα ορισμένα. Τέλος τα προγράμματα είναι πιο προσαρμόσιμα αν τυχόν χρειαστούν αλλαγές.

8.1 Ρόλοι & Σχέσεις των Αντικειμένων MVC

Στην αρχιτεκτονική MVC θεωρούμε πως υπάρχουν τριών ειδών αντικείμενα:

1. αντικείμενα μοντέλου (model)
2. αντικείμενα προβολής (view)
3. αντικείμενα ελέγχου (controller)

Η αρχιτεκτονική ορίζει τους ρόλους που παίζουν τα αντικείμενα μέσα στην εφαρμογή και τον τρόπο με τον οποίο επικοινωνούν μεταξύ τους. Όταν σχεδιάζεται μια εφαρμογή, ένα πολύ σημαντικό βήμα είναι η επιλογή ή η δημιουργία των κλάσεων για τα αντικείμενα που εμπίπτουν σε ένα από τους παραπάνω τύπους. Κάθε τύπος αντικειμένων είναι χωρισμένος από τους υπόλοιπους μέσω ορίων/περιορισμών.

8.1.1 Τα Αντικείμενα Μοντέλου περιλαμβάνουν Δεδομένα & Βασικές

Συμπεριφορές

Τα αντικείμενα μοντέλου αντιπροσωπεύουν ειδική γνώση και εξειδίκευση. Κρατάνε τα δεδομένα της εφαρμογής και ορίζουν τη λογική που διαχειρίζεται και τροποποιεί τα δεδομένα. Μια καλά σχεδιασμένη εφαρμογή έχει **όλα** τα σημαντικά της δεδομένα ορισμένα μέσα σε αντικείμενα μοντέλου. Οποιοδήποτε δεδομένο αποτελεί μέρος της μόνιμα αποθηκευμένης κατάστασης της εφαρμογής (είτε αυτή είναι σε βάση δεδομένων ή σε απλό αρχείο) θα πρέπει να περιέχεται μέσα σε αντικείμενα μοντέλου όταν η εφαρμογή είναι ενεργή και τα δεδομένα φορτωμένα στη μνήμη αυτής. Επειδή αντιπροσωπεύουν γνώση και εξειδίκευση σχετικά με ένα συγκεκριμένο πρόβλημα/περίπτωση/εφαρμογή τείνουν να είναι επαναχρησιμοποιήσιμα.

Ιδανικά, ένα αντικείμενο μοντέλου δεν έχει καμιά άμεση σύνδεση με το γραφικό περιβάλλον το οποίο το παρουσιάζει ή/και το τροποποιεί. Για παράδειγμα, αν έχουμε ένα αντικείμενο μοντέλου που αντιπροσωπεύει ένα tag για ένα τραγούδι και θέλουμε να αποθηκεύσουμε και τι δύναμη έχει αυτό το tag (πόσοι χρήστες το χρησιμοποίησαν για να χαρακτηρίσουν με αυτό ένα τραγούδι) καλό είναι την τιμή αυτή να την αποθηκεύσουμε μέσα στο object tag model. Το πως θα απεικονίζεται αυτή η πληροφορία δεν ανήκει στο αντικείμενο μοντέλου.

8.1.2 Τα Αντικείμενα Προβολής παρουσιάζουν στον χρήστη Πληροφορία

Ένα αντικείμενο προβολής γνωρίζει πως να εμφανίσει, και ενδεχομένως να επιτρέψει στο χρήστη τροποποίηση των δεδομένων του μοντέλου της εφαρμογής. Το αντικείμενο προβολής δεν πρέπει να είναι υπεύθυνο για την αποθήκευση των δεδομένων που παρουσιάζει. Αντίθετα είναι υπεύθυνο να παρουσιάσει μέρος ενός αντικειμένου μοντέλου, ολόκληρο το αντικείμενο ή ακόμα και πολλά αντικείμενα μαζί.



Τα αντικείμενα προβολής είναι συχνά παραμετροποίηση και επαναχρησιμοποιήσιμα και παρέχουν ομοιομορφία μεταξύ των εφαρμογών. Για παράδειγμα το αντικείμενο NSTableView (πάνω) παρουσιάζει δεδομένα σε μορφή πίνακα και το ίδιο αντικείμενο χρησιμοποιείται από εκατοντάδες χιλιάδες εφαρμογές και φαίνεται σε όλες το **ίδιο**.

Το αντικείμενο προβολής πρέπει να εξασφαλίζει πως απεικονίζει σωστά το μοντέλο. Συνεπώς χρειάζεται να παραμένει ενημερωμένο σχετικά με τις αλλαγές που γίνονται στο μοντέλο. Και επειδή τα αντικείμενα μοντέλου δεν πρέπει να είναι "δεμένα" με αντικείμενα προβολής χρειάζονται κάποιον τρόπο ώστε να επικοινωνούν τις τυχόν αλλαγές που υπόκειντο στα δεδομένα που κρατάνε.

Πανεπιστήμιο Πειραιώς

8.1.3 Τα Αντικείμενα Ελέγχου "δένου" το Μοντέλο με την Προβολή

Τα αντικείμενα ελέγχου είναι οι μεσάζοντες μεταξύ των αντικειμένων προβολής και των αντικειμένων μοντέλου. Οι ελεγκτές είναι υπεύθυνοι να εξασφαλίζουν πως οι προβολές έχουν πρόσβαση στο μοντέλο και να ενημερώνουν τις προβολές όταν γίνονται αλλαγές στο μοντέλο. Τα αντικείμενα ελέγχου επίσης εκτελούν ρυθμιστικές και συντονιστικές ενέργειες για μια εφαρμογή και διαχειρίζονται τον κύκλο ζωής των άλλων αντικειμένων.

Σε ένα τυπικό MVC σχεδιασμό, όταν ο χρήστης εισάγει μια τιμή ή δηλώνει μια επιλογή μέσω ενός αντικειμένου προβολής, η τιμή της επιλογής επικοινωνείται σε ένα αντικείμενο ελέγχου. Το αντικείμενο ελέγχου αντιλαμβάνεται την επιλογή του χρήστη (πάντα ανάλογα με τη λογική της εφαρμογής) και έπειτα μπορεί να "πει" σε κάποιο αντικείμενο μοντέλου να κάνει κάτι (πχ να προσθέσει μια τιμή ή να σβήσει μια εγγραφή) ή μπορεί να δώσει εντολή στο αντικείμενο μοντέλου να απεικονίσει μια αλλαγμένη τιμή μιας ιδιότητας του. Βασισμένο στην ίδια εντολή χρήστη κάποια αντικείμενα ελέγχου μπορεί να "πουν" σε αντικείμενα προβολής να αλλάξουν ένα μέρος της απεικόνισης τους ή της συμπεριφοράς τους (πχ απενεργοποίησε το τάδε κουμπί).

Συνεπώς, όταν ένα αντικείμενο μοντέλου αλλάζει, το αντικείμενο αυτό επικοινωνεί την αλλαγή σε ένα αντικείμενο ελέγχου το οποίο με τη σειρά του ζητά από ένα ή περισσότερα αντικείμενα προβολής να ενημερώσουν τους εαυτούς τους.

Τα αντικείμενα ελέγχου μπορεί να είναι επαναχρησιμοποιήσιμα ή μη επαναχρησιμοποιήσιμα, ανάλογα με τον τύπο τους.

Στην συγκεκριμένη εφαρμογή έχει χρησιμοποιηθεί αυστηρά η αρχιτεκτονική MVC.

Πανεπιστήμιο Πειραιώς

9 Η Εφαρμογή

Η εφαρμογή αποτελείται από τρία πολύ διακριτά μέρη. Το μέρος της ανάλυσης της μουσικής του χρήστη, το μέρος της παρουσίασης των καλλιτεχνών της μουσικής του συλλογής και τη δημιουργία των μουσικών λιστών βασισμένες στον καλλιτέχνη της επιλογής του χρήστη. Παρακάτω γίνεται αναλυτική παρουσίαση των τριών αυτών μερών.

9.1 Ανάλυση Μουσικής

Η εφαρμογή χρησιμοποιεί το MediaPlayer library του iOS και ζητά από το λειτουργικό να δώσει σε ένα αντικείμενο `NSArray` όλα τα τραγούδια που έχει ο χρήστης αποθηκευμένα στη συσκευή του. Το κάθε τραγούδι είναι ένα αντικείμενο τύπου `MPMediaItemCollection` και περιέχει διάφορες ιδιότητες όπως `MPMediaItemPropertyPersistentID`, `MPMediaItemPropertyArtist` και `MPMediaItemPropertyTitle` που περιέχουν τον μοναδικό αριθμό κλειδί που αντιστοιχεί (το iOS) σε κάθε τραγούδι, τον Καλλιτέχνη και τον τίτλο του τραγουδιού.

Η εφαρμογή κατασκευάζει ένα νέο νήμα το τοποθετεί στην ουρά `DISPATCH_QUEUE_PRIORITY_BACKGROUND` και το εκκινεί. Αυτό σημαίνει πως αυτόματα απελευθερώνεται το βασικό νήμα και καθίσταται το γραφικό περιβάλλον χρηστικό και αποκρινόμενο.

Ξεκινά ο βρόχος ανάλυσης των τραγουδιών και για κάθε τραγούδι δημιουργείται ένα `ASIFormDataRequest` και αυτό αποστέλλεται στα web services του Last.FM.

```
NSString *strURL =  
@"http://ws.audioscrobbler.com/2.0/?method=track.gettoptags&fo  
rmat=json";  
NSURL *url = [[NSURL alloc] initWithString:strURL];  
ASIFormDataRequest *request = [ASIFormDataRequest  
requestWithURL:url];  
[request addPostValue:artistName forKey:@"artist"];  
[request addPostValue:trackName forKey:@"track"];  
[request addPostValue:@"1" forKey:@"autocorrect"];  
[request  
addPostValue:@"a4906f397ed277f149a02d67034f2b60"  
forKey:@"api_key"];  
[request setDelegate:self];  
[request setTimeoutSeconds:7.0];  
NSLog(@"About to start JSON request");  
[request startSynchronous];
```

Το Last.FM με τη σειρά του επιστρέφει τα top tags που χαρακτηρίζουν το εν λόγω τραγούδι μαζί με τα σκορ τους (πόσο % των χρηστών έχει χαρακτηρίσει με αυτό το tag αυτό το τραγούδι) σε μορφή JSON.

Στη συνέχεια το JSON αρχείο "διαβάζεται" και για κάθε tag μαζί με το σκορ του και το `MPMediaItemPropertyPersistentID` του κάθε τραγουδιού αποθηκεύονται σε ένα αντικείμενο μοντέλου τύπου `TrackTags`. Αυτό με τη σειρά του αποθηκεύεται στη βάση δεδομένων της εφαρμογής.

The screenshot shows the Xcode Entity Inspector. On the left, under 'ENTITIES', the 'TrackTags' entity is selected. On the right, the 'Attributes' table is visible, listing the following attributes:

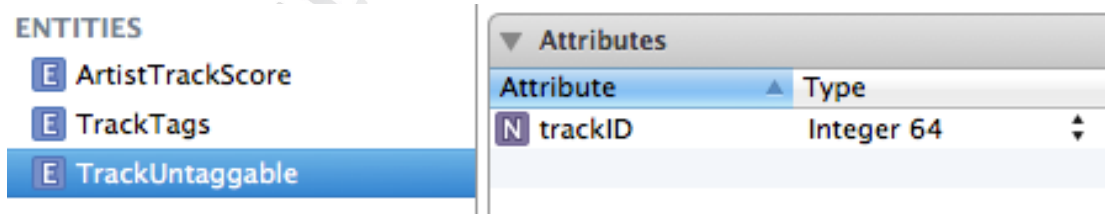
Attribute	Type
score	Integer 16
tagName	String
trackID	Integer 64

Με τον τρόπο αυτό η εφαρμογή γνωρίζει εκατοντάδες λέξεις κλειδιά (tags) μαζί με τη βαρύτητα τους (score) για κάθε τραγούδι που έχει ο χρήστης.

Στην περίπτωση που χαθεί η συνδεσιμότητα με το internet για κάποιο χρόνο (η συσκευή πέρασε από σύνδεση 3G σε σύνδεση Wi-Fi) και δεν έχουν επιστραφεί δεδομένα JSON σε 7 δευτερόλεπτα από τη στιγμή του αιτήματος η εφαρμογή προχωράει στον έλεγχο του επόμενου τραγουδιού.

Η ανάλυση μπορεί να τρέξει πολλές φορές κυρίως σε περίπτωση που ο χρήστης προσθέσει τραγούδια στη συσκευή του. Τότε συμβαίνουν τα εξής:

- η εφαρμογή **δεν** θα ξαναψάξει για tags για τραγούδια που ήδη γνωρίζει
- η εφαρμογή **δεν** θα ξαναψάξει για tags για τραγούδια που έχει επιχειρήσει στο παρελθόν να βρει tags αλλά δεν τα κατάφερε (περίπτωση άγνωστου τραγουδιού ή λανθασμένα ονομασμένου τραγουδιού)



- η εφαρμογή θα ξαναψάξει για tags που λόγω κακής συνδεσιμότητας δεν κατάφερε να βρει στο παρελθόν
- η εφαρμογή θα ψάξει για tags για κάθε νέο τραγούδι που εντοπίζει
- η εφαρμογή θα ψάξει για tags για κάθε τραγούδι που έχει τροποποιηθεί ο καλλιτέχνης ή ο τίτλος του

Κατά τη διάρκεια της ανάλυσης ο χρήστης μπορεί να παρακολουθεί την πρόοδο της διαδικασίας ανάλυσης, το τραγούδι που εξετάζεται εκείνη τη στιγμή, τα tags που βρέθηκαν και πόσα tags είναι ήδη αποθηκευμένα στη βάση δεδομένων.

Τα δεδομένα (tags) που συλλέγονται που συλλέγονται μπορεί να έχουν σχέση με τα είδη μουσικής του τραγουδιού, τον καλλιτέχνη, την χώρα προέλευσης, τη μουσική

σχολή, τη δεκαετία που γράφτηκε, συναισθήματα χρηστών όταν το ακούν, αναμνήσεις του, κλπ. Τα tags εμφανίζονται με σειρά σκορ από το πιο σχετικό / δημοφιλές μέχρι το λιγότερο σχετικό. Τα tags με μηδενικό σκορ δεν αποθηκεύονται ποτέ στη βάση δεδομένων.

Analyse My Music

Calypso
Suzanne Vega

female vocalists
folk
pop
rock
suzanne vega
alternative folk
singer-songwriter
mid
contemporary folk
college rock
80s
classic rock
calypso
female vocal
test...

99592 tags in DB

Music Settings

Απόσπασμα από το Console Log κατά την ανάλυση της μουσικής βιβλιοθήκης

επιτυχής εύρεση tags

```
-----Next Song-----  
0 number of tags in DB for track 682432575  
Getting tags for: Madness - Cardiac Arrest  
About to start JSON request  
About to Parse results  
Parsing results  
Found 56 tags  
Results parsed  
Returned with tags about to write them to DB.  
Tags saved in DB.
```

μη επιτυχής εύρεση tags

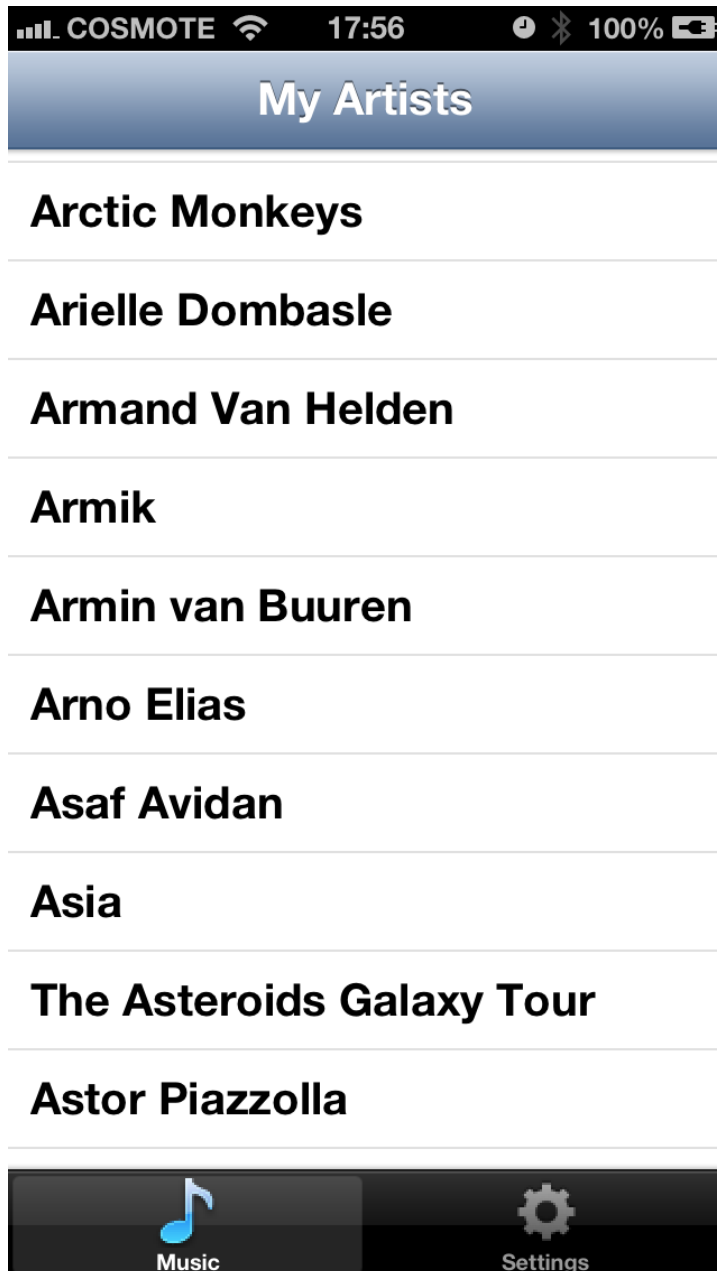
```
-----Next Song-----  
0 number of tags in DB for track 421637690  
Getting tags for: Marisa Sanni - Canzone Per Te  
About to start JSON request  
About to Parse results  
Parsing results  
Found 0 tags  
Results parsed  
Returned with tags about to write them to DB.  
No tags to write to DB. Marking track untaggable.
```

τραγούδι που έχει ήδη αναλυθεί

```
-----Next Song-----  
100 number of tags in DB for track 536035745  
Tags Exist in DB for the Beatles - Across the Universe  
Skipping track analysis
```

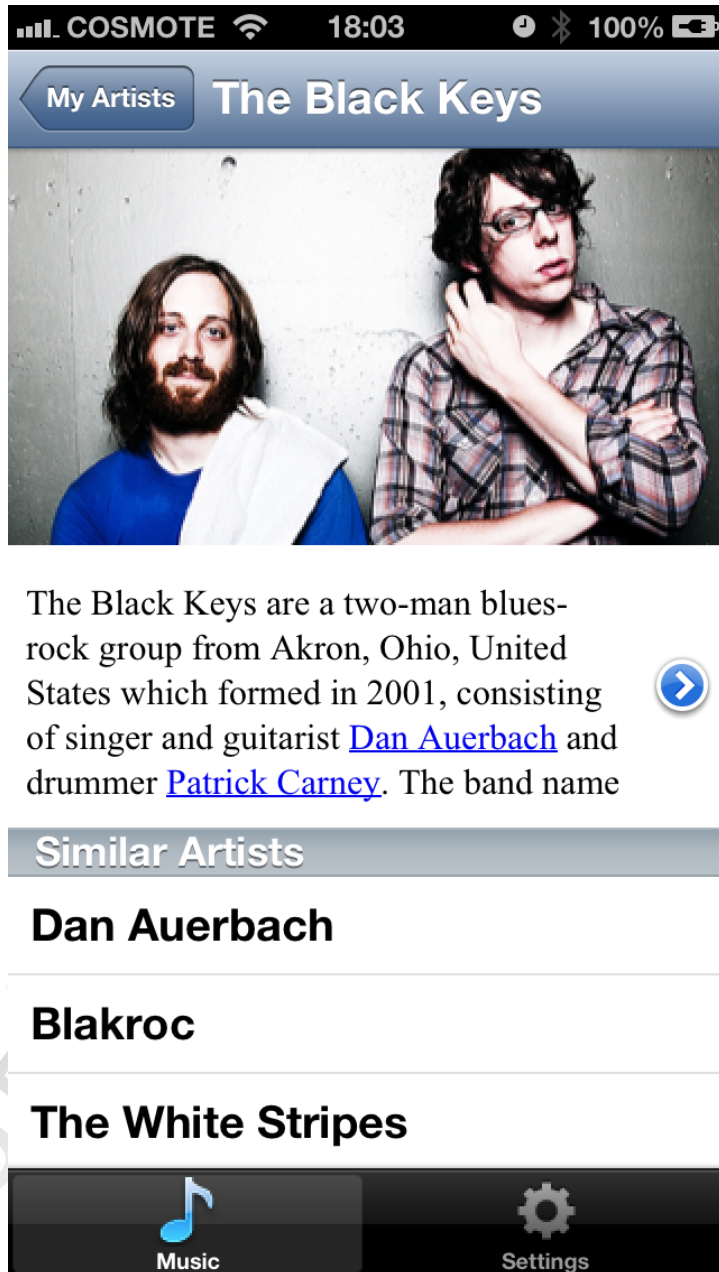
9.2 Παρουσίαση Καλλιτεχνών & των Προφίλ τους

Αρχικά παρουσιάζεται στο χρήστη η πρώτη βασική οθόνη με την λίστα **όλων** των καλλιτεχνών που έστω και ένα τους τραγούδι βρίσκεται **ήδη** αποθηκευμένο



στη μουσική βιβλιοθήκη των συσκευών τους.

Ο χρήστης καλείται να επιλέξει όποιον καλλιτέχνη επιθυμεί. Μόλις συμβεί αυτό δημιουργείται ακόμα ένα `ASIFormDataRequest` αίτημα προς το Last.FM ώστε να επιστραφεί στην εφαρμογή σε JSON μορμά σύνδεσμος για φωτογραφία του καλλιτέχνη, μια μίνι βιογραφία του και άλλοι καλλιτέχνες σχετικοί με τον επιλεγμένο καλλιτέχνη.



```
NSString *strURL =  
@"http://ws.audioscrobbler.com/2.0?method=artist.getinfo&forma  
t=json";  
NSURL *url = [[NSURL alloc] initWithString:strURL];  
ASIFormDataRequest *request = [ASIFormDataRequest  
requestWithURL:url];  
[request addPostValue:artistName forKey:@"artist"];  
[request  
addPostValue:@"a4906f397ed277f149a02d67034f2b60"  
forKey:@"api_key"];  
[request setDelegate:self];  
[request setTimeoutSeconds:7.0];  
[request startAsynchronous];
```

Ο χρήστης μπορεί να επιλέξει έναν από τους σχετικούς καλλιτέχνες πράγμα που θα τον οδηγήσει στο προφίλ τους. Η περιήγηση από προφίλ σε προφίλ και πίσω στη λίστα των καλλιτεχνών γίνεται με το Navigation Bar στο πάνω μέρος της οθόνης.

Στη μίνι βιογραφία ο χρήστης μπορεί να επιλέξει όποιο σύνδεσμο θέλει και θα περιηγηθεί εκτός εφαρμογής στη σελίδα του Last.FM από το Safari τον web browser του iOS για να διαβάσει αναλυτικότερες πληροφορίες για τον καλλιτέχνη που επέλεξε.

Πανεπιστήμιο Πειραιώς

9.3 Δημιουργία Μουσικών Λιστών

Όντας στο προφίλ ενός καλλιτέχνη ο χρήστης μπορεί να επιλέξει την αυτόματη δημιουργία μουσικής λίστας που λαμβάνει υπόψη της τα χαρακτηριστικά του καλλιτέχνη πατώντας το κουμπί



Η εφαρμογή θα δημιουργήσει ένα νέο νήμα στην ουρά DISPATCH_QUEUE_PRIORITY_HIGH για να ξεκινήσει ο αλγόριθμος δημιουργίας της μουσικής λίστας.

Αρχικά θα δημιουργηθεί ένα νέο αίτημα `ASIDataRequest` για να αποσταλεί στο web service του Last.FM.

```
NSString *strURL =  
@"http://ws.audioscrobbler.com/2.0/?method=artist.gettoptags&f  
ormat=json";  
NSURL *url = [[NSURL alloc] initWithString:strURL];  
ASIDataRequest *request = [ASIDataRequest  
requestWithURL:url];  
[request addPostValue:artistName forKey:@"artist"];  
[request addPostValue:@"1" forKey:@"autocorrect"];  
[request  
addPostValue:@"a4906f397ed277f149a02d67034f2b60"  
forKey:@"api_key"];  
[request setDelegate:self];  
[request setTimeoutSeconds:7.0];  
[request startAsynchronous];
```

Μόλις επιστραφούν τα δεδομένα JSON που περιέχουν tags και scores των tags για τον επιλεγμένο καλλιτέχνη η εφαρμογή θα ξεκινήσει την σύγκριση εκατοντάδων χιλιάδων tags με τον εξής τρόπο:

9.3.1 Αλγόριθμος Υπολογισμού Σχετικότητας

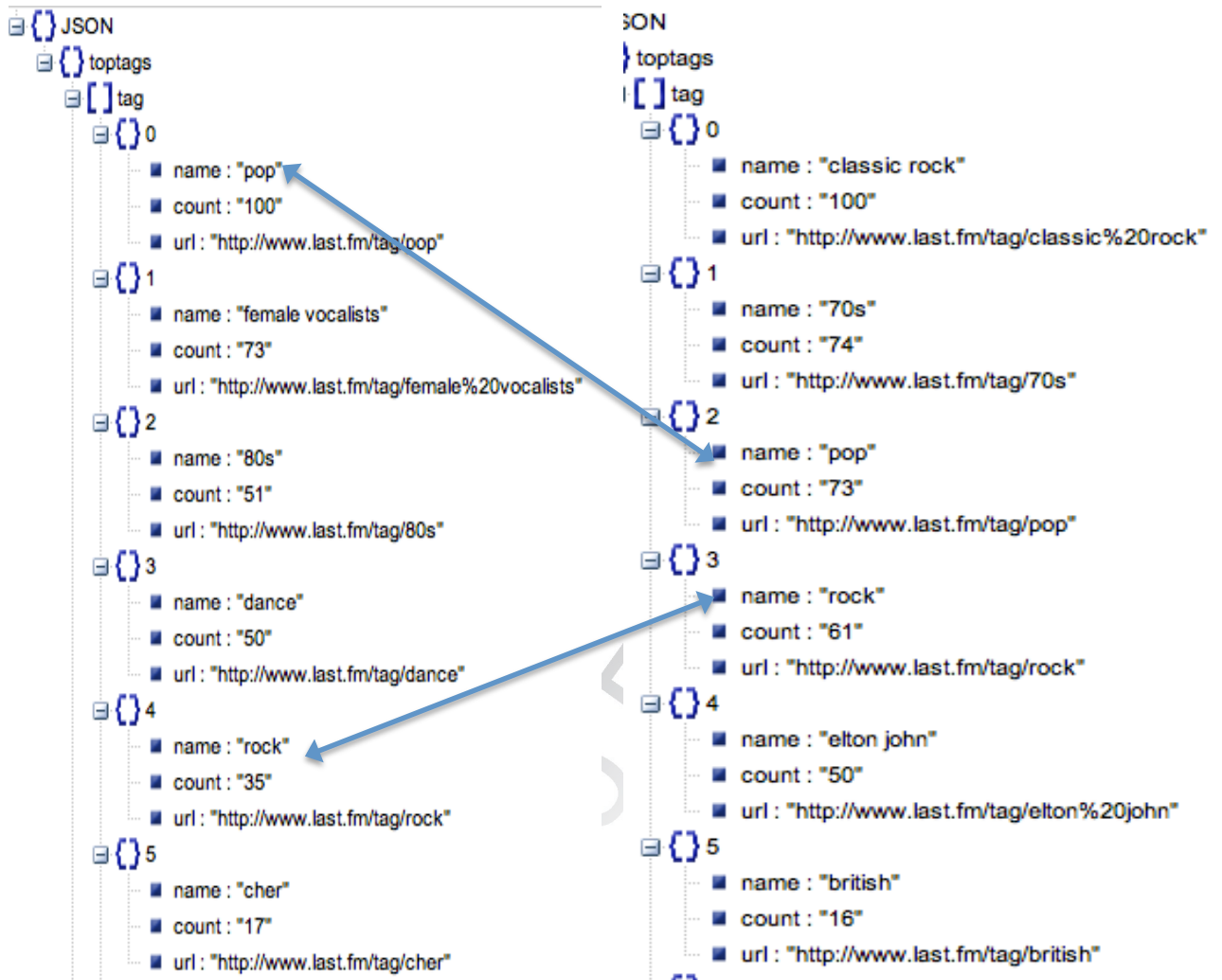
1. Ελέγχει το `NSArray` με τα tags του καλλιτέχνη
2. Για κάθε tag ζητάει μέσω Core Data το σωσμένο στη βάση `TrackTags` αντικείμενο μοντέλο που έχει ως ιδιότητα το ίδιο tag όνομα (`tagName`).
3. Για κάθε αντικείμενο `TrackTags` πολλαπλασιάζει το σκορ του με το σκορ του εν λόγω tag του καλλιτέχνη.
4. Δημιουργεί ένα αντικείμενο μοντέλο `ArtistTrackScore` όπου αποθηκεύει το όνομα του tag και το γινόμενο που προέκυψε από τον πολλαπλασιασμό στο βήμα (3) και το σώζει στη βάση δεδομένων.

ENTITIES	
<input type="checkbox"/> ArtistTrackScore	
<input type="checkbox"/> TrackTags	
<input type="checkbox"/> TrackUntaggable	

FETCH REQUESTS	
----------------	--

Attributes	
Attribute	Type
<input type="checkbox"/> artistID	Integer 64
<input type="checkbox"/> score	Integer 16
<input type="checkbox"/> trackID	Integer 64

5. Αθροίζει όλα τα σωσμένα γινόμενα του βήματος (4) για κάθε `MPMediaItemPropertyPersistentID` (μοναδικός κωδικός τραγουδιού)
6. Ταξινομεί τα αθροίσματα του βήματος (5) από το μεγαλύτερο στο μικρότερο.
7. Επιλέγει τα πρώτα 15 τραγούδια ή ακόμα περισσότερα εάν το άθροισμα είναι πάνω από 10.000 (είναι πολύ σχετικά δηλαδή).
8. Παρουσιάζει τη λίστα με τα τραγούδια στον χρήστη έτοιμη για αναπαραγωγή.



tag name	artist score	track score	total score
pop	100	73	7300
rock	35	61	2135

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ - ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΔΙΚΤΥΟΚΕΝΤΡΙΚΑ ΣΥΣΤΗΜΑΤΑ

Δειγματοληπτική Ανάλυση από την κονσόλα ανάπτυξης κατά την δημιουργία
μουσικής λίστας

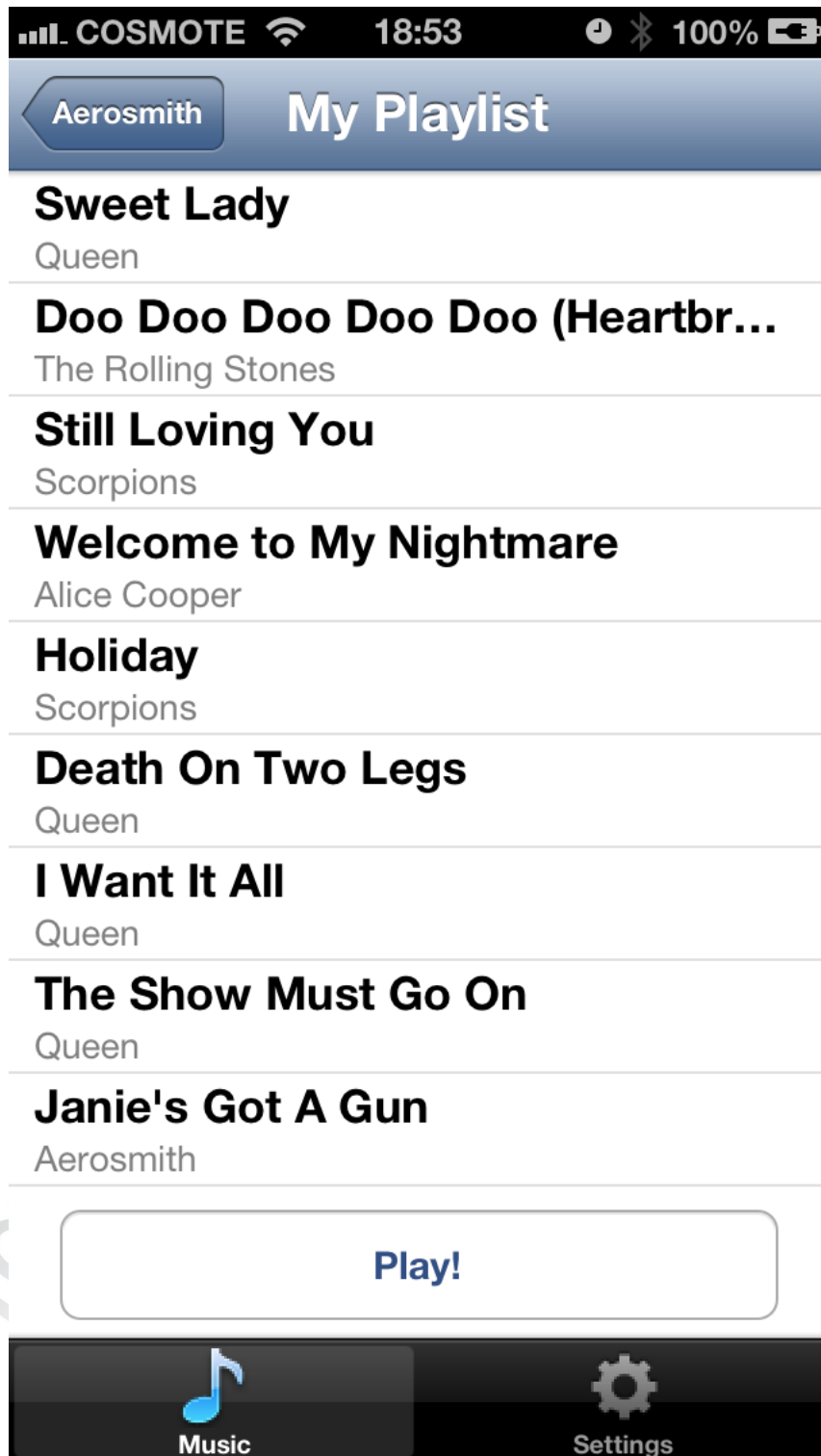
Getting tags for: Aerosmith
Trying to get Artist tags for Aerosmith
Found 100 tags
Tracks found. Calculating Scores.
track 1020503926 matched for tag rock with score 9000
track 1318044201 matched for tag rock with score 4600
track 536035745 matched for tag rock with score 5900
track 421639706 matched for tag rock with score 6300
track 896103699 matched for tag rock with score 4100
track 421652372 matched for tag 70s with score 435
track 421652901 matched for tag 70s with score 1044
track -273429506 matched for tag 70s with score 2349
track 1084931856 matched for tag 70s with score 1914
track -1862355127 matched for tag 70s with score 319
track 421638272 matched for tag 70s with score 290
track 1045357040 matched for tag 70s with score 2117
track -839570485 matched for tag 70s with score 725
track 111057477 matched for tag 70s with score 348

All Scores Calculated and Saved.

856 songs matched

Queen - Sweet Lady matched ranked 1 score 26481
The Rolling Stones - Doo Doo Doo Doo Doo (Heartbreaker)
matched ranked 2 score 26319
Scorpions - Still Loving You matched ranked 3 score 25863
Alice Cooper - Welcome to My Nightmare matched ranked 4 score
25751
Scorpions - Holiday matched ranked 5 score 24343
Queen - Death On Two Legs matched ranked 6 score 24208
Queen - I Want It All matched ranked 7 score 23881
[421652700] is no longer in Music Library hence all tags were
deleted
[421652893] is no longer in Music Library hence all tags were
deleted
Queen - The Show Must Go On matched ranked 8 score 23332
Aerosmith - Janie's Got A Gun matched ranked 9 score 23129
Queen - Princes Of The Universe matched ranked 10 score 22969
ZZ Top - Sharp Dressed Man matched ranked 11 score 22722
Queen - Lazing On A Sunday Afternoon matched ranked 12 score
22581
Queen - Good Company matched ranked 13 score 22557
Queen - Seaside Rendezvous matched ranked 14 score 22410
Led Zeppelin - The Battle Of Evermore matched ranked 15 score
22239
ZZ Top - Gimme All Your Lovin' matched ranked 16 score 22214
Dire Straits - Walk of Life matched ranked 17 score 22213
Queen - Bicycle Race matched ranked 18 score 22198

Led Zeppelin - Immigrant Song matched ranked 19 score 22016



10 Προβλήματα κατά την Ανάπτυξη και Επίλυση τους

10.1 Ανομοιόμορφες εικόνες από Last.FM

Το API call `getArtistInfo` το οποίο επιστρέφει μεταξύ άλλων και την εικόνα που εμφανίζεται στο προφίλ του καλλιτέχνη δεν εγγυάται για το μέγεθος και τις αναλογίες τους.

Μια εικόνα μπορεί να έχει μεγαλύτερο ύψος από ότι πλάτος και το αντίστροφο. Το XCode προσφέρει τις συγκεκριμένες επιλογές για το πως ένα Image View αντικείμενο θα απεικονίσει μια εικόνα.

- Scale to Fill
- Aspect Fit
- Aspect Fill

Η πρώτη περίπτωση το "Scale to Fill" σταματάει να διατηρεί την αναλογία των διαστάσεων μιας εικόνας την προσαρμόζει στο μέγεθος του Image View και το γεμίζει ολόκληρο **αλλά** η εικόνα φαίνεται παραμορφωμένη.



Swedish [pop](#) group Ace of Base was formed in Göteborg, Sweden in 1990. The band's original lineup consisted of [Ulf Ekberg](#) (Buddha), and siblings [Jonas Berggren](#) (Joker), [Linn Berggren](#) and



Similar Artists

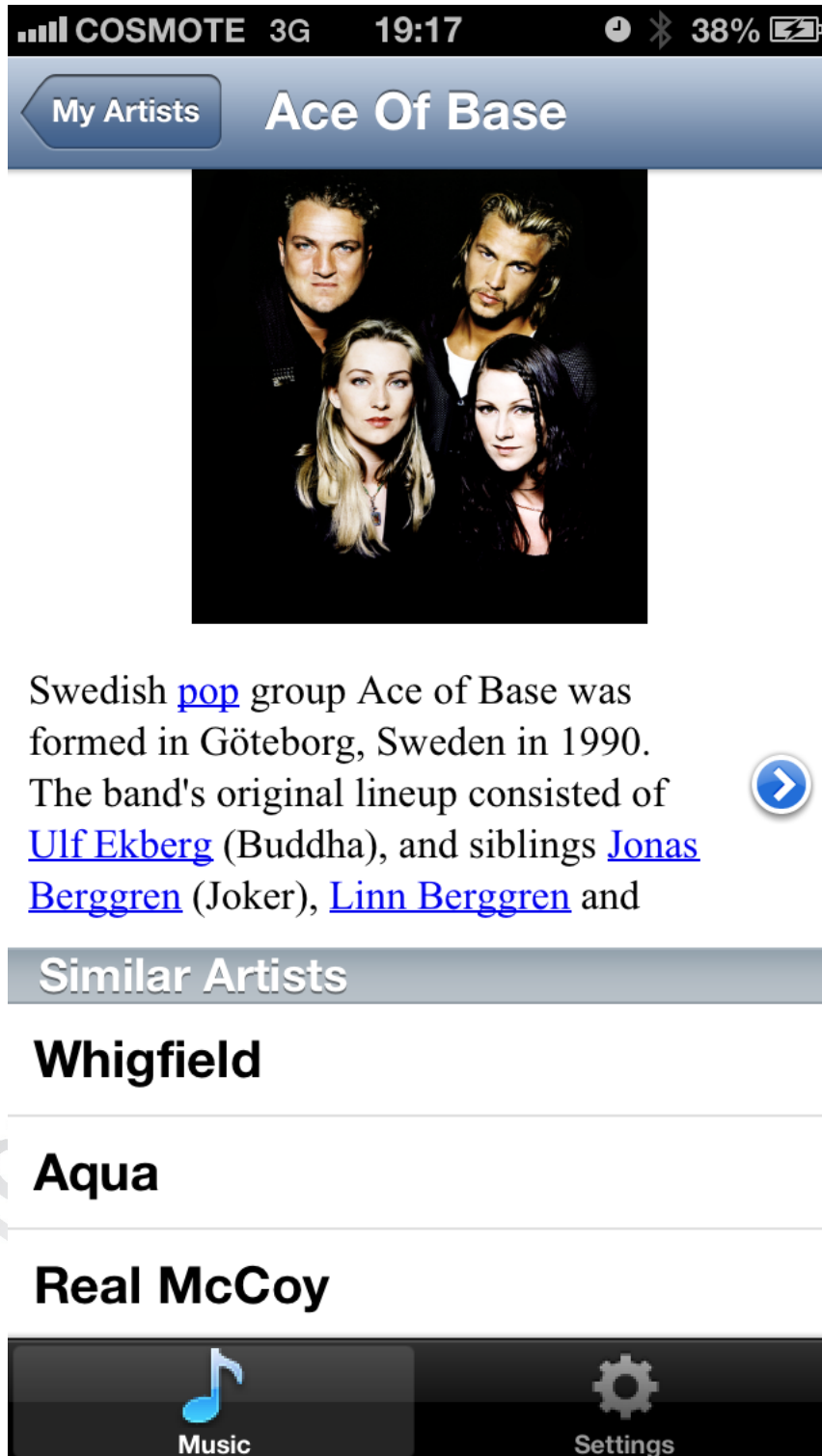
Whigfield

Aqua

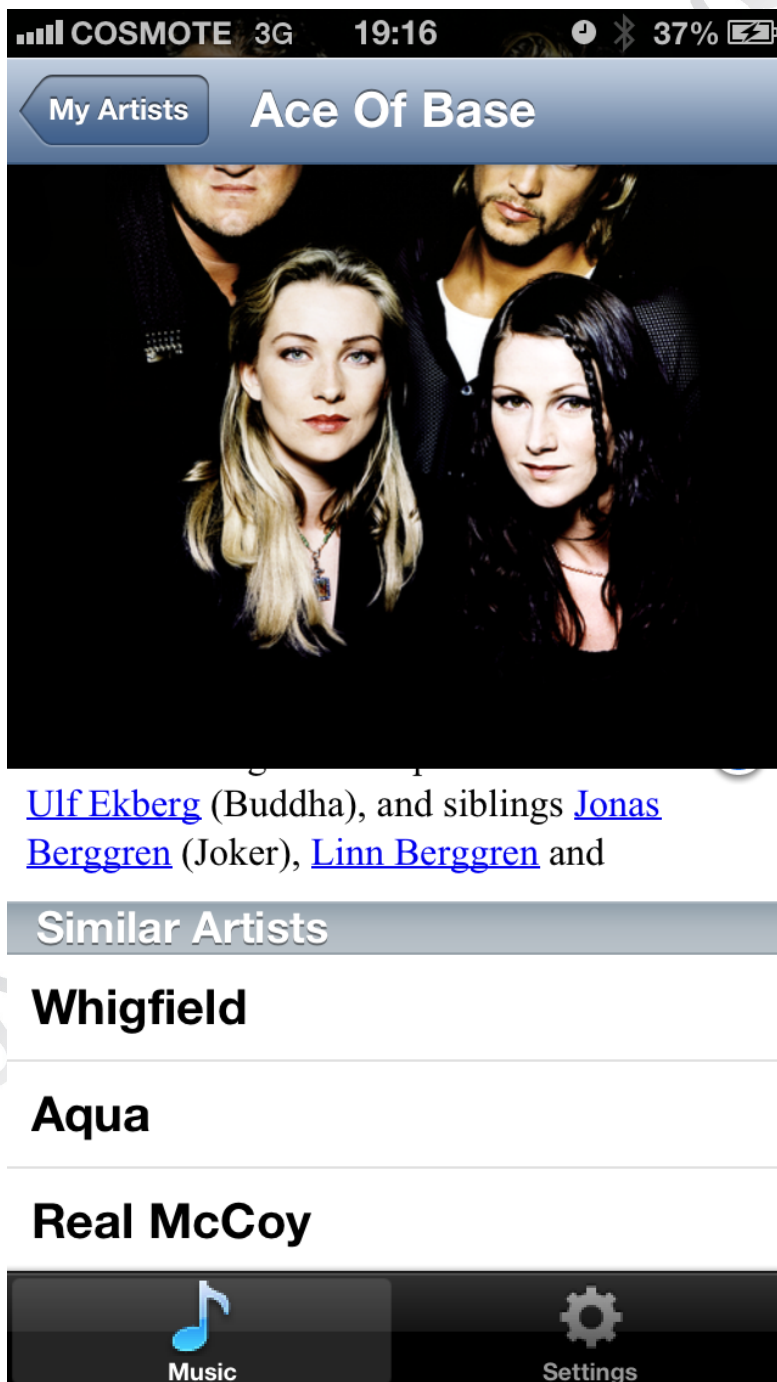
Real McCoy



Η επιλογή Aspect Fit διατηρεί τις αναλογίες της εικόνας και μετατρέπει και τις δύο διαστάσεις της εικόνας ώστε αυτή να χωρέσει στο Image View και να μην είναι παραμορφωμένη. Το αποτέλεσμα είναι το παρακάτω.



Τέλος η επιλογή Aspect Fill επιχειρεί να γεμίσει όλο το Image View χωρίς να παραμορφώσει την εικόνα και χωρίς να αφήσει κενές λευκές περιοχές. Η συγκεκριμένη επιλογή εκτός του ότι δε λειτουργεί πάντα σωστά (αλλάζει το μέγεθος του Image View) κόβει και μέρος της εικόνας που θα θέλαμε να παραμείνει (πχ πρόσωπα).



Η λύση

Η λύση απαιτεί την κατασκευή ενός αλγορίθμου που θα αντιλαμβάνοτανε ποια πλευρά της εικόνας είναι η μεγαλύτερη, ποιες είναι οι διαστάσεις του image view και ανάλογα θα αλλάξει της διαστάσεις της εικόνας και θα την κόψει (crop) ώστε να **μην** είναι παραμορφωμένη, να **γεμίσει** το Image View και να **μην κόψει** σημαντική πληροφορία (πχ πρόσωπα).

Παρακάτω παρατίθεται ο κώδικας που αναλαμβάνει να κάνει τις απαραίτητες μετατροπές. Η μέθοδος δέχεται ένα αντικείμενο διαστάσεων (τις διαστάσεις του Image View) μια εικόνα πηγής (την αρχική εικόνα από το Last.FM) και επιστρέφει τη νέα εικόνα μετά τις μετατροπές.

```
-  
(UIImage*) imageByScalingAndCroppingForSize:(CGSize) targetSize  
: (UIImage*) sourceImage  
{  
    UIImage *newImage = nil;  
    CGSize imageSize = sourceImage.size;  
    CGFloat width = imageSize.width;  
    CGFloat height = imageSize.height;  
    CGFloat targetWidth = targetSize.width;  
    CGFloat targetHeight = targetSize.height;  
    CGFloat scaleFactor = 0.0;  
    CGFloat scaledWidth = targetWidth;  
    CGFloat scaledHeight = targetHeight;  
    CGPoint thumbnailPoint = CGPointMake(0.0,0.0);  
  
    if (CGSizeEqualToSize(imageSize, targetSize) == NO)  
    {  
        CGFloat widthFactor = targetWidth / width;  
        CGFloat heightFactor = targetHeight / height;  
  
        if (widthFactor > heightFactor)  
        {  
            scaleFactor = widthFactor; // scale to fit height  
            NSLog(@"Height");  
        }  
        else
```

```
{
    scaleFactor = heightFactor; // scale to fit width
    NSLog(@"Height");
}

scaledWidth = width * scaleFactor;
scaledHeight = height * scaleFactor;

// center the image
if (widthFactor > heightFactor)
{
    thumbnailPoint.y = (targetHeight - scaledHeight) *
0.2;
}
else
{
    if (widthFactor < heightFactor)
    {
        thumbnailPoint.x = (targetWidth - scaledWidth) *
* 0.5;
    }
}

}

UIGraphicsBeginImageContext(targetSize); // this will crop

CGRect thumbnailRect = CGRectZero;
thumbnailRect.origin = thumbnailPoint;
thumbnailRect.size.width = scaledWidth;
thumbnailRect.size.height = scaledHeight;

[sourceImage drawInRect:thumbnailRect];

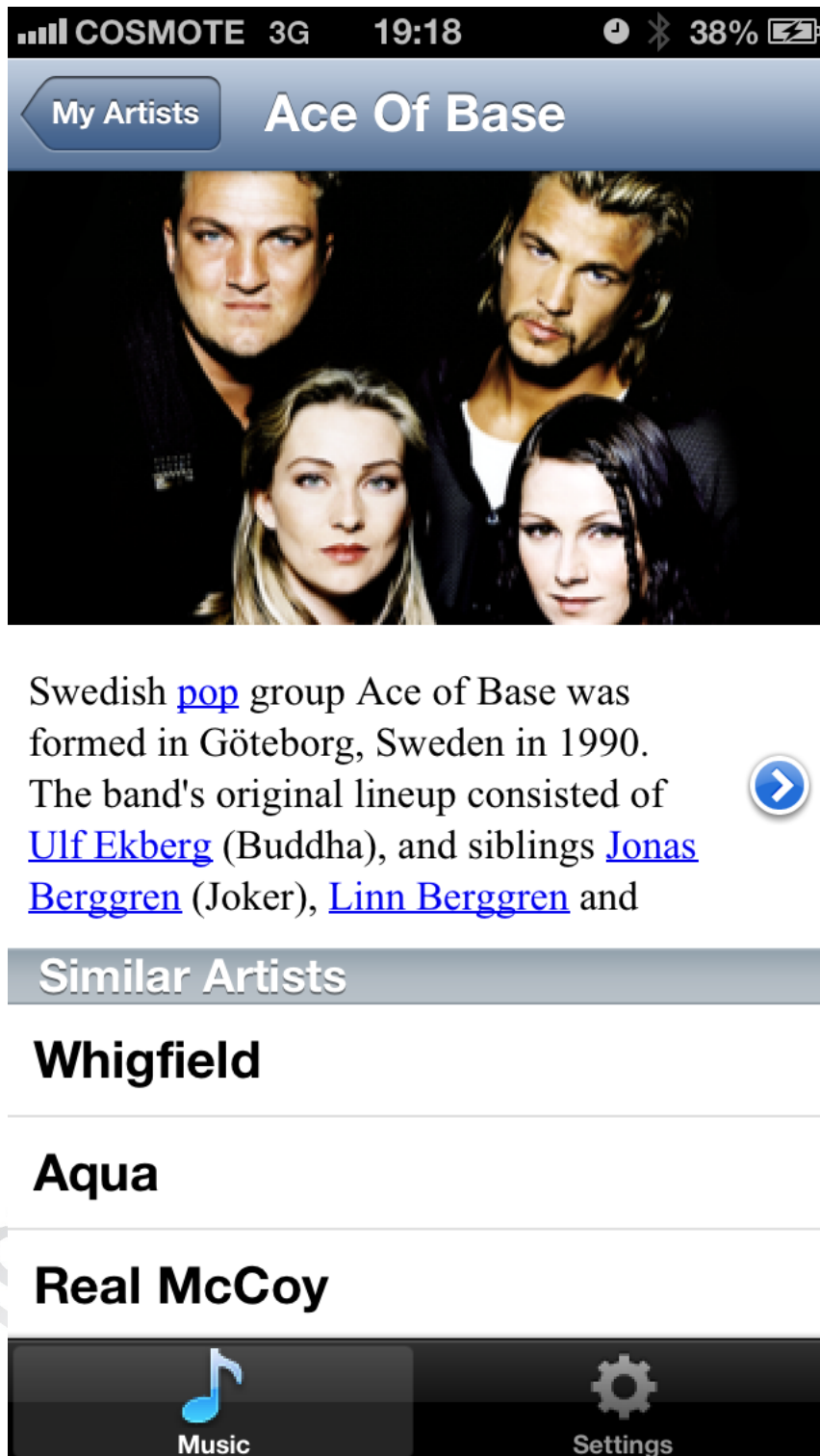
newImage = UIGraphicsGetImageFromCurrentImageContext();

if(newImage == nil)
{
    NSLog(@"could not scale image");
}

//pop the context to get back to the default
UIGraphicsEndImageContext();

return newImage;
}
```

Το τελικό αποτέλεσμα



The screenshot shows an iPhone music application interface. At the top, the status bar displays 'COSMOTE 3G', the time '19:18', and a battery level of '38%'. Below the status bar is a navigation bar with a 'My Artists' button and the artist name 'Ace Of Base'. The main content area features a large photograph of the four band members. Below the photo is a text description: 'Swedish [pop](#) group Ace of Base was formed in Göteborg, Sweden in 1990. The band's original lineup consisted of [Ulf Ekberg](#) (Buddha), and siblings [Jonas Berggren](#) (Joker), [Linn Berggren](#) and'. To the right of the text is a blue circular arrow icon. Below the text is a 'Similar Artists' section with a grey header. Underneath, three artist names are listed: 'Whigfield', 'Aqua', and 'Real McCoy'. At the bottom of the screen is a dark navigation bar with two buttons: 'Music' (with a blue musical note icon) and 'Settings' (with a grey gear icon).

10.2 Εγγραφή στη βάση δεδομένων από ξεχωριστά νήματα ταυτόχρονα

Η objective-c δεν επιτρέπει δυο ξεχωριστά νήματα να κάνουν εγγραφές στο μοντέλο (βάση δεδομένων στην συγκεκριμένη περίπτωση). Πρέπει το κάθε νήμα να χρησιμοποιεί άλλο context και όταν κάθε νήμα επιθυμεί να γράψει στη βάση πρέπει να συγχρονιστεί με τα υπόλοιπα νήματα. Στο παρακάτω παράδειγμα φαίνεται πως ένα νήμα που τρέχει στο background καλεί το main νήμα της εφαρμογής ώστε να συγχρονιστεί μαζί του και να γράψει στη βάση.

```
/* Save notification handler for the background context */  
- (void)backgroundContextDidSave:(NSNotification  
*)notification {  
    /* Make sure we're on the main thread when updating the  
    main context */  
    if (![NSThread isMainThread]) {  
        [self  
performSelectorOnMainThread:@selector(backgroundContextDidSave  
:)  
withObject:notification  
waitUntilDone:NO];  
        return;  
    }  
    // NSLog(@"Called");  
    // [self uiUpdate];  
    /* merge in the changes to the main context */  
    [context  
mergeChangesFromContextDidSaveNotification:notification];  
}
```

10.3 Κλήση Ασύγχρονης μεθόδου από Σύγχρονο νήμα

Η μέθοδος που αναλύει τη μουσική του χρήστη τραγούδι προς τραγούδι τρέχει σε ξεχωριστό νήμα μεν αλλά είναι σύγχρονη. Αυτό σημαίνει πως θα πρέπει να αναλυθεί πρώτα ένα τραγούδι για να ξεκινήσει η ανάλυση του επόμενου.

Όμως κατά τη διάρκεια μιας ανάλυσης ενός τραγουδιού πρέπει να γίνει κλήση στα web services του Last.FM. Οι κλήσεις στο web είναι από τη φύση τους ασύγχρονες διότι δε γνωρίζουμε πότε θα επιστραφούν τα αποτελέσματα από το web service.

Το πρόβλημα που προέκυψε λοιπόν είναι πως έφευγε το ασύγχρονο web service request και ο βρόχος ανάλυσης πήγαινε στο επόμενο τραγούδι. Όταν επέστρεφε αποτελέσματα το web service τα tags γραφόντουσαν στο ID του τραγουδιού στο οποίο είχε φτάσει ο βρόχος, συνεπώς λανθασμένα.

Η λύση βρέθηκε με το να καλείται η ασύγχρονη κλήση στο web service στο main thread που ανάγκαζε το background thread της ανάλυσης να περιμένει.

Ο κώδικας παρακάτω δείχνει πως.

```
if(tagsExist == 0){
    [NSThread sleepForTimeInterval:1]; //LAST.FM
    LIMITATION
    sgt = [[SongGetTags alloc] init];
    sgt.delegate = self;

    songTagsText = @"";

    [self
performSelectorOnMainThread:@selector(requestTagsFromLastFM)
withObject:nil waitUntilDone:YES];
}
```

11 Μελλοντικές Βελτιώσεις

Οι υπηρεσίες και τα συστήματα ανάλυσης μουσικής και δημιουργίας μουσικών προτάσεων θα αναπτύσσονται συνεχώς. Τα μοντέλα θα εξελίσσονται και θα συνδυάζονται με μεγαλύτερη επιτυχία και ο όγκος πληροφορίας στο διαδίκτυο και στις υπηρεσίες κοινωνικής δικτύωσης θα μεγαλώνει συνεχώς.

Η εφαρμογή που κατασκευάστηκε χρήζει πολλών βελτιώσεων όπως:

- Η δημιουργία μουσικών λιστών βασισμένη σε ένα ή περισσότερα τραγούδια (όχι μόνο σε κάποιον καλλιτέχνη).
- Η παρουσίαση στο User Interface της εφαρμογής των δημοφιλέστερων tags που βρέθηκαν στη μουσική του βιβλιοθήκη ώστε να μπορεί να επιλέγει ένα ή περισσότερα από αυτά και έτσι να δημιουργούνται οι μουσικές του λίστες.
- Η βελτιστοποίηση του αλγορίθμου σε επίπεδο ταχύτητας.
- Η βελτιστοποίηση της ανάλυσης της μουσικής βιβλιοθήκης σε επίπεδο ταχύτητας.
- Η βελτιστοποίηση του αλγόριθμου ώστε να αποφεύγεται η πρόταση καλλιτεχνών παραπάνω από X φορές και να προτείνεται διαφορετική λίστα κάθε φορά ακόμα και αν τα κριτήρια είναι τα ίδια.

Τέλος η εφαρμογή θα μπορούσε να μεταφερθεί και σε άλλες πλατφόρμες, ακόμα και σε περιβάλλοντα desktop όπου ενδεχομένως θα μπορούν να χρησιμοποιηθούν ακόμα πιο πολύπλοκοι αλγόριθμοι.

12 Πηγές & Βιβλιογραφία

1. G. Adomavicius and A. Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, June 2005.
2. C. de la Bandera, A.M. Barbancho, L.J. Tardó'n, Simone Sammartino, and Isabel Barbancho. Humming Method for Content-Based Music Information Retrieval. *ISMIR 2011, (Ismir)*:49–54, 2011.
3. Sally Jo Cunningham, David Bainbridge, and Annette Falconer. More of an Art than a Science: Supporting the Creation of Playlists and Mixes. In *Seventh International Conference on Music Information Retrieval*, Victoria, Canada, 2006.
4. Pedro Cano, Markus Koppenberger, and Nicolas Wack. An Industrial-strength Content-based Music Recommendation System. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '05*, page 673, New York, New York, USA, 2005. ACM Press.
5. Qing Li, Byeong Man Kim, Dong Hai Guan, and Duk Oh. A Music Recommender Based on Audio Features. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 532–533, Sheffield, United Kingdom, 2004. ACM.

6. Jonathan L. Herlocker, Joseph a. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems*, 22(1):5–53, January 2004.
7. Kerstin Bischoff, Claudiu S Firan, Raluca Paiu, Wolfgang Nejdl, L S De, Cyril Laurier, and Mohamed Sordo. Music Mood and Theme Classification - A Hybrid Approach. In 10th International Society for Music Information Retrieval Conference, number Ismir, pages 657–662, 2009.
8. M.A. Casey, Remco Veltkamp, Masataka Goto, Marc Leman, Christophe Rhodes, and Malcolm Slaney. Content-based Music Information Retrieval: Current Directions and Future Challenges. *Proceedings of the IEEE*, 96(4):668–696, 2008.
9. T. Eerola and J. K. Vuoskoski. A Comparison of the Discrete and Dimensional Models of Emotion in Music. *Psychology of Music*, 39(1):18–49, August 2010.
10. Patrik N Juslin and Daniel Vastfjall. Emotional Responses to Music: the Need to Consider Underlying Mechanisms. *The Behavioural and brain sciences*, 31(5):559–621, October 2008.
11. Dmitry Bogdanov and Perfecto Herrera. How Much Metadata Do We Need in Music Recommendation? A Subjective Evaluation Using Preference Sets. In 12th International Society for Music Information Retrieval Conference, number ISMIR 2011, pages 97–102, 2011.
12. M. Datcu, H. Daschiel, A. Pelizzari, M. Quartulli, A. Galoppo, A. Colapicchioni, M. Pastori, K. Seidel, P.G. Marchetti, and S. D’Elia. Music Recommendation Using Content and Context Information Mining.

Geoscience and Remote Sensing, IEEE Transactions on, 41(12):2923–2936, 2003.

13. Ricardo A. Baeza-Yates and Chris H. Perleberg. Fast and Practical Approximate String Matching. In Combinatorial Pattern Matching, Third Annual Symposium, pages 185–192, 1992.
14. Luke Barrington, Reid Oda, and G. Lanckriet. Smarter Than Genius? Human Evaluation of Music Recommender Systems. In 10th International Society for Music Information Retrieval Conference, number ISMIR, pages 357–362, 2009.
15. O. Celma Herrada. Music Recommendation and Discovery in the Long Tail. PhD Thesis, 2009.
16. Robin Burke. Hybrid Recommender Systems: Survey and Experiments. User Modelling and User-Adapted Interaction, 12(4):331–370, 2002.
17. Dmitry Bogdanov, J. Serra, Nicolas Wack, Perfecto Herrera, and Xavier Serra. Unifying Low-level and High-level Music Similarity Measures. IEEE Transactions on Multimedia, 13(99):1–1, 2011.
18. Zeina Chedrawy and S. Abidi. A Web Recommender System for Recommending, Predicting and Personalizing Music Playlists. In Web Information Systems Engineering-WISE 2009, pages 335–342. Springer, 2009.
19. Francois Pachet and Daniel Cazaly. A Taxonomy of Musical Genres. In Content-Based Multimedia Information Retrieval Access Conference (RIAO), number April, 2000.
20. Jin Ha Lee, Bobby Bare, and Gary Meek. How Similar is too Similar? Exploring Users' Perception of Similarity in Playlist Evaluation. In

International Conference on Music Information Retrieval 2011, number
ISMIR, pages 109–114, 2011

21. Pedro Cano, Markus Koppenberger, and Nicolas Wack. Content-based Music Audio Recommendation. In Proceedings of the 13th annual ACM international conference on Multimedia, number ACM, pages 211–212, 2005.
22. Yajie Hu and Mitsunori Ogihara. Nextone Player: A Music Recommendation System Based on User Behaviour. In 12th International Society for Music Information
23. Bass Lines, Emiru Tsunoo, George Tzanetakis, and Nobutaka Ono. Beyond Timbral Statistics: Improving Music Classification Using Percussive. IEEE Transactions on Audio, Speech and Language Processing, 19(4):1003–1014, 2011.
24. Fang-Fei Kuo, Meng-Fen Chiang, Man-Kwan Shan, and Suh-Yin Lee. Emotion-based Music Recommendation by Association Discovery from Film Music. In Proceedings of the 13th annual ACM international conference on Multimedia - MUL- TIMEDIA '05, page 507, New York, New York, USA, 2005. ACM Press.
25. Parag Chordia, Mark Godfrey, and Alex Rae. Extending Content-Based Recommendation: The Case of Indian Classical Music. In ISMIR 2008: proceedings of the 9th International Conference of Music Information Retrieval, pages 571–576, 2008.
26. Douglas Eck, Paul Lamere, T. Bertin-Mahieux, and Stephen Green. Automatic Generation of Social Tags for Music Recommendation. Advances in neural information processing systems, 20:385–392, 2007.

27. Y.E. Kim, E.M. Schmidt, Raymond Migneco, B.G. Morton, Patrick Richardson, Jeffrey Scott, J.A. Speck, and Douglas Turnbull. Music Emotion Recognition: A State of the Art Review. In Proc. of the 11th Intl. Society for Music Information Retrieval (ISMIR) Conf, number Ismir, pages 255–266, 2010.
28. Chun-man Mak, Tan Lee, Suman Senapati, Yu-ting Yeung, and Wang-kong Lam. Similarity Measures for Chinese Pop Music Based on Low-level Audio Signal Attributes. In 11th International Society for Music Information Retrieval Conference, number ISMIR, pages 513–518, 2010.
29. Panagiotis Symeonidis, Maria Ruxanda, Alexandros Nanopoulos, and Yannis Manolopoulos. Ternary Semantic Analysis of Social Tags for Personalized Music Recommendation. In Proc. 9th ISMIR Conf, pages 219–224. Citeseer, 2008.
30. Beth Logan. Music Recommendation from Song Sets. In International Conference on Music Information Retrieval 2004, number October, pages 10–14, Barcelona, Spain, 2004.
31. X Hu and J. Stephen Downie. Exploring Mood Metadata: Relationships with Genre, Artist and Usage Metadata. In 8th International Conference on Music Information Retrieval, 2007. Retrieval Conference, number Ismir, pages 103–108, 2011.
32. Yazhong Feng and Y Zhuang. Popular Music Retrieval by Detecting Mood. In International Society for Music Information Retrieval 2003, volume 2, pages 375–376, 2003.
33. S Gosling. A Very Brief Measure of the Big-Five Personality Domains. *Journal of Research in Personality*, 37(6):504–528, December 2003.

34. Jean-julien Aucouturier and Francois Pachet. Music Similarity Measures: What is the Use? In Proceedings of the ISMIR, pages 157–163, 2002.
35. Will Hill, Larry Stead, Mark Rosenstein, George Furnas, and South Street. Recommending and Evaluating Choices in a Virtual Community of Use. Mosaic A Journal For The Interdisciplinary Study Of Literature, pages 5–12, 1995.
36. C Anderson. The Long Tail. Why the Future of Business is selling less of more. Hyperion Verlag, 2006.
37. D Jennings. Net, Blogs and Rock 'n' Rolls: How Digital Discovery Works and What It Means for Consumers. 2007.
38. Jyh-Shing Roger Jang and Hong-Ru Lee. A General Framework of Progressive Filtering and Its Application to Query by Singing/Humming. IEEE Transactions on Audio, Speech, and Language Processing, 16(2):350–358, February 2008.
39. Paul Lamere. Social Tagging and Music Information Retrieval. Journal of New Music Research, 37(2):101–114, June 2008.
40. George Tzanetakis, Student Member, and Perry Cook. Musical Genre Classification of Audio Signals. IEEE Transactions on Speech and Audio Processing, 10(5):293–302, 2002.
41. Terence Magno and Carl Sable. A Comparison of Signal of Signal-Based Music Recommendation to Genre Labels, Collaborative Filtering, Musicological Analysis, Human Recommendation and Random Baseline. In ISMIR 2008: proceedings of the 9th International Conference of Music Information Retrieval, pages 161–166, 2008.

42. Paul Resnick, Hal R Varian, and Guest Editors. Recommender Systems. *Communications of the ACM*, 40(3):56–58, 1997.
43. M Mandel. Song-level Features and Support Vector Machines for Music Classification. In *Proc. International Conference on Music*, 2005.
44. M. Ogihara, Bo Shao, Dingding Wang, and Tao Li. Music Recommendation Based on Acoustic Features and User Access Patterns. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(8):1602–1611, November 2009.
45. Pasi Saari, Tuomas Eerola, and Olivier Lartillot. Generalizability and Simplicity as Criteria in Feature Selection: Application to Mood Classification in Music. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(99):1–1, 2011.
46. J. Stephen Downie. Music Information Retrieval. *Annual Review of Information Science and Technology*, 37(1):295–340, January 2005. A Survey of Music Recommendation Systems.
47. Stuart D Simpson and Costas I Karageorghis. The Effects of Synchronous Music on 400-m Sprint Performance. *Journal of sports sciences*, 24(10):1095–102, October 2006.
48. Francois Pachet. Knowledge Management and Musical Metadata. In *Encyclopedia of Knowledge Management*. 2005.
49. Mark Levy and Mark Sandler. Music Information Retrieval Using Social Tags and Audio. *IEEE Transactions on Multimedia*, 11(3):383–395, 2009.
50. Edith L. M. Law, Luis Von Ahn, Roger B. Dannenberg, and Mike Crawford. Tagatune: A Game for Music and Sound Annotation. In *8th International Conference on Music Information Retrieval*, 2007.

51. A. Russell. A Circumplex Model of Affect. *Journal of personality and social psychology*, 39(6):1161–1178, 1980.
52. Kazuyoshi Yoshii, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and Hi-roshi G Okuno. Improving Efficiency and Scalability of Model-Based Music Recommender System Based on Incremental Training. In *ISMIR 2007: proceedings of the 8th International Conference of Music Information Retrieval*, number ISMIR, 2007.
53. Steffen Pauws, Berry Eggen, and Miles Davis. PATS : Realization and User Evaluation of an Automatic Playlist Generator PATS : Realization and User Evaluation of an Automatic Playlist Generator. In *3rd International Conference on Music Information Retrieval*, 2002.
54. Dan Yang and W.S. Lee. Disambiguating Music Emotion Using Software Agents. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR04)*, pages 52–58, 2004.
55. CBE Plaza. Uncovering Affinity of Artist to Multiple Genres from Social Behavior Data. In *ISMIR 2008: proceedings of the 9th*, pages 275–280, 2008. *A Survey of Music Recommendation Systems*.
56. Peter J. Rentfrow and Samuel D. Gosling. The Do Re Mi's of Everyday Life: The structure and personality correlates of music preferences. *Journal of Personality and Social Psychology*, 84(6):1236–1256, 2003.
57. P.C. Terry and C.I. Karageorghis. Psychophysical Effects of Music in Sport and Exercise: An Update on Theory, Research and Application. In *Proceedings of the 2006 Joint Conference of the Australian Psychological Society and the New Zealand Psychological Society: Psychology Bridging*

the Tasman: Science, Culture and Practice, pages 415–419. Australian Psychological Society, 2006.

58. Janet Marques and Pedro J Moreno. A Study of Musical Instrument Classification Using Gaussian Mixture Models and Support Vector Machines, 1999.
59. Peter J Rentfrow and Samuel D Gosling. Message in a Ballad: the Role of Music Preferences in Interpersonal Perception. *Psychological science*, 17(3):236–42, March 2006.
60. Xiaoyuan Su and Taghi M. Khoshgoftaar. A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, 2009(Section 3):1–19, 2009.
61. Bo Shao, Tao Li, and M. Ogihara. Quantify Music Artist Similarity Based on Style and Mood. In *Proceeding of the 10th ACM workshop on Web Information and Data Management*, pages 119–124. ACM, 2008.
62. KTG Tsoumakas and George Kalliris. Multi-Label Classification of Music into Emotions. In *ISMIR 2008: proceedings of the 9th International Conference of Music Information Retrieval*, pages 325–330, 2008.
63. Elias Pampalk, Tim Pohle, and Gerhard Widmer. Dynamic Playlist Generation Based on Skipping Behaviour. In *Proc. of the 6th ISMIR Conference*, volume 2, pages 634–637, 2005.
64. A Salomon. A Content-based Music Similarity Function. In *Cambridge Research Labs-Tech Report*, number June, 2001.
65. Yoav Shoham and Marko Balabannovic. Content-Based, Collaborative Recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

66. M Mann, TJ Cox, and FF Li. Music Mood Classification of Television Theme Tunes. In 12th International Society for Music Information Retrieval Conference, number Ismir, pages 735–740, 2011.
67. MI Mandel. A Web-based Game for Collecting Music Metadata. In In 8th Inter- national Conference on Music Information Retrieval (ISMIR), 2008.
68. F. Pachet and J.J. Aucouturier. Improving Timbre Similarity: How High is the Sky? Journal of negative results in speech and audio sciences, 1(1):1–13, 2004.
69. Douglas Turnbull, Luke Barrington, and Gert Lanckriet. Five Approaches to Collecting Tags for Music. In ISMIR 2008: proceedings of the 9th International Conference of Music Information Retrieval, pages 225–230, 2008.
70. Mark Levy. A Semantic Space for Music Derived from Social Tags. Austrian Computer Society, 1:12, 2007.
71. FRDCarpentier. EffectsofMusiconPhysiologicalArousal,ExplorationintoTempo and Genre. Media Psychology, 10(3):339–363, 2007.
72. Badrul Sarwar, George Karypis, and Joseph Konstan. Item-based Collaborative Filtering Recommendation Algorithms. Proceedings of the 10th, pages 285–295, 2001.
73. Janto Skowronek and M McKinney. Ground Truth for Automatic Music Mood Classification. In Proc. ISMIR, pages 4–5, 2006.
74. Nava Tintarev and Judith Masthoff. Effective Explanations of Recommendations: User-centred Design. In Proceedings of the 2007 ACM conference on Recommender systems, pages 153–156. ACM, 2007.

75. Rob van Gulik and Fabio Vignoli. Visual Playlist Generation on the Artist Map. In 5th International Conference on Music Information Retrieval, number ISMIR2005, pages 520–523, 2005.
76. Erdem Unal, Elaine Chew, Panayiotis G. Georgiou, and Shrikanth S. Narayanan. Challenging Uncertainty in Query by Humming Systems: A Fingerprinting Approach. IEEE Transactions on Audio, Speech, and Language Processing, 16(2):359– 371, February 2008.
77. Neel Sundaresan. Recommender Systems at the Long Tail. In of the fifth ACM conference on Recommender systems, number RecSys 2011, pages 1–5, 2011.
78. Alexandra Uitdenbogerd and van Schyndel Ron. A Review of Factors Affecting Music Recommender. In 3rd International Conference on Music Information Retrieval (2002), 2002.
79. Jun Wang, A.P. De Vries, and M.J.T. Reinders. Unifying User-based and Item- based Collaborative Filtering Approaches by Similarity Fusion Categories. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pages 501–508. ACM, 2006.
80. C.C. Wang, J.S.R. Jang, and Wennen Wang. An Improved Query by Singing/Humming System Using Melody and Lyrics Information. In 11th International Society for Music Information Retrieval Conference, number Ismir, pages 45–50, 2010.
81. Kazuyoshi Yoshii, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G Okuno. Hybrid Collaborative and Content-based Music Recommendation Using Probabilistic Model with Latent User Preferences.

In Proceedings of the 7th International Conference on Music Information Retrieval, pages 296–301, 2006.

82. Fabio Vignoli. A Music Retrieval System Based on User-driven Similarity and its Evaluation. In International Conference on Music Information Retrieval 2005, 2005.
83. Alan Page Fiske, Shinobu Kitayama, Hazel Rose Markus, and R E Nisbett. The Cultural Matrix of Social Psychology. 1998
84. Ju-chiang Wang, Hung-shin Lee, Hsin-min Wang, and Shyh-kang Jeng. Learning the Similarity of Audio Music in Bag-of-Frames Representation from Tagged Music Data. In International Conference on Music Information Retrieval 2011, number ISMIR, pages 85–90, 2011.
85. Yi-Hsuan Yang. Music Emotion Recognition. Tayler and Francis Group, 2011.
86. DingdingWang, TaoLi, and MitsunoriOgihara. TagsBetterThanAudioFeatures? The Effect of Joint use of Tags and Audio Content Features for Artistic Style Clustering. In International Conference on Music Information Retrieval 2010, number ISMIR, pages 57–62, 2010.
87. Xing Wang, Xiaou Chen, Deshun Yang, and Yuqian Wu. Music Emotion Classification of Chinese Songs Based on Lyrics using TF*IDF and Rhyme. In 12th International Society for Music Information Retrieval Conference, number Ismir, pages 765–770, 2011.