

CAHIER DES CHARGES & SPÉCIFICATIONS

Maîtrise d'œuvre

BAR Vanessa - BLANCHARD Baptiste
DECHENEST Pierre - GIRARD Thibault
HEINE Axel

Assistant à maîtrise d'œuvre

HEINE Axel

Maîtrise d'ouvrage

LOHIER Stéphane

SOMMAIRE

Introduction.....	3
Partie 1 : Cahier des Charges.....	3
I. Analyse générale du besoin.....	3
I.1. Contexte.....	3
I.2. Expression du Besoin.....	4
I.3. Précisions sur le site.....	4
II. Formalisation.....	5
II.1. Acteurs.....	5
II.2. Définition des fonctionnalités.....	6
II.2.a. Fonctionnalités du client.....	6
II.2.b. Fonctionnalités du site web.....	6
II.3. Lots fonctionnels.....	6
II.3.a. Fonctionnalité de l'application.....	6
II.3.b. Lot fonctionnel du site.....	7
III. Architecture technique.....	8
IV. Glossaire.....	8
Partie 2 : Spécifications.....	10
I. Spécifications fonctionnelles.....	10
II. Spécifications graphiques du client.....	23
II.1. Chartes graphiques.....	23
II.1.a. Couleurs.....	23
II.1.b. Typographie.....	23
II.1.c. Icones.....	24
II.1.d. Popups.....	25
II.2. Liste des composants.....	25
II.3. Définitions des écrans.....	26
II.3.a. Windows, Linux et Mac.....	26
II.3.b. Android.....	28

INTRODUCTION

Dans le cadre de notre deuxième année de DUT Informatique, nous devons réaliser un projet tuteuré par groupes de trois à cinq personnes. Notre équipe est composée de Thibault GIRARD, Pierre DECHENEST, Baptiste BLANCHARD, Vanessa BAR et Axel HEINE, notre chef de projet. Nous avons choisi de travailler sur le projet Spotidut : un projet ambitieux de streaming musical sur Raspberry Pi.

Suite à la réunion du 10 Octobre 2014 avec M. Stéphane LOHIER, nous avons entamé la rédaction de ce document. Il reprend donc toutes les exigences du projet et les formalise. Il contient également les spécifications fonctionnelles et graphiques de Spotidut.

PARTIE 1 : CAHIER DES CHARGES

I. Analyse générale du besoin

I.1. Contexte

Avec la dématérialisation et l'essor des nouvelles technologies, on constate une disparition des supports audio. En effet, la musique est de plus en plus fréquemment stockée sur des mémoires flashs et/ou des disques durs, sous forme de simples fichiers MP3, par exemple. Cependant, cette technique peut vite devenir très lourde et causer une perte d'espace de stockage importante notamment si l'on possède plusieurs plateformes puisqu'il faut alors copier les fichiers sur chacune d'elle...

De nos jours, les outils numériques ont également pour vocation de regrouper plusieurs autres outils : avec le téléphone, par exemple, on veut pouvoir téléphoner, aller sur Internet, écouter de la musique... Ainsi, il paraît nécessaire de développer chaque application sur différentes plateformes. En ce qui nous concerne, on constate qu'il est possible d'utiliser un lecteur de musiques aussi bien sur un téléphone ou une montre que sur un lecteur MP3 ou un ordinateur.

Pour répondre à cette problématique de stockage et cette nécessité de multi-supports, des sites web et des applications voient le jour. On peut alors trouver Deezer, YouTube, ou bien Spotify. Ces différents sites permettent tous une écoute illimitée de musiques en ligne, donc, sans stockage : ce sont des sites de streaming.

Cependant, les musiques disponibles sur ce site sont stockées sur des serveurs souvent très onéreux. Spotidut a alors pour ambition de recréer un serveur de streaming sur Raspberry Pi, une plateforme simple et peu onéreuse.

I.2. Expression du Besoin

L'objectif principal est de créer un serveur local d'écoute de musiques (streaming) ainsi qu'un client permettant leur lecture.

Le serveur doit être hébergé sur une Raspberry Pi à laquelle sera connecté un disque dur contenant des musiques. Il doit permettre le stockage des données (musicales et de connexion) et leur envoi. Son état doit être visible simplement grâce à la présence de LEDs indiquant trois états : serveur en fonctionnement, serveur en cours de stream et problème réseau.

Il doit également être connecté à l'application client via un réseau local. L'application sera créée dans le but de rendre accessible la lecture de musique à l'utilisateur. Elle devra être constituée d'un player audio et d'un système de recherche. Une des finalités de ce projet est de réduire les coûts en stockage. Il est alors primordial que l'application soit multi-plateforme : elle devra, dans un premier temps, être disponible sur Linux et Windows puis sur Android.

L'objectif secondaire est de rendre le serveur de streaming accessible via un réseau distant au moyen de l'application client. L'utilisateur doit pouvoir accéder aux musiques stockées sur le serveur avec la même facilité que sur un réseau local.

Un site web sera également mis en ligne afin que les utilisateurs puissent découvrir et comprendre le fonctionnement de cette application. Il devra permettre la création de comptes clients (avec identifiant et mot de passe) rendant disponible le téléchargement gratuit du logiciel.

Il est nécessaire de posséder au moins une Raspberry Pi et un disque dur afin de créer le serveur.

L'équipe du projet pense proposer une traduction de l'application en anglais afin de la rendre accessible à un plus grand nombre d'utilisateurs.

I.3. Précisions sur le site

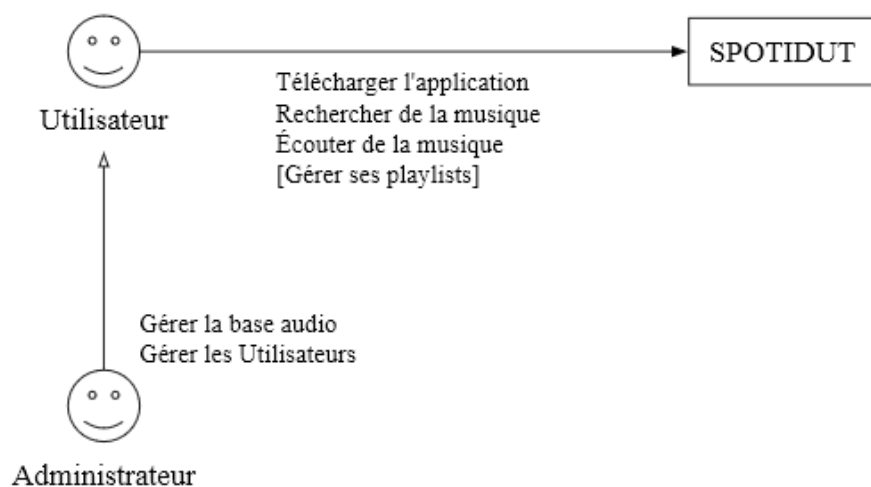
Le site a pour objectif principal l'inscription des utilisateurs dans la base de données. Ainsi, chaque utilisateur pourra s'inscrire afin de pouvoir utiliser l'application (après l'avoir téléchargée). Il utilisera alors son login et son password pour se connecter. Sur ce site, il est alors nécessaire de créer une page d'inscription permettant d'entrer les données dans la base de données située sur le serveur. Vous pouvez trouver les autres pages et apports du site dans le tableau récapitulatif ci dessous.

Tableau du contenu du site :

Page web	Description
A propos	Cette page indique l'identité des développeurs et dans quel but ils ont réalisé ce projet.
Téléchargement	Cette page résume l'utilité de télécharger l'application et propose plusieurs liens afin de télécharger l'application selon votre support (le téléchargement du serveur et du client sera disponible).
Aide	Cette page est mise en ligne afin que les utilisateurs puissent trouver solutions à leurs problèmes. Cette rubrique contiendra également des tutoriels permettant l'installation du client et du serveur sur une Raspberry Pi.
Inscription	Cette page est un simple formulaire permettant l'inscription d'un utilisateur dans la base de donnée afin qu'il puisse utiliser l'application.

II. Formalisation

II.1. Acteurs



L'acteur de ce système sera l'**Utilisateur**. Il s'agit d'un particulier lambda qui souhaite écouter de la musique via son ordinateur ou son mobile. Il pourra se connecter au serveur via le client, et pourra donc rechercher et lire des musiques.

Les Utilisateurs du système pourront également obtenir le rôle d'**Administrateur**, leur permettant un accès à la gestion des morceaux, à la gestion des Utilisateurs et à la mise en ligne du serveur. Cependant, il aura également accès aux autres fonctionnalités du client (lire et rechercher des musiques...).

II.2. Définition des fonctionnalités

II.2.a. Fonctionnalités du client

L'Utilisateur doit pouvoir :

- > Se connecter au serveur via l'application. Il devra entrer un identifiant et un mot de passe. Il doit ensuite pouvoir s'en déconnecter.
- > Une fois connecté au serveur, l'Utilisateur doit avoir accès à la liste des musiques disponibles et doit pouvoir faire des recherches sur cette liste (par titre, par artiste, par album...)
- > Après avoir trouvé la musique qu'il souhaitait, il lui est possible de lancer la lecture. Il peut aussi stopper la lecture en cours et la relancer.
- > Il doit également pouvoir avancer dans la musique ou reculer.
- > Enfin, il est nécessaire de pouvoir revenir à la musique précédente et de pouvoir passer à la musique suivante.

L'Administrateur doit pouvoir :

- > Ajouter ou supprimer une musique (ou un album) du serveur.
- > Enfin, il doit pouvoir supprimer des Utilisateurs (dans le cas où son pseudo est à caractère offensant, par exemple).

Dans la mesure du possible, nous essayerons d'implémenter un système de playlists personnalisées. Cette implémentation conduira alors à de nouvelles fonctionnalités. Ces dernières seront détaillées dans le document présent mais seront encadrées par des crochets.

Ainsi, l'Utilisateur pourra :

- > [Créer et supprimer une playlist]
- > [Ajouter et supprimer une musique de la playlist]

II.2.b. Fonctionnalités du site web

L'Utilisateur doit pouvoir:

- > Se créer un compte sur le site web de Spotidut. Pour cela, il doit remplir un formulaire assez court. (Cette inscription lui donnera accès au téléchargement et donc à l'utilisation de l'application client et à celle de l'application serveur.)

II.3. Lots fonctionnels

II.3.a. Fonctionnalité de l'application

Nom du lot	Authentification
Description	Permet à l'Utilisateur de se connecter et se déconnecter du serveur.
Fonctionnalités	> Se connecter

	> Se déconnecter
--	------------------

Nom du lot	Recherche
Description	Permet à l'Utilisateur de rechercher (et donc de trouver) un morceau dans le serveur.
Fonctionnalités	<ul style="list-style-type: none"> > Rechercher un morceau par titre > Rechercher un morceau par album > Rechercher un morceau par musique > [Rechercher une playlist]

Nom du lot	Lecteur
Description	Permet à l'Utilisateur de lire et d'interagir avec le morceau.
Fonctionnalités	<ul style="list-style-type: none"> > Lecture du morceau > Mettre la lecture en pause > Avancer et reculer dans la lecture > Passer à la musique suivante et précédente

Nom du lot	Administration
Description	Regroupe les fonctionnalités propres à l'Administrateur
Fonctionnalités	<ul style="list-style-type: none"> > Ajouter et supprimer des musiques du serveur > Ajouter et supprimer des albums du serveur > Supprimer un Utilisateur

Nom du lot	[Playlist Personnalisée] (Lot Facultatif)
Description	Regroupe toutes les fonctionnalités permettant la gestion des playlists personnalisées
Fonctionnalités	<ul style="list-style-type: none"> > Créer et supprimer une playlist > Ajouter et supprimer une musique de la playlist

II.3.b. Lot fonctionnel du site

Nom du lot	Site
Description	Présente l'application, ainsi que sa mise en place. Permet aussi la création d'un compte.

Fonctionnalités	<ul style="list-style-type: none"> > Créer un compte > Télécharger l'application
-----------------	--

III. Architecture technique

Le principal défi de ce projet est de réussir à créer un serveur sur Raspberry Pi. Or, ses performances sont du moins réduites. Ils nous faudra donc optimiser toute la partie programmation et opter pour les solutions les plus adéquates.

Raspberry Pi : La Raspberry Pi est un mini ordinateur, ses fonctionnalités sont donc réduites par rapport à celles d'un PC. Le manque de puissance et la petite mémoire de stockage seront nos deux problèmes majeurs. Cependant, c'est aussi là qu'est toute l'originalité de ce projet. Il nous faudra réfléchir et trouver les meilleures solutions pour venir à bout de ce projet. Par exemple, il nous faudra réaliser des tests sur le nombre d'utilisateurs pouvant écouter une musique en même temps (tests de surcharge) afin de réaliser le meilleur serveur possible.

Client : Du côté client, nous rencontrerons certainement moins de problème de puissance et de stockage. Cependant, nous ferons tout de même en sorte que l'application soit rapide et fonctionnelle. Elle sera développée de manière à ce que les utilisateurs Windows et Linux puissent l'utiliser. Enfin, elle devra également être réalisée sur Android.

Base de donnée : Une base de donnée sera nécessaire pour mener à bien le projet. En effet, elle permettra de stocker toutes les informations musicales mais aussi toutes les informations concernant les utilisateurs (identifiant, mot de passe...). Cette base de donnée sera stockée sur la Raspberry Pi. Une des principales difficultés sera donc de faire communiquer la base de donnée avec le site, puisque ce dernier sera, lui, hébergé sur un serveur externe.

IV. Glossaire

- > **Base De Données (BDD)** : Une base de données ou banque de données est un outil permettant de stocker et de retrouver l'intégralité de données brutes ou d'informations en rapport avec un thème ou une activité.
- > **Ethernet** : Norme de protocole de réseau très répandue. La norme Ethernet utilise un type de connexion filaire. C'est la norme la plus utilisée pour les réseaux locaux et la connexion internet en filaire.

- > **LAN (Local Area Network)** : Réseau informatique tel que les terminaux qui y participent s'envoient des trames au niveau de la couche de liaison sans utiliser d'accès à internet.
- > **Mémoire flash** : Support de stockage pour des photos, vidéos, musiques au format numérique. Cette mémoire présente de nombreux avantages : réinscriptibles, les données sont conservées lorsque l'alimentation électrique est coupée. De plus, contrairement au disque dur, ce type de mémoire ne possède pas de pièces mécaniques, ce qui lui confère une grande résistance aux chocs.
- > **Plateforme** : Base de travail à partir de laquelle on peut écrire, lire, utiliser un ensemble de logiciels.
- > **Player** : Un player est une application de lecteur vidéo ou audio.
- > **Protocole**:Un protocole réseau est un mis en œuvre sur un
- > **Raspberry Pi** : Mini mono carte à de la taille d'une carte de crédit.
- > **Serveur** : Un serveur informatique est un dispositif matériel ou logiciel qui offre des services à différents clients. Quelques services utilisés ici :
 - le stockage en base de données,
 - la gestion de l'authentification.
- > **Streaming** : Traitement effectué en continu. Cette technologie est par exemple utilisée afin de diffuser un fichier multimédia sur Internet comme la musique ou la vidéo.
- > **Terminal** : Un terminal de console est une des consoles système fournis dans les terminaux de la console Linux actes kernel.T (moyen pour les opérations d'entrée et de sortie pour un système Linux) Un terminal console Linux est similaire à la ligne de commande dans Microsoft Windows, mais elle diffère en ce qu'il peut effectuer toute opération sur le système
- > **Wi-Fi** : Technologie permettant de se connecter à Internet sans utiliser de fil.

Source:

PARTIE 2 : SPÉCIFICATIONS

I. Spécifications fonctionnelles

Echelle importance :

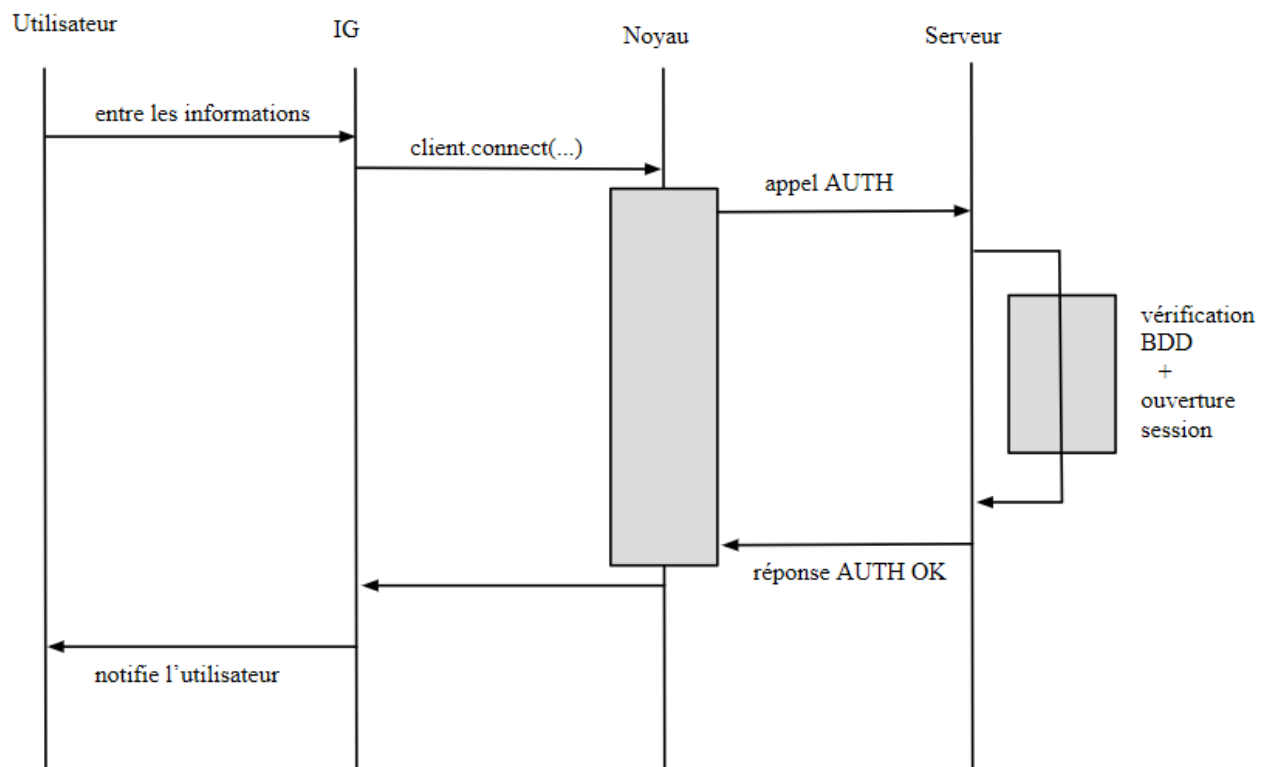
- **critique** : aucune erreur n'est permise. Sinon l'application ne peut pas fonctionner.

- **important** : en cas de problème, la majorité de l'application ne peut pas fonctionner
- **moyenne** : en cas de problème, n'empêche pas le système de fonctionner mais comportement anormal
- **faible** : en cas de problème, ne pose aucun soucis de fonctionnement au système

Echelle fréquence :

- **importante** : CU qui revient très fréquemment
- **moyenne** : CU susceptible d'apparaître une bonne quantité de fois par utilisation de l'application, mais n'est pas dans l'utilisation quotidienne de celle-ci au point d'être "importante" (ajout de musiques par l'administrateur par exemple)
- **faible** : n'arrive qu'une fois environ par utilisation de l'application

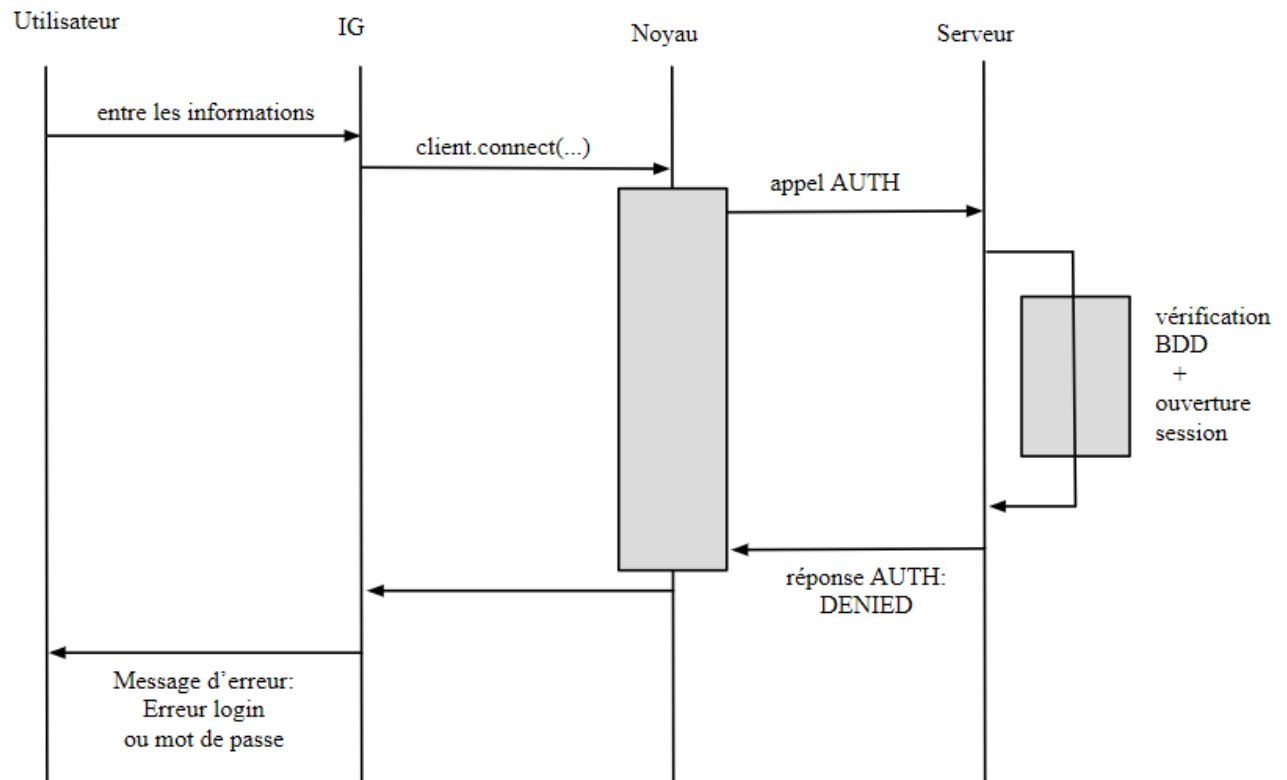
__ Se connecter __

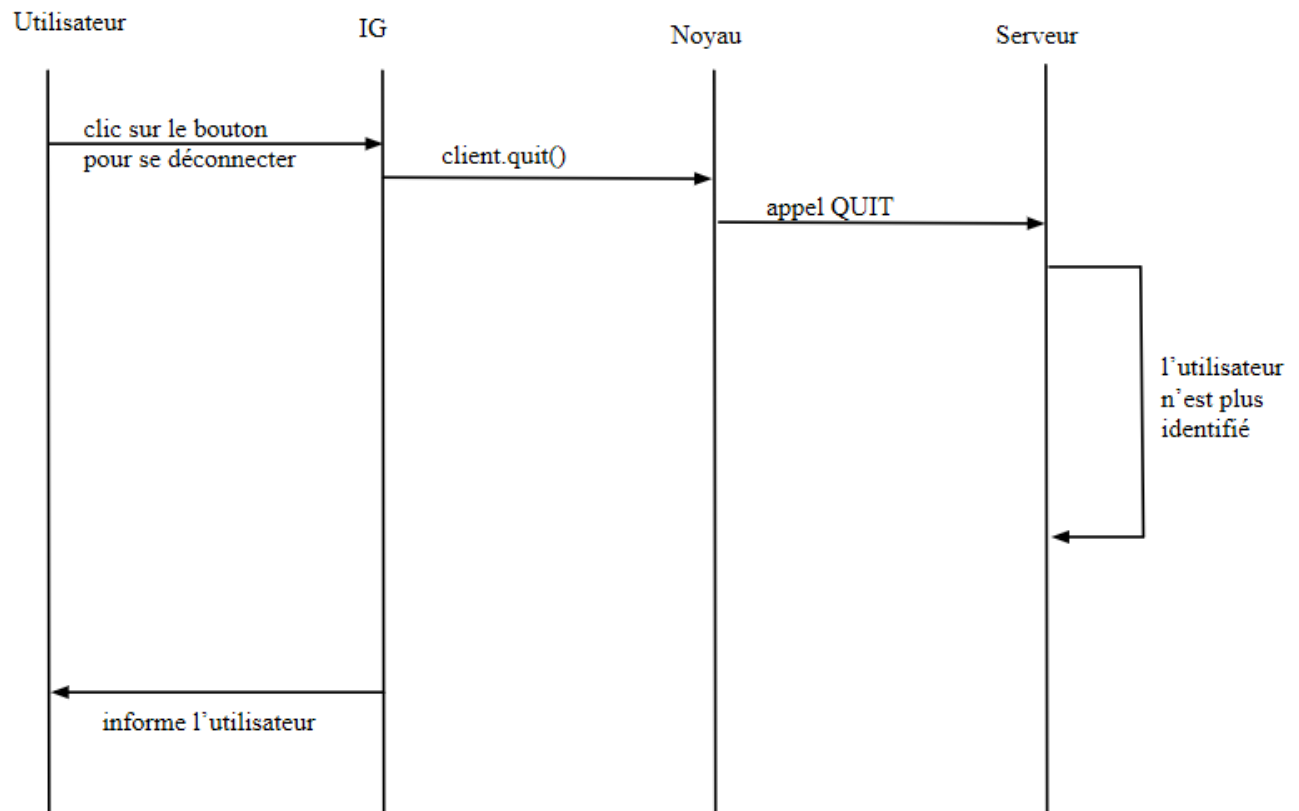


Description	Permet à l'utilisateur de se connecter à l'application et d'accéder aux fonctionnalités de celle-ci.
Fréquence	Moyenne
Importance	Critique

Pré-condition	- Ne pas être connecté - Avoir une connexion au serveur
Post-condition	L'Utilisateur est identifié.

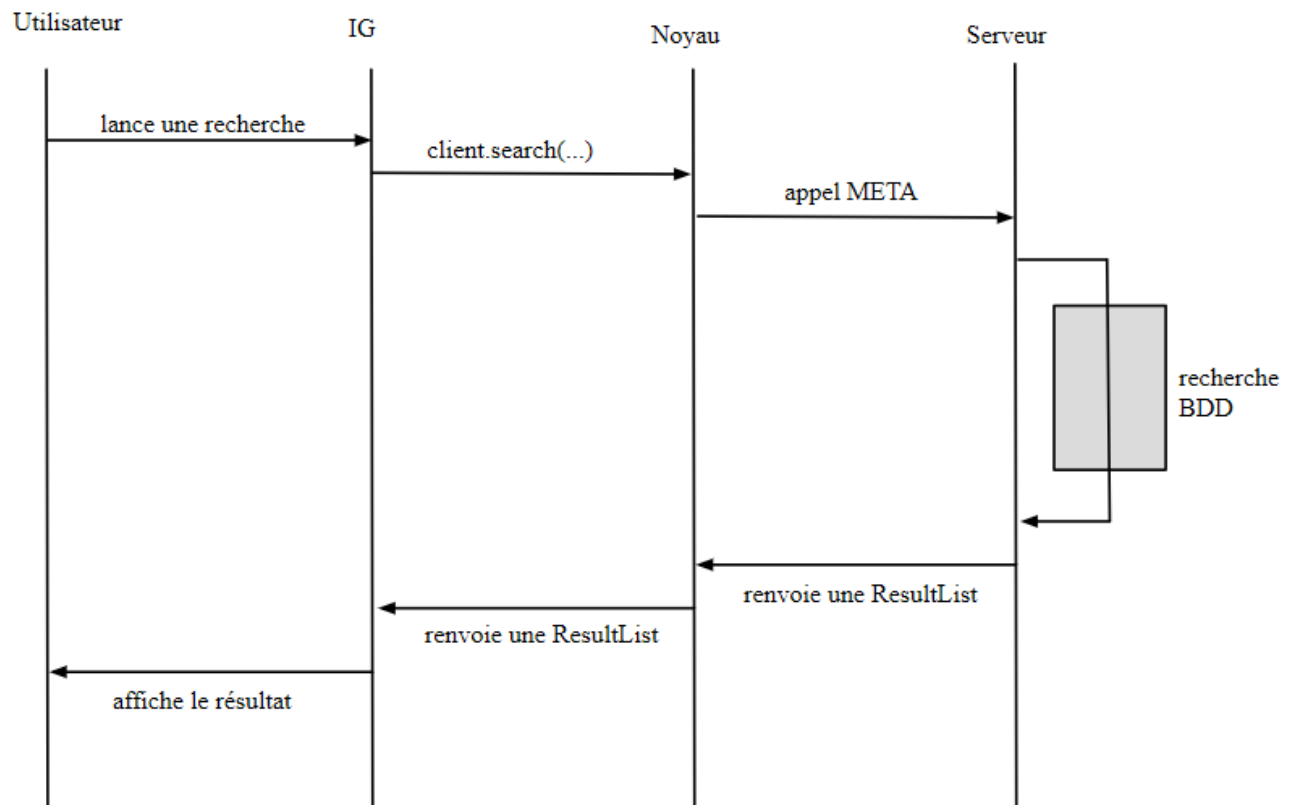
Scénario Alternatif 1: Se connecter - **Erreur Login / mot de passe**



Se déconnecter

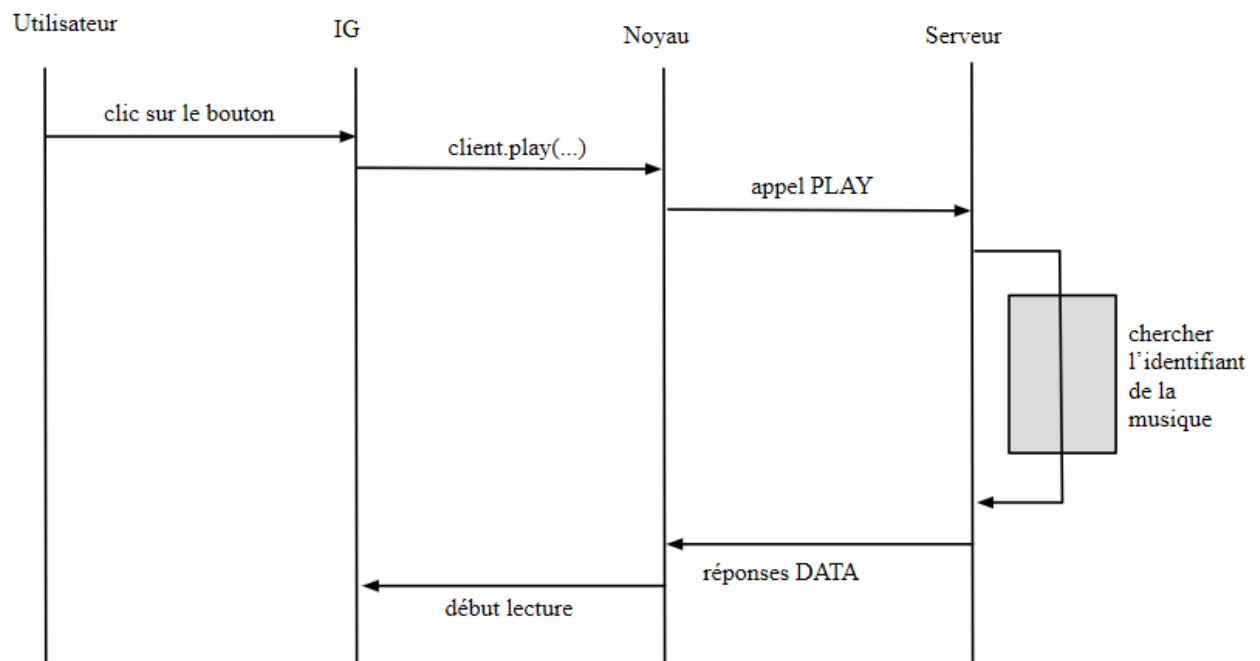
Description	Permet de quitter l'application après l'utilisation de celle ci ou pour se connecter sous un autre compte.
Fréquence	Faible
Importance	Moyenne
Pré-condition	- Etre connecté - Avoir une connexion au serveur
Post-condition	L'Utilisateur n'est plus identifié. Il peut s'identifier sous un autre nom.

Pas de scénario alternatif.

__ Faire une recherche __

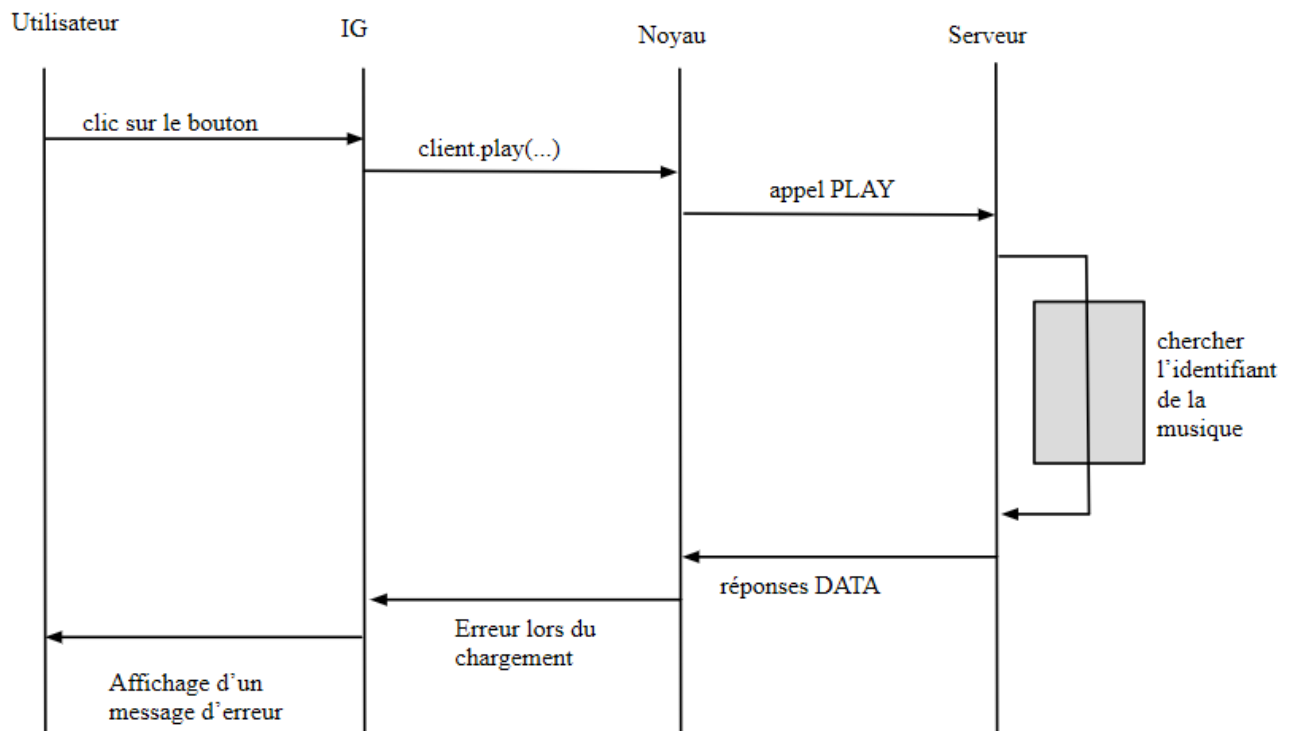
Description	Permet de faire des recherches afin de trouver la musique que l'on souhaite écouter.
Fréquence	Important
Importance	Important
Pré-condition	- Etre identifié
Post-condition	Le résultat de la recherche est affiché.

Pas de scénario alternatif.

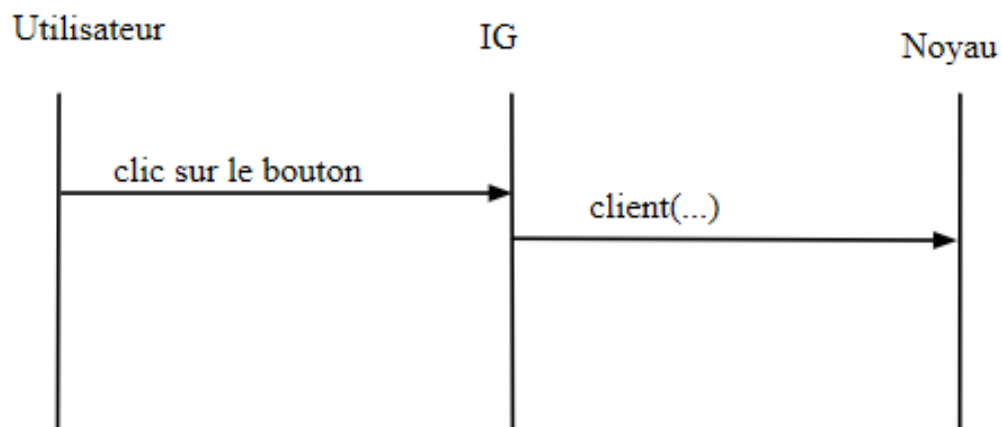
__ Lire une musique __

Description	Permet de lire une musique à partir d'une ResultList
Fréquence	Important
Importance	Critique
Pré-condition	- Etre connecté - Avoir fait une recherche
Post-condition	La musique se charge.

Scénario Alternatif 1 : Lire une musique - **Erreur de chargement**

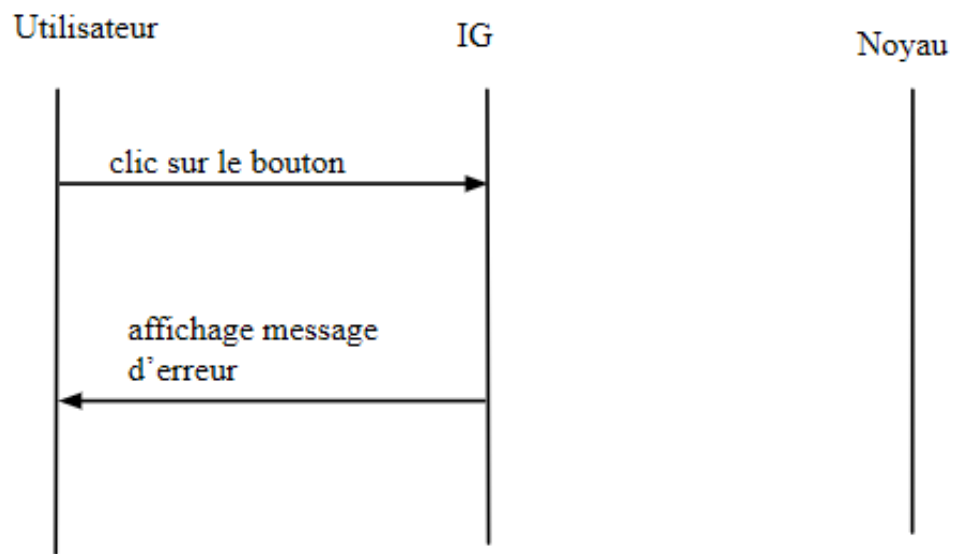


__ Play, Pause, Seek __

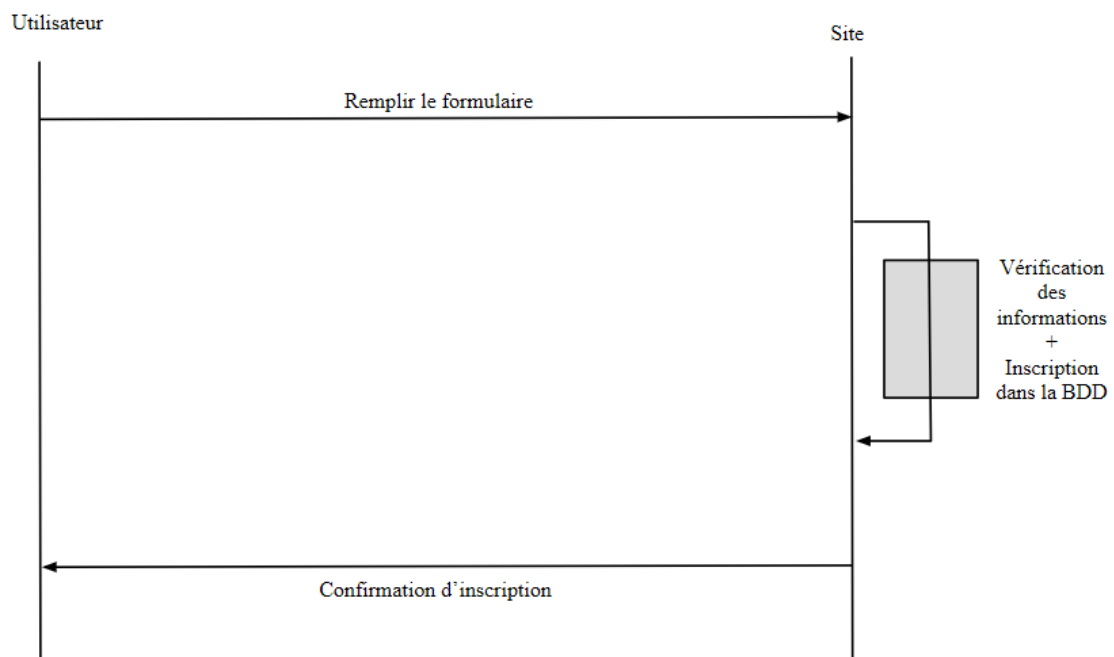


Description	Permet de jouer, mettre en pause ou se déplacer avec la musique
Fréquence	Important
Importance	Moyenne
Pré-condition	- Musique sélectionnée
Post-condition	Mettre une musique en pause, reprendre la lecture, se déplacer dans la musique

Scénario Alternatif 1: Play, Pause, Seek - **Erreur de chargement**

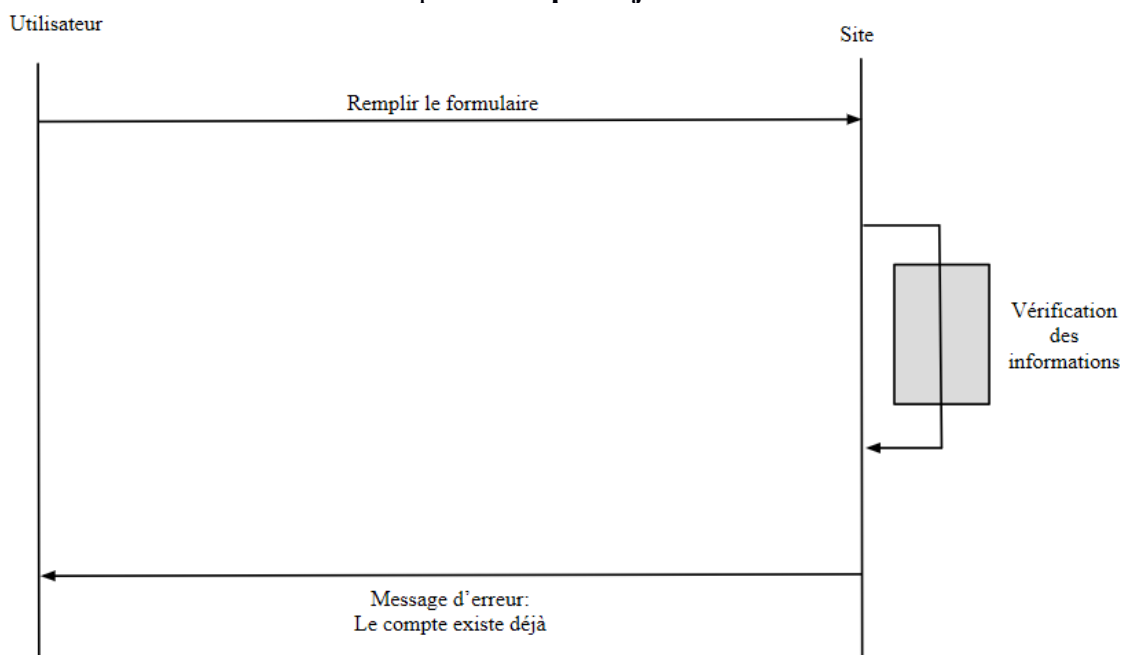


__ Créer un compte __

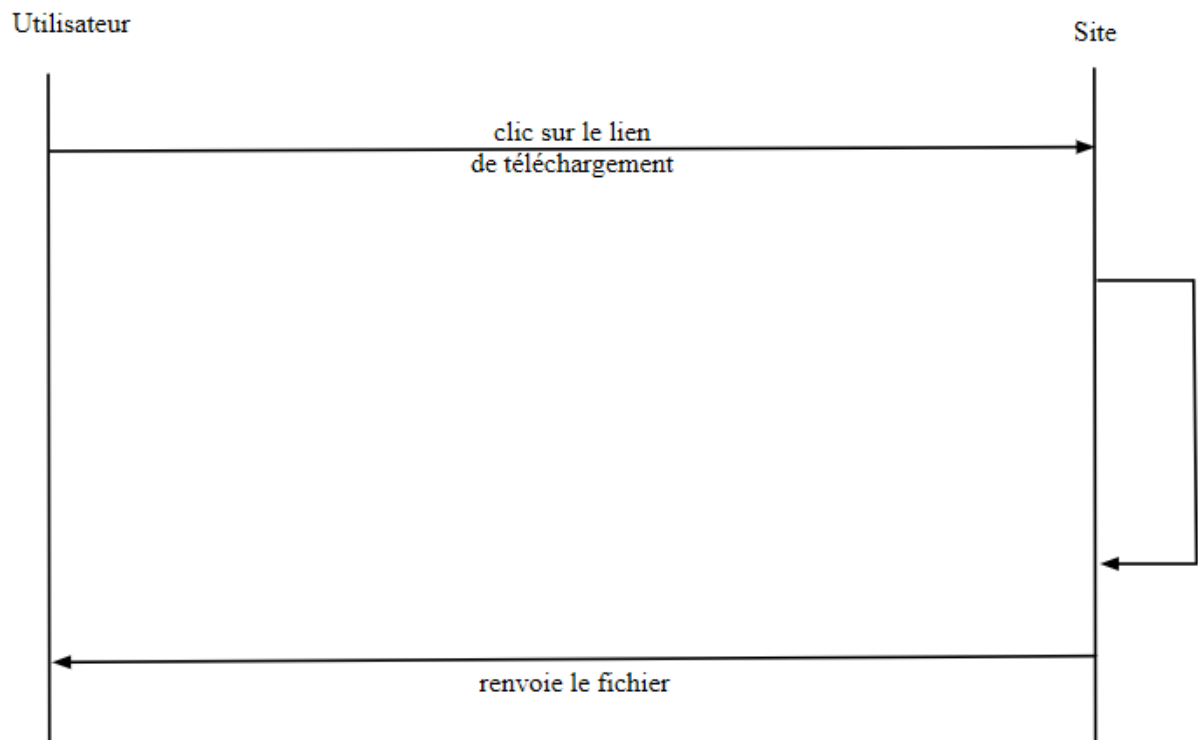


Description	Permet de s'inscrire afin d'utiliser l'application
Fréquence	Faible
Importance	Critique
Pré-condition	- Ne pas avoir un compte
Post-condition	L'Utilisateur possède un compte avec un login et un mot de passe.

Scénario Alternatif 1: Créer un compte - **Compte déjà existant**

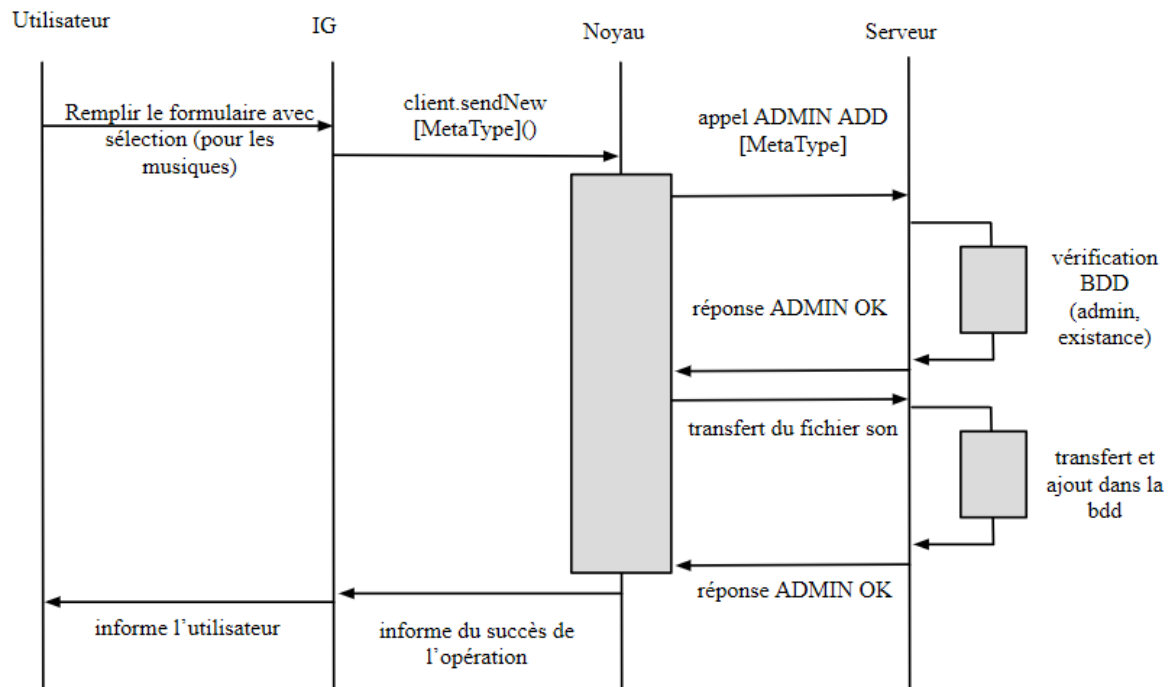


__ Télécharger l'application __



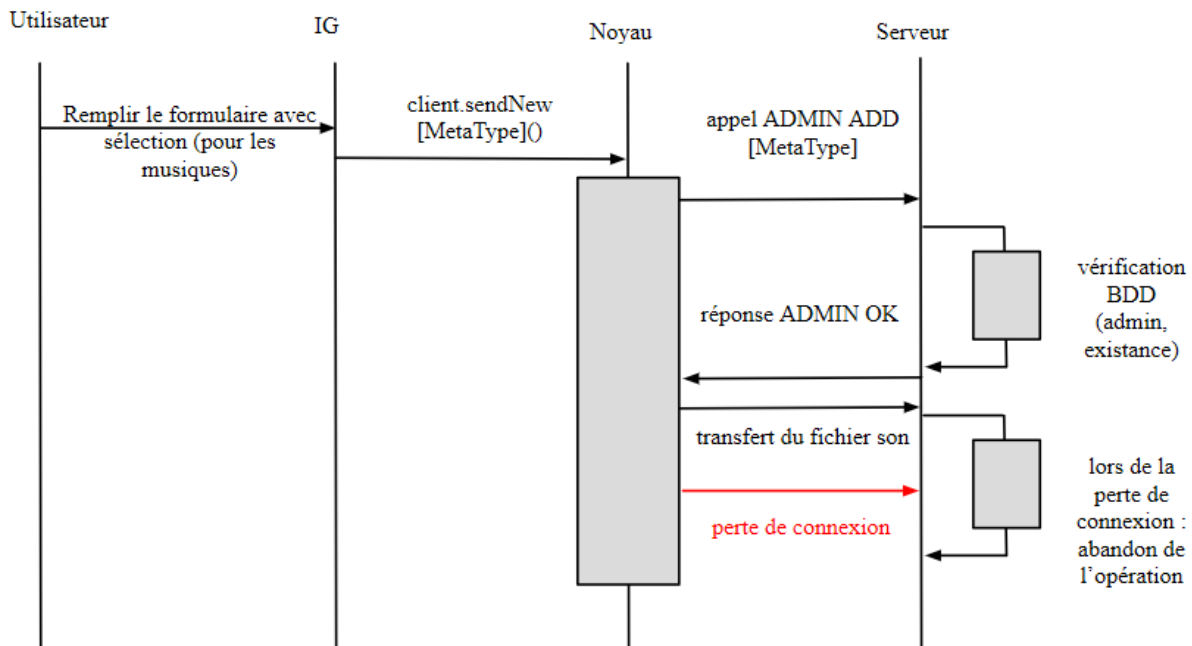
Description	Permet de télécharger l'application
Fréquence	Faible
Importance	Critique
Pré-condition	- Avoir un compte
Post-condition	L'Utilisateur a obtenu l'application.

Pas de scénario alternatif.

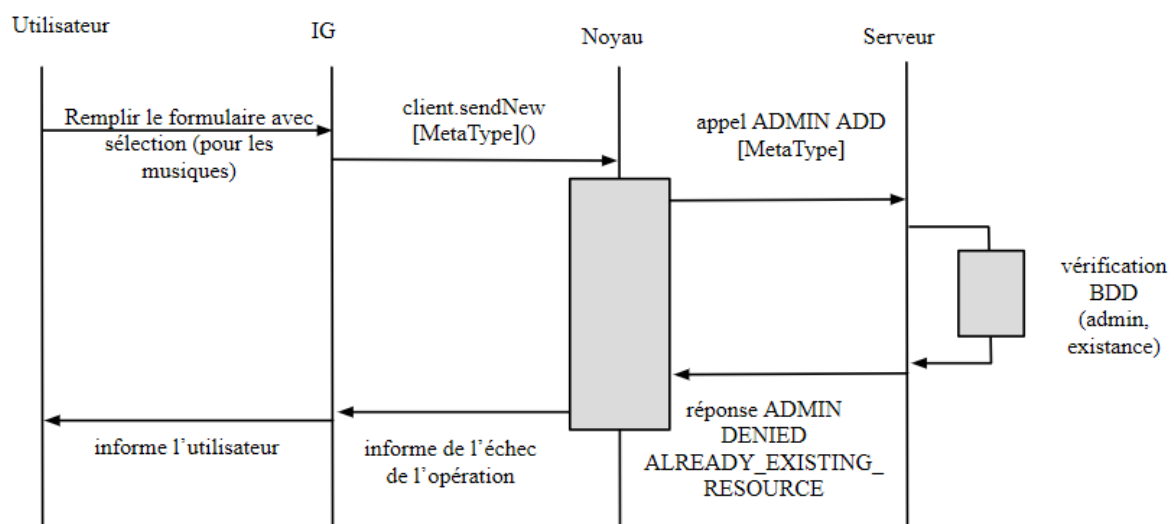
__ Ajouter une ressource __

Description	Permet d'ajouter une ressource
Fréquence	Moyenne
Importance	Critique
Pré-condition	- Posséder un compte administrateur
Post-condition	La ressource a été ajoutée.

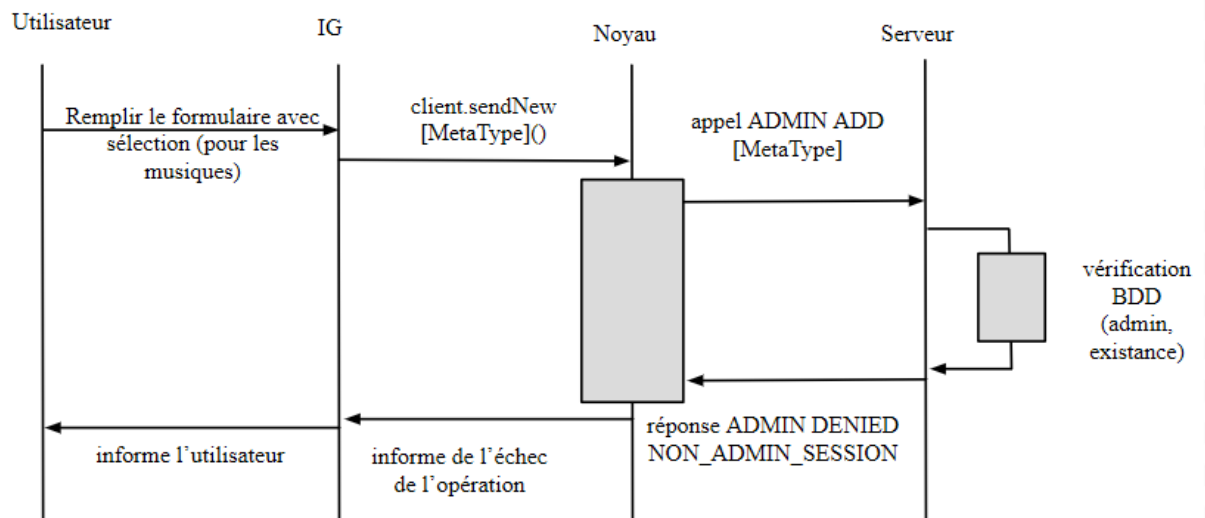
Scénario Alternatif 1 : Ajouter une ressource - **Erreur de transfert (perte de connexion)**



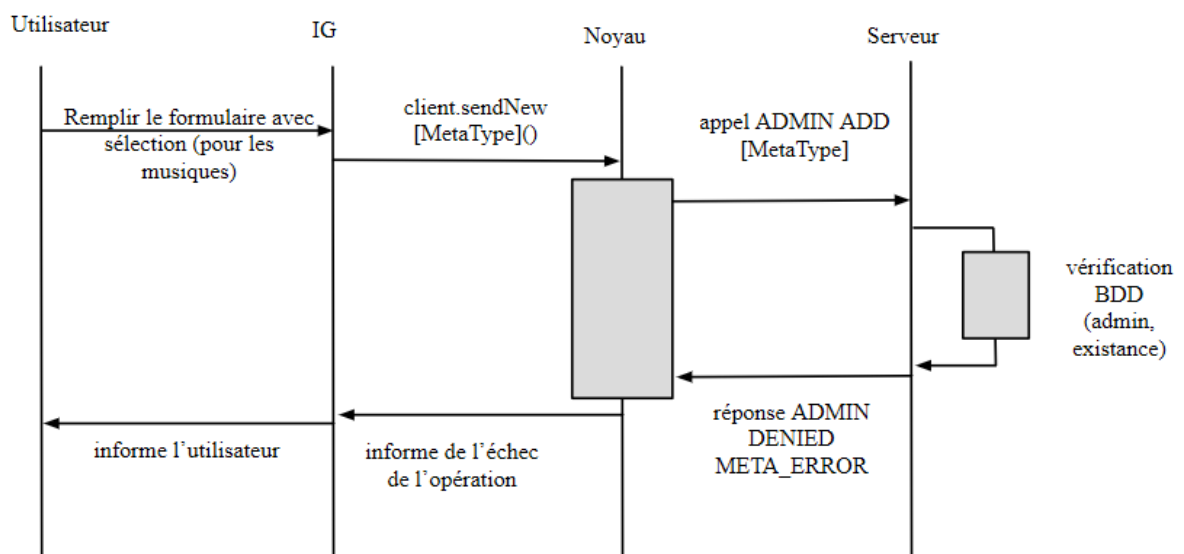
Scénario Alternatif 2: Ajouter une ressource - **Ressource déjà existante**



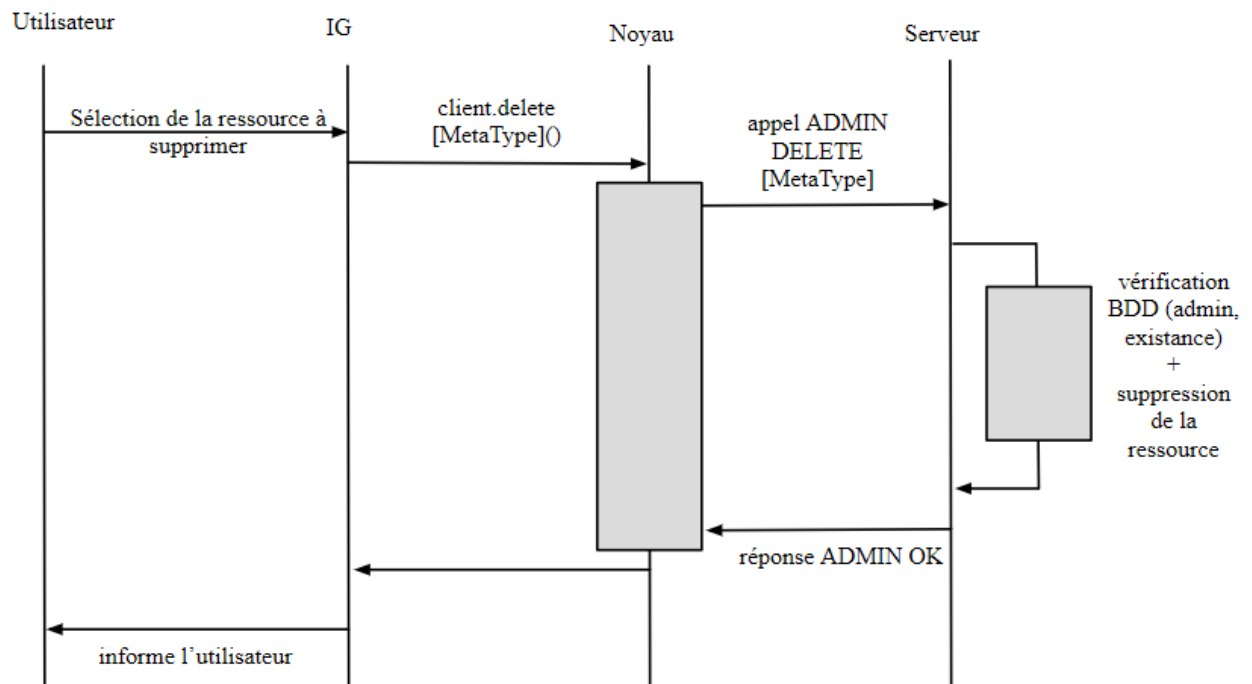
Scénario Alternatif 3 : Ajouter une ressource - **Pas de droit**



Scénario Alternatif 4 : Ajouter une ressource - **Erreur informations**

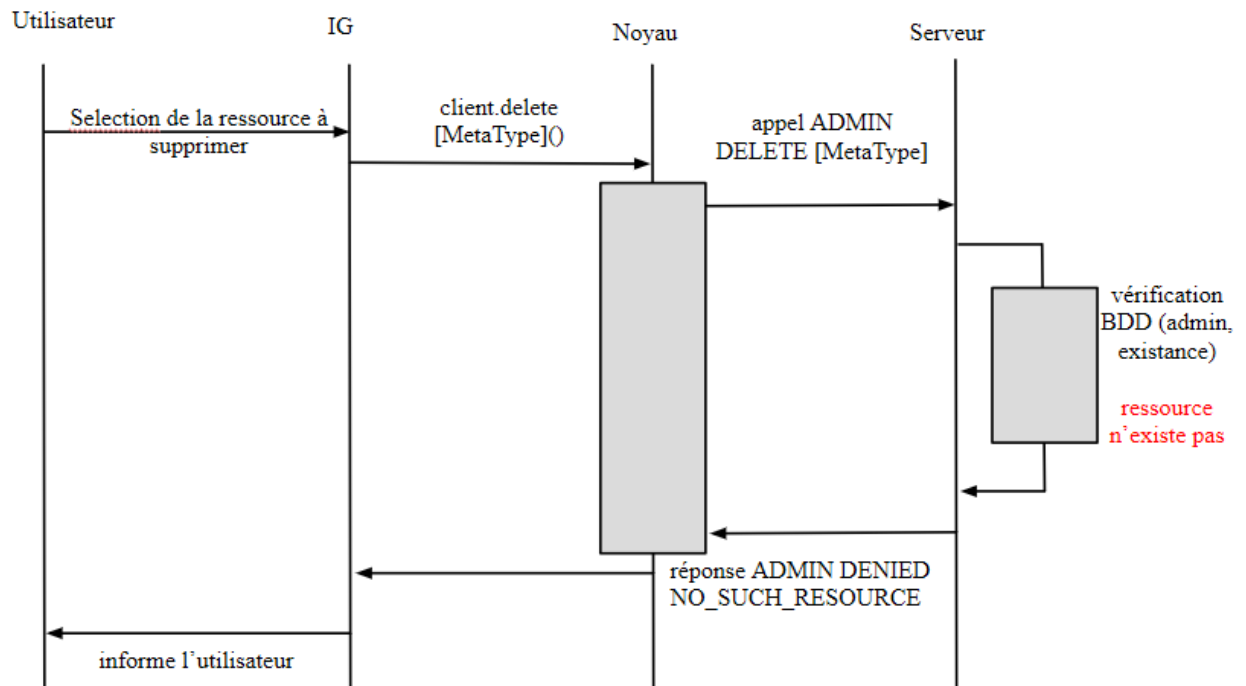


__ Supprimer une ressource __

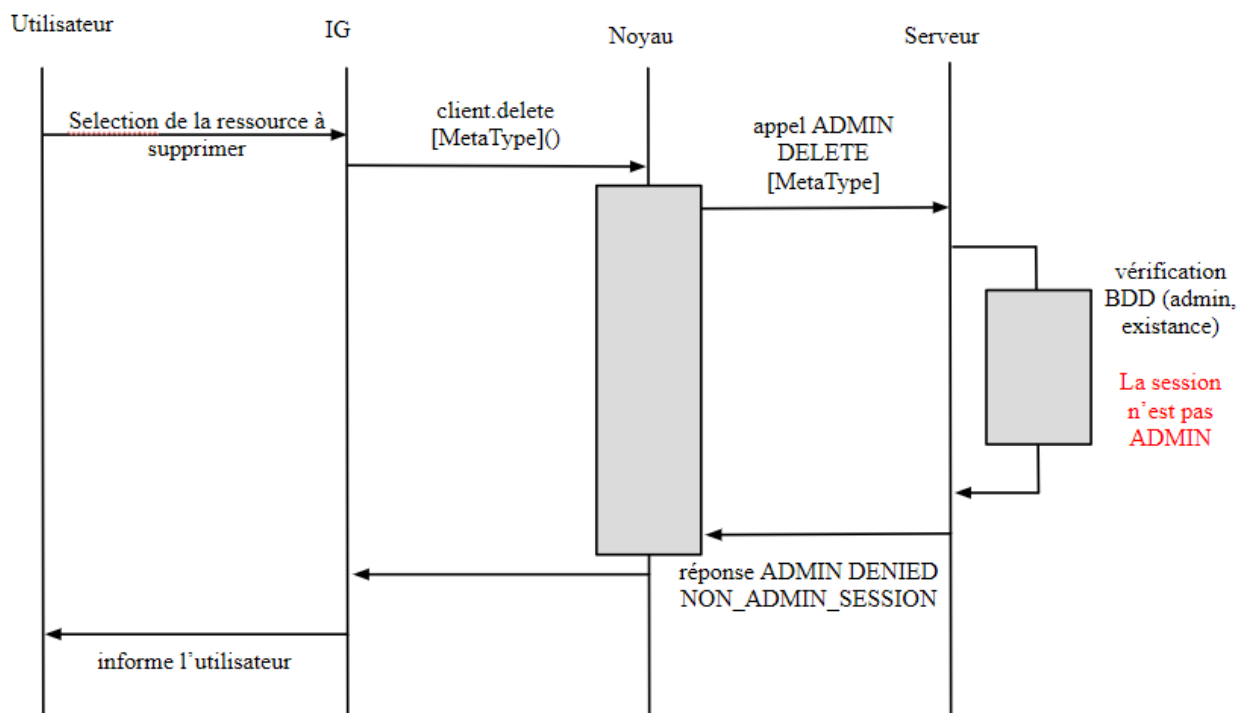


Description	Permet de supprimer une ressource
Fréquence	Faible
Importance	Faible
Pré-condition	- Posséder un compte administrateur
Post-condition	La ressource a été supprimée.

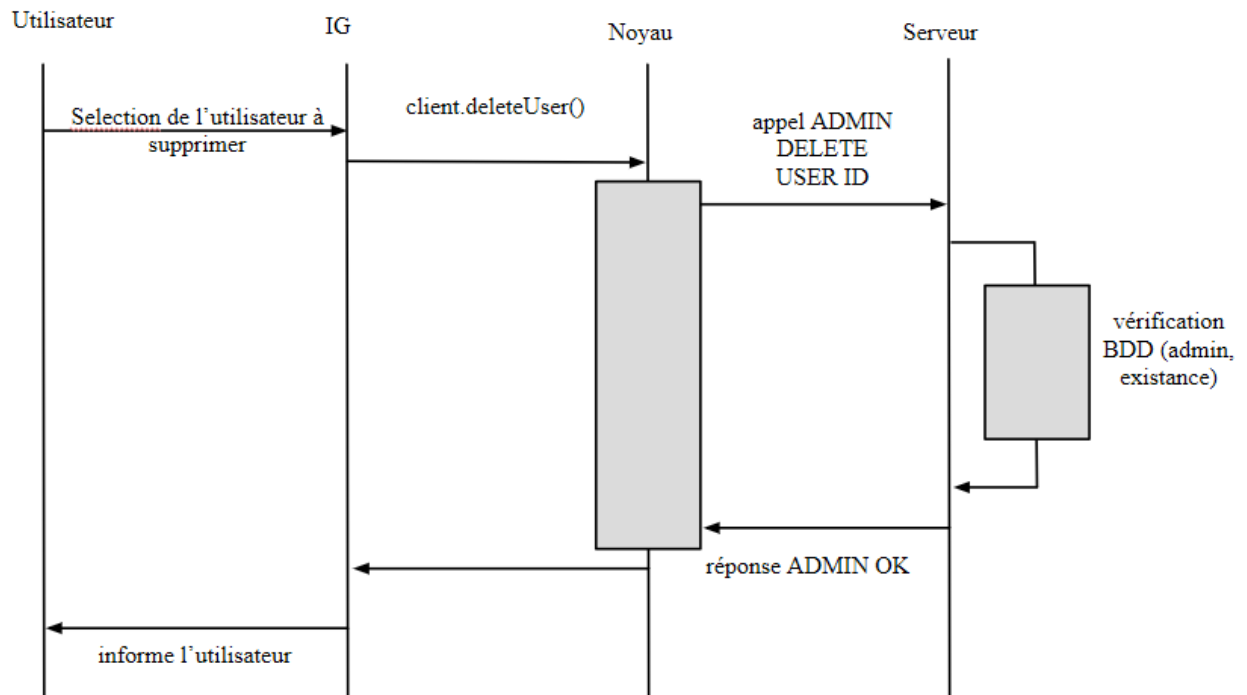
Scénario Alternatif 1: Supprimer une ressource - **La ressource n'existe pas ou plus**



Scénario Alternatif 2 : Supprimer une ressource - Pas de droit

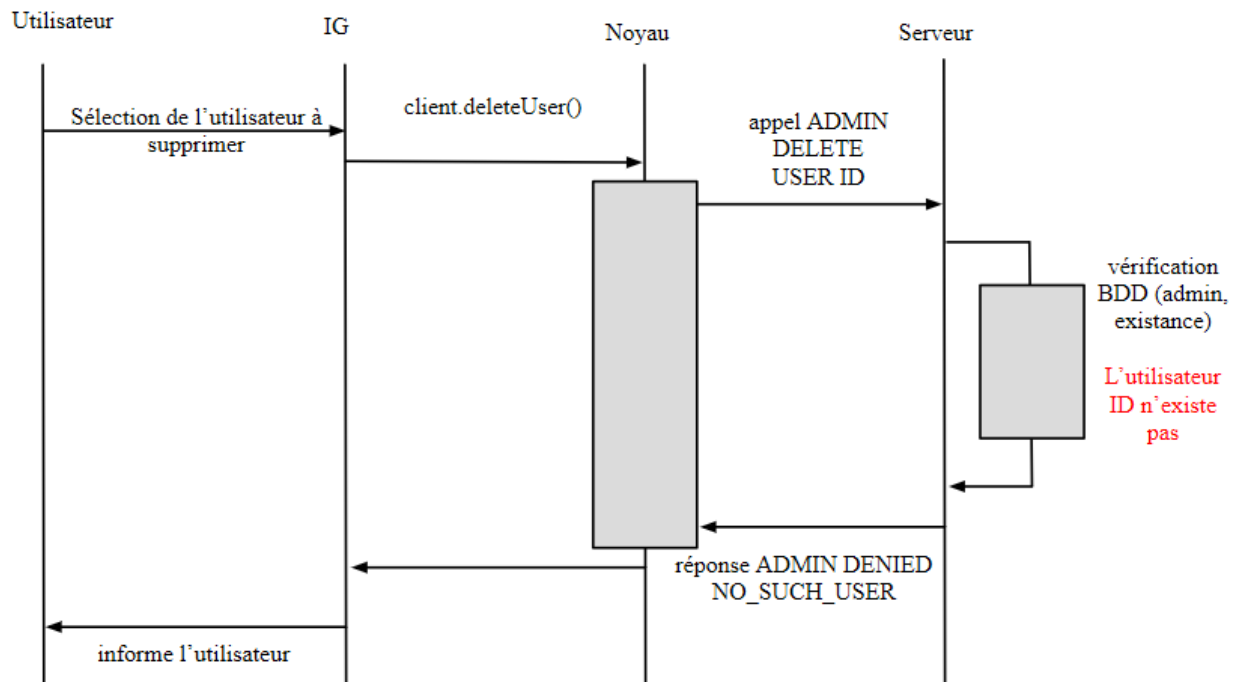


__ Supprimer un utilisateur __

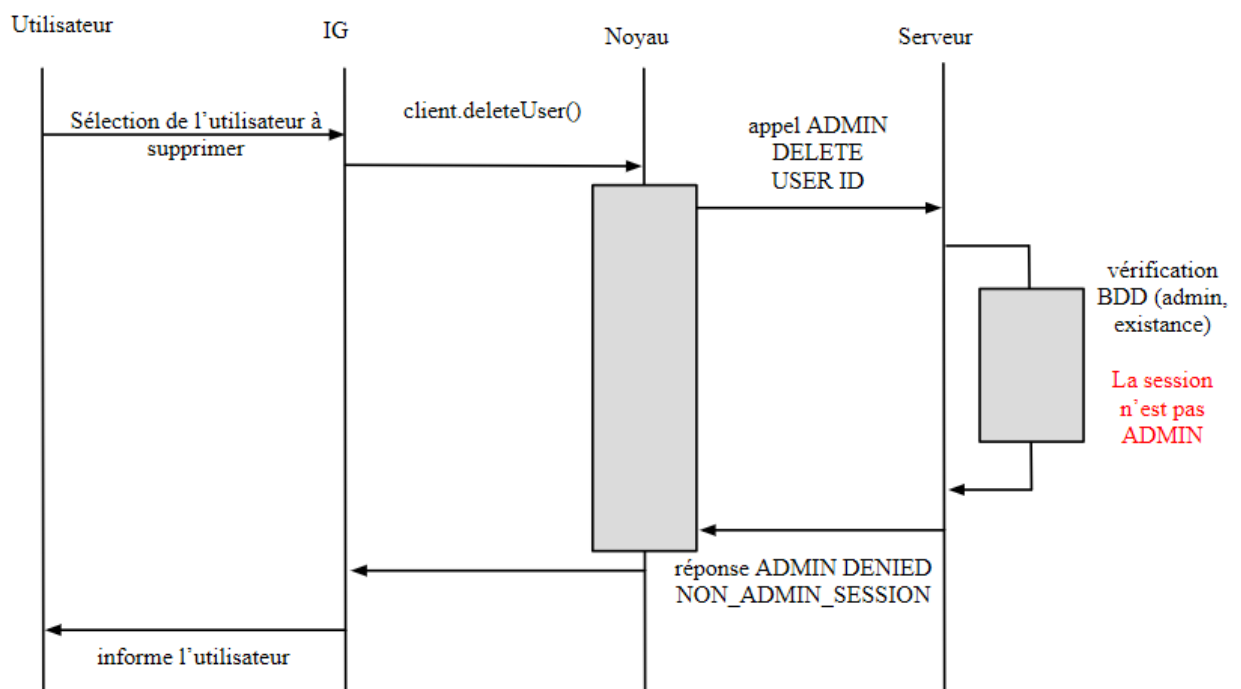


Description	Permet de supprimer un utilisateur
Fréquence	Faible
Importance	Faible
Pré-condition	- Posséder un compte administrateur
Post-condition	L'Utilisateur a été supprimé.

Scénario Alternatif 1: Supprimer un Utilisateur - **Utilisateur non existant**



Scénario alternatif 2 : Supprimer un utilisateur - Session non admin



II. Spécifications graphiques du client

II.1. Chartes graphiques

Notre objectif est produire une application au design flat, pour cela, nous avons défini les caractéristiques suivantes.

II.1.a. Couleurs

La couleur est une des principale définition d'un design flat, nous avons choisi :

- Les nuances de gris pour toute l'application. Le gris est une couleur douce, apaisante et calme, l'utilisateur ne se sentira alors pas agressé et passera beaucoup plus de temps sur l'application. Ci-dessous, la palette des gris utilisés (cette palette sera peut être amenée à de légères modifications) :



Annexe 1 : Palette grise

- L'utilisation du vert pour contraster et faire ressortir les titres et informations importantes. Le vert est également une couleur apaisante mais elle est également tonifiante et ressort très bien sur du gris.



Annexe 2 : Palette verte

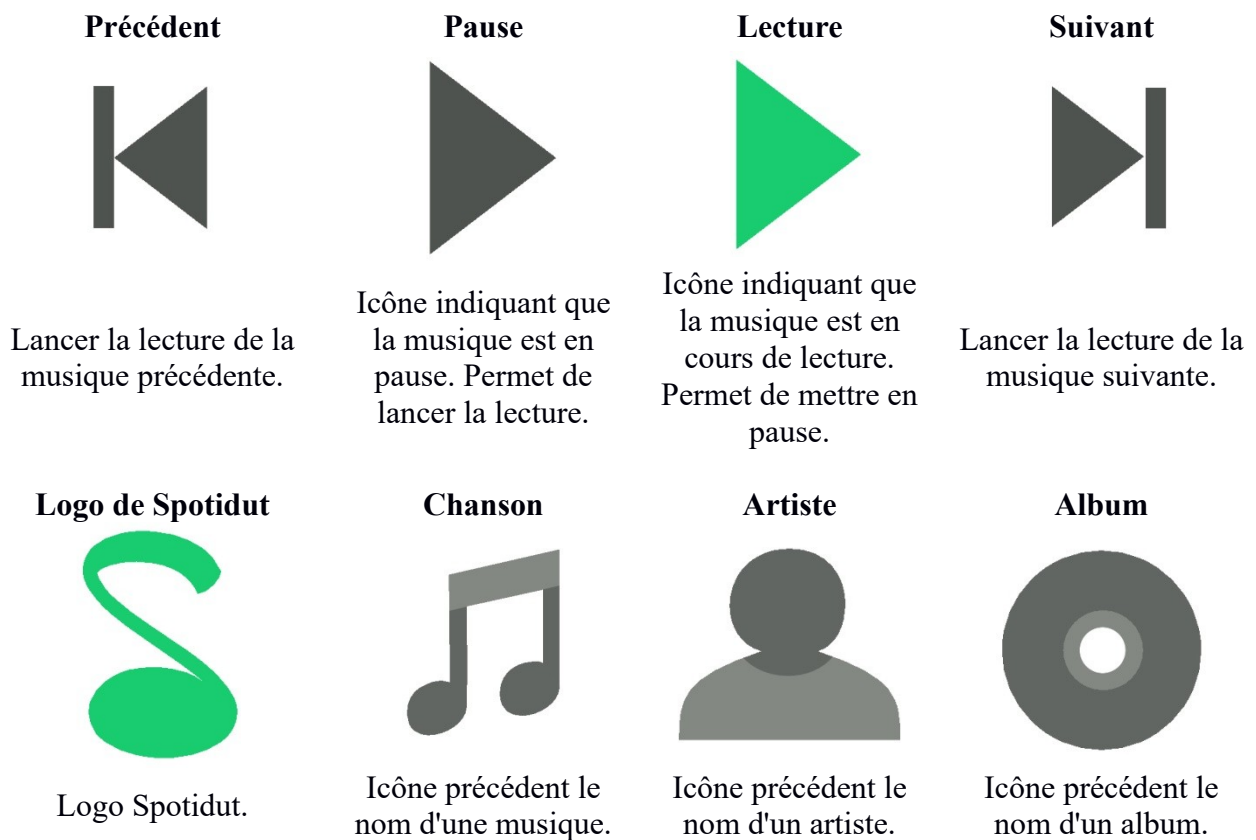
II.1.b. Typographie

Pour des raisons de compatibilité avec toutes les machines et pour des questions de lisibilité (en effet, certaines personnes peuvent mal distinguer des typographies complexes) nous allons opter pour une typographie neutre sans serif. Une police sans serif est une police sans empattement (petites extensions qui forment la terminaison des caractères). On pense notamment à Arial ou Helvetica.

II.1.c. Icones

De nombreuses icônes vont être créées pour l'application. Elles reprennent évidemment les couleurs définies pour cette dernière. Nous utiliserons certainement des icônes de taille 48*48 pour les grosses icônes et de taille 32*32 pour les plus petites.

Vous pouvez trouver ci-dessous des "maquettes" d'icônes, c'est-à-dire qu'il ne s'agit pas encore des icônes définitives mais seulement d'icônes "grossières" qui seront retravaillées plus en détail.



Un des icônes n'est pas encore défini, il s'agit de l'icône de recherche, suivant la barre de recherche et permettant de lancer cette dernière.

Enfin, le logo textuel de l'application reprend le nom de l'application, avec l'idée de remplacer le S par une note de musique.

Spotidut

Les couleurs du logo textuel ne sont pas définitives et pourront certainement changer.

II.1.d. Popups

Un seul popup a été prévu. En effet, dans le cas où le client ne peut pas atteindre le serveur, un popup indiquera la non connexion. L'application se fermera par la suite.

II.2. Liste des composants

Menu	
Onglet Accueil	Page «dynamique» permettant d'afficher : - l'accueil, - les résultats de recherche, - les fiches album, - les fiches artistes.
Onglet Recherche	
Onglet Liste Musique	
Fiche musique	
Lecteur	

L'application comprendra :

> **Un menu :**

Le contenu détaillé n'a pas encore été défini, mais nous savons déjà que l'utilisateur aura ainsi accès à l'aide et à la possibilité de se déconnecter par exemple.

> **Une barre latérale :**

Cette barre latérale sera composée de trois onglets :

- le premier onglet Accueil permettra d'afficher l'accueil dans la "page dynamique",
- l'onglet Recherche possède un champs texte permettant à l'utilisateur de réaliser une recherche musicale. Les résultats de cette recherche s'afficheront dans la "page dynamique",
- le dernier onglet Liste Musique permettra d'accéder à la liste des musiques en cours de lecture (affichée à nouveau dans la "page dynamique".

Sous ces trois onglets, l'application comportera un cadre "Fiche musique" permettant de connaître toutes les informations relatives à la musique en cours de lecture (Nom de la chanson, nom de l'artiste, nom de l'album).

> **Une barre inférieure :**

Cette barre inférieure comprendra le player musical et tous les boutons permettant d'agir sur la musique et sur la liste de musiques.

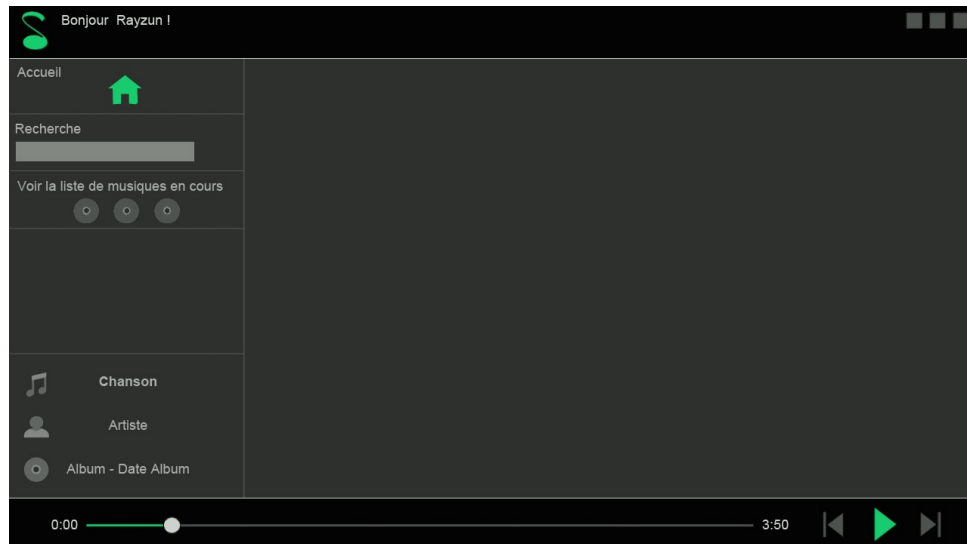
II.3. Définitions des écrans

II.3.a. Windows, Linux et Mac

Nous avons choisi d'utiliser, pour Windows, Linux et Mac, une taille minimale de 640*480 pixels. L'application sera également redimensionnable afin de permettre aux utilisateurs ayant des écrans plus grands d'en profiter pleinement. Windows et Linux

autorisent et permettent tous les deux l'utilisation du clavier et de la souris. Nous partons alors dans l'idée que c'est ces deux moyens de contrôle qui serviront principalement.

Vous trouverez ci-dessous les différentes vues Windows, Linux et Mac.



Annexe 3 : Vue "statique"

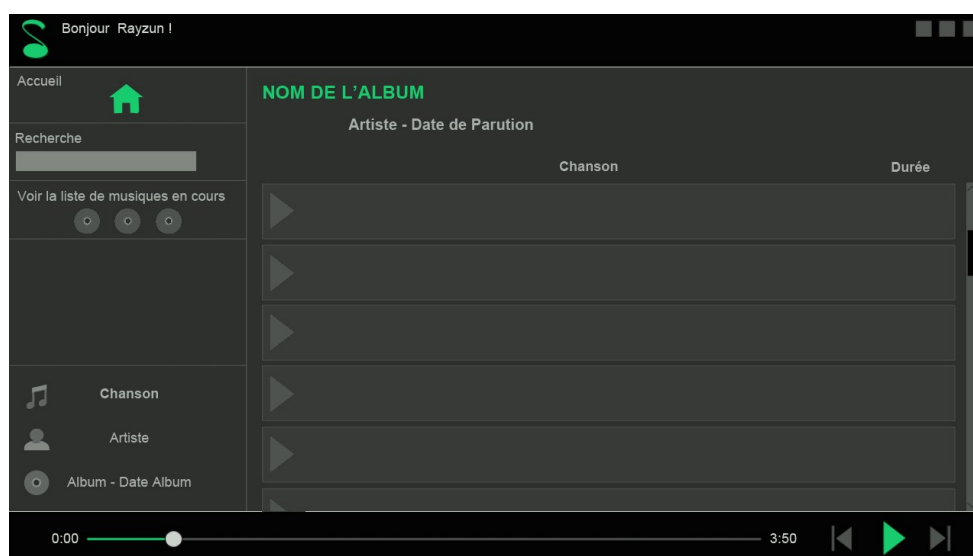


Annexe 4 : Accueil

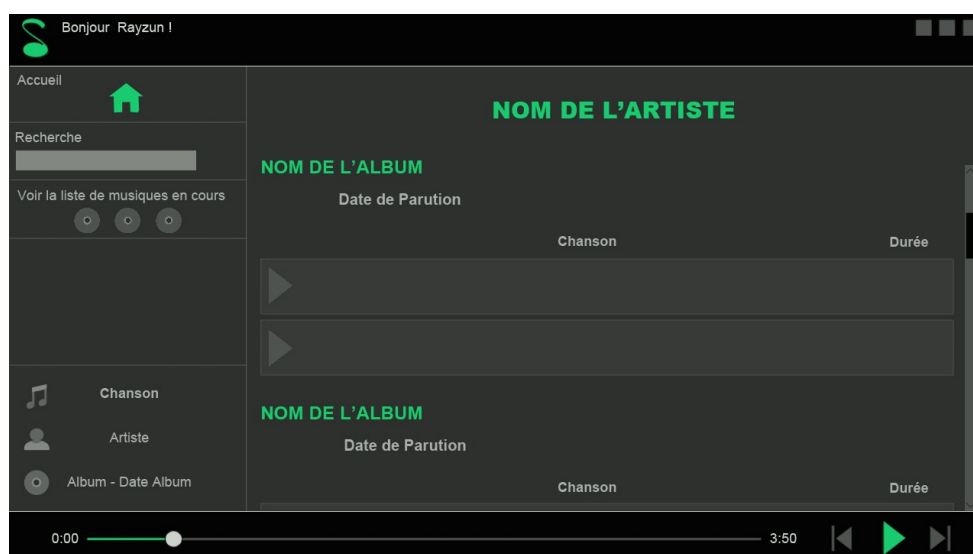
Vue première de l'application. Sera également proposée une liste de musiques aléatoires.



Annexe 5 : Résultat de recherche

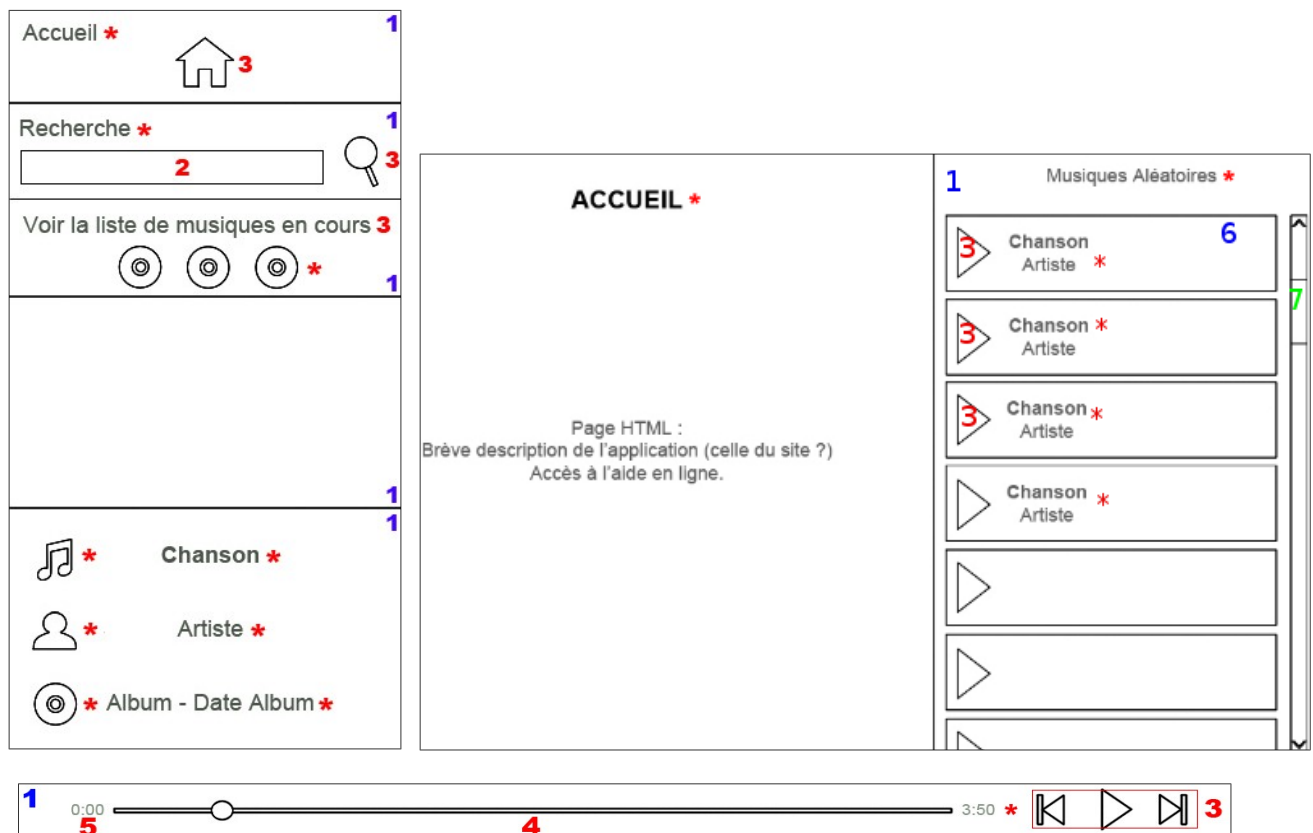


Annexe 6 : Fiche Album



Annexe 7 : Fiche Artiste

Vous trouverez ci-dessous les composants SWING que nous allons utiliser pour la réalisation de l'application. Cette liste non exhaustive sera certainement amenée à évoluer au fil du développement de l'application afin de correspondre au mieux aux attentes.



Le conteneur de l'application est évidemment un JFrame.

Nous allons utiliser des JPanel afin de séparer les différents affichages (chaque onglet sera représenté par un JPanel) ou des JScrollPane (partie scrollable). Afin d'afficher les textes nous utiliserons des JLabel et des JTable. Enfin, pour tout ce qui est interactions, nous utiliserons des JButton mais aussi des JTextField.

II.3.b. Android

Pour le moment, les spécifications Android n'ont pas encore été définies. En effet, n'ayant eu aucun cours de programmation mobile, nous ne savons pas exactement quelles sont les choses réalisables. Nous avons donc décidé de nous concentrer plus en profondeur sur l'application PC.

Néanmoins, même si ces spécifications n'ont pas été réalisées, nous n'y réfléchissons pas moins et avons déjà un bon nombre d'idées. A nouveau, le design restera dans la même optique de celui de l'application PC puisque l'objectif est de garder une cohérence dans l'application, quelle que soit la plateforme.