



Institut
de Formation Supérieure de la Ville de Wavre
IFOSUP

Bachelier en informatique de gestion.

Ankh Fire
Gestionnaire de tags audios

Directeur de travail :
Ph. Robillard

Travail de fin d'études présenté par Thomas STASSIN en vue de l'obtention du
diplôme de Bachelier en Informatique de Gestion

Année académique 2013-2014



Institut
de Formation Supérieure de la Ville de Wavre
IFOSUP

Bachelier en informatique de gestion.

Ankh Fire
Gestionnaire de tags audios

Directeur de travail :
Ph. Robillard

Travail de fin d'études présenté par Thomas STASSIN en vue de l'obtention du
diplôme de Bachelier en Informatique de Gestion

Année académique 2013-2014

Table des matières

1	Introduction	2
2	Contexte	3
2.1	Le client	3
2.2	La demande initiale	3
2.3	Les produits existants	4
3	Cahier des charges	5
3.1	Elaboration	5
3.2	Lots d'information identifiés	5
3.2.1	Lots d'information existants à conserver	7
3.2.2	Lots d'information existants à remplacer	7
3.2.3	Lots d'information à produire	7

Chapitre 1

Introduction

Dans le cadre de mes cours d'informatique de gestion à l'IFOSUP, je dois réaliser un travail de fin d'études. Je donc choisis de faire une application de gestion de fichiers audios afin de pouvoir mettre en pratique les connaissances acquises durant ses trois années d'études.

C'est pour moi aussi l'occasion de réaliser un projet du début à la fin, étant à la fois analyste fonctionnel, analyste technique et développeur.

De plus, la conception de cette application va me permettre de travailler sur avec des technologies qui me tiennent à cœur et de les éprouver dans une situation "professionnelle".

Les multiples facettes de ce programme (un côté client et autre server) vont me confronter avec plusieurs aspects du développement informatique dans le but qu'à la fin de la réalisation de ce travail, j'en ressorte avec le maximum d'expérience concrète.

Chapitre 2

Contexte

Le développement d'une application de gestion de fichiers audios, permettant leur lecture et leur classification à l'aide de tags¹.

Cette application servira à mon client, pour créer des listes de lectures rapidement, afin de pouvoir diffuser des musiques adéquates lors d'événements qui requièrent plusieurs styles d'ambiances audios.

Cette application sera développée pour pouvoir être déployée sur plusieurs plateformes (Linux, Mac-OS, Windows) et pourra se connecter à un serveur pour pouvoir partager et récupérer les tags liés aux fichiers audio.

2.1 Le client

Mon client est un particulier, qui organise des événements où la musique d'ambiance est un élément important du "décor".

Un exemple de ce type d'événement est une soirée jeu de rôles. Lors de ce genre de soirée, l'ambiance sonore est souvent appréciée et si elle est présente, sa gestion doit être fluide, afin de ne pas créer de rupture dans l'ambiance.

2.2 La demande initiale

Mon client cherchait une application qui permettait de taguer des musiques et créer automatiquement des listes de lectures basées sur les tags.

Cette application devrait aussi proposer de pouvoir créer un "tableau de bord" qui permettrait de passer d'une liste de lecture à une autre (avec une phase out / phase in), cela de manière ergonomique en demandant le moins de manipulation possible.

1. Mot anglais désignant une "étiquette" (ou un mot-clé) associé ou assigné à des données (dans ces cas-ci un fichier audio).

Et enfin, les utilisateurs de cette application pourraient partager leurs tags entre eux, via un serveur qui stockerait les tags enregistrer par chacun et permettrait la diffusion de ceux-ci. L'application demandée tournera sur les trois Systèmes d'exploitation suivant :

- Linux
- Windows
- Mac-OS

2.3 Les produits existants

En matière de diffusion de musique audios, il y a pléthore de programme existant. Voici deux exemples que j'ai choisis, car ils ont une caractéristique commune avec l'application demandée : ils sont exécutables sur les 3 OS cité plus haut.

- Quodlibet
- Amarok
- Banshee (en Beta sous Mac-OS et en alpha sous Windows)

Des logiciels qui permettent de taguer efficacement un fichier audio, en permettant de créer ses propres tags, je n'en ai trouvé qu'un :

Ex Falso. C'est un éditeur de tags, livré avec Quodlibet.

Dans la totalité de ces programmes, aucun ne permet de faire de la création de playlist dynamique, du moins pas de manière poussée, et aucun ne permet la création de "tableau de bord" comme souhaiter par mon client. De plus, la partie serveur, avec partage des tags serait quelque chose de totalement neuf, n'ayant pas d'équivalence.

Chapitre 3

Cahier des charges

Nous avons, le client et moi, établis que chaque partie de l'application serait abordée séparément. Comme abordé précédemment, il y a trois parties distinctes :

1. L'édition de tags.
2. La création et le jeu de liste de lecture.
3. Le partage des tags via d'un serveur.

3.1 Elaboration

Lors des premiers rendez-vous, nous avons abordé chaque partie du programme.

Editeur de tags

Le client désire pouvoir encoder les tags de ses musiques à partir de l'application. Il a bien précisé que le but de cet éditeur n'était pas de remplacer Ex Falso¹

Nous avons établi 3 cas d'utilisations :

- Editer un tag dans un fichier audio.
- Supprimer un tag dans un fichier audio.
- Ajouter un tag dans un fichier audio.

3.2 Lots d'information identifiés

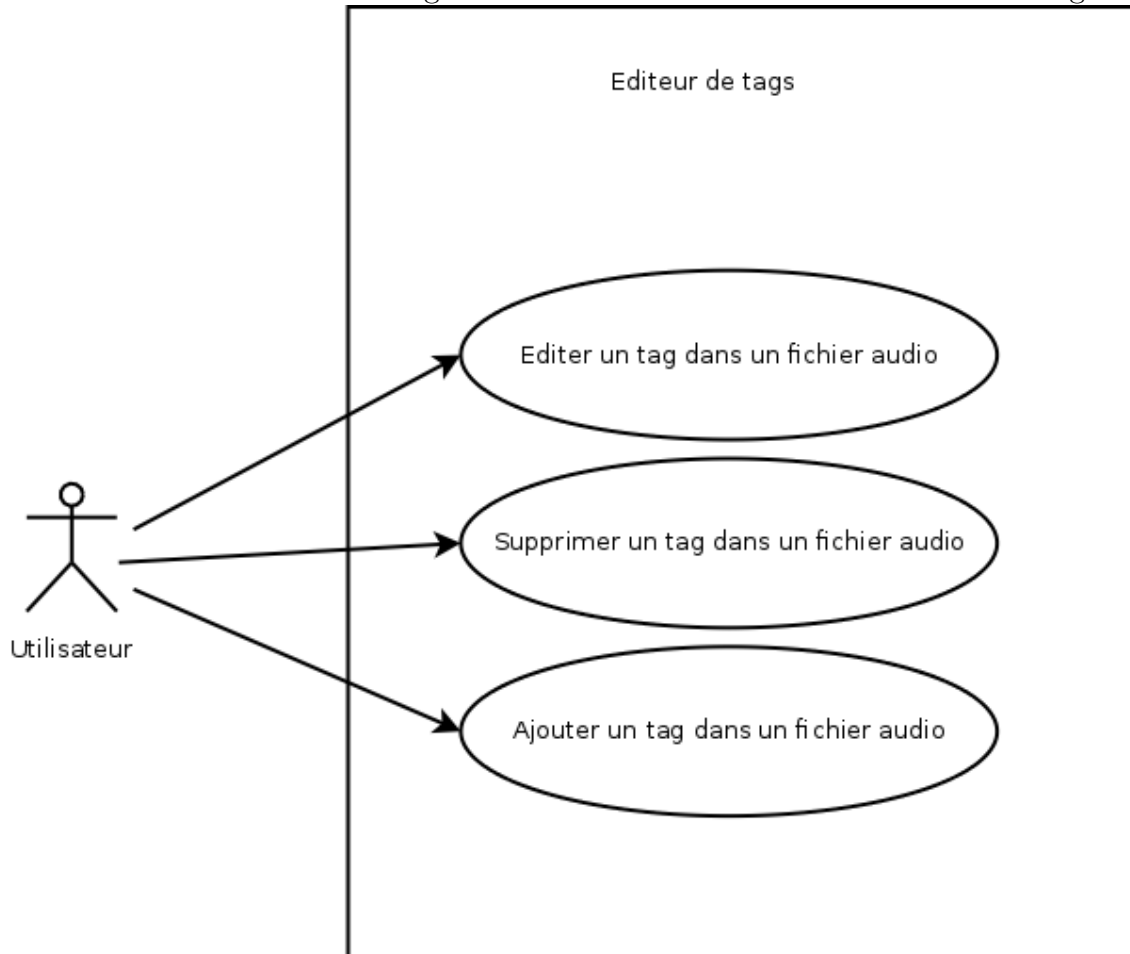
Editeurs de tags

Le client m'a fourni une liste des tags qu'il voudrait que l'application propose par défaut sans devoir faire d'effort supplémentaire de son choix. Ces tags sont les suivants :

- title
- artist
- album

1. module de Quolibet permettant d'éditer les tags d'un fichier audio.

FIGURE 3.1 – Diagramme des cas d'utilisations de l'éditeur de tags



- artistsort
- date
- discnumber
- discsubtitle
- disctype
- genre
- labelid
- replaygain
- replaygain_album_gain
- replaygain_album_peak
- replaygain_track_gain
- replaygain_track_peak
- romanized_artist
- romanized_album
- romanized_title
- tracknumber
- translated_title
- translated_album

3.2.1 Lots d'information existants à conserver

Les fichiers audio existant chez le client seront conservés. De la même façon les tags ID3 déjà présent sur ces fichiers resteront tel quel.

3.2.2 Lots d'information existants à remplacer

Aucun lots d'information existants n'est à remplacer.

3.2.3 Lots d'information à produire