

ARCHITECTURE DU LOGICIEL

Version : 1.1
Date : 16-01-12
Rédigé par : Ouissem Hamdani
Relu par : Safae Rebani

MISES A JOUR

Version	Date	Modifications réalisées
0.1	16/12/11	Création
1.0	16/01/12	Relecture de Safae Rebani
1.1	10/04/12	Ajout du module Clé

1. Objet :

La conception est la phase du processus du développement d'un projet informatique dont l'objet principal est de décider la façon d'implémenter le système.

En effet, la conception est un processus créatif, c'est la phase la plus importante dans le cycle de développement d'un projet puisqu'elle permet de décrire de manière non ambiguë le fonctionnement futur du système, afin de faciliter la réalisation. Pendant cette phase, des décisions stratégiques sont à prendre afin de satisfaire aux exigences fonctionnelles et de qualité voulues pour le système.

2. Documents applicables et de référence

- Intitulé du projet ;
- Spécification technique de besoin (STB).

3. Terminologie et sigles utilisés

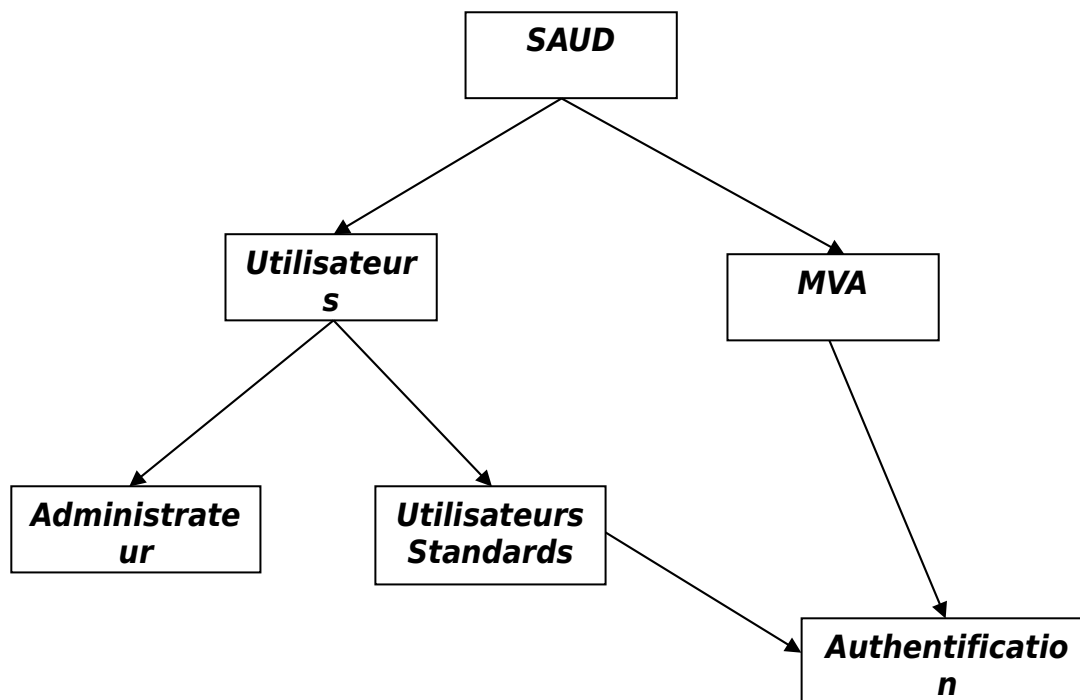
- SAUD : Site d'Authentification Unique et Décentralisée;
- AE : Authentification Externe, module permettant à des sites quelconques de proposer un service d'authentification unique et décentralisée;
- Utilisateur : Personne utilisant le service d'authentification unique, décentralisée et donc inscrit au sein du SAUD;
- Visiteur : Personne non encore inscrit sur le SAUD;
- Administrateur : Utilisateur spécial qui gère un SAUD;
- Identifiant : Mot permettant d'identifier l'utilisateur au sein du SAUD;
- Adresse SAUD : Correspond à la concaténation de l'identifiant, d'un dièse et du nom du SAUD [loginInterne:]loginExterne#saud.net, avec loginInterne le login du site quelconque et loginExterne le login dans le SAUD);
- API : (Application Interface Programming), ensemble de fonctions et de modules permettant de réaliser un but précis;
- RFC : (Requests for comments) est un document officiel décrivant les aspects techniques d'Internet ou de différent matériel informatique (routeurs, serveur DHCP...);
- PHP et JSP : (PHP Hypertext Preprocessor et Java Server Pages) sont des langages de programmation web coté serveur;

4. Architecture physique du matériel utilisé

Les machines utilisées pour développer seront principalement les machines disponibles dans les salles informatiques du département informatique. Nous programmerons sur le système d'exploitation GNU/Linux (distribution : Ubuntu 10.10).

5. Architecture statique du logiciel

5.1. Structure logique



- **Utilisateurs** : gère les utilisateurs. Ce constituant dérive Les utilisateurs standards et l'administrateur.
- **Clé** : module gérant les clés du SAUD, le stockage, la suppression et le test d'existence d'une clé est géré par ce module.
- **StdUtilisateurs** : représente tous les utilisateurs standards (tout internaute s'enregistrant sur le SAUD).
 - Attributs
 - - **id** (=login ?) : représente chaque utilisateur de façon unique.
 - - **mail** : mail de l'utilisateur.
 - - **question/answer** : couple question/réponse propre à l'utilisateur qui servira lors du contrôle d'identité en plus du mot de passe en cas de connexion avec une adresse IP inconnue.
 - - **addressIP** : ensemble (liste, tableau..) d'adresse IP affectée à l'utilisateur.
 - - **authenticate** : booléen définissant le statut d'authentification de l'utilisateur.
 - Méthodes
 - + **addressExist(IP)** : retourne vrai si IP appartient à l'ensemble **addressIP**, faux sinon.
 - + **AddIP(IP)** : ajoute IP à l'ensemble **addressIP**.

- **+ isAuthenticate()** : retourne vrai si l'utilisateur est connecté au SAUD, faux sinon.

➤ **Authentification :**

▪ Méthodes

- **+ isAuthenticate(Utilisateur)** : retourne vrai si l'utilisateur est connecté au SAUD, faux sinon.
- **+ setAuthenticate(Utilisateur, boolean : b)** : définit le statut de connexion de l'utilisateur à b.
- **+ isExist(Utilisateur)** : retourne vrai si l'utilisateur existe dans la BDD du SAUD, faux sinon.
- **+ checkIP(Utilisateur, IP)** : vérifie si IP appartient à l'attribut **adressIP** de l'utilisateur en question.
- **+ checkAnswer(Utilisateur)** : vérifie si le couple question/réponse de l'utilisateur est correct.
- **+ checkMail(Utilisateur)** : vérifie si le mail de l'utilisateur est bien correct.
- **+ addIP(Utilisateur, IP)** : ajoute IP à l'ensemble **adressIP** de l'utilisateur.

➤ **MVA (Module de Vérification d'Authentification) :**

▪ Méthodes

- **+ checkAuthentication(Utilisateurs, IP)** : vérifie si l'utilisateur est bien connecté au SAUD.
- **+ send(boolean)** : envoie vrai au site quelconque si l'utilisateur est connecté, faux sinon.

5.2. Justifications techniques

Nous avons choisie de développer l'application web en PHP car c'est le langage que maîtrise le plus de membres du groupe et le plus simple à appréhender pour ceux qui ne le connaissent pas.

De plus c'est un langage orienté objet qui permet de développer un projet de façon modulaire (classe, paquetage, ...).

6. Fonctionnement dynamique

Les diagrammes de séquences détaillés pour utilisateur :

➤ Création d'un compte avec Authentification Externe :

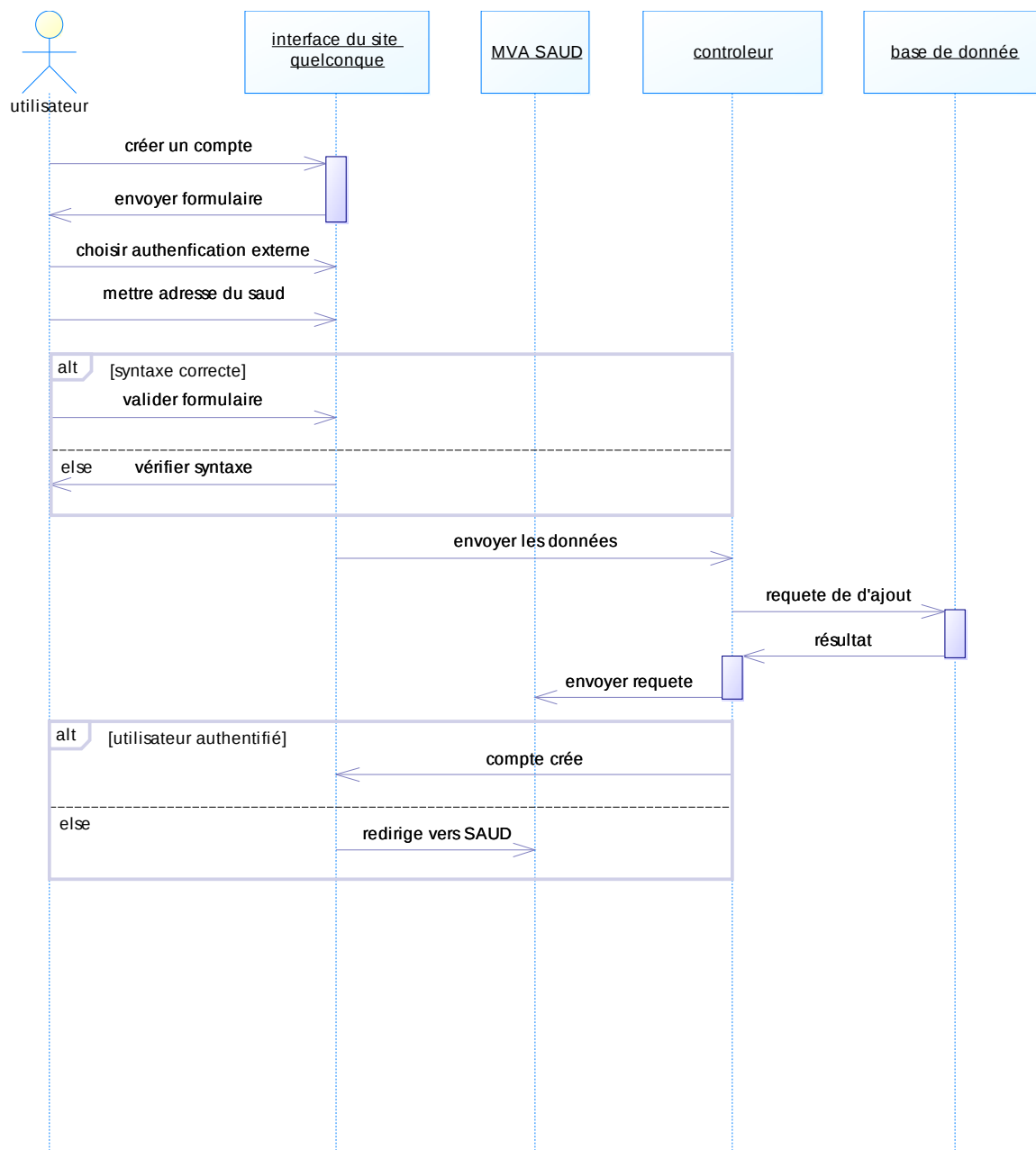


Diagramme de séquence « création d'un compte avec Authentification Externe »

➤ Création d'un compte utilisateur :

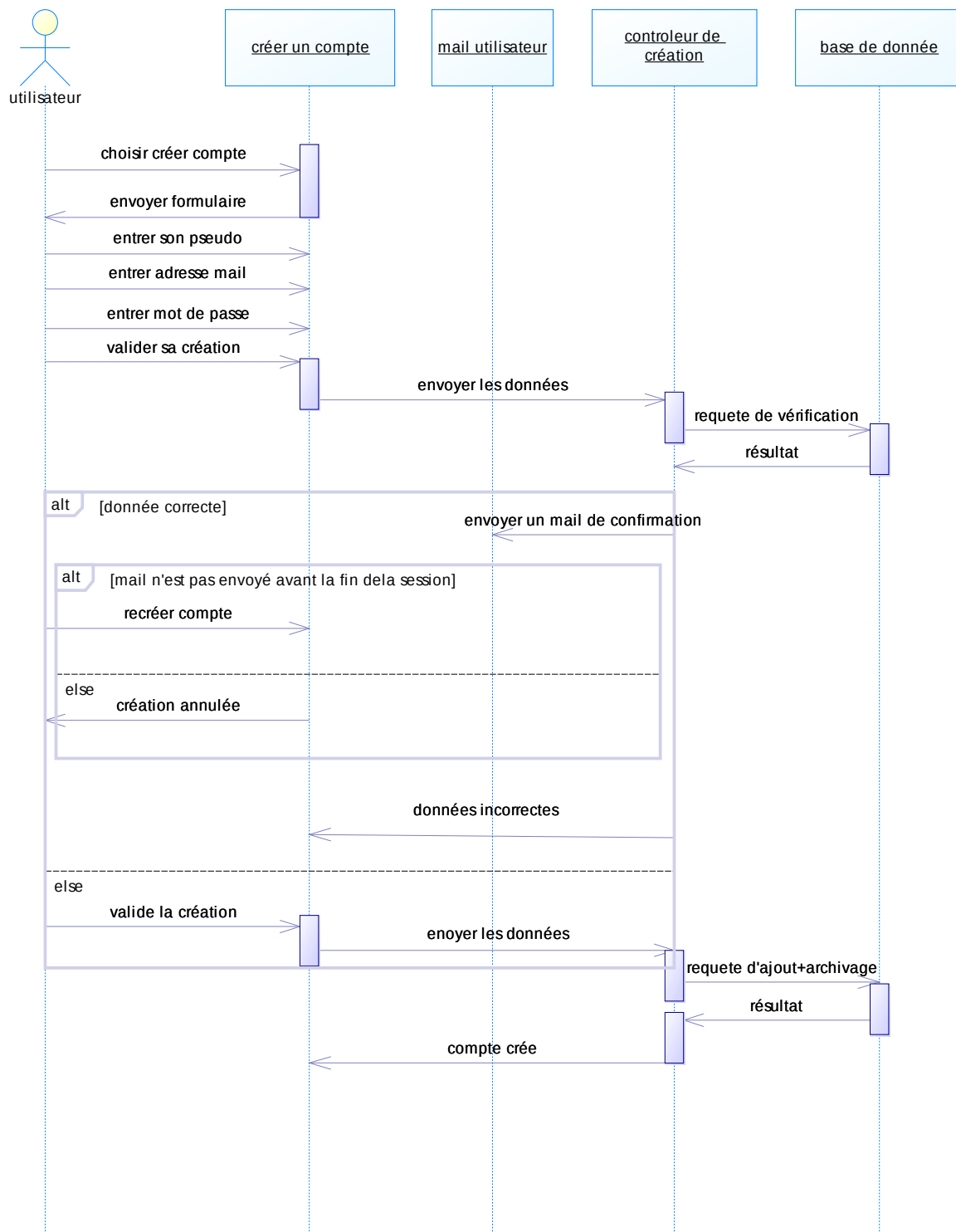


Diagramme de séquence « création d'un compte utilisateur »

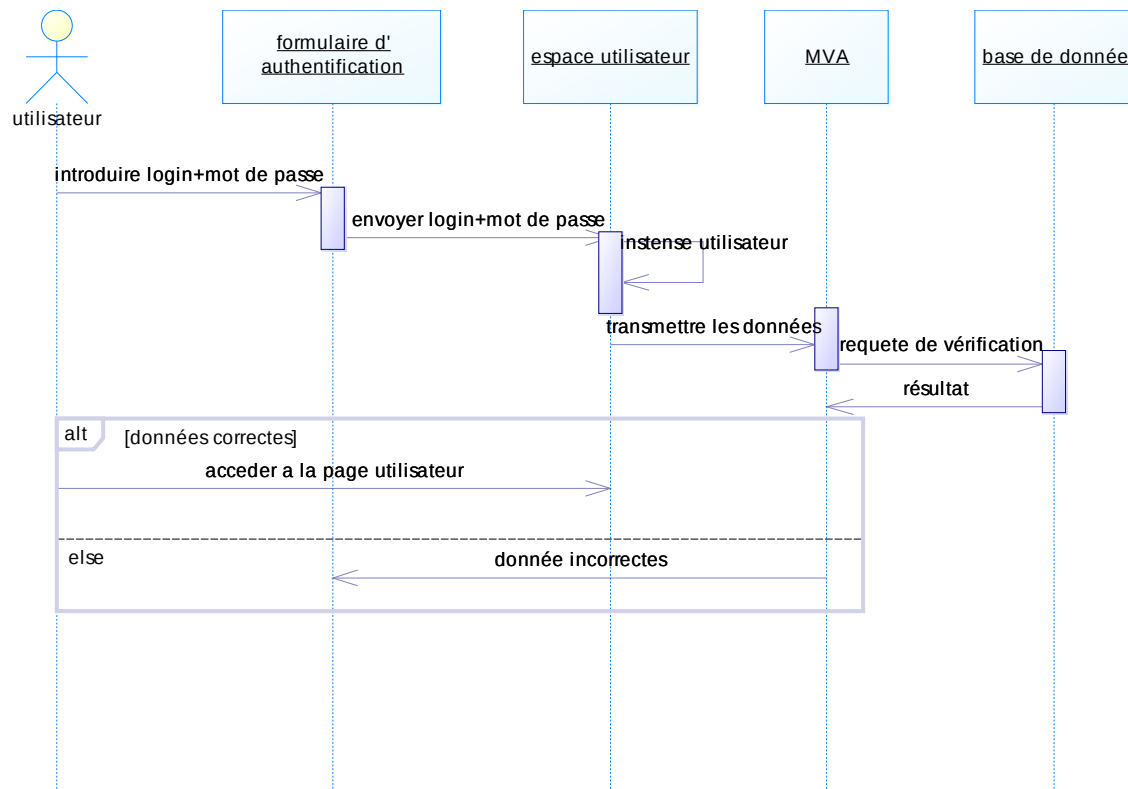


Diagramme de séquence « authentication dans le site SAUD »

- Supprimer compte :

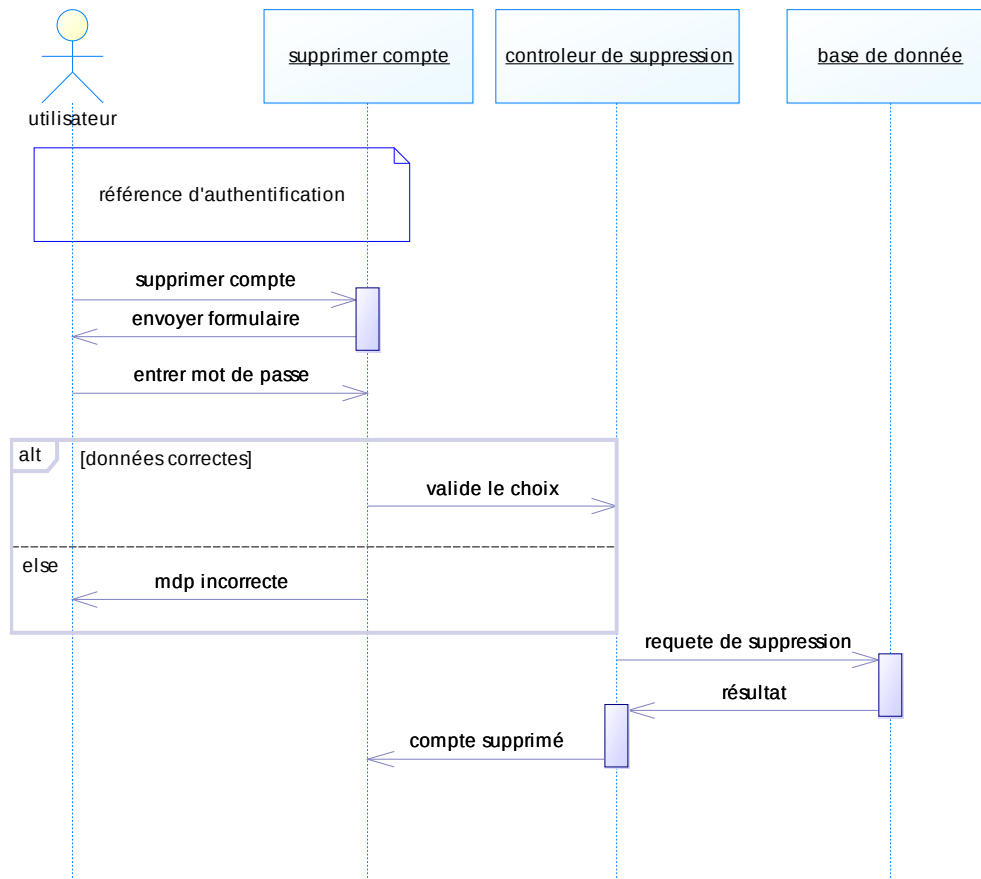


Diagramme de séquence « supprimer compte »

➤ Modifier question secrète :

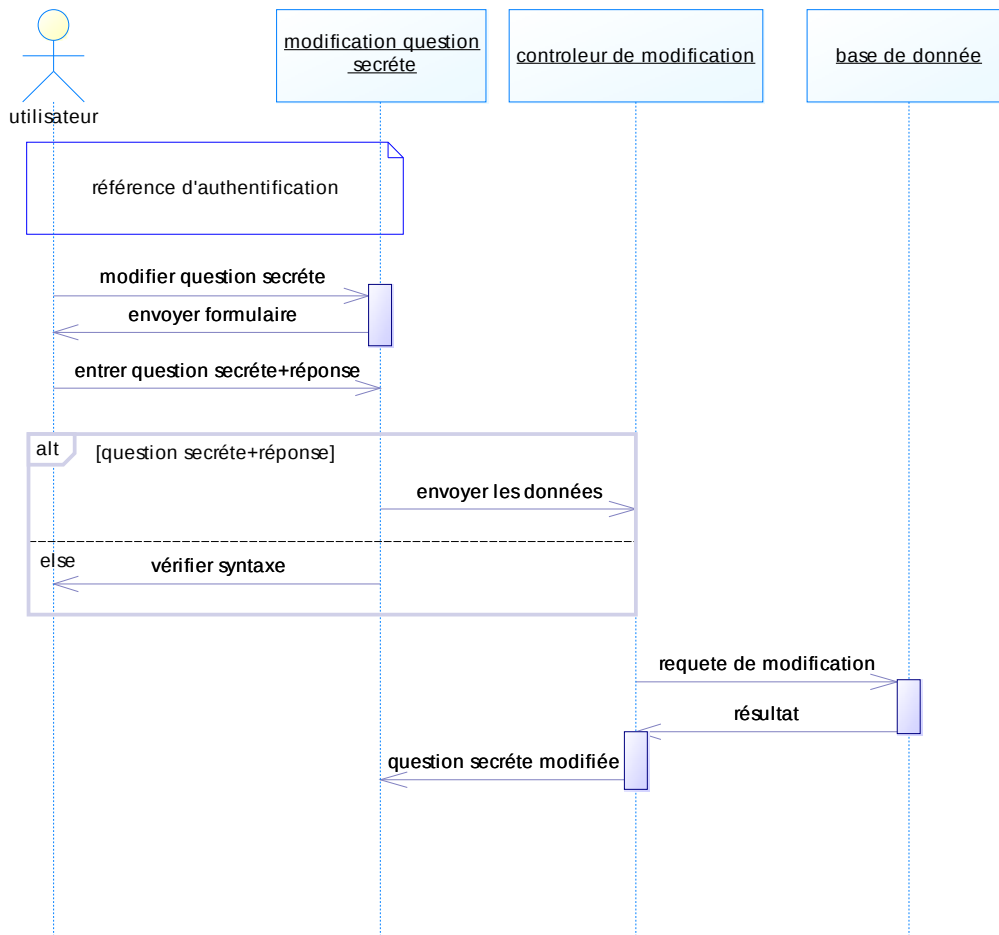


Diagramme de séquence « modifier question secrète »

➤ Oublier mot de passe :

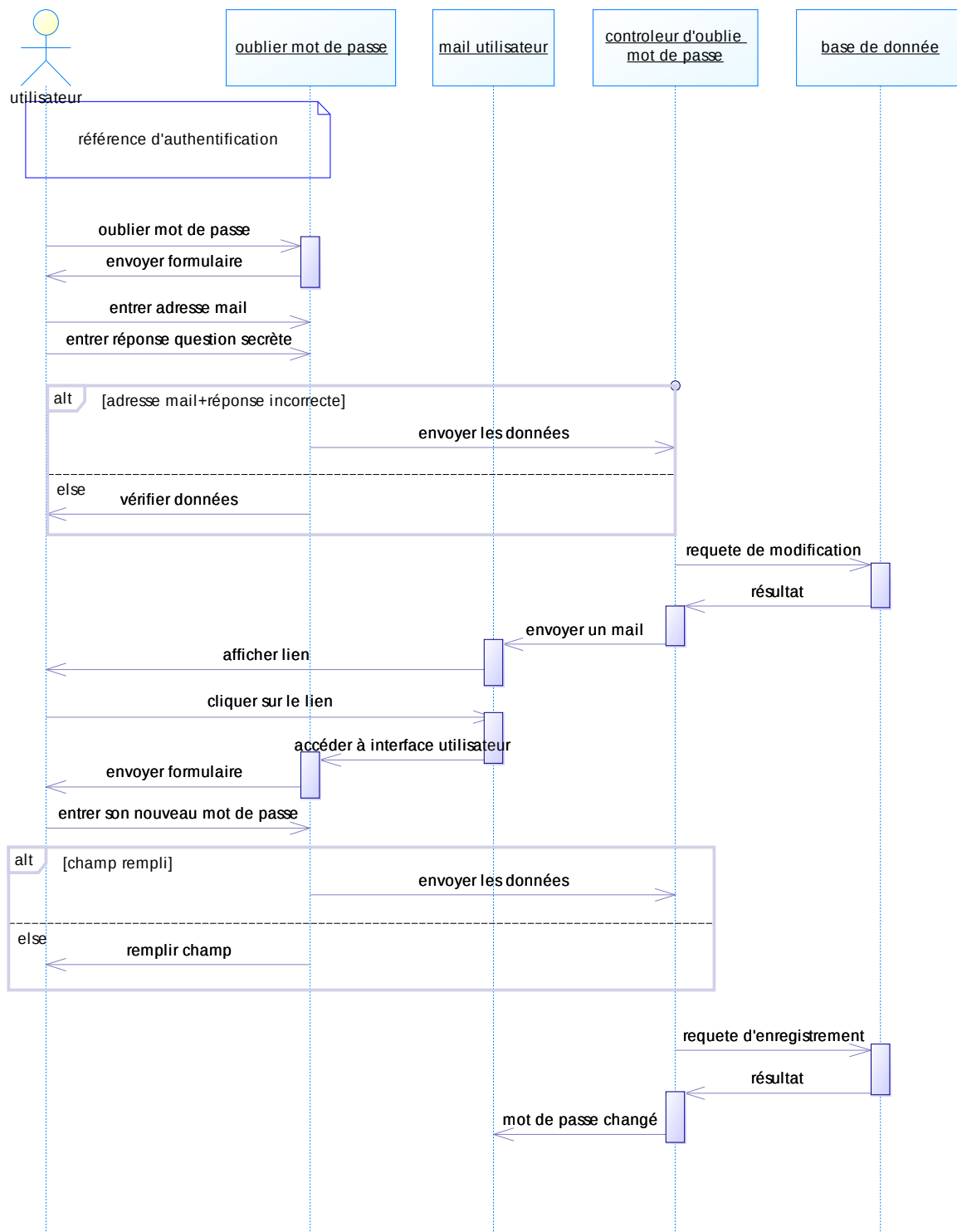


Diagramme de séquence « oublier mot de passe »

➤ Modifier mot de passe :

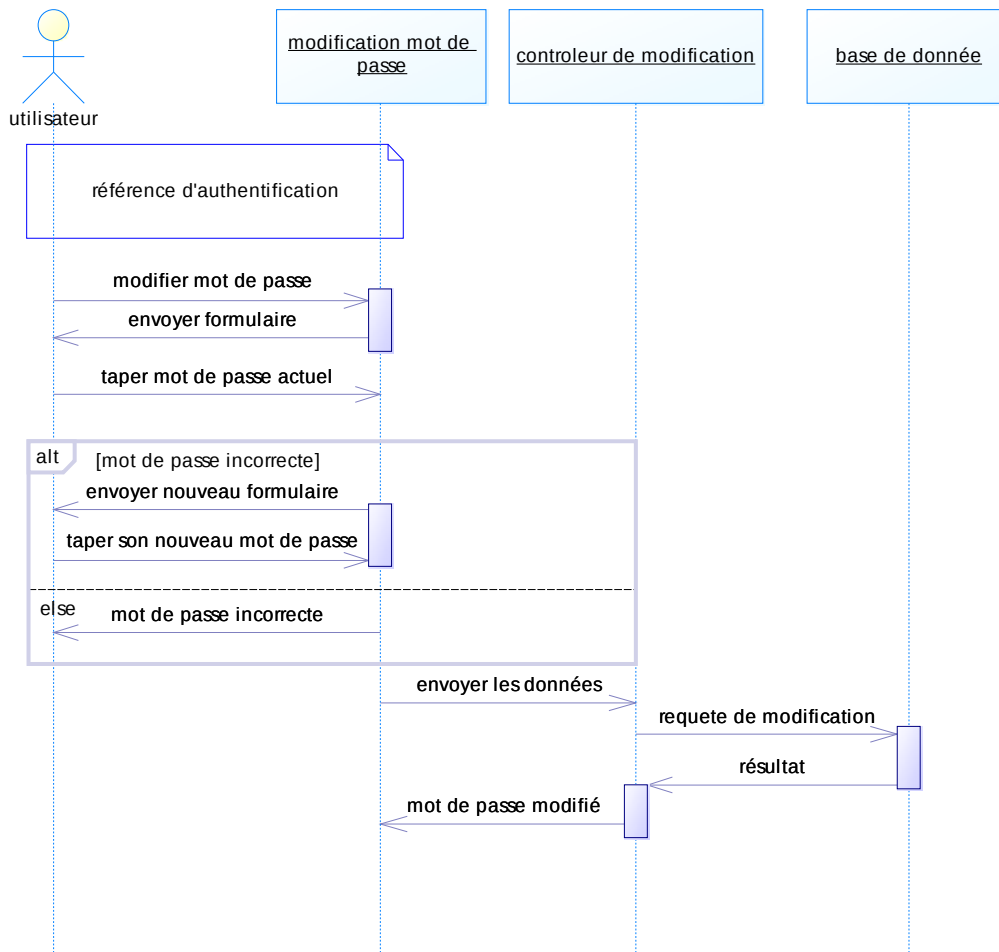


Diagramme de séquence « modifier mot de passe »

➤ Modifier adresse mail :

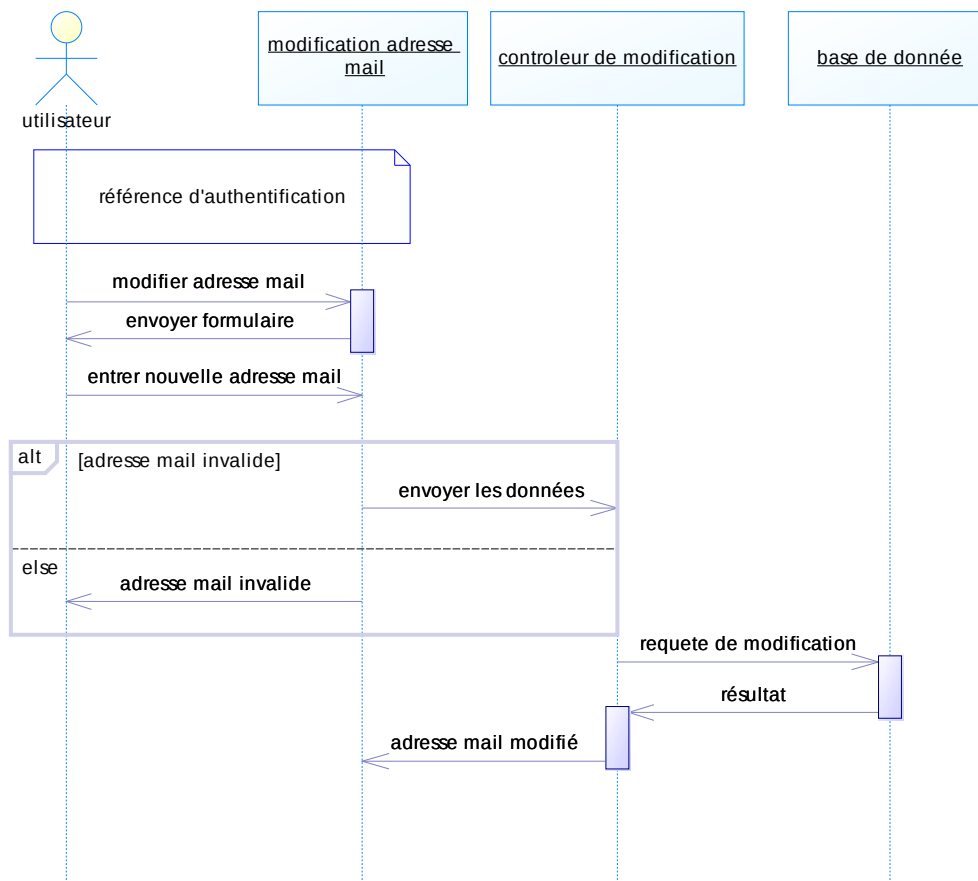


Diagramme de séquence « modifier adresse mail »

Les diagrammes de séquences pour administrateur :

- Supprimer compte utilisateur :

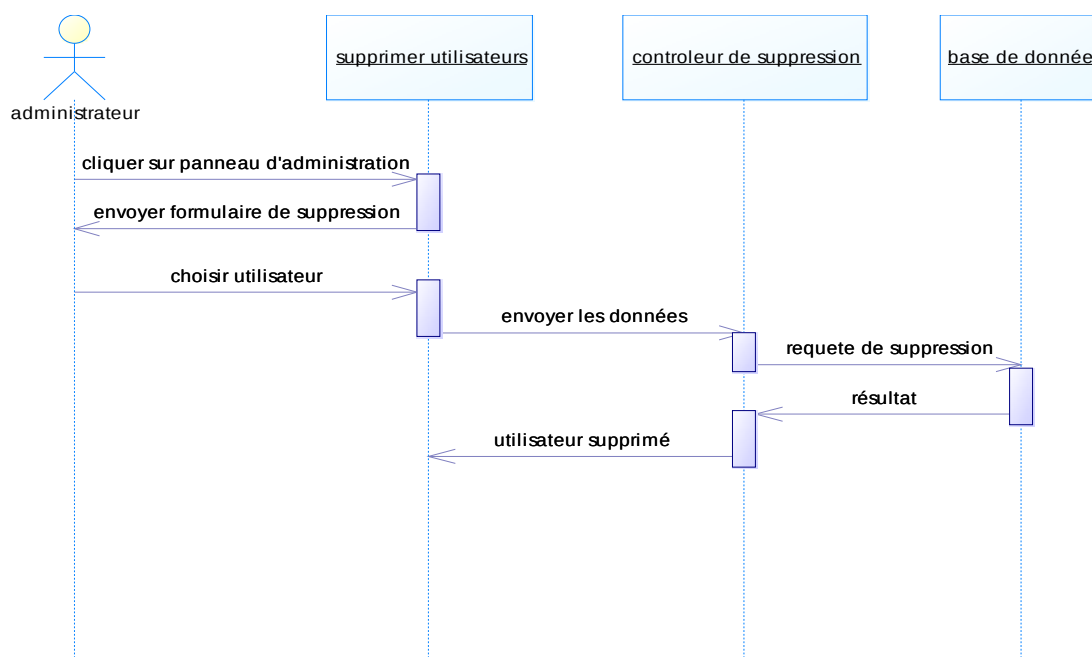


Diagramme de séquence « supprimer compte utilisateur »

- Connexion sur un site quelconque :

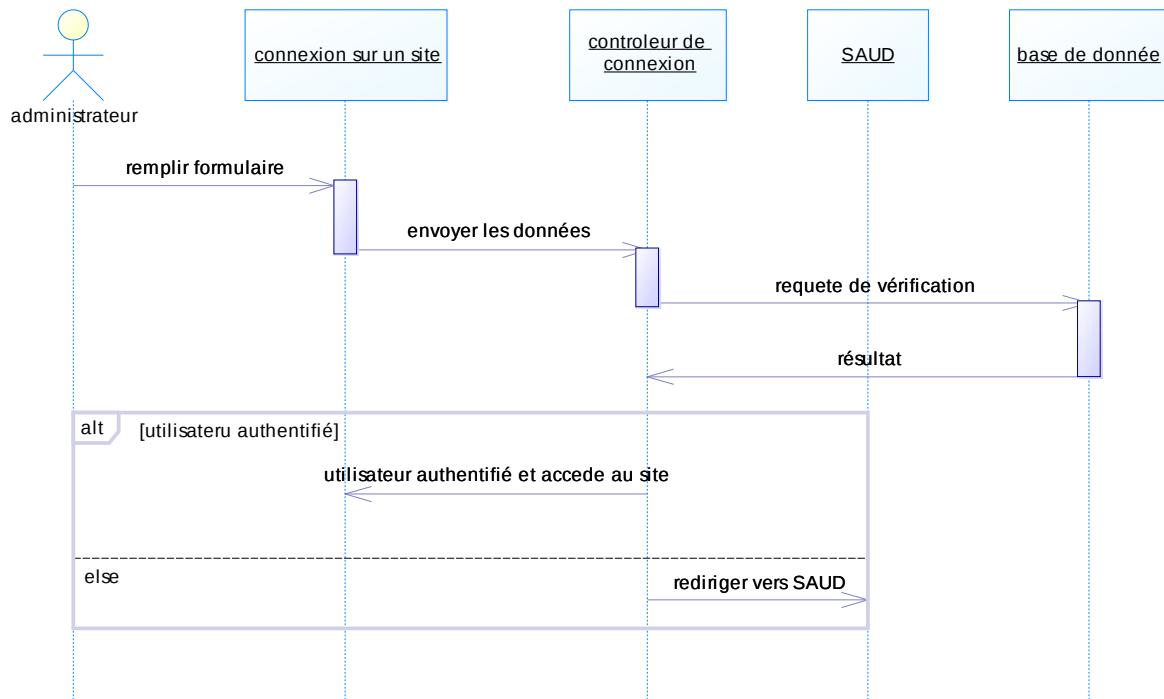


Diagramme de séquence « connexion sur un site quelconque »

7. Traçabilité