# AudioWire

Hugo Defrance, Huai Xie, Stéphane Rose

Guillaume Feraud, Augustin Pasquini,

Vivien Meilhac, Maximilien Brunelle



Bilan d'architecture
1 juillet 2012

*Résumé du document*

Ce bilan d'architecture donne une description conceptuelle du projet AudioWire.

Après avoir brièvement présenté AudioWire, ce document donnera plus de détails sur les différentes fonctionnalités du projet. Il apportera ensuite les différentes solutions qui seront présentées ainsi que l'architecture du projet sous forme de diagrammes UML. Pour conclure ce bilan d'architecture, nous ciblerons les différentes contraintes fonctionnelles et techniques du logiciel afin d'y apporter des solutions.

## Description du document

| | |
|---|---|
| Titre | AudioWire – Bilan d'architecture |
| 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 | Bilan d'architecture |
| Mots-clés | |
| Version | 1.1 |

## Tableau des révisions :

| Date | Auteurs | Section(s) | Commentaire |
|---|---|---|---|
| 01/07/12 | L'ensemble du groupe | Toutes | Première version du bilan d'architecture |
| 13/07/12 | L'ensemble du groupe | Toutes | Version Finale du bilan d'architecture |
| | | | |

# Table des matières

# 1. Rappel de l'EIP

## Epitech, l'école de l'innovation et de la passion informatique

L'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 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 ou l'on amène la connaissance à l'élève afin qu'il puisse résoudre des problèmes, un étudiant à l'Epitech devra rechercher de lui-même les notions dont il a besoin avec son groupe de travail pour résoudre un problème ambiguë dans lequel il n'aura pas à l'origine les bases nécessaires. C'est de ce principe que naitra l'émergence du comportement que recherche l'Epitech, à savoir la capacité d'adaptabilité par rapport à une problématique et des concepts ainsi que la gestion d'une équipe de travail.

## 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 car 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 confirmé. 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.

## 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 l'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. A propos d'AudioWire

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 forcement 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. Vue d'ensemble

AudioWire sera développé à la fois pour les ordinateurs de bureau, que pour les téléphones mobiles. Elle comprendra également une application web qui permettra non seulement de bénéficier des fonctionnalités sociales d'AudioWire mais aussi de faire la promotion du logiciel. Voici ci-dessous les diagrammes d'utilisations associés pour ces trois types d'architecture.

# Description global du projet

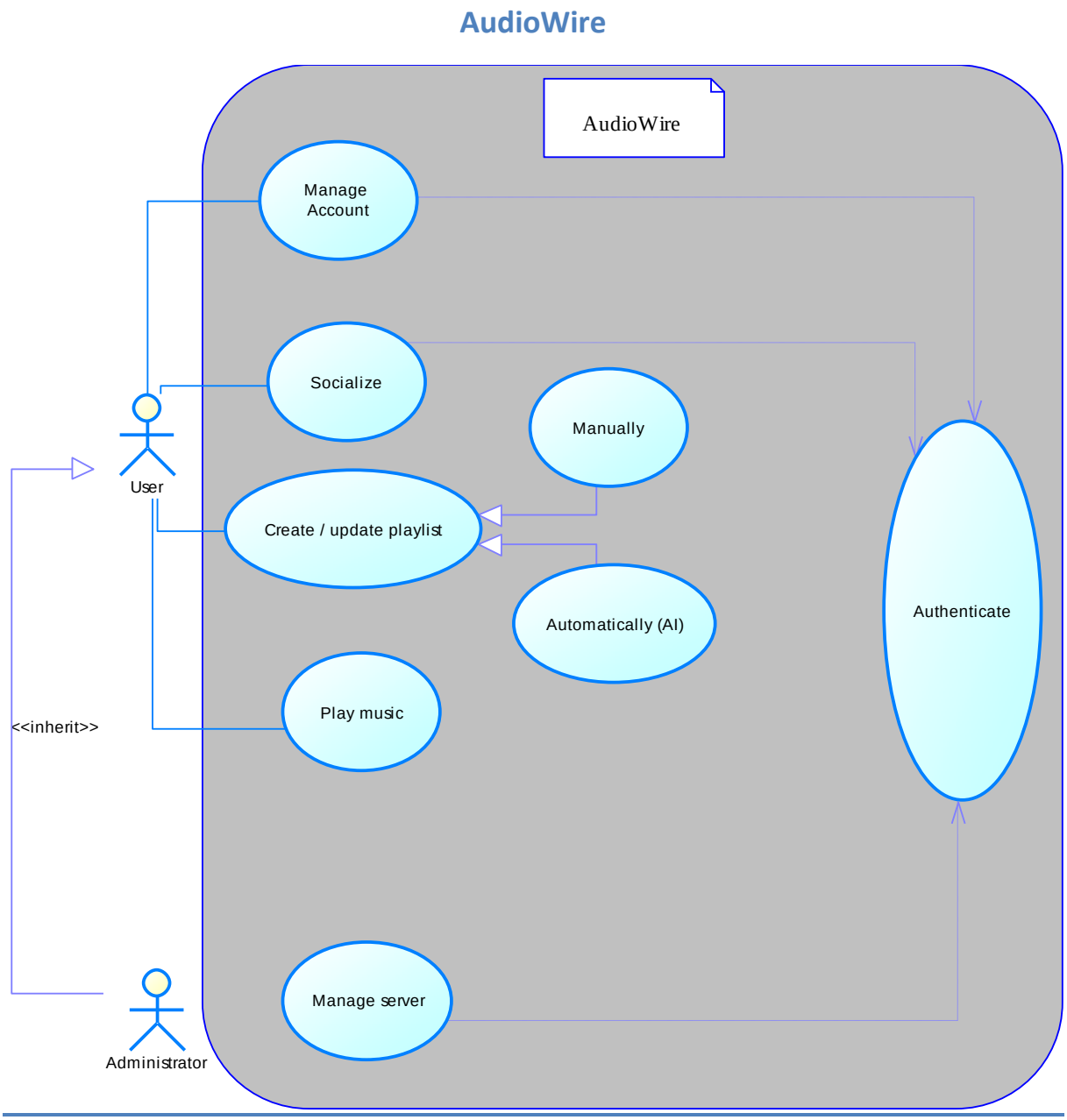



*Diagramme 1 : Description d'AudioWire*

## L’application mobile

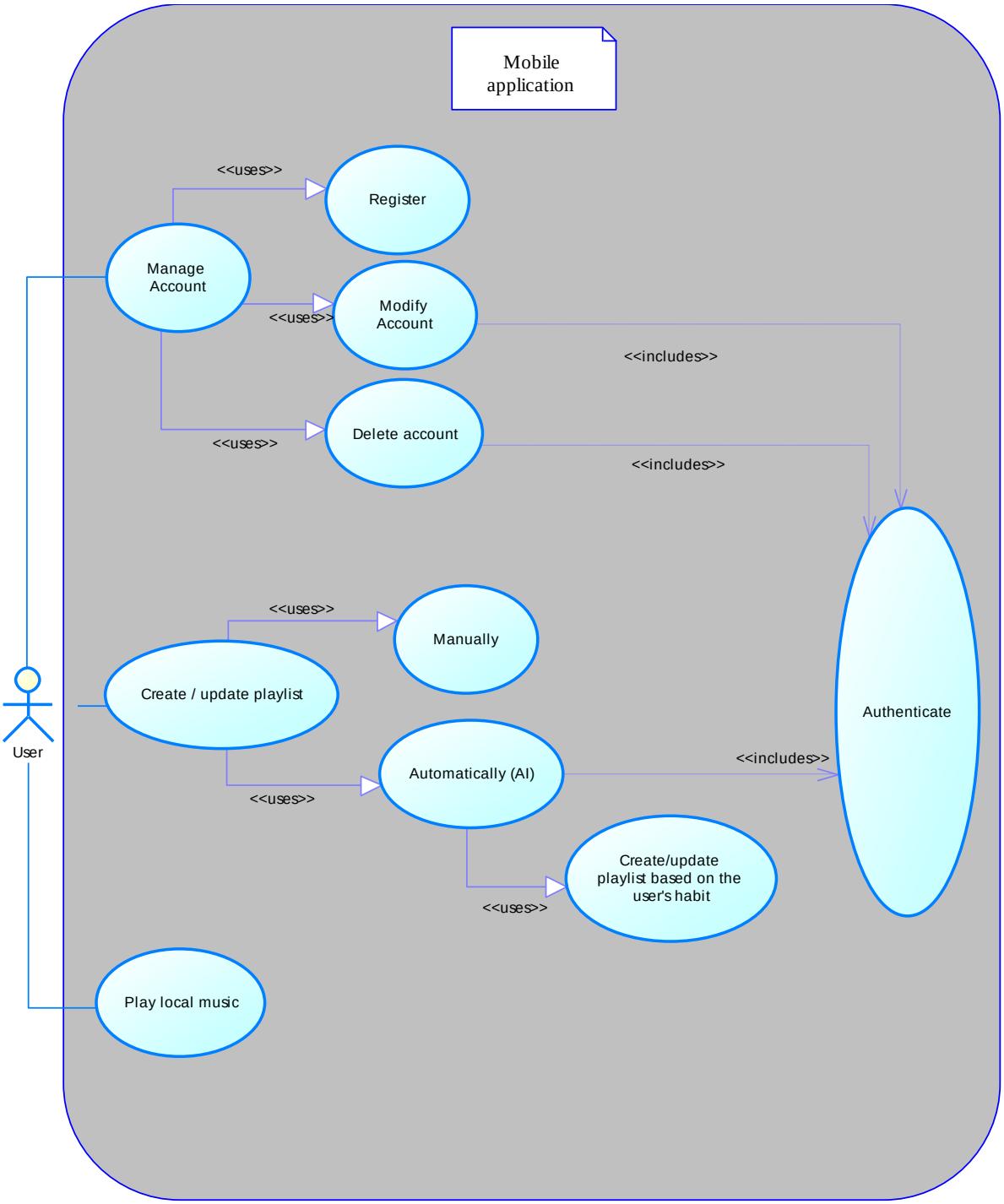



*Diagramme 2 : Description de l'application mobile*

## L'application de bureau

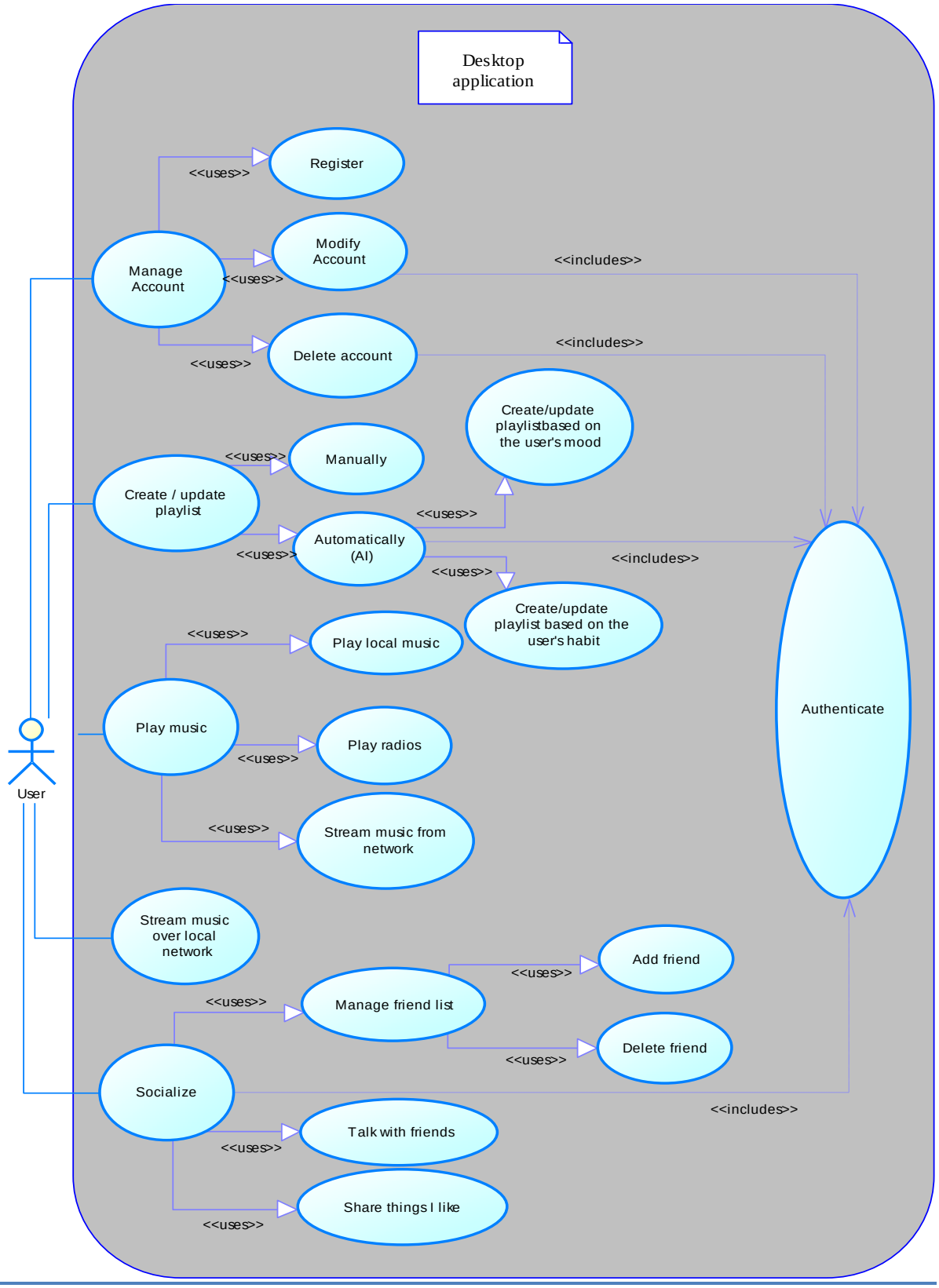



*Diagramme 3 : Description de l'application de bureau*

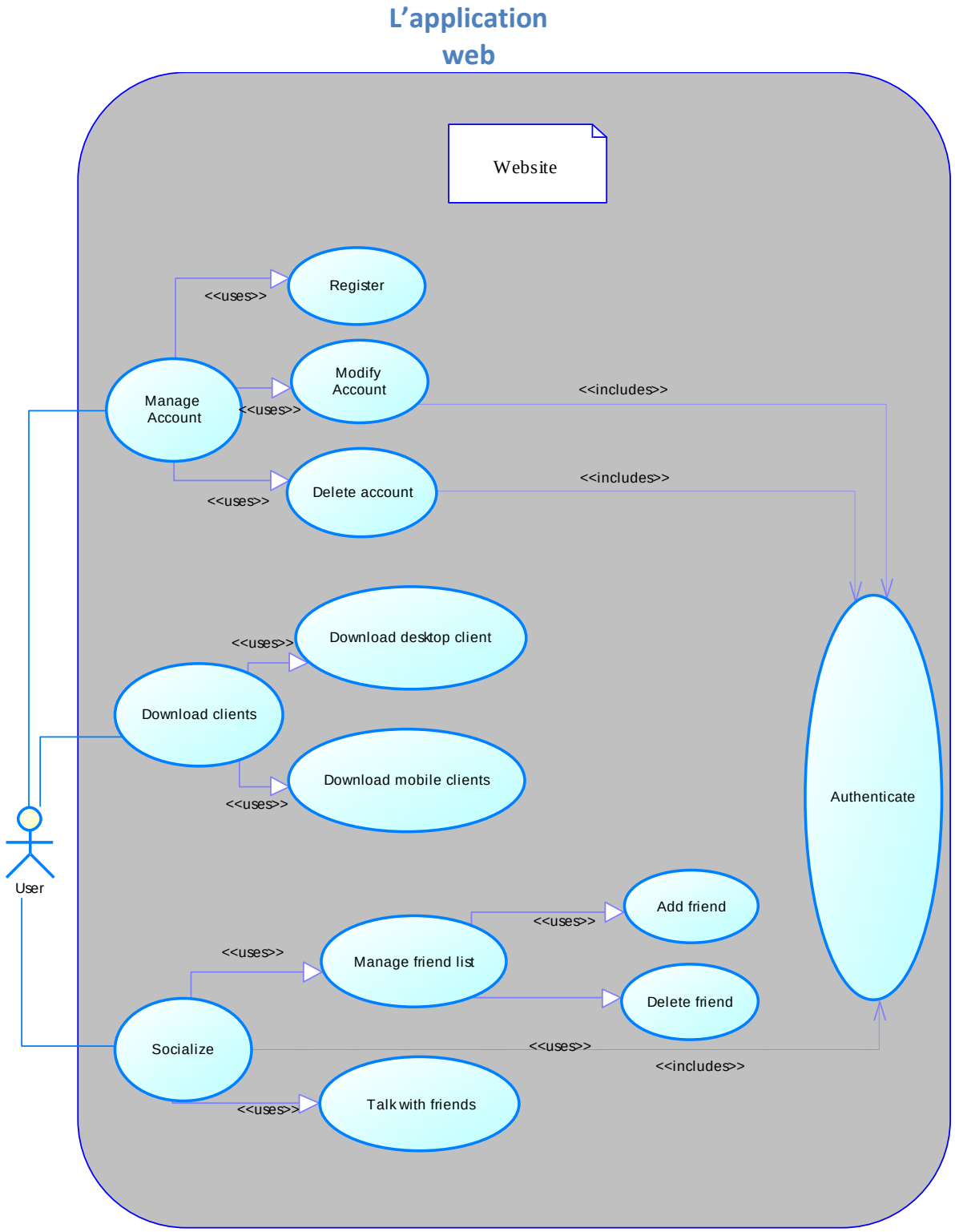



*Diagramme 4 : Description globale du fonctionnement du site internet*

# 4. Architecture du projet

## L'architecture du client web

Une application Django n'étant pas un serveur web, nous aurons besoin d'en utiliser un tel que Nginx pour servir l'application. uWSGI sera lui chargé d'assurer la liaison entre ces deux programmes.

De plus, uWSGI permettra de facilement réguler la charge du serveur afin de s'assurer que celui-ci soit toujours capable d'accepter de nouveaux clients.

L'application Django bénéficiera d'un accès direct à la base de données d'AudioWire.

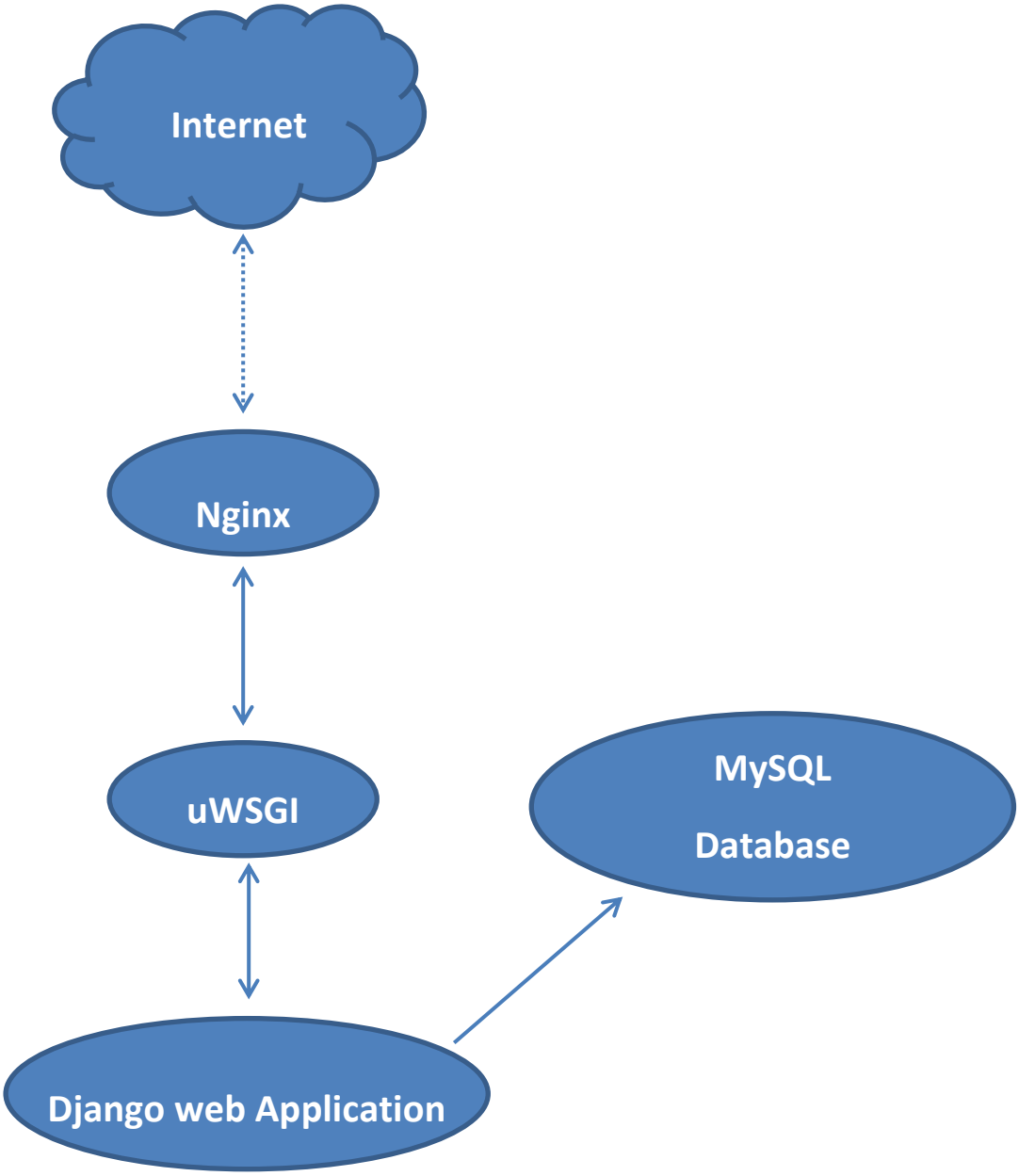



*Diagramme 5 : Architecture du client web*

## Les différentes fonctionnalités du client

Ci-dessous les différentes fonctionnalités du client, les différents types d'action que l'utilisateur peut effectuer pour gérer son profil et sa musique.

Il y aura également un chat, pour que les utilisateurs puissent discuter entre eux, ou partager leur musique tout simplement.

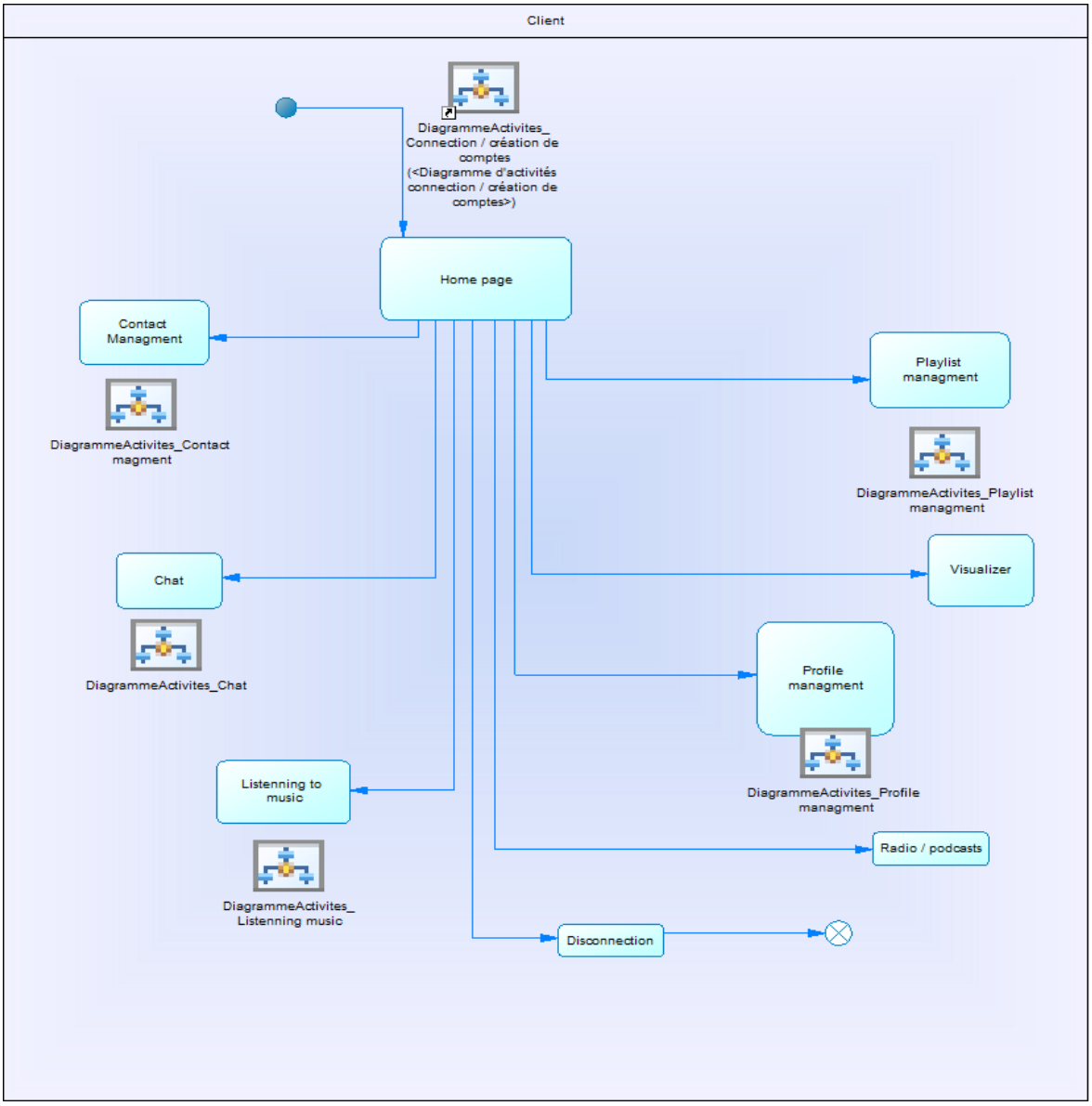



*Diagramme 6 : Les fonctionnalités du client*

## La connexion / création de compte

Ceci est le diagramme de séquence qui décrit les étapes parcourues pour l'authentification d'un utilisateur lors de sa connexion. Chaque utilisateur doit être enregistré préalablement pour bénéficier pleinement de l'application.

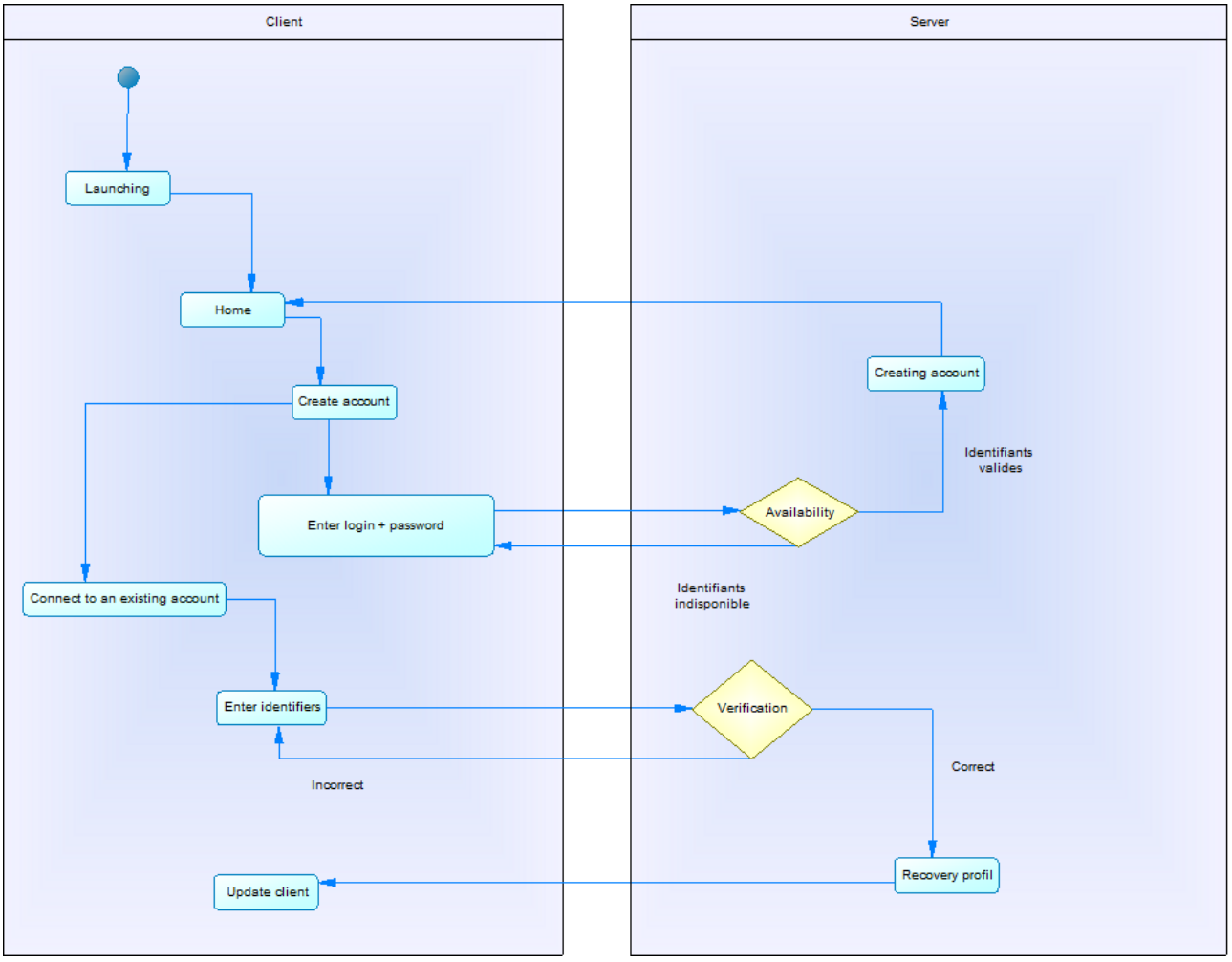



*Diagramme 7 : diagramme d'activité sur la connexion /création de compte du client*

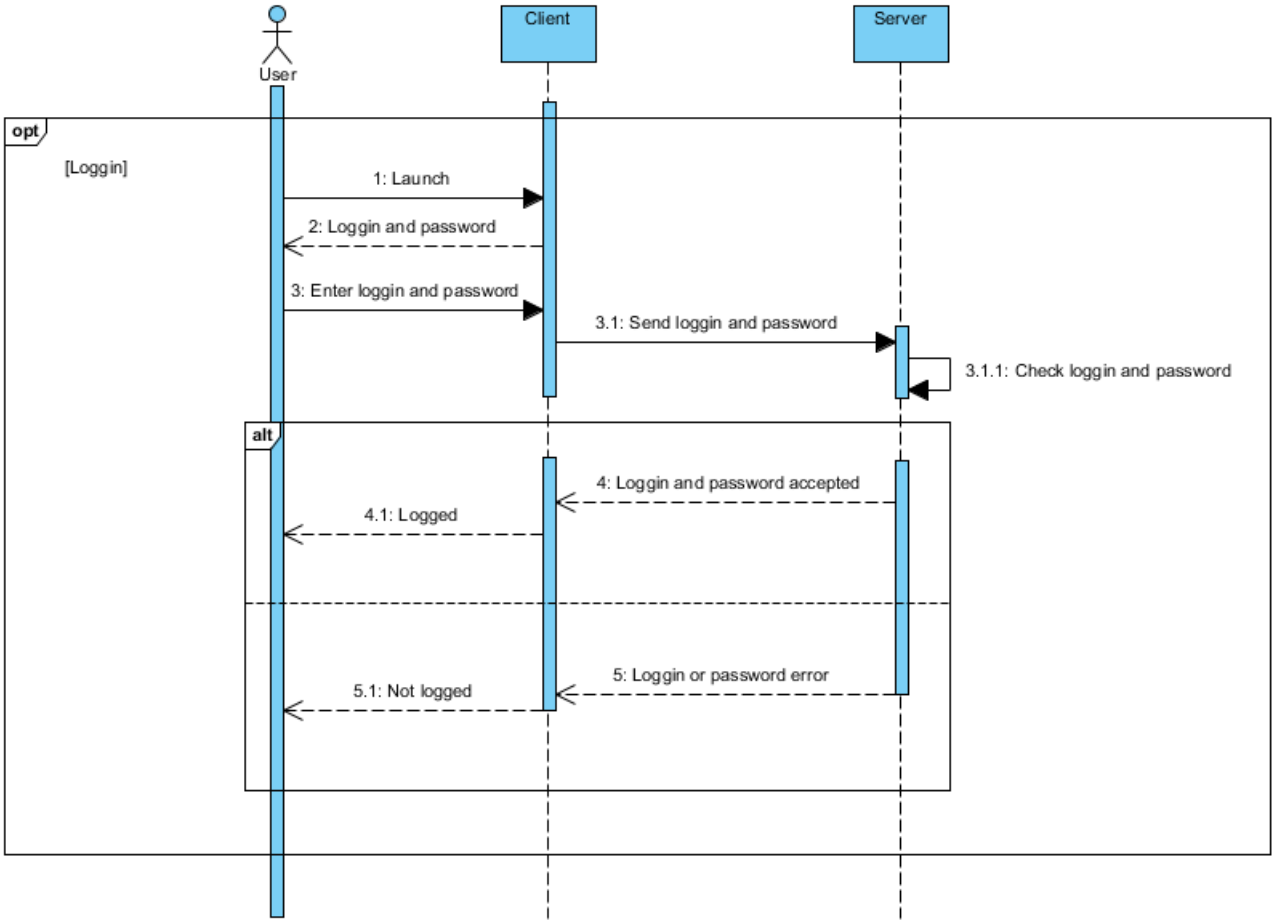



*Diagramme 8 : Diagramme de séquence sur le protocole de connexion du client*

## La gestion de contacts

Voici le diagramme de séquence décrivant les étapes pour ajouter un contact. Il faut savoir que l'utilisateur doit être lui-même enregistré dans la base pour pouvoir ajouter un nouveau contact dans sa liste.

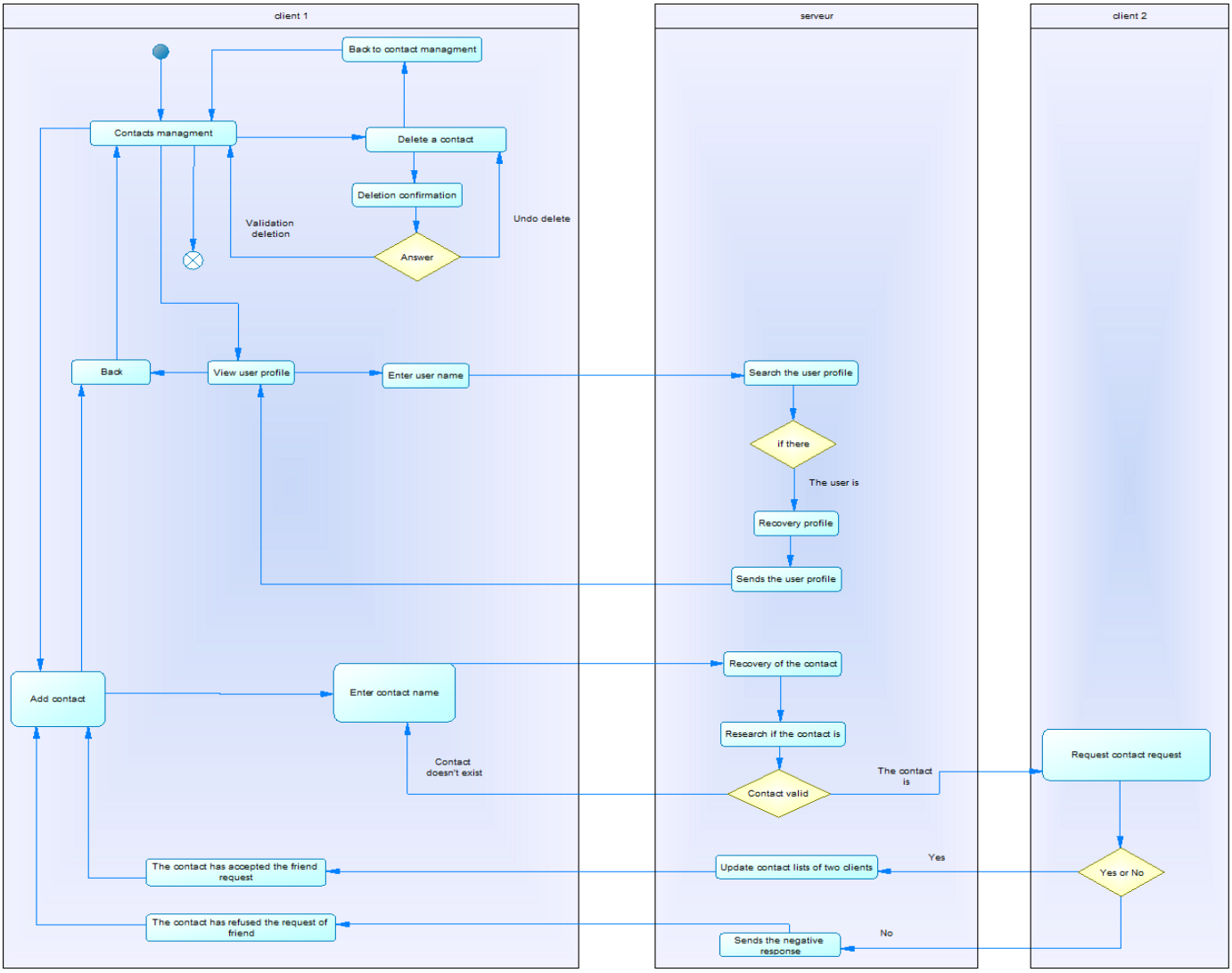



*Diagramme 9 : La gestion des contacts*

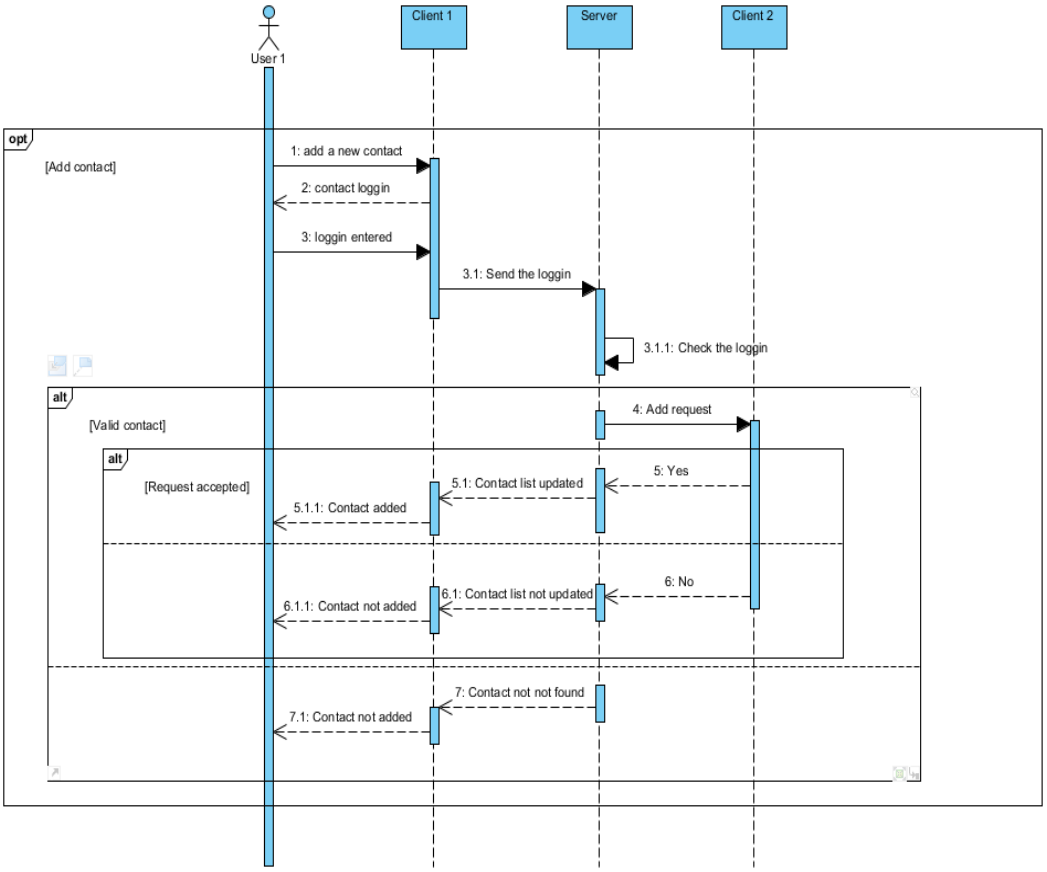



*Diagramme 10 : Diagramme de séquences sur la gestion des contacts*

## L'écoute de la musique

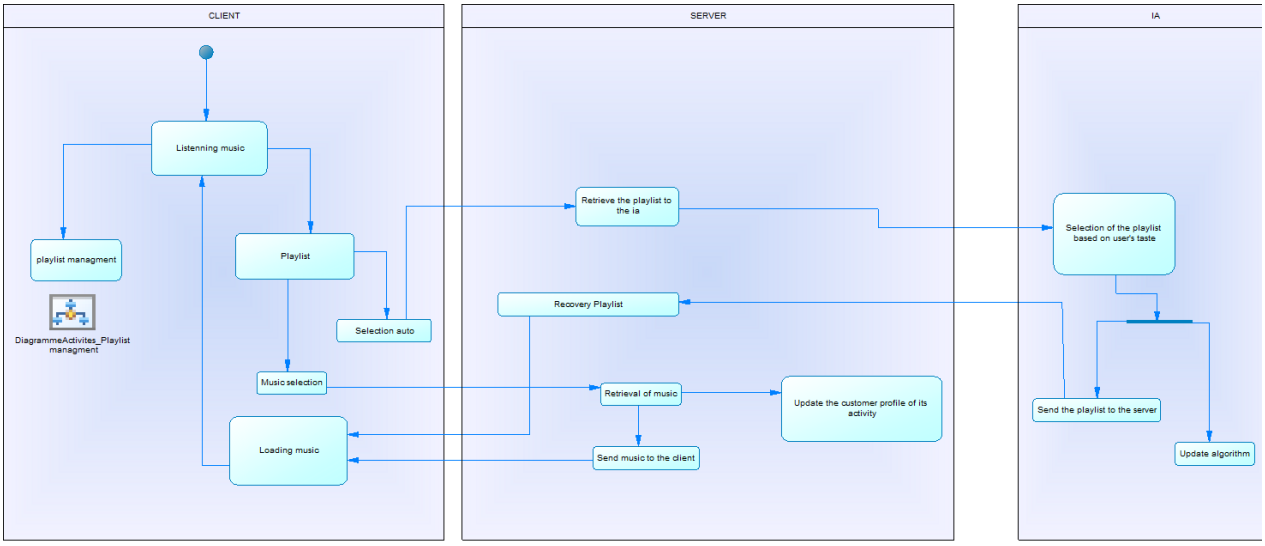



*Diagramme 11 : Diagramme d'activité sur le système d'écoute de musique*

## La messagerie

Le message envoyé peut être reçu par le destinataire même s'il n'était pas connecté lors de l'envoie.

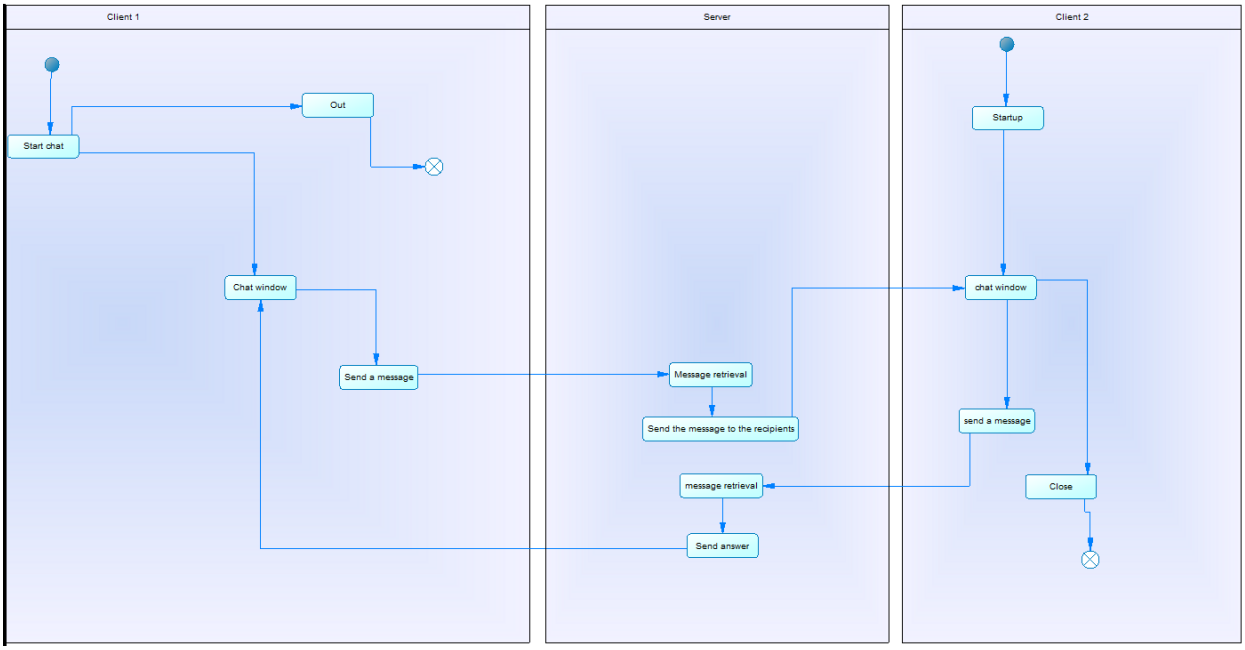



*Diagramme 12 : Diagramme sur le fonctionnement de la messagerie*

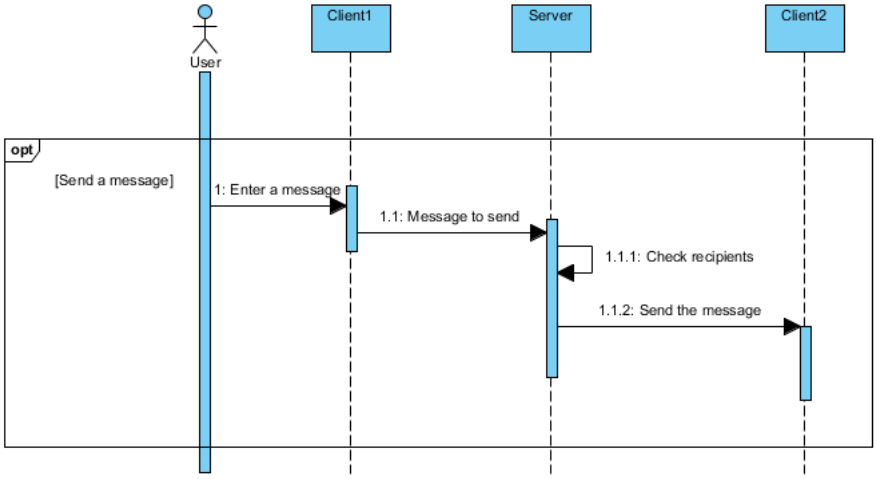



*Diagramme 13 : Description de la messagerie dans un diagramme de séquence*

## La gestion de profil

On peut à tout moment changer ses informations personnelles qui seront changer également sur le serveur, et choisir de les mettre en publique ou non.

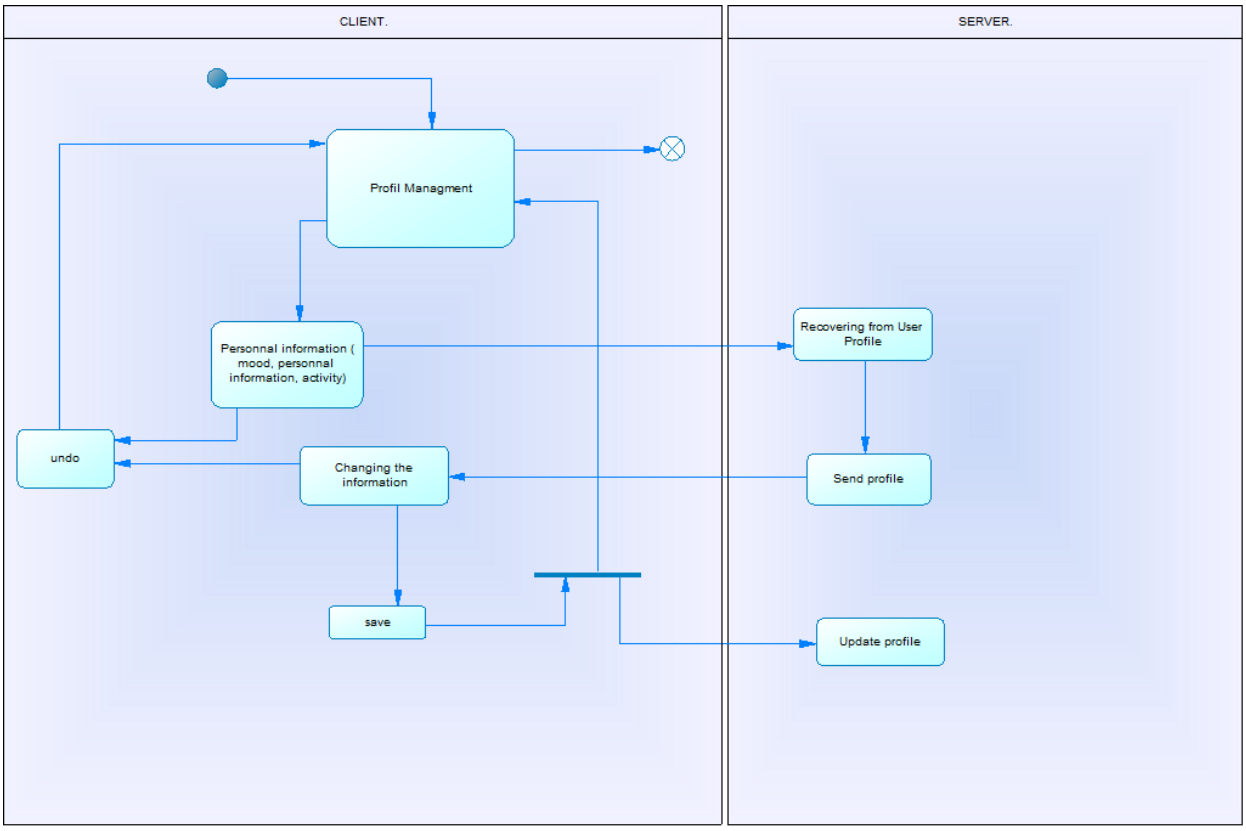



*Diagramme 14 : Diagramme d'activité sur la gestion de profil utilisateur*

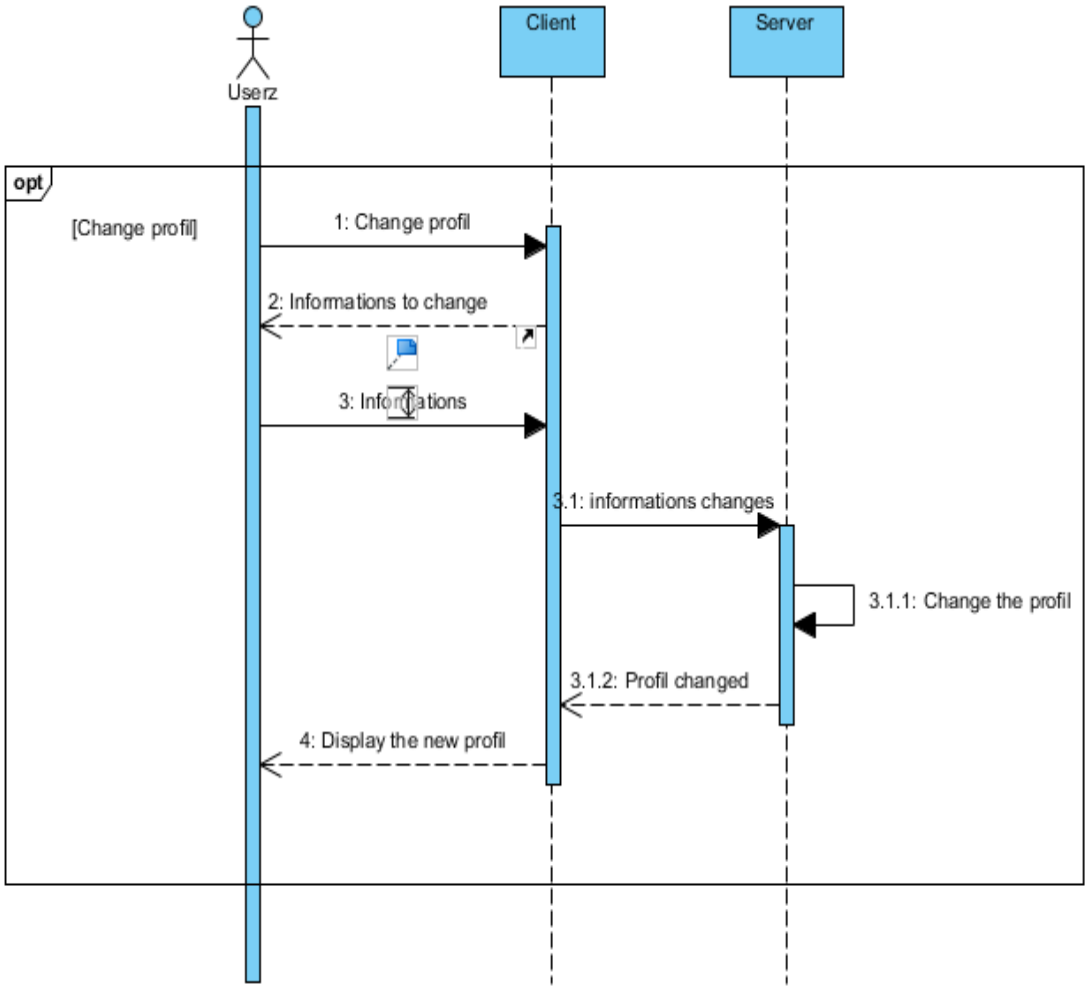



*Diagramme 15 : Gestion de profil*

Un utilisateur authentifié peut à tout moment consulter le profil d'un autre utilisateur, si celui-ci a choisi de mettre en publique ses informations.

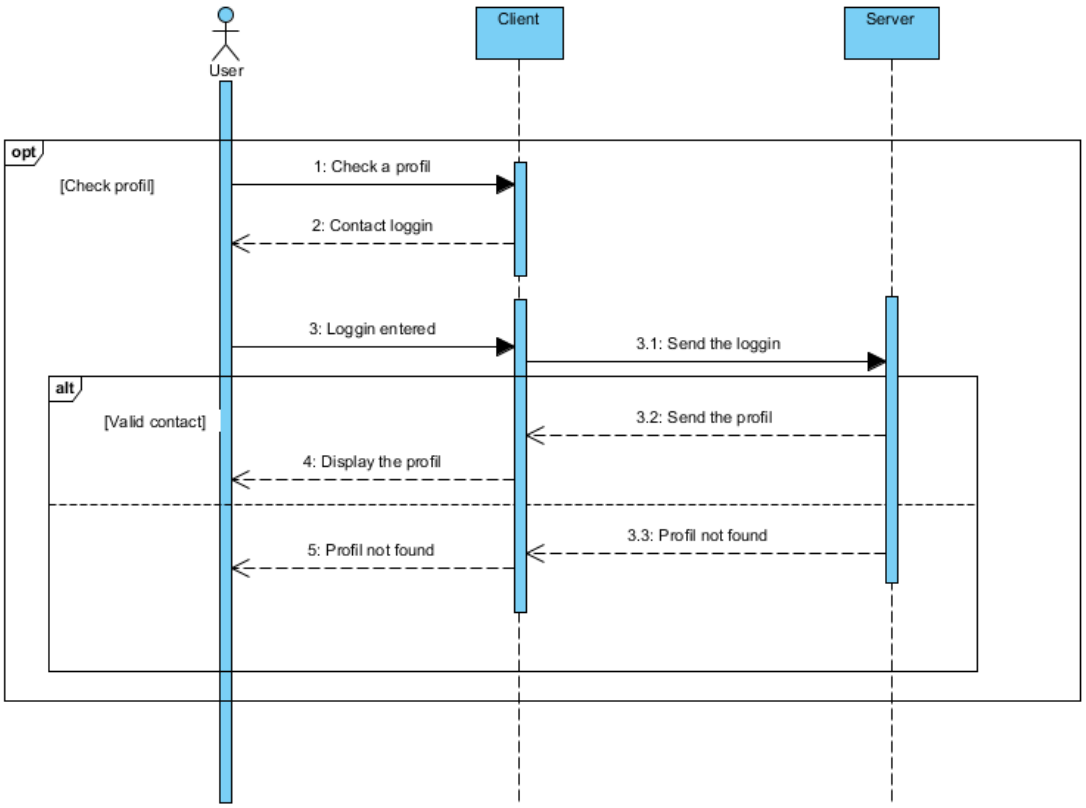



*Diagramme 16 : Diagramme de séquence sur la sélection d'un profil*

## Gestion des playlists

On peut créer une Playlist du moment où l'utilisateur s'est connecté, et que le nom de sa Playlist n'existe pas déjà.

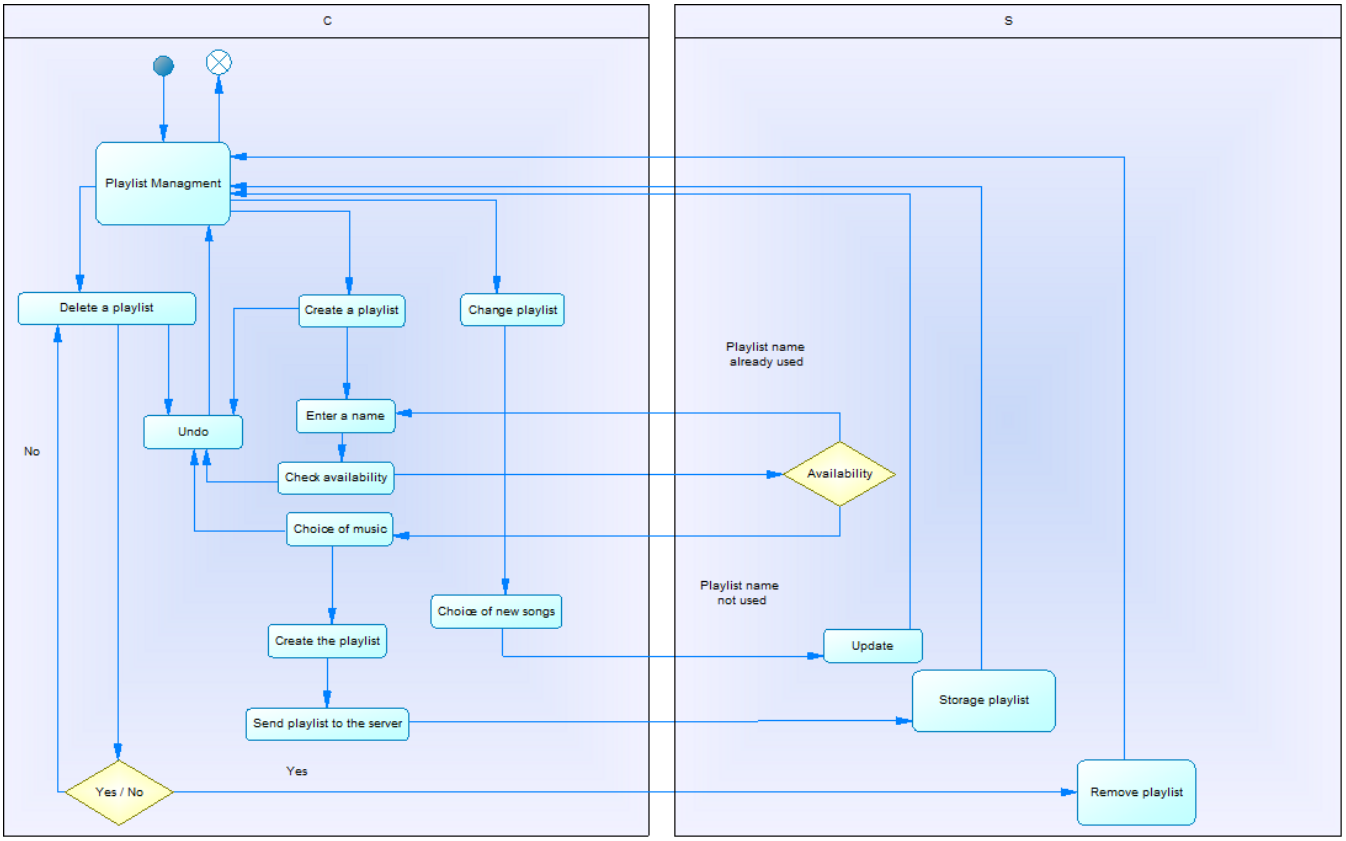



*Diagramme 17 : Diagramme d'activité sur la gestion des playlists*

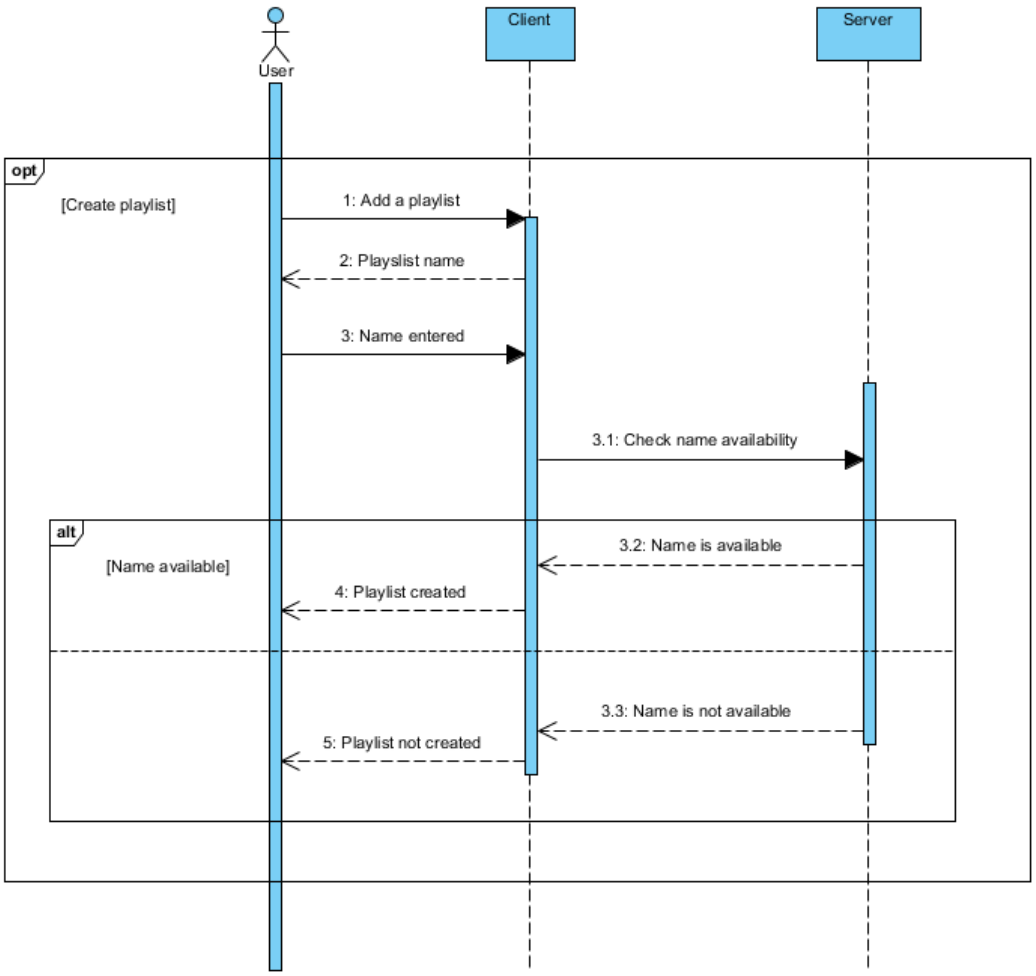



*Diagramme 18 : Diagramme de séquence sur la gestion des playlists*

## Le core du client

Voici le diagramme correspondant aux clients des plateformes Windows, Mac et Linux. Sachant que toutes les actions de l'utilisateur passent directement ou indirectement par le core, il est une des parties les plus importantes de notre projet.

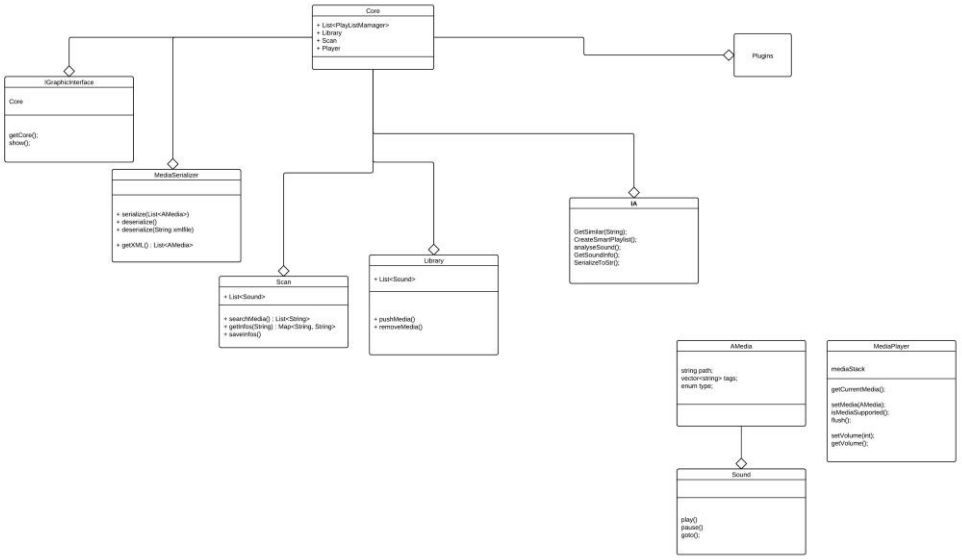
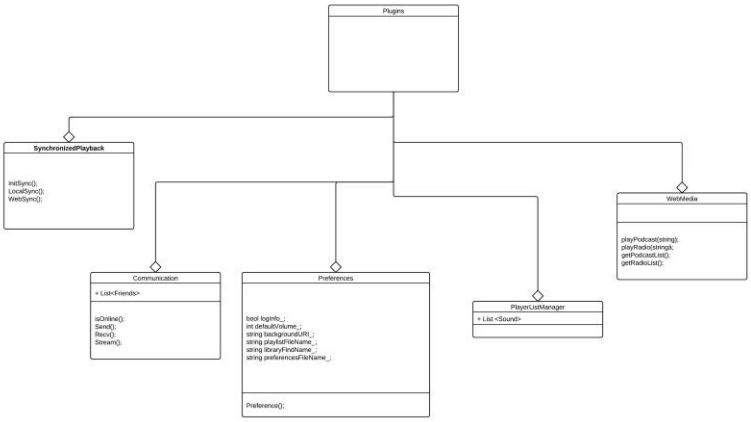



*Diagramme 19 : Diagramme de classes sur le core du serveur*


Plugins
SynchronizedPlayback
InitSync();
LocalSync();
WebSync();
WebMedia
playPodcast(string);
playRadio(stirngs;
getPodcastList();
getRadioList();
Communication
+ List<Friends>
isOnline();
Send();
Recv();
Stream();
Preferences
bool logInfo_;
int defaultVolume_;
string backgroundURI_;
string playlistFileName_;
string libraryFindname_;
string preferenceaFileName_;
Preference();
PlayerListManager
+ List <Sound>
Diagramme 20 : Suite du core

## Vue déploiement

Voilà le diagramme de déploiement, qui représente les liens et les relations entre les infrastructures physiques et les composants du système de notre projet.

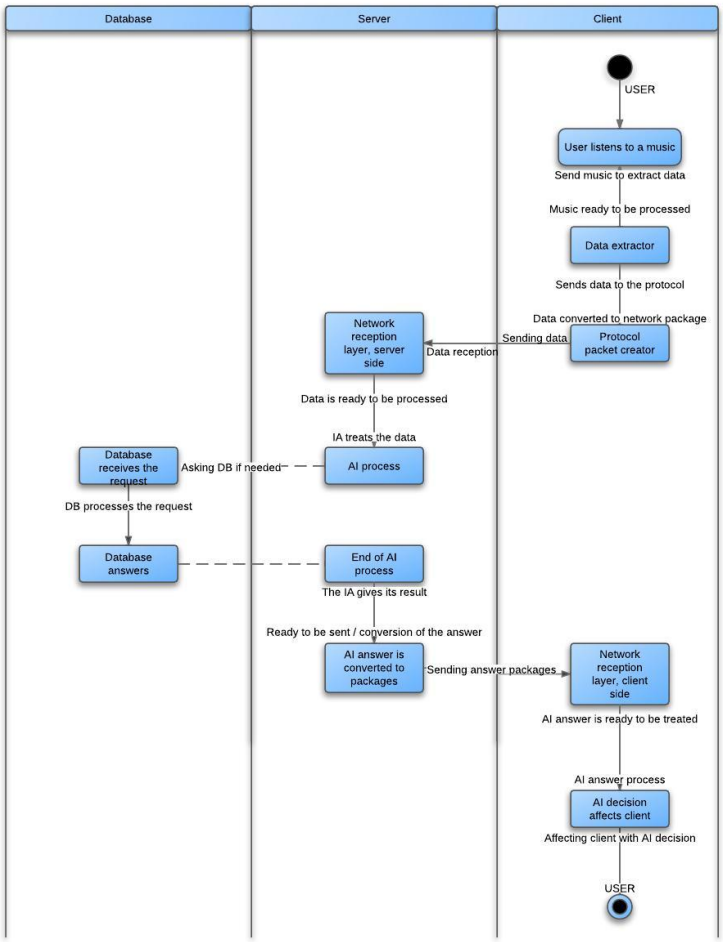



*Diagramme 21 : Vue d'ensemble*

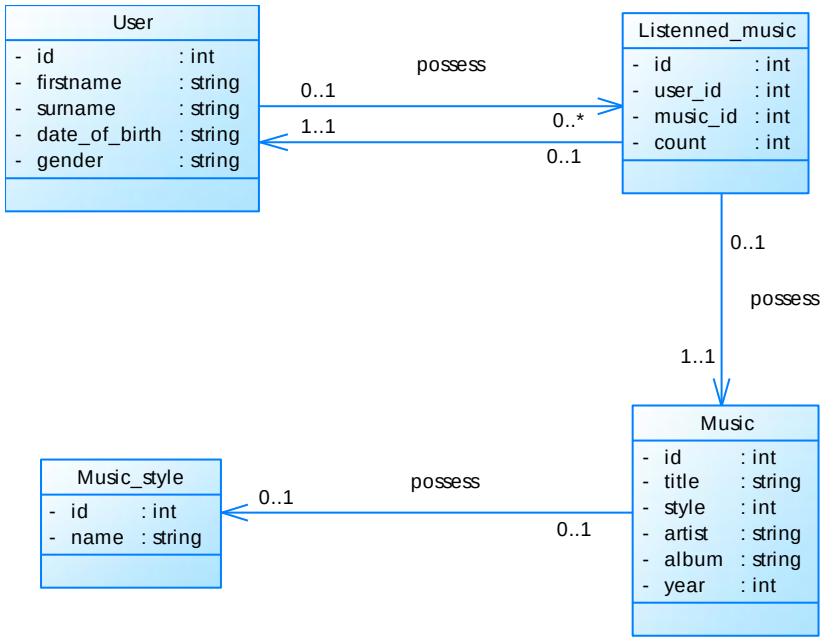
# 5. Base de données

Chaque utilisateur de notre application devra posséder un compte pour pouvoir bénéficier des fonctionnalités avancées de notre service. Ces informations seront donc conservées dans la table User.

Afin de permettre à l'intelligence artificielle de générer des listes de lectures, nous aurons également besoin de stocker les musiques écoutées par les utilisateurs. Nous utiliserons la table Listenned_music à cet effet.

Nous aurons également besoin de conserver de nombreuses informations sur la musique de nos utilisateurs afin de la traiter dans l'intelligence artificielle. Nous utiliserons la table Music à cet effet.

Enfin, toujours pour les besoins de fonctionnement de l'intelligence artificielle, nous aurons également besoin  d'une table Music_style.




*Représentation de l'architecture de la base de données*

# 6. Contraintes fonctionnelle

## a) Synchronisation

Pour la partie synchronisée de notre logicielle nous allons devoir faire face à plusieurs problèmes.
Premièrement nous allons devoir trouver une façon efficace de synchroniser les lectures. Pour cela nous pensons synchroniser en fonction d'une horloge commune. Si cette option n'est pas bonne, nous tenterons de calculer le temps de latence entre les deux postes et de se baser sur ce temps pour lancer les lectures au même moment.
Nous devrons également utiliser ce temps pour poser la lecture au même moment. Il faudra donc par exemple un temps d'attente sur l'ordinateur transférant le fichier afin de s'accorder avec le temps de latence de l'autre ordinateur.

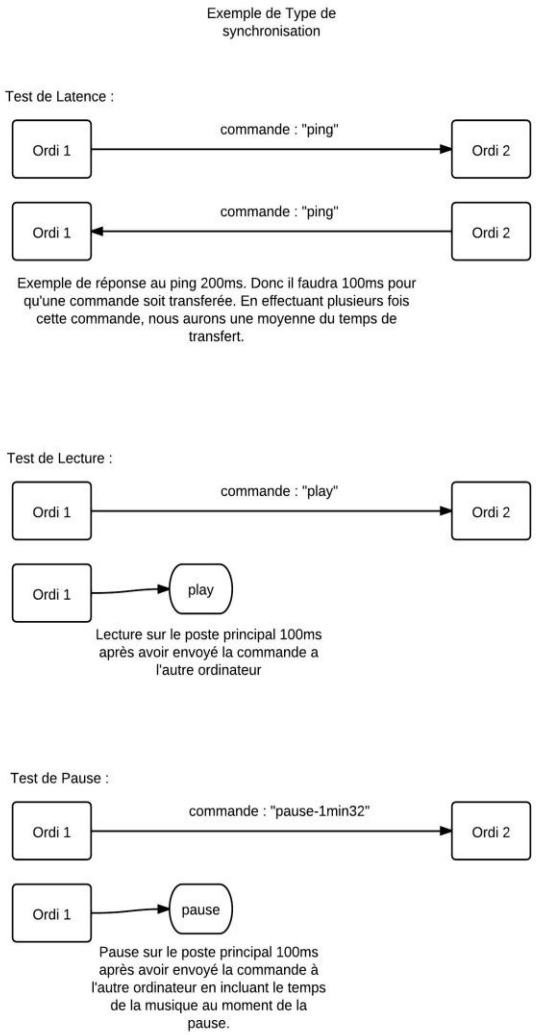



*Diagramme : Explication de la synchronisation réseau*

## b) Intelligence artificielle - qualité des résultats

La partie IA du projet doit être capable d'extraire de chaque fichier audio toutes les informations qui pourront être utiles. C'est ce traitement qui permettra à AudioWire de proposer des musiques en accord avec les gouts de l'utilisateur.

Cependant, ce module repose sur des algorithmes qui peuvent être difficiles à mettre en place, les résultats seront donc à vérifier avec précaution. Des outils de tests et des bases de connaissance nous serons donc grandement utiles.

Afin de réussir au mieux cette partie, le projet se verra aider par le laboratoire Acsel, nous pourrons ainsi profiter du savoir et de l'expérience de ses membres.

## c) Légalité du streaming inter-utilisateur

Le fait de "streamer" un fichier audio vers un autre client peut poser des problèmes au niveau juridique. En effet, ce processus nécessite que le fichier, ou du moins une partie, soit transféré sur l'ordinateur d'un autre utilisateur. Ce procédé peut donc être interprété comme un acte de piratage.
Afin d'éviter ce genre de problèmes, nous spécifierons dans le contrat de License, (à l'installation d'AudioWire) que lors d'un streaming inter-utilisateur, l'utilisateur recevant le fichier s'engage à avoir une copie du fichier en sa possession (CD, cassettes, etc.).
En notifiant ses utilisateurs de cette contrainte, AudioWire se dégagera de toutes les responsabilités lié aux droits d'auteurs.

# 7. Contraintes techniques

## a) Puissance des téléphone mobiles

La majorité des téléphones portables actuels ne peuvent rivaliser avec des ordinateurs de bureau. C'est pour cette raison que nous avons décidé que nos applications mobiles n'offriraient pas exactement les mêmes fonctionnalités que les versions bureaux. Les algorithmes d'intelligence artificielle, par exemple, ne seront pas présents sur les mobiles. Ils risqueraient de ne pas bien fonctionner ou d'utiliser trop les ressources du téléphone (mémoire, processeur,
batterie, etc.).

## b) Bande Passante

Le procédé de "streaming" implique une bonne bande passante. Il sera donc impossible, ou trop lent, dans certains cas d'utiliser l'option de streaming. Notre logiciel, AudioWire, effectuera des tests de bande passante avant d'établir des connexions inter-utilisateurs pour le streaming. Si les tests de bande passante ne s'avèrent pas suffisant, le logiciel préviendra l'utilisateur et le streaming sera indisponible.

## c) Stabilité du serveur

Au sein d'AudioWire, le serveur s'occupe de récupérer les données des utilisateurs, il stocke toutes ces informations et permet aux algorithmes d'intelligence artificielle d'établir de bons résultats. C'est pour cela que la stabilité du serveur est primordiale que ce soit au niveau hardware ou au niveau software. Les tests de premiers niveaux que nous ferons après avoir développé la partie serveur nous permettrons de tester la partie software et de nous assurer de sa stabilité. Du côté hardware, le laboratoire EIP nous a autorisés à utiliser un de leur serveur, nous pensons donc que nous n'aurons

pas de majeurs soucis pendant cette période. Nous verrons une fois le projet terminé si un investissement dans un vrai serveur sera utile.