|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Revision | Impl. Date | Author | Revision Description |
| 0.1 | A | 21-2-2015 | **Ανδρέας Γούναρης** | Initial document (DRAFT for Comments) |
| 1.0 |  |  |  |  |

Συμβάσεις Σχεδιασμού και Υλοποίησης

Οι γενικοί κανόνες που θα ακολουθήσουμε κατά την διαδικασία ανάπτυξης της εφαρμογής έχουν στόχο να εξασφαλίζουν ομοιογένεια, διαχειρισιμότητα, διαφάνεια και επαναχρησιμοποίηση του λογισμικού. Αυτά τα στοιχεία, πέραν του ότι διασφαλίζουν αποδοτική συνεργασία μεταξύ των μελών της ομάδας, αυξάνουν και την προστιθέμενη αξία του προϊόντος, το οποίο γίνεται διαχειρίσιμο, προσαρμοστικό και έτοιμο για χρήση από οποιονδήποτε θελήσει στο μέλλον να το χρησιμοποιήσει ή να το επεκτείνει.

Τεχνολογίες 2

Υλοποίηση 3

Βάση δεδομένων 3

Δομή φακέλων λογισμικού. 3

# Τεχνολογίες

Πλατφόρμα εφαρμογής: Java 1.7  
Ο κώδικας θα είναι συμβατός με την έκδοση της **Java 1.7** και άνω. Παρόλο που η τελευταία έκδοση 1.8 προσφέρει νέες κλάσεις και μεθόδους, δεν είναι ακόμα ευρέως διαδεδομένη στο σύνολο των λειτουργικών συστημάτων Windows, Linux, OSX και των εκδόσεών τους.   
Κρίθηκε λοιπόν απαραίτητο να διατηρηθεί η μέγιστη συμβατότητα με άλλα λειτουργικά συστήματα, τα οποία δεν έχουν εγκατεστημένη την νέα έκδοση της Java.

Βάση Δεδομένων: JavaDB Derby  
Η βάση δεδομένων RDBMS δημιουργείται τοπικά στον υπολογιστή από το εργαλείο Nebtbeans και την έκδοση **glassfish-4.1** Η εφαρμογή RadioStation έχει πρόσβαση στη βάση μόνο από το ίδιο μηχάνημα (localhost).

Εργαλείο ανάπτυξης (IDE): Netbeans 8.x  
Η δημιουργία της βάσης, η ανάπτυξη του κώδικα και όλων των διεπαφών GUI θα γίνει μέσα από το περιβάλλον του **Netbeans 8.x**.

Εργαλείο ελέγχου εκδόσεων (VCS): Git – Google Code  
Η αποθήκευση του πηγαίου κώδικα και της τεκμηρίωσης της εφαρμογής θα γίνει στο περιβάλλον Google Code. Ως πλατφόρμα αποθήκευσης έχει επιλεγεί το **Git** λόγω της ευελιξίας που προσφέρει στην ομαδική ανάπτυξη, ακόμα και αν δεν υπάρχει σύνδεση με το remote repository.

# Υλοποίηση

Σχεδιασμός - Διαγράμματα: Visual Paradigm 11.x CE  
Ολα τα διαγράμματα σχήματος ΒΔ, διαγράμματα κλάσεων, περιπτώσεων χρήσης κλπ θα γίνουν από το εργαλείο αυτό. Μολονότι το Visual Paradigm υποστηρίζει την αυτόματη δημιουργία scripts και κώδικα, η λειτουργία αυτή δεν είναι διαθέσιμη από την έκδοση Community Edition, έτσι θα περιοριστούμε μόνο στην κατασκευή των διαγραμμάτων.

## Βάση δεδομένων

* Το διάγραμμα της ΒΔ σχεδιάζεται στο Visual Paradigm (ER Diagram).
* Το διάγραμμα υλοποιείται χειροκίνητα και καταγράφεται στο schema script με κατάλληλα sql statements.
* Tο διάγραμμα είναι πάντα επικαιροποιημένο. Κάθε αλλαγή που απαιτείται στη ΒΔ αποτυπώνεται πρώτα σε αυτό και μετά στο schema script.
* Ενδεχόμενες αλλαγές στο σχήμα της βάσης όταν αυτή θα έχει ήδη δεδομένα, θα γίνεται με scripts και ALTER TABLE statements. Κάθε τέτοιο script θα έχει τα διακριτικά της έκδοσης στο όνομά του κατά αύξοντα αριθμό, π.χ.
  + RadioStation\_DBSchema\_init.sql
  + RadioStation\_DBSchema\_alter\_01.sql
  + RadioStation\_DBSchema\_alter\_02.sql
  + …

## Δομή φακέλων λογισμικού.

-| RadioStation : κύριος φάκελος (root)

-| build : run-time classes

-| dist : βιβλιοθήκες και jars

-| docs : τεκμηρίωση, διαγράμματα

-| nbproject : φάκελος του Netbeans

-| scripts : SQL scripts

-| src : πηγαίος κώδικας

Πρακτική χρήσης Source Control (Git)  
Το repository βρίσκεται στο Google Code και θα είναι συμβατό με Git. Η πρόσβαση είναι ανοικτή σε όλους μέσω του συνδέσμου <https://code.google.com/p/eap-radio-station/>

* Κάθε μέλος της ομάδας έχει δικό του branch και μπορεί να εργάζεται ανεξάρτητα.
* Στο τέλος κάθε επιμέρους εργασίας, το μέλος της ομάδας που την ολοκλήρωσε εκτελεί merge του branch με το master track.
* Μια νέα έκδοση στο master track θεωρείται ως RC (release candidate) και πρέπει να περάσει τον απαραίτητο έλεγχο (unit test). Οταν ο έλεγχος είναι επιτυχής, η έκδοση σημειώνεται στο master track με την μέθοδο tagging, ακολουθώντας τη σύμβαση αυξητικής αρίθμησης 1.x.x.A:
  + Το πρώτο ψηφίο ορίζει την κύρια έκδοση. Στα πλαίσια της εργασίας του ΕΑΠ, το ψηφίο θα παίρνει πάντα την τιμή 1.
  + Το δεύτερο ψηφίο ορίζει την υποέκδοση, όταν αυτή προσθέτει καινούργια λειτουργικότητα, π.χ. νέα έκδοση που επιτρέπει την εξαγωγή δεδομένων σε XML.
  + Το τρίτο ψηφίο ορίζει την διόρθωση σφαλμάτων προηγούμενων εκδόσεων.
  + Το τέταρτο μέρος είναι αλφαριθμητικό και παίρνει την τιμή **RC** (Release Candidate) όταν υπόκεινται σε έλεγχο QA ή την τιμή **Stable** όταν ο έλεγχος έχει ολοκληρωθεί επιτυχώς. Για παράδειγμα:
  + α) **RadioStation 1.1.2.RC** η έκδοση 1.1.2 βρίσκεται σε διαδικασία ελέγχου
  + β) **RadioStation 1.1.2.Stable** η έκδοση 1.1.2 έχει περάσει επιτυχώς τον έλεγχο και είναι εκτελέσιμη.
* Κάθε νέο commit θα πρέπει να συνοδεύεται από σχόλια σχετικά με το τι περιλαμβάνει η έκδοση. Προτείνεται χωρίς δέσμευση η εξής σύμβαση:

NEW:R4,R6, UPD:R2, FIX:R2,R1

Όπου το NEW είναι υλοποίηση της απαίτησης R4 και R6, το UPD είναι διόρθωση ή επέκταση της απαίτησης R2 και το FIX είναι αποσφαλμάτωση στην υλοποίηση των R2 και R1.

Μοτίβο σχεδίασης λογισμικού (design patterns)  
*...to be defined (JPA, Pojo, Class Interface, DAOs, Spring etc)*