



AudioWire

Hugo Defrance, Huai Xie, Stéphane Rose

Guillaume Feraud, Augustin Pasquini,

Vivien Meilhac, Maximilien Brunelle

Résumé du document

Le projet AudioWire mènera à la réalisation d'un lecteur audio doté d'une intelligence artificielle. Celui-ci permettra la création de listes de lectures adaptées à l'utilisateur en fonction de son humeur par exemple. De plus, il bénéficiera de fonctionnalités sociales. Ce document a pour but de présenter la réalisation de ce projet en détails.

Tout d'abord, nous rappellerons ce qu'est un EIP à Epitech et quel est son but. Puis, nous présenterons de manière détaillée les fonctionnalités attendues à l'issue du projet. Ensuite, nous détaillerons le contexte économique dans lequel se situe le projet AudioWire ainsi que la population ciblée.

Ensuite, nous détaillerons les différentes parties du programme avant de nous intéresser aux détails de réalisation du projet en lui-même. Dans cette partie, nous nous intéresserons particulièrement à l'architecture technique du projet, aux composants déjà présents sur le marché puis nous verrons les aspects concernant la sécurité du projet et ces points sensibles.

De plus, nous étudierons les différents outils et contraintes relatifs au projet avant de comparer les différents types de bases de données présentes sur le marché et expliquerons notre choix dans ce domaine.

Enfin, nous donnerons une liste des principales fonctionnalités qui devront être testées et validées avant la fin du projet: Pour finir, nous détaillerons les étapes majeures de la réalisation du projet.

Description du document :

Titre	AudioWire – Cahier des Charges
Date	18/04/2012
Auteurs	Huai Xie, Stéphane Rose, Guillaume Feraud, Maximilien Brunelle, Augustin Pasquini, Hugo Defrance, Vivien Meilhac
E-Mail	audiowire_2014@labeip.epitech.eu
Sujet	Cahier des Charges
Mots- clés	
Version	1.1

Tableau des révisions :

Date	Auteurs	Section(s)	Commentaire
04/04/12	L'ensemble du groupe	Toutes	Première Version du Cahier des Charges
18/04/12	Defran_h, rose_s, xie_h, brunel_m	Toutes	Deuxième version du Cahier des Charges
04/07/12	L'ensemble du groupe	Toutes	Version finale du Cahier des Charges

Table des matières

1. Rappel de l'EIP

- a. Epitech, l'école de l'innovation et de la passion informatique
- b. Epitech Innovative Project
- c. Les objectifs

2. Sujet de l'EIP

3. Contexte économique et population ciblée

4. Description des parties du programme à réaliser

5. Détails sur le projet

- a. Architecture technique
- b. Composants existants
- c. Questions de sécurité
- d. Points sensibles

6. Outils et contraintes

- a. Outils logiciels
- b. Outils matériels
- c. Contraintes

7. Base de données

8. Tests de premier niveau

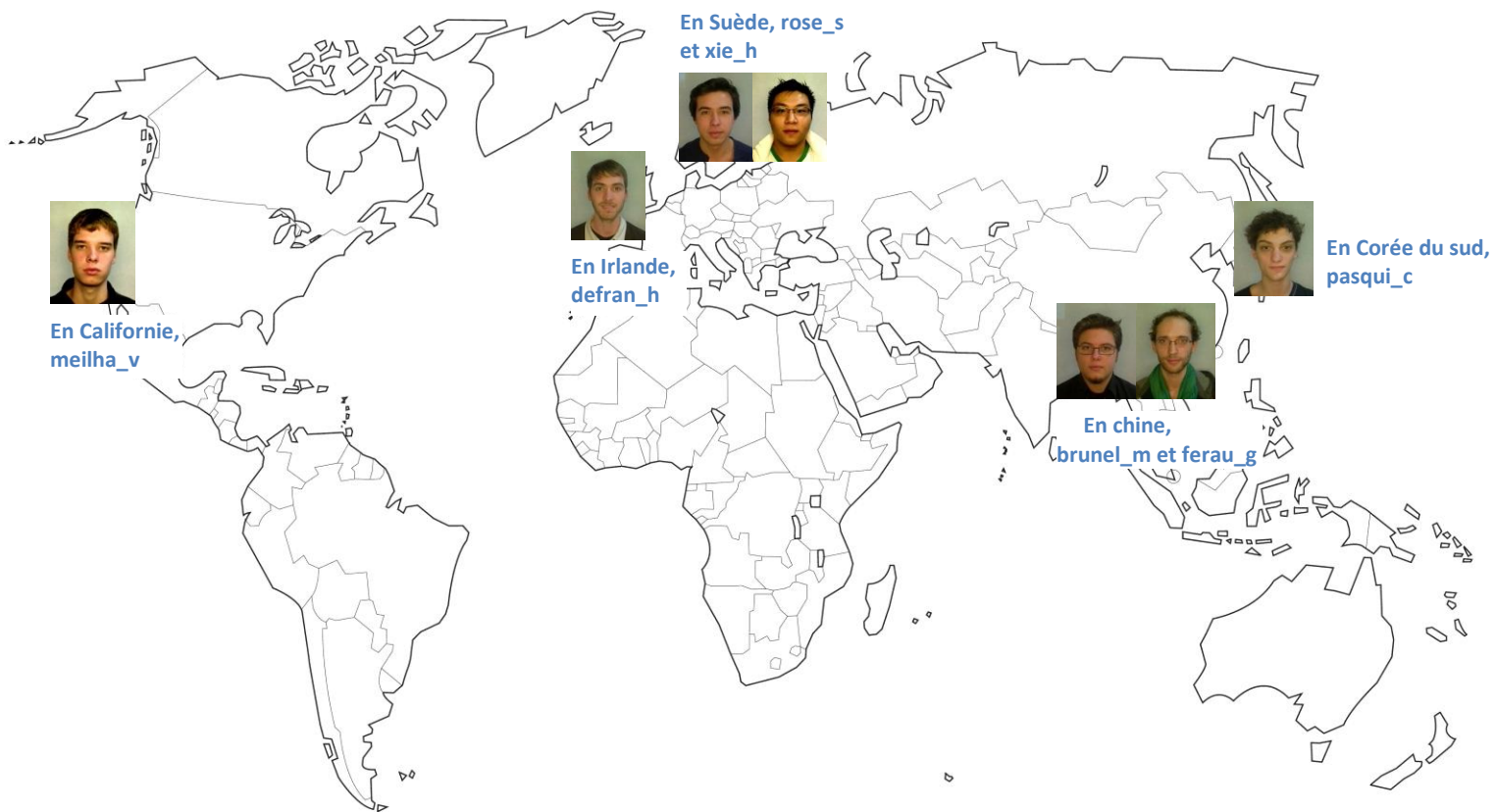
- a. L'interface
- b. L'intelligence artificielle (IA)

9. Organisation du projet

10. Annexes

11. Conclusion

Où sommes-nous ?



1. Rappel de l'EIP

a. Epitech, l'école de l'innovation et de la passion informatique

Epitech, est une école d'expertise en informatique créée en 1999 dans la mouvance de l'Ecole pour l'Informatique et les Techniques Avancées (Epita), pour accueillir les bacheliers passionnés qui souhaitent apprendre l'informatique par la pratique et non la théorie. Le cursus se déroule sur 5 ans et fourni un diplôme d'expert en technologies de l'information reconnu en tant que diplôme de niveau I par la CNCP.

La particularité et la force de cette école repose principalement sur le fait que les étudiants sont amenés à apprendre par eux-mêmes les notions. Notions recherchées dans le cadre de mini-projets de groupes et de projets plus ambitieux comme le PFA (Projet de Fin d'Année) et l'EIP (Epitech Innovative Project).

En effet, contrairement à l'enseignement scolaire classique où l'on amène la connaissance à l'élève afin qu'il puisse résoudre des problèmes, un étudiant à Epitech devra rechercher de lui-même ces notions. Notions nécessaires, dont il a besoin avec son groupe de travail pour résoudre un problème ambigu dans lequel il n'aura aucune base. C'est de ce principe que naîtra l'émergence du comportement que recherche Epitech, à savoir la capacité d'adaptabilité par rapport à une problématique et des concepts, ainsi que la gestion d'une équipe de travail.

b. Epitech Innovative Project

L'Epitech Innovative Project appelé aussi EIP, est un projet de groupe conséquent et innovant dans le domaine de l'informatique. Il doit être réalisé sur une période de 18 mois et est constitué de 5 élèves au minimum. L'EIP illustre parfaitement la pédagogie enseignée par l'école. Il permet de responsabiliser les étudiants sur la recherche et le développement d'une idée

novatrice dans le but de l'amener à la fin de sa réalisation dans les délais impartis, aussi bien sur le plan technique que sur le plan administratif.

Ce projet représente une étape majeure dans le cursus d'un élève à l'Epitech car il lui permet de passer du statut d'étudiant à celui de professionnel. En effet, la réussite de ce projet permettra à la plupart des étudiants de s'installer sur le marché professionnel en créant leur propre entreprise, ou alors en rejoignant une firme déjà existante qui serait intéressée par le concept.

c. Les objectifs

Il faut savoir que l'EIP n'est pas seulement un projet destiné à montrer les connaissances que l'on a acquises lors de notre passage à Epitech. En effet, la technologie et plus particulièrement le domaine de l'informatique est un domaine qui évolue constamment. C'est pourquoi la connaissance à l'état pur n'est pas vraiment une plus-value et n'est que passagère car très vite obsolète. Un des buts principaux de l'EIP est d'apprendre aux étudiants à dépasser les aspects techniques d'un projet en informatique afin de développer la partie documentation et communication extérieure.

Cet aspect pédagogique du projet est très important étant donné que ce sont des correcteurs extérieurs qui jugeront le projet et le travail fournis pendant ces deux années lors de la présentation orale de la soutenance.

2. Sujet de l'EIP

Notre projet AudioWire est un lecteur de musique évolué.

Il présente les fonctionnalités suivantes :

- Gestionnaire de bibliothèque
- Gestion des Playlists
- Playlists "Intelligente" (Algorithme d'apprentissage en fonction des goûts et des envies de l'utilisateur) avec des algorithmes liés au data-mining.
- Lecture synchronisée sur le réseau local (deux postes à deux endroits jouent le même morceau en même temps - le morceau n'étant pas forcément sur les deux postes)
- Visualizer
- Possibilité d'écouter des radios et des podcasts

De plus ce logiciel proposera des fonctionnalités plus axées "web" :

- Un chat et une liste d'amis
- Partage de fichiers
- Partage des goûts (Playlists, Playlists intelligentes ...)
- Streaming inter-utilisateurs, c'est-à-dire la possibilité d'écouter en même temps une même musique entre plusieurs utilisateurs.

AudioWire sera compatible Windows, Mac et Linux. Nous avons également prévu de le porter sur iOS et Android.

3. Contexte économique et population ciblée

De nos jours de très nombreux lecteurs audio existent et offrent de multitudes de possibilités. Tous ces lecteurs tels que iTunes ou Windows Media Player, des fonctionnalités similaires comme un gestionnaire de bibliothèque ou de encore de playlist. A cela, s'ajoutent aussi d'autres fonctionnalités spécifiques à chaque lecteur, « Genius » sur iTunes, ou encore certaines options de Winamp, etc. C'est dans ce contexte que Audiowire fera son apparition, c'est là où il devra montrer sa supériorité face aux autres lecteurs.

Audiowire sera une application totalement gratuite, et fonctionnera sur Windows, Mac, et Linux.

Notre but est de venir concurrencer les autres lecteurs connus en nous servant des fonctionnalités qui ont le plus de succès et en ajoutant des fonctionnalités nouvelles (Streaming inter-utilisateur, synchronisation sur le réseau local, etc.)

AudioWire sera destiné au grand public. En effet nous pensons qu'il est primordial de rendre l'utilisation de notre logiciel la plus simple possible. En le réalisant dans ce but, nous nous assurons de pouvoir toucher un nombre plus grand d'utilisateurs.

Il est cependant important qu'un utilisateur plus avancé puisse définir d'autres options, et nous laisserons donc un moyen de faire des réglages plus poussés, sans que cela nuise à l'utilisation d'un utilisateur lambda.

4. Description des parties du programme à réaliser

Le projet AudioWire se compose de différentes parties.

- L'intelligence artificielle :

L'intelligence artificielle est la partie principale de notre EIP. C'est pourquoi nous avons mis en place un partenariat avec le laboratoire Acsel en plus du laboratoire EIP, auquel nous avons à rendre des comptes mensuellement. L'IA sera capable de classifier les données en fonction de différents facteurs qui vont être implémentés :

- o L'humeur de l'utilisateur : qu'il sélectionne manuellement ou alors que le programme arrive à déterminer.
- o L'analyse des fichiers audio, que ce soit des flux audio, ou des métadonnées.
- o Le recoupement des informations est aussi pris en compte : quel temps fait-il, quelle période de l'année sommes-nous, que s'est-il passé récemment dans l'actualité, suis-je tout seul ou en soirée ?

Nous allons utiliser le logiciel WEKA afin d'analyser les flux audio, y extraire les indices musicaux.

La récupération des flux audio est une conséquente partie de l'IA, et comprend différentes données à traiter :

- o Les fonctions de hachage qui permettent de comparer les différents flux avec leur hach.
- o Le BPM.
- o Les métadonnées.
- o Le tempo (rythme).
- o Les breaks.
- o La gestion des styles simple.

- Le client :

L'EIP AudioWire est un projet très technique. Cependant, il est primordial de développer une interface client agréable à utiliser, "user friendly", moderne.

Les deux membres du groupe qui s'occuperont de cette partie devront mettre en place les fonctionnalités du projet développées du côté de l'intelligence artificielle.

Le développement du client devra respecter de un plan d'action définit dans les UML et diagrammes de transitions à venir. Le client donnera accès à un profil détaillé de l'utilisateur, et permettra d'utiliser toutes les fonctionnalités de l'IA de manière ergonomique.

- Les applications mobiles :

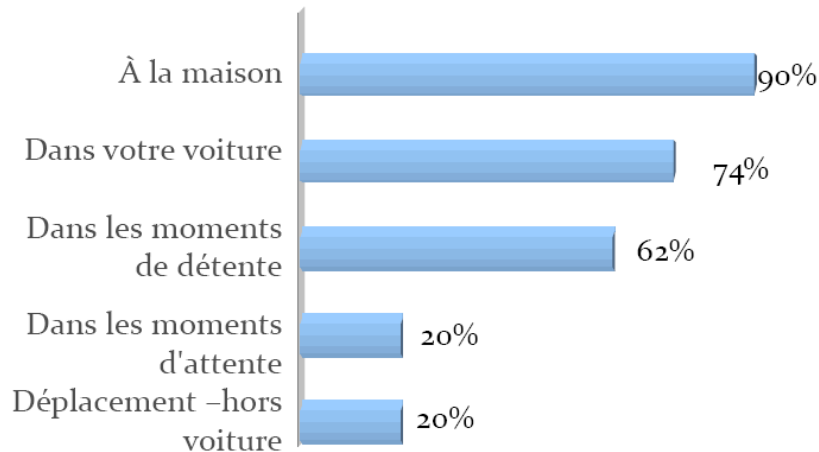
Une importante partie d'AudioWire va se développer sur mobile. En effet, il est prévu de développer une application iPhone et une application Android de lecture de la musique, de manière intelligente. Elles seront disponibles gratuitement au téléchargement sur leur « market » respectif. Les applications intégreront toutes les fonctionnalités principales du projet, notamment la lecture de musiques en fonction de nombreux paramètres, définit par l'utilisateur et l'intelligence artificielle.

- Pourquoi des applications mobiles ? :

Le développement des applications d'Audiowire sur iPhone et sur Android permettra d'étendre les possibilités d'utilisation du projet.

En effet, selon un sondage réalisé par l'opinionWay en 2011 (cf. annexe 6) auprès d'un échantillon de 2000 personnes, on se rend compte que la musique est la troisième activité culturelle de la population française derrière la lecture et la télévision. De plus, on peut voir sur le schéma présenté ci-après que 62 % des français écoutent la musique dans les moments de détente, 20% dans les déplacements hors voiture et 20% dans les moments d'attente. Chez les 15-24 ans l'écoute moyenne par jour se révèle à 1h30 par jour, certaines personnes ont même dit écouter la musique plus de deux heures par jour.

À quel moment ou dans quelles conditions écoutez-vous de la musique ?



C'est pourquoi nous avons décidé d'étendre notre logiciel sur les plateformes mobiles. De cette manière, il serait possible de récupérer encore plus d'information sur le mode de vie des utilisateurs ainsi que sur leurs goûts musicaux pendant leur journée de travail, le week-end, pendant les vacances, etc ... Par exemple on pourra remarquer que certaines personnes préféreront écouter du jazz le matin dans le métro pour aller travailler alors qu'ils opteront pour un style plus rock le soir en rentrant chez eux.

L'application tournera en background sur les téléphones mobiles de manière à être totalement transparente aux yeux de l'utilisateur ce qui facilitera d'autant plus la facilité pour nous de récolter des informations sur le mode de vie des utilisateurs. Une synchronisation régulière avec le serveur se déclenchera automatiquement afin de compléter le profil du client.

Une interface graphique simple d'utilisation sera présente afin de faciliter sa compréhension et son paramétrage.

- La base de données :

Le projet va nous amener à stocker un très grand nombre de données, que ce soit au niveau des musiques que des utilisateurs. Notamment les métadonnées des musiques (genre, album, artiste, etc.), ou les informations récupérées par l'intelligence artificielle (décrites dans la partie dédiée). Le laboratoire EIP est d'accord pour que l'on emprunte un de leurs serveurs afin de stocker toute ces informations.

Cette base sera d'abord enrichie par un corpus de musique fourni par les développeurs, puis peu à peu par les utilisateurs.

5. Détails sur le projet

a. Architecture technique

Le lecteur multimédia sera utilisable sur différentes plateformes (Android, iPhone, Windows, Mac et Linux). Ces clients communiqueront par l'intermédiaire d'un serveur qui assurera la liaison avec la base de données.

Un site web est également prévu pour assurer la communication externe autour du projet et permettre aux utilisateurs d'utiliser les fonctions sociales du produit. Ce site web communiquera de façon directe avec la base de données.

Il existera également un serveur de développement qui sera une copie conforme du serveur de production. Celui-ci permettra de tester toutes les nouvelles fonctionnalités avant la mise en production. Ce qui nous garantira un serveur de production toujours fonctionnel avec une version stable du projet.

b. Composants existants

Afin de concentrer nos efforts sur l'intelligence artificielle, le corps de notre lecteur multimédia sera basé sur le corps d'un projet déjà existant. De plus, nous utiliserons des algorithmes d'intelligence artificielle de traitement des données audio. Ces dernières serviront au fonctionnement des options d'intelligence du lecteur.

c. Questions de sécurité

Afin de préserver la confidentialité des données, tous les transferts d'information jugés sensibles seront chiffrés par le protocole Transport Layer Security (TLS). Ce protocole garanti un chiffrement des données transmises grâce à un système de clé privée/publique.

En ce qui concerne le site internet, les connections non sécurisées (HTTP) seront automatiquement redirigées en connections sécurisées (HTTPS).

d. Points sensibles

Du fait de la nature du projet qui est largement basé sur des algorithmes d'intelligence artificielle, il est possiblement envisageable que certaines options du lecteur multimédia s'avèrent finalement irréalisable. Il est donc important de bien évaluer les possibilités offertes par les algorithmes existants afin de pallier aux éventuels problématiques rencontrées au plus vite.

La manière dont les données sensibles seront transférées et stockées devront faire l'objet d'une attention toute particulière afin de pouvoir garantir la confidentialité et la sécurité de celles-ci.

6. Outils et contraintes

a. Outils logiciels

Afin de réaliser notre EIP, nous comptons utiliser les outils suivants :

- **Weka 3** : il s'agit d'un logiciel que nous comptons utiliser lors de la phase de développement du projet afin de réaliser du data-mining dans des fichiers audio. Cette prospection de données nous permettra de développer notre intelligence artificielle.
- **C++** : ce langage sera majoritairement utilisé au cours du développement de la partie fonctionnelle des clients Windows, Mac et Linux.
- **Python** : python est un langage de script, qui est à la fois facile à prendre en main et permet d'obtenir un code rapidement fonctionnel et puissant. Ce langage est majoritairement utilisé pour les serveurs.
- **Java** : langage portable grâce à l'émulation de son propre environnement. Java sera utilisé pour le développement du client Android.
- **Cocoa/objective-C** : ces deux langages seront utilisés pour l'interface graphique du client Mac ainsi que pour la totalité du client iPhone.
- **C#/.net**: ce langage ainsi que cette API serviront à la réalisation de la partie graphique pour le client Windows.
- **Virtualenv** : ceci est un outil permettant d'isoler un environnement de travail en Python permettant ainsi aux développeurs de travailler sur un environnement commun, indépendamment de la machine hôte. Il sera utilisé sur le serveur gérant l'accès à la base de données.

Pour assurer une bonne communication au sein du groupe, il sera indispensable d'utiliser les outils suivants :

- **Skype** : Pour la communication interne au groupe.
- **Microsoft Live Meeting** : Pour la communication entre le laboratoire EIP et le groupe.

b. Outils matériels

Afin de mener à bien le projet, nous pouvons distinguer deux types d'outils : les outils matériels, et les outils logiciels.

- Les outils matériels sont un serveur gérant la base de données et le site web ainsi que les différents ordinateurs et téléphones nécessaires aux développeurs pour travailler.
- Les outils logiciels nécessaires aux développeurs afin de réaliser les différents clients de notre projet sont des IDE (Integrated development environment). Ces derniers seront Eclipse pour le développement du client Android, Microsoft Visual Studio pour le développement du client Windows et xCode pour le développement des clients Mac et iPhone. Pour le développement sous linux, les développeurs seront libres d'utiliser leur IDE préféré.

c. Contraintes

Comme dans tous projets, notre groupe sera confronté à différents types de contraintes :

- **Contrainte géographique** : les développeurs seront localisés sur différents continents, ce qui impliquera donc des problèmes de décalage horaires.
- **Projet évolutif** : étant donné qu'une intelligence artificielle est, de par sa nature, toujours évolutive, il est fort probable que cette dernière soit encore perfectible après la date de rendu du projet.
- **Contrainte de ressource** : le projet étant portable sur de nombreuses plateformes, certains développeurs seront dans l'incapacité de travailler sur certaines d'entre-elle à cause d'un problème de ressources limitées.

7. Base de données

La base de données est un point clé à ne pas négliger dans la réussite de notre EIP car c'est elle qui va permettre la gestion de toutes les informations dont on aura besoin aussi bien en termes d'utilisateur que d'intelligence artificielle. En effet, les données renseignées par les clients seront directement enregistrées afin de pouvoir être utilisées par les différents algorithmes de recherches de goût. Il va sans dire que plus la base sera importante, plus la recherche sera riche et les résultats intéressants.

Étant donné que le but premier de notre EIP n'est pas la gestion de base de données mais bien la réalisation d'algorithmes perfectionnés dans la recherche de goûts musicaux, il nous faut choisir un éditeur qui soit relativement simple à utiliser et qui nous permette de passer le moins de temps possible à sa conception. De plus il est nécessaire que la recherche et le traitement de ces données soit un tant soit peu optimisé dans le sens où plus le temps d'exécution des requêtes sera court et plus les résultats de nos algorithmes seront percutants.

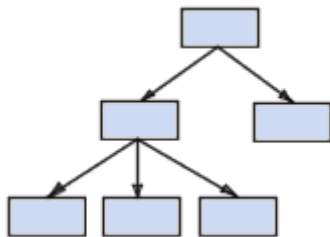
Dans un premier temps, nous avons exposé les différents modèles de bases de données existants, quels sont leurs caractéristiques, quel est leur rôle dans un projet informatique et quels sont les avantages et les inconvénients de ces différents modèles. Puis nous avons essayé d'en déduire la solution qui nous paraît la plus judicieuse pour le bon déroulement du projet en prenant compte des contraintes techniques mais également des contraintes temporelles.

Les modèles de bases de données (cf annexe 1)

En informatique, une base de données est un lot d'informations stockées dans un dispositif de stockage. Aujourd'hui, elles accompagnent quasiment tout projet informatique commercial car c'est elle qui va permettre de garder en mémoire toutes les informations nécessaires au fonctionnement de l'application. Il est possible de modifier et d'interpréter ces données à l'aide de différents langages plus ou moins complexes en fonction des bases (cf annexe 2). Ces derniers seront expliqués un peu plus bas mais également dans les documents présentés en annexe. Il existe plusieurs modèles de base de données. Nous expliquerons succinctement quels sont les différents modèles que l'on peut trouver et comment ils s'organisent.

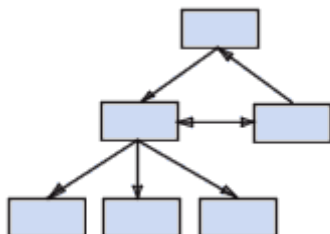
Le modèle hiérarchique :

Les données sont classées hiérarchiquement, selon une arborescence descendante. Ce modèle utilise des pointeurs entre les différents enregistrements. Il s'agit du premier modèle de SGBD



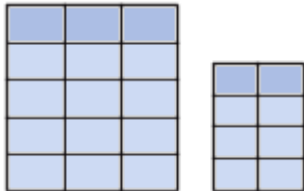
Le modèle réseau :

Comme le modèle hiérarchique ce modèle utilise des pointeurs vers des enregistrements. Toutefois la structure n'est plus forcément arborescente dans le sens descendant



Le modèle relationnel (SGBDR, Système de gestion de bases de données relationnelles) :

Les données sont enregistrées dans des tableaux à deux dimensions (lignes et colonnes). La manipulation de ces données se fait selon la théorie mathématique des relations

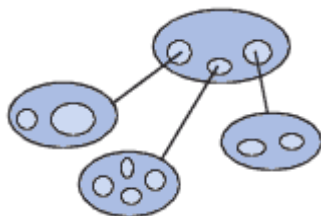


Le modèle déductif :

Les données sont représentées sous forme de table, mais leur manipulation se fait par calcul de prédicats

Le modèle objet (SGBDO, Système de gestion de bases de données objet) :

Les données sont stockées sous forme d'objets. Les champs sont des instances de ces classes.



Il faut savoir qu'à la fin des années 90, les bases relationnelles sont les bases les plus répandues dans le monde (environ $\frac{3}{4}$ des bases de données). Nous allons donc nous intéresser à ce modèle précis de base de données. D'un point de vue technique, il est plus apte à répondre à nos attentes pour notre EIP, c'est-à-dire qu'il y a toujours une relation entre les données en base, par exemple, il y aura toujours un lien entre un artiste et un album ou une chanson.

Modèle relationnel

Selon le modèle relationnel, il peut y avoir plusieurs relations connexes. Dans une base de données relationnelle chaque enregistrement d'une table contient un groupe d'informations relatives à un sujet et les différents sujets sont connexes. Les liens existants entre les informations sont stockés dans les champs des enregistrements (composants des n -uplets) sous forme de clé primaire et clé étrangère. Le n -uplet est une collection de n objets tel qu'il soit possible de dire exactement celui qui est le premier élément, le second élément, ..., le n ème. Les éléments sont aussi appelés composantes.

Une composante N d'un n -uplet est une clé primaire lorsque par nature la valeur de cette composante est différente pour chaque n -uplet d'une relation, et que pour une valeur donnée, il n'existe qu'un seul n -uplet dans toute la relation dont la composante a cette valeur (absence de redondance).

Une composante N d'un n -uplet est une clé étrangère lorsque les valeurs de cette composante sont des références à une clé primaire. Il y a une situation d'intégrité référentielle lorsqu'à chaque valeur de la clé étrangère A correspond une valeur de la clé primaire référencée B .

Ce modèle utilise l'algèbre relationnelle, théorie de mathématiques et d'informatique, qui dit que les informations sont classées dans des relations : des [matrices](#) contenant un ensemble de [n-uplets](#) (un n -uplet est un groupe de N valeurs). Des opérations permettent à partir d'une relation d'obtenir une autre relation ainsi que de combiner deux relations et obtenir une troisième relation.

On distingue les opérations possibles en deux catégories, selon qu'elles soient unaires ou binaires.

- Les **opérations unaires** qui mettent en œuvre un seul opérande : la projection, le « renommage » et la sélection.
- Les **opérations binaires**: la différence, la division, l'intersection, les jointures, le produit cartésien et l'union.

C'est là qu'intervient le langage SQL (Structured Query Language), langage d'interrogation et de manipulation de tous les SGBDR actuels (Oracle, PostgreSQL, MySQL, MS SQLServer, MS Access et tous les autres).

Actuellement, Oracle détient plus de la moitié du marché des SGBD avec MySQL et Oracle Database.

Le Langage SQL (cf annexe 4)

Le SQL (Structured Query Language) est un langage de définition de données, un langage de manipulation de données, et un langage de contrôle de données, pour les bases de données relationnelles.

SQL est un langage de définition de données (LDD), c'est-à-dire qu'il permet de créer des tables dans une base de données relationnelle, ainsi que d'en modifier ou en supprimer. SQL est un langage de manipulation de données (LMD), cela signifie qu'il permet de sélectionner, insérer, modifier ou supprimer des données dans une table d'une base de données relationnelle.

Il est possible avec SQL de définir des permissions au niveau des utilisateurs d'une base de données. On parle de DCL (Data Control Language). Il est aussi possible d'inclure des requêtes SQL dans un programme écrit dans un autre langage, ainsi que d'envoyer directement les requêtes SQL telles quelles au SGBD.

MySQL ou Oracle ?

Suite à ces recherches, notre choix se porterait donc vers un modèle SGBD MySQL ou Oracle car ce sont les deux modèles le plus courants sur le marché et surtout multiplateformes. Cependant, MySQL serait plus intéressant étant donné la nature de notre projet. Il est vrai qu'Oracle est un modèle puissant et très efficace pour le traitement de grosses bases de données mais il demande un coût trop élevé et serait donc un investissement trop important surtout dans le cadre d'un EIP.

C'est pourquoi nous nous dirigeons plus vers l'utilisation d'une base Mysql car ce choix serait plus pertinent dans le cadre d'AudioWire. En effet, grâce à sa facilité de mise en place, son efficacité déjà prouvée et sa place dans le monde de l'open source, cette stratégie nous permettra de tirer parti d'un modèle de données relationnel puissant et rapide d'utilisation.

8. Tests de premier niveau

Le projet AudioWire est composé de deux grandes parties : la première est l'interface du lecteur elle-même, la seconde est l'algorithme de l'intelligence artificielle (IA) qui permettra une lecture appropriée aux goûts et à l'humeur de l'utilisateur.

a. L'interface

Les premiers tests se feront sur l'interface du lecteur car celle-ci devra être rapidement fonctionnelle afin de nous laisser le temps suffisant de développer une IA performante tout en implémentant ces fonctionnalités au fur et à mesure dans les différents clients.

1. Les clients

Les fonctionnalités suivantes devront être testées de manière approfondi sur tous les clients exceptés l'interface web. Pour permettre un test tout au long de l'avancement du projet, des scripts de tests dédiés à chaque plateforme devront être réalisés dans la mesure du possible ou au moins une procédure écrite si la plateforme ne permet pas de faire un script (clients mobiles probablement).

Voici une liste non exhaustive des différents tests à effectuer :

- Gestionnaire de bibliothèque :

- Ajout / suppression d'un fichier à la bibliothèque
- Ajout d'un dossier à la bibliothèque (il faut ajouter son contenu)

- Lecture de musique :

- Lecture de tous les formats de fichiers qui devront être supportés par le lecteur

- Lecture de fichiers corrompus, vides ou ne possédant pas de métadonnées indiquant le format du fichier
- Lecture de fichiers multiples
- Mise en pause / reprise de la lecture d'une musique
- Avance rapide de la lecture d'une musique (en sélectionnant une seconde où recommencer la lecture)
- Arrêt de la lecture avant la fin (bouton "stop")
- Lecture en boucle d'une musique
- Suppression d'un fichier avant / pendant la lecture

AudioWire

- Génération automatique de listes de lecture à partir de musique ou des goûts / de l'humeur de l'utilisateur
- Gestion des listes de lecture :
 - Création / suppression d'une playlist (tester la situation de 0 liste de lecture)
 - Ajout / suppression d'un élément dans une liste de lecture
 - Lecture d'une liste de lecture complète
- Interaction avec un compte utilisateur:
 - Connexion à un compte utilisateur
 - Création de compte
 - Déconnexion
 - Ajout, modification et suppression des informations relatives au compte utilisateur
- Chat et liste d'amis (fonctionnalité disponible uniquement en mode connecté) :
 - Ajout d'un ami qui existe puis un qui n'existe pas
 - Suppression d'un ami (tester le cas de 0 amis)
 - Parler avec un ami connecté puis déconnecté (si l'ami est déconnecté, il faut sauvegarder les messages dans la base de donnée pour lui délivrer à sa prochaine connexion)
- Visualizer :
 - Affichage du visualizer pendant une musique
 - Gestion de la pause et du déplacement rapide dans la musique
- Lecture synchronisée :
 - Découverte des autres clients sur le réseau local
 - Lecture d'une musique sur les appareils synchronisés
 - Déconnexion d'un client et de la source pendant la lecture
 - Tester la synchronisation des lecteurs sources et client (il faut ajuster le délai de diffusion en fonction de la latence du réseau et éventuellement l'empêcher si celui-ci est trop important).

AudioWire

- Ecoute de radio et podcasts :
 - Lister les radios / podcasts
 - Lecture d'une radio et d'un podcast
 - Déconnexion d'internet pendant la lecture
 - Ajout / suppression d'une radio/d'un podcast dans les favoris (en mode connecté uniquement)
- Partage des goûts avec les amis (disponible en mode connecté uniquement) :
 - Partager une musique / liste de lecture avec un ami

Pour toutes ces fonctionnalités, il nous faudra vérifier leur stabilité mais également surveiller leur consommation en ressources (consommation mémoire, consommation CPU, ...).

Les fonctionnalités de streaming audio nécessiteront la réalisation d'un protocole performant afin de permettre d'obtenir des latences les plus faibles possible. Des tests de performance devront être organisés pour bien gérer cette partie du projet.

2. Interface web

Elle devra permettre de bénéficier des fonctionnalités sociales, faire la promotion du produit et en permettre son téléchargement. Comme pour le client, un script de test devra être réalisé afin de tester les fonctionnalités du site mais aussi une procédure qui devra décrire comment doit se comporter l'interface graphique.

Les fonctionnalités qui devront être testées sur le site sont les suivantes :

- Téléchargement des différents clients
 - Interaction avec un compte utilisateur:
 - Connexion
 - Création de compte
 - Déconnexion
 - Ajout, modification et suppression des informations relatives au compte utilisateur
 - Chat et liste d'amis (fonctionnalité disponible uniquement en mode connecté)
- :

- Ajout d'un ami qui existe puis un qui n'existe pas
- Suppression d'un ami (tester le cas de 0 ami)
- Parler avec un ami connecté puis déconnecté (si l'ami est déconnecté, il faut sauvegarder les messages dans la base de données pour lui délivrer à sa prochaine connexion)
- Partage des goûts avec les amis (disponible en mode connecté uniquement) :
 - Partager une musique / liste de lecture avec un ami

b. L'intelligence artificielle (IA)

La seconde partie des tests sera liée à l'algorithme d'IA. Tout au long de l'EIP, nous devons vérifier la viabilité de nos choix techniques auprès des membres du laboratoire ACSEL. Comme un algorithme d'IA peut nécessiter une phase dite « d'entraînement » qui permet de le rendre efficace, nos tests seront réalisés en même temps par le biais de petits programmes/scripts. Le développement d'une IA se déroule en plusieurs étapes, de plus en plus poussées.

Pour être certain d'avoir une IA performante, il faudra tout d'abord lui faire "écouter" quelques musiques de styles très différents et vérifier qu'elle arrive à les différencier.

Une fois cette étape passée, nous pourrons commencer la deuxième phase qui consistera à ce que chacun des développeurs du projet utilise autant que possible le client AudioWire pour écouter sa propre musique. Ce qui permettra d'apprendre beaucoup de styles musicaux différents et donc être plus intelligente.

Une fois que nous obtiendrons des résultats satisfaisant, il sera temps de passer à la phase de tests suivante qui concerne la fonctionnalité de création intelligente de liste de lecture. Il faudra tout d'abord tester la fonctionnalité de complétion de liste de lecture à partir de ce qui est déjà dans la liste. Ensuite, il faudra tester la fonctionnalité de suggestion de musique en fonction de l'humeur de l'utilisateur.

Une fois que toutes ces phases seront terminées et le résultat jugé satisfaisant, nous devrions avoir terminé notre projet et pourrons passer aux tests de charges des serveurs et les adapter à nos besoins.

AudioWire

Il faudra alors réaliser un ou plusieurs scripts à cette occasion afin d'émuler l'activité de nombreux utilisateurs. Concernant le site au moins, des outils existent pour réaliser ces tests.

9. Organisation du projet

Tout d'abord nous allons établir deux états de l'art qui nous permettront de réaliser la partie intelligence artificielle de notre logiciel. Ceux-ci présenteront le Data Mining (à quoi il sert, ses implémentations, etc.) et l'analyse de fichier Audio (Que peut-on récupérer ou déduire à partir d'un signal audio, etc.) Cette partie, en partenariat avec le laboratoire ACSEL, durera jusqu'aux départs à l'étranger des différents membres du groupe. Nous commencerons à ce moment la partie développement du projet.

En ce qui concerne la réalisation du projet, nous comptons, dans un premier temps, réaliser une application pour une des plateformes non mobile, Windows.

Cette stratégie va nous permettre de construire une application stable que nous allons pouvoir par la suite distribuer à nos proches. En effet, la majeure partie de notre projet se base sur l'intelligence artificielle, et plus précisément sur du « Data Mining ». C'est ceci qui va nous permettre d'établir quels sont les goûts des utilisateurs, et qui va nous permettre de leur proposer d'autres morceaux.

Il est donc primordial que nous récupérions les informations des utilisateurs afin de tester notre algorithme et de proposer des résultats optimisés et mieux adaptés aux utilisateurs.

La plateforme Windows s'impose presque d'elle-même pour cette première version de notre logiciel puisqu'elle reste globalement la plateforme la plus utilisée. En choisissant celle-ci, nous nous assurons que nous allons pouvoir faire circuler la version « Beta » de notre logiciel à de nombreuses personnes, et donc de récupérer beaucoup d'informations.

Cette période de « beta » nous permettra non seulement de récupérer des informations sur les goûts des utilisateurs, mais également de trouver les quelques bugs de notre logiciel afin de pouvoir les régler pour la version finale.

Une fois notre version Windows stable, nous entamerons la deuxième étape qui consistera à réaliser d'un côté les versions osX et Linux et de l'autre les versions Android et iOS pour les plateformes mobiles. Nous répartirons le groupe sur ces quatre tâches afin d'optimiser le temps de travail.

AudioWire

Tout comme pour la plateforme Windows, ces plateformes mobiles et non-mobiles nécessitent également des phases de tests intensifs, que ce soit par les membres du groupe, ou par des gens extérieurs au groupe.

10. Annexes

Sources concernant les bases de données (cf chapitre 5) :

Annexe 1 :

http://fr.wikipedia.org/wiki/Base_de_donnees_relationnelle

La page de Wikipédia sur les bases de données relationnelles nous permet d'établir non seulement une rapide définition du concept mais aussi de mieux comprendre son fonctionnement et son rôle dans un projet informatique. On peut donc en déduire assez facilement l'importance d'une telle structure ainsi que pourquoi elle occupe une part importante dans le cadre d'AudioWire.

Annexe 2:

http://fr.wikipedia.org/wiki/Système_de_gestion_de_base_de_donnees

Ce lien sur la gestion de base de données nous apporte une connaissance supplémentaire sur le moyen de communication entre une base de données relationnelle et le programme en lui-même.

Annexe 3:

<http://webtic.free.fr/sql/>

Ce document explique le fonctionnement et la manière de mettre en place un modèle de données relationnelles. Il nous renseigne notamment sur la complexité de concevoir la structure d'une base de données et l'importance de cette dernière. Cela démontre parfaitement l'utilité de bien réfléchir sur la conception d'une base de données solide et stable.

Annexe 4:

<http://www.commentcamarche.net/contents/sql/sqlintro.php3>

Le document explique la signification ainsi que les fonctionnalités du langage SQL. Il définit rapidement comment on peut l'utiliser et dans quels cas. On peut ajouter que le langage SQL est un langage très utilisé dans le monde de l'informatique, et plus particulièrement dans celui des bases de données, notamment grâce à son système de requêtes qui le rend assez simple à comprendre. Notons qu'il existe un concurrent direct, QBE (Query By Example),

AudioWire

qui est un peu plus instinctif à manipuler (sous forme de matrice de réponse) mais plus lourd dans la manière de rédiger la requête.

Weka : et non, ce n'est pas le Wikipédia chinois !

Annexe 5:

http://en.wikipedia.org/wiki/Weka_%28machine_learning%29

Cette page Wikipédia nous explique l'origine de Weka, une suite de logiciels d'apprentissage automatique développé à l'université de Waikato en Nouvelle-Zélande. L'ensemble des outils proposés par Weka nous permet de faire une analyse des données et ainsi d'en faire une modélisation. Les principaux avantages de Weka sont qu'il est libre, portable car développé en Java, et simple d'utilisation grâce à son interface graphique.

Annexe 6:

www.sacem.fr/files/content/sites/fr/files/mediatheque/sacem/presse/etudes/etude_sacem/sondage_francais_musique/sondage_les_francais_et_la_musique_2011.pdf

Voir Diagramme de Gantt.

11. Conclusion

AudioWire est un projet qui sera donc utilisable par une population diverse et variée du moment qu'elle est un tant soit peu amateur de musique. Grace à son développement multi plateforme ainsi qu'à ses applications pour téléphone mobile, on pourra prétendre séduire tous les types d'utilisateurs, quel que soit leur âge. Ainsi, AudioWire intégrant le concept puissant d'algorithme d'intelligence artificielle, dans un registre qui fait partie de la vie de tous les jours, la musique, pourra un jour permettre une ouverture sur ce type de technologie, au quotidien, dans les années à venir.