



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Διαχείριση μουσικών τίτλων για ψηφιακό ραδιόφωνο



Του φοιτητή

Καράμαργιου Αναστάσιου

Αρ. Μητρώου: 011741

Επιβλέπων καθηγητής

Δεληγιάννης Ιγνάτιος

Θεσσαλονίκη 2012

ΠΡΟΛΟΓΟΣ

«Είναι η μουσική στη συνείδησή μας, ο χορός στο πνεύμα μας, με τα οποία δεν θέλουν να εναρμονιστούν όλες οι πουριτανικές λιτανείες, τα ηθικά κηρύγματα και η χρηστότητα..» -Πέρα από το καλό και το κακό-

«Πόσα λίγα χρειαζόμαστε για την ευτυχία! Τον ήχο ενός αυλού... Χωρίς τη μουσική θα 'ταν η ζωή ένα λάθος» (Friedrich Wilhelm Nietzsche).

Με ένα απόσπασμα από το «λυκόφως των ειδώλων» του μεγάλου φιλόσοφου Νίτσε αποφάσισα να κάνω έναν μικρό πρόλογο για την συνέχεια της πτυχιακής μου εργασίας, καθώς όπου υπάρχει μουσική κάνει απλώς τα πράγματα πιο εύκολα και πιο ευχάριστα. Ο συνδυασμός της μουσικής με μια επικοινωνιακή εργασία είναι ο καλύτερος συνδυασμός και αυτό που θα μπορούσα να πράξω με απόλυτη ευχαρίστηση.

ΠΕΡΙΛΗΨΗ

Η εφαρμογή θα αποτελείται από μια ιστοσελίδα στην οποία θα υφίσταται ένας player τραγουδιών. Ο συγκεκριμένος player θα είναι συνδεδεμένος με μια βάση τραγουδιών στην οποία θα είναι καταχωρημένα τα εξής χαρακτηριστικά για κάθε τραγούδι :

- Title (Τίτλος τραγουδιού)
- Artist (Καλλιτέχνης)
- Year (Έτος κυκλοφορίας)
- Language(Γλώσσα τραγουδιού)
- Genre (Είδος)
- Mood (Διάθεση)

Στην εφαρμογή αυτή θα μπορεί να έχει πρόσβαση ένας οποιοσδήποτε χρήστης (user) και ένας συγκεκριμένος διαχειριστής συστήματος (Administrator). Οι λειτουργίες της εφαρμογής λοιπόν καθορίζονται αναλόγως το ποιος την χρησιμοποιεί.

Καταρχήν ο διαχειριστής του συστήματος έχει την δυνατότητα για τον πλήρη έλεγχο της ιστοσελίδας δηλαδή

- Να μπορεί να τροποποιήσει την ιστοσελίδα (edit σελίδας)

- Να μπορεί να έχει πρόσβαση στην βάση δεδομένων των τραγουδιών. Να μπορεί δηλαδή να κάνει εισαγωγή τραγουδιών στην βάση, διαγραφή, επεξεργασία κάποιων στοιχείων κτλ.

Στη συνέχεια οι λειτουργίες της εφαρμογής αφορούν έναν απλό οποιοδήποτε χρήστη και θα είναι οι εξής :

Θα μπορεί να έχει πρόσβαση στην ιστοσελίδα και θα μπορεί να ακούσει τυχαία τραγούδια , ή επιλεγμένα βάση των "φίλτρων" που θα υπάρχουν (Αναλόγως το είδος της μουσικής πχ ροκ, έντεχνο, αναλόγως την δεκαετία πχ 60's). Επίσης θα μπορεί να συνδυάσει τα παραπάνω φίλτρα (multiple).

Επίσης θα του δίνεται η δυνατότητα να ακούσει τραγούδια σε ενότητες οι οποίες η κάθε μια θα περιλαμβάνει τραγούδια διάρκειας 30 λεπτών, και η κάθε ενότητα θα ακολουθεί η μία την άλλη. Δηλαδή, εδώ θα χρησιμοποιηθεί μια λειτουργία ραδιοφώνου που θα επιτρέπει στον χρήστη να εναλλάσει κατηγορίες ανά μισή ώρα αυτόματα.

Τέλος στην ιστοσελίδα θα χρησιμοποιηθεί ένα είδος "feedback" (επικοινωνία μέσω email) τα οποία θα μπορεί να λάβει υπόψιν του ο Administrator για να κάνει κάποιες βελτιώσεις, ή ακόμα και για να προσθέσει κάποιο είδος τραγουδιών αν του ζητηθεί και εφόσον βέβαια υπάρχει η δυνατότητα αυτή.

ABSTRACT

The application will be constituted by a web page in which we will find an mp3 player. The player will be connected with a database of songs in which will be registered the following characteristics for each song:

- Title
- Artist
- Year
- Language
- Genre
- Mood

In this application they might have access any user (simple user) and a system administrator (Administrator). The operations of application therefore are determined proportionally who uses it.

First of all, the system administrator has the potential of full control of the webpage. His potentials are:

- He can edit the webpage
- He can have access in the songs database. (Insert songs, delete songs, edit information of songs etc).

Afterwards the operations of application concern a simple user and will be the following: He can have access in the webpage and he can listen to music. The mp3 songs can be selected randomly or via categories (genre: rock, pop etc, 60's 70's etc) and he can combine multiple filters for his compilation.

The user can also have the ability to listen to music per categories. Each category includes songs that their total duration is 30 minutes. Each category is following the other. The user can use this application as a radio to listen to music by categories automatically

Finally the user has the potential to leave a feedback via email in the administrator to ask for more songs or other genres or anything he wants to help administrator to improve the potentials of the webpage.

ΕΥΧΑΡΙΣΤΙΕΣ

Στην Άννα

Στην οικογένεια μου για την υπομονή και την στήριξη τους

Στους φίλους μου Κώστα και Άκη για την πολύτιμη βοήθεια και υποστήριξη τους

Στον κ. Δεληγιάννη για την άψογη συνεργασία και την βοήθεια του

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ	2
ΠΕΡΙΛΗΨΗ	3
ABSTRACT	4
ΕΥΧΑΡΙΣΤΙΕΣ.....	5
Ευρετήριο σχημάτων	8
Ευρετήριο πινάκων	8
ΕΙΣΑΓΩΓΗ	10
ΚΕΦΑΛΑΙΟ 1	11
1. Γενική προσέγγιση υλοποίησης της εφαρμογής	11
1.1 Η μουσική στο διαδίκτυο	11
1.1.1 Ιστορική αναδρομή	11
1.1.2 Μέσα αναπαραγωγής μουσικής	12
1.2 Διαχείριση μουσικών τίτλων.....	13
1.2.1 Λειτουργία ως player	13
1.2.2 Λειτουργία ως ραδιόφωνο.....	13
1.3 Μέσα υλοποίησης της εφαρμογής.....	14
1.3.1 Πακέτο XAMPP	14
1.3.2 HTML – CSS – Javascript – XML	16
1.3.3 Adobe Dreamweaver	18
1.3.4 Server	19
1.3.5 Μεθοδολογία RUP	19
1.3.6 Visual Paradigm	19
1.3.7 Open proj.....	20
1.4 Περιορισμοί της εφαρμογής.....	20
1.4.1 Περιορισμοί Hardware.....	20
1.4.2 Περιορισμοί Software	20
1.5 Νομιμότητα της εφαρμογής	21
1.6 Επίλογος	21
ΚΕΦΑΛΑΙΟ 2	21
2. Μεθοδολογία υλοποίησης της εφαρμογής.....	21
2.1 Γενικά	22
2.2 Βασικές αρχές	23
2.3 Φάσεις Ανάπτυξης Λογισμικού	24

2.3.1 Σχεδιασμός των φάσεων.....	24
2.3.2 Φάση έναρξης (Inception).....	25
2.3.3 Φάση επεξεργασίας (Elaboration).....	25
2.3.4 Φάση κατασκευής (Construction).....	26
2.3.5 Φάση μετάβασης (Transition).....	27
2.4 Επαναλήψεις.....	27
ΚΕΦΑΛΑΙΟ 3	29
3. Ανάλυση και σχεδίαση.....	29
3.1 Μοντέλο περιπτώσεων χρήσης	29
3.1.1 Πίνακας χρηστών στόχων.....	30
Πίνακας 1. Πίνακας χρηστών - στόχων.....	30
3.1.2 Use case diagram	30
3.1.3 Περιπτώσεις χρήσης	31
3.2 Διάγραμμα ακολουθίας(Sequence diagram).....	35
3.2.1 Διάγραμμα ακολουθίας για την περίπτωση χρήσης ΠΧ1.....	35
3.2.2 Διάγραμμα ακολουθίας για την περίπτωση χρήσης ΠΧ1.....	36
3.2.3 Διάγραμμα ακολουθίας για την περίπτωση χρήσης ΠΧ3.....	37
3.3 Use case estimation effort (εκτίμηση προσπάθειας)	38
3.3.1 TCF (Technical Complexity Factors).....	39
3.3.2 ECF (Environmental Complexity Factors).....	41
3.3.3 UUCP (Unadjusted Use Case Points).....	42
3.3.4 Productivity Factor	44
3.3.5 Υπολογισμός UCP	44
3.4 Open proj	45
3.5 Επίλογος.....	46
ΚΕΦΑΛΑΙΟ 4	47
4. Υλοποίηση.....	47
4.1 Δημιουργία αρχικής σελίδας.....	47
4.1.1 Σχεδίαση	48
4.1.2 Δημιουργία μενού	48
4.2 Δημιουργία βάσης δεδομένων	51
4.2.1 Δημιουργία πίνακα τραγουδιών.....	51
4.2.2 Εισαγωγή τραγουδιών	52
4.3 Δημιουργία φόρμας επικοινωνίας.....	52

4.4 Δημιουργία καρτέλας λίστα τραγουδιών.....	54
4.5 Δημιουργία καρτέλας του player.....	56
4.5.1 Δημιουργία φόρμας επιλογής τραγουδιών.....	56
4.5.2 Δημιουργία φόρμας επιλογής τραγουδιών ανά ενότητες.....	61
4.5.3 Εισαγωγή και λειτουργία player.....	64
4.5.4 Δημιουργία λίστας αναπαραγωγής τραγουδιών.....	66
4.6 Εισαγωγή widgets στην σελίδα.....	68
4.6.1 Εισαγωγή στατιστικών προτιμήσεων στην σελίδα.....	69
4.6.2 Δυνατότητα εισαγωγής περαιτέρων widgets.....	70
4.7 Ανάπτυξη.....	71
4.8 Επίλογος.....	71
ΕΠΙΛΟΓΟΣ – ΣΥΜΠΕΡΑΣΜΑΤΑ.....	72
ΠΗΓΕΣ.....	73
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	76

Ευρετήριο σχημάτων

Σχήμα 1. Φάσεις μεθοδολογίας RUP.....	21
Σχήμα 2. Κατανομή χρονοπρογραμματισμού και προσπάθειας.....	24
Σχήμα 3. Στιγμιότυπο πορείας της μεθοδολογίας RUP κατά την φάση της έναρξης.....	28
Σχήμα 4. Use case diagram.....	30
Σχήμα 5. Στιγμιότυπο πορείας της μεθοδολογίας RUP κατά την φάση της επεξεργασίας.....	34
Σχήμα 6. Διάγραμμα ακολουθίας για την ΠΧ1.....	35
Σχήμα 7. Διάγραμμα ακολουθίας για την ΠΧ2.....	36
Σχήμα 8. Διάγραμμα ακολουθίας για την ΠΧ3.....	37
Σχήμα 9. Χρονοδιάγραμμα υλοποίησης ΠΕ.....	45
Σχήμα 10. Διάγραμμα Gantt.....	46
Σχήμα 11. Στιγμιότυπο πορείας της μεθοδολογίας RUP κατά την φάση της κατασκευής.....	47
Σχήμα 12. Στιγμιότυπο πορείας της μεθοδολογίας RUP κατά την φάση της μετάβασης.....	71

Ευρετήριο πινάκων

Πίνακας 1. Πίνακας χρηστών – στόχων.....	29
Πίνακας 2. Τεχνικοί παράγοντες.....	39
Πίνακας 3. Διαδικασία υπολογισμού συνολικού τεχνικού παράγοντα.....	39

Πίνακας 4. Περιβαλλοντικοί παράγοντες.....	40
Πίνακας 5. Διαδικασία υπολογισμού συνολικού περιβαλλοντικού παράγοντα.....	41
Πίνακας 6. Κατηγορίες περιπτώσεων χρήσης.....	43
Πίνακας 7. Διαδικασία υπολογισμού συνολικού UUCW.....	43
Πίνακας 8. Κατηγορίες χρηστών.....	43
Πίνακας 9. Διαδικασία υπολογισμού συνολικού UAW.....	44

ΕΙΣΑΓΩΓΗ

«Διαχείριση μουσικών τίτλων για ψηφιακό ραδιόφωνο». Στόχος λοιπόν της πτυχιακής εργασίας με τον παραπάνω τίτλο είναι να καταφέρουμε να φτιάξουμε μια εφαρμογή σε κάποια ιστοσελίδα η οποία θα μπορεί να διαχειρίζεται μουσικά κομμάτια. Να μπορεί να δίνεται δηλαδή η ικανότητα στον χρήστη να ακούσει και μεμονωμένα κομμάτια της επιλογής του αλλά και να μπορεί να χρησιμοποιήσει την εφαρμογή αυτήν σαν ραδιόφωνο ακούγοντας μουσικές ενότητες ανά τριάντα λεπτά.

Σκοπός της πτυχιακής αυτής εργασίας είναι να καταφέρουμε να συνδυάσουμε το ψυχαγωγικό με το δημιουργικό κομμάτι. Μέσα δηλαδή από την εφαρμογή θα καταφέρουμε να αποκομίσουμε σημαντικές εμπειρίες και εφόδια και παράλληλα με το ψυχαγωγικό κομμάτι να πράξουμε, πιο ευχάριστα.

Το πρώτο κεφάλαιο περιλαμβάνει μια γενική προσέγγιση σχετικά με το πως θα υλοποιήσουμε την εφαρμογή όσον αφορά την μεθοδολογία, την σχεδίαση και την υλοποίηση. Θα αναφερθούν δηλαδή όλα τα μέσα και τα εργαλεία που θα χρησιμοποιηθούν τα οποία θα μας βοηθήσουν στην διεκπεραίωση της εφαρμογής αυτής.

Στο δεύτερο κεφάλαιο αναφερόμαστε εκτενέστερα στην μεθοδολογία που θα χρησιμοποιηθεί. Θα αφορά μια γενική περιγραφή αλλά και θα αναλυθούν όλα τα βήματά της.

Στο τρίτο κεφάλαιο περιλαμβάνεται η ανάλυση και η σχεδίαση της εφαρμογής. Θα αναφερθούν και θα αναλυθούν οι περιπτώσεις χρήσης, και θα σχεδιαστούν όλα τα απαραίτητα διαγράμματα.

Τέλος στο τέταρτο και τελευταίο μας κεφάλαιο θα αναφερθούμε αποκλειστικά στην υλοποίηση της εφαρμογής μας όσον αφορά το προγραμματιστικό κομμάτι και θα υπάρξουν εκτενείς αναφορές στις γλώσσες και στα κομμάτια κώδικα τα οποία έχουν χρησιμοποιηθεί.

ΚΕΦΑΛΑΙΟ 1

1. Γενική προσέγγιση υλοποίησης της εφαρμογής

Σ' αυτό το κεφάλαιο θα δούμε την υλοποίηση της πτυχιακής εργασίας από μια γενική σκοπιά, θα δούμε δηλαδή κάποια πολύ γενικά πράγματα και θα αναφερθούμε στα εργαλεία και στα μέσα που θα χρησιμοποιηθούν για την υλοποίησή της. Τα παραπάνω θα αναλυθούν εκτενέστερα στα επόμενα κεφάλαια.

1.1 Η μουσική στο διαδίκτυο

Το Ίντερνετ, σε συνδυασμό με την ολοένα αναπτυσσόμενη ψηφιακή τεχνολογία, έχει δημιουργήσει μία τεράστια αγορά γνώσεων/πληροφοριών. Η μουσική την τελευταία δεκαετία οφείλει την ανάπτυξη και την εξάπλωσή της στο διαδίκτυο, καθώς σε σχέση με προηγούμενα χρόνια, η ανάπτυξη και η εφαρμογή όλο και συνεχών καινούριων μέσων καθιστά πολύ εύκολη την διάδοση της.

1.1.1 Ιστορική αναδρομή

Προσπάθειες για την αναπαράσταση πληροφορίας πολυμέσων στους ηλεκτρονικούς υπολογιστές έγιναν στα μέσα περίπου του 20ου αιώνα. Ωστόσο, μικρή πρόοδος είχε παρατηρηθεί για αρκετές δεκαετίες, κυρίως λόγω του υψηλού κόστους και των περιορισμένων υπολογιστικών δυνατοτήτων. Από τα τέλη της δεκαετίας του 1980 έως τα μέσα της δεκαετίας του 1990 οι προσωπικοί υπολογιστές έγιναν αρκετά ισχυροί έτσι ώστε να μπορούν να απεικονίσουν ορισμένα πολυμεσικά στοιχεία.

Τα βασικά τεχνικά θέματα που έπρεπε να αντιμετωπιστούν ήταν η επαρκής επεξεργαστική ισχύς και το εύρος ζώνης για μια αποτελεσματική διαχείρισή της ροής των δεδομένων. Επίσης έπρεπε να πραγματοποιηθεί και η προσαρμογή του λειτουργικού συστήματος έτσι ώστε να μην υπάρχουν καθυστερήσεις στην επεξεργασία. Παρ' όλα αυτά, τα δίκτυα υπολογιστών είχαν ακόμη περιορισμένες δυνατότητες και η πληροφορία μεταφερόταν σε αποθηκευτικά μέσα όπως CD-ROM.

Στις αρχές του 2000 παρουσιάστηκε αύξηση του εύρους ζώνης των δικτύων, αύξηση της προσβασιμότητας στα δίκτυα, ειδικά στο Internet, και η χρήση αναγνωρισμένων πρωτοκόλλων και τυποποιήσεων όπως το TCP/IP, το HTTP και η HTML. Οι παράγοντες αυτοί, σε συνδυασμό με τους ισχυρούς προσωπικούς υπολογιστές και τα σύγχρονα λειτουργικά συστήματα, έκαναν τη ροή πολυμεσικής πληροφορίας προσιτή και πρακτική για τους απλούς καταναλωτές.

Πολλοί ραδιοφωνικοί αλλά και μεγάλοι τηλεοπτικοί σταθμοί, πλέον, παρέχουν ζωντανές ή μαγνητοσκοπημένες εκπομπές του προγράμματος τους μέσω των ιστοσελίδων τους όπως για παράδειγμα το BBC. Οι μαγνητοσκοπημένες

εκπομπές αποθηκεύονται σε κάποιον εξυπηρετητή (*server*) και γίνονται διαθέσιμες στο χρήστη έπειτα από αίτησή του.

Τέλος υπάρχουν πλέον χιλιάδες ιστολόγια και ιστότοποι τα οποία παρέχουν την δυνατότητα να ακούσεις κάποιον ραδιοφωνικό σταθμό ή ακόμα και μεμονωμένα τραγούδια.

1.1.2 Μέσα αναπαραγωγής μουσικής

Με μια απλή αναζήτηση σε οποιονδήποτε από τις μηχανές του διαδικτύου θα δούμε ότι υπάρχουν πάρα πολλά μέσα που χρησιμοποιούνται για να αναπαράγουν μουσική και τα περισσότερα από αυτά είναι open source. Είναι λοιπόν δωρεάν και αυτό τους καθιστά άμεσα “re-useable” για την χρησιμοποίηση έναν εκ των πολλών που υπάρχουν για να τοποθετήσουμε στην εφαρμογή μας. Ένα από τα πιο γνωστά και από τα πιο διαχρονικά είναι και το Windows Media player.

Το Windows Media Player (WMP) είναι μια ψηφιακή εφαρμογή προγράμματος αναπαραγωγής πολυμέσων και ψηφιακής βιβλιοθήκης, που ανεπτύχθη από τη Microsoft, χρησιμοποιείται για την αναπαραγωγή ήχων, βίντεο και προβολής εικόνων σε προσωπικούς υπολογιστές και εκτελείται στο λειτουργικό σύστημα Microsoft Windows, καθώς επίσης σε Pocket PC και σε συσκευές Windows Mobile. Εκδόσεις του Windows Media Player κυκλοφόρησαν επίσης για το Mac OS, για το Mac OS X και για το Solaris, αλλά δεν επανεκδόθηκαν γι' αυτά τα προϊόντα.

Εκτός από τις βασικές του λειτουργίες, το Windows Media Player μπορεί να εγγράφει μουσική από και σε CD, να συγχρονίζει περιεχόμενο με μία συσκευή ψηφιακής μουσικής (MP3 player) ή μεταξύ άλλων συσκευών και να επιτρέπει την αγορά ή το δανεισμό μουσικής από έναν αριθμό μουσικών ηλεκτρονικών καταστημάτων.

Το Windows Media Player αντικατέστησε μια παλιά μορφή λογισμικού που ονομαζόταν απλώς Media Player προσθέτοντας χαρακτηριστικά πέρα από την αναπαραγωγή μουσικής ή βίντεο. Οι προεπιλεγμένες μορφές αρχείων είναι η Windows Media Video (WMV), η Windows Media Audio (WMA) και η Advanced Systems Format (ASF), καθώς επίσης υποστηρίζεται η λίστα αναπαραγωγής σε μορφή XML και ονομάζεται Windows Playlist (WPL). Το λογισμικό Zune, το οποίο, ουσιαστικά, είναι μία τροποποιημένη έκδοση του Windows Media Player, υποστηρίζει επιπροσθέτως τη μορφή ήχου AAC (Advanced Audio Coding) και τις μορφές βίντεο MPEG-4 και H.264.

Για τις ανάγκες της πτυχιακής μας όμως θα χρησιμοποιηθεί ένας open source player που έχει δημιουργηθεί από την www.premiumbeat.com.

Είναι πάρα πολύ εύχρηστος και απλός και καθιστά το έργο μας λοιπόν πολύ πιο εύκολο. Έχει αρκετά χαρακτηριστικά που τον κάνουν πολύ χρήσιμο και στην εγκατάσταση αλλά και στην χρησιμοποίησή του.

Μπορούμε λοιπόν εύκολα να τον εγκαταστήσουμε και πάρα πολύ εύκολο να τον τροποποιήσουμε όσον αφορά το μέγεθος και τα χρώματά του.

Έχει μενού στο οποίο εμφανίζονται τα μουσικά κομμάτια και η διάρκειά τους. Τα τραγούδια τα «φορτώνει» κάθε φορά μέσα μιας δυναμικής λίστας XML και τέλος μπορούν να χρησιμοποιηθούν κατά την λειτουργία του πλήκτρα από το πληκρολόγιο όπως το space το οποίο έχει λειτουργία play/pause. Αυτόν τον player λοιπόν θα χρησιμοποιήσουμε στην σελίδα μας.

1.2 Διαχείριση μουσικών τίτλων

Στόχος λοιπόν της πτυχιακής μας εργασίας είναι να χρησιμοποιήσουμε τον συγκεκριμένο player που αναφέρθηκε στην προηγούμενη παράγραφο και σαν απλό player αλλά και με την δυνατότητα να χρησιμοποιηθεί ως ραδιόφωνο. Αυτά θα δούμε λοιπόν στις επόμενες δυο παραγράφους.

1.2.1 Λειτουργία ως player

Όπως έχουμε προαναφερθεί, η λειτουργία του συγκεκριμένου player που θα χρησιμοποιηθεί είναι να φορτώνει τα μουσικά κομμάτια μέσω μιας δυναμικής λίστας XML. Θα δημιουργήσουμε λοιπόν μία φόρμα με τη βοήθεια php και html που θα αναλυθούν εκτενέστερα σε επόμενα κεφάλαια. Η φόρμα αυτή θα μας δίνει την δυνατότητα να εισάγουμε στην λίστα μας μεμονωμένα τραγούδια από οποιαδήποτε κατηγορία. Θα μπορεί να επιλεγεί συγκεκριμένος καλλιτέχνης κάθε φορά, γλώσσα προέλευσης του τραγουδιού, έτος κυκλοφορίας, είδος μουσικής και αναλόγως την μουσική διάθεση που προδιαθέτουν (χαρούμενη ή μελαγχολική). Με την επιλογή των εκάστοτε φίλτρων λοιπόν θα επιλέγονται τα διαθέσιμα κομμάτια και εφόσον υπάρχουν διαθέσιμα θα εισάγονται στην λίστα αναπαραγωγής μας και έπειτα θα φορτώνονται στον player. Έτσι λοιπόν θα καταφέρουμε να δημιουργήσουμε έναν απλό player ο οποίος θα αναπαράγει κάθε φορά τραγούδια της επιλογής του εκάστοτε χρήστη. Και όπως κάθε μέσο αναπαραγωγής θα έχει σαφώς την δυνατότητα να διακοπεί και να ξαναεπιλεχθούν μουσικά κομμάτια.

1.2.2 Λειτουργία ως ραδιόφωνο

Η άλλη λειτουργία του player μας θα πρέπει να είναι αυτή του ραδιοφώνου. Να επιλέγονται δηλαδή τραγούδια ανά τριάντα περίπου λεπτά από κάθε μουσική ενότητα που αντιστοιχεί σε διαφορετικό είδος. Θα δημιουργήσουμε μια φόρμα δηλαδή με προϋπάρχουσες μουσικές ενότητες (πχ ροκ, ποπ, κτλ) και κάθε φορά που ο χρήστης θα επιλέγει την μουσική ενότητα που επιθυμεί θα επιλέγονται αυτόματα τραγούδια συνολικής διάρκειας 30 λεπτών. Θα εισάγονται επίσης στην λίστα αναπαραγωγής μας ακόμα 30 λεπτά μουσικής από την επόμενη μουσική ενότητα και έπεται συνέχεια στις μουσικές κατηγορίες.

Αυτό λοιπόν δίνει την δυνατότητα στον εκάστοτε χρήστη να χρησιμοποιεί την εφαρμογή ως μουσικό ψηφιακό ραδιόφωνο καθώς κάθε 30 λεπτά εναλλάσσονται οι μουσικές ενότητες.

Η διαδικασία υλοποίησης και αυτής της φόρμας θα αναπτυχθεί εκτενέστερα σε επόμενο κεφάλαιο.

1.3 Μέσα υλοποίησης της εφαρμογής

Σε αυτήν την παράγραφο θα αναφερθούν όλα τα μέσα εκείνα τα οποία θα χρησιμοποιήσουμε για την υλοποίηση της εφαρμογής και σε σχεδιαστικό επίπεδο αλλά και μέσα που αφορούν κώδικα.

1.3.1 Πακέτο XAMPP

Το πακέτο XAMPP λοιπόν είναι ένα πακέτο που κυκλοφορεί ελεύθερα στο διαδίκτυο (free and open source) και απαρτίζεται από τα εξής.

- Apache HTTP Server
- PHP
- MySQL
- Perl

Είναι πολύ εύχρηστο, εύκολο στην εγκατάσταση του και μπορεί να χρησιμοποιηθεί από πολλά λειτουργικά συστήματα. Θα μας βοηθήσει να υλοποιήσουμε την εφαρμογή μας πρώτα, για να την δοκιμάσουμε σε τοπικό δίκτυο.

1.3.1.1 Apache HTTP Server

Ο Apache HTTP αναπτύσσεται από την “Κοινότητα Ανοιχτού Λογισμικού” και η εποπτεία, υποστήριξη, και διάθεση του προγράμματος γίνεται από το Apache Software Foundation. Το πρόγραμμα είναι ανοιχτού κώδικα (open source), κάτι που σημαίνει ότι σύμφωνα με την άδεια χρήσης του (license), διατίθεται δωρεάν και μπορούν να γίνουν ελεύθερα από το χρήστη προσθήκες και τροποποιήσεις στον κώδικα του.

Ο Apache διαθέτει ποικιλία χαρακτηριστικών και μπορεί να υποστηρίξει μια μεγάλη γκάμα εφαρμογών με τις οποίες και συνεργάζεται. Οι δυνατότητες του προγράμματος αυτού καθ' εαυτού και τα χαρακτηριστικά του δεν είναι και τόσα πολλά. Ένα από τα βασικότερα χαρακτηριστικά του όμως, το οποίο και του δίνει μεγάλες δυνατότητες, είναι ότι μπορεί να προσαρμόσει επάνω του πολλές προσθήκες προγραμμάτων (modules), τα οποία με τη σειρά τους παρέχουν διαφορετικές λειτουργίες. Μερικά από τα πιο γνωστά modules του Apache HTTP είναι τα modules πιστοποίησης, όπως για παράδειγμα τα mod_access, mod_auth, mod_digest κ.λπ. Παρέχει επίσης SSL σε TLS μέσω των (mod_ssl), και proxy module (mod_proxy), πραγματοποιεί ανακατευθύνσεις διευθύνσεων (URL rewrites) μέσω του mod_rewrite, καταγραφές συνδέσεων μέσω του mod_log_config, συμπίεση αρχείων μέσω του mod_gzip και πολλά άλλα modules τα οποία διατίθενται είτε απ' το Apache Software Foundation, είτε από τρίτες εταιρίες λογισμικού.

Ένα άλλο χαρακτηριστικό – δυνατότητα του Apache HTTP, όπως έχω αναφέρει πιο πάνω, είναι ότι μπορεί να εγκατασταθεί σε διάφορα λειτουργικά συστήματα. Ο Apache HTTP υποστηρίζει επίσης αρκετές διάσημες εφαρμογές και γλώσσες προγραμματισμού όπως MySQL, PHP, Perl, Python κ.λπ.

Αυτά είναι μερικά από τα χαρακτηριστικά και λειτουργίες του που κάνουν τον Apache τον πιο δημοφιλή Web Server από το 1996 έως τις μέρες μας. Περισσότερο από το 50% των ιστοχώρων του παγκόσμιου ιστού, χρησιμοποιεί τον Apache ως εξυπηρετητή. Το υπόλοιπο ποσοστό καλύπτουν αντίστοιχα προγράμματα, όπως το Microsoft Internet Information Services (IIS), ο Sun Java System Web Server, ο Zeus Web Server κ.

1.3.1.2 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων.

Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU General Public License, καθώς και μέσω ορισμένων ιδιόκτητων συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, η οποία σήμερα ανήκει στην Oracle.

Μπορούμε να προσθέσουμε, να ανακτήσουμε και να διαχειριστούμε πληροφορίες που είναι αποθηκευμένες σε μια βάση δεδομένων. Η σχεσιακή MySQL σημαίνει ότι μια πληροφορία αποθηκεύεται σε χωριστούς πίνακες και όχι σε έναν μεγάλο πίνακα. Μπορούν να καθιερωθούν σχέσεις μεταξύ πινάκων και να ανακτούνται πληροφορίες χρησιμοποιώντας δομημένη γλώσσα διατύπωσης ερωτήσεων (SQL).

Θα μας βοηθήσει λοιπόν να δημιουργήσουμε και να διαχειριστούμε την βάση των τραγουδιών μας και αναλυτικότερα θα δούμε σε επόμενο κεφάλαιο.

1.3.1.3 PHP

Η PHP είναι μια "δυνατή" serverside γλώσσα για την κατασκευή δυναμικών και διαδραστικών ιστοχώρων.

Αρχικά η ονομασία της ήταν PHP/FI από το Forms Interpreter η οποία δημιουργήθηκε το 1995 από τον Rasmus Lerdorf ως μια συλλογή από Perl scripts που τα χρησιμοποιούσε στην προσωπική του σελίδα. Δεν άργησε να τα εμπλουτίσει με λειτουργίες επεξεργασίας δεδομένων με SQL, αλλά τα σημαντικά βήματα που έφεραν και την μεγάλη αποδοχή της PHP ήταν αρχικά η μετατροπή τους σε C και μετέπειτα η δωρεάν παροχή του πηγαίου κώδικα μέσω της σελίδας του ώστε να επωφεληθούν όλοι από αυτό που είχε φτιάξει, αλλά και να τον βοηθήσουν στην περαιτέρω ανάπτυξή της.

Η PHP χρησιμοποιείται ευρέως, δωρεάν, και ουσιαστικά αποτελεί μια εναλλακτική επιλογή σε εμπορικές τεχνολογίες όπως την Active Server Pages της Microsoft. Η

PHP ενδείκνυται για την κατασκευή εφαρμογών και σελίδων στο Διαδίκτυο μιας και μπορεί να ενσωματωθεί απευθείας μέσα σε κώδικα HTML. Η σύνταξη της PHP μοιάζει πολύ με τη σύνταξη της Perl και της C. Συνήθως, χρησιμοποιείται με εξυπηρετητή Apache σε διάφορα λειτουργικά συστήματα. Παράλληλα υποστηρίζει και την διεπαφή Internet Server Application Programming Interface (ISAPI) καθιστώντας την δυνατή να χρησιμοποιηθεί και σε Microsoft Windows με εξυπηρετητή Microsoft Internet Information Server.

Αντίθετα από μια συνηθισμένη HTML σελίδα η σελίδα PHP δεν στέλνεται άμεσα σε έναν πελάτη (client), αντ' αυτού πρώτα αναλύεται και μετά αποστέλλεται το παραγόμενο αποτέλεσμα. Τα στοιχεία HTML στον πηγαίο κώδικα μένουν ως έχουν, αλλά ο PHP κώδικας ερμηνεύεται και εκτελείται. Ο κώδικας PHP μπορεί να θέσει ερωτήματα σε βάσεις δεδομένων, να δημιουργήσει εικόνες, να διαβάσει και να γράψει αρχεία, να συνδεθεί με απομακρυσμένους υπολογιστές, κ.ο.κ. Σε γενικές γραμμές οι δυνατότητες που μας δίνει είναι απεριόριστες.

Η υλοποίηση του ιστοτόπου που αφορά την εφαρμογή και την πτυχιακή μας εργασία θα γίνει κυρίως με PHP όπου κομμάτια κώδικα και ανάλυσή τους θα γίνει σε επόμενο κεφάλαιο.

1.3.2 HTML – CSS – Javascript – XML

Η δομή της ιστοσελίδας έχει στηριχτεί και οργανωθεί με τη χρήση HTML (*HyperText Markup Language*). Ο σχεδιασμός, η οργάνωση και ο έλεγχος παρουσίασης των δομημένων με HTML εγγράφων και γενικότερα της ιστοσελίδας θα γίνει με τη βοήθεια CSS (*Cascading Style Sheets*) ενώ για την δημιουργία του δυναμικού περιεχομένου της λίστας αναπαραγωγής θα χρησιμοποιήσουμε XML (*Extensible Markup Language*). Σε κάποιες περιπτώσεις έγινε χρήση και JavaScript κυρίως για την προσθήκη δευτερευόντων λειτουργιών. Στις παραγράφους που ακολουθούν θα αναφερθούμε απλά στις λειτουργίες των παραπάνω.

1.3.2.1 HTML

Τα αρχικά HTML προέρχονται από τις λέξεις HyperText Markup Language. Η HTML δεν είναι κάποια γλώσσα προγραμματισμού αλλά μία περιγραφική γλώσσα, δηλαδή ένας ειδικός τρόπος γραφής κειμένου τον οποίο αντιλαμβάνονται και μεταφράζουν ανάλογα τα προγράμματα περιήγησης (*browsers*) εκτελώντας τις εντολές που ορίζονται από το κείμενο. Για το σκοπό αυτό η HTML χρησιμοποιεί ειδικές ετικέτες (*tags*) οι οποίες δίνουν τις απαραίτητες οδηγίες στον browser. Τα tags είναι εντολές που συνήθως ορίζουν την αρχή ή το τέλος μιας λειτουργίας και βρίσκονται μεταξύ των συμβόλων < και >, π.χ. <body>. Ένα αρχείο HTML πρέπει να έχει κατάληξη .htm ή .html ενώ σε έναν ιστότοπο που αποτελείται από πολλά αλληλένδετα HTML αρχεία, το αρχείο που αποτελεί την αρχική σελίδα συνήθως πρέπει να έχει την ονομασία index.htm ή index.html. Σε επόμενο κεφάλαιο θα αναλυθεί η πιο ειδική χρήση της και πως θα την χρησιμοποιήσουμε για την εφαρμογή μας.

1.3.2.2 CSS

Τα **CSS** (*Cascading Style Sheets – Μορφή Διαδοχικών Φύλλων*) ή (*αλληλουχία φύλλων*) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών τρόπου εμφάνισης φύλλων που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Πιο πρακτικά χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Ορίζει με λίγα λόγια τον τρόπο που θα εμφανιστούν ή τη μορφή που θα έχουν τα στοιχεία ενός εγγράφου γραμμένου σε HTML. Οι τρόποι/μορφές που ορίζουν τα CSS μπορούν να είναι ενσωματωμένα σε κάθε έγγραφο HTML που μορφοποιούν ή να είναι αποθηκευμένα σαν ξεχωριστό αρχείο με κατάληξη .css και να «δείχνουν» στο αρχείο που μορφοποιούν πετυχαίνοντας το ίδιο ακριβώς αποτέλεσμα. Το πλεονέκτημα της ξεχωριστής αποθήκευσης έγκειται στο γεγονός ότι το αρχείο αυτό μπορεί να χρησιμοποιηθεί και σε άλλα προς μορφοποίηση έγγραφα κάνοντας απλά μερικές μόνο τροποποιήσεις για να έχουμε το επιθυμητό αποτέλεσμα. Ακόμη έχοντας ένα μόνο αρχείο .css θα μπορούσαμε να εφαρμόσουμε κάποια στυλ σε όλες τις σελίδες του ιστότοπού μας.

1.3.2.3 JAVASCRIPT

Η JavaScript αποτελεί μία γλώσσα προγραμματισμού η οποία έχει σαν σκοπό την παραγωγή δυναμικού περιεχομένου και αλληλεπίδρασης σε ιστοσελίδες. Είναι ίσως η πιο δημοφιλής γλώσσα προγραμματισμού στο Internet και υποστηρίζεται από όλα τα κύρια προγράμματα περιήγησης (*browsers*) ενώ, επιπλέον, είναι ελεύθερη για χρήση από όλους δίχως την ανάγκη ειδικής άδειας. Η JavaScript δε θα πρέπει να συγχέεται με την Java ή οποία είναι διαφορετική γλώσσα προγραμματισμού και έχει διαφορετικές εφαρμογές. Ο τρόπος σύνταξης της JavaScript έχει βασιστεί στη γλώσσα προγραμματισμού C με την οποία παρουσιάζει πολλές ομοιότητες. Η επεξεργασία του κώδικα της JavaScript και η παραγωγή του τελικού περιεχομένου πραγματοποιείται στο πρόγραμμα περιήγησης των χρηστών σε αντίθεση με άλλες γλώσσες προγραμματισμού που είναι *server sided* και όλες οι επεξεργασίες γίνονται στην πλευρά του *server* για να παρουσιαστεί το αποτέλεσμα στον τελικό χρήστη. Αυτό, επί της ουσίας, σημαίνει ότι μπορεί κάλλιστα να ενσωματωθεί σε στατικές ιστοσελίδες ενώ έχει πολύ περιορισμένες δυνατότητες σε σχέση με τις *server sided* γλώσσες και δεν προσφέρει δυνατότητα σύνδεσης με βάσεις δεδομένων.

Ο κώδικας της JavaScript σε μία ιστοσελίδα περικλείεται από τις ετικέτες της HTML `<script language=javascript>` και `</script>` ενώ την απεικόνιση κειμένου αναλαμβάνει η εντολή `document.write`. Αν ο κώδικας JavaScript περιέχει περισσότερες από μία εντολές, αυτές θα πρέπει να διαχωριστούν μεταξύ τους με το χαρακτήρα του ελληνικού ερωτηματικού ';' (*δηλαδή της λατινικής άνω τελείας*). Η χρήση του χαρακτήρα αυτού για την τελευταία εντολή δεν είναι απαραίτητη. Η

διαχώριση των εντολών στα νεότερα προγράμματα περιήγησης δεν είναι απαραίτητη.

1.3.2.4 XML

Η γλώσσα XML (Extensible Markup Language) είναι ένας τρόπος με τον οποίο μπορείτε να επαναπροσδιορίσετε δεδομένα σε ένα αρχείο ή να αυτοματοποιήσετε τη διαδικασία αντικατάστασης των δεδομένων σε ένα αρχείο με δεδομένα από άλλο αρχείο. Η γλώσσα XML χρησιμοποιεί ετικέτες για να περιγράψει τμήματα ενός αρχείου, όπως μια επικεφαλίδα ή μια αλυσίδα κειμένου. Αυτές οι ετικέτες μαρκάρουν τα δεδομένα ώστε να είναι δυνατή η αποθήκευσή τους σε ένα αρχείο XML και ο κατάλληλος χειρισμός τους όταν εξάγονται σε άλλα αρχεία. Φανταστείτε τη γλώσσα XML ως ένα μηχανισμό μετάφρασης δεδομένων. Οι XML tag χαρακτηρίζουν κείμενο και άλλο περιεχόμενο σε ένα αρχείο ώστε οι εφαρμογές να αναγνωρίζουν και να παρουσιάζουν τα δεδομένα.

Η XML θεωρείται μια επεκτάσιμη γλώσσα διότι ο κάθε ένας μπορεί να δημιουργήσει τις δικές του XML tag, από μία ετικέτα για κάθε τύπο πληροφορίας που θέλει να επαναπροσδιορίσει. Οι XML tag δεν φέρουν πληροφορίες σχετικά με τον τρόπο που τα δεδομένα υποτίθεται ότι προβάλλονται ή μορφοποιούνται. Οι XML tag προορίζονται αυστηρά για αναγνώριση περιεχομένου.

1.3.3 Adobe Dreamweaver

Αναφέροντας στην παραπάνω παράγραφο όλες τις γλώσσες προγραμματισμού που θα χρησιμοποιηθούν καλό θα ήταν να αναφέρουμε και το που θα υλοποιηθεί όλος αυτός ο κώδικας. Γι αυτήν την υλοποίηση λοιπόν θα επιλέξουμε το Adobe Dreamweaver όπου εκτός από ένας εξαιρετικός text editor για την συγγραφή του κώδικα, έχει απίστευτα πολλές και εξαιρετικές δυνατότητες που θα μας δώσουν την δυνατότητα για ένα καλύτερο αποτέλεσμα.

Το Adobe Dreamweaver είναι ένα εργαλείο ανάπτυξης ιστοσελίδων της Macromedia. Υποστηρίζει CSS, JavaScript, και μια σειρά άλλων παρόμοιων τεχνολογιών. Το πρόγραμμα Dreamweaver βοηθά ακόμα και τους μη-προγραμματιστές (και αρχάριους) στην κατασκευή ιστοσελίδων καθώς έχει ενσωματωμένες πολλές τεχνικές σχεδιασμού. Παράδειγμα αποτελούν οι συναρτήσεις javascript οι οποίες είναι πολύ χρήσιμες σε κάθε εργασία παραγωγής web σελίδων. Ο κώδικας που παράγεται είναι βελτιστοποιημένος έτσι ώστε να προσφέρει το καλύτερο αποτέλεσμα.

Τέλος το DreamWeaver έχει δυνατότητες για δημιουργία δυναμικής HTML (DHMTL) και επιτρέπει κίνηση γραμμής χρόνου, απόλυτη τοποθέτηση περιεχομένων, δημιουργία επιπέδων (layers) και συγγραφή σεναρίων (scripts). Το DreamWeaver περιέχει δικές του συμπεριφορές (behaviors), που είναι έτοιμα scripts τα οποία μπορούμε να προσθέσουμε πολύ εύκολα σ' ένα αντικείμενο.

Το DreamWeaver μάς παρέχει την ελευθερία να σχεδιάσουμε οπτικά την εμφάνιση μιας ιστοσελίδας και τη δύναμη να την κάνουμε να λειτουργεί όπως ακριβώς θέλουμε. Μπορούμε να δημιουργήσουμε τη δική μας προσωπική ιστοσελίδα

(personal web page) ή μια ολόκληρη περιοχή (web site) σ' ένα εταιρικό δίκτυο (intranet).

1.3.4 Server

Εκτός λοιπόν από την υλοποίηση της ιστοσελίδας σε localhost με το XAMPP η εφαρμογή αφού κριθεί ότι είναι έτοιμη θα ανεβεί σε κάποιον online server για να παρουσιαστεί σε online περιβάλλον.

Θα χρησιμοποιηθεί ο server της **Free Web Hosting Area** που είναι διαθέσιμος στις σελίδες <http://www.freewebhostingarea.com> και <http://www.freewha.com>.

Λειτουργεί ανελλιπώς από το 2005 και έχει πολλά πλεονεκτήματα γι' αυτό και επιλέξαμε αυτόν, γιατί μας διαθέτει δωρεάν web space της τάξης των 15000MB με 12MB size limit.

Ο λογαριασμός δεν λήγει ποτέ και στα αρχεία μας γίνονται συνεχώς back up με αποτέλεσμα να μην υπάρχει κίνδυνος να χαθούν.

Υπάρχουν επίσης δωρεάν php hosting, MySQL και το PHPmyAdmin είναι από μόνο του εγκατεστημένο το οποίο κάνει και αυτό backup ανά πάση στιγμή στην βάση δεδομένων μας.

Επιλέγοντας βάση των παραπάνω αποφάσισα να αναρτήσω τον ιστότοπο μου με την εφαρμογή σε αυτόν τον server.

1.3.5 Μεθοδολογία RUP

Για τις ανάγκες λοιπόν της πτυχιακής εργασίας θα χρησιμοποιηθεί η μεθοδολογία της RUP (Rational Unified Process). Είναι ίσως μια από τις πιο διαδεδομένες μεθοδολογίες και αναπτύχθηκε από την εταιρία Rational. Βασίζεται στις αρχές του αντικειμενοστραφούς προγραμματισμού, και χρησιμοποιεί την UML σαν γλώσσα μοντελοποίησης για διαχείριση & ανάπτυξη λογισμικού, καλύπτοντας όλες τις φάσεις ενός τεχνολογικού έργου - από την Ανάλυση Απαιτήσεων μέχρι τη μετάβαση του συστήματος στο περιβάλλον των χρηστών. Η μεθοδολογία αυτή θα αναλυθεί εκτενέστερα στο 2^ο κεφάλαιο.

1.3.6 Visual Paradigm

Για την σχεδίαση των uml διαγραμμάτων θα χρησιμοποιηθεί το visual paradigm το οποίο είναι δωρεάν στο διαδίκτυο (free for non commercial use) και είναι πάρα πολύ εύχρηστο. Το Visual Paradigm είναι ένα εργαλείο CASE (Computer-Aided Software Engineering) που παρέχει εκτεταμένη μοντελοποίηση για διαγράμματα UML, διαγράμματα περιπτώσεων χρήσης (use case diagrams) και διαγράμματα οντοτήτων-συσχετίσεων. Τα στοιχεία του Visual Paradigm που του προσφέρουν ανταγωνιστικό πλεονέκτημα είναι: Η ευκολία στη χρήση, η αλληλεπίδρασή του με μερικά από τα πιο γνωστά IDE όπως το Visual Studio και το Eclipse και η πληθώρα προηγμένων λειτουργιών που υποστηρίζει μεταξύ των οποίων συμπεριλαμβάνονται οι εξής: Παραγωγή κώδικα από διαγράμματα κλάσεων,

Business Process Modeling και Reverse Engineering για κώδικα που έχει γραφτεί σε πάνω από 10 γλώσσες προγραμματισμού.

1.3.7 Open proj

Πέρα από την οπτική αναπαράσταση κάποιων διαγραμμάτων με το εργαλείο Visual Paradigm, θα χρησιμοποιηθεί και το εργαλείο open proj που θα μας δείξει οπτικοποιημένα την πορεία που πραγματοποιήθηκε για την διεκπεραίωση της εφαρμογής μας. Η οπτική λοιπόν αυτή προσέγγιση θα αναφερθεί στο κεφάλαιο 3 και θα δώσει ένα προσεγγιστικό χρονοδιάγραμμα των διαδικασιών που ακολουθήθηκαν για την διεκπεραίωση του στόχου μας, την υλοποίηση δηλαδή της εφαρμογής.

Το OpenProj είναι μία εφαρμογή για διαχείριση εργασιών από τον υπολογιστή μας, που χρησιμοποιεί την ίδια μηχανή σχεδιασμού που αξιοποιείται από το Project-ON-Demand. Μας επιτρέπει να ελέγχουμε, και να παρακολουθείτε και να διαχειριζόμαστε εργασίες που είναι σημαντικές για την υλοποίηση του έργου μας. Προσφέρει διάφορα εργαλεία που είναι πολύ αξιόπιστα για να έχουμε μία καλή διαχείριση των εργασιών μας. Ένα από αυτά είναι το Gantt chart όπου μπορούμε να τοποθετήσουμε τις εργασίες σας μαζί με τις ημερομηνίες έναρξης και τερματισμού. Το OpenProj μας επιτρέπει να οργανώσουμε και να παρακολουθούμε την εργασία μέσω των εργαλείων Task Usage και Resource Usage, ενώ το εργαλείο Report μας δίνει ενημερώσεις για την κατάστασή τους. Επίσης το interface του είναι διαισθητικό και έτσι δεν θα έχουμε κανένα πρόβλημα στη χρήση του.

1.4 Περιορισμοί της εφαρμογής

Οι περιορισμοί που μπορεί να έχει η εφαρμογή αφορούν περιορισμούς του HARDWARE αλλά και του SOFTWARE.

1.4.1 Περιορισμοί Hardware

Όσον αφορά το hardware η "ποσότητα" των τραγουδιών που θα υφίστανται εξαρτώνται από το αποθηκευτικό μέσο που θα χρησιμοποιηθεί (χωρητικότητα σκληρού δίσκου) ή κάποιον server. Το "μέγεθος" του server αλλά και η σύνδεση internet που υπάρχει (upload ADSL γραμμής) μπορούν να καθορίσουν ή καλύτερα να περιορίσουν την επισκεψιμότητα στην ιστοσελίδα και την αναπαραγωγή των τραγουδιών.

1.4.2 Περιορισμοί Software

Όσον αφορά το software, υπάρχουν περιορισμοί για τον web server και τον domain που θα δημιουργηθούν μέσω apache και php, και το πόσο μεγάλη θα μπορεί να είναι η βάση δεδομένων. Στα πλαίσια της πτυχιακής εργασίας δεν αντιμετωπίζουμε κάποιο ιδιαίτερο πρόβλημα όσον αφορά τους περιορισμούς, γιατί

δημιουργούμε μια σχετικά μικρή βάση δεδομένων για τις ανάγκες υλοποίησης της. Αν επιδιώκαμε να κάνουμε κάτι μεγαλύτερο σε μέγεθος ίσως να αντιμετωπίζαμε κάποιο πρόβλημα όσον αφορά το software αλλά και το hardware.

1.5 Νομιμότητα της εφαρμογής

Επειδή η υλοποίηση της πτυχιακής αυτής εργασίας έχει να κάνει με μουσικά κομμάτια πρέπει να λάβουμε στα υπ'όψιν μας το πόσο νόμιμη είναι όσον αφορά τα πνευματικά δικαιώματα των μουσικών τραγουδιών.

Για την νομιμότητα της εφαρμογής αυτής λοιπόν μετά από email αλλά και τηλεφωνική επικοινωνία με την ΑΕΡΙ (Ελληνική Εταιρεία Προστασίας της Πνευματικής Ιδιοκτησίας) δεν υπάρχει κάποιος περιορισμός εφόσον τα τραγούδια προέρχονται μετά από νόμιμη αγορά (πχ αυθεντικά cd).

Επίσης επειδή η εφαρμογή γίνεται στα πλαίσια πτυχιακής εργασίας (για καθαρά εκπαιδευτικούς λόγους δηλαδή) είτε πραγματοποιηθεί online είτε σε τοπικό δίκτυο δεν απαιτείται κάποια σχετική άδεια, αρκεί να παραμείνει online για σύντομο χρονικό διάστημα.

1.6 Επίλογος

Αφού κάναμε λοιπόν μια προσεγγιστική ανάλυση των μέσων και των εργαλείων που θα χρησιμοποιηθούν, στα επόμενα κεφάλαια που ακολουθούν θα υπάρξει εκτενέστερη ανάλυση. Στο Κεφάλαιο 2 που ακολουθεί θα αναλύσουμε την μεθοδολογία που ακολουθήθηκε για την πραγματοποίηση της εφαρμογής μας.

ΚΕΦΑΛΑΙΟ 2

2. Μεθοδολογία υλοποίησης της εφαρμογής

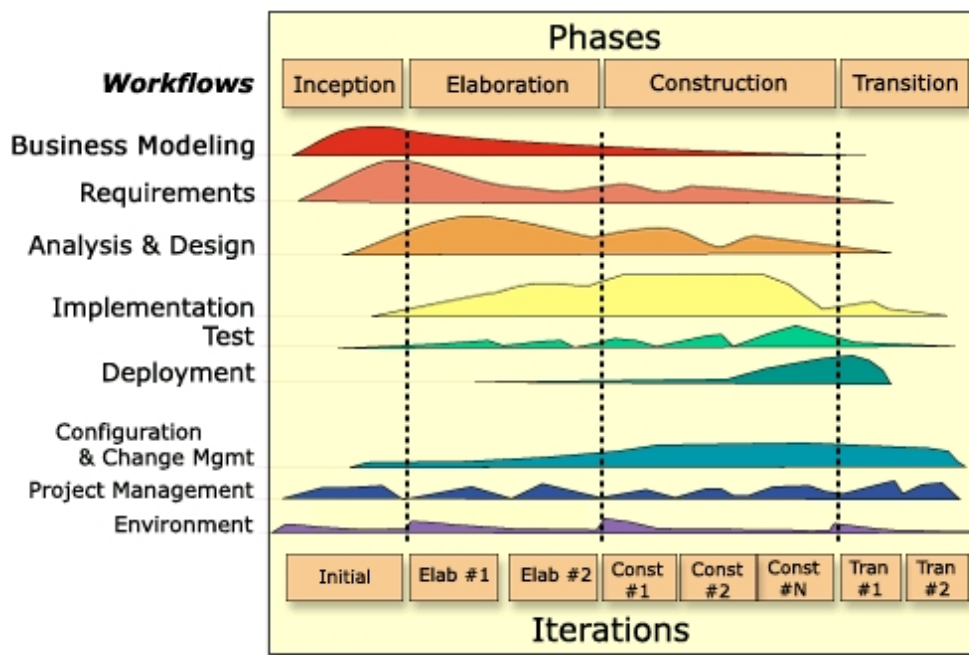
Σ' αυτό το κεφάλαιο λοιπόν θα αναλύσουμε την μεθοδολογία που χρησιμοποιήθηκε για την διεκπεραίωση της εφαρμογής και στο σύνολο της πτυχιακής εργασίας. Όπως αναφέραμε και στο πρώτο κεφάλαιο η μεθοδολογία που θα χρησιμοποιηθεί είναι η RUP.

2.1 Γενικά

Η μεθοδολογία υλοποίησής της εφαρμογής μας λοιπόν βασίζεται στη μεθοδολογία RUP. Η RUP (Rational Unified Process) αποτελεί μια μεθοδολογία λογισμικού η οποία αναπτύχθηκε από την εταιρεία Rational Software Corporation ενώ ανήκει πλέον ως βιομηχανικό πρότυπο στην IBM. Η RUP βασίζεται σε ένα σύνολο από αρχές ανάπτυξης λογισμικού όπως για παράδειγμα:

- Ανάπτυξη λογισμικού επαναληπτικά, δηλαδή τα μέρη του λογισμικού αναπτύσσονται, ελέγχονται και συμπληρώνονται διαρκώς.
- Χειρισμός των απαιτήσεων των χρηστών, δηλαδή καθορισμός των αναγκών των χρηστών και αλλαγή των απαιτήσεων όταν οι ανάγκες αυτές αλλάζουν.
- Χρήση αρχιτεκτονικής με components.
- Οπτική μοντελοποίηση του λογισμικού.
- Επαλήθευση της ποιότητας του λογισμικού.
- Έλεγχος των αλλαγών στο λογισμικό

Η διαδικασία ανάπτυξης ενός συστήματος λογισμικού, με βάση την μεθοδολογία Rational Unified Process, περιγράφεται σε δυο διαστάσεις ή σε δυο άξονες, όπως παριστάνεται και στο ακόλουθο Σχήμα.



Σχήμα 1. Φάσεις μεθοδολογίας RUP

Η γενική αρχιτεκτονική της RUP έχει δύο διαστάσεις:

- Ο οριζόντιος άξονας αντιπροσωπεύει τον χρόνο και δείχνει τα στοιχεία κύκλου ζωής της διαδικασίας καθώς εκτελείται. Αυτή η πρώτη διάσταση επεξηγεί την δυναμική φύση της διαδικασίας και εκφράζεται σε σχέση με φάσεις, επαναλήψεις και ορόσημα.
- Ο κατακόρυφος άξονας αντιπροσωπεύει τομείς που λογικά ομαδοποιούν δραστηριότητες με φυσικό τρόπο. Αυτή η δεύτερη διάσταση δείχνει την στατική φύση της διαδικασίας – πώς περιγράφεται σε σχέση με συστατικά διαδικασιών, τομείς, δραστηριότητες, ροές εργασίας, παραδοτέα και ρόλους.

Το σχεδιάγραμμα δείχνει πως η έμφαση μεταβάλλεται με τον χρόνο. Πχ: σε αρχικές επαναλήψεις αναλώνεται περισσότερος χρόνος σε απαιτήσεις ενώ σε επιπλέον επαναλήψεις, αναλώνεται περισσότερος χρόνος στην υλοποίηση.

2.2 Βασικές αρχές

Οι βασικές αρχές που διέπουν την μεθοδολογία ανάπτυξης είναι:

Επαναληπτική Ανάπτυξη Λογισμικού: Έχει αποδειχθεί ότι η ανάπτυξη πολύπλοκων συστημάτων λογισμικού είναι δύσκολη με το κλασικό γραμμικό μοντέλο όπου η ομάδα ανάπτυξης λογισμικού αρχικά ορίζει το πρόβλημα, σχεδιάζει τη λύση, αναπτύσσει το λογισμικό και τέλος ελέγχει το σύστημα. Απαιτείται μία επαναληπτική προσέγγιση η οποία θα επιτρέπει σταδιακή κατανόηση του προβλήματος και η ανάπτυξη της πιο αποτελεσματικής λύσης θα πραγματοποιείται με συγκεκριμένο αριθμό επαναλήψεων. Η επαναληπτική προσέγγιση υποστηρίζει την διαδικασία ανάπτυξης, αντιμετωπίζοντας υψηλού κινδύνου στοιχεία σε οποιοδήποτε στάδιο της εξέλιξης του έργου εμφανιστούν, μέσω δοκιμών των ενδιάμεσων εκδόσεων του συστήματος και της ανάδρασης από τους χρήστες, επομένως επιδρά σημαντικά στη μείωση των προβλημάτων του έργου συνολικά. Τυπικά, η κάθε επανάληψη ολοκληρώνεται με ένα πρωτότυπο (prototype), ώστε οι ομάδες ανάπτυξης να έχουν συνεχώς στόχο την παραγωγή αποτελεσμάτων, και ο περιοδικός έλεγχος της κατάστασης του συστήματος να επιβεβαιώνει ότι η διαδικασία ανάπτυξης είναι εντός χρονοδιαγράμματος. Στην περίπτωση της πτυχιακής εργασίας η εφαρμογή υλοποιήθηκε με αυτόν τον τρόπο, επαναληπτικά δηλαδή, αφού είχε παρουσιαστεί σε αρκετούς χρήστες και μέσω κυρίως σχολίων βελτιώθηκε και προστέθηκαν αρκετά πράγματα μέχρι να φτάσει στην τελική της μορφή.

Διαχείριση Απαιτήσεων: Η φάση αυτή αφορά τον τρόπο προσδιορισμού και οργάνωσης της λειτουργικότητας και των περιορισμών του έργου. Στο πρώτο κεφάλαιο αναφέρθηκαν οι απαιτήσεις και οι περιορισμοί της εφαρμογής μας. Οι απαιτήσεις βρέθηκαν μετά από σχετική έρευνα και μετά από συνέντευξη με τον επιβλέποντα καθηγητή της πτυχιακής εργασίας. Η διαδικασία αποκλεισμού κάποιων προβληματικών σεναρίων λειτουργίας αποδεικνύεται πολύ χρήσιμη

προκειμένου να προσδιοριστούν οι ακριβείς δυνατότητες του συστήματος σε σχέση με τις ανάγκες των χρηστών. Έτσι οι διαδικασίες ανάπτυξης, υλοποίησης και ελέγχου οδηγούν το σύστημα σε καταστάσεις όπου πληρούνται περισσότερο οι ανάγκες του τελικού χρήστη.

Χρήση Αρχιτεκτονικών βασισμένων σε Δομικά Στοιχεία (Component based Architectures): Εστιασμός στην ανάπτυξη ολοκληρωμένων συστημάτων τα οποία αξιοποιούν στο μέγιστο τη διαθεσιμότητα εμπορικών ή αξιόπιστων και ευρέως διαδεδομένων ανοικτού κώδικα δομικών στοιχείων. Στην εφαρμογή χρησιμοποιήθηκαν τέτοιου είδους components όπως ο music player και η καταγραφή στατιστικών στοιχείων.

2.3 Φάσεις Ανάπτυξης Λογισμικού

Όπως προκύπτει από το παραπάνω Σχήμα η διαδικασία ανάπτυξης λογισμικού αποτελείται από 4 διαδοχικές φάσεις:

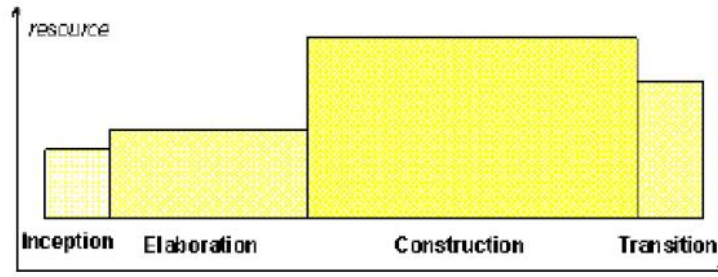
- Φάση Έναρξης (Inception) - Ορισμός του έργου και της έκτασής του
- Φάση Επεξεργασίας (Elaboration) - Κατάστρωση μεθόδου υλοποίησης του έργου, μοντελοποίηση χαρακτηριστικών του έργου, ορισμός της αρχιτεκτονικής του συστήματος.
- Φάση Κατασκευής (Construction) - Υλοποίηση του έργου
- Φάση Μετάβασης στο περιβάλλον του χρήστη (Transition) - Ανάπτυξη του συστήματος στο περιβάλλον χρήσης του

Κάθε φάση έχει προκαθορισμένο σκοπό και ολοκληρώνεται με ένα καλά ορισμένο ορόσημο (milestone), όπου αποφάσεις πρέπει να παρθούν και σημαντικά βήματα πρέπει να έχουν επιτευχθεί.

2.3.1 Σχεδιασμός των φάσεων

Κατά τον σχεδιασμό των φάσεων δεν είναι όλες οι φάσεις ίδιες από πλευράς χρονοπρογραμματισμού και προσπάθειας, και μπορούν να ποικίλουν σημαντικά ανάλογα με το είδος του έργου. Ένας τυπικός αρχικός κύκλος ανάπτυξης για ένα έργο μεσαίου μεγέθους θα μπορούσε να θεωρήσει την ακόλουθη κατανομή χρονοπρογραμματισμού και προσπάθειας. Αυτό θα αποτελέσει και την κατανομή χρονοπρογραμματισμού και για την εφαρμογή που θα υλοποιήσουμε.

	Inception	Elaboration	Construction	Transition
Effort	~5%	20%	65%	10%
Schedule	10%	30%	50%	10%



Σχήμα 2. Κατανομή χρονοπρογραμματισμού και προσπάθειας

2.3.2 Φάση έναρξης (Inception)

Κατά τη διάρκεια της φάσης έναρξης, η ομάδα του έργου καθορίζει το επιχειρησιακό μοντέλο για το σύστημα λογισμικού και σκιαγραφεί το πλαίσιο του έργου. Προκειμένου αυτό να επιτευχθεί, η ομάδα πρέπει να προσδιορίσει όλες τις εξωτερικές οντότητες με τις οποίες θα αλληλεπιδρά το σύστημα καθώς και τη φύση της αλληλεπίδρασης. Αυτό, συνιστά τον προσδιορισμό περιπτώσεων χρήσης (use-cases) και την περιγραφή των πιο σημαντικών. Το επιχειρησιακό μοντέλο περιλαμβάνει τα κριτήρια επιτυχίας, εκτίμηση των κινδύνων και προσδιορισμό των πόρων που θα χρειαστούν καθώς και πλάνο των φάσεων με τις σημαντικότερες ημερομηνίες του έργου (milestones).

Τα αποτελέσματα της φάσης έναρξης περιλαμβάνουν:

- Επιχειρησιακό μοντέλο,
- Αρχικές απαιτήσεις χρήστη: βασικές απαιτήσεις του έργου, βασικά χαρακτηριστικά και κύριοι περιορισμοί,
- Αρχικό μοντέλο περιπτώσεων χρήσης (use-case),
- Αρχική εκτίμηση των κινδύνων,
- Πλάνο τεχνικής υλοποίησης,
- Ανάπτυξη πρωτοτύπου (prototype), σε περίπτωση μεγάλων έργων.

Το σημαντικότερο ορόσημο αυτής της φάσης αφορά τους Στόχους του κύκλου ζωής του λογισμικού.

2.3.3 Φάση επεξεργασίας (Elaboration)

Ο σκοπός αυτής της φάσης είναι να αναλυθεί το πεδίο προβλήματος, να θεμελιωθεί η αρχιτεκτονική του συστήματος και να εξαλειφθούν τα υψηλού

κινδύνου στοιχεία του έργου. Αποφάσεις σχετικά με την αρχιτεκτονική του συστήματος λογισμικού πρέπει να παρθούν έχοντας κατανοήσει πλήρως το πλαίσιο του συστήματος, τις βασικές λειτουργίες του και τις μη λειτουργικές απαιτήσεις, όπως απαιτήσεις απόδοσης.

Τα αποτελέσματα της φάσης αυτής περιλαμβάνουν:

- Σχεδόν ολοκληρωμένο μοντέλο περιπτώσεων χρήσης (use-case model) στο οποίο όλες οι περιπτώσεις χρήσεις και οι εμπλεκόμενοι (actors) έχουν προσδιοριστεί και οι περιγραφές των περιπτώσεων χρήσης έχουν αναπτυχθεί,
- Επιπρόσθετες απαιτήσεις σχετικά με μη λειτουργικές απαιτήσεις και με απαιτήσεις που δεν σχετίζονται με κάποια συγκεκριμένη περίπτωση χρήσης,
- Περιγραφή της αρχιτεκτονικής του συστήματος λογισμικού,
- Μια αναθεωρημένη λίστα με τους πιθανούς κινδύνους,
- Αρχικά εγχειρίδια χρήσης

Σε περιπτώσεις πολύπλοκων και μεγάλων έργων, σε αυτή τη φάση αναπτύσσεται ένα εκτελέσιμο αρχέτυπο (prototype) της αρχιτεκτονικής το οποίο περιλαμβάνει τις κρίσιμες περιπτώσεις χρήσης, που εγκυμονούν τους σημαντικούς τεχνικούς κινδύνους για το έργο.

Το ορόσημο αυτής της φάσης αφορά την αρχιτεκτονική του συστήματος. Η ομάδα έργου εξετάζει λεπτομερειακά το πλαίσιο και τους στόχους του συστήματος, την επιλογή της αρχιτεκτονικής και την επίλυση των σημαντικότερων κινδύνων.

2.3.4 Φάση κατασκευής (Construction)

Στη φάση αυτή όλα τα επιμέρους χαρακτηριστικά και οι δομές της εφαρμογής που προσδιορίστηκαν στα προηγούμενα στάδια αναπτύσσονται, ενοποιούνται και ελέγχονται σχολαστικά. Στη φάση της κατασκευής δίνεται έμφαση στη σωστή διαχείριση των πόρων, στον έλεγχο και συγχρονισμό των επιμέρους λειτουργιών που πραγματοποιούνται, στην ελαχιστοποίηση του κόστους, στα χρονοδιαγράμματα και στην ποιότητα.

Το αποτέλεσμα αυτής της φάσης είναι ένα σύστημα το οποίο μπορεί να χρησιμοποιηθεί από τους τελικούς χρήστες.

Περιλαμβάνει, στο ελάχιστο:

- Το λογισμικό του συστήματος ενοποιημένο στις κατάλληλες πλατφόρμες,
- Τα εγχειρίδια χρήσης,
- Περιγραφή της τρέχουσας έκδοσης του λογισμικού.

Το ορόσημο είναι η Αρχική Λειτουργική Δυνατότητα του συστήματος. Σε αυτό το σημείο, η έκδοση του συστήματος ονομάζεται beta έκδοση.

2.3.5 Φάση μετάβασης (Transition)

Σκοπός αυτής της φάσης είναι η παράδοση του συστήματος λογισμικού στους τελικούς χρήστες. Αφού το σύστημα παραδοθεί, επιπλέον εργασίες μπορούν να προκύψουν όπως διόρθωση λαθών, βελτίωση των χαρακτηριστικών του συστήματος ή προσδιορισμός νέων χαρακτηριστικών για μελλοντικές εκδόσεις του λογισμικού. Η φάση της μετάβασης ξεκινά αφού το σύστημα έχει φτάσει στην κατάσταση εγκατάστασης στον τελικό χρήστη. Αυτό τυπικά απαιτεί την ολοκλήρωση του βασικού συστήματος σε κάποιο αποδεκτό επίπεδο το οποίο διασφαλίζει πλήρως την ποιότητα. Επίσης έχουν ετοιμαστεί τα εγχειρίδια χρήσης ώστε η μετάβαση του συστήματος στον τελικό χρήστη να διασφαλίζει ικανοποιητικά αποτελέσματα για όλες τις ομάδες.

Αυτό περιλαμβάνει:

- Beta έλεγχο έτσι ώστε να επιβεβαιωθεί ότι ικανοποιεί τις προσδοκίες των χρηστών,
- Εκπαίδευση χρηστών και προσωπικού που θα κάνει την συντήρηση του συστήματος,
- Το σύστημα δίνεται στην παραγωγή,
- Σε περίπτωση αντικατάστασης υπάρχοντος συστήματος, δοκιμάζεται η παράλληλη λειτουργία των δύο συστημάτων.

Η φάση της μετάβασης επικεντρώνεται σε εργασίες που απαιτούνται έτσι ώστε το σύστημα να παραδοθεί στους τελικούς χρηστές. Καταβάλλεται σημαντική προσπάθεια ώστε να αναπτυχθούν κατανοητά και ευκολόχρηστα εγχειρίδια χρήσης, να εκπαιδευτούν οι χρήστες, να υποστηριχθούν στην πρώτη επαφή τους με το προϊόν, και να ληφθεί υπόψη και να χρησιμοποιηθεί η ανάδραση από αυτούς.

Το ορόσημο αυτής της φάσης ονομάζεται Έκδοση Συστήματος Λογισμικού.

2.4 Επαναλήψεις

Κάθε μία από τις παραπάνω τέσσερις φάσεις μπορεί να εκτελεστεί σε επαναλήψεις. Η κάθε επανάληψη οδηγεί σε μία έκδοση η οποία μπορεί να είναι είτε εσωτερική είτε εξωτερική. Στην πρώτη το αποτέλεσμα της επανάληψης μένει εντός της ομάδας ανάπτυξης ενώ στην εξωτερική το αποτέλεσμα επιδεικνύεται στους χρήστες με σκοπό την λήψη αντιδράσεων και σχολίων. Η κάθε επανάληψη αποδίδει ένα σύστημα, το οποίο είναι υποσύνολο του τελικού, και δημιουργείται αυξητικά από επανάληψη σε επανάληψη για να δημιουργηθεί στο τέλος το τελικό σύστημα.

Σε σύγκριση με την παραδοσιακή γραμμική διαδικασία, η επαναληπτική διαδικασία έχει τα εξής πλεονεκτήματα:

- Οι κίνδυνοι μετριάζονται πιο νωρίς,
- Οι αλλαγές είναι πιο εύκολα διαχειρίσιμες,
- Το επίπεδο επαναχρησιμοποίησης είναι υψηλότερο,
- Η ομάδα του έργου έχει τη δυνατότητα να μαθαίνει κατά τη διάρκεια εξέλιξης του έργου,
- Το τελικό προϊόν χαρακτηρίζεται από υψηλή ποιότητα.

Μετά την ανάλυση λοιπόν της μεθοδολογίας που θα χρησιμοποιηθεί, στο επόμενο κεφάλαιο θα αναφερθούμε αναλυτικά και εκτενέστερα στον σχεδιασμό και στην ανάλυση της εφαρμογής μας.

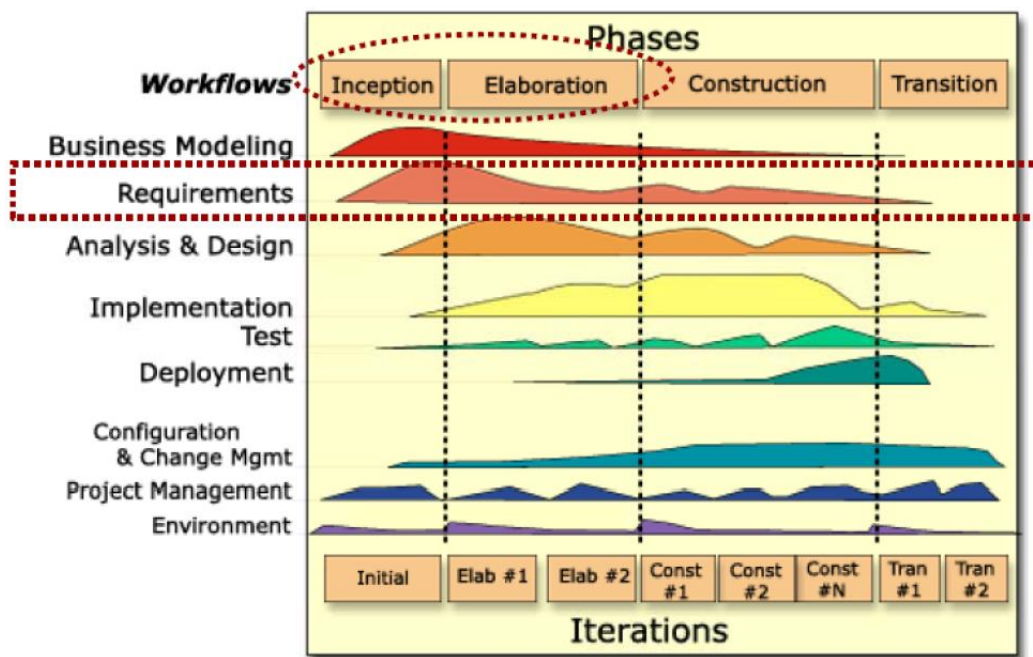
ΚΕΦΑΛΑΙΟ 3

3. Ανάλυση και σχεδίαση

Το κεφάλαιο τρίτο που αφορά την ανάλυση και την σχεδίαση όσον αφορά την υλοποίηση της εφαρμογής μας θα αποτελέσει έναν συνδετικό κρίκο ανάμεσα στο δεύτερο κεφάλαιο που αναπτύχθηκε παραπάνω και στο τέταρτο που θα αναπτυχθεί αμέσως μετά. Πώς δηλαδή θα περάσουμε από τα εργαλεία και την ανάλυση της μεθοδολογίας που θα χρησιμοποιηθεί στην υλοποίηση του κώδικα. Αυτό θα πραγματοποιηθεί με την παρουσίαση κάποιων βασικών διαγραμμάτων (use case, sequence diagram), με την παρουσίαση των περιπτώσεων χρήσεων αλλά και αναλύοντας κάποιες σημαντικές τεχνικές όπως το use case estimation effort (εκτίμηση προσπάθειας).

3.1 Μοντέλο περιπτώσεων χρήσης

Με βάση την μεθοδολογία RUP που αναλύθηκε στο δεύτερο κεφάλαιο θα προχωρήσουμε στην συνέχεια, όπου με βάση τα χαρακτηριστικά της θα αναπτύξουμε τις περιπτώσεις χρήσης που διακρίνονται. Πρώτα από όλα θα πρέπει να δημιουργηθεί ένας πίνακας γνωστός και ως “πίνακας χρηστών στόχων”. Στην φάση αυτή βρισκόμαστε ακόμα στο αρχικό στάδιο της έναρξης (inception), με την ανάλυση των απαιτήσεων και μπορούμε να το διακρίνουμε στην παρακάτω εικόνα.



Σχήμα 3. Στιγμιότυπο πορείας της μεθοδολογίας RUP κατά την φάση της έναρξης

3.1.1 Πίνακας χρηστών στόχων

Οι πελάτες και οι τελικοί χρήστες έχουν στόχους γνωστούς ως ανάγκες οι οποίοι περιμένουν να εκπληρωθούν από τα συστήματα υπολογιστών. Ο καλύτερος τρόπος διατύπωσης & τεκμηρίωσης θα πρέπει να είναι απλός και οικείος με τους εμπλεκόμενους, συμβάλλοντας έτσι τόσο στον ορισμό-σύνταξη τους όσο και στην αξιολόγησή τους.

Με βάση λοιπόν τις ανάγκες των χρηστών δημιουργούμε τον παρακάτω πίνακα.

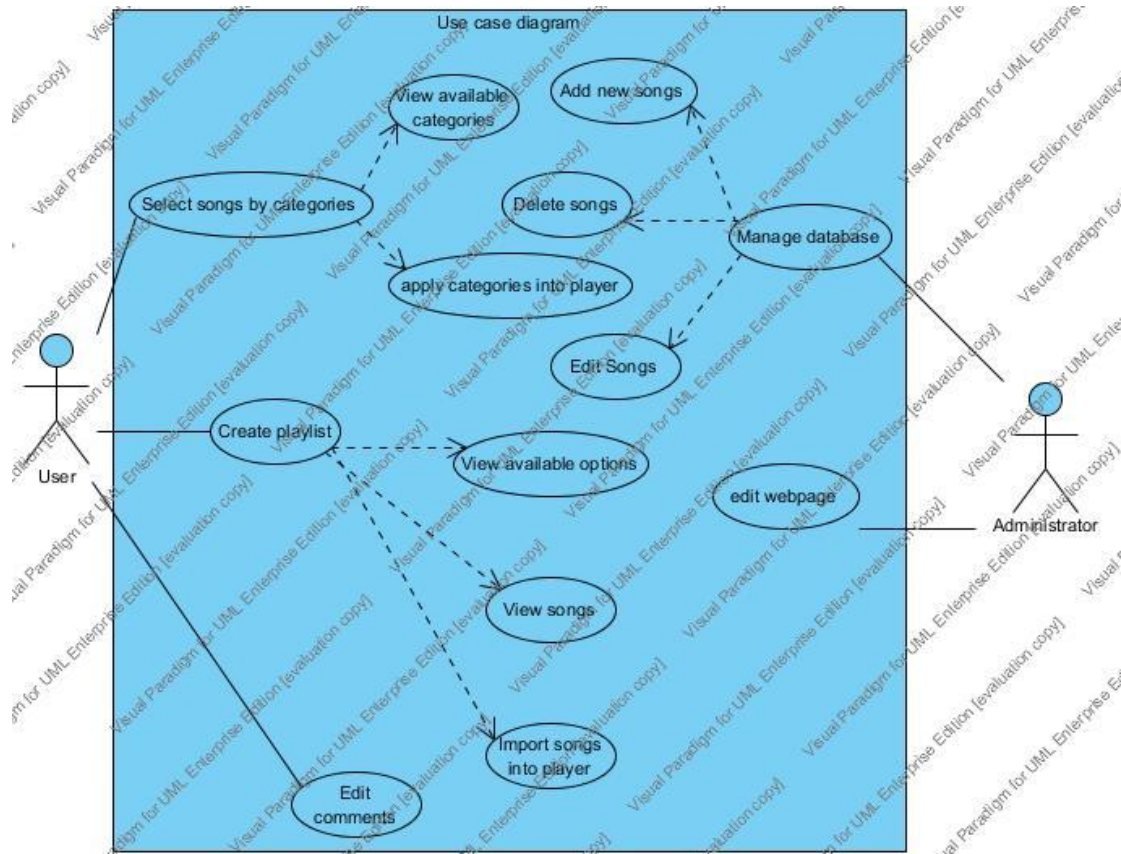
Πίνακας 1. Πίνακας χρηστών - στόχων

Χρήστης	Στόχος
Απλός χρήστης (user)	Είσοδος - έξοδος από την σελίδα Αναπαραγωγή τραγουδιών Επιλογή τραγουδιών βάση φίλτρων Επιλογή τραγουδιών ανά μουσικές ενότητες (λειτουργία ραδιοφώνου) Επικοινωνία με τον διαχειριστή του συστήματος
Διαχειριστής συστήματος (Administrator)	Προσθήκη τραγουδιών στην βάση Τροποποίηση πληροφοριών των τραγουδιών Διαγραφή τραγουδιών Τροποποίηση ιστοσελίδας (edit σελίδας) Διαχείριση πινάκων συστήματος

Με βάση λοιπόν τον παραπάνω πίνακα προχωράμε στην συνέχεια όπου πρέπει να δημιουργήσουμε ένα βασικό διάγραμμα το use case diagram το οποίο θα μας οδηγήσει τελικά στην ανάπτυξη των περιπτώσεων χρήσης.

3.1.2 Use case diagram

Στα πλαίσια λοιπόν του μοντέλου περιπτώσεων χρήσης λαμβάνοντας υπόψιν και τον παραπάνω πίνακα θα δημιουργήσουμε το use case diagram, το οποίο σχεδιάζεται βέβαια όπως και όλα μας τα διαγράμματα με το visual paradigm. Το βλέπουμε λοιπόν παρακάτω.



Σχήμα 4. Use case diagram

Αφού δημιουργήσαμε το παραπάνω διάγραμμα λοιπόν είμαστε έτοιμοι να αναλύσουμε τις περιπτώσεις χρήσης που διακρίνονται και που καθορίζονται όπως έχουμε πει με βάση τις ανάγκες των χρηστών.

3.1.3 Περιπτώσεις χρήσης

Οι περιπτώσεις χρήσης λοιπόν ή αλλιώς οι προδιαγραφές των απαιτήσεων, είναι ένας μηχανισμός ανακάλυψης και καταγραφής των λειτουργικών απαιτήσεων ενός συστήματος. Αποτελούν το κομβικό σημείο της ανάλυσης και ένας ορισμός τους θα μπορούσε να ήταν μια εργασία που εκτελείται από ένα άτομο σε ένα μέρος και σε μια στιγμή, ως απάντηση ενός επιχειρηματικού γεγονότος, το οποίο προσθέτει επιχειρηματική αξία και αφήνει τα δεδομένα σε μια συνεπή κατάσταση.

Οι περιπτώσεις χρήσης μπορούν να συνταχθούν με 3 τρόπους.

- Συνοπτική (μια παράγραφος)
- Συνήθης (άτυπη μορφή μερικών παραγράφων)
- Πλήρους ανάπτυξης

Στοχεύουμε στην καλύτερη δυνατή ανάλυση της εφαρμογή μας οπότε θα χρησιμοποιήσουμε την πλήρους ανάπτυξης σύνταξη η οποία είναι πολύ αναλυτική και δομημένη και όλα τα βήματα και οι παραλλαγές συντάσσονται με λεπτομέρεια.

Με βάση των use case diagram λοιπόν διακρίνουμε τρεις περιπτώσεις χρήσεων οι οποίες θα αναλυθούν στη συνέχεια.

- ΠΧ1 Αναπαραγωγή τραγουδιών με δημιουργία λίστας (create playlist)
- ΠΧ2 Αναπαραγωγή τραγουδιών ανά ενότητες (select songs by categories)
- ΠΧ3 Περίπτωση χρήσης 3 - Επικοινωνία με τον Administrator (Edit comments)

3.1.3.1 Περίπτωση χρήσης ΠΧ1

- **Περίπτωση Χρήσης ΠΧ1** : Αναπαραγωγή τραγουδιών με δημιουργία λίστας (create playlist)
- **Κύριος χρήστης** : Απλός χρήστης(user)
- **Σύντομη περιγραφή** : Ο χρήστης κάνει εισαγωγή στην σελίδα και επιλέγει να ακούσει τραγούδια online με συγκεκριμένη λίστα την οποία ο ίδιος είναι υπεύθυνος να δημιουργήσει.
- **Προϋποθέσεις**:
 - ✓ Ο χρήστης κάνει είσοδο στην σελίδα.
 - ✓ Ο χρήστης επιλέγει την καρτέλα “Ακούστε μουσική”
- **Κύριο σενάριο επιτυχίας (ή Βασική Ροή)** :
 1. Ο χρήστης βρίσκεται στην φόρμα Αναπαραγωγή τραγουδιών.
 2. Ο χρήστης επιλέγει τραγούδια για να ακούσει με βάση συγκεκριμένα κριτήρια τα οποίος ο ίδιος επιλέγει βάση των επιλογών που υπάρχουν στην σελίδα.
 3. Ο χρήστης πατάει το πλήκτρο submit που υπάρχει.
 4. Το σύστημα ανακτά από την βάση δεδομένων τα συγκεκριμένα τραγούδια και τα εμφανίζει.
 5. Το σύστημα αναπαράγει (μέσω του mp3 player) τα τραγούδια που έχουν επιλεγεί.
- **Επεκτάσεις (ή Εναλλακτικές ροές)**
 - 2α. Ο χρήστης θέλει να ξαναεπιλέξει μουσικές ενότητες.
 1. Ο χρήστης πατάει το πλήκτρο reset αντι για το submit.
 2. Το σύστημα “καθαρίζει” τις επιλογές του και του δίνει την δυνατότητα να ξαναεπιλέξει τις μουσικές ενότητες που επιθυμεί.
 - 1α. Ο χρήστης βλέπει το περιεχόμενο της σελίδας και κάνει έξοδο από αυτήν.
 - 4α. Δεν βρέθηκαν τραγούδια με τις συγκεκριμένες επιλογές.
 1. Το σύστημα ελέγχει στην βάση δεδομένων ότι δεν υπάρχουν τραγούδια με τους μουσικούς “περιορισμούς” τους οποίους έθεσε ο χρήστης.

2. Εμφανίζεται ένα μήνυμα στην ιστοσελίδα ότι δεν υπάρχουν τραγούδια και ο χρήστης καλείται να ξαναεπιλέξει.
 - 5α. Ο χρήστης αποφασίζει να διακόψει την αναπαραγωγή των τραγουδιών.
 - 5β. Ο χρήστης διακόπτει την ομαλή και συνεχόμενη αναπαραγωγή των τραγουδιών
 1. Επιλέγει να αναπαράγει ξανά το τραγούδι που είχε αναπαραχθεί μόλις από τον player.
 2. Επιλέγει να παρακάμψει κάποιο από τα τραγούδια της δημιουργημένης λίστας, και να προχωρήσει την αναπαραγωγή κάποιου άλλου τραγουδιού.
 - 5γ. Ο χρήστης μέσω της φόρμας δημιουργεί καινούρια λίστα με αποτέλεσμα η παλιά λίστα να σταματήσει να υφίσταται και να αρχίσει η αναπαραγωγή των τραγουδιών της καινούριας λίστας.

3.1.3.2 Περίπτωση χρήσης ΠΧ2

- **Περίπτωση Χρήσης ΠΧ2** : Αναπαραγωγή τραγουδιών ανά ενότητες (select songs by categories)
- **Κύριος χρήστης** : Απλός χρήστης(user)
- **Σύντομη περιγραφή** : Ο χρήστης κάνει εισαγωγή στην σελίδα και επιλέγει να ακούσει τραγούδια ανά ενότητες που προυπάρχουν στη σελίδα και η κάθε μία διαρκεί 30 λεπτά.
- **Προϋποθέσεις:**
 - ✓ Ο χρήστης κάνει είσοδο στην σελίδα.
 - ✓ Ο χρήστης επιλέγει την καρτέλα “*Ακούστε μουσική*”
- **Μετασυνθήκες:** Ενημερώνονται τα στατιστικά των προτιμήσεων των χρηστών.
- **Κύριο σενάριο επιτυχίας (ή Βασική Ροή)** :
 1. Ο χρήστης βρίσκεται στην φόρμα Αναπαραγωγή τραγουδιών ανά ενότητες.
 2. Ο χρήστης επιλέγει κάποια μουσική ενότητα απο αυτές που προυπάρχουν στην φόρμα.
 3. Το σύστημα ανακτά από την βάση δεδομένων τις μουσικές ενότητες με τραγούδια από 30 λεπτά έκαστη και τα εμφανίζει .
 4. Το σύστημα αναπαράγει (μέσω του mp3 player) τα τραγούδια που έχουν επιλεγεί.
- **Επεκτάσεις (ή Εναλλακτικές ροές)**
 - 1α. Ο χρήστης βλέπει το περιεχόμενο της σελίδας και κάνει έξοδο από αυτήν.

- 4α. Ο χρήστης αποφασίζει να διακόψει την αναπαραγωγή των τραγουδιών.
- 4β. Ο χρήστης διακόπτει την ομαλή και συνεχόμενη αναπαραγωγή των τραγουδιών.
 - Επιλέγει να αναπαράγει ξανά το τραγούδι που είχε αναπαραχθεί μόλις από τον player.
 - Επιλέγει να παρακάμψει κάποιο από τα τραγούδια της δημιουργημένης λίστας, και να προχωρήσει στην αναπαραγωγή κάποιου άλλου τραγουδιού.
- 4γ. Ο χρήστης βλέπει τα τραγούδια που αναπαράγονται και θέλει να ξαναεπιλέξει την ίδια κατηγορία.
- 4δ. Ο χρήστης αποφασίζει να επιλέξει την αρχή των μουσικών ενοτήτων με άλλη ενότητα τελικά.

3.1.3.3 Περίπτωση χρήσης ΠΧ3

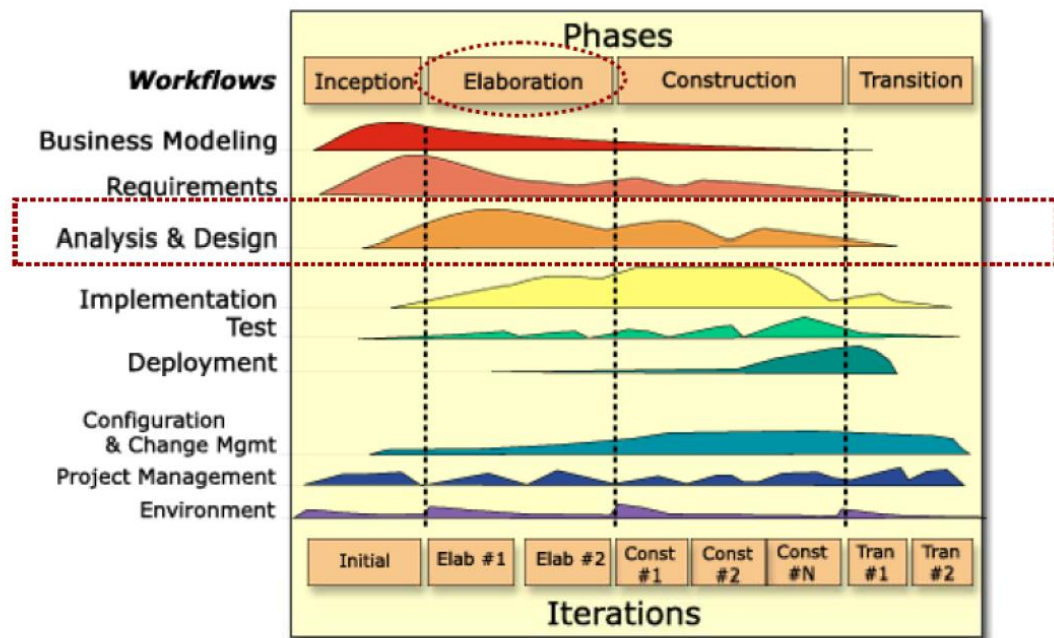
- **Περίπτωση Χρήσης ΠΧ3** : Επικοινωνία με τον Administrator (edit comments)
- **Κύριος χρήστης** : Απλός χρήστης
- **Σύντομη περιγραφή** : Ο χρήστης κάνει εισαγωγή στην σελίδα για να δει το περιεχόμενο της, και μέσω της φόρμας επικοινωνίας επικοινωνεί με τον διαχειριστή του συστήματος.
- **Προϋποθέσεις:**
 - ✓ Ο χρήστης κάνει είσοδο στην σελίδα.
 - ✓ Ο χρήστης επιλέγει την καρτέλα “Επικοινωνία”
- **Κύριο σενάριο επιτυχίας (ή Βασική Ροή)** :
 1. Ο χρήστης βρίσκεται στην φόρμα επικοινωνία με τον διαχειριστή
 2. Ο χρήστης εισάγει τα στοιχεία του στην παραπάνω φόρμα και τα σχόλια του για τον διαχειριστή.
 3. Ο χρήστης πατάει αποστολή.
 4. Το σύστημα αποστέλνει τα στοιχεία και τα σχόλια του χρήστη μέσω email στον διαχειριστή.
 5. Το σύστημα εμφανίζει στην οθόνη του χρήστη ότι η αποστολή του μηνύματος έγινε επιτυχώς.
- **Επεκτάσεις (ή Εναλλακτικές ροές)**
 - 1 α. Ο χρήστης βλέπει το περιεχόμενο της σελίδας και κάνει έξοδο από αυτήν.
 - 5 α. Το μήνυμα δεν έχει πάει επιτυχώς γιατί έγινε κάποιο λάθος κατά την συμπλήρωση στοιχείων από τον χρήστη.
 1. Το σύστημα αναγνωρίζει ότι υπάρχει λάθος στην συμπλήρωση των στοιχείων του χρήστη κατά την αποστολή.

2. Εμφανίζεται ένα μήνυμα στην σελίδα για τον χρήστη ότι η παράδοση του μηνύματος ήταν ανεπιτυχής.

Αυτή ήταν η ανάλυση των τριών περιπτώσεων χρήσης που καθορίστηκαν και στην συνέχεια θα σχεδιαστεί για αυτές ένα ακόμα απο τα σημαντικότερα διαγράμματα του αντικειμενοστραφούς προγραμματισμού, το διάγραμμα ακολουθίας.

3.2 Διάγραμμα ακολουθίας(Sequence diagram)

Σε αυτήν λοιπόν την φάση έχουμε φτάσει πια στην φάση της επεξεργασίας (elaboration) και αυτό μπορούμε να το δούμε στην παρακάτω εικόνα.



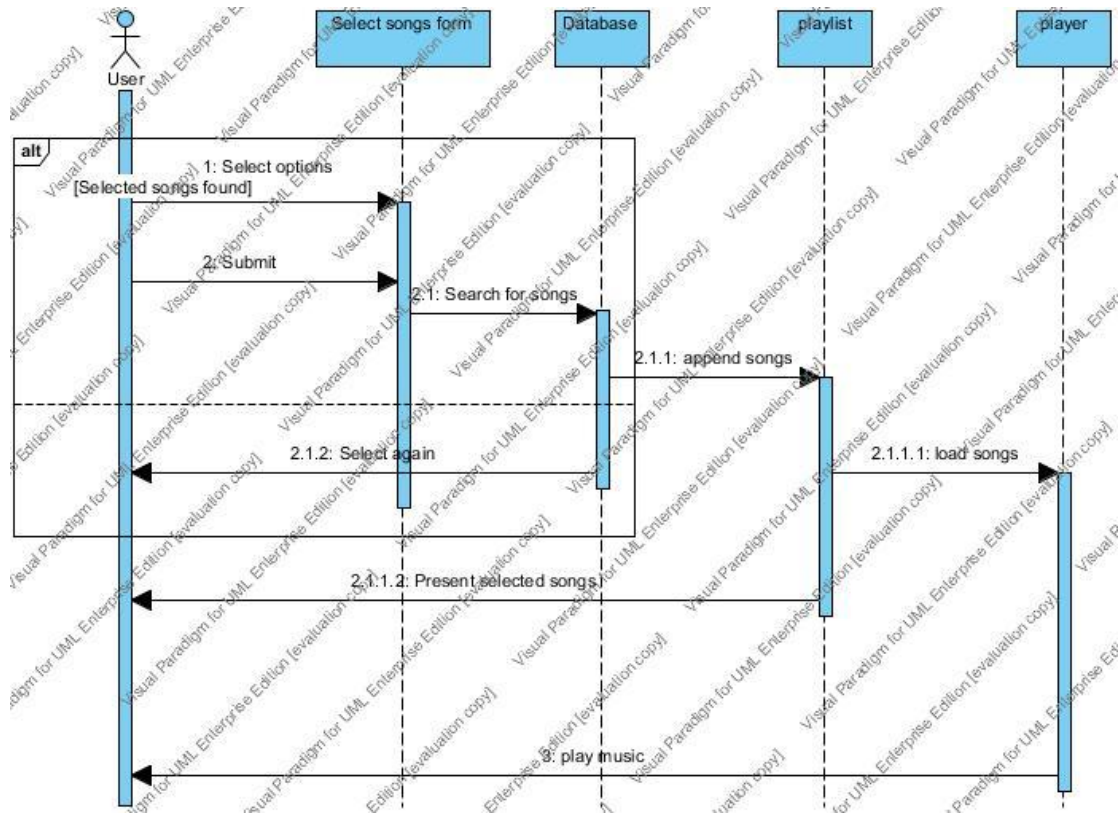
Σχήμα 5. Στιγμιότυπο πορείας της μεθοδολογίας RUP κατά την φάση της επεξεργασίας

Στην φάση αυτή θα δημιουργήσουμε τα διαγράμματα ακολουθίας για κάθε περίπτωση χρήσης.

Τα διαγράμματα ακολουθίας δείχνουν πώς συνεργάζεται ένα σύνολο οντοτήτων για την επίτευξη μιας συγκεκριμένης συμπεριφοράς. Η συμπεριφορά που περιγράφεται σε ένα διάγραμμα ακολουθίας αντιστοιχεί συνήθως σε ένα απλό σενάριο χρήσης.

3.2.1 Διάγραμμα ακολουθίας για την περίπτωση χρήσης PX1

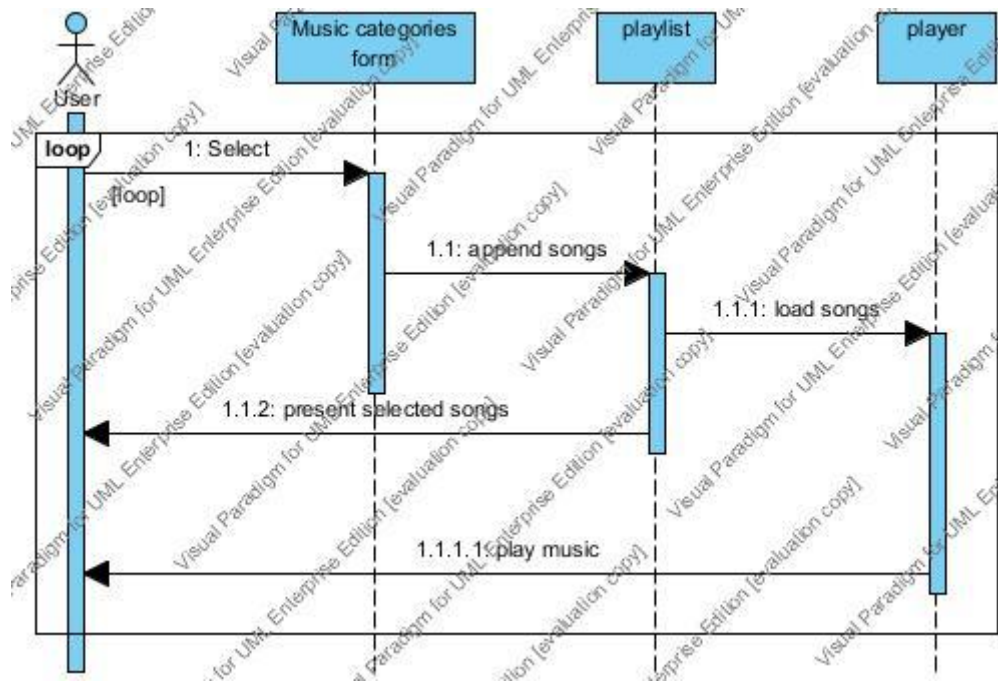
Στο διάγραμμα που ακολουθεί λοιπόν μπορούμε να δούμε υλοποιημένη υπό μορφή την πρώτη περίπτωση χρήσης.



Σχήμα 6. Διάγραμμα ακολουθίας για την ΠΧ1

3.2.2 Διάγραμμα ακολουθίας για την περίπτωση χρήσης ΠΧ1

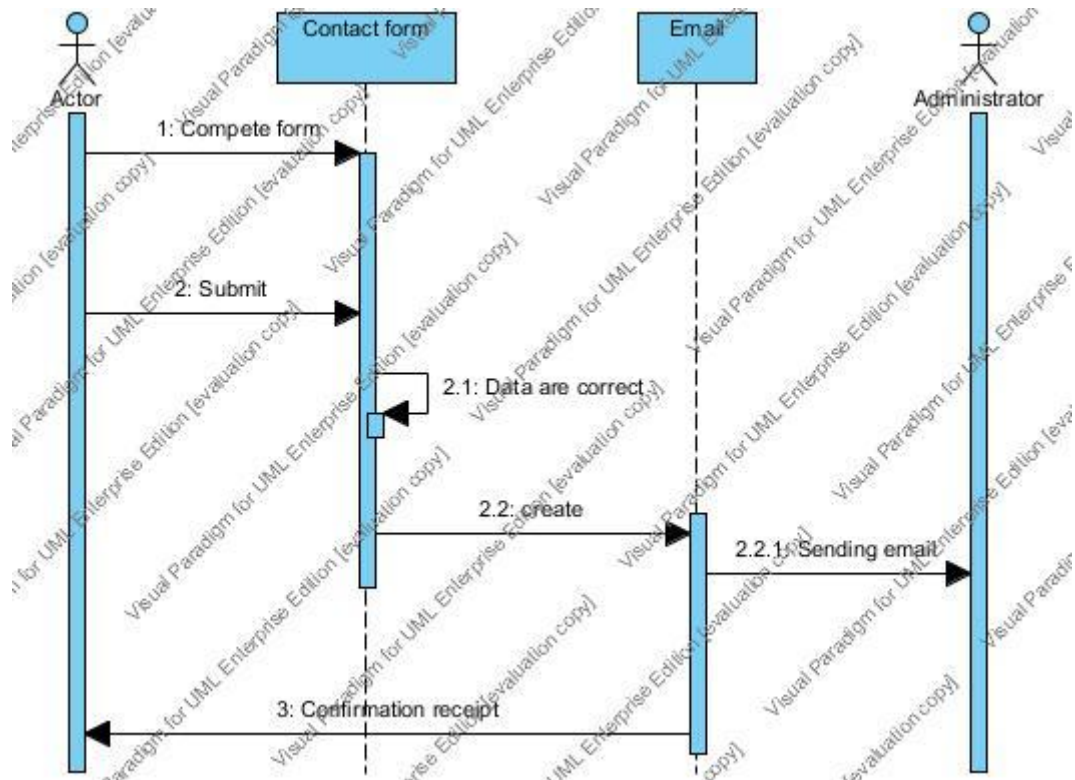
Με την σειρά του βλέπουμε και το δεύτερο διάγραμμα ακολουθίας.



Σχήμα 7. Διάγραμμα ακολουθίας για την ΠΧ2

3.2.3 Διάγραμμα ακολουθίας για την περίπτωση χρήσης ΠΧ3

Και τέλος θα δούμε το διάγραμμα ακολουθίας που αναπαριστά την τρίτη περίπτωση χρήσης.



Σχήμα 8. Διάγραμμα ακολουθίας για την ΠΧ3

Τελειώνοντας και με την παρουσίαση των διαγραμμάτων ακολουθίας έχουμε ολοκληρώσει κατά πολύ το στάδιο της επεξεργασίας. Στη συνέχεια με δύο εργαλεία θα ολοκληρώσουμε το κεφάλαιο μας αυτό περνώντας πια στο τελευταίο στάδιο της υλοποίησης. Τα δύο εργαλεία που αναφερθήκαμε λοιπόν είναι το use case estimation effort (εκτίμηση προσπάθειας) και το open proj.

3.3 Use case estimation effort (εκτίμηση προσπάθειας)

Για να μπορέσει να γίνει ένας σχετικός έλεγχος του χρόνου και του βαθμού δυσκολίας της προς ανάπτυξη εφαρμογής υπάρχει μια μέθοδος που ονομάζεται Use Case Points βάσει της οποίας μπορεί να γίνει μια εκτίμηση της προσπάθειας που απαιτείται, πριν ξεκινήσει η ανάπτυξη της εφαρμογής.

Έτσι το 1993 αναπτύχθηκε από τον Gustav Karner η παρούσα μέθοδος, η οποία λαμβάνοντας υπόψη διάφορους παράγοντες, χρησιμοποιεί μια εξίσωση για την εκτίμηση του χρόνου που χρειάζεται.

Ο χρόνος ολοκλήρωσης μιας εφαρμογής επηρεάζεται από τους παρακάτω παράγοντες:

1. Τον αριθμό βημάτων για να ολοκληρωθούν οι περιπτώσεις χρήσης.
2. Ο αριθμός και η πολυπλοκότητα των χρηστών.

3. Οι τεχνικές απαιτήσεις των περιπτώσεων χρήσης, όπως οι παράλληλες εφαρμογές, η ασφάλεια και η επίδοση και
4. Διάφοροι εξωτερικοί παράγοντες όπως η εμπειρία και οι γνώσεις της ομάδας ανάπτυξης της εφαρμογής.

Την εξίσωση συνθέτουν τέσσερις μεταβλητές:

1. Ο παράγοντας τεχνικής πολυπλοκότητας (TCF),
2. Ο παράγοντας εξωγενούς πολυπλοκότητας (ECF),
3. Σημεία αστάθμητου βάρους στις περιπτώσεις χρήσης (UUCP) και
4. Ο παράγοντας παραγωγικότητας (PF).

Η κάθε μεταβλητή ορίζεται και υπολογίζεται χωριστά και τέλος η εξίσωση έχει τη μορφή:

$$\mathbf{UCP = TCF * ECF * UUCP * PF}$$

Τα απαραίτητα βήματα είναι:

1. Να αποφασίσουμε και να υπολογίσουμε τους τεχνικούς παράγοντες,
2. Να αποφασίσουμε και να υπολογίσουμε τους περιβαλλοντικούς παράγοντες,
3. Να υπολογίσουμε τα αστάθμητου βάρους σημεία των περιπτώσεων χρήσης,
4. Να αποφασίσουμε ποιος είναι ο παράγοντας παραγωγικότητας και
5. Να υπολογίσουμε το τελικό αποτέλεσμα-προϊόν των μεταβλητών.

Καλό θα ήταν να επισημάνουμε ότι σε αυτό το section χρησιμοποιούνται πάρα πολύ δύσκολοι όροι στην αγγλική γλώσσα και θα προσπαθήσουμε να τους αποδώσουμε όσο καλύτερα γίνεται. Μερικοί όροι θα παραμείνουν ως έχουν.

3.3.1 TCF (Technical Complexity Factors)

Οι Technical Complexity Factors λοιπόν (ή παράγοντες τεχνικής πολυπλοκότητας) είναι δεκατρείς στο σύνολο τους, και υπολογίζουν τα διάφορα τεχνικά θέματα που μπορεί να έχει μια εφαρμογή. Ο κάθε παράγοντας καθορίζεται με μια ειδική τιμή (weight) και αφορά το κατά πόσο μπορεί να επηρεάσει την εφαρμογή. Οι τιμές που μπορεί να πάρει ο κάθε ένας από αυτούς κυμαίνεται από 0 έως 5 όπου το 0 αντιπροσωπεύει έναν παράγοντα που δεν έχει καμιά επιρροή στο έργο μας και το 5 ότι παίζει τον πιο σημαντικό ρόλο. Οι παράγοντες αυτοί με την τιμή που τους χαρακτηρίζει φαίνονται στο παρακάτω πίνακα.

Πίνακας 2. Τεχνικοί παράγοντες

Τεχνικός παράγων	Περιγραφή	Βαρύτητα (weight)
T1	Κατανεμημένο σύστημα (Distributed system)	2
T2	Απόδοση (Performance)	1
T3	End User Efficiency	1
T4	Πολυπλοκότητα εξωτερικής επεξεργασίας (Complex internal Processing)	1
T5	Επαναχρησιμοποίηση (Reusability)	1
T6	Ευκολία εγκατάστασης (Easy to install)	0.5
T7	Ευκολία χρήσης (Easy to use)	0.5
T8	Μεταφερισιμότητα (Portable)	2
T9	Ευκολία αλλαγής (Easy to change)	1
T10	Ταυτόχρονη επεξεργασία (Concurrent)	1
T11	Ιδιαίτερα χαρακτηριστικά ασφάλειας (Special security features)	1
T12	Παρέχει άμεση πρόσβαση σε τρίτα μέρη (Provides direct access for third parties)	1
T13	Απαιτήση ειδικού εξοπλισμού εκπαίδευσης χρηστών (Special user training facilities are required)	1

Για κάθε έργο λοιπόν οι τεχνικοί παράγοντες αξιολογούνται από την ομάδα ανάπτυξης λογισμικού με μια άλλη τιμή (perceived complexity) η οποία καθορίζει την σχετική πολυπλοκότητα του κάθε παράγοντα σε σχέση με το έργο. Βέβαια στην περίπτωση της εφαρμογής μας εμείς είμαστε αυτοί που θα καθορίσουν την ειδική τιμή αυτή. Οι τιμές είναι πάλι από το 0 μέχρι το 5. Το 0 σημαίνει ότι δεν υπάρχει καθόλου πολυπλοκότητα, το 3 σημαίνει μέτρια, ενώ το 5 μεγάλη. Στη συνέχεια για τον κάθε παράγοντα πολλαπλασιάζουμε την τιμή της βαρύτητας του (weight) με την τιμή της πολυπλοκότητας (perceived complexity) και παράγεται ο calculated factor. Το άθροισμα όλων των τιμών που θα βρεθούν αποτελεί τον συνολικό παράγοντα (Total Factor). Όλη αυτή η διαδικασία φαίνεται στον παρακάτω πίνακα.

Πίνακας 3. Διαδικασία υπολογισμού συνολικού τεχνικού παράγοντα

Technical Factor	Description	Weight	Perceived Complexity	Calculated Factor (weight*perceived complexity)>
T1	Distributed System	2	3	6
T2	Performance	1	2	2
T3	End User Efficiency	1	1	1
T4	Complex internal Processing	1	1	1
T5	Reusability	1	2	2
T6	Easy to install	0.5	0	0
T7	Easy to use	0.5	0	0

Technical Factor	Description	Weight	Perceived Complexity	Calculated Factor (weight*perceived complexity)>
T8	Portable	2	2	4
T9	Easy to change	1	2	2
T10	Concurrent	1	5	5
T11	Special security features	1	2	2
T12	Provides direct access for third parties	1	0	0
T13	Special user training facilities are required	1	0	0
			Total Factor	26

Ο συνολικός παράγοντας είναι 26 που προκύπτει από την άθροιση των επιμέρους όρων. Για να προκύψει το τελικό TCF, υπάρχει η παρακάτω σχέση:

$$TCF = 0.6 + (0.01 * \text{Total Factor}) \rightarrow TCF = 0.6 + (0.01 * 26) \rightarrow TCF = 0.86$$

3.3.2 ECF (Environmental Complexity Factors)

Στη συνέχεια προχωράμε υπολογίζοντας τους παράγοντες εξωγενής πολυπλοκότητας, τους παράγοντες δηλαδή από το περιβάλλον που επηρεάζουν το έργο μας. Παίρνουν τιμές από 0 έως και 5, με το 0 να σημαίνει ότι οι παράγοντες δεν παίζουν κάποιο σημαντικό ρόλο, το 3 σημαίνει μέτρια συμμετοχή, ενώ το 5 έχουν έντονη επιρροή. Καθορίζονται και αυτοί από ειδική τιμή βαρύτητας και το βλέπουμε στον παρακάτω πίνακα.

Πίνακας 4. Περιβαλλοντικοί παράγοντες

Παράγοντας Περιβάλλοντος	Περιγραφή	Βαρύτητα
E1	Οικειότητα με UML (Familiarity with UML)	1.5
E2	Εμπειρία εφαρμογής (Application Experience)	0.5
E3	Εμπειρία στην Αντικειμενοστρέφεια (Object Oriented Experience)	1
E4	Ικανότητα αρχή-αναλυτή (Lead analyst capability)	0.5
E5	Κίνητρα (Motivation)	1
E6	Σταθερές απαιτήσεις (Stable Requirements)	2
E7	Υπάλληλοι μερικής απασχόλησης (Part-time workers)	-1
E8	Δυσκολία γλώσσας προγράμμου (Difficult Programming language)	2

Όπως και στους TCF έτσι και εδώ θα αποδώσουμε τις ειδικές τιμές για κάθε παράγοντα και θα υπολογιστεί ο συνολικός παράγοντας. Αναλυτικά μπορούμε να το δούμε παρακάτω.

Πίνακας 5. Διαδικασία υπολογισμού συνολικού περιβαλλοντικού παράγοντα

Environmental Factor	Description	Weight	Perceived Impact	Calculated Factor (weight*perceived complexity)
E1	Familiarity with UML	1.5	3	4.5
E2	Application Experience	0.5	3	1.5
E3	Object Oriented Experience	1	3	3
E4	Lead analyst capability	0.5	3	1.5
E5	Motivation	1	1	1
E6	Stable Requirements	2	4	8
E7	Part-time workers	-1	0	0
E8	Difficult Programming language	2	3	6
			Total Factor	25.5

Το συνολικό αποτέλεσμα είναι 25.5 προκύπτει από το άθροισμα των παραπάνω επιμέρους στοιχείων. Για να προκύψει το συνολικό ECF έχουμε:

$$ECF = 1.4 + (-0.03 * \text{Total Factor}) \rightarrow ECF = 1.4 + (-0.03 * 25.5) \rightarrow ECF = 0.635$$

3.3.3 UUCP (Unadjusted Use Case Points)

Στην συνέχεια θα υπολογίσουμε τα UUCP (προσπαθώντας να το αποδώσουμε στα ελληνικά θα ήταν σημεία αστάθμητου βάρους στις περιπτώσεις χρήσης). Προκύπτουν από τον υπολογισμό δύο άλλων μεγεθών του UUCW (Unadjusted Use Case Weight) που βασίζεται στο συνολικό αριθμό των δραστηριοτήτων που περιέχονται στις περιπτώσεις χρήσης και του UAW (Unadjusted Actor Weight) που βασίζεται στην πολυπλοκότητα του χρήστη στην κάθε περίπτωση χρήσης.

Πρώτα λοιπόν καλούμαστε να υπολογίσουμε τα UUCW όπου οι περιπτώσεις χρήσης καθορίζονται με βάση την πολυπλοκότητα τους και αυτό φαίνεται στον παρακάτω πίνακα. Διακρίνονται σε simple, average, και complex και έχουν να κάνουν ανάλογα με το πόσα βήματα χρειάζεται η κάθε περίπτωση χρήσης για να φτάσει σε ένα επιτυχημένο σενάριο. Και σε αυτήν την περίπτωση δεν αποδόθηκαν στα ελληνικά για να μην χάσουν την ακριβή επεξήγηση τους.

Πίνακας 6. Κατηγορίες περιπτώσεων χρήσης

Use Case Type	Description	Weight
Simple	A simple user interface and touches only a single database entity; its success scenario has 3 steps or less; its implementation involves less than 5 classes.	5
Average	More interface design and touches 2 or more database entities; between 4 to 7 steps; its implementation involves between 5 to 10 classes.	10
Complex	Involves a complex user interface or processing and touches 3 or more database entities; over seven steps; its implementation involves more than 10 classes.	15

Στη συνέχεια έχουμε τον χαρακτηρισμό στην κάθε από τις τρεις περιπτώσεις χρήσης μας όπου το άθροισμα τους θα μας δώσει το μέγεθος του UUCW.

Πίνακας 7. Διαδικασία υπολογισμού συνολικού UUCW

Περίπτωση Χρήσης	Ταξινόμηση	Βάρος
Περίπτωση χρήσης 1 - Αναπαραγωγή τραγουδιών (create playlist)	AVERAGE	10
Περίπτωση χρήσης 2- Αναπαραγωγή τραγουδιών ανά ενότητες	AVERAGE	10
Περίπτωση χρήσης 3 - Επικοινωνία με τον Administrator	SIMPLE	5
	Total UUCW	25

Με έναν παρόμοιο τρόπο χαρακτηρίζονται και οι χρήστες σε κάθε περίπτωση χρήσης και με βάση τον παρακάτω πίνακα καλούμαστε να υπολογίσουμε το έταιρο μέγεθος UAW.

Πίνακας 8. Κατηγορίες χρηστών

Actor Type	Description	Weight
Simple	The Actor represents another system with a defined API.	1
Average	The Actor represents another system interacting through a protocol, like TCP/IP.	2
Complex	The Actor is a person interacting via an interface.	3

Υπολογίζουμε λοιπόν το συνολικό UAW λαμβάνοντας υπόψιν το παραπάνω πίνακα και έχουμε τα εξής αποτελέσματα.

Πίνακας 9. Διαδικασία υπολογισμού συνολικού UAW

Actor Type	Description	Weight	Number of Actors	Result
Simple	The Actor represents another system with a defined API	1	1	1
Average	The Actor represents another system interacting through a protocol, like TCP/IP	2	0	0
Complex	The Actor is a person interacting via an interface.	3	2	6
			Total UAW	7

Το UUCP υπολογίζεται προσθέτοντας το UUCW και το UAW. Εδώ συγκεκριμένα έχουμε:

$$UUCP = 25 + 7 = 32.$$

3.3.4 Productivity Factor

Ο παράγοντας παραγωγικότητας είναι μια αναλογία των ωρών εργασίας ανά περίπτωση χρήσης βασιζόμενη σε παλιότερες περιπτώσεις. Αν δεν υπάρχουν παλιότερα στοιχεία, δίνεται μια τιμή ανάμεσα στο 15 και στο 30, με συνήθως προτεινόμενη τιμή το 20.

3.3.5 Υπολογισμός UCP

Το Use Case Points υπολογίζεται πολλαπλασιάζοντας τις παραπάνω τιμές:

$$UCP = TCP * ECF * UUCP * PF$$

Για τη δεδομένη περίπτωση:

$$UCP = 0.86 * 0.635 * 32 * 20 = 238.3874 \text{ ή αλλιώς } 349,5 \text{ ώρες.}$$

Διαιρώντας την παραπάνω τιμή δια του 40 (για μια εβδομάδα εργασίας ενός ανθρώπου) παίρνουμε την τιμή 8.73, δηλαδή σχεδόν 9 εβδομάδες εργασίας .

Αυτή ήταν λοιπόν η εκτίμηση προσπάθειας που μας δίνει ένα θεωρητικό χρονοδιάγραμμα για το σε πόσο χρόνο περίπου θα υλοποιηθεί το έργο μας.

Στην επόμενη και τελευταία παράγραφο του τρίτου κεφαλαίου θα παρουσιαστεί με την βοήθεια του open proj ένα “actual” χρονοδιάγραμμα που χρησιμοποιήθηκε παράλληλα με την πορεία υλοποίησης του έργου μας.

3.4 Open proj

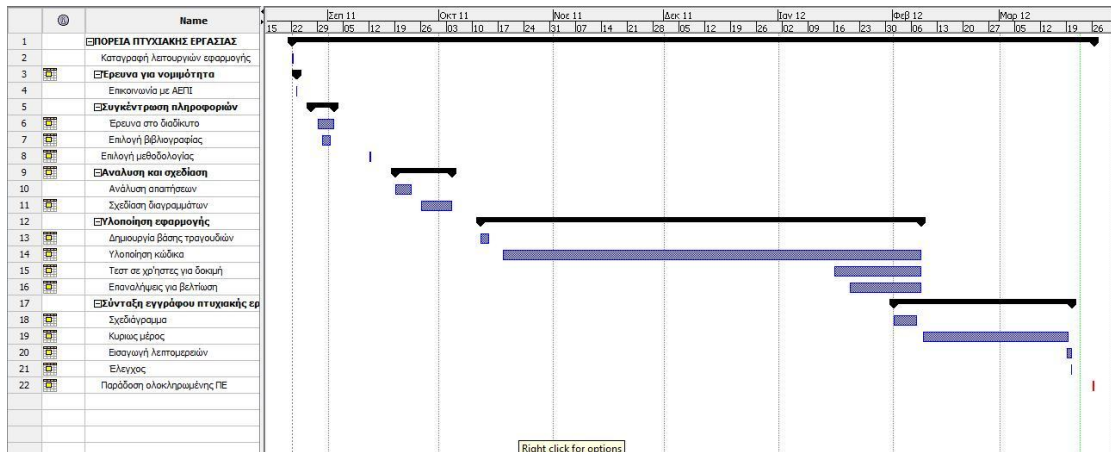
Το open proj είναι ένα εργαλείο όπως αναφέρθηκε και στο πρώτο κεφάλαιο που θα μας βοηθήσει να οπτικοποιήσουμε την πορεία υλοποίησης του έργου μας βάσει χρονοδιαγράμματος.

Σε αυτήν λοιπόν την παράγραφο θα παρουσιαστούν πρώτα το σύνολο των εργασιών που έγινε για την υλοποίηση του έργου μας με τις εξαιρετικές δυνατότητες φυσικά που μας δίνει το Open proj.

		Name	Duration	Start
1		ΠΟΡΕΙΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ	156 days	22/8/2011 8:00 πμ
2		Καταγραφή λειτουργιών εφαρμογής	1 day	22/8/2011 8:00 πμ
3		Ερευνα για νομιμότητα	1 day	23/8/2011 8:00 πμ
4		Επικοινωνία με ΑΕΠΙ	1 day	23/8/2011 8:00 πμ
5		Ευγέντρωση πληροφοριών	5 days	27/8/2011 8:00 πμ
6		Ερευνα στο διαδίκτυο	5 days	27/8/2011 8:00 πμ
7		Επιλογή βιβλιογραφίας	3 days	30/8/2011 8:00 πμ
8		Επιλογή μεθοδολογίας	1 day	10/9/2011 8:00 πμ
9		Ανάλυση και σχεδίαση	12 days	19/9/2011 8:00 πμ
10		Ανάλυση απαιτήσεων	5 days	19/9/2011 8:00 πμ
11		Σχεδίαση διαγραμμάτων	7 days	24/9/2011 8:00 πμ
12		Υλοποίηση εφαρμογής	86 days	12/10/2011 8:00 πμ
13		Δημιουργία βάσης τραπεζιδίων	3 days	12/10/2011 8:00 πμ
14		Υλοποίηση κώδικα	82 days	18/10/2011 8:00 πμ
15		Τεστ σε χρήστες για δοκιμή	18 days	16/1/2012 8:00 πμ
16		Επανάληψης για βελτίωση	14 days	20/1/2012 8:00 πμ
17		Εύνταξη εγγράφου πτυχιακής εργασίας	35 days	1/2/2012 8:00 πμ
18		Σχεδιάγραμμα	5 days	1/2/2012 8:00 πμ
19		Κυρίως μέρος	28 days	9/2/2012 8:00 πμ
20		Εισαγωγή λεπτομερειών	2 days	18/3/2012 8:00 πμ
21		Έλεγχος	1 day	20/3/2012 8:00 πμ
22		Παράδοση ολοκληρωμένης ΠΕ	1 day	26/3/2012 8:00 πμ

Σχήμα 9. Χρονοδιάγραμμα υλοποίησης ΠΕ

Βέβαια αυτό δεν είναι η μοναδική δυνατότητα που μας δίνει το open proj και στη συνέχεια θα παρουσιαστούν και άλλα διαγράμματα. Το παραπάνω μπορούμε να το δούμε και υπό μορφή timeline με το παρακάτω διάγραμμα Gantt.



Σχήμα 10. Διάγραμμα Gantt

3.5 Επίλογος

Φτάνοντας λοιπόν στο τέλος του τρίτου κεφαλαίου είμαστε έτοιμοι να προχωρήσουμε στο τέταρτο και τελευταίο μας κεφάλαιο που αφορά την υλοποίηση του έργου μας. Αφού κάναμε μια καλή ανάλυση έχουμε καλυψει κατά το ήμισι την μεθοδολογία που ακολουθούμε (RUP) και θα την ολοκληρώσουμε στο επόμενο κεφάλαιο.

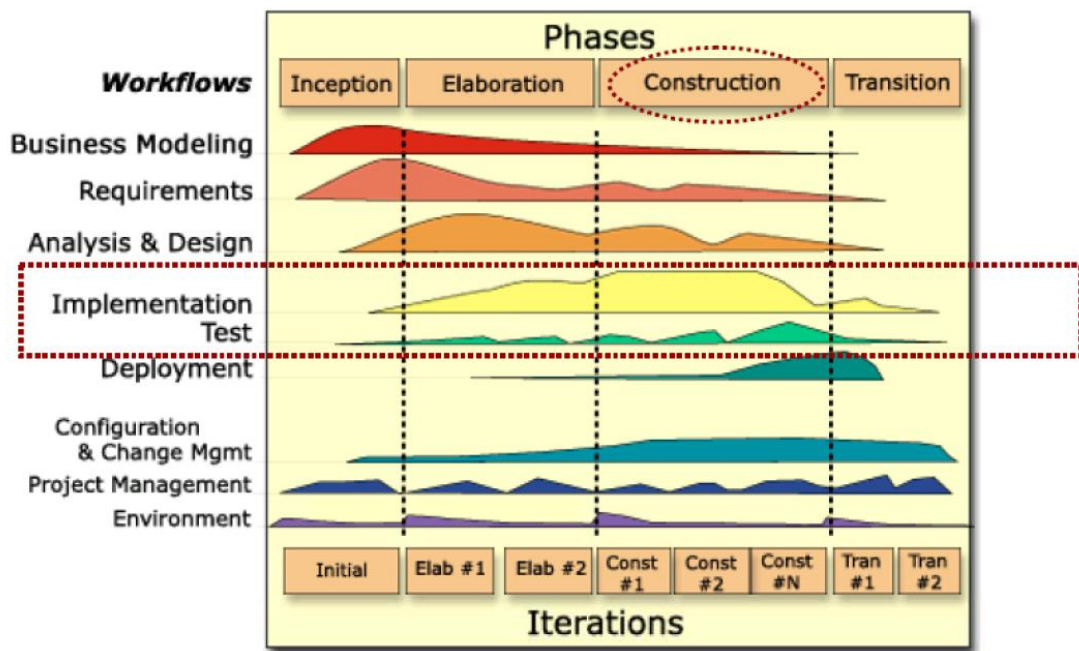
ΚΕΦΑΛΑΙΟ 4

4. Υλοποίηση

Στα προηγούμενα κεφάλαια αναφερθήκαμε αποκλειστικά στα εργαλεία και στην μεθοδολογία που θα χρησιμοποιηθεί για την υλοποίηση και στον σχεδιασμό στο τρίτο κεφάλαιο αλλά δεν αναφερθήκαμε καθόλου στον τρόπο υλοποίησης. Αυτό καλούμαστε λοιπόν να κάνουμε στο τέταρτο και τελευταίο κεφάλαιο όπου θα ασχοληθούμε αποκλειστικά στην πορεία υλοποίησης, στις γλώσσες που χρησιμοποιήθηκαν, και στο πως χρησιμοποιήθηκαν αναφέροντας σημαντικά κομμάτια του κώδικα που υλοποιήθηκε.

Πλέον βρισκόμαστε πια στην φάση της κατασκευής, όπου θα πραγματοποιήσουμε επαναληπτική προσέγγιση κάνοντας συνεχείς ελέγχους σε όλη τη διάρκεια του έργου στοχεύοντας κυρίως στην εύρεση λαθών όσο πιο νωρίς γίνεται.

Στην παρακάτω εικόνα βλέπουμε ένα στιγμιότυπο για το που βρισκόμαστε όσον αφορά την πορεία της μεθοδολογίας μας.



Σχήμα 11. Στιγμιότυπο πορείας της μεθοδολογίας RUP κατά την φάση της κατασκευής

4.1 Δημιουργία αρχικής σελίδας

Ξεκινάμε λοιπόν την κατασκευή της ιστοσελίδας – εφαρμογής δημιουργώντας την αρχική σελίδα που θα αντικρύζει ο χρήστης μπαίνοντας στον ιστότοπο που θα δημιουργήσαμε. Όπως αναφέραμε και στο πρώτο κεφάλαιο μετά από την πραγματοποίηση της σε local host θα αναρτηθεί σε online server και το όνομα της σελίδας θα είναι <http://my-thesis-anaskara.orgfree.com>. Η αρχική σελίδα λοιπόν

απαρτίζεται από το αρχείο index.php και είναι σχεδιασμένο με βασικά στοιχεία της γλώσσας HTML.

4.1.1 Σχεδίαση

Με τα απλά και βασικά στοιχεία της HTML λοιπόν δημιουργούμε την κεντρική σελίδα. Μπορούμε να δούμε στο παρακάτω απόσπασμα κώδικα τις βασικές πληροφορίες που δημιουργούνται με την HTML όπως ο τίτλος της ιστοσελίδας, το χρώμα, κάποιες εικόνες που θα χρησιμοποιηθούν , και την εικόνα που θα αποτελέσει το background.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΚΑΡΑΜΑΡΓΙΟΣ ΑΝΑΣΤΑΣΙΟΣ</title>
6 <link rel="stylesheet" type="text/css" href="menub.css"/>
7 </head>
8
9 <body bgcolor="#BAABAB">
10 <div id="left">
11 
12 
13 </div>
14
15 <h1></h1>
16
17 <body background="background site.jpg">
18
```

Στην συνέχεια θα πρέπει να δημιουργήσουμε ένα μενού καρτελών που θα κάνει την πλοήγηση του χρήστη πιο εύκολη και πιο ευχάριστη.

4.1.2 Δημιουργία μενού

Δημιουργούμε λοιπόν το μενού καρτελών, οι οποίες καρτέλες θα είναι οι εξής :

- Αρχική σελίδα
- Ακούστε μουσική
- Λίστα Τραγουδιών
- Επικοινωνία

Η αρχική σελίδα λοιπόν όπως έχουμε αναφέρει απαρτίζεται από το αρχείο index.php, η επόμενη καρτέλα («Ακούστε μουσική») αποτελείται από το κεντρικό player.php και άλλα php αρχεία που θα δούμε αναλυτικά παρακάτω , η λίστα τραγουδιών από το mp3list.php και η επικοινωνία από το contact.php.

Με την χρήση εξωτερικού CSS το οποίο βλέπουμε στην παρακάτω γραμμή κάνουμε το μενού να είναι πιο ωραίο εμφανισιακά και να βοηθάει τον χρήστη στην πλοήγηση όπως έχουμε αναφέρει.


```
<link rel="stylesheet" type="text/css" href="menub.css"/>
```

Βλέπουμε παρακάτω το κομμάτι κώδικα για την δημιουργία του μενού

```
19 <div id="menu-wrapper">
20   <ul id="hmenu">
21     <li><a href='http://my-thesis-anaskara.orgfree.com/index.php'>Αρχική σελίδα</a></li>
22     <li><a href='http://my-thesis-anaskara.orgfree.com/player.php'>Ακούστε μουσική</a></li>
23     <li><a href='http://my-thesis-anaskara.orgfree.com/mp3list.php'>Λίστα τραγουδιών</a></li>
24     <li><a href='http://my-thesis-anaskara.orgfree.com/contact.php'>Επικοινωνία</a></li>
25   </ul>
26 </div>
```

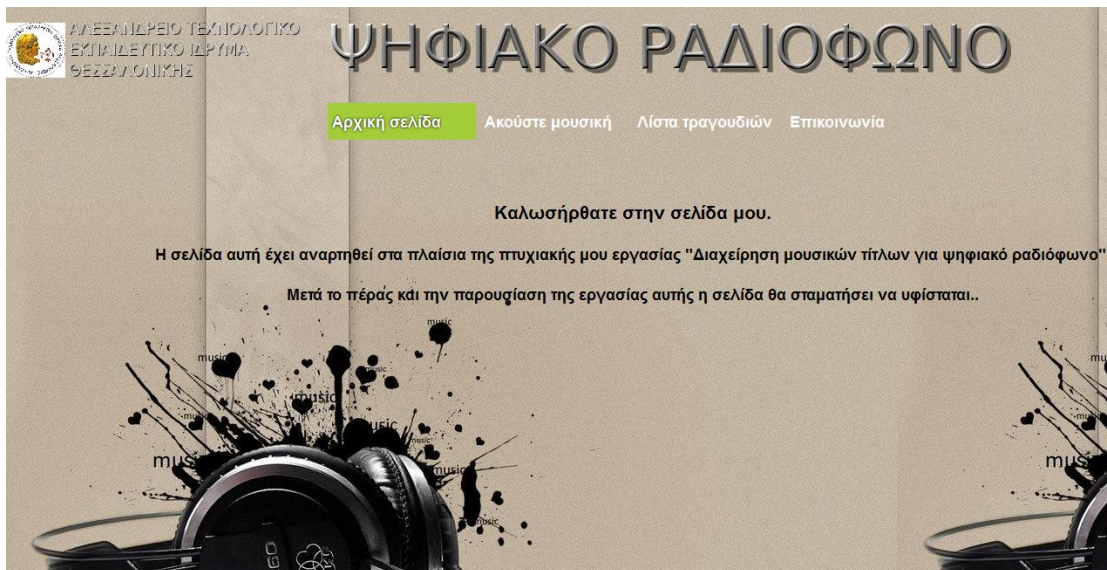
Η ετικέτα <div> ορίζει ένα ξεχωριστό τμήμα σε ένα έγγραφο, τοποθετούμε το menu σε ένα τέτοιο τμήμα και του δίνουμε μία ξεχωριστή ταυτότητα (*id="menu-wrapper"*) για να είναι δυνατή η τροποποίηση της εμφάνισής του όπως επιθυμούμε αργότερα με τη χρήση CSS. Κάθε υποκατηγορία αποτελεί μέρος μιας αδιάτακτης λίστας η οποία ορίζεται με την ετικέτα και το κάθε στοιχείο της με την . Και εδώ η ταυτότητα (*id="hmenu"*) χρησιμοποιείται για να επεξεργαστεί με την χρήση CSS. Σε κάθε περίπτωση λοιπόν το στοιχείο αποτελεί σύνδεσμο (*ετικέτα <a>*) στην ανάλογη σελίδα κατηγορίας. Στην συνέχεια θα δούμε το κομμάτι κώδικα CSS που χρησιμοποιήθηκε για την επεξεργασία του παραπάνω μενού.

```
12 #menu-wrapper ul{
13   display: block;
14   padding: 10px 5px;
15   width: 700px;
16
17 }
18 #hmenu > li{
19   display: inline-block;
20 }
21
22 #hmenu > li > a {
23   display: block;
24   padding: 10px 5px;
25   width: 160px;
26 }
27
28 li ul{
29   position: absolute;
30   display: none;
31 }
32
33 .sub-menu a{
34   display: block;
35   padding: 10px 5px;
36   width: 130px;
37 }
38 li:hover ul{
39   display: block;
40 }
41
42 #menu wrapper, li ul {
43   background: #333;
44 }
45
46 #hmenu{
47   width: 412px;
48   margin: 0 auto;
49 }
50
51 #hmenu a{
52   color: #fff;
53   text-decoration: none;
54   font-weight: bold;
55
56 }
57
58 #hmenu li:hover {
59   background: #a3cd39;
60   text-shadow: 0 0 2px #000;
61 }
```

Επειδή θέλουμε οριζόντιο menu, εφαρμόζουμε το στυλ “display: inline” για κάθε στοιχείο της λίστας. Στη συνέχεια, για να κάνουμε τα link να λειτουργούν σαν κουμπιά, έπρεπε να επεκτείνουμε το link από το κείμενο ώστε να καλύπτει όλο το πλαίσιο γύρω του. Αυτό το καταφέρνουμε με το στυλ “display: block” σε κάθε link του menu. Έπειτα εφαρμόζουμε τα στυλ εμφάνισης του menu όπως ύψος, χρώμα

φόντου, χρώμα και μέγεθος γραμματοσειράς. Με το «hover» όταν το ποντίκι βρίσκεται πάνω στις καρτέλες αλλάζει το χρώμα.

Στιγμιότυπο της αρχικής σελίδα βλέπουμε στην παρακάτω εικόνα.



4.2 Δημιουργία βάσης δεδομένων

Έπειτα από την χρήση της HTML και πριν προχωρήσουμε στην δημιουργία με τη χρήση των φορμών που χρειάζονται για να καταφέρουμε να κάνουμε όσα έχουμε αναλύσει στα προηγούμενα κεφάλαια θα πρέπει να δημιουργήσουμε μια βάση δεδομένων. Την βάση δεδομένων των τραγουδιών μας που θα χρησιμοποιήσουμε για τις ανάγκες της εφαρμογής.

Ο online server στον οποίο θα ανεβάσουμε τα αρχεία μας, όπως έχουμε αναφέρει στα πλεονεκτήματα του, έχει εγκατεστημένο το phpMyAdmin με τη βοήθεια αυτού και της γλώσσας MySQL θα δημιουργήσουμε τον πίνακα με τα στοιχεία των τραγουδιών και έπειτα με τα κατάλληλα ερωτήματα που θα αναφέρουμε παρακάτω θα επιλέγονται κάθε φορά τα εκάστοτε τραγούδια.

Στον Server μας λοιπόν είναι δημιουργημένη η βάση «439701» στην οποία απαιτείται κωδικός για να έχει κάποιος πρόσβαση, στην εφαρμογή μας την δυνατότητα αυτήν την έχει μόνο ο Administrator.

4.2.1 Δημιουργία πίνακα τραγουδιών

Μέσα στην βάση λοιπόν δημιουργούμε έναν πίνακα τον οποίο ονομάζουμε “mp3” ο οποίος έχει δώδεκα πεδία (**id**, **Artist**, **Title**, **Album**, **Language**, **Year**, **Genre**, **Composer**, **Lyricist**, **Mood**, **Time**, **Path**). Τα πεδία με τα έντονα γράμματα είναι αυτά που θα μας απασχολήσουν, και τα άλλα έχουν εισαχθεί για καθαρά εγκυκλοπαιδικούς λόγους. Αυτό που πρέπει να γνωρίζουμε για τη βάση μας είναι ότι το primary key είναι το id το οποίο εισάγεται με AUTO_INCREMENT δηλαδή κάθε φορά που θα κάνουμε κάποια εγγραφή αυτό θα παίρνει τιμή από μόνο του.

Όλα τα πεδία έχουν αναφερθεί ότι θα υπάρξουν, σε προηγούμενα κεφάλαια. Για το μόνο που δεν έχουμε αναφέρει είναι το path το οποίο αναφέρεται στο path του κάθε τραγουδιού, δηλαδή το που βρίσκεται το κάθε αρχείο τραγουδιού το οποίο θα ανακτήσει αργότερα ο player μας. Αυτό που μένει τώρα είναι να εισάγουμε τα στοιχεία των τραγουδιών στη βάση μας και να ανεβάσουμε τα τραγούδια αυτά στον server.

4.2.2 Εισαγωγή τραγουδιών

Όπως αναφέραμε λοιπόν στο πρώτο κεφάλαιο βάση των εντολών που πρέπει να ακολουθήσουμε από την ΑΕΠΙ τα τραγούδια που θα εισάγουμε στην βάση μας πρέπει να προέρχονται μετά από νόμιμη αγορά. Για τις ανάγκες λοιπόν της πτυχιακής εργασίας ανεβάσαμε στον server 46 τραγούδια ο οποίος αριθμός είναι ικανοποιητικός για να μπορέσουμε να δούμε και να κατανοήσουμε καλύτερα την λειτουργία της εφαρμογής μας.

Αφού ανεβάσουμε λοιπόν τα τραγούδια στον server τώρα καλούμαστε να εισάγουμε τα στοιχεία του κάθε τραγουδιού στην βάση μας. Αυτό γίνεται μέσω του rhpMyAdmin όπως έχουμε αναφέρει και κάθε φορά που γίνεται μια εγγραφή εκτελείται ο παρακάτω κώδικας.

```
1 INSERT INTO `439701`.`mp3` (  
2 `id` ,  
3 `Artist` ,  
4 `Title` ,  
5 `Album` ,  
6 `Language` ,  
7 `Year` ,  
8 `Genre` ,  
9 `Composer` ,  
10 `Lyricist` ,  
11 `Mood` ,  
12 `Time` ,  
13 `Path`  
14 )  
15 VALUES (  
16  
17 );
```

Μέσα στο VALUES υπάρχουν τα στοιχεία της εγγραφής μας κάθε φορά. Αφού εισάγουμε τα τραγούδια μας κατά τον παραπάνω τρόπο έχουμε δημιουργήσει πλέον την βάση των τραγουδιών μας και προχωράμε στην συνέχιση υλοποίησης της εφαρμογής μας.

4.3 Δημιουργία φόρμας επικοινωνίας

Αφού έχουμε κάνει λοιπόν τα παραπάνω αρχίζουμε σιγά σιγά την υλοποίηση των υπόλοιπων καρτελών. Πέρα της αρχικής σελίδας που αναλύθηκε θα αναφερθούμε αναλυτικά και στις υπόλοιπες τρεις καρτέλες. Πρώτα λοιπόν θα δημιουργήσουμε την καρτέλα Επικοινωνία.

Ένας από τους στόχους μας λοιπόν που έχει αναλυθεί στο προηγούμενο κεφάλαιο και τον διακρίναμε και σαν περίπτωση χρήσης είναι η επικοινωνία του χρήστη με τον διαχειριστή του συστήματος.

Αυτό βοηθάει όπως είπαμε τον διαχειριστή αλλά και τον εκάστοτε προγραμματιστή να λάβει υπόψιν του τα σχόλια των χρηστών και να αναπτύξει επαναληπτικά καλύτερα το «έργο» του.

Στην σελίδα μας θα υπάρχει μια φόρμα στην οποία ο χρήστης αφήνει το σχόλιο του στον διαχειριστή και αυτό θα του αποστέλλεται μέσω ηλεκτρονικού ταχυδρομείου.

Η μορφή της φαίνεται παρακάτω.

A screenshot of a web form on a light brown textured background. The form contains four input fields: a text box for 'Όνομα:', a text box for 'Επώνυμο:', a text box for 'eMail:', and a larger text area for 'Σχόλιο:'. At the bottom of the form is a button labeled 'Αποστολή'.

Η δημιουργία της φόρμας αυτής γίνεται με HTML και το βλέπουμε παρακάτω στο απόσπασμα κώδικα.

```
27 <div id="myform">
28 <form method="post" action="contact2.php">
29
30 <p><label for="name">Όνομα:</label>
31 <input type="text" size="29" name="name" /></p>
32
33 <p><label for="lastname">Επώνυμο:</label>
34 <input type="text" size="29" name="lastname" /></p>
35
36 <p><label for="email">eMail:</label>
37 <input type="text" size="29" name="email" /></p>
38
39 <label for="comments">Σχόλιο:</label>
40 <textarea name="comments" rows="10" cols="28"></textarea>
41
42 <p><input type="submit" name="submit" value="Αποστολή" /></p>
43
44 </form>
45 </div>
```

Στην πρώτη γραμμή στο div id="myform" μορφοποιείται η φόρμα με CSS ξανά όπως αναφέραμε και στην δημιουργία του μενού.

```
63 #myform {
64   position: relative;
65   background: url(backform11.jpg);
66   width: 250px;
67   margin: auto;
68   margin-top: 50px;
69   padding: 15px;
70   border: 1px solid lightgrey;
71   border-radius: 15px;
72 }
```

Χρησιμοποιείται κάποια εικόνα ως φόντο στην φόρμα αυτή και καθορίζονται οι διαστάσεις και η θέση της φόρμας καθώς και τα περιθώρια της.

Αφού έχουμε δημιουργήσει και μορφοποιήσει την φόρμα λοιπόν πρέπει τα στοιχεία και τα σχόλια να αποσταλούν στον διαχειριστή. Αυτό γίνεται με την γλώσσα και τον κώδικα τον βλέπουμε στο παρακάτω απόσπασμα.

```
27 <?php $to = "anaskara@it.teithe.gr";
28 $subject = "Contact Us";
29 $email = $_REQUEST['email'];
30 $message = $_REQUEST['comments'];
31 $headers = "From: $email";
32 $sent = mail($to, $subject,$message, $headers);
33 if($sent)
34 {
35
36   print "Your mail was sent successfully";
37
38 }
39
40 else {print "We encountered an error sending your mail"; }
41 ?>
```

Το σχόλιο λοιπόν που κάποιος γράφει το παίρνει η php variable \$message με το \$_REQUEST[comments] και αποστέλλεται μέσω ηλεκτρονικού ταχυδρομείου με την συνάρτηση mail όπως την διακρίνουμε και παραπάνω.

Καλό θα ήταν να αναφέρουμε ότι η παραπάνω υλοποίηση δεν μπορεί να επιτευχθεί σε local host γιατί δεν υπάρχει κάποιος mail host που να υποστηρίζει την αποστολή e-mail. Και εδώ διακρίνουμε ακόμα ένα πλεονέκτημα του online server που έχουμε χρησιμοποιήσει.

4.4 Δημιουργία καρτέλας Λίστα τραγουδιών

Και πλέον έχουμε φτάσει στην δημιουργία της προτελευταίας καρτέλας του μενού του ιστότοπου μας. Σε αυτή την καρτέλα λοιπόν δεν θα υπάρχει κάτι το ιδιαίτερο απλώς θα δίνεται η δυνατότητα στον χρήστη να ελέγξει τα τραγούδια που υπάρχουν και τον αριθμό τους. Να κάνει απλά μια περιήγηση για να δει στο περίπου με τι μουσικά ακούσματα έχει να κάνει και αν τον ικανοποιούν τα υπάρχοντα.

Αυτό λοιπόν υλοποιείται πολύ εύκολα με ένα απλό php κώδικα και ένα απλό ερώτημα SQL. Μπορούμε να δούμε παρακάτω την υλοποίηση αυτή.

```

27 <?php
28
29 //Connect and select a database
30 mysql_connect("localhost", "439701", "████████");
31 mysql_select_db("439701");
32
33 $query="SELECT * FROM mp3";
34 $result=mysql_query($query);
35
36 $num=mysql_numrows($result);
37
38 //emfanisi arithmou tragoudiwn
39 echo "<br>";
40 echo "<center><b>ΑΡΙΘΜΟΣ ΤΡΑΓΟΥΔΙΩΝ : $num</b></center>";
41 echo "<br>";
42 echo "<br>";
43
44 //Database output
45 echo "<center><b><blink>ΛΙΣΤΑ ΤΡΑΓΟΥΔΙΩΝ</blink></b><br><br></center>";
46 $i=0;
47 while ($i < $num) {
48
49     $Artist=mysql_result($result,$i,"Artist");
50     $Title=mysql_result($result,$i,"Title");
51
52     echo "<center><b>$Artist - $Title</b><br><center>";
53
54     $i++;
55 }
56
57 ?>

```

Στις γραμμές 30-31 γίνεται η επιλογή και η σύνδεση με την βάση δεδομένων. Το καλυμμένο πλαίσιο αφορά τον κωδικό της βάσης που όπως έχουμε αναφέρει γνώστης είναι μόνο ο διαχειριστής του συστήματος. Έπειτα στην γραμμή 33 έχουμε το SQL ερώτημα. Στην περίπτωση αυτή το ερώτημα αφορά όλες τις εγγραφές των τραγουδιών.

Στην γραμμή 36 η μεταβλητή \$num παίρνει τον αριθμό των εγγραφών και τον εμφανίζει με την γραμμή 40. Εμφανίζει δηλαδή στην περίπτωση μας το σύνολο των τραγουδιών που υπάρχουν στην βάση.

Έπειτα με τον βρόγχο while στις γραμμές 45 με 55 για το σύνολο των εγγραφών εμφανίζονται ο καλλιτέχνης και ο τίτλος του τραγουδιού. Βέβαια και με ακόμα μια εντολή echo αν επιθυμούσαμε θα μπορούσαμε να εμφανίσουμε περαιτέρω πληροφορίες.

Τέλος θα αναφερθούμε στις γραμμές 49-50 όπου βάσει των αποτελεσμάτων που θα βρεθούν η μεταβλητή \$Artist κάθε φορά την τιμή από τον πίνακα της βάσης και έπειτα με την echo τον εμφανίζει. Το ίδιο γίνεται και με τον τίτλο του εκάστοτε τραγουδιού.

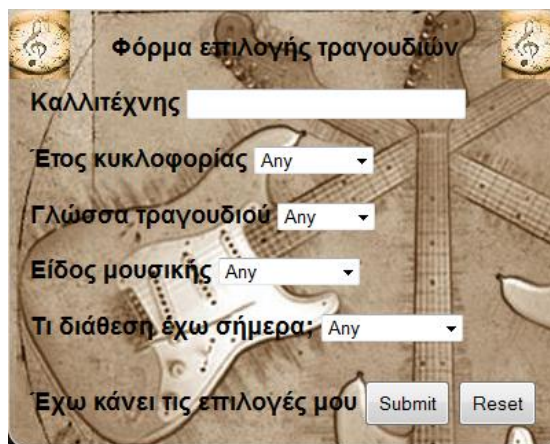
4.5 Δημιουργία καρτέλας του player

Αισίως λοιπόν φτάσαμε στην υλοποίηση της τέταρτης και τελευταίας καρτέλας του ιστοτόπου μας και ταυτόχρονα της πιο σημαντικής. Σε αυτήν την καρτέλα λοιπόν θα υλοποιηθούν τα περισσότερα και οι δυο σημαντικές περιπτώσεις χρήσης που έχουν αναλυθεί στο τρίτο κεφάλαιο.

Η σελίδα λοιπόν αυτή που όπως έχουμε πει είναι κυρίως το αρχείο player.php θα απαρτίζεται από μια φόρμα επιλογής τραγουδιών από μια φόρμα όπου θα δίνεται η δυνατότητα της επιλογής τραγουδιών ανά ενότητες των 30 λεπτών και ασφαλώς από τον mp3 player. Αυτά λοιπόν θα αναλυθούν μεμονωμένα παρακάτω.

4.5.1 Δημιουργία φόρμας επιλογής τραγουδιών

Θέλουμε να δημιουργήσουμε μια φόρμα λοιπόν στην οποία ο χρήστης να φτιάξει μια λίστα πολύ συγκεκριμένων απαιτήσεων. Στην φόρμα θα υπάρχει η δυνατότητα να επιλεχθούν συγκεκριμένος καλλιτέχνης, δεκαετία, γλώσσα τραγουδιού, είδος μουσικής κτλ και βέβαια αυτά τα φίλτρα μπορούν να συνδυαστούν. Η φόρμα αυτή λοιπόν έχει την παρακάτω μορφή.



Η υλοποίηση της φόρμας αυτής γίνεται με HTML και μπορούμε να το δούμε παρακάτω στο απόσπασμα του κώδικα που ακολουθεί


```
133 <div id= "left" >
134 <!--Forma epilogis tragoudiwn-->
135 <div id= "myform2" >
136 <form method= "post" >
137   <center>
138     <label for= "name" ><strong> Φόρμα επιλογής τραγουδιών <br />
139   </center>
140
141   <!--Epilogi me vasi ton kallitexni-->
142   <p><label for= "name" ><strong> Καλλιτέχνης </strong></ label>
143     <input type= "text" size= "29" name= "Artist" /></p>
144
145   <!--Epilogi me vasi tin xronologia ekdosis me list menu-->
146   <p>
147     <strong> Έτος κυκλοφορίας </strong>
148     <select name= "Year" id= "Year" >
149       <option value= "any1" >Any</option>
150       <option value= "five" >1951-1960</option>
151       <option value= "six" >1961-1970</option>
152       <option value= "seven" >1971-1980</option>
153       <option value= "eight" >1981-1990</option>
154       <option value= "nine" >1991-2000</option>
155       <option value= "twenty" >2001-2012</option>
156     </select>
157   </p>
```

Στην γραμμή 133 πάλι με την βοήθεια CSS καταφέρνουμε να κάνουμε divide στην σελίδα και το "left" μας δείχνει το position της φόρμας. Σε αντίστοιχη περίπτωση και με την φόρμα επικοινωνίας η μορφή της φόρμας καθορίζεται απο το "myform2" το οποίο έχει επεξεργαστεί με CSS. Η συνέχεια της υλοποίησης της φόρμας συνεχίζει παρακάτω.

```

158
159 <!--Epilogi me vasi tin glwssa tragoudiou me list menu-->
160 <p><strong> Γλώσσα τραγουδιού </strong>
161   <select name= "Language" id="Language" >
162     <option value= "any2" >Any</option>
163     <option value= "Greek" >Greek</option>
164     <option value= "English" >English</option>
165     <option value= "French" >French</option>
166   </select>
167 </p>
168
169 <!--Epilogi me vasi to eidos mousikis me list menu-->
170 <p><strong> Είδος μουσικής </strong>
171   <select name= "Genre" id="Genre" >
172     <option value= "any3" >Any</option>
173     <option value= "rock" >Rock</option>
174     <option value= "pop" >Pop</option>
175     <option value= "traditional" >Traditional </option>
176     <option value= "country" >Country </option>
177     <option value= "instrumental" >Instrumental </option>
178     <option value= "jazz" >Jazz</option>
179     <option value= "heavymetal" >Heavy metal</option>
180     <option value= "electronic" >Electronic </option>
181   </select>
182 </p>
183
184 <!--Mood agent-->
185 <p><strong> Τι διάθεση έχω σήμερα; </strong>
186   <select name= "Mood" id="Mood" >
187     <option value= "any5" >Any</option>
188     <option value= "happy" >I am happy :)</option>
189     <option value= "sad" >I am sad :( </option>
190   </select>
191 </p>
192
193 <p></p>
194 <p></p>
195
196 <!--Submit button gia na apostaloun oi ekastote epiloges-->
197 <strong> Έχω κάνει τις επιλογές μου </strong>
198 <input type= "submit" name="submit" value="Submit" />
199
200 <!--Reset button gia na katharistoun oi epiloges an xreiastei-->
201 <input type= "reset" name="Reset" id="Reset" value="Reset" />
202
203 </form>
204 </div>

```

Μετά την υλοποίηση της μορφής της φόρμας καλούμαστε να χρησιμοποιήσουμε τα στοιχεία της φόρμας που θα μας οδηγήσουν στην επιλογή των τραγουδιών. Και αυτό προφανώς θα πραγματοποιηθεί με την γλώσσα php.

Μετά το πάτημα του “submit” για κάθε ένα από τα πεδία (fields) πρέπει να δημιουργήσουμε μεταβλητές που θα τους ανατίθενται οι εκάστοτε τιμές. Αυτό γίνεται με το παρακάτω κομμάτι του κώδικα.

```

235 //paimai times sta eiserxomena dedomena
236 $art = $_POST['Artist' ];
237 $year=$_POST['Year'];
238 $lang=$_POST['Language' ];
239 $gen=$_POST['Genre' ];
240 $mood=$_POST['Mood' ];
241
242 //test gia na doume an parthikan swsta ta dedomena
243 /*echo $art;
244 echo $year;
245 echo $lang;
246 echo $gen;
247 echo $mood;*/

```

Όπως βλέπουμε και στα σχόλια στις γραμμές 242 με 247 μπορούμε να κάνουμε ένα τεστ για να δούμε αν όλα πήγαν καλά μέχρι τη συγκεκριμένη χρονική στιγμή.

Έπειτα καλούμαστε να αντιμετωπίσουμε το εξής «πρόβλημα»: Όταν δίνεται η τιμή “any” σε κάποιο από τα πεδία να επιλέγονται όλες οι τιμές του. Πχ στο πεδίο με το είδος μουσικής όταν επιλέγεται το “any” να αφορά όλα τα είδη μουσικής. Στο πεδίο του καλλιτέχνη δεν αντιμετωπίζουμε τέτοιο πρόβλημα καθώς όταν αυτό μείνει κενό επιλέγονται όλοι οι καλλιτέχνες.

Μετά από σχετική έρευνα στο διαδίκτυο και έπειτα από πολλές δοκιμές το παραπάνω πρόβλημα επιλύθηκε με την δημιουργία μιας ακόμα μεταβλητής που θα της ανατίθεται η τιμή του κενού (null) όταν επιλέγεται το any.

Έτσι με το ερώτημα της SQL έπειτα για το any θα επιλέγονται όλες οι τιμές. Αυτό υλοποιείται παρακάτω.

```

249 //dimiourgoume mia kainouria metavliti lang2 pou an exei epilexthei to any sto menu mas
250 //tha paimai tin timi tou kenou " " etsi wste meta sto sql query mas na emfanizei
251 //oles mas tis epiloges.Opoiadipote alli metavliti paimetai kanonika..
252 if ($lang!='any2'){
253     $lang2=$_POST['Language' ];
254 }
255 else{
256     $lang2="";
257 }
258 //To idio me parapanw. Dimiourgoume mia kainouria mettavliti gen2
259 if ($gen!='any3'){
260     $gen2=$_POST['Genre' ];
261 }
262 else {
263     $gen2="";
264 }
265
266 //To idio isxiei kai edw. Dimiourgoume mia kainouria mettavliti mood2
267 if ($mood!='any5'){
268     $mood2=$_POST['Mood' ];
269 }
270 else {
271     $mood2="";
272 }
273

```

Για την τιμή του Year επειδή έχουμε να κάνουμε με δεκαετίες έγινε με διαφορετικό τρόπο το οποίο φαίνεται στον παρακάτω και στα σχόλια του επεξηγείται ακριβώς τι έχει γίνει.

```

275 //Stin periptwsi tis xronologias epeidi exoume na kanoume me dekaeties tha prepei na tsekaroume
276 //me to between stin sql tin xronia ekdosis tou tragoudiou se poia dekaetia anikei.
277 //me ton vroxo switch loipon diakrinoume 7 periptwseis oses kai oi dekaeties.
278 //Me tin ekastote epilogi mas loipon dimiourgoume 2 times to decbegin kai to decend
279 //tin arxi diladi kai to telos tis dekaetias pou anamesa se ayta kaleitai na psaxnei
280 //kathe fora to query tis sql
281
282 switch ($year) {
283
284     case "any1" :
285         $decbegin =1951;
286         $decend =2012;
287         break;
288
289     case "five" :
290         $decbegin =1951;
291         $decend =1960;
292         break;
293
294     case "six" :
295         $decbegin =1961;
296         $decend =1970;
297         break;
298
299     case "seven" :
300         $decbegin =1971;
301         $decend =1980;
302         break;
303
304     case "eight" :
305         $decbegin =1981;
306         $decend =1990;
307         break;
308
309     case "nine" :
310         $decbegin =1991;
311         $decend =2000;
312         break;
313
314     case "twenty" :
315         $decbegin =2001;
316         $decend =2012;
317         break;
318 }

```

Αφού λοιπόν έχουμε πραγματοποιήσει τα παραπάνω σειρά παίρνει το ερώτημα SQL που θα εμφανίζει όλα τα επιλεγμένα τραγούδια βάσει των φίλτρων από την βάση δεδομένων μας, το οποίο φαίνεται παρακάτω.

```

324 //to query stin sql me ola ta pedia tis formas kai na emfanizontai kathe fora tixaia...
325 $query="SELECT * FROM mp3 where Artist like '%$art%' AND Language like '%$lang2%' AND Genre like '%$gen2%' AND Mood like '
    %$mood2%' AND Year between $decbegin and $decend ORDER BY RAND()" ;
326
327 $result=mysql_query ($query) or die ("Data not Found" );
328 $num=mysql_numrows ($result);

```

«Μεταφράζοντας» το παραπάνω κώδικα για κάποιον που ίσως δεν είναι εξοικιωμένος με την SQL, και για να μας δώσει μια εικόνα για το πως λειτουργεί έχουμε: «Επέλεξε από τον πίνακα Mp3 όλα εκείνα τα τραγούδια στα οποία

ισχύουν όλες οι προϋποθέσεις (μετά το AND ακολουθεί καινούρια προϋπόθεση κάθε φορά)».

Το "ORDER BY RAND()" έχει τοποθετηθεί για να εμφανίζει κάθε φορά τα τραγούδια με τυχαίο τρόπο.

Με αντίστοιχο τρόπο με την εμφάνιση όλων των τραγουδιών όπου αναλύθηκε πιο πάνω, με τον βρόγχο while δηλαδή, εμφανίζονται τα επιλεγμένα τραγούδια. Αν βρεθούν τραγούδια με τις επιλογές που έβαλε ο χρήστης εμφανίζονται τα τραγούδια. Αν όμως δεν βρεθούν τότε εμφανίζεται ένα μήνυμα που προτρέπει τον χρήστη να ξαναεπιλέξει. Αυτό υλοποιείται με το παρακάτω κομμάτι.

```
332  if ($num !=0){
        ...
        ...
        ...
362      //else statement gia $num=0 diladi otan den vrethoun apotelesmata me ta
363      //sigkekrimena kritiria [SONGS NOT FOUND]
364      else {
365          //echo "<p style='text-align:right;'><b>NO SONGS FOUND ...</b></p>";
366          //echo "<p style='text-align:right;'><b>Make your choices again !!!</b></p>";
367          echo "<b>NO SONGS FOUND</b>" ;
368          echo "<br>" ;
369          echo "<b>Make your choices again !!!<b>" ;
370      }
```

Το πως θα ανακτηθούν για να αναπαράγονται από το μέσο αναπαραγωγής που χρησιμοποιούμε θα αναλυθεί παρακάτω.

4.5.2 Δημιουργία φόρμας επιλογής τραγουδιών ανά ενότητες

Μετά την υλοποίηση της παραπάνω φόρμας για την επιλογή μεμονωμένων τραγουδιών, σε αυτό το section θα αναλύσουμε το πως θα υλοποιηθεί η φόρμα για την επιλογή τραγουδιών ανά ενότητες.

Δηλαδή την δημιουργία ενός τρόπου που θα πραγματοποιείται η λειτουργία του player σαν ραδιόφωνο.

Καταρχήν θα εμφανίζεται η φόρμα της οποίας την μορφή βλέπουμε παρακάτω.



Τα τέσσερα είδη μουσικής τα οποία τα βλέπουμε με την μορφή κουμπιών είναι αυτά που υπάρχουν μέσα στη βάση. Αν αργότερα θελήσουμε να εισάγουμε κάποιο άλλο είδος θα αλλάξουμε και την παραπάνω φόρμα.

Η μορφή της πραγματοποιείται με τρόπο ανάλογο της δημιουργίας της προηγούμενης φόρμας.

Τώρα όσον αφορά την λειτουργία της, πρέπει να την υλοποιήσουμε με τέτοιο τρόπο έτσι ώστε να πραγματοποιείται μια λειτουργία ραδιοφώνου. Δηλαδή με το πάτημα κάποιου κουμπιού (πχ ροκ) θα επιλέγονται από την βάση τραγούδια ροκ συνολικής διάρκειας τριάντα λεπτών. Έπειτα από αυτά τα λεπτά στην λίστα που θα ανακτά ο player θα εισάγονται άλλα τριάντα λεπτά μουσικής από την επόμενη ενότητα κατά τον τρόπο που είναι διατεταγμένες στην φόρμα μας. Δηλαδή αν πατήσουμε πρώτα ροκ θα ακουστούν τριάντα λεπτά ροκ, έπειτα τριάντα λεπτά ποπ, κοκ. Αν επιλεγεί πρώτα η ενότητα ποπ θα ακουστούν πρώτα τα τριάντα λεπτά ποπ και τελευταία η ενότητα ροκ.

Με τον παραπάνω τρόπο δημιουργείται ένα πρόγραμμα ραδιοφώνου από τέσσερις μουσικές ενότητες συνολικής διάρκειας δύο ωρών.

Τώρα θα αναλύσουμε τον τρόπο με τον οποίο υλοποιήθηκε το πρσπύσνω εμφανίζοντας τα πιο σημαντικά κομμάτια του κώδικα rhr που χρησιμοποιήθηκε.

Για κάθε επιλογή λοιπόν όπως βλέπουμε και παρακάτω καλείται κάποιο συγκεκριμένο αρχείο rhr.

```
109 <?php
110 if (isset($_POST['rock2'])) {
111     include ('selection1.php');
112 }
113 if (isset($_POST['pop2'])) {
114     include ('selectionpop.php');
115 }
116 if (isset($_POST['electronic2'])) {
117     include ('selectionele.php');
118 }
119 if (isset($_POST['instrumental2'])) {
120     include ('selectionins.php');
121 }
122 }
123 }
124 }
125 }
126 ?>
```

Το κάθε ένα από τα αρχεία rhr που βρίσκονται μέσα στις include περιλαμβάνει κώδικα rhr όπου τα τραγούδια επιλέγονται με τον ακόλουθο τρόπο :

Καταρχήν υπάρχουν τέσσερα ερωτήματα SQL για κάθε είδος μουσικής.

```
31 $query="SELECT * FROM mp3 WHERE Genre like 'Rock' ORDER BY RAND()";
32 $result=mysql_query($query);
33 $num=mysql_numrows($result);
34
35 $query2="SELECT * FROM mp3 WHERE Genre like 'Pop' ORDER BY RAND()";
36 $result2=mysql_query($query2);
37 $num2=mysql_numrows($result2);
38
39 $query3="SELECT * FROM mp3 WHERE Genre like 'Electronic' ORDER BY RAND()";
40 $result3=mysql_query($query3);
41 $num3=mysql_numrows($result3);
42
43 $query4="SELECT * FROM mp3 WHERE Genre like 'Instrumental' ORDER BY RAND()";
44 $result4=mysql_query($query4);
45 $num4=mysql_numrows($result4);
```

Σε κάθε αρχείο η σειρά που εισάγονται τα τραγούδια στον player αλλάζει. Έστω ότι επιλέξαμε ροκ. Πρώτα θα επιλεγθούν τα ροκ μετά τα ποπ μετά τα electronic και μετά τα instrumental.

Στον βρόγχο while που θα χρησιμοποιήσουμε όμως τώρα θα πρέπει να βάλουμε έναν περιορισμό έτσι ώστε η συνολική διάρκεια των τραγουδιών που επιλέγονται να είναι περίπου τριάντα λεπτά. Αυτό μπορούμε να το δούμε παρακάτω.

```
56 $total_time=0;
57 $i=0;
58 while ($i < $num) {
59     if ($total_time < 1800){
60         ...
61         ...
62         ...
63
64
65
66
67
68
69
70
71
72
73
74
75     $total_time=$total_time + $Time2;
76
77     $i++;
78     }
79     else {
80         break;
81     }
```

Ο αριθμός 1800 αντιπροσωπεύει τα τριάντα λεπτά σε δευτερόλεπτα.

Στην περίπτωση χρήσης της φόρμας αυτής θα εμφανίζονται απο κάτω ακριβώς τα τραγούδια που έχουν επιλεγθεί από κάθε ενότητα και η συνολική διάρκεια τους. Εδώ ένα μικρό προβληματάκι αντιμετωπίσαμε με τις μεταβλητές των διάφορων χρόνων και διάρκειας αλλά επιλύθηκε πολύ εύκολα με τον παρακάτω τρόπο.

```

85     $total_min=$total_time/60;
86     $total_min2=intval($total_min);
87     $total_sec=$total_time%60;
88     if($total_sec<10){
89         $total_sec2 = "0$total_sec"; // add the zero
90     }
91     else {
92         $total_sec2 = "$total_sec"; // don't add the zero
93     }

```

Τέλος μετά την επιλογή των τραγουδιών θα ανακτηθούν και αυτά από τον player με τρόπο που θα δούμε σε παρακάτω παράγραφο.

4.5.3 Εισαγωγή και λειτουργία player

Μετά τον σχεδιασμό και την εισαγωγή των φορμών στην καρτέλα, σε αυτήν την παράγραφο θα δούμε τα πάντα που αφορούν το μέσο αναπαραγωγής μας, τον player δηλαδή. Όπως αναφέραμε και στο πρώτο κεφάλαιο θα χρησιμοποιηθεί ένας open source player όπου επιλέχθηκε διακρίνοντας πολλά πλεονεκτήματα σε σχέση με άλλους που υπάρχουν στο διαδίκτυο έπειτα από σχετική έρευνα και δοκιμή πολλών από αυτών.

Το πρώτο πράγμα που έχουμε να κάνουμε λοιπόν είναι να τον εισάγουμε στην καρτέλα μας.

4.5.3.1 Εισαγωγή του player στην σελίδα

Ο player μας λοιπόν αποτελείται από δύο αρχεία το ένα που αφορά την λειτουργία του και είναι γραμμένο σε javascript (script2.js) και το άλλο αφορά την μορφή του και είναι δημιουργημένο σε flash player (OriginalMusicPlayerPlaylist.swf).

Αφού ανεβάσουμε αυτά τα δύο αρχεία στον server μας εισάγοντας τον παρακάτω κώδικα ο player θα εμφανίζεται στην σελίδα μας.

```

44 <!--To script tou player-->
45 <script type="text/javascript" src="http://my-thesis-anaskara.orgfree.com/script2.js"></script>
46 <div id="player-holder"></div>
47
48 <script type="text/javascript">
49     var options = {};
50     options.playlistXmlPath = "http://my-thesis-anaskara.orgfree.com/playlist3.xml";
51     options.autoPlay = "true";
52
53     var params = {};
54     params.allowScriptAccess = "always";
55
56     swfobject.embedSWF("http://my-thesis-anaskara.orgfree.com/OriginalMusicPlayerPlaylist.swf",
57         "player-holder", "250", "250", "9.0.0",false, options, params, {});
58 </script>
59

```


Στην γραμμή 45 βλέπουμε ότι καλείται το script του player που αναφέραμε παραπάνω και στην γραμμή 56 το flash αρχείο.

Στην γραμμή 51 κάθε φορά που θα ανακτώνται τα επιλεγμένα τραγούδια θα αναπαράγονται αυτόματα.

Η ανάκτηση των τραγουδιών γίνεται στην γραμμή 50 την οποία ανάκτηση θα αναφέρουμε στην επόμενη παράγραφο.

4.5.3.2 Ανάκτηση τραγουδιών

Ο player λοιπόν ανακτά τα τραγούδια για να τα αναπαράγει μέσω μιας δυναμικής λίστας XML το οποίο βλέπουμε να γίνεται στην γραμμή 50 όπου το αρχείο XML είναι το playlist3.xml. Για να μπορέσει να ανακτήσει τα τραγούδια ο player μέσω της λίστας, η λίστα αυτή πρέπει να είναι δημιουργημένη κατά τον ακόλουθο τρόπο.

```
1 <?xml version='1.0' encoding='utf-8'?>
2
3 <playlist>
4   <item>
5     <title>Don't panic</title>
6     <artist>Coldplay</artist>
7     <path>http://my-thesis-anaskara.orgfree.com/songs2/COLDPLAY - Don't panic.mp3</path>
8   </item>
9
10  <item>
11    <title>Paei tosos kairos</title>
12    <artist>Kitrina podilata</artist>
13    <path>http://my-thesis-anaskara.orgfree.com/Paei tosos kairos.mp3</path>
14  </item>
15
16  <item>
17    <title>Twra siwpi</title>
18    <artist>Kitrina podilata</artist>
19    <path>http://my-thesis-anaskara.orgfree.com/songs2/tora siopi.mp3</path>
20  </item>
21
22 </playlist>
```

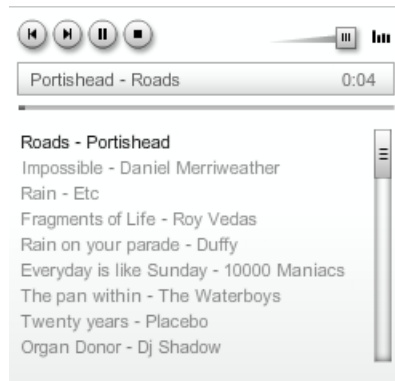
Με οποιαδήποτε άλλη μορφή ο player δεν μπορεί να “φορτώσει” τα τραγούδια.

Για το πως δημιουργείται η λίστα κάθε φορά θα αναφερθούμε στην παράγραφο 4.5.4

4.5.3.3 Λειτουργία player

Εφόσον λοιπόν τα τραγούδια επιλεγθούν, εισαχθούν στην λίστα, και ανακτηθούν σωστά ο player είναι έτοιμος να τα αναπαράγει. Αυτόματα όπως αναφέραμε λίγο παραπάνω.

Ένα στιγμιότυπο του player λοιπόν έχει την παρακάτω μορφή.



Η κύρια λειτουργία λοιπόν του player είναι να αναπαράγει τα τραγούδια αυτόματα. Από κει και πέρα συμπεριφέρεται όπως όλα τα μέσα αναπαραγωγής. Το μενού δηλαδή όπου φαίνονται όλα τα τραγούδια της λίστας. Με έντονα γράμματα επισημαίνεται το τραγούδι που αναπαράγεται κάθε φορά.

Επίσης υπάρχουν κουμπιά έναρξη/σταμάτημα και προηγούμενο/επόμενο τραγούδι, ρύθμιση έντασης, ένα μικρό equalizer, η διάρκεια του τραγουδιού και μπάρα όπου φαίνεται η πρόοδος του κάθε τραγουδιού.

Ανα πάσα στιγμή ο χρήστης, μπορεί να επιλέξει όποιο τραγούδι θέλει μέσα από την λίστα, να επαναλάβει κάποιο και να διακόψει φυσικά εντελώς την αναπαραγωγή.

Οι παραπάνω λειτουργίες μπορούν να γίνουν και από το πληκτρολόγιο του χρήστη. Πχ το πλήκτρο space λειτουργεί σαν play/pause.

Συνοψίζοντας αυτή ήταν και η λειτουργία του μέσου μας λοιπόν και στη συνέχεια θα δούμε πως δημιουργείται η εκάστοτε λίστα των τραγουδιών.

4.5.4 Δημιουργία λίστας αναπαραγωγής τραγουδιών

Μετά λοιπόν την επιλογή των τραγουδιών είτε μέσω της φόρμας ενοτήτων είτε μέσω της φόρμας επιλογής μεμονομένων τραγουδιών που αναλύθηκαν σε προηγούμενες παραγράφους, τώρα σε αυτή την παράγραφο θα αναλυθεί το πως θα δημιουργηθεί η δυναμική λίστα XML για να μπορέσει ο player να ανακτήσει τα τραγούδια.

Το αρχείο λοιπόν που “διαβάζει” ο player είναι το playlist3.xml. Οι πληροφορίες των τραγουδιών πρέπει να εισαχθούν με σωστό τρόπο όπως έχει προαναφερθεί στην παράγραφο 4.5.3.2.

Καταρχήν αυτή η λίστα δεν είναι σταθερή δηλαδή κάθε φορά παίρνει διαφορετικές. Τρέχοντας την εφαρμογή σε local host δεν αντιμετωπίσαμε κάποιο πρόβλημα γιατί το αρχείο αυτό βρίσκεται σε φάκελο του υπολογιστή μας.

Στην υλοποίηση της όμως στον online server μας θα μπορεί μόνο ο διαχειριστής του συστήματος να «πειράξει» την λίστα.

Γι αυτό το πρώτο πράγμα που πρέπει να κάνουμε είναι να αλλάξουμε τα δικαιώματα του αρχείου επιλέγοντας rwx για όλους τους χρήστες με την εντολή CHMOD 777 για να κάνουμε apply όλα τα δικαιώματα. Αφού κάνουμε λοιπόν αυτό είμαστε έτοιμοι να προχωρήσουμε.

Επειδή στην εφαρμογή μας δεν θέλουμε να μένουν αποθηκευμένες προτιμήσεις η προηγούμενη λίστα αναπαραγωγής, ο player μας θα ξεκινάει χωρίς να έχουν εισαχθεί τραγούδια και θα μας δείχνει null.

Αυτό πραγματοποιείται με ένα κομμάτι κώδικα php που τρέχει μόλις ανοίξει η σελίδα και σκοπός του είναι να δημιουργήσει ένα “κενό” αρχείο xml, και το βλέπουμε παρακάτω.

```

28 <?php
29 $xmlfile = "playlist3.xml";
30 $file = fopen($xmlfile,"w");
31
32 //dimiourgeitai to playlist3.xml
33 //eisagoume stin arxi ton typo xml version='1.0' encoding='utf-8'
34 fwrite($file, "<?xml version='1.0' encoding='utf-8'?>");
35
36 //grafoume to root sto xml arxeio mas
37 //ta ypoloipa dedomena kai ta "children" tou root
38 //tha eisaxthoun meta to submit
39 fwrite($file, "<playlist>");
40 ?>

```

Στη συνέχεια αφού επιλεγθούν τα φίλτρα μέσω των φορμών θα γίνει μία ολοκληρωμένη εισαγωγή στοιχείων και τα τραγούδια θα ανακτηθούν από τον player. Για παράδειγμα στην φόρμα επιλογής τραγουδιών μετά το πάτημα του submit ο κώδικας ξεκινάει ως εξής :

```

217 if (isset($_POST['submit'])) {
218
219 //dimiourgoume ena xml file sto opoio tha eisagoume kathw fora
220 //ta tragoudia pou epilegontai
221 $xmlfile = "playlist3.xml";
222 $file = fopen($xmlfile,"w");
223
224 //dimiourgeitai to playlist3.xml
225 //eisagoume stin arxi ton typo xml version='1.0' encoding='utf-8'
226 fwrite($file, "<?xml version='1.0' encoding='utf-8'?>");
227
228 //grafoume to root sto xml arxeio mas
229 //ta ypoloipa dedomena kai ta "children" tou root
230 //tha eisaxthoun meta to submit
231 fwrite($file, "<playlist>");

```

Δηλαδή ξανά δημιουργείται το αρχείο playlist3.xml και είναι έτοιμο να πάρει δεδομένα. Μπορούμε να διακρίνουμε μέχρι στιγμής στον κώδικα την δημιουργία του αρχείου αυτού στην γραμμή 221 και το άνοιγμα για εγγραφή στην 222. Έπειτα στην γραμμή 226 γράφουμε το xml version που πρέπει να υπάρχει για να λειτουργήσει σωστά και στην γραμμή 231 το root <playlist>.

Τα υπόλοιπα δεδομένα θα εισαχθούν κατά την εκτέλεση του επερωτήματος της SQL και του βρόγχου while που έχει προ αναφερθεί για την εμφάνιση των αποτελεσμάτων. Αυτό θα το δούμε στο παρακάτω απόσπασμα κώδικα όπου τα εκτενή σχόλια επεξηγούν ακριβώς πως λειτουργεί η όλη διαδικασία.

```

332 $i=0;
333 if ($num!=0){
334 while ($i < $num) {
335
336     $Artist=mysql_result($result,$i,"Artist");
337     $Title=mysql_result($result,$i,"Title");
338     $downloadlink=mysql_result($result, $i, "Path"); //to path tou kathe mp3
339
340     //emfanizetai gia kathe apotelesma o titlos tragoudiou
341     //kai o kallitexnis
342     //echo "<b>$Artist - $Title</b><br>";
343
344     //meta tin evresi twn apotelesmatwn gia kathe apotelesms me tin fwrite
345     //grafoume sto temp file playlist3 ta akoloutha apo to opoio file
346     //diavasei to script tou player gia na eisagei ta tragoudia
347
348     fwrite($file, "<item>"); //child of root
349
350     //children of the item
351     fwrite($file, "<title>" . $Title . "</title>");
352     fwrite($file, "<artist>" . $Artist . "</artist>");
353     fwrite($file, "<path>" . $downloadlink . "</path>");
354     fwrite($file, "</item>");
355
356
357     $i++;

```

Τα παραπάνω λοιπόν λαμβάνουν χώρα μέσα στο loop, και πλέον μετά από αυτό το αρχείο xml έχει αποκτήσει σχεδόν μια ολοκληρωμένη μορφή. Θα ολοκληρωθεί έξω από το loop με το κλείσιμο των tags της root, του <playlist> δηλαδή, και το κλείσιμο του αρχείου.

```

375 //to telos tou xml arxeiou
376 fwrite($file, "</playlist>");
377 fclose($file);
378

```

Με ανάλογη διαδικασία εισάγονται και τα τραγούδια από την επιλογή μέσω της φόρμας επιλογής τραγουδιών ανά ενότητες. Μετά την ολοκλήρωση λοιπόν της παραπάνω διαδικασίας, ο player ανακτά και αναπαράγει τα τραγούδια που έχουν επιλεγθεί.

4.6 Εισαγωγή widgets στην σελίδα

Μετά την ολοκλήρωση λοιπόν της εφαρμογής και της ιστοσελίδας που αναρτήθηκε καλό θα ήταν να ομορφύνουμε το περιβάλλον που αλληλεπιδρά ο χρήστης εισάγοντας και δημιουργώντας κάποιων ειδών widgets που θα κάνουν την παραμονή του χρήστη στη σελίδα πιο ευχάριστη. Τα widgets είναι μικρά κομμάτια έτοιμοι κώδικα που μπορούν να ενσωματωθούν σε οποιοδήποτε σημείο της

σελίδας και να προσθέσουν λειτουργικότητα σε αυτήν. Τα συναντάμε κυρίως στα blogs και υπάρχουν χιλιάδες από αυτά σε όλο το διαδίκτυο.

Στην σελίδα μας θα επιλέξω να βάλω ένα γράφημα που θα δείχνει τις προτιμήσεις των χρηστών με στατιστικά στοιχεία.

4.6.1 Εισαγωγή στατιστικών προτιμήσεων στην σελίδα

Στην φόρμα όπου επιλέγονται οι μουσικές κατηγορίες ωραίο θα ήταν να κρατάμε τα στατιστικά για να κάνουμε ένα είδους γκάλλοπ για το τι προτιμάνε οι χρήστες να ακούνε πρώτα.

Καταρχήν θα πρέπει να κρατάμε τις τιμές αποθηκευμένες για κάθε κλικ που γίνεται στην κάθε ενότητα.

Για κάθε κλικ λοιπόν που γίνεται οι τιμές αποθηκεύονται σε αρχεία κειμένου, για να μπορεί ο αριθμός αυτός να ανακτηθεί αργότερα και αυτό πραγματοποιείται με το παρακάτω απόσπασμα κώδικα.

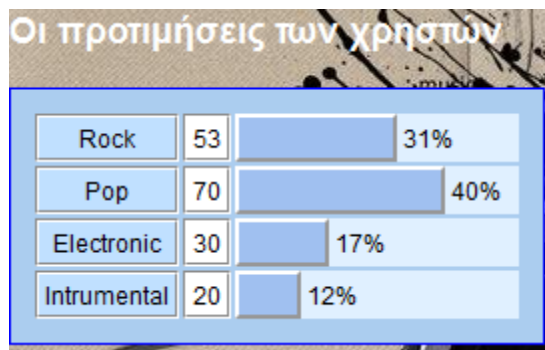
```
1  <?php
2
3  //ανοιγοyme to arxeio
4  $fp = fopen("rockclicks.txt", "r");
5
6  // Pairoyme ton arithmo pouy proyparxei
7  $count = fread($fp, 1024);
8
9  // Kleinoyme to arxeio
10 fclose($fp);
11
12 // Prosthetoume 1 ston arithmo
13 $count = $count + 1;
14
15 // Anoigoume pali to arxeio kai svinoume
16 //ta yparxonta dedomena
17 $fp = fopen("rockclicks.txt", "w");
18
19 // Grafoume ton kainourio metrii sto arxeio
20 fwrite($fp, $count);
21
22 // Kleinoume to arxeio
23 fclose($fp);
24
```

Μετά λοιπόν από αυτήν την διαδικασία θα εισάγουμε ένα open source widget θα δημιουργήσει ένα γράφημα στατιστικών. Τις τιμές θα τις πάρει απο τα text file που έχουν δημιουργηθεί και απόσπασμα του κώδικα μπορούμε να δούμε παρακάτω.

```
61 <?php
62 require once('graphs.inc.php');
63 $popc = file_get_contents('popclicks.txt');
64 $rockc = file_get_contents('rockclicks.txt');
65 $elec = file_get_contents('eleclicks.txt');
66 $insc = file_get_contents('insclicks.txt');
67 //echo $content;
68 $graph = new BAR_GRAPH("hBar");
69 $graph->values = "$rockc,$popc,$elec,$insc";
70 $graph->labels = "Rock,Pop,Electronic,Instrumental";
71 $graph->showValues = 1;
72 $graph->barWidth = 20;
73 $graph->barLength = 1.0;
74 $graph->labelSize = 12;
75 $graph->absValuesSize = 12;
76 $graph->percValuesSize = 12;
77 $graph->graphPadding = 10;
78 $graph->graphBGColor = "#ABCDEF";
```

Είναι ένα μικρό απόσπασμα του κώδικα λοιπόν όπου διακρίνουμε στις γραμμές 63 με 64 τις μεταβλητές που παίρνουν τις τιμές του από τα αρχεία στα οποία αναφέρονται.

Η όψη λοιπόν του γραφήματος αυτός απεικονίζεται παρακάτω.

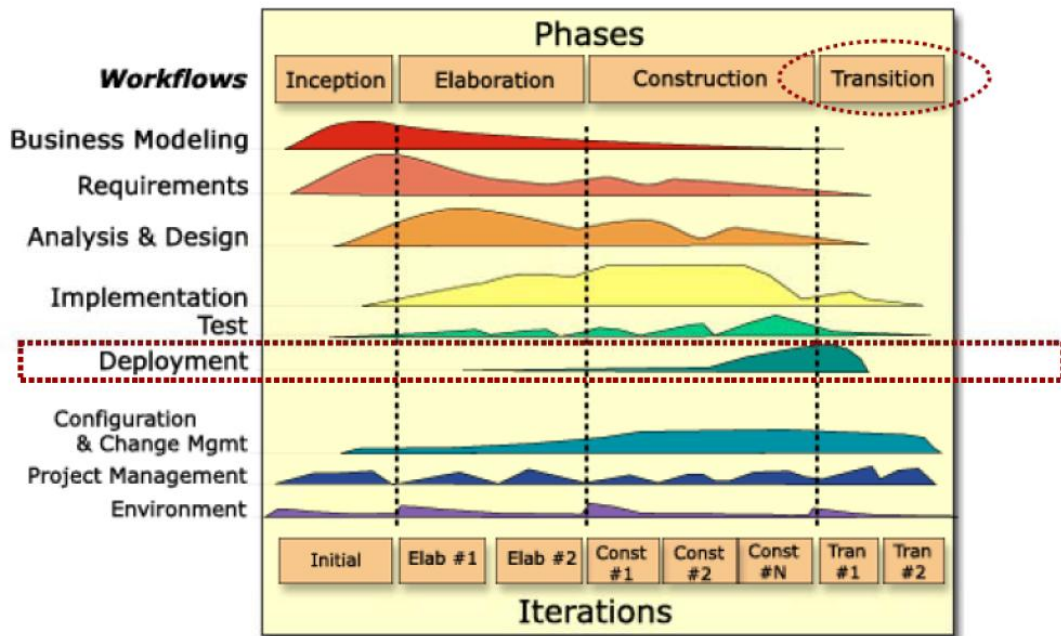


4.6.2 Δυνατότητα εισαγωγής περαιτέρων widgets

Το παραπάνω λοιπόν ήταν ένα δείγμα του πως μπορούμε να ομορφύνουμε το περιβάλλον του χρήστη. Όπως αναφέραμε πέρα από την υλοποίηση της εφαρμογής που αφορά την πτυχιακή εργασία, η εφαρμογή μπορεί να επεκταθεί χρησιμοποιώντας πάρα πολλά τέτοια widgets όπως η εμφάνιση των συνολικών επισκέψεων στην σελίδα, εμφάνιση των περιοχών των χρηστών, από ποια χώρα προέρχονται και πάρα πολλά άλλα που στόχους τους είναι η προσθήκη αισθητικής και λειτουργικότητας στο περιβάλλον του χρήστη.

4.7 Ανάπτυξη

Με βάση την μεθοδολογία που χρησιμοποιούμε έχουμε φτάσει πια στο τελικό στάδιο στην φάση μετάβασης δηλαδή στο περιβάλλον του και είναι ευδιάκριτο στην παρακάτω εικόνα.



Σχήμα 12. Στιγμιότυπο πορείας της μεθοδολογίας RUP κατά την φάση της μετάβασης

Στόχος είναι να παραδώσουμε την εφαρμογή μας στους τελικούς χρήστες και να είμαστε έτοιμοι να καλύψουμε μεγάλο εύρος χρήσεων όπως παροχή βοήθειας και υποστήριξης στους χρήστες.

4.8 Επίλογος

Τέλος λοιπόν του τέταρτου και τελευταίου κεφαλαίου που αφορούσε την υλοποίηση της εφαρμογής / ιστοσελίδας. Αναλύθηκαν και εμφανίστηκαν πολλά κομμάτια κώδικα και αναπτύχθηκαν εκτενέστερα, από τα πιο απλά μέχρι τα πιο σύνθετα πράγματα όπου πραγματοποιήθηκαν με τέτοιο τρόπο ώστε να κατατοπίσουν πλήρως τον αναγνώστη.

ΕΠΙΛΟΓΟΣ – ΣΥΜΠΕΡΑΣΜΑΤΑ

Τελειώνοντας λοιπόν την σύνταξη του κειμένου της πτυχιακής μου εργασίας, διαπιστώνω την πολύ καλή ευκαιρία που μου δόθηκε μέσω αυτής να αναπτύξω όλα εκείνα που είχα αποκομίσει τόσα χρόνια αλλά και να μάθω πολλά πράγματα που αφορούσαν τον προγραμματισμό και τον σχεδιασμό μιας ιστοσελίδας πράγμα που δεν είχα ξαναπράξει στο παρελθόν.

Μέσω του συνδυασμού της προγραμματιστικής άποψης με την ψυχαγωγική, αφού όλα έγιναν με την «συνοδεία» μουσικής, όλο αυτό δεν το αντιμετώπισα σαν την προσπάθεια διεκπεραίωσης ακόμα μιας κοινότυπης εργασίας αλλά σαν μια ευκαιρία που μου δόθηκε να διεκπεραιώσω κάτι πολύ ωραίο στο σύνολο και με προοπτικές για την συνέχεια.

Το τέλος της πτυχιακής μου εργασίας λοιπόν αποτέλεσε μια πολύ ωραία εμπειρία με όλα τα σημαντικά στοιχεία που αποκόμισα και θα με βοηθήσουν για την μετέπειτα πορεία μου.

ΠΗΓΕΣ

Στη συνέχεια θα αναφερθούν όλες εκείνες οι ιστοσελίδες από τις οποίες άντλησα τις περισσότερες πληροφορίες για την υλοποίηση της πτυχιακής μου εργασίας.

<http://infolific.com/technology/methodologies/rational-unified-process/>

http://en.wikipedia.org/wiki/IBM_Rational_Unified_Process

<http://searchsoftwarequality.techtarget.com/definition/Rational-Unified-Process>

http://era.nih.gov/docs/rup_fundamentals.htm

<http://blog.flexcommunity.net/?p=15>

<http://www.youtube.com/watch?v=wSRkTR5-8QA>

<http://blog.flexcommunity.net/?p=9>

<http://www.astahost.com/info/tftls-advice-creating-online-music-database-dont-databases.html>

<http://www.tizag.com/phpT/syntax.php>

<http://bytes.com/topic/php/insights/740327-uploading-files-into-mysql-database-using-php>

<http://bogdan.org.ua/2008/12/25/how-to-fix-mysql-server-has-gone-away-error-2006.html>

<http://drupal.org/node/76156>

<http://stackoverflow.com/questions/3596593/retrieve-mp3-from-mysql-and-force-download-instead-of-playback-in-browser>

<http://www.kidslovepc.com/php-tutorial/php-insert-blob-mysql.shtml>

<http://www.php-mysql-tutorial.com/wikis/mysql-tutorials/uploading-files-to-mysql-database.aspx>

<http://net.tutsplus.com/tutorials/php/object-oriented-php-for-beginners/>

<http://msdn.microsoft.com/en-us/library/ms345151%28v=sql.90%29.aspx>

<http://httpd.apache.org/docs/2.2/>

http://en.wikipedia.org/wiki/Apache_HTTP_Server

<http://www.lestarte.com/pages/apache-tutorial-e.html>

<http://www.web-resources.eu/archives/apache-virtualhost-greek-video-tutorial>

<http://www.mysqltutorial.org/>

<http://www.devshed.com/c/a/MySQL/Beginning-MySQL-Tutorial/>

<http://dev.mysql.com/doc/refman/5.0/en/tutorial.html>

<http://www.mp3tag.de/en/>

<http://en.wikipedia.org/wiki/ID3>

<http://www.dmst.aueb.gr/louridas/lectures/dais/uml/uml.html>

<http://people.cis.ksu.edu/>

<http://www.softwarepractice.org/wiki>

http://www.simventions.com/whitepapers/uml/3000_borcon_uml.html

<http://www.scribd.com/>

<http://www.pageresource.com/html/image5.htm>

<http://css.dzone.com/news/creating-xml-documents-php>

<http://ca.php.net/manual/en/ref.domxml.php>

<http://www.developerfusion.com/code/3944/how-to-create-xml-files/>

<http://alexandre.alapetite.fr/doc-alex/domxml-php4-php5/index.en.html>

http://groups.google.com/group/Google-Maps-How-Do-I/browse_thread/thread/67f0bbab12580257?pli=1

<http://www.tonymarston.net/php-mysql/dom.html>

<http://www.charset.org/>

<http://forums.codewalkers.com/php-applications-45/saving-data-in-xml-file-using-php-html-4549.html>

<http://stackoverflow.com/questions/4445825/converting-text-file-into-xml-using-php>

<http://webdeveloper.com/forum/showthread.php?t=165648>

<http://www.hotscripts.com/forums/php/27096-write-xml-file-php.html>

<http://www.webdeveloper.com/forum/archive/index.php/t-153784.html>

<http://www.techsupportforum.com/forums/f49/dividing-html-into-sections-303112.html>

<http://www.petefreitag.com/item/466.cfm>

<http://www.titov.net/>

<http://www.totallyphp.co.uk/text-file-hit-counter>

http://www.premiumbeat.com/flash_music_players/original/multiple_menu

<http://www.gerd-tentler.de/tools/phpgraphs>

http://help.adobe.com/el_GR/InDesign/5.0/help.html?content=WS014B14E7-381A-440b-8A34-41C90C04B071.html

<http://mytwocents.gr/apache-web-server/>

<http://www.isa.edu.gr/Konstantinou>

<http://www.codeproject.com/Articles/9913/Project-Estimation-with-Use-Case-Points>

<http://www.internetnow.gr/node/86>

<http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-Dreamweaver.html>

ΒΙΒΛΙΟΓΡΑΦΙΑ

Chatzigeorgiou, N.A (2005) Αντικειμενοστρεφής Σχεδίαση: UML, Αρχές, Πρότυπα και Ευρετικοί Κανόνες, Εκδόσεις Κλειδάριθμος, Αθήνα

Dervos, D.A. (1995), Μαθήματα βάσεων δεδομένων, Εκδόσεις Τζιόλα, Θεσσαλονίκη

Pagkalos G. (2004), Βάσεις Δεδομένων (Τράπεζες πληροφοριών), Αφοι Κυριακίδη, εκδ. Ζ', Θεσσαλονίκη

Fowler, M. and Scott, K. (2001), Εισαγωγή στην UML (UML Distilled: A Bried Guide to the Standard Object Modeling Language), Μετάφραση: Εκδόσεις Κλειδάριθμος, Αθήνα

Hart-Davis, G (2010), HTML, XHTML & CSS Quick Steps, Osborne/McGraw-Hill, New York, United States

Karlins, D. (2008), Adobe Dreamweaver CS4 How-Tos: 100 Essential Techniques, Pearson Education (US), United States

Larman, C. (2004) Applying UML and Patterns: An introduction to Object-Oriented Analysis and Design and the Unified Process, 3rd edn.,Prentice Hall, Upper Saddle River, NJ

Lengstorf, J. (2009), PHP for Absolute Beginners, Springer-Verlag New York, United States

Melonie, J. C. (2003), Μάθετε PHP, MySQL και Apache : Όλα σε ένα μετάφραση Γιάννης Β. Σαμαράς. - 1η έκδ. Γκιούρδας Μ., Αθήνα

Niederst Robbins, J. (2007), Learning Web Design, Third Edition, A Beginner's Guide to (X)HTML, StyleSheets, and Web Graphics, O'Reilly Media, Sebastopol, CA

Pfleeger, S.L. (2001) Software engineering : Theory and Practice, 2nd edn., Prentice Hall, Upper Saddle River, NJ

Ramakrishnan, R. and Gehrke, J. (2002) Συστήματα διαχείρισης βάσεων δεδομένων (Database Management Systems), Μετάφραση : Εκδόσεις Τζιόλα, Θεσσαλονίκη

Schaeffer, M. (2008), Adobe Flash Cs4 Professional How-Tos: 100 Essential Techniques, Adobe, Berkeley, CA .

Skidmore, S. (1995), Εισαγωγή στην ανάλυση συστημάτων, Μετάφραση : Εκδόσεις Κλειδάριθμος, Αθήνα

Sommerville, I. (2000), Software Engineering, 6th edition, Addison-Wesley, Harlow, UK.