

Chapitre 2

Étude conceptuelle

La conception d'une application informatique est une phase très importante car elle permet de définir l'ensemble de ses fonctionnalités et d'avoir un premier aperçu du travail à réaliser. Il convient donc de choisir les outils d'analyses les mieux adaptés au projet afin d'en avoir une vue claire et précise.

Donc nous allons modéliser notre application en utilisant un langage de modélisation objet qui est **UML**¹[5] et selon les phases du processus rationnel RUP(Rational Unified Process). La conception de notre application se base sur les diagrammes des cas d'utilisation, diagramme de séquence, diagramme de classe.

2.1 Les phases du processus RUP(Rational Unified Process)

Le processus rationnel unifié (RUP) est l'un des plus célèbres implémentations de la méthode **PU**², permettant de donner un cadre au développement logiciel, répondant aux exigences fondamentales préconisées par les créateurs d'UML.[6]

2.1.1 Phase expression des besoins

L'expression des besoins consiste à capturer les besoins aux quels répond l'application et les différentes interactions entre l'utilisateur et le système à travers les cas d'utilisation.

1. **UML** (Unified Modeling Language) est un langage graphique de modélisation, il modélise l'ensemble des données et des traitements en élaborant des différents diagrammes qui permettent de préciser d'une manière claire la structure et le comportement d'un système.

2. **PU (Process Unified)** : une méthode de développement pour les logiciels orientés objet, elle est générique, itérative et incrémentale.

● Besoins fonctionnels

Les besoins fonctionnels expriment une action qui doit être menée sur l'infrastructure à définir en réponse à une demande. C'est le besoin exprimé par le client. Pour cela, nous aurons :

- Le système doit permettre à l'utilisateur de parcourir et consulter les différents traitements.
- Le système ne doit permettre qu'aux utilisateurs ayant droits d'accès d'effectuer des traitements.
- Le système doit répondre automatiquement à chaque requête introduite par l'utilisateur.
- Le système doit être connecté au réseau.
- Le système doit alerter en cas de dysfonctionnement sur le réseau.
- Le système doit fournir les détails relatifs aux alertes.

● Besoins non fonctionnels

Les besoins non fonctionnels se regroupent autour des points suivant :

- Réduire au maximum le temps de chargement des données.
- Ergonomie et convivialité du produit.
- Facilité d'utilisation.
- Un accès sécurisé et personnalisé.

A. Définition des cas d'utilisation

Les cas d'utilisation permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système, ils identifient les utilisateurs du système (acteurs) et leur interaction avec le système.[5]

B. Identification des acteurs du système

Le travail effectué au sein de cette partie permet d'identifier les acteurs mis en jeu et de visualiser les scénarios des cas d'utilisation de notre application qui permettent de bien superviser des équipements connectés sur un réseau.

Acteur : c'est une entité externe qui agit sur le système.[7]

Les acteurs qui interagissent avec cette application sont les suivants :

- **Administrateur** : désigne la personne chargée de consulter toutes les informations concernant l'ensemble des équipements du réseau, et la gestion des objets (équipements, comptes utilisateurs ...).
- **Superviseur** : désigne la personne qui n'a le droit que de consulter les informations concernant l'ensemble des équipements.

C. Les diagrammes des cas d'utilisation

Les différents diagrammes des cas d'utilisation du système sont les suivants :

Diagramme de cas d'utilisation globale :

Le diagramme suivant donne une vision globale du comportement fonctionnel de ce système. Le superviseur doit s'authentifier pour qu'il puisse consulter la liste des équipements disponibles dans l'architecture réseau, il peut aussi consulter la performance du réseau (débit, vitesse, ...).

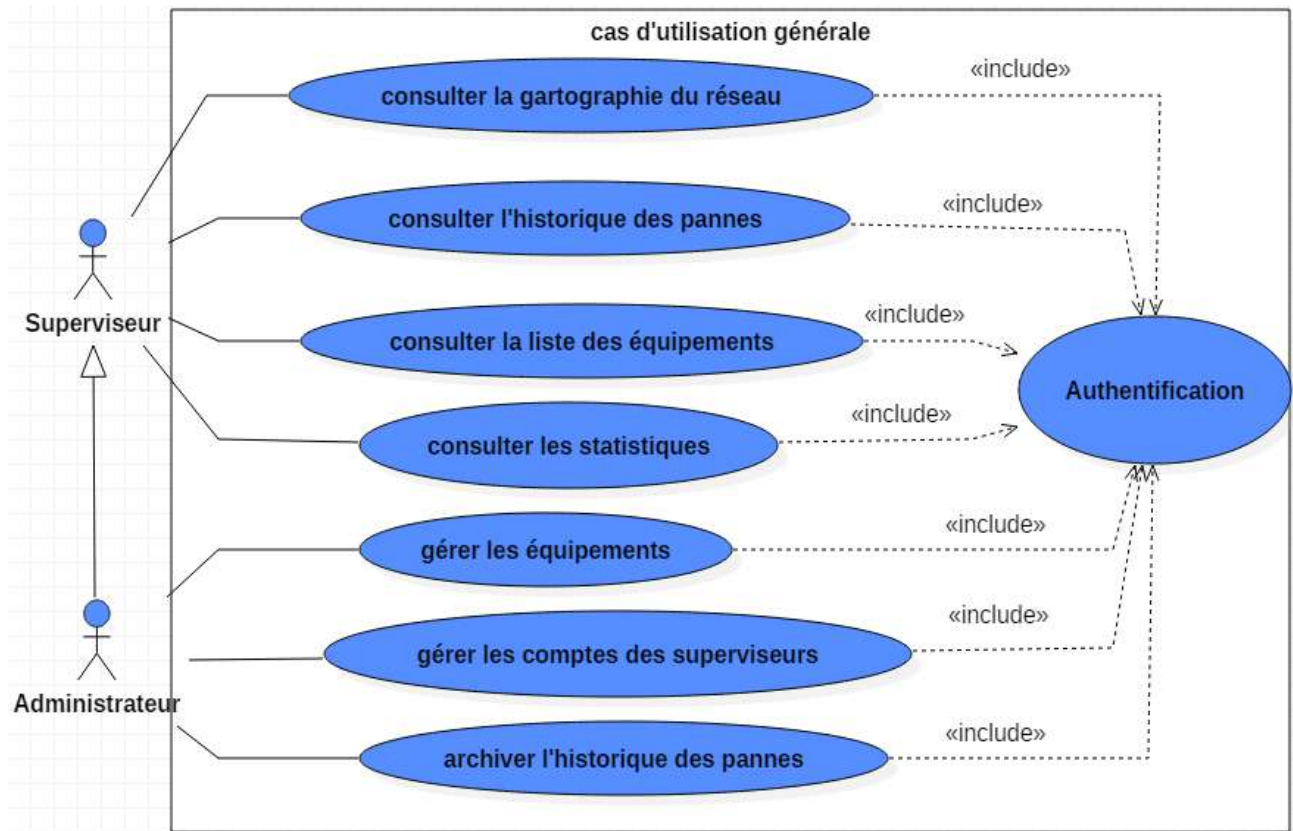


FIGURE 2.1 – Diagramme de cas d'utilisation globale

Diagramme de cas d'utilisation de l'authentification :

Le diagramme suivant montre la manière dont le superviseur s'authentifier.

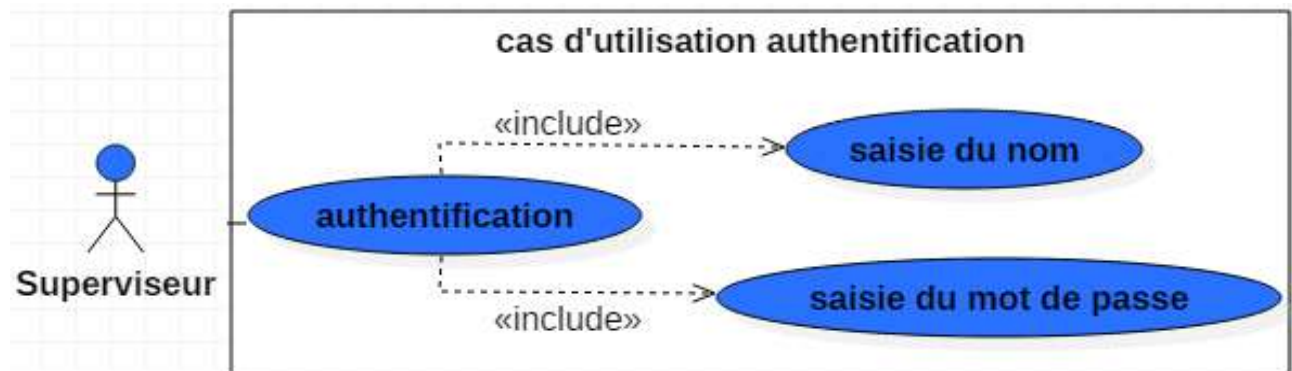


FIGURE 2.2 – Diagramme de cas d'utilisation de l'authentification

Diagramme de cas d'utilisation de gérer les comptes des superviseurs :

Le diagramme suivant montre que seul l'administrateur peut modifier, ajouter et supprimer un compte.

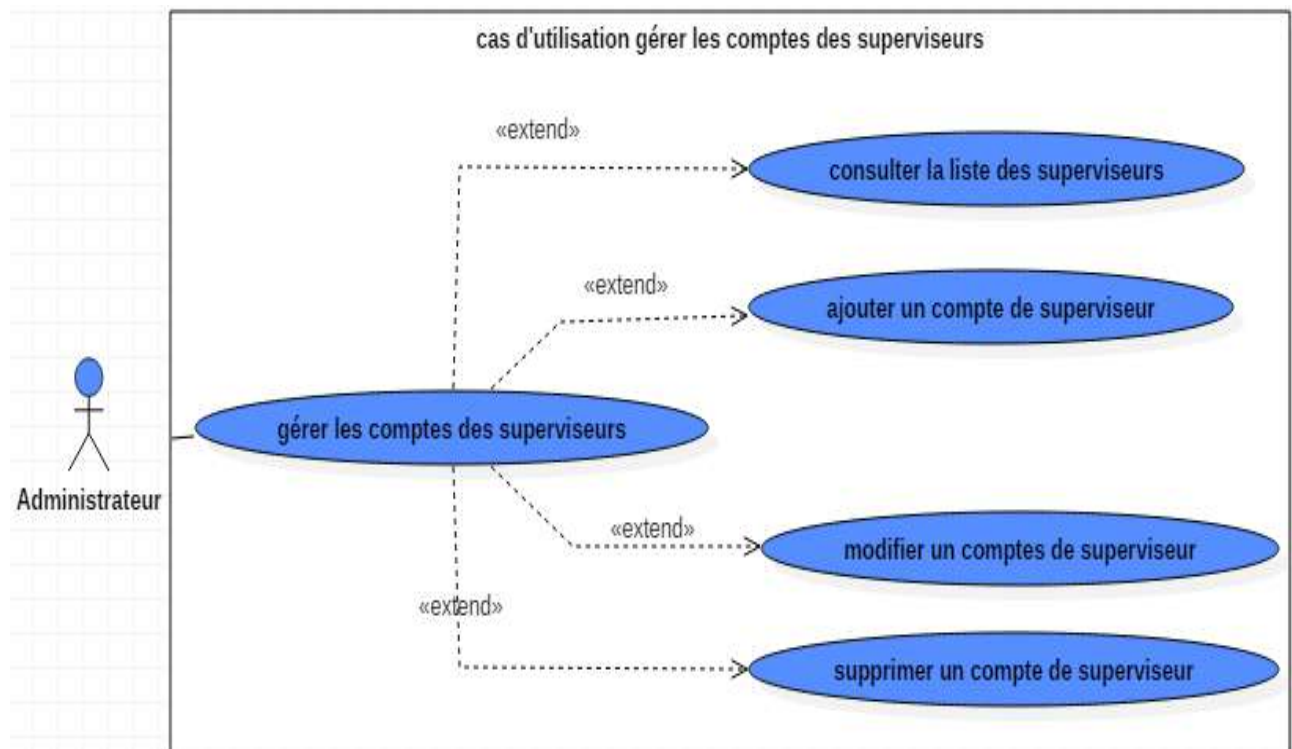


FIGURE 2.3 – Diagramme de cas d'utilisation de gérer les comptes des superviseurs

Diagramme de cas d'utilisation gérer les équipements :

Le diagramme suivant montre que seul l'administrateur peut modifier, ajouter et supprimer un équipement pour le superviser.

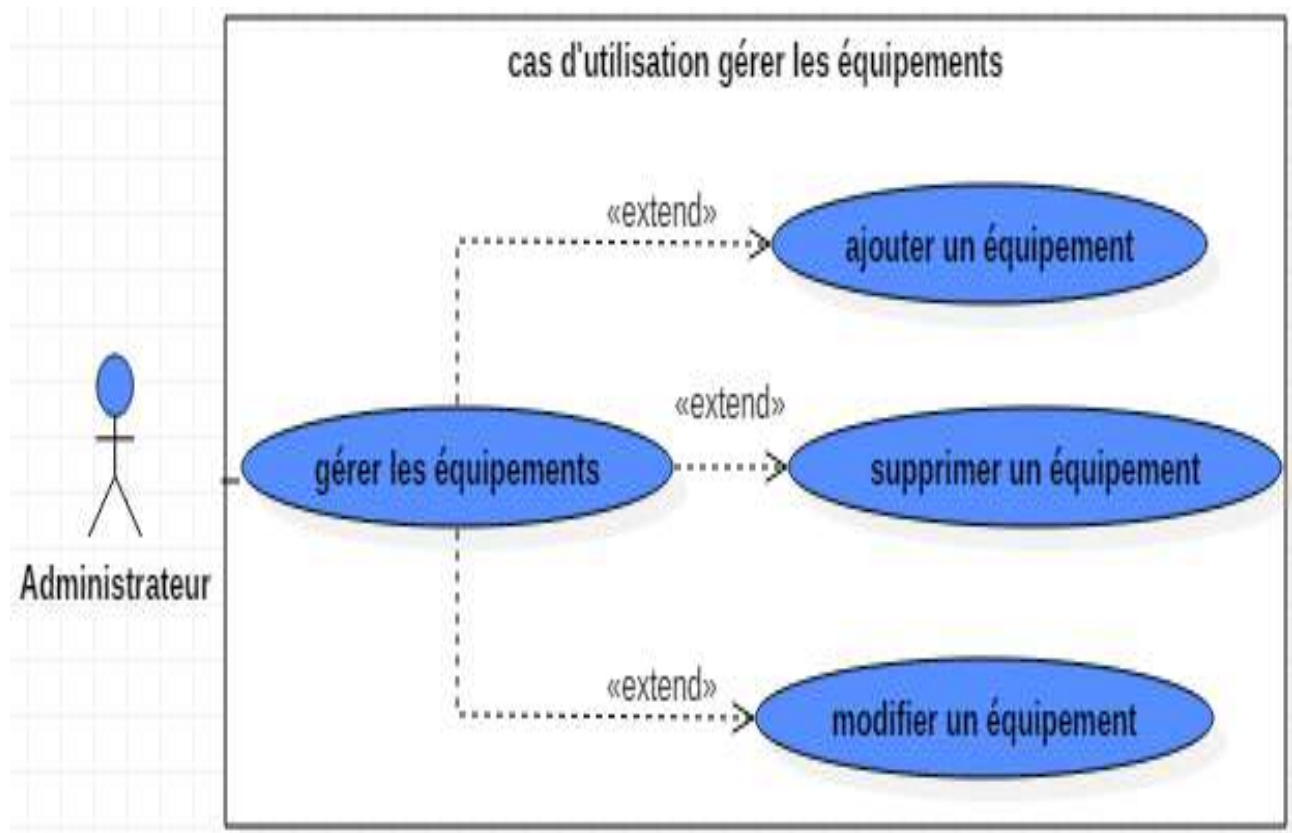


FIGURE 2.4 – Diagramme de cas d'utilisation de gérer les équipements

Après avoir identifié tous les acteurs du système et leur besoins. Nous allons présenter le déroulement des traitements dans la phase suivante.

2.1.2 La phase d'analyse

L'analyse est l'une des phases les plus importantes dans le cycle de développement d'un logiciel. C'est là où on définit, on analyse clairement le problème et on exprime les interactions entre les utilisateurs et le système à l'aide des scénarios et des diagrammes de séquence.

A. Définition de diagramme de séquence

Les diagrammes de séquences permettent de décrire comment les éléments du système interagissent entre eux et avec les acteurs selon un point de vue.[5]

B. Les scénarios et les diagrammes de séquence

- **Le cas d'utilisation Authentification**

- **L'acteur** : tous les acteurs.
- **But** : authentifier les utilisateurs du système.
- **Pré condition** : Le nom et le mot de passe du superviseur doivent être connus par le système.
- **Scénario principale** :
 1. Le superviseur lance l'application.
 2. Le système affiche la page d'authentification
 3. Le superviseur saisit le nom et le mot de passe puis valide.
 4. Le système vérifie le nom et le mot de passe et le type du superviseur.
 5. Le système affiche la page d'accueil selon le type du superviseur.
- **Alternative du déroulement** :
 - À l'étape (3), si les champs sont vides, le système affiche un message d'erreur et reprend à l'étape (2).
 - À l'étape (3), si le nom et/ou le mot de passe ne sont pas valide, le système affiche un message d'erreur et reprend à l'étape (2).
- **Poste condition** : Une session est créée.

Diagramme de séquence de l'authentification :

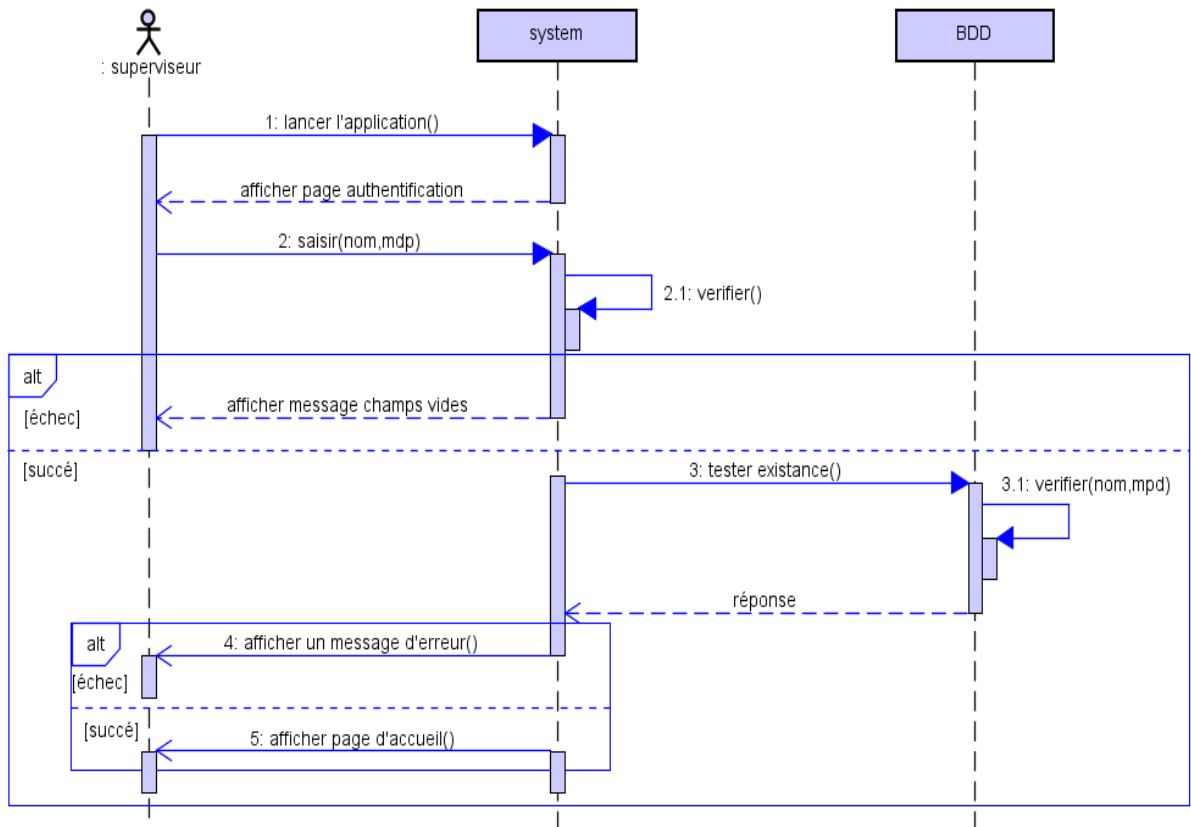


FIGURE 2.5 – Diagramme de séquence d'authentification

● Cas d'utilisation consulter les équipements

- **L'acteur** : tous les acteurs.
- **But** : consulter les équipements à superviser.
- **Pré condition** : être authentifié.
- **Scénario principale** :
 1. Le superviseur choisir le menu "groupes des équipements".
 2. Le système affiche la liste des groupes des équipements.
 3. Le superviseur sélectionne le groupe.
 4. Le système affiche la liste des équipements.
 5. Le superviseur sélectionne l'équipement à consulter.
 6. Le système affiche la page qui contient les détails sur l'équipement sélectionné.
- **Alternative du déroulement** :

En cas d'erreur le système affiche un message d'erreur et reprend a l'étape précédente.

Diagramme de séquence consulter les informations d'un équipement :

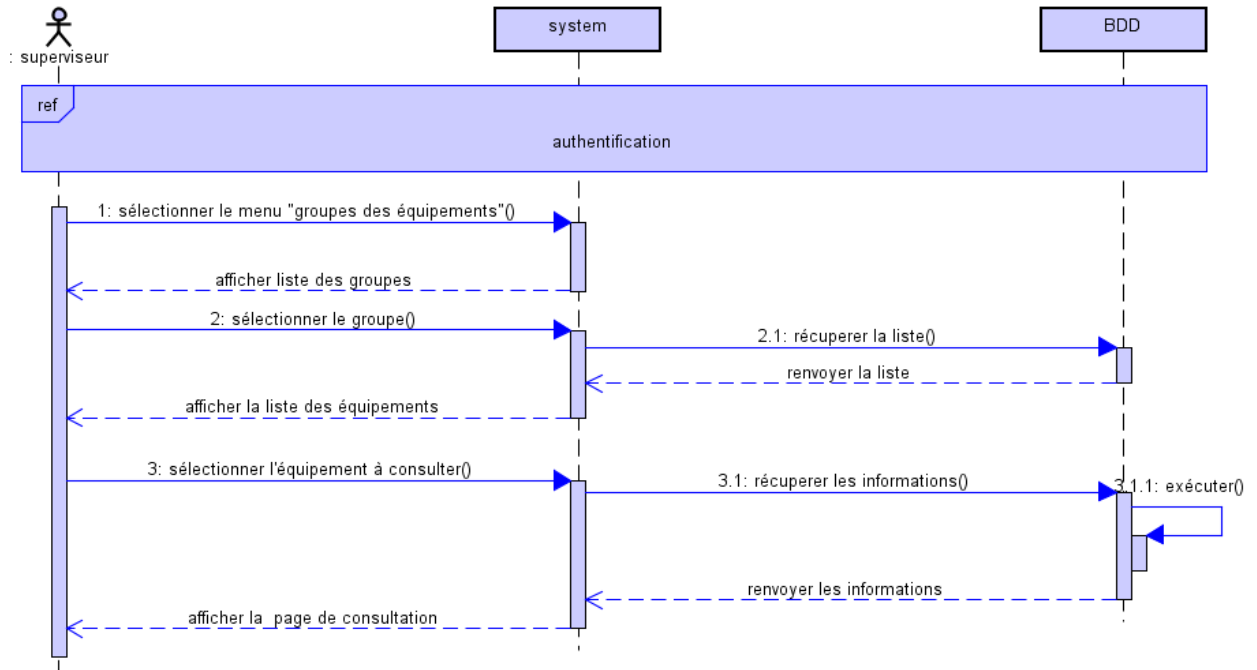


FIGURE 2.6 – Diagramme de séquence consulter les informations d'un équipement

- **Cas d'utilisation modifier les équipements**

- **L'acteur** : Administrateur.
- **But** : modifier les équipements qu'il supervise.
- **Pré condition** : être connecter autant qu'administrateur.
- **Scénario principale** :
 1. Le superviseur choisir le menu "groupes des équipements".
 2. Le système affiche la liste des groupes des équipements.
 3. Le superviseur sélectionne le groupe.
 4. Le système affiche la liste des équipements.
 5. Le superviseur sélectionne l'équipement à modifier.
 6. Le système affiche le formulaire de modification.
 7. L'administrateur remplit le formulaire puis valide.
 8. Le system stocke les données dans la base de données et affiche un message de succès.

— **Alternative du déroulement :**

À l'étape (7) le système affiche un message d'erreur si les champs sont vides.

À l'étape (8) le système affiche un message d'erreur si on a une erreur de stockage.

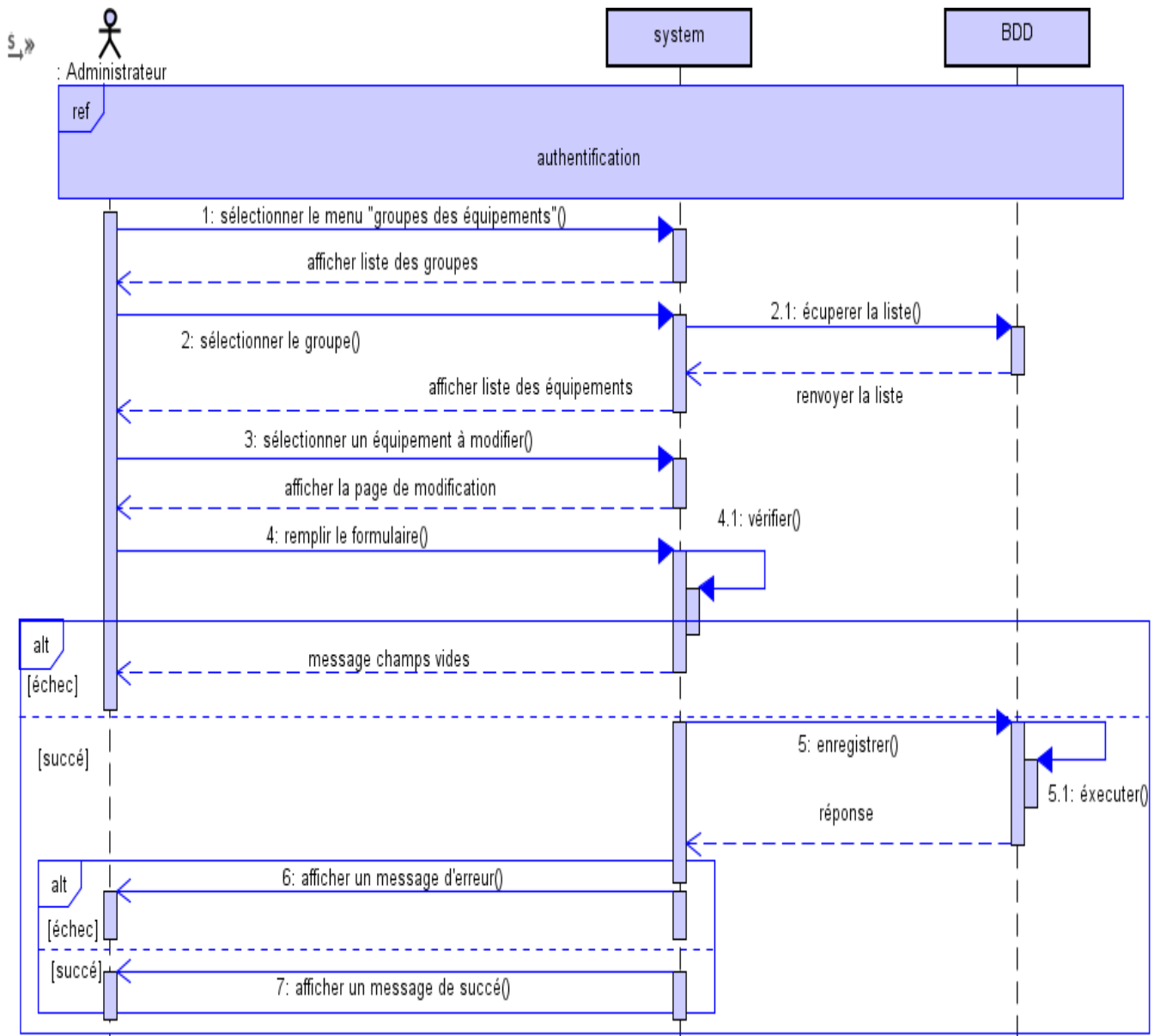
— **Poste condition :** Mettre à jour la base de données.**Diagramme de séquence modifier un équipement**

FIGURE 2.7 – Diagramme de séquence modifier un équipement

● **Cas d'utilisation supprimer les équipements**

- **L'acteur** : Administrateur.
- **But** : supprimer les équipements qu'il supervise.
- **Pré condition** : être connecter autant qu'administrateur.
- **Scénario principale** :
 1. L'administrateur choisir le menu "groupes des équipements".
 2. Le système affiche la liste des groupes des équipements.
 3. L'administrateur sélectionne le groupe.
 4. Le système affiche la liste des équipements.
 5. L'administrateur sélectionne l'équipement à supprimer.
 6. Le système affiche un message de confirmation de la suppression.
 7. L'administrateur confirme la suppression.
 8. Le système effectue la suppression.
- **Alternative du déroulement** :

À l'étape (7) l'administrateur annule la suppression.

À l'étape (8) le système affiche un message d'erreur si on a une erreur au niveau de la base de données.
- **Poste condition** : Mettre à jour la base de données.

Diagramme de séquence supprimer un équipement :

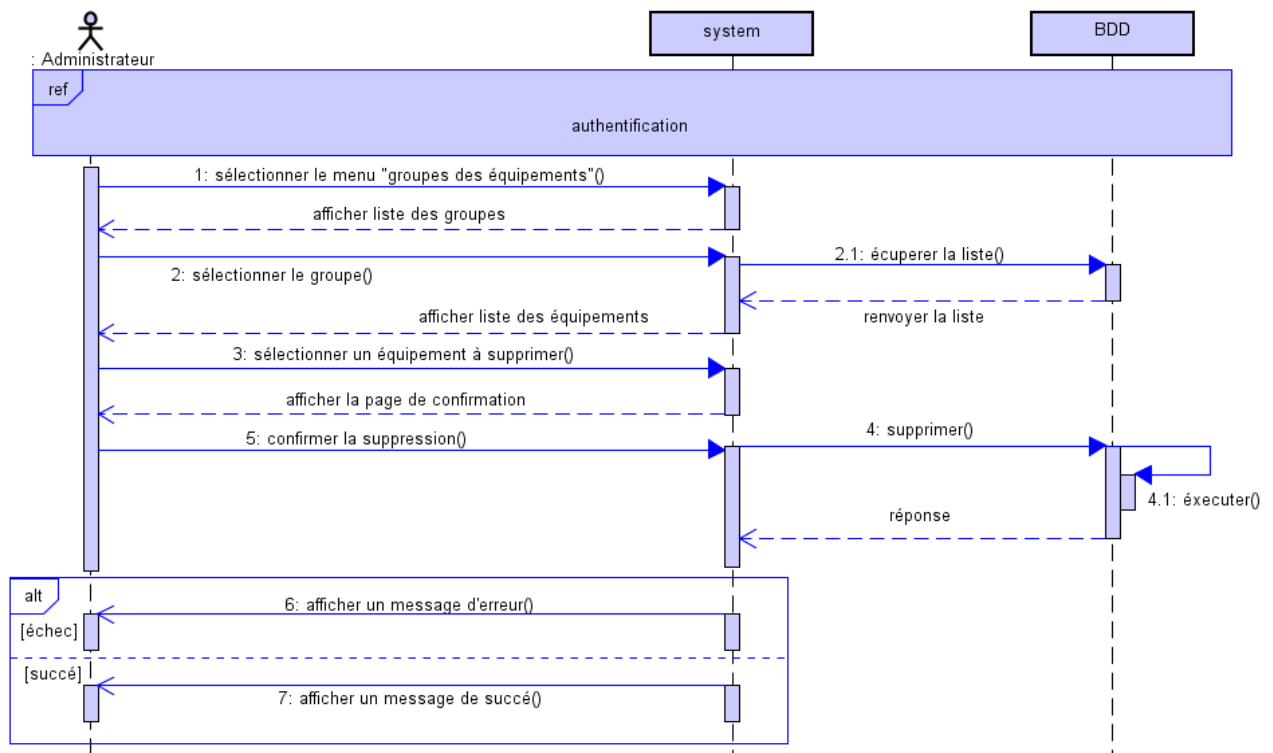


FIGURE 2.8 – Diagramme de séquence supprimer un équipement

● **Cas d'utilisation ajouter équipement**

- **L'acteur** : administrateur.
- **But** : ajouter des équipements à superviser.
- **Pré condition** : être connecter autant qu'administrateur.
- **Scénario principale** :
 1. L'administrateur sélectionne le menu "ajouter équipement" de puis le menu d'administrateur.
 2. Le système affiche le formulaire.
 3. L'administrateur remplit le formulaire puis il valide.
 4. Le système reprend à l'étape (2) avec un message de sucées.
- **Alternative du déroulement** : À l'étape (4) le système affiche un message d'erreur et reprend à l'étape (2) si l'équipement existe déjà dans la base de données ou bien une erreur s'est produite.
- **Poste condition** : Mettre à jour la base de données.

Diagramme de séquence ajouter un équipement :

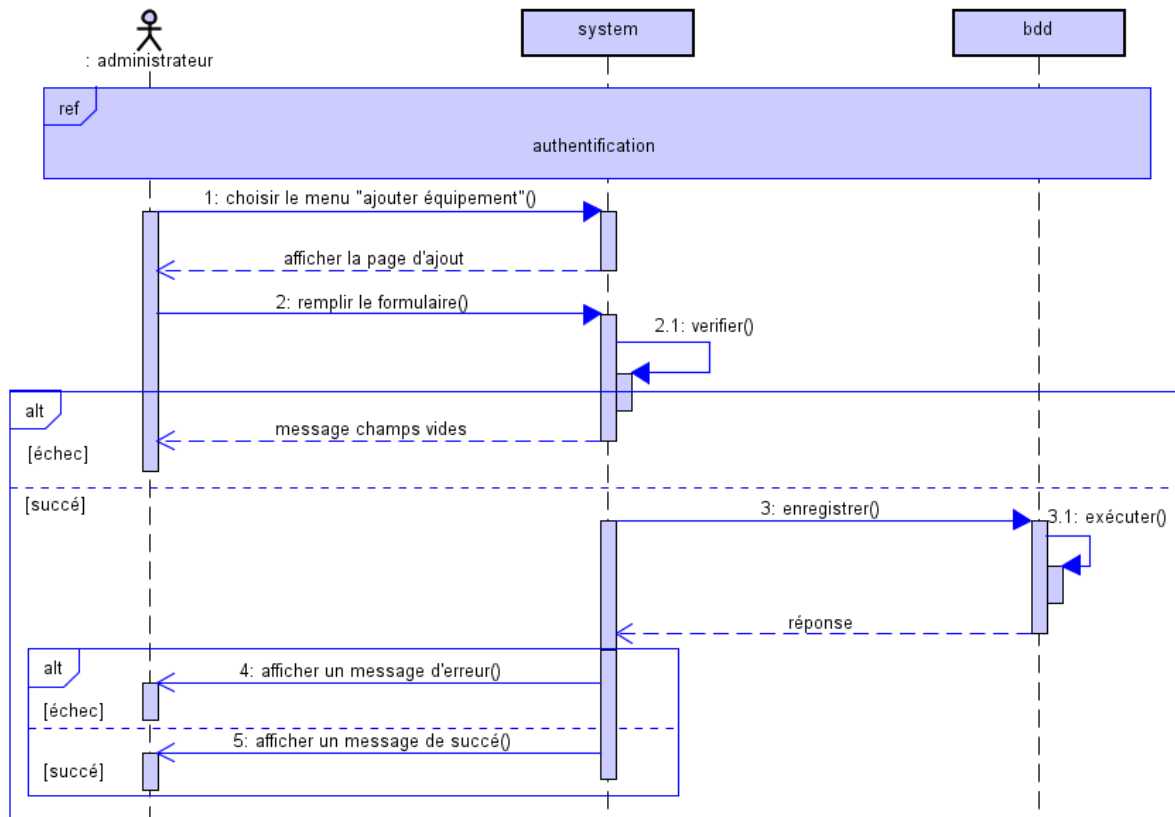


FIGURE 2.9 – Diagramme de séquence ajouter un équipement

- Cas d'utilisation gérer l'historique des pannes
 - **L'acteur** : Administrateur.
 - **But** : L'archivage et la consultation des historiques des pannes.
 - **Pré condition** : Être connecter autant qu'administrateur.
 - **Scénario principale** :
 1. L'administrateur sélectionne le menu "historiques des pannes".
 2. Le système affiche la liste des pannes.
 3. L'administrateur peut consulter les détails sur la panne.
 4. L'administrateur peut conserver une panne dans les archives.
 - **Alternative du déroulement** :
En cas d'erreur le système reprend à l'étape précédente.
 - **Poste condition** : Mettre à jour la base de données.

Diagramme de séquence gérer l'historique des pannes :

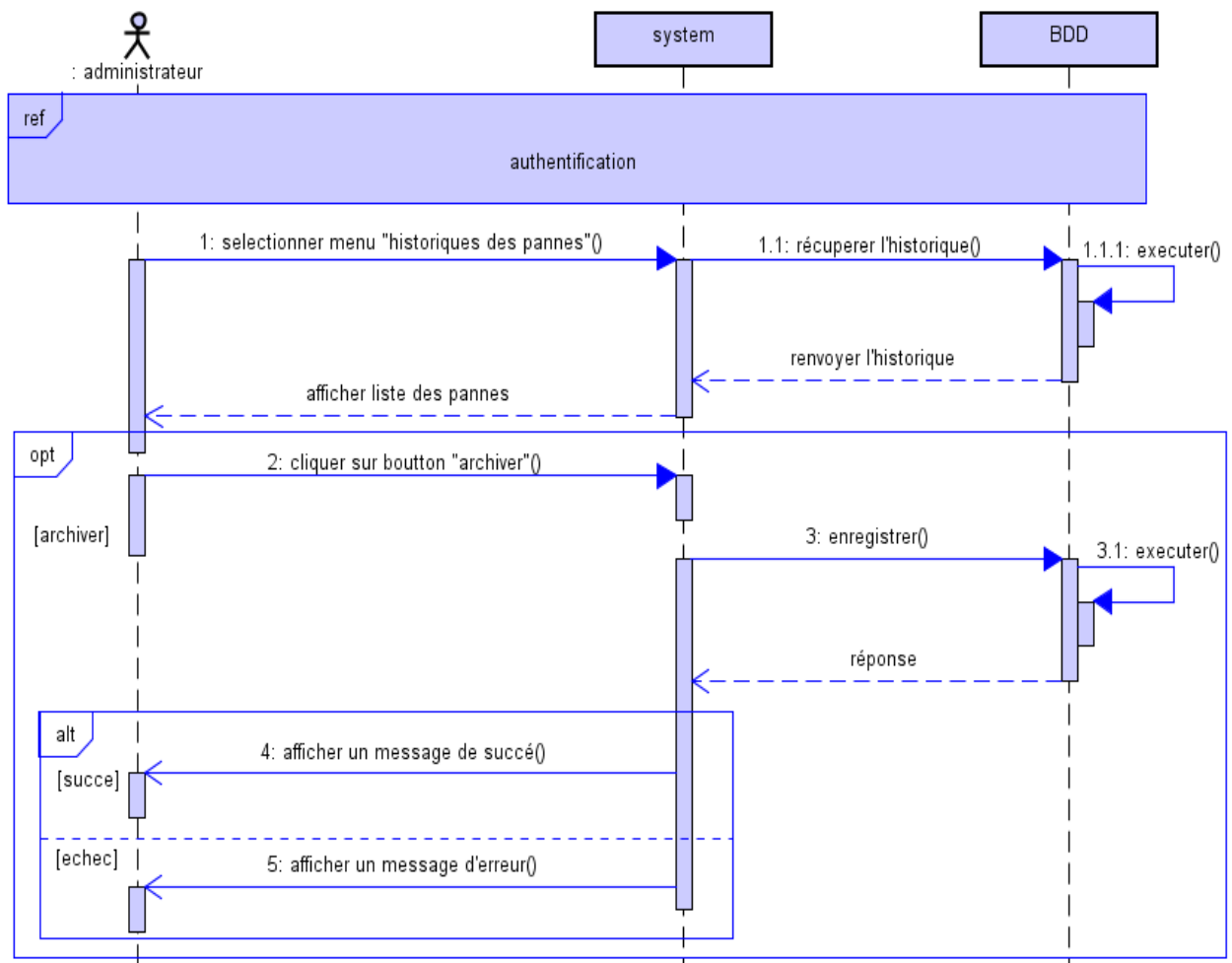


FIGURE 2.10 – Diagramme de séquence de gérer l'historique des pannes

- **Cas d'utilisation gérer les comptes des superviseurs**

- **L'auteur** : administrateur.
- **But** : l'ajout, la modification, la suppression et la consultation des comptes des superviseurs.
- **Pré condition** : être connecter autant qu'administrateur.
- **Scénario principale** :
 1. L'administrateur choisit le menu "paramètre" de puis le menu d'administrateur.
 2. Le système affiche la liste des superviseurs.
 3. L'administrateur peut ajouter un superviseur en remplissant le formulaire puis valider.
 4. L'administrateur peut modifier un superviseur sélectionnée en remplissant un formulaire puis il valide.
 5. L'administrateur peut supprimer un superviseur puis il confirme.
- **Alternative du déroulement** : En cas d'erreur le système reprend à l'étape précédente.
- **Poste condition** : mettre à jour la base de données.

Diagramme de séquence modifier un compte de superviseur :

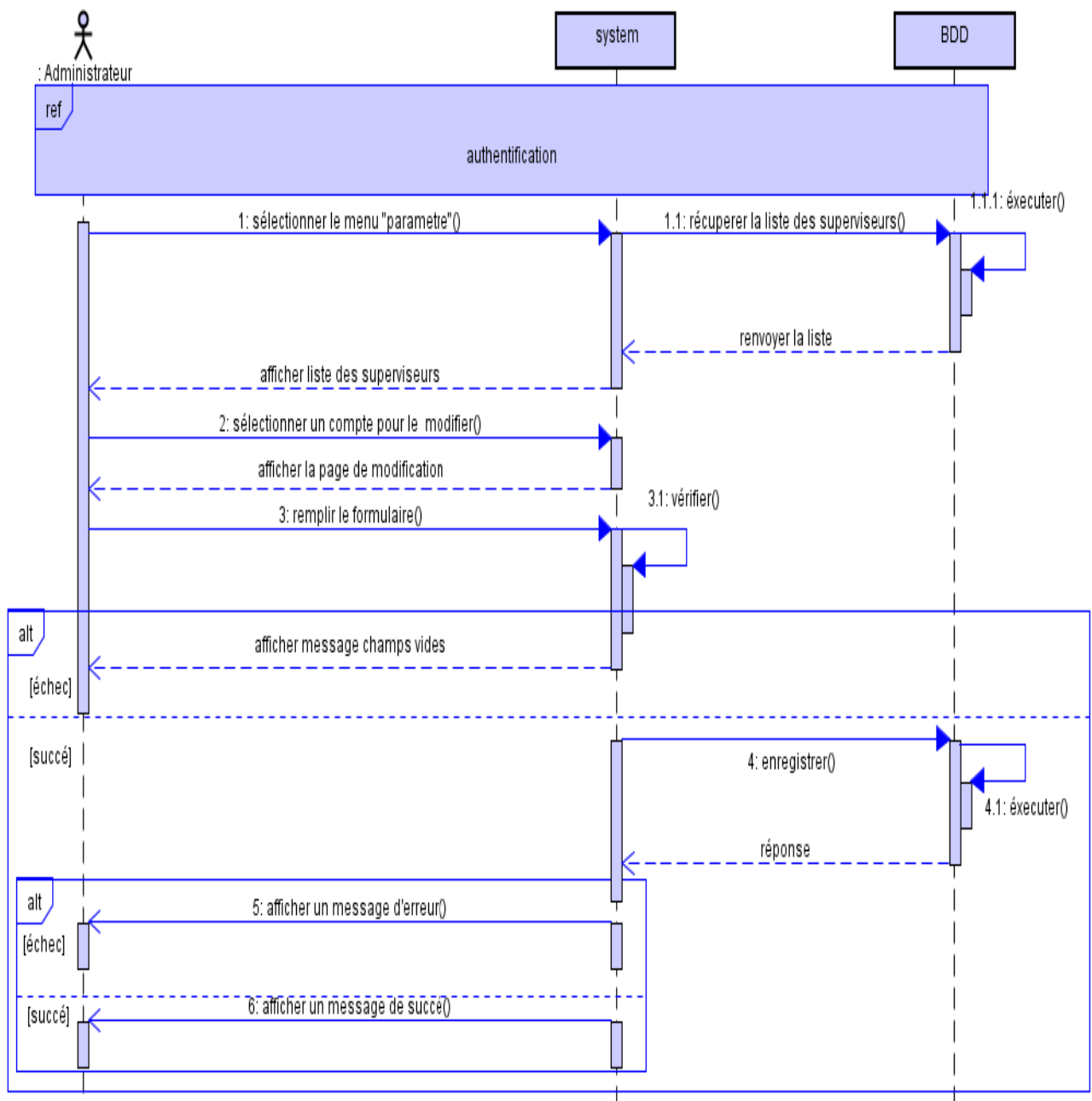


FIGURE 2.11 – Diagramme de séquence modifier un compte de superviseur

Diagramme de séquence supprimer un compte de superviseur :

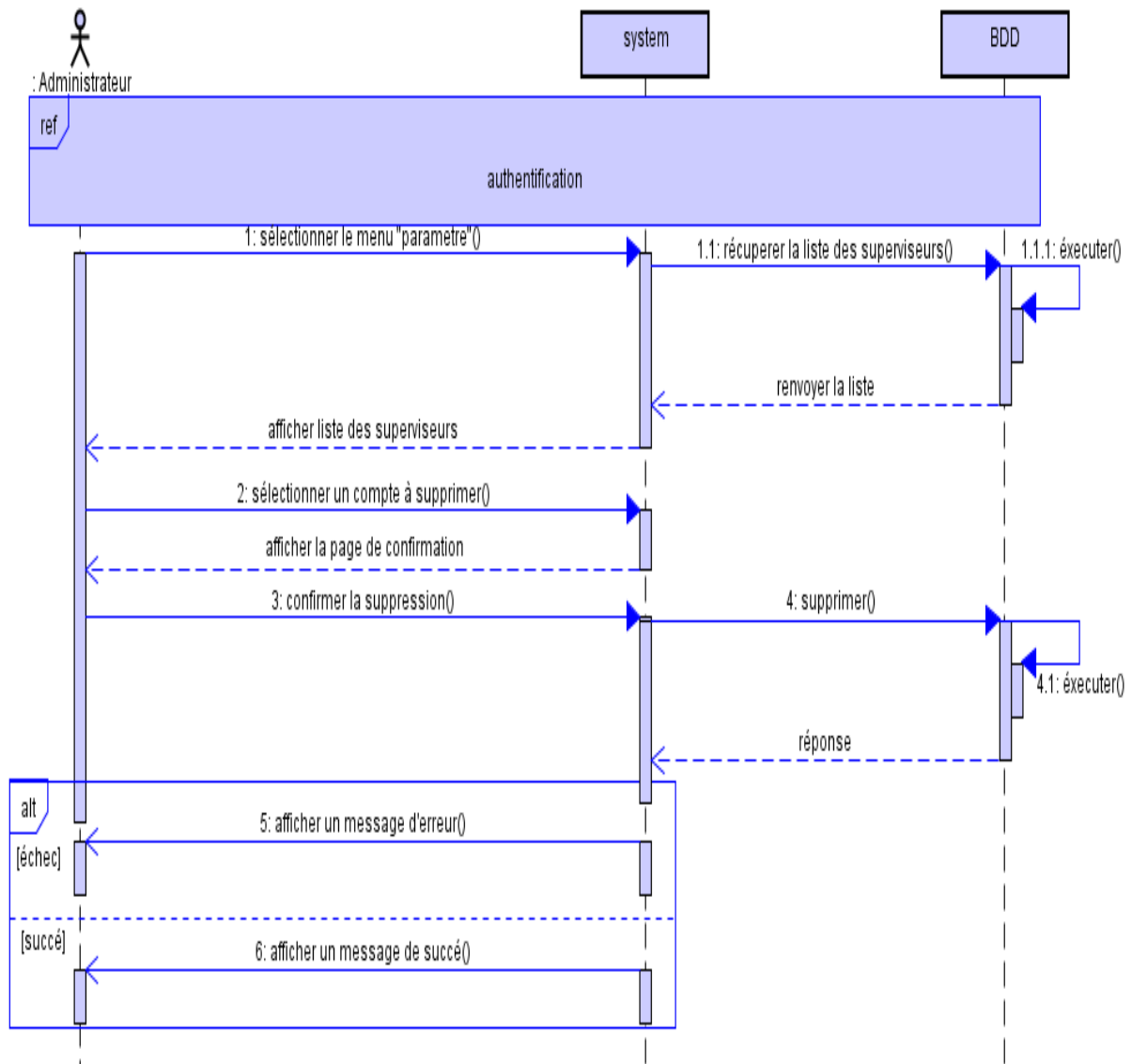


FIGURE 2.12 – Diagramme de séquence supprimer un compte de superviseur

Chapitre 3

Implémentation et réalisation

Après avoir achevé la phase d'analyse et de conception, nous entamons dans ce chapitre la partie réalisation. Nous commençons tout d'abord par présenter l'architecture de l'application, l'environnement matériel et les outils de développement utilisés pour mener à terme ce travail. Par la suite, nous exposerons le travail réalisé à l'aide des captures d'écran. Au terme de cette étape, la solution sera opérationnelle et totalement intégrée au sein du service LET.

3.1 Environnement de travail et technologies utilisées

3.1.1 L'architecture

Le style d'architecture retenue pour notre système est l'architecture en tiers qui spécifie l'organisation **des composants d'exploitation**¹. [3]

Un système client/serveur fait référence à au moins deux types de composants, qui sont les systèmes de base de données en serveur, et les applications qui en exploitent les données en client. [8]

Dans le cadre des architectures d'entreprise, certains concepts et processus sont communs à plusieurs domaines d'activité. Cette caractéristique implique une synchronisation souvent complexe des données entre différents départements de l'entreprise. Le concept d'objet métier consiste à centraliser cette gestion afin d'en maîtriser la complexité. L'objet métier peut être perçu comme un modèle de composant d'exploitation qui s'insère dans le déploiement du système d'entreprise. L'intégration des objets métier sous la forme de composants métiers fait passer l'architecture client/serveur du 2-tiers au 3-tiers, car elle implique un nouveau type de composants d'exploitation qui s'insère entre les clients et les serveurs de données. [8]

A. L'architecture 3-tiers

L'architecture 3-tiers (tiers signifie « partie » en anglais) il consiste à séparer la réalisation des trois parties (stockage des données, logique applicative, présentation). Tout comme dans le client-serveur cette séparation signifie qu'il est possible de déployer chaque partie sur un

1. **des composants d'exploitation** : chaque tiers est représenté par un composant qui joue le rôle de client et/ou de serveur.

serveur indépendant.[9]

Organisation en trois couches :

— **Couche Présentation(interface) :**

Elle correspond à la partie de l'application visible et interactive avec les utilisateurs. On parle d'interface Homme Machine. En informatique, elle peut être réalisée par une application graphique ou textuelle. Elle peut aussi être représentée en HTML pour être exploitée par un navigateur web.[9]

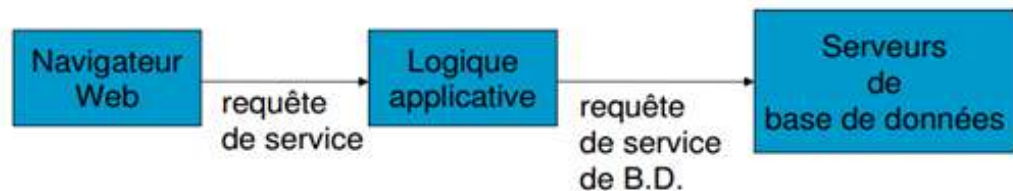
— **Couche logique applicative (Métier) :**

Correspond à la partie fonctionnelle de l'application, celle qui implémente la «logique», et qui décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs, effectuées au travers de la couche présentation.[9]

— **Couche de stockage :**

Elle consiste en la partie gérant l'accès aux gisements de données du système. Ces données peuvent être propres au système, ou gérées par un autre système. La couche métier n'a pas à s'adapter à ces deux cas, ils sont transparents pour elle, et elle accède aux données de manière uniforme.[9]

■ **Architecture 3-tiers (thin client)**



B. Les avantages d'une architecture 3-tiers :

L'architecture à trois niveaux permet de :

1. Bien adaptée pour les applications Web devant supporter plusieurs types de clients.[9]
2. Fournir au système les moyens techniques qui lui permettent de garantir des temps de réponse constants, quel que soit le nombre d'utilisateurs connectés.[9]
3. D'un point de vue développement, la séparation qui existe entre le client, le serveur et le SGBD permet une spécialisation des développeurs sur chaque tiers de l'architecture.[9]

3.1.2 Les grands choix techniques

A. Langage de programmation

Les technologies, logiciels, éditeurs et Framework mis en œuvre pour élaborer l'application sont les suivants :

— **Coté applicatif :**

Le langage choisi pour la réalisation de notre application est PHP.

- **PHP ("HyperText Preprocessor") :** PHP est un langage de scripts orienté objet Open Source, spécialement conçu pour le développement d'applications web et disponible dans plusieurs environnements tels qu'UNIX, WINDOWS ...[10]
PHP est utilisé pour être exécuté par serveur http, mais il peut fonctionner comme n'importe quel langage interprété en utilisant les scripts sur un ordinateur.[10]
Ce choix dépend de sa facilité d'apprentissage dans un délai court et le fait qu'il est destiné spécifiquement aux applications web.



— **Coté interface homme machine (IHM)**

- **JavaScript :** Javascript est un langage de script orienté objet principalement utilisé dans les pages HTML. A l'opposé des langages serveurs (qui s'exécutent sur le site), javascript est exécuté sur l'ordinateur de l'internaute par le navigateur lui-même. Ainsi, ce langage permet une interaction avec l'utilisateur en fonction de ses actions.[11] Comme par exemple, réagir aisément à un changement de focalisation ou un clic de souris.



- **CSS (Cascading Style Sheets) :** Le principe des feuilles de style consiste à regrouper dans un même document des caractéristiques de mise en forme associées à des groupes d'éléments. Il suffit de définir par un nom un ensemble de définition et de caractéristiques de mise en forme, et de l'appeler pour l'appliquer à un texte.[12]



- **ExtJS :**



ExtJS est une bibliothèque écrite en JavaScript, donc toute application réalisée avec cette librairie est exécutée côté client, de ce fait, elle communique avec le serveur en utilisant AJAX, ainsi, tout requête est traité en arrière-plan sans obligation de rafraichissement de la page web ce qui constitue son principal avantage.[10]

B. Serveur d'application

- **WampServer :**



est un serveur virtuel pour la plateforme Windows utilisé dans la programmation web. Il permet de lancer des pages web dynamiques à l'aide du serveur Apache du langage de scripts PHP et de gérer une base de données.[13]

- **Le protocole HTTP** : L'HyperText Transfer Protocol, littéralement "protocole de transfert hypertexte", définit la communication entre un navigateur (le client) et un serveur Web sur le World Wide Web (www).[13]
- **Serveur Apache** : c'est un logiciel libre créé et maintenu au sein de la fondation Apache. C'est le serveur HTTP le plus populaire du World Wide.[13]
- **le protocole SNMP** : signifie Simple Network Management Protocol (traduisez protocole simple de gestion de réseau). Il s'agit d'un protocole qui permet aux administrateurs réseau de gérer les équipements du réseau et de diagnostiquer les problèmes de réseau.[1]

C. Système de gestion de base de données (SGBD) :



De nombreux SGBD sont disponibles sur le marché, partant des SGBD simples jusqu'aux SGBD destinés spécialement aux professionnels, comportant de plus nombreuses fonctionnalités.

Notre choix s'est porté sur MySQL car c'est un système de gestion de base de données qui gère les fichiers constituant une base de données, prend en charge les fonctionnalités de protections et sécurités et fournit un ensemble d'interfaces de programmation (comme PHP). [14]

MySQL est très performant, il propose de très nombreuses fonctionnalités, tout en respectant les standards SQL et il est open source.[15]

D. Environnement de développement

Notre application est développée dans l'environnement SublimText car il est simple et open source.

3.2 Architecture technique de la solution

Pour bien décrire le déploiement de la solution, on utilise le modèle d'UML qui permet de présenter l'architecture de manière simple et compréhensible, comme illustré dans la figure suivante.

3.2.1 Déploiement de la solution

Nous allons présenter les manipulations réalisées sur chaque nœud constituant notre solution en vue de mettre en place l'architecture 3-tiers que nous avons expliqué précédemment.

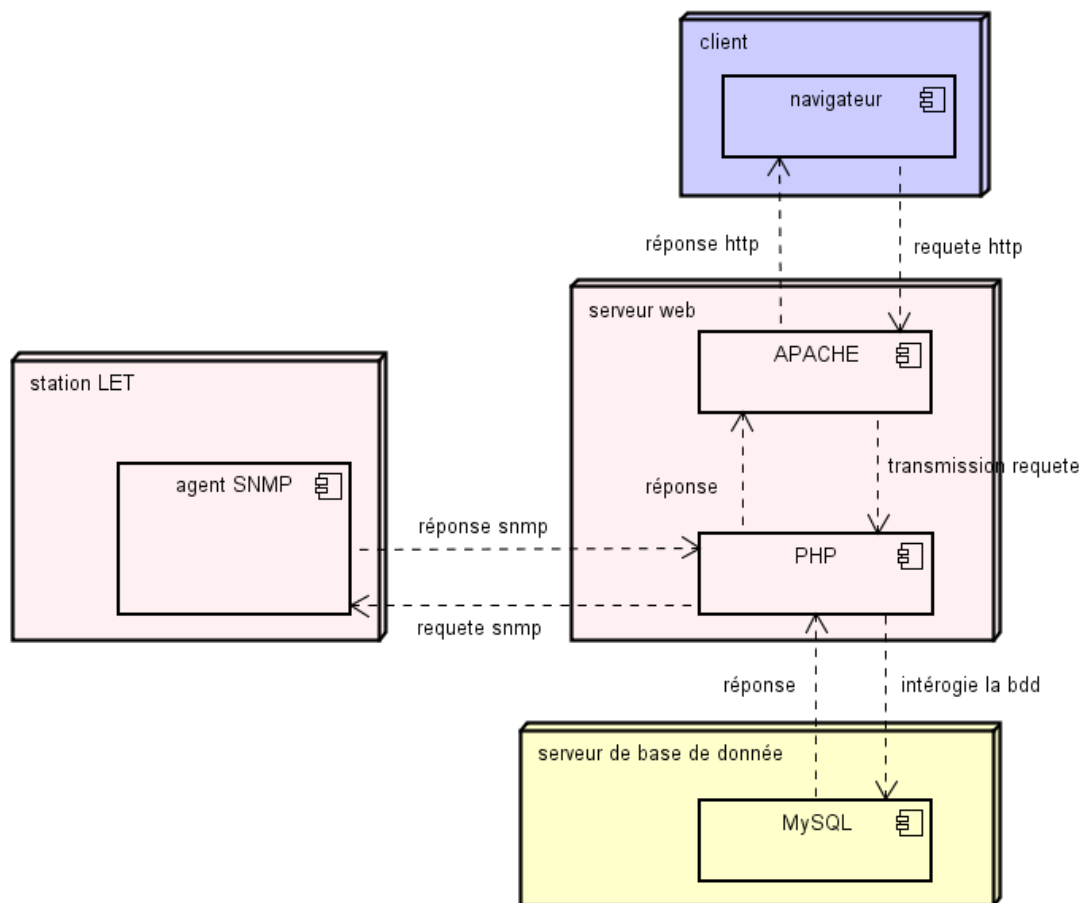


FIGURE 3.1 – Diagramme de déploiement