



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

**ΑΥΤΟΟΡΓΑΝΩΜΕΝΗ
ΔΙΑΔΙΚΤΥΑΚΗ ΡΑΔΙΟΦΩΝΙΑ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΧΡΙΣΤΟΦΟΡΟΥ ΔΗΜΗΤΡΙΑΔΗ

Βόλος, 2011

Η σελίδα αυτή είναι σκόπιμα λευκή.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

ΑΥΤΟΟΡΓΑΝΩΜΕΝΗ ΔΙΑΔΙΚΤΥΑΚΗ ΡΑΔΙΟΦΩΝΙΑ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΧΡΙΣΤΟΦΟΡΟΥ ΔΗΜΗΤΡΙΑΔΗ

Επιβλέπων : Δημήτριος Κατσαρός
Λέκτορας Τ.Μ.Η.Υ.Τ.Δ

Εγκρίθηκε από τη διμελή εξεταστική επιτροπή τον Οκτώβριο 2011.

(Υπογραφή)

.....
Δημήτριος Κατσαρός
Λέκτορας Τ.Μ.Η.Υ.Τ.Δ.

(Υπογραφή)

.....
Παναγιώτης Μποζάνης
Αναπληρωτής Καθηγητής Τ.Μ.Η.Υ.Τ.Δ.

Βόλος, Οκτώβριος 2011

(Υπογραφή)

.....

ΧΡΗΣΤΟΦΟΡΟΣ ΔΗΜΗΤΡΙΑΔΗΣ

Διπλωματούχος Μηχανικός Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων
Πανεπιστημίου Θεσσαλίας

© 2011 – All rights reserved

Περίληψη

Στόχος μας είναι ο σχεδιασμός, η υλοποίηση και η αξιολόγηση των δυνατοτήτων ενός διαδικτυακού ραδιοφώνου το οποίο θα έχει τη δυνατότητα να κατασκευάζει με δυναμικό τρόπο λίστες αναπαραγωγής τραγουδιών βασιζόμενο στις προτιμήσεις της κοινότητας. Πιο συγκεκριμένα, οι λίστες αναπαραγωγής θα δημιουργούνται με βάση αλγόριθμους που βασίζονται κυρίως στην δημοφιλία τους όπως αυτή εκφράζεται από την προτίμηση των ακροατών του ραδιοφώνου μας. Για παράδειγμα τραγούδια τα οποία προτιμώνται σε μεγαλύτερο βαθμό από τους χρήστες θα αναπαράγονται από το ραδιόφωνο συχνότερα σε σχέση με εκείνα τα οποία προτιμώνται από μικρότερες μερίδες της κοινότητας.

Η διαδικτυακή εφαρμογή που δημιουργήσαμε στοχεύει κατά κύριο λόγο σε νέους και αφανείς καλλιτέχνες, οι οποίοι θα έχουν τη δυνατότητα να διαμοιράζουν και να γνωστοποιούν τις δημιουργίες τους χωρίς να ταλαιπωρούνται από λίστες αναπαραγωγής που βασίζονται σε αδιαφανή μη-αντικειμενικά κριτήρια. Βασιστήκαμε στην παρατήρηση ότι υπάρχουν πολλοί καλλιτέχνες που απευθύνονται σε πολύ περιορισμένο κοινό (long tail) και οι δημιουργίες των οποίων θα ήταν ανέφικτο να διανεμηθούν -με πιο παραδοσιακές μεθόδους- στο ευρύ κοινό εξαιτίας των περιορισμών που εισάγονται σε χώρο και κόστος αποθήκευσης. Το διαδίκτυο όμως λύνει το πρόβλημα τόσο της χωρητικότητας όσο και της προσβασιμότητας, συνεπώς είναι δυνατό να φέρουμε σε επαφή αυτόν τον μεγάλο αριθμό καλλιτεχνών με ένα μεγαλύτερο ακροατήριο.

Απώτερος στόχος της εφαρμογής μας είναι το συλλογικό φιλτράρισμα της πληροφορίας, η οποία στην περίπτωση μας αντιπροσωπεύεται από τα τραγούδια που παράγονται από τους καλλιτέχνες, ώστε τελικά να αναδειχθεί το υποσύνολο αυτής που έχει τη θετικότερη απήχηση στην κοινότητα.

Λέξεις Κλειδιά: << διαδικτυακή εφαρμογή, ραδιόφωνο, συμμετοχικότητα, μουσική, long-tail >>

Πίνακας περιεχομένων

1	Εισαγωγή.....	6
1.1	Διαδίκτυο και συμμετοχικότητα	6
1.2	Το φαινόμενο της μακρινής ουράς	7
1.3	Διαδίκτυο και μουσική.....	8
2	Συστήματα πληροφοριακών συστάσεων και ψηφοφοριών	10
2.1	Συλλογικό φιλτράρισμα πληροφοριών	10
2.2	Συστάσεις μουσικής.....	12
2.3	Ιστοχώροι κοινωνικής δημοσίευσης	13
3	Στόχοι και προδιαγραφές.....	16
3.1	Συλλογική ανάδειξη νέων καλλιτεχνών.....	16
3.2	Αυτόματη οργάνωση και διαχείριση μουσικού περιεχομένου.....	18
4	Αλγόριθμοι	23
4.1	Σύστημα ψηφοφορίας	23
4.2	Δημιουργία ραδιοφωνικών σταθμών	28
4.3	Δημιουργία λιστών αναπαραγωγής.....	31
4.3.1	Επιλογή τραγουδιών.....	32
4.3.2	Διάταξη λίστας.....	33
5	Αρχιτεκτονική συστήματος και τεχνολογίες υλοποίησης	36
5.1	Αρχιτεκτονική.....	36
5.2	Τεχνολογίες.....	37
5.2.1	Java.....	38
5.2.2	MySQL.....	44
5.2.3	JavaScript	45
5.2.4	HTML 5.0.....	46
5.2.5	Rest	47
6	Περιβάλλον χρήσης και σενάρια χρήσης.....	48
6.1	Διεπαφή χρήσης.....	48

6.2	Σενάριο χρήσης.....	60
7	Σύνοψη και μελλοντικές προοπτικές	70
8	Σύνδεσμοι.....	73

Πίνακας Εικόνων

Εικόνα 1-1: Long tail graph.....	7
Εικόνα 2-1: Collaborative filtering process.....	11
Εικόνα 2-2: Social publishing traits	14
Εικόνα 3-1: Σχηματική αναπαράσταση των ενεργειών κάθε χρήστη	18
Εικόνα 5-1: Η αρχιτεκτονική της διαδικτυακής εφαρμογής	37
Εικόνα 5-2: Οι λειτουργίες που εκτελεί ένα Servlet	39
Εικόνα 5-3: Διαδικασία μετάφρασης και εκτέλεσης ενός JSP.....	41
Εικόνα 5-4: Βήματα του μοντέλου MVC.....	42
Εικόνα 5-5: Κεντρική σελίδα index	49
Εικόνα 5-6: Σελίδα εγγραφής στο σύστημα	50
Εικόνα 5-7: Κεντρική σελίδα home	50
Εικόνα 5-8: Καρτέλα «καλλιτέχνες».....	51
Εικόνα 5-9: Σελίδα προφίλ του χρήστη.....	54
Εικόνα 5-10: Προσωπική σελίδα καλλιτέχνη όταν ο χρήστης δεν σχετίζεται με αυτόν.....	55
Εικόνα 5-11: Προσωπική σελίδα καλλιτέχνη όταν ο χρήστης σχετίζεται με αυτόν	56
Εικόνα 5-12: Σελίδα του άλμπουμ όταν ο χρήστης δεν σχετίζεται με τον καλλιτέχνη.....	57
Εικόνα 5-13: Σελίδα του άλμπουμ όταν ο χρήστης σχετίζεται με τον καλλιτέχνη	58
Εικόνα 5-14: Σελίδα τραγουδιού.....	59
Εικόνα 5-15: Σελίδα ραδιοφωνικού σταθμού.....	59
Εικόνα 5-16: Γενική αναζήτηση πληροφορίας.....	60
Εικόνα 5-17: Η αρχική σελίδα πριν τη σύνδεση στο σύστημα (login)	61
Εικόνα 5-18: Η αρχική σελίδα μετά τη σύνδεση στο σύστημα (login).....	61
Εικόνα 5-19: Το προφίλ του χρήστη	62
Εικόνα 5-20: Προσθήκη νέου καλλιτέχνη.....	63
Εικόνα 5-21: Σελίδα προφίλ του νέου καλλιτέχνη.....	64
Εικόνα 5-22: Προσθήκη νέου άλμπουμ	65
Εικόνα 5-23: Προσθήκη εικόνας εξωφύλλου στο άλμπουμ.....	66
Εικόνα 5-24: Μεταφόρτωση τραγουδιού για το άλμπουμ	66
Εικόνα 5-25: Επεξεργασία στοιχείων τραγουδιού	67
Εικόνα 5-26: Μεταφόρτωση εικόνας προφίλ για τον καλλιτέχνη.....	67

Εικόνα 5-27: Ενέργειες σελίδας αναπαραγωγής	68
Εικόνα 5-28: Απόδοση ετικετών	69

1

Εισαγωγή

Η ραγδαία εξέλιξη της τεχνολογίας έχει ανοίξει νέους δρόμους στην επικοινωνία, στη συνεργασία και στον τρόπο με τον οποίο διαδίδονται νέες ιδέες. Ένα από τα πιο σημαντικά χαρακτηριστικά του διαδικτύου, όπως το γνωρίζουμε σήμερα, είναι ότι δίνει τη δυνατότητα στους χρήστες του να συνεισφέρουν διαμοιράζοντας περιεχόμενο που οι ίδιοι έχουν δημιουργήσει, δαπανώντας ουσιαστικά ελάχιστο χρόνο και χρήμα. Στις περισσότερες περιπτώσεις μάλιστα, δεν απαιτείται καμία επιπλέον εξειδίκευση στη χρήση συγκεκριμένων εργαλείων, το μόνο που χρειάζεται είναι ένας απλός περιηγητής. Όλα τα παραπάνω έχουν οδηγήσει στη δημιουργία μίας νέας συμμετοχικής κουλτούρας στα πλαίσια της οποίας το άτομο δε δρα μόνο ως αποδέκτης, αλλά και ως δημιουργός/παραγωγός της πληροφορίας.

1.1 Διαδίκτυο και συμμετοχικότητα

Η συμμετοχικότητα είναι ένας όρος που συσχετίστηκε για πρώτη φορά με το διαδίκτυο όταν δημιουργήθηκε η ανάγκη διαχωρισμού των όρων web 2.0 και web 1.0. Εάν προσπαθούσε κανείς να περιγράψει με λίγα λόγια το είδος των υπηρεσιών και εφαρμογών που παρέχονται σήμερα στον παγκόσμιο ιστό, μερικές από τις λέξεις που σίγουρα θα χρησιμοποιούσε είναι οι εξής:

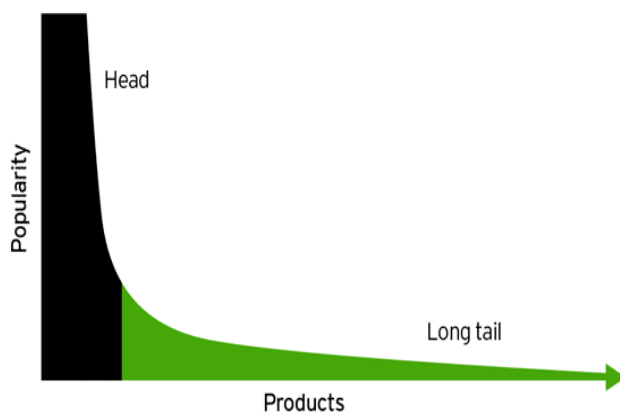
- ευχρηστία (*usability*)
- συνεργασία (*collaboration*)

- σύγκλιση (*convergence*)
- φορητότητα (*mobility*)
- οικονομία (*economy*)
- συμμετοχικότητα (*participation*)

Παρότι όλα τα παραπάνω χαρακτηριστικά - και όχι μόνο - συνθέτουν το σύγχρονο παγκόσμιο ιστό, η συμμετοχικότητα είναι ίσως το χαρακτηριστικό που έχει επιφέρει τις μεγαλύτερες αλλαγές στη μορφή και στον τρόπο χρήσης του διαδικτύου. Η νέα τάση που έχει επικρατήσει στο διαδίκτυο θέλει τις web εφαρμογές να λειτουργούν περισσότερο ως πλατφόρμες δράσης, και όχι ως στατικές ιστοσελίδες, οι οποίες παρέχουν τα μέσα στους χρήστες να δημιουργούν, να διαχειρίζονται και να οργανώνουν το περιεχόμενό τους. Σελίδες όπως το Wikipedia, το YouTube, το Flickr και το MySpace έχουν δώσει τη δυνατότητα στους χρήστες να γνωστοποιούν τις ιδέες τους και να μοιράζονται τις δημιουργίες τους με το ευρύ κοινό, ακόμα και να βελτιώνουν και να αναδιανέμουν υπάρχον περιεχόμενο, προς όφελος της κοινότητας.

1.2 Το φαινόμενο της μακριάς ουράς

Ο όρος «μακριά ουρά» (αγγλ. *long tail*) αναλύθηκε από τον αρχισυντάκτη του περιοδικού Wired Chris Anderson σε σχετικό άρθρο του τον Οκτώβριο του 2004⁴. Σύμφωνα με το φαινόμενο της μακριάς ουράς εντοπίζεται περιορισμένος αριθμός προϊόντων με πολύ μεγάλη ζήτηση, ενώ στον αντίποδα, υπάρχει ένα τεράστιο πλήθος προϊόντων για τα οποία η ζήτηση είναι πολύ περιορισμένη. Αυτή η σχέση ζήτησης-πλήθους προϊόντων έχει παρατηρηθεί πως ακολουθεί εκθετική μείωση όπως φαίνεται στο σχήμα.



Εικόνα 1-1: Long tail graph

Για να γίνει κατανοητή η παραπάνω σχέση, ας θεωρήσουμε το παράδειγμα ενός παραδοσιακού καταστήματος ενοικίασης ταινιών, το οποίο διαθέτει πεπερασμένο -και τις περισσότερες φορές αρκετά περιορισμένο- αποθηκευτικό χώρο. Εάν υποθέσουμε ότι τα ράφια του καταστήματος μπορούν να φιλοξενήσουν μέχρι και 100 διαφορετικές ταινίες, μοιάζει ορθολογικό οι 100 αυτές ταινίες να επιλεγούν από την κατηγορία των μεγαλύτερων εισπρακτικών επιτυχιών της εποχής, καθώς έτσι είναι βέβαιο ότι κάθε ταινία θα ενοικιαστεί αρκετές φορές ώστε να εξισορροπηθεί το κόστος αγοράς και αποθήκευσής της από το κατάστημα. Παρότι μία τέτοια πολιτική μοιάζει επιχειρηματικά ορθή καθώς μεγιστοποιεί το κέρδος του καταστήματος, δε λαμβάνει υπόψη τη μερίδα του καταναλωτικού κοινού που στρέφεται προς λιγότερο εμπορικές ταινίες, δηλαδή στο long tail. Ένα τέτοιο κοινό είναι λιγότερο πιθανό να εντοπίσει ταινίες της αρεσκείας του στο συγκεκριμένο κατάστημα, με συνέπεια να καταναλώνει πολύ λιγότερο.

Τι θα συνέβαινε όμως εάν το κατάστημα διέθετε άφθονο και φθηνό αποθηκευτικό χώρο και το κόστος αγοράς των ταινιών έπεφτε στο μισό; Το πιο πιθανό είναι πως θα αγόραζε όσο το δυνατόν περισσότερες ταινίες του επέτρεπε ο προϋπολογισμός του και κατά συνέπεια θα προσέγγιζε ένα πολύ μεγαλύτερο καταναλωτικό κοινό, με αποτέλεσμα να αυξήσει κατά πολύ τα κέρδη του.

Σύμφωνα με στατιστικά στοιχεία που έχει δώσει στη δημοσιότητα το Amazon και τα οποία επίσης μελετά στο άρθρο του ο Chris Anderson, τουλάχιστον το 50% των πωλήσεων που σχετίζονται με βιβλία, προέρχονται από βιβλία που δεν ανήκουν στα κορυφαία 130,000 σε πωλήσεις. Αυτό αποδεικνύει ότι η αγορά του long tail κάθε άλλο παρά περιορισμένη είναι.

1.3 Διαδίκτυο και μουσική

Η βιομηχανία που στηρίζει τα κέρδη της στη διανομή προϊόντων που σχετίζονται με πολυμέσα είναι ίσως η βιομηχανία που έχει επηρεαστεί περισσότερο από τις ραγδαίες μεταβολές στον τρόπο διακίνησης της πληροφορίας. Τεράστιες ποσότητες δεδομένων διακινούνται σχεδόν ανεξέλεγκτα στο διαδίκτυο, με ελάχιστο ουσιαστικά κόστος σε χρόνο και χρήμα. Επιπλέον, με το κόστος αποθήκευσης να μειώνεται διαρκώς, ένας συνηθισμένος προσωπικός υπολογιστής μπορεί να φιλοξενήσει μερικές εκατοντάδες ταινίες και αρκετές χιλιάδες μουσικά κομμάτια, καθιστώντας έτσι περιττή και ασύμφορη την απόκτηση δίσκων CD και DVD. Οι αναδυόμενες τεχνολογίες cloud επιφέρουν επιπρόσθετες θεμελιώδεις αλλαγές.

Στην παραπάνω παρατήρηση έχουν βασιστεί πολλές εταιρίες διανομής υλικού πολυμέσων, οι οποίες προσαρμοζόμενες στις απαιτήσεις της εποχής και εκμεταλλευόμενες τις δυνατότητες

που τους παρέχει ένα διαρκώς εξελισσόμενο διαδίκτυο, έχουν προχωρήσει στον εκσυγχρονισμό του τρόπου διανομής των προϊόντων τους. Εταιρίες όπως η *Apple* με το *iTunes Store*, διανέμουν με ψηφιακό τρόπο το υλικό τους, κοστολογώντας πλέον το τραγούδι και όχι το άλμπουμ, δίνοντας τη δυνατότητα στους χρήστες να αγοράζουν αποκλειστικά τα τραγούδια ενός άλμπουμ που επιθυμούν χωρίς να επιβαρύνονται με το κόστος αγοράς ολόκληρου του άλμπουμ. Υπηρεσίες όπως το *Rhapsody*, διαθέτουν online βάσεις δεδομένων με μερικές εκατοντάδες χιλιάδες τραγούδια και επιτρέπουν την πρόσβαση σε αυτές με μηνιαία συνδρομή που δεν ξεπερνά το κόστος αγοράς ενός άλμπουμ, ουσιαστικά χωρίς να απαιτούν από το χρήστη να αποθηκεύει ούτε ένα MegaByte στον υπολογιστή του. Υπηρεσίες κοινωνικής δικτύωσης όπως το *MySpace* δίνουν τη δυνατότητα σε ανερχόμενους καλλιτέχνες να έρθουν σε επαφή με το ακροατήριό τους και να διαφημίσουν τη μουσική τους πρακτικά με μηδενικό κόστος, αφαιρώντας την ανάγκη μεσολάβησης δισκογραφικών εταιριών.

Όλες οι παραπάνω μορφές διανομής μουσικού –και όχι μόνο- περιεχομένου έχουν γίνει αποδεκτές με ενθουσιασμό τόσο από την κοινότητα των καταναλωτών όσο και από την κοινότητα των δημιουργών. Όμως, ο συνεχώς αυξανόμενος όγκος ψηφιακής -πλέον- πληροφορίας, καθιστά την αναζήτηση σε αυτή ένα διόλου απλό εγχείρημα. Το να αναζητήσει κανείς μέσα στην αχανή στοίβα από άχυρα που είναι η πληροφορία, τη βελόνα που αντιπροσωπεύει το προσωπικό του γούστο αποτελεί το στοίχημα της εποχής σε ό,τι αφορά τη βελτίωση των υπηρεσιών που παρέχονται μέσω του παγκόσμιου ιστού.

2

Συστήματα πληροφοριακών συστάσεων και ψηφοφοριών

Έχοντας επιλύσει τα προβλήματα που σχετίζονται με τη χωρητικότητα και το κόστος αποθήκευσης, έχουμε περάσει στην εποχή όπου μεγάλη ποσότητα πληροφορίας έχει κατακλύσει το διαδίκτυο. Η έναρξη αυτής της νέας εποχής σηματοδότησε την αρχή μίας προσπάθειας επιβολής της τάξης στο χάος που έχει δημιουργηθεί. Αφενός μεν υπάρχουν χρήστες που γνωρίζουν τί είναι αυτό που ψάχνουν, αλλά δεν έχουν τη δυνατότητα να το εντοπίσουν και αφετέρου υπάρχουν χρήστες που δε γνωρίζουν τι ακριβώς πληροφορία αναζητούν. Η πρώτη κατηγορία χρηστών βρήκε απάντηση στο πρόβλημά της μέσα από τις μηχανές αναζήτησης (βλ. *Google, Bing, Yahoo*) ενώ η δεύτερη μέσα από μία διαφορετική κατηγορία υπηρεσιών, η οποία έχει τη δυνατότητα να τους συστήνει υποσύνολα της πληροφορίας στα οποία με μεγάλη πιθανότητα θα περιέχουν την πληροφορία που αναζητούν.

2.1 Συλλογικό φιλτράρισμα πληροφοριών

Ως συλλογικό φιλτράρισμα πληροφοριών (αγγλ. *Collaborative Filtering*) ορίζεται η μέθοδος δημιουργίας αυτοματοποιημένων προβλέψεων σχετικά με τα ενδιαφέροντα/προτιμήσεις ενός χρήστη βασισόμενη στη συλλογή πληροφοριών από πολλούς διαφορετικούς χρήστες¹.

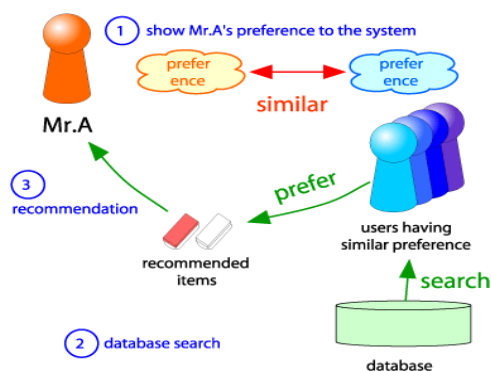
¹ Ορισμός: http://en.wikipedia.org/wiki/Collaborative_filtering

Ουσιαστικά, πρόκειται για μία μέθοδο η οποία, γνωρίζοντας το ιστορικό των προτιμήσεων ενός χρήστη, επιχειρεί να διαπιστώσει τί αναμένεται να ταιριάζει με τα ενδιαφέροντά του ώστε να του το συστήσει. Τέτοια συστήματα συστάσεων βασισμένα σε Collaborative Filtering έχουν ενσωματωθεί με μεγάλη επιτυχία σε πολλούς διαδικτυακούς τόπους. Υπηρεσίες όπως το *Amazon*, χρησιμοποιούν το συλλογικό φιλτράρισμα πληροφοριών για να προωθήσουν προϊόντα σε χρήστες που είναι πιθανό ότι τελικά θα τα αγοράσουν. Ιστοσελίδες όπως το *Last.fm* κρατούν το λεπτομερές ιστορικό των ακροάσεων/προτιμήσεων των χρηστών τους και εν συνεχεία, συγκρίνοντας τα ιστορικά διαφορετικών χρηστών εντοπίζουν μουσική που είναι πολύ πιθανό να αρέσει στους χρήστες. Άλλες υπηρεσίες όπως το *Digg*, κρατώντας στοιχεία για τα «*diggs*» των μελλών τους είναι σε θέση να συστήνουν νέα άρθρα στους αναγνώστες τα οποία με μεγάλη πιθανότητα θα διαβάσουν με ενδιαφέρον.

Παρότι όλα τα παραπάνω παραδείγματα αναφέρονται σε διαφορετικά είδη προϊόντων/πληροφορίας, η μέθοδος εξαγωγής προβλέψεων είναι η ίδια και βασίζεται στη συλλογή και διασταύρωση μεγάλου όγκου πληροφορίας από ένα αρκετά μεγάλο σύνολο χρηστών. Πιο συγκεκριμένα, διακρίνονται δύο βασικά στάδια λειτουργίας της μεθόδου:

1. Συλλογή δεδομένων που σχετίζονται με τις προτιμήσεις του χρήστη στον οποίο θα απευθυνθεί η σύσταση
2. Σύγκριση των παραπάνω δεδομένων με αντίστοιχα δεδομένα που έχουν συγκεντρωθεί για χρήστες με παραπλήσια ενδιαφέροντα

Η διαδικασία αυτή περιγράφεται στην Εικόνα 2-1.



Εικόνα 2-1: Collaborative filtering process

Εξαιτίας του τρόπου λειτουργίας της μεθόδου παρατηρούνται δύο αξιοσημείωτα φαινόμενα:

1. Cold Start
2. Positive Feedback

Το πρώτο φαινόμενο θεωρείται ένα από τα βασικότερα προβλήματα που αντιμετωπίζει η προσέγγιση του Collaborative Filtering και σχετίζεται με την ανεπάρκεια στατιστικών στοιχείων κατά την έναρξη λειτουργίας ενός τέτοιου συστήματος. Ακριβώς επειδή το

συλλογικό φιλτράρισμα απαιτεί τη συλλογή μεγάλου όγκου πληροφοριών, μέχρι να μπορέσει το σύστημα να συλλέξει αυτές τις πληροφορίες, τα συμπεράσματα και οι συστάσεις που θα εξάγει δε θα θεωρούνται, και εν πολλοίς δεν θα είναι, αξιόπιστες. Παρότι έχουν προταθεί λύσεις που ξεφεύγουν από τα πλαίσια του συλλογικού φιλτράρισματος πληροφοριών, το cold start φαινόμενο θεωρείται από πολλούς ένα αναγκαίο κακό που μπορεί απλά να παραβλεφθεί κατά τα πρώτα στάδια λειτουργίας ενός συστήματος.

Για να γίνει κατανοητό το φαινόμενο της θετικής ανάδρασης, ας θεωρήσουμε το παράδειγμα του Amazon το οποίο, έχοντας υλοποιήσει ένα μηχανισμό Collaborative Filtering, έχει τη δυνατότητα να προτείνει σε ένα χρήστη που μόλις αναζήτησε ένα βιβλίο, διαφορετικά βιβλία τα οποία αναμένεται ότι επίσης θα αγοράσει. Η εξαγωγή αυτής της πρότασης βασίζεται σε ένα μοτίβο της μορφής «*οι χρήστες που αγόρασαν το βιβλίο x, αγόρασαν επίσης και το βιβλίο y*». Κατά συνέπεια, στο χρήστη που προτίθεται να αγοράσει το βιβλίο x, θα συσταθεί από το μηχανισμό η αγορά του βιβλίου y. Εάν αυτή η σύσταση αποδειχθεί εύστοχη και ο χρήστης πράγματι αγοράσει τα βιβλία x και y μαζί, τότε ενισχύεται ακόμα περισσότερο η συσχέτιση «*οι χρήστες που αγόρασαν το βιβλίο x, αγόρασαν επίσης και το βιβλίο y*» με αποτέλεσμα το y να προταθεί με ακόμα μεγαλύτερη πιθανότητα στους επόμενους χρήστες που θα αγοράσουν μελλοντικά το βιβλίο x. Συνοπτικά, συσχετίσεις οι οποίες, εκ του αποτελέσματος, αποδεικνύονται εύστοχες επανεμφανίζονται με μεγαλύτερη πιθανότητα στο μέλλον.

2.2 Συστάσεις μουσικής

Ένας από τους τομείς που έχει επωφεληθεί πολύ από την ύπαρξη των συστημάτων πληροφοριακών συστάσεων είναι ο τομέας της μουσικής. Υπηρεσίες που έχουν τη δυνατότητα να προτείνουν στους χρήστες νέα τραγούδια και μουσικά σχήματα διαδέχονται η μία την άλλη και πολλές από αυτές μάλιστα χαίρουν ήδη ευρείας αποδοχής. Υπηρεσίες όπως οι *Genius* και *Ping* της Apple που επεκτείνουν τη λειτουργικότητα του *iTunes*² καθώς και web εφαρμογές όπως το *Last.fm*³, το *Pandora.com*⁴ και το *Slacker.com*⁵ έχουν τη δυνατότητα εντοπίζουν τραγούδια που ταιριάζουν μεταξύ τους αλλά και με τις προτιμήσεις των χρηστών και να δημιουργούν κατάλληλες λίστες αναπαραγωγής. Στον αντίποδα, εφαρμογές όπως το *Future Of Music 2010*⁶, έχουν τη δυνατότητα να εντοπίζουν μουσική που δεν ταιριάζει με τις προτιμήσεις του χρήστη και να τη διαγράφουν αυτόματα από τον υπολογιστή του εξοικονομώντας χώρο.

Όλες οι παραπάνω υπηρεσίες χρησιμοποιούν αλγορίθμους Collaborative Filtering και κατά συνέπεια διατηρούν λεπτομερές ιστορικό σχετικά με τις προτιμήσεις των ακροατών τους. Έτσι είναι σε θέση να εξάγουν μοτίβα της μορφής «*οι χρήστες που ακούν τον καλλιτέχνη x, με μεγάλη πιθανότητα ακούν επίσης τον καλλιτέχνη y*» ή «*οι χρήστες που ακούν τον κομμάτι x, με*

μεγάλη πιθανότητα θα ακούσουν στη συνέχεια το κομμάτι γ». Υπάρχουν όμως και υπηρεσίες που καταλήγουν σε συμπεράσματα, σχετικά με το τί αρέσει στους χρήστες, βασιζόμενες σε συστήματα ψηφοφοριών. Παραδείγματα τέτοιων υπηρεσιών αποτελούν τα *The Hype Machine*⁷ και *Mixpod*⁸, τα οποία κατασκευάζουν λίστες αναπαραγωγής βασιζόμενες στις θετικές/αρνητικές ψήφους των χρηστών. Χαρακτηριστικό πλεονέκτημα μεθόδων που βασίζονται σε συστήματα ψηφοφορίας είναι ο περιορισμός του cold start φαινομένου, καθώς πλέον δε συγκρίνονται προτιμήσεις διαφορετικών χρηστών, αλλά λαμβάνονται υπόψη ψήφοι που τελικά αντιστοιχίζονται σε δημοτικότητα. Κατά συνέπεια, τέτοια συστήματα είναι αρκετά πιθανό ότι θα εξάγουν τα ίδια αποτελέσματα ακόμα και για πιο μικρά ακροατήρια αλλά και στα πρώτα στάδια της λειτουργίας τους.

Σε κάθε περίπτωση όμως, για να μπορέσουν τα παραπάνω συστήματα να εξάγουν αξιόπιστα συμπεράσματα και να καταλήξουν σε εύστοχες συστάσεις απαιτείται συλλογή στατιστικών στοιχείων αναφορικά με τις συνήθειες των χρηστών. Ακόμα και σε αυτήν την περίπτωση όμως, είναι πολλοί εκείνοι που αμφισβητούν τα αποτελέσματα των συστάσεων που βασίζονται αποκλειστικά στο ιστορικό και στις συνήθειες των χρηστών.

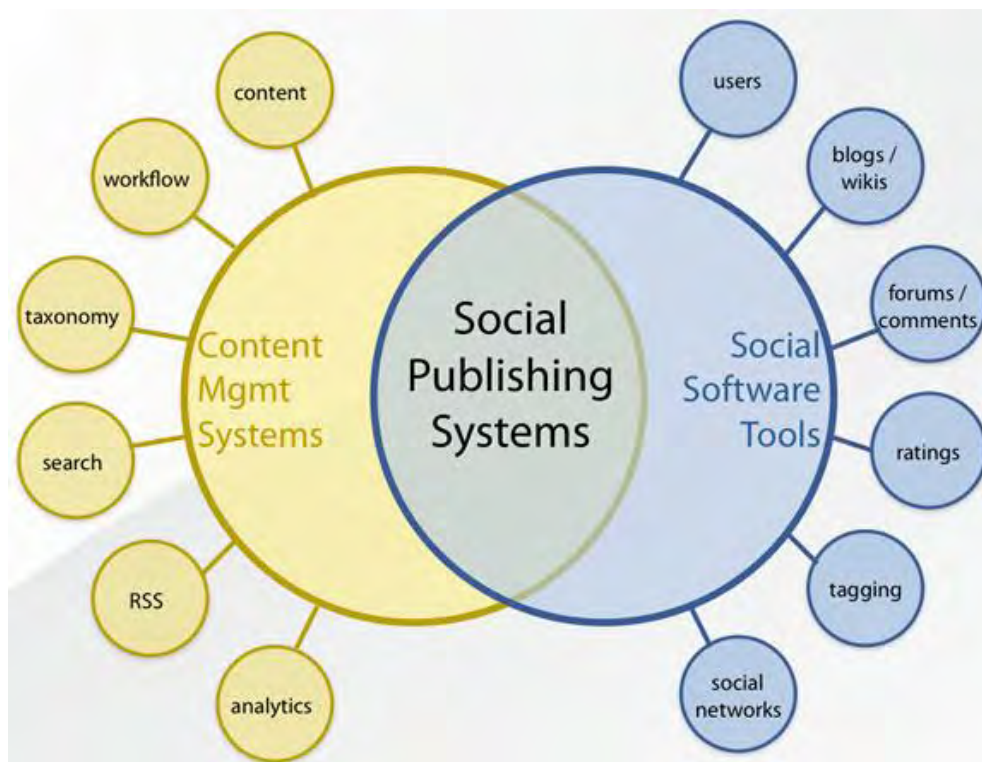
Ο Alex Williams, διαχειριστής των blog *ReadWriteEnterprise*⁹, *ReadWriteCloud*¹⁰ και αρθρογράφος στο *ReadWriteWeb*¹¹, σε άρθρο του σχετικό με τα συστήματα συστάσεων μουσικού περιεχομένου στο blog *ReadWriteWeb*¹², θέτει υπό αμφισβήτηση την αξία των συστάσεων που έχουν προέλθει από την επεξεργασία τέτοιων στατιστικών στοιχείων. Σύμφωνα πάντα με το άρθρο, στόχος μίας σύστασης μουσικής δεν είναι η επιβεβαίωση της ομοιότητας δύο γνωστών καλλιτεχνών (σχεδόν όλοι γνωρίζουν πως η μουσική των Beatles και του John Lennon είναι συγγενική), αλλά η ανακάλυψη καλλιτεχνών -μέσα από το φάσμα του long tail- για την ύπαρξη των οποίων οι χρήστες δύσκολα θα ενημερώνονταν από άλλα μέσα. Με άλλα λόγια, ο πραγματικός στόχος ενός αλγορίθμου εξαγωγής συστάσεων μουσικής θα πρέπει να είναι η σύνδεση των λιγότερο γνωστών καλλιτεχνών με το -έστω και μικρό- ακροατήριό τους.

2.3 Ιστοχώροι κοινωνικής δημοσίευσης

Ανεξαρτήτως του μηχανισμού συστάσεων μουσικού -και όχι μόνο- περιεχομένου που μπορεί να χρησιμοποιεί μία υπηρεσία, στην περίπτωση που υλοποιείται ως web εφαρμογή, είναι απαραίτητη η ύπαρξη μίας υποδομής που θα φιλοξενεί το περιεχόμενο και ταυτόχρονα θα δίνει τη δυνατότητα στους χρήστες να εκφράζουν τις προτιμήσεις τους.

Οι ιστοχώροι κοινωνικής δημοσίευσης (αγγλ. *social publishing websites*) είναι τέτοιου είδους πλατφόρμες, οι οποίες παρέχουν στους χρήστες τη δυνατότητα να διαχειρίζονται το

περιεχομένό τους και ταυτόχρονα να εκφράζουν τα σχόλια και την κριτική τους απέναντι σε αυτό. Πρόκειται για ιστοχώρους που συνδυάζουν χαρακτηριστικά συστημάτων διαχείρισης περιεχομένου (*content management systems*) και εργαλείων κοινωνικής δικτύωσης (*social software tools*).



Εικόνα 2-2: Social publishing traits

Όπως εύστοχα αναφέρει ο Brian Frederick, βοηθός καθηγητή του τμήματος Πολιτικών Επιστημών του Πανεπιστημίου *Bridgewater*¹³, σε άρθρο του², το σημαντικότερο χαρακτηριστικό του social publishing είναι το γεγονός ότι δημιουργεί ένα άμεσο κανάλι επικοινωνίας μεταξύ εκδότη/δημιουργού και αναγνώστη/αποδέκτη μέσα από το οποίο ο τελευταίος μπορεί να μεταφέρει απευθείας στο δημιουργό τις απόψεις και τις ερωτήσεις του οι οποίες πλέον γίνονται αντικείμενο διαλόγου μεταξύ των δύο πλευρών.

Αυτή ακριβώς η δυνατότητα της ανάδρασης (*feedback*) αποδεικνύεται ιδιαίτερα χρήσιμη όταν στόχος είναι το φιλτράρισμα και η οργάνωση του περιεχομένου μίας πλατφόρμας. Δημιουργοί περιεχομένου έχουν τη δυνατότητα να ανεβάζουν το περιεχόμενο που επιθυμούν, ενώ οι χρήστες μπορούν να ανταλλάσσουν σχόλια πάνω σε αυτό τόσο μεταξύ τους όσο και με τους δημιουργούς. Επιπλέον, μπορούν να συνεισφέρουν στην αξιολόγησή του ψηφίζοντας το θετικά ή αρνητικά, αλλά και στην οργάνωσή του αποδίδοντάς του ετικέτες (*tags*) οι οποίες βοηθούν στην κατηγοριοποίησή του και διευκολύνουν την αναζήτηση.

² <http://billboardsandcavedrawings.wordpress.com/2009/10/28/what-the-hell-is-social-publishing/>

Στόχος της εργασίας μας είναι η υλοποίηση μίας τέτοιας πλατφόρμας, τα χαρακτηριστικά και οι προδιαγραφές της οποίας παρουσιάζονται παρακάτω.

3

Στόχοι και προδιαγραφές

Στο κεφάλαιο αυτό θα ασχοληθούμε με τους στόχους και τις προδιαγραφές επί τη βάση των οποίων έχει αναπτυχθεί η διαδικτυακή εφαρμογή.

3.1 Συλλογική ανάδειξη νέων καλλιτεχνών

Κυρίαρχος στόχος της εφαρμογής μας είναι η ανάδειξη και η προώθηση της δισκογραφικής δουλειάς νέων και αφανών καλλιτεχνών μέσω ραδιοφωνικών σταθμών που δημιουργούνται ανάλογα με το είδος μουσικής και μέσω μίας συνεχόμενης ψηφοφορίας στην οποία συμμετέχουν οι εγγεγραμμένοι στο σύστημα χρήστες.

Το γεγονός ότι απευθυνόμαστε κυρίως στους καλλιτέχνες που ανήκουν στο long tail της μουσικής σκηνής μας οδήγησε στο συμπέρασμα πως το κοινό το οποίο αναμένεται να δείξει ενδιαφέρον θα είναι περιορισμένο. Αυτό μας ώθησε στην απόφαση να απορρίψουμε το ενδεχόμενο του συλλογικού φιλτραρίσματος πληροφοριών διότι για να προκύψουν αξιόπιστα αποτελέσματα, απαιτείται μεγάλος όγκος πληροφοριών ο οποίος προέρχεται από τις επιλογές των χρηστών. Στην προκειμένη περίπτωση το φαινόμενο cold start δε θα μπορούσε να εξαλειφθεί αφού το σύστημα θα βασίζεται μονίμως σε σχετικά περιορισμένο αριθμό χρηστών. Συνεπώς, καταλήξαμε στη χρήση ενός συστήματος ψηφοφορίας, το οποίο εξάγει καθολικά συμπεράσματα για τις προτιμήσεις των χρηστών, χωρίς αυτά να εξαρτώνται από τον αριθμό των χρηστών.

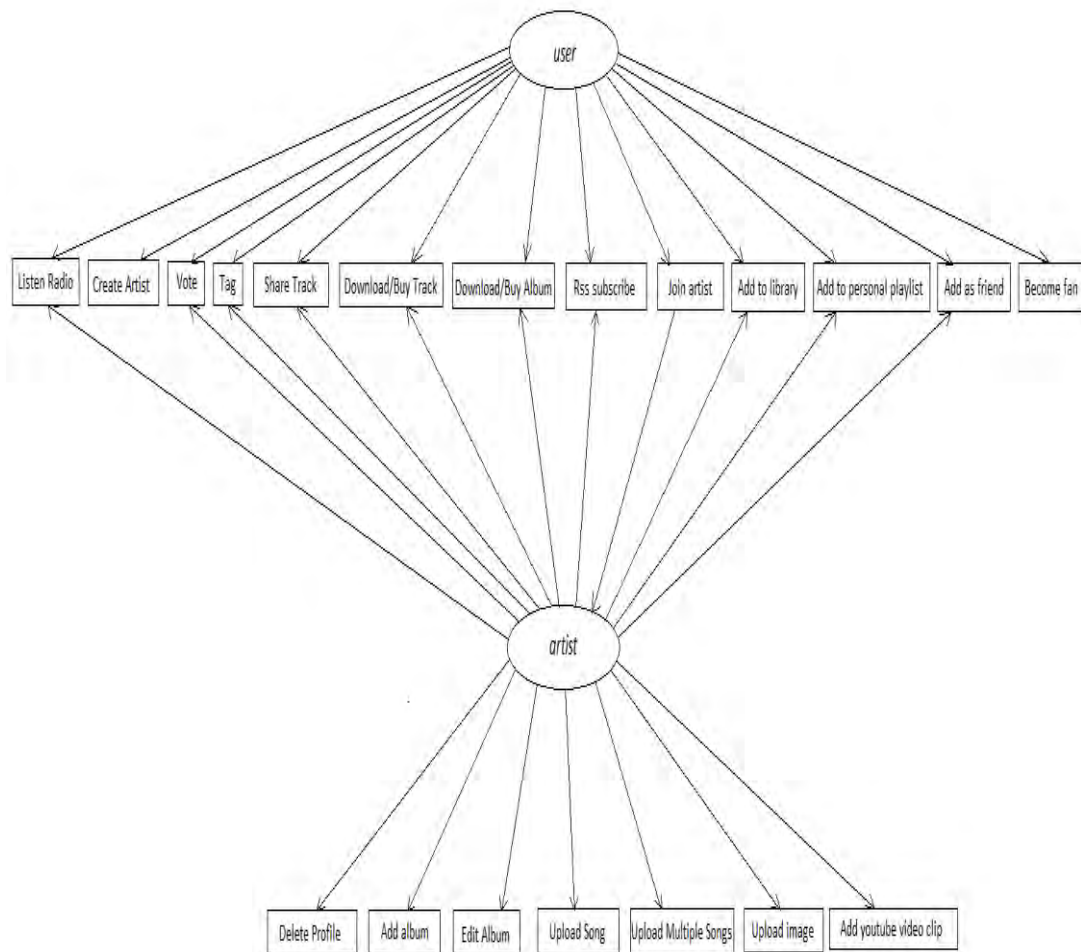
Κάθε χρήστης, που έχει συνδεθεί στην εφαρμογή, μπορεί να ακούσει τα τραγούδια των καλλιτεχνών τα οποία είναι κατηγοριοποιημένα σε μουσικές λίστες ανάλογα με το είδος τους (σταθμούς). Όταν οι χρήστες επιλέξουν έναν σταθμό από την κατηγορία της αρεσκείας τους (ο οποίος είναι ίδιος για όλους), ακούνε μια λίστα αναπαραγωγής με τραγούδια τα οποία έχουν τη δυνατότητα να ψηφίσουν θετικά ή αρνητικά. Οι ψήφοι των χρηστών, αποτελούν τον κύριο γνώμονα με τον οποίο η εφαρμογή αναπροσαρμόζει τη λίστα αναπαραγωγής του κάθε σταθμού ανά τακτά χρονικά διαστήματα, επαναλαμβάνοντας πιο συχνά τα τραγούδια που συγκεντρώνουν θετικές ψήφους. Μπορούμε να φανταστούμε τα τραγούδια σαν ένα μεγάλο chart όπου ανακατατάσσονται δυναμικά ανάλογα με τις ψήφους των ακροατών. Αυτό ισχύει και για τα άλμπουμ και τους καλλιτέχνες. Ως αποτέλεσμα, οι καλλιτέχνες οι οποίοι παρατηρείται ότι «ενθουσιάζουν» τους χρήστες της εφαρμογής με τα τραγούδια τους, αναδεικνύονται και κερδίζουν δημοφιλία διότι τα κομμάτια τους αποκτούν μεγαλύτερη βαρύτητα (credits) και παίζονται πιο συχνά στα ραδιόφωνα. Με αυτόν τον τρόπο λοιπόν επιτυγχάνεται σε πρώτη φάση η ανάδειξη και η προώθηση των καλλιτεχνών που «αρέσουν» στους χρήστες. Σε αυτό το σημείο παρατηρείται το φαινόμενο της *θετικής ανάδρασης*. Τα κομμάτια με υψηλά credits παίζονται πιο συχνά στις λίστες των σταθμών και ακούγονται περισσότερο. Συνεπώς, είναι πολύ πιθανό να συγκεντρώσουν ακόμη περισσότερες θετικές ψήφους και να ακούγονται ακόμη πιο πολλές φορές στα ραδιόφωνα. Αυτό είναι το positive feedback το οποίο βοηθά τους καλλιτέχνες που έχουν ήδη καταφέρει να ξεχωρίσουν, να αναδειχθούν ακόμη περισσότερο ανάμεσα στους υπόλοιπους (the rich get richer). Οι λίστες αναπαραγωγής λοιπόν σε συνδυασμό με την ψηφοφορία παίζουν καταλυτικό ρόλο στην προώθηση των καλλιτεχνών. Σημαντικό ρόλο όμως παίζουν και οι πίνακες top charts στους οποίους εμφανίζονται, ανά είδος, οι δέκα πρώτοι καλλιτέχνες, τα δέκα πρώτα άλμπουμ και τα δέκα πρώτα τραγούδια που φαίνονται να είναι πιο δημοφιλή. Οι πίνακες αυτοί ανανεώνονται κάθε φορά που η εφαρμογή συγκεντρώνει τις ψήφους των χρηστών.

Εξετάζοντας τώρα την εφαρμογή από την σκοπιά της μουσικής βιομηχανίας, αυτό που μπορούμε να πούμε είναι ότι, βασιζόμενοι στο long tail των καλλιτεχνών που υπάρχουν αυτή τη στιγμή στη χώρα μας, δίνουμε τη δυνατότητα να υπάρξει μια «άμεση διασύνδεση» μεταξύ ακροατή και καλλιτέχνη. Ο καλλιτέχνης μοιράζεται, εύκολα και γρήγορα, τη μουσική του με τον κόσμο ο οποίος με τη σειρά του βρίσκει το είδος που του αρέσει και αναδεικνύει πλειοψηφικά τα κομμάτια και τους καλλιτέχνες της αρεσκείας του. Συμπερασματικά, το web application που αναπτύχθηκε μπορεί να δώσει μια «ανάσα» στο μεγάλο κομμάτι των καλλιτεχνών του long tail, οι οποίοι αντιμετωπίζουν ολοένα και περισσότερα εμπόδια στην ανάδειξη τους στο χώρο, δημιουργώντας νέες ευκαιρίες για την προώθησή τους σε μια εποχή όπου η δισκογραφία περνάει τεράστια κρίση.

3.2 Αυτόματη οργάνωση και διαχείριση μουσικού

περιεχομένου

Στο σύστημα που αναπτύξαμε κάθε επίδοξος χρήστης μπορεί να εγγραφεί είτε ως user είτε ως artist. Ο user είναι ο χρήστης που μπαίνει στο σύστημα συνήθως με την ιδιότητα του ακροατή ο οποίος μπορεί να ακούσει μουσική, να βλέπει πληροφορίες για νέους καλλιτέχνες και να ανακαλύπτει νέα κομμάτια ανάλογα το είδος μουσικής που του αρέσει. Ο artist είναι ο νέος καλλιτέχνης που θέλει να προωθήσει τη δισκογραφική του δουλειά μέσω της διαδικτυακής αυτής πλατφόρμας και μπορεί να δημιουργήσει άλμπουμ, να ανεβάσει τραγούδια, να τα επεξεργαστεί και να προσθέσει το φωτογραφικό του υλικό. Στο σχήμα που ακολουθεί φαίνονται αναλυτικά οι ενέργειες της κάθε κατηγορίας χρήστη.



Εικόνα 3-1: Σχηματική αναπαράσταση των ενεργειών κάθε χρήστη

Πιο αναλυτικά, ο χρήστης που εισέρχεται στο σύστημα ως user μπορεί να εκτελέσει τις παρακάτω ενέργειες :

Listen radio: Ο user έχει τη δυνατότητα να επιλέξει για να ακούσει έναν σταθμό από αυτούς που έχουν δημιουργηθεί. Οι σταθμοί δημιουργούνται από τη μουσική που ανεβάζουν οι καλλιτέχνες στην εφαρμογή κατηγοριοποιώντας την ανάλογα με το είδος της. Εκτός όμως από τους σταθμούς, ο χρήστης μπορεί να δει τα άλμπουμ οποιουδήποτε καλλιτέχνη, να ακούσει όλα τα τραγούδια που περιλαμβάνονται σε αυτό (listen to album) ή να ακούσει ένα συγκεκριμένο κομμάτι είτε μέσω της σελίδας του άλμπουμ είτε μέσω της σελίδας του συγκεκριμένου τραγουδιού.

Create Artist: Με αυτήν τη λειτουργία ο user μπορεί να φτιάξει το καλλιτεχνικό του προφίλ στο σύστημα. Συμπληρώνοντας τα απαραίτητα στοιχεία όπως για παράδειγμα το καλλιτεχνικό όνομα και τα είδη μουσικής που τον χαρακτηρίζουν μπορεί εύκολα να δημιουργήσει την καλλιτεχνική του σελίδα.

Vote: Ο user έχει την επιλογή να ψηφίσει θετικά ή αρνητικά οποιοδήποτε κομμάτι. Μπορεί να ψηφίσει το τραγούδι είτε όταν το ακούει σε ένα σταθμό είτε βρίσκοντας τη σελίδα του αντίστοιχου καλλιτέχνη και το αντίστοιχο άλμπουμ στο οποίο εμπεριέχεται.

Tag: Η λειτουργία αυτή δίνει τη δυνατότητα στο χρήστη να προσθέσει μία ετικέτα, ως προς το είδος της μουσικής, στους καλλιτέχνες, στα άλμπουμ και στα τραγούδια τους. Μπορεί να δώσει όποιον χαρακτηρισμό θέλει σε κάθε ένα από τα παραπάνω. Ο χρήστης ακούγοντας ένα σταθμό μπορεί να χαρακτηρίσει μόνο το κομμάτι που ακούει εκείνη τη στιγμή ενώ από τη σελίδα του καλλιτέχνη μπορεί να βάλει ετικέτες στον καλλιτέχνη στα άλμπουμ του αλλά και στα τραγούδια του.

Share track: Ο χρήστης μπορεί να μοιραστεί με τους φίλους του από κάποια κοινωνικά δίκτυα τη σελίδα του τραγουδιού, του άλμπουμ ή του καλλιτέχνη που επιθυμεί.

Download/Buy Track ή Download/Buy Album: Προσφέροντας στο χρήστη αυτή τη λειτουργία, του δίνεται η δυνατότητα να κατεβάσει ή να αγοράσει ένα τραγούδι ή ένα άλμπουμ. Ο καλλιτέχνης είναι αυτός που ορίζει το κομμάτι ή το άλμπουμ ως «Ελεύθερο»(δωρεάν) ή «Επί Πληρωμή»(με πληρωμή). Στη δεύτερη περίπτωση ο χρήστης πρέπει να το αγοράσει. Έτσι ανακατευθύνεται στην PayPal¹⁴ για να δώσει το ποσό που έχει ορίσει ο καλλιτέχνης. Αφού γίνει η ηλεκτρονική αγορά, ο χρήστης μπορεί να κατεβάσει ελεύθερα το τραγούδι ή ολόκληρο το άλμπουμ σε μορφή zip στον υπολογιστή του.

Rss subscribe: Με την εγγραφή στα rss feeds³, ο χρήστης μπορεί να ενημερώνεται μέσω του φυλλομετρητή του για τα δέκα τελευταία τραγούδια, άλμπουμ και καλλιτέχνες που

³ [http:// en.wikipedia.org/wiki/RSS/](http://en.wikipedia.org/wiki/RSS/)

μπήκαν πρόσφατα στο σύστημα. Επίσης, έχει τη δυνατότητα να μαθαίνει ποια είναι τα δέκα πιο δημοφιλή τραγούδια, μαζί με τα άλμπουμ και τους καλλιτέχνες, που αναδεικνύονται από τα αποτελέσματα της ψηφοφορίας των χρηστών.

Join artist: Η λειτουργία αυτή αξιοποιείται όταν ο χρήστης αποτελεί μέλος ενός καλλιτέχνη που υπάρχει ήδη στο σύστημα. Με αυτόν τον τρόπο όλα τα μέλη τα οποία συνθέτουν ένα συγκρότημα και επιθυμούν να έχουν δικαίωμα πρόσβασης και επεξεργασίας στη σελίδα του συγκροτήματος, μπορούν με αυτή τη δυνατότητα να στείλουν αίτημα συμμετοχής στον καλλιτέχνη (δηλαδή στο χρήστη που έφτιαξε το προφίλ του συγκροτήματος) και αυτός να τους αποδεχτεί. Έτσι, τα μέλη του συγκροτήματος θα αποτελούν απλούς χρήστες (user) αλλά με τα δικαιώματα που έχει και ο δημιουργός της σελίδας του γκρουπ στο σύστημα.

Add to library: Με αυτή τη δυνατότητα, ο χρήστης έχει στη διάθεση του μια «βιβλιοθήκη» στην οποία μπορεί να αποθηκεύσει:

Οποιοδήποτε ραδιοφωνικό σταθμό παράγει η εφαρμογή

Οποιαδήποτε λίστα έχει παραχθεί, έχοντας κάνει αναζήτηση ο χρήστης για κάποιο είδος μουσικής, καλλιτέχνη ή τραγούδι.

Έτσι, με αυτόν τον τρόπο, ο χρήστης μπορεί εύκολα και γρήγορα να βρεί τις αγαπημένες του μουσικές λίστες χωρίς να χρειαστεί να ξανακάνει την αναζήτηση που είχε κάνει αρχικά.

Add to personal playlist: Ο χρήστης μπορεί να δημιουργήσει προσωπικές μουσικές λίστες. Οποιοδήποτε τραγούδι ακούσει ο χρήστης, ή προσπελάσει μέσα από την σελίδα του εκάστοτε καλλιτέχνη, έχει τη δυνατότητα να το αποθηκεύσει σε μία ή περισσότερες προσωπικές μουσικές λίστες. Έτσι, ο ακροατής διατηρεί στο προφίλ του τα αγαπημένα του τραγούδια, οργανωμένα στις λίστες που ο ίδιος έχει δημιουργήσει. Μπορεί λοιπόν ανά πάσα στιγμή να ακούσει όποια προσωπική μουσική λίστα θελήσει, εύκολα και γρήγορα.

Add as friend: Ο χρήστης μπορεί να προσθέσει ως φίλο κάποιον άλλον χρήστη της εφαρμογής. Ο χρήστης που δέχεται το αίτημα φιλίας θα πρέπει να επιβεβαιώσει ότι θέλει να προσθέσει τον συγκεκριμένο χρήστη σαν φίλο του. Με αυτόν τον τρόπο, ο χρήστης έχει τη δυνατότητα να δει το προφίλ, τη βιβλιοθήκη, τις προσωπικές μουσικές λίστες και τους καλλιτέχνες που θαυμάζουν οι φίλοι του.

Become fan: Με αυτή τη δυνατότητα, ο απλός χρήστης μπορεί να γίνει θαυμαστής ενός η περισσότερων καλλιτεχνών που θα βρει στην εφαρμογή μας.

Όσον αφορά στον καλλιτέχνη που εισέρχεται στο σύστημα, μπορούμε να αναλύσουμε εκτενέστερα τις λειτουργίες τις οποίες έχει στη διάθεσή του. Πριν γίνει όμως αυτό θα πρέπει να επισημανθεί ότι μπορεί και αυτός να λειτουργήσει εν μέρει ως απλός χρήστης. Αυτό σημαίνει ότι πολλές από τις ενέργειες που μπορεί να εκτελέσει ο χρήστης μπορεί να τις

εκτελέσει και αυτός. Κατά συνέπεια, οι λειτουργίες Listen radio, Vote, Tag, Share track, Download/Buy Track, Download/Buy Album, Rss subscribe, Add to library, Add to personal playlist, Add as friend είναι διαθέσιμες και για τον καλλιτέχνη που χρησιμοποιεί την εφαρμογή. Εκτός όμως από αυτά μπορεί να διαχειριστεί την καλλιτεχνική του σελίδα με τις εξής ενέργειες:

Delete profile: Ο καλλιτέχνης έχει τη δυνατότητα να διαγράψει το προφίλ του οποιαδήποτε στιγμή θέλει. Όταν συμβεί κάτι τέτοιο, τότε διαγράφονται όλα τα άλμπουμ, τα τραγούδια και οι εικόνες που έχει ανεβάσει ο καλλιτέχνης στην εφαρμογή και δε μένει καμία πληροφορία για αυτόν στο σύστημα.

Add album: Αρχικά, όταν ο καλλιτέχνης καταχωρήσει το προφίλ του, δημιουργείται αυτόματα ένα άλμπουμ (dummy άλμπουμ) στο οποίο μπορεί να ανεβάσει τραγούδια τα οποία δεν ανήκουν σε συγκεκριμένο δίσκο. Εάν ο καλλιτέχνης θελήσει να φτιάξει ένα νέο δίσκο τότε επιλέγοντας «Add album» μπορεί να δημιουργήσει το νέο άλμπουμ δίνοντας τις σχετικές πληροφορίες για αυτό.

Edit album: Ο καλλιτέχνης έχει οποιαδήποτε στιγμή τη δυνατότητα να επεξεργαστεί ό,τι έχει σχέση με ένα άλμπουμ το οποίο έχει δημιουργήσει. Συνεπώς, μπορεί εάν θέλει να το διαγράψει ολόκληρο ή να επιλέξει συγκεκριμένα κομμάτια τα οποία θέλει να διαγραφούν. Ο καλλιτέχνης επίσης, έχει την επιλογή να αλλάξει το εξώφυλλο της συλλογής και να επεξεργαστεί ένα προς ένα τα τραγούδια που περιέχονται σε αυτήν. Μπορεί δηλαδή να αλλάξει τον τίτλο αλλά και τον αριθμό σειράς του κάθε κομματιού. Τέλος, έχει τη δυνατότητα να ορίσει αν το κομμάτι είναι «Ελεύθερο» ή «Επί Πληρωμή» το οποίο καθορίζει εάν ο χρήστης θα μπορεί να «κατεβάσει» ελεύθερα το τραγούδι ή εάν θα πρέπει να πληρώσει.

Upload song: Με αυτήν την ενέργεια ο καλλιτέχνης επιλέγει να ανεβάσει ένα κομμάτι στην εφαρμογή σε μορφή mp3 το οποίο είτε ανήκει στο dummy άλμπουμ είτε σε κάποιο άλλο άλμπουμ το οποίο έχει δημιουργήσει ο ίδιος ο καλλιτέχνης. Αφού ολοκληρωθεί η φόρτωση του τραγουδιού, η εφαρμογή διαβάζει τα μεταδεδομένα (metadata) του τραγουδιού και επιστρέφεται μία φόρμα όπου γίνεται η επιβεβαίωση των στοιχείων. Σε αυτό το σημείο ορίζεται ο τίτλος του κομματιού, ο αριθμός σειράς του στο άλμπουμ, το είδος ή είδη μουσικής στα οποία μπορεί να ανήκει και αν το τραγούδι είναι «Ελεύθερο» ή «Επί Πληρωμή».

Upload multiple songs: Ο καλλιτέχνης, σ' αυτή την περίπτωση, μπορεί να φορτώσει τα τραγούδια ενός άλμπουμ μαζικά στην εφαρμογή. Επιλέγει ποια είναι τα κομμάτια που θέλει να ανεβάσει και η εφαρμογή τα διαβάζει ένα-ένα. Σε αυτή την περίπτωση η πληροφορία για τα τραγούδια διαβάζεται αποκλειστικά από τα μεταδεδομένα τους εφόσον δεν επιστρέφεται πίσω κάποια φόρμα για να επιβεβαιωθούν τα στοιχεία. Επομένως θα ήταν φρόνιμο, τα κομμάτια που ανεβάζουν οι καλλιτέχνες μαζικά να έχουν φροντίσει να

συμπληρώσουν τα βασικά στοιχεία όπως ο τίτλος, το όνομα του καλλιτέχνη και το είδος μουσικής στο οποίο ανήκει.

Upload image: Με αυτή τη λειτουργία ο καλλιτέχνης έχει την ευκαιρία να ανεβάσει το φωτογραφικό του υλικό και να δημιουργηθεί αυτόματα μία έκθεση φωτογραφίας όπου θα είναι συγκεντρωμένες όλες οι φωτογραφίες.

Add YouTube video clip: Ο καλλιτέχνης μπορεί να προσθέσει στη σελίδα των τραγουδιών του, το αντίστοιχο video clip από κάθε τραγούδι, το οποίο υπάρχει στο youtube. Έτσι, όταν ο χρήστης μπει στη σελίδα του τραγουδιού στο οποίο ο καλλιτέχνης έχει βάλει το αντίστοιχο video, έχει τη δυνατότητα να παρακολουθήσει το video του τραγουδιού από το youtube, ενσωματωμένο στη σελίδα της εφαρμογής.

4

Αλγόριθμοι

4.1 Σύστημα ψηφοφορίας

Σε κάθε κομμάτι που εισέρχεται στο σύστημα εκχωρείται μία αρχική πίστωση (*credit*). Η πίστωση αυτή χρησιμοποιείται ως μέτρο δημοτικότητας του κάθε κομματιού και χρησιμοποιείται από τον αλγόριθμο κατασκευής λίστας εκτέλεσης. Αρχικά, για τα N κομμάτια που βρίσκονται στο σύστημά μας, έστω C η πίστωση που αντιστοιχεί στο κάθε ένα. Συνολικά δηλαδή στο σύστημά μας έχει ανατεθεί πίστωση ίση με $N \cdot C$. Η αρχική πίστωση C αποτελεί παράμετρος του αλγορίθμου.

Για κάθε νέο κομμάτι που εισέρχεται στο σύστημα, θέλουμε να του ανατίθεται τόση πίστωση, ώστε ο αλγόριθμος κατασκευής λίστας εκτέλεσης είτε να προτιμά το νέο κομμάτι εις βάρος όσων βρίσκονται ήδη στο σύστημα, είτε το αντίστροφο. Μία πολιτική επιλογής πίστωσης εισόδου θα μπορούσε να λαμβάνει υπόψη το μέσο όρο των πιστώσεων για όλα τα κομμάτια που βρίσκονται στο σύστημα, έστω $E(C)$:

$$C[\text{νέο κομμάτι}] = \alpha \cdot E(C)$$

όπου α παράμετρος του αλγορίθμου με $\alpha > 0$. Η παράμετρος α , μας βοηθά να ρυθμίζουμε την προτεραιότητα που θα δίνει ο αλγόριθμος κατασκευής λίστας εκτέλεσης στα νέα κομμάτια.

Εφόσον το σύστημα που σχεδιάζουμε θέλουμε να είναι ανοιχτό, ο συνολικός αριθμός πίστωσης του συστήματος θα μεταβάλλεται κάθε φορά που ένα κομμάτι εισέρχεται στο σύστημα ή εξέρχεται από αυτό. Συγκεκριμένα, θέλουμε με την εξαγωγή ενός κομματιού από

το σύστημα, η συνολική πίστωση του συστήματος να επανέρχεται στο επίπεδο που βρισκόταν πριν από την εισαγωγή του κομματιού. Για να επιτευχθεί αυτός ο στόχος πρέπει να είμαστε σε θέση να χειριζόμαστε τις παρακάτω τρεις περιπτώσεις όπου:

1. Το κομμάτι ακριβώς πριν εξέλθει από το σύστημα διαθέτει ίση πίστωση με την αρχική πίστωση που του ανατέθηκε, **Center = Cexit**
2. Το κομμάτι ακριβώς πριν εξέλθει από το σύστημα διαθέτει περισσότερη πίστωση από την αρχική πίστωση που του ανατέθηκε, **Center < Cexit**
3. Το κομμάτι ακριβώς πριν εξέλθει από το σύστημα διαθέτει λιγότερη πίστωση από την αρχική πίστωση που του ανατέθηκε, **Center > Cexit**

Για παράδειγμα, έστω κομμάτι που εισέρχεται στο σύστημα με αρχική πίστωση $Center = 10$. Έστω, επίσης, ότι πριν την εξαγωγή του από το σύστημα η πίστωση που διαθέτει είναι $Cexit = 5$. Στην περίπτωση αυτή, με τη διαγραφή του κομματιού, το σύστημα χάνει αυτόματα πίστωση ίση με $Cexit$. Για να επανέλθει όμως στην κατάσταση που βρισκόταν πριν την εισαγωγή του κομματιού, πρέπει να αφαιρεθεί συνολικά από το σύστημα επιπλέον πίστωση ίση με $Center - Cexit$. Αν στο σύστημα παραμένουν N κομμάτια, θα αφαιρεθεί πίστωση από το κάθε ένα ίση με $(Center - Cexit)/N$. Αντίστοιχος χειρισμός πρέπει να γίνει και στην περίπτωση όπου $Center < Cexit$.

Καθ' όλη τη διάρκεια ζωής ενός κομματιού μέσα στο σύστημα μας, η πίστωσή του επηρεάζεται από δύο παράγοντες: (α) τους ψήφους των χρηστών και (β) τον χρόνο.

a.

Ψήφοι χρηστών

Κάθε χρήστης, ακούγοντας ένα κομμάτι, έχει τη δυνατότητα να το ψηφίσει θετικά ή αρνητικά. Σε σχέση με την ψήφο των χρηστών πρέπει να αντιμετωπιστούν τρία βασικά ζητήματα:

1. Η διαφορά στη συχνότητα με την οποία ψηφίζουν (θετικά ή αρνητικά) διαφορετικοί χρήστες
2. Η επανειλημμένη αποστολή ψήφου από τον ίδιο χρήστη προς ένα (ίδιο) κομμάτι
3. Το *Reputation Hacking*, δηλαδή η απόπειρα τροποποίησης της δημοτικότητας συγκεκριμένων κομματιών από χρήστες προς όφελος συγκεκριμένων καλλιτεχνών.

Η επίλυση αυτού του τρίτου ζητήματος συνδέεται άμεσα με την επίλυση των προηγούμενων δύο

Για να αντιμετωπιστούν τα παραπάνω ζητήματα, πρέπει:

1. **Να επιλεγεί ένα άνω όριο πίστωσης U που μπορεί ένας χρήστης να «διαχειριστεί» με τις ψήφους του μέσα σε ένα προκαθορισμένο χρονικό διάστημα T , όπου U και T**

μεταβλητές του αλγορίθμου. Με τον τρόπο αυτό δεν περιορίζουμε το πλήθος των ψήφων που μπορεί να δώσει ένας χρήστης στο χρονικό διάστημα T , αναπροσαρμόζουμε όμως το βάρος κάθε μιας, έτσι ώστε οι ψήφοι των χρηστών που ψηφίζουν συχνότερα να έχουν μικρότερη αξία. Ο πιο απλός τρόπος να επιλεγεί το βάρος κάθε ψήφου ενός χρήστη είναι:

$$w = U / (\# \text{ votes in } T)$$

Εάν, επιπλέον, θέλουμε να «συγκρατήσουμε» το βάρος των ψήφων μέσα σε κάποια όρια, μπορούμε να θέσουμε:

$$w = \min\{U / (\# \text{ votes in } T), b\}$$

όπου b μια επιπρόσθετη παράμετρος του αλγορίθμου.

Να σημειωθεί ότι αν η εν λόγω ψήφος είναι αρνητική, αφαιρείται πίστωση ίση με w από το κομμάτι.

Ένας τρόπος υπολογισμού του U (για κάθε χρήστη) θα μπορούσε να σχετίζεται με τη συνολική πίστωση που διαθέτει το σύστημα τη στιγμή του υπολογισμού.

$$U = \mu \cdot (\text{total system credit}) / (\# \text{ users}), 0 < \mu \leq 1$$

όπου μ παράμετρος του αλγορίθμου, που καθορίζει το ποσοστό της συνολικής πίστωσης που θα «μετακινηθεί» μέσω των ψήφων των χρηστών.

Ο παραπάνω υπολογισμός του U γίνεται με περίοδο T , μαζί με τον υπολογισμό του w .

2. **Να μην επιτρέπεται στους χρήστες να ψηφίζουν κομμάτια που έχουν ήδη ψηφίσει**, εκτός εάν κάθε νέα ψήφος τους είναι διαφορετική από την προηγούμενη. Για παράδειγμα, επιτρέπουμε στο χρήστη να ψηφίσει θετικά ένα κομμάτι που έχει καταψηφίσει στο παρελθόν, ακυρώνοντας την αρνητική ψήφο και αποδίδοντας στο κομμάτι πίστωση ίση με το άθροισμα της πίστωσης που έχασε λόγω της (παλιάς) αρνητικής ψήφου συν την πίστωση που αναλογεί στη (νέα) θετική ψήφο.

Επιπλέον, επιθυμούμε κάθε ψήφος που αφαιρεί/προσθέτει πίστωση από/σε ένα κομμάτι, να προσθέτει/αφαιρεί το ίδιο ποσό σε/από όλα τα υπόλοιπα κομμάτια που βρίσκονται εκείνη τη στιγμή στο σύστημα μας. Συνεπώς, ψηφίζοντας ένα κομμάτι δε μεταβάλλουμε τη συνολική πίστωση του συστήματος.

b.

Χρόνος

Θεωρούμε πως η δημοτικότητα όλων των κομματιών μειώνεται με το χρόνο λόγω παλαιότητας (*decay*). Ως αποτέλεσμα, ευνοείται, μέσω του αλγορίθμου κατασκευής λίστας εκτέλεσης, η επιλογή των νέων κομματιών. Για να μοντελοποιήσουμε μια τέτοιου είδους μείωση, θα μπορούσαμε να χρησιμοποιήσουμε μία αναδρομική συνάρτηση της μορφής:

$$\text{credits}(kT) = d \cdot \text{credits}(\{k-1\} \cdot T) \text{ , όπου } 0 < d < 1, k \in \mathbb{Z}, k \geq 1$$

Ισοδύναμα:

$$\text{credits}(kT) = d^k \cdot \text{credits}(0) \text{ , όπου } 0 < d < 1, k \in \mathbb{Z}, k \geq 1$$

Δηλαδή, με το πέρασμα κάθε χρονικού διαστήματος T η δημοτικότητα κάθε κομματιού μειώνεται κατά $(1-d) \cdot 100\%$, όπου d μία ακόμα παράμετρος του αλγορίθμου.

ΥΛΟΠΟΙΗΣΗ

ΑΡΧΙΚΗ ΚΑΤΑΣΤΑΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ:

Θεωρώντας πως στο σύστημά μας υπάρχει ήδη αποθηκευμένος ένας αριθμός κομματιών, σε κάθε κομμάτι εκχωρείται πίστωση ίση με C (όλα τα κομμάτια ξεκινούν με την ίδια δημοτικότητα).

Ο ΑΛΓΟΡΙΘΜΟΣ ΘΑ ΕΚΤΕΛΕΙΤΑΙ ΜΕ ΠΕΡΙΟΔΟ T ΚΑΙ ΘΑ ΕΝΕΡΓΕΙ ΩΣ ΕΞΗΣ:

1. Για κάθε χρήστη:
 - a. Εάν έχει ψηφίσει τουλάχιστον μία φορά στο χρονικό διάστημα T , θα υπολογίζει το βάρος κάθε μίας εκ των ψήφων του:

$$w = U / (\# \text{ votes in } T) \quad \text{ή} \quad w = \min\{U / (\# \text{ votes in } T), b\}$$

- b. Αναθέτει στο U νέα τιμή:

$$U = \mu \cdot (\text{total system credit}) / (\# \text{ users}), 0 < \mu \leq 1$$

2. Για κάθε κομμάτι που έχει ψηφιστεί μέσα στο χρονικό διάστημα T :

Του προσθέτει/αφαιρεί (ανάλογα με το είδος της ψήφου), για κάθε ψήφο που δέχτηκε, πίστωση ίση με w και αφαιρεί/προσθέτει από/σε όλα τα υπόλοιπα κομμάτια πίστωση ίση με:

$$w / (\# \text{ κομματιών} - 1)$$

3. Για κάθε κομμάτι που υπάρχει στη βάση:

Υπολογίζει τη νέα πίστωση ως το γινόμενο της πίστωσης που προέκυψε από το προηγούμενο βήμα με τη σταθερά d .

Σημείωση: Τα βήματα 2 και 3 θα μπορούσαν να εκτελεστούν και με αντίστροφη σειρά. Το μειονέκτημα μίας τέτοιας προσέγγισης, όμως, είναι ότι η πίστωση κάθε κομματιού μεταβάλλεται περισσότερο (είτε θετικά είτε αρνητικά, ανάλογα με το ισοζύγιο και τα βάρη των ψήφων), καθώς ο συντελεστής παλαιότητας είναι μικρότερος της μονάδας.

ΚΑΤΑ ΤΗΝ ΑΠΟΔΟΣΗ ΨΗΦΟΥ ΑΠΟ ΧΡΗΣΤΗ ΣΕ ΚΟΜΜΑΤΙ:

Ελέγχει αν ο χρήστης έχει ψηφίσει ήδη το κομμάτι:

1. Αν το έχει ψηφίσει ήδη, ελέγχει εάν η νέα ψήφος διαφέρει από την προηγούμενη:
 - a. Αν διαφέρει, ελέγχει εάν η προηγούμενη ψήφος αποδόθηκε στο κομμάτι μέσα στα πλαίσια του τρέχοντος χρονικού παραθύρου T :
 - i. Αν ναι, αντικαθιστά την παλιά ψήφο με τη νέα. Συνεπώς, μετά το πέρας του τρέχοντος χρονικού διαστήματος T και αν αποδόθηκαν πολλές ψήφοι από τον ίδιο χρήστη στο ίδιο κομμάτι, θα μετρήσει (μόνο) η τελευταία ψήφος που αλλάζει την κατάσταση για το κομμάτι αυτό
 - ii. Αν όχι, «ακυρώνει» την παλιά ψήφο επιστρέφοντας στο κομμάτι ή αφαιρώντας από αυτό (ανάλογα με το είδος της παλιάς ψήφου) τόσα *credits* όσα αντιστοιχούσαν στην παλιά ψήφο και αντικαθιστά την παλιά ψήφο με τη νέα. Τα *credits* που αντιστοιχούν στη νέα ψήφο, θα αποδοθούν στο κομμάτι ή θα αφαιρεθούν από αυτό στο τέλος του χρονικού παραθύρου T (σύμφωνα με τη διαδικασία που αναφέρθηκε παραπάνω)
 - b. Αν δε διαφέρει, αγνοεί τη νέα ψήφο
2. Αν το ψηφίζει για πρώτη φορά, καταγράφει την ψήφο ώστε να αποδοθούν στο κομμάτι ή να αφαιρεθούν από αυτό τα ανάλογα *credits* μετά το πέρας του τρέχοντος χρονικού διαστήματος T (σύμφωνα με τη διαδικασία που αναφέρθηκε παραπάνω)

Σημείωση: Είναι φανερή η απαίτηση να συγκρατούμε, για κάθε ψήφο, το βάρος (σε credits) της ψήφου καθώς και μία χρονοσφραγίδα (timestamp) που θα αντιστοιχεί στη χρονική στιγμή απόδοσής της, ώστε να έχουμε τη δυνατότητα να ακυρώνουμε παλιές ψήφους, όταν αυτό χρειάζεται.

Σημείωση: Εφόσον, για κάθε χρήστη, μας ενδιαφέρει μόνο η πιο πρόσφατη ψήφος που έχει αποδώσει σε κάθε κομμάτι, κάθε δεδομένη χρονική στιγμή θα υπάρχει αποθηκευμένη στη βάση δεδομένων το πολύ μία εγγραφή που θα συσχετίζει τον συγκεκριμένο χρήστη με ένα (ίδιο) κομμάτι.

ΚΑΤΑ ΤΗ ΔΙΑΓΡΑΦΗ ΚΟΜΜΑΤΙΟΥ ΑΠΟ ΤΟ ΣΥΣΤΗΜΑ:

1. Αν $Center > Cexit$, διαγράφεται το κομμάτι και αφαιρείται από κάθε ένα εκ των υπολοίπων κομματιών (έστω n σε πλήθος) πίστωση ίση με $(Center - Cexit)/n$
2. Αν $Center < Cexit$, διαγράφεται το κομμάτι και προστίθεται σε κάθε ένα εκ των υπολοίπων κομματιών (έστω n σε πλήθος) πίστωση ίση με $(Cexit - Center)/n$
3. Αν $Center = Cexit$ απλά διαγράφεται το κομμάτι

Σημείωση: Κομμάτια διαγράφονται από το σύστημα μόνο κατ' απαίτηση του δημιουργού.

Σημείωση: Εφόσον υπάρχει και ο παράγοντας παλαιότητας, μπορούμε να αντικαταστήσουμε στους παραπάνω υπολογισμούς το C_{enter} με $d^k \cdot C_{enter}$ όπου $k = [(t_{deletion} - t_{enter})/T]$. Με τον τρόπο αυτό είμαστε πιο «δίκαιοι» απέναντι στο σύνολο των κομματιών που παραμένουν στο σύστημα, καθώς το σύστημα με το χρόνο «γεράζει», χάνοντας μέρος της συνολικής του αξίας, οπότε τη γήρανση αυτή πρέπει να τη λάβουμε υπόψη και κατά τη διαγραφή των κομματιών.

ΚΑΤΑ ΤΗΝ ΕΙΣΟΔΟ ΕΝΟΣ ΝΕΟΥ ΚΟΜΜΑΤΙΟΥ ΣΤΟ ΣΥΣΤΗΜΑ:

Εκχωρείται στο νέο κομμάτι πίστωση ίση με $a \cdot E(C)$, όπου a παράμετρος και $E(C)$ η μέση τιμή της πίστωσης για όλα τα κομμάτια που βρίσκονται εκείνη τη στιγμή στο σύστημα.

ΑΝΕΞΑΡΤΗΤΕΣ ΜΕΤΑΒΛΗΤΕΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ:

μ: ποσοστό της συνολικής πίστωσης του συστήματος που μπορούν να αναδιανείμουν (όλοι) οι χρήστες με τις ψήφους τους

T: περίοδος υπολογισμού των w , U και ανάθεσης πίστωσης στα κομμάτια με βάση τις ψήφους των χρηστών

a: παράγοντας που ρυθμίζει την αρχική πίστωση κάθε νέου κομματιού στο σύστημα σε σχέση με το μέσο όρο

d: παράγοντας που επηρεάζει το ρυθμό με τον οποίο φθίνει η δημοτικότητα των κομματιών στο σύστημα με το πέρασμα του χρόνου (ενδεικτική τιμή $d=0,98$)

b: παράγοντας που καθορίζει το μέγιστο δυνατό βάρος μίας ψήφου

C: η πίστωση που εκχωρείται σε κάθε κομμάτι κατά την αρχικοποίηση του συστήματος

4.2 Δημιουργία ραδιοφωνικών σταθμών

Αφήνουμε τους χρήστες ελεύθερους να αποδίδουν στα κομμάτια, στα άλμπουμ και στους καλλιτέχνες τις ετικέτες που επιθυμούν. Ωστόσο, κάθε φορά τους προτείνουμε ένα σύνολο ετικετών, ως πιο ταιριαστές για το καθετί που επιθυμούν να χαρακτηρίσουν. Τα ζητήματα που πρέπει να επιλυθούν είναι τα εξής:

1. Πρέπει να είμαστε σε θέση να αποφασίζουμε για κάθε κομμάτι, άλμπουμ και καλλιτέχνη σε ποιές κατηγορίες ανήκει. Για παράδειγμα, εάν ένα κομμάτι έχει χαρακτηριστεί 200 φορές: 120 ως «rock», 40 ως «alternative», 38 ως «pop rock» και 2 ως «metal» είναι προφανές ότι δεν μπορεί να ενταχθεί στο σταθμό της ετικέτας «metal», καθώς οι δύο χαρακτηρισμοί θεωρούνται άστοχοι μπροστά στο σύνολο των διακοσίων. Για διευκόλυνση, μπορούμε περιοδικά να ελέγχουμε τις ετικέτες των κομματιών, των

άλμπουμς και των καλλιτεχνών και για το κάθε ένα να ανακαλύπτουμε μία ή περισσότερες επικρατούσες ετικέτες.

Σημείωση: Εάν μία επικρατούσα ετικέτα δεν πλειοψηφεί με μεγάλη διαφορά, μπορούμε να λάβουμε υπόψη και μία ή περισσότερες από τις μειοψηφούσες ετικέτες.

2. Εφόσον δεν υπάρχουν εκ των προτέρων καθορισμένες ετικέτες, δεν υπάρχουν και εκ των προτέρων καθορισμένοι ραδιοφωνικοί σταθμοί. Κατά συνέπεια, ο αλγόριθμος πρέπει να έχει τη δυνατότητα να συγχωνεύει τις συγγενικές ετικέτες και να δημιουργεί/καταστρέφει ραδιοφωνικούς σταθμούς όταν ο αριθμός των κομματιών τους γίνεται μεγαλύτερος ή μικρότερος, αντίστοιχα, από ένα προκαθορισμένο όριο (π.χ. απαιτούμε ένας ραδιοφωνικός σταθμός να διαθέτει τουλάχιστον 100 διαφορετικά κομμάτια).

Σημείωση: Σε κάθε περίπτωση, στόχος είναι να αποφευχθεί η δημιουργία μεγάλου αριθμού κατηγοριών με λίγα κομμάτια σε κάθε μία, καθώς έτσι θα είναι δύσκολο να αντιστοιχίσουμε τις κατηγορίες σε σταθμούς, λόγω του μικρού αριθμού κομματιών. Για το λόγο αυτό, είναι επιθυμητό κομμάτια με συγγενικές ετικέτες να ομαδοποιούνται και να εντάσσονται στον ίδιο ραδιοφωνικό σταθμό.

3. Είναι σκόπιμο να μην επιτρέπουμε στους χρήστες να χαρακτηρίζουν κομμάτια που έχουν ήδη χαρακτηρίσει. Για κάθε χρήστη λαμβάνουμε υπόψη μόνο τον τελευταίο χαρακτηρισμό που έχει αποδώσει σε κάθε κομμάτι.

Αλγόριθμος

Ο αλγόριθμος που θα χρησιμοποιήσουμε αποτελεί μία παραλλαγή των αλγορίθμων ιεραρχικής συσταδοποίησης (hierarchical clustering) και ως μέτρο ομοιότητας/απόστασης μεταξύ δύο στοιχείων-ετικετών, έστω a και b , χρησιμοποιεί το μέτρο Jaccard⁴:

$$\text{Similarity}(a,b) = \frac{|S(a) \cap S(b)|}{|S(a) \cup S(b)|}, 0 \leq \text{Similarity}(t,c) \leq 1$$

Για να γίνει κατανοητός ο λόγος για τον οποίο ένα τέτοιο μέτρο ομοιότητας έχει νόημα, έστω δύο ετικέτες a και b . Έστω, επίσης, $S(a)$ το σύνολο των κομματιών που χαρακτηρίζονται από την ετικέτα a και $S(b)$ το σύνολο των κομματιών που χαρακτηρίζονται από την ετικέτα b . Εάν υποθέσουμε ότι $S(a) \cap S(b) \neq \emptyset$, τότε θα υπάρχουν κομμάτια που χαρακτηρίζονται ταυτόχρονα (με ισχυρό τρόπο) και από τις δύο ετικέτες. Όσο μεγαλύτερο είναι το πλήθος των κομματιών

⁴ Για περαιτέρω συζήτηση πάνω στη συσταδοποίηση βασιζόμενη σε ετικέτες, αλλά και πάνω στα διαφορετικά μέτρα ομοιότητας παραπέμπουμε στο σύνδεσμο:

http://www.pui.ch/phred/automated_tag_clustering/

που ανήκουν ταυτόχρονα και στις δύο κατηγορίες, σε σχέση πάντα με την πληθικότητα της ένωσης των κομματιών $|S(a) \cup S(b)|$, τόσο μεγαλύτερη συγγένεια αναμένεται να παρουσιάζουν οι ετικέτες a και b .

Σημείωση: Από εδώ και στο εξής, αναφερόμαστε μόνο στις επικρατούσες ετικέτες των κομματιών, καθώς μόνο αυτές λαμβάνονται υπόψη στα πλαίσια του αλγορίθμου.

Ο αλγόριθμος θα εκτελείται με περίοδο T , μετά την ολοκλήρωση εκτέλεσης του αλγορίθμου ψηφοφορίας που περιγράψαμε στην προηγούμενη ενότητα και θα πραγματοποιεί τις εξής δύο λειτουργίες:

1. Υπολογισμός επικρατούσας ετικέτας για κάθε κομμάτι, άλμπουμ και καλλιτέχνη στη βάση δεδομένων.

Ο τρόπος υπολογισμού των ετικετών που επικρατούν είναι ο ίδιος και στις τρεις περιπτώσεις (κομμάτια, άλμπουμ, καλλιτέχνες): η ετικέτα που έχει αποδοθεί τις περισσότερες φορές εκλαμβάνεται πάντοτε ως επικρατούσα. Από όλες τις υπόλοιπες ετικέτες, πλην της πλειοψηφούσας, ως επικρατούσες εκλαμβάνονται όσες έχουν αποδοθεί σε ποσοστό μεγαλύτερο ή ίσο από 20% στο πλήθος των χαρακτηρισμών.

2. Συσταδοποίηση ετικετών και δημιουργία ραδιοφωνικών σταθμών.

Έχοντας υπολογίσει τις επικρατούσες ετικέτες για κάθε κομμάτι, άλμπουμ και καλλιτέχνη, ο αλγόριθμος λειτουργεί ως εξής:

a. Έστω σύνολο N , στο οποίο αρχικά ανήκουν όλες οι διακριτές επικρατούσες ετικέτες των κομματιών⁵ και κάθε ετικέτα αποτελεί μία συστάδα.

b. Για κάθε ζεύγος ετικετών $t, c: t, c \in N, c \neq t$ υπολογίζουμε την τιμή:

$$\text{Similarity}(t, c) = \frac{|S(t) \cap S(c)|}{|S(t) \cup S(c)|}, 0 \leq \text{Similarity}(t, c) \leq 1$$

όπου $S(c), S(t)$ τα σύνολα των κομματιών στις επικρατούσες ετικέτες των οποίων συμπεριλαμβάνονται οι c και t αντίστοιχα. Μεγαλύτερες τιμές του Similarity καταδεικνύουν μεγαλύτερη ομοιότητα μεταξύ των δύο ετικετών για τις οποίες έχει υπολογιστεί.

c. Έστω $n = |N|$, το πλήθος των ετικετών που ανήκουν αρχικά στο σύνολο N . Εκτελούμε το πολύ $n-1$ βήματα και σε κάθε βήμα ενεργούμε ως εξής:

i. Έστω συστάδες $t', c' \in N, t' \neq c'$:

$$\text{Similarity}(t', c') = \max(\text{Similarity}(t, c)), \forall t, c \in N, t \neq c$$

⁵ Στις επικρατούσες ετικέτες των κομματιών συγκαταλέγονται και οι επικρατούσες ετικέτες που κληρονομούν από το συσχετιζόμενο album και τον συσχετιζόμενο καλλιτέχνη

Αν $\text{Similarity}(t',c') \leq \text{Threshold}$ τερματίζουμε τον αλγόριθμο

ii. $N=N-\{t'\}-\{c'\}+\{t',c'\}$

iii. $\forall t \in N: t \neq \{t',c'\}$ υπολογίζουμε:

$$\text{Similarity}(\{t',c'\},t)=1/((\{t',c'\} \cdot |t|) \cdot \sum_{x \in \{t',c'\}, y \in t} \text{Similarity}(x,y))$$

iv. Αν έχουν πραγματοποιηθεί λιγότερες από n-1 επαναλήψεις, επαναλαμβάνουμε τα βήματα I – III

Όταν θα τερματίσει ο παραπάνω αλγόριθμος, το N θα περιέχει ένα σύνολο συστάδων, οι ετικέτες-μέλη των οποίων θα θεωρούνται συγγενικές (πάντα με βάση τις προτιμήσεις των χρηστών). Για κάθε μία από τις παραπάνω συστάδες ετικετών, υπολογίζουμε το πλήθος των κομματιών που χαρακτηρίζονται από τις ετικέτες της εν λόγω συστάδας⁶. Αν ξεπερνά το κατώφλι που έχουμε θέσει, δημιουργείται ο αντίστοιχος ραδιοφωνικός σταθμός.

4.3 Δημιουργία λιστών αναπαραγωγής

Σε αυτήν την ενότητα θα εξηγήσουμε πως δημιουργούνται οι λίστες των σταθμών της εφαρμογής μας. Το σύστημα αναπροσαρμόζει τις λίστες των σταθμών, οι οποίες έχουν συγκεκριμένη διάρκεια, ανά τακτά χρονικά διαστήματα και φροντίζει τραγούδια τα οποία συγκεντρώνουν πολλές θετικές ψήφους (μεγάλα credits) να εμφανίζονται περισσότερες φορές σε σχέση με τα υπόλοιπα που έχουν χαμηλά credits. Το πρόβλημα που αντιμετωπίσαμε στη δημιουργία των λιστών αναπαραγωγής ταιριάζει με το knapsack⁷ πρόβλημα. Το πρόβλημα του σακιδίου μπορεί να λυθεί με βάση τρεις διαφορετικές προσεγγίσεις :

- Δυναμικού προγραμματισμού
- Άπληστου αλγόριθμου
- Γενετικού αλγόριθμου

Επιλέξαμε την προσέγγιση του γενετικού αλγόριθμου γεγονός που μας οδήγησε στην χρήση του αλγόριθμου της ρουλέτας (roulette wheel algorithm) για να γίνει η σωστή επιλογή των κομματιών που θα μπαίνουν κάθε φορά στις λίστες.

⁶ Επειδή τα κομμάτια κληρονομούν τις ετικέτες από το album και τον καλλιτέχνη στον οποίο ανήκουν, για να ενταχθεί ένα κομμάτι στο ραδιοφωνικό σταθμό μίας ετικέτας, αρκεί η εν λόγω ετικέτα να επικρατεί είτε στο κομμάτι, είτε στο αντίστοιχο album, είτε στον αντίστοιχο καλλιτέχνη

⁷ http://en.wikipedia.org/wiki/Knapsack_problem

4.3.1 Επιλογή τραγουδιών

Ο αλγόριθμος της ρουλέτας, κάθε φορά που εκτελείται, δέχεται σαν παράμετρο μια λίστα με τραγούδια, έστω `list`, τα οποία ταιριάζουν με το είδος μουσικής που θα περιλαμβάνει η λίστα αναπαραγωγής του σταθμού. Αρχικά, ρυθμίζουμε τη διάρκεια που θέλουμε να έχουν οι λίστες. Η διάρκεια της λίστας που θα δημιουργεί το σύστημα συμβολίζεται με `radioStationDuration` και αποτελεί ανεξάρτητη μεταβλητή του αλγορίθμου. Επίσης, υπολογίζει τον μέγιστο επιτρεπόμενο αριθμό εμφάνισης τραγουδιού στη λίστα που ισούται με τη διάρκεια της λίστας διá 2, εφόσον θέλουμε το κάθε τραγούδι να παίζεται το πολύ μία φορά κάθε δύο ώρες στην καλύτερη περίπτωση. Ο αριθμός αυτός συμβολίζεται ως `R`.

Στη συνέχεια, σαρώνεται όλη η λίστα `list`, που η ρουλέτα έχει δεχθεί ως παράμετρο, με σκοπό να ξεχωρίσουν τα τραγούδια που έχουν θετικό `popularity`. Κατά τη διάρκεια αυτής της σάρωσης αθροίζονται τα θετικά `popularities` των τραγουδιών, με αποτέλεσμα να χτίζονται `hashMaps` όπου κάθε ένα έχει ως τιμή το τραγούδι (που έχει θετική δημοφιλία) και ως κλειδί το άθροισμα των `popularities` ως εκείνη τη στιγμή. Το όνομα ενός τέτοιου `hashMap` είναι `songsMap`. Καθ' όλη τη διάρκεια της σάρωσης της αρχικής λίστας `list` δημιουργούνται πολλά `songsMap`, τα οποία αποθηκεύονται, σε κάθε βήμα της επανάληψης σε ένα τελικό `hashMap` με όνομα `sequenceMap`, το οποίο έχει ως τιμή έναν ακέραιο που αντιπροσωπεύει το πλήθος των κομματιών (ως εκείνη τη στιγμή) και ως τιμή το `songsMap` που δημιουργείται στο συγκεκριμένο βήμα της σάρωσης της λίστας.

Όλα τα τραγούδια που βρίσκονται σε αυτό το `hashMap` συμμετέχουν στη διαδικασία επιλογής των κομματιών που θα συμπληρώσουν την τελική ραδιοφωνική λίστα με βάση την δημοφιλία τους. Παράγεται λοιπόν, ένας τυχαίος αριθμός από 0 έως το συνολικό άθροισμα των `popularities` αυτών των τραγουδιών. Ο αριθμός που παράγεται συμβολίζεται ως `rand` και στην ουσία δείχνει ποιο είναι το τραγούδι το οποίο θα επιλεγεί για να μπει στην τελική λίστα. Το κομμάτι θα βρεθεί ύστερα από δυαδική αναζήτηση στο εύρος των `popularities` που υπάρχει στο `sequenceMap`. Η αναζήτηση γίνεται σε κάθε θέση του `sequenceMap`, ελέγχοντας το κλειδί της κάθε θέσης, το οποίο με τη σειρά του περιέχει ένα `hashMap`. Κάθε ένα τέτοιο `hashMap` περιέχει ένα κλειδί το οποίο μας δείχνει το `popularity` του κομματιού, το οποίο μεταφράζεται σε εύρος εάν λάβουμε υπ' όψιν το `popularity` του ακριβώς προηγούμενου τραγουδιού. Με αυτόν τον τρόπο βρίσκουμε σε ποιο εύρος ανήκει ο αριθμός που παράχθηκε και αυτομάτως έχουμε βρει το τραγούδι που θα προστεθεί στην ραδιοφωνική λίστα. Έτσι, τα τραγούδια που είναι πιο δημοφιλή σε σχέση με τα υπόλοιπα έχουν μεγαλύτερη πιθανότητα να επιλεγούν, αφού θα έχουν μεγαλύτερο εύρος σε σχέση με τα υπόλοιπα που θα είναι λιγότερο δημοφιλή.

Στη συνέχεια, το τραγούδι που επιλέγεται, προστίθεται στο hashMap με όνομα repeat, στο οποίο αποθηκεύεται το κάθε τραγούδι που θα αναπαραχθεί στην τελική λίστα μαζί με τον αριθμό των φορών που θα εμφανιστεί στη λίστα αυτή.

Έτσι, αφού ολοκληρωθεί η δημιουργία του hashMap repeat, ο αλγόριθμος της ρουλέτας τερματίζει και καλείται ο αλγόριθμος διάταξης της ραδιοφωνικής λίστας. Ο αλγόριθμος αυτός έχει υλοποιηθεί έτσι ώστε να επιτυγχάνεται η καλύτερη δυνατή κατανομή του κάθε τραγουδιού που θα παίζει παραπάνω από μία φορά στο σταθμό.

Οι ανεξάρτητες μεταβλητές του αλγόριθμου της ρουλέτας

1. radioStationDuration: Η διάρκεια της λίστας αναπαραγωγής των σταθμών στην εφαρμογή

4.3.2 Διάταξη λίστας

Έπειτα από την ολοκλήρωση του αλγόριθμου της ρουλέτας, έχει δημιουργηθεί το HashMap repeat, το οποίο περιέχει σαν κλειδιά τα τραγούδια που θα παίζουν και σαν τιμές το πόσες φορές επαναλαμβάνεται κάθε ένα στη λίστα. Επειδή θα πρέπει να γνωρίζουμε τον μέγιστο αριθμό φορών που ένα κομμάτι θα εμφανιστεί στην λίστα κάνουμε μια αναζήτηση στο repeat για να βρούμε την μέγιστη τιμή στην οποία μπορεί να αντιστοιχεί κάποιο κλειδί (τραγούδι). Αφού βρεθεί η μέγιστη τιμή, θα αποθηκευτεί στην μεταβλητή max. Ο μέγιστος αριθμός φορών που ένα κομμάτι θα εμφανιστεί στην λίστα (max) δεν θα πρέπει να ξεπερνάει τον μέγιστο επιτρεπόμενο αριθμό εμφάνισης ενός τραγουδιού(R). Επομένως θα πρέπει να ισχύει : $1 \leq \max \leq R$.

Στη συνέχεια δημιουργούνται max δοχεία(buckets) τα οποία θα συμπληρωθούν με τα τραγούδια που περιέχονται στην λίστα. Γίνεται σάρωση του repeat ξεκινώντας από το πρώτο στοιχείο. Κάθε τραγούδι τοποθετείται από μία φορά σε τόσα δοχεία όσος είναι ο αριθμός των φορών που θα εμφανιστεί στη λίστα. Η απόσταση που θα έχουν τα buckets στα οποία θα τοποθετηθεί το κάθε τραγούδι προκύπτει από το (στρογγυλοποιημένο) πηλίκο της διαίρεσης του μέγιστου αριθμού φορών που ένα κομμάτι θα εμφανιστεί στην λίστα διά τον αριθμό φορών που θα επαναληφθεί στη λίστα το συγκεκριμένο τραγούδι που εξετάζεται. Δηλαδή η απόσταση αυτή προκύπτει από την πράξη: $\max/\#\text{φορών εμφάνισης τραγουδιού}$, στρογγυλοποιώντας επίσης το αποτέλεσμα. Ο αλγόριθμος συνεχίζει για όλα τα στοιχεία του repeat. Σε κάθε βήμα, δηλαδή για κάθε γραμμή του στοιχείο του repeat, το νέο τραγούδι δεν αρχίζει να τοποθετείται ξεκινώντας πάντα από το πρώτο δοχείο αλλά συνεχίζει απ' το δοχείο που είχε συμπληρωθεί τελευταίο, στο προηγούμενο βήμα του αλγορίθμου. Επίσης κάθε φορά που ο αλγόριθμος ξεπερνάει το n-οστό δοχείο, συνεχίζει από το πρώτο δοχείο.

Υλοποίηση

Ο αλγόριθμος θα τρέχει κάθε n ώρες ($n = \text{radioStationDuration}$), αφού πρώτα έχει η ρουλέτα δημιουργήσει τον πίνακα `repeat` τον οποίο περνάει σαν παράμετρο στον αλγόριθμο κατανομής.

1. Αρχικά,
 - Δημιουργούνται `max` δοχεία.
2. Κάθε τραγούδι που περιέχει το `HashMap repeat` :
 - Αντιγράφεται σε τόσα δοχεία όσες είναι οι φορές που θα εμφανιστεί στη λίστα. Το δοχείο από το οποίο αρχίζει να αντιγράφεται το κομμάτι είναι αυτό που ακολουθεί το δοχείο στο οποίο σταμάτησε να αντιγράφεται το προηγούμενο τραγούδι του `repeat`. Η απόσταση μεταξύ των δοχείων προκύπτει από τη στρογγυλοποίηση της διαίρεσης `max/repeat.get(song)`.
3. Τέλος,
 - Τα κομμάτια όλων των δοχείων αντιγράφονται με τη σειρά, ξεκινώντας από το πρώτο τραγούδι του πρώτου δοχείου, σε μία νέα λίστα η οποία αποτελεί την τελική λίστα αναπαραγωγής.

Αξιολόγηση της τελικής λίστας

Η τελική λίστα αναπαραγωγής μπορεί να αξιολογηθεί από τον εξής τύπο:

$$\delta p = \frac{td}{T} * \sum V_t * \frac{\sum d_i}{D}$$

όπου,

td : Συνολικός αριθμός διακριτών τραγουδιών στην λίστα.

T : Συνολικός αριθμός τραγουδιών στην κατηγορία που ανήκει η λίστα.

V_t : Η δημοφιλία του κάθε κομματιού που υπάρχει στη λίστα.

d_i : Η μέση απόσταση μεταξύ των ίδιων κομματιών.

D : Το άθροισμα των αποστάσεων μεταξύ των τραγουδιών από την αρχή μέχρι το τέλος της λίστας.

Η αξιολόγηση της λίστας θα βοηθήσει στη σύγκριση του αλγορίθμου της ρουλέτας και της κατανομής των τραγουδιών με πιθανές λύσεις που μπορεί να προσφέρουν άλλοι αλγόριθμοι.

5

Αρχιτεκτονική συστήματος και τεχνολογίες υλοποίησης

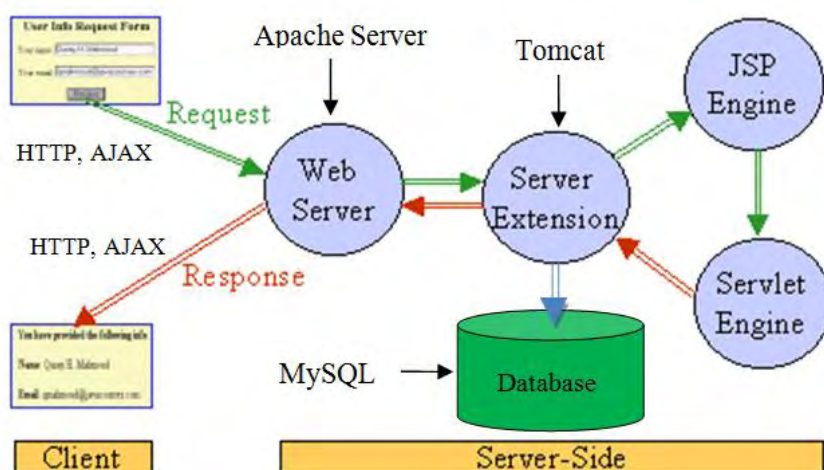
5.1 Αρχιτεκτονική

Σε αυτό το κεφάλαιο θα περιγράψουμε την αρχιτεκτονική της διαδικτυακής μας εφαρμογής (web application). Ως web application ορίζεται ένα λογισμικό το οποίο είναι προσβάσιμο μέσω του παγκόσμιου ιστού (Internet). Για το σκοπό αυτό υπάρχουν συνήθως δύο βασικές συνιστώσες που επιτρέπουν στους χρήστες να αλληλεπιδρούν με την εφαρμογή:

1. Ένας Εξυπηρετητής (web Server) που εξυπηρετεί τις διάφορες αιτήσεις των χρηστών και επιστρέφει ως απάντηση κυρίως αρχεία HTML, αλλά και άλλου τύπου όπως JSON και XML.
2. Μια Βάση Δεδομένων (Database) με την οποία επικοινωνεί ο web Server για να αποθηκεύσει, να ανακτήσει ή και να τροποποιήσει δεδομένα απαραίτητα για τη λειτουργία της εφαρμογής.

Το σημαντικότερο πλεονέκτημα των διαδικτυακών εφαρμογών είναι ότι από την πλευρά των χρηστών (πλευρά πελάτη – client side) δεν απαιτείται κάποιο άλλο είδος λογισμικού εκτός από φυλλομετρητή (browser) όπως Mozilla Firefox, Google Chrome, Opera, Safari και Internet Explorer.

Στην εφαρμογή χρησιμοποιούμε ως Server τον Apache Tomcat, ο οποίος είναι μια επέκταση του Apache Server έτσι ώστε να περιέχει έναν container που υποστηρίζει Servlets και JSP. Θα μιλήσουμε πιο συγκεκριμένα για τον Tomcat container, τα Servlets και το JSP σε παρακάτω υποενότητες. Ως Βάση Δεδομένων επιλέξαμε να χρησιμοποιήσουμε τη MySQL η οποία είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) απλή στην εγκατάσταση και κατάλληλη για το μοντέλο βάσης που δημιουργήσαμε, ως προς τις λειτουργίες διαχείρισης και τις σχέσεις μεταξύ των πινάκων. Έτσι οι διάφοροι χρήστες χρησιμοποιώντας κάποιον browser επικοινωνούν με τον Apache Tomcat Server της εφαρμογής μας και έχουν πρόσβαση στις διάφορες υπηρεσίες που προσφέρουμε. Για τη μεταφορά δεδομένων μεταξύ πελατών (clients) και Server χρησιμοποιείται το πρωτόκολλο HTTP⁸, γνωστό και ως το «πρωτόκολλο του διαδικτύου». Επειδή η υλοποίησή μας υποστηρίζει ασύγχρονη επικοινωνία (χωρίς να φορτώνεται ή να ανανεώνεται μία ιστοσελίδα) η μεταφορά δεδομένων γίνεται και με χρήση της τεχνολογίας AJAX⁹. Η αρχιτεκτονική της υλοποίησής μας φαίνεται στην Εικόνα 5-1.



Εικόνα 5-1: Η αρχιτεκτονική της διαδικτυακής εφαρμογής

5.2 Τεχνολογίες

Για την υλοποίηση μιας διαδικτυακής εφαρμογής είναι απαραίτητη η υιοθέτηση συγκεκριμένων τεχνικών και η ενσωμάτωση αρκετών τεχνολογιών που βοηθούν έναν web developer ανάλογα με το αποτέλεσμα που θέλει να επιτύχει. Τα δυο σημαντικότερα στοιχεία για την υλοποίηση ενός web application είναι η server-side programming language (γλώσσα

⁸ http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

⁹ [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))

προγραμματισμού στη πλευρά του εξυπηρετητή) που θα χρησιμοποιηθεί καθώς και η client-side programming language (γλώσσα προγραμματισμού στη πλευρά του πελάτη) για την επεξεργασία των δυναμικών δεδομένων που εμφανίζονται στην ιστοσελίδα και βλέπει ο κάθε χρήστης. Ως server-side programming language επιλέξαμε τη γλώσσα Java και ως client-side επικεντρωθήκαμε στη JavaScript και σε πλαίσια εργασίας (frameworks) που την υιοθετούν και την αναπτύσσουν μέσω plugins. Επιπρόσθετα, άλλες αρχιτεκτονικές που θα αναφέρουμε παρακάτω, μας βοήθησαν να ξεπεράσουμε σημαντικά προβλήματα που παρουσιάστηκαν στην πορεία της υλοποίησης.

Θα περιγράψουμε τις τεχνολογίες που χρησιμοποιήσαμε στη πλευρά του Server (server-side).

5.2.1 Java



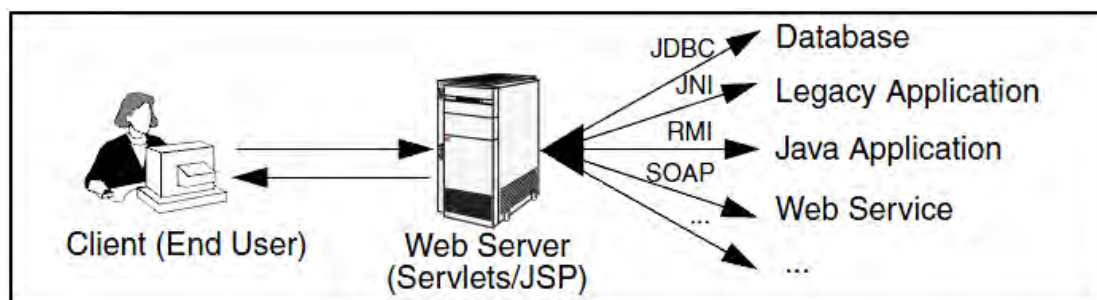
Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού (object-oriented programming language) που σχεδιάστηκε και δημιουργήθηκε από την εταιρεία πληροφορικής Sun Microsystems με επικεφαλής τον James Gosling το 1991. Αρχικά, σχεδιάστηκε για προγραμματισμό οικιακών συσκευών, οι οποίες ελέγχονται από διάφορους τύπους επεξεργαστών. Η ομάδα ανέπτυξε μια διαδικασία μετάφρασης (translation) δύο βημάτων για να απλουστεύσει την εργασία συγγραφής μεταγλωττιστών (compilers) για κάθε διαφορετικό τύπο συσκευής. Ένα από τα βασικότερα στοιχεία της Java σε σχέση με τις άλλες γλώσσες προγραμματισμού είναι η ανεξαρτησία λειτουργικού συστήματος (operating system-OS) και πλατφόρμας (cross-platform). Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux και Macintosh χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compile) ή να αλλάξει ο πηγαίος κώδικας (source code) για κάθε διαφορετικό λειτουργικό σύστημα.

Λόγω της cross-platform υποστήριξης της Java και των πρώτων προγραμμάτων (applet) που σχεδιάστηκαν να εκτελούνται από μια τοποθεσία στο Internet μέσω ενός web browser, η Java έγινε ευρέως γνωστή και υποστηρίχθηκε από ένα μεγάλο ποσοστό προγραμματιστών-developers και εταιρείες πληροφορικής. Στην πορεία έκανε την εμφάνισή της στην ανάπτυξη διαδικτυακών εφαρμογών αναπτύσσοντας την τεχνολογία των Servlets γι' αυτόν το σκοπό. Το

μεγαλύτερο μέρος της υλοποίησης της εφαρμογής γίνεται μέσω των Servlets γράφοντας κώδικα Java και την εξέλιξη αυτών με τεχνολογία JSP (Java Server Pages).

Servlets

Τα Java Servlets είναι μια web τεχνολογία βασισμένη στη Java η οποία καθιστά εφικτή την ανάπτυξη server-side (εκτελούνται στον Server και όχι στον client) εφαρμογών για το Internet. Ένα Java Servlet είναι μια κλάση (class) της Java που επεκτείνει τις δυνατότητες των εξυπηρετητών που φιλοξενούν client-server εφαρμογές υλοποιώντας ένα σχήμα επικοινωνίας request-response. Τα Servlets ουσιαστικά δρουν σαν ένα ενδιάμεσο στρώμα μεταξύ αιτήσεων που έρχονται από web browsers (ή άλλων HTTP clients) και Βάσεων Δεδομένων (Databases) ή εφαρμογές στον HTTP Server. Οι διάφορες λειτουργίες που εκτελεί ένα Servlet φαίνονται στην Εικόνα 5-2.



Εικόνα 5-2: Οι λειτουργίες που εκτελεί ένα Servlet

Ένα Servlet περιέχεται σε έναν container, δηλαδή ένα διερμηνευτή, ο οποίος υποστηρίζει την προδιαγραφή Java Servlet. Ο container διαχειρίζεται το Servlet και μπορεί να είναι ταυτόχρονα και web Server ή ένα add-on component για ένα web Server χωρίς δυνατότητες εκτέλεσης Servlet. Ο Apache Tomcat και συγκεκριμένα η έκδοση 6.0.26 (η τελευταία μέχρι το τέλος της υλοποίησής μας) αποτελεί έναν τέτοιο container και ταυτόχρονα web Server.

Οι λόγοι που μας οδήγησαν να επιλέξουμε την τεχνολογία των Servlets είναι ότι επιτρέπουν την ανάπτυξη πολύ ισχυρών web εφαρμογών που έχουν τα εξής κύρια πλεονεκτήματα:

- Βασίζονται στη γλώσσα προγραμματισμού Java που είναι compiled language (γλώσσα που χρησιμοποιεί μεταφραστή-compiler) και όχι σε μια scripting language (γλώσσα που χρησιμοποιεί διερμηνευτή-interpreter) όπως είναι η γλώσσα προγραμματισμού PHP.
- Μπορούμε με πολύ εύκολο τρόπο να έχουμε πρόσβαση σε Βάσεις Δεδομένων και να πραγματοποιήσουμε τις λειτουργίες διαχείρισης μέσω του JDBC API που προσφέρει η Java. Το JDBC API έχει μεγάλη υποστήριξη και τεράστιο αριθμό online tutorials (οδηγούς εκμάθησης ενός αντικειμένου που προσφέρονται μέσω του Internet).

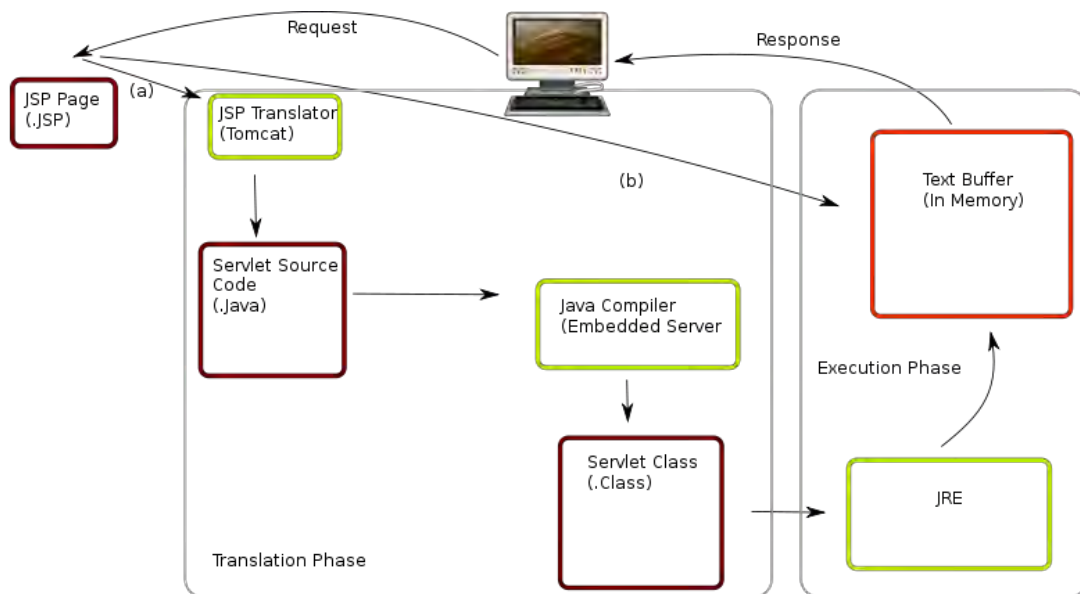
- Υπάρχει ανεξαρτησία Λειτουργικού Συστήματος (cross-platform) αφού τα Servlets εκτελούνται από την Java Virtual Machine (JVM). Η αντίστοιχη τεχνολογία της Microsoft, ASP.NET δεν έχει αυτή τη δυνατότητα με αποτέλεσμα οι εφαρμογές τους να εκτελούνται μόνο σε πλατφόρμες της ίδιας της εταιρείας μέσω του web Server που τις υποστηρίζει IIS (Internet Information Server).

Η υλοποίησή μας έγινε στην έκδοση Servlet 3.0 που παρέχεται μέσω της πλατφόρμας JavaEE 6 (Java Enterprise Edition 6) που χρησιμοποιήσαμε.

JSP, EL, JSTL και Custom JSP Tags

Παρά τα πλεονεκτήματα που προσφέρει η τεχνολογία των Servlets, όταν πρόκειται να ορίσουμε την απόκρισή (response) του (κυρίως σε μορφή HTML) διακρίνουμε ότι είναι δύσκολο να γράψουμε και γενικά να διαχειριστούμε κώδικα HTML. Ακόμα και όταν χρησιμοποιούμε κάποιο IDE (Integrated Development Environment) για την ανάπτυξη της εφαρμογής, όπως είναι το NetBeans 6.9 που επιλέξαμε, η συγγραφή HTML μέσα σε ένα Servlet, με τα προσφερόμενα εργαλεία, δεν είναι εφικτή. Η τεχνολογία των Java Server Pages (JSP) μας επιτρέπει να δημιουργήσουμε δυναμικές ιστοσελίδες περιέχοντας εκτός από δυναμικά στοιχεία και στατική HTML. Η χρήση της JSP τεχνολογίας διευκολύνει τη δημιουργία ιστοσελίδων παραμερίζοντας τα μειονεκτήματα των Servlets. Σε μία γενική μορφή μπορούμε να θεωρήσουμε το JSP σαν σελίδα HTML που περιέχει κώδικα Java και παρέχει όλες τις δυνατότητες των Servlets.

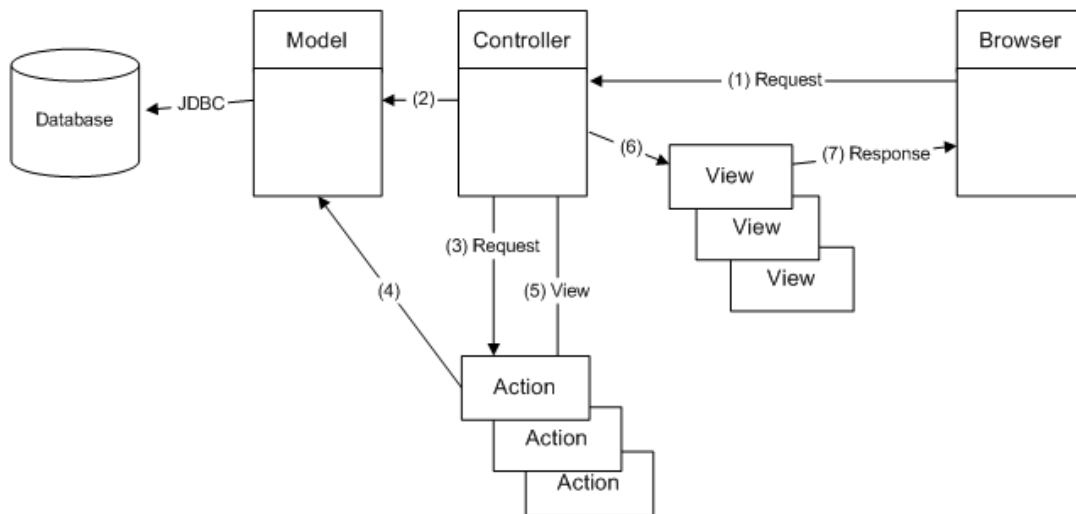
Για την υποστήριξη της τεχνολογίας JSP απαιτείται ένας Servlet container, όπως είναι ο Apache Tomcat που προαναφέραμε, καθώς για να εκτελεστούν οι JSP σελίδες πρέπει πρώτα να μεταφραστούν σε Servlets. Ουσιαστικά γράφοντας JSP είναι σαν να γράφουμε Servlets με άλλο τρόπο. Η διαδικασία μετάφρασης και εκτέλεσης μίας JSP σελίδας φαίνεται στην Εικόνα 5-3.



JSP Container
 (a) Translation occurs at this point if JSP has been changed or is new
 (b) if not, translation is skipped.

Εικόνα 5-3: Διαδικασία μετάφρασης και εκτέλεσης ενός JSP

Τα Servlets είναι καταλληλότερα σε ό,τι αφορά την επεξεργασία δεδομένων, την επικοινωνία και διαχείριση Βάσεων Δεδομένων (μέσω του JDBC API) και γενικότερα σε περιστάσεις όπου απαιτείται η συγγραφή πολλών γραμμών κώδικα Java για την υλοποίηση της επιθυμητής λειτουργικότητας, επικαλώντας έτσι την επιχειρηματική λογική (business-logic). Τα JSP είναι κατάλληλα στην παρουσίαση των δεδομένων και γενικά οτιδήποτε βλέπει ο χρήστης, παράγοντας κώδικα HTML από τις διάφορες αιτήσεις που πραγματοποιούνται. Μπορούμε να χρησιμοποιούμε λοιπόν και τις δύο αρχιτεκτονικές, αντί να χρησιμοποιούμε εξ' ολοκλήρου μία. Η τεχνική αυτή είναι γνωστή ως Model View Controller (MVC) ή Model 2 Architecture. Σύμφωνα με αυτή την τεχνική η αρχική αίτηση-request χειρίζεται από το Servlet. Το Servlet στη συνέχεια επικαλείται την business-logic (κώδικας Java), επικοινωνεί με μία Βάση Δεδομένων, για την πρόσβαση σε δεδομένα, και δημιουργεί beans (classes) για να αναπαραστήσει τα αποτελέσματα εσωτερικά (αυτό αποτελεί το Model στο MVC). Μέσα από το Servlet ορίζουμε την JSP σελίδα που θα αναπαραστήσει αυτά τα αποτελέσματα και προωθούμε την αρχική αίτηση-request σε αυτή (η JSP σελίδα αποτελεί το View στο MVC). Δηλαδή το ίδιο το Servlet αποφασίζει τι business-logic χρειάζεται και ποιά JSP σελίδα θα αναπαραστήσει-εμφανίσει τα αποτελέσματα (το Servlet αποτελεί το Controller στο MVC), όπως παρουσιάζεται και στην Εικόνα 5-4



Εικόνα 5-4: Βήματα του μοντέλου MVC

Η δική μας υλοποίηση βασίζεται εξ' ολοκλήρου στο μοντέλο MVC, γεγονός που καθιστά την ανάπτυξη αισθητά πιο εύκολη και δομημένη. Όταν πρόκειται για επεξεργασία αιτήσεων, σύνδεση με τη Βάση Δεδομένων για λειτουργίες SELECT, INSERT, UPDATE και DELETE και γενικά για συγγραφή κώδικα Java, χρησιμοποιούνται Servlets. Αντίθετα, όταν πρόκειται για απεικόνιση/εμφάνιση των δεδομένων/αποτελεσμάτων χρησιμοποιείται JSP.

Όπως έχει ήδη αναφερθεί, η τεχνολογία JSP αποτιμά και εμφανίζει τιμές και περιεχόμενα Java αντικειμένων (Java Objects). Η Expression Language (EL) και η δυνατότητα να χρησιμοποιούμε προκαθορισμένα JSP Tags (JSP Standard Tag Library) για να επεξεργαζόμαστε τα αποτελέσματα χωρίς τη χρήση κώδικα Java, είναι τα δύο σημαντικότερα χαρακτηριστικά της τεχνολογίας JSP 2.0 που υιοθετήσαμε.

- **JSP Expression Language (EL)**

Για την αναπαράσταση δεδομένων-αποτελεσμάτων που είναι αποθηκευμένα σε beans από τα Servlets, μέσα σε JSP σελίδες, συνήθως χρησιμοποιούμε κατάλληλους οριοθέτες, ειδικές JSP ετικέτες (JSP Tags) και ειδικά JSP elements. Αυτή η προσέγγιση αποτελεί αναλυτική-κουραστική διαδικασία και συνήθως για την επεξεργασία πιο περίπλοκων δεδομένων απαιτείται αρκετή συγγραφή κώδικα Java μέσα στις JSP σελίδες, πράγμα που ουσιαστικά καταργεί το μοντέλο MVC. Για το σκοπό αυτό η JSP 2.0 Expression Language (EL) δίνει τη δυνατότητα απλούστευσης του τρόπου παρουσίασης (presentation layer) των δεδομένων-αποτελεσμάτων, αντικαθιστώντας τα παραπάνω με σύντομες και ευανάγνωστες εκφράσεις-εισαγωγές.

Η δημιουργία JSP σελίδων στην εφαρμογή μας έχει βασιστεί σε αυτή τη λογική, ξεπερνώντας σημαντικά προβλήματα της παλαιότερης τεχνολογίας.

- **JSP Standard Tag Library (JSTL)**

Ακόμα και η Expression Language (EL) δεν είναι αρκετά δυνατή όταν έρχεται η ώρα να υλοποιήσουμε περίπλοκη παρουσίαση (presentation logic) στις JSP σελίδες. Η Expression Language έχει έλλειψη βασικών χαρακτηριστικών, όπως υποστήριξη βρόχων και αποθήκευση μεταβλητών. Για το σκοπό αυτό χρησιμοποιούμε τη τεχνολογία των JSTL (JSP Standard Tag Library), δηλαδή προκαθορισμένα tags με εύκολη ονοματολογία και σύνταξη που κάνουν τη χρήση των παραπάνω χαρακτηριστικών εύκολη χωρίς να απαιτείται εισαγωγή κώδικα Java.

- **Custom JSP Tags**

Στις περισσότερες εφαρμογές μπορούν να χρησιμοποιηθούν τα standard JSP tags, η JSP Expression Language (EL) και η JSP Standard Tag Library (JSTL) έτσι ώστε να ελατωθεί η συγγραφή προγραμματιστικού κώδικα (scripting) στην πλευρά του εξυπηρετητή. Μερικές φορές όμως, αυτές οι τεχνολογίες δεν παρέχουν την απαιτούμενη λειτουργικότητα. Οι developers μπορούν λοιπόν να δημιουργήσουν δικά τους tags (custom tags) για το σκοπό αυτό. Παρόλο που η χρήση JSP scriptlet και expression μπορούν να επιφέρουν το ίδιο αποτέλεσμα, τα custom tags έχουν μερικά πλεονεκτήματα σε σχέση με αυτά. Πρώτον, μειώνουν τη συγγραφή κώδικα Java που απαιτείται στις JSP σελίδες. Δεύτερον, μπορούν να βοηθήσουν στην οργάνωση του κώδικα σε μια εφαρμογή, έτσι ώστε να μειωθεί η περιττή επανάληψη, κάνοντας η διαχείρισή της αισθητά πιο εύκολη.

JSF

Η τεχνολογία Java Server Faces (JSF) είναι ένα request-driven MVC web framework που βασίζεται σε ένα μοντέλο component-driven user interface (UI) που δημιουργήθηκε στις αρχές του 2004 και αποτελεί ουσιαστικά την μετεξέλιξη της JSP τεχνολογίας. Η χρήση των JSF απλουστεύει την ανάπτυξη UI σε web εφαρμογές Java. Οι developers μπορούν γρήγορα να δημιουργήσουν διαδικτυακές εφαρμογές εισάγοντας επαναχρησιμοποιούμενα UI components, συνδέοντας αυτά τα components με μία Βάση Δεδομένων και ενώνοντας γεγονότα (events), που πραγματοποιούνται στην πλευρά του πελάτη, με διαχειριστές γεγονότων (events handlers) στην πλευρά του εξυπηρετητή. Η τελευταία έκδοση των JSF είναι η 2.1 και την χρησιμοποιήσαμε για την υλοποίηση του διαχειριστικού περιβάλλοντος (Administrative Area) της εφαρμογής μας.

JPA

Το Java Persistence API (JPA) είναι ένα framework της γλώσσας προγραμματισμού Java που διαχειρίζεται σχεσιακά δεδομένα (relational data) σε μια εφαρμογή χρησιμοποιώντας διάφορες εκδόσεις της Java (Java Platform, Standard Edition and Java Platform, Enterprise Edition). Οι developers μπορούν με πολύ απλό τρόπο να μεταφέρουν το διαχειριστικό κομμάτι μιας Σχεσιακής Βάσης (MySQL) εξ' ολοκλήρου σε web interface, χρησιμοποιώντας παράλληλα και την τεχνολογία των JSF. Η τελευταία έκδοση του JPA είναι η 2.0 και χρησιμοποιήθηκε σε συνεργασία με την τεχνολογία των JSF στο διαχειριστικό περιβάλλον της εφαρμογής.

5.2.2 MySQL



Η MySQL είναι η πιο γνωστή Βάση Δεδομένων ανοιχτού κώδικα (open source) για τη διαχείριση Σχεσιακών Βάσεων (Relational Database Management System-RDBMS). Χρησιμοποιείται ευρέως από τους από τους web developers γιατί είναι ανέξοδη και εύκολη στη χρήση σε σχέση με αντίστοιχα προϊόντα όπως η Βάση Δεδομένων Oracle ή ο Microsoft SQL Server. Επίσης τρέχει γρήγορα σε σχέση με αυτά τα προϊόντα και μπορεί να εγκατασταθεί στα περισσότερα σύγχρονα συστήματα, ενώ ο Microsoft SQL Server τρέχει μόνο σε συστήματα Windows. Παρόλο που διανέμεται ελεύθερα (με ένα απλό download από το Internet) παρέχει τα περισσότερα χαρακτηριστικά που θα περιμέναμε από ένα σύγχρονο σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Παρέχει υποστήριξη της γλώσσας SQL (Structured Query Language), που αποτελεί το standard στη βιομηχανία των Βάσεων Δεδομένων, υποστήριξη για πολλούς πελάτες (multiple clients) και ασφάλεια.

Στα πλαίσια των διαδικτυακών εφαρμογών, η χρήση της μαζί με τη γλώσσα προγραμματισμού Java αποτελεί εύκολη διαδικασία, μέσω φυσικά και του JDBC API, ενισχύοντας έτσι τους λόγους που μας έκαναν να επιλέξουμε τη MySQL.

Στις επόμενες παραγράφους θα περιγράψουμε τις τεχνολογίες που χρησιμοποιήσαμε στη πλευρά του Client (client-side).

5.2.3 JavaScript

Η JavaScript είναι μια scripting γλώσσα προγραμματισμού, δηλαδή δε χρειάζεται να περάσει από διαδικασία μεταγλώττισης (compile) για να εκτελεστεί και σχεδιάστηκε για να προσθέτει διαδραστικότητα και δυναμικό περιεχόμενο σε HTML σελίδες στην πλευρά του πελάτη. Είναι η πιο διαδεδομένη client-side γλώσσα προγραμματισμού προσθέτοντας την απαραίτητη λειτουργικότητα που θέλουν οι προγραμματιστές στις ιστοσελίδες. Με τη JavaScript μπορούμε να επικυρώσουμε διάφορες φόρμες (validate forms), να αναγνωρίσουμε το φυλλομετρητή (browser) κάθε χρήστη, να αλλάξουμε το χρώμα κάποιου στοιχείου της σελίδας και γενικά να επηρεάσουμε το μοντέλο DOM (Document Object Model) της HTML σελίδας. Η JavaScript υποστηρίζεται από όλους τους browsers και μάλιστα τα τελευταία χρόνια υπάρχει η διαμάχη ποιός θα την υποστηρίξει καλύτερα με την JavaScript Engine που υλοποιεί.

JQuery

Όσο αυξανόταν η χρήση της JavaScript με τα χρόνια, τόσο αυξανόταν η πολυπλοκότητα του κώδικα που χρησιμοποιούσαν οι προγραμματιστές στη πλευρά του πελάτη. Η JQuery είναι ένα από τα πολλά JavaScript frameworks που έχει ως στόχο να απλουστεύσει και να διευκολύνει τον client-side τρόπο προγραμματισμού, ένα χαρακτηριστικό που τονίζεται και στον τρόπο διαφήμισης, «*JQuery: write less, do more*». Λόγω της ευελιξίας που προσφέρει είναι από τα πιο διαδιδόμενα JavaScript frameworks που χρησιμοποιούνται από τους web developers. Πιο συγκεκριμένα, με τη JQuery μπορούμε να προσθέσουμε εκπληκτικά εφέ (effects) στις ιστοσελίδες, με πολύ πιο εύκολο τρόπο από την απλή JavaScript, και να χειριστούμε καλύτερα διάφορες ενέργειες (events) που πραγματοποιούνται κατά την επίδραση πάνω στο DOM της HTML σελίδας.

Τα JQuery Plugins που χρησιμοποιήθηκαν είναι τα εξής:

1. jQuery Ajax Autosuggest: <http://papermashup.com/jquery-php-ajax-autosuggest/>
2. jQuery Corner: <http://jquery.malsup.com/corner/>
3. RapidEdit: <http://plugins.jquery.com/project/rapidedit>
4. Pines Notify jQuery: <http://pines.hg.sourceforge.net/hgweb/pines/pnotify/raw-file/tip/index.html>
5. SimpleModal: <http://www.ericmmartin.com/projects/simplemodal-demos/>
6. Galleria: <http://galleria.aino.se/>
7. jQuery UI: <http://jqueryui.com/>
8. jPlayer: <http://www.happyworm.com/jquery/jplayer/>

AJAX και JSON

Η AJAX δεν είναι μια νέα γλώσσα προγραμματισμού, αλλά μια τεχνολογία που χρησιμοποιεί υπάρχουσες τεχνολογίες όπως JavaScript, XML, HTML και CSS. Προέρχεται από τη συντομογραφία των λέξεων *Asynchronous JavaScript And XML* και χρησιμοποιεί την *ασύγχρονη* μεταφορά δεδομένων μεταξύ της πλευράς του πελάτη και της πλευράς του εξυπηρετητή. Μπορούμε δηλαδή να επικοινωνήσουμε με τον Server και να λάβουμε αποτελέσματα/δεδομένα χωρίς να χρειαστεί να φορτώσουμε (reload) εκ νέου την ίδια ή άλλη σελίδα και γενικά χωρίς να επηρεάσουμε καθόλου τη σελίδα. Πριν από την τεχνολογία AJAX, οποιαδήποτε αίτηση προς το Server αποστέλλονταν με *σύγχρονο* τρόπο, για παράδειγμα μέσω των πεδίων κάποιας φόρμας, δηλαδή η απάντηση σε μια τέτοια αίτηση έπρεπε να παρουσιάζεται σε ξεχωριστή σελίδα. Κυρίαρχο ρόλο στην τεχνολογία AJAX παίζει η γλώσσα JavaScript, η οποία προσθέτει ακόμα περισσότερη ευελιξία στην ασύγχρονη επικοινωνία.

Αρχικά η τεχνολογία AJAX αναπτύχθηκε για να μεταφέρει δεδομένα μεταξύ client και Server σε μορφή XML, εξ' ου και η προέλευση της ονομασίας της. Επειδή όμως οι προγραμματιστές χρειάζονταν μία «ελαφριά» και ευανάγνωστη μορφή δεδομένων υιοθέτησαν τα δεδομένα της μορφής JSON. Το JSON (JavaScript Object Notation) είναι ένα πρότυπο εύκολο στη γραφή και την ανάγνωση από τους ανθρώπους και παράλληλα εύκολο από τα διάφορα συστήματα που το αναλύουν ή το παράγουν, ακόμα και αν η δομή των δεδομένων είναι περίπλοκη. Πρόκειται ουσιαστικά για επικοινωνία με αντικείμενα τύπου JavaScript. Στην εφαρμογή μας η τεχνολογία AJAX χρησιμοποιεί εξ' ολοκλήρου αντικείμενα JSON για την επικοινωνία με τον Server.

5.2.4 HTML 5.0

Η HTML 5.0 είναι μια markup language και εξέλιξη της HTML που δημιουργήθηκε για να απλουστεύσει τη χρήση προχωρημένων τεχνολογιών κατά την υλοποίηση πλούσιων εφαρμογών (Rich Internet Applications-RIA) στο Internet. Προορίζεται να αντικαταστήσει τις υπάρχουσες γλώσσες HTML 4.01 και XHTML 1.0. Με την HTML 5.0 μπορούμε με πολύ εύκολο τρόπο να υποστηρίξουμε βίντεο και ήχο στις ιστοσελίδες καθώς και να τις δομήσουμε με τέτοιον τρόπο ώστε η υλοποίηση να είναι εύκολη και κατανοητή στον προγραμματιστή.

Για την αναπαραγωγή των μουσικών κομματιών στην εφαρμογή μας χρησιμοποιήσαμε στοιχεία (elements) της HTML 5.0 και διάφορα plugins που τη χρησιμοποιούν απλουστεύοντας σε μεγάλο βαθμό την υλοποίηση.

5.2.5 Rest

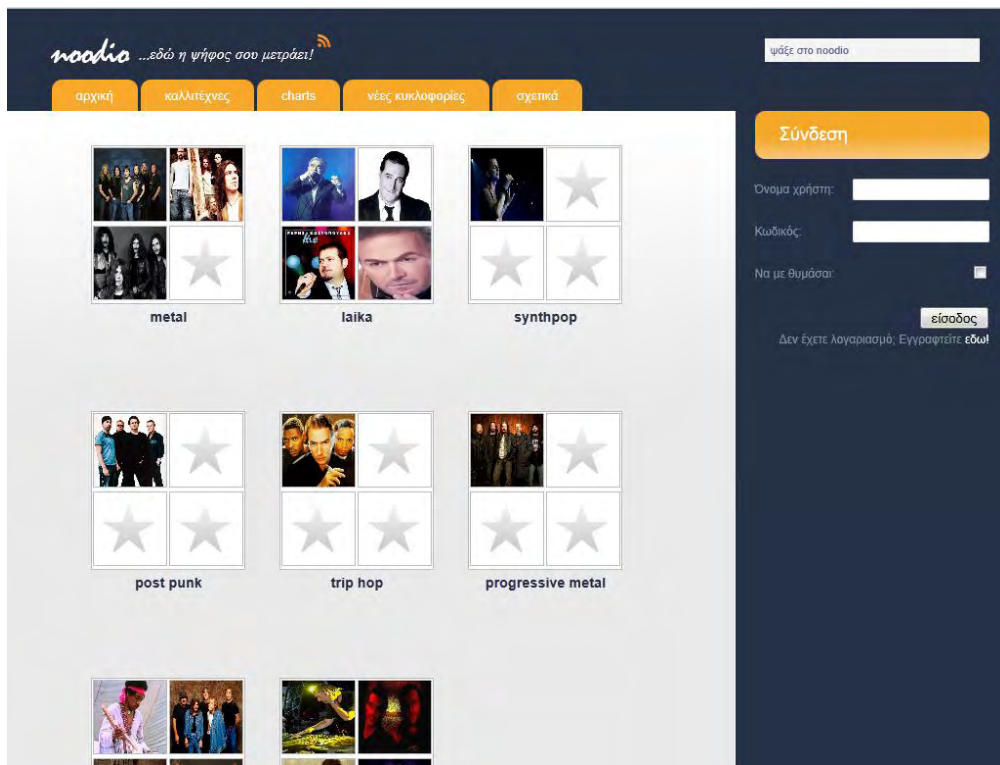
Στον κόσμο των διαδικτυακών υπηρεσιών (web services) η έννοια του REST - REpresentational State Transfer - αποτελεί κλειδί στη σχεδίαση καθώς συμπεριλαμβάνει μη καταστασιακή (stateless) client-server αρχιτεκτονική, στην οποία οι υπηρεσίες θεωρούνται ως πόροι (resources) και μπορούν να αναγνωριστούν από τα URLs τους. Πρόκειται ουσιαστικά για μια αρχιτεκτονική που απευθύνεται σε κατακεντρωμένα υπερμεσικά συστήματα (distributed hypermedia systems) και χρησιμοποιούν οι πελάτες (clients) για να έχουν πρόσβαση σε διάφορους πόρους πάνω από το διαδίκτυο. Οι πελάτες μεταφέρουν περιεχόμενο εφαρμογών χρησιμοποιώντας ένα καθορισμένο σύνολο μεθόδων που περιγράφει τις ενέργειες που θα πραγματοποιηθούν από τον πόρο. Οι εφαρμογές που αναπτύσσονται σύμφωνα με την αρχιτεκτονική REST, ονομάζονται «RESTful». Η χρήση της συγκεκριμένης τεχνολογίας στην εφαρμογή μας, επικεντρώθηκε κυρίως στη δημιουργία ευανάγνωστων και κατανοητών URLs για την καλύτερη πλοήγηση των χρηστών. Η υλοποίηση και επεξεργασία της «RESTful» απεικόνισης των URLs πραγματοποιήθηκε μέσω του `UrlRewriteFilter` (<http://www.tuckey.org/urlrewrite/>).

6

Περιβάλλον χρήσης και σενάρια χρήσης

6.1 Διεπαφή χρήσης

Σε αυτό το κεφάλαιο θα περιγράψουμε το περιβάλλον στο οποίο γίνεται η περιήγηση στην εφαρμογή. Για να υποστηρίξουμε τις λειτουργίες που μπορεί να εκτελέσει ο χρήστης και ο καλλιτέχνης αναπτύξαμε ένα εύχρηστο και κατανοητό περιβάλλον χρήσης. Στην κεντρική σελίδα (index) εμφανίζονται όλοι οι ραδιοφωνικοί σταθμοί που έχει δημιουργήσει το σύστημα και οι επιλογές που προσφέρονται μέσω του οριζόντιου μενού πλοήγησης (horizontal navigation bar). Επίσης, δεξιά εμφανίζεται ένα κάθετο μενού επιλογών (sidebar) για λειτουργίες εγγραφής και εισαγωγής/εξόδου στην εφαρμογή. Το περιβάλλον αυτό φαίνεται στην Εικόνα 5-5.



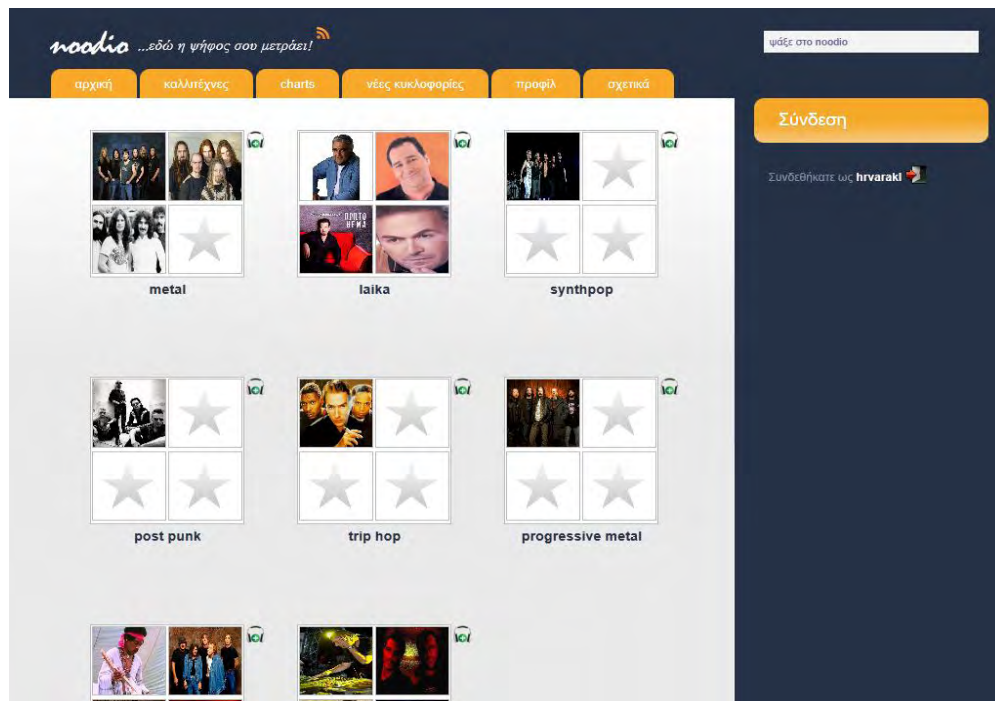
Εικόνα 6-1: Κεντρική σελίδα index

Από το sidebar της κεντρικής σελίδας ο χρήστης μπορεί να κάνει εγγραφή στο σύστημα μέσω της λειτουργίας «Εγγραφείτε εδώ!» και της σελίδας εγγραφής που εμφανίζεται στη συνέχεια όπως φαίνεται στην Εικόνα 5-6. Σε αυτή τη σελίδα ο χρήστης μπορεί να επιλέξει αν θα είναι καλλιτέχνης ή απλός χρήστης στο σύστημα. Επιπλέον, με την ενσωμάτωση της υπηρεσίας Recaptcha¹⁵ μπορούμε να αποτρέψουμε μηχανές bots¹⁰ από το να συμπληρώσουν αυτόματα τα στοιχεία.

¹⁰ Λογισμικό που εκτελεί αυτοματοποιημένες ενέργειες πάνω από τον παγκόσμιο ιστό. Ορισμός: http://en.wikipedia.org/wiki/Internet_bot

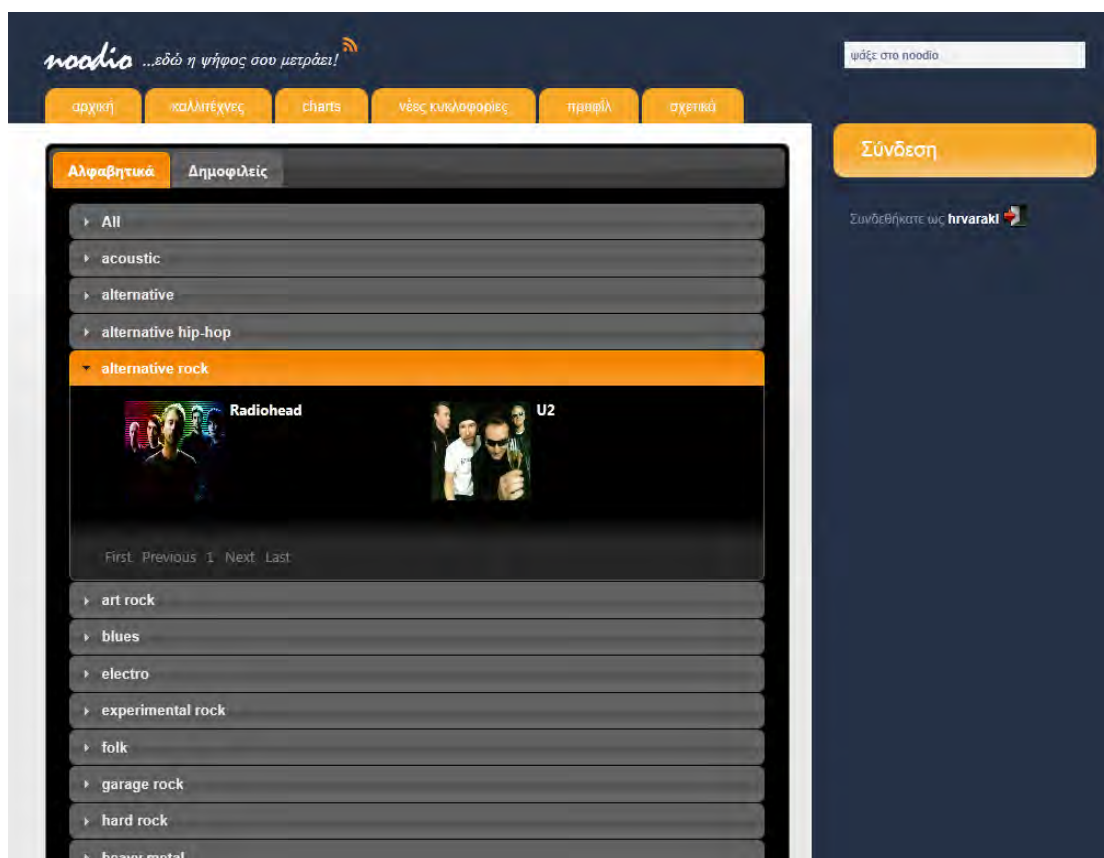
Εικόνα 6-2: Σελίδα εγγραφής στο σύστημα

Όταν ο χρήστης εισέλθει στην εφαρμογή ύστερα από εγγραφή (register) ή εισαγωγή (log in) εμφανίζεται ως κεντρική σελίδα η home όπου το περιεχόμενο είναι ίδιο με εκείνο της index, όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 6-3: Κεντρική σελίδα home

Από το κεντρικό μενού πλοήγησης υπάρχει η επιλογή να δούμε όλους τους καλλιτέχνες που βρίσκονται στο σύστημα μέσω της καρτέλας «καλλιτέχνες», όπως φαίνεται στην εικόνα:



Εικόνα 6-4: Καρτέλα «καλλιτέχνες»

Η καρτέλα «charts» παρουσιάζει τους καλλιτέχνες, τα άλμπουμ και τα τραγούδια που έχουν συγκεντρώσει τις περισσότερες θετικές ψήφους ταξινομημένα ανά είδος (genre). Η καρτέλα «charts» εμφανίζεται στην Εικόνα 5-9.

noadio ...εδώ η ψήφος σου μετράει!

ψάξε στο noadio

αρχική καλλιτέχνες charts νέες κυκλοφορίες προφίλ σχετικά

Σύνδεση

Συνδεθήκατε ως hrvaraki

All house laika metal post punk progressive metal rock synthpop

trip hop

Τραγούδια:

1. Comfortably Numb - Pink Floyd
2. EXP - Jimi Hendrix
3. Squealer - ACDC
4. Little Lover - ACDC
5. Rock N' Roll Train - ACDC
6. Dogs - Pink Floyd
7. Another Time, Another Place - U2
8. Phantom Lord - Metallica
9. With or Without You - U2
10. Shine On You Crazy Diamond (Part One) - Pink Floyd

Καλλιτέχνες:

1. Jimi Hendrix
2. ACDC
3. The Black Keys
4. Scorpions
5. Phil Collins
6. Pink Floyd
7. Deep Purple
8. Linkin Park
9. Radiohead
10. The Dead Weather

Άλμπουμ:

1. High Voltage - ACDC
2. Blackout - Scorpions
3. Love at First Sting - Scorpions
4. Lovedrive - Scorpions
5. 01011001 (Disc 1 - Y) - Ayreon
6. Stiff Upper Lip - ACDC
7. Come Taste the Band - Deep Purple
8. Powerage - ACDC
9. Deep Purple - Deep Purple
10. Back in Black - ACDC

Εικόνα 6-9: Καρτέλα «charts»

Μέσω της καρτέλας «νέες κυκλοφορίες» ο χρήστης έχει τη δυνατότητα να αναζητήσει νέους καλλιτέχνες, νέα άλμπουμ και νέα τραγούδια από κάθε είδος που έχουν εισέλθει στο σύστημα τις τελευταίες ώρες (Εικόνα 5-10):

noadio ...εδώ η ψήφος σου μετράει!

ψάξε στο noadio

αρχική καλλιτέχνες charts νέες κυκλοφορίες προφίλ σχετικά

house laika metal post punk progressive metal rock synthpop

trip hop

Τραγουδία:

1. R2 Units Gone Wild (G.Pal Instrumental vs Break) - G Pal
2. Sidewalk Lovin' (G.Pal's First Afterhours Mix) - G Pal
3. Sonido Profundo (G.Pal's Sweet AMX Remix) - G Pal
4. West Of Athens - G Pal
5. Celluloid Hypnotic (G.Pal's Hypnotic Mix) - G Pal
6. Alive (G.Pal's Strings Of Life Mix) - G Pal
7. Simulation Tribal - G Pal
8. 3rd Of July - G Pal
9. Within (Original Mix) - G Pal
10. After Life (Radio Edit) - G Pal

Καλλιτέχνες:

1. Oliver S
2. Ayreon
3. Scorpions
4. ACDC
5. Pavlos
6. Radiohead
7. Deep Purple
8. Νίκος Παπάζογλου
9. Iron Maiden
10. U2

Άλμπουμ:

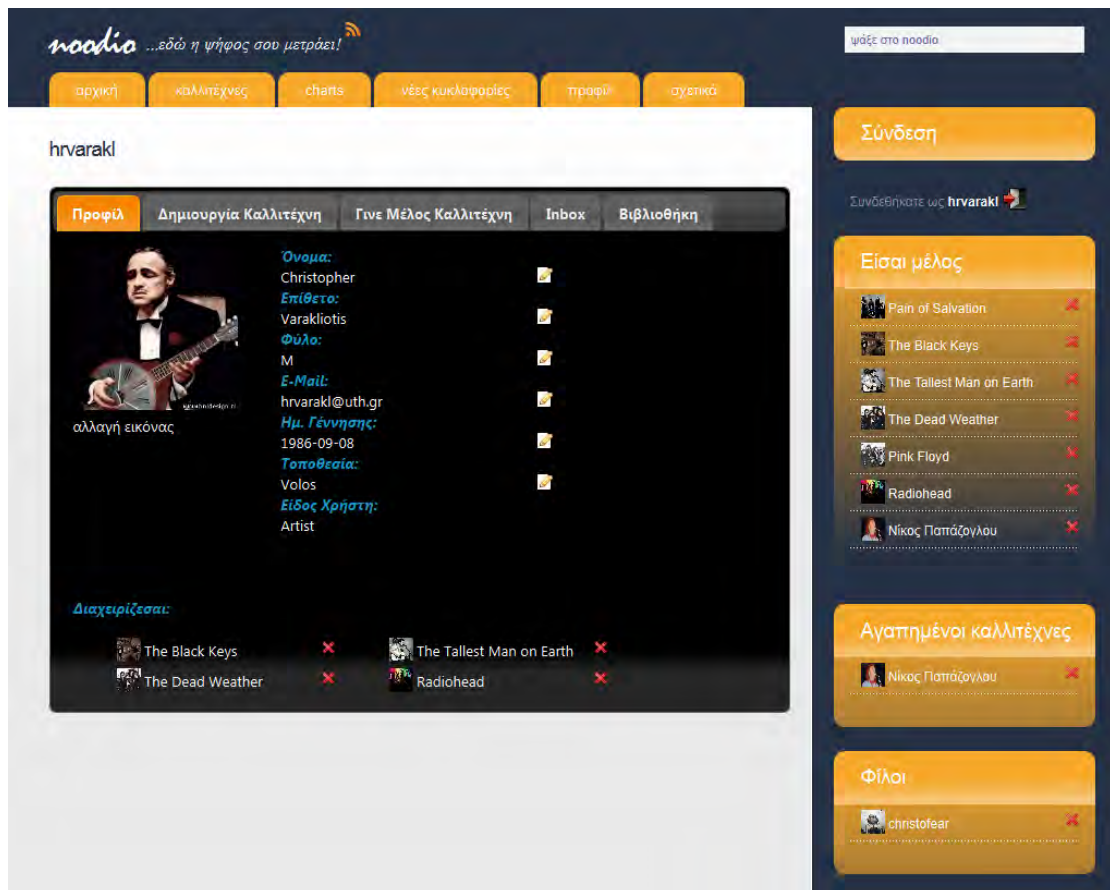
1. Best - G Pal
2. dummy - G Pal
3. Scratch My Back - Peter Gabriel
4. Up - Peter Gabriel
5. Us - Peter Gabriel
6. So - Peter Gabriel
7. Peter Gabriel 4: Security - Peter Gabriel
8. Peter Gabriel 3: Melt - Peter Gabriel
9. Peter Gabriel 2: Scratch - Peter Gabriel
10. Peter Gabriel 1: Car - Peter Gabriel

Σύνδεση

Συνδεθήκατε ως hrvaraki

Εικόνα 6-10: Καρτέλα «νέες κυκλοφορίες»

Ο χρήστης έχει τη δυνατότητα να μεταφερθεί στο προσωπικό του προφίλ είτε μέσω της καρτέλας «προφίλ» του κεντρικού μενού είτε από το δεξιό sidebar κάνοντας κλικ στο username του. Στη σελίδα του προφίλ ο χρήστης μπορεί να δει τα στοιχεία που εισήγαγε στη διαδικασία εγγραφής (register), έχει τη δυνατότητα να τα τροποποιήσει, να δει και να προσπελάσει τους καλλιτέχνες που έχει δημιουργήσει και άλλες λειτουργίες που έχουν περιγραφεί σε προηγούμενες ενότητες (Εικόνα 5-11).



Εικόνα 6-5: Σελίδα προφίλ του χρήστη

Από τις σημαντικότερες σελίδες της εφαρμογής μας, ως προς τις λειτουργίες που προσφέρει, είναι η προσωπική σελίδα του καλλιτέχνη. Στη σελίδα του καλλιτέχνη ο χρήστης μπορεί να δει τα στοιχεία, τη βιογραφία, τα άλμπουμ και να ακούσει μουσική παρόμοια με αυτή που χαρακτηρίζει τον καλλιτέχνη. Αν ο χρήστης εισέλθει σε μία σελίδα καλλιτέχνη και ο ίδιος δεν σχετίζεται με αυτόν βλέπει τη σελίδα που παρουσιάζεται στην Εικόνα 5-12.

The image shows a screenshot of the Noadio website's profile page for Tom Waits. At the top, there is a navigation bar with tabs for 'αρχική', 'καλλιτέχνες', 'charts', 'νέες εκκλιπαριές', 'προφίλ', and 'σχετικά'. A search bar on the right contains the text 'ψάξε στο noadio'. Below the navigation, the profile is titled 'Tom Waits (1979)'. The main content area features a 'Προφίλ' tab, a large album cover for 'BIG TIME', and a bio section. The bio describes Waits as an American singer-songwriter, composer, and actor, born December 7, 1949, in Sonoma County, California. It mentions his distinctive voice and his work in music and film. Below the bio, there are sections for 'Άλμπουμ:' listing 'dummy' and 'Big Time' with their release dates. A sidebar on the right contains buttons for 'Σύνδεση', 'Ευνηθέηστε ως hrvarakl', 'Διαχειριστής', and 'Μέλη' with user avatars.

Εικόνα 6-6: Προσωπική σελίδα καλλιτέχνη όταν ο χρήστης δεν σχετίζεται με αυτόν

Αν ο χρήστης σχετίζεται με έναν καλλιτέχνη και εισέλθει στην προσωπική του σελίδα, τότε έχει τη δυνατότητα πρόσθετων λειτουργιών, όπως να αλλάξει τα στοιχεία του, να δημιουργήσει άλμπουμ και ανεβάσει τραγούδια και εικόνες. Η πρόσθετη αυτή λειτουργικότητα φαίνεται στην Εικόνα 5-13.

The screenshot shows the 'noahio' website interface. At the top, there's a navigation bar with tabs: αρχική, καλλιτέχνες, charts, νέες κυκλοφορίες, προφίλ, and σχετικά. A search bar is on the right. The main content area is titled 'Pain of Salvation (1991)'. It features a 'Προφίλ' tab and a 'Profile' section with a band photo and details: Τοποθεσία: Eskilstuna, Sweden; Τυπός: Group; Ψευδώνυμα: Pain Of Salvation. Below this is a 'Πληροφορίες' section with a detailed description of the band as a Swedish progressive metal band. A 'Δίσκοι' section lists several albums with their release dates: dummy (1997-08-22), Entropia (1997-08-22), One Hour by the Concrete Lake (1998-07-22), The Perfect Element, Part I (2000-10-31), Remedy Lane (2002-02-22), BE (2004-09-22), Scarsick (2007-01-22), and Road Salt One (2010-05-17). A sidebar on the right contains buttons for 'Σύνδεση', 'Διαχειριστής' (Χριστόφορος Δημητριάδης), and 'Μέλη' (Christopher Varakiotis).

Εικόνα 6-7: Προσωπική σελίδα καλλιτέχνη όταν ο χρήστης σχετίζεται με αυτόν

Ο χρήστης έχει τη δυνατότητα να επισκεφθεί τη σελίδα και να δει πληροφορίες ενός συγκεκριμένου άλμπουμ οποιουδήποτε καλλιτέχνη. Εάν ο χρήστης δε σχετίζεται με τον καλλιτέχνη τότε μπορεί να δει την περιγραφή του άλμπουμ, να ακούσει τα τραγούδια που ο καλλιτέχνης έχει ανεβάσει σε αυτή τη συλλογή, καθένα ξεχωριστά άλλα και όλα μαζί σε ξεχωριστή σελίδα, και να μοιραστεί τη σελίδα του άλμπουμ με τους φίλους του σε όποιο κοινωνικό δίκτυο επιθυμεί. Επίσης, ο χρήστης έχει την ευκαιρία να ψηφίσει και να

χαρακτηρίσει όποιο κομμάτι του άλμπουμ θέλει. Η σελίδα του άλμπουμ φαίνεται στην Εικόνα 5-14.

The screenshot shows the NooDio website interface. At the top, there is a navigation bar with buttons for 'αρχική', 'καλλιτέχνες', 'charts', 'νέες κυκλοφορίες', 'προφίλ', and 'σχετικά'. A search bar is located on the right. Below the navigation, the main content area displays the album 'Tom Waits - Rain dogs'. It features the album cover, a play button, and the text 'Ακούστε το άλμπουμ'. Below this, it shows the release date 'Κυκλοφορία: 1999-05-05' and the genre 'Είδος μουσικής: blues, rock'. A table lists the tracks with their durations and genres. On the right side of the album page, there are buttons for 'Σύνδεση', 'Συνδεθήκατε ως hrvaraki', and a set of buttons for 'καρκτηρισμός' (love | hate, μοιράσου, ενέργειες).

	Τραγούδι	Διάρκεια	Αδεια
01.	Singapore	2:45	Ελεύθερο
02.	Clap Hands	3:48	Ελεύθερο
03.	Cemetery Polka	1:46	Ελεύθερο
04.	Jockey Full Of Bourbon	2:47	Ελεύθερο
05.	Tango Till They're Sore	2:51	Ελεύθερο
06.	Big Black Mariah	2:43	Ελεύθερο
07.	Diamonds And Gold	2:32	Ελεύθερο
08.	Hang Down Your Head	2:33	Ελεύθερο
09.	Time	3:56	Ελεύθερο
10.	Rain Dogs	2:57	Ελεύθερο
11.	Midtown (Instrumental)	1:03	Ελεύθερο
12.	9th and Hennepin	1:56	Ελεύθερο

Εικόνα 6-8: Σελίδα του άλμπουμ όταν ο χρήστης δεν σχετίζεται με τον καλλιτέχνη

Στην περίπτωση που ο χρήστης σχετίζεται με τον καλλιτέχνη, τότε υπάρχουν πρόσθετες λειτουργίες στη σελίδα του άλμπουμ. Ο χρήστης δηλαδή μπορεί να διαγράψει τη συλλογή ή κάποιο τραγούδι που αυτή περιέχει και να καθορίσει εάν τα τραγούδια είναι «Ελεύθερα» ή «Επί Πληρωμή». Οι λειτουργίες αυτές φαίνονται στην εικόνα 5-15.

The Black Keys - Brothers

Αλλαγή εξωφύλλου

Ακούστε το άλμπουμ

Κυκλοφορία: 2010-05-18

Είδος μουσικής: rock

Τραγούδι	Διάρκεια	Άδεια
01. Everlasting Light	3:23	Ελεύθερο
02. Next Girl	3:18	Ελεύθερο
03. Tighten Up	3:31	Ελεύθερο
04. Howlin' For You	3:11	Ελεύθερο
05. She's Long Gone	3:06	Ελεύθερο
06. Black Mud	2:09	Ελεύθερο
07. The Only One	5:00	Ελεύθερο
08. Too Afraid to Love You	3:24	Ελεύθερο
09. Ten Cent Pistol	4:29	Ελεύθερο
10. Sinister Kid	3:44	Ελεύθερο

Περιγραφή:
Enter album description here.

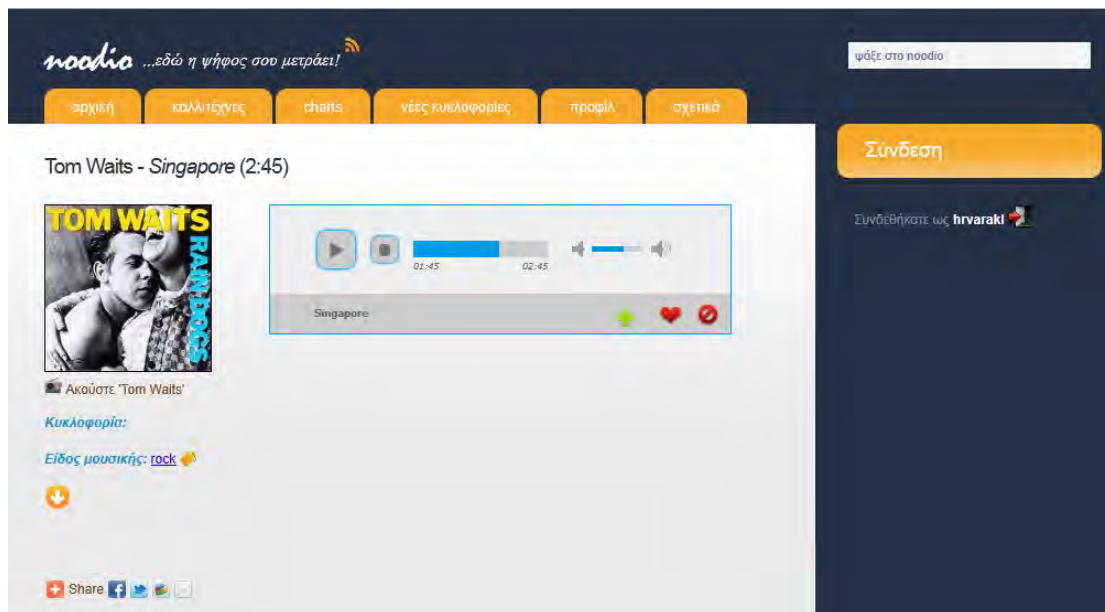
Σύνδεση

Συνδεθήκατε ως hrvaraki

διόρθωση
διαγραφή
επί πληρωμή
καρκτηρισμός
love | hate
μοιράσου
ενέργειες

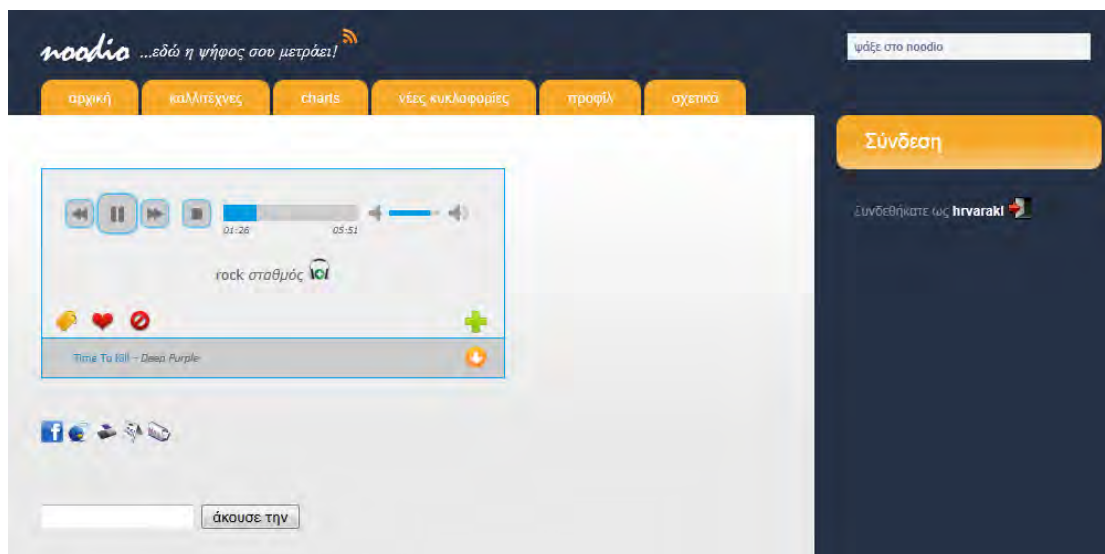
Εικόνα 6-9: Σελίδα του άλμπουμ όταν ο χρήστης σχετίζεται με τον καλλιτέχνη

Ο χρήστης μπορεί να ανοίξει τη σελίδα ενός τραγουδιού. Σε αυτή την περίπτωση μπορεί να ακούσει το συγκεκριμένο κομμάτι, να το ψηφίσει, να το χαρακτηρίσει και να μοιραστεί τη σελίδα αυτή σε κάποιο κοινωνικό δίκτυο. Επιπλέον, αν το τραγούδι έχει οριστεί ως «Ελεύθερο», ο χρήστης μπορεί να το κατεβάσει, αλλιώς έχει τη δυνατότητα εάν θέλει να το αγοράσει. Τέλος ο χρήστης μπορεί να ακούσει στο ραδιόφωνο μουσική παρόμοια με αυτή που χαρακτηρίζει τον καλλιτέχνη στον οποίο ανήκει το συγκεκριμένο κομμάτι (Εικόνα 5-16).



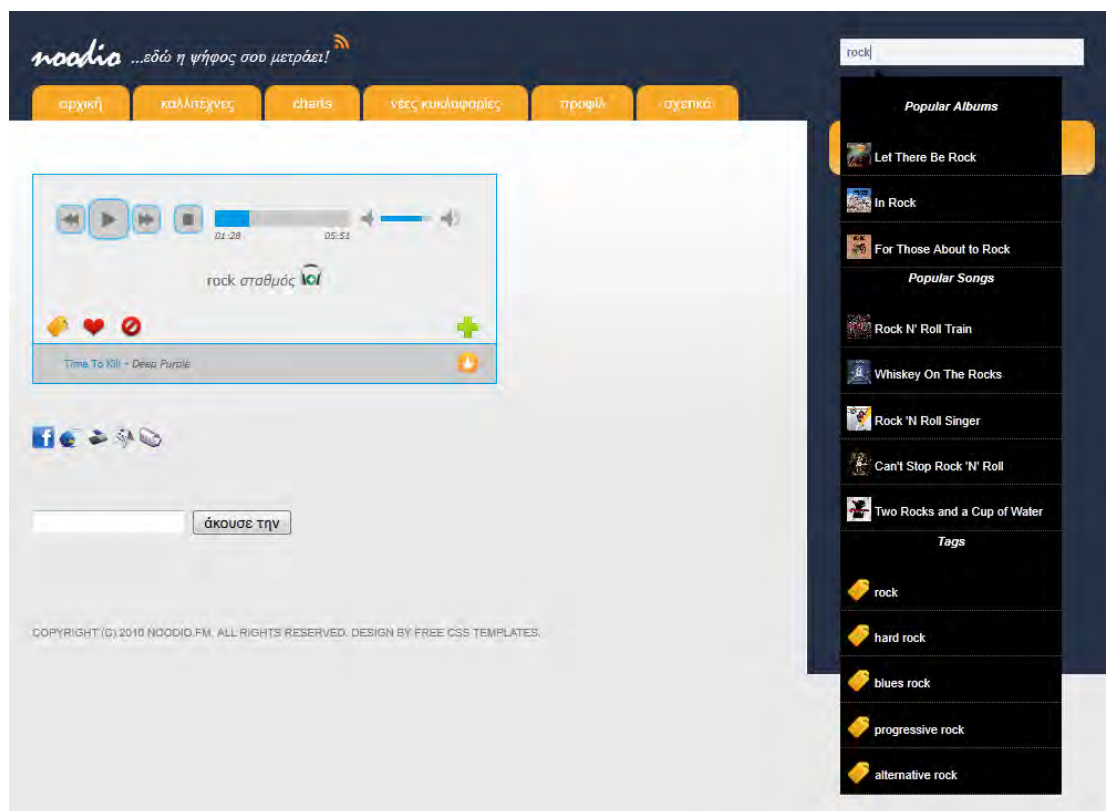
Εικόνα 6-10: Σελίδα τραγουδιού

Η σημαντικότερη σελίδα της διαδικτυακής μας εφαρμογής είναι η σελίδα αναπαραγωγής της λίστας του κάθε ραδιοφωνικού σταθμού. Σε αυτή τη σελίδα ο χρήστης ακούει τη λίστα των τραγουδιών, έχοντας τη δυνατότητα να ψηφίσει και να χαρακτηρίσει το κάθε κομμάτι. Μπορεί να αγοράσει ή να κατεβάσει το τραγούδι που ακούει ανάλογα με τον τρόπο που ο καλλιτέχνης το διανέμει. Επίσης έχει την επιλογή να μοιραστεί τη σελίδα του τραγουδιού που ακούει εκείνη τη στιγμή σε κοινωνικά δίκτυα όπως το *Facebook*¹⁶, *Twitter*¹⁷, *Delicious*¹⁸, *Digg*¹⁹ και *Google Buzz*²⁰ καθώς και μέσω e-mail. Τέλος μπορεί να αναζητήσει και να ακούσει ένα σταθμό της επιλογής του (Εικόνα 5-17).



Εικόνα 6-11: Σελίδα ραδιοφωνικού σταθμού

Τέλος παρατηρούμε ότι σε κάθε σελίδα της εφαρμογής μας υπάρχει η δυνατότητα γενικής αναζήτησης πληροφορίας μέσω του πεδίου «ψάξε στο ποοδίο» που εμφανίζεται πάντα επάνω δεξιά. Σε αυτό το πεδίο, κάθε φορά που ο χρήστης πατάει κάποιο πλήκτρο, το σύστημα προτείνει αυτόματα οτιδήποτε (καλλιτέχνης, άλμπουμ, τραγούδι, είδος μουσικής) ταιριάζει με αυτή την εισαγωγή όπως φαίνεται και στην Εικόνα 5-18.

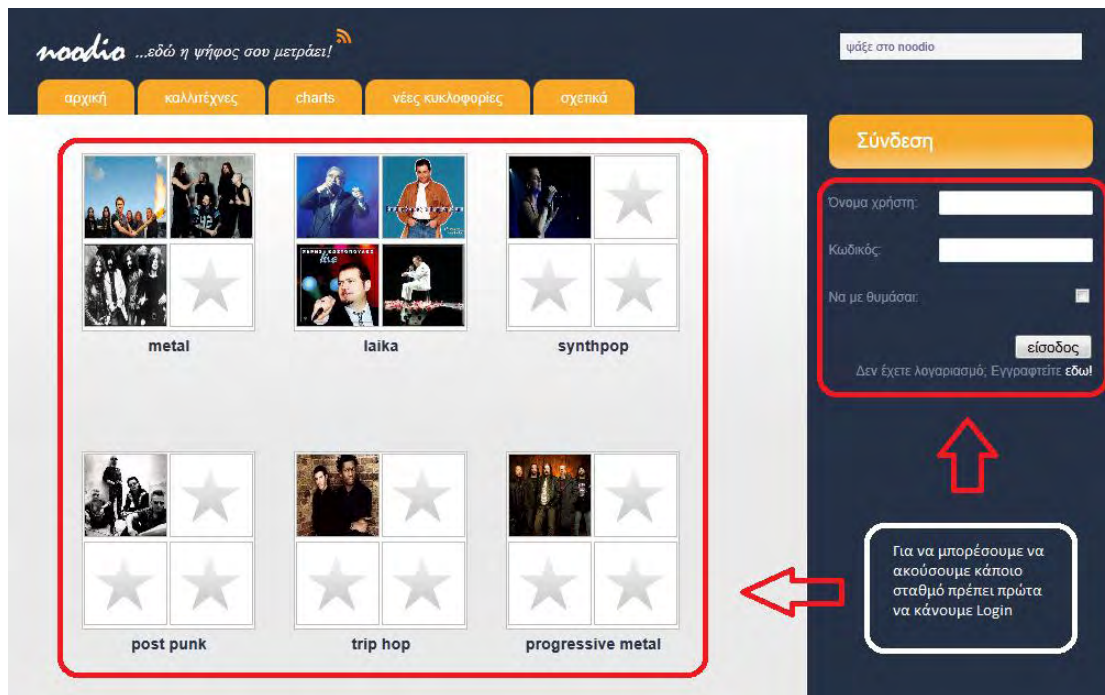


Εικόνα 6-12: Γενική αναζήτηση πληροφορίας

6.2 Σενάριο χρήσης

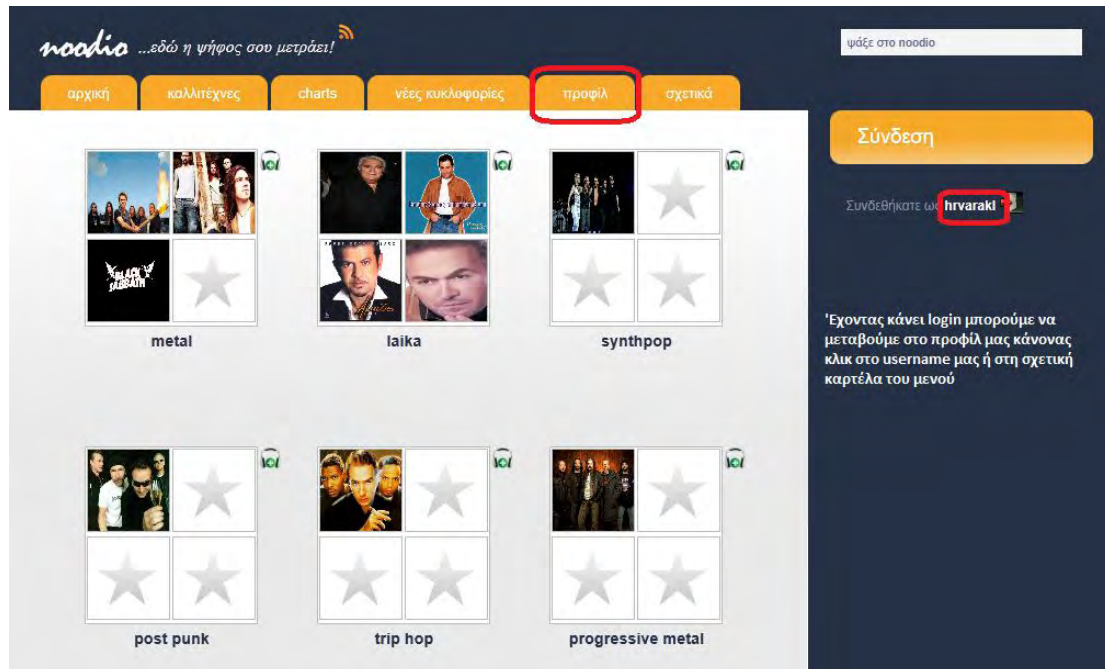
Στην ενότητα αυτή θα παρουσιάσουμε ένα ενδεικτικό σενάριο χρήσης της πλατφόρμας που δημιουργήσαμε.

Βρισκόμενοι στην αρχική σελίδα της εφαρμογής, μπορούμε να συνδεθούμε στο σύστημα (login) από τη φόρμα που υπάρχει στο δεξί μέρος του παραθύρου.



Εικόνα 6-13: Η αρχική σελίδα πριν τη σύνδεση στο σύστημα (login)

Έχοντας κάνει σύνδεση στο σύστημα ως χρήστες, μεταβαίνουμε στη σελίδα του προφίλ μας από όπου μπορούμε να δημιουργήσουμε ένα νέο καλλιτέχνη.



Εικόνα 6-14: Η αρχική σελίδα μετά τη σύνδεση στο σύστημα (login)

The screenshot shows the NooDio website interface. At the top, there's a navigation bar with buttons for 'αρχική', 'καλλιτέχνες', 'charts', 'νέες κυκλοφορίες', 'προφίλ', and 'σχετικά'. The main content area displays the profile for user 'hvaraki'. The profile is divided into several sections: 'Προφίλ' (Profile) with a photo and personal details, 'Δημιουργία Καλλιτέχνη' (Create Artist), 'Γίνε Μέλος Καλλιτέχνη' (Become Artist), 'Inbox', and 'Βιβλιοθήκη' (Library). The profile details include:

- Όνομα: Christopher
- Επίθετο: Varakliotis
- Φύλο: M
- E-Mail: hvaraki@uth.gr
- Ημ. Γεννησης: 1986-09-08
- Τοποθεσία: Volos
- Είδος Χρήστη: Artist

 Below the details is a section for 'Διαχειρίζεσαι:' (Manage) with a grid of artists to follow: The Black Keys, The Dead Weather, The Tallest Man on Earth, and Radiohead. On the right sidebar, there's a 'Σύνδεση' (Login) button, a 'Συνδεθήκατε ως hvaraki' notification, and a 'Είσαι μέλος' (You are a member) section listing bands like Pain of Salvation, The Black Keys, The Tallest Man on Earth, The Dead Weather, Pink Floyd, and Radiohead. At the bottom of the sidebar is a 'Αγαπημένοι καλλιτέχνες' (Favorite artists) section listing Νίκος Παπάζογλου.

Εικόνα 6-15: Το προφίλ του χρήστη

Κάνουμε κλικ στην καρτέλα «Δημιουργία Καλλιτέχνη» και στη φόρμα που παρουσιάζεται συμπληρώνουμε τα στοιχεία για τον καλλιτέχνη που θέλουμε να δημιουργήσουμε. Στο πεδίο «Ψευδώνυμο» μπορούμε να προσθέσουμε συνώνυμα για τον καλλιτέχνη ώστε να διευκολύνουμε την αναζήτηση, μία ενέργεια που συνιστάται κυρίως εάν το όνομα του καλλιτέχνη διαθέτει ειδικά σύμβολα και αριθμούς. Στο πεδίο «Είδος μουσικής» μπορούμε να προσθέσουμε ετικέτες που περιγράφουν το είδος της μουσικής του καλλιτέχνη μας. Δεν υπάρχει περιορισμός στο πλήθος των ετικετών που μπορούμε να χρησιμοποιήσουμε, αρκεί να τις διαχωρίζουμε με κόμματα. Έχοντας ολοκληρώσει τη συμπλήρωση όλων των απαραίτητων στοιχείων κάνουμε κλικ στο κουμπί «Δημιουργία».

The screenshot shows the Noadio website interface. At the top, there's a navigation bar with buttons for 'αρχική', 'καλλιτέχνες', 'charts', 'νέες κηλοφορίες', 'προφίλ', and 'σχετικά'. The main content area is for the user 'hrvaraki', with tabs for 'Προφίλ', 'Δημιουργία Καλλιτέχνη', 'Γίνε Μέλος Καλλιτέχνη', 'Inbox', and 'Βιβλιοθήκη'. The 'Δημιουργία Καλλιτέχνη' form is active, containing the following information:

- Όνομα: The Heavy
- Έτος ίδρυσης: 2007
- Τοποθεσία: Bath, England
- Τύπος: Σόλο Καριέρα, Συγκρότημα
- Περιγραφή: The Heavy are a soul-influenced indie rock band which formed in Bath, England in 2007. The band consists of Kelvin Swaby (vocals), Dan Taylor (guitar), Spencer Page (bass), Chris Elly (drums) and formerly, Hannah Collins (keyboards). According to Songfacts, when The Heavy performed How Do You Like Me Now? on Late Night With David Letterman, they so impressed the host that he asked the band to play some more - the first time that's ever happened on the show.
- Ψευδώνυμα: (empty field)
- Είδος μουσικής: funk, indie, soul, rock, blues
- Θέλεις να είσαι μέλος του καλλιτέχνη; ναι, όχι

At the bottom of the form is a 'Δημιουργία' button. The right sidebar features a 'Σύνδεση' button, a 'Συνδεθήκατε ως hrvaraki' notification, an 'Είσαι μέλος' section listing artists like Pain of Salvation, The Black Keys, The Tallest Man on Earth, The Dead Weather, Pink Floyd, Radiohead, and Νίκος Παπάζογλου, and an 'Αγαπημένοι καλλιτέχνες' section listing Νίκος Παπάζογλου. There is also a 'Φίλοι' section with a user named christofear.

Εικόνα 6-16: Προσθήκη νέου καλλιτέχνη

Μετά τη δημιουργία του νέου καλλιτέχνη μεταβαίνουμε στη σελίδα που αντιστοιχεί στο προφίλ του. Εφόσον ορισθήκαμε μέλη του καλλιτέχνη στην προηγούμενη φόρμα, έχουμε τη δυνατότητα να ανεβάσουμε το πρώτο του άλμπουμ. Να σημειωθεί ότι με τη δημιουργία κάθε νέου καλλιτέχνη, δημιουργείται αυτόματα ένα ειδικής μορφής άλμπουμ -με όνομα *dummy*- που προορίζεται να φιλοξενεί όλα τα κομμάτια του καλλιτέχνη που δεν ανήκουν σε κάποιο άλμπουμ.

The Heavy (2007)

Προφίλ Προσθήκη Άλμπουμ Ανεβάστε Τραγούδια Ανεβάστε Εικόνες

Τοποθεσία:
Bath, England

Τυπός:
Group

Ψευδώνυμο:
The Heavy

Ακούστε 'The Heavy'

Είδος μουσικής: δεν έχει καταχωρηθεί

Προσθήκη ως αγαπημένου

Πληροφορίες:
The Heavy are a soul-influenced indie rock band which formed in Bath, England in 2007. The band consists of Kelvin Swaby (vocals), Dan Taylor (guitar), Spencer Page (bass), Chris Ellul (drums) and formerly, Hannah Collins (keyboards). According to Songfacts, when The Heavy performed How Do You Like Me Now? on Late Night With David Letterman, they so impressed the host that he asked the band to play some more - the first time that's ever happened on the show.

Άλμπουμ:
dummy

Share Like

ψάξε στο podio

Σύνδεση

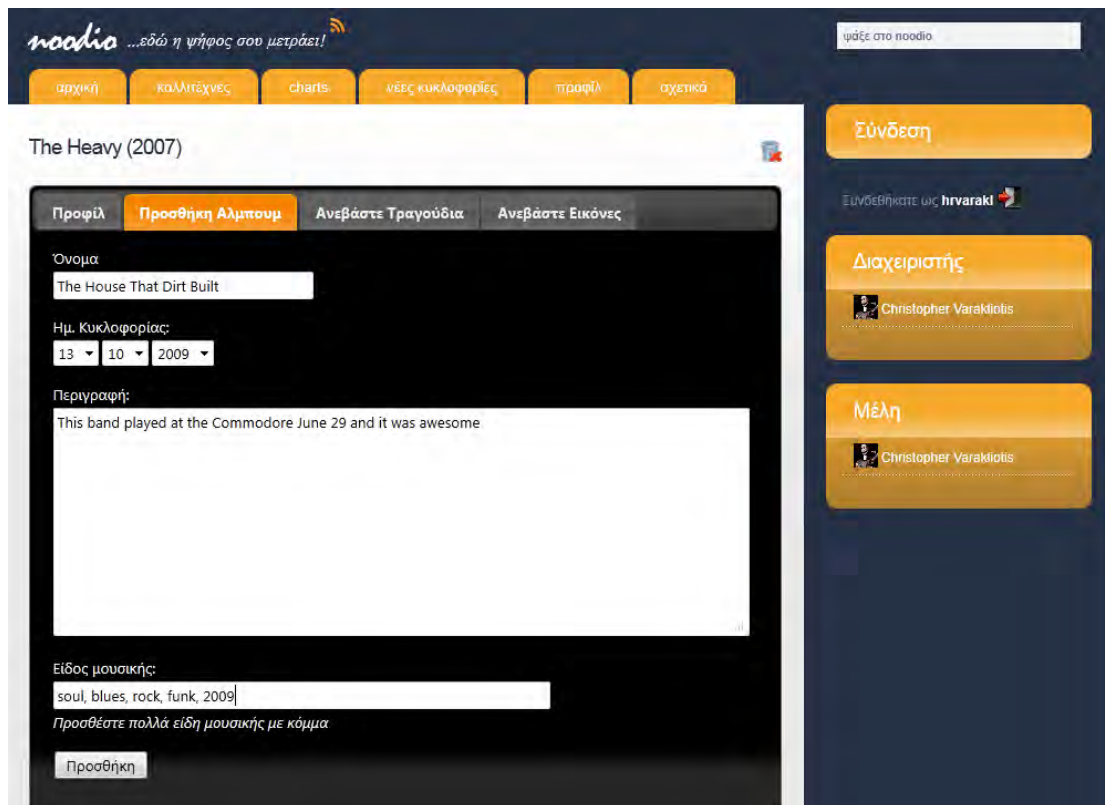
Συνδεθήκατε ως **hvaraki**

Διαχειριστής
Christopher Varakliotis

Μέλη
Christopher Varakliotis

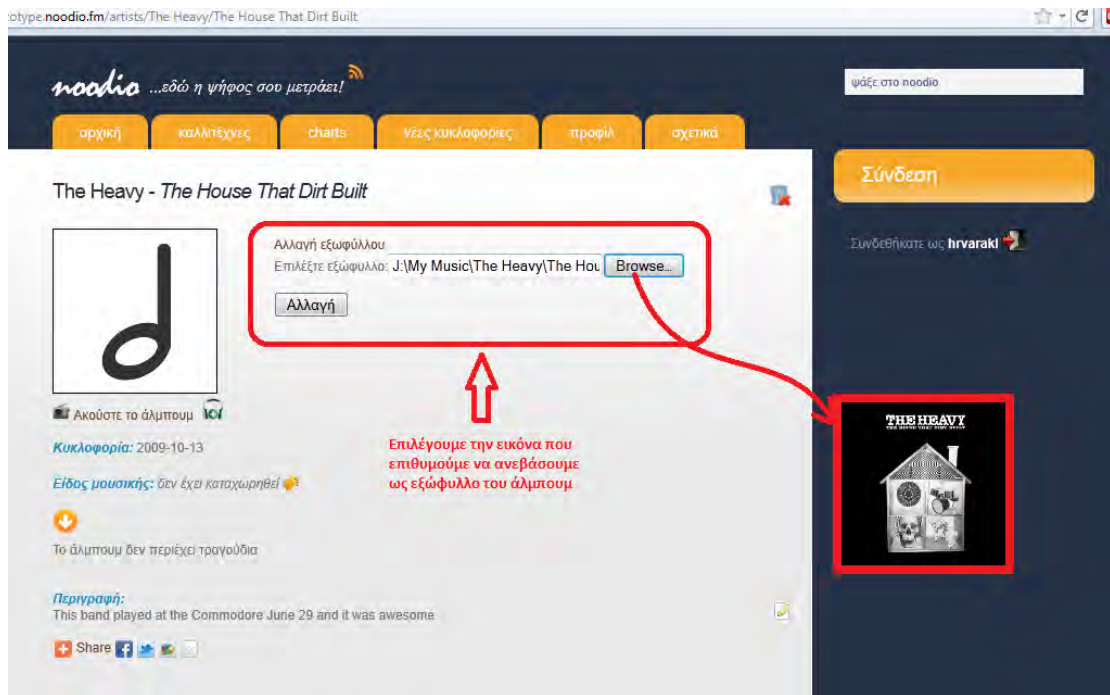
Εικόνα 6-17: Σελίδα προφίλ του νέου καλλιτέχνη

Κάνοντας κλικ στην καρτέλα «Προσθήκη Άλμπουμ» μεταβαίνουμε στη φόρμα δημιουργίας άλμπουμ για τον καλλιτέχνη. Έχοντας συμπληρώσει τα απαραίτητα στοιχεία κάνουμε κλικ στο «Προσθήκη».



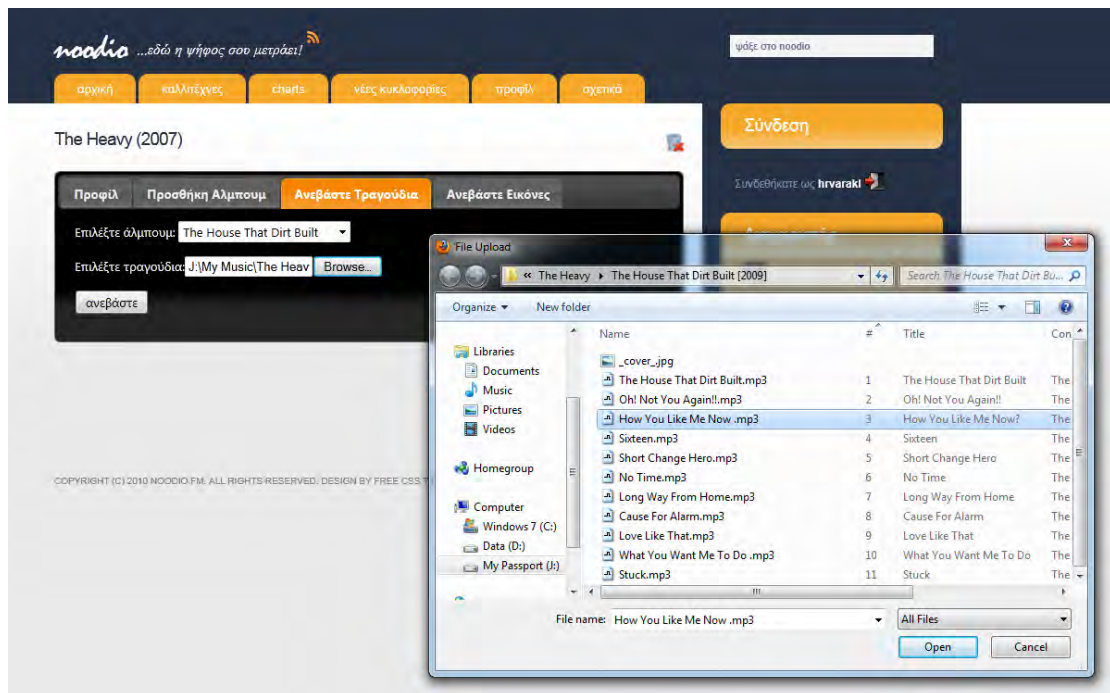
Εικόνα 6-18: Προσθήκη νέου άλμπουμ

Μετά την προσθήκη του νέου άλμπουμ, μεταβαίνουμε στη σελίδα του, από όπου μπορούμε να επεξεργαστούμε τα στοιχεία του. Επιθυμούμε να ανεβάσουμε μία φωτογραφία ως εξώφυλλο για το άλμπουμ μας, οπότε κάνουμε κλικ στην επιλογή «Αλλαγή εξωφύλλου». Επιλέγουμε μία φωτογραφία κάνοντας κλικ στο κουμπί «browse» και έπειτα κάνουμε κλικ στο κουμπί «Αλλαγή».



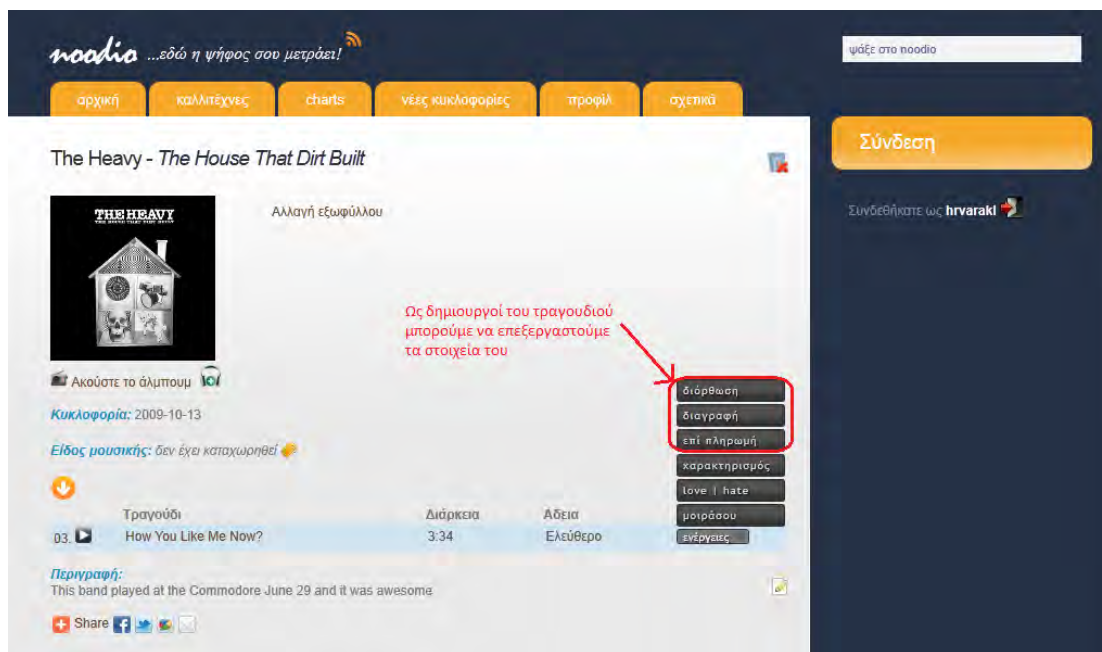
Εικόνα 6-19: Προσθήκη εικόνας εξωφύλλου στο άλμπουμ

Στη συνέχεια, εφόσον επιθυμούμε να ανεβάσουμε ένα τραγούδι για τον καλλιτέχνη μας, επιστρέφουμε στη σελίδα του προφίλ του. Από εκεί, επιλέγουμε την καρτέλα «Ανεβάστε Τραγούδια». Με τη βοήθεια της φόρμας που εμφανίζεται, επιλέγουμε το τραγούδι που επιθυμούμε και κάνουμε κλικ στο κουμπί «ανεβάστε».



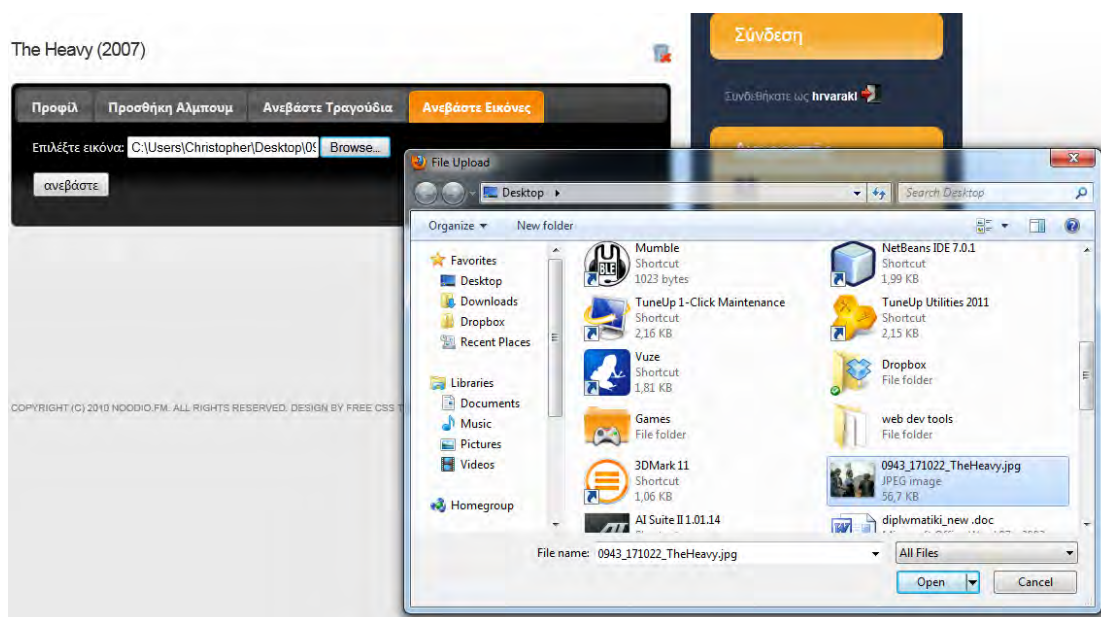
Εικόνα 6-20: Μεταφόρτωση τραγουδιού για το άλμπουμ

Εάν επιθυμούμε να επεξεργαστούμε το κομμάτι που ανεβάσαμε, μπορούμε να μεταβούμε στη σελίδα του άλμπουμ στο οποίο ανήκει.



Εικόνα 6-21: Επεξεργασία στοιχείων τραγουδιού

Έχοντας ολοκληρώσει τη διαδικασία μεταφόρτωσης και επεξεργασίας του νέου τραγουδιού, επιστρέφουμε στη σελίδα του καλλιτέχνη, από όπου θα επιχειρήσουμε να μεταφορτώσουμε μία εικόνα για το προφίλ του.



Εικόνα 6-22: Μεταφόρτωση εικόνας προφίλ για τον καλλιτέχνη

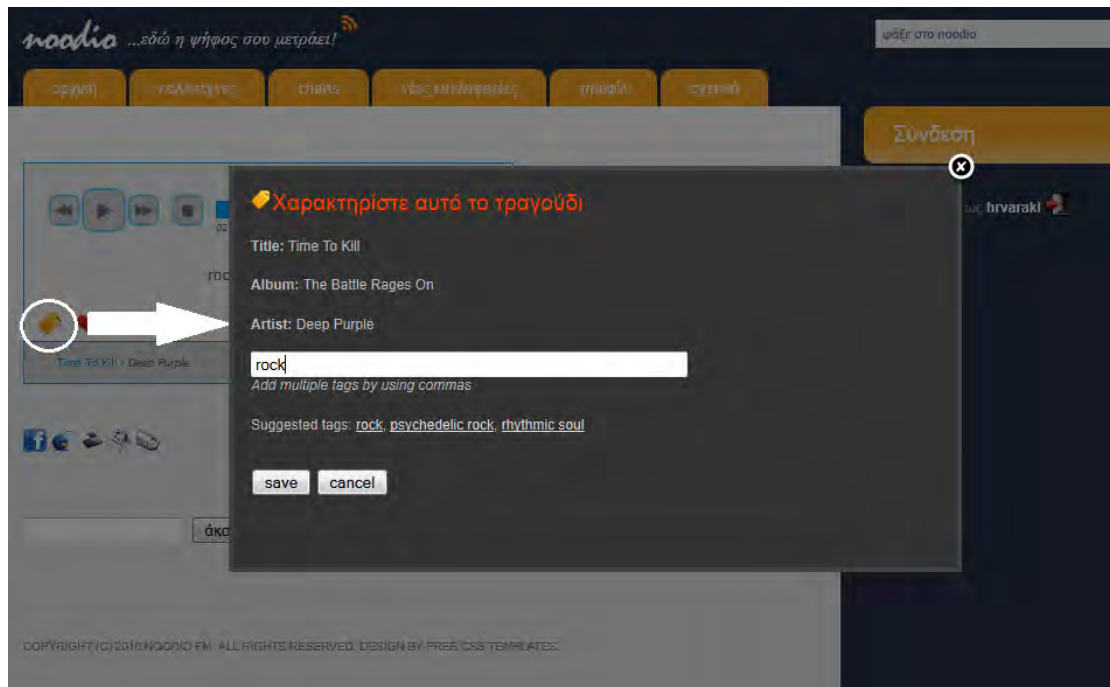
Με τη βοήθεια της φόρμας επιλέγουμε μία εικόνα για το προφίλ του καλλιτέχνη και κάνουμε κλικ στο «ανεβάστε».

Έχοντας ολοκληρώσει τη διαδικασία δημιουργίας του νέου καλλιτέχνη, μπορούμε να επιστρέψουμε στην αρχική σελίδα του site. Από εκεί, μπορούμε να επιλέξουμε έναν από τους διαθέσιμους σταθμούς ώστε να αρχίσουμε να ακούμε μουσική. Από τη σελίδα αναπαραγωγής, μπορούμε να ψηφίσουμε θετικά ή αρνητικά το τραγούδι που ακούμε, ενώ αν θέλουμε να αλλάξουμε σταθμό, μπορούμε είτε να μεταβούμε στην αρχική σελίδα, είτε να χρησιμοποιήσουμε τη φόρμα στο κάτω μέρος της οθόνης. Στο συγκεκριμένο σενάριο, επειδή το τραγούδι που ακούσαμε μας άφησε θετικές εντυπώσεις το ψηφίσαμε θετικά.



Εικόνα 6-23: Ενέργειες σελίδας αναπαραγωγής

Στη συνέχεια επιθυμούμε αποδώσουμε μερικές ετικέτες στο κομμάτι για να βοηθήσουμε στην κατηγοριοποίησή του. Κάνοντας κλικ στο εικονίδιο της ετικέτας, εμφανίζεται ένα παράθυρο διαλόγου στο οποίο μπορούμε να εισάγουμε τις ετικέτες που επιθυμούμε. Εάν δυσκολευόμαστε να διαλέξουμε την κατάλληλη ετικέτα για το κομμάτι, μπορούμε να κάνουμε κλικ σε μία ή περισσότερες από τις προτεινόμενες ετικέτες.



Εικόνα 6-24: Απόδοση ετικετών

7

Σύνοψη και μελλοντικές προοπτικές

Ξεκινήσαμε την παρουσίαση της εργασίας μας με μια αναφορά στην εξέλιξη του διαδικτύου επικεντρώνοντας στον όρο της συμμετοχικότητας η οποία αποτελεί κυρίαρχο χαρακτηριστικό του web 2.0. Μιλήσαμε για το φαινόμενο της μακριάς ουράς και για το ότι η αγορά που ενδιαφέρεται για το long tail δεν είναι καθόλου ευκαταφρόνητη, συμπέρασμα στο οποίο εξάλλου οδηγήθηκε και ο Chris Anderson σύμφωνα με την ανάλυση του. Αναλύσαμε την εξέλιξη του διαδικτύου, την επιρροή του στην παγκόσμια μουσική βιομηχανία και το όφελος των καλλιτεχνών του long tail οι οποίοι μπορούν πλέον να μοιράζουν εύκολα και γρήγορα τη μουσική τους διαδικτυακά.

Στη συνέχεια έγινε αναφορά για τον τεράστιο όγκο πληροφορίας που υπάρχει στο διαδίκτυο και για τις μεθόδους που χρησιμοποιούνται έτσι ώστε να επιτευχθεί η καλύτερη οργάνωση της. Επικεντρωθήκαμε στο συλλογικό φιλτράρισμα πληροφοριών το οποίο στοχεύει να κατευθύνει το χρήστη ανάλογα με τις προτιμήσεις και στα συστήματα ψηφοφοριών τα οποία εξάγουν γενικότερα συμπεράσματα για τις επιλογές των χρηστών. Αναφερθήκαμε επίσης σε μια πληθώρα ιστοσελίδων που χρησιμοποιούν τέτοια συστήματα πληροφοριακών συστάσεων.

Όσον αφορά στην εφαρμογή μας, εξετάσαμε το γεγονός ότι η πλατφόρμα θα στηρίζεται σε περιορισμένο συγκριτικά αριθμό ακροατών γεγονός που μας οδήγησε να εφαρμόσουμε voting system και να αποφύγουμε το collaboration filtering. Στη συνέχεια επικεντρωθήκαμε στον κύριο στόχο μας ο οποίος είναι η ανάδειξη των νέων καλλιτεχνών οι οποίοι βρίσκονται στην αφάνεια της μουσικής σκηνής της χώρας μας. Επιπρόσθετα, αναλύσαμε τους

αλγόριθμους που αναπτύξαμε για την πλατφόρμα μας και οι οποίοι υλοποιούν την αυτόματη οργάνωση του μουσικού περιεχομένου. Αυτοί είναι οι αλγόριθμοι για το σύστημα ψηφοφορίας, την δημιουργία ραδιοφωνικών σταθμών και τη δημιουργία των λιστών αναπαραγωγής.

Έχει γίνει επίσης εκτενής αναφορά στις τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη του web application. Η κύρια server-side γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η Java, περιλαμβάνοντας Servlets, JSP, JSTL και EL σύμφωνα με το μοντέλο MVC. Ως Βάση Δεδομένων επιλέξαμε τη MySQL και ως client-side τεχνολογία Javascript, συμπεριλαμβάνοντας JQuery, Ajax και JSON. Τέλος, σε καίρια σημεία έγινε χρήση HTML 5.0.

Εξετάζοντας τη λειτουργικότητα παρόμοιων εφαρμογών που υπάρχουν ήδη στο διαδίκτυο, παρατηρήσαμε πως υπάρχουν πολλές υπηρεσίες οι οποίες λειτουργούν ως εξατομικευμένα διαδικτυακά ραδιόφωνα, όπως είναι το Last.fm, το Ping της Apple, το Slacker.com και πολλά άλλα που χρησιμοποιούνται σε παγκόσμια κλίμακα και εφαρμόζουν collaborative filtering. Υπάρχουν άλλες εφαρμογές, όπως το mixrod.com, που δεν είναι εξατομικευμένα ραδιόφωνα γιατί χρησιμοποιούν συστήματα ψηφοφορίας που οδηγούν στην εξαγωγή συμπερασμάτων με βάση τις επιλογές που κάνει η πλειοψηφία των χρηστών. Όλες αυτές οι εφαρμογές βέβαια δεν εστιάζουν στους νέους καλλιτέχνες, αλλά σε όλο το φάσμα των καλλιτεχνών. Όσον αφορά στη χώρα μας, έχουν γίνει κάποιες προσπάθειες, με ενδεικτικές τις [jumpingfish.gr²¹](#), [mygreek.fm²²](#) και [musicwave.gr²³](#), οι οποίες έχουν ως βασικό στόχο την προώθηση νέων καλλιτεχνών, διαθέτουν web radio αλλά δεν εφαρμόζουν κάποιο σύστημα πληροφοριακών συστάσεων ή σύστημα ψηφοφορίας.

Η λειτουργικότητα λοιπόν της πλατφόρμας που έχουμε αναπτύξει από όσο γνωρίζουμε δεν προσφέρεται από κάποια άλλη υπηρεσία στην χώρα μας, διότι αφενός μεν στηρίζουμε και προωθούμε τους νέους καλλιτέχνες αφετέρου δε χρησιμοποιούμε το σύστημα ψηφοφορίας έτσι ώστε η εφαρμογή να αποκτήσει την κατάλληλη νοοτροπία και να αντιλαμβάνεται ποια είναι τα τραγούδια και οι καλλιτέχνες που οι χρήστες αναδεικνύουν πλειοψηφικά. Το πλεονέκτημα αυτό που έχει η εφαρμογή μας σε σύγκριση με τις υπόλοιπες, αποτελεί ισχυρό κίνητρο για να μπορέσει να καθιερωθεί και να γίνει γνωστή στους καλλιτέχνες και στους χρήστες.

Είναι προφανές πως οι μελλοντικές προοπτικές για μια τέτοια εφαρμογή είναι ανεξάντλητες αν αναλογιστούμε πως οι απαιτήσεις των χρηστών ποικίλλουν. Έτσι πρέπει να παρέχεται ένα αρκετά ευρύ φάσμα λειτουργιών για την ικανοποίηση των χρηστών. Πιο κάτω, αναφέρουμε μερικές από τις πιο άμεσες λειτουργίες που θα πρέπει να προστεθούν ή να βελτιωθούν :

- Αλλαγή της εμφάνισης και των λειτουργιών των κεντρικών σελίδων index και home. Στη σελίδα home θα προστεθεί το τραγούδι και ο καλλιτέχνης που συγκεντρώνει τις

περισσότερες θετικές ψήφους σε εβδομαδιαία βάση αλλά και το ιστορικό των σταθμών που ο χρήστης έχει ήδη επιλέξει για να ακούσει. Στην σελίδα index θα επικεντρωθούμε σε λειτουργίες διαφημιστικού σκοπού.

- Προσθήκη επιπλέον λειτουργιών κοινωνικού χαρακτήρα. Ο χρήστης θα έχει την επιλογή να αφήνει σχόλια (comments) και να έρχεται σε επαφή με άλλους χρήστες της εφαρμογής.
- Δημιουργία εκδηλώσεων (events). Ο καλλιτέχνης με αυτόν τον τρόπο μπορεί να δημιουργήσει events και έτσι ενημερώνει το κοινό του για την ώρα και τη μέρα της ζωντανής του εμφάνισης.
- Video Gallery. Παρόμοιο με το picture gallery που διαθέτει η εφαρμογή. Ο καλλιτέχνης θα μπορεί να ανεβάζει διάφορα βίντεο τα οποία ο χρήστης θα μπορεί να παρακολουθήσει εάν μπει στο προφίλ του και ανοίξει την video gallery.
- Αλλαγή της σελίδας του ραδιοφώνου, δείχνοντας διάφορες εικόνες του καλλιτέχνη του οποίου το τραγούδι παίζεται εκείνη την στιγμή. Ως αποτέλεσμα, ο χρήστης όταν ακούει ραδιόφωνο θα έχει μπροστά του μια πιο interactive σελίδα.
- Παρόμοιοι ακροατές. Με αυτή τη δυνατότητα ο χρήστης θα μπορεί να ανακαλύψει χρήστες με τους οποίους ακούνε παρόμοια μουσική και να τους προσθέσει στα φίλους.
- Σχεδιασμός ενός πιο εύχρηστου και φιλικού user interface, το οποίο θα προσελκύει τόσο τους απλούς χρήστες, όσο και τους καλλιτέχνες.
- Μηχανισμοί αυτόματης μεταφοράς στοιχείων και πολυμεσικών αντικειμένων από παρόμοια συστήματα π.χ myspace.com.

8

Σύνδεσμοι

1. <http://www.wired.com/wired/archive/12.10/tail.html>
2. <http://www.apple.com/itunes/>
3. <http://www.last.fm>
4. <http://www.pandora.com>
5. <http://www.slacker.com>
6. http://wiki.musichackday.org/index.php?title=Future_of_Music
7. <http://hypem.com>
8. <http://www.mixpod.com>
9. <http://www.readwriteweb.com/enterprise/>
10. <http://www.readwriteweb.com/cloud/>
11. <http://www.readwriteweb.com/>
12. http://www.readwriteweb.com/archives/wisdom_of_the_crowd_isnt_enough_for_music_recommendation_services.php
13. <http://www.bridgew.edu/>
14. <https://www.paypal.com>
15. <http://www.google.com/recaptcha>
16. <http://www.facebook.com>

17. <http://twitter.com>
18. <http://www.delicious.com>
19. <http://digg.com/news>
20. <http://www.google.com/buzz>
21. <http://www.jumpingfish.gr>
22. <http://www.mygreek.fm>
23. <http://www.musicwave.gr>
24. <http://www.we7.com>
25. <http://www.trackitdown.net>
26. <http://www.stereomood.com>
27. <http://grooveshark.com>
28. <http://musicoverly.com>
29. <http://www.rhapsody.com>
30. <http://www.livioradio.com>
31. <http://www.meemix.com>
32. <http://playlistnow.fm>
33. <http://www.play.fm>
34. <http://www.jamendo.com>
35. <http://bandcamp.com>
36. <http://carbonlogic.co.uk>
37. <http://www.reverbnation.com>
38. <http://www.spotify.com>
39. <http://www.deezer.com>
40. <https://www.aupeo.com/>
41. <http://www.thesixtyone.com>
42. <http://shuffler.fm>
43. <http://www.mymajorcompany.com>
44. <http://www.playlist.com/>
45. www.mixlr.com
46. <http://mog.com/>
47. <http://myousic.me>
48. <http://music.google.com>

49. <http://www.rdio.com/>