

A stylized graphic featuring a large, white, multi-lobed cloud shape with a subtle 3D effect, set against a dark blue gradient background. The word "JABBOID" is written in a light blue, sans-serif font across the center of the cloud. A small blue speech bubble icon is positioned above the letter 'I'. Below the cloud, several white lines of varying lengths extend downwards, each ending in a small white circle. These lines are connected to names and a title at the bottom of the image.

J A B B O I D

Coscujuela

Jalabert

Dossier de  
Conception

Ladegaillerie

Rizzo

## Tableau des Validations

Nom	Statut	Date	Signature
Version finale	Manque données + Refaire page de garde	20/06/2014	SJALABERT / TCOSCUJUELA
Version finale	VALIDE	5/08/2014	SJALABERT / TCOSCUJUELA

## Tableau des Révisions

Nom	Date de lecture	Version	Signature
Passage en 0.1	4/06/2014	0.1	DRIZZO
0.5	5/06/2014	0.5	SJALABERT / DRIZZO
0.9	15/06/2014	0.9	SJALABERT

## Contenu

Tableau des Validations .....	2
Tableau des Révisions .....	2
I) DOMAINE D'APPLICATION .....	2
I.1) OBJECTIFS DU SYSTÈME .....	2
I.2) INTERFACES .....	2
I.3) CONTRAINTES DE CONCEPTION .....	2
II) DOCUMENTS DE RÉFÉRENCE .....	3
III) NORMES, STANDARDS & OUTILS .....	4
III.1) ENVIRONNEMENT & OUTILS DE DÉVELOPPEMENT .....	4
III.2) NOTATIONS UTILISÉES .....	4
III.3) STANDARDS DE PROGRAMMATION .....	4
IV) ANALYSE & CONCEPTION .....	7
IV.1) CAS D'UTILISATION .....	7
IV.2) DIAGRAMME DE COMPOSANTS .....	8
IV.3) DIAGRAMME DES PACKAGES .....	9
IV.4) DIAGRAMME DE CLASSES .....	10
IV.5) DIAGRAMME DE SÉQUENCES .....	11
IV.5.1) Envoyer un message .....	11
IV.5.2) Rédiger un message .....	12
IV.5.3) Recevoir un message .....	13
IV.5.4) Envoyer un fichier .....	14
IV.5.5) Recevoir un fichier .....	15
IV.5.6) Joindre un fichier .....	16
IV.5.7) Changer présence .....	17
IV.5.8) Démarrer une conférence texte .....	18
IV.5.9) Afficher la liste des contacts .....	19
IV.5.10) Ajouter un contact .....	20
IV.5.11) Supprimer un contact .....	21
IV.5.12) Édition contact .....	22
IV.5.13) Importer contact par QRCode .....	23
IV.6) PROTOCOLE DU QR-CODE .....	24

## I) DOMAINE D'APPLICATION

### I.1) OBJECTIFS DU SYSTÈME

L'objectif de ce projet est de fournir une application de Chat destiné à la plateforme Android utilisant le protocole XMPP/Jabber.

Ce projet contient une application cliente qui doit être installée sur l'appareil de l'utilisateur, un serveur XMPP doit aussi être installé, nous utiliserons OpenFire.

Cette application de chat permet aux utilisateurs d'accéder, par le biais d'un compte utilisateur inscrit sur le serveur XMPP, à leur messagerie instantanée, ainsi qu'à leur carnet d'adresses, à la création de salons de discussion, ainsi qu'à l'import de contacts via logos QR-Code.

### I.2) INTERFACES

Cette application se présente sous la forme d'une interface simple et ergonomique, utilisant la navigation par fenêtres (ou activités).

A son point d'entrée, cette application comporte une fenêtre de connexion permettant une authentification anonyme ou par nom d'utilisateur et mot de passe.

Dans ce deuxième cas, l'application se synchronise avec le serveur Jabber/XMPP précédemment entré (utilisateur@serveur), pour récupérer la liste des utilisateurs et affecter la présence 'En Ligne'.

Pour accéder aux données et aux services XMPP une connexion TCP/IP est nécessaire.

### I.3) CONTRAINTES DE CONCEPTION

Plusieurs contraintes provenant de différentes sources sont à prendre en compte dans la phase de conception du système. Ci-dessous, un récapitulatif des contraintes imposées par ces sources :

- Le langage JAVA doit être utilisé pour l'application, en utilisant l'environnement Android Development Tools (ADT – Eclipse)

- Un serveur XMPP (ex : OpenFire) doit être installé sur un PC pour pouvoir s'y connecter.
- La phase de conception n'est démarrée qu'après validation du modèle conceptuel de données.
- Les règles de mise en page des documents et de codage doivent être respectées tout au long de la conception.
- Les commentaires contenus dans le code, l'interface utilisateur ainsi que le manuel d'utilisation doivent être rédigés en français.

## II) DOCUMENTS DE RÉFÉRENCE

Les documents suivants sont à utiliser en référence avec la lecture de ce document :

- Le cahier des charges dans sa version finale, qui contient les spécifications initiales des exigences du client. Référence CDCoo\_Nom\_Version
- La maquette des écrans, fenêtres et menus établie dans le dossier de spécifications.
- Le dossier de spécifications, qui contient les spécifications techniques et graphiques nécessaires à la correspondance du produit avec le besoin.
- Le plan d'assurance qualité logiciel, établi en vue d'assurer un niveau de qualité des documents et productions logicielles correspondant aux attentes du client.

### III) NORMES, STANDARDS & OUTILS

#### III.1) ENVIRONNEMENT & OUTILS DE DÉVELOPPEMENT

Le matériel de développement utilisé est une machine préparée pour chaque développeur, équipée du système d'exploitation Windows 8.1 Professionnel, ainsi que d'un processeur et d'une quantité de mémoire vive jugée suffisante pour pouvoir effectuer un développement utilisant l'environnement ADT (Eclipse) de manière optimale.

Pour les tests/débogs de l'application, il est possible d'utiliser une VM (AVT [Officiel] ou Genymotion), ainsi qu'un appareil android en mode développeur.

Le serveur XMPP est Openfire et doit être installé et configuré sous une GNU/Linux Debian 7.

La gestion de la configuration, organisation des documents et fichiers sources est effectuée sur un FTP (Backup), un SVN (code) et un espace de stockage Google Drive (tout), ces trois outils permettant une très grande sécurisation des fichiers (sauvegarde multiple).

#### III.2) NOTATIONS UTILISÉES

La terminologie utilisée dans le projet Jabboid est disponible dans le dossier de spécifications. (Partie PAQL)

Les acronymes qui sont les plus couramment utilisés sont CDC (Cahier des Charges), SPC (Dossier de Spécifications), PAQ (Plan d'Assurance Qualité), DCP (Dossier de Conception) et PT (Plan de Tests).

#### III.3) STANDARDS DE PROGRAMMATION

- Commentaires // pour une ligne, /\*... \*/ pour plusieurs lignes
- Commentaires, nom de variables, méthodes en anglais
- Constantes en majuscules

Ex: MYCONST

- Les noms des variables/méthodes privées seront en minuscules, tandis que la première lettre d'une variable/méthode publique sera en majuscule.

Et les variables composées de plusieurs mots auront la première lettre de chaque mot en majuscule.

**Ex:** Private int example, Public int DoExample(), plusGrandNombrePremier.

- Le nom des variables doit contenir le type de celle-ci.

**Ex:** int iCpt, string strFoo, float fMean.

- Les variables, méthodes, etc, doivent avoir le nom le plus descriptif possible. Les mots doivent être écrits en entier plutôt qu'avec des abréviations, sauf dans le cas de certains mots clés très courants comme Nbr, Max ou Min

**Ex:** Utiliser Counter au lieu de i, PositionX au lieu de x, et surtout pas de variables Temp, Toto, FooBar...

- Les accolades doivent être placées au début du bloc comme ci-dessous :

```
If (someCondition)
{
DoSomething();
}
```

- Les indentations doivent être des caractères de tabulation et sont obligatoires.

- Les lignes doivent avoir une largeur maximale de 120 caractères de façon à faciliter la lecture et l'impression du code.

- Les primitives : if, else, while, for et do doivent obligatoirement être suivies d'un bloc entre accolades, même s'ils ne contiennent qu'une instruction ou qu'ils sont vides.

- On doit utiliser la méthode permissivité minimale. Nous devons éviter les variables globales, les données membres publiques, etc.

Les variables globales doivent être utilisées le moins possible et, le cas échéant, doivent être encapsulées dans un seul objet global.

- Les fichiers et les méthodes devront inclure des entêtes tels que décrits ci-dessous. (Compatible JAVADOC):

## ENTÊTE DE MÉTHODES:

```
/**
 * Validates a chess move.
 *
 * Use {@link #doMove(int, int, int, int)} to move a piece.
 *
 * @param theFromFile file from which a piece is being moved
 * @param theFromRank rank from which a piece is being moved
 * @param theToFile   file to which a piece is being moved
 * @param theToRank   rank to which a piece is being moved
 * @return            true if the move is valid, otherwise false
 */
boolean isValidMove(int theFromFile, int theFromRank, int theToFile,
int theToRank)
{
    ...
}
```

## ENTÊTE DE CLASSES:

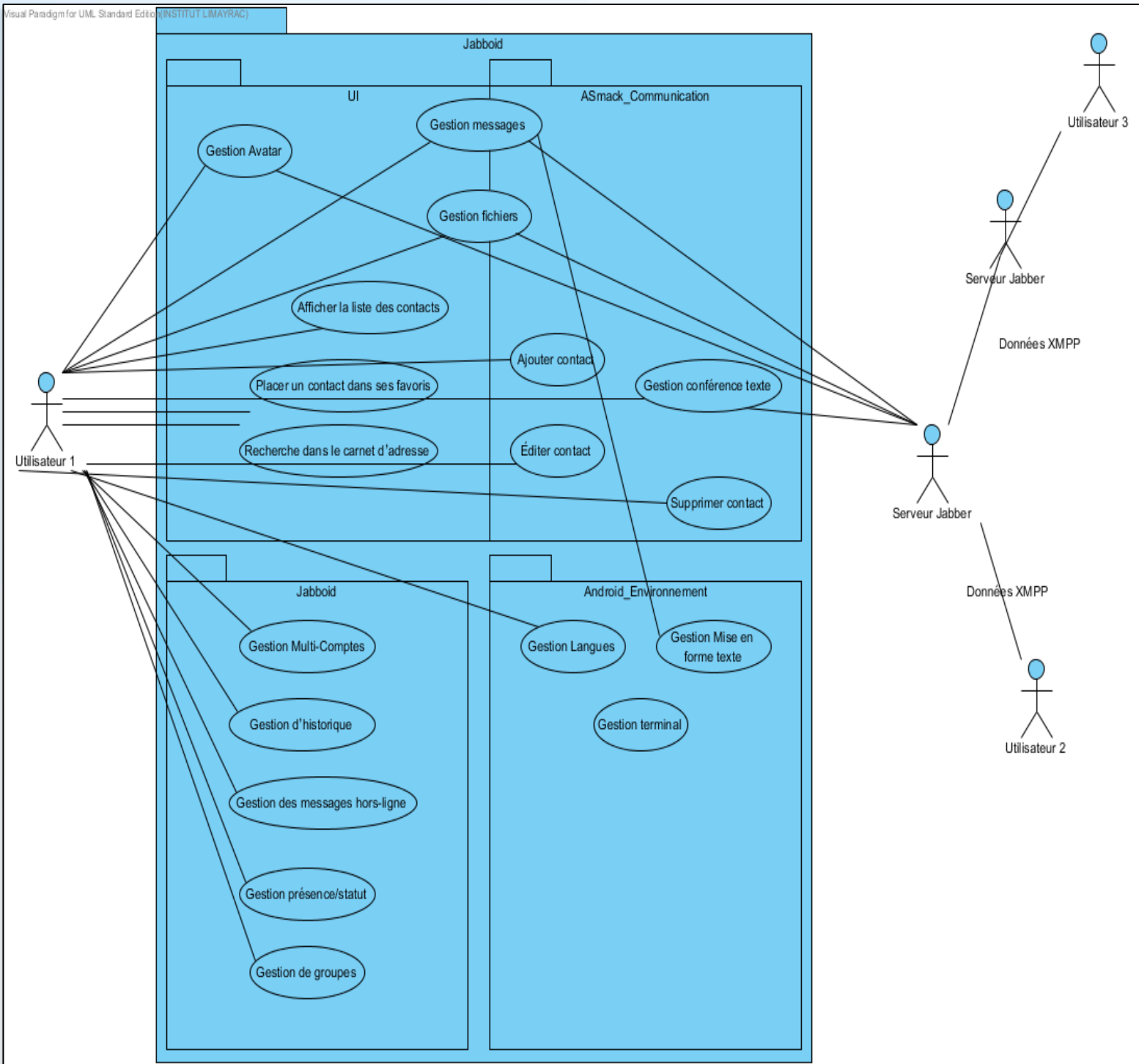
```
/**
 * @author      Firstname Lastname <address @ example.com>
 * @author      Firstname Lastname <address @ example.com>
 * @version     1.6                (current version number of
program)
 * @since      2010-03-31        (the version of the package this class
 * was first added to).
 */
public class Test
{
    // class body
}
```



## IV) ANALYSE & CONCEPTION

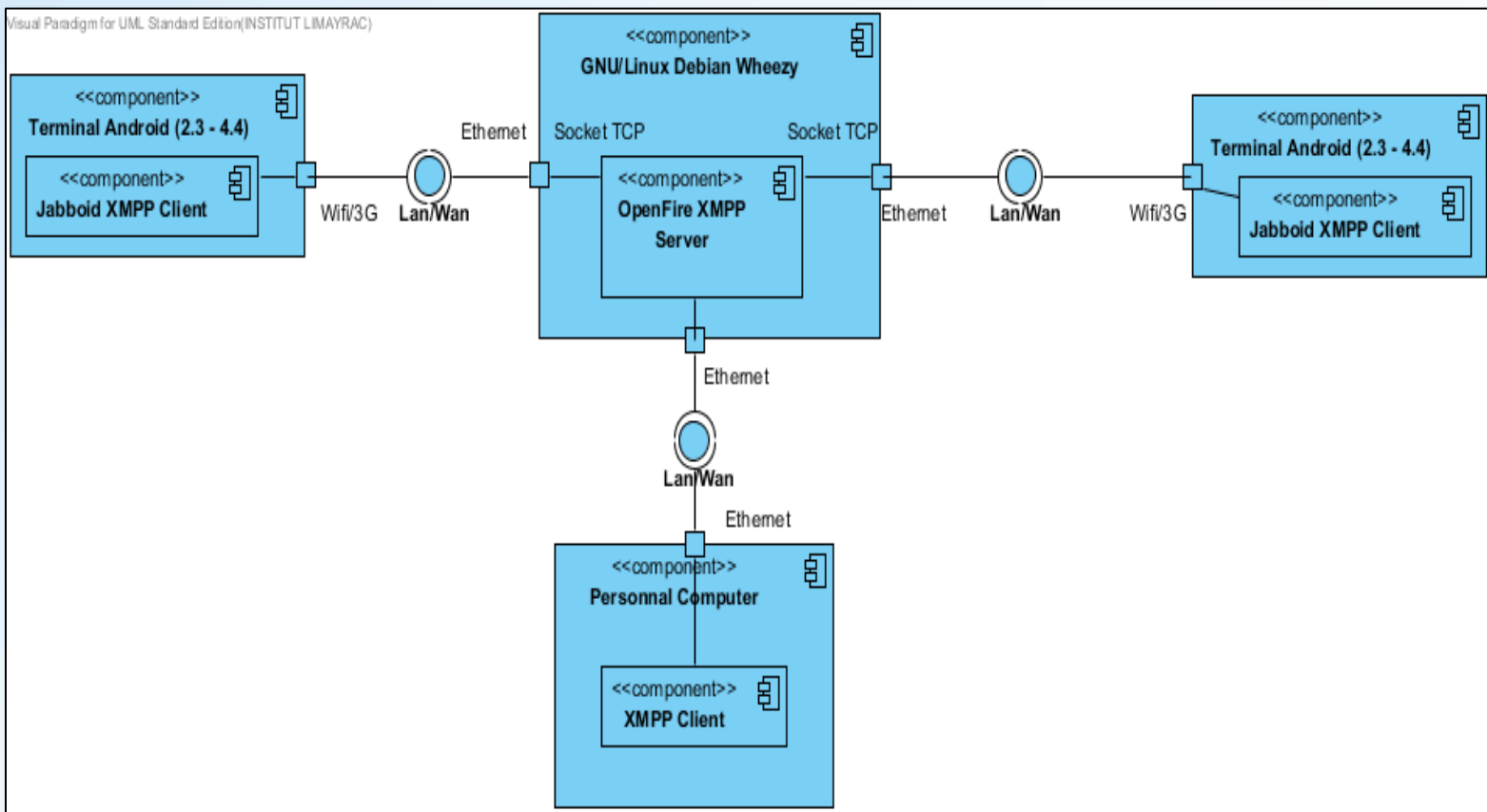
### IV.1) CAS D'UTILISATION

Dans cette partie, les cas d'utilisations des utilisateurs de l'application sont représentés.

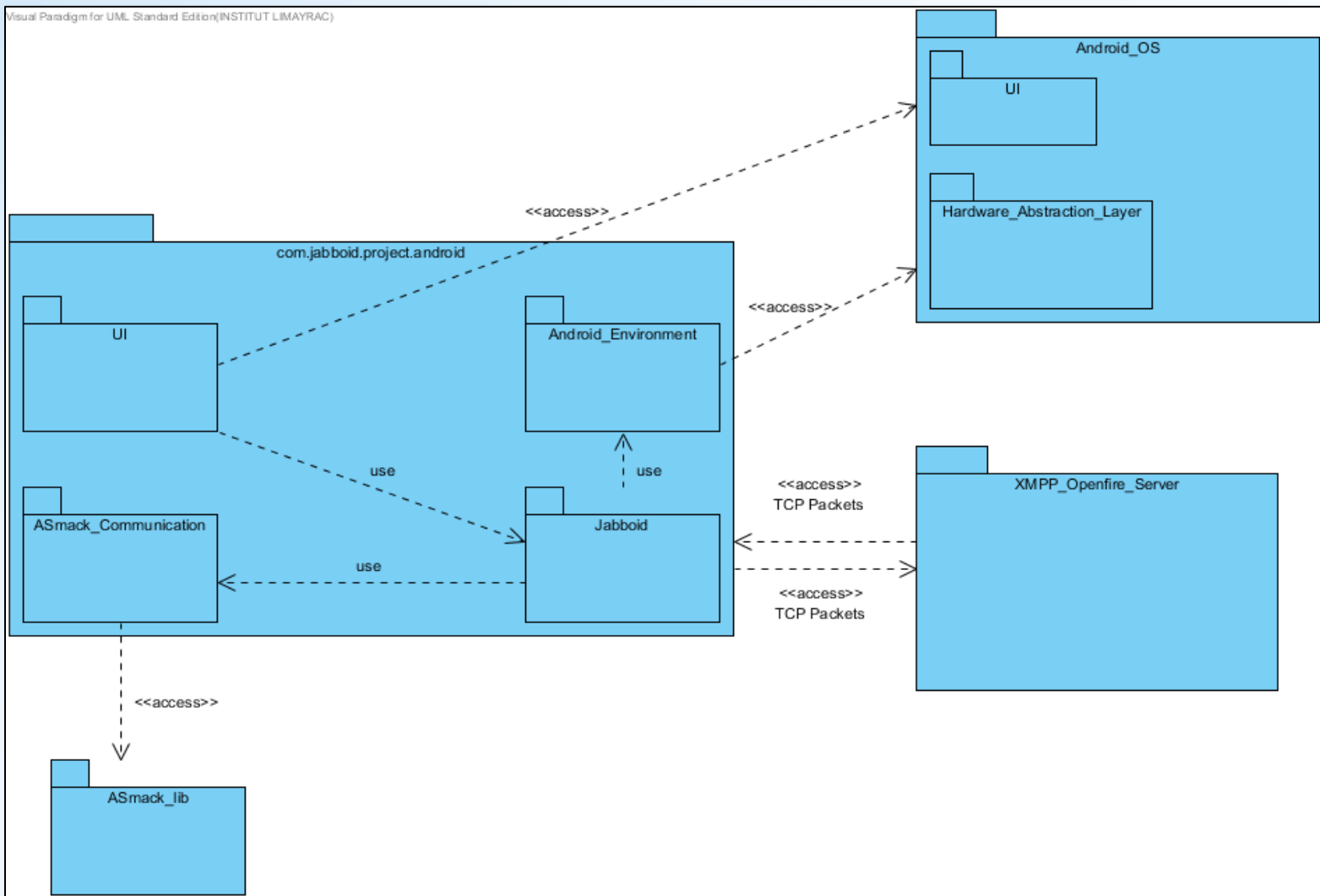


## IV.2) DIAGRAMME DE COMPOSANTS

Visual Paradigm for UML Standard Edition (INSTITUT LIMAYRAC)

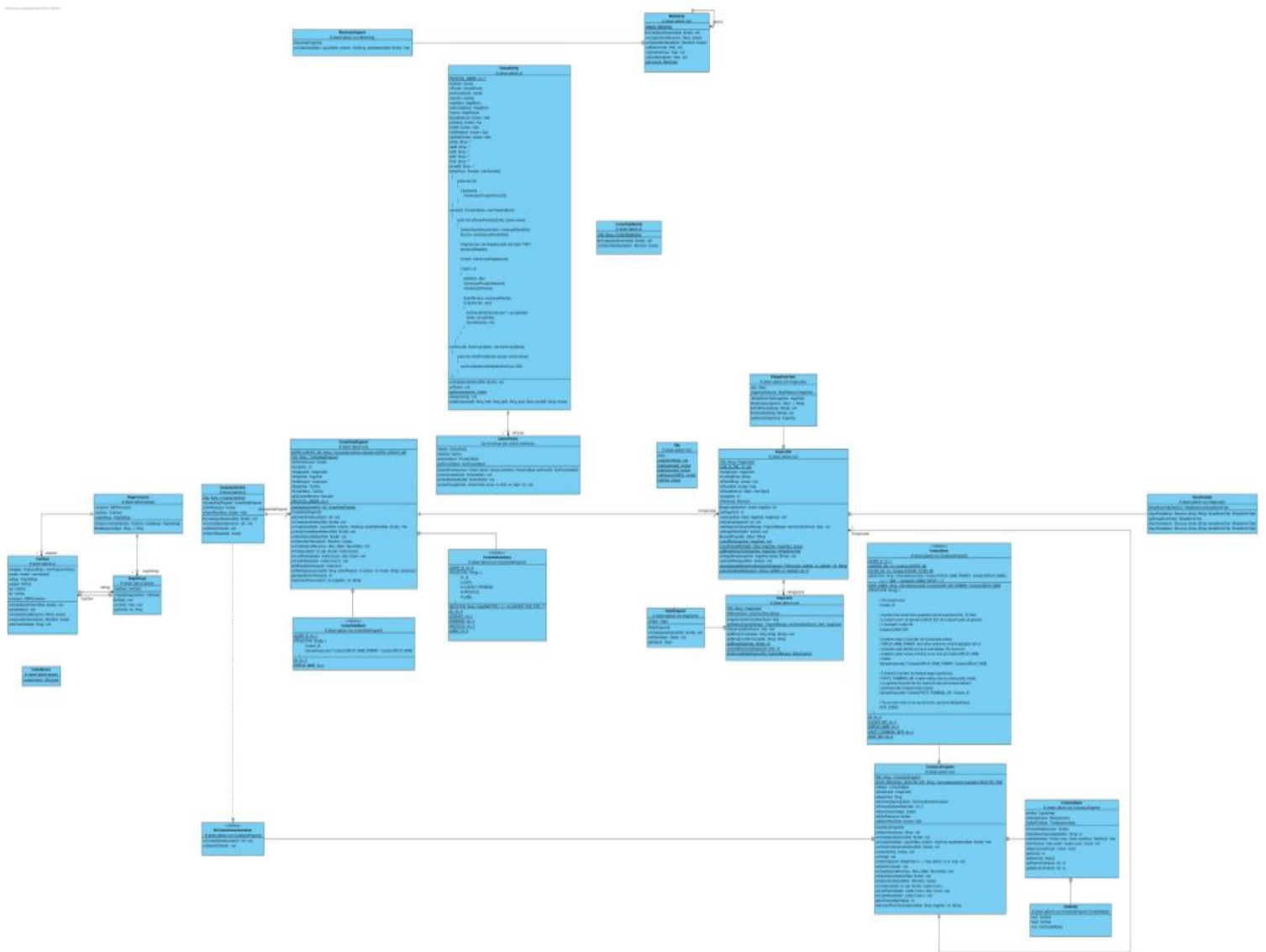


### IV.3) DIAGRAMME DES PACKAGES



Nom	Documentation
Android_OS	Paquet représentant toutes les API disponibles pour le système d'exploitation Android
UI (côté Jabboid)	Sous-paquet regroupant les classes (ou activités) de la partie Interface Utilisateur.
Hardware_Abstraction_Layer	Sous-paquet regroupant la couche d'abstraction matérielle (lien logiciel -> matériel)
com.jabboid.project.android	Paquet général de l'application Jabboid
Android_Environment	Sous-Paquet regroupant les classes qui communiquent avec l'environnement Android (vibration, écran, LED...)
XMPP_Openfire_Server	Paquet représentant le serveur XMPP
ASmack_Communication	Sous-paquet regroupant les classes communiquant avec la librairie ASmack.
Jabboid	Sous-paquet représentant les classes qui implémentent des fonctionnalités propres à Jabboid (ex : QR Code)
ASmack_lib	Paquet représentant la librairie ASmack

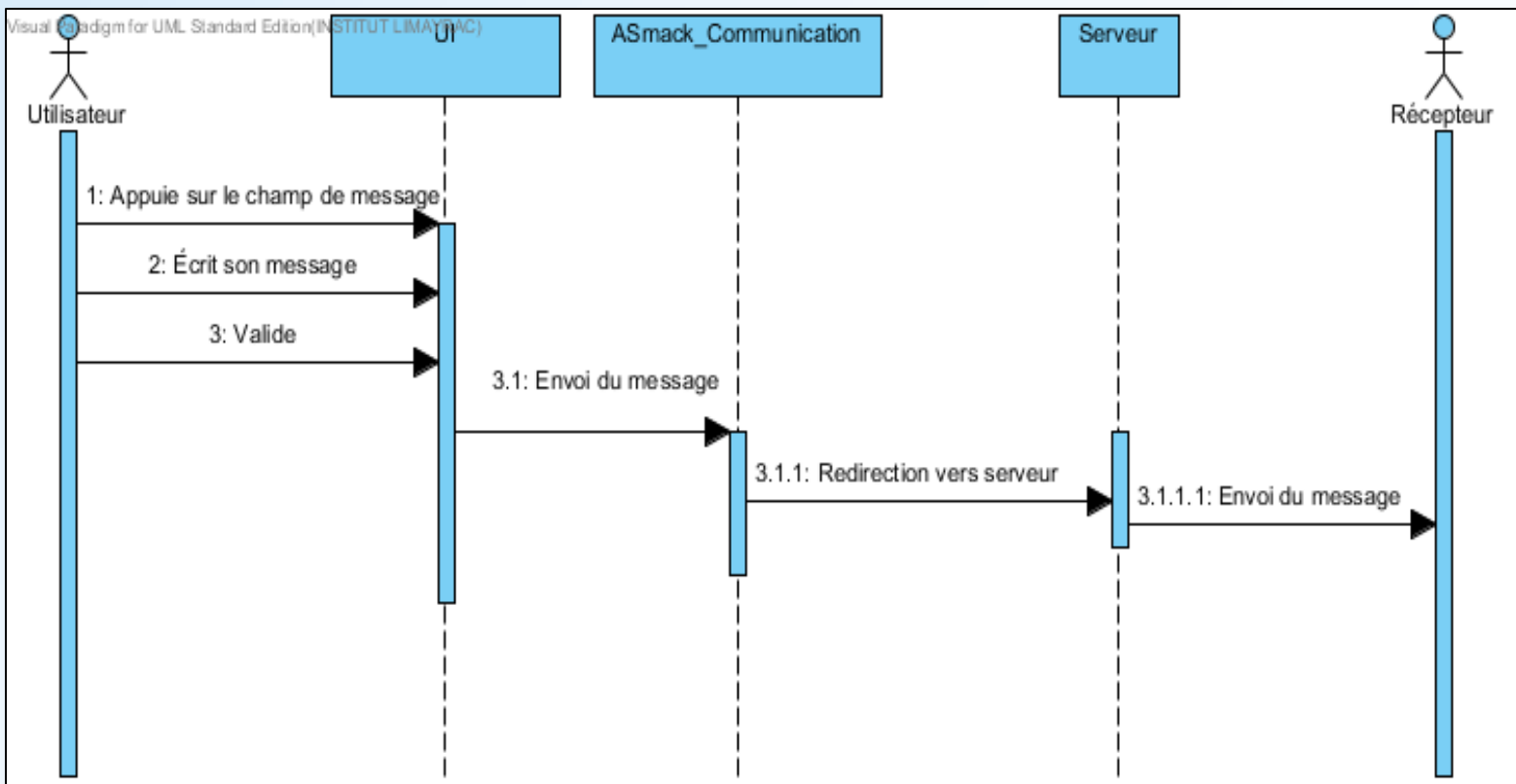
#### IV.4) DIAGRAMME DE CLASSES







Remarque : Voir taille réelle en annexes

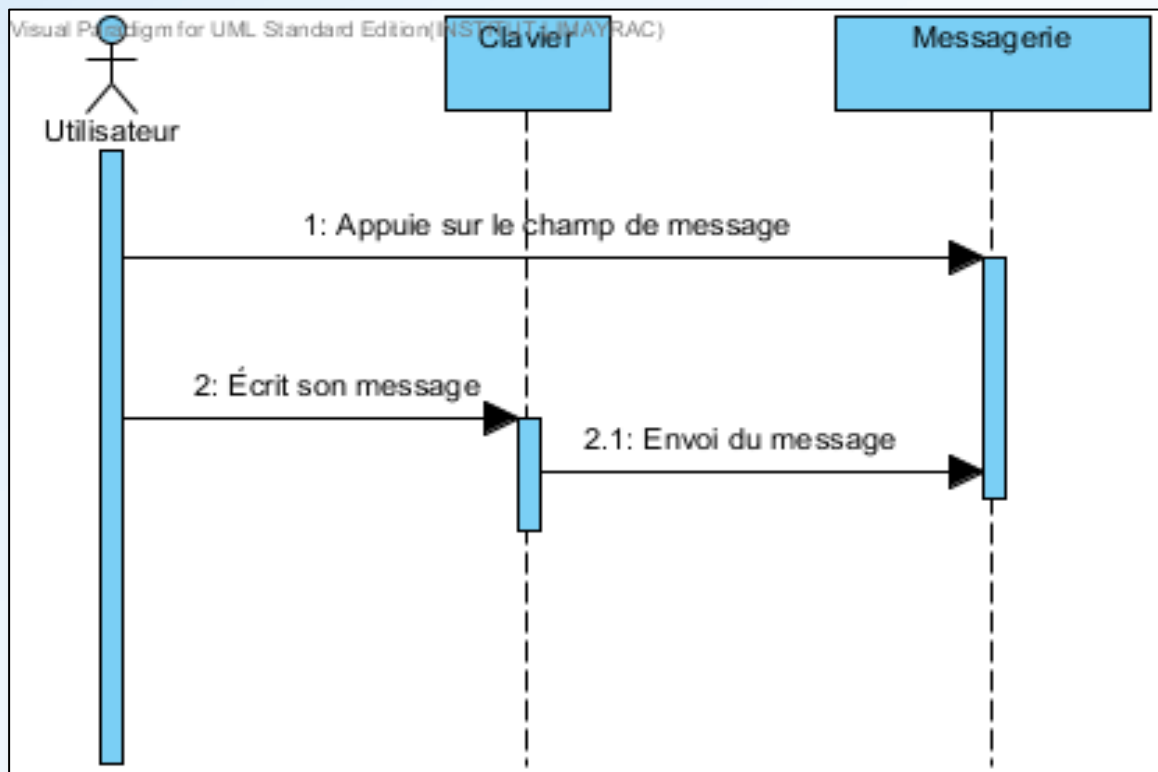
## IV.5) DIAGRAMME DE SÉQUENCES




### IV.5.1) Envoyer un message



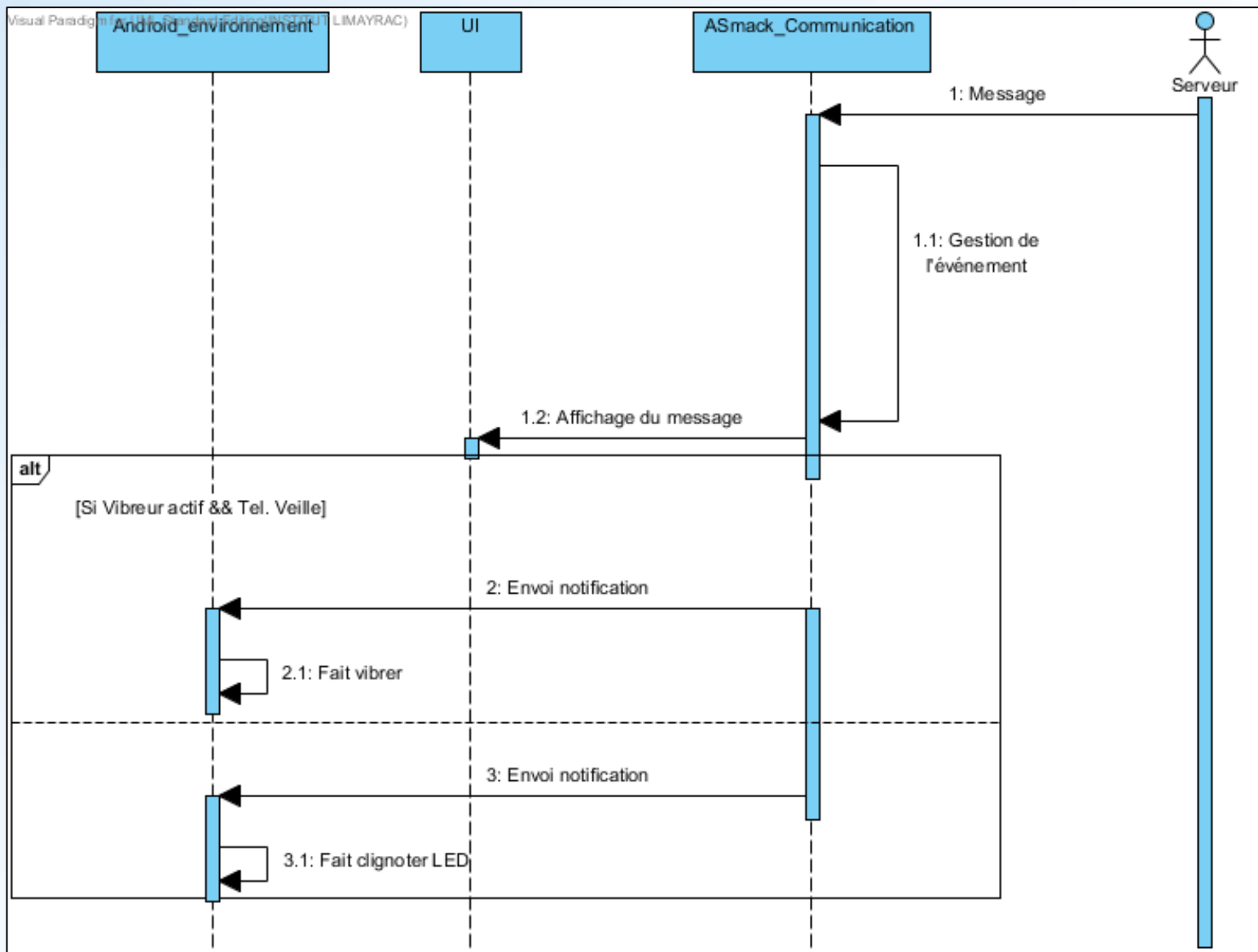
Nom	Documentation
 Utilisateur	Utilisateur qui veut envoyer un message.
 UI	Classe du paquet UI, qui affiche l'interface de chat.
 ASmack_Communication	Classe du paquet ASmack_Communication, qui s'occupe d'envoyer une trame XML au serveur.
 Serveur	Serveur relais XMPP





#### IV.5.2) Rédiger un message



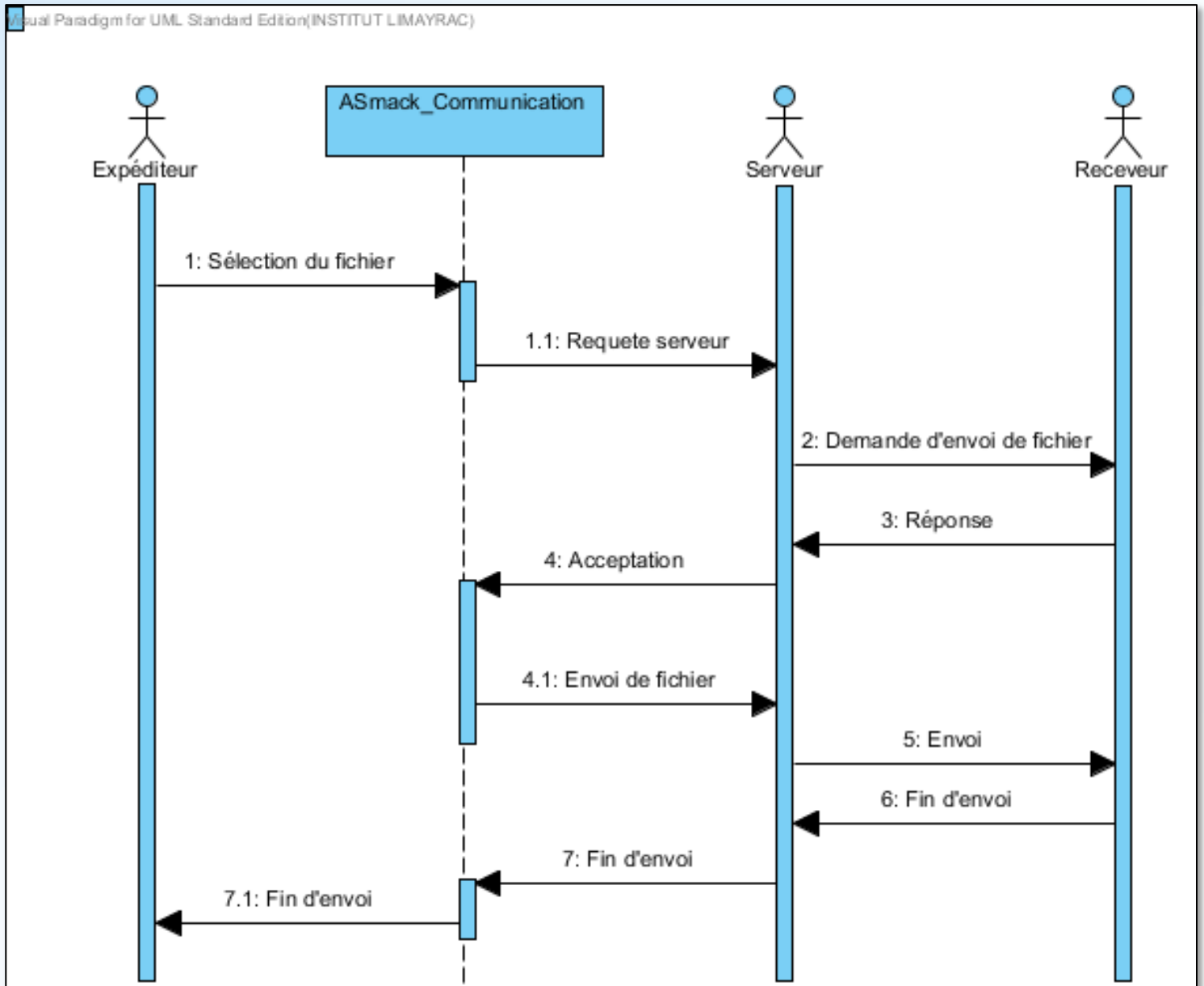
Nom	Documentation
 Utilisateur	Utilisateur qui souhaite rédiger un message.
 Messagerie	Activité permettant d'afficher ce que l'on tape.
 Clavier	Fonction Android affichant un clavier virtuel tactile.

### IV.5.3) Recevoir un message



Nom	Documentation
 Serveur	Serveur relais XMPP
 ASmack_Communication	Classe du paquet ASmack_Communication, qui s'occupe d'envoyer une trame XML au client.
 UI	Classe du paquet UI, qui s'occupe de lister les messages.
 Android_environnement	Classe du paquet Android_environnement, qui s'occupe en fonction d'un réglage utilisateur lorsque l'appareil est en veille, de faire vibrer l'appareil OU de le faire clignoter la LED de celui-ci.

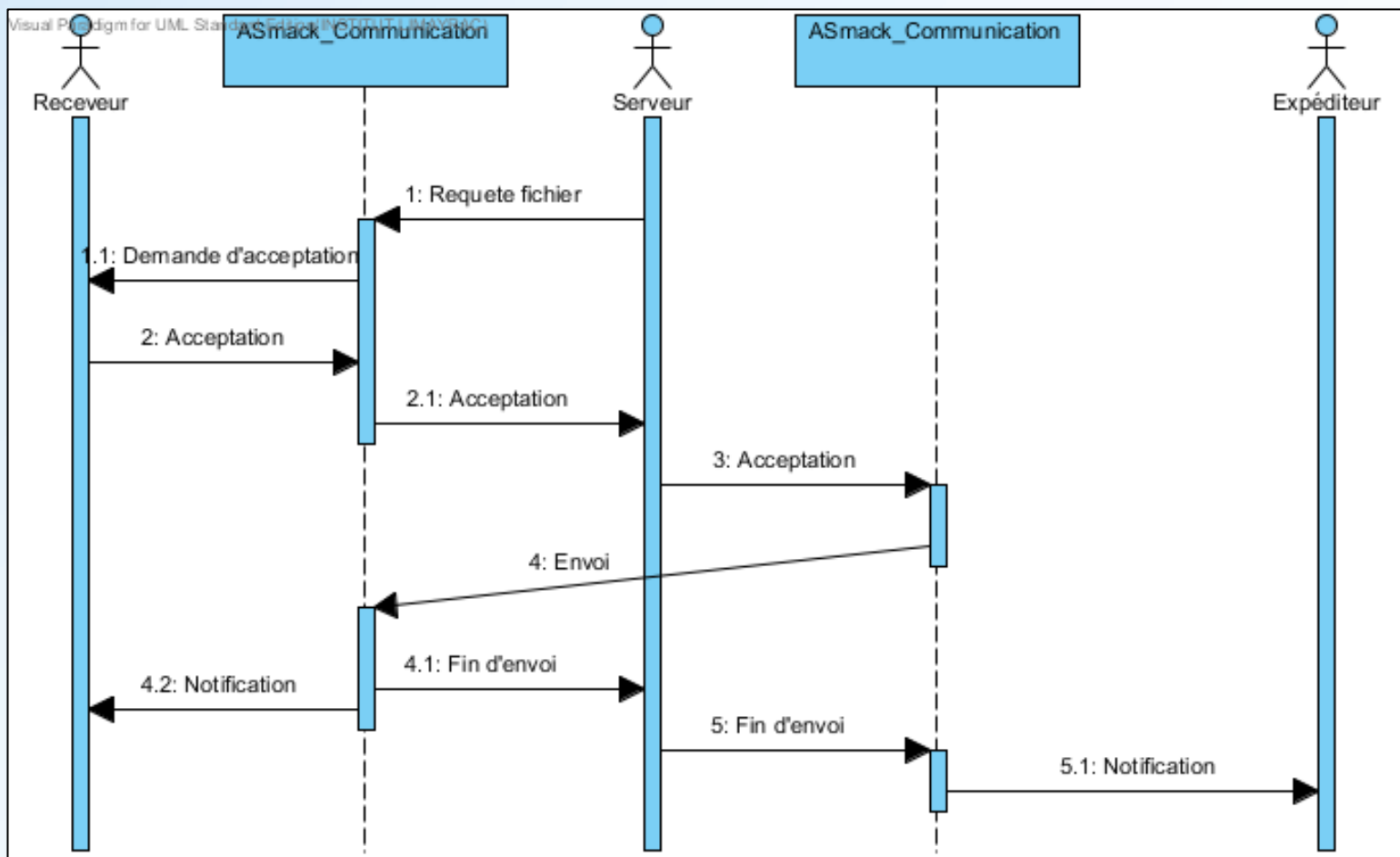
#### IV.5.4) Envoyer un fichier







Nom	Documentation
Expéditeur	Utilisateur souhaitant envoyer un fichier à son contact.
ASmack_Communication	Classe du paquet ASmack_Communication, qui s'occupe d'envoyer une trame XML au serveur.
Serveur	Serveur relais XMPP
Receveur	Utilisateur souhaitant (ou non) recevoir le fichier

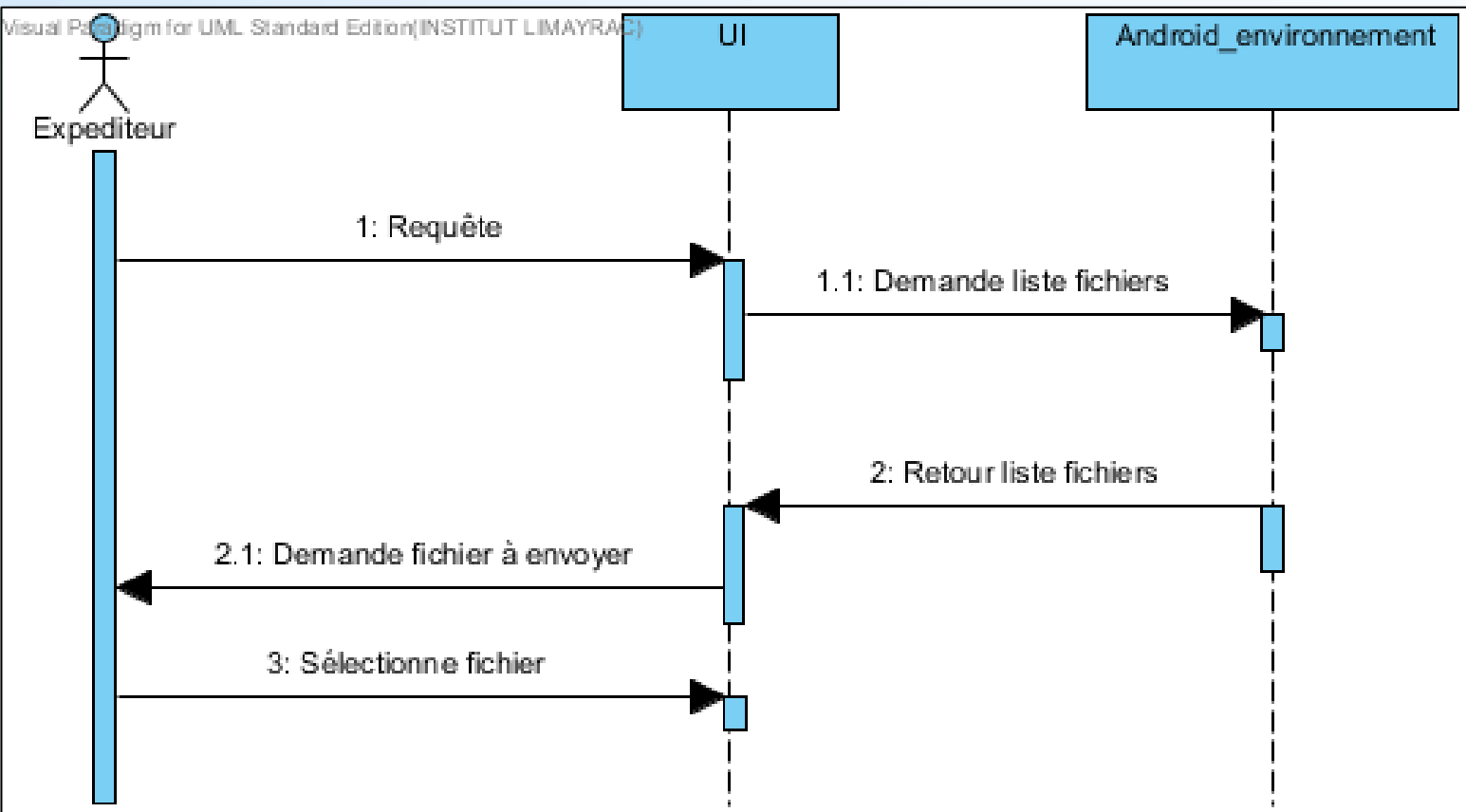





#### IV.5.5) Recevoir un fichier



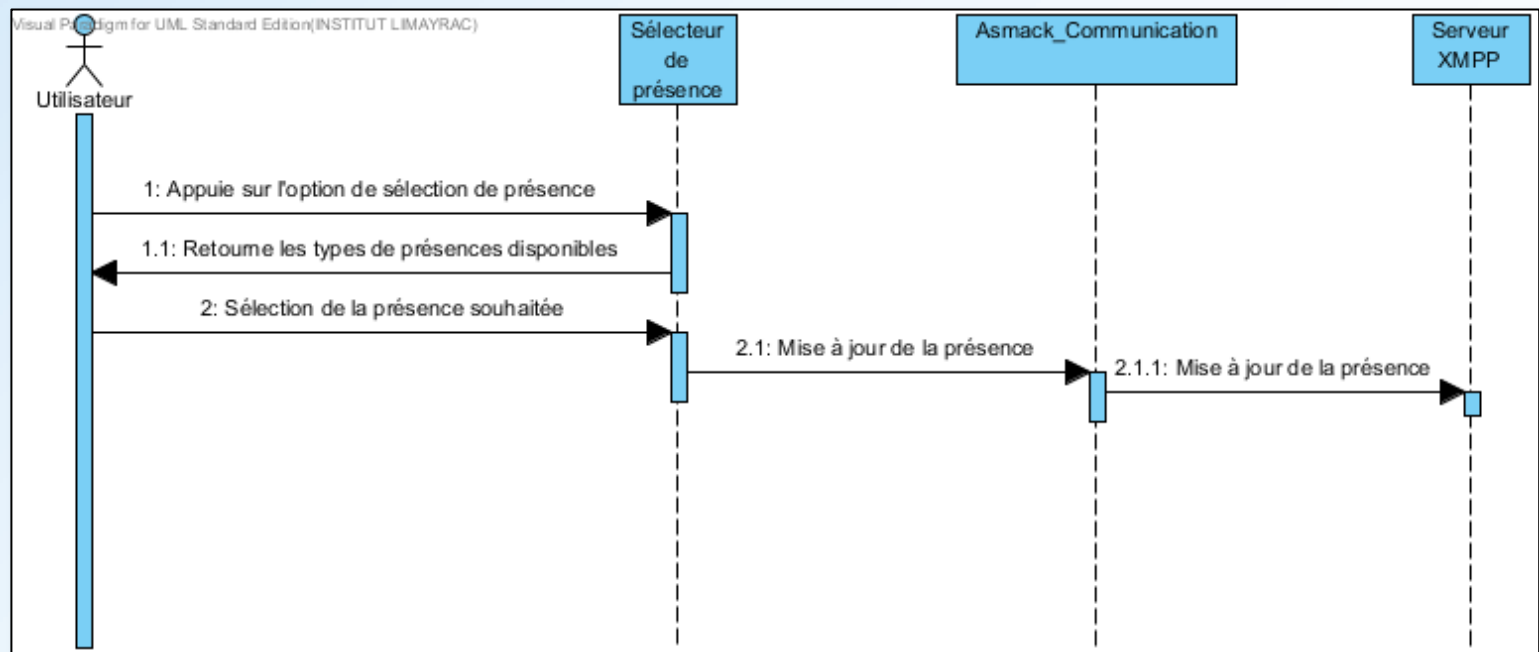
Nom	Documentation
 Serveur	Serveur relais XMPP
 ASmack_Communication	Classe du paquet ASmack_Communication, qui s'occupe d'envoyer une trame XML au serveur.
 Receveur	Utilisateur (Jabboid) souhaitant recevoir un fichier.
 Expéditeur	Utilisateur souhaitant envoyer le fichier.





#### IV.5.6) Joindre un fichier



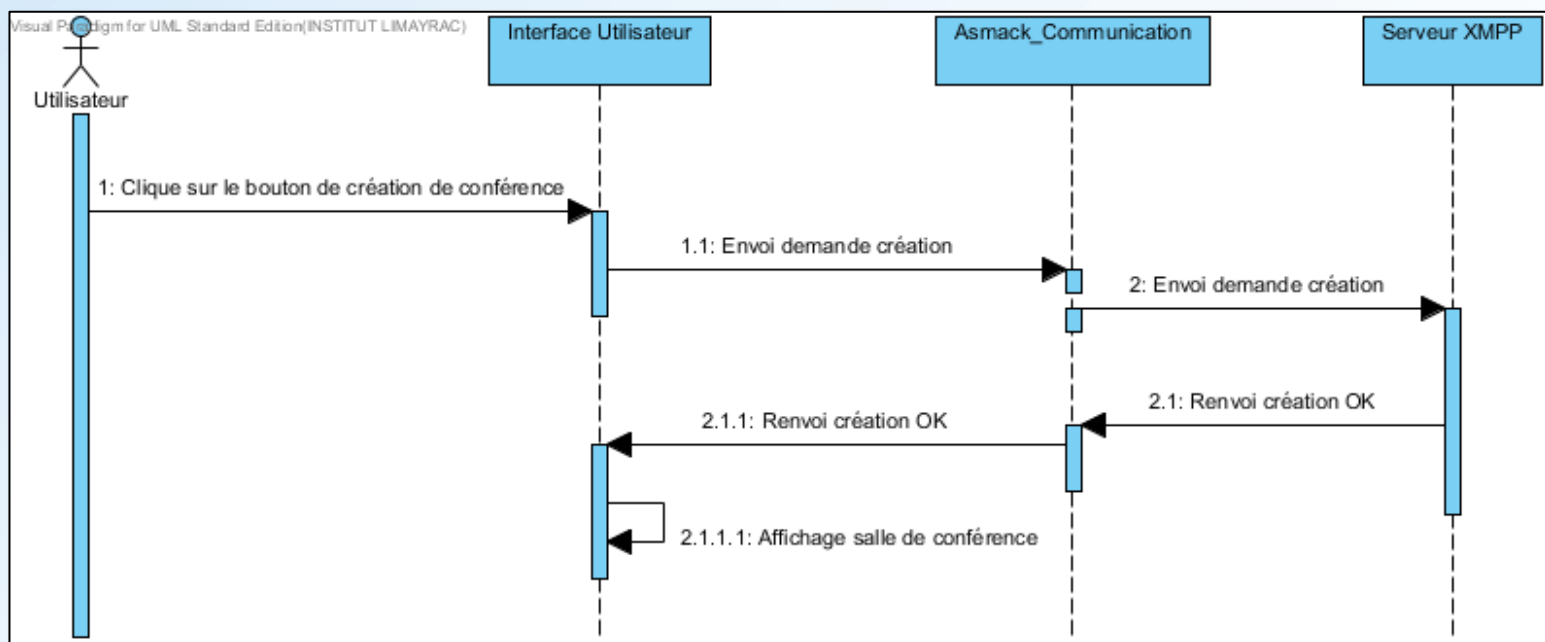
Nom	Documentation
 Expéditeur	Utilisateur souhaitant joindre un fichier (sélectionner pour envoi)
 UI	Classe du paquet UI affichant l'interface de jointe de fichier
 Android_environnement	Classe du paquet Android_environnement, renvoyant le contenu de la mémoire interne ou externe de l'appareil





#### IV.5.7) Changer présence



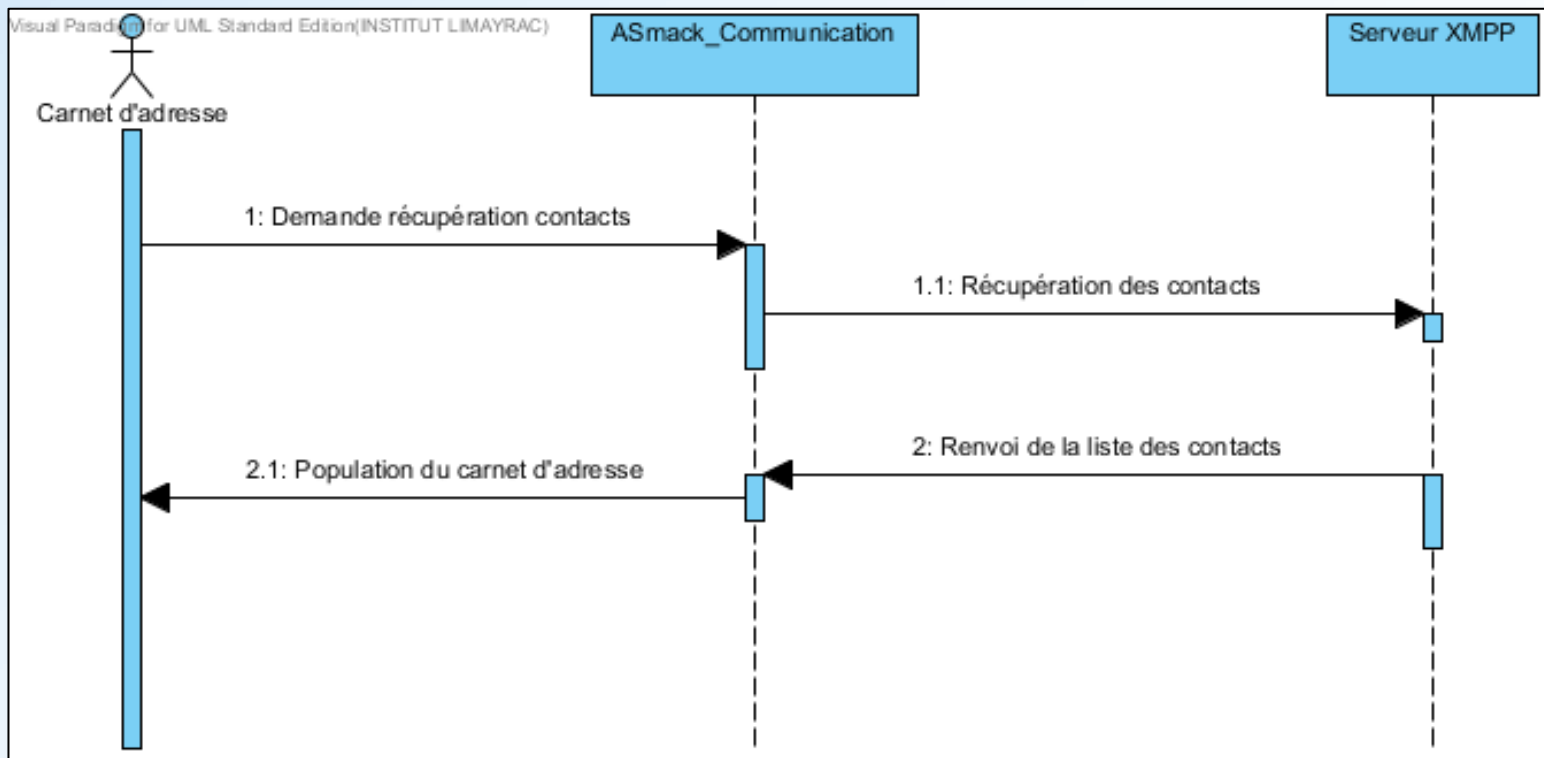
Nom	Documentation
 Utilisateur	Utilisateur souhaitant changer son statut de connexion (ou présence)
 Sélecteur de présence	Classe appartenant au paquet UI permettant d'afficher la sélection de diverses présences
 Asmack_Communication	Classe du paquet ASmack_Communication, qui s'occupe d'envoyer une trame XML au serveur.
 Serveur XMPP	Serveur relais XMPP




#### IV.5.8) Démarrer une conférence texte



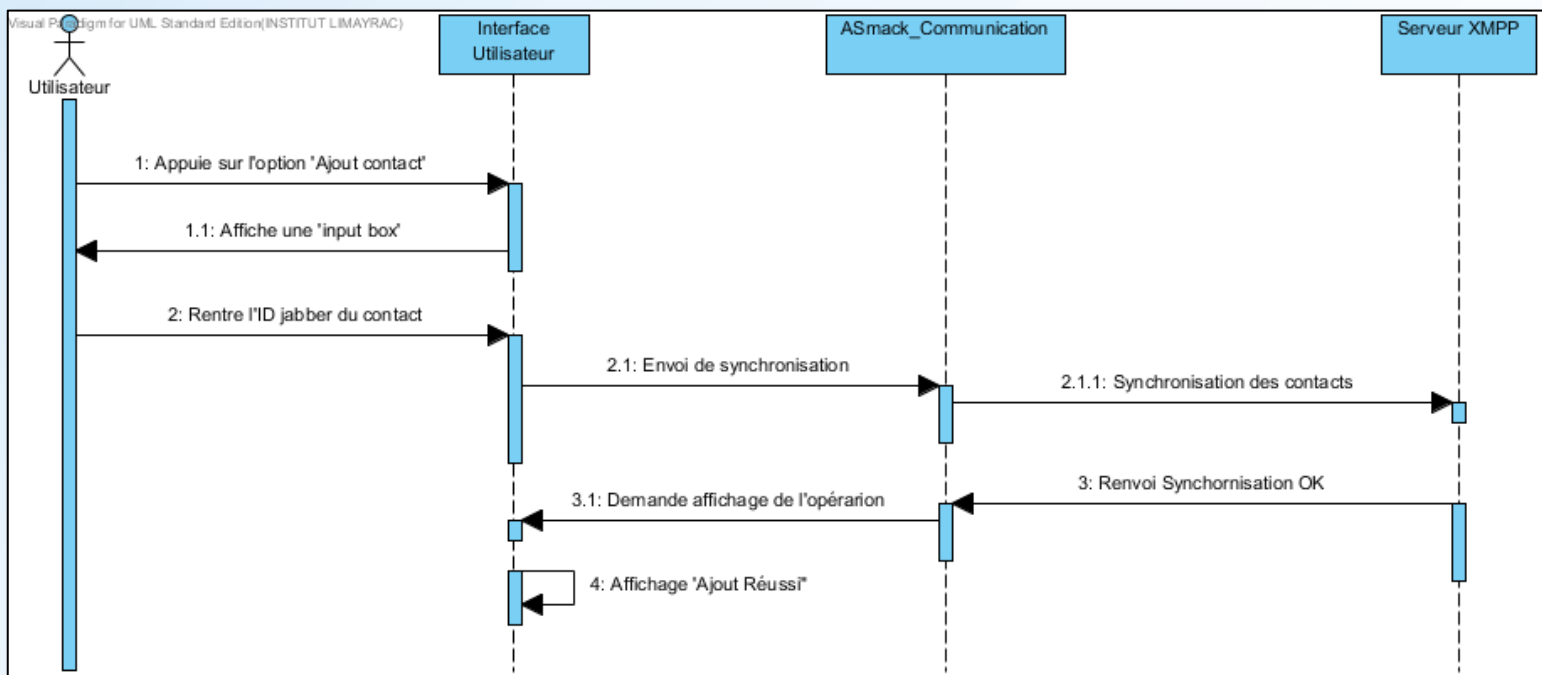
Nom	Documentation
 Utilisateur	Utilisateur souhaitant démarrer une conférence textuelle
 Interface Utilisateur	Classe du paquet UI permettant d'afficher le contenu de la conférence de texte
 Asmack_Communication	Classe du paquet ASmack_Communication, qui s'occupe d'envoyer une trame XML au serveur.
 Serveur XMPP	Serveur relais XMPP





#### IV.5.9) Afficher la liste des contacts



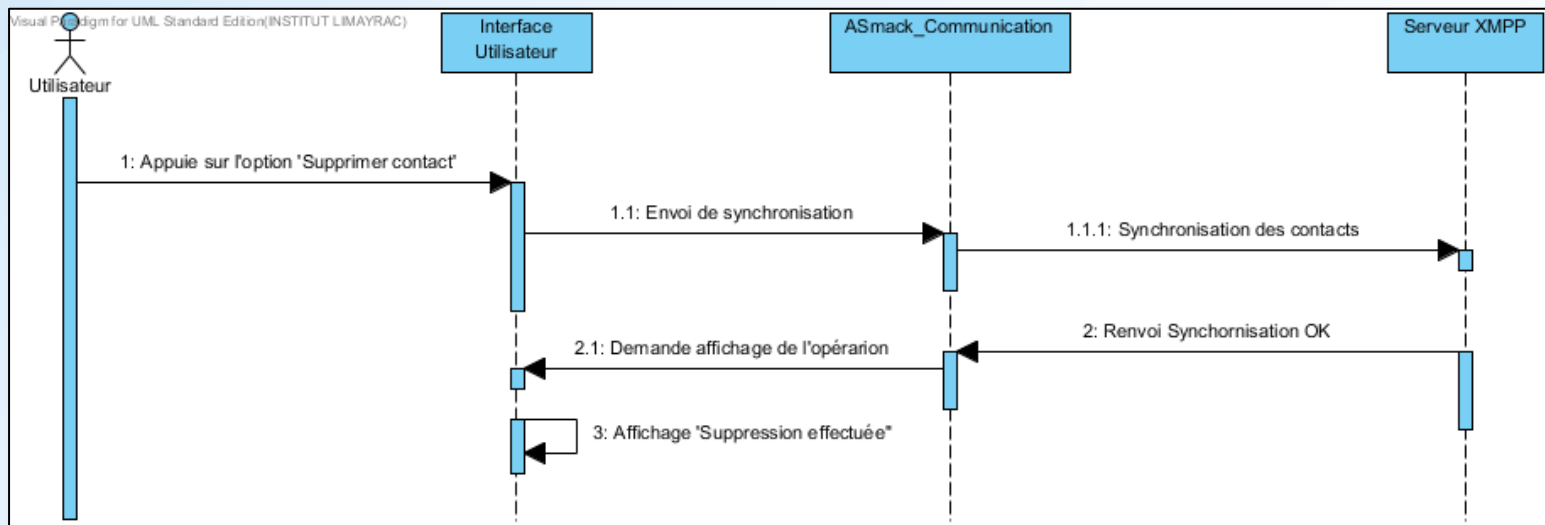
Nom	Documentation
 Carnet d'adresses	Classe du paquet UI permettant de lister les contacts de l'utilisateur
 ASmack_Communication	Classe du paquet ASmack_Communication, qui s'occupe d'envoyer une trame XML au serveur.
 Serveur XMPP	Serveur relais XMPP





#### IV.5.10) Ajouter un contact



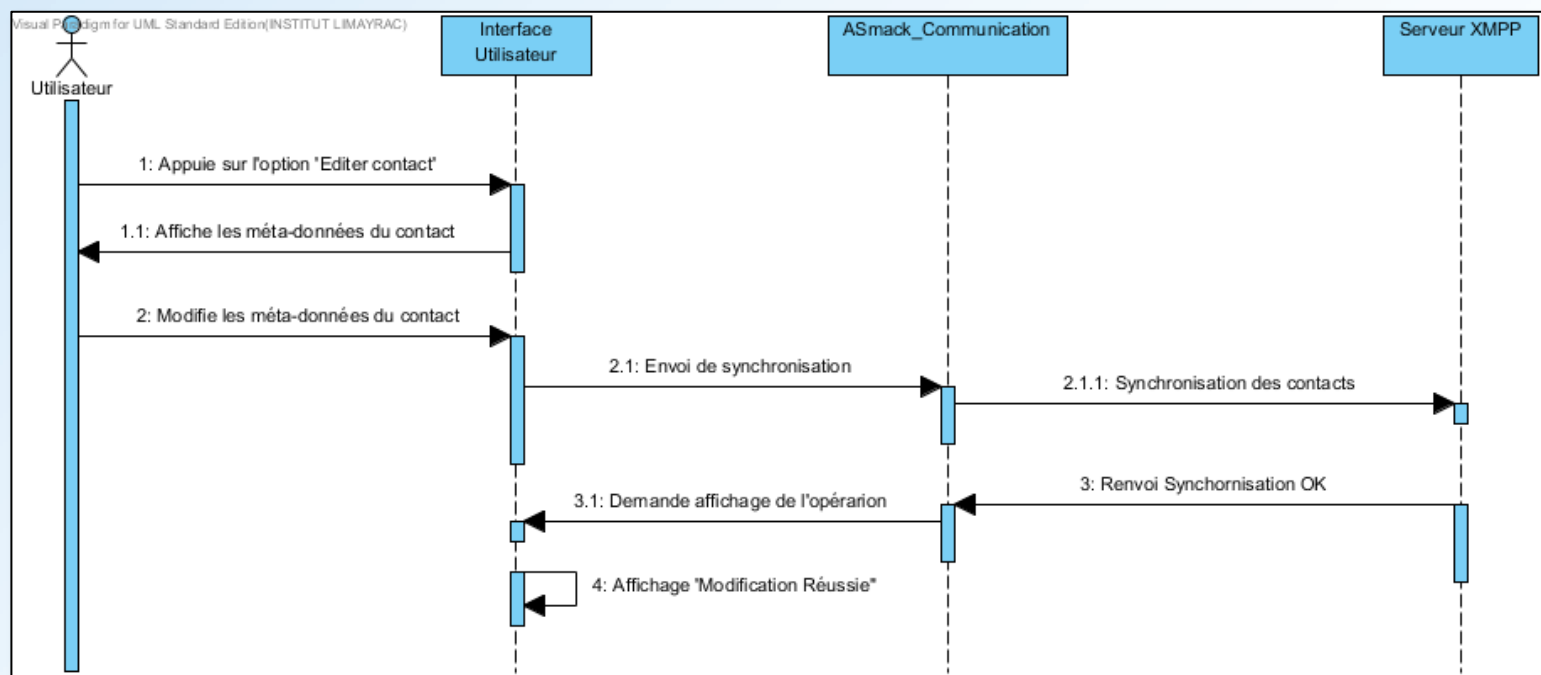
Nom	Documentation
 Utilisateur	Utilisateur souhaitant ajouter un contact
 Interface Utilisateur	Classe du paquet UI, affichant le menu d'ajout de contact
 ASmack_Communication	Classe du paquet ASmack_Communication, qui s'occupe d'envoyer une trame XML au serveur.
 Serveur XMPP	Serveur relais XMPP





#### IV.5.11) Supprimer un contact



Nom	Documentation
 Utilisateur	Utilisateur souhaitant supprimer un contact
 Interface Utilisateur	Classe du paquet UI permettant d'afficher le menu de suppression de contact
 ASmack_Communication	Classe du paquet ASmack_Communication, qui s'occupe d'envoyer une trame XML au serveur.
 Serveur XMPP	Serveur relais XMPP

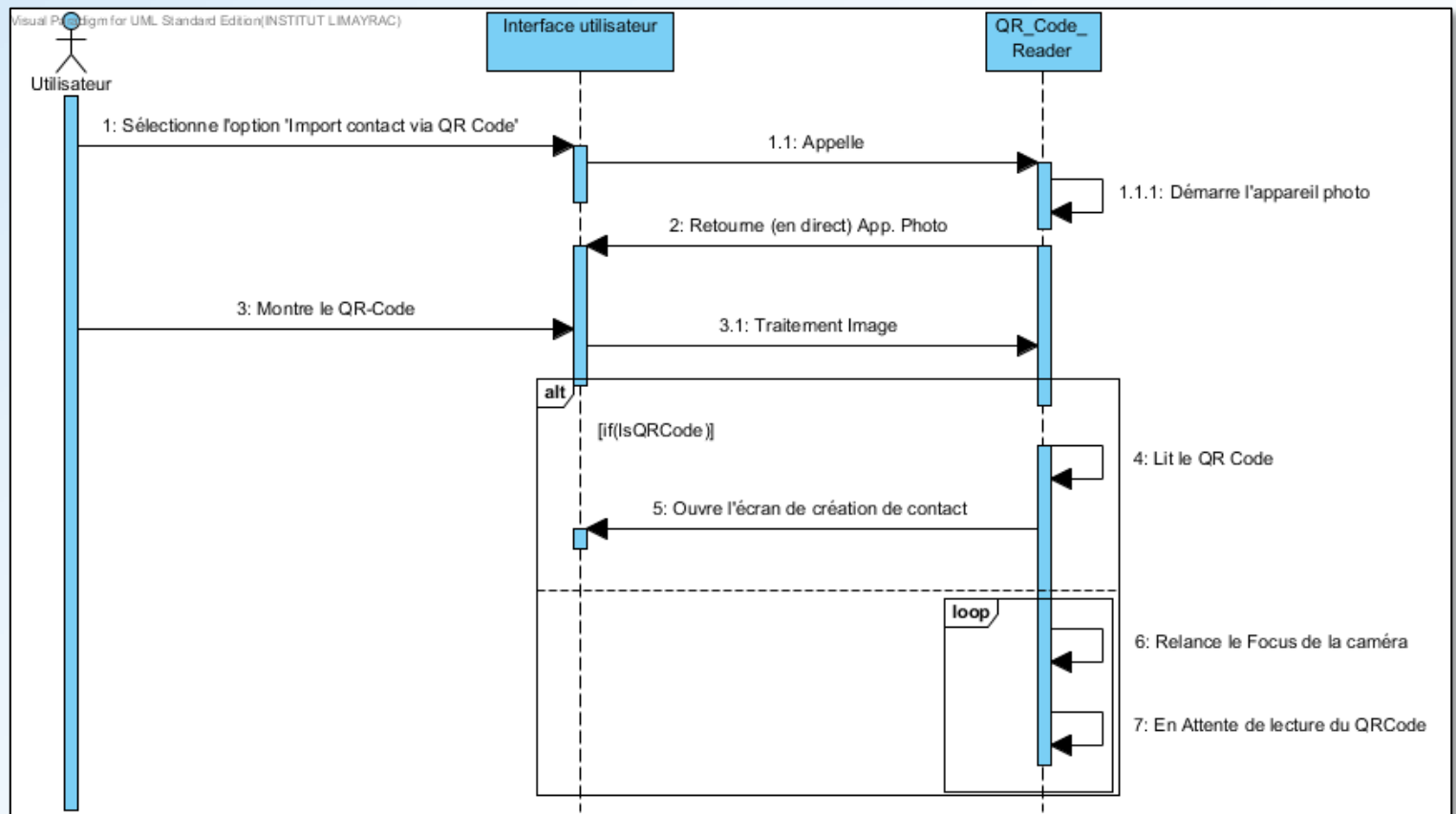
#### IV.5.12) Édition contact








Nom	Documentation
 Utilisateur	Utilisateur souhaitant éditer un contact
 Interface Utilisateur	Classe du paquet UI permettant d'afficher une boîte de modification des métadonnées d'un contact.
 ASmack_Communication	Classe du paquet ASmack_Communication, qui s'occupe d'envoyer une trame XML au serveur.
 Serveur XMPP	Serveur relais XMPP



#### IV.5.13) Importer contact par QRCode



Nom	Documentation
 Utilisateur	Utilisateur souhaitant importer des contacts via QR Codes
 Interface utilisateur	Classe du paquet UI permettant d'afficher le contenu de la caméra ainsi que les diverses options de scan.
 QR_Code_Reader	Classe du paquet Jabboid, permettant d'utiliser l'appareil photo d'un appareil Android, et de capturer des QR Codes
 Loop	Cette boucle infinie relance le focus toutes les X secondes, permet de déflouter l'image en fonction de la distance
 if(!isQRCode)	Si QR Code détecté, on ouvre l'écran de création de contacts, sinon on continue à scanner

## IV.6) PROTOCOLE DU QR-CODE

Trame JSON convertie en QR-Code

```
{  
  "isJabboid": "true",  
  "LastN": "Jalabert",  
  "firstN": "Steven",  
  "jabID": "steven@jabber.fr",  
  "email": "steven@free.fr",  
  "phoneNB": "000000000"  
}
```



La trame JSON sera générée par le programme spécialement créé pour l'occasion : « JSON\_Jabboid\_Maker », Disponible sous Windows & Linux.

Un manuel d'utilisateur de l'application + génération du QR-Code est également disponible.